Project Report on Wine Quality

USING SUPERVISED
AND
UNSUPERVISED LEARNING

SUBMITTED BY: GARGI SINGH TANWAR
SUBMITTED TO: DEBASHISH NANDY SIR
(CELEBAL CEO MENTOR)

Red Wine Quality Dataset

Input variables (based on physicochemical tests):

1 - fixed acidity

2 - volatile acidity

3 - citric acid

4 - residual sugar

5 - chlorides

6 - free sulfur dioxide

7 - total sulfur dioxide

8 - density

9 - pH

10 - sulphates

11 - alcohol

Output variable (based on sensory data):

12 - quality (score between 0 and 10)

Introduction

Finding the quality of wine is of significant importance for various reasons, both from the consumer's perspective and the wine industry as a whole. Here are some key reasons why it is essential to assess and understand the quality of wine:

1. Consumer Satisfaction
2. Market Differentiation
3. Consistency and Brand Reputation
4. Customer Retention

5. Economic Viability
6. Wine Ratings and Awards
7. Sustainable Practices
8. Wine Tourism
9. Industry Development and Innovation

Supervised Learning

The goal of supervised learning is to learn a mapping between the input features and the target labels so that the model can make accurate predictions on unseen data.

Here through supervised we are predicting result into either excellent quality or average quality of wine.

Load the Data

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
```

```
[ ] 1 data = pd.read_csv('winequality-red.csv')
    2 data
```

	fixed_acidity	volatile_acidity	citric_acid	residual_sugar	chlorides	free_sulfur_dioxide	total_sulfur_dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

Data Analysis

Data analysis is the process of inspecting, cleaning, transforming, and interpreting data to discover meaningful patterns, draw conclusions, and make informed decisions.

It is a crucial step in extracting valuable insights from raw data, enabling businesses, researchers, and individuals to better understand their data and use it to support various objectives.

Here are the results of data analysis

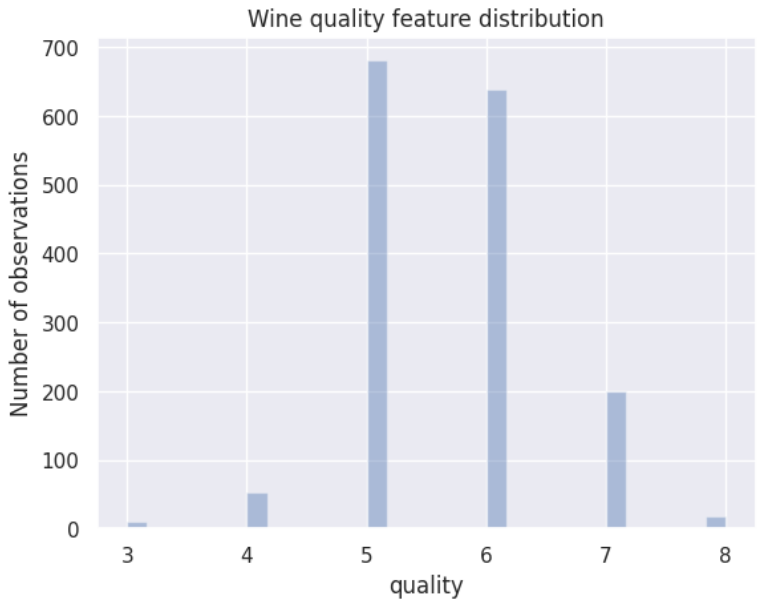
```
Are there missing observations the columns?
 fixed_acidity           False
volatile_acidity        False
citric_acid             False
residual_sugar          False
chlorides               False
free_sulfur_dioxide     False
total_sulfur_dioxide    False
density                 False
pH                      False
sulphates               False
alcohol                 False
quality                 False
dtype: bool

There are 6 Unique values for quality, namely: [3, 4, 5, 6, 7, 8]
```

```
13.57 % of the wines are of top tier quality
82.49 % of the wines are of average quality
3.94 % of the wines are below average quality
```

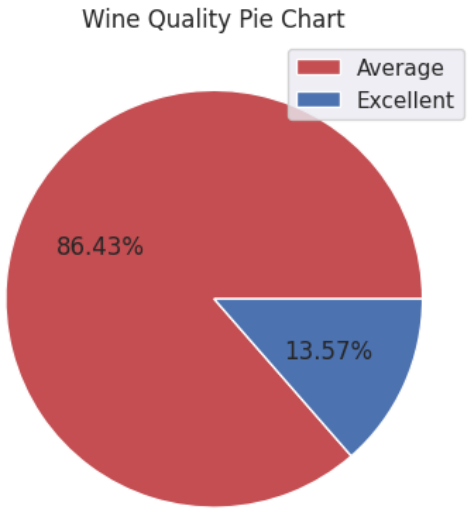
Visualizing and Understanding the Data

In dataset wine quality was divided into 1-10 rating

Show as below.

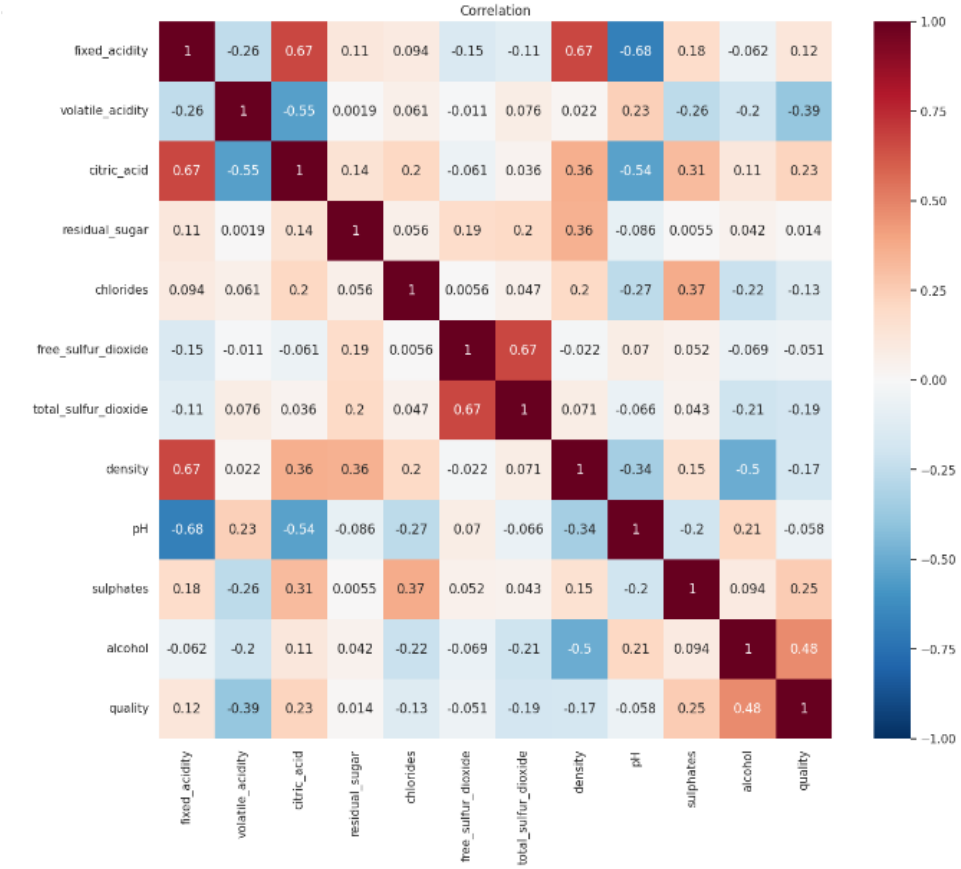


We analyze that there are 86.43% are of average quality and 13.57% are of excellent quality



Finding the Correlation Matrix

Here is the correlation matrix showing correlation of attribute with each other.



Splitting the Dataset

We are splitting the data 75% for training and 25% of testing

```
1 from sklearn.model_selection import train_test_split
2
3 X_train, X_test, y_train, y_test = train_test_split(XX, y, test_size=0.25,
4                                                     random_state=0, stratify=y)
```

```
1 cat = [cname for cname in XX.columns if XX[cname].dtype=='object']
2 print('There are %d columns with categorical entries\n' %len(cat))
```

```
There are 0 columns with categorical entries
```

Training Multiple Models using Supervised Learning Algorithms

For supervised learning I take Random Forest, Decision Tree and XGBoost Algorithms for implementations.

Comparing the three algorithms:

- Decision trees are simple and interpretable, but they can overfit and may not generalize well to new data.
- Random Forest reduces overfitting and increases accuracy by combining multiple decision trees, but it may be slower and less interpretable than a single decision tree.
- XGBoost provides superior performance compared to traditional gradient boosting algorithms by using optimization techniques, and it strikes a balance between accuracy, interpretability, and speed.

Evaluation the Models

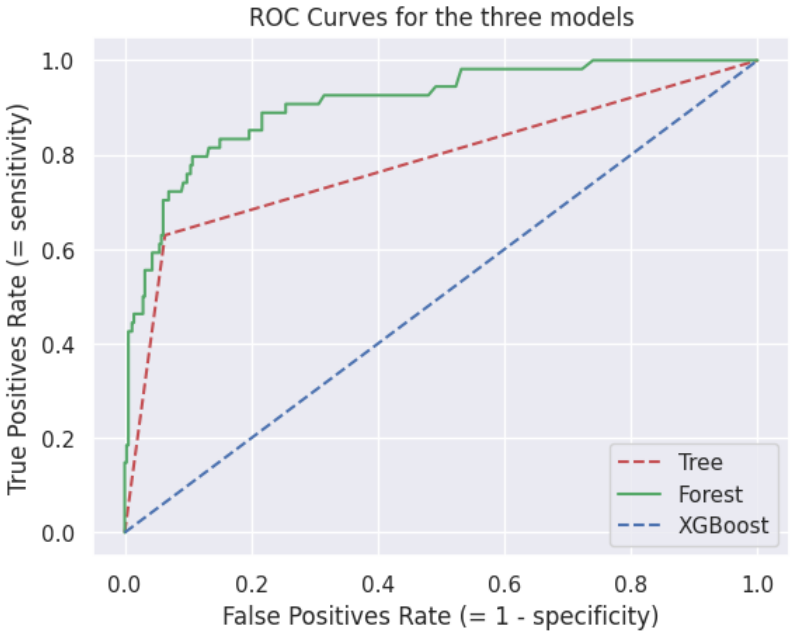
```
Accuracy: Decision Tree = 89.5%
Accuracy: Random Forest = 90.25%
Accuracy: xg boost = 91.0%
```

```
Confusion Matrix for simple, gradient boosted, and random forest tree classifiers:
Simple Tree:
 [[324  22]
 [ 20  34]]

Gradient Boosted:
 [[331  15]
 [ 24  30]]

Random Forest:
 [[327  19]
 [ 17  37]]
```

XGBoost is having the highest Accuracy.

The Receiver Operating Characteristic (ROC) curve is a graphical representation commonly used to evaluate the performance of binary classification models. It illustrates

the trade-off between the true positive rate (sensitivity or recall) and the false positive rate (1-specificity) across various probability thresholds for predicting the positive class.



Training the final best model and saving results with metrics.

Precision:

Precision is the number of true positive predictions divided by the total number of positive predictions made by the mod

Recall:

Recall, also known as sensitivity or true positive rate, is the number of true positive predictions divided by the total number of actual positive instances in the dataset. It measures the ability of the model to correctly identify positive instances. It measures the accuracy of positive predictions

F1 Score:

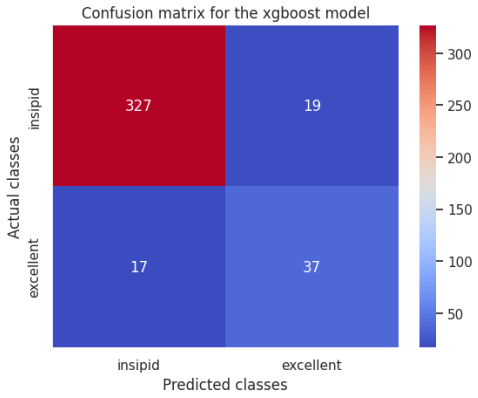
The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is useful when the class distribution is imbalanced.

Support:

Support refers to the number of instances in each class in the dataset. It helps to understand the distribution of different classes and is often used in imbalanced datasets.

Here are results for XGBoost

	precision	recall	f1-score	support
0	0.95	0.95	0.95	346
1	0.66	0.69	0.67	54
accuracy			0.91	400
macro avg	0.81	0.82	0.81	400
weighted avg	0.91	0.91	0.91	400



Model Deployment using flask.

```
fixed_acidity =  7.4#@param {type:"number"}
volatile_acidity  = 0.7#@param {type:"number"}
citric_acid =  0#@param {type:"number"}
residual_sugar =  1.9#@param {type:"number"}
chlorides =  0.076#@param {type:"number"}
free_sulfur_dioxide =  11#@param {type:"number"}
total_sulfur_dioxide =  34#@param {type:"number"}
density =  0.9968#@param {type:"number"}
pH =  3.51#@param {type:"number"}
sulphates =  0.56#@param {type:"number"}
alcohol =  9.4#@param {type:"number"}


output= classifier_major_rf.predict([[fixed_acidity, volatile_acidity, citric_acid, residual_sugar, chlorides,
print( output)
if output==[0]:
  print("Average Quality")
else:
  print("Excellent Quality")

```

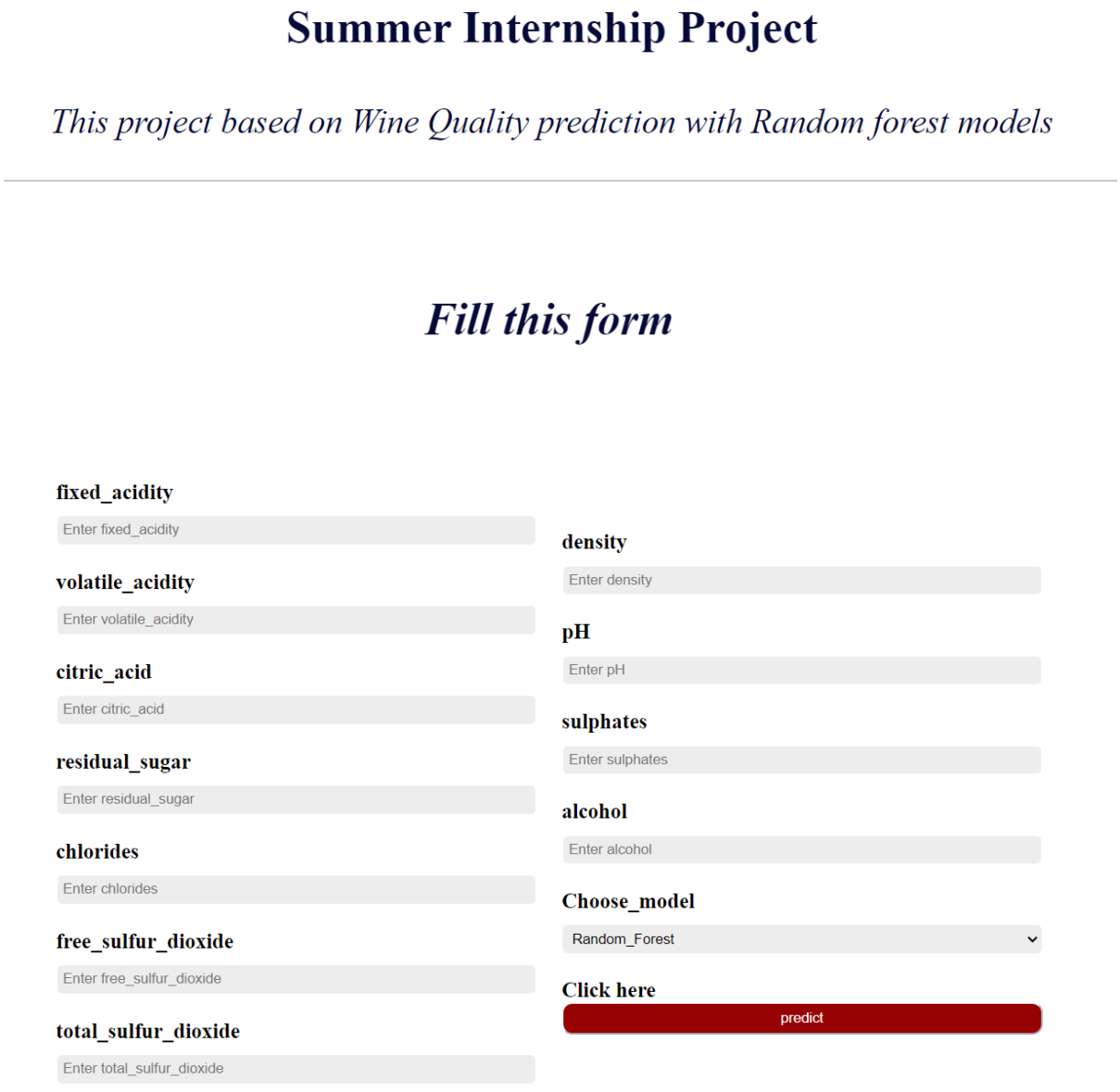
fixed_acidity: 7.4
volatile_acidity: 0.7
citric_acid: 0
residual_sugar: 1.9
chlorides: 0.076
free_sulfur_dioxide: 11
total_sulfur_dioxide: 34
density: 0.9968
pH: 3.51
sulphates: 0.56
alcohol: 9.4

```
[0]
Average Quality
```

Here we are getting 2 cluster Average and Excellent quality of dataset.

Here its showing average for the given data point.

GUI on local host testing



Unsupervised Learning

The goal of unsupervised learning is to identify patterns, relationships, and structures within the data without guidance or predefined labels.

For this I am using Kmeans and Hierarchical clustering Algorithms for finding quality of winedata on basis of Alcohol and amount of sulphate present in it.

K-Means and Hierarchical Clustering are two popular algorithms used for clustering, a task in unsupervised learning where the goal is to group similar data points together based on their similarity.

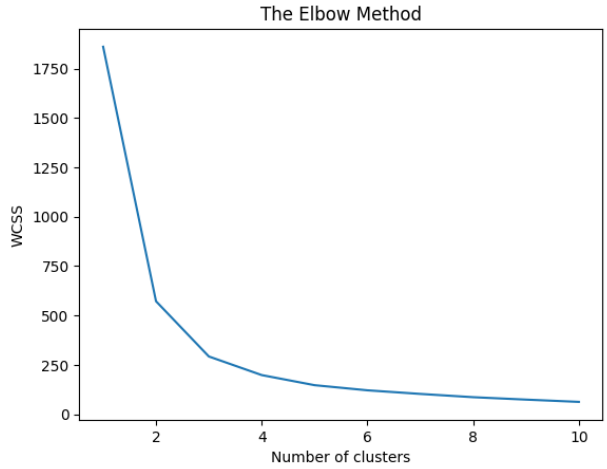
K-Means Clustering:

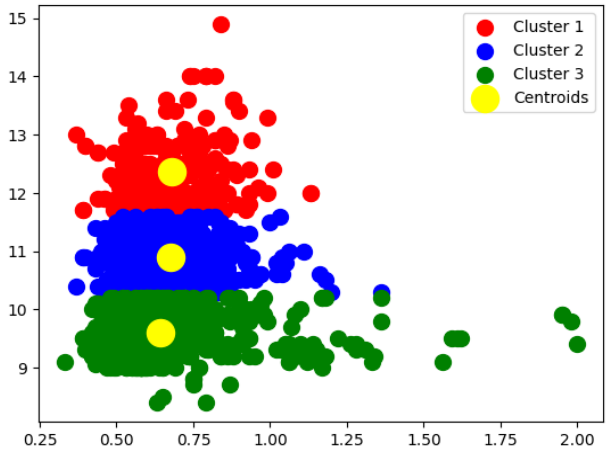
- K-Means is a partition-based clustering algorithm that assigns data points to 'k' clusters, where 'k' is a user-defined parameter representing the number of clusters desired.
- The algorithm starts by randomly initializing 'k' cluster centroids (points that represent the center of each cluster).
- It then iteratively assigns each data point to the nearest centroid based on a distance metric (usually Euclidean distance) and updates the centroids' positions.
- This process continues until convergence, where the centroids stop changing or after a specified number of iterations.
- K-Means aims to minimize the within-cluster sum of squared distances (inertia) from each data point to its assigned centroid.

Visualizing and Understanding the Data for Kmeans

Using WCSS method for finding proper number of clusters

We are getting 3 as an appropriate value





Here 3 clusters are shown describing average, good, and excellent quality of wine.

```
sulphates =  1#@param {type:"number"}
alcohol = 9 #@param {type:"number"}

predict= kmeans.predict([[ sulphates,alcohol ]])
print(predict)
if predict==[0]:
  print("Average quality")
elif predict==[1]:
  print("Good quality")
elif predict==[2]:
  print("Excelent quality")


```

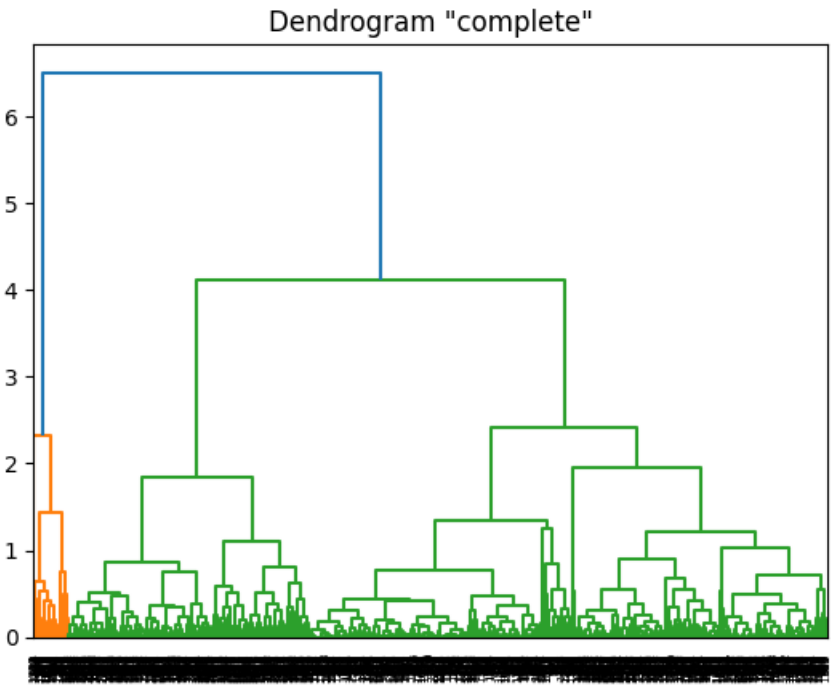
sulphates: 1

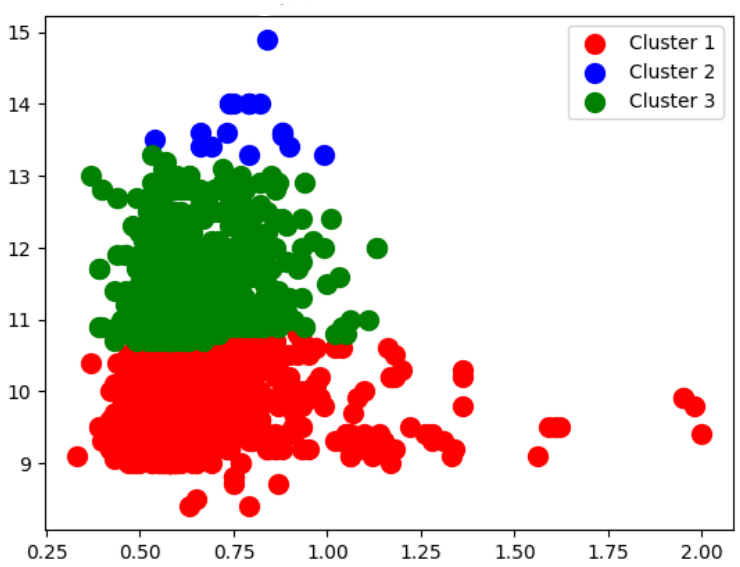
alcohol: 9

```
[2]
Excelent quality
```

Hierarchical Clustering:

- Hierarchical Clustering is a bottom-up (agglomerative) or top-down (divisive) clustering algorithm that creates a hierarchy of clusters represented in the form of a dendrogram.
- In agglomerative hierarchical clustering, each data point starts as its own cluster, and the algorithm repeatedly merges the closest clusters until all data points belong to a single cluster.
- In divisive hierarchical clustering, all data points start in one cluster, and the algorithm recursively divides clusters until each data point is in its own cluster.
- The decision to merge or split clusters is based on a distance or linkage metric that defines how to measure the similarity between clusters.
- The result is a dendrogram that visualizes the hierarchical relationships between clusters, and the desired number of clusters can be determined by cutting the dendrogram at a certain height.





In hierarchical clustering, we take multiple inputs because this clustering algorithm works on a set of data points, where each data point represents an individual sample or observation. The primary goal of hierarchical clustering is to group similar data points together based on some similarity measure or distance metric.

When using hierarchical clustering, we need to consider multiple data points simultaneously to form the clusters. The algorithm starts by treating each data point as its own cluster and then iteratively merges the closest clusters until all data points are grouped into a single cluster or a predefined number of clusters is achieved.

```
sulphates =  39#@param {type:"number"}
alcohol =  91#@param {type:"number"}
sulphates1 =  34#@param {type:"number"}
alcohol1 =  19#@param {type:"number"}
sulphates2 = 34 #@param {type:"number"}
alcohol2 =  65#@param {type:"number"}

predict= hc.fit_predict([[ sulphates,alcohol ],|
print(predict)
```

sulphates: 39

alcohol: 91

sulphates1: 34

alcohol1: 19

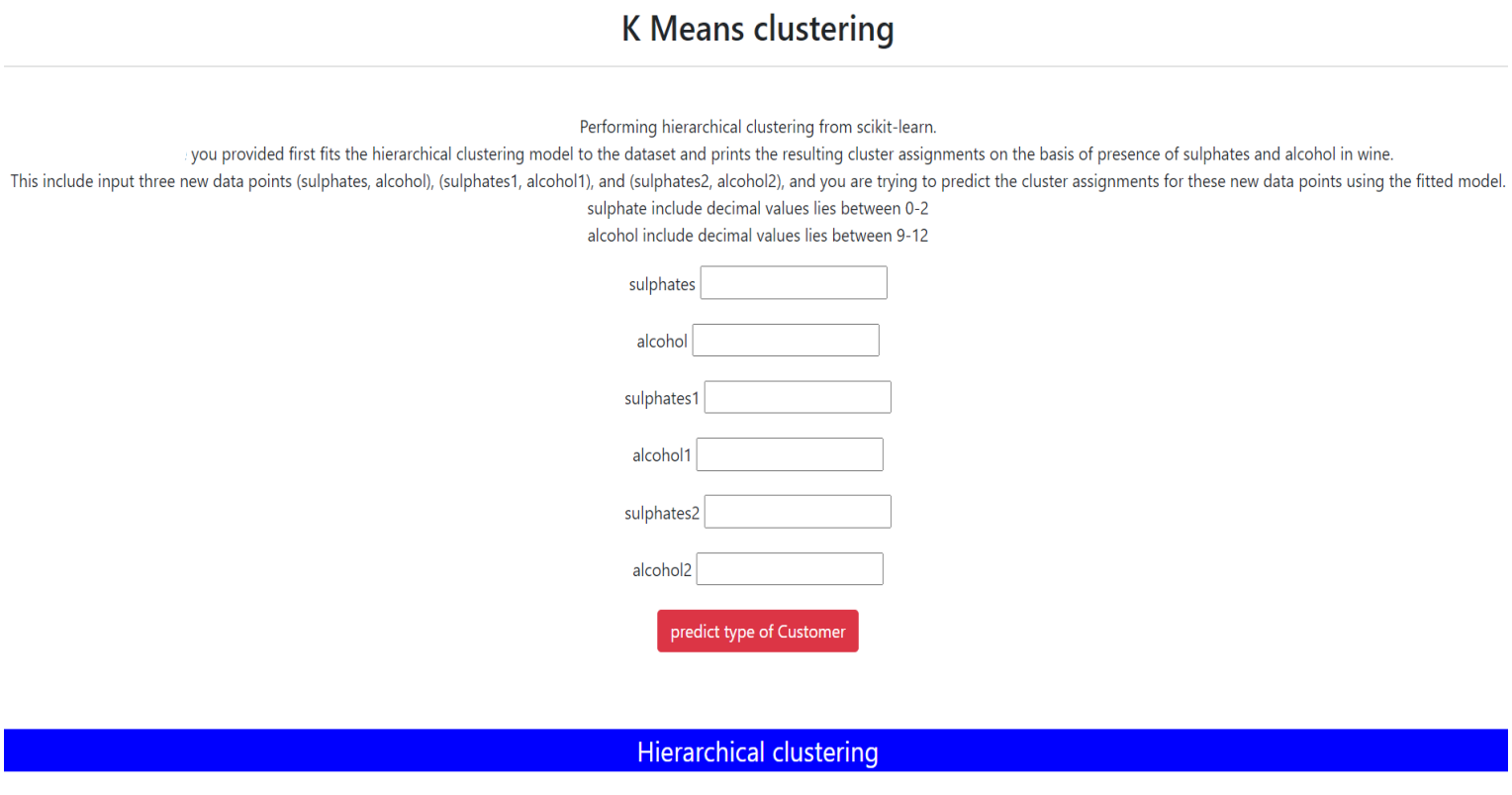
sulphates2: 34

alcohol2: 65

```
[2 1 0]
```

Model Deployment using flask.

GUI on local host testing



Conclusion

After conducting extensive analyses on the wine dataset using both supervised and unsupervised learning techniques, including Decision Tree, Random Forest, XGBoost, KMeans, and Hierarchical Clustering, we have derived valuable insights and generated predictive models. Here are the key conclusions drawn from the study:

1. Supervised Learning: a. Decision Tree, Random Forest, and XGBoost:

- These algorithms demonstrated good performance in predicting wine quality based on the available features. Their ability to handle numerical data made them suitable for this task.
- Among the three models, Random Forest and XGBoost outperformed Decision Tree, indicating the importance of ensemble methods in improving predictive accuracy.
- XGBoost is giving the highest accuracy

2. Unsupervised Learning: a. KMeans:

- The KMeans algorithm efficiently grouped wines into distinct clusters based on their similar attributes, helping to identify natural patterns and segments within the data.
- The optimal number of clusters was determined using various evaluation metrics such as the elbow method and silhouette score.

b. Hierarchical Clustering:

- Hierarchical Clustering allowed us to create a dendrogram that showcased the hierarchical relationships between wines, offering insights into their similarities and differences.
- Different linkage methods and distance metrics were explored to determine the most suitable clustering structure.

In summary, the combination of supervised and unsupervised learning techniques provided a comprehensive understanding of the wine dataset. The predictive models created using Decision Tree, Random Forest, and XGBoost can be utilized for future wine quality predictions with reasonable accuracy. On the other hand, the insights obtained from KMeans and Hierarchical Clustering allow for a better understanding of wine groups and relationships, assisting in market segmentation and targeted marketing strategies.

As with any data analysis, it is essential to consider the specific requirements and objectives of the task at hand when selecting the appropriate machine learning algorithms. Furthermore, continuous evaluation and refinement of the models will be necessary as new data becomes available, ensuring the relevance and accuracy of the predictions and clustering results.