Wine Quality Analysis & Prediction

Problem Statement:

Developing a machine learning model for predicting the quality of Portuguese Vinho Verde wine based on its chemical properties, using a cloud-based infrastructure to enable scalable and efficient training and deployment. The project aims to analyse and predict the quality of wines based on various physicochemical properties using the dataset "Wine Quality" obtained from Kaggle.

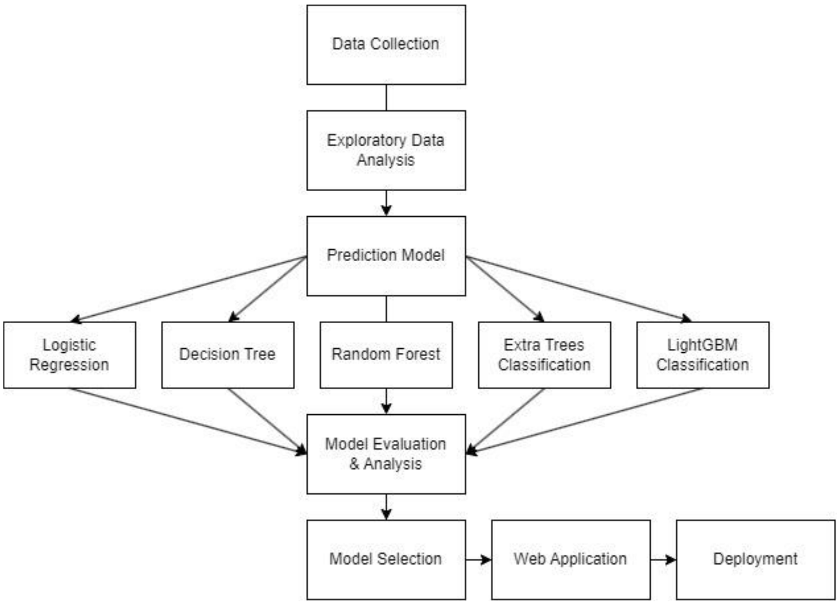
Objective:

The objective of this project is to perform a comprehensive analysis of the dataset obtained from Kaggle, and develop a predictive model that accurately predicts the quality rating of wines based on their physicochemical properties. Specific objectives of this project include conducting Exploratory Data Analysis by performing a thorough exploration of the dataset to gain insights into the distribution, relationships, and patterns among the different physicochemical properties and the quality ratings of wines. The project aims to develop a predictive model by employing various machine learning algorithms like Logistic Regression, Decision Tree Classification, Random Forest Classification, Extra Trees Classification and LGBM Classification and choose the best model that accurately predicts the quality rating of wines. The model will be trained and validated using appropriate techniques such as cross-validation and performance evaluation metrics. Further objectives include deploying a web application for real-time interaction with the model, created using Python Flask and hosted using AWS Platform.

Introduction:

The wine industry has always been a fascinating blend of art and science, where the quality of the final product is influenced by a myriad of factors. Winemakers strive to produce wines that not only captivate the senses but also satisfy the discerning palates of consumers. In this context, the analysis and prediction of wine quality based on physicochemical properties have gained significant attention. Understanding the intricate relationship between these properties and the perceived quality of wines can provide valuable insights for winemakers and consumers alike. This project focuses on the analysis and prediction of wine quality using the "Wine Quality" dataset obtained from Kaggle. The dataset contains information on various physicochemical properties such as acidity, pH, residual sugar, alcohol content, and more, along with corresponding quality ratings assigned by experts. We aim to develop a predictive model capable of accurately forecasting the quality rating of wines. The outcome of this project will provide actionable insights for winemakers to optimize their production processes and improve the overall quality of their wines. Additionally, consumers will benefit from a better understanding of the physicochemical properties that influence wine quality, empowering them to make informed decisions when selecting wines according to their preferences.

Architecture:



Module description:

Data Collection:

The Wine Quality Dataset is downloaded from Kaggle and is stored in the local machine. The required libraries are then installed and imported Dataset Exploration is done to understand its structures, attributes and format. The dataset is then loaded and pre-processed to remove null and missing values. The data is validated to ensure its integrity and consistency.

Exploratory Data Analysis:

Exploratory Data Analysis (EDA) is a crucial step in understanding the characteristics and patterns within a dataset. In the context of the "Wine Quality" dataset, conducting EDA allows us to gain insights into the physicochemical properties of wines and their corresponding quality ratings. Summary statistics such as mean, median and mode are calculated to get a high level overview of the dataset. Bivariate analysis is done between all the columns in the dataset, and the interdependency of attributes is understood. Scatter plots, box plots, pair plots, bar graphs are constructed for the attributes and are visualized.

Building a Prediction Model:

The main objective for this project is to build an accurate predictive model which can assess the quality of wine by analysing its physiochemical attributes. Machine Learning algorithms which were chosen to build a predictive model include Logistic Regression, Decision Tree Classification, Random Forest Classification, Extra Trees Classification and LGBM Classification. The accuracy scores have been calculated and the cross validation scores of the models are compared as well. Moreover the classification reports for the test data were analysed for each model and the predicted quality values were plotted on scatter plots as well.

Web Application:

The best machine learning model is compared and chosen. An interactive web application has been implemented using the Python Flask framework. The user is allowed to enter the physiochemical values of the wine to be tested and using the selected model, the quality of the wine is displayed to the user. The application has been deployed in the cloud through an EC2 instance in the AWS platform.

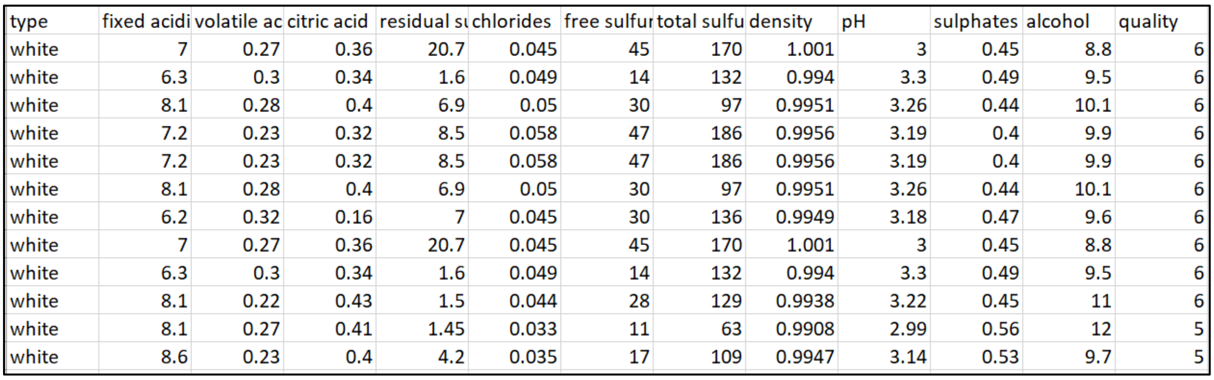
Dataset:

The "Wine Quality" dataset, available on Kaggle, provides valuable information about the physicochemical properties of wines and their corresponding quality ratings. It offers insights into the factors that contribute to the overall quality of wines and allows for analysis and prediction tasks. The dataset can be accessed at the following link: https://www.kaggle.com/datasets/rajyellow46/wine-quality. It is a publicly available dataset contributed by Rajdeep Chatterjee. The dataset contains a total of 6,497 instances or samples of wines. The attributes in the dataset are mentioned below:

- Physicochemical properties
  - ✓ Wine type (Red/White)
  - ✓ Fixed acidity
  - ✓ Volatile acidity
  - ✓ Citric acid
  - ✓ Residual sugar
  - ✓ Chlorides
  - ✓ Free sulfur dioxide
  - ✓ Total sulfur dioxide
  - ✓ Density
  - ✓ pH
  - ✓ Sulphates
  - ✓ Alcohol

- Quality Rating:

The quality rating is provided as an integer value ranging from 3 to 9. Higher ratings indicate better quality wines, while lower ratings suggest wines of lower quality.



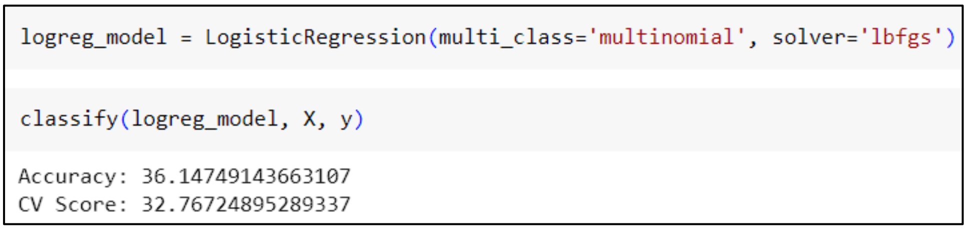
type	fixed acidi	volatile ac	citric acid	residual su	chlorides	free sulfur	total sulfu	density	pH	sulphates	alcohol	quality
white	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
white	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
white	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
white	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
white	7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.4	9.9	6
white	8.1	0.28	0.4	6.9	0.05	30	97	0.9951	3.26	0.44	10.1	6
white	6.2	0.32	0.16	7	0.045	30	136	0.9949	3.18	0.47	9.6	6
white	7	0.27	0.36	20.7	0.045	45	170	1.001	3	0.45	8.8	6
white	6.3	0.3	0.34	1.6	0.049	14	132	0.994	3.3	0.49	9.5	6
white	8.1	0.22	0.43	1.5	0.044	28	129	0.9938	3.22	0.45	11	6
white	8.1	0.27	0.41	1.45	0.033	11	63	0.9908	2.99	0.56	12	5
white	8.6	0.23	0.4	4.2	0.035	17	109	0.9947	3.14	0.53	9.7	5

Results & Outcome:

The individual model performances for all the proposed machine learning models has been discussed below. The accuracy scores and cross validation scores were measured with randomized data. To show comparison between the 5 models, some test data has been used to build classification reports, confusion matrices and visualization. The same test data has been used for all 5 models.

Logistic Regression:

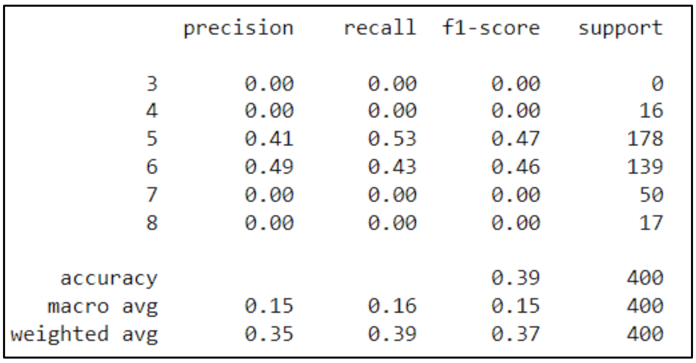
A multiclass logistic regression model was implemented using the ‘lbfgs’ solver.



```
logreg_model = LogisticRegression(multi_class='multinomial', solver='lbfgs')

classify(logreg_model, X, y)

Accuracy: 36.14749143663107
CV Score: 32.76724895289337
```

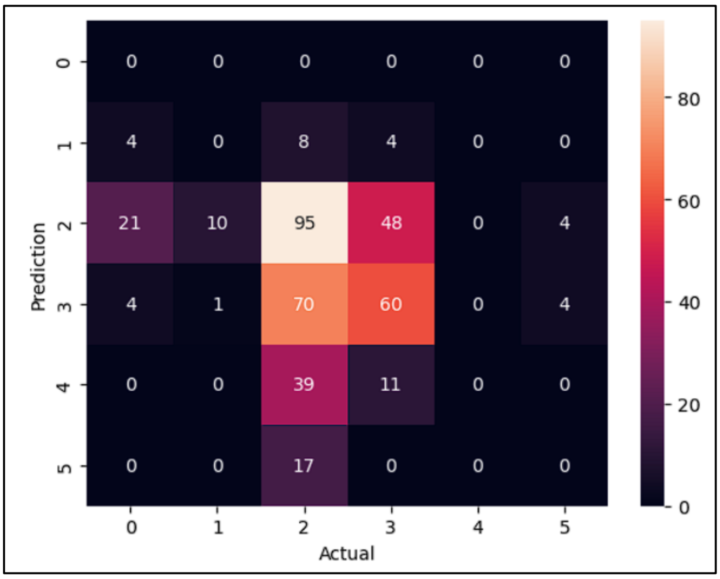
The model performed with a very low accuracy score and an even lower cross validation score.

The classification report for the test values is as follows:



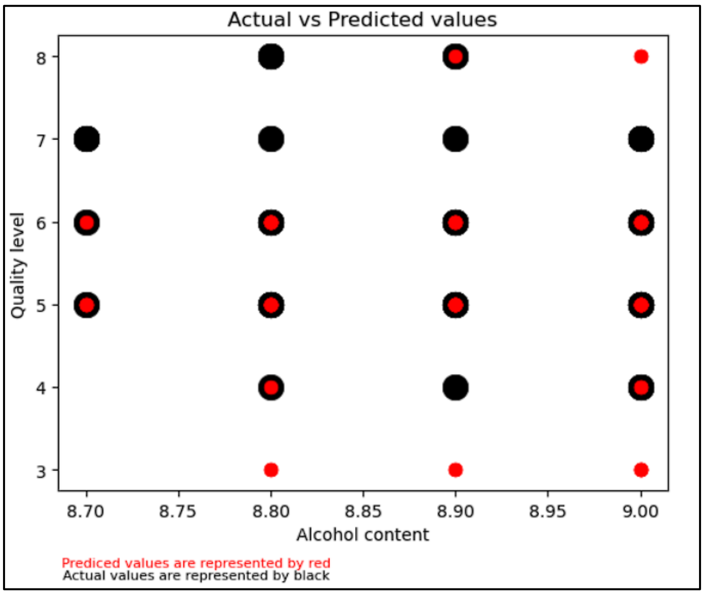
```
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00        16
           5       0.41      0.53      0.47       178
           6       0.49      0.43      0.46       139
           7       0.00      0.00      0.00        50
           8       0.00      0.00      0.00        17

    accuracy                           0.39       400
   macro avg       0.15      0.16      0.15       400
weighted avg       0.35      0.39      0.37       400
```

And the confusion matrix is shown below:



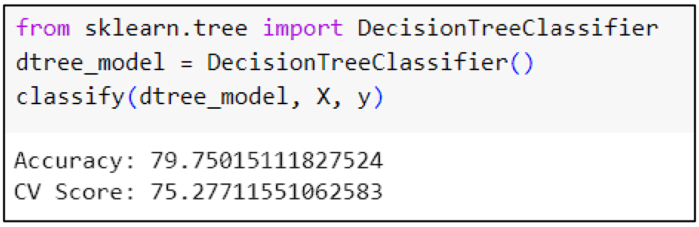
	0	1	2	3	4	5
0	0	0	0	0	0	0
1	4	0	8	4	0	0
2	21	10	95	48	0	4
3	4	1	70	60	0	4
4	0	0	39	11	0	0
5	0	0	17	0	0	0

A scatter plot has been constructed to visualize the actual and predicted values:



Decision Tree Classifier:

A Decision Tree Classification model was implemented using the Scikit-learn library.

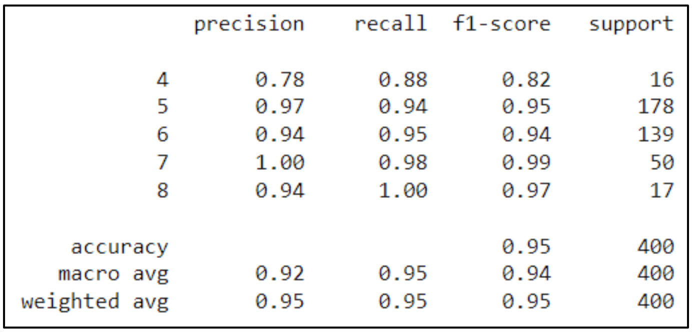


```
from sklearn.tree import DecisionTreeClassifier
dtree_model = DecisionTreeClassifier()
classify(dtree_model, X, y)
```

```
Accuracy: 79.75015111827524
CV Score: 75.27711551062583
```

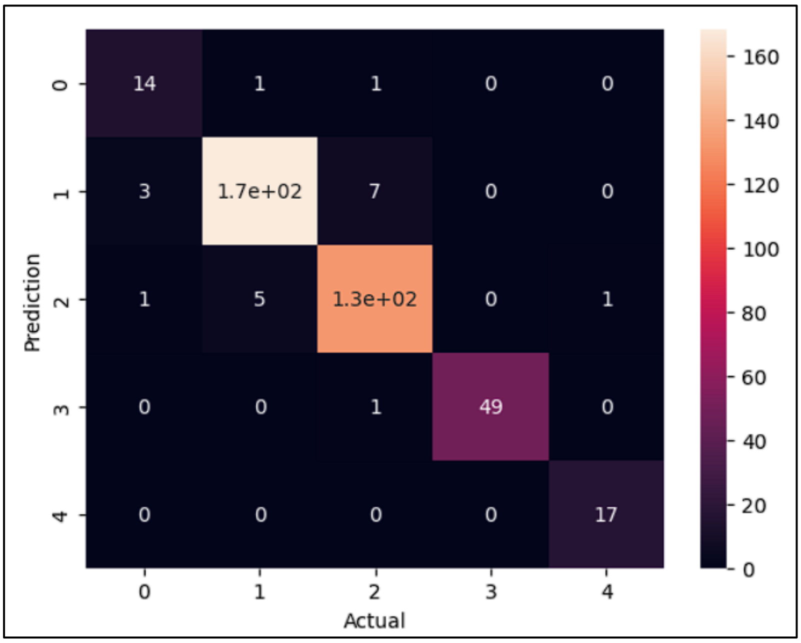
The model performed with an accuracy score and validation score.

The classification report for the test values is as follows:

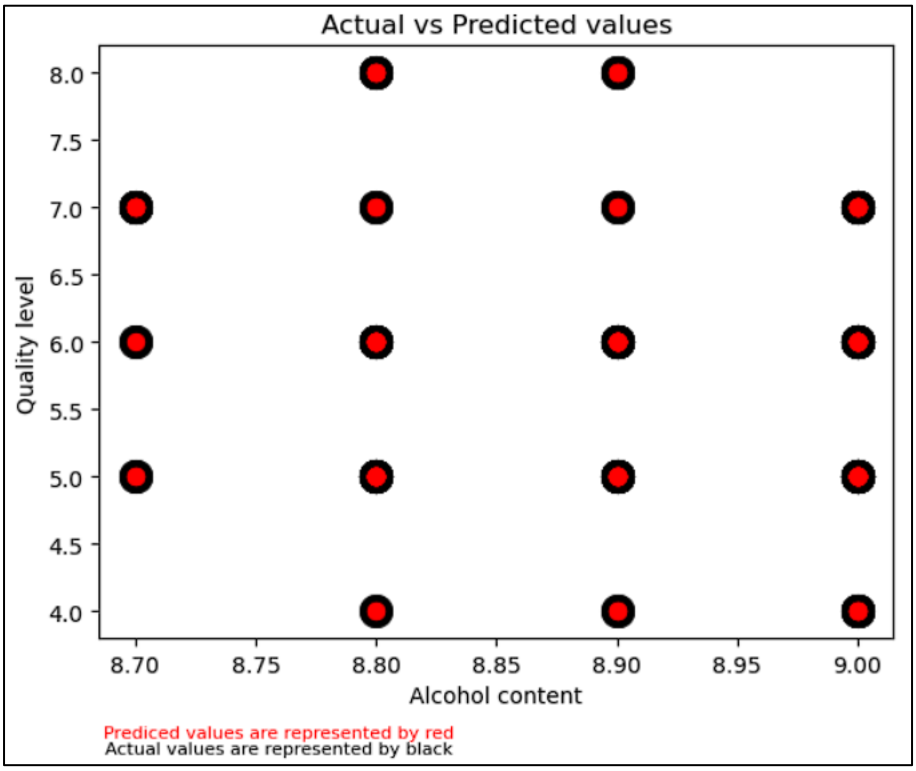


```
              precision    recall  f1-score   support

           4       0.78      0.88      0.82        16
           5       0.97      0.94      0.95       178
           6       0.94      0.95      0.94       139
           7       1.00      0.98      0.99        50
           8       0.94      1.00      0.97        17

    accuracy                           0.95       400
   macro avg       0.92      0.95      0.94       400
weighted avg       0.95      0.95      0.95       400
```

And the confusion matrix is shown below:

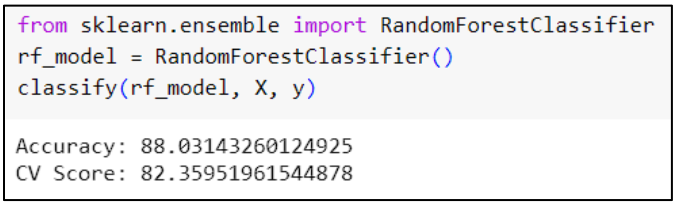


A scatter plot has been constructed to visualize the actual and predicted values:



Random Forest Classifier:

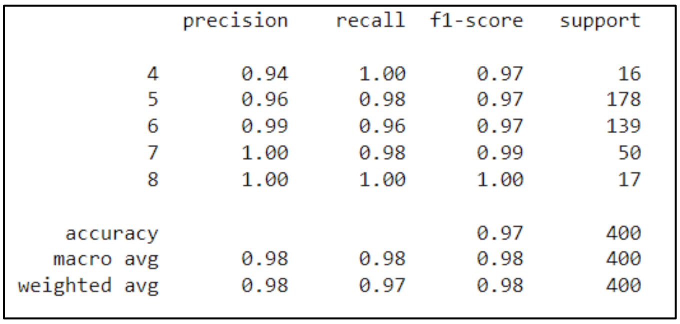
A Random Forest Classification model was implemented using the Scikit-learn library. Default arguments were used to build the model.



```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier()
classify(rf_model, X, y)

Accuracy: 88.03143260124925
CV Score: 82.35951961544878
```

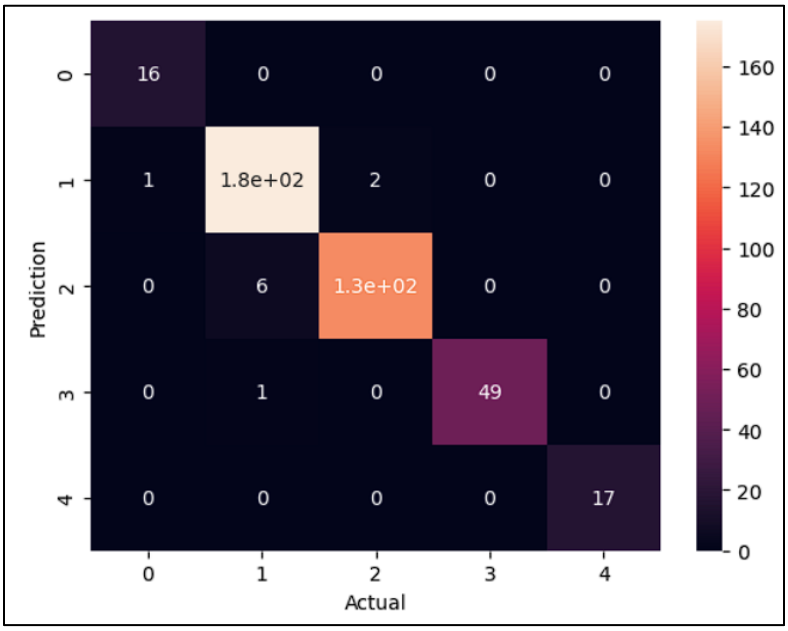
The model performed with an accuracy score and validation score.

The classification report for the test values is as follows:

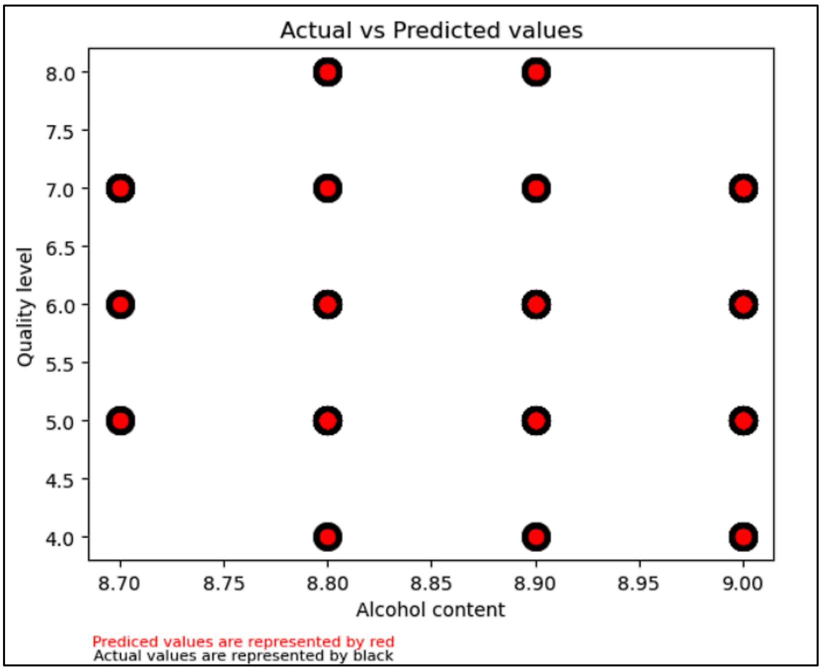


```
              precision    recall  f1-score   support

           4       0.94      1.00      0.97        16
           5       0.96      0.98      0.97       178
           6       0.99      0.96      0.97       139
           7       1.00      0.98      0.99        50
           8       1.00      1.00      1.00        17

    accuracy                           0.97       400
   macro avg       0.98      0.98      0.98       400
weighted avg       0.98      0.97      0.98       400
```

And the confusion matrix is shown below:

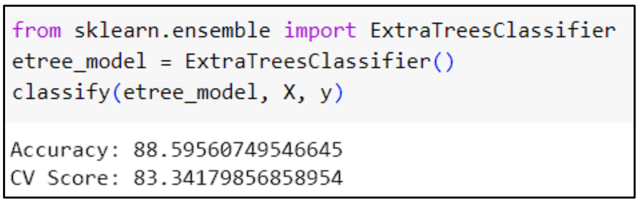


A scatter plot has been constructed to visualize the actual and predicted values:



Extra Trees Classifier:

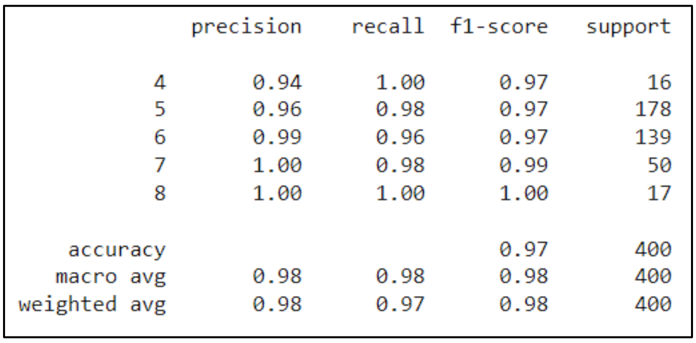
Extra trees (short for extremely randomized trees) is an ensemble supervised machine learning method that uses decision trees and is used by the Train Using AutoML tool. It has been implemented using the Scikit-learn library.



```
from sklearn.ensemble import ExtraTreesClassifier
etree_model = ExtraTreesClassifier()
classify(etree_model, X, y)

Accuracy: 88.59560749546645
CV Score: 83.34179856858954
```

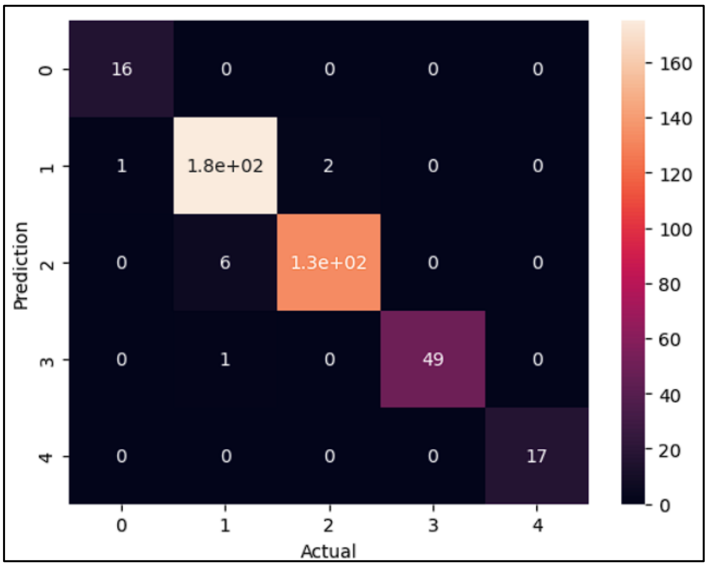
The model performed with an accuracy score and validation score.

The classification report for the test values is as follows:

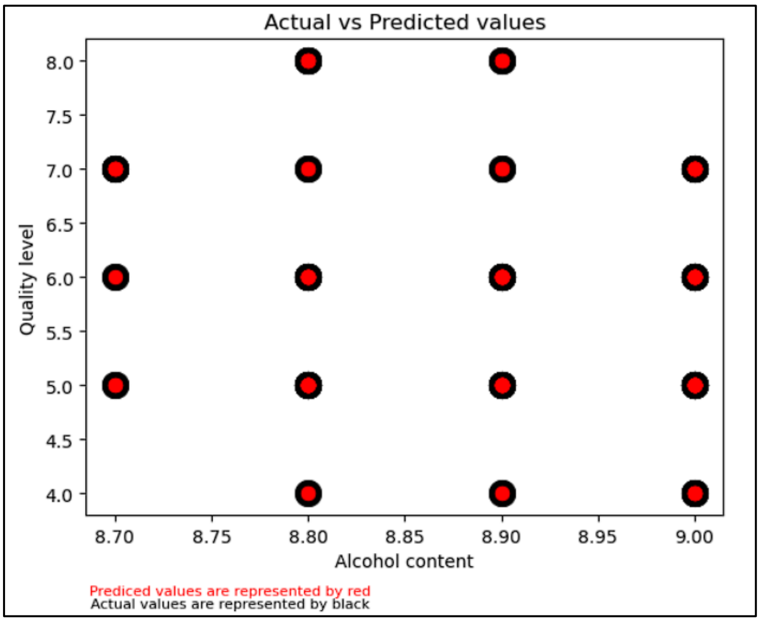


```
              precision    recall  f1-score   support

           4       0.94      1.00      0.97        16
           5       0.96      0.98      0.97       178
           6       0.99      0.96      0.97       139
           7       1.00      0.98      0.99        50
           8       1.00      1.00      1.00        17

    accuracy                           0.97       400
   macro avg       0.98      0.98      0.98       400
weighted avg       0.98      0.97      0.98       400
```

And the confusion matrix is shown below:

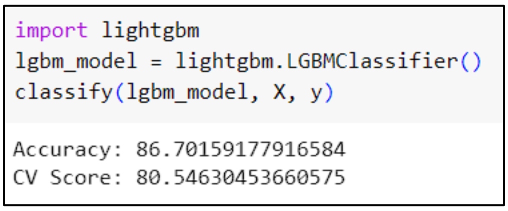


A scatter plot has been constructed to visualize the actual and predicted values:



LightGBM Classifier:

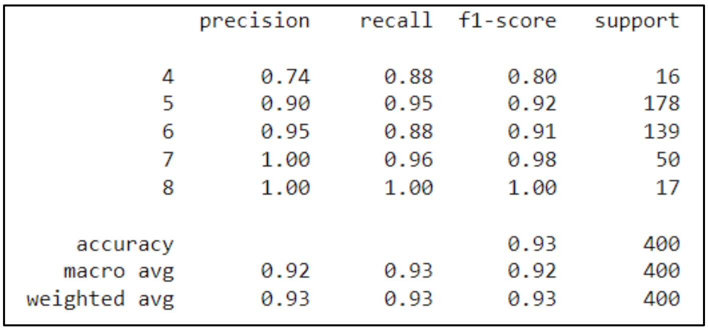
LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient. Scikit-learn library has been used to implement the algorithm.



```
import lightgbm
lgbm_model = lightgbm.LGBMClassifier()
classify(lgbm_model, X, y)

Accuracy: 86.70159177916584
CV Score: 80.54630453660575
```

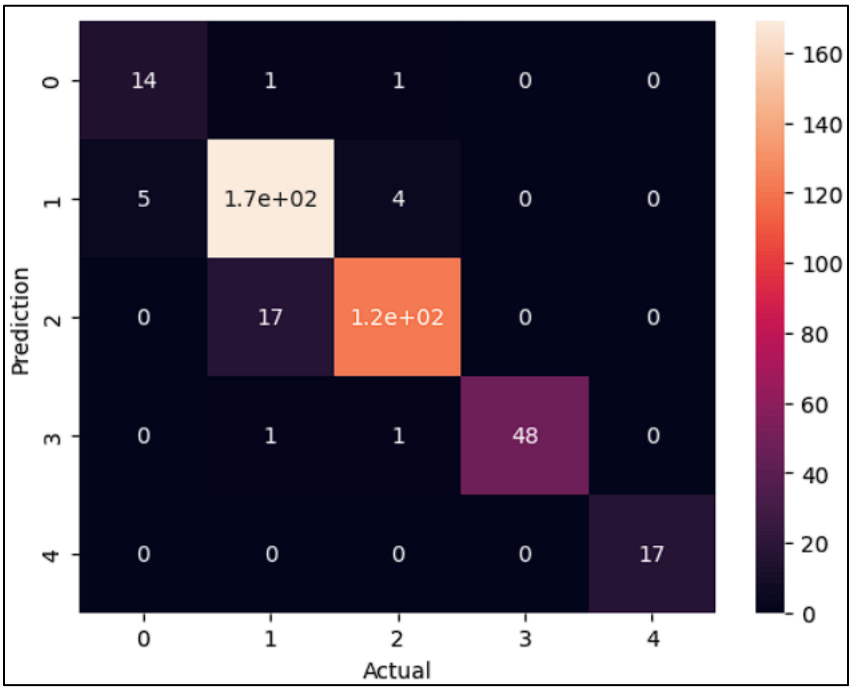
The model performed with an accuracy score and validation score.

The classification report for the test values is as follows:

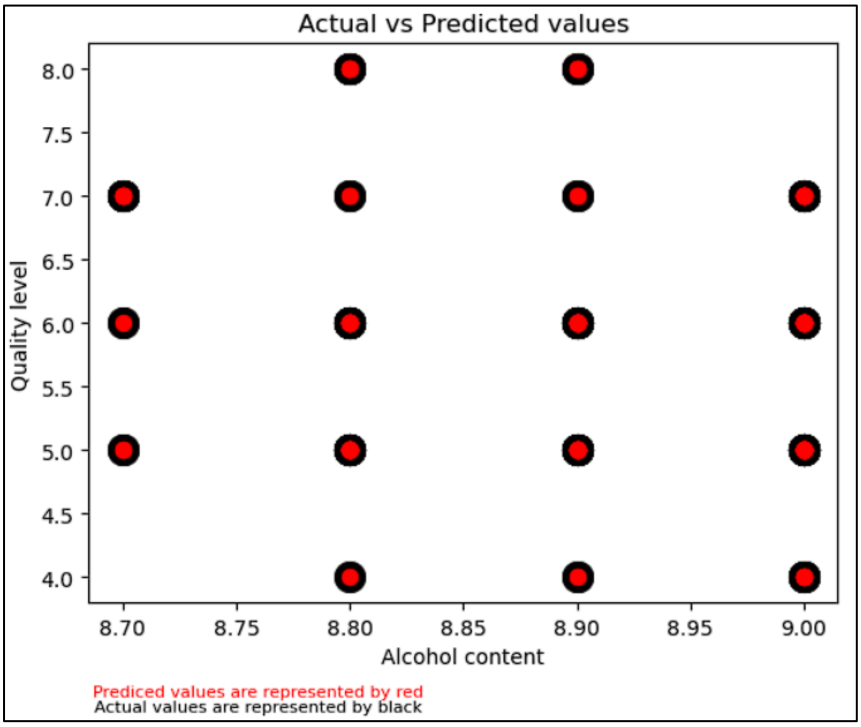


```
              precision    recall  f1-score   support

           4       0.74      0.88      0.80        16
           5       0.90      0.95      0.92       178
           6       0.95      0.88      0.91       139
           7       1.00      0.96      0.98        50
           8       1.00      1.00      1.00        17

    accuracy                           0.93       400
   macro avg       0.92      0.93      0.92       400
weighted avg       0.93      0.93      0.93       400
```

And the confusion matrix is shown below:

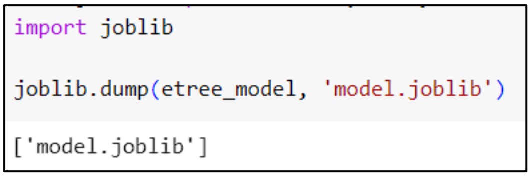


A scatter plot has been constructed to visualize the actual and predicted values:



Model Selection:

The extra trees classification model has performed with a accuracy score of 88.59 and cross validation score of 88.34. It has also achieved and accuracy of 97% in the test data considered. The extra trees model is saved using the ‘joblib’ library.



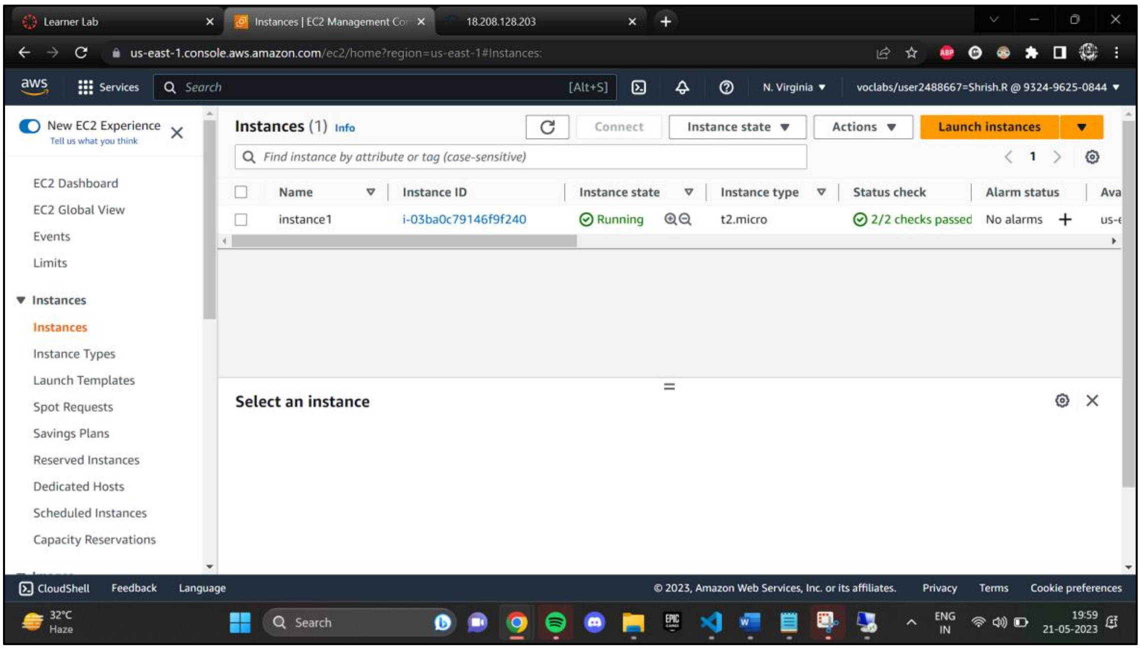
```
import joblib

joblib.dump(etree_model, 'model.joblib')

['model.joblib']
```

Web Application:

.

An interactive web application has been developed using the selected model. The Flask framework has been used to predict the wine quality by analysing the physiochemical properties of the wine presented. The user is able to interact with the system using the frontend user interface developed using HTML and CSS. The application has been deployed using EC2 instance through the AWS platform.

Screenshots:



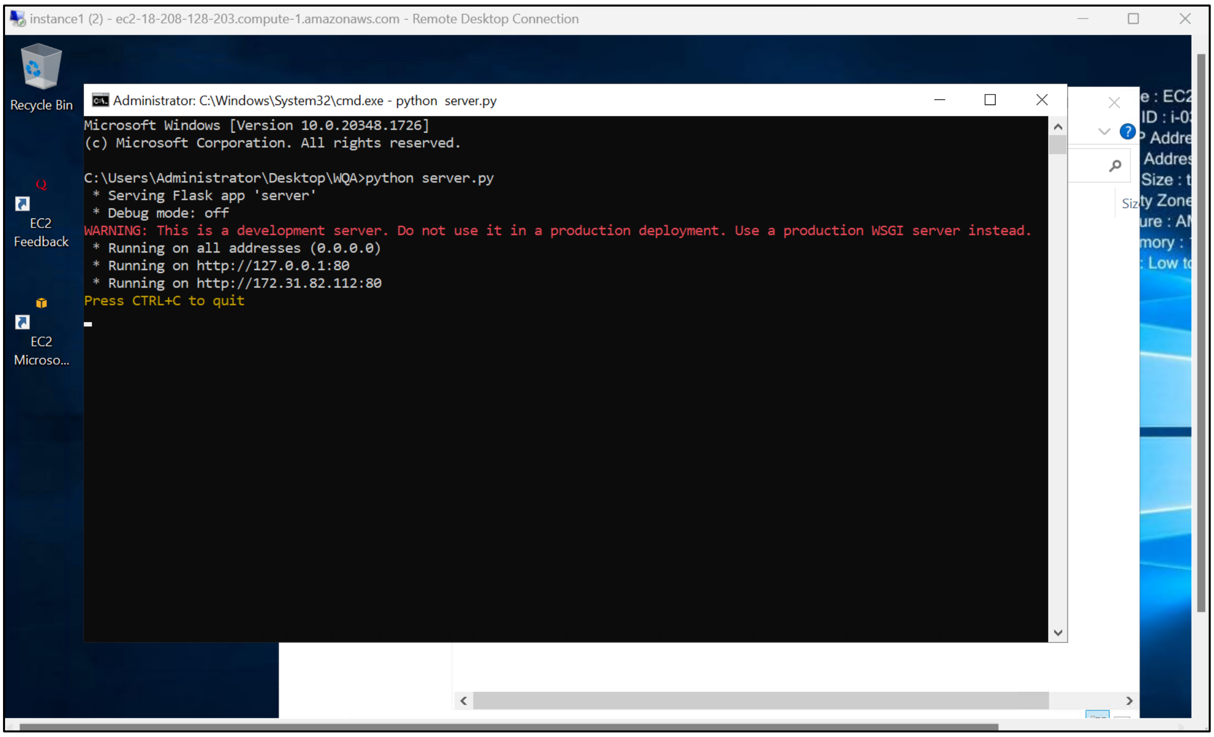
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

voclabs/user2488667=Shrish.R @ 9324-9625-0844

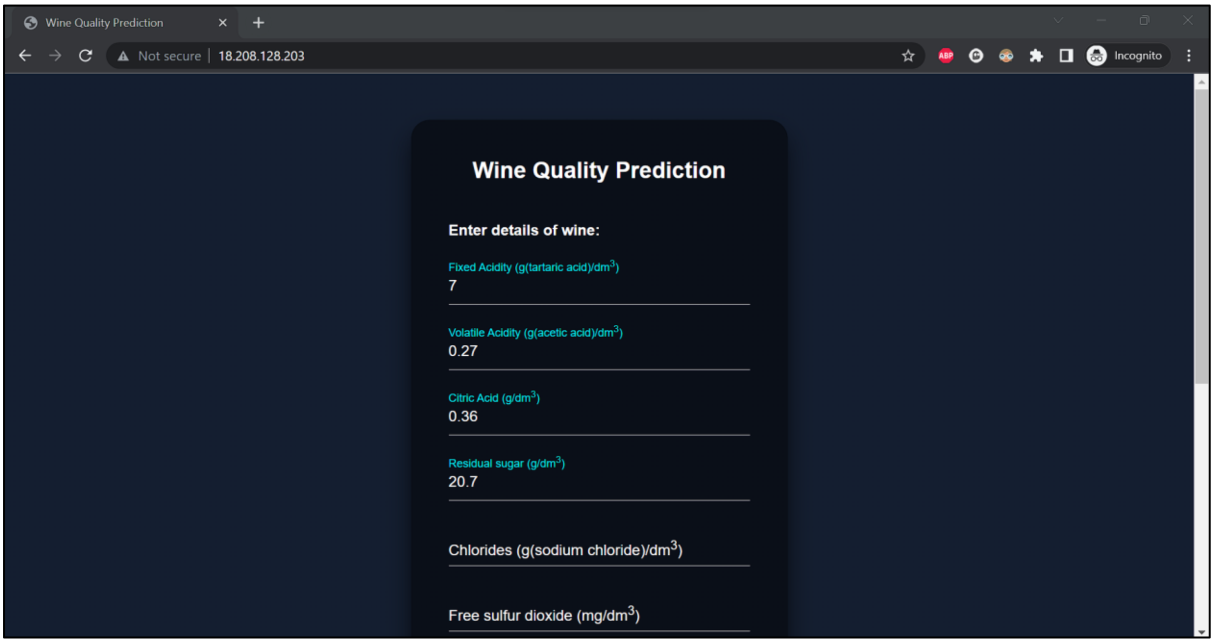
Instances (1)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	instance1	i-03ba0c79146f9f240	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	us-e

Select an instance


instance1 (2) - ec2-18-208-128-203.compute-1.amazonaws.com - Remote Desktop Connection
Recycle Bin
EC2 Feedback
EC2 Microso...
Administrator: C:\Windows\System32\cmd.exe - python server.py

```
Microsoft Windows [Version 10.0.20348.1726]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\WQA>python server.py
 * Serving Flask app 'server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:80
 * Running on http://172.31.82.112:80
Press CTRL+C to quit
```



Wine Quality Prediction

Enter details of wine:

Fixed Acidity (g(tartaric acid)/dm^3)
7

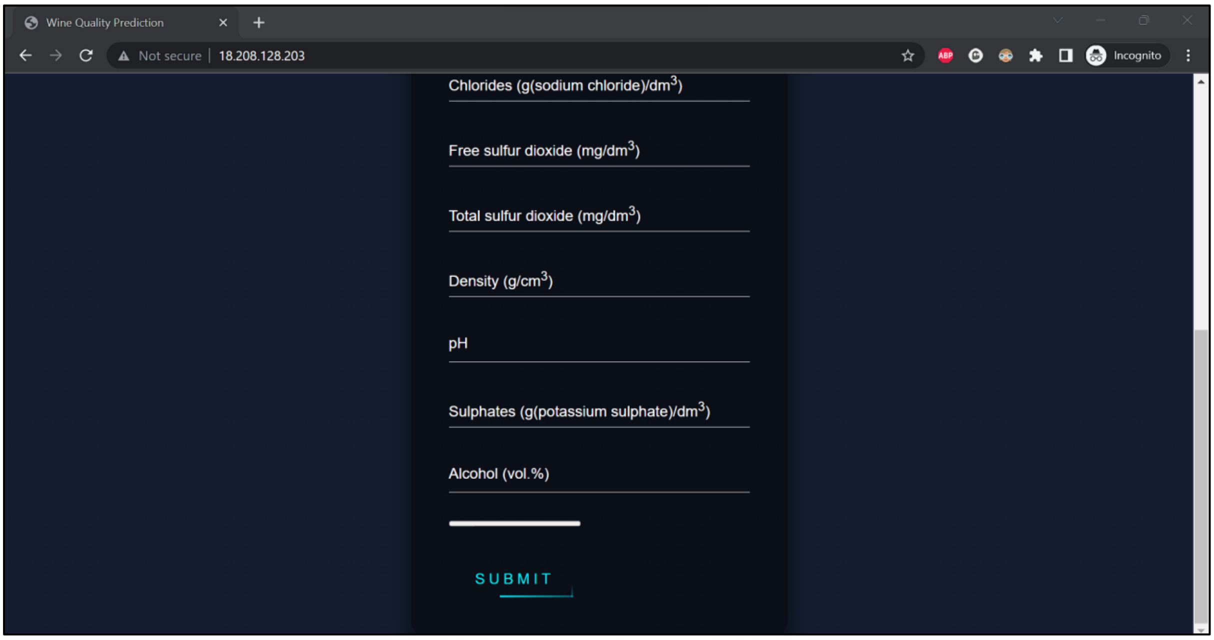
Volatile Acidity (g(acetic acid)/dm^3)
0.27

Citric Acid (g/dm^3)
0.36

Residual sugar (g/dm^3)
20.7

Chlorides (g(sodium chloride)/dm^3)

Free sulfur dioxide (mg/dm^3)



Chlorides (g(sodium chloride)/dm^3)

Free sulfur dioxide (mg/dm^3)

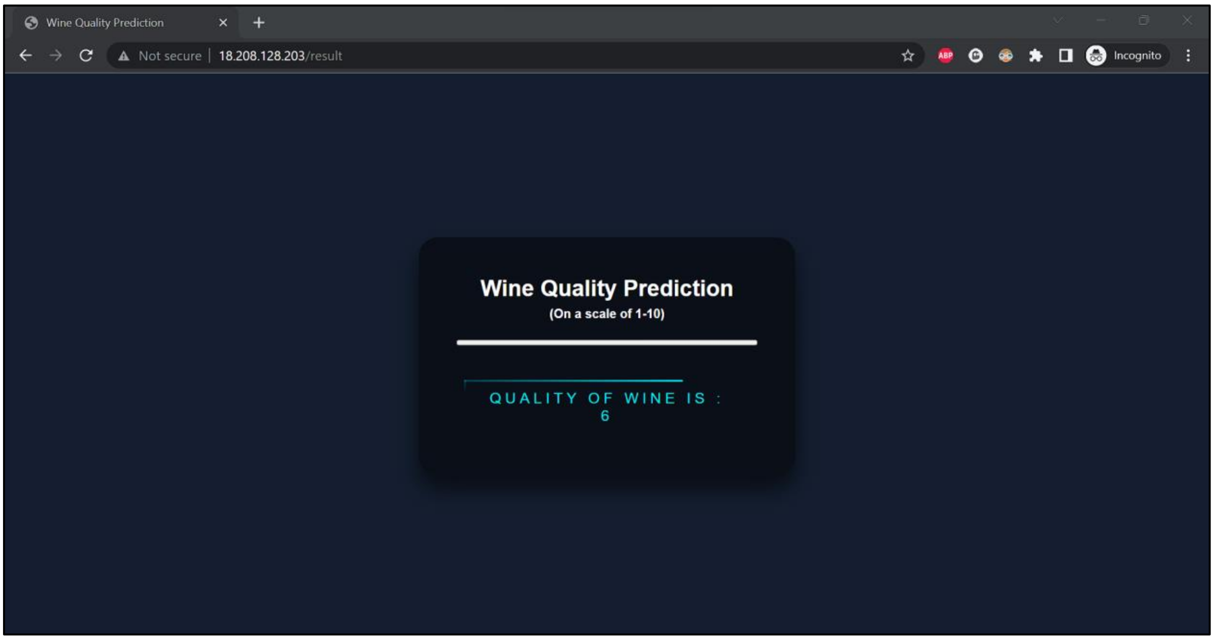
Total sulfur dioxide (mg/dm^3)

Density (g/cm^3)

pH

Sulphates (g(potassium sulphate)/dm^3)

Alcohol (vol.%)



Wine Quality Prediction

(On a scale of 1-10)

QUALITY OF WINE IS : 6

Conclusion:

Thus a machine learning model to predict the quality of wine has been developed and implemented successfully using the extra trees classifier. A web application has also been developed using Flask and has been deployed as well.