Exercise 5 :

Comparison of Clustering Algorithms Using WEKA Tool:

Aim:

The objective of this lab experiment is to compare the results of K-Means, COBWEB, Canopy, and Hierarchical clustering algorithms using the Iris dataset. The comparison will be based on the following factors: Number of clusters.,Clusters instances,Time taken to build the model.

Algorithm:

1. Launch WEKA and Select Explorer.

2. Load the Iris dataset.

For each clustering algorithm (K-Means, COBWEB, Canopy, Hierarchical):

a. Select the algorithm under the "Cluster" tab.

b. Configure the algorithm parameters:

i. **K-Means:**

- numClusters: 3
- distanceFunction: EuclideanDistance
- initializationMethod: kMeansPlusPlus
- maxIterations: 100
- fastDistanceCalc: true
- seed: Any integer

ii. **COBWEB:**

- Acuity and Cutoff: Adjust if needed or leave default

iii. **Canopy:**

- T1: 1.0
- T2: 0.5
- numClusters: 3
- seed: Any integer

iv. **Hierarchical:**

- linkType: Ward
- distanceFunction: Euclidean

- numClusters: 3

c. Execute the algorithm by clicking "Start".

d. Record execution time, number of clusters, and instance distribution.

3. Analyze Data:

- Use external tools like Excel or Python for visualization:

a. Bar graphs for number of clusters.

b. Pie charts/stacked bar graphs for instance distribution.

c. Line graphs/bar graphs for execution times.

- Compare algorithm performance, time efficiency, and clustering effectiveness.

4. Conclude:

- Summarize findings, compare strengths and weaknesses of each algorithm.

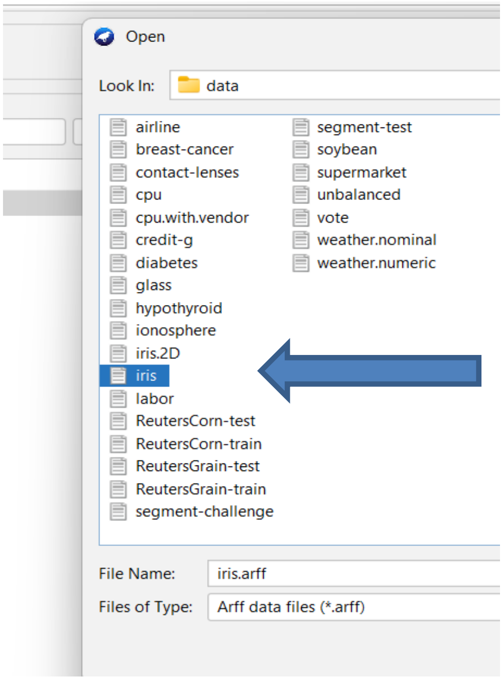
- Provide recommendations based on analysis., time efficiency, and clustering effectiveness.

Implementation:

Data Preparation:

Launch WEKA: Open the WEKA GUI Chooser by clicking on the WEKA icon.

Load Dataset: Go to the "Explorer" -> click "Open file..." and navigate to the Iris dataset. The Iris dataset is typically included in WEKA's default datasets.



Open

Look In: data

airline
breast-cancer
contact-lenses
cpu
cpu.with.vendor
credit-g
diabetes
glass
hypothyroid
ionosphere
iris.2D
iris
labor
ReutersCorn-test
ReutersCorn-train
ReutersGrain-test
ReutersGrain-train
segment-challenge
segment-test
soybean
supermarket
unbalanced
vote
weather.nominal
weather.numeric

File Name: iris.arff

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

Selection of Clustering Algorithms :

a) K-Means

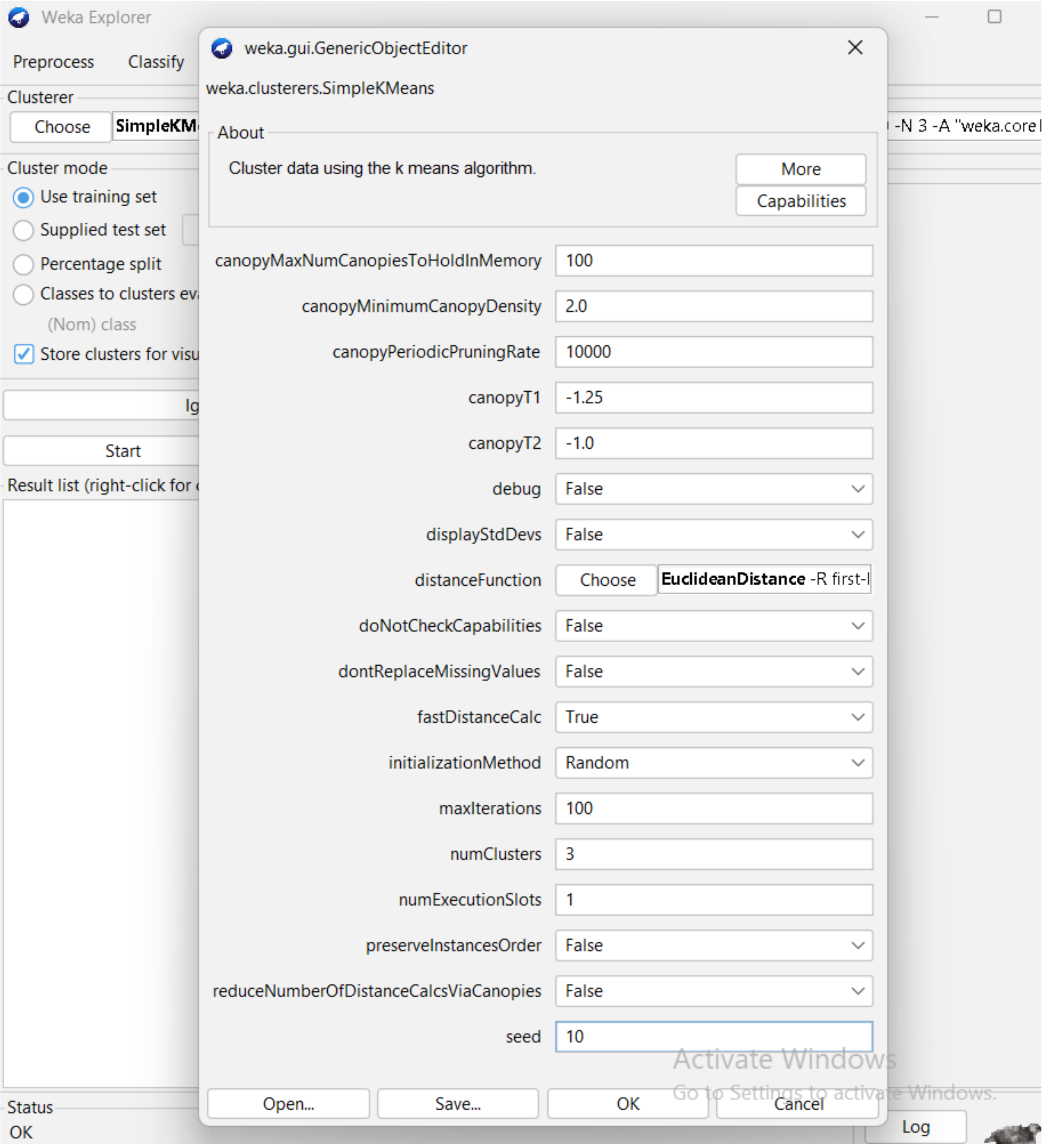
In the Explorer, go to the "Cluster" tab.

Click the "Choose" button, select "SimpleKMeans".

Click on the SimpleKMeans algorithm name to open its options.

Set numClusters to 3, distanceFunction to EuclideanDistance, initializationMethod to kMeansPlusPlus, maxIterations to 100, fastDistanceCalc to true, and seed to any integer.

Other parameters are set as default unless specified otherwise.

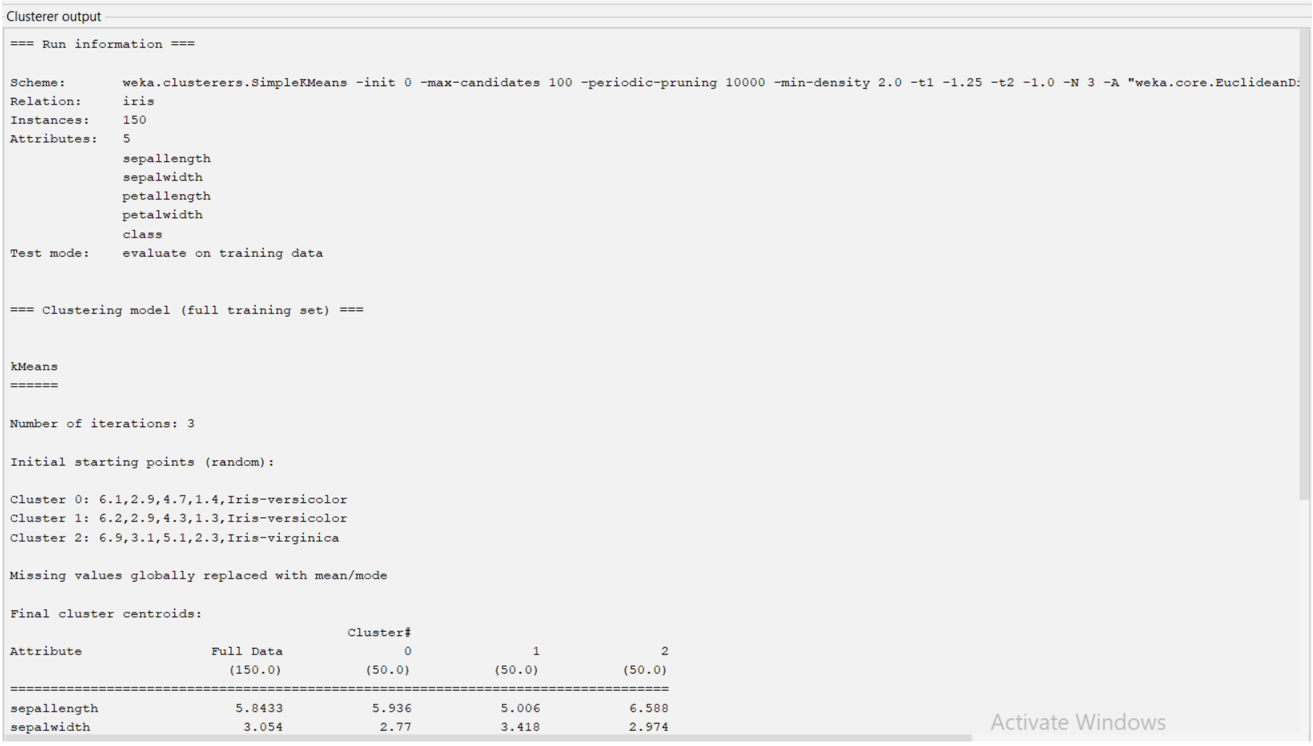


weka.clusterers.SimpleKMeans

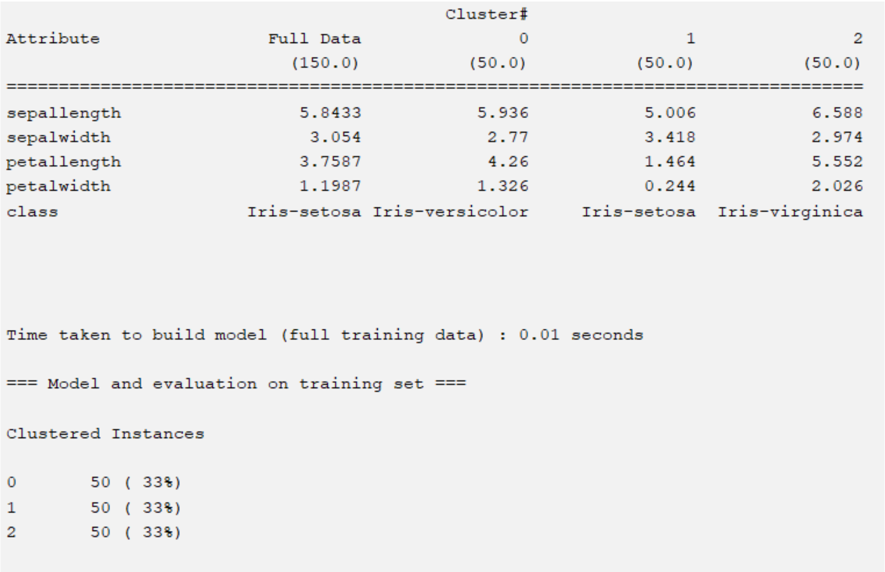
About

Cluster data using the k means algorithm.

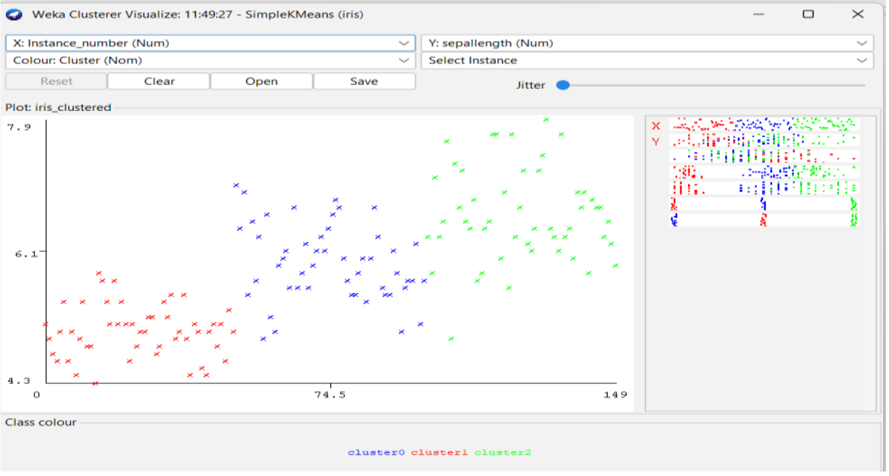
canopyMaxNumCanopiesToHoldInMemory	100
canopyMinimumCanopyDensity	2.0
canopyPeriodicPruningRate	10000
canopyT1	-1.25
canopyT2	-1.0
debug	False
displayStdDevs	False
distanceFunction	Choose EuclideanDistance -R first-l
doNotCheckCapabilities	False
dontReplaceMissingValues	False
fastDistanceCalc	True
initializationMethod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



```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanD:
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===


kMeans
======

Number of iterations: 3

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
```



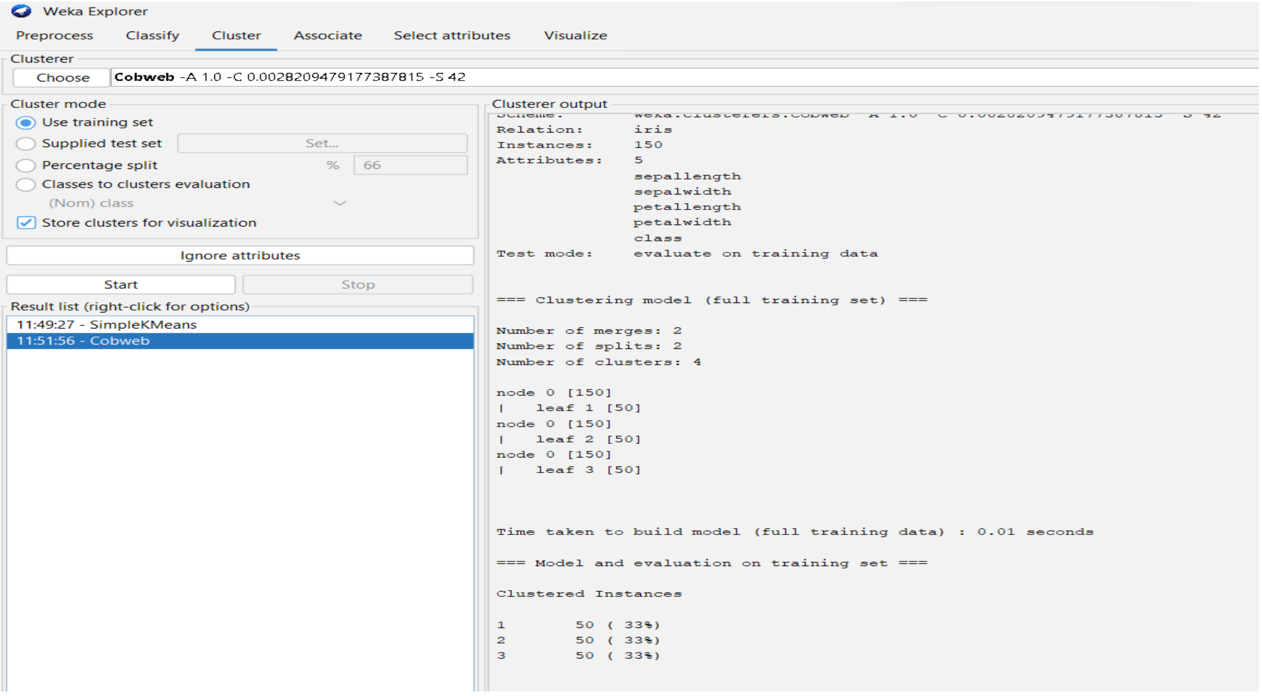
```
                                          Cluster#
Attribute                Full Data               0               1               2
                           (150.0)          (50.0)          (50.0)          (50.0)
==================================================================================
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) : 0.01 seconds

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

Clustered Instances

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



b) COBWEB

Click "Choose", select "Cobweb".

Leave parameters as default or adjust Acuity and Cutoff as needed.

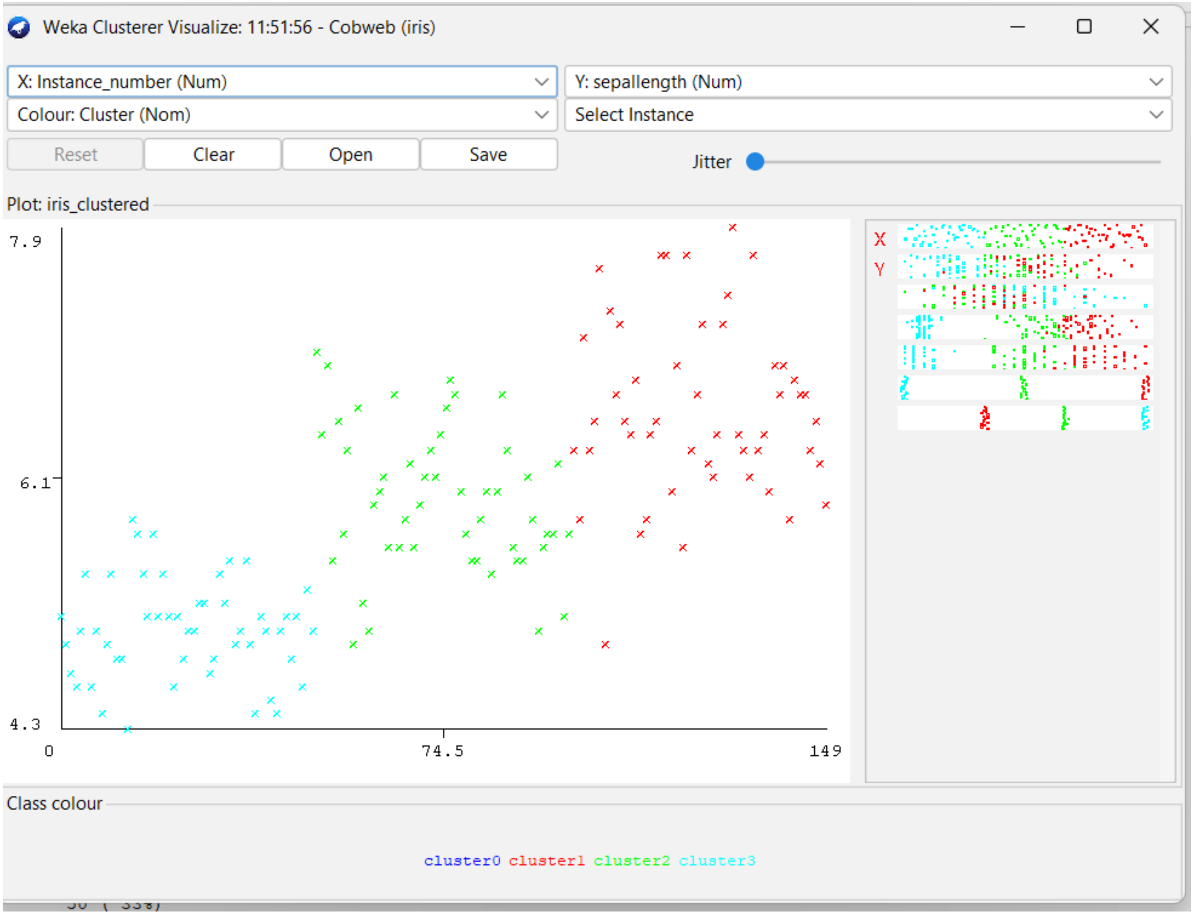


Cobweb -A 1.0 -C 0.0028209479177387815 -S 42

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

11:49:27 - SimpleKMeans
11:51:56 - Cobweb

```
Scheme:       weka.clusterers.Cobweb -A 1.0 -C 0.0028209479177387815 -S 42
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===

Number of merges: 2
Number of splits: 2
Number of clusters: 4

node 0 [150]
|   leaf 1 [50]
node 0 [150]
|   leaf 2 [50]
node 0 [150]
|   leaf 3 [50]



Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

1       50 ( 33%)
2       50 ( 33%)
3       50 ( 33%)
```


Weka Clusterer Visualize: 11:51:56 - Cobweb (iris)
Reset
Clear
Open
Save
Jitter
Plot: iris_clustered
7.9
6.1
4.3
0
74.5
149
X
Y
Class colour
cluster0 cluster1 cluster2 cluster3

X: Instance_number (Num)

Y: sepallength (Num)

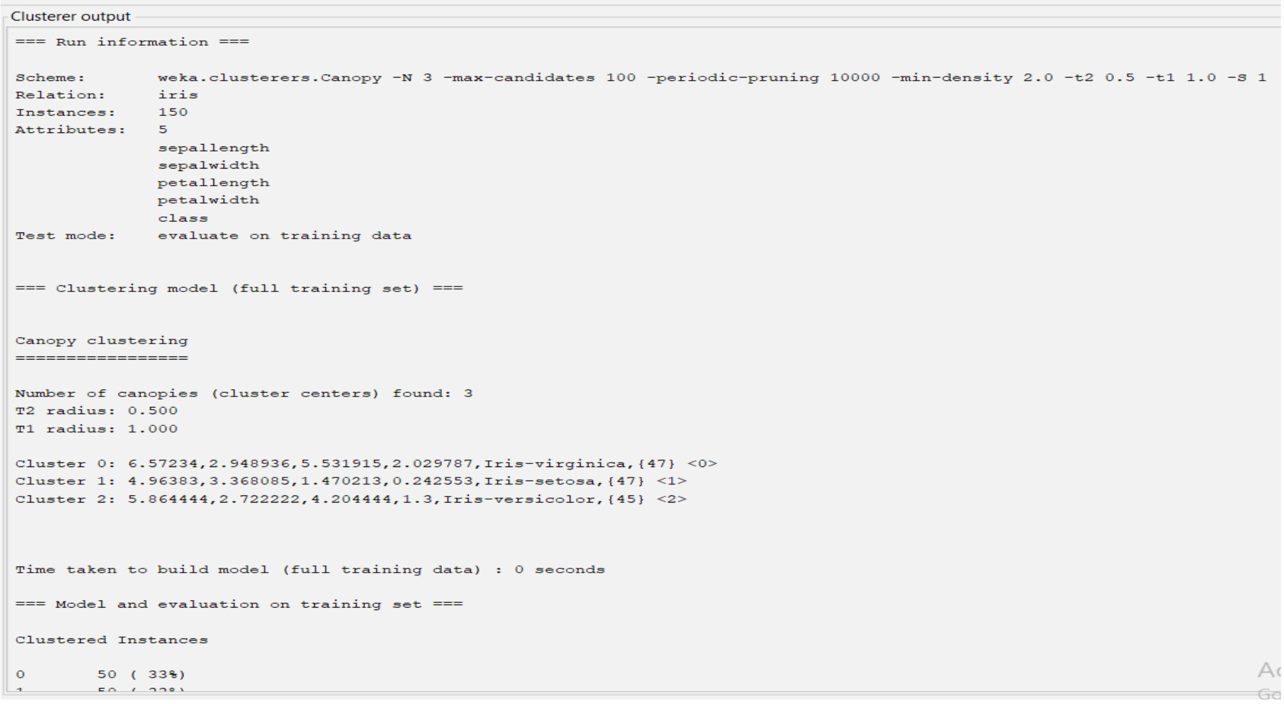
Colour: Cluster (Nom)

Select Instance

c) Canopy

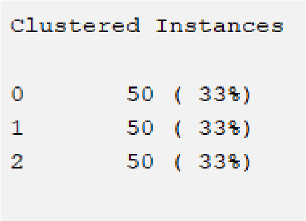
Click "Choose", select "Canopy".

Set T1 to 1.0, T2 to 0.5, seed to any integer, numClusters to 3, and other parameters as specified.

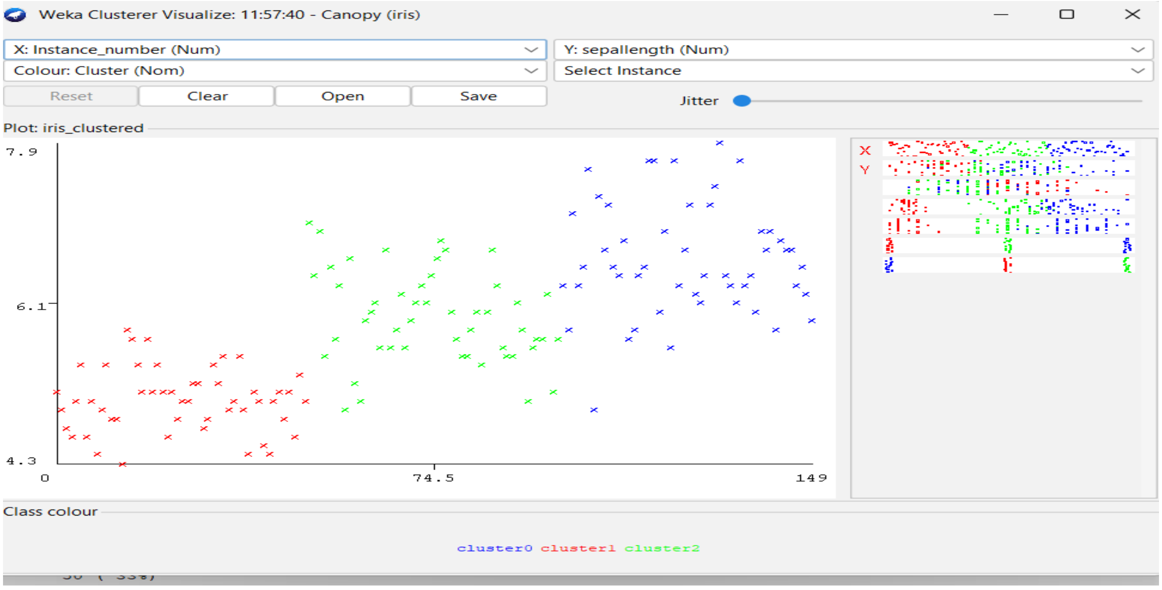


Clusterer output

```
=== Run information ===

Scheme:       weka.clusterers.Canopy -N 3 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t2 0.5 -t1 1.0 -S 1
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===


Canopy clustering
=================

Number of canopies (cluster centers) found: 3
T2 radius: 0.500
T1 radius: 1.000

Cluster 0: 6.57234,2.948936,5.531915,2.029787,Iris-virginica,{47} <0>
Cluster 1: 4.96383,3.368085,1.470213,0.242553,Iris-setosa,{47} <1>
Cluster 2: 5.864444,2.722222,4.204444,1.3,Iris-versicolor,{45} <2>


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

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

Clustered Instances

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



```
Clustered Instances

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



Weka Clusterer Visualize: 11:57:40 - Canopy (iris)

X: Instance_number (Num)

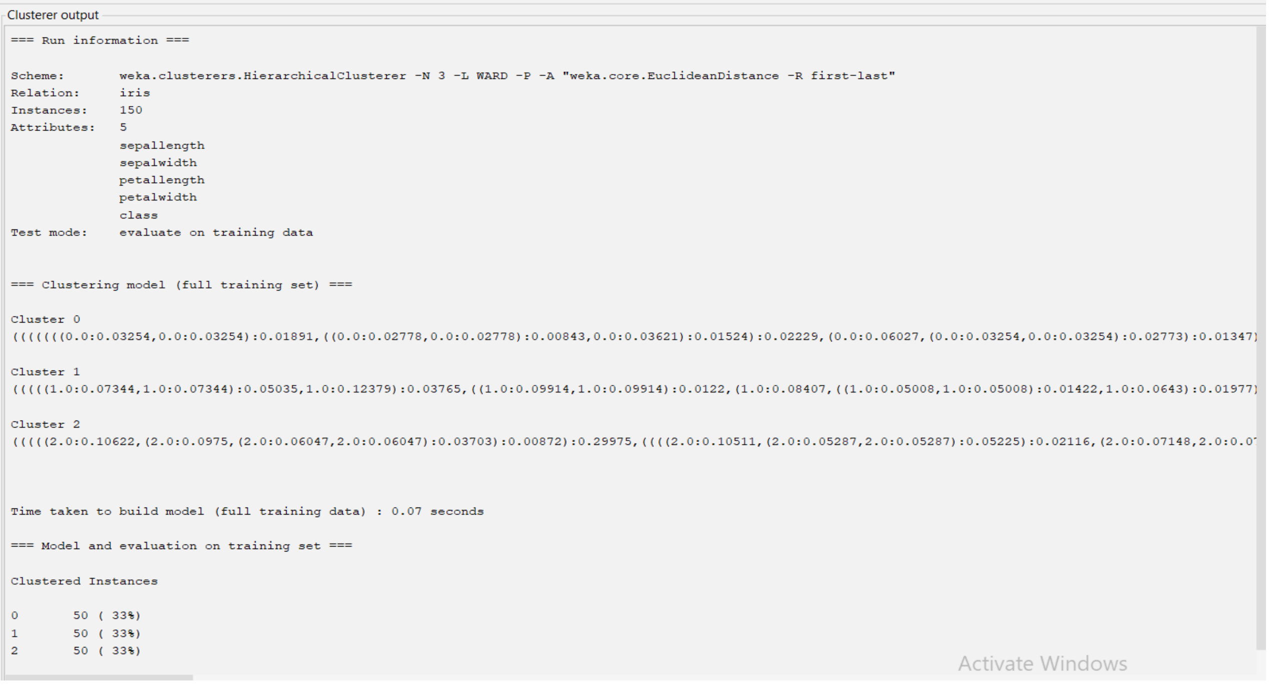
Y: sepallength (Num)

Colour: Cluster (Nom)

d) Hierarchical Clustering

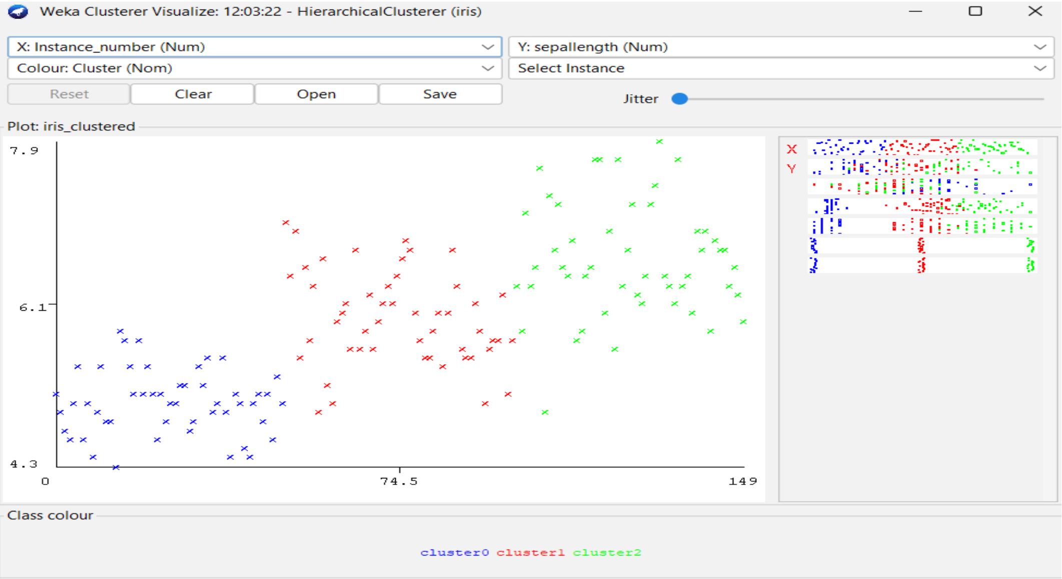
Click "Choose", select "HierarchicalClusterer".

Choose linkType as Ward, distanceFunction as Euclidean, numClusters to 3, and other specified settings.



Clusterer output

```
=== Run information ===

Scheme:       weka.clusterers.HierarchicalClusterer -N 3 -L WARD -P -A "weka.core.EuclideanDistance -R first-last"
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===

Cluster 0
((((((0.0:0.03254,0.0:0.03254):0.01891,((0.0:0.02778,0.0:0.02778):0.00843,0.0:0.03621):0.01524):0.02229,(0.0:0.06027,(0.0:0.03254,0.0:0.03254):0.02773):0.01347)

Cluster 1
((((1.0:0.07344,1.0:0.07344):0.05035,1.0:0.12379):0.03765,((1.0:0.09914,1.0:0.09914):0.0122,(1.0:0.08407,((1.0:0.05008,1.0:0.05008):0.01422,1.0:0.0643):0.01977)

Cluster 2
((((2.0:0.10622,(2.0:0.0975,(2.0:0.06047,2.0:0.06047):0.03703):0.00872):0.29975,((((2.0:0.10511,(2.0:0.05287,2.0:0.05287):0.05225):0.02116,(2.0:0.07148,2.0:0.0


Time taken to build model (full training data) : 0.07 seconds

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

Clustered Instances

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



Weka Clusterer Visualize: 12:03:22 - HierarchicalClusterer (iris)

X: Instance_number (Num)

Y: sepallength (Num)

Colour: Cluster (Nom)

Select Instance

Result:

Executing the clustering algorithms (K-Means, COBWEB, Canopy, Hierarchical) on the Iris dataset in WEKA, the analysis indicates that K-Means and Hierarchical clustering effectively grouped the instances into meaningful clusters, closely reflecting the natural species division, with K-Means showing efficiency in execution time.