INFLUENTIAL FACTORS ON THE QUALITY OF RED WINE

Satya Sai Venkata Sri Harsha, Yalla

WSU Id: M999M297

Introduction:

The red wine industry has experienced significant growth recently, primarily due to the increasing popularity of social drinking. Companies operating in this industry are now leveraging product quality certifications to market their products. However, this certification process is time-consuming, expensive, and requires the expertise of human specialists. Furthermore, the subjective nature of wine appreciation by tasters, whose opinions can vary widely, plays a pivotal role in determining the price of red wine. A crucial component of certifying and evaluating red wine quality involves conducting laboratory-based photochemical tests that analyze factors such as acidity, pH, sugar content, and other chemical properties. If it were possible to establish a clear connection between these chemical properties and human perceptions of quality, it could lead to more stringent regulation of quality assurance, certification, and assessment within the red wine market. The primary objective of this study is to identify which characteristics serve as the most significant predictors of red wine quality and provide a comprehensive analysis of each of these attributes in relation to our model's perception of red wine quality. To achieve this, machine learning algorithms such as SVM, random forest, and decision trees will be employed to make predictions about future wine performance.

Dataset:

The dataset contains a total of 12 features. 11 predictor variables and 1 target variable.

Feature	Type
Fixed Acidity	Numerical
Volatile Acidity	Numerical
Citric Acid	Numerical
Residual Sugar	Numerical
Chlorides	Numerical
Free Sulfur Dioxide	Numerical
Total Sulfur Dioxide	Numerical
Density	Numerical
pH	Numerical
Sulphates	Numerical
Alcohol	Numerical
Quality	Categorical

The data set contains 12 columns and 1599 rows.

Dataset Link:

https://www.kaggle.com/code/vishalyo990/prediction-of-quality-of-wine/input

Analysis:

No null values were found in any of the column from the data set as shown in figure 1.

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Figure 1: Null Values

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

Figure 2: Descriptive Statistics

Figure 2 displays a statistical summary of a wine quality dataset, which seems to be for red wines given the common attributes analyzed. The summary includes the count, mean, standard deviation (std), minimum (min), 25th percentile (25%), 50th percentile (median, 50%), 75th percentile (75%), and maximum (max) values for several physicochemical variables such as fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. These statistics are essential for understanding the distribution, central tendency, and variability of each feature.

The count being uniform across all features suggests no missing values, and the wide range of values for certain variables like free sulfur dioxide and total sulfur dioxide indicates significant variability, which could influence the sensory qualities and preservation of the wine.

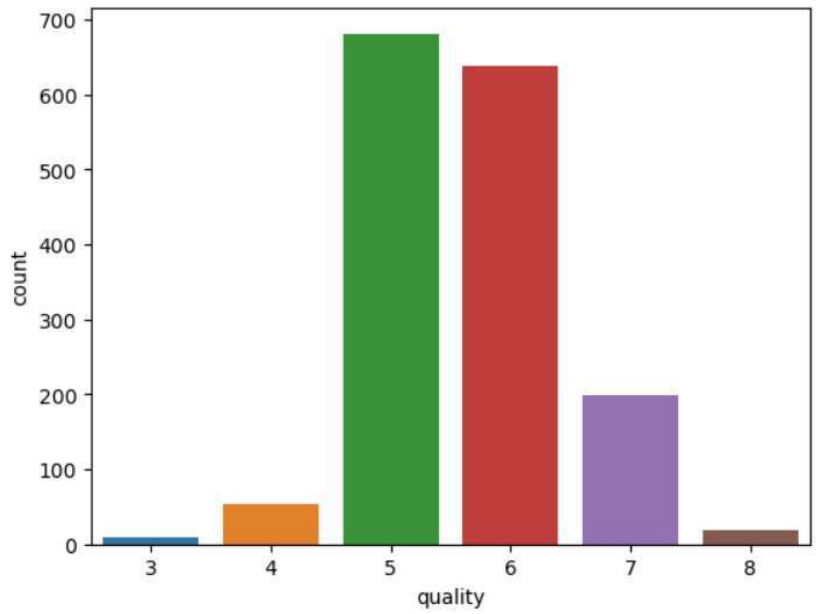
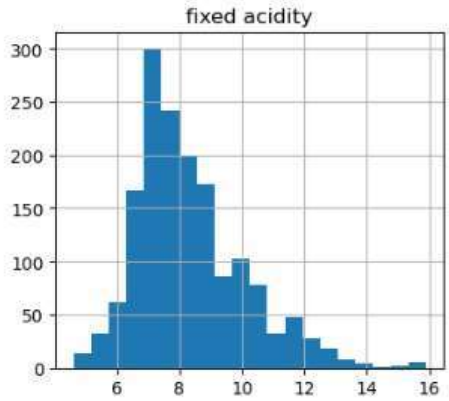


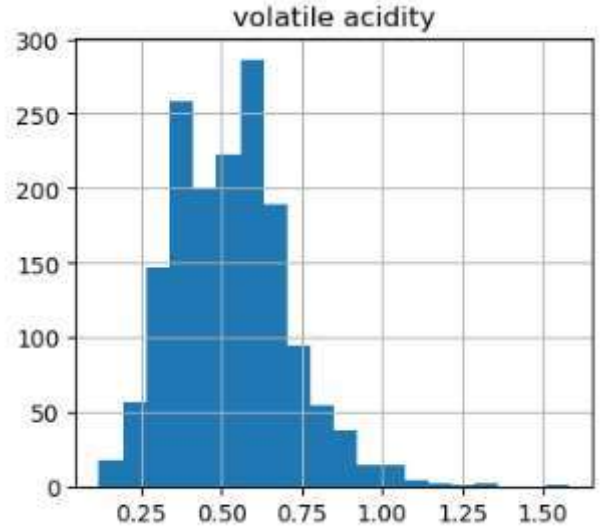
Figure 3: Target Variable Classification

Figure 3 shows the count of each value quality attribute with their frequency in data set. It was found that quality value "5" has occurred the greatest number of times in dataset, quality value "6" has occurred second the greatest number of times in dataset and quality value "3" has occurred least number of times in dataset.

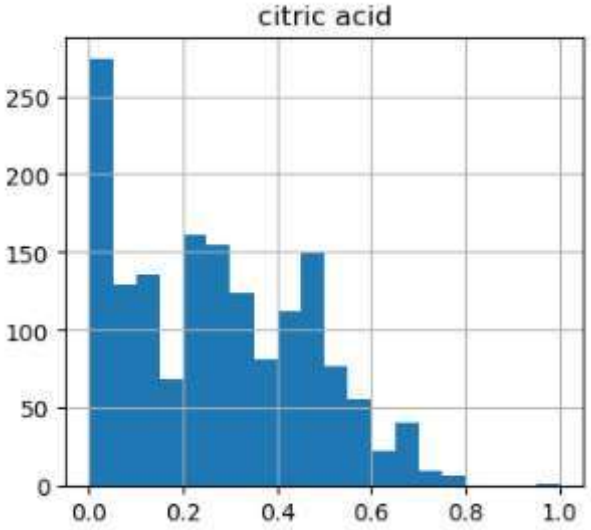
Predictor Variables Analysis:



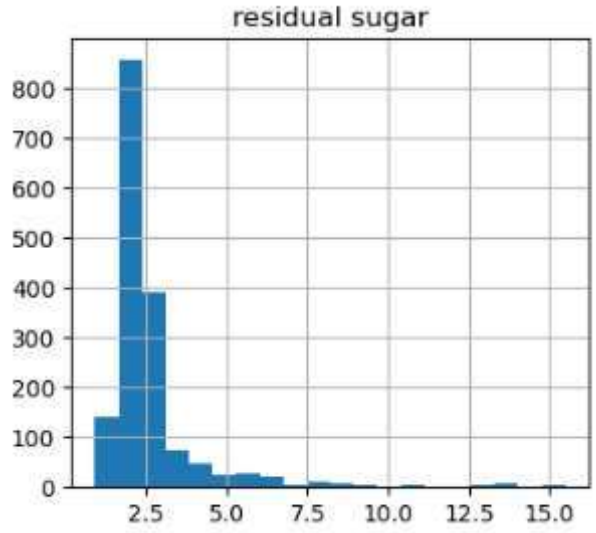
Fixed Acidity: Most wines have a fixed acidity between 6 to 10 g/dm^3, with a peak around 7.5 g/dm^3.



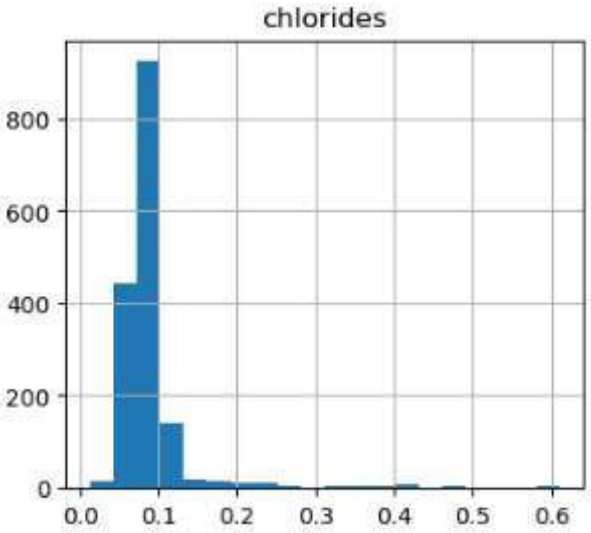
Volatile Acidity: The volatile acidity predominantly ranges from 0.4 to 0.8 g/dm³, with a high concentration around 0.6 g/dm³.



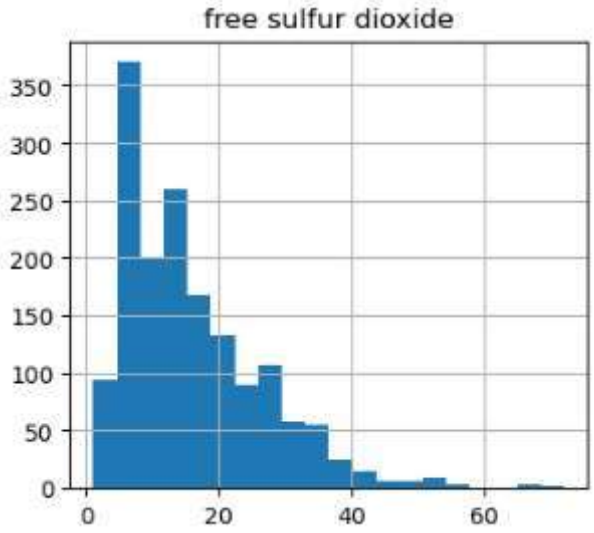
Citric Acid: A significant number of wines have a low citric acid content, with a peak near 0 g/dm³, though there's a spread up to 0.5 g/dm³.



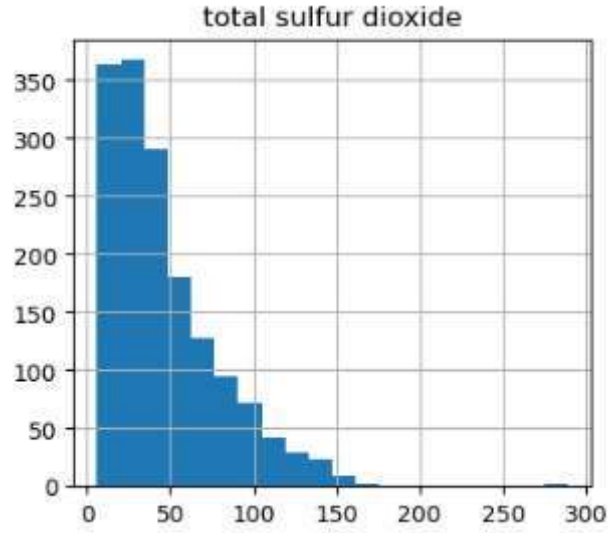
Residual Sugar: The residual sugar is mostly under 2.5 g/dm³, indicating that most of the wines are dry.



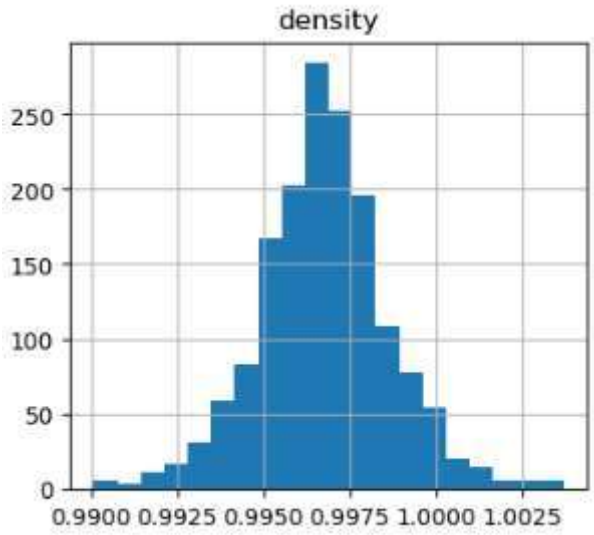
Chlorides: Chloride levels are generally low, primarily concentrated below 0.1 g/dm³.



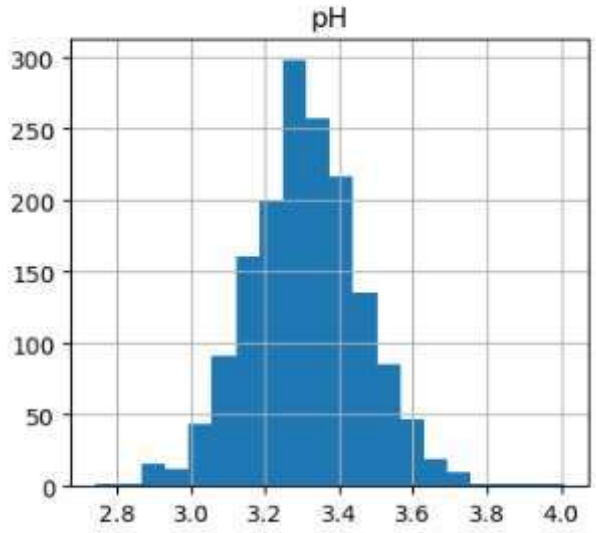
Free Sulfur Dioxide: This feature shows a wide range, with a concentration of values in the lower end, especially around 15 mg/dm³.



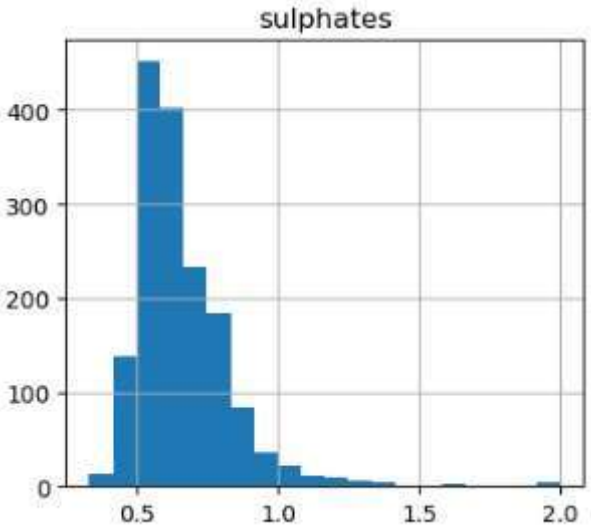
Total Sulfur Dioxide: Most wines have total sulfur dioxide levels below 100 mg/dm³, with a peak around 40 mg/dm³.



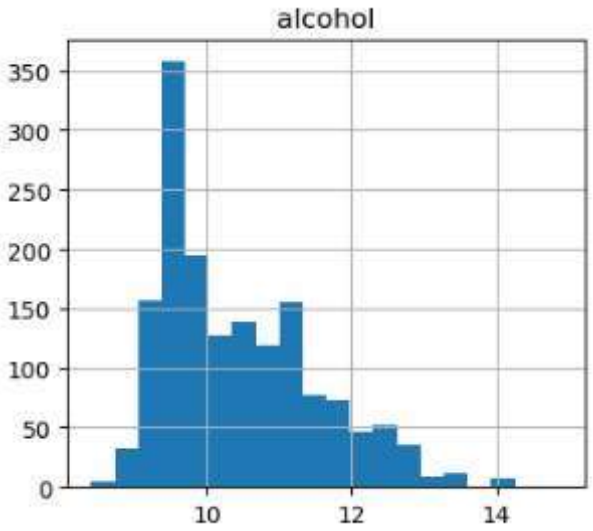
Density: The density of most wines is close to 0.9965 to 0.998 g/cm³, with a slightly right-skewed distribution.



pH: The pH values are mostly concentrated around 3.2 to 3.4, indicating a slightly acidic nature.



Sulphates: Sulphate levels mostly range between 0.5 to 0.75 g/dm³, with a peak around 0.6 g/dm³.



Alcohol: Alcohol content shows a wide distribution, mostly ranging from 9% to 13%, with a concentration around 10%.

Correlation:

Correlation is a statistical measure that describes the degree to which two variables are related or associated. It quantifies the strength and direction of the relationship between two variables. A positive correlation means that as one variable increases, the other tends to increase as well, while a negative correlation indicates that as one variable increases, the other tends to decrease. A correlation value of 1 represents a perfect positive correlation, -1 represents a perfect negative correlation, and 0 represents no correlation at all. Correlation is often used to analyze and understand the relationship between variables in data analysis and research.

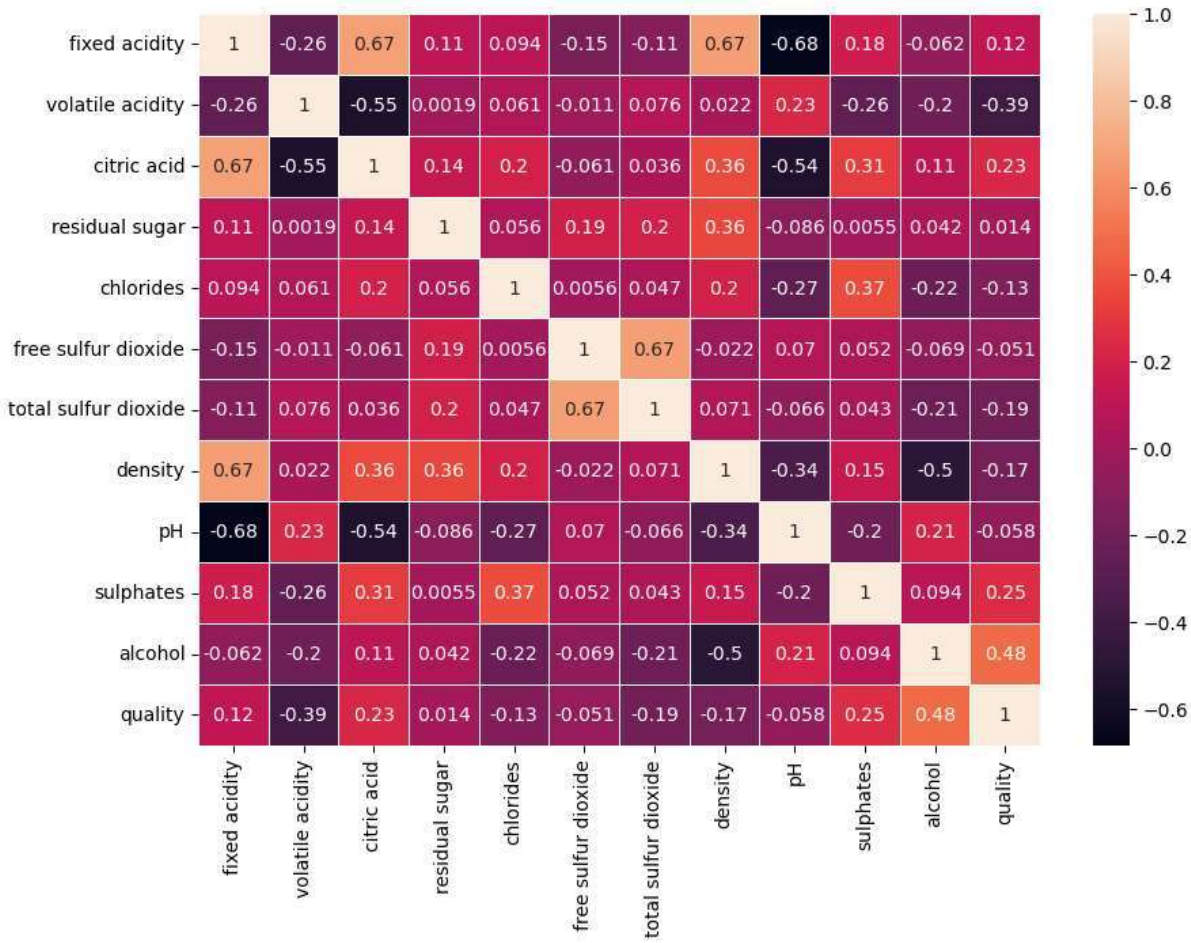


Figure 2: Correlation Matrix as Heatmap

The above correlation matrix shows that the following three columns have a strong positive correlation with quality attributes.

- alcohol
- sulphates
- citric acid

The correlation matrix from Figure 4 shows that the following three columns have a strong negative correlation with quality attributes.

- volatile acidity
- total sulfur dioxide
- density

Data Modeling:

Data modelling is the technique of utilizing words and symbols to describe the data and how it flows to create a streamlined picture of a software system and the data pieces it includes. Data models serve as a guide when creating new databases or redesigning older software.

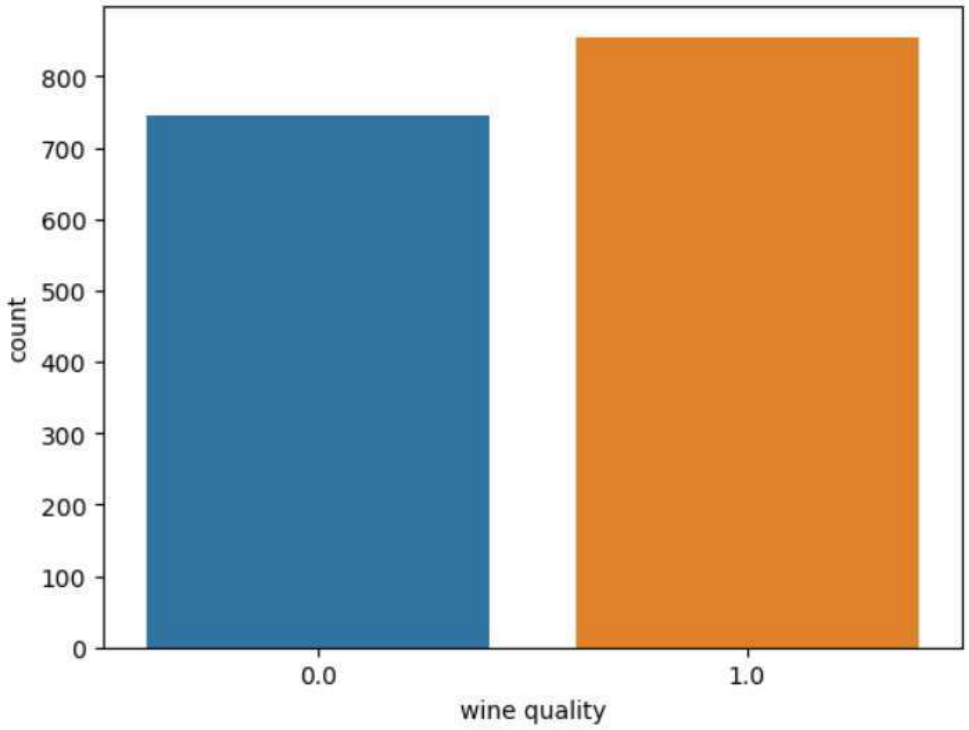


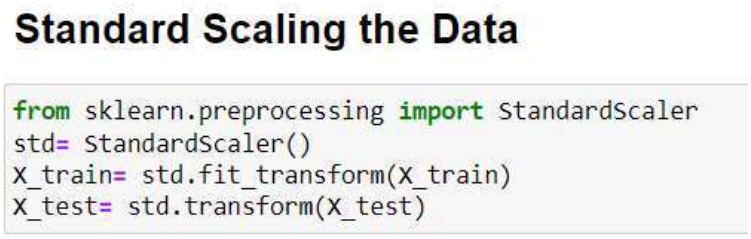
Figure 5: Classification of Wine Quality

Figure 5 is a bar chart depicting the frequency of two categories of wine quality Good and Not Good.

A blue bar corresponding to the category labeled "0.0", which could suggest a classification of lower quality wine. An orange bar corresponding to the category labeled "1.0", possibly indicating a higher quality wine classification.

The height of each bar reflects the number of wines that fall into each quality category.

Standard Scalar:



Standard Scaler is a data preprocessing technique commonly used in machine learning and statistics to scale or standardize the features of a dataset. It is particularly useful when working with numerical data with features that have different scales or units. The main purpose of Standard Scaler is to transform the data so that it has a mean of 0 and a standard deviation of 1.

Here's how Standard Scaler works:

1. Calculate the mean (average) value for each feature in the dataset.

2. Calculate the standard deviation for each feature.

3. For each data point in the dataset, subtract the mean of the feature and then divide by the standard deviation of the feature.

The result of applying Standard Scaler is that each feature will have a similar scale and a mean of 0, making it easier for machine learning algorithms to work with the data. This can improve the performance of certain algorithms, especially those that are sensitive to the scale of the input features, such as gradient descent-based optimization methods.

Standard Scaler is often a crucial step in the data preprocessing pipeline when working with machine learning models, and it helps ensure that the features contribute equally to the model's performance without any undue influence due to differences in their scales.

Methods and Results:

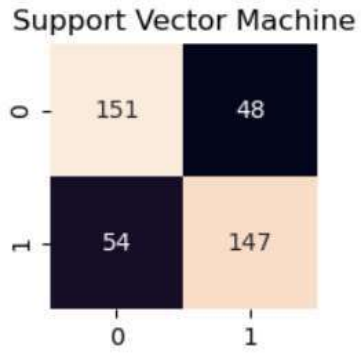
1. Support Vector Machine: Support Vector Machines (SVM) are a versatile and powerful class of supervised machine learning algorithms used for both classification and regression tasks. SVM aims to find a hyperplane in a high-dimensional feature space that best separates data points into distinct classes or predicts a continuous target variable. It accomplishes this by maximizing the margin, which is the distance between the hyperplane and the nearest data points of each class, hence providing robust generalization to unseen data. SVM can handle linear and nonlinear relationships through different kernel functions, making it a valuable tool in various domains, including image classification, text analysis, and anomaly detection.

SVM Code:

```python
from sklearn.svm import SVC

svm = SVC(kernel='linear', C=1, random_state=369)
svm.fit(X_train, y_train)
```

Confusion Matrix for SVM:



Classification Report for SVM:

	precision	recall	f1-score	support
0.0	0.74	0.76	0.75	199
1.0	0.75	0.73	0.74	201
accuracy			0.74	400
macro avg	0.75	0.75	0.74	400
weighted avg	0.75	0.74	0.74	400

Precision: Indicates the accuracy of positive predictions. For class 0.0, the precision is 0.74, meaning that when the model predicts class 0.0, it is correct 74% of the time. For class 1.0, the precision is 0.75.

Recall: Also known as sensitivity or true positive rate, it measures the proportion of actual positives that were identified correctly. For class 0.0, the recall is 0.76, and for class 1.0, it is 0.73.

F1-Score: The harmonic means of precision and recall. It's a single metric that combines both precision and recall into one number. An F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0. Class 0.0 has an F1 score of 0.75, and class 1.0 has an F1 score of 0.74.

Support: The number of actual occurrences of the class in the specified dataset. For class 0.0, there are 199 instances, and for class 1.0, there are 201 instances.

Accuracy: The ratio of correctly predicted observations to the total observations. The accuracy of the model is 0.74, indicating that it correctly predicts the class 74% of the time.

Overall, this classification report indicates that the SVM classifier has a balanced performance on both classes, with a slight preference for class 1.0 in terms of precision, and a slight preference for class 0.0 in terms of recall. The overall accuracy and F1-scores are consistent and moderately high, suggesting that the model is reasonably good at making predictions for this task.

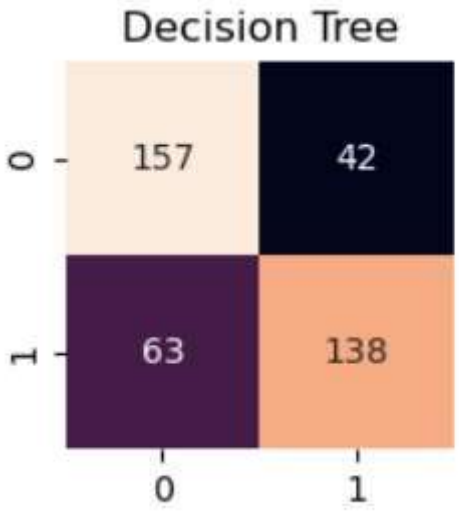
2. Decision Trees: A Decision Tree is a flowchart-like tree structure where an internal node represents a feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition based on the attribute value. It partitions the tree in a recursive manner called recursive partitioning. This flowchart-like structure helps you in decision making. Its visualization is a big advantage of using a decision tree as it is easy to understand. Decision trees are used for both classification and regression problems. They are fundamental components of Random Forests, which consist of many decision trees that vote on an outcome to improve the model's accuracy.

Decision Tree Code:

```
from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(criterion='gini', max_depth=5)
classifier.fit(X_train, y_train)
```

Confusion Matrix for Decision Tree:



Classification Report for Decision Tree:

	precision	recall	f1-score	support
0.0	0.71	0.79	0.75	199
1.0	0.77	0.69	0.72	201
accuracy			0.74	400
macro avg	0.74	0.74	0.74	400
weighted avg	0.74	0.74	0.74	400

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The Decision Tree model has a precision of 0.71 for class 0.0 and 0.77 for class 1.0.

Recall: Recall (or sensitivity) is the ratio of correctly predicted positive observations to all observations in the actual class. Here, the model has a recall of 0.79 for class 0.0, meaning it correctly identified 79% of all actual class 0.0 instances. For class 1.0, the recall is 0.69.

F1-Score: The F1-Score is the weighted harmonic mean of precision and recall. An F1-Score reaches its best value at 1 (perfect precision and recall) and worst at 0. The Decision Tree model has an F1-score of 0.75 for class 0.0 and 0.72 for class 1.0, which are moderate scores indicating a balanced precision-recall trade-off.

Support: This number represents the actual number of occurrences of each class in the dataset. There are 199 instances of class 0.0 and 201 instances of class 1.0.

Accuracy: This metric represents the ratio of correctly predicted observations to the total observations. The overall accuracy of the Decision Tree model is 0.74.

Overall, the model shows a balanced performance with a uniform average score across precision, recall, and the F1-score. The accuracy of the model indicates that it correctly predicts the class 74% of the time, which is the same as the average F1-score, suggesting consistency across the evaluation metrics.

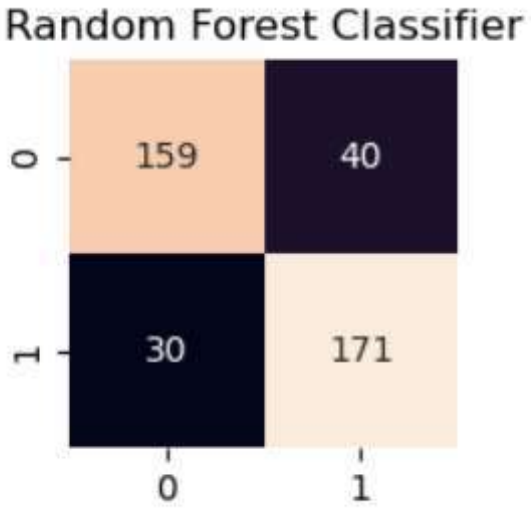
3. Random Forest: Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is one of the most flexible and easy to use algorithms, which, by combining many decision trees, helps to improve accuracy and control over-fitting. Random Forests can handle a large dataset with higher dimensionality in features and can estimate which variables are important in the classification. They are relatively robust to outliers and non-linear data. Random Forest works well with both categorical and numerical features and is a versatile algorithm that can be used for both regression and classification tasks.

Random Forest Code:

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100)
rf.fit(X_train, y_train)
```

Confusion Matrix for Random Forest:



Classification Report for Random Forest:

	precision	recall	f1-score	support
0.0	0.81	0.81	0.81	199
1.0	0.81	0.81	0.81	201
accuracy			0.81	400
macro avg	0.81	0.81	0.81	400
weighted avg	0.81	0.81	0.81	400

Precision: This is the measure of the accuracy of the model in classifying a sample as positive. The Random Forest model has a precision of 0.81 for both classes, meaning it is equally accurate for both classes when it predicts them as positive.

Recall: This is the measure of the model's ability to detect positive samples. The recall of 0.81 for both classes indicates that the model can identify 81% of the actual positives for both classes correctly.

F1-Score: This is the harmonic mean of precision and recall and is a single measure that combines both. An F1 score of 0.81 for both classes suggests a balanced precision and recall, which is generally an indication of a well-performing model.

Support: This refers to the number of actual occurrences of each class in the dataset. There are 199 instances of class 0.0 and 201 instances of class 1.0, indicating a balanced dataset.

Accuracy: This is the ratio of correctly predicted observations to the total observations. The accuracy of the Random Forest model is 0.81, suggesting that it correctly classifies 81% of the cases.

Overall, the classification report indicates that the Random Forest model has a uniform performance across both classes, with no significant bias toward either class. The consistent score of 0.81 across all metrics reflects a strong and balanced predictive performance.

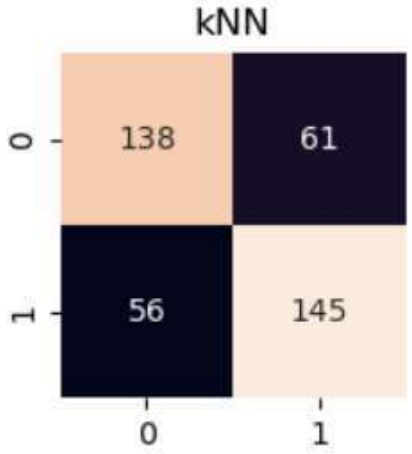
4. k Nearest Neighbors: k-Nearest Neighbors (KNN) is a simple, non-parametric, and versatile algorithm used for classification and regression. In classification tasks, it identifies the k closest training examples in the feature space to a given input and predicts the label most common among them; in regression, it averages the values. The definition of 'closest' typically uses distance metrics like Euclidean, Manhattan, or Minkowski distance. KNN is a type of instance-based learning, where the function is only approximated locally, and all computation is deferred until function evaluation. Despite its simplicity, KNN can achieve high accuracy in many real-world scenarios, but its performance can degrade with high-dimensional data due to the curse of dimensionality and can be computationally intensive for large datasets, as it requires storing the entire dataset and calculating distances for each query.

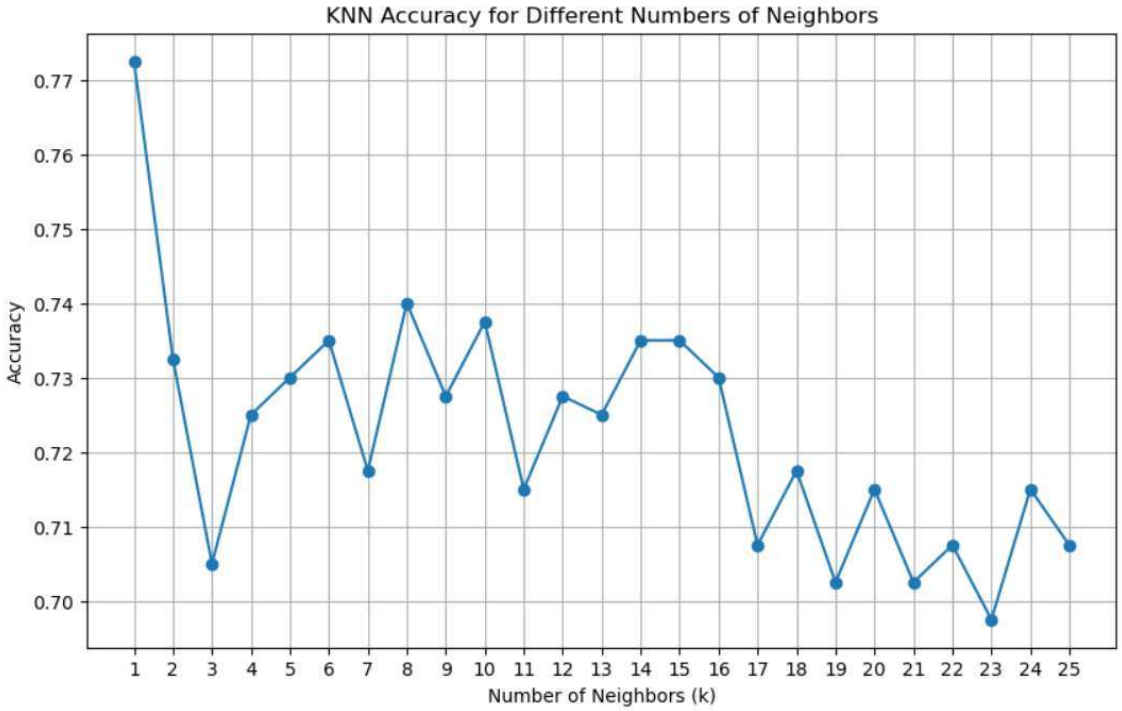
k-NN Code:

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)
```

Confusion matrix for k-NN:





From the above graph we can observe that the highest accuracy is achieved with just one neighbor (this could be due to overfitting, where the model performs very well on the training data but may not generalize well to new, unseen data).

As the number of neighbors increases to 2, there's a significant drop in accuracy, which then fluctuates as increases.

Overall, accuracy seems to have a downward trend with some fluctuations as the number of neighbors increases.

At k=8 there might be an optimal solution.

Classification Report for k-NN:

	precision	recall	f1-score	support
0.0	0.71	0.69	0.70	199
1.0	0.70	0.72	0.71	201
accuracy			0.71	400
macro avg	0.71	0.71	0.71	400
weighted avg	0.71	0.71	0.71	400

Precision: This is the proportion of true positive predictions in all positive predictions for class 0.0. The k-NN model has a precision of 0.71 for class 0.0, The precision for class 1.0 is slightly lower at 0.70, indicating that when the model predicts an instance as class 1.0, it is correct 70% of the time.

Recall: The recall for class 0.0 is 0.69, which means that the model correctly identifies 69% of all actual instances of class 0.0, The model has a slightly higher recall for class 1.0 at 0.72, suggesting that it correctly identifies 72% of all actual instances of class 1.0.

F1-Score: The F1-score is a balance between precision and recall, and for class 0.0, it is 0.70, Class 1.0 has a marginally higher F1-score of 0.71, indicating a slightly better balance between precision and recall for this class.

Support: The support is the number of actual occurrences of each class in the dataset, with 199 for class 0.0 and 201 for class 1.0, indicating a balanced dataset.

Accuracy: The accuracy of the k-NN model is 0.71, suggesting that it correctly predicts the class for 71% of the instances.

5. Linear Regression: Linear regression is a statistical method used for modeling the relationship between a dependent variable (usually denoted as "Y") and one or more independent variables (usually denoted as "X"). It assumes that this relationship is linear, meaning that changes in the independent variables result in proportional changes in the dependent variable.
The goal of linear regression is to find the best-fit line (a straight line in the case of simple linear regression with one independent variable) that minimizes the difference between the predicted values and the actual observed values. This line is represented by an equation in the form of Y = aX + b, where "a" is the slope of the line and "b" is the intercept.

Linear Regression:

OLS Regression Results

Dep. Variable:	quality	**R-squared:**	0.361
Model:	OLS	**Adj. R-squared:**	0.356
Method:	Least Squares	**F-statistic:**	81.35
Date:	Mon, 11 Dec 2023	**Prob (F-statistic):**	1.79e-145
Time:	14:39:58	**Log-Likelihood:**	-1569.1
No. Observations:	1599	**AIC:**	3162.
Df Residuals:	1587	**BIC:**	3227.
Df Model:	11		
Covariance Type:	nonrobust		

R2 Score: From above we can see that R-squared value is too low because the data set that we have chosen is more of a Classification problem. That is why we have performed other classification models and achieved an accuracy of more than 70% for all models. It's a classification model that's incredibly simple to implement and performs admirably with linearly separable classes. The output is a score of approximately 0.361, which is the R-squared value indicating that about 36.1% of the variance in the dependent variable is explained by the model on the test dataset. This suggests that the model's predictive power is relatively low for the given data.

Conclusion:

- The SVM showed precision, recall, and F1-scores around 0.74-0.75 with an accuracy of 0.74.
- The Decision Tree showed slightly lower metrics with precision, recall, and F1-scores in the range of 0.71-0.72 and an accuracy of 0.74.
- The Random Forest had uniform metrics with precision, recall, F1-scores, and accuracy all at 0.81.
- The k-NN showed precision and recall both at 0.71, with F1-scores around 0.70-0.71 and an accuracy of 0.71.

From these metrics, the Random Forest model outperformed the other models with the highest precision, recall, F1-score, and accuracy, all at 0.81. This suggests that the Random Forest model was the most effective at classifying the given dataset among the models compared.

Summary:

The Random Forest model showed the best overall performance, with balanced precision and recall across both classes, indicating a strong and consistent ability to classify the data accurately.

The SVM model followed, with slightly lower but still comparable performance metrics.

The Decision Tree and k-NN models had similar metrics to each other but were lower than those of the SVM and Random Forest models, indicating that they may not capture the complexities of the dataset as effectively as the Random Forest model.