PARSHWANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering
Data Science

Academic Year: 2025-26 **Semester: V**
Class/Branch: T.E. DS **Subject: DWMLab**

EXPERIMENT NO. 8

1. **Aim:** To apply clustering algorithms on a given dataset using WEKA to discover natural groupings within the data.

2. **Objectives:** From this experiment, the student will be able to

   - To implement Simple K-Means Clustering using Weka.
   - To implement the Hierarchical Clustering using Weka.

3. **Theory:**

Clustering using WEKA:

Clustering in WEKA is a technique used to group similar data instances based on their attributes, without using predefined class labels. It falls under unsupervised learning, where the goal is to discover hidden patterns or natural groupings within the dataset. WEKA provides an easy-to-use graphical interface where users can load datasets, select clustering algorithms, and visualize the results to interpret how data points are organized.

Simple K-Means is one of the most commonly used clustering algorithms in WEKA. It partitions the dataset into a specified number of clusters (K) by minimizing the distance between data points and their cluster centers. The algorithm iteratively adjusts centroids until the clusters become stable. It is fast and suitable for large datasets with clearly separated groups.

Hierarchical Clustering builds a hierarchy of clusters using either an agglomerative (bottom-up) or divisive (top-down) approach. It produces a dendrogram in WEKA, which shows how clusters merge or split at different similarity levels. This method does not require specifying the number of clusters beforehand and helps explore relationships among data points.

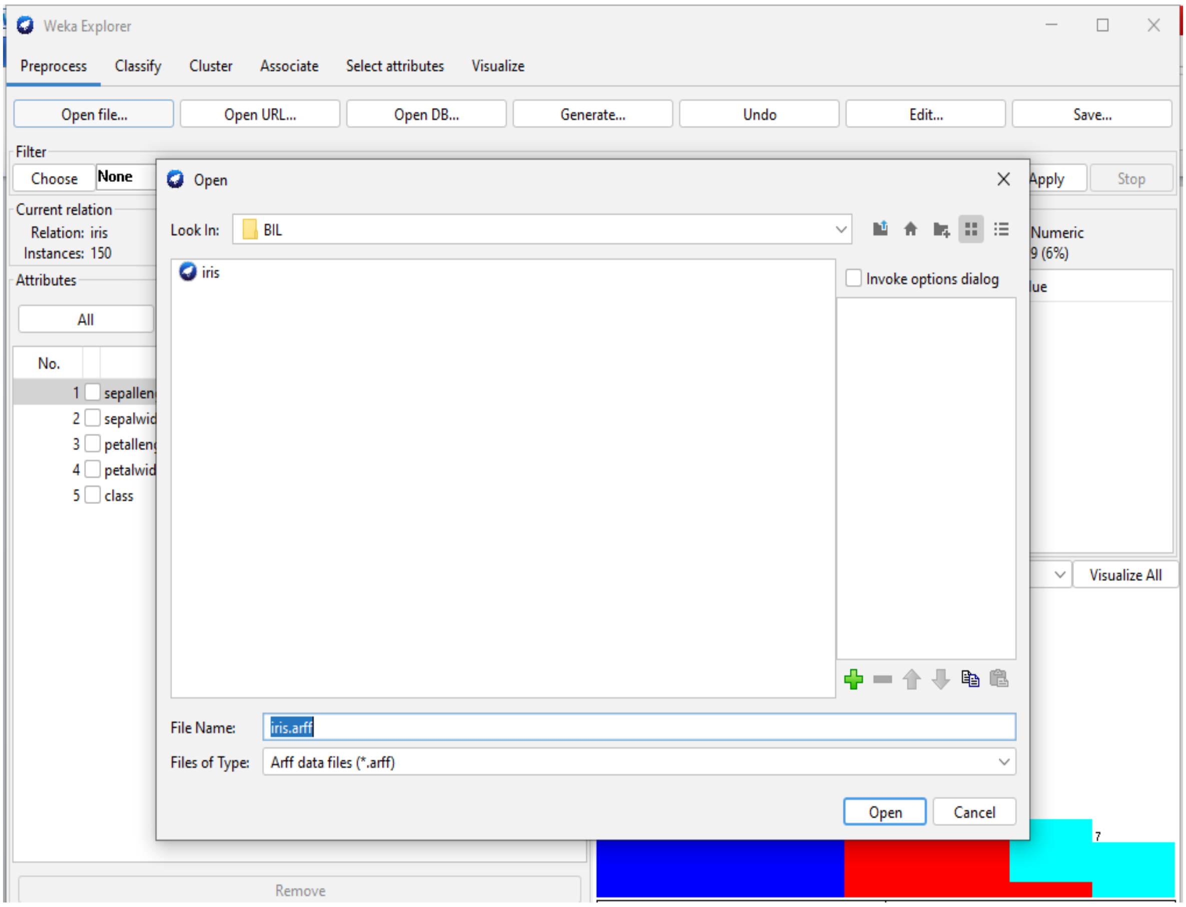
Steps for Performing Clustering in WEKA (Common for K-Means and Hierarchical)

1. Download the Iris dataset and load it into WEKA.
2. Click on the Cluster tab in the Explorer interface.
3. Click on Choose, then select either Simple K-Means (for K-Means) or HierarchicalClusterer (for Hierarchical).
4. Right-click on the selected algorithm in the choose box to open the Properties window.

5. Set the Number of clusters = 3 (or as required).
6. Select the -use training set option.
7. Click Start to run the clustering process.
8. To visualize the results, right-click on the algorithm name in the result list → choose View Cluster Assignments.

The process for k-means is as shown below. The same process should be followed for Hierarchical Clustering.

1. Download Iris dataset

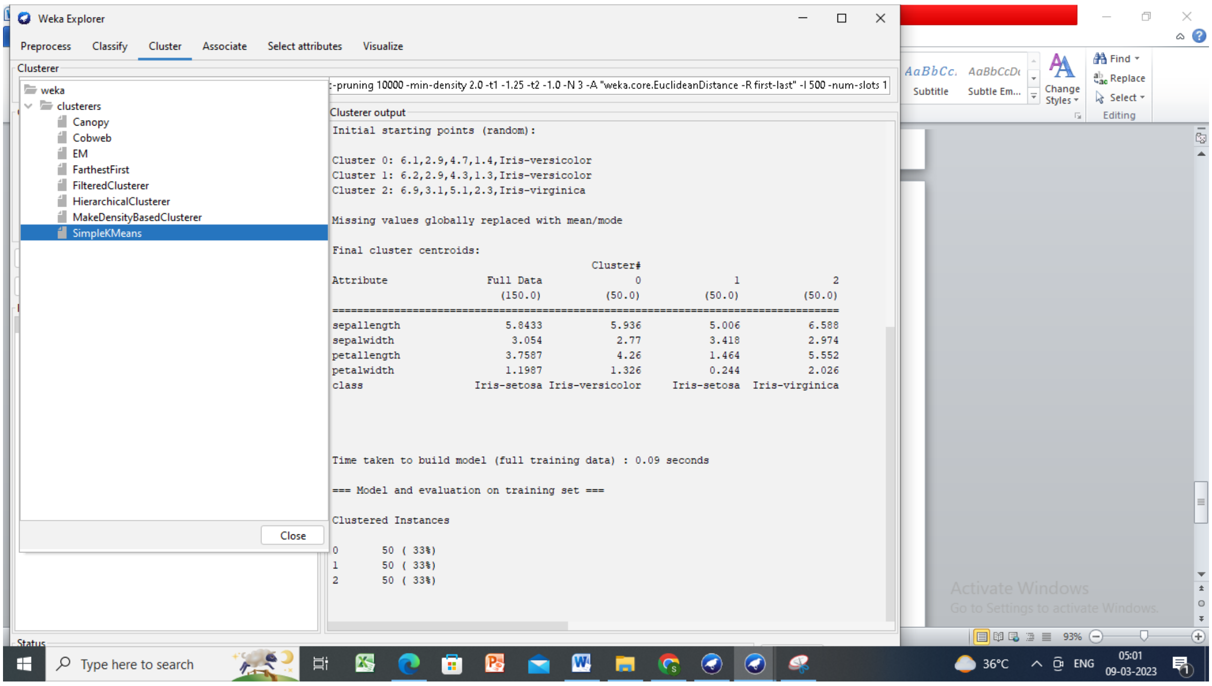


Look In: BIL

File Name: iris.arff

Files of Type: Arff data files (*.arff)

2. Click on cluster then on choose

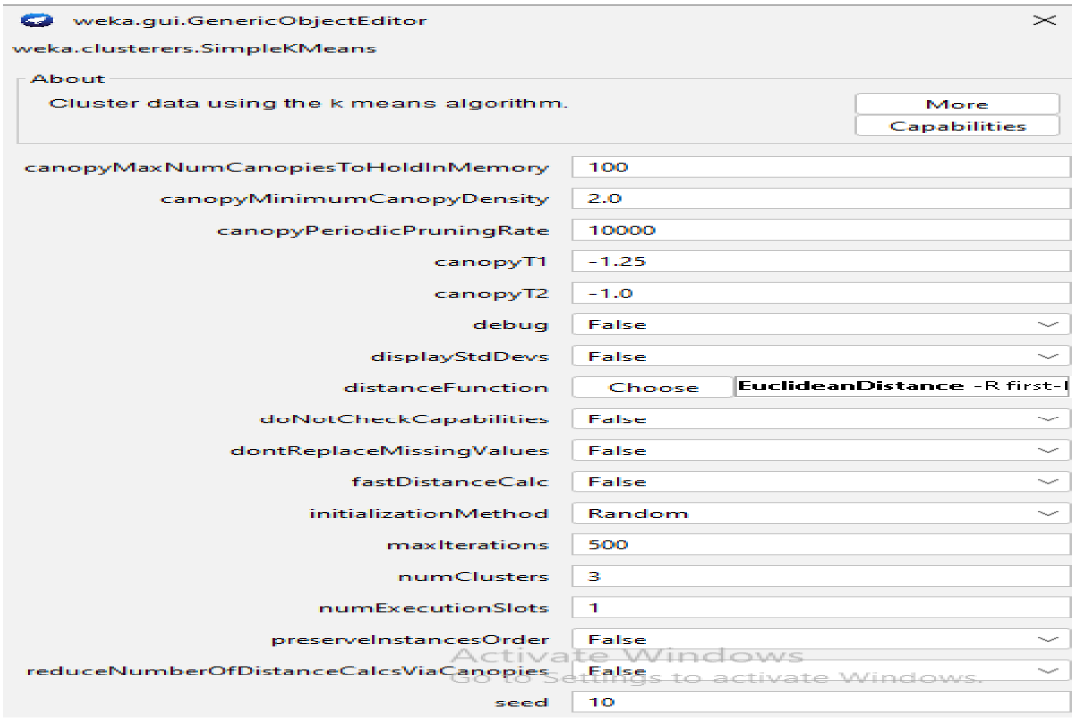


:-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1

Clusterer output

```
Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:
                                   Cluster#
Attribute               Full Data         0            1            2
                          (150.0)    (50.0)       (50.0)       (50.0)
=====================================================================
sepallength                5.8433     5.936        5.006        6.588
sepalwidth                  3.054      2.77        3.418        2.974
petallength                3.7587      4.26        1.464        5.552
petalwidth                 1.1987     1.326        0.244        2.026
class                 Iris-setosa Iris-versicolor Iris-setosa Iris-virginica



Time taken to build model (full training data) : 0.09 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       50 ( 33%)
1       50 ( 33%)
2       50 ( 33%)
```

3. Right clik on simple k means in choose text box to open these properties.
4. Make changes Number of clusters=3



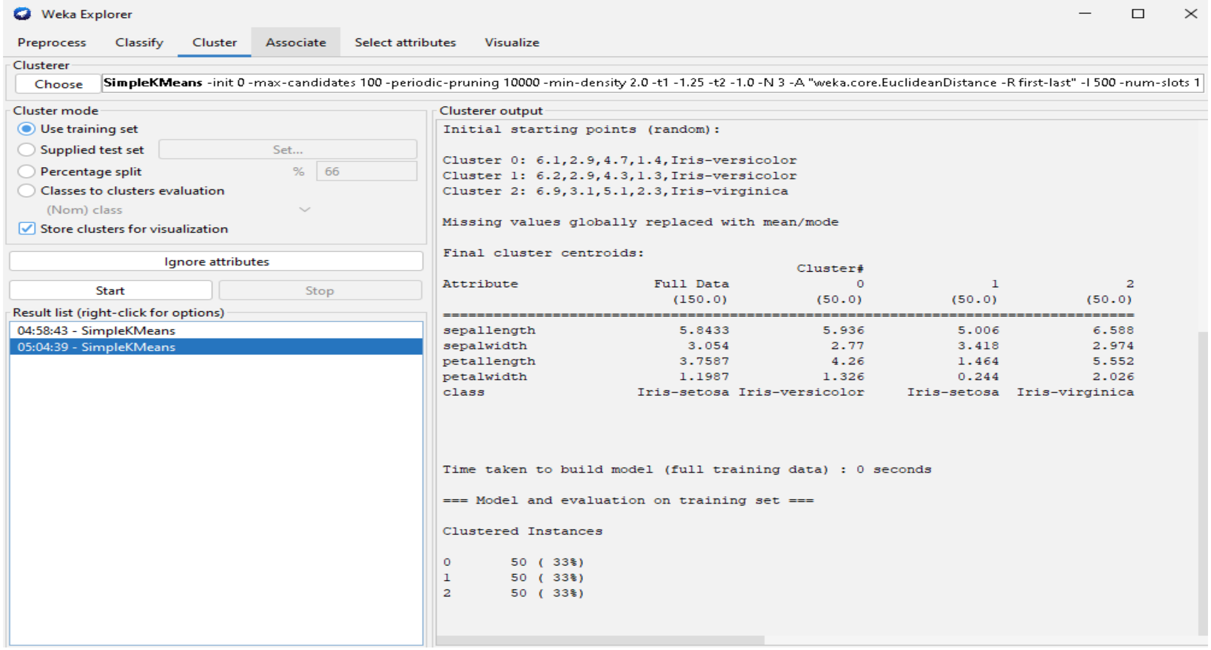
weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm.

canopyMaxNumCanopiesToHoldInMemory	100
canopyMinimumCanopyDensity	2.0
canopyPeriodicPruningRate	10000
canopyT1	-1.25
canopyT2	-1.0
debug	False
displayStdDevs	False
distanceFunction	EuclideanDistance -R first-l
doNotCheckCapabilities	False
dontReplaceMissingValues	False
fastDistanceCalc	False
initializationMethod	Random
maxIterations	500
numClusters	3
numExecutionSlots	1
preserveInstancesOrder	False
reduceNumberOfDistanceCalcsViaCanopies	False
seed	10

5. **Select use training set and then start**



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer: Choose **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1

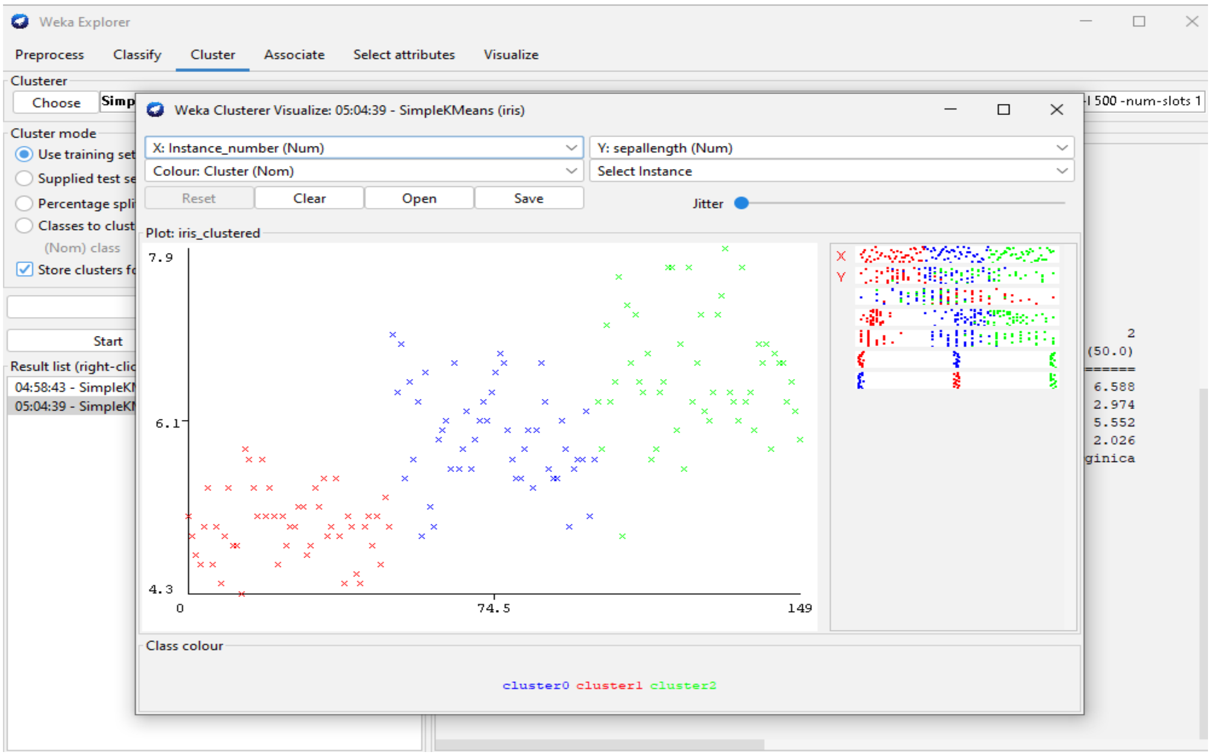
Cluster mode
- ● Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ○ Classes to clusters evaluation (Nom) class
- ☑ Store clusters for visualization

Result list (right-click for options)
- 04:58:43 - SimpleKMeans
- 05:04:39 - SimpleKMeans

Clusterer output

```
Initial starting points (random):

Cluster 0: 6.1,2.9,4.7,1.4,Iris-versicolor
Cluster 1: 6.2,2.9,4.3,1.3,Iris-versicolor
Cluster 2: 6.9,3.1,5.1,2.3,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute                  Full Data           0           1           2
                             (150.0)      (50.0)      (50.0)      (50.0)
============================================================================
sepallength                   5.8433       5.936       5.006       6.588
sepalwidth                     3.054        2.77       3.418       2.974
petallength                   3.7587        4.26       1.464       5.552
petalwidth                    1.1987       1.326       0.244       2.026
class                    Iris-setosa Iris-versicolor Iris-setosa Iris-virginica




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       50 ( 33%)
1       50 ( 33%)
2       50 ( 33%)
```

6. **To visualise right click on simplekmeans in result list →view cluster assignments**



Weka Clusterer Visualize: 05:04:39 - SimpleKMeans (iris)

X: Instance_number (Num)

Y: sepallength (Num)

Colour: Cluster (Nom)

Select Instance

Plot: iris_clustered

Class colour: cluster0 cluster1 cluster2

7. **Conclusion:**