WINE QUALITY PREDICTION

A MINI PROJECT REPORT

Submitted by

SOURABH SHARMA [RA2111042010007]

for the course 18CSE363J – Machine Learning

Under the guidance of

Dr. P. Kanmani
(Assistant Professor, Department of Data Science and Business Systems)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE ENGINEERING AND BUSINESS SYSTEMS
of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

OCTOBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur – 603203.

COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

BONAFIDE CERTIFICATE

Reg. No:

R	A	2	1	1	1	0	4	2	0	1	0	0	0	7

Certified that this is the bonafide record of work done by **SOURABH SHARMA** of V semester B.Tech. COMPUTER SCIENCE WITH BUSINESS SYSTEMS during the academic year 2023-2024 in the 18CSE363J – Machine Learning Laboratory.

------------------------------ ------------------------------

Dr. P. Kanmani
Faculty - Incharge

Dr. M Lakshmi
HOD/DSBS

Submitted for the practical examination held on____________ at SRM Institute of Science and Technology, Kattankulathur, Chennai-603203.

---------------------------- --------------------------

Examinar-1

Examinar-2

TABLE OF CONTENTS

ABSTRACT

This research is a significant step towards predicting the quality of red wine by examining its different characteristics. To do this, we gathered data from various sources, creating a dataset that contains information about different types of red wine. We then used three different computer techniques: Random Forest, Support Vector Machine, and Naïve Bayes, to help us analyze this data and make predictions about the quality of the wine. To see how well these techniques perform, we divided the data into two parts: a training set and a testing set. The training set is like a practice set for the computer. It helps the computer learn how to make predictions. We applied the three techniques to the training set and calculated various measures to understand how accurate and effective they are at predicting wine quality. After analyzing the training set, we compared the results from the three techniques. We looked at which technique did the best job of predicting wine quality based on the training data. This allowed us to choose the most promising technique to use for our predictions. But we didn't stop there. We also explored the possibility of making our predictions even more accurate. To do this, we considered taking the best aspects from each of the three techniques and combining them into one super technique. This could help us improve the accuracy and efficiency of our predictions. In summary, this research is a significant step in the direction of predicting red wine quality by using various attributes. We collected data, applied different techniques, compared their performance on a training set, and selected the best one based on the training results. Additionally, we explored the potential for enhancing our predictions by merging the strengths of these techniques. This work aims to ensure that we can enjoy high-quality red wine while maintaining our health and safety.

PROBLEM STATEMENT

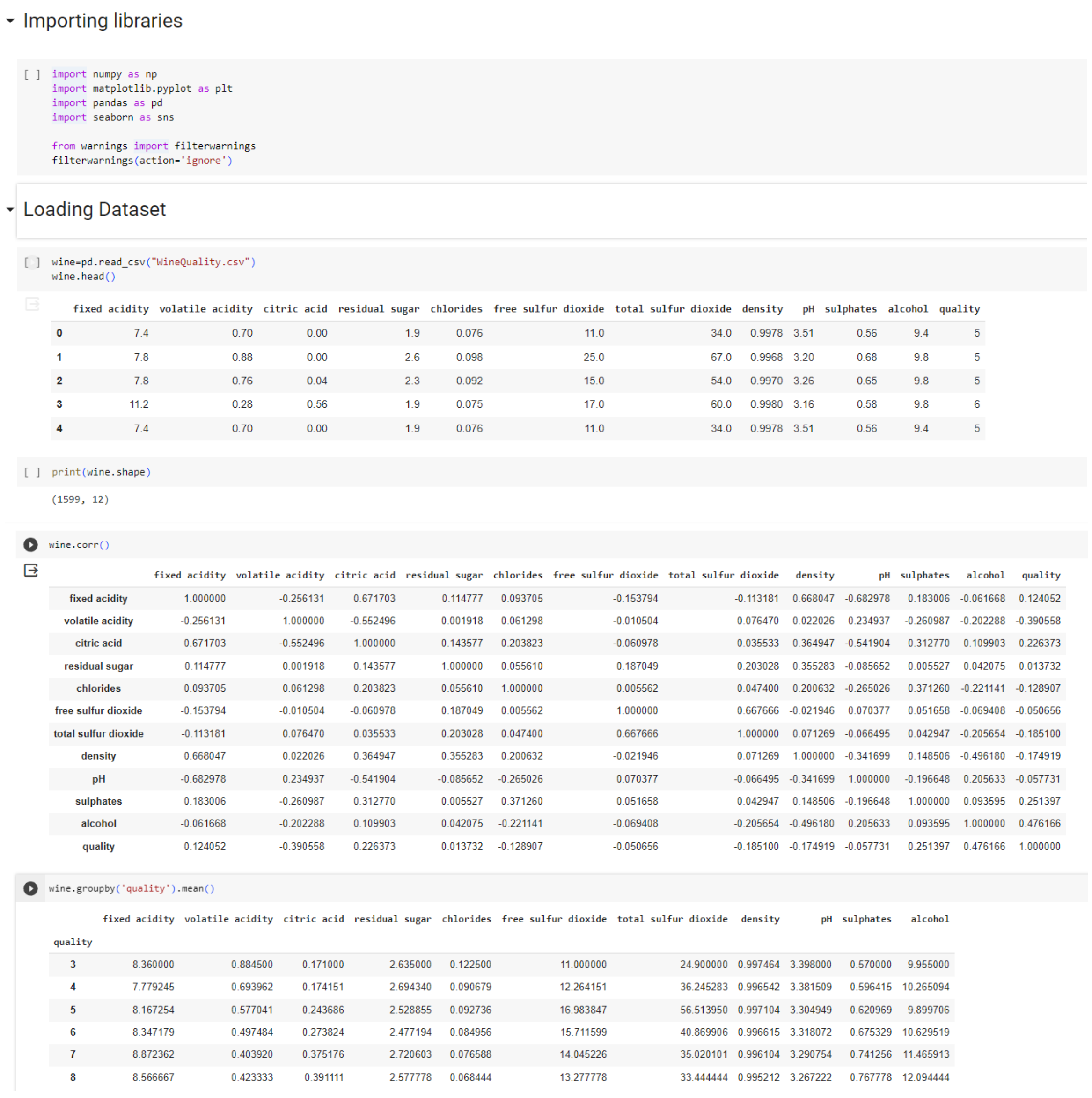
This research project focuses on predicting the quality of red wine by using data analysis and machine learning techniques. We gathered information about various types of red wine from different sources and split it into two parts: a training set and a testing set. The training set helped the computer learn how to make predictions about wine quality. We applied three different computer techniques - Random Forest, Support Vector Machine, and Naïve Bayes - to the training set. These techniques have their unique ways of assessing wine quality. We then measured how accurate and effective they were in predicting wine quality. After evaluating the results, we selected the technique that performed the best on the training data. But we didn't stop there. We also explored the possibility of improving our predictions. One way we considered was combining the best aspects of the three techniques into a single, more powerful approach. In a nutshell, this research is all about helping people enjoy high-quality red wine while ensuring their health and safety. By examining various attributes of red wine and using different techniques to make predictions, we aim to provide a clear understanding of which technique works best. Moreover, we explore ways to make our predictions even more accurate and efficient. This research contributes to the enjoyment of red wine while maintaining quality and safety standards.

OBJECTIVES

- Data Collection: Gather data containing various attributes of red wine, such as acidity, alcohol content, and more.
- Model Development: Create a predictive model to assess wine quality based on these attributes.
- Training and Testing: Split the dataset into training and testing sets to train the model and evaluate its accuracy.
- Prediction Accuracy: Achieve accurate wine quality predictions using metrics like mean squared error or classification accuracy.
- Model Interpretation: Understand which attributes most significantly impact wine quality.
- Enhancement: Explore ways to improve prediction accuracy, potentially through advanced techniques.
- Real-world Application: Translate the research into practical use, benefiting wine production, consumer choices, and wine enthusiasts.
- Validation: Ensure the model's effectiveness across various datasets and wine-related scenarios.

.

PROGRAM CODE



Importing libraries

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    import seaborn as sns

    from warnings import filterwarnings
    filterwarnings(action='ignore')
```

Loading Dataset

```
[ ] wine=pd.read_csv("WineQuality.csv")
    wine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

```
[ ] print(wine.shape)

    (1599, 12)
```

```
wine.corr()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.256131	0.671703	0.114777	0.093705	-0.153794	-0.113181	0.668047	-0.682978	0.183006	-0.061668	0.124052
volatile acidity	-0.256131	1.000000	-0.552496	0.001918	0.061298	-0.010504	0.076470	0.022026	0.234937	-0.260987	-0.202288	-0.390558
citric acid	0.671703	-0.552496	1.000000	0.143577	0.203823	-0.060978	0.035533	0.364947	-0.541904	0.312770	0.109903	0.226373
residual sugar	0.114777	0.001918	0.143577	1.000000	0.055610	0.187049	0.203028	0.355283	-0.085652	0.005527	0.042075	0.013732
chlorides	0.093705	0.061298	0.203823	0.055610	1.000000	0.005562	0.047400	0.200632	-0.265026	0.371260	-0.221141	-0.128907
free sulfur dioxide	-0.153794	-0.010504	-0.060978	0.187049	0.005562	1.000000	0.667666	-0.021946	0.070377	0.051658	-0.069408	-0.050656
total sulfur dioxide	-0.113181	0.076470	0.035533	0.203028	0.047400	0.667666	1.000000	0.071269	-0.066495	0.042947	-0.205654	-0.185100
density	0.668047	0.022026	0.364947	0.355283	0.200632	-0.021946	0.071269	1.000000	-0.341699	0.148506	-0.496180	-0.174919
pH	-0.682978	0.234937	-0.541904	-0.085652	-0.265026	0.070377	-0.066495	-0.341699	1.000000	-0.196648	0.205633	-0.057731
sulphates	0.183006	-0.260987	0.312770	0.005527	0.371260	0.051658	0.042947	0.148506	-0.196648	1.000000	0.093595	0.251397
alcohol	-0.061668	-0.202288	0.109903	0.042075	-0.221141	-0.069408	-0.205654	-0.496180	0.205633	0.093595	1.000000	0.476166
quality	0.124052	-0.390558	0.226373	0.013732	-0.128907	-0.050656	-0.185100	-0.174919	-0.057731	0.251397	0.476166	1.000000

```
wine.groupby('quality').mean()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
quality											
3	8.360000	0.884500	0.171000	2.635000	0.122500	11.000000	24.900000	0.997464	3.398000	0.570000	9.955000
4	7.779245	0.693962	0.174151	2.694340	0.090679	12.264151	36.245283	0.996542	3.381509	0.596415	10.265094
5	8.167254	0.577041	0.243686	2.528855	0.092736	16.983847	56.513950	0.997104	3.304949	0.620969	9.899706
6	8.347179	0.497484	0.273824	2.477194	0.084956	15.711599	40.869906	0.996615	3.318072	0.675329	10.629519
7	8.872362	0.403920	0.375176	2.720603	0.076588	14.045226	35.020101	0.996104	3.290754	0.741256	11.465913
8	8.566667	0.423333	0.391111	2.577778	0.068444	13.277778	33.444444	0.995212	3.267222	0.767778	12.094444

```
[ ] wine.describe(include='all')
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	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	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

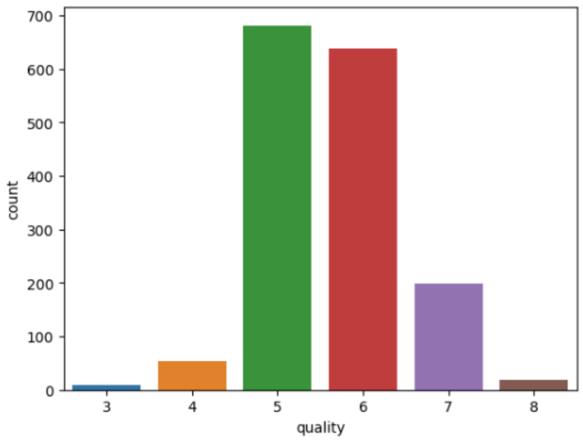
```
[ ] print(wine.isna().sum())
```

```
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
```

Data Analysis

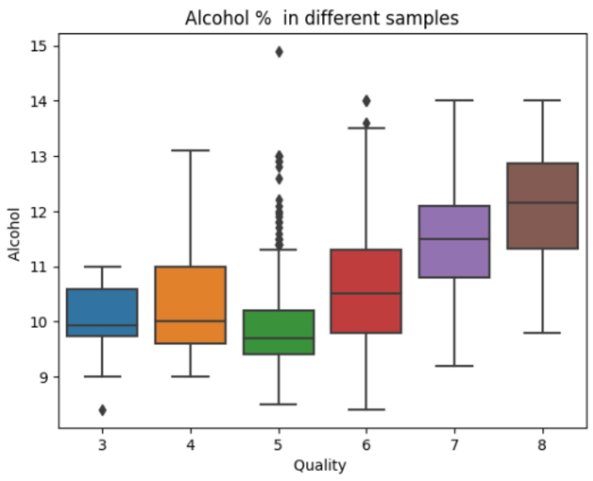
```
sns.countplot(x='quality', data=wine)
```

```
<Axes: xlabel='quality', ylabel='count'>
```



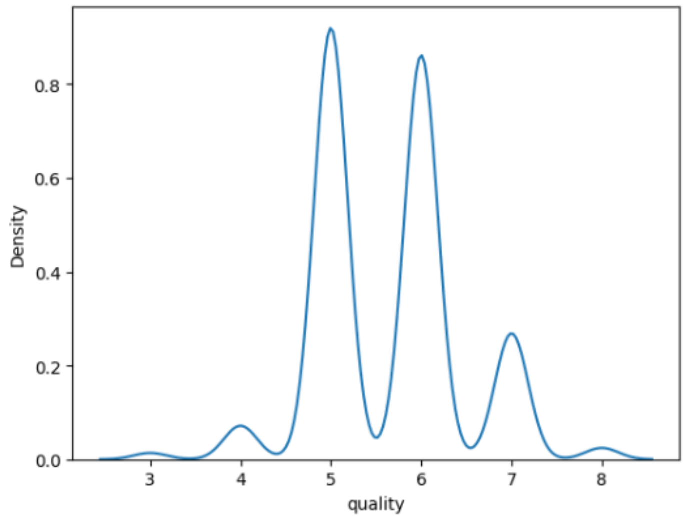
```
[ ] bx = sns.boxplot(x='quality', y='alcohol', data = wine)
    bx.set(xlabel='Quality ', ylabel='Alcohol ', title='Alcohol %  in different samples')
```

```
[Text(0.5, 0, 'Quality '),
 Text(0, 0.5, 'Alcohol '),
 Text(0.5, 1.0, 'Alcohol %  in different samples')]
```



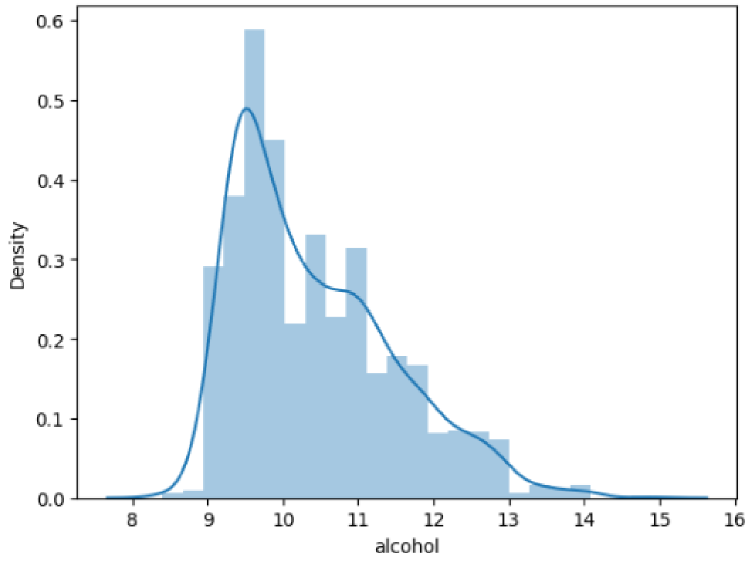
```
[ ] sns.kdeplot(wine.query('quality > 2').quality)
```

```
<Axes: xlabel='quality', ylabel='Density'>
```

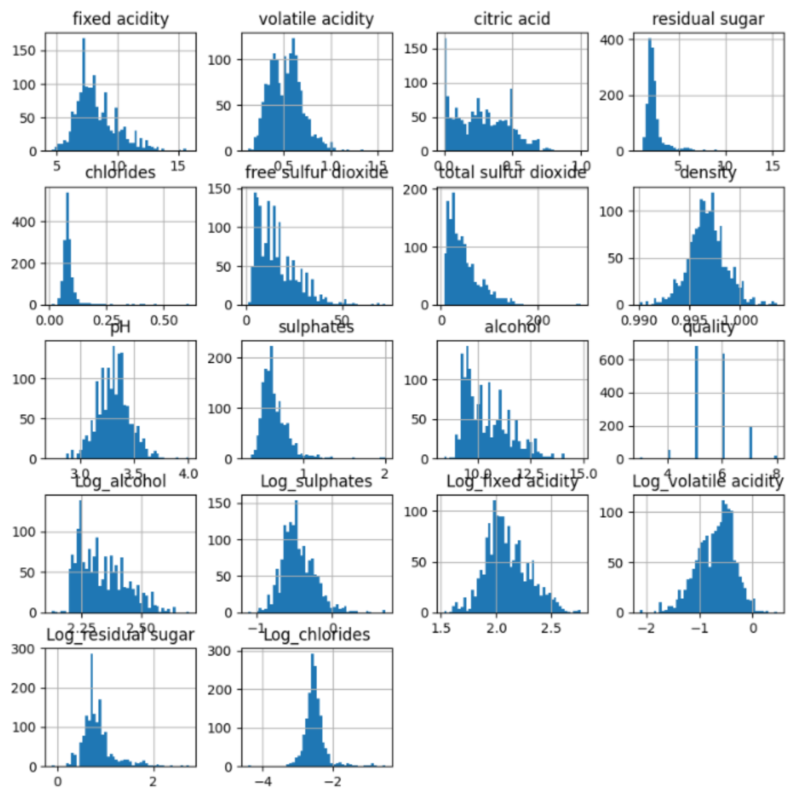


```
sns.distplot(wine['alcohol'])
```

```
<Axes: xlabel='alcohol', ylabel='Density'>
```



```
wine.hist(figsize=(10,10),bins=50)
plt.show()
```



Feature Selection

```
wine['goodquality']=[1 if x>7 else 0 for x in wine['quality']]
X=wine.drop(['quality','goodquality'],axis=1)
y= wine['goodquality']
```

```
wine['goodquality'].value_counts()
```

```
0    1581
1      18
Name: goodquality, dtype: int64
```

```
wine['quality'].value_counts()
```

```
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
```

```
X
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	Log_alcohol	Log_sulphates	Log_fixed acidity	Log_volatile acidity	Log_residual sugar	Log_chlorides
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	2.240710	-0.579818	2.001480	-0.356675	0.641854	-2.577022
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	2.282382	-0.385662	2.054124	-0.127833	0.955511	-2.322788
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	2.282382	-0.430783	2.054124	-0.274437	0.832909	-2.385967
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	2.282382	-0.544727	2.415914	-1.272966	0.641854	-2.590267
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	2.240710	-0.579818	2.001480	-0.356675	0.641854	-2.577022
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	2.351375	-0.544727	1.824549	-0.510826	0.693147	-2.407946
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	2.415914	-0.274437	1.774952	-0.597837	0.788457	-2.780621
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	2.397895	-0.287682	1.840550	-0.673345	0.832909	-2.577022
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	2.322388	-0.342490	1.774952	-0.438505	0.693147	-2.590267
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	2.397895	-0.415515	1.791759	-1.171183	1.280934	-2.703063

1599 rows × 17 columns

Feature Importance

```
[ ] from sklearn.linear_model import LogisticRegression
    model = LogisticRegression()

    from sklearn.ensemble import ExtraTreesClassifier
    classifiern = ExtraTreesClassifier()
    classifiern.fit(X,y)
    score = classifiern.feature_importances_
    print(score)
```

```
[0.05106964 0.05477897 0.06525135 0.04900596 0.04587302 0.05248986
 0.05145439 0.0569887  0.07829926 0.05947128 0.07568819 0.07091763
 0.07060123 0.05481841 0.05440084 0.05401532 0.05487596]
```

Splitting Data set

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size=0.2,random_state=7)
```

Logistic regression

```
[ ] from sklearn.linear_model import LogisticRegression
    model = LogisticRegression()
    model.fit(X_train,Y_train)
    Y_pred = model.predict(X_test)
```

```
[ ] from sklearn.metrics import accuracy_score,confusion_matrix
    print("Accuracy Score:",accuracy_score(Y_test,Y_pred))
    print("Confusion_mat :",confusion_matrix(Y_test,Y_pred))
```

```
Accuracy Score: 0.9875
Confusion_mat : [[316   0]
 [  4   0]]
```

```
[ ] from sklearn.linear_model import LinearRegression
    model = LinearRegression()
    model.fit(X_train,Y_train)
    Y_pred = model.predict(X_test)
```

KNN

```
[ ] from sklearn.neighbors import KNeighborsClassifier
    model1 = KNeighborsClassifier(n_neighbors=3)
    model1.fit(X_train,Y_train)
    y_pred = model1.predict(X_test)

    from sklearn.metrics import accuracy_score
    print("Accuracy Score:",accuracy_score(Y_test,y_pred))
    print("Confusion_mat :",confusion_matrix(Y_test,y_pred))
```

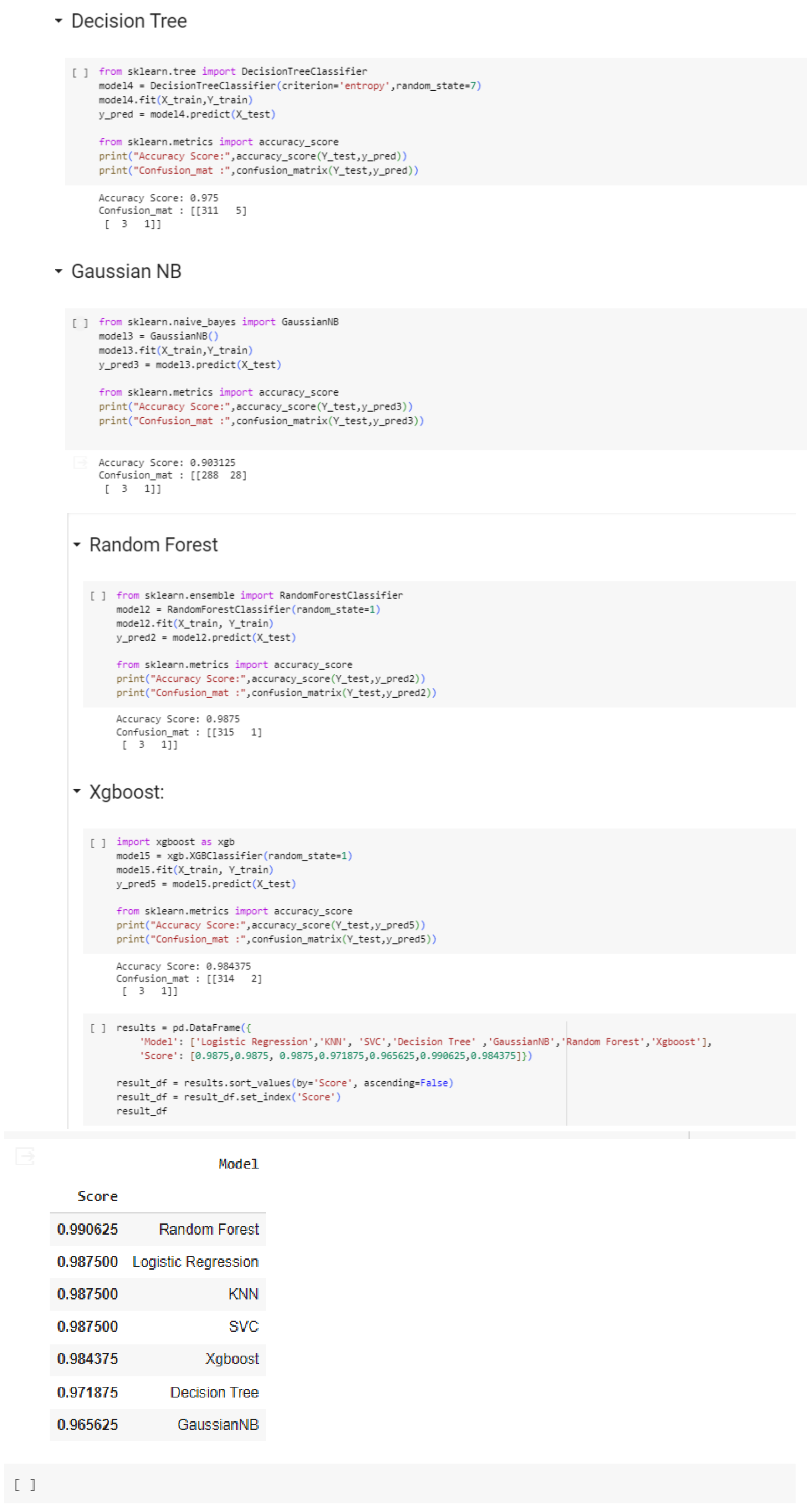
```
Accuracy Score: 0.9875
Confusion_mat : [[316   0]
 [  4   0]]
```

SVC

```
[ ] from sklearn.svm import SVC
    model3 = SVC()
    model3.fit(X_train,Y_train)
    pred_y = model3.predict(X_test)

    from sklearn.metrics import accuracy_score
    print("Accuracy Score:",accuracy_score(Y_test,pred_y))
    print("Confusion_mat :",confusion_matrix(Y_test,pred_y))
```

```
Accuracy Score: 0.9875
Confusion_mat : [[316   0]
 [  4   0]]
```



Decision Tree

```
[ ] from sklearn.tree import DecisionTreeClassifier
    model4 = DecisionTreeClassifier(criterion='entropy',random_state=7)
    model4.fit(X_train,Y_train)
    y_pred = model4.predict(X_test)

    from sklearn.metrics import accuracy_score
    print("Accuracy Score:",accuracy_score(Y_test,y_pred))
    print("Confusion_mat :",confusion_matrix(Y_test,y_pred))
```

```
Accuracy Score: 0.975
Confusion_mat : [[311   5]
 [  3   1]]
```

Gaussian NB

```
[ ] from sklearn.naive_bayes import GaussianNB
    model3 = GaussianNB()
    model3.fit(X_train,Y_train)
    y_pred3 = model3.predict(X_test)

    from sklearn.metrics import accuracy_score
    print("Accuracy Score:",accuracy_score(Y_test,y_pred3))
    print("Confusion_mat :",confusion_matrix(Y_test,y_pred3))
```

```
Accuracy Score: 0.903125
Confusion_mat : [[288  28]
 [  3   1]]
```

Random Forest

```
[ ] from sklearn.ensemble import RandomForestClassifier
    model2 = RandomForestClassifier(random_state=1)
    model2.fit(X_train, Y_train)
    y_pred2 = model2.predict(X_test)

    from sklearn.metrics import accuracy_score
    print("Accuracy Score:",accuracy_score(Y_test,y_pred2))
    print("Confusion_mat :",confusion_matrix(Y_test,y_pred2))
```

```
Accuracy Score: 0.9875
Confusion_mat : [[315   1]
 [  3   1]]
```

Xgboost:

```
[ ] import xgboost as xgb
    model5 = xgb.XGBClassifier(random_state=1)
    model5.fit(X_train, Y_train)
    y_pred5 = model5.predict(X_test)

    from sklearn.metrics import accuracy_score
    print("Accuracy Score:",accuracy_score(Y_test,y_pred5))
    print("Confusion_mat :",confusion_matrix(Y_test,y_pred5))
```

```
Accuracy Score: 0.984375
Confusion_mat : [[314   2]
 [  3   1]]
```

```
[ ] results = pd.DataFrame({
        'Model': ['Logistic Regression','KNN', 'SVC','Decision Tree' ,'GaussianNB','Random Forest','Xgboost'],
        'Score': [0.9875,0.9875, 0.9875,0.971875,0.965625,0.990625,0.984375]})

    result_df = results.sort_values(by='Score', ascending=False)
    result_df = result_df.set_index('Score')
    result_df
```

	Model
Score	
0.990625	Random Forest
0.987500	Logistic Regression
0.987500	KNN
0.987500	SVC
0.984375	Xgboost
0.971875	Decision Tree
0.965625	GaussianNB

```
[ ]
```

SCOPE OF THE PROJECT

- Wine Quality Assessment: The project's primary scope involves developing a model to predict the quality of red wine based on its attributes. This application can have significant implications for wine producers, distributors, and consumers.
- Data Analysis: The project involves the analysis of data related to wine attributes, including chemical composition, sensory characteristics, and more, providing valuable insights into wine quality.
- Machine Learning Techniques: The project explores the application of machine learning algorithms, such as neural networks or other predictive models, to make accurate quality predictions.
- Model Interpretation: Understanding which attributes contribute most to wine quality is a crucial part of the project's scope, as it can offer valuable information to the wine industry.
- Real-world Application: The research can be applied in practical scenarios, aiding wine producers in quality control and assisting consumers in selecting wines that align with their preferences.
- Continuous Improvement: The scope may extend to enhancing the prediction model by integrating advanced techniques or expanding its application to other wine-related aspects, such as wine aging or varietal classification.
- Validation and Generalizability: Ensuring the model's effectiveness across different datasets and wine-related contexts is part of the project's scope, ensuring its generalizability.
- Research Contribution: The research project contributes to the field of wine quality assessment and machine learning applications, potentially benefiting various stakeholders in the wine industry.

CONCLUSION

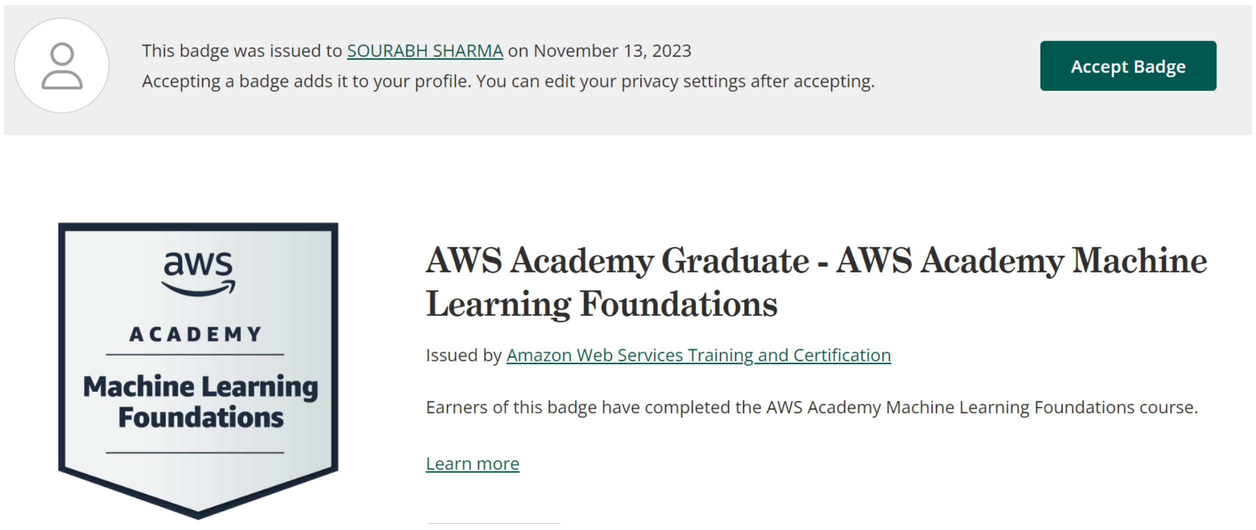
Finally, by utilising data analysis and machine learning approaches, this study effort set out to forecast the quality of red wine. We were able to further our understanding of wine evaluation by building prediction models that could evaluate wine quality through the collection and analysis of data covering a wide range of wine variables.

The development of neural networks and other machine learning techniques opened the door to precise quality forecasts. We found the best approaches for this assignment by thoroughly testing and assessing them on training and testing datasets. Additionally, knowing which characteristics had the greatest impact on wine quality provided insightful information for both wine producers and consumers.

This research provided potential for ongoing development in addition to opening doors to practical applications including helping wine industry quality control and assisting customers in making educated decisions. We investigated the possibilities for improving the model, either by adding cutting-edge methods or expanding the range of uses for wine.

Essentially, this effort improves the evaluation and pleasure of red wine while also making a significant contribution to the machine learning applications and the wine business. It is a step towards a time when wine quality can be accurately anticipated, allowing people to enjoy fine red wine while preserving their health and wellbeing.

CERTIFICATE


This badge was issued to SOURABH SHARMA on November 13, 2023
Accepting a badge adds it to your profile. You can edit your privacy settings after accepting.

AWS Academy Graduate - AWS Academy Machine Learning Foundations

Issued by Amazon Web Services Training and Certification

Earners of this badge have completed the AWS Academy Machine Learning Foundations course.

Learn more