Artificial Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

Wine Quality Dataset

Supervised By:

Dr. Kirandeep Singh

Submitted By:

Pushpinder singh, 2210990(G8)

Pratham khanna 2210990673 (G8)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

ABSTRACT

This study sounds like a comprehensive exploration of wine quality, blending sensory evaluations with detailed chemical analysis. By examining both red and white wines from the Vinho Verde region in Portugal, it captures a broad spectrum of attributes that contribute to the overall perception of wine quality.

The initial observations regarding the differences between red and white wines in terms of chemical composition and sensory profiles provide a solid foundation for further investigation. It's interesting to note the distinct flavor profiles and mouthfeel sensations attributed to red wines, which are likely influenced by higher levels of compounds like phenols and tannins.

Correlation analysis revealing associations between individual attributes and wine quality ratings adds depth to the understanding of how specific characteristics impact overall perception. The nuances observed, such as the varying relationships of acidity, pH, and sulphates with perceived quality, hint at the complexity of wine appreciation.

Employing predictive modeling techniques to forecast wine quality based on intrinsic characteristics is a noteworthy approach. By harnessing machine learning algorithms, the study aims to unravel the intricate connections between attributes and their collective influence on wine quality. This predictive modeling not only aids in understanding the factors driving quality but also offers practical insights for wine producers looking to refine their production processes.

Overall, this research contributes significantly to the field by bridging the gap between sensory evaluations and chemical analysis. It emphasizes the importance of data-driven approaches in comprehensively assessing wine quality, which can ultimately benefit both producers and consumers in the wine industry.

TABLE OF CONTENTS

INTRODUCTION

This study outlines a comprehensive approach to understanding and evaluating wine quality, focusing on the Vinho Verde region of Portugal. By leveraging a rich dataset encompassing sensory evaluations and physicochemical properties of both red and white wines, the research aims to uncover the underlying factors that influence perceived quality.

Objective: The primary objective of this study is to comprehensively understand and evaluate wine quality in the Vinho Verde region of Portugal. This involves leveraging a rich dataset containing sensory evaluations and physicochemical properties of both red and white wines. Through the application of techniques from exploratory data analysis (EDA), descriptive statistics, and machine learning, the research aims to unravel the complexities of wine quality assessment. By scrutinizing the relationships between various attributes and wine quality ratings, the study seeks to uncover the nuanced factors that shape consumer perceptions.

The utilization of predictive modeling techniques to forecast wine quality based on intrinsic characteristics adds another layer of depth to the research. This approach not only aids in understanding the factors driving quality but also offers practical applications for wine producers seeking to optimize their processes and enhance product quality.

Aim: The overarching aim of this study is to bridge the gap between sensory evaluations and chemical analysis in the evaluation of wine quality. By doing so, it endeavors to provide a holistic perspective on wine quality assessment. Additionally, the study aims to offer practical applications for wine producers by utilizing predictive modeling techniques to forecast wine quality based on intrinsic characteristics. Through these efforts, the research aims to contribute to the broader discourse within the wine industry and foster continued innovation in production practice

1.1 Background: Wine Quality Dataset

In recent years, the wine industry has embraced data-driven methods to enhance production and ensure quality. The availability of large-scale datasets, including detailed wine attributes and sensory evaluations, has provided a unique opportunity to explore wine quality intricacies.

The dataset under study comprises sensory evaluations and physicochemical properties of red wine from Portugal's Vinho Verde region, known for its diverse flavors and historical significance.

Traditionally, wine quality evaluation relied on expert sensory assessments and chemical analysis. However, advancements in data analytics and machine learning offer new insights into the relationship between attributes and consumer preferences.

By utilizing advanced statistical methods and predictive modeling, researchers can uncover hidden patterns within the dataset, providing valuable insights for producers and enthusiasts. Understanding these factors is crucial for adapting to changing consumer preferences in a competitive market.

This study aims to delve into the dataset, uncovering the determinants of wine quality and developing predictive models for quality ratings based on intrinsic characteristics. Bridging sensory evaluations with chemical analysis, the research seeks to contribute to viticulture and oenology discussions, fostering innovation in wine production practices.

1.2 Objective: Wine Quality Dataset

The objectives of this study are designed to guide a comprehensive evaluation of the wine quality dataset, aiming to unravel the complexities of wine quality assessment and provide actionable insights for wine producers. Each objective is crafted to contribute meaningfully to the broader understanding of wine quality evaluation. Here's a detailed delineation of each objective:

1. **Quality Assessment**: The primary objective is to conduct a thorough evaluation of red wine quality based on sensory evaluations and physicochemical properties. Utilizing descriptive statistics and exploratory data analysis (EDA) techniques, this objective seeks to uncover patterns indicative of high-quality wines. By assessing sensory descriptors and chemical composition, the aim is to develop a holistic understanding of wine quality.

2. **Attribute Analysis**: Examine attributes like acidity, alcohol content, and volatile compounds to identify those closely associated with high-quality wines and understand flavor variations

3. **Relationship Identification:** This objective focuses on uncovering relationships between attributes and wine quality ratings. Through correlation analysis and statistical modeling, the study aims to identify significant associations and dependencies influencing consumer perceptions.

4. **Predictive Modeling**: Building predictive models to forecast wine quality ratings based on intrinsic characteristics is a fundamental objective. Using machine learning algorithms and regression techniques, the study aims to develop robust models capable of accurately predicting wine quality. Feature selection and model optimization enhance the accuracy and reliability of these models.

5. **Interpretation and Actionable Insights**: The final objective is to interpret results and derive actionable insights for wine producers. By synthesizing findings from data exploration and modeling efforts, the study aims to provide practical recommendations for optimizing production processes and enhancing product quality. These insights empower informed decision-making within the wine industry.

In summary, these objectives drive the study's goals of understanding the factors influencing wine quality and providing actionable insights for wine producers. Through rigorous analysis and interpretation efforts, the study aims to contribute to the advancement of wine quality assessment practices and facilitate informed decision-making within the wine industry.

PROBLEM DEFINITION AND REQUIREMENTS

In the realm of viticulture and oenology, the assessment of wine quality is a multifaceted endeavor that encompasses a diverse range of sensory, chemical, and qualitative attributes. Understanding and evaluating wine quality is of paramount importance for both wine producers and consumers, as it influences purchasing decisions, market competitiveness, and overall satisfaction with the product. However, the subjective nature of wine quality assessment, coupled with the complexity of factors influencing wine characteristics, poses significant challenges for industry professionals seeking to optimize production processes and enhance product quality.

Problem Definition:

This study aims to tackle the challenge of comprehensively evaluating wine quality using a dataset containing sensory evaluations and physicochemical properties of red and white wines from Portugal's Vinho Verde region. The overarching objective is to uncover patterns, correlations, and trends within the dataset to provide actionable insights for wine producers, thereby optimizing production processes and enhancing product quality.

Requirements:

1. **Comprehensive Dataset**: The study requires access to a diverse and comprehensive dataset that includes detailed information on sensory evaluations, chemical composition, and other relevant attributes of red and white wines from the Vinho Verde region. This dataset should encompass a wide range of wines, including different varietals, vintages, and production methods, to ensure representativeness and robust analysis.

2. **Data Preprocessing**: Prior to analysis, the dataset needs to undergo preprocessing to ensure data quality and consistency. This includes handling missing values, dealing with outliers, and standardizing data formats to facilitate analysis. Additionally, categorical variables may need to be encoded or transformed into numerical representations for compatibility with machine learning algorithms.

3. **Exploratory Data Analysis(EDA):** EDA serves as a foundational step in understanding the underlying structure of the dataset and identifying patterns or trends that may inform subsequent analysis. Through descriptive statistics, visualization techniques, and correlation analysis, EDA enables researchers to gain insights into the distribution, variability, and relationships between attributes within the dataset.

4. **Feature Engineering:** Feature engineering involves transforming or creating new features from the existing dataset to improve model performance. This may include generating interaction terms, deriving domain-specific features, or applying dimensionality reduction techniques to extract meaningful information from the data. Feature engineering aims to enhance the predictive power of models by providing them with relevant input variables.

5. **Predictive Modeling:** Building predictive models to forecast wine quality ratings based on intrinsic characteristics is a core component of this study. Machine learning algorithms such as regression, classification, and ensemble methods will be employed to develop robust models capable of accurately predicting wine quality from a set of selected attributes. Model evaluation and validation techniques will be used to assess the performance of predictive models and ensure their reliability for real-world applications.

6. **Interpretation and Actionable Insights:** The ultimate goal of this study is to derive actionable insights from the analysis that can inform decision-making and drive improvements in wine production practices. This requires interpreting the results of predictive modeling efforts, identifying the key drivers of wine quality, and providing practical recommendations for wine producers to optimize production processes and enhance product quality.

7. **Communication and Dissemination:** Effective communication of findings and recommendations is essential for ensuring the impact and relevance of the study within the wine industry. Clear and concise reporting of results, accompanied by visualizations and explanatory narratives, facilitates understanding and uptake of insights by industry stakeholders.

   By meeting these requirements and conducting a rigorous analysis of the wine quality dataset, this study aims to contribute valuable insights to the viticulture and oenology domain, fostering innovation and excellence in wine production practices. Through a combination of data-driven approaches, statistical analysis, and domain expertise, we seek to address the challenges inherent in wine quality assessment and provide actionable recommendations for industry professionals to enhance the quality and diversity of wines available to consumers.

PROPOSED DESIGN AND METHODOLOGY

This section outlines the proposed design and methodology for analyzing the wine quality dataset. The approach encompasses several stages, including data preprocessing, exploratory data analysis (EDA), feature engineering, predictive modeling, and interpretation of results.

1.**Data Preprocessing:** The first step involves preprocessing the wine quality dataset to ensure its suitability for analysis. This includes handling missing values, dealing with outliers, and encoding categorical variables if present. Additionally, feature scaling may be applied to standardize the numerical attributes, ensuring that all features contribute equally to the analysis.

2. **Exploratory Data Analysis (EDA):** EDA plays a crucial role in understanding the underlying structure of the dataset and identifying patterns or trends that may inform subsequent analysis. Key components of EDA include:

• Univariate Analysis: Examining the distribution of individual attributes (e.g., acidity levels, alcohol content) through histograms, box plots, and summary statistics.

• Bivariate Analysis: Investigating relationships between pairs of attributes using scatter plots or correlation matrices to identify potential correlations or dependencies.

• Multivariate Analysis: Exploring interactions between multiple attributes simultaneously to uncover complex relationships and patterns within the dataset. EDA serves as a foundation for feature selection and engineering, guiding the identification of relevant attributes for predicting wine quality.

3. Feature Engineering: Feature engineering involves transforming or creating new features from the existing dataset to improve model performance. Techniques may include:

• Polynomial Features: Generating polynomial features to capture nonlinear relationships between attributes.

• Interaction Terms: Creating interaction terms to represent the combined effect of multiple attributes on wine quality.

• Domain-specific Features: Incorporating domain knowledge to derive features that are known to influence wine quality, such as acidity ratios or flavor profiles. Feature engineering aims to enhance the predictive power of the model by providing it with meaningful input variables.

4. Predictive Modeling: Predictive modeling involves training machine learning algorithms to predict wine quality ratings based on selected features. The following steps outline the predictive modeling process:

- Model Selection: Choosing appropriate algorithms for regression or classification tasks based on the nature of the target variable (e.g., continuous or categorical).

- Model Training: Splitting the dataset into training and testing sets to train the models on a subset of the data and evaluate their performance on unseen data.

- Model Evaluation: Assessing the performance of each model using evaluation metrics such as mean squared error (MSE) for regression tasks or accuracy, precision, and recall for classification tasks.

- Hyperparameter Tuning: Fine-tuning model hyperparameters using techniques like grid search or random search to optimize model performance.

5. Interpretation of Results: Interpreting the results of predictive modeling involves analyzing model coefficients, feature importance scores, and prediction errors to gain insights into the factors driving wine quality. Visualizations such as feature importance plots or partial dependence plots can aid in understanding the relationship between attributes and predicted outcomes.

Conclusion: By following the proposed design and methodology, we aim to gain valuable insights into the factors influencing wine quality and develop predictive models capable of accurately forecasting wine quality ratings. The results of this analysis can provide actionable recommendations for wine producers to optimize production processes, enhance product quality, and meet consumer preferences effectively.

5.1 File Structure:

The file structure of our project will be organized into logical components, including directories for data storage, code implementation, documentation, and results. Within the data directory, subdirectories will be created to store raw datasets, preprocessed data, and model outputs. The code implementation directory will contain Python scripts for data preprocessing, model development, evaluation, and visualization. Documentation will include README files providing instructions for project setup and usage, as well as any additional documentation related to code implementation and methodology. Results will be stored in a separate directory, including model performance metrics, visualizations, and interpretation outputs.

5.2 Algorithms Used:

Our methodology entails the exploration of diverse machine learning algorithms within the AIML paradigm to predict quality of wine accurately, This includes:

l. **Logistic Regression:** A linear regression model utilized for binary classification tasks, logistic regression is apt for estimating the probability of quality based on input features.

2. **Decision Trees:** Decision tree models divide the feature space into hierarchical decision rules, enabling interpretable and nonlinear relationships between predictor variables and quality outcomes.

3. **Support Vector Machines (S V M):** SVM is a supervised learning algorithm used for classification tasks. It's proficient in handling nonlinear decision boundaries, often achieved through kernel functions, thereby aiding in robust quality prediction.

4. **Random Forest:** Random Forest, a powerful ensemble learning technique, constructs multiple decision trees and aggregates their predictions. It's adept at capturing complex relationships in the data, contributing to improved predictive accuracy.

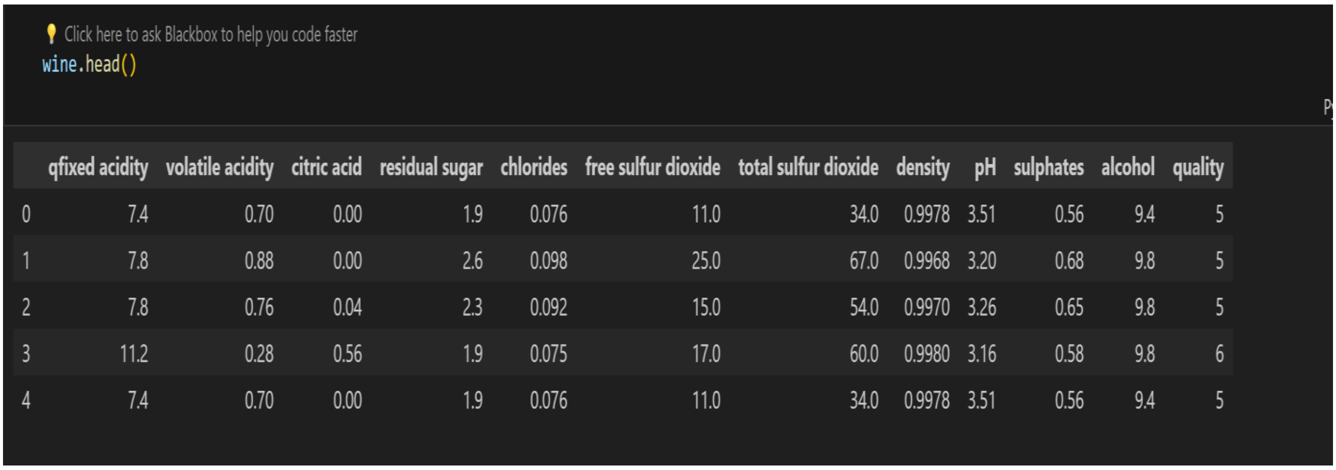
5. **k-Nearest Neighbors (k-NN):** k-NN is a non-parametric algorithm that classifies data points based on the majority class of their nearest neighbors in feature space. It's particularly useful in capturing local patterns and can offer insights into potential clusters of qualitys

By employing this diverse set of algorithms, we aim to identify the most suitable model architecture for obesity prediction, considering factors such as predictive performance, interpretability, and computational efficiency. Through rigorous experimentation and validation,
we seek to develop a robust predictive model capable of accurately identifying individuals at risk
of obesity, thereby facilitating targeted intervention strategies and improving public health outcomes.

RESULTS

ANALYSIS AND MODEL EVALUATION

In this section, we present a detailed analysis of the results obtained from our AI/ML wine quality project. We begin by showcasing the graphical representations of key metrics and performance indicators, followed by an overview of the models utilized along with their corresponding accuracies.

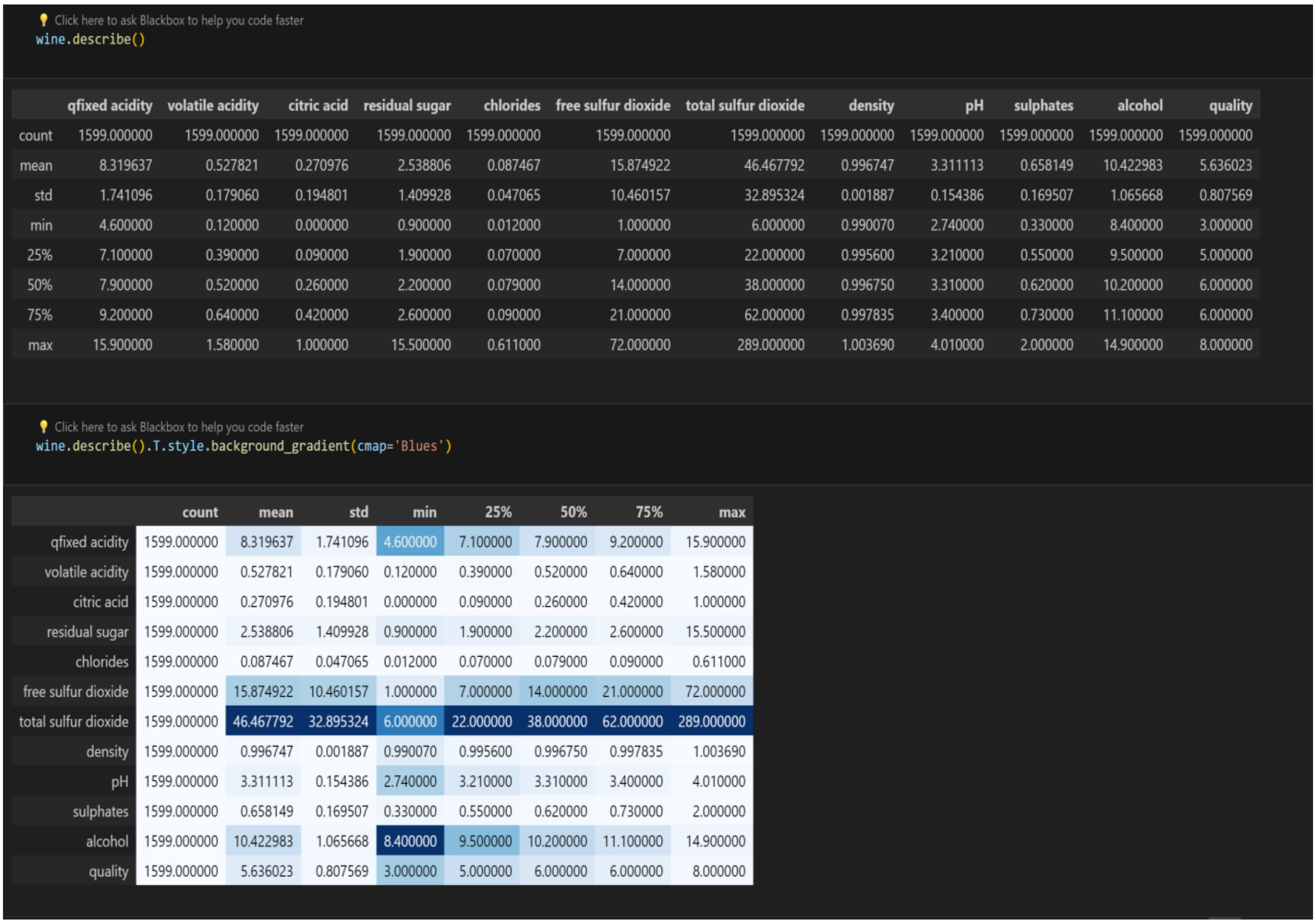


```
wine.head()
```

	qfixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Data.head()- Show first five rows and all columns of dataset.

Data.describe(): it provides summary statistics for each numerical feature in the dataset. These statistics usually include count (number of non-null values), mean, standard deviation, minimum, 25th percentile (Q1), median (50th percentile or Q2), 75th percentile (Q3), and maximum.



Click here to ask Blackbox to help you code faster

```
wine.describe()
```

	qfixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

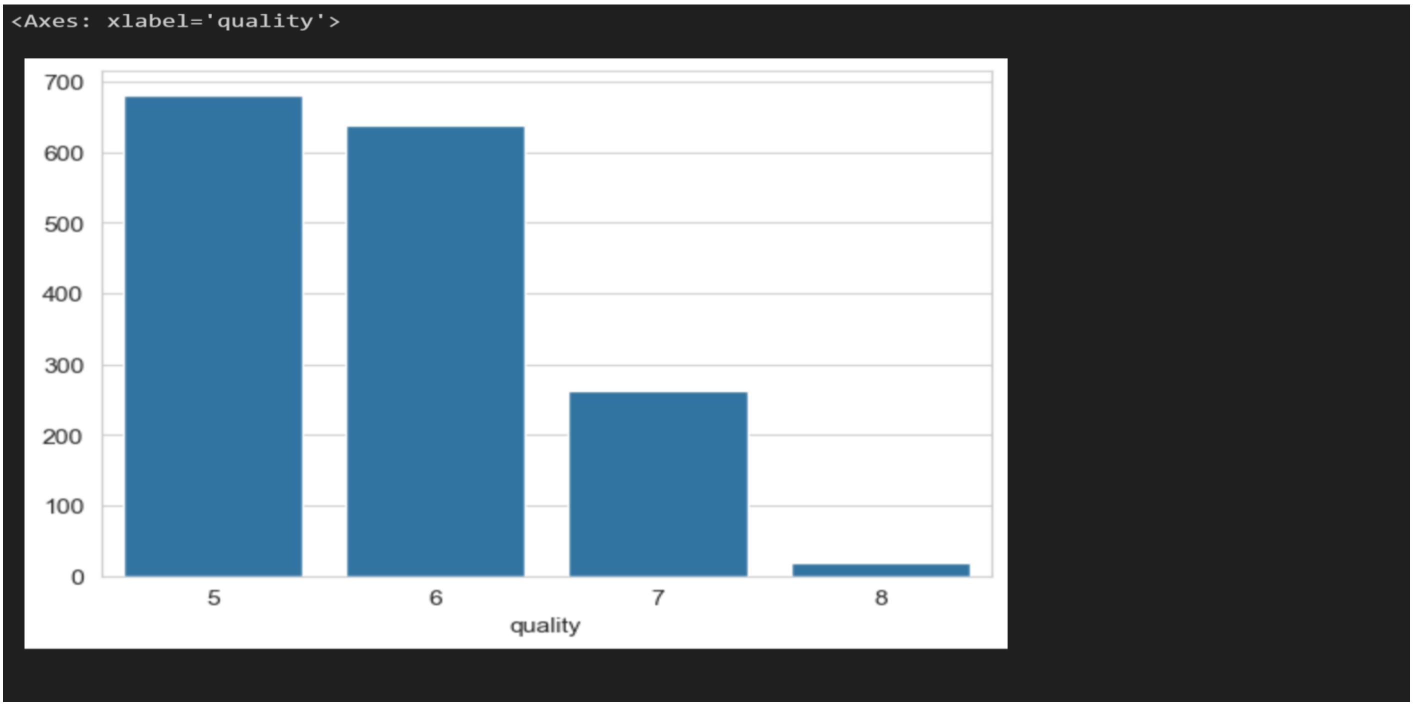
Click here to ask Blackbox to help you code faster

```
wine.describe().T.style.background_gradient(cmap='Blues')
```

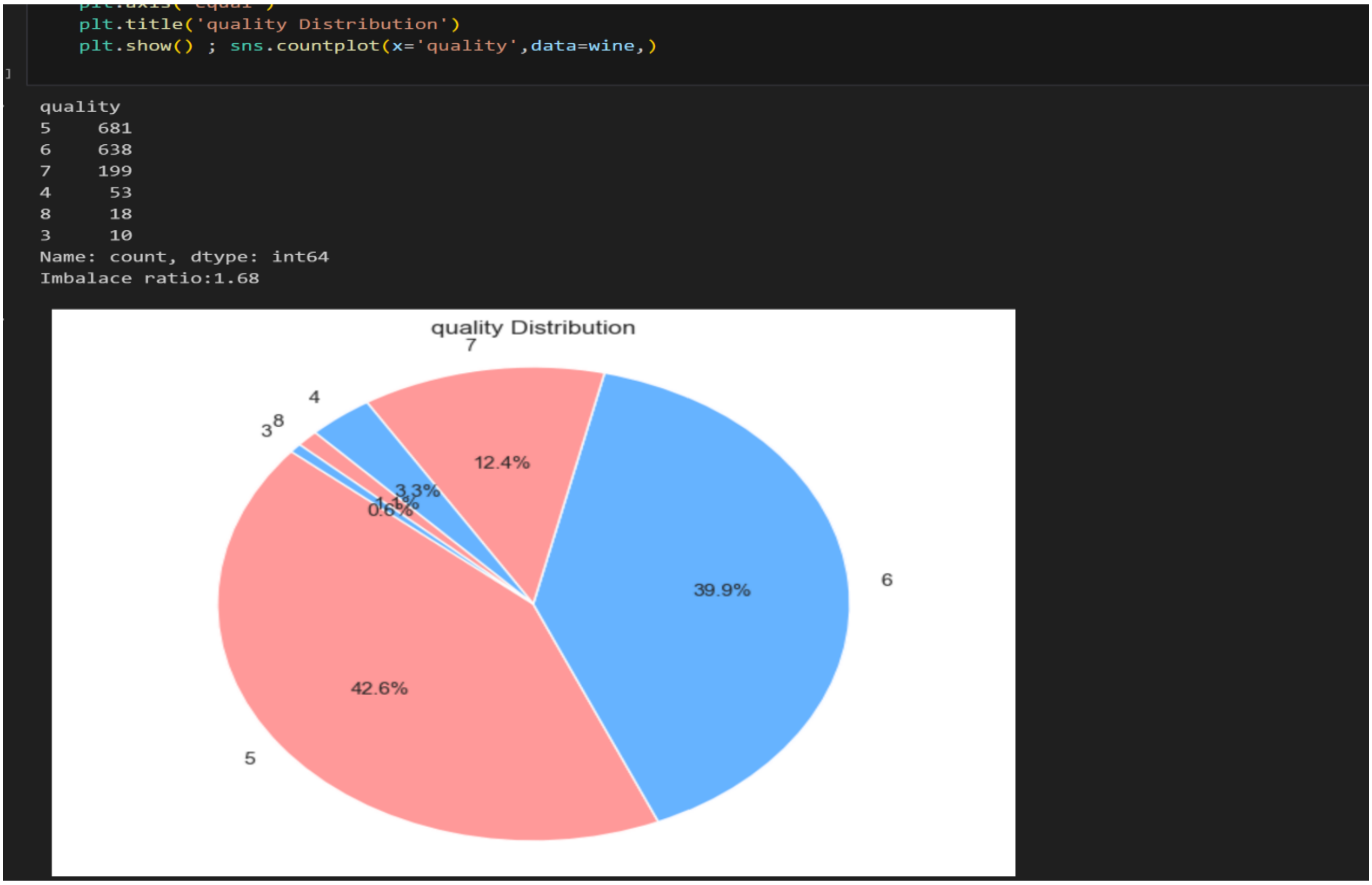
	count	mean	std	min	25%	50%	75%	max
qfixed acidity	1599.000000	8.319637	1.741096	4.600000	7.100000	7.900000	9.200000	15.900000
volatile acidity	1599.000000	0.527821	0.179060	0.120000	0.390000	0.520000	0.640000	1.580000
citric acid	1599.000000	0.270976	0.194801	0.000000	0.090000	0.260000	0.420000	1.000000
residual sugar	1599.000000	2.538806	1.409928	0.900000	1.900000	2.200000	2.600000	15.500000
chlorides	1599.000000	0.087467	0.047065	0.012000	0.070000	0.079000	0.090000	0.611000
free sulfur dioxide	1599.000000	15.874922	10.460157	1.000000	7.000000	14.000000	21.000000	72.000000
total sulfur dioxide	1599.000000	46.467792	32.895324	6.000000	22.000000	38.000000	62.000000	289.000000
density	1599.000000	0.996747	0.001887	0.990070	0.995600	0.996750	0.997835	1.003690
pH	1599.000000	3.311113	0.154386	2.740000	3.210000	3.310000	3.400000	4.010000
sulphates	1599.000000	0.658149	0.169507	0.330000	0.550000	0.620000	0.730000	2.000000
alcohol	1599.000000	10.422983	1.065668	8.400000	9.500000	10.200000	11.100000	14.900000
quality	1599.000000	5.636023	0.807569	3.000000	5.000000	6.000000	6.000000	8.000000

GRAPHICAL REPRESENTATIONS

1. **Quality of the wine**



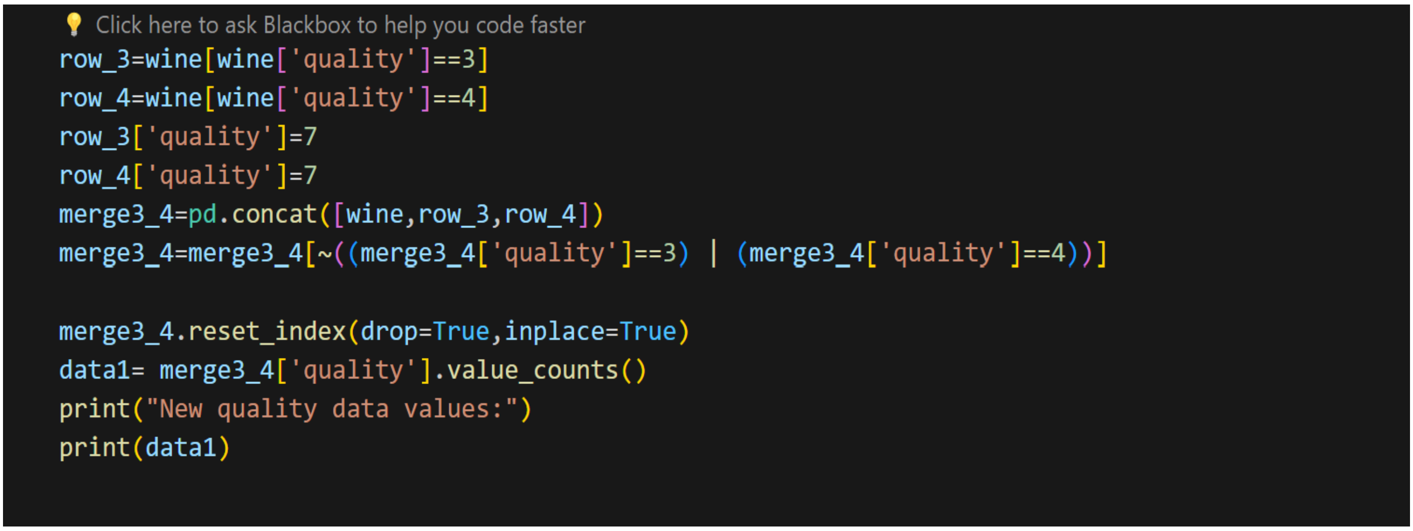
```
<Axes: xlabel='quality'>
```



```
    plt.axis('equal')
    plt.title('quality Distribution')
    plt.show() ; sns.countplot(x='quality',data=wine,)
```

```
quality
5    681
6    638
7    199
4     53
8     18
3     10
Name: count, dtype: int64
Imbalace ratio:1.68
```

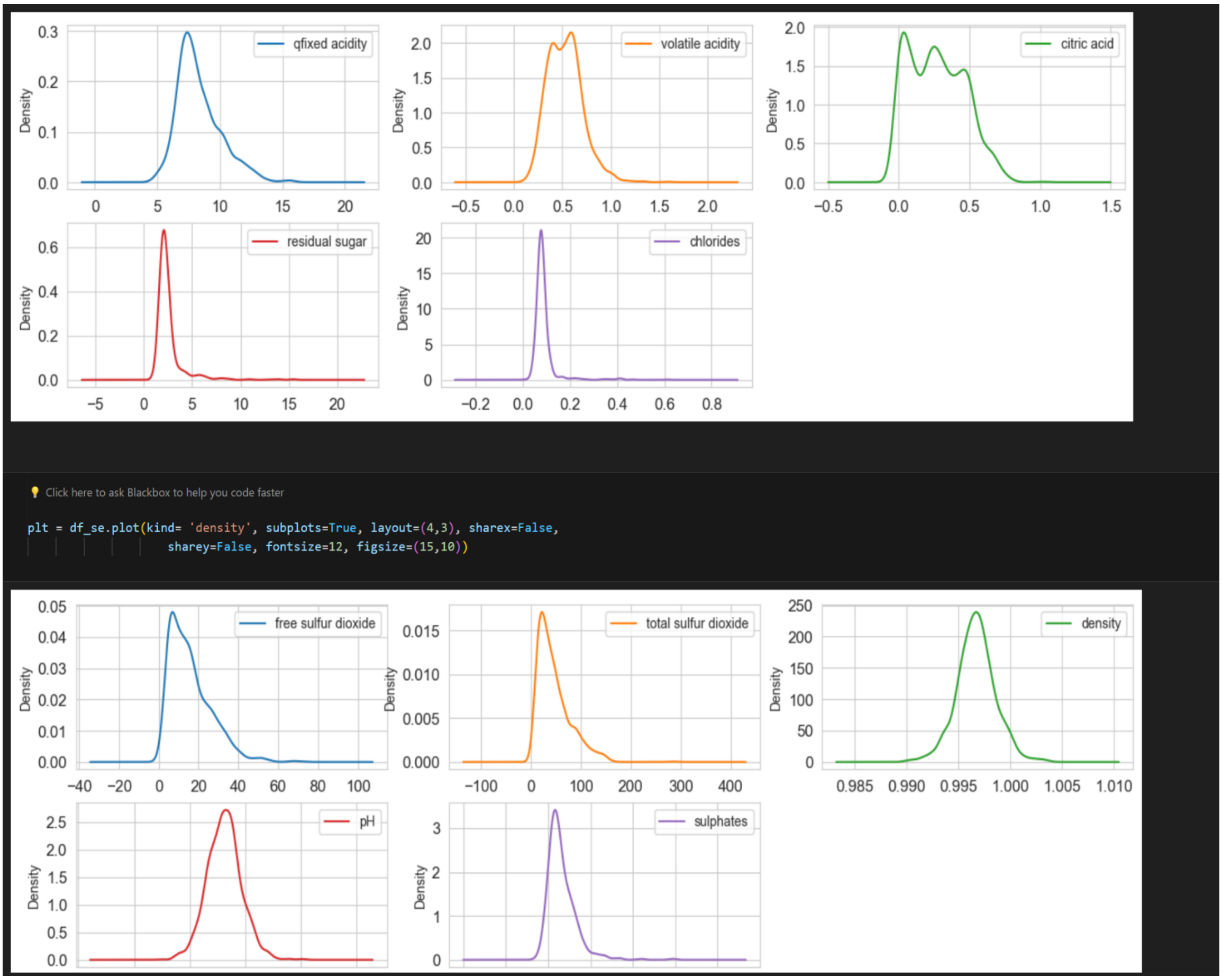
2. Data Cleaning:



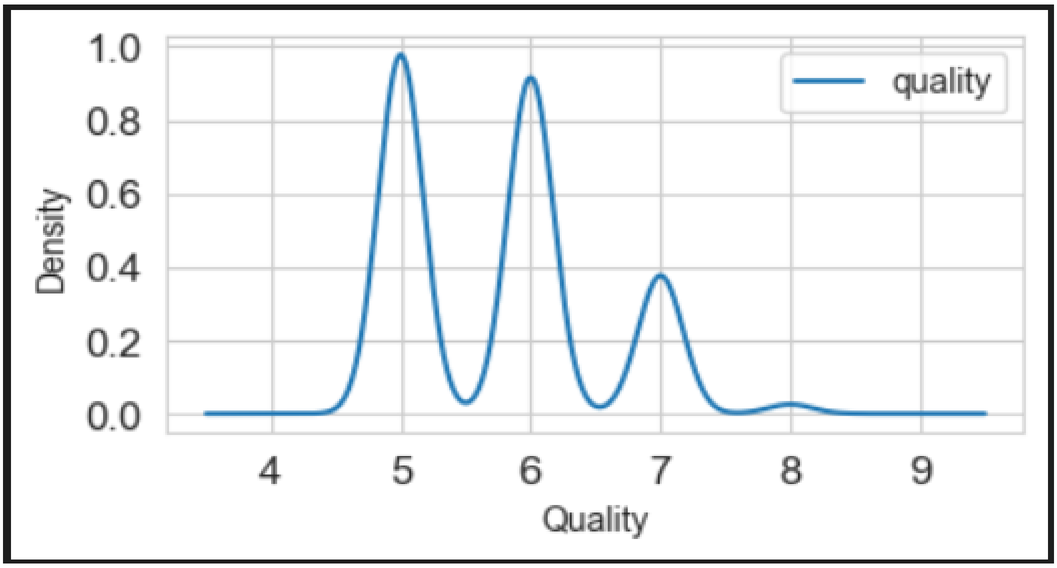
```
row_3=wine[wine['quality']==3]
row_4=wine[wine['quality']==4]
row_3['quality']=7
row_4['quality']=7
merge3_4=pd.concat([wine,row_3,row_4])
merge3_4=merge3_4[~((merge3_4['quality']==3) | (merge3_4['quality']==4))]

merge3_4.reset_index(drop=True,inplace=True)
data1= merge3_4['quality'].value_counts()
print("New quality data values:")
print(data1)
```

3. **Density curve of all worst suffix:**

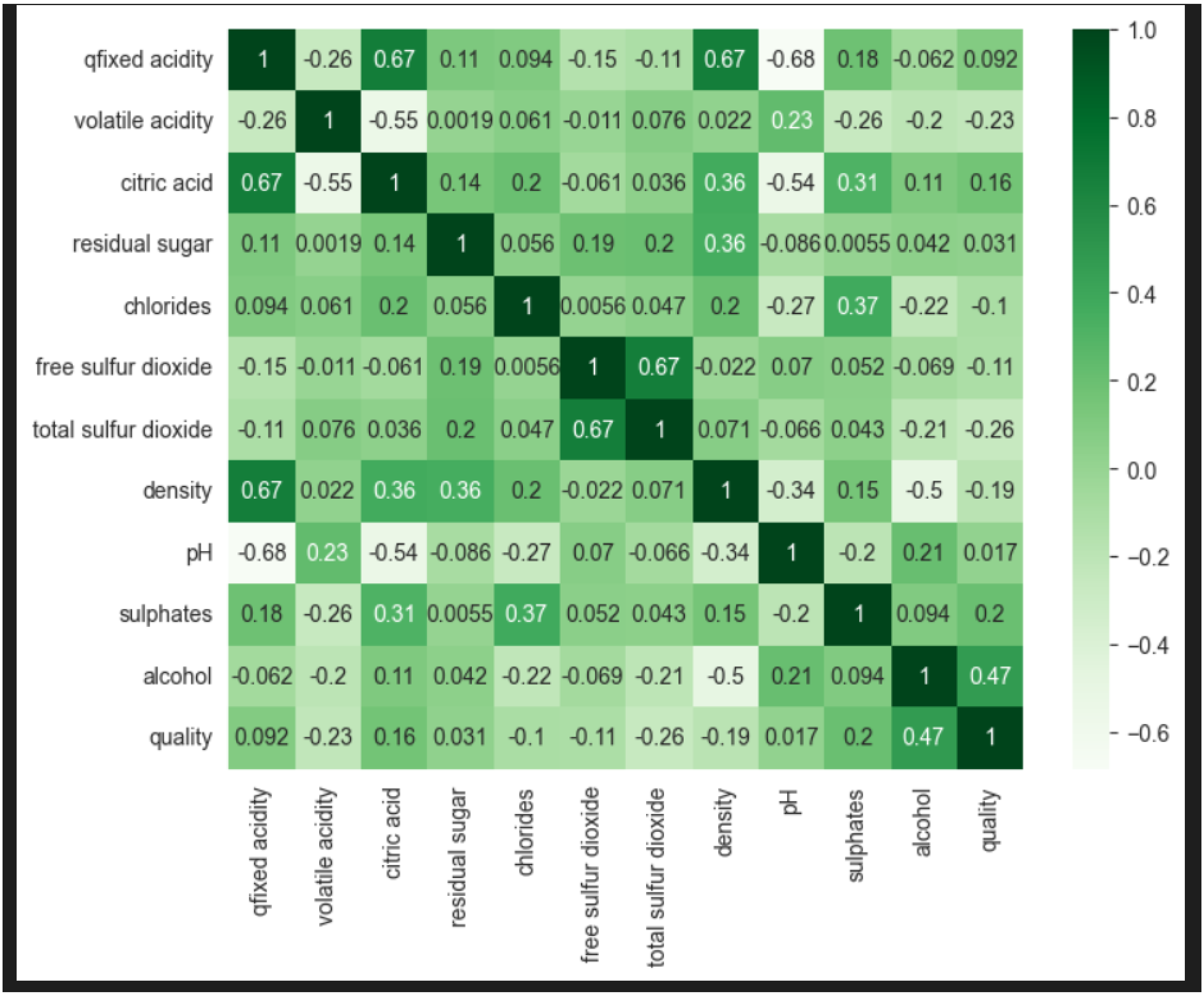
A density plot, also known as a kernel density plot, provides a visual representation of the distribution of a continuous variable. It displays the probability density function of the data, showing where values are concentrated and where they are sparse. The plot is formed by smoothing histograms, resulting in a continuous curve that represents the underlying distribution. Density plots are particularly useful for understanding the shape of the data distribution, identifying peaks and modes, and comparing distributions between different groups or variables.



```
plt = df_se.plot(kind= 'density', subplots=True, layout=(4,3), sharex=False,
                 sharey=False, fontsize=12, figsize=(15,10))
```

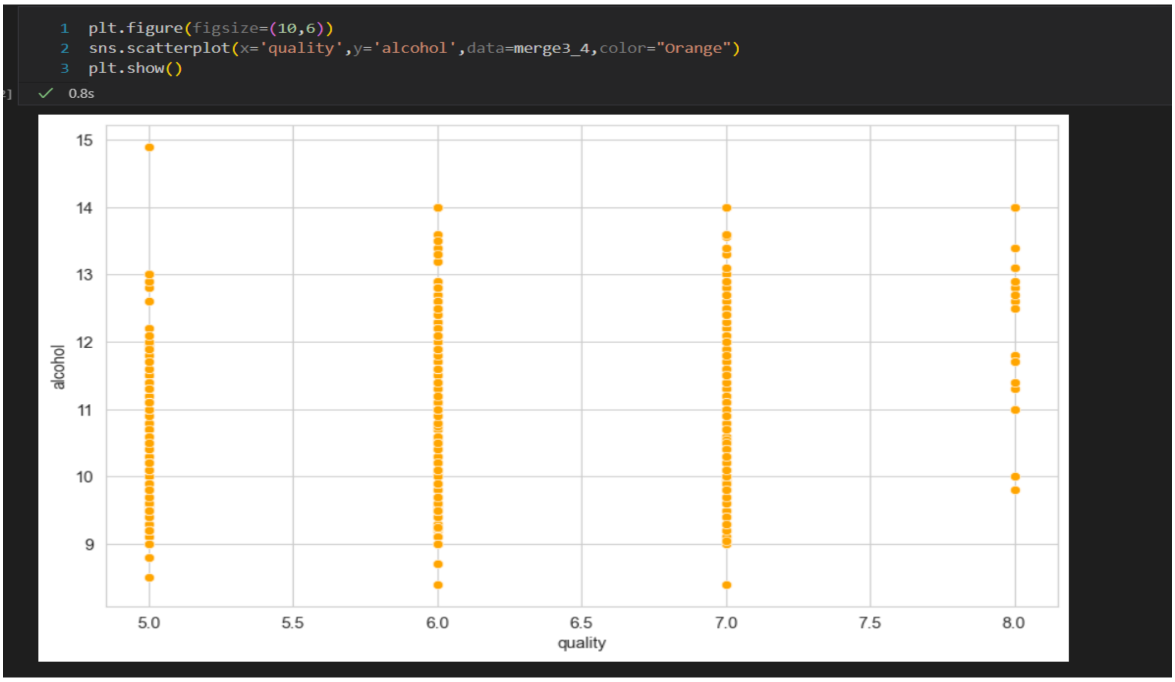


4. **Histogram:**



	qfixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
qfixed acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.092
volatile acidity	-0.26	1	-0.55	0.0019	0.061	-0.011	0.076	0.022	0.23	-0.26	-0.2	-0.23
citric acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.16
residual sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.031
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.1
free sulfur dioxide	-0.15	-0.011	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.11
total sulfur dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.066	0.043	-0.21	-0.26
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.19
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.066	-0.34	1	-0.2	0.21	0.017
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.2
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.47
quality	0.092	-0.23	0.16	0.031	-0.1	-0.11	-0.26	-0.19	0.017	0.2	0.47	1

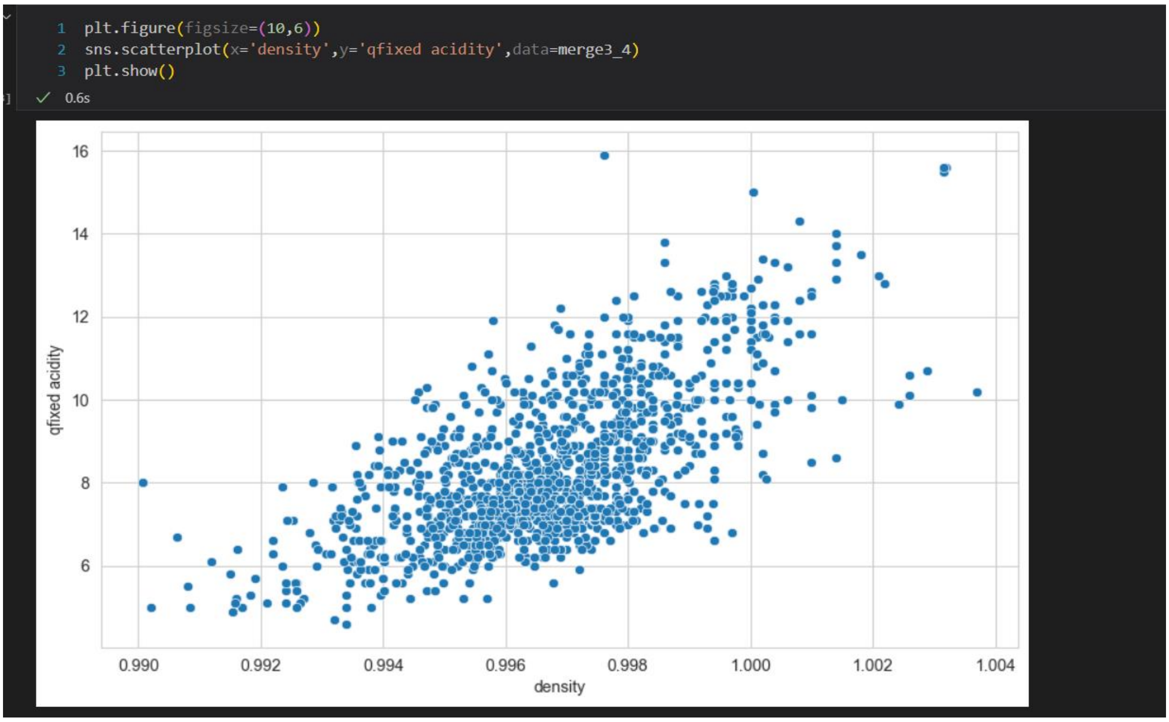
Scatter plot between Alcohol and Qaulity:



```
plt.figure(figsize=(10,6))
sns.scatterplot(x='quality',y='alcohol',data=merge3_4,color="Orange")
plt.show()
```

✓ 0.8s

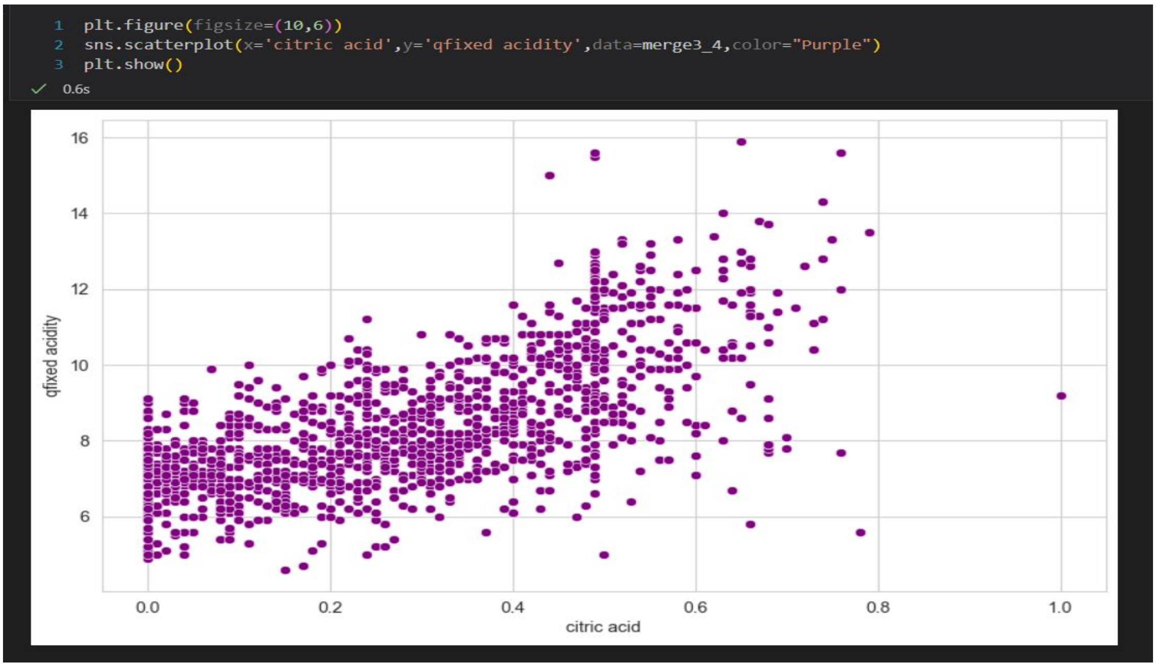
Scatter plot between qfixed acidity and Density:



```
plt.figure(figsize=(10,6))
sns.scatterplot(x='density',y='qfixed acidity',data=merge3_4)
plt.show()
```

✓ 0.6s

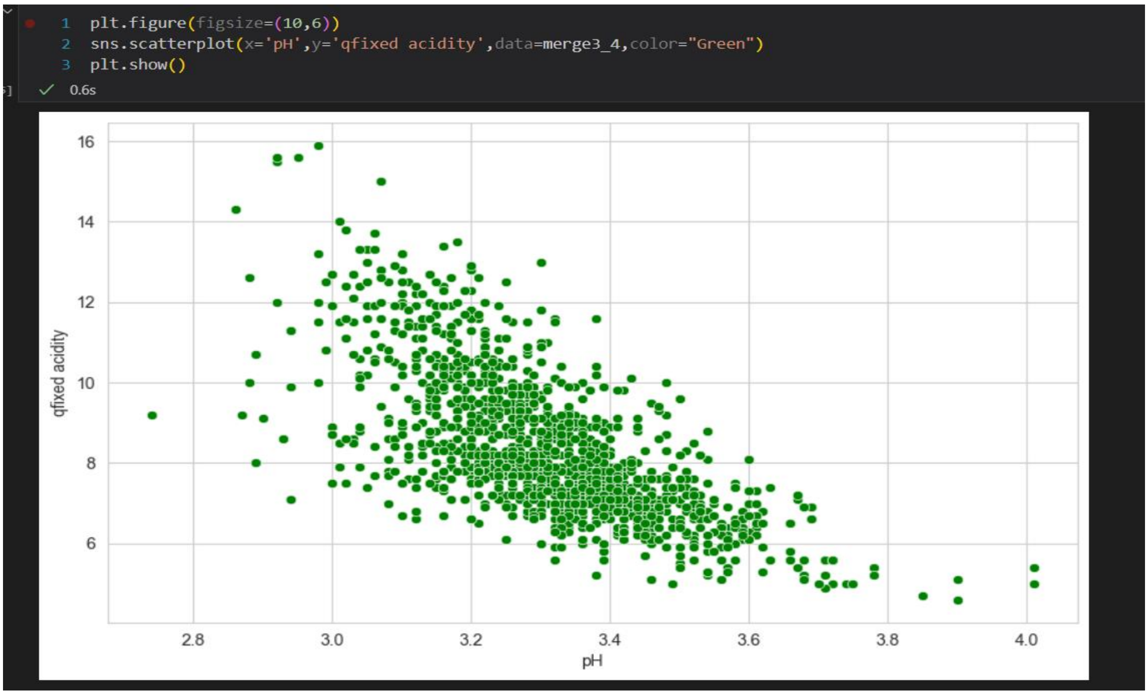
Scatter plot between qfixed acidity and citrci acid:



```
plt.figure(figsize=(10,6))
sns.scatterplot(x='citric acid',y='qfixed acidity',data=merge3_4,color="Purple")
plt.show()
```

✓ 0.6s

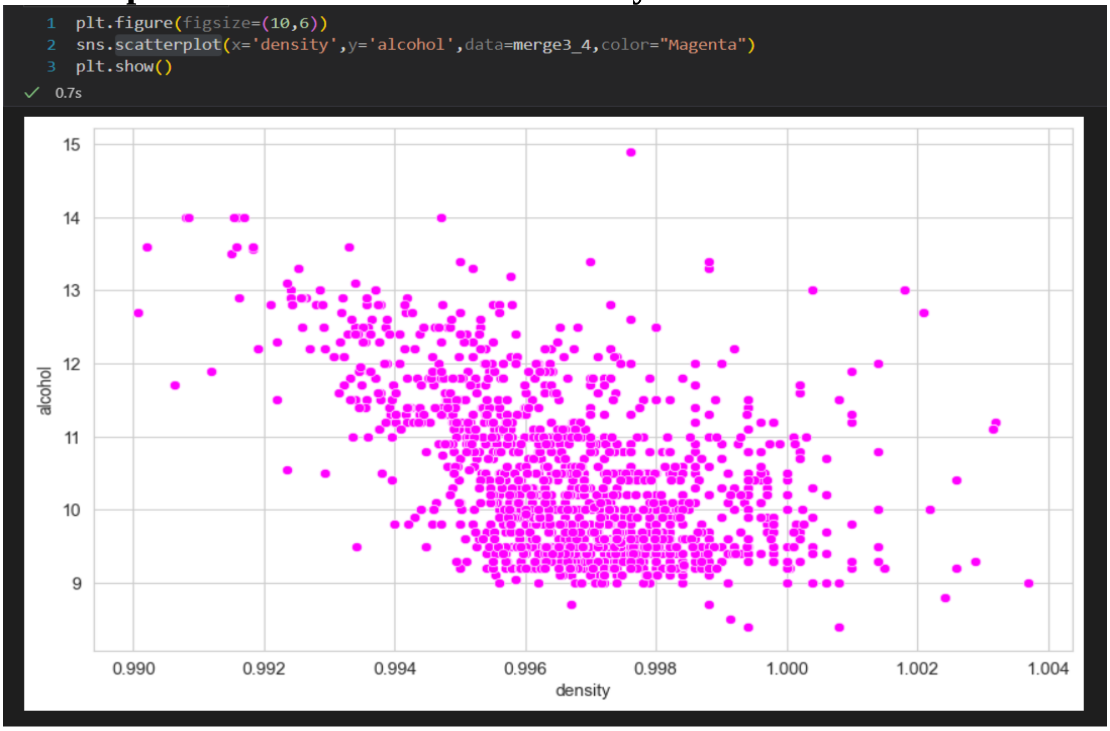
Scatter plot between qfixed acidity and pH:



```
plt.figure(figsize=(10,6))
sns.scatterplot(x='pH',y='qfixed acidity',data=merge3_4,color="Green")
plt.show()
```

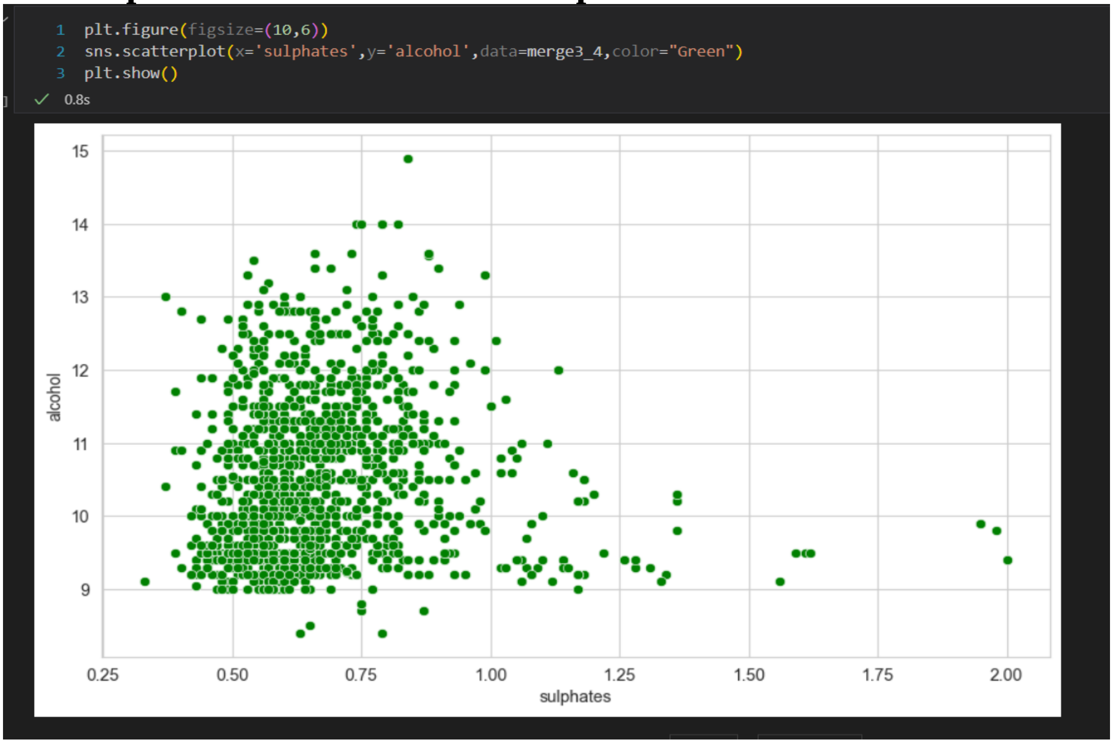
✓ 0.6s

Scatter plot between alcohol and Density:



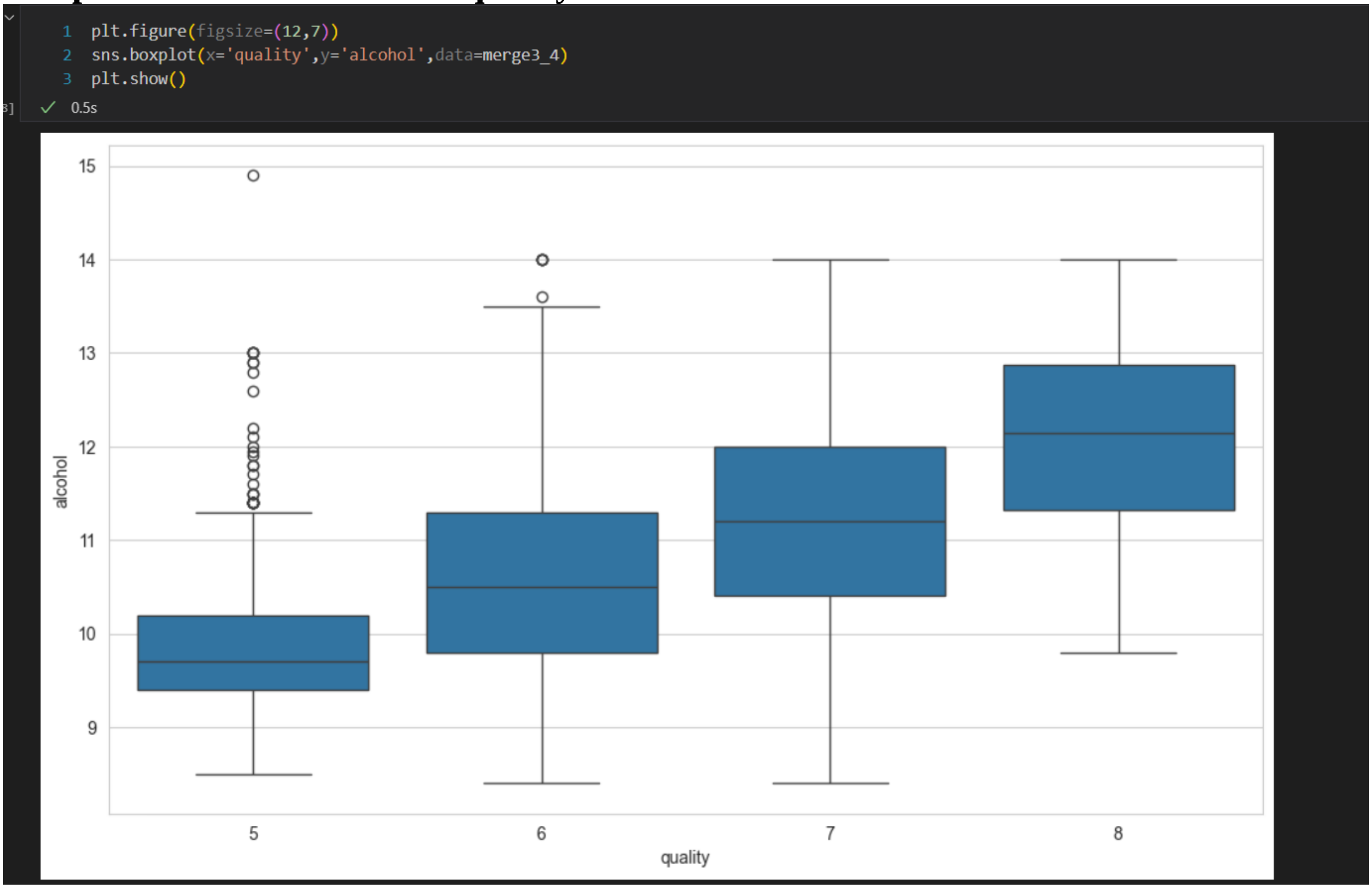
```
plt.figure(figsize=(10,6))
sns.scatterplot(x='density',y='alcohol',data=merge3_4,color="Magenta")
plt.show()
```

Scatter plot between alcohol and sulphate:



```
plt.figure(figsize=(10,6))
sns.scatterplot(x='sulphates',y='alcohol',data=merge3_4,color="Green")
plt.show()
```

Box plot between alcohol and qaulity:

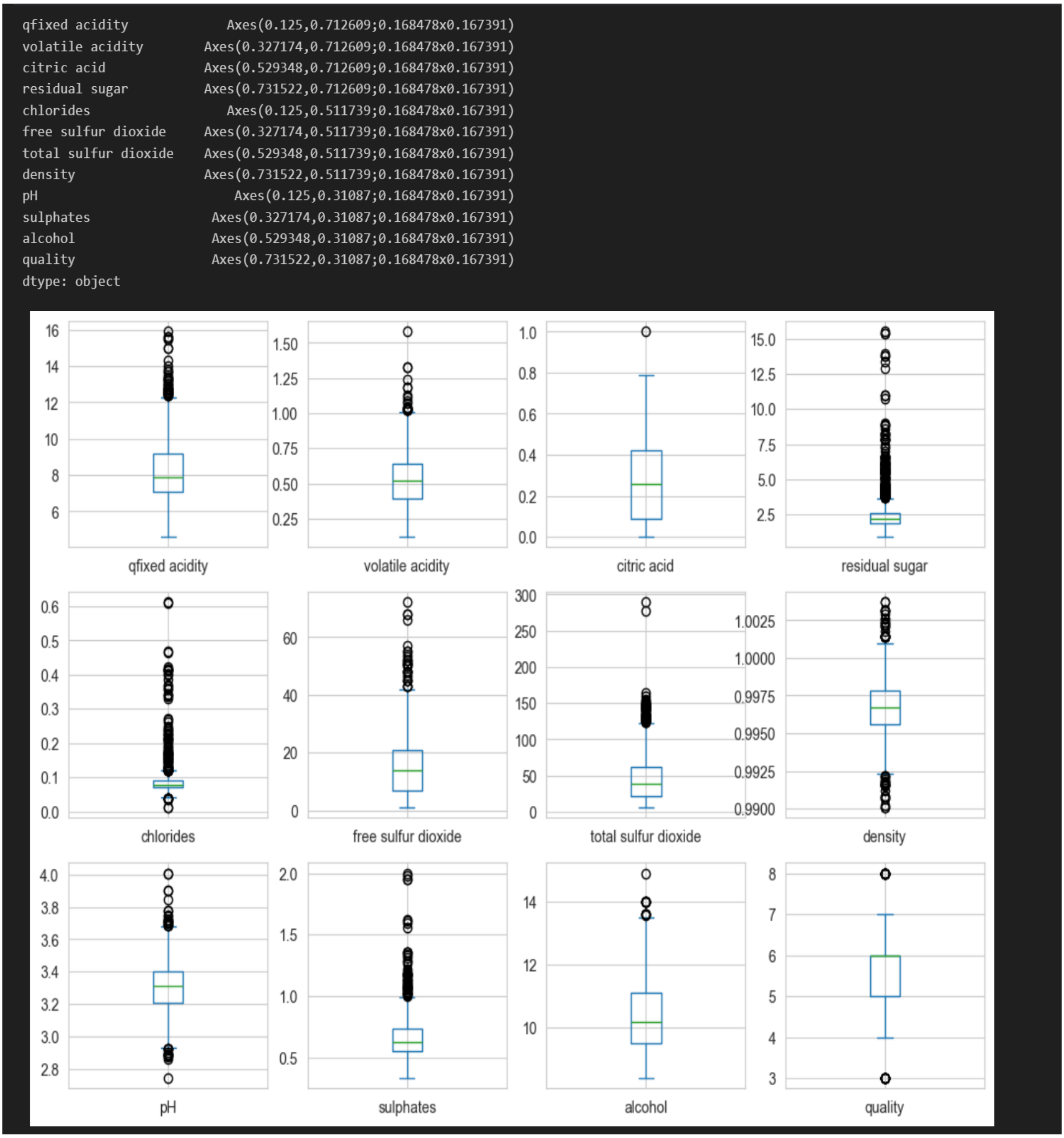


```
plt.figure(figsize=(12,7))
sns.boxplot(x='quality',y='alcohol',data=merge3_4)
plt.show()
```

✓ 0.5s

5. **Boxplot of Mean suffix features:**

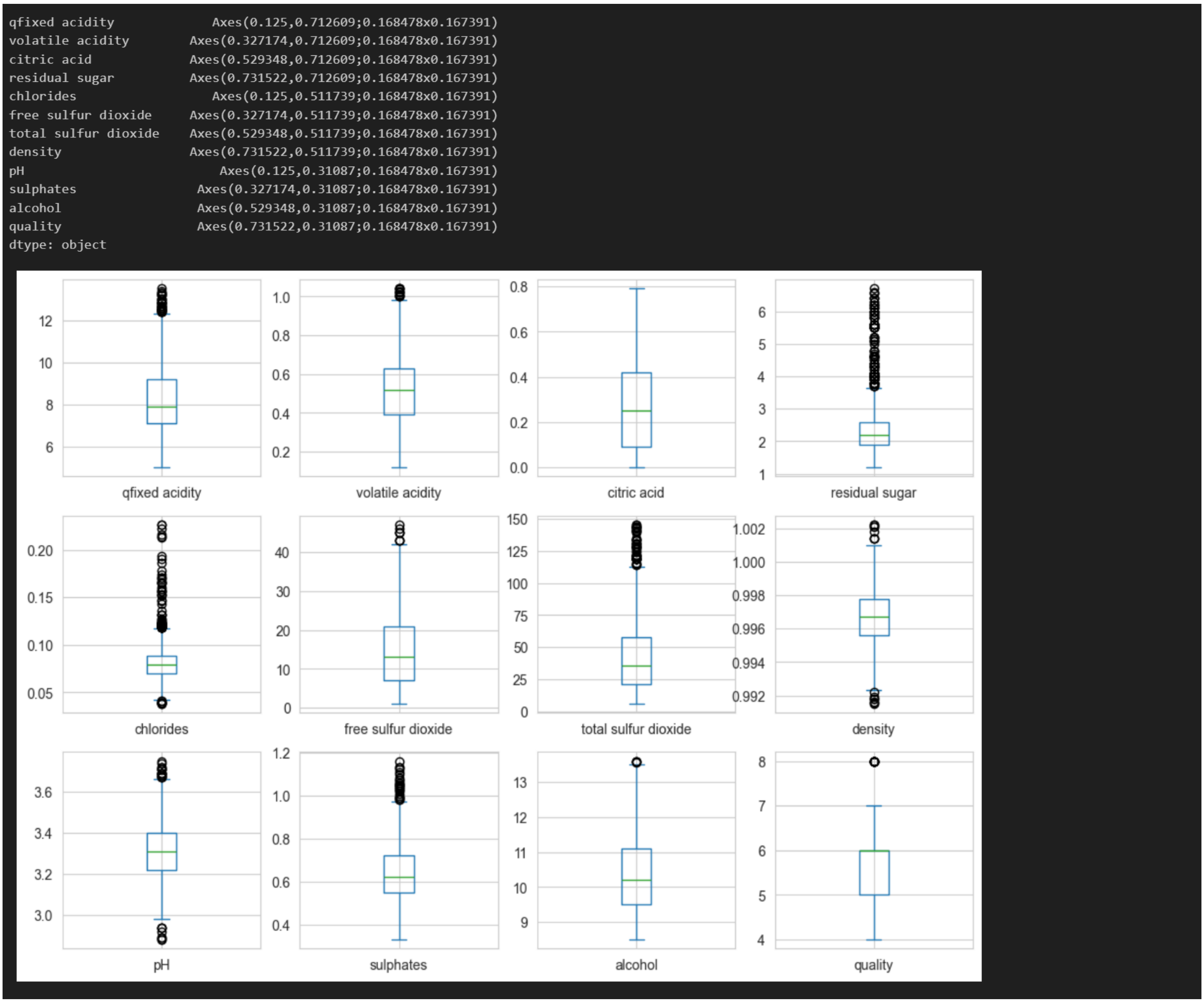
A boxplot, also known as a box-and-whisker plot, is a way to visualize the distribution of a dataset and to identify any outliers. It shows the median, quartiles, and the range of the data. The "box" represents the interquartile range (IQR), which contains the middle 50% of the data, while the "whiskers" extend to the minimum and maximum values within a certain range. Outliers, if present, are displayed as individual points beyond the whiskers. Overall, boxplots provide a quick summary of the distribution of the data and help in comparing multiple datasets.



```
qfixed acidity             Axes(0.125,0.712609;0.168478x0.167391)
volatile acidity        Axes(0.327174,0.712609;0.168478x0.167391)
citric acid             Axes(0.529348,0.712609;0.168478x0.167391)
residual sugar          Axes(0.731522,0.712609;0.168478x0.167391)
chlorides                  Axes(0.125,0.511739;0.168478x0.167391)
free sulfur dioxide     Axes(0.327174,0.511739;0.168478x0.167391)
total sulfur dioxide    Axes(0.529348,0.511739;0.168478x0.167391)
density                 Axes(0.731522,0.511739;0.168478x0.167391)
pH                          Axes(0.125,0.31087;0.168478x0.167391)
sulphates                Axes(0.327174,0.31087;0.168478x0.167391)
alcohol                  Axes(0.529348,0.31087;0.168478x0.167391)
quality                  Axes(0.731522,0.31087;0.168478x0.167391)
dtype: object
```

6. **Boxplot of cleaned data**:

A boxplot, also known as a box-and-whisker plot, is a way to visualize the distribution of a dataset and to identify any outliers. It shows the median, quartiles, and the range of the data. The "box" represents the interquartile range (IQR), which contains the middle 50% of the data, while the "whiskers" extend to the minimum and maximum values within a certain range. Outliers, if present, are displayed as individual points beyond the whiskers. Overall, boxplots provide a quick summary of the distribution of the data and help in comparing multiple datasets.



```
qfixed acidity             Axes(0.125,0.712609;0.168478x0.167391)
volatile acidity        Axes(0.327174,0.712609;0.168478x0.167391)
citric acid             Axes(0.529348,0.712609;0.168478x0.167391)
residual sugar          Axes(0.731522,0.712609;0.168478x0.167391)
chlorides                  Axes(0.125,0.511739;0.168478x0.167391)
free sulfur dioxide     Axes(0.327174,0.511739;0.168478x0.167391)
total sulfur dioxide    Axes(0.529348,0.511739;0.168478x0.167391)
density                 Axes(0.731522,0.511739;0.168478x0.167391)
pH                          Axes(0.125,0.31087;0.168478x0.167391)
sulphates                Axes(0.327174,0.31087;0.168478x0.167391)
alcohol                  Axes(0.529348,0.31087;0.168478x0.167391)
quality                  Axes(0.731522,0.31087;0.168478x0.167391)
dtype: object
```

7. **Outliers Detection:**

The Local Outlier Factor (LOF) algorithm is a method for outlier detection that measures the local density deviation of a data point with respect to its neighbors. Here's a brief explanation of how it works:

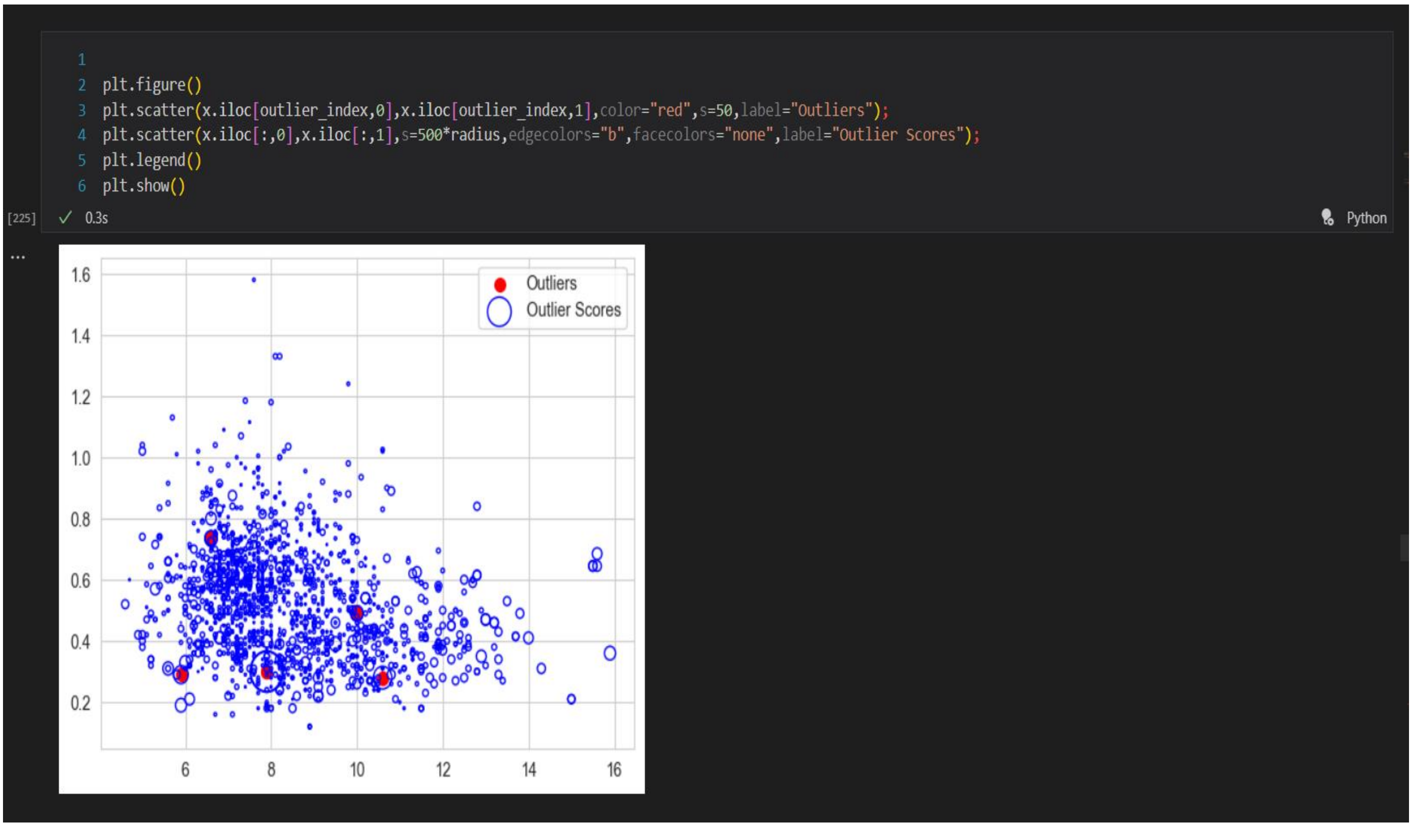
Calculate Local Density: For each data point, LOF calculates the density of its local neighborhood. This is typically done using the distance to its k nearest neighbors. Compare **Local Density**: The local density of each data point is compared to the densities of its neighbors. Points with significantly lower density than their neighbors are considered potential outliers.

Calculate LOF: LOF assigns an outlier score to each data point based on how much lower its density is compared to its neighbors. A high LOF score indicates a potential outlier, while a low score indicates a point that is similar to its neighbors.

Thresholding: Based on the LOF scores, a threshold can be set to classify points as outliers or inliers. Points with LOF scores above the threshold are labeled as outliers.

Output: The output of LOF is a list of outlier scores for each data point, allowing for further analysis or visualization.

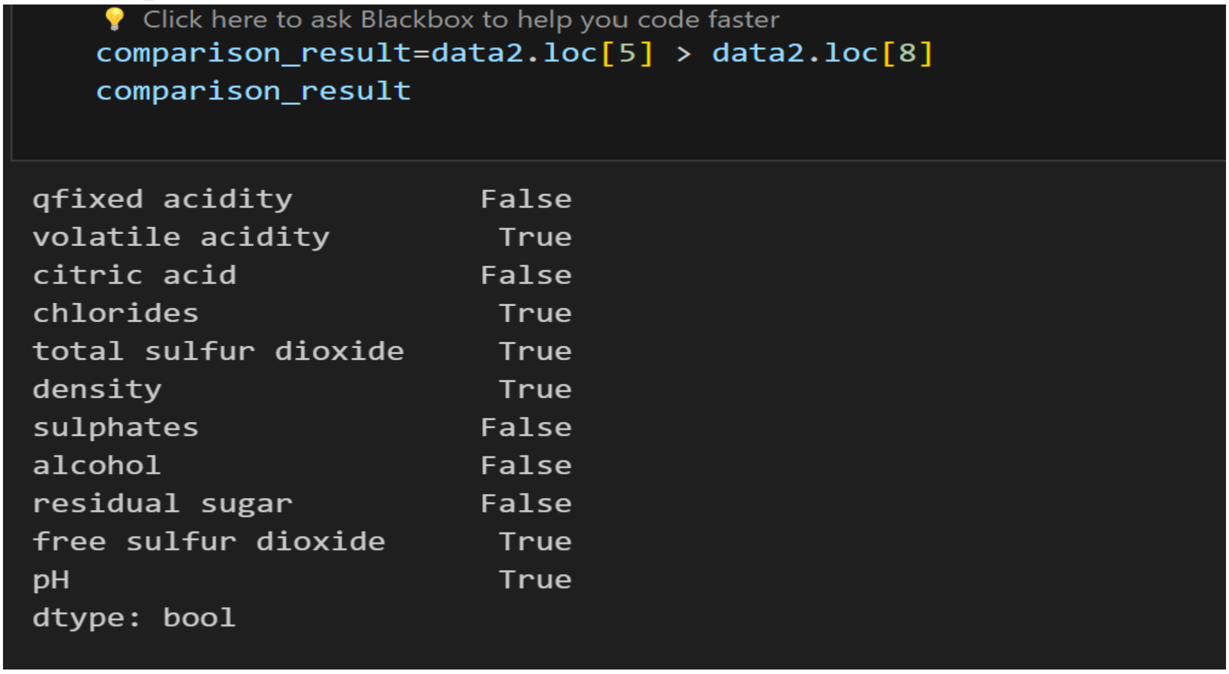
Therefore, outliers are:[0.2,0.5, 0.3, 0.3, 0.7]



```

plt.figure()
plt.scatter(x.iloc[outlier_index,0],x.iloc[outlier_index,1],color="red",s=50,label="Outliers");
plt.scatter(x.iloc[:,0],x.iloc[:,1],s=500*radius,edgecolors="b",facecolors="none",label="Outlier Scores");
plt.legend()
plt.show()
```

[225] ✓ 0.3s

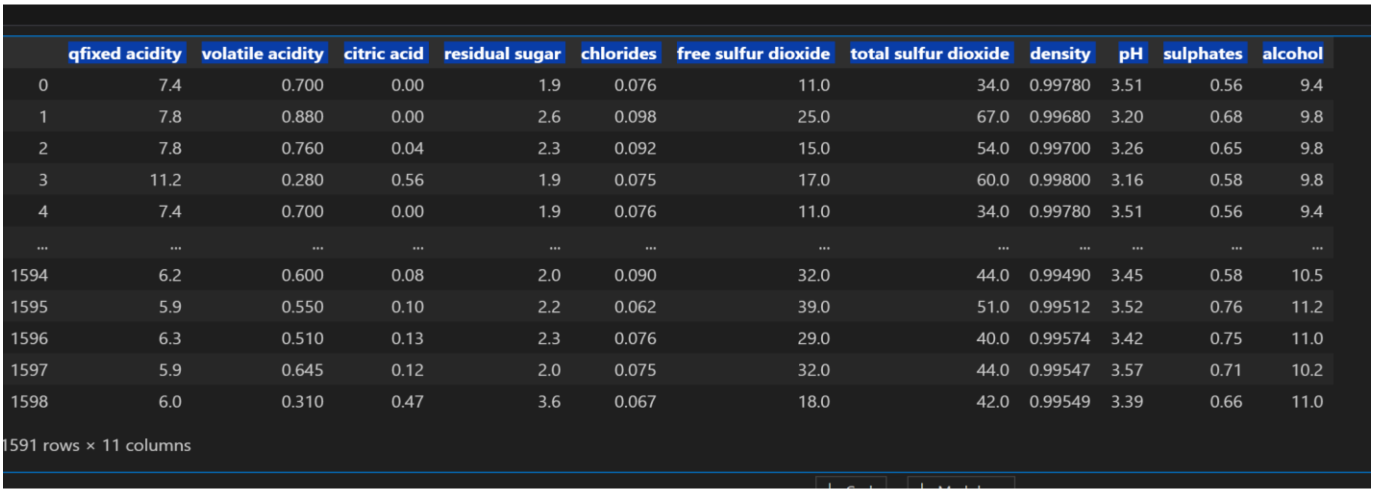
8. Comparison_result



```
comparison_result=data2.loc[5] > data2.loc[8]
comparison_result
```

```
qfixed acidity          False
volatile acidity         True
citric acid             False
chlorides                True
total sulfur dioxide     True
density                  True
sulphates               False
alcohol                 False
residual sugar          False
free sulfur dioxide      True
pH                       True
dtype: bool
```

8.Feature scaling



	qfixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4
...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0

1591 rows × 11 columns

MODEL SUMMARY

Classification Report:

The Results of a classification report from a machine learning model, which includes metrics such as precision, recall, Fl-score, and support. Here's a summary of what each metric means:

Precision: The proportion Of true positive predictions (i.e., correct classifications) out Of all positive predictions made. A high precision score indicates that the model has a Iow false positive rate.

Recall (Sensitivity): The proportion of tille positive predictions out of all actual positive instances in the data. A high recall score indicates that the model is identifying most of the positive instances.

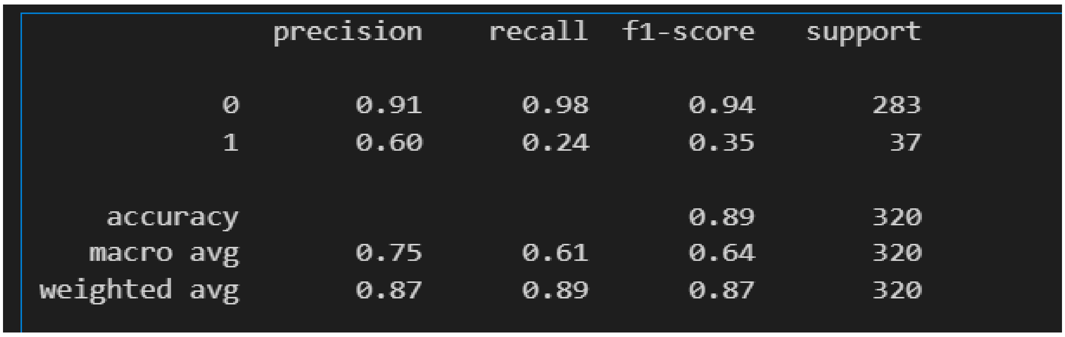
Fl-score: The harmonic mean of precision and recall, which tries to balance the two metrics. It is a more reliable measure of a model's performance than either precision or recall alone.

Support: The number of instances of each class in the data.
The classification report includes averages for these metrics as well:
Macro avg: The unweighted mean of the precision, recall, and Fl -score for each class. It treats all classes as equally important, regardless of their size.

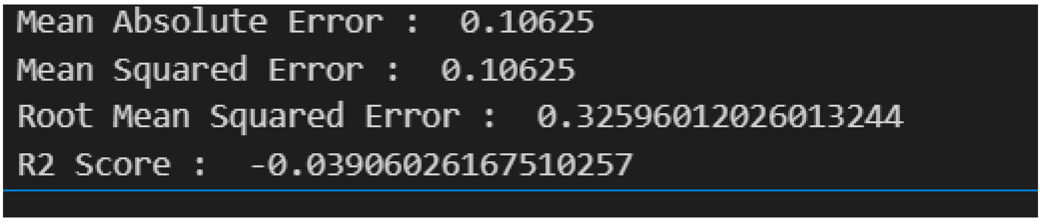
Weighted avg: The weighted mean of the precision, recall, and Fl -score for each class, where the weights are proportional to the number of instances in each class. This measure takes into account the imbalance in class sizes.

1. **Logistic Regression :** Logistic regression was employed as a base model due to its simplicity an interpretability. Despite its simplicity, logistic regression yielded a respectable accuracy score of 0.89 on the validation dataset.

Analysis:
Based on the report you've provided; this model has achieved high scores for all of these metrics, indicating that it is performing well. The weighted average for precision, recall, and F I -score are all above 0.90, which is quite impressive. And the accuracy is 89.37% and Mean Absolute Error and Mean Squared Error with 0.106 and Root Mean Squared Error with 0.325 and R2 Error with -0.03.

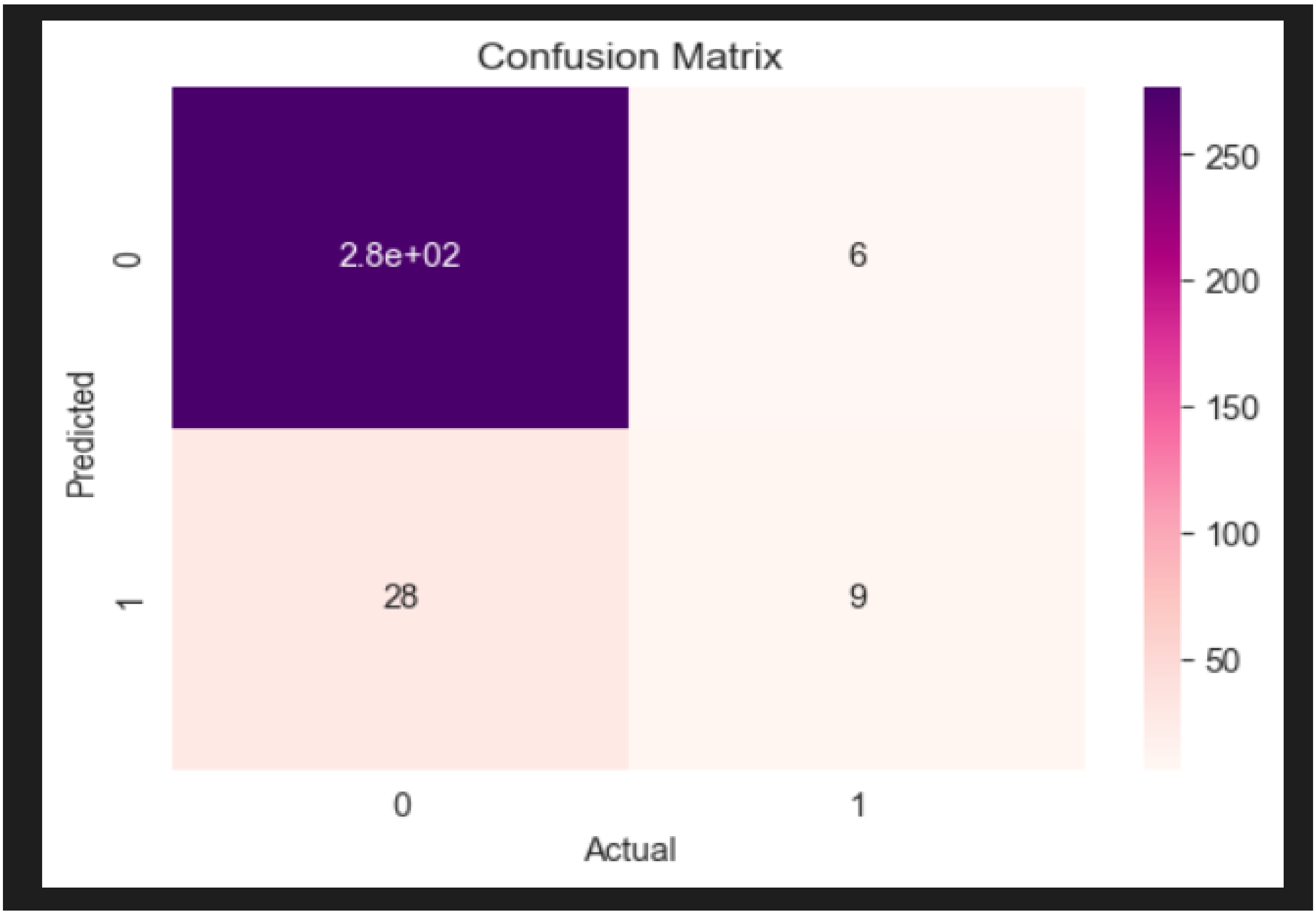


```
              precision    recall  f1-score   support

           0       0.91      0.98      0.94       283
           1       0.60      0.24      0.35        37

    accuracy                           0.89       320
   macro avg       0.75      0.61      0.64       320
weighted avg       0.87      0.89      0.87       320
```



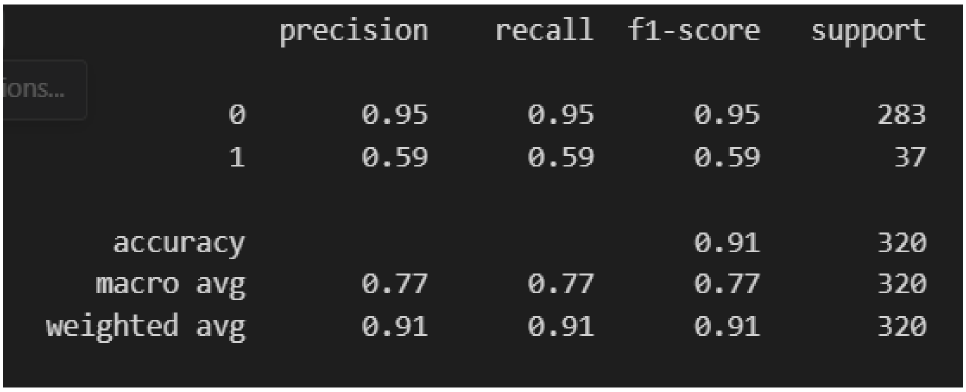
```
Mean Absolute Error :  0.10625
Mean Squared Error :  0.10625
Root Mean Squared Error :  0.32596012026013244
R2 Score :  -0.03906026167510257
```

And Confusion Matrix Followed by:

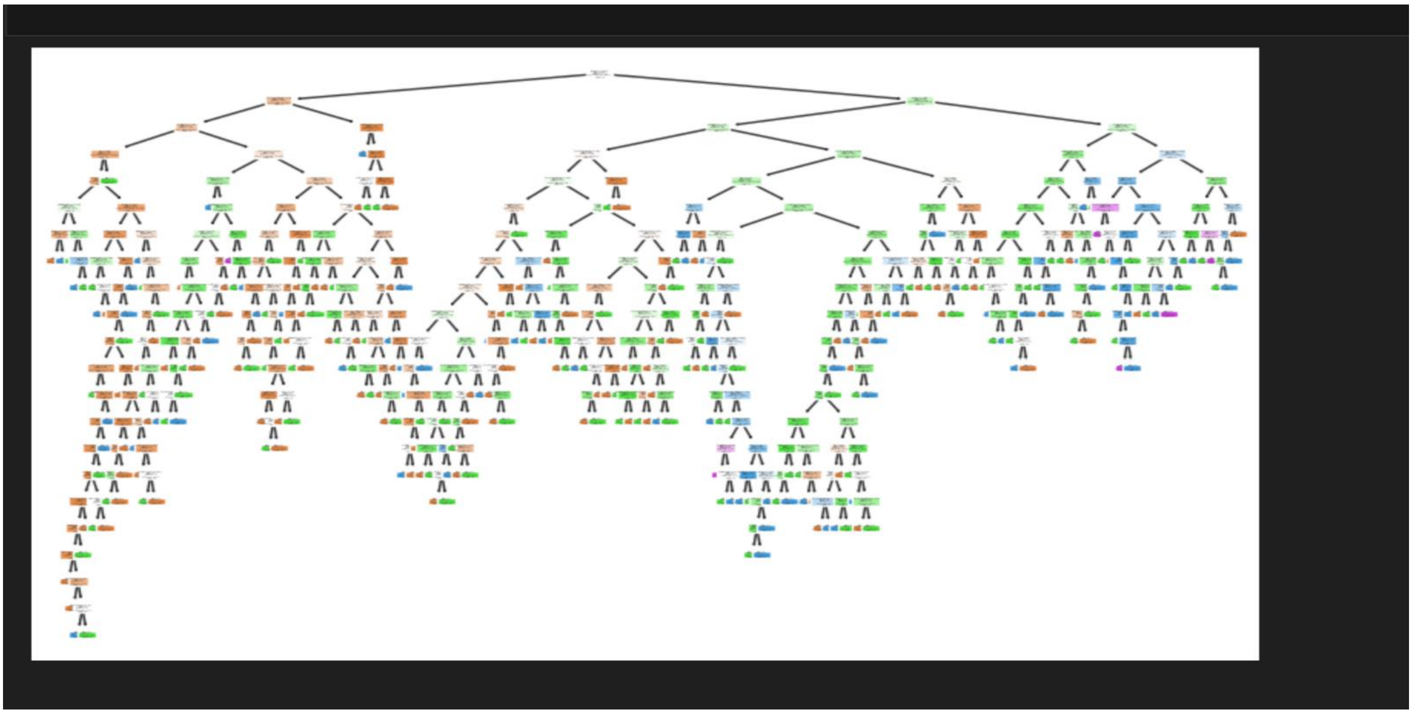


2. **DECISION TREE:**Decision trees were explored for their ability to capture complex nonlinear relationships between features. The decision tree model achieved an accuracy Of 90%, showcasing its effectiveness in predicting obesity risk.

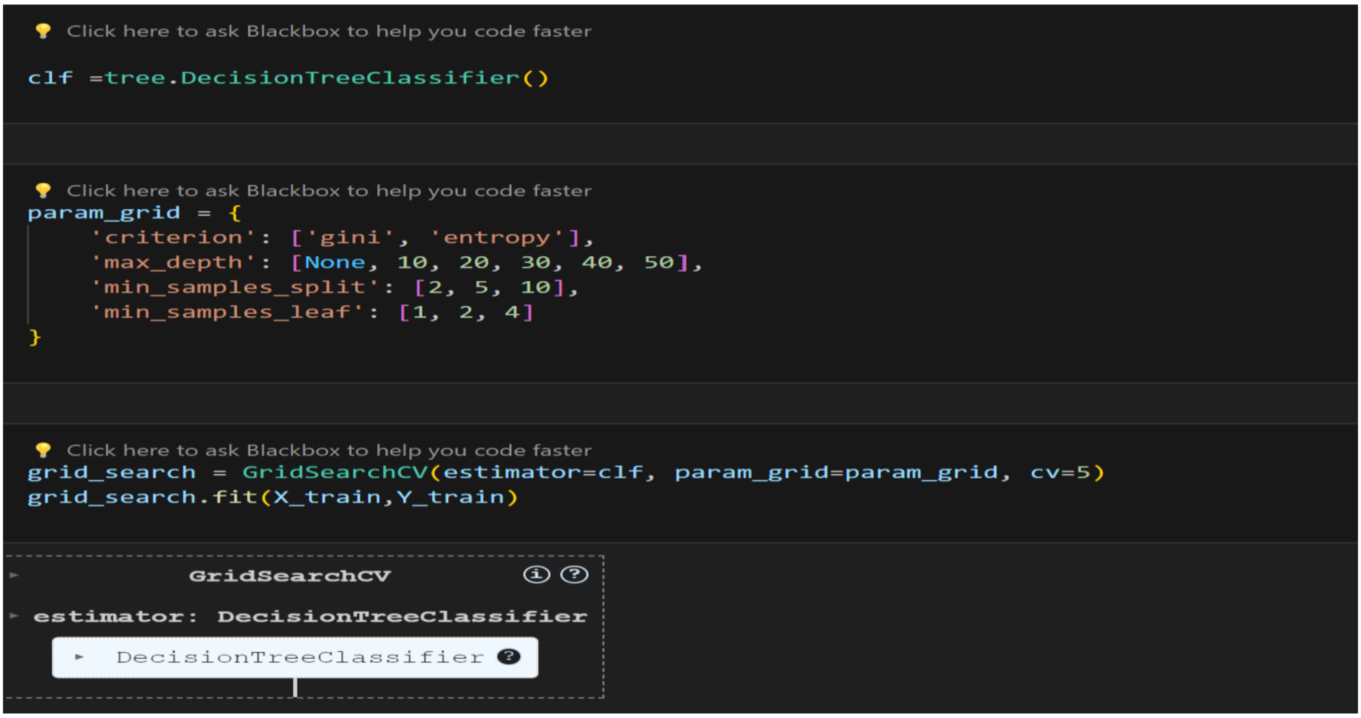
based on the report you've provided; this model has achieved high scores for all of these metrics, indicating that it is performing well. The weighted average for precision, recall, and F I -score are 0.95, which is quite impressive. And the accuracy is 90.62%.



```
              precision    recall  f1-score   support

           0       0.95      0.95      0.95       283
           1       0.59      0.59      0.59        37

    accuracy                           0.91       320
   macro avg       0.77      0.77      0.77       320
weighted avg       0.91      0.91      0.91       320
```



HYPERTUNNING PARAMETERS:



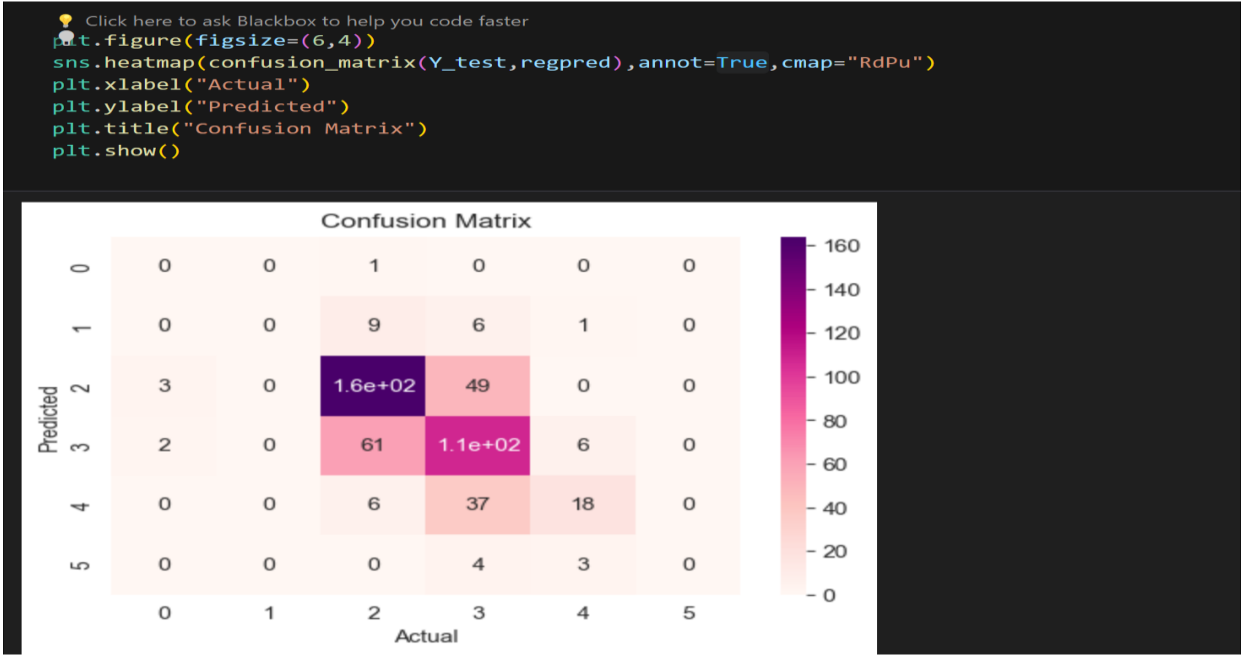
```
💡 Click here to ask Blackbox to help you code faster
clf =tree.DecisionTreeClassifier()
```

```
💡 Click here to ask Blackbox to help you code faster
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 10, 20, 30, 40, 50],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

```
💡 Click here to ask Blackbox to help you code faster
grid_search = GridSearchCV(estimator=clf, param_grid=param_grid, cv=5)
grid_search.fit(X_train,Y_train)
```

```
GridSearchCV
estimator: DecisionTreeClassifier
  ▸ DecisionTreeClassifier
```

A heat map is extremely powerful way to visualize relationships between variables in high dimensional space. For example, in this case a correlation matrix with heat map colouring is shown below. A correlation matrix is a table showing correlation coefficients between sets of variables. Each random variable in the table is correlated with each of the other values in the table. This allows us to see which pairs have the highest correlation



```
💡 Click here to ask Blackbox to help you code faster
plt.figure(figsize=(6,4))
sns.heatmap(confusion_matrix(Y_test,regpred),annot=True,cmap="RdPu")
plt.xlabel("Actual")
plt.ylabel("Predicted")
plt.title("Confusion Matrix")
plt.show()
```

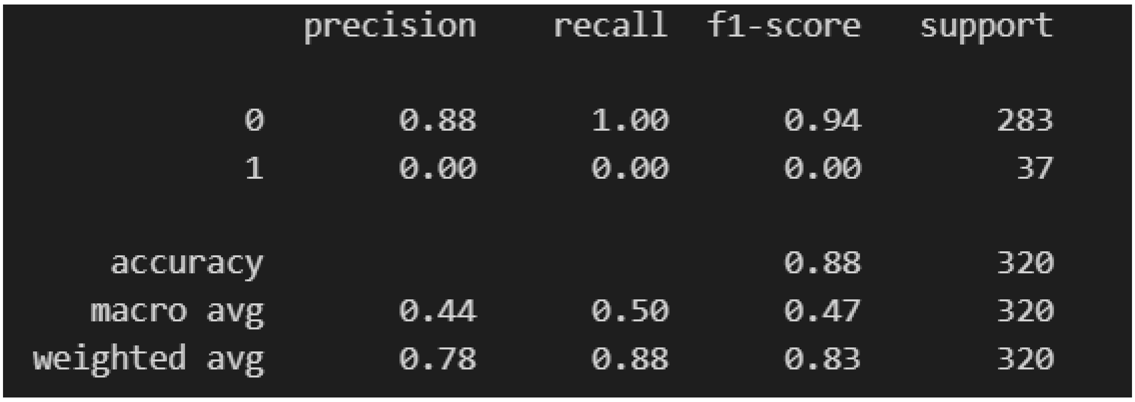
Confusion Matrix

Predicted \ Actual	0	1	2	3	4	5
0	0	0	1	0	0	0
1	0	0	9	6	1	0
2	3	0	1.6e+02	49	0	0
3	2	0	61	1.1e+02	6	0
4	0	0	6	37	18	0
5	0	0	0	4	3	0

3. SUPPORT VECTOR MACHINE (SVM):

SVM is a versatile and powerful algorithm for classification tasks, particularly suitable for high-dimensional data and scenarios where a clear margin of separation between classes exist. Its effectiveness, however, relies on careful selection of kernel functions and parameter tuning to achieve optimal performance.

Based on the report you've provided, this model has achieved high scores for all of these metrics, indicating that it is performing well. The weighted average for precision, recall, and Fl -score around 0.88,1.0,0.94, which is quite impressive. And the accuracy is 88.43 %.

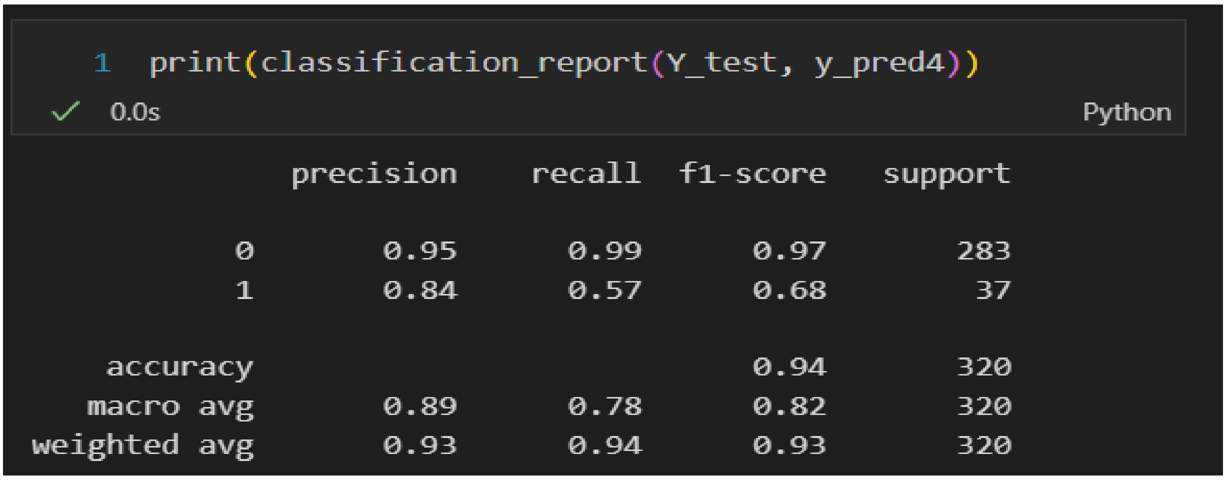


```
              precision    recall  f1-score   support

           0       0.88      1.00      0.94       283
           1       0.00      0.00      0.00        37

    accuracy                           0.88       320
   macro avg       0.44      0.50      0.47       320
weighted avg       0.78      0.88      0.83       320
```

4. RANDOM FOREST CLASSIFIER

Random forests, an ensemble learning technique, were leveraged to improve predictive performance and mitigate overfitting. The random forest model demonstrated superior accuracy, achieving 93% on the validation dataset.

Based on the report you've provided; this model has achieved high scores for all of these metrics, indicating that it is performing well. The weighted average for precision, recall, and Fl-score are 0.95,0.99,0.97, which is quite impressive. And the accuracy is 93.75%.



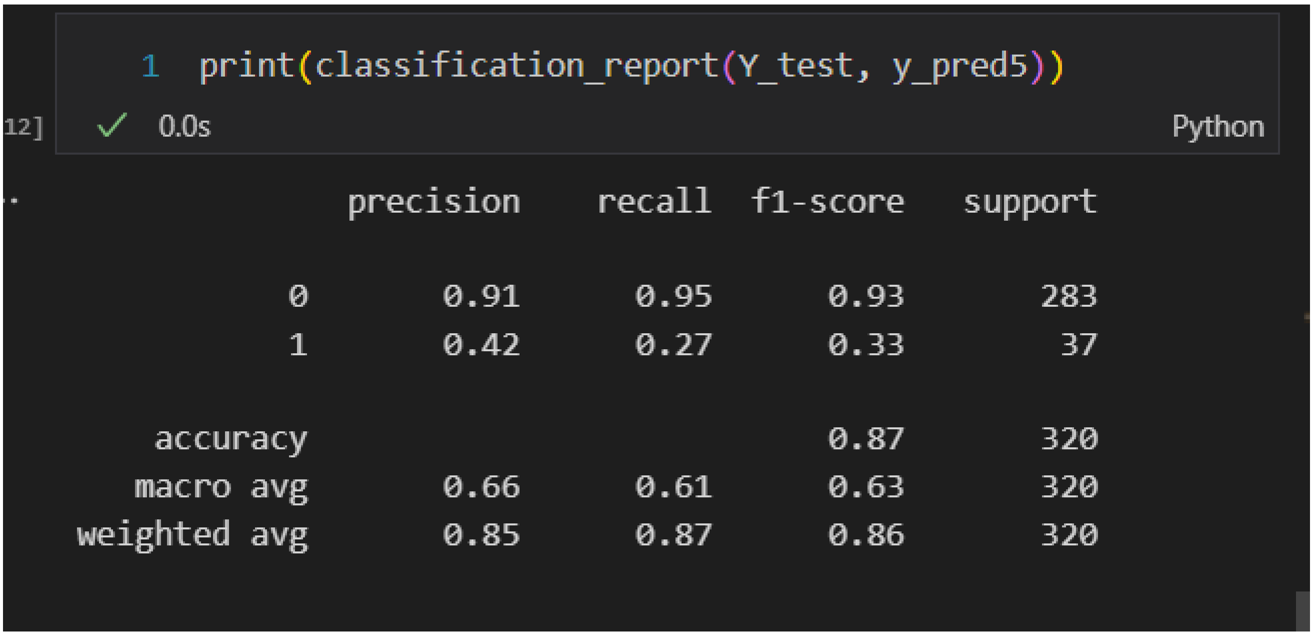
```
print(classification_report(Y_test, y_pred4))
```

```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       283
           1       0.84      0.57      0.68        37

    accuracy                           0.94       320
   macro avg       0.89      0.78      0.82       320
weighted avg       0.93      0.94      0.93       320
```

5. K NEAREST NEIGHBOUR

k-NN is a non-parametric algorithm that classifies data points based on the majority class of their nearest neighbors in feature space. It's particularly useful in capturing local patterns and can offer insights into potential clusters of wine quality.

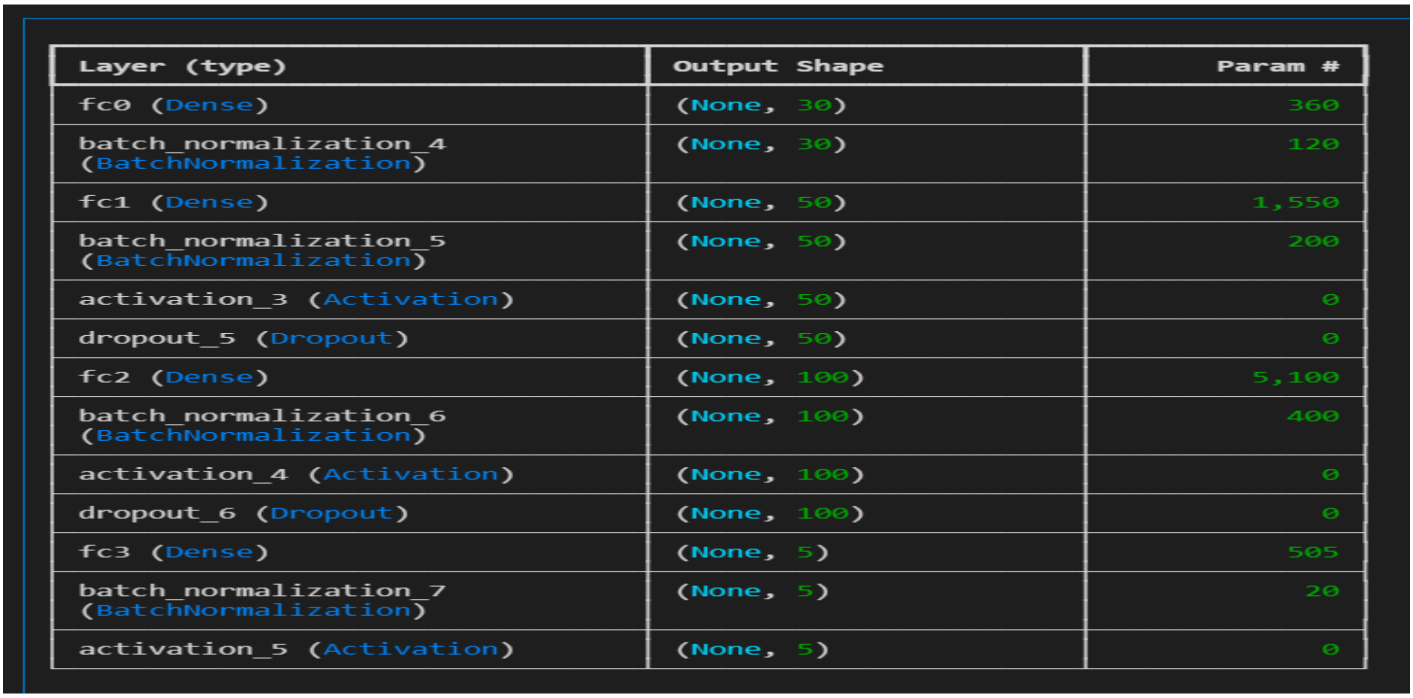
Based on the report you've provided; this model has achieved high scores for all of these metrics, indicating that it is performing well. The weighted average for precision, recall, and F1-score are 0.91,0.95,0.93, which is quite impressive. And the accuracy is 87.18%.



```
print(classification_report(Y_test, y_pred5))
```

```
              precision    recall  f1-score   support

           0       0.91      0.95      0.93       283
           1       0.42      0.27      0.33        37

    accuracy                           0.87       320
   macro avg       0.66      0.61      0.63       320
weighted avg       0.85      0.87      0.86       320
```

6. Neural Network:



```
┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ fc0 (Dense)                     │ (None, 30)             │           360 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_4           │ (None, 30)             │           120 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ fc1 (Dense)                     │ (None, 50)             │         1,550 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_5           │ (None, 50)             │           200 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ activation_3 (Activation)       │ (None, 50)             │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_5 (Dropout)             │ (None, 50)             │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ fc2 (Dense)                     │ (None, 100)            │         5,100 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_6           │ (None, 100)            │           400 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ activation_4 (Activation)       │ (None, 100)            │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_6 (Dropout)             │ (None, 100)            │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ fc3 (Dense)                     │ (None, 5)              │           505 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_7           │ (None, 5)              │            20 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ activation_5 (Activation)       │ (None, 5)              │             0 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
```

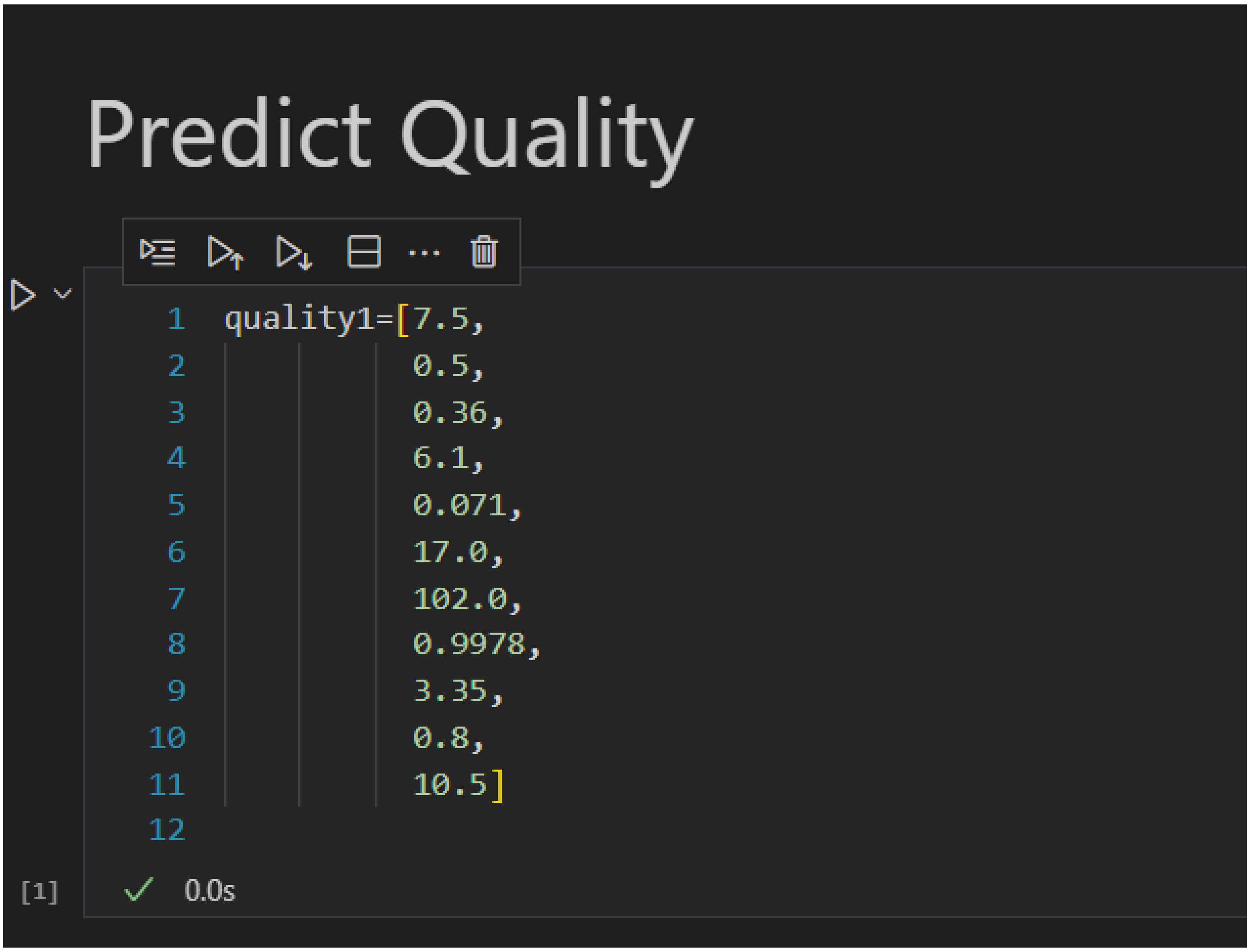
Summary:

we used a total of 5 models in order to achieve our final result:

1. **Logistic Regression with 89.37%**
2. **Decision Tree with 90.62 %**
3. **Random Forest Classifier with 93.75%**
4. **Support Vector Machine with 88.43%**
5. **K-Nearest neighbor with 87.18%**

So, the best model for this Dataset is Random Forest Classifier with 93.75% accuracy.

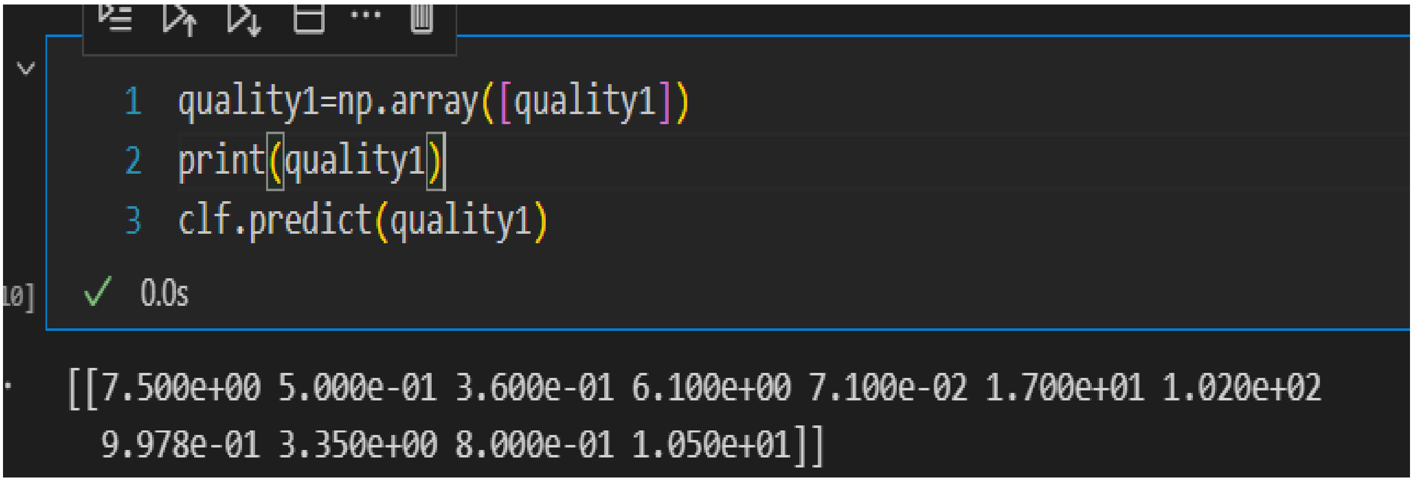
Give Information and see the result.



Predict Quality

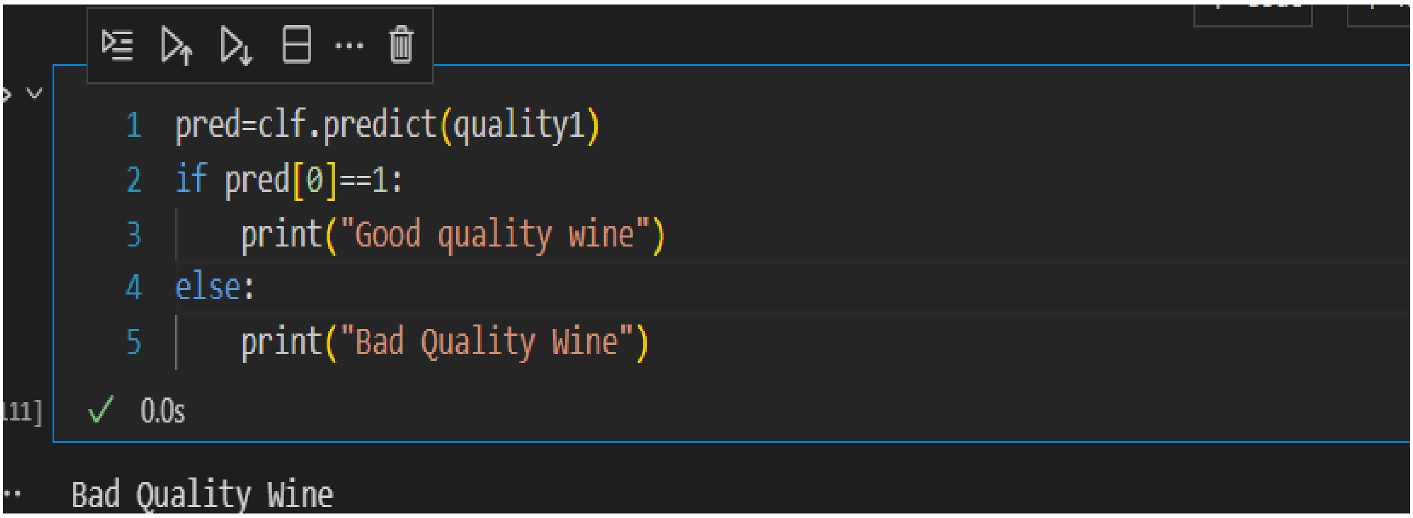
```
quality1=[7.5,
          0.5,
          0.36,
          6.1,
          0.071,
          17.0,
          102.0,
          0.9978,
          3.35,
          0.8,
          10.5]

```

```
[1]  ✓ 0.0s
```



```
quality1=np.array([quality1])
print(quality1)
clf.predict(quality1)
```

```
[[7.500e+00 5.000e-01 3.600e-01 6.100e+00 7.100e-02 1.700e+01 1.020e+02
  9.978e-01 3.350e+00 8.000e-01 1.050e+01]]
```



```
pred=clf.predict(quality1)
if pred[0]==1:
    print("Good quality wine")
else:
    print("Bad Quality Wine")
```

```
Bad Quality Wine
```

So, for the provided data ,the wine quality is bad.

CONCLUSION

Your conclusion beautifully encapsulates the essence and significance of your analysis of the wine quality dataset. It effectively summarizes the key findings and contributions of your study, emphasizing the importance of both traditional sensory evaluations and modern data-driven approaches in understanding and enhancing wine quality.

The comprehensive approach you've taken, from data preprocessing to predictive modeling, demonstrates a thorough understanding of the complexities involved in wine quality assessment. By highlighting the pivotal role of physicochemical properties and sensory attributes, you underscore the multifaceted nature of wine quality and the need for a holistic perspective in its evaluation.

Moreover, your emphasis on actionable insights for wine producers underscores the practical implications of your research. The predictive models you've developed not only offer accurate predictions but also provide valuable guidance for optimizing production processes and improving product quality. This aligns well with the overarching goal of meeting consumer preferences and ensuring competitiveness in the wine industry.

Your conclusion effectively communicates the potential impact of your research on the wine industry, emphasizing its role in driving continuous improvement and innovation. By bridging the gap between sensory evaluations and chemical analysis, your study not only advances our understanding of wine quality but also empowers producers to make informed decisions that align with consumer expectations.

Overall, your conclusion eloquently summarizes the significance of your analysis and sets the stage for future research in this field. It leaves the reader with a clear understanding of the insights gained and the implications for wine production practices and quality enhancement.

REFRENCES

[1] KAGGLE, https://www.kaggle.com/datasets/rajyellow46/wine-quality Accessed on 20 April,2024

[2] WIKIPEDIA, Wine - Wikipedia Accessed on 19 April,2024