CSE422: Artificial Intelligence
Lab Project Report

Group No: 03

Group Members

1. Rajin Ibna Rajuanur Rahman [Student ID: 22101717]
2. MD. Mahfuzul Haque Mueen [Student ID: 22101715]

Date of Submission: 7th January 2025

Table of Contents

Introduction

Wine quality assessment is a critical aspect of the wine industry, ensuring consumer satisfaction and maintaining product standards. This project aims to analyze the Wine Quality Dataset to classify wines as either Good Quality or Bad Quality based on their physicochemical attributes. By leveraging machine learning models such as Logistic Regression, Decision Tree, Random Forest and K-Nearest Neighbors, this study evaluates the predictive performance of these algorithms in determining wine quality.

The primary problem this project addresses is the subjective and time-consuming nature of traditional wine quality evaluation methods, which often rely on human tasters. By creating a data-driven approach, this project aims to provide a more efficient, consistent and scalable solution for wine classification.

Furthermore, the motivation behind the project stems from the growing need for automation and accuracy in the wine industry, where maintaining quality control is essential. Additionally, the use of machine learning offers valuable insights into how chemical properties like acidity, sulfur dioxide content and alcohol concentration influence wine quality.

This project not only contributes to enhancing wine production processes but also showcases the power of data analysis and machine learning in solving real world problems, offering potential applications in other quality control domains.

Dataset Description

Source

Link: Wine Quality Dataset on Kaggle

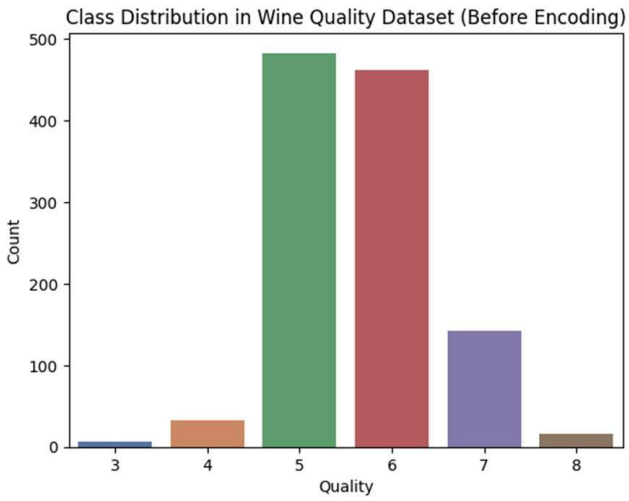
Modified Dataset: Our modified dataset before Data Preprocessing

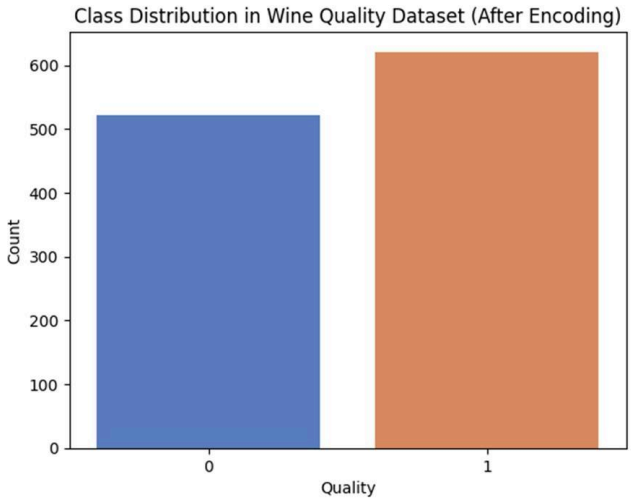
Reference: Yasser H. (Kaggle contributor)

The dataset comprises 11 features, excluding id and comment that were removed during preprocessing. The quality column initially contains numerical data. However, we transformed

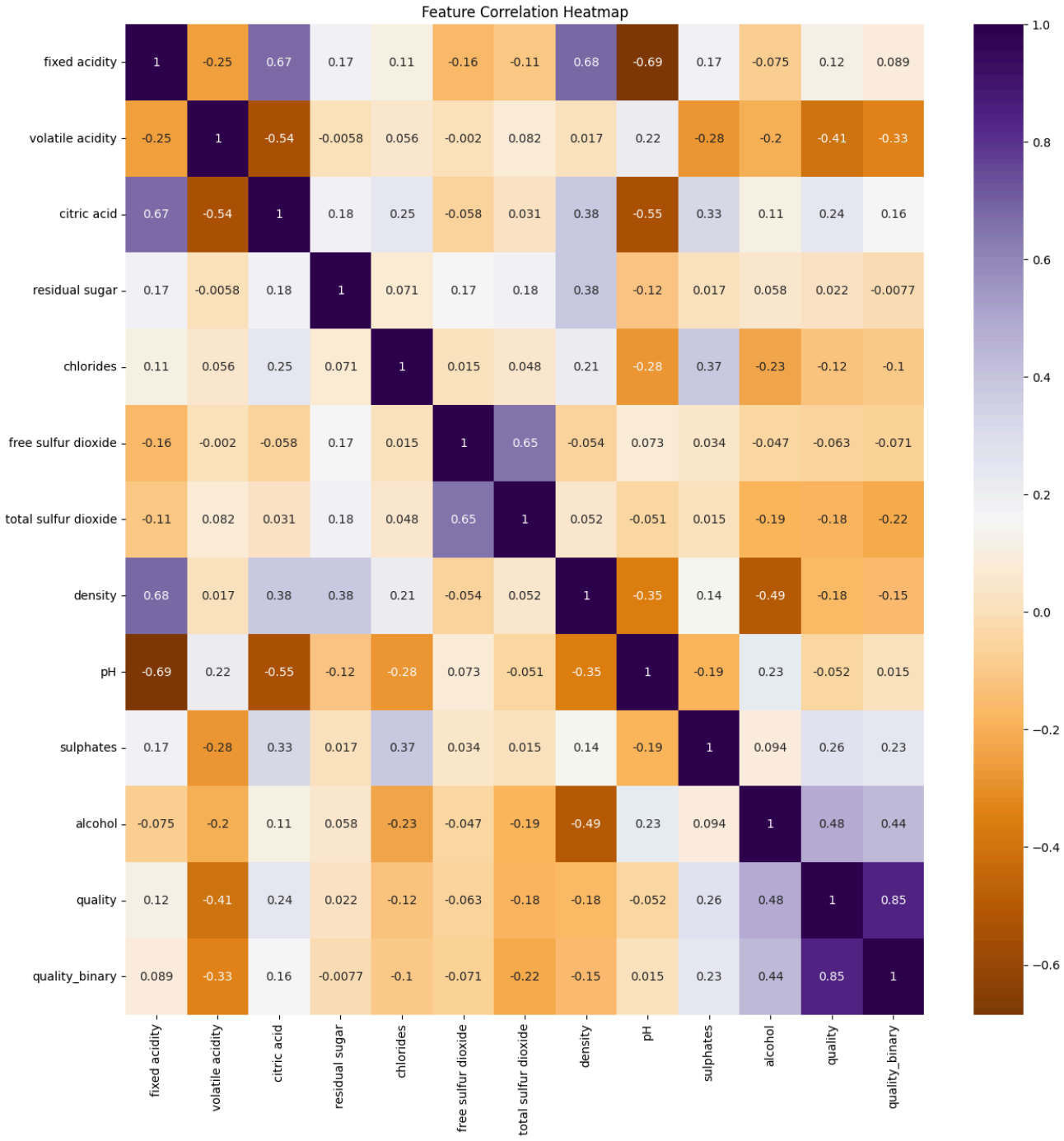
its values into discrete categories, which are stored in the quality_binary column. This new column now represents categorical data and serves as our target variable. Consequently, the problem has been converted into a classification task. The dataset contains 1143 data points (rows) with features like fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol and quality (target variable) all of which are quantitative. A heatmap analysis reveals the correlation between input features (e.g.: alcohol shows a strong positive correlation with quality) and the target feature, indicating how physicochemical properties affect wine quality.

The dataset is imbalanced, with unequal instances across unique quality classes. A bar chart shows that some classes (e.g.: quality 5 and 6) dominate, while others (e.g.: quality 3 and 8) have significantly fewer samples. After encoding the target variable into a binary format- 1 for Good Quality (If quality is greater or equal 6) and 0 for Bad Quality (If quality is less than 6), the class distribution remains imbalanced but simplified. Visualizations before and after encoding highlight this imbalance and guide model evaluation and selection.





Heatmap:


Feature Correlation Heatmap
fixed acidity 1 -0.25 0.67 0.17 0.11 -0.16 -0.11 0.68 -0.69 0.17 -0.075 0.12 0.089
volatile acidity -0.25 1 -0.54 -0.0058 0.056 -0.002 0.082 0.017 0.22 -0.28 -0.2 -0.41 -0.33
citric acid 0.67 -0.54 1 0.18 0.25 -0.058 0.031 0.38 -0.55 0.33 0.11 0.24 0.16
residual sugar 0.17 -0.0058 0.18 1 0.071 0.17 0.18 0.38 -0.12 0.017 0.058 0.022 -0.0077
chlorides 0.11 0.056 0.25 0.071 1 0.015 0.048 0.21 -0.28 0.37 -0.23 -0.12 -0.1
free sulfur dioxide -0.16 -0.002 -0.058 0.17 0.015 1 0.65 -0.054 0.073 0.034 -0.047 -0.063 -0.071
total sulfur dioxide -0.11 0.082 0.031 0.18 0.048 0.65 1 0.052 -0.051 0.015 -0.19 -0.18 -0.22
density 0.68 0.017 0.38 0.38 0.21 -0.054 0.052 1 -0.35 0.14 -0.49 -0.18 -0.15
pH -0.69 0.22 -0.55 -0.12 -0.28 0.073 -0.051 -0.35 1 -0.19 0.23 -0.052 0.015
sulphates 0.17 -0.28 0.33 0.017 0.37 0.034 0.015 0.14 -0.19 1 0.094 0.26 0.23
alcohol -0.075 -0.2 0.11 0.058 -0.23 -0.047 -0.19 -0.49 0.23 0.094 1 0.48 0.44
quality 0.12 -0.41 0.24 0.022 -0.12 -0.063 -0.18 -0.18 -0.052 0.26 0.48 1 0.85
quality_binary 0.089 -0.33 0.16 -0.0077 -0.1 -0.071 -0.22 -0.15 0.015 0.23 0.44 0.85 1
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
quality_binary
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Data Preprocessing

Problem- 01

NULL Values: The ‘total sulfur dioxide’ column contained missing values that could compromise data quality and model performance.

Solution- 01

Handling NULL Values: Used mean imputation to fill missing values, preserving data consistency without removing records. Because, missing values in "total sulfur dioxide" could distort analysis.

Problem- 02

Unnecessary Columns: The dataset included irrelevant columns such as Id and Comment which didn’t contribute to the prediction task.

Solution- 02

Dropping Unnecessary Columns: Removed these columns to streamline the dataset and avoid introducing noise. Because, Id and Comment columns were non-informative for the prediction.

Problem- 03

Class Imbalance: The quality column had multiple unique classes, leading to an uneven distribution of target labels, which could affect model performance.

Solution- 03

Encoding Target Labels: Converted quality into a binary format- quality ≥ 6 labeled as 1 (Good Quality) and < 6 labeled as 0 (Bad Quality), simplifying the classification task. Because, multi-class representation of quality caused imbalance and complexity in model training.

Problem- 04

Skewness in Feature Values: The input features exhibited varying scales and potential skewness, which could negatively influence the training of certain machine learning models.

Solution- 04

Feature Scaling: Standardized the features using StandardScaler, ensuring all features had a mean of 0 and a standard deviation of 1. Because, skewness and varying scales of feature values could hinder model convergence.

Feature Scaling

Feature scaling was applied to standardize the dataset and reduce skewness among input features using the StandardScaler method. This ensured that all features had a mean of 0 and a standard deviation of 1, preventing models from being biased toward features with larger scales. Standardization improved the performance and convergence of machine learning algorithms, particularly those sensitive to feature magnitudes, such as Logistic Regression and K-Nearest Neighbors.

Dataset Splitting

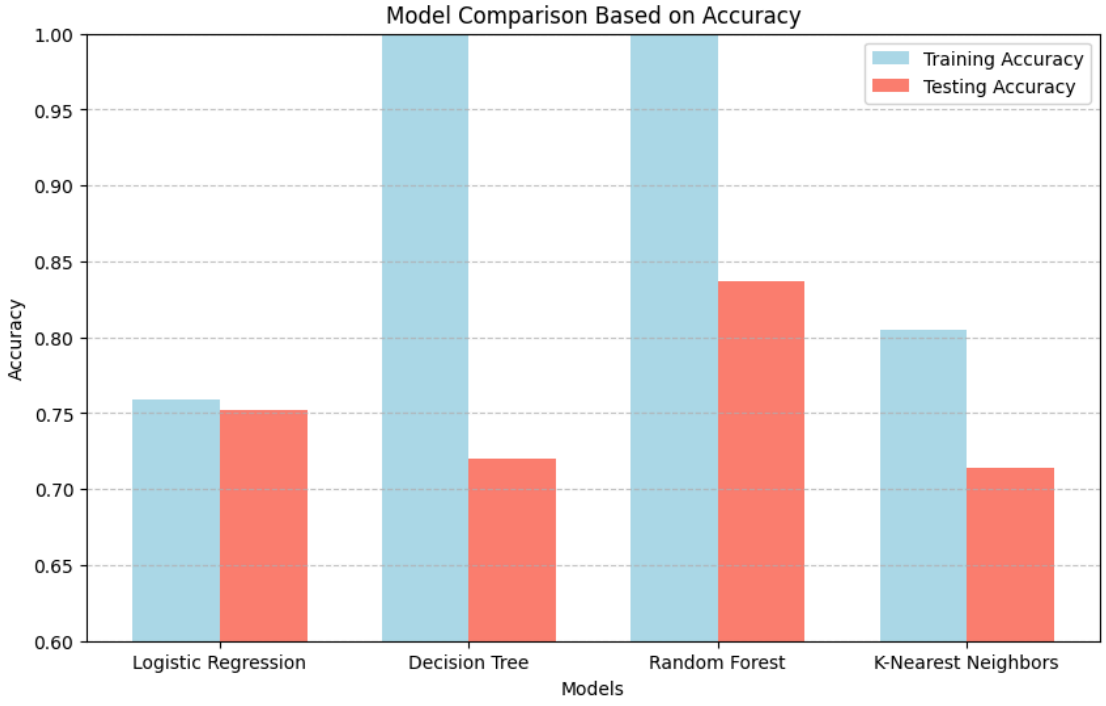
The dataset was split into training and testing sets using a 70 : 30 ratio to ensure robust model evaluation. The **train_test_split** function with stratification preserved the class distribution across splits. This approach ensures balanced representation of Good Quality and Bad Quality labels across splits, which preventing bias, especially in imbalanced datasets. The training set was used to train models, while the testing set validated their performance, ensuring reliable generalization to unseen data.

Model training and testing

1. Logistic Regression
2. Decision Tree
3. Random Forest Classifier
4. K-Nearest Neighbors

Model Selection / Comparison Analysis

The bar chart **Model Comparison Based on Accuracy** highlights the performance of four models- Logistic Regression, Decision Tree, Random Forest and K-Nearest Neighbors. Random Forest achieved the highest accuracy for both training and testing datasets, indicating strong generalization and performance. Logistic Regression and K-Nearest Neighbors showed moderate accuracy, with minimal overfitting. In contrast, the Decision Tree demonstrated high training accuracy but lower testing accuracy, suggesting overfitting. So according to our dataset, Random Forest emerged as the most reliable model, balancing accuracy and generalization effectively.



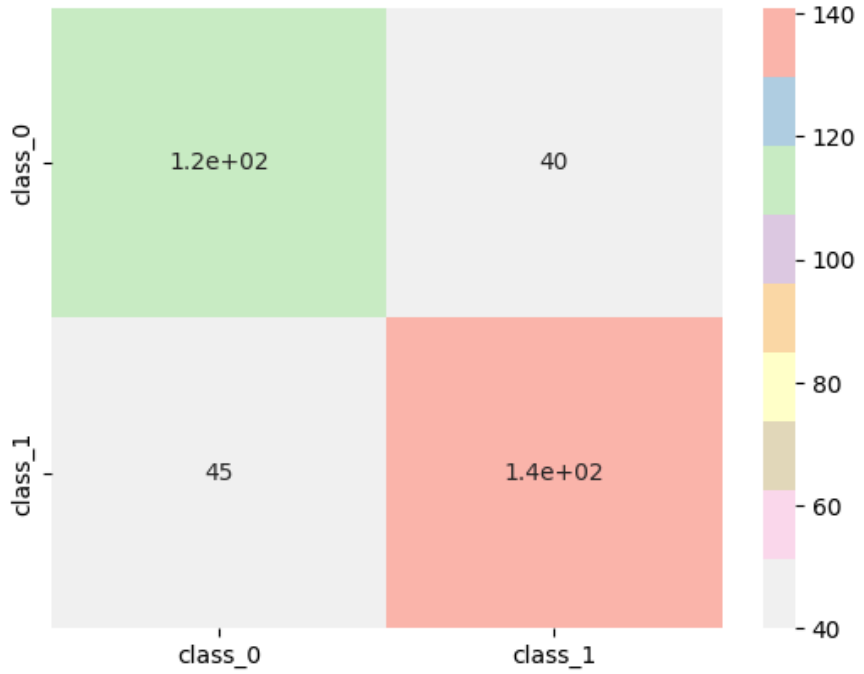
The Precision and Recall Comparison for each model:

The following table summarizes the precision and recall for each model evaluated on the wine quality dataset:

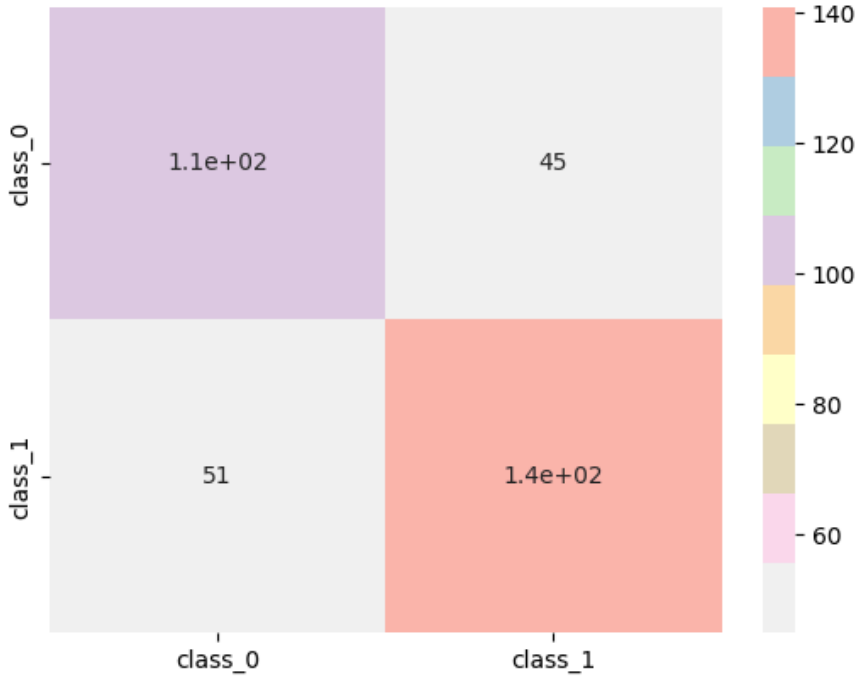
Model	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)
Logistic Regression	0.72	0.78	0.75	0.76
Decision Tree	0.70	0.73	0.68	0.76
Random Forest	0.82	0.85	0.83	0.84
K-Nearest Neighbors	0.70	0.73	0.66	0.76

Confusion Matrix for each Model:

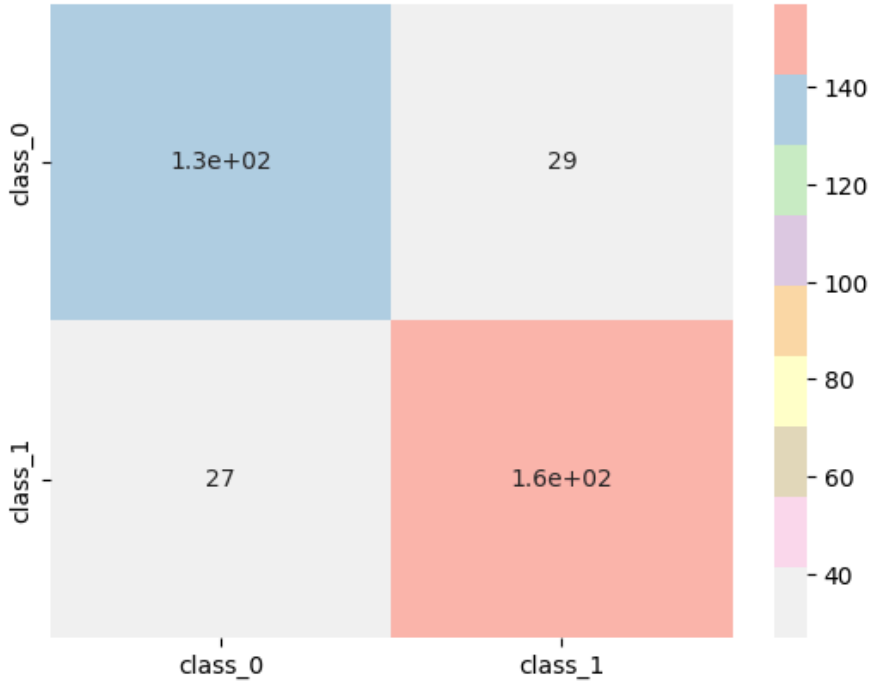
1. **Logistic Regression**



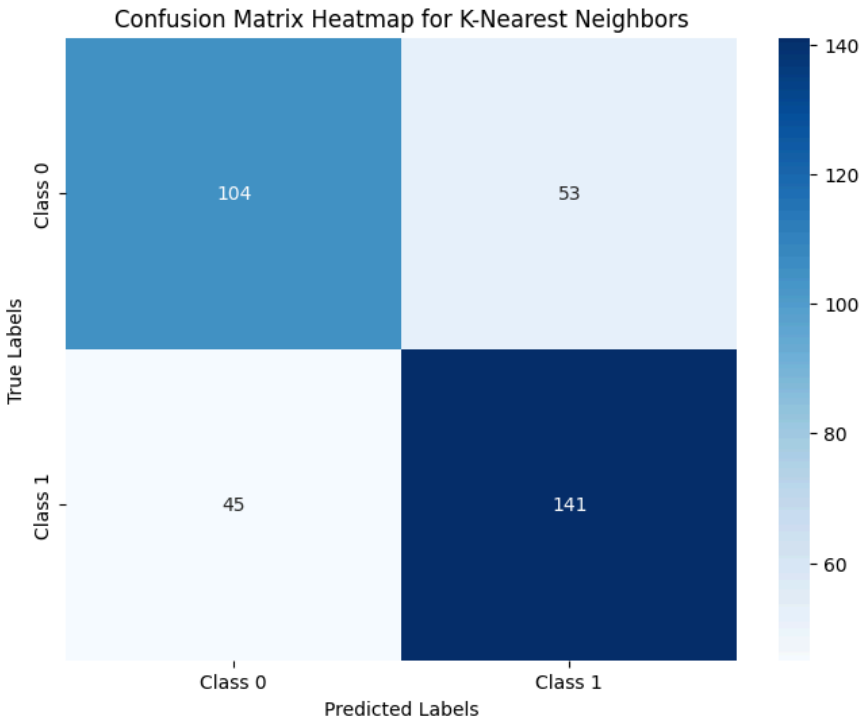
2. Decision Tree



3. Random Forest Classifier



4. K-Nearest Neighbors



Conclusion

This project utilized the wine quality dataset to classify wine samples into good or bad quality based on physicochemical attributes. Key steps included handling missing data, encoding quality levels, standardizing features and splitting the dataset with a stratified approach to preserve class distribution. Four machine learning models- Logistic Regression, Decision Tree, Random Forest and K-Nearest Neighbors were implemented, evaluated and compared based on accuracy, precision, recall and confusion matrices. Here, the Random Forest model gave the best results with the highest accuracy in both training and testing, along with superior precision and recall for both classes. Logistic Regression and K-Nearest Neighbors delivered competitive results, indicating their suitability for similar classification problems. However, the Decision Tree's lower recall hinted at overfitting. Finally, the project highlights the importance of feature scaling, stratified splitting and robust model evaluation metrics in achieving reliable predictions. Random Forest is recommended for future applications in wine quality classification.