Programming Assignment 1
KNN Algorithm

Developed By:

Name: Jeyamaruthi Jayakumar
ID: 1001757737
Department of Computer Science & Engineering
The University of Texas, Arlington
United States of America
Portfolio

Table of Contents

1 Project Description

Creation of KNN algorithm from scratch i.e. without using any libraries. The program consists of a method to evaluate k-fold cross-validation, vectorized computation of the distances using the formula of Euclidean, Hamming & Manhattan.

We developed the KNN model by using 4 different datasets:

- Iris Dataset
- Hayes-Roth Dataset
- Car Evaluation Dataset
- Breast Cancer Dataset.

We have 9 unique functions that are used by the dataset to fulfill the KNN algorithm, predict and display the accuracy score.

2 Method Description

We have a total of 9 methods to predict the classes using the KNN algorithm.

2.1 LoadFile:

- ➢ The method created to read the data file and convert into a list
- ➢ Returns the dataset as a list and utilized for further use.

2.2 StringToFloat:

- ➢ The Method was created to read the columns which have string numeric type and converts the string into a float.
- ➢ The method is handled with an **exception**, which passes the column if it cannot be converted into a float.

2.3 StringToInt:

- The Method created to convert the string column into an integer
- It is done by using a dictionary, which maps a unique value for each individual string, which helps us to predict the classification.

2.4 EvaluationMethod:

- Firstly, it will call the cross-validation function to get the fold segregation of datasets.
- Clean the data to make it adaptive for using it for predicting the classes.
- Finally, calls the accuracyMetric function to get the score of prediction in percentage.

2.5 KFoldCrossValidation:

- This method helps us to segment the dataset with k-folds
- For our assignment, we have used 10, hence the dataset will give u 10 folds, where 1 will be used for testing and others are for training.

2.6 kNearestNeighbours:

- The method which calls the predictClassification method to get the prediction.
- Then, holds the value for each row in a list, which is used for comparing with the actual list.

2.7 PredictClassification:

- Calls the getNeighbors function and stores the output values in a list
- Gets the maximum value of the list for the required number of neighboring elements.

2.8 GetNeighbors:

- It calculates the distance between the datasets by 3 methods.

- If the given D value is 1:
  - *Calculates the distance using Euclidian*
- If the given D value is 2:
  - *Calculates the distance using Hamming*
- If the given D value is 2:
  - *Calculates the distance using Manhattam*
- Sort the calculated distance list and append in neighbors list for the required number of neighbors i.e. if N = 5, it will fetch 5 maximum elements from the output list.

2.9 Methods for Calculation Distance:

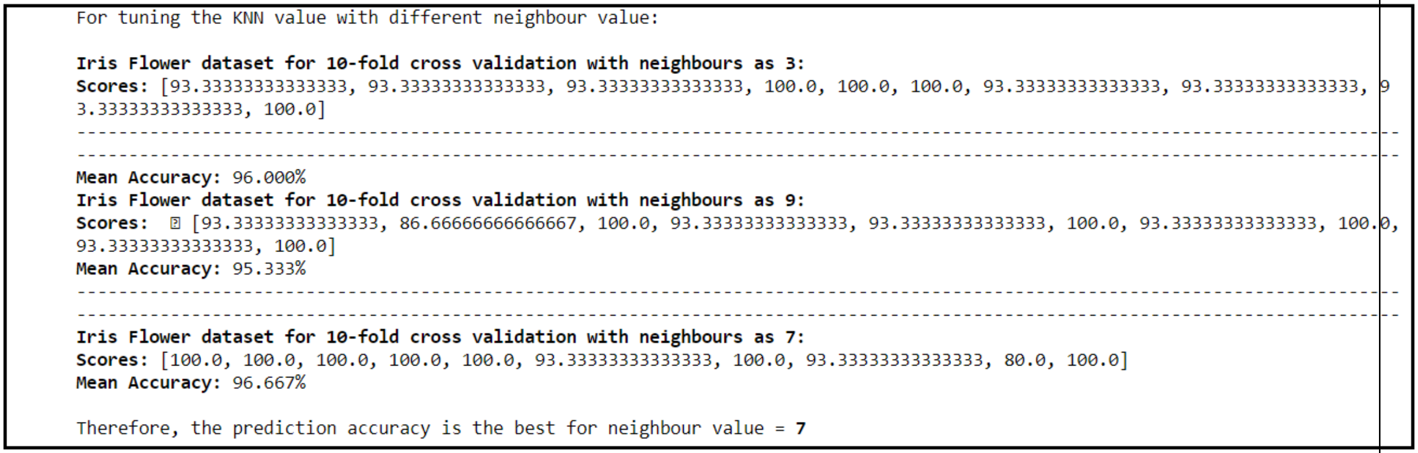
- **EuclideanDistance:** Method used to calculate distance using Euclidean formula.
- **HammingDistance:** Method used to calculate distance using Hamming formula.
- **ManhattanDistance:** Method used to calculate distance using the Manhattan formula.

3 Detail Description

This segment helps to get a detailed overview of how the assignment works:

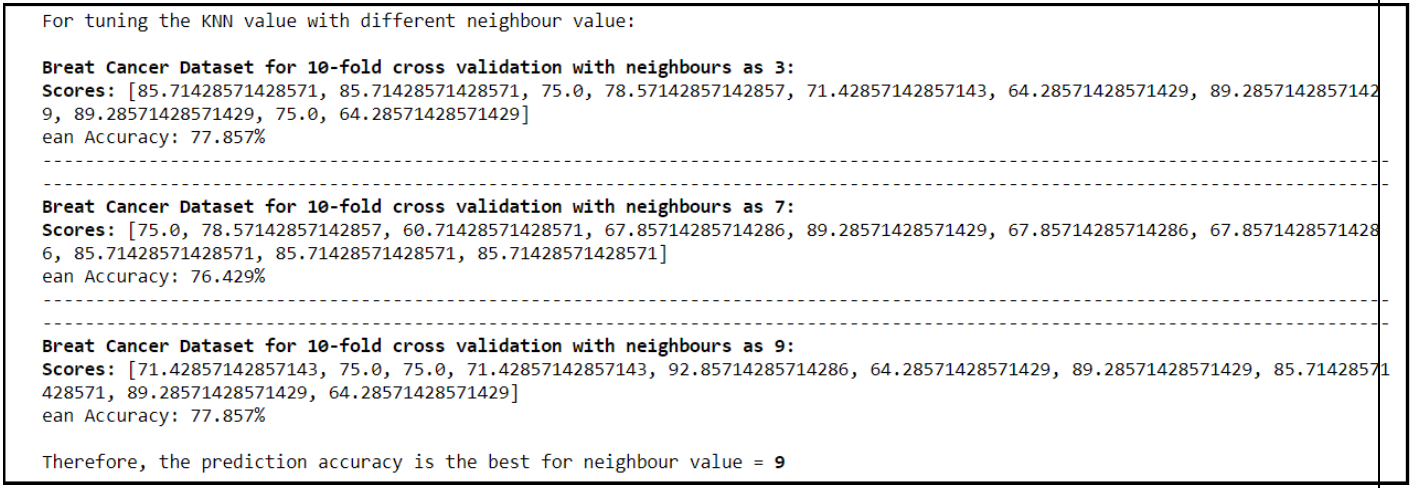
- ✓ After fetching the data using the loadFile method we use the fetched dataset to convert the dataset into appropriate datatype for fitting in the model.
- ✓ Once the desired dataset is attained, we fit it into the model and start predicting the value.
- ✓ After prediction, we compare it with the original dataset and check the prediction score.
- ✓ We run a specific dataset 3 times I.e. for **Euclidean, Hamming & Manhattan** Distance.

- ✓ While running the model, we add a parameter named **d,** which helps the model to understand which formula we are going to use. For example, if d is equal to 1, then it uses the Euclidean Distance formula to calculate the distance.
- ✓ We tune the KNN model, by changing the neighbor value while passing the dataset into the model. For example, neighbors value as: 3, 5, 7 or 9
- ✓ **So,** for the **Iris** dataset the output would be like this (**Using Euclidean Distance)**:



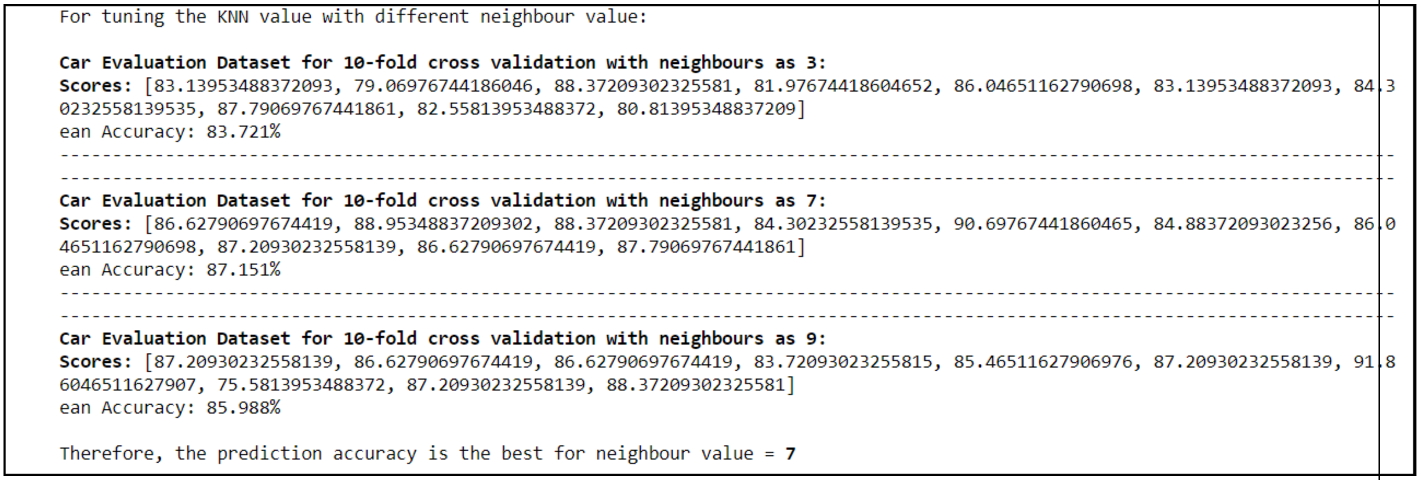
```
For tuning the KNN value with different neighbour value:

Iris Flower dataset for 10-fold cross validation with neighbours as 3:
Scores: [93.33333333333333, 93.33333333333333, 93.33333333333333, 100.0, 100.0, 100.0, 93.33333333333333, 93.33333333333333, 9
3.33333333333333, 100.0]
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
Mean Accuracy: 96.000%
Iris Flower dataset for 10-fold cross validation with neighbours as 9:
Scores:  □ [93.33333333333333, 86.66666666666667, 100.0, 93.33333333333333, 93.33333333333333, 100.0, 93.33333333333333, 100.0,
93.33333333333333, 100.0]
Mean Accuracy: 95.333%
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
Iris Flower dataset for 10-fold cross validation with neighbours as 7:
Scores: [100.0, 100.0, 100.0, 100.0, 100.0, 93.33333333333333, 100.0, 93.33333333333333, 80.0, 100.0]
Mean Accuracy: 96.667%

Therefore, the prediction accuracy is the best for neighbour value = 7
```

- ✓ **Similarly,** for **Breast Cancer** Dataset:



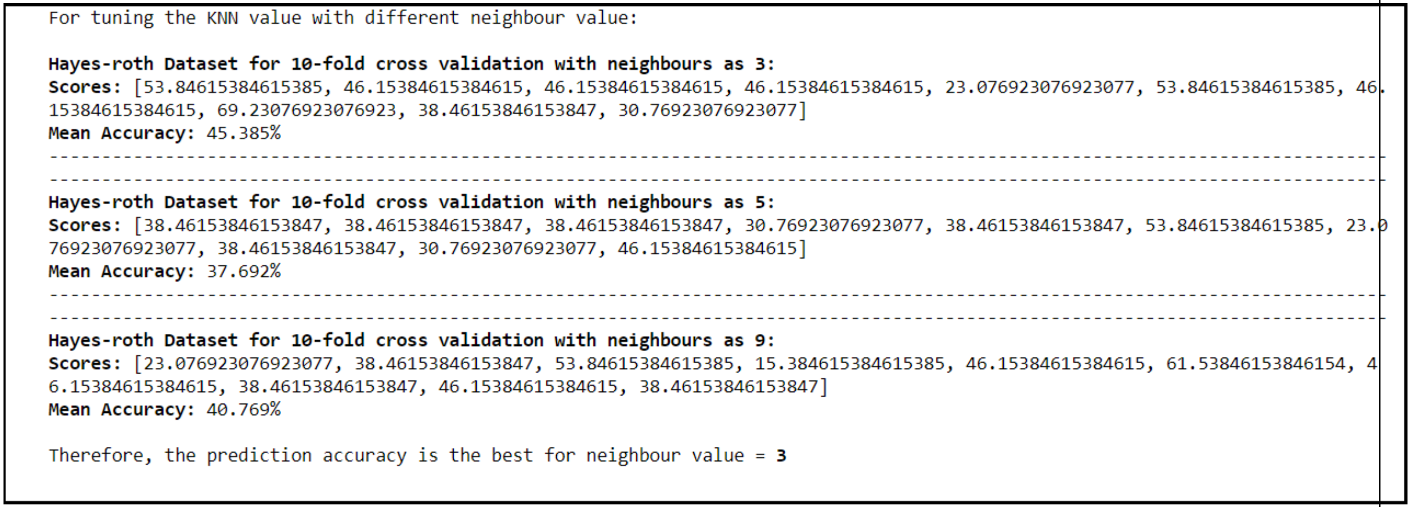
```
For tuning the KNN value with different neighbour value:

Breat Cancer Dataset for 10-fold cross validation with neighbours as 3:
Scores: [85.71428571428571, 85.71428571428571, 75.0, 78.57142857142857, 71.42857142857143, 64.28571428571429, 89.2857142857142
9, 89.28571428571429, 75.0, 64.28571428571429]
ean Accuracy: 77.857%
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 7:
Scores: [75.0, 78.57142857142857, 60.71428571428571, 67.85714285714286, 89.28571428571429, 67.85714285714286, 67.8571428571428
6, 85.71428571428571, 85.71428571428571, 85.71428571428571]
ean Accuracy: 76.429%
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 9:
Scores: [71.42857142857143, 75.0, 75.0, 71.42857142857143, 92.85714285714286, 64.28571428571429, 89.28571428571429, 85.71428571
428571, 89.28571428571429, 64.28571428571429]
ean Accuracy: 77.857%

Therefore, the prediction accuracy is the best for neighbour value = 9
```

- ✓ For **Car Evaluation** Dataset:



```
For tuning the KNN value with different neighbour value:

Car Evaluation Dataset for 10-fold cross validation with neighbours as 3:
Scores: [83.13953488372093, 79.06976744186046, 88.37209302325581, 81.97674418604652, 86.04651162790698, 83.13953488372093, 84.3
0232558139535, 87.79069767441861, 82.55813953488372, 80.81395348837209]
ean Accuracy: 83.721%
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 7:
Scores: [86.62790697674419, 88.95348837209302, 88.37209302325581, 84.30232558139535, 90.69767441860465, 84.88372093023256, 86.0
4651162790698, 87.20930232558139, 86.62790697674419, 87.79069767441861]
ean Accuracy: 87.151%
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 9:
Scores: [87.20930232558139, 86.62790697674419, 86.62790697674419, 83.72093023255815, 85.46511627906976, 87.20930232558139, 91.8
6046511627907, 75.5813953488372, 87.20930232558139, 88.37209302325581]
ean Accuracy: 85.988%

Therefore, the prediction accuracy is the best for neighbour value = 7
```

- ✓ For **Hayes-Roth** Dataset:



```
For tuning the KNN value with different neighbour value:

Hayes-roth Dataset for 10-fold cross validation with neighbours as 3:
Scores: [53.84615384615385, 46.15384615384615, 46.15384615384615, 46.15384615384615, 23.076923076923077, 53.84615384615385, 46.
15384615384615, 69.23076923076923, 38.46153846153847, 30.76923076923077]
Mean Accuracy: 45.385%
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 5:
Scores: [38.46153846153847, 38.46153846153847, 38.46153846153847, 30.76923076923077, 38.46153846153847, 53.84615384615385, 23.0
76923076923077, 38.46153846153847, 30.76923076923077, 46.15384615384615]
Mean Accuracy: 37.692%
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 9:
Scores: [23.076923076923077, 38.46153846153847, 53.84615384615385, 15.384615384615385, 46.15384615384615, 61.53846153846154, 4
6.15384615384615, 38.46153846153847, 46.15384615384615, 38.46153846153847]
Mean Accuracy: 40.769%

Therefore, the prediction accuracy is the best for neighbour value = 3
```

- ✓ **Likewise,** we will get **two** set of 4 more outputs using Hamming & Manhattan.

4 Comparison of WEKA & Program

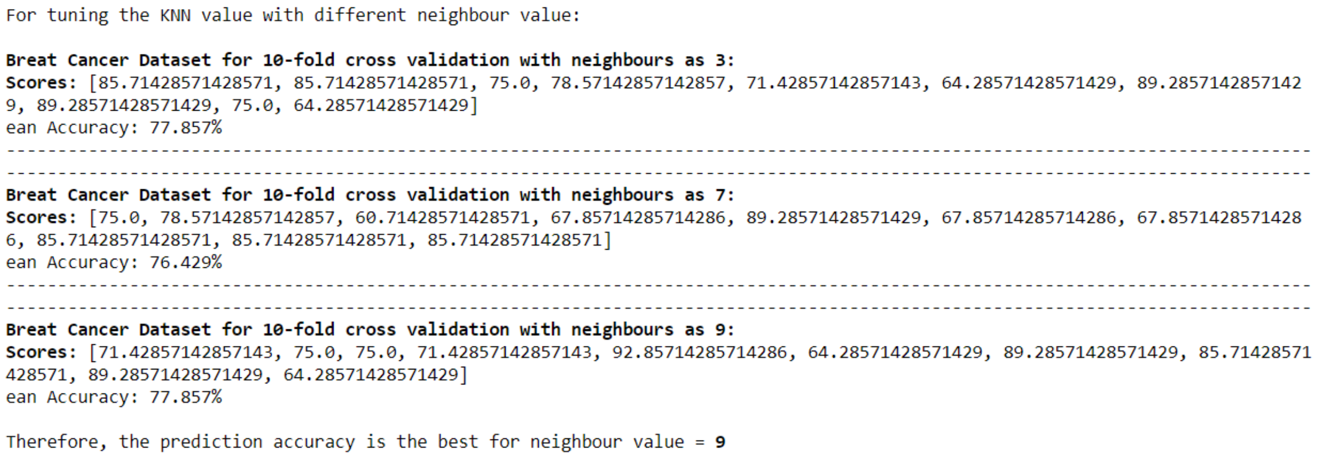
Let us compare the 3 datasets i.e. Car Evaluation, Breast Cancer & Hayes-Roth, with Weka to know how our prediction works.

Also, please check the **table** at last of this section for better comparision

4.1 Breast Cancer Dataset:

- **For Euclidean:**

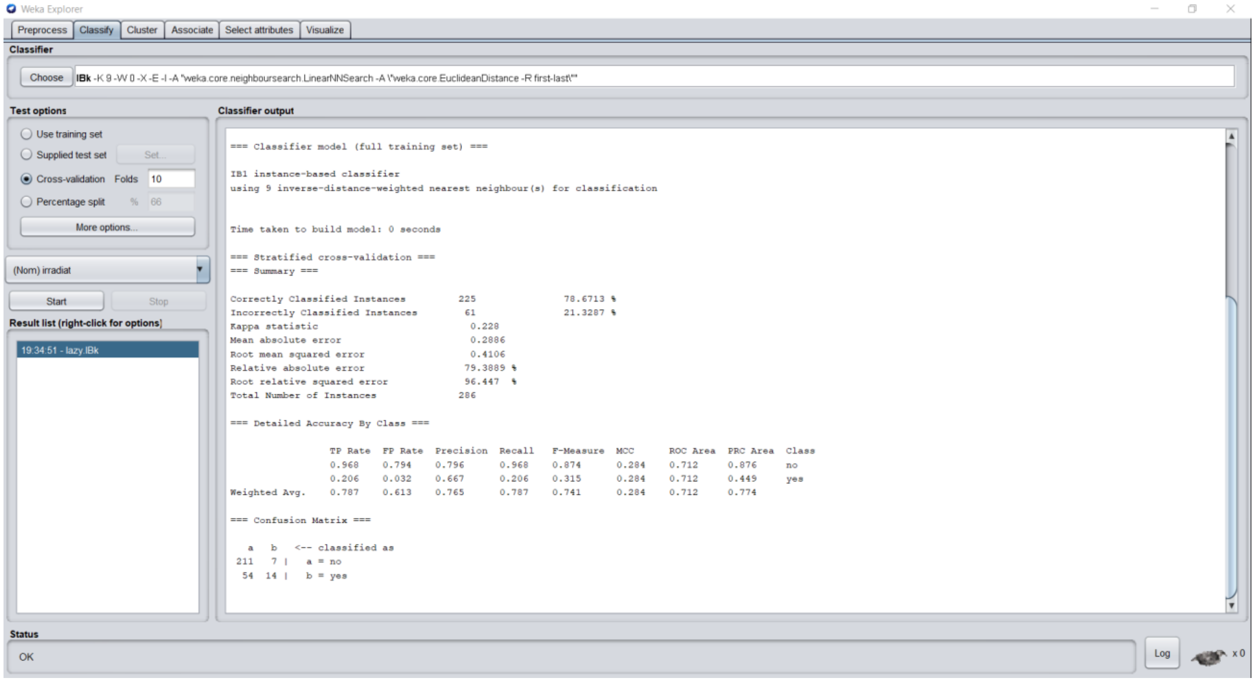
From the program, the screenshot attached will consist of output for 3 different neighbor values.



```
For tuning the KNN value with different neighbour value:

Breat Cancer Dataset for 10-fold cross validation with neighbours as 3:
Scores: [85.71428571428571, 85.71428571428571, 75.0, 78.57142857142857, 71.42857142857143, 64.28571428571429, 89.2857142857142
9, 89.28571428571429, 75.0, 64.28571428571429]
ean Accuracy: 77.857%
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 7:
Scores: [75.0, 78.57142857142857, 60.71428571428571, 67.85714285714286, 89.28571428571429, 67.85714285714286, 67.8571428571428
6, 85.71428571428571, 85.71428571428571, 85.71428571428571]
ean Accuracy: 76.429%
----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 9:
Scores: [71.42857142857143, 75.0, 75.0, 71.42857142857143, 92.85714285714286, 64.28571428571429, 89.28571428571429, 85.71428571
428571, 89.28571428571429, 64.28571428571429]
ean Accuracy: 77.857%

Therefore, the prediction accuracy is the best for neighbour value = 9
```

We will compare the best accuracy of the program with the **Weka**. For example, the above best prediction accuracy is with neighbor KNN value → 9.

Similarly, we run the **Weka** with **9**.



IBk -K 9 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) irradiat

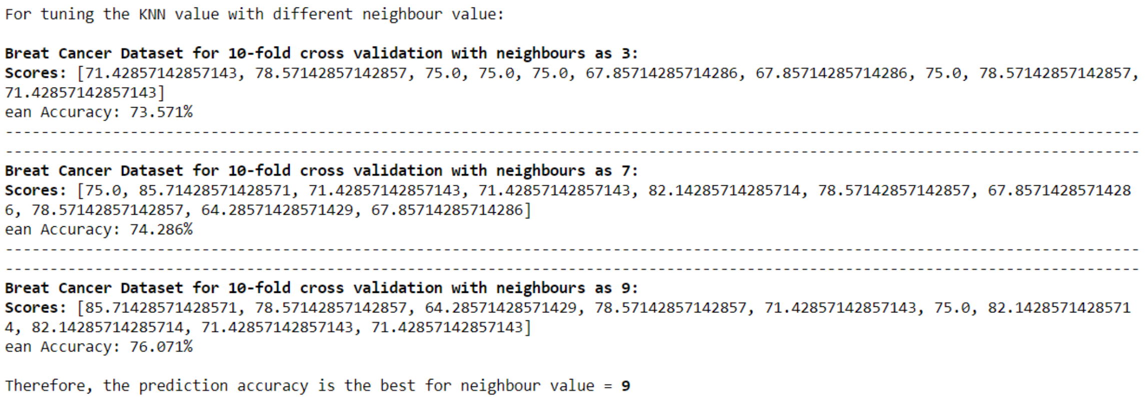
19:34:51 - lazy.IBk

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 9 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         225               78.6713 %
Incorrectly Classified Instances        61               21.3287 %
Kappa statistic                          0.228
Mean absolute error                      0.2886
Root mean squared error                  0.4106
Relative absolute error                 79.3889 %
Root relative squared error             96.447  %
Total Number of Instances              286

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.968    0.794    0.796      0.968    0.874      0.284    0.712     0.876     no
                 0.206    0.032    0.667      0.206    0.315      0.284    0.712     0.449     yes
Weighted Avg.    0.787    0.613    0.765      0.787    0.741      0.284    0.712     0.774

=== Confusion Matrix ===

   a   b   <-- classified as
 211   7 |   a = no
  54  14 |   b = yes
```

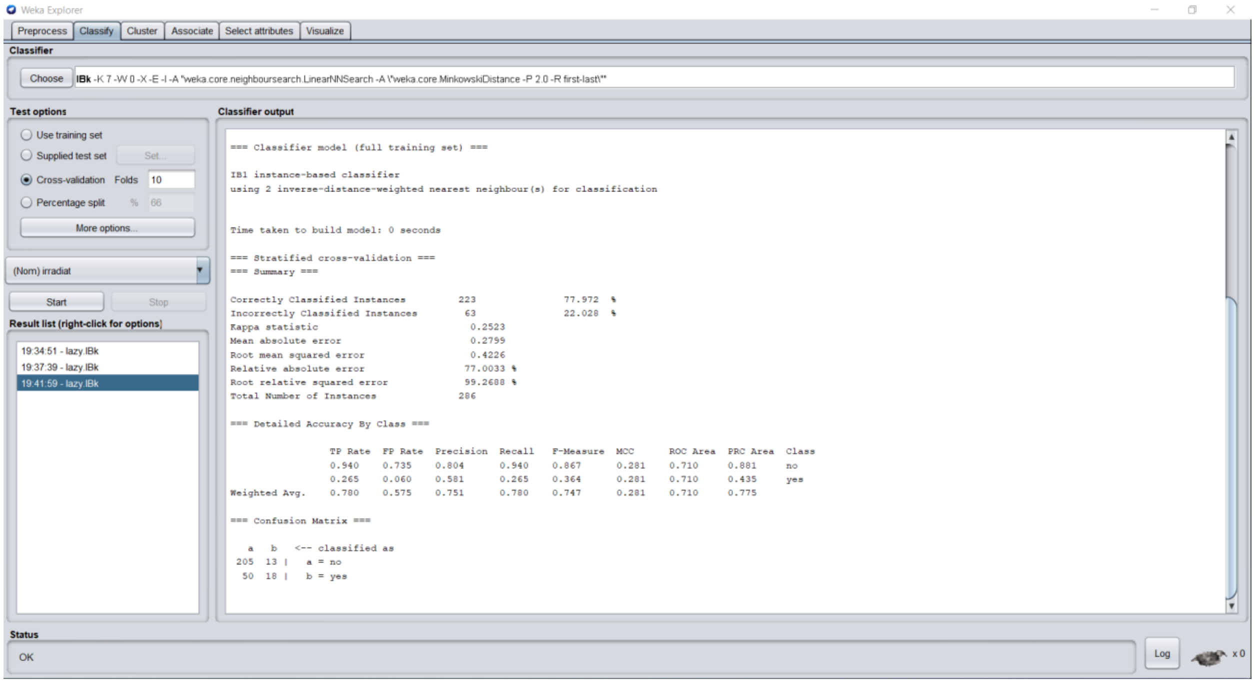
Status: OK

- **For Hamming:**

Like the previous subdivision, we follow the same rule for this and for the upcoming subdivisions.



```
For tuning the KNN value with different neighbour value:

Breat Cancer Dataset for 10-fold cross validation with neighbours as 3:
Scores: [71.42857142857143, 78.57142857142857, 75.0, 75.0, 75.0, 67.85714285714286, 67.85714285714286, 75.0, 78.57142857142857,
71.42857142857143]
ean Accuracy: 73.571%
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 7:
Scores: [75.0, 85.71428571428571, 71.42857142857143, 71.42857142857143, 82.14285714285714, 78.57142857142857, 67.8571428571428
6, 78.57142857142857, 64.28571428571429, 67.85714285714286]
ean Accuracy: 74.286%
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 9:
Scores: [85.71428571428571, 78.57142857142857, 64.28571428571429, 78.57142857142857, 71.42857142857143, 75.0, 82.1428571428571
4, 82.14285714285714, 71.42857142857143, 71.42857142857143]
ean Accuracy: 76.071%

Therefore, the prediction accuracy is the best for neighbour value = 9
```



Classifier

IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.MinkowskiDistance -P 2.0 -R first-last\""

Test options

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) irradiat

Result list (right-click for options)

19:34:51 - lazy.IBk
19:37:39 - lazy.IBk
19:41:59 - lazy.IBk

Classifier output

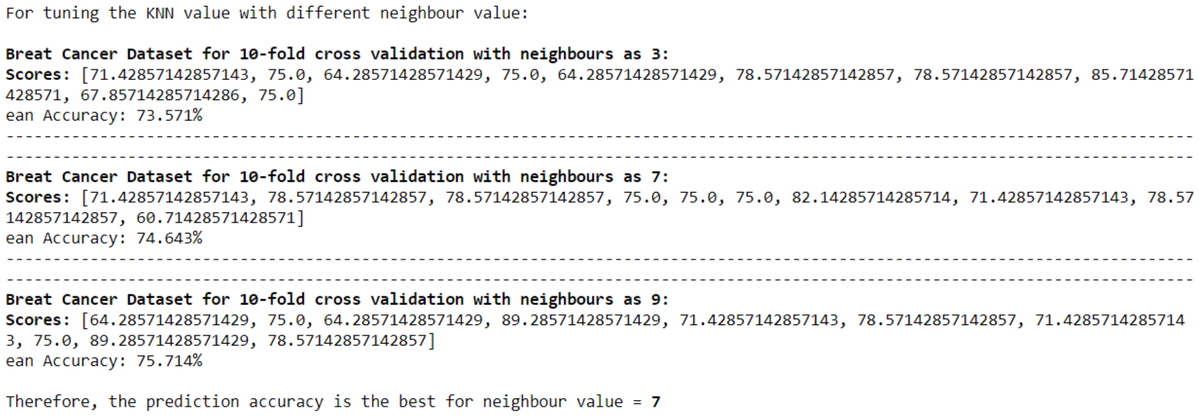
```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 2 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         223               77.972  %
Incorrectly Classified Instances        63               22.028  %
Kappa statistic                          0.2523
Mean absolute error                      0.2799
Root mean squared error                  0.4226
Relative absolute error                 77.0033 %
Root relative squared error             99.2688 %
Total Number of Instances              286

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.940    0.735    0.804      0.940    0.867      0.281    0.710     0.881     no
                 0.265    0.060    0.581      0.265    0.364      0.281    0.710     0.435     yes
Weighted Avg.    0.780    0.575    0.751      0.780    0.747      0.281    0.710     0.775

=== Confusion Matrix ===

   a   b   <-- classified as
 205  13 |   a = no
  50  18 |   b = yes
```

Status

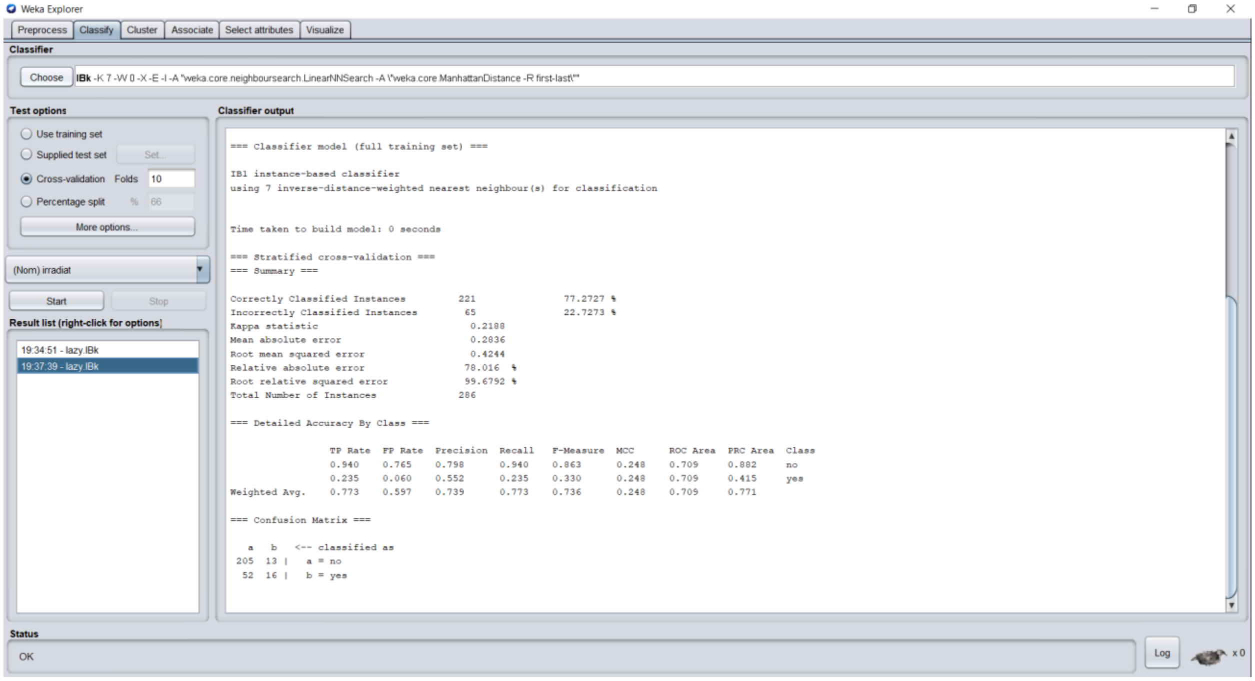
OK

x 0

- **For Manhattan:**



```
For tuning the KNN value with different neighbour value:

Breat Cancer Dataset for 10-fold cross validation with neighbours as 3:
Scores: [71.42857142857143, 75.0, 64.28571428571429, 75.0, 64.28571428571429, 78.57142857142857, 78.57142857142857, 85.71428571
428571, 67.85714285714286, 75.0]
ean Accuracy: 73.571%
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 7:
Scores: [71.42857142857143, 78.57142857142857, 78.57142857142857, 75.0, 75.0, 75.0, 82.14285714285714, 71.42857142857143, 78.57
142857142857, 60.71428571428571]
ean Accuracy: 74.643%
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
Breat Cancer Dataset for 10-fold cross validation with neighbours as 9:
Scores: [64.28571428571429, 75.0, 64.28571428571429, 89.28571428571429, 71.42857142857143, 78.57142857142857, 71.4285714285714
3, 75.0, 89.28571428571429, 78.57142857142857]
ean Accuracy: 75.714%

Therefore, the prediction accuracy is the best for neighbour value = 7
```



IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.ManhattanDistance -R first-last\""

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) irradiat

19:34:51 - lazy.IBk
19:37:39 - lazy.IBk

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 7 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         221               77.2727 %
Incorrectly Classified Instances        65               22.7273 %
Kappa statistic                          0.2188
Mean absolute error                      0.2836
Root mean squared error                  0.4244
Relative absolute error                 78.016  %
Root relative squared error             99.6792 %
Total Number of Instances              286

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.940    0.765    0.798      0.940    0.863      0.248    0.709     0.882     no
                 0.235    0.060    0.552      0.235    0.330      0.248    0.709     0.415     yes
Weighted Avg.    0.773    0.597    0.739      0.773    0.736      0.248    0.709     0.771

=== Confusion Matrix ===

   a   b   <-- classified as
 205  13 |   a = no
  52  16 |   b = yes
```

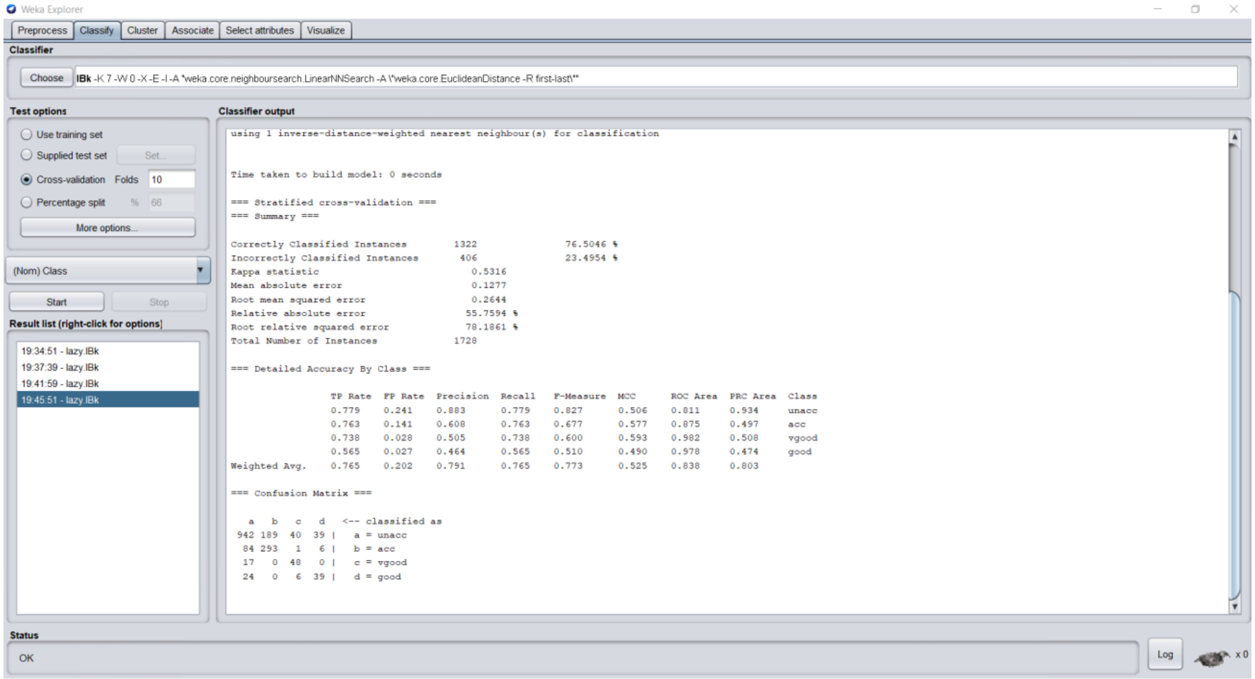
OK

4.2 Car Evaluation Dataset:

Like previous, we follow the same. But while programming, we have **deleted 2 columns (in code)** which are of **least** priority. By doing that the accuracy increased by 10 i.e. it changed from **70 - 75% → 80 – 85%**

- **For Euclidean:**

```
For tuning the KNN value with different neighbour value:

Car Evaluation Dataset for 10-fold cross validation with neighbours as 3:
Scores: [88.37209302325581, 90.11627906976744, 88.95348837209302, 90.11627906976744, 88.95348837209302, 90.11627906976744, 84.3
0232558139535, 85.46511627906976, 84.88372093023256, 83.72093023255815]
ean Accuracy: 87.500%
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 7:
Scores: [85.46511627906976, 83.13953488372093, 84.88372093023256, 88.37209302325581, 90.11627906976744, 88.95348837209302, 84.8
8372093023256, 86.62790697674419, 84.88372093023256, 83.72093023255815]
ean Accuracy: 86.105%
-----------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 9:
Scores: [89.53488372093024, 84.88372093023256, 86.04651162790698, 86.62790697674419, 88.37209302325581, 84.30232558139535, 85.4
6511627906976, 83.13953488372093, 86.62790697674419, 83.13953488372093]
ean Accuracy: 85.814%

Therefore, the prediction accuracy is the best for neighbour value = 7
```



IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

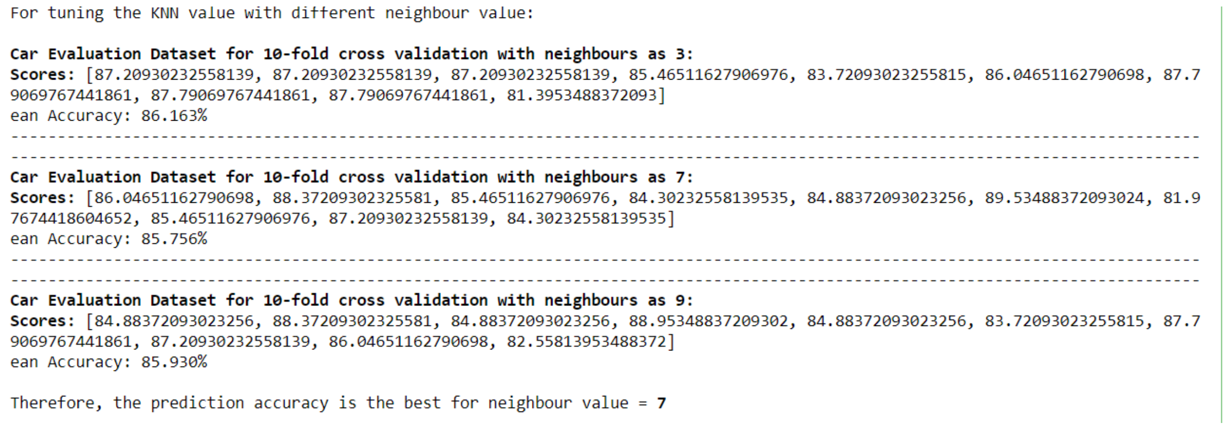
(Nom) Class

19:34:51 - lazy.IBk
19:37:39 - lazy.IBk
19:41:59 - lazy.IBk
19:45:51 - lazy.IBk

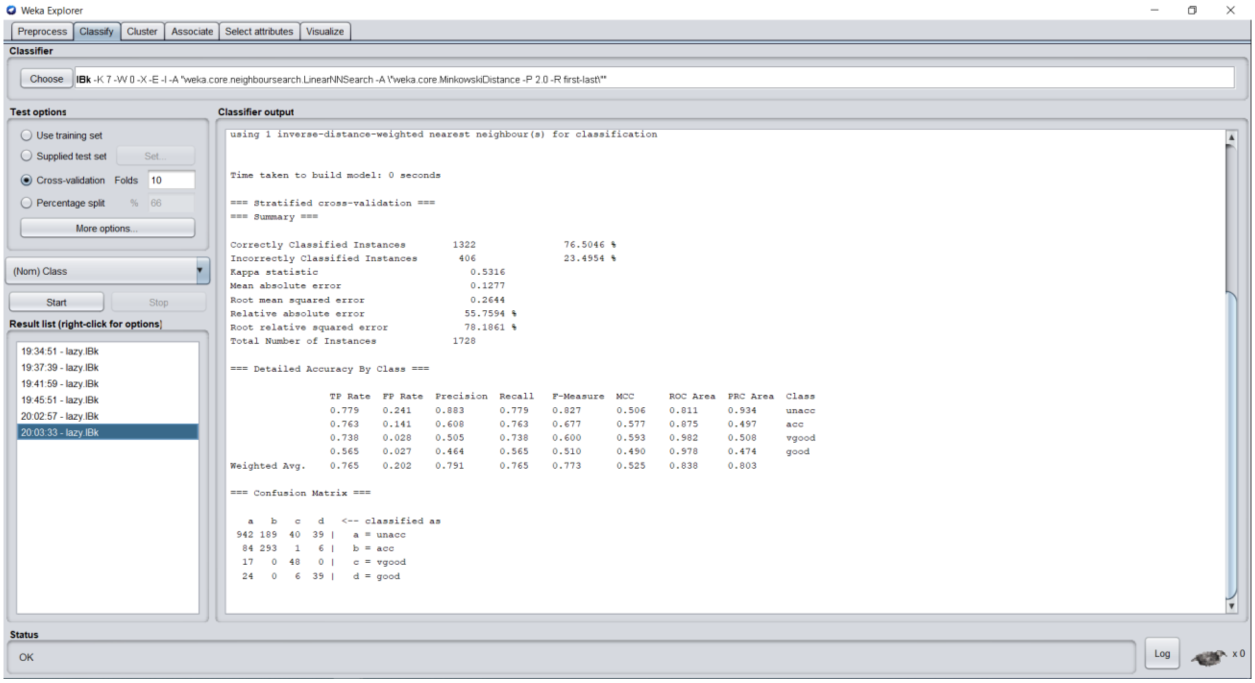
```
using 1 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1322               76.5046 %
Incorrectly Classified Instances       406               23.4954 %
Kappa statistic                          0.5316
Mean absolute error                      0.1277
Root mean squared error                  0.2644
Relative absolute error                 55.7594 %
Root relative squared error             78.1861 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.779    0.241    0.883      0.779    0.827      0.506    0.811     0.934     unacc
                 0.763    0.141    0.608      0.763    0.677      0.577    0.875     0.497     acc
                 0.738    0.028    0.505      0.738    0.600      0.593    0.982     0.508     vgood
                 0.565    0.027    0.464      0.565    0.510      0.490    0.978     0.474     good
Weighted Avg.    0.765    0.202    0.791      0.765    0.773      0.525    0.838     0.803

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 942 189  40  39 |   a = unacc
  84 293   1   6 |   b = acc
  17   0  48   0 |   c = vgood
  24   0   6  39 |   d = good
```

OK

- **For Hamming:**



```
For tuning the KNN value with different neighbour value:

Car Evaluation Dataset for 10-fold cross validation with neighbours as 3:
Scores: [87.20930232558139, 87.20930232558139, 87.20930232558139, 85.46511627906976, 83.72093023255815, 86.04651162790698, 87.7
9069767441861, 87.79069767441861, 87.79069767441861, 81.3953488372093]
ean Accuracy: 86.163%
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 7:
Scores: [86.04651162790698, 88.37209302325581, 85.46511627906976, 84.30232558139535, 84.88372093023256, 89.53488372093024, 81.9
7674418604652, 85.46511627906976, 87.20930232558139, 84.30232558139535]
ean Accuracy: 85.756%
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 9:
Scores: [84.88372093023256, 88.37209302325581, 84.88372093023256, 88.95348837209302, 84.88372093023256, 83.72093023255815, 87.7
9069767441861, 87.20930232558139, 86.04651162790698, 82.55813953488372]
ean Accuracy: 85.930%

Therefore, the prediction accuracy is the best for neighbour value = 7
```



IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.MinkowskiDistance -P 2.0 -R first-last\""

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

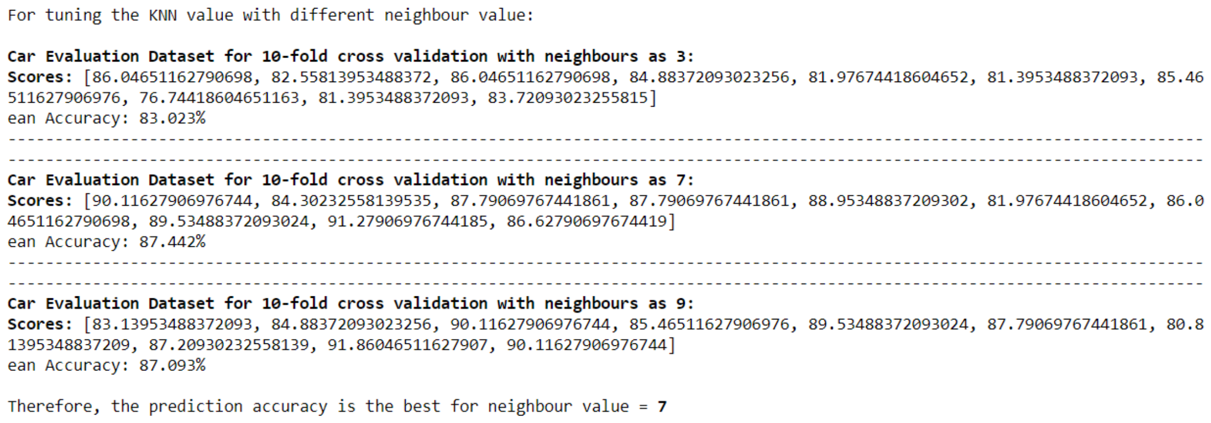
(Nom) Class

19:34:51 - lazy.IBk
19:37:39 - lazy.IBk
19:41:59 - lazy.IBk
19:45:51 - lazy.IBk
20:02:57 - lazy.IBk
20:03:33 - lazy.IBk

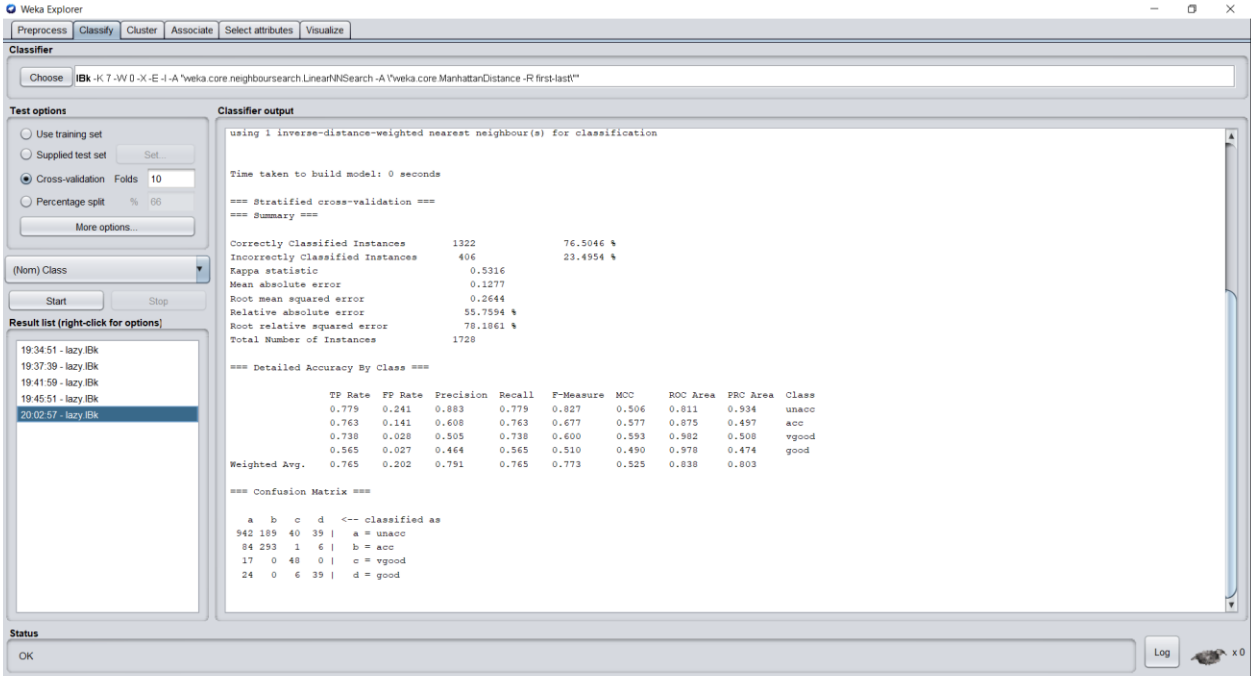
```
using 1 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1322               76.5046 %
Incorrectly Classified Instances       406               23.4954 %
Kappa statistic                          0.5316
Mean absolute error                      0.1277
Root mean squared error                  0.2644
Relative absolute error                 55.7594 %
Root relative squared error             78.1861 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.779    0.241    0.883      0.779    0.827      0.506    0.811     0.934     unacc
                 0.763    0.141    0.608      0.763    0.677      0.577    0.875     0.497     acc
                 0.738    0.028    0.505      0.738    0.600      0.593    0.982     0.508     vgood
                 0.565    0.027    0.464      0.565    0.510      0.490    0.978     0.474     good
Weighted Avg.    0.765    0.202    0.791      0.765    0.773      0.525    0.838     0.803

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 942 189  40  39 |   a = unacc
  84 293   1   6 |   b = acc
  17   0  48   0 |   c = vgood
  24   0   6  39 |   d = good
```

OK

- **For Manhattan:**



```
For tuning the KNN value with different neighbour value:

Car Evaluation Dataset for 10-fold cross validation with neighbours as 3:
Scores: [86.04651162790698, 82.55813953488372, 86.04651162790698, 84.88372093023256, 81.97674418604652, 81.3953488372093, 85.46
511627906976, 76.74418604651163, 81.3953488372093, 83.72093023255815]
ean Accuracy: 83.023%
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 7:
Scores: [90.11627906976744, 84.30232558139535, 87.79069767441861, 87.79069767441861, 88.95348837209302, 81.97674418604652, 86.0
4651162790698, 89.53488372093024, 91.27906976744185, 86.62790697674419]
ean Accuracy: 87.442%
------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------
Car Evaluation Dataset for 10-fold cross validation with neighbours as 9:
Scores: [83.13953488372093, 84.88372093023256, 90.11627906976744, 85.46511627906976, 89.53488372093024, 87.79069767441861, 80.8
1395348837209, 87.20930232558139, 91.86046511627907, 90.11627906976744]
ean Accuracy: 87.093%

Therefore, the prediction accuracy is the best for neighbour value = 7
```



IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.ManhattanDistance -R first-last\""

Folds 10

Percentage split % 66

(Nom) Class

19:34:51 - lazy.IBk
19:37:39 - lazy.IBk
19:41:59 - lazy.IBk
19:45:51 - lazy.IBk
20:02:57 - lazy.IBk

```
using 1 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1322               76.5046 %
Incorrectly Classified Instances       406               23.4954 %
Kappa statistic                          0.5316
Mean absolute error                      0.1277
Root mean squared error                  0.2644
Relative absolute error                 55.7594 %
Root relative squared error             78.1861 %
Total Number of Instances             1728

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.779    0.241    0.883      0.779    0.827      0.506    0.811     0.934     unacc
                 0.763    0.141    0.608      0.763    0.677      0.577    0.875     0.497     acc
                 0.738    0.028    0.505      0.738    0.600      0.593    0.982     0.508     vgood
                 0.565    0.027    0.464      0.565    0.510      0.490    0.978     0.474     good
Weighted Avg.    0.765    0.202    0.791      0.765    0.773      0.525    0.838     0.803

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 942 189  40  39 |   a = unacc
  84 293   1   6 |   b = acc
  17   0  48   0 |   c = vgood
  24   0   6  39 |   d = good
```

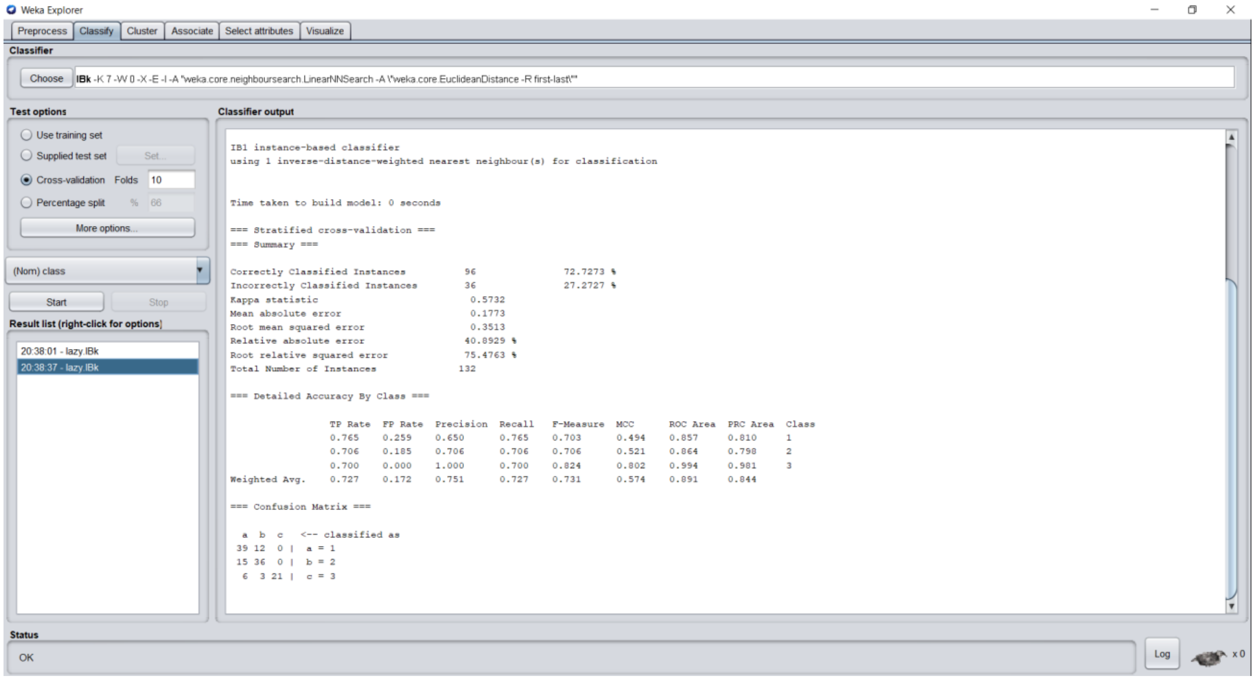
OK

4.3 Hayes- Roth Dataset:

Similar to Car Evaluation Dataset, to increase the accuracy rate of prediction. We will remove the 1st column i.e. **Name** Column from the dataset (via coding).

- ## For Euclidean:

```
For tuning the KNN value with different neighbour value:

Hayes-roth Dataset for 10-fold cross validation with neighbours as 3:
Scores: [84.61538461538461, 46.15384615384615, 84.61538461538461, 69.23076923076923, 69.23076923076923, 69.23076923076923, 61.5
3846153846154, 84.61538461538461, 76.92307692307693, 76.92307692307693]
Mean Accuracy: 72.308%
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 5:
Scores: [69.23076923076923, 46.15384615384615, 53.84615384615385, 46.15384615384615, 84.61538461538461, 61.53846153846154, 76.9
2307692307693, 84.61538461538461, 53.84615384615385, 61.53846153846154]
Mean Accuracy: 63.846%
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 9:
Scores: [46.15384615384615, 69.23076923076923, 76.92307692307693, 69.23076923076923, 69.23076923076923, 69.23076923076923, 61.5
3846153846154, 61.53846153846154, 69.23076923076923, 61.53846153846154]
Mean Accuracy: 65.385%

Therefore, the prediction accuracy is the best for neighbour value = 3
```



IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

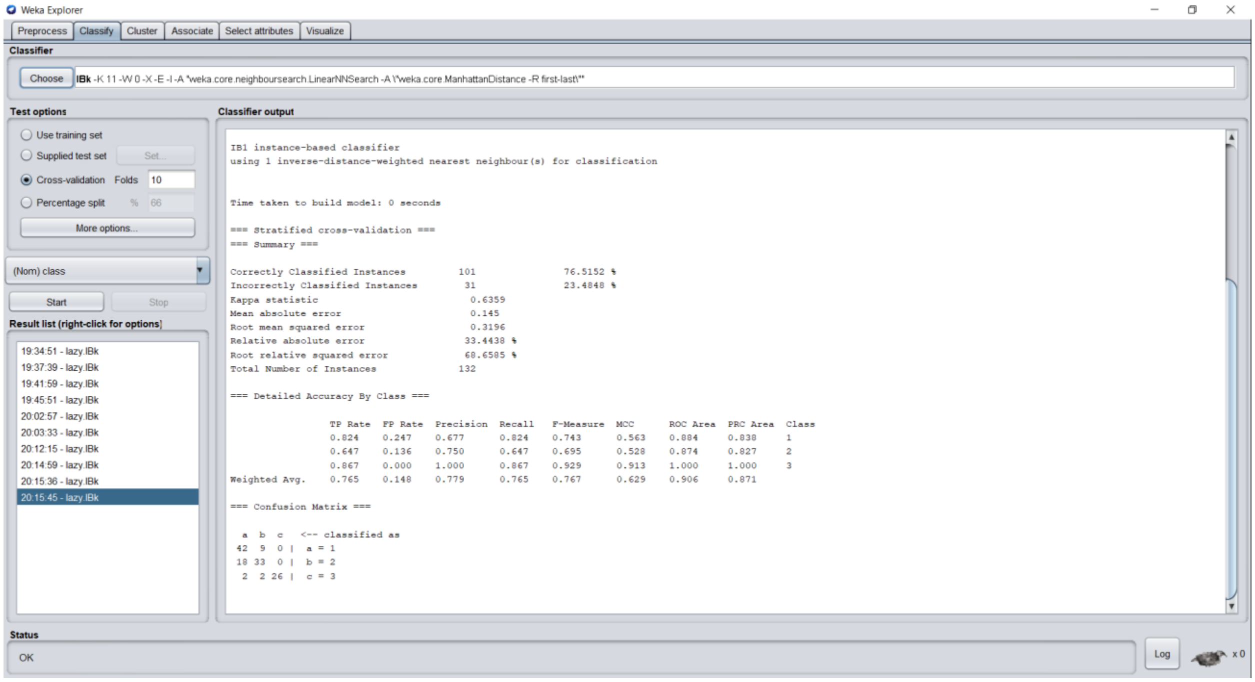
20:38:01 - lazy.IBk
20:38:37 - lazy.IBk

```
IB1 instance-based classifier
using 1 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          96               72.7273 %
Incorrectly Classified Instances        36               27.2727 %
Kappa statistic                          0.5732
Mean absolute error                      0.1773
Root mean squared error                  0.3513
Relative absolute error                 40.8929 %
Root relative squared error             75.4763 %
Total Number of Instances              132

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.765    0.259    0.650      0.765    0.703      0.494    0.857     0.810     1
                 0.706    0.185    0.706      0.706    0.706      0.521    0.864     0.798     2
                 0.700    0.000    1.000      0.700    0.824      0.802    0.994     0.981     3
Weighted Avg.    0.727    0.172    0.751      0.727    0.731      0.574    0.891     0.844

=== Confusion Matrix ===

  a  b  c   <-- classified as
 39 12  0 |  a = 1
 15 36  0 |  b = 2
  6  3 21 |  c = 3
```

OK

- ## For Hamming:

```
For tuning the KNN value with different neighbour value:

Hayes-roth Dataset for 10-fold cross validation with neighbours as 3:
Scores: [61.53846153846154, 53.84615384615385, 53.84615384615385, 76.92307692307693, 69.23076923076923, 69.23076923076923, 46.1
5384615384615, 53.84615384615385, 53.84615384615385, 84.61538461538461]
Mean Accuracy: 62.308%
---------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 5:
Scores: [53.84615384615385, 76.92307692307693, 84.61538461538461, 53.84615384615385, 76.92307692307693, 53.84615384615385, 38.4
6153846153847, 69.23076923076923, 84.61538461538461, 61.53846153846154]
Mean Accuracy: 65.385%
---------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 9:
Scores: [53.84615384615385, 30.76923076923077, 53.84615384615385, 30.76923076923077, 53.84615384615385, 76.92307692307693, 61.5
3846153846154, 53.84615384615385, 38.46153846153847, 53.84615384615385]
Mean Accuracy: 50.769%

Therefore, the prediction accuracy is the best for neighbour value = 3
```



Classifier

IBk -K 11 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.ManhattanDistance -R first-last\""

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) class

Result list (right-click for options)

19:34:51 - lazy.IBk
19:37:39 - lazy.IBk
19:41:59 - lazy.IBk
19:45:51 - lazy.IBk
20:02:57 - lazy.IBk
20:03:33 - lazy.IBk
20:12:15 - lazy.IBk
20:14:59 - lazy.IBk
20:15:36 - lazy.IBk
20:15:45 - lazy.IBk

Classifier output

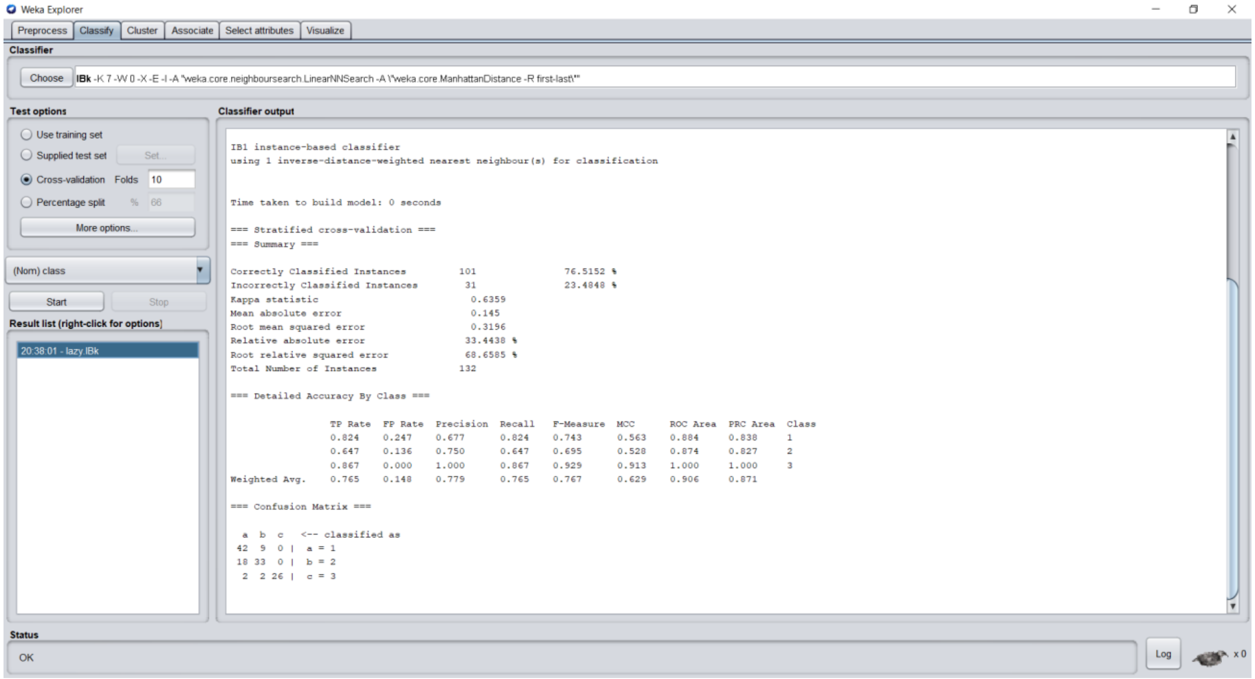
```
IB1 instance-based classifier
using 1 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         101               76.5152 %
Incorrectly Classified Instances        31               23.4848 %
Kappa statistic                          0.6359
Mean absolute error                      0.145
Root mean squared error                  0.3196
Relative absolute error                 33.4438 %
Root relative squared error             68.6585 %
Total Number of Instances              132

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.824    0.247    0.677      0.824    0.743      0.563    0.884     0.838     1
                 0.647    0.136    0.750      0.647    0.695      0.528    0.874     0.827     2
                 0.867    0.000    1.000      0.867    0.929      0.913    1.000     1.000     3
Weighted Avg.    0.765    0.148    0.779      0.765    0.767      0.629    0.906     0.871

=== Confusion Matrix ===

  a  b  c   <-- classified as
 42  9  0 |  a = 1
 18 33  0 |  b = 2
  2  2 26 |  c = 3
```

Status

OK

- ## For Manhattan:

```
For tuning the KNN value with different neighbour value:

Hayes-roth Dataset for 10-fold cross validation with neighbours as 3:
Scores: [69.23076923076923, 76.92307692307693, 84.61538461538461, 76.92307692307693, 61.53846153846154, 30.76923076923077, 92.3
076923076923, 46.15384615384615, 76.92307692307693, 69.23076923076923]
Mean Accuracy: 68.462%
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 5:
Scores: [76.92307692307693, 84.61538461538461, 53.84615384615385, 69.23076923076923, 92.3076923076923, 92.3076923076923, 61.538
46153846154, 84.61538461538461, 53.84615384615385, 61.53846153846154]
Mean Accuracy: 73.077%
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
Hayes-roth Dataset for 10-fold cross validation with neighbours as 9:
Scores: [53.84615384615385, 76.92307692307693, 46.15384615384615, 69.23076923076923, 53.84615384615385, 53.84615384615385, 69.2
3076923076923, 69.23076923076923, 100.0, 76.92307692307693]
Mean Accuracy: 66.923%

Therefore, the prediction accuracy is the best for neighbour value = 3
```



IBk -K 7 -W 0 -X -E -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.ManhattanDistance -R first-last\""

● Cross-validation Folds 10

(Nom) class

20:38:01 - lazy.IBk

OK

Classifier output

```
IB1 instance-based classifier
using 1 inverse-distance-weighted nearest neighbour(s) for classification


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         101               76.5152 %
Incorrectly Classified Instances        31               23.4848 %
Kappa statistic                          0.6359
Mean absolute error                      0.145
Root mean squared error                  0.3196
Relative absolute error                 33.4438 %
Root relative squared error             68.6585 %
Total Number of Instances              132

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.824    0.247    0.677      0.824    0.743      0.563    0.884     0.838     1
                 0.647    0.136    0.750      0.647    0.695      0.528    0.874     0.827     2
                 0.867    0.000    1.000      0.867    0.929      0.913    1.000     1.000     3
Weighted Avg.    0.765    0.148    0.779      0.765    0.767      0.629    0.906     0.871

=== Confusion Matrix ===

  a  b  c   <-- classified as
 42  9  0 |  a = 1
 18 33  0 |  b = 2
  2  2 26 |  c = 3
```

Hence the table for the above comparison:
We will be only comparing the best accuracy **Mean Score** of the program to the WEKA.

Dataset	**Distance Metrices**	**Program (%)**	**Weka (%)**
Breast Cancer	Euclidean	77.857	78.671
Breast Cancer	Hamming	76.071	77.972
Breast Cancer	Manhattan	75.714	77.273
Car Evaluation	Euclidean	87.5	76.504
Car Evaluation	Hamming	86.163	76.518
Car Evaluation	Manhattan	87.442	76.106
Hayes-Roth	Euclidean	72.308	72.727
Hayes-Roth	Hamming	65.385	76.515
Hayes-Roth	Manhattan	73.077	75.512

5 References

- To get basic knowledge Regarding KNN Algorithm: **Reference1** **&** **Reference2**
- Used Professor's **link** to develop code.
- To know more about **K-fold cross Validation.**
- For **Formula** & Knowledge regarding different types of Distance Metrics.

6 Conclusion

We had developed the KNN algorithm from scratch and utilized four different datasets to predict the classification.
We did with different attributes and compared the best results with WEKA.