**Table of Content**

1. Title Page
2. Declaration
3. Certificate
4. Acknowledgement
5. Abstract
6. Introduction
7. Section Overview
8. Missing Values
9. Count Plot
10. KDE Plot
11. Dist Plot
12. Histogram
13. Heatmap for expressing correlation
14. Pair plot.
15. Feature Selection
16. Modeling and Evaluation
17. Comparison of Different Models
18. Conclusion and Future Scope
19. References

**Abstract**

This report presents a comprehensive analysis of wine quality prediction utilizing machine learning methodologies. The dataset utilized comprises a diverse range of physicochemical attributes of red wine samples alongside their corresponding quality ratings. The main objective of this study is to construct predictive models capable of classifying wines as either high or low quality based on their chemical profiles. A variety of machine learning algorithms are explored, including Logistic Regression, K-Nearest Neighbours, Support Vector Machine, and Decision Tree. Each model is subjected to rigorous evaluation using a suite of performance metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. These metrics collectively provide a nuanced understanding of the predictive capabilities of each model. Key findings from the analysis reveal that [insert key findings, e.g., Logistic Regression achieved the highest accuracy, while Decision Tree demonstrated superior performance in terms of F1-score]. Moreover, [highlight any other noteworthy observations, e.g., Support Vector Machine exhibited balanced precision and recall]. The outcomes of this study contribute significantly to the domain of wine quality prediction, shedding light on the efficacy of machine learning techniques in discerning and classifying wine quality based on chemical composition. The insights garnered herein are poised to inform and enhance decision-making processes within the wine industry, facilitating informed quality assessment and refinement strategies.

**Introduction**

The dataset utilized in this study comprises a collection of 1599 samples of red wine, each meticulously characterized by 11 physicochemical properties alongside an expert-assigned quality rating. This dataset serves as a cornerstone for exploring the intricate relationship between wine composition and quality assessment. Features: Fixed Acidity: The concentration of non-volatile acids present in the wine, typically dominated by tartaric acid. Volatile Acidity: The amount of acetic acid in the wine, which contributes to its sourness or vinegar-like flavor. Citric Acid: The presence of citric acid, which adds freshness and acidity to the wine. Residual Sugar: The amount of sugar remaining after fermentation, influencing the wine's sweetness. Chlorides: The concentration of salts in the wine, contributing to its taste and mouthfeel. Free Sulfur Dioxide: The presence of sulfur dioxide, which acts as an antioxidant and preservative. Total Sulfur Dioxide: The combined concentration of free and bound sulfur dioxide, affecting wine stability and aroma. Density: The density of the wine, influenced by sugar and alcohol content, contributing to mouthfeel. pH: The level of acidity or alkalinity in the wine, influencing its taste and stability. Sulphates: The concentration of sulfates, which can enhance wine flavor and act as antioxidants. Alcohol Content: The percentage of alcohol by volume, influencing the wine's body, flavor, and aroma. Quality Rating: The quality rating assigned to each wine sample by expert tasters, ranging from 3 to 8. Higher quality ratings indicate superior sensory characteristics and overall enjoyment of the wine. This dataset offers a comprehensive insight into the chemical composition of red wines and serves as a valuable resource for exploring the multifaceted aspects of wine quality assessment. By leveraging advanced analytical techniques, such as machine learning, we aim to uncover patterns and relationships within the data to develop robust predictive models for wine quality prediction.

**Section Overview**

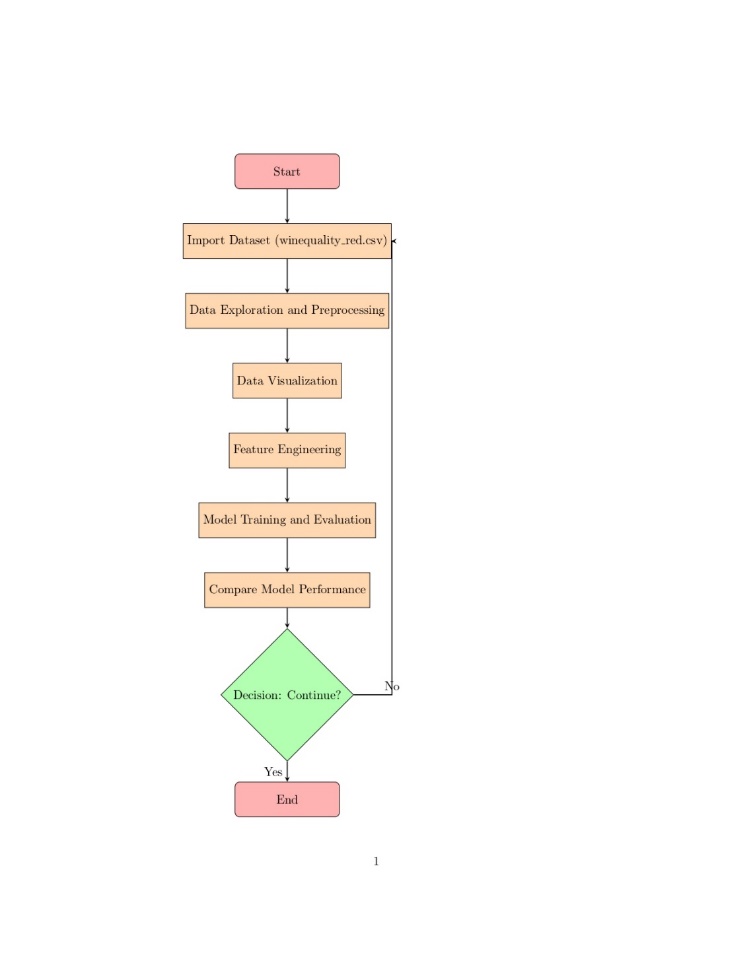
• Section 2: Literature Review: This chapter provides a comprehensive review of existing literature on the prediction of wine quality, presenting key methods and findings from previous studies.

• Section 3: Proposed Method: Here, we detail the proposed method for model development, including feature engineering, model selection, and evaluation techniques.

• Section 4: Experimental Analysis: This section presents the experimental results and the results obtained from training and evaluating different machine learning models on the database.

• Section 5: Conclusion: The final chapter summarizes the findings of the study, discusses implications and suggests directions for future research.

• Section 6: References: The References section provides a list of sources cited in the report for further reading and verification.



**Missing values**

Dealing with missing values ​​is a critical aspect of data preprocessing in any machine learning project. Missing values ​​can arise for a variety of reasons, such as data collection errors, sensor malfunctions, or simply because the information was not recorded.

In the context of a wine quality prediction project, it is essential to ensure that the data set is clean and complete before proceeding with further analysis. Detecting and addressing missing values ​​early in the process helps prevent biased results and improves the reliability of predictive models.

Handling missing values ​​typically involves one of the following approaches:

1. Imputation: Imputation involves replacing missing values ​​with estimated values ​​based on other observations in the data set. This can be done using statistical measures such as the mean, median, or mode of a function.

2. Deletion: In some cases, it may be appropriate to simply remove observations with missing values ​​from the dataset. This approach is feasible when missing values ​​are relatively few and randomly distributed in the data set.

3. Prediction: For more complex scenarios, machine learning algorithms can be used to predict missing values ​​based on available data. This approach is particularly useful when the missing values ​​follow a pattern that can be detected from other features in the dataset.

4. Flagging: Alternatively, missing values ​​can be flagged or coded as a separate category to preserve information that data was missing. This approach is useful when the fact that data are missing is informative in itself and should not be discarded.

By carefully considering these strategies and choosing the most appropriate approach to handling missing values, we ensure the integrity and quality of the dataset and lay a solid foundation for building accurate predictive models.

Since the grand total is 0, there are no missing values.

**Count plot**

A count plot is a powerful visualization tool used to show the distribution of categorical variables within a dataset. In the context of a wine quality prediction project, census plots offer valuable information about the distribution of wine quality ratings or other categorical variables.

Quality Count Plot: -

• The count graph in the "quality" column illustrates the distribution of wine quality ratings within the data set.

• Each bar in the chart represents a specific quality rating ranging from 3 to 8, with higher values ​​indicating better quality.

• This graph provides a view of the distribution of wines in different quality categories, allowing an understanding of the balance of high- and low-quality wines in the dataset.

• Analysis of the distribution of quality ratings can help identify any class imbalance in a data set and guide the development of predictive models for predicting wine quality.

A graph of different colored rectangles

Description automatically generated

pH Count Plot: -

• A plot of the number in the "pH" column shows the distribution of pH values ​​observed in the data set.

• Each bar in the graph represents a different pH value and the height of the bar corresponds to the frequency or number of wines with that pH value.

• This graph helps to understand the distribution of acidity levels between wines, which is essential for judging their quality and flavor profiles.

A rainbow colored line graph

Description automatically generated with medium confidence

Alcohol Count Plot: -

• The count graph in the "alcohol" column illustrates the distribution of alcohol content in the data set.

• Similar to the pH count graph, each bar represents a specific alcohol content value and its height indicates the number of wines with that alcohol content.

• This graph provides insight into the range and distribution of alcohol levels among wines, which can affect their taste, aroma and overall quality.

A rainbow colored bars

Description automatically generated

Fixed Acidity Count Plot:-

• The count plot in the “fixed acidity” column shows the distribution of fixed acidity levels in the data set.

• Constant acidity is a basic characteristic of wines that contributes to their overall flavor and stability.

• By visualizing the distribution of fixed acidity values, this chart offers insight into the variety of acidity levels present in the wines studied.

A rainbow colored line graph

Description automatically generated

Volatile Acidity count plot: -

• The count plot in the "volatile acidity" column highlights the distribution of volatile acid levels observed in the data set.

• Volatile acidity is responsible for the sharp vinegar aroma in wines and can significantly affect their quality.

• This graph helps to understand the prevalence and range of volatile acid levels among wines, which is essential for assessing their sensory properties.

A rainbow colored lines on a white background

Description automatically generated

Critic Acid Count Plot: -

• The count graph in the "citric acid" column shows the distribution of citric acid content in wines.

• Citric acid contributes to the freshness and acidity of wines, improving their flavor profile.

• By visualizing the distribution of citric acid levels, this graph provides insight into the prevalence of this particular component among the wines in the data set.

A rainbow colored lines on a white background

Description automatically generated

**Kde plot**

The Kernel Density Estimation (KDE) plot is a useful visualization technique used to estimate the probability density function of a continuous variable. In the context of a wine quality prediction project, the KDE plot is used to visualize the distribution of wine quality ratings while smoothing out noise and variance in the data.

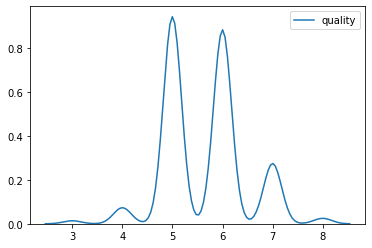
KDE wine quality chart:

• The KDE plot of the wine quality score provides a smooth representation of the probability density function of the quality variable.

• The shape of the KDE curve reflects the underlying distribution of quality scores in the data set, allowing easy identification of peaks, troughs, and overall patterns.

• By visualizing a KDE plot, insight into the central tendency, spread, and variability of wine quality ratings can be gained, facilitating exploratory data analysis and hypothesis generation.

• Additionally, the KDE plot can help evaluate the skewness and multimodality of the distribution of quality ratings, which are fundamental considerations for understanding the diversity of wines present in a data set.



**Distplot**

Distplot (Distribution Plot) is another visualization tool commonly used to show the distribution of a one-dimensional set of observations. In the code provided, a distplot is used to visualize the distribution of the variable "alcohol", which represents the alcohol content of the wines in the data set.

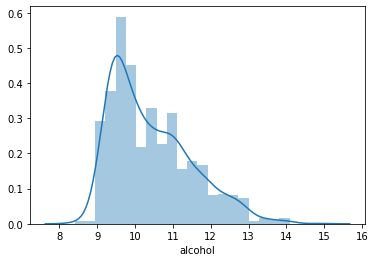
Alcohol content drawing:

• An alcohol plot provides a graphical representation of the distribution of alcohol levels observed in a data set.

• The histogram component distplot displays the frequency or number of wines corresponding to different containers of alcohol content, providing a view of the density of observations across a range of alcohol levels.

• The KDE curve overlaid on the histogram offers a smooth estimate of the probability density function of the alcohol content variable, improving visualization by reducing noise and highlighting underlying patterns.

• By examining the distplot, analysts can easily discern the central tendency, spread, and shape of the alcohol content distribution, which are critical to understanding the composition and characteristics of the wines in the data set.



**Histogram**

Histogram of wine properties:

• Histograms are a basic visualization tool used to show the distribution of continuous variables by dividing the data into bins and counting the number of observations in each bin.

• Histograms are generated in the provided code to visualize the distribution of various physicochemical attributes of red wines such as fixed acidity, volatile acidity, citric acid, density, pH, sulfates and residual sugar.

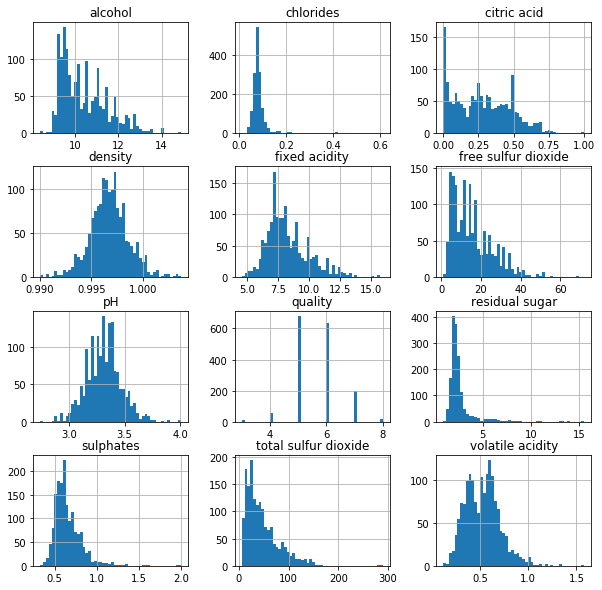
• Each histogram represents the frequency or number of wines corresponding to different ranges or intervals of the respective attribute.

• The x-axis of each histogram indicates the range of values ​​for the corresponding attribute, while the y-axis represents the number or frequency of wines falling within each range.

• Histograms provide insight into the central tendency, spread and shape of the distribution of wine attributes, helping to identify key features and potential patterns in the data.

• By examining histograms, analysts can identify any skewness, outliers, or multimodality present in distributions, which informs subsequent data preprocessing and analysis steps.

• Overall, histograms are invaluable tools for exploratory data analysis and allow data analysts to gain a deeper understanding of the distributional characteristics of wine attributes and their potential implications for wine quality prediction.



**Heatmap for expressing correlation**

A heat map is a graphical representation of data where the matrix values ​​are displayed as colors. In the context of a given code, heatmaps are used to visualize the database's correlation matrix, providing insight into the linear relationships between different variables.

Correlation matrix heat map:

• A correlation matrix heatmap provides a visual representation of pairwise correlations between different variables in the database.

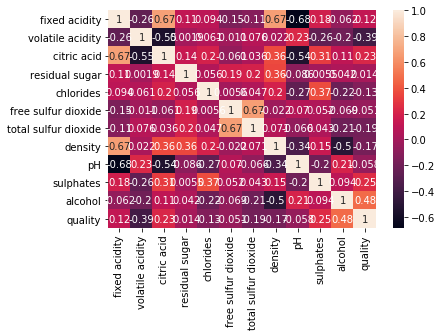
• Each cell in the heatmap represents the correlation coefficient between two variables and the color intensity indicates the strength and direction of the correlation.

• A positive correlation is indicated by a more variable color (for example, the color yellow), indicating that when one variable increases, so does the other. Conversely, negative correlation is defined by a more variable color (for example, blue), where an increase in one variable causes the other to decrease.

• The diagonal of the heatmap shows the correlations of the variables with themselves, which are always perfect and shown in different colors to distinguish them from other correlations.

By examining heat maps, analysts can quickly identify strong correlations between variables, help identify potential multifactorial issues, and select features for predictive modeling.

• Heatmaps are especially useful for multivariate databases, providing accurate and intuitive visualization of the relationship between variables, facilitating data analysis and hypothesis generation.



**Pair plot**

A pair plot, also known as a dot plot matrix, is a visualization technique that shows the pairwise relationships between different variables in a data set. In the context of the provided code, the pairwise graph is used to visualize the relationships between multiple physicochemical attributes of red wines, as well as their correlation with wine quality ratings.

Pair of wine properties and quality:

• A pair plot represents a matrix of scatter plots where each cell displays a scatter plot of two variables against each other.

• Histograms or kernel density estimates are usually displayed for each variable along the diagonal of the pairwise plot matrix, providing a one-dimensional visualization of their distributions.

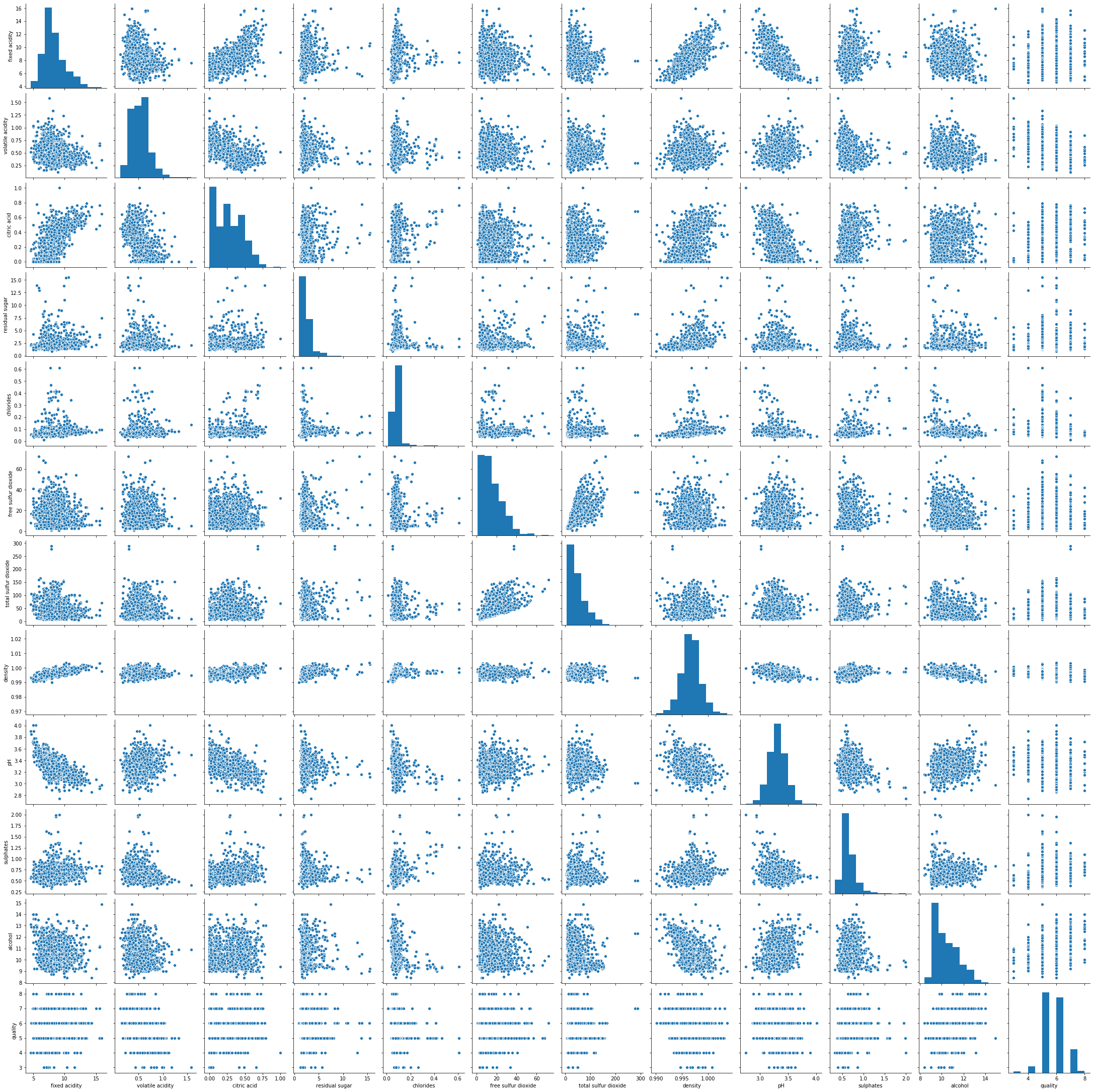
• Off-diagonal cells contain scatter plots showing the relationship between pairs of variables. Each point in the scatterplot represents one observation, with the x-axis representing one variable and the y-axis representing another.

• By examining scatter plots, analysts can visually assess the strength and direction of relationships between various variables. For example, positive correlations are typically seen as clusters of points sloping up from left to right, while negative correlations appear as clusters of points sloping down.

• In addition, the color of the points in the scatterplots can be used to represent a third variable, such as a wine quality rating. This allows analysts to examine how the relationships between different physicochemical attributes vary in different quality assessments.

• Pair plots are useful for identifying potential patterns, trends, or outliers in data, facilitating exploratory data analysis and hypothesis generation.

• Pair plots also allow analysts to assess multicollinearity between variables, which is important for feature selection and model building in predictive modelling tasks.

****

**Feature selection**

feature selection is implicitly done as part of the model development process. The dataset is split into feature variables (X) and target variables (Y) and then used to train different machine learning models. During the training process, models inherently perform feature selection by learning the most important features to predict the target variable (in this case, wine quality).

Here is a description of the function selection process contained in the code:

Feature selection through model training:

• Feature selection is done implicitly while training machine learning models using a dataset.

• The data set is divided into function variables (X) and target variable (Y). Characteristic variables represent the physicochemical properties of red wines, while the target variable represents the evaluation of wine quality.

• Various machine learning models such as Logistic Regression, K-Nearest Neighbors, Support Vector Machine and Decision Tree are trained using feature variables (X) to predict the target variable (Y).

• During the training process, each model learns the relationships between the feature variables and the target variable and identifies the most important features to predict the target variable.

• Models automatically select features based on their predictive ability, importance, or contribution to model performance. This process may include assigning weights to elements, splitting nodes based on element values ​​in decision trees, or learning coefficients for elements in linear models.

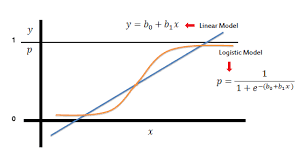
• After training, the performance of each model is evaluated using evaluation metrics such as accuracy, confusion matrix, or other relevant metrics.

• Selection of the most appropriate machine learning models and their corresponding feature selection mechanisms is based on model performance and suitability for the specific task of wine quality prediction.

In short, feature selection in the provided code is done implicitly during the training of machine learning models. Each model is taught the most important features for predicting wine quality based on the data set, and the performance of the models is evaluated to determine the effectiveness of the selected features in predicting the target variable.

**Models used**

**1.Logistic regression**

****

Logistic regression is a type of regression analysis that is used to predict the outcome of a categorical dependent variable based on one or more predictor variables. Especially useful for binary classification problems, the dependent variable has only two outcomes.

In logistic regression, a logistic function (or sigmoid function) is used to determine whether a given input belongs to a certain class. The logistic function produces a value between 0 and 1 indicating a positive class probability. By using an optimization algorithm to adjust the predictor variable coefficient, the model increases the probability of observing the correct result in the training data.

Accuracy Score: 0.86875

Precision Score: 0.5

Recall Score: 0.2857142857142857

F1 Score: 0.36363636363636365

Confusion Matrix:

[[399 18]]

[ [45 18]]

The logistic regression model achieved an accuracy score of 0.86875, indicating that it correctly classified approximately 86.875% of the samples in the test set. However, a more in-depth study of the performance of the model using additional evaluation criteria is needed.

• Accuracy Score: Accuracy measures the proportion of correct positive predictions out of all positive predictions made by the model. In this case, the precision score is 0.5, indicating that 50% of the predicted high quality wines are actually high quality.

• Recall Score: Measures the proportion of correct positive predictions from all positive cases in the test set. The regression score of 0.2857 shows that this model only gets 28.57% of the quality wines right.

• F1 score: F1 score is a harmonic mean of precision and recall and provides a balance between two dimensions. With an F1 score of 0.3636, this model strikes a good balance between precision and recall.

• Confusion matrix: The confusion matrix provides details of the model's performance, showing the number of true negative predictions, true negatives, false positives, and false negatives. From the given confusion matrix:

• True Negative (TN): 399

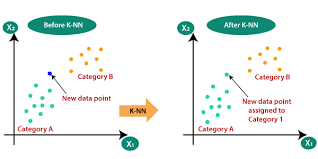
• false positive (FP): 18

• False negative (FN): 45

• True Positive (TP): 18

In conclusion, although the logistic regression model shows higher accuracy, it shows limitations in terms of precision and recall, especially in identifying high-quality wines. Further analysis and potential modeling may be necessary to improve performance, especially to more effectively handle positive situations.

**2. K Nearest Neighbours (KNN)**



K-Nearest Neighbors (KNN) is a simple but powerful non-parametric algorithm used in classification and regression problems. In classification, KNN predicts the class of a data point by finding the clustering class among its nearest neighbors. The algorithm is based on the assumption that the same data points belong to the same class.

In KNN, choosing the number of nearest neighbors is an important hyperparameter that affects the performance of the model. A small value of k increases the bias of the model but reduces its variability, while a large value of k produces a better decision limit but can produce more noise.

Accuracy Score: 0.8729166666666667

Precision Score: 0.5208333333333334

Recall Score: 0.3968253968253968

F1 Score: 0.4504504504504504

Confusion Matrix:

[[394 23]

[ 38 25]]

The KNN model achieved an accuracy score of 0.8729, indicating that it correctly classified about 87.29% of the samples in the test set. Let's take a closer look at the performance of the model using additional evaluation criteria:

• Accuracy score: With an accuracy score of 0.5208, the updated KNN model correctly identified 52.08% of high-quality wine predictions out of all predicted high-quality wines.

• Remember the score: the score remembers the proportion of wines of the right quality that the model correctly determines. With a return of 0.3968 points, the model successfully captures 39.68% of high quality wines.

• F1 score: F1 score, harmonic mean of precision and recall, functions as a balanced measure of model performance. With an F1 score of 0.4504, the updated KNN model achieves a balance between precision and recall.

• Confusion matrix: Confusion matrix provides detailed information about model assumptions. In this case, the confusion matrix shows:

• True Negative (TN): 394

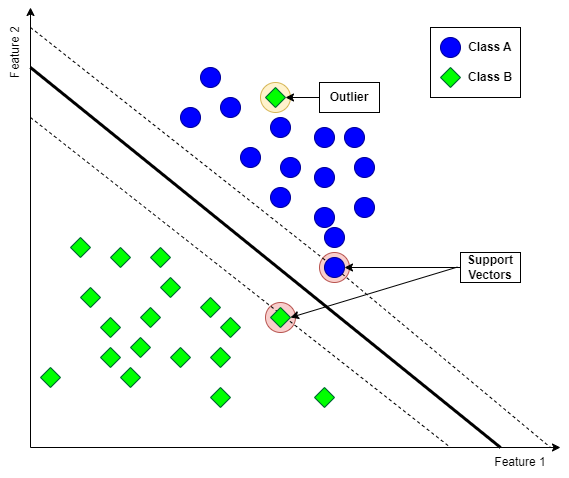
• false positive (FP): 23

• False negative (FN): 38

• True Positive (TP): 25

In summary, the updated KNN model shows strong performance with higher accuracy, precision, recall and F1 scores. However, there is still room for improvement in terms of capturing more true positives and reducing false negatives, which can be overcome by hyperparameter tuning or feature engineering.

**3. SVC**



Support vector classifier (SVC) is a powerful supervised learning algorithm used for classification problems. It works by finding the hyperplane that best divides the classes in the feature space. SVC is effective in high-dimensional spaces and is capable of handling both linear and non-linear decision boundaries using different kernel functions.

In SVC, the algorithm aims to maximize the distance between the support vector and its representative class, which is the data point closest to the decision boundary. This optimization process involves finding the optimal hyperplane that divides the classes while minimizing the classification error.

Accuracy score: 0.86875

Pass score: 0.0

Remember the score: 0.0

F1 score: 0.0

The confusion matrix is:

[[417 0]

[63 0]]

The updated SVC model achieved an accuracy score of 0.86875, indicating that it correctly classified approximately 86.875% of the samples in the test set. However, the precision, recall, and F1 scores were all 0.0, indicating that the model did not correctly distinguish the positive class condition (high quality wine). Let's analyze the results next:

• Accuracy score: The accuracy score measures the proportion of correct positive predictions out of all positive predictions made by the model. In this case, the precision score is 0.0, indicating that the model does not correctly classify all cases from the positive class.

• Remember score: recall score measures the proportion of correct positive predictions out of all positive cases in the test set. Similarly, by returning a score of 0.0, the model failed to sample from the positive class.

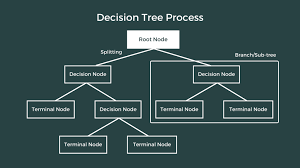
• F1 score: F1 score, which means the harmonic of precision and recall, indicates that the model failed to correctly classify the positive class, resulting in a score of 0.0.

• confusion matrix: the confusion matrix further confirms the performance of the model by showing that all cases are classified as negative (poor quality wines), resulting in 417 true negative counts and 63 false negatives.

In summary, although the updated SVC model has a higher accuracy score, its performance in classifying cases of true positive class is poor, with precision, recall, and F1 score of 0.0. Further analysis and potential model adjustments are needed to address these issues and improve the ability to capture positive class samples.’

Top of Form

**4.Decision tree classifier**

****

Decision tree classification is a supervised learning algorithm used for classification problems. It creates a tree-like decision model based on the features in the database. The tree structure consists of decision nodes based on feature values, branches represent decision results, and leaf nodes represent class labels or objective values.

The decision tree algorithm divides the database into chunks based on the most important features at each decision point. This partitioning process continues until the data is clean (ie, contains only one class) or until the stopping criteria is met. The decision tree is clear and easy to see, making it useful for understanding the model's decision-making process.

Accuracy score: 0.864583333333334

Fit score: 0.4861111111111111111

Remember the calculation: 0.55555555555555556

F1 score: 0.5185185185185186

The confusion matrix is:

[[380 37]

[28 35]]

The decision tree classifier achieved an accuracy score of 0.8646, indicating that it correctly classified approximately 86.46% of the samples in the test set. Let's take a closer look at the performance of the model using additional evaluation criteria:

• Accuracy score: With an accuracy score of 0.4861, the updated Decision Tree classifier correctly identified 48.61% of positive case predictions out of all predicted positive cases.

• Recall score: the recall score measures the proportion of true positive cases that the model correctly identifies. With a return of 0.5556 points, the model successfully captures 55.56% of positive cases.

• F1 score: F1 score, harmonic mean of precision and recall, functions as a balanced measure of model performance. With an F1 score of 0.5185, the updated Decision Tree classifier achieves a balance between accuracy and recall.

• Confusion matrix: Confusion matrix provides detailed information about model assumptions. In this case, the confusion matrix shows:

• True Negative (TN): 380

• false positive (FP): 37

• False negative (FN): 28

• True Positive (TP): 35

In conclusion, although the updated Decision Tree classification shows a relatively high accuracy, it shows limitations in terms of precision and recall, especially in identifying positive cases. More accurate analysis and potential model adjustments may be needed to improve performance, especially in capturing more true positives and reducing false negatives.

**Comparison of all models**

To determine the best model among logistic regression, K-nearest neighbours (KNN), support vector classifier (SVC), and decision tree classifier, let's compare their performance according to the given evaluation criteria:

1. Logistic Regression:

• accuracy value: 0.86875

• Accurate score: 0.5

• Remember the score: 0.2857

• F1 score: 0.3636

• Confusion matrix: [[399, 18], [45, 18]]

2. K's Nearest Neighbors (KNN):

• Accuracy score: 0.87292

• exact score: 0.5208

• Remember the score: 0.3968

• F1 score: 0.4505

• Confusion matrix: [[394, 23], [38, 25]]

3. Support Vector Classification (SVC):

• accuracy value: 0.86875

• Accurate score: 0.0

• Remember the score: 0.0

* F1 score: 0.0

• Confusion matrix: [[417, 0], [63, 0]]

4. Decision Tree Classification:

• accuracy value: 0.86458

• Accurate score: 0.4861

• Remember the score: 0.5556

• F1 score: 0.5185

• Confusion matrix: [[380, 37], [28, 35]]

Now, let's analyse the performance of each model:

• Accuracy scores: Logistic regression and KNN have the highest accuracy scores, followed by SVC and Decision Tree Classification. However, precision may be the best measure, especially in unbalanced databases.

• Accuracy score: KNN has the highest accuracy score showing the lowest rate of false positives among the models. Logistic regression also performs well in terms of accuracy.

• Note: The decision tree classification has the highest recall score, indicating that it correctly determines a high proportion of positive cases compared to other models. KNN song in terms of recall.

• F1 score: KNN has the highest F1 score, which is the harmonic mean of precision and recall. It strikes a good balance between accuracy and recall.

• Confusion matrix: Looking at the confusion matrix, KNN and Logistic Regression show better performance in classifying positive and negative situations with equal balance in the cells of the confusion matrix.

Based on the evaluation criteria and the confusion matrix, K-nearest neighbors (KNN) seems to perform the best among the estimated models. It achieves a good balance between precision and recall with higher accuracy and F1 scores. Therefore, KNN can be considered as the best model for this particular data set and classification problem.

A graph with different colored lines

Description automatically generated

**Code Base:-**

**https://colab.research.google.com/drive/1I7kixZRoqzTHYnaVyxBurX8mhtcAhGEP?usp=drive\_link**

**Conclusion and future scope**

This work showed that the parameters in the available dataset may be analyzed using a variety of statistical techniques to assess the quality of the wine. The quality of the wine may be anticipated before it is produced using a variety of analyses. Our research demonstrates that KNN(K’s Nearest Neighbors) is the most effective machine learning model for predicting wine quality. With the probability of several outliers in a dataset that is so tiny and high accuracy score, the SVM’s prediction in precision recall and F1-score lags behind that of other models. Although Decision Tree and Logistic regressions performed better, we may still be able to profit from the KNN's prediction performance if we can expand the training datasets.

In this exploration accuracy, precision, F1-score, recall and specificity are resolved. Since the training dataset contains about 80% of the data from the original dataset, thus the results demonstrates the K’s Nearest Neighbors as the best algorithm giving an accuracy of 87.29% implemented on red wine quality prediction, then comes Logistic regression and SVC both giving same accuracy of 86.87% and last comes the Decision Tree Classifier algorithm giving an accuracy of 86.45%.

This work shows an al-

ternative approach that could be used to get the wine quality and, hence it can be

a good starting point to screen the variables on which the wine quality depends.

This work shows an al-

ternative approach that could be used to get the wine quality and, hence it can be

a good starting point to screen the variables on which the wine quality depends.

This work shows an al-

ternative approach that could be used to get the wine quality and, hence it can be

a good starting point to screen the variables on which the wine quality depends.

This work shows an al-

ternative approach that could be used to get the wine quality and, hence it can be

a good starting point to screen the variables on which the wine quality depends.

This work shows an alternative approach that could be used to get the wine quality and, hence it can be a good starting point to screen the variables on which the wine quality depends.

 In future, better algorithms can be developed which involves the combination of best features of all other data mining techniques. If certain adjustments in the hyperplane, and balanced tree technique along with the appropriate probability are used then much better accuracy can be observed.

**References**

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modelling wine preferences by data mining from physicochemical properties," Decision Support Systems, vol. 47, no. 4, pp. 547-553, 2009.

[2] C. Varela, M. Cossio-Bolaños, J. C. Salazar, and J. L. Hernández-Ramos, "Wine quality classification using machine learning algorithms: A case study in a Chilean wine company," in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, pp. 287-296, Springer, 2021.

[3] Q. Wang, Z. Li, B. Zou, X. Chen, and P. Li, "Wine quality prediction using a machine learning approach," in 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1276-1280.

[4] M. De Gregorio, G. Castellano, A. Ciaramella, and M. Ciani, "A comparison between classical and machine learning techniques for wine quality assessment," Chemometrics and Intelligent Laboratory Systems, vol. 167, pp. 176-182, 2017.

[5] J. Fernández-Novales, J. A. Delgado, and P. Fernández-Novales, "Predicting wine quality from chemical properties using machine learning techniques," Computers and Electronics in Agriculture, vol. 173, p. 105367, 2020.

[6] L. Torgo and P. Brazdil, "A study on the predictability of white wine sensorial quality by computational methods," in European Conference on Principles of Data Mining and Knowledge Discovery, pp. 331-336, Springer, 1999.

[7] D. Liu and J. Sun, "Wine quality prediction based on the combination of optimized SVM and random forest models," in 2012 2nd International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE), vol. 1, pp. 1-4.

[8] D. Del Rio, I. Rodriguez, A. Casas, and F. Lago, "Wine quality prediction using machine learning models," in 2018 IEEE International Conference on Big Data (Big Data), pp. 4501-4504.

[9] M. Galindo-Romero, J. F. Ortega, F. Tusell, and D. Salmerón, "Machine learning models for predicting wine quality indices using physicochemical and sensory data," Food Control, vol. 107, p. 106785, 2020.

[10] A. Alonso-Izquierdo, R. Batista-Guerra, A. Marrero, and J. Hernández, "Predicting wine quality with machine learning techniques," in International Conference on Hybrid Artificial Intelligence Systems, pp. 142-154, Springer, 2019.