Assignment - 1 Big Data Analytics (CSOE17)
REPORT

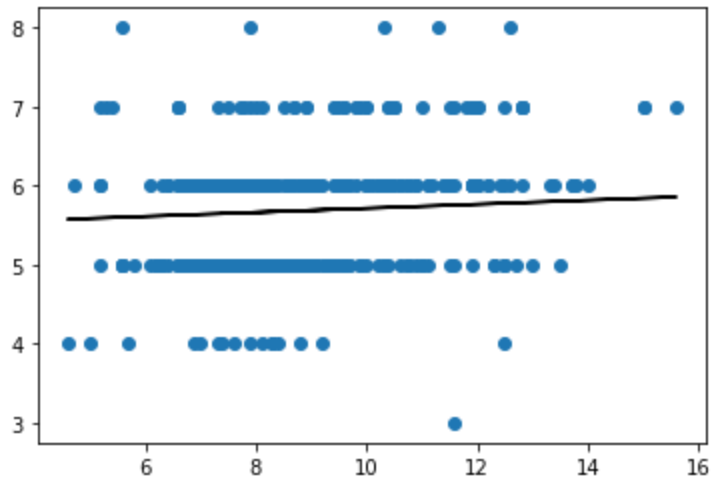
Thrishik Senthilkumar

110118092

TASK 1

The Quality attributes of a wine given in a dataset, are estimated through linear regression . The training dataset are used to train the linear regression model and Tested with testing data, the output which we get is the predicted data. I have formed a Graphical representation of a line fitted over the testing data considering for a single feature (Fixed acidity).

```
Intercept: 5.457538106303752
Slope: [0.02557288]
**********************************************************************
fixed acidity x_test vs y_pred
```



Considering multi variable linear regression, the sum of squared errors are calculated for the test dataset. The output is attached below. The line is found to have a slope of 0.0255 and its Y-intercept is around 5.458.

```
Mean Absolute Error: 0.5224781243375761
Mean Squared Error: 0.4463021333444595
Root Mean Squared Error: 0.6680584804824047
```

TASK 2

In the given dataset, we treat quality to be the class attribute and by taking those tuples with quality value greater than 7 to be “good” and others “bad”, we now handle it as a two-class problem.

```
X_TRAINING DATA:
```

Out[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
0	10.6	0.28	0.39	15.5	0.069	6.0	23.0	1.0026	3.12	0.66	9.2
1	9.4	0.30	0.56	2.8	0.080	6.0	17.0	0.9964	3.15	0.92	11.7
2	10.6	0.36	0.59	2.2	0.152	6.0	18.0	0.9986	3.04	1.05	9.4
3	10.6	0.36	0.60	2.2	0.152	7.0	18.0	0.9986	3.04	1.06	9.4
4	10.6	0.44	0.68	4.1	0.114	6.0	24.0	0.9970	3.06	0.66	13.4

In [3]:
```
print("*****************************************************************************************************")
# CLASSIFING quality 0-bad AND 1-good and forming Y_TRAINING DATASET
y_train = train_wine["quality"].apply(lambda  q:0 if q<7 else 1)

print("Y_TRAINING DATA:")
y_train.head(5)
```

```
*****************************************************************************************************
Y_TRAINING DATA:
```

Out[3]:
```
0    0
1    1
2    0
3    0
4    0
Name: quality, dtype: int64
```

As seen here, the quality attribute of training data has been converted to a 2-class problem with 0 and 1 being the 2 classes. Similarly we classify the quality attribute for testing data as well.

We apply the 4 types of classifier algorithms and observe the outcome.

We must now create classification algorithms and train them with training data and apply them on the testing data to predict the classes of each tuple. By comparing these predicted values and actual outputs, we can evaluate the effectiveness of our algorithms.

(i)LINEAR REGRESSION CLASSIFIER:

We first apply the Linear Regression based classifier on the test data, and then classify the prediction data based on the mean.

```
Y_PREDICTION DATA:(first 10 values)
array([-0.12480267, -0.04486957, -0.02591986,  0.17223449, -0.12480267,
       -0.12215259, -0.07652599,  0.01531073, -0.01059442,  0.22524849])

mean:  0.10580178551233062

Y_PREDICTION DATA after classification:(first 10 values)
[0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
```

(ii)LOGISTIC REGRESSION CLASSIFIER:

```
Y_PREDICTION:(first 10 values)
[0 0 0 0 0 0 0 0 0 0]
```

(iii) SVM CLASSIFIER:

```
Y_PREDICTION:(first 10 values)
[0 0 0 0 0 0 0 0 0 0]
```

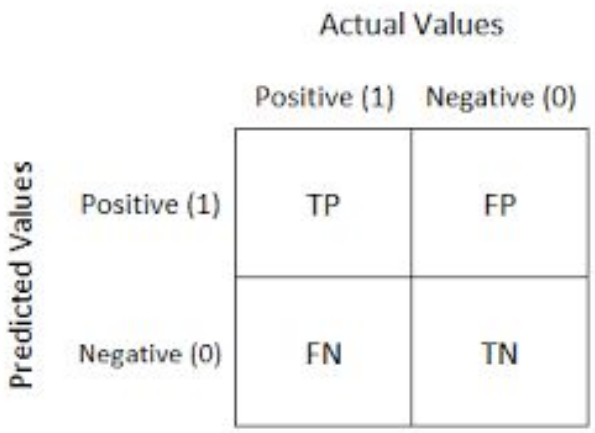
(iv) NAÏVE BAYES CLASSIFIER:

```
BernoulliNB(binarize=True)
Y_PREDICTION:(first 10 values)
[0 0 0 0 0 0 0 0 0 0]
```

These are the predicted outcomes of the first 10 tuples generated by the 4 different classifier algorithms.

TASK 3

We must compare these predicted outcomes to the actual outcomes and determine the performance measures of the 4 algorithms, namely the accuracy, precision, recall, f measure, sensitivity and specificity of each algorithm, derived from the confusion matrix.



We derive the mentioned parameters from the confusion matrix given.
PRECISION(P) = TP/(TP+FP)
RECALL(R) = TP/ (TP+FN)
F-MEASURE = 2*P*R/(P+R)
ACCURACY = (TP+TN)/(TP+TN+FP+FN)
SENSITIVITY = TP/(TP+FN)
SPECIFICITY = TN/(TN+FP)
We look at these measures to determine how effective the classifier is.

PERFORMANCE MEASURES

(i) PERFORMANCE OF LINEAR REGRESSION CLASSIFIER:

```
Confusion Matrix :
 [[280 146]
 [  3  51]]
Accuracy :  0.6895833333333333
Precision :  0.9893992932862191
Recall :  0.8459214501510574
```

```
f measure:  0.9120521172638437
Sensitivity :  0.6572769953051644

Specificity :  0.9444444444444444
```

(ii) <u>PERFORMANCE OF LOGISTIC REGRESSION CLASSIFIER:</u>

```
Confusion Matrix :
 [[414  12]
 [ 43  11]]
Accuracy :  0.8854166666666666
Precision :  0.9059080962800875
Recall :  0.9741176470588235
f measure:  0.9387755102040816
Sensitivity :  0.971830985915493
Specificity :  0.2037037037037037
```

(iii) <u>PERFORMANCE OF SVM CLASSIFIER:</u>

```
Confusion Matrix :
 [[280 146]
 [  3  51]]
Accuracy :  0.6895833333333333
Precision :  0.9893992932862191
Recall :  0.8459214501510574
f measure:  0.9120521172638437
Sensitivity :  0.6572769953051644
Specificity :  0.9444444444444444
```

(iv) <u>PERFORMANCE OF NAÏVE BAYES CLASSIFIER:</u>

```
Confusion Matrix :
 [[426   0]
 [ 54   0]]
Accuracy :  0.8875
Precision :  0.8875
Recall :  1.0
f measure:  0.9403973509933775
Sensitivity :  1.0
Specificity :  0.0
```

These determine the performance of each classifier in predicting the outcome of the given dataset tested for quality.

TASK 4

PCA is dimensionality-reduction method used to reduce dimension of larger dataset, by converting larger dataset into smaller dataset but that still contains almost all the information.
We need to standardize the data, then find a covariance matrix to compute eigenvectors and eigenvalues of the covariance matrix to identify principal components. And we implement pca by using library sklearn which has inbuilt functions to apply PCA to the dataset.

PCA for 7 attributes
For Linear Regression Classifier
Accuracy is : 0.45416666666666666
Precision is : 0.9226804123711341
Recall is : 0.8211009174311926
F-measure is : 0.8689320388349516
sensitivity is : 0.42018779342723006
specificity is : 0.7222222222222222
For Logistic Regression Classifier
Accuracy is : 0.875
Precision is : 0.8877118644067796
Recall is : 0.9976190476190476
F-measure is : 0.9394618834080718
sensitivity is : 0.9835680751173709
specificity is : 0.018518518518518517
For SVM Classifier
Accuracy is : 0.88125
Precision is : 0.888421052631579
Recall is : 0.9976359338061466
F-measure is : 0.9398663697104677
sensitivity is : 0.9906103286384976
specificity is : 0.018518518518518517
For Naive Bayesian Classifier
Accuracy is : 0.8854166666666666
Precision is : 0.8888888888888888
Recall is : 0.9976470588235294
F-measure is : 0.9401330376940134
sensitivity is : 0.9953051643192489
specificity is : 0.018518518518518517

PCA for 4 attributes
For Linear Regression Classifier
Accuracy is : 0.4354166666666667
Precision is : 0.9015544041450777
Recall is : 0.8325358851674641
F-measure is : 0.8656716417910447
sensitivity is : 0.4084507042253521
specificity is : 0.6481481481481481
For Logistic Regression Classifier
Accuracy is : 0.8875
Precision is : 0.8875
Recall is : 1.0
F-measure is : 0.9403973509933775
sensitivity is : 1.0
specificity is : 0.0
For SVM Classifier
Accuracy is : 0.8875
Precision is : 0.8875
Recall is : 1.0
F-measure is : 0.9403973509933775
sensitivity is : 1.0
specificity is : 0.0
For Naive Bayesian Classifier
Accuracy is : 0.8875
Precision is : 0.8875
Recall is : 1.0
F-measure is : 0.9403973509933775
sensitivity is : 1.0
specificity is : 0.0

And as we observe,

Accuracy has reduced compared with dataset passed to modal without PCA, and also PCA with 7 components has more accuracy compared with PCA of 4 components. But not very large differences are observed.
