

Experiment:-6

- Objective :- Implementation of KNN algorithm using weka explorer.
- Theory :- KNN algorithm is supervised learning algorithm that uses the idea that similar data points are close together.
- Dataset :- Iris dataset.
- Steps :-
 - ① Load the data
 - ② Select the IBk algo (Classify → choose → lazy → IBk)
 - ③ Click the algo name to open the object window editor.
 - ④ Adjust parameters
 - ⑤ Click start to see result.

- Result :- We have successfully implemented the KNN algo in weka achieving accurate classification result.

Viva-question:-

Q1. What does k in KNN represent? How does the value of k is selected w.r to any dataset?

→ k represent no. of nearest neighbours consider the classification.

EXPERIMENT 6

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose | IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) class

Start Stop

Result list (right-click for options)

09:22:50 - lazy.IBk

Classifier output

```
=== Run information ===

Scheme:      weka.classifiers.lazy.IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     iris
Instances:    150
Attributes:    5
  sepallength
  sepalwidth
  petallength
  petalwidth
  class
Test mode:    10-fold cross-validation

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

IB1 instance-based classifier
using 3 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

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

Correctly Classified Instances      143           95.3333 %
Incorrectly Classified Instances      7           4.6667 %
Kappa statistic                    0.93
Mean absolute error                  0.04
Root mean squared error              0.1703
Relative absolute error              5.0013 %
Root relative squared error          36.1192 %
Total Number of Instances           150

=== Detailed Accuracy By Class ===
=== Summary ===

Correctly Classified Instances      143           95.3333 %
Incorrectly Classified Instances      7           4.6667 %
Kappa statistic                    0.93
Mean absolute error                  0.04
Root mean squared error              0.1703
Relative absolute error              5.0013 %
Root relative squared error          36.1192 %
Total Number of Instances           150

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCG     ROC Area  FRC Area  Class
      1.000    0.000    1.000     1.000    1.000     1.000    1.000     1.000    Iris-setosa
      0.940    0.040    0.922     0.940    0.931     0.896    0.963     0.910    Iris-versicolor
      0.920    0.030    0.939     0.920    0.929     0.895    0.958     0.919    Iris-virginica
Weighted Avg.   0.953    0.023    0.953     0.953    0.953     0.930    0.974     0.943

=== Confusion Matrix ===

 a  b  c  <-- classified as
50  0  0 | a = Iris-setosa
 0 47  3 | b = Iris-versicolor
 0  4 46 | c = Iris-virginica
```

Status
OK

Log x 0

Value of k is typically selected through cross validation, with a common strategy being to choose an old value to avoid ties and test various values to find optimal balance b/w bias and variance.

Q2 Discuss & enlist limitations, of KNN implementation using mika explorer

- Limitations :-
- ① Computationally expensive
 - ② Sensitive to irrelevant features.
 - ③ No model training
 - ④ Curse of dimensionality.