DATA MINING

Mini Project

Members:

Harshil Shah 1213348082

Jay Bhanushali 1213436781

Ronak Vikam 1213203327

Viren Bhanushali 1213232850

DATA DESCRIPTION:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed acidity           1599 non-null float64
volatile acidity        1599 non-null float64
citric acid             1599 non-null float64
residual sugar          1599 non-null float64
chlorides               1599 non-null float64
free sulfur dioxide     1599 non-null float64
total sulfur dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

These are the attributes of data and our goal is to predict the quality of the Wine. There are 12 categorical variables and the target variable is to be predicted from 0 to 10.

Q.1)

The Accuracy of the model is generated using three ways, Analysis on Training data, analysis on Test data and Analysis on validation data. Based on the code we have generated; the values of accuracy are as follows:

```
In [73]: print("Accuracy for Complete Traing Data = " , accuracy_score(actuals, foldhats))

Accuracy for Complete Traing Data =  0.564727954972
```

```
In [74]: print("Accuracy for Train_Test split = " , accuracy_score(actuals1, foldhats1))

Accuracy for Train_Test split =  0.55
```

```
In [67]: print ("Accuracy for Kfold Method")
         print ("CVerror = ", metrics.accuracy_score(actuals2,hats2))

Accuracy for Kfold Method
CVerror =  0.547217010632
```

Based on the result, we find that Accuracy for Training Data is higher than the accuracy for Test Data. This also validates our Theoretical assumption that Accuracy for Training data should be greater than the accuracy for test data.

Depending on the Data we have taken, it seems our model can predict approximately 56% of the data. The Naïve Bayes model may not be exactly appropriate for this data, but if we apply some other algorithms we can predict the model more efficiently.

Q.1.b) Provide the code and a confusion matrix, summary statistics, and a ROC curve calculated from the **crossvalidation only**.

```
In [66]: print(metrics.classification_report(Yset, hats2))
```

	precision	recall	f1-score	support
3	0.00	0.00	0.00	10
4	0.05	0.06	0.05	53
5	0.43	0.43	0.43	681
6	0.42	0.38	0.40	638
7	0.15	0.18	0.16	199
8	0.00	0.00	0.00	18
avg / total	0.37	0.36	0.36	1599

Figure 1: Summary Statistics

```
In [46]: cm = ConfusionMatrix(actuals2,hats2)
         print (cm)
```

Predicted	3.0	4.0	5.0	6.0	7.0	8.0	__all__
Actual							
3.0	1	2	6	1	0	0	10
4.0	3	4	28	15	2	1	53
5.0	3	34	449	170	23	2	681
6.0	1	20	188	319	97	13	638
7.0	0	2	13	78	101	5	199
8.0	0	0	0	4	13	1	18
__all__	8	62	684	587	236	22	1599

Figure 2: Confusion Matrix

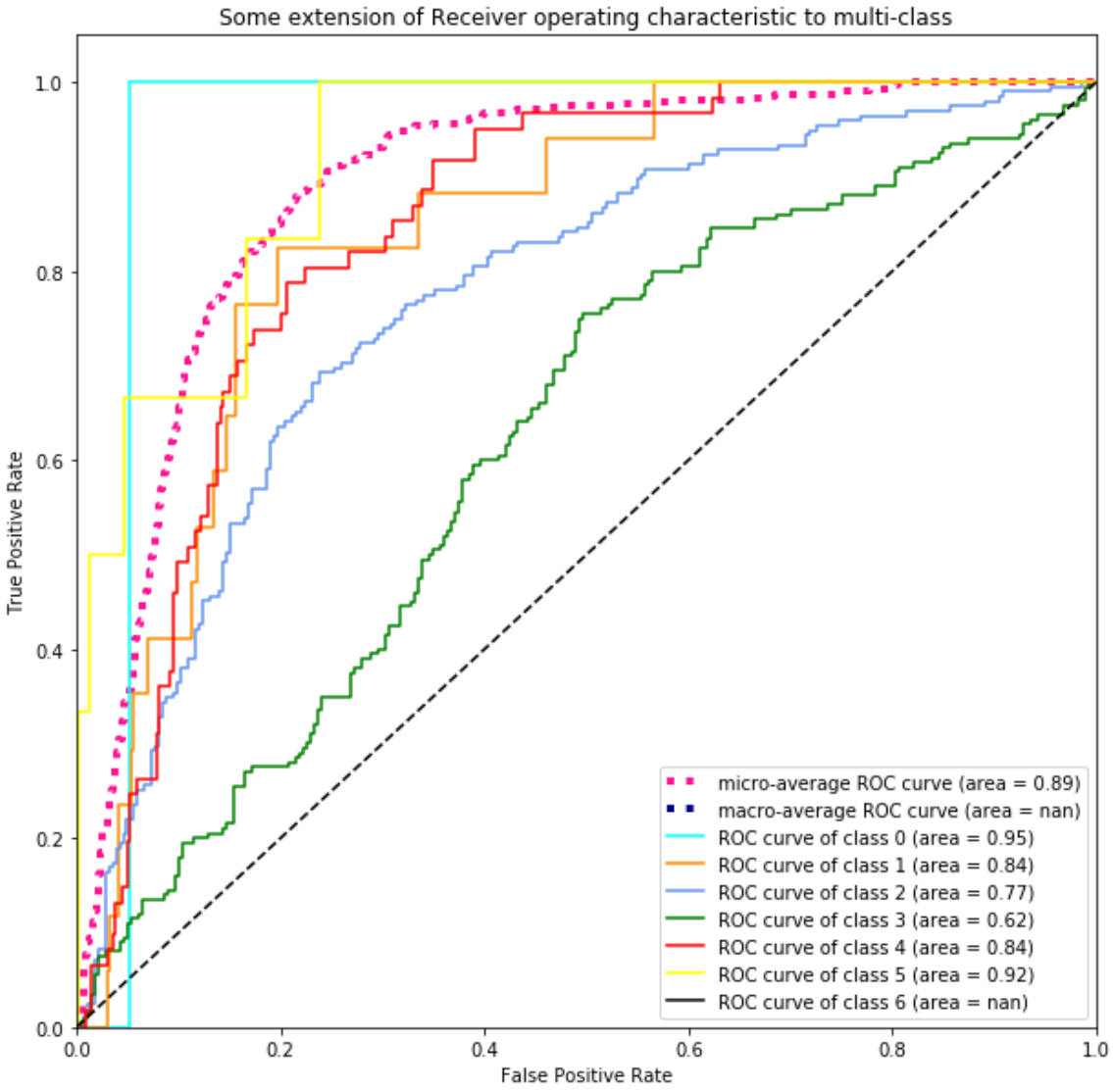

Figure 3: ROC Curve

We have generated the ROC curve for a particular class using the binaries. We have 6 different classes and each curve represents the area under the curve for a class.

Q.2.a) The Box Plot has been displayed below for X=5 and X=15. The Box Plot is distinguished by the Width of the Plot. The Box Plot with smallest Width is defined by Degree of Polynomial 1 similarly the Width greater than that is Degree of polynomial 2 and Largest width is for Degree of Polynomial 5. The Red Line is the line for mean for that respective Degree of polynomial.

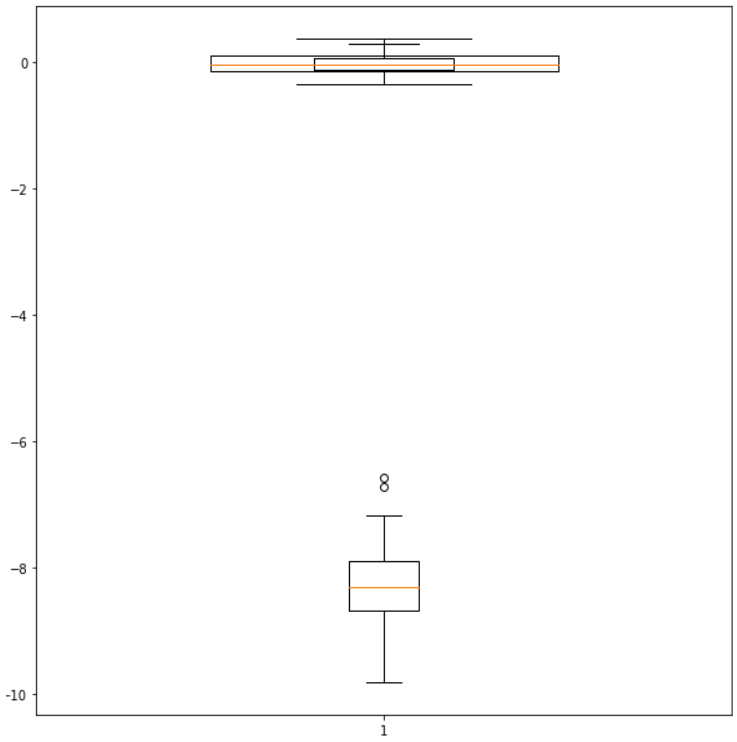

Figure 4: Box Plot for X=5

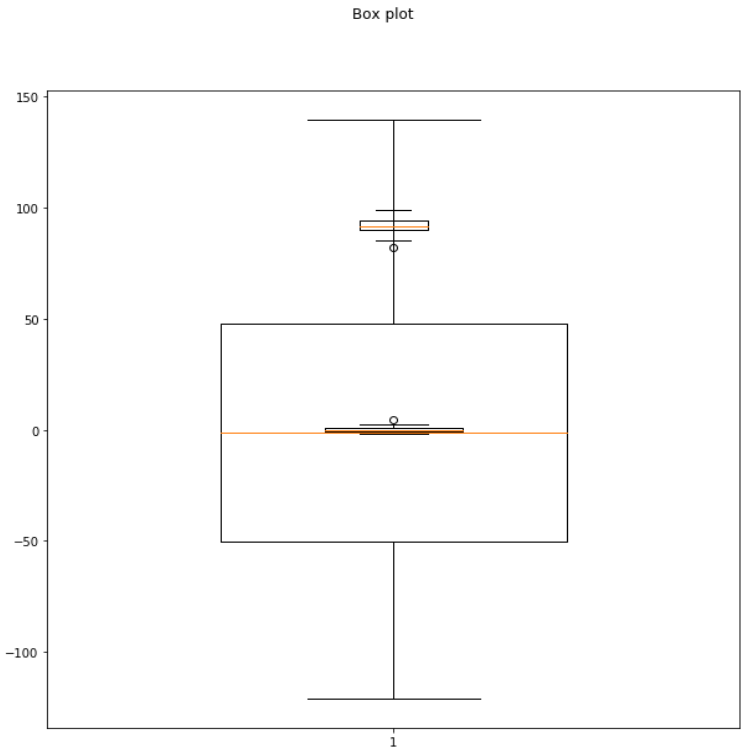

Figure 5:Box Plot for X=15

Q.2.b) Provide estimates of variance and bias for each model in a table for $x = 5$ and $x = 15$.

Bias of Y for x =5 for polynomial degree 1 = -8.279919724719
Bias of Y for x =5 for polynomial degree 2 = -0.01693649687504717
Bias of Y for x =5 for polynomial degree 5 = -0.010845469934650964
Bias of Y for x =15 for polynomial degree 1 = 91.80547447389334
Bias of Y for x =15 for polynomial degree 2 = 0.05494875769429086
Bias of Y for x =15 for polynomial degree 5 = 3.234264276601124

Variance of Y for x =5 for polynomial degree 1 = 0.46871040287200505
Variance of Y for x =5 for polynomial degree 2 = 0.019058222504026966
Variance of Y for x =5 for polynomial degree 5 = 0.029414694683230182
Variance of Y for x =15 for polynomial degree 1 = 12.789694967318253
Variance of Y for x =15 for polynomial degree 2 = 1.4133583212357796
Variance of Y for x =15 for polynomial degree 5 = 4159.042427899485

Q.2.c)

The equation for the model is given by $y = 5 - 2x + x_2 + e$

The Model with greatest Bias is X=15 for degree of Polynomial 1 = 91.805

The Model with greatest Variance is X=15 for degree of Polynomial 5 = 4159.042

Yes, we had expected these results because as the Number of variables increases, the complexity of model increases and therefore Bias Decreases and Variance Increases as we add more polynomial Degrees to the Model.

The Results vary as the degree of polynomial differs. As we can see, X=5 gives a better predicted of the model because we are taking the uniform distribution and it lies within the range 0 to 10. On the contrary, The model for X=15 is out of data thus we need to do Data Extrapolation of the data.