Novel Machine Learning Algorithms for Personalized Medicine and Insurance

by

Agni Orfanoudaki

B.S., Athens University of Economics and Business (2016)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author .....
Sloan School of Management
May 7, 2021

Certified by .....
Dimitris Bertsimas
Boeing Leaders for Global Operations Professor
Thesis Supervisor

Accepted by .....
Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Novel Machine Learning Algorithms for Personalized Medicine and Insurance

by

Agni Orfanoudaki

Submitted to the Sloan School of Management
on May 7, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Over the past decades, analytics have provided the promise of revolutionizing healthcare, providing more effective, patient-centered, and personalized care. As an increasing amount of data is being collected, computational performance is improved, and new algorithms are developed, machine learning has been viewed as the key analytical tool that will advance healthcare delivery. Nevertheless, until recently, despite the enthusiasm about the potential of *big data*, only a few examples have impacted the current clinical practice. This thesis presents a combination of predictive and prescriptive methodologies that will empower the transition to personalized medicine.

We propose new machine learning algorithms to address major data imperfections like missing values, censored observations, and unobserved counterfactuals. Leveraging a wide variety of data sources, including health and claims records, longitudinal studies, and unstructured medical reports, we demonstrate the potential benefit of analytics in the context of cardiovascular and cerebrovascular diseases. To propel the adoption of these methodologies, we lay the foundations in the area of algorithmic insurance, proposing a quantitative framework to estimate the litigation risk of machine learning models. This work emphasizes interpretability and the design of models that facilitate clinician engagement and integration into the healthcare system.

Part I introduces data-driven algorithms for missing data imputation, clustering, and survival analysis that lie at the intersection of machine learning and optimization. Part II highlights the potential of prescriptive and predictive analytics in the medical field. We develop a new framework for personalized prescriptions and apply it for the treatment of coronary artery disease. Part II also presents predictive models that could support the early diagnosis and improve the management of stroke patients. Finally, Part III proposes a novel risk evaluation methodology that will enable healthcare institutions to manage the risk exposure resulting from the implementation of analytical decision tools.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Leaders for Global Operations Professor

Acknowledgments

"As you set out for Ithaka, hope your road is a long one, full of adventure, full of discovery."
— C.P. Cavafy, *Ithaca*

I would like to thank first of all my advisor Dimitris Bertsimas for encouraging me to apply to the Operations Research Center (ORC), for believing in me from the day we met, and for helping me to become a better and independent researcher ever since. Dimitris has inspired me to always strive for positive impact upon society and view adversity with optimism and determination. Witnessing his own dedication and love for his job was instrumental in my own decision to pursue an academic career. I have been truly fortunate to be his student and I cannot thank him enough for caring so deeply about my personal and professional development as an advisor, as a mentor, and as a friend.

I was very fortunate to have Regina Barzilay and Nikos Trichakis in my thesis committee. Their insightful questions and recommendations, as well as perceptive discussions significantly improved the quality of the dissertation. I am particularly grateful to Nikos Trichakis for his continuous guidance and support throughout my doctoral studies and especially during the academic job market. He has been a great role model for me not only as a researcher but also as an educator. I would also like to thank Vivek Farias for being on my general exam committee and providing me with valuable advice and feedback on research early on. I am particularly thankful to Georgia Perakis who has genuinely cared for my well-being and progress since I arrived in Cambridge.

During the past four years, I have been working in collaboration with Hartford HealthCare. Their unprecedented enthusiasm has been the springboard of various research projects in neurology, cardiac surgery, and COVID-19. This joint effort would not have been possible without the vision and leadership of Barry Stein. I would like to deeply thank him for trusting me to work with his colleagues and greatly supporting me throughout the academic job market. I am also grateful to Amre Nouh and Robert Hagberg for working closely with me to demonstrate the edge that analytics can bring in stroke and cardiac surgery risk estimation.

I am also particularly fortunate to have worked with the team of Stelios Smirnakis and Charlene Ong. Stelios is a truly exemplary, caring, and humble researcher whose ideas and passion prove that science has no limits. Charlene has always impressed me not only with her continuous strive and dedication to achieve her goals but also with her particularly caring personality. I am grateful to Rory B. Weiner for his medical insights and guidance throughout our collaboration on coronary artery disease.

During the course of my doctoral studies, the ORC family has been a formidable source of support and friendships. Collaborating with great people has been a constant source of learning and motivation, and a major reason why I enjoy research. First and foremost, I want to thank Holly Wiberg with whom I have spent countless hours exploring how machine learning can positively affect the medical practice. She is extraordinarily talented, a true force of nature, and one of the kindest people I have ever met. I am very fortunate to be her collaborator and book co-author but most importantly her friend. Colin Pawlowski is an exceptional and meticulous scientist and a thoughtful friend who has taught me a lot beyond missing data imputations. Other than an excellent collaborator, Emma Gibson has been a source of inspiration and help, whose calm and insightful advice I can always rely on. I would like to thank Rebecca Zhang for being a great research partner with exemplary determination and inquisitiveness. It has been a real pleasure to be the mentor and observe the continuous progress of Antonin Dauvin, Alison Borenstein, Emma Chesley, Francois Pierre Caprasse, and Christian Cadisch. They have all been great collaborators and I am sure they will continue to be successful in whatever they decide to dedicate themselves to. I am also grateful to Jack Dunn for introducing me to the Optimal Trees framework and helping me improve my coding skills. I am really proud of the work that we collectively accomplished with the COVIDAnalytics team. It has been a very rewarding experience that provided me with hope and optimism during the COVID-19 pandemic.

I also cannot leave behind the names of many friends I have made from the ORC: Julia Romanski, Bartolomeo Stellato, Elisabeth Paulson, Jean Pauphilet, Arthur Delarue, Tamar Cohen, Matthew Sobiesk, Ryan Cory-Wright, Andrew Li, Léonard Boussioux, Peter Cohen,

Patricio Foncea, Cynthia Zeng, Antoine Dedieu, Jonathan Amar, Jackie Baek, and Emily Meigs. My time at MIT would not have been nearly as special if it was not for my friends in the Greek community: Ilias, Yannis, Ted, Vassilis, Konstantina, Korina, Alexandros, Aggelos, Akis, Giorgos, Sophie, Themis, Chara, Manolis, Konstantinos, Marinos, and Lydia. In addition, I would like to thank my Spanish friends: Carlos and Marga for their warm friendship and for sharing our time together during the COVID-19 pandemic; Jose for the numerous ski adventures; and María and Ferran for the delightful dinners and the exciting tennis matches.

I also cherish my dear friends Katerina, Panayotis, Ioanna, Yannis, Lydia, Sotiris, and Panos who are always on my side no matter the distance. I am deeply grateful to Christos Tarantilis who introduced me to the field of Operations Research and encouraged me to apply for a doctoral degree in the US. I want to thank Cristina Augoustis for dedicating numerous hours of her time to teach me Spanish and Carmen for opening her lovely home to me and my family. I am also grateful to my new family in Spain, Mercedes, Álvaro, and Gabriel, for their continuous support from the other side of the Atlantic.

My sincere and greatest thanks are due to my partner Álvaro Fernández Galiana with whom we navigated this voyage together. Thank you for pushing me to become a better version of myself every day and showing me what unconditional love really means. Our journey has been full of adventures and discoveries. I look forward to what lies ahead of us.

Last but not least, I would like to thank my family in Greece who have been a continuous presence and comfort in my life. My grandparents, Agni and Nikos, have been a precious source of wisdom providing me with their boundless love and unwavering encouragement, prompting me to set high goals and aspirations. Elena, Giorgos, Manos, and Nikos have always been constant supporters and motivators, invigorating me to live a balanced life, of which research is just one part. I am also grateful to my grandmother Marika and my aunt Dimitra who have always loved me from the bottom of their heart.

Finally, I would like to thank my parents for their immeasurable love and support, for their unwavering confidence in my ability, and for standing by my side and helping me realize

my dreams during my Ph.D. and over my whole life. I cannot thank you enough for the guidance and direction you have given me, and for your impact in making me who I am today. I dedicate this thesis to the two of you.

This work has been supported by the National Science Foundation Grant No. 6926678, by Swiss Re, by the Theodore Vassilakis Fellowship Fund, and by the C3.ai Digital Transformation Institute.

To my parents, Nantia and Vaggelis

Contents

Chapter 1

Introduction

1.1 Motivation

ML models have started to play a major role in modern organizations. They are quickly becoming key sources of transformation, disruption, and competitive advantage in today's fast-changing economy and society. At the forefront of scientific fields awaiting this impact are healthcare and insurance. These areas are characterized by uncertainty and variability that pose major challenges in the decision making process of clinicians, policy makers, and business leaders. There are too many parameters to consider, a multitude of potential complications, and a paucity of specialized information for minority groups.

Medical practice is still mostly driven by traditional statistical techniques that draw conclusions from limited sample sizes and risk factors. Data-driven processes have not been integrated in hospital decision making while widely established medical guidelines predominantly address the general population, lacking personalization in the vast majority of cases. Analytics and ML create an unprecedented opportunity for the field, providing new techniques that can harness the power of *big data*, uncovering new insights at the individual level. The goal of this thesis is to show how we can leverage these valuable resources to personalize decision making, and ultimately lead to better outcomes for patients, healthcare institutions, and insurance organizations.

1.1.1 Machine Learning Methods for Healthcare Applications

From electronic health and claims records to longitudinal studies and unstructured medical reports, the healthcare industry uses a wide variety of data sources that require specialized algorithms. The complexity of the problems encountered in this field, along with data imperfections, pose major challenges to realizing its full potential. Part I presents new ML algorithms that leverage optimization techniques to address some of the most common data problems encountered in healthcare applications: missing values, clustering, and censoring. In Chapter 2, we design a new method, *MedImpute*, for imputing missing clinical covariates in multivariate panel data. In Chapter 3, we propose *Interpretable Clustering via Optimal Trees (ICOT)*, a novel unsupervised learning method that recovers interpretable data clusters. In Chapter 4, we address the challenge of censoring with the Optimal Survival Trees (OST) algorithm, generating globally optimized survival tree models. We demonstrate the superior computational performance of these algorithms compared to existing well-established methods on a wide variety of datasets and settings. This first part provides evidence that interpretability need not come at the expense of accuracy, providing a new set of tools that can play a crucial role in the adoption of data-driven models in healthcare.

1.1.2 Prescriptive and Predictive Analytics for Clinical Data

Part II illustrates the transformative power of analytics on the healthcare industry, highlighting our joint research efforts with medical investigators in creating prescriptive models and predictive scores that facilitate clinical decision making. First, we showcase how we can employ available ML algorithms to provide treatment recommendations at the patient level, enabling the transition to *personalized medicine*. Our work uncovers individualized, highly effective treatments by synthesizing observed heterogeneous responses to different regimens among a large pool of patients. Our prescriptive algorithm leverages a combination of generic supervised learning models based on a voting scheme. Its performance is measured via a series of novel evaluation metrics that consider the counterfactual outcomes for multiple treatments under various ground truths. Thus, we assess the accuracy, effectiveness, and robustness

of the prescriptive methodology. We apply this technique to the disease management of Coronary Artery Disease (CAD), one of the clinical conditions with the highest toll on human health (Chapter 5).

Next, we focus on predictive models centered on stroke patients. We highlight both the model derivation and external validation process and propose potential techniques to identify actionable insights from non-linear models. Using structured data from the widely known Framingham Heart Study we present, in Chapter 6, a new model for healthy individuals to estimate the 10-year risk of stroke. This model has been prospectively validated at the Boston Medical Center (BMC) and is undergoing retrospective evaluation at the primary care facilities of Hartford HealthCare. Chapter 7 turns to unstructured information, introducing a comprehensive framework to extract patient information from unstructured radiographic text. Employing a combination of natural language processing and supervised learning methods, we automatically detect the potential presence, location, and acuity of ischemic stroke. This model is now successfully used at Brigham and Women's Hospital and the BMC for patient characterization.

Throughout these investigations, we have aimed at the adoption and clinical integration of these models. In an effort to provide useful and interpretable tools that affect the medical practice, we have developed online web applications that communicate the results of the proposed recommendation systems. These interfaces have been proven crucial in ensuring that the models are used by physicians and deliver real impact in the healthcare organizations where they are deployed.

1.1.3 Algorithmic Insurance

The implementation of data-driven tools in modern healthcare organizations simultaneously disrupts the insurance sector. Analytics have already started to overtake traditional actuarial approaches in health insurance by providing powerful predictive models to estimate the probability of adverse events (i.e., heart attack, cancer, etc.) that may result in a claim. In the future, ML algorithms are expected to play a more central role as they will be called to

replace human decision making in cases where their predictive and prescriptive performance yields better outcomes. This shift gives rise to challenging questions: "Who bears the responsibility if the algorithm's recommendation is wrong?" and "How do we protect the decision maker from erroneous algorithmic predictions?" As artificial intelligence starts to be integrated into the decision making process of organizations, new types of insurance products will have to be developed to protect their owners from risk. Potential examples include image recognition systems applied to radiology that may bear medical liability and extend beyond healthcare to self-driving cars or predictive maintenance algorithms for manufacturing, among many other applications. Part III lays the foundations of a new research area called algorithmic insurance. We present a comprehensive quantitative process to estimate the risk exposure of insurance contracts for algorithmic liability taking into consideration the predictive performance, the interpretability, and generalizability of a binary classification model. We showcase an implementation of our approach in the context of medical malpractice.

1.2 Outline and Main Contributions

The contributions in this thesis can be summarized as follows, listed by chapter.

Chapter 2: Medical Imputations for Time Series

Missing data is a major problem in healthcare research as incomplete information is very often present in patient records. In this chapter, we present a new framework, *MedImpute*, for imputing missing clinical covariates in multivariate panel data. This approach proposes a flexible optimization formulation that can be modified to account for different imputation algorithms. It can use as input a wide range of clinical datasets, including information from clinical trials and Electronic Health Records (EHR), which are of particular research interest in personalized medicine. We summarize our contributions below:

- We formulate the problem of missing data imputation with time series information under the MedImpute framework, extending the OptImpute framework proposed by

Bertsimas et al. (2018) [32]. We focus on a k-Nearest Neighbors (k-NN) formulation to solve the optimization problem and derive a corresponding fast first-order algorithm `med.knn`.

- We conduct a series of computational experiments that test the performance of the method across three real-world datasets, varying the percentage of missing data, the number of observations per individual, and the mechanism of missing data.
- We demonstrate that `med.knn` consistently leads to the best predictive performance and lowest imputation error across all the experiments relative to other state-of-the-art missing data imputation metthods.
- We propose a new custom tuning procedure to efficiently learn the hyperparameters in the optimization problem that leads to both superior scaling performance and better imputation accuracy compared to standard cross-validation.

The work in this chapter appeared at *Machine Learning* [41].

Chapter 3: Interpretable Clustering: An Optimization Approach

Widely established clustering techniques do not provide intuitive reasoning behind data separations, limiting their interpretability. In real-world applications, and specifically in healthcare settings, the latter poses a major barrier to the adoption and integration of ML tools by decision makers. In this chapter, we present a tree-based unsupervised learning method that obtains interpretable clusters with comparable or superior performance to other existing algorithms. Our contributions are as follows:

- We provide a MIO formulation of the unsupervised learning problem that leads to the creation of globally optimal clustering trees, motivating our new algorithm *ICOT*.
- We propose an implementation of our method with an iterative Coordinate Descent (CD) approach that scales to larger problems, well-approximating the globally optimal solution.

- We introduce additional techniques that leverage sampling and the geometric principles of cluster creation to improve the algorithm's efficiency.
- We demonstrate that ICOT is competitive against various clustering approaches using synthetic datasets across multiple internal validation criteria.
- We provide examples of how the algorithm can be used in real-world settings and test the scaling capability of ICOT to large problem instances.

The work in this chapter appeared in *Machine Learning* [31].

Chapter 4: Optimal Survival Trees

Survival analysis addresses the challenges that arise in datasets with censored observations in which the outcome of interest is generally the time until an event, but the exact time of the event is unknown for some individuals. Censored outcomes are ubiquitous in healthcare research and, as a result, ML methods for survival analysis are increasingly popular. We present the OST algorithm that leverages MIO and local search techniques to generate globally optimized survival tree models. We demonstrate that OST improves upon the accuracy of existing survival tree methods, particularly in large datasets. The key contributions of this chapter are:

- We present a survival trees algorithm that utilizes the Optimal Trees framework to generate interpretable trees for censored data.
- We propose a new accuracy metric that evaluates the fit of Kaplan-Meier curve estimates relative to known survival distributions in simulated datasets.
- We evaluate the performance of our method in both simulated and real-world datasets and demonstrate improved accuracy relative to two existing algorithms.
- We provide examples of how the algorithm can be used to predict the risk of adverse events and yield clinical insights in real-world datasets.

The work in this chapter has been submitted for publication [28].

Chapter 5: Personalized Treatment for Coronary Artery Disease Patients: A Machine Learning Approach

In this chapter, our objective is to find the best primary treatment for a CAD patient to maximize the TAE (myocardial infarction or stroke). We propose a data-driven methodology to assign to each patient the regimen with the best predicted outcome simultaneously leveraging multiple regression algorithms. We develop predictive and prescriptive models that provide personalized treatment recommendations and a quantitative framework to evaluate them. The main contributions of this chapter are:

- We present a new methodology to treat right censored patients that utilizes a k-NN approach to estimate the true survival time for real-world data.
- We develop interpretable as well as accurate binary classification and regression models that predict the risk and the timing of a potential adverse event for CAD patients.
- We propose the first prescriptive methodology that utilizes EHR to provide treatment recommendations for CAD, combining multiple state-of-the-art regression models with clinical expertise.
- We introduce a novel evaluation framework to measure the out-of-sample performance of prescriptive algorithms.
- We create an online application where physicians can test the performance of the algorithm in real time bridging the gap with the clinical practice.

The work in this chapter appeared at *Healthcare Management Science* [42].

Chapter 6: The Non-linear Framingham Stroke Risk Score

The vast majority of strokes occur in people without prior history of infarction, highlighting the need for accurate stroke risk assessment tools for healthy individuals. Standard stroke risk scores are based on the assumption that there is a linear relationship between the risk

factors and the prevalence of the disease. However, the mathematical and medical realities suggest that the interactions of these factors are far from linear, and that some variables gain or lose significance due to the absence or presence of other variables. This chapter presents the *N-SRS*; a new model that predicts the 10-year risk of stroke. Leveraging ML algorithms, our risk calculator increases the accuracy of event prediction and uncovers new relationships between the patient characteristics in an interpretable fashion. The main contributions of this chapter are the following:

- We present a new way of leveraging data from longitudinal studies for supervised learning models, allowing multiple instances of the same patient in the training and testing cohort.
- We develop and validate the first non-linear, interpretable, predictive score for the 10-year risk of stroke, using data from the well-known Framingham Heart Study.
- We show how the N-SRS tree structure led to the identification of 23 stroke risk profiles, highlighting the role of new variables in the disease progression, such as hematocrit levels or abnormalities shown in the ECG results.
- We build a dynamic online application as the user-friendly interface of the algorithms for use by clinical providers.

The work in this chapter appeared in PLOS one [257].

Chapter 7: Natural Language Processing Methods to identify ischemic stroke, acuity and location from radiology reports

Expeditious, accurate data extraction could provide considerable improvement in identifying stroke in large datasets, triaging critical clinical reports, and quality improvement efforts. However, the widely available ICD-9/10 codes often misclassify ischemic stroke events and do not distinguish acuity or location. In this chapter, our goal is to develop a tool that will enable the extraction of clinical stroke information from unstructured text in an accurate

and automated fashion. We develop and report a comprehensive framework studying the performance of simple and complex stroke-specific Natural Language Processing (NLP) and supervised learning techniques to determine presence, location, and acuity of ischemic stroke from radiographic text. We summarize our contributions below:

- We collect 60,564 radiology reports from 17,864 patients from two large academic medical centers. Neurology experts labeled 1,359 reports to identify stroke presence, location, and acuity.
- We apply standard text featurization techniques and develop neurovascular specific word GloVe embeddings.
- We train and validate various binary classification algorithms to identify the outcomes of interest from radiology reports.
- We demonstrate that the proposed GloVe word embeddings paired with deep learning had the best discrimination performance of all methods for our three tasks in both the derivation and validation cohort.

The work in this chapter appeared in PLOS one [256].

Chapter 8: Pricing Algorithmic Risk

The insurance industry has not developed tailored contracts that protect ML modelers and decision makers from the litigation risk of algorithmic mistakes. In this chapter, we propose a new class of insurance products for litigation claims against binary classification models as well as quantitative tools to evaluate them. This work provides a comprehensive analytical process to assess the financial risk of such models, laying the foundations in the novel area of algorithmic insurance. The key contributions of this chapter are:

- We propose a quantitative framework that estimates the risk exposure of a model based on its discrimination performance, interpretability, and generalizability.

- We employ an optimization formulation to simultaneously estimate the premium and the litigation risk for a given classification model. We extend the formulation using robust optimization and different types of uncertainty sets around potential scenarios of loss.
- We provide a case-study of breast cancer detection for medical liability and study the effect of the model parameters in computational experiments.

The work in this chapter has been submitted for publication [30].

Part I

Machine Learning Methods for Healthcare Applications

Chapter 2

Imputation of Clinical Covariates in Time Series

Missing data is a common problem in longitudinal datasets which include multiple instances of the same individual observed at different points in time. We introduce a new approach, MedImpute, for imputing missing clinical covariates in multivariate panel data. This approach integrates patient specific information into an optimization formulation that can be adjusted for different imputation algorithms. We present the formulation for a K-nearest neighbors model and derive a corresponding scalable first-order method `med.knn`. Our algorithm provides imputations for datasets with both continuous and categorical features and observations occurring at arbitrary points in time. In computational experiments on three real-world clinical datasets, we test its performance on imputation and downstream predictive tasks, varying the percentage of missing data, the number of Observations Per Patient (OPP), and the mechanism of missing data. The proposed method improves upon both the imputation accuracy and downstream predictive performance relative to the best of the benchmark imputation methods considered. We show that this edge is consistently present both in longitudinal and EHR datasets as well as in binary classification and regression settings.

2.1 Introduction

Machine learning applied to healthcare data can generate actionable insights ranging from predicting the onset of disease to streamlining hospital operations. Statistical models that leverage the variety and richness of clinical data are still relatively rare and offer an exciting avenue for further research [60]. As an increasing amount of information becomes available the medical field expects machine learning to become an indispensable tool for clinicians [251].

This information will come from various clinical and epidemiological sources. Claims records, clinical trials, and data from longitudinal studies have been an invaluable resource for medical research over the past decades. In many of these datasets, data from individual subjects is gathered over time via continuous or repeated monitoring of both risk factors and health outcomes. For example, longitudinal cohort studies are used to discover relationships between exposures of interest and long term health effects including adverse events and chronic disease. By design, these studies mitigate recall bias in participants by collecting data prospectively and prior to knowledge of a possible subsequent event [64].

Another valuable source of clinical data are EHR. Over the past years, widespread uptake of EHR has generated massive datasets that contain quantitative, qualitative, and transactional data [331]. Their hospital adoption has skyrocketed in part due to the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009, which provided $30 billion in incentives for hospitals and physician practices to adopt EHR systems [49]. While primarily designed for archiving patient information and performing administrative healthcare tasks, many researchers have found secondary use of these records for various clinical informatics applications [314]. Because heterogeneous labs, measurements, and notes are recorded for patients during each visit, EHR data has a rich and complex structure with time series information.

However, it is algorithms and not merely datasets that will prove transformative for the medical field [251]. To make progress, we need to develop new statistical tools tailored to clinical applications which address the challenges and leverage common structure encountered

in healthcare data. One of the most important issues is the ubiquitous presence of missing time series data [265], particularly for variables requiring complex, time-sensitive, or resource-intensive procedures to collect. There are many reasons for "missingness", including missed study visits, patients lost to follow-up, missing information in source documents, lack of availability (e.g., laboratory tests that were not performed), and clinical scenarios preventing collection of certain variables (e.g., missing coma scale data in sedated patients) [67]. Thus, creating a consistent dataset for individuals over multiple visits even at the same healthcare organization for a fixed set of covariates remains a challenge. Even in longitudinal studies, where a set of covariates is collected over time, missing data are pervasive and complete ascertainment of all variables is rare [197].

The presence of missing data poses considerable challenges in the analyses and interpretation of clinical investigations' results [357], potentially weakening their validity and leading to biased inferences. Their presence may complicate interpretation or even invalidate an otherwise important study [350]. Many methods commonly used for handling missing values during data analysis can yield biased results, decrease study power, or lead to underestimates of uncertainty, all reducing the chance of drawing valid conclusions [67]. As many statistical models and machine learning algorithms rely on complete datasets, it is key to handle the missing data appropriately.

2.1.1 Review of Methods for Handling Missing Values

In this section, we present some of the most common approaches for missing data imputation. First, we introduce fairly simple and intuitive techniques that do not require the use of sophisticated machine learning methods. We then provide brief descriptions of advanced missing data imputation algorithms, both general purpose methods as well as approaches tailored to medical records and time series.

Excluding observations that contain missing values has been a standard practice for clinical research, primarily due to the lack of interpretable, accurate machine learning methods that can be easily applied by medical researchers [325, 177]. Unsurprisingly, complete case analysis

may suffer from severe bias and the reduced sample size results in lower study power [67]. Recent advances in machine learning have allowed missing values to be accurately imputed prior to running statistical analyses on the complete dataset. The benefit of the latter approach is that once a set (or multiple sets) of complete data has been generated, practitioners can easily apply their own learning algorithms to the imputed dataset. In healthcare settings, often times those datasets contain numerous visits of the same person corresponding to various patterns of missing data. This special structure challenges state-of-the-art missing data methods which do not consider the connection of multiple observations to the same individual [72].

A variety of machine learning approaches have been introduced in the literature to impute missing values ignoring the potential dependency between observations of the same individual. The simplest approach is the `mean` imputation that uses the mean of the observed values to replace those missing for the same covariate [218]. However, `mean` imputation underestimates the variance, ignores the correlation between the features leading to poor imputation outcomes.

Another common method called `bpca` uses the singular value decomposition (SVD) of the data matrix and information from a Bayesian prior distribution on the model parameters to impute missing values. This method outperforms basic SVD methods [250]. In cases where the level of missing data is above 30%, we have found that this method reduces to `mean` imputation, leading to similar biases [119].

Joint modeling assumes the existence of a joint distribution on the entire dataset and a parametric density function on the data given model parameters. Current implementations of the method estimate the model parameters using an Expectation-Maximization (EM) approach in order to maximize the likelihood function. One widely used software package which implements this approach, `Amelia I`, assumes that data are drawn from a multivariate normal distribution [166]. In practice, healthcare data typically violate this condition [325].

Recent review articles indicate that single imputation methods can lead to seriously misleading results and advise us to consider multiple imputation [177, 218]. This approach,

implemented in the software package `mice`, allows for uncertainty about the missing data by creating several different plausible imputed datasets and appropriately combining results obtained from each of them [304]. The `Amelia I` package was extended to multiple imputation in the `Amelia II` algorithm [167]. Multiple imputation entails two stages: (1) generating replacement values for missing data and repeating this procedure many times, resulting in many datasets with replaced missing information, and (2) analyzing the many imputed datasets and combining the results [261]. As a result, multiple imputation methods are slower and require pooling results, which may not be appropriate for certain applications. For example, in clinical applications, where the interpretability of the underlying model matters, a single imputed dataset and simple predictive model may be preferred.

Most recently, Bertsimas et al. [32] proposed a general optimization framework with a predictive model-based cost function that can explicitly handle both continuous and categorical variables and can be used to generate single, as well as multiple, imputations. This optimization perspective has led to new scalable algorithms for more accurate data imputation. We describe this method OptImpute in more detail in Section 2.2.2, which we use as a foundation for the imputation method proposed in this chapter.

The algorithms above are not tailored to multivariate time series datasets despite the fact that covariates may be strongly correlated over time [217]. Preliminary work has been done demonstrating their performance in that setting [371]. Recurrent Neural Network approaches have also been employed to handle missing values in time series among the covariates for a particular prediction task [217, 72]. However, these approaches differ from traditional imputation methods because they also use features derived from the missing pattern itself, and they require that the downstream learning method is a neural network. In contrast, our method produces a single imputed dataset that can be used as training data for any supervised learning method which is preferred for the downstream task.

In practice, simpler techniques are more commonly applied in the panel data setting. Researchers often opt for a moving average approach with a fixed time window using previous observations from the same individual [126]. For example, the last-observation-carried-forward

method is used to impute a present missing value by carrying only the last non-missing value forward for a defined time period [318]. However, these techniques ignore the correlation between covariates which is leveraged by other more advanced imputation methods. There have been a few methods that give weights to instances of the same patient in temporal data. For example, this approach has been applied to adverse drug events monitoring [372]. In addition, similar methods have been applied in the political science and economics fields where time-series cross-sectional data are quite common [315].

2.1.2 Contributions

Given multivariate time series data, we develop a novel imputation method that utilizes optimization and machine learning techniques and outperforms state-of-the-art algorithms. Our contributions are as follows:

1. We formulate the problem of missing data imputation with time series information under the MedImpute framework, extending the OptImpute framework proposed by [32]. Our approach can be adjusted to account for different imputation models based on predictive methods such as K-NN, SVM, and trees. We focus on a K-NN formulation to solve the problem and derive a corresponding fast first-order algorithm `med.knn`. This method provides imputations for datasets with both continuous and categorical features and observations occurring at arbitrary points in time.

2. We design a series of computational experiments on three real-world sets of data with direct clinical implications. We consider the Framingham Heart Study (FHS) and the Parkinson's Progression Markers Initiative (PPMI), two longitudinal datasets with rich time series data recorded at regular time intervals, and EHR data from the Dana Farber Cancer Institute (DFCI), which is less structured and more sparse time series data. We provide a comprehensive framework for our experiments that tests the performance of our method across a diverse range of scenarios, varying parameters including: (1) the percentage of missing data, (2) the number of observations per individual, and (3)

the mechanism of missing data. For the latter, we consider different mechanisms for the longitudinal and EHR datasets corresponding to the different patterns of missing data which are typically observed in real-world datasets. We demonstrate that `med.knn` obtains the best predictive performance and lowest imputation error as we vary the missing percentage from 10% to 50%. In addition, we show that for all datasets, the relative performance of `med.knn` improves as we increase the number of observations per individual. Finally, we demonstrate that `med.knn` performs well on missing patterns commonly encountered in practice for both longitudinal studies and EHR data. These improvements are relative to the best of the comparator methods among `amelia`, `moving average`, `mean`, `bpca`, `mice`, and `opt.knn`, which are described in Section 2.3.

3. We propose a new custom tuning procedure to efficiently learn the hyperparameters in the optimization problem avoiding the use of traditional approaches such as Grid Search. Our methodology allows for decoupling the problem into multiple parts, enabling parallel computation that can decrease the run time. We create synthetic EHR data to test the scaling performance of the algorithm as we increase the number of observations and features. Our results show that the custom tuning approach leads to both superior scaling performance and better imputation accuracy compared to standard cross-validation. The tuning procedure is described in Section 2.2.4 and the scaling experiments with synthetic data are provided in Section 2.4.

The structure of the chapter is as follows. In Section 2.2, we describe our framework for imputation of clinical covariates in time series and proposed method `med.knn`. In Section 2.3, we describe computational experiments on three real-world datasets evaluating both imputation and prediction accuracy. In Section 2.4, we present scaling experiments on simulated clinical datasets. In Section 2.5, we discuss properties of our algorithm and key insights from our experiments. We conclude our work in Section 2.6.

2.2 Methods

In this section, we describe our proposed method for imputation. In Section 2.2.1, we define variables and notation that we use in this chapter. In Section 2.2.2, we review the OptImpute framework for missing data imputation. In Section 2.2.3, we introduce our new framework for imputation MedImpute which directly models clinical covariates in time series, and we present the k-NN based formulation. In Section 2.2.4, we describe a custom tuning procedure to efficiently learn the hyperparameters in the optimization problem. Finally, in Section 2.2.5 we provide the detailed steps of the first-order method `med.knn` that can be used to find high-quality solutions.

2.2.1 Variables and Notation

In this chapter, we consider the single imputation problem for which our task is to fill in the missing values of dataset $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n observations (rows) and p features (columns). Without loss of generality, we assume that the first p_0 features are continuous and that the next $p_1 = p - p_0$ features are categorical, and the missing and known indices are specified by the following sets:

$$\begin{aligned}
\mathcal{M}_0 &= \{(i,d) : \text{entry } x_{id} \text{ is missing},\ 1 \le d \le p_0,\ 1 \le i \le n\}, \\
\mathcal{N}_0 &= \{(i,d) : \text{entry } x_{id} \text{ is known},\ 1 \le d \le p_0,\ 1 \le i \le n\}, \\
\mathcal{M}_1 &= \{(i,d) : \text{entry } x_{id} \text{ is missing},\ p_0+1 \le d \le p_0+p_1,\ 1 \le i \le n\}, \\
\mathcal{N}_1 &= \{(i,d) : \text{entry } x_{id} \text{ is known},\ p_0+1 \le d \le p_0+p_1,\ 1 \le i \le n\}, \\
\mathcal{I} &= \{i : \mathbf{x}_i \text{ has one or more missing values}\}.
\end{aligned} \tag{2.1}$$

Here, $\mathcal{M}_0$, $\mathcal{M}_1$ are the sets of indices of the missing values in the continuous and categorical variables, respectively. Similarly, $\mathcal{N}_0$, $\mathcal{N}_1$ are the sets of indices of the known values in the continuous and categorical variables, respectively. $\mathcal{I}$ is the set of rows which contains at least one missing value.

We suppose that all of the continuous variables are normalized with unit standard

deviation and that the d^{th} categorical variable takes value among k_d classes. Given this data, we introduce the decision variables $\mathbf{W} \in \mathbb{R}^{n \times p_0}$, $\mathbf{V} \in \{1, \ldots, k_{p_0+1}\} \times \ldots \times \{1, \ldots, k_{p_0+p_1}\}$ to be the matrices of imputed continuous and categorical variables, respectively. For each entry x_{id}, w_{id} is the imputed value if $d \in \{1, \ldots, p_0\}$, and v_{id} is the imputed value if $d \in \{p_0 + 1, \ldots, p_0 + p_1\}$. We refer to the full imputation for observation $\mathbf{x}_i$ as $(\mathbf{w}_i, \mathbf{v}_i)$. For the MedImpute method, we also assume that each observation $\mathbf{x}_i$ corresponds to a particular patient with the unique ID y_i observed at time-stamp t_i.

2.2.2 Review of OptImpute

Next, we review the OptImpute framework for general imputation which we use as a foundation for our method. In this approach, we formulate the missing data problem as an optimization problem in which all entries are simultaneously filled in and used as covariates to predict the other entries. Our key decision variables are the imputed values $\{w_{id} : (i, d) \in \mathcal{M}_0\}$ and $\{v_{id} : (i, d) \in \mathcal{M}_1\}$. We will also introduce auxiliary decision variables $\mathbf{Z}$. For any given set of imputed values and a corresponding data $\mathbf{X}$, we associate a cost function $c(\cdot)$ to it. Thus, our objective is to solve the following optimization problem:

$$\begin{aligned} \min \quad & c(\mathbf{Z}, \mathbf{W}, \mathbf{V}; \mathbf{X}) \\ \text{s.t.} \quad & w_{id} = x_{id} && (i, d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} && (i, d) \in \mathcal{N}_1, \\ & (\mathbf{Z}, \mathbf{W}, \mathbf{V}) \in \mathcal{Z}, \end{aligned} \tag{2.2}$$

where $\mathcal{Z}$ is the set of all feasible combinations $(\mathbf{Z}, \mathbf{W}, \mathbf{V})$ of auxiliary vectors and imputations. In this chapter, we only consider an OptImpute formulation based upon K-Nearest Neighbors (K-NN), however it is also possible to consider formulations based upon SVM and trees [32].

In the K-NN formulation, the objective is to impute the missing values so that each point is as close to its K-nearest neighbors as possible. First, we define a distance metric on the

dataset. Given two observations i and j, we say that the distance between them is:

$$d_{ij} := \sum_{d=1}^{p_0} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \mathbb{1}_{\{v_{id} \neq v_{jd}\}}. \tag{2.3}$$

In this distance metric, we weight the contributions from the continuous and categorical variables equally, but it is also possible to introduce a scaling factor to weight these terms differently. Given this distance metric, we introduce the binary variables $\mathbf{Z} \in \{0,1\}^{|\mathcal{I}| \times n}$, where

$$z_{ij} = \begin{cases} 1, & \text{if } j \text{ is among the } K\text{-nearest neighbors of } i \\ & \text{with respect to distance metric (2.3)}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

The OptImpute formulation with the K-NN objective function is

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} z_{ij} \left(\sum_{d=1}^{p_0} (w_{id} - w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1} \mathbb{1}_{\{v_{id} \neq v_{jd}\}} \right) \\ \text{s.t.} \quad & w_{id} = x_{id} && (i,d) \in \mathcal{N}_0, \\ & v_{id} = x_{id} && (i,d) \in \mathcal{N}_1, \\ & z_{ii} = 0 && i \in \mathcal{I}, \\ & \sum_{j=1}^{n} z_{ij} = K && i \in \mathcal{I}, \\ & \mathbf{Z} \in \{0,1\}^{|\mathcal{I}| \times n}, \end{aligned} \tag{2.5}$$

where $\mathcal{I} = \{i : \mathbf{x}_i \text{ has one or more missing values}\}$. Problem (2.5) is non-convex with integer constraints for the categorical variables. In order to solve this problem, the authors find near optimal feasible solutions using first-order methods with random and targeted warm starts, resulting in a new imputation algorithm called `opt.knn` [32].

At a high level, the `opt.knn` algorithm works as follows. The user provides as input an incomplete data matrix $\mathbf{X}$, a convergence threshold $\delta_0 > 0$, and a warm start imputation

$(\mathbf{W}^0, \mathbf{V}^0)$. The output of the algorithm is the full matrix $\mathbf{X}^{imp}$ with the imputed variables. In each iteration, we alternate updating the auxiliary variables $\mathbf{Z}$ and the imputation $(\mathbf{W}, \mathbf{V})$ using either CD or Block Coordinate Descent (BCD). The problem of updating $\mathbf{Z}$ given an imputation reduces to a simple sorting procedure on the distances. To update $(\mathbf{W}, \mathbf{V})$ in CD, we locally optimize each imputed value (w_{id} or v_{id}) one at a time. To update $(\mathbf{W}, \mathbf{V})$ in BCD, for each continuous or categorical feature we solve a Quadratic Optimization problem or a MIO problem, respectively. We continue updating these values until the objective value stops improving by a sufficiently large amount δ_0. Notice that the objective function value is strictly decreasing by at least δ_0 at every iteration until the algorithm terminates. As a result, the number of steps required for the algorithm termination is:

$$T = \frac{1}{\delta_0} c(\mathbf{Z}^0, \mathbf{W}^0, \mathbf{V}^0; \mathbf{X}), \tag{2.6}$$

where $\mathbf{W}^0, \mathbf{V}^0$ are the warmstart values, $\mathbf{X}$ is data, and $\mathbf{Z}^0$ is the initialized auxiliary variables. There are no analytical guarantees that the algorithm will find the globaly optimal solution [358]. We repeat this process for multiple warm starts and take the solution with the best objective value to be the final imputation. The algorithm for a single warm start is summarized in Algorithm 1.

Algorithm 1 `opt.knn`

Input: Incomplete data matrix $\mathbf{X}$,
warm start $[\mathbf{W}^0, \mathbf{V}^0]$,
max number of iterations $T \geq 0$.
Output: $\mathbf{X}^{imp}$ a full matrix with imputed values.
Procedure:
Initialize $t \leftarrow 0$, $\mathbf{W}^* \leftarrow \mathbf{W}^0$, $\mathbf{V}^* \leftarrow \mathbf{V}^0$.
while $t < T$ **do**
① Find the K nearest neighbors for each observation i, and update $\mathbf{Z}^*$ accordingly.
② Update the imputation $(\mathbf{W}^*, \mathbf{V}^*)$, following either BCD
or CD (details in [32]).
③ Increment $t \leftarrow t + 1$.
end while
return $\mathbf{X}^{imp} \leftarrow [\mathbf{W}^*; \mathbf{V}^*]$.

2.2.3 MedImpute

In this section, we present the MedImpute framework for imputation of clinical covariates in time series. We extend the general OptImpute framework by weighting instances of the same person in the imputation model. We focus on the K-NN classifier and provide the specific formulation to solve this problem. Our new framework takes into account the time series structure frequently encountered in healthcare data. In addition, unlike univariate time series methods, this approach leverages statistical correlations between multiple clinical covariates.

Suppose that we are given the same problem setup for single imputation as described in Section 2.2.2. In addition, assume that each observation i corresponds to an individual patient with unique identifier $y_i \in \{1, \ldots, M\}$ recorded at a particular time point. For datasets with multiple observations of individuals over time, we have $M < n$. Define $t_i \in \mathbb{R}^+$ as the number of (days/months/years) after a reference date that observation i was recorded. It follows that $|t_i - t_j|$ is the time difference in (days/months/years) between observations i and j. Note that this framework captures the common structure of many clinical datasets collected over time, including longitudinal studies, insurance claims, and EHR data.

For each clinical covariate $d = 1, \ldots, p$, we introduce the parameters α_d, h_d. We learn α_d and h_d via a custom tuning procedure which we describe in Section 2.2.4. The first learned parameter $\alpha_d \in [0, 1]$ is the relative weight given to the time series component of the objective function for variable d. At the extremes, $\alpha_d = 0$ corresponds to imputing covariate d under the OptImpute objective, and $\alpha_d = 1$ corresponds to imputing covariate d using each individual's time series information independently. The second learned parameter $h_d \in (0, \infty)$ is the halflife parameter for the covariate d. This parameter is called the "halflife" parameter because it is the halflife of an exponential decay function $f(x) = 2^{-x/h_d}$ that we use to determine the relative weights for multiple observations of the same patient.

We introduce this parameter h_d so that observations from the same individual at nearby points in time will be weighted most heavily in the imputation. We make this design decision under the assumption that each clinical covariate can be approximated as a continuous function which is relatively smooth over time. For example, Body Mass Index (BMI) is

a clinical covariate with values that are relatively smooth over time. Under this model, we assume that a BMI measurement from one week ago is more predictive of a patient's current BMI than a BMI measurement from one year ago. However, we do not make any assumptions about how much more/less predictive these different measurements are, only that their relative weights follow an exponential distribution. The halflife of this exponential distribution for covariate d is the modelling parameter that we refer to as h_d.

For each pair of observations i, j, covariate d, and corresponding halflife parameter h_d, define the two derived parameters:

$$\begin{aligned} C_{ijd} &= \begin{cases} 2^{-|t_i - t_j|/h_d}, & \text{if } y_i = y_j, \\ 0, & \text{otherwise,} \end{cases} \\ \overline{C}_{ijd} &= \frac{C_{ijd}}{\sum\limits_{\{j' : y_i = y_{j'}, j' \neq i\}} C_{ijd}}. \end{aligned} \tag{2.7}$$

The first derived parameter C_{ijd} is the relative weight that observation j is given for time-series based imputation of observation i in covariate d. Note that this parameters is only non-zero when $y_i = y_j$, i.e. i and j are observations from the same patient. For example, if $h_d = 7$ days, then past observations of covariate d from one week and two weeks ago from the same patient would be given relative weights 0.5 and 0.25, respectively. The second derived parameter, $\overline{C}_{ijd}$, is the normalized variation of C_{ijd}. In particular, $\overline{C}_{ijd}$ is the relative weight that observation j is given to impute observation i in covariate d, divided by the sum of all relative weights of observations from the same patient in covariate d.

The MedImpute formulation with the K-NN objective function is

$$
\begin{aligned}
\min \quad & \frac{1}{K}\sum_{i\in\mathcal{I}}\sum_{j=1}^{n} z_{ij}\left(\sum_{d=1}^{p_0}(1-\alpha_d)(w_{id}-w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1}(1-\alpha_d)\mathbb{1}_{\{v_{id}\neq v_{jd}\}}\right) \\
& \qquad + \sum_{i\in\mathcal{I}}\sum_{j=1}^{n}\left(\sum_{d=1}^{p_0}\alpha_d\overline{C}_{ijd}(w_{id}-w_{jd})^2 + \sum_{d=p_0+1}^{p_0+p_1}\alpha_d\overline{C}_{ijd}\mathbb{1}_{\{v_{id}\neq v_{jd}\}}\right) \\
\text{s.t.} \quad & w_{id} = x_{id} \qquad (i,d)\in\mathcal{N}_0, \\
& v_{id} = x_{id} \qquad (i,d)\in\mathcal{N}_1, \\
& z_{ii} = 0 \qquad i\in\mathcal{I}, \\
& \sum_{j=1}^{n} z_{ij} = K \qquad i\in\mathcal{I}, \\
& \mathbf{Z}\in\{0,1\}^{|\mathcal{I}|\times n},
\end{aligned}
\tag{2.8}
$$

where $\mathcal{I} = \{i : \mathbf{x}_i \text{ has one or more missing values}\}$ and α_d, $\overline{C}_{ijd}$ are constants. This problem is equivalent to (2.5) plus a penalty term in the objective for each feature d with different weights α_d in order to account for instances of the same person in the dataset. At the optimal solution, the objective function is the sum of the distances from each point to its K-nearest neighbors with respect to distance metric (2.3), plus the sum of the distances from each point to other observations from the same individual.

We derive a fast algorithm to provide high quality solutions to this problem using first order methods with random restarts, alternatively updating the binary variables and the imputed values as in `opt.knn` [26]. In Algorithm 2, we summarize the `med.knn` method for a single warm start. In the next section, we describe the steps of this algorithm in detail.

MedImpute provides a flexible framework that can be easily extended as well. For example, we may consider other predictive models besides K-NN such as support vector machines and decision tree based methods by adjusting the objective functions of the corresponding OptImpute formulations appropriately. We refer the reader to [32] for more discussion on these alternate formulations, which is a possible area of future work. In these cases, we add the same penalty term to the objective functions that we added in formulation (2.8), and

we solve using first-order methods with random starts. In this manuscript, we focus on the K-NN formulation due to the method's simplicity that is close to the medical practice. The idea of imputing a patient's missing values using the mean or the mode of the covariates from the most similar individuals to that observation is intuitive. Various implementations of the heuristic K-NN approach are already widely accepted and used in practice [88]. For these reasons, we decided to extend upon those combining the time series component and an optimization framework.

The method can also be adapted to a multiple imputation setting. However, while multiple imputation has been considered for several years to be the most accurate method for dealing with missing data [299], there is a tradeoff because single imputation is more interpretable. In particular, with single imputation we obtain one downstream predictive model that can be easily presented and explained to an entire clinical team, which is a critical step in the process of data-driven medical research [316].

2.2.4 Learning α_d and h_d

In this section, we describe a custom tuning procedure to efficiently learn α_d and h_d, which are hyperparameters in the optimization problem (2.8). We run this custom tuning procedure as a pre-processing step before the `med.knn` algorithm, which allows us to learn these parameters without using cross-validation. This is a heuristic procedure which decouples the problem into multiple parts, first learning h_d for each covariate, and then learning α_d for each covariate. As a result, this custom tuning procedure is more computationally efficient and scales to larger problem sizes than cross-validation. In Section 2.4, we present the results from computational experiments comparing the speed and imputation accuracy of this custom tuning procedure against a traditional cross-validation method for selecting α_d and h_d.

In the first step of the custom tuning procedure, we learn the halflife parameter h_d for each covariate. As in cross-validation, we tune the halflife parameters over a discrete range of values, denoted as $\mathcal{H}$. For example, in the computational experiments, we set $\mathcal{H} = \{1, 7, 30, 90, 365, 1000\}$, representing halflife values of 1 day, 1 week, 1 month, etc. For

each covariate d, we compute the leave-one-out error for each halflife value $h_d \in \mathcal{H}$. In particular, to compute the leave-one-out error for the halflife value h_d, first we derive the weights $\overline{C}_{ijd}$, then we impute the known values in covariate d using these weights, and finally we compute the sum-of-squared errors. Afterwards, we select the halflife parameter h_d which yields the lowest leave-one-out error.

For each continuous covariate $d \in \{1, \ldots, p_0\}$, the leave-one-out error is defined as:

$$\sum_{\{i:(i,d)\in\mathcal{N}_0\}} (x_{id} - \hat{w}_{id})^2, \tag{2.9}$$

where:

$$\hat{w}_{id} := \sum_{j=1}^{n} \overline{C}_{ijd} x_{jd}. \tag{2.10}$$

Here, $\hat{w}_{id}$ is equivalent to the MedImpute imputation of a continuous covariate x_{id} when $\alpha_d = 1$. For each categorical covariate $d \in \{p_0 + 1, \ldots, p_0 + p_1\}$, the leave-one-out error is defined as:

$$\sum_{\{i:(i,d)\in\mathcal{N}_1\}} \mathbb{1}_{\{x_{id} \neq \hat{v}_{id}\}}, \tag{2.11}$$

where:

$$\hat{v}_{id} := \arg\max_{v_{id}} \sum_{j=1}^{n} \overline{C}_{ijd} \mathbb{1}_{\{x_{jd}=v_{id}\}}. \tag{2.12}$$

Intuitively, $\hat{v}_{id}$ is the weighted mode of covariate d, where the weights are $\overline{C}_{ijd}$. This is equivalent to the MedImpute imputation of the categorical covariate x_{id} when $\alpha_d = 1$.

Note that we are able to learn h_d independently from α_d because the selection of $\overline{C}_{ijd}$ which minimizes the objective function (2.8) for any fixed value of α_d also minimizes the objective function for any choice of $\alpha_d \in [0, 1]$. Similarly, we can learn the halflife parameters $\{h_1, h_2, \ldots, h_p\}$ independently from one another, because the optimal choice of h_d which minimizes the objective function (2.8) does not depend upon the values of $\{h_1, \ldots, h_{d-1}, h_{d+1}, \ldots, h_p\}$. Therefore, in this custom tuning procedure, we take advantage of this fact, and tune each of the halflife parameters as an initial step.

In the second step of the custom tuning procedure, we learn the MedImpute weight parameter α_d for each covariate. As in cross-validation, we tune the MedImpute weight parameters over a discrete range of values, denoted as $\mathcal{A}$. For example, in the computational experiments, we set $\mathcal{A} = \{0, 0.05, \ldots, 0.95, 1.0\}$, denoting relative MedImpute weights of 0%, 5%, ..., 100%, respectively. For each covariate d, we compute the k-fold error for each MedImpute weight value $\alpha_d \in \mathcal{A}$. In particular, to compute the k-fold error for the MedImpute weight value α_d, first we split the dataset into k subsets (aka "folds"), next we impute each data subset using the rest of the subsets as training data, and finally we compute the total sum-of-squared errors across all of the folds. We select the MedImpute weight parameter α_d which yields the lowest k-fold error. For continuous covariates, the k-fold error is defined as:

$$\sum_{\ell=1}^{k} \sum_{\{i:(i,d)\in\mathcal{N}_0^\ell\}} (x_{id} - \hat{w}_{id}^\ell)^2, \tag{2.13}$$

where $\mathcal{N}_0^\ell$ are the known continuous values in the ℓth fold. The imputed values $\hat{w}_{id}^\ell$ are given by:

$$\hat{w}_{id}^\ell := (1-\alpha_d) w_{id}^{\mathrm{OPT}_\ell} + \alpha_d \sum_{\{i:(i,d)\in\mathcal{N}_0\setminus\mathcal{N}_0^\ell\}} \overline{C}_{ijd} x_{jd}, \tag{2.14}$$

where $w_{id}^{\mathrm{OPT}_\ell}$ is the OptImpute imputation of x_{id} using the data from the other $k-1$ folds, and $\mathcal{N}_0 \setminus \mathcal{N}_0^\ell$ are the known continuous values not in the ℓth fold. For categorical covariates, the k-fold error is defined as:

$$\sum_{\ell=1}^{k} \sum_{\{i:(i,d)\in\mathcal{N}_1^\ell\}} \mathbb{1}_{\{x_{id}\neq\hat{v}_{id}^\ell\}}, \tag{2.15}$$

where $\mathcal{N}_1^\ell$ are the known categorical values in the ℓth fold. The imputed values $\hat{v}_{id}^\ell$ are given by:

$$\hat{v}_{id}^{\ell} := \arg\max_{v_{id}} \left[(1-\alpha_d) \mathbb{1}_{\{v_{id}^{\mathrm{OPT}_\ell} = v_{id}\}} + \alpha_d \sum_{\{i:(i,d)\in\mathcal{N}_0\setminus\mathcal{N}_0^{\ell}\}} \overline{C}_{ijd} \mathbb{1}_{\{x_{jd}=v_{id}\}} \right]. \tag{2.16}$$

where $v_{id}^{\mathrm{OPT}_\ell}$ is the OptImpute imputation of x_{id} using the data from the other $k-1$ folds, and $\mathcal{N}_1 \setminus \mathcal{N}_1^{\ell}$ are the known categorical values not in the ℓth fold. Intuitively, $\hat{v}_{id}^{\ell}$ is the weighted mode of the OptImpute value and the other known values of the same covariate, where the weights are $(1-\alpha_d)$ and $\alpha_d \overline{C}_{ijd}$, respectively.

Finally, we note that there is another hyperparameter that we may tune for the `med.knn` algorithm, K, which is the number of nearest-neighbors. In the computational experiments, we fix $K = 10$, which works well for the datasets that we consider here. Previously, it has been shown that the OptImpute methods are relatively robust even if their hyperparameters are misspecified [32]. Thus, while the accuracy of the `med.knn` algorithm can be improved slightly by tuning over K, the relative improvement in imputation accuracy is outweighed by the increased computational costs.

Algorithm 2 `med.knn`

Input: Incomplete data matrix $\mathbf{X}$,
warm start $[\mathbf{W}^0, \mathbf{V}^0]$,
max number of iterations $T \geq 0$,
weight parameters $\{\alpha_d\}_{d=1}^{p}$,
halflife parameters $\{h_d\}_{d=1}^{p}$.
Output: $\mathbf{X}^{imp}$ a full matrix with imputed values.
Procedure:
Initialize $t \leftarrow 1$, $\mathbf{W}^* \leftarrow \mathbf{W}^0$, $\mathbf{V}^* \leftarrow \mathbf{V}^0$.
while $t < T$ **do**
① Find the K nearest neighbors for each observation i, and update $\mathbf{Z}^*$ accordingly.
② Update the imputation $(\mathbf{W}^*, \mathbf{V}^*)$, following either BCD
or CD (details in Section 2.2.5).
③ Increment $t \leftarrow t+1$.
end while
return $\mathbf{X}^{imp} \leftarrow [\mathbf{W}^*; \mathbf{V}^*]$.

2.2.5 The `med.knn` algorithm

In this section, we provide details for the updates in the `med.knn` imputation algorithm. This is a first-order method to find locally optimal solutions to Problem (2.5). As in the `opt.knn` algorithm, in this algorithm we alternatively update $\mathbf{Z}$ and $(\mathbf{W}, \mathbf{V})$ until the solution converges. The update for $\mathbf{Z}$ is identical to the one for `opt.knn`, and is computed with a simple sorting procedure on the distances. However, the update for $(\mathbf{W}, \mathbf{V})$ is modified and depends upon the MedImpute parameters α_d, C_{ijd}. As in `opt.knn`, we can update the values of $(\mathbf{W}, \mathbf{V})$ either with BCD or CD which are described in the following subsections. The `opt.knn` updates for both BCD and CD are equivalent to the corresponding `med.knn` updates when $\alpha_d = 0$ for all $d = 1, \ldots, p$.

Block Coordinate Descent

In this approach, we update all of the imputed values at once. We call this approach BCD because we update the variables $(\mathbf{W}, \mathbf{V})$ as an entire block, keeping $\mathbf{Z}$ fixed. Our formulation Problem (2.8) decomposes by dimension into p_0 Quadratic Optimization problems for the continuous features and p_1 MIO problems for the categorical features. To update the imputed values $\mathbf{w}^d$ for continuous feature $d = 1, \ldots, p_0$, we solve:

$$\begin{aligned} \min_{\mathbf{w}^d} \quad & \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} z_{ij}(1-\alpha_d)(w_{id} - w_{jd})^2 + \sum_{i \in \mathcal{I}} \sum_{j=1}^{n} \alpha_d \overline{C}_{ijd}(w_{id} - w_{jd})^2 \\ \text{s.t.} \quad & w_{id} = x_{id} \quad (i,d) \in \mathcal{N}_0. \end{aligned} \tag{2.17}$$

Taking the partial derivative of the objective function with respect to w_{id} for some missing entry $(i, d) \in \mathcal{M}_0$ and setting it to zero, we obtain after some simplifications:

$$
\begin{aligned}
0 =& \left((1-\alpha_d)K + \alpha_d + \sum_{j \in \mathcal{I}} [(1-\alpha_d) z_{ji} + \alpha_d \overline{C}_{jid}] \right) w_{id} \\
& - \sum_{(j,d) \in \mathcal{M}_0} [(1-\alpha_d)(z_{ij} + z_{ji}) + \alpha_d (\overline{C}_{ijd} + \overline{C}_{jid})] w_{jd} \\
& - \sum_{(j,d) \in \mathcal{N}_0} [(1-\alpha_d)(z_{ij} + \mathbb{1}_{\{j \in \mathcal{I}\}} z_{ji}) + \alpha_d (\overline{C}_{ijd} + \mathbb{1}_{\{j \in \mathcal{I}\}} \overline{C}_{jid})] x_{jd}.
\end{aligned} \tag{2.18}
$$

This follows directly from equation (9) in [32]. For each feature $d = 1, \ldots, p_0$, we have a system of equations of the above form which we can solve to determine the optimal imputed values $w_{id}, (i,d) \in \mathcal{M}_0$. Simplifying the notation, suppose that the missing values for the dimension d are $\widetilde{\mathbf{w}}^d := (w_{1d}, \ldots, w_{ad})$ and the known values are $\mathbf{x}^d := (x_{(a+1)d}, \ldots, x_{nd})$. Then the set of optimal imputed values $w_{id}^d, (i,d) \in \mathcal{M}_0$ is the solution to the linear system

$$
((1-\alpha_d)\mathbf{Q} + \alpha_d \mathbf{P}) \widetilde{\mathbf{w}}^d = ((1-\alpha_d)\mathbf{R} + \alpha_d \mathbf{Y}) \mathbf{x}^d, \tag{2.19}
$$

where the matrices $\mathbf{Q}$, $\mathbf{P}$, $\mathbf{R}$, and $\mathbf{Y}$ are defined as

$$
\mathbf{Q} = \begin{bmatrix}
K + \sum_{j \in \mathcal{I}} z_{j1} - 2z_{11} & -z_{12} - z_{21} & \cdots & -z_{1a} - z_{a1} \\
-z_{21} - z_{12} & K + \sum_{j \in \mathcal{I}} z_{j2} - 2z_{22} & \cdots & -z_{2a} - z_{a2} \\
\vdots & \vdots & \ddots & \vdots \\
-z_{a1} - z_{1a} & -z_{a2} - z_{2a} & \cdots & K + \sum_{j \in \mathcal{I}} z_{ja} - 2z_{aa}
\end{bmatrix}, \tag{2.20}
$$

$$
\mathbf{P} = \begin{bmatrix}
\sum_{j \in \mathcal{I}} \overline{C}_{j1d} - 2\overline{C}_{11d} & -\overline{C}_{12d} - \overline{C}_{21d} & \cdots & -\overline{C}_{1ad} - \overline{C}_{a1d} \\
-\overline{C}_{21d} - \overline{C}_{12d} & \sum_{j \in \mathcal{I}} \overline{C}_{j2d} - 2\overline{C}_{22d} & \cdots & -\overline{C}_{2ad} - \overline{C}_{a2d} \\
\vdots & \vdots & \ddots & \vdots \\
-\overline{C}_{a1d} - \overline{C}_{1ad} & -\overline{C}_{a2d} - \overline{C}_{2ad} & \cdots & \sum_{j \in \mathcal{I}} \overline{C}_{jad} - 2\overline{C}_{aad}
\end{bmatrix}, \tag{2.21}
$$

$$\mathbf{R} = \begin{bmatrix} z_{1(a+1)} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} z_{(a+1)1} \dots z_{1n} + \mathbb{1}_{\{n\in\mathcal{I}\}} z_{n1} \\ \vdots \qquad\qquad \vdots \\ z_{a(a+1)} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} z_{(a+1)a} \dots z_{an} + \mathbb{1}_{\{n\in\mathcal{I}\}} z_{na} \end{bmatrix}, \tag{2.22}$$

$$\mathbf{Y} = \begin{bmatrix} \overline{C}_{1(a+1)d} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} \overline{C}_{(a+1)1d} \dots \overline{C}_{1nd} + \mathbb{1}_{\{n\in\mathcal{I}\}} \overline{C}_{n1d} \\ \vdots \qquad\qquad \vdots \\ \overline{C}_{a(a+1)d} + \mathbb{1}_{\{(a+1)\in\mathcal{I}\}} \overline{C}_{(a+1)ad} \dots \overline{C}_{and} + \mathbb{1}_{\{n\in\mathcal{I}\}} \overline{C}_{nad} \end{bmatrix}. \tag{2.23}$$

Without loss of generality, there exists a closed-form solution

$$\widetilde{\mathbf{w}}^d = ((1-\alpha_d)\mathbf{Q} + \alpha_d \mathbf{P})^{-1}((1-\alpha_d)\mathbf{R} + \alpha_d \mathbf{Y})\mathbf{x}^d \tag{2.24}$$

to this system of equations for each feature $d = 1, \dots, p_0$. To update the imputed values $\mathbf{v}^d$ for each categorical feature $d = (p_0+1), \dots, p$, we solve the following MIO problem:

$$\begin{aligned} \min_{\mathbf{v}^d} \quad & \sum_{i\in\mathcal{I}} \sum_{j=1}^{n} ((1-\alpha_d) z_{ij} + \alpha_d \overline{C}_{ijd}) y_{ij} \\ \text{s.t.} \quad & v_{id} = x_{id} && (i,d) \in \mathcal{N}_1, \\ & v_{id} - v_{jd} \le y_{ij} k_d && i = 1, \dots, n, j = 1, \dots, n, \\ & v_{jd} - v_{id} \le y_{ij} k_d && i = 1, \dots, n, j = 1, \dots, n, \\ & y_{ij} \in \{0,1\}^{|\mathcal{I}| \times n}. \end{aligned} \tag{2.25}$$

This is a MIO problem, which is practically solvable as the BCD update for `opt.knn`. Since the BCD update step requires inverting a matrix with $O(n^2)$ entries and solving an optimization problem with $O(n^2)$ binary variables, this method works best for smaller problem sizes $n \le 10,000$.

Coordinate Descent

In CD, we update the imputed values one at a time. In order to update the imputed value for x_{id}, we fix all of the variables in Problem (2.8) except for w_{id} or v_{id} and solve the corresponding one-dimensional optimization problem. This results in fast, closed-form updates for both the continuous and categorical variables. Each $w_{id}, (i,d) \in \mathcal{M}_0$ is imputed as the minimizer of the following:

$$\min_{w_{id}} \quad \sum_{r\in\mathcal{I}}\sum_{j=1}^{n} z_{rj} \sum_{d=1}^{p_0}(1-\alpha_d)(w_{rd}-w_{jd})^2 + \sum_{r\in\mathcal{I}}\sum_{j=1}^{n}\sum_{d=1}^{p_0}\alpha_d\overline{C}_{rjd}(w_{rd}-w_{jd})^2. \tag{2.26}$$

Solving the above gives the closed-form solution for every $(i,d) \in \mathcal{M}_0$:

$$w_{id} = \frac{\sum_{j=1}^{n}((1-\alpha_d)z_{ij}+\alpha_d\overline{C}_{ijd})w_{jd} + \sum_{j\in\mathcal{I}}((1-\alpha_d)z_{ji}+\alpha_d\overline{C}_{jid})}{K + \sum_{j=1}^{n}\alpha_d\overline{C}_{ijd} + \sum_{j\in\mathcal{I}}((1-\alpha_d)z_{ji}+\alpha_d\overline{C}_{jid})}. \tag{2.27}$$

Similarly, each categorical variable $v_{id}, (i,d) \in \mathcal{M}_1$ is imputed as the minimizer of the following:

$$\min_{v_{id}} \quad \sum_{r\in\mathcal{I}}\sum_{j=1}^{n} z_{rj} \sum_{d=p_0+1}^{p_0+p_1}(1-\alpha_d)\mathbb{1}_{\{v_{rd}\neq v_{jd}\}} + \sum_{r\in\mathcal{I}}\sum_{j=1}^{n}\sum_{d=p_0+1}^{p_0+p_1}\alpha_d\overline{C}_{rjd}\mathbb{1}_{\{v_{rd}\neq v_{jd}\}}. \tag{2.28}$$

Suppose that the value of categorical variable v_{id} is one of k_d distinct categories $\{1,2,\ldots,k_d\}$. Then, the solution to problem (2.28) is

$$\arg\max_{k\in\{1,\ldots,k_d\}}\left[\sum_{j=1}^{n}\left((1-\alpha_d)z_{ij}+\alpha_d\overline{C}_{ijd}\right)\mathbb{1}_{\{v_{jd}=k\}} + \sum_{j\in\mathcal{I}}\left((1-\alpha_d)z_{ji}+\alpha_d\overline{C}_{ijd}\right)\mathbb{1}_{\{v_{jd}=k\}}\right]. \tag{2.29}$$

Here, we set the imputed variable to be the value with the highest frequency in the neighborhood, with instances of the same person i receiving additional weight calibrated by the parameters $\{\overline{C}_{ijd}\}_{j=1}^{n}$ and α_d.

This approach scales to large problem sizes (n in the 100,000's), and it is the method that we implement for the computational experiments.

2.3 Computational Experiments on Real-World Clinical Datasets

In this section, we run a series of computational experiments testing the performance of `med.knn` imputing missing values in real-world clinical datasets. In Section 2.3.1, we provide an overview of the three datasets and their baseline characteristics. In Section 2.3.2, we describe the mechanisms for generating Missing Not At Random (MNAR) data that are used in some of the experiments. In Section 2.3.3, we describe the setup of the computational experiments, and we describe the imputation methods that we run for comparison across all of the computational experiments. In Section 2.3.4, we report the results of the experiments on the imputation tasks. In Section 2.3.5, we report the results of the experiments on the downstream predictive tasks. In Section 2.3.6 we discuss the results and major takeaways from the computational experiments.

2.3.1 Description of Real-World Clinical Datasets

In this section we describe the three real-world clinical datasets used in the computational experiments.

Framingham Heart Study (FHS) dataset

The FHS was started in 1948 with the goal of observing a large population of healthy adults over time to better understand the factors that lead to cardiovascular disease. Over 80 variables were collected from 5,209 people at a time for more than 40 years. The FHS is arguably the most influential longitudinal study in the field of cardiovascular and cerebrovascular research. This data has now been used in more than 2,400 studies and is considered one of the top 10 cardiology advances of the twentieth century alongside the electrocardiogram and open-heart surgery [209].

In our computational experiments, we consider all individuals from the FHS Original Cohort [122] with 10 or more observations, which includes $M = 1,107$ unique patients.

For each patient, we take the 10 most recent observations, so the dataset has $n = 11,070$ observations total. We include $p = 13$ continuous (Age, BMI, Systolic Blood Pressure, High-Density Lipoproteins, Hematocrit, Blood Glucose levels) and categorical covariates (Gender, Smoking, presence of Cardiovascular Disease, presence of Atrial Fibrillation, presence of diabetes, currently under prescription of antihypertensive medication, presence of Left Ventricular Hypertrophy from ECG results).

Overall, there are 12.56% missing values in the FHS dataset. Due to the design of the longitudinal study, the 10 observations for each patient occur at regular intervals spaced 2 years apart, for a total span of 18 years. For the imputation tasks, we add in additional missing values to the FHS dataset, and evaluate the accuracy of `med.knn` and comparison methods against the ground-truth values. For the downstream tasks, we evaluate classification models which predict 10-year risk of stroke given the imputed training data.

Dana Farber Cancer Institute (DFCI) dataset

The DFCI dataset was obtained from a recently published work on predicting mortality in late-stage cancer patients [36]. In this study, the authors retrospectively obtained patient data from EHR and linked Social Security Administration mortality data for cancer patients at the Dana Farber Cancer Institute / Brigham and Women's Cancer Center from 2004 through 2014. Predictive models were fit for the entire population and individual cancers, including breast, lung, colorectal, kidney, and prostate cancer. Study eligibility required adult patients that have received at least one anticancer treatment over the course of their care, including chemotherapy, immunotherapy, and targeted therapy.

In our computational experiments, we consider all patients with late-stage breast cancer from the DFCI dataset. Each observation corresponds to a patient initiating an anticancer regimen which was systematically recorded in the hospital's database. As a result, for every patient who followed more than one regimen, multiple observations were collected. For each patient, we include all of their observations in either the training set or testing set, respectively. In total, we have 12,206 observations that correspond to 5,987 unique patients.

This includes 3,228 individuals who have just one line of therapy and therefore only appear once in this dataset. For each observation, there are 106 covariates which describe the patient at that point in time, including demographics, lab tests, vital signs, current medications, medical history, biomarkers, and variables derived from the patient's temporal EHR history.

Overall, there are 10.79% missing values in the DFCI dataset. Due to the nature of this observational study, the observations for each patient occur at irregular intervals, which correspond to hospital visits. In addition, in the dataset each patient has anywhere from 1 to 12 observations. For the imputation tasks, we add in additional missing values to the DFCI dataset, and evaluate the accuracy of `med.knn` and comparison methods against the ground-truth values. For the downstream tasks, we evaluate classification models which predict 60-day risk of mortality given the imputed training data.

Parkinson's Progression Markers Initiative (PPMI) dataset

The PPMI was a landmark observational clinical study with the aim to comprehensively evaluate patient cohorts using imaging, biologic sampling as well as clinical and behavioral data to identify biomarkers of Parkinson's disease progression [227].

In our computational experiments, we consider data from the PPMI baseline examination as well as the following three years of follow-up. In this longitudinal study, 20 patients appeared only in one follow-up examination, 33 in two while the rest of the population participated in all 352 clinical evaluations. As a result, in total we have 1,547 observations corresponding to 405 distinct patients. For each observation, there are 116 covariates which describe the demographic characteristics, the results of behavioral tests, clinical test results, as well as the presence or absence of genetic mutations related to the disease.

Overall, there are 2.61% missing values in the PPMI dataset. Due to the design of the longitudinal study, the 4 observations for each patient occur at regular intervals spaced 1 year apart, for a total span of 4 years. For the imputation tasks, we add in additional missing values to the PPMI dataset, and evaluate the accuracy of `med.knn` and comparison methods against the ground-truth values. For the downstream tasks, we evaluate regression models

which predict the Montreal Cognitive Assessment (MoCA) score one year in advance. The MoCA score is a rapid screening instrument for mild cognitive dysfunction, a clinical state that often progresses to dementia [245].

2.3.2 Mechanisms for Generating Missing Not at Random (MNAR) data

Missing data can either be Missing Completely At Random (MCAR), Missing At Random (MAR), or Missing Not at Random (MNAR) [218]. The type of missingness can be determined through an understanding of the specific feature and what systematic biases may exist in its collection process. Different types of missingness must be treated differently for meaningful analysis. In reality, missing data are most commonly associated with the MNAR category where the presence of unknown values is systematically related to unobserved factors.

In this section, we describe mechanisms for generating MNAR data for our computational experiments. We consider different mechanisms for the longitudinal and EHR datasets corresponding to the different patterns of missing data which are typically observed in real-world datasets. First, we describe the missing data mechanism that we use for the MNAR experiments on the two longitudinal datasets: FHS and PPMI. Then, we describe the missing data mechanism that we use for the MNAR experiments on the EHR dataset: DFCI.

For all MNAR experiments, the total percentage of missing data is fixed to 30%. For each individual experiment, we assume that the dataset is (γ30% MNAR, $(1-\gamma)$30% MCAR), where γ is a constant that we select between 0 and 1. To generate the missing data patterns, first we generate the γ30% MNAR patterns, and then we randomly select an additional $(1-\gamma)$30% subset of the data to be MCAR. In the following two sections, we describe the specific ways that we generate MNAR data for longitudinal studies and EHR data, which are influenced by real-world missing data mechanisms.

MNAR Mechanism for Data from Longitudinal Studies

In longitudinal studies, missing data patterns often result from changes in the experiment design. Researchers may decide to include an additional set of variables as the study progresses over time due to new information from other investigations. Thus, it is common for feature d to be missing for the first t_d rounds of long-term longitudinal studies. For example, ECG results were only first recorded in the FHS study 14 years after the study began [91, 223].

To generate $\gamma 30\%$ MNAR patterns under this mechanism, we use the following process. First, we randomly select a covariate d and a discrete uniform random variable $t_d \in \{1, 2, \ldots, N\}$, where $N = 10$ for the FHS dataset and $N = 4$ for the PPMI dataset. The value t_d corresponds to the last round of the longitudinal study that covariate d is missing. For example, if $t_d = 2$ for the covariate Left Ventricular Hypertrophy (LVH), then the value for LVH will be missing for all observations in the two first clinical examinations. We continue this process until we have introduced $\gamma 30\%$ MNAR missing values. Afterwards, we introduce additional MCAR missing values to the remaining dataset in order to obtain the final dataset with 30% missing values.

MNAR Mechanism for Data from EHR

In EHR data, missing data patterns may be correlated with the severity of patient's condition. Consider the case of a patient whose physician suspects the existence of chronic kidney disease. The associated record is more likely to have a recorded value for Glomerular Filtration Rate since it is a direct indication of the kidney's functional status [208]. Therefore, observed values are more likely to be below the threshold of 60mL/min/1.73 m^2 since they correspond to sicker patients.

To generate $\gamma 30\%$ MNAR patterns under this mechanism, we suppose that missing indicators are independent Bernoulli random variables where the probability that entry x_{id} is missing equals the probability that a normal random variable $N(x_{id}, \epsilon)$ is greater than a particular threshold for covariate d. The threshold for each covariate d is the quantile of $\mathbf{X}^d$ which corresponds to the desired missing percentage level $\gamma 30\%$. Then, we introduce

additional MCAR missing values to the remaining dataset in order to obtain the final dataset with 30% missing values total for this experiment.

2.3.3 Experimental Setup

In this section, we describe the setup of computational experiments that compare `med.knn` to other state-of-the-art imputation methods. We use data from three distinct sources to test the performance of our algorithm on both longitudinal cohort study and EHR datasets. The codebase for the computational experiments is publicly available at `https://github.com/colin78/medimpute_computational_experiments`.

In our experiments, we take the full dataset to be the ground truth. First, we normalize the data so that each continuous covariate has mean zero and standard deviation equal to one. Then, we run some of the most commonly-used and state-of-the-art methods for imputation to predict the missing values and compare against `med.knn`. The methods that we compare are as follows:

1. **Mean** (`mean`): This is the simplest method. For each continuous feature, we impute the mean of the observed values and, for each categorical feature, we impute the mode of the observed values [218].

2. **Moving Average** (`moving.avg`): This method takes into account only observations of the same entity (i.e., patient) and imputes their averages under a given time window. In cases where only one observation per entity is available, the method reduces to the `mean`. For each dataset, we consider a different time horizon depending on the relative scale of the data (i.e, years, months, or days). Implemented in the Julia programming language.

3. **Bayesian Principal Component Analysis** (`bpca`): This method takes a singular value decomposition (SVD) of the data matrix and information from a Bayesian prior distribution on the model parameters to impute missing values [250]. Implemented using the `pcaMethods` package in the R programming language.

4. **Multivariate Imputation via Chained Equations** (`mice`): In this multiple imputation method, we begin from m random starts and iteratively update each one to produce m independent imputations. In each iteration, we update the imputed values in feature d by drawing from a distribution conditional on all other features [341]. We use Classification Trees for the categorical features and Regression Trees for the continuous features. Implemented using the `mice` package in in the R programming language.

5. **Multiple Imputation with Boostrap Expectation Maximization (`Amelia II`):** This is another multiple imputation method that builds upon the `Amelia I` framework, which assumes that the data is jointly distributed as multivariate normal and uses an expectation-maximization (EM) algorithm with bootstrapping [167, 191]. In addition, a newer version of the method allows for the imputation of cross-sectional time series data. It can build a general model of patterns within variables across time by creating a sequence of polynomials of the time index. Thus, it is able to capture variables that are recorded over time within a cross-sectional unit and are observed to vary smoothly over time. Implemented using the `amelia` package in the R programming language.

6. **OptImpute under K-NN Objective** (`opt.knn`): This method finds a high quality solution to Problem (2.5) minimizing the sum of distances from each point to its K-Nearest Neighbors [32]. We find solutions to this problem using Algorithm 1 with the CD update. Fixing $K = 10$, we use several warm and random restarts and select the imputation with the best objective value. Implemented using the `OptImpute` package in the Julia programming language.

7. **MedImpute under K-NN Objective** (`med.knn`): This method finds a high quality solution to Problem (2.8) minimizing the sum of distances from each point to its K-Nearest Neighbors and other instances of the same individual. We find solutions to this problem using Algorithm 2 with the CD update. For each feature d, we perform cross-validation to tune the parameters α_d, h_d with the rest of the MedImpute parameters set equal to zero. Fixing $K = 10$, we use several warm and random restarts and select

the imputation with the best objective value. The `med.knn` algorithm is implemented in Julia and is available to academic researchers under a free academic license.*

For each experiment, we evaluate the imputation accuracy of each method using the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics, which are extended to accommodate both continuous and categorical covariates. Let $\mathcal{M}_0^{test}$, $\mathcal{M}_1^{test}$ be the hold-out sets for the missing continuous and categorical covariates, respectively. We define the MAE and RMSE metrics to be:

$$\text{MAE} := \frac{1}{|\mathcal{M}_0^{test}|} \sum_{(i,d)\in\mathcal{M}_0^{test}} |w_{id} - x_{id}| + \frac{1}{|\mathcal{M}_1^{test}|} \sum_{(i,d)\in\mathcal{M}_1^{test}} \mathbb{1}_{\{v_{id}\neq x_{id}\}}, \tag{2.30}$$

$$\text{RMSE} := \sqrt{\frac{1}{|\mathcal{M}_0^{test}|} \sum_{(i,d)\in\mathcal{M}_0^{test}} (w_{id} - x_{id})^2 + \frac{1}{|\mathcal{M}_1^{test}|} \sum_{(i,d)\in\mathcal{M}_1^{test}} \mathbb{1}_{\{v_{id}\neq x_{id}\}}}. \tag{2.31}$$

In addition to comparing the accuracy of each method on the imputation task, we also compare their performance on downstream predictive tasks which are tailored for each dataset. In these experiments, we use the imputation methods to fill in the missing values of the datasets, and then we train machine learning models with the data from completed datasets. By comparing the accuracy of the predictive models on the downstream tasks, we can see the relative impact of using one imputation method versus another in a machine learning pipeline. For the FHS dataset, the downstream task is to predict 10-year risk of stroke, a classification task. For the DFCI dataset, the downstream task is to predict 60-day risk of mortality, which is also a classification task. For the PPMI dataset, the downstream task is to predict the Montreal Cognitive Assessment (MoCA) score for next year, which is a regression task.

To evaluate the accuracy on the downstream predictive task, first we split the patients from the completed dataset into a training and testing set using a 75%/25% ratio. For the longitudinal datasets (FHS and PPMI) we include only one visit per patient, the most recent

*The codebase for the computational experiments is publicly available at `https://github.com/colin78/medimpute_computational_experiments`.

one. Thus, the time series component of the dataset is only present in the missing data imputation process but not in the supervised learning part of the experiment. This setup allows us to quantify the relative benefit of `med.knn` per individual. For the EHR dataset (DFCI), we include all of the observations from each patient in either the training or testing set for the supervised learning task.

Next, we train predictive models on the training set and report the out-of-sample accuracy on the testing set. For the classification tasks, we train ℓ_1-regularized logistic regression models and report the out-of-sample Area Under the ROC Curve (AUC). For the regression task, we train ℓ_1-regularized linear regression models and report the out-of-sample MAE. These two metrics are commonly used evaluation criteria in machine learning [161]. We repeat all experiments for 25 random seeds and average the results. Each iteration corresponds to a different random split of the patients into the training and testing sets, a random warmstart, and a randomly generated missing data pattern. In particular, we note that the patient IDs and the time stamps corresponding to each row of the dataset are maintained across the different random seeds, so that the temporal sequence of the records remains the same as the original dataset.

We artificially created missing data under different mechanisms and random patterns to compare the imputation accuracy of the proposed method. The missing data generation process was independently applied to each column. For a fixed missing percentage $f\%$, we remove the necessary number of known values for each feature to reach the $f\%$ target. The patient ID y_i was not factored in the missing data generation process and all rows were considered independent observations. If the existing percent of missing data for a column was higher than the target $f\%$, we do not generate any artificial missing values for the covariate, and thus the feature does not contribute to the estimation of the imputation accuracy metrics.

Given this framework for evaluating imputation methods on both imputation and downstream tasks, we conduct a variety of experiments which vary the pattern of the missing data. In particular, we conduct three different types of experiments that correspond to variations in the form of missing data that we frequently encounter in medical datasets:

1. **Percentage of Missing Data:** We generate patterns of missing data for various percentages ranging from 10% to 50% under the MCAR mechanism. Given a target proportion of missing data f (i.e., $f = 20\%$), we generate among all observed data f missing values at each column independently from the rest completely at random.

2. **Number of Observations Per Patient:** With the missing percentage fixed at 50% MCAR, we vary the time frame during which patient observations are included in the imputation task. Our goal is to quantify the effect of the time series component as we vary its intensity.

3. **Mechanism of Missing Data:** With the missing percentage fixed at 30%, we vary the missing data mechanism from MCAR to MNAR on a gradient scale. In particular, we suppose that the missing pattern is (γ30% MNAR, $(1-\gamma)$30% MCAR), where γ varies from 0 to 1. We consider two different MNAR mechanisms that correspond to distinct missing data patterns observed in longitudinal studies and EHR.

The objective of the first set of experiments is to determine which imputation methods perform best at high and low levels of missing data. For these experiments, we also report the results from statistical hypothesis tests (Friedman Rank and pairwise t-tests) to evaluate whether the rankings and differences between the imputation algorithms are statistically significant. The objective of the second set of experiments is to determine how the performance of `med.knn` and other imputation methods varies as the amount of time series information available on each patient fluctuates. Finally, the objective of the third set of experiments is to determine how robust each imputation method is with respect to the missing data mechanism. In the previous section, we describe the two mechanisms for generating MNAR data for the third set of experiments. Below, we summarize all of the steps required to run one of the computational experiments for a single random seed:

1. Fix a random seed s, a dataset, a desired missingness percentage level f%, a missing data imputation method, and a value for the γ parameter.

2. Generate a random missing data pattern in the given dataset using the targeted percentage of missing values $f\%$, the random seed s, and the value of the γ parameter.

3. Impute the missing values in the provided dataset using the specified algorithm (i.e. `med.knn`, `mean`, `bpca`).

4. Calculate the imputation error using the MAE and RMSE metrics (see Equations 2.30-2.31) on the artificially generated missing data.

5. Split the patients in the dataset into a training and testing set using a 75%/25% ratio. For the longitudinal datasets, only include the most recent observation from each individual in the training and testing sets. For the EHR (DFCI) dataset, include all of the observations from each individual in the training or testing set.

6. Train a downstream predictive model on the training set using the `cv.glmnet` function from the R `glmnet` package [132]. For the FHS and DFCI datasets which have binary outcomes variables, train a logistic regression model with l_1 regularization. For the PPMI dataset which has a continuous outcome variable, train a linear regression model with l_1 regularization.

7. Report the out-of-sample performance of the trained model on the testing set. For the classification tasks, report the out-of-sample AUC, and for the regression task, report the out-of-sample MAE.

2.3.4 Imputation Results

In this section, we provide the results from all experiments on the imputation tasks. In particular, we present the imputation results from the 1) Percentage of Missing Data, 2) Number of OPP, and 3) Mechanism of Missing Data experiments.

Percentage of Missing Data In Figure 2.1, we show the MAE imputation accuracy results from the first set of experiments in which we vary the percentage of missing data

from 10% to 50%, and the missing data mechanism is fixed to MCAR. Across all of the datasets, `med.knn` achieves the lowest average MAE for all of the missing percentages tested. On the FHS longitudinal dataset with 50% MCAR data, `med.knn` has an average MAE of 0.289compared to the next best method `opt.knn` with an average MAE of 0.503, a 42.54% reduction. Similarly, on the PPMI longitudinal dataset with 50% MCAR data, `med.knn` has an average MAE of 1.286 compared to the next best method `opt.knn` with an average MAE of 1.99, a 35.37% reduction. On the DFCI dataset with 50% MCAR data, `med.knn` has an average MAE of 3.568 compared to the next best method `mean` with an average MAE of 4.367, a 22.39% reduction.

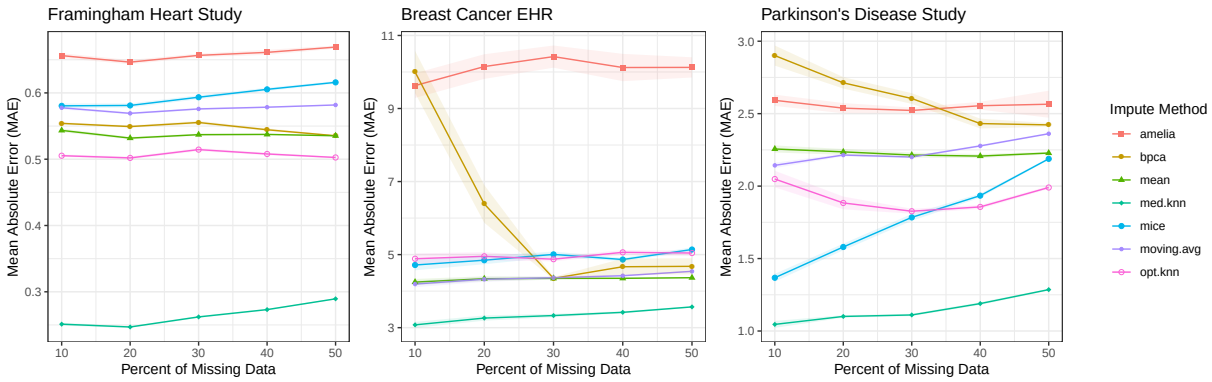


Figure 2.1: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50%. The missing data mechanism is fixed to MCAR.

In Figure 2.2, we present the RMSE imputation accuracy results. In general, the results are similar to the MAE imputation accuracy results, and `med.knn` produces the imputation with the lowest RMSE across all experiments. One notable difference is on the DFCI dataset, the relative improvement of `med.knn` compared to `bpca`, `moving.avg`, and `mean` is much smaller. Because the `mean` imputation method performs relatively well, this suggests that there are some difficult-to-impute covariates in the DFCI dataset which are resulting in large RMSE values for all of the more complex methods.

χ^2 statistic (adjusted p-value)			
%	**FHS**	**DFCI**	**PPMI**
10	130 (<0.001***)	210 (<0.001***)	75 (<0.001***)
20	130 (<0.001***)	220 (<0.001***)	53 (<0.001***)
30	130 (<0.001***)	260 (<0.001***)	74 (<0.001***)
40	110 (<0.001***)	230 (<0.001***)	58 (<0.001***)
50	140 (<0.001***)	270 (<0.001***)	71 (<0.001***)

(a) MAE

χ^2 statistic (adjusted p-value)			
%	**FHS**	**DFCI**	**PPMI**
10	130 (<0.001***)	210 (<0.001***)	75 (<0.001***)
20	130 (<0.001***)	220 (<0.001***)	53 (<0.001***)
30	130 (<0.001***)	260 (<0.001***)	74 (<0.001***)
40	110 (<0.001***)	230 (<0.001***)	58 (<0.001***)
50	140 (<0.001***)	270 (<0.001***)	71 (<0.001***)

(b) RMSE

Table 2.1: The Friedman Rank test results for the imputation tasks varying the percentage of missing data from 10-50% MCAR, using either the MAE or RMSE metric for comparison. Each table shows the value of Friedman's Chi-squared statistic and p-value for the hypothesis test comparing `med.knn` against the benchmark methods for each experiment.

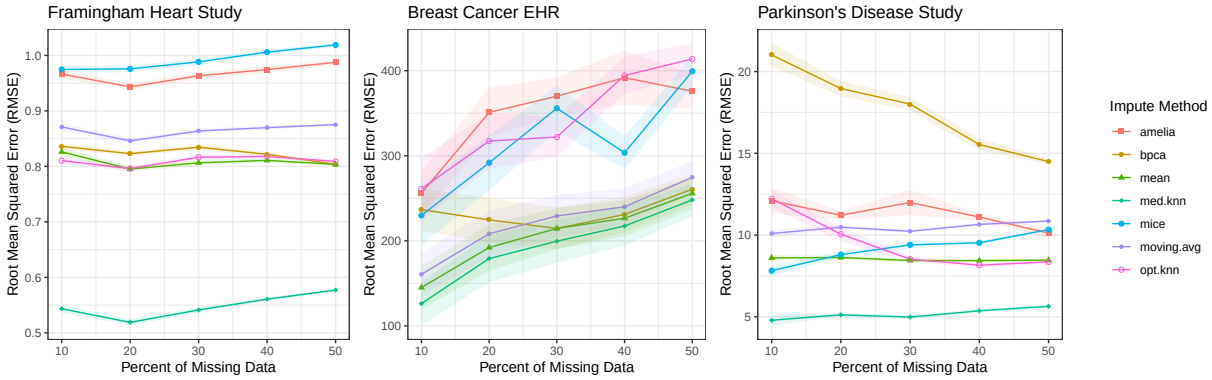


Figure 2.2: Imputation errors for each method using the RMSE metric on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50%. The missing data mechanism is fixed to MCAR.

In Table 2.1, we present the results from the Friedman Rank test for each of the Missing Data imputation experiments. In this statistical test, we compare the relative rank of `med.knn` against the relative ranks of the comparator methods for each of the 25 random seeds. These results demonstrate that the `med.knn` method is consistently ranked higher than the others across each of the experiments.

In Table 2.2, we present the results from the pairwise t-test for each of the experiments. In this statistical test, we evaluate the differences in MAE between `med.knn` and each of the comparison methods. In all of the experiments, we observe that the differences in MAE are statistically significant with p-values less than 0.001. In most cases, we observe that the

FHS

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`moving.avg`	`amelia`	`bpca`	`mean`	`opt.knn`
10	-0.33 (<0.001***)	-0.33 (<0.001***)	-0.41 (<0.001***)	-0.30 (<0.001***)	-0.29 (<0.001***)	-0.25 (<0.001***)
20	-0.33 (<0.001***)	-0.32 (<0.001***)	-0.40 (<0.001***)	-0.30 (<0.001***)	-0.28 (<0.001***)	-0.26 (<0.001***)
30	-0.33 (<0.001***)	-0.31 (<0.001***)	-0.39 (<0.001***)	-0.29 (<0.001***)	-0.27 (<0.001***)	-0.25 (<0.001***)
40	-0.33 (<0.001***)	-0.31 (<0.001***)	-0.39 (<0.001***)	-0.27 (<0.001***)	-0.26 (<0.001***)	-0.23 (<0.001***)
50	-0.33 (<0.001***)	-0.29 (<0.001***)	-0.38 (<0.001***)	-0.25 (<0.001***)	-0.25 (<0.001***)	-0.21 (<0.001***)

DFCI

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	-1.64 (<0.001***)	-6.55 (<0.001***)	-1.92 (<0.001***)	-6.92 (<0.001***)	-1.17 (<0.001***)	-1.81 (<0.001***)
20	-1.58 (<0.001***)	-6.89 (<0.001***)	-1.86 (<0.001***)	-3.12 (<0.001***)	-1.08 (<0.001***)	-1.69 (<0.001***)
30	-1.67 (<0.001***)	-7.09 (<0.001***)	-1.84 (<0.001***)	-1.02 (<0.001***)	-1.02 (<0.001***)	-1.54 (<0.001***)
40	-1.46 (<0.001***)	-6.71 (<0.001***)	-1.81 (<0.001***)	-1.26 (<0.001***)	-0.93 (<0.001***)	-1.62 (<0.001***)
50	-1.57 (<0.001***)	-6.56 (<0.001***)	-1.77 (<0.001***)	-1.11 (<0.001***)	-0.80 (<0.001***)	-1.48 (<0.001***)

PPMI

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	-0.32 (<0.001***)	-1.55 (<0.001***)	-1.10 (<0.001***)	-1.86 (<0.001***)	-1.21 (<0.001***)	-1.00 (<0.001***)
20	-0.48 (<0.001***)	-1.44 (<0.001***)	-1.10 (<0.001***)	-1.61 (<0.001***)	-1.14 (<0.001***)	-0.78 (<0.001***)
30	-0.67 (<0.001***)	-1.36 (<0.001***)	-1.09 (<0.001***)	-1.49 (<0.001***)	-1.10 (<0.001***)	-0.72 (<0.001***)
40	-0.75 (<0.001***)	-1.37 (<0.001***)	-1.08 (<0.001***)	-1.24 (<0.001***)	-1.02 (<0.001***)	-0.67 (<0.001***)
50	-0.90 (<0.001***)	-1.40 (<0.001***)	-1.07 (<0.001***)	-1.14 (<0.001***)	-0.94 (<0.001***)	-0.70 (<0.001***)

Table 2.2: Pairwise t-tests between `med.knn` and benchmark methods for imputation tasks varying the percentage of missing data from 10-50% MCAR, using the MAE metric for comparison. The p-values are adjusted for multiple comparisons.

relative improvement of `med.knn` decreases as the percentage of missing data increases. This is because the comparator methods perform similarly across all levels of missing data from 10-50%, while the `med.knn` performs best at the lowest missing percentages. One exception is `mice` on the PPMI dataset, which declines in performance rapidly as the percentage of missing data increases. Another exception is the `bpca` method, which surprisingly improves in performance as the percentage of missing data increases for the DFCI and PPMI datasets. One explanation for these results could be that `bpca` is overfitting on the datasets which have few missing values.

Number of Observations Per Patient In Figure 2.3, we present the MAE imputation accuracy results from the experiments in which we vary the number of OPP. Across all of the experiments, we observe that as the time horizon increases, the performance of `med.knn` generally improves. This is expected, because as the time horizon increases, we include more

OPP in the dataset, so there is more time series information that can be leveraged during the imputation process.

Similarly, the imputation accuracy of the `moving.avg` method generally improves as the time horizon increases. One notable exception is in the FHS dataset, the MAE of the `moving.avg` method increases as the time horizon increases from 10 to 20 years, while the MAE of `med.knn` remains relatively constant. From this, we can deduce that past observations of patients in the FHS dataset from 10 to 20 years prior have little predictive power for the other imputed values, which causes simple time series methods such as `moving.avg` to perform worse with more data. In contrast, the `med.knn` method has an exponential halflife parameter that we can tune so that observations from 10+ years ago are weighted less heavily in the imputation, so the performance remains about the same with the additional data.

One surprising trend that we observe in these graphs is the performance of `amelia`, which is another imputation method that takes into account time series information. On the DFCI dataset, as the time horizon increases, the imputation error increases. In addition, on the FHS dataset, as time horizon increases, the imputation error remains about the same. Only in the PPMI dataset does the performance of `amelia` noticeably improve as the time horizon increases.

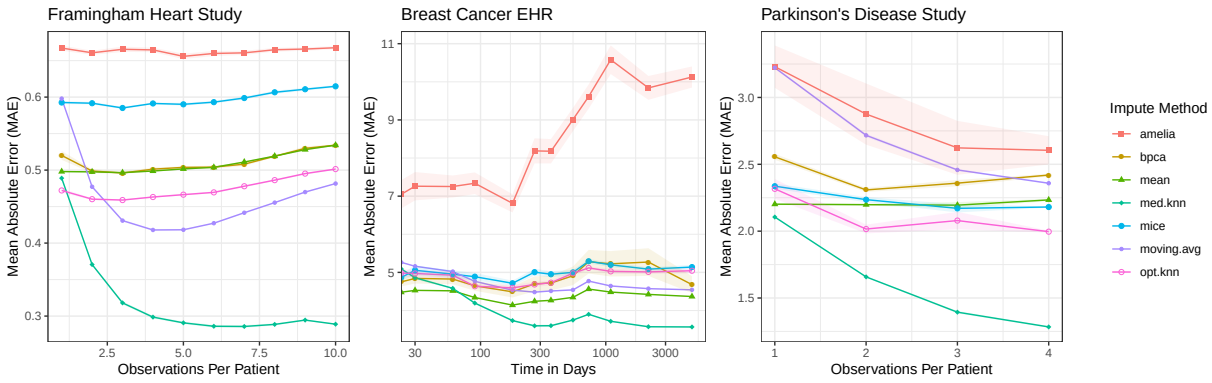


Figure 2.3: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of OPP. The missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%.

In Figure 2.4, we present the RMSE imputation accuracy results for the OPP experiments.

The results are similar to the MAE imputation accuracy results, and `med.knn` produces the imputation with the lowest RMSE across all experiments. One characteristic of the RMSE results is that they are much noisier, and in particular on the DFCI dataset the RMSE values do not decrease monotonically in a smooth fashion. Since the RMSE metric is more sensitive to outliers than the MAE metric, this suggests that there may be some outliers in the DFCI data which are added into the dataset at different time horizons.

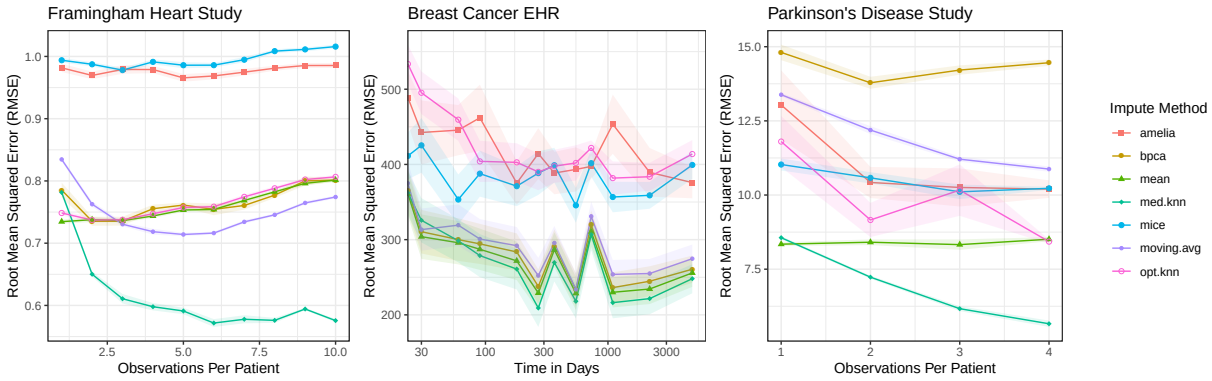


Figure 2.4: Imputation errors for each method using the RMSE metric on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of OPP. The missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%.

In addition to evaluating the imputation accuracy of `med.knn` on datasets with varying numbers of OPP, we can also evaluate the imputation accuracy on subsets of patients within the DFCI dataset which have varying numbers of observations. In Figure 2.5, we present the imputation errors for `med.knn` on the DFCI dataset with 30% MCAR missing data, for subgroups of patients which have $1, 2, \ldots, 12$ OPP in the dataset. Overall, the MAE for the entire dataset is 3.331. For patients with one visit, and therefore one observation in the dataset, the average MAE is almost 3.5. In contrast, for patients with 10 or more visits, the average MAE is below 2.5. This suggests that in datasets with heterogeneous numbers of OPP, the `med.knn` imputation may be most accurate for the patients with the most observations in the dataset.

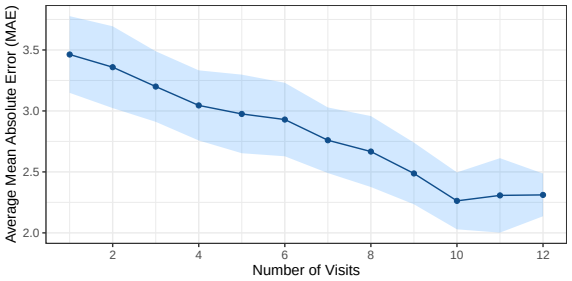


Figure 2.5: Imputation errors for `med.knn` on the DFCI dataset with 30% MCAR missing data for subgroups of patients which have varying numbers of visits in the dataset.

Overall, from the OPP experiments, we can conclude that `med.knn` method performs best with the additional time series information. As the time horizon increases, the imputation accuracy of `med.knn` generally improves or remains the same, while in a few cases the other time series methods `moving.avg` and `amelia` perform significantly worse with additional time series data. In addition, the imputation accuracy of the methods which do not take into account time series information (`bpca`, `mean`, `mice`, `opt.knn`) remains relatively constant as the time horizon varies. Furthermore, within a dataset that has heterogeneous numbers of OPP, such as EHR datasets, we may expect `med.knn` to most accurately impute values for the patients with the most observations in the dataset.

Mechanism of Missing Data In Figure 2.6, we present the MAE imputation accuracy results from the experiments in which we vary the mechanism of missing data. Across all of these experiments, we observe that `med.knn` has the best average MAE values by a significant margin.

In general, the imputation accuracy of all of the imputation methods increases or remains the same as the proportion of MNAR data increases. Two exceptions are the `moving.avg` method on the FHS dataset and the `amelia` method on the DFCI experiments, which both improve in performance at first as a small proportion of MNAR data is added. One possible explanation for this is that the MNAR data acts as a regularizer which helps these methods avoids overfitting to the dataset. However, in most cases the imputation error increases or remains constant as the percentage of MNAR data increases.

In the FHS MNAR experiments, the performance of all of the methods remains relatively constant, however the imputation error of `moving.avg` improves at $\gamma = 0.1$. Because `moving.avg` is the second-best performing method in these experiments, this means that the edge of the `med.knn` method slightly decreases in these experiments. In the PPMI MNAR experiments, the imputation error of all methods increases approximately linearly as the proportion of MNAR data increases. In the DFCI MNAR experiments, the imputation error for all methods except for `amelia` increases sharply at $\gamma = 0.1$, and then increases linearly afterwards as γ increases. As a result, for the experiments on the DFCI and PPMI datasets, the absolute improvement of `med.knn` over the comparator methods remains about the same as the proportion of MNAR data increases.

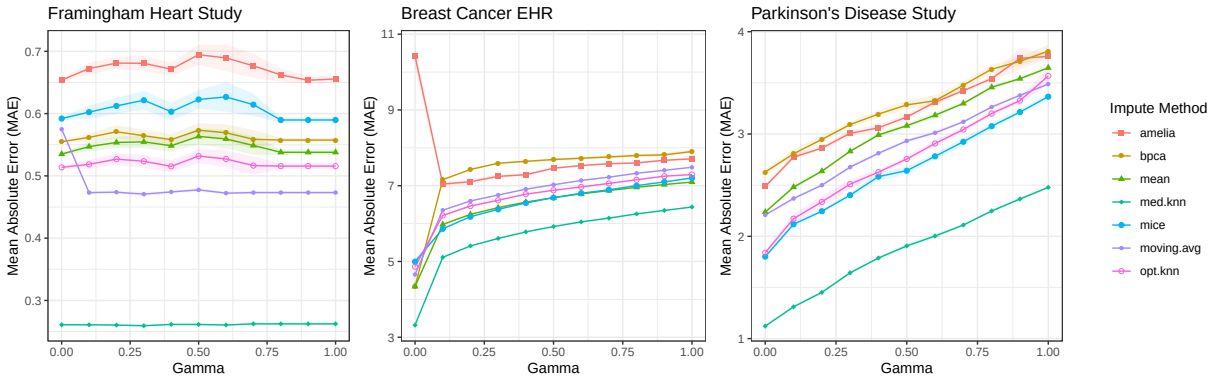


Figure 2.6: Imputation errors for each method using the MAE metric on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). The total percentage of missing data is fixed to 30%.

In Figure 2.7, we present the RMSE imputation accuracy results for the missing data mechanism experiments. The results are largely consistent with the MAE imputation accuracy results. In particular, `med.knn` produces the imputation with the lowest RMSE by a significant margin across all experiments.

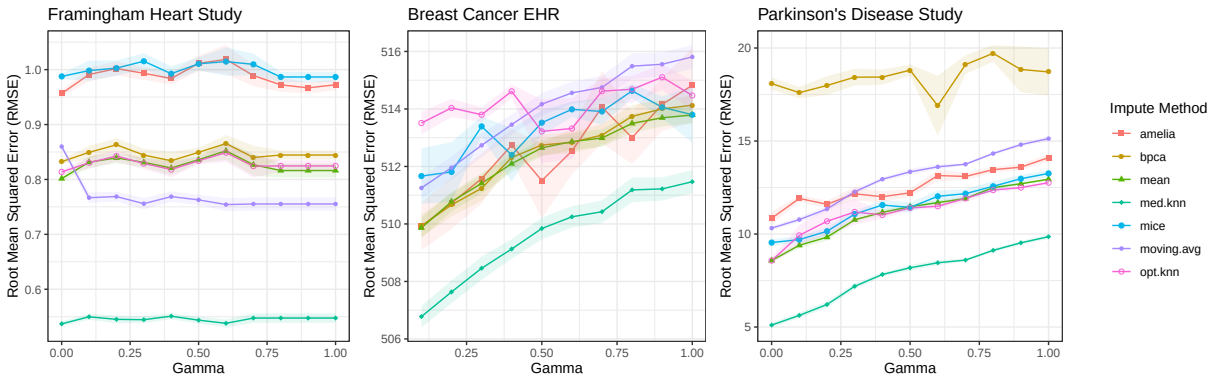


Figure 2.7: Imputation errors for each method using the RMSE metric on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). The total percentage of missing data is fixed to 30%.

Overall, these experiments demonstrate that the `med.knn` method performs well relative to the other imputation methods even as the mechanism of missing data changes. In the MNAR experiments for the longitudinal datasets, FHS and PPMI, the relative imputation accuracy of the comparator methods remains approximately the same with the `med.knn` method performing best, with the exception of the `moving.avg` method which performs significantly worse. Thus, we can conclude that the `med.knn` method is well suited for imputing missing values according to the particular MNAR mechanism designed for longitudinal datasets which is described in Section 2.3.2. In the MNAR experiments for the EHR dataset DFCI, the relative imputation accuracy of the comparator methods remains approximately the same with the `med.knn` method performing best, with the exception of the `amelia` method which performs significantly better. Therefore, we can also conclude that the `med.knn` is suitable for imputing missing values according to the MNAR mechanism for EHR datasets as described in Section 2.3.2.

2.3.5 Prediction Results

In this section, we provide the results from all experiments on the downstream prediction tasks. In particular, we present the downstream prediction results from the 1) Percentage of

Missing Data, 2) Number of OPP, and 3) Mechanism of Missing Data experiments. For the FHS and DFCI datasets, in which we train and evaluate classification models, we report the average out-of-sample AUC results. For the PPMI dataset, in which we train and evaluate regression models, we report the average out-of-sample MAE results.

Percentage of Missing Data In Figure 2.8, we present the performance on the downstream tasks from the experiments in which we vary the percentage of missing data. Across all of the datasets, the `med.knn` method performs best, and the downstream performance of all methods generally declines as the missing level increases. In particular, the AUC values generally decrease for the classification tasks and the MAE values generally increase for the regression tasks as the percentage of missing data increases.

For the FHS dataset, while the downstream performance of all methods declines as the percentage of missing data increases, the downstream performance of `med.knn` declines least rapidly. In particular, with 20% missing data, the downstream AUC of `med.knn` is 0.897, compared to downstream AUC of 0.861 from the second-best method `bpca` and the baseline AUC of 0.901 with no additional missing data. With 50% missing data, the downstream AUC of `med.knn` is 0.864, compared to 0.826 for the second-best method `moving.avg`.

Similarly, for the DFCI dataset, the `med.knn` method performs best across all levels of missing data, and the downstream AUC values generally decrease as the missing level increases. The only exception is for the `amelia` method, where we do not observe a smooth trend because this method does not converge in some cases. In addition, the relative improvement of `med.knn` compared to the other imputation methods is lower for this dataset. At 50% missing data, the downstream AUC of `med.knn` is 0.889, compared to 0.884 for the second-best method `bpca` and the baseline AUC of 0.92 with no additional missing data.

Lastly, in the PPMI dataset, we observe the same trends that the `med.knn` method performs best, and the performance of all methods declines as the missing level increases. In this case, the downstream MAE for each method increases as the percentage of missing data increases. Across all levels of missing data, `med.knn` achieves the lowest downstream MAE. At 50% missing data, the downstream MAE of `med.knn` is 1.917, compared to 2.092 for the

	χ^2 statistic (adjusted p-value)		
%	**FHS**	**DFCI**	**PPMI**
10	130 (<0.001***)	210 (<0.001***)	75 (<0.001***)
20	130 (<0.001***)	220 (<0.001***)	53 (<0.001***)
30	130 (<0.001***)	260 (<0.001***)	74 (<0.001***)
40	110 (<0.001***)	230 (<0.001***)	58 (<0.001***)
50	140 (<0.001***)	270 (<0.001***)	71 (<0.001***)

Table 2.3: The Friedman Rank test results for the downstream predictive tasks varying the percentage of missing data from 10-50% MCAR. The table shows the value of Friedman's Chi-squared statistic and p-value for the hypothesis test comparing `med.knn` against the benchmark methods for each experiment. The p-values are adjusted for multiple comparisons.

second-best method `opt.knn` and the baseline MAE of 1.170 with no additional missing data.

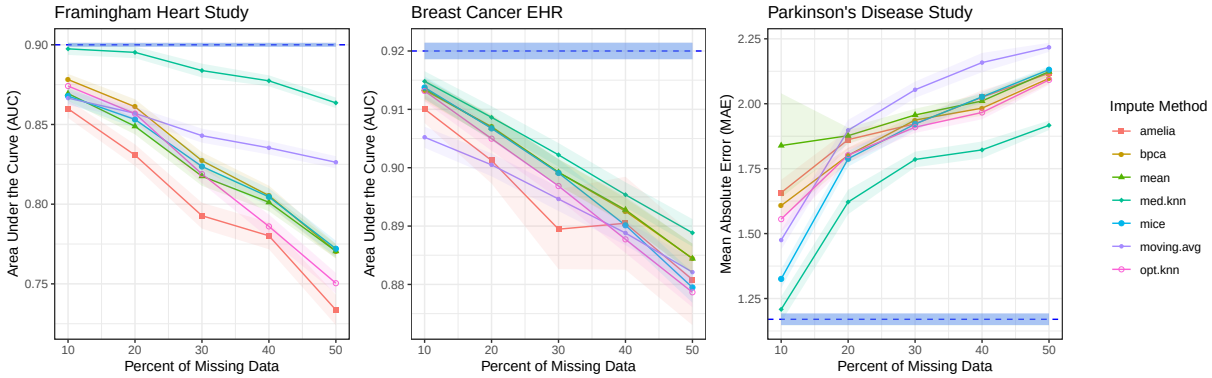


Figure 2.8: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the percentage of missing data from 10% to 50% according to the MCAR mechanism. On each plot, we overlay the downstream accuracy of a baseline model trained with no additional missing data as a dotted blue line (shaded with standard error bars).

In Table 2.3, we present the results from the Friedman Rank tests for each of the downstream predictive tasks varying the percentage of missing data. Similar to Friedman Rank tests for the imputation tasks, each test is significant with a p-value less than 0.001. These results demonstrate that the `med.knn` method is consistently ranked higher than the others for each of the downstream predictive tasks.

In Table 2.4, we present the results from the pairwise t-tests for each of the experiments. In this statistical test, we evaluate the differences in downstream predictive performance

between `med.knn` and each of the comparison methods. We consider the differences in downstream AUC for the classification tasks, and we consider the differences in downstream MAE for the regression tasks. In most of the experiments, we observe that the differences in downstream AUC/MAE are statistically significant with p-values less than 0.001. These results demonstrate that the relative improvement in imputation accuracy for the `med.knn` method carries over to a relative improvement in performance on the downstream predictive tasks with different levels of MCAR data. Between the two classification tasks, we observe that the `med.knn` gives larger improvements in AUC on the FHS dataset than the DFCI dataset. In addition, we observe that as the percentage of missing data increases, the relative improvement of `med.knn` increases in general. These results are expected because as the percentage of missing data increases, the impact of the imputation method on the training data and the final prediction task increases as well. Since `med.knn` provides substantial improvements in imputation accuracy for all levels of missing data, having larger amounts of missing data generally leads to larger gains in downstream predictive accuracy. There are a few exceptions to this, for example `amelia`, `bpca`, `mean`, and `opt.knn` on the PPMI dataset, and `moving.avg` on the DFCI dataset. In these cases, the largest improvement for `med.knn` occurs at the 10% missing level. For these several examples, it follows that `med.knn` does a much better job at simulating the training dataset with 10% missing data, but the other methods begin to catch up as the percentage of missing data increases.

Number of Observations Per Patient In Figure 2.9, we present the performance on the downstream tasks from the experiments in which we vary the time horizon which determines the number of OPP. Across all of the experiments, we observe that the downstream performance of `med.knn` tends to improve as the time horizon increases, so that the dataset includes more OPP. However, for each dataset, after a certain point there are diminishing returns, so that adding more OPP to the dataset does not improve the performance on the downstream task.

For the FHS dataset, in which the task is to predict 10-year risk of stroke, the downstream AUC of `med.knn` plateau starts to plateau at a time horizon of 6 years. For the DFCI dataset,

FHS: Predicting 10-year Risk of Stroke

	Δ AUC (adjusted p-value)					
Missing %	`mice`	`moving.avg`	`amelia`	`bpca`	`mean`	`opt.knn`
10	0.0296 (<0.001***)	0.0309 (<0.001***)	0.0378 (<0.001***)	0.0193 (<0.001***)	0.0280 (<0.001***)	0.0233 (<0.001***)
20	0.0421 (<0.001***)	0.0382 (<0.001***)	0.0645 (<0.001***)	0.0341 (<0.001***)	0.0464 (<0.001***)	0.0384 (<0.001***)
30	0.0602 (<0.001***)	0.0408 (<0.001***)	0.0908 (<0.001***)	0.0566 (<0.001***)	0.0663 (<0.001***)	0.0649 (<0.001***)
40	0.0728 (<0.001***)	0.0420 (<0.001***)	0.0997 (<0.001***)	0.0720 (<0.001***)	0.0762 (<0.001***)	0.0913 (<0.001***)
50	0.0915 (<0.001***)	0.0373 (<0.001***)	0.1266 (<0.001***)	0.0931 (<0.001***)	0.0931 (<0.001***)	0.1132 (<0.001***)

DFCI: Predicting 60-day Risk of Mortality

	Δ AUC (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	0.0010 (0.234)	0.0050 (<0.001***)	0.0196 (<0.001***)	0.0015 (0.261)	0.0013 (0.407)	0.0016 (<0.001***)
20	0.0019 (0.004**)	0.0060 (0.092)	0.0181 (<0.001***)	0.0016 (0.318)	0.0018 (0.234)	0.0037 (<0.001***)
30	0.0031 (0.003**)	0.0114 (0.052)	0.0176 (<0.001***)	0.0030 (0.037*)	0.0030 (0.037*)	0.0053 (<0.001***)
40	0.0056 (<0.001***)	0.0046 (0.075)	0.0169 (<0.001***)	0.0033 (0.032*)	0.0030 (0.044*)	0.0081 (<0.001***)
50	0.0094 (<0.001***)	0.0077 (0.065)	0.0167 (<0.001***)	0.0044 (0.003**)	0.0044 (0.003**)	0.0102 (<0.001***)

PPMI: Predicting the MoCA score

	Δ MAE (adjusted p-value)					
Missing %	`mice`	`amelia`	`moving.avg`	`bpca`	`mean`	`opt.knn`
10	-0.117 (0.027*)	-0.435 (<0.001***)	-0.288 (<0.001***)	-0.399 (<0.001***)	-0.631 (0.027*)	-0.347 (<0.001***)
20	-0.167 (0.004**)	-0.249 (0.002**)	-0.329 (<0.001***)	-0.180 (<0.001***)	-0.255 (<0.001***)	-0.181 (0.004**)
30	-0.137 (<0.001***)	-0.167 (<0.001***)	-0.296 (<0.001***)	-0.152 (<0.001***)	-0.171 (<0.001***)	-0.124 (<0.001***)
40	-0.204 (<0.001***)	-0.153 (<0.001***)	-0.362 (<0.001***)	-0.161 (<0.001***)	-0.188 (<0.001***)	-0.144 (0.002**)
50	-0.214 (<0.001***)	-0.207 (<0.001***)	-0.312 (<0.001***)	-0.181 (<0.001***)	-0.207 (<0.001***)	-0.175 (<0.001***)

Table 2.4: Pairwise t-tests between `med.knn` and benchmark methods for imputation tasks varying the percentage of missing data from 10-50% MCAR. The p-values are adjusted for multiple comparisons.

in which the task is to predict 60-day risk of mortality, the downstream AUC of `med.knn` starts to plateau around 3 years. Similarly, for the PPMI dataset, in which the task is to predict the next year MoCA score, the downstream MAE reaches a minimum value at 3 years.

In comparison to the other methods, we observe that `med.knn` tends to perform relatively better with more OPP in the dataset. This indicates that the `med.knn` method is able to leverage the additional time series information more efficiently than the other methods. The only exception to this is `amelia` on the DFCI dataset, which outperforms `med.knn` with time horizons of 3 and 5 years, respectively. However, we observe that the `amelia` method is more unstable, and `med.knn` outperforms this method for the longest time horizon of 10 years.

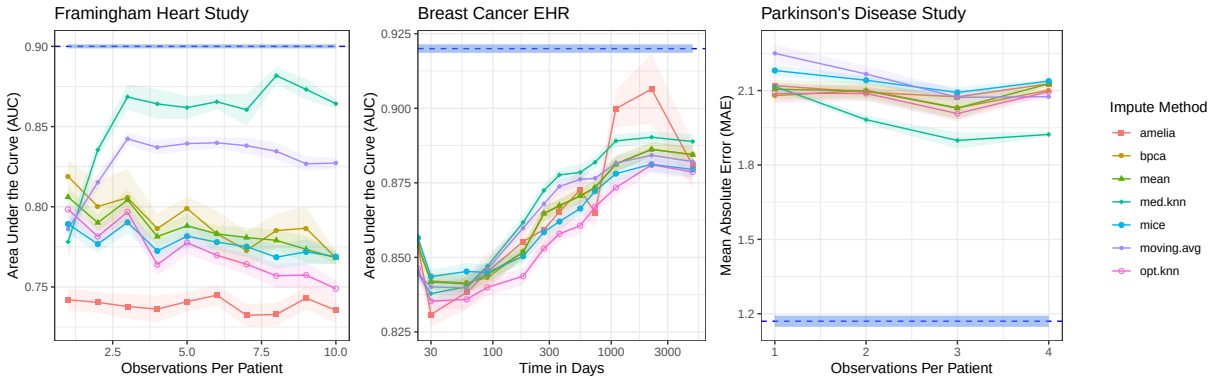


Figure 2.9: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the time horizon which determines the number of OPP. In these experiments, the missing data mechanism is fixed to MCAR, and the total percentage of missing data is fixed to 50%. On each plot, we overlay the downstream accuracy of a baseline model trained with no additional missing data as a dotted blue line (shaded with standard error bars).

Mechanism of Missing Data In Figure 2.10, we present the performance on the downstream tasks from the experiments in which we vary the mechanism of missing data. In all of the experiments, we observe that the `med.knn` acheives the best downstream accuracy, typically by a substantial margin.

In the FHS dataset, the average AUC for `med.knn` remains around 0.89 and above across all proportions of MNAR data, while the second-best performing method `moving.avg` has an average AUC below 0.87. In the PPMI dataset, the downstream MAE values for all of

the methods increases approximately linearly as the ratio of MNAR data increases. As a result, the relative improvement of `med.knn` on downstream tasks remains large for all of the MNAR experiments on longitudinal datasets.

On the other hand, the relative improvement of `med.knn` on downstream tasks is more varied for the MNAR experiments on EHR data. In the DFCI dataset, the downstream AUC values for each of the methods increases significantly when gamma = 0.1, and then decreases gradually as gamma increases further. These results are somewhat counterintuitive because the imputation errors for most of these methods increase significantly at gamma = 0.1, and then increase gradually afterwards. One possible explanation is that the DFCI dataset has some outlier values that tend to be missing under the MNAR mechanism for EHR data (described in Section 2.3.2), which typically skew the downstream prediction results. At the peak when gamma = 0.1, the relative improvement of `med.knn` is very small, with a downstream AUC of 0.916 compared to the next best method `mice` which has a downstream AUC of 0.915. At the extreme when gamma = 1, the downstream AUC of `med.knn` is 0.912 compared to 0.904 for the next best methods (`mice` and `bpca`).

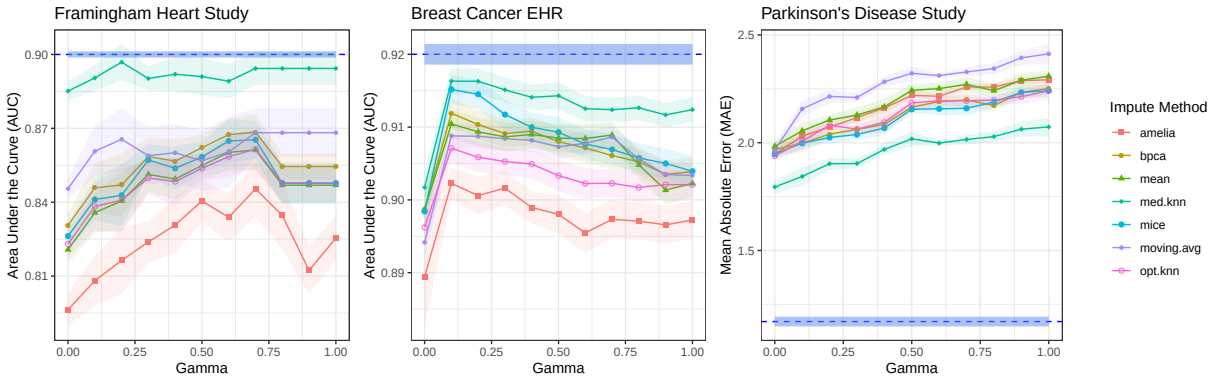


Figure 2.10: Downstream accuracy results for each method on the FHS, DFCI, and PPMI datasets, varying the ratio of the missing data mechanism from $\gamma = 0$ (30% MCAR, 0% MNAR) to $\gamma = 1$ (0% MCAR, 30% MNAR). On each plot, we overlay the downstream accuracy of a baseline model trained with no additional missing data as a dotted blue line (shaded with standard error bars).

2.3.6 Discussion of the Computational Experiments on Real-World Clinical Datasets

In this section, we discuss the major takeaways from the computational experiments on real-world clinical datasets. For each dataset, we consider downstream models to predict patient outcomes that are clinically relevant, in order to simulate the performance of `med.knn` in practical applications. For the FHS and PPMI datasets, which are longitudinal studies, the clinical outcomes of interest are 10-year risk of stroke and next year MoCA score, which can be predicted using the most recent observation for each patient. For the DFCI dataset, which is an EHR dataset, the clinical outcome of interest is 60-day risk of mortality for late-stage cancer patients, which requires us to train models using all of the observations from each patient (using the latest observation for each patient would bias the results). As a result, the evaluation of the downstream models is different between the datasets. Furthermore, we conduct non-identical experiments on each dataset due to inherent dissimilarities in the time series structure.

Due to the significant differences between each dataset, we can draw separate conclusions from each one as a separate case study. The FHS dataset is a long term longitudinal study with many patients, few covariates, and a downstream classification task. In contrast, the PPMI dataset is a shorter longitudinal study with fewer patients, more covariates, and a downstream regression task. Finally, the DFCI dataset is an EHR dataset with irregularly recorded observations, the most patients, the most covariates, and a downstream classification task. The results from the computational experiments demonstrate that `med.knn` performs well across this range of diverse case studies. In particular, we show that this method performs well on datasets with: 1) large or small numbers of patients, 2) large or small numbers of covariates, and 3) regularly or irregularly recorded observations. Moreover, the application of `med.knn` for imputation led to improved downstream predictive performance on two binary classification tasks and one regression task.

Prior to training the downstream models, we do not perform any further preprocessing on the imputed data, so we preserve the correlation structure of the original dataset. As a

result, since these are real-world datasets, there may be unexpected correlations between the predictors which impact the accuracy of the downstream models. One could apply PCA or another dimensionality-reduction method to transform the feature space prior to training downstream models on the imputed datasets. However, this analysis is outside of the scope of this set of computational experiments.

In the Percentage of Missing Data experiments, we observe that increased imputation accuracy does not always translate into increased downstream model accuracy. For example, on the DFCI dataset, `bpca` performs poorly on the imputation task (see Figure 2.1), but is one of the top-performing methods on the downstream predictive task (see Figure 2.8). This is possible because in the downstream predictive task, some features are more significant than others, so having a large imputation error on the insignificant features may only result in a small decline in downstream model accuracy. However, we also observed that in all datasets, `med.knn` consistently performed best on both the imputation and downstream tasks, by a significant margin in most cases. These results suggest that for all three of the real-world datasets considered here, `med.knn` leads to improvements in imputation accuracy on the clinically significant covariates in each downstream model.

In the OPP experiments, the major trend that we observe is that the `med.knn` method performs significantly better with more time series data. For example, in the FHS dataset, the imputation accuracy and downstream performance of `med.knn` improves dramatically as OPP increases from one to four. This makes sense because as we include more OPP in the dataset, there is more relevant information available to impute the missing covariates for each patient. We expect that this explains why the relative improvement of `med.knn` is less significant on the DFCI dataset for several of the experiments. In this dataset, over half of the patients have a single observation, so there is limited time series available to fill in the missing values for these patients. In contrast, in the FHS dataset, every patient has 10 observations in the full dataset, so there is more data available to aid the imputation.

In the MNAR experiments, we demonstrate that `med.knn` works under missing data mechanisms that are frequently encountered in practice. Longitudinal studies often contain

systematic missing information on some clinical examinations based on decisions made by the designers of the study. For example, the FHS dataset has expanded over time as clinicians have incorporated more and more variables that are suspected to be correlated with heart disease [223]. However, since some of these variables were not recorded initially, they are systematically missing from this dataset. In EHR datasets, clinical covariates recorded for each visit typically vary based the health condition of the patient. Patients at higher risk are likely to undergo more detailed medical examinations, resulting in fewer missing values. Through the MNAR experiments for each case study, we show that `med.knn` is an effective method for imputing missing values under these specific mechanisms of missing data for longitudinal studies and EHR datasets.

2.4 Scaling Experiments on Simulated Clinical Datasets

In this section, we present scaling experiments on simulated clinical datasets. In Section 2.4.1, we describe the data generation process which allows us to construct simulated longitudinal clinical datasets with $10,000$'s of observations and 100's of features. In Section 2.4.2, we describe the experimental setup of the scaling experiments, which considers two variations of the `med.knn` method. In Section 2.4.3, we report the results of the scaling experiments, including the imputation accuracy and timing results.

2.4.1 Simulated Data: Synthea

We create synthetic EHR to test the performance of the algorithm in higher instances of both the number of observations and the number of features using the Synthea synthetic patient population simulator. It constitutes an open-source, synthetic patient generator that aims to model the medical history of patients using specific demographic information [347]. Patient records are generated using simulation processes that follow disease progression patterns published in the medical literature. For each synthetic patient, Synthea data contains a complete medical history, including medications, allergies, medical encounters, and social

determinants of health. We pre-processed the records combining them into a single dataset that contains a summary of all the information available at each visit.

Since we leverage this data source for experiments testing the scalability of the algorithm, we do not limit the amount of observations to a specific number. Each patient in the data is associated on average with 20 distinct visits (observations). We aggregate the EHR into 344 distinct features. Each experiment randomly samples a subset of these features to compare the computational time needed by the algorithm. The covariates that comprise the data include demographic characteristics, diagnosis and procedure codes, medical prescriptions, and lab test results. We do not include any downstream prediction task.

2.4.2 Experimental Setup for the Scaling Experiments

In this section, we go over the experimental setup for the scaling experiments. We use synthetically generated data for EHR varying both the number of observations n and the number of features p. Our goal is to evaluate the scaling performance and accuracy of the algorithm comparing the two proposed methods for tuning the hyperparameters α_d and h_d.

One of the most well-established approach for hyperparameter tuning in machine learning is K-fold cross-validation [192]. In the time series setting, [22] showed that this technique is applicable for time series models, in particular for the case of autoregression models. However, due to the large number of combinations of different values for α_d and h_d, in the case of `med.knn`, the computation time for the K-fold cross-validation scales at an quadratic rate as the number of covariates increases. For this reason, we propose a custom tuning procedure to select the hyperparameters. We conduct a series of experiments comparing the following hyperparameter selection processes:

1. **Grid Search**: This approach uses the well-established 10-fold cross-validation process to determine the hyperparameters h_d and α_d for every variable. Prior to solving the algorithm, 10% of the values of each feature are artificially removed. A set of values is defined and all their combinations are evaluated for each feature individually when solving the reduced version of the dataset. The grid for α_d was set to $[0.0, 0.1, \ldots, 1.0]$

and for h_d to $[90, 180, 365, 1000]$.

2. **Custom Tuning**: The custom tuning procedure proposed in Section 2.2.4. This is a heuristic method to decompose the problem into multiple parts, first learning h_d for each covariate, and then learning α_d for each covariate. This approach does not involve cross-validation and allows for parallel computations as the problem is fully decoupled.

For each experiment, we evaluate the imputation accuracy of each approach using the MAE and RMSE metrics, as defined in Equations 2.30 and 2.31. In addition, we also compare their scaling performance by measuring the average time needed for completion. In these experiments, we did not consider the prediction task as in Section 2.3. Here, we limit the types of experiments only to Percentage of Missing Data following the experimental set up of Section 2.3.3.

We vary the number of features between $[50, 100, 200, 300]$ and the number of observations between $[1000, 12500, 25000, 50000, 75000]$. These bounds were chosen as they represent the most common spectra of problem sizes that we encounter in healthcare applications. We repeat all experiments for five random seeds and average the results.

2.4.3 Results of the Scaling Experiments

In this section, we present the results from the scaling experiments. In Figure 2.11, we demonstrate the timing results. While both the methods scale to the largest problem size with $n = 75000$ observations and $p = 300$ features, the Custom Tuning procedure is -60.42% faster than Grid Search; the traditional cross-validation procedure. Across all experiments, Custom Tuning is on average -87.05% faster than Grid Search. We notice that for the lower problem sizes, the Custom Tuning approach leads almost instantaneous algorithm completion while Gridsearch requires up to 12 hours to solve.

Figure 2.12 presents the results referring to imputation accuracy. The two procedures lead to minimal differences in imputation performance. Across all experiments, the Custom Tuning procedure is slightly more accurate than the GridSearch procedure, with an average

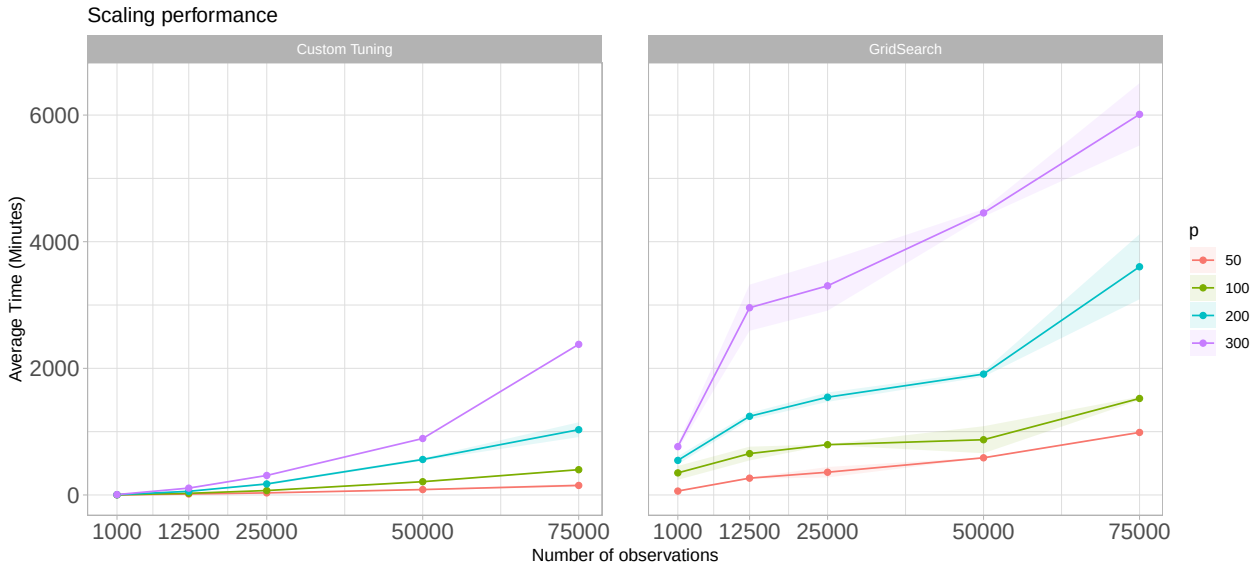


Figure 2.11: Average time for MedImpute methods to complete imputation tasks on the Synthea dataset using different procedures for hyperparameter tuning, with varying numbers of observations n and features p in the dataset.

improvement of -4.36% in MAE. The gap between the two processes is larger when $n \in [25000, 50000]$ leading to an average reduction of -8.81% of the imputation error. We also note that only when $n = 1000$, GridSearch as the MAE is increased on average by 2.82% by the new method. In all other combinations, Custom Tuning leads to more accurate results with the maximum improvement reaching a reduction of 10.48% ($n = 50000$, $p = 100$).

2.4.4 Discussion of the Scaling Experiments on Simulated Clinical Datasets

The results from the scaling experiments demonstrate that the custom tuning procedure for the MedImpute hyperparameters α_d and h_d is highly effective and efficient. In particular, the proposed method significantly reduces the computational time required, while also giving a slight improvement in imputation accuracy as well compared to traditional cross-validation. Using the methodology, we are able to scale the algorithm to higher problem instances without sacrificing its imputation performance.

An analysis of the runtime complexity of the two hyperparameter selection methods

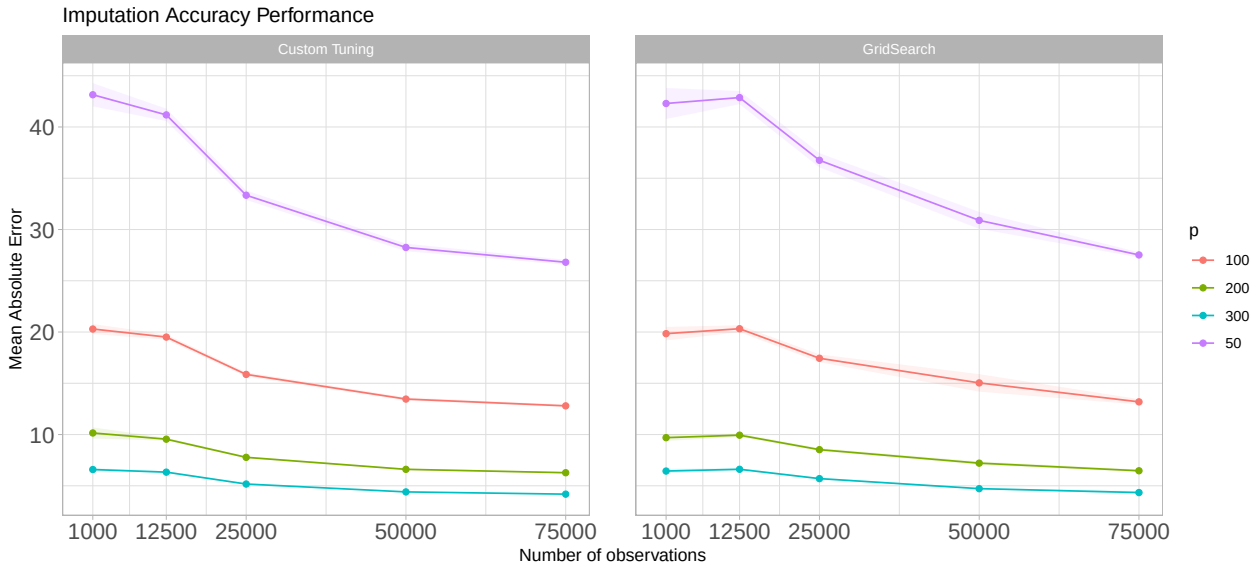


Figure 2.12: Average MAE imputation errors for MedImpute methods on the Synthea dataset using different procedures for hyperparameter tuning, with varying numbers of observations n and features p in the dataset.

provides further insights into these results. The key bottleneck of the `med.knn` algorithm is computing the K-NN assignment on $\mathbf{X}$ to update $\mathbf{Z}$ in each CD step, which requires $\mathcal{O}(n \log n)$ operations. The Grid Search procedure requires $\mathcal{O}(p^2)$ iterations to identify the best values for α_d and h_d, so the complete runtime for this method is $\mathcal{O}(np^2 \log n)$. On the other hand, the Custom tuning procedure only requires $\mathcal{O}(p)$ iterations because each hyperparameter for each covariate can be computed independently of the remaining covariates. As a result, this method scales in a linear fashion with respect to the number of covariates, and the full runtime is $\mathcal{O}(np \log n)$.

Despite these theoretical asymptotic runtime guarantees, we recognize that the `med.knn` method with the Custom Tuning procedure for hyperparameter tuning still takes up to 16 hours in datasets with $n \sim 50{,}000$ observations. However, given that the imputation task usually takes place once in the pre-processing part of the data analysis, we believe that the time cost is not significantly high. Moreover, the Custom tuning process allows for decoupling the problem in smaller instances. Thus, the application of parallel computing techniques can further improve the scaling performance of the algorithm.

2.5 Discussion

MedImpute is an extension of the OptImpute framework introduced by [32]. MedImpute uses the same optimization approach to solving the missing data problem. However, the optimization formulation is significantly different and more general than the OptImpute formulations in order to incorporate additional time series information present in cross-sectional data. The new formulation provides a structured way of accounting for observations from the same entity and re-weighting the objective function to incorporate time series information. As a result, the resulting imputation algorithm `med.knn` from the MedImpute framework outperforms `opt.knn` from the OptImpute framework and other benchmark imputation methods on real-world clinical datasets with patients observed over time.

In the MedImpute formulation, two new parameters are introduced, α_d, h_d, that are specific to each covariate d. The proposed Custom Tuning procedure allows for learning the values of these parameters more efficiently compared to a traditional Grid Search approach. In addition, these parameters are interpretable in a clinical context, yielding insights regarding the significance of time in their determination. For example, in the FHS dataset, we learn different values of α_d for chronic disease indicators such as Type 2 Diabetes Mellitus (T2DM) and lab values such as Systolic Blood Pressure (SBP). It is likely that an individual diagnosed with T2DM will continue to have this diagnosis regardless of the other covariates [8], so MedImpute finds α_{d} relatively close to 1 for this feature. On the other hand, the lab measurement of SBP may vary significantly during a single day [236], so previous observations of this covariate from the same individual provide relatively less information. For this feature, MedImpute finds α_d closer to 0 so that the K-nearest neighbors are weighted more heavily in the imputation. In addition, we learn h_d to determine the relative weights that we give to observations of feature d from the same individual based on time elapsed. MedImpute selects higher values of h_d for features that change slowly over time such as the BMI and lower values for features that change rapidly over time such as SBP.

Beyond the healthcare setting, cross-sectional datasets are also quite common in other areas such as finance and economics. Our algorithm can be generalized and applied to any

data where there is a time series component and multiple observations are tied to the same entity. The entity may represent a patient, as we portray in this work, or something else that is observed over time such as a financial organization, region, or country. Therefore, the MedImpute imputation framework and the associated `med.knn` algorithm may be applied to impute missing values in other domains as well.

2.6 Conclusions

In this chapter, we propose the optimization framework MedImpute that addresses the missing data problem for multivariate data in time series encountered in medical applications. We introduce a new imputation algorithm `med.knn` that yields high quality solutions using optimization techniques combined with fast first-order methods. Through computational experiments on three real-world clinical datasets, including two longitudinal studies and one EHR dataset, we show that `med.knn` offers statistically significant gains in imputation quality over state-of-the-art imputation methods, which leads to improved out-of-sample performance on downstream tasks. Through scaling experiments on a synthetic EHR dataset, we demonstrate that `med.knn` can be applied to complete datasets with 10,000's of observations and 100's of features. As a flexible, accurate, and intuitive approach, MedImpute has the potential to become an indispensable tool for applications with longitudinal missing data. Promising areas for future work include: (1) applications of this method to longitudinal datasets that are not related to healthcare, (2) additional experiments to assess the performance on downstream predictive tasks with transformed feature spaces, (3) extensions of the optimization framework to incorporate more specialized structure that is present in longitudinal healthcare datasets.

Chapter 3

Interpretable Clustering: An Optimization Approach

State-of-the-art clustering algorithms provide little insight into the rationale for cluster membership, limiting their interpretability. In complex real-world applications, the latter poses a barrier to machine learning adoption when experts are asked to provide detailed explanations of their algorithms' recommendations. We present a new unsupervised learning method that leverages MIO techniques to generate interpretable tree-based clustering models. Utilizing a flexible optimization-driven framework, our algorithm approximates the globally optimal solution leading to high quality partitions of the feature space. We propose a novel method which can optimize for various clustering internal validation metrics and naturally determines the optimal number of clusters. It successfully addresses the challenge of mixed numerical and categorical data and achieves comparable or superior performance to other clustering methods on both synthetic and real-world datasets while offering significantly higher interpretability.

3.1 Introduction

Clustering is the unsupervised classification of patterns, observations, data items, or feature vectors, into groups. The clustering problem has been addressed in many machine learning contexts where there is no clear outcome of interest, such as data mining, document retrieval, image segmentation, and pattern classification; this reflects its broad appeal and usefulness in exploratory data analysis [162]. In many such problems, there is little prior information available about the data, and the decision-maker must make as few assumptions about the data as possible. It is under these restrictions that clustering methodology is particularly appropriate for the exploration of relationships between observations to make an assessment, perhaps preliminary, of their structure.

Unlike supervised classification, there are no class labels and thus no natural measure of accuracy. Instead, the goal is to group objects into clusters based only on their observable features, such that each cluster contains objects with similar properties and different clusters have distinct features. There have been numerous approaches to generating these clusters. Partitional methods such as K-means provide a single partition of the data into a fixed number of clusters [222]; these methods have been improved by new initialization methods in recent decades [13]. Hierarchical methods produce a nested series of partitions based on a distance metric [321]. Other more sophisticated methods include model-based clustering and density-based clustering which are better able to capture clusters of irregular shape or varied density [162, 115].

The end product of a clustering algorithm is a partition of the dataset. In some cases, this final cluster assignment is sufficient for the machine learning purpose, such as when one wants to simply assess the separability of the data points into distinct clusters or use it as a preprocessing step in certain prediction tasks. However, in many other decision-making applications, there is a need to interpret the resulting clusters and characterize their distinctive features in a compact form [127]. For example, consider a medical setting in which we seek to group similar patients together to understand subgroups within a patient base. In this application, it is critical to understand how the resulting clusters differ, whether by

demographics, diagnoses, or other factors.

While the importance of cluster interpretability is well-understood, there has been limited success in addressing the issue [101]. None of the clustering algorithms described above were constructed with a goal of interpretability in the original feature space. They therefore require a post-processing step to synthesize the cluster meanings. The notion of cluster representation was introduced by [109] and was subsequently studied by [96] and [324]. The representation of a cluster of points by its centroid has been popular across various applications [276]. This works well when the clusters are compact or isotropic, but fails when the clusters are elongated or non-isotropic [176]. These clusters can be better characterized computing additional metrics, such as the variance in each dimension. However, this increases the number of summary statistics used for each cluster and creates a high burden in interpretation, especially when the number of features grows large. Another common approach is the visualization of clusters on a two-dimensional graph using Principle Component Analysis (PCA) projections [178, 280]. However, in reducing the dimensionality of the feature space, PCA obscures the relationship between the clusters and the original variables.

Tree-based supervised learning methods, such as Classification and Regression Trees (CART), [56] are a natural fit for problems that prioritize interpretability, since their feature splits and decision paths offer insight into the differentiating features between members in each leaf. Most recursive partitioning algorithms generate trees in a top-down, greedy manner, which means that each split is selected in isolation without considering its effect on subsequent splits in the tree. [27, 34] have proposed a new algorithm which leverages modern MIO techniques to form the entire decision tree in a single step, allowing each split to be determined with full knowledge of all other splits. The Optimal Classification Trees (OCT) algorithm enables the construction of decision trees for classification and regression that have performance comparable with state-of-the-art methods such as random forests and gradient boosted trees without sacrificing the interpretability offered by a single tree.

A general hybrid approach can leverage such methods by first running a partitional or hierarchical clustering method and using the resulting assignments as class labels. The data

can then be fit using a classification tree, in which each leaf is given a cluster label based on the most common assignment of observations in that leaf, and the decision paths leading to each cluster's leaves give insight into the differentiating features [176]. [156] use decision trees to interpret and refine hierarchical clustering results for global sea surface temperatures. While these trees give an explicit delineation of cluster attributes, the methods involve a two-step process of first building the clusters and subsequently identifying their differentiating features. Thus, the main clustering mechanism utilizes a different architecture compared to the decision tree which might be hard to capture with univariate feature splits.

Several algorithms have been proposed to build interpretable clusters, where interpretability is a consideration during cluster creation rather than considered as a later analysis step. [70] presented a method that constructs binary clustering trees characterized by a novel transformation of the feature space. Further efforts focused on alternative measures for feature selection in the transformation function as well as new algorithmic implementation schemes [18]. In both of these cases, the feature space transformation involved in these methods takes a toll on interpretability. Other researchers have proposed methods to construct decision trees in the original feature space, which more closely matches our objective. [219] introduced the idea of translating a clustering problem to a supervised problem that is amenable to decision tree construction. A modified purity criterion is used to evaluate splits in a way that identifies dense regions as well as sparse regions. However, this method requires additional pre-processing through the introduction of synthetic data in order to create a binary classification setting. [50] also proposed a general top-down tree induction framework with applicability to clustering ("Predictive Clustering Trees") as well as other supervised learning tasks. [131] developed another clustering algorithm, Clustering sing Unsupervised Binary Trees (CUBT), which forms greedy splits to optimize a cluster heterogeneity measure. Though these algorithms make progress towards the goal of constructing clusters directly using trees, they both employ a greedy splitting approach and do not offer flexibility in the choice of cluster validation criterion.

The need for accurate and interpretable machine learning methods is undoubtedly present,

being voiced even from regulatory organizations such as the European Union [148]. Even though tree-based methods have been introduced, no existing interpretable unsupervised learning algorithm can accurately partition the feature space both for numerical and categorical data.

3.1.1 Contributions

Motivated by the limitations of existing solutions to interpretable clustering, we develop a novel tree-based unsupervised learning method that leverages traditional optimization and machine learning techniques to obtain interpretable clusters with comparable or superior performance when compared to existing algorithms. Our contributions are as follows:

1. We provide an MIO formulation of the unsupervised learning problem that leads to the creation of globally optimal clustering trees, motivating our new algorithm *Interpretable Clustering via Optimal Trees* (ICOT). Our method builds upon the OCT algorithm and extends it to the unsupervised setting. In ICOT, interpretability is taken into consideration during cluster creation rather than considered as a later analysis step.

2. We provide an implementation of our method with an iterative CD approach that scales to larger problems, well-approximating the globally optimal solution. We use widely two established validation criteria, the Silhouette Metric [297] and the Dunn Index [107], as the algorithm's objective function. We propose additional techniques that leverage the geometric principles of cluster creation to improve the algorithm's efficiency. Furthermore, we introduce sampling heuristics that recover fast, high-quality solutions in our empirical experiments and provide a complexity analysis of the local search procedure for one iteration of the algorithm.

3. We develop our algorithm in a way such that tuning of the tree's complexity is redundant. This is enabled by the fact that our loss functions take into account both intra-cluster density as well as inter-cluster separation. The user can optionally tune the algorithm

by selecting the maximum depth of the tree and the minimum number of observations in each cluster.

4. We propose a solution to the incorporation of both mixed numerical and categorical data. Our re-weighted distance measure prevents a single variable type from dominating the distance calculation and allows users to optionally tune the balance the two types of covariates.

5. We evaluate the performance of our method against various clustering approaches across synthetic datasets from the Fundamental Clustering Problems Suite (FCPS) [337] which offer different levels of variance and compactness. We demonstrate ICOT's superior performance against a two-step supervised learning method across both the Silhouette Metric and Dunn Index, offering a 27.8% and 352.7% score improvement respectively. We also compare ICOT against several state-of-the-art methods that represent various clustering approaches, namely partitional, hierarchical, model-based, and density-based clustering. We find that ICOT is competitive against these methods across multiple internal validation criteria.

6. We provide examples of how the algorithm can be used in real-world settings. We perform clustering on patients at risk of cardiovascular disease from the FHS dataset [209, 122] to identify similar patient profiles and group economic profiles of European countries during the Cold War [193]. Through these experiments, we illustrate the effect of varying key parameters in the ICOT algorithm. We also compare ICOT to other state-of-the-art algorithms in the FHS experiment and to CUBT in the economic profile experiment. We discuss the interpretability of the methods as well as their performance on the internal validation criteria.

7. Finally, we test the capability of the algorithm to scale to large problem instances using both the FCPS as well as real-world data from a Boston-based bike sharing program. We demonstrate that our suggested heuristic techniques do not significantly impact the quality of the recovered solutions. In addition, our experiments illustrate that ICOT

can efficiently handle datasets of sizes up to hundreds of thousands of observations.

The structure of the chapter is as follows. In Section 3.2, we formulate the problem of optimal tree creation within an MIO framework. Section 3.3 provides a comprehensive description of the algorithm implementation. In Sections 3.4 and 3.5, we conduct a range of experiments using synthetic and real-world datasets to evaluate the performance and interpretability of our method compared to other state-of-the-art algorithms. In Section 3.6, we investigate the effect of our scaling methods on runtime and solution quality. In Section 3.7, we discuss the key findings from our work and in Section 3.8 we include our concluding remarks.

3.2 MIO Formulation

In this section, we present an MIO approach which allows us to construct globally optimal tree-based models in an unsupervised learning setting. In Section 3.2.1, we provide an overview of the MIO framework introduced by [27, 34]. Section 3.2.2 introduces the validation criteria that are used as objective functions in the optimization problem. In Section 3.2.3, we outline the complete ICOT formulation for one of the loss functions considered.

3.2.1 The Optimal Trees Optimization Framework

The OCT algorithm formulates tree construction using MIO which allows us to define a single problem, as opposed to the traditional recursive, top-down methods that must consider each of the tree decisions in isolation. It allows us to consider the full impact of the decisions being made at the top of the tree, rather than simply making a series of locally optimal decisions, avoiding the need for pruning and impurity measures.

We are given the training data $(\mathbf{X}, \mathbf{Y})$, containing n observations $(\mathbf{x_i}, y_i)$, $i = 1, \ldots, n$, each with p features and a class label $y_i \in \{1, \ldots, K\}$ as an indicator of which of the K potential labels is assigned to point i. We assume without loss of generality that the values of each training vector are normalized such that $\mathbf{x_i} \in [0, 1]^p$. A decision tree recursively

partitions the feature space to identify a set of distinct, hierarchical regions that form a classification tree. The final tree $\mathcal{T}$ is comprised of nodes that can be categorized in:

- Branch Nodes: Nodes $t \in \mathcal{T}_\mathcal{B}$ apply a split with parameters $\mathbf{a}$ and b. For observation i, if the corresponding vector $\mathbf{x_i}$ satisfies the relation $\mathbf{a}^T\mathbf{x_i} < b$, the point will follow the left branch from the node. Otherwise it takes the right branch.

- Leaf Nodes: Nodes $t \in \mathcal{T}_\mathcal{L}$ assign a class to all the points that fall into them. Each leaf node is characterized by one class which is generally determined by the most frequently occurring class among the observations that belong to it.

First, we formally define the constraints that construct the decision tree. We use the notation $p(t)$ to refer to the parent node of node t, and $A(t)$ to denote the set of ancestors of node t. We define the split applied at node $t \in \mathcal{T}_\mathcal{B}$ with variables $\mathbf{a}_t \in \mathbb{R}^p$ and $b_t \in \mathbb{R}$. The vector $\mathbf{a}_t$ indicates which variable is chosen for the split, meaning that $a_{jt} = 1$ for the variable j used at node t. b_t gives the threshold for the split, which is between $[0, 1]$ after normalization of the feature vector. If a branch node does not apply a split, then we model this by setting $\mathbf{a}_t = \mathbf{0}$ and $b_t = 0$. Together, these form the constraint $\mathbf{a}_t^T x < b_t$. The indicator variables d_t are set to 1 for branch nodes and 0 for leaf nodes. Using the above variables, we introduce the following constraints that allows us to model the tree structure (for a detailed analysis of the constraints, see [27]):

$$\sum_{j=1}^{p} a_{jt} = d_t, \ \forall t \in \mathcal{T}_\mathcal{B}, \tag{3.1}$$

$$0 \leq b_t \leq d_t, \ \forall t \in \mathcal{T}_\mathcal{B}, \tag{3.2}$$

$$a_{jt} \in \{0, 1\}, \ j = 1, \ldots, p, \ \forall t \in \mathcal{T}_\mathcal{B} \tag{3.3}$$

We next enforce the hierarchical structure of the tree. Branch nodes are allowed to apply a split only if their parent nodes apply a split:

$$d_t \leq d_{p(t)}, \ \forall t \in \mathcal{T}_\mathcal{B} \backslash \{1\} \tag{3.4}$$

Next we present the corresponding constraints that track the allocation of points to leaves. For this purpose, we introduce the indicator variables $z_{it} = \mathbb{1}\{x_i \text{ is in node } t\}$ and $l_t = \mathbb{1}\{\text{leaf } t \text{ contains any points}\}$. We let N_{min} be a constant that defines the minimum number of observations required in each leaf. We apply the following constraints:

$$z_{it} \leq l_t, \ \forall t \in \mathcal{T}_{\mathcal{L}}, \tag{3.5}$$

$$\sum_{i=1}^{n} z_{it} \geq N_{min} l_t, \ \forall t \in \mathcal{T}_{\mathcal{L}} \tag{3.6}$$

We also enforce each point to belong to exactly one leaf:

$$\sum_{t \in \mathcal{T}_{\mathcal{L}}} z_{it} = 1, \ i = 1, \ldots, n \tag{3.7}$$

Finally, we introduce constraints that force the assignments of observations to leaves to obey the structure of the tree given by the branch nodes. We want to apply a strict inequality for points going to the lower leaf. To accomplish this, we define the vector $\epsilon \in \mathbb{R}^p$ as the smallest separation between two observations in each dimension p, and ϵ_{max} as the maximum over this vector.

$$a_m^\intercal x_i \geq b_t - (1 - z_{it}), \ i = 1, \ldots, n, \quad \forall t \in \mathcal{T}_{\mathcal{B}}, \quad \forall m \in A_R(t) \tag{3.8}$$

$$a_m^\intercal (x_i + \epsilon) \leq b_t + (1 + \epsilon_{max})(1 - z_{it}), \ i = 1, \ldots, n, \quad \forall t \in \mathcal{T}_{\mathcal{B}}, \quad \forall m \in A_L(t) \tag{3.9}$$

In the classification setting the objective function of MIO formulation is comprised of two components, prediction accuracy and tree complexity. The tradeoff between those two parameters is controlled by the complexity parameter α. Given the training data $(\mathbf{x_i}, y_i)$, $i = 1 \ldots n$, a general formulation of the objective function is the following:

$$\underset{T}{\text{minimize}} \quad R_{xy}(T) + \alpha |T|$$

where $R_{xy}(t)$ is a loss function assessed on training data and $|T|$ is the number of branch nodes in the tree T.

The above model can be used as an input for an MIO solver. Empirical results suggest that such a model leads to optimal solutions in minutes when the maximum depth of the tree is small (approximately 4). Effectively, the rate of finding solutions is directly dependent to the number of binary variables z_{it} and therefore a faster implementation was needed for more complex problems. For this reason, the authors introduced the idea of warm starts as the initial starting point of the method. Using a high-quality integer feasible solution as a warm start increases the speed of the algorithm and provides a strong initial upper bound on the final solution. In addition, heuristics, like local search, allow a further speed up as shown in [27, 34] that leads to a good approximation of the optimal solution.

3.2.2 Loss Functions for Cluster Quality

Clustering validation, the evaluation of the quality of a clustering partition [229], has long been recognized as one of the vital issues essential to the success of a clustering application [220]. External clustering validation and internal clustering validation are the two main categories of clustering quality metrics. The main difference lies in whether or not external labels are used to assess the clusters; internal measures evaluate the goodness of a clustering structure without respect to ground-truth labels [198]. An example of external validation measure is entropy, which evaluates the "purity" of clusters based on the given class labels [359]. True class labels are not present in real-world datasets, and thus these cases necessitate the use of internal validation measures for cluster validation.

We will consider two internal validation measures as loss functions for our MIO formulation of our problem. The chosen loss functions consider the global assignment of observations to clusters. The score of a clustering assignment depends on both the compactness of the observations within a single cluster, as well as its separation from observations in other clusters. Compactness measures how closely related the objects in a cluster are. Separation measures how distinct a cluster is from other clusters. Several internal validation metrics have

been proposed to balance these two objectives [220]. Two common criteria, the Silhouette Metric and Dunn Index, are outlined below.

Silhouette Metric The Silhouette Metric introduced by [297] compares the distance from an observation to other observations in its cluster relative to the distance from the observation to other observations in the second closest cluster. The Silhouette Metric for observation i is computed as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}, \tag{3.10}$$

where $a(i)$ is the average distance from observation i to the other points in its cluster, and $b(i)$ is the average distance from observation i to the points in the second closest cluster. In other words, $b(i) = \min_k b(i,k)$ where $b(i,k)$ is the average distance of i to points in cluster k, minimized over all clusters k other than the cluster that point i is assigned to. From this formula it follows that $-1 \leq s(i) \leq 1$.

When $s(i)$ is close to 1, one may infer that the i^{th} sample has been "well-clustered", i.e. it was assigned to an appropriate cluster. If observation i has score close to 0, it suggests that it could also be assigned to the nearest neighboring cluster with similar quality. If $s(i)$ is close to -1, one may argue that such a sample has been assigned to the wrong partition. These individual scores can be averaged to reflect the quality of the global assignment.

$$SM = \frac{1}{n} \sum_{i=1}^{n} s(i), \tag{3.11}$$

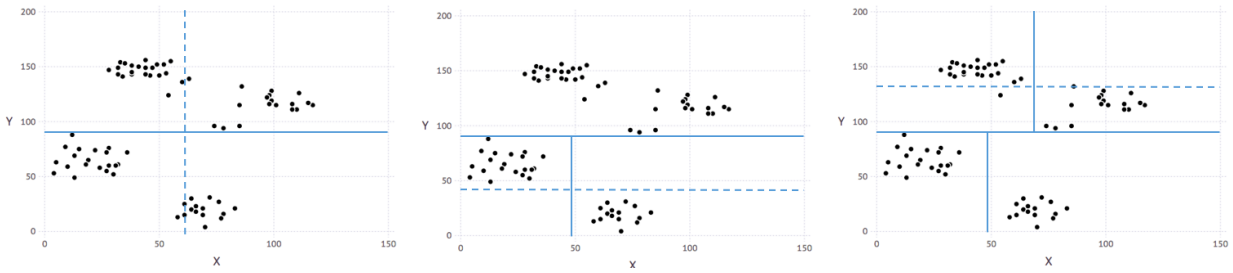
Dunn Index The Dunn Index [107] characterizes compactness as the maximum distance between observations in the same cluster, and separation as the minimum distance between two observations in different clusters. The metric is computed as the ratio of the minimum inter-cluster separation to the maximum intra-cluster distance.

$$DI = \frac{\min\limits_{1 \leq i < j \leq m} \delta(C_i, C_j)}{\max\limits_{1 \leq k \leq m} \Delta_k}, \tag{3.12}$$

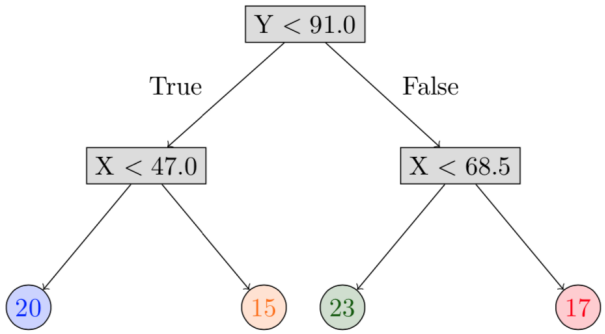
where we let the maximum distance of cluster C be denoted by Δ_C and the distance between clusters i and j be denoted by $\delta(C_i, C_j)$. If the dataset contains compact and well-separated clusters, the distance between the clusters is expected to be large and the diameter of the clusters is expected to be small. Thus, large values of the metric correspond to better partitions and signify that the distance between clusters is large relative to the distance between points within a cluster.

We provide an example to illustrate how an internal validation criterion can be used to geometrically partition the space through a decision tree. In Figure 3.1, we cluster observations from the Ruspini dataset [301] using the Silhouette Metric. In Figure 3.1a, the algorithm identifies the best candidate splits on both features, x_1 and x_2, at the root node, and then compares their resultant cluster scores, as measured by the Silhouette Metric. The x_2 split provides a better cluster assignment, so this split is chosen as denoted by the solid line. After the first data partition, splits are considered for each of the child nodes, which corresponds to further separating the lower and upper halves of the graph. Upon identification of candidate x_1 and x_2 splits on the left child node, the x_1 split is chosen based on the Silhouette Metric of the global cluster assignment, as shown in Figure 3.1b. The process is then completed for the right child node, and an x_1 split is also chosen here in Figure 3.1c. Now, each of the four leaves is evaluated, which corresponds to exploring splits in the four quadrants defined by the solid blue lines. There are no splits within any of these four leaves that improve the overall score of the clustering assignment, so the tree construction is complete. The final tree is shown in Figure 3.1d. The resultant tree provides a final partition which clearly elucidates the distinguishing features of each group. We note that this example demonstrates a *greedy* tree construction. In the ICOT algorithm, all splits would be subsequently reoptimized with respect to the overall tree. However, in this case the greedy tree is able to provide the optimal partition.

Note that both of our considered criteria require the definition of at least two clusters since they both involve a pairwise distance computation between clusters to measure separation. As a result, calculations for the null-case are not considered. The determination of the best



(a) Root Node (b) Left Child Node (c) Right Child Node



(d) Final Tree

Figure 3.1: An example of a clustering tree built on the Ruspini dataset.

internal validation criterion for a given dataset remains an open question in the field of unsupervised learning theory [220]. As stated in [154], the Dunn Index is more computationally expensive and more sensitive to noisy data compared to the Silhouette Metric. It is also less robust to outliers compared to the Silhouette Metric which averages an observation-based score for the global assignment. However, empirical results suggest that the Dunn Index has superior performance in returning intuitive partitions of the data when they are well-separated.

3.2.3 The ICOT Formulation

The OCT framework needs to be modified to address an unsupervised learning task. We present changes in the original MIO formulation of OCT to be able to partition the data space into distinct clusters following the same structure and notation as in Section 3.2.1. We outline in detail the model for the Silhouette Metric loss function. The Dunn Index formulation follows closely and is thus omitted. There are two primary modifications in the ICOT formulation compared to the OCT:

1. The objective function is comprised solely by the chosen cluster quality criterion, such as the Silhouette Metric, and does not include any penalty for the tree complexity. The separation component of the validation criterion naturally controls the complexity of the tree and thus for the ICOT formulation the complexity parameter is rendered redundant.

2. Each leaf of the tree is equivalent to a cluster. Observations in different leaves are not allowed to belong to the same cluster.

The objective of the new formulation is to maximize the Silhouette Metric (SM) of the overall partition. The Silhouette Metric quantifies the difference in separation between a point and points in its cluster, versus the separation between that point and points in the second closest cluster.

Let d_{ij} be the distance (i.e. Euclidean) of observation i from observation j. We define K_t to be number of points assigned assigned to cluster t.

$$K_t = \sum_{i=1}^{n} z_{it},\ \forall t \in \mathcal{T}_{\mathcal{L}} \tag{3.13}$$

We define c_{it} to be the average distance of observation i from cluster t:

$$c_{it} = \frac{1}{K_t} \sum_{j=1}^{n} d_{ij} z_{jt},\ \forall i = 1, \ldots, n,\ t \in \mathcal{T}_{\mathcal{L}}. \tag{3.14}$$

We define r_i to be the average distance of observation i from all the points assigned in the same cluster:

$$r_i = \sum_{t \in \mathcal{T}_{\mathcal{L}}} c_{it} z_{it},\ \forall i = 1, \ldots, n. \tag{3.15}$$

We then let q_i denote the minimum average distance of observation i to the observations from the next closest cluster. We define auxiliary variables γ_{it} to enforce this constraint, such that γ_{it} an indicator of whether t is the second closest cluster for observation i.

$$q_i \geq \sum_{t \in \mathcal{T}_{\mathcal{L}}} \gamma_{it} c_{it},\ i = 1, \ldots, n. \tag{3.16}$$

$$\sum_{t \in \mathcal{T}_{\mathcal{L}}} \gamma_{it} = 1,\ i = 1, \ldots, n. \tag{3.17}$$

$$\gamma_{it} \leq M(1 - z_{it}),\ i = 1, \ldots, n,\ \forall t \in \mathcal{T}_{\mathcal{L}}. \tag{3.18}$$

Finally, to define the Silhouette Metric of observation i, we will need the maximum value between r_i and q_i which normalizes the metric.

$$m_i \geq r_i,\ i = 1, \ldots, n. \tag{3.19}$$

$$m_i \geq q_i,\ i = 1, \ldots, n. \tag{3.20}$$

The score for the Silhouette Metric for each observation is computed as $s(i)$ and the overall

score for the clustering assignment is then the average overall all the Silhouette Metric scores from the training population:

$$s_i = \frac{q_i - r_i}{m_i},\ i = 1, \ldots, n. \tag{3.21}$$

$$SM = \frac{1}{n} \sum_{i=1}^{n} s_i. \tag{3.22}$$

Putting all of this together gives the following MIO formulation for the ICOT model:

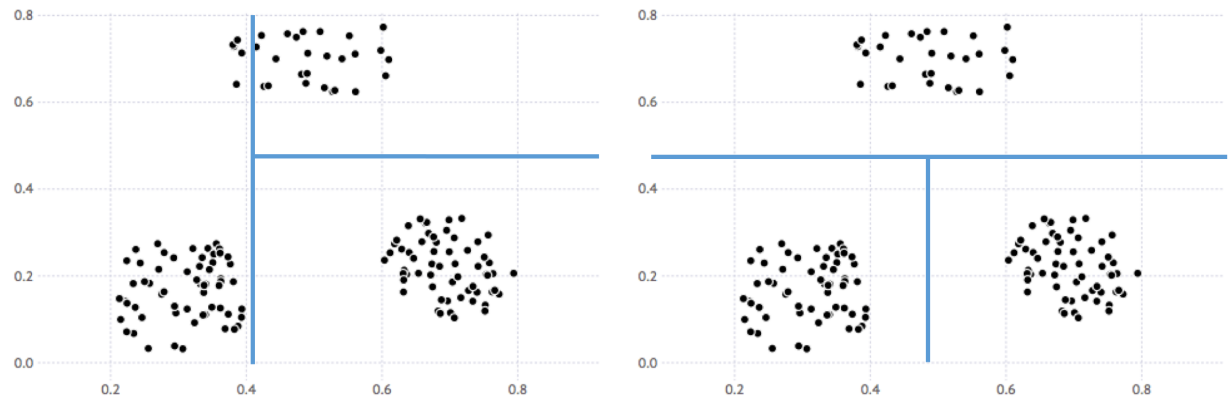
$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & -\frac{1}{n} \sum_{i=1}^{n} s_i & \\
\text{subject to} \quad & s_i = \frac{q_i - r_i}{m_i}, & i = 1, \ldots, n, \\
& m_i \geq q_i, & i = 1, \ldots, n, \\
& m_i \geq r_i, & i = 1, \ldots, n, \\
& q_i \geq \sum_{t \in \mathcal{T}_{\mathcal{L}}} \gamma_{it} c_{it}, & i = 1, \ldots, n, \\
& \sum_{t \in \mathcal{T}_{\mathcal{L}}} \gamma_{it} = 1, & i = 1, \ldots, n, \\
& \gamma_{it} \leq M(1 - z_{it}), & i = 1, \ldots, n,\ \forall t \in \mathcal{T}_{\mathcal{L}}, \\
& r_i = \sum_{\forall t \in \mathcal{T}_{\mathcal{L}}} c_{it} z_{it}, & i = 1, \ldots, n, \\
& c_{it} = \frac{1}{K_t} \sum_{j=1}^{n} d_{ij} z_{jt}, & i = 1, \ldots, n, \forall t \in \mathcal{T}_{\mathcal{L}}, \\
& K_t = \sum_{i=1}^{n} z_{it} & \forall t \in \mathcal{T}_{\mathcal{L}}, \\
& \sum_{j=1}^{p} a_{jt} = d_t, & \forall t \in \mathcal{T}_{\mathcal{B}}, \\
& 0 \leq b_t \leq d_t, & \forall t \in \mathcal{T}_{\mathcal{B}}, \\
& d_t \leq d_{p(t)}, & \forall t \in \mathcal{T}_{\mathcal{B}} \backslash \{1\},
\end{aligned}$$

$$
\begin{aligned}
& z_{it} \leq l_t, && \forall t \in \mathcal{T}_{\mathcal{L}}, \\
& \sum_{i=1}^{n} z_{it} \geq N_{min} l_t, && \forall t \in \mathcal{T}_{\mathcal{L}}, \\
& \sum_{t \in \mathcal{T}_{\mathcal{L}}} z_{it} = 1,\ i = 1, \ldots, n, && \\
& a_m^\intercal x_i \geq b_t - (1 - z_{it}), && i = 1, \ldots, n,\ \forall t \in \mathcal{T}_{\mathcal{B}},\ m \in A_R(t), \\
& a_m^\intercal (x_i + \epsilon) \leq b_t + (1 + \epsilon_{max})(1 - z_{it}), && i = 1, \ldots, n,\ \forall t \in \mathcal{T}_{\mathcal{B}}, ; m \in A_L(t), \\
& a_{jt}, d_t \in \{0, 1\}, && j = 1, \ldots, p,\ \forall t \in \mathcal{T}_{\mathcal{B}}, \\
& z_{it}, l_t \in \{0, 1\}, && i = 1, \ldots, p,\ \forall t \in \mathcal{T}_{\mathcal{L}}, \\
& \gamma_{it} \in \{0, 1\}, && i = 1, \ldots, n,\ \forall t \in \mathcal{T}_{\mathcal{L}}.
\end{aligned}
$$

Figure 3.2 illustrates the benefit of an optimization framework over greedy tree construction. The synthetic dataset seen in the figure has two dense lower regions and one less dense upper region. In a greedy approach, the first split separates the lower clusters and cuts through the upper cluster. While it is clearly better to split horizontally first (since it does not split a region), a greedy algorithm chooses the split without consideration of the possibility of future splits. Therefore, if the tree can only make one split, it is better to separate the lower clusters since they have such high density. ICOT's optimization approach considers the global tree structure, avoiding such pitfalls and identifying the true optimal partition. It starts by making a horizontal split and subsequently separates the high-density lower regions without cutting through the upper cluster. A globally optimal partition has Silhouette Metric score equal to 0.758 whereas the greedy tree yields only 0.688.

3.3 Algorithm Overview

In this section, we outline the practical details of the algorithm implementation. Section 3.3.1 describes ICOT's CD algorithm that approximates the globally optimal solution in an efficient and intuitive manner. Section 3.3.2 addresses the challenge of computing distance scores



(a) Greedy Tree Approach (b) ICOT Approach

Figure 3.2: An illustration in a synthetic example of a local optimum that might be identified by a greedy unsupervised learning algorithm.

in the presence of mixed numerical and categorical variables and introduces a solution for appropriately handling distance in this setting. Finally, in Section 3.3.3 we propose heuristics in our algorithm implementation which leverage the underlying structure of the data to more quickly traverse the search space and identify high-quality solutions.

3.3.1 Coordinate-Descent Implementation

The MIO formulation provides the optimization framework for our problem solving approach. In practice, the algorithm is implemented using a CD procedure which allows it to scale to much higher dimensions than directly solving the optimization problem. The implementation provides a good approximation of the optimal solution while still abiding by the same core principles of the original formulation.

ICOT initializes a greedy tree and subsequently runs a local search procedure until the objective value, a cluster quality measure, converges. This process is repeated from many different starting greedy trees, generating many candidate clustering trees. The final tree is chosen as the one with the highest cluster quality score across all candidate trees. This single tree is returned as the output of the algorithm.

The initial greedy tree is constructed from a single root node. A split is made on a

randomly chosen feature by scanning over all potential thresholds for splitting observations into the lower and upper leaves. At each candidate split, we compute the global score for the potential assignment. We choose the split threshold that gives the highest score and update the node to add the split if this score improves upon the global score of the current assignment. We perform the same search for each leaf that gets added to the tree, continuing until either the maximum tree depth is reached or no further improvement in our objective value is achieved through further splitting on a leaf.

Following the creation of the greedy tree, a local search procedure is performed to optimize the clustering assignment. Tree nodes are visited in a randomly chosen order, and various modifications are considered. A branch node has two options; it can be deleted, in which case it is replaced with either its lower or upper subtree, or a new split can be made at the node using a different feature and threshold. A leaf node can be further split into two leaves. At each considered node, the algorithm finds the best possible change and updates the tree structure only if it improves the objective from its current value. All nodes get added back to the list of nodes to search once an improvement has been found. The algorithm terminates when the objective value converges. The algorithm is explained further in Algorithm 3.

The user can specify to optimize either the Silhouette Metric or Dunn Index described in Section 3.2.2. These metrics penalize low separation, which naturally limits the depth of the tree. In traditional tree-based algorithms such as CART or OCT, the loss function improves with successive tree splits. Thus, these methods require a pruning step or additional parameter, such as a complexity penalty of maximum depth, to control the tree size. ICOT does not require the explicit control of tree size due to this natural balance between separation and compactness in the cluster quality metrics. This eliminates the need for setting an explicit K parameter, which is typically required in both partitional and hierarchical clustering methods. The tree continues to split until further splits no longer improve the quality of the overall assignment, and so the final number of leaves represents the optimal number of clusters.

The user can enforce further structure on the tree through setting the optional minimum bucket parameter, N_C. This controls the minimum number of observations that are required

Algorithm 3 ICOT Algorithm.

Input: Feature vectors $\mathbf{x}^1, \ldots, \mathbf{x}^n$
Output: Cluster assignments $y^1, \ldots, y^n$

Initialize a greedy tree, with clusters $c_1, \ldots, c_K$ and loss l_0.
Indices to search: $S = \{1, \ldots, K\}$; Loss: $l = l_0$.
while S not empty **do**
  for all $k \in S$ **do**
    if C_k is leaf node **then**
      Find best possible new split with loss $\hat{l}$.
    else
      Find best possible node modification, either through a different split or split deletion, with loss $\hat{l}$.
    end if
    if $\hat{l} < l$ **then**
      Update tree and add all leaves to S. $l \leftarrow \hat{l}$.
    else
      Remove k from S.
    end if
  end for
end while

in each leaf and effectively in each cluster. Note that there is not a monotonic relationship between the magnitude of N_C and the number of leaves (clusters) generated by the algorithm. Smaller minimum buckets may lead to smaller cluster counts due to the positive effect of isolated outlier clusters on the metrics; overfitting is difficult to quantify in an unsupervised learning setting because there is no ground truth to compare against, and thus the metrics do not naturally penalize single outliers. Thoughtful choice of the minimum bucket parameter allows ICOT to avoid creating clusters of single or small sets of outliers, which often lack meaning and generalizability in grouping tasks. Traditional methods, such as K-means, deal with outliers by increasing the K parameter and forcing the algorithm to provide with a higher number of clusters. N_C can significantly affect the clustering solution and should be cross-validated or experimented on in order to get accurate and intuitive results from ICOT. The maximum depth can be used to impose an upper bound on the number of clusters if desired, although this parameter does not address potential outlier issues.

The ICOT algorithm is implemented in Julia [46] and is available to academic researchers

under a free academic license.*

3.3.2 Mixed-Variable Handling

Both the Silhouette Metric and Dunn Index assess the quality of a given cluster assignment using the pairwise distance matrix of the observations. Distance is quantified differently for numerical and categorical variables and thus must be adjusted appropriately in the presence of mixed variable types. In the case of continuous features, the data are first normalized to be in the $[0, 1]$ range. The pairwise numerical distance matrix d^N is computed using the Euclidean distance between each pair of normalized variables. In the case of categorical features, distance is defined based on whether the observations take on different values. For example, if one observation takes on category A and another observation takes on category B on a given feature, the distance on this feature will be 1. The distance is zero if the observations take on the same value. For each pair of observations, these indicators are summed over all categories to define the categorical feature distance matrix d^C.

When the feature space includes both numerical and categorical variables, special consideration must be given to avoid over-weighting the categorical variables. In particular, categorical variables are often one-hot encoded (i.e. converted to binary 0/1 columns) to allow them to be treated as numerical in machine learning methods. This adjustment is insufficient in our case as it will result in placing too high of an importance on the categorical distance.

We handle this issue by taking a linear combination of the two separate distance matrices for numerical and categorical variables. We first compute separate distance matrices for the numerical and categorical features. We let S^N denote the set of indices for the numerical features, and S^C denote the categorical indices. The computations for d^N and d^C are explicitly defined in Equations 3.23 and 3.24.

$$d_{ij}^N = \sqrt{\sum_{k \in S^N} (x_k^i - x_k^j)^2} \tag{3.23}$$

*Please email icot@mit.edu to request an academic license for the ICOT package.

$$d_{ij}^{C} = \sum_{k \in S^C} \mathbb{1}\{x_k^i \neq x_k^j\} \tag{3.24}$$

We then compute the final distance matrix by taking a linear combination of these two matrices, given in Equation 3.25.

$$d_{ij} = \alpha d_{ij}^{N} + (1 - \alpha) d_{ij}^{C} \tag{3.25}$$

By default, the two distances are weighted according to their proportion of all covariates, so $\alpha = \frac{|S^N|}{|S^N| + |S^C|}$. The user can also specify an alternative α parameter. At $\alpha = 1$, the distance matrix only accounts for numerical covariates, whereas $\alpha = 0$ only considers disagreements in categorical variables.

3.3.3 Scaling Methods

Our CD procedure is more computationally intensive than the original OCT algorithm due to unique characteristics of clustering. In particular, we must compute a global clustering quality score at each split threshold evaluation, unlike classification tasks in which the loss change for a potential split can be assessed locally at the node. This global score assessment involves higher computational effort per split evaluation and thus motivates the development of more efficient search procedures. We introduce two scaling methods to take advantage of the geometric intuition behind cluster creation as well as existing clustering methods. We furthermore propose a subsampling approach to allow the algorithm to scale to much larger problems.

Restricted Geometric Search Space

ICOT leverages the geometric structure of the feature space by restricting the set of candidate splits to those with sufficient separation. An exhaustive search of candidate splits on a given numerical feature requires $n_k - 1$ threshold evaluations, where n_k is the number of observations in a given node. This is due to the fact that there are exactly $n_k - 1$ different possible partitions of the data on the given feature at node k (less if multiple observations

have the same value on this feature).

To improve the efficiency of our algorithm, we only consider a subset of these thresholds. For any feature, we refer to a threshold's gap as the separation between the observations directly below and above it. Since the quality of a clustering assignment is directly tied to the distance separating distinct clusters, the cluster quality will be superior when considering thresholds with large gaps. We take advantage of this intuition by skipping over thresholds with small gaps.

We control the extent of search space restriction through the parameter T. When considering a numerical feature split at node k, all threshold gaps for observations in the node are sorted ($n_k - 1$ values). Only thresholds above the T^{th} percentile of gap size are considered. For example, if $T = .9$ and $n_k = 100$, only the thresholds with the 10 largest gaps are considered, reducing the number of computations per node by 90%.

Figure 3.3 provides an illustration of how the Restricted Geometric Search would be applied in a simple example. When $T = 0.7$, ICOT will investigate only the top 30% of the gaps between observations. Thus only the larger, bold, gaps would be potential splits for a branch node that considers the covariate corresponding to the horizontal axis.

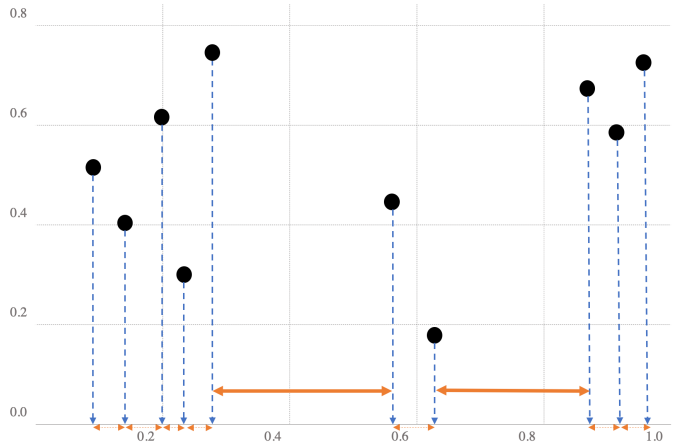


Figure 3.3: An example of the Restricted Geometric Search Function

K-means Warm Start

We also employ warm starts to more efficiently identify high-quality clustering trees. We leverage the K-means algorithm to partition the data into clusters and use OCT to generate a tree that reasonably separates these clusters. This becomes the starting point of ICOT's coordinate descent algorithm. The algorithm first runs K-means on the original data across various K parameters and selects the assignment that optimizes our chosen cluster quality criterion. The resulting assignments are used as class labels for the construction of a supervised classification tree using OCT. ICOT's CD procedure then begins from the resultant OCT tree rather than a greedy tree. Each leaf from the OCT tree becomes a separate cluster when initializing the ICOT algorithm, even though the predicted class labels may match between multiple leaves. Overall, the K-means warm start expedites tree initialization and improves the efficiency of the search procedure.

Bootstrapping

We introduce bootstrapping on the number of input observations, N. Our goal is to make the algorithm amenable to solve problems of larger sample size. This procedure involves subsampling a reduced population of size N_r and solving smaller problems N_{rep} times. This allows the algorithm to scale linearly with respect to the number of repetitions. It can be easily parallelized as it contains multiple independent sub-problems. Each iteration samples N_r observations without replacement and runs ICOT, returning a tree model which is then evaluated on a validation population. Upon completion of all N_{rep} iterations, the algorithm selects the best performing tree model on the validation criterion. Beyond improving the speed of the algorithm, bootstrapping provides a lot of flexibility to the user. The choice of N_r and N_{rep} may vary depending on the time constraints and the required quality of the final solution. We explore the latter in greater detail in Sections 3.6.2-3.6.2.

Complexity Analysis

We provide a brief analysis of the worst-case complexity for each iteration of the CD implementation of the algorithm. The argument is an extension of the complexity analysis for Optimal Classification Trees [108]. First, we consider the complexity of calculating our cluster quality criteria.

An initial step for the computation of any score is the construction of a distance matrix that contains all the distances between each point $i, j \in [N]$, the training population. The matrix creation involves $\frac{n(n-1)}{2}$ calculations, which has complexity $\mathcal{O}(n^2)$.

Silhouette Metric (SM): For each observation i, we must compute the average distance between i and the members of each cluster. If we have T nodes, and each cluster contains at most n points, this has complexity $\mathcal{O}(nT)$. We need to find the distance to the next-closest cluster for which i is not a member. As we iterate through each of the clusters, we track the closest distance found so far and update if it improves. We note that the number of clusters is $\mathcal{O}(T)$ and is upper bounded by the total number of nodes. This computation is repeated for all n observations. Thus, the complexity of computing the Silhouette Metric is

$$cp_{SM} = \mathcal{O}(n(nT)) = \mathcal{O}(n^2T)$$

.

Dunn Index (DI): For each cluster, we must find the largest distance between any two points within the cluster and the smallest distance between a point in the cluster and outside of the cluster. This involves sorting at worst all pre-computed pairwise distances of which there are $\frac{n(n-1)}{2}$, giving complexity $\mathcal{O}(Tn^2 \log(n))$. As we iterate through the sorted values, we track the highest intra-cluster and lowest inter-cluster distances and update if we find a value that improves either metric. In total, this yields complexity

$$cp_{DI} = \mathcal{O}(Tn^2 \log(n)) = \mathcal{O}(Tn^2 \log(n))$$

.

We now move on to the calculation of the algorithm's complexity in each iteration. Once an initial tree is constructed, each inner iteration of ICOT's local search consists of identifying the best potential split change at a given node. For each of the p features, there are at most $n-1$ potential split thresholds (if all observations are in this node). At each of these thresholds, we must (1) find the assignment of all points to clusters (i.e. tree leaves), which has complexity $\mathcal{O}(nT)$, where T is the total number of nodes in the tree and (2) calculate the cluster quality criterion cp, either cp_{SM} or cp_{DI}. Thus, the inner iteration has complexity $\mathcal{O}(np(nT + cp))$. We must repeat this for each leaf, which adds a factor of T.

Ultimately, one iteration of ICOT when trained on the Silhouette Metric has worst-case complexity:

$$\mathcal{O}(npT(nT + n^2T)) = \mathcal{O}(n^2pT^2 + n^3pT^2) = \mathcal{O}(n^2pT^2 + n^3pT^2)$$

When optimizing the Dunn Index, ICOT's complexity is:

$$\mathcal{O}(npT(nT + n^2T\log(n))) = \mathcal{O}(n^2pT^2 + n^3pT^2\log(n))$$

Both of these results demonstrate that each iteration of ICOT is highly sensitive to scaling with respect to n, with a higher cost when training on the Dunn Index (by a factor of $\log(n)$. Through the geometric search in Section 3.3.3, we are able to reduce the number of splits considered by a constant factor; with a threshold of 0.99, rather than considering np splits, we only consider $0.01 * np$ splits. Additionally, the warm-starts explained in Section 3.3.3 provide higher quality starting solutions which reduces the number of iterations required to reach convergence and thus reduces runtime. This is demonstrated empirically in Section 3.6. Finally, the sub-sampling method introduced in Section 3.3.3 allows us to leverage ICOT for arbitrarily large problems; Section 3.6 also shows empirical evidence that the resultant trees still generalize well to the larger datasets despite only being trained on a subset.

3.4 Experiments based on Synthetic Datasets

In this section, we present results of ICOT across various synthetic datasets. We use these experiments to assess the quality of the algorithm's solution on both validation criteria. We compare ICOT to other popular clustering alternatives in terms of their ability to recover high-quality clustering assignments when training on both the Silhouette Metric and Dunn Index. We also examine the tradeoff between the two metric scores when training on one and evaluating on the other.

3.4.1 Experimental Setup

We evaluated ICOT on the FCPS datasets [337], a standard set of synthetic datasets for unsupervised learning evaluation. These datasets have ground truth cluster labels, which allow for an objective comparison of cluster quality. Our experiments only consider nine of the 10 FCPS datasets, as the tenth contains no true clusters and thus does not offer insight into clustering algorithms.

The ICOT experiments use the "fully scaled" version of the algorithm, with a K-means warm start and a geometric threshold of 0.99. We left the minimum bucket size at its default value (1 observation) and restricted the maximum depth of the tree to depth 3. We left the α parameter at its default value. We ran 100 random restarts of the algorithm in each experiment.

We consider six alternative clustering algorithms which span a range of methodological approaches and interpretations. The following methods are compared:

1. Optimal Classification Trees Hybrid Method (`OCT`): A two-step K-means and OCT hybrid approach, in which K-means clusters serve as class labels for a supervised multi-class classification problem. Each observation is assigned a label based on the predicted class of its leaf. `OCT` is implemented using the InterpretableAI package in Julia [27, 34].

2. K-`means++`: We run K-means with a K-means++ initialization, which was introduced

by [13] and has been shown to improve upon a standard K-means implementation. K-`means++` has been incorporated in the `ClusterR` R package [243]. We run the method with 100 random restarts and a maximum of 100 clustering iterations.

3. Hierarchical Clustering (`Hclust`): Hierarchical clustering is the most popular agglomerative clustering method. It combines individual points into clusters using a linkage measure until all points end up in a single cluster, returning a single dendogram that exhaustively links all individual points [162]. While this is a tree-based method, it does not have binary splits and cannot be explicitly represented as a function of the features. `Hclust` is implemented in R using average linkage.

4. Gaussian Mixture Models (`GMM`): `GMM` assigns observations to clusters characterized by Gaussian distributions. The algorithm uses expectation-maximization (EM) to find the parameters for each of K Gaussian distributions, each representing a cluster [162]. This approach has a key advantage of accounting for cluster variance in assignment, which is a deficiency of traditional methods such as K-means. For each observation, this method returns a soft-assignment, which gives a probability of belonging to each cluster. To make this assignment amenable to our quantitative comparison which requires an explicit assignment, we assign observations to their most likely cluster. `GMM` is implemented in the `ClusterR` R package [243]. We run the method with 20 EM and K-means iterations and confirmed that the results stabilize by this point. We compute observation distances using Euclidean distance.

5. Density-based Spatial Clustering of Applications with Noise (`DBSCAN`): `DBSCAN` is a popular method that constructs clusters based on the highest density regions of a dataset [115]. `DBSCAN` does not return a complete assignment; outliers in low-density areas are left out of any clusters. While this exclusion approach makes the method robust to outliers, it complicates quantitative evaluation. To allow for a fair comparison on the internal validation metrics, we assign each outlier point to the most common cluster of its five nearest neighbors. If all neighbors are also unassigned, we assign the

point to its own cluster. This method is implemented in the `DBSCAN` package in R [153], with additional post-processing to complete the outlier assignment.

6. Predictive Clustering Trees (`PCT`): Predictive clustering trees build recursive binary decision trees for clustering tasks [50]. The methodology is implemented in Java through the `Clus` package. We adopt the default "VarianceReduction" splitting heuristic.

We are unable to present synthetic comparisons to other recent work in interpretable clustering, such as CUBT, as there are no available implementations of the algorithms. We present results of ICOT against the CUBT experiments presented by [131] in Section 3.5.3.

We run all of the comparison methods on normalized data. ICOT normalizes the distance matrix within the algorithm, and we input a normalized dataset into the other comparison method functions. For each of the comparison methods, we tune key parameters to optimize the Silhouette Metric (or Dunn Index). In K-`means++`, `Hclust`, and `GMM`, we tune the number of clusters $K \in [2, 10]$. `DBSCAN` does not have an explicit K parameter, but the ϵ parameter informs the neighborhood size when constructing clusters; larger ϵ values generally translate to larger clusters (and lower K). We tune $\epsilon \in [0.1, 0.11, 0.12 \ldots, 1.0]$. Finally, `PCT` matches our methodology most closely and does not require an explicit cluster number (K) or density threshold (ϵ); for this algorithm, we simply tune the maximum depth from 1 to 3. In all cases, we select the parameter value that yields the best internal validation score on the metric of interest.

In the following experiments, all results are averaged over five experiments per algorithm and parameter combination. All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

3.4.2 Solution Quality

In these experiments, we look to assess various clustering methods in terms of their recovery of high-quality solutions, as measured by both the Silhouette Metric and the Dunn Index. We additionally investigate the performance of the "true" cluster labels on both of these criteria.

Tables 3.1 and 3.2 show the results of these methods along with the true FCPS labels, evaluated with both the Silhouette Metric and Dunn Index.

Data	(N,P)	ICOT	OCT	K-means++	Hclust	GMM	DBSCAN	PCT	Truth
Atom	(800,2)	0.503	0.433	0.611*	0.593	0.565	0.540	0.516	0.311
Chainlink	(1000,2)	0.396	0.28	0.479	0.496*	0.409	0.357	0.312	0.158
EngyTime	(4096,2)	0.573*	0.4	0.439	0.379	0.433	0.450	0.377	0.398
Hepta	(212,3)	0.453	0.332	0.702*	0.702*	0.608	0.702*	0.368	0.702*
Lsun	(400,2)	0.549	0.534	0.569*	0.554	0.537	0.439	0.564	0.439
Target	(770,2)	0.629*	0.409	0.593	0.619	0.578	0.533	0.516	0.295
Tetra	(400,3)	0.504*	0.266	0.504*	0.504*	0.504*	0.504*	0.307	0.504*
TwoDiamonds	(800,2)	0.486*	0.486*	0.486*	0.485	0.412	0.266	0.486*	0.486*
WingNut	(1070,2)	0.422	0.393	0.426*	0.418	0.407	0.384	0.422	0.384
Count Best/Tie		4	1	6	3	1	2	1	3
Average Score		0.502	0.393	0.534	0.528	0.495	0.464	0.430	0.409
Std. Dev Score		0.074	0.089	0.091	0.101	0.081	0.126	0.095	0.153

Table 3.1: Comparison of methods across the FCPS datasets, when trained and evaluated on the Silhouette Metric.

The asterisks indicate the best score across all algorithms for each criterion.

Data	(N,P)	ICOT	OCT	K-means++	Hclust	GMM	DBSCAN	PCT	Truth
Atom	(800,2)	0.137	0.035	0.052	0.097	0.048	0.371*	0.064	0.371*
Chainlink	(1000,2)	0.028	0.013	0.038	0.037	0.016	0.265*	0.018	0.265*
EngyTime	(4096,2)	0.064*	0.002	0.005	0.014	0.004	0.029	0.002	0.000
Hepta	(212,3)	0.357	0.162	1.080*	1.080*	0.482	1.080*	0.293	1.080*
Lsun	(400,2)	0.077	0.027	0.056	0.071	0.117*	0.117*	0.026	0.117*
Target	(770,2)	0.550*	0.011	0.029	0.550*	0.113	0.117	0.013	0.253
Tetra	(400,3)	0.200*	0.044	0.200*	0.200*	0.200*	0.200*	0.046	0.200*
TwoDiamonds	(800,2)	0.044	0.022	0.031	0.049*	0.021	0.030	0.022	0.022
WingNut	(1070,2)	0.063*	0.020	0.026	0.036	0.016	0.063*	0.063*	0.063*
Count Best/Tie		4	0	2	4	2	6	1	6
Average Score		0.169	0.037	0.169	0.237	0.113	0.253	0.061	0.264
Std. Dev Score		0.176	0.048	0.347	0.358	0.153	0.330	0.090	0.330

Table 3.2: Comparison of methods across the FCPS datasets, when trained and evaluated on the Dunn Index

The asterisks indicate the best score across all algorithms for each criterion.

ICOT dominates the two-step supervised learning method in all cases for both metrics, offering an average Silhouette Metric improvement of 27.8% and Dunn Index improvement of 352.7% over OCT. This demonstrates the advantage of building clusters directly through a tree-based approach rather than using a hybrid supervised learning method that applies a tree to cluster labels *a posteriori*.

ICOT matches or outperforms the best alternative clustering method in 4/9 cases with both the Silhouette Metric and with the Dunn Index. ICOT ties or beats K-means++ in 7/9 cases on the Dunn Index and 4/9 on the Silhouette Metric, attesting to its competitiveness against the most widely-used clustering technique. We also note that when measured against our most interpretable alternative, PCT, ICOT ties or wins in all cases on the Dunn Index and 7/9 on the Silhouette Metric.

When considering performance by the ranked wins/ties of each method, K-means++ is the best method for the Silhouette Metric and DBSCAN is the best method for the Dunn Index. No method dominates ICOT in the win/tie ranking; namely, there is no method that performs better on both the Silhouette Metric and Dunn Index. When looking at the average score across all nine datasets, Hclust is the only method to dominate ICOT on both training metrics. However, we note that Hclust also has a significantly higher standard deviation on both metrics, indicating a lack of consistency in solution recovery quality.

Our method is weakest when the underlying clusters are non-separable with parallel splits, since ICOT places hard constraints on an observation's cluster membership based on splits in feature values. In these cases, such as with the Hepta dataset, ICOT is unable to recover the true structure. The flexibility offered by alternative methods is advantageous in these cases. Overall, our results demonstrate that despite the highly constrained setting that we impose on the solution structure, we are still able to perform competitively with far less constrained (and less interpretable) methods.

Cluster quality evaluation is highly dependent on the chosen metric; the ground truth assignment is only the "best" method in 3/9 cases with the Silhouette Metric and 6/9 cases with the Dunn Index. ICOT identifies strictly "better" clusters than the ground truth in 6/9 cases for the Silhouette Metric and 3/9 cases for the Dunn Index, as measured by their scores on the respective metrics. This phenomenon raises the broader question of how to assess cluster quality, as recovering known labels in synthetic data does not necessarily translate to meaningful cluster assignments.

Sensitivity to Training Criterion Choice

Table 3.3 shows the ICOT scores on the FCPS datasets as measured by each validation criterion, broken down by training loss function. The values refer to the average score across all nine datasets. As expected, both metrics have their best performance when they are used as the training criterion to optimize for ICOT. The choice to train on the Silhouette Metric results in a 12.4% loss in Dunn Index score as compared to when training on the Dunn Index. Similarly, training originally on the Dunn Index results in a loss of 15.8% in the Silhouette Metric. This quantifies the sensitivity to the choice of training criterion. Both metrics incur a cost in terms of performance loss on other internal validation criteria, with a slightly lower loss on the Dunn Index.

Training Criterion	**Silhouette Metric**	**Dunn Index**
Silhouette Metric	0.475	0.149
Dunn Index	0.416	0.177

Table 3.3: Comparison of internal validation scores by choice of training criterion in the ICOT algorithm.

3.5 Experiments based on Real-World Datasets

In this section, we present results for two real-world examples. We address two important questions often encountered in practice and demonstrate the value of clustering in their analysis; interpretability and performance on internal validation criteria. We illustrate models produced by ICOT, `OCT`, K-`means++`, `Hclust`, `GMM`, `DBSCAN`, `PCT`, and the CUBT algorithm. We also consider the impact of tuning key user-defined parameters on the ICOT model. Section 3.5.2 outlines a patient similarity case study utilizing data from the well-known FHS. In these models we consider results across several minimum bucket sizes which offer different levels of granularity in the final output. We also experiment with various α parameters, allowing us to control the weight of numerical vs. categorical features in the distance matrix.

Section 3.5.3 focuses on grouping economic profiles of European countries during the Cold War using only tree-based unsupervised learning techniques.

3.5.1 Experimental Setup

We adopted a similar experimental setup to the one described in Section 3.4.1 for the synthetic experiments. In particular, the ICOT experiments use the "fully scaled" version of the algorithm, with a K-means warm start and a geometric threshold of 0.99. We ran 100 random restarts of the algorithm in each experiment. The α and minimum bucket parameters are varied as part of the experiments. We ran all of the experiments on normalized data, which is particularly relevant in this setting where features vary greatly in magnitude.

We consider the same six alternative clustering algorithms: `OCT`, K`-means++`, `Hclust`, `GMM`, `DBSCAN`, and `PCT`. The latter four methods cannot integrate both categorical and numerical features, so we updated the feature space to one-hot encode the categorical variables as binary features. We used the same fixed algorithm parameters for all methods as outlined in Section 3.4.1. We tuned the K parameter over the range of 2 to 10 clusters for all methods other than `DBSCAN`. We tuned $\epsilon \in [1, 5]$ for `DBSCAN`. All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

3.5.2 Patient Similarity for The Framingham Heart Study

Patient similarity is the concept of identifying groups of individuals with comparable health profiles from their EHR, often with the goal of assessing treatment receptivity and outcomes. The goal is to cluster patients in compact groups without any particular outcome of interest and to study the health progression for those individuals over time. Clustering methods have been particularly popular in this application as they do not require an independent covariate in model creation.

We provide an illustration of our method using data from the Offspring Cohort from the FHS, a large-scale longitudinal clinical study. It started in 1948 with the goal of observing

a large population of health adults over time to better understand cardiovascular disease risk factors. Over 80 variables were collected for 5,209 people over the course of more than 40 years. The FHS is arguably one of the most influential longitudinal studies in the field of cardiovascular and cerebrovascular research. This data has now been used in more than 2,400 studies and is considered one of the top 10 cardiology advances of the twentieth century alongside the electrocardiogram and open-heart surgery [209].

Our dataset consists of 1,200 observations from distinct participants of the Offspring Cohort and 11 covariates (age, gender, presence of diabetes, levels of High-Density Lipoprotein (HDL), BMI status, Blood Pressure (BP) status, blood glucose levels, hematocrit levels, history of myocardial infarction, history of stroke, and current smoking habits) [209, 122]. We explore how the ICOT model is impacted as we vary the α parameter and the minimum bucket parameter, N_C. Subsequently, we compare the results of ICOT with other clustering methods in terms of interpretability and quantitative performance on the validation criteria.

The Effect of the α Parameter

In this set of experiments, we focus on the impact of the α parameter on the creation of the ICOT model. The FHS dataset contains mixed numerical and categorical attributes and thus the determination of this parameter clearly affects the feature selection process during tree construction as well as the final number of clusters. We fix the minimum bucket parameter, $N_C = 50$, requiring at least 50 patients in each cluster to ensure that groups are not skewed by outliers in the data.

Figure 3.4 shows the model output when $\alpha = 0.3$. The number of observations in each group is indicated by the numbers in the leaves. When the distance matrix places 70% weight on categorical features, the algorithm partitions the feature space based only on those. As a result, only BP status and gender appear as splits in the tree. ICOT identifies eight groups of patients: (1) 100 women with Elevated BP; (2) 175 men with Elevated BP; (3) 96 women with Hypertensive Status I; (4) 163 women with Hypertensive Status II; (5) 163 men with Hypertensive Status I; (6) 172 men with Hypertensive Status II; (7) 135 women with normal

BP; (8) 196 men with normal BP.

When $\alpha = 0.6$ the output model contains variables from both types of data, balancing better the numerical and categorical feature space. Due to the distance metric re-weighting, the new model is now able to incorporate both numerical and categorical features, yielding intuitive groups of participants by cardiovascular risk. Figure 3.5 illustrates the final tree with five split nodes and six clusters. Given these parameters, ICOT distinguishes between female and male participants in the presence or absence of diabetes. Moreover, it highlights the importance of smoking solely for the diabetic subgroup.

Finally, when $\alpha = 0.9$, ICOT only distinguishes the FHS population based on numeric features such as smoking and diabetes. These results highlight the importance of the algorithm tuning process when leveraging data with mixed features. In the absence of a ground truth, the decision maker is called to select the most appropriate model depending on the application or a potential downstream predictive task. The ability to directly parametrize the distance matrix provides the user with higher flexibility and clarity during the model development process.

The effect of the Minimum Bucket Parameter

In these experiments, we set $\alpha = 0.6$ to balance the distance between numerical and categorical features and we vary the minimum number of observations required to form a distinct cluster. Figures 3.5, 3.7, and 3.8 show the models produced by the algorithm for different values of the minimum bucket, N_C, when training on the Silhouette Metric. Note that varying this constraint directly affects the end model, changing the structure of the final tree. Even though our empirical results may suggest that there is a monotonic relation between the size of the minimum bucket and the number of clusters identified, this assumption is not necessarily a general rule.

Comparing between Figures 3.5 and 3.7, we see that the output is stable given the minimum bucket restrictions. Both models share the same features in the splits. In the latter model, splits that already had at least 100 members in both leaves (the leftmost two clusters)

remained intact and new ones were created in order to closely match the tree with $N_C = 50$. When we increase the minimum sample size to 200 participants, the resulting model only separates the population by gender.

Notice that across all the experiments presented, three variables appear to bear the highest importance in the clustering task: smoking habits, diabetic status, and gender. The results appeared to be stable in the feature selection process, confirming the intuition behind the effect of both the minimum bucket and α. ICOT's interpretable structure allowed us to specify the key differentiating characteristics between the participants and contextualize them in the medical setting.

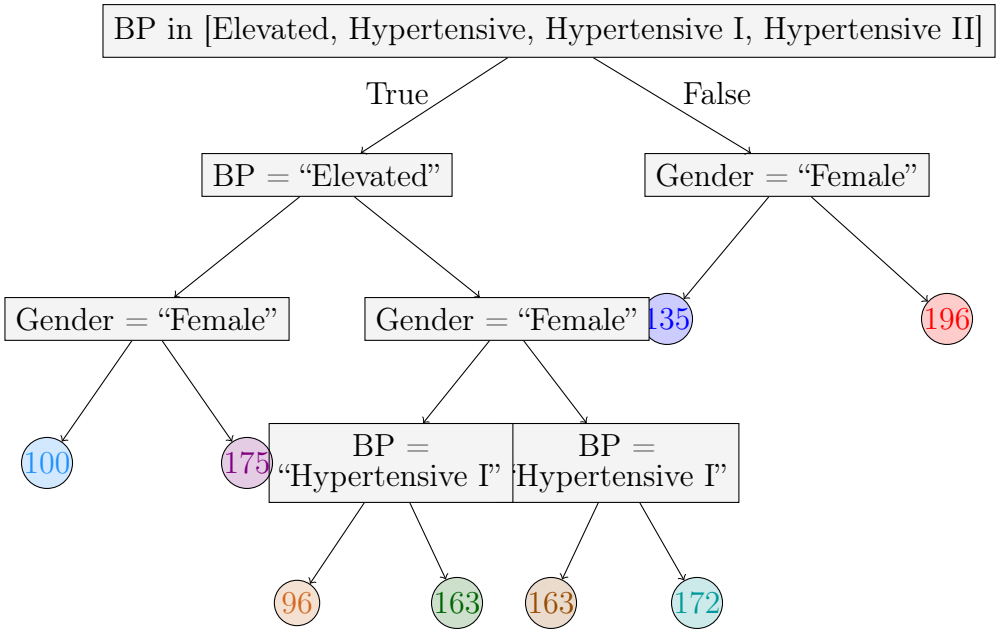


Figure 3.4: ICOT tree for minimum bucket = 50 and $\alpha = 0.3$.

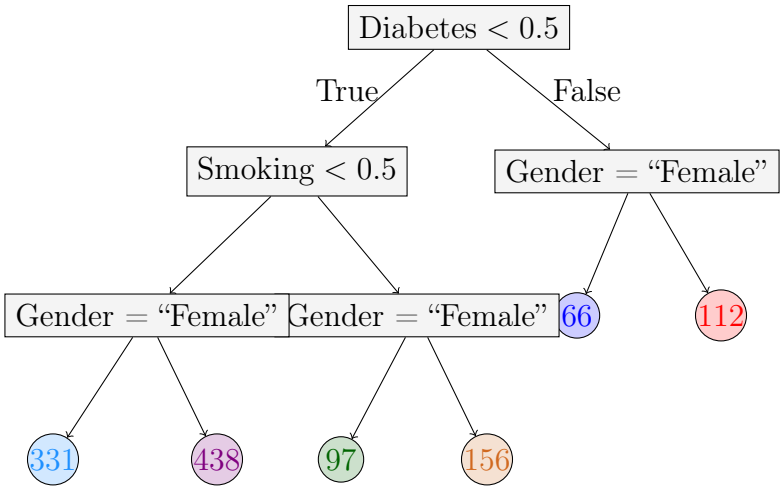


Figure 3.5: ICOT tree for minimum bucket = 50 and $\alpha = 0.6$.

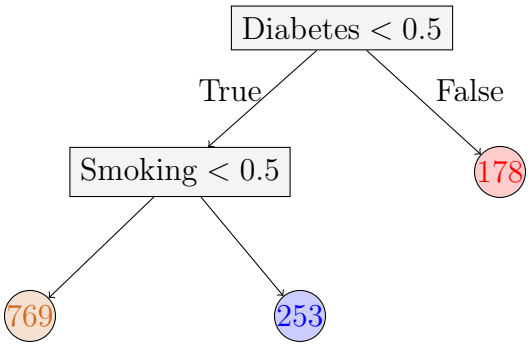


Figure 3.6: ICOT tree for minimum bucket = 50 and $\alpha = 0.9$.

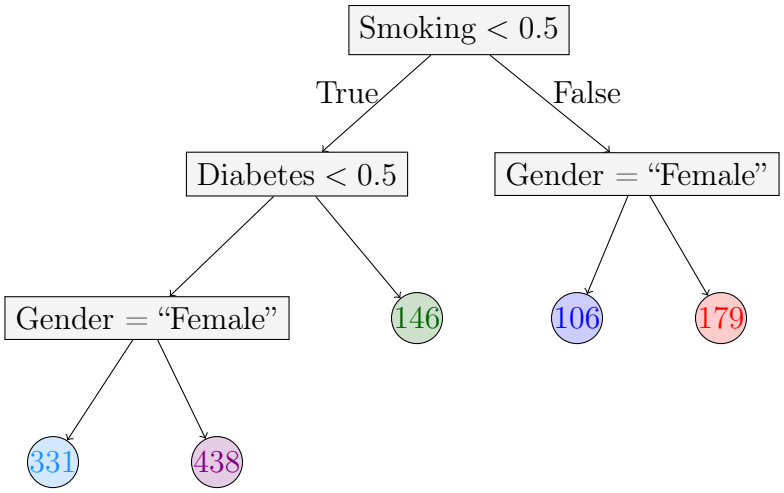


Figure 3.7: ICOT tree for minimum bucket = 100 and $\alpha = 0.6$.

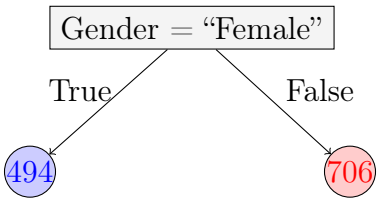


Figure 3.8: ICOT tree for minimum bucket = 200 and $\alpha = 0.6$.

Results on Interpretability

In this section, we compare the interpretability of partitions from different clustering algorithms. For tree based approaches, such as the two step OCT method and `PCT`, we present the final model. For the rest of the algorithms, we outline the centroids of each cluster. Since these methods also do not allow us to directly control the minimum number of observations per cluster, we present the results of each algorithm for the number of clusters that maximizes the Silhouette Metric. Here, we present detailed results for the K-`means++`.

Figures 3.4-3.8 demonstrate different ICOT models when we vary the algorithm's hyperparameters. Note that the trees provide meaningful categorizations that clinicians frequently use and think about in stratifying patient risk. Elevated BP measurements, gender, smoking are all commonly used categories that determine future health trajectories, such as the

risk of cardiovascular events or potential interventions for managing chronic diseases (i.e., blood pressure). The role of these variables has been widely recognized in medical literature [184, 356, 252, 116].

Variable Names	Cluster 1		Cluster 2		Cluster 3	
	Mean	Std. Deviation	Mean	Std. Deviation	Mean	Std. Deviation
Gender: female	0.367	0.485	0.376	0.485	0.487	0.5
Gender: male	0.633	0.485	0.624	0.485	0.513	0.5
Diabetes	0.922	0.269	0.054	0.227	0.142	0.35
Smoking	0.2	0.402	0.249	0.433	0.226	0.419
Age	64	7.114	61.102	9.976	65.335	9.156
HDL	39.497	12.679	46.681	14.592	46.547	14.663
Blood Glucose Levels	198.901	39.916	98.792	10.908	103.898	15.428
Myocardial Infarction	0.333	0.519	0.337	0.632	0.239	0.518
Hematocrit Levels	44.929	3.163	43.942	3.866	43.409	3.634
Blood Pressure Status: Elevated	0.211	0.41	0.358	0.48	0	0
Blood Pressure Status: Hypertensive Crisis	0.044	0.207	0	0	0.066	0.249
Blood Pressure Status: Hypertensive Status 1	0.256	0.439	0.239	0.427	0.165	0.372
Blood Pressure Status: Hypertensive Status 2	0.356	0.481	0	0	0.769	0.422
Blood Pressure Status: Normal	0.133	0.342	0.404	0.491	0	0
BMI Category: Normal	0.1	0.302	0.263	0.44	0.246	0.431
BMI Category: Obese	0.489	0.503	0.296	0.457	0.305	0.461
BMI Category: Overweight	0.411	0.495	0.44	0.497	0.447	0.498
BMI Category: Underweight	0	0	0.001	0.037	0.003	0.05
Number of Observations	90		716		395	

Table 3.4: The centroid mean, standard deviation values, and number of observations for all identified clusters from the K-means++ algorithm on the one-hot encoded dataset.

Table 3.4 shows the covariate values of the cluster centroids created by the K-means++ algorithm. Notice that there is no clear distinction of features that characterize each cluster. For the categorical ones, the centroid value depends on the relative frequency of the classes in the particular covariate and not only on its predominance in the cluster. For example, the fact that the Smoking value for Centroid 1 is equal to 0.2 does not provide deep insights in the smoking habits of the participants in that group. There is a similar proportion of smokers in this cluster compared to Clusters 2 and 3. It is difficult to provide intuitive labels for the groups with clinical implications by only studying Table 3.4. Furthermore, analyzing

the centroid means and standard deviations to gain intuition into the distinctive attributes and spread of each cluster becomes increasingly harder as the number of features increases. Relative ranking of the centroid values could be used in the FHS case, where $p = 18$ (after one-hot encoding) and the number of clusters is small. In a high dimensional dataset, delving into such a table would be practically impossible.

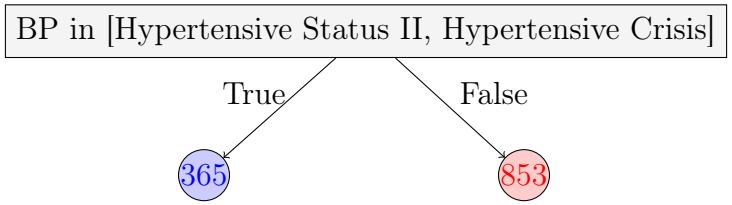


Figure 3.9: Two-step OCT tree, optimized with respect to the Silhouette Metric.

Figure 3.9 shows the result of the hybrid OCT tree. The model contains just one split, resulting in two clusters providing limited insights regarding the data. In this setting, changing the minimum bucket did not affect the final solution. Figure 3.10 shows the final PCT tree. This method proposes a deeper tree involving four features: Gender, Diabetes status, BMI status, and SBP. It suggests that diabetes status is a differentiator only in obese patients (BMI above 30). It also suggests that the relevant SBP threshold is higher for "less healthy" patients, namely those who are diabetic or have higher BMI.

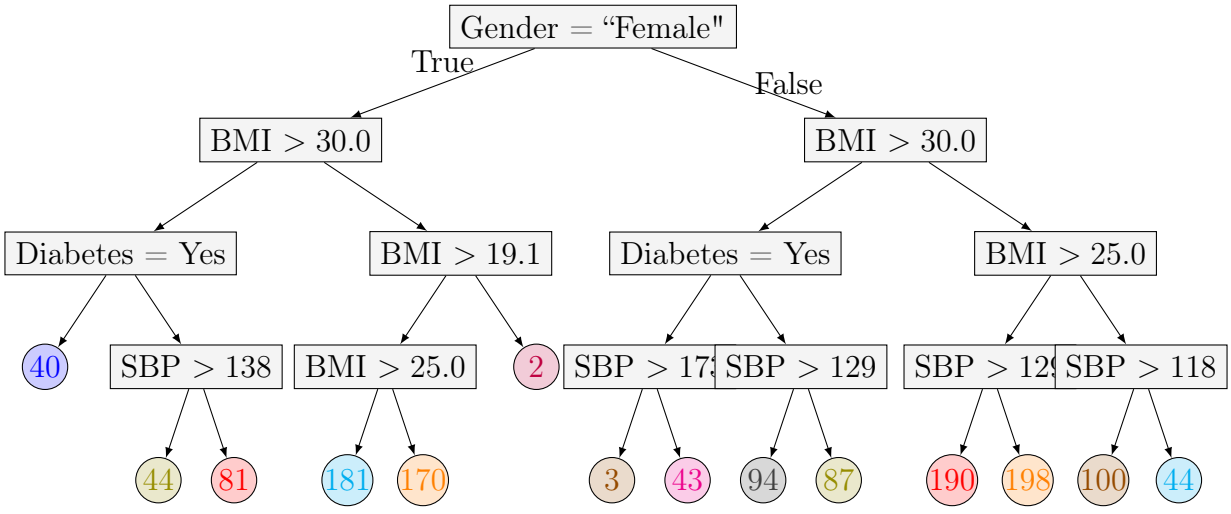


Figure 3.10: PCT Tree for FHS patients.

Results on Quantitative Performance

Although interpretability is our primary objective in cluster development, we also want to ensure that our resultant groupings are reasonable from the perspective of the internal validation criteria which provide a quantitative evaluation. Table 3.5 shows the metric scores obtained for both the Silhouette Metric and the Dunn Index. For each metric, we use the respective criterion to cross-validate and find the optimal number of clusters. We then report the score for the entire population.

ICOT dominates all competing algorithms in the Dunn Index (0.509) and has the second to best performance in the Silhouette Metric (0.296) after `DBSCAN` (0.511). In particular, we note that it has an advantage over `PCT` in both metrics, consistent with our findings in the synthetic experiments. Overall these results suggest that ICOT's advantage in interpretability does not come at the expense of identifying well-separated and compact clusters. The gains over OCT also attest to the value of ICOT's ability to train directly on the cluster quality criterion over simply applying a two-step method where K-means clusters are used as class labels for a supervised problem.

Metric	**ICOT**	**OCT**	**K-means++**	**Hclust**	**GMM**	**DBSCAN**	**PCT**
Silhouette Metric	0.296	0.131	0.264	0.270	0.224	0.511	0.249
Dunn Index	0.561	0.256	0.150	0.469	0.503	0.448	0.503

Table 3.5: The validation criteria results for ICOT, K-`means++`, `Hclust`, `GMM`, `DBSCAN`, `PCT` and the two-step hybrid OCT method when trained on each metric.

3.5.3 Economic Profiles of European Countries

In this section we consider European countries by their employment statistics during the Cold War to develop groupings of similar economic profiles. We present this example to offer a comparison to the CUBT algorithm [131] as this is the primary real-world experiment offered in their work.

Our dataset [193] provides the breakdown of where citizens were employed in 1979 across

major industry sectors: agriculture (Agr), mining (Min), manufacturing (Man), power supplies services (PS), construction (Con), service industries (SI), finance (Fin), social and personal services (SPS), and transportation and communication (TC). Thus our feature space includes nine covariates ($p = 9$) observed for 26 distinct European countries ($n = 26$).

Results on Interpretability: ICOT

We trained a clustering tree using the Silhouette Metric, the default α parameter, and a minimum bucket size of 3 to prevent individual outlier countries from dominating the tree in a single split. The final tree is shown in Figure 3.11, and the resulting groupings are shown in Table 3.6.

ICOT's chosen partition is highly intuitive given the economic and political climate of the Cold War. With the exception of Yugoslavia, all Eastern Bloc countries are placed in Cluster 1 due to their particularly low percentage of workers in the financial sector. This split reflects the broader political setting for those countries that were under a Communist regime. Greece, Turkey and Yugoslavia are grouped together due to their notably high agricultural sector employment. They are also located in the same geographical region and thus their economy similarity is justified. The rest of the countries form Cluster 2, which is composed of all the Western European countries.

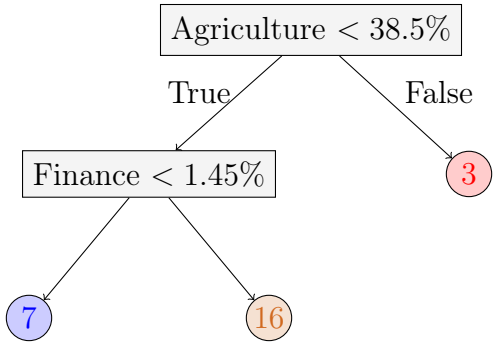


Figure 3.11: Visualization of the ICOT tree for the European Jobs dataset

Cluster 1	Cluster 2		Cluster 3
Bulgaria	Austria	Belgium	Greece
Czechoslovakia	Denmark	Finland	Turkey
E. Germany	France	Ireland	Yugoslavia
Hungary	Italy	Luxembourg	
Poland	Netherlands	Norway	
Romania	Portugal	Spain	
USSR	Sweden	Switzerland	
	United Kingdom	W. Germany	

Table 3.6: European country clusters from the ICOT algorithm

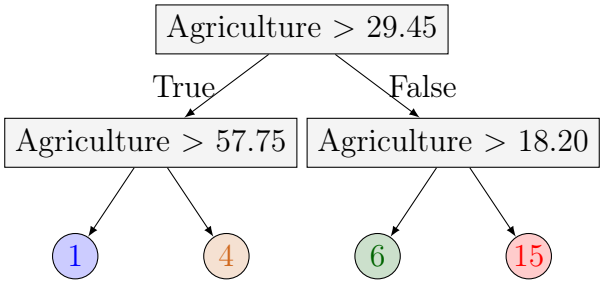


Figure 3.12: CUBT tree with four clusters

Results on Interpretability: CUBT

[131] provide two alternative clustering partitions using their proposed CUBT algorithm, one with four clusters and the other with five clusters. The resultant tree for $K = 4$ is shown in Figure 3.12 with the groupings listed in Table 3.7. The corresponding results for $K = 5$ are presented in Figure 3.13 and Table 3.8, respectively. Due to inconsistencies between the trees and country groups listed in the paper [131], we report results based on the tree models presented. It is possible to select a minimum bucket size in the CUBT algorithm, but the authors chose to omit it in these experiments, resulting in isolated clusters with single outlier countries. While this provides insight on its own, we chose to enforce a sufficiently large leaf size to make our results more generalizable and insightful for the full set of European countries.

The tree with four clusters splits only on agriculture sector employment through a series of recursive splits, providing less insight into the differentiating characteristics of the countries. The tree with five clusters splits on high agriculture employment first to separate

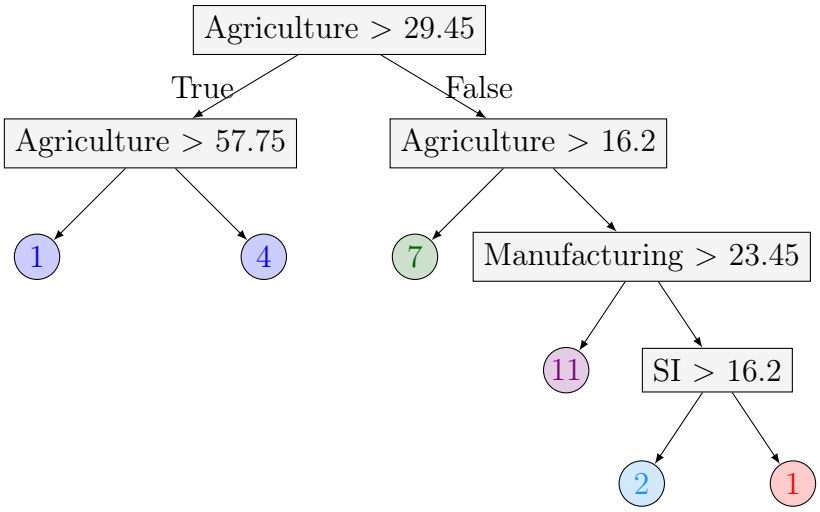


Figure 3.13: CUBT tree with five clusters

out the first two clusters, but then further differentiates the low agriculture countries on both manufacturing and service industry employment. The bulk of the countries fall into the third cluster, which is characterized by a manufacturing-heavy workforce. Note that CUBT allows for cluster re-joining in the algorithm, which results in multiple leaves being assigned to the same cluster (indicated by a single color). Overall, while the CUBT algorithm provides high interpretability as with ICOT, a qualitative analysis of the resulting clusters suggests that there is a slight loss in meaningful cluster separation.

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	
Turkey	Greece	Bulgaria	Austria	Belgium
	Poland	Hungary	Czechoslovakia	Denmark
	Romania	Ireland	E. Germany	Finland
	Yugoslavia	Portugal	France	Italy
		Spain	Luxembourg	Netherlands
		USSR	Norway	Sweden
			Switzerland	United Kingdom
			W. Germany	

Table 3.7: European country clusters from the CUBT algorithm, with $K = 4$

Cluster 1	Cluster 2	Cluster 3		Cluster 4	Cluster 5
Greece	Bulgaria	Austria	Belgium	Netherlands	Denmark
Poland	Czechoslovakia	E. Germany	Finland	Norway	
Romania	Hungary	France	Italy		
Turkey	Ireland	Luxembourg	Sweden		
Yugoslavia	Portugal	Switzerland	United Kingdom		
	Spain	W. Germany			
	USSR				

Table 3.8: European country clusters from the CUBT algorithm, with $K = 5$

Results on the Validation Criteria

The quantitative performance of these models on our two key internal validation criteria are shown in Table 3.9. ICOT obtains significantly better clusters as quantified by both the Dunn Index and Silhouette Metric. We note that ICOT has an advantage in the Silhouette Metric due to the fact that it was trained to optimize this criterion, whereas the CUBT results were trained via a different method. However, the Dunn Index provides a neutral evaluation criterion and shows a preference towards ICOT's results as well.

Metric	ICOT	CUBT ($K = 4$)	CUBT ($K = 5$)
Silhouette Metric	0.344	0.140	0.044
Dunn Index	0.346	0.262	0.259

Table 3.9: Comparison of ICOT (trained on the Silhouette Metric) and the CUBT algorithm on the internal validation criterion

3.6 Scaling Experiments

In this section, we present results regarding the effect of scaling techniques on ICOT with respect to both the quality of the final solutions as well as the degree to which the algorithm is able to scale. In Section 3.6.1, we discuss the impact of algorithm heuristics, such as the K-means warm start and the geometric threshold, using the FCPS suite. We use real-world data from Hubway for testing the scalability and quantitative performance of bootstrapping in Section 3.6.2.

3.6.1 Scaling via Algorithm Heuristics

In this section, we evaluate the impact of implementing the scaling methods described in Section 3.3.3. We first consider how the heuristics affect solution recovery and then examine the runtime reductions that we obtain as we vary the scaling parameters.

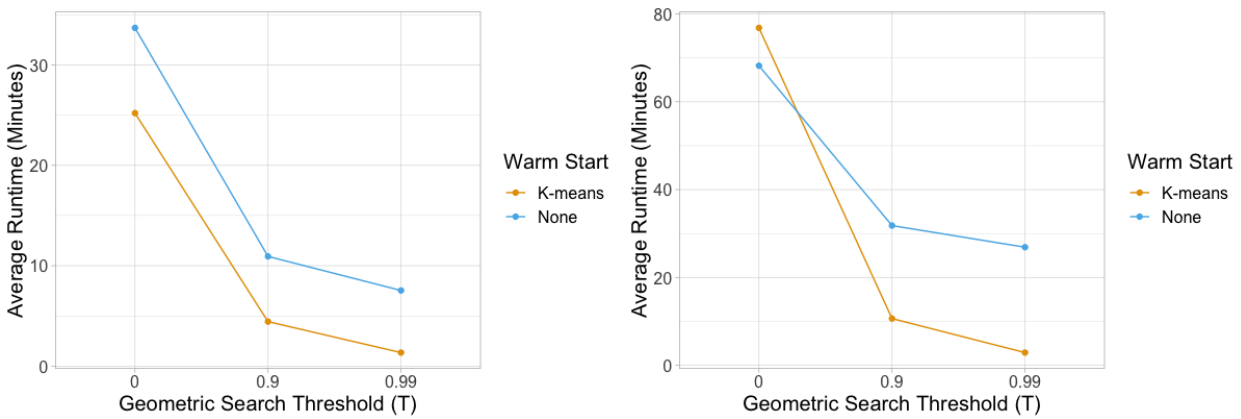
Experimental Setup

We evaluated the impact of our scaling methods on algorithm speed through a comparison of the average runtime across eight datasets in the FCPS suite with various parameters. The ninth dataset (EngyTime) was omitted as the experiment size was intractable on the unscaled method. We ran experiments over restricted geometric search thresholds of $T = 0$ (scan all thresholds), $T = 0.9$ and $T = 0.99$. We also repeated the experiments with and without the K-means warm start. The parameter pair ($T = 0$, no warm start) represents the original "baseline" method, and the pair ($T = 0.99$, K-means warm start) represents the fully scaled method. We ran each dataset and parameter combination across five seeds and present the averaged results.

All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

Scaling Runtimes

The runtimes for the Silhouette Metric and Dunn Index are shown in Figure 3.14. The geometric search alone reduces the runtime by 77.6% (60.6%) at the $T = 0.99$ threshold for the Silhouette Metric (Dunn Index). When combining the geometric search ($T = .99$) with the K-means warm start, our fully scaled method offers a 96.0% (95.7%) reduction in algorithm runtime for Silhouette (Dunn). We observe that the baseline method actually has a slight runtime advantage over the K-means warm start when there is no restriction on the search space ($T = 0$). The apparent shorter runtime with the baseline method at $T = 0$ can be explained by the possibility of getting caught in a locally optimal solution with a naive start, which can lead the algorithm to terminate faster.



(a) Silhouette Metric Runtimes

(b) Dunn Index Runtimes

Figure 3.14: Average runtimes across FCPS datasets with varied scaling parameters for the geometric search threshold (T) and choice to use a warm start.

Due to the speedups from these two scaling techniques, ICOT is able to scale to handle datasets with a number of observations (N) in the thousands and the number of covariates (p) in the hundreds. The scaled algorithm solves within several hours for problems of this magnitude.

High Quality Solution Recovery

The scores of the baseline model and our fully scaled version are shown in Table 3.10. The scaled method yields an average loss of -0.28% over the baseline when trained on the Silhouette Metric, and gives an average improvement of 0.64% with the Dunn Index. Of the eight datasets considered using the Silhouette Metric (Dunn Index), three (five) have identical cluster recovery in both the original and fully scaled experiments; three (two) have a slight loss when using scaling heuristics, and two (one) actually improve with the scaling methods. These results suggest that the scaled ICOT algorithm still yields high quality results.

The differences in the score between the baseline and scaled versions are largely attributable to the warm start rather than the choice of geometric threshold. The score improves in the scaled version when the baseline algorithm was caught in a local optimum, but the K-means warm start enabled it to avoid this. This score improvement offered by the K-means warm starts further supports the use of this heuristic beyond runtime improvements.

	Silhouette Metric			Dunn Index		
Dataset	**Baseline**	**Fully Scaled**	**% Change**	**Baseline**	**Fully Scaled**	**% Change**
Atom	0.521	0.503	-3.45%	0.137	0.137	0.00%
Chainlink	0.391	0.396	1.28%	0.032	0.028	-12.62%
Hepta	0.455	0.453	-0.44%	0.357	0.357	0.00%
Lsun	0.567	0.549	-3.17%	0.117	0.077	-34.10%
Target	0.629	0.629	0.00%	0.362	0.550	51.93%
Tetra	0.504	0.504	0.00%	0.200	0.200	0.00%
TwoDiamonds	0.486	0.486	0.00%	0.044	0.044	0.00%
WingNut	0.406	0.422	3.94%	0.063	0.063	0.00%
Average Score	0.495	0.493	-0.23%	0.164	0.182	0.65%

Table 3.10: Comparison of cluster quality scores with the original vs. fully scaled ICOT versions.

3.6.2 Scaling via Bootstrapping

In Section 3.6.2, we introduce the Hubway dataset, a real-world collection of user ride data from a Boston-based bike sharing program. First, we outline the experimental setup, providing details on the parameters of the method. Subsequently, we explore the effect of the bootstrapping methodology on the quality of the final solution and the algorithm runtime respectively.

The Hubway dataset

In this setting, our goal is to identify similar groups of registered users of the Hubway bike-sharing program [40]. This Boston-based company allows citizens to rent bicycles from any of their 140 stations and ride to any other station in the city. The platform has emerged as a popular form of transportation for daily commuters and leisure riders alike. Our dataset includes 194,301 observations from Hubway trips taken from June 2012 through September 2012. The dataset contains nine mixed numerical and categorical attributes, including the duration of the trip, the age and the gender of the rider, the time period of the ride and whether it took place during the week or the weekend.

This experiment illustrates an application of clustering for market segmentation. This is a strategy that divides a broad target market into smaller groups of similar customers. It can then be used to tailor marketing strategies to individual groups through means such as

promotions or differentiated pricing. Unsupervised learning is often employed for this task since it naturally identifies similar groups within a given dataset.

Experimental Setup

In these experiments, we aim to quantify the benefit of using bootstrapping as a wrapper function over the ICOT algorithm. We explore the effect of three key parameters that might affect both the quality and runtime of the solutions.

1. Sample Size (N): The number of observations included in the training set. Since the Hubway dataset contains 194,301 data points, we sub-sample randomly without replacement to create a sample of size N. We follow the same process to create a different testing set that is used for the evaluation of the validation criterion. We restrict N to numbers that can be efficiently solved by ICOT, $N \in [2500, 5000, 10000]$, to allow us to compare to the algorithm's solutions on the full input data.

2. Size of reduced data (N_r): The number of observations included in each iteration of the bootstrap algorithm. Each sub-sample is randomly created from the training set without replacement, but the iteration samples are constructed independently. Thus, different iterations can contain the same observation. We let $N_{\text{rep}} \in [250, 500]$.

3. Number of repetitions (N_{rep}): The number of iterations of the bootstrapping method. We test the quality and runtime of the final model by letting $N_{\text{rep}} \in [25, 50, 75, 100, 200, 500, 1000]$.

All results presented for ICOT use a version of the algorithm that includes the K-means warm start and a geometric threshold of 0.99. The minimum bucket size is set to one and the maximum depth of the tree to depth four. We assigned to the α parameter its default value. Similarly to the FCPS experiments, we ran 100 random restarts of the algorithm in each round. Results summarize the outcomes of five randomized repetitions of each experiment.

In the following experiments, all results are averaged over 50 experiments per algorithm and parameter combination. All experiments were conducted on two CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 30GB of NUMA enabled memory were used per CPU.

Scaling Performance

The purpose of introducing bootstrapping into the ICOT framework is to extend its application to problems of larger size that the fully scaled version was not able to efficiently manage. Bootstrapping provides a lot of flexibility to the user and thus can be easily adapted to the speed requirements of a specific case study. In this section, our aim is to demonstrate how choices regarding the parameters affect the overall running time and compare the outcomes with and without bootstrapping. Figure 3.15 provides an overview of the results when the algorithm was trained on the Silhouette Metric. We report the log(time) to render the y-scale more comprehensible to the reader, especially for higher instances of N. The average runtime scales linearly with respect to N_{rep} and exponentially to N_r. As we include additional repetitions, the method sequentially runs more iterations of the same "reduced" experiment. However, as we increase the N_r, the runtime scales at the same rate as the original ICOT method. When $N_{\text{rep}} > 500$, bootstrapping starts improving on the original algorithm only for instances of $N > 2500$. Nevertheless, in cases of larger sample size ($N = 10,000$), bootstrapping can achieve the same solution quality ($N_r = 250, N_{\text{rep}} = 500$) in 27.65 minutes instead of 554.693. When $N = 5,000$, the discrepancy is not as high but still considerable, 13.095 and 96.529 minutes respectively.

These results indicate the value of adding bootstrapping into the ICOT framework, as it solves in reasonable time problems of much larger size that otherwise would have been out of the algorithm's scope.

High Quality Solution Recovery

The bootstrapping approach constructs trees on a sub-group of the overall population and thus does not access the full input data. We sought to ensure that the speed-up in runtime would not come at a high toll with respect to solution quality. Thus, we performed a direct comparison of the two methods over the validation criteria for different ranges of the parameters described above. Figure 3.16 provides a results summary for the Silhouette Metric. The shaded region around ICOT indicates the standard deviation of the metric. Similarly,

the error bars illustrate the same measure for each combination of the tuning parameters. As expected, larger sample sizes are positively correlated with the validation score. The graphs show that increasing the number of repetitions can significantly improve the quality of the solution. We notice that for $N_{\text{rep}} > 500$, bootstrapping can achieve equivalent performance to ICOT, with minor losses in some cases. The effect of the N_r parameter is less evident, though, as the results indicate minor discrepancies between $N_r = 250$ and $N_r = 500$. In conclusion, these experiments provide evidence that bootstrapping does not result in a high toll on the quality of suggested feature partitions.

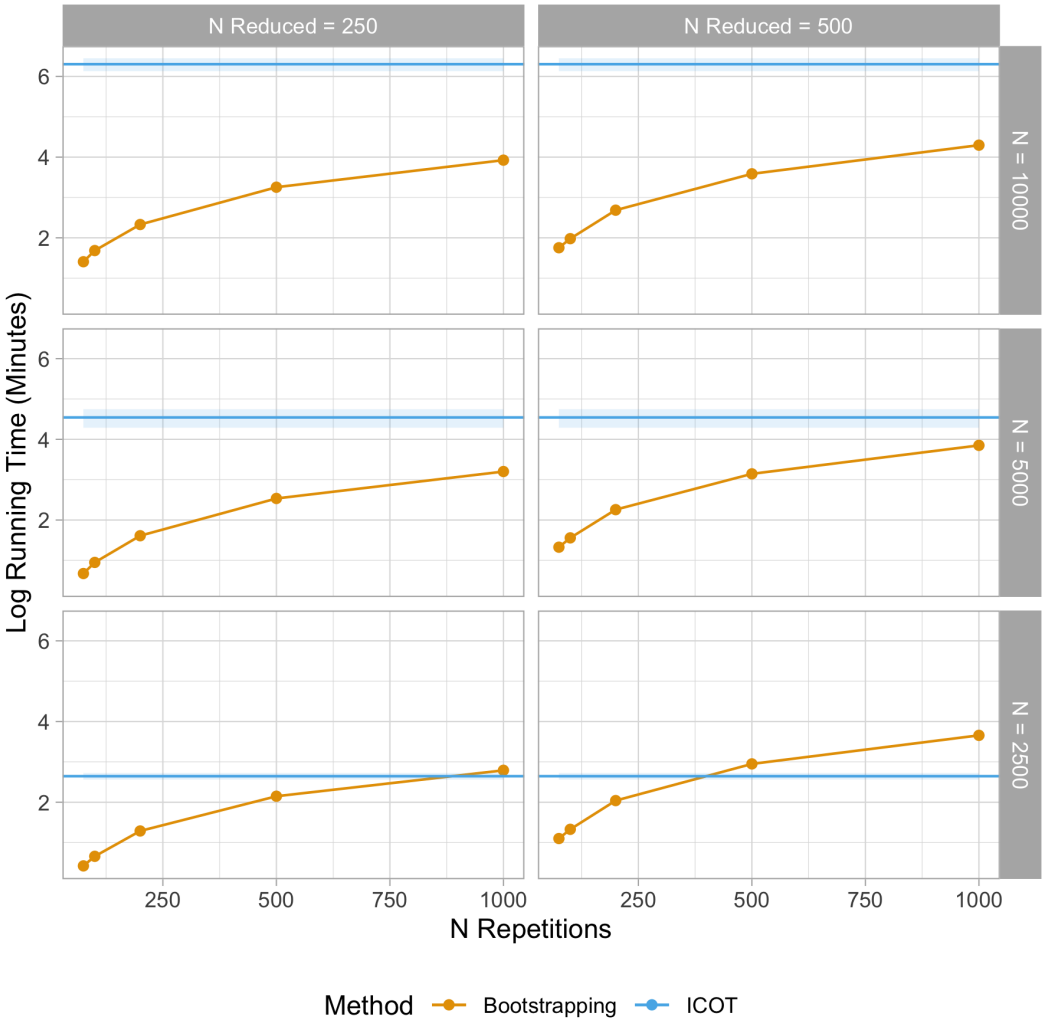


Figure 3.15: Results regarding the impact of bootstrapping on the runtime (Log of Minutes) as the number of repetitions (N_{rep}), sub-sample size (N_r), and sample size (N) change. Both methods were trained on the Silhouette Metric. The error bars express the standard deviation of the metric.

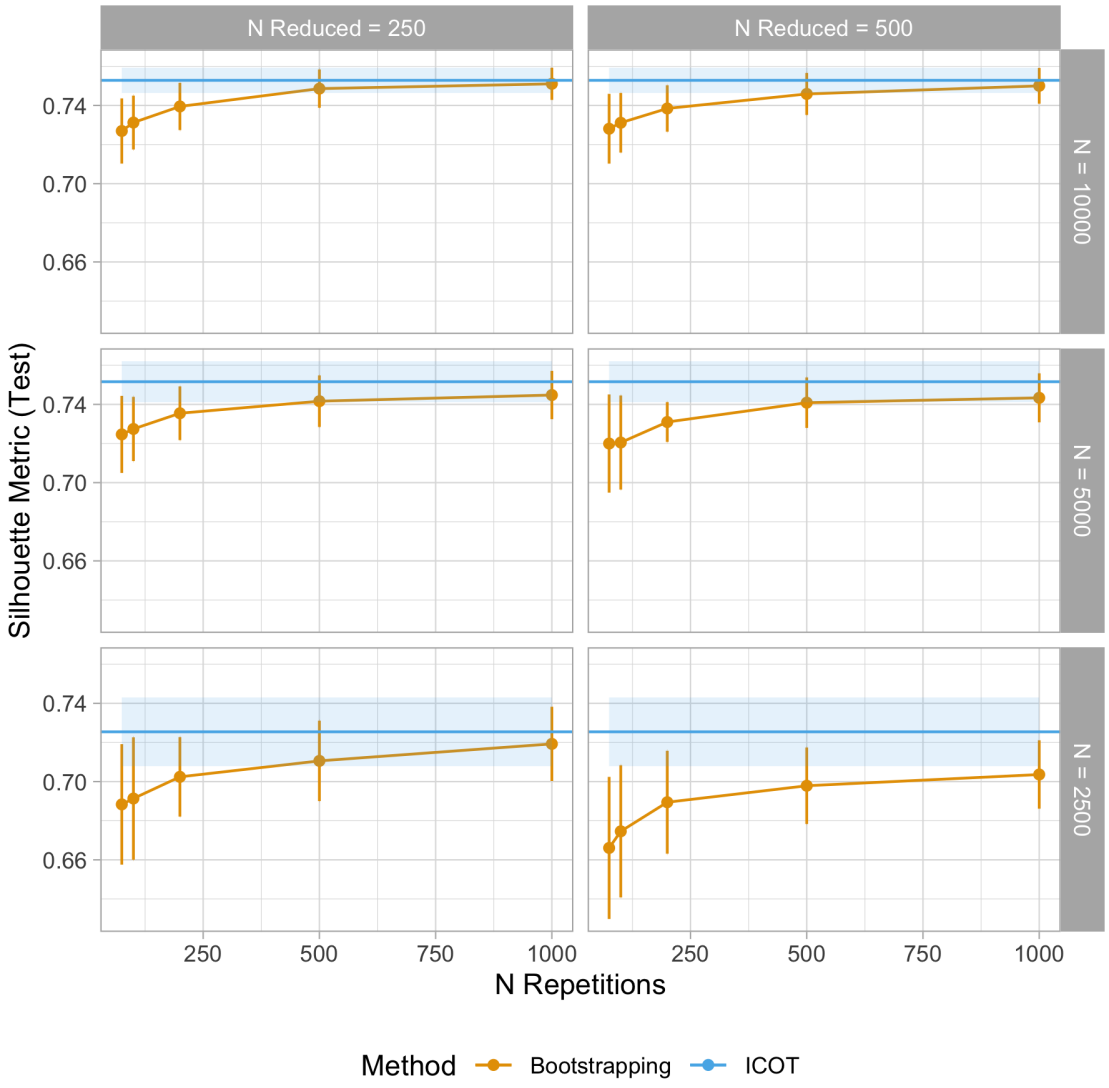


Figure 3.16: Results regarding the impact of bootstrapping on the Silhouette Metric as the number of repetitions (N_{rep}), sub-sample size (N_r), and sample size (N) change. The error bars express the standard deviation of the metric.

3.7 Discussion

ICOT builds trees that provide explicit separations of the data on the original feature set, creating interpretable models with real-world applicability to a wide range of settings. From healthcare to revenue management to macroeconomics, our algorithm can significantly benefit practitioners that may find value in unsupervised learning techniques in their work.

Our empirical results on the FCPS dataset offer insight into ICOT's performance against existing methods, including traditional approaches such as K-means, density-based, and hierarchical algorithms. We also report results with respect to other interpretable methods, including the Predictive Clustering Trees framework and the hybrid two-step supervised approach. Overall, our proposed method is superior to the majority of the algorithms for both validation criteria. Specifically, in Section 3.4, we show that when assessing clusters with the Silhouette Metric, ICOT is the second best method after K-`means++` while on the Dunn Index ICOT is only outperformed by `DBSCAN`. Essentially, our experiments demonstrate that our newly proposed framework is able to achieve comparable performance to the state-of-the-art clustering algorithms while enabling the explicit characterization of cluster membership. We thus accept a slight decrease in the validation criteria for the gain in interpretability, which is critical in many settings.

We also observe significant improvements in ICOT over other interpretable approaches. The relatively poor performance of the two-step OCT approach validates the utility of a method that simultaneously builds clusters and identifies a tree-based structure rather than simply employing existing tree-based methods on clustered data *a posteriori*. Additionally, ICOT offers a considerable advantage over PCT and CUBT, suggesting that our algorithmic approach improves upon on existing interpretable clustering work and offers a novel contribution to the space.

Most clustering methods, including ICOT, identified data partitions with higher cluster quality scores than the true FCPS data labels, highlighting the subjectivity of what constitutes good clusters. We leave the choice of cluster quality metric to the user, since both criterion have their respective merits and perform well in different data contexts. In general, the

Dunn Index excels on well-separated datasets but is not robust to outliers. In contrast, the Silhouette Metric is often better at accounting for mixed densities and identifying meaningful separation in less structured data settings.

The additional scaling experiments on the FCPS dataset demonstrate substantial runtime reductions offered by both the restricted geometric search space and K-means warm start. Overall these empirical results suggest that the scaling methods are successful at significantly decreasing runtime while maintaining high-quality cluster identification. The geometric search heuristic is particularly useful for problems with a high number of observations as it lowers the computational load per node evaluation by a factor of T. We note that despite the efficiency gains offered by our scaling methods, our current implementation of ICOT does not scale beyond 1000s of observations and 100s of covariates. However, using the Hubway dataset we were able to demonstrate that the ICOT algorithm coupled with bootstrapping is able to scale to even hundreds of thousands of observations at a reasonable time without a considerable toll on the solution quality. This functionality broadens the method's applicability to even high-dimensional settings; for example, bootstrapping might be particularly useful when clustering a large company's customer transaction records (n in the millions). This is a case where we would recommend the subsampling approach. A similar technique could be applied for cases where the number of features is very high (p in the 10000s), such as when using genomic profiles for patients. Additionally, variables could be preprocessed to restrict to the most significant subset, either using traditional statistical tests or the variable importance ranking provided in the K-means algorithm output.

Therefore, we believe that ICOT is the best performing alternative for interpretable clustering although computationally more intensive. PCTs are more efficient but in many cases lead to lower quality solutions. Our method has an edge over K-means++ and DBSCAN due to the transparency it offers, although these alternatives sometimes show a slight edge on the Silhouette Metric and the Dunn Index. ICOT is most appropriate in applications where the user values both interpretation of the cluster labels and high performance on clustering metrics, and the efficiency of the algorithm is not a bottleneck. These conditions are generally

true in the exploratory analysis contexts where clustering is most often applied.

Our work's handling of numerical and categorical features offers a contribution beyond the realm of clustering. The issue of mixed-type attributes is considered among specialists as one of the most important challenges in machine learning [268, 362]. The overwhelming majority of state-of-the-art clustering algorithms are restricted to numerical objects, like vectors or metric objects, which does not correspond to datasets usually found in practice. This problem extends more broadly to algorithms that rely on distance computations, such as k-Nearest Neighbors. In contrast, our solution gives a comprehensive answer to this problem by introducing a novel distance metric for the algorithm.

We note that the algorithm's single-variable splits are unable to represent all possible cluster shapes and could potentially cut through clusters. This structure allows us to maintain the direct interpretation of a tree leaf representing a single cluster. In many applications, a simple interpretation of the tree partition is highly valued, which was a key motivation behind this method's development. In order to capture more complex structures, one could consider the possibility of "rejoining" leaves, namely allowing multiple leaves to be considered as a single cluster. Rejoining can occur between two adjacent leaves coming from a single parent node through the local search's consideration of split deletions. However, we do not consider the possibility of joining other leaves. While ICOT does not natively support this, it could easily be incorporated as a post-processing step. After obtaining the final ICOT tree, one can consider the effect of merging different node combinations on the chosen metric.

We finally observe that despite the tree structure of our algorithm output, our model does not obey a hierarchical structure. Namely, truncating the tree to a lower depth does not necessarily represent the optimal clustering solution at this depth. Our CD algorithm allows for nodes to be re-optimized with knowledge of deeper nodes. In contrast, a hierarchical interpretation only holds in cases where the tree grows greedily since the shallow truncated tree cannot be affected by deeper levels.

The application of ICOT to real-world datasets reveals the significant benefit on both interpretability and performance in the unsupervised learning field. The combination of the

OCT mechanism, the employment of established internal validation criteria as well as the systematic handling of mixed numerical and categorical attributes allow ICOT to provide complete partitions of the feature space with actionable insights to practitioners. Moreover, the flexibility of the method to user specific constraints with respect to the minimum bucket size, the maximum depth of the tree and the α parameter render the algorithm particularly amenable to a wide range of applications from various fields.

3.8 Conclusions

In this chapter, we have introduced a new methodology of cluster construction that addresses the issue of cluster interpretability. We propose a novel unsupervised learning tree-based algorithm that yields high-quality solutions via an optimization approach. Through computational experiments with benchmark and real-world datasets, we show that ICOT offers significant gains in interpretability over state-of-the-art clustering methods while achieving comparable or even better performance as measured by well-established internal validation criteria. This makes ICOT an ideal tool for exploratory data analysis as it reveals natural separations of the data with intuitive reasoning.

Chapter 4

Optimal Survival Trees

Tree-based models are increasingly popular due to their ability to identify complex relationships that are beyond the scope of parametric models. Survival tree methods adapt these models to allow for the analysis of censored outcomes, which often appear in medical data. We present the Optimal Survival Trees (OST) algorithm that leverages MIO and local search techniques to generate globally optimized survival tree models. We demonstrate that the OST algorithm improves on the accuracy of existing survival tree methods, particularly in large datasets.

4.1 Introduction

Survival analysis is a cornerstone of healthcare research and is widely used in the analysis of clinical trials as well as large-scale medical datasets such as EHR and insurance claims. Survival analysis methods are required for censored data in which the outcome of interest is generally the time until an event (onset of disease, death, etc.), but the exact time of the event is unknown (censored) for some individuals. When a lower bound for these missing values is known (for example, a patient is known to be alive until at least time t) the data is said to be right-censored.

A common survival analysis technique is Cox proportional hazards regression which

models the hazard rate for an event as a linear combination of covariate effects [87]. Although this model is widely used and easily interpreted, its parametric nature makes it unable to identify non-linear effects or interactions between covariates [53].

Recursive partitioning techniques (also referred to as *trees*) are a popular alternative to parametric models. When applied to survival data, survival tree algorithms partition the covariate space into smaller and smaller regions (*nodes*) containing observations with homogeneous survival outcomes. The survival distribution in the final partitions (*leaves*) can be analyzed using a variety of statistical techniques such as Kaplan-Meier curve estimates [185]. Several authors have proposed algorithms for building survival trees using censored datasets [333, 203, 170], many of which have been implemented within recursive partitioning software packages [332, 169].

Most recursive partitioning algorithms generate trees in a top-down, greedy manner, which means that each split is selected in isolation without considering its effect on subsequent splits in the tree [56, 273, 272]. This approach can have a negative impact on the quality of the model, such as unnecessarily increasing complexity or decreasing accuracy, resulting in poor out-of-sample performance.

To address these issues, researchers have proposed the construction of optimal decision trees, leveraging optimization techniques [75, 249, 307, 345, 344]. Such approaches lead to higher quality solutions while providing the flexibility to impose additional constraints on the trees. As the problem of tree construction is NP-complete [200], recovering the optimal partition in high-dimensional dataset poses scalability issues. [27, 34] have proposed an efficient algorithm which uses modern MIO techniques and addresses this weakness. Similar to other optimization-based approaches, this *Optimal Trees* algorithm forms the entire decision tree in a single step, allowing each split to be determined with full knowledge of all other splits. It allows the construction of single decision trees for classification and regression that have performance comparable with state-of-the-art methods such as Random Forest (RF) and Gradient Boosted Trees (GBT), without sacrificing the interpretability offered by a single-tree model.

The key contributions of this chapter are:

1. We present *Optimal Survival Trees* (OST), a new survival trees algorithm that utilizes the *Optimal Trees* framework to generate interpretable trees for censored data.

2. We propose a new accuracy metric that evaluates the fit of Kaplan-Meier curve estimates relative to known survival distributions in simulated datasets. We also demonstrate that this metric is reasonably consistent with the Integrated Brier Score [150], which can be used to evaluate the fit of Kaplan-Meier curves when the true distributions are unknown.

3. We evaluate the performance of our method in both simulated and real-world datasets and demonstrate improved accuracy relative to two existing algorithms.

4. Finally, we provide an example of how the algorithm can be used to predict the risk of adverse events associated with cardiovascular health in the FHS dataset.

The structure of this chapter is as follows. We review existing survival tree algorithms in Section 4.2 and discuss some of the technical challenges associated with building trees for censored data. In Section 4.4, we give an overview of the Optimal Trees algorithm proposed by [27] and we adapt this algorithm for Optimal Survival Trees in Section 4.4. Section 4.5 begins with a discussion of existing survival tree accuracy metrics, followed by the new accuracy metrics that we have introduced to evaluate survival tree models in simulated datasets. Simulation results are presented in Section 4.6 and results for real-world datasets are presented in Sections 4.7–4.8. We conclude in Section 4.9 with a brief summary of our contributions.

4.2 Review of Survival Trees

Recursive partitioning methods have received a great deal of attention in the literature, the most prominent method being the CART algorithm [56]. Tree-based models are appealing due to their logical, interpretable structure as well as their ability to detect complex interactions

between covariates. However, traditional tree algorithms require complete observations of the dependent variable in training data, making them unsuitable for censored data.

Tree algorithms incorporate a splitting rule which selects partitions to add to the tree, and a pruning rule determines when to stop adding further partitions. Since the 1980s, many authors have proposed splitting and pruning rules for censored data. Splitting rules in survival trees are generally based on either (a) node distance measures that seek to maximize the difference between observations in separate nodes or (b) node purity measures that seek to group similar observation in a single node [374, 240].

Algorithms based on node distance measures compare the two adjacent child nodes that are generated when a parent node is split, retaining the split that produces the greatest difference in the child nodes. Proposed measures of node distance include the two-sample logrank test [77], the likelihood ratio statistic [76] and conditional inference permutation tests [170]. We note that the score function used in Cox regression models also falls into the class of node distance measures, as the partial likelihood statistic is based on a comparison of the relative risk coefficient predicted for each observation.

Dissimilarity-based splitting rules are unsuitable for certain applications (such as the Optimal Trees algorithm) because they do not allow for the assessment of a single node in isolation. We will therefore focus on node purity splitting rules for developing the OST algorithm.

[149] published the first survival tree algorithm with a node purity splitting rule based on Kaplan-Meier estimates. [94] used a splitting rule based on the negative log-likelihood of an exponential model, while [333] proposed using martingale residuals as an estimate of node error. [202] suggested comparing the log-likelihood of a saturated model to the first step of a full likelihood estimation procedure for the proportional hazards model and showed that both the full likelihood and martingale residuals can be calculated efficiently from the Nelson-Aalen cumulative hazard estimator [246, 1]. More recently, [240] proposed a new approach to adjust loss functions for uncensored data based on inverse probability of censoring weights (IPCW).

Most survival tree algorithms make use of cost-complexity pruning to determine the

correct tree size, particularly when node purity splitting is used. Cost-complexity pruning selects a tree that minimizes a weighted combination of the total tree error (i.e., the sum of each leaf node error) and tree complexity (the number of leaf nodes), with relative weights determined by cross-validation. A similar split-complexity pruning method was suggested by [203] for node distance measures, using the sum of the split test statistics and the number of splits in the tree. Other proposals include using the Akaike Information Criterion (AIC) [77] or using a p-value stopping criterion to stop growing the tree when no further significant splits are found [170].

Survival tree methods have been extended to include other non-linear learners, including support vector machines, tree ensembles, and neural networks [129, 168, 214]. [54] adapted the CART-based random forest algorithm to survival data, while both [171] and [175] proposed more general methods that generate survival forests from any survival tree algorithm. "Survival forest" algorithms aggregate the results of multiple trees and aim to produce more accurate predictions by avoiding the instability of single-tree models. In addition, the formulation of the SVM problem has been extended in the survival setting with the objective of maximizing the concordance index for comparable pairs of observations [340, 117]. Neural network survival analysis includes various structures, such as feed forward, deep, and recurrent neural networks [47, 289, 128, 142].

Unlike decision trees, these approaches lead to "black-box" models which are not interpretable and provide little information about how they arrive at their predictions [303, 66]. The issue of interpretability has become central to the adoption and implementation of artificial intelligence models over the past several years [139], particularly in application areas like medicine where algorithmic decisions can directly impact patient lives [279, 59]. Single tree models provide a clear answer to this problem as they are able to capture intrinsic non-linear effects in the data while offering transparency to the user with the full characterization of potential risk profiles [34].

Relatively few survival tree algorithms have been implemented in publicly available, well-documented software. Two user-friendly options are available in **R** [274] packages: Therneau's

algorithm based on martingale residuals is implemented in the **rpart** package [332] and Hothorn's conditional inference (**ctree**) algorithm in the **party** package [169].

4.3 Review of Optimal Predictive Trees

In this section, we briefly review approaches to constructing decision trees, and in particular, we outline the Optimal Trees algorithm. The purpose of this section is to provide a high-level overview of the Optimal Trees framework; interested readers are encouraged to refer to [34] and [106] for more detailed technical information. Section 3.2.1 also summarizes the MIO formulation.

Traditionally, decision trees are trained using a greedy heuristic that recursively partitions the feature space using a sequence of locally-optimal splits to construct a tree. This approach is used by methods like CART [56] to find classification and regression trees. The greediness of this approach is also its main drawback—each split in the tree is determined independently without considering the possible impact of future splits in the tree on the quality of the here-and-now decision. This can create difficulties in learning the true underlying patterns in the data and lead to trees that generalize poorly. The most natural way to address this limitation is to consider forming the decision tree in a single step, where each split in the tree is decided with full knowledge of all other splits

The first efforts in the direction of optimal decision tree construction involved the use of pattern mining techniques to construct a global model [248, 249]. [244] proposes the use of a Boolean satisfiability model for computing small-size decision trees with optimality guarantees. [345] introduce an alternative binary formulation that employs Integer Linear Programming to render the model size largely independent from the training data size, achieving better performance and shorter running times. [344] recently suggested an even more efficient way to decompose the learning problem with a constraint programming approach. Other attempts in the literature to construct globally optimal predictive trees involve the ones of [21, 323, 151]. However, these methods could not scale to datasets of the sizes required by practical applications, and therefore did not displace greedy heuristics as the approach used

in practice.

Optimal Trees is a novel approach for decision tree construction that outperforms many existing decision tree methods [34]. It formulates the decision tree construction problem from the perspective of global optimality using MIO and solves this problem with CD to find optimal or near-optimal solutions in practical run times. These Optimal Trees are often as powerful as state-of-the-art methods like RF or GBT, yet they are just a single decision tree and hence are readily interpretable. This obviates the need to trade off between interpretability and state-of-the-art accuracy when choosing a predictive method.

The Optimal Trees framework is a generic approach that tractably and efficiently trains decision trees according to a loss function of the form

$$\min_{T} \quad \texttt{error}(T, D) + \alpha \cdot \texttt{complexity}(T), \tag{4.1}$$

where T is the decision tree being optimized, D is the training data, $\texttt{error}(T, D)$ is a function measuring how well the tree T fits the training data D, $\texttt{complexity}(T)$ is a function penalizing the complexity of the tree (for a tree with splits parallel to the axis, this is simply the number of splits in the tree), and α is the *complexity parameter* that controls the tradeoff between the quality of the fit and the size of the tree.

Unlike the others, Optimal Trees is able scale to large datasets (n in the millions, p in the thousands) by using CD to train the decision trees towards global optimality. When training a tree, the splits in the tree are repeatedly optimized one-at-a-time, finding changes that improve the global objective value in Problem (4.1). To give a high-level overview, the nodes of the tree are visited in a random order and at each node we consider the following modifications:

- If the node is not a leaf, delete the split at that node;
- If the node is not a leaf, find the optimal split to use at that node and update the current split;
- If the node is a leaf, create a new split at that node.

For each of the changes, we calculate the objective value of the modified tree with respect to Problem (4.1). If any of these changes result in an improved objective value, then the modification is accepted. When a modification is accepted or all potential modifications have been dismissed, the algorithm proceeds to visit the nodes of the tree in a random order until no further improvements are found, meaning that this tree is a locally optimal for Problem (4.1). The problem is non-convex, so we repeat the CD process from various randomly-generated starting decision trees, before selecting the final locally-optimal tree with the lowest overall objective value as the best solution. For a more comprehensive guide to the CD process, we refer the reader to [34].

Although only one tree model is ultimately selected, information from multiple trees generated during the training process is also used to improve the performance of the algorithm. For example, the Optimal Trees algorithm combines the result of multiple trees to automatically calibrate the complexity parameter (α) and to calculate variable importance scores in the same way as RF or boosted trees. More detailed explanations of these procedures can be found in [106].

The CD approach used by Optimal Trees is generic and can be applied to optimize a decision tree under any objective function. For example, the Optimal Trees framework can train OCT by setting $\texttt{error}(T, D)$ to be the misclassification error associated with the tree predictions made on the training data. We provide a comparison of performance between various classification methods from [34] in Figure 4.1. This comparison shows the performance of two versions of OCT: OCT with parallel splits (using one variable in each split); and OCT with hyperplane splits (using a linear combination of variables in each split). These results demonstrate that not only do the Optimal Tree methods significantly outperform CART in producing a single predictive tree, but also that these trees have performance comparable with some of the best classification methods.

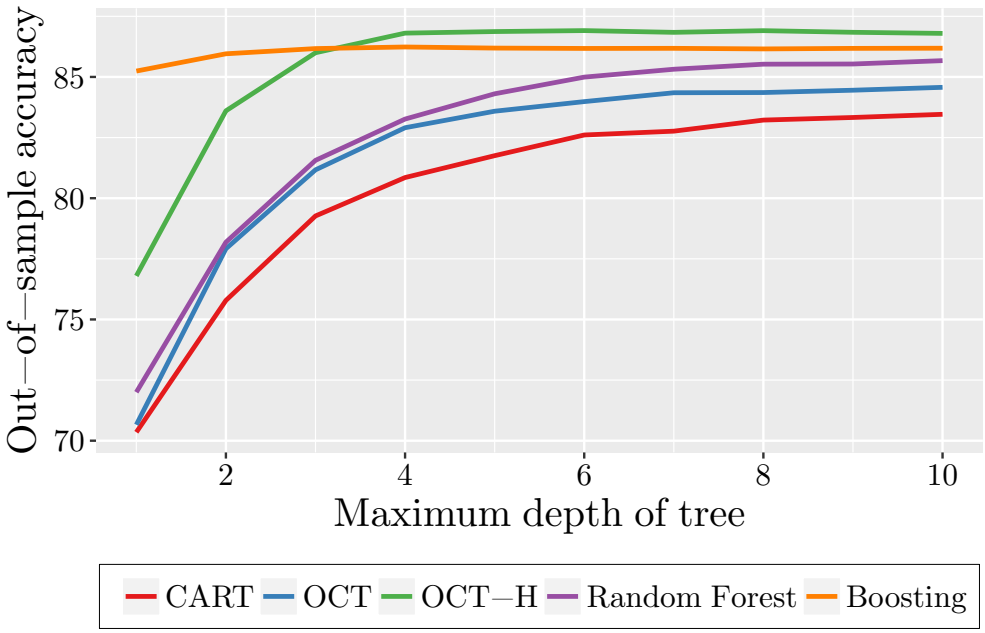


Figure 4.1: Performance of classification methods averaged across 60 real-world datasets. OCT and OCT-H refer to Optimal Classification Trees without and with hyperplane splits, respectively.

4.4 Survival Tree Algorithm

In this section, we adapt the OCT algorithm for the analysis of censored data. For simplicity, we will use terminology from survival analysis and assume that the outcome of interest is the time until death. We begin with a set of observations $(t_i, \delta_i)_{i=1}^n$ where t_i indicates the time of last observation and δ_i indicates whether the observation was a death ($\delta_i = 1$) or a censoring ($\delta_i = 0$).

Like other tree algorithms, the OST model requires a target function that determines which splits should be added to the tree. Computational efficiency is an important factor in the choice of target function, since it must be re-evaluated for every potential change to the tree during the optimization procedures. A key requirement for the target function is that the "fit" or error of each node should be evaluated independently of the rest of the tree. In this case, changing a particular split in the tree will only require re-evaluation of the subtree directly below that split, rather than the entire tree.

Due to these computational constraints, splits in the OST model cannot be evaluated by any methods that require the comparison of two or more nodes within the tree. This requirement restricts the choice of target function to the node purity approaches described in Section 4.2.

The splitting rule implemented in the OST algorithm is based on the likelihood method proposed by [202]. This splitting rule is derived from a proportional hazards model which assumes that the underlying survival distribution for each observation is given by

$$\mathrm{P}(S_i \leq t) = 1 - e^{-\theta_i \Lambda(t)}, \tag{4.2}$$

where $\Lambda(t)$ is the baseline cumulative hazard function and the coefficients θ_i are the adjustments to the baseline cumulative hazard for each observation.

In a survival tree model we replace $\Lambda(t)$ with an empirical estimate for the cumulative probability of death at each of the observation times. This is known as the Nelson-Aalen estimator [246, 1],

$$\hat{\Lambda}(t) = \sum_{i:t_i \leq t} \frac{\delta_i}{\sum_{j:t_j \geq t_i} 1}. \tag{4.3}$$

Assuming this baseline hazard, the objective of the survival tree model is to optimize the hazard coefficients θ_i. We impose that the tree model uses the same coefficient for all observations contained in a given leaf node in the tree, i.e. $\theta_i = \hat{\theta}_{T(i)}$. These coefficients are determined by maximizing the within-leaf sample likelihood

$$L = \prod_{i=1}^{n} \left(\theta_i \frac{d}{dt} \Lambda(t_i) \right)^{\delta_i} e^{-\theta_i \Lambda(t_i)}, \tag{4.4}$$

to obtain the node coefficients

$$\hat{\theta}_k = \frac{\sum_i \delta_i I_{\{T_i = k\}}}{\sum_i \hat{\Lambda}(t_i) I_{\{T_i = k\}}}. \tag{4.5}$$

To evaluate how well different splits fit the available data we compare the current tree model to a tree with a single coefficient for each observation. We will refer to this as a fully saturated tree, since it has a unique parameter for every observation. The maximum likelihood estimates

for these saturated model coefficients are

$$\hat{\theta}_i^{sat} = \frac{\delta_i}{\hat{\Lambda}(t_i)}, \quad i = 1, \ldots, n. \tag{4.6}$$

We calculate the prediction error at each node as the difference between the log-likelihood for the fitted node coefficient and the saturated model coefficients at that node:

$$\texttt{error}_k = \sum_{i:T(i)=k} \left(\delta_i \log \left(\frac{\delta_i}{\hat{\Lambda}(t_i)} \right) - \delta_i \log(\hat{\theta}_k) - \delta_i + \hat{\Lambda}(t_i)\hat{\theta}_k \right). \tag{4.7}$$

The overall error function used to optimize the tree is simply the sum of the errors across the leaf nodes of the tree T given the training data D:

$$\texttt{error}(T, D) = \sum_{k \in \text{leaves}(T)} \texttt{error}_k(D). \tag{4.8}$$

We can then apply the Optimal Trees approach to train a tree according to this error function by substituting this expression into the overall loss function (4.1). At each step of the CD process, we determine new estimates for $\hat{\theta}_k$ for each leaf node k in the tree using (4.5). We then calculate and sum the errors at each node using (4.7) to obtain the total error of the current solution, which is used to guide the CD and generate trees that minimize the error (4.8).

The algorithm is implemented in `Julia` [45] and is available to academic researchers under a free academic license.*

4.5 Survival tree accuracy metrics

In order to assess the performance of the OST algorithm, we now introduce a number of accuracy metrics for survival tree models. We will use the notation T^{true} to represent a tree model, where $T_i^{true} = T^{true}(X_i)$ is the leaf node classification of observation i with covariates

*Please email survival-trees@mit.edu to request an academic license for the Optimal Survival Trees package.

X_i in the tree T^{true}. We will use the notation T^0 to represent a null model (a tree with no splits and a single node).

4.5.1 Review of survival model metrics

We begin by reviewing existing accuracy metrics for survival models that are commonly used in both the literature as well as practical applications.

1. **Cox Partial Likelihood Score**

   The Cox proportional hazards model [87] is a semi-parametric model that is widely used in survival analysis. The Cox hazard function estimate is

   $$\lambda(t|X_i) = \lambda_0(t) \exp(\beta_1 X_{i1} + \cdots + \beta_p X_{ip}) = \lambda_0(t) \exp(\beta^T X_i), \tag{4.9}$$

   where $\lambda_0(t)$ is the baseline hazard function and β is a vector of fitted coefficients. This proportional hazards model does not make any assumptions about the form of $\lambda_0(t)$, and its parameters can be estimated even when the baseline is completely unknown [86]. The coefficients β are estimated by maximizing the partial likelihood function[†],

   $$L(\beta) = \prod_{\mathrm{t_i uncensored}} \frac{\exp(X_i\beta)}{\sum_{t_j \geq t_i} \exp(X_j\beta)} = \prod_{\mathrm{t_i uncensored}} \frac{\theta_i}{\sum_{t_j \geq t_i} \theta_j}. \tag{4.10}$$

   For computational convenience, the Cox model is generally implemented using the log partial likelihood,

   $$l(\beta) = \log L(\beta) = \sum_{\mathrm{t_i uncensored}} X_i\beta - \log(\sum_{t_j \geq t_i} \exp(X_j\beta)). \tag{4.11}$$

   In the context of survival trees, we can find the Cox hazard function associated with a

[†]This definition of the partial likelihood assumes that there are no ties in the data set (i.e., no two subjects have the same event time).

particular tree model by assigning one coefficient to each leaf node in the tree, i.e.,

$$\lambda_T(t) = \lambda_0(t) \exp\left(\sum_{k \in T} \beta_k \mathbb{1}(T_i = k)\right) = \lambda_0(t) \exp\left(\beta_{T_i}\right). \tag{4.12}$$

We define the Cox Score for a tree model as the maximized log partial likelihood for the associated Cox model, $\max_\beta l(\beta|T)$. To assist with interpretation, we also define the Cox Score Ratio (CSR) as the percentage reduction in the Cox Score for tree T relative to a null model,

$$CSR(T) = 1 - \frac{\max_\beta l(\beta|T)}{\max_\beta l(\beta|T^0)}. \tag{4.13}$$

Due to its widespread use in the context of Cox Regression, the Cox Score is a useful metric for assessing the fit of survival tree models and contrasting the structure of these models with more commonly used linear hazard functions. However, it is important to consider the implications of applying a metric designed for continuous hazard predictions in the context of decision trees, which produce a discrete hazard coefficient for every node. Each additional leaf node in the tree allows an additional degree of freedom in equation (4.12), and increasing the number of nodes in the tree may inflate Cox score even if the overall quality of the model does not improve.

Another significant drawback of the Cox score is its reliance on the proportional hazards assumption (4.2). Although this assumption is commonly used in survival analysis, it may not be appropriate in many applications. This metric should be interpreted with caution when comparing the results of survival tree algorithms that use the proportional hazards model in node splitting rules (such as the OST algorithm) to other algorithms that rely on non-parametric splitting rules.

2. **The Concordance Statistic**

Applying a ranking approach to survival analysis is an effective way to deal with the skewed distributions of survival times as well as censored of the data. The Concordance

Statistic, which is most familiar from logistic regression, is another popular metric that has been adapted to measure goodness-of-fit in survival models [159]. The concordance index is defined as the proportion of all *comparable* pairs of observations in which the model's predictions are *concordant* with the observed outcomes.

Two observations are *comparable* if it is know with certainty that one individual died before the other. This occurs when the actual time of death is observed for both individuals (neither is censored) or when the one individual's death is observed before the other is censored. A comparable pair is *concordant* if the predicted risk (ρ) is higher for the individual that died first, and the pair is discordant if the predicted risk is lower for the individual that died first. Thus, the number of concordant pairs in a sample is given by

$$CC = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\rho_i < \rho_j)\delta_j, \tag{4.14}$$

and the number of discordant pairs is

$$DC = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\rho_i > \rho_j)\delta_j, \tag{4.15}$$

where the indices i and j refer to pairs of observations in the sample. Multiplication by the factor δ_j discards pairs of observations that are not comparable because the smaller survival time is censored, i.e., $\delta_j = 0$. These definitions do not include comparable pairs with tied risk predictions, so we denote these pairs as

$$TR = \sum_{i,j} \mathbb{1}(t_i > t_j)\mathbb{1}(\rho_i = \rho_j)\delta_j. \tag{4.16}$$

The number of concordant and discordant pairs is commonly summarized using Harrell's C-index [159],

$$H_C = \frac{CC + 0.5 \times TR}{CC + DC + TR}. \tag{4.17}$$

Harrell's C takes values between 0 and 1, with higher values indicating a better fit.

Note that randomly assigned predictions have an expected score of $H_C = 0.5$.

More recently, [338] introduced a modified C-Statistic that weights comparable pairs of observations based on the distribution of censoring times,

$$U_{C_t} = \frac{\sum_{i,j}(\hat{G}(t_j))^{-2}\mathbb{1}(t_i > t_j, t_j < t)\mathbb{1}(\rho_i < \rho_j)\delta_j}{\sum_{i,j}(\hat{G}(t_j))^{-2}(\mathbb{1}(t_i > t_j, t_j < t)\mathbb{1}(\rho_i > \rho_j)\delta_j + \mathbb{1}(t_i > t_j, t_j < t)\mathbb{1}(\rho_i < \rho_j)\delta_j)}, \tag{4.18}$$

where $\hat{G}(\cdot)$ is the Kaplan-Meier estimate for the censoring distribution. Due to these coefficients, U_C converges to a quantity that is independent of the censoring distribution. U_C takes values between 0 and 1, with higher values indicating a better fit.

The above definition of Uno's C-statistic was intended for continuous models, and (4.18) may be very unstable in small trees due to the large number of observations with tied risks which are not counted in either the numerator or denominator. To avoid this, we include these pairs of observations in a similar manner to Harrell's C-statistic, i.e., weighted by 0.5 in the numerator and 1 in the denominator. The resulting concordance statistic is

$$U^*_{C_t} = \frac{\sum_{i,j}(\hat{G}(t_j))^{-2}\mathbb{1}(t_i > t_j, t_j < t)\left(\mathbb{1}(\rho_i < \rho_j) + 0.5 \times \mathbb{1}(\rho_i = \rho_j)\right)\delta_j}{\sum_{i,j}(\hat{G}(t_j))^{-2}(\mathbb{1}(t_i > t_j, t_j < t)\mathbb{1}(\rho_i > \rho_j)\delta_j + \mathbb{1}(t_i > t_j, t_j < t)\mathbb{1}(\rho_i \leq \rho_j)\delta_j)}. \tag{4.19}$$

This modification improves the stability of the concordance statistics but also makes these metrics somewhat less informative in the context of discrete models, since a large number of tied pairs tend to dominate both the numerator and denominator. More generally, concordance statistics do not account for incomparable pairs of observations, which may be problematic when there is significant censoring. The binary definition of concordance fails to account for the magnitude of the difference in predicted risks for comparable observations. As a result, these metrics may be less informative in datasets with significant variations in risk.

Unlike the Cox Score, concordance statistics do not explicitly rely on any parametric assumptions. For proportional hazards models it is natural to define the predicted risk

in terms of the hazard coefficients in (4.2), i.e., $\rho_i = \theta_i$. However, it is also possible to contrast the predicted risk of a comparable pair of observations via the predicted survival probabilities, the expected survival times, or any other comparable prediction extracted from the model. In our analysis we evaluate concordance based on the predicted survival probabilities extracted from the Kaplan-Meier curves at each node, i.e., $\rho_i(\tau) = 1 - \hat{S}_i(\tau)$. When comparing the risks of a pair of observations, survival probabilities are evaluated at the time of the first event, $\tau = \min\{t_i, t_j\}$.

3. **Integrated Brier score**

The Brier score metric is commonly used to evaluate classification trees [57]. It was originally developed to verify the accuracy of a probability forecast, primarily for weather forecasting. The most common formula calculates the mean squared prediction error:

$$B = \frac{1}{n}\sum_{i}^{n}(\hat{p}(y_i) - y_i)^2, \tag{4.20}$$

where n is the sample size, $y_i \in \{0, 1\}$ is the outcome of observation i, and $\hat{p}(y_i)$ is the forecast probability of this observed outcome. In the context of survival analysis, the Brier score may be used to evaluate the accuracy of survival predictions at a particular point in time relative to the observed deaths at that time. We will refer to this as the Brier Point Score:

$$BP_\tau = \frac{1}{|\mathcal{I}_\tau|}\sum_{i\in\mathcal{I}_\tau}(\hat{S}_i(\tau) - \mathbb{1}(t_i > \tau))^2, \tag{4.21}$$

$$\text{where } \mathcal{I}_\tau = \{i \in \{1, \ldots, n\}, |t_i \geq \tau \text{ or } \delta_i = 1\}.$$

In this case, $\hat{S}_i(\tau)$ is the predicted survival probability for observation i at time τ and $\mathcal{I}_\tau$ is the set of observations that are known to be alive/dead at time τ. Observations

censored before time τ are excluded from this score, as their survival status is unknown.

Applying this version of the Brier score may be useful in applications where the main outcome of interest is survival at a particular time, such as the 1-year survival rates after the onset of a disease. In the experiments that follow, the point-wise Brier Score will be evaluated at the median observation time in each dataset. For easy interpretation, the reported scores are normalized relative to the score for a null model, i.e.

$$BPR_\tau = 1 - \frac{BP_\tau(T)}{BP_\tau(T^0)}. \tag{4.22}$$

The Brier Point score has two significant disadvantages in survival analysis. First, it assesses the predictive accuracy of survival models a single point in time rather than over the entire observation period, which is not well-suited to applications where survival distributions are the outcome of interest. Second, it becomes less informative as the number of censored observations increases, because a greater number of observations are discarded when calculating the score.

[150] have addressed these challenges by proposing an adjusted version of the Brier Score for survival datasets with censored outcomes. Rather than measuring the accuracy of survival predictions at a single point, this measure aggregates the Brier score over the entire time interval observed in the data. This modified measure is commonly used in the survival literature and has been interchangeably called the Brier Score or the Integrated Brier Score by various authors [282]. In this chapter, we will refer to the metric specific to survival analysis as the Integrated Brier score (IB), defined as

$$IB = \frac{1}{t_{max}} \frac{1}{n} \sum_{i=1}^{n} \int_0^{t_i} \frac{(1 - \hat{S}_i(t))^2}{\hat{G}(t)} dt + \delta_i \int_{t_i}^{t_{max}} \frac{(\hat{S}_i(t))^2}{\hat{G}(t_i)} dt. \tag{4.23}$$

The IB score uses Kaplan-Meier estimates for both the survival distribution, $\hat{S}(t)$, and the censoring distribution, $\hat{G}(t)$. In a survival tree model, these estimates are obtained by pooling observations in each node in the tree, i.e., $\hat{S}_i(t) = \hat{S}_{T(i)}(t)$. The IB score

is a weighted version of the original Brier Score, with the weights being $1/\hat{G}(t_i)$ if an event occurs before time t_i, and $1/\hat{G}(t)$ if the event occurs after time t. This metric addresses many of the deficiencies identified in the Cox and concordance scores above: it is non-parametric, counts both censored and uncensored observations, and evaluates accuracy of the predicted survival functions over the entire time horizon.

In subsequent sections, we report a normalized version of this metric, the Integrated Brier score ratio (IBR), which compares the sum of the Integrated Brier scores in a given tree to the corresponding Integrated Brier scores in a null tree[‡]:

$$IBR = 1 - \frac{IB(T)}{IB(T^0)}. \tag{4.24}$$

Aside from the limitations already discussed, we note that all of the above metrics are subject to noise and often provide contradictory assessments when comparing different tree models. For example, our empirical experiments comparing three candidate models were only able to identify a non-dominated model for about 30% of the instances. In the other 70% of our test cases, none of the three candidate models scored at least as high as the other models on all metrics. These limitations make it difficult to obtain an unambiguous comparison between the performance of different survival tree algorithms. To address this challenge, we will now introduce a simulation procedure and associated accuracy metrics that are specifically designed to assess survival tree models.

4.5.2 Simulation accuracy metrics

A key difficulty in selecting performance metrics for survival tree models is that the definition of "accuracy" can depend on the context in which the model will be used. For example, consider a survival tree that models the relationship between lifestyle factors and age of death. A medical researcher may use such a model to *identify risk factors* associated with early death, while an insurance firm may use this model to *predict mortality risks* for individual clients

[‡][275] calls this *explained residual variation.*

in order to estimate the volume of life insurance policy pay-outs in the coming years. The medical researcher is primarily interested whether the model has identified important splits, while the insurer is more focused on whether the model can accurately estimate survival distributions.

In subsequent sections we refer to these two properties as *tree recovery* and *prediction accuracy*. We develop metrics to measure these outcomes in simulated datasets with the following structure:

Let $i = 1, \ldots, n$ be a set of observations with independent, identically distributed covariates $\mathbf{X}_i = (X_{ij})_{j=1}^m$. Let T^{true} be a tree model that partitions observations based on these covariates such that $T_i^{true} = T^{true}(\mathbf{X}_i)$ is the index of the leaf node in T^{true} that contains individual i. Let S_i be a random variable representing the survival time of observation i, with distribution $S_i \sim F_{T_i^{true}}(t)$. The survival distribution of each individual is entirely determined by its location in the tree T^{true}, and so we refer to T^{true} as the "true" tree model.

This underlying tree structure provides an unambiguous target against which we can measure the performance of empirical survival tree models. In this context, an empirical survival tree model T has high accuracy if it achieves the following objectives:

1. Tree recovery: the model recovers structure of the true tree (i.e., $T(\mathbf{X}_i) = T^{true}(\mathbf{X}_i)$).

2. Prediction accuracy: the model recovers the corresponding survival distributions of the true tree (i.e., $\hat{F}_{T_i}(t) = F_{T_i^{true}}(t)$).

It is important to recognize that these two objectives are not necessarily consistent, particularly in small samples. For example, models with perfect tree recovery may have a small number of observations in each leaf node, leading to noisy survival estimates with low prediction accuracy.

Tree recovery metrics

We measure the tree recovery of an empirical tree model (T) relative to the true tree (T^{true}) using the following metrics:

1. **Node homogeneity** The node homogeneity statistic measures the proportion of the observations in each node $k \in T$ that have the same true class in T^{true}. This metric is equivalent to the misclassification error and cluster purity metrics which are commonly used in the clustering and tree-based binary classification evaluation contexts respectively [133, 286]. Let $p_{k,l}$ be the proportion of observations in node $k \in T$ that came from class $\ell \in T^{true}$ and let $n_{k,l}$ be the total number of observations at node $k \in T^{true}$ from class $\ell \in C$. Then,

$$NH = \frac{1}{n} \sum_{k \in T} \sum_{l \in T^{true}} n_{k,l} p_{k,l}. \tag{4.25}$$

A score of $NH = 1$ indicates that each node in the new tree model contains observations from a single class in T^{true}. This does not necessarily mean that the structure of T is identical to T^{true} — For example, a saturated tree with a single observation in each node would have a perfect node homogeneity score (see Figure 4.2). The node homogeneity metric is therefore biased towards larger tree models with few observations in each node.

2. **Class recovery**

Class recovery is a measure of how well a new tree model is able to keep similar observations together in the same node, thereby avoiding unnecessary splits. Class recovery is calculated by counting the proportion of observations from a true class $\ell \in T^{true}$ that are placed in the same node in T. Let $q_{k,l}$ be the proportion of observations from class $\ell \in T^{true}$ that are classified in node $k \in T$ and let $n_{k,l}$ be the total number of observations at node $k \in T$ from class $\ell \in T^{true}$. Then,

$$CR = \frac{1}{n} \sum_{\ell \in T^{true}} \sum_{k \in T} n_{k,l} q_{k,l}. \tag{4.26}$$

This metric is biased towards smaller trees, since a null tree with a single node would have a perfect class recovery score. It is therefore useful to consider both the class

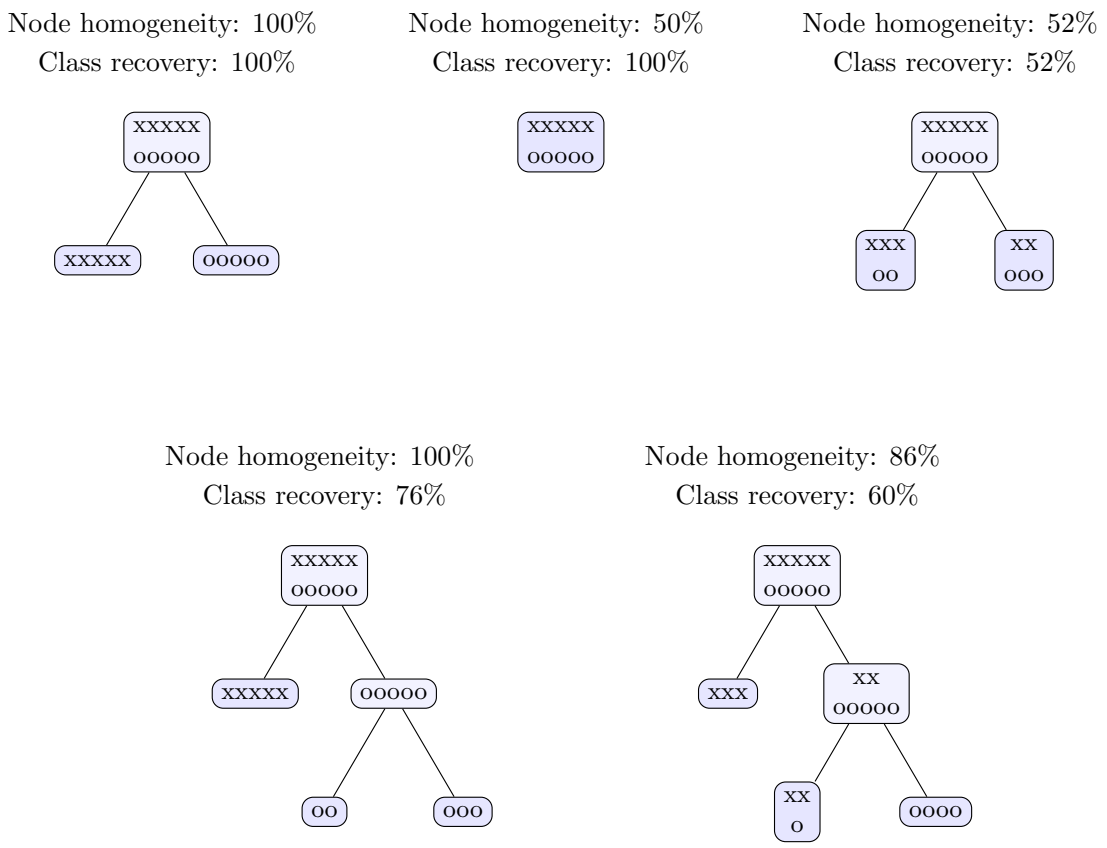


Figure 4.2: Tree recovery metrics for a survival tree with two classes of observations. The top left tree represents the true tree model.

recovery and node homogeneity scores simultaneously in order to assess the performance of a tree model (see Figure 4.2 for examples). When used together, these metrics indicate how well the model T reflects the structure of the true model T^{true}.

The node homogeneity and class recovery scores can also be used to compare any two tree models, T^a and T^b. In this case, these metrics should be interpreted as a measure of structural similarity between the two tree models. Note that when T^a and T^b are applied to the same dataset, the node homogeneity for model T^a relative to T^b is equivalent to the class recovery for T^b relative to T^a, and vice versa. The average node homogeneity score for T^a and T^b is therefore equal to the average class recovery score for T^a and T^b. We will refer to this as the *similarity score* for models T^a and T^b.

Prediction accuracy metric

Our prediction accuracy metric measures how well the non-parametric Kaplan-Meier curves at each leaf in T estimate true the survival distribution of each observation.

1. **Area between curves (ABC)**

   For an observation i with true survival distribution $F_{T_i^{true}}(t)$, suppose that $\hat{S}_{T_i}(t)$ is the Kaplan-Meier estimate at the corresponding node in tree T (see Figure 4.3). The area between the true survival curve and the tree estimate is given by

   $$ABC_i^T = \frac{1}{t_{max}} \int_0^{t_{max}} |1 - F_{T_i^{true}}(t) - \hat{S}_{T_i}(t)| dt. \tag{4.27}$$

   To make this metric easier to interpret, we compare the area between curves in a given tree to the score of a null tree with a single node (T^0). The area ratio (AR) is given by

   $$AR = 1 - \frac{\sum_i ABC_i^T}{\sum_i ABC_i^{T^0}}. \tag{4.28}$$

   Similar to the popular R^2 metric for regression models, the AR indicates how much accuracy is gained by using the Kaplan-Meier estimates generated by the tree relative to the baseline accuracy obtained by using a single estimate for the whole population.

4.6 Simulation results

In this section we evaluate the performance of the Optimal Survival Trees (OST) algorithm and compare it to two existing survival tree models available in the **R** packages **rpart** and **ctree**. Our tests are performed on simulated datasets with the structure described in Section 4.5.2.

4.6.1 Simulation procedure

The procedure for generating simulated datasets in these experiments is as follows:

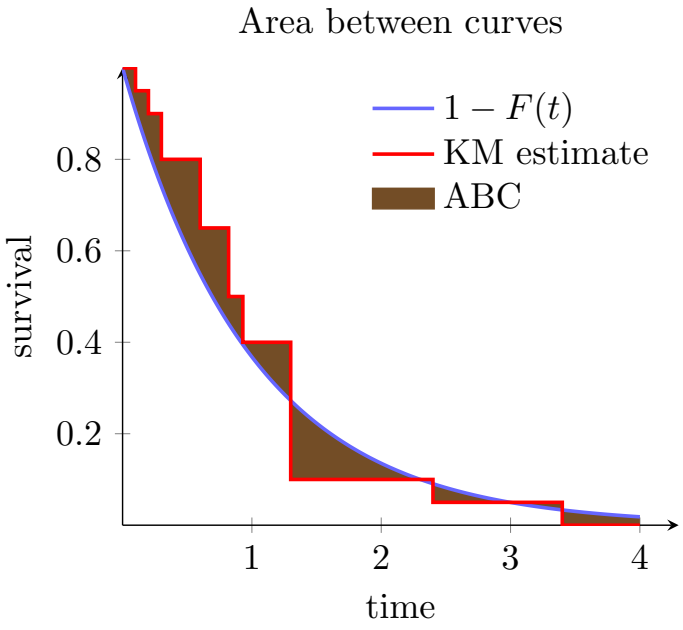


Figure 4.3: An illustration of the area between the true survival distribution and the Kaplan-Meier curve.

1. Randomly generate a sample of 20,000 observations with six covariates. The first three covariates are uniformly distributed on the interval $[0, 1]$ and remaining three covariates are discrete uniform random variables with 2, 3 and 5 levels.

2. Generate a random "ground truth" tree model, T^{true}, that partitions the dataset based on these six covariates.

3. Assign a survival distribution to each leaf node in the tree T^{true}.

4. Classify observations into node classes $T_i^{true} = C(\mathbf{X}_i)$ according to the ground truth model. Generate a survival time, s_i, for each observation based the survival distribution of its node: $S_i \sim F_{T_i^{true}}(t)$.

5. Generate a censoring time for each observation, $c_i = \kappa(1 - u_i^2)$, where u_i follows a uniform distribution and κ is a non-negative parameter used to control the proportion of censored individuals.

6. Assign observation times $t_i = \min(s_i, c_i)$. Individuals are marked as censored ($\delta_i = 0$) if $t_i = c_i$.

We used this procedure to generate 1000 datasets based on ground truth trees with a minimum depth of 3 and a maximum depth of 4 (i.e., $2^4 = 16$ leaf nodes). In each dataset, 10000 observations were set aside for testing the tree models. Training datasets of n observations were sampled from the remaining data for $n \in \{100, 200, 500, 1000, 2000, 5000, 10000\}$.

In addition to varying the size of the training dataset, we also varied the proportion of censored observations in the data by adjusting the parameter κ. Censoring was applied at nine different levels to generate examples with low censoring (0%, 10%, 20%), moderate censoring (30%, 40%, 50%) and high censoring (60%, 70%, 80%). In total, 63 OST models were trained for each dataset to test each of the seven training sample sizes at each of the nine censoring levels.

We evaluated the performance of the OST algorithm relative to two existing survival tree algorithms available in the **R** packages **rpart** [332] and **ctree** [169]. Each of the three algorithms was trained and tested on exactly the same data in each dataset.

Each of the three algorithms tested require two input parameters that control the model size: a maximum tree depth and a complexity/significance parameter that determines which splits are worth keeping in the tree (the interpretation of the **ctree** significance parameter is different to the complexity parameters in the OST and **rpart** algorithms, but it serves a similar function).

Since neither **rpart** nor **ctree** have built-in methods for selecting tree parameters, we used a similar 5-fold cross-validation procedure on the training data to select the parameters for each algorithm. We considered tree depths up to three levels greater than the true tree depth and complexity parameter/significance values between 0.001 and 0.1 for the **rpart** and **ctree** algorithms (the OST complexity parameter is automatically selected during training). Equation (4.7) was used as the scoring metric to evaluate out-of-sample performance during cross-validation, and the minimum node size for all algorithms was fixed at 5 observations.

4.6.2 Results

To demonstrate the effect of this cross-validation procedure, we summarize the average size of the models produced by each algorithm in Figure 4.4. We see a clear link between tree size and the number of training observations, indicating the cross-validation procedure is selecting more conservative depth/complexity parameters when relatively little data is available. In larger datasets, the OST models grow to approximately the same size as the true tree models (6 nodes, on average), while the **rpart** and **ctree** models models are slightly larger.

Survival analysis metrics

Figure 4.5 summarizes the performance of each algorithm in our simulations using the five survival model metrics from Section 4.5.1. The values displayed in each chart are the average performance statistics across all test datasets.

As expected, the average performance of all three algorithms consistently improves as the size of the training dataset increases. The performance statistics also increase as the proportion of censored observations increases, which seems counter-intuitive (we would expect more censoring to lead to less accurate models). In the case of the Cox partial likelihood and C-statistics, this trend is directly linked to the number of observed deaths, since only observations with observed deaths contribute to the partial likelihood and concordance scores. Similarly, censored observations do not contribute to the Integrated Brier Score after their

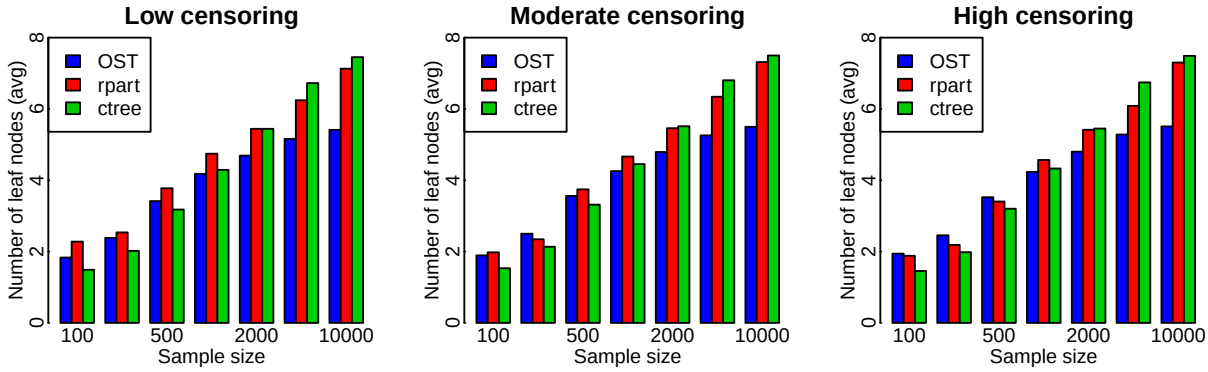


Figure 4.4: The average tree size for models trained on various sample sizes.

	Low censoring			Moderate censoring			High censoring		
n	rpart	ctree	OST	rpart	ctree	OST	rpart	ctree	OST
100	38/87	**40**/77	37/**93**	38/90	**40**/78	37/**92**	37/89	**40**/78	37/**90**
200	42/89	**45**/76	43/**91**	42/**90**	**46**/77	45/90	42/**91**	45/78	**45**/90
500	53/84	56/71	**57**/**88**	55/84	57/70	**59**/**88**	53/85	56/72	**59**/**88**
1000	63/82	66/63	**68**/**89**	65/82	67/63	**70**/**89**	64/82	66/64	**70**/**89**
2000	70/81	73/57	**76**/**89**	72/81	75/57	**78**/**90**	72/81	74/58	**78**/**90**
5000	76/80	82/53	**84**/**91**	77/80	83/53	**85**/**92**	77/80	82/53	**85**/**91**
10000	82/79	85/50	**87**/**91**	84/79	86/51	**89**/**92**	84/78	86/51	**88**/**91**

Table 4.1: A summary of the average node homogeneity/class recovery scores for synthetic experiments.

censoring time.

Each chart also indicates the performance of the true tree model, C, as a point of comparison for the other algorithms. The true tree model performs significantly better than the empirical models trained on smaller datasets, but all three algorithms approach the performance of the true tree for very large sample sizes.

Based on these results, we conclude that the average performance of the OST algorithm in these simulations is consistently better than either of the other two algorithms. In order to understand why this algorithm is able to generate better models, we now analyse the results of the tree metrics introduced in Section 4.5.2.

Tree recovery

The test set tree recovery metrics for all three algorithms are summarized in Table 4.1 and Figure 4.6. The average node homogeneity/class recovery scores are given side-by-side to allow for a comprehensive assessment of each algorithm's performance. These results confirm that the OST models perform significantly better than the other two models across all censoring levels.

The node homogeneity scores for all three algorithms increase with larger sample sizes, indicating that the availability of additional data leads to better detection of relevant splits. In large populations, the OST algorithm selects more efficient splits than the other models and is able to achieve better node homogeneity with fewer splits (recall Figure 4.4 — the OST

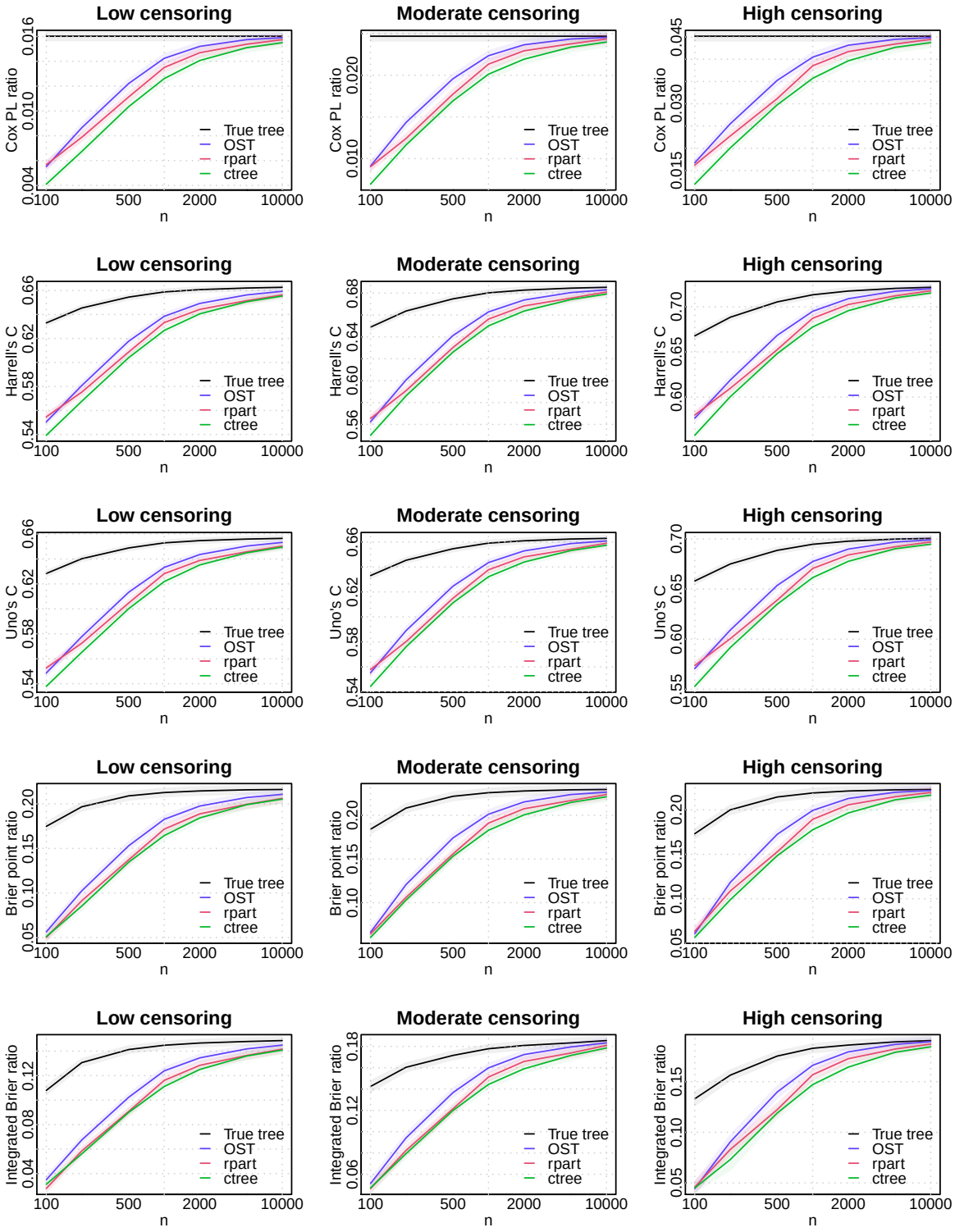


Figure 4.5: A summary of the survival model metrics from simulation experiments. The average test set outcomes for each algorithm are shown in color, while the performance of the true tree model, T^{true}, in indicated in black. Shaded areas indicate 95% confidence intervals.

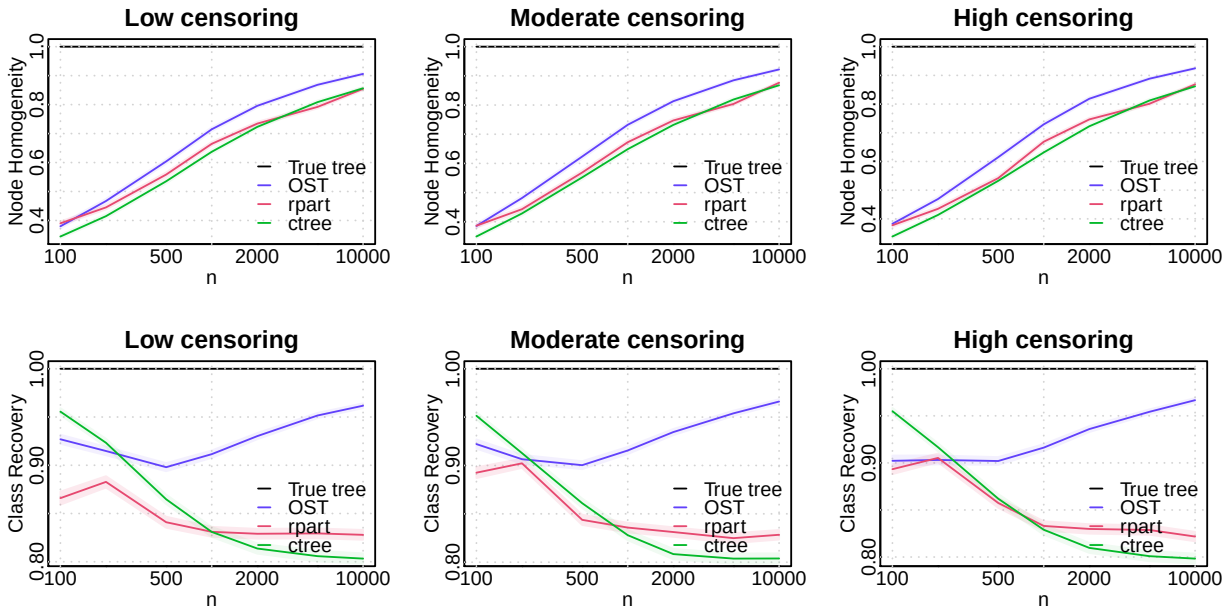


Figure 4.6: A summary of the tree recovery metrics for survival tree algorithms.

models trained on large data sets have fewer leaf nodes than the other models, on average).

The relationship between tree size and class recovery rates is somewhat more complicated. In datasets smaller than 500 observations the class recovery rates seem to be closely linked to the tree size: the **ctree** models have the highest average class recovery for models trained on 100 and 200 observations, and also the smallest number of nodes (see Figure 4.4). However, this trend does not hold in datasets with 500 observations, where OST models are larger than the **ctree** models on average, but also have slightly better class recovery. This suggests that tree size is no longer a dominant factor in larger datasets ($n \geq 500$).

In these larger datasets we observe distinct trends in class recovery scores. The OST class recovery rate increases consistently despite the increases in model size, which means that the OST models are able to produce more complex trees without overfitting in the training data. By contrast, both of the other algorithms have consistently worse class recovery rates as sample size increases and their models become larger. Based on this trend, neither of these algorithms will reliably converge to the true tree.

	Low censoring			Moderate censoring			High censoring		
n	rpart	ctree	OST	rpart	ctree	OST	rpart	ctree	OST
100	6.87	4.79	**9.30**	10.61	7.74	**11.01**	**10.79**	7.76	9.99
200	18.69	16.82	**20.99**	21.93	21.09	**25.25**	24.20	21.24	**26.13**
500	35.03	32.56	**41.17**	40.14	37.12	**47.16**	40.84	38.34	**48.21**
1000	51.27	44.29	**56.44**	57.28	49.68	**61.99**	58.86	51.30	**63.95**
2000	62.76	55.04	**67.97**	68.71	60.30	**73.53**	70.35	61.67	**75.31**
5000	72.62	66.94	**79.45**	77.26	71.63	**83.50**	79.22	72.38	**84.68**
10000	80.06	73.57	**84.41**	84.84	77.44	**87.77**	85.80	77.94	**88.72**

Table 4.2: A summary of the average Kaplan-Meier area ratio (AR) scores for simulation experiments.

Prediction accuracy

The test set prediction accuracy metric for each of the three algorithms is summarized in Table 4.2 and Figure 4.7. Overall, the results indicate that sample size plays the most significant role in test set accuracy across all three algorithms. There is also a small increase in accuracy when censoring is increased, which is due to the reduction in the maximum observed time, t_{max}. The OST results are generally better than the other algorithms across all sample sizes, although the performance gap is relatively small in smaller datasets.

To illustrate the effect of sample size on the accuracy of the Kaplan-Meier estimates, Figure 4.7 also shows the curve accuracy metrics for the true tree, T^{true}. It is immediately apparent that even the true tree models produce poor survival curve estimates in small datasets. Based on these results, it may be necessary to increase the minimum node size to at least 50 observations in applications where Kaplan-Meier curves will be used to summarize survival tree nodes.

Comparison of accuracy metrics

Table 4.3 shows the correlation between each pair of accuracy metrics used in the simulation experiments. All outcome metrics are positively correlated with the exception of class recovery, which has both weak positive and weak negative correlations with other metrics. These mixed results are due to the different trends in class recovery among the three algorithms – OST

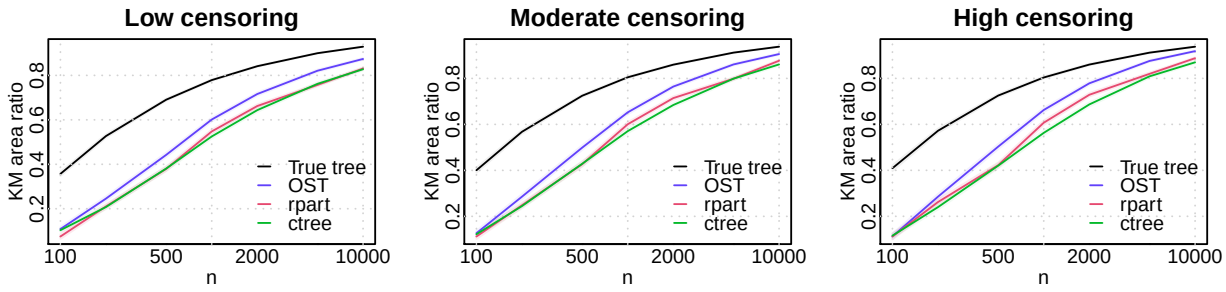


Figure 4.7: A summary of the average Kaplan-Meier Area Ratio results for simulation experiments. The performance of the true tree model is indicated in black.

class recovery was highest for trees trained on larger datasets, while the other algorithms had lower class recovery in these instances (see Figure 4.6). Node homogeneity was positively correlated with other metrics, but the correlations were somewhat weaker than average. This reflects the incomplete information captured by this metric – node homogeneity alone does not guarantee a good model, as discussed in Section 4.5.2.

Among the other metrics, the highest correlation was observed between the two concordance statistics (0.98), which also had the strongest correlation with most other metrics. There was also high correlation between the two Brier metrics (0.86). The Cox score was most strongly correlated with the concordance statistics (0.87), followed by the Brier statistics (0.77). The Kaplan-Meier area ratio had slightly lower average correlations and was most strongly correlated with the node homogeneity statistic. This is likely due to the fact that both of these metrics are based on the true tree structure, while other metrics reflect how well a model fits the available data.

Stability

A frequent criticism of single-tree models is their sensitivity to small changes in the training data. This may be apparent when a tree algorithm produces very different models for different training datasets sampled from the same population. This type of instability is often an indication that the model will not perform well on unseen data.

Given the challenges associated with measuring the test set accuracy for survival tree

	Cox PL	Harrell's C	Uno's C	Brier point	Integrated Brier	Node Homogeneity	Class Recovery	KM area
Cox PL	1.00	0.87	0.87	0.78	0.77	0.49	-0.03	0.59
Harrell's C	0.87	1.00	0.98	0.90	0.80	0.71	-0.12	0.80
Uno's C	0.87	0.98	1.00	0.87	0.79	0.71	-0.12	0.81
Brier point	0.78	0.90	0.87	1.00	0.86	0.60	0.00	0.71
Integrated Brier	0.77	0.80	0.79	0.86	1.00	0.55	0.02	0.66
Node Homogeneity	0.49	0.71	0.71	0.60	0.55	1.00	-0.03	0.87
Class Recovery	-0.03	-0.12	-0.12	0.00	0.02	-0.03	1.00	0.02
KM area	0.59	0.80	0.81	0.71	0.66	0.87	0.02	1.00

Table 4.3: Correlation between different accuracy metrics in simulation experiments.

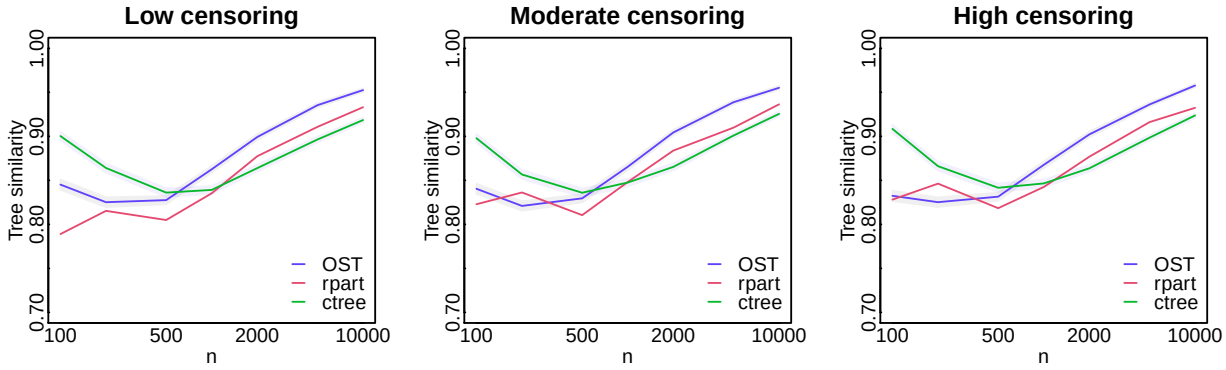


Figure 4.8: A summary of the average similarity scores between pairs of trees trained on mutually exclusive sets of observations.

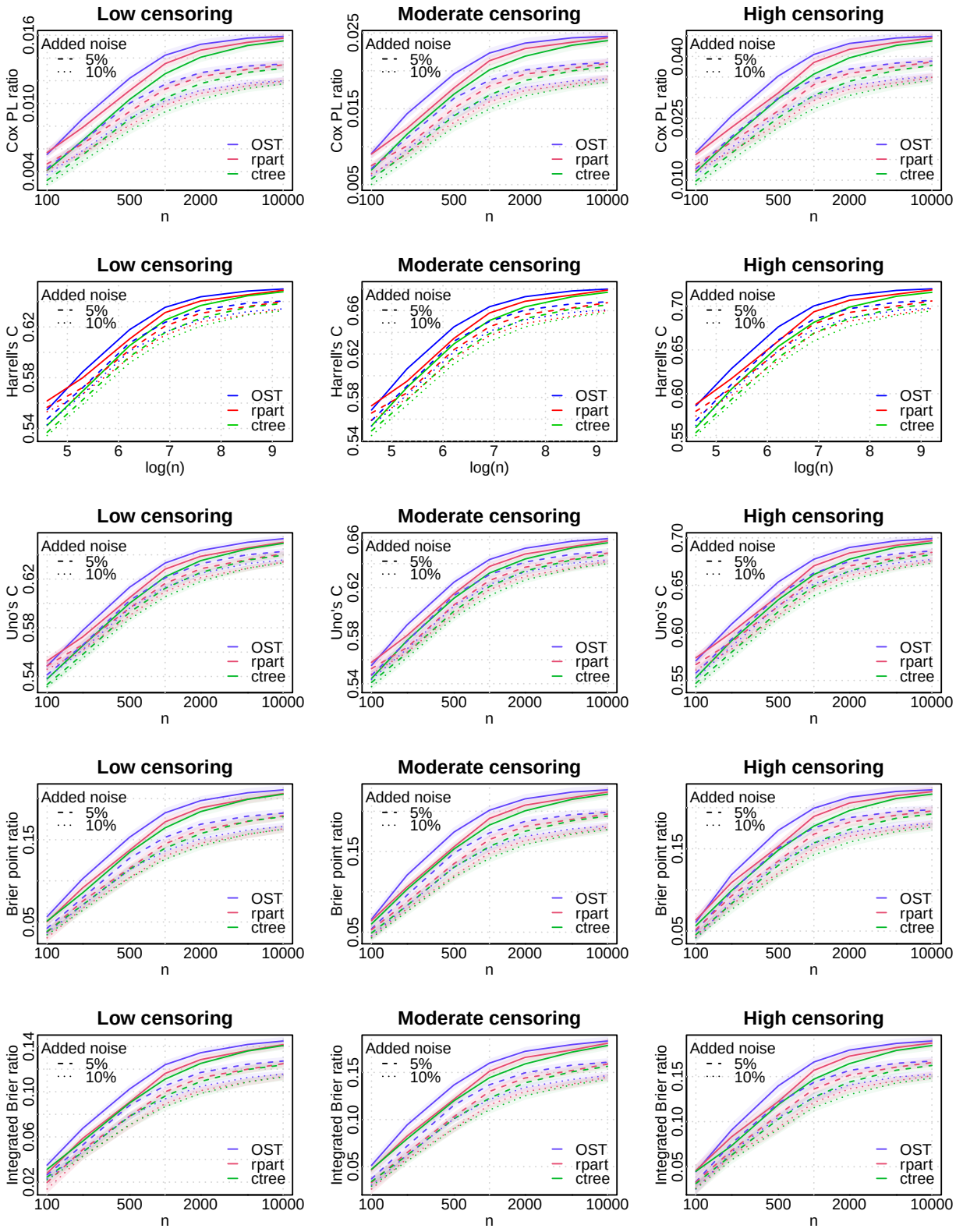


Figure 4.9: A summary of survival tree accuracy metrics for datasets with added noise.

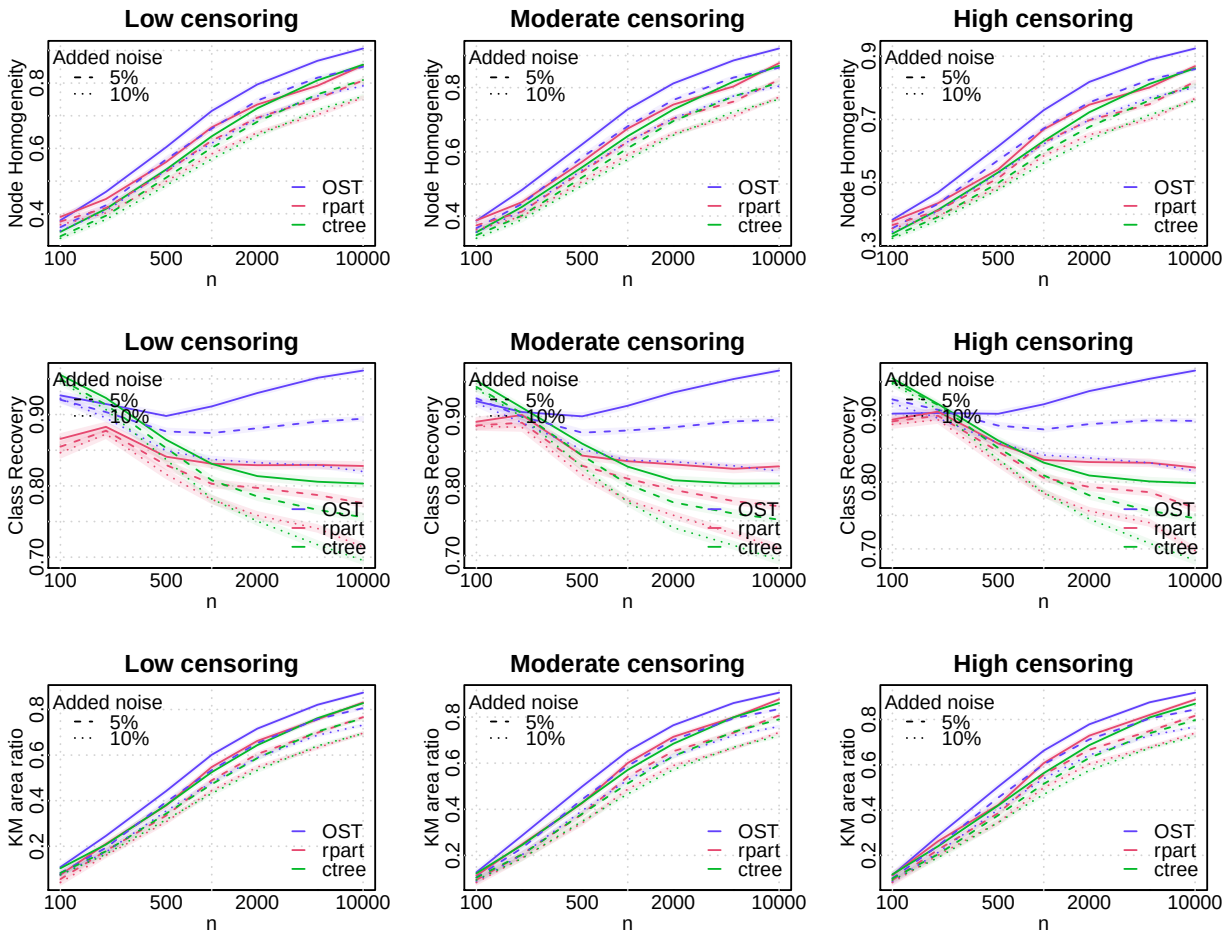


Figure 4.10: A summary of simulation accuracy metrics for datasets with added noise.

algorithms, it may be tempting to use stability as a performance metric for these models. Stability is a necessary condition for accuracy in tree models (provided that a tree structure is suitable for the data) but stable models are not necessarily accurate. For example, greedy tree models with depth 1 may select the same split for all permutations of the training data, but these models will not be accurate if the data requires a tree of depth 3.

Although stability is not necessarily a good indicator of the quality of a model, it is nevertheless interesting to consider how the stability of globally optimized trees may differ to the stability of greedy trees. Globally optimized trees are theoretically capable of greater stability because they may include splits that are not necessarily locally optimal for a particular training dataset. However, globally optimized trees also consider a significantly larger number of possible tree configurations and therefore have many more opportunities for overfitting on features of a particular training dataset.

We ran two sets of experiments to investigate the stability of the survival tree models in our simulations. In the first set of experiments we used each algorithm to train two models, T^a and T^b, on non-overlapping training datasets of equal size drawn from the same population. We then applied each model to the entire dataset (20000 observations) and used the tree similarity score described in Section 4.5.2 to assess the structural similarity between the two models. The average similarity scores for each algorithm are illustrated in Figure 4.8.

These results demonstrate that stability across different training datasets is not a sufficient condition for accuracy: models trained on 100 and 200 observations are both more stable and less accurate than models trained on 500 observations. The **ctree** algorithm produced the most stable results in smaller datasets due to the smaller model sizes selected during cross-validation. For example, 33.1% of **ctree** models trained on 100 observations had fewer than 2 splits, compared to 29.5% of the **rpart** models and 26.5% of the OST models.

The stability results for larger training datasets ($n > 1000$) are reasonably consistent with the accuracy metrics discussed above, and both stability and accuracy increase with sample size across all three algorithms. The OST models have the highest average similarity scores in large datasets and the **rpart** models are slightly more stable than the **ctree** models.

In the second set of stability experiments we investigated how small perturbations to the covariate values in the training dataset affect the test set accuracy of each model. We added noise to the training data by replacing the original continuous covariate values, x_{ij}, with "noisy" values $\tilde{x}_{ij} = x_{ij} + \epsilon_{ij}$. The initial covariates were uniformly distributed between 0 and 1 and the added noise terms were generated from the following two distributions:

$$\epsilon_{ij} \sim U(-0.05, 0.05) \qquad \text{(5\% noise), and}$$
$$\epsilon_{ij} \sim U(-0.1, 0.1) \qquad \text{(10\% noise).}$$

A similar approach was applied to the categorical variables, which were generated by rounding off continuous values (x_{ij} or $\tilde{x}_{ij}$) to the appropriate thresholds. Note that noise was only added to the observations used for training data; the testing data was unchanged.

The results of these experiments are contrasted with the initial outcomes (without added noise) in Figures 4.9-4.10. The effects of additional noise in the training data are visible in the results of all three algorithms and the drop in accuracy appears to be fairly consistent. Overall, the OST models maintain the highest scores regardless of noise.

These results indicate that perturbations in the training data affect the OST and greedy tree algorithms in similar ways. The OST algorithm's performance is diminished by adding noise to the training data, but its ability to consider a wider range of split configurations does not make it more sensitive to these perturbations. In fact, the OST algorithm is generally slightly more stable than the greedy algorithms across permutations of the training data because it tends to produce models that are consistently closer to the true tree.

Scaling Performance

We now provide an overview of the computational performance of the OST algorithm on the synthetic censored datasets. We use the procedure described in Section 4.6.1 to create simulated data varying the number of observations n, the number of features p, and the percentage of censoring. We consider datasets of size $n \in [5000, 10000, 25000, 50000, 100000]$ and $p \in [10, 50, 100]$. We consider three percentages of censoring $[10\%, 50\%, 80\%]$ that

correspond to low, moderate, and high censoring respectively. We repeat the experiment for each combination of these parameters on 100 randomized datasets and report the average scaling performance [§] and the associated 95% confidence intervals. We perform cross validation using grid search to select the best parameters for each model and we report the computational time of the training procedure. Figure 4.11 illustrates our findings.

Across all experiments, the algorithm was able to complete in less than an hour. There was no significant change in the average running time across the different levels of censoring. However, the number of features, p, did have a substantial impact on the computational performance. For $p < 100$, we note that all instances were able to solve within 40 minutes. By contrast, for datasets where the number of covariates is restricted to 10, the average time to solve is less than 25 minutes even when the sample size is 100,000. Increasing the number of observations appears to affect the computational performance in a linear way while the number of features empirically shows an exponential effect.

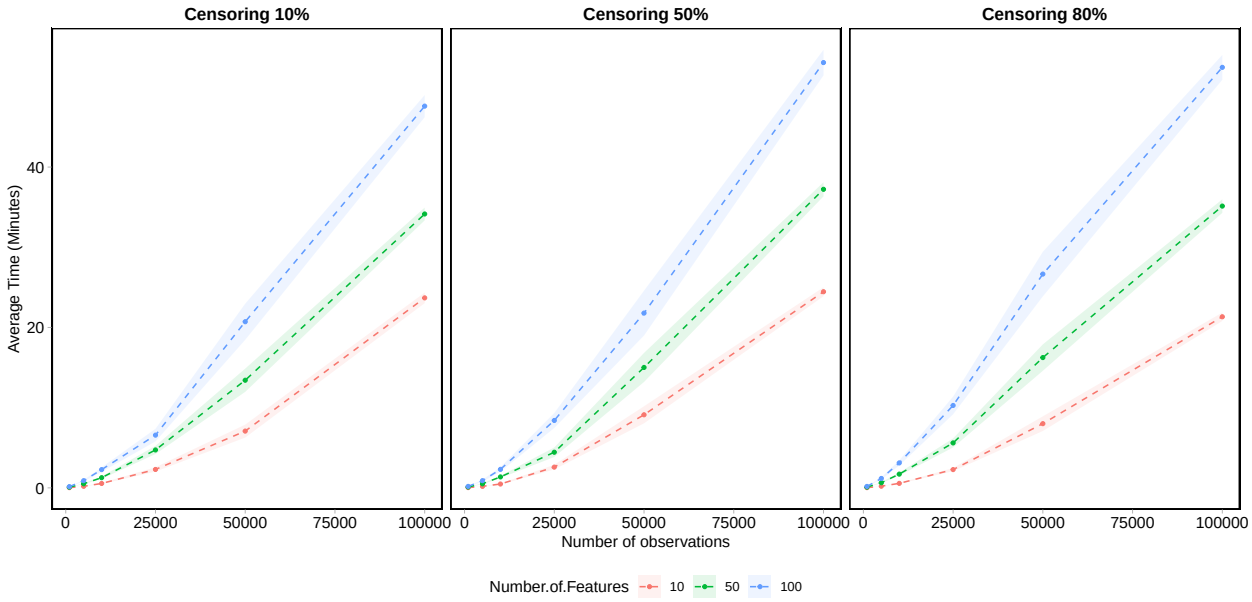


Figure 4.11: Average computational time for OST tree construction on synthetically generated datasets, with varying numbers of observations n and covariates p. The shaded region corresponds to the 95% confidence intervals.

[§] All experiments were conducted on four CPUs of type 2 socket Intel E5-2690 v4 2.6 GHz/35M Cache; 16GB of NUMA enabled memory were used per CPU.

4.7 Computational experiments with artificial censoring in real-world datasets

We compare the performance of the OST, **rpart** and **ctree** algorithms on 44 real-world datasets. The datasets used for this analysis were sourced from the UCI repository [102] and contained continuous outcome measures. In this section, we present the outcome of our analysis on non-survival specific data from a well-established resource for the ML community where we induce artificial censoring to test the algorithms' performance.

The selected datasets[¶] had sample sizes ranging from 63 observations to 100,000, and the maximum number of features considered was 383. We used the censoring procedure described in Section 4.6.1 to generate 9 versions of each dataset with different levels of censoring (0%,10%,...,80%). We then split each dataset into training and testing sets (50%) and compared the performance of the three tree algorithms on each dataset.

We applied the 5-fold cross-validation procedure described in Section 4.6.1 to select the depth and complexity of each tree, allowing tree depths of up to 7 (128 leaf nodes). Both the OST and **ctree** algorithms produced trees with over 100 leaf nodes in some of the largest datasets, while the largest **rpart** trees had only 77 nodes. The smaller size of the **rpart** trees indicates that larger models performed poorly in the cross-validation step.

On average, the OST models outperformed the other two algorithms across all 5 accuracy metrics. A summary of each algorithm's performance is given in Tables 4.4–4.5 and Figure 4.12, and aggregated results for each dataset are displayed in Table 4.6. The difference in performance was not statistically significant for the Cox ratios and Harrell's C scores, where all three algorithms had very similar average outcomes, but OST models did score significantly better than the other algorithms on the remaining metrics. OST models achieved the best score in 48-60% of the datasets tested, while the other algorithms each had undominated

[¶]We excluded the following types of datasets from our analysis: (1) datasets used for time series predictions (multiple observations of each individual); (2) datasets with unclear variable definitions; (3) datasets which required significant cleaning, pre-processing, or recording; (4) datasets with too many variables (p) to cross-validate all three algorithms in reasonable times. Dataset selection was independent of the analysis of model accuracy.

scores in 27-39% of datasets.

	Mean score			Paired T-Test H_1:	
	OST	**rpart**	**ctree**	$S_{OST} > S_{rpart}$	$S_{OST} > S_{ctree}$
Cox Ratio	**0.1118**	0.1091	0.1090	p=0.2288	p=0.2222
Harrell's C	**0.7873**	0.7866	0.7818	p=0.4355	p=0.1045
Uno's C	**0.6650**	0.6523	0.6441	**p=0.0288**	**p=0.0013**
Brier Point Ratio	**0.3841**	0.3627	0.3516	**p=0.0001**	**p< 10^{-5}**
Intg. Brier Ratio	**0.4451**	0.4262	0.4231	**p=0.0135**	**p=0.0055**

Table 4.4: Average scores for OST, **rpart** and **ctree** models on real-world datasets. The final columns show the one-sided p-values for paired t-tests comparing the outcome metrics on each dataset.

	OST	**rpart**	**ctree**
Cox Ratio	48.7	32.8	36.4
Harrell's C	57.3	30.8	33.6
Uno's C	59.3	27.3	34.1
Brier Point Ratio	56.6	33.3	38.4
Intg. Brier Ratio	57.6	30.6	33.6

Table 4.5: The percentage of datasets for which each algorithm was undominated by the other algorithms. Note that rows do not sum to 100, as several datasets were tied.

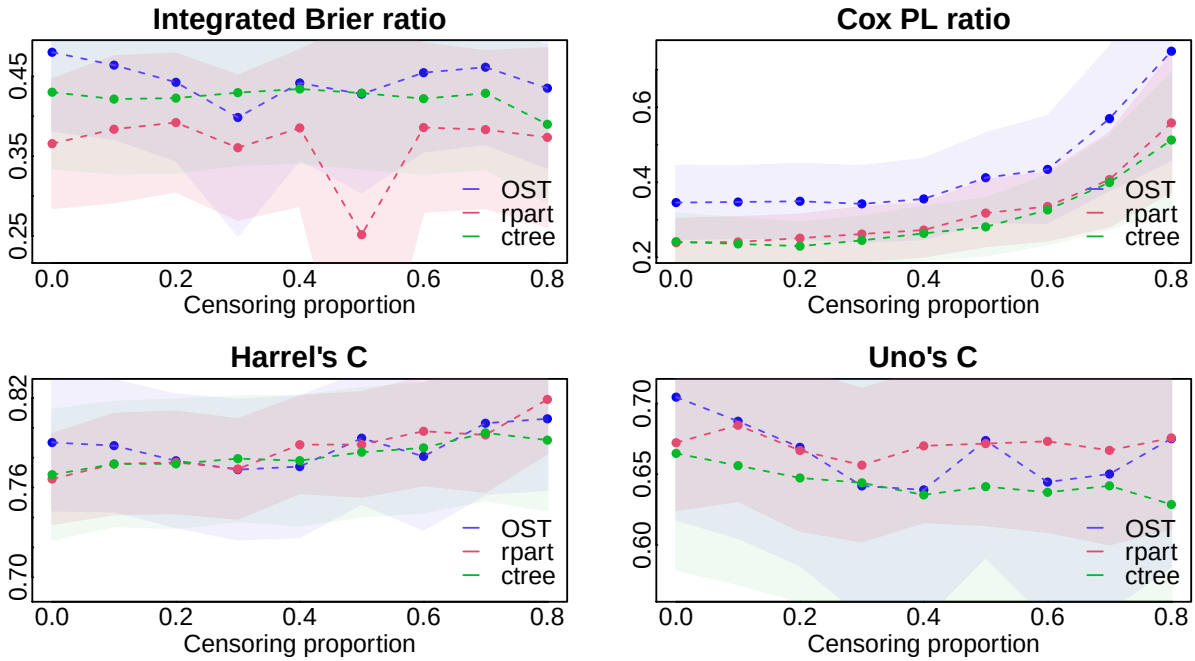


Figure 4.12: Average performance of survival tree models on real datasets with different levels of censoring. Confidence intervals are large due to the significant variability between datasets. However matched pairs analysis yields statistically significant results.

			Integrated Brier Score			Harrell's C Score			Uno's C Score			Cox Partial Likelihood			Brier Point Ratio		
Dataset	**n**	**p**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**	**OST**	**rpart**	**ctree**
3D Spatial Network [186]	100000	1	**0.44**	0.33	0.39	**0.82**	0.77	0.79	**0.79**	0.73	0.76	**0.05**	**0.05**	**0.05**	**0.48**	0.35	0.41
Airfoil Self Noise [102]	1503	4	**0.39**	0.33	0.35	**0.83**	0.78	0.78	**0.77**	0.7	0.71	**0.09**	**0.09**	0.08	**0.53**	0.42	0.46
Appliances Energy Prediction [61]	19735	25	**0.19**	0.18	0.18	**0.74**	0.73	**0.74**	**0.7**	0.69	**0.7**	**0.03**	**0.03**	**0.03**	**0.14**	0.13	0.12
Automobile [61]	164	23	0.03	**0.07**	0.06	0.53	**0.65**	0.61	0.08	**0.41**	0.27	0.01	**0.05**	0.03	0	**0.11**	**0.11**
Auto MPG [102]	398	7	0.55	**0.56**	0.55	0.85	**0.87**	**0.87**	**0.79**	0.78	0.77	0.19	0.2	**0.21**	0.58	**0.6**	0.58
Behavior Urban Traffic	135	16	0.18	**0.2**	0.18	0.66	**0.67**	0.64	0.37	**0.41**	0.33	0.08	**0.09**	0.08	0.13	**0.16**	0.14
Bike Sharing	17379	13	0.92	0.88	**0.93**	**0.98**	0.96	**0.98**	**0.96**	0.93	**0.96**	0.15	0.09	**0.2**	0.94	0.91	**0.95**
Blog Feedback [118]	52397	279	**0.39**	**0.39**	0.38	0.84	**0.85**	**0.85**	0.79	0.8	**0.82**	**0.03**	**0.03**	**0.03**	0.17	**0.18**	0.17
Buzz in Social Media [187]	100000	76	**0.77**	0.75	**0.77**	**0.92**	0.91	**0.92**	**0.91**	0.88	0.9	**0.13**	0.12	0.12	**0.76**	0.74	0.75
Cargo2000 [234]	3943	95	**1**	**1**	0.84	**1**	**1**	0.95	**1**	**1**	0.9	**0.21**	**0.21**	0.16	0.22	**0.23**	0.17
Communities Crime [283]	2215	145	0.64	0.65	**0.69**	0.89	0.89	**0.91**	0.81	0.83	**0.85**	0.17	0.17	**0.19**	0.68	0.7	**0.75**
Computer Hardware [102]	209	8	**0.69**	0.61	0.65	**0.86**	0.83	0.85	**0.74**	0.67	0.7	0.24	0.27	**0.29**	**0.73**	0.68	0.62
Concrete Slump [365]	103	6	0.07	**0.14**	0.03	0.62	**0.66**	0.56	0.27	**0.35**	0.14	0.04	**0.07**	0.02	0.11	**0.13**	0.05
Concrete Strength [364]	1030	7	**0.42**	0.41	0.4	**0.84**	0.83	0.82	0.74	0.74	**0.75**	0.11	**0.13**	0.12	0.5	0.47	**0.51**
CSM [5]	232	11	0.25	**0.32**	0.25	0.71	**0.76**	0.73	0.48	0.56	**0.57**	0.08	**0.11**	0.09	0.34	**0.42**	0.26
Cycle Power	9568	3	**0.73**	0.71	**0.73**	**0.92**	0.91	**0.92**	**0.89**	0.86	**0.89**	0.16	0.17	**0.18**	**0.75**	0.72	**0.75**
Electrical Stability [102]	10000	11	**0.4**	0.34	0.39	**0.82**	0.79	**0.82**	**0.79**	0.75	**0.79**	**0.08**	0.06	**0.08**	**0.44**	0.37	**0.44**
Energy efficiency 1 [335]	1296	7	**0.95**	0.9	0.9	**0.99**	0.97	0.98	**0.98**	0.95	0.93	**0.35**	0.3	0.31	-0.11	**-0.04**	-0.14
Energy efficiency 2 [335]	1296	7	**0.94**	0.9	0.9	**0.99**	0.97	0.97	**0.97**	0.95	0.96	**0.27**	0.13	0.21	-0.14	**-0.01**	-0.16
Faceboook Comments [183]	40949	52	**0.56**	**0.56**	0.55	0.88	0.88	**0.89**	0.84	0.84	**0.86**	**0.06**	**0.06**	**0.06**	-0.1	**-0.09**	-0.11
Faceboook Metrics [242]	500	6	**0.03**	0.02	0.02	0.55	**0.56**	0.53	0.1	**0.14**	0.05	**0.01**	**0.01**	**0.01**	**0.05**	**0.05**	0.02
Fires	517	11	**0**	**0**	**0**	**0.5**	**0.5**	**0.5**	**0**	**0**	**0**	**0**	**0**	**0**	**0.11**	**0.11**	**0.11**
GeoMusic [373]	1059	115	0.03	**0.06**	0.03	0.58	**0.61**	0.59	0.32	0.37	**0.38**	0.01	**0.02**	0.01	0.02	**0.06**	0.03
Insurance Company Benchmark [30]	5822	84	0.02	0.02	**0.03**	0.59	0.6	**0.62**	0.24	0.25	**0.27**	**0**	**0**	**0**	**0.33**	**0.33**	**0.33**
KEGG Directed [102]	53413	22	**0.81**	0.78	0.79	**0.96**	0.95	0.95	**0.94**	0.93	0.93	**0.11**	**0.11**	**0.11**	0.06	**0.07**	**0.07**
KEGG Undirected [102]	65554	25	**0.87**	0.81	0.86	0.96	0.95	**0.97**	**0.97**	0.94	0.96	0.14	0.16	**0.17**	**0.87**	0.81	0.85
Kernel Performance [16]	100000	13	**0.73**	0.67	0.69	**0.84**	0.8	0.82	**0.84**	0.79	0.81	**0.09**	0.07	0.08	**0.53**	0.43	0.47
Las Vegas Strip [241]	504	18	**0.02**	0	0.01	**0.54**	0.51	0.52	**0.12**	0.05	0.05	**0**	**0**	**0**	**0.02**	-0.01	0
Online News Popularity	39644	58	**0.05**	**0.05**	**0.05**	0.62	0.62	**0.63**	0.56	0.57	**0.58**	**0.01**	**0.01**	**0.01**	0.16	0.16	**0.17**
Online Video Characteristics [102]	68784	19	0.75	0.7	**0.76**	**0.92**	0.91	**0.92**	**0.91**	0.89	**0.91**	0.06	**0.15**	**0.15**	**0.76**	0.73	**0.76**
Optical Interconnection Network [3]	640	8	-0.08	**0.32**	0.27	**0.81**	0.79	**0.81**	0.72	0.67	**0.73**	**0.12**	**0.12**	0.11	**0.7**	0.68	0.69
Parkinson Telemonitoring [334]	5875	18	**0.59**	0.49	0.42	**0.85**	0.82	0.8	**0.8**	0.77	0.73	**0.14**	0.1	0.08	**0.67**	0.55	0.48
PM2.5-Beijing [212]	50387	12	**0.35**	0.32	0.34	**0.8**	0.79	**0.8**	**0.77**	0.75	0.76	0.05	0.05	**0.06**	**0.43**	0.4	0.42
Propulsion Plant [80]	11934	15	**0.64**	0.43	0.28	**0.88**	0.8	0.72	**0.87**	0.74	0.57	**0.11**	0.08	0.04	**0.7**	0.46	0.32
Protein [102]	45730	8	**0.3**	0.26	0.26	**0.75**	0.73	0.72	**0.72**	0.69	0.69	**0.04**	0.03	0.03	**0.32**	0.27	0.27
Real Estate 1 [366]	414	5	**0.44**	0.4	0.42	**0.83**	0.79	0.8	**0.67**	0.58	0.64	**0.18**	0.15	0.15	**0.59**	0.56	0.56
Real Estate 2 [366]	53500	383	**0.8**	0.75	0.76	0.55	**0.92**	0.88	**0.9**	0.87	0.88	0.02	**0.1**	0.05	**0.83**	0.76	0.78
Residential Building [278]	372	107	0.63	**0.64**	0.62	0.86	**0.87**	**0.87**	0.73	**0.76**	0.74	0.24	**0.27**	0.24	0.63	**0.68**	**0.68**
Servo [102]	167	3	**0.51**	0.4	0.39	**0.85**	0.73	0.69	**0.75**	0.5	0.41	**0.26**	0.16	0.14	**0.48**	0.29	0.24
Stock Market Istanbul [6]	536	6	0.11	0.12	**0.14**	0.68	**0.69**	**0.69**	0.47	**0.5**	**0.5**	0.04	**0.05**	**0.05**	**0.18**	0.17	0.2
Stock Portfolio [221]	63	11	**0.42**	0.26	0.29	**0.78**	0.73	0.74	**0.53**	0.46	0.43	**0.34**	0.26	0.27	**0.49**	0.4	0.35
Student Performance [239]	395	29	0.05	**0.08**	**0.08**	0.58	**0.61**	**0.61**	0.2	0.21	**0.26**	**0.02**	**0.02**	**0.02**	0.07	**0.11**	0.1
Wine Quality [81]	6497	10	0.16	0.16	**0.18**	0.73	0.74	**0.76**	0.63	0.65	**0.71**	0.02	0.02	**0.03**	**-0.04**	**-0.04**	-0.07
Yacht [102]	308	5	**0.84**	0.8	0.82	**0.94**	0.91	0.9	**0.85**	0.81	0.77	0.37	**0.41**	0.4	0.8	0.76	**0.82**

Table 4.6: Average scores for OST, **rpart**, **ctree** for each dataset across all levels of censoring.

4.8 Computational experiments with censored data from longitudinal studies and surveys

In this section, we focus on different aspects of algorithmic performance using three widely known surveys and longitudinal studies. In Section 4.8.1, we present results from the Wisconsin Longitudinal Study and highlight discrepancies in performance as we vary the mix of categorical and numerical features. In Section 4.8.2, we leverage the Health and Lifestyle survey to compare the algorithms on a large set of features. Finally, in Section 4.8.3, we showcase an application of the algorithm on heart disease using data from the monumental FHS.

4.8.1 The Wisconsin Longitudinal Study

In 1957, the Wisconsin Longitudinal Study (WLS) randomly sampled 10 317 Wisconsin high school graduates (one-third of all graduates) for a decades-long study, observing them until 2011 [165]. The aim of the study was to understand how factors such as social background, schooling, military service, labor market experiences, family characteristics and events, and social participation, may affect mortality and morbidity, family functioning, and health. We have included in our analysis data from all recorded participants for 518 variables that were collected either from the original respondents or their parents.

We removed from our dataset all features for which more than 50% of the values are missing. We imputed the missing values with the mean of each covariate for numerical features and the mode for categorical and binary variables. In total, we collect 317 categorical, 103 numerical, and 77 binary covariates. In each randomized experiment, we sampled between $[10, 15, 20, 25, 30]$ features from each category. Our goal was to observe the algorithms' performance as we vary the combination of different types of covariates.

Our results show minimal variability in performance as we change the number of numerical and binary features. However, all three methods show trends in the average performance scores for different numbers of categorical features, as shown in Figure 4.13. Specifically,

both OST and **rpart** algorithms show slight decreases in performance with larger feature sets, likely due to overfitting, while the **ctree** algorithm performs slightly better on larger feature sets.

Overall, OST clearly outperforms the other methods in terms of the Integrated Brier Score and the Cox PL ratio, and is on par with **rpart** in both concordance statistics. The **ctree** algorithm performs poorly relative to the other algorithms across all metrics.

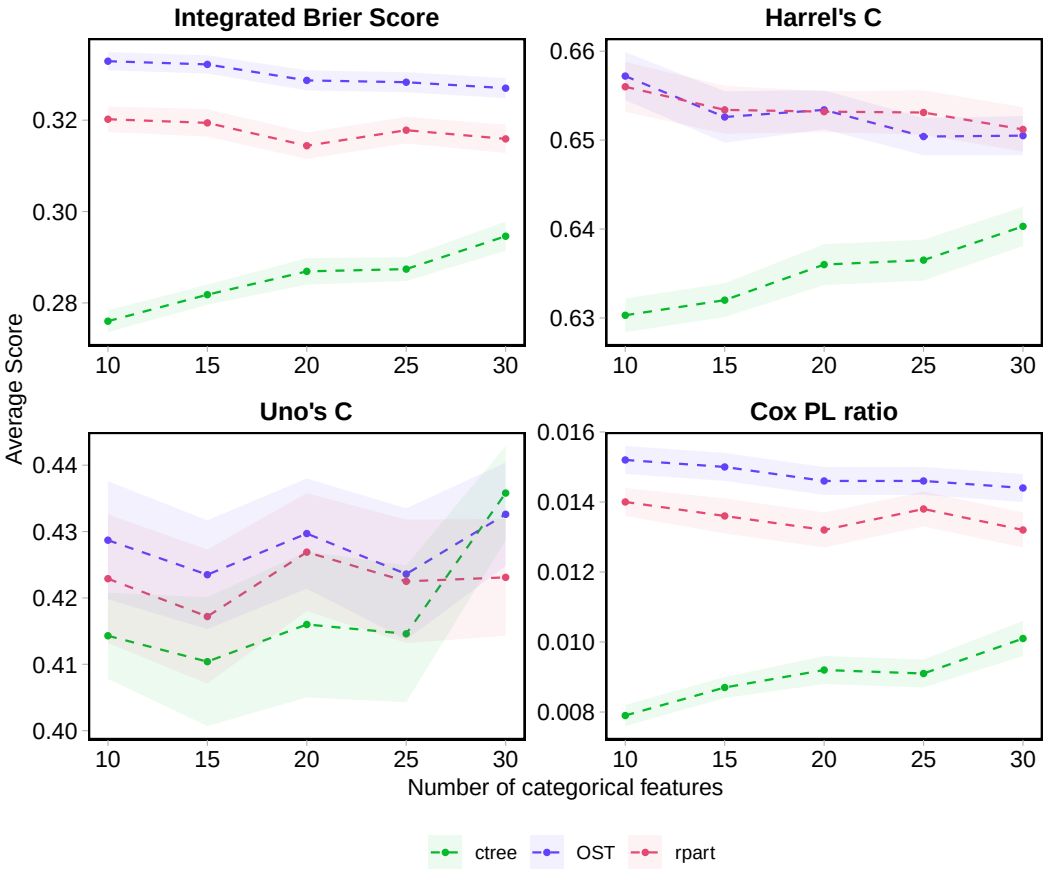


Figure 4.13: Average performance of survival tree models on subsets of features from the WLS dataset with varying numbers of categorical variables. The shaded regions represent 95% confidence intervals across 100 randomized experiments.

4.8.2 The Health and Lifestyle Survey

The first Health and Lifestyle Survey [83] was carried out in 1984-1985 on a random sample of the population of England, Scotland and Wales. Its objective was to help researchers understand the impact of self-reported health, attitudes to health, and beliefs about causes of disease in relation to measurements of health and lifestyle in adults from different parts of Great Britain. In our numerical experiments, the outcome of interest is the age of death of study participants as observed by follow-up studies until 2009. Our dataset includes 9003 individuals and 112 binary features. We conducted 100 randomized experiments to train each tree algorithm.

Method	IBR	95% CI	Cox PL	95% CI	Harrell's C	95% CI	Uno's C	95% CI
OST	0.6114	(0.6075, 0.6153)	0.0125	(0.0119, 0.0131)	0.6211	(0.618, 0.6241)	0.3987	(0.3912, 0.4062)
ctree	0.6056	(0.6016, 0.6095)	0.0107	(0.0102, 0.0112)	0.6113	(0.6076, 0.6151)	0.4098	(0.4026, 0.4171)
rpart	0.6105	(0.6068, 0.6143)	0.0124	(0.0117, 0.0131)	0.6185	(0.6152, 0.6218)	0.395	(0.3854, 0.4046)

Table 4.7: Average scores for OST, **rpart**, **ctree** models on the HALS dataset. For each metric, we report the 95% confidence intervals in 100 randomized experiments.

Table 4.7 outlines the results of our analysis on the HALS dataset. The OST algorithm outperforms the other methods in all metrics other than the Uno's C metric. Specifically, OST is associated with an average Integrated Brier Score of 0.6114 compared to 0.6056 and 0.6105 for **ctree** and **rpart** respectively. In terms of the Cox PL ratio, OST offers an 8% improvement over the next best method (**rpart**) with an average score of 0.0125. With respect to the Harrell's C metric, OST average Harrell's C metric is 0.6211. **ctree** and **rpart** scored 0.6113 and 0.6185 respectively. Contrary to the other measures of performance, **ctree** achieves the best score in this series of experiments with an average metric of 0.4098 with a 0.0111 margin fromOST. Our findings from this study are in line with the results in Sections 4.6 and 4.7.

4.8.3 The Framingham Heart Study

In this section, we focus on the interpretation of the tree models using data from the FHS. Analysis of the FHS successfully identified the common factors or characteristics

that contribute to Coronary Heart Disease (CHD) using the Cox regression model [87]. In our survival tree model, we include all participants in the study from the original cohort (1948-2014) and the offspring cohort (1971-2014) who were diagnosed with CHD. The event of interest in this model is the occurrence of a myocardial infarction or stroke. All 2296 patients were followed for a period of at least 10 years after their first diagnosis of CHD and observations are marked as censored if no event was observed while the patient was under observation.

We applied our algorithm to the primary variables that have been used in the established 10-year Hard CHD Risk Calculator and the Cardiovascular Risk Calculator [255, 92]. For each participant who was diagnosed with CHD, we include the following covariates in our training dataset: gender, smoking status (smoke), SBP, Diastolic Blood Pressure (DBP), use of Anti-Hypertensive (AHT) medication, BMI, and T2DM (diabetes). We did not include cholesterol levels in our analysis because these variables are highly correlated with the use of lipid lowering treatment and a high proportion of the sample population did not have sufficient data to account for this interaction.

In Figure 4.14 we illustrate the output of our algorithm on the FHS dataset. Every node of the tree provides the following information:

- The node number.
- Number of observations classified into the node.
- Proportion of the node population which has been censored.
- A plot of survival probability vs. time. In this example, the x-axis represents age and the y-axis gives the Kaplan-Meier estimate for the probability of experiencing no adverse events.
- Color-coded survival curves to describe the different sub-populations. In each node, the blue curves describe the individuals classified into that node.
- In internal (parent) nodes, the orange/green curves describe the sub-populations that

are split into the left/right child node. After each split, the sub-population with higher likelihood of survival goes into the left node.

- In leaf nodes, the red curve shows the average survival curve for the entire tree. This facilitates easy comparisons between the survival of a specific node and the rest of the population.

The splits illustrated in Figure 4.14 include known risk factors for heart disease and are consistent with well-established medical guidelines. The algorithm identified a BMI threshold of 25 as the first split (node 1), which is in accordance with the NIH BMI ranges that classify an individual as overweight if his/her BMI is greater than or equal to 25. Multiple splits indicated a higher risk of heart attack or stroke in patients who smoke (nodes 2, 6). The group with the highest risk of an adverse event was overweight patients with diabetes (node 9).

Figures 4.15 and 4.16 illustrate the output of the **ctree** and **rpart** algorithms applied to the same FHS population. The **rpart** model has a single split (BMI), while the **ctree** model contains the same variables as the OST output. The Brier scores for each model are 0.0486 (OST), 0.0249 (**rpart**) and 0.0467 (**ctree**).

The discrepancy in the Brier scores for the OST and **ctree** models is due to slight differences in the threshold and position of certain splits. For example, both methods identify that BMI is the most appropriate variable for the first split, but the BMI threshold differs. The **ctree** model sets the splitting threshold to 24.117, which is the locally optimal value for the split when building the tree greedily (the same threshold is used in the **rpart** model). By contrast, the OST algorithm selects a threshold of 25.031. This example demonstrates how the OST algorithm's efforts to find a globally optimal solution differ from the results of locally optimal splits.

A second difference between the tree models is the order of the smoking and diabetes splits within the overweight population. The **ctree** model splits on smoking first, since this split has the most significant p-value of the variables at node 5 in the **ctree** tree. The algorithm also recognizes that diabetes is a risk factor and incorporates this in the subsequent split.

Since greedy approaches like **ctree** do not reevaluate the spits once they have been decided, the algorithm does not recognize that the overall quality of the tree can be improved by reversing the order of these splits. This discrepancy in two otherwise similar trees highlights the advantages of the more sophisticated optimization conducted byOST.

Leaf Nodes	Algorithm	IBR	Cox PL	Harrell's C	Uno's C
2	**ctree**	0.393	0.0016	0.5368	0.5936
	OST	0.3932	0.0021	0.5429	0.5894
	rpart	0.3914	0.0021	0.5429	0.5894
4	**ctree**	0.3996	0.0029	0.5588	0.585
	OST	0.4028	0.0031	0.5615	0.5861
	rpart	0.3881	0.0027	0.546	0.5714
8	**ctree**	0.4006	0.0064	0.5645	0.5897
	OST	0.4041	0.0066	0.567	0.5897
	rpart	0.4031	0.0072	0.5686	0.5871
16	**ctree**	0.3851	0.0069	0.5608	0.5738
	OST	0.3878	0.0098	0.5713	0.5858
	rpart	0.3566	0.0052	0.5488	0.5577

Table 4.8: Average scores in 100 randomized experiments for OST, **rpart**, **ctree** models on the FHS dataset for different values of the maximum depth parameter.

We performed another series of experiments to systematically study the relation between the size of the tree and the model's quantitative performance. In this setting, we measure interpretability as a function of the number of leaf nodes at the model. We train fully saturated trees, by setting the complexity parameter to zero, for different values of the maximum depth parameter. Thus, each algorithm results in the best performing tree given two, four, eight, and 16 leaf nodes. Table 4.8 presents the results of our analysis for the three algorithms considered with respect to the IBR, Cox PL, Harrell's C, and Uno's C metrics. We do not report confidence intervals as the same tree was recovered for each set of parameters across all algorithms. A lower number of leaf nodes is arguably associated with higher model interpretability as patient profiles can be characterized with less features. OST results to the best performing model across all tree sizes for the IBR and Uno's C metrics and in the majority of cases for the Cox PL and Harrell's C metrics. Our findings indicate that OST is

able to recover more accurate data partitions when we restrict the model to a smaller number of splits.

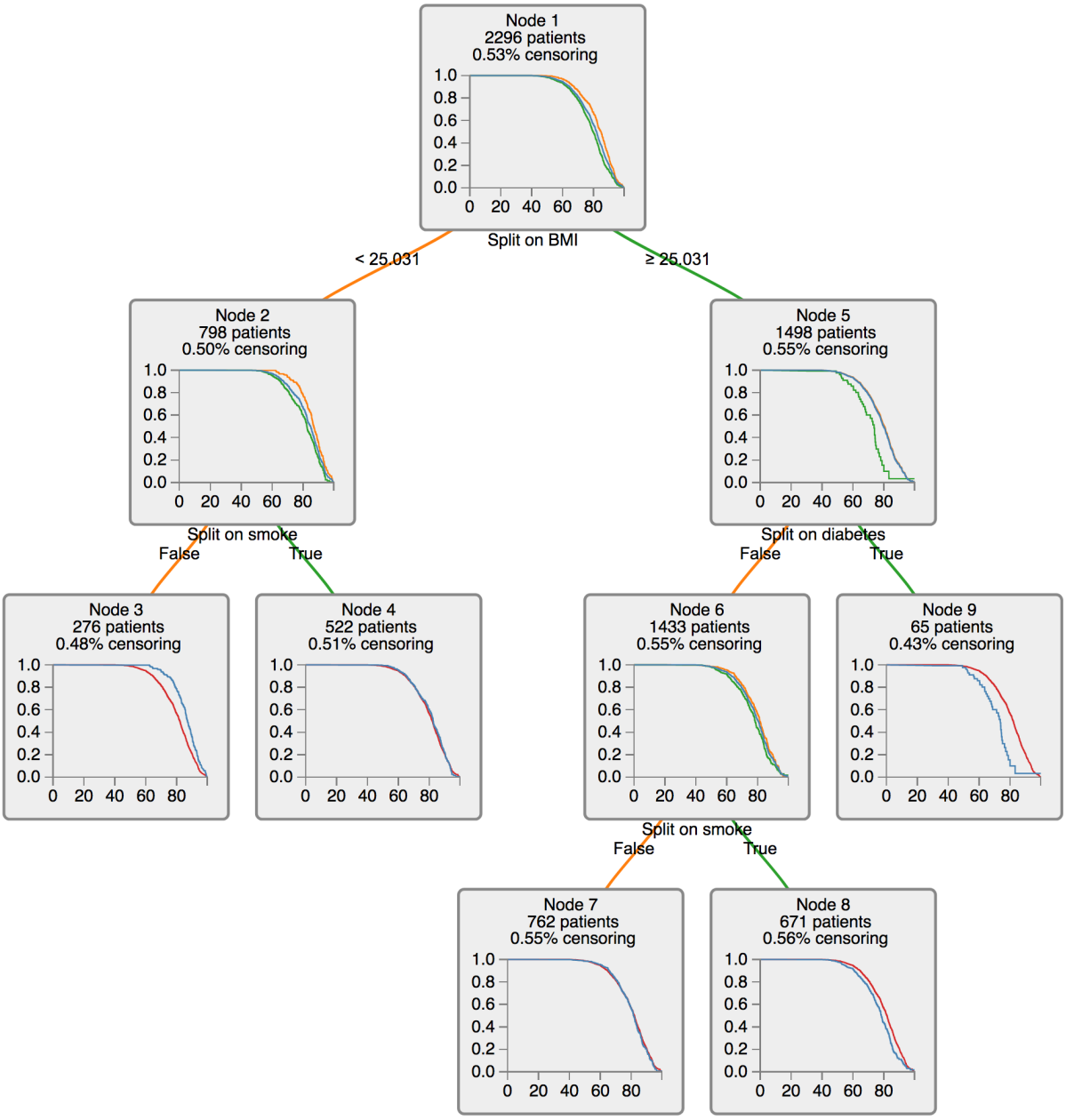


Figure 4.14: An illustration of Optimal Survival Trees for chd patients in the FHS.

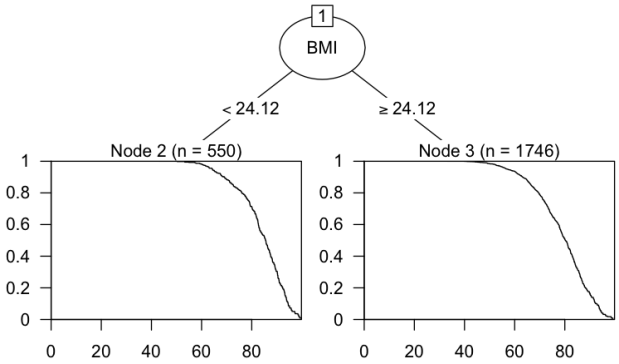


Figure 4.15: Illustration of the **rpart** output for chd patients in the FHS.

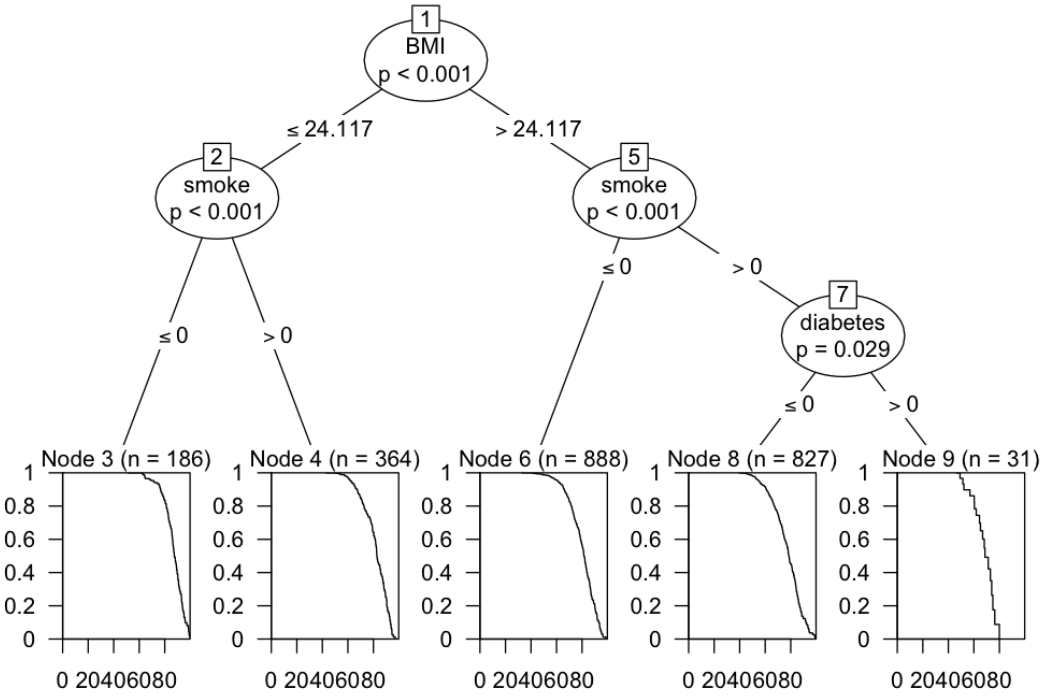


Figure 4.16: Illustration of the **ctree** output for chd patients in the FHS.

4.9 Conclusions

In this chapter, we have extended the Optimal Trees framework to generate interpretable models for censored data. We have also introduced a new accuracy metric, the Kaplan-Meier Area Ratio, which provides an effective way to measure the predictive power of survival tree models in simulations.

The OST algorithm improves on the performance of existing algorithms in terms of both classification and predictive accuracy. Our results in simulations indicate that the OST models improve consistently with increasing sample size, whereas existing algorithms are prone to overfitting in larger datasets. This is particularly important, given that the volume of medical data available for research is likely to increase significantly over the coming years.

Part II

Prescriptive & Predictive Analytics for Clinical Data

Chapter 5

Personalized Treatment for Coronary Artery Disease Patients: A Machine Learning Approach

Current clinical practice guidelines for managing CAD account for general cardiovascular risk factors. However, they do not present a framework that considers personalized patient-specific characteristics. Using the electronic health records of 21,460 patients, we created data-driven models for personalized CAD management that significantly improve health outcomes relative to the standard of care. We develop binary classifiers to detect whether a patient will experience an adverse event due to CAD within a 10-year time frame. Combining the patients' medical history and clinical examination results, we achieve 81.5% AUC. For each treatment, we also create a series of regression models that are based on different supervised machine learning algorithms. We are able to estimate with average $R^2 = 0.801$ the outcome of interest; the time from diagnosis to an adverse event. Leveraging combinations of these models, we present `ML4CAD`, a novel personalized prescriptive algorithm. Considering the recommendations of multiple predictive models at once, the goal of `ML4CAD` is to identify for every patient the therapy with the best expected TAE using a voting mechanism. We evaluate its performance by measuring the prescription effectiveness and robustness under

alternative ground truths. We show that our methodology improves the expected TAE upon the current baseline by 24.11%, increasing it from 4.56 to 5.66 years. The algorithm performs particularly well for the male (24.3% improvement) and Hispanic (58.41% improvement) subpopulations. Finally, we create an interactive interface, providing physicians with an intuitive, accurate, readily implementable, and effective tool.

5.1 Introduction

The clinical condition of CAD also referred to as ischemic heart disease, is present when a patient presents one or more symptoms or complications from an inadequate blood supply to the myocardium [135]. This is most commonly attributed to the obstruction of the epicardial coronary arteries due to atherosclerosis [296]. CAD remains the number one cause of death in the United States, accounting for over 360,000 annual casualties [4]. CAD is mostly prevalent in older patients (above the age of 50 years) in the form of a chronic condition which requires a principal intervention and subsequent systematic medical therapy and monitoring [135]. The primary care of patients with CAD includes ascertainment of the diagnosis and its severity (with non-invasive and/or invasive imaging), control of symptoms, and therapies to improve survival [158]. The mainstay of treatment is medical therapy. The latter may or may not be combined with coronary revascularization (either Coronary Artery Bypass Graft (CABG) surgery or Percutaneous Coronary Intervention (PCI)) in an effort to slow the progress of the disease and relieve its symptoms. Considering the magnitude and the repercussions of CAD, the importance of medical therapy to reduce its symptoms and prolong life expectancy is being increasingly recognized [308].

There has been growing interest in using clinical evidence to understand the effects of treatments in patients with CAD. Nowadays, there are numerous evidence-based clinical guidelines for CAD management [125, 124] and angiographic tools for grading its complexity, such as the SYNTAX Score [311, 317]. However, it is not clear how to choose among different types of available therapies (pharmacological, percutaneous intervention, and surgery) to maximize effectiveness at an individual level. This is likely due to the multitude of parameters

that define the form of the disease for each patient and the uncertainty that lies behind an individual patient's response to a particular treatment [351]. One of the greatest challenges in developing evidence-based guidelines applicable to large populations is paucity of information about special subpopulations with unique characteristics. This is attributed to the absence of specialized clinical trials [124].

Considering the challenges and the significance of CAD, a personalization approach may greatly impact the effective management of the disease. Personalization is the problem of identifying the best treatment option for a given instance, i.e., a display add [375] or medical therapy [206]. There are two main challenges for designing personalized prescriptions for a patient as a function of the features recorded in the data:

1. While the outcome of the administered treatment for each patient is observed, the counterfactual outcomes are unknown. That is, the outcomes that would have occurred had another treatment been administered. Note that if this information were known, the prescription problem would reduce to a multi-class classification problem. Thus, the counterfactual outcomes need to be inferred.

2. In the data, there is an inherent bias that needs to be taken into account. The nature of data from EHR is observational as opposed to data from randomized trials. In a randomized trial setting, patients are randomly assigned different treatments, while in an observational setting, the assignment of treatments potentially depends on features of the population.

5.1.1 Literature Review

Our objective is to solve the problem of prescribing the best option among a set of predefined treatments to a given patient as a function of the samples' features. We are provided with observational data of the form $\{(\mathbf{x}_i, y_i, z_i)\}_{i=1}^{n}$, comprising n observations. Each data point $\{(\mathbf{x}_i, y_i, z_i)\}$ is characterized by features $\mathbf{x}_i \in \mathbb{R}^p$, the prescribed treatment $z_i \in [T] = \{1, \ldots, T\}$, and the corresponding outcome $y_i \in \mathbb{R}$. We denote $y(1), \ldots, y(T)$ the T "possible outcomes" resulting from assigning each of the T treatments respectively.

A similar question has been studied in the causal inference literature. In this setting, the main focus lies on observational studies to identify causal relationships between an intervention and outcomes in a particular population [264]. Introduced by Neyman and popularized by Rubin, the Potential Outcomes Framework uses a probabilistic assignment mechanism to mathematically describe how treatments are given to patients. It also accounts for a potential dependence on background variables and the potential outcomes themselves [298, 10]. More specifically, it focuses on the case where $S = \{C, T\}$ (treatment and control). For each patient i, the potential outcome $y_i(T)$ is the experienced outcome if exposed to treatment T. The causal effect of T compared to C is then computed as $\delta_i := y_i(T) - y_i(C)$. Thus, causal effects are solely defined for one treatment relative to another and only if the individual could have been reasonably exposed to both. The fundamental problem of causal inference is that $(y_i(T), y_i(C))$ are not jointly observable. That is, only one observed response is present depending on the treatment assignment. As a result, [294] focus on the average treatment effect for a completely randomized experiment. This scenario considers the difference of the sample means for the units receiving the treatment and control.

$$\text{ATE} = \frac{1}{n_T} \sum_{j:z_j=T} y_j(T) - \frac{1}{n_C} \sum_{j:z_j=C} y_j(C). \tag{5.1}$$

However, in observational studies, treatment assignment is not independent of the potential outcomes. Thus, further analysis is required to account for latent differences between the treated and control groups on the basis of observed covariates X (inverse probability weighting, propensity score matching, nonparemetric regression, etc.) [293].

Causal effect approaches do not provide personalized estimations of the treatment effect for each unit since they focus on the aggregate population level. A personalized prescription methodology would require a quantification of the impact of each regimen for every individual in isolation. This is the essence of the personalized medicine field [155]: identifying the optimal therapy for a particular set of phenotypic and genetic patient characteristics. ML algorithms are expected to enable the utilization of rich datasets. They could provide improved solutions for patients by learning the outcome function for each treatment. They will particularly

impact those that belong to very specific subgroups and respond in unusual ways to the available treatments [134].

A common approach in the literature to leverage these algorithms is called "Regress and Compare". It identifies the expected effect $y_i(z_i)$ of treatment $z_i \in [T]$ for each patient i based on the covariates $\mathbf{x}_i$ and consequently prescribes the regimen with the best potential impact;

$$\max_{z_i \in [T]} y_i(z_i|\mathbf{x}_i) \quad \forall i \in [n],$$

where [n] is the set of patients in the sample. The "Regress and Compare" methodology follows this paradigm, choosing a treatment by maximizing among T regression functions. A different regression model is fitted to the subset of the data that received each treatment. It subsequently uses them to predict outcomes and pick the one with the more optimistic prediction [326]. This approach has been historically followed by several authors in clinical research [123], and more recently by researchers in statistics [271] and operations research [38]. The online version of this problem, called the contextual bandit problem, has been studied by several authors [210, 145] in the multi-armed bandit literature [141]. Even though it is intuitive, this methodology is subject to prediction errors and potential biases of a single method.

In the field of precision medicine, [38], first, introduced a personalized prescriptive algorithm for diabetes management that harnesses the power of EHR. It was based on a "Regress and Compare" k-NN approach. This methodology yielded substantial improvements in patient outcomes relative to the standard of care. Moreover, it provided physicians with a prototyped dashboard visualizing the algorithm's recommendations. Their work showed that tailored approaches to particular diseases coupled with medical expertise provide the medical community with highly accurate and effective tools that will ameliorate patient treatment. Even though this effort provided promising results, the k-NN approach is not applicable to diseases where the effects of a treatment are not promptly observable. The same individual was tracked via multiple visits in the hospital system. Thus, the algorithm suggested alterations in the medication only when there was significant reduction on the

expected Hemoglobin A1c measurement. The physician could measure the effectiveness of a treatment by ordering a blood test in the near future. On the contrary, at the CAD setting the adverse effects of the disease are observed in the span of ten years from the time of diagnosis.

Focusing mostly on the personalization and not the prediction objective, [182] proposes a recursive partitioning methodology for personalization using observational data. This new algorithm is tailored to optimize a personalization impurity measure. As a result, it hardly places any emphasis on the predictive task. Therefore, it raises questions regarding the accuracy of the suggested treatment effect. [35] modify the latter's objective to account for the prediction error, and use the methodology of [34, 27] to design near optimal trees, improving performance substantially. Continuing on tree based approaches, [15], and [346] also use a recursive splitting procedure of the feature space to construct causal trees and causal forests respectively. They estimate the causal effect of a treatment for a given sample, or construct confidence intervals for the treatment effects. However, they do not infer explicit prescriptions or recommendations. Also, causal trees (or forests) are designed exclusively for studies comparing binary treatments.

In the cardiovascular field, the benefit of ML based personalization methods has been recognized and is expected to play a significant role in facilitating precision cardiovascular medicine [194]. Nevertheless, in the case of CAD, personalization approaches have been primarily focused on utilizing genomic information [19], and not on employing EHR and ML. Since 2014, the US mandated all public and private healthcare providers to adopt and demonstrate "meaningful use" of EHR to maintain their existing Medicaid and Medicare reimbursement levels. This decision contributed to the creation of clinical databases that contain in-depth information for many patients. These data can be leveraged using ML to construct models and algorithms that can learn from and make predictions on data [292].

One of the greatest challenges of EHR is the presence of right censored patients [195, 174], which arises when a patient disappears from the database after diagnosis and treatment of the disease. Traditional approaches to address right censoring, including the Cox proportional

hazards model [84] or the Weibull Regression [172], do not allow for time-varying effects of covariates. Their weaknesses are especially relevant to datasets that span over long periods of time, providing results that are not validated by the medical literature (e.g. positive correlation between a patient's BMI and his/her expected time to adverse event).

Our work addresses most of the challenges encountered in the personalized prescription setting that uses EHR, including counterfactual estimation and censoring.

5.1.2 Contributions

In this chapter, our objective is to find the best primary treatment for a CAD patient to maximize the TAE (myocardial infarction or stroke). We consider the latter as the primary endpoint of our models. Our dataset includes CAD patients who were administered treatment through the BMC, a private, not-for-profit, 487-bed, academic medical center located in Boston, MA, USA. We retrieved each patient's medical history, the primary treatment followed after diagnosis, and the most recent clinical examination results to the time of diagnosis. We considered five primary prescription approaches available for each patient. We developed predictive and prescriptive algorithms that provide personalized treatment recommendations. We propose a new prescription algorithm to assign the regimen with the best predicted outcome leveraging simultaneously multiple regression models. The effect of the prescriptive algorithm was evaluated by comparing the expected TAE under our recommended therapy with the observed outcome prescribed by physicians at the medical center. Successful treatment recommendations increase the TAE. On the contrary, ineffective prescriptions negatively impact the patient, decreasing the time from diagnosis to a myocardial infarction or stroke. We tested the robustness and effectiveness of our methodology. We considered different ground truths regarding the treatment effect of a given therapy to a patient. The ground truths comprise the standard of care as well as combinations or individual predictions from ML models. The main contributions of this chapter are:

1. A new methodology to treat right censored patients that utilizes a k-NN approach to estimate the true survival time from real-world data.

2. Interpretable and accurate binary classification and regression models that predict the risk and timing of a potential adverse event for CAD patients. We selected a diverse set of well-established supervised machine learning algorithms for these tasks.

3. The first prescriptive methodology that utilizes EHR to provide treatment recommendations for CAD. Our algorithm, `ML4CAD`, combines multiple state-of-the-art ML regression models with clinical expertise at once. In particular, it uses a voting scheme to suggest personalized treatments based on individual data.

4. A novel evaluation framework to measure the out-of-sample performance of prescriptive algorithms. It compares counterfactual outcomes for multiple treatments under various ground truths. Thus, we assess both the accuracy, effectiveness, and robustness of our prescriptive methodology. Using this evaluation mechanism, we demonstrate that `ML4CAD` improves upon the standard of care. Its expected benefit was validated by all considered ground truths and TAE estimation models.

5. An online application where physicians can test the performance of the algorithm in real time bridging the gap with the clinical practice.

The structure of the chapter is as follows. In Section 5.2, we describe the data used to train and validate our methods. In Section 5.3, we outline the method used to handle the challenge of censoring. Section 5.4 describes the methods and results of the binary classification models, and similarly Section 5.5 refers to regression. In Section 5.6, we present the personalized prescription algorithm and its evaluation framework. Results under different ground truths and recommendation policies are compared in Section 5.7. We conclude our work in Section 5.8.

5.2 Data

In this section, we provide detailed information about the dataset under consideration. We outline the patient inclusion criteria as well as a description of the covariates included in

the ML models. Subsequently, we refer to the treatments identified from the EHR and their aggregation as features for our algorithms. We also present the missing data imputation procedure that was followed.

5.2.1 Sample Population Description

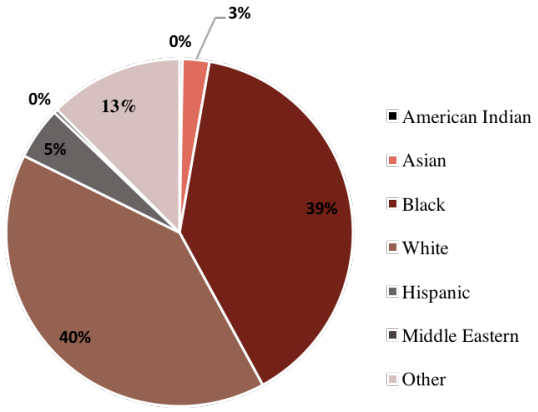
Through a partnership with the BMC we obtained EHR for 1.1 million patients from 1982 to 2016. In this dataset, 21,460 patients met, at least, one of the following inclusion criteria:

- **Population 1:** Patients associated with CAD risk of at least 10% based on the Framingham Heart Study formula [354] who were prescribed antihypertensive medication as primary treatment. The 10% threshold was selected since it is considered one of the primary indications for physicians to prescribe CAD treatment to their patients [353];
- **Population 2:** Patients who were administered at least one CABG surgery or, at least, one PCI and were prescribed antihypertensive medication;

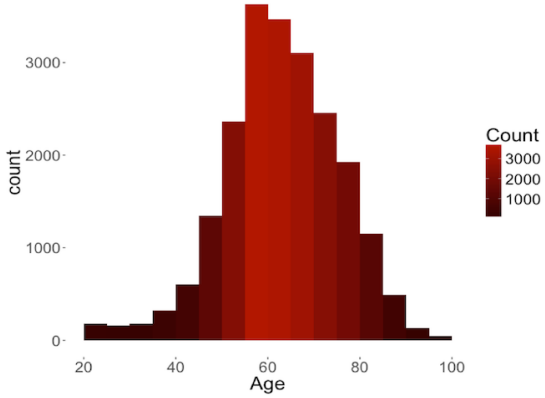
We used the conditions outlined above due to the absence of a systematic CAD diagnosis code in the system [329]. Note that the two inclusion criteria are mutually exclusive as a primary CAD prescription could either involve exclusively pharmacological treatment or a drug combination with a CABG surgery or a PCI. All patient EHR were processed to identify the time t_0 that corresponds to the point of initial diagnosis prior to any coronary revascularization. We reverted to the record that corresponds to this time to create the patient features X. Thus, we avoided the inclusion of two populations whose conditions are fundamentally dissimilar. Our sample comprised recently diagnosed CAD patients, similar to the ones physicians encounter in practice. We identified, using the totality of the EHR after the time t_0, the main therapy prescribed to each patient while being in the system. Notice that every member of the sample population was medicated with antihypertensive drugs. If in addition to the pharmacological therapy they were administered surgical or percutaneous interventions, we set the latter as the main treatment administered by the hospital.

BMC patients come predominantly from underprivileged socioeconomic backgrounds. As

a result, in most cases they do not have the financial capability to support alternative health providers. They need to appeal to the BMC for healthcare services for the majority of their medical needs. Thus, most of their EHR are concentrated in the same database, allowing us to follow the trajectory of each patient's health from a single source. The ethnicity and age distributions of the population are depicted in Figures 5.1a and 5.1b, respectively.



(a) Ethnicity distribution.



(b) Age distribution.

Figure 5.1: Demographic Characteristics of the population

We excluded all patients whose diagnosis date was identical to their last observation in the healthcare system. Moreover, we removed from the data those whose cause of death was observed but not related to heart disease (e.g., cancer non-survivors). We retrieved for each patient a set of values that describe their demographics, medical therapy, and clinical characteristics at the time of diagnosis t_0 (Table 5.1). We used ICD-9, CPT, and hospital

specific codes to identify the corresponding records as well as lab test results for particular measurements (i.e., Low-Density Lipoprotein (LDL) or HDL levels). Along with demographic information, we included features that are considered risk factors for heart disease, according to the medical literature. We excluded all covariates whose values were not known for at least 50% of the patients in the dataset. We identified an adverse event (myocardial infarction or stroke) attributable to CAD and recorded the date of occurrence. This way, we define the time between a diagnosis and an adverse event. In case the patient disappeared from the EHR before the lapse of 10 years after diagnosis, we recorded that the patient was right censored. We did not take into account the severity of the adverse event in our evaluation.

5.2.2 Treatment Options

We considered five primary options for each patient, shown in Table 5.2. These options are mutually exclusive and thus each patient received only one of them as primary treatment. CAD is a chronic disease whose management may differ across time. However, we noticed that a certain pattern was followed for the vast majority of the patients throughout their presence in the academic medical center. Coronary revascularization is a major operation and thus we distinguish CABG and PCI as separate treatment categories. In agreement with the general guidelines of the American Heart Association (AHA) for the management of Stable Ischemic Heart Disease [124], most of the patients are prescribed blocking medication to treat hypertension and statins as a lipid lowering treatment. Therefore, we chose combinations of those two lines of therapy as primary prescription options. Nevertheless, the pharmaceutical treatment for a CAD patient may include not only blockers, but also a more complicated combination of drugs, depicted in Table 5.3 under "Treatment". As the set of all those combinations is too wide, we considered only the most common prescription options. We did not account for aspirin (ASA) since all patients were prescribed this line of therapy.

Note that we did not consider ACE inhibitors as a prescription option because they usually accompany another type of antihypertensive medication for CAD patients [285]. They are prescribed in combination to blockers or as a substitute of the latter in cases where a patient

Category	Variable Name	% NA
Demographics	Age	0.0%
	Gender	0.0%
	Ethnicity	0.0%
	Language	0.0%
	Marital Status	15.3%
	Ethnicity	0.0%
Treatment	ACE inhibitors	0.0%
	Adrenergic Receptors	0.0%
	Angiotensin Agonists	0.0%
	Antiarrhythmics	0.0%
	Blockers (beta, alpha, etc.)	0.0%
	CABG	0.0%
	Cardiac Glycosides	0.0%
	Diuretics	0.0%
	Lipid Lowering medication	0.0%
	Muscle relaxants	0.0%
	Nitrates	0.0%
	Other antihypertensive	0.0%
	PCI	0.0%
	Phosphodiesterase inhibitors	0.0%
	Statins	0.0%
Family History	Diabetes	26.8%
	Hypertension	23.9%
Medical Records	BMI	16.6%
	LDL Cholesterol	21.4%
	HDL Cholesterol	21.3%
	DBP	7.1%
	SBP	7.1%
	Diabetes	0.5%
Observed Behavior	Smoking	23.6%
	Time observed in the EHR database	0.0%

Table 5.1: Patient characteristics considered. The column "% NA" indicates the percent of missing data that was present in the original dataset

has some prohibitive medical condition to the former. Thus, the majority of the population that belongs in the "Drugs 2 and 3" categories are effectively under ACE inhibitors. The latter drug class was administered in less than 50% of the sample population. As a result, a separate pharmacological treatment option would thin the training sets presented in the following sections significantly.

Option	**Description**	**Num. of patients**	**%**
CABG	Coronary Artery Bypass Graft Surgery with pharmaceutical treatment	1854	8.64%
PCI	Percutaneous Coronary Intervention with pharmaceutical treatment	4042	18.85%
Drugs 1	Pharmaceutical treatment including blockers and statins	6833	31.86%
Drugs 2	Pharmaceutical treatment including blockers and excluding statins	3767	17.56%
Drugs 3	Pharmaceutical treatment excluding blockers (potentially including statins)	4964	23.09%

Table 5.2: The Prescription Options.

5.2.3 Handling of missing values

We collected each patient's medical records (lab test results and clinical measurements) associated with the most recent clinical examination before or at the time of diagnosis. We omitted from our analysis any risk factors whose missing values proportion was higher than 50% (i.e., ejection fraction, ECG measurements). Table 5.1 shows the percent of missing data that was present in the original dataset. Note that all demographic variables other than Marital Status were consistently recorded for all patients. A treatment was considered to be

Treatment Name	Proportion
ACE inhibitors	46.12%
Adrenergic Receptors	6.38%
Angiotensin Agonists	13.62%
Antiarrhythmics	13.65%
Blockers (beta, alpha, etc.)	68.03%
CABG	7.01%
Cardiac Glycosides	2.45%
Diuretics	47.90%
Lipid Lowering medication	5.29%
Muscle relaxants	4.81%
Nitrates	77.02%
Other antihypertensive	11.37%
PCI	19.60%
Phosphodiesterase inhibitors	3.59%
Statins	58.78%

Table 5.3: The percentage of the overall population that received each treatment based on the sample population. Note that the same patient may have been prescribed multiple treatments.

present if there was an active prescription for the patient in the EHR. If there was no record of a treatment, we assumed that the patient was not administered the specific medication. Thus, the missing percentage for all treatments is 0.0%. Family history and smoking habits were available in the database for only a portion of the patients. Continuous features, such as cholesterol and blood pressure levels, were extracted from the vitals and lab tests records.

We imputed missing values using `opt.cv`, the state-of-the-art ML algorithm proposed by [43]. Given that the underlying pattern of missing data was not known, we opted for a method whose performance remained consistent across different types of "missingness". In [43], the authors demonstrated on 84 data sets that the accuracy of their algorithm relative to benchmark ones does not appear to differ drastically between the MCAR and MNAR patterns. The latter constitutes the most common type of missing data in health care applications, as values are not usually randomly incomplete for reasons such as missed study visits, patients lost to follow-up, missing information in source documents, and lack of availability among others. We created artificial missing data under the MNAR mechanism and compared `opt.cv`

with other well-established missing data imputation techniques in our dataset. We evaluated the resulting imputation error and the effect on downstream predictive performance for the binary classification task. Our results showed that `opt.cv` provided an edge across all metrics considered. Thus, it was selected as the imputation algorithm for the independent covariates of both the binary classification and regression models.

5.3 Estimating time to adverse event for right censored patients

In censored datasets the outcome of interest is generally the time until an event (onset of disease, death, etc.), but the exact time of the event is unknown (censored) for some individuals. When a lower bound for these missing values is known (for example, a patient is known to be alive until at least time t) the data is said to be right censored. In our dataset, we considered the time of censoring to be the last event-free visit of the patient to the academic medical center. Thus, for each patient i where $t_i < 10$ (years) and no adverse event (stroke/heart attack) has been recorded, we set the censoring time $c_i = t_i$, the last time observed in the EHR. Our sample was comprised of 13,498 censored observations (62.9% of the overall population).

Methods from the survival analysis literature are usually employed in the presence of censored populations. A common survival analysis technique is the Cox proportional hazards regression [84] which models the hazard rate for an event as a linear combination of covariate effects. Although this model is widely used and easily interpreted, its parametric nature makes it unable to identify non-linear effects or interactions between covariates [53].

We propose a data-driven methodology that utilizes a k-NN approach to identify patients with similar outcomes and known trajectories based on their covariates. We consider the set A (B) of patients that had (did not have) an adverse event within 10 years. Note that within set B the EHR indicate that no adverse event occurred within the defined time frame. Let C be the set of censored patients that did not have an adverse event within a time t_c

(less than 10 years) and they disappear from the EHR after t_c. It is not known whether they experienced an adverse event within 10 years or not. In order to estimate the TAE for patient X in the set C, we consider patients within $A \cup B$ such that:

1. They have the same gender as X. It has been recognized that women form a distinct subpopulation within patients with CAD [291].

2. They belong to the same age group as X. Age at time of diagnosis plays a major role in the development and the effects of CAD [354].

3. Their ground truth outcome metric is greater or equal to the censoring time of X. The patient will potentially experience an adverse event after the censoring time t_c.

Based on the Euclidean distance across the patient specific factors depicted in Table 5.1 (factors with continuous values were normalized to have zero mean and standard deviation of one), we find the k-nearest neighbors of X within the cohort outlined. We assign to the censored patient X the average time to adverse event of their k-nearest neighbors. We used cross-validation to set the parameter $k = 50$. The outcome of interest was the AUC performance of the binary classification model presented in Section 5.4 (Figure 5.2). We selected the value of the unsupervised learning model parameter according to the performance of the binary classification model on the 10-year risk task. Our method allows us to build for every censored patient a unique cluster of k-NN, introducing a personalization aspect in the estimation of TAE.

The k-NN algorithm's performance is $R^2 = 0.81$ according to the following process:

1. Select a sample of the population which was not censored (the TAE t_i is known).

2. Artificially generate a censoring time t_i^c, sampled uniformly across the interval [1,t_i] corresponding to a day in the 10 year time frame.

3. Apply the k-NN algorithm to estimate the TAE and compare the results with the ground truth that is known.

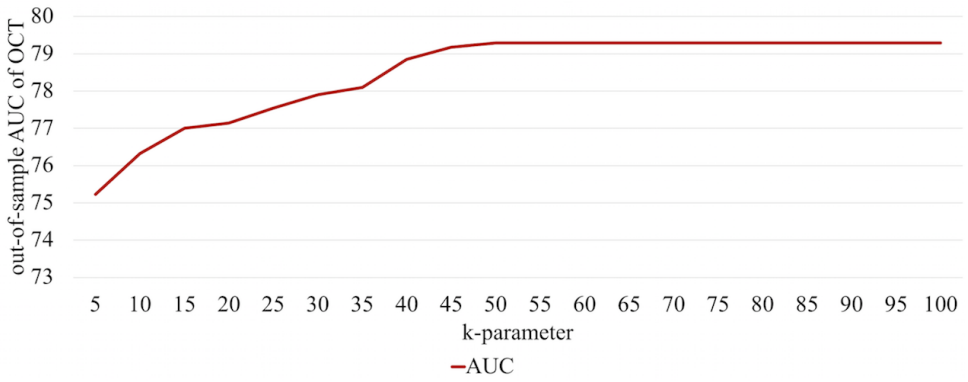


Figure 5.2: Graph of a Cross-validation results for the selection of the k parameter for the k-NN model.

We impute the outcomes of 13,679 censored observations, following this approach. We create a complete dataset that is further used for the creation and validation of the predictive and prescriptive models. The inclusion of the censored patients permitted a higher sample size for the binary classification and regression models that led to more accurate and stable results. The exclusion of such cases would restrict the overall population to only 7,962 observations, limiting the downstream predictive performance of the models.

5.4 The Binary Classifications Models

The first problem we addressed is the creation of personalized risk prediction models for CAD patients. Our binary outcome of interest is the occurrence of an adverse event (stroke or heart attack) within a 10-year time period. This time frame is in accordance with the vast majority of established CAD risk calculators [143, 110, 288]. The medical community recognizes the chronic nature of the disease and as a result it focuses on evaluating its impact on the health of the patient over a long-term horizon. Both the AHA and the American College of Cardiology annually update their guidelines on the primary prevention of cardiovascular disease releasing new versions of 10-year CAD risk scores [12]. Although this time frame is challenging and the health condition can significantly change over years, we decided to follow the paradigm of the existing literature.

We apply widely established ML algorithms to the data and compare their out-of-sample performance on the testing set. Table 5.4 provides a summary of the results for Logistic Regression, RF [55], GBT [73], CART [56], and OCT [27, 34].

We split the $n = 21,460$ patients in 75% for Training and Validation and 25% for Testing, using $p = 31$ patient characteristics (Table 5.1). Our sample includes all censored observations whose values were imputed using the methodology described in Section 5.3. These observations were not excluded as a higher sample size improved the model's out-of-sample performance. A higher sample size had a significant positive effect on the downstream performance of the binary classification models. We evaluated the predictive power of the algorithm under additional random splittings of the data. Thus, we ensured that the evaluation of the global algorithm was not sensitive to a particular split of the dataset.

L_2 regularization was used for the logistic regression model and 10-fold-cross-validation was employed to set the hyper-parameters of each method. In the case of OCT and CART, we tuned the complexity parameter, the maximum depth, and minimum bucket. Based on cross-validation results, the number of greedy trees used for the RF model was set to 500.

Our objective is to create an accurate model that would have high chances of affecting the medical practice. Even though there has been a steep increase in publications that utilize artificial intelligence and ML in the field of medicine, only a small proportion of those models have been integrated into the healthcare system [112]. Clinicians need actionable insights and guidelines they can explain and understand [247]. Algorithms have to satisfy this condition. Otherwise, the final outputs of these methods do not actually impact the patients. The [121] validated such concerns by mandating the use of interpretable ML models when it comes to medical decision making.

For this reason, we decided to focus on the model of the OCT algorithm, which was proposed by [27], see also [34]. Its tree structure accounts for non-linear interactions among variables providing an edge compared to Logistic Regression and comparable performance to ensemble approaches such as RF or GBT (see Table 5.4). RF (84.29%) yields better AUC results compared to OCT (81.54%), although quite similar in terms of accuracy for a fixed

	Out-of-sample AUC	In-sample AUC	Out-of-sample Accuracy	In-sample Accuracy
OCT	81.54%	81.35%	81.45%	81.36%
CART	73.33%	72.66%	80.23%	80.12%
RF	84.29%	83.29%	81.88%	82.35%
Logistic Regression	80.83%	82.21%	80.55%	80.98%
GBT	81.43%	82.76%	81.03%	81.27%
Baseline			73.51%	73.51%

Table 5.4: Results of the different ML algorithms used to predict the occurrence of an adverse event within 10 years after diagnosis. We consider as Baseline the simple model that predicts that all patients will experience an adverse event. Accuracy is measured using a probability above 50% as the threshold. The term "Out-of-sample" signifies the performance of the model on the Test set and "In-sample" on the Training set.

threshold (81.88%, 81.45% respectively). However, RF grows multiple decision trees and assigns for each observation the class that is indicated by the majority of the decision trees. OCT provides us with a single tree whose branches can be easily explained to physicians. Each path leads to comprehensible clinical decision rules that could positively affect the cardiovascular practice. Its model achieves superior performance in both accuracy and AUC when compared to all other ML methods, including the advanced ensemble algorithm of GBT. Moreover, Logistic Regression (80.83% AUC) is more accurate compared to CART (73.33% AUC), but slightly under-performing with respect to more sophisticated algorithms (81.43% AUC).

The final OCT model is depicted in Figures 5.3, 5.4, 5.5. Table 5.5 presents its ten most significant variables. An analysis of the most predictive features follows below:

- **Time in the System** (TimeinSystem): the time that the patient has been observed in the BMC database (from the first record until time of diagnosis t_0). It serves as an indicator of their medical condition and history information depth. TimeinSystem does not incorporate any patient details after the time t_0, avoiding the inclusion of survivorship bias in the data. As shown in Figures 5.3, 5.4, 5.5, higher values of the TimeinSystem variable are associated with leaves that predict positive outcomes for the patient. This result indicates that physicians are more effective when they have

extensive amount of information available and follow their patients' trajectories over longer periods of time.

- **Prescription of Medication** (Nitrates/ Beta Blockers/ Statins/ ACE Inhibitors): whether a patient has been systematically treated with one particular type of medication. Depending on the decision path of the tree, the risk of an adverse event might increase or decrease if the medication has been prescribed. There need not be a causality relation for the changes in risk. Only association can be deduced from such a model. However, these results reinforce the argument that personalization in the treatment can indeed affect the survival of the CAD population.

- **CABG/PCI**: whether the patient has performed a revascularization procedure. We notice that positive values in these two variables are associated with leaves that suggest pessimistic patient prognoses. Diagnosed CAD patients with more severe symptoms of atherosclerosis are usually suggested to perform at least one of these interventions (CABG, PCI) [124].

- **Patient Age at Diagnosis**: the age of the patient at the time of diagnosis in the EHR system. Across the model we notice that older populations are associated with higher risk, confirming a wide range of CAD risk calculators published in the medical literature [79, 269, 110].

- **HDL (mg/dL) levels**: the HDL (mg/dL) levels from a blood test conducted at the time of diagnosis. Depending on the position of the split in the tree, higher levels of HDL may positively or negatively impact the ten year risk of CAD.

- **Median Systolic Blood Pressure**: the median of the systolic blood pressure measurements recorded in the EHR across all visits in a window of three months before t_0. We consider the median due to the noise frequently encountered in systolic blood pressure measurements [336, 114, 104].

Feature	Importance
Time in the System	27.40%
Prescription of Nitrates	19.80%
Prescription of Beta Blockers	15.01%
PCI operation	12.96%
Prescription of Statins	10.53%
CABG operation	3.23%
Patient Age at Diagnosis	2.87%
Prescription of ACE inhibitors	1.86%
HDL (mg/dL) levels	1.31%
Median Systolic Blood Pressure	1.06%

Table 5.5: Demonstration of the independent variable ranking in the OCT binary classification model. The importance of each variable is measured as the total decrease in the loss function as a direct result of each split in a tree that uses this variable. The results are normalized so that they sum to one.

5.4.1 Analysis of characteristic decision paths

We analyze distinctive risk profiles from the OCT model that provide interesting insights for the management of CAD patients.

- **Paths 1 & 2:** Contain samples whose presence in the EHR was recorded only for two months before the diagnosis. Leaf 1 refers to patients that are administered a PCI operation and leaf 2 to those who perform a CABG surgery. Both paths associate extremely high risk to the corresponding population.

- **Paths 3 & 4:** Refer to individuals who are present in the BMC system at least seven years. They are not treated with PCI, neither with beta blockers nor statins. Their baseline risk of an adverse event is 7.78%. However, this risk differs depending on the age group they belong. Specifically, those individuals under 68 years old have 1.45% probability of having a stroke or heart attack over the next ten years. On the contrary, older patients have 18.11% chance of experiencing an adverse event.

- **Paths 5 & 6:** Include patients who are present in the BMC system for at least two months and are prescribed PCI but no CABG surgery. They are not treated with beta

blockers nor statins and their blood glucose levels are lower than 149 mg/dL. Their baseline risk of an adverse event is 12.53%. This risk differs again depending on the age group they belong. Specifically, those under 57 years old have 95.19% probability of avoiding a stroke or heart attack over the next ten years. On the contrary, patients older than 57 years of age have 14.03% chance of experiencing such an event.

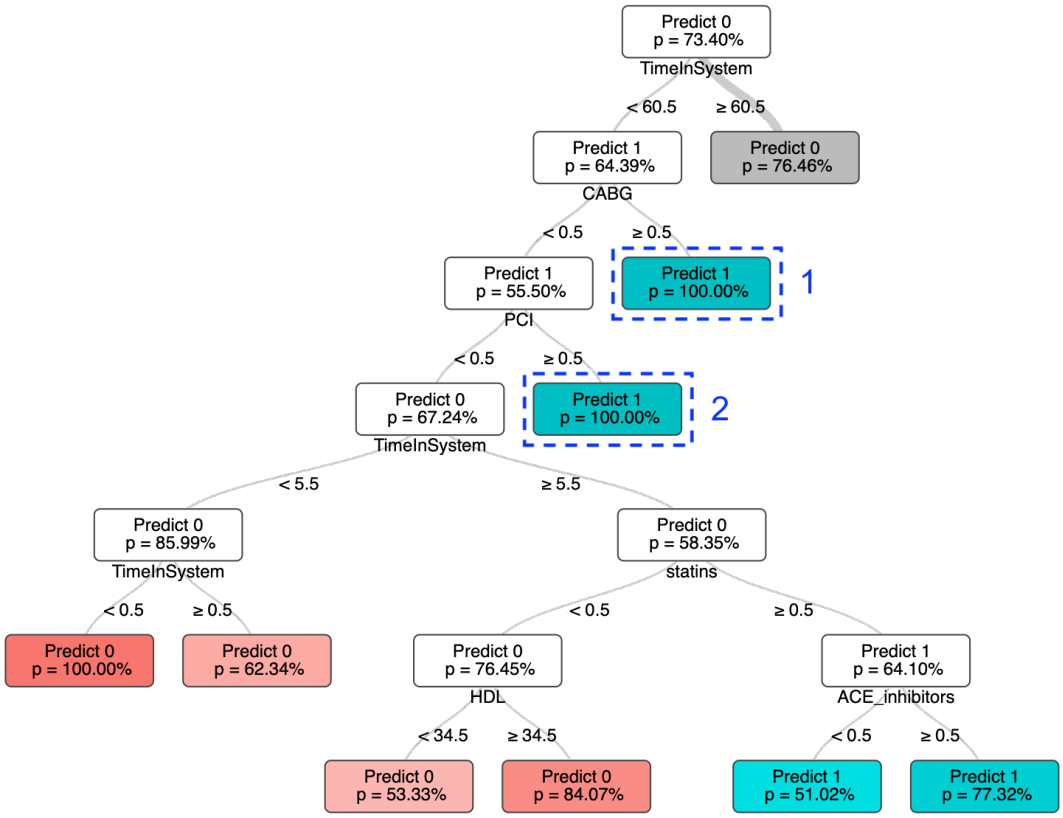


Figure 5.3: Visualization of the first part of the OCT model. Paths 1 and 2 are indicated with blue dashed rectangular frames. Shaded nodes include a collapsed subset of the tree model.

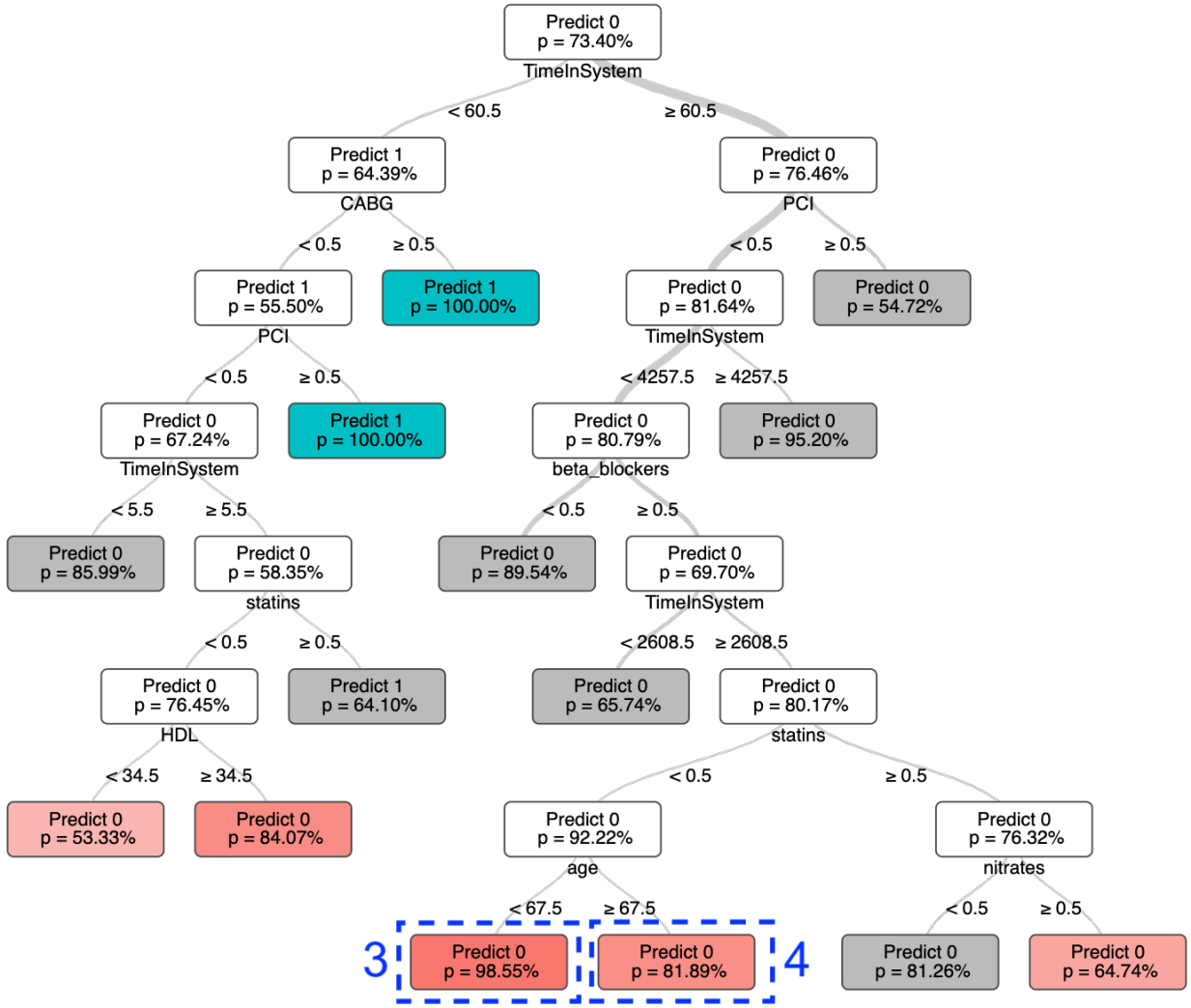


Figure 5.4: Visualization of the second part of the OCT model. Paths 3 and 4 are indicated with blue dashed rectangular frames. Shaded nodes include a collapsed subset of the tree model.

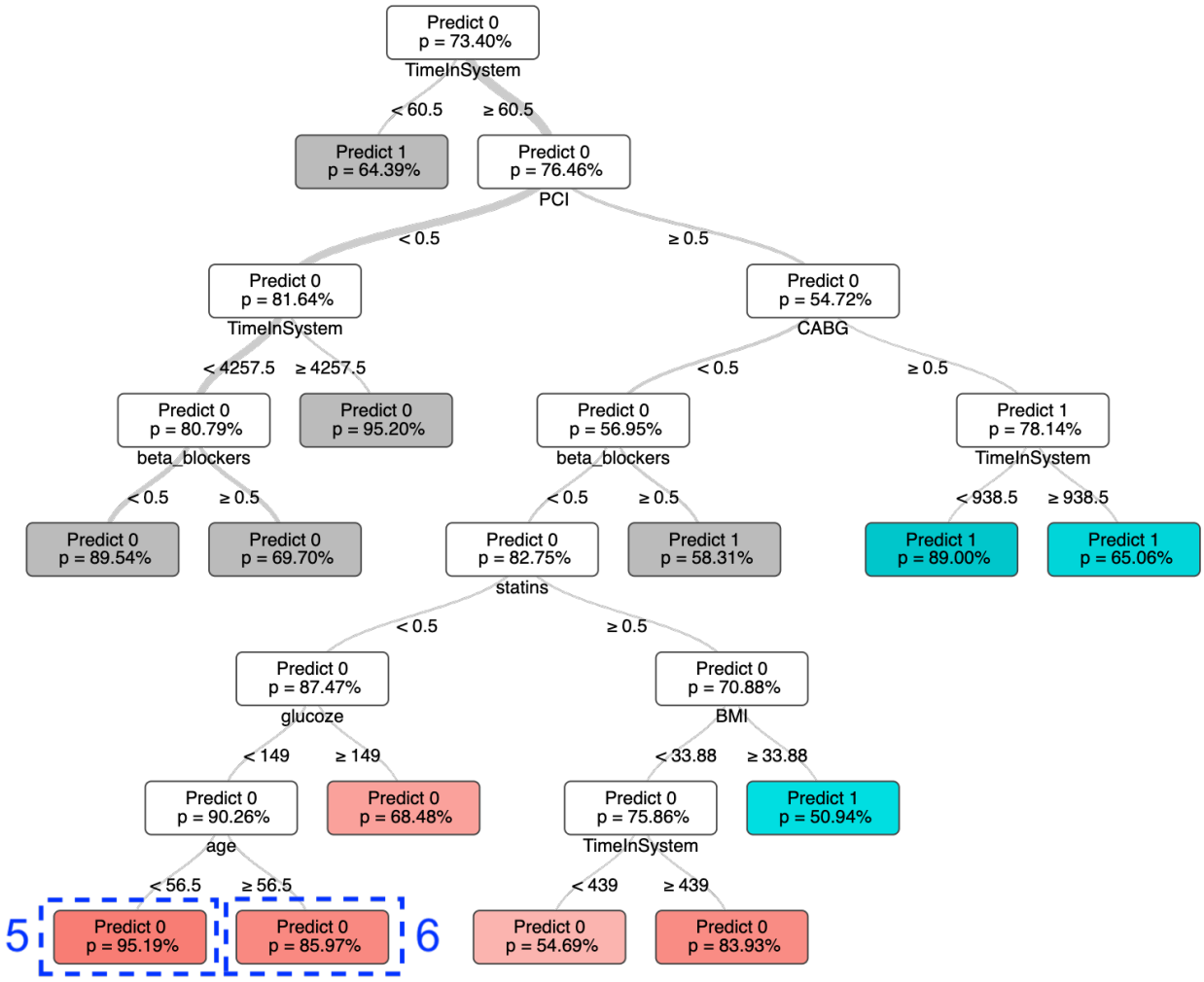


Figure 5.5: Visualization of the third part of the OCT model. Paths 3 and 4 are indicated with blue dashed rectangular frames. Shaded nodes include a collapsed subset of the tree model.

	ORT	CART	RF	Linear Regression	GBT
CABG	73.14%	71.91%	**83.00%**	80.32%	80.06%
PCI	68.30%	67.73%	**74.58%**	73.21%	73.21%
Drugs 1	78.64%	75.35%	**83.92%**	82.94%	82.48%
Drugs 2	73.46%	72.56%	**80.02%**	79.98%	79.50%
Drugs 3	67.10%	69.03%	**77.71%**	75.34%	75.29%

Table 5.6: Results of supervised ML algorithms to predict the TAE since diagnosis. We report the "Out-of-sample" R^2 performance of each model on the Testing set.

5.5 The Regression Models

Predicting the risk of an adverse event within a 10-year time frame is an important question that we address in Section 5.4. However, a personalized prescriptive algorithm requires the creation of accurate regression models that, given the condition of a patient, estimate the exact TAE for each potential treatment. We leveraged various state-of-the-art ML methods, both interpretable and non-interpretable, to generate a set of estimations at an individual level [56, 55, 27, 34, 73]. We trained a separate model for each combination of method and treatment using as sample population patients that exclusively received this regimen. For example, we applied the RF algorithm to generate five predictive models that correspond to CABG, PCI, Drugs 1, 2, and 3. We followed the same process for CART, Linear Regression, GBT, and Optimal Regression Trees (ORT). As in the classification task, we applied 10-fold-cross-validation to determine the hyper-parameters of each model, including the complexity parameter, the maximum depth, and minimum bucket for ORT and CART. Based on the cross-validation results for the regression task, the number of greedy trees for the RF model was set to 250 in contrast to 500 that were chosen for the binary classification outcome. We used L_2 regularization for the linear regression model. Table 5.6 provides a summary of each method's out-of-sample performance for every treatment option in terms of the R^2 metric.

The results from Table 5.6 indicate that RF outperforms the other methods in all tasks in terms of the R^2 metric. CART, on the other hand, appears as the least performing method across all tasks. ORT have an edge over the greedy tree-based approach, other

than in the case of category "Drugs 3". We observe that Linear Regression and GBT have comparable performance for all types of treatment. We will leverage all these models as the main component of our prescriptive algorithm, presented in Section 5.6.

We created separate models for each treatment population to avoid biases in the prediction due to the existing treatment prescription patterns in the EHR [138]. Our goal was to identify, for each patient, what is the therapy that would maximize their TAE. Therefore, a distinction was needed between the different populations that received each treatment option. The existing regimen allocation process could have significantly biased the prescriptive algorithm if included as an independent feature in the set of covariates X [306]. For instance, if physicians in BMC prescribed CABG only to the younger population, the ML model would not have been able to distinguish between the effect of CABG and the age of the patient.

5.6 `ML4CAD`: The Prescription Algorithm

The regression models serve as the basis for the prescription algorithm, utilizing the point predictions as counterfactual estimations. The objective of the prescription algorithm is to understand the potential effect of every therapy that each patient would have experienced, had it been prescribed to them. For example, knowing the outcome of patient X who received CABG surgery, we aim to estimate the outcome metric of a PCI intervention and for each of the Drugs options. We present `ML4CAD`, a personalized prescriptive algorithm that utilizes multiple ML models at once to identify the most effective therapy for CAD patients. Our method is structured as follows:

1. We impute the missing values of the patient characteristics (Table 5.1) using a state-of-the-art optimization framework [43].

2. We compute the TAE for right censored patients.

3. We split the population into training and test sets. The training set is used to train the regression models and the test set is utilized to assess the predictive and prescriptive performance of the algorithm.

4. We train a separate regression model for each treatment option for all predictive algorithms to estimate the TAE. The set of covariates X' used to create the predictive models does not include any features that refer to the treatment options (see Table 5.1 for a summary of the independent features and Table 5.2 for the list of prescription options).

5. We use all models to get estimations of the TAE for each treatment option and every patient in the test set. Thus, we have at our disposal a table of estimations for any new individual considered. Table 5.7 provides an illustration of the output for patient X.

6. We select the most effective treatment for the patient according to a voting scheme among the ML methods:

   (a) If the majority of the regression models votes a single treatment (regimen with the best expected effect), the algorithm recommends this therapy to the physician. In the example of patient X (see Table 5.7), `ML4CAD` suggests the prescription of CABG.

   (b) If there are ties between the different therapies (i.e., two methods suggest Drugs 1 and two others indicate Drugs 2), then the votes get weighted by the out-of-sample accuracy of the predictive models. For the analysis of this chapter, the R^2 metric was used.

7. The final TAE is computed as the average of the ML methods whose suggestion agreed with the algorithm recommendation.

`ML4CAD` provides a new framework for personalized prescriptions which is structured on the plurality of different ML models. In contrast to the simple Regress and Compare approach, it combines multiple ML models to identify the most beneficial treatment option. The validity of the algorithm's recommendations gets reinforced by an increasing number of underlying ML models that provide accurate estimations of the counterfactuals. In other words, the user gains more confidence in the capability of the algorithm to identify the optimal therapy the

ML Method	CABG	PCI	Med. 1	Med. 2	Med. 3
ORT	**4.65**	4.59	3.89	3.76	3.54
CART	**7.13**	3.38	6.10	4.16	3.96
RF	**5.77**	4.93	5.44	4.26	4.49
Linear Regression	**5.75**	3.53	**5.75**	4.17	4.44
GBT	4.08	**6.28**	5.39	5.31	3.37

Table 5.7: Estimations of TAE (years) for patient X from the five ML methods considered for each treatment option. We highlight the best treatment option for each ML model. Note that four out of the five models agree on the CABG recommendation.

more models are available for comparison. This methodology also allows for transparency towards the decision maker. Potential recommendations can be compared at an individual level to be decided what would be the best option for each particular case.

5.6.1 Bridging the gap with practitioners

We created an online `ML4CAD` application for physicians who would be interested to inform their decision making process using our personalized algorithm. Practitioners can now have access to our website (`https://personalized.shinyapps.io/ML4CAD/`), where they are able to quickly test the recommendations of the algorithm on new patient data. Figure 5.6 shows an image of the main application dashboard. The platform computes online a table similar to Table 5.7, demonstrating to the user all the available options and their projected outcomes. The final `ML4CAD` suggestion is highlighted on the right of the screen. A detailed comparison of the out-of-sample performance of all ML models across the five treatment tasks is also available. Moreover, clinicians can view aggregate results about the treatment allocation mechanism according to different demographic features such as gender, ethnicity, or age group. With this application we aspire to turn the proposed ML-based recommendation system into an actionable framework for the cardiovascular community. The latter can now leverage this tool as an assistance to its decision making process and prolong the life expectancy of its patients.

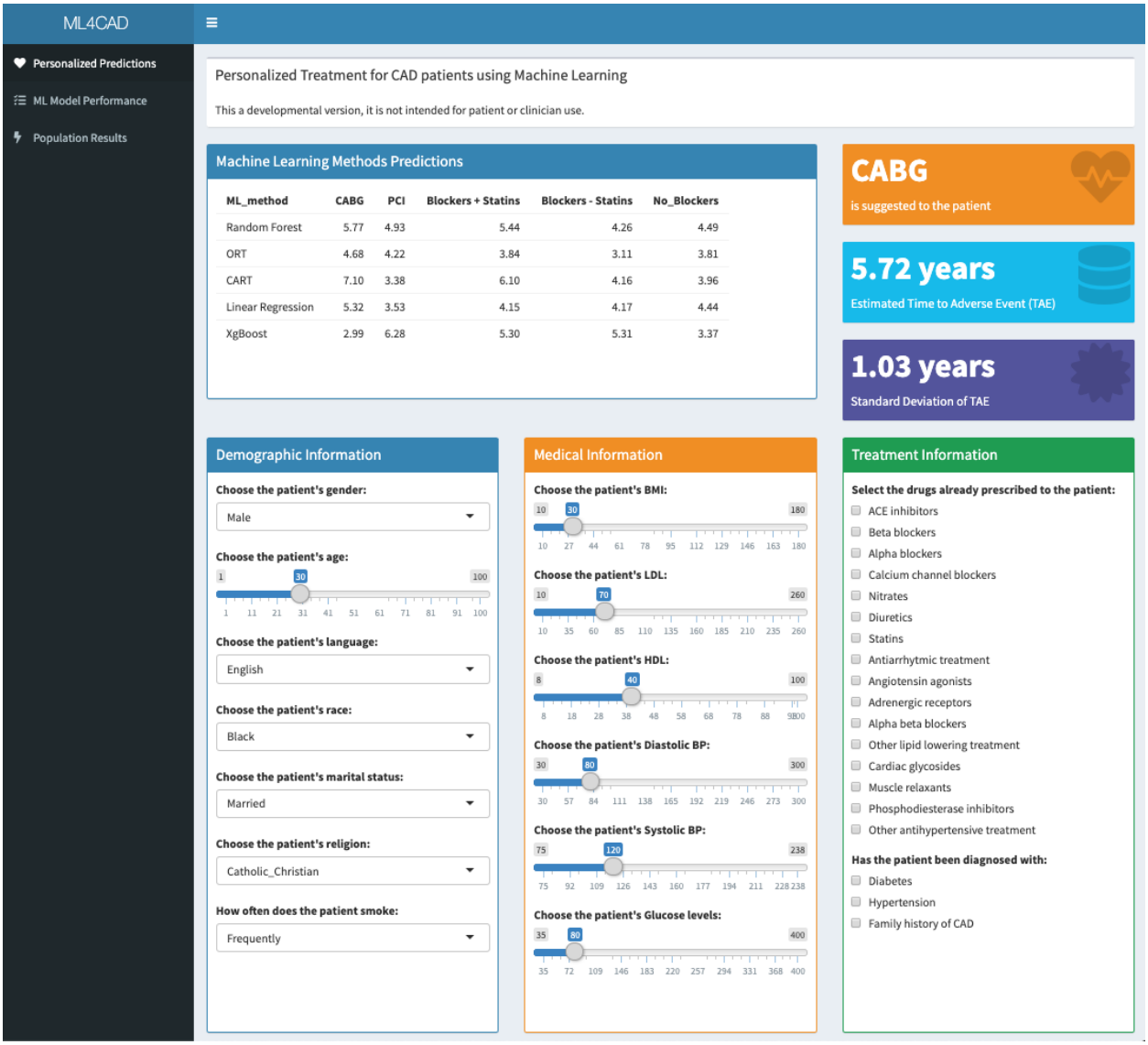


Figure 5.6: Visualization of the ML4CAD online application.

5.6.2 Prescriptive algorithm evaluation

Assessing the quality of the prescriptive algorithm poses a challenge. We do not have at our disposal data that indicate the TAE for all counterfactual outcomes of each patient. We created appropriate metrics that provide an objective evaluation framework of the algorithm's performance. We define the problem as follows, let:

- p be a variable that takes values in the set $[T]$ of all the prescriptive options;
- j be a variable that takes values in the set $[M]$ of all the predictive models;
- z_i be the treatment that patient i followed at the standard of care;
- t_i be the TAE for patient i and treatment z_i;
- τ_i be the treatment recommendation of `ML4CAD` for patient i;
- θ_i^j be the treatment recommendation of machine learning model $j \in [M]$ for patient i using a simple "Regress and Compare approach";
- $g_i^j(p)$ be the estimated TAE for patient i for treatment p from the regression model j, where $j \in [M]$;
- $y_i(p)$ to be the estimated TAE for patient i when `ML4CAD` recommends treatment p;
- $\overline{t_p}$ average TAE observed in the data for all patients who were prescribed treatment p.

Using the notation above, the expected TAE for patient i is according to `ML4CAD`:

$$\begin{aligned} y_i(\tau_i) &= \frac{1}{K} \sum_{j:\arg\max_p g_i^j(p)=\tau_i} g_i^j(\tau_i), \\ K &= |j : \arg\max_p g_i^j(p) = \tau_i|, \quad i \in [n]. \end{aligned} \tag{5.2}$$

We evaluate the quality of the algorithm's personalized recommendations based on the following metrics:

1. **Prescription Effectiveness and Robustness**:

   The goal of the first metrics is to compare the performance of the `ML4CAD` recommendations with the regimens prescribed at the standard of care. Due to the uncertainty in counterfactual estimation, we consider different predictions of the TAE and a multitude of ground truths. Our baseline ground truth refers to realizations of TAE that we observe in the BMC database. This ground truth provides us with the exact TAE

associated to the treatment regimen that was prescribed by the physicians at the hospital. Alternative ground truths refer to estimations of the TAE by treatment-based regression models.

- **Prescription Effectiveness (PE)**

  We fix, for each patient $i \in [n]$, the treatment suggestion τ_i from the `ML4CAD` algorithm. We know the outcome t_i for treatment choice z_i (observed in the data - baseline ground truth). Thus, comparing the prescription effectiveness of the `ML4CAD` versus the standard of care would be equal to:

$$\text{PE}(\texttt{ML4CAD}) = \frac{1}{n}\sum_{i=1}^{n} y_i(\tau_i) - t_i. \tag{5.3}$$

  `ML4CAD` averages the TAE projected by the regression models that agree on the most beneficial treatment for patient i, namely τ_i. We can evaluate the prescription effectiveness of this recommendation by considering each ML model in isolation. Each regression model j provides for patient i and regimen p an estimation $g_i^j(p)$. Therefore, if we fix $p = \tau_i$, we can get an evaluation of the projected TAE and compare it to the standard of care.

$$\begin{aligned} \text{PE}(\texttt{ML}_j) = \frac{1}{n}\sum_{i=1}^{n} g_i^j(\tau_i) - t_i, \\ \forall j \in \{1, \ldots, M\}. \end{aligned} \tag{5.4}$$

  Comparing multiple ML estimations for the TAE of the recommendation τ_i renders the results more credible to biases of a specific predictive algorithm.

- **Prescription Robustness (PR)**

  The PE metric measures the effect of the `ML4CAD` recommended therapies against a fixed given ground truth from the EHR of the BMC. Nevertheless, knowing that each patient i was given a treatment t_i, we can generate alternative ground truths.

We can, then, evaluate the benefit of the personalization approach against those. Each ground truth corresponds to an estimation of what would happen to patient i if ML model j was an oracle that knew the reality and the effects of treatment z_i.

$$\begin{aligned} \text{PR}(\texttt{ML}_{j,k}) = \frac{1}{n}\sum_{i=1}^{n}(g_i^j(\tau_i) - g_i^k(z_i)), \\ \forall j,k \in [M]. \end{aligned} \tag{5.5}$$

In this setting, decisions τ_i, z_i are fixed and we evaluate all the combinations between RF, CART, ORT, GBT, and Linear Regression. We include also the case where `ML4CAD` is used to estimate the effect of τ_i but not the one of t_i.

$$\begin{aligned} \text{PR}(\texttt{ML4CAD}_k) = \frac{1}{n}\sum_{i=1}^{n}(y_i(\tau_i) - g_i^k(z_i)), \\ \forall k \in [M]. \end{aligned} \tag{5.6}$$

The goal of this metric is to evaluate the robustness of the treatment effect under different ground truths. In Section 5.7, we perform an extensive comparison over all methods and ground truths considered (see Table 5.8). We introduce this approach to avoid biased estimates of performance. The latter could not have been avoided if we were comparing our results only to the baseline ground truth.

2. **Prediction accuracy of TAE**:

$$\begin{aligned} \tilde{R}^2(\texttt{ML4CAD}) = 1 - \frac{\sum_{i\in S}(y_i(z_i) - t_i)^2}{\sum_{i\in S}(erlinet_{z_i} - t_i)^2}, \\ S = \{i : \tau_i = z_i\}, \quad i \in [n]. \end{aligned} \tag{5.7}$$

This metric follows the same structure as the well-known coefficient of determination

R^2. We apply it for each patient $i \in S$, the set of all samples where there is agreement between the ML4CAD and baseline prescription; $S = \{i : \tau_i = z_i\}$. Similar to the original measure, the known outcome t_i is compared to the estimated treatment effect $y_i(z_i)$ and to a baseline estimation. The latter in our case is $erlinet_{z_i}$, the mean TAE observed in the data for all patients who were prescribed treatment z_i. The adjusted coefficient of determination $\tilde{R}^2$ helps us evaluate whether the outcome that ML4CAD predicts for the known counterfactuals is accurate or not. It is impossible to evaluate the prescriptive algorithm across all treatment options. Only one out of the five is actually realized in practice. We focused on comparing for each patient the TAE according to the algorithm versus the one present in the data only for the cases where there was agreement between the two. This estimation, even though limited, provides us with a good baseline regarding the accuracy of our recommendations. We can extend the use of this metric to the "Regress and Compare" approach. Thus, we can estimate the $\tilde{R}^2(\texttt{ML}_j)$ of each predictive model $j \in [M]$.

$$\begin{aligned} \tilde{R}^2(\texttt{ML}_j) &= 1 - \frac{\sum_{i \in S}(g_i^j(z_i) - t_i)^2}{\sum_{i \in S}(erlinet_{z_i} - t_i)^2}, \\ S &= \{i : \theta_i^j = z_i\}, \quad i \in [n]. \end{aligned} \tag{5.8}$$

3. **Degree of ML Agreement (DMLA)**:

This measure refers to the degree of agreement among the ML models (DMLA) with the recommended treatment τ_i. For each patient, we count the number of methods that agree on the ML4CAD suggested treatment τ_i. We report the distribution of this metric across the whole population. Cases where there is high degree of agreement are associated with higher confidence on the suggested prescription. On the contrary, we are less confident in cases where there is misalignment between the ML models regarding the best treatment option.

5.7 Prescriptive algorithm results

In this Section, we present numerical results with respect to the evaluation metrics introduced in Section 5.6. We provide insights regarding different sample population subgroups. We also discuss new treatment allocation patterns based on `ML4CAD` recommendations.

5.7.1 Prescription Effectiveness (PE) and Robustness (PR)

We summarize our results with respect to the PE and PR metrics in Table 5.8. The first table column corresponds to PE (baseline ground truth), whereas the rest of the columns refer to PR (ML-based ground truths). Table 5.8 presents the expected relative gain in TAE of `ML4CAD` over the baseline. Its values demonstrate the average benefit in years of TAE when comparing the current and `ML4CAD` treatment allocation plan across different estimation models. Each ground truth (column) refers to alternative estimations of the TAE under the current treatment allocation plan. Thus, if the ground truth is the baseline (BMC Database), the suggested times correspond the TAE observed in the data. When the ground truth is set to be the ORT algorithm, the predicted times $g_i^{ORT}(z_i)$ mirror ORT estimations when the treatment allocation is fixed to the physicians' decisions from the hospital (z_i). Each prediction model (row) provides us with a continuous prediction of a patient's TAE when the treatment allocation plan is set by the `ML4CAD` algorithm (τ_i). Thus, the values in Table 5.8 correspond to the metrics defined in Equations 5.4 (first column) and 5.5 (subsequent columns).

When compared to the current allocation scheme, our prescription algorithm improves the average TAE by 24.11%, with respect to the PE metric, with an increase from 4.56 to 5.66 years (13 months). Column "Baseline (PE)" of Table 5.8 summarizes the results with respect to all regression models considered. `ML4CAD` provides the most optimistic estimations. It suggests a higher TAE versus its counterparts by at least 0.18 years (2 months). Linear

*The PE of the algorithm when the estimation model g^j is `ML4CAD` and the ground truth relates to the patient outcomes observed in the BMC database (See Equation 5.4).

†The PR of the algorithm when CART is the chosen estimation model g^j for the prescriptions $z_i, i \in [n]$ and the ground truth outcomes are computed according to the Linear Regression model g^k (See Equation 5.5).

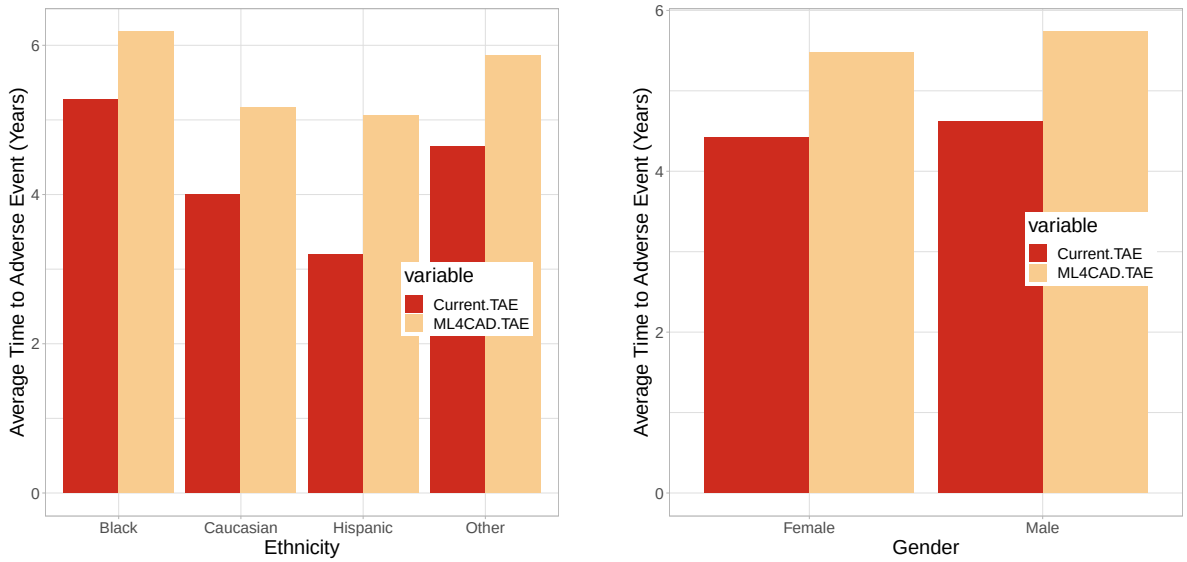
	Ground Truth					
Estimation Model	**Baseline**	**ORT**	**CART**	**RF**	**Linear Regression**	**GBT**
ML4CAD	1.101*	1.162	1.158	1.140	1.178	1.283
ORT	0.779	0.840	0.835	0.818	0.855	0.961
CART	0.923	0.983	0.979	0.965	0.999	1.105
RF	0.757	0.818	0.813	0.796	0.833†	0.939
Linear Regression	0.485	0.546	0.541	0.524	0.561	0.667
GBT	0.591	0.652	0.647	0.630	0.667	0.773

Table 5.8: Comparison of the "Prescription effectiveness" (PE) and "Prescription robustness" (PR) metrics for all estimation models and ground truths considered. The first column (Baseline) presents results with respect to the PE metric and refers to the TAE observed in the BMC database. All subsequent columns refer to the PR measure. Each of them represents a distinct ground truth. All units are shown in years. See Equations 5.4,5.3,5.5.

Regression appears to be the most pessimistic method with an average benefit over the baseline of 6 months (0.59 years). ORT and RF provide similar estimations of 0.77 and 0.75 years of improvement, respectively.

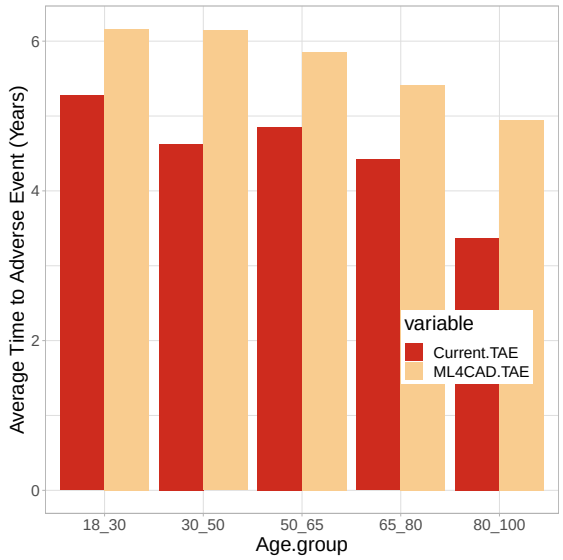
The comparable performance of the various estimation models presented in Table 5.8 reinforces the credibility of the prescription algorithm. We show that there is agreement between the potential improvement in the average TAE by an alternative treatment allocation scheme. Even in cases where we include ML models that did not participate in the ML4CAD recommendation, there is substantial benefit in the patients' life expectancy.

We observe better results across all age and ethnicity patient subgroups and for both genders. The benefit of using the algorithm was 17.09% (0.9 years) for Black patients, 29.03% (1.16 years) for Caucasian patients and 58.41% (1.86 months) for Hispanic patients. We also note 22.5% (0.99 years) improvement for patients $65-80$ years of age and 46.9% (1.58 years) for patients aged 80 or older. Male patients are expected to increase their time from 4.62 years to 5.73 (24.19% improvement) similar to female patients (from 4.42 years to 5.48). The performance of the prescriptive algorithm for selected patient subgroups compared to the BMC baseline is summarized in Figure 5.7.

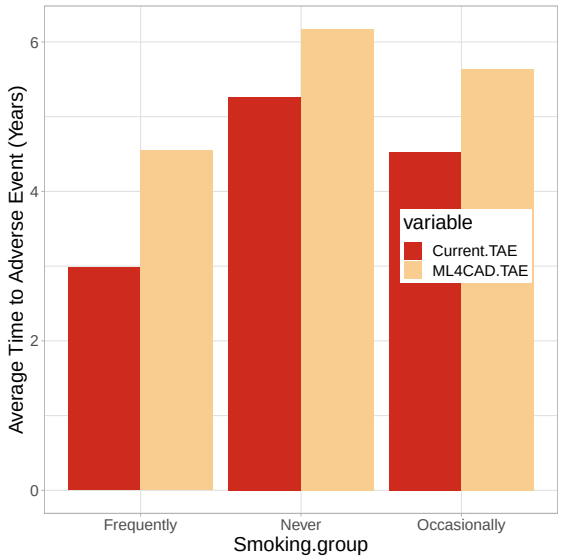


(a) Ethnicity Subgroups

(b) Gender Subgroups



(c) Age Subgroups



(d) Smoking Subgroups

Figure 5.7: Comparison of the expected years to adverse event after diagnosis for the age and ethnicity subgroups considered. The difference between the two bars for each sub-population refers to the prescription effectiveness (PE) of the algorithm for each respective patient group. "Current.TAE" refers to the outcomes observed in the EHR of the BMC. "ML4CAD.TAE" represents the expected TAE according to the prescription algorithm.

In terms of the PR metric, our results demonstrate a consistent improvement of the patient population TAE across all ground truths and estimation models. Table 5.8 summarizes the results of our analysis. We note that `ML4CAD` achieves the highest benefit when compared to all alternative scenarios of outcome realization. This is due to the incorporation of the voting system for the selection of the most effective treatment that accounts for all ML models. We show that even in the case of more pessimistic estimators, such as GBT or Linear Regression, there is a substantial benefit compared to the standard of care. Our approach does not guarantee optimality for the treatment selection problem. Nevertheless, it is experimentally shown that it can bring about substantial benefit to the CAD population.

We can also identify for each estimation model combinations with ground truths that outperform the rest of the alternatives. All methods demonstrate the highest improvement when associated with the GBT ground truth. For example, the ORT and CART model increase the average TAE by 0.96 and 1.10 years respectively. The next most optimistic contestant is Linear Regression. This is due to the fact that some methods on average overestimate or underestimate the expected TAE, translating these discrepancies in the PR metric.

5.7.2 Prediction accuracy of TAE

The "prediction accuracy of TAE" for the proposed prescriptive algorithm is $\tilde{R}^2(\texttt{ML4CAD}) = 78.7\%$. Table 5.9 provides a summary of the results for both the suggested method as well as "Regress and Compare" approaches from the baseline ML models. `ML4CAD` achieves better performance compared to the single prediction model counterparts. Aggregated predictions from different regression models lead to more accurate outcomes. The suggested voting scheme, not only reduces the uncertainty and bias of the estimations (See Section 5.7.1), but also results in highly accurate predictions.

Method	$\tilde{\mathbf{R}}^{\mathbf{2}}$
ML4CAD	**78.70%**
ORT	72.68%
CART	70.54%
RF	77.25%
Linear Regression	76.66%
GBT	76.59%

Table 5.9: Results summary for the Prediction Accaracy of TAE ($\tilde{R}^2$) metric.

5.7.3 Degree of ML agreement (DMLA)

The majority of the ML4CAD recommendations z_i are based on a common suggestion between at least three distinct ML models. Specifically, in 14.53% of the patients all methods suggest the same treatment for each individual. In 26.74% of the cases there is agreement between four models and in 34.48% of the observations three methods participate in the decision. Only in 0.26% of the samples, each regression model suggests a different prescription. In such cases, the ML4CAD recommendation is solely based on the suggestion of the most accurate one.

Table 5.10 provides detailed results for each treatment option. The last table column summarizes the results as a function of the total population. Each treatment specific column presents the proportional degree of agreement for all patients for which this treatment was suggested. Thus, we notice that CABG as well as Drugs 1 & 2 recommendations are, on average, more confident compared to Drugs 3 or PCI due to the higher degree of agreement. This is particularly true in the case of Drugs 1, where for 85.49% of the patients, three out of the five methods voted for the same regimen.

5.7.4 Treatment Allocation Patterns

In this section, we present insights regarding the ML4CAD treatment allocation patterns and we perform comparisons with the standard of care at the BMC. Our method agrees with the physicians' decisions in 28.24% of the cases. The results indicate a shift towards drug therapy and CABG, reducing the overall proportion of PCI (from 18.84% to 6.04%). The

Number of ML methods that agree with the recommendation	CABG	Drugs 1	Drugs 2	Drugs 3	PCI	Population Proportion
1	1.13%	0.22%	0.00%	0.00%	0.00%	0.26%
2	20.82%	14.29%	41.54%	**59.65%**	**49.10%**	23.99%
3	**35.41%**	32.30%	**43.98%**	36.23%	39.07%	**34.48%**
4	27.34%	**33.58%**	13.26%	3.64%	10.28%	26.74%
5	15.30%	19.61%	1.22%	0.47%	1.54%	14.53%

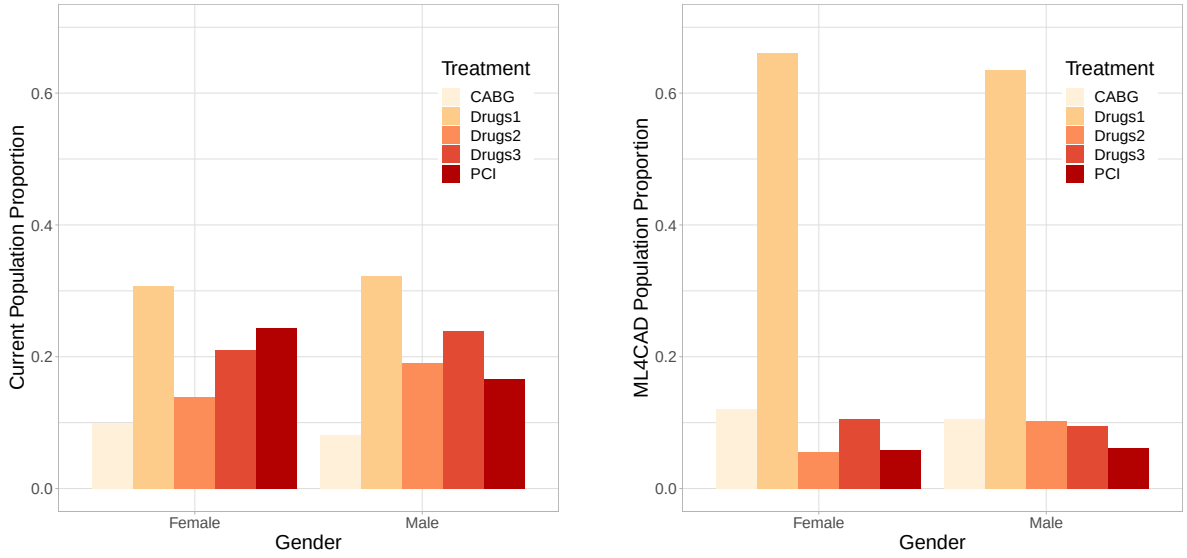
Table 5.10: Degree of ML Agreement between the models analyzed for each treatment option as well as a function of the overall test population.

prediction model indicates that patients with severe symptoms do not benefit significantly from a PCI versus a CABG surgery due to the eminent need for revascularization. Figure 5.8 illustrates a significant shift towards "Drugs 1" for both women and men. The algorithm also recognizes that treatment "Drugs 2" is less effective on female patients versus male. The `ML4CAD` allocation is in agreement with the most recent guidelines published by the AHA [327]. In the vast majority of cases, a combination of antihypertensive drugs (Blockers) with lipid lowering treatment (statins) is suggested. The overall proportion of the population that is recommended an invasive intervention is reduced due to the significant decline of PCI operations.

Figure 5.9 illustrates a comparison of the treatment allocation patterns between the `ML4CAD` algorithm, individual "Regress and Compare" models, and the standard of care we observe in the data. The graph demonstrates an agreement across all methods other than CART to increase the proportion of the population under "Drugs 1". The `ML4CAD` algorithm is more aligned with the RF policy due to the high predictive performance associated with the latter. We also note the reduction of "Drugs 2 & 3" across all methods. In the case of CABG there is disagreement between the ML models. GBT and Linear Regression suggest a significant raise in the proportion of CABG surgery at the expense of "Drugs 1". On the other hand, ORT, RF, and CART identify CABG as the optimal therapy for a lower proportion of the patient population.

		ML4CAD Allocation				
	Treatment	**CABG**	**Drugs 1**	**Drugs 2**	**Drugs 3**	**PCI**
Current Allocation	**CABG**	1.3%	4.1%	0.9%	1.6%	0.8%
	Drugs 1	2.3%	22.1%	3.7%	2.1%	1.7%
	Drugs 2	2.0%	12.3%	2.0%	0.2%	1.0%
	Drugs 3	3.2%	16.3%	1.0%	1.4%	1.1%
	PCI	2.2%	9.5%	1.3%	4.5%	1.4%

Table 5.11: Allocation of patients in the treatment options based on the standard of care and ML4CAD.



(a) Current Treatment Allocation

(b) ML4CAD Treatment Allocation

Figure 5.8: Population allocation to treatments split by gender.

5.8 Discussion

Combining historical data from a large EHR database and state-of-the-art ML algorithms resulted in an average TAE benefit of 24.11%% (1.1 years) for patients diagnosed with CAD. Our results show that differing medication regimens and revascularization strategies may produce varying clinical outcomes for patients. The use of ML may facilitate the identification of the optimal treatment strategy. Such efforts could directly address the primary objectives of the clinical cardiovascular practice, leading to symptoms reduction and an increase in the population life expectancy. Our findings uncover the greatest clinical benefit in medical

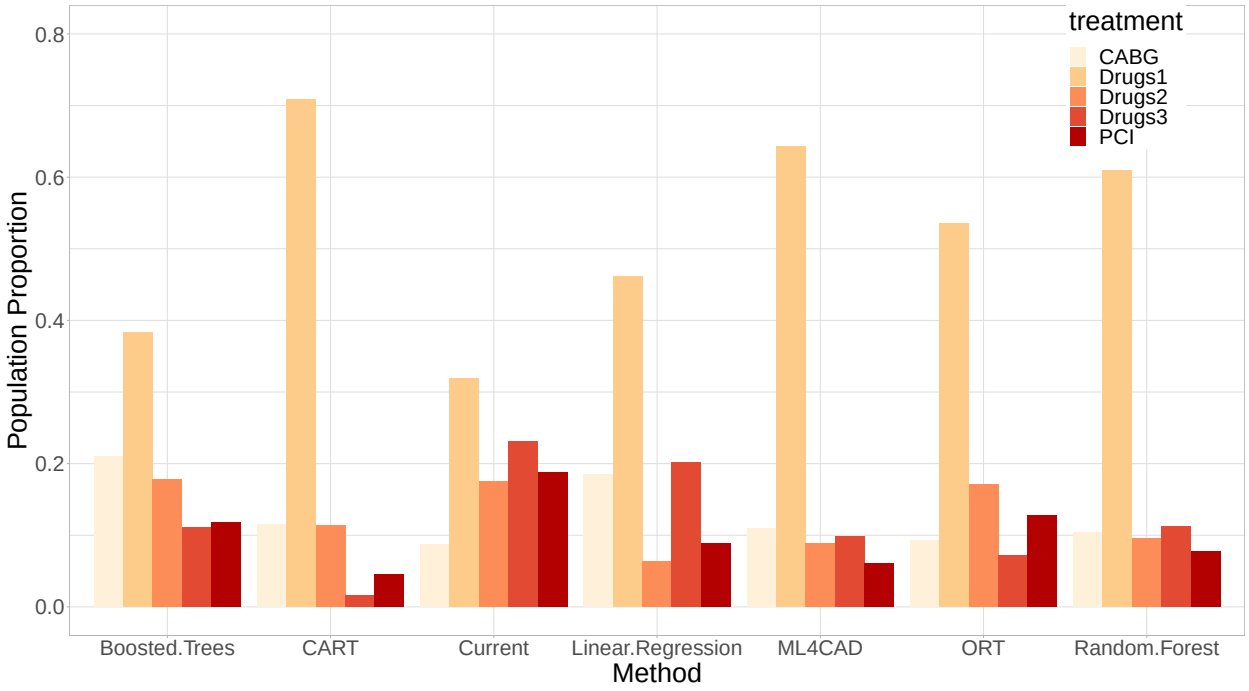


Figure 5.9: Treatment Allocation patterns between different ML methods.

therapy changes, consistent with themes that have emerged in clinical trials [51]. The optimal revascularization strategy in patients with multi-vessel CAD is an area of active investigation, with efforts focused on identifying which patient subgroups may benefit from different revascularization procedures [120]. Our technique may add clarity to this clinical challenge.

Our prescriptive approach is accurate, highly interpretable, and flexible for other healthcare applications. The use of multiple ground truths derived from independent ML models renders credibility to the results. In prescriptive problems where counterfactual outcomes cannot be evaluated against a known reference, leveraging multiple ML models can reduce the uncertainty behind suggested recommendations. For this reason, we believe that metrics such as the prescription effectiveness and robustness are key to the validation process.

Moreover, our online application bridges the gap between clinicians and the algorithm. Users can directly and simultaneously interact with multiple ML models from a user-friendly interface. Our method should easily accommodate alternative cardiovascular disease-

management approaches within specific disease subpopulations, such as arrhythmia and valvular disease management. A novelty of our approach is in the personalization of the decision-making process. It incorporates patient-specific factors, and provides guidelines for the physician at the time of diagnosis / clinical encounter. We believe this personalization is the primary driver of benefit relative to the standard of care. Similarly, there is emerging data on use of ML techniques to improve cardiac imaging phenotyping of cardiac disease states, such as heart failure [254].

The widespread use of EHR in clinical medicine was initially viewed with much optimism, however more recently it has been met with frustration by clinical providers. Concerns are being raised over the administrative burden to document the EHR and the resultant development of clinician “burn out". The methodology presented in this chapter identifies a mechanism to harness the power of the EHR in an effort to improve patient care and make it more personalized. It is true that the clinical acumen developed over time spent caring for patients cannot be replaced by algorithms. Nevertheless, the prospect of ML to guide clinicians and complement clinical decision making may help improve clinical outcomes for patients with cardiovascular and other diseases [111].

Our work has several limitations due to the nature of the EHR. A large percentage of the sample was right-censored. Patients were not randomized into treatment groups. Our data does not include socioeconomic factors or patient preferences that may be important in treatment decisions, such as income or fear of invasive treatment strategies. Although our matching methodology controls for several confounding factors that could explain differences in treatment effects, we can only estimate counterfactual outcomes. In addition, the study population of BMC is not representative of the general U.S. population as we observe a higher representation of non-Caucasian patients. As a result, the ability of `ML4CAD` to generalize in other institutions needs to be tested. Similarly to other studies, we recommend prospective validation of the models to the new population prior to the application of the algorithm to a different healthcare system [259]. Moreover, we should consider that the accuracy of the prediction model is limited, though significantly better than the baseline model. It leaves

room for improvement in that field by including new variables and further risk factors that are associated with CAD. Due to lack of sufficient data, we did not take into account different types of CABG surgery (i.e. arterial versus venous conduits) and PCI (i.e. newer versus older generation drug eluting stents, or bare metal stents versus drug eluting stents). Should more data were available, we could further differentiate the prescription categories beyond the five we include in this analysis, including drug specific recommendations. Moreover, the algorithm does not agree with the standard of care in most cases. This result indicates that new personalization techniques would need further input from clinicians that was not originally recorded in the EHR. Future research could address the issue of right censored patients with different approaches, which incorporate the time varying effects of the explanatory variables using optimization rather than heuristic methodologies. The ultimate validation of our algorithm would be the realization of a clinical trial. There, we would test the personalized recommendation to patients directly utilizing their EHR from the hospital system.

5.9 Conclusions

Despite these limitations, our approach establishes strong evidence for the benefit of individualizing CAD care. To our knowledge, this work represents the first ML study in treating cardiovascular disease and serves as a proof of concept. Moreover, the success of this data-driven approach invites further testing using datasets from other hospitals and patient populations. That includes care settings that contain more detailed information regarding the patients' condition, such as electrocardiogram findings and exercise and other lifestyle factors. The algorithm could be integrated in practice into existing EHR systems to generate dynamically personalized treatment recommendations. Testing the prescriptive algorithm in a clinical trial setting could provide conclusive evidence of clinical effectiveness. As large-scale genomic data become more widely available, the algorithm could readily incorporate such data to reach the full potential of personalized medicine in cardiovascular disease care. Our work is a key step toward a fully patient-centered approach to coronary artery disease management and the application of modern analytics in the medical field.

Chapter 6

The Non-linear Framingham Stroke Risk Score

Current stroke risk assessment tools presume the impact of risk factors is linear and cumulative. However, both novel risk factors and their interplay influencing stroke incidence are difficult to reveal using traditional additive models. The goal of this study was to improve upon the established Revised Framingham Stroke Risk Score (R-FSRS) and design an interactive N-SRS. Leveraging ML algorithms, our work aimed at increasing the accuracy of event prediction and uncovering new relationships in an interpretable fashion. A two-phase approach was used to create our stroke risk prediction score. First, clinical examinations of the Framingham offspring cohort were utilized as the training dataset for the predictive model. OCT were used to develop a tree-based model to predict 10-year risk of stroke. Unlike classical methods, this algorithm adaptively changes the splits on the independent variables, introducing non-linear interactions among them. Second, the model was validated with a multi-ethnicity cohort from the BMC. Our stroke risk score suggests a key dichotomy between patients with history of cardiovascular disease and the rest of the population. While it agrees with known findings, it also identified 23 unique stroke risk profiles and highlighted new non-linear relationships; such as the role of T-wave abnormality on electrocardiography and hematocrit levels in a patient's risk profile. Our results suggested that the non-linear approach significantly improves upon

the baseline in the AUC (training 87.4% (CI 0.85-0.90) vs. 73.74% (CI 0.70-0.76); validation 75.29% (CI 0.74-0.76) vs 65.93% (CI 0.64-0.67), even in multi-ethnicity populations. The clinical implications of the new risk score include prioritization of risk factor modification and personalized care at the patient level with improved targeting of interventions for stroke prevention.

6.1 Introduction

Over 70% of strokes occur in people without prior history of adverse events, emphasizing the importance of primary prevention [262]. Over the past four decades, several risk scores have been introduced to identify individuals at high risk for cerebrovascular disease [355, 226, 68]. These scores highlighted the benefit of introducing blood pressure treatment and other medication, leading to the significant decline of stroke rates over the past 15 years [62, 260].

The Framingham Heart Study Stroke Risk Score (FSRS) is one of the most established and respected standards for estimating 10-year stroke risk [355]. The FHS started with the goal of observing a large population of adults over time to better understand the factors that lead to cardiovascular and cerebrovascular disease. The original FSRS was based on stroke data from the 1960s and 1970s, but its application on contemporary cohorts showed overestimation of stroke risk [48, 230]. Recently, a Revised FSRS (R-FSRS) was introduced to account for temporal trends using data from the offspring cohort and reflecting updated stroke rate incidence [105].

These approaches apply traditional statistical tools such as the Cox Proportional Hazards model [85], which assume a linear, log-linear, or logit-linear relationship between the risk factors and the prevalence of the disease. While useful, they presume that the variables in their models interact in a mere additive fashion. The mathematical and medical realities, however, suggest that the interaction of risk factors and markers of disease acuity are far from linear, and that some variables gain or lose significance due to the absence or presence of other variables [199, 37]. In a logistic regression setting, interactions between risk factors can only be incorporated via cross-multiplication to estimate the combined relative risk.

However, this approach requires a significant augmentation of the feature space while it does not generalize to higher numbers of risk factors.

On that ground, we recognized the substantial benefit that algorithmic approaches and ML could bring in this field. We propose the N-SRS using the clinical examination data from the offspring cohort of the FHS to estimate the 10-year stroke risk. To achieve our objective, we utilize novel ML methods to predict the progression of cerebrovascular disease [27, 41]. Our model considers a wider spectrum of potential risk factors that include the prescribed medical regimen at the time of the examination. We suggest a new way of utilizing data from longitudinal studies that allows the creation of a larger dataset that can boost the performance of ML methods without introducing bias in the data. Our predictive algorithm is a tree-based method called OCT that allows the physician to explore the exact model and assess the interpretability of its results. Compared with other binary classification methods, such as Neural Networks that are not explainable [164], OCT is comprehensible and can be easily visualized in a tree form [215]. The final model optimally estimates the probability of stroke with superior performance compared to other stroke risk scores. These findings were validated with a separate multi-ethnic population of 17,527 individuals from an academic medical center.

6.2 Methods

The creation, evaluation, and validation of a new prediction model involves a series of analyses that are necessary to prove its statistical significance. Our methodology comprised the following steps:

- Identification of the derivation and validation cohort and definition of inclusion criteria. Observations were split into the training (75%) and the testing (25%) sets.
- Definition of stroke risk factors and outcomes and association with every participant visit included in the data.
- Imputation of missing values in independent variables using the MedImpute algorithm

[41]. Multiple computational experiments were conducted in order to select the most appropriate missing data imputation method.

- Creation of the Non-linear Framingham Stroke Risk Score (N-SRS) using the OCT algorithm. A risk profile analysis was conducted to validate its insights from the medical literature. The latter was part of an iteration process in tandem with hyperparameter tuning.

- Training of other ML models using a varied set of supervised learning binary classification algorithms, including Logistic Regression.

- Discrimination and calibration performance evaluation of all ML models and the R-FSRS for the testing sets of the derivation and validation cohorts. Separate results summary tables and figures were created for each population.

- Creation of an interactive web-based interface for the communication of the N-SRS model to the clinical community.

6.2.1 The Derivation Cohort

Our study sample comprises the Framingham offspring and the spouses of the offspring cohort enrolled in 1971 and reexamined approximately once every four years since then [122]. To be included, participants were required to be stroke-free and above 40 years of age at each baseline examination. We exclude younger patients following the paradigm of the R-FSRS model [105]. ML methods perform significantly better as the number of the training sample size increases. Thus, we considered for every participant each clinical examination as a distinct observation. We applied the following inclusion criteria:

- The participant had not experienced a stroke event prior to the date of the baseline clinical examination. Patients with prior history of such adverse events receive specific treatment and their future trajectory highly depends on the severity of their primary

stroke. Thus, for these cases we refer the reader to secondary stroke specific risk prediction tools [349].

- The participant was not censored within 10 years from the time of the clinical examination. For every observation we required that (a) either the participant experienced a stroke within the defined time-frame or (b) the participant was censored after the lapse of 10 years.

This methodology of population sampling resulted in the inclusion of 4,385 unique participants, which translated in 18,793 distinct visits (Table 6.1 – Framingham Dataset 1 (FD1)). The dataset was split into the training (75%) and testing (25%) population to allow for unbiased evaluation of the algorithms' performance. Note that visits from the same individual were only included in one of the two sets. Thus, we avoided the introduction of bias in the algorithm evaluation process.

Dataset Name	**Parameter**	**Value**
Framingham Dataset 1 (FD1)	Sample size	18,793
	Number of participants	4,385
	Number of stroke cases	1,013
	Number of distinct participants with stroke	460
	Proportion of female population	53.97%
Framingham Dataset 2 (FD2)	Sample size	2,989
	Number of stroke cases	221
	Proportion of female population	54.26%
BMC - Caucasian	Sample size	9,029
	Number of stroke cases	909
	Proportion of female population	58.63%
BMC - Black	Sample size	2,862
	Number of stroke cases	230
	Proportion of female population	58.97%
BMC - Hispanic	Sample size	5,636
	Number of stroke cases	406
	Proportion of female population	50.19%

Table 6.1: Baseline characteristics of the derivation and validation populations.

Category	Variable
Demographic Factors	Age
	Gender
Categorical Risk Factors	Current cigarette smoking
	Presence of Cardiovascular disease
	Presence of Atrial Fibrillation
	History of Transient Ischemic Attacks
	History of Myocardial Infarctions
	Diabetes mellitus
	Blood Pressure Category
Medication and Treatment related Factors	AHT medication
	Statins
	Nitrates
	Diuretics
	CABG
	PCI
Electrocardiogram (ECG) results	X-ray Enlargement
	Left Ventricular Hypertrophy
	Presence of T-Wave abnormality
	Intraventricular Block
	Atrioventricular Block
	ST-Segment abnormality
	U-Wave abnormality
	Premature beats
Continuous Risk Factors	SBP
	HDL
	BMI
	Hematocrit
	Fasting plasma glucose level

Table 6.2: Stroke Risk Factors considered in the N-SRS model.

6.2.2 The Validation Cohort

The stroke risk model was subsequently validated in a prospective multiethnic cohort of 17,527 patients from the BMC, a private, not-for-profit, 487-bed, academic medical center located in Boston, MA, USA. We identified, using the EHR, a stroke-free population at baseline who satisfied the inclusion criteria without censoring (Table 6.1 – BMC datasets). We retrieved each patient's medical and family history and formulated a dataset that measured the same characteristics as the Derivation Cohort. Every observation in this population corresponds to a unique patient visit. However, no patient was included more than once in the data set. At least 50% of the independent features were known for all selected samples. Missing values were subsequently imputed using a ML algorithm. Prior visits from the same database were used to identify demographic information or data related to the medical and family history of the patient.

6.2.3 Definition of Stroke Risk Factors

We used data collated from each clinical examination including all the risk factors considered in the R-FSRS [105], as well as medication, previous treatment information, ECG results, and additional variables considered in other stroke risk scores [122, 342]. Considering the impact of managing blood pressure levels to the progression of cerebrovascular disease, we hypothesized that the inclusion of treatment specific variables could lead to more personalized stroke risk estimation. A full list of all considered independent variables is presented in Table 6.2. Age, SBP, HDL levels, BMI, hematocrit and fasting blood glucose were treated as continuous features while the rest of the covariates where considered factor variables. SBP was recorded as the mean of two physician recorded measurements made on the left arm of the seated subject, using a mercury column sphygmomanometer and a cuff of appropriate width. Baseline CVD was recorded as present if coronary artery disease, congestive heart failure or peripheral vascular disease had been documented in the participant at, or prior to, the clinical examination. Current cigarette smoking was defined as smoking in the year prior to the baseline examination. We used SBP and DBP measurements to define a new

variable called "Blood Pressure Category" based on current AHA guidelines [63]. We utilized the ECG results provided in each clinical examination of the FHS as additional covariates in our model as well as medical treatment details (i.e. participant underwent CABG or PCI or was under antihypertensive medication at the time et al.). Diabetic status was defined based on the FHS data dictionary similarly to the ECG results. The status of antihypertensive medication was split in two levels (0 = no current prescription of antihypertensive treatment, 1 = currently or in the past under antihypertensive treatment).

6.2.4 Definition of Stroke

Stroke was modeled as a binary outcome and defined as an acute onset focal neurological deficit of vascular etiology, persisting for more than 24 hours, concordant with the World Health Organization (WHO) definition; both ischemic and hemorrhagic strokes were included as in the original FSRS and updated R-FSRS. We used the FHS definition of stroke to specify the outcomes in our dataset; detailed description is defined in previous work [355, 105, 90, 312].

6.2.5 Missing Data Imputation

Missing values were encountered in the majority of the included risk factors. Some participants did not answer the totality of the questionnaires in some of their visits. Moreover, earlier examinations did not record some of the variables, such as echocardiogram results, and thus they were unknown for a subset of the observations [173]. Employing imputation techniques instead of complete case analysis, allows the inclusion of a wider set of features which otherwise would have been omitted by the model [67]. We imputed missing values using a recently developed ML method called MedImpute [41]. The decision to use this algorithm was based on a series of computational experiments that compared both the missing data imputation accuracy as well its effect on downstream predictive performance on these data. It leverages the fact that the same participant could have been included multiple times in the dataset, corresponding to various clinical examinations that satisfied the inclusion criteria. Compared to multiple imputation approaches, such as MICE [341], MedImpute does not

require pooling results that affect the interpretability of the final data set. This methodology has been tested to be robust to the particular missing data patterns which are frequently encountered in longitudinal studies [41]. The algorithm outperformed in both imputation accuracy and downstream prediction performance other standard imputation methods, such as mean [218], k-NN [88], OptImpute [43], MICE [341]. MedImpute reduced the mean absolute imputation error in the Framingham dataset by 5% and increased the AUC in the testing set from 85.21% (MICE) to 87.43%. The authors of the algorithm have also done further experiments using data from the Framingham Heart Study under different missing data regimens, including varying levels of missingness from 10% to 50%, increasing number of observations per participant, and different missing data patterns (MCAR, MNAR) [41]. The method was independently applied to the training and testing sets of the Framingham population as well as the BMC cohort.

6.2.6 Creating the N-SRS

The N-SRS utilizes the OCT algorithm that focuses on both accuracy and interpretability [27, 34] (see also Chapters 3-4). Through this algorithm, we produce a predictive model for 10-year risk of stroke which adaptively changes the splits on the variables, accounting for non-linear interactions among them. The stroke risk is calculated via a series of questions whose order changes dynamically depending on the response. The non-linearity effect is attributed to the absence of a fixed risk coefficient to each independent covariate. The contribution of each feature to the overall score is conditional to other patient characteristics and thus may vary significantly. As we saw in Chapter 5, decision tree methods' final output is very easy to understand, and thus appropriate for applications where interpretability is important. Its structure allows predictions through a few decision splits on a small number of high-importance variables, contrary to other ensemble approaches and neural networks [164, 152].

The selection of the final model involved an iterative process during which a risk profile analysis was conducted for each path of the tree. Every path is associated with a unique set

of risk factors whose interaction and significance was validated from the medical literature. We trained other well-established ML algorithms (i.e. CART, RF, GBT) on the derivation population data to have a fair comparison of the OCT performance in addition to the R-FSRS results [56, 55, 73]. Logistic regression with L1 regularization (Log.Reg) is also employed to specify the performance of a linear model using the same features, data format and missing data imputation as the N-SRS [163]. We used 10-fold cross-validation to set the parameters for each model. The OCT maximum depth was set to eight and the minimum bucket to 20 observations.

6.2.7 Measurement of Model Performance

The OCT algorithm performance and its ability to predict 10-year risk of stroke was measured using the AUC [157]. We report the average performance across five random partitions of the data with replacement in the derivation population. For each random split, a distinct training sample was used to create the predictive models. Their performance was subsequently evaluated on both the testing sets of the Framingham cohorts as well as the BMC validation cohort. Confidence intervals (95%) were calculated for the bootstrapped results. We also report the average sensitivity, specificity, precision, negative predictive value, positive predictive value for all cohorts and methods when the probability threshold is set to 0.5. In addition, we compare the Hosmer-Lemeshow calibration χ^2 statistic to measure how closely the outcomes predicted by a given model approximate the observed outcomes [93]. We used three different datasets to measure the performance of the prediction models, including the R-FSRS. In the first set of experiments, we evaluated each model's outcomes using the testing set of the Framingham Dataset 1 (FD1). The FD1 includes all the clinical examinations of the offspring cohort that satisfied the inclusion criteria but did not participate in the model training process. The Framingham Dataset 2 (FD2) comprises of the observations that the R-FSRS used for its development (Table 6.1 - FD2). We carefully split the dataset such that observations used in the FD2 are only part of the testing set of FD1. As a result, all reported metrics refer to out-of-sample results. The FD2 does not include any samples from the FD1

training set. We subsequently compared the performance of the N-SRS with the R-FSRS on the validation cohort (Table 6.1 - BMC) against the same metric.

6.2.8 The User-Friendly Interface

Leveraging the tree nature of the final N-SRS, we built a dynamic online application as the user-friendly interface of the algorithms for use by clinical providers (`http://www.mit.edu/~agniorf/files/questionnaire_Cohort2.html`). The application is in the form of an interactive questionnaire. The questions are adaptive corresponding to risk factors; the subject of each new question depends on the answer to the prior question. When all questions are answered, the user receives the final risk estimate of stroke for the particular patient. The software follows the same interface as the POTTER score, which has been already implemented at the Massachusetts General Hospital, for the estimation of emergency surgery mortality and morbidity risk, with great success [37]. Due to its format, the application could be integrated into an EHR environment, pulling the most available variables directly from the database in an automated fashion. Once integrated into the EHR, the user would only be required to answer questions that cannot be pulled in automatically. If there is full EHR automation, the risk would be calculated at once.

6.3 Results

A comprehensive decision-making algorithm was designed, and a user-friendly model, the N-SRS was created using the training set of FD1; a total of 14,195 clinical examinations (75%) from the Framingham offspring cohort. Figure 6.1 provides a visualization of our model in a tree structure. While each node of the tree model reveals important information regarding the associated risk of patients, it should not be considered in isolation. On the contrary, the final risk profile of individuals should be based on the full path until the final "leaf" node of the tree model. Thus, we identify 23 different stroke risk profiles, all of which highlight the effect that these factors might impose in the risk of stroke while introducing new non-linear

relationships when combined. Each profile follows a different path of the tree and is affected only by the risk factors that appear in that path (Figure 6.1).

6.3.1 N-SRS Performance on the Framingham Datasets

Table 6.3 demonstrates the superior performance of the N-SRS compared to the R-FSRS calculator and other established ML methods in both the FD1 and FD2. Notice, that the OCT approach is significantly more accurate compared to the R-FSRS approach leading up to a 15% AUC improvement in FD1 and 9% in FD2 populations, both for male and female. Moreover, the results indicate equivalent performance with respect to other less interpretable ML methods (GBT, RF) in the testing set since the absolute difference in the AUC is less than 1%. The Log.Reg models achieve better performance compared to the R-FSRS improving the out-of-sample discrimination metric by 9.37% and 3.55% in FD1 and FD2 respectively. Non-linear ML methods, though, demonstrate superior predictive power that is up to 7.81% (5.06%) higher in the FD1 (FD2) cohorts. The ranking of the methods in terms of downstream performance remains intact between the two datasets. Similar conclusions are also reflected on the sensitivity, specificity, precision, negative and positive predictive value metrics.

Most importantly, the N-SRS is able to better estimate the true risk of stroke, at different levels of risk. Its Hosmer-Lemeshow calibration χ^2 statistic is 1.96/2.75 (FD1/FD2) for 8.05/7.3 the N-SRS and R-FSRS respectively. We constructed calibration curves for our models, where best performance is represented by a slope of 45 degrees. The R-FSRS models suffered a decline in calibration, especially at medium risk predicted probabilities. The N-SRS classifier appeared to have the best calibration across all levels. The calibration curves are depicted in Figure 6.2 for the N-SRS and R-FSRS.

6.3.2 N-SRS Performance on the Validation Cohort

Table 6.4 shows an overview of the results for the N-SRS, R-FSRS, and other ML methods on the Validation Cohort. The non-linear approach (N-SRS) improves the aggregated stroke risk AUC by 16.17% for men and 10.59% for women upon the R-FSRS. Similar results are

A) Framingham Dataset 1 (FD1)

	N-SRS	R-FSRS (both genders)	R-FSRS (men)	R-FSRS (women)	Log.Reg	CART	Random Forest	GBT
Sensitivity	0.9142	0.8510	0.8461	0.8554	0.8933	0.8802	0.9175	0.9167
Specificity	0.7238	0.6902	0.6890	0.7043	0.7102	0.7099	0.7161	0.7354
Precision	0.9408	0.9620	0.9353	0.9758	0.9701	0.9736	0.9605	0.9412
NPV	0.0592	0.0380	0.0647	0.0242	0.0423	0.0380	0.0863	0.0588
PPV	0.9408	0.9620	0.9353	0.9758	0.9621	0.9736	0.9137	0.9412
AUC	0.8743	0.7374	0.7188	0.7552	0.8065	0.7981	0.8829	0.8846
AUC 95% CI	0.8569-0.9014	0.6976-0.7619	0.6765-0.7636	0.7081-0.8102	0.772-0.8351	0.7676-0.8287	0.8578-0.9081	0.8643-0.9048
calibration χ^2	1.96	8.05	11.98	5.44	2.88	3.04	1.43	1.58

B) Framingham Dataset 2 (FD2)

	N-SRS	R-FSRS (both genders)	R-FSRS (men)	R-FSRS (women)	Log. Reg	CART	Random Forest	GBT
Sensitivity	0.8948	0.8533	0.8605	0.8487	0.8763	0.8504	0.8938	0.8934
Specificity	0.5097	0.4217	0.4066	0.4800	0.4867	0.2505	0.4994	0.5110
Precision	0.9693	0.9617	0.9531	0.9712	0.9688	0.9393	0.9816	0.9700
NPV	0.3973	0.2233	0.2321	0.1933	0.2576	0.1704	0.3804	0.4053
PPV	0.9693	0.9617	0.9531	0.9712	0.9401	0.9486	0.9535	0.9700
AUC	0.8238	0.7488	0.7281	0.7677	0.7754	0.6884	0.8216	0.8260
AUC (95% CI)	0.791-0.8558	0.7145-0.7831	0.6775-0.7788	0.7149-0.8204	0.738-0.8119	0.6435-0.7333	0.7881-0.8536	0.7938-0.8567
calibration χ^2	2.75	7.3	12.1	4.1	6.5	20.34	2.81	2.7

Table 6.3: Comparison of the N-SRS, the R-FSRS, and other ML methods performance on the testing set of the Framingham datasets. Reported metrics include sensitivity, specificity, precision, negative predictive value (NPV), and positive predictive value (PPV) at the probability threshold of 0.5. The Table also presents the overall AUC and calibration χ^2 results.

also recorded in the ethnicity-specific populations. We notice that both stroke risk scores are less accurate in the BMC dataset compared to FD1 and FD2 (-7.09% N-SRS, -8.00% R-FSRS). However, the N-SRS is more robust to other sources of data. Its performance is less affected compared to the R-FSRS. The performance of other ensemble ML algorithms is equivalent to the N-SRS providing an edge of 0.8-0.91%. The Log.Reg models improve upon the R-FSRS by 5.55% but is still weaker than the N-SRS by 3.38%. Table 6.4 shows that the predictive accuracy of our model remains the same between the Caucasian and the Black population (74.5%) and gets slightly negatively impacted in the Hispanic population (72.8%). All other ML models achieve higher performance in the Caucasian sample compared to other ethnicity sub-populations.

The calibration statistic demonstrates an edge of N-SRS (7.12) over the R-FSRS for both women (35.98) and men (37.42), following a similar trend to what was shown for the Framingham datasets. Figure 6.2 shows that the R-FSRS is associated with poor identification of true risk for groups higher than 30%. In terms of sensitivity and sensitivity, we found that

the N-SRS model achieved up to 89% and 40%, respectively while R-FSRS achieved 84% and 36.6%.

	N-SRS	R-FSRS (both genders)	R-FSRS (men)	R-FSRS (women)	Log.Reg	CART	Random Forest	GBT
Sensitivity	0.8986	0.8403	0.8411	0.8396	0.8576	0.8402	0.9055	0.9076
Specificity	0.4019	0.3663	0.3786	0.3565	0.3733	0.3599	0.4078	0.4092
Precision	0.9395	0.9320	0.9329	0.9313	0.9349	0.9348	0.9407	0.9455
NPV	0.2771	0.1815	0.1882	0.1762	0.2026	0.1805	0.2811	0.2818
PPV	0.9395	0.9320	0.9329	0.9313	0.9345	0.9317	0.9421	0.9446
AUC	0.7403	0.6491	0.6246	0.6735	0.7065	0.6829	0.7482	0.7501
AUC (95% CI)	0.7149-0.771	0.6266-0.6716	0.5931-0.6555	0.6411- 0.7058	0.6772-0.7558	0.6484-0.7175	0.7198-0.7801	0.7202-0.7856
calibration χ^2	7.12	36.66	37.42	35.98	25.03	35.76	6.67	6.52

Table 6.4: Comparison of the N-SRS, the R-FSRS, and other ML methods performance on the Validation Cohort. Reported metrics include sensitivity, specificity, precision, negative predictive value (NPV), and positive predictive value (PPV) at the probability threshold of 0.5. The overall AUC and calibration χ^2 results are also presented. The results refer to the aggregated population.

	BMC-White		**BMC- Black**		**BMC - Hispanic**	
Model	**AUC**	**95% CI**	**AUC**	**95% CI**	**AUC**	**95% CI**
N-SRS	74.30%	0.7149-0.771	75.80%	0.7345-0.767	72.79%	0.6889-0.7671
R-FSRS (both genders)	64.91%	0.6266-0.6716	64.85%	0.6304-0.6666	61.04%	0.5601-0.6587
R-FSRS (women)	67.35%	0.6411-0.7058	65.22%	0.628-0.6764	61.06%	0.5548-0.6663
R-FSRS (men)	62.46%	0.5931-0.6555	64.49%	0.6181-0.6717	61.01%	0.5621-0.6587
Log.Reg	71.55%	0.6823-0.7402	69.77%	0.6823-0.7402	70.46%	0.6765-0.7359
CART	69.01%	0.6627-0.7134	66.41%	0.6272-0.6609	66.10%	0.6286-0.6934
RF	75.08%	0.7162-0.7855	73.14%	0.7139-0.749	70.80%	0.6807-0.7354
GBT	77.32%	0.7582-0.7881	74.88%	0.7133-0.7842	74.27%	0.7187-0.7667

Table 6.5: Comparison of the N-SRS, and the R-FSRS performance on the Validation Population using the AUC metric. Detailed results are shown for the main ethnicity groups.

6.4 Discussion

To the best of our knowledge, this is the first validated non-linear, interpretable stroke risk predictor that outperforms the established R-FSRS, providing additional insightful information. Overall, our results demonstrate the superior capability that sophisticated ML methods and data utilization can bring in adverse event prediction when coupled with data from large population cohorts. In our ever-changing medical landscape, linear models

that entail an additive effect for each known risk factor do not answer many practical questions faced by patients. Patients with multiple medical comorbidities may not be reflected with traditional risk stratification scores such as the FSRS. The N-SRS methodology has introduced novel risk factors that are associated with stroke incidence. Moreover, a "one size fits all" approach may not work for a particular patient. Although correlative, the superior interpretability of the model can allow for better patient education when addressing risk factor modification strategies.

Khosla et al and colleagues have previously demonstrated the superiority of ML over cox-hazard methods for stroke prediction with an AUC as high as 0.777 utilization patient data from 5201 patients from the cardiovascular heart study between 1989-1999. Several novel risk factors were identified using this methodology including total medications, maximal inflation level, general health and any ECG abnormality [188]. In contrast to this paper, our methodology utilized interpretable OCT and utilized a robust data set (the Framingham heart study) therefore risk factors where more specific (T-wave abnormality on EKG as compared to "any ECG abnormality) making its utility more relevant.

Other novel ML methods have evaluating stroke risk in specific high-risk populations. Letham et al., developed and interpretable and accurate model for stroke risk prediction in patients with atrial fibrillation utilizing the Bayesian Rule List (BRL) model in contrast to the established linear prediction scores; the CHADS2 and CHA2DS2-VASc risk scores [207]. In this study, claims data from the MarketScan Medicaid Multi-State Database was utilized to study a patient with diagnosis of atrial fibrillation (one year of observation time prior to the diagnosis and one year of observation time following the diagnosis) yielding 12,586 patient with 1786 (14%) suffering a stroke within a year of the atrial fibrillation diagnosis. The BRL performance had a higher performance by AUC as compared to the CHADS2, CHA2DS2-VASc and CART methods (0.756 vs. 0.721, 0.677 and 0.704) respectively. However, as known with claims data and coding, the true interpretability of this methodology is questionable. For example, the BRL states: "if cerebrovascular disorder then stroke risk 47.8% (44.8%–50.7%) else if transient ischemic attack then stroke risk 23.8% (19.5%–28.4%) else if occlusion and

stenosis of carotid artery without infarction then stroke risk is 15.8% (12.2%–19.6%)". These terms are non-specific and descriptive at best and do not mean anything from a physician perspective. The terms transient ischemic attack and occlusion and stenosis of carotid artery without infarction are both similar clinically, and interchangeable from a coding perspective and cannot be used to risk stratify adequately.

Primary prevention targeting stroke risk factors have been effective in reducing stroke morbidity and mortality in generalized populations (46). However, they do not consider the potential to predict which of the risk factors would affect each individual and lead to stroke occurrence; a key element in practical disease prevention, targeted therapy and the most compelling finding of our study. Our approach introduces tree-based decision rules where the number of variables required to determine the stroke risk profile is not fixed by our preconceived understanding of comorbidities and attributable risk [58].

The N-SRS model was developed using the Framingham data, a well-established longitudinal data set in contrast to static datasets typically utilized for risk prediction [263]. The model established several key branching points in the tree that confirm the medical validity of this model as well as novel points uncovering new medical insights that had not been evaluated for stroke risk in the past. It also demonstrates the correlation of interplay between risk factors and weighted relevance they may possess in contrast to the binary effect they carry.

The model was validated using an external independent cohort comprised of diverse ethnicities. Our results revealed a superior performance of the N-SRS over the R-FSRS in the training and validation population for both women and men. Additional experiments show that other less transparent non-linear algorithms achieve equivalent performance. Logistic regression models using the same data pre-processing and training sample improve upon the N-SRS but do not outperform more sophisticated ML methods. We hypothesize that the performance of the latter is improved compared to the R-FSRS due to the higher sample size, larger number of features, and the application of an advanced missing data imputation algorithm. Since the accuracy of the N-SRS was higher and more robust to populations from

other ethnicities, our model can be generalized with higher degree of confidence compared to the existing stroke risk score. We believe that the increased accuracy of N-SRS is due to the introduction of a larger sample size, new risk factors, and new missing data imputation and binary classification methodologies.

Our proposed way of leveraging the longitudinal study data avoids the induction of bias in the model due to its clear delineation between the training and the testing population. We strictly require that observations from the same individual belong in at most one of these two sets, avoiding potential natural boosts in the downstream performance. Moreover, our results from the multi-ethnicity validation cohort of the BMC demonstrate that the N-SRS generalizes better than its predecessor (R-FSRS).

The main benefit of using decision trees over other methods is their interpretability which, in applications such as healthcare. This attribute is not only essential but often preferred over the maybe higher accuracy that other, non-interpretable, methods may offer [121]. In our models, we show that less transparent, "black-box" algorithms have comparable performance to our suggested model. The latter offers the physician the opportunity to evaluate the risk profile itself and assess the correlation of risk factors relevant for each patient. It also addresses concerns related to the transparency and fairness of the model [238].

Known findings that appeared as branching nodes in the N-SRS include patients with the lowest stroke risk profile being non-diabetic with HDL levels > 39.1 mg/dl and non-hypertensive with an approximately 1% 10-year stroke risk. In contrast, patients with history of cardiovascular disease, diabetes and hypertension carry a 90% stroke risk over 10 years (Figure 6.1). Of note, these modifiable risk factors weigh heaviest and are independent of other concomitant factors or non-modifiable ones such as age or gender. In fact, the relevance of gender was only pertinent in a subset of patients with no cardiovascular disease or diabetes but with hypertension and low HDL levels.

Note that in some cases to characterize the risk of stroke for certain profiles of the population only three to two variables might be relevant. For people with no history of cardiovascular disease and diabetes, smoking affects dramatically their risk projection

increasing the overall stroke score from 29.73% to 82.66% (Figure 6.3). We notice also that for patients with prior history of cardiovascular disease diabetes is the defining factor of their stroke risk increasing it to 71.05% from 31.95% (Figure 6.3). The presence or absence of any other risk factor does not influence the overall prediction of the ML algorithm.

An illustration of novel findings includes the relevance of T-wave abnormality on ECG and hematocrit levels in a patient's 10-year stroke risk profile. For example, the association of major and minor ST-T wave abnormalities on ECG and associated stroke risk has been previously evaluated in a small cohort of Japanese patients but found to be only relevant in men with minor ST changes and both genders for major ST changes based on the small sample size. Furthermore, stroke risk was reduced after adjusting for hypertension [253]. Therefore, the applicability is minimal in evaluating preventative strategies and guiding patient education or intervention. In the N-SRS model, T wave abnormalities were pertinent in some scenarios. A characteristic case refers to patients with history of cardiovascular disease, non-diabetic, with 0-1 MI events, and HCT levels of $<$38.2% where the 10-year stroke risk changes from 32% to 65% in the absence or presence of T-wave abnormalities respectively (Figure 6.3).

Such assessments of risk factors and their respective weighted relevance could not be established by linear methodologies and can explain innumerable circumstances where patients may have or lack traditional risk factors and either develop strokes or not. This is the key to personalizing a customized approach to primary prevention.

For instance, the N-SRS shows that the 10-year stroke risk is actually dramatically impacted by smoking changing from 5% to 77.5%. If this patient was not hypertensive in the first place, her 10-year stroke risk would be 2.5% and smoking would not drive this number (Figure 6.3). This validated risk prediction can highly impact the patient and provider understanding of stroke risk factor associated with incidence for effective guided counseling given the precious resources and time available to practitioners and patients.

Although this is the first validated interpretable ML model applied to stroke for 10-year risk prediction, similar applications in other disease entities provided insights obscured by

traditional linear methodology and therefore influence personalized care. Bertsimas and colleagues recently evaluated outcomes of 13 different medication regimen therapies in over 10,000 patients with type 2 diabetes and predicted change in target glycated hemoglobin A1c levels [29]. In this model, patients where a suggested change in therapy based on the machine algorithm was made, a predicted reduction by close to 0.5% points in Ac1c was observed. Similar mortality and morbidity risk calculators have also been introduced in the areas of elective surgery, oncology, and transplantation with great success [36, 37, 39]. Such ML-based algorithms can drive personalized medicine and influence outcomes.

We have created an interactive web-based interface through a series of short specific yes and no questions to improve efficiency and usability of the N-SRS decision-tree (Figure 6.4). A user's answer to the first question will dictate what the next grouping the results into 23 categories of risk profiles. Each interaction with the application corresponds to a unique decision-tree node and is based on the specific patient characteristics.

As a second phase of this study, we intend to prospectively follow a patient population in the primary care setting utilizing the N-SRS to guide preventative strategy. In this prospective study, we will not only be able to study real-time prospective stroke risk, but also a completely novel experience of personalized stroke risk assessment care and intervention. This has not been effectively studied in patients at risk for cerebrovascular disease and opens many potential possibilities for other cerebrovascular diseases other than stroke.

Limitations

The key limitation of our model is the use of input data solely from the FHS which is a predominantly Caucausian population. Moreover, there is potentially lack of generalizability to populations from other geographic regions in the United States as well as internationally, and socioeconomically different populations from those of the FHS or BMC. Even though we validate our results in a multi-ethnicity population, we believe that we will need to retrain our algorithm with data from other longitudinal studies and not only EHR.

The validation population is based solely on hospital records and as a result it tends to

be sicker than the Framingham cohort. Each observation corresponds to a unique patient visit. Thus, the presence of a patient in the data set is mostly correlated with how detailed was the clinical examination during the visit and if there was any family or personal history recorded in the past at the same clinic.

In addition, we would like to stress that our data is not independent and identically distributed. However, we believe that no bias has been introduced in the training process since both the accuracy and the calibration of the N-SRS is significantly higher than the R-FSRS in both the Framingham 2 dataset and the validation cohort from BMC. Another limitation refers to causality between the variables and the outcomes, which is still not proven despite the high degree of association connectivity between the two. The performance of N-SRS has not been directly compared to other stroke risk functions, such as the CHADS2 or the CHA2DS2-VASc score for atrial fibrillation stroke risk [136]. Future work could leverage other validation populations to relate the N-SRS predictive performance with these studies.

We also acknowledge prospective validation of this model would outperform validation of blinded data sets, and provide insights beyond performance such as adoption among healthcare providers, interpretability for patients and effects on primary prevention strategies and counseling. A prospective trial design is currently under evaluation.

6.5 Conclusions

We have developed N-SRS, an accurate stroke risk calculator that outperforms, in accuracy and user-friendliness, the existing stroke risk prediction tool. N-SRS might prove useful as an evidence-based, adaptive, and interactive risk calculator tool for primary prevention of stroke. Further studies are needed to explore the ability of N-SRS to predict the occurrence of stroke in other populations. Future work will focus on defining the N-SRS risk levels that warrant therapeutic treatment for primary stroke prevention similar to that available for the primary atherosclerotic cardiovascular disease prevention.

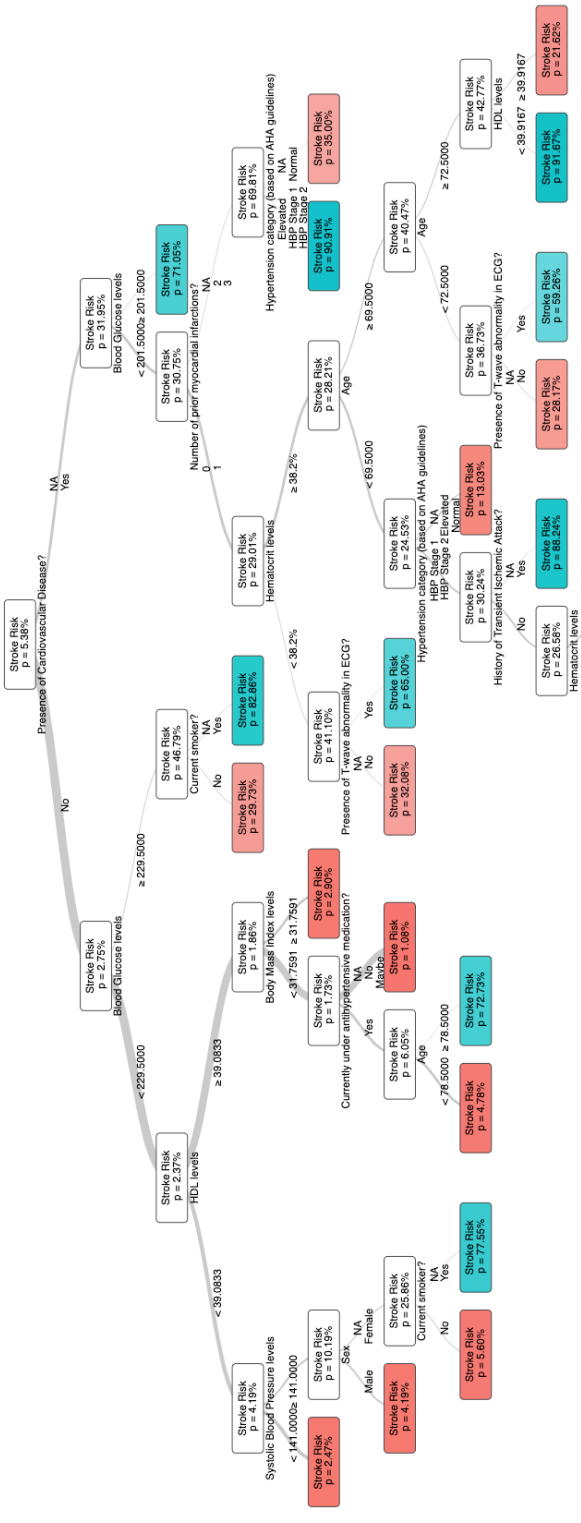
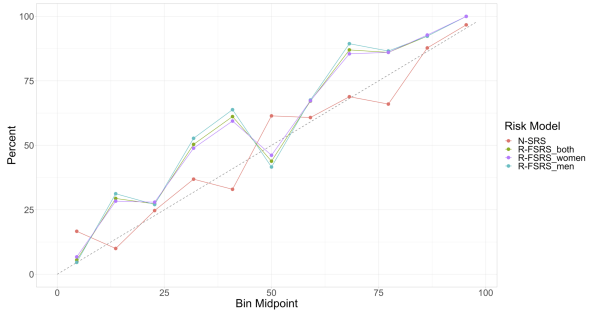
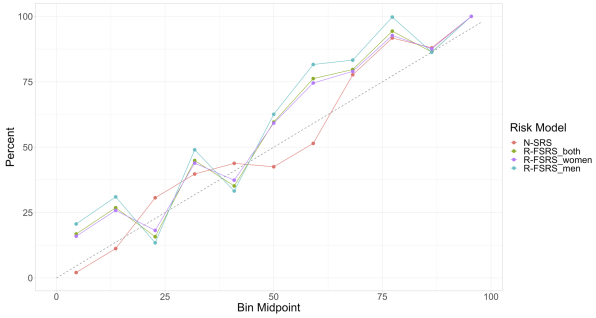


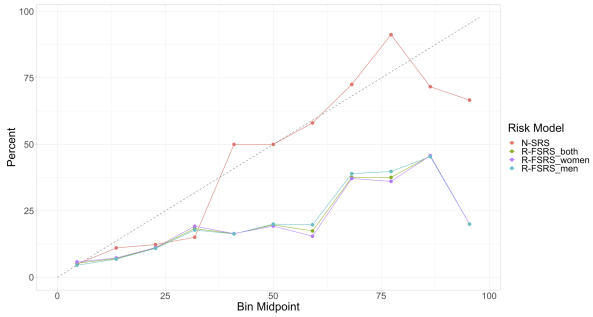
Figure 6.1: A visualization of the N-SRS tree-based model.



(a) Framingham Dataset 1 (FD1)



(b) Framingham Dataset 2 (FD2)



(c) Validation Cohort (BMC-Aggregated)

Figure 6.2: Calibration plots for all models on the Derivation and Validation Cohorts. The plots show the relation between the true class of the samples and the predicted probabilities. Samples were binned to their class probabilities generated by the model. The following intervals were defined: [0,10%], (10,20%], (20,30%], ... (90,100%]. The event rate for each bin was subsequently identified. For example, if 4 out of 5 samples falling into the last bin are actual events, then the event rate for that bin would be 80%. The calibration plot displays the bin mid-points on the x-axis and the event rate on the y-axis. Ideally, the event rate should be reflected as a 45 degrees line.

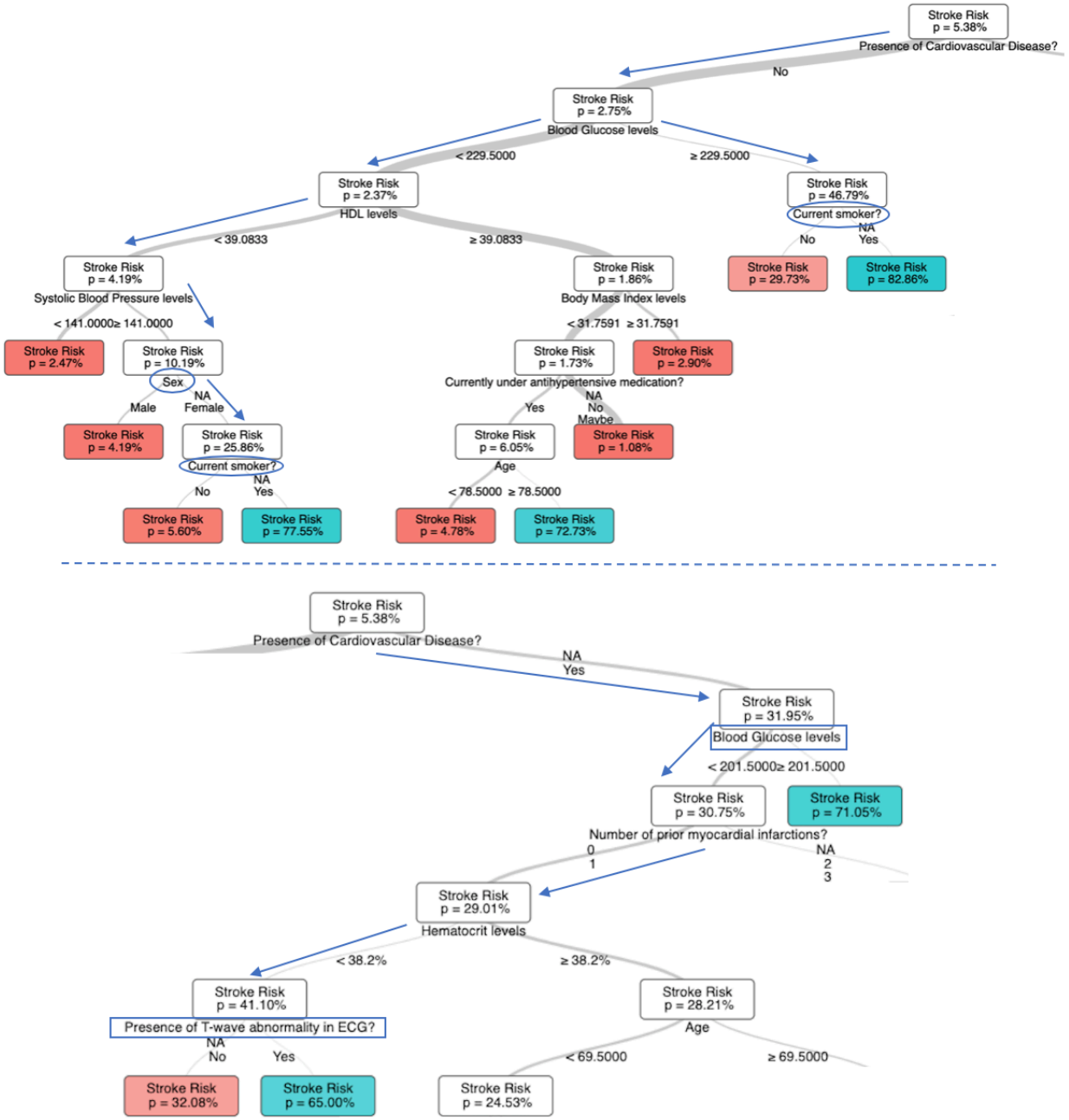


Figure 6.3: Deep-dives in insightful risk profiles of the N-SRS model.

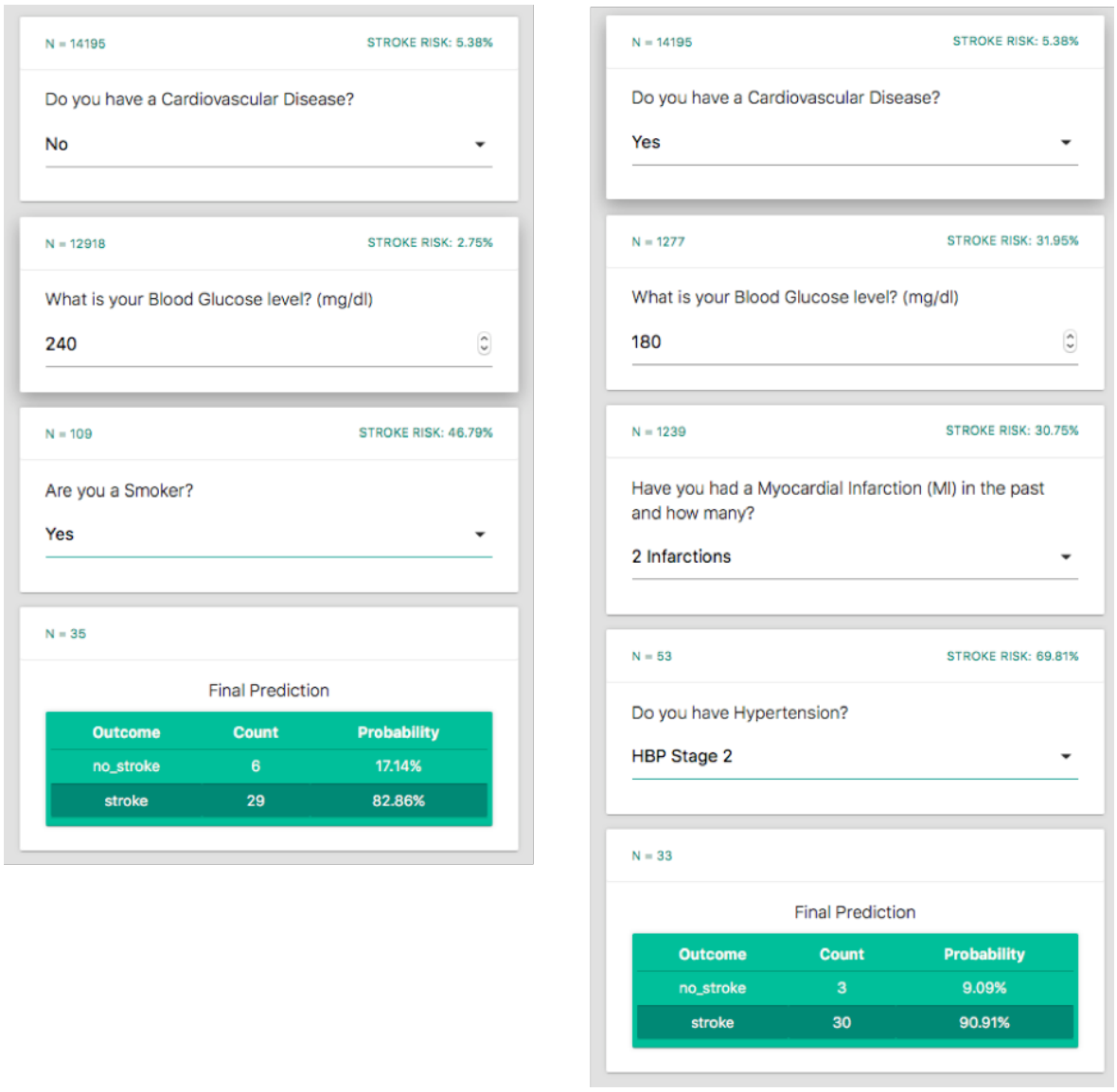


Figure 6.4: An example illustrating the user-friendly interface of N-SRS. Due to its interactive nature the answer to a question dictates the next question. In this specific example, depending on whether the provider answer yes to no to the question regarding CVD, the algorithm and the questions take a different direction.

Chapter 7

Natural Language Processing Techniques for Stroke Identification from Radiology Reports

Accurate, automated extraction of clinical stroke information from unstructured text has several important applications. ICD-9/10 codes can misclassify ischemic stroke events and do not distinguish acuity or location. Expeditious, accurate data extraction could provide considerable improvement in identifying stroke in large datasets, triaging critical clinical reports, and quality improvement efforts. In this study, we developed and report a comprehensive framework studying the performance of simple and complex stroke-specific NLP and ML methods to determine presence, location, and acuity of ischemic stroke from radiographic text. We collected 60,564 Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) Radiology reports from 17,864 patients from two large academic medical centers. We used standard techniques to featurize unstructured text and developed neurovascular specific word GloVe embeddings. We trained various binary classification algorithms to identify stroke presence, location, and acuity using 75% of 1,359 expert-labeled reports. We validated our methods internally on the remaining 25% of reports and externally on 500 radiology reports from an entirely separate academic institution. In our internal population,

GloVe word embeddings paired with deep learning (Recurrent Neural Networks (RNN)) had the best discrimination of all methods for our three tasks (AUCs of 0.96, 0.98, 0.93 respectively). Simpler NLP approaches (Bag of Words) performed best with interpretable algorithms (Logistic Regression) for identifying ischemic stroke (AUC of 0.95), MCA location (AUC 0.96), and acuity (AUC of 0.90). Similarly, GloVe and RNN (AUC 0.92, 0.89, 0.93) generalized better in our external test set than Bag of Words (BOW) and Logistic Regression for stroke presence, location and acuity, respectively (AUC 0.89, 0.86, 0.80). Our study demonstrates a comprehensive assessment of NLP techniques for unstructured radiographic text. Our findings are suggestive that NLP/ML methods can be used to discriminate stroke features from large data cohorts for both clinical and research-related investigations.

7.1 Introduction

Radiographic findings on head CT or MRI are frequently used to support or confirm the diagnosis of ischemic stroke in clinical practice. Radiologists interpret images in narrative reports that detail stroke occurrence and other pertinent information including acuity, location, size and other incidental findings. Because of their unstructured nature, radiology reports do not make it easy to employ these information-rich data sources for either large-scale, retrospective review, or for real-time identification of stroke in the clinical workflow. The ability to automate the extraction of meaningful data from radiology reports would enable quick and accurate identification of strokes and relevant features such as location and acuity. Such a system could help clinicians triage critical reports, target patients eligible for time-sensitive interventions or outpatient follow up, and identify populations of interest for research [369].

NLP is a field that spans multiple scientific disciplines including linguistics, computer science, and artificial intelligence. The main objective of NLP is to develop and apply algorithms that can process and analyze unstructured language. A distinctive subfield of NLP focuses on the extraction of meaningful data from narrative text using ML methods [225]. ML-based NLP involves two steps: text featurization and classification. Text featurization converts

narrative text into structured data. Examples of text featurization methods include BOW, Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings [225, 270]. Word embedding methods, including Word2Vec and GloVe [270, 267, 235], learn a distributed representation for words. The result of these methods is a numerical representation of text that can be subsequently used for myriad applications. One particular medical application of these methods is the classification of salient findings from unstructured radiographic reports. After converting language into relevant binary or continuous features through text featurization, supervised classification models can separate reports into desired categories (i.e. presence or absence of acute middle cerebral artery stroke). These models are trained on a portion of the cohort, and then tested on unseen data to determine how accurately they classify observations. Previous efforts to automate diagnoses from radiologic text have resulted in algorithms that can identify pneumonia, breast cancer, and critical head CT findings [369, 270]. Specifically, Zech and colleagues found that simpler featurization and classification techniques perform comparably to more sophisticated deep learning approaches in identifying binary critical head CT classifiers (i.e. critical v. non critical; ischemia v. no ischemia) [369]. However, clinicians and radiologists use diverse language patterns to characterize stroke features. For instance, "subacute" is a relative term and can describe strokes that occurred anywhere from hours to months prior to the diagnostic study. Specific descriptions of ischemia on head CTs (i.e. hypodensities or sulcal effacement) or MRIs (decreased Apparent Diffusion Coefficient (ADC)) provide clinicians with more context that allows them to infer timing, severity and likely diagnosis.

We hypothesized that simpler NLP featurization approaches that rely on counting how many times a relevant word occurred in text, like BOW or TF-IDF, may not sufficiently capture the language describing stroke features. Word-embedding approaches that account for word relationships might better identify characteristics of interest. In this study, we aimed to: 1) expand the application of NLP to identify both the presence of ischemia and relevant characteristics including location subtype, and acuity; and 2) compare whether a neurovascular-tailored NLP featurization algorithm (GloVe) outperforms simpler methods

(BOW, TF-IDF) in identifying key qualifying characteristics.

7.2 Methods

7.2.1 Study Population

We collected 60,564 radiology reports consisting of head CT, or CT Angiography (CTA) studies, brain MRI, or MR Angiography (MRA) studies from a cohort of 17,864 patients over 18 with ICD-9 diagnosis codes of ischemic stroke (433.01, 433.11, 433.21, 433.31, 433.81, 433.91, 434.01, 434.11, 434.91 and 436) from 2003-2018 from the Research Patient Data Registry (RPDR), at Massachusetts General and Brigham and Women's Hospitals [179]. We chose these four imaging modalities because a generalizable algorithm that identifies stroke characteristics from multiple imaging report subtypes would have greater practical application. We externally validated our best performing classification methods on 500 radiographic reports from 424 patients who were admitted to Boston Medical Center between 2016-2018. Boston Medical Center is the largest safety-net hospital in New England, and thus has a markedly different racial-ethnic and socioeconomic population than our training cohort. The Partners Human Research Committee and Boston Medical Center local IRBs approved this study.

7.2.2 Manual Radiographic Report Labeling

1,359 original radiology reports from 297 patients (883 Head CTs or CTAs, 476 MRIs or MRAs) were hand-labeled by study team members trained by attending physicians and/or senior staff members. Each report included the text, type of scan (CT, MRI, CTA, or MRA), date, and report time. Reports were distributed randomly among the labelers. Each reporter independently labeled 1) the presence or absence of ischemic stroke, 2) middle cerebral artery (MCA) territory involvement, and 3) stroke acuity. Stroke occurrence, acuity, and MCA location were classified as either present or absent. Labelers identified "stroke" if the report definitively reported a diagnosis of ischemic stroke or if ischemic stroke was determined as

probable by the labeler based on the radiology report text. A stroke was labeled as acute if: the reporting radiologist reported it as acute in their report, diffusion restriction or apparent diffusion coefficient hypointensity without T2 prolongation was mentioned on MRI report, or it was interpreted as having occurred within the last 7-10 days. MCA stroke location was defined as a reported MCA territory or thrombus in MCA with corresponding symptoms in the history section of report. We focused on the identification of MCA stroke as this stroke subtype is particularly clinically actionable via thrombectomy and at high risk for stroke sequelae including edema and hemorrhagic transformation. Study data were collected and managed using a Research Electronic Data Capture (REDCap) electronic database [160]. Each report was separately labeled twice. Any discrepancies between the two labels were reviewed by attending neurologists CJO or SS. If labelers felt that identification of stroke occurrence or characteristics were indeterminate, they were labeled as absent. A board-eligible Attending Neurocritical Care physician (CJO) conducted a blinded analysis, and then adjudicated 300 radiology reports by review of images. In an assessment of 10% of the final reports labels from both the derivation and external cohorts by a trained physician and labeler (HS), percent agreement for stroke presence, MCA location, and acuity were 91%, 87%, and 93%, respectively, suggesting good to excellent inter-rater reliability. Additionally, a board-certified Neurologist and Neurointensivist (CJO) assessed the percent agreement of 300 reports and raw images. She found percent agreement for presence of ischemic stroke, MCA location, and acuity were 97%, 95%, and 98%, respectively. The most common cause for discrepancies resolved upon adjudication included small chronic strokes, strokes referred to in the report that were only identified on a prior scan, or subtle early changes that were consistent with symptoms listed in the report and available to the radiologist.

7.2.3 Text Preprocessing and Featurization

To remove basic non-uniformities in unstructured text data, we used the following steps to preprocess radiology reports for further analysis.

1. We removed any incomplete reports, header text (i.e. patient or visit information,

procedure details), non-diagnostic standardized language (i.e. names or electronic provider signatures), non-narrative text including "======".

2. We converted commonly used word groups, "word tokens", to "n-grams", or a single word group unit without spaces. For example, middle cerebral artery was converted to: "middlecerebralartery".

3. We standardized all whitespace, removed punctuation and converted all text to lowercase.

After preprocessing, narrative text was "featurized" to convert unstructured data into classifiable, numeric information for a ML algorithm [224, 144]. We compared simple traditional text featurization methods (BOW, TF-IDF) with a recent word embedding technique trained on neurology-specific text. The specific featurization techniques used in our analysis are detailed below:

1. Bag of Words (BOW): Bag of words is the simplest model for text featurization, disregarding context, semantic proximity and grammar. Each word, or grouping of words (n-gram) in the main corpus/body of the text is considered a distinct feature. The value of each feature corresponds to the number of times a word was found in a given report.

2. Term Frequency-Inverse Document Frequency (TF-IDF): The term frequency-inverse document frequency method (TF-IDF) re-weights document features based on the relative importance of the word in the text [225]. Weighting of words is positively correlated to the number of times a word appears in a given document, but is offset by frequency in the training corpus.

3. Global Vectors for Word Representation (GloVe): GloVe is a word-embedding method that quantifies how often pairs of words co-occur in some window in a given text, since these frequencies are likely to have semantic meaning [267]. For example, the pairs of terms "ice"-"solid" and "steam"-"gas" co-occur much more frequently than pairs "ice"-"gas" and "steam"-"solid." Exact frequencies depend on the specific training set GloVe uses.

7.2.4 Radiologic Stroke Featurization Training Corpus for GloVe

Since standard widely available text corpora do not provide frequent exposure to our concepts of interest (i.e. ischemic stroke), and more specifically the likely co-occurrence of word pairs relevant to stroke, we developed a neurovascular specific corpus to train our GloVe featurization algorithm, including:

1. The complete set of neurology articles on UpToDate^{TM}, to capture general neurologically focused medical language [130].

2. Stroke, Pathophysiology Diagnosis and Management, to capture stroke-specific language [233].

3. Yousem's Neuroradiology: The Requisites, to capture neuroradiology specific language [368].

4. A random sample of 10,000 radiology reports from 2010-2017, separate to our testing and training set, to capture language specific to radiology reports of all types.

This training resulted in the first neuroradiology specific set of vector representations, which we made available for other clinical NLP applications and can be found at this link: `http://www.mit.edu/~agniorf/files/Glove_Neurology_Embeddings.csv`. Our GloVe model parameters included word vector dimension of 100, number of iterations of 50, a window size of 10, and a learning rate of 0.05.

7.2.5 Report Classification

To classify the radiology reports for our three outcomes of interest 1) presence of stroke, 2) stroke location (MCA territory), and 3) stroke acuity, we created predictive models using logistic regression, k-NN, CART, OCT with and without hyperplanes (OCT-H), RF, and RNN [161, 82, 56, 27, 34, 55, 196]. Our analysis leverages a wide range of traditional state-of-the-art algorithms including linear regression, tree-based, ensemble, and Neural Networks (NN) models. The choice of RNN among the various types of NN structures was based on prior

research in the NLP field that indicated superior performance when applied to sequential text [196, 367]. RNN coupled with LSTM gates allow for back propagation of information, and thus are able to leverage the order of words in the text [164]. In the derivation cohort, we reported results across a comprehensive combination of all text featurization and predictive techniques outlined above. We performed further external validation using 500 "unseen" reports from an additional medical center, leveraging our two combinations of text featurization techniques and binary classification algorithms. Specifically, we report the performance of interpretable, simple models that use Logistic Regression with BOW and the more complex RNN models coupled with neurology-specific GloVe embeddings.

For validation of our models, we used a grid search and 10-fold cross-validation to select the appropriate values of tuning parameters for all binary classification algorithms. Our parameters for model development included the selection of the regularization term λ, using a maximum of 1000 iterations and a tolerance threshold of 0.01 for logistic regression and the k parameter for the k-NN algorithm from the range of [5, 10, 15, 20]. We selected minimum bucket and maximum depth parameters for tree-based methods across a range of 1-10, and used AUC, entropy, gini, and misclassification accuracy to refine and select the final model. The maximum number of greedy trees for RF was set to 200. Our RNN model used an LSTM network with two hidden layers, including a layer of sentence vectors, and a second layer in which the relations of sentences are encoded in document representation [281].

We trained our models on 75% of the original cohort of 1,359 reports and tested on a withheld test set of 25% for internal validation. For our derivation cohort, we used bootstrapping to randomly split the data five times into training and testing sets. The entire external validation cohort was tested across all five splits of the data. To evaluate model performance on both cohorts, we compared discrimination by reporting the AUC with confidence intervals. We also reported sensitivities, specificities, accuracy, precision, and recall. In the derivation cohort for each prediction task, we report the latter metrics only for the best performing method (GloVe/RNN). For both the internal and external validations, we prioritized sensitivity, and chose a threshold in which sensitivity of $>$90% produced the

highest specificity. For each outcome on our derivation cohort, we evaluated the models' calibration using calibration curves. Moreover, we selected the two best performing classifiers and compared them using the McNemar test [97]. A 2-sided P-value of 0.05 was considered significant. Similar to other NLP studies, we used this test to validate the hypothesis that the two predictive models are equivalent [330]. We report the average performance across all five partitions of the data for each evaluation criterion. Confidence intervals were calculated for the bootstrapped results.

7.3 Results

Of 1,359 hand-labeled reports from 297 patients in the derivation cohort, 925 had ischemic strokes, 350 were labeled as "MCA territory" and 522 were labeled as acute. 129 patients were female (43%), and median age at report time was 68 years [IQR 55,79]. In the validation cohort, 500 reports were used from 424 patients with a median age of 69 [IQR 59,79] at report time. The sample included 192 female patients (45%). After labeling, 266 reports were classified as strokes, 90 as "MCA territory" and 106 were characterized as acute.

We compared performance of multiple text featurization and classification methods to classify our outcomes of interest. For stroke, MCA location, and acuity, we observed best discrimination using our developed GloVe word embedding and RNN classifier algorithm with AUC values of 0.961, 0.976, and 0.925 respectively (Table 7.1).

Stroke							
Average AUC (95% CI)	**Logistic Regression**	**k-NN**	**CART**	**OCT**	**OCT-H**	**RF**	**RNN**
BOW	0.951	0.808	0.889	0.805	0.915	0.922	0.838
	(0.943:0.959)	(0.767:0.848)	(0.868:0.91)	(0.774:0.836)	(0.899:0.92)	(0.902:0.942)	(0.811:0.866)
TF-IDF	0.939	0.857	0.883	0.813	0.894	0.929	0.843
	(0.933:0.945)	(0.825:0.889)	(0.859:0.907)	(0.801:0.825)	(0.853:0.906)	(0.909:0.948)	(0.816:0.869)
GloVe	0.904	0.867	0.734	0.722	0.767	0.892	0.961
	(0.889:0.918)	(0.836:0.898)	(0.703:0.765)	(0.69:0.753)	(0.775:0.834)	(0.868:0.916)	(0.955:0.967)
Location							
Average AUC (95% CI)	**Logistic Regression**	**k-NN**	**CART**	**OCT**	**OCT-H**	**RF**	**RNN**
BOW	0.959	0.841	0.949	0.867	0.937	0.96	0.896
	(0.944:0.974)	(0.816:0.867)	(0.93:0.969)	(0.838:0.896)	(0.919:0.955)	(0.943:0.978)	(0.873:0.926)
TF-IDF	0.962	0.903	0.944	0.862	0.934	0.965	0.956
	(0.943:0.981)	(0.873:0.933)	(0.918:0.97)	(0.828:0.896)	(0.917:0.951)	(0.947:0.983)	(0.936:0.977)
GloVe	0.906	0.843	0.734	0.699	0.809	0.873	0.976
	(0.884:0.927)	(0.819:0.868)	(0.677:0.791)	(0.662:0.722)	(0.787:0.83)	(0.854:0.892)	(0.968:0.983)
Acuity							
Average AUC (95% CI)	**Logistic Regression**	**k-NN**	**CART**	**OCT**	**OCT-H**	**RF**	**RNN**
BOW	0.898	0.815	0.797	0.735	0.797	0.901	0.754
	(0.874:0.922)	(0.775:0.854)	(0.748:0.846)	(0.705:0.764)	(0.742:0.852)	(0.883:0.919)	(0.733:0.779)
TF-IDF	0.893	0.857	0.801	0.733	0.807	0.902	0.899
	(0.865:0.921)	(0.826:0.888)	(0.762:0.839)	(0.703:0.764)	(0.764:0.843)	(0.876:0.923)	(0.875:0.922)
GloVe	0.881	0.842	0.73	0.719	0.82	0.866	0.925
	(0.842:0.92)	(0.805:0.879)	(0.684:0.776)	(0.66:0.778)	(0.766:0.873)	(0.824:0.908)	(0.894:0.955)

Table 7.1: Average AUC Performance across five splits of the data for Natural Language Processing and Classification Methods Considered on the Derivation Cohort.

For simpler tasks, like the identification of stroke, Logistic Regression combined with BOW performed comparably to more complex word embedding methods (AUC of 0.951 with Logistic Regression/BOW vs. AUC of 0.961 with GloVe/RNN). However, the difference in discrimination was larger for more nuanced features like acuity (AUC of 0.898 for Logistic Regression/BOW vs. 0.925 for GloVe/RNN). The word embedding approach did not perform as well when paired with logistic regression or single-decision tree methods (Table 7.1).

Receiver Operator Curves (ROCs) are included in Figure 7.1. We constructed calibration curves for our models, where best performance is represented by a slope of 45 degrees, and the three best classifiers are included in Figure 7.2. RF classifiers suffered a decline in calibration, especially in the MCA location task at high predicted probabilities. GloVe/RNN methods appeared to have the best calibration across tasks.

Outcome	**Sensitivity**	**Specificity**	**Accuracy**	**Precision**	**Threshold**
Stroke	0.902	0.872	0.892	0.935	0.69
MCA Location	0.902	0.911	0.908	0.766	0.42
Acuity	0.911	0.689	0.772	0.935	0.33

Table 7.2: Sensitivity, Specificity, Accuracy and Precision for GloVe Models combined with RNN.

In terms of accuracy, (the fraction of reports from both positive and negative classes that are correctly classified), we found that GloVe/RNN models achieved up to 89 and 91% for stroke presence and MCA location, respectively. Corresponding sensitivities and specificities were both high (0.90 and 0.87 for stroke presence) and (0.90 and 0.91 for MCA location). For the acuity task, while we prioritized sensitivity (0.91), accuracy was less (0.77), reflecting the greater difficulty of this classification (Table 7.2).

Finally, we used McNemar's test to compare our best performing GloVe model with the best performing simpler NLP model for each task. Specifically, we compared GloVe/RNN with the second-best performing combination of supervised learning and text featurization technique. For the presence of stroke task, Logistic Regression coupled with BOW had a χ^2 value of 4.79 (p=0.056). For both the location and acuity outcome, we used the models of TF-IDF/RF and showed that both had equivalent performances, 14.78 (p=0.023) and 26.74 (p=0.031) respectively.

In our external validation cohort, we tested our most sophisticated (GloVe/RNN) and the simplest (BOW/Logistic Regression) methods. We found that BOW/Logistic Regression (AUCs 0.89, 0.86 and 0.80 respectively for stroke, location, acuity) did not generalize as well as GloVe/RNN (AUC 0.92, 0.89, 0.93) in the external population (Table 3). We continued to

prioritize sensitivities in the external validation population for GloVe/RNN (0.90-0.92), and specificities were decreased for stroke and MCA location (0.75, 0.70) compared to the internal validation population (0.87, 0.91). Specificities remained the same (0.69) for the acuity task.

Method	Stroke	Location	Acuity
BOW+Log.Reg	0.892 (0.875:0.91)	0.857 (0.845:0.869)	0.797 (0.768:0.828)
GloVe+RNN	0.920 (0.908:0.932)	0.893 (0.88:0.905)	0.925 (0.906:0.946)

Table 7.3: Average AUC metric across all five splits of the data on the Validation Cohort across all outcomes for BOW with Logistic Regression and RNN with GloVe.

7.4 Discussion

Accurate automated information extraction will be increasingly important as more medical researchers, hospital systems, and academic institutions leverage "big data" from electronic medical records. Unlike structured, discrete data like laboratory values or diagnoses codes, unstructured text is challenging to analyze. However, clinicians frequently record essential observations, interpretations, and assessments that are otherwise absent from the remainder of the medical record. In order to fully leverage our ability to access such data through the medical record, we must have validated methods to extract meaningful information. Specific to radiology reports, there are several important applications of accurate automated extraction of information through NLP. Automatic, real-time identification of specific subpopulations (such as patients with acute MCA stroke) can improve clinical workflow and management by triaging eligible patients to timely treatments or higher levels of care [270]. NLP approaches

Outcome	Sensitivity	Specificity	Accuracy	Precision	Threshold
Stroke	0.915	0.752	0.828	0.764	0.3
MCA Location	0.898	0.7	0.862	0.932	0.85
Acuity	0.914	0.689	0.866	0.916	0.9

Table 7.4: Sensitivity, Specificity, Accuracy and Precision for GloVe Models combined with RNN on the BMC Validation Cohort.

can facilitate research by identifying both populations (i.e. patients with stroke, tumor or aneurysms) and outcomes (i.e. presence of hemorrhagic conversion or edema) more feasibly than manual review, and potentially more accurately than billing codes. Indeed, of our 1,359 radiographic reports derived from patients with billing codes of stroke, only 925 (68%) had a radiographically reported ischemic stroke, which raises the question as to whether NLP can assist in improving diagnostic classification. In this study, we developed a comprehensive framework to create a vector-based NLP method specifically targeted to identify stroke features from unstructured text. We then tested the ability of multiple ML methods to classify specific stroke features and compared performance. We designed our study to identify these three tasks separately as opposed to a single task ("acute middle cerebral artery stroke") because our objective was to create an NLP identification system that can be expanded to multiple stroke types in the future.

We found that NLP methods perform well at extracting featurized information from radiology reports (AUCs >0.9 for all three tasks). True to our hypothesis, word-embedding methods like GloVe improved overall accuracy of feature identification, especially when paired with deep learning methods like RNN, which are less interpretable (harder to distinguish features contributing to performance) than simpler classification algorithms like logistic regression or single-decision trees. However, RNN's have been particularly successful in NLP applications, where the sequence of words in the text can crucially alter the overall meaning of the corpus [144]. Because the field of NLP is rapidly expanding, variations of featurization methods are used and trialed for different purposes. We chose to use BOW, TF-IDF and GloVe because they were representative of the simplest, the most frequently used, or an innovative word-embedding approach that better captures semantic meaning, respectively.

We acknowledge that there are various widely accepted word embedding techniques, such as Word2Vec, the Distributed memory (DM)-document vector (DV) model, the continuous bag of words (cBOW) model, the continuous skip-gram (CSG) model, and FastText [369, 235, 181]. Recently, investigators also proposed a hybrid method, called Intelligent Word Embedding (IWE), that combines semantic-dictionary mapping and a neural embedding technique

for creating context-aware dense vector representation of free-text clinical narratives [17]. However, our aim was to demonstrate whether a neurology-specific embedding model could improve upon simpler techniques that do not consider context and semantic meaning in their word representations. Given the significant computational resources required for the creation of the embeddings and prior research demonstrating equivalence between the algorithms' objectives, we limited our analysis to one word embedding technique [313]. We chose to use GloVe because this approach outperformed other word-embedding methods, and has been shown to do so with smaller training sets, which is important when considering how our contributions may be applied to other investigators for research and/or clinical use [267].

This investigation is part of a wider literature that employs deep learning in clinical NLP [360]. In this study, we employ a specific RNN structure that had been previously and successfully used in combination with GloVe embeddings [196]. An increasing number of deep learning structures are being employed in similar applications such as autoencoders [211, 205], deep belief networks [213], memory residual NN [300], and attention mechanisms like BERT [95]. Future research directions could focus on leveraging these other NLP structures with neurology-specific embeddings and comparing their performance.

Our work is consistent with other studies reporting simple methods like BOW are suitable for extracting unstructured text information. One group found that BOW paired with lasso logistic regression had high performance (AUCs of >0.95) for critical head CT findings [369]. Kim and colleagues' found that a single decision tree outperformed more complicated support vector machines in identifying acute ischemic stroke on MRIs [189]. Garg and colleagues used various ML methods to classify stroke subtype from radiology reports and other unstructured data [137]. They achieved a kappa of 0.25 using radiology reports alone, which improved to 0.57 when they used combined data. In our study, our GloVe embedded vector approach was specifically tailored for the detection of vascular neurologic disorders, and outperformed other methods in correctly classifying stroke acuity, particularly when paired with a neural network structure. Additional analysis also demonstrated that general purpose embeddings such as the ones trained only on Wikipedia provide significantly lower

performance. Namely, an RNN classifier achieved 0.74 (0.70:0.75) AUC for presence, 0.75 (0.72:0.79) AUC for location, and 0.693 (0.61:0.73) AUC for acuity of stroke - a decrease of at least 0.2 in discriminatory performance compared to our proposed embeddings. These results emphasize the need for radiographic-specific word representations that capture the semantic relations of medical vocabulary. Because RNNs account for word order, we expect these methods will be increasingly used for accurate natural language processing of medical text data.

Limitations: There are several important limitations to our work. Similar to other studies, our radiology corpus consisted of reports from only two hospitals, which may reduce our generalizability in other systems. Also, the use of both CT and MRI reports increases heterogeneity for model development; however, given the finite number of ways in which reports describes stroke characteristics regardless of imaging modality, we sought to test a method that could be widely applied to radiographic text.

Strengths and Future Directions: Strengths of our study include the development of a tailored word-embedding approach to vascular neurologic disorders, the development of multiple models testing the optimal combination of NLP and classification algorithms, generalizability to both CT scans and MRIs, its external validation in a racial-ethnic and socio-economically diverse cohort, and the ability to expand this framework to additional stroke characteristics (increased locations, hemorrhagic conversion). While our word-embedding approach was specifically tailored to neurovascular disorders, similar approaches could be used to generate word vectors for other disease states, including oncology and cardiology. Moreover, while our data extraction of unstructured text focused on radiology reports, further work in this area could assist in the retrieval of essential information in progress notes, and interrogation of discrepancies in the medical record that result from "copy/paste". As we gather more electronic data on patients, easy information retrieval will become increasingly important as a strategy to scale research and improve quality.

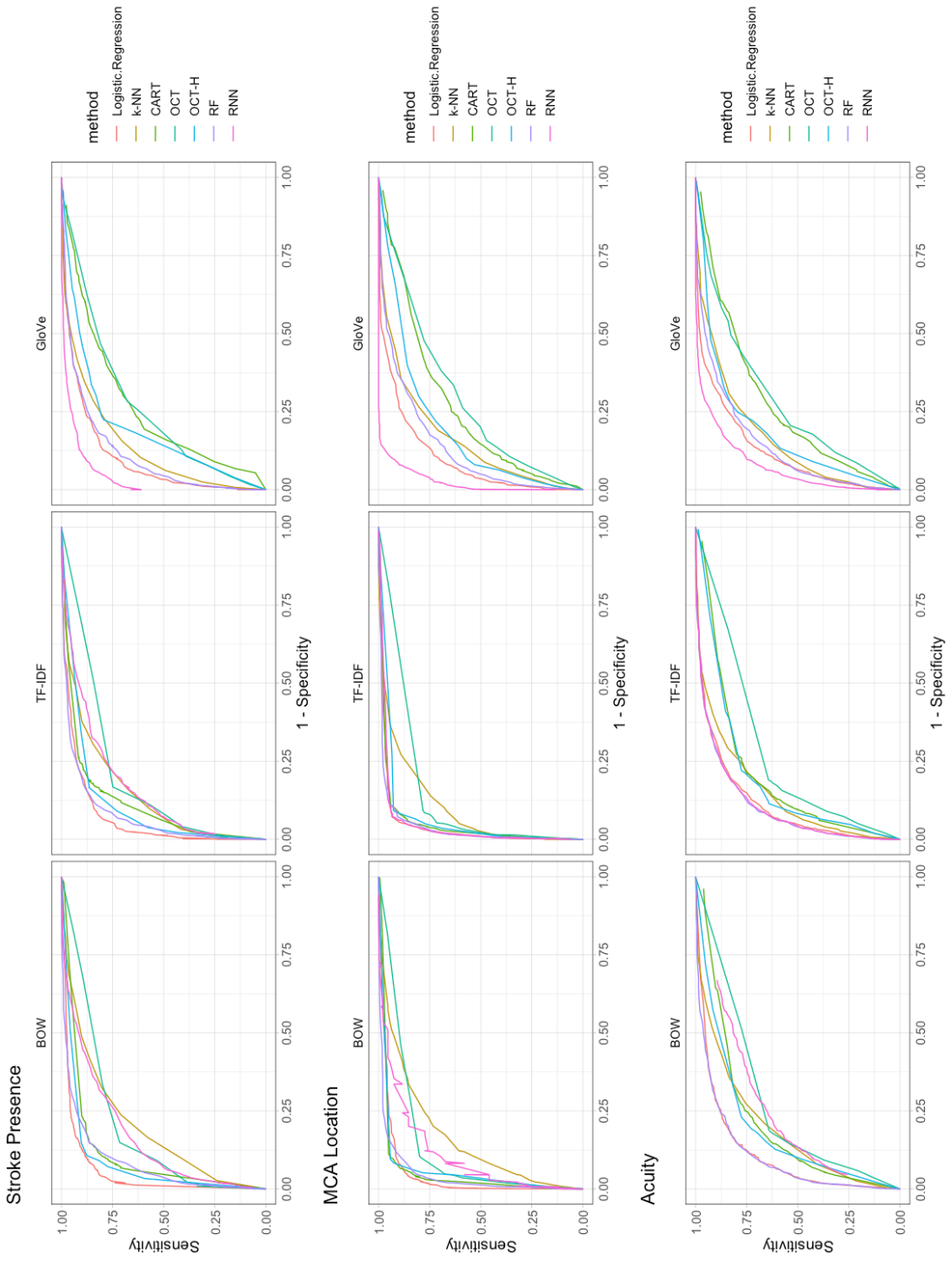


Figure 7.1: Receiver Operating Curves for NLP classification. A, stroke presence; B, MCA location; C, acuity. We present the average the mean sensitivity and specificity over five random splits of the data.

7.5 Conclusions

Automated ML methods can extract diagnosis, location and acuity of stroke with high accuracy. Word-embedding approaches and RNNs achieved the best performance in correct classification of stroke and stroke characteristics. Our results provide a framework for expeditiously identifying salient stroke features from radiology text that can triage high-risk imaging findings and identify patient populations of interest for research. Future directions include improving performance through the study of hybrid rule-based and ML methods. Work in this area is particularly important as accurate, accessible methods to automate data extraction will become increasingly relevant for academic, tertiary, and non-tertiary centers who aim to improve clinical, administrative, and quality care.

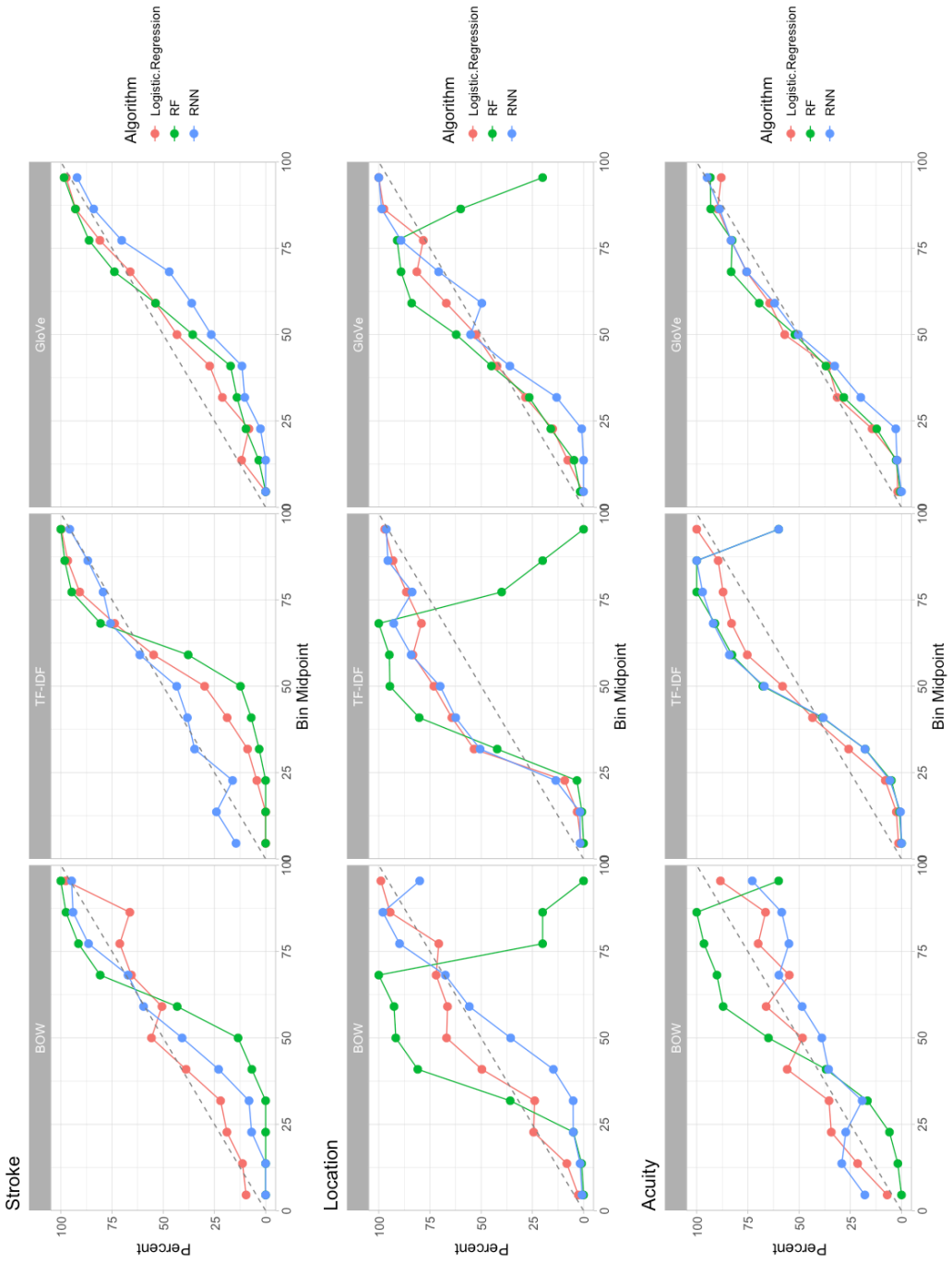


Figure 7.2: Calibration Curves for NLP classification. We binned the samples according to their class probabilities generated by the model. We defined the following intervals: [0,10%], (10,20%], (20,30%], ... (90,100%]. We subsequently identified the event rate of each bin. The calibration plot displays the bin mid-points on the x-axis and the event rate on the y-axis. Ideally, the event rate should be reflected as a 45 degree line. We present the average the mean sensitivity and specificity over five random splits of the data. We show results of the three best performing methods in each task.

Part III

Algorithmic Insurance

Chapter 8

The Cost of Algorithmic Risk

As ML algorithms start to get integrated into the decision-making process of companies and organizations, insurance products will be developed to protect their owners from risk. We introduce a quantitative framework for insurance companies and ML modelers to price the risk of these products. Using properties of the model, such as accuracy, interpretability, generalizability and robustness, we provide a mathematical formulation for the model's financial evaluation. We present a case study of medical malpractice in the context of breast cancer detection where we estimate the risk exposure of a binary classifier.

8.1 Introduction

Data-driven analytical models and ML algorithms have started transforming large facets of the economy, becoming the driver of innovation in digital marketing, self-driving cars and medical imaging, among others. While the use of Artificial Intelligence (AI) expands across all segments of society, algorithms are expected to replace human judgement in many cases [78]. However, there is still a lot of resistance in employing these tools in practice as different types of concerns are emerging about algorithmic decision-making [69]. Dietvorst et al. (2015) defined this phenomenon as *algorithm aversion*, identifying a wide range factors, including ethical issues, related to the use of algorithms that are not resolved yet [98, 99]. One of the

persistent challenges related to the implementation of ML models in practice is centered around the question of responsibility in case of erroneous algorithmic decisions making [320].

This is not the first time that modern societies are called to face such dilemmas. Equivalent situations haven gained spotlight in the past, such as when cars became accessible to the broader consumer audience in urban areas after World War I. By that time, motor vehicles had become fast and less expensive and manufacturers were expecting an increasing portion of society to enter the market. Nevertheless, the vast majority of households were still reluctant to adopt the technology as it could entail disproportionately high financial risks. In the absence of insurance policies, injured victims would seldom get any compensation in an accident, and drivers often faced considerable costs for damage to their car and property. To ensure that all vehicle owners and drivers can be protected against the risk of causing injury or death to third parties, the Road Traffic Act was established in 1930 in the United Kingdom, introducing the first compulsory car insurance scheme [310]. Over the years, legal questions related to responsibility became inextricably linked to third party liability. In general terms, third party liability insurance provides protection against claims resulting from physical injuries to people and/or damage to property [284]. Nowadays, liability coverage has become so important that it is often required for automotive insurance policies, product manufacturers, and anyone who practices medicine or law.

As ML algorithms are expected to replace human decision-making in cases where their predictive and prescriptive performance yields better outcomes, a new type of liability insurance could be developed to protect their owners from risk. Potential examples are self-driving cars (third party liability) or image recognition systems for MRI machines (medical liability). If there were legal contracts that individuals and organizations could sign to protect themselves from algorithmic mistakes, the adoption and implementation of ML tools would be significantly faster and less contentious [25].

In this chapter, our goal is to show how liability insurance can be extended in the case of erroneous algorithmic decision-making. The prerequisites of this type of contracts are, as in all cases of insurance, (i) an agreement between the parties, (ii) the existence of a risk to the

insured party or potential third parties, (iii) the payment of a premium [24]. The premium is determined as a function of the assumed risk. Hence it varies according to both the likelihood of its occurrence (frequency), and the magnitude of the consequences that may arise once it materializes (severity). Thus, the main focus of the insurance industry is the determination of an appropriate premium pricing strategy for every policy. The pricing of traditional non-life insurance products is usually conducted using the fundamental principles of actuarial science and asset pricing theory [89, 113]. Leveraging historic observations from prior realizations of risk, underwriters are able to create probabilistic models that estimate the potential risk exposure, the relative frequency of adverse events, and associate an appropriate price. The key challenge with machine-based decisions lies in the absence of such data.

At the time of initial implementation of ML-driven decision-making tools, very limited information regarding historical algorithmic mistakes would be available. For example, since fully autonomous cars have not been deployed yet, it is impossible to collect information with respect to past accidents they have caused. Although, the corresponding claims cost associated with these accidents will likely be known from previous litigation cases of human-based decision-making, the challenge will arise when trying to estimate the risk of incident realization. Nevertheless, there is a paradox in this statement. Self-driving cars, like any other AI system, are based on the use of data; past observations that participated in the model training and validation process. Though there are no claims data from prior liability cases, we have at our disposal large datasets that are used to train the underlying ML model associated with the outcome of interest. By leveraging this valuable resource and other properties of the models, we propose a novel quantitative framework that enables pricing the risk of these products. We propose a series of tools for insurance companies and data scientists that will allow them to evaluate the litigation risk associated with the implementation of a ML algorithm under different scenarios.

8.1.1 Contributions

We introduce for the first time a data-driven model that quantifies the risk and the associated premium for insurance contracts protecting against adverse events that may result from erroneous algorithmic decision-making. Our contributions can be summarized as follows:

1. We propose an optimization formulation that leverages measures of risk from the financial literature to simultaneously estimate the risk exposure and price for a given binary classification model. We extend the formulation using robust optimization to different types of uncertainty sets around historical scenarios of loss.

2. We estimate the expected financial loss due to algorithmic liability based on the predictive performance, the interpretability and the generalizability of a binary classifier.

3. We introduce a data-driven approach to apply the proposed pricing framework. We provide a case-study for medical liability and demonstrate the potential effect of the model parameters in simulated experiments.

The structure of this chapter is as follows. In Section 8.2, we introduce a case study of medical liability in the context of breast cancer detection. In Section 8.3, we present the baseline optimization formulation that leads to the simultaneous estimation of price and risk exposure. In Section 8.4, we incorporate the predictive accuracy of a binary classification model into the estimation process of future expected loss. In Section 8.5, we demonstrate how to include a ML algorithm's interpretability into the pricing strategy. Section 8.6 focuses on data generalizability and its effect on the premium determination. In each of the Sections 8.4-8.6, we use the medical liability case study to showcase a practical implementation of the framework and highlight the effect of the model's parameters. In Section 8.7, we discuss the key findings from the computational experiments, the limitations of the framework, and future applications of the proposed approach. We conclude in Section 8.8.

8.2 A Case Study of Medical Liability for Malignant Tumor Detection

In this section, we introduce our case study of interest, focusing on algorithmic insurance for medical liability. We will delve into breast cancer, a carcinoma that is estimated to affect one out of eight women at some point in their lifetime [309]. According to the American Cancer Society, more than 250,000 women are diagnosed with invasive breast cancer every year in the US [319]. Due to widely established screening policies and improved therapies, breast cancer has now one of the lowest mortality rates among carcinomas [23]. Nevertheless, medical malpractice relating to breast cancer and breast imaging remains common and costly for both radiologists and healthcare organizations involved [204]. In the future, the widespread implementation of AI algorithms is expected to improve the diagnostic accuracy and reduce diagnostic errors in carcinomas [190, 361].

Given the transformative role that ML can play in this application area, we focus on a case study for breast cancer detection to illustrate a practical implementation of the proposed pricing framework. First, we describe the dataset that we use to train the underlying predictive model. Subsequently, we present an overview of historical medical malpractice lawsuits for breast cancer detection.

8.2.1 Data Description

For our analysis, we will use the Breast Cancer Wisconsin Diagnostic dataset from the UCI ML Repository [103]. The features of the dataset represent characteristics of the cell nuclei of a breast mass [328]. This information was acquired from digitized images of Fine Needle Aspirate (FNA) analyses. FNA biopsies are recommended to women who are suspected to suffer from breast cancer. During this procedure, a small amount of breast tissue or fluid is taken from the suspicious area and is checked for cancer cells. The dataset contains ten features related to the cell nucleus of each sample, including radius, texture, perimeter, area, smoothness, degree of compactness, concavity, presence concave points, symmetry, and fractal

dimension. The outcome of interest is whether the sample belongs to a benign or malignant tumor.

8.2.2 Medical Malpractice Lawsuits for Breast Cancer

Lawsuits involving breast cancer are the most common cause of medical malpractice litigation in the United States [352]. An analysis from credentialing data of 8401 radiologists, revealed that breast cancer was the most frequently missed diagnosis, followed by nonvertebral fractures and spinal fractures [352]. Breast cancer imaging lawsuits involve physicians from multiple specialties, radiology being the most common. Lee et al. (2020) identified 253 cases in the US from 2005 to 2015 that resulted in plaintiff payment where the average award amount was $978,858. The median award amount in cases with a verdict was $862,500 with interquartile range (IQR) ($500,000 to $2,009,460) while, in cases that concluded with a settlement, it was as high as $1,162,500 with IQR ($17,000 to $2,000,000).

In a separate study conducted from 1995 to 1997, 218 surgical pathology and FNA claims were reviewed. Breast FNA corresponded to 6% of those records while breast biopsy accounted for another 14% [9]. 54% of breast biopsy claims referred to false-negative diagnoses of breast carcinoma, whereas 35% were for the false positive diagnosis of cancer, demonstrating the importance of high sensitivity and specificity for the binary classification model [201]. Malpractice claims from false positive FNA s are usually attributed to wrong interpretations by the medical team. The most common case is when a fibroadenoma is misclassified as carcinoma resulting in unnecessary mastectomy or axillary node sampling if breast conservation is elected [9]. Such cases can result in malpractice claims of more than $800,000 [295, 140].

In the following sections, we analyze the impact of predictive performance, interpretability, and generalizability on the insurance contract. Our goal is to understand the effect of the model parameters and highlight the algorithmic aspects that may significantly affect the risk exposure of an insurance contract.

8.3 Quantifying Risk Exposure

We assume that the determination of the insurance premium is a function of how much risk the organization or the modeler is willing to assume. Our goal in this section is to quantify the expected risk exposure of the contract based on data regarding the severity and frequency of the losses.

In our framework, we estimate risk exposure using tools from the finance literature. We resort to two well established statistical techniques used to measure the level of financial risk within a firm or an investment portfolio over a specific time frame; the Value-at-Risk (VaR) and the Conditional-Value-at-Risk (CVaR). VaR answers the question of what is the maximum loss with a specified confidence level [180]. Though VaR is a widely accepted and used measure, it cannot be considered a *coherent* metric [14]. It does not have the sub-additivity property, it is non-convex, non-smooth, and it may result to multiple local extrema [290]. As a result, optimizing this measure under different constraints is quite complex.

For this reason, we will focus on CVaR, an alternative measure of financial risk that does not bear the weaknesses of VaR [339]. CVaR is both a coherent and consistent measure of losses that does not violate the sub-additivity and complexity property [14]. While VaR represents a worst-case loss associated with a probability and a time horizon, CVaR is the expected loss if that worst-case threshold is ever crossed.

Suppose that we break our problem in n cases, corresponding to the different classes of claims that we insure. For example, referring to the medical malpractice problem, we could classify our patients into three age groups $([0, 30), [30, 50], [50, 120))$, in which case $n = 3$. Let $f(\mathbf{x}, \mathbf{y})$ be a loss function that takes as inputs a decision vector of premiums $\mathbf{x} = (x_1, \ldots, x_n)$ and a random vector of losses $\mathbf{y} = (y_1, \ldots, y_n)$. The decision vector $\mathbf{x}$ belongs to a feasible set of prices $\mathbf{X}$. The loss function $f(\mathbf{x}, \mathbf{y})$ is then equal to the difference between the price and the future loss in case there is a claim:

$$f(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{n} \max\{0, y_j - x_j\}. \tag{8.1}$$

Uryasev (2000) proposed a framework for the simultaneous calculation of VaR and CVaR as well as the determination of the associated premium based on the distribution of the losses [339]. The author noticed that in the real-world we do not have the analytical representation of the density function of $\boldsymbol{y}$ to estimate the probability that the loss function does not exceed a threshold value α. However, using past observations, we have access to scenarios $\mathbf{y}_j$, with $j \in [J]$, sampled from the density $p(\mathbf{y})$. Thus, they proposed the following CVaR approximation:

$$\widetilde{F}_\beta(\mathbf{x}, \alpha) = \alpha + \nu \sum_{j=1}^{J} (f(\mathbf{x}, \mathbf{y}_j) - \alpha)^+, \tag{8.2}$$

where β is the a given confidence level, $\nu = \frac{1}{(1-\beta)J}$ and α is the variable that represents VaR. If $f(\mathbf{x}, \mathbf{y})$ is convex w.r.t. to $\mathbf{x}$, then $\widetilde{F}_\beta(\mathbf{x}, \alpha)$ is a convex non-smooth function w.r.t. to $(\mathbf{x}, \alpha)$. Moreover, if $f(\mathbf{x}, \mathbf{y})$ is linear w.r.t. to $\mathbf{x}$, then we can use linear optimization to solve our problem of interest:

$$\begin{aligned} \text{minimize} \quad & \alpha + \nu \sum_{j=1}^{J} z_j \\ \text{subject to} \quad & z_j \geq f(\mathbf{x}, \mathbf{y}_j) - \alpha, \quad j \in [J], \\ & z_j \geq 0, \qquad\qquad\quad j \in [J], \\ & \mathbf{x} \in \mathbf{X}. \end{aligned} \tag{8.3}$$

Several studies have shown that this formulation provides a very powerful, fast, and numerically stable technique which can solve problems with a large number of variables and past scenarios [232, 228].

8.3.1 A Nominal Formulation for Algorithmic Insurance

We adjust the formulation introduced by Uryasev (2000) to the algorithmic insurance setting [339]. Equation (8.3) provides a linear optimization formulation that is based on data-driven scenarios $\mathbf{y}_j$. We will present in the following sections a quantitative process to simulate J scenarios $\mathbf{y}_j$ for a given number of observations N. The loss function $f(\cdot)$ is not linear since it is defined as $f(x, y) = \max\{0, (y - x)\}$. We can also restrict the price vector $\mathbf{x}$ within fixed

lower (l_p) and upper bounds (H_p) that reflect the pricing constraints of the insured party. We let $[P]$ represent the set of premium categories that are included in the contract. We propose the following baseline formulation:

$$\begin{aligned}
\text{minimize} \quad & \alpha + \nu \sum_{j=1}^{J} z_j \\
\text{subject to} \quad & z_j \geq \sum_{p=1}^{P} \max\{0, (y_{pj} - x_p)\} - \alpha, \quad j \in [J], \\
& z_j \geq 0, \qquad\qquad\qquad\qquad\quad j \in [J], \\
& l_p \leq x_p \leq H_p \qquad\qquad\qquad\quad p \in [P].
\end{aligned} \tag{8.4}$$

Finally, we can solve Equation (8.4) using a cutting planes algorithm or by linearizing the constraints, which results in:

$$\begin{aligned}
\text{minimize} \quad & \alpha + \nu \sum_{j=1}^{J} z_j \\
\text{subject to} \quad & z_j \geq \sum_{p=1}^{P} w_{pj} - \alpha, \quad j \in [J], \\
& z_j \geq 0, \qquad\qquad\quad j \in [J], \\
& w_{pj} \geq 0, \qquad\qquad\quad j \in [J], p \in [P], \\
& w_{pj} \geq y_{pj} - x_p, \qquad j \in [J], p \in [P], \\
& l_p \leq x_p \leq H_p \qquad\quad p \in [P].
\end{aligned} \tag{8.5}$$

8.3.2 A Robust Formulation for Algorithmic Insurance

Solutions to optimization problems can exhibit high sensitivity to perturbations in the problem parameters [20]. In the setting of algorithmic insurance, uncertainty lies at the center of the problem, gaining even higher significance. In the absence of real-world past scenarios $\mathbf{y}_j$, we propose a data-driven way to generate them in Sections 8.4-8.6, combining the model properties with historical claims of past cases based on human decisions. Undoubtedly, our modeling approach contains noise in the proposed scenarios $\mathbf{y}_j$ which can be attributed either to the probabilistic assumptions or to the model performance. Robust optimization

offers a solution to this problem proposing uncertainty models that are not stochastic, but rather deterministic and set-based. Leveraging these techniques, we introduce two robust formulations.

Box Uncertainty with Γ-Robustness. We will apply the notion of Γ-Robustness proposed by [44]. We first assume that the scenarios $\mathbf{y}_j$ lie in the uncertainty set:

$$\mathcal{U}_1 = \{y | \mu_{pj} - \delta_{pj}\gamma_{pj} \leq y_{pj} \leq \mu_{pj} + \delta_{pj}\gamma_{pj}, \|\gamma\|_\infty \leq \Gamma\}. \tag{8.6}$$

Therefore, in this setting, the formulation of the robust counterpart is the following:

$$\begin{aligned}
\text{minimize} \quad & \alpha + \nu \sum_{j=1}^{J} z_j \\
\text{subject to} \quad & z_j \geq \sum_{p=1}^{P} w_{pj} - \alpha, && j \in [J], \\
& z_j \geq 0, && j \in [J], \\
& w_{pj} \geq 0, && j \in [J], p \in [P], \\
& w_{pj} + x_p \geq \mu_{pj} + \delta_{pj}\gamma_{pj}, && j \in [J], p \in [P], \\
& l_p \leq x_p \leq H_p && p \in [P].
\end{aligned} \tag{8.7}$$

Polyhedral Uncertainty with Γ-Robustness. Here, we assume that the scenarios $\mathbf{y}_j$ lie in the uncertainty set:

$$\mathcal{U}_2 = \{y | \mu_{pj} - \delta_{pj}\gamma_{pj} \leq y_{pj} \leq \mu_{pj} + \delta_{pj}\gamma_{pj}, \|\gamma\|_1 \leq \Gamma\}. \tag{8.8}$$

Based on this definition, the corresponding robust formulation follows:

$$
\begin{aligned}
\text{minimize} \quad & \alpha + \nu \sum_{j=1}^{J} z_j \\
\text{subject to} \quad & z_j \geq \sum_{p=1}^{P} w_{pj} - \alpha, && j \in [J], \\
& z_j \geq 0, && j \in [J], \\
& w_{pj} \geq 0, && j \in [J], p \in [P], \\
& w_{pj} + x_p \geq \mu_{pj} q_{pj} - \mu_{pj} s_{pj} + \Gamma r, && j \in [J], p \in [P], \\
& q_{pj} - s_{pj} = 1, && j \in [J], p \in [P], \\
& -\delta_{pj} q_{pj} - \delta_{pj} s_{pj} + r \geq 0, && j \in [J], p \in [P], \\
& q_{p,j} \geq 0, && j \in [J], p \in [P], \\
& s_{p,j} \geq 0, && j \in [J], p \in [P], \\
& r \geq 0 \\
& l_p \leq x_p \leq H_p && p \in [P].
\end{aligned}
\tag{8.9}
$$

8.4 The Cost of Predictive Performance

The optimization formulations in Section 8.3 are based on historical scenarios of loss $\mathbf{y}$. In this section, we derive a baseline approximation of a contract's expected loss using a data-driven approach that leverages available information regarding both the frequency and severity of future claims. We argue that the claim frequency of an erroneous algorithmic decision is a function of the model's predictive performance (e.g., AUC). Higher sensitivity reduces the probability of a false negative algorithmic mistake while models with higher specificity are less likely to perform a false negative error. The expected claims cost (severity), though, will depend on the nature of the decision and the type of error.

Going back to our case study, suppose that a pathologist receives FNA samples from which, using their knowledge and experience, determines whether a patient has a malignant breast tumor or not. Depending on the doctor's response the patient will or will not follow cancer treatment. In the case the physician proposes an erroneous diagnosis, there is an associated cost with this decision:

- If the patient is diagnosed with cancer but does not actually have it, there is the additional cost of unnecessary treatment that may even result to a needless mastectomy. We will assume that this cost is captured by a random variable K with mean μ and variance σ_μ.

- If the patient is not diagnosed with cancer but actually has the disease, the severity of the outcome for the patient is likely increased, since it is known that early detection is critical in cancer patients. This increase of severity is associated with a higher litigation cost, which is captured by a random variable L with mean M and variance σ_M.

Suppose now that instead of a doctor, a ML model is taking up the task of deciding, based on the FNA samples, whether the patient has cancer or not. This is not only a hypothetical example as at the Massachusetts General Hospital radiology department a ML model is partially responsible for the screening process of patients with mammograms [361]. Typically, the output of such binary classification algorithms is a prediction score. The model assigns to each input observation an individual risk score that indicates how likely it is for each sample to be associated with the outcome of interest (e.g., cancer diagnosis). To map each observation to a crisp class label, a classification threshold τ must be defined. For example, if the pathology department has specified a classification threshold $\tau = 0.3$, then all patients whose FNA outcome has probability of being positive > 0.3 are diagnosed with breast cancer. In the same example, all samples for which the model predicts a score of ≤ 0.3 are classified as cancer-free.

Let $\mathbf{x}_i$ be the feature vector of patient i and $g(\cdot)$ is the probability that patient i has breast cancer. Then the class of i is defined as follows:

$$\text{class}(i) = \begin{cases} 1, & \text{if } g(\mathbf{x_i}) > \tau, \\ 0, & \text{otherwise} \end{cases}$$

Depending on the value of this threshold τ, the ability of the algorithm to identify false positives and false negative cases varies. Higher values of τ improve the specificity of the

model, avoiding unnecessary alerts to healthy patients. On the other hand, lower values of τ improve the sensitivity of the model, resulting in the timely warning of a higher number of cancer cases. Both measures are threshold dependent. Thus, we can formally define them as follows:

- Let $\kappa_\tau \in [0, 1]$ be the specificity of the ML model for a classification threshold τ. The probability that a sick patient will be erroneously classified is then $1 - \kappa_\tau$.
- Let $\lambda_\tau \in [0, 1]$ be the sensitivity of the ML model for a classification threshold τ. Thus, the probability that a healthy patient will be wrongly classified is $1 - \lambda_\tau$.

Taking all the above into consideration, the claim cost of a new patient that is diagnosed by the ML model is captured by the random variable S:

$$S = (1 - \kappa_\tau)K + (1 - \lambda_\tau)L. \tag{8.10}$$

Therefore, the expected value of the individual claim cost is equal to:

$$\mathbb{E}(S) = (1 - \kappa_\tau)\mu + (1 - \lambda_\tau)M. \tag{8.11}$$

The corresponding variance is $\sigma = (1-\kappa_\tau)^2\sigma_\mu^2+(1-\lambda_\tau)^2\sigma_M^2+2(1-\kappa_\tau)(1-\lambda_\tau)\mathrm{Cov}(K, L)$, and the respective correlation coefficient is $\rho(K, L)$. If we assume that N patients are expected to arrive at the hospital during the contract period, then the total expected loss of the insurance is:

$$\mathbb{E}(C) = N\,\mathbb{E}(S) = N((1 - \kappa_\tau)\mu + (1 - \lambda_\tau)M). \tag{8.12}$$

8.4.1 Case Study: Experimental Setup

We perform a series of computational experiments in the case study of interest to evaluate the effect of the model parameters on the risk appreciation framework. We fix our number of scenarios to use $J = 1000$ and assume that $N = 100$ patients are served within the contract period. We hypothesized that our litigation cost variables K, L follow independent

normal distributions. In this setting, we do not distinguish between different price segments, assuming that $P = 1$. We vary the values of $\mu, \sigma_\mu, M, \sigma_M$ between the lower and upper ranges obtained from historic medical malpractice cases of breast cancer such as the ones presented in Section 8.2. We study the impact of the classification threshold τ as well as the confidence level β. We constrain the contract premium to \$10,000 and \$50,000. We quantify the effect of the Γ parameter in the robust optimization formulation. We use bootstrapping across 10 random seeds. We report the average performance across all iterations in our results. The parameter ranges are detailed in Table 8.1.

The data was randomly split into training (75%) and testing sets (25%). Missing values in each partition were imputed using the `MedImpute` algorithm [41]. We use the RF algorithm to train the binary classification models [55]. We apply 10-fold cross-validation to set the number of estimators and the maximum depth of the individual tree-based models. The average AUC of the final model on the testing set is 99.36%. The statistical analysis was conducted using Python 3.7 and Julia 1.3 [266, 45]. The codebase for all of the experiments is available as a Github repository [258].

Parameter	**Range**
Γ	3
β	0.9, 0.95, 0.99
l_p	\$10,000
H_p	\$10,000, \$50,000
μ	\$100,000 \$500,000
σ_{mu}	\$25,000, \$150,000
M	\$500,000, \$1,000,000
σ_M	\$150,000, \$400,000
τ	0.01 - 0.75
J	1,000
N	100

Table 8.1: Parameter ranges for the computational experiments.

8.4.2 Case Study: The Implementation Framework

The proposed formulation allows us to estimate for a given confidence level (β) and a vector of historic claims ($\mathbf{y}$): (i) the prices ($\mathbf{x}$) for each product class (i.e., age groups, vehicle types); (ii) the VaR (α); (iii) the CVaR which corresponds to the objective function ($\min \alpha + \nu \sum_{j=1}^{J} z_j$).

The input necessary to apply it involves:

- A binary classification model (e.g., image recognition classifier for mammograms) with a representative testing set $g(\cdot)$;
- Random variables K, L that represent the litigation cost for false negative and false positive cases with means μ, M, variances σ_μ^2, σ_M^2, and covariance $\text{Cov}(K, L)$ respectively;
- The number of patients that the algorithm will serve during the contract, N;
- The number of past scenarios for the optimization formulation, J;
- Upper and lower bounds for the price, l_p, H_p.

If past data from the implemented algorithm with prior cases of litigation claims were available, we would use the historical observations. In the absence of such information, we use random variable realizations to get the cost approximation of false positive and negative cases. Thus, the total cost for scenario j of a fixed contract period for price segment p can be computed as follows:

$$y_{pj} = \sum_{i=1}^{N} (1 - \kappa_{\tau p}) K_{pji} + (1 - \lambda_{\tau p}) L_{pji} \tag{8.13}$$

Algorithm 4 summarizes the proposed process that combines all the components of our approach.

8.4.3 Case Study: The Effect of the Classification Threshold

The first question that we aim to answer is what is the effect of the classification threshold τ on the estimated CVaR of the contract. In Figure 8.1 we depict two scenarios of a low

Algorithm 4 Framework Implementation Procedure

Output: prices ($\mathbf{x}$), VaR (α), CVaR

Input: $\beta, g(\cdot), l_p, H_p, K, L, J, N, X_{train}, X_{test}$

```
Train model g(·) using X_train
Get predicted probabilities for X_test
while j ≤ J do
    while i ≤ N do
        Calculate scenario y_j sampling from K, L for patient i.
        i=i+1
    end while
    j=j+1
end while
Solve the optimization formulation
```

and a high litigation claims distribution. The blue curve corresponds to $\mu = \$100,000, \sigma_\mu = \$25,000, M = \$500,000, \sigma_M = \$150,000$. The yellow curve corresponds to $\mu = \$500,000, \sigma_\mu = \$150,000, M = \$1,000,000, \sigma_M = \$450,000$. On the horizontal axis we project the threshold values and on the vertical axis the CVaR. Our findings reveal that the most significant determinant of CVaR is the selected classification threshold τ. The expected costs for the false positive and false negative case re-scale the CVaR function. Moreover, the effect of μ is more prominent for lower values of τ that negatively impact the specificity of the model. On the contrary, M gains more significance for higher values of τ. The average AUC of the binary classification models is 99.36% and thus for any given threshold, the probability of a false negative and a false positive is very low. Notice that the CVaR of both curves in Figure 8.1 is minimized for $\tau = 0.3$. This is due to the fact that when $\tau \leq 0.3$, the model sensitivity on the testing set is equal to one. Therefore, for $\tau = 0.3$ the model specificity is maximized for the best possible value of sensitivity. This analysis demonstrates that the selected classification threshold τ can dramatically affect the CVaR value even for fixed distributions of litigation costs.

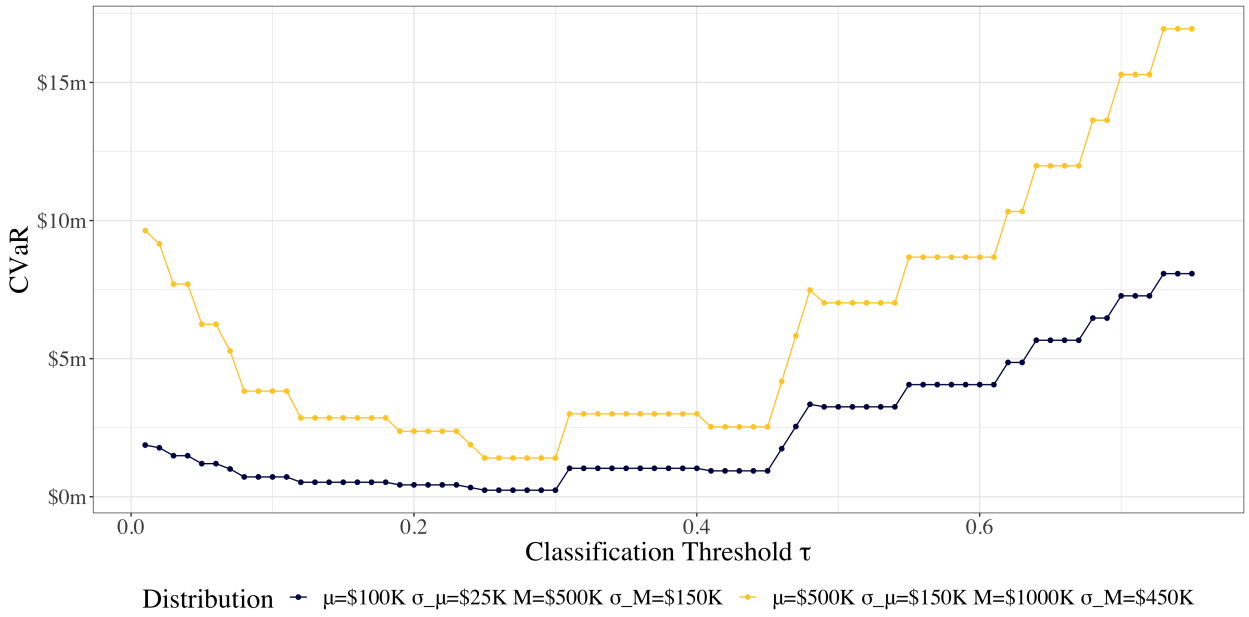
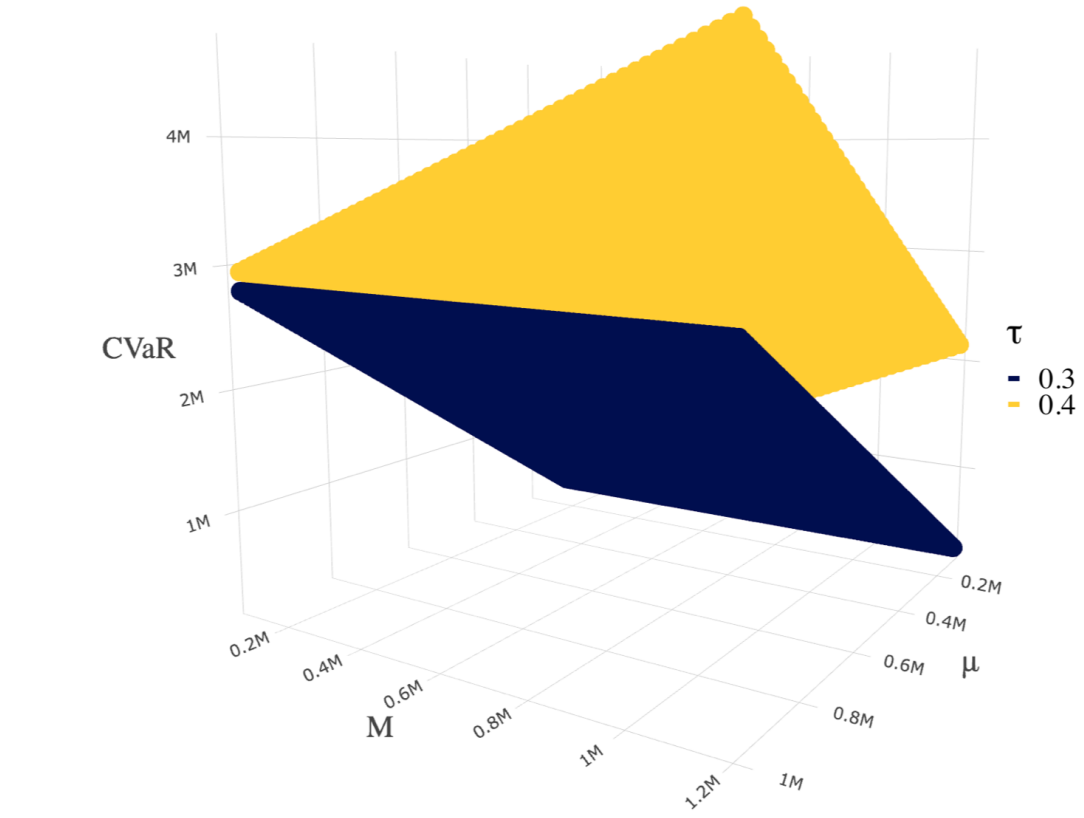


Figure 8.1: CVaR as a function of the τ parameter for two different combinations of the K, L distributions.

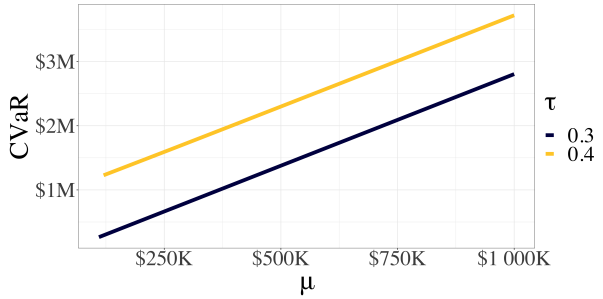
8.4.4 Case Study: The Effect of the Claims Cost Expected Value

The next question that we address relates to what is the impact of the means of the random variables K, L on the contract's risk exposure. Figure 8.2 provides three-dimensional and two-dimensional illustrations of CVaR as we vary the expected values M and μ for $\tau \in \{0.3, 0.4\}$. In these experiments, σ_μ and σ_M correspond to 20% of μ and M respectively. Figure 8.2a shows CVaR as a function of both M and μ. When $\tau = 0.3$, the model does not include any litigation cost for false negative claims and CVaR is a linear function of the model specificity. This is evident in Figures 8.2b-8.2c too. For a fixed value of μ, any increase or decrease of M does not affect the contract's financial risk. On the other hand, as illustrated in Figures 8.2b-8.2c, when $\tau = 0.4$ both model sensitivity and specificity affect the exposed risk of the contract in a linear fashion. In this case, CVaR depends on the distribution of both K and L and thus it is a linear function of μ and M. As we decrease the value of τ below 0.3, the probability of a false positive claim is increasing and as a result CVaR is also increasing. For example, when $\tau = 0.25$ the model specificity is 97.24% while when $\tau = 0.15$ the specificity drops to 94.49%. Our results highlight that CVaR is a linear function of both

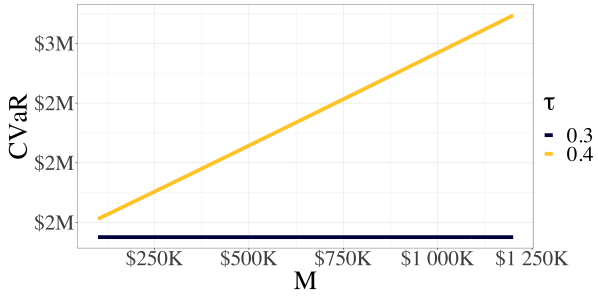
M and μ.



(a) 3-Dimensional Illustration.



(b) 2-Dimensional Illustration when $M = \$600K$.



(c) 2-Dimensional Illustration when $\mu = \$500K$.

Figure 8.2: CVaR as a function of μ for $\tau = 0.3$ and $\tau = 0.4$.

Case Study: The Effect of Robust Optimization, the Premium and the Confidence Level β

Subsequently, we investigate the role of the the premium, the type of the formulation, and the confidence level β in the determination of CVaR. We summarize our findings in Table 8.2. The premium value x_p is set in the vast majority of the experiments by the upper bound H_p. Only in cases where both M and μ are below $\$50,000$, this constraint is not binding. Naturally, higher premium values result to lower VaR and CVaR for the contract. The β parameter also has a linear effect on the final risk estimation as it is incorporated as a scalar in the objective function of the formulation. In addition, the results in Table 8.2 reveal the benefit of the robust optimization approach. As we expected, the box uncertainty sets are more conservative than the polyhedral uncertainty approach. The latter yields very similar results to the nominal problem while also accounting for uncertainty in the scenarios $\mathbf{y}$.

Premium	**Formulation**	$\boldsymbol{\beta} = 0.9$	$\boldsymbol{\beta} = 0.95$	$\boldsymbol{\beta} = 0.99$
\$ 10,000	box	\$ 300,536	\$ 302,959	\$ 306,714
\$ 10,000	nominal	\$ 276,117	\$ 278,071	\$ 281,631
\$ 10,000	polyhedral	\$ 279,611	\$ 282,084	\$ 285,251
\$ 50,000	box	\$ 260,536	\$ 262,959	\$ 266,714
\$ 50,000	nominal	\$ 236,117	\$ 238,071	\$ 241,631
\$ 50,000	polyhedral	\$ 239,611	\$ 242,084	\$ 245,251

Table 8.2: Average CVaR as we vary the premium price, the type of formulation, and the confidence level β. These results correspond to $\tau = 0.3, \mu = \$100,000, \sigma_\mu = \$25,000, \Gamma = 3$.

8.5 The Cost of Interpretability

In our effort to price the risk of algorithmic decision-making, interpretability may play a crucial role. Consequential decision-making up until recently has been strictly controlled by humans. In this setting, the outcome of any decision can be associated with reasoning that would justify the action. Thus, a human decision can be evaluated based on the logic followed and, subsequently, the decision agent can be held accountable for their judgement. Supervised learning algorithms do not necessarily provide a reason why a given observation

should receive a specific label. They can only state that certain inputs are correlated with that label. As a result, interpretability has remained an ill-defined term of ML [216].

In the context of algorithmic liability, one could argue that the interpretability of a model is a measure of how much human input could be involved in the risk estimation process. We will focus on the setting of complete automation where human input is possible only prior to the model implementation in practice. In this context, experts may be called to review and approve the algorithm prior to its integration to avoid erroneous decision rules in the learner. Consider the case of a fully interpretable, tree-based model for malignant tumor detection. The physician in charge can easily review the algorithm's recommendations based on their own knowledge and experience prior to its implementation. The level of algorithmic transparency directly affects the degree to which human judgement can be involved. Interpretable models allow for synergies between the ML algorithm and the experts' input. Therefore, we argue that the combination of artificial and human intelligence is likely to lead to more accurate estimations and may improve the risk exposure of the contract.

The goal of this section is to quantify this effect and provide measures of how interpretability can impact algorithmic risk evaluation. Suppose that c_h is the risk exposure for an insurance contract when a human expert is making all the decisions. Notice that c_h is known from historical claims. c_{ml} is the risk exposure for the same contract when a ML model is the sole decision maker. We assume that $c_h > c_{ml}$ to ensure that there are financial incentives from the use of the ML model. We let $\theta \in [0, 1]$ be the interpretability parameter that measures the degree of algorithmic transparency and assume that a model's risk exposure $c(\theta)$ is a function of the interpretability parameter.

When $\theta = 0$, the ML model is treated as a "black-box" and thus a human agent is unable to provide additional input that may improve the model's performance ($c(0) = c_{ml}$). To the contrary, when $\theta = 1$, the model is intuitive and explainable for the decision maker and as a result there are synergies between the ML model and the human agent, resulting in a lower cost $c(1) = \xi c_{ml}$, $\xi \in (0, 1)$.

The determination of the ξ parameter depends on the ML model, the application and

the problem under consideration. We assume that the relative benefit of interpretability directly depends on the relative ratio of $\frac{c_{ml}}{c_h} \in (0, 1)$. The latter ratio captures the relative improvement of a ML model over human judgement in economic terms. In cases where the edge of algorithmic decision making is small ($c_{ml} \sim c_h$), the value of interpretability is high, since an expert's opinion can yield equivalent results to an algorithm. In this setting, the synergies between the decision maker and the machine are stronger, correspondingly decreasing the expected risk exposure. On the other hand, when c_h is significantly higher than c_{ml}, interpretability gains less importance as human input might not be as informative. Based on these assumptions, a potential value for ξ is $(1 - \frac{c_{ml}}{c_h})$, capturing the synergies between the two types of decision makers. It follows that when $\theta = 1$, the cost is equal to $c = c_{ml}(1 - \frac{c_{ml}}{c_h})$. If we model the contractual risk exposure c for all values of $\theta \in (0, 1)$ as the linear interpolation of these two scenarios (see Figure 8.3), then $c = -\frac{c_{ml}^2}{c_h}\theta + c_{ml}$.

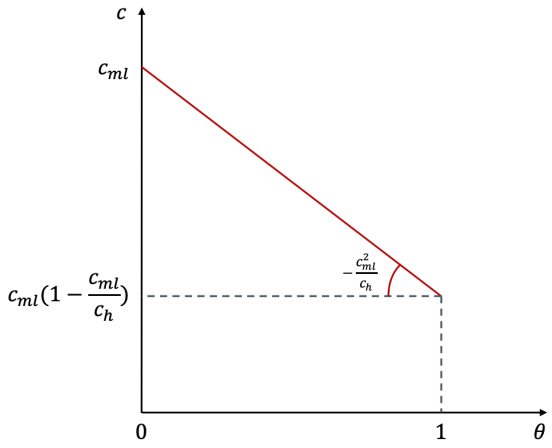


Figure 8.3: The linear interpolation of the c function for $\theta \in [0, 1]$ when $\xi = (1 - \frac{c_{ml}}{c_h})$.

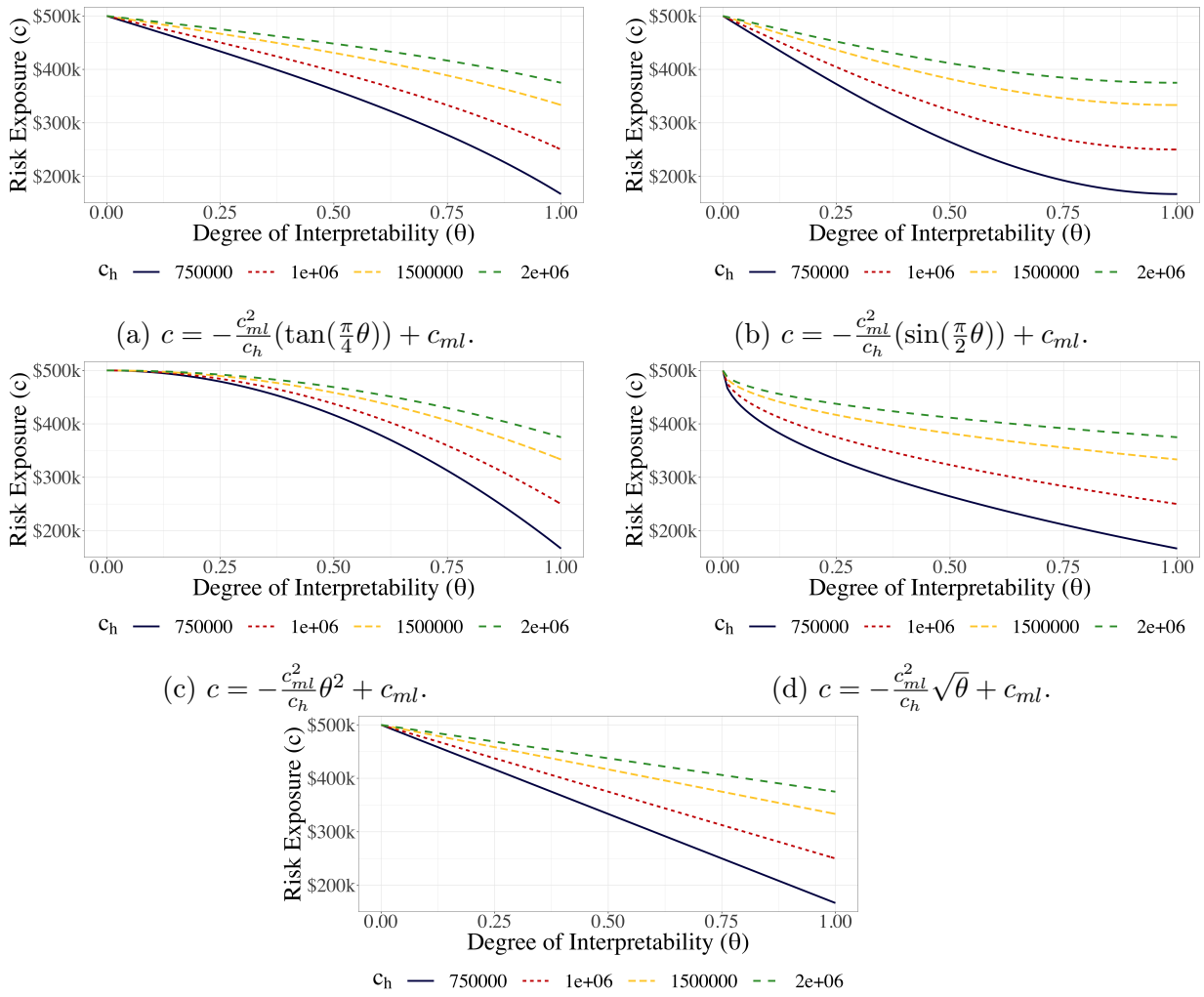
However, interpretability does not necessarily have a linear effect on risk exposure and model performance. One can hypothesize that the effect of interpretability on the cost is non-linear. If c is concave function, the positive effect of interpretability on the risk exposure is more prominent for higher values of θ. To the contrary, if we assume that c is convex, even with small degrees of interpretability, we can observe significant reductions in the expected risk. Though it remains challenging to fully characterize the interpretability effect, our approach provides flexible options to decision makers to measure its impact as a function of

the θ parameter.

Identifying a single value for this parameter and determining a specific definition or degree of interpretability has been a major challenge in the ML field [65]. Most definitions involve human input in the evaluation process which impedes systematic quantitative analysis [215]. Bertsimas et al. (2019) have recently introduced a quantitative approach to specify the price of interpretability as the tradeoff with predictive accuracy for a given model [33]. Alternative approaches that could be directly incorporated in our pricing framework include the work of Schmidt et al. (2019) and Ribeiro et al. (2016) [305, 287]. We expect that an increasing number of interpretability definitions will be available in the future considering the importance of *understanding* a model's proposed associations between the input variables and the output labels.

8.5.1 Case Study: The Effect of Interpretability

In Figure 8.4, we provide concrete examples of functions that model the effect of interpretability in the risk estimation process for the case study of medical liability. On the horizontal axis we project the θ parameter and on the vertical axis the CVaR. Each graph corresponds to a different function c, including concave, convex, and linear examples. In this setting, we assume that $\xi = (1 - \frac{c_{ml}}{c_h})$, $c_{ml} = \$500K$, and consider four distinct scenarios of c_h. Notice when c is convex, such as in Figures 8.4b and 8.4d, even with low degrees interpretability, we can derive effective synergies between the model and the human agent that can significantly reduce the risk exposure. Respectively, when modeling the effect of interpretability with a concave function, like the ones presented in Figures 8.4a,8.4c, the majority of the risk reductions will only be observed for higher values of θ. This intuition can guide the decision for the determination of the interpretability function in future applications of the framework.



(a) $c = -\frac{c_{ml}^2}{c_h}(\tan(\frac{\pi}{4}\theta)) + c_{ml}$.

(b) $c = -\frac{c_{ml}^2}{c_h}(\sin(\frac{\pi}{2}\theta)) + c_{ml}$.

(c) $c = -\frac{c_{ml}^2}{c_h}\theta^2 + c_{ml}$.

(d) $c = -\frac{c_{ml}^2}{c_h}\sqrt{\theta} + c_{ml}$.

(e) $c = -\frac{c_{ml}^2}{c_h}\theta + c_{ml}$.

Figure 8.4: Risk Exposure c as a function of the interpretability parameter θ and risk exposure c_h for a fixed value of $c_{ml} = \$500\text{K}$.

8.6 The Cost of Model Generalizability

Another prominent aspect of algorithmic performance is generalizability, which refers to a model's ability to properly adapt to new, previously unseen data, drawn from the same or a different distribution as the one used to create the model. In the context of algorithmic insurance, generalizability becomes pertinent when the model is applied to a new population whose data have not been tested by the algorithm in the past. Going back to the case

study of interest, let the binary classification model for malignant tumor detection be exclusively trained on samples drawn from a Caucasian population at an academic hospital in Massachusetts. Suppose that a community hospital in Louisiana is now interested to integrate the system in its EHR database which started collecting medical data only one year ago. What is the appropriate pricing strategy for this center? If the hospital had at its disposal data from the patient population it serves throughout a span of multiple years, we would directly be able to externally validate its performance. However, in the absence of this valuable resource, an alternative approach is needed to adjust the pricing strategy of the contract.

We propose testing the predictive performance of the algorithm on synthetic observations that resemble the original training set, controlling the degree of similarity to get estimations of risk variability. Multiple approaches have been proposed in the literature to generate synthetic data. A quite common approach is to induce noise with Gaussian random variables in the existing dataset. Nevertheless, this simple technique directly affects the existing correlations and associates between the covariates distorting the true geometry of the feature space. Another well established technique for data augmentation is referred to as the Synthetic Minority Oversampling Technique (SMOTE) which is very frequently applied in classification datasets that have a severe class imbalance [71]. However, this approach does not let the user directly control the degree of similarity between the newly created instances and the original data distribution.

The proposed framework requires the use of parameters that effectively control the degree of similarity between the real and the synthetic data. Our goal is to introduce a mechanism that would allow decision makers to quantitatively compare the additional cost of the insurance contract with respect to the ability of the model to generalize to datasets with different levels of variability from the original training and testing sets. Though it is not possible to provide theoretical guarantees about the predictive performance of a given classifier in a new unknown distribution, we can conduct extensive simulations that test the discrimination capability of the learner in adverse scenarios.

8.6.1 Generative Adversarial Networks to Generate Synthetic Data

To perform this analysis we resort to Generative Adversarial Networks (GANs). This neural network architecture was first introduced in 2014 with the goal of synthesizing artificial images that are indistinguishable from authentic figures [147]. Since then, GANs have become the predominant method of data augmentation for images and text [348]. They are used to increase the amount of data by adding slightly modified copies of already existing samples or newly created synthetic data from existing observations [277, 146, 370].

GANs involve a unique architecture in which a pair of networks are trained simultaneously and in competition with each other. Training a robust GAN architecture is a non-trivial task due to problems like vanishing gradients and mode-collapse which may result in poor discrimination performance and synthetic samples with limited diversity [277, 302]. The Wasserstein GANs (WGAN) architecture was introduced to remedy these issues using the Wasserstein distance as the loss function [11]. The Conditional GAN (CGAN) is a specific class of GANs that involves the conditional generation of images by a generator model, resulting in new data observations with associated class labels [237]. This modification to the GAN architecture permits learning the distributions specific to each class label, producing samples for both labels with higher quality.

Recently, researchers showed that WCGANs can lead to very promising results in synthesizing tabular data, comprising only densely connected layers [231, 322]. These efforts highlighted the potential impact that GANs may have in structured data sources, providing researchers with higher flexibility and control in the data generation process. This technique also offers a promising alternative to solve our problem of interest; creating synthetic data while explicitly controlling the degree of similarity. By varying the number of epochs, we will control the differences in the distributions between the synthetic and the real-world data. This parameter constitutes the number of complete passes through the training process. A higher number of epochs gives the opportunity to the algorithm to converge, minimizing the loss function. Thus, we can directly compare the discriminator loss function to the number of epochs, adjusting the degree of dissimilarity between the synthetic samples and the real

training set.

Due to their complicated structure and despite large strides in terms of theoretical progress, evaluating and comparing GANs remains a challenging task. While several metrics have been proposed, there is no consensus in the scientific community as to which measure provides a more holistic and objective model evaluation [7, 52]. We propose the use of the GAN quality index (GQI) for our analysis [363]. To compute this metric, we will compare the performance of the generator G and the model we are seeking to price C_{real}. First, we generate the synthetic samples with the associated class labels using the CGAN architecture. A second classifier, called the GAN-induced classifier C_{GAN} is trained on the generated data. The GQI is defined as the ratio of the accuracies (or AUCs) of the two classifiers when applied to the real test data:

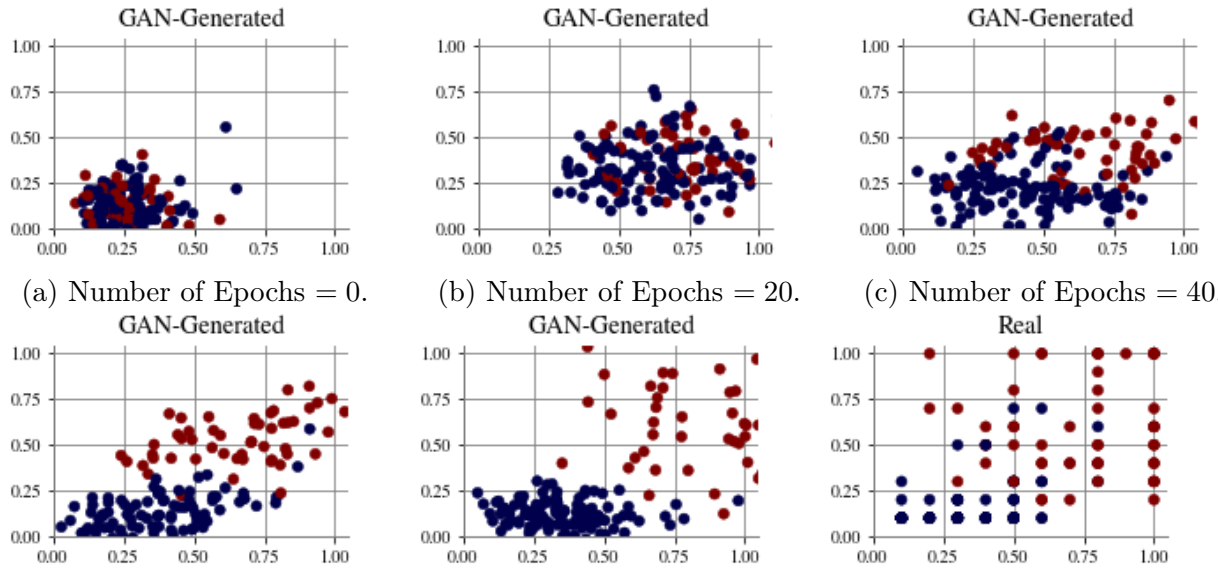
$$GQI = \frac{ACC(C_{\text{GAN}})}{ACC(C_{\text{REAL}})}$$

Higher GQI means that the GAN distribution better matches the real data distribution.

8.6.2 Case Study: The Effect of Generalizability

We apply the proposed approach to the case study of medical liability using a GAN architecture. Our goal is to investigate the effect of generalizability on the contract's risk exposure using synthetic samples while controlling the degree of similarity to the original dataset. The GAN network was built leveraging the GAN-Sandbox package and was implemented in Python using the Keras library with a Tensorflow backend [100, 74, 2].

In Figure 8.5, we present the changes in the features' distribution as a function of the number of epochs parameter in a WCGAN architecture. Following the architecture presented by Vega et al. (2019), the GAN model comprises a generator network with one input layer and three dense layers and an adversarial network with one input layer and four dense layers [343]. We measure the similarity of the derived distributions between the synthetic (GAN-Generated) and the real data for a different number of epochs in the training process. An epoch is defined as one cycle through the training process of the network, corresponding to the number of training iterations between the generator and the adversarial network. Our



(a) Number of Epochs = 0. (b) Number of Epochs = 20. (c) Number of Epochs = 40.

(d) Number of Epochs = 60. (e) Number of Epochs = 150. (f) Real Data.

Figure 8.5: The derived distributions from the GAN model of two of the most predictive features. Cell Shape Uniformity is depicted on the vertical axis and Clump Thickness is illustrated on the horizontal axis. All values have been normalized to [0,1]. Each graph corresponds to the output of the GAN model for a given number of epochs. Figure 8.5f corresponds to the real distribution of the features.

hypothesis in Section 8.6 is validated, as we observe that a higher number of epochs results in features distributions that better resemble the real data distribution (Figure 8.5). This effect is directly present in quantitative metrics, such as the GQI index, which is also positively correlated with the number of epochs in the model.

In Figure 8.6, we project CVaR as a function of the number of epochs parameter. The underlying binary classification model is the same as the one used in Figure 8.1 where $\mu = \$100,000, \sigma_\mu = \$25,000, M = \$500,000, \sigma_M = \$150,000$, derived on the training sample of the original Breast Cancer Wisconsin Diagnostic dataset. The CVaR is then measured with respect to the discrimination performance of the classifier on the GAN-Generated synthetic data. The output of CGAN architectures includes both independent features and associated labels that are subsequently used to compute the sensitivity and specificity of the model for different values of τ. In Figure 8.6, we only project the best CVaR value across all potential thresholds τ. This graph reveals a linear relationship between the number of epochs and the

CVaR. This finding highlights to decision makers the cost effect of applying a pre-trained learner in datasets with varied degrees of distribution similarity to the original training population.

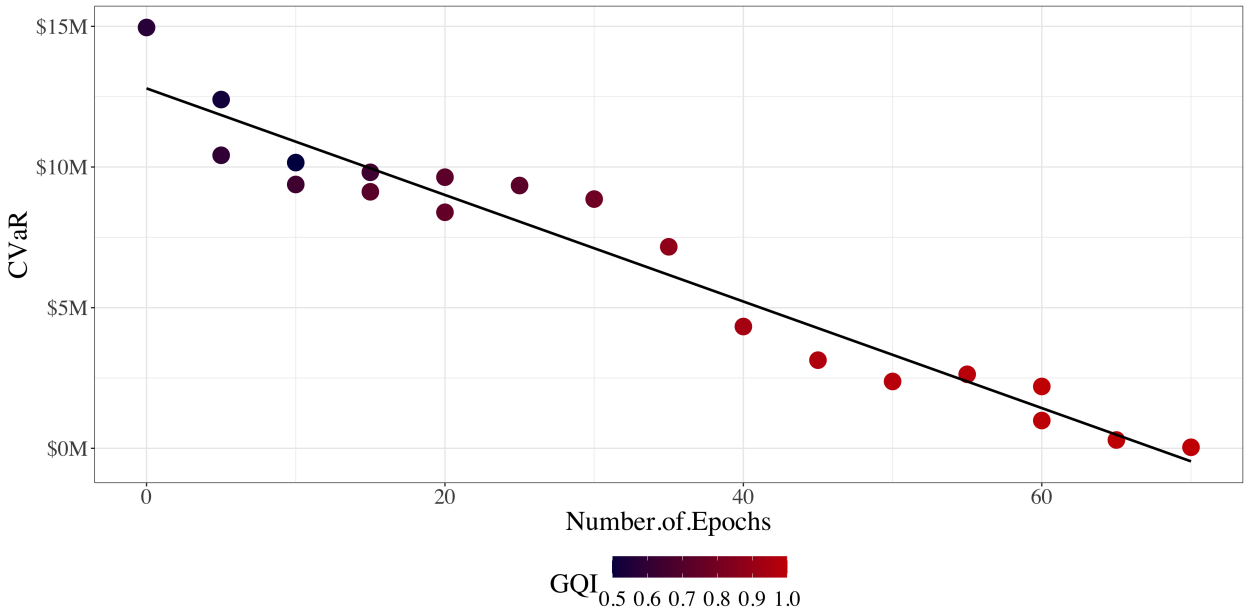


Figure 8.6: CVaR as a function of the number of epochs parameter. The color indicates the GQI metric for the underlying binary classification model.

8.7 Discussion

In this chapter, we construct a framework for pricing algorithmic risk and provide a comprehensive case study for its implementation. Our work constitutes the first attempt to quantify the litigation risk resulting from erroneous algorithmic decision-making in the context of binary classification models. The framework is agnostic to the type of learner and application area. It can be easily extended in other fields, such as predictive maintenance and autonomous vehicles among others. The proposed models are data-driven and parametric, since such contracts have not been implemented by the industry yet and extensive experimentation is needed prior to launching them.

Our work reveals that, given fixed distributions for litigation costs, the choice of the

classification threshold τ plays the most significant role in the determination of the contract's financial risk. This finding provides us with a new way of assessing the balance between sensitivity and specificity. The direct association of risk exposure to the threshold τ allows us to make better decisions for the financial implementations of the classifier in practice. Our findings validate the assumption that when a false positive and a false negative have disproportionate implications for the organization the choice of the τ threshold becomes even more critical.

The proposed framework incorporates the critical aspect of model interpretability. We argue that in cases where the risk of human decision making is comparable to algorithm-driven decision rules, interpretability gains greater value. We also show that the convexity of the interpretability function indicates whether a smaller or a higher degree of model transparency is needed to achieve significant synergies between human agents and the ML algorithm.

We consider the use of GANs to evaluate the effect of model generalizability in the risk evaluation process. We illustrate a linear relationship between CVaR and the number of training epochs of the GAN network. The latter is also positively correlated with similarity metrics between the generated and the true data distribution. Thus, we allow decision makers to adjust the contract pricing to settings where the classifier is applied to a new population that was not part of the original training and validation set.

Future work would need to consider the regulatory framework of specific applications and account for other types of supervised learning methods such as regression algorithms. As interpretability and generalizability remain two areas of growing scientific interest for the ML community, we expect that new methods and tools will be developed that will allow the quantification of their impact.

8.7.1 Limitations

Central to the limitations of this study is the absence of historic claims records from real-world litigation cases of malpractice. We assume that insurance companies have in their possession this kind of information from which the scenarios $\mathbf{y}$ could be directly constructed. Our

analysis takes a conservative view over the litigation process assuming that every erroneously classified sample will resort to a litigation claim. In reality, only a portion of misdiagnosed patients file a malpractice lawsuit and only a subset of those are successful. In this chapter, our computational experiments have been based on normally distributed random variables but other distributions could also be explored in future investigations. We would also like to note that medical malpractice is a particularly challenging field and the implementation of ML models such as the one in Section 8.2 faces a lot of regulation constraints. As a result, it is very likely that such models will continue to be validated by medical experts in the coming years. Finally, our work does not account for dynamic decision-making processes, such as the triple FNA strategy, that usually results in improved clinical outcomes for the patients and significantly reduces the amount of claims [9].

8.8 Conclusions

Our work aims to set the foundations in the novel area of algorithmic insurance. We propose quantitative tools that allow decision makers, modelers, and insurance companies to estimate the litigation risk of binary classification models. This approach takes into consideration the predictive performance of the classifier accounting also for uncertainty in the data. We incorporate measures of interpretability and generalizability to provide a holistic appreciation of the model. We believe that this framework can serve as the basis of a new research area that will expedite the adoption and implementation of ML models in practice.

Chapter 9

Conclusions

This thesis provides a roadmap to personalized medicine and insurance using ML and optimization techniques.

Part I proposes new generalizable methodologies to tackle three of the most challenging problems encountered in healthcare and insurance datasets. Our approach involves formulating well-established ML tasks, such as clustering or missing data imputation, as MIO problems. We introduce new algorithms that scale to large data instances with superior performance compared to existing greedy methodologies. This effort emphasizes user transparency, which remains a critical factor for the adoption and success in industries with high-stakes decision-making. We focus on learners such as the k-NN or the Optimal Trees framework and demonstrate on synthetic and real-world data that interpretability does not need to be pursued at the expense of accuracy. To measure these effects, we design extensive computational experiments that address multiple aspects of algorithmic performance, providing a holistic evaluation of the proposed techniques.

These methodologies, although necessary, are insufficient to fully propel the transition to personalized medicine. The goal of Part II is to showcase how to leverage such ML algorithms to derive individualized prescriptions and predictions at the point-of-care. First, we propose a data-driven approach that combines multiple supervised learning models to provide treatment recommendations. Our prescriptive framework extends the classical *Regress and Compare*

approach by aggregating an ensemble of ML models. The algorithm recommends the therapy with the best expected outcome through a voting mechanism that considers the predictions from each of the regression models. Leveraging the different geometries of individual learners, the voting scheme avoids biases and pitfalls that are specific to a single method, providing a more holistic perspective to the decision maker. We showcase the potential benefit of this framework in the context of CAD, using a new evaluation framework that takes into consideration the effectiveness and robustness of the prescriptions. Part II also focuses on the predictive setting, highlighting how the use of analytics can lead to the creation of clinical decision support tools that take as input not only tabular but also unstructured sources of data. We integrate into the model evaluation the external validation process to provide evidence that the proposed binary classification models are generalizable and could be deployed in healthcare organizations that did not participate in the original study. We hope that these chapters will encourage further research in the fields of operations research and ML to continue developing and deploying personalized models in other medical domains.

In Chapters 5 and 6, we accompany the analytical models with prototypes, making available to physicians online interactive tools that communicate the algorithm recommendations. In the prescriptive setting, our tools illustrate a dashboard where the expected outcomes under alternative treatments are plotted per patient, providing justification on why a particular therapy is recommended. The N-SRS application presents the results in the form of a dynamic questionnaire minimizing the necessary input from the health practitioner. Thus, the decision maker is provided with explicit criteria and rationale behind the algorithmic output. In the future, we envision that these models will be fully integrated into the clinical workflow of medical professionals as part of the EHR system.

The last chapter of the thesis refers to algorithmic insurance. Our extensive collaborations with medical institutions at the forefront of clinical research showed that there is still a lot of reservation and resistance in employing ML-based tools in practice. Our work provides an evaluation framework for a new class of insurance products that will protect decision makers from algorithmic mistakes, boosting their implementation and integration into the medical

practice. Such contracts will be able to provide financial incentives to healthcare organizations to augment human decision making with analytical approaches that can ultimately improve patient outcomes and the quality of the provided care. This framework is certainly applicable beyond the context of healthcare as we showcase in Chapter 8. We believe that this work sets the foundations of a new broad field where a lot of research can be conducted to set its basis and support its implementation in the future.

To conclude, the ultimate goal of this thesis is to show how ML can positively influence clinical practice. By proposing new generalizable methods, prescriptive and predictive models, and insurance products, we hope we will be able to advance medicine to the next level, further ameliorating and personalizing the quality of care delivered.

Bibliography

[1] Odd Aalen. Nonparametric inference for a family of counting processes. *The Annals of Statistics*, pages 701–726, 1978.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[3] Cigdem Inan Aci and Mehmet Fatih Akay. A hybrid congestion control algorithm for broadcast-based architectures with multiple input queues. *The Journal of Supercomputing*, 71(5):1907–1931, May 2015.

[4] AHA. Heart disease and stroke statistics 2017. *AHA Centers for Health Metrics and Evaluation*, 2017.

[5] M Ahmed, M Jahangir, H Afzal, A Majeed, and I Siddiqi. Using crowd-source based features from social media and conventional features to predict the movies popularity. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 273–278, Dec 2015.

[6] Oguz Akbilgic, Hamparsum Bozdogan, and M Erdal Balaban. A novel hybrid RBF neural networks model as a forecaster. *Statistics and Computing*, 24, 05 2013.

[7] Hamed Alqahtani, Manolya Kavakli-Thorne, Gulshan Kumar, and Ferozepur SBSSTC. An analysis of evaluation metrics of gans. In *International Conference on Information Technology and Applications (ICITA)*, 2019.

[8] Diabetes Association American. Standards of medical care in diabetes—2010. *Diabetes Care*, 33(Supplement 1):S11–S61, 2010.

[9] Richard E Anderson and David B Troxel. Breast cancer litigation. In *Medical Malpractice*, pages 153–166. Springer, 2005.

[10] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, 91(434):444–455, 1996.

[11] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[12] Donna K. Arnett, Roger S. Blumenthal, Michelle A. Albert, Andrew B. Buroker, Zachary D. Goldberger, Ellen J. Hahn, Cheryl Dennison Himmelfarb, Amit Khera, Donald Lloyd-Jones, J. William McEvoy, Erin D. Michos, Michael D. Miedema, Daniel Muñoz, Sidney C. Smith, Salim S. Virani, Kim A. Williams, Joseph Yeboah, and Boback Ziaeian. 2019 acc/aha guideline on the primary prevention of cardiovascular disease. *Journal of the American College of Cardiology*, 74(10):e177–e232, 2019.

[13] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[14] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

[15] Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

[16] Rafael Ballester-Ripoll, Enrique G. Paredes, and Renato Pajarola. Sobol tensor trains for global sensitivity analysis. *Reliability Engineering and System Safety*, 183, 12 2017.

[17] Imon Banerjee, Matthew C Chen, Matthew P Lungren, and Daniel L Rubin. Radiology report annotation using intelligent word embeddings: Applied to multi-institutional chest ct cohort. *Journal of biomedical informatics*, 77:11–20, 2018.

[18] Jayanta Basak and Raghu Krishnapuram. Interpretable hierarchical clustering by constructing an unsupervised decision tree. *Knowledge and Data Engineering, IEEE Transactions on*, 17:121– 132, 02 2005.

[19] Amber L Beitelshees. Personalised antiplatelet treatment: a rapidly moving target. *The Lancet*, 379(9827):1680 – 1682, 2012.

[20] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424, 2000.

[21] Kristin P Bennett and J Blue. Optimal decision trees. *Rensselaer Polytechnic Institute Math Report*, 214, 1996.

[22] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, 120:70 – 83, 2018.

[23] Donald A Berry, Kathleen A Cronin, Sylvia K Plevritis, Dennis G Fryback, Lauren Clarke, Marvin Zelen, Jeanne S Mandelblatt, Andrei Y Yakovlev, J Dik F Habbema, and Eric J Feuer. Effect of screening and adjuvant therapy on mortality from breast cancer. *New England Journal of Medicine*, 353(17):1784–1792, 2005.

[24] Andrea Bertolini. Robots as products: the case for a realistic analysis of robotic applications and liability rules. *Law, innovation and technology*, 5(2):214–247, 2013.

[25] Andrea Bertolini, Pericle Salvini, Teresa Pagliai, Annagiulia Morachioli, Giorgia Acerbi, Filippo Cavallo, Giuseppe Turchetti, and Paolo Dario. On robots and insurance. *International Journal of Social Robotics*, 8(3):381–391, 2016.

[26] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[27] D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

[28] D Bertsimas, E Gibson, and A Orfanoudaki. Optimal survival trees. Working Paper.

[29] D. Bertsimas, N. Kallus, A. Weinstein, and Y. Zhuo. Personalized diabetes management using electronic medical records. *Diabetes Care*, 40(2):210–217, 2017.

[30] D Bertsimas and A Orfanoudaki. Pricing algorithmic risk. Working paper.

[31] D Bertsimas, A Orfanoudaki, and H Wiberg. Interpretable clustering: An optimization approach. *Machine Learning*, pages 1–50, 2020. Machine Learning for Health (ML4H) Workshop at NeurIPS 2018; arXiv preprint arXiv:1812.00539.

[32] D Bertsimas, C Pawlowski, and Y D Zhuo. From predictive methods to missing data imputation: An optimization approach. *Journal of Machine Learning Research*, 18(196):1–39, 2018.

[33] Dimitris Bertsimas, Arthur Delarue, Patrick Jaillet, and Sebastien Martin. The price of interpretability. *arXiv preprint arXiv:1907.03419*, 2019.

[34] Dimitris Bertsimas and Jack Dunn. *Machine learning under a modern optimization lens*. Dynamic Ideas LLC, 2019.

[35] Dimitris Bertsimas, Jack Dunn, and Nishanth Mundru. Optimal prescriptive trees. *INFORMS Journal on Optimization*, 1(2):164–183, 2019.

[36] Dimitris Bertsimas, Jack Dunn, Colin Pawlowski, John Silberholz, Alexander Weinstein, Ying Daisy Zhuo, Eddy Chen, and Aymen A Elfiky. Applied informatics decision support tool for mortality predictions in patients with cancer. *JCO clinical cancer informatics*, 2:1–11, 2018.

[37] Dimitris Bertsimas, Jack Dunn, George C. Velmahos, and Haytham M. A. Kaafarani. Surgical risk is not linear: Derivation and validation of a novel, user-friendly, and machine-learning-based predictive optimal trees in emergency surgery risk (potter) calculator. *Annals of Surgery*, 268(4):574–583, 2018.

[38] Dimitris Bertsimas, Nathan Kallus, Alexander M Weinstein, and Ying Daisy Zhuo. Personalized diabetes management using electronic medical records. *Diabetes care*, 40(2):210–217, 2017.

[39] Dimitris Bertsimas, Jerry Kung, Nikolaos Trichakis, Yuchen Wang, Ryutaro Hirose, and Parsia A Vagefi. Development and validation of an optimized prediction of mortality for candidates awaiting liver transplantation. *American Journal of Transplantation*, 19(4):1109–1118, 2019.

[40] Dimitris Bertsimas, Allison K. O'Hair, and William R. Pulleybank. *The Analytics Edge*. Dynamic Ideas LLC, 2016.

[41] Dimitris Bertsimas, Agni Orfanoudaki, and Colin Pawlowski. Imputation of clinical covariates in time series. *Machine Learning*, 110(1):185–248, 2021.

[42] Dimitris Bertsimas, Agni Orfanoudaki, and Rory B Weiner. Personalized treatment for coronary artery disease patients: a machine learning approach. *Health Care Management Science*, 23(4):482–506, 2020.

[43] Dimitris Bertsimas, Colin Pawlowski, and Ying Daisy Zhuo. From predictive methods to missing data imputation: an optimization approach. *The Journal of Machine Learning Research*, 18(1):7133–7171, 2018.

[44] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[45] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[46] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, 2012.

[47] Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in medicine*, 17(10):1169–1186, 1998.

[48] Sebastien Bineau, Carole Dufouil, Catherine Helmer, Karen Ritchie, Jean-Philippe Empana, Pierre Ducimetiere, Annick Alperovitch, Marie Germaine Bousser, and Christophe Tzourio. Framingham stroke risk function in a large population-based cohort of elderly people: the 3c study. *Stroke*, 40(5):1564–1570, 2009.

[49] Guthrie S. Birkhead, Michael Klompas, and Nirav R. Shah. Uses of electronic health records for public health surveillance to advance public health. *Annual Review of Public Health*, 36(1):345–359, 2015. PMID: 25581157.

[50] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. *arXiv preprint cs/0011032*, 2000.

[51] William E. Boden, Robert A. O'Rourke, Koon K. Teo, Pamela M. Hartigan, David J. Maron, William J. Kostuk, Merril Knudtson, Marcin Dada, Paul Casperson, Crystal L. Harris, Bernard R. Chaitman, Leslee Shaw, Gilbert Gosselin, Shah Nawaz, Lawrence M. Title, Gerald Gau, Alvin S. Blaustein, David C. Booth, Eric R. Bates, John A. Spertus, Daniel S. Berman, G.B. John Mancini, and William S. Weintraub. Optimal medical therapy with or without pci for stable coronary disease. *New England Journal of Medicine*, 356(15):1503–1516, 2007. PMID: 17387127.

[52] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[53] Imad Bou-Hamad, Denis Larocque, Hatem Ben-Ameur, et al. A review of survival trees. *Statistics Surveys*, 5:44–71, 2011.

[54] L Breiman. Software for the masses, 2002.

[55] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[56] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[57] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthey Weather Review*, 78(1):1–3, 1950.

[58] Leslie L Bronner, Daniel S Kanter, and JoAnn E Manson. Primary prevention of stroke. *New England Journal of Medicine*, 333(21):1392–1400, 1995.

[59] Federico Cabitza, Raffaele Rasoini, and Gian Franco Gensini. Unintended consequences of machine learning in medicine. *Jama*, 318(6):517–518, 2017.

[60] Alison Callahan and Nigam H. Shah. Chapter 19: Machine learning in healthcare. In Aziz Sheikh, Kathrin M. Cresswell, Adam Wright, and David W. Bates, editors, *Key Advances in Clinical Informatics*, pages 279 – 291. Academic Press, 2017.

[61] Luis M Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and buildings*, 140:81–97, 2017.

[62] Raphael Carandang, Sudha Seshadri, Alexa Beiser, Margaret Kelly-Hayes, Carlos S Kase, William B Kannel, and Philip A Wolf. Trends in incidence, lifetime risk, severity, and 30-day mortality of stroke over the past 50 years. *Jama*, 296(24):2939–2946, 2006.

[63] Robert M Carey and Paul K Whelton. Prevention, detection, evaluation, and management of high blood pressure in adults: synopsis of the 2017 american college of cardiology/american heart association hypertension guideline. *Annals of internal medicine*, 168(5):351–358, 2018.

[64] E. J. Caruana, M. Roman, J. Hernández-Sánchez, and P. Solli. Longitudinal studies. *Journal of thoracic disease*, 7(11):E537–40, 2015.

[65] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[66] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[67] Newgard CD and Lewis RJ. Missing data: How to best account for what is not known. *JAMA*, 314(9):940–941, 2015.

[68] Lloyd E Chambless, Gerardo Heiss, Eyal Shahar, Mary Jo Earp, and James Toole. Prediction of ischemic stroke risk in the atherosclerosis risk in communities study. *American journal of epidemiology*, 160(3):259–269, 2004.

[69] Danton S Char, Nigam H Shah, and David Magnus. Implementing machine learning in health care—addressing ethical challenges. *The New England journal of medicine*, 378(11):981, 2018.

[70] Marie Chavent, Christiane Guinot, Yves Lechevallier, and Michel Tenenhaus. Méthodes divisives de classification et segmentation non supervisée : recherche d'une typologie de la peau humaine saine. *Revue de Statistique Appliquée*, 47(4):87–99, 1999.

[71] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002.

[72] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

[73] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *arXiv preprint arXiv:1603.02754*, 2016.

[74] Francois Chollet et al. Keras, 2015.

[75] Philip A. Chou. Optimal partitioning for classification and regression trees. *IEEE Computer Architecture Letters*, 13(04):340–354, 1991.

[76] A Ciampi, C-H Chang, S Hogg, and S McKinney. Recursive partition: A versatile method for exploratory-data analysis in biostatistics. In *Biostatistics*, pages 23–50. Springer, 1987.

[77] Antonio Ciampi, Johanne Thiffault, Jean-Pierre Nakache, and Bernard Asselain. Stratification by stepwise regression, correspondence analysis and recursive partition: a comparison of three methods of analysis for survival data with covariates. *Computational statistics & data analysis*, 4(3):185–204, 1986.

[78] Cary Coglianese and David Lehr. Regulating by robot: Administrative decision making in the machine-learning era. *Geo. LJ*, 105:1147, 2016.

[79] RM Conroy, K Pyörälä, AP el Fitzgerald, S Sans, A Menotti, Gui De Backer, Dirk De Bacquer, P Ducimetiere, P Jousilahti, U Keil, et al. Estimation of ten-year risk of fatal cardiovascular disease in europe: the score project. *European heart journal*, 24(11):987–1003, 2003.

[80] Andrea Coraddu, Luca Oneto, Aessandro Ghio, Stefano Savio, Davide Anguita, and Massimo Figari. Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 230(1):136–153, 2016.

[81] Paulo Cortez, António Cerdeira, Fernando L. Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47:547–553, 2009.

[82] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[83] BD Cox, M Blaxter, ALJ Buckle, NP Fenner, JF Golding, M Gore, FA Huppert, J Nickson, M Roth, J Stark, et al. Health and lifestyle survey, 1984–1985. *computer file]. Colchester, Essex: UK Data Archive [distributor]*, 1988.

[84] D. R. Cox. Regression Models and Life-Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972.

[85] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[86] David R Cox. Partial likelihood. *Biometrika*, 62(2):269–276, 1975.

[87] David R Cox et al. Regression models and life tables (with discussion). *Journal of the Royal Statistical Society*, 34:187–220, 1972.

[88] Nicholas L Crookston and Andrew O Finley. yaimpute: an r package for knn imputation. *Journal of Statistical Software. 23 (10). 16 p.*, 2008.

[89] J David Cummins. Asset pricing models and insurance ratemaking. *ASTIN Bulletin: The Journal of the IAA*, 20(2):125–166, 1990.

[90] R. B. D'Agostino, P. A. Wolf, A. J. Belanger, and W. B. Kannel. Stroke risk profile: adjustment for antihypertensive medication. the framingham study. *Stroke*, 25(1):40–3, 1994.

[91] Ralph B. D'Agostino, Michael J. Pencina, Joseph M. Massaro, and Sean Coady. Cardiovascular disease risk assessment: Insights from Framingham. *Global Heart*, 8(1):11 – 23, 2013. Framingham Legacy Issue.

[92] Ralph B. D'Agostino, Ramachandran S. Vasan, Michael J. Pencina, Philip A. Wolf, Mark Cobain, Joseph M. Massaro, and William B. Kannel. General cardiovascular risk profile for use in primary care. *Circulation*, 117, 2008.

[93] RB D'Agostino, JL Griffith, CH Schmid, and N Terrin. Measures for evaluating model performance. In *Prroceedings-American Statistical Association Biometrics Section*, pages 253–258, 1997.

[94] Roger B Davis and James R Anderson. Exponential survival trees. *Statistics in Medicine*, 8(8):947–961, 1989.

[95] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[96] E. Diday and J. C. Simon. *Clustering Analysis*, pages 47–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 1976.

[97] Thomas G Dietterich. Statistical tests for comparing supervised classification learning algorithms. *Oregon State University Technical Report*, 1:1–24, 1996.

[98] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114, 2015.

[99] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. Overcoming algorithm aversion: People will use imperfect algorithms if they can (even slightly) modify them. *Management Science*, 64(3):1155–1170, 2018.

[100] M. Dietz. Gan-sandbox, 2017.

[101] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning.(2017), 2017.

[102] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[103] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[104] Tony Duan, Pranav Rajpurkar, Dillon Laird, Andrew Y Ng, and Sanjay Basu. Clinical value of predicting individual treatment effects for intensive blood pressure therapy: A machine learning experiment to estimate treatment effects from randomized trial data. *Circulation: Cardiovascular Quality and Outcomes*, 12(3):e005010, 2019.

[105] Carole Dufouil, Alexa Beiser, Leslie A McLure, Philip A Wolf, Christophe Tzourio, Virginia J Howard, Andrew J Westwood, Jayandra J Himali, Lisa Sullivan, and Hugo J Aparicio. Revised framingham stroke risk profile to reflect temporal trends. *Circulation*, 135(12):1145–1159, 2017.

[106] J Dunn. *Optimal Trees for Prediction and Prescription*. PhD thesis, Massachusetts Institute of Technology, 2018.

[107] J C Dunn. Well-Separated Clusters and Optimal Fuzzy Partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[108] Jack William Dunn. *Optimal trees for prediction and prescription*. PhD thesis, Massachusetts Institute of Technology, 2018.

[109] Benjamin Duran and Patrick Odell. *Cluster Analysis*. 100. Springer-Verlag Berlin Heidelberg, 1 edition, 1974.

[110] Ralph B D'agostino, Ramachandran S Vasan, Michael J Pencina, Philip A Wolf, Mark Cobain, Joseph M Massaro, and William B Kannel. General cardiovascular risk profile for use in primary care. *Circulation*, 117(6):743–753, 2008.

[111] Joseph E. Ebinger, Brandon R. Porten, Craig E. Strauss, Ross F. Garberich, Christopher Han, Sharon K. Wahl, Benjamin C. Sun, Raed H. Abdelhadi, and Timothy D. Henry. Design, challenges, and implications of quality improvement projects using the electronic medical record. *Circulation: Cardiovascular Quality and Outcomes*, 9(5):593–599, 2016.

[112] Ezekiel J. Emanuel and Robert M. Wachter. Artificial Intelligence in Health Care: Will the Value Match the Hype?Artificial Intelligence in Health Care—Will the Value Match the Hype?Artificial Intelligence in Health Care Will the Value Match the Hype? *JAMA*, 05 2019.

[113] Paul Embrechts. Actuarial versus financial pricing of insurance. *The Journal of Risk Finance*, 2000.

[114] Christina Christina Lynn Epstein. *An analytics approach to hypertension treatment*. PhD thesis, Massachusetts Institute of Technology, 2014.

[115] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[116] James Everhart and David Wright. Diabetes mellitus as a risk factor for pancreatic cancer: a meta-analysis. *Jama*, 273(20):1605–1609, 1995.

[117] Ludger Evers and Claudia-Martina Messow. Sparse kernel methods for high-dimensional survival data. *Bioinformatics*, 24(14):1632–1638, 2008.

[118] Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2-3):113–127, 2014.

[119] José Cláudio Faria, Clarice Garcia Borges Demétrio, and Ivan Bezerra Allaman. *bpca: Biplot of Multivariate Data Based on Principal Components Analysis*. UESC and ESALQ, Ilheus, Bahia, Brasil and Piracicaba, Sao Paulo, Brasil, 2018.

[120] Michael E. Farkouh, Michael Domanski, Lynn A. Sleeper, Flora S. Siami, George Dangas, Michael Mack, May Yang, David J. Cohen, Yves Rosenberg, Scott D. Solomon, Akshay S. Desai, Bernard J. Gersh, Elizabeth A. Magnuson, Alexandra Lansky, Robin Boineau, Jesse Weinberger, Krishnan Ramanathan, J. Eduardo Sousa, Jamie Rankin, Balram Bhargava, John Buse, Whady Hueb, Craig R. Smith, Victoria Muratov, Sameer Bansilal, Spencer III King, Michel Bertrand, and Valentin Fuster. Strategies for multivessel revascularization in patients with diabetes. *New England Journal of Medicine*, 367(25):2375–2384, 2012. PMID: 23121323.

[121] FDA. Clinical and patient decision support software - guidance for industry and food and drug administration staff. Available at `http://www.fda.gov/regulatory-information/search-fda-guidance-documents/clinical-and-patient-decision-support-software` (2017/05/27), 2017.

[122] M Feinleib, W B Kannel, R J Garrison, P M McNamara, and W P Castelli. The framingham offspring study. design and preliminary data. *Preventive Medicine*, 4(4):518–525, 1975.

[123] Michael L Feldstein, Edwin D Savlov, and Russell Hilf. A statistical model for predicting response of breast cancer patients to cytotoxic chemotherapy. *Cancer research*, 38(8):2544–2548, 1978.

[124] Stephan D. Fihn, James C. Blankenship, Karen P. Alexander, John A. Bittl, John G. Byrne, Barbara J. Fletcher, Gregg C. Fonarow, Richard A. Lange, Glenn N. Levine, Thomas M. Maddox, Srihari S. Naidu, E. Magnus Ohman, and Peter K. Smith. 2014 acc/aha/aats/pcna/scai/sts focused update of the guideline for the diagnosis and management of patients with stable ischemic heart disease: A report of the american college of cardiology/american heart association task force on practice guidelines, and the american association for thoracic surgery, preventive cardiovascular nurses association, society for cardiovascular angiography and interventions, and society of thoracic surgeons. *Journal of the American College of Cardiology*, 64(18):1929 – 1949, 2014.

[125] Stephan D. Fihn, Julius M. Gardin, Jonathan Abrams, Kathleen Berra, James C. Blankenship, Apostolos P. Dallas, Pamela S. Douglas, JoAnne M. Foody, Thomas C. Gerber, Alan L. Hinderliter, Spencer B. King, Paul D. Kligfield, Harlan M. Krumholz, Raymond Y.K. Kwong, Michael J. Lim, Jane A. Linderbaum, Michael J. Mack, Mark A. Munger, Richard L. Prager, Joseph F. Sabik, Leslee J. Shaw, Joanna D. Sikkema, Craig R. Smith, Sidney C. Smith, John A. Spertus, and Sankey V. Williams. 2012 accf/aha/acp/aats/pcna/scai/sts guideline for the diagnosis and management of patients with stable ischemic heart disease: A report of the american college of cardiology foundation/american heart association task force on practice guidelines, and the american college of physicians, american association for thoracic surgery, preventive cardiovascular nurses association, society for cardiovascular angiography and interventions, and society of thoracic surgeons. *Circulation*, 60(24):e44 – e164, 2015.

[126] Anibal Flores, Hugo Tito, and Carlos Silva. Local average of nearest neighbors: Univariate time series imputation. *International Journal of Advanced Computer Science and Applications*, 10(8):45–50, 2019.

[127] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

[128] Stephane Fotso. Deep neural networks for survival analysis based on a multi-task framework. *arXiv preprint arXiv:1801.05512*, 2018.

[129] Césaire JK Fouodo, Inke R König, Claus Weihs, Andreas Ziegler, and Marvin N Wright. Support vector machines for survival analysis with r. *R Journal*, 10(1), 2018.

[130] Gary N Fox and Nashat S Moawad. Uptodate: a comprehensive clinical database. *Journal of family practice*, 52(9):706–710, 2003.

[131] Ricardo Fraiman, Badih Ghattas, and Marcela Svarc. Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7(2):125–145, 2013.

[132] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009.

[133] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[134] Holger Frohlich, Rudi Balling, Niko Beerenwinkel, Oliver Kohlbacher, Santosh Kumar, Thomas Lengauer, Marloes H. Maathuis, Yves Moreau, Susan A. Murphy, Teresa M. Przytycka, Michael Rebhan, Hannes Rost, Andreas Schuppert, Matthias Schwab, Rainer Spang, Daniel Stekhoven, Jimeng Sun, Andreas Weber, Daniel Ziemek, and Blaz Zupan. From hype to reality: data science enabling personalized medicine. *BMC Medicine*, 16(1):150, Aug 2018.

[135] Valentin Fuster, Lina Badimon, Juan J Badimon, and James H Chesebro. The pathogenesis of coronary artery disease and the acute coronary syndromes. *New England journal of medicine*, 326(5):310–318, 1992.

[136] Brian F Gage, Amy D Waterman, William Shannon, Michael Boechler, Michael W Rich, and Martha J Radford. Validation of clinical classification schemes for predicting stroke: results from the national registry of atrial fibrillation. *Jama*, 285(22):2864–2870, 2001.

[137] Ravi Garg, Elissa Oh, Andrew Naidech, Konrad Kording, and Shyam Prabhakaran. Automating ischemic stroke subtype classification using machine learning and natural language processing. *Journal of Stroke and Cerebrovascular Diseases*, 28(7):2045–2051, 2019.

[138] Milena A Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. Potential biases in machine learning algorithms using electronic health record data. *JAMA internal medicine*, 178(11):1544–1547, 11 2018.

[139] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[140] Gismondi&Associates. Unnecessary mastectomy performed due to misdiagnosis | the law offices of gismondi & associates. `https://www.gislaw.com/2012/11/unnecessary-mastectomy-performed-due-to-misdiagnosis/`. (Accessed on 03/12/2021).

[141] John C Gittins, Kevin D Glazebrook, Richard Weber, and Richard Weber. *Multi-armed bandit allocation indices*, volume 25. Wiley Online Library, 1989.

[142] Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar. Rnn-surv: A deep recurrent model for survival analysis. In *International Conference on Artificial Neural Networks*, pages 23–32. Springer, 2018.

[143] David C. Goff, Donald M. Lloyd-Jones, Glen Bennett, Sean Coady, Ralph B. D'Agostino, Raymond Gibbons, Philip Greenland, Daniel T. Lackland, Daniel Levy, Christopher J. O'Donnell, Jennifer G. Robinson, J. Sanford Schwartz, Susan T. Shero, Sidney C. Smith, Paul Sorlie, Neil J. Stone, and Peter W.F. Wilson. 2013 acc/aha guideline on the assessment of cardiovascular risk. *Journal of the American College of Cardiology*, 63(25 Part B):2935–2959, 2014.

[144] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

[145] Alexander Goldenshluger and Assaf Zeevi. A linear response bandit problem. *Stochastic Systems*, 3(1):230–261, 2013.

[146] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2017.

[147] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[148] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

[149] Louis Gordon and Richard A Olshen. Tree-structured survival analysis. *Cancer treatment reports*, 69(10):1065–1069, 1985.

[150] E Graf, C Schmoor, W Sauerbrei, and M Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

[151] Thomas Grubinger, Achim Zeileis, and Karl-Peter Pfeiffer. evtree: Evolutionary learning of globally optimal classification and regression trees in r. *Journal of statistical software*, 61(1):1–29, 2014.

[152] Kevin Gurney. *An introduction to neural networks*. CRC press, 2014.

[153] Michael Hahsler, Matthew Piekenbrock, and Derek Doran. dbscan: Fast density-based clustering with r. *Journal of Statistical Software, Articles*, 91(1):1–30, 2019.

[154] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, Dec 2001.

[155] Margaret A Hamburg and Francis S Collins. The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304, 2010.

[156] T P Hancock, D H Coomans, and Y L Everingham. Supervised Hierarchical Clustering Using CART. In *Proceedings of MODSIM 2003 International Congress on Modelling and Simulation*, pages 1880–1885, Townsville, QLD, Australia, 2003.

[157] J.A. Hanley and Barbara Mcneil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 05 1982.

[158] Göran K Hansson. Inflammation, atherosclerosis, and coronary artery disease. *New England Journal of Medicine*, 352(16):1685–1695, 2005.

[159] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

[160] P. A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, and J. G. Conde. Research electronic data capture (redcap)–a metadata-driven methodology and workflow process for providing translational research informatics support. *J Biomed Inform*, 42(2):377–81, 2009.

[161] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[162] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In *The elements of statistical learning*, pages 485–585. Springer, 2009.

[163] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[164] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[165] Pamela Herd, Deborah Carr, and Carol Roan. Cohort profile: Wisconsin longitudinal study (wls). *International journal of epidemiology*, 43(1):34–41, 2014.

[166] James Honaker, Anne Joseph, Gary King, Kenneth Scheve, and Naunihal Singh. Amelia: A program for missing data. *Department of Government, Harvard University*, 1999.

[167] James Honaker, Gary King, and Matthew Blackwell. Amelia II: A program for missing data. *Journal of Statistical Software, Articles*, 45(7):1–47, 2011.

[168] Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark J Van Der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2005.

[169] Torsten Hothorn, Kurt Hornik, Carolin Strobl, and Achim Zeileis. Party: A laboratory for recursive partytioning, 2010.

[170] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006.

[171] Torsten Hothorn, Berthold Lausen, Axel Benner, and Martin Radespiel-Tröger. Bagging survival trees. *Statistics in medicine*, 23(1):77–91, 2004.

[172] Joseph G Ibrahim, Ming-Hui Chen, and Debajyoti Sinha. Bayesian survival analysis. *Wiley StatsRef: Statistics Reference Online*, 2014.

[173] Joseph G Ibrahim, Haitao Chu, and Ming-Hui Chen. Missing data in clinical studies: issues and methods. *Journal of clinical oncology*, 30(26):3297, 2012.

[174] Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* Cambridge University Press, New York, NY, USA, 2015.

[175] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, pages 841–860, 2008.

[176] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[177] Kristel J.M. Janssen, A. Rogier T. Donders, Frank E. Harrell, Yvonne Vergouwe, Qingxia Chen, Diederick E. Grobbee, and Karel G.M. Moons. Missing covariate data in medical research: To impute is better than to ignore. *Journal of Clinical Epidemiology*, 63(7):721 – 727, 2010.

[178] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[179] Sydney A Jones, Rebecca F Gottesman, Eyal Shahar, Lisa Wruck, and Wayne D Rosamond. Validity of hospital discharge diagnosis codes for stroke: the atherosclerosis risk in communities study. *Stroke*, 45(11):3219–3225, 2014.

[180] Philippe Jorion. *Value at risk*. McGraw-Hill Professional Publishing, 2000.

[181] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[182] Nathan Kallus. Recursive partitioning for personalization using observational data. In *International Conference on Machine Learning*, pages 1789–1798, 2017.

[183] Singh Kamaljot, Sandhu Ranjeet Kaur, and Dinesh Kumar. Comment volume prediction using neural networks and decision trees. In *IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015)*, Cambridge, United Kingdom, March 2015.

[184] William B Kannel. Blood pressure as a cardiovascular risk factor: prevention and treatment. *Jama*, 275(20):1571–1576, 1996.

[185] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[186] Manohar Kaul, Bin Yang, and Christian S Jensen. Building accurate 3d spatial networks to enable next generation intelligent transportation systems. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 137–146. IEEE, 2013.

[187] François Kawala, Ahlame Douzal-Chouakria, Eric Gaussier, and Eustache Dimert. Prédictions d'activité dans les réseaux sociaux en ligne. In *4ième conférence sur les modèles et l'analyse des réseaux : Approches mathématiques et informatiques*, page 16, France, October 2013.

[188] Aditya Khosla, Yu Cao, Cliff Chiung-Yu Lin, Hsu-Kuang Chiu, Junling Hu, and Honglak Lee. An integrated machine learning approach to stroke prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 183–192, 2010.

[189] Chulho Kim, Vivienne Zhu, Jihad Obeid, and Leslie Lenert. Natural language processing and machine learning algorithm to identify brain mri reports with acute ischemic stroke. *PloS one*, 14(2):e0212778, 2019.

[190] Hyo-Eun Kim, Hak Hee Kim, Boo-Kyung Han, Ki Hwan Kim, Kyunghwa Han, Hyeonseob Nam, Eun Hye Lee, and Eun-Kyung Kim. Changes in cancer detection and false-positive recall in mammography using artificial intelligence: a retrospective, multireader study. *The Lancet Digital Health*, 2(3):e138–e148, 2020.

[191] Gary King, James Honaker, Anne Joseph, and Kenneth Scheve. Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *American political science review*, 95(1):49–69, 2001.

[192] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[193] Hamid Krim and A Ben Hamza. *Geometric methods in signal and image analysis*. Cambridge University Press, 2015.

[194] Chayakrit Krittanawong, HongJu Zhang, Zhen Wang, Mehmet Aydar, and Takeshi Kitai. Artificial intelligence in precision cardiovascular medicine. *Journal of the American College of Cardiology*, 69(21):2657 – 2664, 2017.

[195] SW Lagakos. General right censoring and its impact on the analysis of survival data. *Biometrics*, 35(1):139–156, March 1979.

[196] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015.

[197] Mary Beth Landrum and Mark P. Becker. A multiple imputation strategy for incomplete longitudinal data. *Statistics in Medicine*, 20(17-18):2741–2760, 2001.

[198] Daniel T Larose and Chantal D Larose. *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, 2014.

[199] Susanna C Larsson, Alice Wallin, Alicja Wolk, and Hugh S Markus. Differing association of alcohol consumption with different stroke types: a systematic review and meta-analysis. *BMC medicine*, 14(1):178, 2016.

[200] Hyafil Laurent and Ronald L Rivest. Constructing optimal binary decision trees is np-complete. *Information processing letters*, 5(1):15–17, 1976.

[201] Michelle T Le, Carmel E Mothersill, Colin B Seymour, and Fiona E McNeill. Is the false-positive rate in mammography in north america too high? *The British journal of radiology*, 89(1065):20160045, 2016.

[202] Michael LeBlanc and John Crowley. Relative risk trees for censored survival data. *Biometrics*, pages 411–425, 1992.

[203] Michael LeBlanc and John Crowley. Survival trees by goodness of split. *Journal of the American Statistical Association*, 88(422):457–467, 1993.

[204] Michelle V Lee, Katerina Konstantinoff, Alison Gegios, Katie Miles, Catherine Appleton, and Dawn Hui. Breast cancer malpractice litigation: A 10-year analysis and update in trends. *Clinical imaging*, 60(1):26–32, 2020.

[205] Scott H Lee. Natural language generation for electronic health records. *NPJ digital medicine*, 1(1):1–7, 2018.

[206] LJ Lesko. Personalized medicine: elusive dream or imminent reality? *Clinical Pharmacology & Therapeutics*, 81(6):807–816, 2007.

[207] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[208] Andrew S. Levey, Kai-Uwe Eckardt, Yusuke Tsukamoto, Adeera Levin, Josef Coresh, Jerome Rossert, Dick D.E. Zeeuw, Thomas H. Hostetter, Norbert Lameire, and Garabed Eknoyan. Definition and classification of Chronic Kidney Disease: A position statement from kidney disease: Improving global outcomes (kdigo). *Kidney International*, 67(6):2089 – 2100, 2005.

[209] D Levy and S Brink. *A change of heart : unraveling the mysteries of cardiovascular disease*. New York : Vintage, 2006.

[210] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[211] Rumeng Li, Baotian Hu, Feifan Liu, Weisong Liu, Francesca Cunningham, David D McManus, and Hong Yu. Detection of bleeding events in electronic health record notes using convolutional neural network models enhanced with recurrent neural network autoencoders: deep learning approach. *JMIR medical informatics*, 7(1):e10788, 2019.

[212] Xuan Liang, Shuo Li, Shuyi Zhang, Hui Huang, and Song Xi Chen. Pm2.5 data reliability, consistency, and air quality assessment in five chinese cities. *Journal of Geophysical Research: Atmospheres*, 121(17):10,220–10,236, 2016.

[213] Zhaohui Liang, Jun Liu, Aihua Ou, Honglai Zhang, Ziping Li, and Jimmy Xiangji Huang. Deep generative learning for automated ehr diagnosis of traditional chinese medicine. *Computer methods and programs in biomedicine*, 174:17–23, 2019.

[214] Knut Liestbl, Per Kragh Andersen, and Ulrich Andersen. Survival analysis and neural nets. *Statistics in medicine*, 13(12):1189–1200, 1994.

[215] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[216] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[217] Zachary C Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with RNNs: Improved classification of clinical time series. In Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 253–270, Children's Hospital LA, Los Angeles, CA, USA, 18–19 Aug 2016. PMLR.

[218] Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*, volume 793. Wiley, 2019.

[219] Bing Liu, Yiyuan Xia, and Philip S Yu. Clustering through decision tree construction. In *Proceedings of the ninth international conference on Information and knowledge management - CIKM '00*, pages 20–29, McLean, VA, 2000.

[220] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE, 2010.

[221] Yi-cheng Liu and I-Cheng Yeh. Using mixture design and neural networks to build stock selection decision support systems. *Neural Computing and Applications*, 28, 11 2015.

[222] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[223] Syed S Mahmood, Daniel Levy, Ramachandran S Vasan, and Thomas J Wang. The Framingham Heart Study and the epidemiology of cardiovascular disease: A historical perspective. *The Lancet*, 383(9921):999 – 1008, 2014.

[224] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[225] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[226] Teri A Manolio, Richard A Kronmal, Gregory L Burke, Daniel H O'Leary, and Thomas R Price. Short-term predictors of incident stroke in older adults: the cardiovascular health study. *Stroke*, 27(9):1479–1486, 1996.

[227] Kenneth Marek, Danna Jennings, Shirley Lasch, Andrew Siderowf, Caroline Tanner, Tanya Simuni, Chris Coffey, Karl Kieburtz, Emily Flagg, Sohini Chowdhury, et al. The parkinson progression marker initiative (ppmi). *Progress in neurobiology*, 95(4):629–635, 2011.

[228] Harry M Markowitz. Portfolio theory: as i still see it. *Annu. Rev. Financ. Econ.*, 2(1):1–23, 2010.

[229] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654, 2002.

[230] Leslie A McClure, Dawn O Kleindorfer, Brett M Kissela, Mary Cushman, Elsayed Z Soliman, and George Howard. Assessing the performance of the framingham stroke risk score in the reasons for geographic and racial differences in stroke cohort. *Stroke*, 45(6):1716–1720, 2014.

[231] Susan McKeever and Manhar Singh Walia. *Synthesising Tabular Datasets Using Wasserstein Conditional GANS with Gradient Penalty (WCGAN-GP)*. Technological University Dublin, 2020.

[232] Alexander J McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton university press, 2015.

[233] A David Mendelow, Eng H Lo, Ralph L Sacco, MD MS FAHA FAAN, and Lawrence KS Wong. *Stroke: pathophysiology, diagnosis, and management*. Elsevier Health Sciences, 2015.

[234] A. Metzger, P. Leitner, D. Ivanović, E. Schmieders, R. Franklin, M. Carro, S. Dustdar, and K. Pohl. Comparing and combining predictive business process monitoring techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2):276–290, Feb 2015.

[235] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[236] MichaelW Millar-Craig, CharlesN Bishop, and E.B Raftery. Circadian variation of blood-pressure. *The Lancet*, 311(8068):795 – 797, 1978. Originally published as Volume 1, Issue 8068.

[237] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[238] Brent Daniel Mittelstadt, Patrick Allo, Mariarosaria Taddeo, Sandra Wachter, and Luciano Floridi. The ethics of algorithms: Mapping the debate. *Big Data & Society*, 3(2):2053951716679679, 2016.

[239] Ahmed Mohamed, Ahmet Rizaner, and Ali Hakan Ulusoy. Using data mining to predict instructor performance. *Procedia Computer Science*, 102:137 – 142, 2016. 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria.

[240] Annette M Molinaro, Sandrine Dudoit, and Mark J Van der Laan. Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis*, 90(1):154–177, 2004.

[241] Sergio Moro, Paulo Rita, and Joana Coelho. Stripping customers' feedback on hotels through data mining: The case of Las Vegas Strip. *Tourism Management Perspectives*, 23:41 – 52, 2017.

[242] Sergio Moro, Paulo Rita, and Bernardo Vala. Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. *Journal of Business Research*, 69(9):3341 – 3351, 2016.

[243] Lampros Mouselimis. *ClusterR: Gaussian Mixture Models, K-Means, Mini-Batch-Kmeans, K-Medoids and Affinity Propagation Clustering*, 2019. R package version 1.2.0.

[244] Nina Narodytska, Alexey Ignatiev, Filipe Pereira, Joao Marques-Silva, and IS RAS. Learning optimal decision trees with sat. In *IJCAI*, pages 1362–1368, 2018.

[245] Ziad S Nasreddine, Natalie A Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L Cummings, and Howard Chertkow. The montreal cognitive assessment, moca: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005.

[246] Wayne Nelson. Theory and applications of hazard plotting for censored failure data. *Technometrics*, 14(4):945–966, 1972.

[247] Linda Nevin, PLOS Medicine Editors, et al. Advancing the beneficial use of machine learning in health care and medicine: Toward a community understanding, 2018.

[248] Siegfried Nijssen and Elisa Fromont. Mining optimal decision trees from itemset lattices. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 530–539, 2007.

[249] Siegfried Nijssen and Elisa Fromont. Optimal constraint-based decision tree induction from itemset lattices. *Data Mining and Knowledge Discovery*, 21(1):9–51, 2010.

[250] Shigeyuki Oba, Masa-aki Sato, Ichiro Takemasa, Morito Monden, Ken-ichi Matsubara, and Shin Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

[251] Ziad Obermeyer and Ezekiel J. Emanuel. Predicting the future: Big data, machine learning, and clinical medicine. *New England Journal of Medicine*, 375(13):1216–1219, 2016. PMID: 27682033.

[252] Patrick J Offner, Ernest E Moore, and Walter L Biffl. Male gender is a risk factor for major infections after surgery. *Archives of Surgery*, 134(9):935–940, 1999.

[253] Tetsuya Ohira, Hiroyasu Iso, Hironori Imano, Akihiko Kitamura, Shinichi Sato, Yuko Nakagawa, Yoshihiko Naito, Tomoko Sankai, Takeshi Tanigawa, and Kazumasa Yamagishi. Prospective study of major and minor st-t abnormalities and risk of stroke among japanese. *Stroke*, 34(12):e250–e253, 2003.

[254] Alaa Mabrouk Salem Omar, Sukrit Narula, Mohamed Ahmed Abdel Rahman, Gianni Pedrizzetti, Hala Raslan, Osama Rifaie, Jagat Narula, and Partho P. Sengupta. Precision phenotyping in heart failure and pattern clustering of ultrasound data for the assessment of diastolic dysfunction. *JACC: Cardiovascular Imaging*, 10(11):1291 – 1303, 2017.

[255] Expert Panel on Detection, Evaluation, and Treatment of High Blood Cholesterol in Adults. Executive summary of the third report of the national cholesterol education program (ncep) expert panel on detection, evaluation, and treatment of high blood cholesterol in adults (adult treatment panel iii). *JAMA*, 285, 2001.

[256] C J Ong, A Orfanoudaki, R Zhang, F P Caprasse, M Hutch, L Ma, D Fard, O Balogun, M I Miller, M Minnig, et al. Machine learning and natural language processing methods to identify ischemic stroke, acuity and location from radiology reports. *PloS one*, 15(6):e0234908, 2020.

[257] A Orfanoudaki, E Chesley, C Cadisch, B Stein, A Nouh, M J Alberts, and D Bertsimas. Machine learning provides evidence that stroke risk is not linear: The non-linear framingham stroke risk score. *PloS one*, 15(5):e0232414, 2020.

[258] Agni Orfanoudaki. Algorithmicinsurance, 2021.

[259] Agni Orfanoudaki, Emma Chesley, Christian Cadisch, Barry Stein, Amre Nouh, Mark J Alberts, and Dimitris Bertsimas. Machine learning provides evidence that stroke risk

is not linear: The non-linear framingham stroke risk score. *PloS one*, 15(5):e0232414, 2020.

[260] Bruce Ovbiagele, Lee H Schwamm, Eric E Smith, Adrian F Hernandez, DaiWai M Olson, Wenqin Pan, Gregg C Fonarow, and Jeffrey L Saver. Recent nationwide trends in discharge statin treatment of hospitalized patients with stroke. *Stroke*, 41(7):1508–1513, 2010.

[261] Li P, Stuart EA, and Allison DB. Multiple imputation: A flexible tool for handling missing data. *JAMA*, 314(18):1966–1967, 2015.

[262] Priya Parmar, Rita Krishnamurthi, M Arfan Ikram, Albert Hofman, Saira S Mirza, Yury Varakin, Michael Kravchenko, Michael Piradov, Amanda G Thrift, Bo Norrving, et al. The s troke r iskometer tm a pp: Validation of a data collection tool and stroke risk predictor. *International Journal of Stroke*, 10(2):231–244, 2015.

[263] Kimberly C Paul, Jessica Schulz, Jeff M Bronstein, Christina M Lill, and Beate R Ritz. Association of polygenic risk score with cognitive decline and motor progression in parkinson disease. *JAMA neurology*, 75(3):360–366, 2018.

[264] Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[265] Alma B Pedersen, Ellen M Mikkelsen, Deirdre Cronin-Fenton, Nickolaj R Kristensen, Tra My Pham, Lars Pedersen, and Irene Petersen. Missing data and multiple imputation in clinical epidemiological research. *Clinical Epidemiology*, 9, 2017.

[266] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[267] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[268] Gregory Piatetsky-Shapiro, Chabane Djeraba, Lise Getoor, Robert Grossman, Ronen Feldman, and Mohammed Zaki. What are the grand challenges for data mining?: Kdd-2006 panel report. *ACM SIGKDD Explorations Newsletter*, 8(2):70–77, 2006.

[269] Tamar S Polonsky, Robyn L McClelland, Neal W Jorgensen, Diane E Bild, Gregory L Burke, Alan D Guerci, and Philip Greenland. Coronary artery calcium score and risk classification for coronary heart disease prediction. *Jama*, 303(16):1610–1616, 2010.

[270] Ewoud Pons, Loes MM Braun, MG Myriam Hunink, and Jan A Kors. Natural language processing in radiology: a systematic review. *Radiology*, 279(2):329–343, 2016.

[271] Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of statistics*, 39(2):1180, 2011.

[272] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[273] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[274] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[275] Martin Radespiel-Tröger, Thomas Rabenstein, H Thomas Schneider, and Berthold Lausen. Comparison of tree-based methods for prognostic stratification of survival data. *Artificial Intelligence in Medicine*, 28(3):323–341, 2003.

[276] Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919 – 938, 2004.

[277] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[278] Mohammad Hossein Rafiei and Hojjat Adeli. A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, 142(2):04015066, 2016.

[279] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

[280] C. Radhakrishna Rao. The use and interpretation of principal component analysis in applied research. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):329–358, 1964.

[281] Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing*, 308:49–57, 2018.

[282] Anupama Reddy and Louis-Philippe Kronek. Logical analysis of survival data: prognostic survival models by detecting high-degree interactions in right-censored data. *Bioinformatics*, 24(16):i248–i253, 08 2008.

[283] Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

[284] Chris Reed, Elizabeth Kennedy, and Sara Silva. Responsibility, autonomy and accountability: legal liability for machine learning. *Queen Mary School of Law Legal Studies Research Paper*, (243), 2016.

[285] Lars Rejnmark, Peter Vestergaard, and Leif Mosekilde. Treatment with beta-blockers, ace inhibitors, and calcium-channel blockers is associated with a reduced fracture risk: a nationwide case–control study. *Journal of hypertension*, 24(3):581–589, 2006.

[286] Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi, and Elvia M Quiroz. Internal versus external cluster validation indexes. *International Journal of computers and communications*, 5(1):27–34, 2011.

[287] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

[288] Paul M Ridker, Julie E. Buring, Nader Rifai, and Nancy R. Cook. Development and Validation of Improved Algorithms for the Assessment of Global Cardiovascular Risk in WomenThe Reynolds Risk Score. *JAMA*, 297(6):611–619, 02 2007.

[289] Brian D Ripley and Ruth M Ripley. Neural networks as statistical methods in survival analysis. *Clinical applications of artificial neural networks*, pages 237–255, 2001.

[290] R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[291] Jeanine E Roeters van Lennep, H. Tineke Westerveld, D. Willem Erkelens, and Ernst E van der Wall. Risk factors for coronary heart disease: implications of gender. *Cardiovascular Research*, 53(3):538–549, 02 2002.

[292] Foster Provost Ron Kohavi. Glossary of terms. *Machine Learning*, 30:271–274, 1998.

[293] Paul R Rosenbaum. *Design of observational studies*, volume 10. Springer, 2010.

[294] Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 04 1983.

[295] Rosen&Perry. Patient undergoes unnecessary mastectomy after being misdiagnosed with breast cancer | pittsburgh medical malpractice settlements | rosen & perry. `https://www.caringlawyers.com/verdicts-settlements/patient-undergoes-unnecessary-mastectomy-after-being-misdiagnosed-with-breast-cance` (Accessed on 03/12/2021).

[296] Russell Ross. Atherosclerosis—an inflammatory disease. *New England journal of medicine*, 340(2):115–126, 1999.

[297] P J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[298] Donald B Rubin. Comment: Neyman (1923) and causal inference in experiments and observational studies. *Statistical Science*, 5(4):472–480, 1990.

[299] Donald B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91(434):473–489, 1996.

[300] Li Rumeng, N Jagannatha Abhyuday, and Yu Hong. A hybrid neural network model for joint prediction of presence and period assertions of medical events in clinical notes. In *AMIA Annual Symposium Proceedings*, volume 2017, page 1149. American Medical Informatics Association, 2017.

[301] Enrique H. Ruspini. Numerical methods for fuzzy clustering. *Information Sciences*, 2(3):319–350, 1970.

[302] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[303] Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer, 2019.

[304] Joseph L. Schafer and Maren K. Olsen. Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behavioral Research*, 33(4):545–571, 1998. PMID: 26753828.

[305] Philipp Schmidt and Felix Biessmann. Quantifying interpretability and trust in machine learning systems. *arXiv preprint arXiv:1901.08558*, 2019.

[306] Kenneth F Schulz, Iain Chalmers, Richard J Hayes, and Douglas G Altman. Empirical evidence of bias: dimensions of methodological quality associated with estimates of treatment effects in controlled trials. *Jama*, 273(5):408–412, 1995.

[307] Clayton Scott and Robert D Nowak. Minimax-optimal classification with dyadic decision trees. *IEEE transactions on information theory*, 52(4):1335–1353, 2006.

[308] Steven P. Sedlis, Pamela M. Hartigan, Koon K. Teo, David J. Maron, John A. Spertus, G.B. John Mancini, William Kostuk, Bernard R. Chaitman, Daniel Berman, Jeffrey D. Lorin, Marcin Dada, William S. Weintraub, and William E. Boden. Effect of pci on long-term survival in patients with stable ischemic heart disease. *New England Journal of Medicine*, 373(20):1937–1946, 2015. PMID: 26559572.

[309] SEER. Female breast cancer — cancer stat facts. `https://seer.cancer.gov/statfacts/html/breast.html`, 2020. (Accessed on 03/12/2021).

[310] Mark Senn. *Road Traffic Act 1930*, 1930 (accessed October 23, 2020).

[311] Patrick W. Serruys, Marie-Claude Morice, A. Pieter Kappetein, Antonio Colombo, David R. Holmes, Michael J. Mack, Elisabeth Ståhle, Ted E. Feldman, Marcel van den

Brand, Eric J. Bass, Nic Van Dyck, Katrin Leadley, Keith D. Dawkins, and Friedrich W. Mohr. Percutaneous coronary intervention versus coronary-artery bypass grafting for severe coronary artery disease. *New England Journal of Medicine*, 360(10):961–972, 2009. PMID: 19228612.

[312] Sudha Seshadri, Alexa Beiser, Margaret Kelly-Hayes, Carlos S Kase, Rhoda Au, William B Kannel, and Philip A Wolf. The lifetime risk of stroke: estimates from the framingham study. *Stroke*, 37(2):345–350, 2006.

[313] Tianze Shi and Zhiyuan Liu. Linking glove with word2vec. *arXiv preprint arXiv:1411.5595*, 2014.

[314] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5):1589–1604, Sept 2018.

[315] Boris Shor, Joseph Bafumi, Luke Keele, and David Park. A bayesian multilevel modeling approach to time-series cross-sectional data. *Political Analysis*, 15(2):165–181, 2007.

[316] Fiona M. Shrive, Heather Stuart, Hude Quan, and William A. Ghali. Dealing with missing data in a multi-question depression scale: A comparison of imputation methods. *BMC Medical Research Methodology*, 6(1):57, Dec 2006.

[317] Georgios Sianos, Marie-Angèle Morel, Arie-Pieter Kappetein, Marie-Claude Morice, Antonio Colombo, Keith D. Dawkins, Marcel van den Brand, N van Dyck, ME Russell, and Patrick W. Serruys. The syntax score: an angiographic tool grading the complexity of coronary artery disease. *EuroIntervention*, 1(2):219–227, 08 2005.

[318] Ohidul Siddiqui and Mirza W Ali. A comparison of the random-effects pattern mixture model with last-observation-carried-forward (locf) analysis in longitudinal clinical trials with dropouts. *Journal of biopharmaceutical statistics*, 8(4):545–563, 1998.

[319] Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. Cancer statistics, 2018. *CA: A Cancer Journal for Clinicians*, 68(1):7–30, 2018.

[320] Jatinder Singh, Ian Walden, Jon Crowcroft, and Jean Bacon. Responsibility & machine learning: Part of a process. *Available at SSRN 2860048*, 2016.

[321] Peter HA Sneath, Robert R Sokal, et al. *Numerical taxonomy. The principles and practice of numerical classification.* 1973.

[322] Leichombam Somorjit and Mridula Verma. Variants of generative adversarial networks for credit card fraud detection. In *International Conference on Computational Intelligence, Security and Internet of Things*, pages 133–143. Springer, 2020.

[323] Nguyen Hung Son. From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae*, 34(1, 2):145–174, 1998.

[324] Robert E. Stepp and Ryszard S. Michalski. Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence*, 28(1):43–69, 1986.

[325] Jonathan AC Sterne, Ian R White, John B Carlin, Michael Spratt, Patrick Royston, Michael G Kenward, Angela M Wood, and James R Carpenter. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338:b2393, 2009.

[326] J Stoehlmacher, DJ Park, W Zhang, D Yang, S Groshen, S Zahedy, and HJ Lenz. A multivariate analysis of genomic polymorphisms: prediction of clinical outcome to 5-fu/oxaliplatin combination chemotherapy in refractory colorectal cancer. *British journal of cancer*, 91(2):344, 2004.

[327] Karen K Stout, Curt J Daniels, Jamil A Aboulhosn, Biykem Bozkurt, Craig S Broberg, Jack M Colman, Stephen R Crumb, Joseph A Dearani, Stephanie Fuller, Michelle Gurvitz, et al. 2018 aha/acc guideline for the management of adults with congenital heart disease: a report of the american college of cardiology/american heart association task force on clinical practice guidelines. *Circulation*, pages CIR–0000000000000603, 2018.

[328] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. International Society for Optics and Photonics, 1993.

[329] Brian L Strom. Data validity issues in using claims data. *Pharmacoepidemiology and drug safety*, 10(5):389–392, 2001.

[330] Maxwell Taggart, Wendy W Chapman, Benjamin A Steinberg, Shane Ruckel, Arianna Pregenzer-Wenzler, Yishuai Du, Jeffrey Ferraro, Brian T Bucher, Donald M Lloyd-Jones, and Matthew T Rondina. Comparison of 2 natural language processing methods for identification of bleeding among critically ill patients. *JAMA network open*, 1(6):e183451–e183451, 2018.

[331] Murdoch TB and Detsky AS. The inevitable application of big data to health care. *JAMA*, 309(13):1351–1352, 2013.

[332] Terry M Therneau, Beth Atkinson, and Maintainer Brian Ripley. The rpart package, 2010.

[333] Terry M Therneau, Patricia M Grambsch, and Thomas R Fleming. Martingale-based residuals for survival models. *Biometrika*, 77(1):147–160, 1990.

[334] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57(4):884–893, April 2010.

[335] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560 – 567, 2012.

[336] Katherine L Tucker, James P Sheppard, Richard Stevens, Hayden B Bosworth, Alfred Bove, Emma P Bray, Kenneth Earle, Johnson George, Marshall Godwin, Beverly B Green, et al. Self-monitoring of blood pressure in hypertension: A systematic review and individual patient data meta-analysis. *PLoS medicine*, 14(9):e1002389, 2017.

[337] A Ultsch. Fundamental clustering problems suite (fcps). Technical report, University of Marburg, 2005.

[338] Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and LJ Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

[339] Stanislav Uryasev. Conditional value-at-risk: Optimization algorithms and applications. In *Proceedings of the IEEE/IAFE/INFORMS 2000 Conference on Computational Intelligence for Financial Engineering (CIFEr)(Cat. No. 00TH8520)*, pages 49–57. IEEE, 2000.

[340] Vanya Van Belle, Kristiaan Pelckmans, Sabine Van Huffel, and Johan AK Suykens. Support vector methods for survival analysis: a comparison between ranking and regression approaches. *Artificial intelligence in medicine*, 53(2):107–118, 2011.

[341] Stef van Buuren and Karin Groothuis-Oudshoorn. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software, Articles*, 45(3):1–67, 2011.

[342] Hendrika A van den Ham, Olaf H Klungel, Daniel E Singer, Hubert GM Leufkens, and Tjeerd P van Staa. Comparative performance of atria, chads2, and cha2ds2-vasc risk scores predicting stroke in patients with atrial fibrillation: results from a national primary care database. *Journal of the American College of Cardiology*, 66(17):1851–1859, 2015.

[343] Belén Vega-Márquez, Cristina Rubio-Escudero, José C Riquelme, and Isabel Nepomuceno-Chamorro. Creation of synthetic data with conditional generative adversarial networks. In *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, pages 231–240. Springer, 2019.

[344] Hélene Verhaeghe, Siegfried Nijssen, Gilles Pesant, Claude-Guy Quimper, and Pierre Schaus. Learning optimal decision trees using constraint programming. *Constraints*, pages 1–25, 2020.

[345] Sicco Verwer and Yingqian Zhang. Learning optimal classification trees using a binary linear program formulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1625–1632, 2019.

[346] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[347] Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018.

[348] Jason Wang, Luis Perez, et al. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 11, 2017.

[349] Joanna M Wardlaw, Miriam Brazzelli, Francesca M Chappell, Hector Miranda, Kirsten Shuler, Peter AG Sandercock, and Martin S Dennis. Abcd2 score and secondary stroke prevention: meta-analysis and effect per 1,000 patients triaged. *Neurology*, 85(4):373–380, 2015.

[350] James H. Ware, David Harrington, David J. Hunter, and Ralph B. D'Agostino. Missing data. *New England Journal of Medicine*, 367(14):1353–1354, 2012.

[351] Carole A. Warnes. Adult congenital heart disease: the challenges of a lifetime. *European Heart Journal*, 38(26):2041–2047, 07 2017.

[352] Jeremy S Whang, Stephen R Baker, Ronak Patel, Lyndon Luk, and Alejandro Castro III. The causes of medical malpractice suits against radiologists in the united states. *Radiology*, 266(2):548–554, 2013.

[353] Wilson. Estimation of cardiovascular risk in an individual patient without known cardiovascular disease. *UpToDate, Waltham, MA*, 2017.

[354] Peter W. F. Wilson, Ralph B. D'Agostino, Daniel Levy, Albert M. Belanger, Halit Silbershatz, and William B. Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

[355] Philip A Wolf, Ralph B D'Agostino, Albert J Belanger, and William B Kannel. Probability of stroke: a risk profile from the framingham study. *Stroke*, 22(3):312–318, 1991.

[356] Philip A Wolf, Ralph B D'Agostino, William B Kannel, Ruth Bonita, and Albert J Belanger. Cigarette smoking as a risk factor for stroke: the framingham study. *Jama*, 259(7):1025–1029, 1988.

[357] Angela M Wood, Ian R White, and Simon G Thompson. Are missing outcome data adequately handled? A review of published randomized controlled trials in major medical journals. *Clinical Trials*, 1(4):368–376, 2004. PMID: 16279275.

[358] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[359] Junjie Wu, Hui Xiong, and Jian Chen. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 877–886. ACM, 2009.

[360] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, and Yang Xiang. Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association*, 27(3):457–470, 2020.

[361] Adam Yala, Constance Lehman, Tal Schuster, Tally Portnoi, and Regina Barzilay. A deep learning mammography-based model for improved breast cancer risk prediction. *Radiology*, 292(1):60–66, 2019.

[362] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.

[363] Yuancheng Ye, Lijuan Wang, Yue Wu, Yinpeng Chen, Yingli Tian, Zicheng Liu, and Zhengyou Zhang. Gan quality index (gqi) by gan-induced classifier, 2018.

[364] I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.

[365] I-Cheng Yeh. Modeling slump flow of concrete using second-order regressions and artificial neural networks. *Cement and Concrete Composites*, 29(6):474–480, 2007.

[366] I-Cheng Yeh and Tzu-Kuang Hsu. Building real estate valuation models with comparative approach through case-based reasoning. *Applied Soft Computing*, 65, 04 2018.

[367] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[368] David M Yousem and Robert I Grossman. *Neuroradiology: the requisites*. Elsevier Health Sciences, 2010.

[369] John Zech, Margaret Pain, Joseph Titano, Marcus Badgeley, Javin Schefflein, Andres Su, Anthony Costa, Joshua Bederson, Joseph Lehar, and Eric Karl Oermann. Natural language–based machine learning models for the annotation of clinical radiology reports. *Radiology*, 287(2):570–580, 2018.

[370] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2017.

[371] Zhongheng Zhang. Multiple imputation for time series data with Amelia package. *Annals of Translational Medicine*, 4(3), 2016.

[372] Jing Zhao and Aron Henriksson. Learning temporal weights of clinical events using variable importance. *BMC medical informatics and decision making*, 16(2):71, 2016.

[373] F. Zhou, Q. Claire, and R. D. King. Predicting the geographical origin of music. In *2014 IEEE International Conference on Data Mining*, pages 1115–1120, Dec 2014.

[374] Yan Zhou and John J McArdle. Rationale and applications of survival tree and survival ensemble methods. *psychometrika*, 80(3):811–833, 2015.

[375] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *International conference on algorithmic applications in management*, pages 337–348. Springer, 2008.

Acronyms

k-NN k-Nearest Neighbors.

AHA American Heart Association.

AHT Anti-Hypertensive.

AI Artificial Intelligence.

ASA aspirin.

AUC Area Under the ROC Curve.

BCD Block Coordinate Descent.

BMC Boston Medical Center.

BMI Body Mass Index.

BOW Bag of Words.

BP Blood Pressure.

CABG Coronary Artery Bypass Graft.

CAD Coronary Artery Disease.

CART Classification and Regression Trees.

CD Coordinate Descent.

CHD Coronary Heart Disease.

CT Computed Tomography.

CUBT Clustering sing Unsupervised Binary Trees.

CVaR Conditional-Value-at-Risk.

DBP Diastolic Blood Pressure.

DFCI Dana Farber Cancer Institute.

ECG Electrocardiogram.

EHR Electronic Health Records.

FCPS Fundamental Clustering Problems Suite.

FHS Framingham Heart Study.

FNA Fine Needle Aspirate.

FSRS Framingham Heart Study Stroke Risk Score.

GBT Gradient Boosted Trees.

GloVe Global Vectors for Word Representation.

HDL High-Density Lipoprotein.

IB Integrated Brier.

IBR Integrated Brier Score.

ICOT Interpretable Clustering via Optimal Trees.

LDL Low-Density Lipoprotein.

LP Linear Programming.

LVH Left Ventricular Hypertrophy.

MAE Mean Absolute Error.

MCAR Missing Completely At Random.

MIO Mixed Integer Optimization.

ML Machine Learning.

MNAR Missing Not At Random.

MRI Magnetic Resonance Imaging.

NLP Natural Language Processing.

NN Neural Networks.

N-SRS Non-Linear Stroke Risk Score.

OCT Optimal Classification Trees.

OPP Observations Per Patient.

ORC Operations Research Center.

ORT Optimal Regression Trees.

OS overall survival.

OST Optimal Survival Trees.

PCA Principle Component Analysis.

PCI Percutaneous Coronary Intervention.

PPMI Parkinson's Progression Markers Initiative.

R-FSRS Revised Framingham Stroke Risk Score.

RF Random Forest.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Networks.

SBP Systolic Blood Pressure.

SVM Support Vector Machines.

T2DM Type 2 Diabetes Mellitus.

TAE Time from diagnosis to a potential Adverse Event.

TF-IDF Term Frequency-Inverse Document Frequency.

VaR Value-at-Risk.