



ITS
Institut
Teknologi
Sepuluh Nopember



BACHELOR THESIS & COLLOQUIUM - ME234804

DEVELOPMENT OF MACHINE LEARNING MODEL ON ESTIMATING FUEL CONSUMPTION FOR KMP. KIRANA VII

JAMES JANTO JUNIOR

NRP 5019201101

SUPERVISORS

Ir. Hari Prastowo, M.Sc

NIP 196510301991021001

Ir. Dwi Priyanta, MSE.

NIP 196807031994021003

Double Degree Program

Department Of Marine Engineering

Faculty Of Marine Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2024

“This page is intentionally left blank”



BACHELOR THESIS & COLLOQUIUM - ME234804

DEVELOPMENT OF MACHINE LEARNING MODEL ON ESTIMATING FUEL CONSUMPTION FOR KMP. KIRANA VII

JAMES JANTO JUNIOR
NRP. 5019201101

IR. HARI PRASTOWO, M.SC
NIP 196510301991021001

IR. DWI PRIYANTA, MSE.
NIP 196807031994021003

**DOUBLE DEGREE PROGRAM
DEPARTMENT OF MARINE ENGINEERING
FACULTY OF MARINE TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024**

“This page is intentionally left blank”

APPROVAL SHEET

**DEVELOPMENT OF MACHINE LEARNING MODEL ON ESTIMATING FUEL
CONSUMPTION FOR KMP. KIRANA VII
BACHELOR THESIS**

Submitted to fulfil one of the requirements
For obtaining a bachelor's degree in engineering at
Undergraduate Study Program of Marine Engineering
Department of Marine Engineering
Faculty of Marine Technology
Institut Teknologi Sepuluh Nopember

By: James Janto Junior

NRP. 5019201101

Approved by Hochschule Wismar Germany:

Signature :

Representative : Wolfgang Busse, Dr-Ing.

Date : July 2024

Approved by Head of Marine Engineering Department:



Signature :

Head of Department : Beny Cahyono, S.T, M.T., Ph.D

NIP./NPP. : 197903192008011008

Date : July 2024

“This page is intentionally left blank”

APPROVAL SHEET

DEVELOPMENT OF MACHINE LEARNING MODEL ON ESTIMATING FUEL CONSUMPTION FOR KMP. KIRANA VII

Submitted To Fulfill One of The Requirements
For obtaining a bachelor's degree in engineering at
Undergraduate Study Program of Marine Engineering
Department of Marine Engineering
Faculty of Marine Technology
Institut Teknologi Sepuluh Nopember

By: **James Janto Junior**
NRP. 5019201101

Approved by Final Examiner Team:

- | | |
|---|---------------|
| 1. Ir. Hari Prastowo, M.Sc. | Supervisor |
| 2. Ir. Dwi Priyanta, MSE. | Co-Supervisor |
| 3. Dr. Eng. M. Badrus Zaman, S.T., M.T. | Examiner |
| 4. Dr. Eng. Trika Pitana, S.T., M.Sc. | Examiner |
| 5. Dr. Nurhadi Siswanto, S.T., M.T. | Examiner |



SURABAYA

July 2024

“This page is intentionally left blank”

STATEMENT OF ORIGINALITY

The undersign below:

Name of student/NRP : James Janto Junior/5019201101
Department : Marine Engineering
Supervisor I : Ir. Hari Prastowo, M.Sc.
Supervisor II : Ir. Dwi Priyanta, MSE.

Hereby declare that the Final Project with the title of "DEVELOPMENT OF MACHINE LEARNING MODEL ON ESTIMATING FUEL CONSUMPTION FOR KMP. KIRANA VII" is the result of my own work, is original, and is written by the following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

Surabaya,

Student,



James Janto Junior
NRP. 5019201101

Acknowledge
Supervisor 1



Ir. Hari Prastowo, M.Sc.
NIP. 196510301991021001

Acknowledge
Supervisor 2



Ir. Dwi Priyanta, MSE.
NIP. 196807031994021003

“This page is intentionally left blank”

ABSTRACT

DEVELOPMENT OF MACHINE LEARNING MODEL ON ESTIMATING FUEL CONSUMPTION FOR KMP. KIRANA VII

Name / NRP : James Janto Junior / 5019201101
Departement : Marine Engineering FTK-ITS
Supervisors : Ir. Hari Prastowo, M.Sc
: Ir. Dwi Priyanta, MSE.

Abstract

Fuel consumption in shipping operations constitutes a substantial portion of operational costs and contributes to environmental pollution. Efficiently estimating fuel consumption is crucial for optimizing vessel operations, reducing operational expenses, and minimizing the environmental impact of ships. This study focuses on combining fuel optimization and machine learning, which is a subfield of artificial intelligence, offers powerful tools for analyzing complex data patterns and making accurate predictions. By training models on historical data, machine learning algorithms can learn the relationships between various factors and fuel consumption, enabling the development of accurate predictive models. This study is using 2 algorithms, Ridge regressor and XGBoost. Models made from both algorithms respectively have a R^2 of 0,886 and 0,9306. XGBoost model is much more accurate than using ridge regressor. The model can be more developed overtime by inputing more datas.

Keywords: Machine Learning, R^2 , Ridge Regressor, XGBoost

“This page is intentionally left blank”

ABSTRAK

PENGEMBANGAN MODEL MACHINE LEARNING PADA ESTIMASI KONSUMSI BAHAN BAKAR UNTUK KMP. KIRANA VII

Nama : James Janto Junior
NRP : 5019201101
Departemen : Teknik Sistem Perkapalan FTK-ITS
Pembimbing : Ir. Hari Prastowo, M.Sc
: Ir. Dwi Priyanta, MSE.

Abstrak

Konsumsi bahan bakar dalam operasi pelayaran merupakan bagian substansial dari biaya operasional dan berkontribusi terhadap pencemaran lingkungan. Memperkirakan konsumsi bahan bakar secara efisien sangat penting untuk mengoptimalkan operasi kapal, mengurangi biaya operasional, dan meminimalkan dampak lingkungan dari kapal. Penelitian ini berfokus pada penggabungan optimasi bahan bakar dan pembelajaran mesin, yang merupakan subbidang dari kecerdasan buatan, yang menawarkan alat yang ampuh untuk menganalisis pola data yang kompleks dan membuat prediksi yang akurat. Dengan melatih model pada data historis, algoritma pembelajaran mesin dapat mempelajari hubungan antara berbagai faktor dan konsumsi bahan bakar, sehingga memungkinkan pengembangan model prediksi yang akurat. Penelitian ini menggunakan 2 algoritma, yaitu Ridge regressor dan XGBoost. Model yang dibuat dari kedua algoritma tersebut masing-masing memiliki R^2 sebesar 0,886 dan 0,9306. Model XGBoost jauh lebih akurat dibandingkan dengan menggunakan ridge regressor. Model ini dapat dikembangkan lebih lanjut dengan memasukkan lebih banyak data.

Kata kunci: *Machine Learning, R^2 , Ridge Regressor, XGBoost*

“This page is intentionally left blank”

PREFACE

All praise to the Almighty God, for all his blessings, the author can start, work, finish this bachelor thesis. For the author, this bachelor thesis represents an attempt to contribute to efforts in reducing fuel consumption. Nowadays, everything is digitalized by then autor try to implementing technology to marine side. By predicting fuel consumption of the ship, author hope that it will also reducing not only the cost for shipping company, but also prolong planet sustainability for the next generation to come. This thesis could not be completed without helps from others. The author would like to acknowledge people who helped and contributed in the process of this bachelor thesis completion, among others:

1. The author's beloved parents, for all their unconditional love, prayers, and supports.
2. Ir. Hari Prastowo, M.Sc., and Ir. Dwi Priyanta, MSE. as supervisors that have spent the time giving guidance and direction in this thesis completion.
3. Beny Cahyono, S.T., M.T., Ph. D., as the head of Marine Engineering Department, ITS, Surabaya.
4. All lecturers of the Marine Engineering Department, ITS, Surabaya, Indonesia, and Hochschule Wismar, Germany, who give a lot of knowledge and logical thinking in engineering.
5. Fleet Division of PT.DLU who helped the author to access shipping data.
6. Member of Digital Marine Operational and Maintenance Lab who always give advises and supports since the day one.

The author hopes that the writing of the Final Project Proposal can be useful and provide information to the reader. Because of the limitations of the author, constructive criticisms and suggestions are indispensable for perfection in this report.

“This page is intentionally left blank”

TABLE OF CONTENTS

APPROVAL SHEET	i
APPROVAL SHEET	iii
STATEMENT OF ORIGINALITY	v
ABSTRACT	vii
ABSTRAK	ix
TABLE OF CONTENTS	xiii
LIST OF TABLES	xv
LIST OF FIGURES.....	xvii
LIST OF ABBREVIATIONS	xix
LIST OF SYMBOL.....	xxi
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Research Objectives	3
1.4 Scope of Study	3
1.5 Research Benefit	3
CHAPTER 2 LITTERATURE REVIEW	5
2.1 Related Study	5
2.2 Ship Fuel Consumption.....	5
2.3 Machine Learning	6
2.4 Machine Learning Algorithms	6
2.4.1 XGBoosting Model.....	6
2.4.2 SVM (Support Vector Machine).....	7
2.4.3 Perceptron	8
2.4.4 Multiple Linear Regression	9
2.4.5 K-Nearest Neighbors	9
2.4.6 Ridge Regression	10
2.4.7 Linear Regression	11
2.4.8 Lasso Regression	12
2.4.9 Random Forest.....	13
2.5 Python Language.....	14
2.6 Supervised Machine Learning.....	15
2.7 Unsupervised Machine Learning	15
2.8 Decicion Tree.....	15
2.9 R-Squared.....	16
2.10 Scikit-Learn.....	17
2.11 Google Collab	17
2.12 Visual Studio.....	17
CHAPTER 3 RESEARCH METHODOLOGY.....	19
3.1 Research Flowchart.....	19
3.2 Data Collecting and Database	20
3.3 Choosing 1 Algorithm for Model Development	20
3.4 Data Processing and Making of the Coding.....	20
3.5 Feature Preprocessing preventing errors to the model	21
3.6 Model Testing	21
3.7 Choosing The Best Algorithm	22
3.8 Make predictions for the ship’s fuel consumption	22
CHAPTER 4 RESULT AND DISCUSSION	25

4.1	Data Acquisition.....	25
4.2	Choosing Algorithm.....	26
4.3	Preprocessing	27
4.4	Processed Dataset.....	35
4.5	Data Input.....	41
4.6	Model Training	42
4.7	Ridge Regression	43
4.8	XGBoost.....	50
4.9	Discussion	57
CHAPTER 5	Conclusion	59
5.1	Conclusion	59
5.2	Suggestion.....	59
REFERENCES.....		61
ATTACHMENT A		63
ATTACHMENT B.....		65
ATTACHMENT C.....		67
ATTACHMENT D		83
ATTACHMENT E.....		91
ATTACHMENT F		101
AUTHOR BIOGRAPHY		111

LIST OF TABLES

Table 4.1 Data Sample	41
Table 4.2 Ridge Regression Sample	48
Table 4.3 XGBoost Regression sample value	55
Table 5.1 XGBoost Prediction values	92
Table 5.2 Ridge regression values difference	102

“This page is intentionally left blank”

LIST OF FIGURES

Figure 1.1.1 Increasing Shipping CO ₂ Emissions (UNCTAD,2023).....	1
Figure 2.1 Gradient Boosting Model (Cheng et al., 2020).....	7
Figure 2.2 Support Vector Machine model, (spiceworks.com).....	7
Figure 2.3 Plot iris SVM (Scikit, 2011)	8
Figure 2.4 Perceptron works, source (towardsdatascience.com)	8
Figure 2.5 Multiple Linear Regression model (source: python.plainenglish.io).....	9
Figure 2.6 K-Nearest Neighbors model (source: towardsdatascience.com)	10
Figure 2.7 Ridge Regression, source (pianalytix.com)	11
Figure 2.8 Linear Regression, Source: (kdnuggets.com)	12
Figure 2.9 Lasso Regression, source: (medium.com)	13
Figure 2.10 Random Forest algorithm (source: anasbrital98.github.io)	14
Figure 2.11 Decision Tree Regression Model (Scikit, 2011).....	16
Figure 3.1 Flow Chart	19
Figure 4.1 KM. Kirana VII Ship	25
Figure 4.2 Ridge Regression algorithm code	26
Figure 4.3 XGBoost Regression code	27
Figure 4.4 List of data gathered	28
Figure 4.5 X variables scales.....	28
Figure 4.6 Pre-processed Sailing Speed.....	29
Figure 4.7 Pre-processed Beaufort scale	30
Figure 4.8 Pre-processed Displacement	31
Figure 4.9 Pre-processed Sea Condition	32
Figure 4.10 Pre-processed Draft.....	33
Figure 4.11 Pre-processed Graph Fuel Consumption	34
Figure 4.12 Eliminating 0 values	35
Figure 4.13 Cleaned Sailing speed data	36
Figure 4.14 Cleaned Beaufort scale data.....	37
Figure 4.15 Cleaned displacement data.....	38
Figure 4.16 Cleaned sea condition data.....	39
Figure 4.17 Cleaned draft data	40
Figure 4.18 Cleaned Fuel consumption data	41
Figure 4.19 Values of the datas inputed	42
Figure 4.20 Train data and testing data splitting	43
Figure 4.21 Ridge Regression Value.....	43
Figure 4.22 Scatter plot of the Actual vs Predicted values for ridge regression	44
Figure 4.23 Ridge tested vs prediction.....	45
Figure 4.24 Ship Speed vs Fuel Consumption Prediction.....	45
Figure 4.25 Draft vs Fuel Consumption Prediction	46
Figure 4.26 Result XGBoost	50
Figure 4.27 Scatter plot of fuel oil consumption original vs predicted XGBoost regression.....	51
Figure 4.28 XGBoost Regressor Predictions vs target.....	51
Figure 4.29 Relation between Ship speed vs Fuel consumption prediction.....	52
Figure 4.30 Draft vs Fuel Consumption Prediction	53
Figure 4.31 XGBoost original and predicted values sample	54

“This page is intentionally left blank”

LIST OF ABBREVIATIONS

RF	: Random Forest
SVM	: Support Vector Machine
RR	: Ridge Regression
BHP	: Break Horse Power
V _s	: Ship Service Speed
wsa	: Wetted Surface Area
DT	: Decision Tree
GBoost	: Gradient Boost
MLR	: Multiple Linear Regression
ML	: Machine Learning

“This page is intentionally left blank”

LIST OF SYMBOL

y_i	- dependent variable
x_i	- explanatory variable
Ω	- y-Intercept
A	- Alpha
Λ	- Gamma

“This page is intentionally left blank”

CHAPTER 1 INTRODUCTION

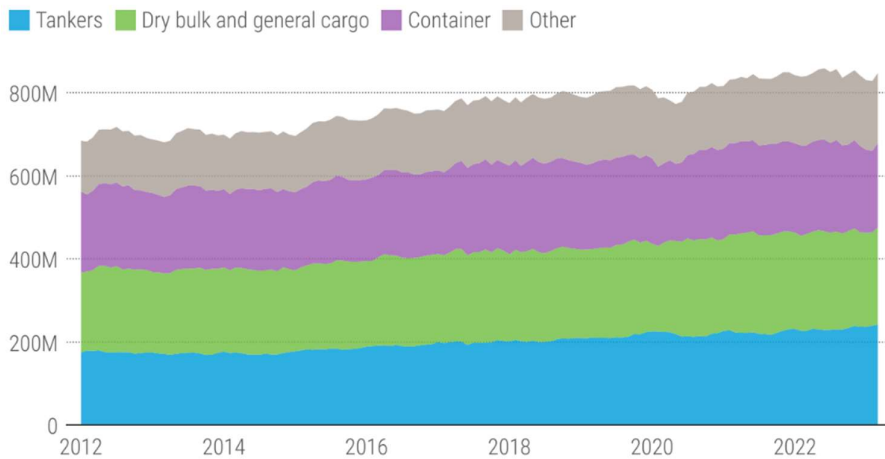
1.1 Background

The maritime industry plays a vital role in global trade and transportation at around 80% of the total world trade (Dere and Deniz, 2019; UNCTAD, 2020), serving as the backbone of the global economy. However, the industry also faces significant challenges, including rising fuel costs, stringent environmental regulations, and the urgent need to reduce greenhouse gas emissions (Joung et al, 2020). Fuel consumption in shipping operations constitutes a substantial portion of operational costs and contributes to environmental pollution.



Shipping emissions are headed in the wrong direction

Carbon dioxide emissions by main vessel types, tons, 2012–2023



Note: The group "other" includes vehicles and roll-on/roll-off ships, passenger ships, offshore ships and service and miscellaneous ships.

Source: UNCTAD based on data provided by Marine Benchmark, June 2023.

Figure 1.1.1 Increasing Shipping CO₂ Emissions (UNCTAD,2023)

Efficiently estimating fuel consumption is crucial for optimizing vessel operations, reducing operational expenses, and minimizing the environmental impact of ships. Traditionally, fuel consumption estimation has relied on manual calculations and empirical formulas, which may lack accuracy and fail to consider the intricate interplay of multiple factors influencing fuel consumption. To address these limitations and enhance the accuracy of fuel consumption estimation (Xie et al., 2023).

Machine learning, a subfield of artificial intelligence, offers powerful tools for analyzing complex data patterns and making accurate predictions. By training models on historical data, machine learning algorithms can learn the relationships between various factors and fuel consumption, enabling the development of accurate predictive models. These models can then be used to estimate fuel consumption for different operational scenarios, providing valuable insights for optimizing vessel performance and fuel efficiency (Uyanık et al., 2020). The primary objective of this thesis is to develop and evaluate a machine learning model for estimating the fuel consumption of KMP. Kirana VII . By leveraging available data on vessel characteristics, operational parameters, weather conditions, and historical fuel consumption records, this research aims to create a predictive model that can accurately estimate fuel consumption in real-time.

The implementation of machine learning techniques for fuel consumption estimation on KMP. Kirana VII holds significant potential benefits. First, it can provide KMP. Kirana VII with a more accurate and reliable tool for estimating fuel consumption, enabling better decision-making regarding operational planning, route optimization, and fuel management. Second, by identifying the most influential factors affecting fuel consumption, the model can help uncover opportunities for fuel efficiency improvements and emission reductions.

Furthermore, this research contributes to the broader field of maritime operations by showcasing the applicability and effectiveness of machine learning in addressing fuel consumption estimation challenges. The findings and recommendations from this thesis can serve as a valuable resource for other shipping companies seeking to enhance their fuel efficiency, reduce operational costs, and meet environmental sustainability goals. By developing and evaluating a machine learning model for estimating the fuel consumption , this research seeks to provide an innovative and data-driven approach to optimize vessel operations, reduce fuel costs, and minimize the environmental impact of shipping operations (Kim et al., 2021)

1.2 Problem Statement

The key problems that could be explored:

- How does machine learning algorithm work for estimating fuel consumption of the KMP. Kirana VII ?
- What is the variable to determine the fuel consumption at KMP. Kirana VII ?

- What's the machine learning algorithm that has the highest R^2 for KMP. Kirana VII ?

1.3 Research Objectives

- Determine the process of machine learning algorithm for estimating fuel consumption of the KMP. Kirana VII .
- Determining the variable of fuel consumption at KMP. Kirana VII .
- Comparing the highest R^2 for every machine learning method of the ships.

1.4 Scope of Study

- Machine learning program used for estimating the fuel consumption is using python language.
- Machine learning model used for estimating the fuel consumption is using Ridge, and XGBoost.
- The object of the research is focused on KMP. Kirana VII based on the ship data and log data.

1.5 Research Benefit

- Can be a reference which is aimed at ship operation efficiency and ship fuel estimation.
- The model of machine learning could be used to provide real time data as it could be further improved.

“This page is intentionally left blank”

CHAPTER 2

LITTERATURE REVIEW

2.1 Related Study

The previous research is focusing on the aspects of machine learning methods such as using prediction model Artificial Neural Network (ANN) and using the multiple linear regression (MLR). Where the ANN records and produces the best score prediction for the precision accuracy. On which the research here can try to alternate the algorithm thus using another algorithm for other ship. Which is used on a 13,000 TEU class container ship. Models used such as artificial neural network (ANN) present the accuracy goodness ranged from 0.9709 to 0.9936. the sensitivity analysis of the draught indicated in 14.79, which was very close to the design draught of the ship, thus resulting optimal fuel consumption efficiency (Kim et al., 2021).

Another study was using the Vector Machine algorithm, which results achieved a higher accuracy with R-Squared metric value of 0.97 than other related work using the same Support Vector Machine algorithm. The author concludes that using Support Vector Machine has a great effect when used for fuel consumption (Hamed et al., 2021). Other study used R^2 , XGBoost, and Random Forest (RF). The values are 0.9977 for the R^2 , for the XGBoost and RF are 0.9922 (Xie et al., 2023). From the research of the main purpose is to find the higher the R^2 which means the lower the error margin.

Other studies used parameters such as main engine, cargo weight, ship draft, sea condition, weather condition, etc. The study uses models such as Multiple Linear Regression, Ridge and Lasso Regression, Support Vector Regression, Tree-Based Algorithms, Boosting Algorithms are used for prediction of fuel consumption in on container ship. The study concludes that nearest estimation of the actual fuel consumption data is made by multiple linear regression and ridge regression on which the root mean square error is 0,001, and 0,002 mean absolute error thus the determination score coefficient is 99.9% (Uyanik et al., 2020)

2.2 Ship Fuel Consumption

Ship Fuel Consumption is contributed to function such as the resistance of the ship, payload, ship displacement, ship speed, ship draft, main engine load, etc. Average of ship fuel consumption depends on the ship's type & engine power on which the correlation between these

is not linear (Wijanarko., 2020). Ship speed and power is not linear, which the power of the ship is increased with speed squared. Therefore, optimal fuel consumption is crucial for realizing energy efficiency management, cost reduction, and efficiency improvement of operational ships (Fan et al., 2022).

2.3 Machine Learning

Machine learning is a field of study that focuses on the development of algorithms and statistical models that enable computer systems to learn and improve from experience without being explicitly programmed. A bibliometric analysis conducted by Jordan and Mitchell (2015) examined the growth and trends in machine learning research. The analysis revealed a significant increase in the number of publications related to machine learning over the years, indicating the growing interest and importance of this field. The study also highlighted the key areas of research within machine learning, such as supervised learning, unsupervised learning, and reinforcement learning. Additionally, the analysis identified the most influential researchers and institutions in the field, further highlighting the collaborative nature of machine learning research.

2.4 Machine Learning Algorithms

One commonly used algorithm is linear regression, which can establish a relationship between fuel consumption and various input variables such as ship speed, engine power, and cargo load. Decision trees are another powerful algorithm that can be applied in this context. They create a tree-like model of decisions and their possible consequences. Furthermore, ensemble methods like random forests (RF) and gradient boosting (GBoost) can be employed to improve the accuracy and robustness of fuel consumption predictions. Support vector machines (SVM) can also be utilized to classify and predict fuel consumption based on given variables.

2.4.1 XGBoosting Model

The gradient boosted trees has been around for a while, and there are a lot of materials on the topic. This tutorial will explain boosted trees in a self-contained and principled way using the elements of supervised learning. The XGBoost model concept is also using ensemble which is decision tree. (Cheng et al., 2020).

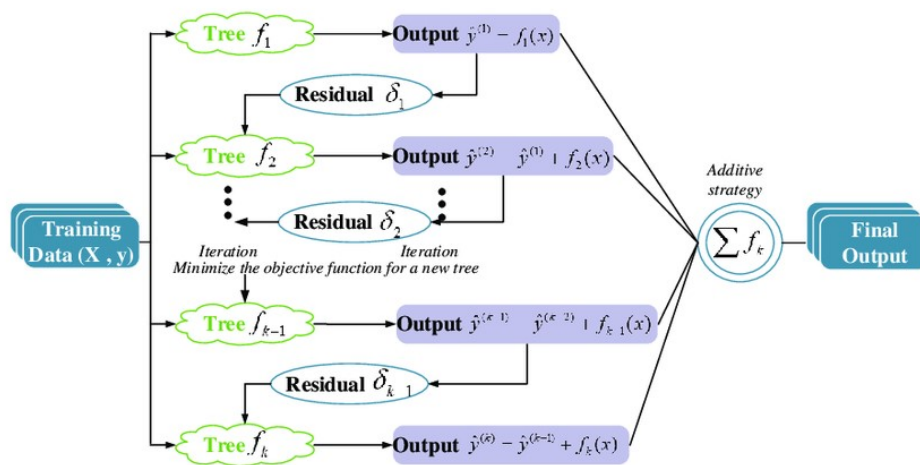


Figure 2.1 Gradient Boosting Model (Cheng et al., 2020)

2.4.2 SVM (Support Vector Machine)

Support vector machine (SVM) that can be used as classification and regression is one of the supervised learning models. The data which is entered in this algorithm are transformed into a feature area where the solution based on optimization techniques takes place. SVMs have effective results on proper generalization, accuracy, and precision (Alsarraf et al., 2019).

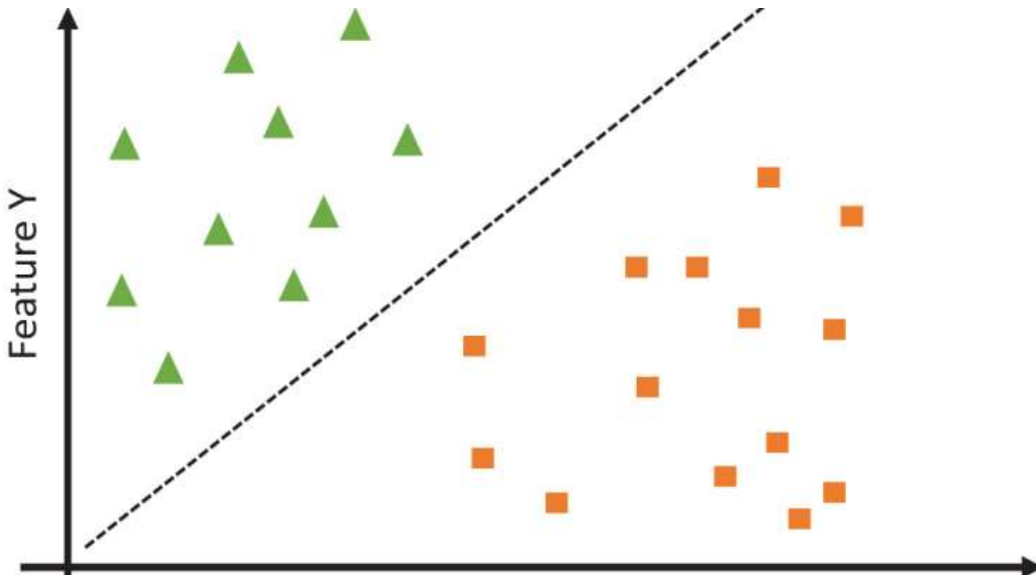


Figure 2.2 Support Vector Machine model, (spiceworks.com)

The Support Vector Regression (SVR) was proposed by (Smola and Schölkopf, 2004) as a version of the SVM. SVR performs a solution by preserving all the main features

that characterize the algorithm, but it does this with a few differences from the SVM. The SVR is approximately set to a tolerance margin. The purpose of adjusting this margin is to minimize the error and individualize the hyperplane that maximizes the margin.

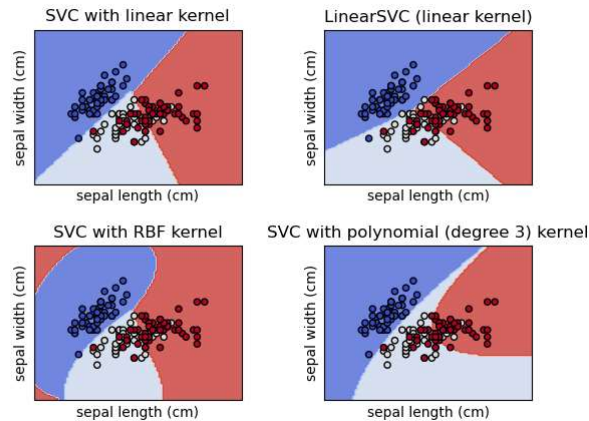


Figure 2.3 Plot iris SVM (Scikit, 2011)

2.4.3 Perceptron

Perceptron provides the first methods to iteratively learn classification of data's to discrete categories. To classify n-dimensional data, the Perceptron learns a weight vector with n parameters as well as a bias term.

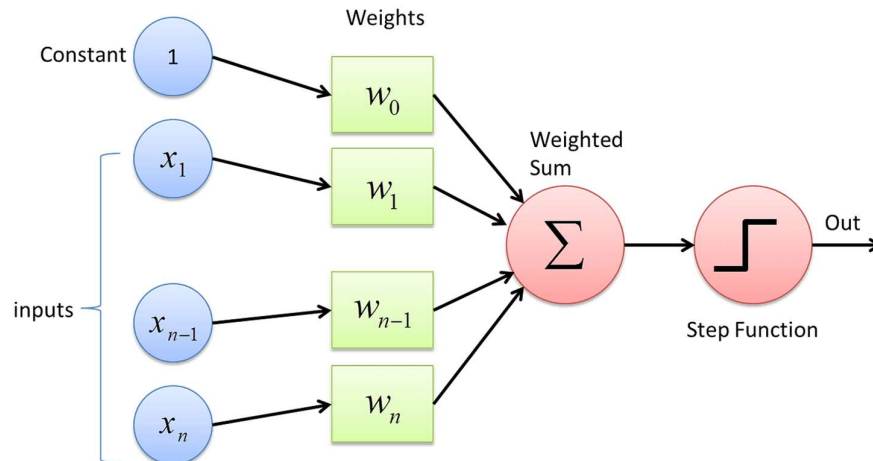


Figure 2.4 Perceptron works, source (towardsdatascience.com)

Both the weight vector and bias consist of positive integers greater than or equal to zero which encode a linear hyperplane separating two or more categories in n-dimensional space. (Kelby, 2020)

2.4.4 Multiple Linear Regression

Multiple Linear Regression (MLR) is a mathematical model that can estimate a dependent output with multiple variable inputs. The dependent variable y can be found by the following formula (Bocchetti et al., 2013):

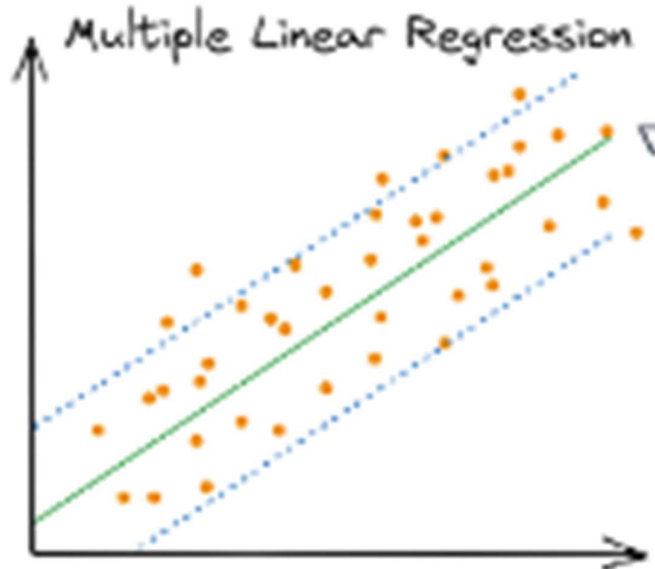


Figure 2.5 Multiple Linear Regression model (source: python.plainenglish.io)

$$y = \omega_0 + \omega_1x_1 + \omega_2x_2 + \omega_3x_3 = + + + \dots + \omega_nx_n$$

2.4.5 K-Nearest Neighbors

It is one of the nonparametric models, and the K-nearest neighbors (KNN) algorithm determines the k nearest neighbors to a given point (x_q) as distance-wise (Russell and Norvig, 2003) and k parameter can be selected by the user ($n_neighbors$). The KNN algorithm is varied from others because an action is performed on the data while classifying instead of training by the past acquired dataset of the system.

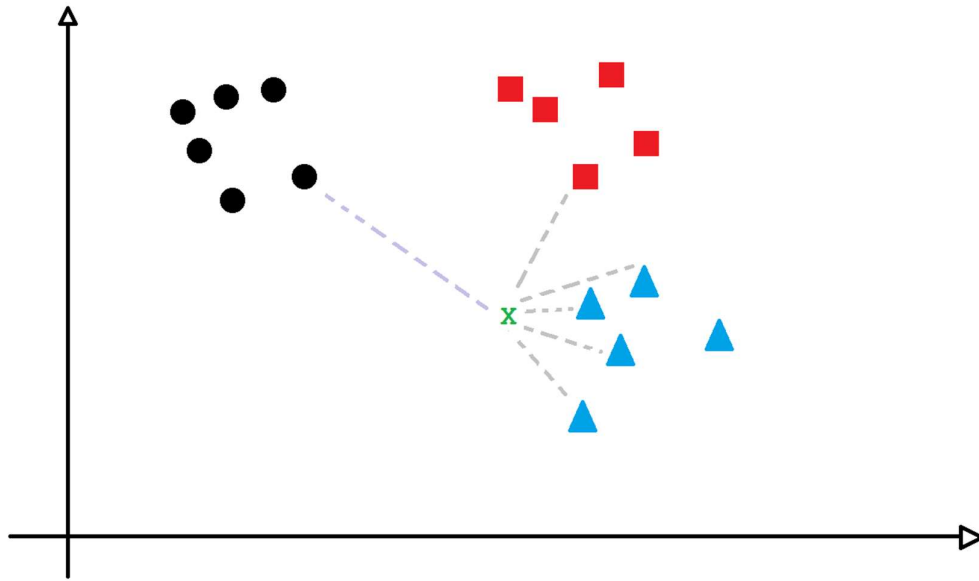


Figure 2.6 K-Nearest Neighbors model (source: towardsdatascience.com)

In Scikit-learn implementation, the calculation of the closest neighbors is determined automatically according to the input values. Algorithm hyperparameter is the leaf size (`leaf_size`) in which the algorithm speed and memory usage are affected when the nonbrute-force approach is used. Minkowski distance (LP) is used for the determination of the distance between a given point (x_q) and the other any point (x_j).

2.4.6 Ridge Regression

Ridge Regression (RR) is a powerful statistical technique that employs shrinkage to improve the estimation of coefficients in linear regression models, particularly when dealing with multicollinearity among predictor variables. Unlike traditional least squares methods, which can produce highly variable coefficient estimates when predictors are correlated, Ridge Regression adds a penalty term to the loss function.

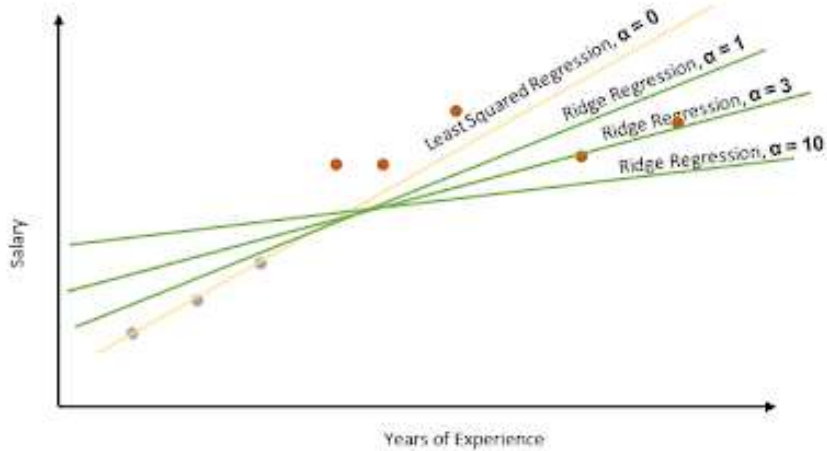


Figure 2.7 Ridge Regression, source (pianalytix.com)

This penalty is proportional to the square of the magnitude of the coefficients, effectively shrinking them towards zero. This process not only stabilizes the estimates but also enhances the model's predictive performance by reducing overfitting, especially in situations where the number of predictors is large relative to the number of observations.

$$\alpha = \lambda_1 + \lambda_2$$

$$\lambda \text{ratio} = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

2.4.7 Linear Regression

Linear regression is a foundational statistical method in machine learning that models the relationship between a dependent variable and one or more independent variables, assuming a linear relationship between them. Linear regression can be categorized into simple linear regression, which involves a single independent variable, and multiple linear regression, which encompasses two or more independent variables. For the model to be valid, it must meet several key assumptions, including linearity, independence, homoscedasticity, and normality of residuals. Common evaluation metrics for linear regression models include Mean Squared Error (MSE) and R-squared (R^2), which help assess the model's predictive performance.

The applications of linear regression are vast and span multiple fields, from economics to education. For instance, it can be utilized to predict economic indicators like GDP based on historical data or to forecast academic performance based on various

factors such as study habits and prior achievements (Predicting Students' Academic Performance Using Linear Regression, 2021). In educational research, linear regression models have been employed to identify the factors influencing students' academic success, enabling targeted interventions for at-risk students. Despite its limitations, such as sensitivity to outliers and the assumption of linearity, linear regression remains a robust and interpretable framework for understanding and predicting outcomes across diverse applications (Overview of Machine Learning Concepts, 2021).

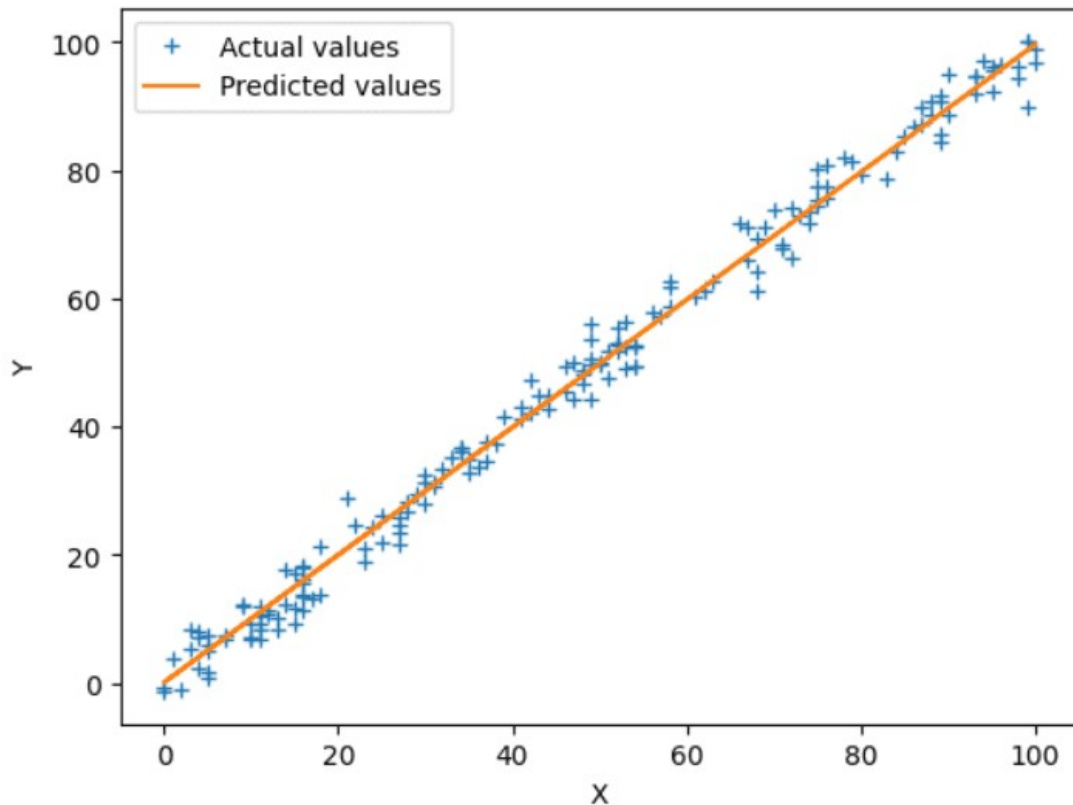


Figure 2.8 Linear Regression, Source: (kdnuggets.com)

2.4.8 Lasso Regression

Lasso regression, or Least Absolute Shrinkage and Selection Operator, is a regularization technique used in linear regression to prevent overfitting and enhance model interpretability by imposing a penalty on the absolute size of the regression coefficients (Tibshirani, 1996). This method adds an L1 penalty to the loss function, which encourages sparsity in the model by shrinking some coefficients to zero, effectively performing variable selection (Hastie, Tibshirani, & Friedman, 2009). The regularization parameter, lambda (λ), controls the strength of this penalty; larger values of λ increase the amount of shrinkage, leading

to simpler models with fewer predictors retained. Lasso regression is particularly useful in high-dimensional datasets where multicollinearity is a concern, as it helps to mitigate the effects of correlated predictors and improves the model's generalization to new data (Zou & Hastie, 2005).

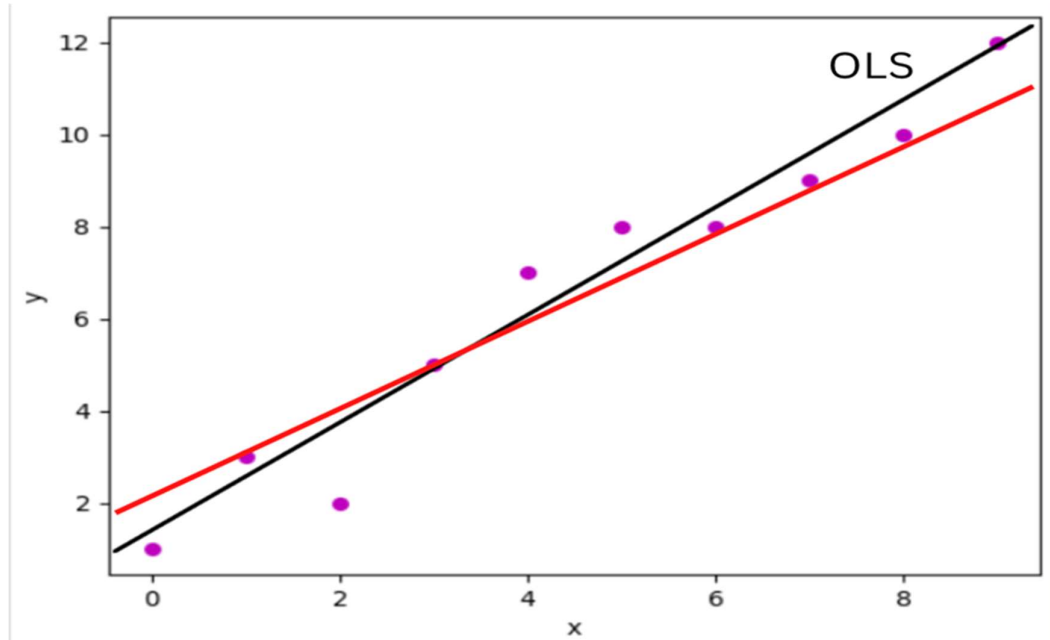


Figure 2.9 Lasso Regression, source: (medium.com)

2.4.9 Random Forest

The Random Forest algorithm has gained significant traction in recent years as a robust ensemble learning technique, particularly noted for its ability to handle high-dimensional data and complex interactions among features. Recent studies have demonstrated its effectiveness in various applications, such as predicting graduation outcomes and detecting misinformation in social media during elections (Indra et al., 2024; Su et al., 2023). The algorithm operates by constructing multiple decision trees using bootstrapped subsets of the training data, which enhances its predictive performance through the averaging of individual tree outputs (Liaw & Wiener, 2002).

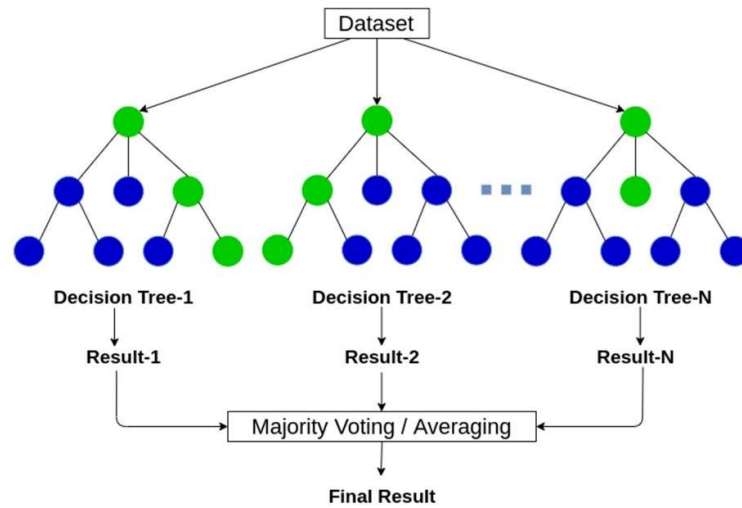


Figure 2.10 Random Forest algorithm (source: anasbrital98.github.io)

Furthermore, advancements like the Rotation Double Random Forest have been proposed to improve model diversity and accuracy by transforming input data, thereby addressing challenges such as class imbalance (Purnomo et al., 2024). Overall, Random Forest remains a preferred choice in machine learning due to its simplicity, interpretability, and strong performance across diverse datasets (Zhou, 2022).

2.5 Python Language

Python is a widely-used programming language known for its simplicity, readability, and versatility. In recent years, there has been a surge of interest in Python, as evidenced by the growing number of scholarly publications related to the language (Milmann and Avaizis, 2011).

Python is a widely-used programming language that has gained significant popularity in recent years. Its simplicity, readability, and versatility have made it an attractive choice for developers across various domains. The language's ability to seamlessly integrate with other tools and libraries, making it a powerful tool for scientific computing, data analysis, and visualization. Furthermore, the study found that Python's popularity is not limited to academia, as evidenced by its adoption in industry for web development, automation, and artificial intelligence applications.

2.6 Supervised Machine Learning

Supervised learning is a fundamental approach in machine learning that involves training a model to make predictions or decisions based on labeled data. In this paradigm, the model learns from a dataset consisting of input-output pairs, where each input is associated with a corresponding desired output. Most commonly, supervised learning leaves the probability for input undefined, such as an input where the expected output is known. This process provides dataset consisting of features and labels (Nasteski, 2017). This acquired knowledge enables the model to make predictions on new, unlabeled data by applying the learned rules or patterns. Supervised learning encompasses a wide range of algorithms, including decision trees, support vector machines, and neural networks, each with its own strengths and weaknesses. By leveraging the labeled data, supervised learning enables machines to perform tasks such as classification, regression, and even complex tasks like natural language processing and image recognition.

2.7 Unsupervised Machine Learning

Unsupervised learning is a branch of machine learning that focuses on extracting patterns and structures from unlabeled data. Unlike supervised learning, unsupervised learning algorithms do not rely on predefined labels or target outputs. Instead, they aim to uncover inherent relationships and discover hidden patterns within the data itself. By exploring the characteristics and similarities among the data points, unsupervised learning algorithms can identify clusters, anomalies, or latent variables that may exist in the dataset. These algorithms utilize various techniques such as clustering, dimensionality reduction, and generative modeling to uncover meaningful insights. Unsupervised learning plays a crucial role in exploratory data analysis, data pre-processing, and feature engineering. It enables machines to autonomously learn from vast amounts of unstructured data and discover underlying structures or patterns that may not be apparent to human observers. This can lead to valuable discoveries, insights, and improved decision-making in fields such as anomaly detection, customer segmentation, and recommendation systems.

2.8 Decision Tree

Decision trees are powerful and widely used tools in machine learning and data analysis. These models employ a tree-like structure to make predictions or decisions based on a series of hierarchical, binary splits on input features thus as classifiers Charbuty, B., & Abdulazeez, A.

(2021). Decision trees are particularly useful for classification and regression tasks, as they can handle both categorical and numerical data. They offer several advantages, such as interpretability, scalability, and ease of implementation. The interpretability of decision trees allows users to understand and explain the decision-making process, making them valuable in domains where transparency is crucial. Decision trees are also scalable, capable of handling large datasets efficiently. Furthermore, their simplicity and intuitive nature enable users to build decision tree models with relative ease, even without extensive programming or mathematical knowledge.

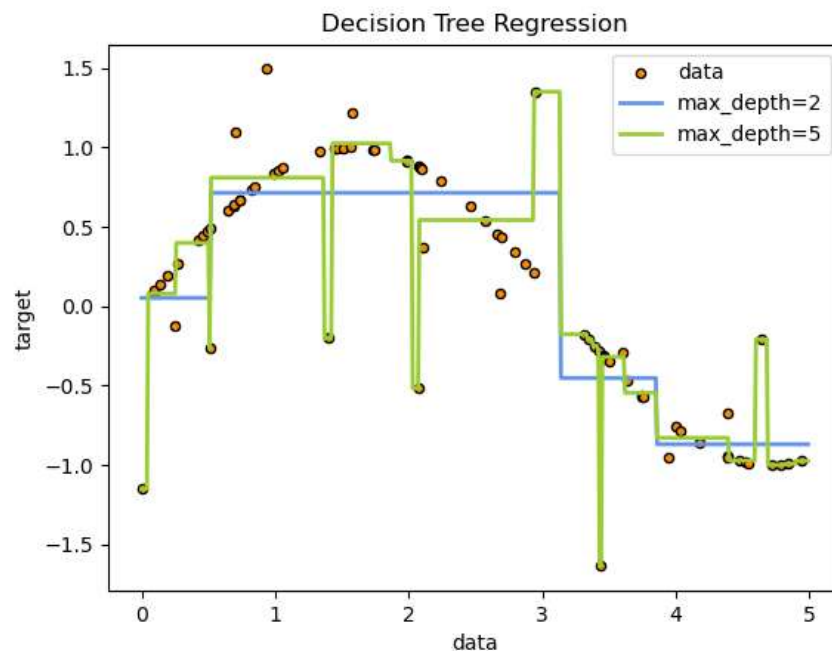


Figure 2.11 Decision Tree Regression Model (Scikit, 2011)

2.9 R-Squared

R^2 , the coefficient of determination, is a statistical measure that indicates how well a model fits the observed data. It is calculated as the proportion of the variance in the dependent variable that is predictable from the independent variables. R^2 values range from 0 to 1, where 0 indicates no relationship and 1 indicates a perfect fit. In machine learning, a high R^2 value suggests that the model can accurately predict the dependent variable based on the independent

variables, making it a crucial metric for evaluating model performance and selecting the best model for a given dataset.

R^2 , the coefficient of determination, is a statistical measure that indicates how well a model fits the observed data (Kohavi & Provost, 1998). It is calculated as the proportion of the variance in the dependent variable that is predictable from the independent variables (Hastie, Tibshirani, & Friedman, 2009). R^2 values range from 0 to 1, where 0 indicates no relationship (Kohavi & Provost, 1998) and 1 indicates a perfect fit (Hastie et al., 2009). In machine learning, a high R^2 value suggests that the model can accurately predict the dependent variable based on the independent variables (Kohavi & Provost, 1998), making it a crucial metric for evaluating model performance and selecting the best model for a given dataset (Hastie et al., 2009).

2.10 Scikit-Learn

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It provides a wide range of tools and algorithms for various tasks in machine learning, including classification, regression, clustering, dimensionality reduction, and model selection. It ranges from Data preprocessing, Feature extraction and selection, Model training and evaluation, Ensemble methods, Dimensionality reduction, and Model interpretation. scikit-learn is a versatile library that can support various aspects, from data preprocessing to model training, evaluation, and interpretation (Pedregosa *et al*,2011).

2.11 Google Collab

Google Colab, short for Colaboratory, is an innovative online platform provided by Google that enables users to write, run, and share Python code within a Jupyter Notebook environment. This cloud-based service offers free access to a variety of computing resources, including CPUs, GPUs, and TPUs, making it particularly advantageous for tasks related to machine learning and data analysis. One of the key features of Google Colab is its collaborative nature, allowing multiple users to work on the same notebook simultaneously, which enhances teamwork and facilitates group projects.

2.12 Visual Studio

Microsoft Visual Studio is a powerful Integrated Development Environment (IDE) developed by Microsoft, designed to facilitate software development across various platforms and programming languages, including C#, C++, Visual Basic, and Python. Launched in 1997

and continually evolving, Visual Studio offers a rich set of features that enhance productivity, such as IntelliSense for code suggestions, advanced debugging tools, and a user-friendly Solution Explorer for project organization. It supports collaborative development through integration with version control systems like Git, allowing multiple developers to work on the same project simultaneously. Available in three editions—Community, Professional, and Enterprise—Visual Studio caters to individual developers, small teams, and large organizations, making it a versatile choice for creating high-quality applications efficiently. Its extensive capabilities and customizable environment make it a preferred tool for developers aiming to streamline their coding processes and enhance overall workflow.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Research Flowchart

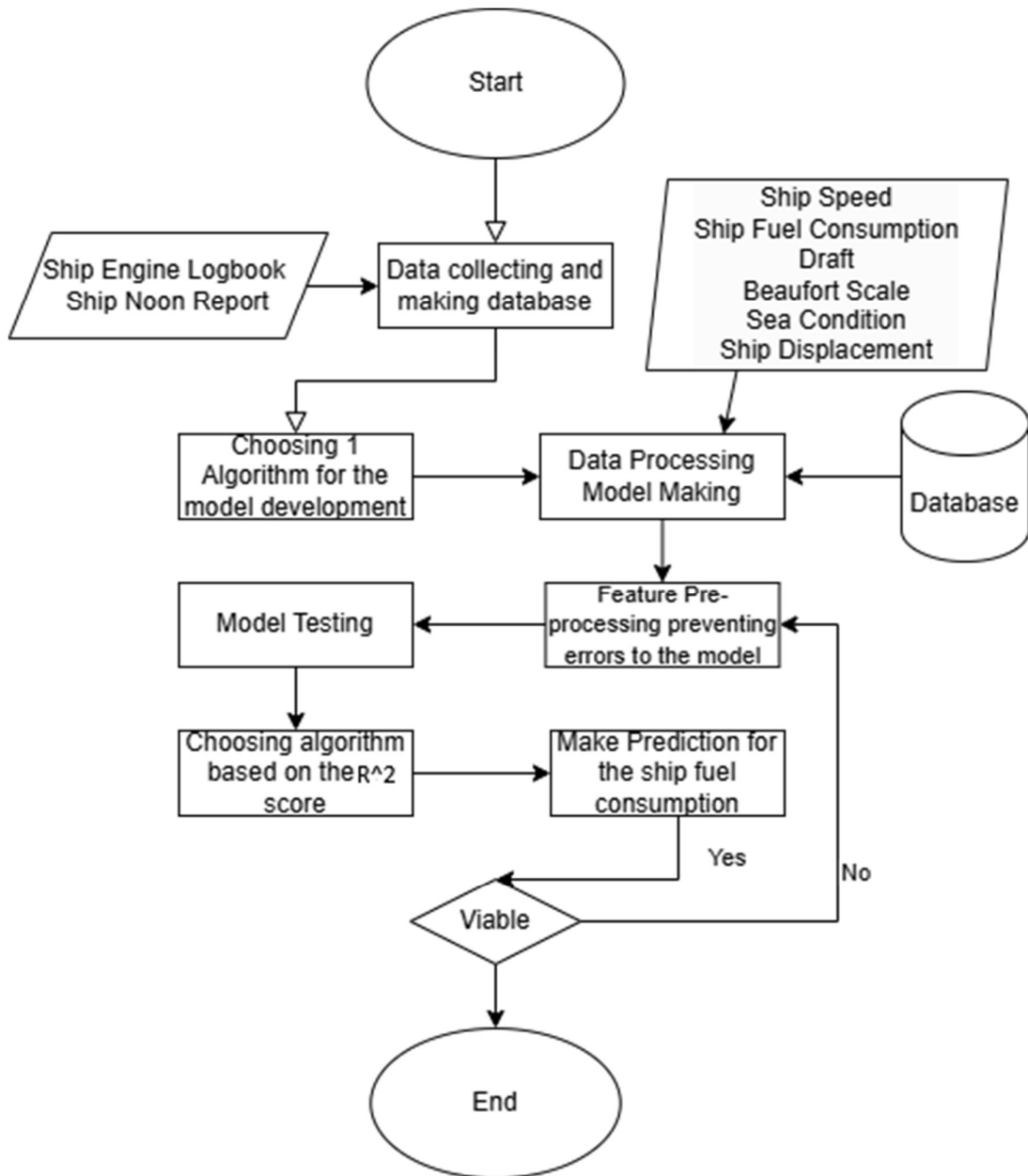


Figure 3.1 Flow Chart

3.2 Data Collecting and Database

Issue recognizable proof is the primary step to start this composing strategy. In this process a list of indicated issues is made additionally make the appropriate premise for identifying the solution of the issue both from presently and within the future. Collecting data and how do the author input it to make the database. Thus, the more the data then the more accurate the data will. Such data gathered are ship logbook, ship operation data, noon report. The data was then made into a database manually.

3.3 Choosing 1 Algorithm for Model Development

When selecting the most suitable algorithm for a given predictive modeling task, options such as XGBoost and Ridge regression often come into consideration. XGBoost, or Extreme Gradient Boosting, is an ensemble learning technique that excels in handling large datasets and complex relationships by combining the predictions of multiple weak learners to produce a robust model. In contrast, Ridge regression is a linear regression technique that incorporates L2 regularization to prevent overfitting, making it particularly effective for datasets with multicollinearity or when the number of features exceeds the number of observations.

This approach allows for a focused evaluation of each algorithm's performance, enabling a clear comparison of their respective R^2 values and other metrics. By systematically testing XGBoost and Ridge regression individually, researchers can identify which algorithm yields better predictive accuracy and aligns more closely with the goals of the analysis. Ultimately, this careful selection process is vital for optimizing model performance and ensuring reliable outcomes in real-world applications.

3.4 Data Processing and Making of the Coding

After the database has been established, it is essential to carefully examine the data before it is inputted into the model. This examination ensures that the data is clean, accurate, and suitable for analysis, which is critical for the model's performance.



Once the data has been validated, it can be inputted into the model, which is built upon a specific algorithm. At this stage, it is important to define the dependent variable (y) and the independent variables (x), as these will form the basis of the model's predictions. On which the code will be made on Visual Studio and processed there.

3.5 Feature Preprocessing preventing errors to the model.

To ensure the robustness and accuracy of predictive models, it is crucial to thoroughly examine the data to prevent issues such as overfitting, which can lead to erroneous results. Overfitting occurs when a model learns the noise in the training data rather than the underlying patterns, making it less effective when applied to new data. To mitigate this risk, the data should be normalized, which involves adjusting the values to a common scale without distorting differences in the ranges of values.

One effective method for normalization is standardization, which transforms the data to have a mean of zero and a standard deviation of one. This can be easily accomplished using the standardization command from the popular Scikit-learn (sklearn) library in Python. By applying this command, practitioners can ensure that their data is appropriately scaled, facilitating better model performance and enhancing the generalizability of the results. This careful preprocessing step is essential for developing reliable models that can accurately predict outcomes in real-world scenarios.

3.6 Model Testing

Model testing is a critical phase in the development of predictive algorithms, as it allows to evaluate the performance of the model and determine its R^2 value. The R^2 , or coefficient of determination, provides insight into how well the model explains the variability of the data, serving as a key metric for assessing its effectiveness. After conducting the initial tests, further research is conducted to identify the optimal model that best fits the data and meets the specific

objectives of the analysis. This process may involve comparing multiple algorithms, fine-tuning parameters, and employing techniques such as cross-validation to ensure that the selected model is not only accurate but also robust against overfitting. By systematically analyzing the results and iterating on model selection, researchers can arrive at a solution that maximizes predictive accuracy and reliability, ultimately leading to more informed decision-making in practical applications. This thorough approach to model testing and optimization is essential for achieving high-quality outcomes.

3.7 Choosing The Best Algorithm

Once the data has been thoroughly processed, the resulting outcomes can be applied and simulated in real-life conditions. This application allows for a practical assessment of the model's effectiveness in predicting fuel consumption under various scenarios. Following this implementation, the results obtained from real-world testing can be compared against the predictions made by the model. This comparison is essential for evaluating the model's performance and reliability. The next step involves selecting the algorithm that demonstrates the highest R^2 value, which indicates the best fit for the data.

Typically, a common approach in this process is to partition the dataset, using 75% for training the model and the remaining 25% for testing its accuracy. This division ensures that the model is trained on a substantial amount of data while still retaining a portion for validation. By following this structured methodology, practitioners can confidently identify the most effective algorithm for estimating fuel consumption, paving the way for improved decision-making and enhanced operational efficiency in real-world applications.

3.8 Make predictions for the ship's fuel consumption.

Once the model has been thoroughly trained and the algorithm with the highest R^2 value has been identified, it can be effectively implemented in real-world applications to estimate fuel consumption. This process involves leveraging the selected model's predictive capabilities, which are rooted in the relationships established between the input variables and fuel consumption outcomes during the training phase.

The choice of variables is crucial, as they must accurately reflect the factors influencing fuel efficiency, such as vehicle type, driving conditions, engine performance, and even external factors like weather and terrain. By integrating this model into operational systems, businesses

and individuals can make informed decisions regarding fuel usage, optimize routes, and ultimately reduce costs and environmental impact. Furthermore, continuous monitoring and refinement of the model may be necessary to ensure its accuracy and relevance as new data becomes available or as driving patterns evolve. This dynamic approach not only enhances the model's reliability but also supports sustainable practices in fuel consumption management.

“This page is intentionally left blank”

CHAPTER 4

RESULT AND DISCUSSION

4.1 Data Acquisition

Data obtained for the paper is manually inserted to Excel to be proceed. Data is obtained from KMP. Kirana VII from PT. Dharma Lautan Utama. The ship is a ro-ro-Passenger ship, with LOA 68,94 m, breadth 17 m, height 4,5 m, draught 3,1 mm, with the service speed 12 knots, max speed 12,7 knots. The data is collected through engine log book and ship deck log book. Data range is 1 year through out 21 February 2023 to 21 February 2024.



Figure 4.1 KM. Kirana VII Ship

Source: (author)

The ship engine uses 2 MITSUBHISI SI2R-T2MPTK with 1400 hp and 1650 RPM. The reading of this data is done by ship crew then these data written manually in ship logbook in a 6 hours interval. The data then manually convert to Microsoft excel file so the phyton library can read the dataset. The ship have is between Surabaya – Lembar. Data is accuired from the ship deck logbook and ship engine logbook. The data used is Sailing Speed (Knots), Beaufort number, Displacement (tons), Sea Condition, Draft (m), and Fuel consumption (L).

As to make the variable “sea condition” be viable to the model so the data is changed to string (numbers). As in 0 is for “docking”, 1 is for “fine waters”, 2 is for “Calm waters”, 3 is for “rippled waters”, 4 is for “smooth waters”, 5 is for “Slight waters”, and 6 is for “Moderate sea”.

4.2 Choosing Algorithm

Choosing the algorithm is an entirely different part, so author makes 2 different model/file. One is Ridge as can be seen like in figure 4.2

```
# Extract features and target variable
X = avg_data[['Speed', 'Wind', 'Displ', 'Sea', 'Draftm']]
y = data['F0']

# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X)
X_test_scaled = scaler.transform(X)

# Define the hyperparameters grid for Ridge Regression
rr_params = {
    'alpha': [0.1, 1.0, 100.0, 10.0, 1000.0]
}

# Initialize variables to track the best random state
best_random_state = None
best_r2 = -np.inf

# Try different random states
for random_state in range(250,500):
    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X_train_scaled, y, test_size=0.25, random_state=random_state)

    # Perform grid search for Ridge Regression
    rr = Ridge()
    grid_search = GridSearchCV(rr, rr_params, cv=5)
    grid_search.fit(X_train, y_train)
```

Figure 4.2 Ridge Regression algorithm code

The data that is accommodate with “X” is the overall data of the algorithm which is speed, beaufort scale, displacement, sea condition, and draft. The data is accommodate with “Y” is the ship fuel consumption. The data is scaled as to process the data then Ridge regression used is the alpha state, the machine learning model will test every alpha and will choose the best alpha with resulting high R-square score. The random states are random seeds that the data takes from the sample dataset. On this the random seed is in the range of 250 – 500. The grid search is also used for the systematic method used in machine learning and statistics to find the optimal parameters for a given model or function. Using 5 repeater then the data is converted to 25% tested, 75% trained.

XGBoost algorithm code is different than the ridge regression, it can be seen in figure 4.2 below;

```
# Extract features and target variable
X = avg_data[['Speed', 'Wind', 'Displ', 'Sea', 'Draftm']]
y = data['FO']

# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X)
X_test_scaled = scaler.transform(X)

# Define the hyperparameters grid for XGBoost
xgb_params = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.1, 0.5, 1.0],
    'n_estimators': [50, 100, 200]
}

# Initialize variables to track the best random state
best_random_state = None
best_r2 = -np.inf

# Try different random states
for random_state in range(90,100):
    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X_train_scaled, y, test_size=0.25, random_state=random_state)

    # Perform grid search for XGBoost
    xgb = XGBRegressor()
    grid_search = GridSearchCV(xgb, xgb_params, cv=3)
    grid_search.fit(X_train, y_train)
```

Figure 4.3 XGBoost Regression code

The code from the x, y until the scalling is the same as ridge regressor, the difference is the parameter that makes the XGBoost regressor and the algorithm differ than ridge regressor. The max_depth paramater can be choosen from 3, 5, and 7 as the model work itself to resulting the highest R-square score. Learning rate is the process of the model to understand how well the complicated relationship of each variable to the output. N_estimators is the how much trees is made to estimate the best result for the model.

4.3 Preprocessing

The process of model training depending on the quality of the data that is inserted. Hence, if the model is trained on the wrong data, it will try to reduce the error in that dataset and the model. There are no data for draft from 26/02/23 to 13/03/23, so that time period is filtered out and was replaced to NaN (not a number), and not to be counted into the data set. the data that will included in the model consist of: beufourt number, speed, displacement, draft ,and sea condition as the input and fuel consumption as the output.

	Current Time Date	Sailing speed	Beaufort	Displacement	Sea Condition	\
0	2023-02-20	11.0	1	1873.536	2	
1	2023-02-21	11.6	1	1873.536	2	
2	2023-02-22	10.7	1	1990.632	3	
3	2023-02-23	11.7	1	2049.180	3	
4	2023-02-24	10.7	1	1932.084	3	
..	
360	2024-02-17	11.0	1	1873.536	2	
361	2024-02-18	11.0	1	1990.632	2	
362	2024-02-19	11.2	1	1814.988	2	
363	2024-02-20	10.5	1	1990.632	2	
364	2024-02-21	12.0	0	1639.344	5	
	Draft	Fuel consumption rate				
0	3.2	7665				
1	3.2	7700				
2	3.4	7784				
3	3.5	7456				
4	3.3	7774				
..				
360	3.2	7621				
361	3.4	7543				
362	3.1	7691				
363	3.4	7613				
364	2.8	7695				

[365 rows x 7 columns]

Figure 4.4 List of data gathered

Then the data is scaled to different parts which, can easily be identified the range of values. The variables such as Sailing Speed, Beaufort number, Displacement, Sea condition, and Draft. The data were described as Figure 4.4.

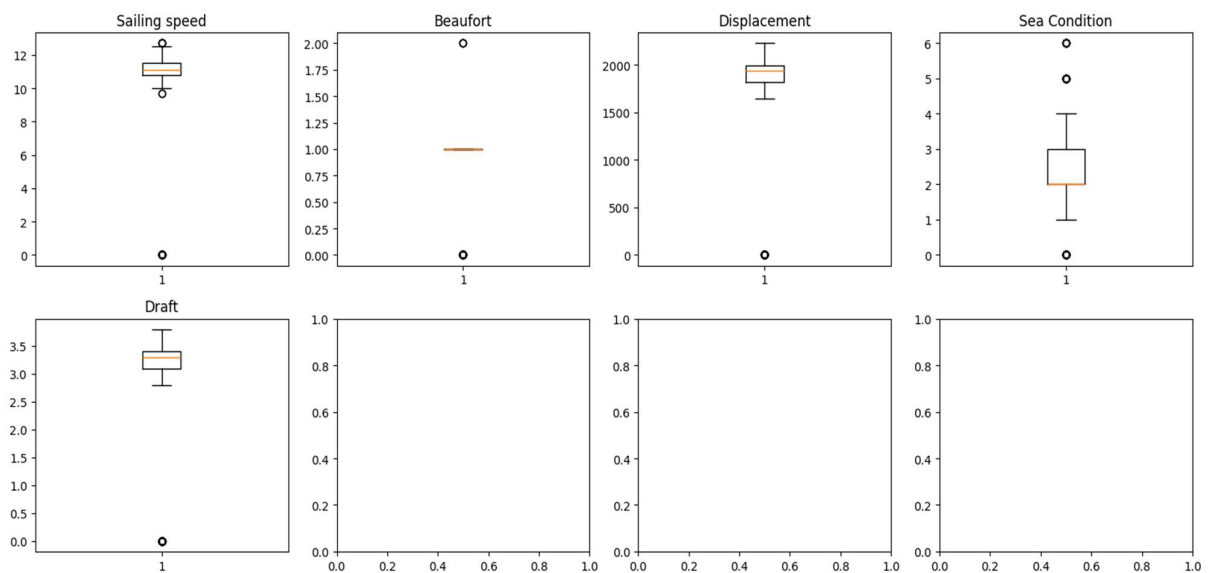


Figure 4.5 X variables scales

The information is composed to the log book by the team who physically recorded the estimation each 4 hours, this suggests a expansive vulnerability of each hours of utilization. As

this prepare in not robotized, the comes about of the perusing it can contrast very noteworthy in correct time of the day. Since of this on the off chance that the dispatch halt some time recently the fourth hour, at that point the estimation esteem still composed within the fourth hour. Which is the data is filtered due to human mistakes, etc.

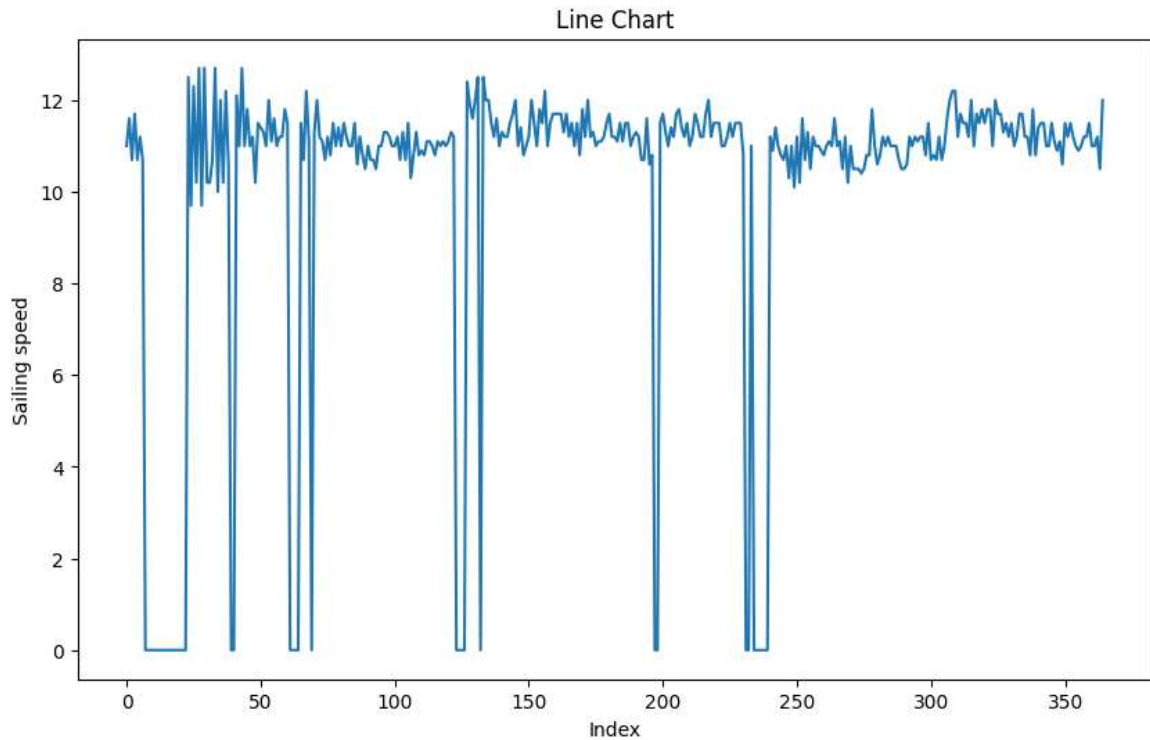


Figure 4.6 Pre-processed Sailing Speed

The data here is the chart of sailing speed which is in knots. With the highest is 12,7 knots and the lowest is 0 knots. The data here is not processed yet, this is due to there are 0/null value. The data here have variative values so the algorithm can estimate and correlate the variable with the output.

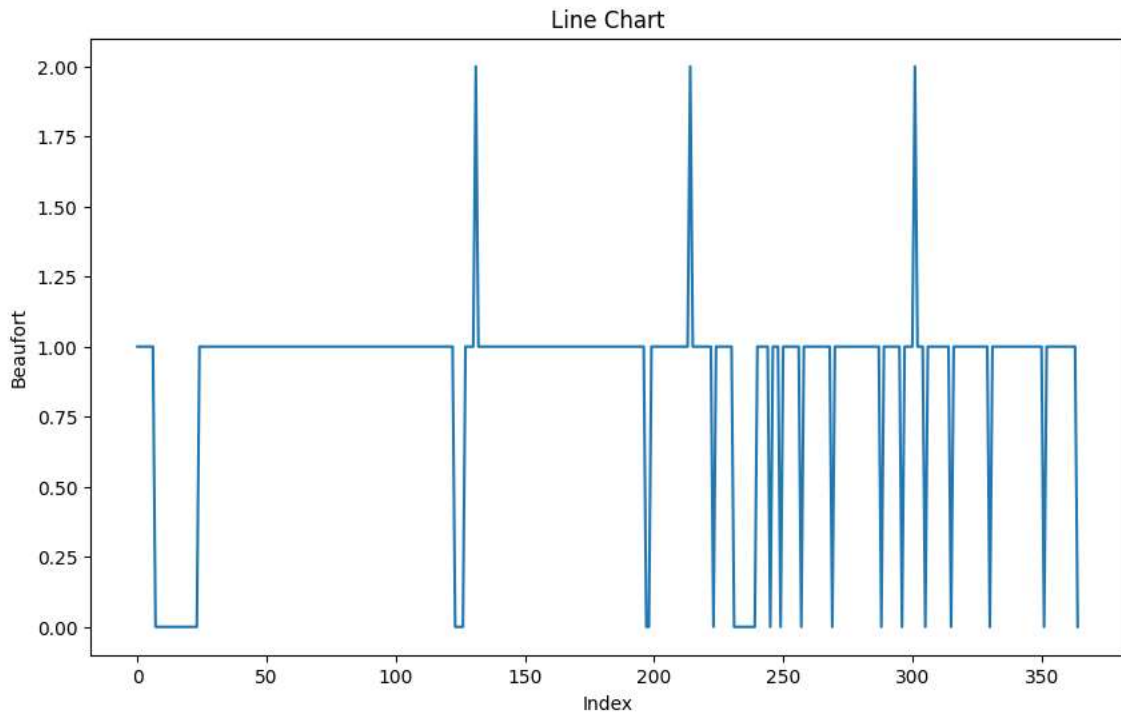


Figure 4.7 Pre-processed Beaufort scale

The data here is the chart of sailing speed which is in knots. With the highest is 2 beaufort scale and the lowest is 0 beaufort scale. The data here is not processed yet, this is due to there are 0/null value. The data here have variative values so the algorithm can estimate and correlate the variable with the output.

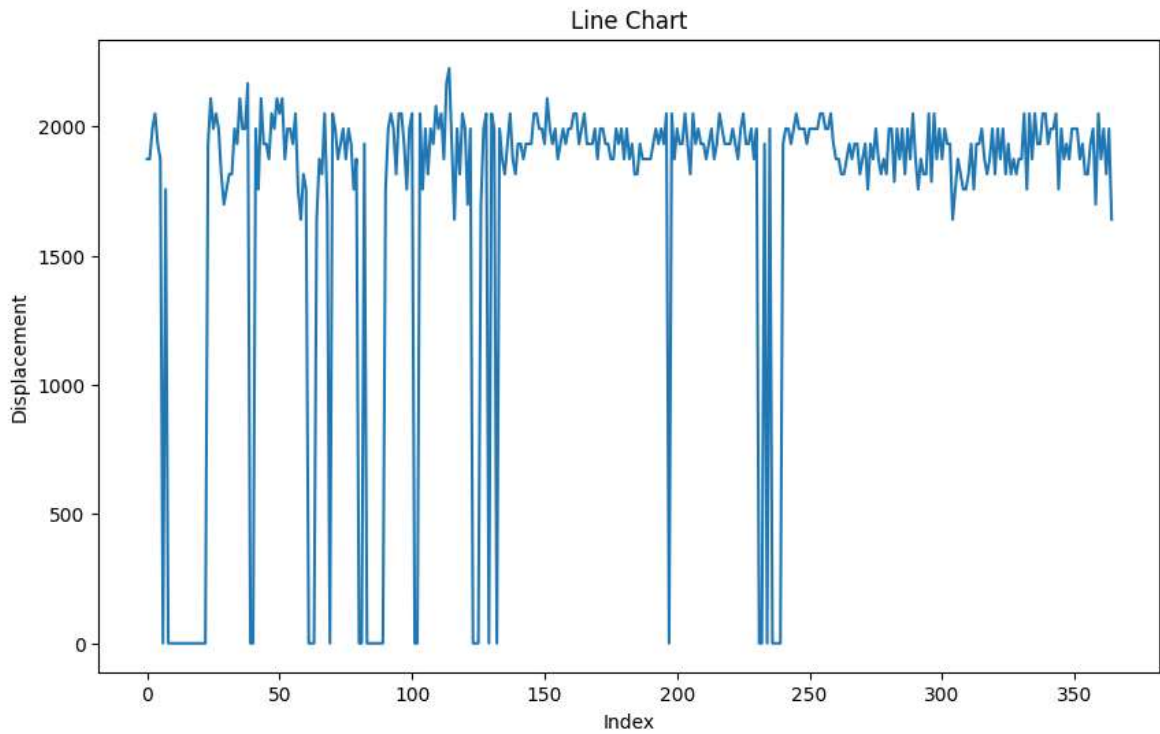


Figure 4.8 Pre-processed Displacement

The data here is the chart of sailing speed which is in knots. With the highest around 2200 tons and the lowest is 0 tons. The data here is not processed yet, this is due to there are 0/null value. The data here have variative values so the algorithm can estimate and correlate the variable with the output.

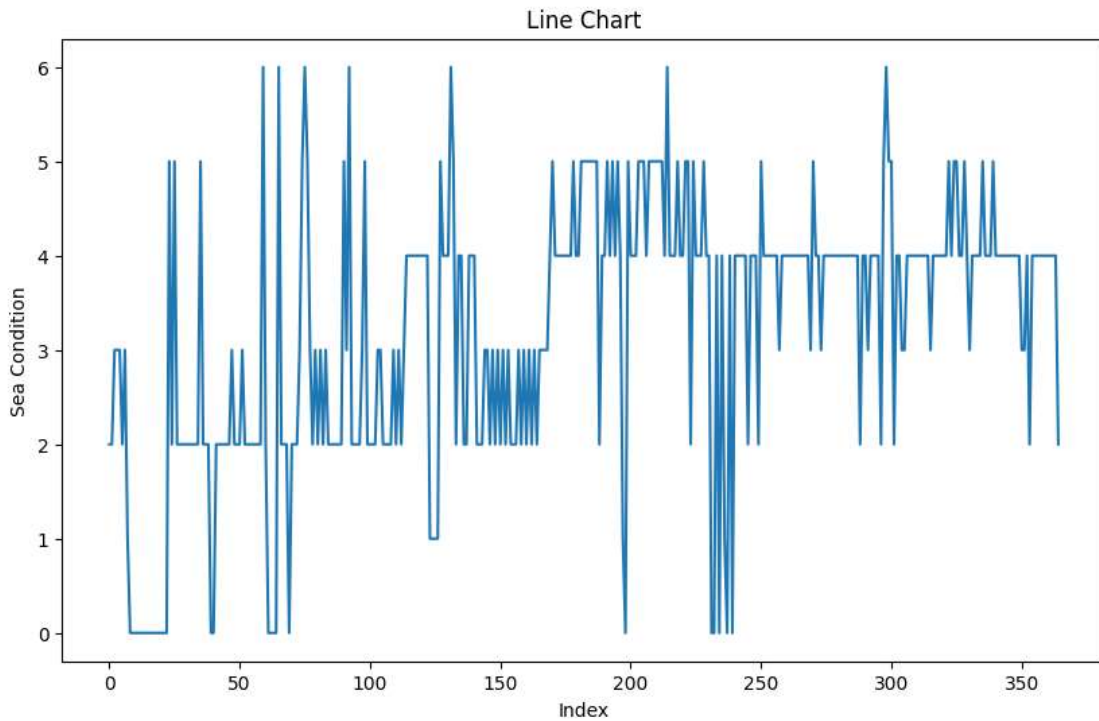


Figure 4.9 Pre-processed Sea Condition

The data here is the chart of sailing speed which is in knots. With the highest is scale 6 which is (moderate sea) and the lowest is scale 0 (docking). The data here is not processed yet, this is due to there are 0/null value. The data here have variative values so the algorithm can estimate and correlate the variable with the output.

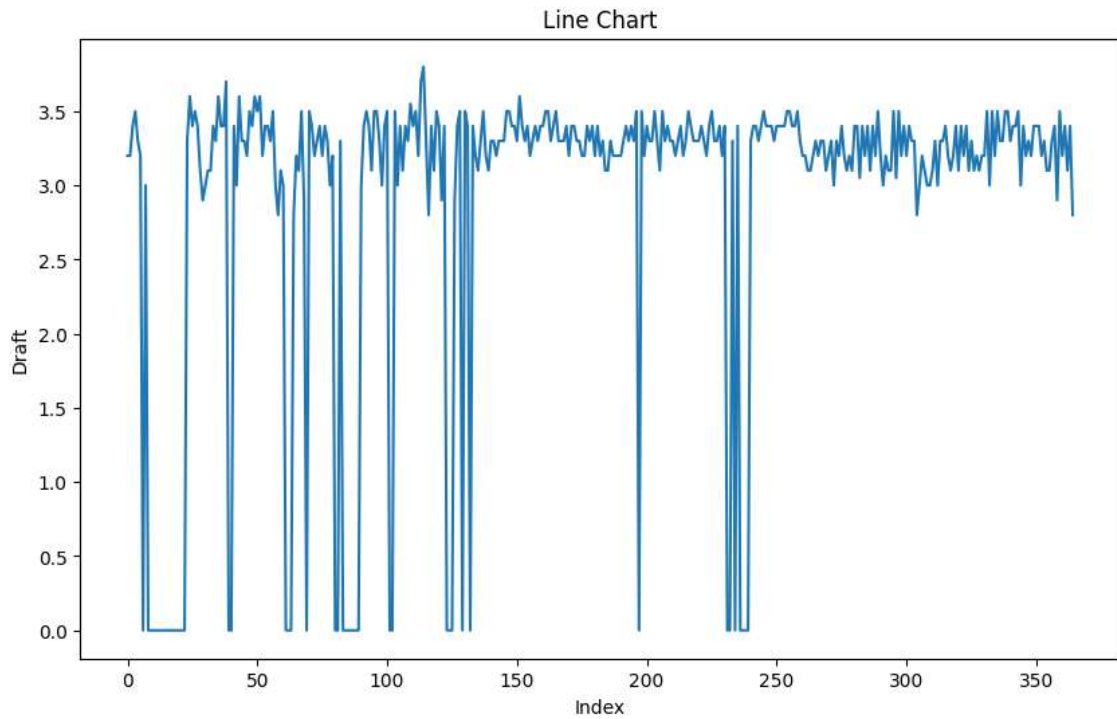


Figure 4.10 Pre-processed Draft

The data here is the chart of sailing speed which is in knots. With the highest is 3,7 m and the lowest is 0 m. The data here is not processed yet, this is due to there are 0/null value. The data here have variative values so the algorithm can estimate and correlate the variable with the output.

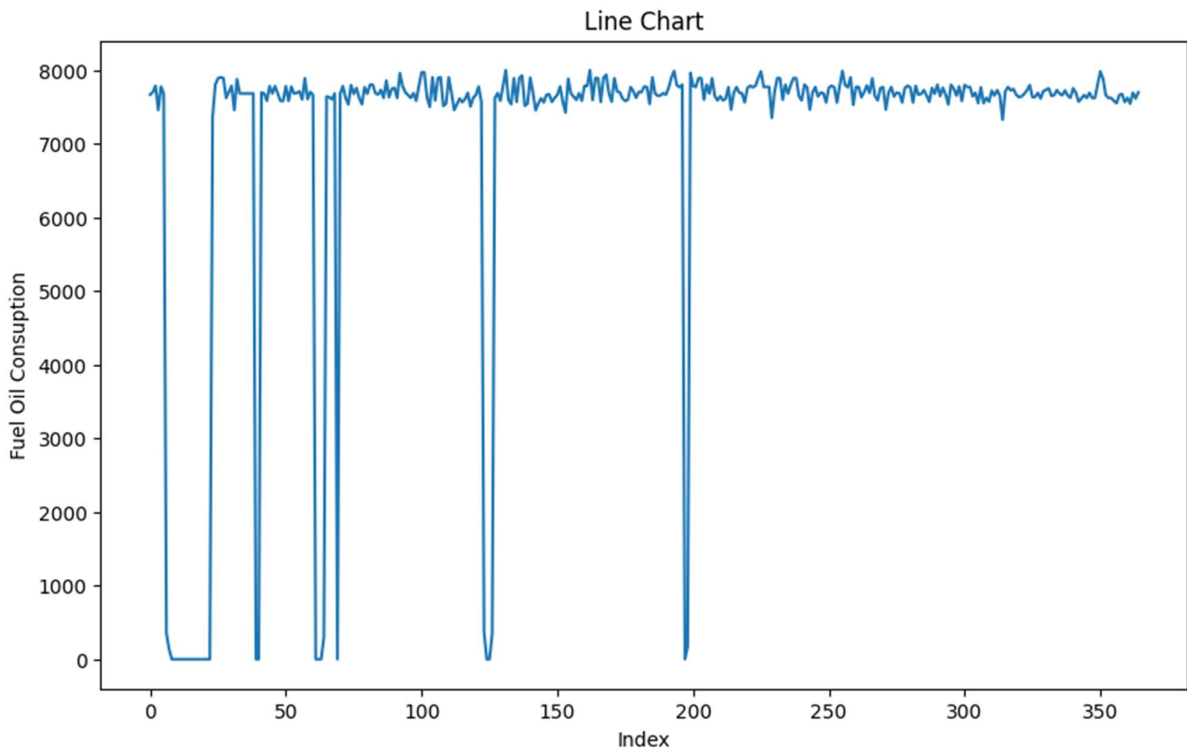


Figure 4.11 Pre-processed Graph Fuel Consumption

The data is then putted to the graph and then it can be seen that there are NaN or “null” values that could mess up the model, so by then the data need to be processed/cleaned to prevent this kind of error. By adding “ignore null” then the data can be processed best.

```
new_data = data.dropna(axis= 0)
new_data.isnull().sum()
✓ 0.0s
Current Time Date      0
Sailing speed          0
Beaufort               0
Displacement           0
Sea Condition          0
Draft                 0
Fuel consumption rate  0
dtype: int64
```

Figure 4.12 Eliminating 0 values

The reason to drop the null or “0” datas is to ensure the model prediction, so for the algorithm know relationship on every part of the X variable. There are also the Y variable dropped to ensure the model to be fitted for testing.

4.4 Processed Dataset

In this context, the dataset is processed to eliminate null values, resulting in much more reliable and accurate values for prediction, as part of the data cleaning process. The significance of processed data lies in its ability to support decision-making processes, validate hypotheses, and provide actionable insights for stakeholders. The benefits of processed data include enhanced decision-making by providing clear insights, improved data management through structured formats, increased efficiency by streamlining analysis, and greater accessibility of insights, which fosters collaboration and informed decision-making across an organization.

Data that already cleaned then could be use for the model of the machine learning training as on the figure 4.13

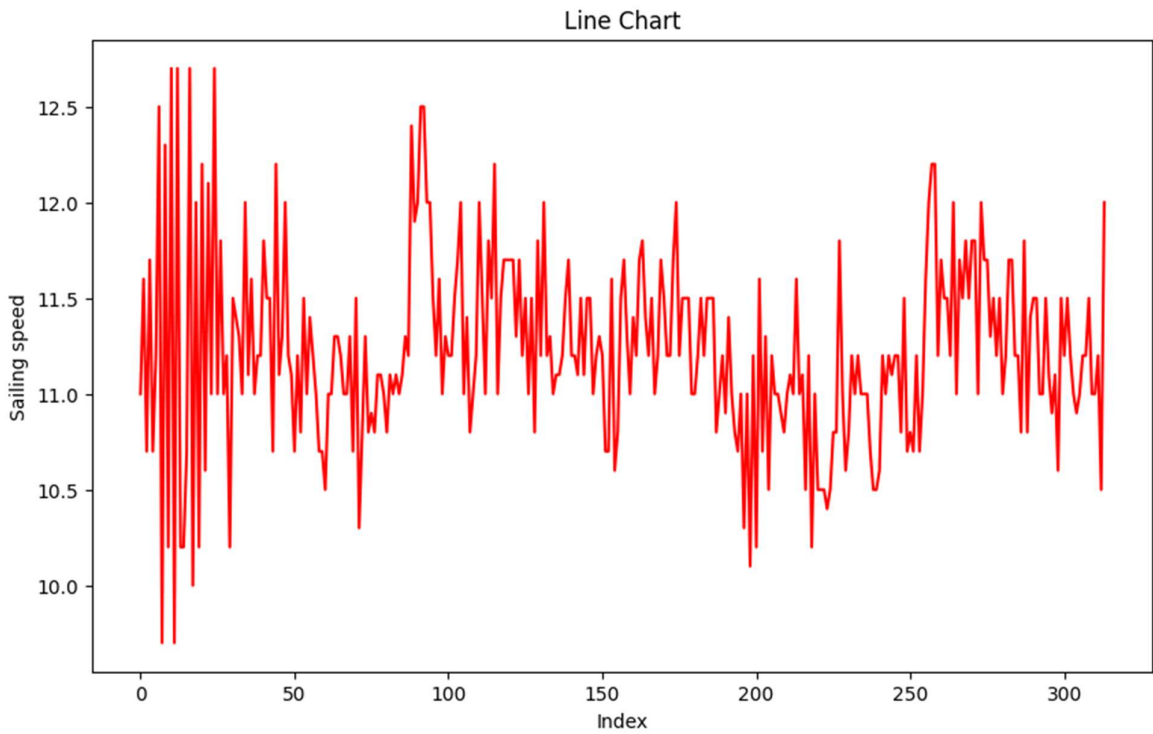


Figure 4.13 Cleaned Sailing speed data

The data line here is the sailing speed data which the null values will be outlined using drop 'null' value, this is to ensure the dataset used to be much more clean and help the model to interpret the correlation between each variable with the output. The line its much more better due to the elimination and the data is variative which allows the model to better perform than if the data is constant.

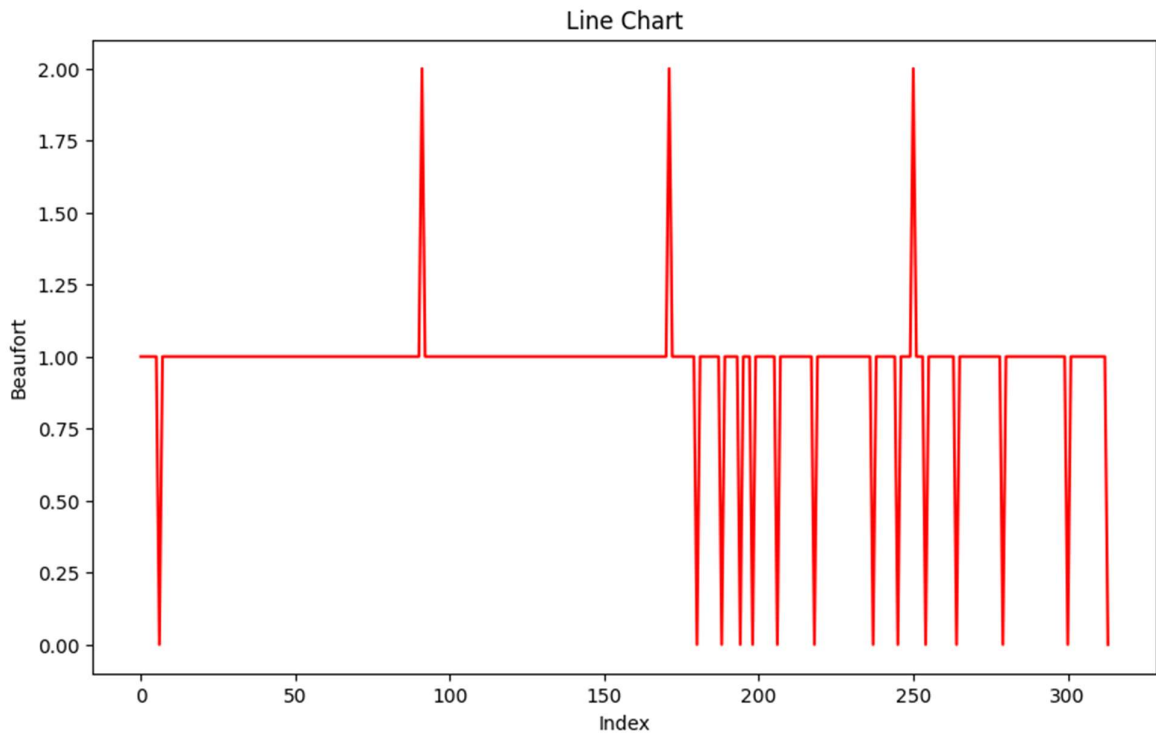


Figure 4.14 Cleaned Beaufort scale data

The data line here is the Beaufort scale data, and the null values will be delineated using the drop 'null' value, ensuring that the dataset utilized is much cleaner and allowing the model to interpret the association between each variable and the result. The line is significantly better as a result of the removal, and the data is variable, allowing the model to perform better than if the data was constant.

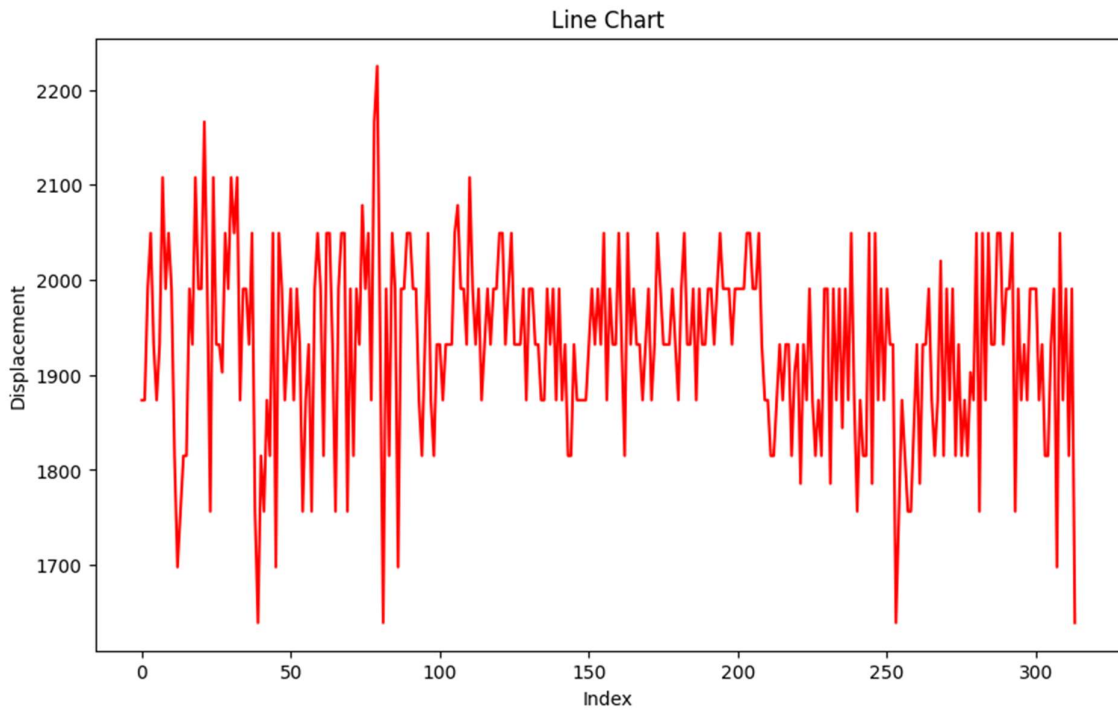


Figure 4.15 Cleaned displacement data

The ship displacement data is crucial for understanding the performance and stability of maritime vessels. To ensure the integrity of our analysis, we will eliminate any null values from the dataset. This step is vital, as it not only cleans the data but also enhances the model's ability to accurately interpret the relationships between each variable and the outcome of interest. By removing these null entries, we create a more robust dataset that reflects the true variability of the data, which is essential for effective modeling. Cleaner dataset allows the model to focus on meaningful patterns and associations, thereby improving predictive accuracy. The variability in the remaining data points contributes to a more nuanced understanding of how different factors interact, leading to better insights and more reliable predictions.

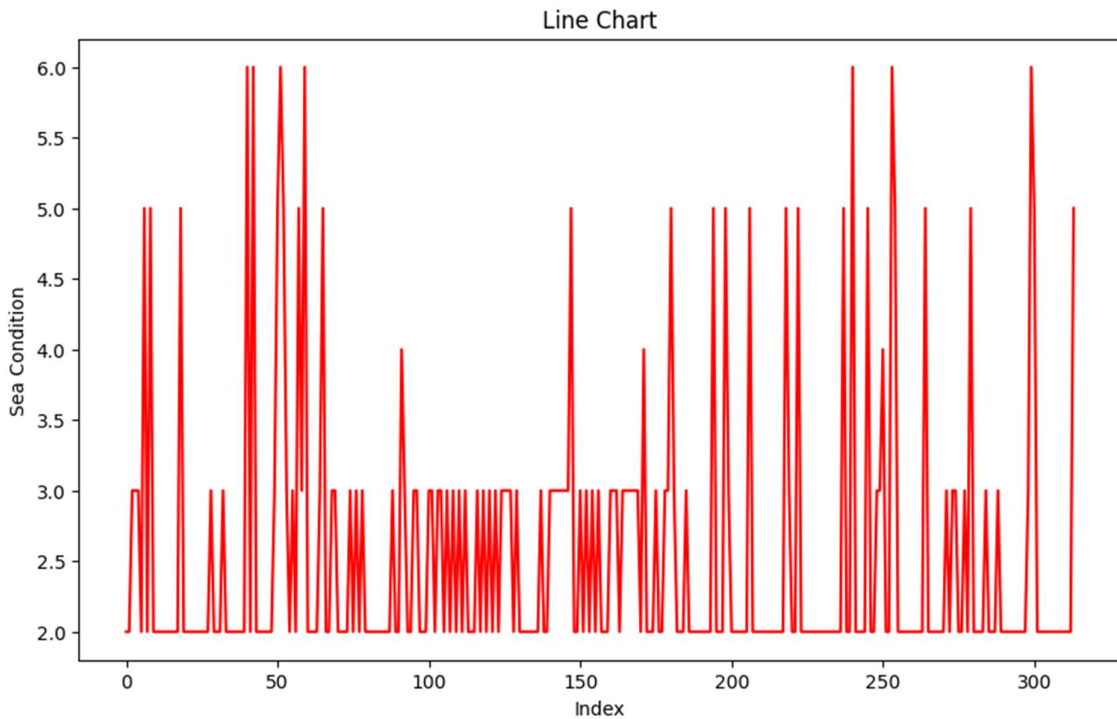


Figure 4.16 Cleaned sea condition data

The sea condition data is crucial for understanding the performance and stability of maritime vessels. To ensure the integrity of our analysis, we will eliminate any null values from the dataset. This step is vital, as it not only cleans the data but also enhances the model's ability to accurately interpret the relationships between each variable and the outcome of interest. By removing these null entries, we create a more robust dataset that reflects the true variability of the data, which is essential for effective modeling. A cleaner dataset allows the model to focus on meaningful patterns and associations, thereby improving predictive accuracy. The variability in the remaining data points contributes to a more nuanced understanding of how different factors interact, leading to better insights and more reliable predictions.

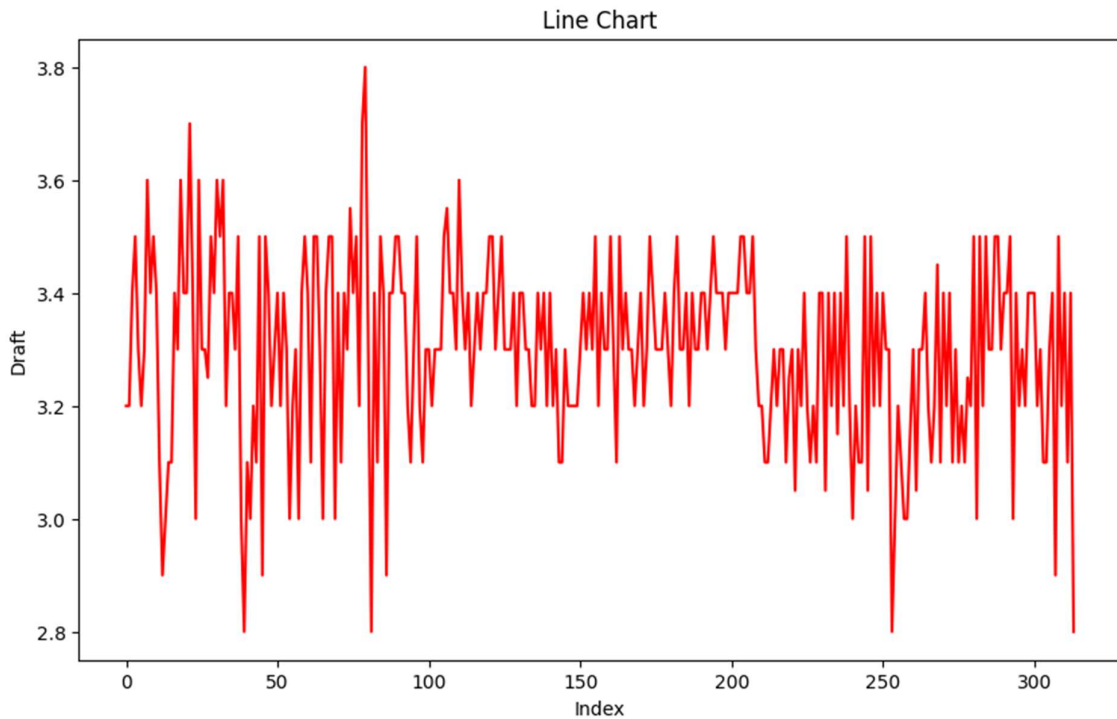


Figure 4.17 Cleaned draft data

The ship draft data is critical for understanding the performance and stability of marine ships. To protect the integrity of our study, we shall remove any null values from the dataset. This step is critical because it not only cleans the data but also improves the model's capacity to correctly comprehend the links between each variable and the result of interest. By deleting these blank entries, we build a more robust dataset that accurately reflects the data's variability, which is required for effective modeling. A cleaner dataset enables the model to focus on significant patterns and correlations, resulting in higher predicted accuracy. The remaining data points' unpredictability adds to a more detailed understanding of how multiple aspects interact, resulting in superior insights.

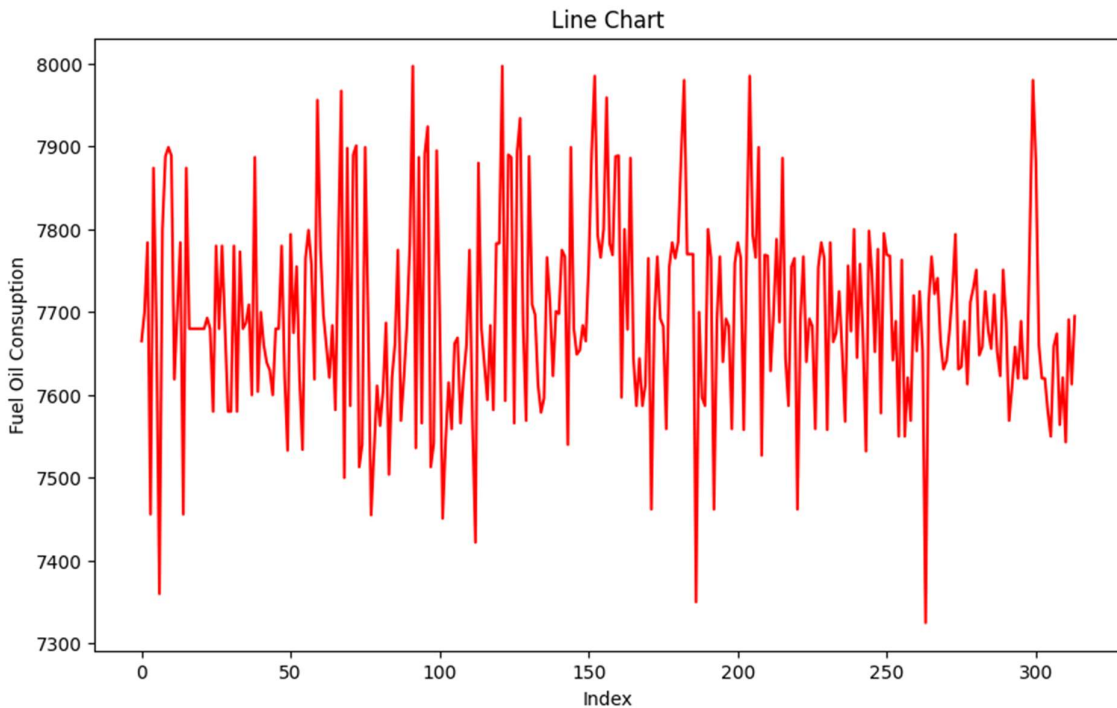


Figure 4.18 Cleaned Fuel consumption data

Analyzing fuel consumption data is essential for evaluating the efficiency and operational performance of maritime vessels. To maintain the integrity of our analysis, we will remove any null values from the dataset. This crucial step not only cleans the data but also enhances the model's capability to accurately discern the relationships between various variables and the target outcomes. By eliminating these null entries, we create a more comprehensive dataset that captures the true variability of the data, which is vital for effective modeling. Furthermore, a refined dataset allows the model to concentrate on significant patterns and correlations, leading to improved predictive accuracy. The diversity in the remaining data points fosters a deeper understanding of how different factors interact, resulting in more insightful analyses and dependable predictions.

4.5 Data Input

Data from the gathering will be processed and then compiled into a dataset. Example of the dataset can be seen at Tabel 4.1

Table 4.1 Data Sample

Current Time Date	Sailing speed	Beaufort	Displacement	Sea Condition	Draft	Fuel consumption rate
-------------------	---------------	----------	--------------	---------------	-------	-----------------------

20 02 2023	11,0000	1	1873,5360	2,0000	3,2	7665,0000
21 02 2023	11,6000	1	1873,5360	2,0000	3,2	7700,0000
22 02 2023	10,7000	1	1990,6320	3,0000	3,4	7784,0000
23 02 2023	11,7000	1	2078,4540	3,0000	3,55	7456,0000

From the table there are parameters that will be used for the machine learning algorithm. The data is compiled into 1 day, so the data is every 1 day. Every data is inputted manually so it requires precision in inputting the data.

	Current Time Date	Sailing speed	Beaufort	Displacement	Sea Condition	Draft	Fuel consumption rate
count		335	335.000000	335.000000	335.000000	335.000000	335.000000
mean	2023-09-02 08:40:07.164179200	10.918806	0.943284	1815.599696	2.486567	3.101045	7697.322388
min	2023-02-20 00:00:00	0.000000	0.000000	0.000000	0.000000	0.000000	7325.000000
25%	2023-06-07 12:00:00	11.000000	1.000000	1873.536000	2.000000	3.200000	7620.000000
50%	2023-09-03 00:00:00	11.200000	1.000000	1932.084000	2.000000	3.300000	7684.000000
75%	2023-11-29 12:00:00	11.500000	1.000000	1990.632000	3.000000	3.400000	7771.500000
max	2024-02-21 00:00:00	12.700000	2.000000	2224.824000	6.000000	3.800000	7997.000000
std		NaN	1.881243	0.267626	467.734015	0.996535	119.259732

Figure 4.19 Values of the data inputted

The data look like the figure 4.8 the data of 'fuel consumption' is already deleted then the data that has null values need to be deleted more as the rows are removed. After that the data will be processed to the model to be trained.

4.6 Model Training

The Model training for the algorithm is evaluated by R-squared value and RMSE value, to check the accuracy of the regression model fits on the datasets. R-squared value if closer to 1 then the model would fit perfectly, the RMSE value if differ the predicted value the closer to the actual value then the model turns out to be perfect.

Data used to construct the regression model which comes from the training and testing datasets. Usually the data train is 75% and the testing data is 25%. The dataset is chosen randomly based on the random seed taken. After the model is trained then the testing used to evaluate the model created by the train dataset.

```

for random_state in range(250,500):
    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X_train_scaled, y, test_size=0.25, random_state=random_state)

    # Perform grid search for Ridge Regression
    rr = Ridge()
    grid_search = GridSearchCV(rr, rr_params, cv=5)
    grid_search.fit(X_train, y_train)

    # Predict and evaluate the model
    y_pred = grid_search.best_estimator_.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mse) # Calculate RMSE

```

Figure 4.20 Train data and testing data splitting

The dataset then inputted to the coding by using Python Language, Microsoft visual studio to make the model. After the model is done and ready to launch. The model output is the R^2 value which is used to show the accuracy of the regression model. The mean squared error tells the standard value of the model made.

Which the inputted data consist of ship speed, ship displacement, ship draft, beaufort scale, and sea condition. The output for the model is the fuel consumption. The model operate the data and training, some of the data were used for testing. After the testing is done then the model will estimate the R^2 score.

The data set will be used for 2 regressors, using the Ridge Regression and XGBoost Regressor. After both regressor were used then it will resulting a score, then the score will be used to determine the model accuracy. If preferable then the highest model will be chosen as the 'best' for the KMP. Kirana VII.

4.7 Ridge Regression

This model using the 5 inputs (variables) and 1 output the Ridge regression results were around 88,87% accuracy for the R^2 , with the random seed of 288.

```

print(f'Best hyperparameters for Ridge Regression: {best_model.best_params_}')
print(f'Best R-squared: {best_r2}')

```

[28] ✓ 16.4s

```

... Best random state: 288
Best hyperparameters for Ridge Regression: {'alpha': 10.0}
Best R-squared: 0.8866067516833636

```

Figure 4.21 Ridge Regression Value

The results is quite good for the model to determine the actual value. The ridge regression model could be used to determine the actual values used for the ship.

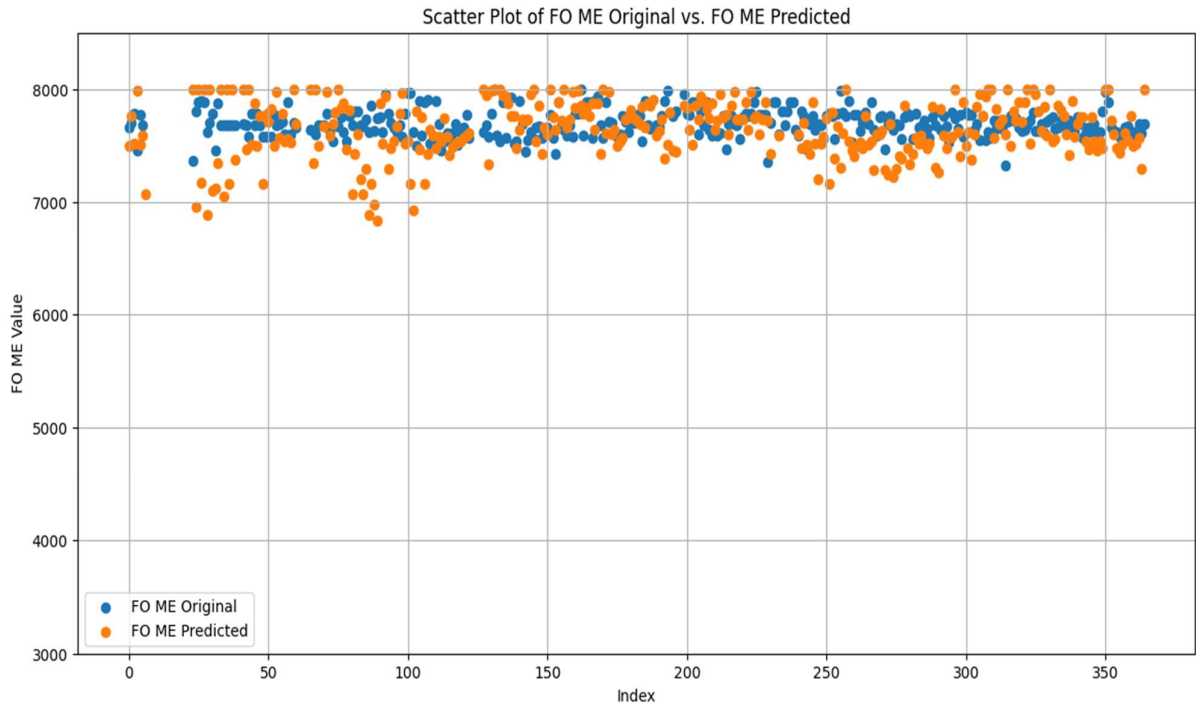


Figure 4.22 Scatter plot of the Actual vs Predicted values for ridge regression

The model graph used can be seen the actual vs predicted is quite close, but some predicted numbers from the quartal of the index is not quite accurate, but after that the model could more or less predict near the actual value. Due to this the model does quite need more calibration by adding more datas, more variables, and adding parameters for the model, so that the model could get more connection between each data and interpret the prediction data better and closer to the actual value.

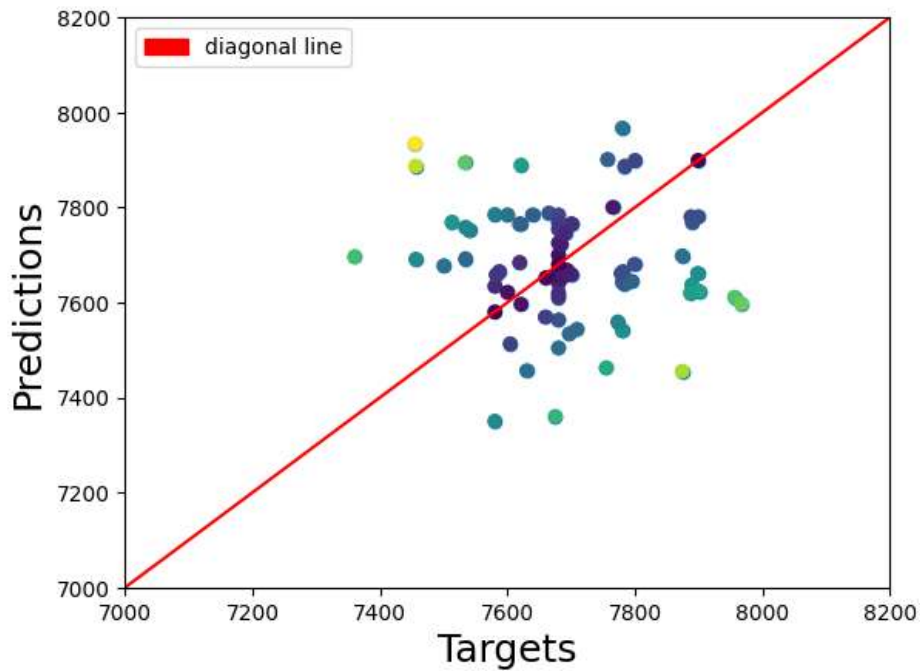


Figure 4.23 Ridge tested vs prediction

It also can be seen that the prediction lowest having around 6800 and the highest up to 8000 liters for the prediction. The speed relation with fuel consumption is exponential its shown from the figure 4.18

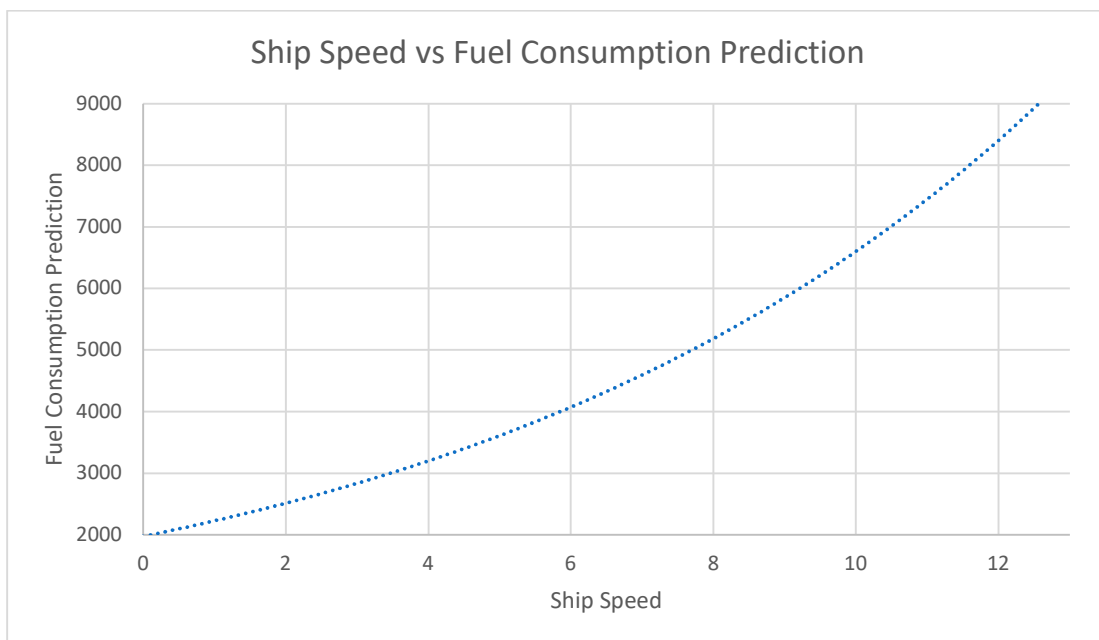


Figure 4.24 Ship Speed vs Fuel Consumption Prediction

The formula relation between Speed Fuel Consumption and Ship Speed were determined as;

$$BHP = K1 * K2 * VS^3$$

The connection between the formula and the graph is that the BHP correlated to fuel consumption, thus the ship fuel consumption relation between ship speed is the ship speed with the power of 3. As seen from the graph the result were correlated to the formula.

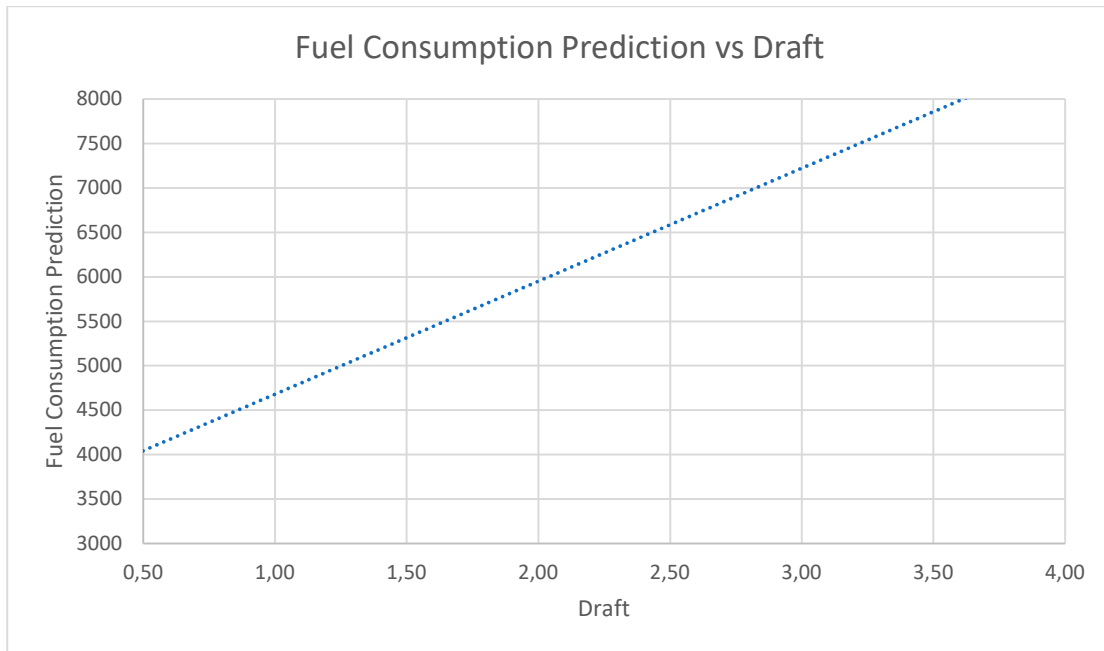


Figure 4.25 Draft vs Fuel Consumption Prediction

The relation of figure 4.13 is ship middle draft and ship fuel consumption prediction. The graph is linear, original formula between draft and fuel consumption are;

$$BHP = k1 * Ct * \rho * wsa * VS^2$$

$$wsa = T * surface\ area$$

Formula of fuel consumption estimation and draft can be seen that T variable is used for calculating wetted surface area (wsa). Thus the relation of the graph is linear. The graph correlation is align with the formula used.

From this the ridge method can be used for the prediction, 88,82% for the R^2 value, with 5 variables; ship speed, displacement, ship draft, beaufort number, and sea condition. Relation between the variable is displayed on the figure 4.7 and 4.8 were related to the original formula. Using regression then the equation of the model is determined as

$$\text{Fuel Consumption} = 7046,73 + 1592.483(\text{Ship Speed in Knots}) + 2,87 (\text{Beaufort Number}) + 54,53 (\text{Displacement}) + 154,39 (\text{Sea Condition}) + 54,53 (\text{Draft})$$

The output of the regression model reflects the relationship between the dependent variable (y) and the independent variables (x) that influence it. In this case, ship speed has a significant impact on the dependent variable, more so than the other variables in the model. Essentially, the model quantifies the influence of each independent variable on the regression outcome, deriving a formula that encapsulates these relationships.

It's important to note that the regression model's calculations are complex and cannot be easily replicated through manual calculations. This complexity arises from the various fitting techniques and algorithms employed by the model to establish correlations between the independent variables and the output. These techniques allow the model to capture intricate patterns and interactions within the data, leading to more accurate predictions. Regression model provides valuable insights into how different factors, particularly ship speed, affect the outcome variable. However, the sophisticated nature of the model means that its outputs are best interpreted within the context of the model itself, rather than through straightforward manual calculations..

Table 4.2 Ridge Regression Sample

Target	Prediction	Residual	Difference (%)
7665	7493,47	171,53	2,24
7700	7766,35	-66,35	-0,86
7784	7519,43	264,57	3,40
7456	7987,61	-531,61	-7,13
7774	7506,04	267,96	3,45
7680	7584,43	95,57	1,24

The analysis of the model's performance using Ridge Regression reveals a nuanced picture when comparing predicted values to target values. In this study, the dataset was divided into two segments: 75% was utilized for training the model, while the remaining 25% served as a testing set to evaluate its predictive capabilities. The Ridge Regression model demonstrated its effectiveness by achieving its closest approximation for a target value of 7700 liters, predicting 7776.35 liters. This resulted in a remarkably small difference of just 0.86%, indicating that the model can perform well under certain conditions. However, the results also showcased considerable discrepancies in other predictions. For instance, when the target was set at 7456 liters, the model predicted 7987.61 liters, leading to a significant difference of 7,13%. Such variability in performance suggests that while the model has the potential for accuracy, it also faces challenges in consistently aligning its predictions with the targets across the dataset.

In light of these findings, the model was subsequently tested on the complete dataset to further evaluate its overall effectiveness and reliability. This comprehensive evaluation underscores the importance of not only assessing the model's performance on a subset of data but also understanding how it behaves with the full range of inputs. The significant differences observed between some predicted and target values highlight the necessity for further refinement of the model. This could involve exploring additional features that may enhance the model's predictive power, optimizing the underlying algorithms to improve performance, or employing more sophisticated techniques, such as feature engineering or hyperparameter tuning, to enhance predictive accuracy. Continuous evaluation and iterative enhancements are essential to ensure that the model can consistently meet desired performance standards across

various scenarios. Ultimately, a comprehensive approach to model development will be crucial in harnessing its full potential and delivering reliable predictions.

In the context of predicting fuel consumption, the evaluation of model performance is essential to understand how accurately a model can estimate values compared to actual targets. Each prediction generated by the Ridge Regression model is based on its underlying estimation algorithms, which analyze various input features to forecast fuel consumption. To assess the effectiveness of these predictions, we compare the predicted values against the targeted values of fuel consumption. This comparison yields residual values, which represent the differences between the predicted and actual values. These residuals are crucial for understanding the model's accuracy, as they highlight areas where the model may underperform or excel.

To quantify the model's predictive capability, we can express these residuals as percentages. This percentage calculation provides a clear metric to evaluate the model's performance across different scenarios, allowing us to identify how closely the predictions align with the actual fuel consumption values. By analyzing these percentages, we can gain insights into the model's strengths and weaknesses, enabling us to make informed adjustments to improve accuracy. The use of Ridge Regression, with its ability to handle multicollinearity and regularize coefficients, has emerged as a powerful tool for predicting fuel consumption. Its effectiveness in managing complex datasets and capturing intricate relationships between features makes it particularly suitable for this domain. By leveraging advanced algorithms and optimization techniques, Ridge Regression consistently delivers reliable predictions, making it a preferred choice for practitioners in the field.

4.8 XGBoost

This model using 5 inputs and 1 output, but XGBoost use grid parameter and every grid is consist of different variables such as `learning_rate`, `max_depth`, and `n_estimators`. So XGB model search for the grid and estimates the best parameters for the best model. After the best model is decided then the model is trained to estimate the value of R^2 . The results of the model were around 93,06%. The value of the R^2 is realistic, so the model can be used for estimating the fuel consumption estimation. The model could be optimized by adding more data to the model to optimize it further.

```
print(f'Feature Importances: {feature_importances}')  
[11] ✓ 54.6s  
... Best random state: 92  
Best R-squared: 0.9306723368515235  
Best hyperparameters for XGBoost: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}  
Feature Importances: [0.01014676 0.00510329 0.00515263 0.97959733 0.      ]
```

Figure 4.26 Result XGBoost

The resulting R-squared value of 93.06% indicates a strong fit of the regression model to the dataset utilized for the analysis. This high level of R-squared suggests that the model explains a significant portion of the variance in the data, demonstrating its effectiveness in capturing the underlying relationships. In Figure 4.27, we can observe the graphical representation of the XGBoost model, which compares the actual values against the predicted values. This visual comparison provides valuable insights into the model's performance, highlighting areas where predictions align closely with actual observations as well as instances of deviation. The close proximity of the predicted values to the actual values in the graph reinforces the reliability of the XGBoost model in forecasting outcomes based on the given dataset. Overall, the combination of a high R-squared value and the graphical analysis underscores the robustness of the model.

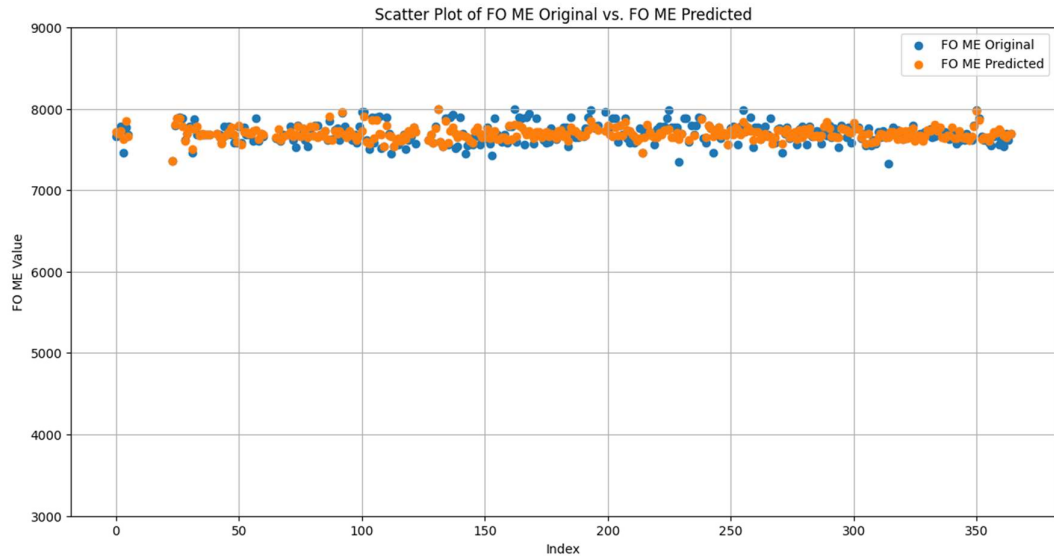


Figure 4.27 Scatter plot of fuel oil consumption original vs predicted XGBoost regression

Scatter line of the graph as can be seen that index per index the data is plotted, the model able to predict the consumption using 5 variables; ship speed, displacement, beaufort number, draft, and sea condition. The XGBoost graph results in a more accurate plot between the actual value and predicted value. The plotting is not fully aligned, some were align, then the model could not be perfectly used for the prediction.

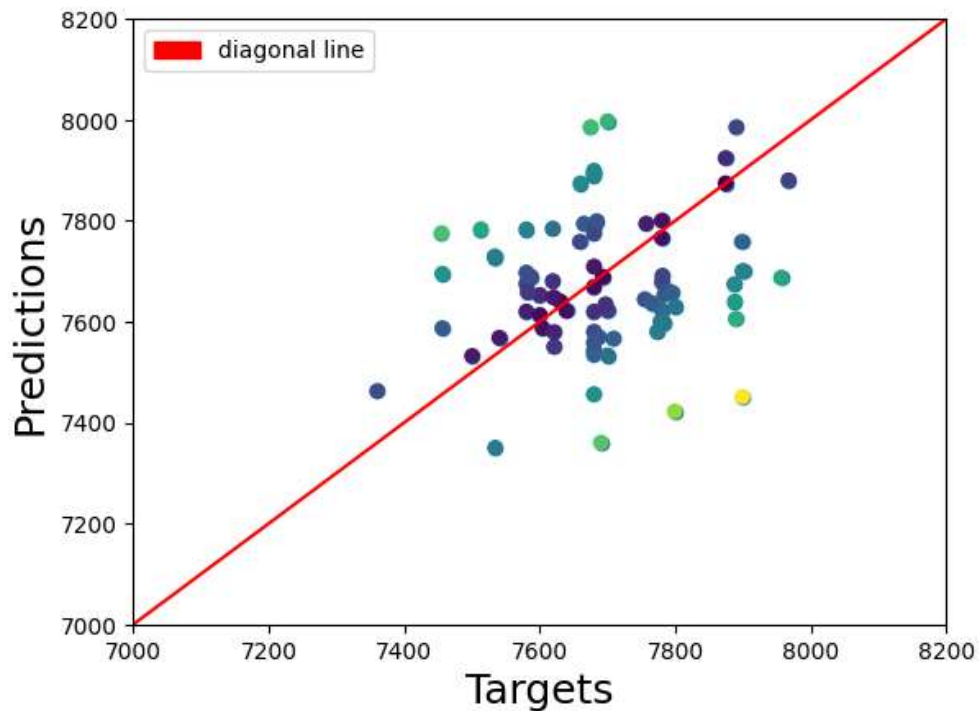


Figure 4.28 XGBoost Regressor Predictions vs target

To make the model much more accurate then it need to be more data used for the prediction. The scaling also need to be used for the gradient boosting so the data can be more accurate for continuous model learning.

The relation between the predicted fuel consumption values and ship speed can be seen on figure 4.29

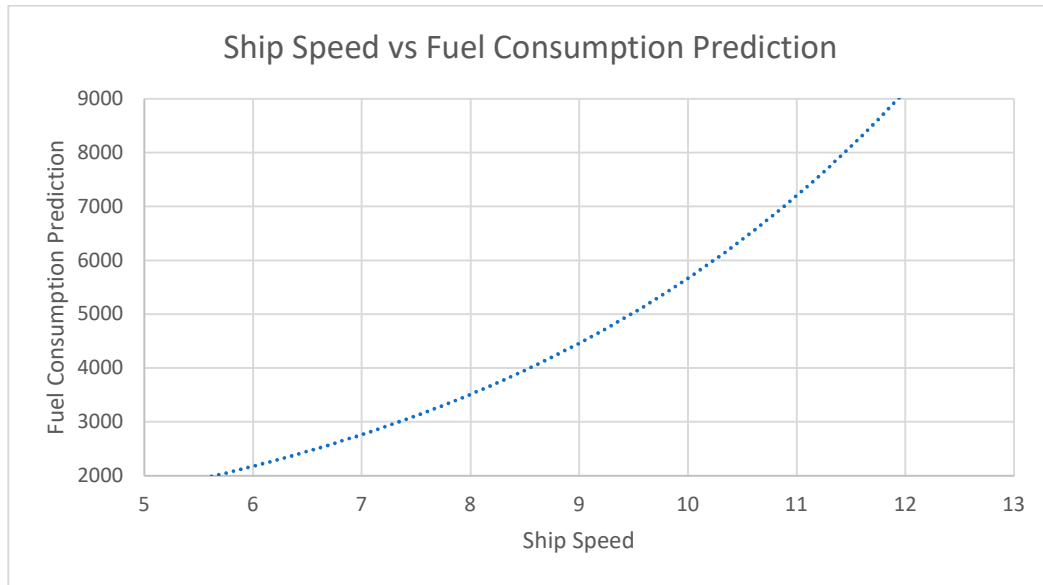


Figure 4.29 Relation between Ship speed vs Fuel consumption prediction

The formula relation between Speed Fuel Consumption and Ship Speed was determined as;

$$BHP = K1 * K2 * VS^3$$

The relationship between the formula and the graph illustrates a clear correlation between Brake Horsepower (BHP) and fuel consumption, demonstrating how ship fuel consumption is related to ship speed raised to the power of three. This cubic relationship indicates that as ship speed increases, fuel consumption rises significantly, highlighting the non-linear nature of this dependency. The graph effectively supports this finding, as the plotted results align closely with the predictions made by the formula. This correlation underscores the importance of understanding these dynamics for optimizing fuel efficiency in maritime operations.

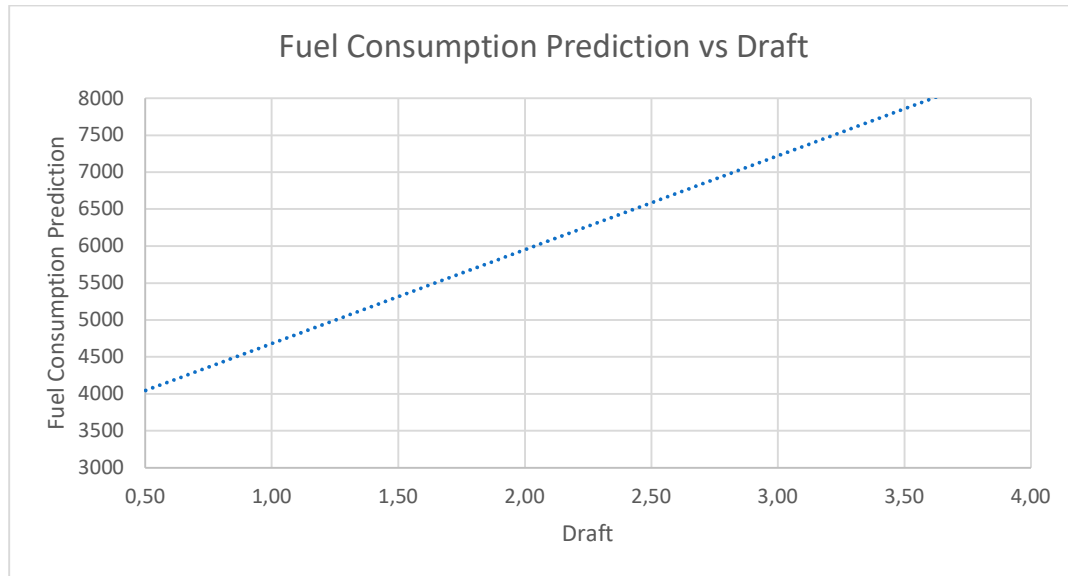


Figure 4.30 Draft vs Fuel Consumption Prediction

The relation of figure 4.17 is ship middle draft and ship fuel consumption prediction. The graph is linear, original formula between draft and fuel consumption are;

$$BHP = k1 * Ct * \rho * wsa * Vs^2$$

$$wsa = T * surface\ area$$

Formula of fuel consumption estimation and draft can be seen that T variable is used for calculating wetted surface area (wsa). Thus the relation of the graph is linear. The graph correlation is align with the formula used.

From this the ridge method can be used for the prediction, 93,06% for the R^2 value, with 5 variables; ship speed, displacement, ship draft, beaufort number, and sea condition. Relation between the variable is displayed on the figure 4.10 and 4.11 were related to the original formula.

XGBoost method provides feature importance, the values from 5 inputs are respectively 0.010 , 0.005 , 0.005 , 0.98 ,and 0.005. Which means the input sea condition holds the more importance to the model learning and the ship service speed also provides a good number for the model.

	FO ME Original	FO ME Predicted	Residual	Percentage Differential
0	7665	7457.044922	207.955078	2.713047
1	7700	7660.073242	39.926758	0.518529
2	7784	7631.309570	152.690430	1.961593
3	7456	7998.193848	-542.193848	-7.271913
4	7774	7602.805664	171.194336	2.202140
..
360	7621	7457.044922	163.955078	2.151359
361	7543	7514.052246	28.947754	0.383770
362	7691	7496.217773	194.782227	2.532599

Figure 4.31 XGBoost original and predicted values sample

The provided data consists of the model's predicted outputs, which include residuals and percentage differentials, with additional comprehensive details available in the appendix. This analysis highlights the discrepancies between the predicted values and the actual observations, offering insights into the model's performance and accuracy. By examining the residuals and their corresponding percentage differentials, we can assess the extent of prediction errors and identify areas for potential improvement in the model's forecasting capabilities. For a deeper understanding of the data and its implications, please refer to the appendix for the complete dataset and further context..

Table 4.3 XGBoost Regression sample value

Target	Prediction	Residual	Differential (%)
7665	7457,05	207,95	2,71
7700	7660,07	39,93	0,52
7784	7631,31	152,69	1,96
7456	7998,19	-542,19	-7,27
7774	7602,81	171,19	2,20
7680	7524,72	155,28	2,02
7360	8272,17	-912,17	-12,39

The analysis of the model's performance reveals a nuanced picture when comparing the predicted values to the target values. In this study, the dataset was divided into two segments: 75% was utilized for training the model, while the remaining 25% served as a testing set to evaluate its predictive capabilities. Among the predictions made, the model achieved its closest approximation for a target value of 7700 liters, with a prediction of 7660.07 liters. This resulted in a remarkably small difference of just 0.52%, indicating that the model can perform well under certain conditions. However, the results also showcased considerable discrepancies in other predictions. For instance, when the target was set at 7360 liters, the model predicted 8272.17 liters, leading to a significant difference of 12.39%. Such variability in performance suggests that while the model has the potential for accuracy, it also faces challenges in consistently aligning its predictions with the targets across the dataset.

In light of these findings, the model was subsequently tested on the complete dataset to further evaluate its overall effectiveness and reliability. The mixed results highlight the importance of not only assessing the model's performance on a subset of data but also understanding how it behaves with the full range of inputs. The significant differences observed between some predicted and target values underscore the necessity for further refinement of the model. This could involve exploring additional features, optimizing algorithms, or employing more sophisticated techniques to improve predictive accuracy. Continuous evaluation and iterative enhancements are essential to ensure that the model can consistently meet desired performance standards. Ultimately, a comprehensive approach to model development will be crucial in harnessing its full potential and delivering reliable predictions across various scenarios..

In the context of predicting fuel consumption, the evaluation of model performance is essential to understand how accurately a model can estimate values compared to actual targets. Each prediction generated by the model is based on its underlying estimation algorithms, which analyze various input features to forecast fuel consumption.

To assess the effectiveness of these predictions, we compare the predicted values against the targeted values of fuel consumption. This comparison yields residual values, which represent the differences between the predicted and actual values. These residuals are crucial for understanding the model's accuracy, as they highlight areas where the model may underperform or excel.

To quantify the model's predictive capability, we can express these residuals as percentages. This percentage calculation provides a clear metric to evaluate the model's performance across different scenarios, allowing us to identify how closely the predictions align with the actual fuel consumption values. By analyzing these percentages, we can gain insights into the model's strengths and weaknesses, enabling us to make informed adjustments to improve accuracy.

Among various predictive modeling techniques, the XGBoost model has emerged as a powerful tool for predicting fuel consumption. Its ability to handle complex datasets and capture intricate relationships between features makes it particularly effective in this domain. By leveraging advanced algorithms and optimization techniques, XGBoost consistently delivers reliable predictions, making it a preferred choice for practitioners in the field.

In conclusion, the systematic comparison of predicted and targeted values, along with the calculation of residuals and their percentage differentials, provides a comprehensive framework for evaluating model performance. The use of XGBoost in this context not only enhances predictive accuracy but also contributes to more efficient fuel consumption management strategies.

4.9 Discussion

The normal value is 75% of the data is trained and 25% of the data is used for testing, the data is chosen randomly throughout the model. Every model is different then the testing is different between the 2 algorithms Ridge and XGBoost. Between the 2 algorithms Ridge and XGBoost it can be seen that XGBoost having the most accurate value than the Ridge regressor value this XGBoost having the highest R^2 score of 0,9306 and the Ridge has R^2 around 0,8882. Which means for the KMP. Kirana VII Passenger Roro ship using the XGBoost algorithm is better than Ridge regressor for continuous data prediction. XGBoost having the advantage due to more complexity of the model in correlating all the variables given which the machine learning try to work on.

Both model is validated through testing in making graph between the fuel consumption and the variables taken to make the model. For example the draft correlation to the fuel consumption is linear, while the ship speed correlation to the fuel consumption is ship speed to the power of 3. For the displacement and others that could be seen on the formula, ship fuel consumption mainly depends on the resistance and ship speed. If the variable could be more described and check on every aspect, then the model could capture more connection to the output.

The value could change if the data is much more than intended. For example if the data is 5 times larger in sample the value could change. The current data is a much more accurate XGBoost than Ridge regressor. Xgboost algorithm is specified into some grids which the difference resulting “best model”.

XGBoost, a gradient boosting algorithm, is capable of modeling non-linear relationships between features, which is crucial for our dataset. Ridge, a linear regression technique, is limited to modeling linear relationships and may not capture the intricate patterns present in the data. XGBoost's iterative learning process, where each iteration builds upon the previous one (grid), allows it to adapt to the nuances of the dataset more effectively. This iterative process enables XGBoost to identify and correct errors more efficiently, leading to better predictions. Ridge, does not have this iterative learning capability and relies solely on the initial parameters.

Furthermore, XGBoost's use of gradient boosting, which combines multiple weak models into a strong one, provides a robust and stable solution. This approach helps to reduce overfitting and improve generalization, which is essential for real-world applications. Ridge, as

a linear model, is more prone to overfitting and may not generalize well to unseen data. The superior performance of XGBoost over Ridge in the datasets can be attributed to its ability to handle complex interactions, iterative learning, and robustness against overfitting. These advantages make XGBoost a more suitable choice for the dataset and demonstrate its potential for better predictive performance in similar applications.

In overall the model accuracy could be higher than this because if the data taken is in the lower interval resulting the number of samples increase, thus resulting in the increased accuracy for the model. Furthermore the increase in correlating variables of the output (fuel consumption) also help the model to learn more and resulting in the better value of accuracy. Future research could focus on exploring other algorithms that can handle complex interactions and iterative learning, such as neural networks and decision trees. These algorithms could be used in combination with XGBoost to further improve the model's performance. Additionally, more data could be collected from different sources to increase the sample size and improve the model's generalizability. This would allow the model to better capture the nuances of the data and make more accurate predictions.

CHAPTER 5

Conclusion

5.1 Conclusion

In this study the regression method that compared between Ridge and XGBoost using the Visual Studio Code. The conclusion between the 2 algorithm is that XGBoost is much more accurate for KMP. Kirana VII having the R^2 score of 0,9306. XGBoost is usable if the numbers data taken is much more. The model could be used for estimating the fuel consumption for KMP. Kirana VII. The graph can be seen that the Ridge is much more accurate than XGBoost method.

The comparison between Ridge and XGBoost regression algorithms for estimating fuel consumption in ships using the Visual Studio Code revealed that XGBoost is significantly more accurate than Ridge. The R^2 score for XGBoost was 0.9306, while Ridge achieved an R^2 score of 0.887. The results demonstrate that XGBoost outperformed Ridge in terms of accuracy. This superiority can be attributed to XGBoost's ability to handle complex interactions between features and its robustness against overfitting.

Study highlights the effectiveness of XGBoost in estimating fuel consumption for the KMP. Kirana VII. The high R^2 score achieved by XGBoost indicates its potential for better predictive performance in similar applications. Future research could explore other algorithms that can handle complex interactions and iterative learning, such as neural networks and decision trees, to further improve the model's accuracy and generalizability.

5.2 Suggestion

Based on the research some suggestion could be made:

1. Increasing the accuracy of the machine learning is by adding more numbers of data so the accuracy would be increased.
2. Parameters could be added more to increase the accuracy
3. Data collecting could be digitalized due to the precision of manual data collection to minimize human error.

In summary, the centralization of displacement data and the assumption of a constant Coefficient Block are integral to the efficient calculation of ship displacement. While

these practices enhance the speed and consistency of assessments, it is crucial for professionals in the field to be aware of their limitations and to incorporate empirical validation into their processes. By doing so, they can ensure that their calculations remain robust and reflective of the actual performance of the vessel in various conditions.

REFERENCES

- Agand, P., Kennedy, A., Harris, T., Bae, C., Chen, M., & Park, E. J. (2023). Fuel consumption prediction for a passenger ferry using machine learning and in-service data: A comparative study. *Ocean Engineering*, 284, 115271. <https://doi.org/10.1016/j.oceaneng.2023.115271>
- Bonaccorso, G. (2017). *Machine Learning Algorithms*. Birmingham: Packt Publishing
- Cakmak & Das. (2018). *Hands-On Automated Machine Learning*. Packt Publishing.
- Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 2(01), 20 - 28.
- Cheng, Fei & Yang, Chunhua & Zhou, Can & Lan, Lijuan & Zhu, Hongqiu & Li, Yonggang. (2020). Simultaneous Determination of Metal Ions in Zinc Sulfate Solution Using UV–Vis Spectrometry and SPSE-XGBoost Method. *Sensors*. 20. 4936. 10.3390/s20174936.
- Gómez Ruiz, M.Á.; de Almeida, I.M.; Pérez Fernández, R. Application of Machine Learning Techniques to the Maritime Industry. *J. Mar. Sci. Eng.* 2023, 11, 1820. <https://doi.org/10.3390/jmse11091820>
- Guo, B., Liang, Q., Tvete, H. A., Brinks, H., & Vanem, E. (2022). Combined machine learning and physics-based models for estimating fuel consumption of cargo ships. *Ocean Engineering*, 255. <https://doi.org/10.1016/j.oceaneng.2022.111435>
- Hamed, M. A., Khafagy, M. H., & Badry, R. M. (2021). Fuel Consumption Prediction Model using Machine Learning. *International Journal of Advanced Computer Science and Applications*, 12(11), 406–414.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.
- Kelby, R. J. (2020). Formalized Generalization Bounds for Perceptron-Like Algorithms, (August).
- Kim, Y. R., Steen, S., Kramel, D., Muri, H., & Strømman, A. H. (2023). Modelling of ship resistance and power consumption for the global fleet: The MariTEAM model. *Ocean Engineering*, 281. <https://doi.org/10.1016/j.oceaneng.2023.114758>
- Kim, Y.-R.; Jung, M.; Park, J.-B. Development of a Fuel Consumption Prediction Model Based on Machine Learning Using Ship In-Service Data. *J. Mar. Sci. Eng.* 2021, 9, 137.

- Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, 30(2-3), 203-222.
- Park, J., & Oh, J. (2023). A machine learning based predictive maintenance algorithm for ship generator engines using engine simulations and collected ship data. *Energy*, 285. <https://doi.org/10.1016/j.energy.2023.129269>
- [Scikit-learn: Machine Learning in Python](#), Pedregosa *et al.*, *JMLR* 12, pp. 2825-2830, 2011.
- Uyanik, T., Karatuğ, Ç., & Arslanoğlu, Y. (2020). Machine learning approach to ship fuel consumption: A case of container vessel. *Transportation Research Part D: Transport and Environment*, 84. <https://doi.org/10.1016/j.trd.2020.102389>
- van Rossum, G., Drake, F. L., & O'Hair, A. (2018). Python language and its impact on scientific computing. *Computing in Science & Engineering*, 20(2), 95-99.
- Wijanarko, Fernanda Rizqi. (2020). PREDICTION OF SHIP FUEL CONSUMPTION AT MV MERATUS WAINGAPU USING AUTOMATED MACHINE LEARNING. Institute Technology Sepuluh Nopember.
- Xie, X.; Sun, B.; Li, X.; Olsson, T.; Maleki, N.; Ahlgren, F. Fuel Consumption Prediction Models Based on Machine Learning and Mathematical Methods. *J. Mar. Sci. Eng.* 2023, 11, 738.
- Z. Hu, Y. Jin, Q. Hu, S. Sen, T. Zhou, & M. T. Osman. (2019). Prediction of Fuel Consumption for Enroute Ship Based on Machine Learning. *IEEE Access*, 7, 119497–119505. <https://doi.org/10.1109/ACCESS.2019.2933630>
- Zhang, M., Tsoulakos, N., Kujala, P., & Hirdaris, S. (2024). A deep learning method for the prediction of ship fuel consumption in real operational conditions. *Engineering Applications of Artificial Intelligence*, 130. <https://doi.org/10.1016/j.engappai.2023.107425>

ATTACHMENT A

Data Example


Example of data

Current Time Date	Sailing speed (Knots)	Beaufort	Displacement (Tons)	Sea Condition	Draft (M)	Fuel consumption (1/day)
1 07 2023	12,5000	2	1990,6320	4,0000	3,4	7997,0000
2 07 2023	0,0000	1	0,0000	3,0000	0	7593,0000
3 07 2023	12,5000	1	1990,6320	3,0000	3,4	7536,0000
4 07 2023	12,0000	1	1873,5360	2,0000	3,2	7887,0000
5 07 2023	12,0000	1	1814,9880	2,0000	3,1	7566,0000
6 07 2023	11,5000	1	1932,0840	3,0000	3,3	7891,0000
7 07 2023	11,2000	1	2049,1800	3,0000	3,5	7924,0000
8 07 2023	11,6000	1	1873,5360	2,0000	3,2	7513,0000
9 07 2023	11,0000	1	1814,9880	2,0000	3,1	7541,0000
10 07 2023	11,3000	1	1932,0840	2,0000	3,3	7895,0000
11 07 2023	11,2000	1	1932,0840	3,0000	3,3	7696,0000
12 07 2023	11,2000	1	1873,5360	3,0000	3,2	7451,0000
13 07 2023	11,5000	1	1932,0840	2,0000	3,3	7545,0000
14 07 2023	11,7000	1	1932,0840	3,0000	3,3	7615,0000
15 07 2023	12,0000	1	1932,0840	3,0000	3,3	7559,0000

ATTACHMENT B

Ship Particular

Ship Particular

	P.T. DHARMA LAUTAN UTAMA armada pelayaran nasional	FM 4006 - 2006	versi 1.3
DATA-DATA KAPAL PARTICULARS OF SHIP			
I. PEMILIK	:	PT.DHARMA LAUTAN UTAMA	
II. LINTAS PENYEBERANGAN	:	
III. KAPAL PENYEBERANGAN	:	
1. Nama Kapal	:	KM.KIRANA-VII	
2. Tempat Pembuatan	:	INDONESIA	
3. Galangan Pembangunan	:	PT.DOCK ASSI	
4. Tahun Pembuatan	:	2021	
5. Bahan	:	BAJA	
6. Type Kapal	:	FERRY-RORO	
7. Klasifikasi	:	BKI	
8. Surat Ukur No.	:	2021 Ka.No.8827/L	
IV. UKURAN UTAMA	:	
1. Panjang seluruhnya	:	68,94	METER
2. Panjang Garis Air	:	60,00	METER
3. Lebar	:	17,00	METER
4. Dalam	:	4,50	METER
5. Sarat Maximum	:	3,10	METER
6. G.R.T	:	1931
V. KAPASITAS TANGKI	:	
1. Tangki Bahan Bakar	:	110,03	TON
2. Tangki Air Tawar	:	77,5	TON
3. Tangki Ballast	:	
VI. MESIN UTAMA	:	
1. Merk	:	MITSUBHISI	
2. Type	:	SI2R-T2MPTK	
3. Tenaga Kuda / PK	:	1400 hp	
4. Jumlah Mesin	:	2	
5. Kecepatan Maksimum	:	13 KNOT	
6. Th. Pembuatan Mesin	:	2021	
7. R.P.M	:	1650	
8. Jenis Bahan Bakar	:	SOLAR	
VII. MESIN BANTU	:	
1. Merk	:	DOOSAN INTRACORE	
2. Type	:	AD.126 TIF	
3. Tenaga Kuda / PK	:	280X3 UNIT	
4. Jumlah Mesin	:	3 UNIT	
VIII. KAPASITAS MUAT	:	
1. Jumlah penumpang	:	Penumpang	
2. Jumlah Kendaraan	:	TB.21	KK.17
3. Jumlah Crew	:	27 Orang	
IX. PINTU RAMPA	:	
1. Pintu Rampa Haluan	=	P : 6,25	Mtr, L : 6,30 Mtr
2. Pintu Rampa Buritan	=	P : 6,25	Mtr, L : 6,30 Mtr
3. Pintu Rampa Kiri	=	P :	Mtr, L : Mtr
4. Pintu Rampa Kanan	=	P : 10	Mtr, L : 6,30 Mtr

ATTACHMENT C

Full Ship Fuel Consumption Data

Ship Fuel Consumption Data

Current Time Date	Sailing speed	Beaufort	Displacement	Sea Condition	Draft	Fuel consumption rate
20 02 2023	11,0000	1	1873,5360	2,0000	3,20	7665,0000
21 02 2023	11,6000	1	1873,5360	2,0000	3,20	7700,0000
22 02 2023	10,7000	1	1990,6320	3,0000	3,40	7784,0000
23 02 2023	11,7000	1	2049,1800	3,0000	3,50	7456,0000
24 02 2023	10,7000	1	1932,0840	3,0000	3,30	7774,0000
25 02 2023	11,2000	1	1873,5360	2,0000	3,20	7680,0000
26 02 2023	10,7000	1		3,0000	NaN	350,0000
27 02 2023	NaN	0	1756,4400	1,0000	3,00	133,0000
28 02 2023	NaN	0		NaN	NaN	NaN
1 03 2023	NaN	0		NaN	NaN	NaN
2 03 2023	NaN	0		NaN	NaN	NaN
3 03 2023	NaN	0		NaN	NaN	NaN
4 03 2023	NaN	0		NaN	NaN	NaN
5 03 2023	NaN	0		NaN	NaN	NaN
6 03 2023	NaN	0		NaN	NaN	NaN
7 03 2023	NaN	0		NaN	NaN	NaN
8 03 2023	NaN	0		NaN	NaN	NaN
9 03 2023	NaN	0		NaN	NaN	NaN
10 03 2023	NaN	0		NaN	NaN	NaN
11 03 2023	NaN	0		NaN	NaN	NaN
12 03 2023	NaN	0		NaN	NaN	NaN

13 03 2023	NaN	0		NaN	NaN	NaN
14 03 2023	NaN	0		NaN	NaN	NaN
15 03 2023	12,5000	0	1932,0840	5,0000	3,30	7360,0000
16 03 2023	9,7000	1	2107,7280	2,0000	3,60	7800,0000
17 03 2023	12,3000	1	1990,6320	5,0000	3,40	7888,0000
18 03 2023	10,2000	1	2049,1800	2,0000	3,50	7899,0000
19 03 2023	12,7000	1	1990,6320	2,0000	3,40	7889,0000
20 03 2023	9,7000	1	1814,9880	2,0000	3,10	7619,0000
21 03 2023	12,7000	1	1697,8920	2,0000	2,90	7700,0000
22 03 2023	10,2000	1	1756,4400	2,0000	3,00	7784,0000
23 03 2023	10,2000	1	1814,9880	2,0000	3,10	7456,0000
24 03 2023	10,7000	1	1814,9880	2,0000	3,10	7874,0000
25 03 2023	12,7000	1	1990,6320	2,0000	3,40	7680,0000
26 03 2023	10,0000	1	1932,0840	2,0000	3,30	7680,0000
27 03 2023	12,0000	1	2107,7280	5,0000	3,60	7680,0000
28 03 2023	10,2000	1	1990,6320	2,0000	3,40	7680,0000
29 03 2023	12,2000	1	1990,6320	2,0000	3,40	7680,0000
30 03 2023	10,6000	1	2166,2760	2,0000	3,70	7680,0000
31 03 2023	NaN	1		NaN	NaN	NaN
1 04 2023	NaN	1		NaN	NaN	NaN
2 04 2023	12,1000	1	1990,6320	2,0000	3,40	7693,0000
3 04 2023	11,0000	1	1756,4400	2,0000	3,00	7680,0000
4 04 2023	12,7000	1	2107,7280	2,0000	3,60	7580,0000
5 04 2023	11,0000	1	1932,0840	2,0000	3,30	7780,0000
6 04 2023	11,8000	1	1932,0840	2,0000	3,30	7680,0000
7 04 2023	11,0000	1	1873,5360	2,0000	3,20	7780,0000

8 04 2023	11,2000	1	2049,1800	3,0000	3,50	7680,0000
9 04 2023	10,2000	1	1990,6320	2,0000	3,40	7580,0000
10 04 2023	11,5000	1	2107,7280	2,0000	3,60	7580,0000
11 04 2023	11,4000	1	2049,1800	2,0000	3,50	7780,0000
12 04 2023	11,3000	1	2107,7280	3,0000	3,60	7580,0000
13 04 2023	11,0000	1	1873,5360	2,0000	3,20	7773,0000
14 04 2023	12,0000	1	1990,6320	2,0000	3,40	7680,0000
15 04 2023	11,1000	1	1990,6320	2,0000	3,40	7687,0000
16 04 2023	11,6000	1	1932,0840	2,0000	3,30	7709,0000
17 04 2023	11,0000	1	2049,1800	2,0000	3,50	7600,0000
18 04 2023	11,2000	1	1756,4400	2,0000	3,00	7887,0000
19 04 2023	11,2000	1	1639,3440	2,0000	2,80	7604,0000
20 04 2023	11,8000	1	1814,9880	6,0000	3,10	7700,0000
21 04 2023	11,5000	1	1756,4400	2,0000	3,00	7660,0000
22 04 2023	NaN	1		NaN	NaN	NaN
23 04 2023	NaN	1		NaN	NaN	NaN
24 04 2023	NaN	1		NaN	NaN	NaN
25 04 2023	NaN	1	1639,3440	NaN	2,80	300,0000
26 04 2023	11,5000	1	1873,5360	6,0000	3,20	7639,0000
27 04 2023	10,7000	1	1814,9880	2,0000	3,10	7630,0000
28 04 2023	12,2000	1	2049,1800	2,0000	3,50	7600,0000
29 04 2023	11,1000	1	1697,8920	2,0000	2,90	7680,0000
30 04 2023	NaN	1		NaN	NaN	NaN
1 05 2023	11,3000	1	2049,1800	2,0000	3,50	7680,0000
2 05 2023	12,0000	1	1990,6320	2,0000	3,40	7780,0000
3 05 2023	11,2000	1	1873,5360	2,0000	3,20	7621,0000

4 05 2023	11,1000	1	1932,0840	3,0000	3,30	7533,0000
5 05 2023	10,7000	1	1990,6320	5,0000	3,40	7794,0000
6 05 2023	11,2000	1	1873,5360	6,0000	3,20	7675,0000
7 05 2023	10,8000	1	1990,6320	5,0000	3,40	7755,0000
8 05 2023	11,5000	1	1932,0840	3,0000	3,30	7620,0000
9 05 2023	11,0000	1	1756,4400	2,0000	3,00	7534,0000
10 05 2023	11,4000	1	1873,5360	3,0000	3,20	7765,0000
11 05 2023	11,0000	1		2,0000	NaN	7677,0000
12 05 2023	11,5000	1		3,0000	NaN	7799,0000
13 05 2023	11,2000	1	1932,0840	2,0000	3,30	7799,0000
14 05 2023	11,0000	1		3,0000	NaN	7684,0000
15 05 2023	11,0000	1		2,0000	NaN	7667,0000
16 05 2023	11,5000	1		2,0000	NaN	7729,0000
17 05 2023	10,6000	1		2,0000	NaN	7621,0000
18 05 2023	11,2000	1		2,0000	NaN	7855,0000
19 05 2023	10,8000	1		2,0000	NaN	7630,0000
20 05 2023	10,5000	1		2,0000	NaN	7743,0000
21 05 2023	11,0000	1	1756,4400	5,0000	3,00	7757,0000
22 05 2023	10,7000	1	1990,6320	3,0000	3,40	7619
23 05 2023	10,7000	1	2049,1800	6,0000	3,50	7956,0000
24 05 2023	10,5000	1	1990,6320	2,0000	3,40	7778,0000
25 05 2023	11,0000	1	1814,9880	2,0000	3,10	7697,0000
26 05 2023	11,0000	1	2049,1800	2,0000	3,50	7659,0000
27 05 2023	11,3000	1	2049,1800	2,0000	3,50	7621,0000
28 05 2023	11,3000	1	1932,0840	3,0000	3,30	7684,0000
29 05 2023	11,2000	1	1756,4400	5,0000	3,00	7582,0000

30 05 2023	11,0000	1	1990,6320	2,0000	3,40	7781,0000
31 05 2023	11,0000	1	2049,1800	2,0000	3,50	7967,0000
1 06 2023	11,2000	1		2,0000	NaN	7967,0000
2 06 2023	10,7000	1		2,0000	NaN	7622,0000
3 06 2023	11,3000	1	2049,1800	3,0000	3,50	7500,0000
4 06 2023	10,7000	1	1756,4400	3,0000	3,00	7898,0000
5 06 2023	11,5000	1	1990,6320	2,0000	3,40	7587,0000
6 06 2023	10,3000	1	1814,9880	2,0000	3,10	7890,0000
7 06 2023	10,8000	1	1990,6320	2,0000	3,40	7901,0000
8 06 2023	11,3000	1	1932,0840	2,0000	3,30	7513,0000
9 06 2023	10,8000	1	2078,4540	3,0000	3,55	7540,0000
10 06 2023	10,9000	1	1990,6320	2,0000	3,40	7899,0000
11 06 2023	10,8000	1	2049,1800	3,0000	3,50	7690,0000
12 06 2023	11,1000	1	1873,5360	2,0000	3,20	7455,0000
13 06 2023	11,1000	1	2166,2760	3,0000	3,70	7534,0000
14 06 2023	11,0000	1	2224,8240	2,0000	3,80	7611,0000
15 06 2023	10,8000	1	1932,0840	2,0000	3,30	7563,0000
16 06 2023	11,1000	1	1639,3440	2,0000	2,80	7605,0000
17 06 2023	11,0000	1	1990,6320	2,0000	3,40	7687,0000
18 06 2023	11,1000	1	1814,9880	2,0000	3,10	7504,0000
19 06 2023	11,0000	1	2049,1800	2,0000	3,50	7621,0000
20 06 2023	11,1000	1	1990,6320	2,0000	3,40	7661,0000
21 06 2023	11,3000	1	1697,8920	2,0000	2,90	7775,0000
22 06 2023	11,2000	1	1990,6320	2,0000	3,40	7569,0000
23 06 2023	NaN	0		1,0000	NaN	368,0000
24 06 2023	NaN	0		1,0000	NaN	NaN

25 06 2023	NaN	0		1,0000	NaN	NaN
26 06 2023	NaN	0	1697,8920	1,0000	2,90	343,0000
27 06 2023	12,4000	1	1990,6320	3,0000	3,40	7621,0000
28 06 2023	11,9000	1	2049,1800	2,0000	3,50	7684,0000
29 06 2023	11,6000	1		2,0000	NaN	7582,0000
30 06 2023	12,0000	1	2049,1800	2,0000	3,50	7783,0000
1 07 2023	12,5000	2	1990,6320	4,0000	3,40	7997,0000
2 07 2023	NaN	1		3,0000	NaN	7593,0000
3 07 2023	12,5000	1	1990,6320	3,0000	3,40	7536,0000
4 07 2023	12,0000	1	1873,5360	2,0000	3,20	7887,0000
5 07 2023	12,0000	1	1814,9880	2,0000	3,10	7566,0000
6 07 2023	11,5000	1	1932,0840	3,0000	3,30	7891,0000
7 07 2023	11,2000	1	2049,1800	3,0000	3,50	7924,0000
8 07 2023	11,6000	1	1873,5360	2,0000	3,20	7513,0000
9 07 2023	11,0000	1	1814,9880	2,0000	3,10	7541,0000
10 07 2023	11,3000	1	1932,0840	2,0000	3,30	7895,0000
11 07 2023	11,2000	1	1932,0840	3,0000	3,30	7696,0000
12 07 2023	11,2000	1	1873,5360	3,0000	3,20	7451,0000
13 07 2023	11,5000	1	1932,0840	2,0000	3,30	7545,0000
14 07 2023	11,7000	1	1932,0840	3,0000	3,30	7615,0000
15 07 2023	12,0000	1	1932,0840	3,0000	3,30	7559,0000
16 07 2023	11,0000	1	2049,1800	2,0000	3,50	7662,0000
17 07 2023	11,4000	1	2049,1800	3,0000	3,50	7669,0000
18 07 2023	10,8000	1	1990,6320	2,0000	3,40	7566,0000
19 07 2023	11,0000	1	1990,6320	3,0000	3,40	7621,0000
20 07 2023	11,2000	1	1932,0840	2,0000	3,30	7661,0000

21 07 2023	12,0000	1	2107,7280	3,0000	3,60	7775,0000
22 07 2023	11,5000	1	1990,6320	2,0000	3,40	7569,0000
23 07 2023	11,0000	1	1932,0840	3,0000	3,30	7422,0000
24 07 2023	11,8000	1	1990,6320	2,0000	3,40	7880,0000
25 07 2023	11,5000	1	1873,5360	2,0000	3,20	7680,0000
26 07 2023	12,2000	1	1932,0840	2,0000	3,30	7637,0000
27 07 2023	11,0000	1	1990,6320	3,0000	3,40	7594,0000
28 07 2023	11,5000	1	1932,0840	2,0000	3,30	7684,0000
29 07 2023	11,7000	1	1990,6320	3,0000	3,40	7582,0000
30 07 2023	11,7000	1	1990,6320	2,0000	3,40	7783,0000
31 07 2023	11,7000	1	2049,1800	3,0000	3,50	7783,0000
1 08 2023	11,7000	1	2049,1800	2,0000	3,50	7997,0000
2 08 2023	11,3000	1	1932,0840	3,0000	3,30	7593,0000
3 08 2023	11,7000	1	1990,6320	2,0000	3,40	7890,0000
4 08 2023	11,2000	1	2049,1800	3,0000	3,50	7887,0000
5 08 2023	11,5000	1	1932,0840	3,0000	3,30	7566,0000
6 08 2023	11,0000	1	1932,0840	3,0000	3,30	7894,0000
7 08 2023	11,5000	1	1932,0840	3,0000	3,30	7934,0000
8 08 2023	10,8000	1	1990,6320	2,0000	3,40	7683,0000
9 08 2023	11,8000	1	1873,5360	3,0000	3,20	7569,0000
10 08 2023	11,2000	1	1990,6320	2,0000	3,40	7888,0000
11 08 2023	12,0000	1	1990,6320	2,0000	3,40	7709,0000
12 08 2023	11,2000	1	1932,0840	2,0000	3,30	7697,0000
13 08 2023	11,3000	1	1932,0840	2,0000	3,30	7612,0000
14 08 2023	11,0000	1	1873,5360	2,0000	3,20	7579,0000
15 08 2023	11,1000	1	1873,5360	2,0000	3,20	7596,0000

16 08 2023	11,1000	1	1990,6320	2,0000	3,40	7766,0000
17 08 2023	11,2000	1	1932,0840	3,0000	3,30	7711,0000
18 08 2023	11,5000	1	1990,6320	2,0000	3,40	7623,0000
19 08 2023	11,7000	1	1873,5360	2,0000	3,20	7701,0000
20 08 2023	11,2000	1	1990,6320	3,0000	3,40	7698,0000
21 08 2023	11,2000	1	1873,5360	3,0000	3,20	7775,0000
22 08 2023	11,1000	1	1932,0840	3,0000	3,30	7767,0000
23 08 2023	11,5000	1	1814,9880	3,0000	3,10	7540,0000
24 08 2023	11,1000	1	1814,9880	3,0000	3,10	7899,0000
25 08 2023	11,5000	1	1932,0840	3,0000	3,30	7680,0000
26 08 2023	11,5000	1	1873,5360	3,0000	3,20	7649,0000
27 08 2023	11,0000	1	1873,5360	5,0000	3,20	7654,0000
28 08 2023	11,2000	1	1873,5360	2,0000	3,20	7684,0000
29 08 2023	11,3000	1	1873,5360	2,0000	3,20	7665,0000
30 08 2023	11,2000	1	1932,0840	3,0000	3,30	7758,0000
31 08 2023	10,7000	1	1990,6320	2,0000	3,40	7895,0000
1 09 2023	10,7000	1	1932,0840	3,0000	3,30	7985,0000
2 09 2023	11,6000	1	1990,6320	2,0000	3,40	7793,0000
3 09 2023	10,6000	1	1932,0840	3,0000	3,30	7766,0000
4 09 2023	10,8000	1	2049,1800	2,0000	3,50	7800,0000
5 09 2023	NaN	0		1,0000	NaN	NaN
6 09 2023	NaN	0	2049,1800	NaN	3,50	175,0000
7 09 2023	11,5000	1	1873,5360	3,0000	3,20	7959,0000
8 09 2023	11,7000	1	1990,6320	2,0000	3,40	7783,0000
9 09 2023	11,3000	1	1932,0840	2,0000	3,30	7769,0000
10 09 2023	11,0000	1	1932,0840	2,0000	3,30	7888,0000

11 09 2023	11,4000	1	2049,1800	3,0000	3,50	7889,0000
12 09 2023	11,2000	1	1932,0840	3,0000	3,30	7597,0000
13 09 2023	11,7000	1	1814,9880	3,0000	3,10	7800,0000
14 09 2023	11,8000	1	2049,1800	2,0000	3,50	7679,0000
15 09 2023	11,4000	1	1932,0840	3,0000	3,30	7886,0000
16 09 2023	11,2000	1	1990,6320	3,0000	3,40	7644,0000
17 09 2023	11,5000	1	1932,0840	3,0000	3,30	7587,0000
18 09 2023	11,0000	1	1932,0840	3,0000	3,30	7644,0000
19 09 2023	11,2000	1	1873,5360	3,0000	3,20	7587,0000
20 09 2023	11,7000	1	1932,0840	3,0000	3,30	7612,0000
21 09 2023	11,5000	1	1990,6320	2,0000	3,40	7765,0000
22 09 2023	11,2000	2	1873,5360	4,0000	3,20	7462,0000
23 09 2023	11,2000	1	1932,0840	2,0000	3,30	7691,0000
24 09 2023	11,7000	1	2049,1800	2,0000	3,50	7767,0000
25 09 2023	12,0000	1	1990,6320	2,0000	3,40	7692,0000
26 09 2023	11,2000	1	1932,0840	3,0000	3,30	7683,0000
27 09 2023	11,5000	1	1932,0840	2,0000	3,30	7559,0000
28 09 2023	11,5000	1	1932,0840	2,0000	3,30	7754,0000
29 09 2023	11,5000	1	1990,6320	3,0000	3,40	7784,0000
30 09 2023	11,0000	1	1932,0840	3,0000	3,30	7765,0000
1 10 2023	11,0000	0	1873,5360	5,0000	3,20	7785,0000
2 10 2023	11,2000	1	1990,6320	3,0000	3,40	7889,0000
3 10 2023	11,5000	1	2049,1800	2,0000	3,50	7980,0000
4 10 2023	11,2000	1	1932,0840	2,0000	3,30	7770,0000
5 10 2023	11,5000	1	1932,0840	2,0000	3,30	7770,0000
6 10 2023	11,5000	1	1990,6320	3,0000	3,40	7770,0000

7 10 2023	11,5000	1	1873,5360	2,0000	3,20	7350,0000
8 10 2023	10,8000	1	1990,6320	2,0000	3,40	7700,0000
9 10 2023	NaN	0		NaN	NaN	7888,0000
10 10 2023	NaN	0		NaN	NaN	7889,0000
11 10 2023	11,0000	0	1932,0840	2,0000	3,30	7597,0000
12 10 2023	NaN	0		NaN	NaN	7800,0000
13 10 2023	NaN	0	1990,6320	2,0000	3,40	7679,0000
14 10 2023	NaN	0		1,0000	NaN	7800,0000
15 10 2023	NaN	0		NaN	NaN	7890,0000
16 10 2023	NaN	0		3,0000	NaN	7886,0000
17 10 2023	NaN	0		NaN	NaN	7644,0000
18 10 2023	11,2000	1	1932,0840	2,0000	3,30	7587,0000
19 10 2023	10,9000	1	1990,6320	2,0000	3,40	7800,0000
20 10 2023	11,4000	1	1990,6320	2,0000	3,40	7765,0000
21 10 2023	11,0000	1	1932,0840	2,0000	3,30	7462,0000
22 10 2023	10,8000	1	1990,6320	2,0000	3,40	7691,0000
23 10 2023	10,7000	0	2049,1800	5,0000	3,50	7767,0000
24 10 2023	11,0000	1	1990,6320	2,0000	3,40	7640,0000
25 10 2023	10,3000	1	1990,6320	2,0000	3,40	7692,0000
26 10 2023	11,0000	1	1990,6320	2,0000	3,40	7683,0000
27 10 2023	10,1000	0	1932,0840	5,0000	3,30	7559,0000
28 10 2023	11,2000	1	1990,6320	3,0000	3,40	7759,0000
29 10 2023	10,2000	1	1990,6320	2,0000	3,40	7784,0000
30 10 2023	11,6000	1	1990,6320	2,0000	3,40	7765,0000
31 10 2023	10,7000	1	1990,6320	2,0000	3,40	7558,0000
1 11 2023	11,3000	1	2049,1800	2,0000	3,50	7784,0000

2 11 2023	10,5000	1	2049,1800	2,0000	3,50	7985,0000
3 11 2023	11,2000	1	1990,6320	2,0000	3,40	7793,0000
4 11 2023	11,0000	0	1990,6320	5,0000	3,40	7766,0000
5 11 2023	11,0000	1	2049,1800	2,0000	3,50	7899,0000
6 11 2023	10,9000	1	1932,0840	2,0000	3,30	7527,0000
9 11 2023	10,8000	1	1873,5360	2,0000	3,20	7769,0000
10 11 2023	11,0000	1	1873,5360	2,0000	3,20	7768,0000
11 11 2023	11,1000	1	1814,9880	2,0000	3,10	7629,0000
12 11 2023	11,0000	1	1814,9880	2,0000	3,10	7699,0000
13 11 2023	11,6000	1	1873,5360	2,0000	3,20	7788,0000
14 11 2023	11,0000	1	1932,0840	2,0000	3,30	7688,0000
15 11 2023	11,1000	1	1873,5360	2,0000	3,20	7886,0000
16 11 2023	10,5000	1	1932,0840	2,0000	3,30	7644,0000
17 11 2023	11,2000	1	1932,0840	2,0000	3,30	7587,0000
18 11 2023	10,2000	0	1814,9880	5,0000	3,10	7755,0000
19 11 2023	11,0000	1	1873,5360	3,0000	3,20	7765,0000
20 11 2023	10,5000	1	1932,0840	2,0000	3,30	7462,0000
21 11 2023	10,5000	1	1756,4400	2,0000	3,00	7691,0000
22 11 2023	10,5000	1	1932,0840	5,0000	3,30	7767,0000
23 11 2023	10,4000	1	1873,5360	2,0000	3,20	7640,0000
24 11 2023	10,5000	1	1990,6320	2,0000	3,40	7692,0000
25 11 2023	10,8000	1	1873,5360	2,0000	3,20	7683,0000
26 11 2023	10,8000	1	1814,9880	2,0000	3,10	7559,0000
27 11 2023	11,8000	1	1873,5360	2,0000	3,20	7754,0000
28 11 2023	11,0000	1	1814,9880	2,0000	3,10	7784,0000
29 11 2023	10,6000	1	1990,6320	2,0000	3,40	7765,0000

30 11 2023	10,8000	1	1990,6320	2,0000	3,40	7558,0000
1 12 2023	11,2000	1	1785,7140	2,0000	3,05	7784,0000
2 12 2023	11,0000	1	1990,6320	2,0000	3,40	7664,0000
3 12 2023	11,2000	1	1873,5360	2,0000	3,20	7675,0000
4 12 2023	11,0000	1	1990,6320	2,0000	3,40	7725,0000
5 12 2023	11,0000	1	1814,9880	2,0000	3,10	7654,0000
6 12 2023	11,0000	1	1990,6320	2,0000	3,40	7568,0000
7 12 2023	10,7000	0	1873,5360	5,0000	3,20	7756,0000
8 12 2023	10,5000	1	2049,1800	2,0000	3,50	7677,0000
9 12 2023	10,5000	1	1873,5360	2,0000	3,20	7800,0000
10 12 2023	10,6000	1	1756,4400	6,0000	3,00	7645,0000
11 12 2023	11,2000	1	1873,5360	2,0000	3,20	7758,0000
12 12 2023	11,0000	1	1814,9880	2,0000	3,10	7668,0000
13 12 2023	11,2000	1	1814,9880	2,0000	3,10	7532,0000
14 12 2023	11,1000	1	2049,1800	2,0000	3,50	7798,0000
15 12 2023	11,2000	0	1785,7140	5,0000	3,05	7746,0000
16 12 2023	11,2000	1	2049,1800	2,0000	3,50	7652,0000
17 12 2023	10,8000	1	1873,5360	2,0000	3,20	7776,0000
18 12 2023	11,5000	1	1990,6320	3,0000	3,40	7578,0000
19 12 2023	10,7000	1	1873,5360	3,0000	3,20	7795,0000
20 12 2023	10,8000	2	1990,6320	4,0000	3,40	7769,0000
21 12 2023	10,7000	1	1932,0840	2,0000	3,30	7768,0000
22 12 2023	11,2000	1	1932,0840	2,0000	3,30	7642,0000
23 12 2023	10,7000	1	1639,3440	6,0000	2,80	7689,0000
24 12 2023	11,0000	0	1756,4400	5,0000	3,00	7550,0000
25 12 2023	11,6000	1	1873,5360	2,0000	3,20	7763,0000

26 12 2023	12,0000	1	1814,9880	2,0000	3,10	7550,0000
27 12 2023	12,2000	1	1756,4400	2,0000	3,00	7621,0000
28 12 2023	12,2000	1	1756,4400	2,0000	3,00	7569,0000
29 12 2023	11,2000	1	1814,9880	2,0000	3,10	7720,0000
30 12 2023	11,7000	1	1932,0840	2,0000	3,30	7653,0000
31 12 2023	11,5000	1	1756,4400	2,0000	3,00	7725,0000
1 01 2024	11,5000	1	1932,0840	2,0000	3,30	7653,0000
2 01 2024	11,2000	1	1932,0840	2,0000	3,30	7325,0000
3 01 2024	12,0000	0	1990,6320	5,0000	3,40	7710,0000
4 01 2024	11,0000	1	1873,5360	2,0000	3,20	7767,0000
5 01 2024	11,7000	1	1814,9880	2,0000	3,10	7722,0000
6 01 2024	11,5000	1	1873,5360	2,0000	3,20	7741,0000
7 01 2024	11,8000	1	1990,6320	2,0000	3,40	7665,0000
8 01 2024	11,5000	1	1814,9880	2,0000	3,10	7631,0000
9 01 2024	11,8000	1	1990,6320	2,0000	3,40	7641,0000
10 01 2024	11,8000	1	1873,5360	3,0000	3,20	7677,0000
11 01 2024	11,0000	1	1990,6320	2,0000	3,40	7724,0000
12 01 2024	12,0000	1	1814,9880	3,0000	3,10	7794,0000
13 01 2024	11,7000	1	1932,0840	3,0000	3,30	7631,0000
14 01 2024	11,7000	1	1814,9880	2,0000	3,10	7634,0000
15 01 2024	11,3000	1	1873,5360	2,0000	3,20	7689,0000
16 01 2024	11,5000	1	1814,9880	3,0000	3,10	7613,0000
17 01 2024	11,2000	1	1873,5360	2,0000	3,20	7712,0000
18 01 2024	11,5000	0	1873,5360	5,0000	3,20	7729,0000
19 01 2024	11,0000	1	2049,1800	2,0000	3,50	7751,0000
20 01 2024	11,2000	1	1756,4400	2,0000	3,00	7648,0000

21 01 2024	11,7000	1	2049,1800	2,0000	3,50	7658,0000
22 01 2024	11,7000	1	1873,5360	2,0000	3,20	7725,0000
23 01 2024	11,2000	1	2049,1800	3,0000	3,50	7678,0000
24 01 2024	11,2000	1	1932,0840	2,0000	3,30	7656,0000
25 01 2024	10,8000	1	1932,0840	2,0000	3,30	7721,0000
26 01 2024	11,8000	1	2049,1800	2,0000	3,50	7652,0000
27 01 2024	10,8000	1	2049,1800	3,0000	3,50	7623,0000
28 01 2024	11,4000	1	1932,0840	2,0000	3,30	7751,0000
29 01 2024	11,5000	1	1990,6320	2,0000	3,40	7689,0000
30 01 2024	11,5000	1	1990,6320	2,0000	3,40	7569,0000
31 01 2024	11,0000	1	2049,1800	2,0000	3,50	7611,0000
1 02 2024	11,0000	1	1756,4400	2,0000	3,00	7658,0000
2 02 2024	11,5000	1	1990,6320	2,0000	3,40	7620,0000
3 02 2024	11,1000	1	1873,5360	2,0000	3,20	7689,0000
4 02 2024	10,9000	1	1932,0840	2,0000	3,30	7620,0000
5 02 2024	11,1000	1	1873,5360	2,0000	3,20	7620,0000
6 02 2024	10,6000	1	1990,6320	3,0000	3,40	7800,0000
7 02 2024	11,5000	1	1990,6320	6,0000	3,40	7980,0000
8 02 2024	11,2000	0	1990,6320	5,0000	3,40	7880,0000
9 02 2024	11,5000	1	1873,5360	2,0000	3,20	7660,0000
10 02 2024	11,2000	1	1932,0840	2,0000	3,30	7620,0000
11 02 2024	11,0000	1	1814,9880	2,0000	3,10	7620,0000
12 02 2024	10,9000	1	1814,9880	2,0000	3,10	7580,0000
13 02 2024	11,0000	1	1932,0840	2,0000	3,30	7550,0000
14 02 2024	11,2000	1	1990,6320	2,0000	3,40	7659,0000
15 02 2024	11,2000	1	1697,8920	2,0000	2,90	7674,0000

16 02 2024	11,5000	1	2049,1800	2,0000	3,50	7564,0000
17 02 2024	11,0000	1	1873,5360	2,0000	3,20	7621,0000
18 02 2024	11,0000	1	1990,6320	2,0000	3,40	7543,0000
19 02 2024	11,2000	1	1814,9880	2,0000	3,10	7691,0000
20 02 2024	10,5000	1	1990,6320	2,0000	3,40	7613,0000
21 02 2024	12,0000	0	1639,3440	5,0000	2,80	7695,0000

ATTACHMENT D

Example Code

Example of Code

```
import numpy as np
import os
import pandas as pd
from sklearn.metrics import mean_squared_error
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.datasets import make_regression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split, GridSearchCV
import joblib
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from xgboost import XGBRegressor

# Assuming train_file contains the path to the Excel file
data = pd.read_excel(train_file)

# Assuming column names in Excel are the same
X = data[['Sailing speed', 'Beaufort', 'Displacement', 'Sea Condition', 'Draft']]
y = data['Fuel consumption rate']

print (data)

import matplotlib.pyplot as plt
# Create a figure and axis object
fig, axs = plt.subplots(nrows=3, ncols=4, figsize=(15, 10))

# Iterate over the features and create a box plot for each
features = X.columns
for i, feature in enumerate(features):
    row = i // 4
    col = i % 4
    axs[row, col].boxplot(X[feature].values)
    axs[row, col].set_title(feature)

# Show the plot
plt.tight_layout()
plt.show()
```



```

data2 = new_data.copy()

data2 = data2[data2['Fuel consumption rate']>7200].copy()

data2.describe(include='all')

# Create a new dataframe with the average values and other relevant columns
avg_data = pd.DataFrame({
    'Speed': data['Sailing speed'],
    'Wind': data['Beaufort'],
    'Displ': data['Displacement'],
    'Sea': data['Sea Condition'],
    'Draftm': data['Draft'],
    'FO': data['Fuel consumption rate']
})

# Concatenate the original dataframe with the new dataframe
data = pd.concat([avg_data], axis=1)

# Extract features and target variable
X = avg_data[['Speed', 'Wind', 'Displ', 'Sea', 'Draftm']]
y = data['FO']

# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X)
X_test_scaled = scaler.transform(X)

# Define the hyperparameters grid for XGBoost
xgb_params = {
    'max_depth': [3, 5, 7],
    'learning_rate': [0.1, 0.5, 1.0],
    'n_estimators': [50, 100, 200]
}

# Initialize variables to track the best random state
best_random_state = None
best_r2 = -np.inf

```

```

# Try different random states
for random_state in range(50,200):
    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X_train_scaled, y, test_size=0.25,
random_state=random_state)

    # Perform grid search for XGBoost
    xgb = XGBRegressor()
    grid_search = GridSearchCV(xgb, xgb_params, cv=3)
    grid_search.fit(X_train, y_train)

    # Predict and evaluate the model
    y_pred = grid_search.best_estimator_.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    rmse = np.sqrt(mse) # Calculate RMSE

    # Update the best random state if the current one is better
    if r2 > best_r2:
        best_r2 = r2
        best_random_state = random_state
        best_model = grid_search

print(f'Best random state: {best_random_state}')
print(f'Best R-squared: {best_r2}')
print(f'Best hyperparameters for XGBoost: {best_model.best_params_}')
print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("Root Mean Squared Error (RMSE):", rmse)

from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV

# Create an XGBoost model
xgb_model = XGBRegressor()

# Define the hyperparameters to tune
param_grid = {

```

```

    'max_depth': [3, 5, 7],
    'learning_rate': [0.1, 0.5, 1.0],
    'n_estimators': [50, 100, 200]
}

# Create an XGBoost model with grid search
grid_search = GridSearchCV(xgb_model, param_grid, cv=5, scoring='neg_mean_squared_error')

# Fit the XGBoost model to the data
grid_search.fit(X_train_scaled, y)

# Get the best model and its parameters
best_model = grid_search.best_estimator_
best_params = grid_search.best_params_

# Print the best model and its parameters
print(f"Best Model: {best_model}")
print(f"Best Parameters: {best_params}")

# Predict using the best model
y_pred = best_model.predict(X_test_scaled)

# Ensure y and y_pred are 1-dimensional arrays
y = y.values.flatten()
y_pred = y_pred.flatten()

# Create a new DataFrame with the original and predicted 'FO ME' values
results = pd.DataFrame({
    'FO ME Original': y,
    'FO ME Predicted': y_pred
})

results.to_excel('prediction_results_XGBOOST.xlsx', index=False)
# Print the DataFrame
print(results)

import matplotlib.pyplot as plt
import pandas as pd

# Assuming y and y_pred are your actual and predicted values

results = pd.DataFrame({

```

```

    'FO ME Original': y,
    'FO ME Predicted': y_pred
})

plt.figure(figsize=(14, 7))
plt.scatter(results.index, results['FO ME Original'], label='FO ME Original')
plt.scatter(results.index, results['FO ME Predicted'], label='FO ME Predicted')
plt.xlabel('Index')
plt.ylabel('FO ME Value')
plt.title('Scatter Plot of FO ME Original vs. FO ME Predicted')
plt.legend()
plt.grid(True)
plt.ylim(3000, 9000) # Set y-axis limits
plt.show()

import matplotlib.patches as mpatches
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Ensure y and y_test have the same dimensions
if len(y) != len(y_test):
    if len(y) > len(y_test):
        y = y[:len(y_test)]
    else:
        y_test = y_test[:len(y)]

# Create a DataFrame for plotting
results = pd.DataFrame({
    'FO ME Original': y,
    'FO ME Predicted': y_test
})

# Create the scatter plot
plt.scatter(y, y_test)

# Add labels and title
plt.xlabel('Targets', size=18)
plt.ylabel('Predictions', size=18)

# Add the diagonal line
plt.plot([7200, 8000], [7200, 8000], color='red', label='diagonal line')

```

```
# Add the legend
red_patch = mpatches.Patch(color='red', label='diagonal line')
plt.legend(handles=[red_patch])

# Set the y-axis limits
plt.ylim(7200, 8000)
plt.xlim(7200, 8000)

# Calculate the distance from the fit line
distance = np.abs(y_test - y)

# Color the points based on their distance from the fit line
plt.scatter(y, y_test, c=distance, cmap='viridis')

# Display the plot
plt.show()
```

“This page is intentionally left blank”

ATTACHMENT E

XGBoost Regression Values

Table 5.1 XGBoost Prediction values

Target	Prediction	Residual	Differential (%)
7665	7457,05	207,95	2,71
7700	7660,07	39,93	0,52
7784	7631,31	152,69	1,96
7456	7998,19	-542,19	-7,27
7774	7602,81	171,19	2,20
7680	7524,72	155,28	2,02
7360	8272,17	-912,17	-12,39
7800	7131,17	668,83	8,57
7888	8610,26	-722,26	-9,16
7899	7271,85	627,15	7,94
7889	8089,30	-200,30	-2,54
7619	6988,65	630,35	8,27
7700	7946,78	-246,78	-3,20
7784	7129,33	654,67	8,41
7456	7157,84	298,16	4,00
7874	7327,03	546,97	6,95
7680	8089,30	-409,30	-5,33
7680	7147,17	532,83	6,94
7680	8565,75	-885,75	-11,53
7680	7243,35	436,65	5,69
7680	7920,11	-240,11	-3,13
7680	7464,21	215,79	2,81
7693	7886,27	-193,27	-2,51
7680	7400,04	279,96	3,65
7580	8146,31	-566,31	-7,47
7780	7485,55	294,45	3,78
7680	7756,25	-76,25	-0,99
7780	7457,05	322,95	4,15
7680	7829,00	-149,00	-1,94
7580	7243,35	336,65	4,44
7580	7740,25	-160,25	-2,11
7780	7677,91	102,09	1,31
7580	7891,35	-311,35	-4,11
7773	7457,05	315,95	4,06
7687	7547,89	139,11	1,81
7709	7688,58	20,42	0,26
7600	7542,56	57,44	0,76
7887	7467,71	419,29	5,32
7604	7410,71	193,29	2,54
7700	8574,33	-874,33	-11,35

Target	Prediction	Residual	Differential
7660	7569,23	90,77	1,18
7639	8501,32	-862,32	-11,29
7630	7327,03	302,97	3,97
7600	7948,61	-348,61	-4,59
7680	7405,37	274,63	3,58
7680	7644,07	35,93	0,47
7780	7852,43	-72,43	-0,93
7621	7524,72	96,28	1,26
7533	7738,16	-205,16	-2,72
7794	8068,85	-274,85	-3,53
7675	8399,80	-724,80	-9,44
7755	8102,69	-347,69	-4,48
7620	7873,51	-253,51	-3,33
7534	7400,04	133,96	1,78
7765	7811,17	-46,17	-0,59
7677	6544,93	1132,07	14,75
7799	6932,89	866,11	11,11
7799	7553,23	245,77	3,15
7684	6763,70	920,30	11,98
7667	6544,93	1122,07	14,64
7729	6714,12	1014,88	13,13
7621	6409,58	1211,42	15,90
7855	6612,60	1242,40	15,82
7630	6477,25	1152,75	15,11
7743	6375,74	1367,26	17,66
7757	8056,35	-299,35	-3,86
7619	7631,31	-12,31	-0,16
7956	8316,12	-360,12	-4,53
7778	7344,86	433,14	5,57
7697	7428,54	268,46	3,49
7659	7542,56	116,44	1,52
7621	7644,07	-23,07	-0,30
7684	7805,83	-121,83	-1,59
7582	8124,03	-542,03	-7,15
7781	7514,05	266,95	3,43
7967	7542,56	424,44	5,33
7967	6612,60	1354,40	17,00
7622	6443,41	1178,59	15,46
7500	7862,84	-362,84	-4,84
7898	7517,29	380,71	4,82
7587	7683,24	-96,24	-1,27

Target	Prediction	Residual	Differential
7890	7191,68	698,32	8,85
7901	7446,38	454,62	5,75
7513	7587,06	-74,06	-0,99
7540	7707,90	-167,90	-2,23
7899	7480,21	418,79	5,30
7690	7693,65	-3,65	-0,05
7455	7490,88	-35,88	-0,48
7534	7852,17	-318,17	-4,22
7611	7628,07	-17,07	-0,22
7563	7417,87	145,13	1,92
7605	7376,87	228,13	3,00
7687	7514,05	172,95	2,25
7504	7462,38	41,62	0,55
7621	7542,56	78,44	1,03
7661	7547,89	113,11	1,48
7775	7473,05	301,95	3,88
7569	7581,73	-12,73	-0,17
7621	8206,56	-585,56	-7,68
7684	7847,10	-163,10	-2,12
7582	6747,96	834,04	11,00
7783	7880,94	-97,94	-1,26
7997	8836,43	-839,43	-10,50
7593	3041,51	4551,49	59,94
7536	8240,39	-704,39	-9,35
7887	7795,43	91,57	1,16
7566	7766,92	-200,92	-2,66
7891	7873,51	17,49	0,22
7924	7829,00	95,00	1,20
7513	7660,07	-147,07	-1,96
7541	7428,54	112,46	1,49
7895	7587,06	307,94	3,90
7696	7772,00	-76,00	-0,99
7451	7743,49	-292,49	-3,93
7545	7654,74	-109,74	-1,45
7615	7941,19	-326,19	-4,28
7559	8042,70	-483,70	-6,40
7662	7542,56	119,44	1,56
7669	7896,68	-227,68	-2,97
7566	7446,38	119,62	1,58
7621	7732,82	-111,82	-1,47
7661	7553,23	107,77	1,41

Target	Prediction	Residual	Differential
7775	8128,21	-353,21	-4,54
7569	7683,24	-114,24	-1,51
7422	7704,32	-282,32	-3,80
7880	7784,76	95,24	1,21
7680	7626,24	53,76	0,70
7637	7891,61	-254,61	-3,33
7594	7732,82	-138,82	-1,83
7684	7654,74	29,26	0,38
7582	7969,69	-387,69	-5,11
7783	7750,92	32,08	0,41
7783	7998,19	-215,19	-2,76
7997	7779,42	217,58	2,72
7593	7805,83	-212,83	-2,80
7890	7750,92	139,08	1,76
7887	7829,00	58,00	0,74
7566	7873,51	-307,51	-4,06
7894	7704,32	189,68	2,40
7934	7873,51	60,49	0,76
7683	7446,38	236,62	3,08
7569	7946,52	-377,52	-4,99
7888	7581,73	306,27	3,88
7709	7852,43	-143,43	-1,86
7697	7553,23	143,77	1,87
7612	7587,06	24,94	0,33
7579	7457,05	121,95	1,61
7596	7490,88	105,12	1,38
7766	7547,89	218,11	2,81
7711	7772,00	-61,00	-0,79
7623	7683,24	-60,24	-0,79
7701	7693,91	7,09	0,09
7698	7800,50	-102,50	-1,33
7775	7743,49	31,51	0,41
7767	7738,16	28,84	0,37
7540	7816,50	-276,50	-3,67
7899	7681,15	217,85	2,76
7680	7873,51	-193,51	-2,52
7649	7845,01	-196,01	-2,56
7654	8113,36	-459,36	-6,00
7684	7524,72	159,28	2,07
7665	7558,56	106,44	1,39
7758	7772,00	-14,00	-0,18

Target	Prediction	Residual	Differential
7895	7412,54	482,46	6,11
7985	7602,81	382,19	4,79
7793	7717,08	75,92	0,97
7766	7568,97	197,03	2,54
7800	7474,88	325,12	4,17
7959	7845,01	113,99	1,43
7783	7750,92	32,08	0,41
7769	7587,06	181,94	2,34
7888	7485,55	402,45	5,10
7889	7896,68	-7,68	-0,10
7597	7772,00	-175,00	-2,30
7800	7884,18	-84,18	-1,08
7679	7813,26	-134,26	-1,75
7886	7839,67	46,33	0,59
7644	7800,50	-156,50	-2,05
7587	7873,51	-286,51	-3,78
7644	7704,32	-60,32	-0,79
7587	7743,49	-156,49	-2,06
7612	7941,19	-329,19	-4,32
7765	7683,24	81,76	1,05
7462	8339,53	-877,53	-11,76
7691	7553,23	137,77	1,79
7767	7779,42	-12,42	-0,16
7692	7852,43	-160,43	-2,09
7683	7772,00	-89,00	-1,16
7559	7654,74	-95,74	-1,27
7754	7654,74	99,26	1,28
7784	7902,01	-118,01	-1,52
7765	7704,32	60,68	0,78
7785	7736,09	48,91	0,63
7889	7800,50	88,50	1,12
7980	7711,75	268,25	3,36
7770	7553,23	216,77	2,79
7770	7654,74	115,26	1,48
7770	7902,01	-132,01	-1,70
7350	7626,24	-276,24	-3,76
7700	7446,38	253,62	3,29
7888	2007,93	5880,07	74,54
7889	2007,93	5881,07	74,55
7597	7108,28	488,72	6,43
7800	2007,93	5792,07	74,26

Target	Prediction	Residual	Differential
7679	3414,60	4264,40	55,53
7800	2226,70	5573,30	71,45
7890	2007,93	5882,07	74,55
7886	2664,24	5221,76	66,22
7644	2007,93	5636,07	73,73
7587	7553,23	33,77	0,45
7800	7480,21	319,79	4,10
7765	7649,41	115,59	1,49
7462	7485,55	-23,55	-0,32
7691	7446,38	244,62	3,18
7767	7720,09	46,91	0,60
7640	7514,05	125,95	1,65
7692	7277,19	414,81	5,39
7683	7514,05	168,95	2,20
7559	7460,05	98,95	1,31
7759	7800,50	-41,50	-0,53
7784	7243,35	540,65	6,95
7765	7717,08	47,92	0,62
7558	7412,54	145,46	1,92
7784	7644,07	139,93	1,80
7985	7373,37	611,63	7,66
7793	7581,73	211,27	2,71
7766	7793,10	-27,10	-0,35
7899	7542,56	356,44	4,51
7527	7451,71	75,29	1,00
7769	7389,37	379,63	4,89
7768	7457,05	310,95	4,00
7629	7462,38	166,62	2,18
7699	7428,54	270,46	3,51
7788	7660,07	127,93	1,64
7688	7485,55	202,45	2,63
7886	7490,88	395,12	5,01
7644	7316,36	327,64	4,29
7587	7553,23	33,77	0,45
7755	7436,88	318,12	4,10
7765	7675,82	89,18	1,15
7462	7316,36	145,64	1,95
7691	7230,85	460,15	5,98
7767	7972,67	-205,67	-2,65
7640	7254,02	385,98	5,05
7692	7344,86	347,14	4,51

Target	Prediction	Residual	Differential
7683	7389,37	293,63	3,82
7559	7360,87	198,13	2,62
7754	7727,75	26,25	0,34
7784	7428,54	355,46	4,57
7765	7378,70	386,30	4,97
7558	7446,38	111,62	1,48
7784	7481,97	302,03	3,88
7664	7514,05	149,95	1,96
7675	7524,72	150,28	1,96
7725	7514,05	210,95	2,73
7654	7428,54	225,46	2,95
7568	7514,05	53,95	0,71
7756	7634,58	121,42	1,57
7677	7373,37	303,63	3,96
7800	7287,86	512,14	6,57
7645	8139,77	-494,77	-6,47
7758	7524,72	233,28	3,01
7668	7428,54	239,46	3,12
7532	7496,22	35,78	0,48
7798	7576,40	221,60	2,84
7746	7761,01	-15,01	-0,19
7652	7610,23	41,77	0,55
7776	7389,37	386,63	4,97
7578	7902,01	-324,01	-4,28
7795	7574,30	220,70	2,83
7769	8261,18	-492,18	-6,34
7768	7384,04	383,96	4,94
7642	7553,23	88,77	1,16
7689	8116,60	-427,60	-5,56
7550	7679,08	-129,08	-1,71
7763	7660,07	102,93	1,33
7550	7766,92	-216,92	-2,87
7621	7806,10	-185,10	-2,43
7569	7806,10	-237,10	-3,13
7720	7496,22	223,78	2,90
7653	7722,42	-69,42	-0,91
7725	7569,23	155,77	2,02
7653	7654,74	-1,74	-0,02
7325	7553,23	-228,23	-3,12
7710	8131,48	-421,48	-5,47
7767	7457,05	309,95	3,99

Target	Prediction	Residual	Differential
7722	7665,41	56,59	0,73
7741	7626,24	114,76	1,48
7665	7784,76	-119,76	-1,56
7631	7597,73	33,27	0,44
7641	7784,76	-143,76	-1,88
7677	7946,52	-269,52	-3,51
7724	7514,05	209,95	2,72
7794	7985,69	-191,69	-2,46
7631	7941,19	-310,19	-4,06
7634	7665,41	-31,41	-0,41
7689	7558,56	130,44	1,70
7613	7816,50	-203,50	-2,67
7712	7524,72	187,28	2,43
7729	7905,28	-176,28	-2,28
7751	7542,56	208,44	2,69
7648	7467,71	180,29	2,36
7658	7779,42	-121,42	-1,59
7725	7693,91	31,09	0,40
7678	7829,00	-151,00	-1,97
7656	7553,23	102,77	1,34
7721	7417,87	303,13	3,93
7652	7813,26	-161,26	-2,11
7623	7693,65	-70,65	-0,93
7751	7620,90	130,10	1,68
7689	7683,24	5,76	0,07
7569	7683,24	-114,24	-1,51
7611	7542,56	68,44	0,90
7658	7400,04	257,96	3,37
7620	7683,24	-63,24	-0,83
7689	7490,88	198,12	2,58
7620	7451,71	168,29	2,21
7620	7490,88	129,12	1,69
7800	7597,47	202,53	2,60
7980	8558,33	-578,33	-7,25
7880	7860,78	19,22	0,24
7660	7626,24	33,76	0,44
7620	7553,23	66,77	0,88
7620	7428,54	191,46	2,51
7580	7394,70	185,30	2,44
7550	7485,55	64,45	0,85
7659	7581,73	77,27	1,01

Target	Prediction	Residual	Differential
7674	7439,21	234,79	3,06
7564	7711,75	-147,75	-1,95
7621	7457,05	163,95	2,15
7543	7514,05	28,95	0,38
7691	7496,22	194,78	2,53
7613	7344,86	268,14	3,52
7695	7960,46	-265,46	-3,45

ATTACHMENT F

Ridge Regression Values

Table 5.2 Ridge regression values difference

Target	Prediction	Residual	Difference (%)
7665	7493,47	171,53	2,24
7700	7766,35	-66,35	-0,86
7784	7519,43	264,57	3,40
7456	7987,61	-531,61	-7,13
7774	7506,04	267,96	3,45
7680	7584,43	95,57	1,24
7360	8674,16	-1314,16	-17,86
7800	6955,77	844,23	10,82
7888	8518,36	-630,36	-7,99
7899	7169,79	729,21	9,23
7889	8293,39	-404,39	-5,13
7619	6888,85	730,15	9,58
7700	8226,47	-526,47	-6,84
7784	7102,86	681,14	8,75
7456	7116,24	339,76	4,56
7874	7343,64	530,36	6,74
7680	8293,39	-613,39	-7,99
7680	7052,06	627,94	8,18
7680	8408,69	-728,69	-9,49
7680	7156,40	523,60	6,82
7680	8065,99	-385,99	-5,03
7680	7378,48	301,52	3,93
7693	8020,51	-327,51	-4,26
7680	7466,70	213,30	2,78
7580	8320,16	-740,16	-9,76
7780	7506,85	273,15	3,51
7680	7870,69	-190,69	-2,48
7780	7493,47	286,53	3,68
7680	7760,21	-80,21	-1,04
7580	7156,40	423,60	5,59
7580	7774,41	-194,41	-2,56
7780	7715,54	64,46	0,83
7580	7819,08	-239,08	-3,15
7773	7493,47	279,53	3,60
7680	7975,04	-295,04	-3,84
7687	7565,72	121,28	1,58
7709	7779,73	-70,73	-0,92
7600	7533,62	66,38	0,87
7887	7557,66	329,34	4,18
7604	7530,89	73,11	0,96
7700	8386,43	-686,43	-8,91

Target	Prediction	Residual	Difference (%)
7660	7694,10	-34,10	-0,45
7639	8263,38	-624,38	-8,17
7630	7343,64	286,36	3,75
7600	8079,38	-479,38	-6,31
7680	7498,79	181,21	2,36
7680	7670,06	9,94	0,13
7780	7975,04	-195,04	-2,51
7621	7584,43	36,57	0,48
7533	7687,96	-154,96	-2,06
7794	7790,68	3,32	0,04
7675	8126,94	-451,94	-5,89
7755	7836,16	-81,16	-1,05
7620	7869,88	-249,88	-3,28
7534	7466,70	67,30	0,89
7765	7811,01	-46,01	-0,59
7677	7065,14	611,86	7,97
7799	7428,16	370,84	4,75
7799	7597,81	201,19	2,58
7684	7200,76	483,24	6,29
7667	7065,14	601,86	7,85
7729	7292,53	436,47	5,65
7621	6883,22	737,78	9,68
7855	7156,10	698,90	8,90
7630	6974,18	655,82	8,60
7743	6837,74	905,26	11,69
7757	7873,58	-116,58	-1,50
7619	7519,43	99,57	1,31
7956	7939,69	16,31	0,20
7778	7292,84	485,16	6,24
7697	7480,08	216,92	2,82
7659	7533,62	125,38	1,64
7621	7670,06	-49,06	-0,64
7684	7778,92	-94,92	-1,24
7582	7964,54	-382,54	-5,05
7781	7520,24	260,76	3,35
7967	7533,62	433,38	5,44
7967	7156,10	810,90	10,18
7622	6928,70	693,30	9,10
7500	7805,69	-305,69	-4,08
7898	7465,89	432,11	5,47
7587	7747,64	-160,64	-2,12
7890	7161,72	728,28	9,23
7901	7429,28	471,72	5,97

Target	Prediction	Residual	Difference (%)
7513	7643,29	-130,29	-1,73
7540	7584,98	-44,98	-0,60
7899	7474,76	424,24	5,37
7690	7578,29	111,71	1,45
7455	7538,95	-83,95	-1,13
7534	7741,50	-207,50	-2,75
7611	7573,78	37,22	0,49
7563	7415,89	147,11	1,95
7605	7485,41	119,59	1,57
7687	7520,24	166,76	2,17
7504	7525,56	-21,56	-0,29
7621	7533,62	87,38	1,15
7661	7565,72	95,28	1,24
7775	7589,75	185,25	2,38
7569	7611,20	-42,20	-0,56
7621	8292,58	-671,58	-8,81
7684	7942,94	-258,94	-3,37
7582	7338,01	243,99	3,22
7783	7988,42	-205,42	-2,64
7997	8395,46	-398,46	-4,98
7593	2198,00	5395,00	71,05
7536	8338,06	-802,06	-10,64
7887	7948,26	-61,26	-0,78
7566	7934,88	-368,88	-4,88
7891	7869,88	21,12	0,27
7924	7760,21	163,79	2,07
7513	7766,35	-253,35	-3,37
7541	7480,08	60,92	0,81
7895	7643,29	251,71	3,19
7696	7733,44	-37,44	-0,49
7451	7720,05	-269,05	-3,61
7545	7734,25	-189,25	-2,51
7615	7960,84	-345,84	-4,54
7559	8097,28	-538,28	-7,12
7662	7533,62	128,38	1,68
7669	7851,17	-182,17	-2,38
7566	7429,28	136,72	1,81
7621	7655,87	-34,87	-0,46
7661	7597,81	63,19	0,82
7775	8137,43	-362,43	-4,66
7569	7747,64	-178,64	-2,36
7422	7642,48	-220,48	-2,97
7880	7884,08	-4,08	-0,05

Target	Prediction	Residual	Difference (%)
7680	7720,87	-40,87	-0,53
7637	8052,61	-415,61	-5,44
7594	7655,87	-61,87	-0,81
7684	7734,25	-50,25	-0,65
7582	7974,22	-392,22	-5,17
7783	7838,60	-55,60	-0,71
7783	7987,61	-204,61	-2,63
7997	7851,98	145,02	1,81
7593	7778,92	-185,92	-2,45
7890	7838,60	51,40	0,65
7887	7760,21	126,79	1,61
7566	7869,88	-303,88	-4,02
7894	7642,48	251,52	3,19
7934	7869,88	64,12	0,81
7683	7429,28	253,72	3,30
7569	7992,93	-423,93	-5,60
7888	7611,20	276,80	3,51
7709	7975,04	-266,04	-3,45
7697	7597,81	99,19	1,29
7612	7643,29	-31,29	-0,41
7579	7493,47	85,53	1,13
7596	7538,95	57,05	0,75
7766	7565,72	200,28	2,58
7711	7733,44	-22,44	-0,29
7623	7747,64	-124,64	-1,64
7701	7811,83	-110,83	-1,44
7698	7746,83	-48,83	-0,63
7775	7720,05	54,95	0,71
7767	7687,96	79,04	1,02
7540	7843,11	-303,11	-4,02
7899	7661,19	237,81	3,01
7680	7869,88	-189,88	-2,47
7649	7856,49	-207,49	-2,71
7654	7900,35	-246,35	-3,22
7684	7584,43	99,57	1,30
7665	7629,91	35,09	0,46
7758	7733,44	24,56	0,32
7895	7383,80	511,20	6,47
7985	7506,04	478,96	6,00
7793	7793,12	-0,12	0,00
7766	7460,56	305,44	3,93
7800	7442,66	357,34	4,58
7959	7856,49	102,51	1,29

Target	Prediction	Residual	Difference (%)
7783	7838,60	-55,60	-0,71
7769	7643,29	125,71	1,62
7888	7506,85	381,15	4,83
7889	7851,17	37,83	0,48
7597	7733,44	-136,44	-1,80
7800	7934,07	-134,07	-1,72
7679	7897,46	-218,46	-2,84
7886	7824,40	61,60	0,78
7644	7746,83	-102,83	-1,35
7587	7869,88	-282,88	-3,73
7644	7642,48	1,52	0,02
7587	7720,05	-133,05	-1,75
7612	7960,84	-348,84	-4,58
7765	7747,64	17,36	0,22
7462	7777,45	-315,45	-4,23
7691	7597,81	93,19	1,21
7767	7851,98	-84,98	-1,09
7692	7975,04	-283,04	-3,68
7683	7733,44	-50,44	-0,66
7559	7734,25	-175,25	-2,32
7754	7734,25	19,75	0,25
7784	7883,26	-99,26	-1,28
7765	7642,48	122,52	1,58
7785	7978,58	-193,58	-2,49
7889	7746,83	142,17	1,80
7980	7761,02	218,98	2,74
7770	7597,81	172,19	2,22
7770	7734,25	35,75	0,46
7770	7883,26	-113,26	-1,46
7350	7720,87	-370,87	-5,05
7700	7429,28	270,72	3,52
7888	1869,34	6018,66	76,30
7889	1869,34	6019,66	76,30
7597	7585,08	11,92	0,16
7800	1869,34	5930,66	76,03
7679	2595,70	5083,30	66,20
7800	2004,97	5795,03	74,30
7890	1869,34	6020,66	76,31
7886	2276,23	5609,77	71,14
7644	1869,34	5774,66	75,54
7587	7597,81	-10,81	-0,14
7800	7474,76	325,24	4,17
7765	7702,16	62,84	0,81

Target	Prediction	Residual	Difference (%)
7462	7506,85	-44,85	-0,60
7691	7429,28	261,72	3,40
7767	7882,30	-115,30	-1,48
7640	7520,24	119,76	1,57
7692	7201,88	490,12	6,37
7683	7520,24	162,76	2,12
7559	7582,65	-23,65	-0,31
7759	7746,83	12,17	0,16
7784	7156,40	627,60	8,06
7765	7793,12	-28,12	-0,36
7558	7383,80	174,20	2,30
7784	7670,06	113,94	1,46
7985	7306,23	678,77	8,50
7793	7611,20	181,80	2,33
7766	8005,35	-239,35	-3,08
7899	7533,62	365,38	4,63
7527	7461,37	65,63	0,87
7769	7402,51	366,49	4,72
7768	7493,47	274,53	3,53
7629	7525,56	103,44	1,36
7699	7480,08	218,92	2,84
7788	7766,35	21,65	0,28
7688	7506,85	181,15	2,36
7886	7538,95	347,05	4,40
7644	7279,45	364,55	4,77
7587	7597,81	-10,81	-0,14
7755	7601,36	153,64	1,98
7765	7629,10	135,90	1,75
7462	7279,45	182,55	2,45
7691	7239,30	451,70	5,87
7767	7686,34	80,66	1,04
7640	7220,59	419,41	5,49
7692	7292,84	399,16	5,19
7683	7402,51	280,49	3,65
7559	7389,12	169,88	2,25
7754	7857,31	-103,31	-1,33
7784	7480,08	303,92	3,90
7765	7338,32	426,68	5,49
7558	7429,28	128,72	1,70
7784	7564,35	219,65	2,82
7664	7520,24	143,76	1,88
7675	7584,43	90,57	1,18
7725	7520,24	204,76	2,65

Target	Prediction	Residual	Difference (%)
7654	7480,08	173,92	2,27
7568	7520,24	47,76	0,63
7756	7842,14	-86,14	-1,11
7677	7306,23	370,77	4,83
7800	7266,07	533,93	6,85
7645	7827,29	-182,29	-2,38
7758	7584,43	173,57	2,24
7668	7480,08	187,92	2,45
7532	7571,04	-39,04	-0,52
7798	7579,10	218,90	2,81
7746	8049,46	-303,46	-3,92
7652	7624,58	27,42	0,36
7776	7402,51	373,49	4,80
7578	7883,26	-305,26	-4,03
7795	7492,66	302,34	3,88
7769	7622,31	146,69	1,89
7768	7370,41	397,59	5,12
7642	7597,81	44,19	0,58
7689	7846,00	-157,00	-2,04
7550	7951,81	-401,81	-5,32
7763	7766,35	-3,35	-0,04
7550	7934,88	-384,88	-5,10
7621	8012,45	-391,45	-5,14
7569	8012,45	-443,45	-5,86
7720	7571,04	148,96	1,93
7653	7825,21	-172,21	-2,25
7725	7694,10	30,90	0,40
7653	7734,25	-81,25	-1,06
7325	7597,81	-272,81	-3,72
7710	8460,15	-750,15	-9,73
7767	7493,47	273,53	3,52
7722	7798,44	-76,44	-0,99
7741	7720,87	20,13	0,26
7665	7884,08	-219,08	-2,86
7631	7707,48	-76,48	-1,00
7641	7884,08	-243,08	-3,18
7677	7992,93	-315,93	-4,12
7724	7520,24	203,76	2,64
7794	8070,51	-276,51	-3,55
7631	7960,84	-329,84	-4,32
7634	7798,44	-164,44	-2,15
7689	7629,91	59,09	0,77
7613	7843,11	-230,11	-3,02

Target	Prediction	Residual	Difference (%)
7712	7584,43	127,57	1,65
7729	8205,98	-476,98	-6,17
7751	7533,62	217,38	2,80
7648	7557,66	90,34	1,18
7658	7851,98	-193,98	-2,53
7725	7811,83	-86,83	-1,12
7678	7760,21	-82,21	-1,07
7656	7597,81	58,19	0,76
7721	7415,89	305,11	3,95
7652	7897,46	-245,46	-3,21
7623	7578,29	44,71	0,59
7751	7688,77	62,23	0,80
7689	7747,64	-58,64	-0,76
7569	7747,64	-178,64	-2,36
7611	7533,62	77,38	1,02
7658	7466,70	191,30	2,50
7620	7747,64	-127,64	-1,68
7689	7538,95	150,05	1,95
7620	7461,37	158,63	2,08
7620	7538,95	81,05	1,06
7800	7473,95	326,05	4,18
7980	8290,15	-310,15	-3,89
7880	8096,31	-216,31	-2,75
7660	7720,87	-60,87	-0,79
7620	7597,81	22,19	0,29
7620	7480,08	139,92	1,84
7580	7434,60	145,40	1,92
7550	7506,85	43,15	0,57
7659	7611,20	47,80	0,62
7674	7544,27	129,73	1,69
7564	7761,02	-197,02	-2,60
7621	7493,47	127,53	1,67
7543	7520,24	22,76	0,30
7691	7571,04	119,96	1,56
7613	7292,84	320,16	4,21
7695	8379,83	-684,83	-8,90

“This page is intentionally left blank”

AUTHOR BIOGRAPHY



James Janto Junior, born in Jakarta on July 14, 2001, Oldest son of the 2 brothers. He completed his formal education at Canisius High School from 2017 to 2020. During the writing of this bachelor thesis, he pursued an undergraduate degree in Marine Engineering at Institut Teknologi Sepuluh Nopember and Hochschule Wismar as part of a double degree program. Throughout his academic journey, he acquired both academic and non-academic experiences related to his field of study and beyond. His notable academic experiences include an internship at PT DKB II Indonesia and another at PT PAL Surabaya as an Engineering Intern.