

Experiment No.07

A.1 Aim: Implementation of Agglomerative hierarchical clustering in any programming language like JAVA, C++, C# or WEKA tool.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: Comps TE B	Batch: B3
Date of Experiment: 15/04/2021	Date of Submission: 15/04/2021
Grade:	

B.1 Software Code written by a student:

(Paste your problem statement related to your case study completed during the 2 hours of practice in the lab here)

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
dataset = pd.read_csv('age_bmi.csv')
dataset.head()
X = dataset.iloc[:, [0, 1]].values

import scipy.cluster.hierarchy as shc
plt.figure(figsize=(10, 7))
plt.title("Patient Dendrograms")
dend = shc.dendrogram(shc.linkage(dataset, method='ward'))

from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
labels=cluster.fit_predict(dataset)

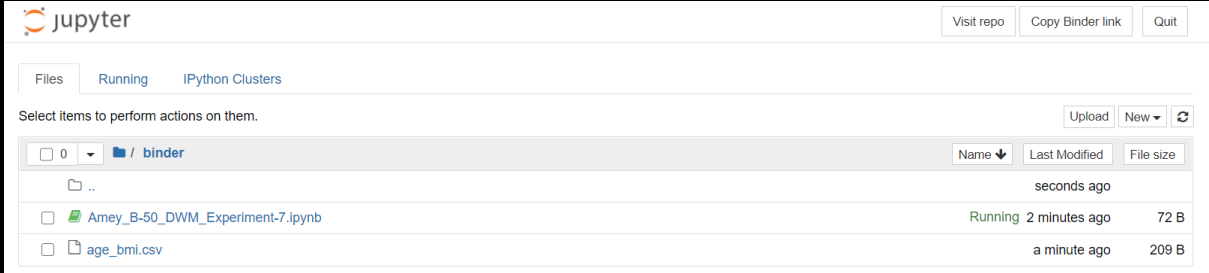
plt.scatter(X[labels==0, 0], X[labels==0, 1], s=50, marker='o',
color='green')
plt.scatter(X[labels==1, 0], X[labels==1, 1], s=50, marker='o',
color='blue')
plt.scatter(X[labels==2, 0], X[labels==2, 1], s=50, marker='o',
color='red')
plt.show()
```

B.2 Input and Output:

(Paste your program input and output in the following format, If there is an error then paste the specific error in the output part. In case of an error with the due permission of the faculty, an extension can be given to submit the error-free code with output in due course of time. Students will be graded accordingly.)

Jupyter Notebook

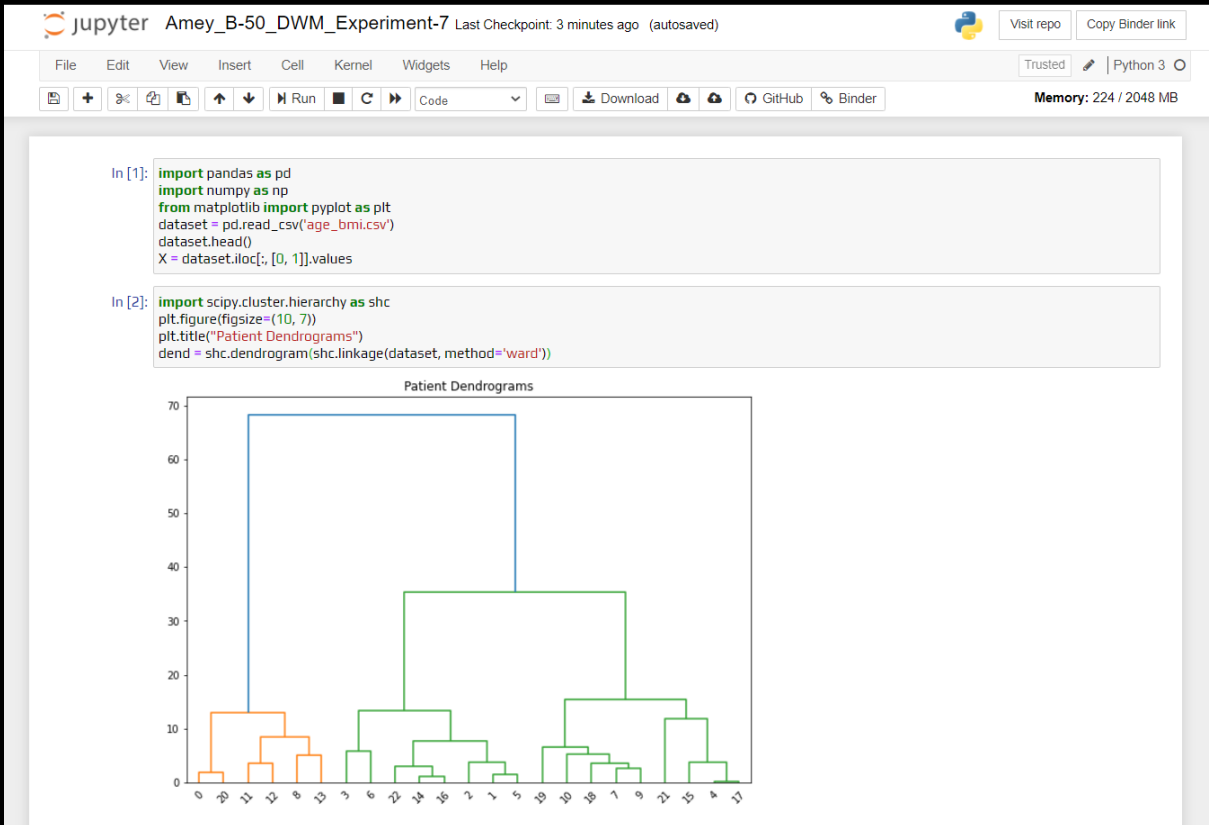
Jupyter/Binder:



The screenshot shows the Jupyter/Binder interface. At the top, there's a header with the Jupyter logo and buttons for "Visit repo", "Copy Binder link", and "Quit". Below the header, there are tabs for "Files", "Running", and "IPython Clusters". The "Files" tab is active, showing a list of files and folders. The files are:

Name	Last Modified	File size
..	seconds ago	
Amey_B-50_DWM_Experiment-7.ipynb	Running 2 minutes ago	72 B
age_bmi.csv	a minute ago	209 B

Jupyter Notebook:

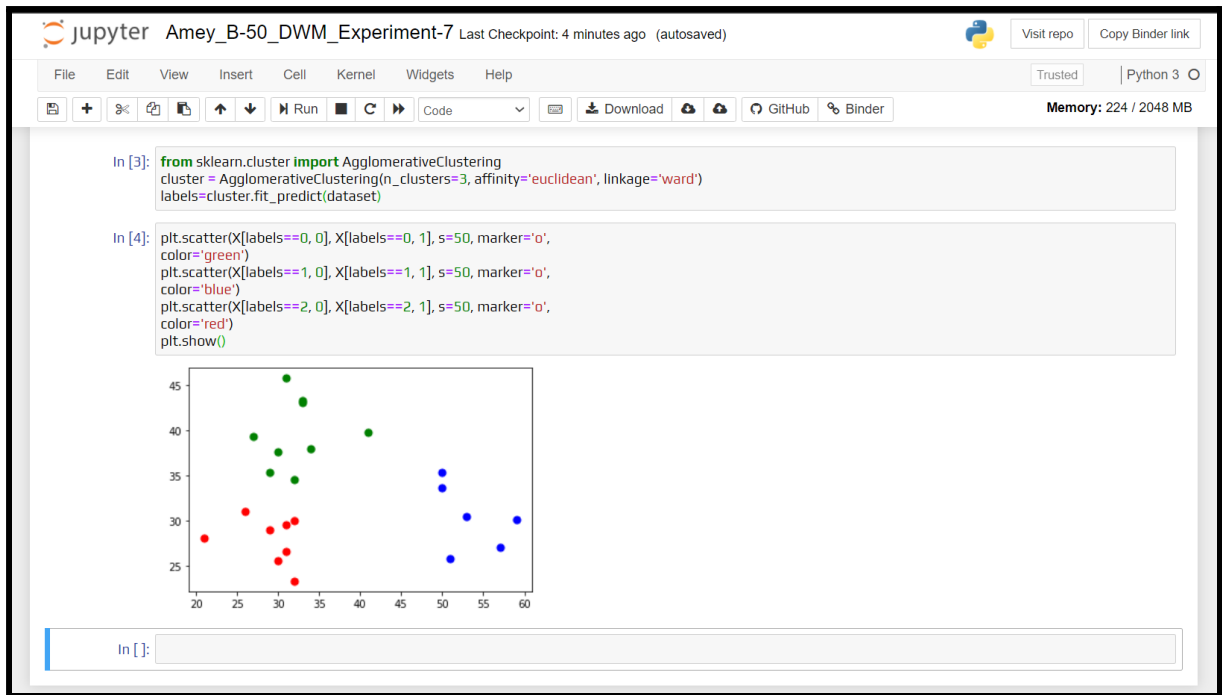


The screenshot shows a Jupyter Notebook interface. The top bar includes the Jupyter logo, the notebook name "Amey_B-50_DWM_Experiment-7", and buttons for "Visit repo" and "Copy Binder link". The notebook is running on Python 3. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and saving. The notebook content consists of two code cells and a plot.

```
In [1]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
dataset = pd.read_csv('age_bmi.csv')
dataset.head()
X = dataset.iloc[:, [0, 1]].values
```

```
In [2]: import scipy.cluster.hierarchy as shc
plt.figure(figsize=(10, 7))
plt.title("Patient Dendrograms")
dend = shc.dendrogram(shc.linkage(dataset, method='ward'))
```

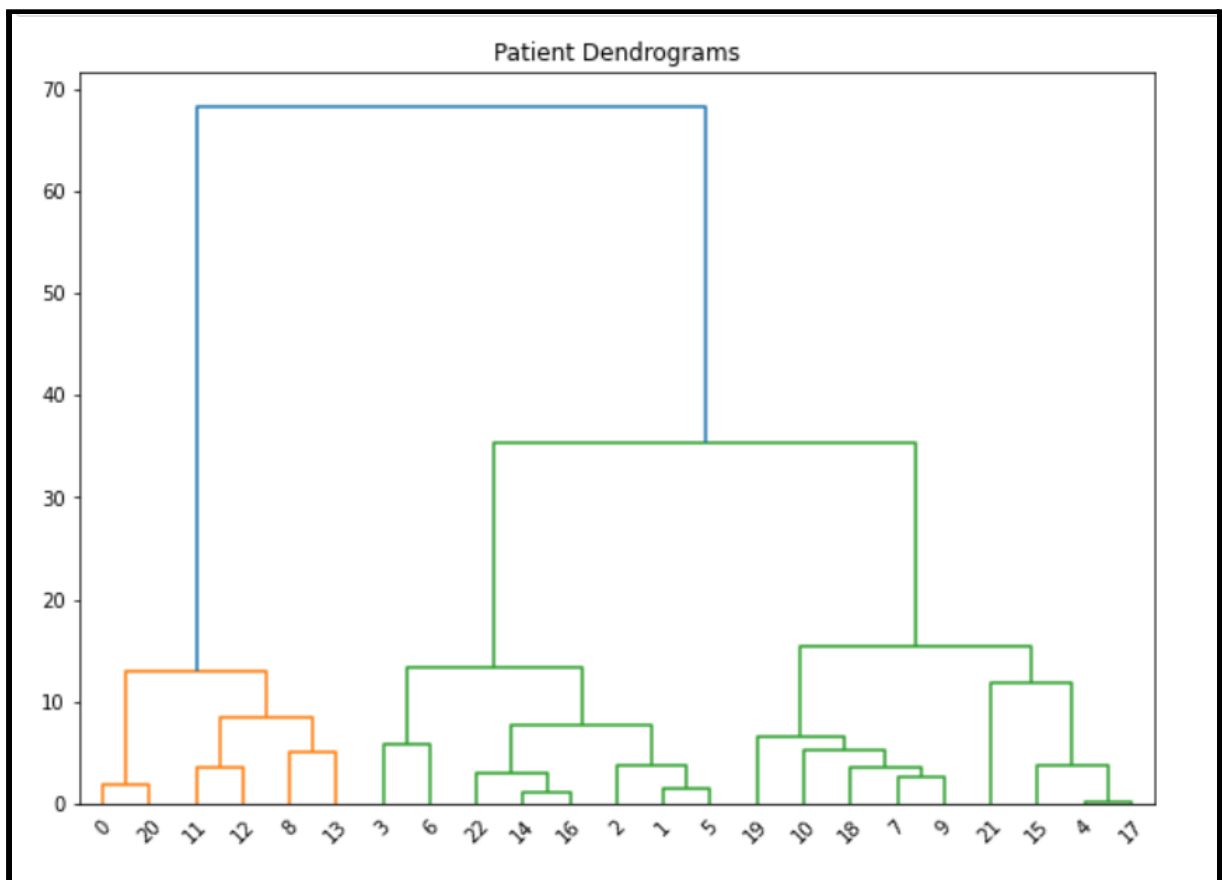
The plot, titled "Patient Dendrograms", shows a hierarchical clustering dendrogram. The x-axis represents the patients, and the y-axis represents the distance between clusters. The dendrogram shows two main clusters joining at a distance of approximately 68. The first main cluster is composed of two sub-clusters: one with 5 patients (labeled 0, 1, 2, 3, 4) and another with 10 patients (labeled 5, 6, 7, 8, 9, 10, 11, 12, 13, 14). The second main cluster is composed of two sub-clusters: one with 5 patients (labeled 15, 16, 17, 18, 19) and another with 5 patients (labeled 20, 21, 22, 23, 24).



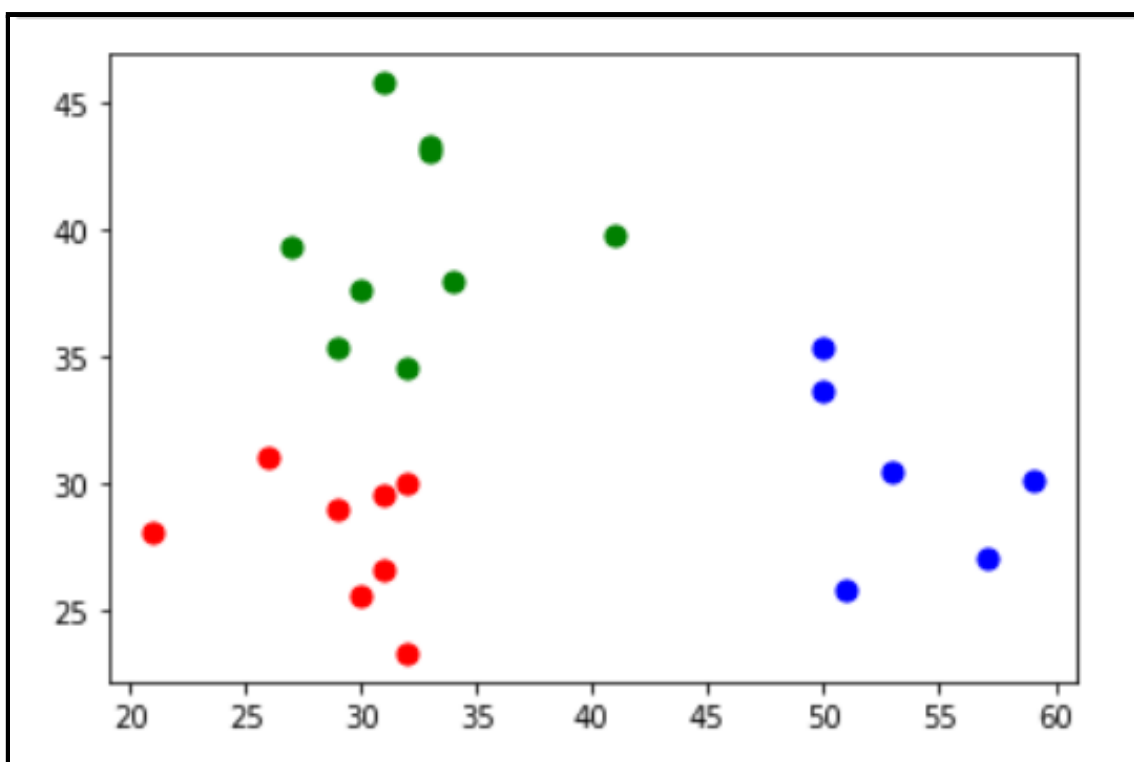
Sample Dataset:

	age	mass
0	50	33.6
1	31	26.6
2	32	23.3
3	21	28.1
4	33	43.1

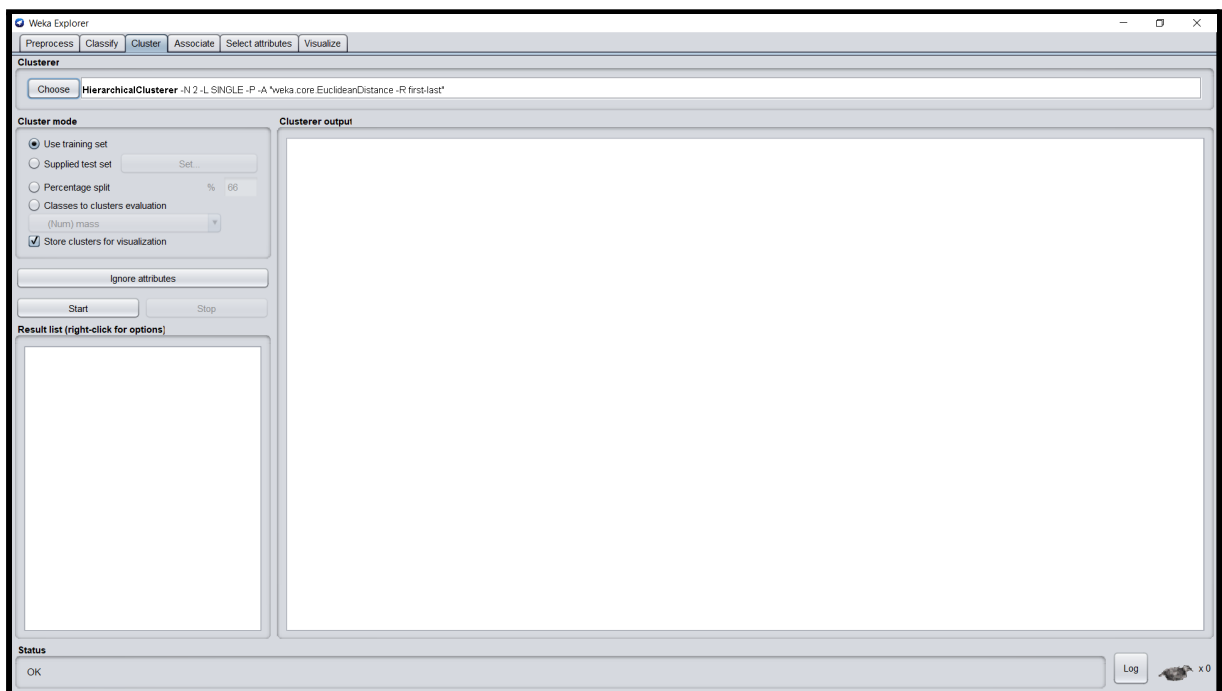
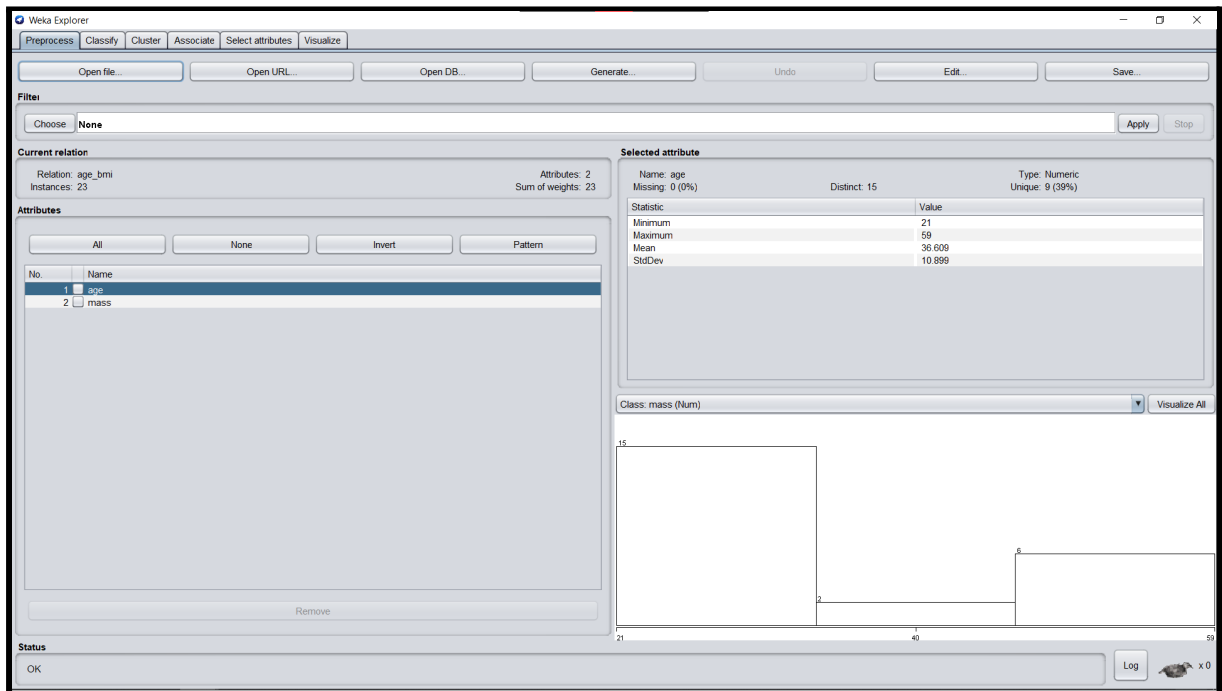
Dendrogram:



Cluster Visualization:



Weka Tool



weka.gui.GenericObjectEditor

weka.clusterers.HierarchicalClusterer

About

Hierarchical clustering class.

More

Capabilities

debug False

distanceFunction Choose **EuclideanDistance** -R first-last

distancelsBranchLength False

doNotCheckCapabilities False

linkType WARD

numClusters 3

printNewick True

Open... Save... OK Cancel

Weka Output:

The screenshot shows the Weka Explorer interface with the HierarchicalClusterer algorithm selected. The 'Cluster mode' section on the left has 'Use training set' selected. The 'Clusterer output' section on the right displays the following information:

```
=== Run information ===
Scheme:      weka.clusterers.HierarchicalClusterer -N 3 -L WARD -P -A "weka.core.EuclideanDistance -R first-last"
Relation:    age_bmi
Instances:   23
Attributes:  2
              age
              mass
Test mode:   evaluate on training data

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

Cluster 0
((33.6:0.08,35.4:0.08):0.32318,((30.5:0.13736,(27.1:0.14335,30.1:0.14335):-0.00598):0.05311,25.8:0.19047):0.21271)

Cluster 1
(((26.6:0.05165,25.6:0.05165):0.08005,23.3:0.1317):0.21384,(28.1:0.25272,(31.0:0.15412,((30.0:0.03176,29.6:0.03176):0.03636,29.0:0.06812):0.086):0.0986):0.09282)

Cluster 2
((((43.1:0.00889,43.3:0.00889):0.15161,45.8:0.16049):0.12879,39.8:0.28928):0.37716,(((35.3:0.08486,34.6:0.08486):0.02184,37.6:0.1067):0.025,38.0:0.1317):0.01347)

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

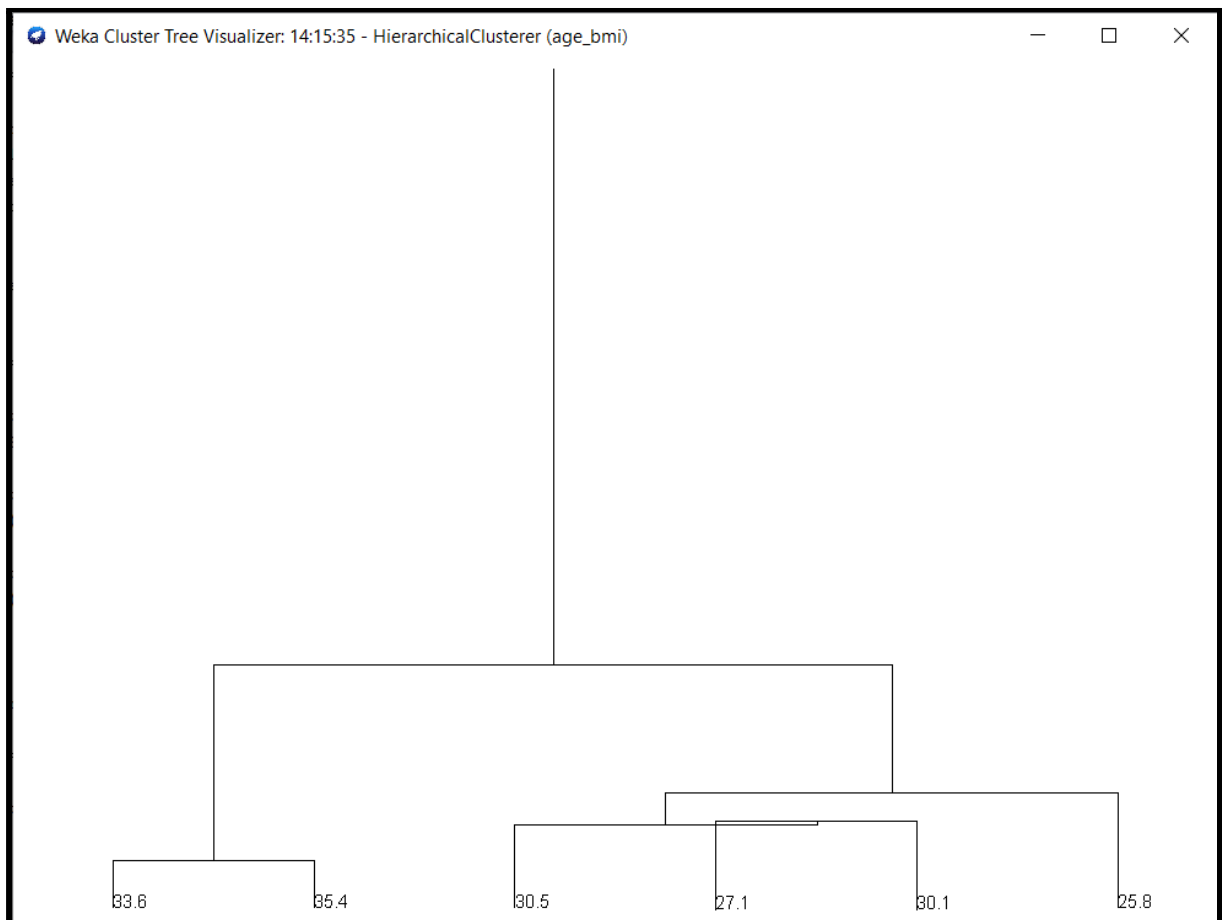
=== Model and evaluation on training set ===

Clustered Instances

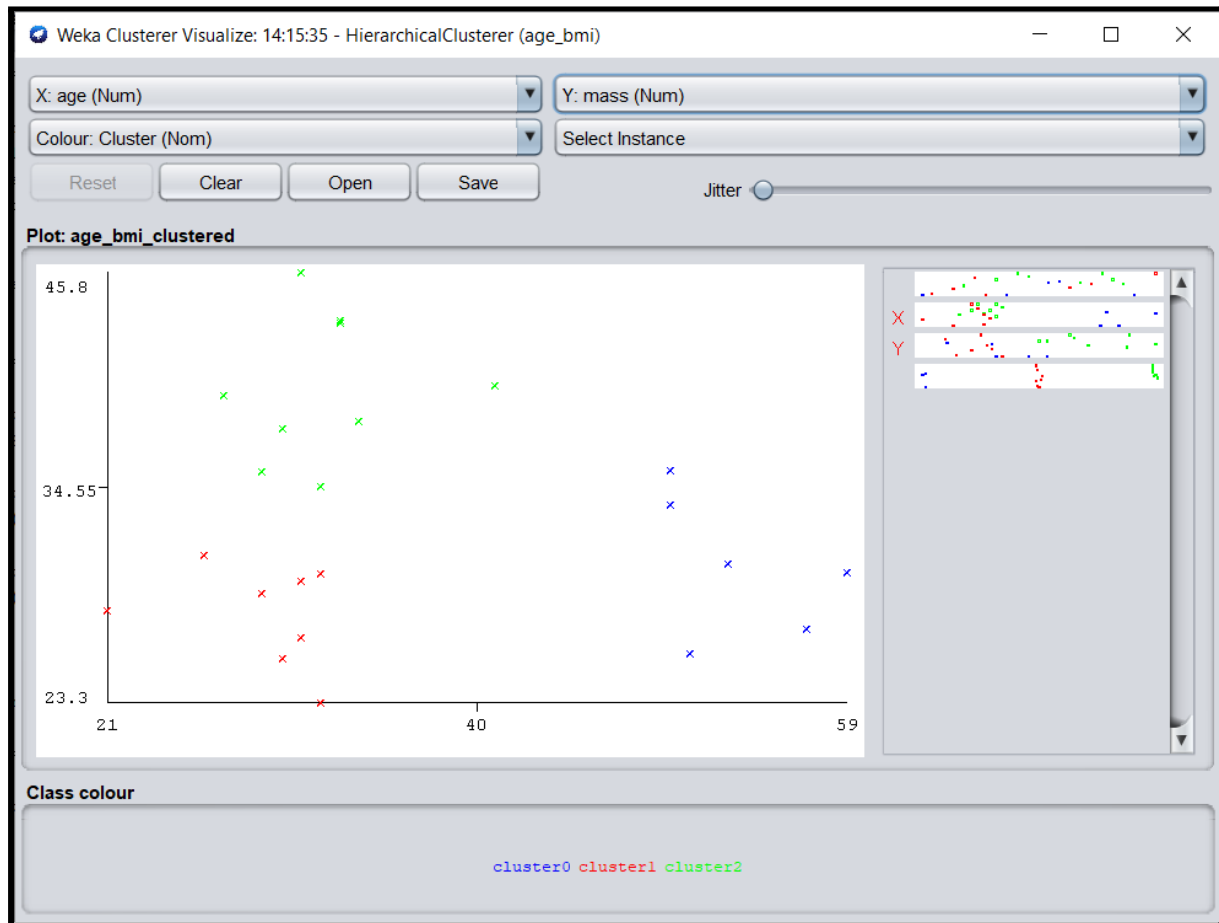
0      6 ( 26%)
1      8 ( 35%)
2      9 ( 39%)
```

The 'Result list' on the left shows '14:15:35 - HierarchicalClusterer' as the selected result.

Dendrogram:



Cluster Visualization:



B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ subpart assigned)

Agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It's also known as AGNES (Agglomerative Nesting). The algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all objects. The result is a tree-based representation of the objects, named dendrogram.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

Understood that hierarchy within the final cluster has the following properties:

- Clusters generated in early stages are nested in those generated in later stages.
- Clusters with different sizes in the tree can be valuable for discovery.

Hence we've successfully implemented Agglomerative clustering through Python as well as Weka Tool.

B.5 Question of Curiosity

(To be answered by the student based on the practical performed and learning/observations)

1. Explain the advantages and disadvantages of agglomeration and hierarchical clustering.

Ans:

Advantages

- It can produce an ordering of the objects, which may be informative for data display.
- Smaller clusters are generated, which may be helpful for discovery.

Disadvantages

- No provision can be made for a relocation of objects that may have been 'incorrectly' grouped at an early stage. The result should be examined closely to ensure it makes sense.
- The use of different distance metrics for measuring distances between clusters may generate different results. Performing multiple experiments and comparing the results is recommended to support the veracity of the original results.

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

Ans:

- The agglomerative hierarchical clustering method allows the clusters to be read from bottom to top and it follows this approach so that the program always reads from the sub-component first then moves to the parent. Whereas, divisive uses a top-bottom approach in which the parent is visited first then the child.
- Agglomerative hierarchical methods consist of objects in which each object creates its clusters and these clusters are grouped to create a large cluster. It defines a process of merging that carries on till all the single clusters are merged into a complete big cluster that will consist of all the objects of child clusters. Whereas, in divisive the parent cluster is divided into smaller clusters and it keeps on dividing till each cluster has a single object to represent.
- Divisive clustering is more complex as compared to agglomerative clustering, as in the case of divisive clustering we need a flat clustering method as a “subroutine” to split each cluster until we have each data having its singleton cluster.
- The divisive algorithm is also more accurate. Agglomerative clustering makes decisions by considering the local patterns or neighbour points without initially taking into account the global distribution of data. These early decisions cannot be undone. whereas divisive clustering takes into consideration the global distribution of data when making top-level partitioning decisions.