Predicting the quality of Red wine using Linear Regression Model of Machine Learning.

Abstract : Machine Learning is at the heart of modern computational statistics. It allows us to predict results on the basis of a large dataset by prediction a sufficing algorithm and it has done quite well for prediction and statistical analysis. In this report I am proposing better results for wine quality based in Linear Regression. This report proposes the potential better results on the basis of features that are relevant to the study making results more accurate and reasonable with the sensory recipients of humans.

INTRODUCTION:

The red wine industry shows a recent exponential growth as social drinking is on the rise. Nowadays, industry players are using product quality certifications to promote their products. This is a time-consuming process and requires the assessment given by human experts, which makes this process very expensive. Also, the price of red wine depends on a rather abstract concept of wine appreciation by wine tasters, opinion among whom may have a high degree of variability. Another vital factor in red wine certification and quality assessment is physicochemical tests, which are laboratory-based and consider factors like acidity, pH level, sugar, and other chemical properties. The red wine market would be of interest if the human quality of tasting can be related to wine's chemical properties so that certification and quality assessment and assurance processes are more controlled. This project aims to determine which features are the best quality red wine indicators and generate insights into each of these factors to our model's red wine quality.

Problem Statement

The dataset is related to the red variant of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

These datasets can be viewed as regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones). Apply Regression and find the quality of Wine.

RELATED WORK

In previous work [5] on this field, they have stated the problem of quality assessment as a regression model. For this classification they have considered the UC Irvine Dataset of Wine from the Region in southern European country Portugal. Wine certification is usually considered to be assessed by physiochemical properties of the wine but on contrary the taste of wine and quality is in accord to human senses. Thus, it makes hard for us to classify the wines as there are really complex factors that differ taste of wine and relation among the physicochemical properties and sensory analysis are still not fully explained. Progress in Information Technologies have made it easy to store complex and big data. All stored data contains valuable patterns and trends which can be used for further prediction and decision making after optimizing it further. A regression approach should be modeled to conserve order of grades. Performance of regression is usually measured as mean absolute deviation (MAD) and regression models can be used to differentiate different regression models. The results that they produced were relevant to wine science domain, helping to understand on how physicochemical characteristics affects the final quality of the wine. The data-driven approach is done by objective test and can be further integrated into a decision driven support system, improving accuracy and quality of oenologist performance. Second approach on making linear regression model to predict better quality wine was proposed. By looking at the correlation from the heatmap and taking a threshold of 0.05 they've considered 10 features among the 12 and further processed the data. For training and testing the model they've used 30% of the data for testing the model, and 70% of data to train the model. After training the model using Linear Regression model, they've got the scores for RMSE of training data as 0.63 and a R2_Score of 0.31.

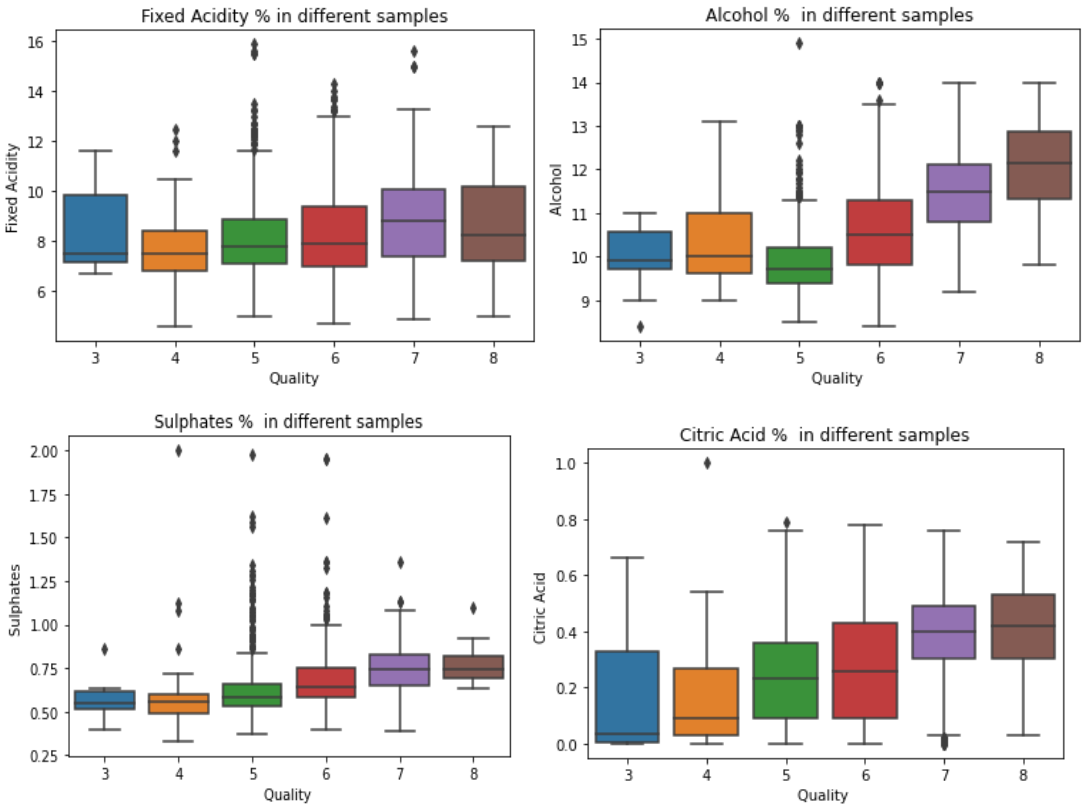
PROPOSED WORK:

In this work we used the Linear Regression Model of Machine Learning to improve the RMSE (Root Mean Square Errors) and MAE (Mean Square Error) for better quality prediction of Red Wine. The Wine Dataset that we have used is from UC Irvine. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Dataset consists of physiochemical (i.e. sulphates, citric acid, etc.) and sensory (quality) variable and these data aren't correlated and hence we will be taking only those features which will be considered as more related to wines taste, aroma and color as human sensory does depends on quality . For formatting proper data, we will take features that have correlation to the human sensory space.

Outlier detection algorithms could be used to detect the few excellent or poor wines to remove their samples from the Dataset as Linear Regression Models is dependent on values nearer to the expected Linear Approach. Also, we are not sure if all input

variables are relevant so we will be comparatively study for the data based on different physiochemical properties.

FIGURE 1:



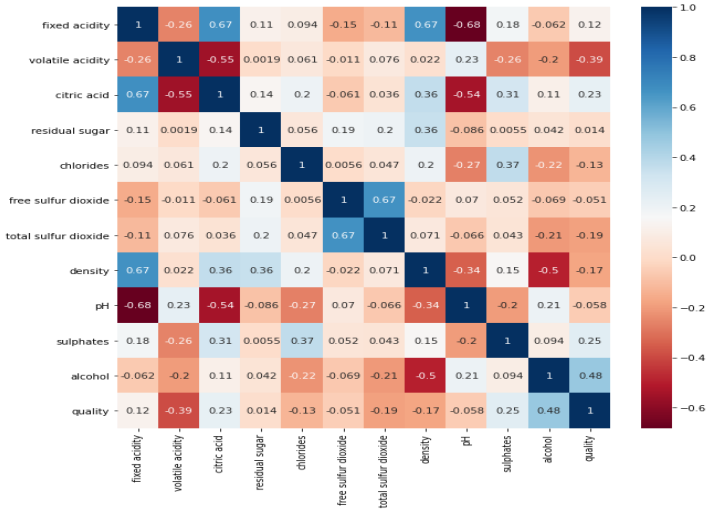


FIGURE 2:

By Observing the Boxplots of various physiochemical properties (Figure 1) we can see the trends in data and it's spread across the whole dataset. In heatmap of the Dataset (Figure 2) we can observe the correlations among the various features of wine. Observing the heatmap we conclude the regular trends in dataset and which can classify the data for relevant processing. From observing the Boxplot, it is clear that some outliers are persistent in the data which can decrease the efficiency of our Model and Hence, we will be using outlier detection algorithm (Interquartile range) and remove them as required for better efficiency. The interquartile range (IQR), technically Hspread, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, IQR = Q3 − Q1. It is a measure of the dispersion similar to standard deviation or variance, hence it is efficient towards detecting outliers from the data. We can use calculated IQR score to filter out the outliers by keeping only valid values and hence proceed further with our data analysis. For training the data we have considered 70% of the all samples and remaining 30% for testing the predicted values later. Then after applying ordinary least squares Linear Regression and fitting the data we have trained our model. Then after getting regressor intercept and regression coefficients of features, we test our model for the quality prediction using our trained model and testing it our test data. On analyzing the test results, we found the results (Figure 3) for our newly trained Model. As from graph all the necessary error correction from previous models can be seen.

SOFTWARE:

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

1. Zero configuration required
2. Free access to GPUs
3. Easy sharing

Colab Input File:

Hence to prose the data I have used two data fules for detection which is

1. Minor Project(Sumaya)Related Work.ipynb
2. Minor Project (Sumaya)-Improving R2 Score.ipynb-Proposed Work(Improving R2 score using outlier algorithm)

ALGORITHM:

What is Regression?

The main goal of regression is the construction of an efficient model to predict the dependent attributes from a bunch of attribute variables. A regression problem is when the output variable is either real or a continuous value i.e salary, weight, area, etc.

We can also define regression as a statistical means that is used in applications like housing, investing, etc. It is used to predict the relationship between a dependent variable and a bunch of independent variables. There are various types of regression techniques.

Types Of Regression

The following are types of regression.

Simple Linear Regression

Polynomial Regression

Support Vector Regression

Decision Tree Regression

Random Forest Regression

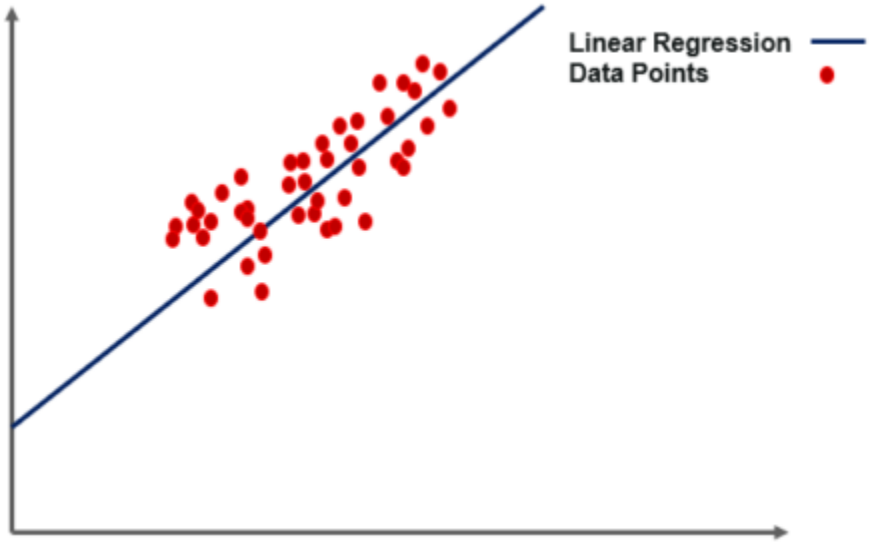
Simple Linear Regression

One of the most interesting and common regression technique is simple linear regression. In this, we predict the outcome of a dependent variable based on the independent variables, the relationship between the variables is linear. Hence, the word linear regression.

What is Linear Regression?

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and

the number of independent variables being used.



Use Case – Implementing Linear Regression

The process takes place in the following steps:

1. Importing the Data
2. Exploring the Data
3. Slicing The Data
4. Train and Split Data
5. Generate The Model
6. Evaluate The accuracy

RELATED WORK

USE CASE-STEPS :

1.IMPORTING DATA

First, I imported all of the relevant libraries that I'll be using as well as the data itself.

Importing Libraries

We can start with the Wine quality data set that is already present in the sklearn(scikit-learn) data sets module to begin our journey with linear regression.

```
import pandas as pd
import seaborn as sea
import matplotlib.pyplot as plt
import numpy as np
```

Reading Data

```
df = pd.read_csv("/content/winequality-red.csv")
```

Data Understanding

My analysis will use Red Wine Quality Data Set, available on the UCI machine learning repository (https://drive.google.com/file/d/1Jiee6bvIgMOGIOwpg_DxYpk4fwZaJpWE/view?usp=sharing). I obtained the red wine samples from the north of Portugal to model red wine quality based on physicochemical tests. The dataset contains a total of 12 variables, which were recorded for 1,599 observations. This data will allow us to create different regression models to determine how different independent variables help predict our dependent variable, quality. Knowing how each variable will impact the red wine quality will help producers, distributors, and businesses in the red wine industry better assess their production, distribution, and pricing strategy.

```
[3] df
```

	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

1599 rows × 12 columns

Columns Description:

- **Fixed Acidity** : Amount of Tartaric Acid in wine, measured in g/dm^3
- **Volatile Acidity** : Amount of Acetic Acid in wine, measured in g/dm^3
- **Citric Acid** : Amount of citric acid in wine in g/dm^3. Contributes to crispness of wine.
- **Residual Sugar** : amonunt of sugar left in wine after fermentation. Measured in in g/dm^3
- **Chlorides** : amount of Sodium Cholride (salt) in wine. Measured in g/dm^3
- **Free Sulfur Dioxide** : Amount of SO_2 in free form. Measured in mg/dm^3
- **Total Sulfur Dioxide** : Total Amount of SO_2. Too much SO_2 can lead to a pungent smell. SO_2 acts as antioxidant and antimicrobial agent.
- **Density** : Density of Wine in g/dm^3
- **pH** : pH of Wine on a scale of 0-14 . 0 means highly Acidic, while 14 means highly basic.
- **Sulphates** : Amount of Potassium Sulphate in wine, measured in g/dm^3.Contributes to the formation of SO_2.
- **Alcohol** : alcohol content in wine (in terms of % volume)
- **Quality** : Wine Quality graded on a scale of 1 - 10 (Higher is better)

Missing 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
quality                 0
dtype: int64
```

This is a very beginner-friendly dataset. I did not have to deal with any missing values, and there isn't much flexibility to conduct some feature engineering given these variables. Next, I wanted to explore my data a little bit more.

Data Preparation

Data Cleaning

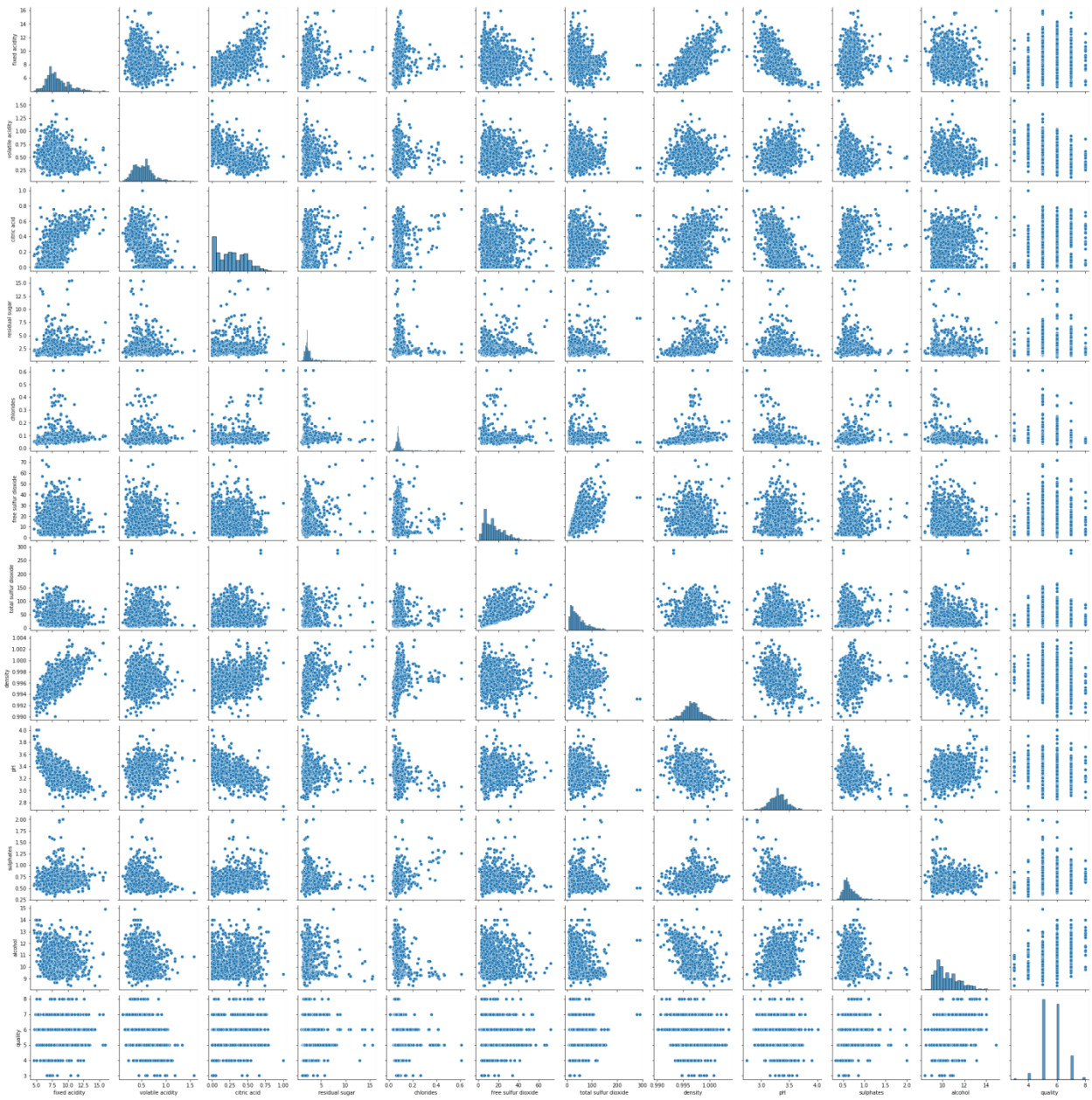
My first step was to clean and prepare the data for analysis. I went through different steps of data cleaning. First, I checked the data types focusing on numerical and categorical to simplify the correlation's computation and visualization. Second, I tried to identify any missing values existing in our data set. Last, I researched each column/feature's statistical summary to detect any problem like outliers and abnormal distributions.

```
[6] df.describe()
```

	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

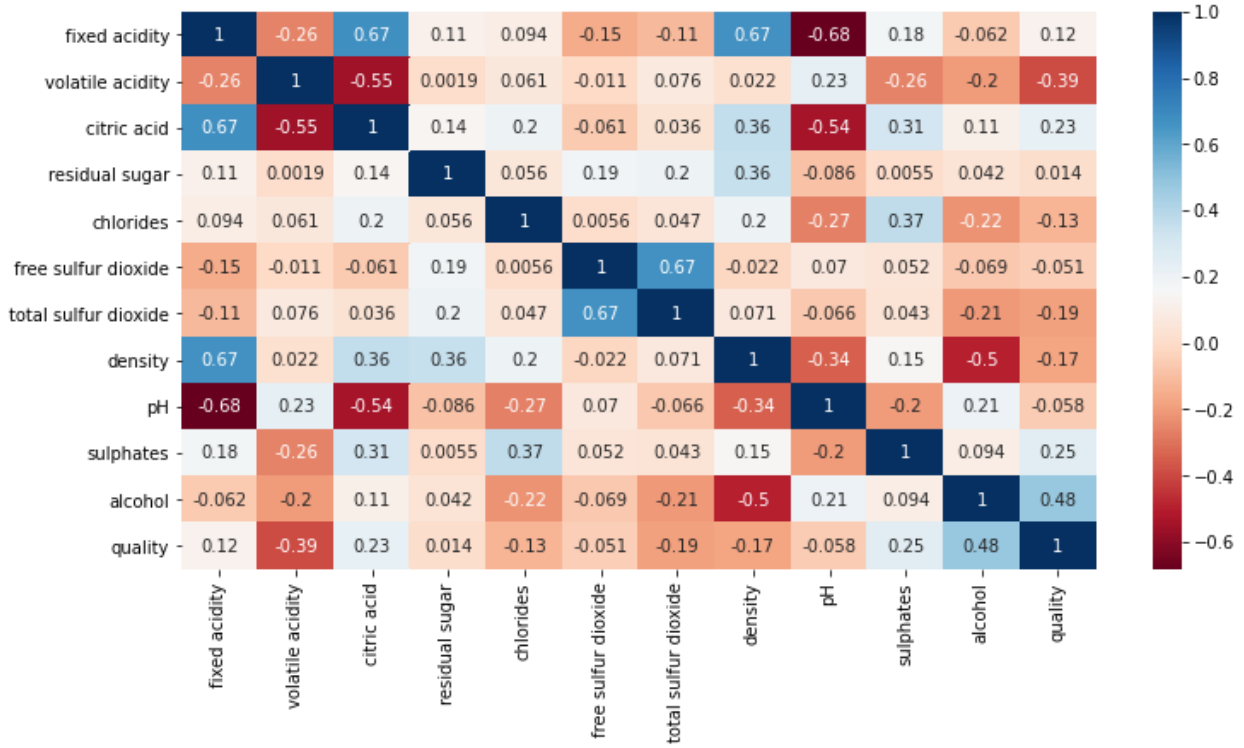
2.EXPLORATORY DATA ANALYSIS

After we are done importing the data, we can start exploring by simply checking the labels by using the following code.



Correlation Matrix

To see which variables are likely to affect the quality of red wine the most, I ran a correlation analysis of our independent variables against our dependent variable, quality. This analysis ended up with a list of variables of interest that had the highest correlations with quality.



In order of highest correlation, these variables are:

1. Alcohol: the amount of alcohol in wine

2. Volatile acidity: are high acetic acid in wine which leads to an unpleasant vinegar taste

3. Sulphates: a wine additive that contributes to SO2 levels and acts as an antimicrobial and antioxidant

4. Citric Acid: acts as a preservative to increase acidity (small quantities add freshness and flavor to wines)

5. Total Sulfur Dioxide: is the amount of free + bound forms of SO2

6. Density: sweeter wines have a higher density

7. Chlorides: the amount of salt in the wine

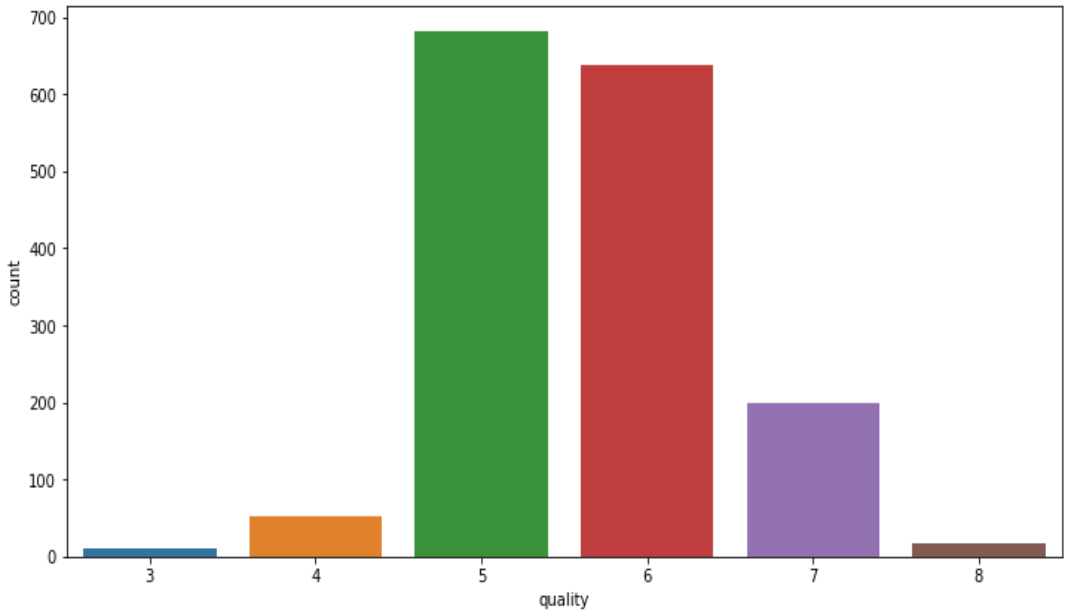
8. Fixed acidity: are non-volatile acids that do not evaporate readily

9. pH: the level of acidity

10. Free Sulfur Dioxide: it prevents microbial growth and the oxidation of wine

11. Residual sugar: is the amount of sugar remaining after fermentation stops. The key is to have a perfect balance between — sweetness and sourness (wines > 45g/ltrs are sweet)

Starting with our dependent variable, quality, I found the popularity of the medium/average values of quality: 5 and 6. Considering the dependent variable's transformation, I found out that our data is normally distributed. This conclusion can be verified by running a count plot, which shows no need to transform our data.



Let us have a look at dataset and try to identify the Independent and Dependant variables. From the dataset we can see that parameters such as fixed acidity, volatile acidity, citric acid, residual sugar all the way till alcohol are used in making the red wine and hence they are our Independent variables. The last column, quality is dependent on the ingredients and is our Dependent variable or the target.

3. SPLIT THE DATA IN Train and Test Data

Now, we are ready to split the data into the training set and the test set. Our data consists of 1600 entries, so we can the split the data in the 80 : 20 ratio for the training set and test set respectively. Our model uses the training set to train itself in order to predict the test set values.

Using the train_test_split class from the scikit-learn library to split our dataset.

```
Splitting the dataset into Training Set and Test set

[20] from sklearn.model_selection import train_test_split

[21] train_x,test_x,train_y,test_y = train_test_split(x,y, test_size = 0.2, random_state = 0)
```

```
[44] train_x

array([[0.46428571, 0.28767123, 0.45      , ..., 0.51181102, 0.15337423,
        0.15384615],
       [0.54464286, 0.09589041, 0.45      , ..., 0.30708661, 0.10429448,
        0.18461538],
       [0.46428571, 0.15753425, 0.55      , ..., 0.40944882, 0.25766871,
        0.33846154],
       ...,
       [0.28571429, 0.30821918, 0.31      , ..., 0.43307087, 0.19631902,
        0.16923077],
       [0.74107143, 0.23972603, 0.49      , ..., 0.44094488, 0.19018405,
        0.66153846],
       [0.45535714, 0.5890411 , 0.32      , ..., 0.4015748 , 0.06748466,
        0.15384615]])

[45] test_x

array([[10.8  ,  0.47 ,  0.43 , ...,  3.17 ,  0.76 , 10.8  ],
       [ 8.1  ,  0.82 ,  0.   , ...,  3.36 ,  0.53 ,  9.6  ],
       [ 9.1  ,  0.29 ,  0.33 , ...,  3.26 ,  0.84 , 11.7  ],
       ...,
       [ 9.1  ,  0.34 ,  0.42 , ...,  3.18 ,  0.55 , 11.4  ],
       [ 9.1  ,  0.765,  0.04 , ...,  3.29 ,  0.54 ,  9.7  ],
       [ 8.2  ,  0.32 ,  0.42 , ...,  3.27 ,  0.55 , 12.3  ]])
```

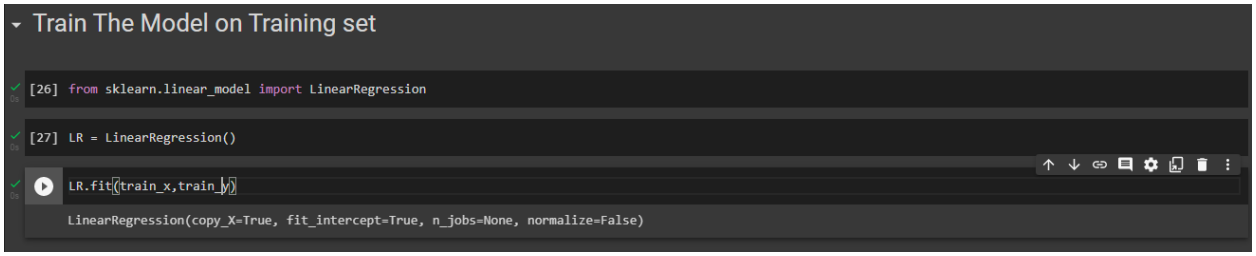
Now, lets apply feature scaling to our training set and test set. Using the MinMaxScaler class, we scale our training and test set.

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

[23] scaler = MinMaxScaler()

[24] train_x = scaler.fit_transform(train_x)
     test_x = scaler.fit_transform(test_x)
```

At this point, we have completed the data preprocessing step and are ready to construct our Machine learning model.

4. GENERATING THE MODEL

Let us build the **Multiple Linear Regression model**

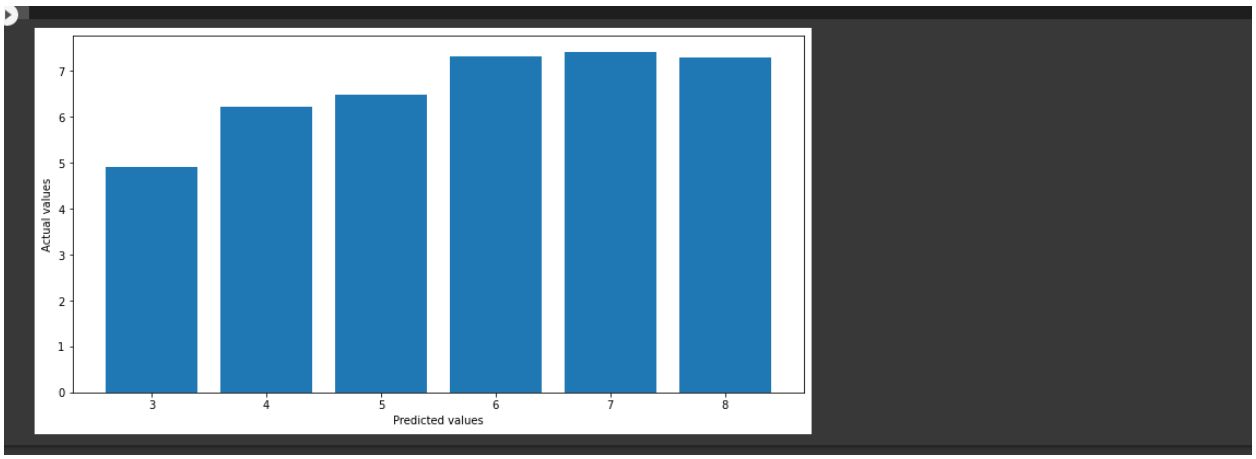


Again, from the scikit-learn we import the LinearRegression class. We create an object called LR for the LinearRegression class that we imported. Now, we fit this object onto our training set which contains independent variables and the dependent variable.

At this point, we have successfully applied Multiple Linear Regression machine learning model to our dataset.

Our model is now ready to predict the test set values.

Let us have a look at the **predicted values and the actual values.**



As we can see from the results, that our model is predicting the actual values quite well. There are some values which are far from the test set value but still our model does a good job at predicting most of the values.

5. EVALUATION

Now let us evaluate the performance of our model using Mean squared error criterion from the scikit-learn..

```
from sklearn.metrics import mean_absolute_error as mae
from sklearn.metrics import mean_squared_error as mse
mae(test_Y,pred_Y)
0.4685914604980187
```

```
mse(test_Y,pred_Y)
0.4054957106125317
```

```
rmse = np.sqrt(mse(test_Y, pred_Y))
print('Root Mean Squared Error:',rmse)
Root Mean Squared Error: 0.6367854510056992
```

```
from sklearn.metrics import r2_score
r2_score(test_Y,pred_Y)
0.312719134555031
```

We can see that our mean squared error is 0.405. This indicates that there is scope for improvement and other Machine learning should be explored to get a better MSE value.

Finally the accuracy of the model is 37%

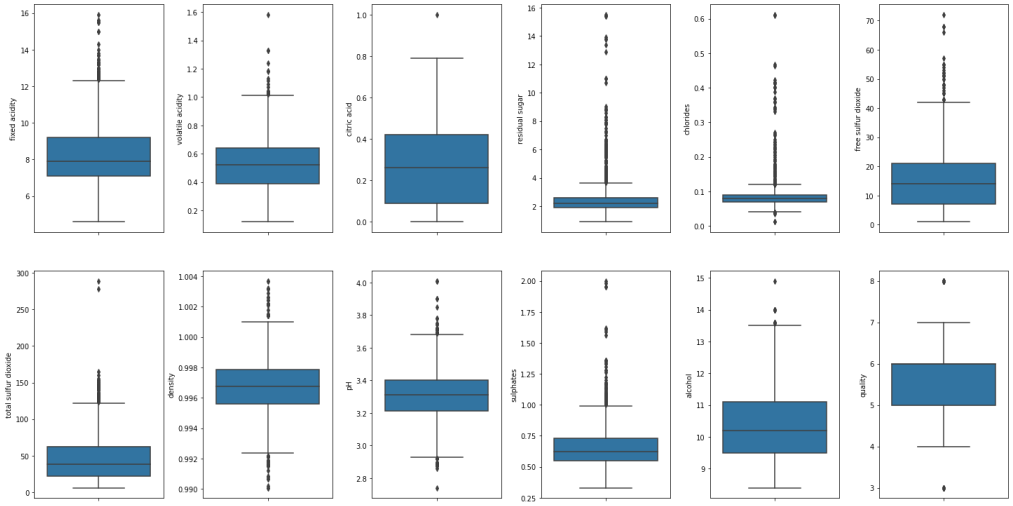
```
45] accuracy = LR.score(train_x,train_y)
    print("Accuracy: {}%".format(int(round(accuracy*100))))

    Accuracy: 37%
```

Now, in order to identify the ingredients that impact the quality of wine most, I used Backward Elimination approach on this model.

Proposed Work:

IMPROVING THE R2 SCORE AND ACCURACY OF THE MODEL :

STEP 1: This figure indicates the outliers in the data



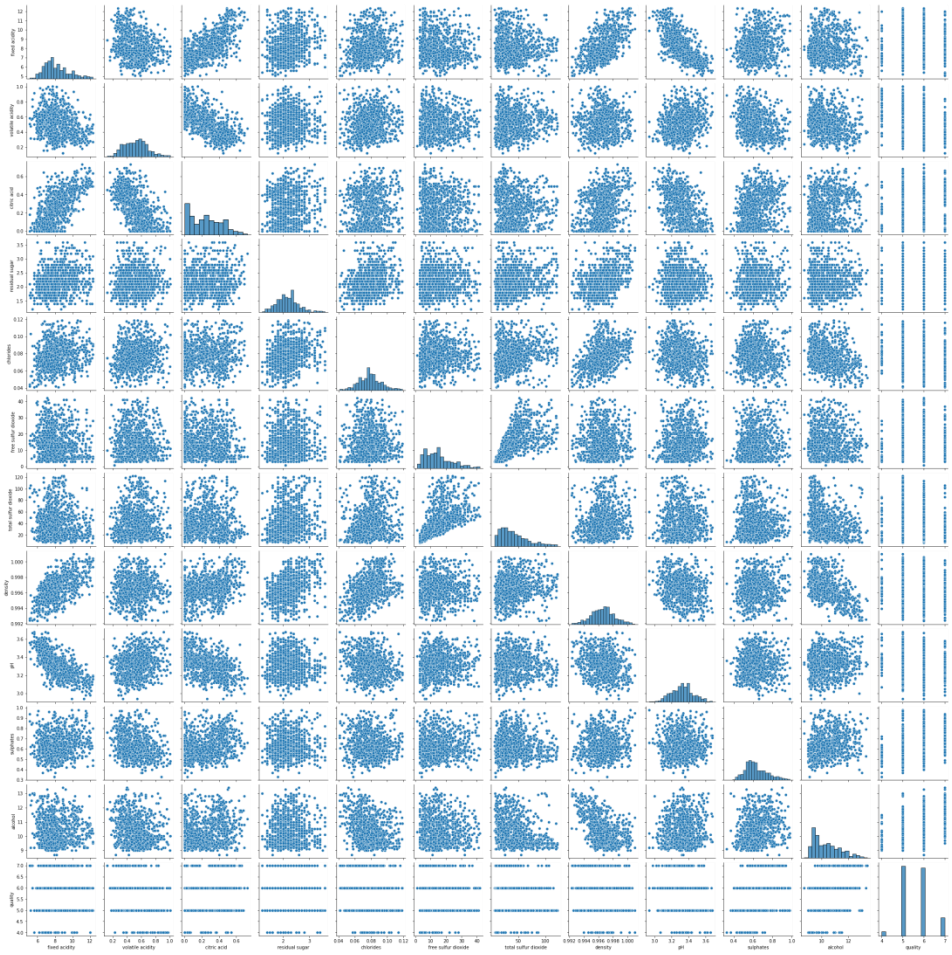
STEP 2:REMOVING the OUTLIERS from the DATA by using this technique.

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
df_out = df[~((df < (Q1 -
 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR))).any(axis=1)]

df_out.shape
```

```
sorted data after the outliers are removed.So ,there are 1179 entries and
12 columns.
```

```
(1179, 12)
```

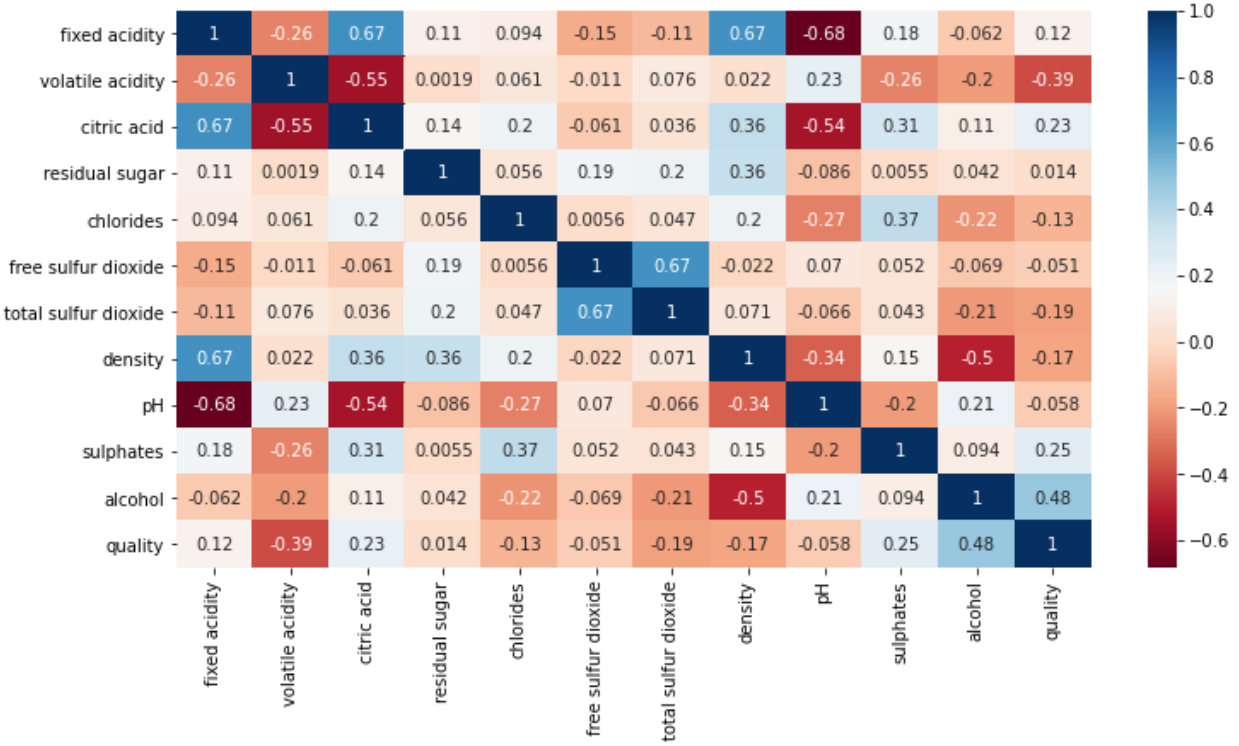
EDA



We can see that there is a normal disturbution and clearly see that outliers are removed from dataset.

STEP 3:

CORRELATION MATRIX:



STEP 3:

taking features with correlation more than 0.05 as input x and quality as target variable y

```
features = get_features(0.05)
print(features)
x = df_out[features]
y = df_out['quality']
```

```
['fixed acidity', 'volatile acidity', 'citric acid', 'chlorides', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']
```

STEP 5: SPLIT THE DATA:

```
train_x,test_x,train_y,test_y = train_test_split(x,y, test_size = 0.3, ran
dom_state = 0)
```

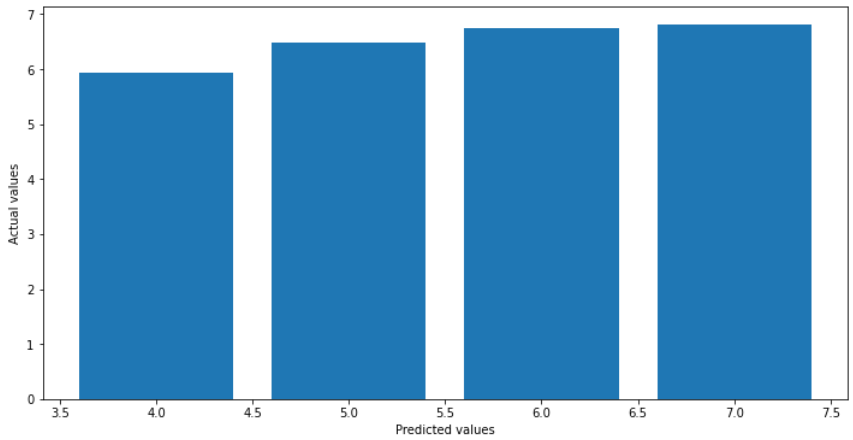
STEP 6: APPLYING SCALER

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
train_x = scaler.fit_transform(train_x)
test_x = scaler.fit_transform(test_x)
```

STEP 7: LINEAR REGRESSION MODEL

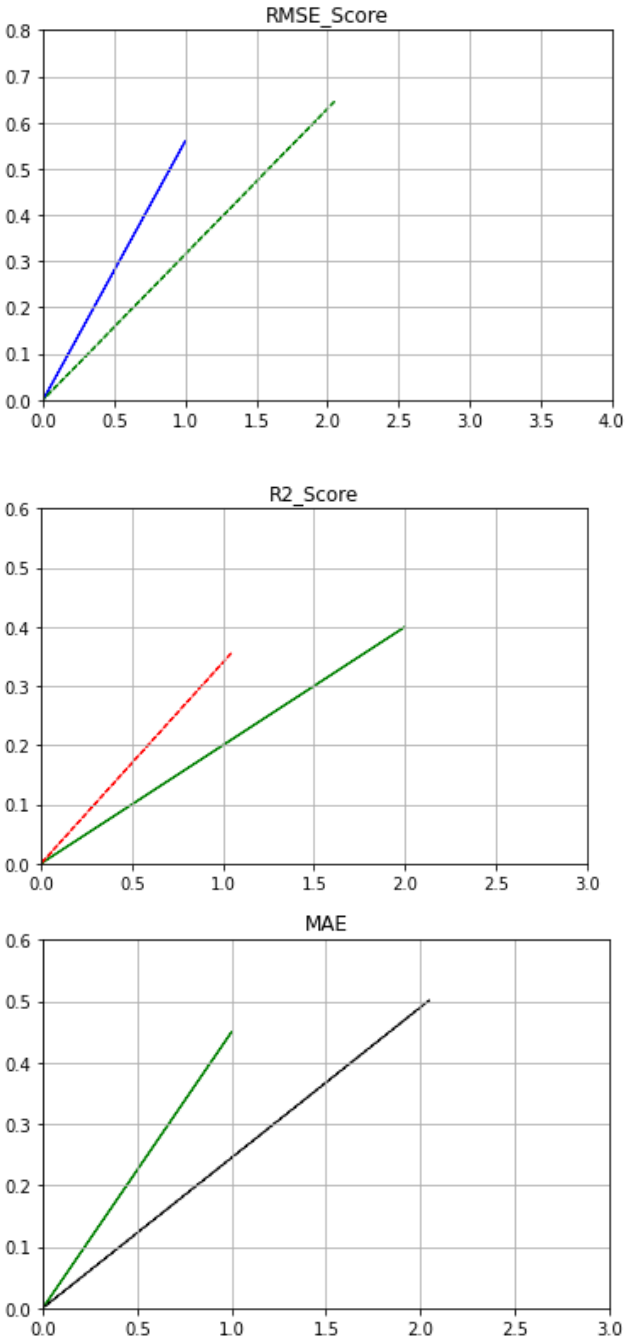
```
from sklearn.linear_model import LinearRegression
LR = LinearRegression()
LR.fit(train_x,train_y)
pred_y = LR.predict(test_x)
```

STEP 8:EVALUATION



RESULT:

FIGURE 3:



These results which were produced are beyond previous works using linear regression or SVM model, showing an increase in prediction quality drastically. Contrary to the findings using Linear Regression Model we did not find any efficiency drops with previous Linear Regression approaches. Since Linear Regression uses a degree 1

approach to fit the model it is very sensitive to outliers and because of this potential limitation, we treat the data with outlier detection Algorithms. Apart from the Limitations of Linear Regression Model it is a considerable prediction quality accuracy with RMSE of `0.5670861234989516` and a R2_Score of `0.40704840254077124`. Regarding the limitations of Regression Model, it could be argued that quality of wine is also dependent on various intrinsic factors such as pleasure, balance, drinkability, and many more are a good measure for quality of wine. Other dimensions such as involvement with wine, physical location of place, price, affordability also can we concludes from marketing point of view for further spread of wine for better quality production.

CONCLUSIONS :

Wine is a worldwide alcoholic beverage that is gaining it's popularity by the day and reaches millions of people for consumption. Due this increasing popularity and demand the markets in various continents are looking for better quality wines at reasonable cost and better profits to meet the modern needs. In developing countries, due to developing infrastructure it is hard to set up labs and surveys for production of better quality of wine to meet demands. So, machine learning based on a large dataset is a better measure for prediction of better-quality wine as it will add up to more accuracy and cut down the production and testing costs considerably based on the chemical properties of the wine itself. In this work we found the accuracy from Linear Regression model of 40% to produce better quality wine it has to considered that data classification is based on different measure of quality and accuracy can increase considerable if quality measures are decreased to just Good or Bad wine. But inclusion of various other factors in data can lead us to better prediction model and Further more accurate model can be presented using Linear Regression Itself.