A Mini-Project Report On

“Wine Quality Testing”

Submitted By

Nikhil Pingale

URN No: 06071999

M.CA. (Data Science)

School of Engineering

Ajeenkya D Y Patil University

Pune – 411015

Academic Year 2022-2024

APRIL – 2023

CERTIFICATE

This is to certify that
Nikhil Pingale Urn No: 06071999,

Of **M.CA(Data Science)** has successfully completed his Project in Machine Learning

"Wine Quality Testing"

to our satisfaction and submitted the same during the academic year 2022- 2024

towards the partial fulfillment of degree of Master **of Computer Application in Data Science** of AJEENKYA D Y PATIL UNIVERSITY.

Table of Contents

1. INTRODUCTION

1.1 Domain of the problem statement

Testing of wine is one of the major tasks today , as the taste of the wine depends on many various factors which are still unknow to people . Each and every factor affects the taste and quality of the wine. Our project focuses on predicting quality of wine based on these various features.

In this mini project we will be predicting the quality of wine .

1.2 Motivation

Winemakers need to ensure that their wines meet the desired quality standards, comply with regulatory requirements, and are safe for consumption. However, there are various factors that can affect the quality and safety of the wine, such as grape variety, fermentation conditions, storage conditions, and processing techniques. Therefore, winemakers need to conduct thorough testing and analysis of their wines to identify any potential issues and optimize their production processes. The challenge is to select the appropriate tests and analytical methods that can provide accurate and reliable results, as well as to interpret and apply the data effectively to improve the quality, safety, and marketability of the wine.

1.3 Problem statement

To explore wine testing prediction and classification models for the Winemakers so that their wine quality is up to the mark and so that it helps the industry to optimize meet the supply and demand.

2. SOLUTION DESIGN

2.1 Solution Approach

We have used Wine Quality data from Kaggle. This dataset contains more than 1500 records. First, we have performed EDA on all dataset to understand the data if it contains any null value, missing value, any outliers, etc. After performing EDA we got the basic understanding of our dataset. Based on our observations we have considered some columns and those which were highly correlated were removed by performing label encoding. After performing all these task we have implemented certain machine learning algorithm to know the accuracy and precision of our data.

2.2 Technology Stack

We have used Google collab notebook platform
Language:- python
Libraries:-

- numpy
- Pandas
- matplotlib.pyplot
- seaborn
- Sklearn-EDA, pre-processing as well as for model building.

2.3 Designing model

On our Wine Quality dataset, we have applied Logistic regression, Random forest ,Gaussian Naive Bayes, SVC and Ensemble model. Out of these algorithms SVC, Ensemble classifier & Logistic Regression gives highest accuracy.

Our goal is to classify the types of wine and their quality depending on the other factors such as alcohol quantity, fixed acidity, volatile acidity, determination of density, pH etc.

3. SOLUTION IMPLEMENTATION AND RESULT

3.1 Obtaining Data

The information gathered is from Kaggle. It produces several types of wines. It’s a known fact that the older the wine, the better the taste. However, there are several factors other than age that go into wine quality certification which include physiochemical tests like alcohol quantity, fixed acidity, volatile acidity, determination of density, pH, and more.

We have collected data from online source i.e

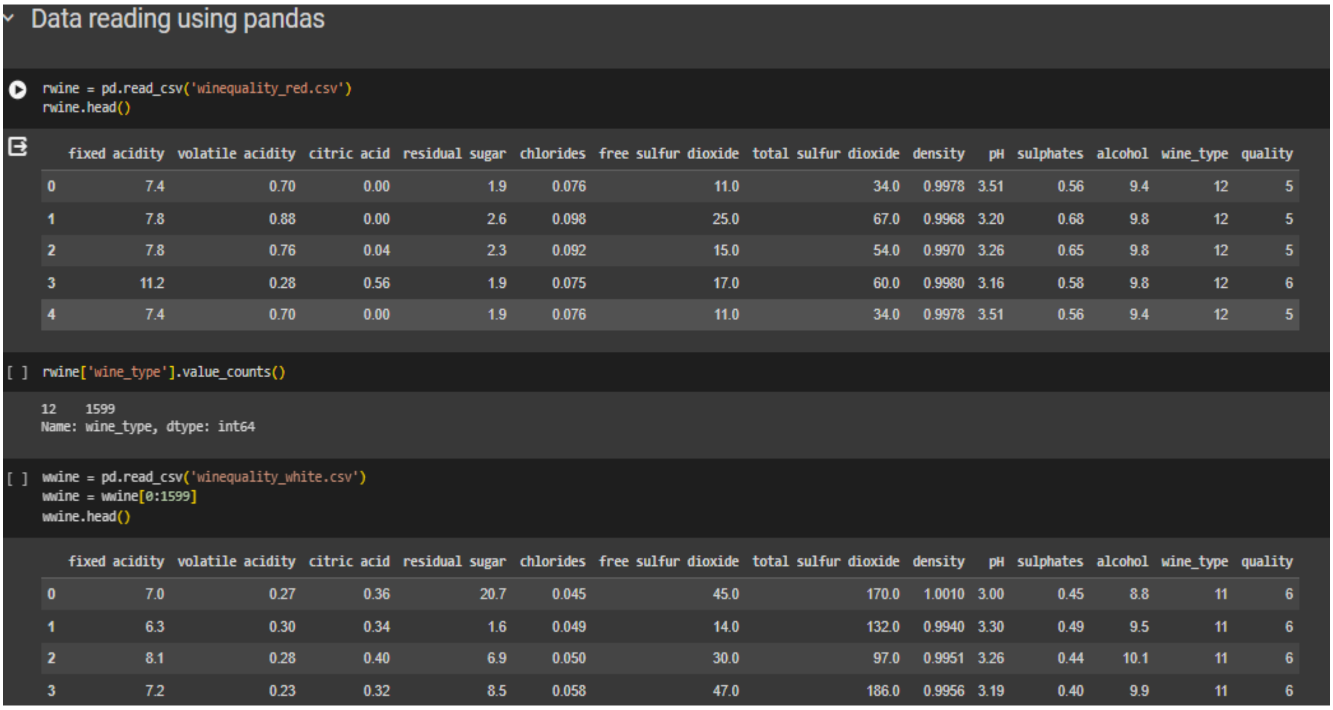
https://www.kaggle.com/

Total Observation: 1599-Rows,13-Columns

The above data is collected from a smart small-scale wine producing industry.

4.2 EDA

Using python and Google collab notebook we have done EDA on Wine Quality Testing dataset. We described the dataset and based on that performed further processing. Also plotted heat map to see correlation between the data points so that we could proceed for feature selection. Also plotted the distribution



Data reading using pandas

```
rwine = pd.read_csv('winequality_red.csv')
rwine.head()
```

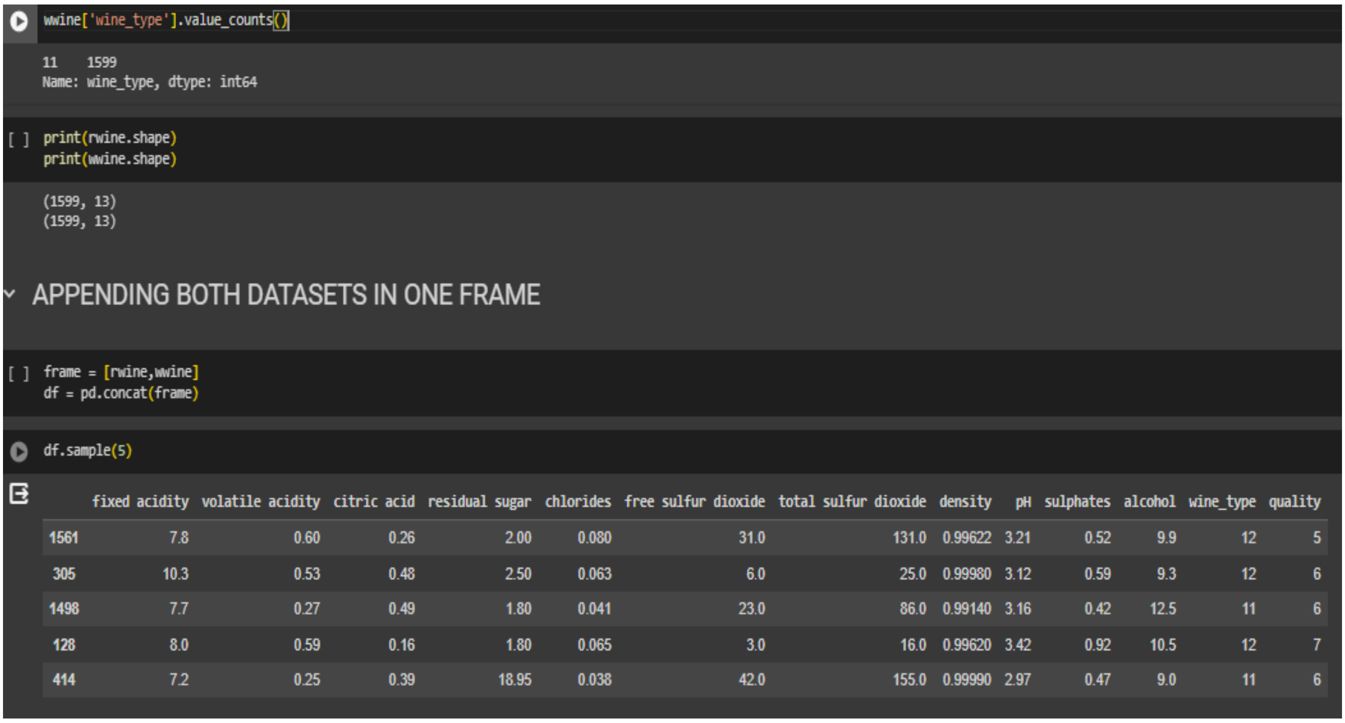
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	wine_type	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	12	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	12	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	12	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	12	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	12	5

```
rwine['wine_type'].value_counts()
```

```
12    1599
Name: wine_type, dtype: int64
```

```
wwine = pd.read_csv('winequality_white.csv')
wwine = wwine[0:1599]
wwine.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	wine_type	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	11	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	11	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	11	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	11	6



```
wwine['wine_type'].value_counts()
```

```
11    1599
Name: wine_type, dtype: int64
```

```
print(rwine.shape)
print(wwine.shape)
```

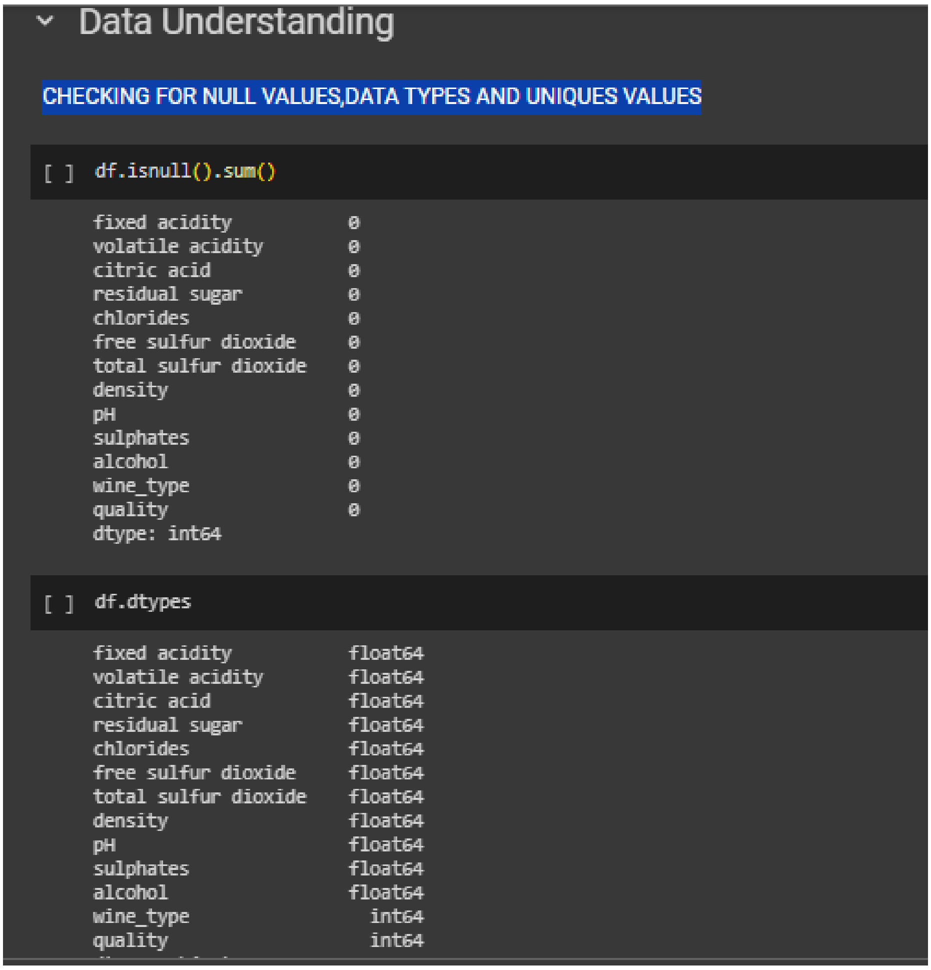
```
(1599, 13)
(1599, 13)
```

APPENDING BOTH DATASETS IN ONE FRAME

```
frame = [rwine,wwine]
df = pd.concat(frame)
```

```
df.sample(5)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	wine_type	quality
1561	7.8	0.60	0.26	2.00	0.080	31.0	131.0	0.99622	3.21	0.52	9.9	12	5
305	10.3	0.53	0.48	2.50	0.063	6.0	25.0	0.99980	3.12	0.59	9.3	12	6
1498	7.7	0.27	0.49	1.80	0.041	23.0	86.0	0.99140	3.16	0.42	12.5	11	6
128	8.0	0.59	0.16	1.80	0.065	3.0	16.0	0.99620	3.42	0.92	10.5	12	7
414	7.2	0.25	0.39	18.95	0.038	42.0	155.0	0.99990	2.97	0.47	9.0	11	6



Data Understanding

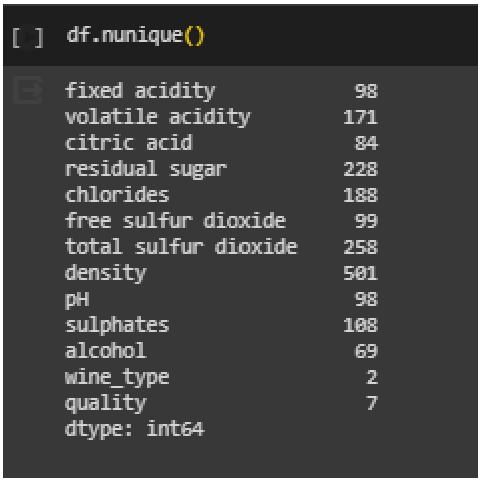
CHECKING FOR NULL VALUES,DATA TYPES AND UNIQUES VALUES

```
df.isnull().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
wine_type               0
quality                 0
dtype: int64
```

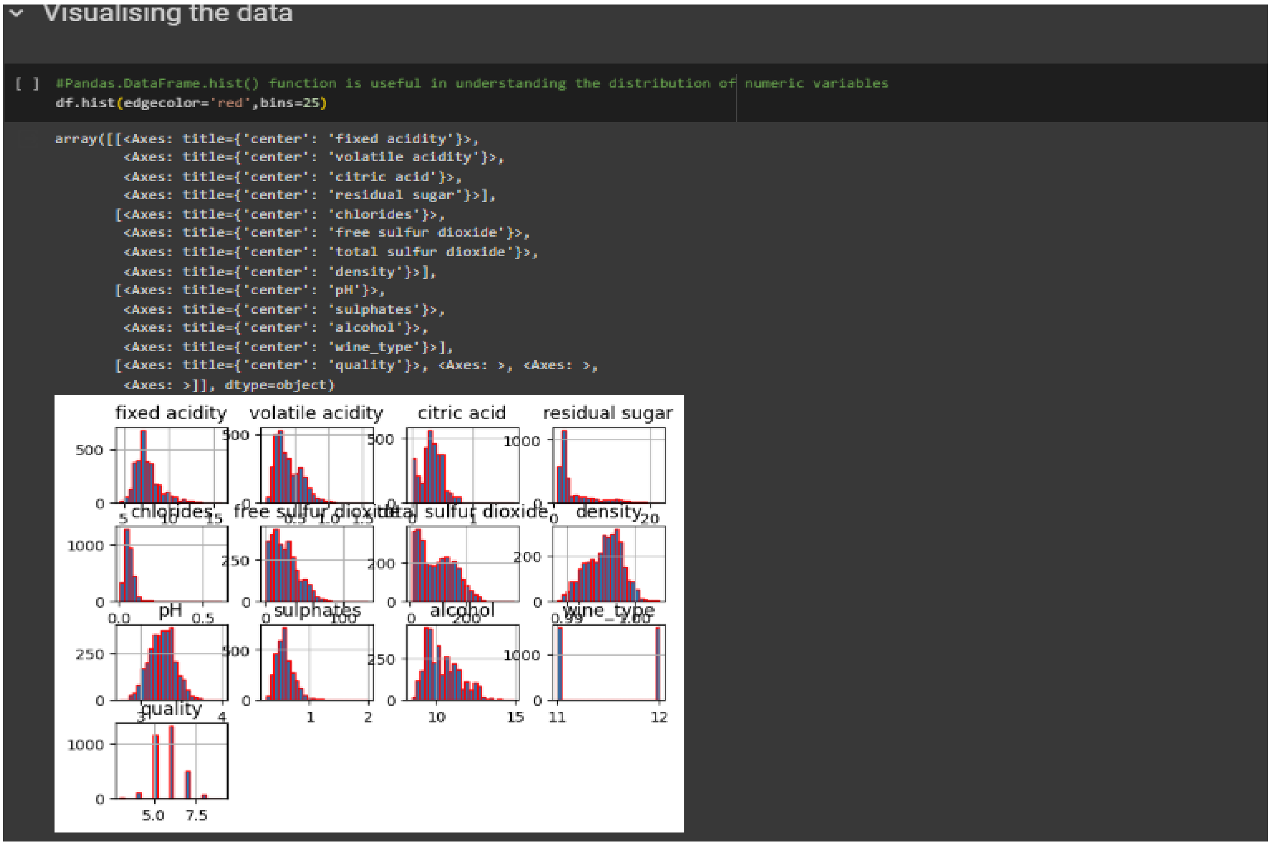
```
df.dtypes
```

```
fixed acidity           float64
volatile acidity        float64
citric acid             float64
residual sugar          float64
chlorides               float64
free sulfur dioxide     float64
total sulfur dioxide    float64
density                 float64
pH                      float64
sulphates               float64
alcohol                 float64
wine_type                 int64
quality                   int64
```



```
df.nunique()
```

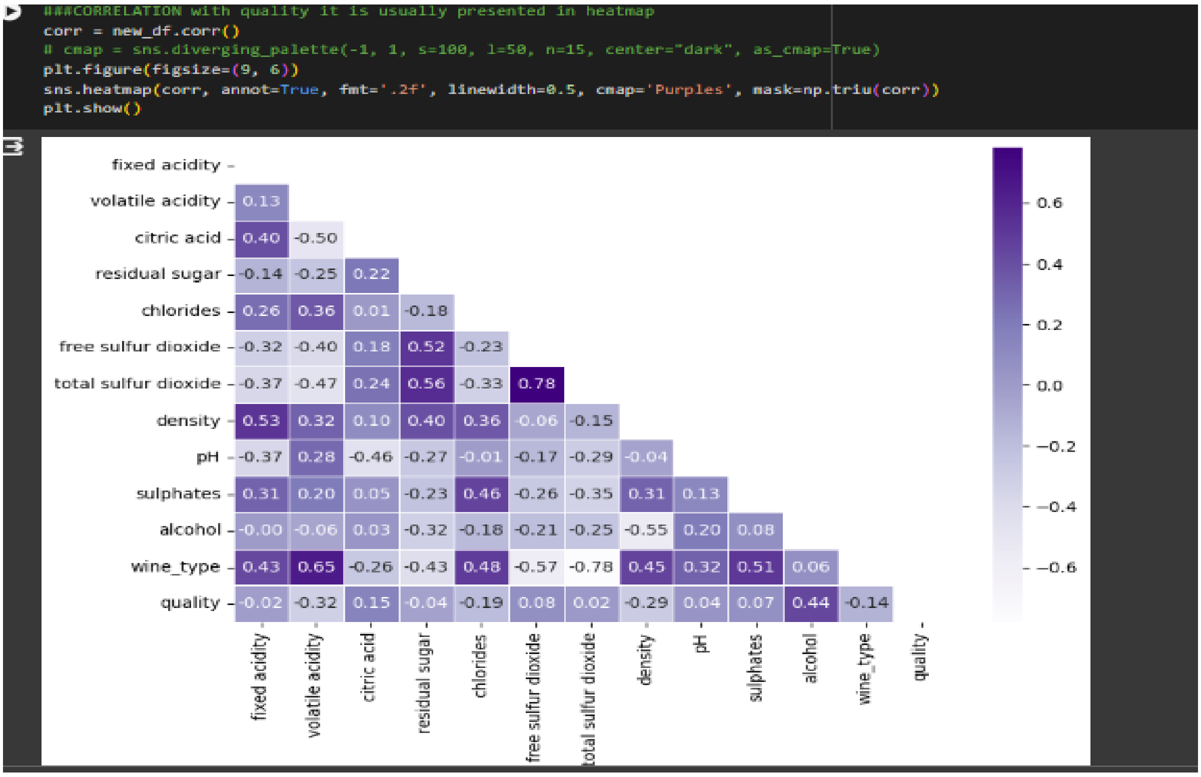
```
fixed acidity            98
volatile acidity        171
citric acid              84
residual sugar          228
chlorides               188
free sulfur dioxide      99
total sulfur dioxide    258
density                 501
pH                       98
sulphates               108
alcohol                  69
wine_type                 2
quality                   7
dtype: int64
```



Visualising the data

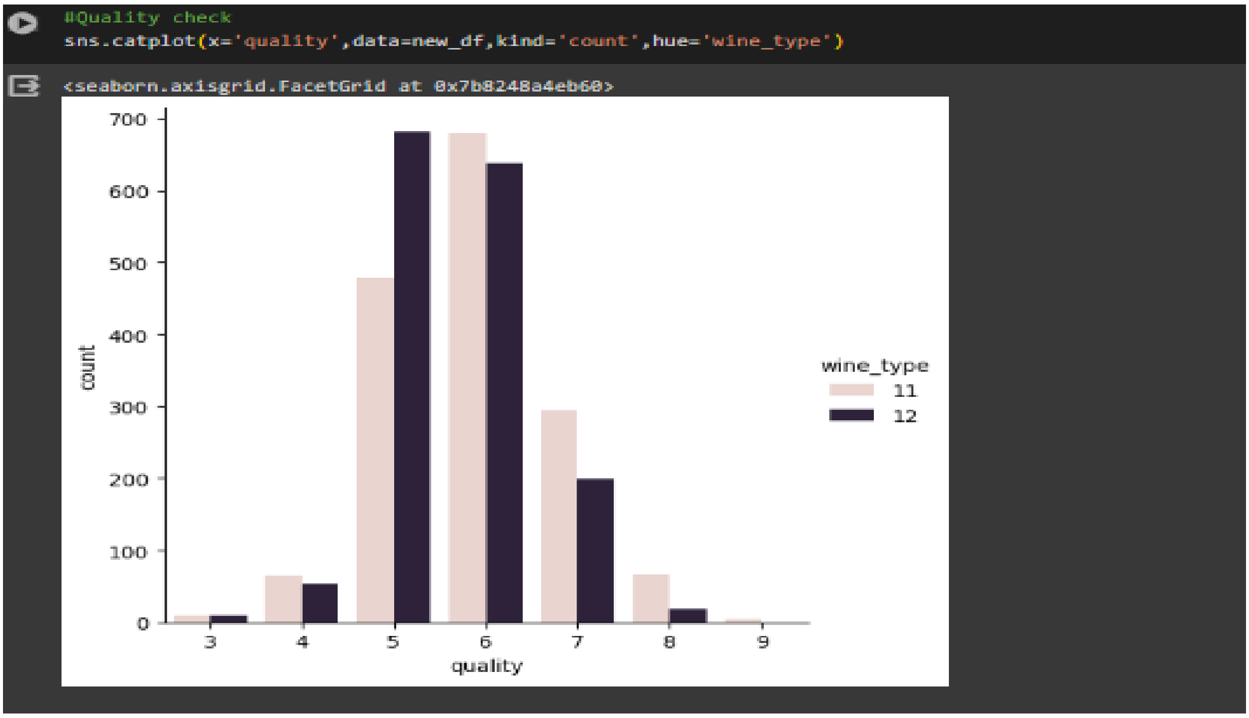
```
[ ] #Pandas.DataFrame.hist() function is useful in understanding the distribution of numeric variables
    df.hist(edgecolor='red',bins=25)
```

```
array([[<Axes: title={'center': 'fixed acidity'}>,
        <Axes: title={'center': 'volatile acidity'}>,
        <Axes: title={'center': 'citric acid'}>,
        <Axes: title={'center': 'residual sugar'}>],
       [<Axes: title={'center': 'chlorides'}>,
        <Axes: title={'center': 'free sulfur dioxide'}>,
        <Axes: title={'center': 'total sulfur dioxide'}>,
        <Axes: title={'center': 'density'}>],
       [<Axes: title={'center': 'pH'}>,
        <Axes: title={'center': 'sulphates'}>,
        <Axes: title={'center': 'alcohol'}>,
        <Axes: title={'center': 'wine_type'}>],
       [<Axes: title={'center': 'quality'}>, <Axes: >, <Axes: >,
        <Axes: >]], dtype=object)
```



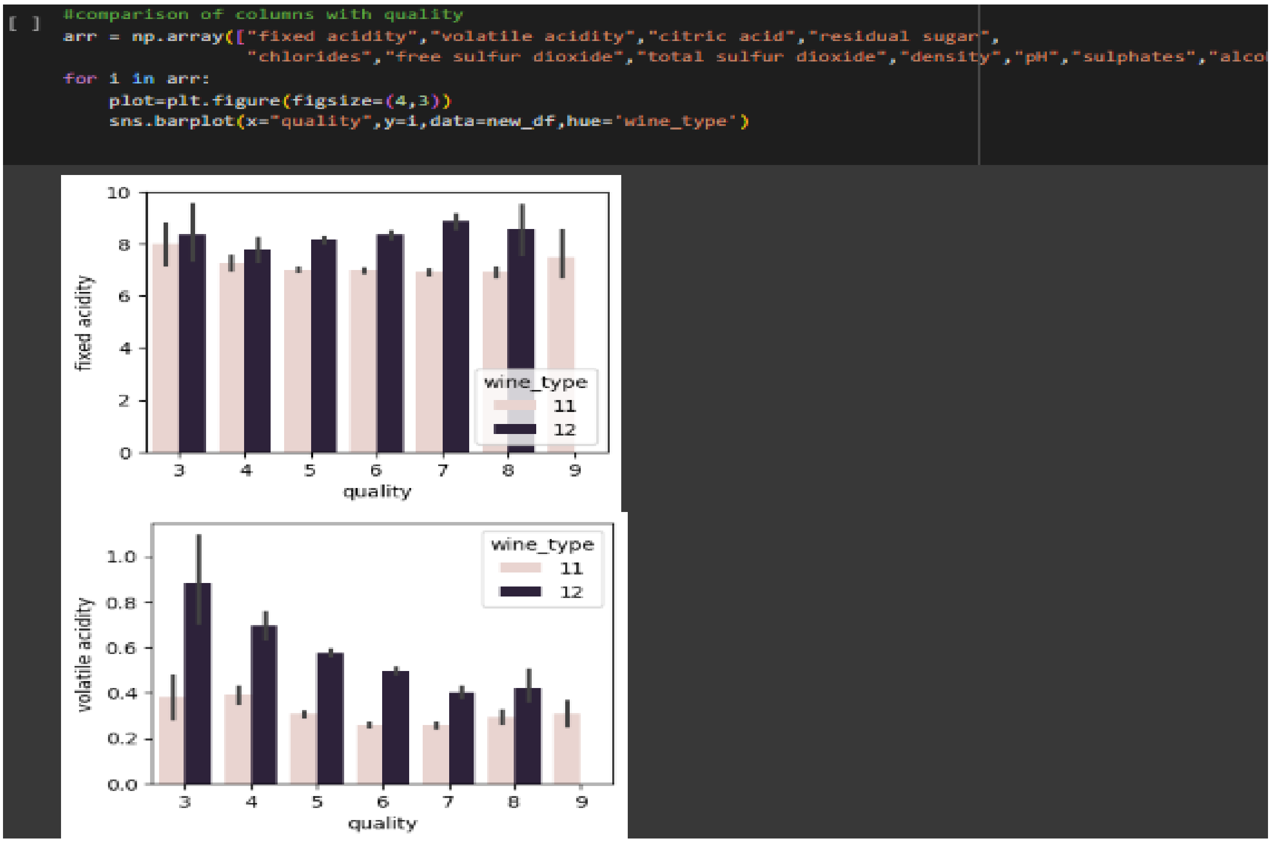
```
##CORRELATION with quality it is usually presented in heatmap
corr = new_df.corr()
# cmap = sns.diverging_palette(-1, 1, s=100, l=50, n=15, center="dark", as_cmap=True)
plt.figure(figsize=(9, 6))
sns.heatmap(corr, annot=True, fmt='.2f', linewidth=0.5, cmap='Purples', mask=np.triu(corr))
plt.show()
```

fixed acidity
volatile acidity 0.13
citric acid 0.40 -0.50
residual sugar -0.14 -0.25 0.22
chlorides 0.26 0.36 0.01 -0.18
free sulfur dioxide -0.32 -0.40 0.18 0.52 -0.23
total sulfur dioxide -0.37 -0.47 0.24 0.56 -0.33 0.78
density 0.53 0.32 0.10 0.40 0.36 -0.06 -0.15
pH -0.37 0.28 -0.46 -0.27 -0.01 -0.17 -0.29 -0.04
sulphates 0.31 0.20 0.05 -0.23 0.46 -0.26 -0.35 0.31 0.13
alcohol -0.00 -0.06 0.03 -0.32 -0.18 -0.21 -0.25 -0.55 0.20 0.08
wine_type 0.43 0.65 -0.26 -0.43 0.48 -0.57 -0.78 0.45 0.32 0.51 0.06
quality -0.02 -0.32 0.15 -0.04 -0.19 0.08 0.02 -0.29 0.04 0.07 0.44 -0.14
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
wine_type
quality
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

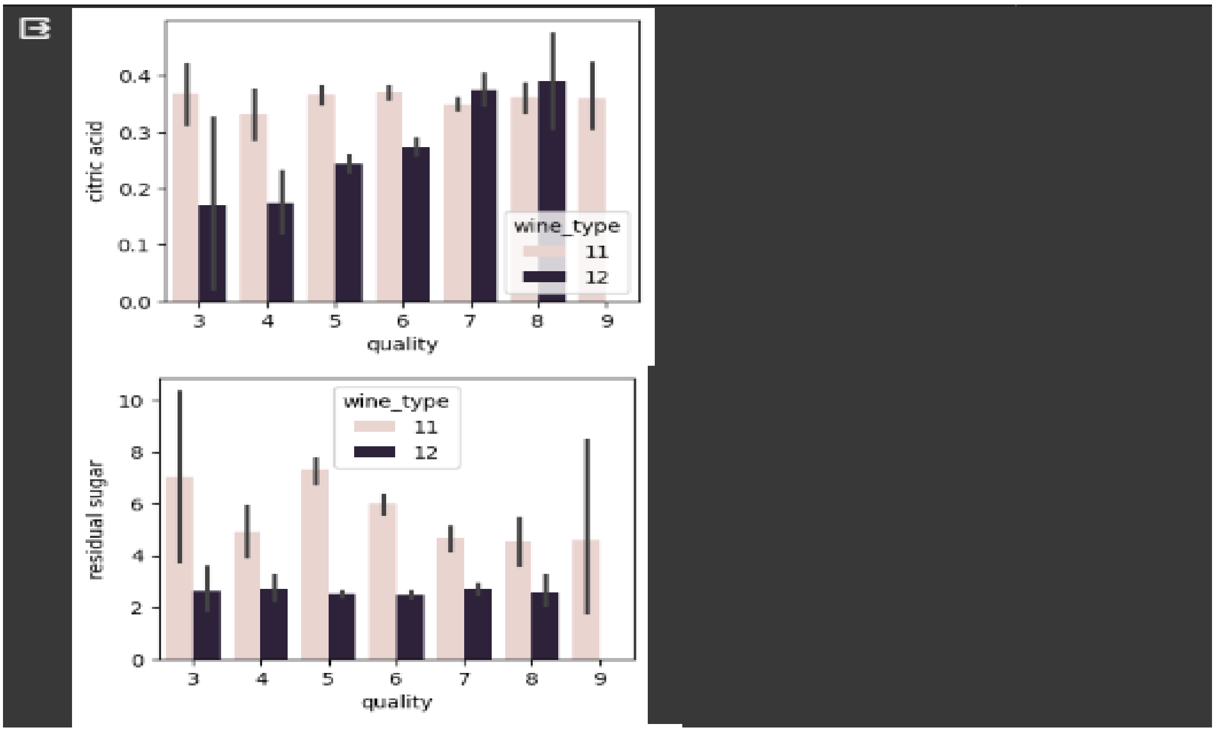


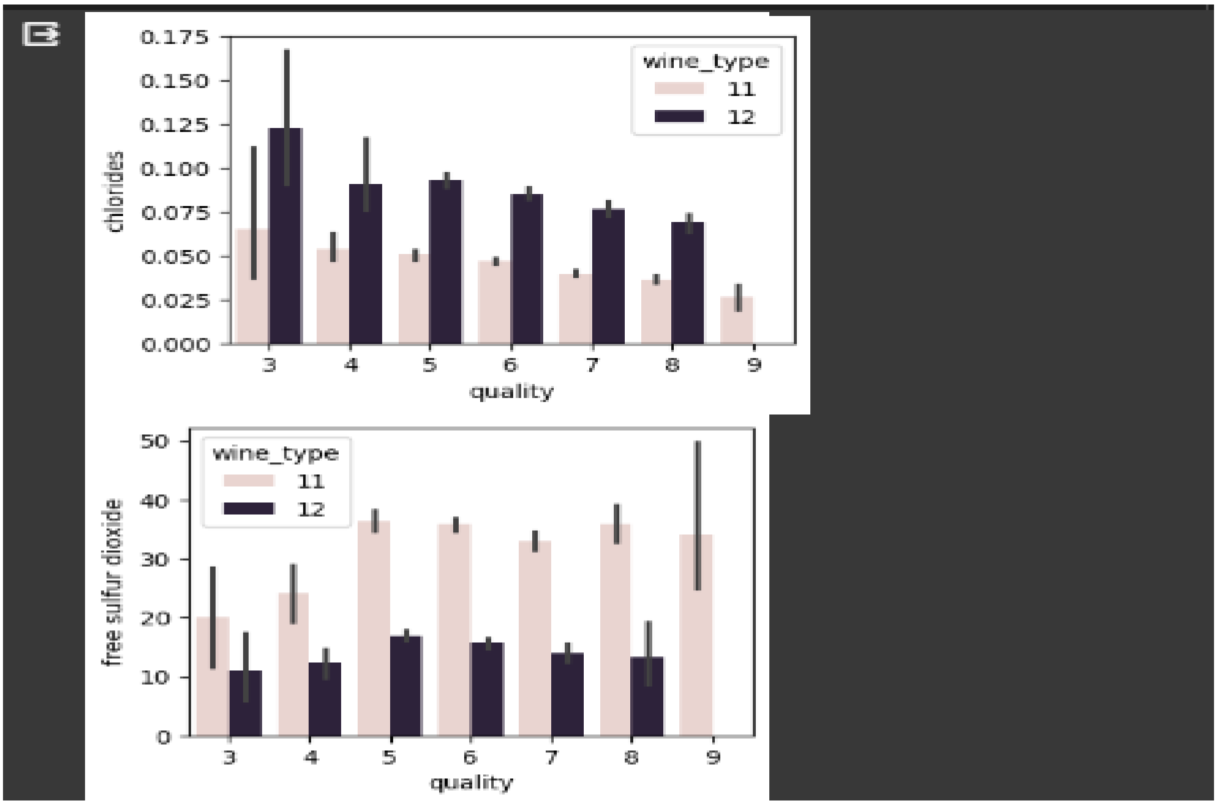
```
#Quality check
sns.catplot(x='quality',data=new_df,kind='count',hue='wine_type')
```

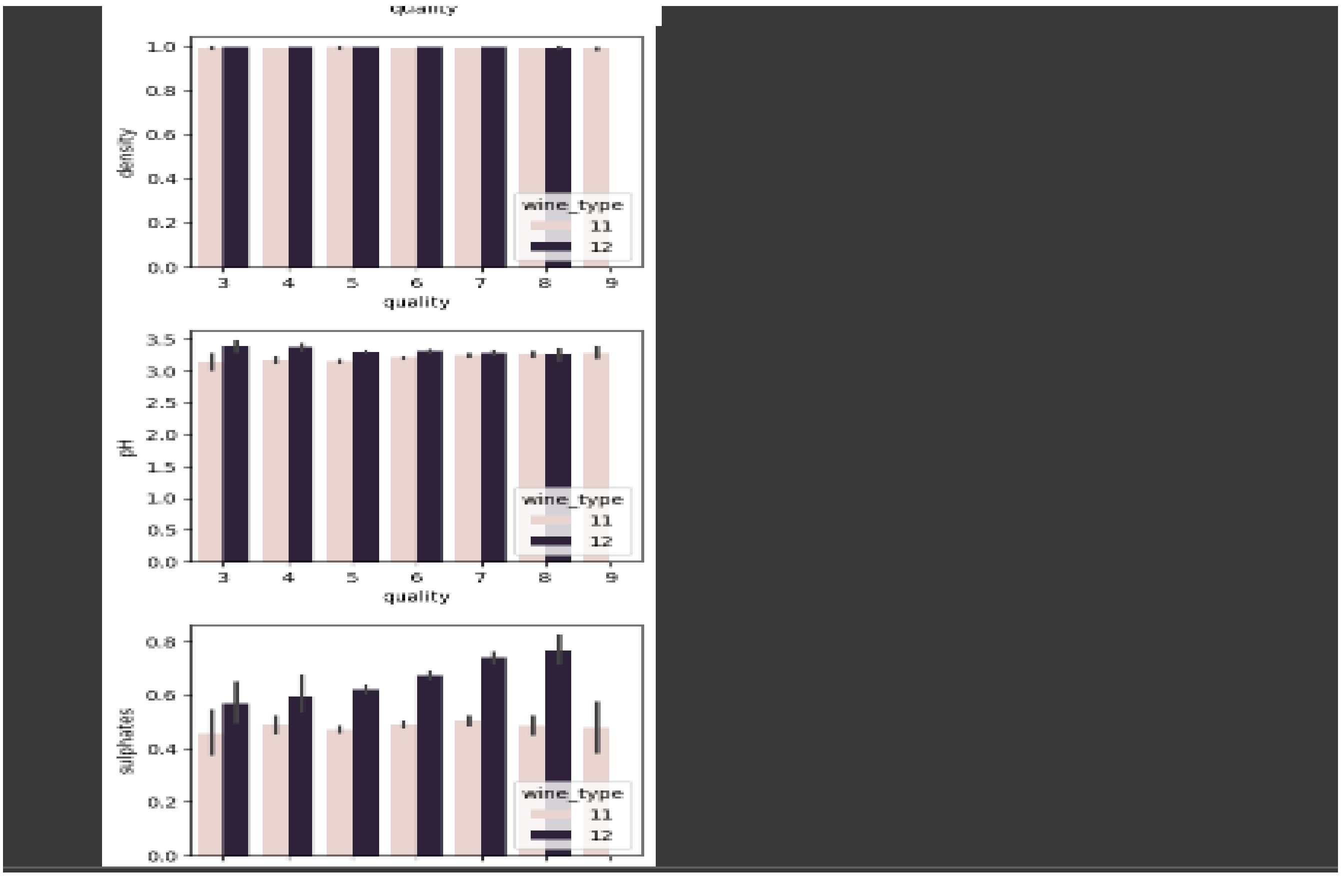
```
<seaborn.axisgrid.FacetGrid at 0x7b8248a4eb60>
```



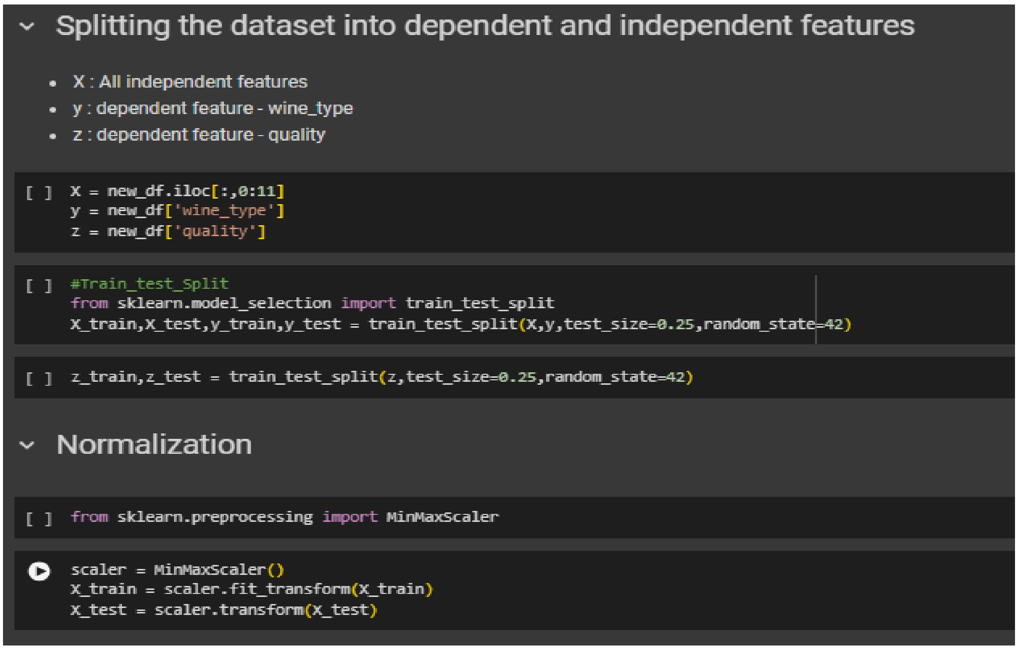
```
[ ]
#comparison of columns with quality
arr = np.array(["fixed acidity","volatile acidity","citric acid","residual sugar",
                "chlorides","free sulfur dioxide","total sulfur dioxide","density","pH","sulphates","alco
for i in arr:
    plot=plt.figure(figsize=(4,3))
    sns.barplot(x="quality",y=i,data=new_df,hue='wine_type')
```


citric acid
0.0
0.1
0.2
0.3
0.4
3
4
5
6
7
8
9
quality
wine_type
11
12
residual sugar
0
2
4
6
8
10
wine_type
11
12
3
4
5
6
7
8
9
quality


chlorides
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
wine_type
11
12
3
4
5
6
7
8
9
quality
free sulfur dioxide
0
10
20
30
40
50
wine_type
11
12
3
4
5
6
7
8
9
quality



3.3 Pre-processing



Splitting the dataset into dependent and independent features

- X : All independent features
- y : dependent feature - wine_type
- z : dependent feature - quality

```
[ ] X = new_df.iloc[:,0:11]
    y = new_df['wine_type']
    z = new_df['quality']
```

```
[ ] #Train_test_Split
    from sklearn.model_selection import train_test_split
    X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.25,random_state=42)
```

```
[ ] z_train,z_test = train_test_split(z,test_size=0.25,random_state=42)
```

Normalization

```
[ ] from sklearn.preprocessing import MinMaxScaler
```

```
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

3.4 Machine Learning Algorithm used

After the Dataset is pre-processed, it is then ready to feed to the Machine Learning Model. We have used Logistic regression, Support Vector Classifier, Random forest, Gaussian Naive Bayes and Ensemble model. The model which performs the best will be used for deployment. We have used classifier models as the target variable is a categorical value. The features selected for prediction are Usage alcohol quantity, fixed acidity, volatile acidity, determination of density, pH . These features were selected based on correlation with target variable and within the features itself.

3.5 Results

Logistic Regression:

We are using logistic regression to classify load types. The model uses a logistic function, which maps any input value to an output value between 0 and 1. This output value represents the probability of the binary outcome being 1. The logistic regression model estimates the coefficients of the independent variables that maximize the likelihood of the observed data given the model. After applying logistic regression our model is giving 0.99 as accuracy score.

The data was split into 75-25% for training data and testing data respectively, with the random state as 42.

Provides us with 0.99 recall for Red Wine & 0.99 for White wine

Random Forest Classifier:

Random Forest Classifier is a popular machine learning algorithm that is widely used for classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make more accurate predictions.

Random Forest Classifier works by building multiple decision trees on random subsets of the training data and random subsets of the features. The decision trees are constructed using a random selection of features at each node to split the data. This helps to reduce the correlation between the trees and improve the overall performance.

The data was split into 75-25% for training data and testing data respectively, with the random state as 42

We used 100 Decision Trees

The accuracy for the model is 0.64125 , we have focused more on recall for class 2(High Load) as our goal is to help the food & wine industry predict t he quality of wine based on its feature values.

Gaussian Naive Bayes :

Gaussian Naive Bayes is commonly used in classification problems with continuous input features, and it can work well with small datasets. However, its assumption of independent features can limit its performance on complex datasets where the features are correlated.

The data was split into 75-25% for training data and testing data respectively, with the random state as 42
Accuracy = 0.46375

Ensemble model :

An ensemble model is a machine learning model that combines the predictions of multiple individual models to improve overall prediction accuracy and robustness. The basic idea behind ensemble modelling is that multiple models can provide better performance than a single model, especially if the individual models have different strengths and weaknesses. Ensemble models are commonly used in machine learning to improve the accuracy and stability of predictions, especially in complex and high-dimensional datasets.

Ensemble models can provide several benefits, including improved prediction accuracy, reduced overfitting, and increased model stability.

However, they can also be more computationally intensive and require more data to train than individual models.

Models used:-Gaussian Naive Bayes,Random Forest

The data was split into 75-25% for training data and testing data respectively, with the random state as 42

Accuracy of mean_squared_error =0.4890625

Support Vector Classifier:

The data was split into 75-25% for training data and testing data respectively, with the random state as 42
Accuracy = 0.9925

4. CONCLUSION AND FUTURE WORK

4.1 Conclusion

After observing the accuracy and performance metrics of the all the models, it is concluded that Logistic Model is the best suited model for the task of predicting wine type & Support Vector Classifier is best for task of predicting quality of wine.

4.2 Future Work

In the future, to improve the accuracy of the individual models using more varied datasets , it is clear that the algorithm or the data must be adjusted. We recommend feature engineering, using potential relationships between wine quality by applying more chemical features , or applying the boosting algorithm on the more accurate method. In addition, by applying the other performance measurement and other machine learning algorithms for the better comparison on results. This study will help the food industries to predict the quality of the different types of wines based on certain features, as it will be helpful for them to make a good product.

5. References:

- https://www.kaggle.com/
- https://realpython.com/logistic-regression
- https://seaborn.pydata.org/
- https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/