Capstone Project 2

Project Title

“Red Wine Quality”

Project Proposed by

Anshul Bhardwaj

Project Mentor

Gritank Dhamija

Problem Statement

The quality of any wine depends on various factors such as amount of alcohol, sugar, acidity, pH, density, sulphates, citric acid, chlorides etc.

By using the wine quality data provided by Kaggle, I created a model which helps in categorizing Wine into High Quality and Low Quality on the basis of various input variables such as –

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

I used various machine learning algorithms such as Logistics Regression, Random Forest Classifier and Xgboost classifier to build the model.

Data Wrangling

The raw dataset from Kaggle contained 1599 rows and 12 columns. It was already a clean dataset containing no empty or NaN values. It had some duplicate rows which were removed.

Removing Duplicate rows

```
df.drop_duplicates(keep = 'first', inplace = True)
```

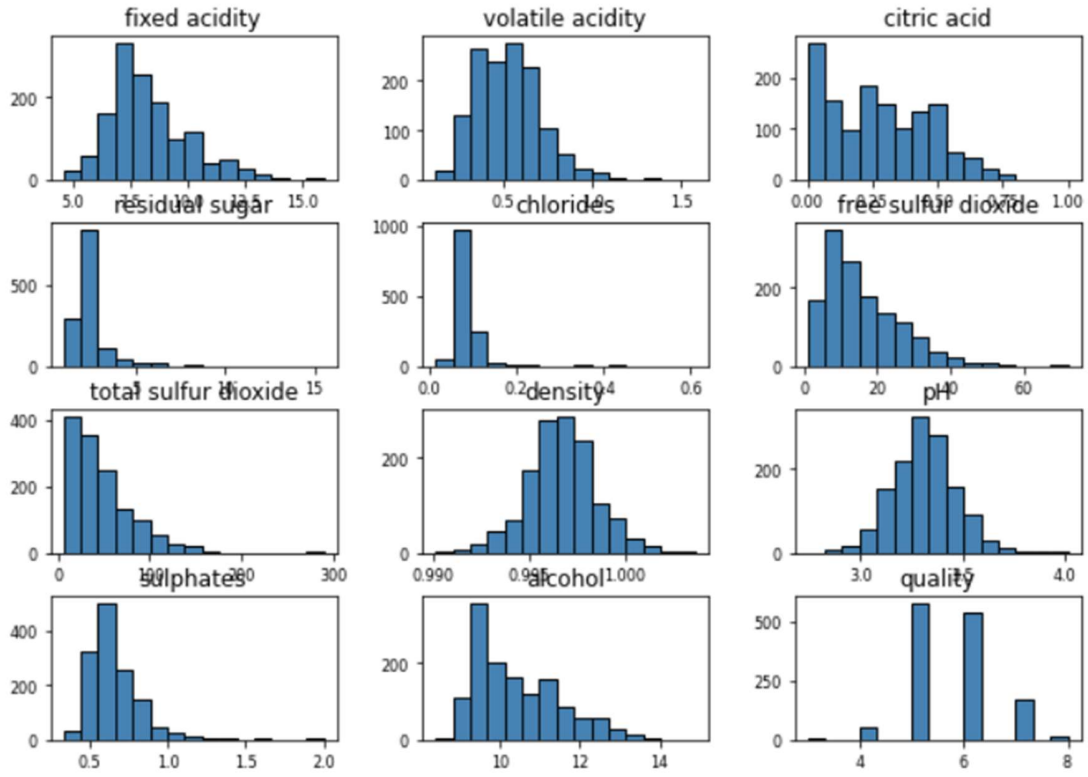
```
df.shape
```

```
(1359, 12)
```

After the removal of duplicate rows our Data Frame was left with 1359 rows and 12 columns.

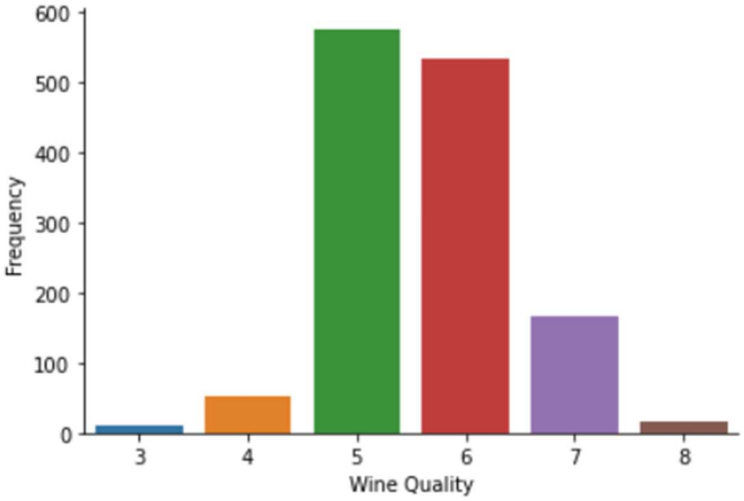
Exploratory Data Analysis

The following figure shows the distribution of various features of Dataset.



The following figure shows the distribution of Wine Quality.

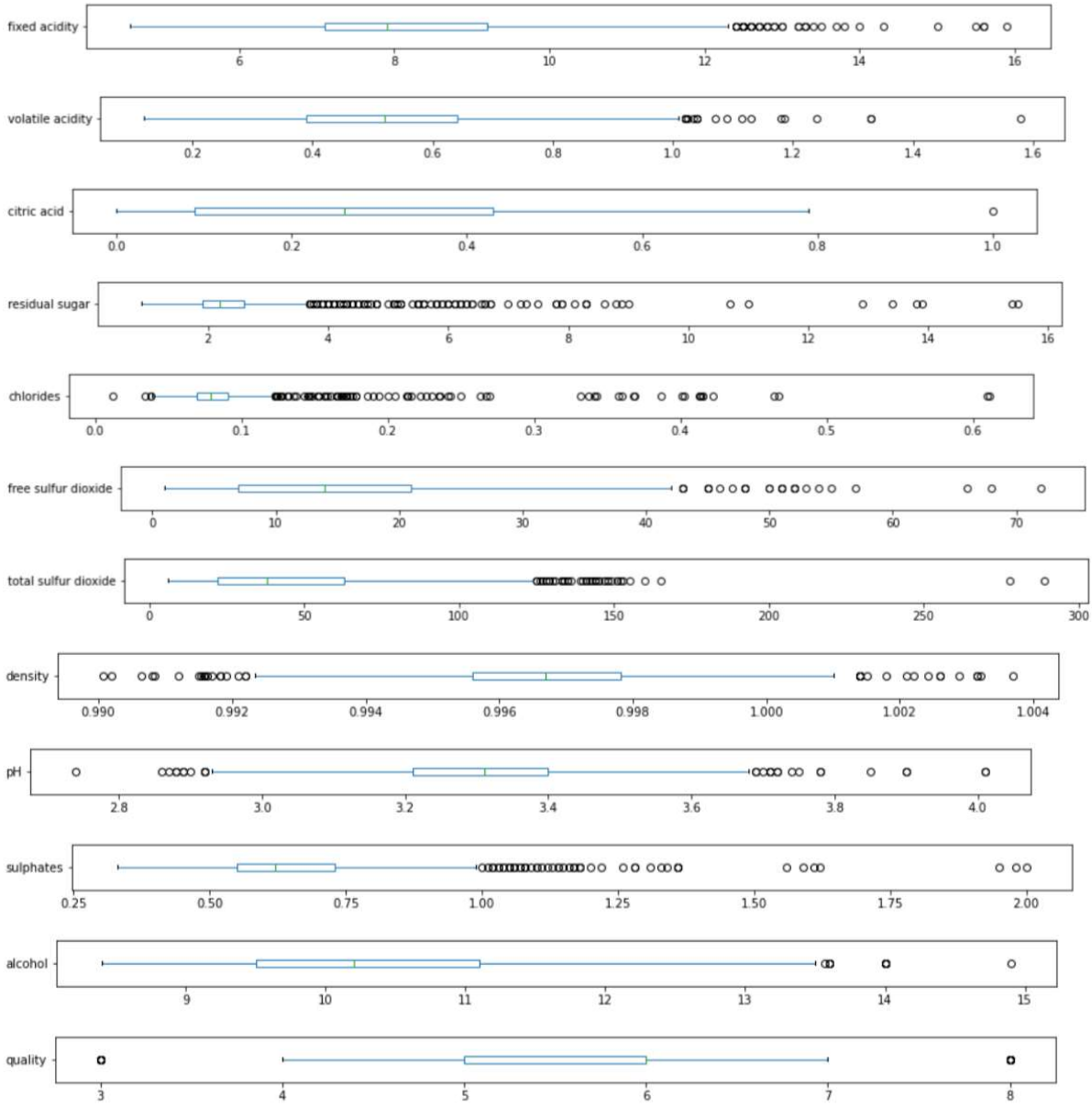
```
In [4]: sns.countplot(data=df,x='quality')
        plt.xlabel('Wine Quality')
        plt.ylabel('Frequency')
        sns.despine()
        plt.show()
```



This shows that in the Wine Quality is ranging from 3 to 8 where 3 signifying the worst and 8 signifying the best quality wine.

The following figure shows the boxplots for all the features in Wine Quality Data Frame.

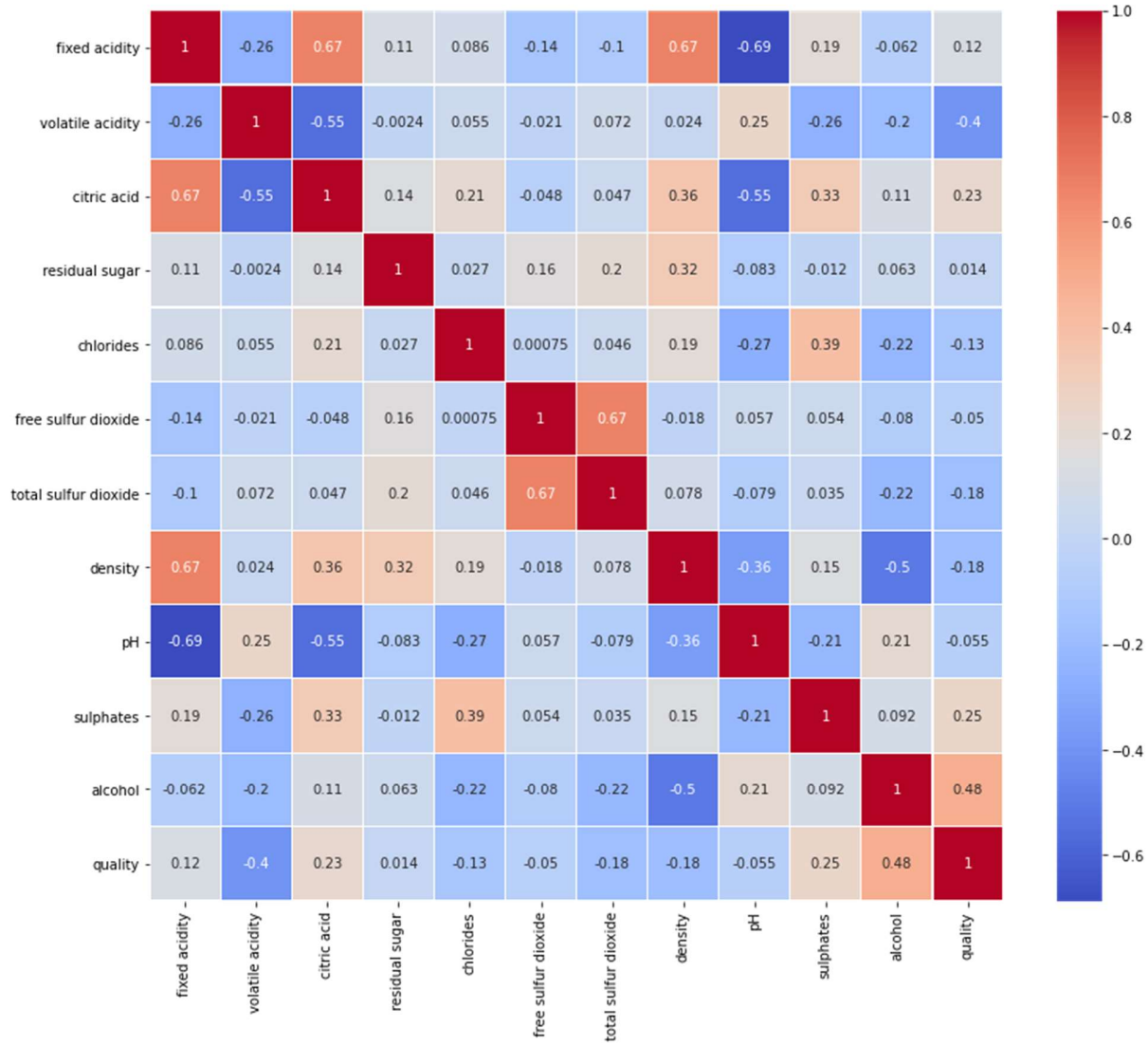
```
for col in df.columns:
    plt.figure(figsize=(15,.7))
    df[[col]].boxplot(grid=False,vert=False)
```



The presence of potential outliers can be clearly seen.

The following figures show the correlation between features.

```
#create the correlation matrix heat map
plt.figure(figsize=(14,12))
sns.heatmap(df.corr(),linewidths=.1,cmap='coolwarm', annot=True)
plt.yticks(rotation=0);
```

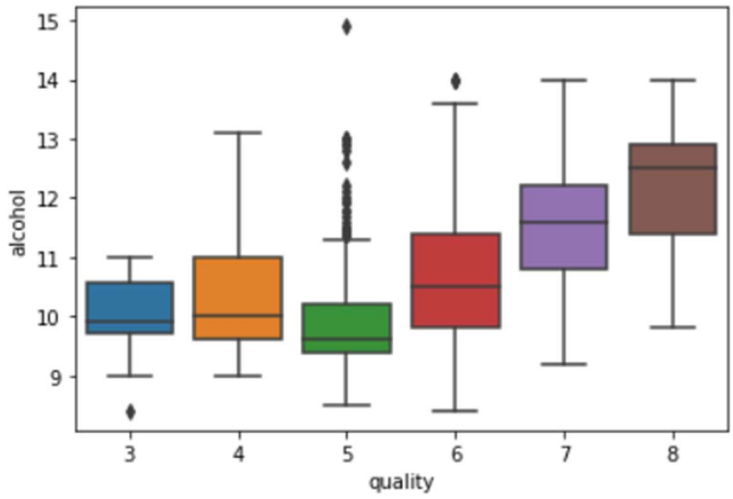


Observations:

- There is high correlation between fixed acidity and citric acid, fixed acidity and density, alcohol and quality, free sulfur dioxide and total sulfur dioxide.
- High Negative correlation is observed between fixed acidity and pH, volatile acidity and quality, volatile acidity and citric acid, citric acid and pH, density and alcohol.
- Very weak correlation between residual sugar and quality

The following figure shows that Higher amount of alcohol leads to better quality.

```
In [58]: sns.boxplot(x='quality', y='alcohol', data=df)
Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x27a13bbe760>
```



The following figure shows that High volatile acidity is bad for Wine Quality.

Preprocessing

Next, I create a new Quality feature from old Quality feature by splitting the old quality in two as follows -

1,2,3,4,5 -> Bad Quality

6,7 -> Good Quality.

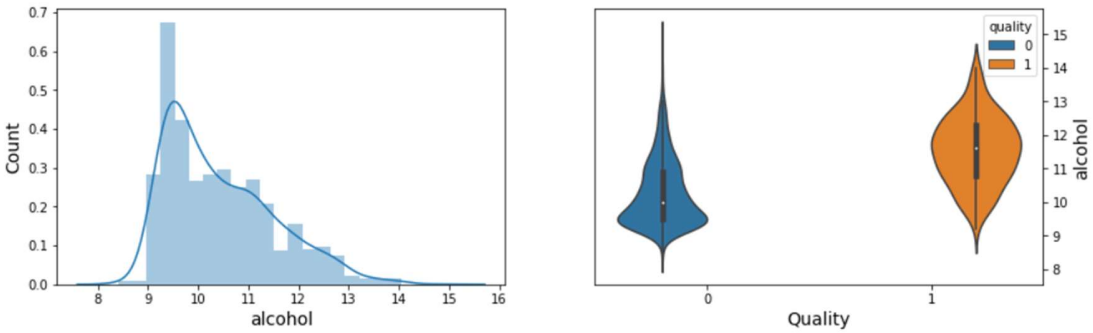
```
:
bins = (2, 6.5, 8)
labels = ['bad', 'good']
df['quality'] = pd.cut(x = df['quality'], bins = bins, labels = labels)
```

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

```
bad     1175
good     184
Name: quality, dtype: int64
```

Out of the total 1359 rows, total of 1175 had bad quality label and rest of 184 had good quality label.

The following figure shows the violin plots for alcohol columns for both label good and bad quality.

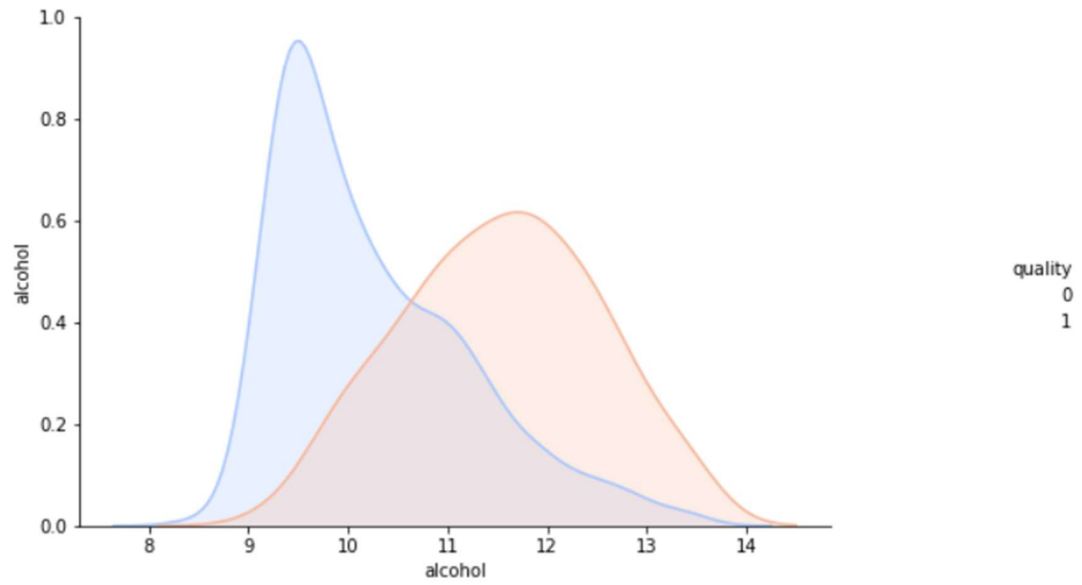


It can be observed that violin plot for good quality is very much different from violine plot for good quality. Hence, I decided to explore this particular feature further.

The figure below depicts the pair plot of alcohol.

```
plt.figure(figsize=(20,10))
g = sns.pairplot(df[['alcohol','quality']], hue ="quality", palette ='coolwarm')
g.fig.set_size_inches(10,5)
```

```
<Figure size 1440x720 with 0 Axes>
```



It can be observed from the figure that wine is very much likely to be good if alcohol amount in it is greater than 12 units. It can also be observed that wine quality is likely to be bad if alcohol amount in it is less than 10 units.

But it will be very ambiguous to say about wine quality if amount of alcohol lies between 10 and 12 units.

Hence, I decided to categorize alcohol feature into 3 categories low, medium and High.

```
#bins = np.linspace(df['alcohol'].min(),df['alcohol'].max(),num=4)
bins = [0,10,12,16]
```

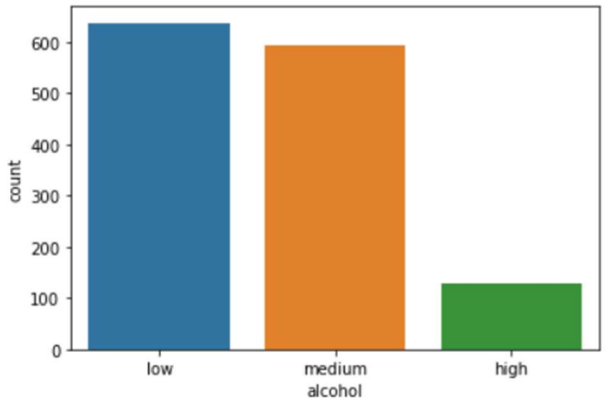
```
bins
```

```
[0, 10, 12, 16]
```

```
labels2 = ['low', 'medium','high']
df['alcohol'] = pd.cut(x = df['alcohol'], bins = bins, labels = labels2)
```

```
sns.countplot(df['alcohol'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x26b6ba926a0>
```




```
df['alcohol'].value_counts()
```

```
low       637
medium    594
high      128
Name: alcohol, dtype: int64
```

Standardization

Features were standardized using StandardScaler()

```
scaler = StandardScaler()
df[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates']] = scaler.fit_transform(df[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates']])
#scaled_df = pd.DataFrame(scaled_df,columns=df.columns)
```

Outlier Treatment

Any row containing a value with Z score high than 3 was removed as part of outlier treatment.

```
from scipy import stats
z = np.abs(stats.zscore(df))
print(z)
```

```
threshold = 3
print(np.where(z > 3))
```

```
df = df[(z < 3).all(axis=1)]
```

```
df.shape
```

```
(1092, 12)
```

Train-Test Split

```
X = df.drop('quality', axis = 1).values
y = df['quality'].values.reshape(-1,1)
```

```
from sklearn.model_selection import train_test_split
X_train0, X_test, y_train0, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

print("Shape of X_train: ",X_train0.shape)
print("Shape of X_test: ", X_test.shape)
print("Shape of y_train: ",y_train0.shape)
print("Shape of y_test",y_test.shape)
```

```
Shape of X_train:  (873, 11)
Shape of X_test:  (219, 11)
Shape of y_train:  (873, 1)
Shape of y_test (219, 1)
```

```
pd.DataFrame(y_train0).value_counts()
```

```
0    753
1    120
dtype: int64
```

SMOTE for Balancing Data

After train test split, the training set has 753 rows for bad quality wine and 120 rows for good quality wine. This is an unbalanced Data set. Hence, I used SMOTE for increasing the number of rows for good quality wine such that the we have half the rows for good quality wine than those that are available for Bad quality wine.

```
oversample = SMOTE(sampling_strategy=0.5, random_state=42)
X_train, y_train = oversample.fit_resample(X_train0, y_train0)
```

```
oversample.get_params()
```

```
{'k_neighbors': 5,
 'n_jobs': None,
 'random_state': 42,
 'sampling_strategy': 0.5}
```

```
from collections import Counter
print(Counter(y_train))
```

```
Counter({0: 753, 1: 376})
```

Finally, we have 753 rows for bad quality wine and 376 rows for good quality wine.

Model Selection

I trained three models on this data. These models were as follows -

1. Logistic Regression
2. Random Forest Classifier
3. Xgboost Classifier.

The metric I followed on building my model was roc_auc score.

The roc_auc score for the models were as follows

Model	roc_auc score
Logistic Regression	.868
Random Forest Classifier	.850
Xgboost Classifier	.887

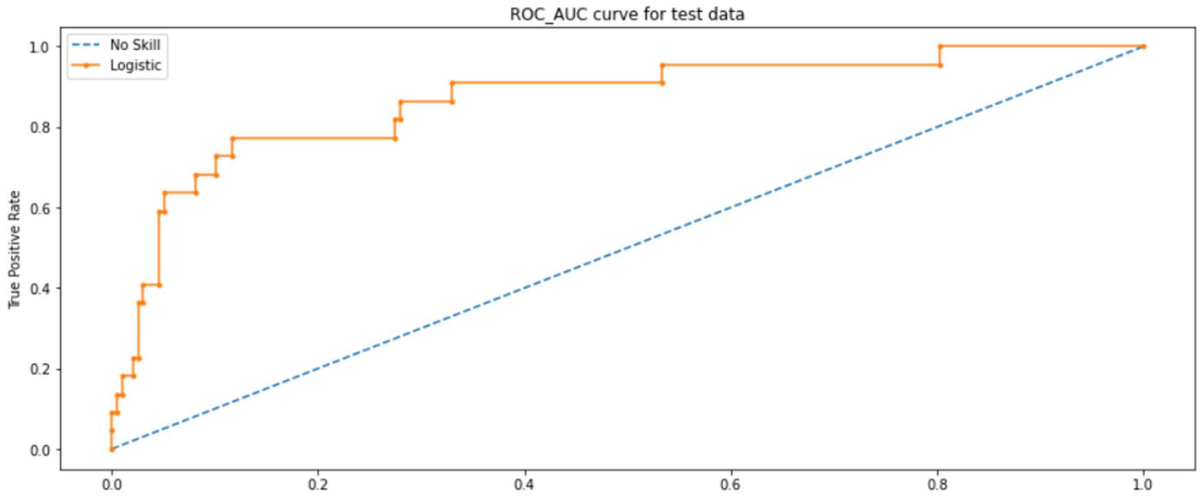
The roc_auc score for the Xgboost Classifier was the highest.

The following is the ROC_AUC curve for **logistic regression**.

Full Report :

	precision	recall	f1-score	support
0	0.92	0.96	0.94	190
1	0.64	0.48	0.55	29
accuracy			0.89	219
macro avg	0.78	0.72	0.74	219
weighted avg	0.89	0.89	0.89	219

```
No Skill: ROC AUC=0.500
Logistic Regression: ROC AUC=0.868
```

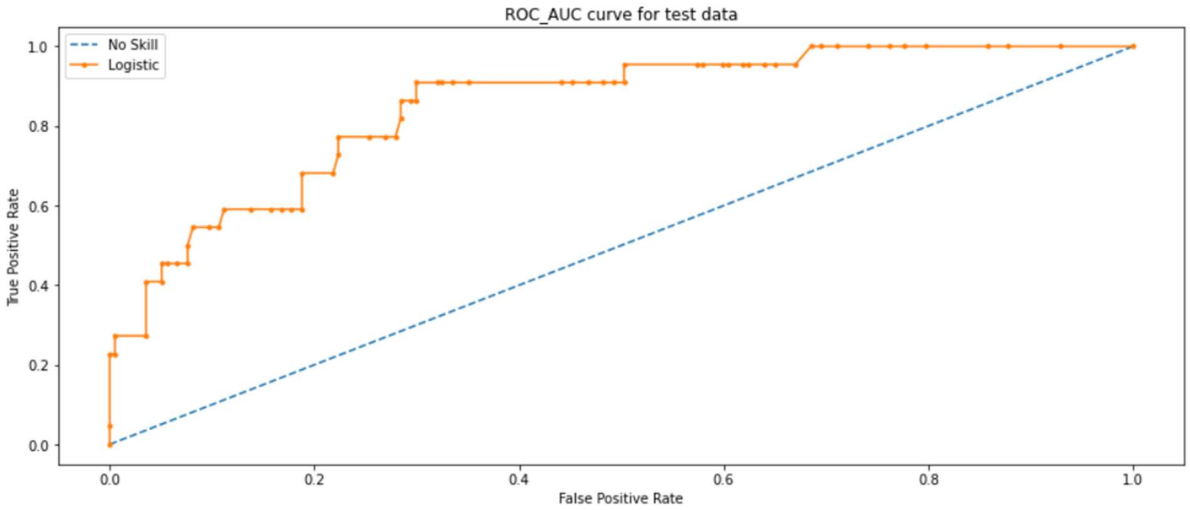


The following is the Model Report and ROC_AUC curve for **Random Forest Classifier**.

Full Report :

	precision	recall	f1-score	support
0	0.96	0.94	0.95	203
1	0.41	0.56	0.47	16
accuracy			0.91	219
macro avg	0.69	0.75	0.71	219
weighted avg	0.92	0.91	0.92	219

```
No Skill: ROC AUC=0.500
Random Forest Classifier: ROC AUC=0.850
```



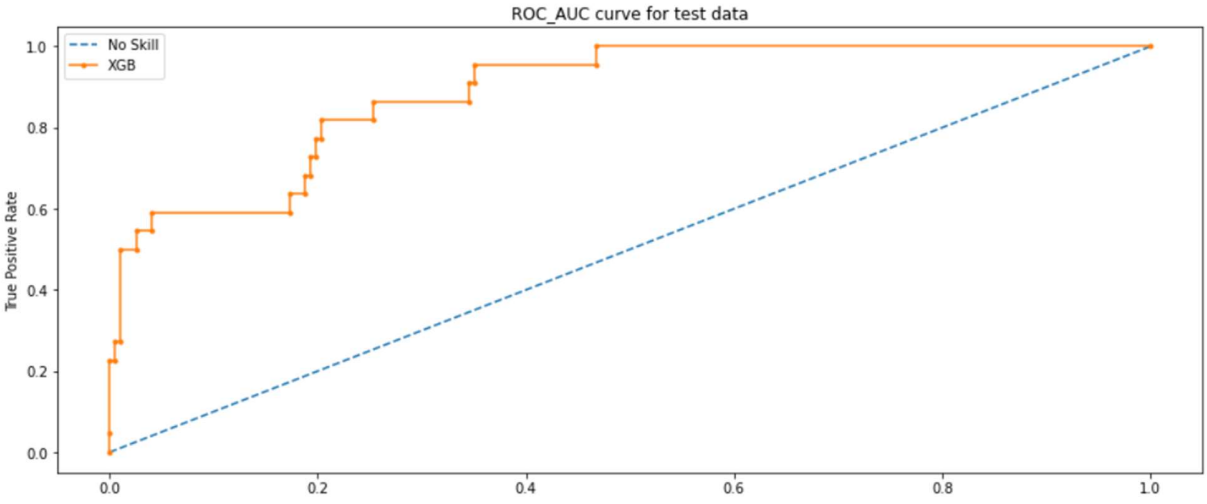
The following is the Model Report and ROC_AUC curve for **Xgboost Classifier.**

```
Full Report :
              precision    recall  f1-score   support

           0       0.96      0.95      0.96       200
           1       0.55      0.63      0.59        19

    accuracy                           0.92       219
   macro avg       0.75      0.79      0.77       219
weighted avg       0.93      0.92      0.92       219
```

```
No Skill: ROC AUC=0.500
XGBOOST: ROC AUC=0.887
```



When it came to the models Xgboost performed the best.

Future Improvements

1. In future I will try to spending more time on doing better outlier detection and removal.
2. I believe that by doing deep analysis of correlation between various features the model results can be improved.