WHITE WINE QUALITY PREDICTION

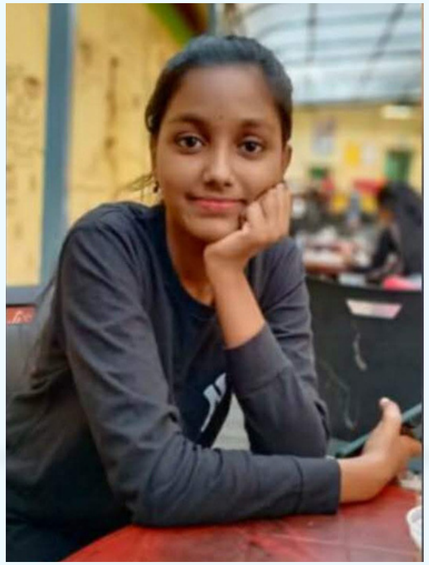
CLASSIFICATION METHOD

Presented by group 8

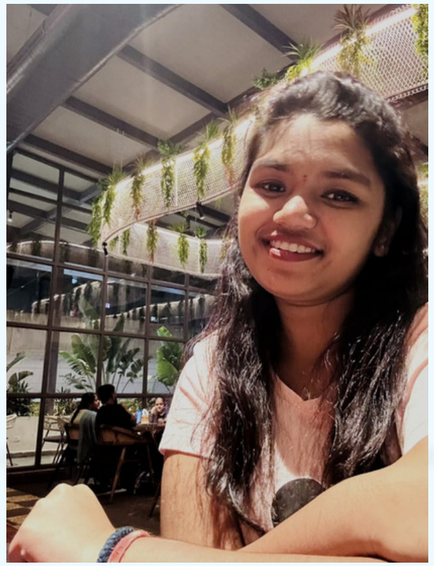
DG Ruparel College of Arts, Science and Commerce | RBVRR COLLEGE FOR WOMENS

Meet The Group

VRUTI TAILOR



S DHAARANI



DARA DARSHITHA

KAKKAD SIMRAN

SHESHI REKHA

Our Project

- Data Preprocessing
- Data Exploration
- Machine learning algorithms
- Summary and Conclusion

Background

Wine quality prediction is a task that can benefit from the use of machine learning. By training a model on a dataset of wine characteristics and corresponding quality ratings, a machine learning algorithm can learn to predict the quality of new wines based on their characteristics.They define wine quality as the 'color–flavor–fragrance intensity of a given wine with respect to all the other wines in its appellation. '

Objective

The objectives of this project are as follows :

1. To experiment with different classification methods to see which yields the highest accuracy.
2. To determine which features are the most indicative of a good quality wine

Path: Implementing Multiple machine learning models to fit the best model for the data set.

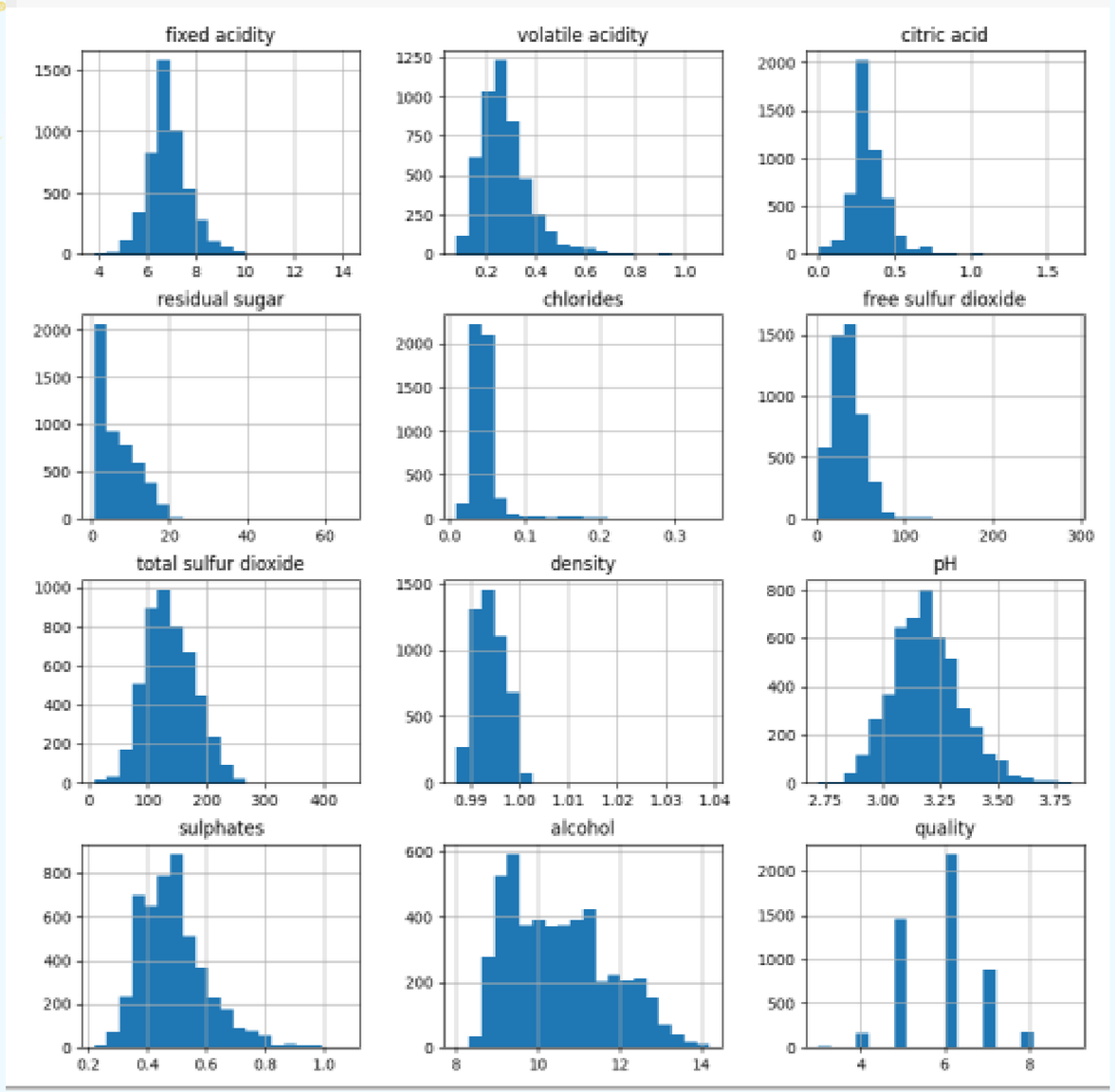
Data And Data Quality Check

- **Data Introduction :** The Data consists of 4898 rows × 12 columns.
- **Variable:** Fixed acidity,Volatile acidity,Citric acid,Residual sugar,Chlorides,Free sulfur dioxide,Total sulfur dioxide,Density,pH,Sulfates,Alcohol,Quality.
- **Missing Values :** The data set has no missing values.
- **Dropping Variables:** There are no dropped values

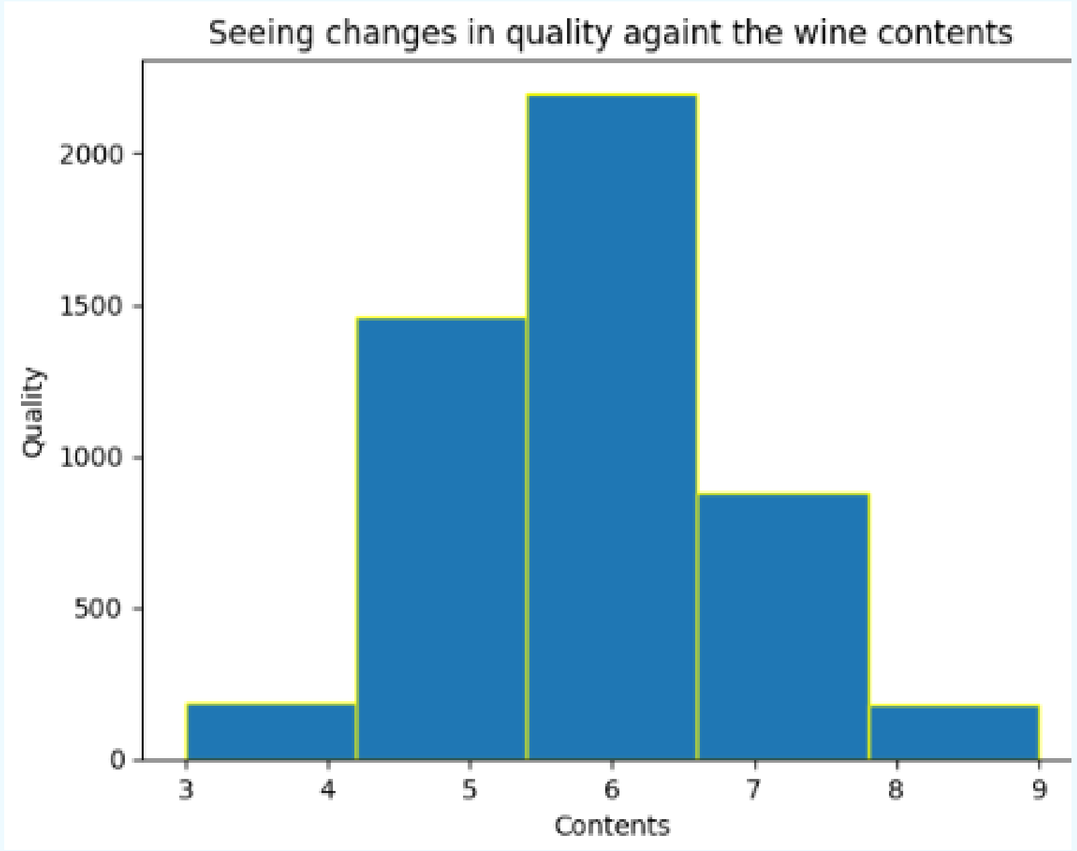
Data Visualization

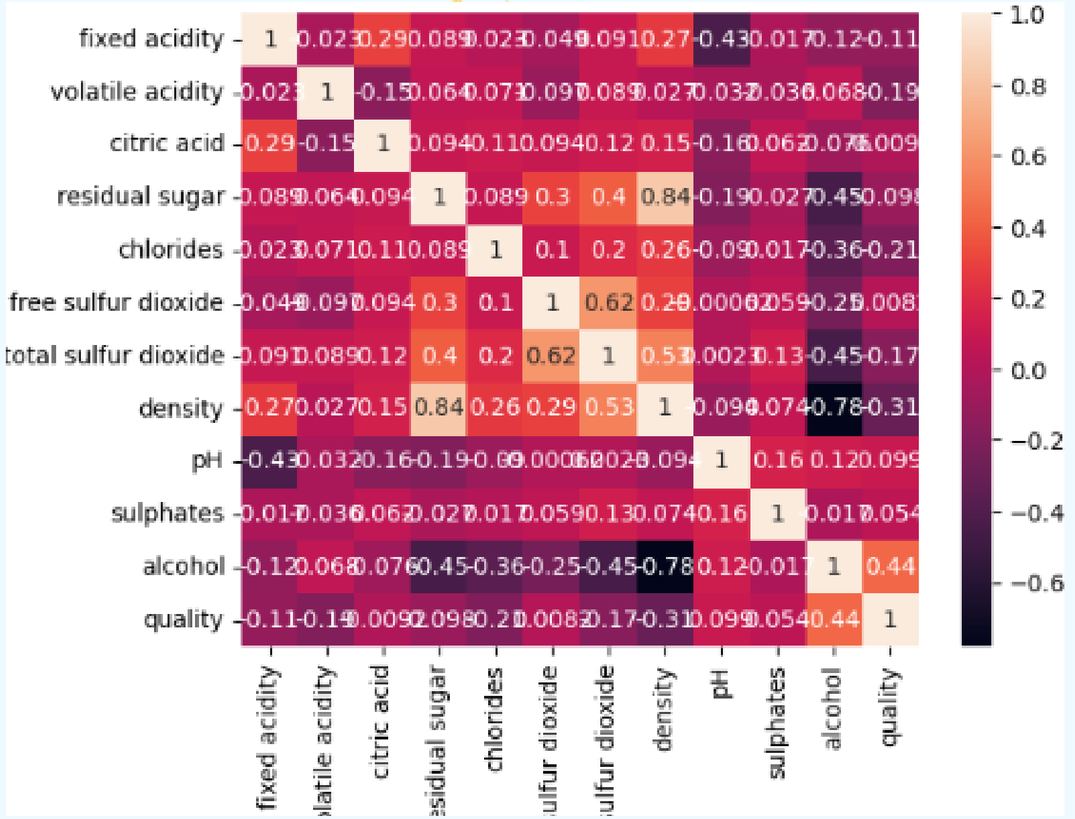
We know that the “image speaks everything” here the visualization came into the work, we use visualization for explaining the data. In other words, we can say that it is a graphic representation of data that is used to find useful information.

The image reveals that how that data is easily distributed on features.

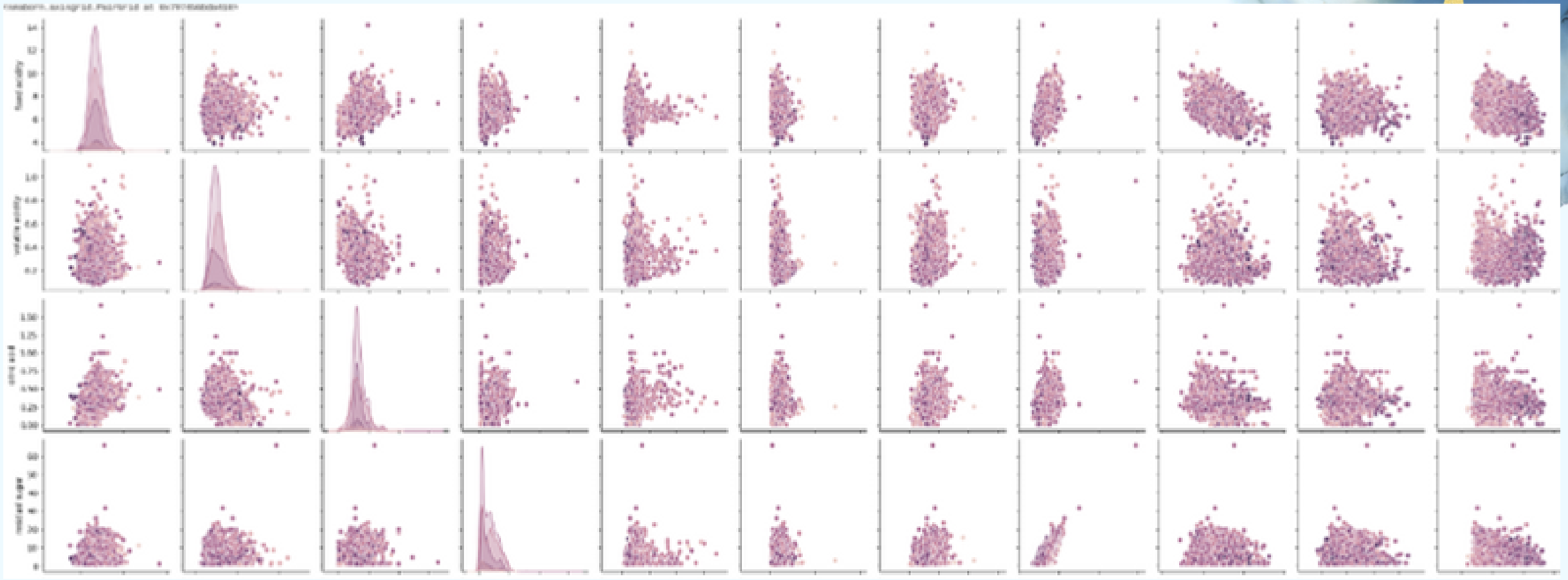


In the below histogram we can see the changes in quality against wine contents



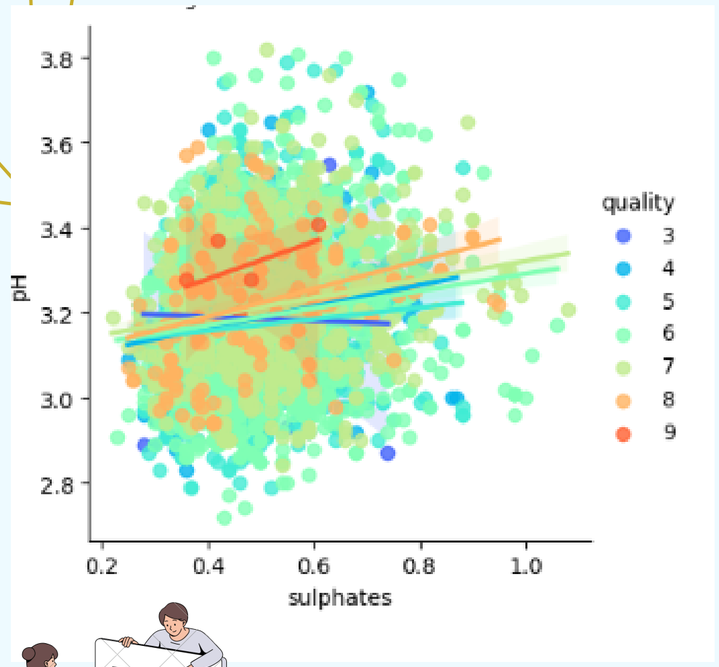


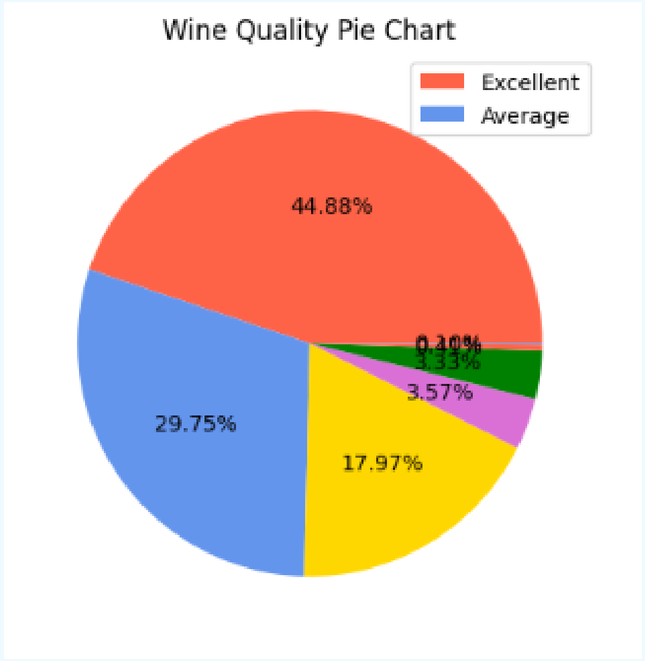
From the above heatmap we can conclude that the ‘total sulphur dioxide’ and ‘free sulphur dioxide‘ are highly correlated features so, we will remove them.

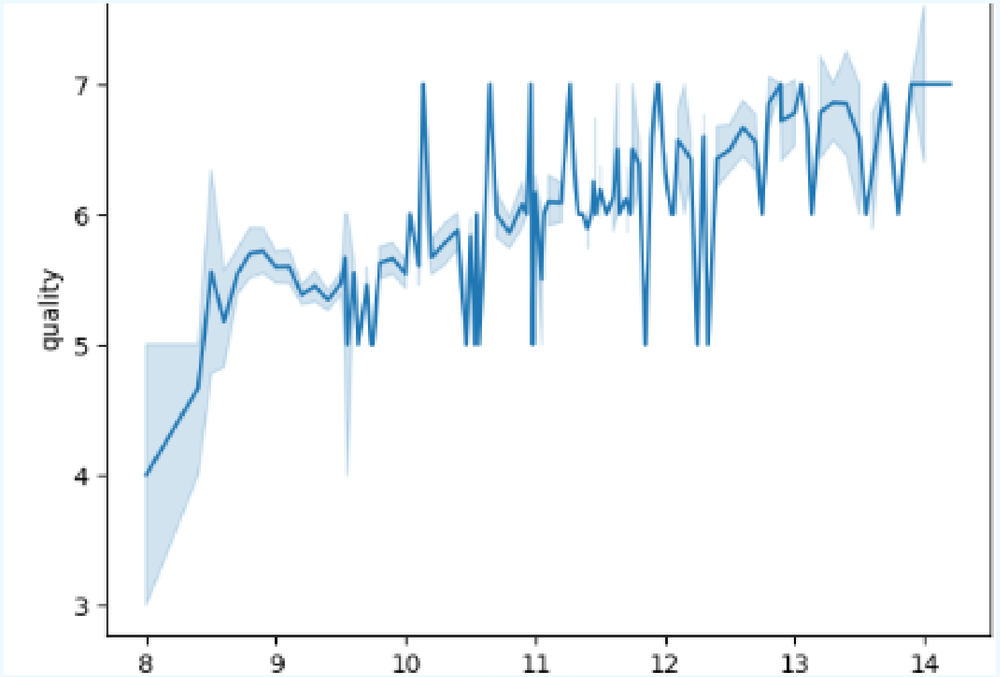


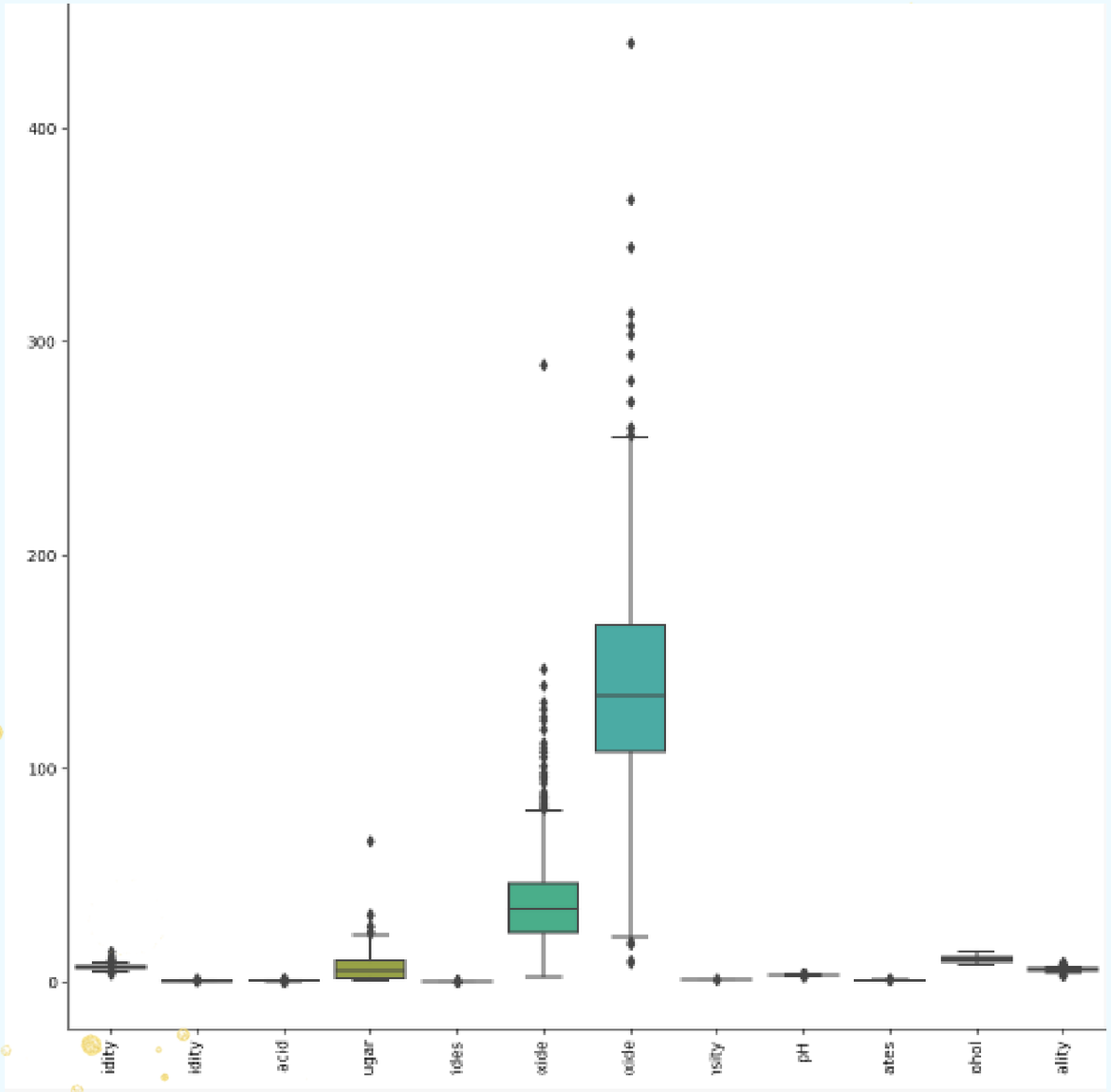
We aim to visualize the correlation of each feature pair in a dataset against the class distribution. Pairplot is a module of seaborn library which provides a high-level interface for drawing attractive and informative statistical graphics.

Pie Chart: for showing the relative shares of wine quality in a total.
Line Chart: for showing trends in a series over time between quality and alcohol.
Scatter Plot: for showing the relationship between sulphates and pH.

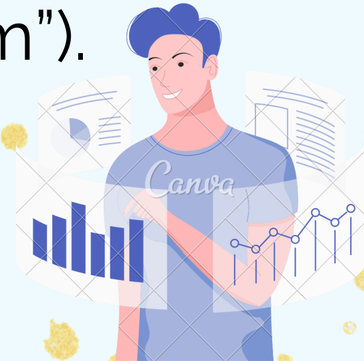








A boxplot is a standardized way of displaying the distribution of data based on its five-number summary (“minimum”, first quartile [Q1], median, third quartile [Q3] and “maximum”).



Machine Learning Algorithms

Logistic Regression

Decision Tree Classifier

Random Forest Classifier

K nearest neighnour (KNN)

Support Vector Machine Classifier

Logistic Regression

- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Model Statistics using a test ratio of 0.20
accuracy 0.54
Model Statistics using a test ratio of 0.35
accuracy 0.34

Logistic Regression model accuracy (in %): 51.63%

Decision tree Classifier

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

Model Statistics using a test ratio of 0.20
Max depth 10: accuracy 0.56
Max depth 5 : accuracy 0.53
Max depth 20: accuracy 0.58
Max depth 100: accuracy 0.58

Decision tree model Accuracy (in %): 58.00%

Random Forest Classifier

"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Model Statistics using a test ratio of 0.35
Max depth 10 : accuracy 0.63
Model Statistics using a test ratio of 0.30
Max depth 15: accuracy 0.69

Random Forest Classifier Acuuracy (in %): 69%

K- nearest neighbour (KNN)

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

Model Statistics using a test ratio of 0.25

accuracy : 0.57

Model Statistics using a test ratio of 0.20

accuracy : 0.59

Model Statistics using a test ratio of 0.30

accuracy : 0.57

K-NN Classifier Accuracy in % : 59%

Support Vector Machine Classifier

The main objective of the SVM algorithm is **to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space**. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible.

Model Statistics using a test ratio of 0.25

accuracy : 0.60

Model Statistics using a test ratio of 0.35

accuracy : 0.59

Model Statistics using a test ratio of 0.20

accuracy : 0.59

SVM Classifier Accuracy in % : 60%

Comparison of Implemented Models

Model	Accuracy_score
Logistic Regression	51.63
Decision Tree	58.00
Random Forest	69.00
K Nearest Neighbor	59.00
Support vector Machine	60.00

Summary and Conclusion

Wine quality prediction is a meaningful task considering the great popularity of wine worldwide. The main difficulty of applying machine learning techniques for wine quality prediction is the limited availability of training data and imbalanced data. This work is focused on exploring the factors that might influence the quality of the wine. Wine with great quality could not only appeal to more customers and even enjoy a worldwide reputation.

FUTURE INSIGHTS THAT CAN HELP INCREASE CUSTOMER RETENTION INCLUDE THE FOLLOWING:-

- Total sulfur dioxide, chlorides and volatile acidity are the most important for predicting wine type, while alcohol, citric acid and residual sugar are the least important.

- Future researchers should take into account more than currently used machine learning models such as logistic regression, multiple regression, and SVM etc, so that better comparisons between all of the various options could be made.

THANK YOU

Click here to refer:

https://colab.research.google.com/drive/1-Rv8KsdzNGZgkqq8ZekQU0xrZzymJawW?usp=sharing

Presented By :

S.Dhaarani

Vruti Tailor

Dara Darshitha

Kakkad Simran

K.Sheshi Rekha

Appendix:

LOGISTIC REGRESSION

```
[28] X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.20, stratify = y, random_state=30)
```

```
[29] len(X_test)
```

```
980
```

```
[30] len(X_train)
```

```
3918
```

```
[31] lgmodel = LogisticRegression()
     lgmodel.fit(X_train, y_train)
     prediction = lgmodel.predict(X_test)
```

DECISION TREE

```
[37] X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.20, stratify = y, random_state=30)
```

```
[38] dmodel = DecisionTreeClassifier(max_depth = 10)
     dmodel.fit(X_train , y_train)
     prediction = dmodel.predict(X_test)
```

RANDOM FOREST

```
[46] X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.35, stratify = y, random_state=30)
```

```
[47] rfmodel = RandomForestClassifier(n_estimators =  100 , max_depth = 10)
     rfmodel.fit(X_train , y_train)
     prediction = rfmodel.predict(X_test)
```

```
[48] print(classification_report(y_test , prediction))
```

K Nearest Neighbor Classifier (KNN)

```
56] X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.25, stratify = y, random_state=30)
    knnc = KNeighborsClassifier()
    knnc.fit(X_train , y_train)
    prediction = knnc.predict(X_test)
    print(classification_report(y_test , prediction))
    print(confusion_matrix(y_test , prediction))
```

Support Vector Machine

```
[64] X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.25, stratify = y, random_state=30)
     sv = SVC()
     sv.fit(X_train , y_train)
     prediction = sv.predict(X_test)
     print(classification_report(y_test , prediction))
     print(confusion_matrix(y_test , prediction))
```