Machine Learning for Downstream Oil and Gas Refineries: Applications for Solvent Deasphalting

by

Christian Dowell

B.S., University of California, Davis (2013)

Submitted to the System Design and Management Program
in partial fulfillment of the requirements for the degree of

Master of Science in Engineering and Management

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author .....
System Design and Management Program
July 30, 2021

Certified by .....
Alexandre Jacquillat
Associate Professor of Operations Research and Statistics
Thesis Supervisor

Accepted by .....
Joan Rubin
Executive Director, System Design and Management Program

Machine Learning for Downstream Oil and Gas Refineries: Applications for Solvent Deasphalting

by

Christian Dowell

Submitted to the System Design and Management Program
on July 30, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering and Management

Abstract

This thesis seeks to provide continuous Deasphalted oil (DAO) yield estimations for a Solvent deasphalting (SDA) unit by constructing modern machine learning models using data sets from a commercial downstream oil and gas refinery in the United States. These data sets include plant operating parameters and laboratory measurements for feed properties. The best machine learning model, determined via an extensive cross-validation procedure, exhibits high out-of-sample R^2 values of 0.76. Furthermore, this predictive machine learning model is incorporated into a linear optimization framework to enhance crude oil purchasing decisions for a downstream refinery. Results suggest that the proposed approach, combining predictive and prescriptive analytics, can result in significant profitability gains estimated at $730,000 annually. The results of this model can be utilized for more accurate plant monitoring within oil & gas downstream refineries, as well as improved decision making by oil and gas planning professionals.

Thesis Supervisor: Alexandre Jacquillat
Title: Associate Professor of Operations Research and Statistics

Acknowledgments

This thesis is the culmination of a one-year journey through a two-year program at MIT which would have not been possible without the extensive support of colleagues, friends, and family. There are more people than I have space to list here - I owe all of you a tremendous debt.

I'd like to thank my research supervisor, Professor Alexandre Jacquillat, for supporting me on a journey through the energy domain. Professor Jacquillat was the instructor for my first analytics course at MIT and after demonstrating the success of analytics in other domains, he sparked a flame for finding ways to apply analytical approaches to my own domain. Professor Jacquillat's novel thinking, probing questions, and business acumen inspired many processes described in this thesis. I'm extremely grateful for his constant push to improve and his patience with my obtuseness.

I would like to thank the 2020 MIT System Design and Management cohort for the community, trust, and development of relationships over this academic year. I will cherish these friendships after graduation.

The unwavering support of my wife while at MIT has made it possible for me to complete this degree. She has been an incredible support for me while I have been here. She is the rock that kept me steady throughout this entire year.

Lastly, I'd like to thank my son Milo, who helped me to have fun in an otherwise stressful year.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms

DAO Deasphalted oil. 2, 5, 15, 16, 18, 20, 21, 22, 23, 24, 36, 54, 74, 75, 77, 81, 82, 83, 84, 85

ET Extremely Randomized Trees. 6, 9, 10, 39, 45, 46, 48, 71, 72, 73, 82

KNN K-Nearest Neighbors. 6, 9, 10, 11, 29, 39, 40, 41, 43, 48, 67, 68, 69, 70, 72, 73

MAE Mean Absolute Error. 35, 36

RF Random Forest. 6, 9, 10, 39, 44, 45, 46, 48, 70, 71, 72, 73, 74, 82

RMSE Root Mean Squared Error. 35, 36, 37, 41, 67

SDA Solvent deasphalting. 2, 5, 9, 11, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 51, 54, 56, 59, 61, 73, 77, 81, 82, 83, 84

SHAP Shapley Additive Explanations. 73, 82

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Prelude

Machine learning analytics have experienced a resurgence in popularity, stemming from both recent technological developments and improved access to large data sets. Coincident with its prevalence, human competency in machine learning continues to grow with new technology applications occurring rapidly. The oil and gas industry continues to undergo a digital transformation, and, along with it, there continues to be rapid growth in the number of machine learning applications including price forecasting, asset optimization, and predictive maintenance. These firms can be further motivated to do this due to recent research that indicates firms with increasing spending on digital infrastructure experience improved firm performance (Brynjolfsson & Hitt, 1996). Additionally, recent research has found that firms that utilize predictive analytics experience improved productivity (Brynjolfsson, Jin, & McElheran, 2021).

1.2 Motivations

This thesis seeks to provide continuous Deasphalted oil (DAO) yield estimations for a Solvent deasphalting (SDA) unit by utilizing historical operating data and modern machine learning techniques. This stems from experience with issues utilizing physics-based models as discussed in Section 1.7 and potential economic benefits to existing

linear programming models as discussed in Section 1.4.

As part of this thesis, I seek to address several questions to pursue this problem.

Primary Research Questions

1. How can machine learning based models augment their physics based counterparts, to improve the accuracy of SDA DAO yields?
2. Which features are the most important in impacting SDA DAO yields?
3. How can we incorporate predictive machine learning analytics into an optimization framework to produce tangible business value?

1.3 Industry Overview

It is convenient to split the oil and gas industry into three sectors: upstream, midstream, and downstream as seen in Figure 1-1.

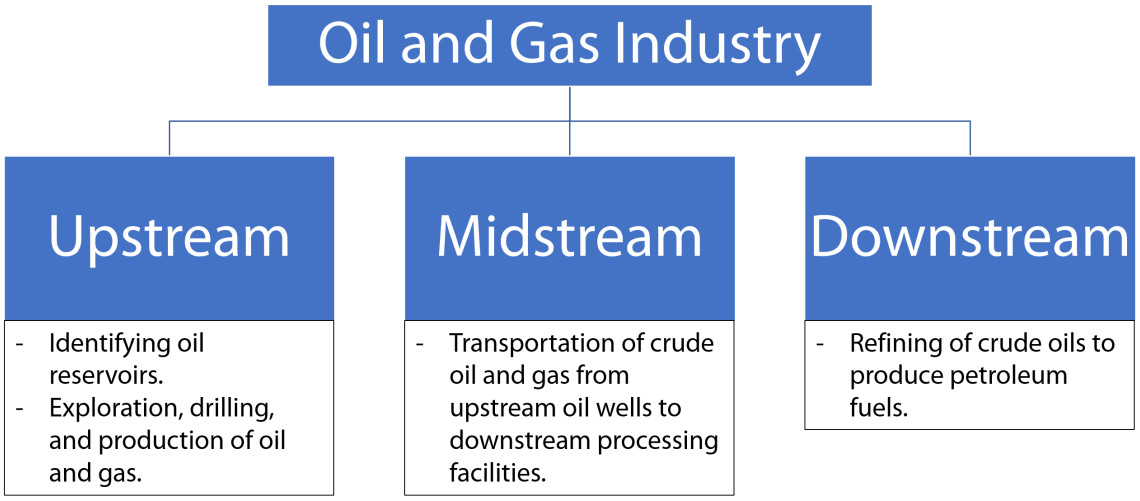

Figure 1-1: L1 decomposition of oil and gas industry.

The upstream oil and gas sector focuses on identifying oil reservoirs below the Earth's surface, producing crude oil and gas. While natural gas can provide significant value, this thesis is focused primarily on the crude oil value chain, and therefore we eschew further discussion of the natural gas produced by reservoirs. After locating

potential crude oil reserves, detailed planning, exploration, drilling, and production of oil and gas follows within the upstream industry.

The midstream sector is responsible for the transportation of crude oil produced from upstream oil wells to downstream processing facilities. This transportation can take place via pipelines, tankers, trucks, or rail cars. Additionally, the midstream sector is responsible for the storage and marketing of petroleum commodities, including crude oil and natural gas.

The downstream sector includes the refining of crude oils to produce lower molecular weight petroleum fuels. These fuels are then blended to result in finished petroleum products for mass consumption. These petroleum products include motor gasoline, jet, diesel, asphalt, and lubricants. This sector comprises petrochemical plants, distribution operations, retail centers, and oil refineries (Pandey, Rastogi, Kainkaryam, Bhattacharya, & Saputelli, 2020).

In oil and gas refineries, high-value petroleum products such as gasoline, jet fuel, and diesel can be produced by first distilling crude oil. This distillation produces gas oil, distilled oil, and atmospheric residue. Further distillation of the atmospheric residue under reduced pressure creates vacuum gas oil, distilled oil, and vacuum residue (Lee et al., 2014).

This vacuum residue, often called "resid" for short, contains asphaltene compounds that contain large amounts of heavy metal, sulfur, and nitrogen (Lee et al., 2014). Additionally, it is very viscous and cannot readily be used as transportation fuels. While these heavy-end materials can be upgraded, the conversion process can be very costly due to the need for a large amount of hydrogen for hydrotreating or hydrocracking techniques. An alternative approach is to utilize SDA to remove the fractions most responsible for the low quality in advance of hydrotreating or hydrocracking (Brons & Yu, 1995).

1.4 Refinery Economics

A full review of refinery economics is outside the scope of this thesis. Nevertheless, it is important to briefly discuss a selection of the economic drivers as they pertain to this work and further provide motivation and context for the machine learning model.

The purchase of feedstock, crude oil and other blend stocks, accounts for about 85% of a refinery's operating cost (Robinson, 2007). The sale of refined products constitute the revenue for a refinery. Generally, refineries operate to make as much of the refined high-value light products (gasoline, jet fuel, and diesel) as possible, with other products acting similar to by-products (*McKinsey Energy Insights, Products*, n.d.). This is true in the case of SDA, where the finished products from downstream units that produce transportation fuels provide more revenue than the asphalt pitch from an SDA.

To maximize refinery profitability, many refiners utilize linear programming optimization tools to make near-term commercial decisions. These tools are used to find the crude and product slate which maximizes the profitability of the refinery, subject to market and operational constraints (*McKinsey Energy Insights, Optimization*, n.d.). Amongst other decision variables, these linear programs optimize decision variables for:

- Feedstock selection - which crude oils to buy and in what quantities.
- Product slate - which refined products to make, how much of each product to make and the quality of each product.

Many linear programs utilize crude oil data to estimate feed parameters. Additionally, heuristics or physics-based models can be used to estimate intermediate parameters within the refining process. In the case of SDA, a physics-based models can be used to estimate intermediate yield values which in turn can be used to estimate the amount of DAO produced by the unit. However, in the case where a heuristic or physics-based model is inaccurate, it can lead to poor estimation of intermediate products leading to suboptimal feedstock and product slate selection.

1.5 Crude Oil Characteristics

It is important to discuss the properties of crude oil so as to better understand the complexity of the system under analysis. In contrast to other industries which may deal with products that are pure chemicals, crude oil is a mixture of a variety of chemical compounds called hydrocarbons.

One of the most important characteristics of crude oil is it's behavior when it's temperature increases. To illustrate this point, let us consider pure liquid water ($H_2O_{(l)}$) at atmospheric pressure. Liquid water, when brought to it's boiling point of 212 °F (100 °C) will begin to boil and, if given sufficient time and heat input to maintain the temperature, will vaporize entirely. However, bringing crude oil to the same temperature, a portion of it will boil off, and the remainder will remain liquid. This phenomenon is because crude oil can contain thousands of hydrocarbon compounds, with each of these compounds having a unique boiling point. Generally, one can observe that an increase in the number of carbon atoms in the molecular increases the boiling temperatures (Fahim, Al-Sahhaf, & Elkilani, 2010). A table illustrating this is presented in Table 1.1.

Name	Molecular Formula	Boiling Point (°C)
Methane	CH_4	-164
Pentane	C_5H_{12}	36
Octane	C_8H_{18}	125
Dodecane	$C_{12}H_{26}$	216
Triacontane	$C_{30}H_{62}$	450

Table 1.1: Boiling Point Temperatures for Selected Hydrocarbons

1.5.1 Distillation Curves

Due to the variety of compounds present in crude oil, it is useful to characterize crude oils by their boiling properties, often by their distillation (or boiling) curve. This curve presents important information of the bulk behavior of the crude oil. An example of a distillation curve can be found in Figure 1-2. As will be discussed further

in section 2.1.1, it is useful to consider portions of this distillation curve, such as the 90% recovery point, the temperature at which 90% of the volume of the oil has been boiled off.

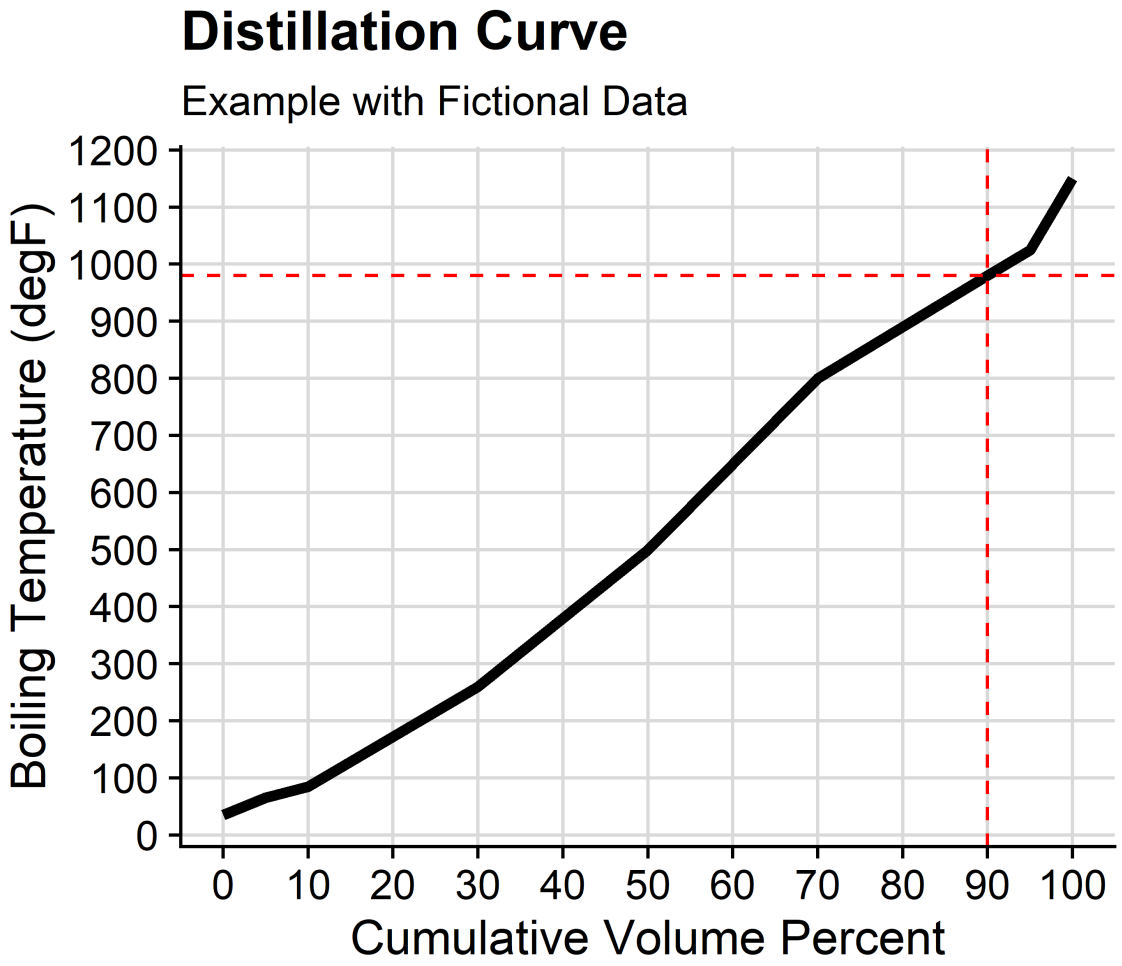

Figure 1-2: Example distillation curve with 90% recovery point noted in red

1.6 SDA Overview

A high yield of DAO can be achieved in the SDA process by removing asphaltenes based on solvent separation. Figure 1-3 shows a basic process flow diagram of an SDA unit. Vacuum residue is fed into a solvent extractor, which produces a mixture of DAO laden with solvent overhead, and asphaltene-rich pitch with a small amount

of solvent is drawn off the bottom of the extractor column. The mixture of DAO and solvent is then separated in a DAO/solvent separator to produce DAO, and the recovered solvent is then recycled to the extractor. Any remaining solvent is then removed in a DAO stripper utilizing stripping steam. In the asphalt stripper, a small amount of residual solvent is separated for recycling, and a concentrated asphaltene pitch is produced (Lee et al., 2014). This separation is done by solvent extraction without any chemical reaction taking place.

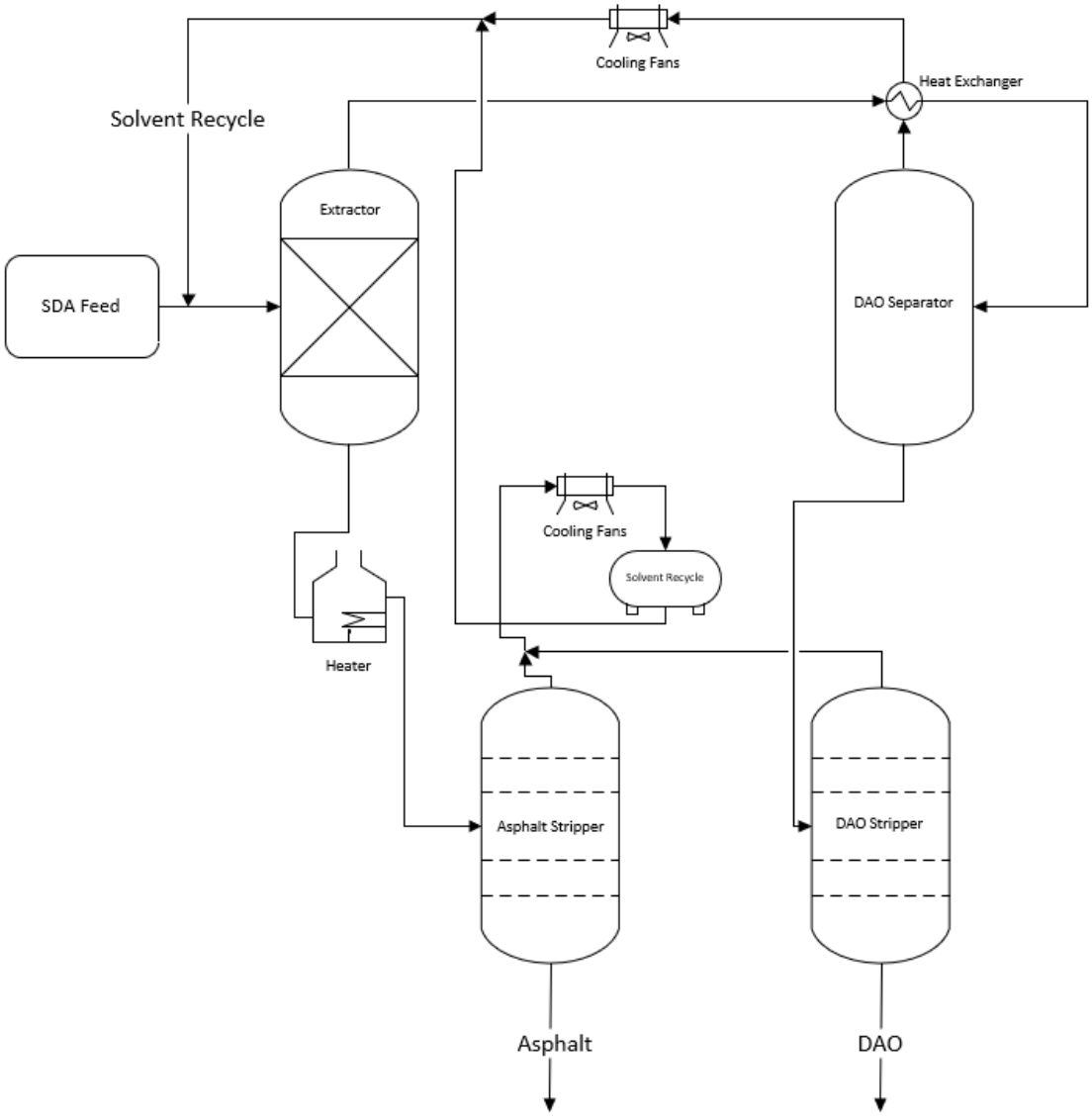

Figure 1-3: Process Flow Diagram of an SDA unit

The DAO produced from an SDA unit can then be used for further conversion

into transportation fuels and chemical raw materials through additional refinement. The asphaltene pitch is used as road-packing material for low-grade fuel and can be utilized to produce heat and hydrogen by gasification (Fahim et al., 2010).

Typically, light hydrocarbon solvents ranging from propane to hexane are used as solvents to extract the DAO product. Increasing the carbon number of the solvent will generally reduce the quality of the DAO but increases the volume of DAO that is produced due to the increased average molecular weight of the hydrocarbons that are soluble in the solvent (Pang et al., 2010; Brons & Yu, 1995). Propane and butane have widely been used to produce high-quality DAO. Refiners have utilized the use of higher carbon solvents such as pentane and hexane with increased demand for light oil and consequently effective heavy oil upgrading.

1.7 Physics Based Models for Predicting SDA DAO Yield

Studies on SDA have emphasized extracting DAO under various conditions, in an attempt to understand the impact of plant operation, solvent properties, and feed properties on DAO yields. Baek et al. (Baek, Kim, Kim, & Hong, 1993) investigated the effects of both temperature and pressure on DAO yield, finding DAO yield to vary significantly with varying temperature and pressure. This was further corroborated by Ng who noted that DAO yields decreased with increasing temperature. This was attributed to higher temperatures increasing the difference in solubility parameters between the solvent and feedstock (Ng, 1997).

Gillis and Tine took this further, studying the effect of solvent-to-oil ratio and temperature on DAO yield (Gillis & Tine, 1998). This work noted that increasing solvent-to-oil ratio at the same extractor overhead temperature increased the DAO yield.

Feed properties have also been found to affect DAO yields. Literature suggests that increasing feed density will decrease DAO yields (Maples, 2008). Notably, how-

ever, this work did not incorporate the effects of extractor temperatures, pressures, and solvent-to-oil ratios discussed above which also impact DAO yields.

Existing literature provides an excellent foundation for analytical SDA DAO yield models. Yet, there are still gaps that make it challenging to apply these models commercially.

First, literature on physics-based models do not look at the refinery system, instead focusing exclusively on the SDA itself. Many physics-based models do not consider upstream plant operation, how this operation affects SDA feed properties and it's corresponding effects on the DAO yield. Without analysis of the refining system, these models will not be able to accurately quantify expected DAO yields.

Second, the overwhelming majority of physics-based models are developed from laboratory settings in which limited data is collected. This is primarily due to feasibility and cost limitations, especially in the context of processing heavy residuum oils. The limited data collected can prove useful for observing general trends and inspiring future work. However, building an accurate model for prediction of DAO yields requires a large amount of data which may only be feasibly obtained at commercial scale.

Third, individual studies have focused on one, or in some cases a few, parameters at a time and how these impact DAO yields. However, as there are a plethora of features which can impact DAO yield, there is no well-established holistic model to incorporate plant operating parameters, solvent properties, and feed properties to predict DAO yield.

To address some of these issues, private industrial firms have developed proprietary models to predict DAO yield as a function of plant operating parameters, solvent properties, and feed properties. However, just like the literature-based counterparts, many of these models rely on laboratory data for SDA feed or asphalt pitch. A hindrance to the success and utility of these DAO models in commercial applications is the challenge of obtaining accurate, reliable, and consistent laboratory data. Due to the nature of vacuum residue, it is often challenging to routinely sample SDA service streams due to the high viscosity and high melting point of the asphaltene.

Therefore, models which require large amounts of physical laboratory data often run into practical issues in an oil and gas refinery when samples cannot be reliably obtained.

1.8 Structure

This thesis seeks to provide continuous DAO yield estimations for a SDA unit by constructing modern machine learning models using data sets from a commercial refinery in the United States. These data sets include plant operating parameters and laboratory measurements for feed properties. These predictive machine learning models exhibit high out-of-sample R^2 values of 0.76. Furthermore, this predictive machine learning model is incorporated into a linear optimization framework for a hypothetical downstream refinery, improving profitability by an estimated $730,000 annually.

This thesis is structured as follows:

- The remainder of Chapter 1 will provide an introduction to the oil and gas industry, with an emphasis on SDA, as well as important facets of crude oil characteristics that will prove important in future chapters.
- Chapter 2 provides descriptive information on the datasets and features as well as a brief discussion on the various scaling methods utilized for this analysis.
- Chapter 3 describes the various predictive modeling methods used in this thesis for continuous estimation of DAO yields.
- Chapter 4 discusses applying the aforementioned predictive modeling techniques into an optimization framework.
- Chapter 5 discusses pertinent results from the modeling work completed in Chapters 3 and 4.
- Chapter 6 draws conclusions and presents opportunities for future work.

Chapter 2

Data Sets and Features

The data sets compiled for this machine learning problem come from a commercial SDA unit in the United States. The type of solvent used in this unit is maintained at consistent concentrations (i.e., the solvent is not changing significantly throughout operation). The exact solvent utilized will not be disclosed due to confidential classifications. The data sets include a comprehensive compilation of daily process variables and laboratory results. Additionally, laboratory data was taken from the upstream vacuum distillation unit.

2.1 Data Sets

2.1.1 Predictive Modeling

Features from SDA Process Instrumentation

SDA units, like most refinery units, have multiple instruments which provide information about the temperature, pressure, and flow rate for various streams. Within process equipment, including those presented in Figure 1-3, additional instrumentation can be added to monitor for temperature, pressure, and level within the equipment. Table 2.1 contains information regarding the features from SDA process instrumentation considered in this analysis.

Feature	Description	Units
Solvent to Feed Ratio	Ratio of Solvent to SDA Feed	$\frac{bpd}{bpd}$
Total Recycled Solvent	Solvent recycle back to Extractor	bpd
Extractor Top Temperature	Temperature of the top stream of the Extractor	°F
DAO Stripper Bottoms Temperature	Temperature of the Bottom of the DAO Stripper	°F
DAO Separator Bottoms Temperature	Temperature of the Bottom of the DAO Separator	°F
DAO Separator Top Pressure	Pressure of the Top of the DAO Separator	psig

Table 2.1: Features from SDA Process Instrumentation. Reference Figure 1-3 for equipment descriptions

Features from SDA Laboratory Analysis

Similar to many other refinery units, routine laboratory testing is completed on streams to check for a variety of physical and chemical qualities. Of particular interest in our analysis are the properties of the feedstock for the SDA unit, previously denoted as "resid". As previously mentioned in Section 1.6, obtaining laboratory samples for SDA feedstock can be quite challenging. Due to a large amount of missing values for the laboratory data, feed properties have been generated utilizing steady-state process simulation. Table 2.2 contains information regarding the SDA laboratory variables considered in this analysis.

Feature	Description	Units
Asphaltene Content	Fraction of Feed determined to be Asphaltene	wt. %
Viscosity at 212 °F	Kinematic Viscosity of the Feed at 212 °F	cSt
Viscosity at 122 °F	Kinematic Viscosity of the Feed at 122 °F	cSt
Specific Gravity	Otherwise Referred to as the Relative Density, the ratio of the density of the Feed with respect to water, expressed as $\frac{\rho_{Feed}}{\rho_{H_2O}}$	Unitless

Table 2.2: Features from SDA Laboratory Analysis.

Features from Vacuum Distillation Unit Laboratory Analysis

SDA units typically receive the bottoms stream from the vacuum distillation unit. Upstream influences on the SDA unit must be considered as the operations of the upstream vacuum distillation unit can have significant impacts on the SDA. The properties of the next highest stream, referred to in this analysis as the "heavy gas oil" stream, are critical. It is crucial to evaluate the distillation properties of this heavy gas oil stream to understand how much "DAO-like" material is present in this stream. For example, a heavy gas oil stream with increasing 90% recovery point, represents a stream that has higher boiling points than those with lower 90% recovery points. Due to this increase in 90% recovery point, the heavy gas oil stream would be laden with material that, if sent to an SDA, would likely become DAO. Therefore, one would expect that an increase in the 90% recovery point of this heavy gas oil stream would result in a decrease of DAO yields at the SDA unit. The overlapping tails of heavy gas oil and the SDA feedstock are a typical result of distillation (petroleum-refining-in-nontechnical-language, 2008).

Variable	Description	Units
Heavy gas oil 90% recovery point	Temperature at which 90% of the volume of the heavy gas oil sample has vaporized	°F

Table 2.3: Feature from Vacuum Distillation Unit Laboratory Analysis

2.1.2 Exploratory Data Analysis

Correlation

Figure 2-1 displays the correlation matrix for the variables in the feature set. Readily apparent is the high value of correlation between the viscosity at 212 °F and the viscosity at 122 °F as well as the high correlation between the DAO Separator top and bottom temperatures. The high correlation between the viscosities is further discussed in Section 2.1.4.

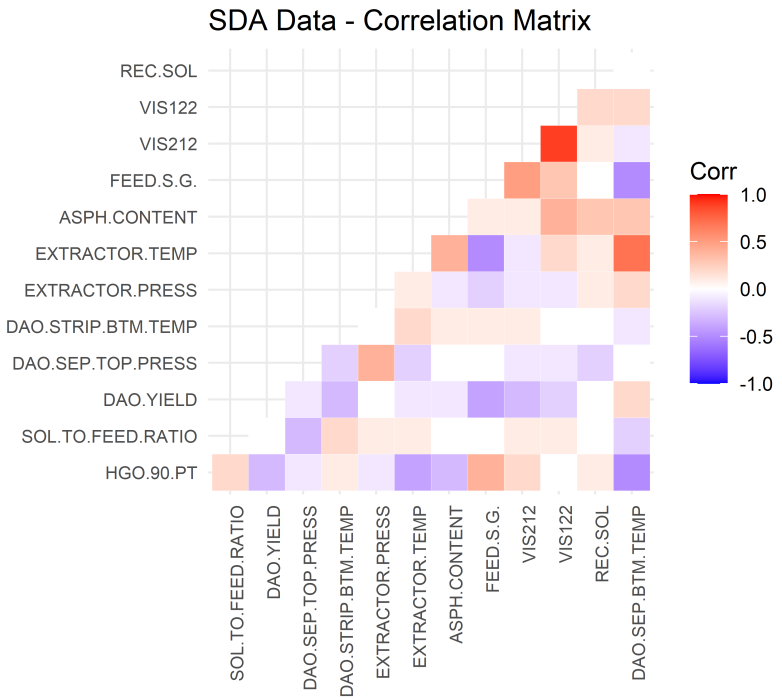

Figure 2-1: Correlation Matrix for Feature Set

Condition Number for Determination of Multicollinearity

In the context of linear regression, the condition number provides a method for detecting multicollinearity and is readily provided in modern machine learning software packages (Seabold & Perktold, 2010). Since non-experimental data will rarely be orthogonal, multicollinearity will always be present (Greene, 2003). However, at what point does this multicollinearity become a problem? Values in excess of 20 are suggested as indicative of a multicollinearity problem (Belsley, Kuh, & Welsch, 1980). This dataset has a condition number of 3.18, suggesting that multicollinearity is not strongly present in our dataset.

2.1.3 Scaling Transformations

Because these data sets have confidential classifications, they are required to be scaled with a scaling transformation in order to anonymize the true values. Standardization of a dataset is common practice for many machine learning projects. However, the choice of scaling can have ramifications on a variety of machine learning algo-

rithms. For example, methods that incorporate distance calculations, such as K-Nearest Neighbors (KNN), will weigh high magnitude features more than those with low magnitudes. Additionally, features with varying scales can have large deleterious effects on a learning algorithms' computational efficiency, as is the case with neural networks. Therefore, it is important to explore a variety of scaling transformations in order to select the appropriate transformation for the problem at hand.

Linear Transformations

Linear transformations preserve the linear relationships between variables. These transformations change the dataset and are characterized by adding, subtracting, multiplying, or dividing the variables in a dataset by a constant. Three linear transformations are considered for this analysis. A standard scaling transformation approach is to scale the variables to have a mean of zero with a unit variance with equations presented in Figure 2-2. An alternative to the zero mean, unit variance scaling is the "Min Max Scaler" which can be done by scaling according to the formulation provided in Figure 2-3. This has the benefit of putting all variables on the range of $[0, +1]$ but can be sensitive to the presence of outliers (Pedregosa et al., 2011). To combat the effects of outliers, we consider a "Robust Scaler" presented in Figure 2-4 as an alternative to both methods, which can be more robust in the presence of outliers by removing the median and dividing by the innerquartile range. All three methods are applied to the data and are presented in Figure 2-5.

Nonlinear Transformations

Nonlinear transformations change the linear relationships between variables. Two nonlinear transformations are explored utilizing three methodologies, namely the the Yeo-Johnson transformation (Yeo & Johnson, 2000), and a Quantile transformation (Pedregosa et al., 2011). It should be noted that the Box-Cox transformation (Box & Cox, 1964) was avoided as it is only applicable to positive data and cannot be applied to negative data. The Yeo-Johnson and Quantile Transformations presented in Figure 2-6.

$$z = \frac{x - \mu}{\sigma} \tag{2.1}$$

where the mean is represented as

$$\mu = \frac{1}{N}\sum_{i=0}^{N}(x_i) \tag{2.2}$$

with standard deviation defined by

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(x_i - \mu)^2} \tag{2.3}$$

where:

$x = N$ x 1 column vector in $\mathbb{R}$

$z =$ transformation of x

Figure 2-2: Standard Scaling transform for numeric variables providing a mean of 0 with unit variance

$$\phi(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2.4}$$

where:

$x = N$ x 1 column vector in $\mathbb{R}$

$\phi(x) =$ transformation of x

$\min(x) =$ minimum of x

$\max(x) =$ maximum of x

Figure 2-3: Min Max Scaling transform for numeric variables to the range [0, +1]

$$\Phi(x) = \frac{x - \tilde{x}}{IQR(x)} \tag{2.5}$$

where:

$x = N$ x 1 column vector in $\mathbb{R}$

$\tilde{x}$ = median of x

$IQR(x)$ = Innerquartile range of x (Upton & Cook, 1996)

Figure 2-4: Scaling Transform utilizing RobustScaler Method (Pedregosa et al., 2011).

2.1.4 Feature Engineering

Viscosity Index

As noted in Section 2.1.2, there is high correlation between the viscosity of the feed at 122 °F and 212 °F. As a means of addressing this high correlation, but without losing valuable data, the viscosity index (VI) of the feed was calculated consistent with ASTM D2270 (ASTM, 2016). VI measures the change in viscosity between two standard temperatures, 40 °C and 100 °C. However, because the data collected for the lower temperature viscosity was at 122 °F (50 °C) and was not at the appropriate temperature (40 °C, 104 °F), and because the calculations within ASTM D2270 require kinematic viscosity at precisely 40 °C and 100 °C, we utilize ASTM D321 (ASTM, 2020) to calculate a kinematic viscosity at 40 °C (104 °F), then calculated the viscosity index consistent with ASTM D2270.

The utility of this feature was unfortunately unsuccessful and it provided less value than using either of the provided viscosity values and therefore was dropped from the remainder of the analysis.

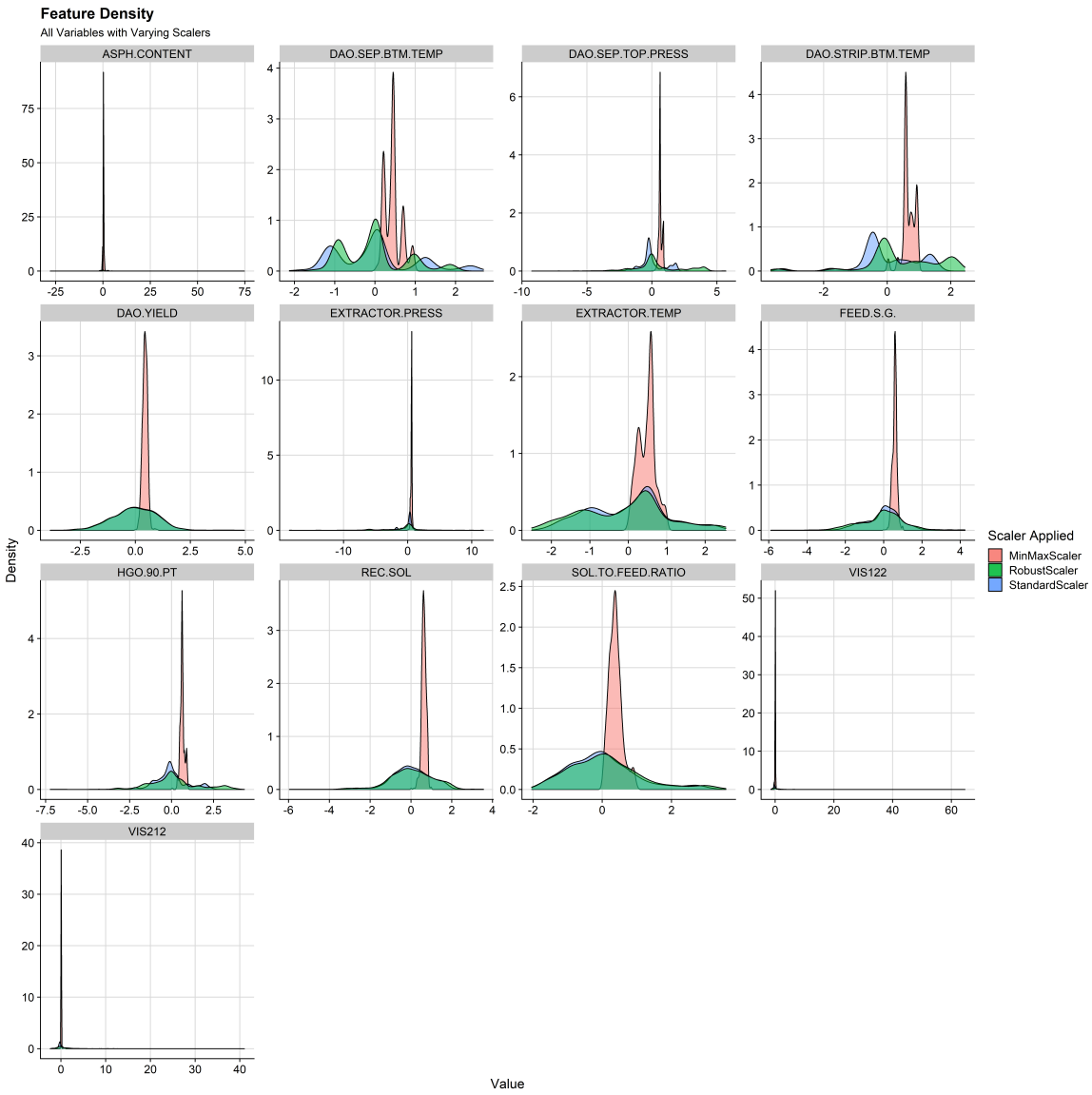

Figure 2-5: Feature Density Utilizing Standard Scaling, Min Max Scaling, and Robust Scaling.

[Feature Density Comparison using Linear Transformations.]

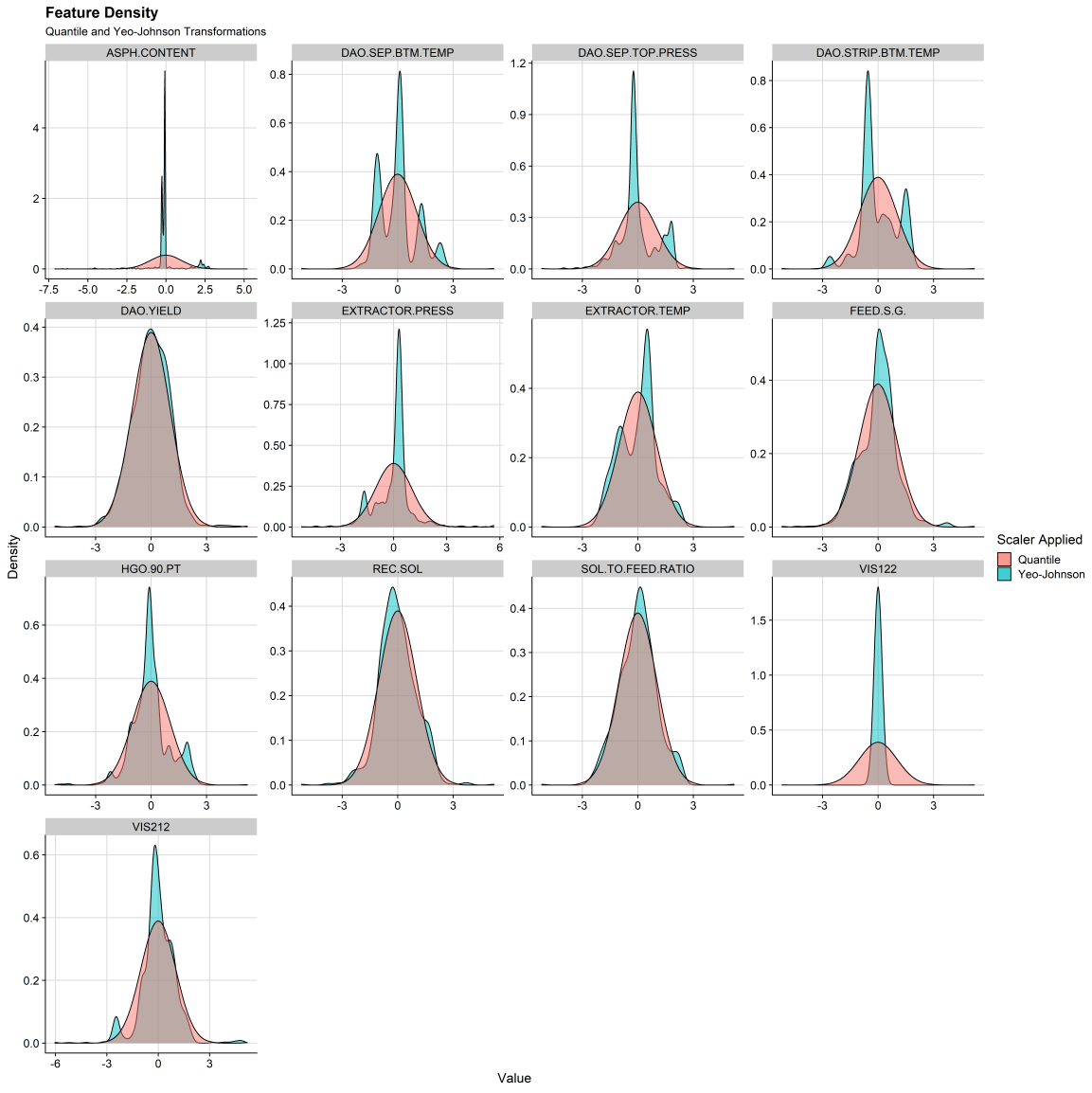

Figure 2-6: Feature Density Utilizing Quantile (Pedregosa et al., 2011) and Yeo-Johnson (Yeo & Johnson, 2000) Nonlinear Transformations.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Predictive Modeling

3.1 Problem Class

There are many different problem classes within machine learning, including, but not limited to, supervised learning, unsupervised learning, and reinforcement learning. This thesis aims to understand a supervised learning problem, where the inputs and outputs are known and can be mapped together. The problem at hand could be further classified as supervised regression where there exists a training data D_n which contains a set of pairs $(x_1, y_1), ..., (x_n, y_n)$ where x_i is a $d-$dimensional vector of real values and $y_i \in \mathbb{R}$.

3.2 Evaluation Criteria

Commonly utilized error metrics for measuring a model's predictive performance include the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination, commonly referred to as R^2. RMSE is calculated by first determining the residuals' value, averaging them, and then taking the square root of the result as seen in Equation 3.1. MAE is calculated by taking the residuals' absolute value and averaging them as seen in Equation 3.2. R^2 is obtained by calculating the sum of squared errors, dividing it by the total sum of squares, and subtracting this value from one as seen in Equation 3.3.

This thesis compares model performance utilizing RMSE as the primary error metric as this metric will heavily weight the larger errors. While errors are expected when estimating yields, very high errors will present problems when forecasting future DAO yields and erode confidence in the machine learning model. In contrast to MAE, RMSE can be less interpretable. Therefore in some cases, MAE is presented alongside RMSE to provide greater interpretability.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.1}$$

where:

n = number of observations

y_i = observed value

$\hat{y}_i$ = predicted value from machine learning model

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|)^2 \tag{3.2}$$

where:

n = number of observations

y_i = observed value

$\hat{y}_i$ = predicted value from machine learning model

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.3}$$

where:

n = number of observations

y_i = observed value

$\hat{y}_i$ = predicted value from machine learning model

$\bar{y}$ = mean value of y in the training set

3.3 Resampling Methods

Resampling methods are an incredibly useful tool in which samples from a training set are repeatedly redrawn and a model is refit on each sample in order to obtain additional information about the fitted model (Hastie, Tibshirani, & Friedman, 2001). Two resampling methods were utilized in this thesis:

- k-fold Cross Validation
- Bootstrap

3.3.1 k-fold Cross-Validation

In k-fold cross-validation, a training set is randomly partitioned into k equal sized groups, or folds. Of the k folds, a single fold is retained as the validation data for testing the model, with the remaining $k-1$ folds used as training data. The RMSE is calculated on the observations in the validation fold. This procedure is repeated such that each of the k folds are used exactly once as the validation data (Hastie et al., 2001). Then, the k-fold cross-validation RMSE is calculated as described in Equation 3.4. This process can be repeated multiple times such that each k resample represents a different portion of the dataset.

$$\text{RMSE}_{\text{CV}} = \frac{1}{k}\sum_{i=1}^{k}\text{RMSE}_i \tag{3.4}$$

In the case of this thesis, 5-fold cross-validation was used where $k = 5$ repeated three times. A graphical representation of the 5-fold cross validation process is shown in Figure 3-1.

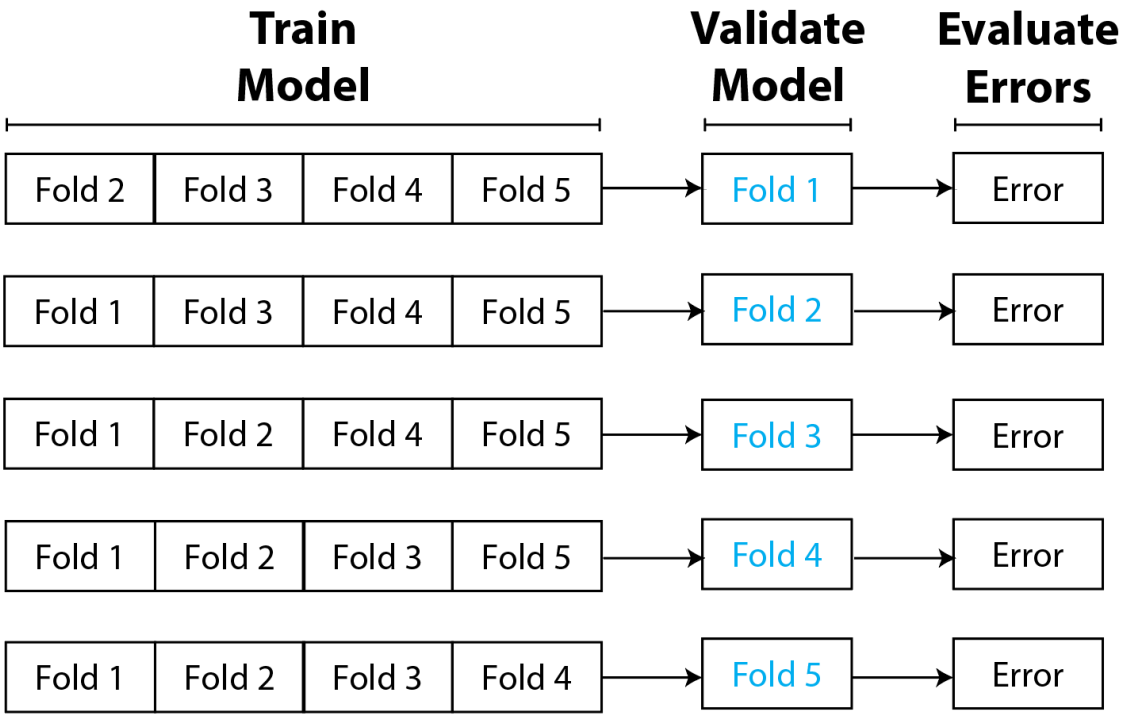

Figure 3-1: Graphical Depiction of k-fold Cross-Validation with $k = 5$.

3.3.2 Bootstrap

Bootstrap resampling is a technique used for estimating quantities of a population by averaging estimates from multiple small data samples. Samples are constructed by drawing observations from a large data sample one at a time, then returning them to the data sample after they have been chosen. This permits a given observation to be included in a given sample more than once, commonly referred to as *sampling with replacement*. The non-selected data points can be used as the validation set (James, Witten, Hastie, & Tibshirani, 2014).

3.4 Classes of Models

Multiple regression algorithms were considered for this prediction problem, including:

- Linear Regression
- K-Nearest Neighbors (KNN)
- Random Forest (RF)
- Extremely Randomized Trees (ET)
- XGBoost

Many of these algorithms contain hyperparameters, parameters which can be adjusted to control the learning process of the algorithm. The subsequent sections will discuss each model in greater detail and, if necessary, the hyperparameter tuning of that algorithm.

3.4.1 Linear Regression

Linear regression is a parametric regression model which assumes that the DAO yield function, $f(x)$, is linear. This makes it computationally inexpensive to fit the model since we need to estimate only a small number of coefficients. Additionally, the coefficients have very straightforward interpretations and it is easy to perform tests of

statistical significance on these coefficients (James et al., 2014). The linear regression model provides a useful baseline model to compare complex models as proprietary physics-based models could not be published in this thesis due to confidentiality agreements. No hyperparameters are tuned in linear regression.

A large disadvantage of linear regression is that by design it makes a strong assumption of the functional form. In this case, if the DAO yield function $f(x)$ is non linear, then the resulting model will not be able to provide a good fit.

3.4.2 K-Nearest Neighbors (KNN)

In contrast to linear regression, KNN is a non-parametric, data-driven process that makes no assumptions about the feature set, thereby providing an alternative and more flexible approach for performing regression. Given a new observation, the algorithm looks for nearby values in the training data to decide on predictions, with the final prediction given by local interpolation of the nearby values.

Hyperparameters

There are a variety of hyperparameters to tune in the KNN algorithm. These include:

- Number of neighbors to consider, k
- Distance metric
- Weighting function

Number of Neighbors to Consider k. For the number of neighbors to consider, k, if the value of k is too low, then the algorithm can be overfit with high variance. In contrast, if the values are too large, k can fail to capture richness in the data and exhibit high bias.

Distance Metric. Further, as mentioned in Section 3.4.2, the term nearby implies a distance. In this study, two common distance metrics are considered, Manhattan distance and Euclidean distance. Manhattan distance represents the distance between

two points measured along axes at right angles. Euclidean distance represents the length of a direct line between two points. Both distances can be represented by the Minkowski distance formula as noted in Figure 3.5, with Manhattan and Euclidean distances having $p = 1$ and $p = 2$, respectively.

The Minkowski distance of order $p \in \mathbb{Z}$ between two points $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ...y_n) \in \mathbb{R}$ is defined as:

$$D(X, Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{3.5}$$

Weighting Function. It is important to consider the weighting function which is utilized to perform the weighted average for the KNN algorithm. Three options are considered for this weighting function:

- Uniform weighting - performing an average of all the k nearest neighbors.
- Distance weighting - utilizing the distance metric to perform a weighted average of the k nearest neighbors adjacent to the point.
- Gaussian kernel weighting function - utilizing a kernel function, described in Equation 3.6, to define the weights for averaging the k nearest neighbors.

For the Gaussian kernel weighting function, tuning of the kernel width σ for both Manhattan and Euclidean distances can be found in Figure 3-2. Upon observation we find optimal values of RMSE occur at $\sigma = 4$ with the Manhattan distance significantly outperforming Euclidean distance for nearly all σ and k under consideration.

Figure 3-3 shows the comparison of the Gaussian kernel function with $\sigma = 4$, uniform, and distance weighting functions. Both Manhattan and Euclidean distance metrics are utilized. Upon observation, we find optimal values of RMSE occur using the Gaussian kernel at $k = 7$, utilizing Manhattan distance.

$$w_i = e^{\frac{-(D(X,Y))^2}{\sigma}} \tag{3.6}$$

where:

$D(X, Y) =$ Minkowski distance presented in Equation 3.5

$\sigma =$ Kernel width

$w_i =$ Weight of neighbor i

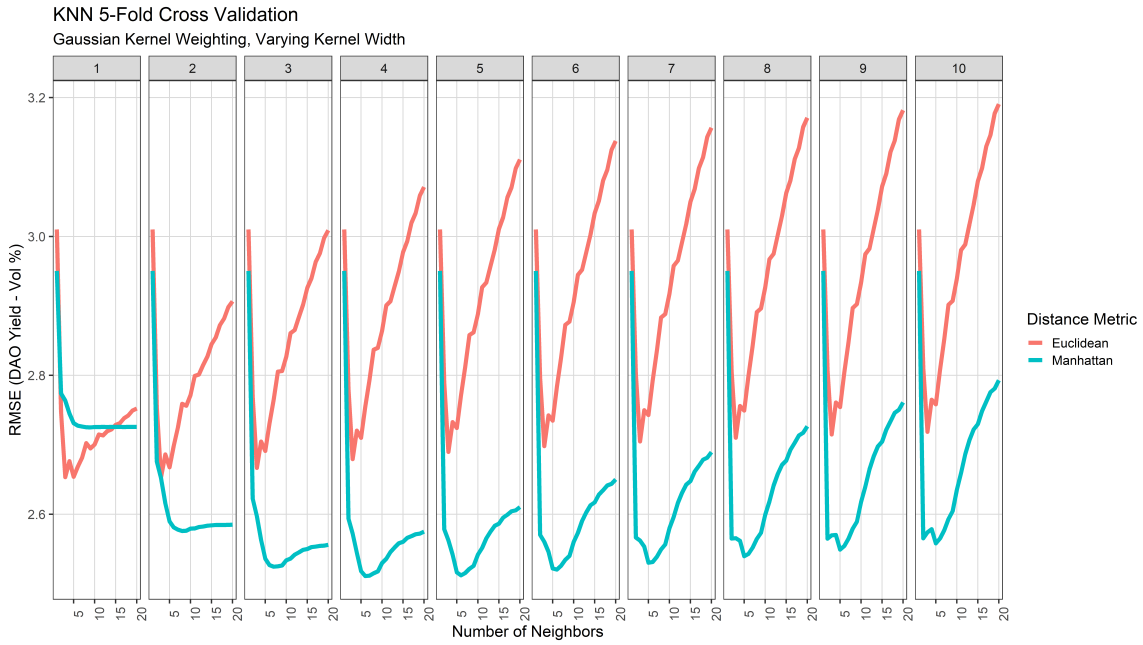

Figure 3-2: KNN: Tuning of σ for use in Gaussian kernel weighting considering both Manhattan and Euclidean distances. $k \in (0, 20]$.

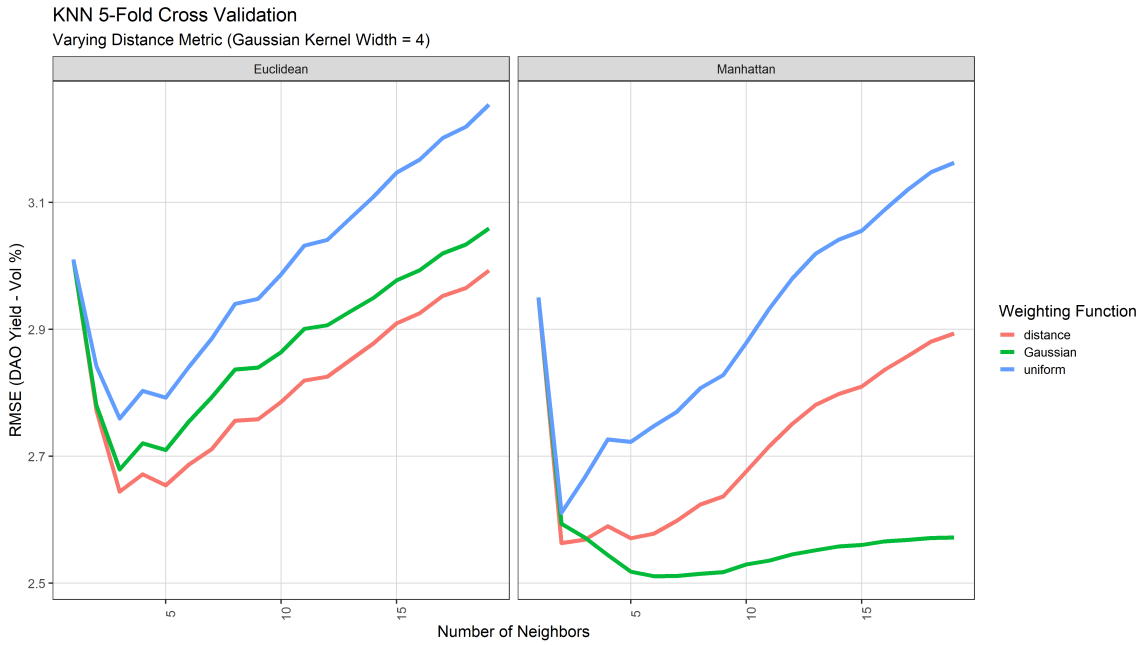

Figure 3-3: KNN: Comparison of Gaussian ($\sigma = 4$), uniform, and distance weighting functions for both Manhattan and Euclidean distance metrics.

3.4.3 Random Forest (RF)

Random Forest (RF) is an ensemble method that builds upon the concepts of bagged decision trees. RF generates multiple decision trees, and when building each decision tree, considers only a random sample of the total predictors available in the dataset, utilizing only the predictors in this random sampling when considering splits in the decision tree (James et al., 2014). This algorithm is useful for predicting non-linear interactions among predictors. For this thesis, RF utilizes bootstrap samples to construct trees.

Hyperparameters

While there are a plethora of hyperparameters available to tune in modern RF implementations, we focus our attention on two in particular:

- Number of features considered at each split
- Minimum number of observations in each terminal node

Number of features considered at each split Particularly challenging within RF is the lack of consistent variable names within open source packages. For example, in the `ranger` package for **R** (Wright & Ziegler, 2017), this hyperparameter is referred to as `mtry`. In Python's sklearn (Pedregosa et al., 2011), this is called `max_features`. Regardless of the package, in selecting smaller values for this hyperparameter, each tree in the random forest will consider less features when splitting at each node, but as we select more features at each split, we begin to approach bagged forests.

Minimum number of observations in each terminal node Another important hyperparameter to consider is the minimum number of observations in each terminal node of the trees. Again, in the `ranger` package for **R** (Upton & Cook, 1996), this hyperparameter is referred to as `min.node.size`. In Python's **sklearn** (Pedregosa et al., 2011), this is referred to as `min_samples_leaf`. Regardless of the package, when selecting lower values for this parameter, the RF will be comprised of deeper trees.

In contrast, when selecting higher values, the ensemble will be restricted to shallower trees.

Results of tuning these hyperparameters can be found in Figure 3-4. One can observe error decreasing with lower observations per terminal node, indicating a preference for deeper trees. Further, one can see that we hit a plateau of error improvement when considering more than four features at each split.

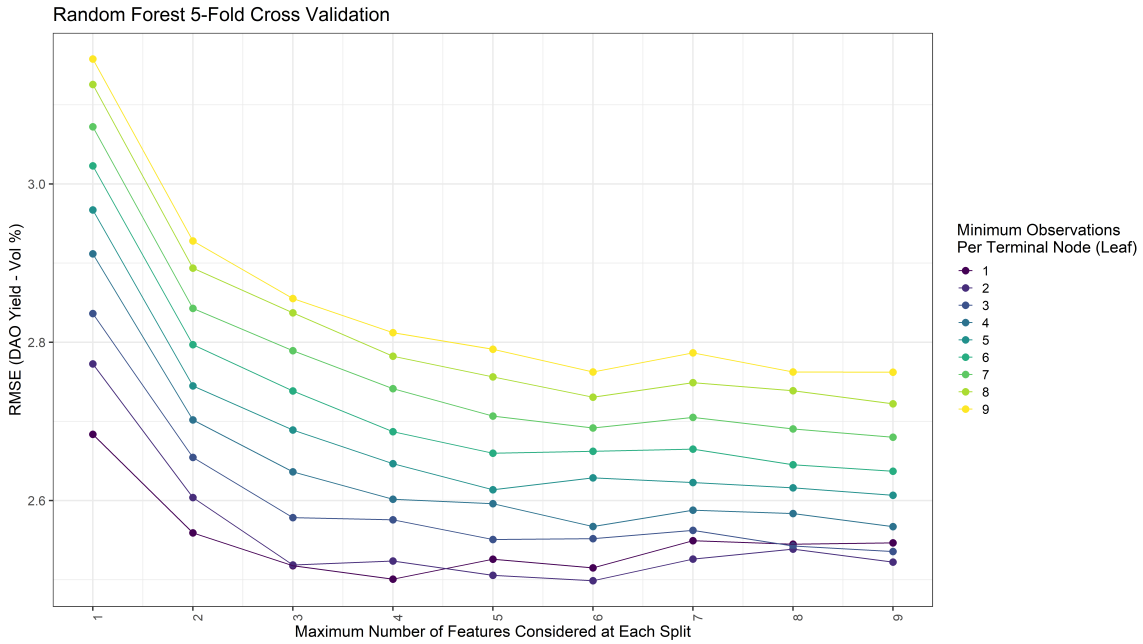

Figure 3-4: RF: Tuning of maximum number of variables considered per split and minimum number of observations in each terminal node.

3.4.4 Extremely Randomized Trees (ET)

The Extremely Randomized Trees (ET) algorithm builds an ensemble of unpruned regression trees similar in concept to RF. However, it is has two primary differences from RF. First is the difference in splitting at each decision node. While both methods consider a random subset of features for each split in the decision node, RF makes the split by determining the best split from the random subset of features under consideration. In contrast, ET splits the tree at each decision node randomly without any consideration of the "best split". Second, ET does not perform bootstrap resampling and instead uses the whole training set to grow the trees.

Hyperparameters

Similar to RF, the primary hyperparameters under consideration are the number of features considered at each split and the minimum number of observations in each terminal node of the trees. In Python's `sklearn` (Pedregosa et al., 2011), these are called `max_features` and `min_samples_leaf`, respectively.

Results of tuning these hyperparameters can be found in Figure 3-5. Similar to RF, we see that the extremely randomized trees algorithm has lower error with decreasing observations per terminal node, indicating a preference for deeper trees. Further, one can observe a plateau in error improvement when considering six features per split.

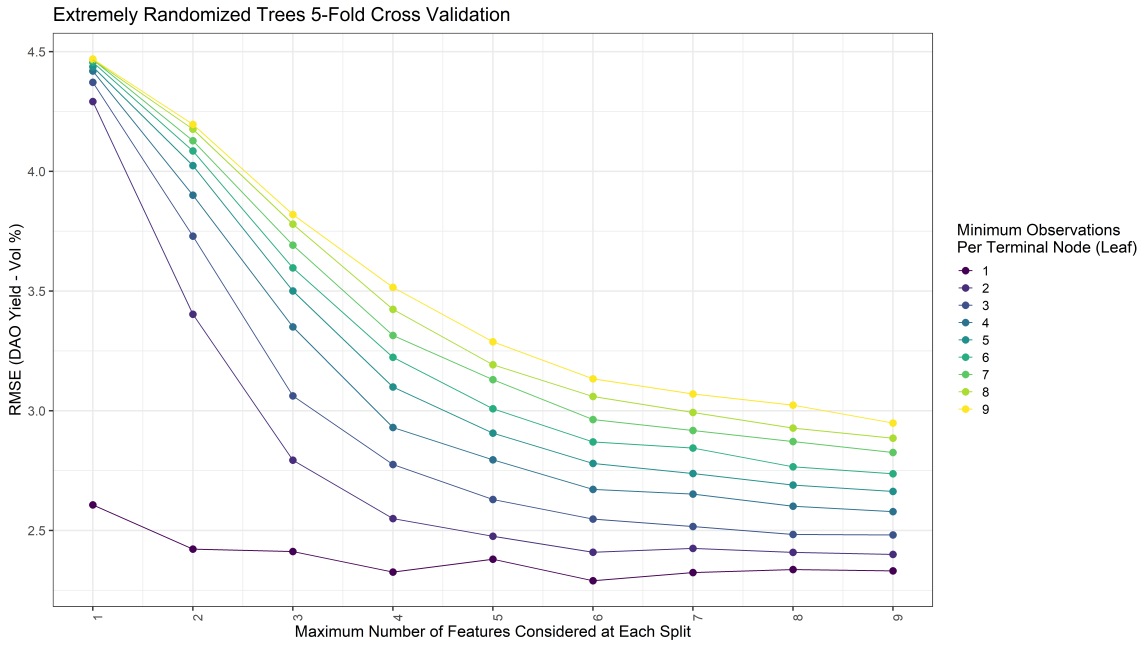

Figure 3-5: ET: Tuning of maximum number of variables considered per split and minimum number of observations in each terminal node.

3.4.5 XGBoost

XGBoost is a gradient boosting framework (Chen & Guestrin, 2016). Similar to other boosting algorithms, XGBoost combines multiple weak base learners to form an ensemble model by training each new base learner on the residuals of the ensemble from

the previous iteration. XGBoost controls for over-fitting by utilizing a regularized model. XGBoost's tree booster is utilized for the purposes of this study in order to better model non-linear relationships within the data.

Hyperparameters

XGBoost has multiple hyperparameters of interest. After exploratory analysis on the cross validation results, it was determined that there were two hyper parameters that affected the model performance the most. The first is subsample ratio of columns when construction each tree, referred to as `colsample_bytree` within XGBoost's Sci-Kit Learn API (Chen & Guestrin, 2016). Next is the maximum tree depth for each base learner, referred to as `max_depth` in XGBoost's Sci-kit Learn API.

Results of tuning these hyperparameters can be seen in Figure 3-6. We see that for this problem, XGBoost generally has lower error with increasing `colsample_bytree` and with increasing `max_depth`. However, we still find that the error is minimized with `colsample_bytree` $= 0.9$ and `max_depth` $= 6$.

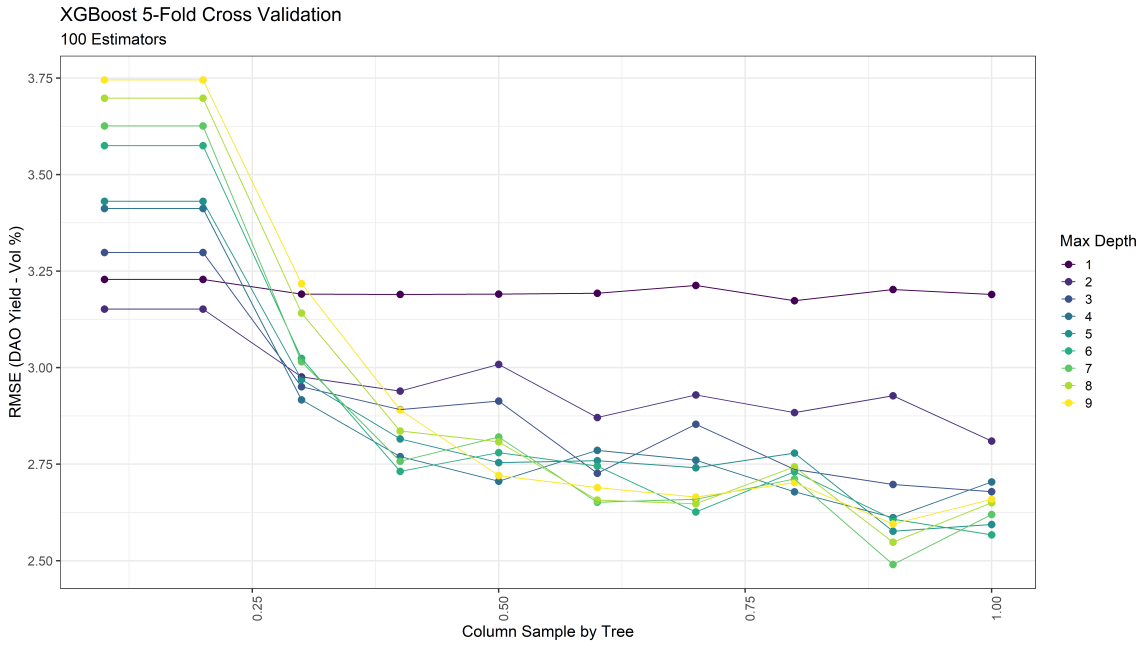

Figure 3-6: XGBoost: Tuning of subsample ratio of columns (Column Sample by Tree) and maximum tree depth for each base learner (Max Depth) with 100 estimators.

3.5 Summary of Hyperparameters

Table 3.1 summarizes all of the hyperparameters under consideration for this thesis.

Hyperparameter	Model	Description
k	KNN	Number of nearest neighbors to use in averaging.
Distance Metric	KNN	Type of distance metric to use when determining distance. Can be Manhattan or Euclidean distance.
Weighting Function	KNN	Type of weighting function to use when averaging. Can be uniform, distance, or Gaussian kernel.
σ	KNN	Kernel width. Used only when Gaussian Kernel Weighting Function is used.
`mtry`	RF, ET	Number of features considered at each split. Also referred to as `max_features`.
`min.node.size`	RF, ET	Minimum number of observations in each terminal node. Also referred to as `min_samples_leaf`.
`colsample_bytree`	XGBoost	Subset ratio of columns when constructing each tree.
`max_depth`	XGBoost	Maximum tree depth for each base learner.

Table 3.1: Hyperparameter summary for all methods.

Chapter 4

Linear Optimization

Linear optimization is a method to achieve the best outcome in a mathematical model whose requirements are represented by linear relationships. Specifically, linear optimization is a technique for the optimization of a linear objective function which is subject to linear constraints.

This chapter discusses utilization of the predictive models developed in Chapter 3 in a linear optimization framework and utilizes work completed by Gurobi (Gurobi Optimization, 2021) with modifications made to suit the problem at hand.

A linear optimization problem can be described in two parts:

1. A linear function to be maximized or minimized.

2. Problem constraints

This chapter will first discuss the data utilized for this optimization problem, then provide a description of the problem at hand. After discussing these items, we will address the linear function to be maximized and the constraints under consideration, formulating the optimization problem in greater detail.

4.1 Dataset

4.1.1 Crude Oil

Data for nine crude oils were obtained utilizing publicly available data provided by Exxon-Mobil (ExxonMobil, 2020). The prices for these crude oils were obtained using OilPrice (*Crude Oil Prices Today*, 2021). Parameters utilized in this analysis are presented in Table 4.1.

Property	Units	Description
Light Naphtha Cut Volume	% LV	Fraction of Crude Oil that will distill to Light Naphtha
Medium Naphtha Cut Volume	% LV	Fraction of Crude Oil that will distill to Medium Naphtha
Heavy Naphtha Cut Volume	% LV	Fraction of Crude Oil that will distill to Heavy Naphtha
Light Oil Cut Volume	% LV	Fraction of Crude Oil that will distill to Light Oil
Heavy Oil Cut Volume	% LV	Fraction of Crude Oil that will distill to Heavy Oil
Resid Cut Volume	% LV	Fraction of Crude Oil that will distill to Resid
Specific Gravity (60 °F)	Unitless	Otherwise Referred to as the Relative Density, the ratio of the density of the Feed with respect to water, expressed as $\frac{\rho_{Feed}}{\rho_{H_2O}}$
Resid Viscosity (212 °F)	cSt	Kinematic Viscosity of the Feed at 212 °F

Table 4.1: Crude oil properties utilized for this analysis. Data obtain from publicly available information published by Exxon-Mobil (ExxonMobil, 2020)

.

4.1.2 Finished Products

Finished product pricing is presented in Table 4.2.

Finished Product	Price ($/Bbl)
Premium Gasoline	52.00
Regular Gasoline	52.00
Jet Fuel	49.00
Fuel Oil	45.50
Asphalt	41.00

Table 4.2: Finished Petroleum Product Pricing. Motor gasoline pricing obtained from the U.S. Energy Information Administration (EIA, 2021). Jet fuel pricing were obtained from the International Air Transport Association (IATA, 2021)

4.2 Decision Variables

A table of the decision variables can be found in Table 4.3. All decision variables are non-negative variables, meaning that they must be ≥ 0.

4.3 Problem Description and Constraints

For this optimization problem, we consider a hypothetical downstream oil gas refinery which purchases K types of crude oil and refines them through a five-step process of distillation, reforming, cracking, SDA, and blending. These steps are done so that the refinery may produce finished petroleum products for sale. A graphical representation of this model is presented in Figure 4-1.

4.3.1 Distillation

As previously mentioned in Chapter 1, the distillation process separates crude oil into fractions according to their boiling points. We consider separation into six fractions: light naphtha, medium naphtha, heavy naphtha, light oil, heavy oil, and resid.

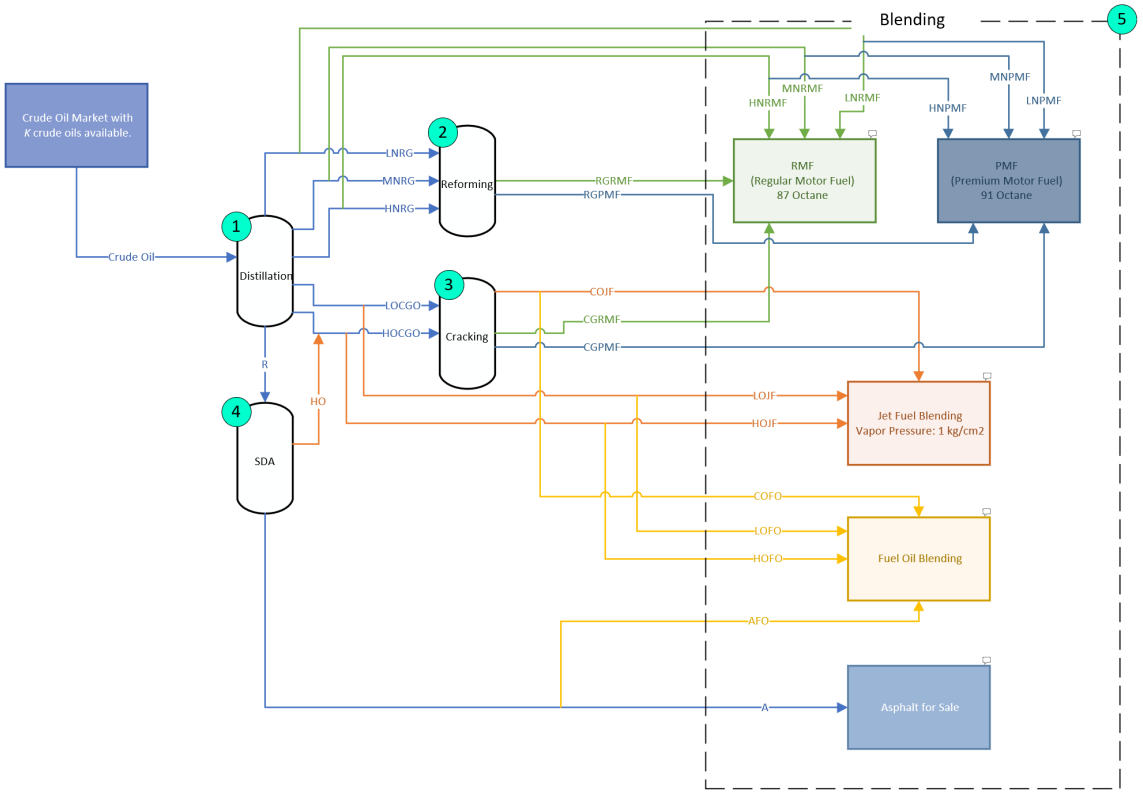

Figure 4-1: Process Flow Diagram of a hypothetical oil refinery for use in a linear optimization problem. Decision variable descriptions can be found in Table 4.3.

Decision Variable	Description
CR	Number of barrels of crude i to buy.
LN	Number of barrels of light naphtha to distill.
MN	Number of barrels of medium naphtha to distill.
HN	Number of barrels of heavy naphtha to distill.
LO	Number of barrels of light oil to distill.
HO	Number of barrels of heavy oil to distill.
R	Number of barrels of residuum to distill.
LNRG	Number of barrels of light naphtha used to produce reformed gasoline.
MNRG	Number of barrels of medium naphtha used to produce reformed gasoline.
HNRG	Number of barrels of heavy naphtha used to produce reformed gasoline.
RG	Number of barrels of reformed gasoline to produce.
LOCGO	Number of barrels of light oil used to produce cracked gasoline and cracked oil.
HOCGO	Number of barrels of heavy oil used to produce cracked gasoline and cracked oil.
CG	Number of barrels of cracked gasoline to produce.
CO	Number of barrels of cracked oil to produce.
LNPMF	Number of barrels of light naphtha used to produce premium motor fuel.
LNRMF	Number of barrels of light naphtha used to produce regular motor fuel.
MNPMF	Number of barrels of medium naphtha used to produce premium motor fuel.
MNRMF	Number of barrels of medium naphtha used to produce regular motor fuel.
HNPMF	Number of barrels of heavy naphtha used to produce premium motor fuel.
HNRMF	Number of barrels of heavy naphtha used to produce regular motor fuel.
RGPMF	Number of barrels of reformed gasoline used to produce premium motor fuel.
RGRMF	Number of barrels of reformed gasoline used to produce regular motor fuel.
CGPMF	Number of barrels of cracked gasoline used to produce premium motor fuel.
CGRMF	Number of barrels of cracked gasoline used to produce regular motor fuel.
LOJF	Number of barrels of light oil used to produce jet fuel.
HOJF	Number of barrels of heavy oil used to produce jet fuel.
COJF	Number of barrels of cracked oil used to produce jet fuel.
RAS	Number of barrels of residuum used to produce asphalt.
PMF	Number of barrels of premium motor fuel to produce.
RMF	Number of barrels of regular motor fuel to produce.
JF	Number of barrels of jet fuel to produce.
FO	Number of barrels of fuel oil to produce.

Table 4.3: Table of decision variables and abbreviations.

4.3.2 Reforming

After distillation, light, medium, and heavy naphtha can be blended together into regular or premium gasoline, or they can go through a process called reforming. The

output of this reforming process is a product referred to as reformed gasoline with an octane number of 115.

Each type of naphtha yields a different amount of reformed gasoline. The yields of reformed gasoline for each type of naphtha are given in Table 4.4.

Naphtha	Yield of Reformed Gasoline ($\frac{\text{bbl}}{\text{bbl}}$)
Light	0.6
Medium	0.52
Heavy	0.45

Table 4.4: Reformed Gasoline Yields from varying Naphtha Types.

4.3.3 Cracking

Light and heavy oils can be blended into jet fuel or put through a process known as fluidized catalytic cracking. The catalytic cracked produces cracked oil and cracked gasoline.

Cracked gasoline has an octane number of 105 with yields presented in Table 4.5.

Oil	Yield of Cracked Oil ($\frac{\text{bbl}}{\text{bbl}}$)	Yield of Cracked Gasoline ($\frac{\text{bbl}}{\text{bbl}}$)
Light Oil	0.68	0.28
Heavy Oil	0.75	0.2

Table 4.5: Cracked oil and gasoline yields for the cracking process.

4.3.4 SDA

Resid is fed to the SDA, which produces either heavy oil or asphalt. In an attempt to parametrize the flow of oil out of the SDA, we are presented with two options:

1. Utilize fixed values of DAO yield to predict the heavy oil and asphalt yields that come out of the SDA.

2. Utilize the predictive machine learning models developed in Chapter 3 to predict the heavy oil and asphalt yields that come out of the SDA.

We will contrast these two approaches in order to consider the value that the predictive machine learning models bring to an optimization framework with results presented in Chapters 5.

4.3.5 Blending

Blending is the process of bringing together all of the previous streams in order to produce finished products for sale. Four products are developed in this refinery:

1. Gasoline
2. Jet Fuel
3. Fuel Oil
4. Asphalt

Gasoline There are two kinds of gasoline, regular and premium. These are made by blending naphtha, reformed gasoline, and cracked gasoline. The primary requirement under consideration is the octane content of the gasoline. Regular gasoline must have an octane of at least 84 and premium gasoline must have an octane number of at least 94. We assume that octane numbers blend linearly by volume. We consider the octane numbers for the light, medium, and heavy naphthas are 90, 80, and 70 respectively. The octane numbers of reformed gasoline and cracked gasoline were previously noted in Section 4.3.2 and 4.3.3, respectively.

Jet Fuel Jet fuel is made by blending light, heavy, and cracked oils which have vapor pressures of 1.0, 0.6, and 1.5 $\frac{\text{kg}}{\text{cm}^2}$, respectively. Jet Fuel must have a vapor pressure that does not exceed 1.0 $\frac{\text{kg}}{\text{cm}^2}$. We assume that vapor pressures blend linearly by volume.

Fuel Oil Fuel oil is produced by blending light oil, cracked oil, heavy oil, and asphalt in a ratio of $10 : 4 : 3 : 1$.

Asphalt Asphalt is produced as a low-value product from the SDA and does not have product quality constraints.

4.3.6 Throughput Constraints

We consider the refinery to have the following throughput constraints:

1. At most, 45,000 barrels of crude can be distilled per day.
2. At most, 10,000 barrels of naphtha can be reformed per day.
3. At most, 8,000 barrels of oil can be cracked per day.
4. To meet minimum flow requirements on the bottoms stream of the SDA, we must produce a minimum of 500 barrels of Asphalt.
5. Premium gasoline production must be at least 40 % of regular gasoline production.
6. There is unlimited availability of all K crude oils under consideration.

4.4 Network Flows

It is convenient to consider the refinery linear optimization problem analogous to a network flow optimization problem. To that end, it is useful to represent the refinery as a directed graph with each step in the process as a node, and the amount of flow cannot exceed the capacity of each arc. However, in order to complete such a graph, we must add further nodes into the diagram representing **bypass nodes**.

Bypass nodes represent decision nodes in which we must make a decision about where to send the incoming streams. As an example, referencing Figure 4-2, we can examine bypass node 1. At this node, we must make a decision if the light naphtha should go to the regular motor fuel blending pool, the premium motor fuel blending pool, or be sent to be reformed and processed into reformed gasoline. Nine of these bypass nodes are added so as to construct a proper network diagram.

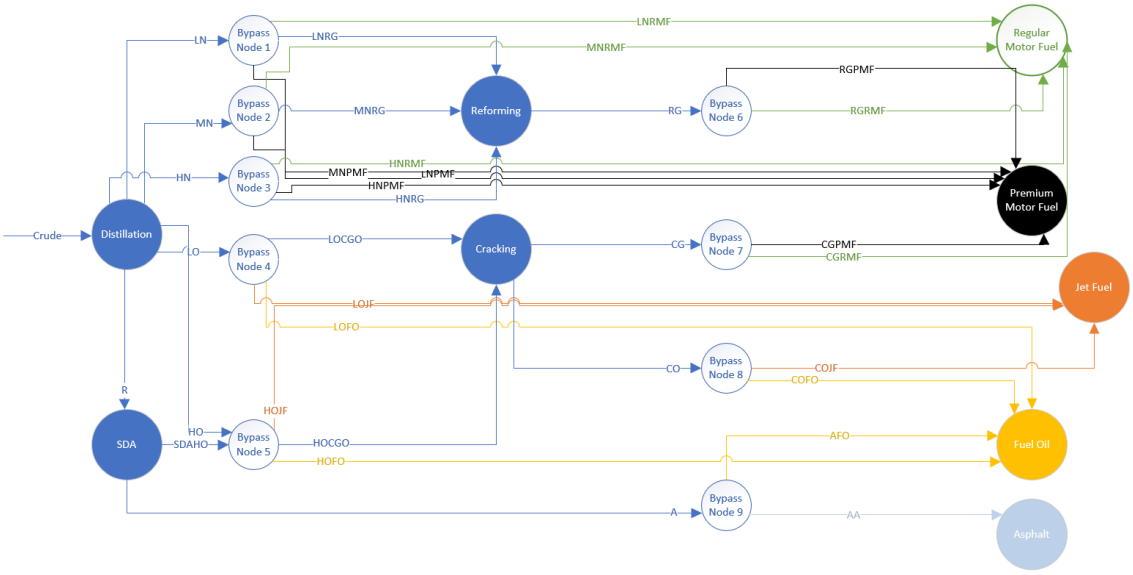

Figure 4-2: Network flow diagram of a hypothetical oil refinery for use in a linear optimization problem. Decision variable descriptions can be found in Table 4.3.

4.5 Objective

The objective of this linear optimization problem is to maximize the profit of the refinery. We can represent this profit as noted in Equation 4.1.

$$\text{Profit} = \sum_{j=1}^{N} (\text{Price}_j * \text{Flow}_j) - \sum_{i=1}^{K} (\text{Cost}_i * \text{Purchase}_i) \tag{4.1}$$

where:

$$i = 1, ..., K \text{ (Crude Oils)}$$

$$j = 1, ..., N \text{ (Products)}$$

4.6 Constraints

Based upon our problem description, we can classify our constraints into four categories:

- Capacity Constraints
- Yield Constraints
- Flow Constraints
- Product Quality Constraints

Please note that all acronyms and descriptions for variables are described in Table 4.3.

4.6.1 Capacity Constraints

Capacity constraints are the limits placed on the hypothetical refinery as a result of limited distillation, reforming, and cracking capacity. The distillation capacity constraint can be represented by Equation 4.2.

$$\sum_{i \in \text{ Crudes}} \text{CR}_i \leq 45{,}000 \tag{4.2}$$

Similarly, the reforming capacity constraint can be represented by Equation 4.3.

$$\text{LNRG} + \text{MNRG} + \text{HNRG} \leq 10{,}000 \tag{4.3}$$

Again, the cracking capacity constraint can be represented by Equation 4.4.

$$\text{LOCGO} + \text{HOCGO} \leq 8{,}000 \tag{4.4}$$

Lastly, the minimum flow requirements for the SDA can be represented by Equation 4.5.

$$\mathrm{A} \geq 500 \tag{4.5}$$

4.6.2 Yield Constraints

With the exception of Blending, each of the major processes (Distillation, Reforming, Cracking, and SDA) under consideration have associated yields that must be explicitly declared.

Distillation Yield Constraints

Light Naphtha Light naphtha is produced by distillation of the crude oils. The amount of light naphtha created by distillation can be represented by Equation 4.6.

$$\sum_{i=1}^{K}(\mathrm{LN}_{\mathrm{CR}_i} * \mathrm{CR}_i) = \mathrm{LN} \tag{4.6}$$

Medium Naphtha Similarly, medium naphtha is produced by distillation of the crude oils. The amount of medium naphtha created by distillation can be represented by Equation 4.7.

$$\sum_{i=1}^{K}(\mathrm{MN}_{CR_i} * \mathrm{CR}_i) = \mathrm{MN} \tag{4.7}$$

Heavy Naphtha Heavy naphtha is produced by distillation of the crude oils. The amount of heavy naphtha created by distillation can be represented by Equation 4.8.

$$\sum_{i=1}^{K}(\mathrm{HN}_{CR_i} * \mathrm{CR}_i) = \mathrm{HN} \tag{4.8}$$

Light Oil Light Oil is produced by distillation of the crude oils. The amount of light oil created by distillation can be represented by Equation 4.9.

$$\sum_{i=1}^{K}(\mathrm{LO}_{CR_i} * \mathrm{CR}_i) = \mathrm{LO} \tag{4.9}$$

Heavy Oil Heavy Oil is produced by distillation of the crude oils. The amount of light oil created by distillation can be represented by Equation 4.10.

$$\sum_{i=1}^{K}(\mathrm{HO}_{CR_i} * \mathrm{CR}_i) = \mathrm{HO} \tag{4.10}$$

Resid Lastly, resid is produced by distillation of the crude oils. The amount of resid created by distillation can be represented by Equation 4.11.

$$\sum_{i=1}^{K}(\mathrm{R}_{CR_i} * \mathrm{CR}_i) = \mathrm{R} \tag{4.11}$$

Reforming Yield Constraints

Reformed Gasoline Reformed gasoline is produced through reformation. The amount of reformed gasoline created by reforming can be represented by Equation 4.12. Note that the coefficients within the equation come from Table 4.4.

$$0.60 * \mathrm{LNRG} + 0.52 * \mathrm{MNRG} + 0.45 * \mathrm{HNRG} = \mathrm{RG} \tag{4.12}$$

Cracking Yield Constraints

Cracked Oil Cracked oil is produced through the cracking process. The amount of cracked oil created by cracking can be represented by Equation 4.13. Note that the coefficients within the equation come from Table 4.5.

$$0.68 * \text{LOCGO} + 0.75 * \text{HOCGO} = \text{CO} \tag{4.13}$$

Cracked Gasoline Cracked gasoline is also produced through the cracking process. The amount of cracked gasoline created by cracking can be represented by Equation 4.14. Note that the coefficients within the equation come from Table 4.5.

$$0.28 * \text{LOCGO} + 0.20 * \text{HOCGO} = \text{CG} \tag{4.14}$$

SDA Yield Constraints

As noted in Section 4.3.4, the SDA produces both heavy oil and asphalt.

Heavy Oil from SDA We can represent the amount of heavy oil produced from the SDA by Equation 4.15.

$$\text{SDAHO} = \sum_{i=1}^{K} (\text{R}_{\text{CR}_i} * \text{CR}_i * \eta_i) \tag{4.15}$$

where:

η_i = DAO Yield for crude i

Asphalt from SDA We can represent the amount of asphalt produced from the SDA by Equation 4.16.

$$\text{A} = \sum_{i=1}^{K} (\text{R}_{\text{CR}_i} * \text{CR}_i * (1 - \eta_i)) \tag{4.16}$$

4.6.3 Flow Constraints

Flow constraints represent mass conservation, meaning that the amount of barrels flowing into the node must equal the amount of barrels flowing out of the node. These apply to each of the bypass nodes as well as the product nodes.

Bypass Nodes

For the bypass nodes, it is helpful to utilize the numbering provided in Figure 4-2.

Bypass Node 1

$$\text{LN} = \text{LNPMF} + \text{LNRG} + \text{LNRMF} \tag{4.17}$$

Bypass Node 2

$$\text{MN} = \text{MNPMF} + \text{MNRG} + \text{MNRMF} \tag{4.18}$$

Bypass Node 3

$$\text{HN} = \text{HNPMF} + \text{HNRG} + \text{HNRMF} \tag{4.19}$$

Bypass Node 4

$$\text{LO} = \text{LOCGO} + \text{LOJF} + \text{LOFO} \tag{4.20}$$

Bypass Node 5

$$\text{SDAHO} + \text{HO} = \text{HOCGO} + \text{HOJF} + \text{HOFO} \tag{4.21}$$

Bypass Node 6

$$\text{RG} = \text{RGPMF} + \text{RGRMF} \tag{4.22}$$

Bypass Node 7

$$\mathrm{CG} = \mathrm{CGPMF} + \mathrm{CGRMF} \tag{4.23}$$

Bypass Node 8

$$\mathrm{CO} = \mathrm{COJF} + \mathrm{COFO} \tag{4.24}$$

Bypass Node 9

$$\mathrm{A} = \mathrm{AFO} + \mathrm{AA} \tag{4.25}$$

Finished Products

For each of the finished product nodes, the amount of finished product is equal to the amount of incoming streams. This is expressed for each of the finished products.

Regular Motor Fuel (Gasoline)

$$\mathrm{RMF} = \mathrm{RGRMF} + \mathrm{LNRMF} + \mathrm{MNRMF} + \mathrm{HNRMF} + \mathrm{CGRMF} \tag{4.26}$$

Premium Motor Fuel (Gasoline)

$$\mathrm{PMF} = \mathrm{RGPMF} + \mathrm{LNPMF} + \mathrm{MNPMF} + \mathrm{HNPMF} + \mathrm{CFPMF} \tag{4.27}$$

Jet Fuel

$$\mathrm{JF} = \mathrm{LOJF} + \mathrm{HOJF} + \mathrm{COJF} \tag{4.28}$$

Fuel Oil

$$\mathrm{FO} = \mathrm{LOFO} + \mathrm{HOFO} + \mathrm{COFO} + \mathrm{AFO} \tag{4.29}$$

4.6.4 Product Quality Constraints

Lastly, we consider the product quality constraints previously discussed.

Motor Fuel (Gasoline) Octane Tolerance

As discussed in Section 4.3.5, there are octane requirements for both regular and premium motor gasoline. These can be represented by Equations 4.30 and 4.31 where each coefficient represents the octane number of the corresponding stream.

Regular Motor Fuel (Gasoline)

$$\begin{aligned} 90 * \text{LNRMF} + 80 * \text{MNRMF} + 70 * \text{HNRMF} \\ +115 * \text{RGRMF} + 105 * \text{CGRMF} \geq 87 * \text{RMF} \end{aligned} \tag{4.30}$$

Premium Motor Fuel (Gasoline)

$$\begin{aligned} 90 * \text{LNPMF} + 80 * \text{MNPMF} + 70 * \text{HNPMF} \\ +115 * \text{RGPMF} + 105 * \text{CGPMF} \geq 91 * \text{PMF} \end{aligned} \tag{4.31}$$

Premium-to-Regular Motor Fuel (Gasoline) Ratio

Further, as discussed in Section 4.3.5, we have a requirement to produce a ratio of premium-to-regular motor gasoline in order to meet contractual obligations. This can be represented by Equation 4.32.

$$\text{PMF} \geq 0.40 * \text{RMF} \tag{4.32}$$

Jet Fuel Vapor Pressure Tolerance

As discussed in Section 4.3.5, there is a vapor pressure target for jet fuel. This can be represented by Equation 4.33.

$$1.0 * \text{LOJF} + 0.6 * \text{HOJF} + 1.5 * \text{COJF} \leq 1.0 * \text{JF} \tag{4.33}$$

Fuel Oil Ratio

As discussed in Section 4.3.5, the fuel oil is created by blending together components in a specific ratio. This can be represented by Equation 4.34.

$$\frac{10}{18}\text{LOFO} + \frac{4}{18}\text{COFO} + \frac{3}{18}\text{HOFO} + \frac{1}{18}\text{AFO} = \text{FO} \tag{4.34}$$

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Results

This chapter provides the modeling results the scaling methods discussed in Chapter 2, the predictive modeling discussed in Chapter 3, and the linear optimization discussed in Chapter 4.

5.1 Scaling Methodology

The success of the scaling methods discussed in 2.1.3 was evaluated utilizing a KNN regressor, described in Section 3.4.2. The results are presented in Table 5.1.

Scaling Transformation	In-Sample RMSE	Out-of-Sample RMSE
StandardScaler	1.31	2.21
MinMaxScaler	1.78	2.11
RobustScaler	1.25	2.48
Yeo-Johnson	1.25	2.16
Quantile Transformer	1.26	1.83

Table 5.1: Results of varying scaling transformation. All values are presented in DAO Yield (Volume %). KNN algorithm used to evaluate in and out of sample performance using a Gaussian kernel with $\sigma = 4$, $k = 7$, and Manhattan distance. Reference Section 2.1.3 for descriptions of various scaling transformations.

These results suggest that in-sample, the Yeo-Johnson and Robust Scaler are the most effective scaling transformations to minimize RMSE. However, in order to minimize the RMSE out-of-sample, the quantile transformation with normal distribu-

tion is preferred, closely followed by the simple MinMaxScaler. Results for all KNN algorithms presented this point forward utilize a quantile transformer with normal distribution as it performed very well both in and out-of-sample.

5.2 Predictive Modeling

This section discusses the results of the predictive modeling described in Chapter 3.

5.2.1 Model Classes

Linear Regression

The results of the linear regression model on the train and test set are presented in Figure 5-1. We can see that the model doesn't overfit the data but fails to capture significant richness in the data that other algorithms are able to capture.

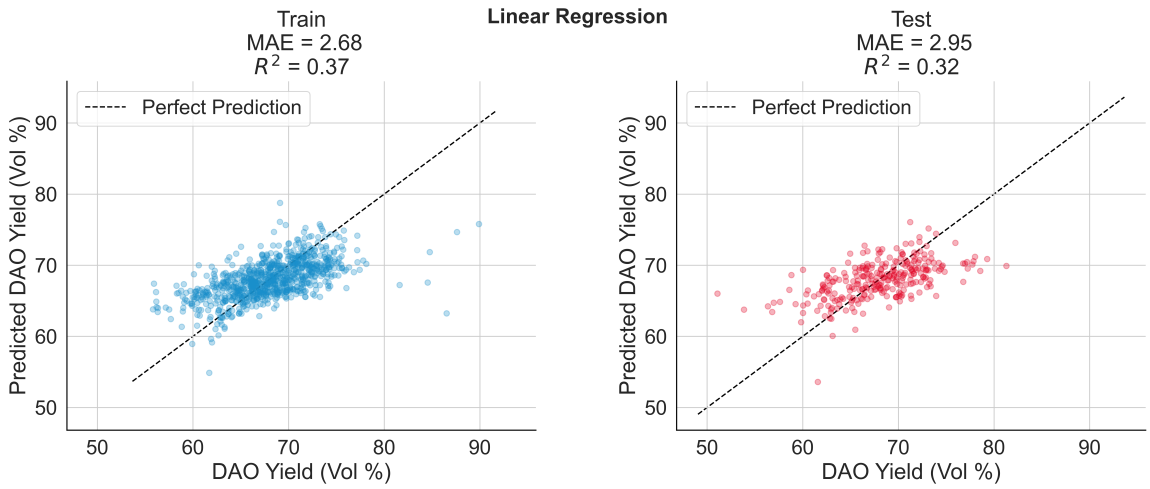

Figure 5-1: Linear regression results for the training set and testing set.

KNN

As discussed in Chapter 3, three different weighting algorithms are considered as part of KNN. Namely, these are uniform weighting, distance weighting, and Gaussian weighting. Results are presented for all three of these weighting methods.

KNN, Uniform Weighting The results of the KNN, uniform weighting model on the train and test set are presented in Figure 5-2. We can see that the model doesn't overfit the data and actually performs quite well on the testing data.

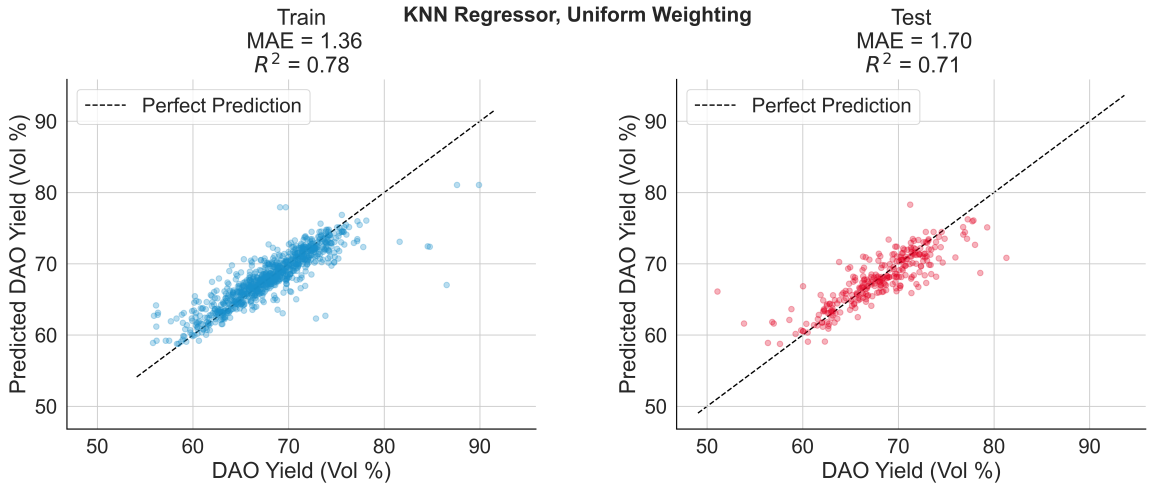

Figure 5-2: KNN, Uniform Weighting results for the training set and testing set.

KNN, Distance Weighting The results of the KNN, distance weighting model on the train and test set are presented in Figure 5-3. In this case, we can observe significant overfitting. Training data is modeled perfectly while the testing data fails to come close to the in-sample performance.

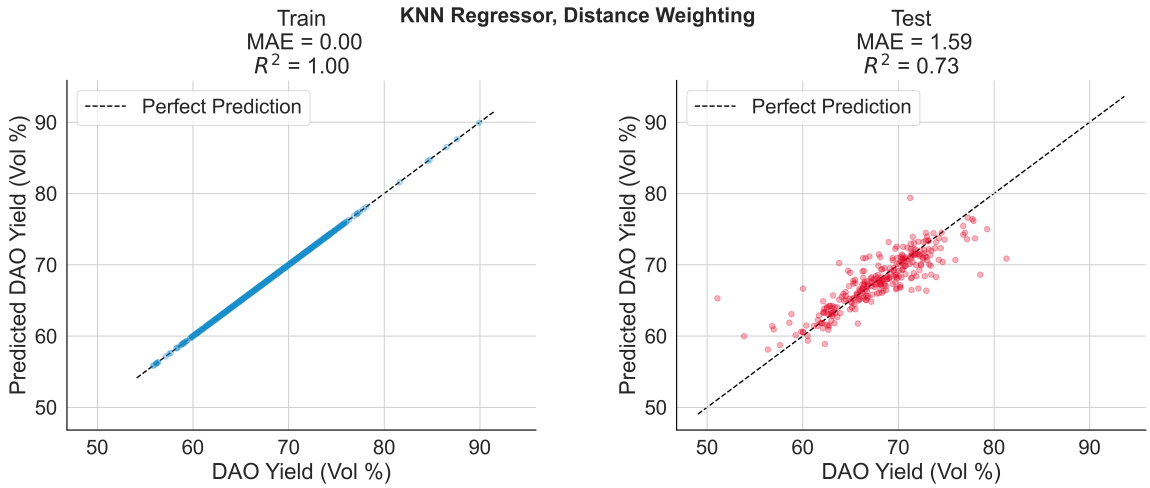

Figure 5-3: KNN, Distance Weighting results for the training set and testing set.

KNN, Gaussian Weighting The results of the KNN, Gaussian weighting model on the train and test set are presented in Figure 5-4. In this case, we can observe slight overfitting. Training data is modeled quite well while the performance on the testing data falls short. Nevertheless, the KNN, Gaussian weighting model performs best on the testing set of all the KNN models.

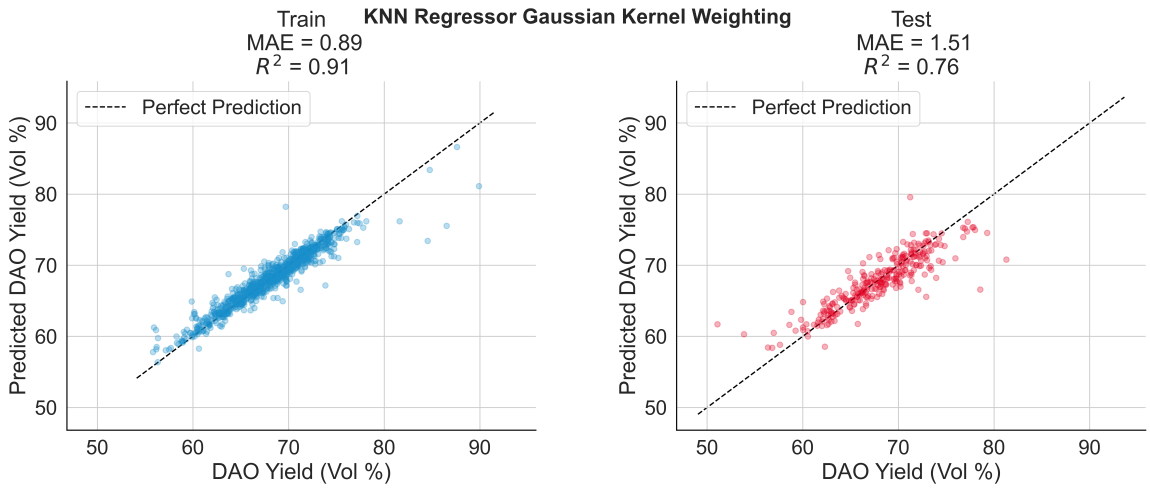

Figure 5-4: KNN, Gaussian Weighting results for the training set and testing set.

RF

The results of the RF model on the train and test set are presented in Figure 5-5. Again, we can observe slight overfitting. Training data is modeled quite well while the performance on the testing data falls short.

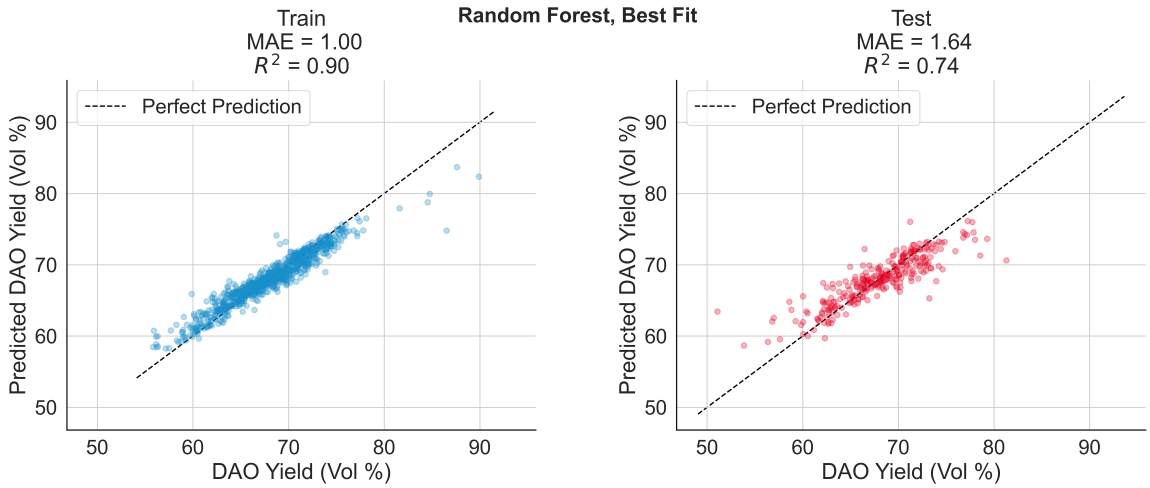

Figure 5-5: RF results for the training set and testing set.

ET

The results of the ET model on the train and test set are presented in Figure 5-6. We can observe significant overfitting. Training data is modeled perfectly while the testing data fails to come close to the in-sample performance.

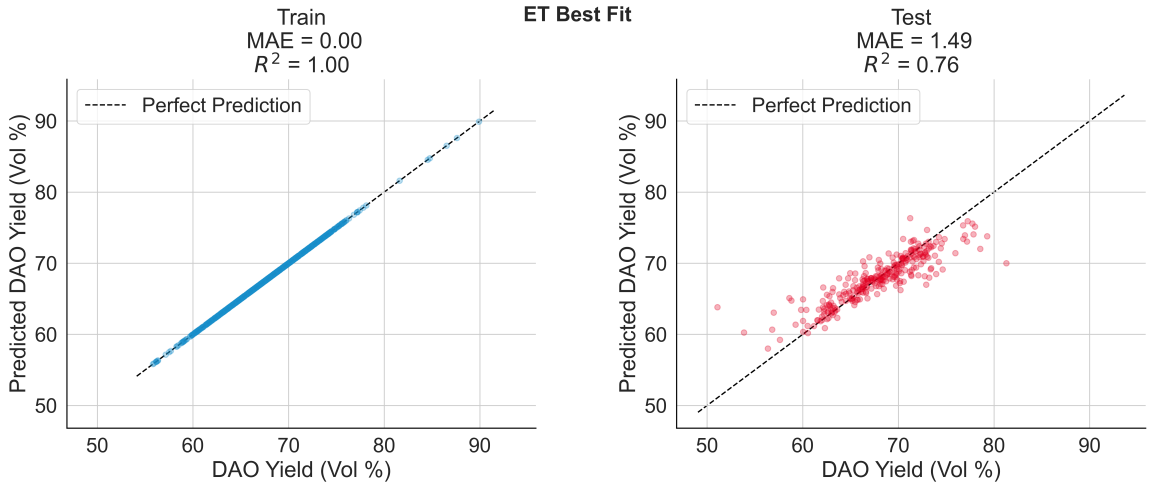

Figure 5-6: ET results for the training set and testing set.

XGBoost

The results of the XGBoost model on the train and test set are presented in Figure 5-7. We can observe significant overfitting. Training data is modeled perfectly while

the testing data fails to come close to the in-sample performance.

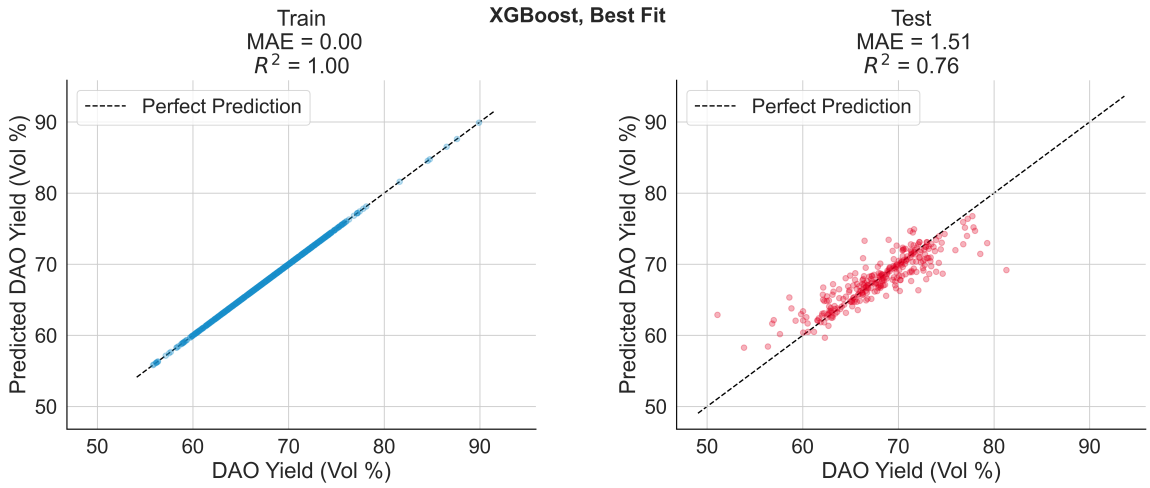

Figure 5-7: XGBoost results for the training set and testing set.

Summary of In-Sample and Out-of-Sample Performance

Table 5.2 summarizes the in and out-of-sample performance of each of the models.

Model	In-Sample R^2	Out-of-Sample R^2
Linear Regression	0.37	0.32
KNN, Uniform Weighting	0.78	0.71
KNN, Distance Weighting	1.00	0.73
KNN, Gaussian Weighting	0.91	0.76
RF	0.90	0.74
ET	1.00	0.76
XGBoost	1.00	0.76

Table 5.2: Predictive model performance for both in and out-of-sample.

5.2.2 Validation and Test Set Performance

It can be useful to judge the accuracy of the results for each algorithm considering only the training and testing sets as previously discussed. However, it is also beneficial to analyze how the algorithm performs on the validation set as well. Figure 5-8 presents the results of each tuned algorithm on a 10-fold validation set, repeated three times alongside the results from the testing set. All models outperform the base linear

regression in both testing and validation sets. While KNN with Gaussian weighting demonstrates the best out-of-sample performance, this model shows a much wider range of results in comparison to RF and ET algorithms, suggesting it may be less robust than other algorithms for this problem.

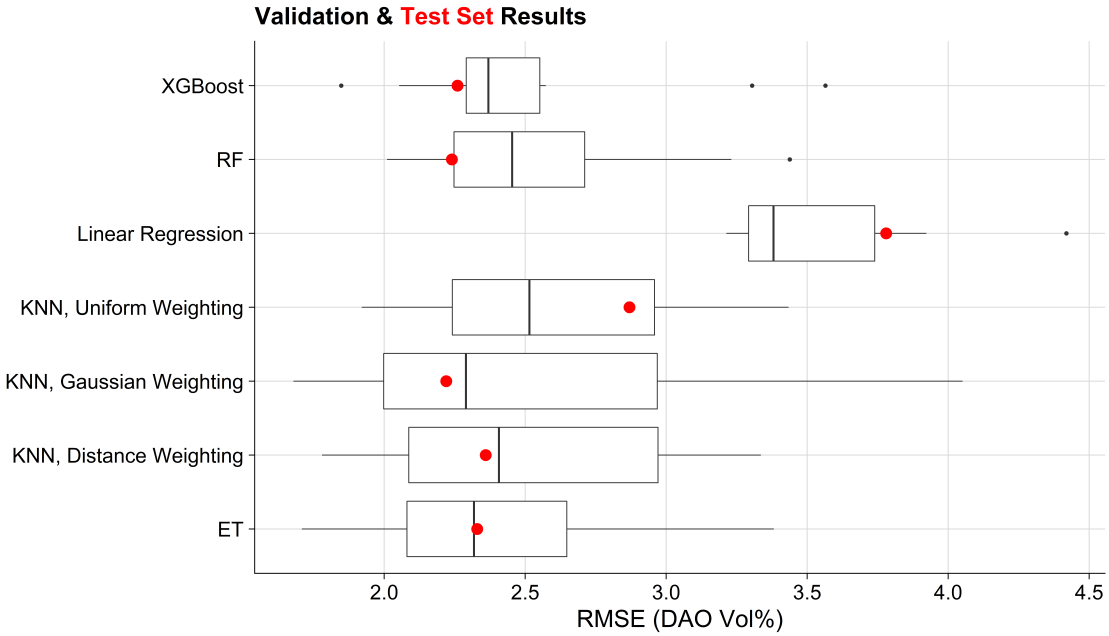

Figure 5-8: Testing set (red) and validation set (box and whisker) performance for all model algorithms.

5.2.3 Feature Importance

We can utilize Shapley Additive Explanations (SHAP) (Lundberg & Lee, 2017) to better explain and understand the output of machine learning models. This is especially useful for tree ensemble methods like RF, ET, and XGBoost which are considered in this thesis. SHAP provides insight into these models which can otherwise be opaque and challenging to understand.

Figure 5-9 presents a beeswarm plot of SHAP values for the RF algorithm sorted by importance in descending order. We find that the three most important variables relate to the feed viscosity, the 90% recovery point of the feed, and the temperature of the SDA extractor. This has implications on the physics-based models that were

previously discussed in 1.7.

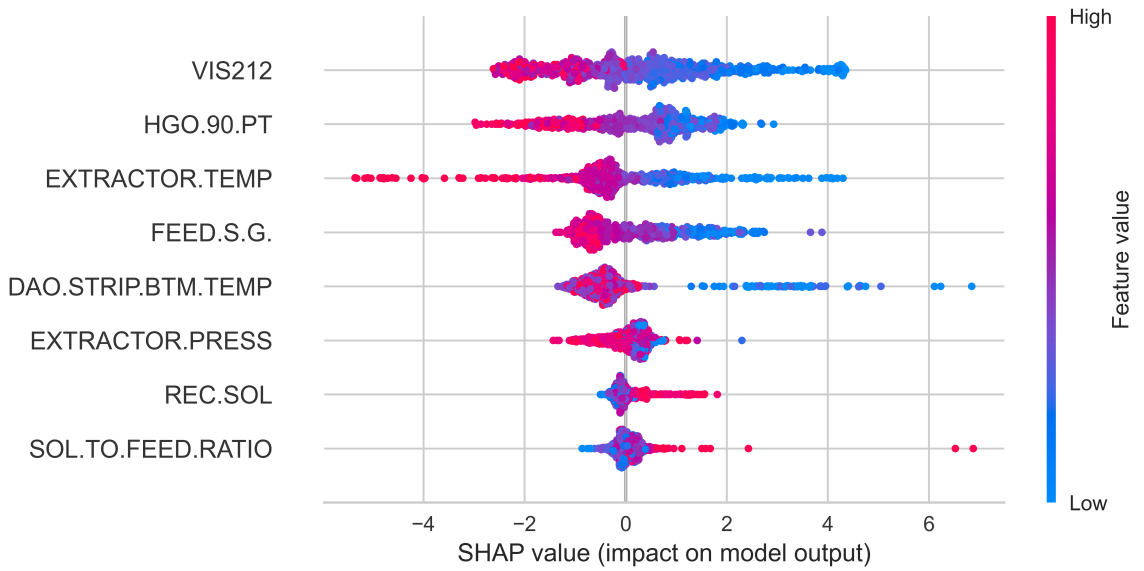

Figure 5-9: SHAP values for the RF algorithm.

5.3 Linear Optimization

This section describes the results of the linear optimization model, primarily describing the economic benefits as well as the robustness of the model under varying conditions.

5.3.1 Economic Evaluation

To properly quantify the economic impact of integrating the predictive model within the linear optimization problem, it's important to consider three cases:

1. **Base Case** - What would be the economic impact to the refinery if it did not utilize a predictive machine learning model to predict DAO yields?

2. **Machine Learning Case** - What would be the economic impact of the refinery if it utilizes a predictive machine learning model to predict DAO yields?

3. **Omniscient Case** - What would be the economic impact of the refinery in the impossible scenario whereby the refinery already knew exactly what the DAO yield, would be?

While it can be tempting to quantify the benefits by only comparing items one and two above, it is highly beneficial to understand the gap between the baseline (item one) and omniscience to understand how much closer we get to an omniscient solution by implementing more complicated methods. It is important to note that evaluating the omniscient case is only possible in a look back scenario when DAO yields have already been observed.

The method for economic evaluation is presented in Figure 5-10. For each case, we can utilize the linear optimization framework to produce the crude decision variables for a given DAO yield. Average DAO yield is denoted by $\bar{\eta}$. The DAO yield predicted by the machine learning model is denoted by $\hat{\eta}$. Lastly, the true DAO yield is denoted by η. Utilizing these yields in the linear optimization framework, we can obtain a variety of decision variables including the crude decision variables. Now, we can utilize these decision variables along with the true DAO yield, η, to determine the profit for that case.

In selecting a random data point from the training set, we find that there is noticeable profit improvement as denoted in Table 5.3. Utilizing the machine learning predictive model, we can reduce the profitability gap between the average DAO yield and the omniscient DAO yield by more than 36%.

Case	Profit ($/Day)	Gap between Omniscient Case ($/Day)
Average DAO Yield	887,000	5,500
Predictive ML DAO Yield	889,000	3,500
Omniscient DAO Yield	892,500	-

Table 5.3: Profitability Results from Linear Optimization Framework.

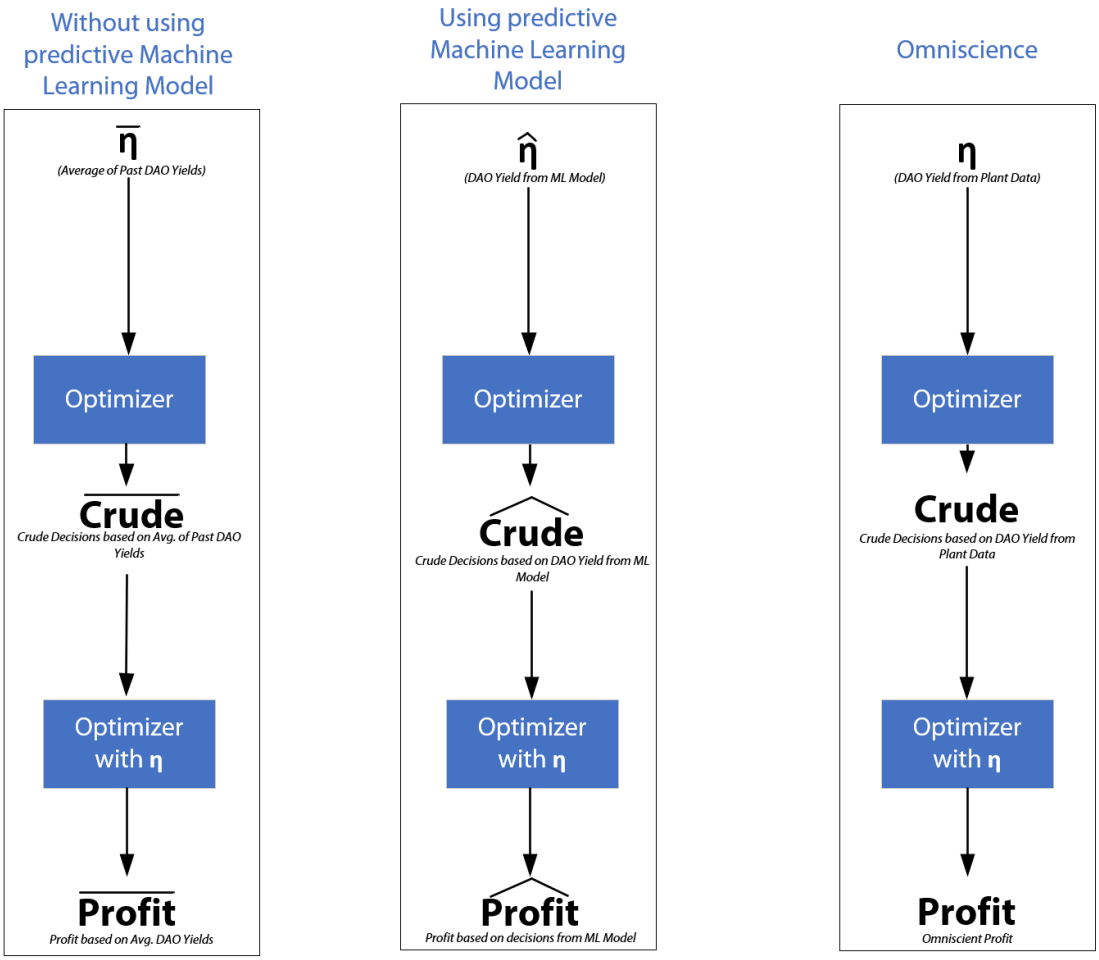

Figure 5-10: Economic evaluation framework for linear optimization problem.

5.3.2 Decision Variable Comparison

It's interesting to compare the values of each decision variable of each case to understand how each case arrived at it's final profit. This is particularly interesting since each case arrived at similar profit values in vastly different avenues.

Crude Purchasing Comparison

Figure 5-11 compares the crude purchasing decisions for each of the three cases. Of all nine crudes under consideration, all three cases only chose crude 2 and crude 8. Interestingly, the case utilizing predictive machine learning for the SDA DAO yields chose to only crude 8 which is in contrast to the omniscient case which utilized only crude 2.

Table 5.4 shows the amount of money spent in each case on crude oil. Interestingly, the case utilizing predictive machine learning for the SDA DAO yields actually spends the most money on crude oil.

Case	Expenses for Crude Oil ($/Day)
Average DAO Yield	2,860,846
Predictive ML DAO Yield	2,903,440
Omniscient DAO Yield	2,743,650

Table 5.4: Crude expenses for each case.

Finished Products Comparison

Figure 5-12 compares the finished product decision variables for each of the three cases. Notably the omniscient case produced much more asphalt than either of the other cases, and opted to produce more jet fuel.

Table 5.5 shows the revenue in each case for all finished products. Interestingly, we can observe that the omniscient cases maximized profits not by maximizing revenue, but by minimizing crude expenditures.

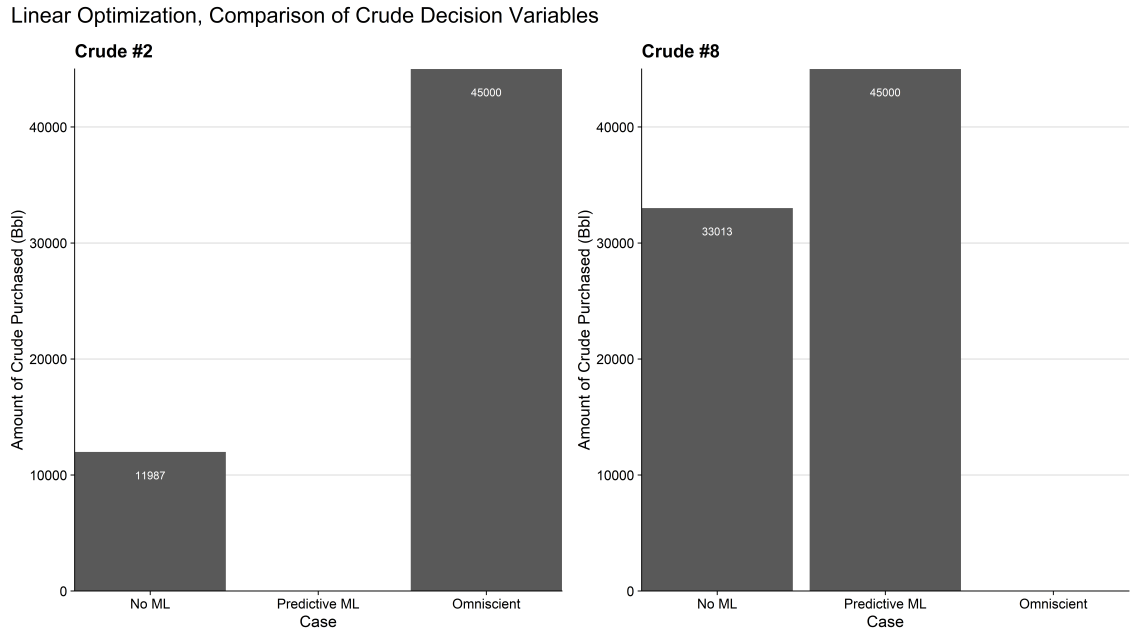

Figure 5-11: Comparison of Crude Decision Variables between the three cases. Note that, as previously stated in Chapter 4, the refinery has a crude distillation capacity constraint of 45,000 bbl.

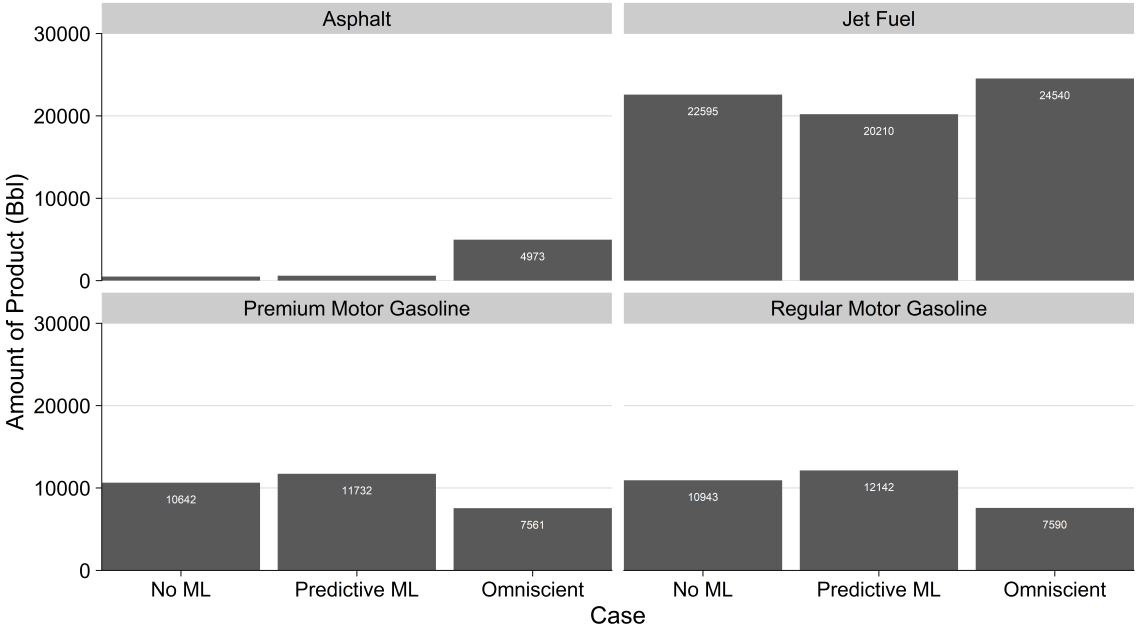

Figure 5-12: Comparison of finished product production between the three cases. Note that no case opted to generate fuel oil.

Case	Finished Product Revenue ($/Day)
Average DAO Yield	3,735,234
Predictive ML DAO Yield	3,793,405
Omniscient DAO Yield	3,608,640

Table 5.5: Finished Product Revenue for each case.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusion

This thesis has provided a continuous DAO yield estimations for a SDA unit by use of modern machine learning models using data sets from a commercial refinery in the United States. Additionally, this predictive machine learning model has been incorporated into a linear optimization framework for a hypothetical downstream refinery, demonstrating improved profitability by $730,000 annually.

6.1 Research Questions

How can machine learning based models augment their physics based counterparts, to improve the accuracy of SDA DAO yields? In the past chapter we have reviewed the results of multiple machine learning models and their relative accuracy in comparison to one another. But how do these compare to existing physics based models? While many models are proprietary, owned by private firms, others are published in literature. Utilizing Maples model (Maples, 2008) we can reconstruct a physics based approach and compare these results to the machine learning approaches considered as part of this study. When doing this, the physics based approach returns an $R^2 = 0.15$, significantly lower than those presented in Table 5.2.

This begs the question - why are machine learning methods successful in developing models to predict SDA DAO yields? I believe it's for three key reasons.

First, machine learning methods provide scalable and flexible frameworks to pro-

cess, interpret, and predict on large volumes of data. As discussed in Chapter 1, many SDA research papers focus on small amounts of data, the majority of which are collected in laboratory environments. These smaller datasets cannot provide the same amount of richness of information as large commercial datasets. When utilizing larger commercial datasets, the same methodology that has previously been used in laboratory environments may not be successful, and more advanced methods are required to solve these problems.

Second, humans have typically been in charge of determining the relationships on these smaller datasets. In contrast, this thesis utilized machine learning algorithms which, with the help of a human, helped determine the relationships of the independent and dependent variables. Advanced ensemble algorithms such as RF, ET, or XGBoost provide a method for determining relationships that may not otherwise be feasible for a human being to determine when reviewing through the data.

Lastly, machine learning allows for consideration of many independent variables at the same time. Human limitations often fail to visualize relationships beyond two or three dimensions. In contrast, machine learning methods are successful in higher order dimensions well beyond what can be visualized on a cartesian coordinate system.

It is for these three reasons that I believe that machine learning methodology can dramatically improve existing engineering and physics-based methods which seek to predict dependent variables. However, I do not advocate that machine learning models replace physics-based models. Rather, I think that these machine learning models should supplement physics-based models in areas where they can improve process monitoring and optimization. Significant value is provided from many physics-based models, which, in many cases, can provide very accurate results.

Which features are the most important in impacting SDA DAO yields? As mentioned in Section 5.2.3, we can utilize SHAP to understand the importance of features in the dataset. We find the top eight features visualized in Figure 5-9. The temperature of the SDA extractor was previously noted as an important feature in many SDA DAO yield papers. However, the discussion of viscosity and upstream

90% recovery point were not well discussed in the literature. The remaining variables, feed specific gravity, DAO stripper bottoms temperature, extractor pressure, recycled solvent, and solvent-to-oil ratio all align with existing literature.

Using machine learning models, we can gain significant insight into the relative importance of these variables and their effects on DAO yields. This allows a refining professional to better understand the implications of their operational decisions and provides a jumping off point for future research into the SDA process.

Furthermore, as previously discussed, these machine learning models are more accurate than existing physics-based models, suggesting that the relationships between the features and the dependent variable are not adequately described in the literature.

How can we incorporate predictive machine learning analytics into an optimization framework to produce tangible business value? The predictive machine learning model was successfully integrated into a linear optimization framework and proved to be a profitable endeavor. However, there are several gaps in this analysis that could be addressed in future work.

First is the sequential nature of the decision framework utilized in this thesis. The framework considered in this thesis utilized a sequential framework where the following steps are taken, in the following order (Jacquillat, 2020):

1. Train a machine learning model on training data, to minimize in-sample loss.
2. Use machine learning model to make predictions on new data.
3. Solve deterministic optimization with an estimate.

This sequential approach has significant benefits as it is straightforward, interpretable, and easy to communicate to outside stakeholders. Furthermore, we can incorporate high complexity machine learning models with linear optimization methods in order to provide a cohesive model in sequence.

However, this approach did not address the concept of uncertainty. The linear optimization methodology comprised in Chapter 4 assumes that all quantities are precisely understood. In practice however, each of the parameters are understood

to have inherent uncertainty whether this is due to fluctuating operating conditions or economic circumstances. Therefore, the errors that exist in the machine learning model can propogate into the linear optimization framework. In truth, these parameters are at best understood probabilistically. In order to avoid falling into a flaw of averages, a stochastic programming model could be utilized to improve the overall profitability of the refinery by making a decision that acknowledges and considers the uncertainty in these parameters. A stochastic programming model which utilizes input data to both make predictions on the SDA DAO yields and utilizes this data for stochastic scenario generation can aid in addressing the uncertainty in both the machine learning predictions as well as the other parameters in the model (Bertsimas et al., 2020).

Additionally, we are attempting to minimize our loss function which attempts to minimize the error associated with our SDA DAO yield predictions. However for all downstream oil & gas refineries, the goal is to maximize profitability. The loss function for the SDA DAO machine learning models is not directly connected to the objective function of the optimization problem. Therefore, minimization of the prediction error should, but may not always, lead to the best prediction in the context of decision making.

6.2 Summary

This thesis reviewed through various machine learning models in an effort to provide continuous Deasphalted oil (DAO) yield estimations for a Solvent deasphalting (SDA) unit. These models were constructed using data sets from a commercial downstream oil and gas refinery in the United States which include plant operating parameters and laboratory measurements for feed properties. The machine learning models exhibit high out-of-sample R^2 values of 0.76.

Additionally, this predictive machine learning model was incorporated into a linear optimization framework for a hypothetical downstream refinery, improving profitability by $730,000 annually. The results of this model can be utilized for more accurate

plant monitoring within oil gas downstream refineries, as well as improved decision making by oil and gas planning professionals.

6.3 Future Work

There are many actions which can be taken in succession to this thesis that are listed below, broken by predictive modeling and linear optimization.

Future Work: Predictive Modeling

1. Notably this thesis utilized a dataset wherein the solvent composition was held constant and therefore would not impact the prediction of DAO yields. Adding features to the dataset which show the solvent composition can help further develop a holistic model for DAO yield predictions.

2. Expand predictive analytics to investigate stacked ensemble modeling and/or deep learning methodologies.

Future Work: Optimization

1. Consider uncertainty in model via development of a stochastic programming model.

2. Expand the linear optimization problems to consider additional crude oils. Explore the utility of this linear regression problems with more homogeneous and heterogeneous mixtures to best understand where it provides the most value.

3. Incorporate fluctuations of pricing of crude oils over time.

4. Incorporate robust optimization methodology into the linear optimization framework to ensure decisions that are feasible and an optimal solution for a worst-case objective function.

THIS PAGE INTENTIONALLY LEFT BLANK

References

ASTM. (2016). *ASTM D2270 Standard Practice for Calculating Viscosity Index from Kinematic Viscosity at 40 °C and 100 °C* (Standard). West Conshohocken, PA: American Society for Testing and Materials.

ASTM. (2020, May). *ASTM D341 Standard Practice for Viscosity-Temperature Equations and Charts for Liquid Petroleum or Hydrocarbon Products* (Standard). West Conshohocken, PA: American Society for Testing and Materials.

Baek, I., Kim, C., Kim, S., & Hong, S. (1993). Extraction of deasphalted oil from vacuum residue. *J. Energy Eng.*, *2*, 68-74.

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression diagnostics: Identifying influential data and sources of collinearity.*

Bertsimas, D., Boussioux, L., Cory-Wright, R., Delarue, A., Digalakis, V., Jacquillat, A., ... Zeng, C. (2020). From predictions to prescriptions: A data-driven response to covid-19. *medRxiv*. Retrieved from `https://www.medrxiv.org/content/early/2020/06/29/2020.06.26.20141127` doi: 10.1101/2020.06.26.20141127

Box, G., & Cox, D. (1964). An analysis of transformations. *Journal of the Royal Statistical Society B*, *26*, 211-252.

Brons, G., & Yu, J. M. (1995). Solvent deasphlating effects on whole cold lake bitumen. *Energy Fuels*, *9*, 641-647.

Brynjolfsson, E., & Hitt, L. (1996). Paradox lost? firm-level evidence on the returns to information systems spending. *Management science*, *42*(4), 541–558.

Brynjolfsson, E., Jin, W., & McElheran, K. (2021). The power of prediction: Predictive analytics, workplace complements, and heterogeneous firm performance1.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. , 785–794. Retrieved from `http://doi.acm.org/10.1145/2939672.2939785` doi: 10.1145/2939672.2939785

Crude oil prices today. (2021). Retrieved from `https://oilprice.com/`

EIA. (2021). *Gasoline and diesel fuel update - u.s. energy information administration (eia).* Retrieved from `https://www.eia.gov/petroleum/gasdiesel/`

ExxonMobil. (2020). *Assays available for download.* Retrieved from `https://corporate.exxonmobil.com/Crude-oils/Crude-trading/Assays-available-for-download`

Fahim, M. A., Al-Sahhaf, T., & Elkilani, A. (2010). *Fundamentals of petroleum refining.* Amsterdam, The Netherlands: Elsevier.

Gillis, D. B., & Tine, F. V. (1998). Uop foster wheeler team - uop llc. In *What is*

new in solvent deasphalting?

Greene, W. H. (2003). *Econometric analysis.*

Gurobi Optimization, L. (2021). *Gurobi optimizer reference manual.* Retrieved from `http://www.gurobi.com`

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning.* New York, NY, USA: Springer New York Inc.

IATA. (2021). *Jet fuel price monitor.* International Air Transport Association. Retrieved from `https://www.iata.org/en/publications/economics/fuel-monitor/`

Jacquillat, A. (2020, November). *Lecture notes in advanced analytics edge.* Massachusetts Institute of Technology.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With applications in r.* Springer Publishing Company, Incorporated.

Lee, J. M., Shin, S., , Ahn, S., Chun, J. H., Lee, K. B., ... Nho, N. S. (2014). Separation of solvent and deasphalted oil for solvent deasphalting process. *Fuel Processing Technology*, *119*, 204-210. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0378382013003664` doi: https://doi.org/10.1016/j.fuproc.2013.11.014

Leffler, W. L. (2008). *Petroleum refining in nontechnical language.* PennWell Corporation.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions..

Maples, R. E. (2008). *Petroleum refinery process economics.* PennWell Corporation.

Mckinsey energy insights, optimization. (n.d.). Retrieved from `https://www.mckinseyenergyinsights.com/resources/refinery-reference-desk/optimization/`

Mckinsey energy insights, products. (n.d.). Retrieved from `https://www.mckinseyenergyinsights.com/resources/refinery-reference-desk/refined-products/`

Ng, S. H. (1997, November). Nonconventional Residuum Upgrading by Solvent Deasphalting and Fluid Catalytic Cracking. *Energy & Fuels*, *11*(6), 1127–1136. Retrieved from `https://doi.org/10.1021/ef970010u` (Publisher: American Chemical Society) doi: 10.1021/ef970010u

Pandey, Y., Rastogi, A., Kainkaryam, S., Bhattacharya, S., & Saputelli, L. (2020). *Machine learning in the oil and gas industry.*

Pang, W., Lee, J.-K., Yoon, S.-H., Mochia, I., Ida, T., & Ushio, M. (2010). Compositional analysis of deasphalted oils from arabian crude and their hydrocracked products. *Fuel Processing Technology*, *91*, 1517-1524.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Robinson, P. (2007, 01). Petroleum processing overview. In (Vol. 1, p. 1-78). doi: 10.1007/978-0-387-25789-1_1

Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th python in science conference.*

Upton, G., & Cook, I. (1996). *Understanding statistics.* Oxford University Press.

Wright, M. N., & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17. doi: 10.18637/jss.v077.i01

Yeo, I.-K., & Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, *87(4)*, 949-959.