Project report

Anand Karandikar (askarand), Rohan Ingale(ringale)

1. **Dataset Description:**
   a. Wine Quality dataset:
   It consists of two sub data sets for red and white wine. We have used this data set for a classification problem to predict whether the wine quality is good or bad.
   The data set consists of 1599 red wine records and 4898 white wine records.
   Number of attributes = 11 + class label
   The output attribute in the actual data set had quality which rated the wine between 0 and 10. We converted it into a binary classification problem by classifying all the wines with quality below 6 as “bad” and above 6 as “good”
   We have divided the data sets into 9:1 ratio for training and testing by random sampling but at the same time ensuring that the class bias proportion remains same in train and test data set. The following is analysis on white wine data:
   Number of class 0 records in train data: 1476
   Number of class 1 records in train data: 2932
   Number of class 0 records in train data: 164
   Number of class 1 records in train data: 326

   b. Bank dataset:
   This data set is related to bank marketing campaigns. The goal of classification is to predict whether a client will subscribe to the bank term deposit scheme.
   We have used the smaller dataset available which consists of 4521 records.
   The class label contained values “yes” and “no” which we have mapped to “1” and “0” respectively. Also several attributes had string values (discrete) which we mapped to numerical discrete values for easy computation using Logistic regression and knn. The dataset did not have any missing values.
   The dataset is highly class biased with
   Number of class 0 records = 4000
   Number of class 1 records = 521
   We have divided the data sets into 9:1 ratio for training and testing by random sampling but at the same time ensuring that the class bias proportion remains same in train and test data set.
   Number of class 0 records in train data: 400
   Number of class 1 records in train data: 469
   Number of class 0 records in test data: 3600
   Number of class 1 records in test data: 52

   Since we have implemented knn algorithm, we have normalized all the attributes in both the datasets.

2. Results:

We have implemented knn and logistic regression algorithms to solve the classification problem on the above described data sets. The algorithms are implemented in python and experiments were performed for different settings.

We have used Weka to perform feature evaluation. The feature evaluation was performed using following specifications:
Evaluator: weka.attributeSelection.GainRatioAttributeEval
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
The following was the attribute index order obtained, sorted by their rank:
Best predictors for wine: 10,9,1,6,7,2,4,0,3,8,5
Best predictors for bank: 2,0,12,4,5,13,12,1,11,6,5,7,8,9

We used this result to perform feature selection for the algorithms that we implemented to experiment performance for varying number of features selected based on their ranking. The feature selection plays an important role in knn algorithm since the algorithm cannot tell anything regarding the importance of each feature.

The average accuracy, true positive rate, false positive rate and confusion matrix for the two algorithms are as follows:

a. K-nearest neighbor:

For K-nearest neighbor, the average of results over values of ‘k’ between 1 and 20 is taken. This is done for 3 different distance metrics i.e. Euclidean distance, Manhattan distance, Complete link.

For red wine dataset:

Euclidean:

```
Average values for the confusion matrix:
                    Predicted
             |Class (1) |Class (0) |
Actual (1) | TP - 79.55 | FN - 5.45 |
Actual (0) | FP - 54.55 | TN - 19.45 |
Accuracy : 62.264155 %.
Average TPR - 0.9358 Average FPR - 0.7371
```

Manhattan:

```
Average values for the confusion matrix:
                    Predicted
             |Class (1) |Class (0) |
Actual (1) | TP - 79.8 | FN - 5.2 |
Actual (0) | FP - 57.15 | TN - 16.85 |
Accuracy : 60.786165 %.
Average TPR - 0.938 Average FPR - 0.772
```

Complete link:

```
Average values for the confusion matrix:
                    Predicted
             |Class (1) |Class (0) |
Actual (1) | TP - 46.7 | FN - 38.3 |
Actual (0) | FP - 21.7 | TN - 52.3 |
Accuracy : 62.26416 %.
Average TPR - 0.5494 Average FPR - 0.2932
```

For white wine dataset:

Euclidean:

```
Average values for the confusion matrix:
                         Predicted
               |Class (1) |Class (0) |
Actual (1) | TP - 318.55 | FN - 7.45 |
Actual (0) | FP - 146.4 | TN - 17.6 |
Accuracy : 68.60203 %.
Average TPR - 0.9771 Average FPR - 0.8926
```

Manhattan:

```
Average values for the confusion matrix:
                         Predicted
               |Class (1) |Class (0) |
Actual (1) | TP - 317.7 | FN - 8.3 |
Actual (0) | FP - 147.0 | TN - 17.0 |
Accuracy : 68.30612 %.
Average TPR - 0.9745 Average FPR - 0.8963
```

Complete Link:

```
Average values for the confusion matrix:
                         Predicted
               |Class (1) |Class (0) |
Actual (1) | TP - 151.95 | FN - 174.05 |
Actual (0) | FP - 31.2 | TN - 132.8 |
Accuracy : 58.11225 %.
Average TPR - 0.4661 Average FPR - 0.19024
```

For Bank Dataset:

Euclidean:

```
Average values for the confusion matrix:
                         Predicted
               |Class (1) |Class (0) |
Actual (1) | TP - 40.1 | FN - 11.9 |
Actual (0) | FP - 163.5 | TN - 236.5 |
Accuracy : 61.19469 %.
Average TPR - 0.7711 Average FPR - 0.40875
```

Manhattan:

```
Average values for the confusion matrix:
                         Predicted
               |Class (1) |Class (0) |
Actual (1) | TP - 38.85 | FN - 13.15 |
Actual (0) | FP - 157.05 | TN - 242.95 |
Accuracy : 62.34513 %.
Average TPR - 0.7471 Average FPR - 0.39262
```

Complete Link:

```
Average values for the confusion matrix:
                         Predicted
               |Class (1) |Class (0) |
Actual (1) | TP - 0.0 | FN - 52.0 |
Actual (0) | FP - 0.0 | TN - 400.0 |
Accuracy : 88.4956 %.
Average TPR - 0.0 Average FPR - 0.0
```

b. Logistic Regression:

For Logistic Regression, the predictions were computed for etta values of 0.9, 0.81, 0.65, 0.43 and 0.18 with iterations going from 10 to 100. The average confusion matrix, accuracy, tpr and fpr are computed. The values shown here are only for etta values that gave best performance for each dataset respectively.

Red Wine Dataset:

```
Etta ==  0.43046721
Average accuracy =  68.36479
Average TPR =  0.490588235294
Average FPR =  0.0945945945946
Confusion Matrix:
         Predicted
           |Class (1) |Class (0) |
Actual (1) | TP - 41 | FN - 43 |
Actual (0) | FP - 7 | TN - 67 |
```

White Wine Dataset:

```
Etta ==  0.185302018885
Average accuracy =  71.14286
Average TPR =  0.946932515337
Average FPR =  0.756707317073
Confusion Matrix:
         Predicted
           |Class (1) |Class (0) |
Actual (1) | TP - 308 | FN - 17 |
Actual (0) | FP - 124 | TN - 39 |
```

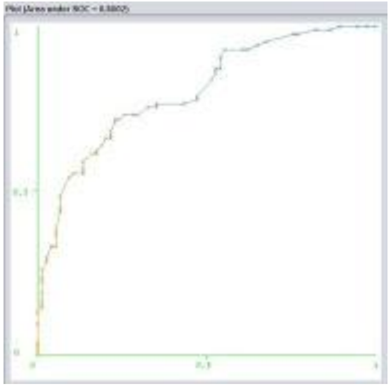
Bank Dataset:

```
Etta ==  0.185302018885
Average accuracy =  87.54425
Average TPR =  0.221153846154
Average FPR =  0.0395
Confusion Matrix:
         Predicted
           |Class (1) |Class (0) |
Actual (1) | TP - 11 | FN - 40 |
Actual (0) | FP - 15 | TN - 384 |
```

The ROC curves for the two algorithms plotted using weka are as follows:

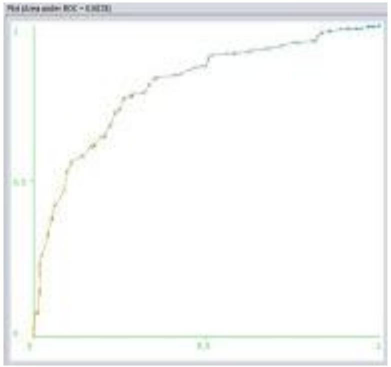
a. **K nearest neighbor:**

Red wine dataset



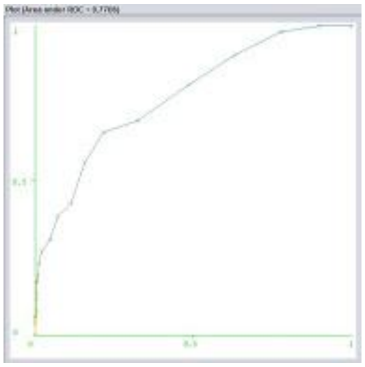
AUC = 0.8002

White Wine dataset



AUC = 0.8038

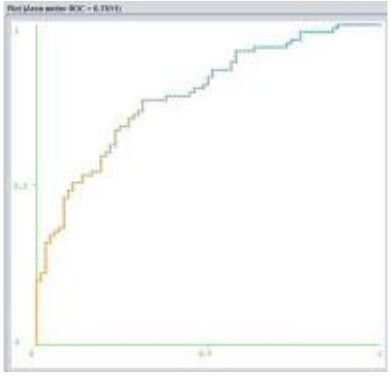
Bank Dataset:



AUC = 0.7706

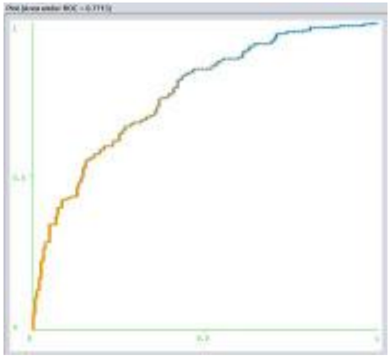
b. Logistic Regression:

Red Wine Dataset:



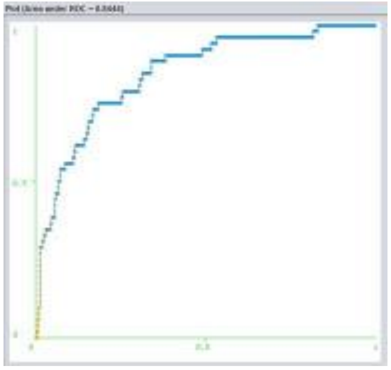
AUC: 0.7411

White Wine Dataset



AUC: 0.7713

Bank Dataset:



AUC: 0.8444

3. Performance:

K-nearest neighbours:

For K-nearest neighbours the performance was measured using 3 different distance metric:

a. Euclidean distance

b. Manhattan

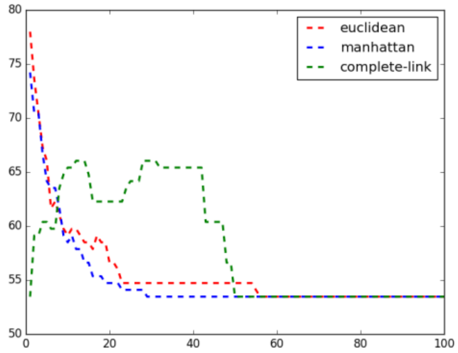
c. Complete Link

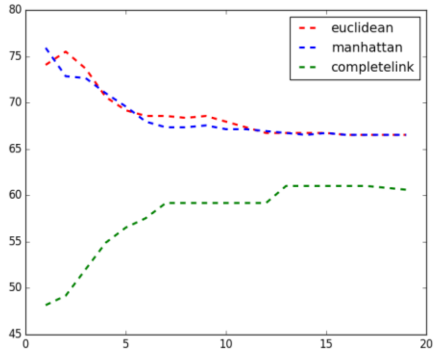
For each of these distance metrics, performance was measured over different values of k ranging between 1 and 50. The performance was initially measured for all the features.

Following are the performance graphs:

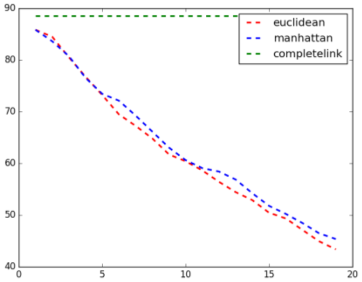
Red Wine Dataset:

White Wine Dataset:





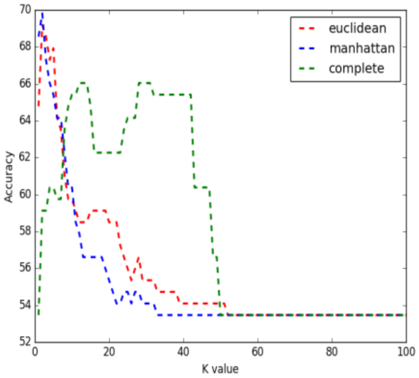
Bank Dataset:



Then the performance was measured by selecting the best features according to weka.

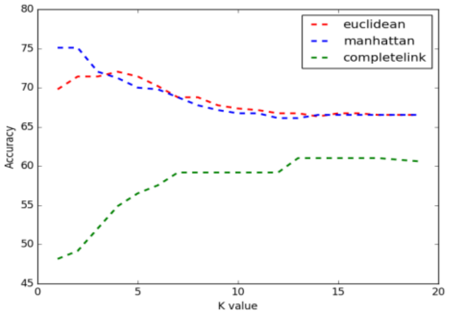
Red Wine Dataset:

The best performance was observed by selecting top 4 features. The performance graph is as below:



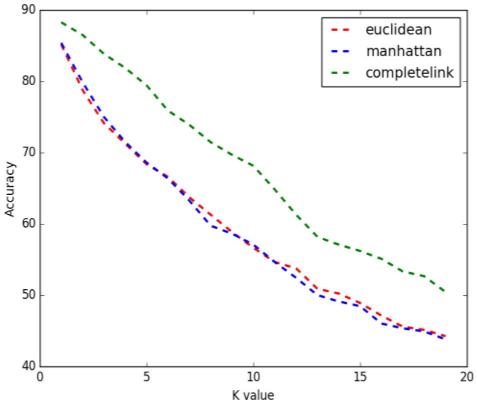
White Wine Dataset:

The best performance was observed by selecting top 5 features. The performance graph is as below:



Bank Dataset:

The best performance can be obtained by selecting top 7 features. The performance was observed as shown in the graph.



Analysis:

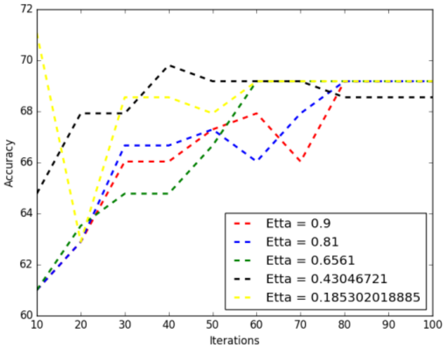
It can be observed that by performing the feature selection, the performance improves for values of k ≈40. Thus, it can also be observed that using irrelevant features affects the performance of knn algorithm and hence, feature selection plays an important role in k nearest neighbor.

However as the value of k goes on increasing the performance reduces drastically and converges to a much lower accuracy. In case of bank dataset, the performance doesn't improve significantly with increase in value of k. By observing performance over range of k values, it might be deduced that the performance decreases drastically after certain value of k in wine dataset. This might be due to generalization.

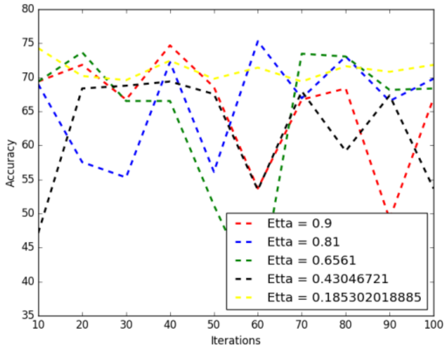
Logistic Regression:

For logistic regression, the performance was measured over multiple iterations with varying etta values:

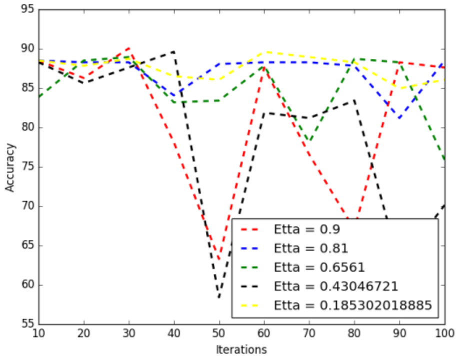
Red Wine Dataset:



White Wine Dataset:



Bank Dataset:



From the above graphs it can be seen that:

For wine data set the weights converge around 70 iterations with best accuracy for etta = 0.1853 (accuracy = 69.0741).

For white wine the weights converge around 110 iterations with best accuracy for etta = 0.1853 (accuracy = 73.62).

For Bank data set, the weights converge around 105 iterations except for etta = 0.43 with best accuracy for etta = 0.81 (accuracy = 88. 7853).

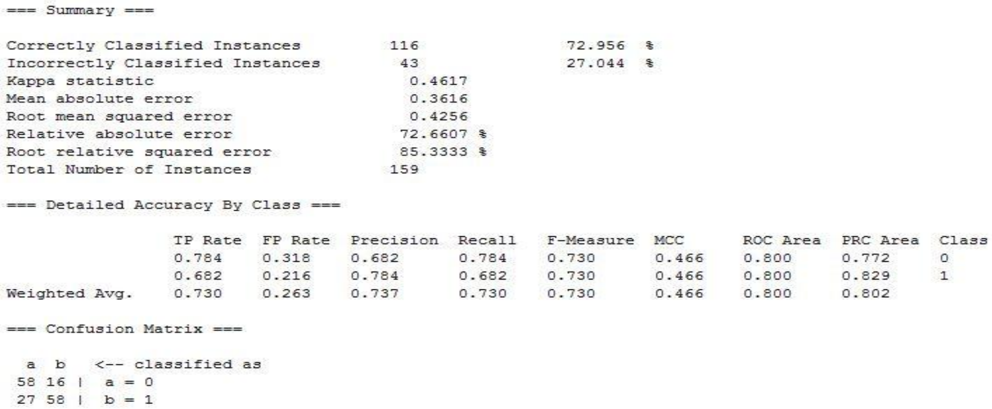
Analysis:

It can be observed that Logistic Regression gives better performance over all the datasets compared to k-nn algorithm. It might be because the data sets might be linearly separable. Also logistic regression seems to be much more robust to outliers compared to k nearest neighbours and converges to a descent accuracy if ran over multiple iterations. Also, logistic regression can be seen to handle class bias better than knn.

4. Weka Results:

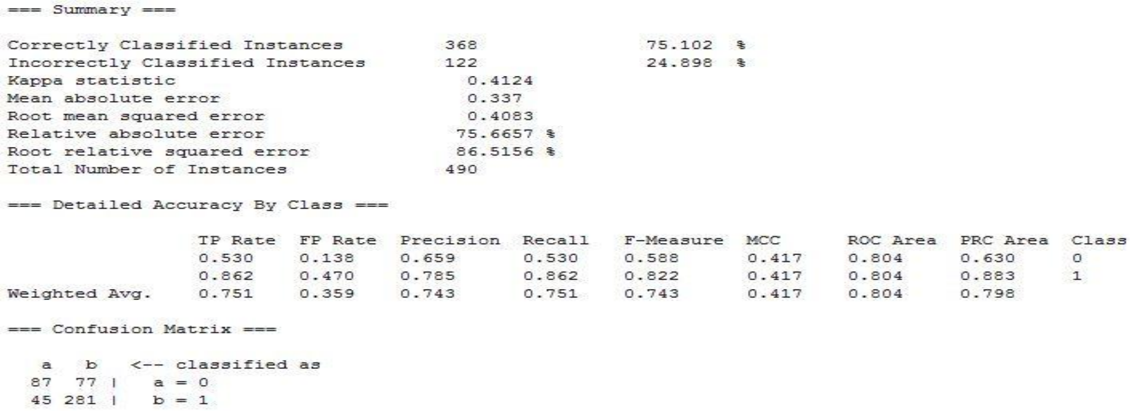
a. K nearest neighbor:

Red Wine Dataset:



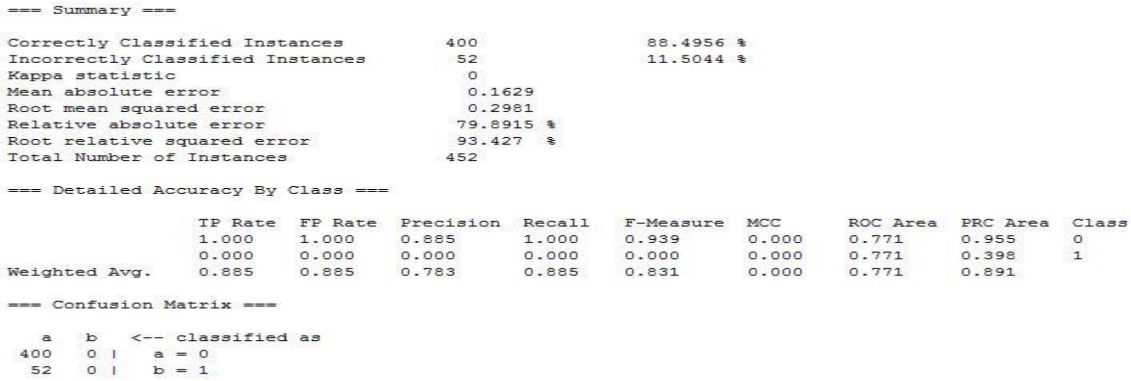
```
=== Summary ===

Correctly Classified Instances         116               72.956  %
Incorrectly Classified Instances        43               27.044  %
Kappa statistic                          0.4617
Mean absolute error                      0.3616
Root mean squared error                  0.4256
Relative absolute error                 72.6607 %
Root relative squared error             85.3333 %
Total Number of Instances              159

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.784    0.318    0.682      0.784    0.730      0.466    0.800     0.772     0
                 0.682    0.216    0.784      0.682    0.730      0.466    0.800     0.829     1
Weighted Avg.    0.730    0.263    0.737      0.730    0.730      0.466    0.800     0.802

=== Confusion Matrix ===

  a  b   <-- classified as
 58 16 |  a = 0
 27 58 |  b = 1
```

White Wine Dataset:



```
=== Summary ===

Correctly Classified Instances         368               75.102  %
Incorrectly Classified Instances       122               24.898  %
Kappa statistic                          0.4124
Mean absolute error                      0.337
Root mean squared error                  0.4083
Relative absolute error                 75.6657 %
Root relative squared error             86.5156 %
Total Number of Instances              490

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.530    0.138    0.659      0.530    0.588      0.417    0.804     0.630     0
                 0.862    0.470    0.785      0.862    0.822      0.417    0.804     0.883     1
Weighted Avg.    0.751    0.359    0.743      0.751    0.743      0.417    0.804     0.798

=== Confusion Matrix ===

   a   b   <-- classified as
  87  77 |   a = 0
  45 281 |   b = 1
```

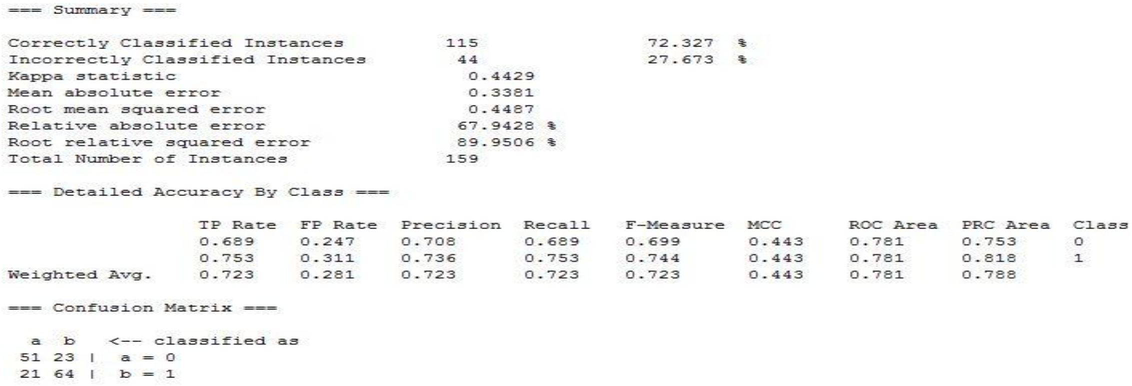
Bank Dataset:



```
=== Summary ===

Correctly Classified Instances         400               88.4956 %
Incorrectly Classified Instances        52               11.5044 %
Kappa statistic                          0
Mean absolute error                      0.1629
Root mean squared error                  0.2981
Relative absolute error                 79.8915 %
Root relative squared error             93.427  %
Total Number of Instances              452

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.885      1.000    0.939      0.000    0.771     0.955     0
                 0.000    0.000    0.000      0.000    0.000      0.000    0.771     0.398     1
Weighted Avg.    0.885    0.885    0.783      0.885    0.831      0.000    0.771     0.891

=== Confusion Matrix ===

   a   b   <-- classified as
 400   0 |   a = 0
  52   0 |   b = 1
```

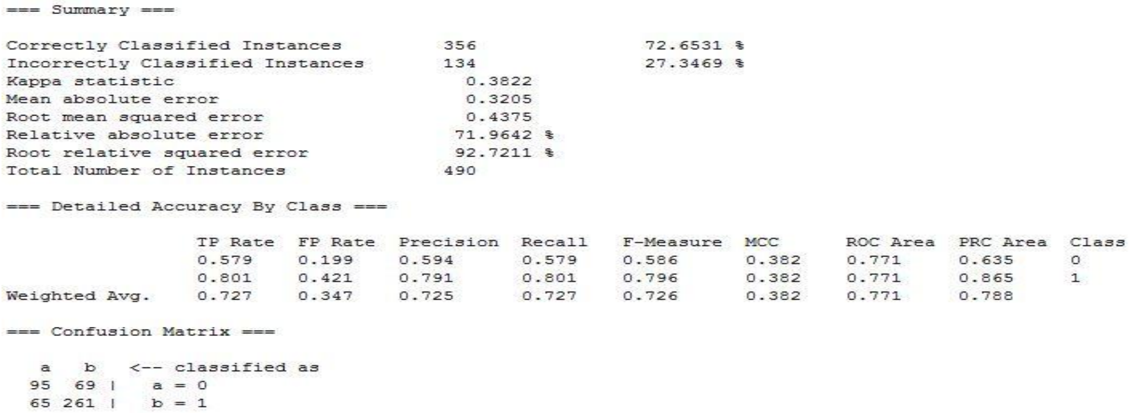
b. Logistic Regression:

Red Wine Dataset:



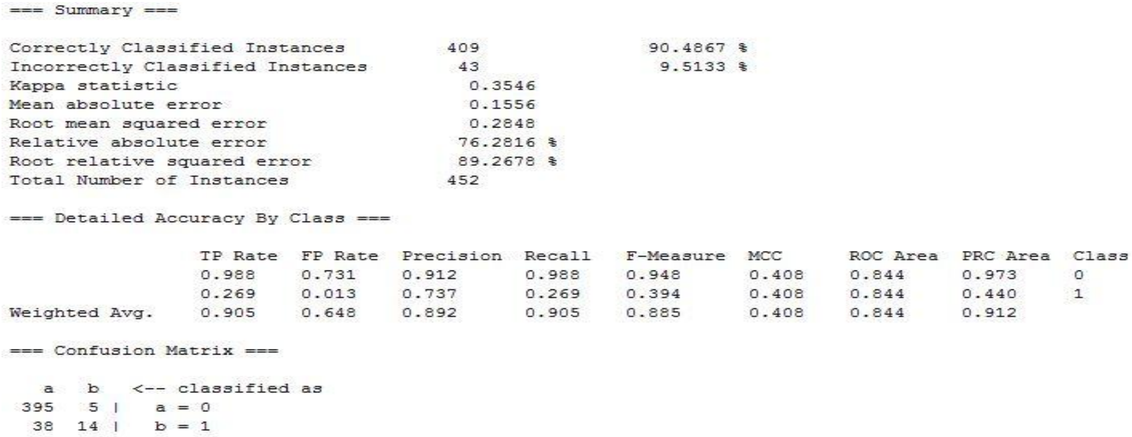
```
=== Summary ===

Correctly Classified Instances         115               72.327  %
Incorrectly Classified Instances        44               27.673  %
Kappa statistic                          0.4429
Mean absolute error                      0.3381
Root mean squared error                  0.4487
Relative absolute error                 67.9428 %
Root relative squared error             89.9506 %
Total Number of Instances              159

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.689    0.247    0.708      0.689    0.699      0.443    0.781     0.753     0
                 0.753    0.311    0.736      0.753    0.744      0.443    0.781     0.818     1
Weighted Avg.    0.723    0.281    0.723      0.723    0.723      0.443    0.781     0.788

=== Confusion Matrix ===

  a  b   <-- classified as
 51 23 |  a = 0
 21 64 |  b = 1
```

White Wine Dataset:



```
=== Summary ===

Correctly Classified Instances         356               72.6531 %
Incorrectly Classified Instances       134               27.3469 %
Kappa statistic                          0.3822
Mean absolute error                      0.3205
Root mean squared error                  0.4375
Relative absolute error                 71.9642 %
Root relative squared error             92.7211 %
Total Number of Instances              490

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.579    0.199    0.594      0.579    0.586      0.382    0.771     0.635     0
                 0.801    0.421    0.791      0.801    0.796      0.382    0.771     0.865     1
Weighted Avg.    0.727    0.347    0.725      0.727    0.726      0.382    0.771     0.788

=== Confusion Matrix ===

   a   b   <-- classified as
  95  69 |   a = 0
  65 261 |   b = 1
```

Bank Dataset:



```
=== Summary ===

Correctly Classified Instances         409               90.4867 %
Incorrectly Classified Instances        43                9.5133 %
Kappa statistic                          0.3546
Mean absolute error                      0.1556
Root mean squared error                  0.2848
Relative absolute error                 76.2816 %
Root relative squared error             89.2678 %
Total Number of Instances              452

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.988    0.731    0.912      0.988    0.948      0.408    0.844     0.973     0
                 0.269    0.013    0.737      0.269    0.394      0.408    0.844     0.440     1
Weighted Avg.    0.905    0.648    0.892      0.905    0.885      0.408    0.844     0.912

=== Confusion Matrix ===

   a   b   <-- classified as
 395   5 |   a = 0
  38  14 |   b = 1
```