RED WINE QUALITY ANALYSIS

<u>TEAM MEMBERS:</u>

KRUTI SHAH (306650284)

RAVI AMIN (306598765)

RIDDHIBEN PATEL (306612701)

RUTVIBEN PATEL (306019563)

SMITKUMAR KAUSHIKKUMAR PATEL (306587208)

DATA SET

wineQualityReds.csv

Data set link : https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

PROJECT DESCRIPTION

- It shows relation between quality and other variables of wine. We want to do transformation to see if we can increase correlation coefficient between them. Used stepwise variable selection method to choose best predictor of wine quality.

GOAL

- Our focus is to see how each chemical component influences the quality of wine (0 'very bad' to 10 'very excellent'). The usage of this analysis will help to understand whether by modifying the variables, it is possible to increase the quality of the wine on the market.

ABOUT THE DATASET

- In this project we do Analysis of **Red Wine Data** which contains 1,599 red wines with 12 variables on the chemical properties of the wine.

INPUT VARIABLES

- Fixed acidity: most acids involved with wine or fixed or nonvolatile (do not evaporate readily)
- Volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste
- Citric acid: found in small quantities, citric acid can add 'freshness' and flavor to wines
- Residual sugar: the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet

INPUT VARIABLES (CONT.)

- Chlorides: the amount of salt in the wine
- Free sulfur dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; - it prevents microbial growth and the oxidation of wine
- Total sulfur dioxide: amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at - free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine
- Density: the density of water is close to that of water depending on the percent alcohol and sugar content

INPUT VARIABLES (CONT.)

- pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale
- Sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant
- Alcohol: the percent alcohol content of the wine

Output variable:

- Quality (score between 0 and 10)

WHAT I FOUND FROM THE ANALYSIS OF THIS DATASET?

- For the whole data set most of the people gave rating 5 and 6.
- Nobody gave rating 0, 1, 2, 9, 10. This might be because most of the people randomly choose the rating 5 and 6 And surprisingly no body rated 9 and 10 means the wine quality might not be good in reality.
- I first thought that acidity has predictive capability As quality increases with increase value of citric acid and decreases with increased value of volatile acidity.
- For residual sugar nobody gave rating 3 and 8 for the value greater than 6.8.
- May be only one people gave rating 4 for residual sugar value greater than 6.8.

WHAT I FOUND FROM THE ANALYSIS OF THIS DATASET?(CONT.)

- Most of the rating 5 falls below the alcohol value 11.
- Most of the rating 7 lies above the alcohol value 11.
- Rating 4, 6 are randomly distributed.
- The interesting fact is for the total sulfur dioxide value from 99 to 153 people gave rating 5 except of some outliers.
- People gave high rating for low value of pH.
- No people rated 8 for having chloride value greater than 0.121.
- For sulphate value greater than 0.94 people did not give rating 3.
- May be only one people gave rating 8. Most of the people gave rating 4.

WHAT I FOUND FROM THE ANALYSIS OF THIS DATASET?

- Density showed predictor for quality as it has trend. For higher value of density, quality is low and for lower value of density, quality is high.
- The linear model gave me seven final variables (volatile acidity, log10(chlorides), free sulfur dioxide, total sulfur dioxide, pH, log10(sulphates), alcohol) for prediction of quality of wine.
- There might be other variables (which are not present in our data) we need to consider for wine quality prediction.

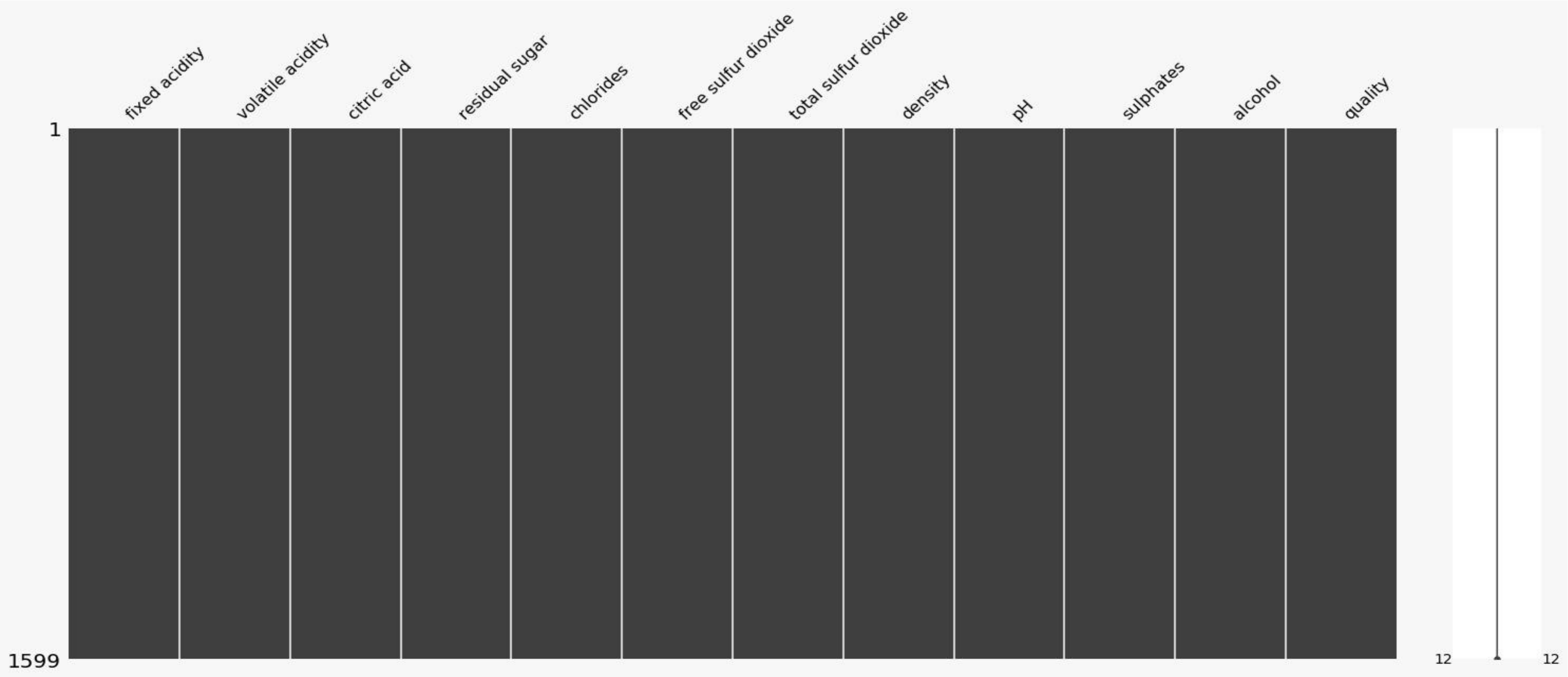
CHECKING THE NULL/MISSING VALUE IN THE DATASET

[CLEANING DATASET]

```
In [73]: import missingno as msno
         wine.isnull().sum()

Out[73]: 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 function count the colums which contain null value but here Data is pre processed and cleaned with dummy and null values.

BAR-CHART REPRESENTATION FOR SHOWING THE NULL VALUES



STASTICAL INFORMATION FOR DATASET

```
wine.describe().T
```

	count	mean	std	min	25%	50%	75%	max
fixed acidity	1599.0	8.319637	1.741096	4.60000	7.1000	7.90000	9.200000	15.90000
volatile acidity	1599.0	0.527821	0.179060	0.12000	0.3900	0.52000	0.640000	1.58000
citric acid	1599.0	0.270976	0.194801	0.00000	0.0900	0.26000	0.420000	1.00000
residual sugar	1599.0	2.538806	1.409928	0.90000	1.9000	2.20000	2.600000	15.50000
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	0.07900	0.090000	0.61100
free sulfur dioxide	1599.0	15.874922	10.460157	1.00000	7.0000	14.00000	21.000000	72.00000
total sulfur dioxide	1599.0	46.467792	32.895324	6.00000	22.0000	38.00000	62.000000	289.00000
density	1599.0	0.996747	0.001887	0.99007	0.9956	0.99675	0.997835	1.00369
pH	1599.0	3.311113	0.154386	2.74000	3.2100	3.31000	3.400000	4.01000
sulphates	1599.0	0.658149	0.169507	0.33000	0.5500	0.62000	0.730000	2.00000
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	10.20000	11.100000	14.90000
quality	1599.0	5.636023	0.807569	3.00000	5.0000	6.00000	6.000000	8.00000

TARGET VECTOR(OUTPUT COLUMN)

- Our Target Vector is QUALITY. Nobody gave rating 0, 1, 2, 9, 10. This might be because most of the people randomly choose the rating 5 and 6 And surprisingly no body rated 9 and 10 means the wine quality might not be good in reality.

CONVERTING NUMERICAL VALUE TO CATEGORICAL VALUE OF TARGET VARIABLES

```
conditions = [
    (wine['quality'] >= 7),
    (wine['quality'] <= 4)
]
rating = ['good', 'bad']
wine['rating'] = np.select(conditions, rating, default='average')
wine.rating.value_counts()
```

```
average    1319
good        217
bad          63
Name: rating, dtype: int64
```

- We divide the Wine Quality into 3 Categories:

  bad: 1-4

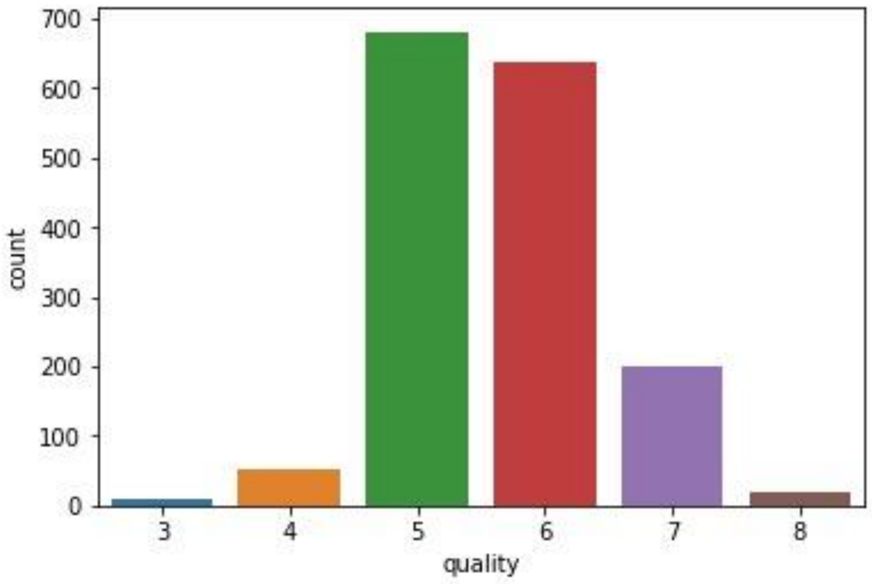
  average: 5-6

  good: 7-10

TARGET VECTOR (OUTPUT COLUMN)

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

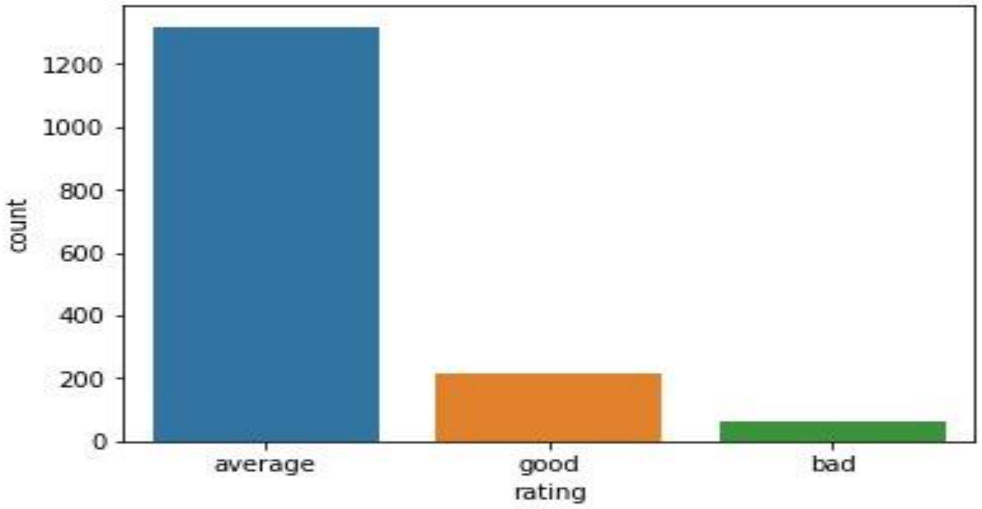
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f650dd16d8>
```



- This distribution shows the range for response variable (*quality*) is between 3 to 8.

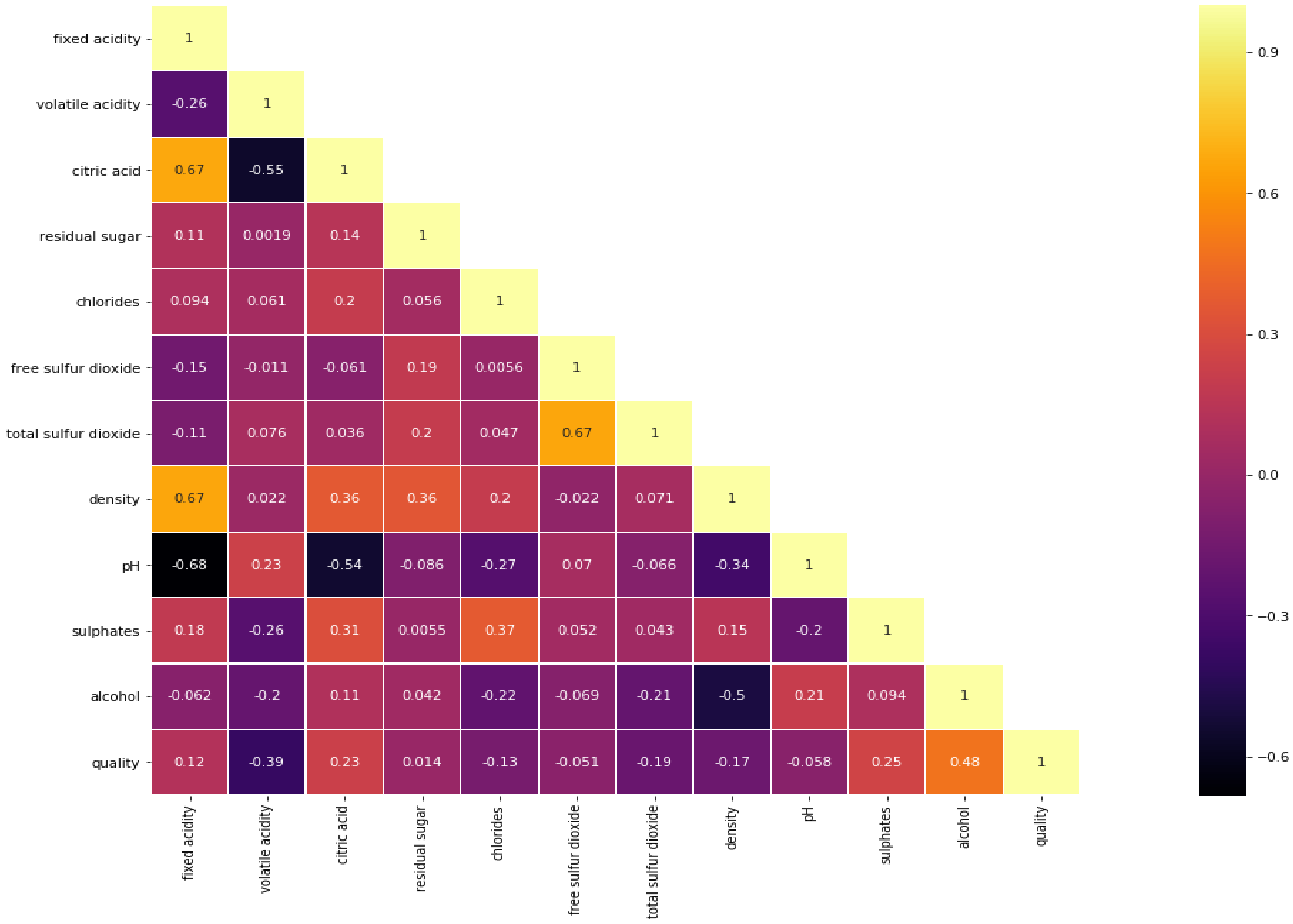
BAR-CHART REPRESENTATION

```
In [20]: sns.countplot(x='rating', data=wine)

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1f650ceca90>
```



CORRELATION BETWEEN FEATURES/VARIABLES

```
# correlation = wine.corr()
# plt.figure(figsize=(12, 5))
# #sns.heatmap(correlation, annot=True, linewidths=0, vmin=-1, cmap="RdBu_r")

correlation= wine.corr()
colormap = plt.cm.inferno
mask = np.array(correlation)
mask[np.tril_indices_from(mask)] = False
fig=plt.gcf()
fig.set_size_inches(30,12)
sns.heatmap(data=correlation ,mask=mask,square=True,annot=True,cbar=True,cmap=colormap, linecolor='White', linewidths=0.1)
```


fixed acidity 1
volatile acidity -0.26 1
citric acid 0.67 -0.55 1
residual sugar 0.11 0.0019 0.14 1
chlorides 0.094 0.061 0.2 0.056 1
free sulfur dioxide -0.15 -0.011 -0.061 0.19 0.0056 1
total sulfur dioxide -0.11 0.076 0.036 0.2 0.047 0.67 1
density 0.67 0.022 0.36 0.36 0.2 -0.022 0.071 1
pH -0.68 0.23 -0.54 -0.086 -0.27 0.07 -0.066 -0.34 1
sulphates 0.18 -0.26 0.31 0.0055 0.37 0.052 0.043 0.15 -0.2 1
alcohol -0.062 -0.2 0.11 0.042 -0.22 -0.069 -0.21 -0.5 0.21 0.094 1
quality 0.12 -0.39 0.23 0.014 -0.13 -0.051 -0.19 -0.17 -0.058 0.25 0.48 1
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality
0.9
0.6
0.3
0.0
−0.3
−0.6

CORRELATION BETWEEN FEATURES/VARIABLES

- Most affecting Factors are:
  - Alcohol
  - Volatile acidity
  - Sulphates
  - Critic Acid
- Least affecting Factors:
  - Residual sugar
  - Free Sulphur Dioxide
  - Ph

- Positive Correlated Factors:
  - Alcohol
  - Sulphates
  - Citric acid
  - Fixed acidity

  *(all the factors are in decreasing order **Most to least**)*
- Negative Correlated Factors:
  - volatile acidity
  - total sulfur dioxide
  - density
  - chlorides

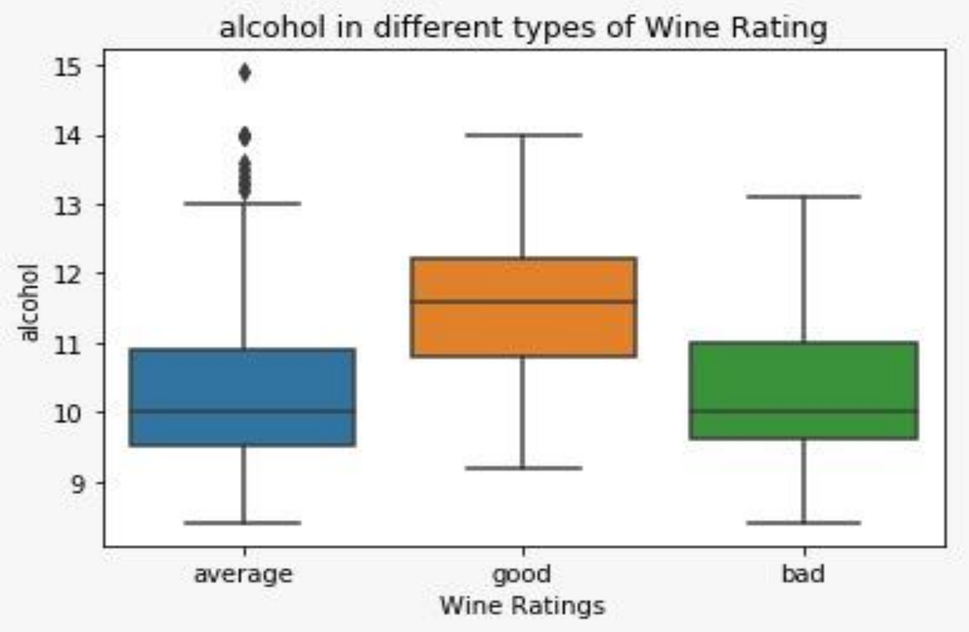
  *(all the factors are in decreasing order **Most to least**)*

ANALYSIS FOR HOW DIFFERENT FACTORS AFFECT WINE-QUALITY

ANALYSIS OF ALCOHOL PERCENTAGE VS WINE QUALITY

```
bx = sns.boxplot(x="rating", y='alcohol', data = wine)
bx.set(xlabel='Wine Ratings', ylabel='alcohol', title='alcohol in different types of Wine Rating')
```

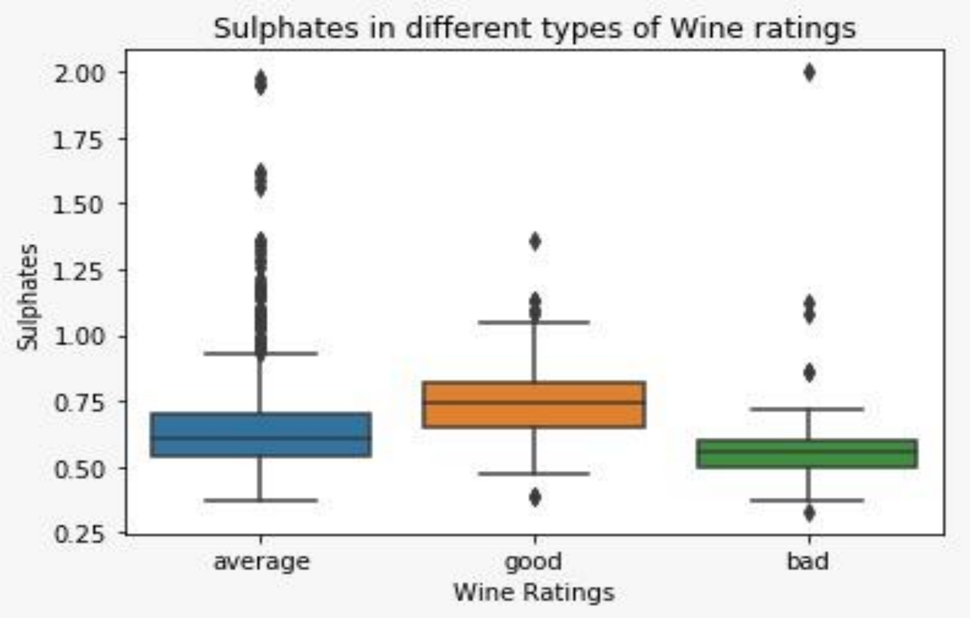
```
[Text(0,0.5,'alcohol'),
 Text(0.5,0,'Wine Ratings'),
 Text(0.5,1,'alcohol in different types of Wine Rating')]
```



ANALYSIS OF SULPHATES VS WINE RATINGS:

```
bx = sns.boxplot(x="rating", y='sulphates', data = wine)
bx.set(xlabel='Wine Ratings', ylabel='Sulphates', title='Sulphates in different types of Wine ratings')
```

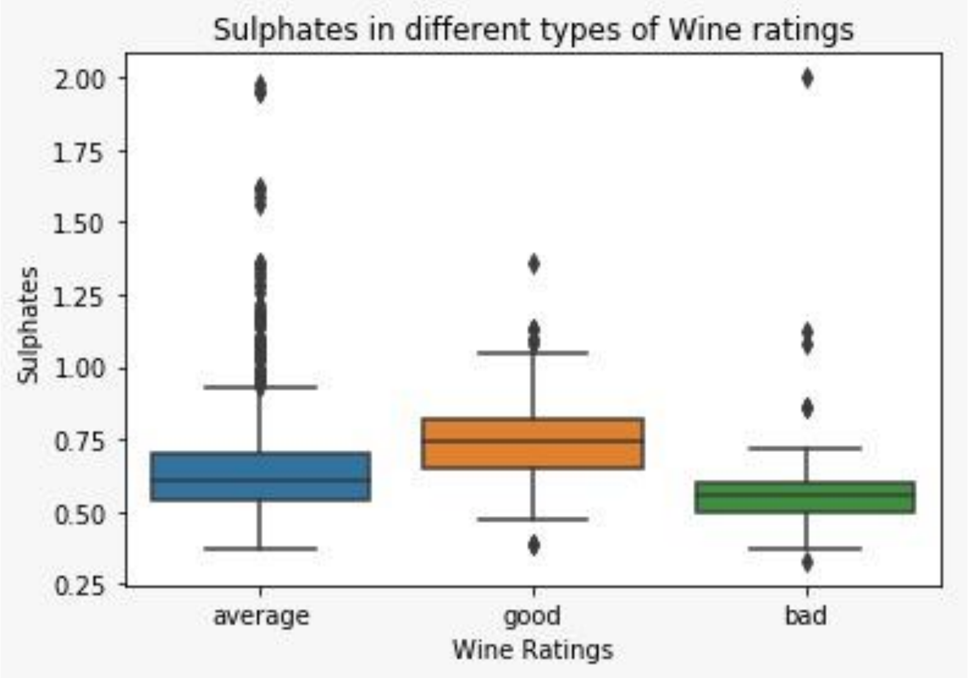
```
[Text(0,0.5,'Sulphates'),
 Text(0.5,0,'Wine Ratings'),
 Text(0.5,1,'Sulphates in different types of Wine ratings')]
```



ANALYSIS OF CITRIC ACID VS WINE RATINGS

```
bx = sns.boxplot(x="rating", y='sulphates', data = wine)
bx.set(xlabel='Wine Ratings', ylabel='Sulphates', title='Sulphates in different types of Wine ratings')
```

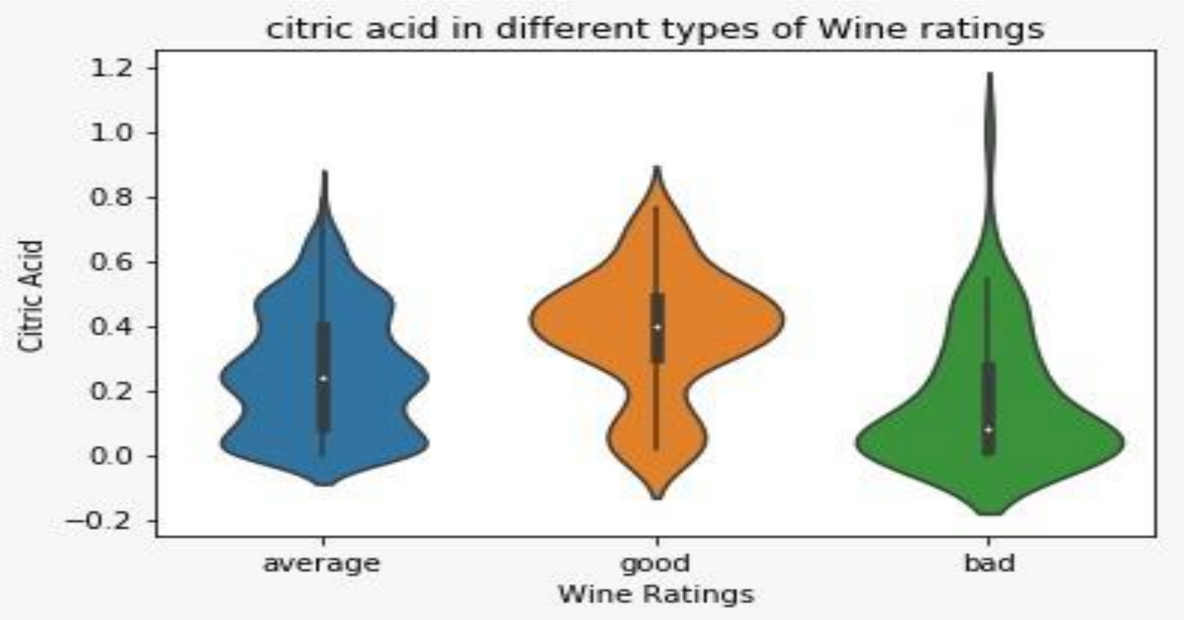
```
[Text(0,0.5,'Sulphates'),
 Text(0.5,0,'Wine Ratings'),
 Text(0.5,1,'Sulphates in different types of Wine ratings')]
```



ANALYSIS OF CITRIC ACID VS WINE RATINGS

```
bx = sns.violinplot(x="rating", y='citric acid', data = wine)
bx.set(xlabel='Wine Ratings', ylabel='Citric Acid', title='citric acid in different types of Wine ratings')
```

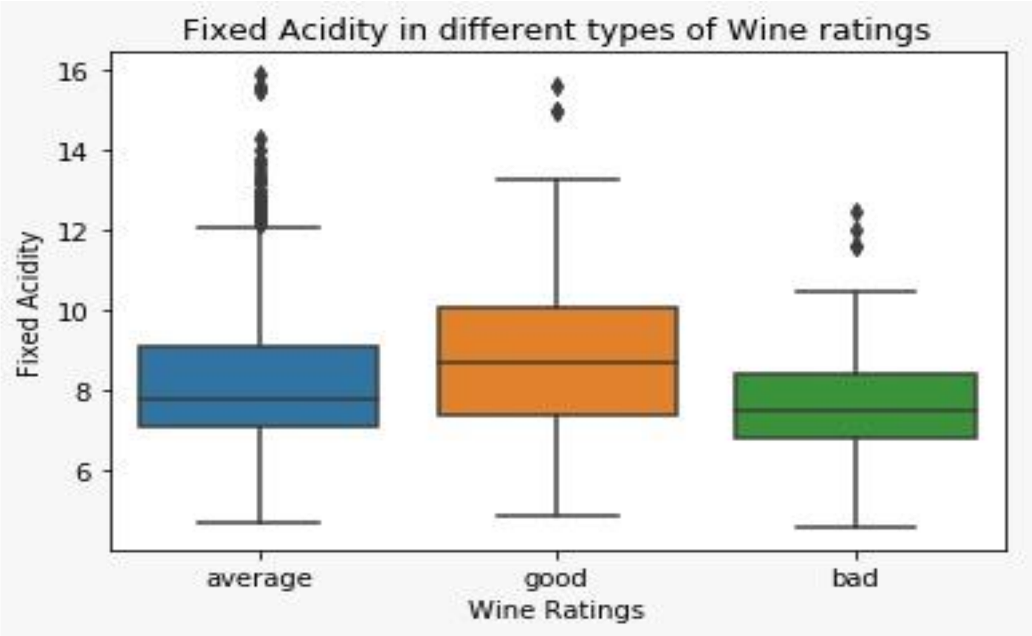
```
[Text(0,0.5,'Citric Acid'),
 Text(0.5,0,'Wine Ratings'),
 Text(0.5,1,'citric acid in different types of Wine ratings')]
```



ANALYSIS OF FIXED ACIDITY & WINE RATINGS

```
bx = sns.boxplot(x="rating", y='fixed acidity', data = wine)
bx.set(xlabel='Wine Ratings', ylabel='Fixed Acidity', title='Fixed Acidity in different types of Wine ratings')
```

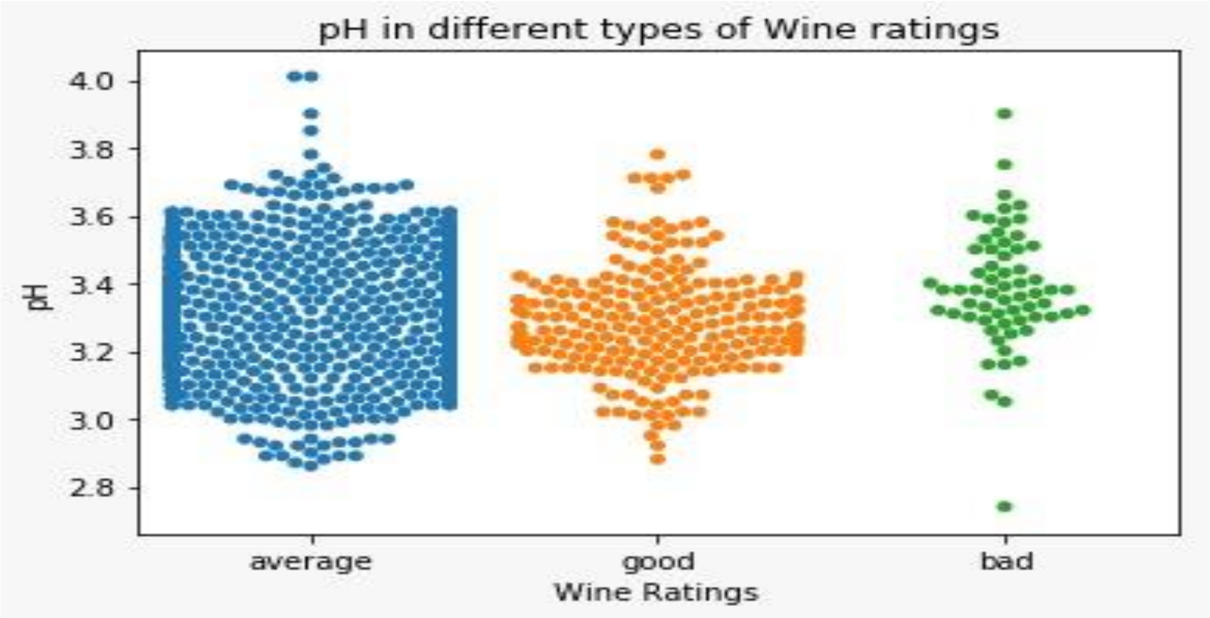
```
[Text(0,0.5,'Fixed Acidity'),
 Text(0.5,0,'Wine Ratings'),
 Text(0.5,1,'Fixed Acidity in different types of Wine ratings')]
```



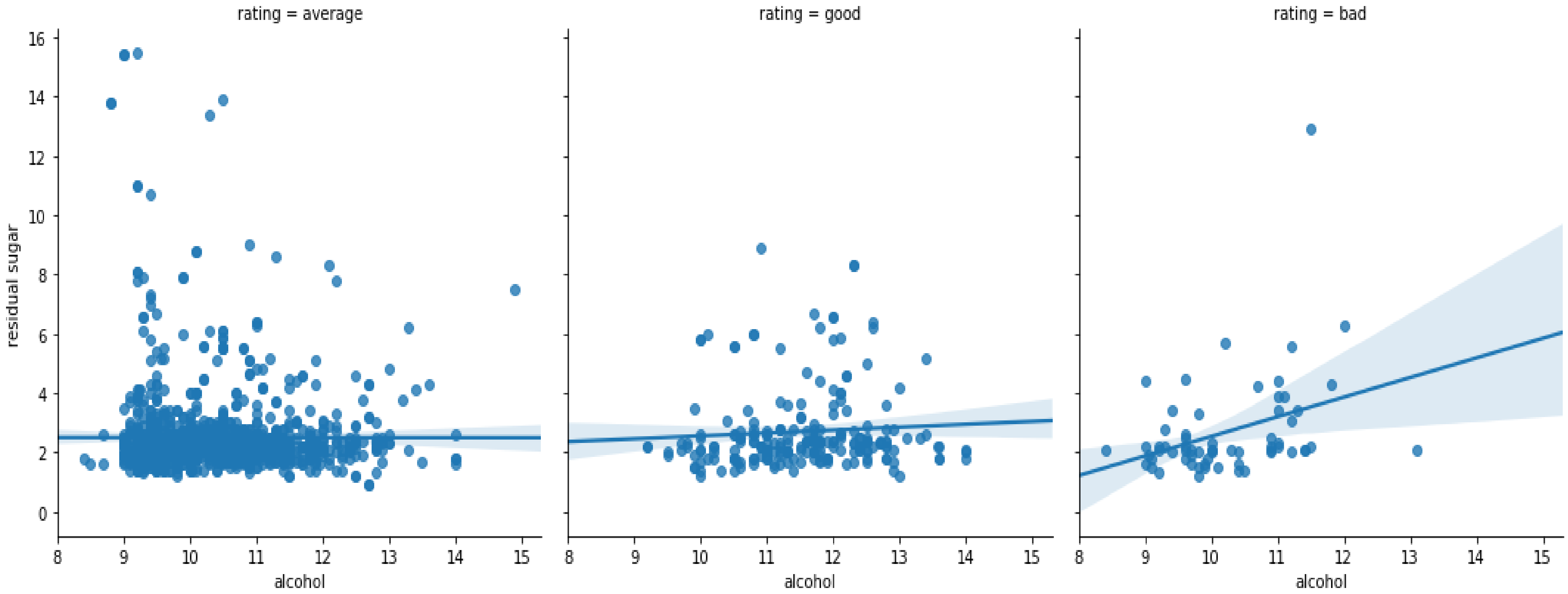
ANALYSIS OF PH VS WINE RATINGS

```
bx = sns.swarmplot(x="rating", y="pH", data = wine);
bx.set(xlabel='Wine Ratings', ylabel='pH', title='pH in different types of Wine ratings')
```

```
[Text(0,0.5,'pH'),
 Text(0.5,0,'Wine Ratings'),
 Text(0.5,1,'pH in different types of Wine ratings')]
```



LINEAR REGRESSION



LINEAR REGRESSION

- The linear regression plots above for different wine quality ratings (bad, average & good) shows the regression between alcohol and residual sugar content of the red wine.
- We can observe from the trendline that, for good and average wine types the residual sugar content remains almost constant irrespective of alcohol content value. Whereas for bad quality wine, the residual sugar content increases gradually with the increase in alcohol content.
- This analysis can help in manufacturing the good quality wine with continuous monitoring and controlling the alcohol and residual sugar content of the red wine.

APPLY DIFFERENT CLASSIFIER ON DATASET

- **For that we divide wine quality in label vector 1(good) and 0(bad)**

  1 (good) quality >= 6.5

  0 (bad) quality < 6.5

```
def label_vector_design(x):
    if x >= 6.5:
        return 1
    elif x < 6.5 :
        return 0
wine['label'] = wine['quality'].apply(label_vector_design)
```

CLASSIFICATION TECHNIQUES FOR PREDICTING ACCURACY

K Nearest Neighbors

Logistic Regression

Random Forest Classifier

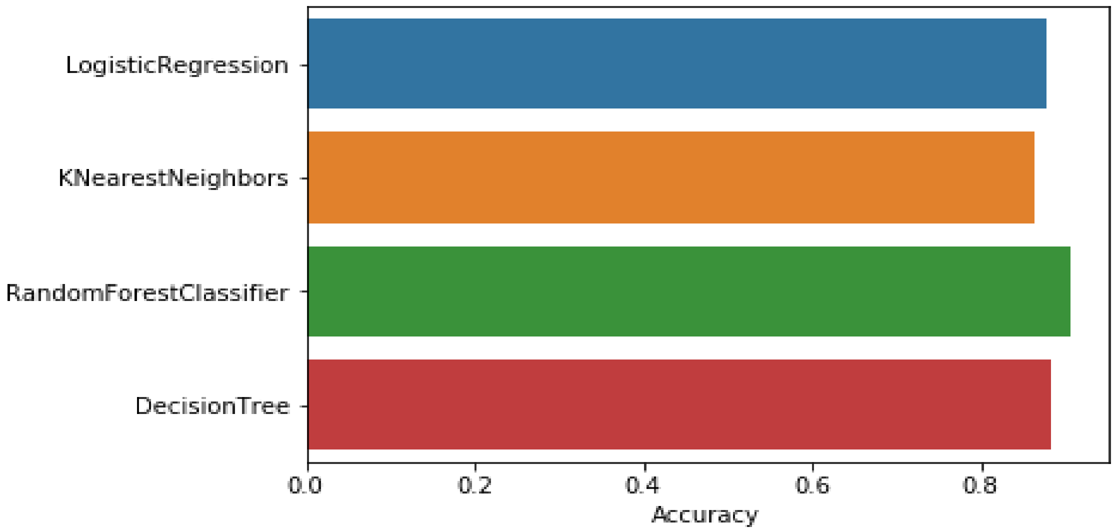
Decision Tree Classifier

ACCURACY (WITHOUT NORMALIZATION)

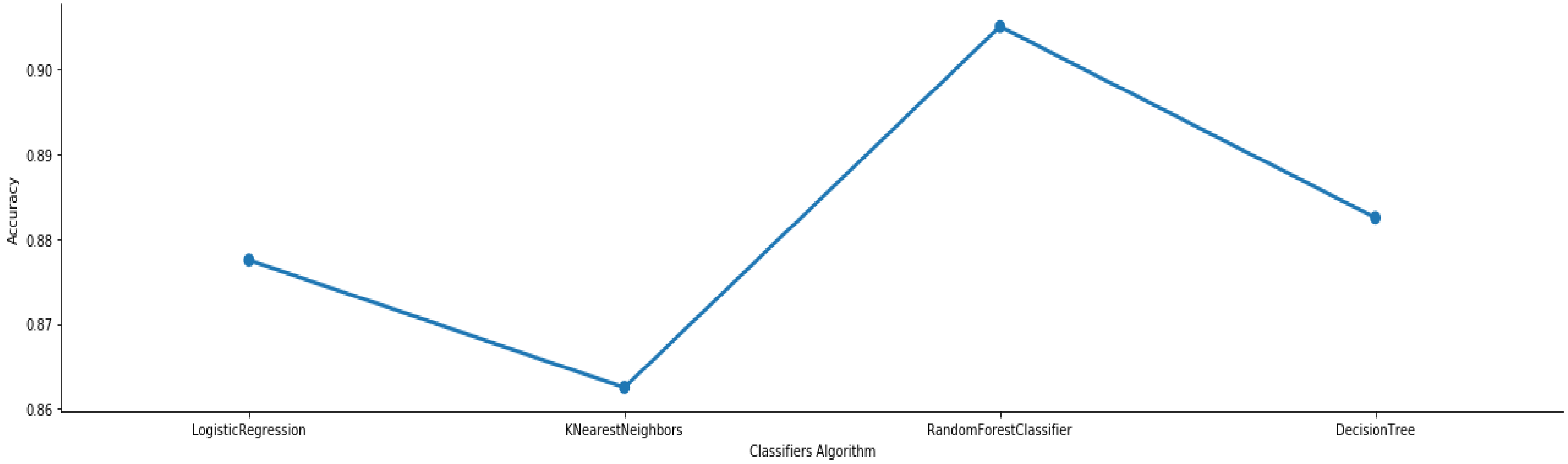
```
X_train,X_test,y_train,y_test=train_test_split(X, y, test_size=0.25, random_state=42)

#Accuracy
Accuracy = [ ]

classifiers=[LogisticRegression(),KNeighborsClassifier(),
             RandomForestClassifier(random_state=41),DecisionTreeClassifier(random_state=42),]
classifiers_names=['LogisticRegression','KNearestNeighbors','RandomForestClassifier','DecisionTree']

result={}

for classifier  in range(len(classifiers)):
    c=classifiers[classifier]
    c.fit(X_train,y_train)
    y_predict=c.predict(X_test)
    Accuracy.append(accuracy_score(y_predict,y_test))

result={'Classifiers Algorithm':classifiers_names,'Accuracy':Accuracy}

Accuracy_frame=pd.DataFrame(result)
Accuracy_frame
```

ACCURACY (WITHOUT NORMALIZATION)

	Classifiers Algorithm	Accuracy
0	LogisticRegression	0.8775
1	KNearestNeighbors	0.8625
2	RandomForestClassifier	0.9050
3	DecisionTree	0.8825



ACCURACY (WITHOUT NORMALIZATION)



ACCURACY (AFTER NORMALIZATION)

Normalization of Feature Matrix

```
In [113]: X_Scaled = preprocessing.scale(X, axis=0, with_mean=True, with_std=True, copy=True)
          X_Scaled

Out[113]: array([[-0.52835961,  0.96187667, -1.39147228, ...,  1.28864292,
                  -0.57920652, -0.96024611],
                 [-0.29854743,  1.96744245, -1.39147228, ..., -0.7199333 ,
                   0.1289504 , -0.58477711],
                 [-0.29854743,  1.29706527, -1.18607043, ..., -0.33117661,
                  -0.04808883, -0.58477711],
                 ...,
                 [-1.1603431 , -0.09955388, -0.72391627, ...,  0.70550789,
                   0.54204194,  0.54162988],
                 [-1.39015528,  0.65462046, -0.77526673, ...,  1.6773996 ,
                   0.30598963, -0.20930812],
                 [-1.33270223, -1.21684919,  1.02199944, ...,  0.51112954,
                   0.01092425,  0.54162988]])
```

ACCURACY (AFTER NORMALIZATION)

```
X_train,X_test,y_train,y_test=train_test_split(X_Scaled, y, test_size=0.25, random_state=42)

#Accuracy
Accuracy_Scaled = [ ]

classifiers=[LogisticRegression(),KNeighborsClassifier(),
                RandomForestClassifier(random_state=41),DecisionTreeClassifier(random_state=42),]
classifiers_names=['LogisticRegression','KNearestNeighbors','RandomForestClassifier','DecisionTree']

result_normalized={}

for classifier  in range(len(classifiers)):
    c=classifiers[classifier]
    c.fit(X_train,y_train)
    y_predict=c.predict(X_test)
    Accuracy_Scaled.append(accuracy_score(y_predict,y_test))

result_normalized={'Classifiers Algorithm':classifiers_names,'Accuracy_Scaled':Accuracy_Scaled}

Accuracy_frame_Normalized=pd.DataFrame(result_normalized)
Accuracy_frame_Normalized
```

ACCURACY (AFTER NORMALIZATION)

	Classifiers Algorithm	Accuracy_Scaled
0	LogisticRegression	0.8775
1	KNearestNeighbors	0.8950
2	RandomForestClassifier	0.9050
3	DecisionTree	0.8825

We found that after normalization Accuracy remain same so, Data is Already Normalized

RESPONSIBILITY OF EACH TEAM MEMBERS


Thank You