Lab Number: 10

Title
Perform Hierarchical Clustering

Objective
To apply hierarchical clustering on a suitable dataset in WEKA, generate a dendrogram to visualize cluster formation, and analyze the resulting cluster structure.

IDE/Tools Used
Weka 3.8.6

Theory
Clustering: Clustering is an unsupervised machine learning technique that groups similar data points into clusters based on defined similarity measures, such as Euclidean distance. Data points within a cluster are more similar to each other than to those in other clusters, which helps in identifying patterns and structure in unlabeled data. It's a versatile tool with applications in fields like market segmentation, social network analysis, and data compression.

Hierarchical Clustering: Hierarchical clustering is an unsupervised learning technique that builds a hierarchy of clusters, either in an agglomerative (bottom-up) or divisive (top-down) manner.

- **Agglomerative:** Starts with each instance as a separate cluster and merges the closest clusters iteratively until all points form a single cluster.
- **Divisive:** Starts with all instances in one cluster and splits recursively until each instance forms its own cluster.

The linkage methods are:

- **Single linkage:** Distance between the closest points of two clusters.
- **Complete linkage:** Distance between the farthest points of two clusters.
- **Average linkage:** Average distance between all pairs of points in two clusters.

Dendrogram: Hierarchical clustering produces a dendrogram, which visually represents the merging or splitting of clusters at different levels, allowing selection of the desired number of clusters.

Implementation

The following steps are performed to implement hierarchical clustering in WEKA.

1. Preparing the Dataset

Select or create a dataset suitable for clustering. Save it as .csv, then open it in WEKA =>Tools => ARFF Viewer and save as .arff. Also ensure attributes are numeric, as hierarchical algorithm operates on continuous data (Using the same dataset as the K-means).

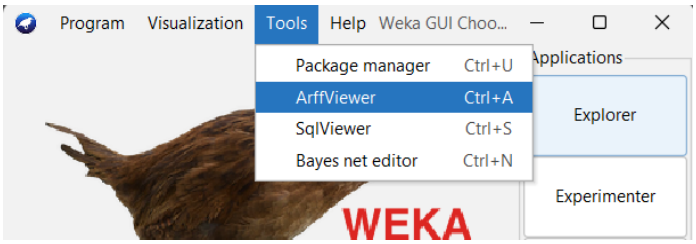


Figure 1: Opening the Arff Viewer

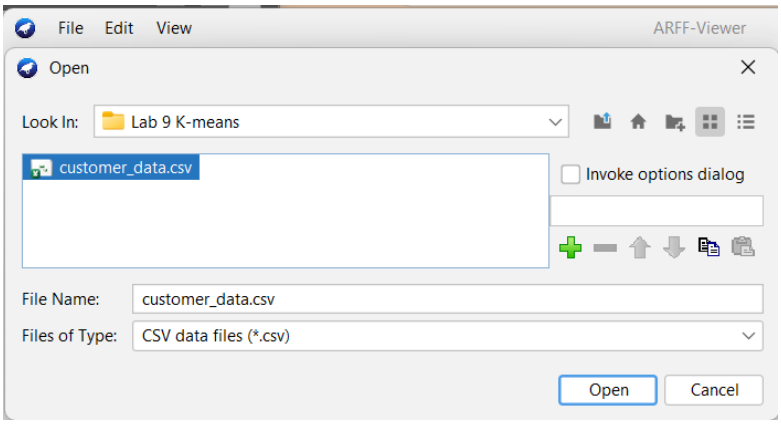
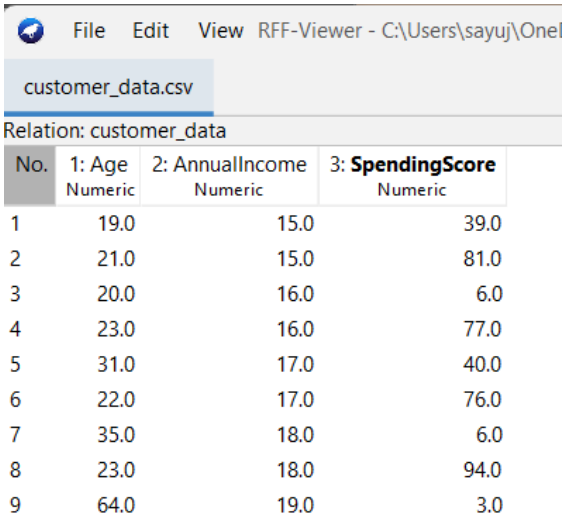


Figure 2: Selecting the csv file



No.	1: Age Numeric	2: AnnualIncome Numeric	3: **SpendingScore** Numeric
1	19.0	15.0	39.0
2	21.0	15.0	81.0
3	20.0	16.0	6.0
4	23.0	16.0	77.0
5	31.0	17.0	40.0
6	22.0	17.0	76.0
7	35.0	18.0	6.0
8	23.0	18.0	94.0
9	64.0	19.0	3.0

Figure 3: Visualization of the dataset

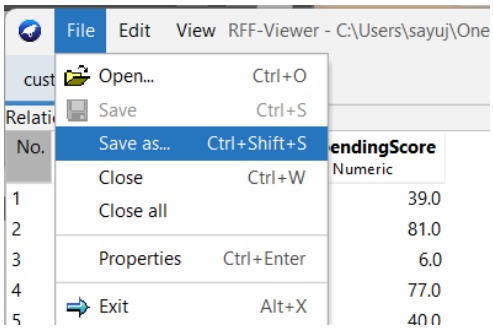


Figure 4: Option to save as arff

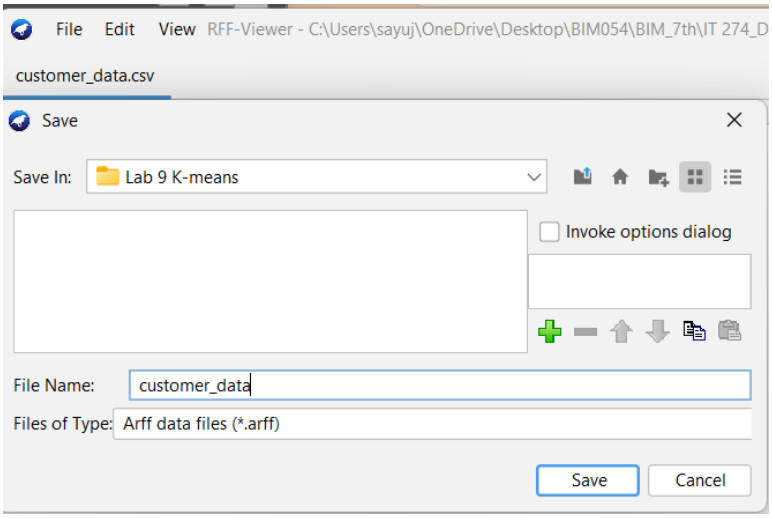


Figure 5: Saving as arff format

2. Loading the Dataset

2.1. Open WEKA Explorer => Preprocess => Open File

2.2. Load the prepared .arff file

2.3. Verify attribute types are numeric and clean any missing values if present

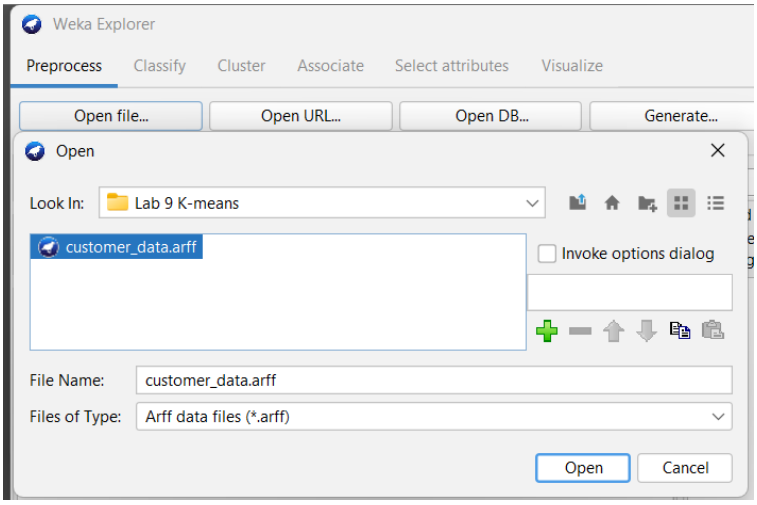
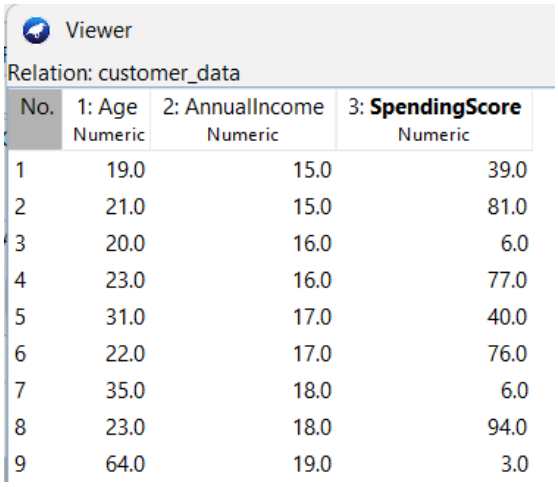


Figure 6: Opening the dataset in WEKA



No.	1: Age Numeric	2: AnnualIncome Numeric	3: **SpendingScore** Numeric
1	19.0	15.0	39.0
2	21.0	15.0	81.0
3	20.0	16.0	6.0
4	23.0	16.0	77.0
5	31.0	17.0	40.0
6	22.0	17.0	76.0
7	35.0	18.0	6.0
8	23.0	18.0	94.0
9	64.0	19.0	3.0

Figure 7: Verifying numeric data type

3. Running Hierarchical Clustering

3.1. Go to the Cluster tab in WEKA Explorer.

3.2. Select HierarchicalClusterer as the clustering algorithm.

3.3. Set Number of Clusters (K) = 2.

3.4. Configure parameters:

- Distance function (default: Euclidean)
- Linkage type: Single, Complete, Average. Repeated for each
- Number of clusters to generate: 4

Finally, Click Start to run Hierarchical clustering. The output will then show in the Clusterer output section in the left.

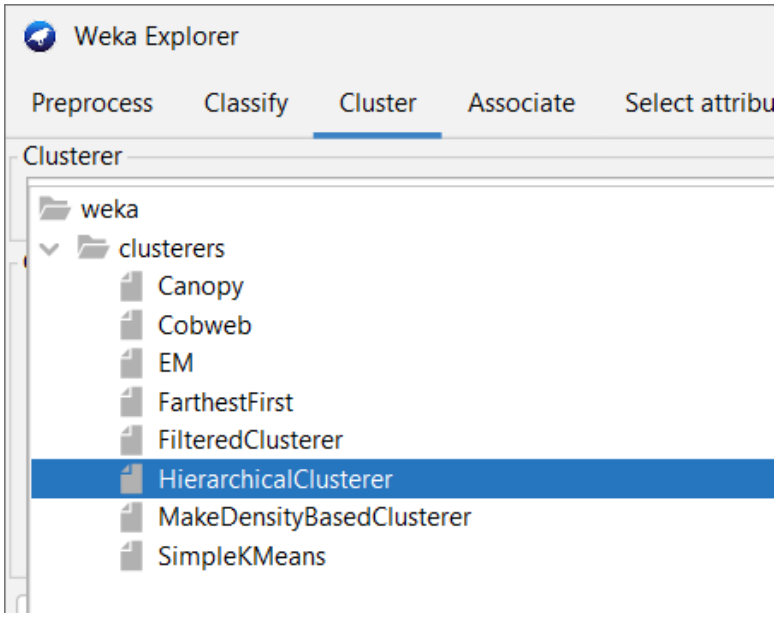


Figure 8: Choosing the Hierarchical clusterer

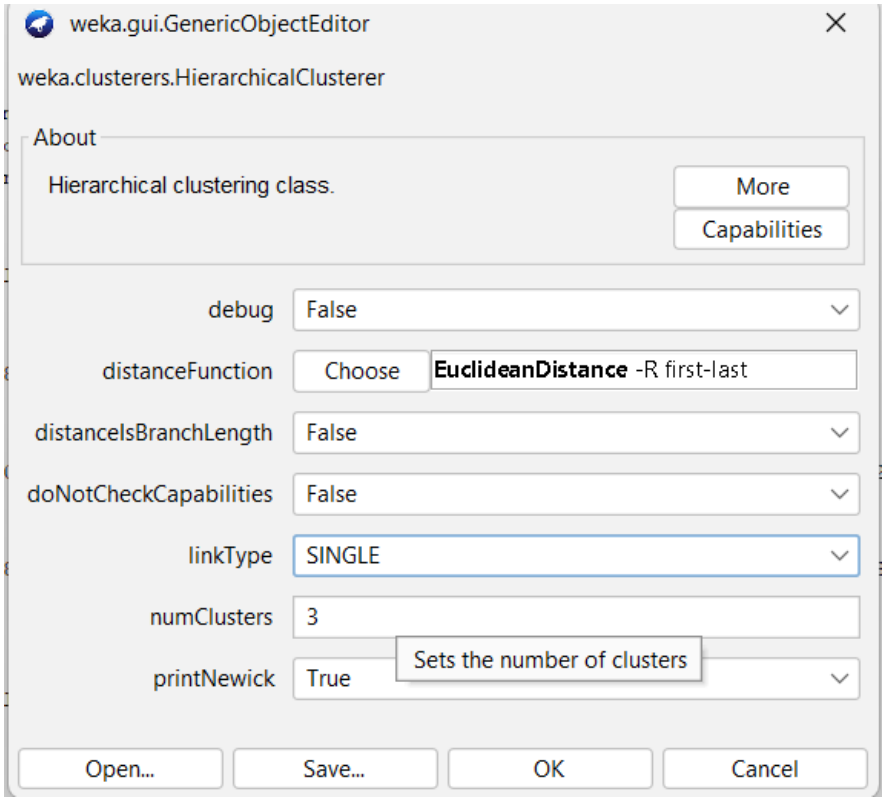


Figure 9: Adjusting the parameters for the algorithm (Single)

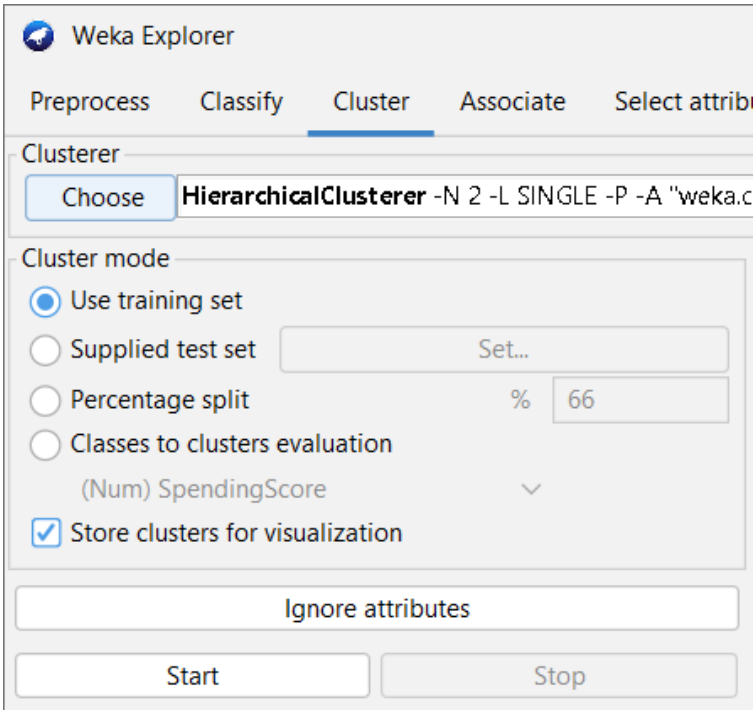


Figure 10: Starting the process

Output for Single Linkage

```
=== Run information ===

Scheme:          weka.clusterers.HierarchicalClusterer -N 3 -L SINGLE -P -A
"weka.core.EuclideanDistance -R first-last"
Relation:     customer_data
Instances:    30
Attributes:   3
              Age
              AnnualIncome
              SpendingScore
Test mode:    evaluate on training data


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

Cluster 0
(39.0:0.28813,40.0:0.28813)

Cluster 1
(((((81.0:0.0926,(77.0:0.07513,76.0:0.07513):0.01747):0.10913,94.0:0.20172):0
.02171,(((72.0:0.19684,(((77.0:0.08528,79.0:0.08528):0.07339,66.0:0.15867):0.
03258,(73.0:0.14397,(73.0:0.12501,82.0:0.12501):0.01896):0.04728):0.00559):0,
87.0:0.19684):0.01047,61.0:0.20731):0.01612):0.06918,(99.0:0.2859,98.0:0.2859
):0.00671):0.04445,((6.0:0.2978,(13.0:0.24363,(35.0:0.14286,35.0:0.14286):0.1
0077):0.05417):0.00269,(((3.0:0.14879,14.0:0.14879):0.05213,15.0:0.20091):0.0
8685,((29.0:0.26846,(5.0:0.20413,14.0:0.20413):0.06433):0.00616,(32.0:0.12673
,31.0:0.12673):0.14788):0.01315):0.01273):0.03656)



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

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

Clustered Instances

0       2 (  7%)
1      27 ( 90%)
2       1 (  3%)
```

Output for Average Linkage

```
=== Run information ===

Scheme:         weka.clusterers.HierarchicalClusterer -N 3 -L AVERAGE -P -A
"weka.core.EuclideanDistance -R first-last"
Relation:     customer_data
Instances:    30
Attributes:   3
              Age
              AnnualIncome
              SpendingScore
Test mode:    evaluate on training data


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

Cluster 0
((39.0:0.28813,40.0:0.28813):0.13061,(6.0:0.34361,6.0:0.34361):0.07513)

Cluster 1
(((((81.0:0.12304,(77.0:0.07513,76.0:0.07513):0.04791):0.10568,94.0:0.22871):
0.15358,(72.0:0.27901,((77.0:0.08528,79.0:0.08528):0.09466,66.0:0.17994):0.09
907):0.10329):0.07681,(99.0:0.2859,98.0:0.2859):0.1732):0.1308,(((73.0:0.1670
4,(73.0:0.12501,82.0:0.12501):0.04203):0.09734,61.0:0.26438):0.0366,87.0:0.30
098):0.28893)

Cluster 2
(((3.0:0.14879,14.0:0.14879):0.06531,15.0:0.2141):0.41686,((13.0:0.28006,(35.
0:0.14286,35.0:0.14286):0.13721):0.14124,((29.0:0.29118,(5.0:0.20413,14.0:0.2
0413):0.08705):0.0566,(32.0:0.12673,31.0:0.12673):0.22105):0.07352):0.20966)



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

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

Clustered Instances

0       4 ( 13%)
1      15 ( 50%)
2      11 ( 37%)
```

Output for Complete Linkage

```
=== Run information ===

Scheme:         weka.clusterers.HierarchicalClusterer -N 3 -L COMPLETE -P -A
"weka.core.EuclideanDistance -R first-last"
Relation:     customer_data
Instances:    30
Attributes:   3
              Age
              AnnualIncome
              SpendingScore
Test mode:    evaluate on training data


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

Cluster 0
((((39.0:0.28813,40.0:0.28813):0.13972,6.0:0.42785):0.25331,(6.0:0.2978,13.0:
0.2978):0.38336):0.26899,(((81.0:0.15348,(77.0:0.07513,76.0:0.07513):0.07835)
:0.10341,94.0:0.25689):0.09748,72.0:0.35437):0.59578)

Cluster 1
(((3.0:0.14879,14.0:0.14879):0.0785,15.0:0.22728):0.39806,(29.0:0.3139,(5.0:0
.20413,14.0:0.20413):0.10977):0.31145)

Cluster 2
((((99.0:0.2859,98.0:0.2859):0.2257,((77.0:0.08528,79.0:0.08528):0.11592,66.0
:0.2012):0.3104):0.18939,((73.0:0.19011,(73.0:0.12501,82.0:0.12501):0.0651):0
.13337,87.0:0.32348):0.37752):0.26122,((35.0:0.14286,35.0:0.14286):0.30536,((
32.0:0.12673,31.0:0.12673):0.24022,61.0:0.36696):0.08126):0.51399)



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

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

Clustered Instances

0      10 ( 33%)
1       6 ( 20%)
2      14 ( 47%)
```

4. Visualizing with dendrogram

Now for each result, right-click on the result and press on visualize tree to view the dendrogram of each linkage.

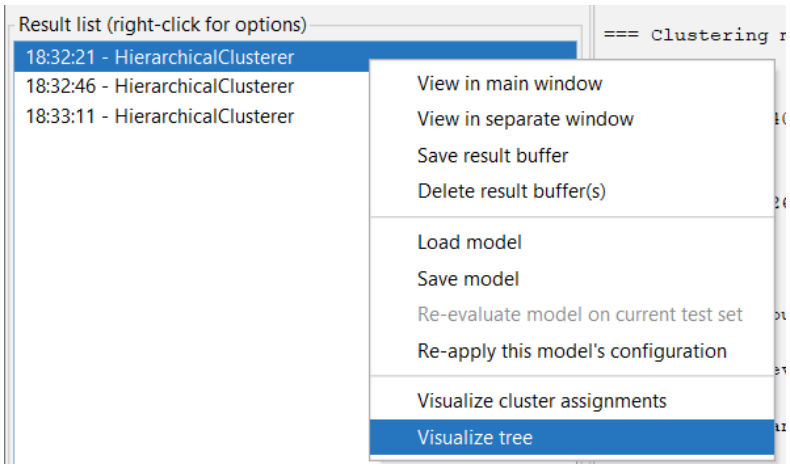


Figure 11: Option to visualize tree

Figure 12: Dendrogram for single linkage

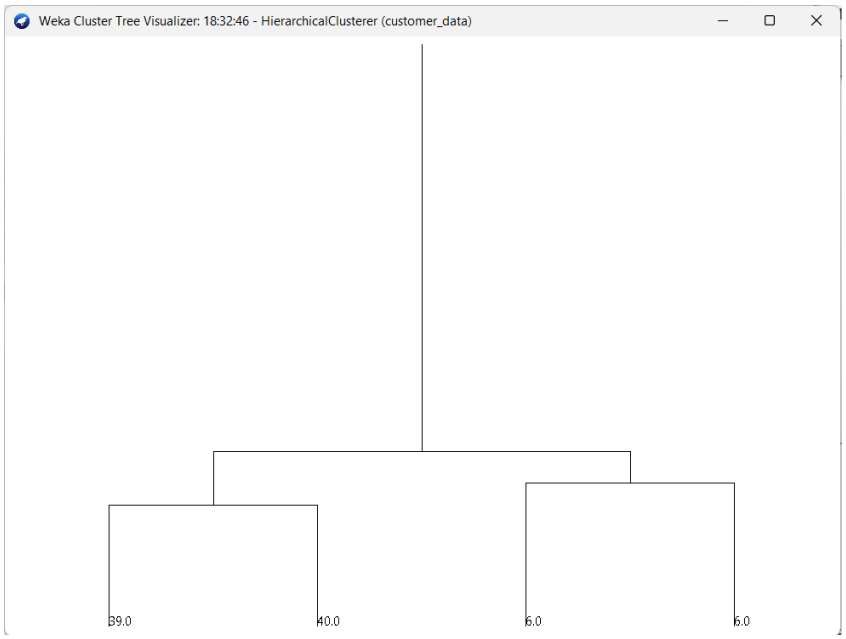


Figure 13: Dendrogram for average linkage

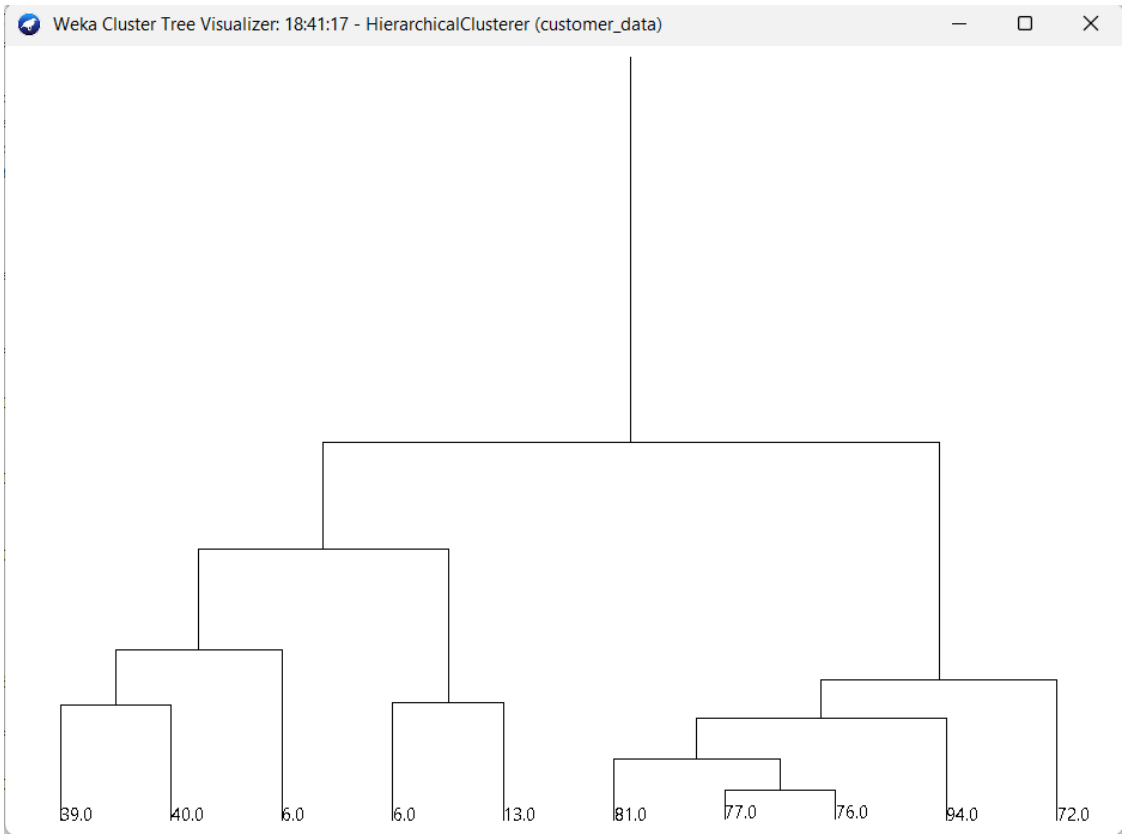


Figure 14: Dendrogram for complete linkage

Discussion

The hierarchical clustering analysis was performed using three linkage criteria: Single Linkage, Average Linkage, and Complete Linkage, each producing different cluster structures from the same customer dataset (Age, Annual Income, Spending Score).

Single Linkage produced highly uneven clusters, with one large cluster (90%), one very small cluster (7%), and one noise-like singleton cluster (3%). This reflects the well-known chaining effect of single linkage, where clusters form long, loose chains and small outliers often become single clusters.

Average Linkage generated a more balanced structure, forming three clusters of sizes 13%, 50%, and 37%. This method considers the average distance between cluster members, so it naturally avoids chaining and produces more meaningful, moderate-shaped clusters. These clusters show clearer separation of customer groups based on their combined income and spending similarities.

Complete Linkage resulted in more compact and well-separated clusters, with cluster sizes 33%, 20%, and 47%. Complete linkage tends to maximize inter-cluster distances, leading to tight clusters and clear boundaries. In this dataset, it grouped customers in a way that suggests stronger internal similarity within each cluster compared to single and average linkage.

Overall, the results show that the choice of linkage method greatly affects cluster formation. Single linkage is sensitive to noise, whereas average and complete linkage provide more stable and interpretable customer groupings. Also, across all methods, the dendrogram helps visually understand where clusters merge, how far apart they are, and which linkage method gives the most meaningful segmentation. For this dataset, the dendrograms for complete linkage display clearer grouping patterns compared to single and average linkage.

Conclusion

The hierarchical clustering done in this lab shows that the choice of linkage method strongly affects the resulting customer segments. Single linkage produced imbalanced clusters with chaining, as reflected in its elongated dendrogram structure. Average linkage generated moderately separated clusters with smoother merges, while complete linkage provided the most compact and distinct clusters, with clearly separated branches in the dendrogram. Based on the clustering patterns and dendrogram interpretability, complete linkage is the most suitable method for creating meaningful customer groups in this dataset. These well-separated clusters can be effectively used for customer segmentation and decision-making in marketing and business analysis.