Web Based VDR Application Development for Oil and Gas Data Visualization

EXPLORING PREDICTIVE MODELS AND DEPLOYING IT AS A BACKEND SERVICE

THESIS

PROJECT

by

Kotrakona Harinatha Sreeya Reddy 2201816165

BINUS INTERNATIONAL
BINUS UNIVERSITY
JAKARTA
2022

Web Based VDR Application Development for Oil and Gas Data Visualization

EXPLORING PREDICTIVE MODELS AND DEPLOYING IT AS A BACKEND SERVICE

THESIS

Proposed as a requirement for obtaining
Sarjana degree at
Program Computer Science
Education Level Strata-1 (Sarjana/Bachelor)

by

Kotrakona Harinatha Sreeya Reddy 2201816165

BINUS INTERNATIONAL
BINUS UNIVERSITY
JAKARTA
2022

EXPLORING PREDICTIVE MODELS AND DEPLOYING IT AS A BACKEND SERVICE

THESIS

Prepared By :

Kotrakona Harinatha Sreeya Reddy

2201816165

Approved by:

Supervisor	**Co-Supervisor**
Ida Bagus Kerthyayana Manuaba, S.T., Ph.D.	**Ardimas Andi Purwita, S.T., M.T., Ph.D.**

BINUS UNIVERSITY

Jakarta

2022

BINUSINTERNATIONAL

FM-BINUS-AA-FPU-442/R1

People
Innovation
Excellence

STATEMENT FROM THE BOARD OF EXAMINERS

We, the members of the Board of Examiners for the S-1 Thesis Defense,
Hereby declare that

KOTRAKONA HARINATHA SREEYA REDDY (2201816165)

Who presented an S-1 Thesis entitled

EXPLORING PREDICTIVE MODELS AND DEPLOYING IT AS A BACKEND SERVICE

Successfully passed the S-1 Thesis Defense Examination conducted on 2022

		Name	Signature
1.	Chair	Nunung Nurul Qomariyah, S.Kom., M.T.I., Ph.D	.....
2.	Member	Jude Joseph Lamug Martinez, MCS	.....
3.	Supervisor	Ida Bagus Kerthyayana Manuaba, S.T., Ph.D	.....

PERNYATAAN
STATEMENT

Dengan ini, saya,

With this, I,

Nama (*Name*): Kotrakona Harinatha Sreeya Reddy

NIM (*Student ID*): 2201816165

Judul Tesis (*Thesis Title*): Exploring Predictive Models and Deploying it as a Backend Service

Memberikan kepada Universitas Bina Nusantara hak non-eksklusif untuk menyimpan, memperbanyak, dan menyebarluaskan tesis saya, secara keseluruhan atau hanya sebagian atau hanyaringkasannya saja, dalam bentuk format tercetak atau elektronik.
Hereby grant to my school, Bina Nusantara University, the non-exclusive right to archive, reproduce, and distribute my/our thesis, in whole or in part, whether in the form of a printed or electronic format.

Menyatakan bahwa saya, akan mempertahankan hak eks saya, untuk menggunakan seluruh atausebagian isi tesis saya/kami, guna pengembangkan karya di masa depan, misalnya dalam bentukartikel, buku, perangkat lunak, ataupun sistem informasi.
I acknowledge that I retain exclusive rights of my thesis by using all or part of it in future work oroutput, such as an article, a book, software, or information system.

Catatan: Pernyataan ini dibuat dalam 2 (dua) bahasa, Indonesia dan Inggris, dan apabila terdapatperbedaan penafsiran, maka yang berlaku adalah versi Bahasa Indonesia.
Note: This Statement is made in 2 (two) languages, Indonesian and English, and in the case of adifferent interpretation, the Indonesian version shall prevail.

Jakarta,

Kotrakona Harinatha Sreeya Reddy
2201816165

BINUS INTERNATIONAL

BINUS UNIVERSITY

Major Computer Science
Bachelor of Science Computer Thesis
Semester 8 - Year 2022

EXPLORING PREDICTIVE MODELS AND DEPLOYING IT AS A BACKEND SERVICE

Kotrakona Harinatha Sreeya Reddy 2201816165

ABSTRACT

There is a high demand for oil and gas as the oil and gas industry contributes significantly to the growth of an economy. Oil and gas companies contains lots of data, thus they often make use of dashboard-based software applications to help them manage it, e.g., Virtual Data Room (VDR) application. The oil and gas industry faces several problems, e.g., identifying wells that contain abundant oil and gas. The wells that do not contain much oil and gas end up abandoned and the company's time and money would have been wasted. Therefore, this thesis will focus on developing an oil and gas predictive model and deploying it as part of the backend service. It will also focus on developing an API for the communication between the client-side application made by the author's team and the oil and gas prediction service. The predictive model would help oil and gas company drill in wells where there is large amounts of oil and gas which would save the company's time and money. This thesis will conduct a comparative study on the performance of random forest and gradient boosting models in predicting oil and gas production values. The models will be tested on several hyperparameters to determine the best performing model. This thesis will also investigate the effects of data imputation on model performance. As the missing values in the dataset reaches 42%, this method will explore a different method of data imputation, namely self-supervised imputation. The results show that in terms of oil production, based on the Root Mean Square Error (RMSE) value, the gradient boosting model with the self-supervised imputation dataset performed 23% better than the poorest performing model. In terms of gas production, the RMSE values show that the gradient boosting model with the self-supervised imputation dataset performed 15% better than the poorest performing model. The best performing model is then deployed as part of the backend service and an API is developed to allow the client-side application to communicate to the backend service. The VDR website application created by the author's teammates would connect to the API endpoints to obtain the oil and gas production values.

Keywords

Oil, gas, gradient boosting, random forest, self-supervised imputation, hyperparameter optimization, API

ACKNOWLEDGEMENT

This thesis is developed by the author and the author's team to complete their undergraduate study in Binus International University. The development of this thesis would not have been possible without the help of those who supported the author throughout this journey.

The author would like to express their gratitude to those have helped and supported the author:

- To the author's family who provided the author with love and support,
- To the author's supervisors, Sir Ida Bagus Kerthyayana Manuba and Sir Ardimas Andi Purwita for their constant guidance and meaningful advice,
- To the author's team members and best friends, Chan Elizabeth and Vicky Vanessa, who have worked together with the author since the start of this thesis and stayed together through it all,
- To the author's close confidant, La Myra Bening, who has been there since day 1 and provided never-ending emotional support, and,
- To all Computer Science lecturers in Binus International University who have taught the author with patience

Table of Content

List of Figures

List of Tables

CHAPTER 1
INTRODUCTION

This chapter introduces the project the author worked on alongside the author's team. It also includes the background of the project as well as the scope, objective, and aims of carrying out this project. It will also describe the structure and provide insights into the remaining chapters.

1.1 Background

There is no doubt that oil and gas are vital elements to the growth of the economy. There have been traces of oil trade ever since 1875 BC [1]. In this modern, technologically-advanced society, the demand for oil and gas has only continued to grow stronger. It is used for many modern inventions enjoyed by a vast majority of people, such as vehicles, fuels, medical equipment, agriculture, and many more [2]. Additionally, the oil and gas industry has also provided jobs to thousands of individuals [3].

There are many oil and gas reserves in different corners of the world. In Indonesia, in particular, the Energy Ministry has recorded that in January 2021, there is a total reserve of 2.44 billion barrels of oil and 43.6 trillion cubic feet of gas [4]. However, due to the rapidly increasing population and a growing economy, the demand for oil and gas in Indonesia is rising [5]. Furthermore, 50% of Indonesia's energy is derived from oil [5]. This reliance on oil results in Indonesia importing nearly 350,000 barrels per day (BPD) and 50,000 barrels of fuel per day from other countries [5].

Oil and gas have many uses and have a substantial impact on the economy of a country. Therefore, oil and gas industries often make use of dashboard-based software applications in order to help them manage it, such as a Virtual Data Room (VDR) application. A VDR is an online repository that can store data securely and can be accessed by multiple users simultaneously [6]. These kinds of applications can help the oil and gas industries discover which areas could have more oil and gas. It can also help clients visualize the oil and gas data. Lynx and INTViewer are examples of software applications capable of data visualization [7] [8]. These applications are similar, yet they also have their differences. Lynx offers petroleum data services and geophysical and Geographical Information System (GIS) services [7]. It offers 2D and 3D seismic viewers and costs at least £250 per user per year [7]. On the other hand, INTViewer is a platform that allows users to check seismic data, geospatial integrity, and also process datasets [8]. It can cost up to $4,000/person a year [9]. These types of applications can benefit the oil and gas industries greatly. Therefore, the goal of this group project will be to develop a VDR application for the oil and gas industries.

The author's team will be developing a VDR application with features similar to Lynx and INTViewer based on the request of the author's client, PT Geodwipa Teknika Nusantara [10]. The client's request for the product can be seen in Appendix B Figure B.1. The product owner of this application, Mr. Ardimas Andi Purwita, states that there are some oil and gas companies in Indonesia that want a custom VDR application software similar to Lynx and INTViewer [11]. The author's client is one of those companies that want this type of software as they deal with lots of oil and gas data [12].

The oil and gas industry contains lots of data, data that can be used to obtain more information. This is where data science comes in. Data science is the method of obtaining meaningful insights, such as patterns, from a large set of data [13]. Data science is useful for the VDR as users will be able to understand the oil and gas data, thus gaining meaningful insights from it. In a more specific sense, the author's client wishes to have a VDR that contains a predictive model that can predict oil and gas production, as shown in Appendix Figure B.1. This thesis will focus on:

- developing an oil and gas predictive model,
- building an oil and gas production data prediction application as a backend service,
- integrating the oil and gas prediction service with other backend services provided by other developers, and
- creating an API for the backend prediction application for communication between the frontend and the backend prediction service.

The scope of the author's role and responsibility is detailed more in Section 1.2.2.

1.2 Scope

This section describes the scope of the author's group as well as the author's individual scope for this project.

1.2.1 Group Scope

In this project, the author and the author's team had different responsibilities, as shown in Table 1.1. The main goal of the project is to develop a VDR application that will benefit the oil and gas industry. As has been mentioned in Section 1.1, there are

existing applications for this purpose; however, the author's client wishes to have VDR application with customized features. Therefore, the author and the author's team will make use of open-source libraries and hand-pick essential features based on the request of the customer. Additionally, the author and the author's team will also develop custom features requested by the customers. The application will consist of several features, such as:

- visualization of oil and gas data,
- map application along with the showcase feature,
- prediction model to predict the oil and gas production, and
- file management to store the user's file.

Table 1.1: Scope of Activities

Name	Role
Kotrakona Harinatha Sreeya Reddy	- Evaluating different machine learning algorithms on the oil and gas dataset to find the best predictive model - Developing oil and gas production data prediction application as a backend service for the VDR website application, where communication between the frontend and the oil and gas prediction service is performed over an API - Act as Scrum Team

Elizabeth Chan	- Design the front-end of pages that implement GIS inside the website application - Develop the front-end of the page for oil and gas visualization that integrated GIS - Develop a custom showcase of data on the map showcase page that can easily understand by non-technical user - Design and develop the form pages where users need to input the data for visualization and showcase - Act as Scrum Team
Vicky Vanessa	- Developing the frontend of the website application - Visualizing the data of oil and gas - Act as Scrum Master

1.2.2 Individual Scope of Work

The author's responsibility was to :

- create a predictive model capable of predicting oil and gas production,
- conduct a comparative study between different machine learning algorithms to discover the best predictive model,
- build an oil and gas production data prediction application as a backend service,
- integrating this service with the existing backend services, and

- creating an API for the backend prediction application for communication between the frontend and the backend prediction service.

1.3 Aim and Benefits

This section will cover the aim and benefits of this project.

1.3.1 Aims

This subsection will cover both the group aim and the individual aim for this project.

1.3.1.1 Group Aim

The aim of the group is to develop a VDR website application with customized features based on the request of the author's client as shown in Appendix B Figure B.1. These features include:

- visualization of oil and gas data,
- map application along with the showcase feature,
- prediction model to predict the oil and gas production, and
- file management to store the user's file.

1.3.1.2 Individual Aim

The aims of the author for this project can be broken down into these bullet points :

- provide an oil and gas prediction feature in the VDR Website application,
- determine the best oil and gas prediction model by conducting a comparative study on machine learning algorithms,
- build an oil and gas production data prediction application as a backend service,
- integrate the prediction service with the existing backend services, and

- create an API for the backend oil and gas production data prediction application for communication between the frontend and the backend prediction service.

1.3.2 Benefits

This section will discuss both the main benefits of the VDR website application as well as the benefits of the author's aim.

1.3.2.1 Main Benefits

The main benefits of developing this VDR website application are:

- integrating the VDR website application with petrotechnical solutions such as:
    - visualizing oil and gas data to makes it easy for users to understand the data,
    - a map application along with the showcase feature helps the client see the data they have for a certain reserve, and
    - a prediction model to predict the oil and gas production helps the clients see the wells that contain more oil and gas.

1.3.2.2 Benefits of Personal Aim

As mentioned in Section 1.3.1.2, the author has several aims to accomplish in this project. The benefits of these aims are:

- the oil and gas prediction feature is beneficial as it would help the client see which areas contain more oil and gas, thus making it possible to focus only on the wells which contain more oil and gas,
- comparing the different types of prediction machine learning algorithms would help the author discover the best model and,

- using an API would make it possible for the client-side application to communicate with the backend oil and gas prediction service to display the oil and gas production values.

1.4 Structure

This thesis consists of seven chapters which will be briefly described in this section.

1.4.1 Chapter 1: Introduction

Chapter 1 introduces the author's topic, the scope, objectives, aims, vision, and mission of this project.

1.4.2 Chapter 2: Theoretical Foundation

Chapter 2 describes the fundamental theories behind what the author uses to develop the project. It defines specific terms and provides further insights into the problem.

1.4.3 Chapter 3: Problem Analysis

Chapter 3 will detail the problem even further and describe the works related to the author's project. It will briefly describe the model the author intends to train and outline how the author intends to develop the backend prediction application.

1.4.4 Chapter 4: Solution Design

Chapter 4 focuses on the design of the solution devised by the author; it includes data pre-processing as well as how the models will be evaluated. It will also discuss how the models will be saved and also discuss the API endpoints for the backend prediction application the author intends to develop.

1.4.5 Chapter 5: Implementation

Chapter 5 will center on the results obtained from model training and evaluation. It will also analyse the information obtained from the results. It will also show the endpoints created by the author.

1.4.6 Chapter 6: Discussion

Chapter 6 will describe the key results observed in the thesis and analyse them further while relating them to the author's aim.

1.4.7 Chapter 7: Conclusion and Recommendations

Chapter 7 will conclude all the important results and observations obtained; it will also discuss recommendations for possible future works

CHAPTER 2

THEORETICAL FOUNDATION

This chapter will delve into the theories and techniques the author used while developing this project. It will probe into how oil and gas are produced in the reservoir. Additionally, it will discuss how the author intends to build the model to predict oil and gas production using machine learning. Afterwards, this chapter will discuss missing data, outliers, and feature correlation in the dataset used to train the model. It will also examine how to evaluate the performance of the model and how to develop and package an API. It also discusses how the methodology the author's team will use to develop the project.

2.1 Oil and Gas in a Reservoir

In order to build an oil and gas predictive model, it is vital to understand how oil and gas are formed in a reservoir and the factors that affect its formation.

2.1.1 Oil Formation

A formula that can be taken into account for oil formation is the oil formation volume factor (B_o). It is the ratio of oil volume and dissolved gas at a specific temperature and pressure that is needed to make one barrel of oil [14]. B_o is either greater than or equal to unity [15].

The equation for the oil formation volume factor is :

$$B_o = \frac{(V_o)pT}{(V_o)_{sc}}. \qquad (2.1)$$

In Equation 2.1, B_o is the oil volume factor, V_o is the volume of oil, $(V_o)sc$ is the volume of oil measured under standard conditions, p is the pressure at the reservoir, whereas T is the temperature at the reservoir [14]. From Equation 2.1, it can be inferred that temperature and pressure are essential factors in the formation of oil. Once the oil reaches the surface, it loses the dissolved gas, which leads to changes in the reservoir oil obtained. First of all, the mass of the oil will reduce as it loses the dissolved gas, then the oil will also contract as the temperature decreases on the surface [14]. Afterwards, the oil will again expand as the pressure increases [14]. Often the effect of the temperature and pressure changes when the oil reaches the surface is minimal and will cancel out each other [14].

2.1.2 Gas Formation

A formula that can be taken into account for gas formation is the gas formation volume factor (B_g). It is the ratio of the volume of gas at a specific temperature and pressure that is needed to manufacture one standard volume of gas [16]. This equation for gas formation volume factor can be expressed as :

$$B_g = \frac{V_{p,T}}{V_{sc}} \qquad (2.2)$$

In Equation 2.2, B_g is the gas formation volume, $V_{p,T}$ is the volume of gas at the reservoir pressure and temperature and V_{sc} is the volume of gas at standard conditions.

In real life, gases follow the real gas law, which can be expressed mathematically as :

$$pV = znRT, \qquad (2.3)$$

where p is the pressure, V is the volume, n is the number of moles of gas, R is the universal gas constant, T is the temperature, and z is the gas compressibility factor [17]. Variable z can be expressed as :

$$z = \frac{V_a}{V_i}, \tag{2.4}$$

where V_a is the actual volume of *n*-moles of gas at a certain temperature and pressure, and V_i is the ideal volume of n-moles of gas at the same temperature and pressure [17]. Therefore, the equation for real gas law should be applied to Equation 2.2. Equation 2.3 is applied onto Equation 2.2 by substituting for the volume (V), which will result in Equation 2.5.

$$B_g = \frac{zTP_{sc}}{T_{sc}P}. \tag{2.5}$$

In Equation 2.5, B_g is the gas formation volume, P is the pressure, T is the temperature, P_{sc} is 1 atm, T_s is 60°F, and z is the gas compressibility factor at standard conditions (1.0) [17]. With the assumption that the standard conditions are represented by $P_{sc} = 14.7\ psia$ and $T_{sc} = 520$, Equation 2.5 can be reduced to :

$$B_g = 0.0283\frac{zT}{P}. \tag{2.6}$$

2.1.3 Pressure and Temperature in Oil and Gas Formation

Figure 2.1 shows the phase diagram of oil and gas in a reservoir. As stated previously in Section 2.1.1, when oil is drilled, it also contains dissolved gas. Therefore, in a reservoir, there exist 2 phases, namely liquid and gas. Based on the current pressure and temperature, the phase diagram shows that there is a region where the mixture will be either liquid or gas only and a region where both liquid and gas are at equilibria.

The black line, known as the Bubble Point Line, denotes where both phases begin to appear [18]. Before the bubble point, the only phase that exists is liquid. However, at a constant temperature, as pressure decreases, the total volume of gas increases, whereas the volume of oil decreases [18]. This property is supported by the Le Chatelier's Principle, which states that an increase in volume or decrease in pressure would increase the formation of the gaseous product.

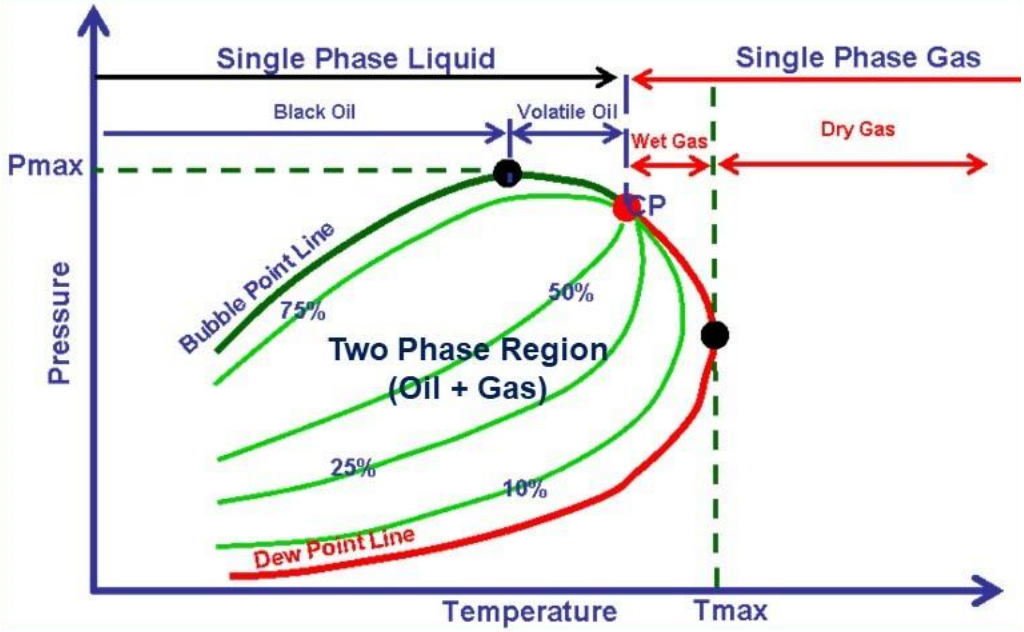


Figure 2.1: Phase Diagram of Oil and Gas *[18]*

As the pressure continues to decrease, more heavier molecules become gaseous, increasing the density and viscosity of the gas [18]. Subsequently, there will be a point where only a small portion of liquid remains; this is called the Dew Point [18]. If the pressure drops below this point, the only phase that exists is gas [18].

2.2 Machine Learning

Upon briefly explaining the oil and gas formation process, this section will delve into machine learning. Machine learning is the central machinery in building a prediction model of the oil and gas production data. It is defined as *the capability of a system to be able to learn from data and algorithms to automate the process of solving certain tasks* [19]. Machine learning is a part of Artificial Intelligence (AI) which centers on using data and algorithms to echo the way humans act and learn [20]. Machine learning helps uncover insights, make classifications, and make predictions from the data given in order to aid users [20]. Machine learning depends on a dataset, which is a collection of data that will be regarded as one unit by the machine [21]. This dataset will act as the "training data" for the machine to learn. It is preferable to have large amounts of data as this means the machines would learn more efficiently and be able to solve problems with better accuracy. However, the quantity of the dataset is not the only significant factor in machine learning; the quality of the dataset is also a notable factor. A machine would perform significantly better with a high-quality dataset in contrast to a poor-quality dataset.

Machine learns in different ways, namely, unsupervised learning, reinforcement learning, and supervised learning. Unsupervised learning aims to derive meaningful information from unlabelled data [22]. It is not as commonly used as supervised learning [22]. On the other hand, reinforcement learning is another complex part of AI where the model is trained to make decisions sequentially [23]. The output is dependent on the state of the current input, and the following input would then be reliant on the output of the previous output [23]. Supervised learning is a part of machine learning and artificial intelligence; it is learning by means of mapping

between a set of input variables and output variables [24]. The input variables are fed into the machine learning model, and after the training phase, it will apply what it learned to unknown data [25]. This type of machine learning is one of the most common methods and is usually used for classification and regression problems [25]. There are several types of supervised machine learning models, namely Naive Bayes, linear regression, support vector machine (SVM), KNN, and others [26]. There is also another new form of supervised machine learning, which is known as self-supervised machine learning. In this method, a machine learning model trains by using a part of the input data to learn another part of the input data [27]. In terms of a prediction model, supervised learning is ideal, especially with limited computational resources.

2.3 Predictive Models

Predictive modelling is a part of machine learning. It is the process of predicting future outcomes from data gathered beforehand. There are different types of predictive model algorithms that can be used to predict values. These algorithms include classification and regression algorithms. A part of machine learning that can be used for predictive modelling is deep learning [28].

2.3.1 Classification Algorithms

Classification is a type of machine learning algorithm that aims to categorize the data into specific class labels or categories [29]. Classification algorithms include K-Nearest Neighbours, Naïve Bayes, and Support Vector Machines [29]. This algorithm can be used for image classification or email spam classification.

2.3.2 Deep Learning

Deep learning consists of multiple layers of algorithms known as an artificial neural network (ANN). An ANN is designed to behave similarly to a human brain. The simplest ANN consists of a single neuron, also known as a perceptron [28]. These neurons will be stacked on top of one another, which will create layers [28]. Each layer will learn something new and pass it on to the next layer that will learn something else. There are different types of ANN, such as recurrent neural network (RNN) and convolution neural network (CNN). CNN is a kind of neural network that works well for image and video data, whereas RNN works well with sequential data [28] [30]. An extension of RNN is a Long and Short Term Memory (LSTM) network designed to handle situations where RNNs might not be sufficient [31].

2.3.3 Regression Algorithms

Regression is another type of machine learning algorithm that aims to discover correlations between the dependent and independent variables and predict the continuous values of the output based on the input [32]. Regression algorithms include Linear Regression, Polynomial Regression, and Decision Tree Regression [29]. This algorithm can be used for production prediction, weather prediction, or house price prediction [29].

In a study [33], a researcher compared the use of deep learning algorithms and tree-based regression machine algorithms on a variety of datasets for prediction. The research showed that deep learning tends to perform better on unstructured data, such as images or voice [33]. On the other hand, tree-based algorithms function better with tabular structured data compared to deep learning [33]. Another study where a

researcher compared tree-based algorithms and deep learning models on a variety of tabular datasets with different learning objectives also gave the same result [34]. Tabular data is a dataset that consists of a set of rows and columns; it is one of the most common types of datasets.

2.3.3.1 Tree-based Algorithms

Tree-based algorithms are a well-known part of machine learning, more specifically, predictive modelling. Tree-based regression algorithms are commonly used for predictive analysis of numerical values [35]. This regression model works by investigating the connection between variables [35]. It will determine the value of one variable based on the other variables present [35].

Random Forest Algorithm

A commonly used algorithm for predictive models is the random forest algorithm [35]. This is a supervised learning algorithm that is based on the ensemble learning method [35]. Ensemble learning is the process of combing the prediction results of several machine learning algorithms [35]. The goal of this is to make the prediction results more accurate. The random forest algorithm combines the predictive results of several decision trees [35]. The respective decision trees do not interfere with one another [35]. There are two steps for the random forest algorithm; the first step is building *n* decision tree regressors, where *n* is the number of decision tree regressors [35]. These trees can be modified by specified hyperparameters, such as the strategy best used to split the node into sub-nodes or the function used to measure the quality of the split [36]. The final step would be to take the average prediction values of the decision tree regressors; this average will serve as the final output of the model [35].

Gradient Boosting Algorithm

Another algorithm for predictive models is the gradient boosting algorithm. This algorithm is based on the concept of boosting [37]. In terms of regression, boosting is a procedure of building strong regressors by combing weak learners [37]. This algorithm has three requirements, namely loss function, weak learners, and additive model.

A loss function would measure how similar the values predicted by the algorithm are to the actual values. In terms of regression problems, the loss function used could be Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Coefficient of Determinant (R^2). [37]. Additionally, this algorithm is based on the idea that combing multiple weak learners would result in an accurate result. The weak learners used in gradient boosting are typically decision trees [37]. Gradient boosting is also an additive model as it adds the weak learners one by one. Every new predictor would gain new knowledge from the error of the previous predictor, and it would work to correct the error, which would result in a better model [37].

2.4 Data Analytics Pipeline

The predictive models have to be trained on a dataset so that they can learn; however, before training, it is vital to understand and clean the dataset used. The steps that can be taken to understand and clean the dataset before model training are shown in Figure 2.2. These steps will be explained further in the upcoming sections.

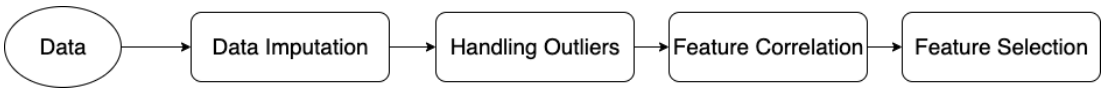


Figure 2.2: Data Analytics Pipeline

2.5 Data Imputation

An essential part of model training is the quality of the dataset. A possible problem in a dataset is missing data. Missing data in a dataset could prove to be problematic as it could affect the model's ability to perform well.

2.5.1 Mechanism of Missingness

There are three possible mechanisms for missing data in a dataset; these mechanisms are Missing Completely at Random (MCAR), Missing at Random (MAR), and Missing Not at Random (MNAR).

In the MCAR mechanism, the missing values are unrelated to the other values in the dataset, both missing and present; therefore, these missing values are random. In this situation, the missing values are considered negligible as they would not significantly impact the model performance [38].

In the MAR mechanism, the missing values are also random such as in MCAR; however, there are possibilities of the data in question being dependent on other values in the dataset. In this situation, the missing values should be considered as they could affect the model's performance. However, the effect is not extreme [38].

In the MNAR mechanism, the missing values are strongly dependent on the other values in the dataset, both missing and present. MNAR is the most serious reason mechanism for missing data as it cannot be ignored and could affect the model's performance [38]. In these cases, it is recommended to validate the data collection process [38].

If the mechanism of missingness is either MNAR or MAR, the effects of the missing values could greatly impact the model's performance. The mechanism can be determined by calculating the correlation of missingness between each feature in a dataset. Correlation is known as a statistical measure that describes how one feature is related to another feature [39]. In this situation, a correlation value close to 1 would indicate that if the value of one feature is empty, then the value of the other feature would be empty as well [40]. A correlation value close to -1 would indicate that if the value of one feature is empty, then the value of the other feature would not be empty [40]. On the other hand, a correlation value close to 0 indicates that there is no or barely any relationship between the features [40].

In order to counteract the effects of the missing values on the dataset, data imputation methods could be used. Data imputation methods include central value imputation and forward filling imputation.

2.5.2 Central Value Imputation

Central value imputation is the process of filling in the missing data in the dataset with their central tendencies [41]. These central tendencies could either be the mean, median, or mode. The mode is typically used to fill in the missing data for categorical variables, whilst the mean and median are often used to fill in for numerical variables [42]. The central tendencies are deemed as reasonable estimates for filling in the missing data. However, this method would not yield ideal results if the missing data follows the MNAR mechanism, and it could also introduce bias in the dataset [43]. Additionally, filling in the missing values with the mean could reduce the variance in the data set [41].

2.5.3 Forward Filling Imputation

Forward filling is the process of filling in the missing data with the value observed before the missing value [44]. For instance, in a dataset such as Table 2.1, the forward filling imputation method could be used to fill in the missing data. Using this method would change the dataset, as shown in Table 2.2. This method is generally used for time series datasets and is one of the easiest ways to deal with missing values. However, a disadvantage of this method is that it will not be able to fill in the missing value if there is no value prior to the missing value.

Table 2.1: Sample Dataset

5	NaN	4
NaN	3	2
3	2	NaN

Table 2.2: Sample Dataset after Forward Filling Imputation

5	NaN	4
5	3	2
3	2	2

2.6 Outliers

Besides missing data, another problem possible in a dataset is the presence of outliers. Outliers can be defined as *a data in a dataset that strays from the other data* [35]. It is necessary to detect these outliers as they could skew the model's training which would reduce the accuracy of the model [45]. The removal of outliers is usually one of the earliest steps in a machine learning problem [45]. There are several methods that can be utilized in order to identify these outliers. One of those methods is to use Tukey's method. The Tukey's Method is based on statistics where data is expected to follow a distribution model such as normal distribution [46]. A data is considered an outlier if it deviates from the model [46]. The Tukey's Method divides the dataset into quartiles; the quartiles commonly used are the lower quartile (Q_1), median (Q_2), and upper quartile (Q_3) [46]. The equation for a quartile is :

$$Q_r = l_1 + \frac{r\left(\frac{N}{4}\right) - c}{f}(l_2 - l_1), \tag{2.7}$$

where Q_r is the r^{th} quartile, l_1 is the lower limit, l_2 is the upper limit, f is the frequency, and c is the cumulative frequency of the class preceding the quartile class [46]. The Tukey's Method involves calculating the Interquartile Range (IQR) between the lower quartile and the upper quartile in a boxplot [46]. The equation for the IQR is

$$IQR = Q_3 - Q_1. \quad (2.8)$$

In order to accurately determine which data is an outlier, the Tukey's Method calculates the upper limit and lower limit of the data distribution. The equation for the upper limit is

$$Upper\ Limit = Q_3 + (1.5 * IQR). \quad (2.9)$$

On the other hand, the equation for the lower limit is

$$Lower\ Limit = Q_1 - (1.5 * IQR). \quad (2.10)$$

The Tukey's Method will remove any data that does not fall between the upper limit and lower limit [46].

2.7 Correlations

In order to better understand a dataset, the correlation between features in the dataset could be considered. As mentioned in Section 2.5.1, correlation is known as a statistical measure that describes how one feature is related to another feature [39]. It is often used during Exploratory Data Analysis (EDA) to gain a better understanding of how a feature affects other features in the dataset. There are different types of correlations, namely positive correlation, negative correlation, and no correlation [39].

A positive correlation denotes that as the value of a certain feature rises, the value of another feature would rise as well [39]. In a graph format, a strong positive correlation would have a positive gradient, as shown in Figure 2.3.

A negative correlation denotes that as the value of a certain feature falls, the value of another feature would fall as well [39]. A negative correlation would have a negative gradient, as shown in Figure 2.4.

No correlation indicates that the features being assessed are not related; therefore, a change in one feature would not impact the other feature [39]. In a graph format, features with no correlation would look like Figure 2.5.

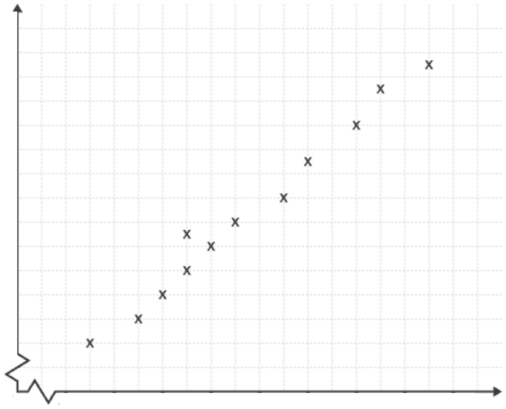


Figure 2.3: Positive Correlation *[47]*

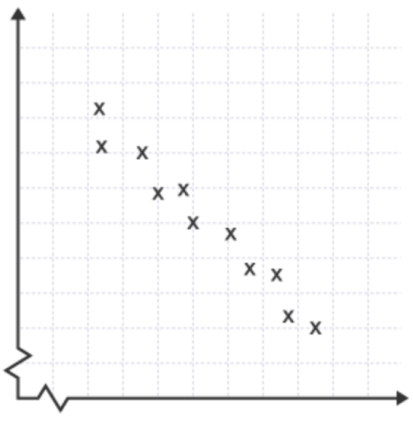


Figure 2.4: Negative Correlation *[47]*

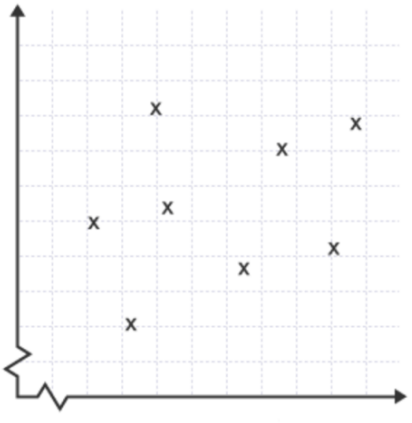


Figure 2.5: No Correlation *[47]*

For numeric features, the commonly used methods for measuring the correlation between features are Pearson Correlation and Spearman Correlation.

2.7.1 Pearson Correlation

In Pearson correlation, the features being compared get assigned a value between -1 and 1 [48]. A correlation value of 1 or – 1 would mean that the features being compared are strongly related to one another. A correlation value of 1 expresses that if one feature is present, then the other feature will unquestionably be present as well [48]. In addition to this, a correlation value of -1 would mean that if one feature is present, then the other feature will undeniably be absent [48]. There are also possibilities of having a correlation value of <1 or >-1. This means that the correlation is almost exactly positive or negative; however, there exists a small number of records that behave differently [48]. On the other hand, a correlation value of 0 would mean that the absence or presence of a feature is in no way related to the presence or absence of another feature [48].

The equation for Pearson correlation is :

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}, \qquad (2.11)$$

where r is the Pearson correlation coefficient, x is the values in the first set of data, y is the values in the second set of data, and n is the total number of values.

2.7.2 Spearman Correlation

The Spearman correlation is a method that measures the strength and direction of the relationship between two features in a dataset [49]. Spearman correlation requires

continuous data, which has a monotonic relationship. This means that when one feature increases, the other feature could either increase or decrease [49]. However, the relationship between the features does not have to be linear [49]. The correlation values in Spearman correlation follow the same principle as those in Pearson correlation. The values range from – 1 to 1 as well. If the correlation value is – 1, then as one variable increases, the other variable would decrease [49]. If the correlation value is 0, then a change in a variable would not affect the other variable [49]. On the other hand, if the correlation value is 1, then as one variable increases, the other variable would increase as well [49].

There are two equations that can be used to calculate Spearman's correlation. The first equation is :

$$\rho = 1 - \frac{6\sum_i d_i^2}{n(n^2-1)}, \tag{2.12}$$

where ρ is the spearman correlation, d_i is the difference between the features, and n is the total number of values [50]. Equation 2.12 can only be used if there are no duplicates in the dataset. If duplicates exist in the dataset, then the second equation will be used. The second equation is :

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}}, \tag{2.13}$$

where ρ is the spearman correlation, x is the value of feature x, $\bar{x}$ is the mean of feature x, y is the value of feature y, and $\bar{y}$ is the mean of feature y [50].

2.8 Feature Selection

After understating the dataset, the process of feature selection could be implemented. Feature selection is the process of cutting down the input variables which will be fed into the models [51]. This is useful as it gets rid of the noise in the dataset so that the model can focus on valuable information [51]. In order to determine which features are ideal to be used in the dataset, the Pearson correlation of the features should be taken into consideration as the values in the dataset are numerical [52]. It is ideal to add highly correlated features for the model's training. However, highly correlated parameters should not be the only features added to the model as they could reduce the model's accuracy [52]. It would lead to a lack of variation in the data or even result in data leakage, which would make the model perform unrealistically well [52].

2.9 Model Training and Hyperparameter Optimization

After the dataset has been cleaned and processed, the model will be trained on that dataset. A model consists of parameters that control the way it learns on the dataset, otherwise known as hyperparameters [53] [54]. Every algorithm has its own hyperparameter that can be defined by the data engineer. The gradient boosting algorithm and random forest algorithm also have several hyperparameters, such as the number of trees or maximum depth. These hyperparameters can affect the model's performance; therefore, it is imperative to find the best value for each hyperparameter [53] [54]. The process of discovering the ideal hyperparameters for a model is known as hyperparameter optimization or hyperparameter tuning [53] [54]. There are different methodologies for hyperparameter tuning, namely manual tuning, grid search, randomized search, and Bayesian optimization [55].

2.9.1 Manual Tuning

Manual tuning is the process of altering the hyperparameters manually to check the performance of the model with the hyperparameters specified [55]. After noting the model's performance, the hyperparameter is altered again manually without using any automation [55]. This is a time-consuming process; however, it helps the engineer gain an in-depth understanding of the hyperparameters [55].

2.9.2 Grid Search

In grid search optimization, a set of hyperparameters is defined from the beginning, and the model will train on all of it [55]. Afterwards, the best set of hyperparameters is returned. This is an exhaustive search process as every possible combination defined will be covered [55]. However, it could take a long time to compute, especially if there are a lot of hyperparameters that need to be tuned [55].

2.9.3 Random Search

Random search optimization is similar to grid search; however, the hyperparameters will be chosen randomly [55]. The number of times the process will run can be defined by the data engineer [55]. This method is not as computationally taxing as grid search; however, there are chances that the parameters will not be explored properly [55].

2.9.4 Bayesian Optimization

Bayesian optimization differs from other hyperparameter optimization methods as it is capable of remembering previous optimization results [55] [56]. It uses this memory to select the next set of hyperparameters to test on. The idea of Bayesian optimization is to reduce expensive computations by remembering the set of hyperparameters that performed well previously [55] [56]. There are four vital parts in Bayesian

optimization, the objective function, domain space, optimization algorithm, and the result history [56]. The objective function is the loss or error of a machine learning model based on the hyperparameters [56]. The domain space is the set of hyperparameter values defined beforehand [56]. The optimization algorithm is the method for choosing the next set of hyperparameters. The result history is the outcomes of the objective function for every iteration [56].

2.10 Model Evaluation

After the model has been trained, it is time to evaluate the performance of the model. Model evaluation is vital as it allows researchers to determine whether or not the model made is accurate. In order to evaluate models, researchers make use of metrics; the metrics for regression models are MAE, MSE, RMSE and, R^2. The MAE, MSE, and RMSE metrics greatly penalize outliers as their' value increases significantly in the presence of outliers [57]. For these metrics, a higher value indicates poor performance. However, RMSE is generally preferred over MAE and MSE as RMSE uses the same units as the variable in the y-axis [57]. The R^2 metric is also another ideal metric to consider as it is able to explain how well the model can predict the value compared to the original value [57].

2.10.1 Root Mean Square Error

This metric is the root squared average difference between the actual value and the predicted value [57]. RMSE is the square root of the MSE metric. The lower this value, the lower the deviations between the actual and predicted values [57].

The formula for RMSE is,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_p - y)^2}{n}}, \quad (2.14)$$

where y_p is the predicted value, y is the actual value, and n is the number of values [57].

2.10.2 Coefficient of Determinant

This metric is the measure of how well the regression model has predicted the value based on the actual value [57]. R^2 generally ranges from 0 to 1; however, there are instances when the value could be negative [57]. A R^2 value closer to 1 would mean that the model gives an accurate prediction. The formula for R^2 is

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_p - y)^2}{\sum_{i=1}^{n}(\bar{y} - y)^2}, \quad (2.15)$$

where y_p is the predicted value, y is the actual value, and $\bar{y}$ is the average of the actual values [57].

2.11 API

Sometimes the models developed might be used by external applications such as websites. In these cases, API (e.g., REST API or RESTful API) could be used to connect the models to the external application. Representational State Transfer (REST) is a type of architectural style that specifies principles that will act as a guide for website architecture design [58].

The REST API allows users to access web services in a simple manner. Users use Hypertext Transfer Protocol (HTTP) methods, namely GET, POST, DELETE, PUT,

and PATCH, to operate the resources such as websites [58]. The GET method is mainly used to read information; this method does not allow information modification [58]. The POST method is used to create new resources which are subordinate to another parent resource [58]. The DELETE method is used to delete an existing resource [58]. The PUT method is used to update a resource that is present; if the resource specified is not present, then a new resource could be generated [58]. The PATCH method is also used to update resources, similar to the PUT method [58]. However, the PATCH method only performs partial updates; it will not wholly change the resource [58]. Unlike the PUT method, the PATCH method is not capable of creating a new resource [58]. The RESTful API and REST API are very similar as RESTful APIs are APIs that follow the REST constraints [58].

There are several frameworks that can be used to create REST APIs or RESTful APIs, such as Flask or FastAPI. Flask is lightweight and modular and is 100% compliant with WSGI, which makes it easy to deploy for production [59]. However, it can also be time-consuming to use for large projects [59]. Furthermore, the Flask framework also relies on a lot of dependencies [59]. On the other hand, FastAPI is a new framework that has gained a lot of attention [59]. It contains all the features Flask contains; however, its speed is one of the features that distinguishes FastAPI from Flask. It can outperform Flask by 100%, and many consider FastAPI as the fastest python web framework [59]. Additionally, it is quite convenient to test FastAPI endpoints using the SwaggerUI provided [59]. SwaggerUI is part of Swagger; Swagger allows users to describe the structure of the API so that the machines are able to read it. SwaggerUI is an open source tool which will generate a webpage documenting the APIs generated by the Swagger specification [59]. The SwaggerUI makes testing the

endpoints quite efficient [59]. However, in FastAPI, as everything is tied to the file where the FastAPI app is defined, the main file tends to be crowded [59].

After developing an API, Docker can be used to package it. Docker is an open container-based platform that allows users to deploy and control applications on it. Docker provides a consistent and isolated environment for each container [60] [61]. Each container can access the resource they require without disrupting another container [60] [61]. Therefore, it reduces the chances of issues such as downtime. Furthermore, any application can be removed cleanly by simply deleting the container [60]. Another advantage of using Docker is that it is portable [60] [61]. It can run on any platform as long as the host operating system supports Docker. It can be deployed to any system that supports Docker, as all the dependencies will be packaged into a container [60] [61]. In addition to this, Docker also comes with an in-built version control system. It allows users to roll back to a previous version of a Docker image in case there are issues with the current version [60] [61]. Docker is also considered more secure as every application remains isolated from other applications. Therefore, a container cannot access another container without authorization [60] [61].

2.12 Cloud Computing

After an API has been packaged using Docker, it can be deployed on cloud services. Cloud computing is a technology that allows the running of a computer application as well as data savings over the internet platform [62]. The concept of cloud in cloud computing entails groups of computers, with each group involving thousands of computers connected to a network [62]. A cloud is a computing centre providing the

users with cloud applications and cloud data storage [62]. In the cloud, there are several computing services [63], such as:

- servers,
- storage,
- database, and
- networking.

Some popular cloud computing products are the Amazon Web Service (AWS) Elastic Computer, AWS Lambda, and Google Cloud Engine [63]. While the popular cloud computing services include the AWS Elastic Computing Cloud (EC2) and Google Cloud Platform [63].

2.13 Software Development Life Cycle

The Software Development Life Cycle (SDLC) methodology can be used when developing a project such as websites. SDLC is the process that is made up of steps that a particular software can follow in order to develop in a proper manner [64]. This would make it more likely for the project to be accomplished on time whilst ensuring the quality of the product is suitable for the user [65]. The activities for a specific SDLC can be labelled as [66]:

- understanding the case,
- deciding solution scheme,
- coding based on the solution decided, and
- testing.

However, these activities are quite broad; therefore, they can be broken down even further to illustrate the SDLC process better [64]. The phases of SDLC are

requirements analysis, design, development, testing, and deployment and maintenance [65].

Requirements analysis is the first phase of SDLC. In this phase, the business requirements of the project are gathered. The project managers and stakeholders will discuss to define the requirements of the software. These requirements could include answering questions such as “who will use the software” or “how will the system be used” [65]. After the discussion, a Software Requirement Specification (SRS) document will be created, which will contain the results of the discussion [65].

The main objective of the design phase is to turn the requirements specified in the first phase into an architecture [65]. In this phase, the hardware and system requirements are specified so that the architecture of the software can be defined [65]. Additionally, this phase is where testers are required to define what needs to be tested and how it should be tested [65].

In the development phase, the results of the design phase are converted into a system that meets the user requirements. A common name for this phase is the coding phase. All the developers and engineers play an active role in this phase, and they are required to follow the required guidelines defined beforehand [67]. It is the most extensive yet most crucial phase in the entire SDLC process. Additionally, the process of the development phase will be recorded in a document entitled Source Code Document (SCD) [67].

The next phase is the testing phase, where the software developed in the previous phase will be tested. There is usually a specific team whose purpose is solely to test the software; their job is to conduct a series of tests on the software [67]. The testing team

will document any errors they encounter and send this report to the development team so that the developers can attempt to remove the errors [65]. The testing phase is one of the most essential phases as it decides whether the software is eligible to be released to the users [67].

In the deployment and maintenance phase, the software has passed the testing phase and is bug-free; therefore, it is now deployed and useable by the client [67]. Additionally, in this phase, there are possibilities that the software needs to be updated due to technological advancements. Therefore, the developers need to maintain the software to ensure that its performance will not decline [64].

Over the years, the SDLC model has been adapted into different kinds of models. These models include the *Waterfall Model*, *V-shaped Model*, *Incremental Model*, *Agile Methodology,* and many more [66]. The agile methodology, in particular, is known for constant iterations for software testing and development [64]. In this methodology, it is typical for the development phase and the testing phase to occur concurrently [66]. The Agile methodology contains twelve core principles, which are [68] :

- customer satisfaction,
- adaptive to changing requirements,
- regular software delivery, the faster the better,
- productive collaboration between developers and stakeholders,
- support developers by supplying an ideal work environment and believe that they will accomplish the project,
- direct face-to-face communication for team discussion,
- assess progress by checking on working software,

- encourage maintainable development,
- constant focus on technical quality and design,
- simplicity is vital,
- working units that can organize themselves will provide the ideal output (design, software architecture, requirements), and
- occasional reflection so that the team can improve.

The agile methodology also consists of a framework entitled SCRUM. SCRUM is an agile development methodology that is based on an iterative as well as an incremental process [66]. One of the main features of SCRUM is that it focuses more on feedback, revisions, and frequent customer engagement rather than documenting procedures and predicting a plan of action for accomplishing the project [66]. In SCRUM, there are three prominent roles which are Product Owner (PO), Scrum Master (SM), and Scrum Team (ST). There is a lack of guidelines or descriptions for how the project should be accomplished in SCRUM; most of the decision-making is left to the team doing the project as the team knows best [66]. There are three constants in SCRUM, which are Product Backlog, Sprint Backlog, and Sprint Goal [66]. Product Backlog is the list of things that need to be done by the PO, Sprint Backlog is the list of things selected by the ST that needs to be done in the current sprint cycle, whereas Sprint Goal is the endgame of the current sprint [66]. There are software for tracking project progress that implements the agile methodology such as Jira. Jira allows teams to create scrum boards and it provides customized agile reports such as a Cumulative Flow Diagram (CFD). A CFD can be used by teams in order to determine if the project progress of the team is stable, over-capacitated or if the team is taking on more tasks than they can handle [69]. SCRUM Methodology and the Jira software can be beneficial for complicated projects, and greatly helps the project progress efficiently [66].

CHAPTER 3

PROBLEM ANALYSIS

This chapter will discuss the problem statement of this project as well as the proposed solution for the problem. It will also discuss existing works done in this field.

3.1 Problem Statement

In a new area, oil and gas companies have to drill exploratory wells to discover whether or not there is a presence of oil and gas [70]. If the presence of oil and gas is detected, then the company would continue to drill more wells, known as development wells, to obtain the oil and gas [70]. However, these development wells do not always live up to the companies' expectations as it does not contain large amounts of oil and gas; therefore these wells end up being abandoned [70]. Abandoning these wells would mean that both the time and money of the company have been wasted.

The oil and gas industry has a lot of data; data which includes the pressure and temperature of the wells [12]. These data can be used to help lessen the impact of the problem stated previously. These data could be used by machine learning algorithms to predict oil and gas production values. There are lots of machine learning algorithms that can be used for oil and gas prediction. Therefore, it is necessary to discover the ideal machine learning algorithm to predict oil and gas production for this project. Afterwards, the author has to provide a way to connect the model to the VDR website application so that the author's teammates can visualize the predicted oil and gas production values in the website.

3.2 Related Works

In [71], Xie, Chao, Qin, and Li made use of 2 models to predict the concentration of gas. Xie et al. used a LSTM model and a random forest model and compared the results. An LSTM model is a variant of RNN, which is capable of remembering past information, which makes it suitable for predicting features. However, as mentioned in Section 2.3, deep learning models do not perform as well as tree-based algorithms when it comes to structured tabular data. On the other hand, the random forest algorithm is a tree-based algorithm that performs exceptionally well on tabular data. In this study, the models were evaluated with the R-squared score, RMSE, and MAE. The LSTM has an R-squared value of 0.31, an RMSE value of 0.45, and a MAE value of 0.56. On the other hand, the random forest model has a R-squared value of 0.95, RMSE value of 0.23, and MAE value of 0.34. From the values of these evaluation metrics, the researchers concluded that the random forest model was simpler and gave better results than the LSTM model.

In [72], Chauhan made use of Facebook's Prophet model in order to predict gas production. The dataset used by Chauhan was Canadian's natural gas production; the dataset contained two columns which were the date and the volume of gas. The model was evaluated with the R-squared score and the MAE metric. The R-squared value was 0.911, whereas the MAE score was 7782. An advantage of using the Prophet model is that the results are easy to understand [73]. However, it requires a large dataset as it is recommended to have at least two or three years of historic data [73]. The Prophet model works better if the dataset contains daily and weekly observations [73]. Furthermore, though this model performs quickly, the results are often less accurate compared to when other algorithms are used [74].

Chahar in [75] shows the performance of linear regression in predicting oil production. The dataset used was the Volve dataset which is located in the North Sea and was updated on a daily basis from 2005 to 2016. The linear regression model was evaluated with the R-squared score; the value was 0.55. An advantage of linear regression is that there are low chances of the model overfitting [76]. Additionally, it is easy to implement and works exceptionally well on variables that have a linear relationship [76]. On the other hand, linear regression models perform poorly when the relationship between variables is non-linear [76]. Furthermore, there are possibilities for linear regression models to underfit [76].

The performance of polynomial regression in predicting oil production was also shown in [75]. The same dataset used for the linear regression model was used for this polynomial regression. The model was evaluated with the R-squared score; the value was 0.95. An advantage of polynomial regression is that it works well even if the variables do not have a linear relationship [77]. Furthermore, the dataset size does not matter, as polynomial regression works well regardless of dataset size [77]. On the other hand, polynomial regression is extremely sensitive to outliers; the results could change drastically with the presence of one outlier [77].

To best of the author's knowledge, there is no similar VDR application that provides the oil and gas production data prediction.

Table 3.1 summarizes the comparison of the studies mentioned earlier.

Table 3.1: Summary of Research

Title	Model Used	Metrics / Performance	Predicted Feature	Reference
"Using a stochastic forest prediction model to predict the hazardous gas concentration in a one-way roadway"	LSTM	R-squared : 0.31 RMSE : 0.45 MAE: 0.56	Gas Concentration	[60]
"Using a stochastic forest prediction model to predict the hazardous gas concentration in a one-way roadway"	Random Forest	R-squared : 0.95 RMSE : 0.23 MAE: 0.34	Gas Concentration	[60]
"Using Facebook Prophet for Forecasting Natural Gas Production"	Facebook's Prophet	R-squared : 0.911 MAE : 7782	Gas Production Value	[62]
"Prediction of Oil Production by applying Machine Learning on Volve Field Production Data."	Linear Regression	R-squared : 0.55	Oil Production Value	[65]
"Prediction of Oil Production by applying Machine Learning on Volve Field Production Data."	Polynomial Regression	R-squared : 0.95	Oil Production Value	[65]

3.3 Proposed Solution

Machine learning can help determine how likely the well would contain oil and gas. Through machine learning predictions, users can focus on the wells which contain more oil and gas which would save time and money as they would not waste time on wells that contain less oil or gas. The oil and gas industry contains lots of data; data which can be used for machine learning [12]. The author plans to develop a predictive model that can predict oil and gas production. In order to determine the ideal model for predicting oil and gas production, the author will conduct a comparative study to find the best performing model. The author will also build an oil and gas prediction application as a backend service and integrate this service with the existing backend services. The communication between the client-side application and the server-side prediction service is performed over an API.

3.3.1 Model Selection

As shown in Section 3.2, there are several models that have been used in the field of oil and gas production prediction. Amongst these models, the random forest algorithm was shown to achieve one of the best results.

Oil and gas datasets are structured and tabular, thus, as mentioned in Section 2.3, tree-based algorithms are more ideal for tabular datasets [33] [34]. Tree-based models find tabular structured data more natural [33] [34]. Additionally, tree-based models are also deterministic which means that the output is determined solely on the input feature values [33] [34]. This makes it ideal for oil and gas prediction as it oil and gas production values depend on the parameter values. Therefore, this project will use make use of tree-based algorithms.

The author will compare the random forest algorithm and gradient boosting algorithms to determine which one performs better. The gradient boosting algorithm has the capability of giving a more accurate result compared to the random forest algorithm. This is because in the gradient boosting algorithm, the trees are trained one by one; thus, the current tree is capable of correcting the error of the previous one [78]. The author will test the algorithms on different hyperparameters to determine which hyperparameter would give a better result.

3.3.2 Server-Side Oil and Gas Production Data Prediction Application

After the best oil prediction and gas prediction models has been selected, the author will save the best models. The FastAPI framework will then be used to create an API for the backend prediction application. The application will then be packaged as a container using Docker in the backend and deployed as part of the backend service. The author's teammates will then connect the client-side application to the API endpoints in order to obtain the oil and gas production data that will be visualized in the website.

CHAPTER 4

SOLUTION DESIGN

This chapter will briefly depict the overall system architecture of the VDR website application develop by the author's team. It will delve into how the author will develop the prediction model and which will be deployed as part of the backend service. It will also discuss how the author plans to develop and package the API for communicating between the frontend and the backend prediction service.

4.1 System Architecture

The system architecture of the VDR website application is shown in

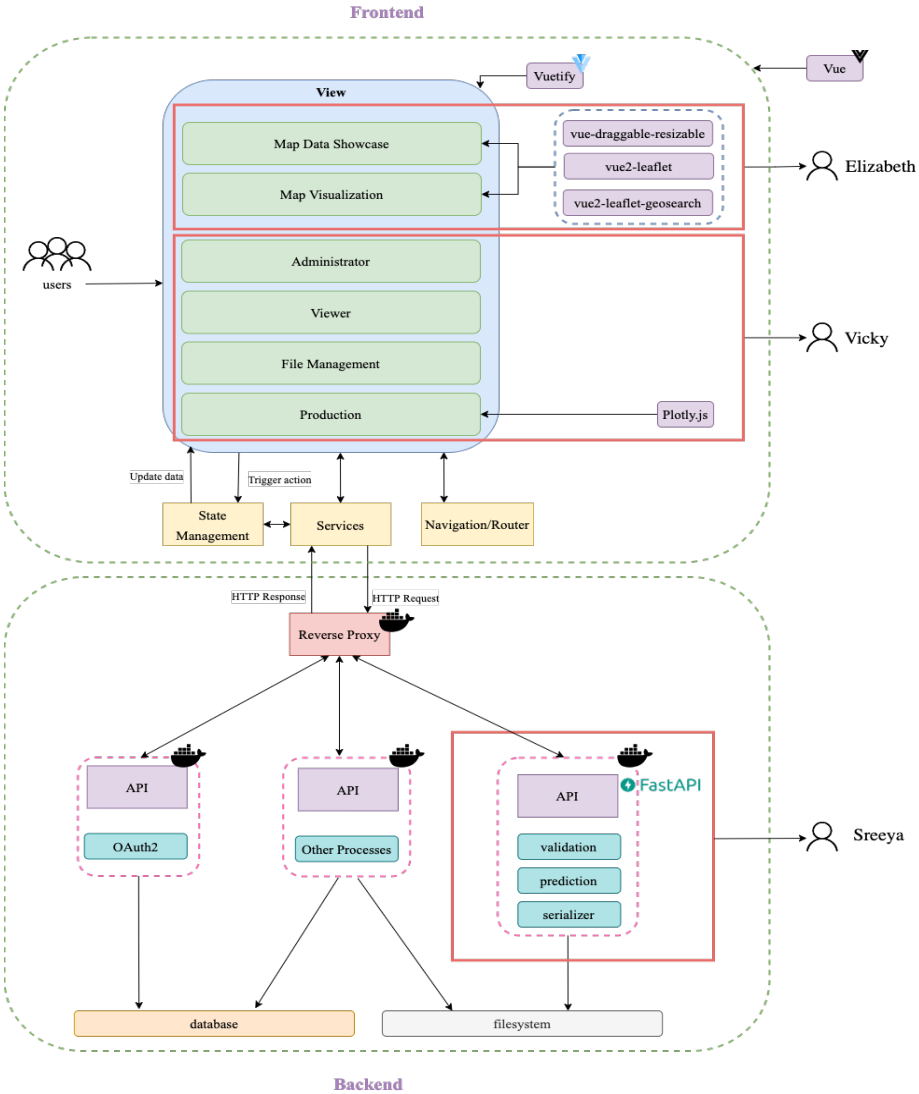


Figure 4.1.

Figure 4.1: System Architecture

The author will be responsible for developing the server-side oil and gas production data prediction application. The author will select two models, one for predicting oil production data, and another model for predicting gas production data. For the development of this project, the author's team will be following the development using sprints in JIRA as shown in Figure 4.2.

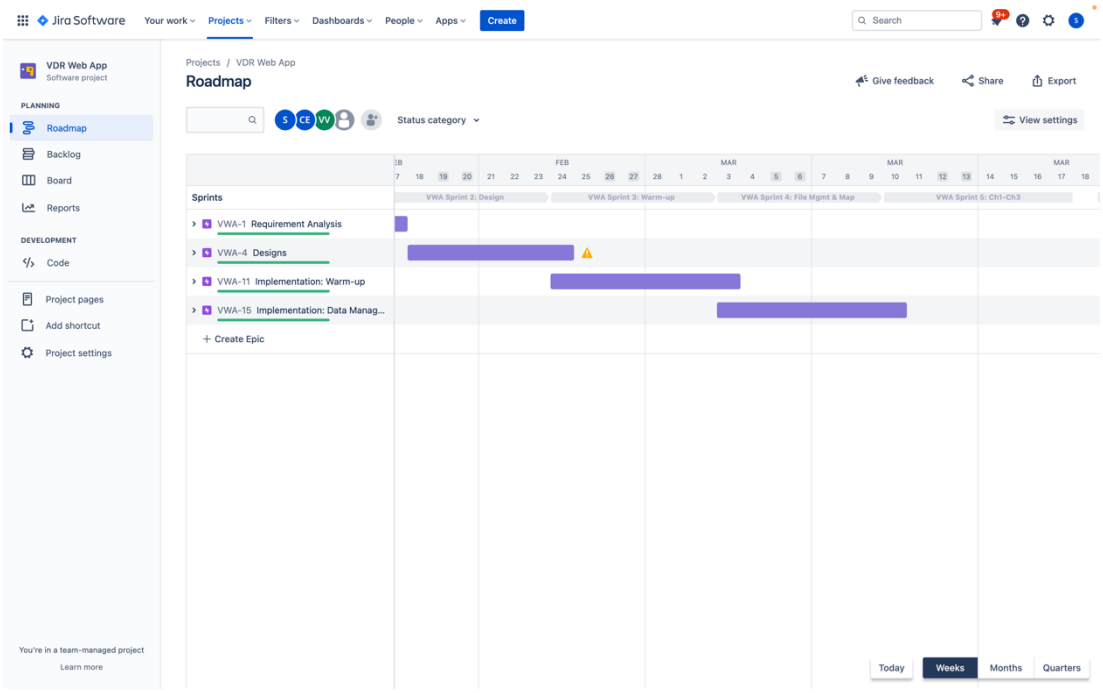


Figure 4.2: Project Roadmap

4.2 Machine Learning Models

This section will describe the steps taken to develop the prediction model, such as data cleaning and pre-processing, model training, and model evaluation. It will also discuss the experiments that will be conducted on the models in order to determine which would give a better performance.

Figure 4.3 describes the overall methodology for data preparation, model training, and evaluation.

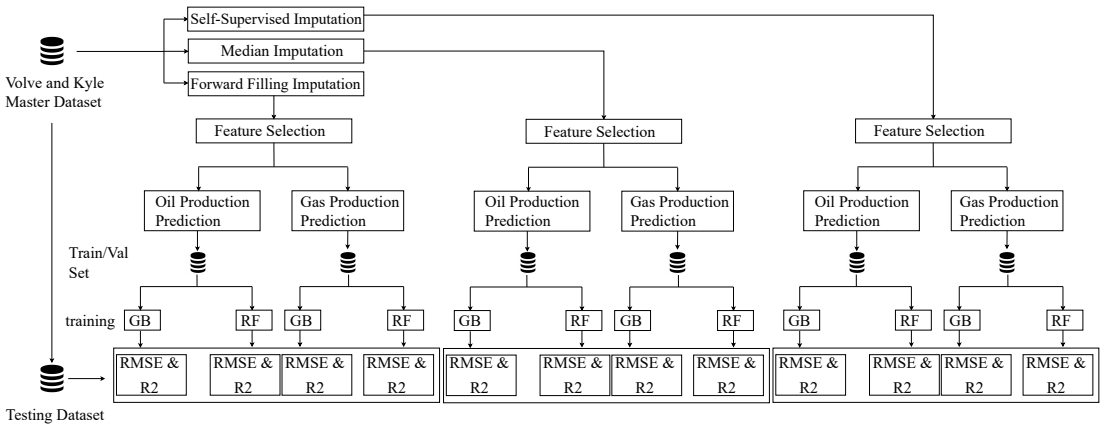


Figure 4.3: Methodology; where self-supervised imputation is the author's proposed imputation method

The missing values in the dataset will be filled in using three different methods as shown in Figure 4.3. Afterwards, only certain features necessary for model training will be selected. Then, the model will train on the dataset and predict oil and gas production. After the model has been trained, the model will be evaluated with the RMSE and R-squared metric. As shown in Figure 4.3, there will be six different datasets, thus there will be twelve models. These twelve models will be evaluated and the best models will be selected.

4.2.1 Data Collection

For the project, the author utilized two open-sourced datasets. The first dataset is entitled Volve, whilst the second dataset is entitled Kyle Master. The Volve dataset contained 15,634 rows of data and was obtained from *Kaggle*[1]. On the other hand, the Kyle Master dataset contained 27,324 rows of data and was obtained from the online data centre of the *Oil and Gas Authority*[2]. It is ideal to use a large dataset as it would lead to lower estimation variance, which means the model will be able to predict more

[1] https://www.kaggle.com/datasets/nazarmahadialseied/volve-field-production-dataset-oil-and-gas

[2] https://experience.arcgis.com/experience/50b61d215bff4072bf0649efe6e8d845/page/Page/?views=by-Field

accurately. Both Volve and Kyle Master datasets contain valuable information. The columns, their unit of measurement, as well as their meanings are shown in Table 4.1.

Table 4.1: Columns in Volve and Kyle Master dataset

Volve Dataset	**Kyle Master Dataset**	**Unit of Measurement**	**Feature Description**
DATEPRD	Date	-	Date
WELL_BORE_CODE	Wellbore ID	-	ID of the wellbore
NPD_WELL_BORE_CODE	-	-	ID of the wellbore per Norwegian Petroleum Directorate
NPD_WELL_BORE_NAME	-	-	Name of the wellbore per Norwegian Petroleum Directorate
NPD_FIELD_CODE	-	-	Field code per Norwegian Petroleum Directorate
NPD_FIELD_NAME	-	-	Field name per Norwegian Petroleum Directorate
NPD_FACILITY_CODE	-	-	Facility code per Norwegian Petroleum Directorate
NPD_FACILITY_NAME	-	-	Facility name per Norwegian Petroleum Directorate
ON_STREAM_HRS	Hours Online	hours	How long the machine has been operating
AVG_DOWNHOLE_PRESSURE	Av. DHP (bar)	bar	Pressure measured at the bottom of the well
AVG_DOWNHOLE_TEMPERATURE	Av. DHT (Deg C)	ºC	Temperature measured at the bottom of the well

AVG_DP_TUBING	-	bar	Pressure build-up in the tubing
AVG_ANNULUS_PRESS	-	bar	Pressure between the tubing and the casing
AVG_CHOKE_SIZE_P	Platform Choke %	%	Size of choke
AVG_CHOKE_UOM	-	%	Unit of measurement
AVG_WHP_P	Av. WHP (bar)	bar	Pressure difference measured at the top of the well
AVG_WHT_P	Av. WHT (Deg C)	ºC	Temperature difference measured at the top of the well
DP_CHOKE_SIZE	-	-	Size of choke
BORE_OIL_VOL	Oil (m3)	m^3	Volume of oil produced
BORE_GAS_VOL	Gas (m3)	m^3	Volume of gas produced
BORE_WAT_VOL	Produced Water (m3)	m^3	Volume of water produced
BORE_WI_VOL	-	-	Volume of injected water
FLOW_KIND	-	-	What kind of well is it (production/injector)
WELL_TYPE	-	-	Type of well

4.2.1.1 Correlation

The author made use of Pearson's correlation, which was described in Section 2.7.1, to calculate the correlation between the features. Table 4.2 describes the features with high correlation values in Volve and Kyle Master datasets before data cleaning. On the other hand, Table 4.3 describes the features with low correlation values in Volve and Kyle Master datasets before data cleaning.

Table 4.2: Features with High Pearson Correlation in Volve and Kyle Master

Volve		**Kyle Master**	
Features	**Correlation**	**Features**	**Correlation**
BORE_OIL_VOL and BORE_GAS_VOL	0.999	Av. WHT (Deg C) and Oil (m3)	0.565
AVG_DOWNHOLE_PRESSURE and AVG_ DP _TUBING	0.949	Av. WHT (Deg C) and Gas (m3)	0.552
BORE_WAT_VOL and AVG_CHOKE_SIZE_P	0.760	Av. DHP (bar) and Av. DHT (Deg C)	0.577

Table 4.3: Features with Low Pearson Correlation in Volve and Kyle Master

Volve		**Kyle Master**	
Features	**Correlation**	**Features**	**Correlation**
BORE_GAS_VOL and BORE_WAT_VOL	-0.009	Platform Choke % and Gas (m3)	-0.0585
BORE_WI_VOL and NPD_WELL_BORE_CODE	-0.055	Produced Water (m3) and Av. DHP (bar)	-0.009
DP_CHOKE_SIZE and AVG_DP_TUBING	0.093	Platform Choke % and Produced Water (m3)	-0.17

4.2.1.2 Feature Selection

This section will explain and justify which features will be used for the model's training. Table 4.4 shows the features that are selected for model training. As the goal is to create a model that can predict oil and gas production, it is essential to include their production values. In the Volve dataset, the first two features selected for model training are *BORE_OIL_VOL* and *BORE_GAS_VOL*. As mentioned in Section 2.1, oil and gas formation are also reliant on pressure and temperature. Therefore, *AVG_DOWNHOLE_PRESSURE, AVG_DOWNHOLE_TEMPERATURE,*

AVG_WHP_P and AVG_WHT_P are also included. *ON_STREAM_HRS* will also be added as this column shows how long the machine operates. In the Kyle Master dataset, the first two features selected are *Oil (m3)* and *Gas (m3),* as these features contain the production value of oil and gas. Additionally, as oil and gas production is reliant on the pressure and temperature of the reservoir, the features *Av. DHT (Deg C), Av. DHP (bar), AV. WHT (Deg C), and AV. WHP (bar)* are added for the model's training. Lastly, *Hours Online* will also be added for training the model. Table 4.4 shows the features that have been selected for model training.

Table 4.4: Features selected for training in Volve and Kyle Dataset

Volve Dataset	**Kyle Master Dataset**	**Unit of Measurement**	**Selected for Model Training**
DATEPRD	Date	-	No
WELL_BORE_CODE	Wellbore ID	-	No
NPD_WELL_BORE_CODE	-	-	No
NPD_WELL_BORE_NAME	-	-	No
NPD_FIELD_CODE	-	-	No
NPD_FIELD_NAME	-	-	No
NPD_FACILITY_CODE	-	-	No
NPD_FACILITY_NAME	-	-	No
ON_STREAM_HRS	Hours Online	hours	Yes
AVG_DOWNHOLE_PRESSURE	Av. DHP (bar)	bar	Yes
AVG_DOWNHOLE_TEMPERATURE	Av. DHT (Deg C)	ºC	Yes
AVG_DP_TUBING	-	bar	No
AVG_ANNULUS_PRESS	-	bar	No
AVG_CHOKE_SIZE_P	Platform Choke %	%	No

AVG_CHOKE_UOM	-	%	No
AVG_WHP_P	Av. WHP (bar)	bar	Yes
AVG_WHT_P	Av. WHT (Deg C)	°C	Yes
DP_CHOKE_SIZE	-	-	No
BORE_OIL_VOL	Oil (m3)	m^3	Yes
BORE_GAS_VOL	Gas (m3)	m^3	Yes
BORE_WAT_VOL	Produced Water (m3)	m^3	No
BORE_WI_VOL	-	-	No
FLOW__KIND	-	-	No
WELL_TYPE	-	-	No

These datasets both have similar columns even though the names are different. For instance, the features *Av. DHT (Deg C)* and *Av. DHP (bar)* in the Kyle Master dataset has the same meaning as the features *AVG_DOWNHOLE_PRESSURE* and *AVG_DOWNHOLE_TEMPERATURE* in the Volve dataset. Additionally, *Oil (m3)* and *Gas (m3)* in the Kyle Master dataset have the same meaning as *BORE_OIL_VOL* and *BORE_GAS_VOL* in the Volve dataset.

4.2.1.3 Feature Statistics

In order to better understand the selected features in the dataset, several techniques were employed to understand how the data is distributed. Table 4.5 describes the selected features of the Volve dataset, whereas Table 4.6 describes the selected features for the Kyle Master dataset. The histogram, boxplot with and without outliers for these features can be seen in Appendix A in Figures A.5 – A.18.

Table 4.5: Feature Statistics for Volve Dataset

Feature	Range	Outlier Count	Mean	Standard Deviation	Variance
ON_STREAM_HRS	25 hours	715	23 hours	3 hours	9 hours
AVG_DOWNHOLE_PRESSURE	307 bar	144	240 bar	22 bar	484 hours
AVG_DOWNHOLE_TEMPERATURE	107.7 °C	156	104 °C	4 °C	16 °C
BORE_OIL_VOL	5,900 m^3	283	1,476 m^3	1,464 m^3	2,143,296 m^3
BORE_GAS_VOL	86,863 m^3	182	215,541 m^3	207,094 m^3	42,887,924, 836 m^3
AVG_WHP_P	120 bar	44	48 bar	20 bar	400 bar
AVG_WHT_P	86 °C	352	73 °C	18 °C	324 °C

Table 4.6: Feature Statistics for Kyle Master Dataset

Feature	Range	Outlier Count	Mean	Standard Deviation	Variance
Hours Online	1,912 hours	1,326	23 hours	27 hours	729 hours
Av. DHP (bar)	1,122 bar	3	111 bar	39 bar	1,521 bar
Av. DHT (Deg C)	245 °C	645	94 °C	9 °C	81 °C
Oil (m3)	3,509 m^3	447	380 m^3	328 m^3	107,584 m^3
Gas (m3)	1,304,298,362, 420 m^3	226	178,525,800,000 m^3	175,599,300,000 m^3	30,835,114,160, 490,000,000,00 0 m^3

Av. WHP (bar)	325 bar	48	57 bar	35 bar	875 bar
Av. WHT (Deg C)	228 ºC	597	62 ºC	19 ºC	361 ºC

In Table 4.5 and Table 4.6, range denotes the range of the specified feature; more specifically, it is the difference between the lowest value up to the highest value of the feature. Outlier count is the number of outliers in the feature. The mean is the average of the feature. Standard deviation is the measure of how varied the feature is relative to the mean.

From Table 4.5 and Table 4.6, it can be seen that the range of values for all the selected features in the Kyle dataset is larger than the features in the Volve dataset. This denotes that the data in the Kyle dataset is more dispersed compared to the Volve dataset. In addition to this, the standard deviation and variance of the features in the Kyle dataset are much larger than the features in the Volve dataset. This observation further supports the fact that the features in the Kyle dataset are more spread out than the features in the Volve dataset.

4.2.2 Data Cleaning and Pre-processing

Volve and Kyle Master contained missing data; therefore, it is imperative to check the relationship between the features in the dataset. This is done so that it can be determined whether or not the presence of the missing value is correlated to other values in the dataset. In order to check this, a heatmap was used to see the correlation values on both datasets. The heatmap can be seen in Appendix A in Figure A.1 and Figure A.2. Table 4.7 describes the observations derived from the heatmaps. As stated

in Table 4.7, the Volve dataset follows the MNAR mechanism, whereas the Kyle Master dataset follows the MAR mechanism. Section 2.5.1 states that these missing mechanisms imply that the missing values are dependent on one another. Thus, it should not be ignored and should either be deleted or filled in using data imputation methods. Furthermore, the missing oil and gas values in the dataset reaches 42%, which furthermore reinforces that the missing values should not be ignored due to how high the percentage is.

Table 4.7: Observations for Missing Data

Dataset	**Volve**	**Kyle Master**
Observation	Contains mainly "<1" and "1" feature correlation values, meaning the features are highly dependent on one another. A value of "<1" denotes that the correlation is almost exactly 1.	Feature correlation values are mostly 0.1, and some features have a correlation value of 1, meaning most of the features do not show much correlation, however, few features are highly correlated.
Missing Data Mechanism	MNAR	MAR

For this project, the author will use three methods and compare them to see which method would make the model perform better. The first method the author will use is forward filling imputation, where the empty value is replaced by the last observed record. The second method used is central value imputation, where the author will fill in the missing values with the median value of the feature. Due to the high percentage of missing values in the dataset, the author believes that another type of data imputation method should be explored. The third method the author used is a self-supervised imputation method where the missing values will be filled in by a machine learning model. The author noticed that there are data in the dataset where the pressure and temperature values are filled in. Therefore, a baseline gradient boosting and

random forest model was used to predict these values. Afterwards, the author placed these values into the original dataset.

For the model's training, the author combined both the Volve and Kyle Master datasets. The selected columns in the Volve dataset and the Kyle Master datasets have the same meaning. Therefore when combining the datasets, the columns in the Volve dataset were renamed to match the columns in the Kyle Master dataset. Table 4.8 describes the selected features of the combined dataset after forward filling imputation is used. Additionally, Table 4.9 describes the selected features of the combined dataset after median imputation is used. On the other hand, Table 4.10 describes the selected features of the combined dataset after self-supervised imputation is used. The boxplot, histogram, and kernel density function for these features can be seen in Appendix A in Figures A.19 – A.39.

Table 4.8: Feature Statistics for Volve and Kyle Master Dataset after forward filling imputation

Feature	**Range**	**Mean**	**Standard Deviation**	**Variance**
Hours Online	1,912 hours	23 hours	22 hours	484 hours
Av. DHP (bar)	162 bar	102 bar	19 bar	361 bar
Av. DHT (Deg C)	344 °C	158 °C	74 °C	5,476 °C
Oil (m3)	5,900 m^3	801 m^3	1,117 m^3	1,247,689 m^3
Gas (m3)	1,164,213 m^3	176,375 m^3	178,282 m^3	31,784,471,524 m^3
Av. WHP (bar)	325 bar	49 bar	29 bar	841 bar

Av. WHT (Deg C)	228 ºC	64 ºC	18 ºC	324 ºC

Table 4.9: Feature Statistics for Volve and Kyle Master Dataset after median imputation

Feature	Range	Mean	Standard Deviation	Variance
Hours Online	1,912 hours	22 hours	19 hours	361 hours
Av. DHP (bar)	307 bar	182 bar	69 bar	4,761 bar
Av. DHT (Deg C)	108 ºC	100 ºC	4 ºC	16 ºC
Oil (m3)	5,900 m^3	733 m^3	927 m^3	859,329 m^3
Gas (m3)	13,044,298 m^3	155,145 m^3	162,085 m^3	26,271,547,225 m^3
Av. WHP (bar)	325 bar	45 bar	23 bar	529 bar
Av. WHT (Deg C)	228 ºC	69 ºC	17 ºC	289 ºC

Table 4.10: Feature Statistics for Volve and Kyle Master Dataset after self-supervised imputation

Feature	Range	Mean	Standard Deviation	Variance
Hours Online	1,912 hours	23 hours	22 hours	484 hours
Av. DHP (bar)	308 bar	163 bar	70 bar	4,900 bar

Av. DHT (Deg C)	105 °C	99 °C	5 °C	25 °C
Oil (m3)	5,888 m^3	827 m^3	1109 m^3	1,229,881 m^3
Gas (m3)	13,044,297 m^3	194,119 m^3	190,655 m^3	36,349,329,025 m^3
Av. WHP (bar)	325 bar	49 bar	28 bar	784 bar
Av. WHT (Deg C)	228 °C	66 °C	18 °C	324 °C

Table 4.8, Table 4.9, and Table 4.10 show that the mean and standard deviation of the datasets after data imputation has changed slightly. Most of the mean and standard deviation of the features in Table 4.10 is greater than the features in Table 4.8 and Table 4.9. This shows that the distribution of the dataset after self-supervised imputation is more dispersed compared to the dataset after forward filling and median imputation.

4.2.3 Data Splitting

The combined dataset will then be split into three sets, namely a training set, a testing set, and a validation set. The purpose of the training set will be to train the model, additionally, the testing set will be used to evaluate the model's performance. The validation set will be used during hyperparameter optimization. The ratio for splitting is 80% for training, 10% for testing, and 10% validation.

4.2.4 Hyperparameter Optimization

As mentioned in Section 3.3.1, the author will test the algorithms on different hyperparameters to determine which would result in a better performance. The hyperparameters that will be tuned for the algorithms are described in Section 4.2.4.1 and Section 4.2.4.2.

4.2.4.1 Hyperparameter Optimization for Gradient Boosting

In the gradient boosting algorithm, the hyperparameters that will be tuned are the learning rate, number of trees, and the maximum depth. Learning rate is an important hyperparameter as it controls how quickly the model learns [79]. The learning rate hyperparameter is closely related to the number of trees parameters. The number of trees denotes the number of trees that will be used [79]. A fine balance has to be achieved between the learning rate and the number of trees, as the smaller the learning rate, the higher the number of trees should be. It is ideal to use a low learning rate because it would let the model train slower, which makes it more efficient [79]. However, the number of trees should not be too high as it would result in overfitting [79]. The third hyperparameter that the author will tune is the maximum depth. The maximum depth is how deep the tree is allowed to be. The deeper the tree, the more it will split and learns information about the dataset. However, if the depth is too high, it could lead to overfitting [79].

4.2.4.2 Hyperparameter Optimization for Random Forest

In the random forest algorithm, the hyperparameters that will be tuned are the number of trees, max depth, and the minimum number of samples needed to split the leaf node, also known as min_samples_split. The number of trees and max depth

hyperparameters have been explained in the previous section. The other hyperparameter that the author will tune for this model is the min_samples_split. The lower the value of this parameter, the more the tree will split [80]. If the tree splits too much, it could lead to overfitting [80]. However, a high value for this parameter is not ideal as well. A high value would mean that the tree would not split as much, which would result in the model underfitting [79] [80]. Therefore, it is necessary to find the right number for this parameter.

4.2.5 Evaluation Technique

As shown in Figure 4.3, the evaluation techniques used by the author are RMSE and R-Squared. Based on these metrics, the author will determine the best performing model. The best performing model should have a low RMSE value and a high R-Squared value.

4.3 Server-Side Oil and Gas Production Data Prediction Application

This section will describe how the author will save the best models selected and also contain an overview on the API endpoints that the author plans to create for the client-side application for communication between the frontend and the backend prediction service.

4.3.1 Saving the Models

The best oil production and gas production prediction model will be chosen and saved into pickle files. Pickle is a Python module which can be used to serialize and deserialize a Python object. Through pickle, the machine models can be saved which eliminates the need to retrain the model as the already trained model can simply be reloaded using pickle. The author will be using the pickle module because the author

plans to use Python for the machine learning models, and the pickle module is a Python tool which makes saving these models simple.

4.3.2 RESTful API

The author will develop a RESTful API which will have two types of endpoints, one for oil prediction and another for gas prediction. These endpoints will be of the method POST, it will take in the features (e.g. pressure and temperature) as input and use the saved predictive models to output the prediction values. The endpoints will act as one end of the communication channel. Since the communication will be over HTTP, the endpoints will be in a Uniformed Resource Language (URL) form.

CHAPTER 5

IMPLEMENTATION

This chapter will discuss the results obtained during this project. It will show the development of the author's team project and also the results of conducting hyperparameter optimization using the Bayesian optimisation method. It will also show the performance of each model that the author compared. Afterwards, it will discuss the API endpoints made by the author that will be connected to the client-side application.

5.1 JIRA Sprints

In order to develop the project while collaborating efficiently with team members, the author made use of the SCRUM framework which is part of the agile methodology. For the author's team, a sprint will last for a week and each member is assigned several tasks that should be accomplished by the end of the current sprint. At the end of each sprint, the goals for the next sprint will be assigned by the SCRUM master. An example of a sprint is shown in Figure 5.1.

Completed issues — View in issue navigator

Key	Summary	Issue type	Epic	Status	Assignee	Story points
VWA-73	Mock up WireFrame Layout	Story		DONE	CE	-
VWA-80	data showcase, download, sorting, area geometry	Story		DONE	CE	-
VWA-85	design production page UI for sreeya's AI	Story		DONE	S	-
VWA-86	remove navbar, replace w vdr logo	Story		DONE	VV	-
VWA-87	prepare layout for sreeya's AI	Story		DONE	VV	-
VWA-88	profile pic, specify user (regular, premium, admin)	Story		DONE	VV	-
VWA-89	specify client details	Story		DONE	VV	-
VWA-90	latitude longitude, reverse geocoding	Story		DONE	CE	-
VWA-91	remake showcase page, w map, showcase, and add wells	Story		DONE	CE	-
VWA-92	parsing the data to treeview	Story		DONE	CE	-
VWA-93	splitting training experiment	Story		DONE	S	-
VWA-94	finish the model n the github	Story		DONE	S	-

Figure 5.1: Example of Group Sprint; where VV refers to Vicky, CE refers to Elizabeth, and S refers to Sreeya

If due to unforeseen reason, a member fails to accomplish the task assigned to them, the task will be moved to the next sprint. The member will then strive to accomplish the tasks while ensuring that the progress of the other tasks planned beforehand will not be derailed.

In addition to this, based on the author's team CFD shown in Figure 5.2, the bands are generally progressing parallelly, thus the project progress of the author's team is quite stable.

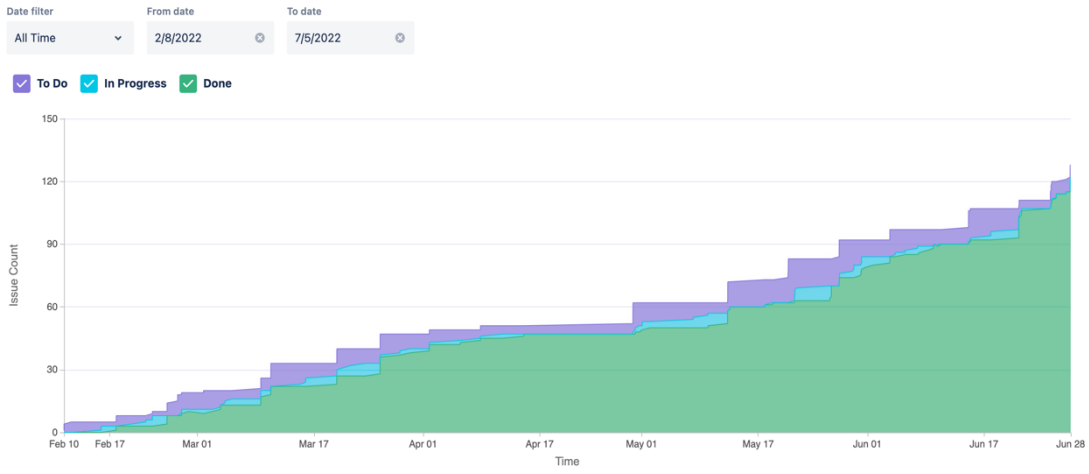


Figure 5.2: Cumulative flow diagram

5.2 Machine Learning Models

This section will discuss the results obtained during model training and evaluation.

5.2.1 Hyperparameter Optimization

In order to optimise the models to deliver optimal results, the Bayesian Optimization method could be used. This method was used on the different datasets. For the gradient boosting model, the range for the number of trees is 5 to 100, the range for maximum depth is 2 to 50, whereas the range for learning rate is 0 to 1. For the random forest

model, the range for the number of trees is 5 to 100, the range for maximum depth is 2 to 50, whereas the range for min_samples_split is 2 to 50. Every model went through 1,000 iterations during the hyperparameter optimization before the final hyperparameters were obtained. During the hyperparameter optimization, for each model, the RMSE metric was used on the validation dataset as the performance metric. The results of the hyperparameter optimization for oil production is shown in Table 5.1, whereas the results for gas production is shown in Table 5.2.

Table 5.1: Hyperparameter Optimization for Oil Production

Imputation Method	Model Used	Obtained Hyperparameters	Validation Set RMSE / m3
Forward Filling Imputation	Gradient Boosting	Learning rate : **0.29268132226068777** Maximum depth : **10.0** Number of trees: **27.0**	138
	Random Forest	Maximum depth : **28** Min_samples_split : **9** Number of trees : **120**	128
Median Imputation	Gradient Boosting	Learning rate : **0.19885221540658993** Maximum depth : **11** Number of trees: **41**	173
	Random Forest	Maximum depth : **20** Min_samples_split : **3** Number of trees : **25**	181
Self-supervised Imputation	Gradient Boosting	Learning rate : **0.17000031038373792** Maximum depth : **11** Number of trees: **40**	156

	Random Forest	Maximum depth : **25** Min_samples_split : **2** Number of trees : **34**	160

Table 5.2: Hyperparameter Optimization for Gas Production

Imputation Method	Model Used	Obtained Hyperparameters	Validation Set RMSE / m3
Forward Filling Imputation	Gradient Boosting	Learning rate : **0.20604737829083644** Maximum depth : **20** Number of trees: **50**	64618
	Random Forest	Maximum depth : **50** Min_samples_split : **9** Number of trees : **120**	55816
Median Imputation	Gradient Boosting	Learning rate : **0.19965130013395532** Maximum depth : **11** Number of trees: **41**	52815
	Random Forest	Maximum depth : **24** Min_samples_split : **2** Number of trees : **100**	52793
Self-supervised Imputation	Gradient Boosting	Learning rate : **0.13470450722751402** Maximum depth : **9** Number of trees: **119**	59339
	Random Forest	Maximum depth : **80** Min_samples_split : **2** Number of trees : **100**	54153

5.2.2 Model Performance and Evaluation

In this section, the performance of each model will be shown and the model will be evaluated on the test dataset using RMSE as the primary performance metric and R-Squared as the secondary performance metric

5.2.2.1 Gradient Boosting

For the gradient boosting model, the model was evaluated by plotting the performance of the training set against the testing set. The graphs are shown in Figure 5.3, Figure 5.4 and Figure 5.5.

Forward Filling Imputation

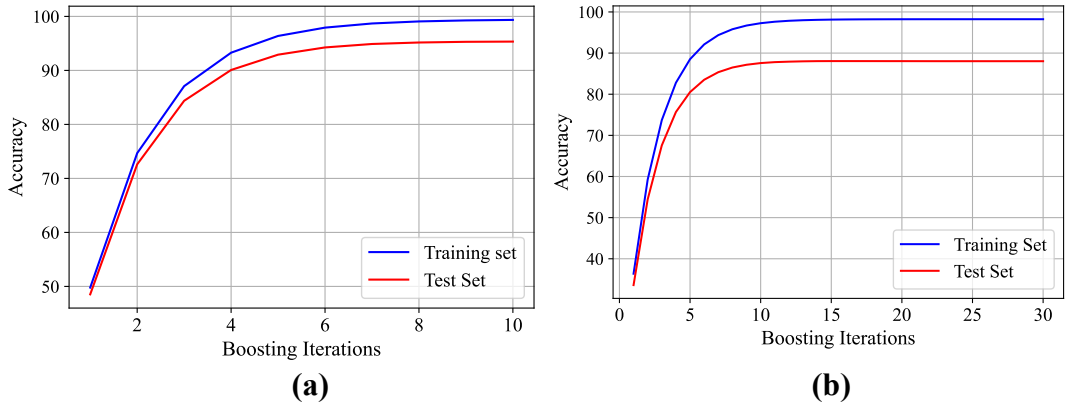


Figure 5.3: Training Curve for Gradient Boosting Model with Forward Filling Imputation Dataset; (a) oil and (b) gas

Median Imputation

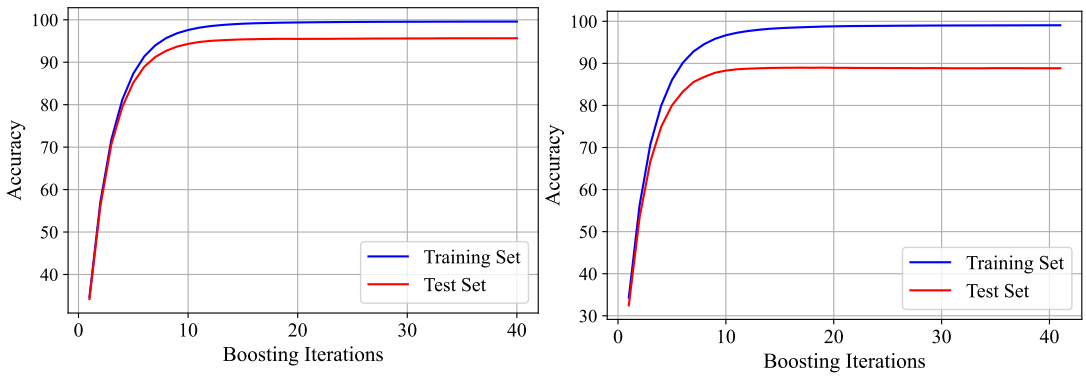


Figure 5.4: Training Curve for Gradient Boosting Model with Median Imputation Dataset; (a) oil and (b) gas

Self-Supervised Imputation

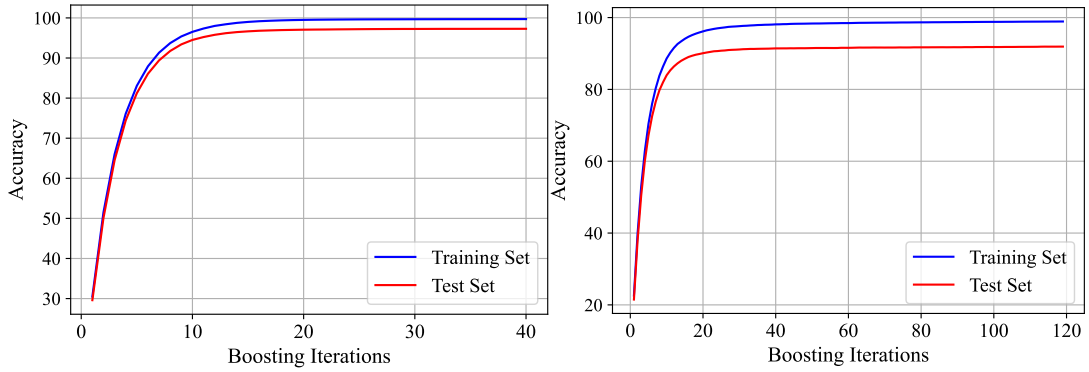


Figure 5.5: Training Curve for Gradient Boosting Model with Self-Supervised Imputation Dataset; (a) oil and (b) gas

Figure 5.3, Figure 5.4, and Figure 5.5 show that for the oil production, the model with the self-supervised imputation dataset gave the best result. It had the highest accuracy and the gap between the training set and the testing set is the smallest when the self-supervised imputation dataset was used. However, the model with forward filling imputation dataset gave the worst result.

For gas production, the model with the self-supervised imputation dataset also gave the best result. It also had the highest accuracy and the gap between the training and testing set is the smallest with the self-supervised imputation dataset. Similar to the oil production model, the model with forward filling imputation dataset performed the worse.

5.2.2.2 Random Forest

For the random forest model, the model was evaluated by plotting the predicted values against the actual values. The diagrams are shown in Figure 5.6, Figure 5.7, and Figure 5.8.

Forward Filling Imputation

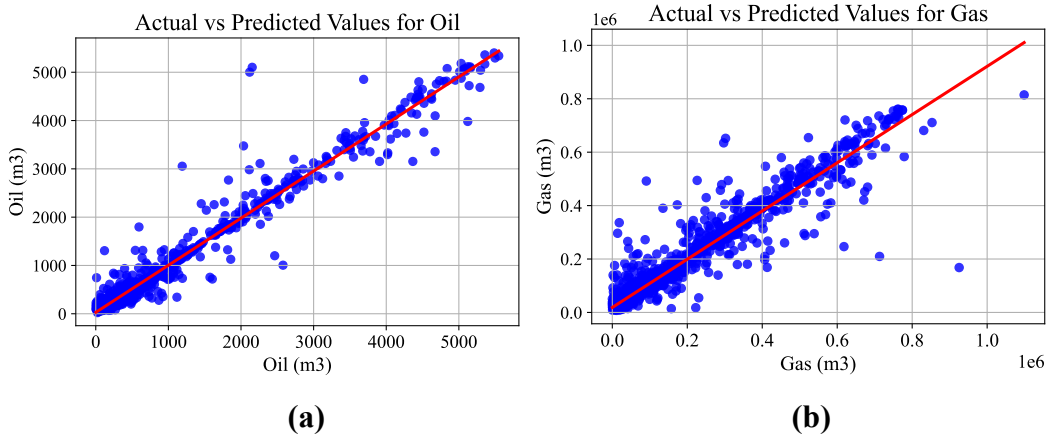


Figure 5.6: Actual vs Predicted Values for Random Forest Model with Forward Filling Imputation Dataset; (a) oil and (b) gas

Median Imputation

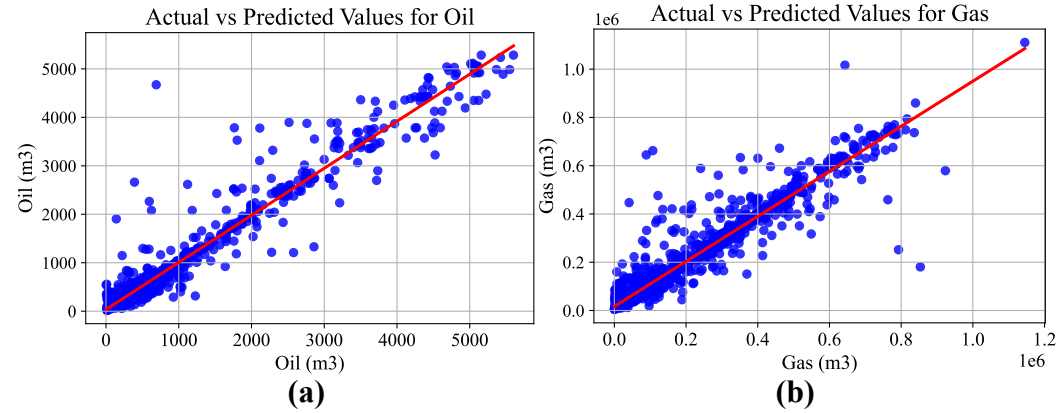
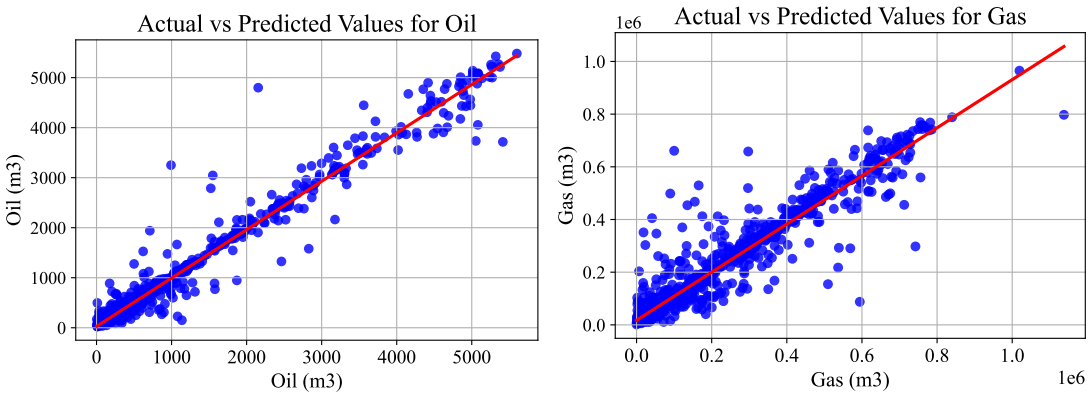


Figure 5.7: Actual vs Predicted Values for Random Forest Model with Median Imputation Dataset; (a) oil and (b) gas

Self-Supervised Imputation



(a) **(b)**

Figure 5.8: Actual vs Predicted Values for Random Forest Model with Self-Supervised Imputation Dataset; (a) oil and (b) gas

For oil production, it can be seen that Figure 5.6, Figure 5.7, and Figure 5.8 are quite similar as the models predicted values that are similar to the actual values. However, it can be seen that the blue dots are somewhat more scattered for the model with the median imputation dataset. This indicates that this model performed the worst.

For gas production, Figure 5.6, Figure 5.7, and Figure 5.8 are quite similar as well. From this it can be seen that all the models predicted values that are similar to the actual values. However, the blue dots are slightly more scattered for the model with the median imputation dataset. Therefore, for gas production the model with the median imputation dataset performed the worse.

5.2.2.3 Inference time

This section describes the inference time for each model to predict oil and gas production. The inference time was calculated 10 thousand times on a laptop with RAM 16 GB and Intel Core i5. Table 5.3, Table 5.4, and Table 5.5 show the mean inference time and standard deviation for each model.

Forward Filling Imputation

Table 5.3: Inference Time for Models with Forward Filling Imputation Dataset; GB denotes gradient boosting, RF denotes random forest

	GB Oil	GB Gas	RF Oil	RF Gas
Mean Inference Time / sec	0.00449	0.01744	0.0105	0.0415
Standard Deviation / sec	0.000488	0.00147	0.0017	0.00595

Median Imputation

Table 5.4: Inference Time for Models with Median Imputation Dataset; GB denotes gradient boosting, RF denotes random forest

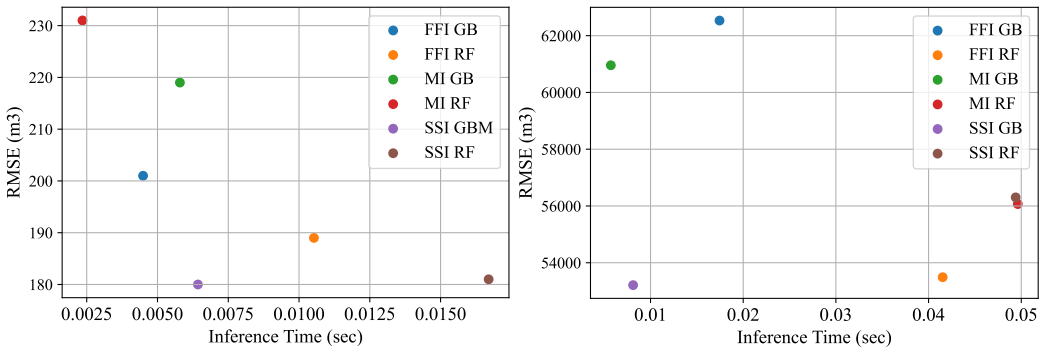
	GBM Oil	GBM Gas	RF Oil	RF Gas
Mean Inference Time / sec	0.00579	0.00571	0.00235	0.0496
Standard Deviation / sec	0.000666	0.000885	0.000578	0.00843

Self-Supervised Imputation

Table 5.5: Inference Time for Models with Self-Supervised Imputation Dataset; GB denotes gradient boosting, RF denotes random forest

	GB Oil	GB Gas	RF Oil	RF Gas
Mean Inference Time / sec	0.00642	0.00812	0.01669	0.0494
Standard Deviation / sec	0.000830	0.000647	0.00205	0.00600

Figure 5.9 shows the inference time for each model against the model's RMSE values for oil and gas production respectively.



(a) **(b)**

Figure 5.9: Inference Time for Models; FFI denotes forward filling imputation, MI denotes median imputation, SSI denotes self-supervised imputation, GB denotes gradient boosting, RF denotes random forest

For oil production, Figure 5.9 shows that the random forest model with the median imputation dataset performed the fastest while the random forest model with the self-supervised imputation dataset took the longest to predict the oil production value. The models with the highest RMSE values is the random forest model with median imputation dataset. On the other hand, the model with the lowest RMSE value is the gradient boosting model with the self-supervised imputation dataset. From Figure 5.9, it can be inferred that the gradient boosting model with the self-supervised imputation dataset performed the best. There are two reasons why this model is deemed as the best performing model based on Figure 5.9. First of all, although this model is not as fast as the random forest model with the median imputation dataset, it is still one of the fastest performing models. In addition to being fast, it also has the lowest RMSE score, meaning it is more accurate than the other models. Therefore, the gradient boosting model with the self-supervised imputation dataset is the best model for oil production.

For gas production, Figure 5.9 shows that the gradient boosting model with the median imputation performs the fastest while the random forest model with the median

imputation dataset took the longest to predict the gas production value. Figure 5.9 also showed that the gradient boosting model with the forward filling imputation dataset had the highest RMSE value. On the other hand, the gradient boosting model with the self-supervised imputation dataset had the lowest RMSE value. Similar to oil production, Figure 5.9 shows that the gradient boosting model with the self-supervised imputation dataset is the best performing model. The reasoning behind this observation is because the model has the lowest RMSE value while still performing fast. Therefore, the gradient boosting model with the self-supervised imputation dataset is the best model for gas production as well.

5.2.2.4 Model Evaluation

The result of the evaluation for oil production is shown in Table 5.6. On the other hand, the evaluation result for gas production is shown in Table 5.7.

Table 5.6: Evaluation for Oil Production Model

Imputation Method	Model Used	RMSE / m3	R-Squared
Forward Filling Imputation	Gradient Boosting	201	96.3
	Random Forest	189	96.7
Median Imputation	Gradient Boosting	219	95.6
	Random Forest	231	95.1
Self-supervised Imputation	Gradient Boosting	180	97.3
	Random Forest	181	97.2

Table 5.7: Evaluation for Gas Production Model

Imputation Method	Model Used	RMSE / m3	R-Squared
Forward Filling Imputation	Gradient Boosting	62,532	88.0
	Random Forest	53,490	91.2
Median Imputation	Gradient Boosting	60,957	88.8
	Random Forest	56,578	90.5
Self-supervised Imputation	Gradient Boosting	53,211	91.9
	Random Forest	56,367	90.9

From Table 5.6 and Table 5.7, it can be seen that for both oil and gas production, the gradient boosting model with the self-supervised imputation dataset performed the best.

In terms of oil production with the random forest model, the RMSE values indicate that the model with the self-supervised imputation dataset performed 22% better than the model with the median imputation dataset. Additionally, it performed 5% better than the model with the forward filling imputation dataset. On the other hand, the R-Squared values indicate that the model with the self-supervised imputation dataset performed nearly 1% better than the model with the forward filling dataset imputation and around 2% better than the model with the median imputation dataset.

In terms of oil production with the gradient boosting model, the RMSE values showed that the model with the self-supervised imputation dataset performed 18% better than the model with the median imputation dataset. It also performed 11% better than the model with the forward filling imputation dataset. In addition to this, the R-Squared values show that the model with the self-supervised imputation dataset performed 1%

better than the model with the forward filling dataset imputation and around 2% better than the model with the median imputation dataset.

In terms of gas production with the random forest model, the RMSE values denote that the model with the self-supervised imputation dataset performed 1% better than the model with the median imputation dataset. It also performed 5% worse than the model with the forward filling imputation dataset. Additionally, the R-Squared values denote that the model with the self-supervised imputation dataset performed nearly 1% better than the model with the median imputation dataset and almost 1% worse than the model with the forward filling imputation dataset.

In terms of gas production with the gradient boosting model, the RMSE values convey that the model with the self-supervised imputation dataset performed 13% better than the model with the median imputation dataset. Furthermore, it performed 15% better than the model with the forward filling imputation dataset. Additionally, the R-Squared values convey that the model with the self-supervised imputation dataset performed 0.8% better than the model with the forward filling dataset imputation and around 3% better than the model with the median imputation dataset.

5.3 Server-Side Oil and Gas Production Data Prediction Application

As has been mentioned in Section 4.3.1, the author will save the best performing models as pickle files. These pickle files will be reloaded into a Python program and the FastAPI framework will be used to create the API for the server-side prediction application. This program will parse user input, validate the user input, calculate the oil and gas prediction value, and store this output so that it will be returned to the user.

Afterwards, the backend application will be packaged as a container using Docker. The oil and gas prediction service needs to be integrated with the existing backend application that has been deployed in an AWS EC2 service. Besides the Python program that contains the saved models, a file containing the configurations to port the oil and gas prediction service to the existing backend application is needed. These configurations include a list of library dependencies, a list of environment variables, a list of mapped networking ports, and also a list of mapped filesystem. The configurations can be located in a Python requirement text file and the Dockerfile. The prediction service will then be defined in the docker-compose.yml file which is the Docker compose configuration file. The docker-compose.yml file will contain the existing services as well as the oil and gas prediction service developed by the author. In the docker-compose.yml file, multiple Docker containers can be deployed at the same time.

After the oil and gas prediction service has been integrated with the existing services, the client-side application will use the API to communicate with the backend service. There will be a total of four endpoints created by the author, two of these endpoints were made for predicting oil production value while the other two endpoints were made for predicting gas production value. There are two methods that the users can use to predict the values, namely singular data prediction and excel file prediction.

5.3.1 Singular Data Prediction

These endpoints make the user input the feature values manually. Additionally, based on the request from the client, the user will also be able to choose the unit for the

temperature values. The user could make the temperature in the form of Celsius or Fahrenheit. Figure 5.10 shows a sample response from the oil production endpoint.

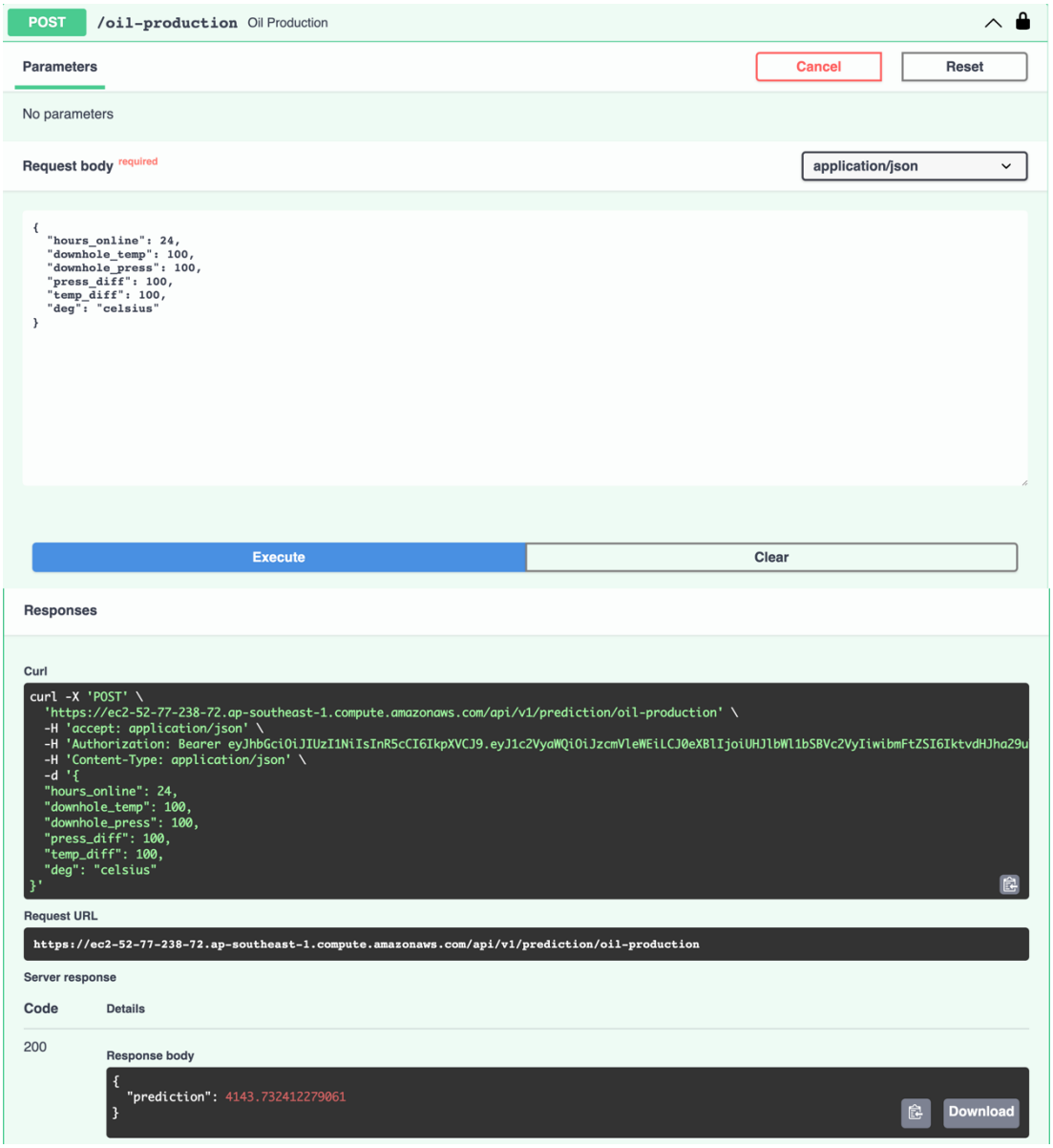


Figure 5.10: Sample Response from Oil Production Endpoint

The *hours_online, downhole_temp, downhole_press, press_diff* and *temp_diff* are values that will be used by the machine learning model to predict the oil production value. On the other hand, deg is where the user can decide whether they wish the temperature values to be in Celsius or Fahrenheit. The gas production endpoint has the

same parameters as the oil production endpoint. In addition to this, as this prediction feature is only accessible by premium users, the status of the users will be checked beforehand. If the user is not a premium user, then it will return an error code, preventing the user from accessing the feature.

5.3.2 Excel File Prediction

These endpoints will accept an Excel file as an input and predict oil and gas production values from the file. The file should be uploaded in the file management section made by the author's teammate, therefore, the user would just have to select the file. The file has to follow the template provided to the premium user. If the file uploaded by the user differs from the template provided, then it will return an error code. Furthermore, the feature values should only consist of numerical values, or else it will return an error code. Additionally, there should not be any missing values in the excel file, or else it will return an error code. If all the conditions are met properly, then the endpoint will predict the oil and gas production values. Figure 5.11 shows a sample response from the oil production excel.

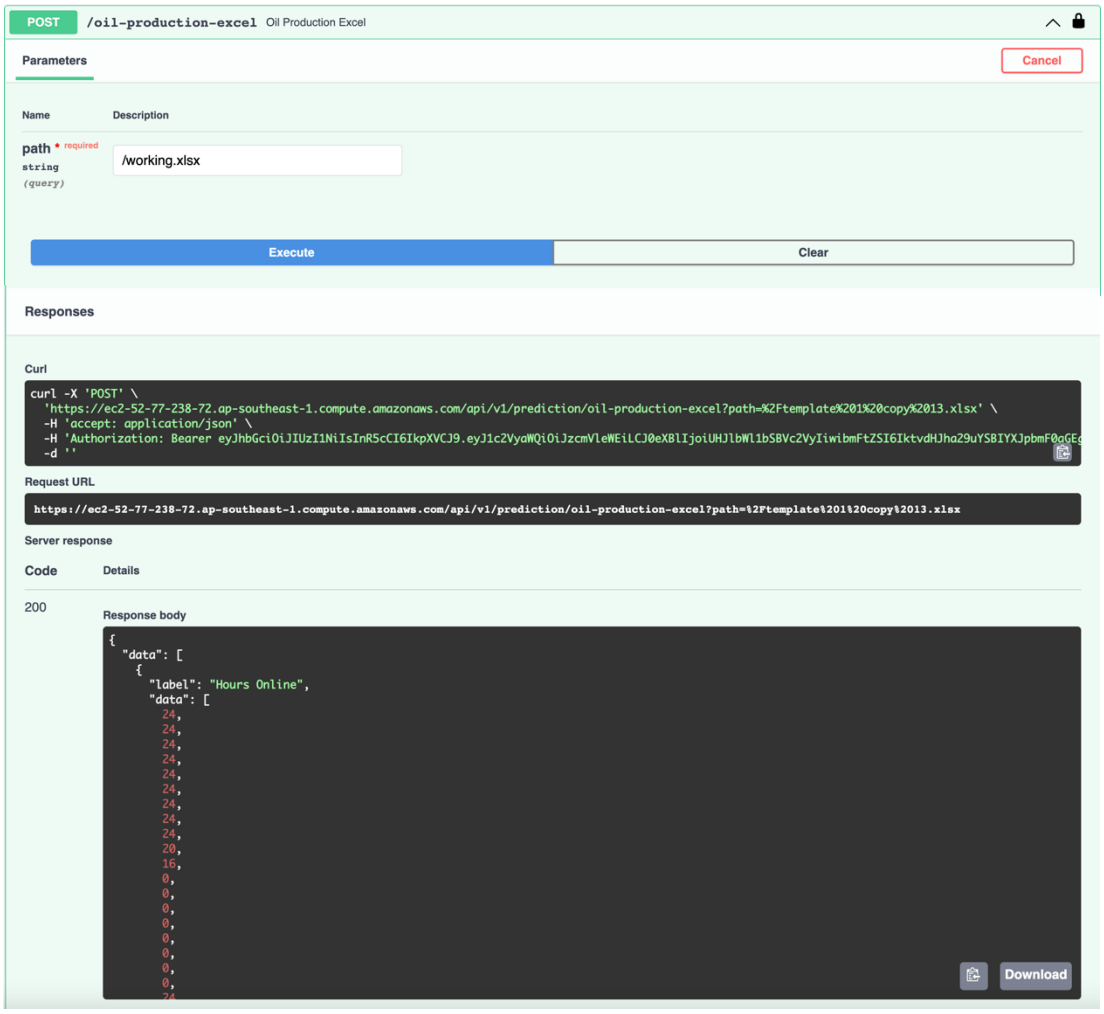


Figure 5.11: Sample Response from Oil Production Excel Endpoint

Table 5.8 contains the summary of the endpoints created by the author and Table 5.9 contains the list of error codes used for validation.

Table 5.8: Endpoints

Endpoints	Method	Description	Request	Response (200)
/oil-production	POST	Predicting oil production value based on singular imputation from user	{ "hours_online": float,	{ "prediction": float }

			"downhole_temp": float, "downhole_press": float, "press_diff": float, "temp_diff": float, "deg": string }	

/oil-production-excel	POST	Predicting oil production value(s) based on a excel file	{
			“path”: string
			
			}
				{
				“data”: array[
				“label”: string,
				“data”: array[]
				]
				
				}

/gas-production	POST	Predicting gas production value based on singular imputation from user	{
			"hours_online": float,
			
			"downhole_temp": float,
			
			"downhole_press": float,
			"press_diff": float,
				{
				"prediction": float
				}

			"temp_diff": float, "deg": string }	
/gas-production-excel	POST	Predicting gas production value(s) based on a excel file	{ "path": string }	{ "data": array["label": string, "data": array[]] }

Table 5.9: Error Codes for Endpoints

Endpoints	**Methods**	**Description**	**Error Code**	**Response (detail)**
/oil-production	POST	Oil production prediction based on singular data	400	Hours Online range should be 0 or above!
				Average Downhole Temperature / Deg C range should fall between 0 to 172!
				Pressure Difference of the Well / bar range should fall between 0 to 325!
				Temperature Difference of the Well / Deg C range should fall between 0 to 190!
				Average Downhole Pressure / bar range should fall between 0 to 308!
	POST	Oil production prediction	422	There should be exactly 5 columns filled in!

/oil-production/excel		based on excel file		Your file contains missing values, please fill it in!
				File does not follow the template
/gas-prediction	POST	Gas production prediction based on singular data	400	Hours Online range should be 0 or above!
				Average Downhole Temperature / Deg C range should fall between 0 to 172!
				Pressure Difference of the Well / bar range should fall between 0 to 325!
				Temperature Difference of the Well / Deg C range should fall between 0 to 190!
				Average Downhole Pressure / bar range should fall between 0 to 308!
/gas-prediction-excel	POST	Gas production prediction based on excel file	422	There should be exactly 5 columns filled in!
				Your file contains missing values, please fill it in!
				File does not follow the template

CHAPTER 6

DISCUSSION

This chapter will evaluate and clarify the key results obtained in this project and what it means for this thesis.

6.1 Discussion

The results showed that the models were capable of predicting oil and gas production values. Both the gradient boosting and random forest models could perform well for predicting oil and gas production values.

The results also showed that data imputation method has an effect on the model's performance. The performance of the model changes depending on the dataset used. For oil production, the models with the self-supervised imputation dataset performed better than the models with the median imputation dataset. However, the models with the forward filling imputation dataset performed worse than the median imputation dataset. From this, it can be inferred that the self-supervised imputation method improved the dataset better than the conventional imputation methods e.g. forward filling imputation and median imputation. Therefore, this method could be used in domains where open-sourced datasets are difficult to find such as in oil and gas production. Furthermore, as the model was able to predict oil and gas production values with high accuracy, the dataset can be used by other researchers to work in the field of oil and gas production prediction.

For oil and gas production, amongst all the models that the author evaluated, the models with the self-supervised imputation dataset outperformed the models with the other datasets. In addition to this, between the gradient boosting and random forest models, the gradient boosting model was shown to outperform the random forest model. The gradient boosting model with the self-supervised dataset had the lowest RMSE value and the highest R-squared value, making it the best model. Hence, as the model was able to predict oil and gas production successfully, the author's aim of developing an oil and gas prediction model for this thesis has been met.

The chosen gradient boosting model for oil production had an inference time of 0.00642 seconds whereas, the chosen gradient boosting model for gas production has an inference time of 0.00812 seconds. The inference time of the model should not be slow as the model would be used for the oil and gas predictive service for the VDR website application. It would not be ideal if it took a long time for the users to obtain the results. The results show that the inference time for both models is fast, this shows that model is suitable to be used in the website as it can predict quickly.

The client-side application was successfully able to connect to the API endpoints and show the prediction results for oil and gas production. The author tested the API using the Swagger UI that was automatically generated by FastAPI. In the Swagger UI, the author was able to test the API endpoints made directly on the browser. The author tested each endpoint to see if it is able to predict oil and gas production values. The testing showed that the endpoints could predict oil and gas production values based on an excel file and singular data. Therefore, the author's aim of developing an server-side oil and gas production data prediction application as a backend service and

creating an API for the backend prediction application for communication between the frontend and the backend prediction service has been met.

At the end of the project development, the author's team demonstrated the product to the product owner as well as the clients. As shown in Appendix Figure B.2, the clients were satisfied with the product and its functionalities in the product.

There were constraints faced in this thesis. The main constraint faced was the lack of good quality open source datasets for oil and gas production values. These types of data tend to be private company information, hence it was challenging to find good quality datasets. Furthermore, the dataset that was available contained a lot of missing values.

CHAPTER 7

CONCLUSION AND RECOMMENDATION

This chapter will conclude the results obtained in this thesis and provide recommendations that can be implemented for future works.

7.1 Conclusion

The results show that the tree-based models can perform well on tabular structured data as the gradient boosting and random forest models gave good results. In addition to this, the results showed that the model with the self-supervised imputation dataset gave better results than the other conventional imputation methods. For oil production, the best performing model was the gradient boosting model with the self-supervised imputation dataset. On the other hand, the worst performing model for oil production is the random forest model with the median imputation dataset. The results show the best oil production model had a RMSE value of 180 m^3 and an R-Squared value of 97.3%. On the other hand, the worst oil production model had a RMSE value of 231 m^3 and an R-Squared value of 95.1%. For gas production, the best performing model is also the gradient boosting model with the self-supervised imputation dataset, while the worst model is the gradient boosting model with the forward filling imputation dataset. Overall, the best model gave a RMSE value of 53,211 m^3 and an R-squared value of 91.9%. Whereas the worse model gave a RMSE value of 62,532 m^3 and an R-Squared value of 88.0%. The best oil and gas models were then used for the oil and gas prediction service. The author then developed a server-side oil and gas production data prediction application and integrated it with the existing backend services. The frontend application was able to show the prediction results from the predictive model.

7.2 Recommendation

The model was able to predict oil and gas production with high accuracy; however, there is still room for improvement. Although oil and gas production is mainly influenced by pressure and temperature, there are other factors that could play a role in its production. A factor that could affect production is the state of the equipment used to obtain the oil and gas. Perhaps a dataset that also contains the state of equipment, e.g. new or old, could improve the performance of the models.

Another method that could improve the performance of the model is by improving the self-supervised imputation method. For this thesis, the author only made use of a baseline gradient boosting and random forest model to fill in the missing values. Perhaps optimizing the models with the Bayesian optimization method could improve the dataset, hence improving the prediction results.

REFERENCES

[1] M. S. Vassiliou, Historical dictionary of the petroleum industry, 2018.

[2] International Association of Oil & Gas Producers, "Oil and gas in Everyday Life," International Association of Oil & Gas Producers, [Online]. Available: https://www.iogp.org/oil-natgas-in-everyday-life/. . [Accessed February 2022].

[3] W. P. Council, "Why are oil and gas important?," [Online]. Available: https://www.world-petroleum.org/edu/221-why-are-oil-and-gas-important#:~:text=Oil%20is%20one%20of%20the%20most%20important%20raw,about%20two%20million%20tonnes%20of%20oil%20and%20gas. . [Accessed March 2022].

[4] R. Ranggasari, "Oil and gas reserves potential in eastern Indonesia reaches 9.8bn barrels," Tempo, [Online]. Available: https://en.tempo.co/read/1536679/oil-and-gas-reserves-potential-in-eastern-indonesia-reaches-9-8bn-barrels#:~:text=Overall%2C%20the%20Energy%20Ministry%20recorded%20there%20are%2070,2.44%20billion%20barrels%20and%20gas%20of%2043.6%20TCF. .

[5] Indonesia Investment, "Crude Oil Indonesia," [Online]. Available: https://www.indonesia-investments.com/business/commodities/crude-oil/item267.. [Accessed February 2022].

[6] W. Kenton, "Virtual Data Room (VDR)," 23 June 2021. [Online]. Available: https://www.investopedia.com/terms/v/virtual-data-room-vdr.asp#:~:text=Virtual%20Data%20Rooms%2C%20or%20VDRs%2C%20exist%20as%20a,joint%20venture%20that%20requires%20access%20to%20shared%20data.. [Accessed 19 March 2022].

[7] Lynx, "License Pricing - Lynx Information Systems," Lynx Information System, [Online]. Available: http://www.lynxinfo.co.uk/download-pricing.html.

[8] Intviewer, "Intviewer - Fast Geoscience Visualization, Analysis & QC,," Intviewer, 02 August 2021. [Online]. Available: https://www.int.com/products/intviewer/#:~:text=INTViewer%20is%20a%20platform%20and%20application%20that%20allows,to%20a%20desktop%20or%20remotely%20via%20the%20cloud.. [Accessed 2022].

[9] INTViewer, "INTViewer. Geoscience Analysis and QC, Simplied.," INTViewer, [Online]. Available: https://www.int.com/products/intviewer/.

[10] Geodwipa Teknika Nusantara, "Geodwipa Teknika Nusantara," Geodwipa Teknika Nusantara, [Online]. Available: https://ptgtn.com/.

[11] A. A. Purwita, Interviewee, *Oil and Gas Companies.* [Interview]. 8 March 2022.

[12] CS Binus International , "VDRWEBAPP demo presentation to client 1st," 27 June 2022. [Online]. Available: https://www.youtube.com/watch?v=XK7GrYNpMiQ.

[13] IBM, "Data Science," IBM, 15 May 2020. [Online]. Available: https://www.ibm.com/cloud/learn/data-science-introduction. [Accessed April 2022].

[14] A. El-Banbi, A. Ahmed and E.-M. Ahmed , "Black Oils," in *PVT Property Correlations*, Elsevier, 2018, p. 147–182.

[15] S. Mokhatab, W. A. Poe and J. Y. Mak, "Natural Gas Fundamentals," in *Handbook of Natural Gas Transmission and Processing*, Elsevier, 2019.

[16] A. El-Banbi, A. Ahmed and E.-M. Ahmed, "Dry Gases," in *PVT Property Correlations*, Elsevier, 2018.

[17] T. Ahmed, "Reservoir-Fluid Properties," in *Reservoir-Fluid Properties* , Elsevier, 2010, pp. 29-135.

[18] I. Fetoui, "Hydrocarbon Phase Behavior," [Online]. Available: https://production-technology.org/category/pvt/.

[19] P. Z. a. K. H. C. Janiesch, "Machine learning and deep learning," *Electronic Markets,* vol. 31, no. 3, pp. 685-695, 2021.

[20] IBM Cloud Education, "What is machine learning," IBM, 15 July 2020. [Online]. Available: https://www.ibm.com/cloud/learn/machine-learning.. [Accessed 12 February 2022].

[21] I. Sydorenko, "What is dataset in Machine Learning," High quality data annotation for Machine Learning, 5 April 2021. [Online]. Available: https://labelyourdata.com/articles/what-is-dataset-in-machine-learning. [Accessed February 2022].

[22] D. N. Dimid, "Unsupervised learning algorithms cheat sheet,," 17 February 2022. [Online]. Available: https://towardsdatascience.com/unsupervised-learning-algorithms-cheat-sheet-d391a39de44a. . [Accessed 10 February 2022].

[23] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *Cambridge, MA,* 2020.

[24] W. X. P. C. M. C. a. S. D. Y. Gong, "Supervised Learning," *Machine learning techniques for multimedia,* p. 21–49., 2008.

[25] IBM Cloud Education, "What is Supervised Learning," IBM, 2022. [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning. . [Accessed 20 February 2022].

[26] A. M. J. A. V. M. A. A. L. a. A. A. S. A. R. van Loon, "Understanding supervised, unsupervised, and reinforcement learning," *Big Data Made Simple,* 2019.

[27] A. Jaiswal , A. R. Babu, M. Z. Zadeh, D. Banerjee and F. Makedon, "A Survey on Contrastive Self-Supervised Learning," in *PETRA*, 2020.

[28] z_ai, "Deep Learning for NLP: ANNs, RNNs and LSTMs explained!," 8 July 2019. [Online]. Available: https://towardsdatascience.com/deep-learning-for-nlp-anns-rnns-and-lstms-explained-95866c1db2e4. [Accessed May 2022].

[29] Javatpoint, "Regression vs. Classification in Machine Learning," Javatpoint, [Online]. Available: https://www.javatpoint.com/regression-vs-classification-in-machine-learning. [Accessed February 2022].

[30] aravindpai, "CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning download Share," 17 February 2020. [Online].

Available: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/. [Accessed 10 May 2022].

[31] GeeksforGeeks, "Understanding of LSTM Networks," GeeksforGeeks, 25 June 2021. [Online]. Available: https://www.geeksforgeeks.org/understanding-of-lstm-networks/. [Accessed 15 March 2022].

[32] "Classification vs. Regression Algorithms in Machine Learning," 16 February 2022. [Online]. Available: https://www.projectpro.io/article/classification-vs-regression-in-machine-learning/545#mcetoc_1fp6av4s69. [Accessed February 2022].

[33] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep Learning is not all you need," *Information Fusion,* vol. 81, pp. 84-90, 2022.

[34] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawlczyk and G. Kasneci, "Deep Neural Networks and Tabular Data: A Survey," 2021.

[35] Elzain, Hussam Eldin, Chung, Sang Yong, Senapathi, Venkatramanan, Sekar, Selvam, Lee, Seung Yeop, Roy, Priyadarsi D., Hassan, Amjed and Sabarathinam, Chidambaram, "Comparative study of machine learning models for evaluating groundwater vulnerability to nitrate contamination," *Ecotoxicology and Environmental Safety,* vol. 229, pp. 61-113, 2022.

[36] Scikit, "ScikitLearn," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html.

[37] A. Kumar, "Introduction to the Gradient Boosting Algorithm," 20 May 2020. [Online]. Available: https://medium.com/analytics-vidhya/introduction-to-the-gradient-boosting-algorithm-c25c653f826b. [Accessed April 2022].

[38] S. Rosenthal, "Data Imputation," in *The International Encyclopedia of Communication Research Methods*, Wiley, 2017, pp. 1-12.

[39] F. Malik, "Understanding value of correlations in data science projects," Medium, 10 June 2019. [Online]. Available: https://medium.com/fintechexplained/did-you-know-the-importance-of-finding-correlations-in-data-science-1fa3943debc2#:~:text=Correlation%20is%20a%20statistical%20measure.%20Correlation%20explains%20how,%28variables%29%20can%20be%20positively%20correlated%.

[40] A. McDonald, "Using the missingno Python library to Identify and Visualise Missing Data Prior to Machine Learning," 10 June 2021. [Online]. Available: https://towardsdatascience.com/using-the-missingno-python-library-to-identify-and-visualise-missing-data-prior-to-machine-learning-34c8c5b5f009. [Accessed March 2022].

[41] A. Swalin, "How to handle missing data," Medium, 19 March 2018. [Online]. Available: https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4.. [Accessed March 2022].

[42] H. Kang, "The prevention and handling of the missing data," in *Korean Journal of Anesthesiology*, vol. 64, 2013, p. 402.

[43] W. Badr, "“6 different ways to compensate for missing data (data imputation with examples)," 12 January 2019. [Online]. Available: https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-

values-data-imputation-with-examples-6022d9ca0779.. [Accessed February 2022].

[44] K. N, "Part-1 : Data Preparation Made Easy with python!!," Medium, 09 May 2020. [Online]. Available: https://medium.com/analytics-vidhya/part-1-data-preparation-made-easy-with-python-e2c024402327.. [Accessed 12 March 2022].

[45] Á. Fernández, J. R. Dorronsoro and J. Bella, "Supervised outlier detection for classification and regression," *Neurocomputing,* vol. 486, pp. 77-92, 2022.

[46] C. M. Salgado, C. Azevedo, H. Proença and S. M. Vieira, "Noise Versus Outliers," in *Secondary Analysis of Electronic Health Records*, Cham, Springer International Publishing, 2016, pp. 163-183.

[47] BBC Bitesize, "Types of correlation - scattergraphs - national 4 application of Maths Revision," BBC News, [Online]. Available: https://www.bbc.co.uk/bitesize/guides/zmt9q6f/revision/2.. [Accessed 20 February 2022].

[48] Nettleton, David, "Selection of Variables and Factor Derivation," in *Commercial Data Mining*, Elsevier, 2014, pp. 79-104.

[49] "Spearman correlation coefficient: Definition, formula and calculation with example," QuestionPro, 15 January 2020. [Online]. Available: https://www.questionpro.com/blog/spearmans-rank-coefficient-of-correlation/.. [Accessed March 2022].

[50] Swapnilbobe, "Spearman's correlation," 13 April 2021. [Online]. Available: https://medium.com/analytics-vidhya/spearmans-correlation-f34c094d99d8#:~:text=Here%2C%20we%20are%20calculating%20spearman%E2%80%99s%20correlation%20using%20the,of%20relationship%20between%20ranks%20of%20two%20individual%20features.. [Accessed March 2022].

[51] V. Kumar and S. Minz, "Feature Selection: A literature Review," *Smart Computing Review,* vol. 4, pp. 1-19, 2014.

[52] K. Menon, "Feature selection in machine learning," Simplilearn, 16 September 2021. [Online]. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/feature-selection-in-machine-learning#what_is_feature_selection.. [Accessed March 2022].

[53] Javatpoint, "Hyperparameters in Machine Learning," [Online]. Available: https://www.javatpoint.com/hyperparameters-in-machine-learning. [Accessed 2 April 2022].

[54] M. Feurer and F. Hutter, "Hyperparameter Optimization," in *Automated Machine Learning*, Springer, 2019, pp. 3-35.

[55] M. Dei, "Hyperparameter Tuning Explained — Tuning Phases, Tuning Methods, Bayesian Optimization, and Sample Code!," 13 December 2019. [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-explained-d0ebb2ba1d35. [Accessed 1 April 2022].

[56] W. Koehrsen, "Automated Machine Learning Hyperparameter Tuning in Python," 3 July 2018. [Online]. Available: https://towardsdatascience.com/automated-machine-learning-

hyperparameter-tuning-in-python-dfda59b72f8a#8811. [Accessed April 2022].

[57] A. Chugh, "MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?," 8 December 2020. [Online]. Available: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e.

[58] Zhao, Yunwei and Wan, Xin, "The Design of Embedded Web System based on REST Architecture," in *IEEE*, 2019.

[59] R. Naushad, "Comparison of FastAPI and Flask. Simple Explanation!," 24 July 2020. [Online]. Available: https://medium.datadriveninvestor.com/comparison-of-fastapi-and-flask-simply-explanation-c8c075f6aa80. [Accessed 30 June 2022].

[60] Docker, "Docker," [Online]. Available: https://docs.docker.com/.

[61] GeeksforGeeks, "Why Should You Use Docker – 7 Major Reasons!," 26 April 2021. [Online]. Available: https://www.geeksforgeeks.org/why-should-you-use-docker-7-major-reasons/. [Accessed 12 June 2022].

[62] L. Li, "The future of academic libraries in the digital age," *Chandos Digital Information Review,* pp. 253-268, 2013.

[63] S. Bhosale, A. Sheth and H. Kadam , "Research Paper on Cloud Computing," *CONTEMPORARY RESEARCH IN INDIA,* 2021.

[64] G. Gurung, R. Shah and D. P. Jaiswal, "Software development life cycle models-A comparative study," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* pp. 33-27, 2020.

[65] A. Mishra and D. Dubey, "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology,* vol. 1, no. 5, pp. 1-6, 2013.

[66] J. Shah, "A Comparative Study of Software Development Life Cycle Models".

[67] P. Pedamkar, "What is SDLC: Different phases and models of SDLC," *EDUCBA,* 2022.

[68] "Principles behind the Agile Manifesto," Agile, [Online]. Available: https://agilemanifesto.org/principles.html..

[69] M. Thakur, "Scrum JIRA," [Online]. Available: https://www.educba.com/scrum-jira/. [Accessed July 2022].

[70] M. Taylor, "Machine Learning in the Oil and Gas Industry," 21 January 2021. [Online]. Available: https://newengineer.com/blog/machine-learning-in-the-oil-and-gas-industry-1507752. [Accessed February 2022].

[71] C. Xie, L. Chao, Y. Qin, J. Cao and Y. Li, "Using a stochastic forest prediction model to predict the hazardous gas concentration in a one-way roadway," *AIP Advances,* vol. 10, no. 11, 2020.

[72] R. Sharma, "Using Facebook Prophet for Forecasting Natural Gas Production," Medium, 13 March 2021. [Online]. Available: https://medium.com/mlearning-ai/forecast-using-prophet-canadian-natural-gas-production-dataset-b1f9c57548d8.

[73] S. Goled, "Why Are People Bashing Facebook Prophet," 18 October 2021. [Online]. Available: https://analyticsindiamag.com/why-are-people-bashing-facebook-prophet/. [Accessed March 2022].

[74] L. Menculini, A. Marini, M. Proietti, A. Garinei, A. Bozza, C. Moretti and M. Marconi, "Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices," *Forecasting,* vol. 3, no. 3, pp. 644-662, 2021.

[75] J. Chahar, "Prediction of Oil Production by applying Machine Learning on Volve Field Production Data.," 17 December 2020. [Online]. Available: https://www.linkedin.com/pulse/prediction-oil-production-applying-machine-learning-volve-chahar/.

[76] Opengenus.org, "Advantages and Disadvantages of Linear Regression," [Online]. Available: https://iq.opengenus.org/advantages-and-disadvantages-of-linear-regression/. [Accessed 12 February 2022].

[77] A. Pant, "Introduction to Linear Regression and Polynomial Regression," 13 January 2019. [Online]. Available: https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb#:~:text=Disadvantages%20of%20using%20Polynomial%20Regression%20The%20presence%20of,analysis.%20These%20are%20too%20sensitive%20to%20the%20outliers.. [Accessed 12 March 2022].

[78] EDUCBA, "Difference Between Random forest vs Gradient boosting," [Online]. Available: https://www.educba.com/random-forest-vs-gradient-boosting/. [Accessed 24 March 2022].

[79] S. Dash, "Gradient Boosting – A Concise Introduction from Scratch," 21 October 2020. [Online]. Available: https://www.machinelearningplus.com/machine-learning/gradient-boosting/#:~:text=Using%20a%20low%20learning%20rate%20can%20dramatically%20improve,iterations%20to%20converge%20to%20a%20final%20loss%20value.. [Accessed April 2022].

[80] R. Meinert, "Optimizing Hyperparameters in Random Forest Classification," 6 June 2019. [Online]. Available: https://towardsdatascience.com/optimizing-hyperparameters-in-random-forest-classification-ec7741f9d3f6#:~:text=Hyperparameter%20tuning%20can%20be%20advantageous%20in%20creating%20a,values%20can%20be%20very%20time%20consuming%20as%20well.. [Accessed April 2022].

APPENDICES

Appendix A

I. Heatmap to show the correlation of missing values in the Volve dataset

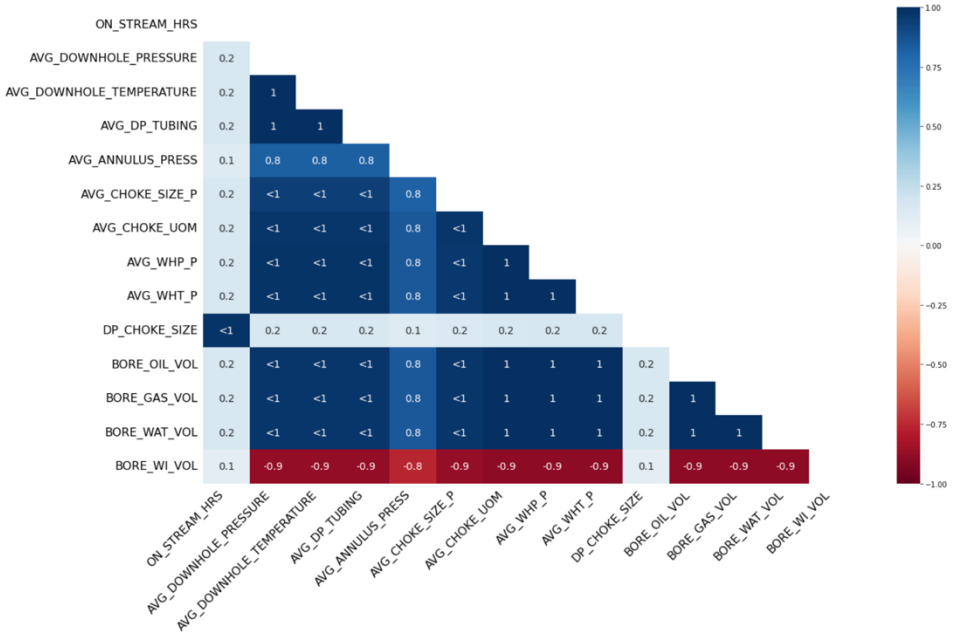


Figure A.1: Missing value correlation in Volve

II. Heatmap to show the correlation of missing values in the Kyle Master

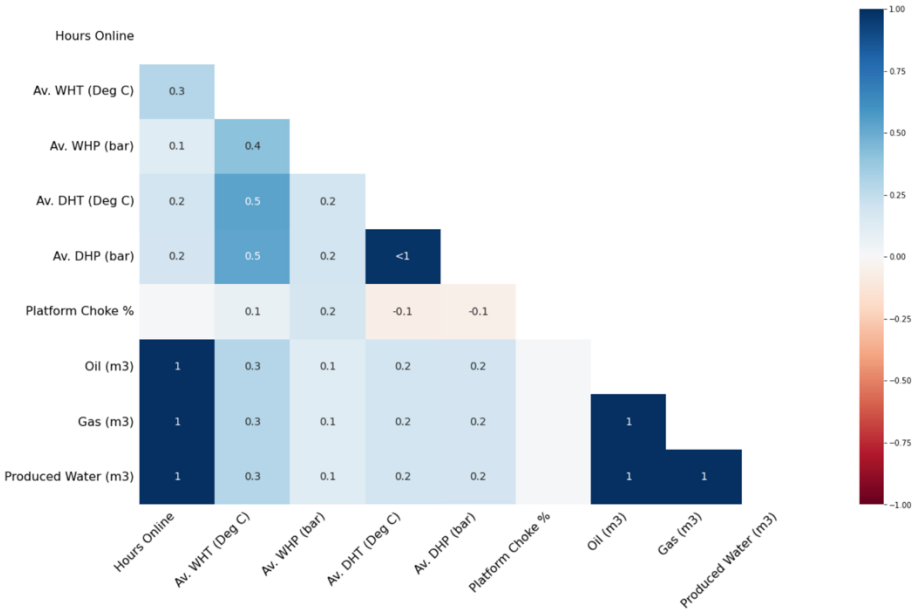


Figure A.2: Missing value correlation in Kyle Master

III. Feature correlation in Volve and Kyle Master datasets

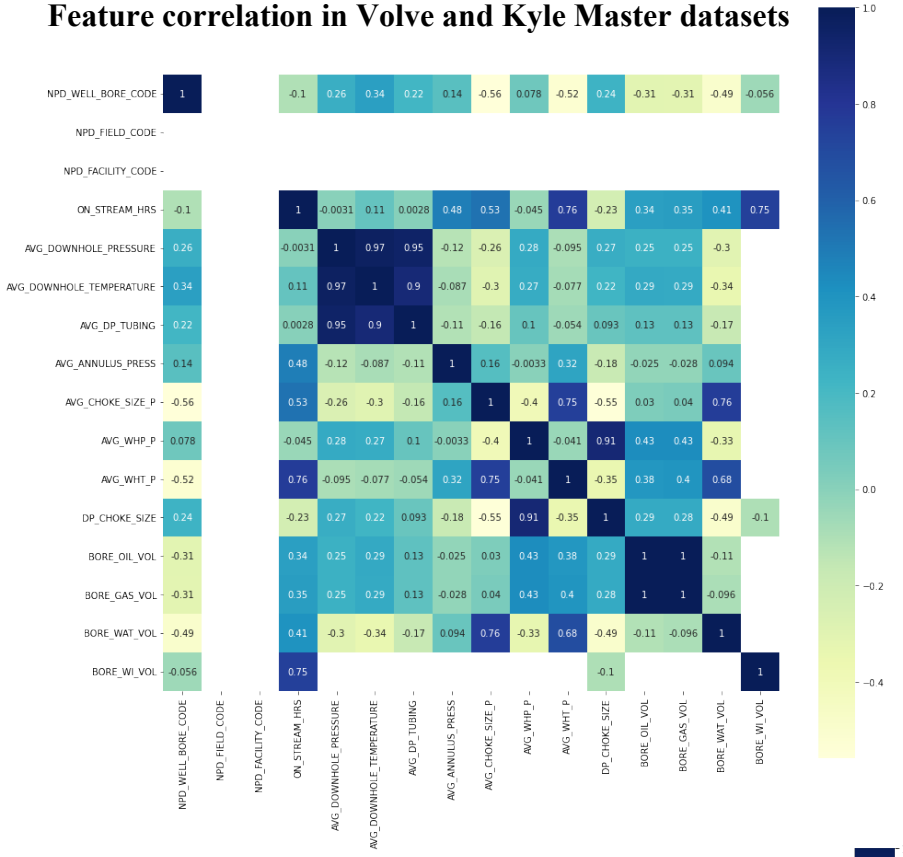


Figure A.3: Feature correlation for Volve dataset

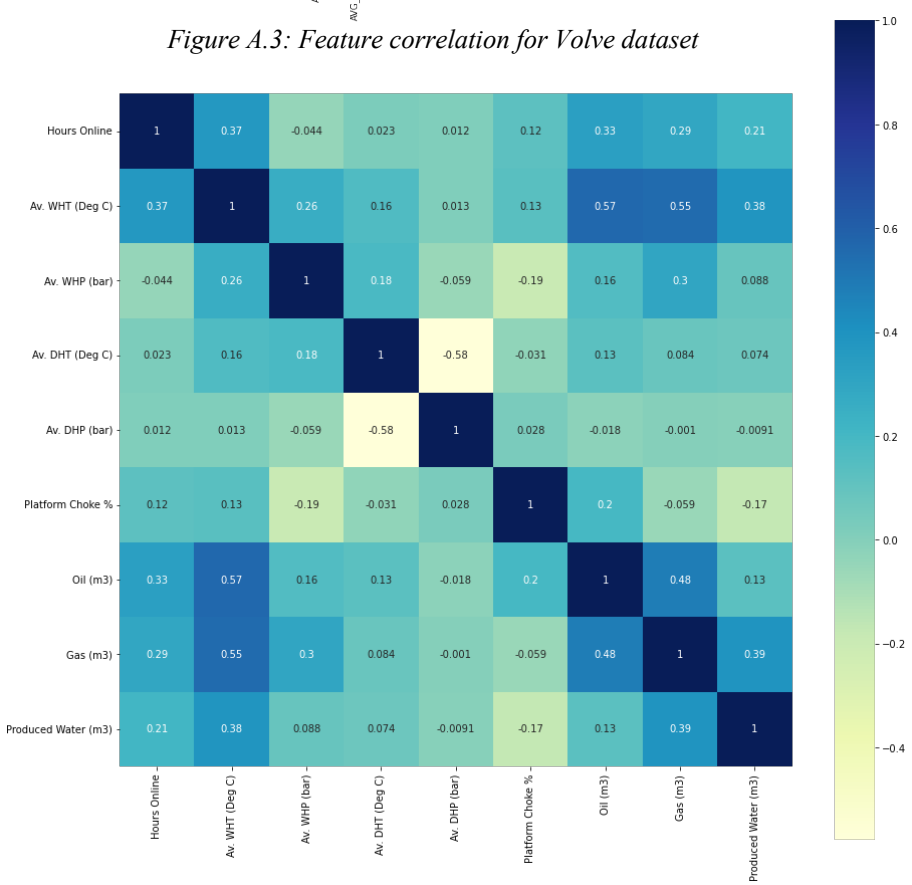
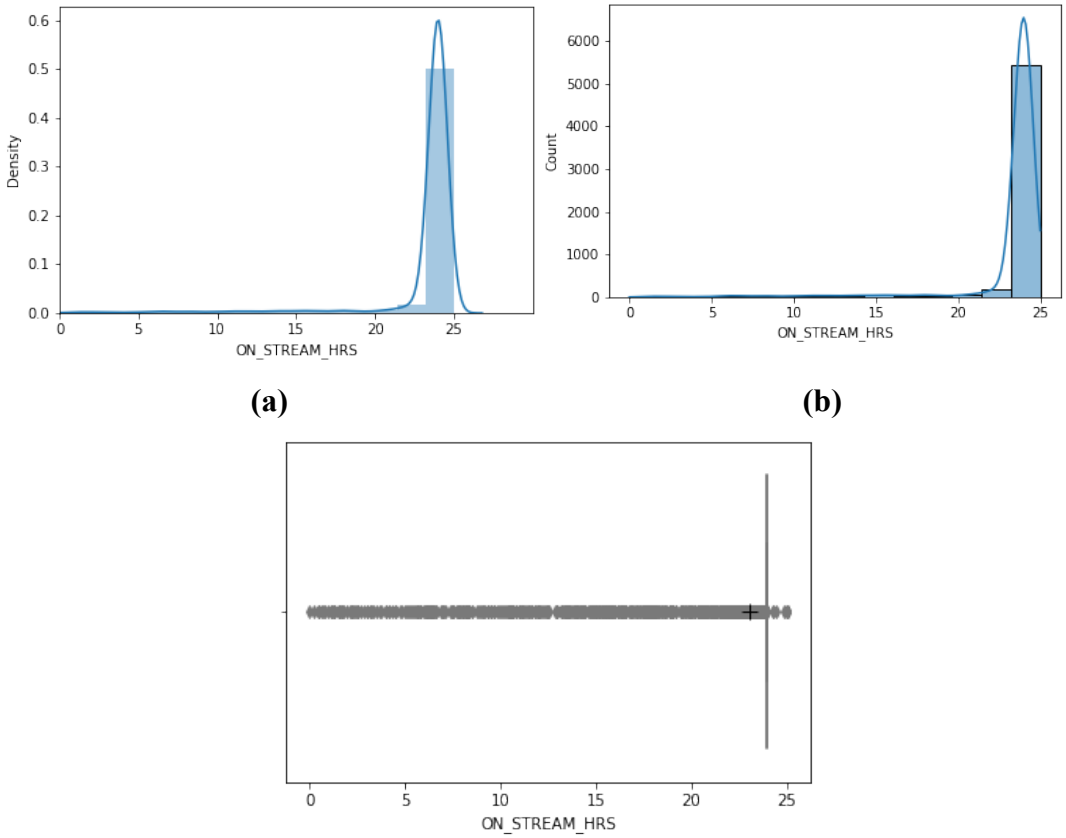


Figure A.4: Feature correlation for Kyle Master dataset

IV. Feature statistics in Volve dataset

ON_STREAM_HRS

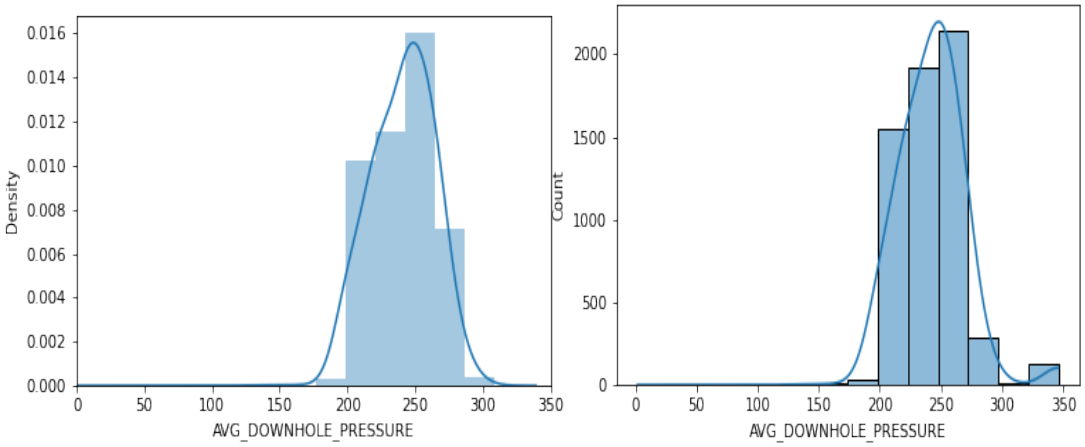


(a) **(b)**

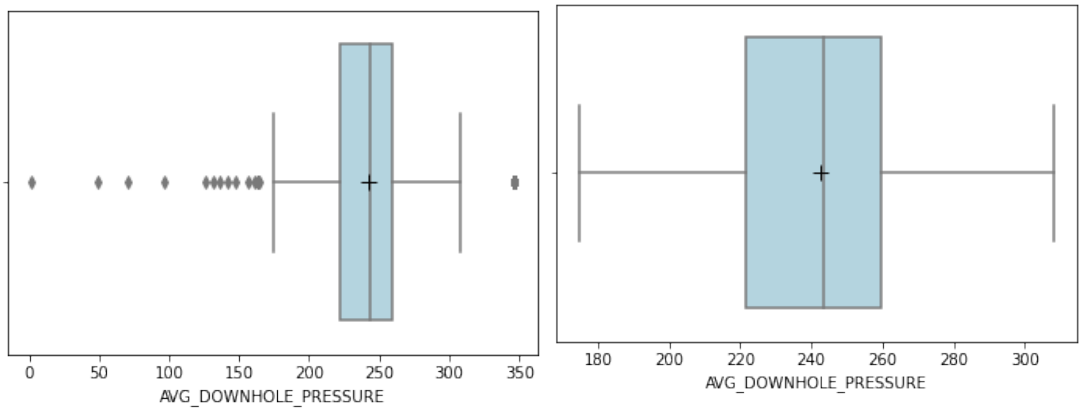
(c)

Figure A.5: (a) Kernel Density Estimation plot for ON_STREAM_HRS (b) Histogram for ON_STREAM_HRS (c) Boxplot for ON_STREAM_HRS

AVG_DOWNHOLE_PRESSURE



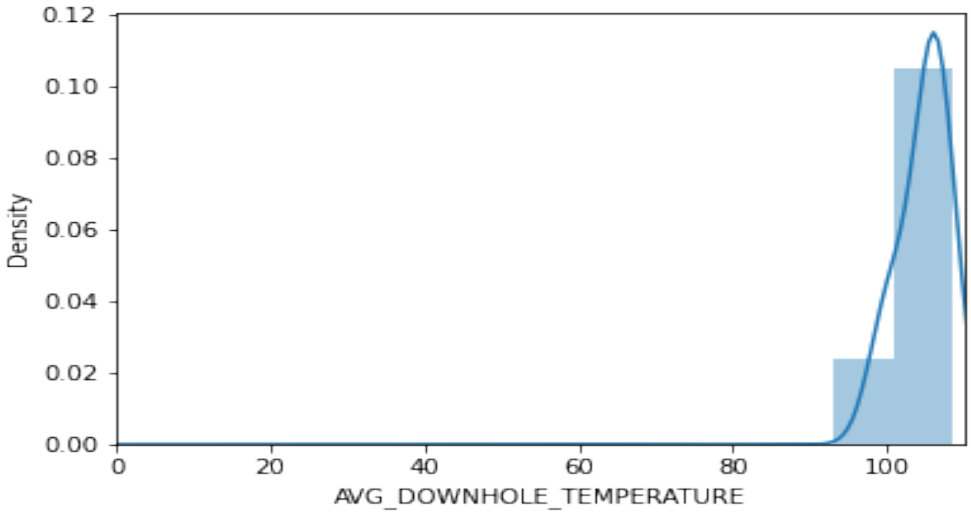
(a) **(b)**



(c) **(d)**

Figure A.6: (a) Kernel Density Estimation plot for AVG_DOWNHOLE_PRESSURE (b) Histogram for AVG_DOWNHOLE_PRESSURE (c) Boxplot for AVG_DOWNHOLE_PRESSURE (d) Boxplot without outliers for AVG_DOWNHOLE_PRESSURE

AVG_DOWNHOLE_TEMPERATURE



(a)

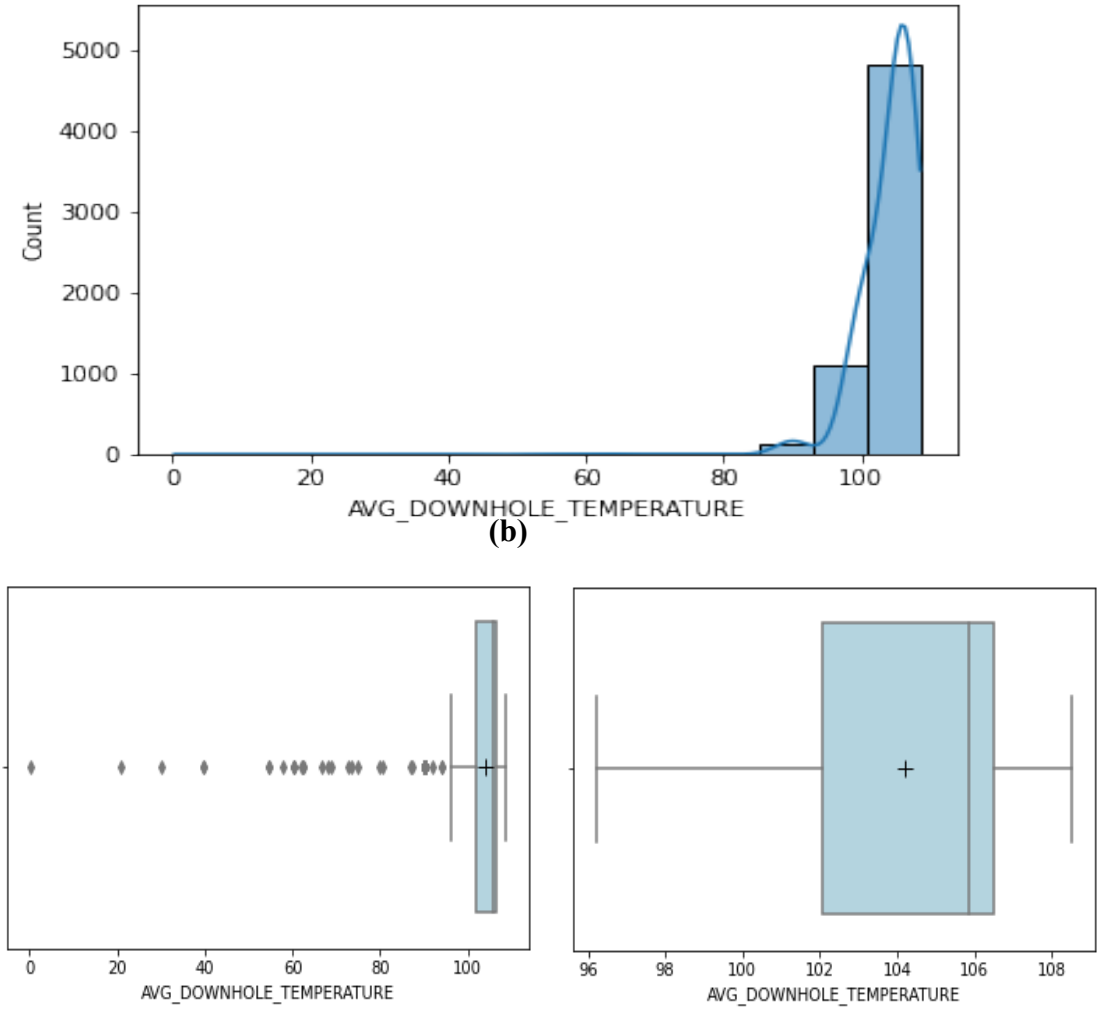
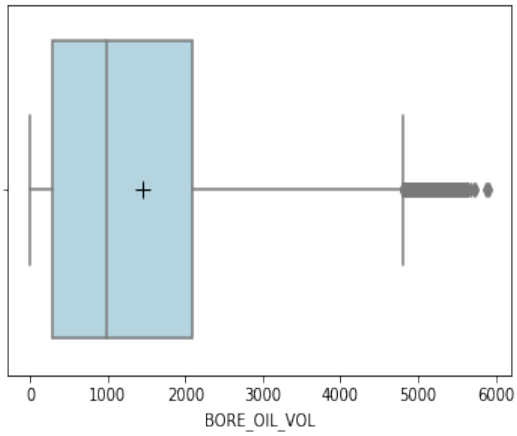
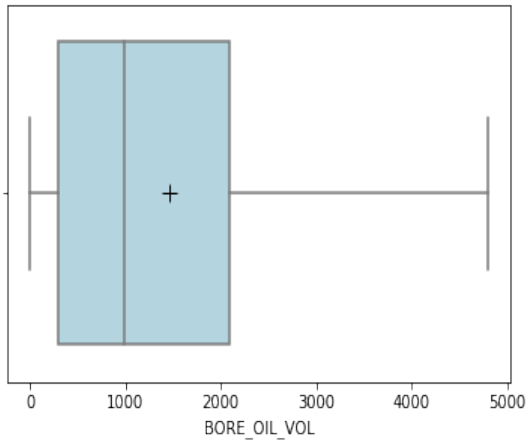


Figure A.7: (a) Kernel Density Estimation plot for AVG_DOWNHOLE_TEMPERATURE (b) Histogram for AVG_DOWNHOLE_ TEMPERATURE (c) Boxplot for AVG_DOWNHOLE_ TEMPERATURE (d) Boxplot without outliers for AVG_DOWNHOLE_ TEMPERATURE

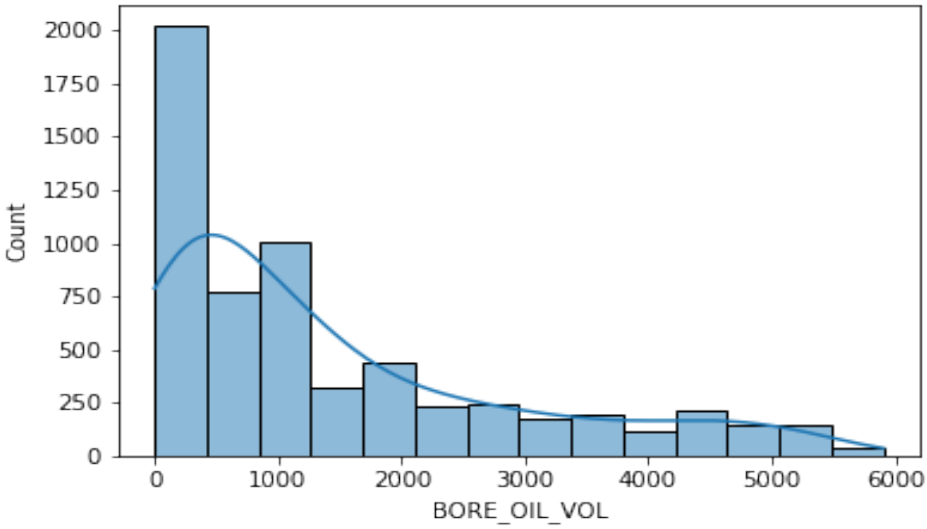
BORE_OIL_VOL



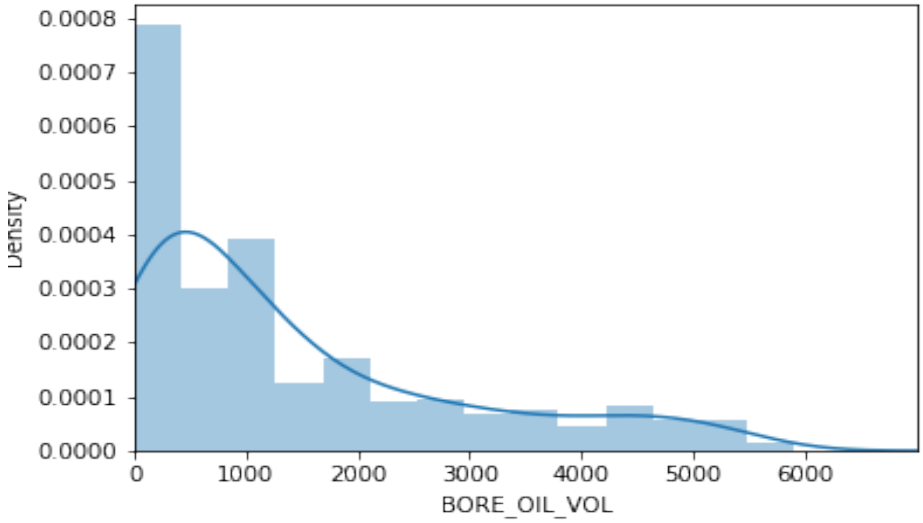
(a)



(b)



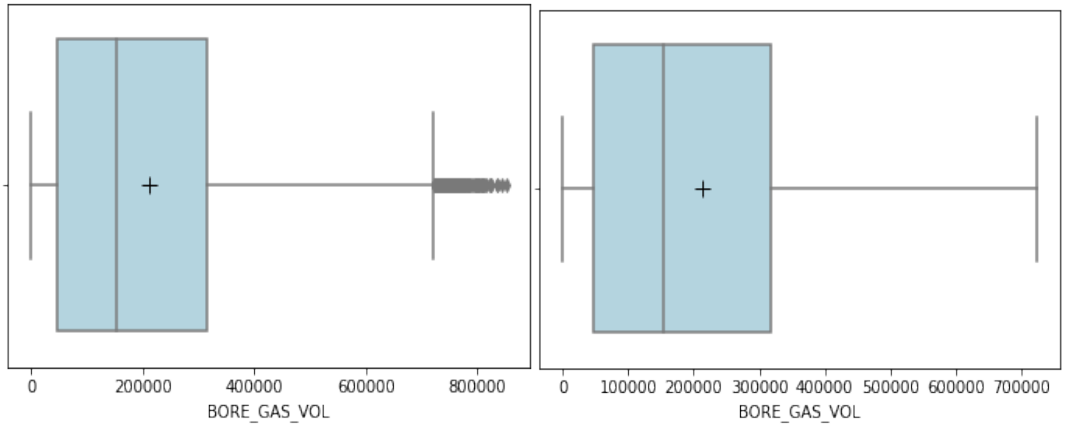
(c)



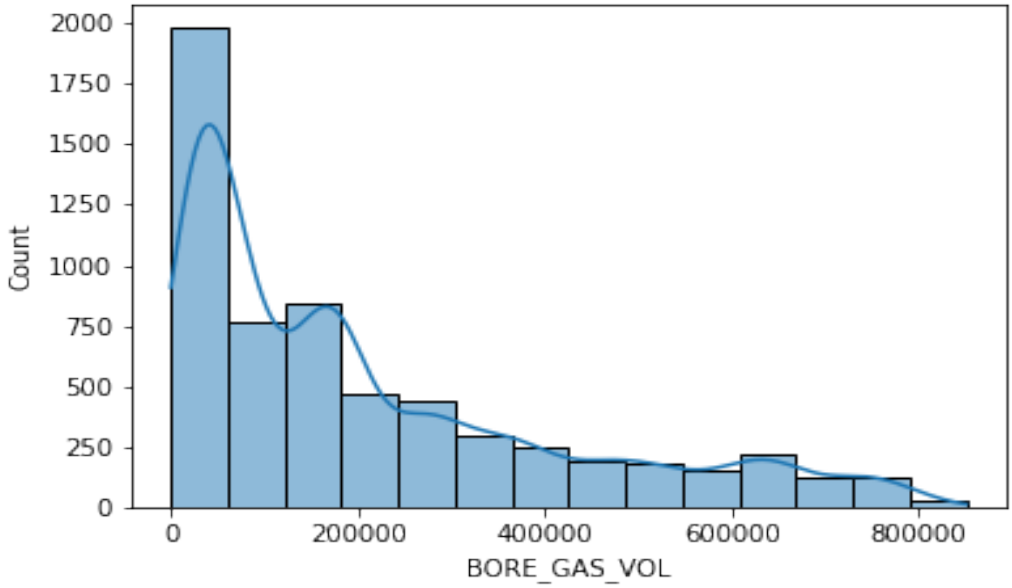
(d)

Figure A.8: (a) Boxplot for BORE_OIL_VOL (b) Boxplot without outliers for BORE_OIL_VOL (c) Histogram for BORE_OIL_VOL (d) Kernel Density Estimation plot for BORE_OIL_VOL

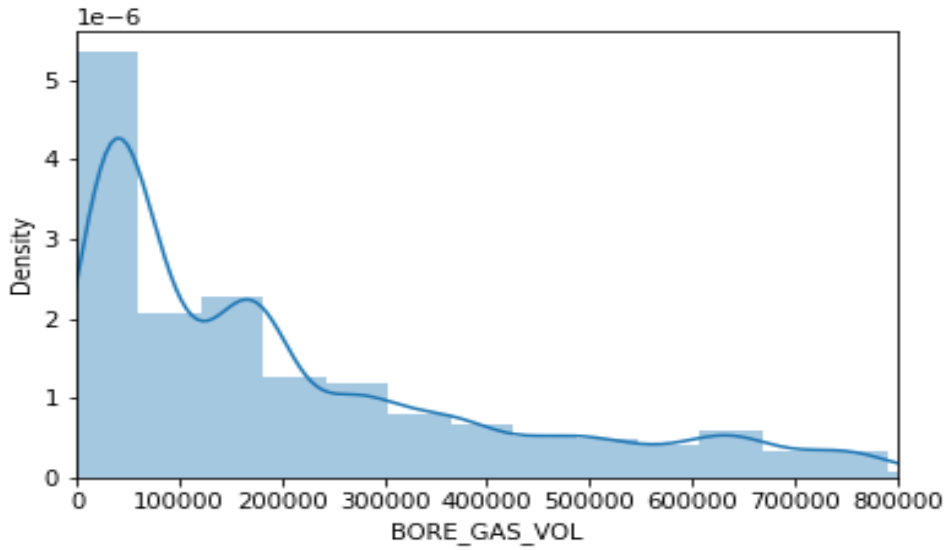
BORE_GAS_VOL



(a) **(b)**



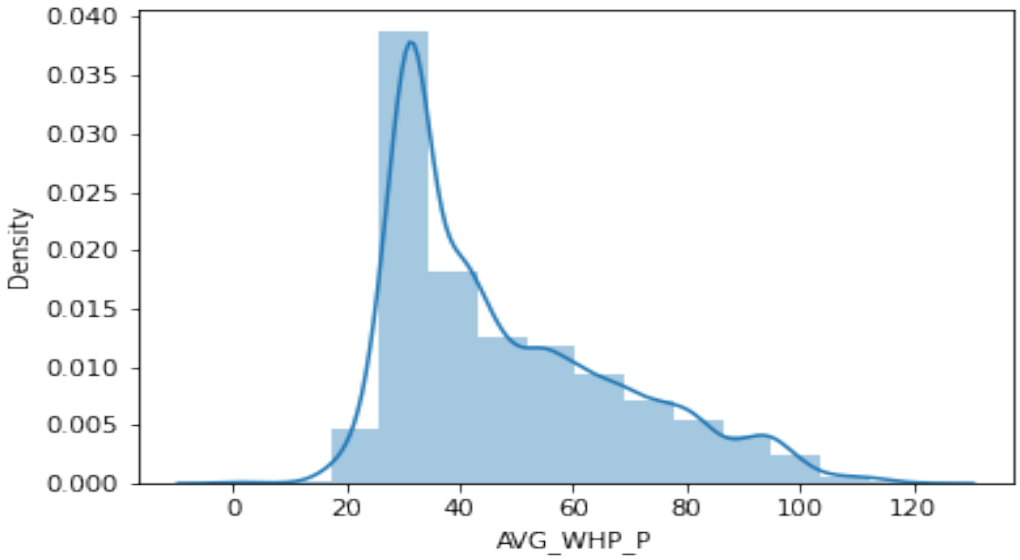
(c)



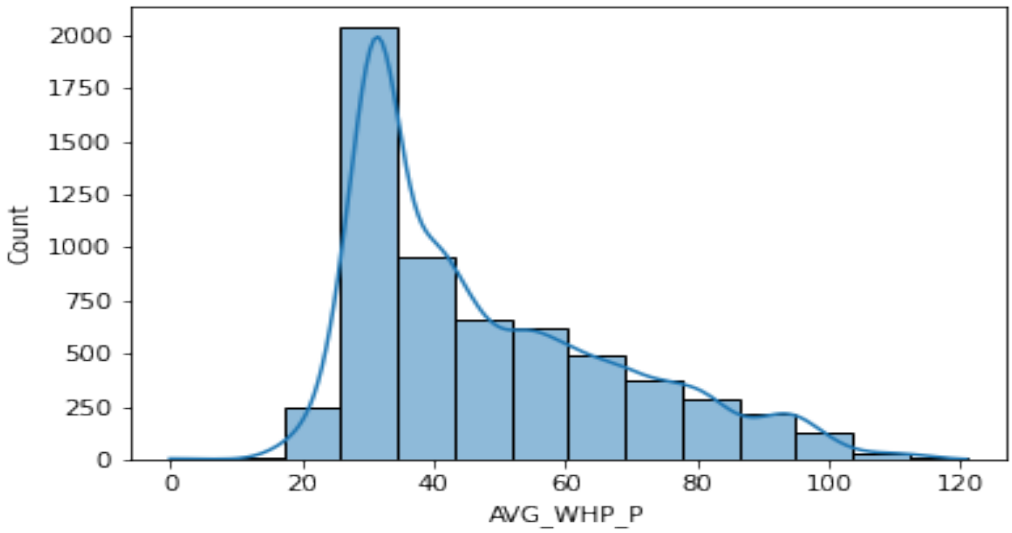
(d)

Figure A.9: (a) Boxplot for BORE_GAS_VOL (b) Boxplot without outliers for BORE_GAS_VOL (C) Histogram for BORE_GAS_VOL (c) Kernel Density Estimation plot for BORE_ GAS _VOL

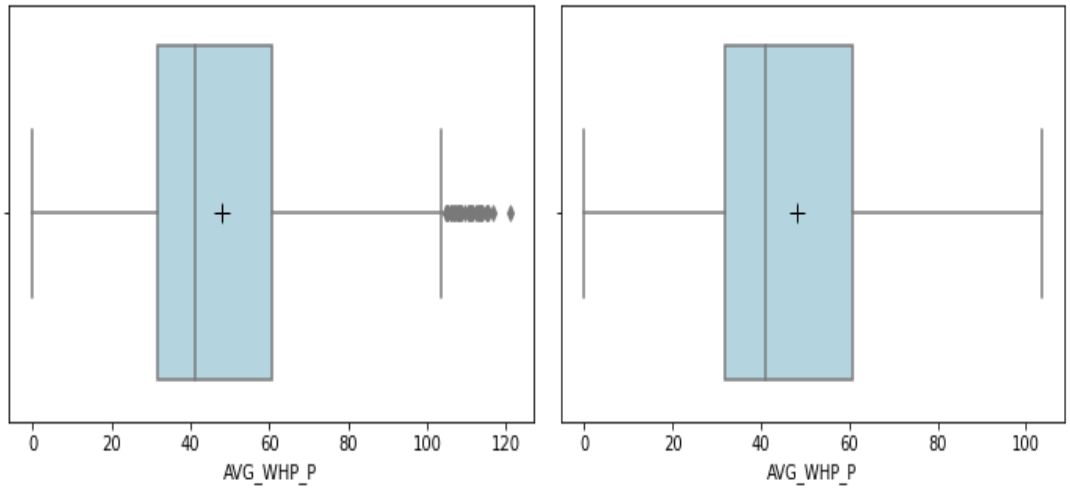
AVG_WHP_P



(a)



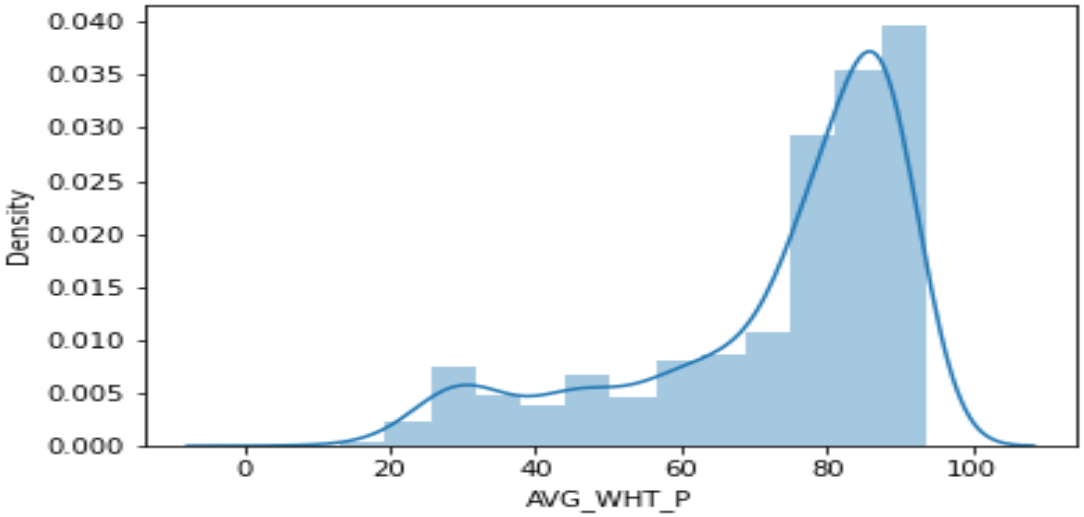
(b)



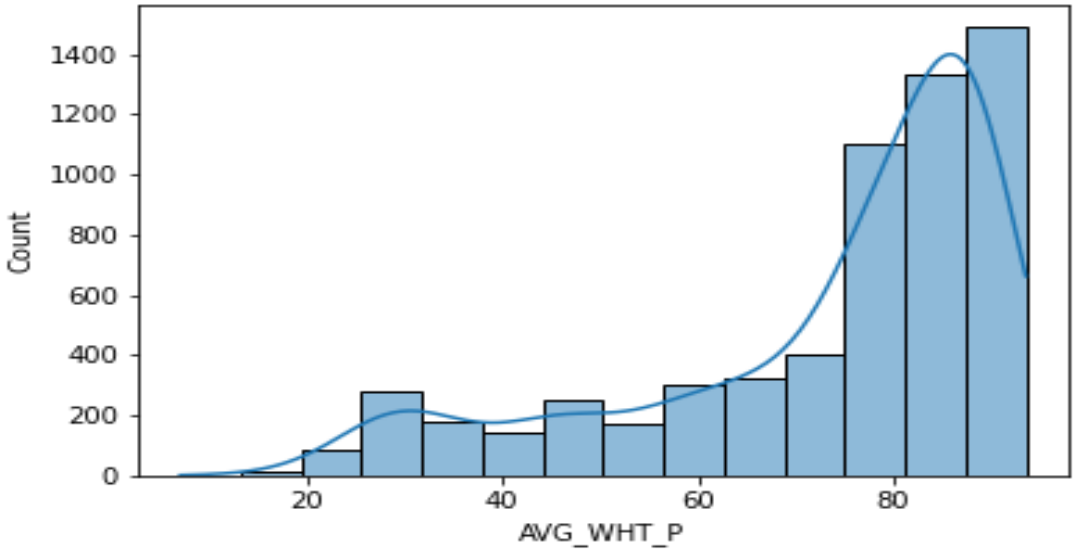
(c) **(d)**

Figure A.10: (a) Kernel Density Estimation plot for AVG_WHP_P (b) Histogram for AVG_WHP_P (c) Boxplot for AVG_WHP_P (d) Boxplot without outliers for AVG_WHP_P

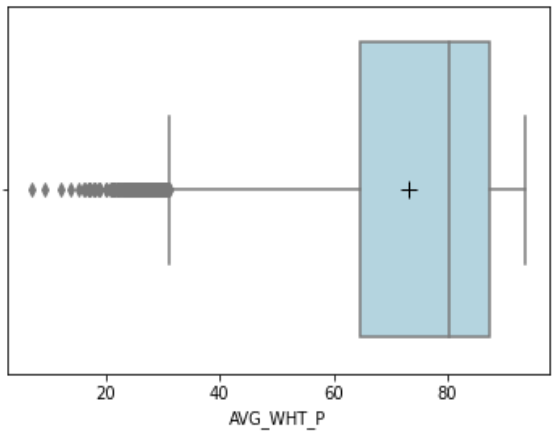
AVG_WHT_P



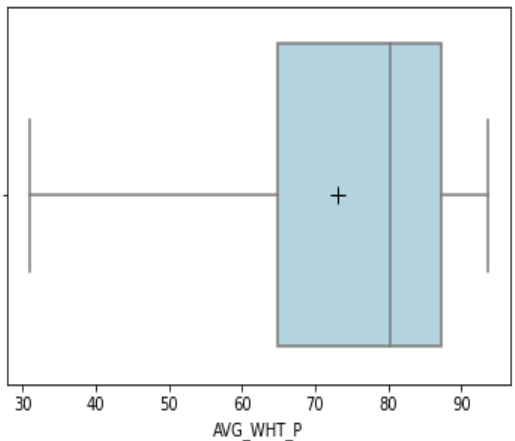
(a)



(b)



(c)

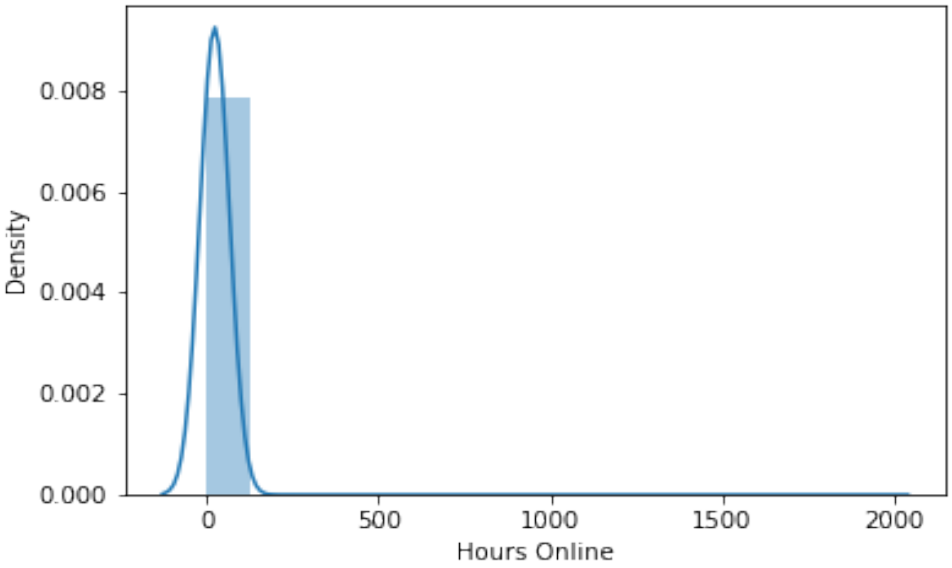


(d)

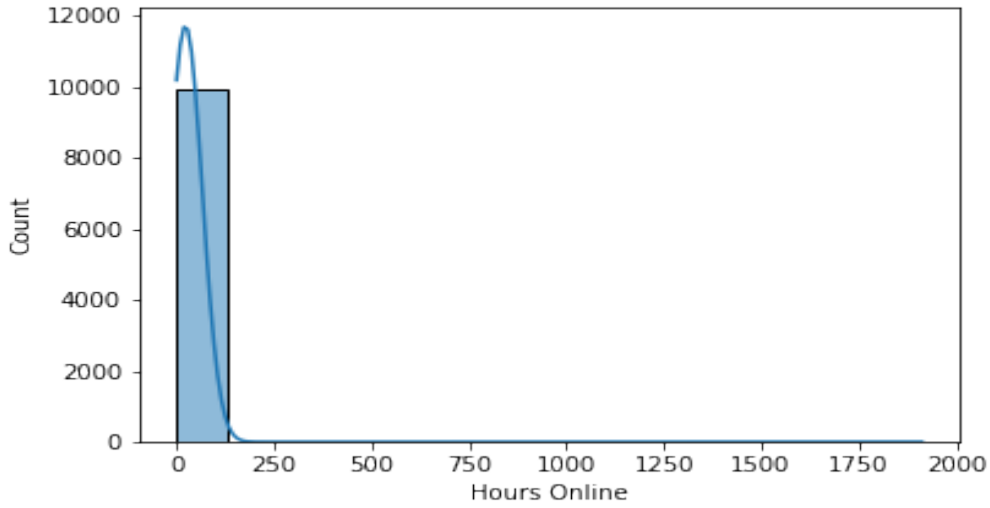
Figure A.11: (a) Kernel Density Estimation plot for AVG_WHT_P (b) Histogram for AVG_WHT_P (c) Boxplot for AVG_WHT_P (d) Boxplot without outliers for AVG_WHT_P

V. Feature statistics in Kyle Master dataset

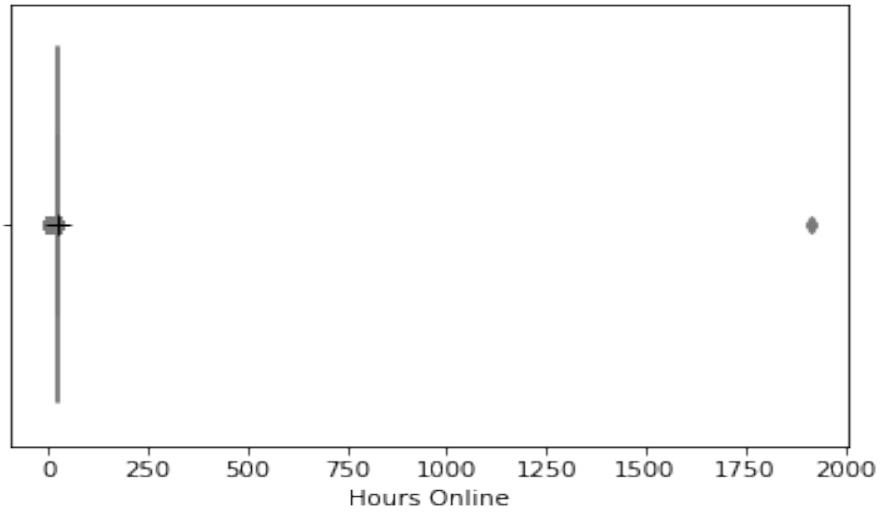
Hours Online



(a)



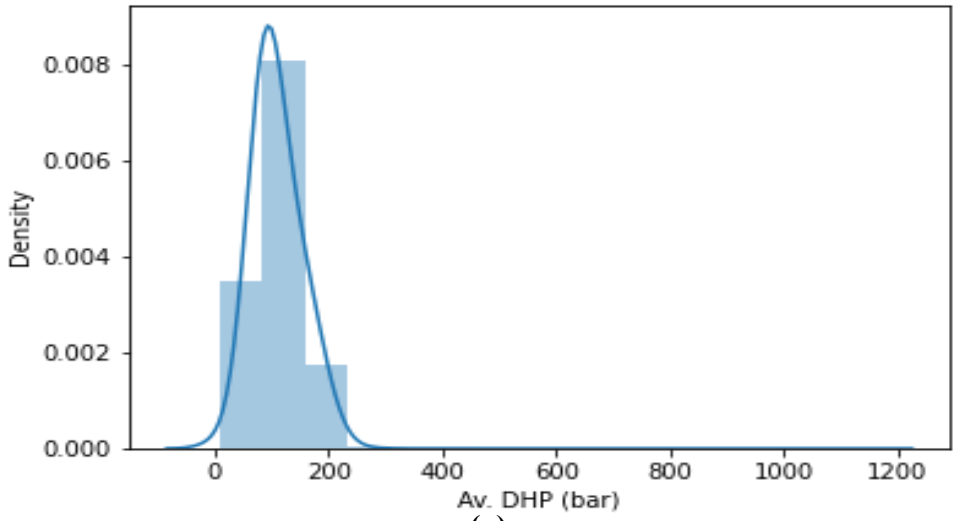
(b)



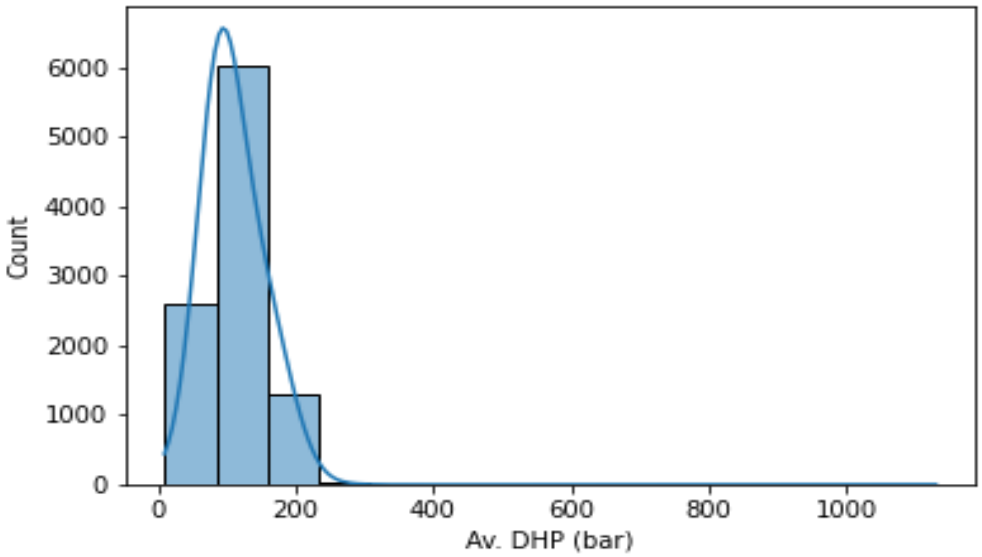
(c)

Figure A.12: (a) Kernel Density Estimation plot for Hours Online (b) Histogram for Hours Online (c) Boxplot for Hours Online

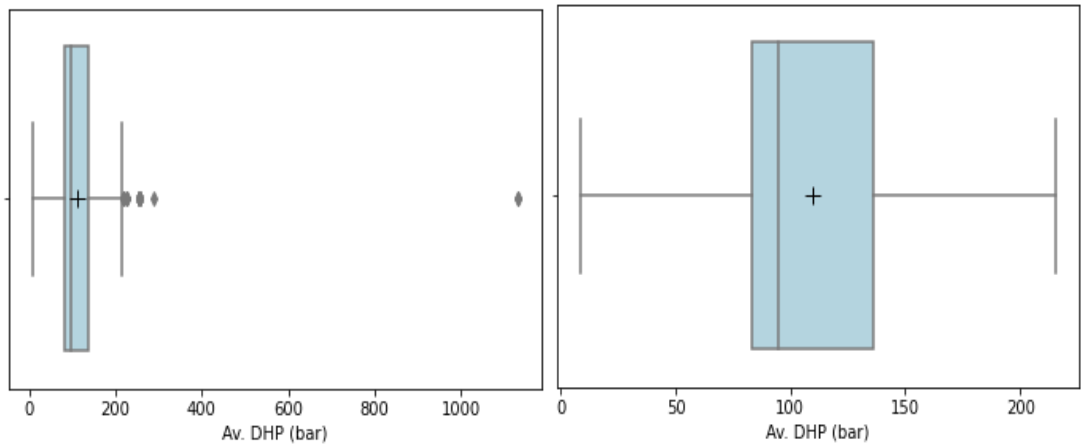
Av. DHP (bar)



(a)



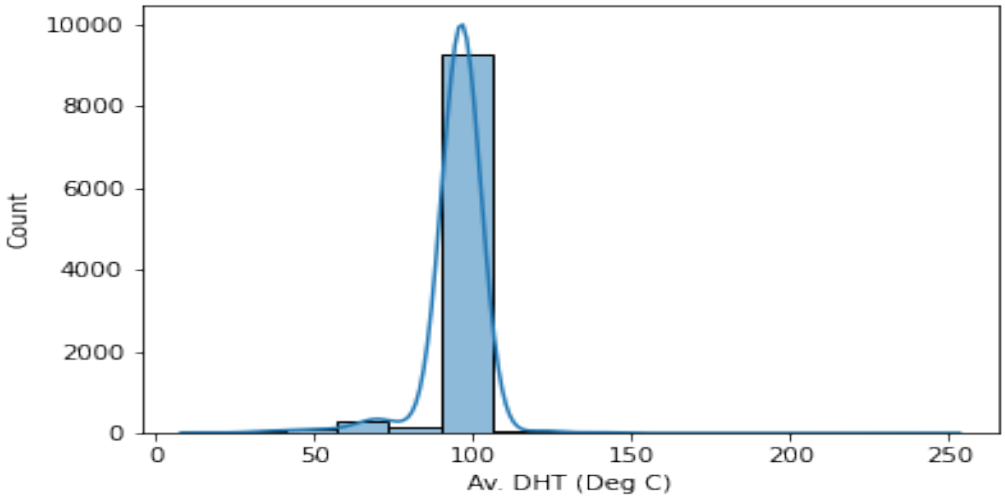
(b)



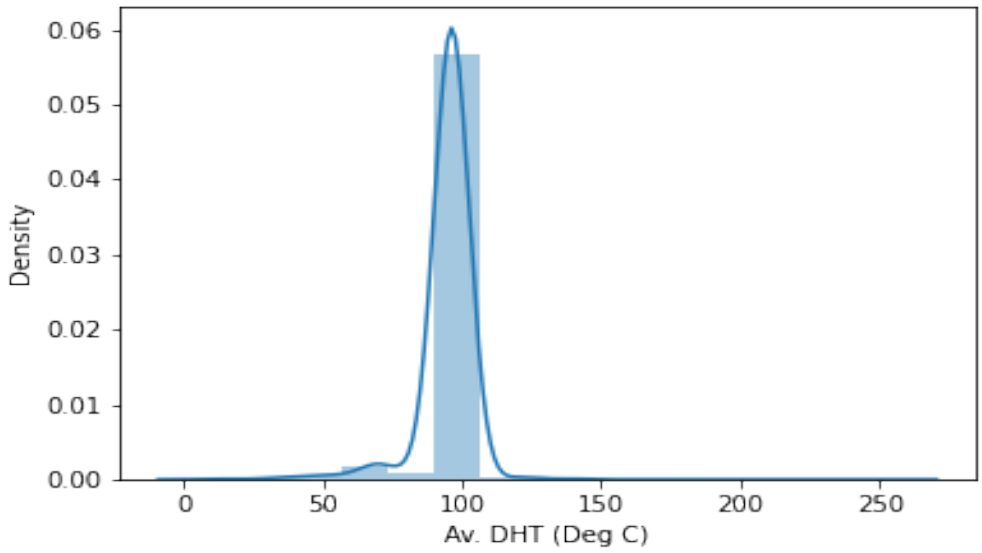
(c) **(d)**

Figure A.13: (a) Kernel Density Estimation plot for Av. DHP (bar) (b) Histogram for Av. DHP (bar) (c) Boxplot for Av. DHP (bar) (d) Boxplot without outliers for Av. DHP (bar)

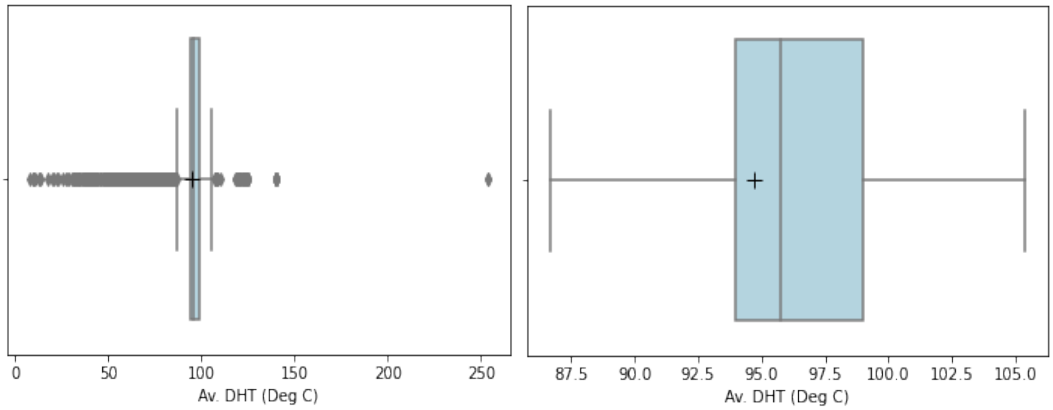
Av. DHT (Deg C)



(a)



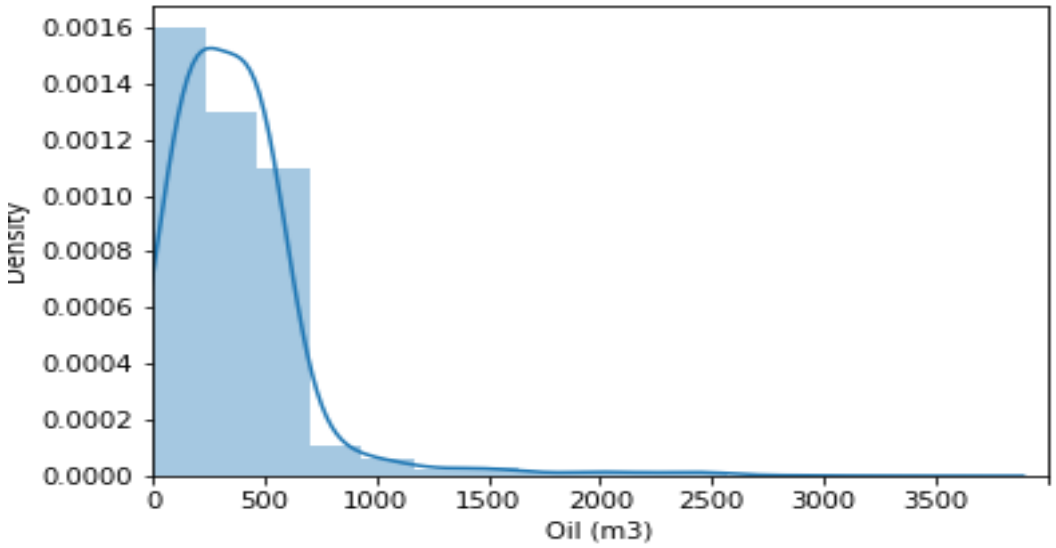
(b)



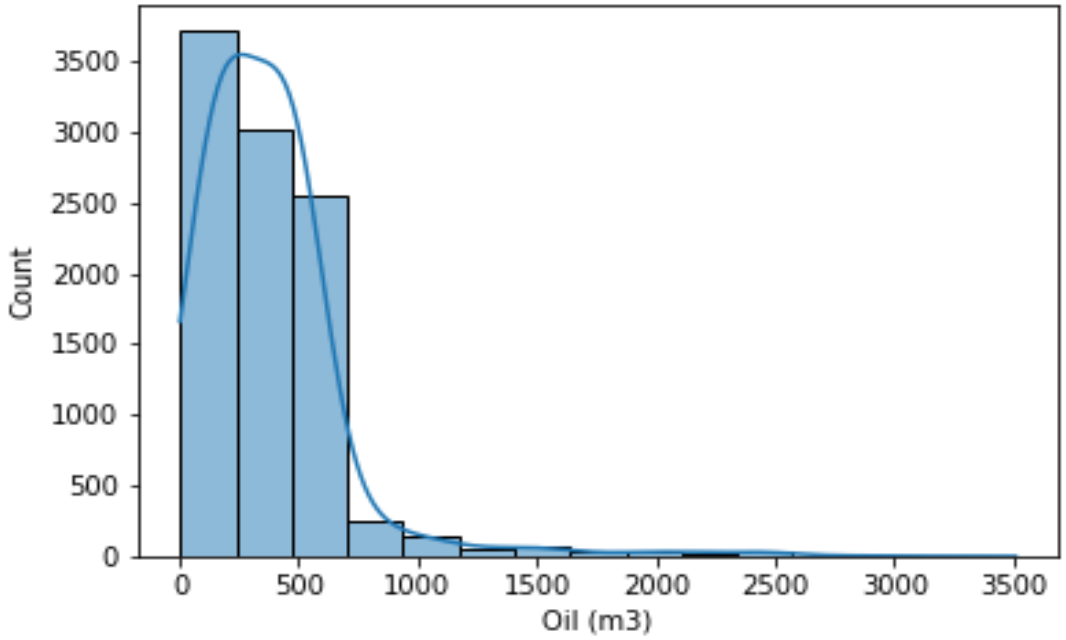
(c) **(d)**

Figure A.14: (a) Histogram for Av. DHT (Deg C) (b) Kernel Density Estimation plot for Av. DHT (Deg C) (c) Boxplot for Av. DHT (Deg C) (d) Boxplot without outliers for Av. DHT (Deg C)

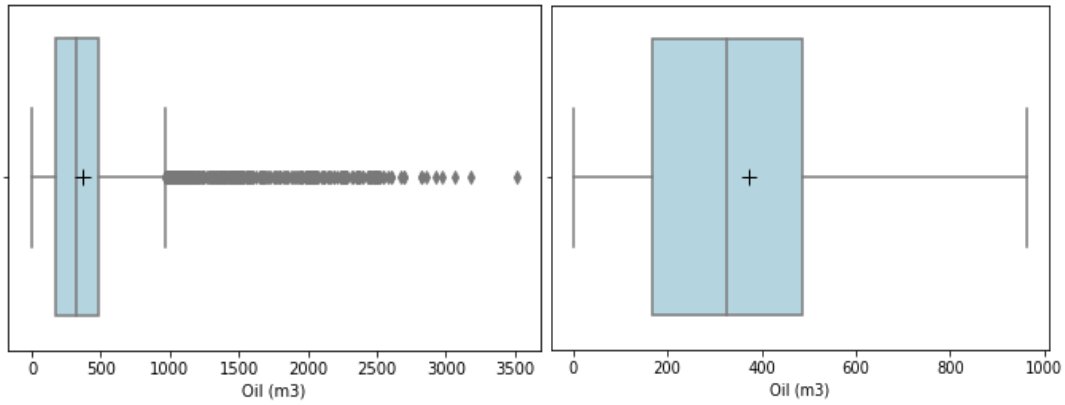
Oil (m3)


0.0016
0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
Density
0
500
1000
1500
2000
2500
3000
3500
Oil (m3)

(a)


3500
3000
2500
2000
1500
1000
500
0
Count
0
500
1000
1500
2000
2500
3000
3500
Oil (m3)

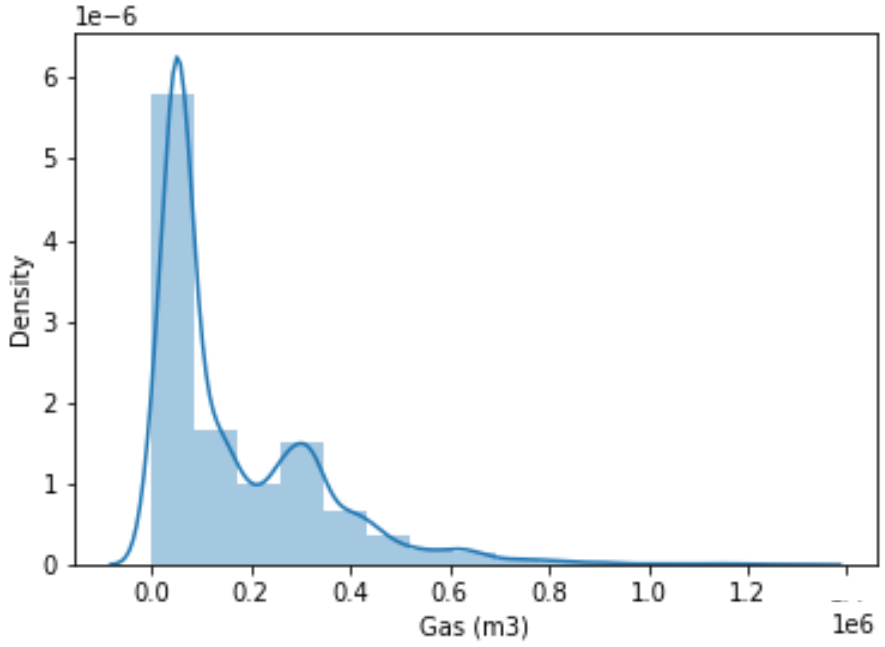
(b)



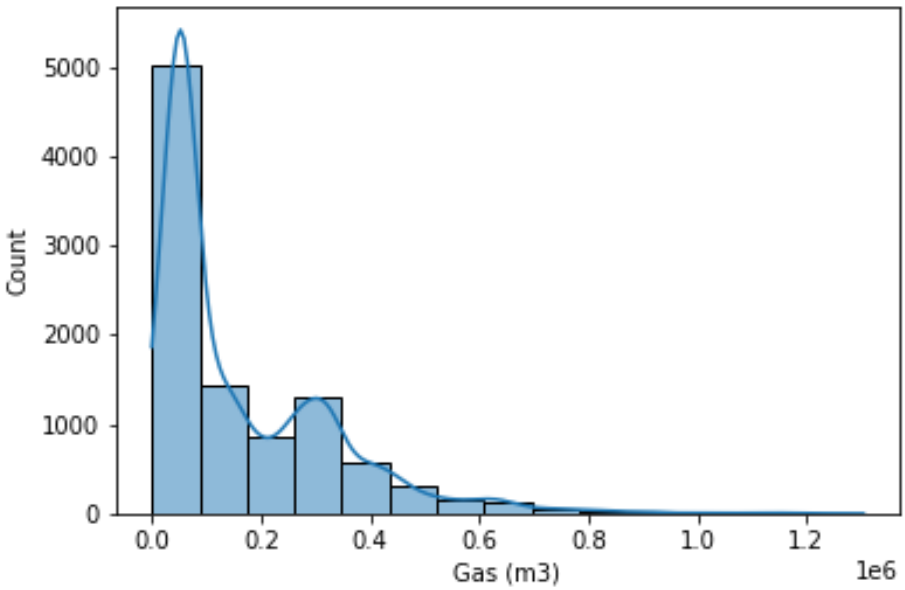
(c) **(d)**

Figure A.15: (a) Kernel Density Estimation plot for Oil (m3) (b) Histogram for Oil (m3) (c) Boxplot for Oil (m3) (d) Boxplot without outliers for Oil (m3)

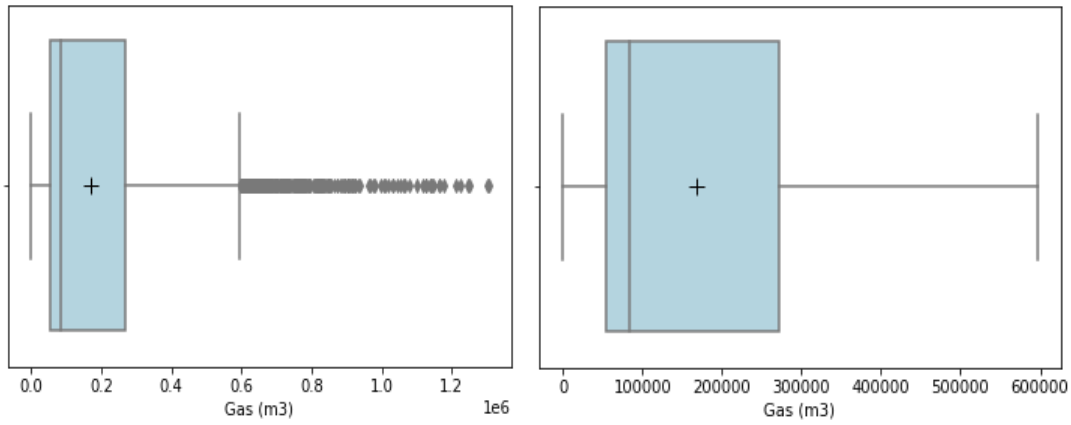
Gas (m3)



(a)



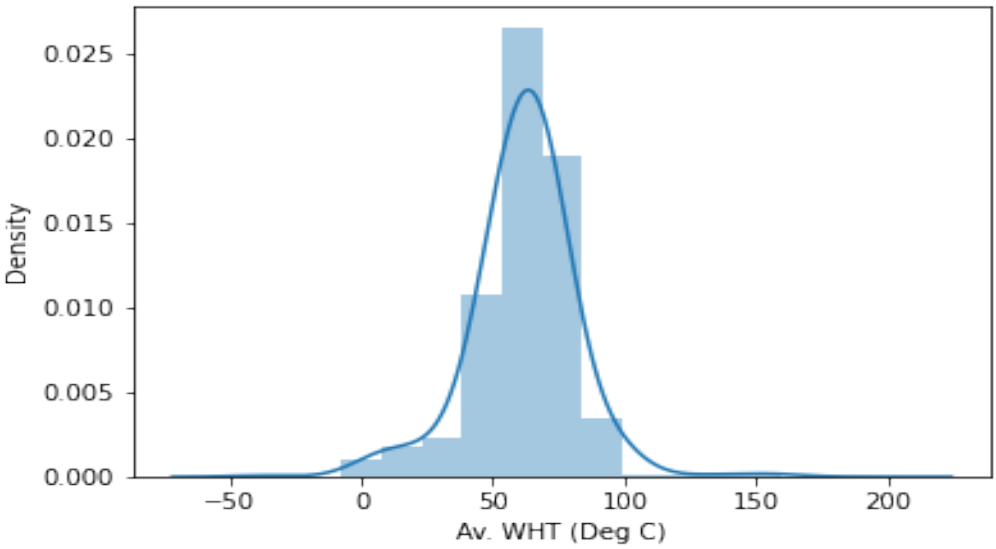
(b)



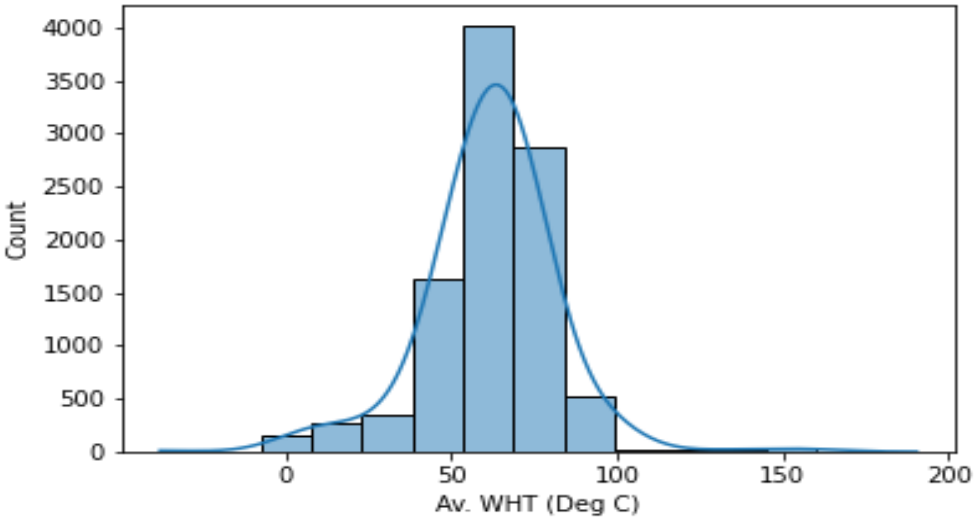
(c) **(d)**

Figure A.16: (a) Kernel Density Estimation plot for Gas (m3) (b) Histogram for Gas (m3) (c) Boxplot for Gas (m3) (d) Boxplot without outliers for Gas (m3)

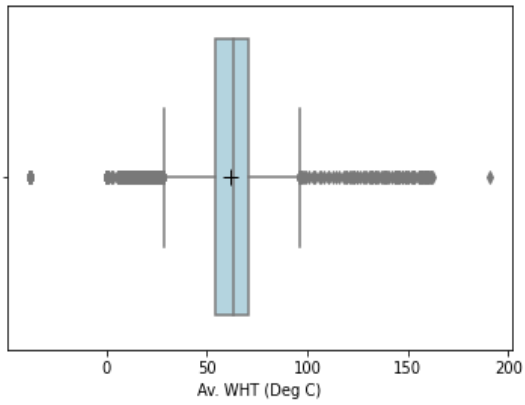
Av. WHT (Deg C)



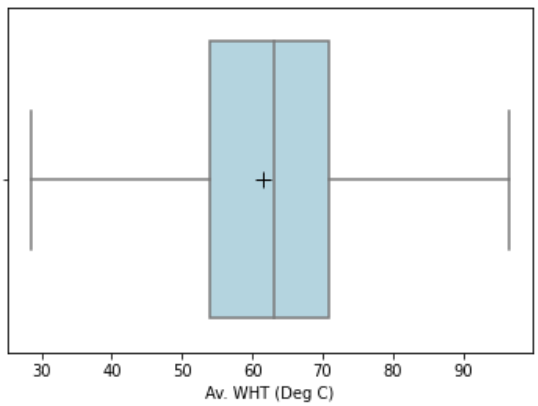
(a)



(b)



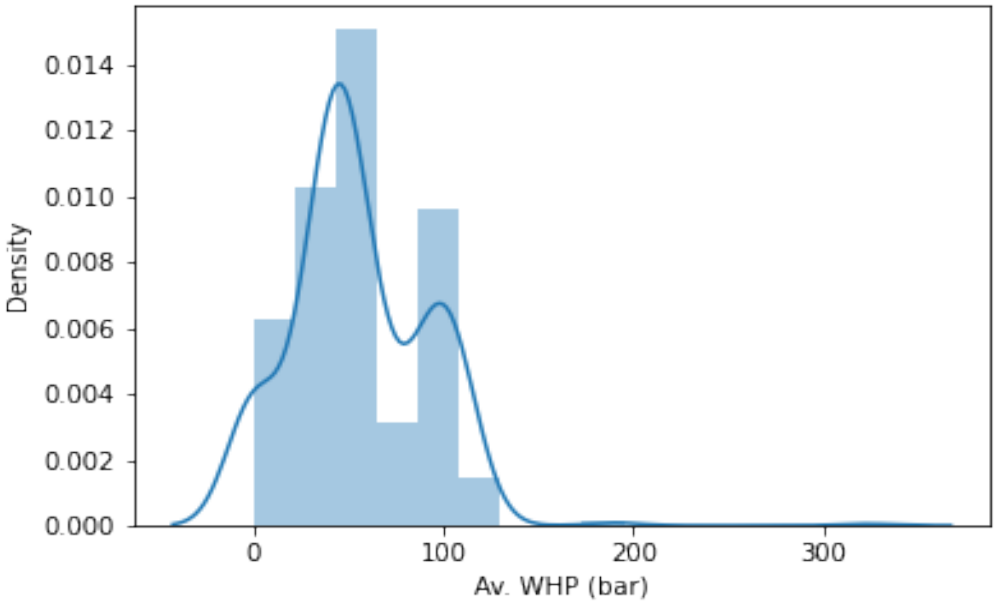
(c)



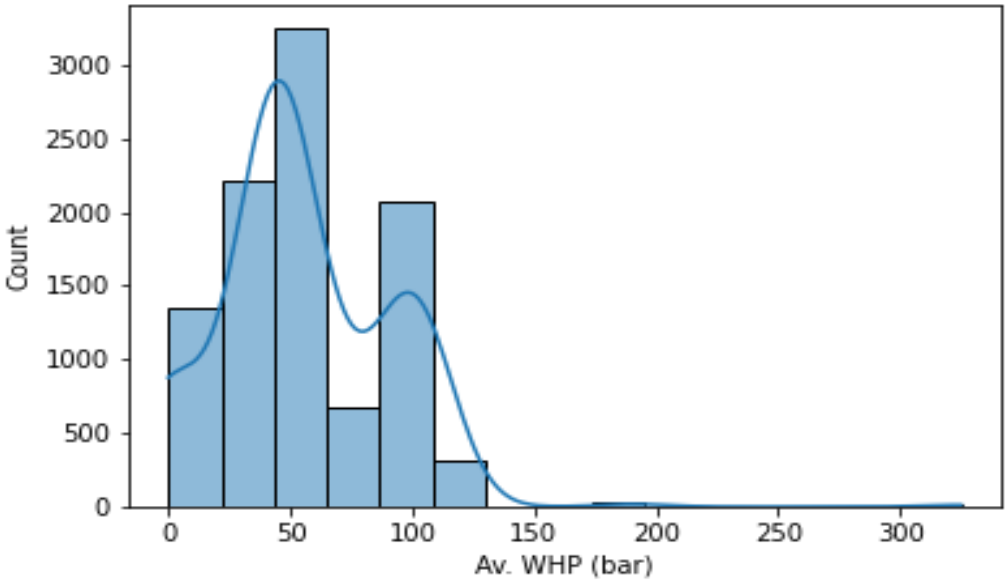
(d)

Figure A.17: (a) Kernel Density Estimation plot for Av. WHT (Deg C) (b) Histogram for Av. WHT (Deg C) (c) Boxplot for Av. WHT (Deg C) (d) Boxplot without outliers for Av. WHT (Deg C)

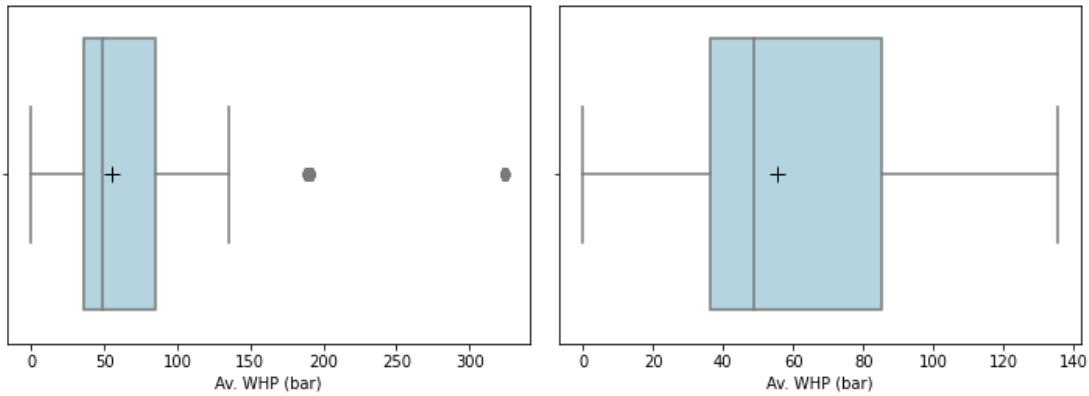
Av. WHP (bar)



(a)



(b)



(c) **(d)**

Figure A.18: (a) Kernel Density Estimation plot for Av. WHP (bar) (b) Histogram for Av. WHP (bar) (c) Boxplot for Av. WHP (bar) (d) Boxplot without outliers for Av. WHP (bar)

VI. Feature statistics in Volve and Kyle Master dataset after forward filling

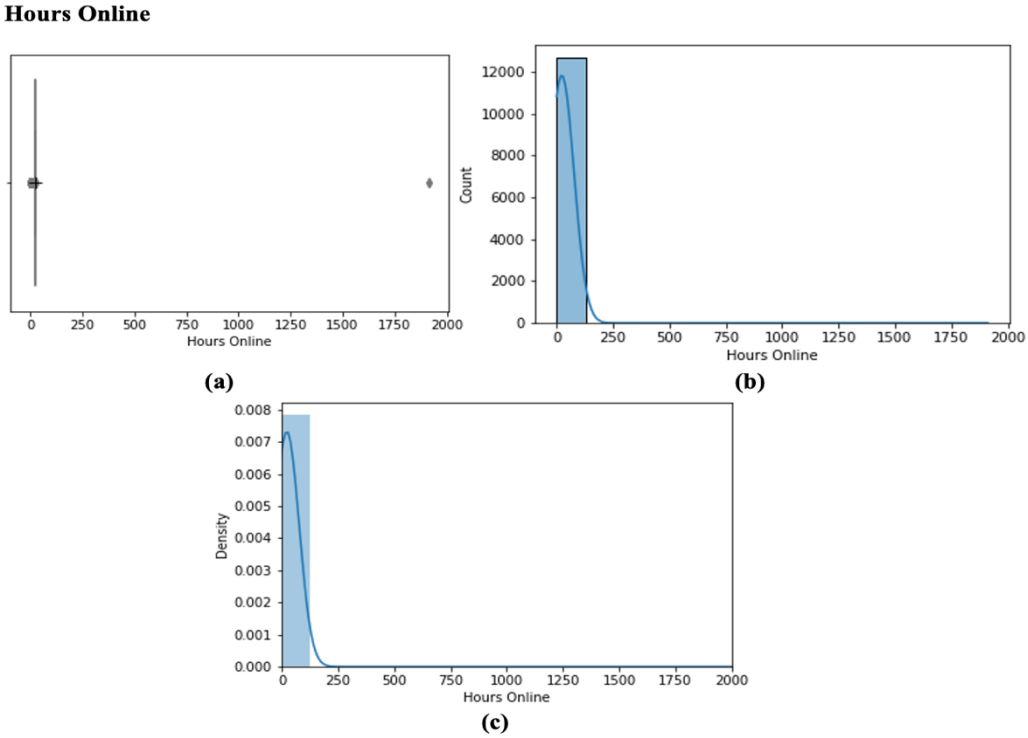


Figure A.19: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Hours Online after forward filling

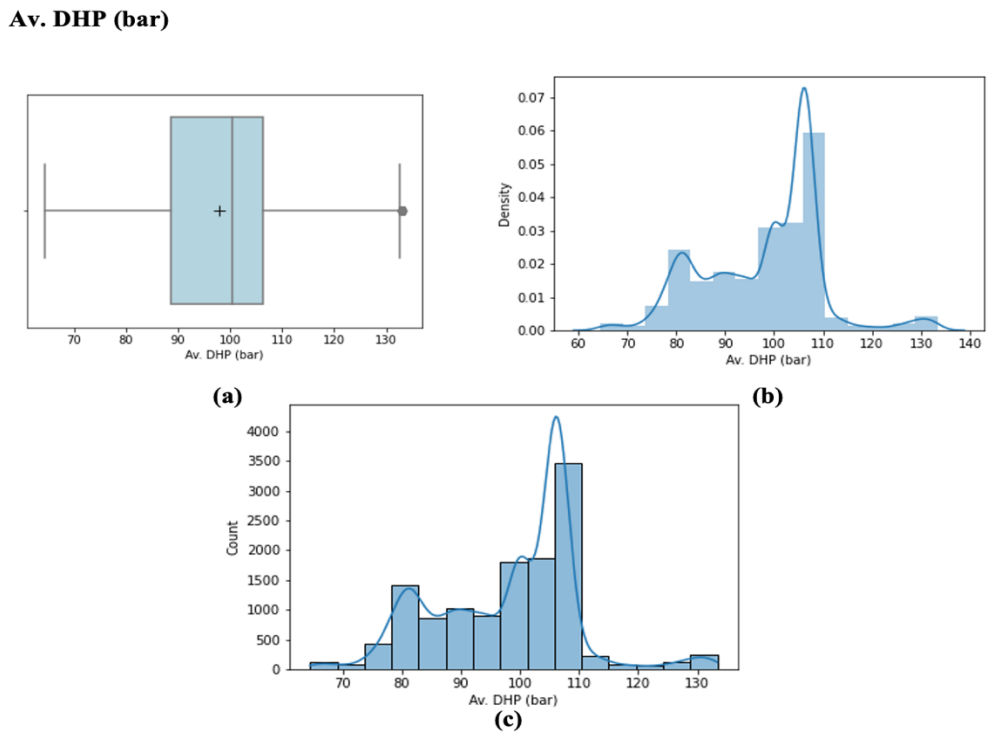


Figure A.20: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. DHP (bar) after forward filling

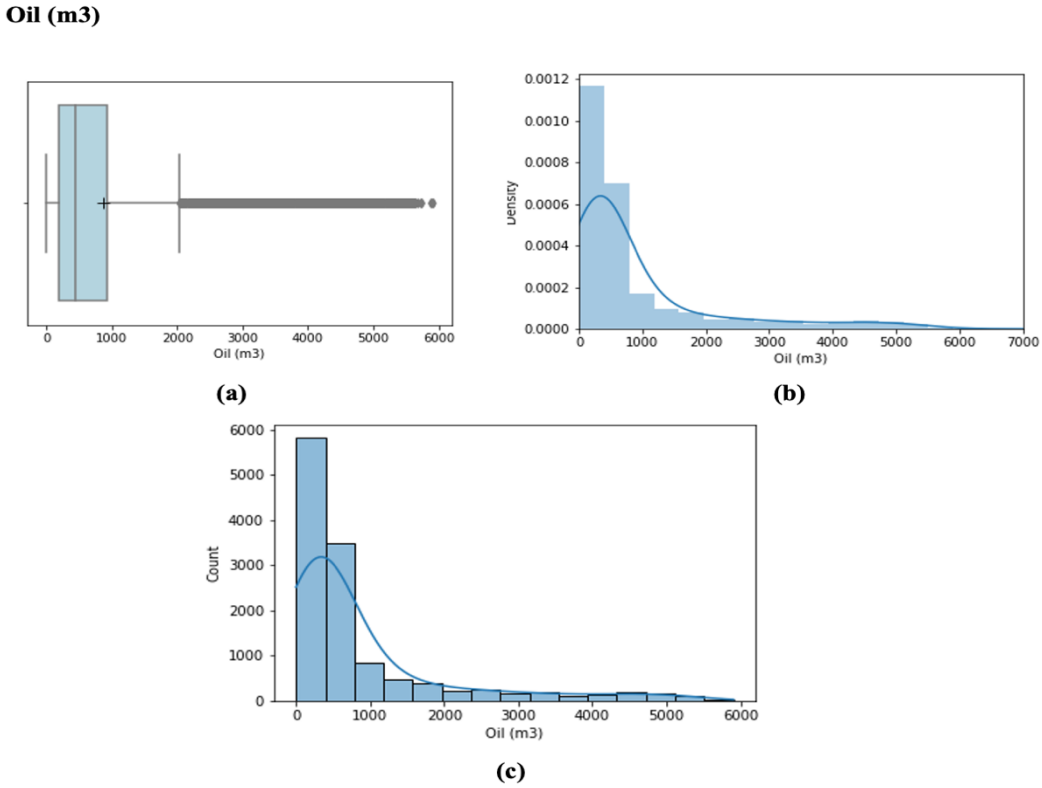


Figure A.21: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Oil (m3) after forward filling

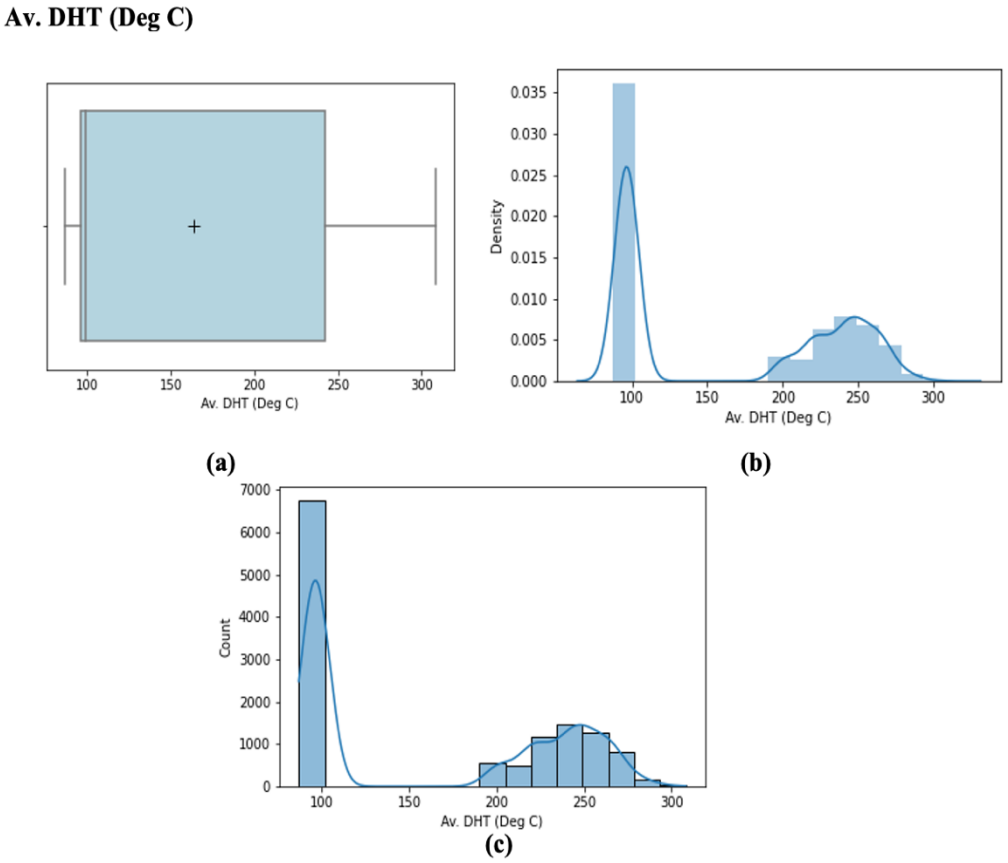


Figure A.24: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. DHT (Deg C) after forward filling

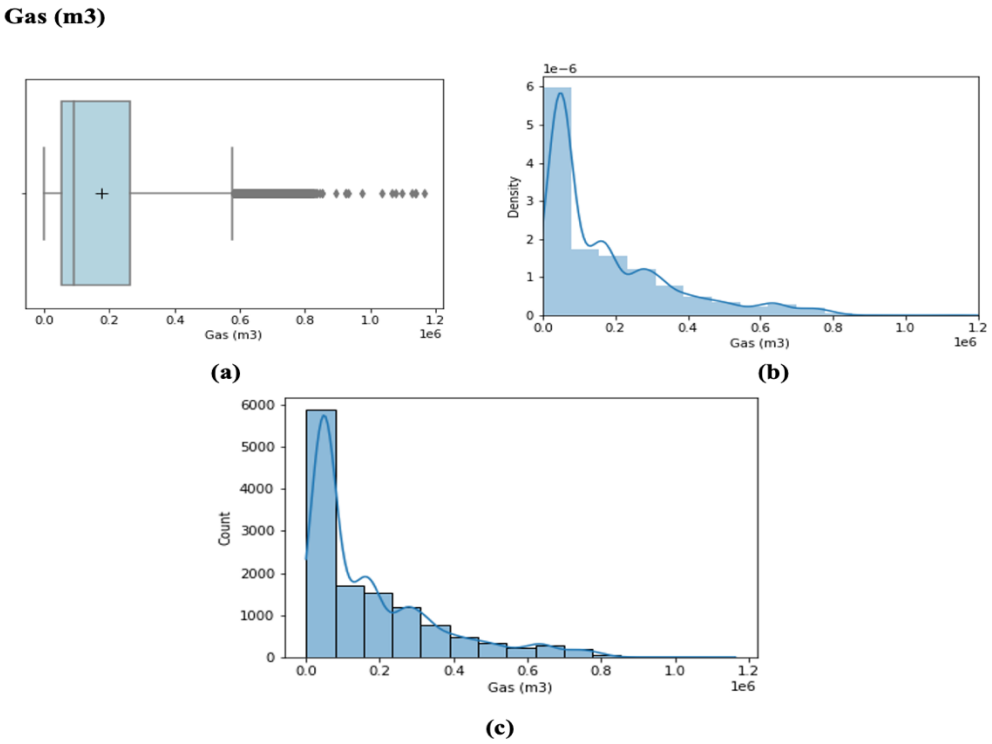


Figure A.22: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Gas (m3) after forward filling

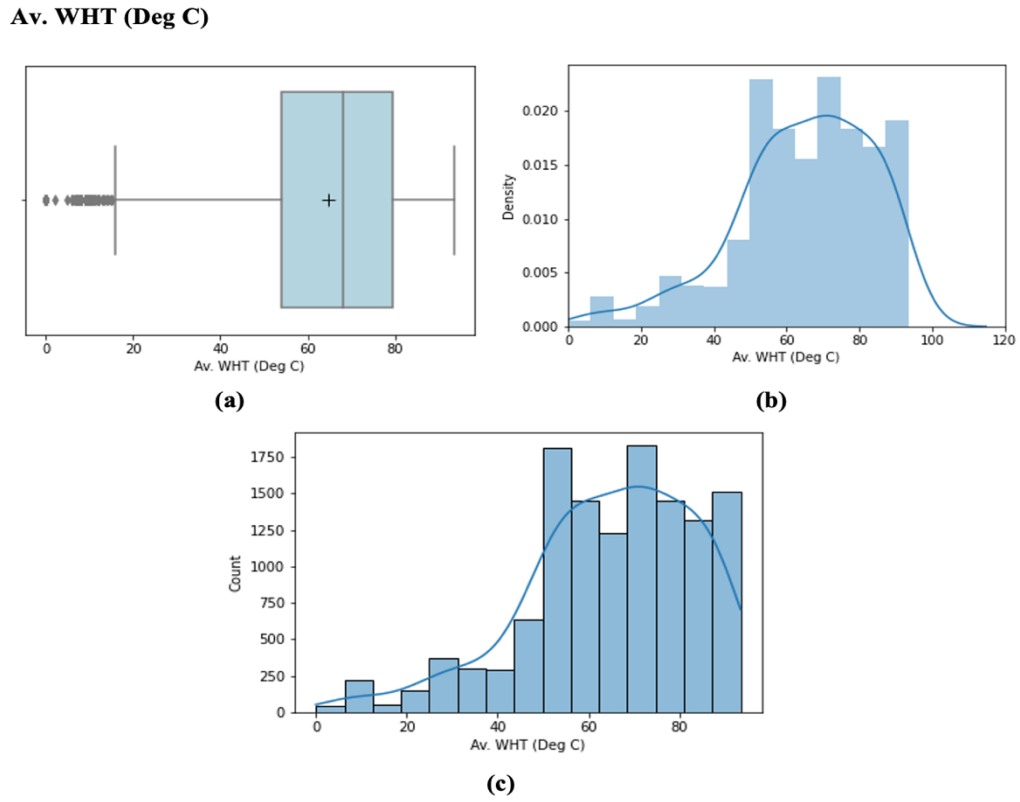


Figure A.23: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. WHT (Deg C) after forward filling

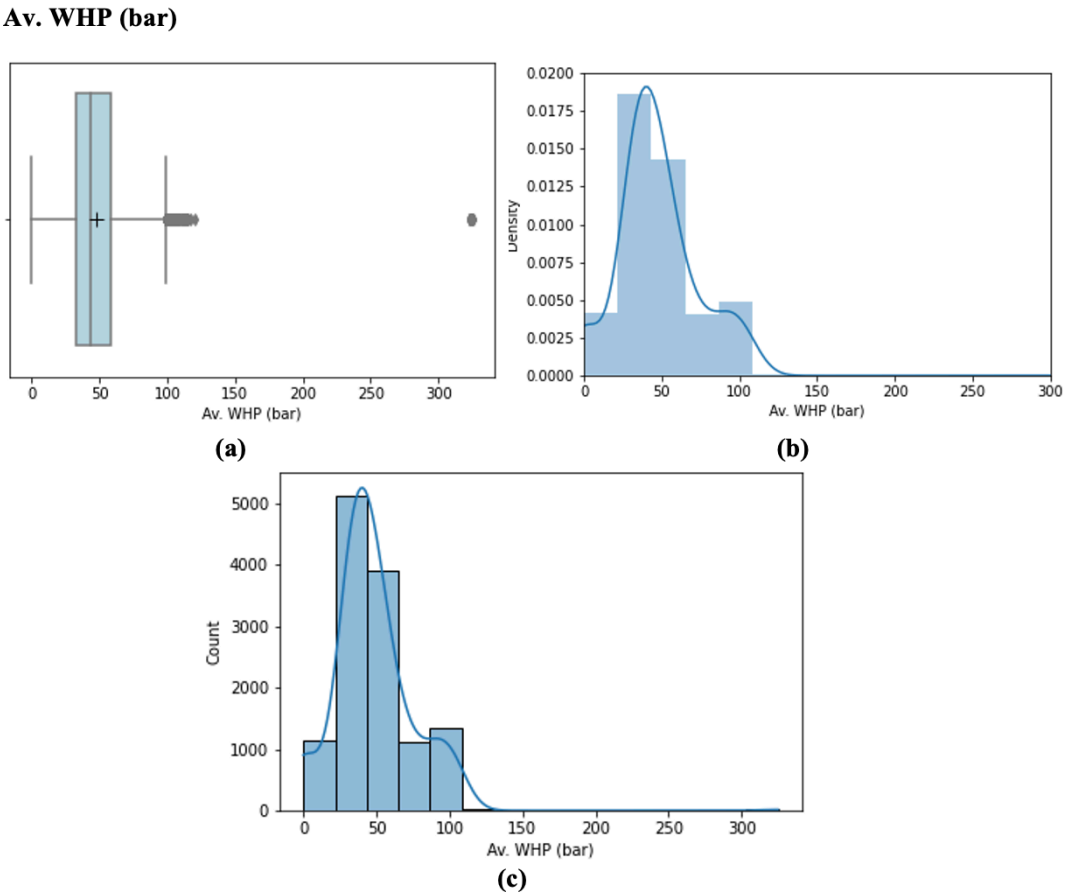


Figure A.25: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. WHP (bar) after forward filling

VII. Feature statistics in Volve and Kyle Master dataset after median imputation

Hours Online

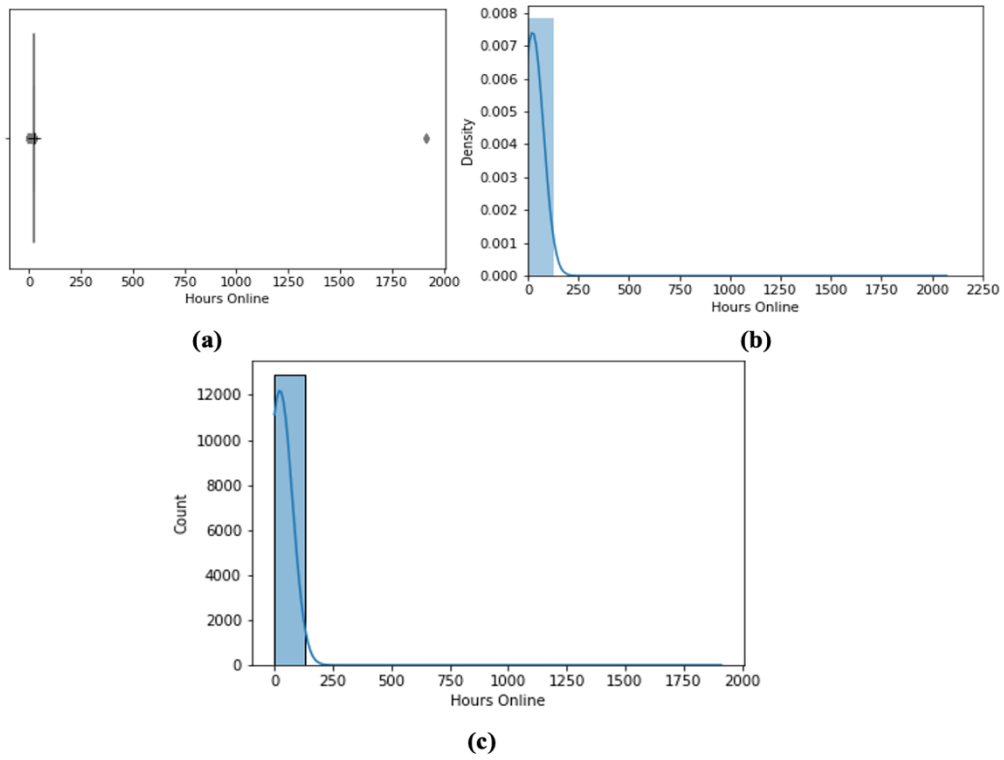


Figure A.26: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Hours Online after median imputation

Av. DHP (bar)

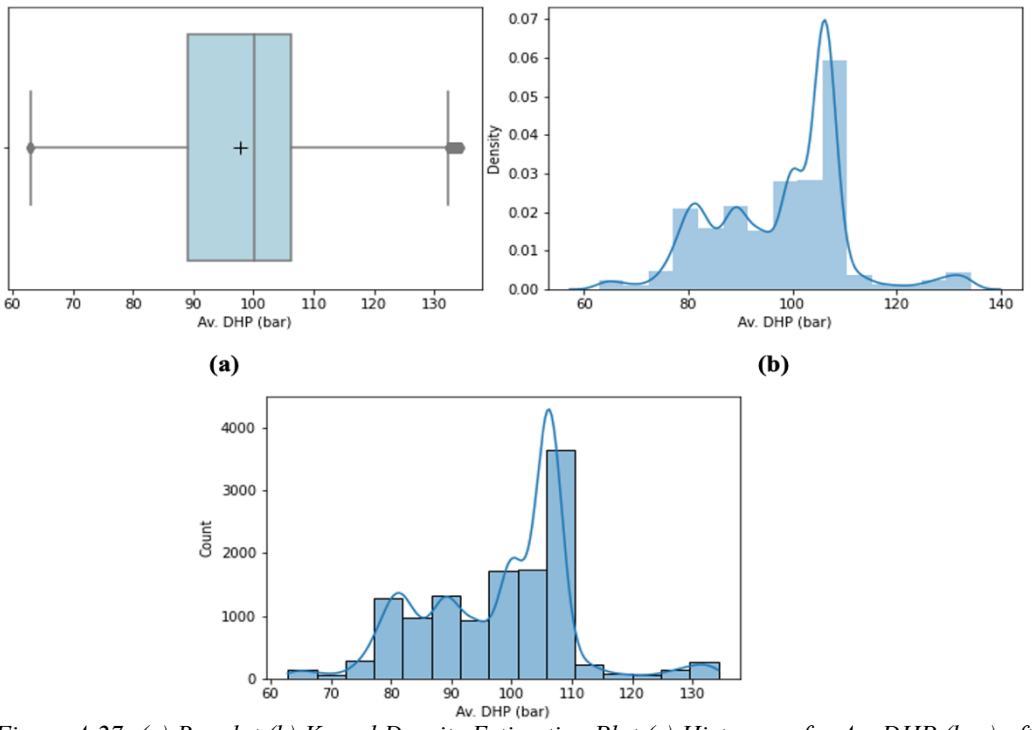


Figure A.27: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. DHP (bar) after median imputation

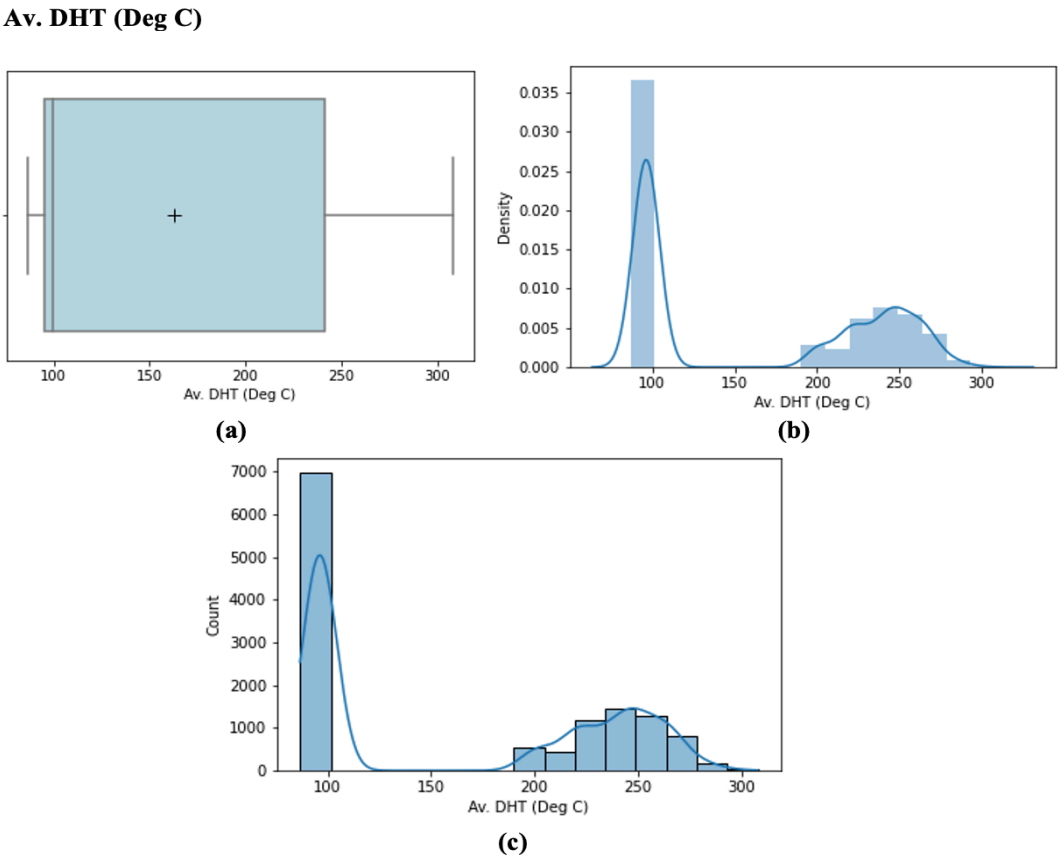


Figure A.28: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. DHT (Deg C) after median imputation

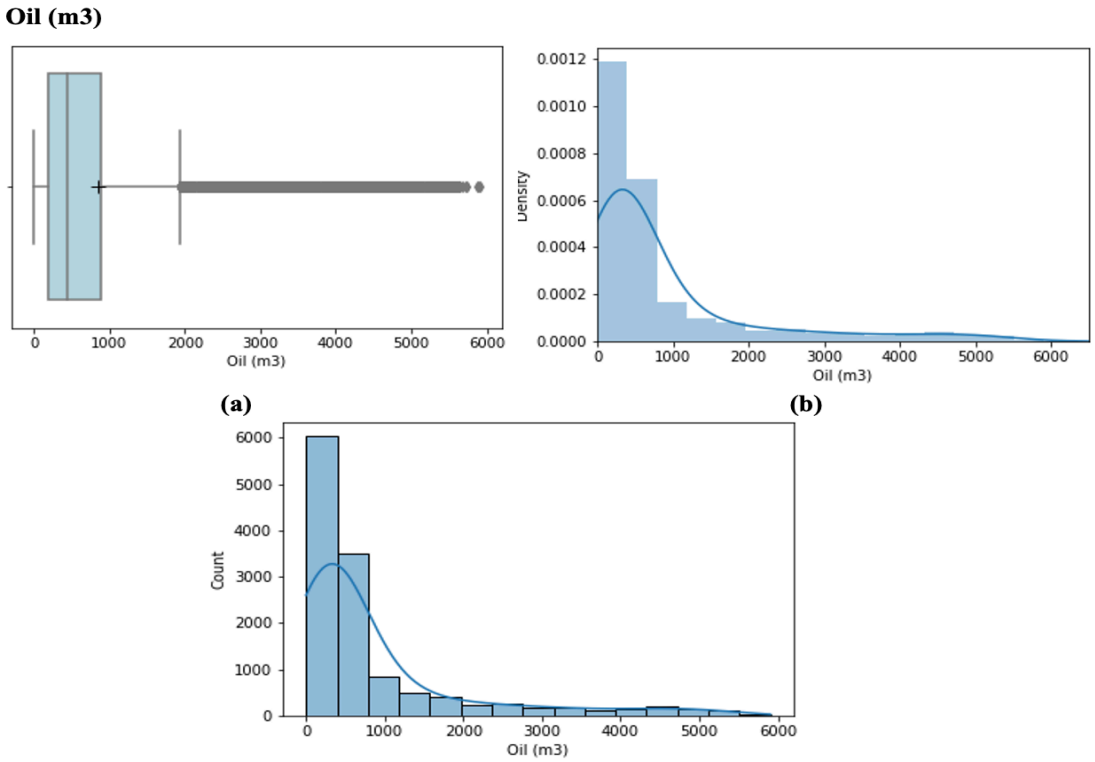


Figure A.29: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Oil (m3) after median imputation

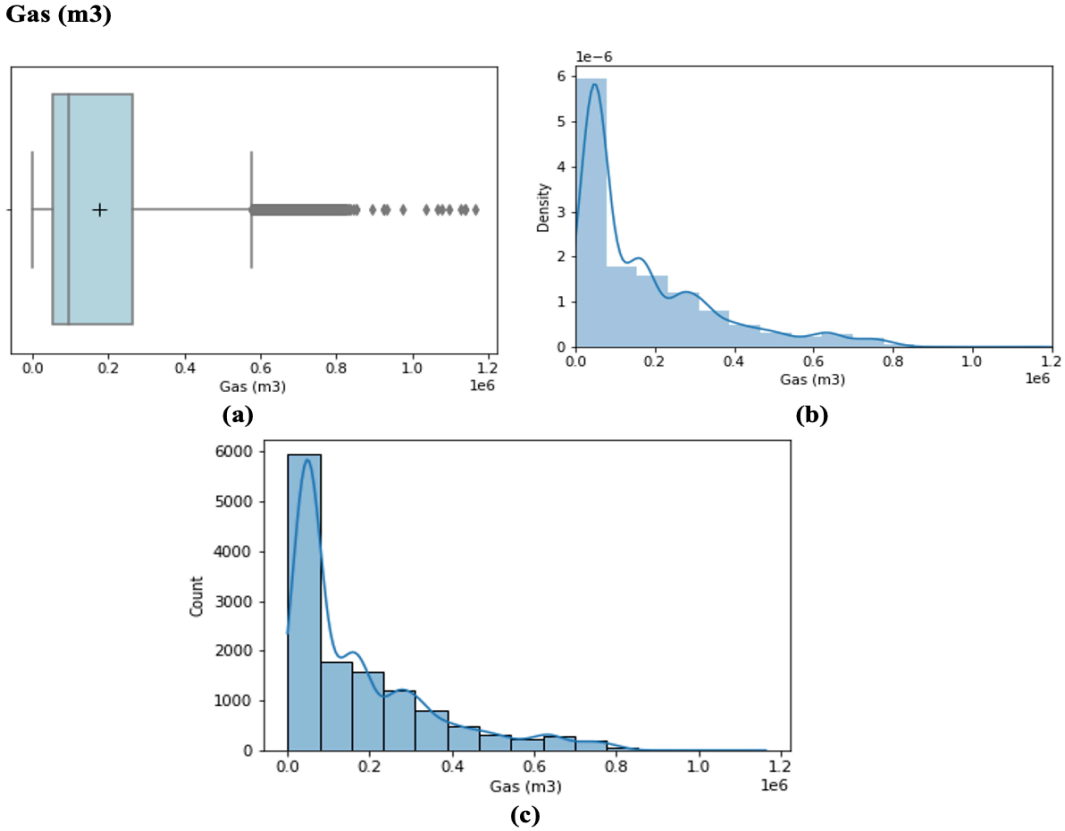


Figure A.30: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Gas (m3) after median imputation

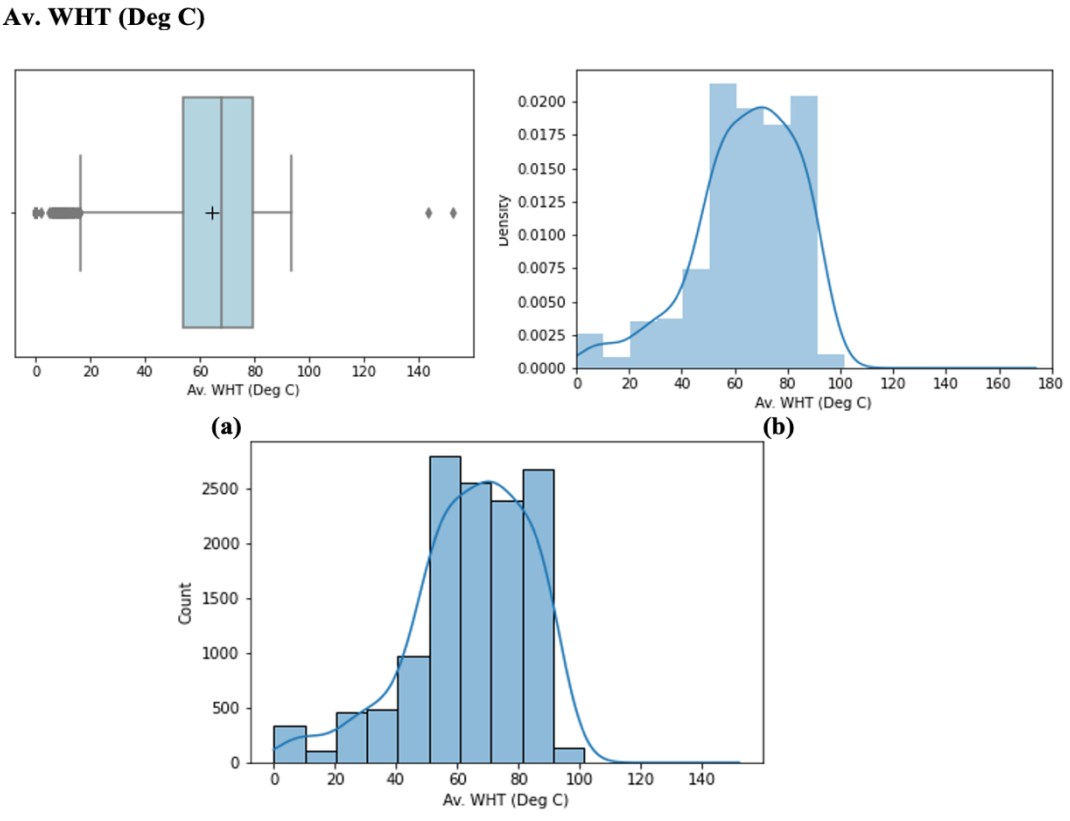


Figure A.31: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. WHT (Deg C) after median imputation

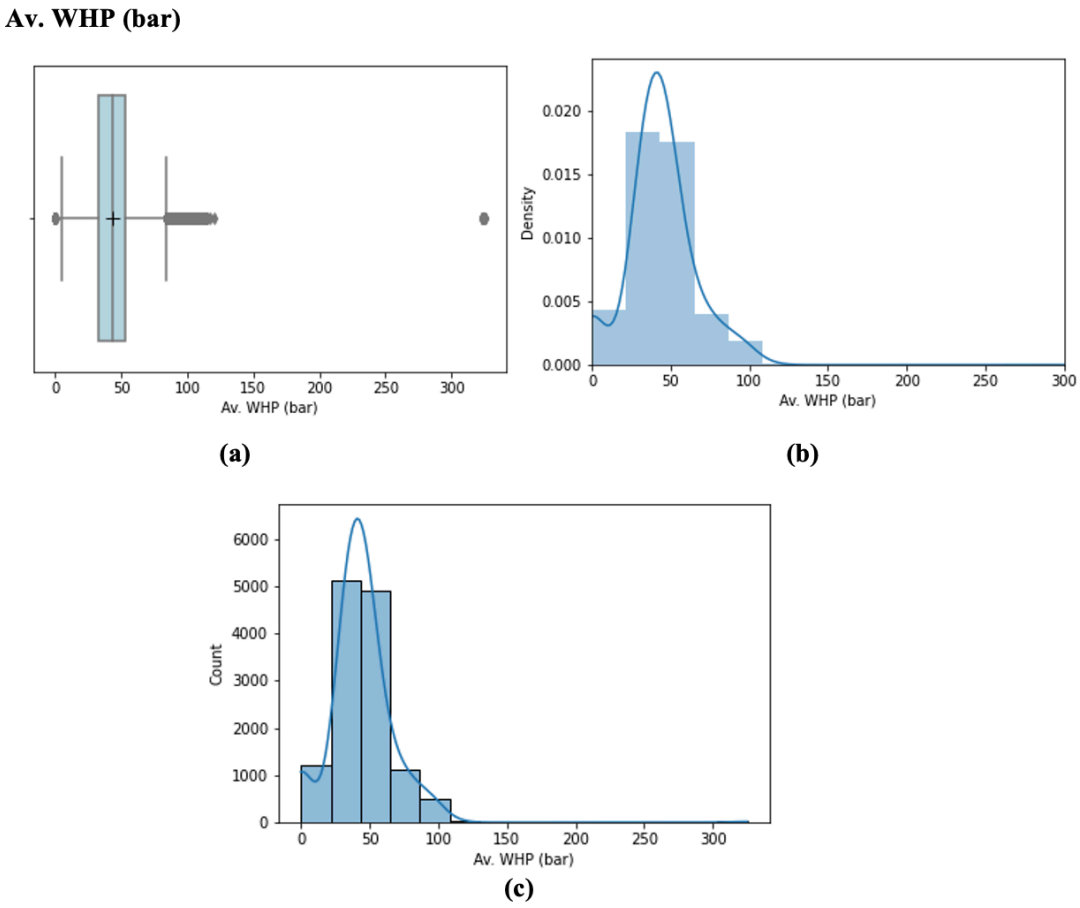


Figure A.32: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. WHP (bar) after median imputation

VIII. Feature statistics in Volve and Kyle Master dataset after self-supervised imputation

Hours Online

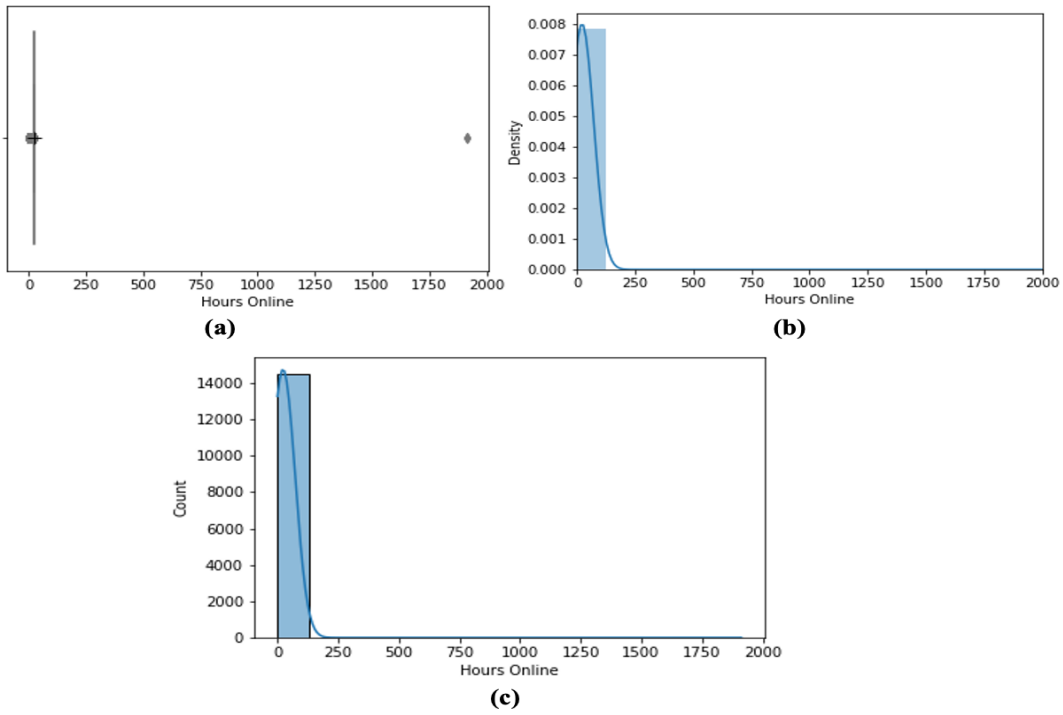


Figure A.33: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Hours Online after self-supervised imputation

Av. DHP (bar)

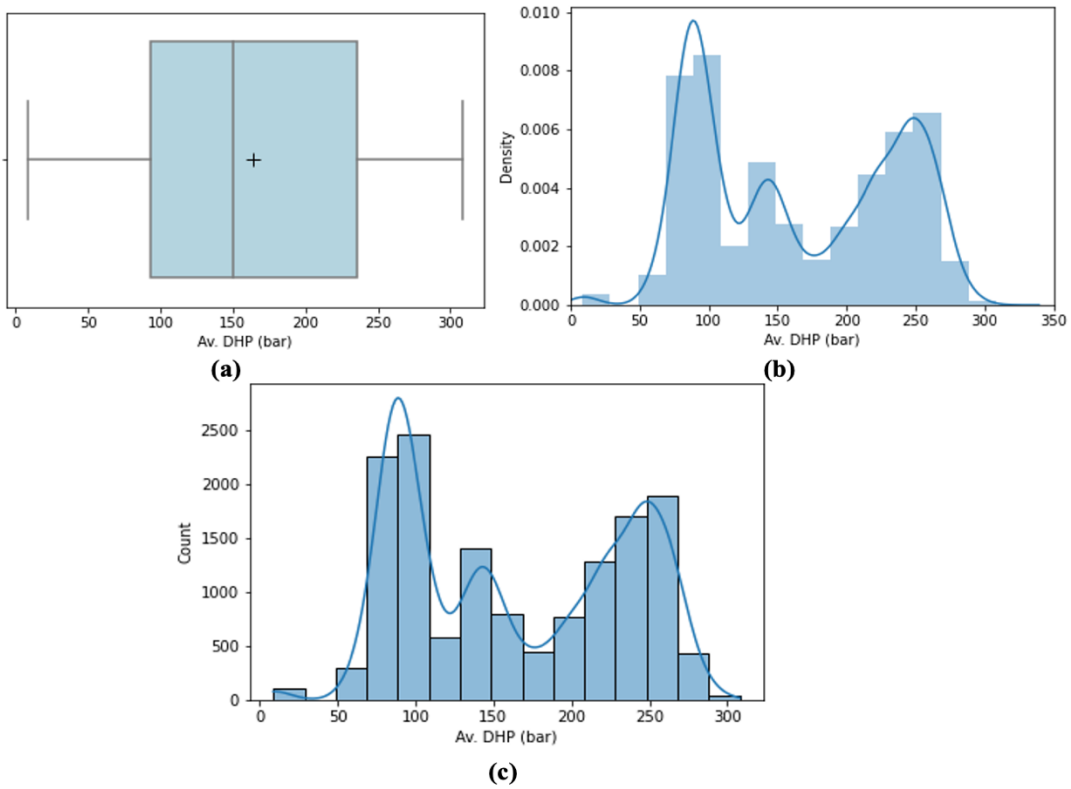


Figure A.34: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. DHP (bar) after self-supervised imputation

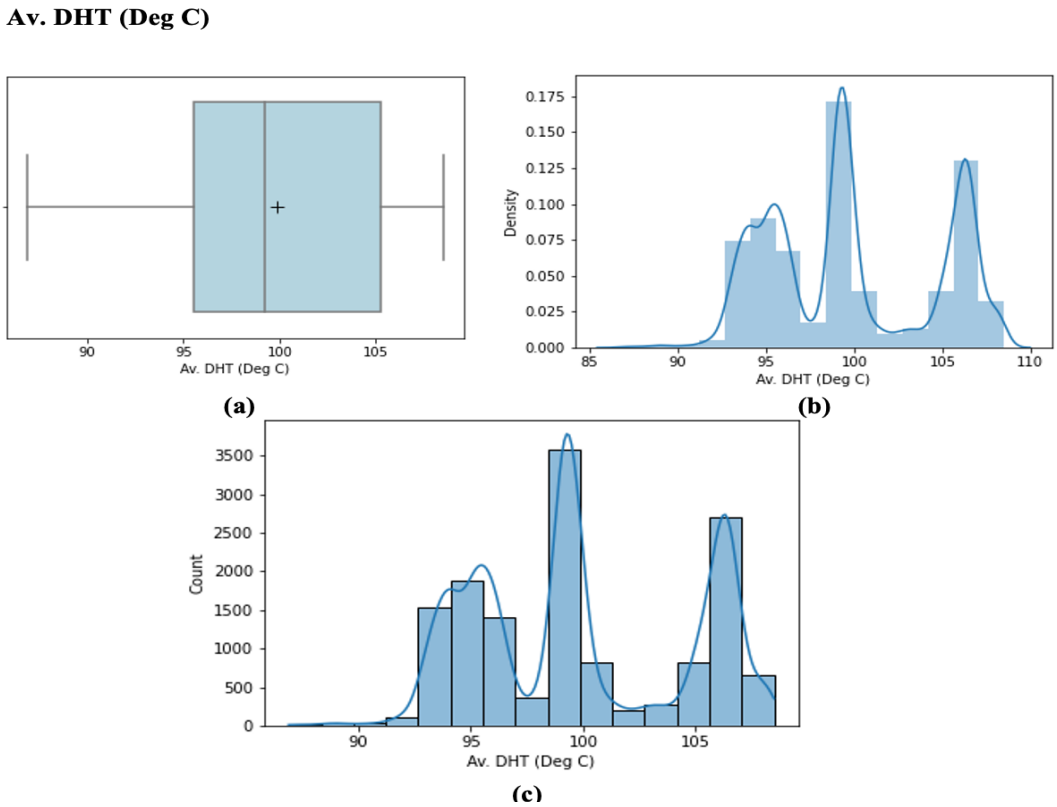


Figure A.35: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. DHT (Deg C) after self-supervised imputation

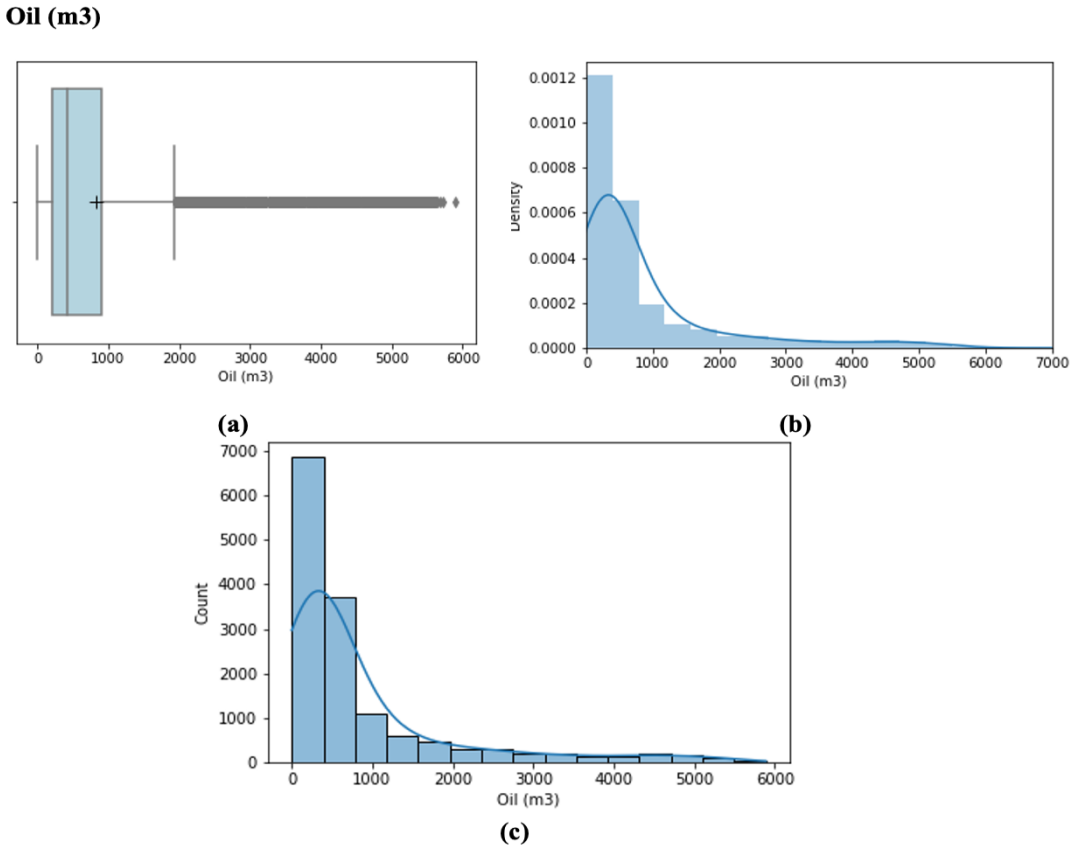


Figure A.36: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Oil (m3) after self-supervised imputation

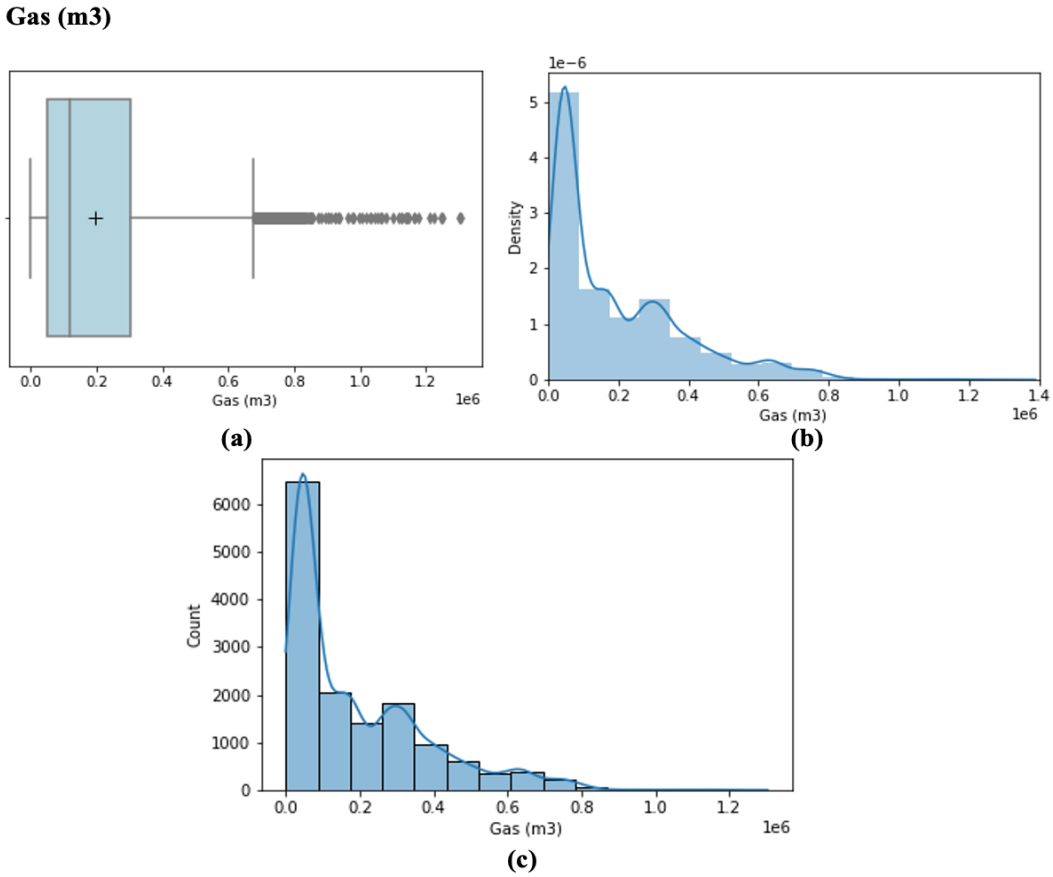
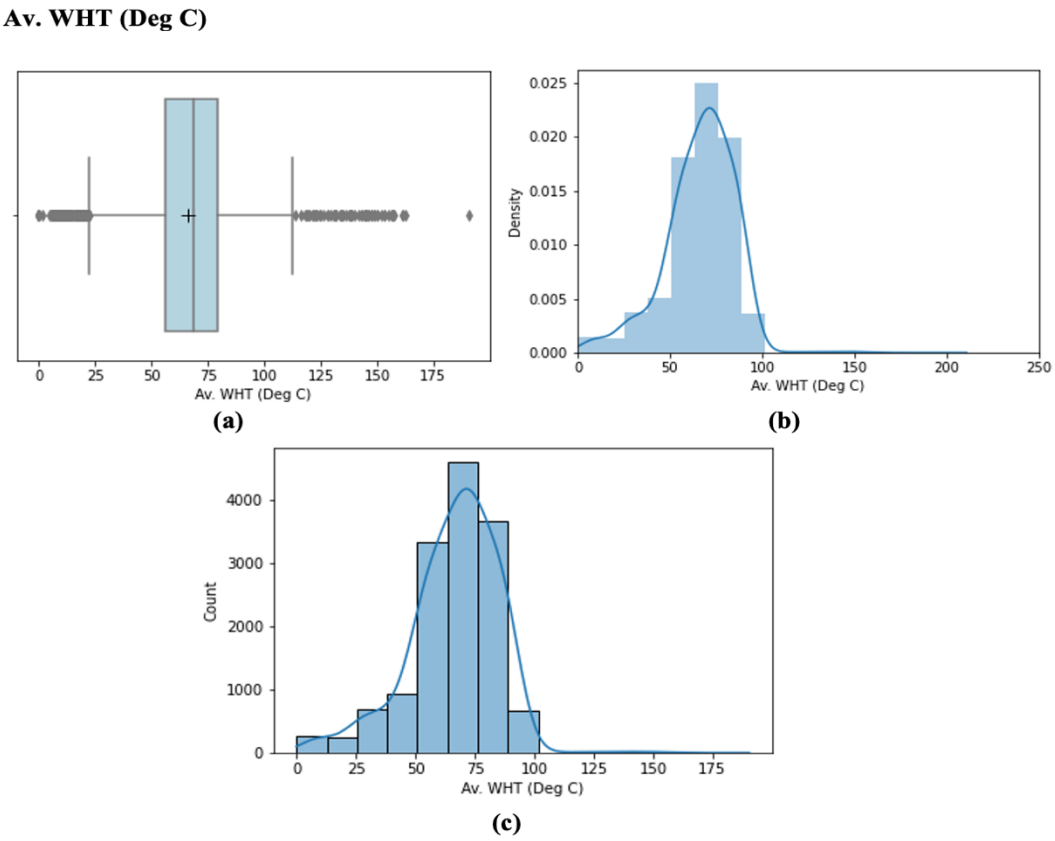


Figure A.37: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Gas (m3) after self-supervised imputation



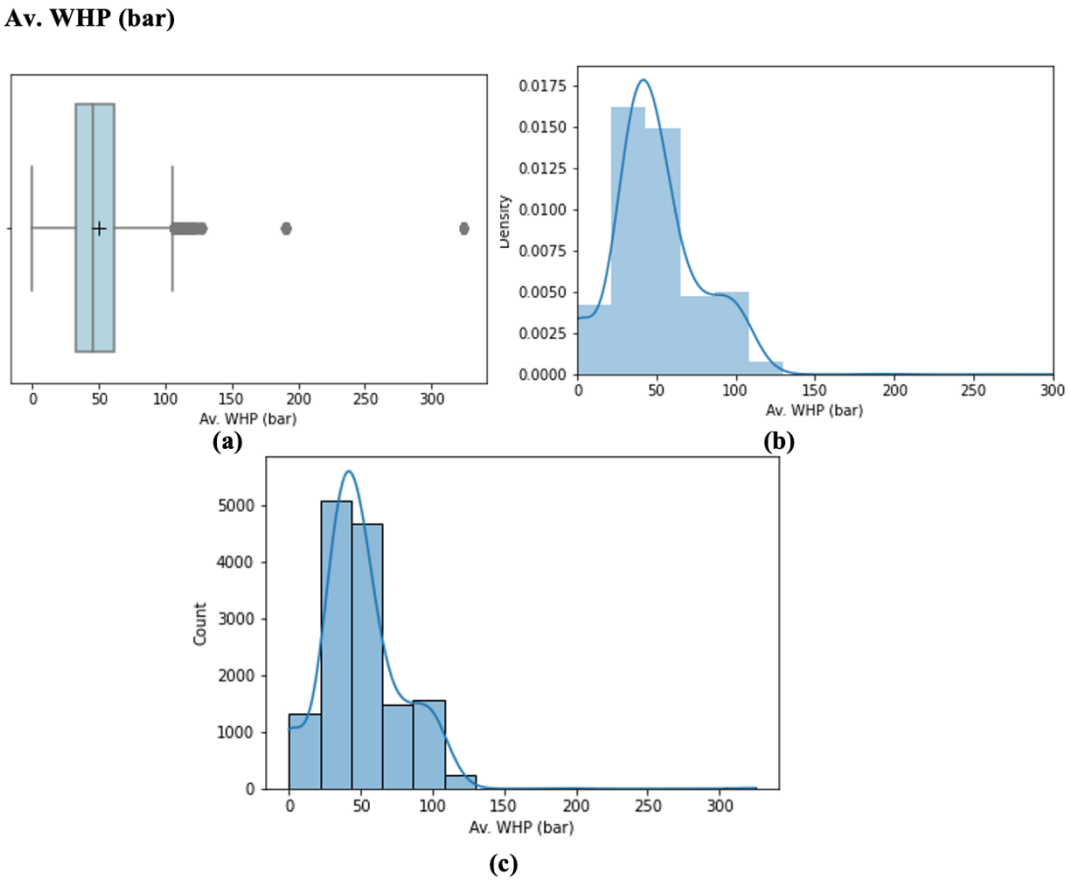


Figure A.39: (a) Boxplot (b) Kernel Density Estimation Plot (c) Histogram for Av. WHP (bar) after self-supervised imputation

Appendix B

Scope of Work

VDR Website Application

The party involved in this development consist of:

Product Owner:	**Ardimas Andi Purwita**
Client:	**Widodo Nugroho - PT Geodwipa Teknika Nusantara (GTN)**
Developers:	**Chan Elizabeth**
	Kotrakona Harinatha Sreeya Reddy
	Vicky Vanessa

Problem Statement:

The client wishes to have a more cost-effective application as an alternative since the application they were using is not cost-effective enough for the client's company.

Scope of work of the project:

The developers are to create a website application frontend prototype for the VDR Website Application to Visualize Production Data for the Oil and Gas Industry. The aims of a fully functional prototype are:

- visualizing oil and gas data,
- being more cost-effective, and
- a predictive model capable of predicting oil and gas production.

The prototype will consist of the features that the client demand which are:

1. Visualization of oil and gas data

   Building a viewer's page which allows the user to choose which file they choose to visualize. Visualization is available in 2D and 3D images, i.e., well logs and seismic data. The file they choose is the file that the user uploaded in the file management, which will be described later.
2. Map application along with the showcase feature

   An application which allows the user to see and upload the location of oil and gas reserves. It shows the data the client has for the reserves for the showcase, e.g., tabular data, snapshots of the location.
3. Prediction model to predict the oil and gas production

   Using pressure and temperature sensor data to predict oil and gas production values so that the user can focus on wells that contain more oil and gas.
4. File management to store the user's files

   The user can store their files and store them into folders. These files can and will be used in the other features.

Approved by

Ardimas Andi Purwita
Product Owner

Widodo Nugroho
Client

Figure B.40: Scope of Work

Proof of Acceptance

VDR Website Application

The party involved in this development consist of:

Product Owner:	**Ardimas Andi Purwita**
Client:	**Widodo Nugroho - PT Geodwipa Teknika Nusantara (GTN)**
Developers:	**Chan Elizabeth**
	Kotrakona Harinatha Sreeya Reddy
	Vicky Vanessa

Statement:

In the meeting on 27 June 2022, the client stated that the product delivered by the developers has fulfilled the requirements stated in the scope of work for the developers' thesis entitled:

- GIS APPLICATIONS IN THE VDR by Chan Elizabeth,
- PREDICTING OIL AND GAS PRODUCTION DATA BY USING DATA SCIENCE by Kotrakona Harinatha Sreeya Reddy, and
- FRONTEND DEVELOPMENT by Vicky Vanessa.

Approved by

Ardimas Andi Purwita
Product Owner

Widodo Nugroho
Client

Figure B.41: Proof of Acceptance

CHAPTER 8

CURRICULUM VITAE

SREEYA KOTRAKONA

COMPUTER SCIENCE

CONTACT

+62 85697378590

sreeya99@gmail.com

Jakarta, Indonesia

TECHNICAL SKILLS

- Python
- HTML
- CSS
- MySQL

PUBLICATIONS

Bina Nusantara International University

Conference: International Seminar on Intelligence Technology and Its Application (ISITIA)

July 2021

"Evaluating Extractive Text Summarization Techniques on News Articles"

WORK EXPERIENCE

Bank Central Asia | Jakarta, Indonesia

Data Analyst Intern

September 2021 - February 2022

- Completed filing, data entry and copying for Metadata Management team
- Cleaned and backed up data to maintain data integrity during extraction, manipulation and processing
- Presented presentations to superiors and teammates regarding project progress and results.

Royal European Internship Council, NHS England, PT Inovasi Telematika | Remote Internship

Project Manager Intern

Februrary 2021 - May 2021

- Achieved project deadlines by coordinating with team members e.g. software developers and digital marketers, to manage performance
- Maintained open communication by presenting regular updates on project status to superiors
- Tracked project and team members performance closely to be able to intervene in mistakes or unfortunate delays
- Scheduled and facilitated meetings between superiors and team members to discuss deliverables, schedules and conflicts

Bina Nusantara International University | Jakarta, Indonesia

Teacher Assistant

September 2020 - January 2021

- Assisted in teaching MySQL database management
- Assisted in teaching non-relational databases, focusing on MongoDB

EDUCATION

Bina Nusantara International University

2018-2022

Bachelor's Degree in Computer Science