A Machine Learning Automation System for Utilization Management

by

Rohil Verma

B.S. Computer Science and Engineering
Massachusetts Institute of Technology, 2020

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2020

Signature of Author: _____

Department of Electrical Engineering and Computer Science
August 14, 2020

Certified by: _____

Robert M. Freund
Professor of Operations Research
Thesis Supervisor

Accepted by: _____

Katrina LaCurts
Chair, Master of Engineering Thesis Committee

A Machine Learning Automation System for Utilization Management

by

Rohil Verma

Submitted to the Department of Electrical Engineering and Computer Science

August 14, 2020

In partial fulfillment of the requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract: We develop high-performance machine learning automation systems for utilization management that are effective across all specialties. We were motivated by the knowledge that current automation systems for utilization management are rules-based and focused on narrow subsets of healthcare specialties. We develop models that can automate nearly 90% of a utilization management team's workload with less than 1% error. We evaluate these models on both historical data and as part of a live system in industry. The performance and efficacy of our models are consistent across both evaluation domains, demonstrating the generalizability of our work.

Thesis Supervisor: Robert M. Freund
Theresa Seley Professor in Management Science, Sloan School of Management

Acknowledgements

"If I have seen further, it is by standing on the shoulders of giants" - Isaac Newton

I have been extremely blessed through my time at MIT to have the support of many brilliant, kind people. It is impossible to list all of them; but a few stand out.

My advisor, Rob, whose insight and breadth of knowledge constantly pushed me to learn more and work harder. More than once, when I was lost for ideas, Rob was able to right the course. Without his expertise, support and thoughtfulness, this project would have been fruitless.

My mentor, Kang, whose optimism and commitment have been instrumental in the success of this project. Kang's commitment to principled innovation has been a constant source of energy for me, and without his aid, this project would never have got off the ground.

My family, Rajiv, Anjana, Shivali and Margaret, who have always stood by my side, and to whom I owe everything. They have supported me in innumerable ways and have always been the source of my strength.

This project would never have reached this stage if not for these people. Thank you.

Contents

List of Figures

List of Tables

Overview

Computer science and healthcare have long been intertwined. Students, professionals, and academics have always sought practical avenues to apply their talents, and there are few nobler professions to aid than healthcare. Unfortunately, it has not been easy to propagate modern technology into modern healthcare. The healthcare industry is large, complex, highly regulated and decentralized. In such a system, effecting change is challenging, but it **can** be done.

Furthermore, particularly in healthcare, many important problems are unknown to all but those that are deeply affected by them. Discovering and solving these problems often has a tremendous impact on their stakeholders, making people's lives easier, more productive, and more enjoyable.

One such problem is the efficient execution of the practices surrounding utilization management. Utilization management is a critical operation that every healthcare provider in the country is required to perform. It involves ensuring that the procedures a doctor recommends are appropriate, cost-effective, and easily available for patients. Each procedure is evaluated as a *referral*, which stores all the information the recommending doctor deems relevant. Utilization management, through the evaluation of these referrals, costs payers and providers billions of dollars and thousands of hours each year. [1] Insurance companies dedicate large teams to this task, and healthcare providers often retain teams of physicians and medical staff to interface with them. The efforts of these teams have a very tangible impact on the quality of care that their patients receive.

Today, the utilization management process is highly manual, and requires significant investments of time, effort, and money. In recent years, rule-based systems have been built to automate this task, but they are not widespread, and they do not address

a sufficient proportion of the problem. More than 60% of the referrals a provider processes in the utilization management process cannot be automated through rule-based systems due to the complexity in enumerating these rules. As a result, the average physician still spends almost 40% of their time working on utilization management related tasks. [1] Moreover, the process is slow, and patients are often unable to receive their recommended treatment due to delays or associated issues.

In an attempt to improve the current UM process, we set a goal to use modern machine learning techniques to learn from utilization management data and provide a superior automation tool. We managed to improve performance significantly over rule-based systems and were able to demonstrate the performance of our experimental models in practice through a system that has been functional at a large healthcare provider in the US for over a year.

The rest of this thesis has been split into four chapters. The first describes the utilization management process in more detail, shares primary data and reveals the key challenges. The second describes related work and the third describes our methods. Finally, the fourth details our results, both on historical data and in practical use.

1 Introduction to Utilization Management

1.1 Objectives

Healthcare services cost the world roughly 8 trillion USD in 2017, a figure which was estimated to rise to over 10 trillion USD by 2022 [16], pre-coronavirus. The US government alone spent over 3.5 trillion USD in 2017, almost 20% of its GDP [29]. These costs are projected to grow higher and higher as more and more of the world's population gains access to high-quality medical care.

For many medical providers, utilization management serves as a cost-containment strategy [5]. As provider costs rise, almost all major providers have dedicated utilization management teams that review procedures for appropriateness. According to the Utilization Review Accreditation Commission (URAC), a Washington DC non-profit, utilization management is "the evaluation of the medical necessity, appropriateness, and efficiency of the use of health care services, procedures, and facilities under the provisions of the applicable health benefits plan, sometimes called 'utilization review'." [12] There are three main types of utilization management: *prospective review, concurrent review,* and *retrospective review.* [4]

Prospective review is conducted prior to a medical procedure being performed, *concurrent review* is conducted during the course of care, and *retrospective review* is conducted after the procedure has been performed. All three review types share the goal of measuring the appropriateness of suggested medical care.

When utilization management is done well, patients typically enjoy higher quality of care and more appropriate medical procedures. Simultaneously, medical providers avoid unnecessary procedures, and insurers avoid unnecessary costs. When UM is conducted poorly, it can hurt patient outcomes, create financial risks and lead to litigation [22].

1.2 Existing Processes

Utilization management (UM) is typically conducted by utilization management reviewers, who are led by a UM team leader. Reviewers are trained to perform UM duties, and may be solely devoted to performing UM duties, or be involved in other administrative or medical duties. For example, UM teams commonly involve some number of nurses who can provide additional input on referral appropriateness and can offer medical knowledge. Similarly, there is a smaller number of doctors who are called on to offer their input when needed. This triaged process is quite costly in both time and resources.

The primary duty of a UM team is to evaluate a recommended procedure and decide whether it should be *approved* or *denied*. We focus on the *prospective review* process, where a procedure is typically a physician referral. *Approval* indicates that the patient will undergo the procedure, while *denial* indicates that they will not receive medical care in its currently recommended form. Denials do not disqualify patients from receiving care in the future and are uncommon – over 90% of referrals are typically approved. In some cases, a referral may be *cancelled*, such as in the case of duplicate requests, or insurance incompatibility. Referrals may also undergo a lengthy appeals process.

Referrals contain a lot of multidisciplinary data, bringing together patient demographic data, medical history, insurance data, and provider data into a single referral. They also include referral-specific data and clinical notes. UM reviewers are responsible for reviewing this data and typically briefly study the clinical notes which can span a single line or multiple pages.

Medical providers employing UM teams have typically employed entirely manual review processes, which are lengthy and often delay patient care. In extreme cases, a delayed referral that is ultimately denied would severely impact a patient's outcomes.

With the advancement of technology, many providers have begun to employ computerized rule-based systems to automate the review process for simpler referrals. This reduced referral turnaround time from days to mere seconds for common referrals that could be easily approved or denied, greatly improving the speed and quality of the average patient's care. Today, some medical providers are able to review between 30-40% of their prospective review referrals through such rules.

1.2.1 Referral Life Cycle

To walk through the complete life cycle of a referral today, imagine that your doctor recommends that you undergo surgery to repair your broken foot. The doctor fills out a referral request which is sent to the UM team. A first-level review is then conducted either by a rule-based system, or by a member of the UM team, who is typically a nurse-level professional. This nurse is supported in their decision-making process by a team member typically at the physician level. [32]

The vast majority of referrals end their lives here - they are approved through manually or through simple rules and the patient is notified about scheduling options for the procedure. However, if a referral is denied, the patient or doctor typically appeals, and explains why the procedure is necessary. Patients are entitled to both an internal second-level review, and an external third-level review by an external organization. [32] In practice, the doctor and patient are sometimes unable to pursue their first choice treatment due to the UM process, instead compelled to choose an alternate one. [1]

1.3 Challenges of Utilization Management

Utilization management in the United States today is expensive, labor-intensive, and time-consuming. There are over 1.5 billion referrals processed each year, costing organizations between 22 and 30 billion USD each year. The average American physician refers over 1600 patients per year [1], and the average cost to process a single referral varies between $15-$20 depending on the size and efficiencies of the processor organization.

The challenges of utilization management do not end at costs. For healthcare providers, utilization management often necessitates huge paper trails, logistical challenges, and delays in care provision. According to a survey of doctors by the American Medical Association, physicians and staff spend almost two full business days per week handling prospective referrals. [1]

Patients may have to wait days or even weeks to determine whether they can undergo a treatment with a particular doctor. In some circumstances, a patient's prior procedure may be deemed "medically unnecessary", [17] [2] leading to heavy financial burdens on patients. Physicians estimate that 75% of patients do not undergo the recommended form of treatment due to delays or other issues with the referral process. Costs, communication overhead, and delayed decision-making necessitate a better framework for utilization management. [1]

Finally, the challenges listed above are for systems that already employ simple rule-based computer algorithms for UM automation. Rule-based systems cannot be practically extended to the breadth of healthcare specialties, as we will see in the related work. That means that future systems need to seek new avenues for improvement, because these traditional rule-based automation systems have reached their limits.

1.4 Pros and Cons of an Algorithmic Solution / Decision-Making Guiding Principles

An algorithmic solution naturally possesses both advantages and disadvantages. In this section we discuss some of the most prominent of these. Our decisions on questions such as which features to select or which algorithms to use were guided by our answers to these questions.

A programmatic, non-rules-based solution to the challenge of UM offers the following advantages:

- it would be faster, reducing referral turnaround time for the patient and medical staff;
- it would be scalable and general-purpose, applicable to all healthcare specialties, rather than focused on a single, narrow domain;
- it would be less costly for healthcare providers, lowering administrative costs;
- it would reduce the time investment required from doctors and nurses;
- and it would not be subject to overt bias, providing more consistent decision-making to patients and medical staff.

A programmatic solution will remove incentives to deny procedures purely based on cost or profit, and focus on evaluating the medical appropriateness of care. Finally, the use of these solutions will raise questions of fairness and discrimination when it comes to approving or denying care - and such scrutiny can only raise standards across the country.

However, there are significant challenges to the construction of such a system as well:

- UM spans all areas of healthcare, from physical therapy, to ophthalmology, to cancer treatment. How can a single, general-purpose model be developed?

- UM aggregates a lot of data from medical providers, insurance providers and patient demographics. Furthermore, some data is structured, and some is unstructured. What are the right features to use, and what transformations are necessary?

- UM deals with confidential protected health information (PHI). How can systems balance privacy with performance?

- UM systems must gain and maintain the trust of patients, healthcare providers and insurance providers. How can they be made interpretable? How will systems balance interpretability with performance?

- UM automation systems, particularly data-driven ones will be learning from past actions. How can they be made robust to biased data so that they do not learn discriminatory processes?

It is critical to keep in mind both the advantages and disadvantages of innovative systems in the context of utilization management in order to develop systems that can be widely utilized. We have touched upon the major ones here, and will detail how they have affected our choices in the rest of the thesis.

2 Related Work

2.1 Machine Learning in Healthcare

Scientists and physicians have long worked together to bring the latest advances in mathematics and computing to improve healthcare processes. As far back as the late 1980s, regression analyses were used to generate TRISS scores - used to evaluate trauma care at hospitals around the country. [8] In the 2000s, scientists and physicians developed new statistical methods of mortality risk assessment for patients suffering from colonic peritonitis [6]. Today, computer scientists are trying to help solve the next generation of healthcare challenges using advanced machine learning techniques.

With the advancement of machine learning, the intersections between computer science and healthcare have grown exponentially. Today, the research frontiers of machine learning in healthcare include screening patients for breast cancer [27], identifying patients at risk of sepsis [3], and even ascertaining patient symptoms from doctor's notes and electronic medical records. [19]

2.2 Computing in Utilization Management

Despite this progress, the last academic paper referencing the use of computerized systems for utilization management dates back to 1994. [23] In this article, Nelson et al. designed, implemented and evaluated a tool for screening adult patients for inappropriate days of care. The article described an approach towards retrospectively evaluating medical care, and used a rule engine based on the Appropriateness Evaluation Protocol (AEP).

Does the lack of directly related work indicate an absence of importance or interest in the subject? On the contrary, as we see below, it reveals the challenges associated with developing a general-purpose model, and the need for advances beyond rule-engines.

For example, in 2005 Sun and Chang described a rule-engine approach to determining the medical appropriateness of prescribing antibiotics [31]. In 2010 Vartanians et al. described the effects of employing the American College of Radiology's rule-based imaging (CT, MR, nuclear medicine exams) appropriateness criteria [33]. And as recently as 2019, Quintens et al described the development of "Check of Medication Appropriateness," a rule-based screening procedure for medication appropriateness. [26]

Every single one of these papers had the goal of verifying medical appropriateness, which is exactly the goal utilization management is meant to fulfill. They all used rule-based systems, and they were all focused on a single problem - antibiotic appropriateness, radiological test appropriateness, or medication appropriateness. Given that these works are still using rule-engines on small, focused areas, it is clear that it is time to try something new and more sophisticated. Industry efforts are not much farther along, and we detail these below.

Carolinas HealthCare, a healthcare provider based in North Carolina, South Carolina and Georgia, uses a machine learning model to automatically approve referrals. The model achieves 99% accuracy with its predictions, but is applicable on less than 10% of its total referrals. [24]

Optum, one of the largest healthcare organizations in the US, has developed a product called Optum360, which they state uses AI to stratify referrals of differing complexities. They also claim to use NLP to support their recommendations. They do not provide any public metrics. [11]

EXL is a software consultancy that describes their use of rule engines for simple referral approvals. They also describe the use of NLP for OCR-style field data extractions. They do not provide any metrics, and appear to be proposing older, simpler rule-based systems. [28]

Public industry efforts are primarily rule-based or simplistic, non-machine learning NLP systems. The machine learning systems that exist are narrow and do not extend to a significant proportion of the data. There is a clear opportunity for improvement.

Having explored the state-of-the-art related work in academia and industry, we shall now present the data available, discuss how we chose our features and algorithms, and explain how we evaluate our performance.

3 Methods

3.1 Utilization Management Problem Setting

The dataset we have constructed consists of prospective review physician referrals. The different kinds of data available as part of a referral comprise two major types: patient-specific data, and referral-specific data.

3.1.1 Data Types

Patient data comprises:

- Demographic data: The patient's name, age, gender, race, ethnicity, spoken languages, etc.
- Insurance data: The insurance carrier, line of business, health plan, etc.
- Medical data: The patient's PCP, medical history, etc.

Referral-specific data comprises:

- Provider data: doctor names, specialties, insurance statuses
- Diagnosis data: diagnosis codes, procedure codes, procedure quantities, clinical notes

3.1.2 Sample Referral

A typical referral may look like the following example. Here, our patient has fractured his leg so has gone to visit his primary care provider (PCP), a general practitioner, who has referred him to a second doctor who specializes in orthopaedic surgery. Both the referring doctor and the referred-to doctor could be the same person, although not so in this example. Note also that while there is always at least one *primary* procedure and diagnosis code, in about 30% of cases there are additional codes. Procedure codes

are typically accompanied by a procedure quantity and a procedure modifier, each of which convey a specific medical meaning to a UM reviewer.

- **Patient Name, DOB, Gender:** Amanda Smith, 05/07/1965, F
- **Patient Insurance Carrier, Plan, Line of Business:** Humana, Humana Standard Plan, Seniors
- **Referring Doctor's Name and Practice Location:** Dr. John Adams, 888 Commonwealth Ave, Boston MA
- **Referring Doctor's Specialty:** General Practice
- **Referring Doctor's Insurance Status:** Contracted

- **Referred-to Doctor's Name and Practice Location:** Dr. Jolene Williams, 890 Commonwealth Ave, Boston MA
- **Referred-to Doctor's Specialty:** Surgery-Ortho
- **Referred-to Doctor's Insurance Status:** Contracted
- **Diagnosis Codes (Primary in bold): R41.82**
- **Procedure Codes (Primary in bold): 95043**
- **Procedure Quantity:** 1
- **Referring doctor's clinical notes**

There is a wealth of information within this data. The majority of the data is structured, categorical data, along with a few non-categorical structured fields, such as the patient's age or procedure quantity. The key source of unstructured data is the

Figure 1: Splitting a year of data into training, validation and testing sets.

doctor's clinical notes.

Our target variable is a *referral status* variable. This is also categorical, and takes values in *approved*, *denied*, and *cancelled*. While approved referrals and denied referrals typically look fairly different, cancelled referrals are far noisier in that there is no consistent methodology for cancelling a referral that can be observed purely from a referral itself. This is because a referral may be cancelled due to duplication, scheduling issues, or a simple mistake during data entry. It is very challenging to attempt to identify these referrals purely from the data we have.

3.1.3 Data Duration and Splits

We have 3 years of such data from a Los Angeles-based medical group, comprising almost 3 million physician referrals and their approval status. The data is not all drawn from the same distribution – it experiences *dataset shift*. [25] This is because the protocols with which a medical provider's UM team reviews referrals change over time. As such, we are careful to test models on time periods with a consistent review protocol, i.e., no dataset shift.

We split our data into training, validation and testing sets as seen in Figure 1. In order to simulate how our models would be used in production, we hold out the final month of data as our testing set, the penultimate month as our validation set, and all previous data without dataset shift as our training set.

3.2 Features

To determine which features to select, we returned to our decision-making criteria. We selected all the features described in the sample referral, with the following modifications:

- we excluded clinical text due to the presence of PHI and the complexity associated with accessing features in an anonymized fashion;
- we replaced date of birth (PHI) with the non-PHI age;
- we replaced PHI names with anonymized identifiers; and
- we excluded office addresses, patient name and ethnicity to avoid the presence of PHI or overtly discriminatory features;

Our goal was to select as many features as possible to improve overall performance and to build a single, general-purpose model. We avoided features that could bias the model towards discriminatory outcomes or that would necessitate the use of PHI or complex anonymization processes. Anonymization / de-identification processes or algorithms are rarely perfect and even anonymized data is not secure - it is possible to re-identify anonymized data. [14] With these constraints, we selected all possible features.

3.3 Algorithms

While considering which algorithms to train on our features, we evaluated using both out-of-the-box models, such as in Python's *scikit-learn*, as well as writing our own custom models from scratch. Our primary goal alongside performance was interpretability, as these models will not be used in a vacuum, but rather as a piece of a system with many parts. They will likely be called upon to explain why a particular decision was made or evaluated to ensure PHI or overtly discriminatory criteria were not used to make decisions.

The three algorithms that we settled upon were Logistic Regression, Decision Trees and Random Forests. These algorithms, with the exception of Random Forests, are highly interpretable. Relative to the others, Random Forests trade-off some interpretability for predictive power. We chose not to use Neural Networks for reasons of interpretability - it quickly becomes far too difficult to evaluate the decision-making process, making it unsuitable for our aims. Furthermore, given the relatively low dimensionality of our data, we doubted that the additional predictive power would be utilized.

Now, we will briefly describe the algorithms, the parameters we chose to tune, and our training procedure.

3.3.1 Logistic Regression

The Logistic Regression model is a binary classification model used to model the probability of an event occurring, for example, that the next insect you see is a bee, or that the next referral your UM team sees is approved. It has a long history stretching back to the 19th century [13], but is still extremely popular, and notable for a few distinctive properties, which we describe below. See Figure 2 for an example.

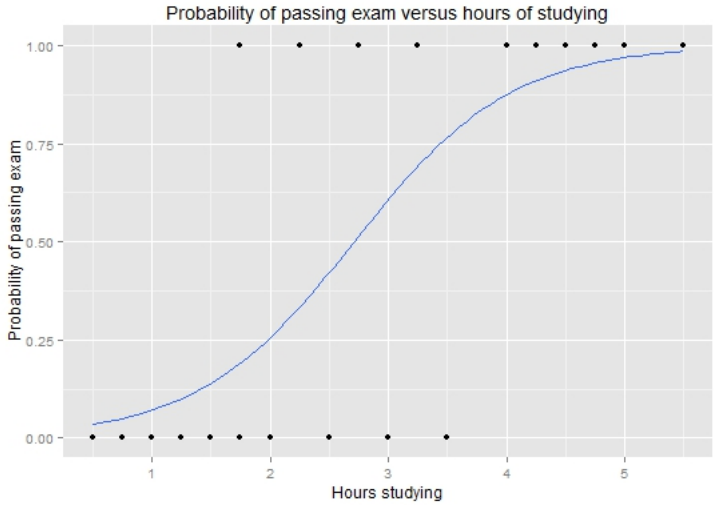

Figure 2: A Logistic Regression model depicting the probability of passing an exam versus hours of studying [18]

The Logistic Regression model is a linear model, and this property is both a weakness and a strength. While it indicates that the logistic regression model is only capable of identifying linear relationships within data, at the same time this property makes the model very easily interpretable. The higher a weight it assigns to a particular variable, the higher that variable's importance is to the model. Particularly in the field of healthcare, where the explainability of a decision is of utmost importance, interpretability is a critical quality.

We used the one-vs-all extension [7] to Logistic Regression as we have 3 classes (approved, denied, cancelled). We optimized the Logistic Regression loss model using stochastic gradient descent (SGD). [30] We tune the regularization penalty, the regularization type (L1 and L2) and class weights. [20] We encoded our input features using a one-hot encoding.

3.3.2 Decision Trees

Decision trees [10] are another extremely popular classification model. They identify relationships in a dataset using a decision tree, as seen in the example in Figure 3.

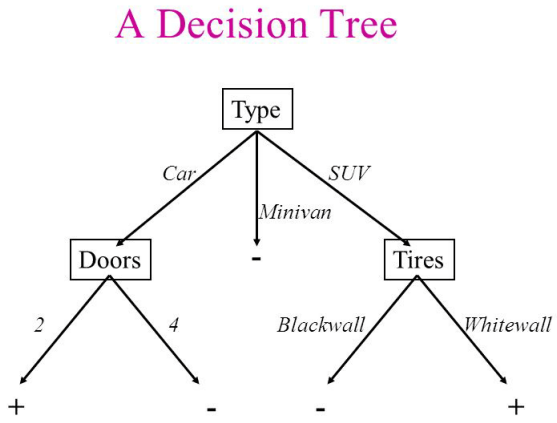

Figure 3: A simple decision tree representing the likelihood of purchasing a car based on its type, number of doors and tires. [15]

They use a statistical splitting algorithm to build the tree based on a specified metric such as Gini impurity, information gain, or variance reduction. They typically use a random subset of the features at each split.

Decision trees can fit to arbitrarily complex relationships in data by tuning their depth parameter, and are usually very interpretable due to their transparent decision-making process. Once a decision tree is constructed, it is straightforward for humans to follow the decision-making process that led to a prediction. In theory this is the best of both worlds, but increasing the depth to allow for more complex relationships typically makes it harder to comprehend the rationale behind a prediction.

We used the CART algorithm [10] to build our trees using Gini impurity as our purity metric. We tuned the maximum depth of the trees, varying their complexity, and the number of features considered at each split, varying the randomness at each split. We also varied the class weights.

3.3.3 Random Forests

Random Forests [21] [9] extend the ideas of decision trees. Random forests have been very successful in practice and typically outperform decision trees [34]. They involve training multiple decision trees and choosing the majority prediction. Each decision tree is trained from a random sample of the data, and typically use a random subset of the features at each split.

There are two properties that make random forests superior to decision trees. [34] First, by training many trees, each of which is trained on a different sample of the data, individual trees are able to focus on smaller subsets of the data. This improves the performance of the overall forest. Second, training many trees reduces the impact any individual tree has on the final prediction. This prevents the overfitting of any individual tree from dominating the prediction, improving the performance of the overall forest.

We used the CART algorithm to build our trees using Gini impurity as our purity metric. We tuned the maximum depth of the trees, varying their complexity, the number of trees trained, and the number of features considered at each split, varying the randomness at each split. We also varied the class weights.

3.3.4 Training Procedure

As described in the section on Data Duration and Splits, we use the final two months of data as the validation and testing sets, training our algorithms on all previous data. We also explore two alternative training variants:

- training models on all 3 classes
- training models on just approval and denied referrals

We consider the second approach because of the noisy nature of cancelled referrals. The idea was that ignoring the noisier data would allow the models to better hone in

on the approval-denial decision boundary. We do not evaluate it using accuracy, as it cannot even predict that a referral would be cancelled. We do, however, evaluate its denial false positive rate, which we describe below.

Approved	Denied	Cancelled
89%	6%	5%

Table 1: Table showing class percentages

3.4 Evaluation

We use multiple lines of thought to evaluate our performance. The baseline performance metric we use is the majority class occurrence percentage from the data. The most common class, an approved referral, occurs about 89% of the time. See Table 1 for details on the other classes. Accuracy above this baseline is good, and accuracy below is very bad. We compute accuracy using a threshold of 0.5.

The second metric of interest is the denial false positive rate at different approval thresholds. Here, we evaluate the following question: of the number of referrals that our model would approve, how many actually ought to be denied? The more referrals we can approve while remaining below a false positive rate of 1% the better.

We focus on the denial false positive rate for several reasons.

The first is due to the healthcare system. In the utilization management process, when a referral is approved, it is rarely re-evaluated. On the other hand, if it is denied, it goes through an appeals process. In our production system, for example, the existing UM team would be called upon to manually evaluate denials, **but not so for approvals,** which would go straight to scheduling. Thus, if we incorrectly say an approval should be denied, it will be manually reviewed and approved, and the only cost is the review cost. On the other hand if we say a denial should be approved, it will not be re-evaluated and the patient may undergo an expensive, unneccessary procedure. A referral that ought to be cancelled does not factor into our thinking, as referrals are typically cancelled for administrative reasons such as incorrect data entry, duplication, or scheduling challenges, which will always surface.

The second reason is due to the expectations of healthcare providers for a certain ceiling on false positive rate. There is little data available on how often utilization management teams make incorrect decisions, as counterfactuals are rarely available and there is wide variation. However, primary data and online materials such as those put out by Carolinas Healthcare [24] indicate a general consensus estimate of at least a 1% error rate. This becomes a reasonable false positive baseline for us. The more referrals we can approve while remaining below this false positive rate the better.

Thus, it becomes important to explore the denial false positive rates at different thresholds, as this explores the trade-off between acceptable error on the approvals, and the proportion of the data that the model is able to handle at a particular precision.

These performance metrics should hint at the learning that our methods, while well beyond state-of-the-art today, are not optimal solutions, and need improvements. It is extremely unlikely that machine learning models will completely replace UM teams. Instead, they will act as an evolution of rule-based systems today, taking on a larger proportion of referrals than rule-based systems alone, and working in concert with leaner, more efficient UM teams to handle the toughest decisions.

Hyperparameters	Values
Penalty type	L1, L2
Regularization coefficient	0.1, 0.01, 0.001, 0.0001, 1e-5, 1e-6, 1e-7
Class weights	Unweighted, Balanced

Table 2: Table showing logistic regression hyperparameters

4 Results

This section is divided into two parts: the first explores our experimental results on historical data; the second evaluates how well our models do in practice. Our practical results hold pretty closely to our experimental ones, which is promising; however, they also bring additional implementation details which must be understood. Thus, we have split this section into two parts.

4.1 Experimental Results

Accuracy for all models is computed using a decision threshold of 0.5.

4.1.1 Logistic Regression

The hyperparameters for our logistic regression are listed in Table 2. Considering 100% of the features at a split corresponds to choosing one of the optimal splits. The unweighted class weighting corresponds to all samples having an equal weight of 1, whereas the balanced class weighting corresponds to adjusting weights inversely proportional to class frequencies. Thus the combined weight of each class is the same.

Table 3 lists the accuracies obtained by a model trained on all 3 classes with a particular set of hyperparameters. Each set of hyperparameters was used to train 3 models, and we report both the mean and the standard deviation of the accuracies here. We chose to train 3 models for computational ease.

The best classifier by accuracy was the unweighted logistic regression with 1e-5 L1

Hyperparameters (penalty, coeff, weights)	**Mean Test Accuracy**	**Standard Deviation**
L1, 0.1, Unweighted	88.5%	0%
L1, 0.1, Balanced	88.5%	0%
L1, 0.01, Unweighted	88.5%	0%
L1, 0.01, Balanced	88.3%	0.1%
L1, 0.001, Unweighted	88.5%	0%
L1, 0.001, Balanced	85.7%	0.1%
L1, 0.0001, Unweighted	89.2%	0.05%
L1, 0.0001, Balanced	84.8%	0.2%
L1, 1e-5, Unweighted	89.5%	0.08%
L1, 1e-5, Balanced	84.1%	0.2%
L1, 1e-6, Unweighted	89.3%	0.04%
L1, 1e-6, Balanced	83.3%	0.1%
L1, 1e-7, Unweighted	88.5%	0.03%
L1, 1e-7, Balanced	81.4%	0.2%
L2, 0.1, Unweighted	88.5%	0%
L2, 0.1, Balanced	88.5%	0%
L2, 0.01, Unweighted	88.5%	0%
L2, 0.01, Balanced	88.2%	0.02%
L2, 0.001, Unweighted	88.6%	0%
L2, 0.001, Balanced	86.6%	0.03%
L2, 0.0001, Unweighted	89.2%	0.03%
L2, 0.0001, Balanced	85.9%	0.08%
L2, 1e-5, Unweighted	89.5%	0.06%
L2, 1e-5, Balanced	85.2%	0.7%
L2, 1e-6, Unweighted	89.2%	0.1%
L2, 1e-6, Balanced	82.3%	1.2%
L2, 1e-7, Unweighted	88.3%	0.3%
L2, 1e-7, Balanced	80.3%	1.4%

Table 3: Table showing logistic regression accuracy results

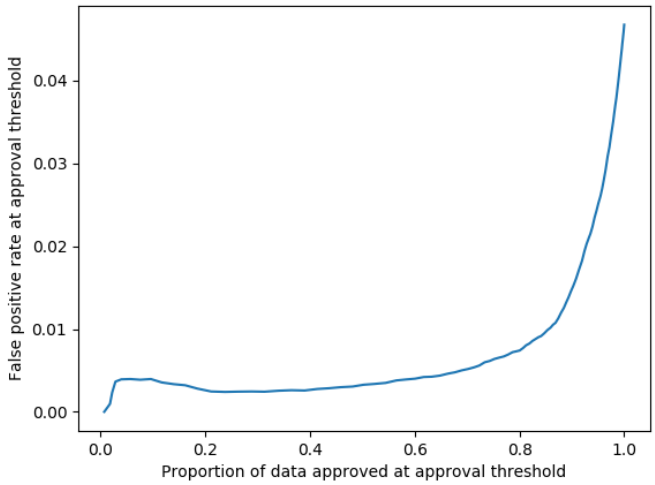

Figure 4: Best logistic regression denial false positive curve trained on 3 classes. Approves 86% of the data with a denial rate below 1%.

penalty. Decreasing the regularization parameter helped until 1e-5, after which the classifiers began to overfit.

The best denial false positive rate from classifiers trained on all 3 classes was created by the balanced logistic regression using L1 penalty and a coefficient of 1e-5. The denial false positive curve can be seen in Figure 4. The x-axis depicts the proportion of data that would be approved by the model based on the chosen approval threshold, and the y-axis depicts the denial false positive rate at this threshold. This best performing model was able to approve 86% of the data with a denial rate below 1%.

The best denial false positive rate across all our logistic regressions came from a logistic regression that was trained only on the approval and denial classes. It was also a balanced logistic regression using a coefficient of 1e-5, but the penalty in this case was L2. This logistic regression was able to handle 87% of the data with a denial false positive rate below 1%. The denial false positive curve can be seen in Figure 5.

The remaining curves can be found in the appendix.

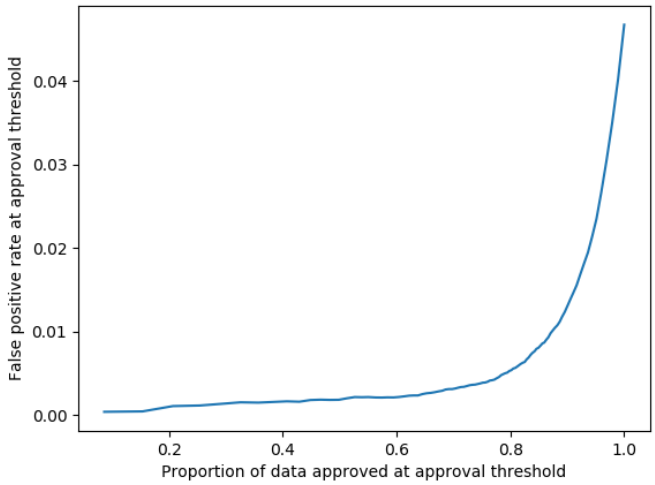

Figure 5: Best logistic regression denial false positive curve trained on 2 classes. Approves 87% of the data with a denial rate below 1%.

Hyperparameters	**Values**
Max Depth	5, 10, 20, 30
Percentage of features evaluated at splits	30%, 50%, 70%, 100%
Class weights	Unweighted, Balanced

Table 4: Table showing decision tree hyperparameters

4.1.2 Decision Trees

The hyperparameters for our decision trees are listed in Table 4.

Table 5 lists the accuracies obtained by a model trained on all 3 classes with a particular set of hyperparameters. Each set of hyperparameters was used to train 3 models, and we report both the mean and the standard deviation of the accuracies here. We chose to train 3 models for computational ease.

Balancing the classes consistently led to worse accuracies. Increasing the depth helped until a depth of 10, after which the unweighted trees began to overfit.

The best denial false positive rate from trees trained on all 3 classes was created

Hyperparameters (depth, pct, weights)	**Mean Test Accuracy**	**Standard Deviation**
5, 30%, Unweighted	88.6%	0.02%
5, 30%, Balanced	44%	6%
5, 50%, Unweighted	88.6%	0.04%
5, 50%, Balanced	40%	1.3%
5, 70%, Unweighted	88.6%	0.03%
5, 70%, Balanced	54%	1.4%
5, 100%, Unweighted	88.6%	0%
5, 100%, Balanced	53%	0%
10, 30%, Unweighted	88.8%	0.1%
10, 30%, Balanced	47.5%	4%
10, 50%, Unweighted	89.1%	0.1%
10, 50%, Balanced	54%	0.4%
10, 70%, Unweighted	89.2%	0.1%
10, 70%, Balanced	59%	1.9%
10, 100%, Unweighted	89.4%	0%
10, 100%, Balanced	59%	0%
20, 30%, Unweighted	88.2%	0.2%
20, 30%, Balanced	66%	0.3%
20, 50%, Unweighted	88.2%	0.2%
20, 50%, Balanced	67%	1%
20, 70%, Unweighted	88.2%	0.1%
20, 70%, Balanced	68%	0.02%
20, 100%, Unweighted	88.2%	0.1%
20, 100%, Balanced	67%	0.02%
30, 30%, Unweighted	85%	0.1%
30, 30%, Balanced	80%	0.5%
30, 50%, Unweighted	85.1%	0.07%
30, 50%, Balanced	80.2%	0.6%
30, 70%, Unweighted	85%	0.09%
30, 70%, Balanced	80.8%	0.4%
30, 100%, Unweighted	84.6%	0.03%
30, 100%, Balanced	80.8%	0.04%

Table 5: Table showing decision tree accuracy results

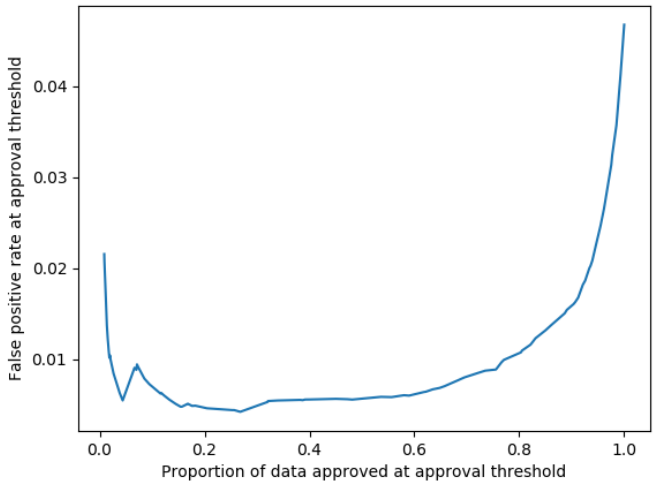

Figure 6: Best decision tree denial false positive curve trained on 3 classes. Handles 77% of the data with a denial false positive rate below 1%.

by the balanced decision tree with depth 10, looking at 100% of the available features. This tree also shows the best mean accuracy in Table 5. The denial false positive curve can be seen in Figure 6. This best performing model was able to handle 77% of the data with a denial false positive rate below 1%.

But the best denial false positive rate across all our decision trees again came from a tree that was trained only on the approval and denial classes. It was also the decision tree with depth 10, looking at 100% of the available features, however it was unweighted. Furthermore, this tree was able to handle 82% of the data with a denial false positive rate below 1%. The denial false positive curve can be seen in Figure 7.

The remaining curves can be found in the appendix.

4.1.3 Random Forests

The hyperparameters for our random forests are listed in Table 6. Recall that considering 100% of the features at a split corresponds to choosing one of the optimal splits. We do not report accuracies for balanced forests since our experiments with logistic

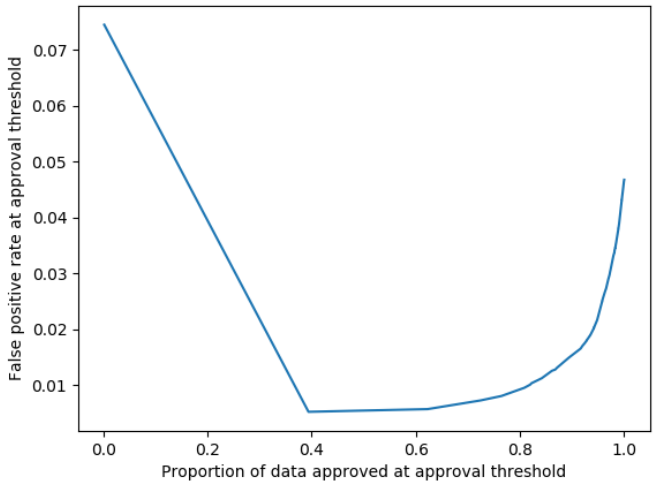

Figure 7: Best decision tree denial false positive curve trained on 2 classes. Handles 82% of the data with a denial false positive rate below 1%.

Hyperparameters	**Values**
Max Depth	5, 10, 20, 30
Percentage of features evaluated at splits	30%, 50%, 70%, 100%
Number of trees	10, 100, 200
Class weights	Unweighted, Balanced

Table 6: Table showing random forest hyperparameters

regression and decision trees have demonstrated that balancing classes does not lead to better accuracies, but we do consider them while evaluating the best denial false positive curves and these curves can be found in the appendix.

Table 7 lists the accuracies obtained by a model trained on all 3 classes with a particular set of hyperparameters.

Hyperparameters (depth, pct, number of trees)	**Test Accuracy**
5, 30%, 10	88.5%
5, 30%, 100	88.5%
5, 30%, 200	88.5%
5, 50%, 10	88.6%
5, 50%, 100	88.6%

5, 50%, 200	88.6%
5, 70%, 10	88.6%
5, 70%, 100	88.6%
5, 70%, 200	88.6%
5, 100%, 10	88.6%
5, 100%, 100	88.6%
5, 100%, 200	88.6%
10, 30%, 10	88.8%
10, 30%, 100	88.8%
10, 30%, 200	88.8%
10, 50%, 10	89.2%
10, 50%, 100	89.2%
10, 50%, 200	89.2%
10, 70%, 10	89.4%
10, 70%, 100	89.4%
10, 70%, 200	89.4%
10, 100%, 10	89.6%
10, 100%, 100	89.6%
10, 100%, 200	89.6%
20, 30%, 10	89.8%
20, 30%, 100	89.9%
20, 30%, 200	89.9%
20, 50%, 10	90%
20, 50%, 100	90.1%
20, 50%, 200	90.2%
20, 70%, 10	90%
20, 70%, 100	90.2%
20, 70%, 200	90.3%
20, 100%, 10	89.9%
20, 100%, 100	90.1%
20, 100%, 200	90.2%
30, 30%, 10	89.2%
30, 30%, 100	89.8%
30, 30%, 200	89.8%
30, 50%, 10	89.3%
30, 50%, 100	89.8%
30, 50%, 200	90%
30, 70%, 10	89.4%
30, 70%, 100	89.8%

30, 70%, 200	89.9%
30, 100%, 10	89.2%
30, 100%, 100	89.8%
30, 100%, 200	89.8%

Table 7: Table showing random forest accuracy results

Increasing the depth helped until a depth of 20, after which the unweighted forests began to overfit. The forest with the best accuracy had 200 trees, depth 20 and was trained on 70% of the available features.

The best denial false positive rate from forests trained on all 3 classes was created by the unweighted random forest with 200 trees, depth 20 and looking at 30% of the available features. The denial false positive curve can be seen in Figure 8. This best performing model was able to handle 84% of the data with a denial false positive rate below 1%.

The best denial false positive rate across all our random forests actually came once again from a forest that was trained only on the approval and denial classes. It was the unweighted random forest with 200 trees, depth 20, looking at 70% of the available features. This forest was able to handle 88% of the data with a denial false positive rate below 1%. The denial false positive curve can be seen in Figure 9.

The remaining curves can be found in the appendix.

4.2 Results from Industry

We used our models as part of a production machine learning service that made approval predictions on new referrals as they were entered into the system. We used the Flask web framework behind a Windows IIS server to expose our models through APIs. The models were exposed to our industry partner HealthFortis Associates' utilization

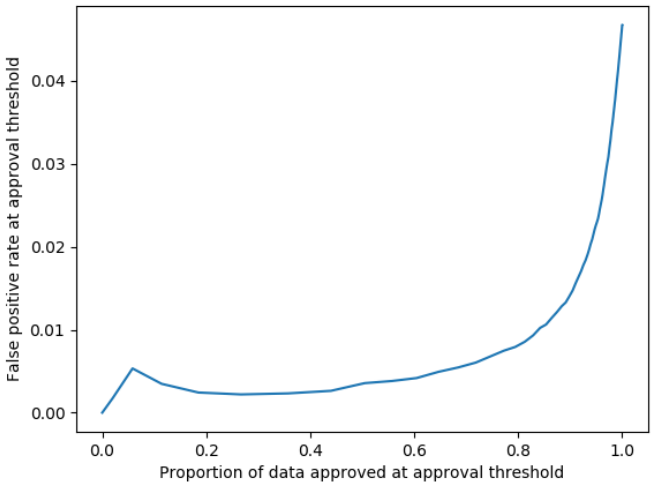

Figure 8: Best random forest denial false positive curve trained on 3 classes. Handles 84% of the data with a denial false positive rate below 1%.

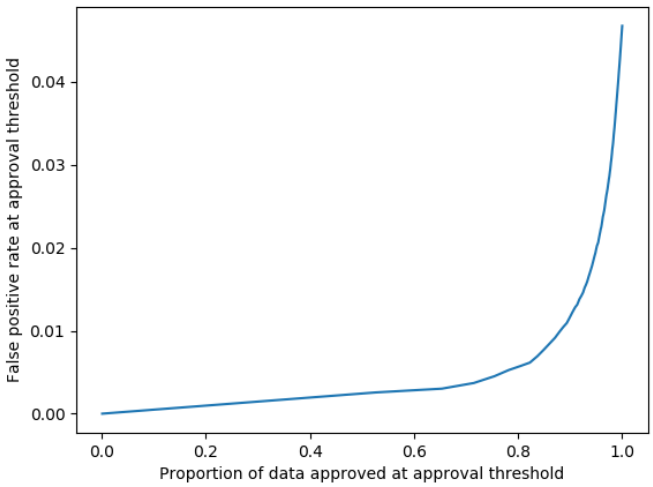

Figure 9: Best random forest denial false positive curve trained on 2 classes. Handles 88% of the data with a denial false positive rate below 1%.

management web portal.

Below, we detail our models' performance over a period of 14 months, from May of 2019 to July of 2020. The models were used as part of the UM process for a southern Californian healthcare provider. Models and thresholds were varied over time as we re-trained our models each month. We processed over 1.2 million referrals, approving close to 250,000 referrals.

Our model was rolled out in a phased manner. Initially, the threshold was set high such that the model would be more conservative in its predictions. The goal was that the model's false positive rate would remain below 1%. Later as the provider gained more confidence in the model, the threshold was lowered, allowing our model to make approval recommendations on a larger proportion of the data.

Thus, the model began by approving just 15% of referrals, with the threshold later lowered such that the model was approving approximately 25% of referrals. In practice, about 20% of referrals contained some attribute that was new to the model, i.e., a new feature value. As such, the model was really approving 25% of about 80% of the data, bringing its approval percentage closer to 20%. See Figure 10 for details. The A/B Test column corresponds to the referrals the model approved, the Pending column corresponds to the referrals the model denied (sent to the UM team), and the No Prediction column corresponds to referrals containing a new feature value.

Referrals that the model approved were put through an A/B test at random to allow us to collect data on the accuracy of the models predictions. Figure 11 shows that the denial rate indicated by this A/B test was 0.82%.

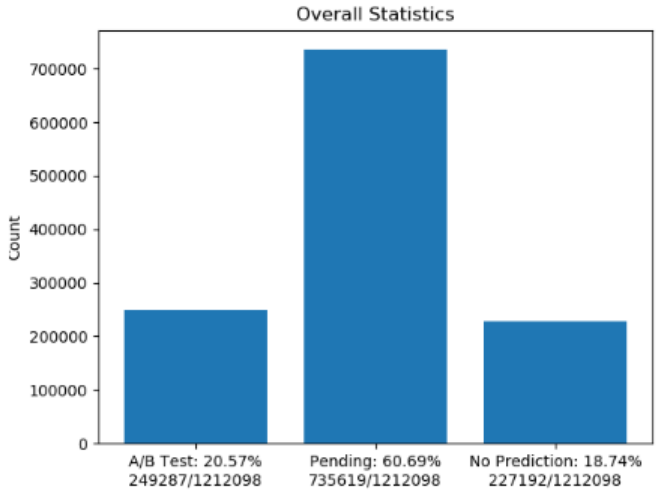

Figure 10: Overall statistics from the ML service dashboard. Credit: HealthFortis Associates

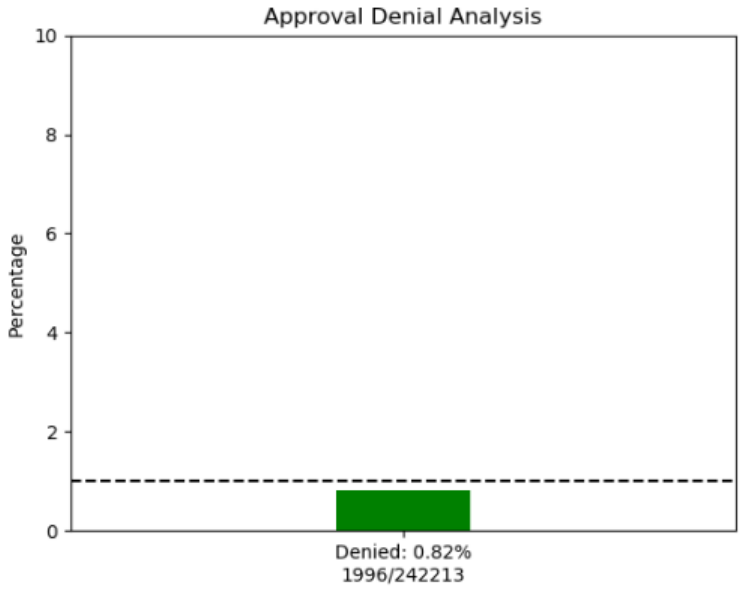

Figure 11: Denial rates from the ML service dashboard. Credit: HealthFortis Associates

Conclusion

Our goal in this thesis project was threefold:

- to understand the utilization management process,
- develop experimental machine learning models based on historical data, and
- evaluate them on historical data and as part of a real-time system.

Through background research and primary data on the utilization management process, we learned that approvals were rarely re-evaluated, denials were often appealed, and that cancels were noisy data. This allowed us to focus on the false positive denial rate - the number of referrals the model predicted should be approved, but actually ought to be denied. Our goal became to lower the approval threshold as far as possible while maintaining a ceiling on the false positive denial rate. The ceiling we chose was 1%.

To accomplish this, we experimented with three different learning algorithms: logistic regressions, decision trees, and random forests. We tuned the hyperparameters of these algorithms, experimented with class balancing, and explored the efficacy of filtering noisy data (cancelled referrals). The very best model we trained was able to approve 88% of the data with a false positive rate of less than 1%. It was an unweighted random forest trained on only the approval and denial classes and had 200 trees, each of which had a maximum depth of 20. It considered a randomly selected 70% of available features at each possible split.

To evaluate our models on real-time data, we used our best logistic regression model, which demonstrated slightly worse performance, but was significantly more interpretable and faster at making evaluations. This model was able to approve 87% of the data with a false positive rate of less than 1%. It was trained on just the approval and denial classes and the model was balanced. It also employed an L2 penalty with

a regularization parameter of 1e-5.

In practical use, our models never exceeded the 1% denial rate threshold and were used at an approval threshold where they automated approximately 20% of the provider's UM team workload. They were effective across all specialties and far exceeded the capabilities of rule-based systems.

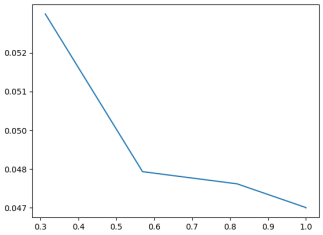

Figure 12: Curve for 3 class unweighted logistic regression with 0.1 l1 penalty

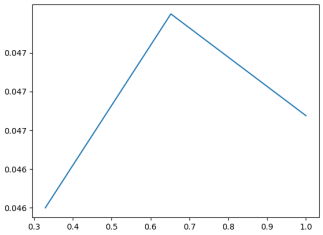

Figure 13: Curve for 3 class balanced logistic regression with 0.1 l1 penalty

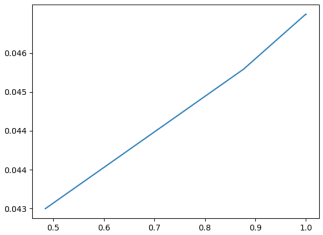

Figure 14: Curve for 3 class unweighted logistic regression with 0.01 l1 penalty

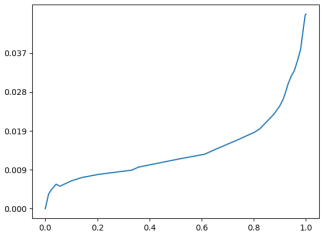

Figure 15: Curve for 3 class balanced logistic regression with 0.01 l1 penalty

Appendices

A Denial False Positive Curves

A.1 Logistic Regression

See Figures 12-39 for the denial false positive curves trained on 3 classes. See Figures 40-67 for the denial false positive curves trained on 2 classes. As a reminder, the x-axis depicts the proportion of data that would be evaluated by the model based on the chosen approval threshold, and the y-axis depicts the denial false positive rate at this threshold.

A.2 Decision Trees

See Figures 68-99 for the denial false positive curves trained on 3 classes. See Figures 100-131 for the denial false positive curves trained on 2 classes. As a reminder, the x-axis depicts the proportion of data that would be evaluated by the model based on the chosen approval threshold, and the y-axis depicts the denial false positive rate at this threshold.

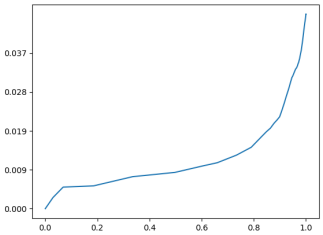

Figure 16: Curve for 3 class unweighted logistic regression with 0.001 l1 penalty

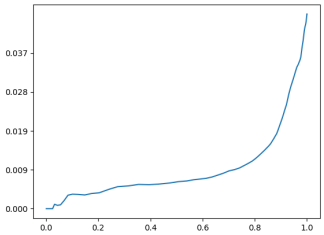

Figure 17: Curve for 3 class balanced logistic regression with 0.001 l1 penalty

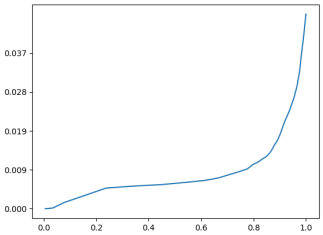

Figure 18: Curve for 3 class unweighted logistic regression with 0.0001 l1 penalty

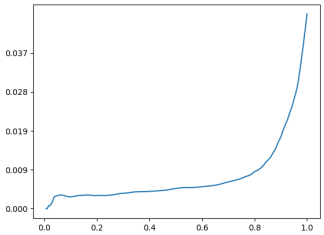

Figure 19: Curve for 3 class balanced logistic regression with 0.0001 l1 penalty

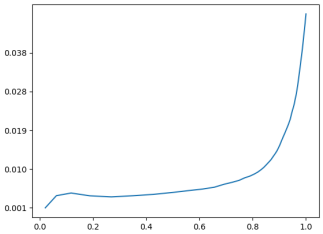

Figure 20: Curve for 3 class unweighted logistic regression with 1e-05 l1 penalty

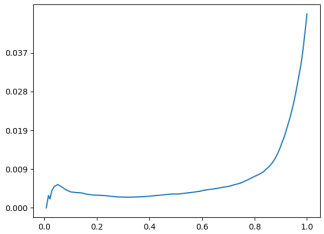

Figure 21: Curve for 3 class balanced logistic regression with 1e-05 l1 penalty

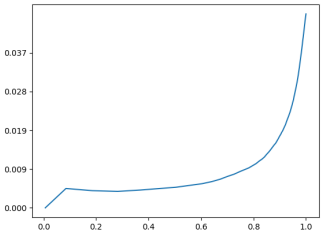

Figure 22: Curve for 3 class unweighted logistic regression with 1e-06 l1 penalty

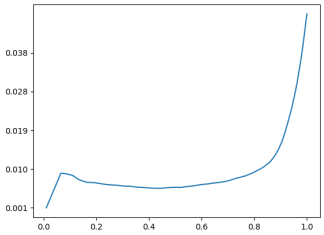

Figure 23: Curve for 3 class balanced logistic regression with 1e-06 l1 penalty

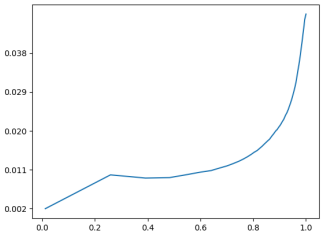

Figure 24: Curve for 3 class unweighted logistic regression with 1e-07 l1 penalty

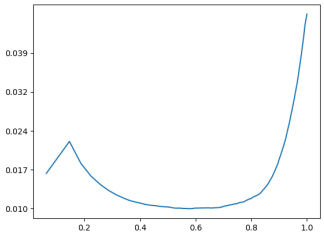

Figure 25: Curve for 3 class balanced logistic regression with 1e-07 l1 penalty

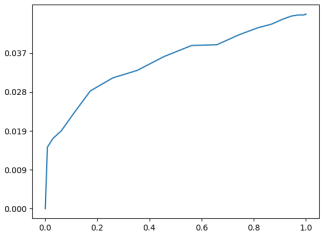

Figure 26: Curve for 3 class unweighted logistic regression with 0.1 l2 penalty

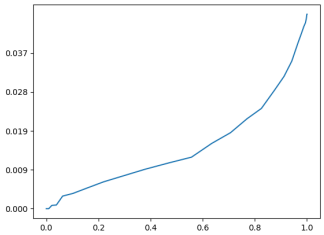

Figure 27: Curve for 3 class balanced logistic regression with 0.1 l2 penalty

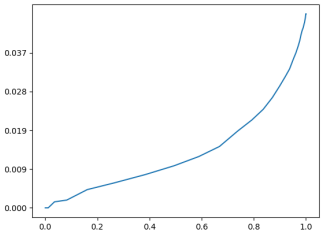

Figure 28: Curve for 3 class unweighted logistic regression with 0.01 l2 penalty

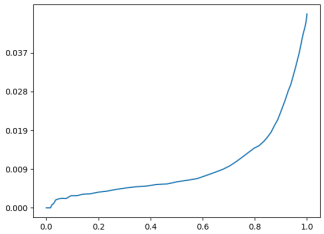

Figure 29: Curve for 3 class balanced logistic regression with 0.01 l2 penalty

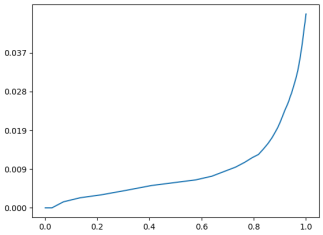

Figure 30: Curve for 3 class unweighted logistic regression with 0.001 l2 penalty

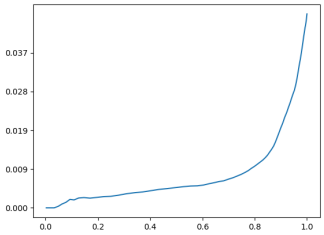

Figure 31: Curve for 3 class balanced logistic regression with 0.001 l2 penalty

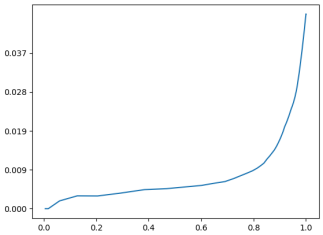

Figure 32: Curve for 3 class unweighted logistic regression with 0.0001 l2 penalty

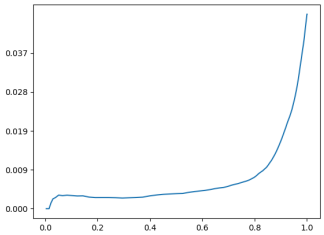

Figure 33: Curve for 3 class balanced logistic regression with 0.0001 l2 penalty

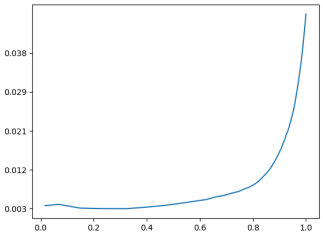

Figure 34: Curve for 3 class unweighted logistic regression with 1e-05 l2 penalty

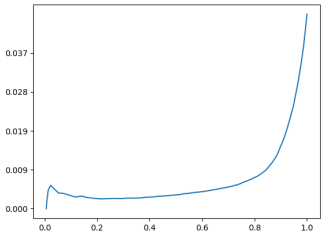

Figure 35: Curve for 3 class balanced logistic regression with 1e-05 l2 penalty

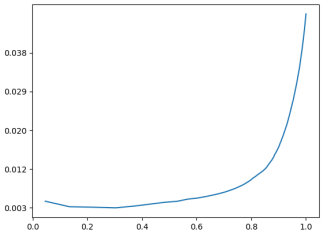

Figure 36: Curve for 3 class unweighted logistic regression with 1e-06 l2 penalty

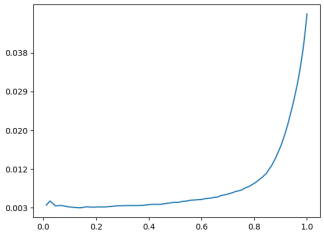

Figure 37: Curve for 3 class balanced logistic regression with 1e-06 l2 penalty

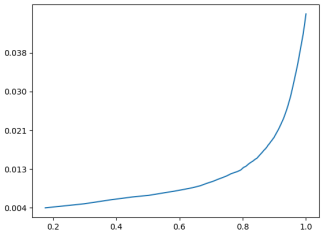

Figure 38: Curve for 3 class unweighted logistic regression with 1e-07 l2 penalty

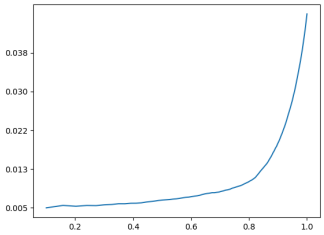

Figure 39: Curve for 3 class balanced logistic regression with 1e-07 l2 penalty

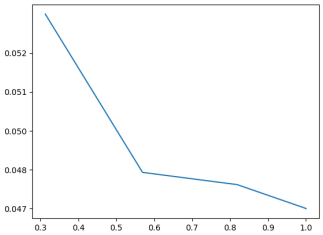

Figure 40: Curve for 2 class unweighted logistic regression with 0.1 l1 penalty

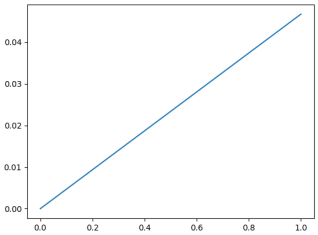

Figure 41: Curve for 2 class balanced logistic regression with 0.1 l1 penalty

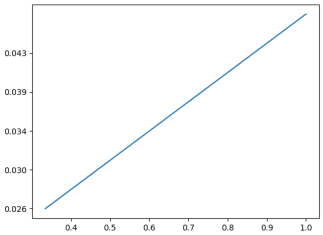

Figure 42: Curve for 2 class unweighted logistic regression with 0.01 l1 penalty

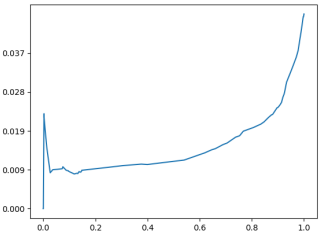

Figure 43: Curve for 2 class balanced logistic regression with 0.01 l1 penalty

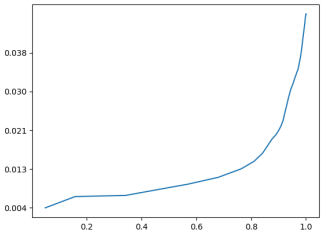

Figure 44: Curve for 2 class unweighted logistic regression with 0.001 l1 penalty

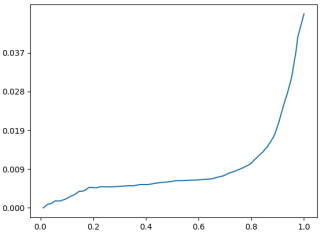

Figure 45: Curve for 2 class balanced logistic regression with 0.001 l1 penalty

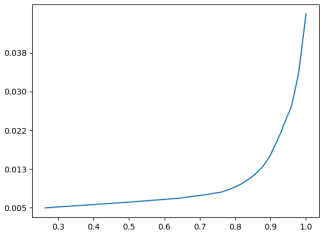

Figure 46: Curve for 2 class unweighted logistic regression with 0.0001 l1 penalty

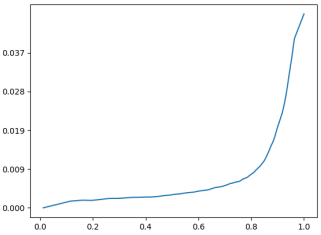

Figure 47: Curve for 2 class balanced logistic regression with 0.0001 l1 penalty

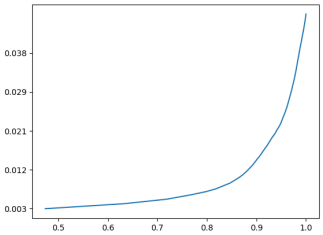

Figure 48: Curve for 2 class unweighted logistic regression with 1e-05 l1 penalty

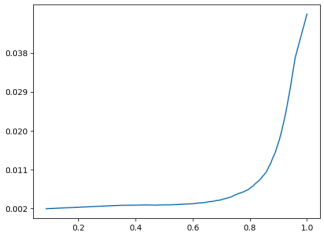

Figure 49: Curve for 2 class balanced logistic regression with 1e-05 l1 penalty

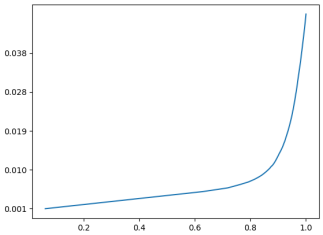

Figure 50: Curve for 2 class unweighted logistic regression with 1e-06 l1 penalty

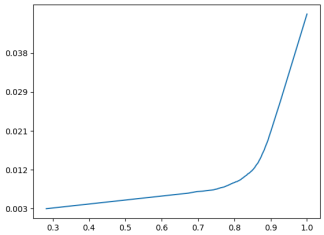

Figure 51: Curve for 2 class balanced logistic regression with 1e-06 l1 penalty

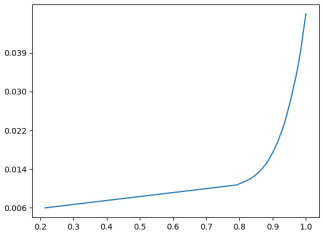

Figure 52: Curve for 2 class unweighted logistic regression with 1e-07 l1 penalty

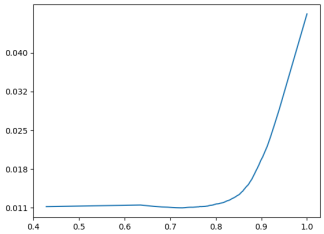

Figure 53: Curve for 2 class balanced logistic regression with 1e-07 l1 penalty

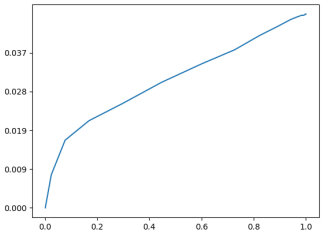

Figure 54: Curve for 2 class unweighted logistic regression with 0.1 l2 penalty

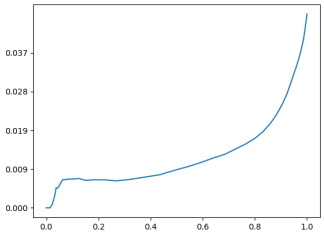

Figure 55: Curve for 2 class balanced logistic regression with 0.1 l2 penalty

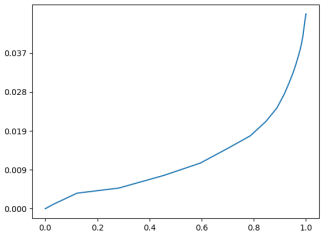

Figure 56: Curve for 2 class unweighted logistic regression with 0.01 l2 penalty

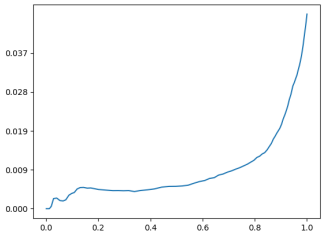

Figure 57: Curve for 2 class balanced logistic regression with 0.01 l2 penalty

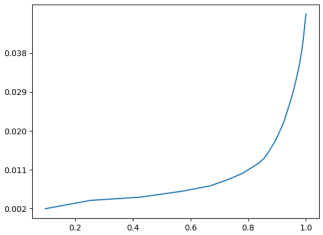

Figure 58: Curve for 2 class unweighted logistic regression with 0.001 l2 penalty

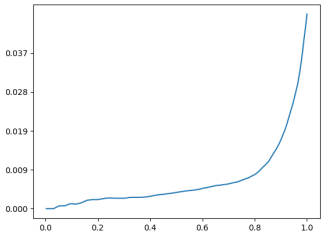

Figure 59: Curve for 2 class balanced logistic regression with 0.001 l2 penalty

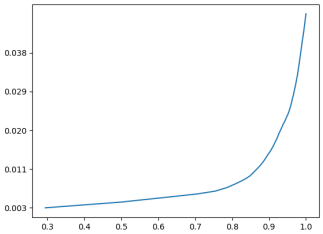

Figure 60: Curve for 2 class unweighted logistic regression with 0.0001 l2 penalty

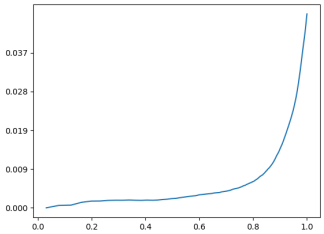

Figure 61: Curve for 2 class balanced logistic regression with 0.0001 l2 penalty

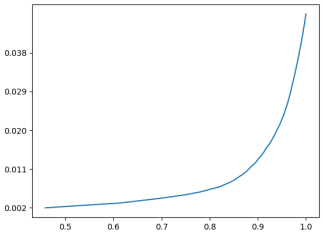

Figure 62: Curve for 2 class unweighted logistic regression with 1e-05 l2 penalty

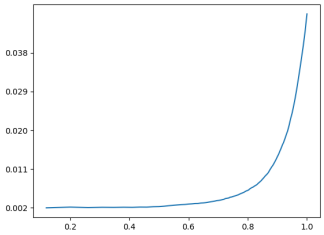

Figure 63: Curve for 2 class balanced logistic regression with 1e-05 l2 penalty

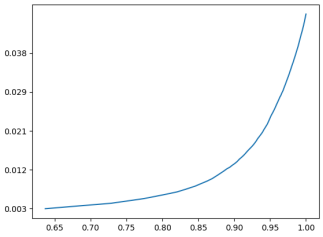

Figure 64: Curve for 2 class unweighted logistic regression with 1e-06 l2 penalty

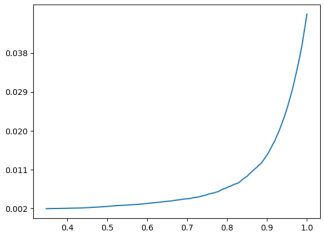

Figure 65: Curve for 2 class balanced logistic regression with 1e-06 l2 penalty

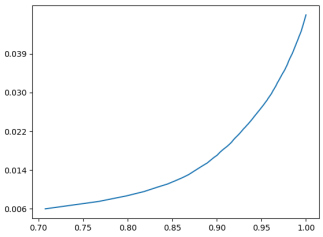

Figure 66: Curve for 2 class unweighted logistic regression with 1e-07 l2 penalty

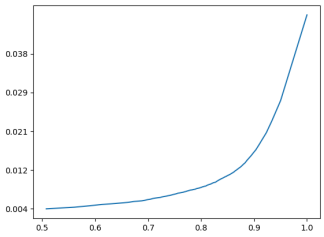

Figure 67: Curve for 2 class balanced logistic regression with 1e-07 l2 penalty

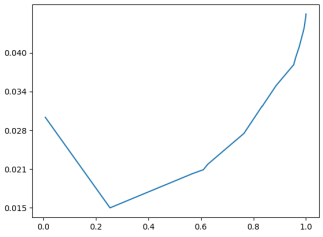

Figure 68: Curve for 3 class unweighted decision tree with depth 5 and 30% features

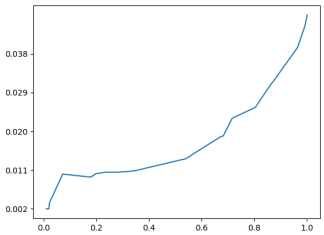

Figure 69: Curve for 3 class balanced decision tree with depth 5 and 30% features

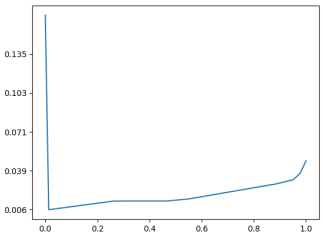

Figure 70: Curve for 3 class unweighted decision tree with depth 5 and 50% features

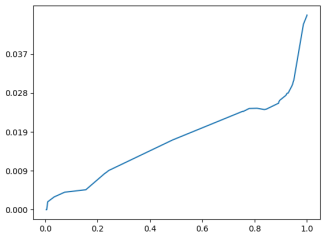

Figure 71: Curve for 3 class balanced decision tree with depth 5 and 50% features

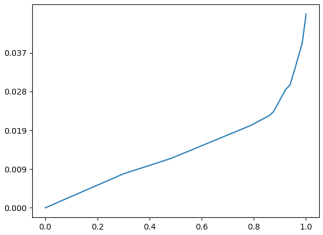

Figure 72: Curve for 3 class unweighted decision tree with depth 5 and 70% features

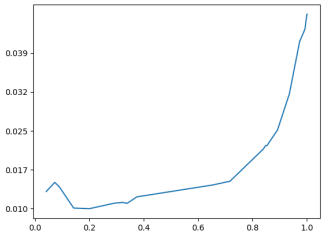

Figure 73: Curve for 3 class balanced decision tree with depth 5 and 70% features

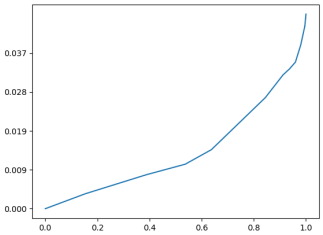

Figure 74: Curve for 3 class unweighted decision tree with depth 5 and 100% features

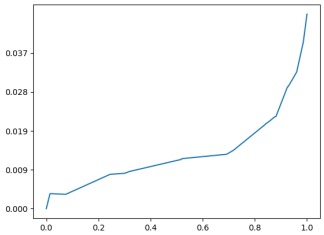

Figure 75: Curve for 3 class balanced decision tree with depth 5 and 100% features

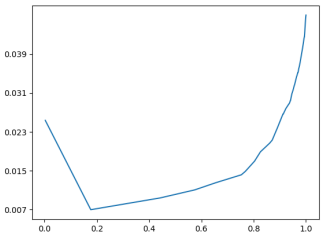

Figure 76: Curve for 3 class unweighted decision tree with depth 10 and 30% features

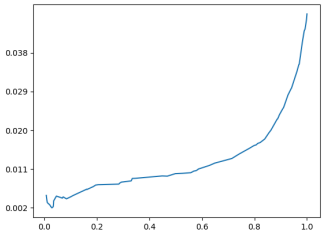

Figure 77: Curve for 3 class balanced decision tree with depth 10 and 30% features

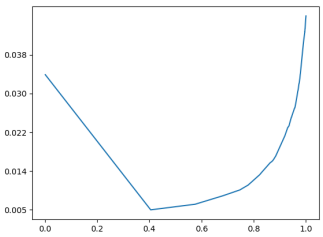

Figure 78: Curve for 3 class unweighted decision tree with depth 10 and 50% features

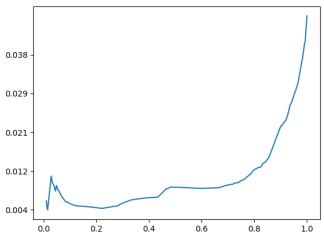

Figure 79: Curve for 3 class balanced decision tree with depth 10 and 50% features

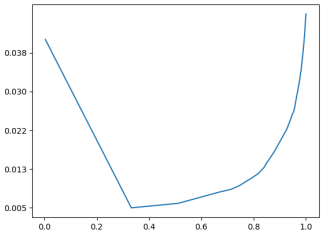

Figure 80: Curve for 3 class unweighted decision tree with depth 10 and 70% features

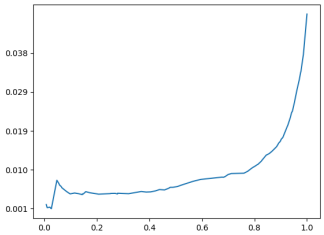

Figure 81: Curve for 3 class balanced decision tree with depth 10 and 70% features

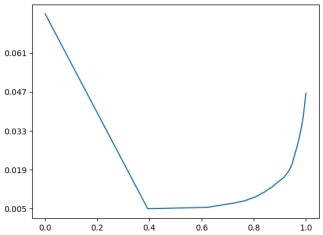

Figure 82: Curve for 3 class unweighted decision tree with depth 10 and 100% features

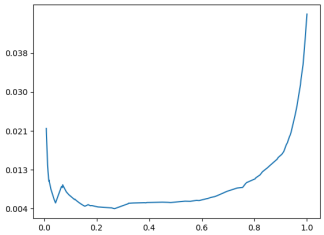

Figure 83: Curve for 3 class balanced decision tree with depth 10 and 100% features

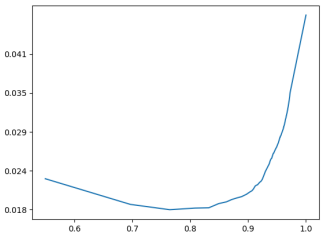

Figure 84: Curve for 3 class unweighted decision tree with depth 20 and 30% features

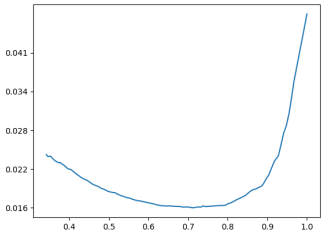

Figure 85: Curve for 3 class balanced decision tree with depth 20 and 30% features

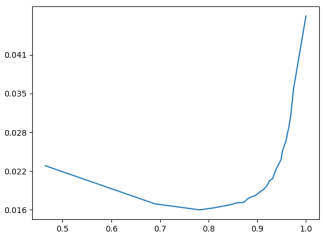

Figure 86: Curve for 3 class unweighted decision tree with depth 20 and 50% features

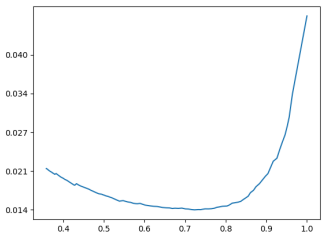

Figure 87: Curve for 3 class balanced decision tree with depth 20 and 50% features

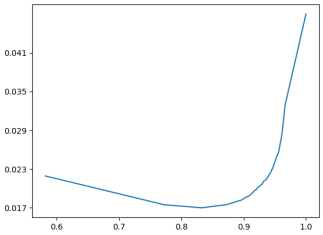

Figure 88: Curve for 3 class unweighted decision tree with depth 20 and 70% features

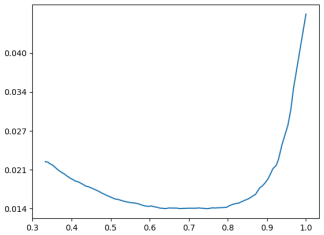

Figure 89: Curve for 3 class balanced decision tree with depth 20 and 70% features

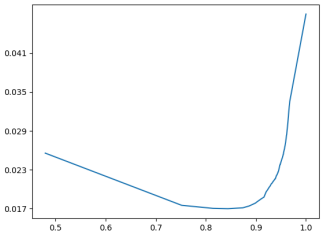

Figure 90: Curve for 3 class unweighted decision tree with depth 20 and 100% features

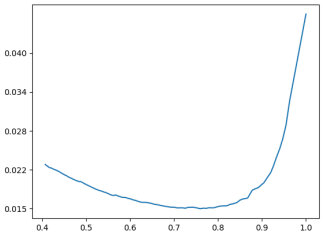

Figure 91: Curve for 3 class balanced decision tree with depth 20 and 100% features

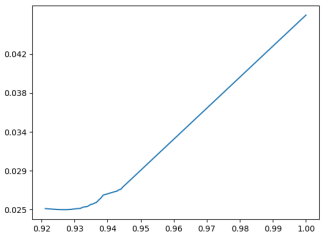

Figure 92: Curve for 3 class unweighted decision tree with depth 30 and 30% features

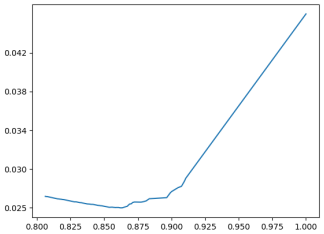

Figure 93: Curve for 3 class balanced decision tree with depth 30 and 30% features

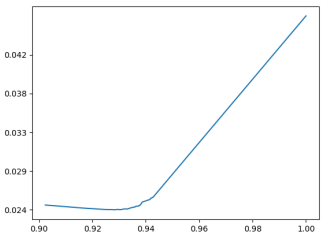

Figure 94: Curve for 3 class unweighted decision tree with depth 30 and 50% features

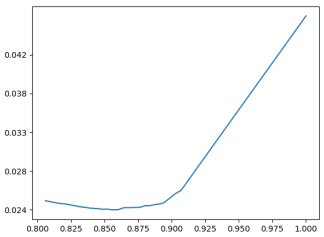

Figure 95: Curve for 3 class balanced decision tree with depth 30 and 50% features

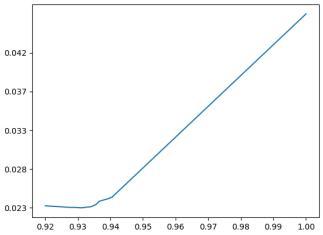

Figure 96: Curve for 3 class unweighted decision tree with depth 30 and 70% features

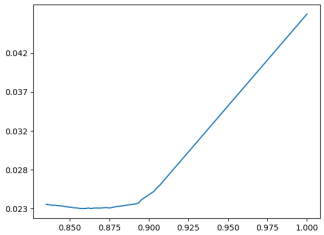

Figure 97: Curve for 3 class balanced decision tree with depth 30 and 70% features

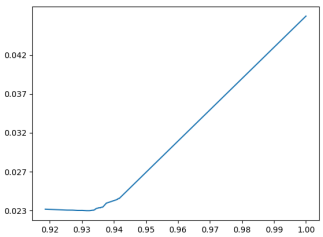

Figure 98: Curve for 3 class unweighted decision tree with depth 30 and 100% features

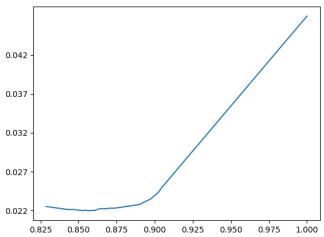

Figure 99: Curve for 3 class balanced decision tree with depth 30 and 100% features

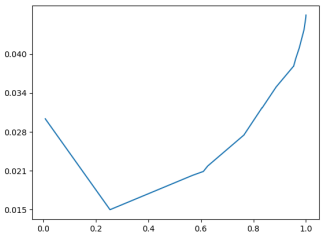

Figure 100: Curve for 2 class unweighted decision tree with depth 5 and 30% features

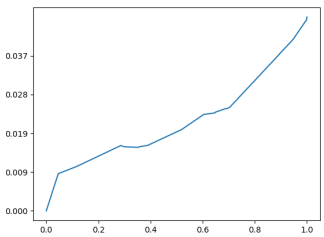

Figure 101: Curve for 2 class balanced decision tree with depth 5 and 30% features

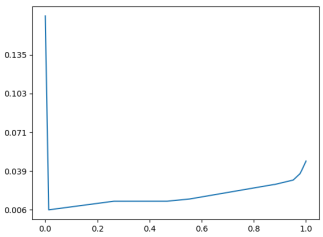

Figure 102: Curve for 2 class unweighted decision tree with depth 5 and 50% features

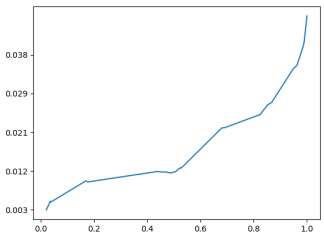

Figure 103: Curve for 2 class balanced decision tree with depth 5 and 50% features

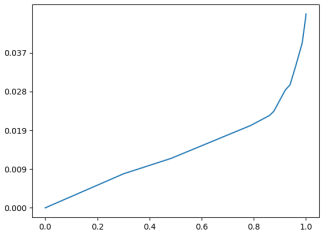

Figure 104: Curve for 2 class unweighted decision tree with depth 5 and 70% features

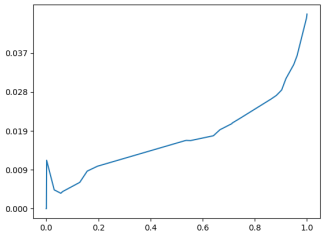

Figure 105: Curve for 2 class balanced decision tree with depth 5 and 70% features

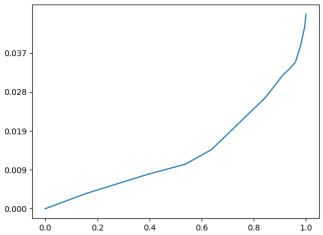

Figure 106: Curve for 2 class unweighted decision tree with depth 5 and 100% features

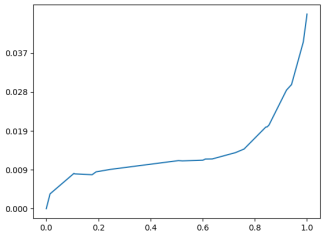

Figure 107: Curve for 2 class balanced decision tree with depth 5 and 100% features

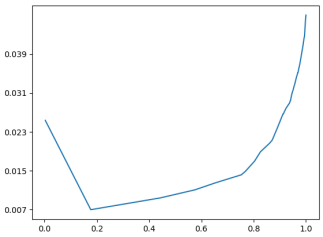

Figure 108: Curve for 2 class unweighted decision tree with depth 10 and 30% features

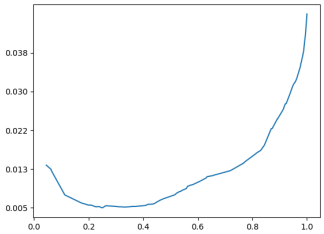

Figure 109: Curve for 2 class balanced decision tree with depth 10 and 30% features

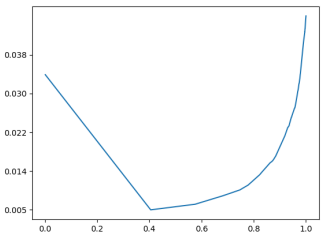

Figure 110: Curve for 2 class unweighted decision tree with depth 10 and 50% features

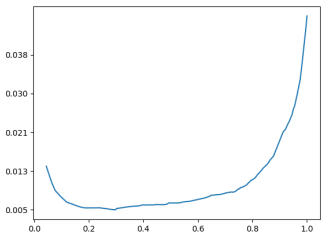

Figure 111: Curve for 2 class balanced decision tree with depth 10 and 50% features

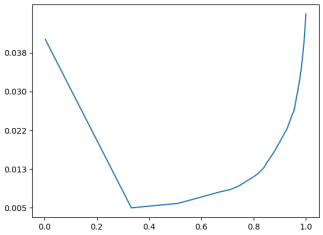

Figure 112: Curve for 2 class unweighted decision tree with depth 10 and 70% features

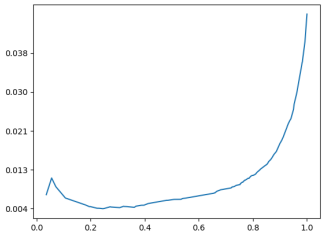

Figure 113: Curve for 2 class balanced decision tree with depth 10 and 70% features

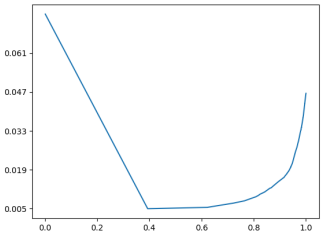

Figure 114: Curve for 2 class unweighted decision tree with depth 10 and 100% features

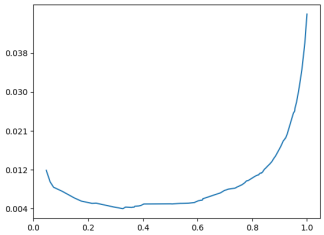

Figure 115: Curve for 2 class balanced decision tree with depth 10 and 100% features

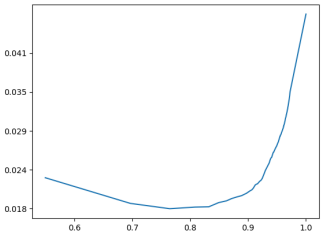

Figure 116: Curve for 2 class unweighted decision tree with depth 20 and 30% features

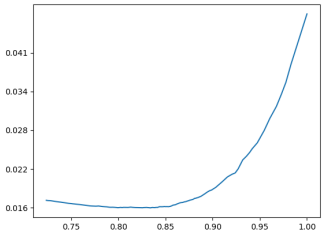

Figure 117: Curve for 2 class balanced decision tree with depth 20 and 30% features

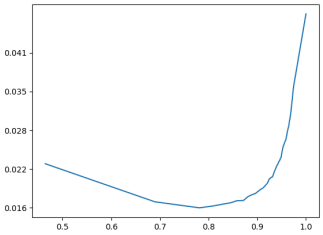

Figure 118: Curve for 2 class unweighted decision tree with depth 20 and 50% features

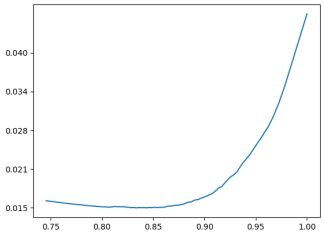

Figure 119: Curve for 2 class balanced decision tree with depth 20 and 50% features

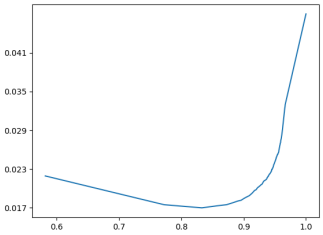

Figure 120: Curve for 2 class unweighted decision tree with depth 20 and 70% features

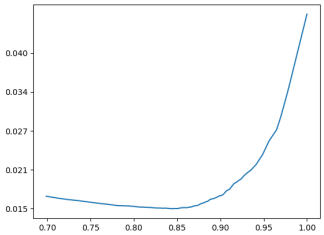

Figure 121: Curve for 2 class balanced decision tree with depth 20 and 70% features

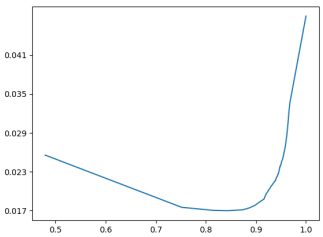

Figure 122: Curve for 2 class unweighted decision tree with depth 20 and 100% features

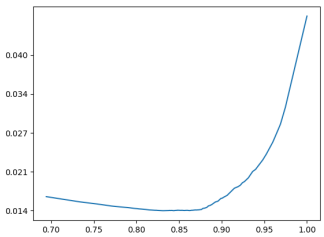

Figure 123: Curve for 2 class balanced decision tree with depth 20 and 100% features

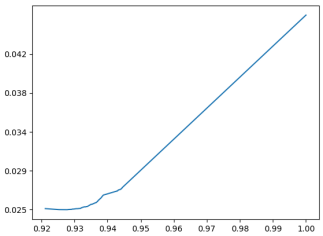

Figure 124: Curve for 2 class unweighted decision tree with depth 30 and 30% features

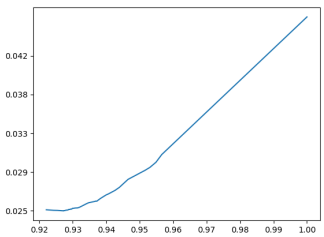

Figure 125: Curve for 2 class balanced decision tree with depth 30 and 30% features

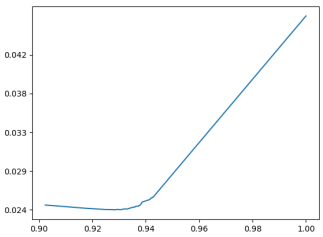

Figure 126: Curve for 2 class unweighted decision tree with depth 30 and 50% features

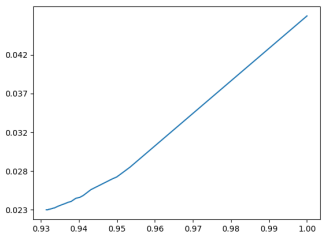

Figure 127: Curve for 2 class balanced decision tree with depth 30 and 50% features

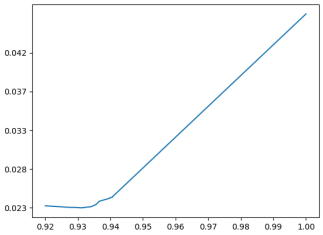

Figure 128: Curve for 2 class unweighted decision tree with depth 30 and 70% features

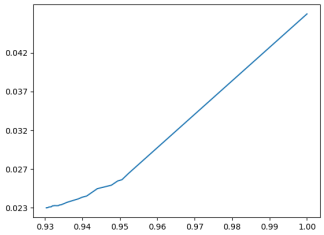

Figure 129: Curve for 2 class balanced decision tree with depth 30 and 70% features

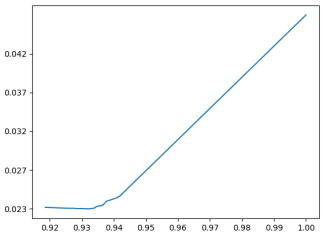

Figure 130: Curve for 2 class unweighted decision tree with depth 30 and 100% features

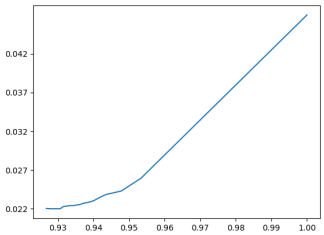

Figure 131: Curve for 2 class balanced decision tree with depth 30 and 100% features

A.3 Random Forest

See Figures 132-227 for the denial false positive curves trained on 3 classes. See Figures 228-323 for the denial false positive curves trained on 2 classes. As a reminder, the x-axis depicts the proportion of data that would be evaluated by the model based on the chosen approval threshold, and the y-axis depicts the denial false positive rate at this threshold.

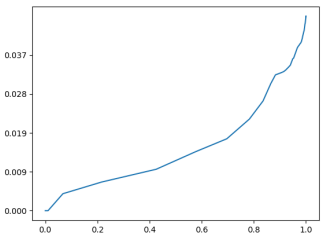

Figure 132: Curve for unweighted 3 class random forest with depth 5 and 30% features

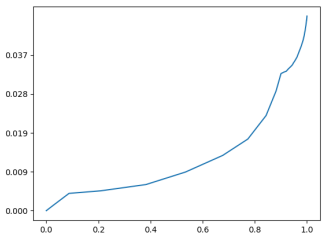

Figure 133: Curve for unweighted 3 class random forest with depth 5 and 30% features

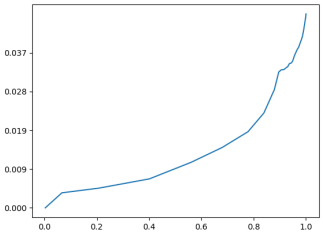

Figure 134: Curve for unweighted 3 class random forest with depth 5 and 30% features

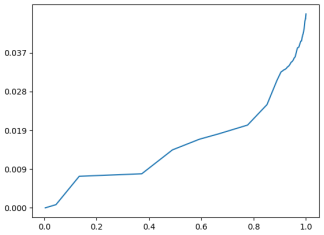

Figure 135: Curve for unweighted 3 class random forest with depth 5 and 50% features

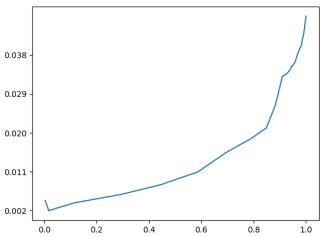

Figure 136: Curve for unweighted 3 class random forest with depth 5 and 50% features

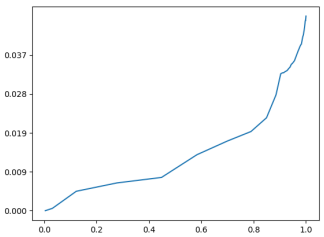

Figure 137: Curve for unweighted 3 class random forest with depth 5 and 50% features

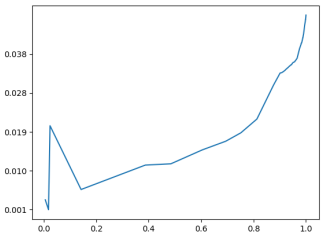

Figure 138: Curve for unweighted 3 class random forest with depth 5 and 70% features

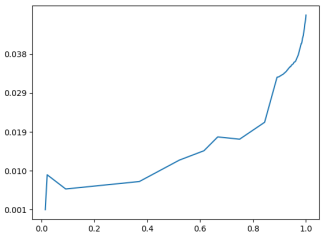

Figure 139: Curve for unweighted 3 class random forest with depth 5 and 70% features

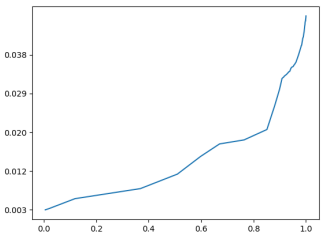

Figure 140: Curve for unweighted 3 class random forest with depth 5 and 70% features

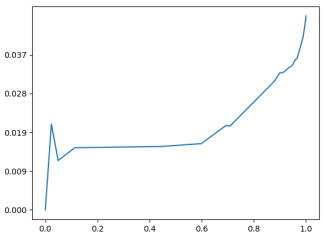

Figure 141: Curve for unweighted 3 class random forest with depth 5 and 100% features

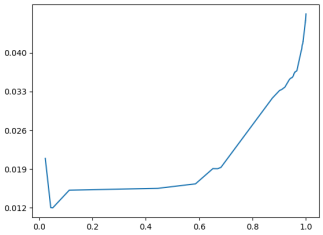

Figure 142: Curve for unweighted 3 class random forest with depth 5 and 100% features

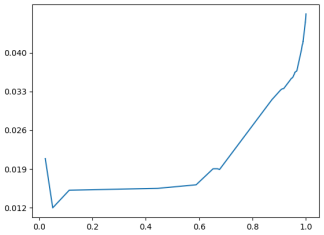

Figure 143: Curve for unweighted 3 class random forest with depth 5 and 100% features

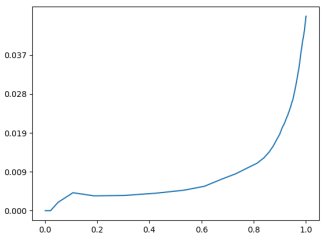

Figure 144: Curve for unweighted 3 class random forest with depth 10 and 30% features

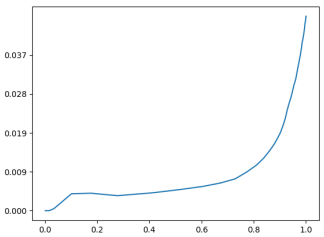

Figure 145: Curve for unweighted 3 class random forest with depth 10 and 30% features

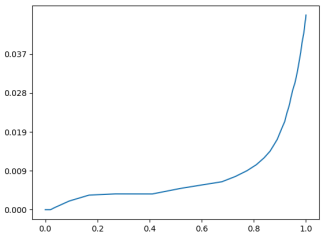

Figure 146: Curve for unweighted 3 class random forest with depth 10 and 30% features

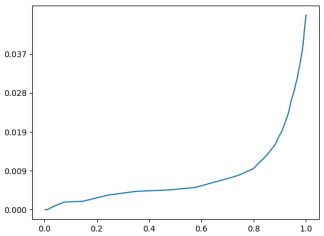

Figure 147: Curve for unweighted 3 class random forest with depth 10 and 50% features

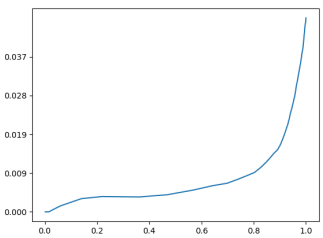

Figure 148: Curve for unweighted 3 class random forest with depth 10 and 50% features

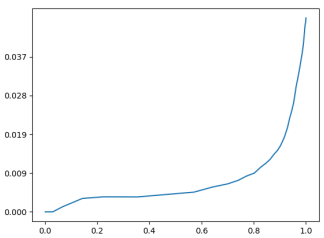

Figure 149: Curve for unweighted 3 class random forest with depth 10 and 50% features

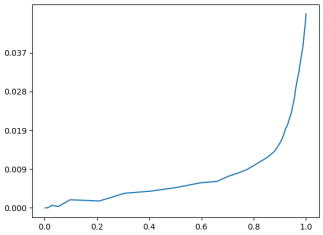

Figure 150: Curve for unweighted 3 class random forest with depth 10 and 70% features

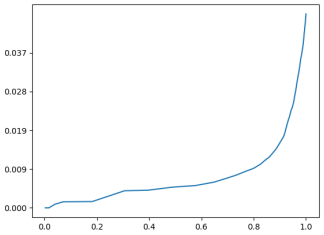

Figure 151: Curve for unweighted 3 class random forest with depth 10 and 70% features

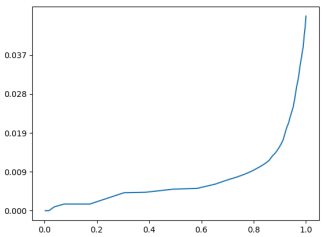

Figure 152: Curve for unweighted 3 class random forest with depth 10 and 70% features

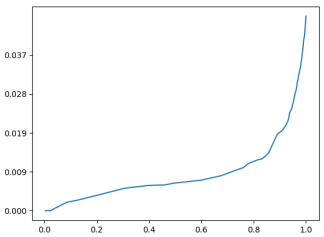

Figure 153: Curve for unweighted 3 class random forest with depth 10 and 100% features

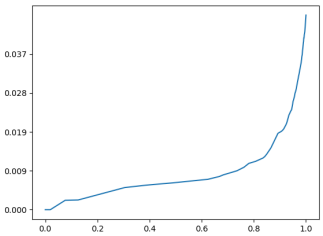

Figure 154: Curve for unweighted 3 class random forest with depth 10 and 100% features

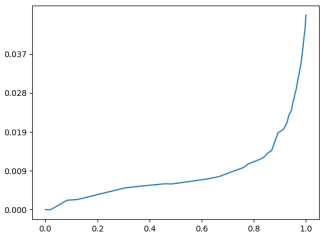

Figure 155: Curve for unweighted 3 class random forest with depth 10 and 100% features

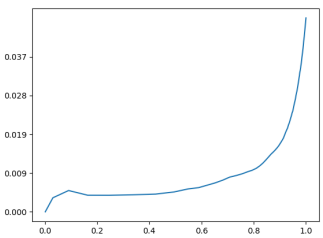

Figure 156: Curve for unweighted 3 class random forest with depth 20 and 30% features

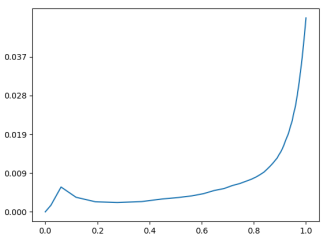

Figure 157: Curve for unweighted 3 class random forest with depth 20 and 30% features

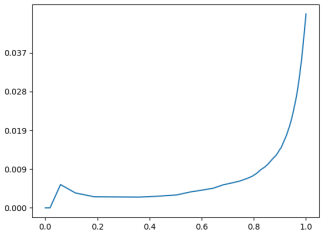

Figure 158: Curve for unweighted 3 class random forest with depth 20 and 30% features

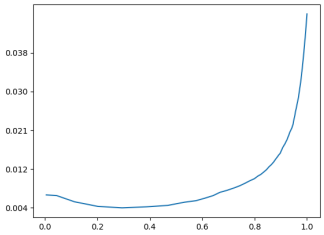

Figure 159: Curve for unweighted 3 class random forest with depth 20 and 50% features

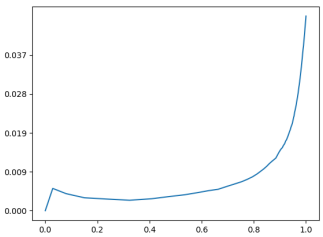

Figure 160: Curve for unweighted 3 class random forest with depth 20 and 50% features

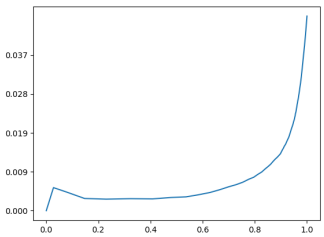

Figure 161: Curve for unweighted 3 class random forest with depth 20 and 50% features

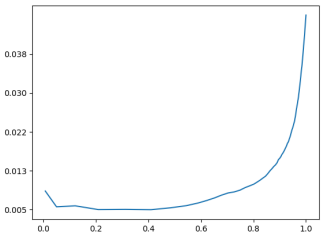

Figure 162: Curve for unweighted 3 class random forest with depth 20 and 70% features

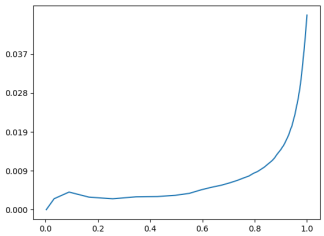

Figure 163: Curve for unweighted 3 class random forest with depth 20 and 70% features

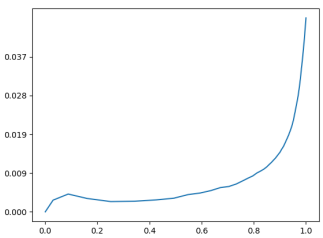

Figure 164: Curve for unweighted 3 class random forest with depth 20 and 70% features

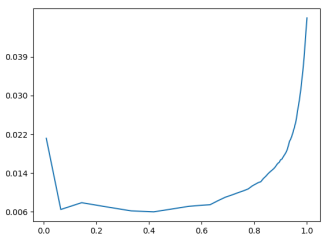

Figure 165: Curve for unweighted 3 class random forest with depth 20 and 100% features

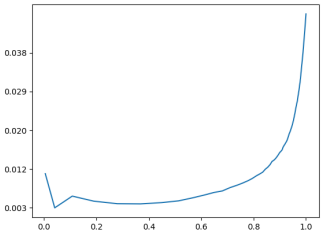

Figure 166: Curve for unweighted 3 class random forest with depth 20 and 100% features

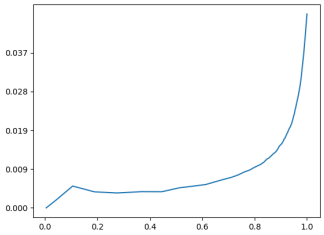

Figure 167: Curve for unweighted 3 class random forest with depth 20 and 100% features

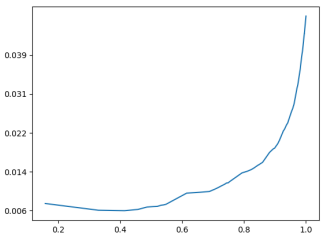

Figure 168: Curve for unweighted 3 class random forest with depth 30 and 30% features

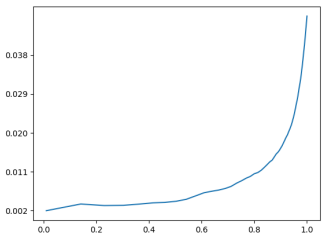

Figure 169: Curve for unweighted 3 class random forest with depth 30 and 30% features

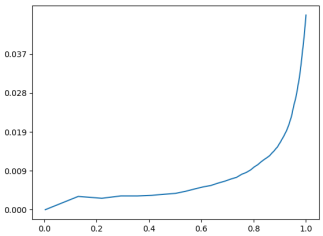

Figure 170: Curve for unweighted 3 class random forest with depth 30 and 30% features

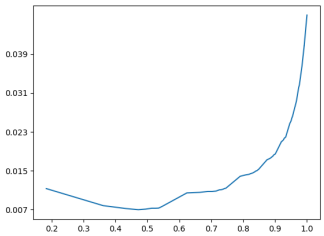

Figure 171: Curve for unweighted 3 class random forest with depth 30 and 50% features

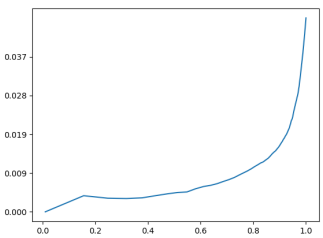

Figure 172: Curve for unweighted 3 class random forest with depth 30 and 50% features

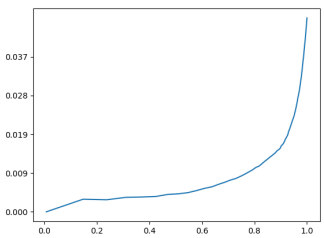

Figure 173: Curve for unweighted 3 class random forest with depth 30 and 50% features

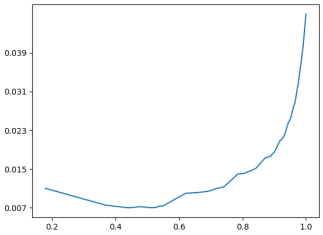

Figure 174: Curve for unweighted 3 class random forest with depth 30 and 70% features

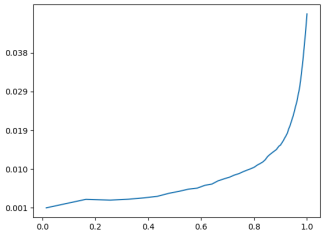

Figure 175: Curve for unweighted 3 class random forest with depth 30 and 70% features

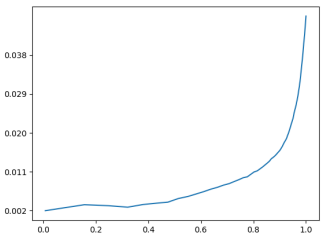

Figure 176: Curve for unweighted 3 class random forest with depth 30 and 70% features

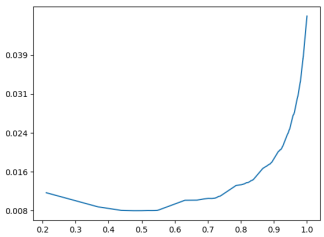

Figure 177: Curve for unweighted 3 class random forest with depth 30 and 100% features

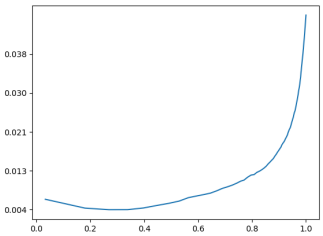

Figure 178: Curve for unweighted 3 class random forest with depth 30 and 100% features

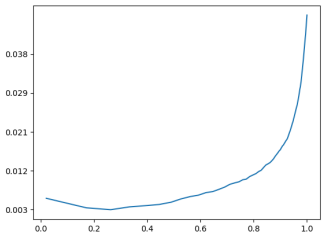

Figure 179: Curve for unweighted 3 class random forest with depth 30 and 100% features

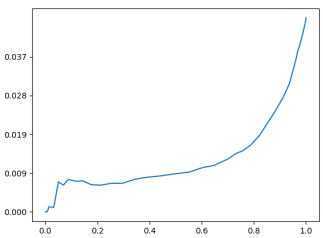

Figure 180: Curve for balanced 3 class random forest with depth 5 and 30% features

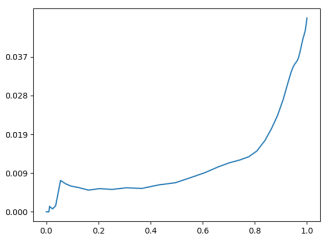

Figure 181: Curve for balanced 3 class random forest with depth 5 and 30% features

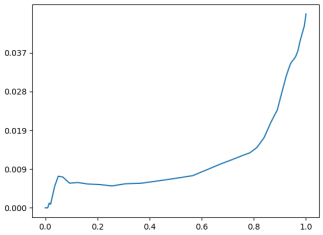

Figure 182: Curve for balanced 3 class random forest with depth 5 and 30% features

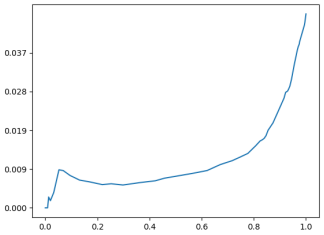

Figure 183: Curve for balanced 3 class random forest with depth 5 and 50% features

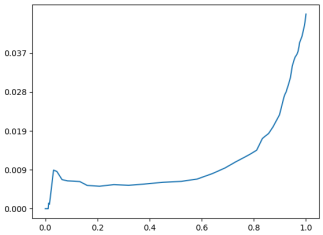

Figure 184: Curve for balanced 3 class random forest with depth 5 and 50% features

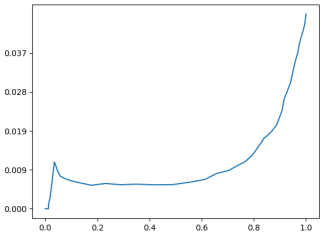

Figure 185: Curve for balanced 3 class random forest with depth 5 and 50% features

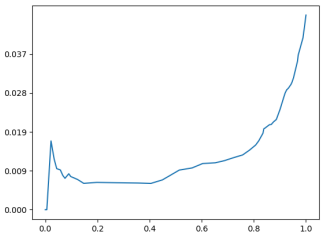

Figure 186: Curve for balanced 3 class random forest with depth 5 and 70% features

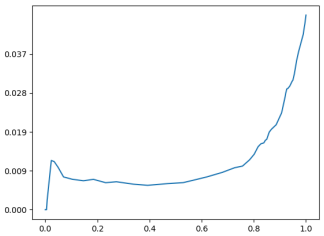

Figure 187: Curve for balanced 3 class random forest with depth 5 and 70% features

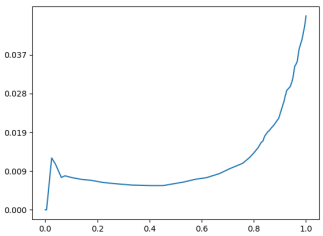

Figure 188: Curve for balanced 3 class random forest with depth 5 and 70% features

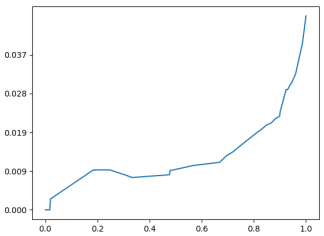

Figure 189: Curve for balanced 3 class random forest with depth 5 and 100% features

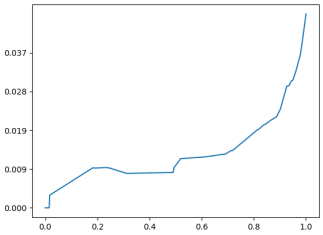

Figure 190: Curve for balanced 3 class random forest with depth 5 and 100% features

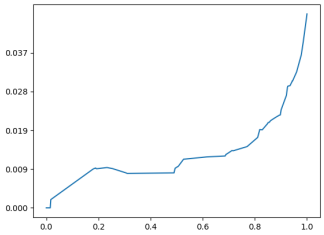

Figure 191: Curve for balanced 3 class random forest with depth 5 and 100% features

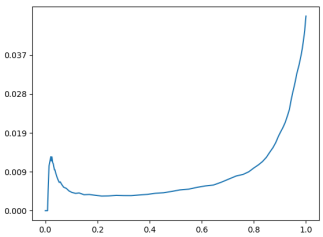

Figure 192: Curve for balanced 3 class random forest with depth 10 and 30% features

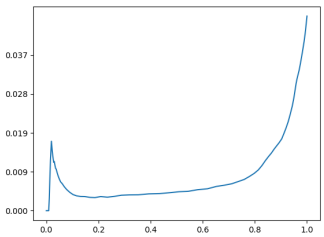

Figure 193: Curve for balanced 3 class random forest with depth 10 and 30% features

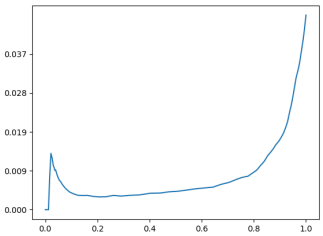

Figure 194: Curve for balanced 3 class random forest with depth 10 and 30% features

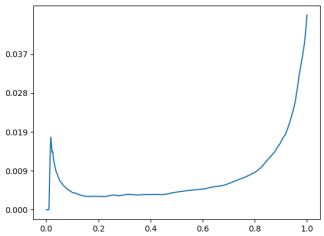

Figure 195: Curve for balanced 3 class random forest with depth 10 and 50% features

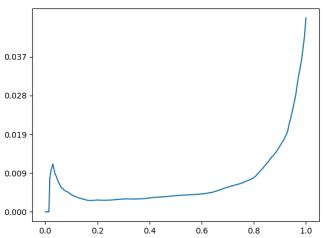

Figure 196: Curve for balanced 3 class random forest with depth 10 and 50% features

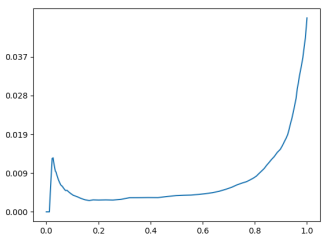

Figure 197: Curve for balanced 3 class random forest with depth 10 and 50% features

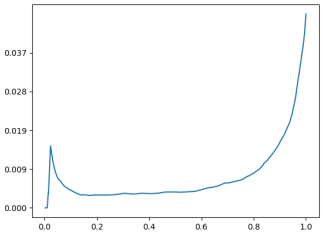

Figure 198: Curve for balanced 3 class random forest with depth 10 and 70% features

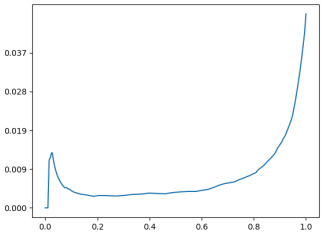

Figure 199: Curve for balanced 3 class random forest with depth 10 and 70% features

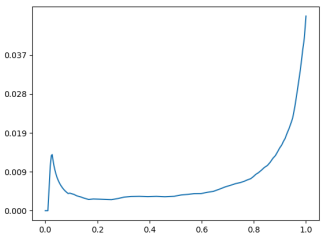

Figure 200: Curve for balanced 3 class random forest with depth 10 and 70% features

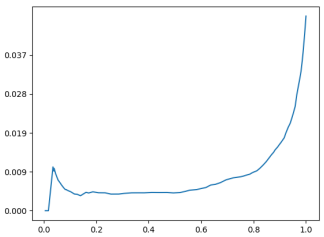

Figure 201: Curve for balanced 3 class random forest with depth 10 and 100% features

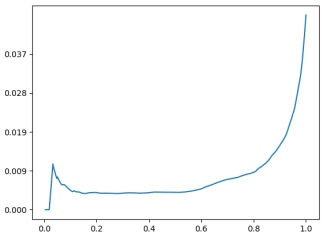

Figure 202: Curve for balanced 3 class random forest with depth 10 and 100% features

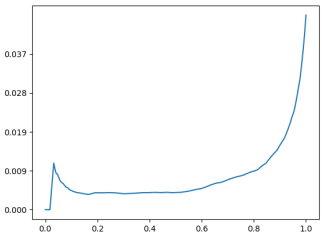

Figure 203: Curve for balanced 3 class random forest with depth 10 and 100% features

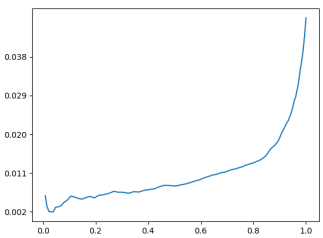

Figure 204: Curve for balanced 3 class random forest with depth 20 and 30% features

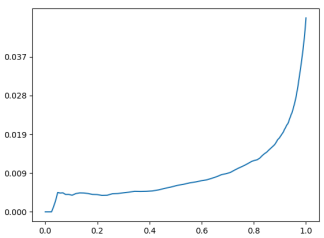

Figure 205: Curve for balanced 3 class random forest with depth 20 and 30% features

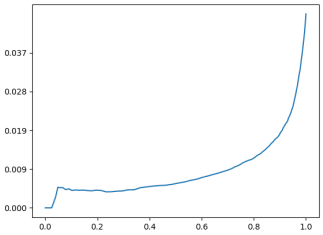

Figure 206: Curve for balanced 3 class random forest with depth 20 and 30% features

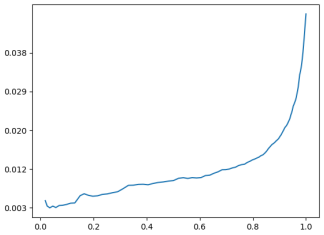

Figure 207: Curve for balanced 3 class random forest with depth 20 and 50% features

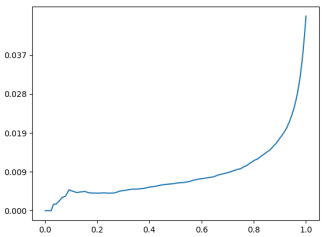

Figure 208: Curve for balanced 3 class random forest with depth 20 and 50% features

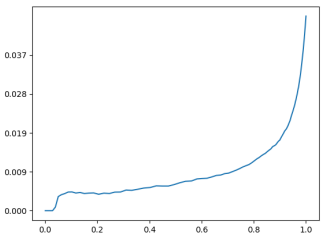

Figure 209: Curve for balanced 3 class random forest with depth 20 and 50% features

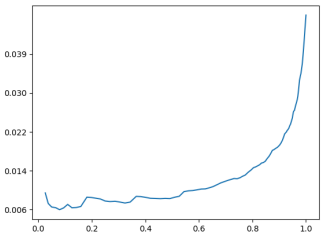

Figure 210: Curve for balanced 3 class random forest with depth 20 and 70% features

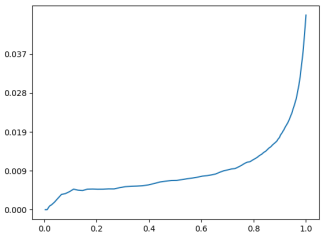

Figure 211: Curve for balanced 3 class random forest with depth 20 and 70% features

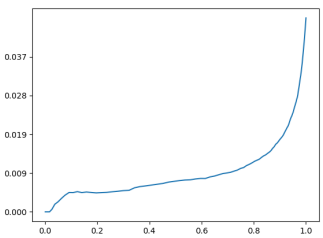

Figure 212: Curve for balanced 3 class random forest with depth 20 and 70% features

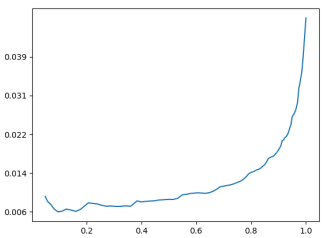

Figure 213: Curve for balanced 3 class random forest with depth 20 and 100% features

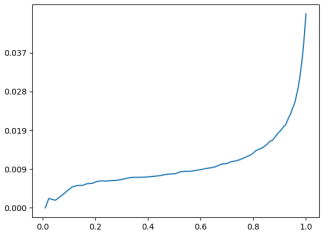

Figure 214: Curve for balanced 3 class random forest with depth 20 and 100% features

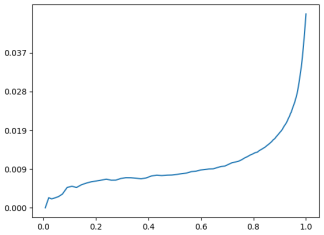

Figure 215: Curve for balanced 3 class random forest with depth 20 and 100% features

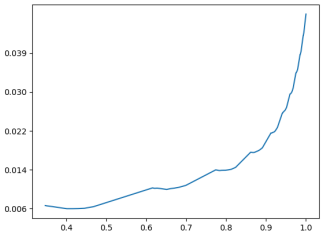

Figure 216: Curve for balanced 3 class random forest with depth 30 and 30% features

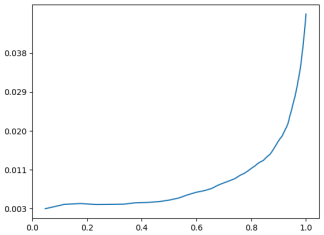

Figure 217: Curve for balanced 3 class random forest with depth 30 and 30% features

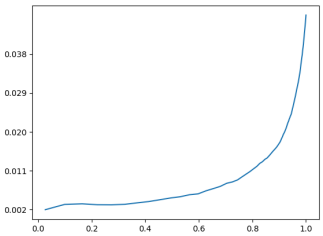

Figure 218: Curve for balanced 3 class random forest with depth 30 and 30% features

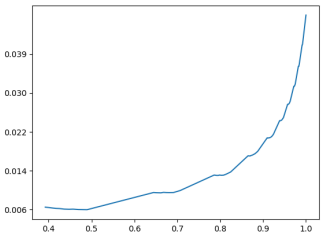

Figure 219: Curve for balanced 3 class random forest with depth 30 and 50% features

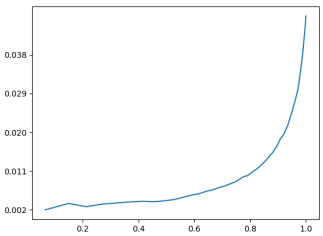

Figure 220: Curve for balanced 3 class random forest with depth 30 and 50% features

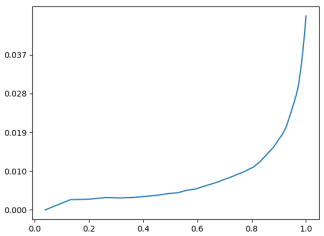

Figure 221: Curve for balanced 3 class random forest with depth 30 and 50% features

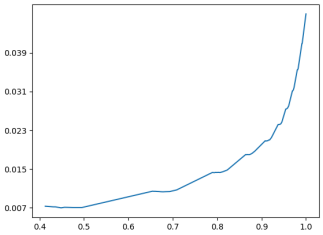

Figure 222: Curve for balanced 3 class random forest with depth 30 and 70% features

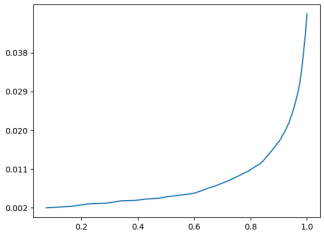

Figure 223: Curve for balanced 3 class random forest with depth 30 and 70% features

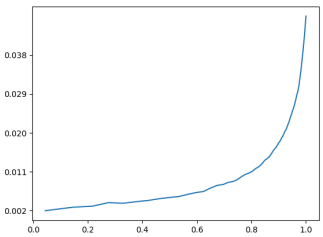

Figure 224: Curve for balanced 3 class random forest with depth 30 and 70% features

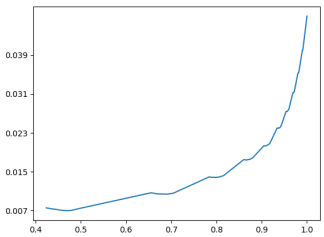

Figure 225: Curve for balanced 3 class random forest with depth 30 and 100% features

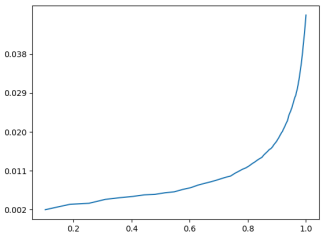

Figure 226: Curve for balanced 3 class random forest with depth 30 and 100% features

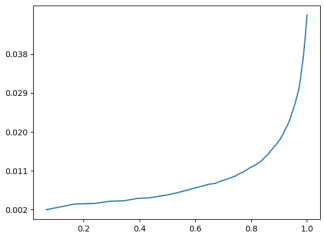

Figure 227: Curve for balanced 3 class random forest with depth 30 and 100% features

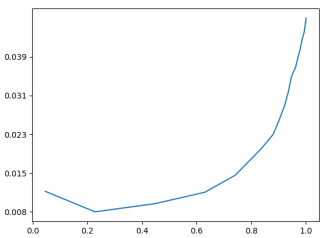

Figure 228: Curve for unweighted 2 class random forest with depth 5 and 30% features

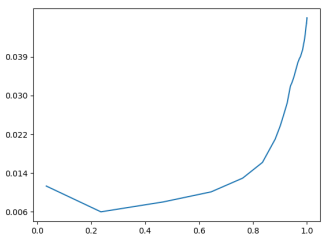

Figure 229: Curve for unweighted 2 class random forest with depth 5 and 30% features

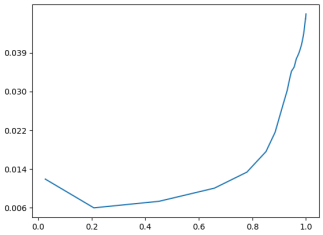

Figure 230: Curve for unweighted 2 class random forest with depth 5 and 30% features

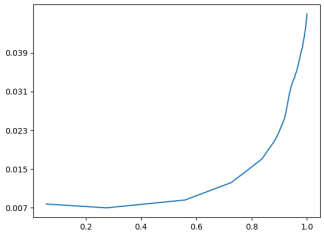

Figure 231: Curve for unweighted 2 class random forest with depth 5 and 50% features

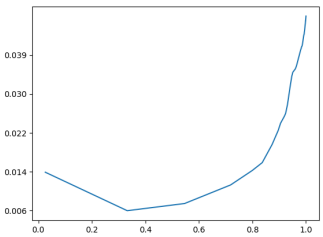

Figure 232: Curve for unweighted 2 class random forest with depth 5 and 50% features

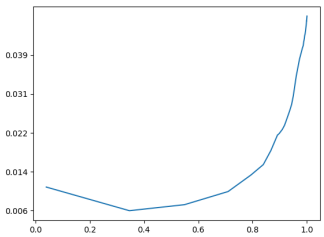

Figure 233: Curve for unweighted 2 class random forest with depth 5 and 50% features

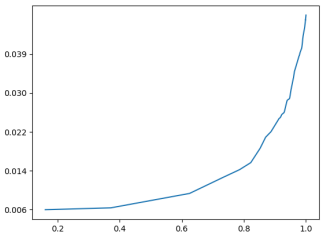

Figure 234: Curve for unweighted 2 class random forest with depth 5 and 70% features

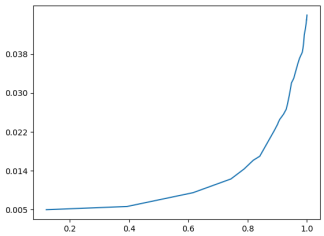

Figure 235: Curve for unweighted 2 class random forest with depth 5 and 70% features

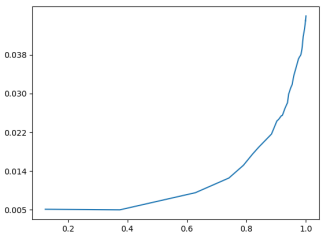

Figure 236: Curve for unweighted 2 class random forest with depth 5 and 70% features

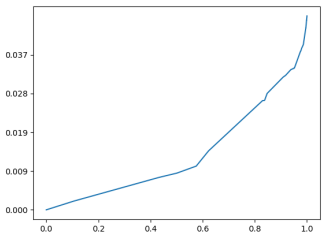

Figure 237: Curve for unweighted 2 class random forest with depth 5 and 100% features

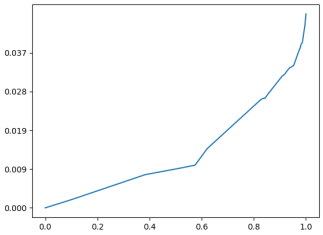

Figure 238: Curve for unweighted 2 class random forest with depth 5 and 100% features

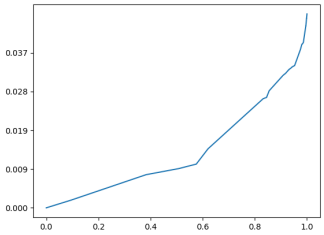

Figure 239: Curve for unweighted 2 class random forest with depth 5 and 100% features

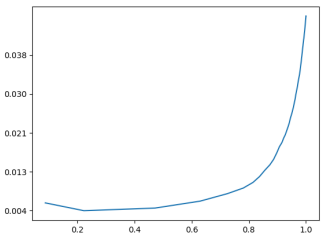

Figure 240: Curve for unweighted 2 class random forest with depth 10 and 30% features

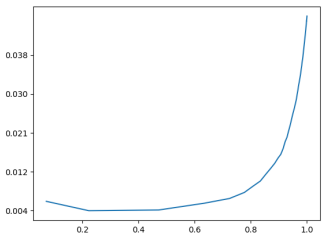

Figure 241: Curve for unweighted 2 class random forest with depth 10 and 30% features

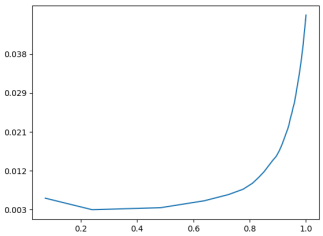

Figure 242: Curve for unweighted 2 class random forest with depth 10 and 30% features

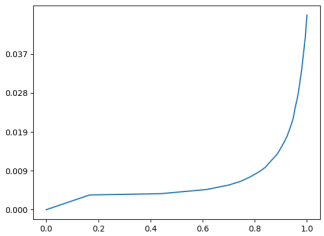

Figure 243: Curve for unweighted 2 class random forest with depth 10 and 50% features

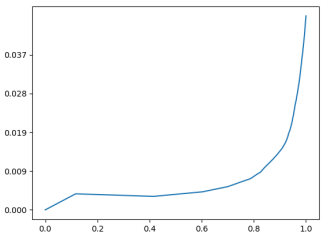

Figure 244: Curve for unweighted 2 class random forest with depth 10 and 50% features

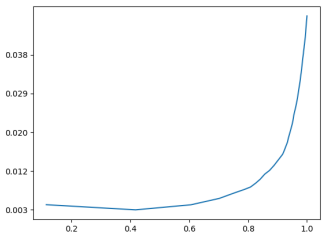

Figure 245: Curve for unweighted 2 class random forest with depth 10 and 50% features

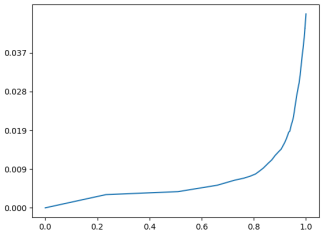

Figure 246: Curve for unweighted 2 class random forest with depth 10 and 70% features

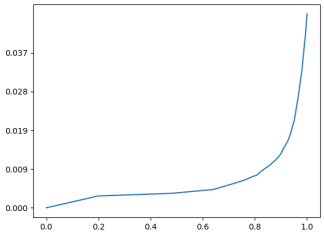

Figure 247: Curve for unweighted 2 class random forest with depth 10 and 70% features

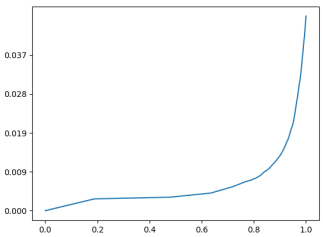

Figure 248: Curve for unweighted 2 class random forest with depth 10 and 70% features

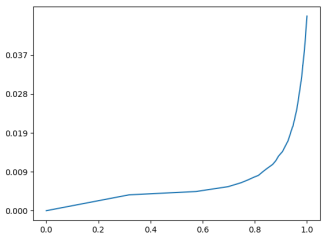

Figure 249: Curve for unweighted 2 class random forest with depth 10 and 100% features

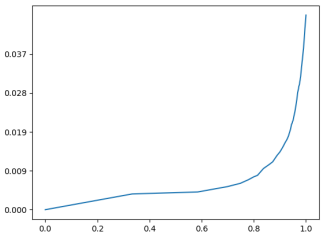

Figure 250: Curve for unweighted 2 class random forest with depth 10 and 100% features

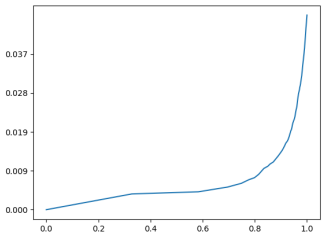

Figure 251: Curve for unweighted 2 class random forest with depth 10 and 100% features

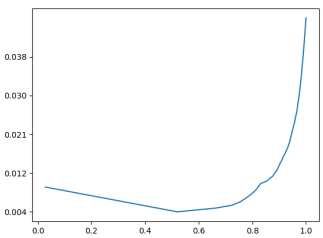

Figure 252: Curve for unweighted 2 class random forest with depth 20 and 30% features

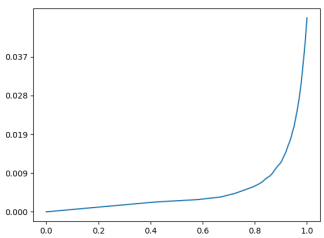

Figure 253: Curve for unweighted 2 class random forest with depth 20 and 30% features

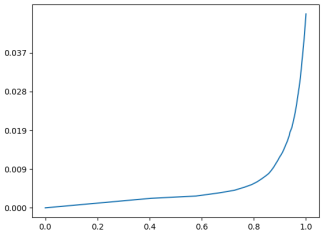

Figure 254: Curve for unweighted 2 class random forest with depth 20 and 30% features

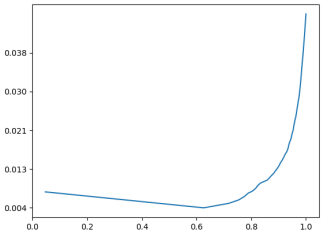

Figure 255: Curve for unweighted 2 class random forest with depth 20 and 50% features

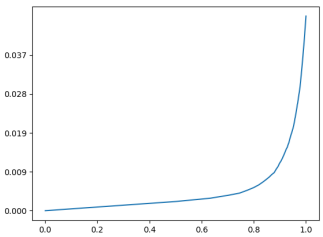

Figure 256: Curve for unweighted 2 class random forest with depth 20 and 50% features

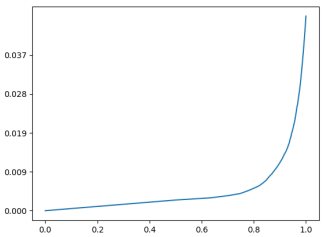

Figure 257: Curve for unweighted 2 class random forest with depth 20 and 50% features

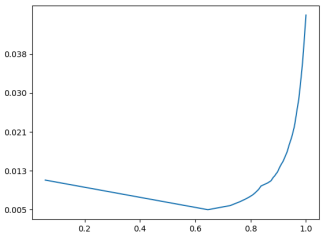

Figure 258: Curve for unweighted 2 class random forest with depth 20 and 70% features

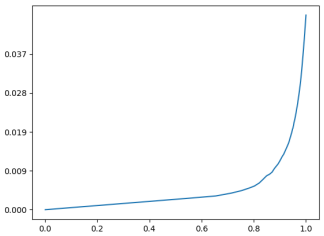

Figure 259: Curve for unweighted 2 class random forest with depth 20 and 70% features

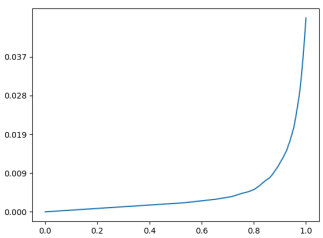

Figure 260: Curve for unweighted 2 class random forest with depth 20 and 70% features

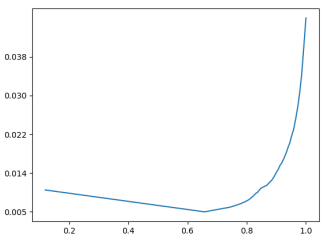

Figure 261: Curve for unweighted 2 class random forest with depth 20 and 100% features

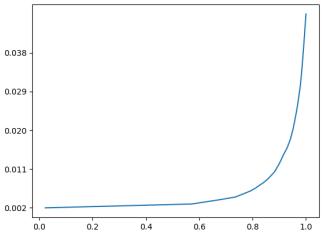

Figure 262: Curve for unweighted 2 class random forest with depth 20 and 100% features

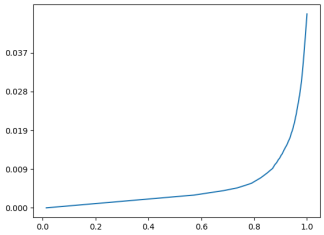

Figure 263: Curve for unweighted 2 class random forest with depth 20 and 100% features

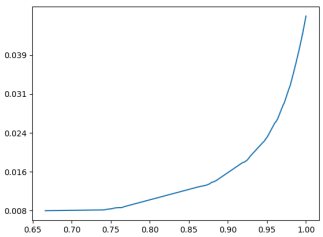

Figure 264: Curve for unweighted 2 class random forest with depth 30 and 30% features

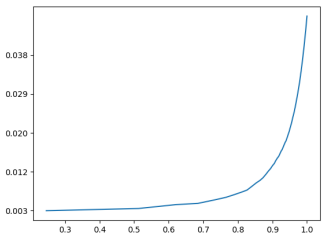

Figure 265: Curve for unweighted 2 class random forest with depth 30 and 30% features

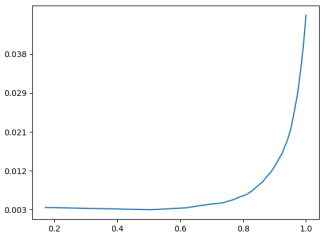

Figure 266: Curve for unweighted 2 class random forest with depth 30 and 30% features

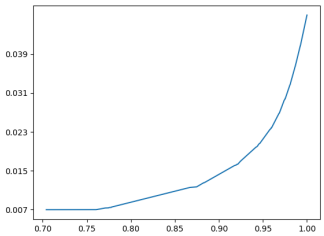

Figure 267: Curve for unweighted 2 class random forest with depth 30 and 50% features

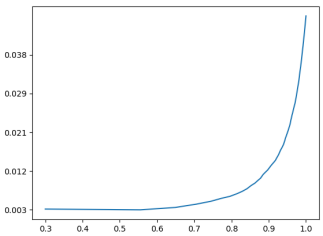

Figure 268: Curve for unweighted 2 class random forest with depth 30 and 50% features

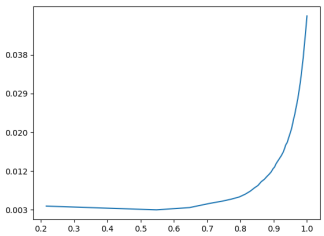

Figure 269: Curve for unweighted 2 class random forest with depth 30 and 50% features

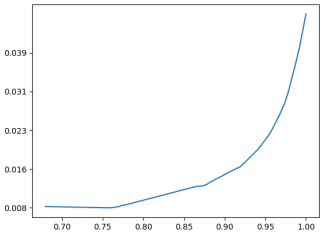

Figure 270: Curve for unweighted 2 class random forest with depth 30 and 70% features

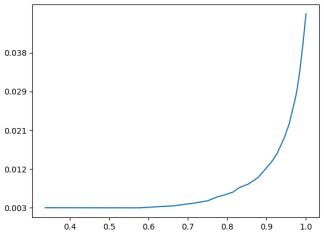

Figure 271: Curve for unweighted 2 class random forest with depth 30 and 70% features

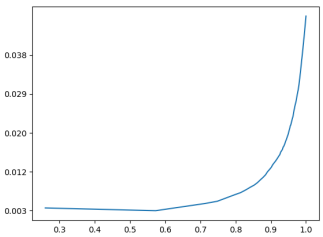

Figure 272: Curve for unweighted 2 class random forest with depth 30 and 70% features

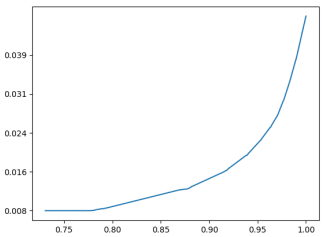

Figure 273: Curve for unweighted 2 class random forest with depth 30 and 100% features

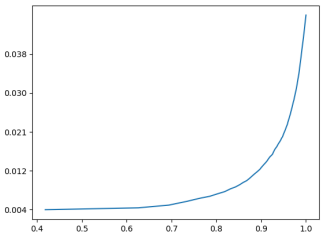

Figure 274: Curve for unweighted 2 class random forest with depth 30 and 100% features

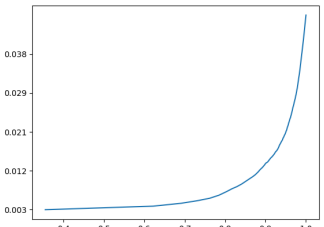

Figure 275: Curve for unweighted 2 class random forest with depth 30 and 100% features

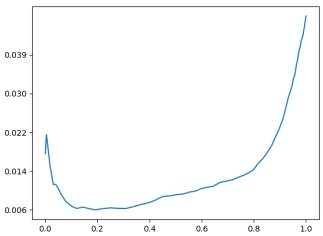

Figure 276: Curve for balanced 2 class random forest with depth 5 and 30% features

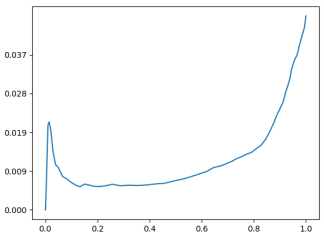

Figure 277: Curve for balanced 2 class random forest with depth 5 and 30% features

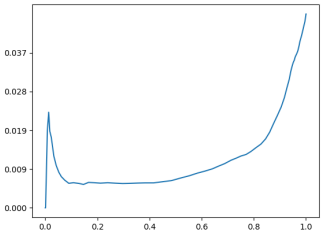

Figure 278: Curve for balanced 2 class random forest with depth 5 and 30% features

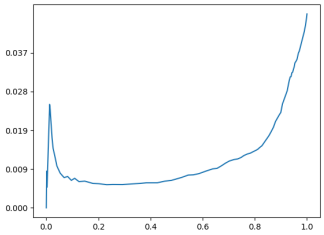

Figure 279: Curve for balanced 2 class random forest with depth 5 and 50% features

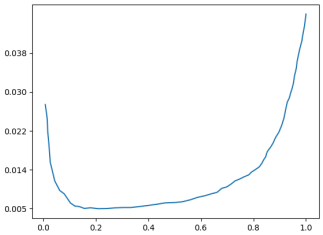

Figure 280: Curve for balanced 2 class random forest with depth 5 and 50% features

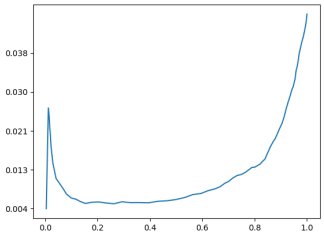

Figure 281: Curve for balanced 2 class random forest with depth 5 and 50% features

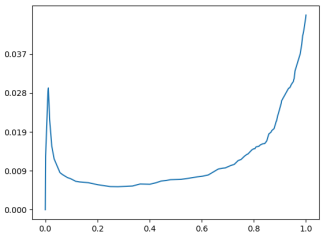

Figure 282: Curve for balanced 2 class random forest with depth 5 and 70% features

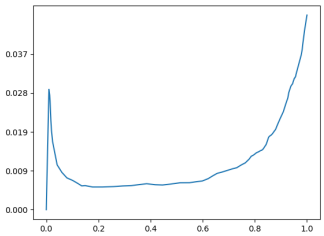

Figure 283: Curve for balanced 2 class random forest with depth 5 and 70% features

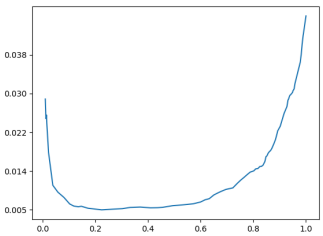

Figure 284: Curve for balanced 2 class random forest with depth 5 and 70% features

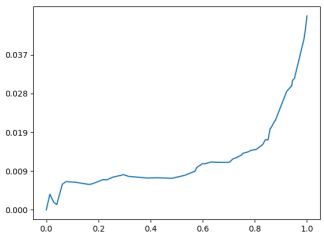

Figure 285: Curve for balanced 2 class random forest with depth 5 and 100% features

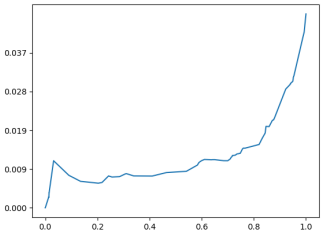

Figure 286: Curve for balanced 2 class random forest with depth 5 and 100% features

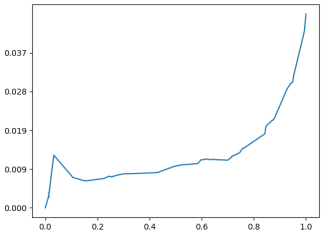

Figure 287: Curve for balanced 2 class random forest with depth 5 and 100% features

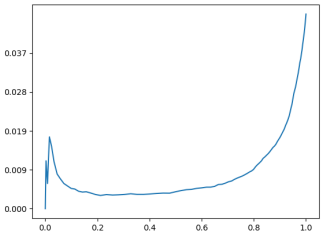

Figure 288: Curve for balanced 2 class random forest with depth 10 and 30% features

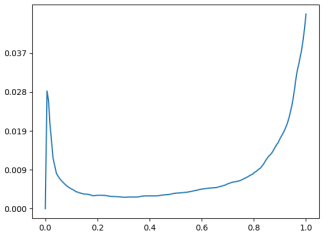

Figure 289: Curve for balanced 2 class random forest with depth 10 and 30% features

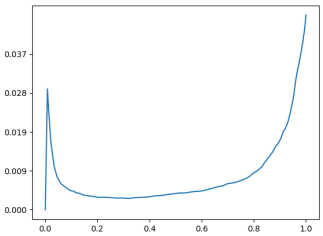

Figure 290: Curve for balanced 2 class random forest with depth 10 and 30% features

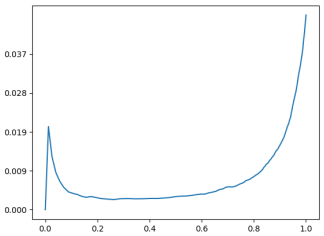

Figure 291: Curve for balanced 2 class random forest with depth 10 and 50% features

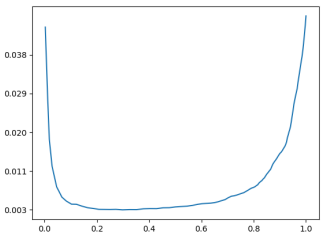

Figure 292: Curve for balanced 2 class random forest with depth 10 and 50% features

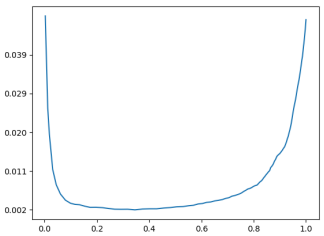

Figure 293: Curve for balanced 2 class random forest with depth 10 and 50% features

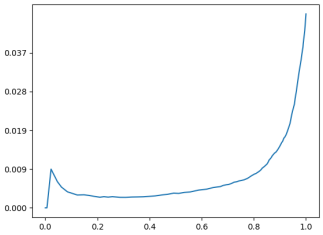

Figure 294: Curve for balanced 2 class random forest with depth 10 and 70% features

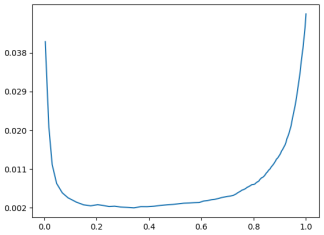

Figure 295: Curve for balanced 2 class random forest with depth 10 and 70% features

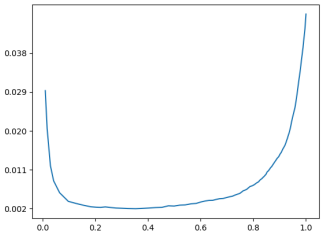

Figure 296: Curve for balanced 2 class random forest with depth 10 and 70% features

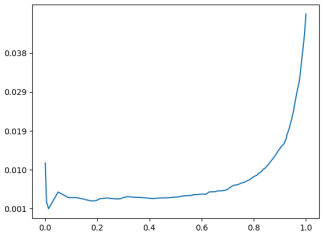

Figure 297: Curve for balanced 2 class random forest with depth 10 and 100% features

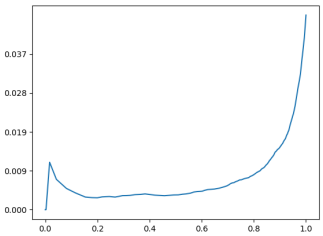

Figure 298: Curve for balanced 2 class random forest with depth 10 and 100% features

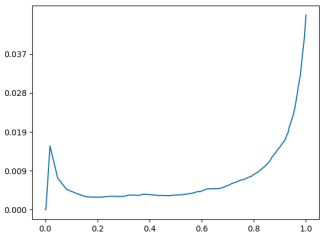

Figure 299: Curve for balanced 2 class random forest with depth 10 and 100% features

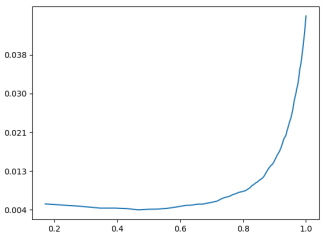

Figure 300: Curve for balanced 2 class random forest with depth 20 and 30% features

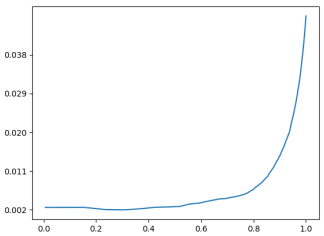

Figure 301: Curve for balanced 2 class random forest with depth 20 and 30% features

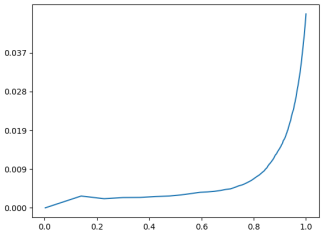

Figure 302: Curve for balanced 2 class random forest with depth 20 and 30% features

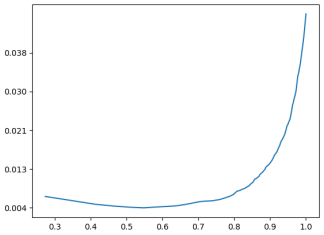

Figure 303: Curve for balanced 2 class random forest with depth 20 and 50% features

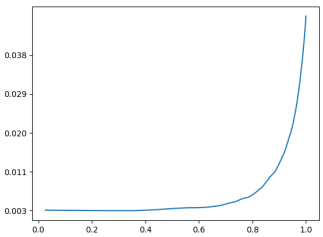

Figure 304: Curve for balanced 2 class random forest with depth 20 and 50% features

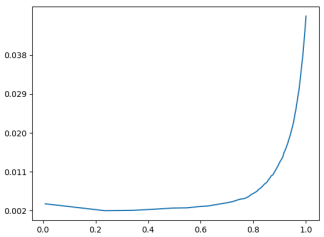

Figure 305: Curve for balanced 2 class random forest with depth 20 and 50% features

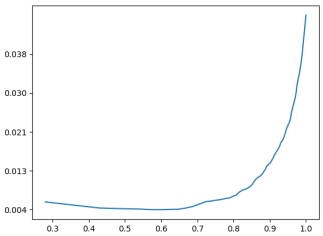

Figure 306: Curve for balanced 2 class random forest with depth 20 and 70% features

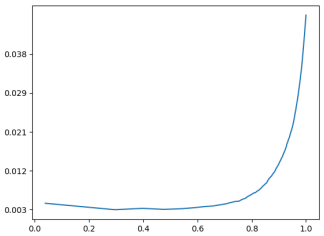

Figure 307: Curve for balanced 2 class random forest with depth 20 and 70% features

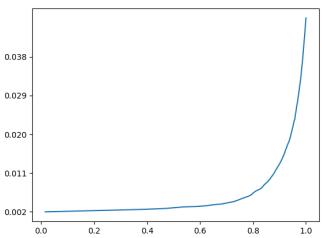

Figure 308: Curve for balanced 2 class random forest with depth 20 and 70% features

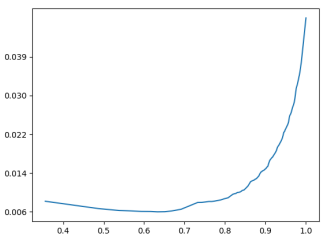

Figure 309: Curve for balanced 2 class random forest with depth 20 and 100% features

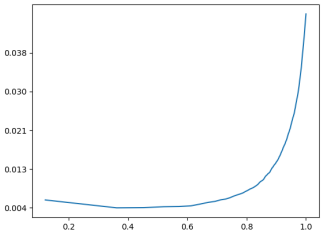

Figure 310: Curve for balanced 2 class random forest with depth 20 and 100% features

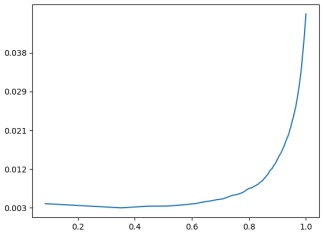

Figure 311: Curve for balanced 2 class random forest with depth 20 and 100% features

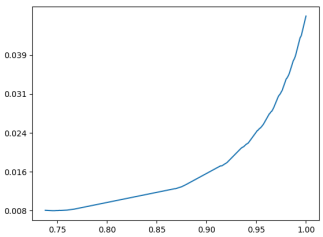

Figure 312: Curve for balanced 2 class random forest with depth 30 and 30% features

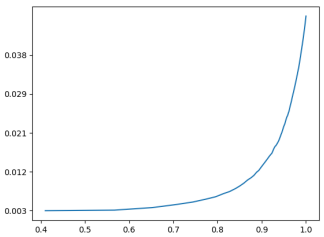

Figure 313: Curve for balanced 2 class random forest with depth 30 and 30% features

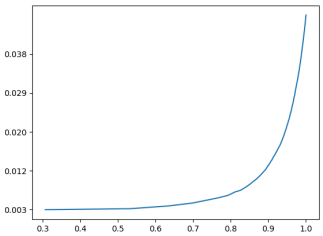

Figure 314: Curve for balanced 2 class random forest with depth 30 and 30% features

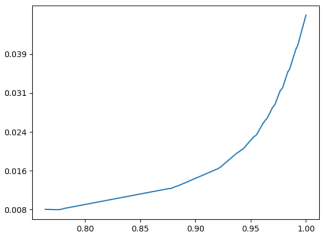

Figure 315: Curve for balanced 2 class random forest with depth 30 and 50% features

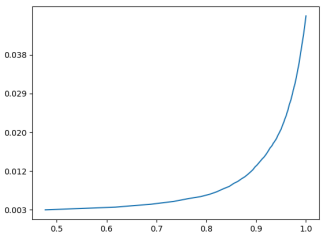

Figure 316: Curve for balanced 2 class random forest with depth 30 and 50% features

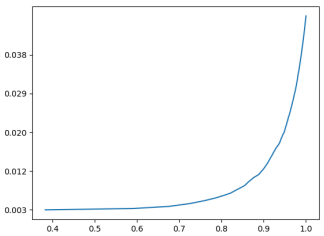

Figure 317: Curve for balanced 2 class random forest with depth 30 and 50% features

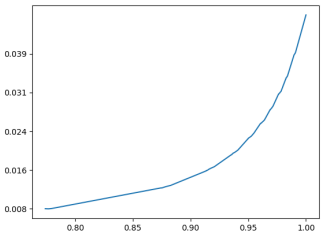

Figure 318: Curve for balanced 2 class random forest with depth 30 and 70% features

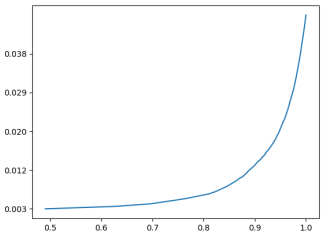

Figure 319: Curve for balanced 2 class random forest with depth 30 and 70% features

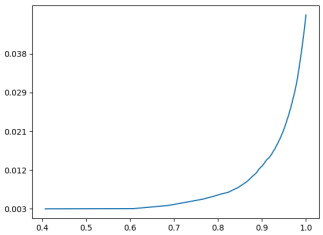

Figure 320: Curve for balanced 2 class random forest with depth 30 and 70% features

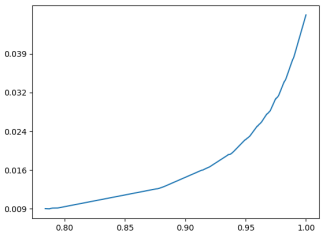

Figure 321: Curve for balanced 2 class random forest with depth 30 and 100% features

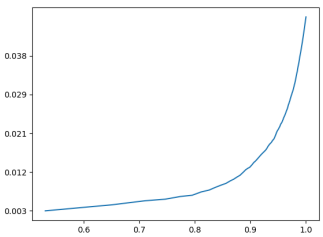

Figure 322: Curve for balanced 2 class random forest with depth 30 and 100% features

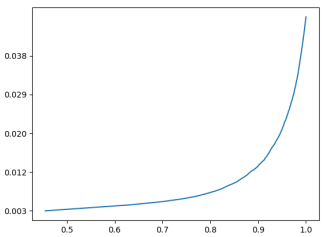

Figure 323: Curve for balanced 2 class random forest with depth 30 and 100% features

References

[1] American Medical Association (AMA). *2018 AMA Prior Authorization (PA) Physician Survey*. Accessed: 06/12/2020. 2019. URL: https://www.ama-assn.org/system/files/2019-02/prior-auth-2018.pdf.

[2] AJC. *Anthem's emergency room coverage denials draw scrutiny*. Accessed: 06/11/2020. URL: https://www.ajc.com/news/state--regional-govt--politics/anthem-emergency-room-coverage-denials-draw-scrutiny/sWT8ts3TYv6vNjrEN99kdO/.

[3] Robery C. Amland and Kristin E. Hahn-Cover. "Clinical Decision Support for Early Recognition of Sepsis". In: *American Journal of Medical Quality* (2016).

[4] American Physical Therapy Association. *What is Utilization Management?* Accessed: 12/18/2019. URL: https://www.apta.org/WhatIsUM/.

[5] Howard L. Bailit and Cary Sennett. "Utilization management as a cost-containment strategy". In: *Health Care Financing Review* (1991).

[6] S Biondo et al. "Prognostic Factors for Mortality in Left Colonic Peritonitis: A New Scoring System". In: *Journal of the American Medical Informatics Association* (2016).

[7] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] Carl R. Boyd, Mary Ann Tolson, and Wayne S. Copes. "Evaluating Trauma Care: The TRISS Method". In: *The Journal of Trauma* (1987).

[9] Leo Breiman. "Random Forests". In: *Machine Learning* (2001).

[10] Leo Breiman et al. *Classification and regression trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[11] *Case Advisor Services*. Accessed: 08/07/2020. 2019. URL: https://www.optum360.com/content/dam/optum3/optum/en/resources/PDFs/ai-powered-service-summary-flyer-hr.pdf.

[12] Utilization Review Accreditation Commission. *Health Utilization Management Accreditation*. Accessed: 12/18/2019. URL: https://www.urac.org/programs/health-utilization-management-accreditation.

[13] J. S. Cramer. "The origins of logistic regression". In: *Tinbergen Institute Discussion Paper* (2002).

[14] *Data Anonymization and De-Identification:Challenges and Options August 2019*. Accessed: 08/05/2020. 2019. URL: https://privacy.uw.edu/wp-content/uploads/2019/08/finaldad-web-brand.pdf.

[15] *Decision Trees: A simple way to visualize a decision*. Accessed: 06/16/2020. 2018. URL: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb.

[16] Deloitte. *2020 Global Health Care Outlook*. Accessed: 06/08/2020. URL: https://www2.deloitte.com/global/en/pages/life-sciences-and-healthcare/articles/global-health-care-sector-outlook.html.

[17] Healthcare Dive. *Anthem ER policy could deny 1 in 6 visits if universally adopted, JAMA study warns*. Accessed: 06/11/2020. URL: https://www.healthcaredive.com/news/anthem-er-policy-could-deny-1-in-6-visits-if-universally-adopted-jama-stud/540241/.

[18] *File:Exam pass logistic curve.jpeg*. Accessed: 06/16/2020. 2015. URL: https://commons.wikimedia.org/wiki/File:Exam_pass_logistic_curve.jpeg.

[19] Yoni Halpern et al. "Electronic medical record phenotyping using the anchor and learn framework". In: *Journal of the American Medical Informatics Association* (2016).

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning*. Springer, 2001.

[21] Tin Kam Ho. "Random Decision Forests". In: *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (1995).

[22] Institute of Medicine. *Controlling Costs and Changing Patient Care?: The Role of Utilization Management*. National Academies Press, 1989.

[23] Brent D. Nelson et al. "Computerized Decision Support for Concurrent Utilization Review Using the HELP System". In: *Journal of the American Medical Informatics Association* (1994).

[24] *Precision utilization management: How artificial intelligence is reshaping UM*. Accessed: 08/07/2020. 2017. URL: https://www.beckershospitalreview.com/healthcare-information-technology/precision-utilization-management-how-artificial-intelligence-is-reshaping-um.html.

[25] Joaquin Quiñonero-Candela et al. *Dataset Shift in Machine Learning*. Neural Information Processing Series, 2009.

[26] Charlotte Quintens et al. "Development and implementation of "Check of Medication Appropriateness" (CMA): advanced pharmacotherapy-related clinical rules to support medication surveillance". In: *BMC Med Inform Decis Mak* (2019).

[27] Erik R. Ranschaert, Sergey Morozov, and Paul R. Algra. *Artificial Intelligence in Medical Imaging*. Springer, 2019.

[28] *Reducing Healthcare Costs with Data-driven Utilization Management*. Accessed: 08/07/2020. URL: https://www.exlservice.com/reducing-healthcare-costs-with-datadriven-utilization-management.

[29] Committee for a Responsible Federal Budget. *American Health Care: Health Spending and the Federal Budget*. Accessed: 12/18/2019. May 2018. URL: https://www.crfb.org/papers/american-health-care-health-spending-and-federal-budget.

[30] D.J. Sakrison. "Estimating parameters of covariance function with modified stochastic approximation". In: *International Journal of Engineering Science* (1965).

[31] Chi-cheng Sun and Polun Chang. "Automatic Appropriateness-Evaluation and Consultation-Suggestion of Antibiotics Usage via Mining of Previous Prescription Data in Hospital Information System". In: *AMIA Annu Symp Proc* (2005).

[32] *Utilization Management*. Accessed: 08/07/2020. 2020. URL: https://nationalmedicalreviews.com/services/utilization-management/.

[33] Vartan M. Vartanians et al. "Increasing the Appropriateness of Outpatient Imaging: Effects of a Barrier to Ordering Low-Yield Examinations". In: *Radiology* (2010).

[34] *Why random forests outperform decision trees*. Accessed: 06/16/2020. 2018. URL: https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5.