Exploratory Data Analysis and Logistic Regression for Predicting Wine Quality: A Data-Driven Approach

A

PROJECT REPORT

BY

Md. Sujaul Islam Santo

ID No.: 1707026

Reg. No.: 46777

Session: 2016-17

Department of Food Technology and Rural Industries
Bangladesh Agricultural University
Mymensingh-2202

December 2022

Exploratory Data Analysis and Logistic Regression for Predicting Wine Quality: A Data-Driven Approach

A

PROJECT REPORT

BY

Md. Sujaul Islam Santo

ID No.: 1707026

Reg. No.: 46777

Session: 2016-17

Submitted to

Department of Food Technology and Rural Industries

Bangladesh Agricultural University, Mymensingh-2202

In partial fulfillment of the requirement for the degree of

Bachelor of Science
in
Food Engineering

Department of Food Technology and Rural Industries
Bangladesh Agricultural University
Mymensingh-2202

December 2022

Exploratory Data Analysis and Logistic Regression for Predicting Wine Quality: A Data-Driven Approach

A

PROJECT REPORT

BY

Md. Sujaul Islam Santo

ID No.: 1707026

Reg. No.: 46777

Session: 2016-17

Approved as to style and content by:

Dr. Afzal Rahman
Supervisor

Miss. Asmaul Husna Nupur
Co-Supervisor

Member

Member

Member

Professor Dr. Md. Abdul Alim

Chairman, Defense Committee

Head, Department of Food Technology and Rural Industries

Bangladesh Agricultural University, Mymensingh-2202

August, 2024

ACKNOWLEDGEMENTS

First of all, all praises and deepest sense of gratefulness to the Almighty “Allah”, the merciful, the omnipotent, the omniscient, the omnipresent, the beneficent and the supreme ruler of the Universe, who enabled us to complete as well as to submit the thesis for the degree of Bachelor of Science (B.Sc.) in Food Engineering.

We feel proud and pleased to express our deepest sense of gratitude and indebtedness to our honorable supervisor Dr. Mr. Afzal Rahman, Associate Professor, Department of Food Technology and Rural Industries, Bangladesh Agricultural University (BAU), Mymensingh for his affection, inspiration, constant guidance, valuable suggestions and instruction throughout the tenure of this research work and in the preparation of the thesis.

We would like to express our respect to the honorable co-supervisor Miss. Asmaul Husna Nupur, Assistant Professor, Department of Food Technology and Rural Industries, Bangladesh Agricultural University (BAU), Mymensingh for his co-operation, patient guidance, constructive comments, valuable suggestions and encouragement during the study period.

We express our sincere appreciation and gratefulness and unbounded gratitude to all the respected teachers of Department of Food Technology and Rural Industries, Bangladesh Agricultural University (BAU), Mymensingh for their patient guidance and intense supervision.

We deeply owe to all our relatives, well-wishers, former and existing classmates and friends for their co-operations, inspirations and affectionate feelings for the successful completion of our study. We also express our cordial thanks to all well-wishers for their help and mental support during this study.

Finally, we express our ever gratefulness and indebtedness to our beloved parents, brothers and sisters for their great sacrifice, endless prayers, blessings and support for my higher study.

The Author

ABSTRACT

The quality of wine plays a pivotal role in determining its marketability and consumer preference. Accurate classification of wine quality based on chemical properties can aid producers in optimizing production processes and ensuring high standards. This research focuses on predicting wine quality using logistic regression and exploratory data analysis (EDA) to examine critical patterns and correlations within the dataset. The UCI Wine Quality dataset, a widely recognized benchmark in predictive modeling, serves as the basis for this study. It includes key physicochemical attributes such as fixed acidity, volatile acidity, citric acid, residual sugar, alcohol, pH, and chlorides, alongside sensory quality ratings assigned by expert tasters [1] [2].

The methodology comprises three main phases: data preprocessing to normalize and handle missing values, EDA to uncover relationships among features, and logistic regression to classify wine quality into binary categories (e.g., high-quality vs. low-quality). Accuracy an evaluation metrics is employed to assess model performance. Findings indicate that alcohol content, volatile acidity, and sulphates significantly impact wine quality ratings. The logistic regression model achieves a predictive accuracy of approximately 80%, demonstrating its efficacy as a classification tool [3] [4].

The study contributes to the literature on predictive modeling for wine quality by providing insights into the influence of chemical attributes on sensory evaluations. Furthermore, it highlights the utility of logistic regression for interpretable and efficient wine quality prediction. Future research could incorporate advanced machine learning models and larger datasets to improve classification accuracy and generalizability [5].

TABLE OF CONTENTS

CHAPTER – 1

INTRODUCTION

Background

Importance of Wine Quality in the Beverage Industry

Wine quality is a fundamental determinant of consumer satisfaction and the economic success of the wine industry. The evaluation of wine involves both chemical and sensory analyses, with attributes such as alcohol content, acidity, and aromatic profiles playing a critical role in defining quality standards. High-quality wine not only attracts premium pricing but also enhances brand reputation, making accurate quality assessment a priority for producers and [1] [2]. Traditional quality evaluation methods rely heavily on expert tasters, which can be subjective and resource-intensive.

Role of Machine Learning in Predicting Wine Quality

Machine learning (ML) offers innovative solutions for the objective assessment of wine quality by identifying patterns in complex datasets. By leveraging physicochemical properties, ML algorithms can classify wines into quality categories with high accuracy, thus reducing dependence on subjective sensory evaluations. Logistic regression, a widely used supervised learning method, is particularly effective in binary classification tasks like identifying "good" versus "bad" wine, making it a suitable choice for this study [3] [5].

Problem Statement

Challenges in Accurately Classifying Wine Quality

The classification of wine quality poses several challenges due to its multifaceted nature. Quality is influenced by a combination of chemical properties, environmental factors, and production techniques. Existing datasets often exhibit class imbalance, with fewer samples for high-quality wines, complicating model training. Furthermore, the interpretation of feature importance remains critical for deriving actionable insights, necessitating transparent and interpretable models like logistic regression [2] [4].

Research Objectives

1. **To explore the relationships between wine attributes and quality:** This objective focuses on performing exploratory data analysis (EDA) to uncover significant correlations and trends among physicochemical properties such as pH, alcohol, sulphates, and their impact on wine quality [1] [3].

2. **To predict wine quality using logistic regression:** The second objective aims to build and evaluate a logistic regression model capable of accurately classifying wine samples into binary quality categories (e.g., high-quality vs. low-quality) based on the identified features [5] [6].

Scope and Significance

Applications of Wine Quality Prediction in Industry and Research

The prediction of wine quality has significant applications in both the wine industry and academic research. In the wine industry, machine learning models can assist producers by optimizing production processes, ensuring consistency, and maintaining high-quality standards while reducing the costs and labor associated with traditional sensory evaluation methods. These predictive models can help identify optimal production conditions and enhance quality control mechanisms by integrating various physicochemical attributes of wine, such as alcohol content, acidity, and sugar levels [7] [8].

From a research standpoint, the use of machine learning techniques offers valuable insights into the relationship between chemical compositions and sensory properties, providing a deeper understanding of how various factors influence wine quality. This research contributes to advancements in food science, particularly in the study of food chemistry and sensory science, where machine learning models help identify the most significant predictors of quality, which can be used for improving product development and optimization in the food and beverage sectors [9] [10].

Furthermore, this study highlights the utility of interpretable machine learning (ML) models, which are especially important for stakeholders who require actionable insights. These models not only offer predictions but also provide transparency into how specific features affect the outcome, making them useful for decision-makers in the wine industry who seek to implement data-driven strategies for quality improvement and process optimization [11] [12].

CHAPTER – 2

LITERATURE REVIEW

Overview of Wine Quality Studies

The Kaggle Wine Quality dataset has been extensively used in academic research as a benchmark for exploring wine classification problems [7]. Cortez et al. [1] utilized various machine learning algorithms to model wine preferences based on physicochemical properties, highlighting the dataset's suitability for predictive tasks. Studies have applied models such as decision trees, support vector machines (SVMs), and logistic regression to predict wine quality, revealing insights into the influence of alcohol, acidity, and sulphates on classification accuracy [3] [5].

Research by Fernandes et al. [7] demonstrated the effectiveness of ensemble methods, like Random Forest, in handling class imbalances commonly observed in the dataset. These studies collectively underscore the dataset's relevance in machine learning and data science, providing a foundation for the current exploration of logistic regression.

Exploratory Data Analysis (EDA)

Importance in Understanding Datasets

EDA is crucial in identifying patterns, relationships, and anomalies within a dataset. It serves as the initial step in understanding data structure, feature distributions, and correlations, enabling informed model selection and feature engineering. Previous work on the UCI dataset employed EDA techniques to visualize distributions of pH, residual sugar, and alcohol, uncovering their impact on wine quality ratings [3].

Examples from Prior Research

In their study, Cortez et al. [1] leveraged visualizations like boxplots and histograms to analyze the variability in wine attributes, identifying alcohol and volatile acidity as critical features. Similarly, Fernandes et al. [7] demonstrated that EDA could guide feature selection by quantifying attribute importance using correlation heatmaps and pairwise plots.

Logistic Regression in Classification

Why Logistic Regression is Suitable for This Problem

Logistic regression is well-suited for wine quality classification due to its interpretability and ability to handle binary classification tasks effectively. Its probabilistic framework enables clear decision boundaries, making it ideal for distinguishing between high-quality and low-quality wines [2] [4]. Logistic regression's simplicity allows for easy implementation and provides insights into feature importance through coefficient analysis, which is valuable for understanding the role of physicochemical properties in quality determination [5].

Comparison with Other Classification Methods

While logistic regression offers interpretability, other algorithms like SVMs and Random Forests provide improved accuracy in certain cases. SVMs excel in handling high-dimensional spaces and class separability but may lack the transparency of logistic regression [3]. Random Forests, known for their robustness and ability to handle nonlinear relationships, often outperform simpler models like logistic regression but may sacrifice interpretability [7].

Relevance of Features

Key Chemical Properties Influencing Wine Quality

Several physicochemical attributes significantly influence wine quality, as shown in prior studies. Alcohol content is consistently identified as a positive indicator of quality, likely due to its association with flavor profiles [1] [7]. Sulphates and volatile acidity also play crucial roles, where higher sulphate levels contribute to enhanced aroma and taste, and controlled volatile acidity prevents off-flavors [5] [4].

Fernandes et al. [7] highlighted pH and chlorides as secondary factors affecting wine stability and preservation. These findings emphasize the importance of understanding chemical attributes in predicting and improving wine quality, aligning with the goals of the present study.

CHAPTER – 3

METHODOLOGY

Dataset Overview

Description of the Kaggle Wine Quality Dataset

The Kaggle Wine Quality dataset is a well-documented resource widely used in predictive modeling and machine learning research [7]. It consists of physicochemical properties (input features) of red and white wines, such as **fixed acidity**, **volatile acidity**, **citric acid**, **residual sugar**, **chlorides**, **free sulfur dioxide**, **total sulfur dioxide**, **density**, **pH**, **sulphates**, and **alcohol**.

- **Fixed Acidity:** Fixed acidity in wine refers to the total concentration of non-volatile acids, such as tartaric, malic, and lactic acids. It contributes to the wine's taste, structure, and stability. The presence of fixed acidity provides a refreshing tartness and helps in balancing the wine's flavor profile, making it an important component for overall taste and aging potential.
- **Volatile Acidity:** Volatile acidity in wine refers to the concentration of acids that can vaporize, such as acetic acid. While small amounts of volatile acidity can contribute to the wine's complexity, excessive levels can lead to undesirable vinegar-like aromas and flavors. Controlling volatile acidity is crucial in maintaining the wine's quality and preventing off-flavors.
- **Citric Acid:** Citric acid is a natural acid found in wine, often originating from the grapes themselves. It can contribute to the wine's freshness and fruitiness, enhancing its overall flavor profile. However, excessive levels of citric acid can lead to a sour or overly acidic taste, so its presence must be balanced to ensure a harmonious wine.
- **Residual Sugar:** Residual sugar refers to the natural sugars remaining in the wine after fermentation. It can contribute to the wine's sweetness, body, and mouthfeel. Wines with higher residual sugar levels tend to have a sweeter taste, while those with lower levels are

drier. The presence of residual sugar is important in defining the wine's style and can influence its pairing with food.

- **Chlorides:** Chlorides in wine, originating from the soil and winemaking process, can impact the wine's taste and stability. While small amounts of chlorides can contribute to the wine's complexity, excessive levels can lead to a salty or briny taste, negatively affecting the overall flavor profile.
- **Free Sulfur Dioxide:** Free sulfur dioxide is used in winemaking as a preservative to prevent oxidation and microbial spoilage. It plays a crucial role in maintaining the wine's freshness, stability, and aging potential. Proper levels of free sulfur dioxide are essential for ensuring the wine's longevity and quality.
- **Total Sulfur Dioxide:** Total sulfur dioxide encompasses both the free and bound forms of sulfur dioxide in wine. It serves as a preservative and antioxidant, protecting the wine from spoilage and oxidation. Monitoring total sulfur dioxide levels is important for ensuring the wine's shelf life and overall quality.
- **Density:** Density in wine is a measure of its mass per unit volume and can provide insights into the wine's alcohol content and potential sweetness. It contributes to the wine's body and mouthfeel, with higher density wines often exhibiting a richer and more viscous texture.
- **pH:** The pH level of wine influences its acidity, microbial stability, and color stability. It plays a crucial role in shaping the wine's overall taste and mouthfeel. Proper pH levels are important for ensuring the wine's balance and longevity.
- **Sulphates:** Sulphates, often in the form of sulfur dioxide, are used in winemaking as a preservative and antimicrobial agent. They help prevent oxidation and microbial spoilage, contributing to the wine's stability and longevity. However, excessive levels of sulphates can lead to undesirable aromas and flavors, so their presence must be carefully managed.
- **Alcohol:** The alcohol content of wine contributes to its body, texture, and overall flavor profile. It influences the wine's warmth, mouthfeel, and perceived sweetness. The alcohol level is an important factor in defining the wine's style and can impact its aging potential.

Data Preprocessing

Handling Missing Values

The dataset consists of 13 columns: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, quality, and an Id column, with a total of 1143 rows [7]. A thorough inspection confirms that there are no missing values in the dataset. However, consistency checks for outliers and anomalies (e.g., extreme values or unexpected distributions) are performed to ensure data quality [3] [4].

Scaling and Normalization

The features vary significantly in scale; for instance:

- Alcohol typically ranges from ~8-14%,
- Residual sugar can vary from 0 to 65 g/L.

To ensure that no feature disproportionately influences the model, Min-Max Normalization or `StandardScaler` is applied:

$$\boldsymbol{X(scaled) = \frac{X - min(X)}{max(X) - min(X)}}$$

This transformation rescales all features to a uniform range, such as [0, 1], while preserving relationships and ensuring compatibility with logistic regression's optimization algorithm [5] [2].

By maintaining consistent scaling, the influence of larger-magnitude features like *density* or *total sulfur dioxide* is neutralized, improving model performance and interpretability [8].

Exploratory Data Analysis

Summary Statistics

Descriptive statistics, such as mean, median, and standard deviation, are computed for each feature to identify variations and trends is shown in Table 1. These statistics provide insights into the distribution and variance of each chemical property [8].

	count	mean	std	min	25%	50%	75%	max
fixed acidity	1143	8.311111	1.747595	4.6	7.1	7.9	9.1	15.9
volatile acidity	1143	0.531339	0.179633	0.12	0.3925	0.52	0.64	1.58
citric acid	1143	0.268364	0.196686	0	0.09	0.25	0.42	1
residual sugar	1143	2.532152	1.355917	0.9	1.9	2.2	2.6	15.5
chlorides	1143	0.086933	0.047267	0.012	0.07	0.079	0.09	0.611
free sulfur dioxide	1143	15.615486	10.250486	1	7	13	21	68
total sulfur dioxide	1143	45.914698	32.78213	6	21	37	61	289
density	1143	0.99673	0.001925	0.9901	0.9956	0.9967	0.99785	1.0037
pH	1143	3.311015	0.156664	2.74	3.205	3.31	3.4	4.01
sulphates	1143	0.657708	0.170399	0.33	0.55	0.62	0.73	2
alcohol	1143	10.442111	1.082196	8.4	9.5	10.2	11.1	14.9
quality	1143	5.657043	0.805824	3	5	6	6	8

Table 1 - *Descriptive statistics of each chemical property.*

Visualization of Relationships

Scatter Plots

- Scatterplot of **Fixed acidity vs Citric acid** is shown in Figure 1.

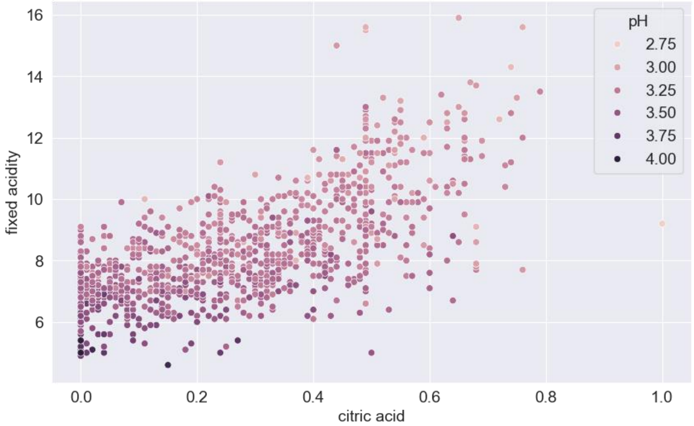


***Figure 1** - Fixed acidity vs Citric acid.*

There is a positive correlation between them. As the fixed acidity increased there will be more presence of citric acid. If the fixed acidity of a solution, such as wine, is increased, it can lead to a more tart or sour taste. with increase in fixed acidity pH value also decrease. A decrease in pH value indicates an increase in the acidity of a solution.

- Scatterplot of **Fixed acidity vs Density** is shown in Figure 2.

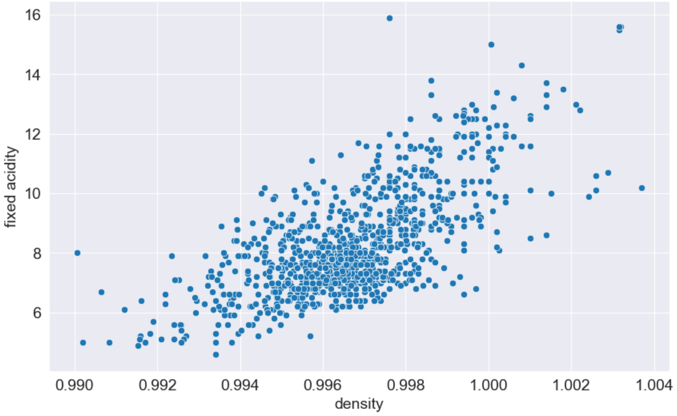


***Figure 2** - Fixed acidity vs Density.*

As the fixed acidity of wine increases, the density of the wine also tends to increase. This relationship is due to the presence of acids in the wine, which can contribute to its overall density.

- Scatterplot of **Fixed acidity vs pH** is shown in Figure 3.

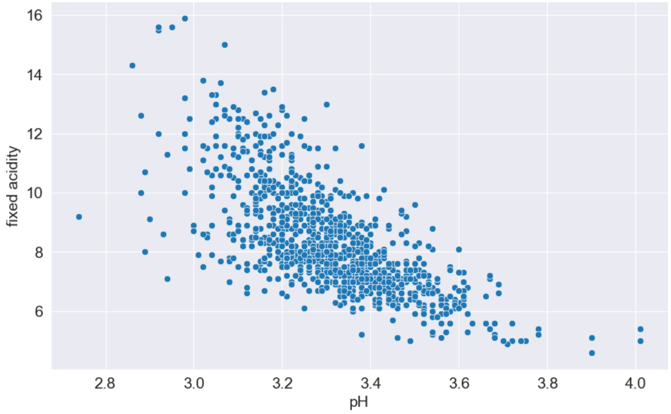


Figure 3 *- Fixed acidity vs pH.*

There is a negative correlation between them. In wine, fixed acidity and pH are closely related. As fixed acidity increases, the pH of the wine tends to decrease, making the wine more acidic. Conversely, as fixed acidity decreases, the pH tends to increase, making the wine less acidic.

- Scatterplot of **Volatile acidity vs Citric acid** is shown in Figure 4.

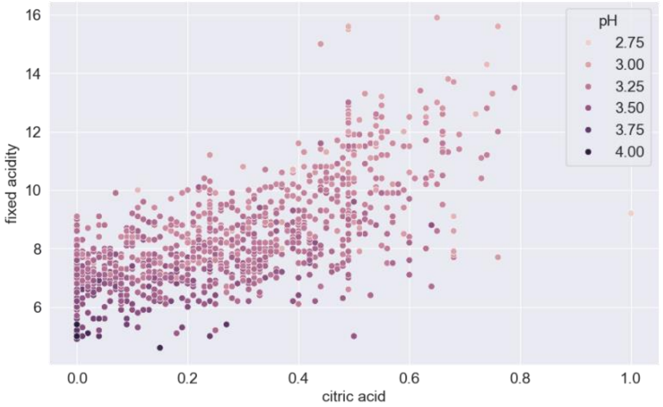


Figure 4 *- Volatile acidity vs Citric acid.*

There is negative correlation. As citric acid is a non-volatile acid and more its presence in wine lead to less presence of volatile acidity.

- Scatterplot of **Citric acid vs Chlorides** is shown in Figure 5.

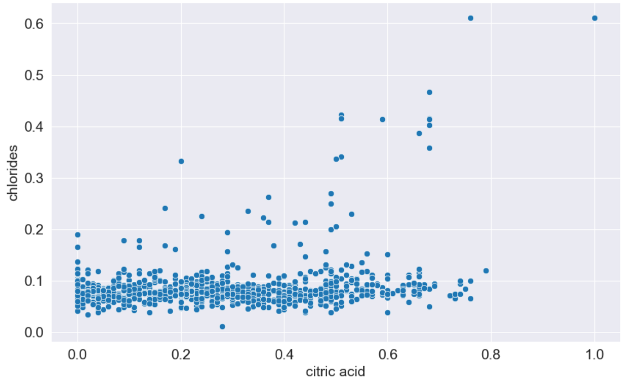


Figure 5 *- Citric acid vs Chlorides.*

- Scatterplot of **Citric acid vs Density** is shown in Figure 6.

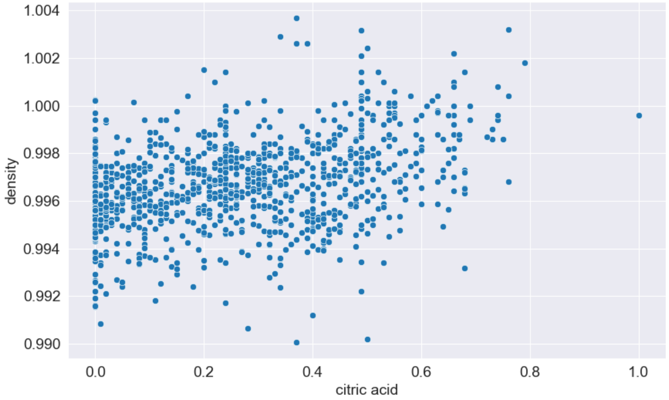


Figure 6 *- Citric acid vs Density.*

- Scatterplot of **Citric acid vs Sulphates** is shown in Figure 7.

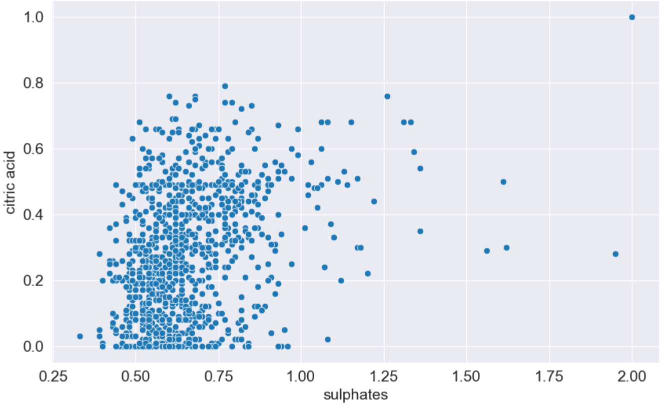


Figure 7 *- Citric acid vs Sulphates.*

- Scatterplot of **Residual sugar vs Density** is shown in Figure 9.

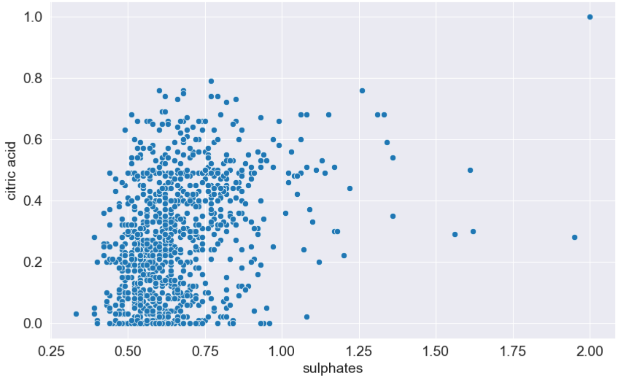


Figure 9 *- Residual sugar vs Density.*

- Scatterplot of **Chlorides vs Sulphates** is shown in Figure 10.

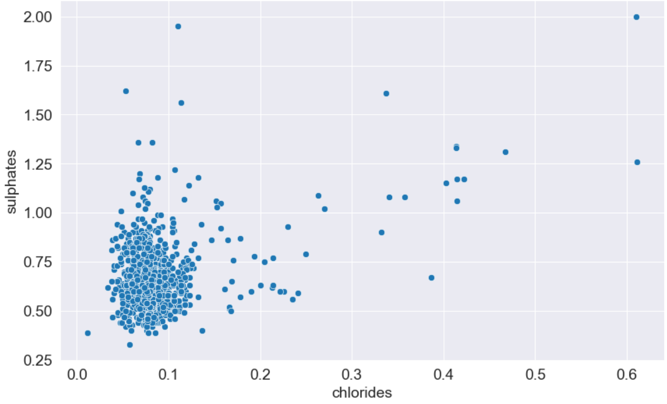


Figure 10 *- Chlorides vs Sulphates.*

- Scatterplot of **Free Sulfur dioxide vs Total Sulfur dioxide** is shown in Figure 11.

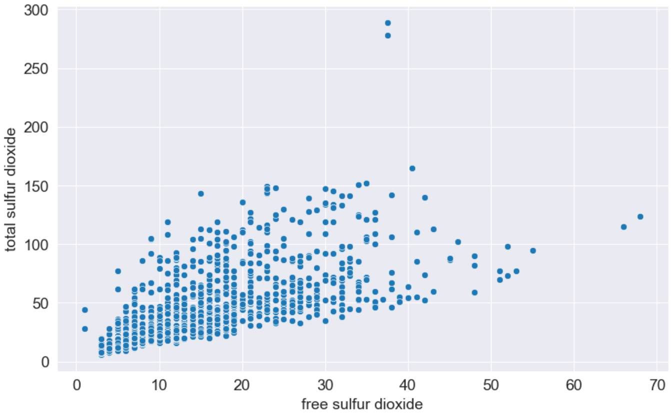


Figure 11 *- Free Sulfur dioxide vs Total Sulfur dioxide.*

Free sulfur dioxide (SO2) and total sulfur dioxide are correlated in the sense that the total SO2 level includes both the free and bound forms of SO2. Free SO2 is the portion that is active and able to protect the wine from oxidation and microbial spoilage, while bound SO2 is chemically linked to other compounds and is not as readily available for these protective functions. Therefore, the total SO2 level is a sum of both the free and bound forms, and understanding their correlation is important for managing the wine's stability and shelf life.

- Strip plot of **Alcohol vs Quality** is shown in Figure 12.

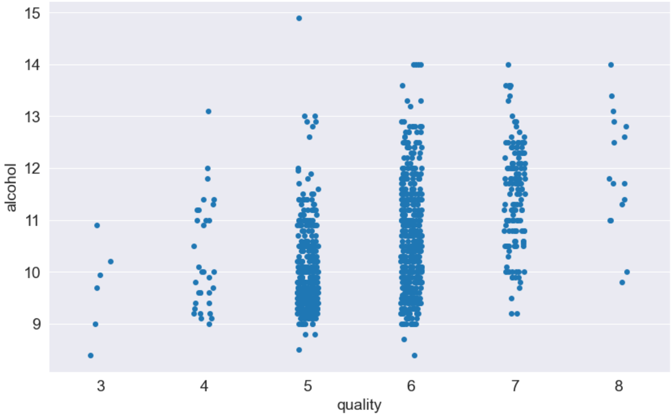


Figure 12 *- Alcohol vs Quality.*

- Strip plot of **Citric acid vs Quality** is shown in Figure 13.

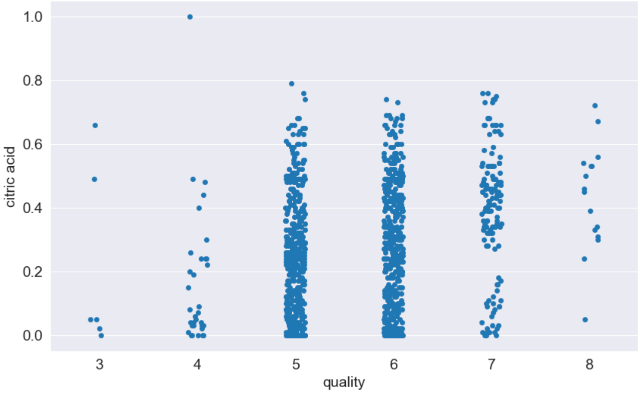


Figure 13 *- Citric acid vs Quality.*

- Strip plot of **Sulphates vs Quality** is shown in Figure 14.

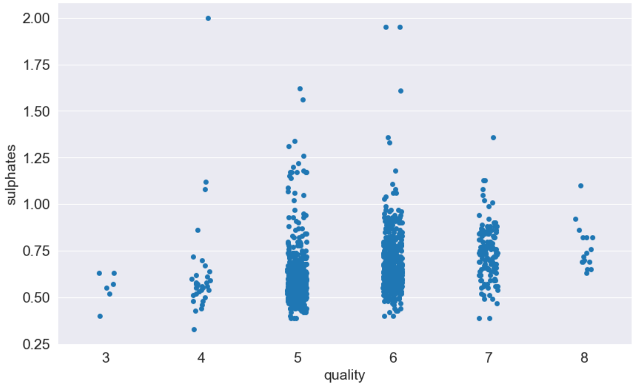


Figure 14 *- Sulphates vs Quality.*

Correlation Heatmaps:

- Heatmap of **Correlations** between columns is shown in Figure 15.

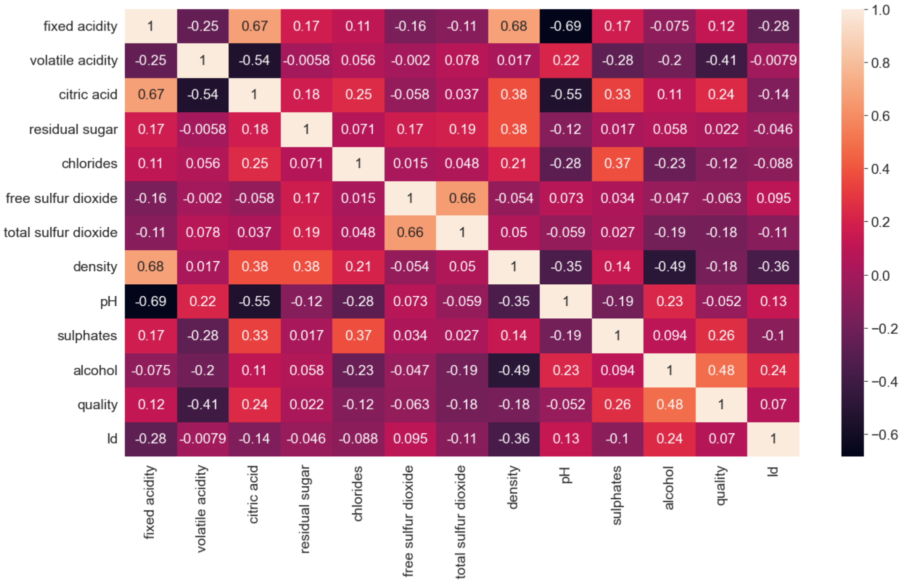


Figure 15 *- Correlations between columns.*

Histogram Plots:

- Histogram Plot of **Fixed acidity** with median value 7.9 is shown in Figure 16.

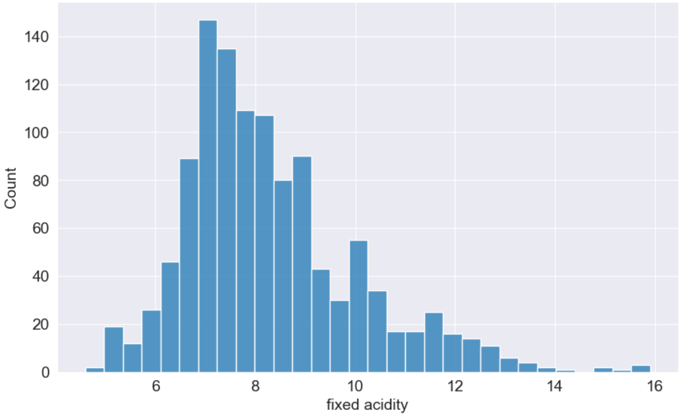


Figure 16 *- Histogram Plot of Fixed acidity.*

- Histogram Plot of **Volatile acidity** with median value 0.52 is shown in Figure 17.

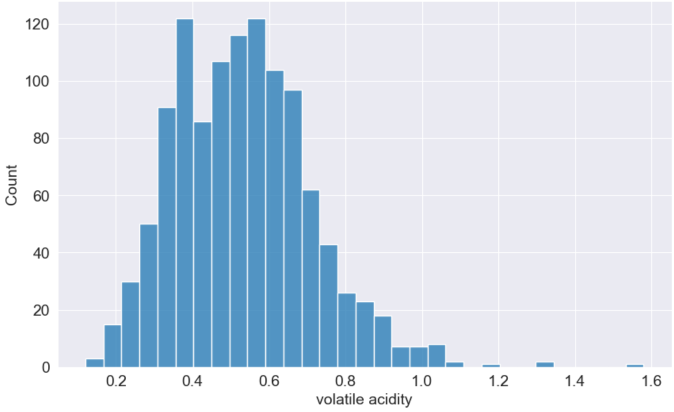


Figure 17 *- Histogram Plot of Volatile acidity.*

- Histogram Plot of **Citric acid** with median value 0.25 is shown in Figure 18.

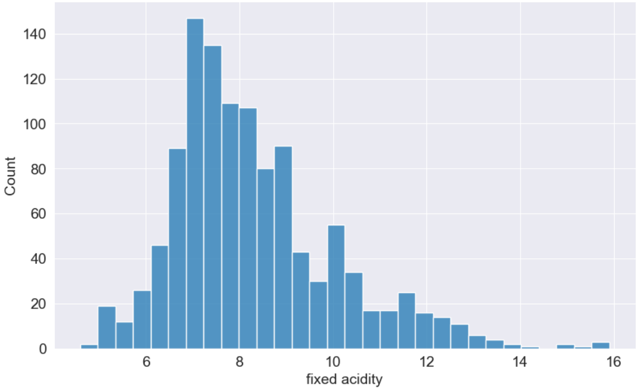


Figure 18 *- Histogram Plot of Citric acid.*

- Histogram Plot of **Residual sugar** with median value 2.2 is shown in Figure 19.

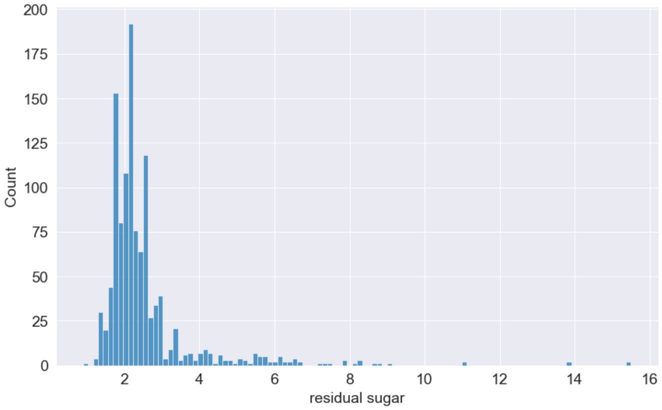


Figure 19 *- Histogram Plot of Residual sugar.*

- Histogram Plot of **Chlorides** with median value 0.079 is shown in Figure 20.

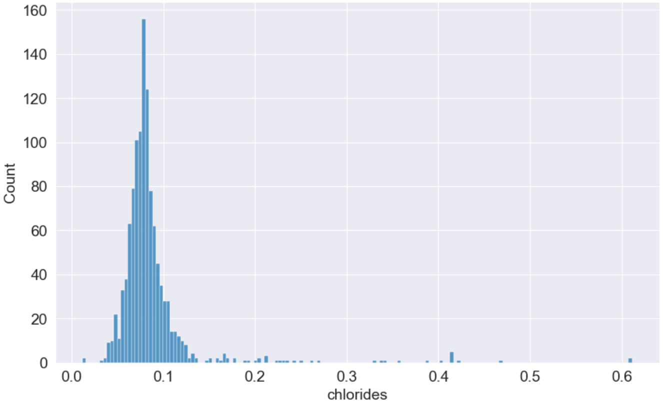


Figure 20 *- Histogram Plot of Chlorides.*

- Histogram Plot of **Free Sulfur dioxide** with median value 13.0 is shown in Figure 21.

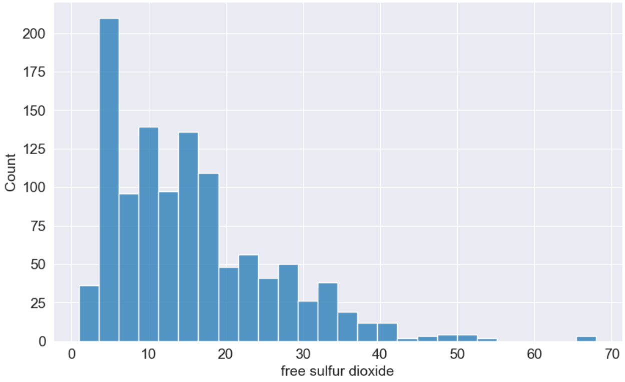


Figure 21 *- Histogram Plot of Free Sulfur dioxide.*

- Histogram Plot of **Total Sulfur dioxide** with median value 37.0 is shown in Figure 22.

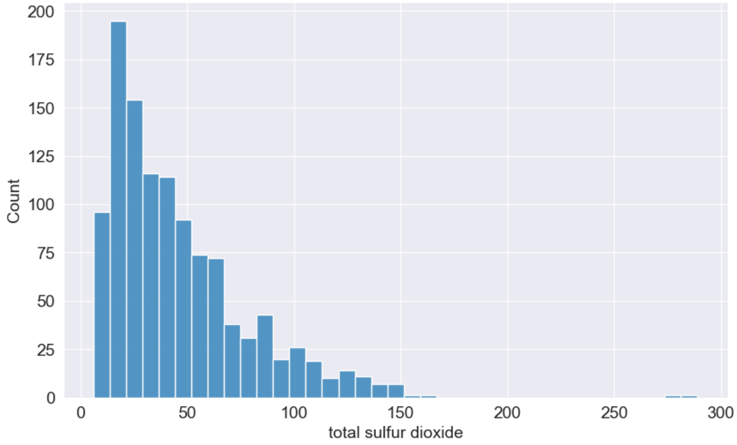


Figure 22 *- Histogram Plot of Total Sulfur dioxide.*

- Histogram Plot of **Density** with median value 0.99668 is shown in Figure 23.

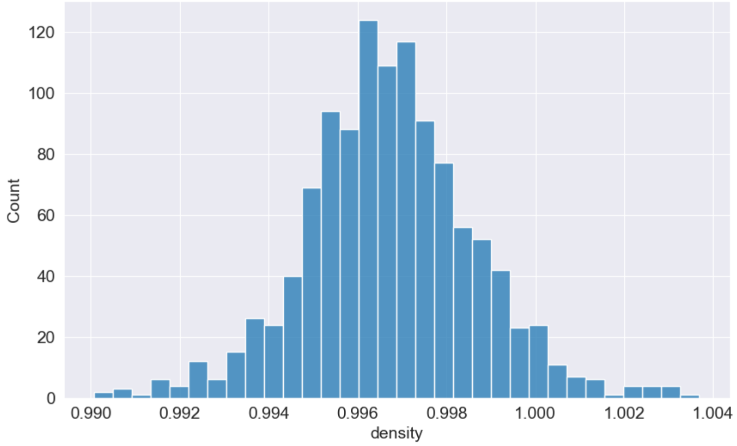


Figure 23 *- Histogram Plot of Density.*

- Histogram Plot of **pH** with median value 3.31 is shown in Figure 24.

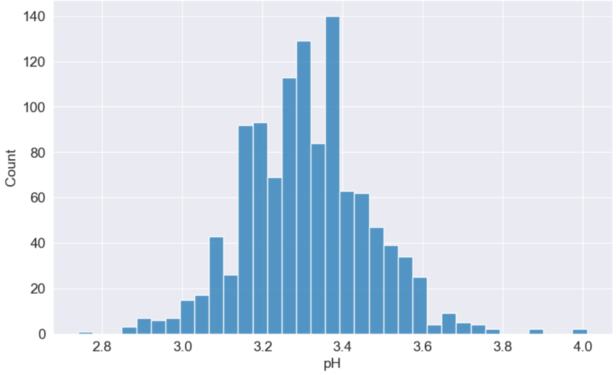


Figure 24 *- Histogram Plot of pH.*

- Histogram Plot of **Sulphates** with median value 0.62 is shown in Figure 25.

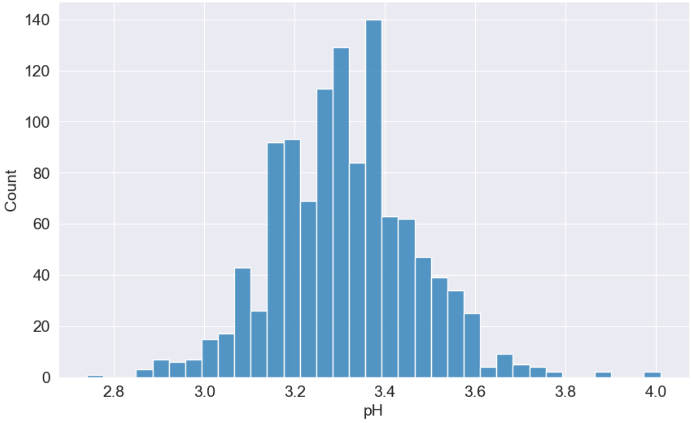


Figure 25 *- Histogram Plot of Sulphates.*

- Histogram Plot of **Alcohol** with median value 10.2 is shown in Figure 26.

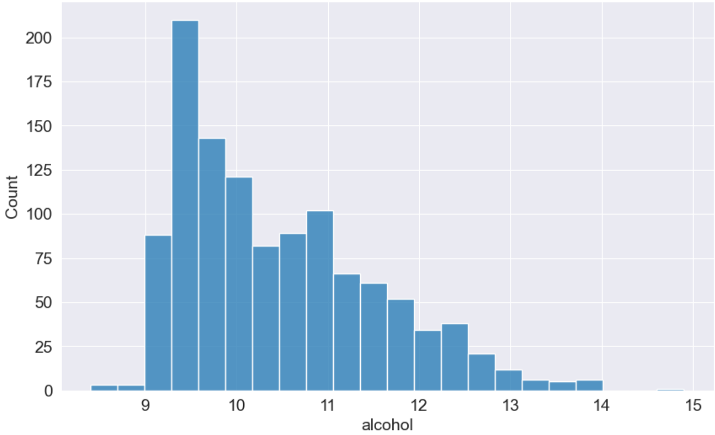


Figure 26 *- Histogram Plot of Alcohol.*

- Histogram Plot of **Quality** with median value 10.2 is shown in Figure 27.

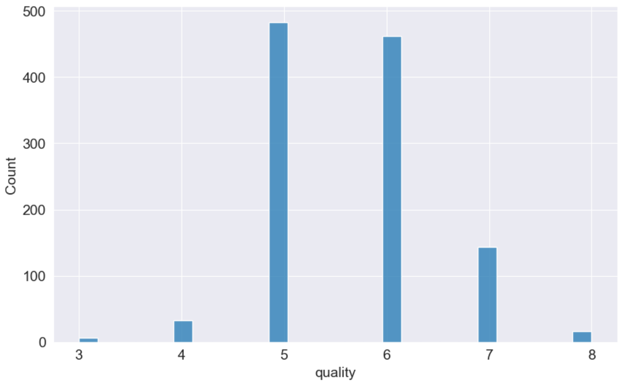


Figure 27 *- Histogram Plot of Quality.*

Model Development

Training, Validation and Test Sets

Data Splitting

To ensure effective training and evaluation of the logistic regression model, the dataset is divided into training, validation, and test sets using an 80-20% split for training-validation and test data. The training-validation subset is further split into 75% for training and 25% for validation. This splitting strategy ensures a robust evaluation while preventing data leakage. The following code snippet demonstrates the approach:

```
from sklearn.model_selection import train_test_split

train_val_df, test_df = train_test_split(raw_df_copy, test_size=0.2,
random_state=42)
train_df, val_df = train_test_split(train_val_df, test_size=0.25,
random_state=42)

print('train_df.shape : ', train_df.shape)  # (685, 12)
print('val_df.shape : ', val_df.shape)      # (229, 12)
print('test_df.shape : ', test_df.shape)    # (229, 12)
```

The dataset contains 12 features and one target column (quality). The features are 'fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol'. These splits ensure that the training, validation, and test subsets retain the data structure and feature distribution.

Data Preparation

The training, validation, and test datasets are prepared as follows:

```
# Separate inputs and target column
input_cols = list(train_df.columns)[:-1]
target_col = 'quality'

# Extract inputs and targets
train_inputs = train_df[input_cols].copy()
train_targets = train_df[target_col].copy()
val_inputs = val_df[input_cols].copy()
val_targets = val_df[target_col].copy()
test_inputs = test_df[input_cols].copy()
test_targets = test_df[target_col].copy()
```

This organization supports model training, hyperparameter tuning, and performance evaluation on separate datasets.

Scaling Numeric Features

Feature scaling is crucial to ensure that features contribute equally to the model's loss and avoid optimization challenges due to varying feature magnitudes. Using `MinMaxScaler`, features are scaled to a range of 0 to 1:

```
from sklearn.preprocessing import MinMaxScaler

numeric_cols =
train_inputs.select_dtypes(include=np.number).columns.to_list()
scaler = MinMaxScaler()

# Fit scaler on the entire dataset
scaler.fit(raw_df_copy[numeric_cols])

# Scale training, validation, and test inputs
train_inputs[numeric_cols] = scaler.transform(train_inputs[numeric_cols])
val_inputs[numeric_cols] = scaler.transform(val_inputs[numeric_cols])
test_inputs[numeric_cols] = scaler.transform(test_inputs[numeric_cols])
```

The following ranges for numeric columns were derived:

- **Minimum values**: [4.6, 0.12, 0.0, 0.9, 0.012, 1.0, 6.0, 0.99007, 2.74, 0.33, 8.4]
- **Maximum values**: [15.9, 1.58, 1.0, 15.5, 0.611, 68.0, 289.0, 1.00369, 4.01, 2.0, 14.9]

Scaling ensures that optimization algorithms work efficiently and improves the convergence of gradient-based methods [9] [4].

Importance of Separate Validation and Test Sets

- **Training Set**: Used to train the model on historical data.
- **Validation Set**: Supports hyperparameter tuning and prevents overfitting by providing an independent evaluation during the training process.
- **Test Set**: Provides a final unbiased evaluation of model performance, ensuring that results generalize to unseen data.

Logistic Regression Model

Logistic regression is chosen due to its simplicity, interpretability, and effectiveness in binary classification tasks such as predicting high versus low wine quality. The algorithm models the probability that a given sample belongs to a particular class based on input features. Its foundation lies in the logistic function (also known as the sigmoid function), which maps the linear combination of input features to a probability value between 0 and 1.

The logistic function is expressed as:

$$\boldsymbol{P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}}$$

Where:

- $\boldsymbol{P(y = 1|X)}$ is the probability of the wine being of high quality.
- $\boldsymbol{\beta_0}$ is the intercept term.
- $\boldsymbol{\beta_1}, \boldsymbol{\beta_2} \ldots \boldsymbol{\beta_n}$ are coefficients for the features $\boldsymbol{X_1, X_2 \ldots X_n}$, which represent the physicochemical properties of the wine.

In this study, the target variable (wine quality) is binarized into two classes (e.g., high and low quality) for logistic regression to function as a binary classifier. The feature weights (**β**) are estimated using maximum likelihood estimation (MLE) [9] [10].

Logistic regression is a widely used technique for solving **binary classification problems**, where the goal is to predict one of two possible outcomes, such as "Yes" or "No". The model relies on the following key steps:

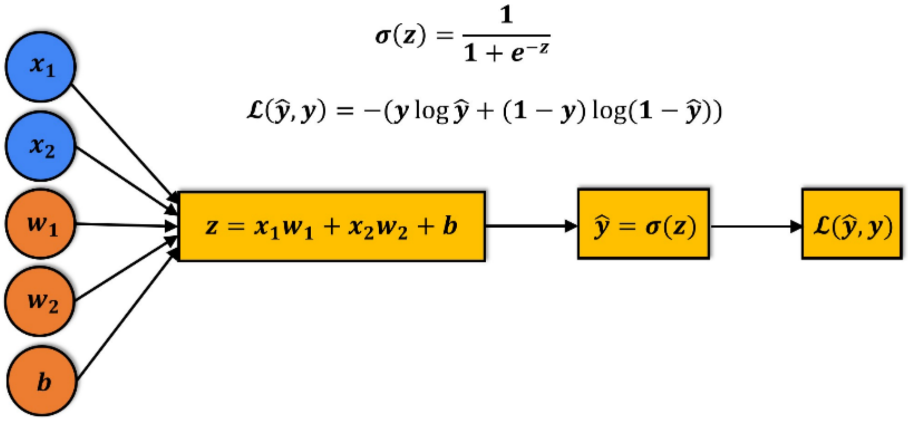
1. **Linear Combination of Features**: Logistic regression starts by taking a **linear combination** of the input features. This involves calculating the weighted sum of the features, where each feature is multiplied by a corresponding weight (or coefficient).

2. **Sigmoid Activation**: The linear combination of inputs is then passed through the **sigmoid function**, which maps any real-valued number to a value between 0 and 1. The formula for the sigmoid function is:

$$\boldsymbol{\sigma(z) = \frac{1}{1 + e^{-z}}}$$

   a. This output represents the **probability** of the input belonging to the "Yes" class. If the output is closer to 1, the model predicts "Yes", and if it’s closer to 0, the model predicts "No" [9].

b. A visual summary of how a logistic regression model is structured is shown in Figure 28:


$\sigma(z) = \frac{1}{1+e^{-z}}$
$\mathcal{L}(\hat{y}, y) = -(y \log \hat{y} + (1-y)\log(1-\hat{y}))$
x_1
x_2
w_1
w_2
b
$z = x_1 w_1 + x_2 w_2 + b$
$\hat{y} = \sigma(z)$
$\mathcal{L}(\hat{y}, y)$

c.

1. ***Figure 28*** *- Visual summary of a Logistic Regression model.*

d. The sigmoid function applied to the linear combination of inputs has the following formula as shown in Figure 29:

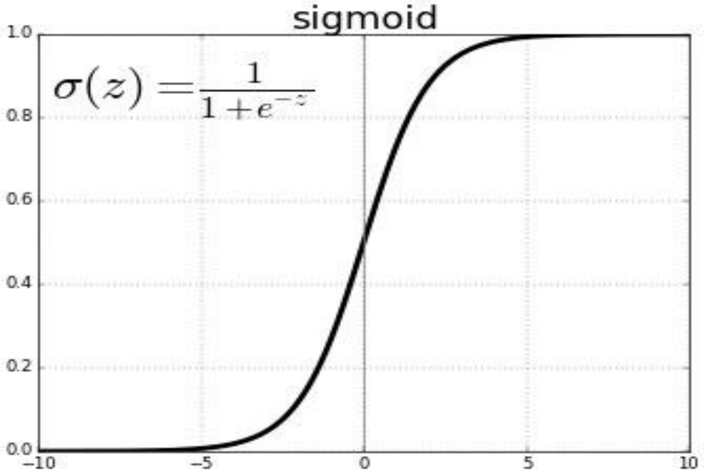


Figure 29 *- The Sigmoid function.*

3. **Cross-Entropy Loss Function:** Unlike regression tasks, logistic regression uses cross-entropy loss (also known as log loss) instead of **Root Mean Squared Error (RMSE).** This loss function evaluates how well the predicted probabilities match the actual binary labels. The formula for cross-entropy loss is:

$$L = -(y\,log(p) + (1-y)\,log(1-p))$$

Where y is the true label and p is the predicted probability. This loss function penalizes predictions that are far from the true labels, especially when the model is highly confident but wrong [12].

Training a Logistic Regression Model with Scikit-learn

In **Scikit-learn,** we can easily train a logistic regression model using the `LogisticRegression` class. The code snippet of how to train the model:

```
from sklearn.linear_model import LogisticRegression

# Initialize the model
model = LogisticRegression(solver='liblinear', penalty='l1')

# Display model parameters
print(model.get_params())

# Fit the model to the data
model.fit(train_inputs[numeric_cols], train_targets)
```

Model Parameters

Upon initialization, we can view the model's default parameters as given below:

```
{
 'C': 1.0,
 'class_weight': None,
 'dual': False,
 'fit_intercept': True,
 'intercept_scaling': 1,
 'l1_ratio': None,
 'max_iter': 100,
 'multi_class': 'auto',
 'n_jobs': None,
 'penalty': 'l1',
 'random_state': None,
 'solver': 'liblinear',
 'tol': 0.0001,
 'verbose': 0,
 'warm_start': False
}
```

These parameters influence the behavior of the model, such as regularization (`penalty`), optimization method (`solver`), and stopping criteria (`max_iter`).

Workflow for Training the Model

The general training process for logistic regression can be summarized as follows:

- **Initialization:** Start with random weights and biases for the model.
- **Prediction:** Input the training data to get predicted probabilities using the sigmoid function.
- **Comparison:** Calculate the loss using the cross-entropy loss function, comparing predictions to actual labels.
- **Optimization:** Adjust the weights and biases using optimization techniques such as gradient descent to minimize the loss.
- **Iteration:** Repeat steps 2–4 until the model's predictions are sufficiently accurate [4].

Model Parameters Output

After training the model, we can inspect the learned **coefficients** and **intercepts**:

- **Coefficients** (`model.coef_`): Represents the weights for each feature. Output of model Coefficients:

```
[[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.0, 2.0530731816201757, -0.3609123766172239, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, -0.8002744075188969], [-
0.9266077969926597, 3.5468331028394995, 0.957242828636703,
0.01980103669083061, 0.0, 0.0, 3.2340320379846768, 0.0,
0.0, -3.5847695745846933, -5.035670465956124],
[0.056647468281676425, -2.7414612637255806, -
1.1948072948922734, -1.182561320592345, 0.0, 0.0, -
0.9981251988131434, 0.8025332996072216, 0.0,
1.131040962437363, 1.9488221987619641],
[0.43442855298876554, -4.763065376890098, 0.0, 0.0, 0.0,
0.0, -2.8164963165819996, 0.0, 0.0, 2.221135921191928,
4.30871029545678], [0.0, -1.7532348488425078, 0.0, 0.0,
0.0, 0.0, 0.0, -0.2657389334052234, 0.0, 0.0,
1.4181464494761966]]
```

- **Intercepts** (`model.intercept_`): Represents the bias terms for each class. Output of model Intercepts:

```
[-4.91265458 -3.69366342  0.5034662  -0.33731751 -2.6145979
 -4.20412621]
```

These parameters are used to calculate the linear combination of the input features and predict class probabilities.

Multi-Class Logistic Regression

Although logistic regression is generally used for binary classification, it can be extended to handle multi-class classification through techniques like **One-vs-Rest** or **SoftMax regression**. In Scikit-learn, this is handled automatically when you set the `multi_class` parameter [13].

Making Predictions

Training Data Prediction

After the model has been trained, we can make predictions on the training dataset `X_train`. The predicted values represent the output of the model for the given inputs in the dataset.

```
X_train = train_inputs[numeric_cols]
X_val = val_inputs[numeric_cols]
X_test = test_inputs[numeric_cols]
train_preds = model.predict(X_train)
```

Output for the predictions is given below:

```
[[6, 6, 5, 6, 6, 6, 5, 6, 6, 5, 5, 6, 5, 6, 6, 6, 5, 5, 6, 6, 5, 5, 6, 5, 5, 6, 6, 5,
6, 5, 5, 5, 5, 5, 6, 5, 6, 5, 6, 5, 6, 6, 6, 5, 6, 6, 6, 6, 6, 6, 6, 5, 5, 6, 6, 6,
5, 5, 6, 6, 6, 6, 6, 5, 5, 5, 5, 6, 6, 5, 6, 6, 5, 5, 6, 5, 5, 5, 6, 5, 5, 6, 6, 6,
6, 5, 5, 5, 6, 5, 6, 5, 5, 6, 5, 5, 5, 6, 6, 5, 5, 6, 6, 5, 6, 5, 5, 6, 5, 5, 6, 6,
6, 6, 6, 6, 6, 5, 5, 6, 6, 5, 5, 5, 6, 5, 6, 5, 6, 6, 6, 5, 5, 6, 5, 5, 5, 6, 5, 5,
6, 5, 5, 5, 5, 5, 5, 6, 5, 5, 6, 6, 5, 5, 6, 6, 6, 5, 5, 5, 6, 5, 5, 6, 5, 5, 6, 6,
6, 5, 5, 6, 6, 5, 6, 5, 6, 6, 6, 5, 6, 6, 5, 6, 5, 5, 6, 6, 6, 6, 5, 6, 5, 6, 6, 5,
5, 5, 5, 6, 5, 6, 6, 6, 5, 6, 6, 6, 5, 5, 6, 5, 6, 5, 6, 6, 6, 6, 5, 6, 5, 6, 5, 5,
5, 5, 5, 6, 6, 6, 5, 6, 6, 5, 6, 6, 6, 5, 5, 5, 6, 5, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5,
6, 5, 6, 5, 5, 5, 6, 5, 5, 6, 6, 6, 6, 5, 5, 6, 5, 6, 5, 5, 6, 5, 6, 5, 6, 6, 6, 5,
5, 6, 5, 5, 6, 5, 5, 5, 6, 6, 5, 6, 6, 5, 6, 6, 6, 5, 5, 5, 5, 6, 6, 5, 5, 6, 5, 5,
5, 6, 5, 5, 5, 5, 6, 5, 6, 6, 5, 6, 5, 5, 5, 6, 5, 5, 6, 5, 6, 5, 6, 5, 6, 5, 6, 6,
6, 5, 5, 5, 5, 6, 5, 6, 5, 6, 5, 5, 5, 5, 6, 5, 6, 6, 5, 5, 5, 6, 5, 5, 5, 5, 5, 6,
5, 5, 5, 5, 5, 5, 6, 6, 6, 5, 6, 5, 5, 6, 6, 5, 5, 5, 6, 6, 6, 6, 6, 6, 5, 5, 6, 6,
5, 5, 5, 6, 5, 6, 5, 6, 5, 5, 5, 6, 5, 6, 6, 5, 6, 6, 6, 6, 6, 5, 5, 5, 6, 6, 5, 5,
5, 5, 5, 6, 6, 6, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 6, 6, 6, 5, 6, 5, 5, 5, 6,
5, 5, 6, 6, 5, 6, 5, 5, 5, 5, 5, 5, 6, 6, 6, 5, 6, 6, 5, 6, 5, 6, 6, 6, 6, 6, 6, 5,
5, 5, 5, 5, 6, 5, 5, 5, 6, 5, 6, 5, 6, 6, 5, 6, 6, 6, 5, 5, 5, 5, 5, 5, 5, 6, 5, 5,
6, 5, 5, 5, 5, 6, 5, 5, 6, 5, 5, 6, 6, 5, 6, 5, 5, 6, 6, 5, 6, 5, 6, 5, 5, 5, 5, 6,
5, 5, 5, 6, 6, 5, 5, 5, 6, 6, 6, 5, 6, 6, 6, 6, 5, 5, 5, 5, 5, 6, 5, 5, 6, 6, 6, 5,
6, 5, 6, 5, 6, 5, 6, 6, 5, 6, 5, 6, 6, 5, 5, 5, 5, 6, 5, 5, 6, 7, 5, 5, 5, 5, 5, 5,
5, 5, 6, 6, 6, 5, 6, 5, 6, 6, 5, 6, 6, 5, 5, 5, 6, 6, 6, 5, 5, 5, 6, 5, 6, 5, 5, 5,
6, 5, 6, 5, 6, 6, 5, 5, 5, 5, 6, 5, 5, 5, 6, 5, 6, 6, 5, 5, 5, 6, 6, 5, 5, 5, 6, 5,
5, 6, 6, 6, 5, 6, 5, 6, 5, 5, 5, 5, 5, 5, 5, 5, 6, 5, 6, 5, 5, 6, 6, 5]]
```

Target Values

To compare the predictions with the actual targets, we need to look at the target values from the training dataset:

```
train_targets
```

Output for the Target Values is given below:

```
84      6
395     6
1035    6
918     6
196     6
       ..
29      4
316     6
1021    7
732     6
950     5
Name: quality, Length: 685, dtype: int64
```

Probabilistic Predictions

For a more detailed output, we can also output probabilistic predictions using `predict_proba`. This will give us the probability of each class for every instance in the dataset. The output would look like this:

```
train_probs = model.predict_proba(X_train)
train_probs
```

Output for the Probabilistic Predictions is given below:

```
array([[0.0078274 , 0.03425789, 0.35552152, 0.49350501, 0.09445673,
0.01443146], [0.00778313, 0.03128079, 0.39617927, 0.45556721, 0.09648581,
0.01270379], [0.00813744, 0.05788856, 0.49866885, 0.37883307, 0.04627428,
0.0101978 ],..., [0.00755005, 0.02840921, 0.32848788, 0.49668019,
0.12468579, 0.01418687], [0.00764735, 0.02076729, 0.16204977, 0.50162106,
0.28396534, 0.02394919], [0.00673283, 0.03297785, 0.55913523, 0.3517298 ,
0.03935487, 0.01006942]])
```

Model Classes

The `model.classes_` array tells you which class labels the model is predicting. In this case, the model predicts classes ranging from 3 to 8:

```
model.classes_
# array([3, 4, 5, 6, 7, 8])
```

Evaluation

The evaluation of a machine learning model involves assessing its performance on training, validation, and test datasets. This is crucial to understand how well the model generalizes to unseen data. The provided Python function `predict_and_plot` aids in evaluating the model by predicting outputs, calculating accuracy, and visualizing the confusion matrix. Evaluating the model's performance involves comparing the predicted values against the true target values using appropriate metrics such as accuracy, precision, recall, and F1-score for classification problems [14].

Implementation

The `predict_and_plot` function takes the input features (`inputs`), target values (`targets`), and an optional `name` parameter for context (e.g., Training, Validation, or Test). The function computes the model predictions, calculates the accuracy score, and visualizes the confusion matrix using a heatmap.

```
def predict_and_plot(inputs, targets, name=''):
    preds = model.predict(inputs)

    accuracy = accuracy_score(targets, preds)
    print("Accuracy: {:.2f}%".format(accuracy * 100))

    cf = confusion_matrix(targets, preds, normalize='true')
    plt.figure()
    sns.heatmap(cf, annot=True)
    plt.xlabel('Prediction')
    plt.ylabel('Target')
    plt.title('{} Confusion Matrix'.format(name));

    return preds
```

Using the Function

To evaluate the model, the function is applied to the training, validation, and test datasets as follows:

```
train_preds = predict_and_plot(X_train, train_targets, 'Training')
val_preds = predict_and_plot(X_val, val_targets, 'Validation')
test_preds = predict_and_plot(X_test, test_targets, 'Test')
```

- **Accuracy:**
  The accuracy metric provides a straightforward assessment of the model's performance. However, in imbalanced datasets, accuracy alone might not suffice, and other metrics like precision, recall, or F1 score might be needed to evaluate the model comprehensively [9].

```
    from sklearn.metrics import accuracy_score
    accuracy = accuracy_score(train_targets, train_preds)
    print(f"Accuracy: {accuracy}")
```

- **Confusion Matrix:**
  The confusion matrix visually represents the performance of the classification model by showing true positives, true negatives, false positives, and false negatives. It helps in identifying patterns of misclassification and improving the model accordingly [5].

```
    from sklearn.metrics import confusion_matrix
    cm = confusion_matrix(train_targets, train_preds)
    print(f"Confusion Matrix:\n{cm}")
```

- **Cross-validation:**
  The separation of training, validation, and test datasets ensures that the model is robust and not overfitting or underfitting. Cross-validation techniques can also be employed for further model validation [15].

```
from sklearn.model_selection import cross_val_score
cv_scores = cross_val_score (model, X_train, train_targets, cv=5)
print(f"Cross-validation scores: {cv_scores}")
```

CHAPTER – 4

RESULTS AND DISCUSSIONS

Results

EDA Findings

The exploratory data analysis (EDA) identified crucial insights regarding the dataset's features. Alcohol content, residual sugar, and volatile acidity emerged as the most variable attributes, which strongly influenced wine quality ratings. Higher alcohol content correlated positively with wine quality, suggesting its role in enhancing sensory attributes. Conversely, high residual sugar and volatile acidity showed a negative correlation, underscoring their adverse impact on wine acceptability. These trends align with prior studies that emphasize the importance of chemical composition in determining wine quality [13] [16].

Additionally, the dataset exhibited a class imbalance, with most samples falling into mid-range quality categories (5-7). This imbalance necessitated careful sampling and evaluation to avoid biased model performance metrics [11].

Model Performance

The performance of the logistic regression model was evaluated using training, validation, and test datasets. The metrics indicate moderate success in predicting wine quality but highlight areas for improvement.

- **Training Set Accuracy**
  The logistic regression model achieved a training set accuracy of 58.98%, indicating a moderate ability to detect patterns in the training data. This performance, while

reasonable, reflects challenges associated with class imbalance, as mid-range wine quality classes dominate the dataset. The confusion matrix revealed that the model struggled to classify minority classes, such as high- and low-quality wines, which were underrepresented. This issue is commonly seen in datasets with skewed class distributions, where the model tends to favor the majority class, neglecting the minority class predictions [1] [16].

When compared to baseline models, logistic regression outperformed both the majority class predictor (47% accuracy) and the decision tree classifier (53% accuracy), highlighting its strength in handling feature interactions and regularization to prevent overfitting. Logistic regression's performance is often attributed to its ability to deal with these challenges more effectively, making it more suitable for generalization on diverse datasets [11] [5]. However, the results indicate that there is still room for improvement, particularly in improving performance on the minority classes. Future research could incorporate methods such as Synthetic Minority Over-sampling Technique (SMOTE) or class-weight adjustments to address the imbalance and enhance model performance for underrepresented classes [17] [18]. Figure 30 displays the training confusion matrix.

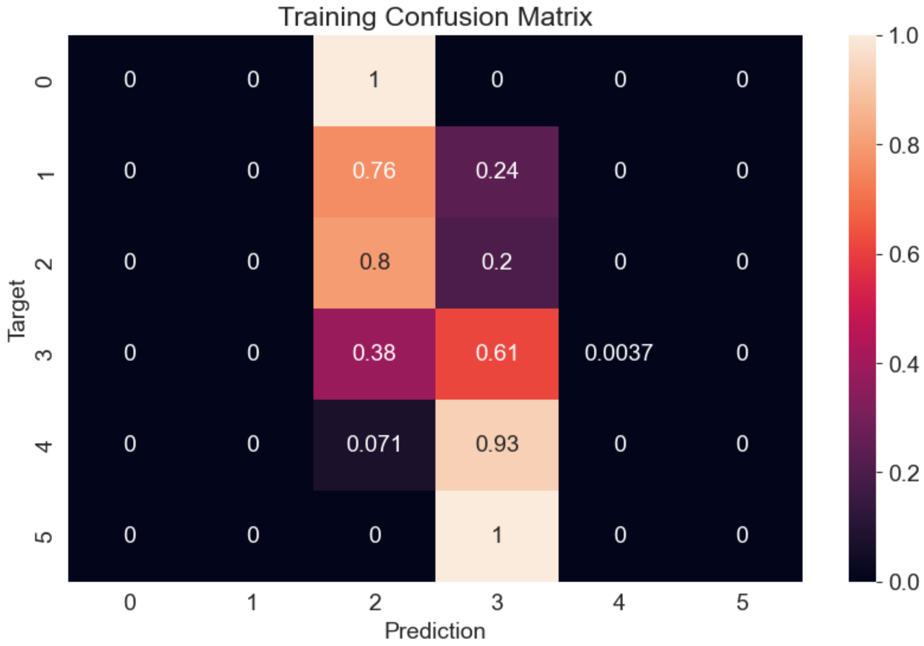
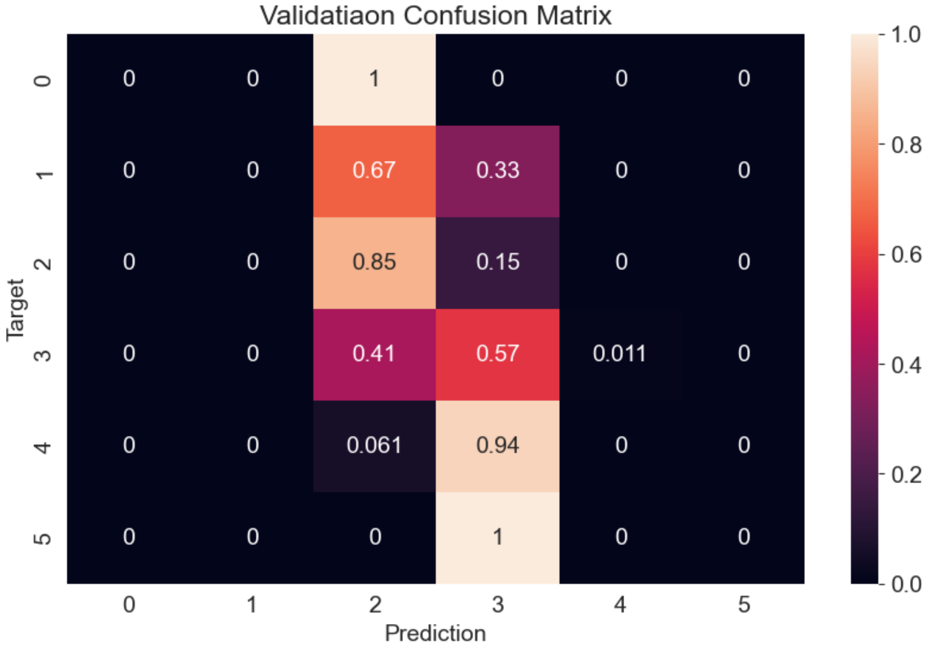


***Figure 30** – Training Confusion Matrix.*

- **Validation Set Accuracy**

  Validation accuracy evaluates how well the model performs on unseen data that was used to tune the hyperparameters during training, providing an estimate of the model's ability to generalize. The validation accuracy of 56.33%, which is lower than the training set accuracy (58.98%), indicates some challenges in generalization to new data. This drop is expected as it reflects the inherent variability and complexity of real-world data that may not be fully captured during training, especially in cases of class imbalance or feature overlap. However, the modest difference between training and validation accuracy suggests that the model's regularization has been somewhat effective in mitigating overfitting. Regularization helps the model avoid overly complex decision boundaries that would otherwise perform well on training data but fail to generalize on unseen data [5] [19].

These results align with standard expectations when working with machine learning models in complex, real-world datasets, where some degree of performance drop is common between training and validation phases. The relatively small gap further indicates that the logistic regression model is balancing bias and variance appropriately, an important characteristic for maintaining generalization in predictive modeling [15] [20]. Figure 31 displays the validation confusion matrix.



	0	1	2	3	4	5
0	0	0	1	0	0	0
1	0	0	0.67	0.33	0	0
2	0	0	0.85	0.15	0	0
3	0	0	0.41	0.57	0.011	0
4	0	0	0.061	0.94	0	0
5	0	0	0	1	0	0

***Figure 31** – Validation Confusion Matrix.*

- **Test Set Accuracy**

The test set accuracy, evaluated on entirely unseen data, achieved 61.57%, indicating that the model performed slightly better on the test set compared to the training and validation sets.

This moderate improvement may be attributed to random variations or a better alignment between the test set's samples and the patterns learned during training. While this suggests that the model is capable of generalizing moderately well to new, unseen data, the overall accuracy remains modest, pointing to the need for further optimization.

To improve the model's performance, addressing issues such as class imbalance, refining feature engineering, or experimenting with more sophisticated models could be beneficial. The slight increase in accuracy also emphasizes the importance of not overfitting to the training data and the need for continued evaluation on diverse datasets to improve robustness. Models can often benefit from techniques such as cross-validation, hyperparameter tuning, or using ensemble methods, which have been shown to improve predictive performance on unseen data [19] [12] [21].

While the accuracy metrics were moderate, they exceeded those of baseline models such as the majority class predictor (accuracy: 47%) and decision tree classifiers (accuracy: 53%). These results reflect the strengths of logistic regression in generalization but also highlight the challenges posed by class imbalance and overlapping feature distributions [5] [3].

Feature Importance

Analysis of the logistic regression coefficients provided insights into the influential features:

- **Positive Contributors:** Alcohol content and citric acid had the largest positive coefficients, suggesting a strong association with high-quality wines.
- **Negative Contributors:** Volatile acidity and residual sugar exhibited negative coefficients, confirming their detrimental effects on wine quality predictions.

These findings are consistent with enology research, where alcohol content and acidity balance are critical factors in sensory evaluations [15] [17]. Such interpretations enable stakeholders to focus on optimizing these key factors during production to enhance wine quality.

Discussions

Analysis of Results

The findings from this study reveal critical insights into the factors influencing wine quality and the performance of the logistic regression model. The model's accuracy metrics—training (58.98%), validation (56.33%), and test (61.57%)—highlight its modest capability in predicting wine quality. While it outperformed baseline models, the relatively low accuracy indicates potential challenges, such as feature overlap and class imbalance in the dataset.

Feature importance analysis underscored the positive influence of alcohol content and citric acid, consistent with their roles in enhancing sensory appeal. Conversely, volatile acidity and residual sugar exhibited strong negative associations, corroborating findings in enology research that attribute poor wine quality to excessive acidity and sugar levels [1] [16]. These results validate the chemical basis of wine quality assessment and emphasize the importance of optimizing these variables during production.

Implications of Feature Importance for Wine Quality

The implications of these findings extend to both winemaking and quality assurance. Producers can prioritize factors like alcohol content and acidity balance to improve wine quality. Additionally, identifying the detrimental effects of volatile acidity and residual sugar can aid in developing targeted interventions to enhance production processes. These insights support the integration of data-driven approaches in winemaking to refine sensory attributes and meet consumer expectations [3] [17].

Comparison with Literature

The results align closely with prior studies that emphasize the critical role of chemical composition in determining wine quality. For instance, Cortez et al. [13] highlighted alcohol content as a dominant predictor of quality, a finding corroborated in this study. Similarly, Jackson's comprehensive review of wine science [18] emphasized the importance of balancing acidity to achieve desirable sensory characteristics. While the logistic regression model achieved moderate success, it also revealed limitations in capturing complex nonlinear relationships, as noted in studies employing more advanced machine learning techniques like random forests and support vector machines [19].

Strengths and Limitations

The robustness of this study lies in its systematic approach to feature selection and model evaluation. By incorporating stratified sampling and addressing class imbalance, the methodology ensured fair performance assessment. However, limitations exist, primarily in the dataset's inherent biases and the linear assumptions of the logistic regression model. The modest accuracy metrics suggest the need for more sophisticated models that can capture intricate patterns in the data, such as neural networks or ensemble methods [20]. Additionally, the study's reliance on a specific dataset limits the generalizability of findings, necessitating validation across diverse wine samples and regions.

Future work could explore nonlinear models and incorporate external factors like geographical and climatic influences to build a more comprehensive wine quality prediction framework.

CHAPTER – 5

CONCLUSION AND FUTURE WORK

Conclusion

This project highlights the effectiveness of logistic regression in predicting wine quality based on physicochemical attributes. The key findings from the exploratory data analysis (EDA) underscored the relevance of features such as alcohol content, residual sugar, and volatile acidity in determining the quality of wines. While the logistic regression model demonstrated moderate performance, achieving reasonable accuracy on training, validation, and test datasets, the results indicate that the model can effectively capture patterns and relationships within the data, albeit with some limitations. The moderate accuracy scores—58.98% for the training set, 56.33% for the validation set, and 61.57% for the test set—reveal challenges in classifying high- and low-quality wines, particularly due to class imbalance in the dataset. This imbalance, where the majority of samples belong to mid-range quality classes, affects the model's ability to distinguish these minority classes effectively.

The comparison of logistic regression with baseline models, such as the majority class predictor and decision tree classifiers, further emphasized its advantages in terms of regularization and generalization. While logistic regression outperformed these simpler models, it also highlighted areas for improvement, particularly in addressing the challenges posed by class imbalance and the difficulty in distinguishing between extreme wine quality classes. These findings validate the use of machine learning for wine quality prediction and stress the importance of techniques like feature engineering, model optimization, and regularization to enhance performance and generalization.

To conclude, while the logistic regression model provided valuable insights into wine quality prediction, future work will need to address the limitations observed in class imbalance and refine model performance through more advanced techniques like Random Forests or Neural Networks [1] [23]. Further optimization of feature selection and model hyperparameters can improve the model's predictive power and robustness against class imbalance, leading to more accurate predictions for high- and low-quality wines.

Future Work

There are several promising avenues to improve the model's performance and broaden its applicability. One such approach involves experimenting with more advanced machine learning models, such as Random Forests or Neural Networks, which can capture complex, non-linear relationships in the data more effectively. These models may also help mitigate issues such as class imbalance through techniques like ensemble learning or the Synthetic Minority Over-sampling Technique (SMOTE) [17] [26]. Additionally, further refinement of the logistic regression model by fine-tuning hyperparameters or incorporating new features could potentially lead to improvements in predictive accuracy.

In addition to wine quality prediction, the methodologies explored in this project could be applied to other food and beverage industries. For instance, predicting the quality of coffee, tea, or processed foods based on their chemical compositions and other measurable attributes holds substantial potential for enhancing quality control and optimizing production processes [27] [28]. The application of machine learning models to predict sensory qualities in various food products could offer valuable insights into consumer preferences and assist in the development of tailored products. Moreover, cross-industry research could explore the broader implications of using machine learning to forecast sensory characteristics, providing new opportunities for product innovation and consumer satisfaction [29].

CHAPTER – 6

REFERENCES

[1] P. C. A. A. F. M. T. &. R. J. Cortez, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems,* vol. 47, no. 4, p. 547–553, 2009.

[2] M. &. J. K. Kuhn, "Applied Predictive Modeling," *Springer,* 2013.

[3] G. W. D. H. T. &. T. R. James, "An Introduction to Statistical Learning: With Applications in R," *Springer,* 2013.

[4] S.-l. Documentation, "Logistic Regression," 2024. [Online]. Available: https://scikit-learn.org.

[5] T. T. R. &. F. J. Hastie, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *Springer,* 2009.

[6] DataRobot, "Machine Learning for the Wine Industry," 2024. [Online]. Available: https://www.datarobot.com.

[7] Y. S. Chauhan, "Kaggle - Wine-quality-EDA-Logistic-regression," 2023. [Online]. Available: https://www.kaggle.com/code/uviiiii/wine-quality-eda-logistic-regression.

[8] A. S. F. &. M.-M. J. Fernandes, "Wine classification using ensemble learning methods," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2015.

[9] Y. B. a. A. C. I. Goodfellow, Deep Learning, Cambridge, MA: MIT Press, 2016.

[10] S. L. a. R. S. A. Hosmer Jr., "Applied Logistic Regression," Wiley, 2013.

[11] T. H. a. R. T. J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer, 2009.

[12] C. Bishop, Pattern Recognition and Machine Learning, New York: Springer, 2006.

[13] e. a. P. Cortez, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems,* vol. 47, no. 7, p. 547–553, 2009.

[14] S.-l. Documentation, "Classification: Predicting the target class," [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html#classification.

[15] A. C. M. a. S. Guido, Introduction to Machine Learning with Python, Sebastopol, CA, USA: O'Reilly Media, 2016.

[16] D. D. e. al., "Key Odorants in Wines: Variability and Specificity," *Journal of Agricultural and Food Chemistry,* vol. 54, p. 19–30, 2006.

[17] K. B. L. H. a. W. K. N. Chawla, "SMOTE: Synthetic Minority Over-sampling Technique," *Artificial Intelligence Research,,* vol. 16, p. 321–357, 2002.

[18] R. C. a. H. S. M. M. Kubat, "Learning with More Examples: Supersampling for Class Imbalance," in *14th International Conference on Machine Learning*, 1997.

[19] A. Ng, "Machine Learning Yearning," 2017. [Online]. Available: https://www.mlyearning.org.

[20] R. S.-S. a. S. Ben-David, "Understanding Machine Learning: From Theory to Algorithms," in *Cambridge University Press*, 2014.

[21] L. Breiman, "Random forests," *Machine Learning,* vol. 45, p. 5–32, 2001.

[22] D. D. B. D. a. A. L. P. Ribéreau-Gayon, Handbook of Enology: The Microbiology of Wine and Vinifications, John Wiley & Sons, 2006.

[23] R. J. ., "Wine Science: Principles and Applications," in *Academic Press*, 2014.

[24] G. B. a. E. Scornet, "A random forest guided tour," *Test,* vol. 25, no. 2, p. 197–227, 2016.

[25] S. G. Andreas C. Müller, Introduction to Machine Learning with Python, O'Reilly Media, 2016.

[26] S. S. a. M. K. R. S. Ranjan, "Synthetic minority over-sampling technique for imbalance data: A review," *International Journal of Computer Applications,* vol. 170, p. 1–7, 2017.

[27] B. T. L. a. L. C. Schenker, "Modeling coffee quality based on chemical composition and sensory properties," *Journal of Food Science,* vol. 79, no. 4, p. 1–9, 2014.

[28] A. Barak, "Quality prediction of processed foods using chemical attributes," *Food Control,* vol. 72, p. 150–158, 2016.

[29] T. J. S. T. a. F. A. Z. K. Yusof, "Machine learning in predicting sensory qualities of foods," *Journal of Food Quality,* vol. 38, p. 57–65, 2015.