Red Wine Quality Prediction

Submitted In Partial Fulfillment of Requirements

For the Degree Of

Honours in Data Science and Analytics

(Offered by Department of Computer Engineering)

By

Sanyukta Joshi

Roll No: 16010120019

Manas Marathe

Roll No: 16010120027

Zenith Mehta

Roll No: 16010420028

Guide

Dr. Shruti Javkar

SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya Vidyavihar University
Vidyavihar, Mumbai - 400 077

2020-24

SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya Vidyavihar University

K. J. Somaiya College of Engineering

Certificate

This is to certify that the dissertation report entitled **Red Wine Quality Prediction** submitted by Sanyukta Joshi, Manas Marathe and Zenith Mehta at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfillment of requirements for the degree Honours in Data Science (**Offered by Department of Computer Engineering**) of Somaiya Vidyavihar University

____________________	________________________
Guide	Head of the Department

____

Examiner

Date:

Place: Mumbai-77

Abstract

The prediction of wine quality is of paramount importance in the wine industry for ensuring product excellence and customer satisfaction. In this study, we employed various machine learning algorithms to predict wine quality based on a comprehensive dataset comprising different wine attributes.

The performance of algorithms including K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Random Forest Classifier, AdaBoost Classifier, and XGBoost Classifier was evaluated based on testing accuracy and F1 score. Results indicated that the Random Forest Classifier and the XGBoost Classifier with hyperparameter tuning emerged as the top-performing algorithms, exhibiting strong predictive capabilities and robustness in handling complex data distributions. Support Vector Classifier also demonstrated competitive performance, underscoring its effectiveness in classifying wine quality classes in high-dimensional spaces.

Furthermore, hyperparameter tuning using Optuna significantly improved the performance of the XGBoost Classifier, highlighting the importance of optimizing hyperparameters for superior predictive performance.

The findings from this study provide valuable insights for wine producers and researchers in leveraging machine learning techniques for wine quality prediction, with implications for enhancing production processes and product quality in the wine industry.

Keywords: Random Forest Classifier, XGBoost Classifier, Support Vector Classifier, Hyperparameter tuning, Optuna

Contents

List of Figures………………………………………………………………… i

List of Tables……………………………………………………………….. ii

1 Introduction…………………………………………………………..... 1

1.1 Background/Motivation 1

1.2 Problem Statement 1

1.3 Scope 2

1.4 Objectives

1.5 Hardware and software requirements for development

1.6 Hardware and software requirements for deployment

1.7 Other (depending on project)

1.8 Dataset Dictionary (If Applicable)

2 Literature Survey…………………………………………………… 4

3 Project design ………………………………………………………… 8

3.1 Proposed System model/ Architecture…………………………….

3.2 Software Project Management Plan……………………………….

3.3 Software Design Document (All applicable diagrams)...................

4 Implementation and Experimentation….………..…………………… 24

4.1 Proposed system model implementation 24

4.2 Inclusion of Any additional details as suggested by Project Guide/during progress seminar 25

4.3 Software Testing (Software testing reports at various levels) 27

4.4 Experimental results and its analysis 31

5 Conclusions and scope for further work…….……………………… 46

5.1 Conclusions and discussion 46

5.2 Scope for future work 47

Bibliography ………………………………………………………………… 49

Acknowledgements……………………………………………………….. 51

List of Figures

1.1 Sample data from dataset……………………………………………………….

3.1 System Flow Diagram………………………………………………………..

3.2 Data Analysis Workflow………………………………………………………..

4.1 Countplot for different quality levels…………………………………………

4.2 Boxplot for Fixed Acidity……………………………………………………

4.3 Boxplot for Volatile Acidity………………………………………………….

4.4 Boxplot for Citric Acid………………………………………………………

4.5 Boxplot for Residual Sugar…………………………………………………..

4.6 Boxplot for Chlorides………………………………………………………...

4.7 Boxplot for Free Sulfur Dioxide……………………………………………..

4.8 Boxplot for Total Sulfur Dioxide……………………………………………

4.9 Boxplot for Density…………………………………………………………..

4.10 Boxplot for Ph………………………………………………………………..

4.11 Boxplot for Sulphates………………………………………………………...

4.12 Correlation Matrix……………………………………………………………

4.13 Results for K-Nearest………………………………………………………...

4.14 Results for Support Vector Classifier………………………………………...

4.15 Results for Random Forest…………………………………………………...

4.16 Results for AdaBoost Classifier……………………………………………...

4.17 Fig 4.17 Results for K-Nearest………………………………………………

4.18 Result for Optuna Hyperparameter Tuning…………………………………..

List of Tables

1.1 Dataset Details…………………………………………………………………..

2.1 Literature Survey……………………………………………………………..

4.1 XGBoost Hyperparameters…………………………………………………...

4.2 Experiment Results…………………………………………………………...

Chapter 1

Introduction

This chapter presents the foundation for our exploration into red wine quality prediction. It begins by highlighting the wine industry's immense global importance, emphasizing its pivotal role in delivering high-quality products to consumers worldwide. Delving into the problem statement, we confront the challenges inherent in predicting red wine quality. These challenges arise from the multifaceted nature of wine evaluation, which encompasses a diverse array of chemical and sensory attributes. Our objective is to develop sophisticated models capable of accurately assessing quality based on these complex variables, addressing issues such as non-linearity and multicollinearity along the way. The chapter also delineates the hardware and software requirements essential for both development and deployment phases of our research. Lastly, the dataset dictionary provides a comprehensive overview of the attributes included in our dataset, setting the stage for subsequent analyses and model development.

1.1 Background

The wine industry stands as a global economic powerhouse, encompassing millions of producers and consumers worldwide. With an annual production exceeding billions of liters, the wine industry commands a formidable presence in both domestic and international markets.

Amidst this vast landscape, ensuring wine quality remains paramount. Quality not only influences consumer preferences but also shapes the industry's reputation and competitiveness. As such, winemakers constantly seek innovative approaches to enhance quality assurance and product excellence.

Advancements in data science and machine learning have ushered in a new era of wine quality assessment, offering vintners and researchers powerful tools to decipher the myriad factors that influence wine characteristics. By analyzing vast datasets encompassing a multitude of chemical and sensory attributes, it becomes possible to discern patterns, correlations, and predictive models that illuminate the essence of wine quality.

1.2 Problem Statement

Predicting the quality of red wine presents a significant challenge due to the diverse variables and subjective nature of sensory evaluation. The goal is to develop reliable models that can accurately assess red wine quality based on chemical composition and sensory attributes. This involves navigating complex relationships between variables and addressing challenges such as non-linearity and multicollinearity.

To tackle this problem, we aim to leverage comprehensive datasets encompassing various chemical parameters, sensory descriptors, and quality ratings. By employing machine learning techniques, we seek to develop predictive models capable of providing valuable insights to winemakers, guiding decisions related to production processes and quality assurance. Ultimately, our goal is to empower the wine industry with data-driven tools that enhance product excellence and consumer satisfaction.

1.3 Scope

This study delves into the realm of red wine quality prediction, aiming to develop robust predictive models leveraging machine learning techniques. Our scope encompasses the analysis of comprehensive datasets comprising diverse chemical parameters, sensory descriptors, and quality ratings associated with red wine samples.

Through the application of advanced machine learning algorithms, we seek to unravel the intricate relationships between these variables and accurately predict red wine quality. Specifically, our focus extends to exploring the predictive capabilities of algorithms such as K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Random Forest Classifier, AdaBoost Classifier, and XGBoost Classifier.

1.4 Objectives

- Investigate a variety of predictive models for assessing the quality of red wine, encompassing classification algorithms.
- Conduct thorough performance evaluations utilizing established metrics.
- Employ advanced optimization techniques like hyperparameter tuning with Optuna to refine model performance iteratively, aiming for optimal configurations that enhance predictive accuracy and generalization capabilities.

1.5 Hardware and Software Requirements for Development

- Running this project requires sufficient processing power and memory to handle data analysis and machine learning tasks efficiently.
- Python programming language (version 3.0 or above) for coding machine learning algorithms and data analysis tasks.
- Integrated Development Environment (IDE) such as Google Colab
- Necessary Python libraries and packages for data manipulation (Pandas, NumPy), visualization (Matplotlib, Seaborn), and machine learning (e.g., Scikit-learn, XGBoost, Optuna)

1.6 Hardware and software requirements for Deployment

Deployment requires a server or cloud computing platform capable of hosting the machine learning models. Containerization may be considered for streamlined deployment and management, while GPU support may be beneficial for tasks requiring intensive computation, such as hyperparameter tuning using Optuna.

1.7 Dataset Dictionary

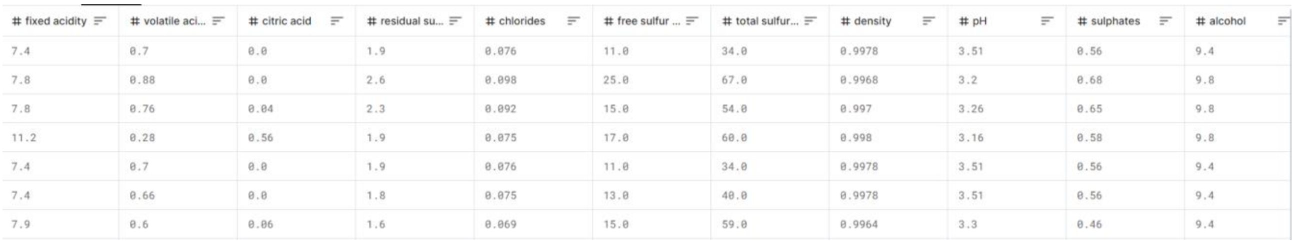
Dataset title: Red Wine Quality [1599 rows, 12 columns]

Dataset link: https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

- **Fixed Acidity:** The concentration of fixed acids, such as tartaric acid, in the wine.
- **Volatile Acidity:** The concentration of volatile acids, which can contribute to off-flavors and wine spoilage.
- **Citric Acid:** The concentration of citric acid, which can add acidity and freshness to the wine.
- **Residual Sugar:** The amount of sugar remaining in the wine after fermentation, influencing its sweetness.
- **Chlorides:** The concentration of chlorides, which can affect the wine's taste and mouthfeel.
- **Free Sulfur Dioxide:** The amount of sulfur dioxide present in its free form, which acts as a preservative.
- **Total Sulfur Dioxide:** The total amount of sulfur dioxide, including both free and bound forms.

- **Density:** The density of the wine, which is influenced by alcohol content and sugar concentration.
- **pH:** The acidity or alkalinity of the wine, affecting its taste and stability.
- **Sulphates:** The concentration of sulfates, which can contribute to wine aroma and stability.
- **Alcohol:** The alcohol content of the wine, influencing its body and perceived warmth.
- **Quality:** The overall quality rating of the wine, typically assessed through sensory evaluation.

Term	**Need**	**Value Range**
Fixed Acidity	Provides backbone of flavor, acidity balance	4 - 15 g/L
Volatile Acidity	Indicates spoilage	< 0.8 g/L
Citric Acid	Adds freshness, balances flavors	Up to 0.5 g/L
Residual Sugar	Determines sweetness level	0 - 220 g/L
Chlorides	Contributes to taste	< 0.1 g/L
Free Sulfur Dioxide	Preserves wine	10 - 40 mg/L
Total Sulfur Dioxide	Preserves wine	Up to 150 mg/L
Density	Indicates body and texture	0.98 - 1.05 g/mL
pH	Affects taste and stability	3.0 - 4.0
Sulphates	Acts as preservative	0.3 - 2.0 g/L
Alcohol	Influences body and richness	9% - 16% by volume
Quality	Overall evaluation of the wine	Typically rated on a scale of 3 to 8 or 0 to 10



# fixed acidity	# volatile aci...	# citric acid	# residual su...	# chlorides	# free sulfur ...	# total sulfur...	# density	# pH	# sulphates	# alcohol
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
7.8	0.88	0.0	2.6	0.098	25.0	67.0	0.9968	3.2	0.68	9.8
7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.997	3.26	0.65	9.8
11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.998	3.16	0.58	9.8
7.4	0.7	0.0	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
7.4	0.66	0.0	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4
7.9	0.6	0.06	1.6	0.069	15.0	59.0	0.9964	3.3	0.46	9.4

Fig 1.1 Sample data from dataset "Red Wine Quality"

Chapter 2

Literature Survey

This chapter presents the recent research endeavors concerning the prediction of red wine and beverage quality using machine learning techniques. Five distinct studies are scrutinized, each presenting various methodologies and approaches to tackle the challenge of quality prediction. From comparing different machine learning models like Naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest Classifier in the analysis of red wine quality, to exploring the effectiveness of techniques such as hyperparameter optimization and imbalance data handling in enhancing prediction accuracy, the literature survey provides valuable insights into the evolving landscape of predictive analytics in the beverage industry.

Sr.No	Title	Objective	Model/Algorithm	Dataset	Conclusion
1.	Quality of Red Wine: Analysis and Comparative Study of Machine Learning Models [1]	To develop and optimize a predictive model using machine learning algorithms, focusing on decision tree classification, for accurately classifying the quality of red wine based on chemical parameters.	Naive Bayes, Logistic Regression, Support Vector Machine and Random Forest Classifier	Consists of 12 attributes and was divided into two classes based on wine quality (>6 considered good).	Random Forest Classifier demonstrated superior performance with 100% training accuracy and 98.44% model accuracy, outperforming other classifiers.
2.	Red Wine Quality	Aims to predict the quality of	Random Forest, Support Vector	The dataset	SVM showing higher accuracy

	Prediction Using Machine Learning Techniques [2]	red wine using various attributes, employing various techniques. By comparing results between training and testing sets, the most effective technique will be determined,	Machine, and Naïve Bayes.	contains 1599 instances with 12 variables for red wine data. Qualities are in the range 3-8, where '3' predicts poor quality of red wine and '8' predicts excellent quality of red wine.	(67.25% for training, 68.64% for testing) compared to Random Forest (65.83% for training, 65.46% for testing) and Naïve Bayes (55.91% for training, 55.89% for testing)
3.	Enhancing red wine quality prediction through Machine Learning approaches with Hyperparam	Compare classification and regression methods for predicting red wine quality, evaluating the performance of various classifiers and	Logistic regression, Gradient boosting, Extra Tree Classifier, AdaBoost classifier, Decision Tree, Support Vector	Red wine samples with their chemical compositions and qualitative evaluations have been	Employing machine learning algorithms like Random Forest, AdaBoost, and Gradient Boosting, among others, demonstrated high accuracy in predicting red wine

	eters optimization technique [3]	regressors with hyperparameter tuning and imbalance data handling techniques		included in the UCI Machine-Learning database Wine Grade Collection.	quality. Importance of EDA.
4.	Utilization of Random Forest Classifier (RFC) To Predict the Quality of Beverages [4]	Aims to predict beverage quality by analyzing its attributes using the Random Forest algorithm, with performance measures compared between training and testing sets to determine overall quality prediction accuracy.	Random Forest Classifier	Data obtained from the UCI ML repository. The dataset comprises 1599 observations and 8 variables pertaining to beverage data	Using Random forest algorithm, enabled precise prediction of wine quality, achieving an impressive accuracy rate of 87% alongside an F1 score of 0.93
5.	Smart Agriculture and Digital Transformation on Case	Propose an intelligent system for wine quality prediction using	Decision Tree Based ML Methodology	UCI Machine learning Repository (2009.)	Potential for digital transformation in smart agriculture by automating wine quality assessment.

	of Intelligent System for Wine Quality Prediction	decision tree-based machine learning methodology, leveraging emerging technologies		Wine Quality Data Set	The effectiveness of classification and regression trees has improved a great deal to help in prediction

Chapter 3

Project Design

This chapter explains how we planned and carried out the project to make models that predict how good red wine is. We start by explaining the overall plan for our system, including how different parts work together. Then, we talk about how we managed the project, including who was involved, when things were done, and how we dealt with potential problems. After that, we describe the detailed design of the software we used, including diagrams to help understand how everything fits together. By carefully organizing and explaining these aspects, we set the stage for creating effective models that can help improve the quality of red wine.

3.1 Proposed System Model/Architecture

System Overview:

The proposed system aims to develop robust predictive models for assessing red wine quality by leveraging machine learning techniques. The architecture comprises several components, including data preprocessing, model development, evaluation, and deployment.

Components:

- **Data Collection:** The system collects data from various sources such as databases, APIs, or files. In this case, the red wine dataset is obtained from a CSV file.
- **Preprocessing:** The collected data undergoes preprocessing steps including handling missing values, solving class imbalance, encoding categorical variables, and scaling numerical features to prepare it for model training.
- **Exploratory Data Analysis (EDA):** Exploring the dataset through visualizations and statistical analysis to gain insights into the distribution, relationships, and patterns of the data.
- **Model Selection:** Choosing suitable machine learning algorithms based on the nature of the problem and the characteristics of the dataset. Algorithms such as K-Nearest Neighbors, Support Vector Classifier, Random Forest Classifier, AdaBoost Classifier, and XGBoost Classifier are considered in this project.

- **Model Training:** Training the selected machine learning models using the preprocessed dataset to learn patterns and relationships between input features and target labels.
- **Model Evaluation:** Assessing the performance of trained models using metrics such as accuracy, F1 score, confusion matrix, and ROC curves. This helps in comparing the effectiveness of different algorithms and selecting the best-performing model.
- **Hyperparameter Tuning:** Optimizing the hyperparameters of selected models to further improve their performance. Techniques like Optuna are used for hyperparameter tuning in this project, specifically for the XGBoost Classifier.
- **Deployment:** Deploying the trained and optimized model into a production environment where it can be used for real-time prediction of wine quality based on input data.

3.2 Software Project Management Plan

Requirements Gathering:

We started by understanding the project requirements, focusing on developing predictive models for red wine quality assessment.

We gathered relevant data sources and identified the necessary features for model development.

Design:

Once we had a clear understanding of the requirements, we proceeded to design the data preprocessing steps, ensuring they covered data cleaning, feature engineering, and normalization.

We planned the structure of our predictive models, selecting appropriate algorithms like regression or classification to meet the quality assessment criteria.

Implementation:

With the design finalized, we implemented the data preprocessing steps, addressing issues such as missing values and feature transformations.

We then developed the predictive models using the chosen algorithms, meticulously writing and testing the code to train and optimize the models for accuracy.

Testing:

After completing the implementation phase, we rigorously tested the models to ensure they met our performance expectations.

We evaluated the models using established metrics to validate their effectiveness in assessing red wine quality accurately.

Communication Plan:

Regular meetings to discuss progress, challenges, and adjustments.

Documentation of decisions, updates, and issues for transparency and reference.

3.2 Software Design Document

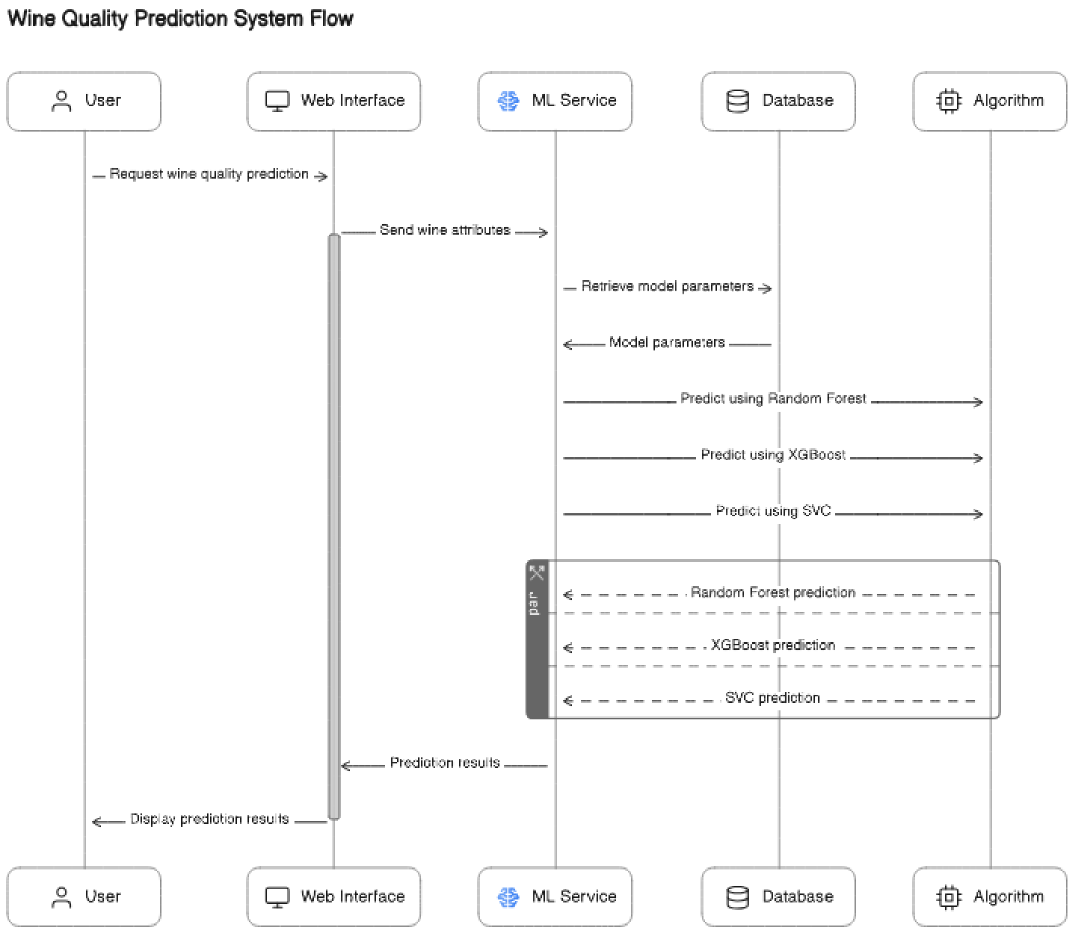


Fig 3.1 System Flow Diagram

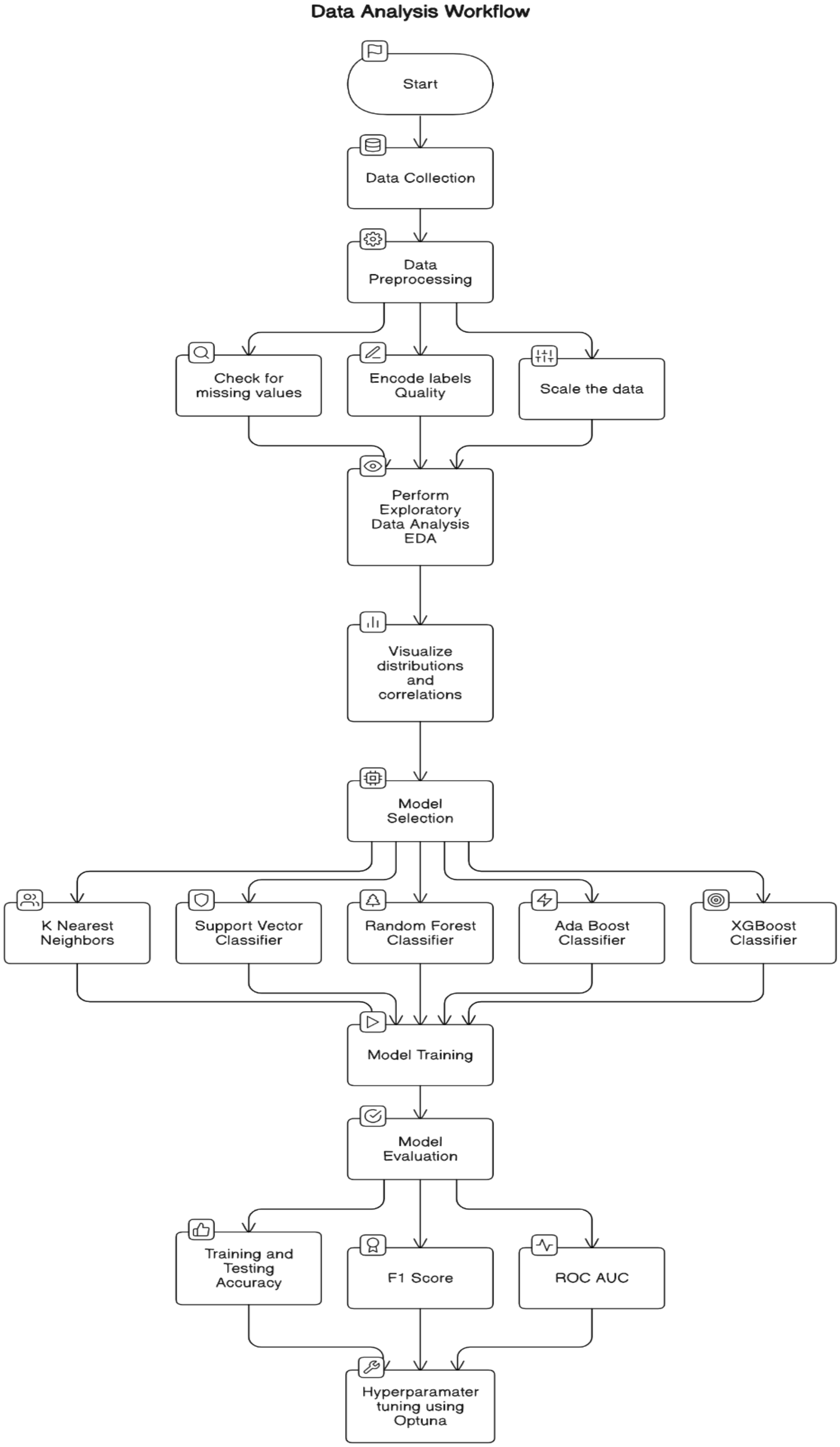


Fig 3.2 Data Analysis Workflow

Chapter 4

Implementation and Experimentation

This chapter presents the implementation of the proposed system model and its evaluation. The chapter begins with the importation of necessary libraries and the reading of the dataset. Exploratory Data Analysis (EDA) techniques are employed to gain insights into the dataset's characteristics and distributions. Preprocessing steps, including handling missing values, addressing class imbalance, and encoding labels, are executed to prepare the data for modeling. Following data preprocessing, the dataset is split into training and testing sets. Several machine learning algorithms, including K Nearest Neighbors, Support Vector Classifier, Random Forest Classifier, AdaBoost Classifier, and XGBoost Classifier, are trained on the data. Each algorithm's performance is evaluated using metrics such as accuracy, F1 score, and visualizations like confusion matrices and ROC curves. Furthermore, hyperparameter tuning using Optuna is applied to enhance the performance of the XGBoost Classifier. The chapter concludes with a summary of experimental results and analysis, providing insights into the strengths and weaknesses of each algorithm and the effectiveness of various techniques employed in the model implementation process.

4.1 Proposed system model implementation

Libraries used:

- **NumPy (np):** A fundamental library for numerical computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- **Pandas (pd):** A powerful data manipulation library in Python, offering data structures like DataFrame and Series to efficiently handle and analyze structured data.
- **Matplotlib (plt):** A comprehensive plotting library in Python, used for creating static, interactive, and animated visualizations in various formats.
- **Seaborn (sns):** Built on top of Matplotlib, Seaborn is a statistical data visualization library that provides a high-level interface for drawing attractive and informative statistical graphics.
- **Scikit-learn:** A versatile machine learning library in Python, offering simple and efficient tools for data mining and data analysis, including various algorithms for classification, regression, clustering, and dimensionality reduction.
- **XGBoost:** Another popular gradient boosting library known for its speed and performance, widely used in machine learning competitions and industry applications.

- **Optuna:** An automated hyperparameter optimization framework, which simplifies the process of tuning parameters for machine learning models.

Preprocessing - Finding the number of missing values

- Preprocessing involves preparing data for analysis or modeling by transforming, cleaning, and organizing it.
- Handling missing values is a crucial step in preprocessing, where strategies such as imputation or removal are employed to address incomplete data points.
- We counted the number of missing values in each column to identify them. The dataset in use did not contain any.

Exploratory data analysis

Exploratory Data Analysis (EDA) is the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables.

The conducted Exploratory Data Analysis (EDA) on the red wine quality dataset encompasses three primary visualizations:

1. A **countplot** provides an overview of wine quality distribution, indicating that wines with a quality rating of 5 are most prevalent, while those with a rating of 3 are notably scarce
2. Series of **boxplots** depict the distributions of various wine features, such as fixed acidity, volatile acidity, and pH, offering insights into their central tendencies and variability.
3. **Correlation matrix** elucidates the interrelationships between these features, revealing potential dependencies and guiding feature selection for subsequent modeling efforts.

Countplot:

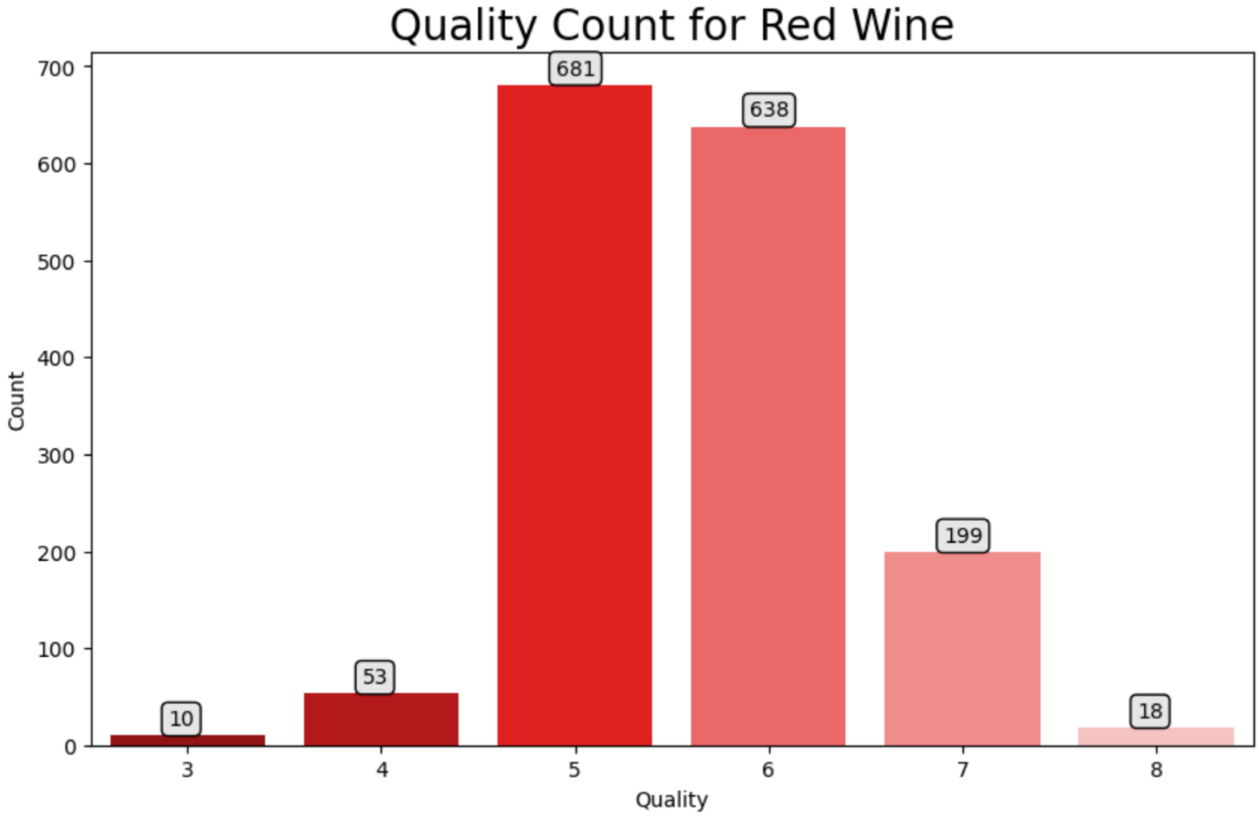


Fig 4.1 Countplot for different quality levels

- It is observed that wines with a quality rating of 5 exhibit the highest frequency, while those with a quality rating of 3 are notably scarce, representing the lowest count. This discrepancy in frequency across quality ratings underscores the presence of a class imbalance within the dataset.
- Class imbalance refers to a situation where the distribution of classes within a dataset is disproportionate, potentially leading to biased model performance during machine learning model training.
- Hence, pre-processing steps to address class imbalances before commencing the training phase of machine learning models is imperative.

Box Plots:

Boxplots are a standardized way of displaying the distribution of data. It gives a good indication of how the values in the data are spread out for each parameter of the dataset.

Based on the boxplots generated the following observations can be made:

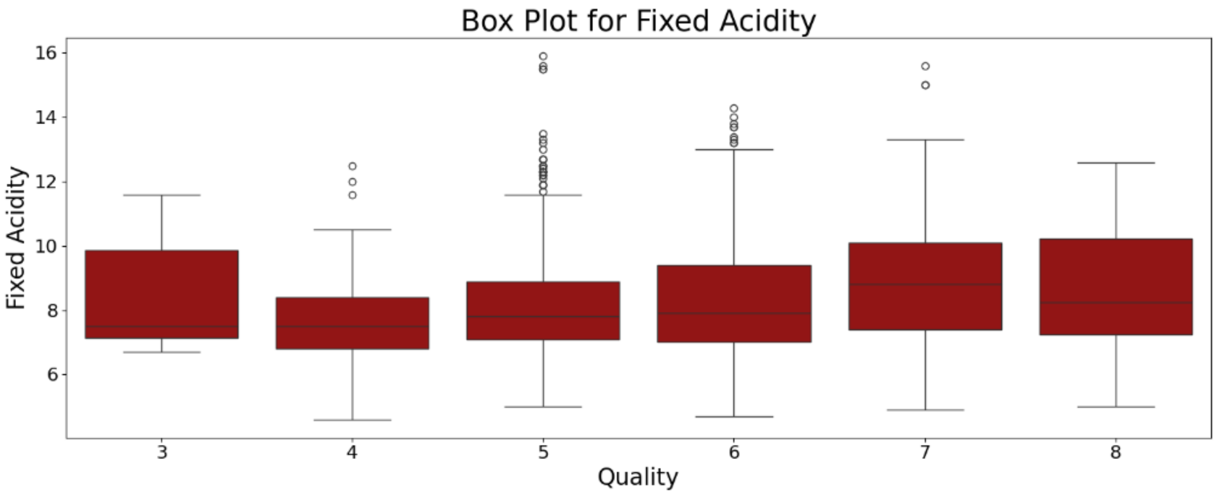


Fig 4.2 Fixed Acidity

Fixed Acidity: The median values for fixed acidity remain consistent across different qualities of wine, with the highest number of outliers observed for wines rated with a quality of 5.

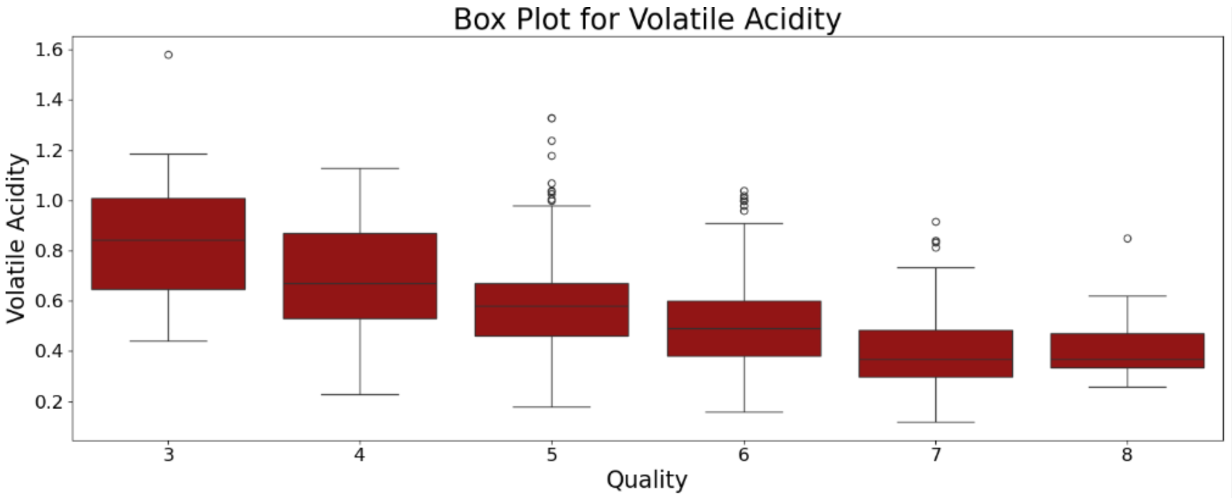


Fig 4.3 Volatile Acidity

Volatile Acidity: There is an inverse relationship between wine quality and volatile acidity, where higher quality wines exhibit lower median values of volatile acidity.

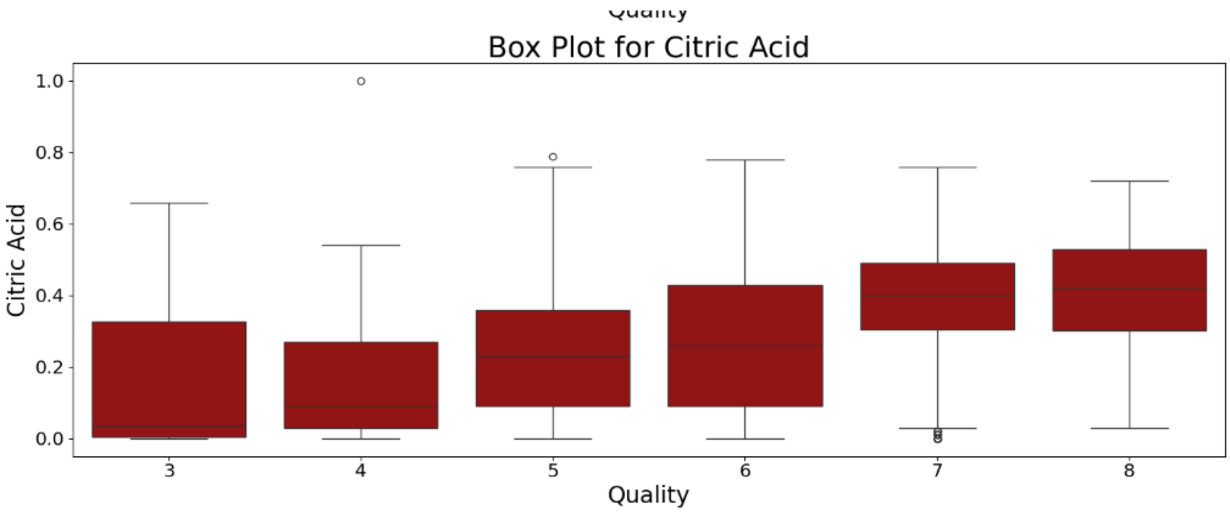


Fig 4.4 Citric Acid

Citric Acid: Contrary to volatile acidity, an increase in wine quality corresponds to higher median values of citric acid, indicating a positive correlation between quality and citric acid content.

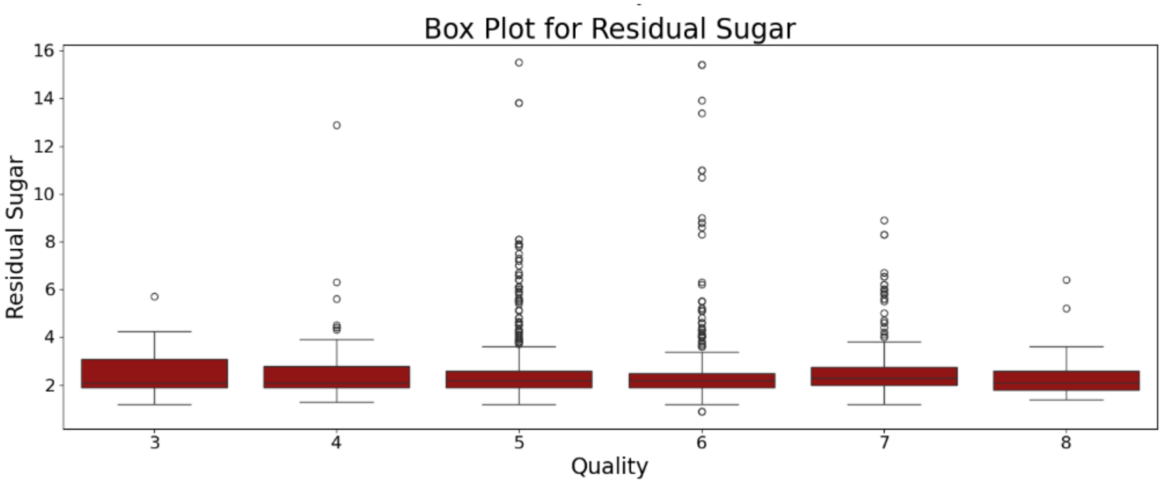


Fig 4.5 Residual Sugar

Residual Sugar: The median values for residual sugar remain relatively constant across different wine qualities, with wines rated 5 or 6 exhibiting the highest number of outliers.

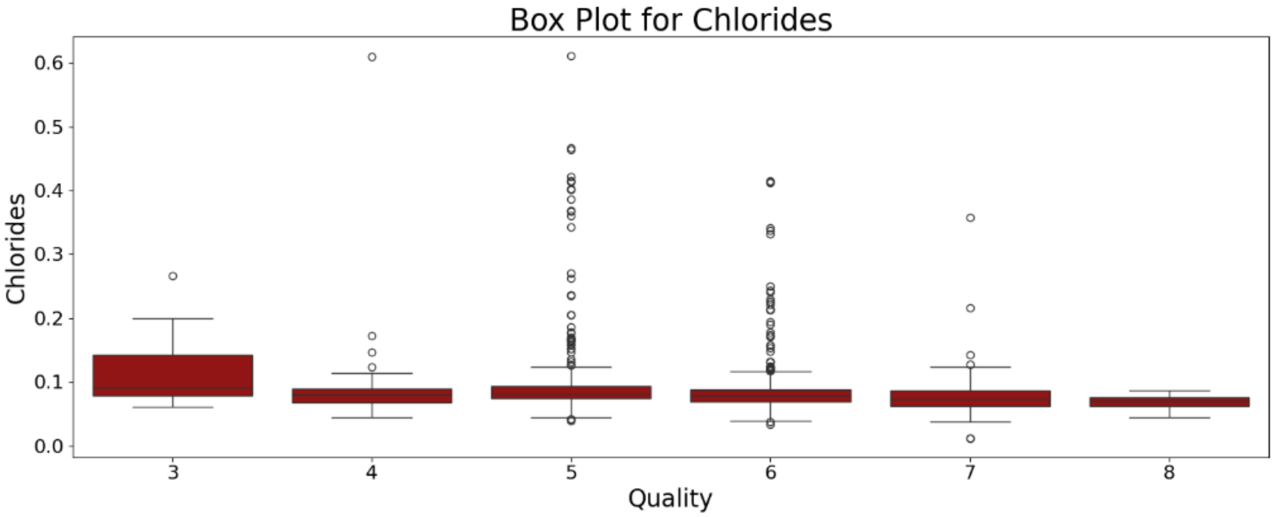


Fig 4.6 Chlorides

Chlorides: Chlorides show consistent median values across different wine qualities, suggesting minimal variation with quality.

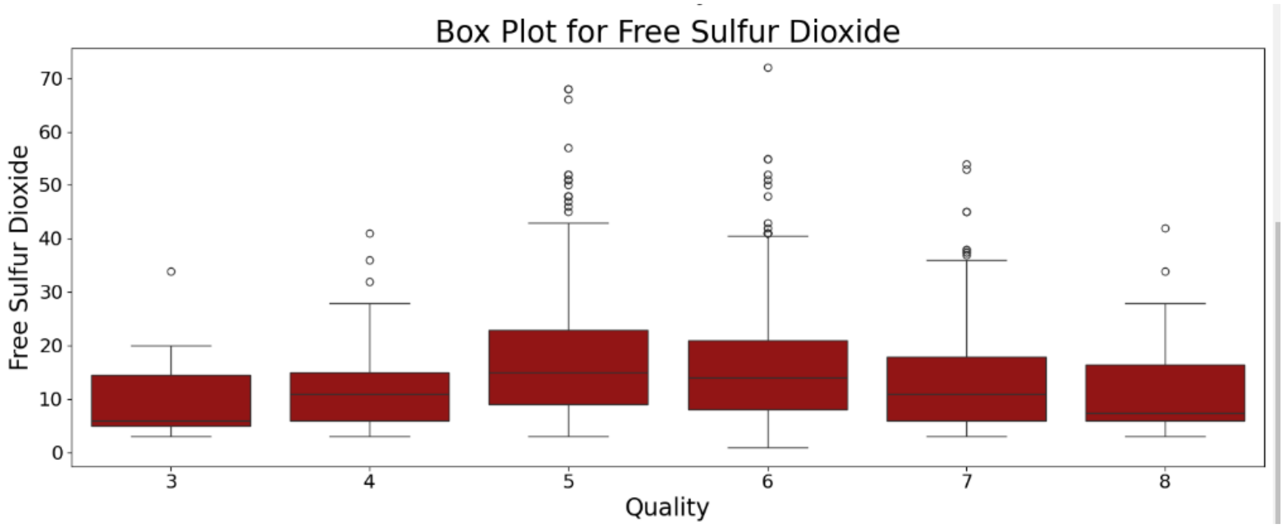


Fig 4.7 Free Sulfur Dioxide

Free Sulfur Dioxide: Wines with a quality rating of 5 demonstrate the highest median value for free sulfur dioxide.

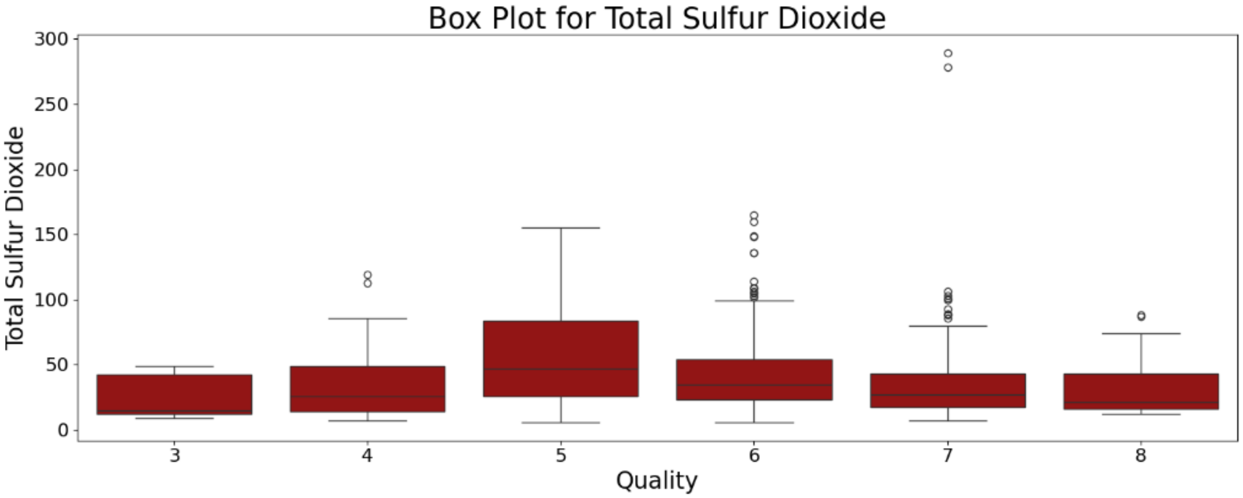


Fig 4.8 Total Sulfur Dioxide

Total Sulfur Dioxide: Wines rated 5 exhibit the widest Interquartile Range (IQR) for total sulfur dioxide.

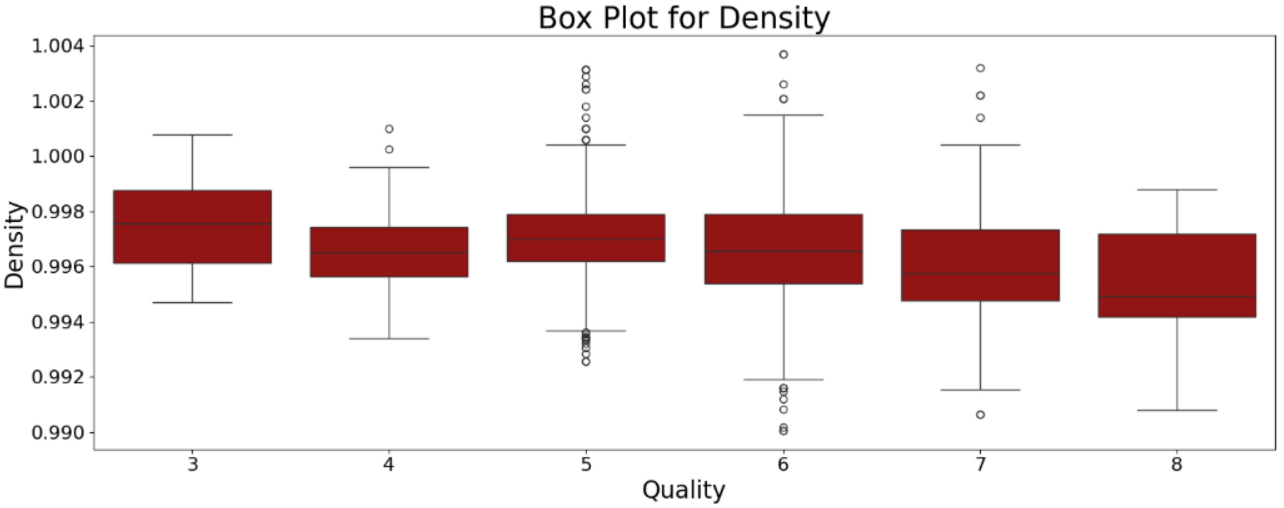


Fig 4.9 Density

Density: Wines with a quality rating of 3 have the highest median value for density.

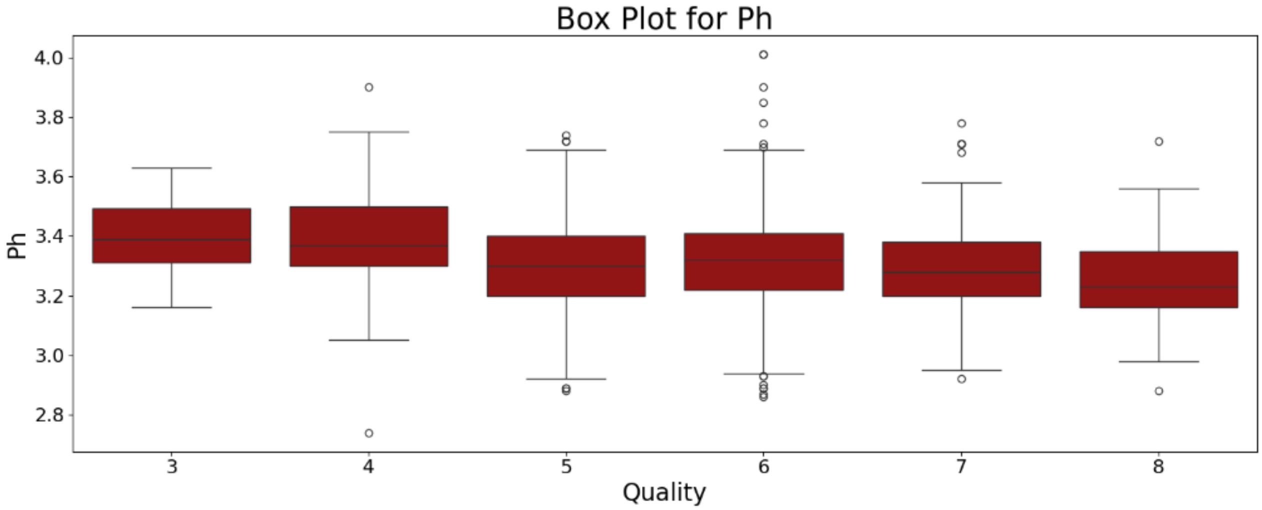


Fig 4.10 Ph

pH: Median pH values decrease with increasing wine quality. Wines rated 5 and 6 display longer tails and heads compared to others.

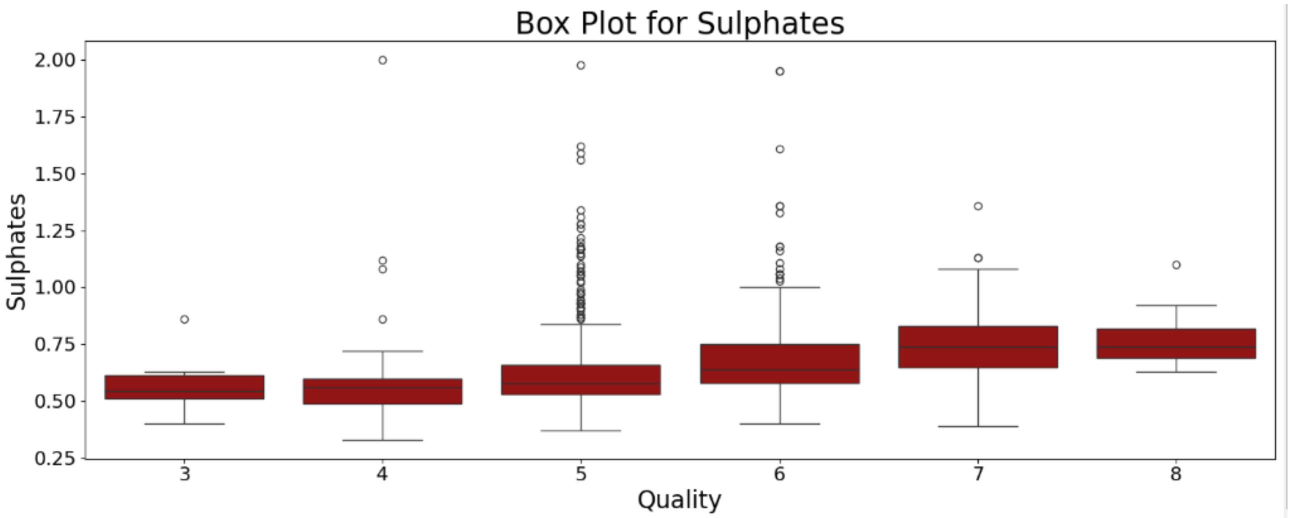
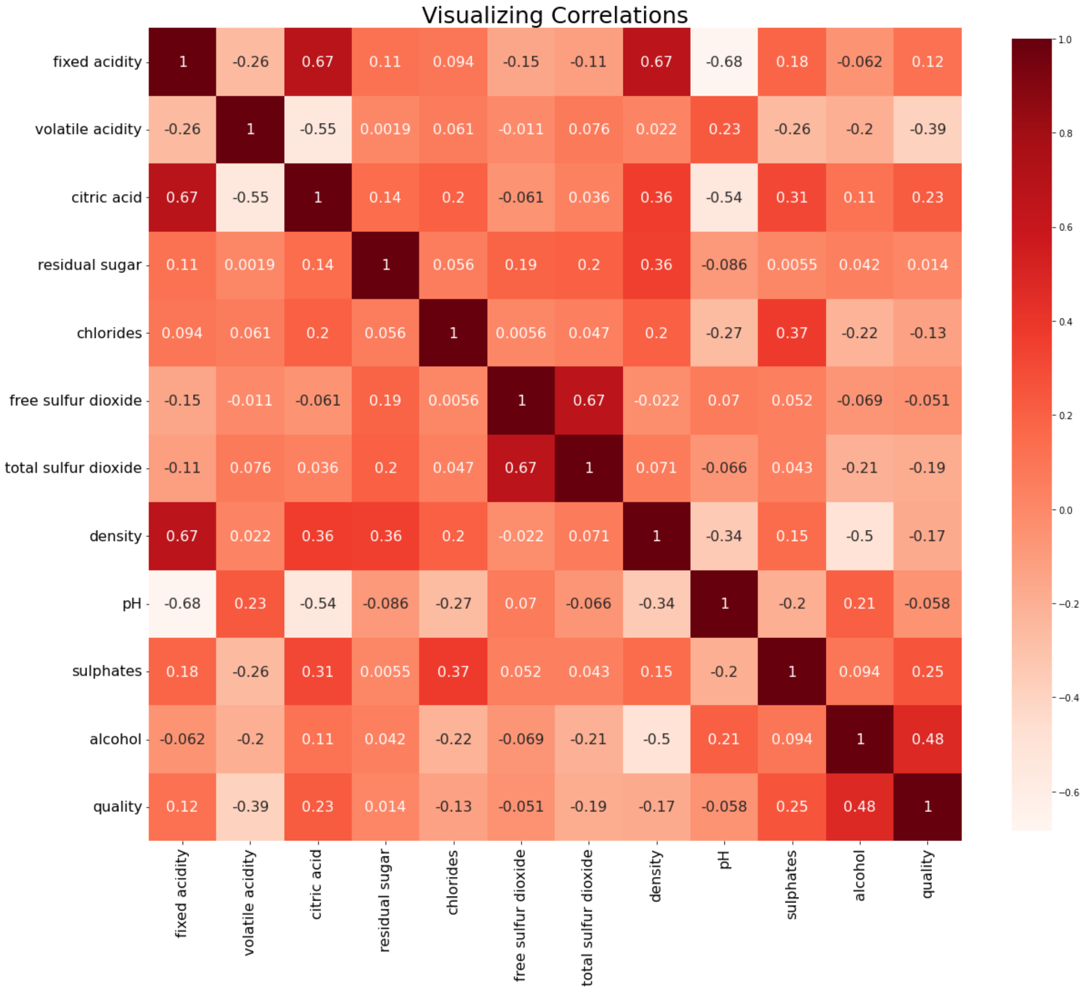


Fig 4.11 Sulphates

- **Sulphates:** Median sulphate values increase with wine quality. Similar to pH, wines rated 5 and 6 exhibit longer tails and heads.

Checking for correlation:



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.12
volatile acidity	-0.26	1	-0.55	0.0019	0.061	-0.011	0.076	0.022	0.23	-0.26	-0.2	-0.39
citric acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.23
residual sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.014
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.13
free sulfur dioxide	-0.15	-0.011	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.051
total sulfur dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.066	0.043	-0.21	-0.19
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.17
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.066	-0.34	1	-0.2	0.21	-0.058
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.25
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.48
quality	0.12	-0.39	0.23	0.014	-0.13	-0.051	-0.19	-0.17	-0.058	0.25	0.48	1

Fig 4.12 Correlation Matrix

- Correlation analysis is a fundamental technique used to examine the relationships between variables within a dataset. It quantifies the strength and direction of linear associations between pairs of features, aiding in identifying potential dependencies or patterns
- Colormaps ensures that shades of red are employed to represent correlation values, with lighter shades indicating higher values and darker shades indicating lower values.
- Colormaps, denoted by cmap, play a crucial role in visualizing data, allowing for the effective communication of patterns and trends.

Preprocessing - Solving class imbalance and encoding labels

- Class imbalance occurs when one class in a classification problem has significantly fewer samples than the others, potentially leading to biased model performance.
- We redefined the wine quality ratings into two categories, 'bad' and 'good', based on a predefined threshold. This transformation helps mitigate the imbalance by grouping similar quality ratings together.
- We then used LabelEncoder to convert the categorical quality labels into numerical values, facilitating the use of machine learning algorithms that require numerical inputs. 1 - good quality and 0 - bad quality

Preprocessing - Scaling the data

- Scaling is a crucial preprocessing step in machine learning. It ensures that all features contribute equally to the model training process by standardizing their distributions.
- It is to maintain all features on a similar scale, preventing features with larger magnitudes from dominating those with smaller magnitudes during model training.
- StandardScaler is used to transform data such that it has a mean of 0 and a standard deviation of 1, making features comparable and improving model performance.

Train - Test Split

Dataset is split into training and testing data.

X_train: Features for training the model.

X_test: Features for testing the model.

y_train: Target variable for training the model.

y_test: Target variable for testing the model.

Training the models

1. K-Nearest Neighbors (KNN)

A simple algorithm that classifies data points based on the majority class of their nearest neighbors.

- Choose the number of neighbors (K) and a distance metric.
- For each new data point:
- Measure the distance to all existing data points.
- Select the K nearest data points.
- Assign the most common label among these neighbors to the new data point.

2. Support Vector Classifier (SVC):

Constructs a hyperplane or set of hyperplanes in a high-dimensional space that can be used for classification. It's effective in high-dimensional spaces and can handle non-linear data using kernel tricks.

- Identify the best hyperplane that separates different classes in the feature space.
- Choose the hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data point of each class.
- Classify new data points based on which side of the hyperplane they fall on.

3. Random Forest Classifier:

An ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes of the individual trees. It's robust, handles overfitting well, and works for both classification and regression tasks.

- Create multiple decision trees using random subsets of the training data and features.
- Each tree independently predicts the class of a data point.
- Aggregate the predictions from all trees to determine the final prediction through voting or averaging.

4. AdaBoost Classifier:

An ensemble learning method that combines multiple weak classifiers to create a strong classifier. It focuses on improving the classification accuracy of the algorithm iteratively by adjusting the weights of incorrectly classified instances.

- Initially, assign equal weights to all training data points.
- Train a weak learner (e.g., decision tree) on the data and calculate the error.
- Increase the weights of misclassified data points and decrease the weights of correctly classified data points.
- Repeat the process with updated weights for multiple iterations.
- Combine the weak learners into a strong learner by giving more weight to the ones with lower error rates.

5. XGBoost Classifier:

An advanced implementation of gradient boosting algorithm with improvements in performance and efficiency. It builds multiple decision trees sequentially, where each tree corrects the errors of the previous one, achieving state-of-the-art results in various machine learning competitions.

- Initialize the model with a simple decision tree.
- Calculate the residuals (the difference between predicted and actual values) for each data point.
- Fit a new decision tree to predict these residuals.
- Combine predictions from all trees and update the model.
- Repeat the process with multiple iterations, optimizing a loss function to minimize errors.

<u>Hyperparameter tuning for XGBoost using Optuna</u>

The objective of the hyperparameter tuning is to maximize the accuracy of the model on the test data. We performed Hyperparameter tuning to improve the accuracy of XGBoost.

The following are the hyperparameters of XGBoost:

Hyperparameter	**Explanation**
`tree_method`	Method used to build decision trees.
`lambda`	Regularization parameter controlling L2 regularization term (ridge regularization).
`alpha`	Regularization parameter controlling L1 regularization term (lasso regularization).
`colsample_bytree`	Fraction of features to consider when building each tree.
`subsample`	Fraction of samples used for training each tree.
`learning_rate`	Rate at which model learns from the training data.
`n_estimators`	Number of trees (boosting rounds) to build.
`max_depth`	Maximum depth of a tree.
`random_state`	Seed for random number generation.
`min_child_weight`	Minimum sum of instance weight (hessian) needed in a child

4.3 Software Testing (Software testing reports at various levels)

Training Results

- **Confusion Matrix:** A confusion matrix is a table that is used to evaluate the performance of a classification model. It allows us to visualize the performance of an algorithm by comparing predicted class labels with actual class labels. The matrix consists of four main components: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These components help in calculating various performance metrics such as accuracy, precision, recall, and F1 score.
- **F1 Score:** The F1 score is a metric used to evaluate the performance of a classification model, particularly when dealing with imbalanced datasets. It is the harmonic mean of precision and recall, providing a single score that balances both measures. The F1 score ranges from 0 to 1, with higher values indicating better model performance. It is calculated as 2 * (precision * recall) / (precision + recall).
- **ROC AUC:** ROC AUC is a performance metric used to evaluate the ability of a binary classification model to discriminate between positive and negative classes across different thresholds. It measures the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (sensitivity) against the false positive rate (1 - specificity). A higher ROC AUC score (closer to 1) indicates better model performance in distinguishing between the two classes. ROC AUC is particularly useful when the dataset is imbalanced or when different threshold values need to be considered for making predictions.

K-Nearest:

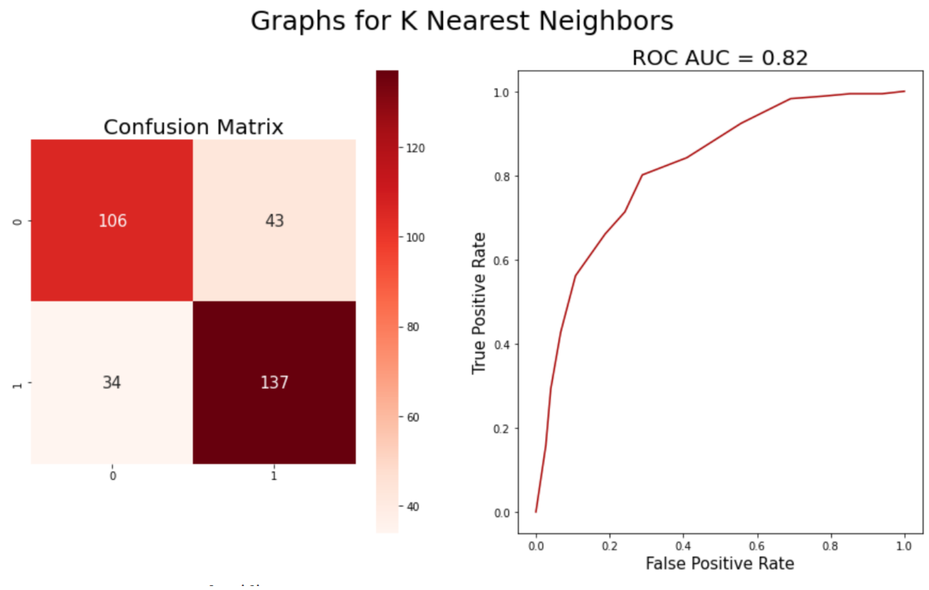


Fig 4.13 Results for K-Nearest

Support Vector Machine:

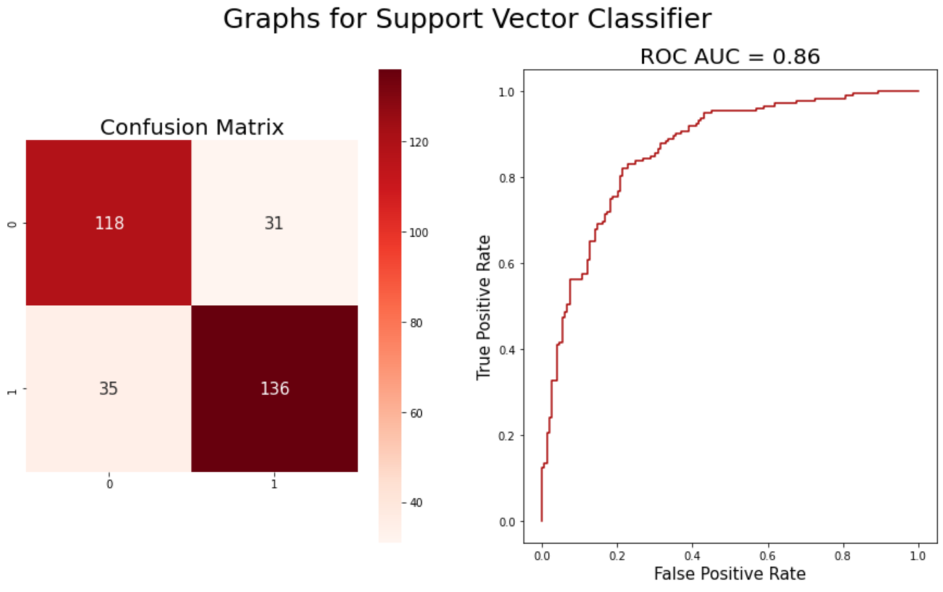


Fig 4.14 Results for Support Vector Classifier

Random Forest:

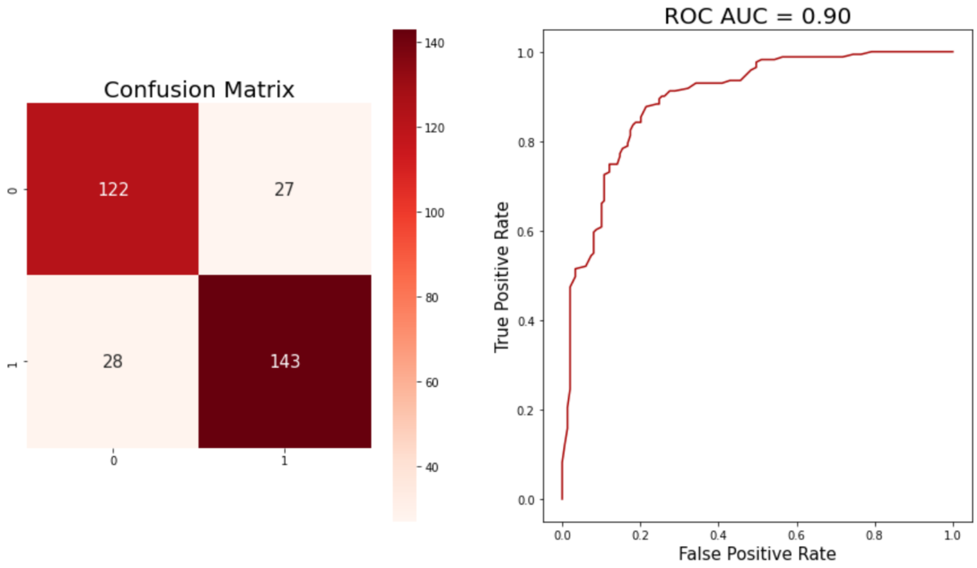


Fig 4.15 Results for Random Foresr

ADA Boost:

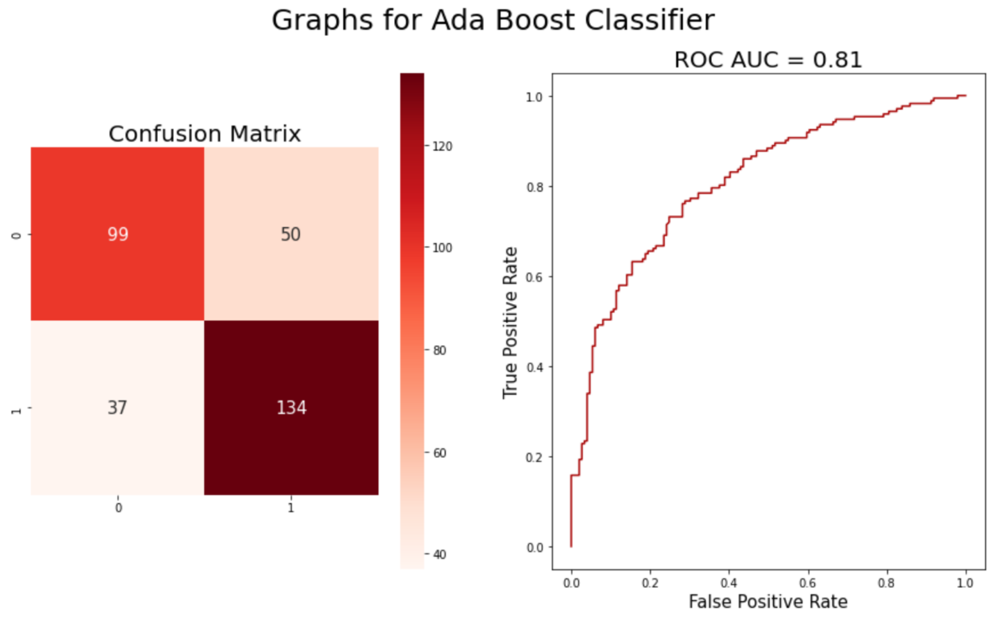


Fig 4.16 Results for AdaBoost Classifier

XGBoost Classifier:

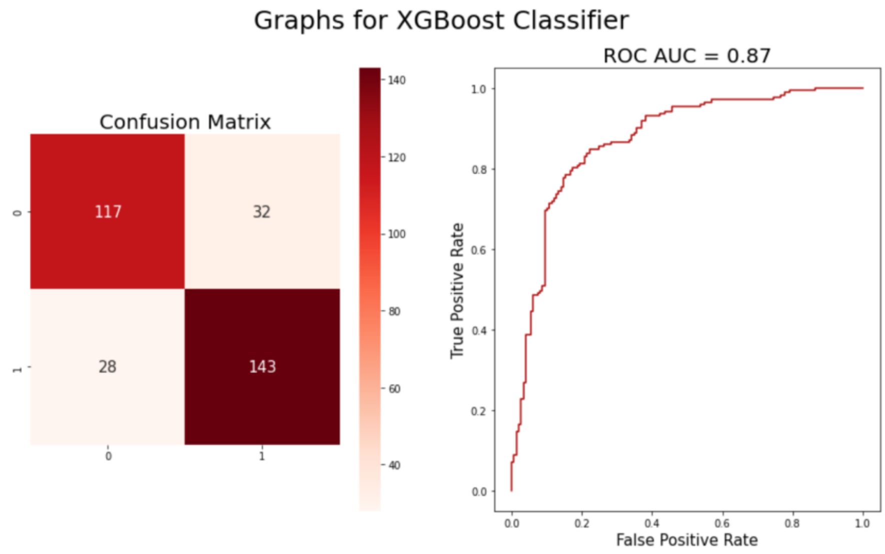
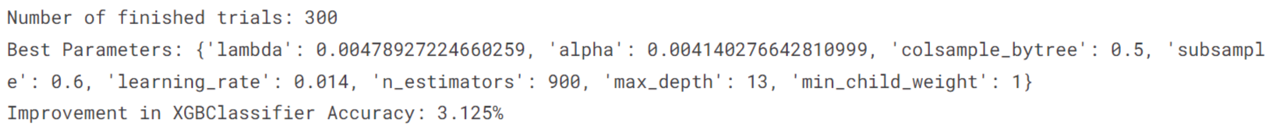


Fig 4.17 Results for K-Nearest

4.3 Experimental results and its analysis

Algorithm	Training Accuracy	Testing Accuracy	F1 Score
K-Nearest Classifier	77.80%	75.94%	0.78
Support Vector Classifier	78.65%	79.65%	0.80
Random Forest Classifier	100%	82.81%	0.84
ADABoost Classifier	78.18%	72.81%	0.75
XGBoost Classifier	100%	81.87%	0.83

Results after introducing Hyperparameter tuning using Optuna:



```
Number of finished trials: 300
Best Parameters: {'lambda': 0.00478927224660259, 'alpha': 0.004140276642810999, 'colsample_bytree': 0.5, 'subsampl
e': 0.6, 'learning_rate': 0.014, 'n_estimators': 900, 'max_depth': 13, 'min_child_weight': 1}
Improvement in XGBClassifier Accuracy: 3.125%
```

Fig 4.18 Result for Optuna Hyperparameter Tuning

Accuracy: 84.99%

Chapter 5

Conclusion and Future Work

This chapter presents a comprehensive conclusion and discussion on the performance of various machine learning algorithms in predicting wine quality. It highlights the strengths and weaknesses of each algorithm and discusses the impact of hyperparameter tuning on model performance. Additionally, it outlines future research directions and opportunities for enhancing predictive performance in wine quality prediction tasks.

5.1 Conclusion and Discussion

Performance Overview:

- K-Nearest Neighbors (KNN): While KNN demonstrated a respectable testing accuracy of 75.94% and an F1 score of 0.78, it fell slightly short compared to other algorithms. This suggests that its performance might be limited by the dataset's characteristics or the choice of hyperparameters.
- Support Vector Classifier (SVC): SVC exhibited solid performance with a testing accuracy of 79.65% and an F1 score of 0.80. It demonstrated effectiveness in separating wine quality classes, showcasing its robustness in handling complex data distributions.
- Random Forest Classifier: With a perfect training accuracy of 100% and a testing accuracy of 82.81%, the Random Forest Classifier showcased strong predictive capabilities. Its F1 score of 0.84 indicates its ability to achieve a balance between precision and recall.
- AdaBoost Classifier: AdaBoost's performance, with a testing accuracy of 72.81% and an F1 score of 0.75, was comparatively lower than other algorithms. This suggests that while AdaBoost can perform well, it may struggle with certain characteristics of the dataset or require further optimization.
- XGBoost Classifier: The XGBoost Classifier achieved a testing accuracy of 81.87% and an F1 score of 0.83, indicating robust performance. Its effectiveness in handling large datasets and complex relationships makes it a suitable choice for wine quality prediction.

Impact of Hyperparameter Tuning:

Hyperparameter tuning using Optuna significantly improved the performance of the XGBoost Classifier. The testing accuracy increased to 84.99%, highlighting the importance of optimizing hyperparameters for achieving superior predictive performance.

Conclusion:

- The Random Forest Classifier and the XGBoost Classifier with hyperparameter tuning emerged as the top-performing algorithms for wine quality prediction.
- These algorithms showcased strong predictive capabilities and demonstrated effectiveness in handling the complexities of the wine quality dataset.
- Support Vector Classifier also exhibited competitive performance, underscoring its suitability for classification tasks in high-dimensional spaces.
- While KNN and AdaBoost demonstrated respectable performance, they may benefit from further optimization or exploration of alternative algorithms.

5.2 Scope for Future Work

- Further experimentation with different algorithms, feature engineering techniques, and ensemble methods could potentially enhance predictive performance.
- Continuous monitoring and refinement of models are necessary to adapt to changing data distributions and improve generalization capabilities.
- Exploring advanced techniques that could provide additional insights and potentially improve performance further.

Bibliography

- [1] S. Kumari, A. Misra, A. Wahi and P. S. Rathore, "Quality of Red Wine: Analysis and Comparative Study of Machine Learning Models," 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2023
- [2] S. Kumar, K. Agrawal and N. Mandan, "Red Wine Quality Prediction Using Machine Learning Techniques," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104095. keywords: {processes;data extraction;Naïve Bayes;SVM;Random Forest;quality},
- [3] M. S. Amzad Basha, K. Desai, S. Christina, M. M. Sucharitha and A. Maheshwari, "Enhancing red wine quality prediction through Machine Learning approaches with Hyperparameters optimization technique," 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichirappalli, India, 2023
- [4] M. V. Gupta and S. K, "Utilization of Random Forest Classifier (RFC) To Predict the Quality of Beverages," 2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI), Greater Noida, India, 2023
- [5] D. Oreški, I. Pihir and K. Cajzek, "Smart Agriculture and Digital Transformation on Case of Intelligent System for Wine Quality Prediction," 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 2021
- https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners
- https://xgboost.readthedocs.io/en/stable/
- https://www.ibm.com/topics/random-forest
- https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/
- https://optuna.org/

Acknowledgement

We express our deepest gratitude to Dr. Shruti Javkar for her exceptional mentorship, invaluable guidance, and unwavering support throughout the implementation of this engineering project. Dr. Javkar's profound expertise, patience, and dedication have been instrumental in shaping our perspectives and methodologies.

We would also like to extend our sincere thanks to the Department of Computer Engineering and Dr. Prasanna Shette, the Department Head, for his continuous support and encouragement throughout the duration of this project.