Fuse Machines

AI FELLOWSHIP 2023

A Final Project Report on:

Wine Quality Classification

Submitted By:

Nirajan Bekoju

Bishal Adhikari

Rijan Pokhrel

Manoj Khatri

Submitted To:

Fuse Machines

Submission Date:

1^{st} May, 2023

Contents

List of Figures

List of Tables

Chapter 1

Abstract

This report details a machine learning project focused on predicting the quality of red wines using their chemical properties. The dataset used in this project contained 11 features describing the chemical composition of wines, as well as a quality rating ranging from 0 to 10. To improve the performance of machine learning algorithms, various preprocessing techniques such as standard scaler, min max scaler, and logarithmic and boxcox transformation were applied to the data. Exploratory data analysis was also performed, visualizing the data distributions, box plots, and scatter plots to better understand the relationships between the features and the target variable. Several popular machine learning algorithms were trained on the preprocessed data, including logistic regression, SVM, Random forest, decision trees, and boosting algorithms, and their performance was compared. Finally, the best algorithm and preprocessing technique were identified based on performance metrics such as accuracy, precision, recall, and F1 score. The results and conclusions of this project are presented in detail in this report.

Keywords: Wine Quality, Data Analysis, Machine Learning, SVM, Random Forest

Chapter 2

Introduction

2.1 Problem Statement

The dataset is related to red variants of the Portuguese "Vinho Verde" wine.The dataset describes the amount of various chemicals present in wine and their effect on it's quality. The datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones).Task is to predict the quality of wine using the given data.

A simple yet challenging project, to anticipate the quality of wine.

The complexity arises due to the fact that the dataset has fewer samples, & is highly imbalanced.

Data Source : https://archive.ics.uci.edu/ml/datasets/wine+quality

2.2 About Dataset

The dataset contains the following columns:

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates

11. alcohol
12. quality (Targe Variable) : ranges from 0 to 10

The data information is as follow:

```
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1143 non-null   float64
 1   volatile acidity      1143 non-null   float64
 2   citric acid           1143 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide   1143 non-null   float64
 6   total sulfur dioxide  1143 non-null   float64
 7   density               1143 non-null   float64
 8   pH                    1143 non-null   float64
 9   sulphates             1143 non-null   float64
 10  alcohol               1143 non-null   float64
 11  quality               1143 non-null   int64
 12  Id                    1143 non-null   int64
dtypes: float64(11), int64(2)
memory usage: 116.2 KB
```

Figure 2.1: Data Information

2.3 Dataset Statistics

The count plot of the whole dataset on the basis of quality of wine is shown below.

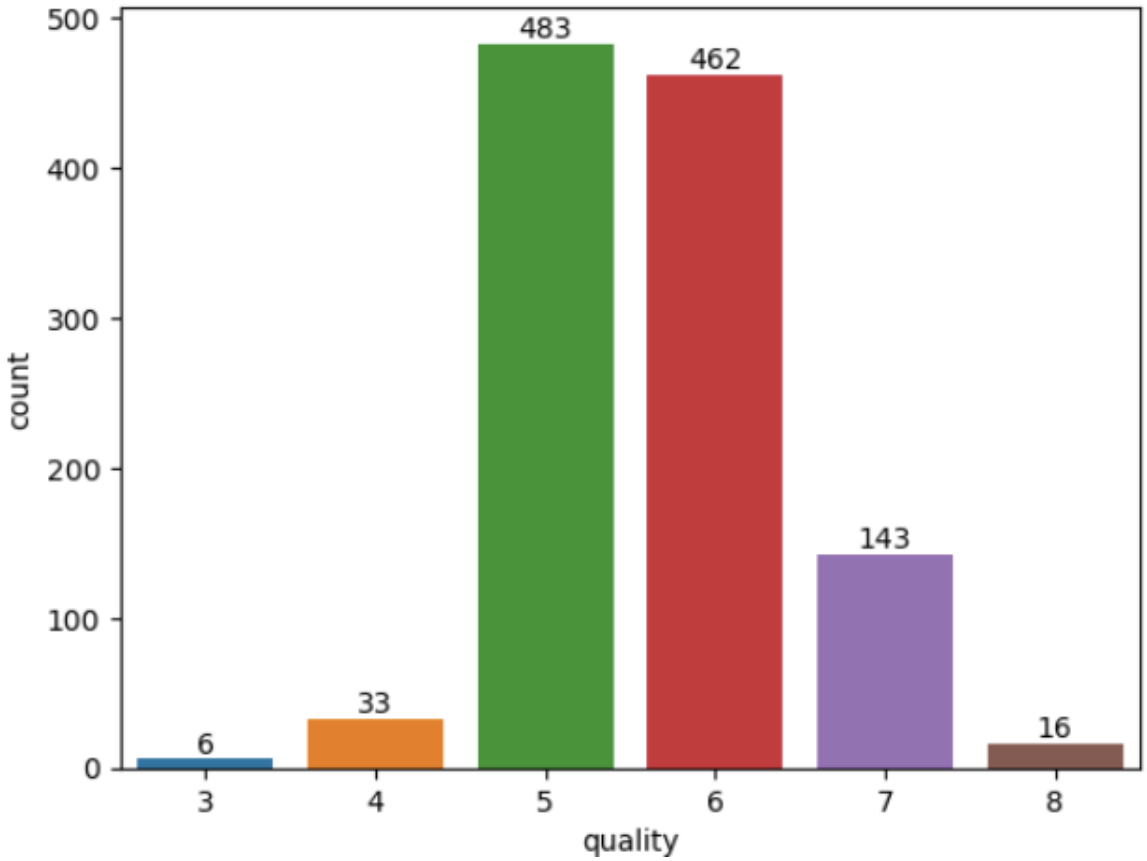

Figure 2.2: Quality Count plot

We can see from above countplot, that the data is highly imbalanced and quality of wine(0, 1, 2, 9, 10) are not present in the dataset.

In order to deal with this imbalanced, at first we are going to merge the quality labels (3 and 4) and (7 and 8) as shown below.

```
def wineQualityTransform(quality):
    wine_quality_transformation = {3: 0, 4: 0, 5:1, 6:2, 7: 3, 8: 3}
    return wine_quality_transformation[quality]

data_df["quality"] = data_df["quality"].apply(wineQualityTransform)
print(data_df.quality.value_counts())
```

Figure 2.3: Code for transformation of quality attribute

The resulting count plot is as shown below:

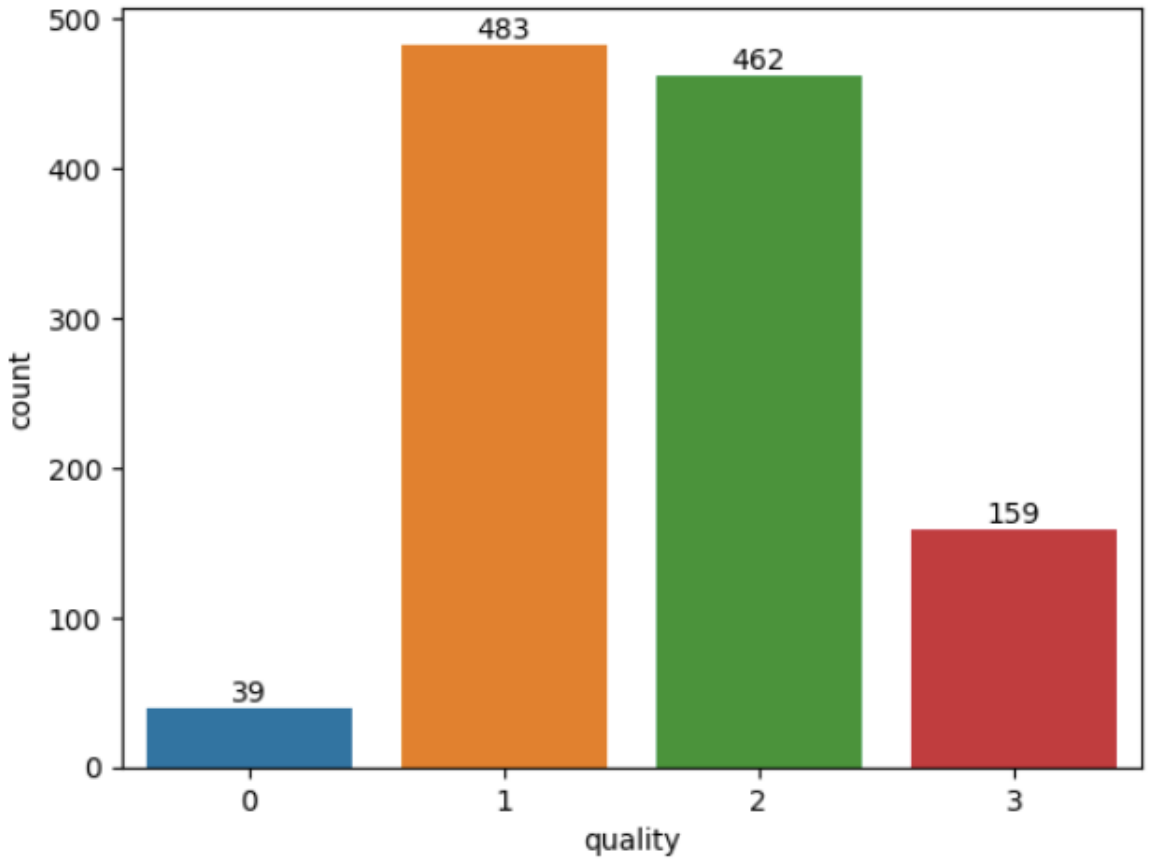

Figure 2.4: Quality Count plot after transformation

2.3.1 Train, validation and test dataset

We are going to use 80%, and 20% of the dataset as training, validation and test data. The countplot are as shown below:

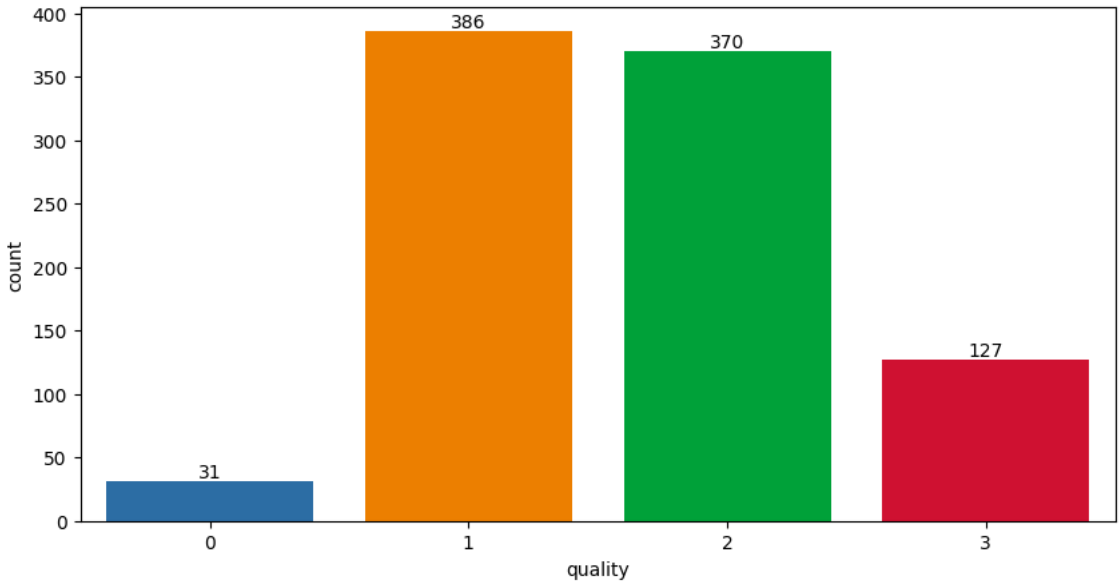

Figure 2.5: Train Quality Count plot after transformation

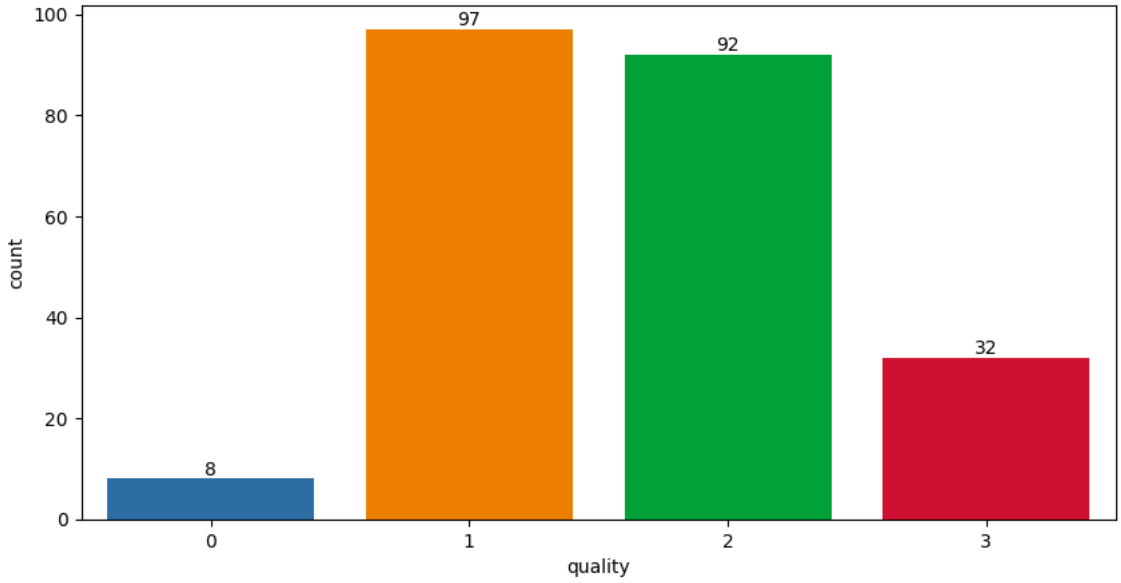

Figure 2.6: Test Quality Count plot after transformation

2.4 Expected Outcome

We expect the output to be the list of probabilites of belonging to a class[0, 1, 2 or 3]. Among these, the class with higher probability is out predicted class.

Chapter 3

Project Management

3.1 Agile Development Methodology

To develop the wine quality classification, we decided to use Agile methodology. Agile methodology is an iterative approach to software development that emphasizes flexibility, collaboration, and rapid prototyping. We chose Agile methodology because it is well-suited to projects that require frequent feedback and adaptation to changing requirements.

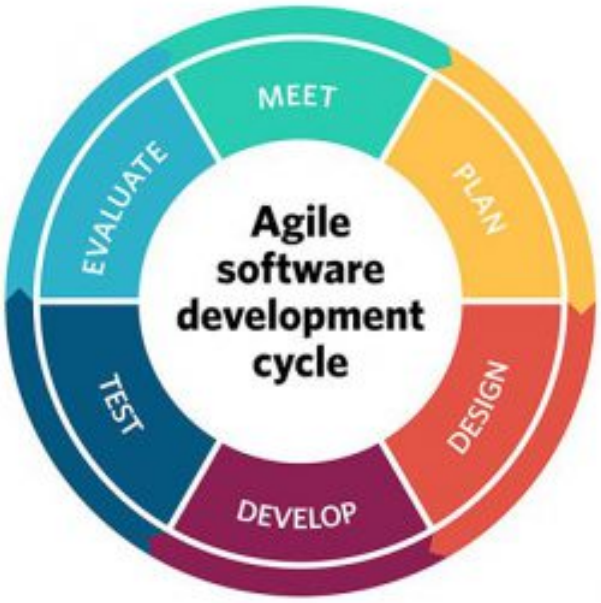

Figure 3.1: Agile Development Methodology

3.2 Trello

We have used Trello's default agile board for task management.

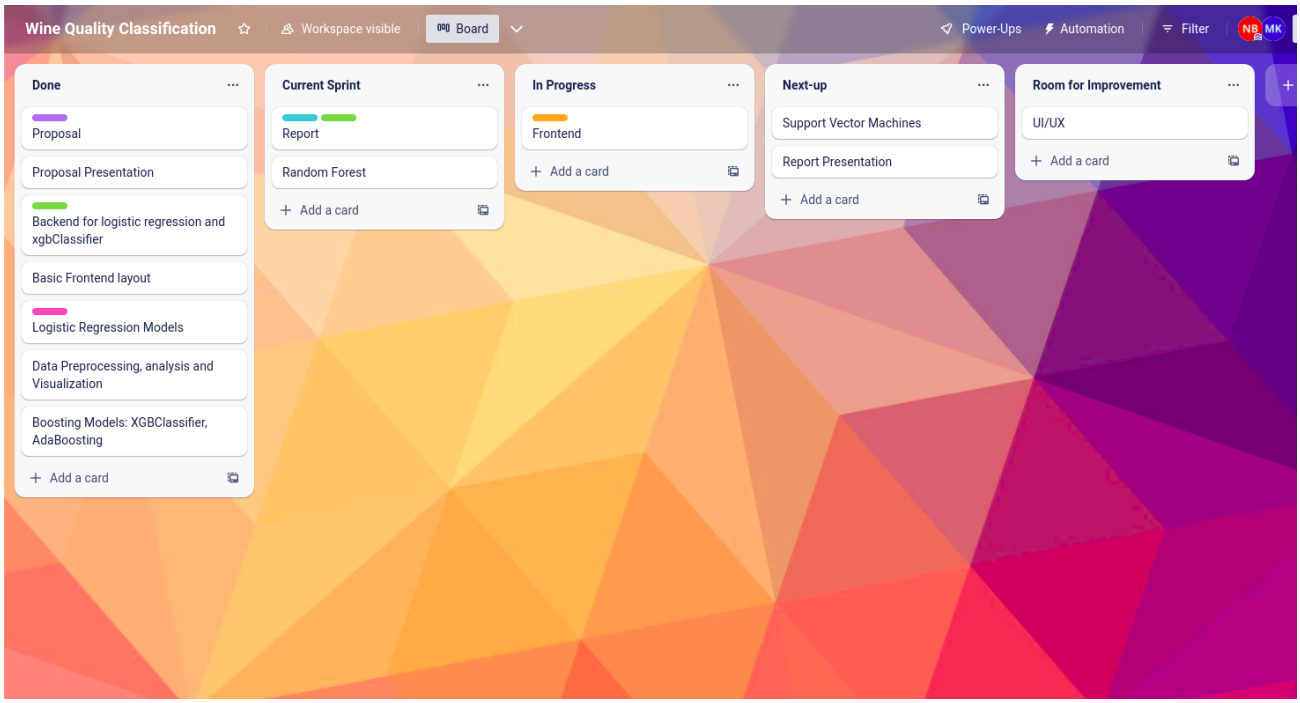

Figure 3.2: Trello

3.3 Discord

We have used Discord server for effective communication and team meetings.

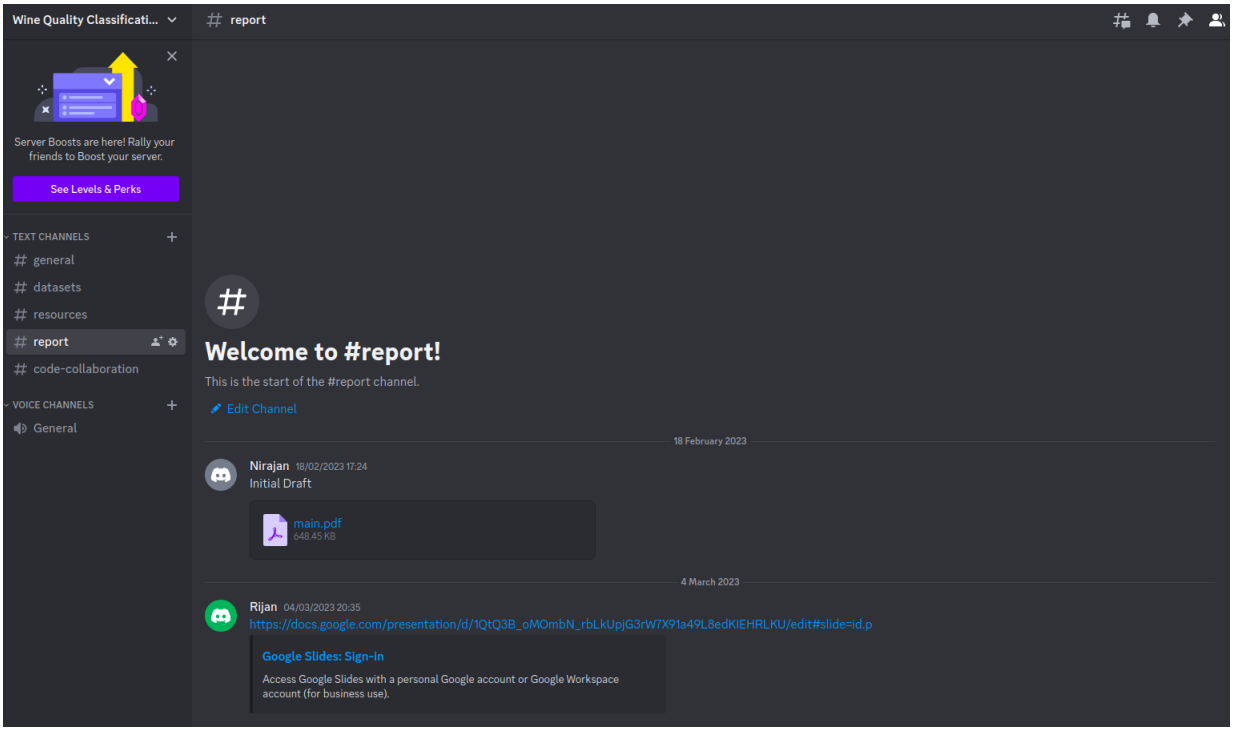

Figure 3.3: Discord

3.4 Excel

We have used Excel for experiment tracking.

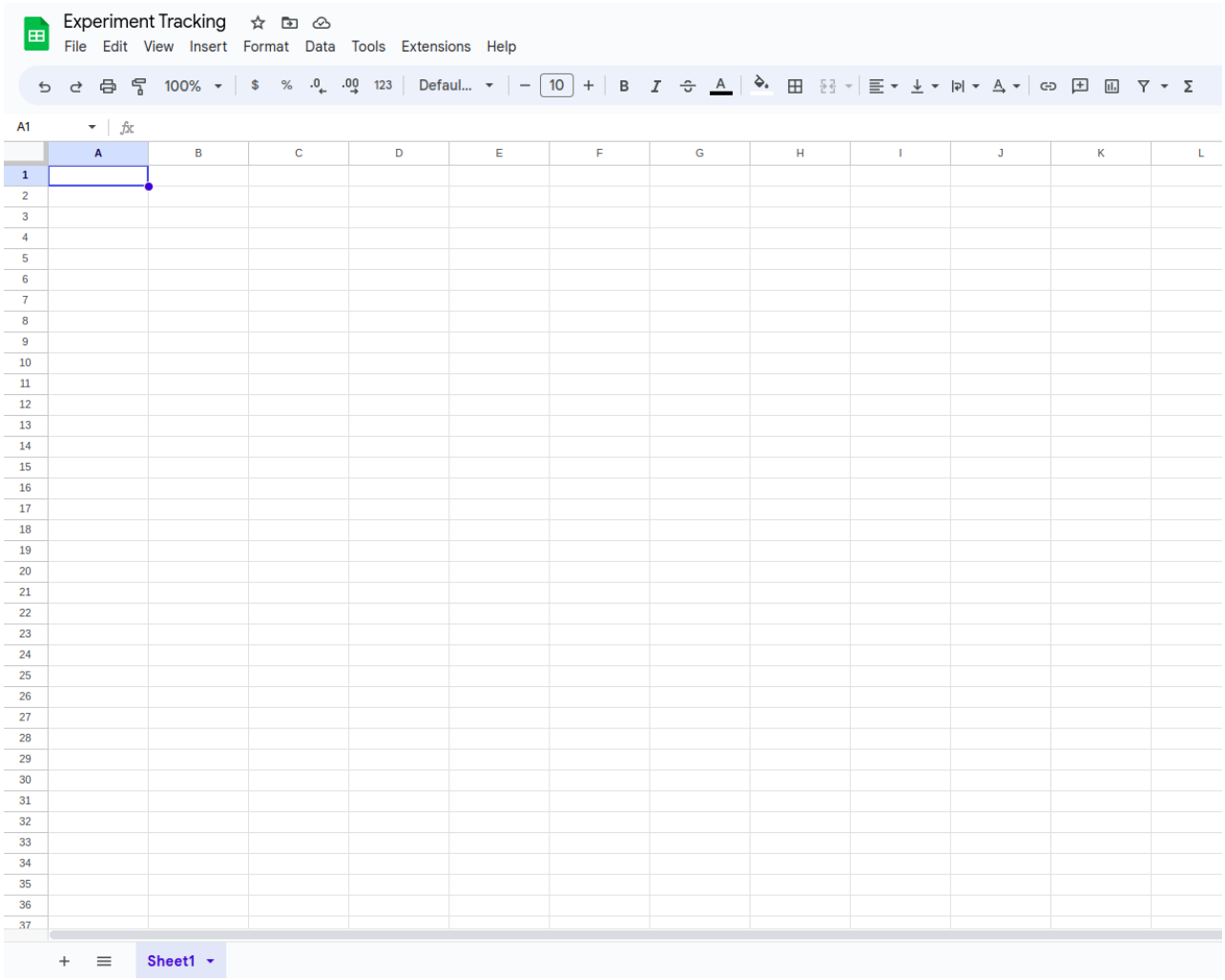

Figure 3.4: Google Sheet

Chapter 4

Feature Engineering and Exploratory Data Analysis

This chapter discusses the process of feature engineering, which involves techniques such as feature scaling and feature transformation are described, which can help to normalize the range and distribution of features. Data visualization is also emphasized as a tool for gaining insights into the data, with examples given of how boxplots, distribution plots, and heat maps can be used to identify patterns and relationships in the data.

4.1 Dataset Exploration

Basic statistics of each features are shown below:

stats	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide
mean	8.311	0.5313	0.264	2.532	0.0869	15.615
std	1.747	0.179	0.196	1.355	0.0477	10.250
min	4.600	0.120000	0.000	0.900	0.0120	1.00
25%	7.100	0.39250	0.0900	1.900	0.070	7.000
50%	7.900	0.5200	0.2500	2.200	0.0790	13.000
75%	9.100	0.6400	0.4200	2.6000	0.090	21.000
max	15.900	1.580	1.00	15.50	0.6110	68.00

Table 4.1: Data Description 1

stats	total sulfur dioxide	density	pH	sulphates	alcohol
mean	45.914698	0.996730	3.311015	0.657708	10.442111
std	32.782130	0.001925	0.156664	0.170399	1.082196
min	6.000000	0.990070	2.740000	0.330000	8.400000
25%	21.000000	0.995570	3.205000	0.550000	9.500000
50%	37.000000	0.996680	3.310000	0.620000	10.200000
75%	61.000000	0.997845	3.400000	0.730000	11.100000
max	289.000000	1.003690	4.010000	2.000000	14.900000

Table 4.2: Data Description 2

The correlation between data is shown in following heat map.

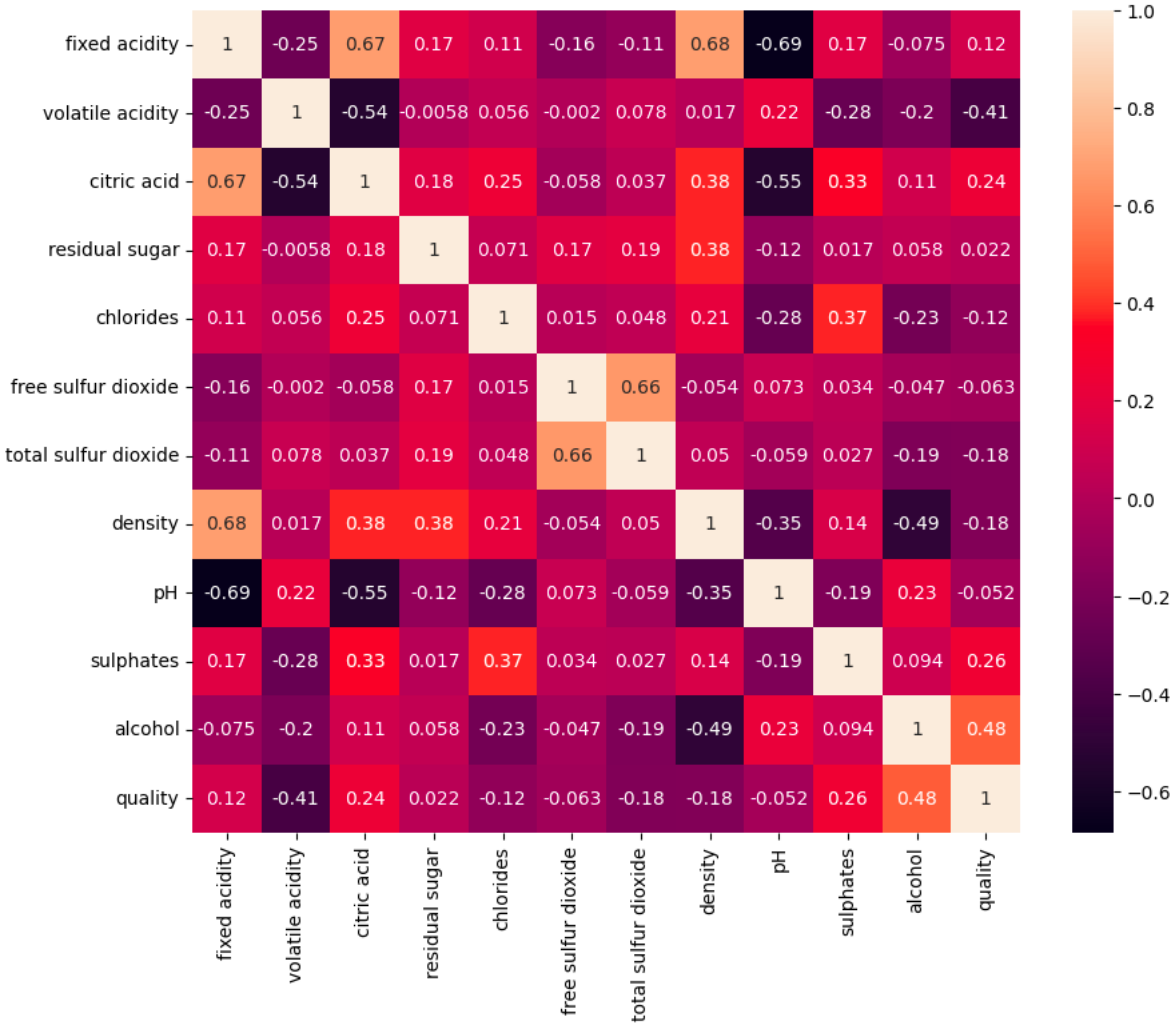

Figure 4.1: Correlation Heat Map

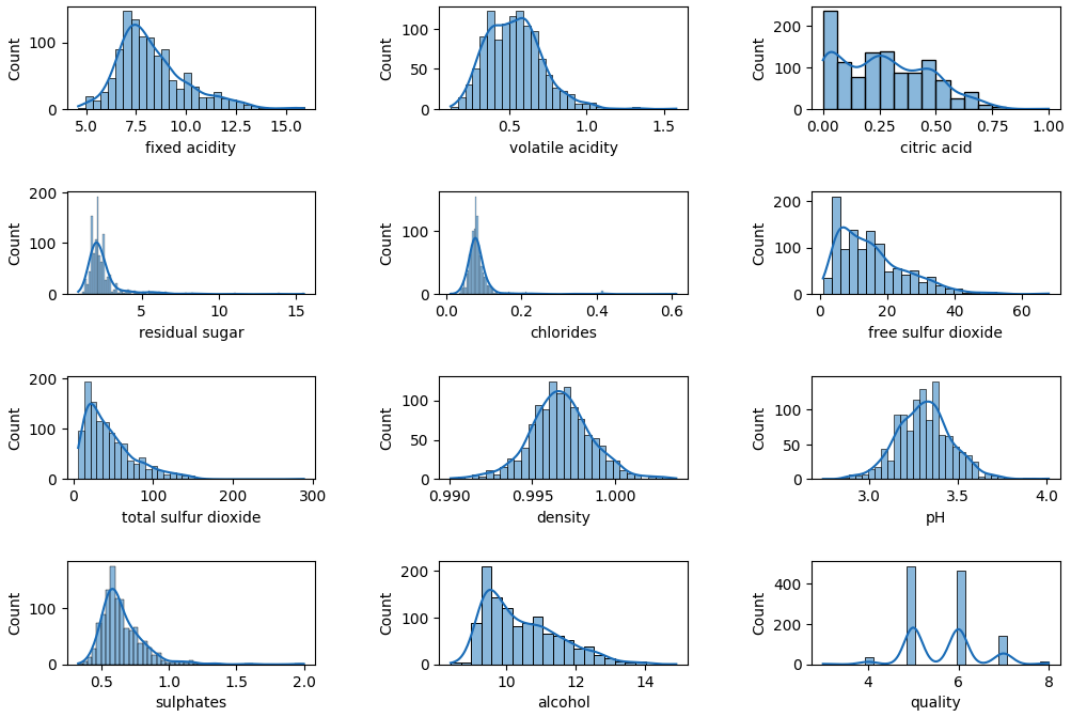

Figure 4.2: Distribution Plot of the entire datasets

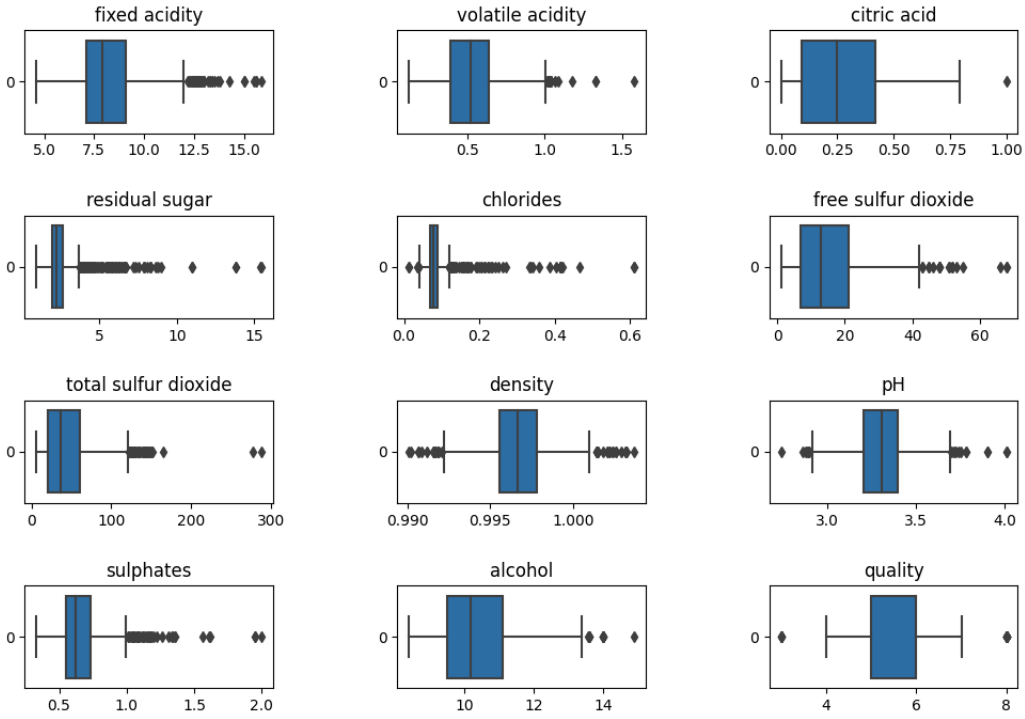

Figure 4.3: Box Plot of the entire datasets

4.2 Min Max Scaling

Min max scaling was used as it scales all the data features in the range of 0 and 1 due to which it becomes easier to transform the data using log transformation and boxcox transformation.

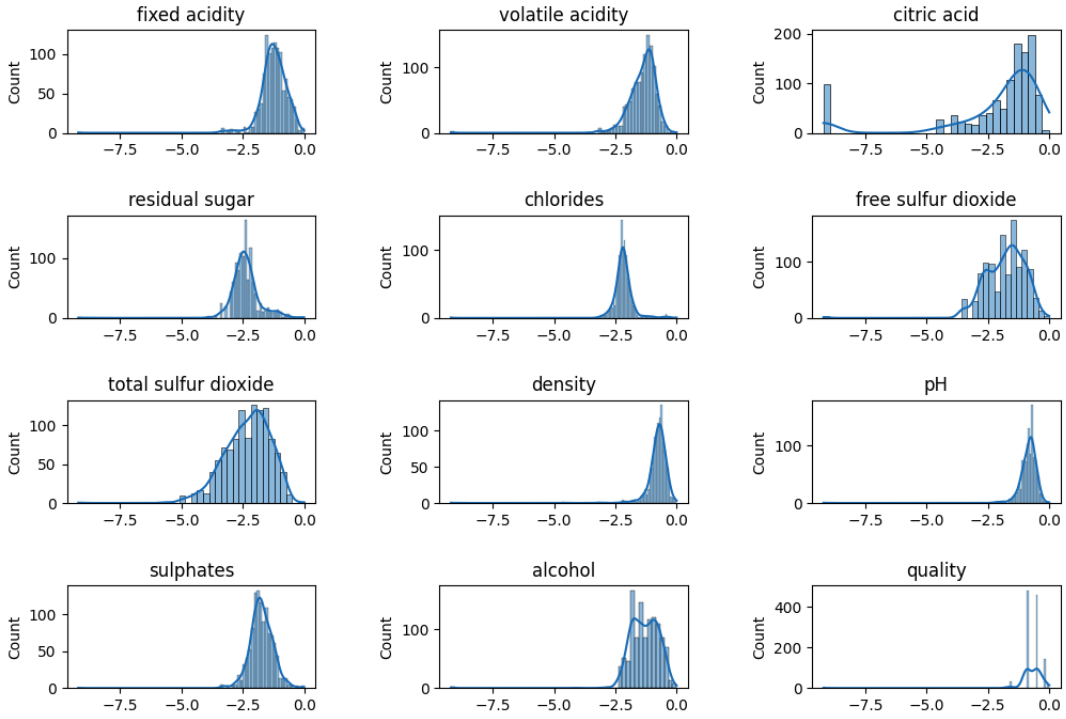

Figure 4.4: Distribution Plot of log transformed data after minmax scaling

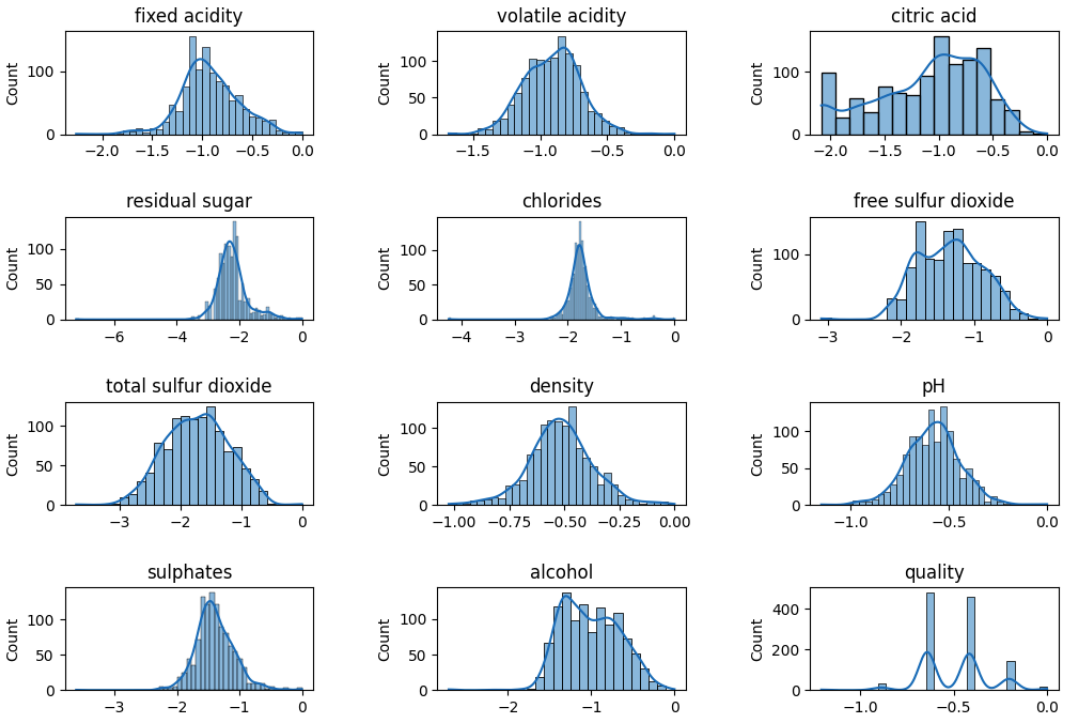

Figure 4.5: Distribution Plot of boxcox transformed data after minmax scaling

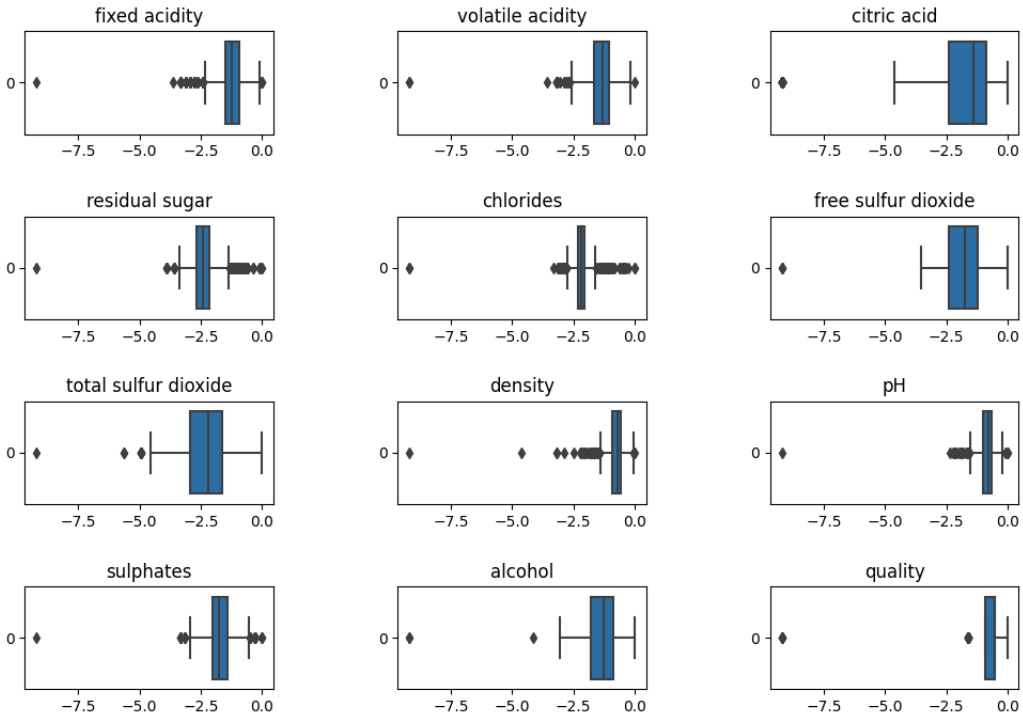

Figure 4.6: Box Plot of log transformed data after minmax scaling

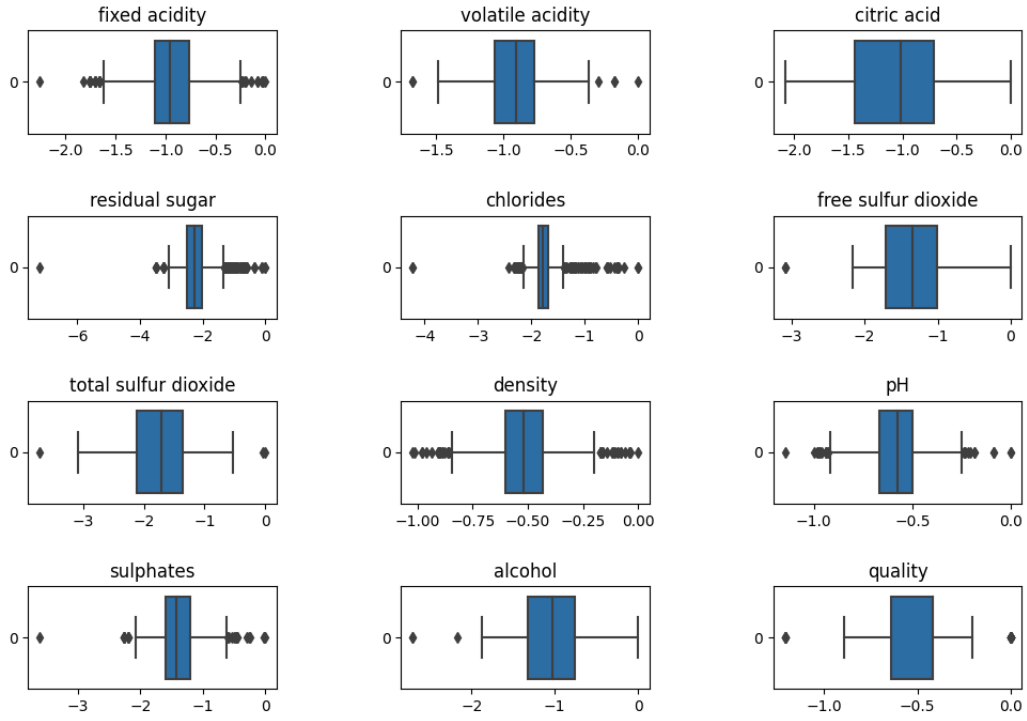

Figure 4.7: Box Plot of boxcox transformed data after minmax scaling

Chapter 5

Model Development and Evaluation

5.1 Model Development

Our data is highly imbalanced as discussed in chapter 1. So, we are going to use different methods like undersampling, oversampling, smote and class weights to handle the imbalanced data.

Since this is a classification task, we can use various machine learning models (e.g : Logistic regression, Decision Trees, Ensemble methods, etc.)

5.2 Model Evaluation

We have highly imbalanced data, so we are going to use macro F1-score as the primary model evaluation metrics. Different other evaluation like roc, auc-score, precision and recall, etc. will also be studied.