LAB Manual
PART A
(PART A: TO BE REFFERED BY STUDENTS)

Experiment No.08

A.1 Aim:

Implementation of Agglomerative hierarchical clustering in any programming language like JAVA, C++, C# or WEKA tool.

A.2 Prerequisite:

Familiarity with the WEKA tool and programming languages.

A.3 Outcome:

After successful completion of this experiment students will be able to

□ Use classification and clustering algorithms of data mining.

A.4 Theory:

THEORY:

Hierarchical Clustering:-

Build a tree-based hierarchical taxonomy (dendrogram) from a set of documents.

One approach: recursive application of a partitional clustering algorithm.

Dendogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.

Hierarchical Clustering algorithms:-

Agglomerative (bottom-up):

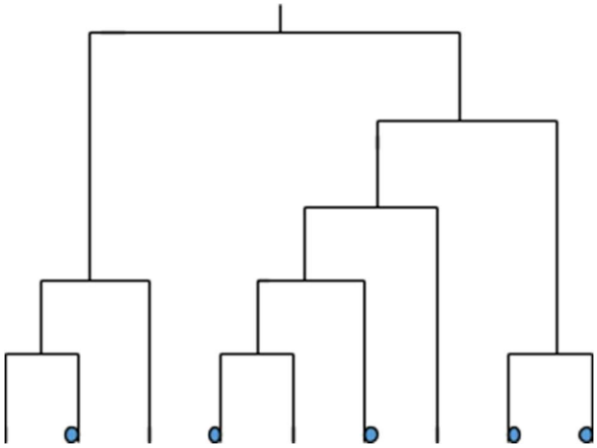
1. Start with each document being a single cluster.
2. Eventually all documents belong to the same cluster.

Divisive (top-down):

1.
2. Start with all documents belong to the same cluster.
3. Eventually each node forms a cluster on its own.
4. Does not require the number of clusters ***k*** in advance
5. Needs a termination/readout condition
6. The final mode in both Agglomerative and Divisive is of no use.

Dendogram: Hierarchical Clustering

Clustering obtained by cutting the **dendrogram** at a desired level: each connected component forms a cluster.



Many variants to defining closest pair of clusters:-

Single-link: Similarity of the *most* cosine-similar (single-link)

Complete-link: Similarity of the "furthest" points, the *least* cosine-similar

Centroid : Clusters whose centroids (centers of gravity) are the most cosine-similar

Average-link: Average cosine between pairs of elements

PART B

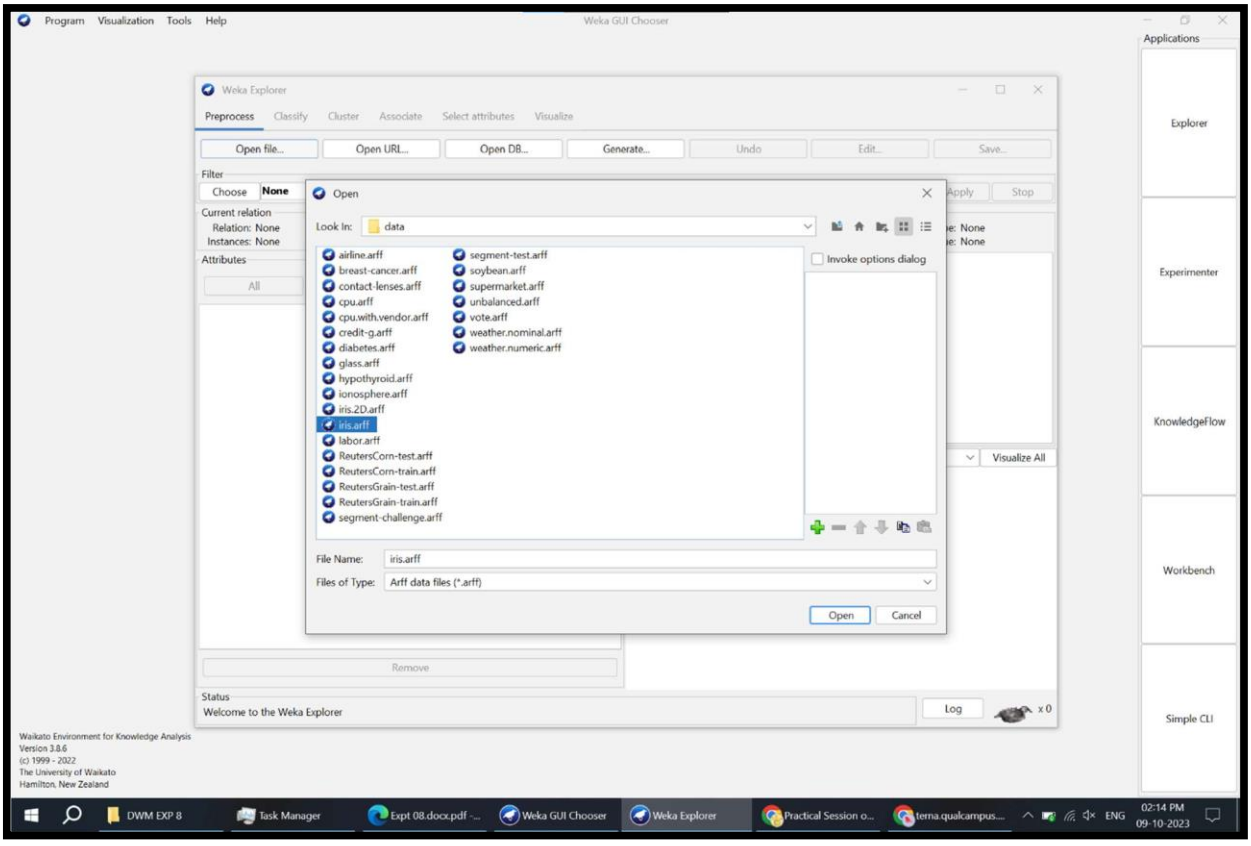
Roll. No.: A12	Name: Sufiyan Khan
Class: TE(AI&DS)	Batch: A1
Date of Experiment:	Date of Submission:
Grade:	

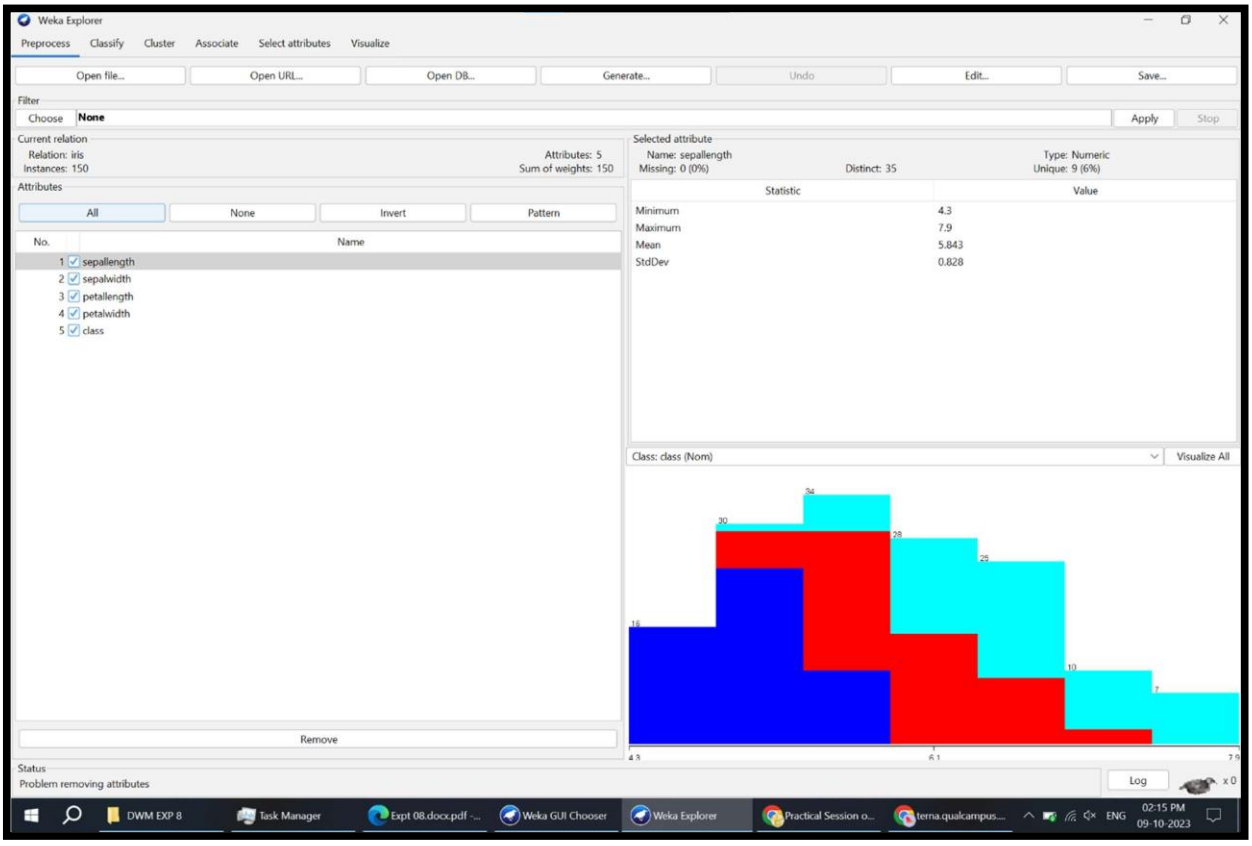
B1. Software Code written by student:

Implementation of Agglomerative hierarchical clustering in WEKA tool.

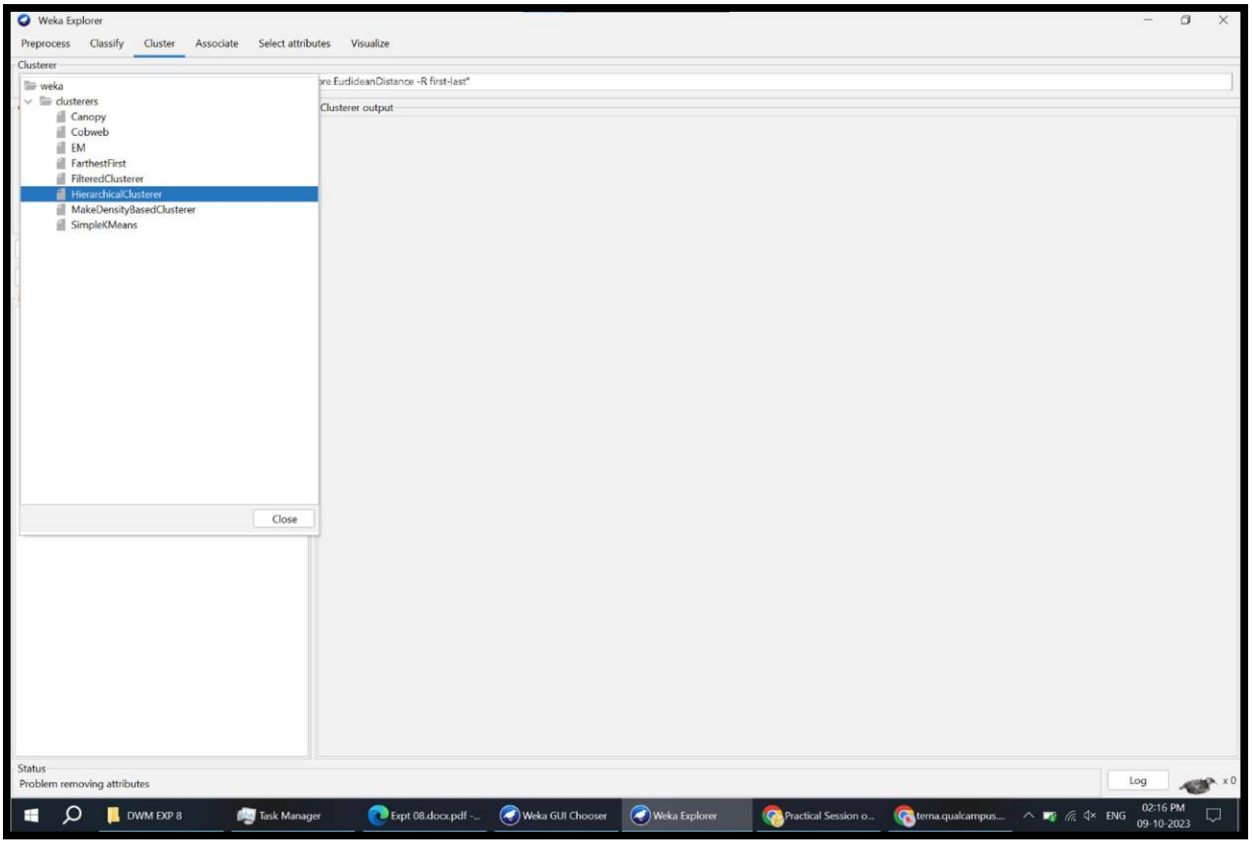
B.2 Input and Output:

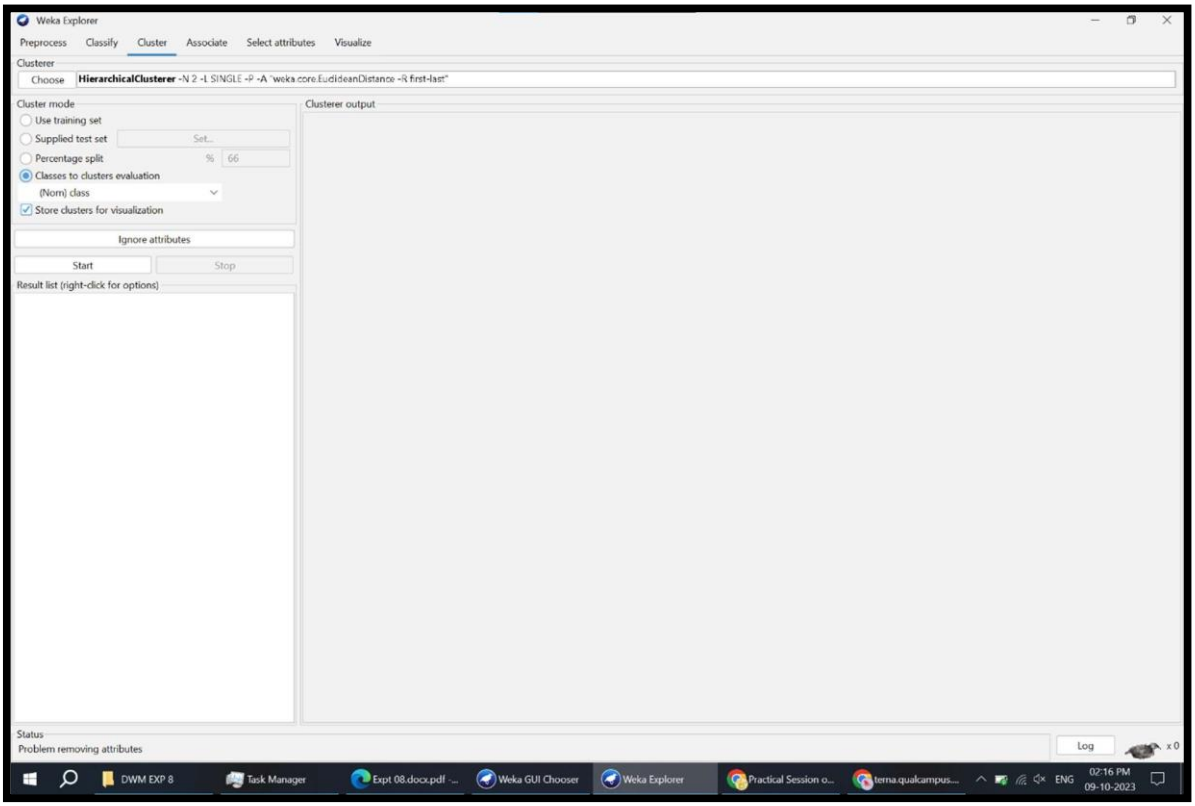
Step 1: Selecting the iris.arff file from database.



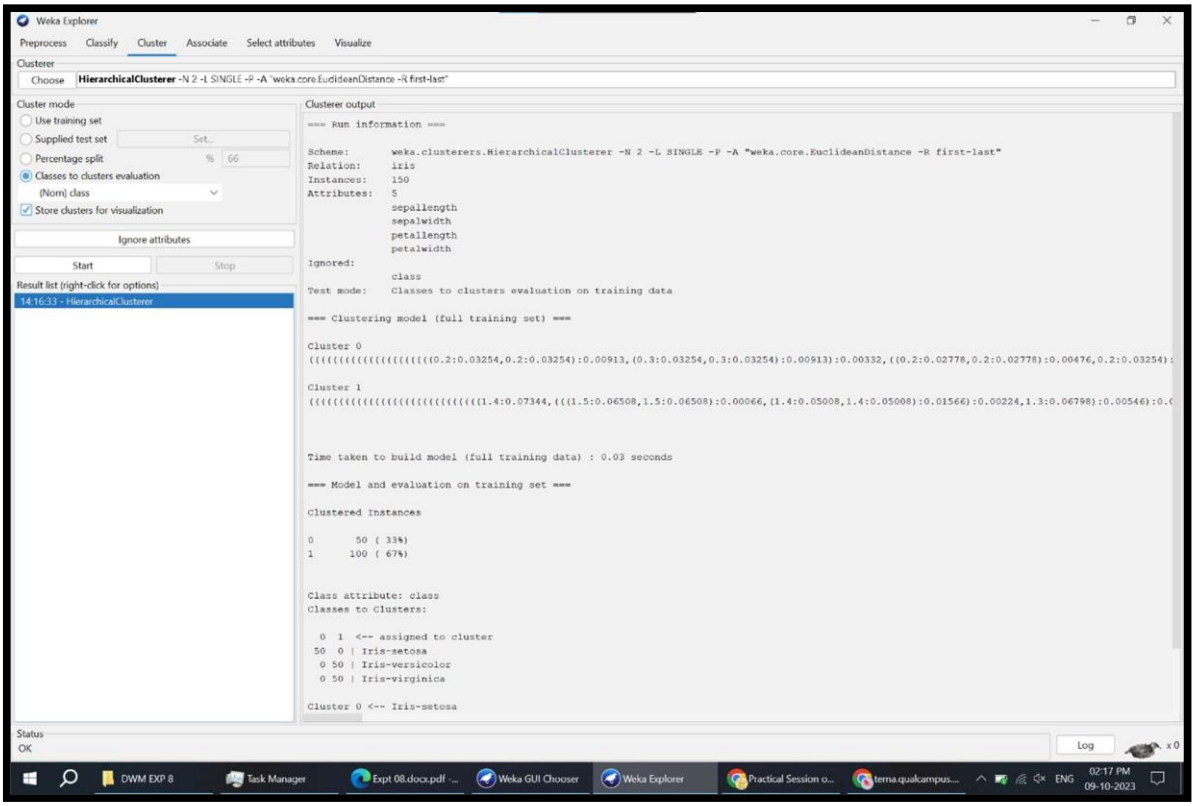


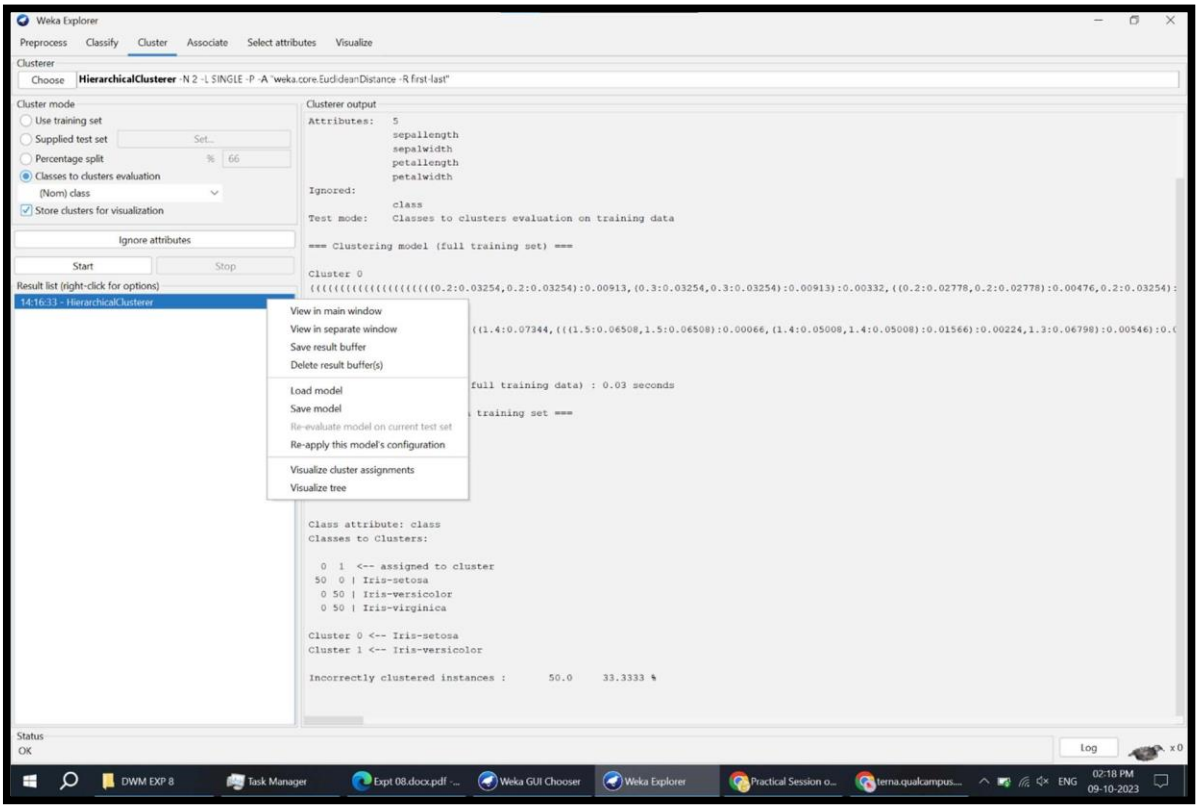
Step 2: Click on Cluster in the above tab, and select HierarchicalCluster, then click on Start button.

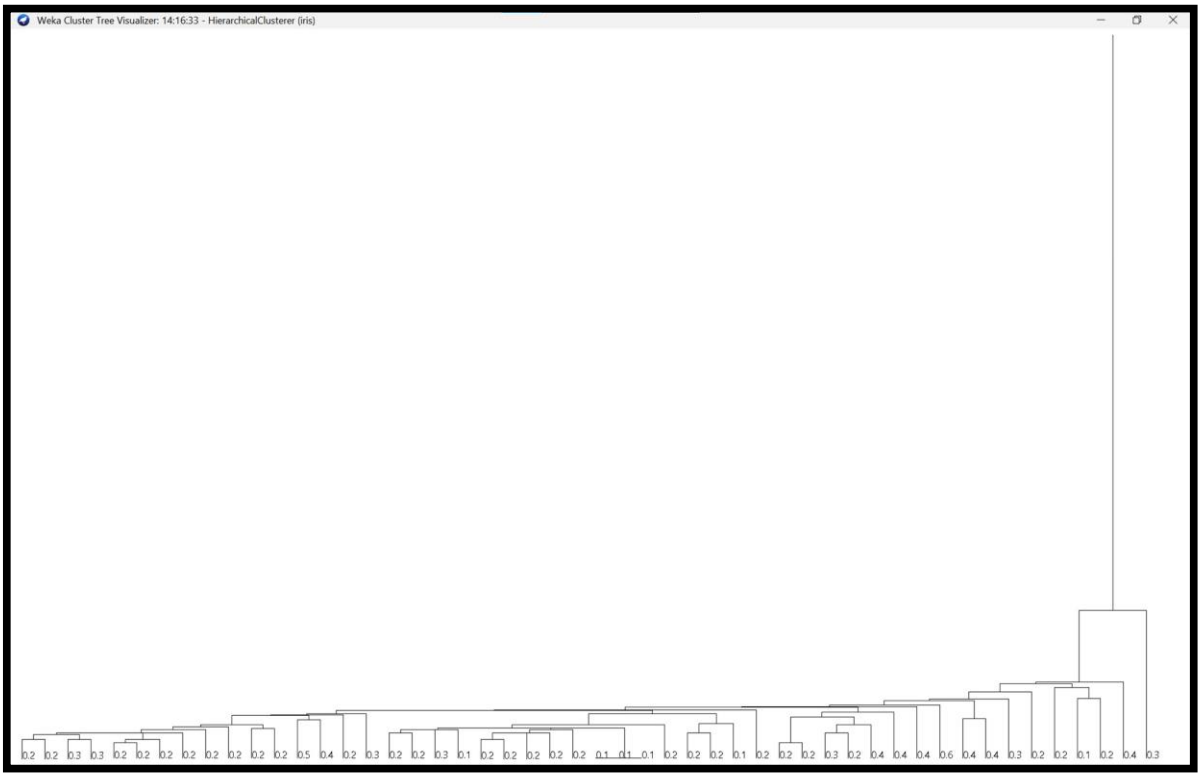




Step 3: Data will be shown, Right click on HierarchicalCluster (highlighted below) and select the Visualize tree.




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
14:16:33 - HierarchicalClusterer
View in main window
View in separate window
Save result buffer
Delete result buffer(s)
Load model
Save model
Re-evaluate model on current test set
Re-apply this model's configuration
Visualize cluster assignments
Visualize tree
Clusterer output
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Cluster 0
full training data) : 0.03 seconds
training set ===
Class attribute: class
Classes to Clusters:
0 1 <-- assigned to cluster
50 0 | Iris-setosa
0 50 | Iris-versicolor
0 50 | Iris-virginica
Cluster 0 <-- Iris-setosa
Cluster 1 <-- Iris-versicolor
Incorrectly clustered instances : 50.0 33.3333 %
Status
OK
Log
x 0
DWM EXP 8
Task Manager
Expt 08.docx.pdf - ...
Weka GUI Chooser
Weka Explorer
Practical Session o...
terna.qualcampus....
ENG
02:18 PM
09-10-2023


Weka Cluster Tree Visualizer: 14:16:33 - HierarchicalClusterer (iris)

B.3 Observation and learning:

In WEKA, agglomerative hierarchical clustering enables the creation of hierarchical clusters, offering flexibility through various linkage methods for cluster formation. It's essential to select the most suitable linkage method based on your data and goals. Hierarchical clustering is valuable for exploring intricate data relationships, and visualizing the dendrogram aids in comprehending cluster hierarchies. To enhance clustering outcomes, experiment with different distance metrics and parameters, tailoring the approach to your specific dataset and objectives.

B.4 Conclusion:

In conclusion, agglomerative hierarchical clustering in WEKA offers versatile clustering capabilities through different linkage methods. Careful method selection, visualization, and experimentation with parameters are key to harnessing its potential for exploring complex data structures effectively.

B.5 Question of Curiosity

Q1: Explain the advantages and disadvantages of agglomeration and hierarchicalclustering.

Agglomeration and hierarchical clustering are two common techniques used in data analysis and data mining to group similar data points together based on certain criteria. Each method has its own advantages and disadvantages, making them suitable for different scenarios and data types.

Agglomeration (or bottom-up) clustering:

Advantages:

Simplicity: Agglomeration is straightforward and easy to understand. It starts with each data point as its own cluster and then gradually merges them based on similarity, creating a hierarchical structure.

Hierarchical representation: It results in a hierarchical tree-like structure (dendrogram), which can be visually insightful and provide multiple levels of clustering granularity.

No need to specify the number of clusters: Agglomeration doesn't require you to predefine the number of clusters, making it useful when you don't have prior knowledge about the data structure.

Can handle various distance metrics: It can be adapted to different distance or similarity measures depending on the nature of the data.

Disadvantages:

Computationally intensive: Agglomeration can be computationally expensive, especially for large datasets, as it involves pairwise comparisons between data points at each step.

Sensitive to noise: It can be sensitive to outliers or noise in the data, as a single outlier can significantly affect the merging process.

Non-reversible: Once clusters are merged, you cannot easily undo the process to obtain a different clustering solution without starting over.

Hierarchical clustering (or divisive clustering):

Advantages:

Control over cluster granularity: Hierarchical clustering allows you to control the granularity of clusters by specifying the number of clusters or by cutting the dendrogram at a certain level. This flexibility is beneficial when you have prior knowledge about the desired number of clusters.

Less sensitive to noise: Divisive clustering is less sensitive to outliers compared to agglomeration, as it starts with all data points in a single cluster and splits them based on dissimilarity.

Better for non-hierarchical data: It can be a better choice when the data does not exhibit a clear hierarchical structure.

Disadvantages:

Complexity: Divisive clustering can be more complex to implement and understand compared to agglomeration because it involves recursively dividing clusters.

Difficulty in choosing the number of clusters: Unlike agglomeration, you need to specify the number of clusters in advance, which can be challenging without prior knowledge of the data.

Limited hierarchical representation: While divisive clustering can create a hierarchical structure, it may not be as visually informative as the dendrogram produced by agglomeration.

Q2: What is the relationship between top-down, bottom-up and division /agglomeration?

Top-down, bottom-up, and division/agglomeration are terms often used to describe different approaches to hierarchical clustering. They represent the different directions in which the clustering process can proceed within the hierarchical framework.

Top-Down (Divisive) Clustering:

In top-down clustering, also known as divisive clustering, you start with all data points grouped together in a single cluster.
The process involves recursively dividing this single cluster into smaller clusters based on dissimilarity until you reach the desired number of clusters or another

stopping criterion.
Divisive clustering is a "divide and conquer" approach, where you start with one large cluster and divide it into smaller clusters in a hierarchical manner.

Bottom-Up (Agglomerative) Clustering:

In bottom-up clustering, also known as agglomerative clustering, you start with each data point as its own cluster.
The process involves merging the closest clusters iteratively based on similarity until all data points are in a single cluster or until you reach a desired number of clusters.
Agglomerative clustering is a "build-up" approach, where you start with individual data points and progressively merge them into larger clusters in a hierarchical manner.

Division/Agglomeration:

Division and agglomeration are general terms used to describe the two fundamental actions in hierarchical clustering, whether they are performed top-down or bottom-up.
Division refers to the process of splitting clusters into smaller subclusters based on dissimilarity, which is associated with divisive (top-down) clustering.
Agglomeration refers to the process of merging clusters into larger clusters based on similarity, which is associated with agglomerative (bottom-up) clustering.