**Code Results:**

Once the code starts running it will call the evaluate function, which will further call k-fold where cross validation will be done. Once that is done it runs the K nearest algorithm, Distances are calculated, and predictions are made.

We use 4 distance measures to calculate:

* Euclidian Distance
* Hamming Distance
* Minkowski Distance
* Manhattan Distance.

1. **Car dataset**:

Dataset is used from : <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

In the code when the user enters ‘1’ , user selects car data set.

**The Output of code**:

A screenshot of a social media post

Description automatically generated

**Output of Weka:**

=== Run information ===

Scheme: weka.classifiers.lazy.IBk -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: car-weka.filters.unsupervised.attribute.StringToNominal-R3-4-weka.filters.unsupervised.attribute.StringToNominal-R3-4

Instances: 1728

Attributes: 7

1

2

3

4

5

6

7

Test mode: 10-fold cross-validation

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

IB1 instance-based classifier

using 5 nearest neighbour(s) for classification

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 1616 93.5185 %

Incorrectly Classified Instances 112 6.4815 %

Kappa statistic 0.853

Mean absolute error 0.1122

Root mean squared error 0.1953

Relative absolute error 48.9977 %

Root relative squared error 57.7645 %

Total Number of Instances 1728

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.998 0.066 0.973 0.998 0.985 0.949 1.000 1.000 unacc

0.911 0.058 0.818 0.911 0.862 0.822 0.988 0.958 acc

0.708 0.000 1.000 0.708 0.829 0.836 1.000 1.000 vgood

0.188 0.000 1.000 0.188 0.317 0.427 0.994 0.859 good

Weighted Avg. 0.935 0.059 0.940 0.935 0.925 0.896 0.997 0.985

=== Confusion Matrix ===

a b c d <-- classified as

1207 3 0 0 | a = unacc

34 350 0 0 | b = acc

0 19 46 0 | c = vgood

0 56 0 13 | d = good

A screenshot of a computer

Description automatically generated

**Accuracy from code: 90.116 %**

**Accuracy from weka: 93.5185 %**

1. **Hayes-Roth dataset**:

Dataset is used from : <https://archive.ics.uci.edu/ml/datasets/Hayes-Roth>

In the code when the user enters ‘2’, user selects hayes roth data set.

**Output of code:**

A screenshot of a social media post

Description automatically generated

**Output from weka:**

=== Run information ===

Scheme: weka.classifiers.lazy.IBk -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: hayes-roth-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last

Instances: 132

Attributes: 6

1

2

3

4

5

6

Test mode: 10-fold cross-validation

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

IB1 instance-based classifier

using 5 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 84 63.6364 %

Incorrectly Classified Instances 48 36.3636 %

Kappa statistic 0.4143

Mean absolute error 0.3501

Root mean squared error 0.4079

Relative absolute error 80.7647 %

Root relative squared error 87.6388 %

Total Number of Instances 132

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.843 0.370 0.589 0.843 0.694 0.463 0.840 0.748 1

0.686 0.222 0.660 0.686 0.673 0.461 0.850 0.779 2

0.200 0.000 1.000 0.200 0.333 0.402 0.988 0.945 3

Weighted Avg. 0.636 0.229 0.710 0.636 0.604 0.448 0.877 0.805

=== Confusion Matrix ===

a b c <-- classified as

43 8 0 | a = 1

16 35 0 | b = 2

14 10 6 | c = 3

**A screenshot of a social media post

Description automatically generated**

**Accuracy from code: 45.385 %**

**Accuracy from weka: 63.6364 %**

1. **Breast-Cancer Dataset**

Dataset is used from : <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer>

In the code when the user enters ‘3’, user selects Breast Cancer data set.

**Output of code**:

A screenshot of a social media post

Description automatically generated

**Output from Weka:**

=== Run information ===

Scheme: weka.classifiers.lazy.IBk -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: breast-cancer

Instances: 286

Attributes: 10

1

2

3

4

5

6

7

8

9

10

Test mode: 10-fold cross-validation

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

IB1 instance-based classifier

using 5 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 220 76.9231 %

Incorrectly Classified Instances 66 23.0769 %

Kappa statistic 0.181

Mean absolute error 0.2975

Root mean squared error 0.4158

Relative absolute error 81.8349 %

Root relative squared error 97.6636 %

Total Number of Instances 286

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.950 0.809 0.790 0.950 0.863 0.216 0.680 0.867 no

0.191 0.050 0.542 0.191 0.283 0.216 0.680 0.427 yes

Weighted Avg. 0.769 0.629 0.731 0.769 0.725 0.216 0.680 0.762

=== Confusion Matrix ===

a b <-- classified as

207 11 | a = no

55 13 | b = yes

A screenshot of a social media post

Description automatically generated

**Accuracy from code: 76.429 %**

**Accuracy from weka: 76.923 %**

1. **Irish Dataset**

Dataset is used from: <https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv>

In the code when the user enters ‘4’, user selects Irish data set.

**Output of Code:**

A screenshot of a social media post

Description automatically generated

**Accuracy from code: 96.667 %**

**References**

1. <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>
2. <https://machinelearningmastery.com/k-fold-cross-validation/>
3. <https://ljvmiranda921.github.io/notebook/2017/02/09/k-nearest-neighbors/>
4. [https://machinelearningmastery.com/distance-measures-for-machine-learning/#:~:text=of%20Distance%20Measures-,Distance%20measures%20play%20an%20important%20role%20in%20machine%20learning.,objects%20in%20a%20problem%20domain.&text=Another%20unsupervised%20l](https://machinelearningmastery.com/distance-measures-for-machine-learning/#:~:text=of%20Distance%20Measures-,Distance%20measures%20play%20an%20important%20role%20in%20machine%20learning.,objects%20in%20a%20problem%20domain.&text=Another%20unsupervised%20learning%20algorithm%20that,the%20K%2Dmeans%20clustering%20algorithm)
5. <https://www.programiz.com/python-programming/methods/list/remove>

**Extensions in Code**

1. True KNN: Tried Larger and larger value of k (number of neighbors).
2. Performed Vectorization of dataset.
3. Implement different distance measure*.*