Vidyavardhini’s College of Engineering and Technology

Department of Computer Engineering

Academic Year : 2023-24 (Odd Sem)

Aim: To Study & implement single linkage hierarchical clustering algorithm

Objective: Develop a program to implement Single Linkage Algorithm

Theory:

- Single linkage clustering is also called as nearest neighbour method

-The minimum distance from any object of one clusters to any object of another cluster is considered

-In Single-linkage method(A,B) is computed as,

$$D(A,B)=\min\{D(i,j)\}$$

Where,

i=object in cluster A j= Object in cluster B

Algorithm:

1. Start

2. Take objects & their measured features

3. Compute distance matrix

4. Repeat

5. Set object as the clusters.

6. If no. of cluster is not 1.

7. Then merge to closest cluster

8. Update the distance matrix

9. End if

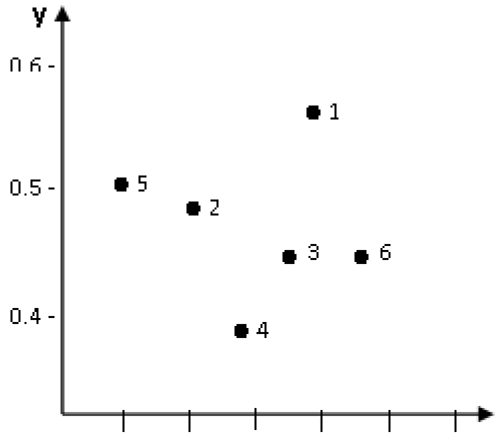
10. Else Stop

Example: Assume that the database D is given by the table below. Follow single link technique to find clusters in D. Use Euclidean distance measure.

p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Solution:

Step 1. Plot the objects in *n*-dimensional space (where *n* is the number of attributes). In our case we have 2 attributes – x and y, so we plot the objects p1, p2, … p6 in 2-dimensional space:



Step 2. Calculate the distance from each object (point) to all other points, using Euclidean distance measure, and place the numbers in a distance matrix.

We recall from the previous lecture, the formula for Euclidean distance between two points i and j is:

$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j1}|^2 + \dots + |x_{ip} - x_{jp}|^2}$$

where x_{i1} is the value of attribute 1 for i and x_{j1} is the value of attribute 1 for j, and so on, as many attributes we have … shown up to p - x_{ip} in the formula.

In our case, we only have 2 attributes. So, the Euclidean distance between our points p1 and p2,

which have attributes x and y would be calculated as follows:

$$d(p1, p2) = \sqrt{|x_{p1} - x_{p1}|^2 + |y_{p1} - y_{p2}|^2}$$

$$= \sqrt{|0.40 - 0.22|^2 + |0.53 - 0.38|^2}$$

$$= \sqrt{|0.18|^2 + |0.15|^2}$$

$$= \sqrt{0.0324 + 0.0225}$$

$$= \sqrt{0.0549}$$

$$= 0.2343$$

Analogically, we calculate the distance to the remaining points, and we will receive the following values:

Distance matrix

p1	0					
p2	0.24	0				
p3	0.22	0.15	0			
p4	0.37	0.20	0.15	0		
p5	0.34	0.14	0.28	0.29	0	
p6	0.23	0.25	0.11	0.22	0.39	0
	p1	p2	p3	p4	p5	p6

Step 3 Identify the two clusters with the shortest distance in the matrix, and merge them together.

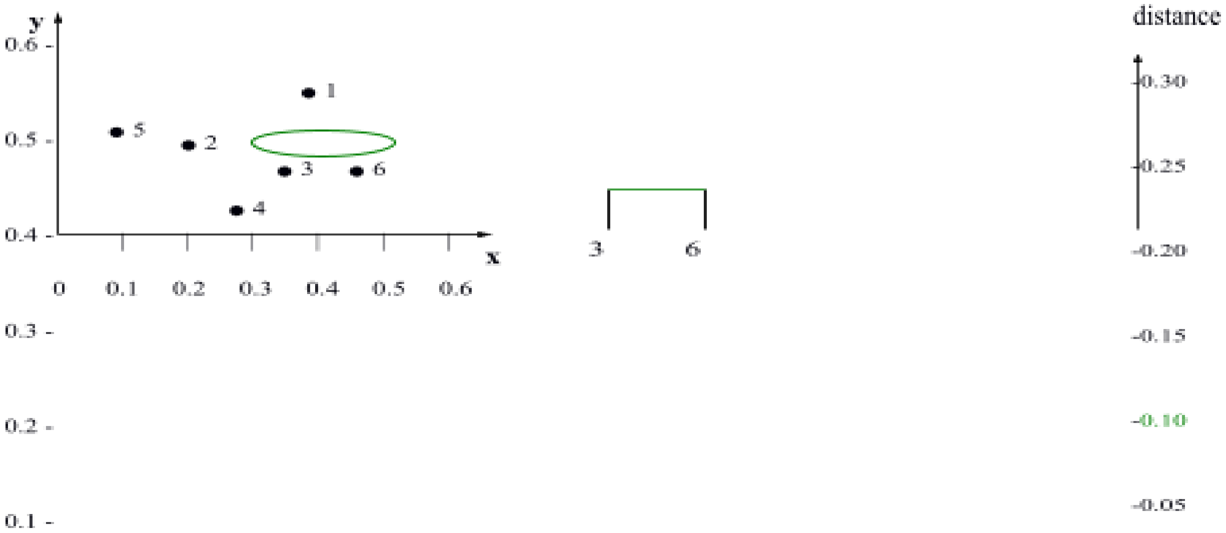
Re-compute the distance matrix, as those two clusters are now in a single cluster,

(no longer exist by themselves).

By looking at the distance matrix above, we see that p3 and p6 have the smallest distance

from all - 0.11 So, we merge those two in a single cluster, and re-compute the distance matrix.

space dendogram



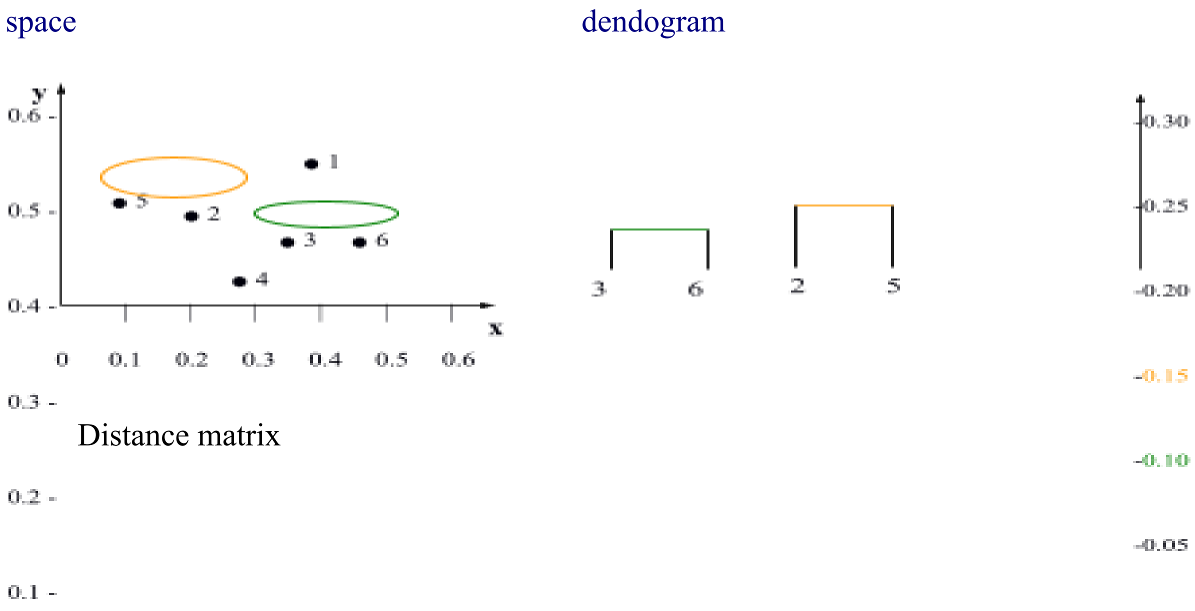
Distance matrix

Since, we have merged (p3, p6) together in a cluster, we now have one entry for (p3, p6) in the table, and no longer have p3 or p6 separately. Therefore, we need to re-compute the distance from each point to our new cluster - (p3, p6). We recall that, with the single link method the proximity of two clusters is defined as the minimum of the distance between any two points in the two clusters. Therefore, the distance between let's say (p3, p6) and p1 would be calculated as follows:

$$\begin{aligned} dist(\,(p3, p6),\ p1\,) &= \text{MIN}\,(\,dist(p3, p1)\,,\ dist(p6, p1)\,) \\ &= \text{MIN}\,(\,0.22\,,\ 0.23\,) \qquad \text{//from original matrix} \\ &= 0.22 \end{aligned}$$

<u>Step 4</u> Repeat Step 3 until all clusters are merged.

<u>a.</u> So, looking at the last distance matrix above, we see that p2 and p5 have the smallest distance from all - 0.14 So, we merge those two in a single cluster, and re-compute the distance matrix.



Distance matrix

Since, we have merged (p2, p5) together in a cluster, we now have one entry for (p2, p5) in the table, and no longer have p2 or p5 separately. Therefore, we need to re-compute the distance from all other points / clusters to our new cluster - (p2, p5). The distance between (p3, p6) and (p2, p5) would be calculated as follows:

dist((p3, p6), (p2, p5)) = MIN (*dist*(p3, p2) , *dist*(p6, p2), *dist*(p3, p5), *dist*(p6, p5))

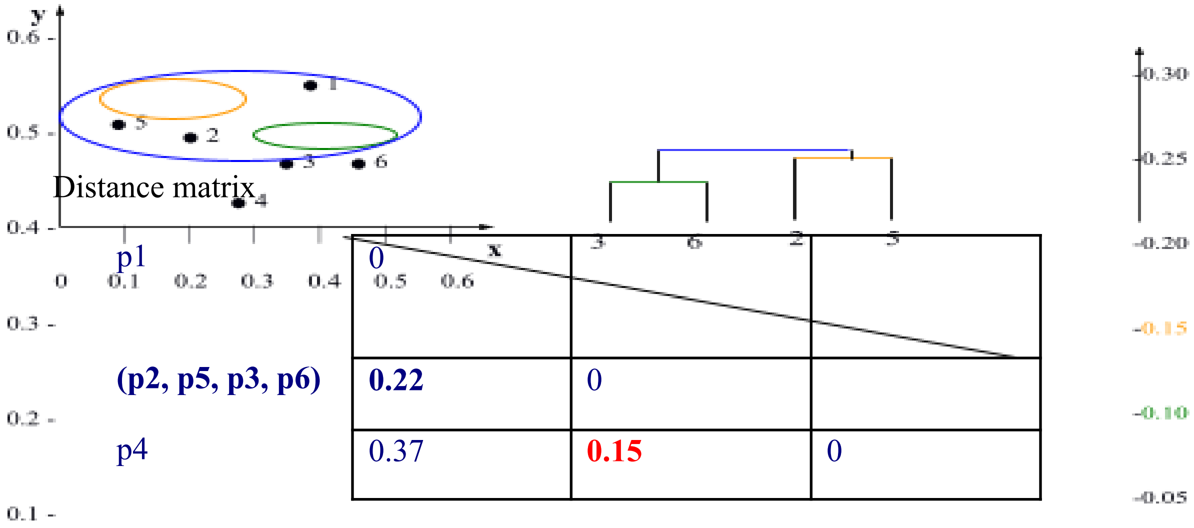
= MIN (0.15 , 0.25, 0.28, 0.39) //from original matrix

= 0.15

b. Since we have more clusters to merge, we continue to repeat Step 3.

So, looking at the last distance matrix above, we see that (p2, p5) and (p3, p6) have the smallest distance from all - 0.15 . We also notice that p4 and (p3, p6) have the same distance - 0.15 . In that case, we can pick either one. We choose (p2, p5) and (p3, p6). So, we merge those two in a single cluster, and re-compute the distance matrix.

space dendogram



Distance matrix

p1	0		
(p2, p5, p3, p6)	**0.22**	0	
p4	0.37	**0.15**	0

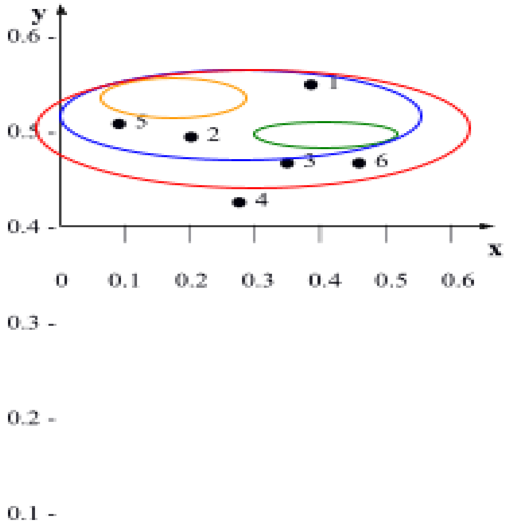
p1 **(p2, p5, p3, p6)** p4

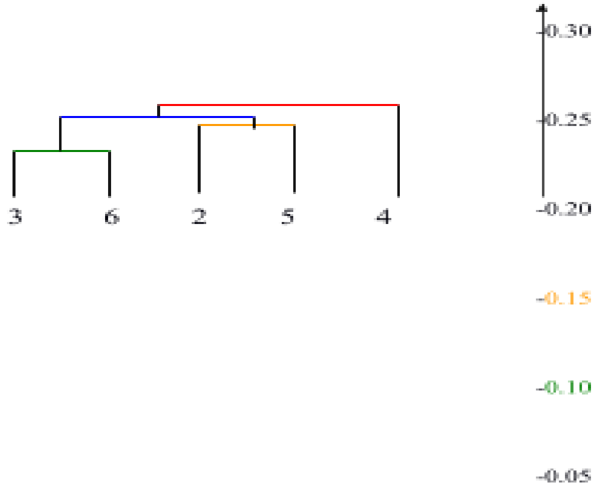
c. Since we have more clusters to merge, we continue to repeat Step 3.

So, looking at the last distance matrix above, we see that (p2, p5, p3, p6) and p4 have the smallest distance from all - 0.15 . So, we merge those two in a single cluster, and re-compute the distance matrix.

space

dendogram





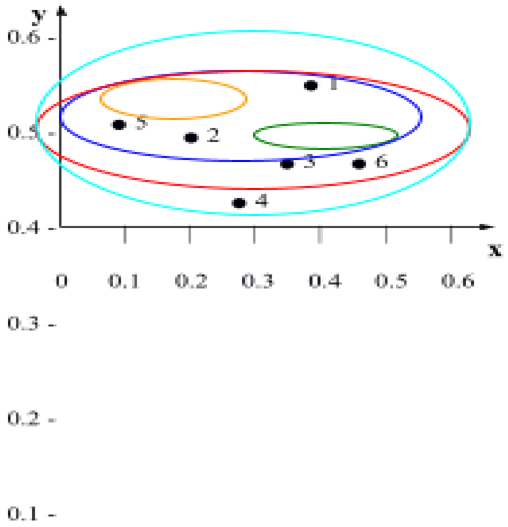
Distance matrix

	p1	**(p2, p5, p3, p6, p4)**
p1	0	
(p2, p5, p3, p6, p4)	0.22	0

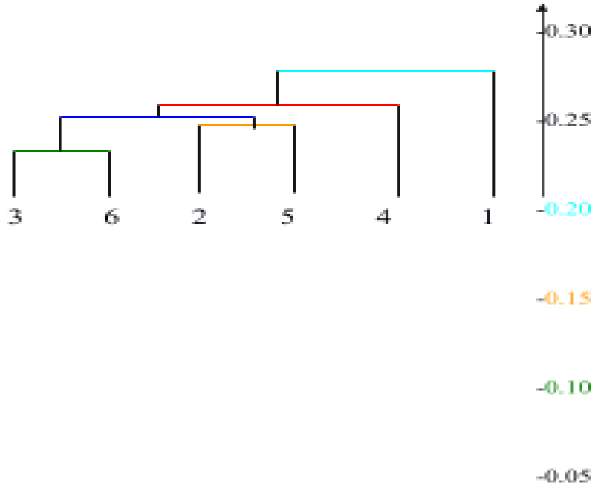
d. Since we have more clusters to merge, we continue to repeat Step 3.

So, looking at the last distance matrix above, we see that (p2, p5, p3, p6, p4) and p1 have the smallest distance - 0.22 (the only one left). So, we merge those two in a single cluster. There is no need to re-compute the distance matrix, as there are no more clusters to merge.

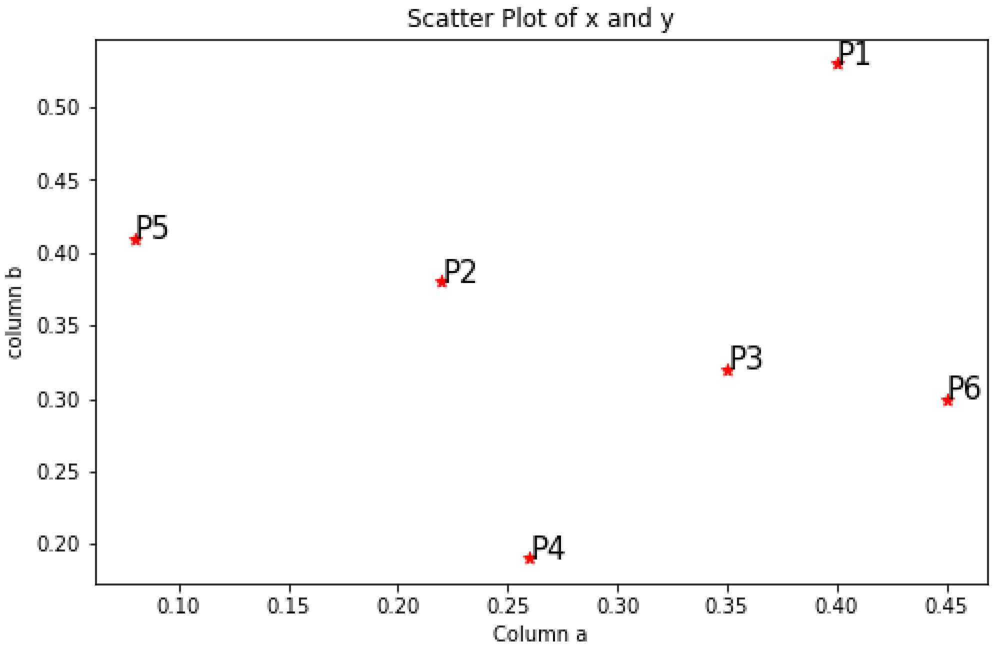
Space

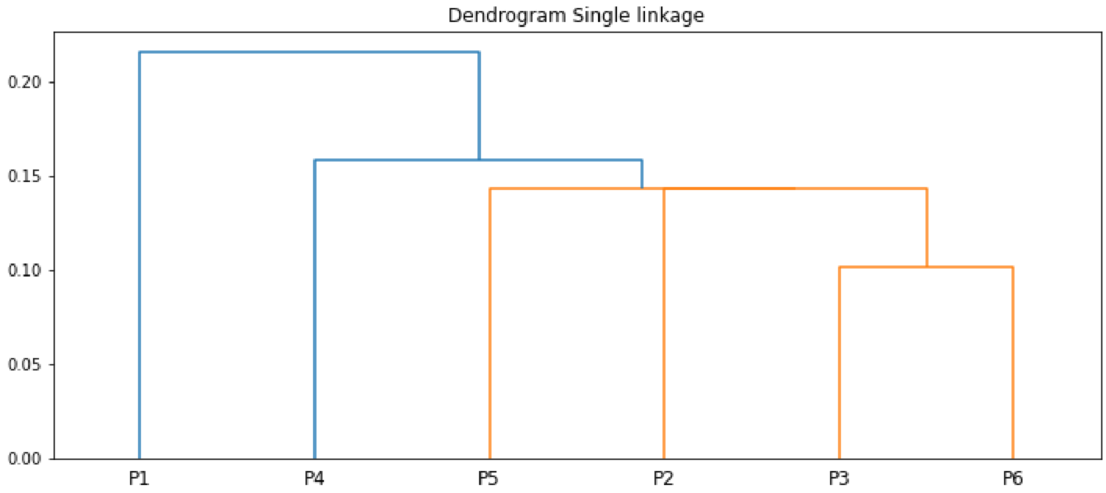


dendogram

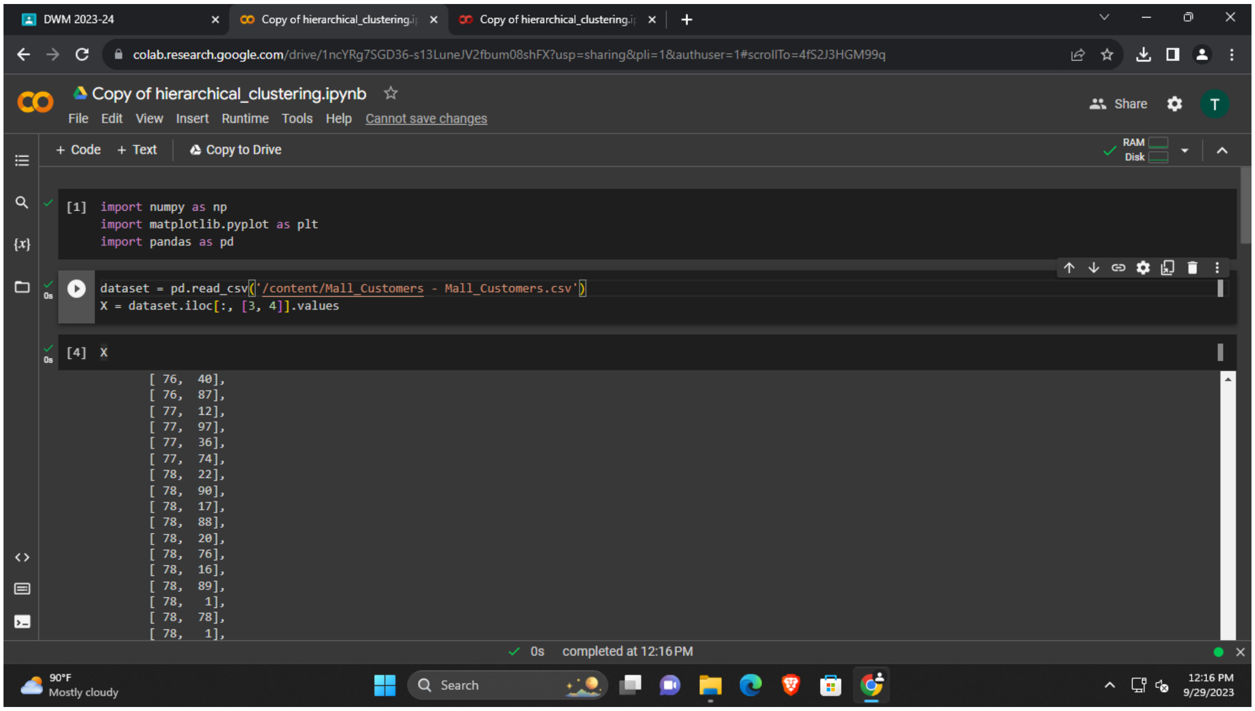


In the example above, we have merged all points into a single cluster at the end. Of course, this is not the goal of the user. The user would like the data partitioned into several clusters for unsupervised learning purposes. Therefore, the algorithm has to stop clustering at some point – either the user will specify the number of clusters he/she would like to have, or the algorithm has to make a decision on its own.



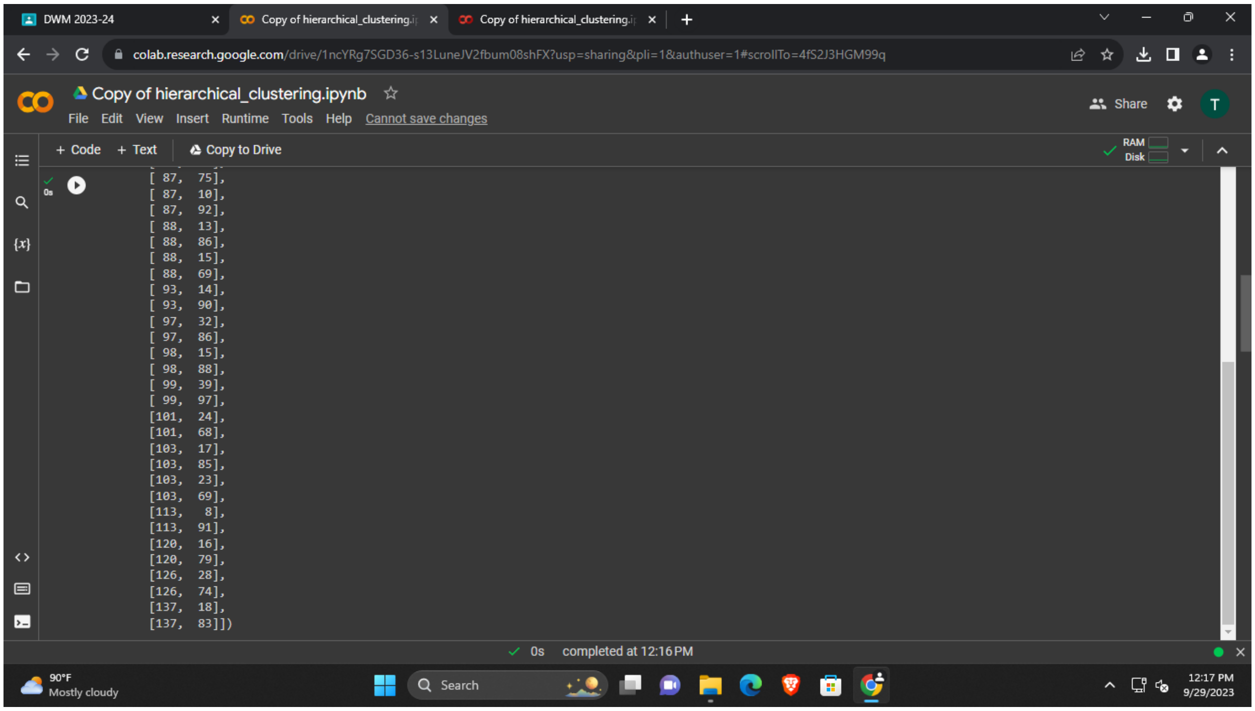


Code and output:

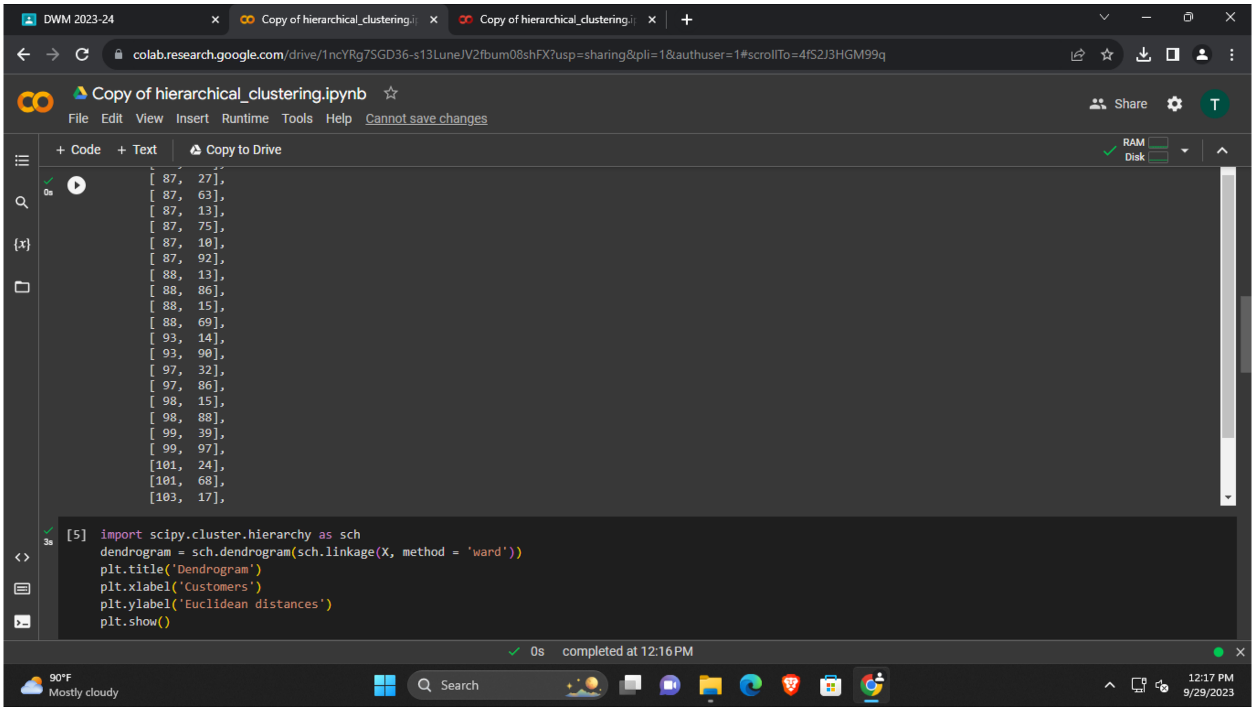

Copy of hierarchical_clustering.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive

```
[1] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd

dataset = pd.read_csv('/content/Mall_Customers - Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values

[4] X
```

```
        [ 76,  40],
        [ 76,  87],
        [ 77,  12],
        [ 77,  97],
        [ 77,  36],
        [ 77,  74],
        [ 78,  22],
        [ 78,  90],
        [ 78,  17],
        [ 78,  88],
        [ 78,  20],
        [ 78,  76],
        [ 78,  16],
        [ 78,  89],
        [ 78,   1],
        [ 78,  78],
        [ 78,   1],
```

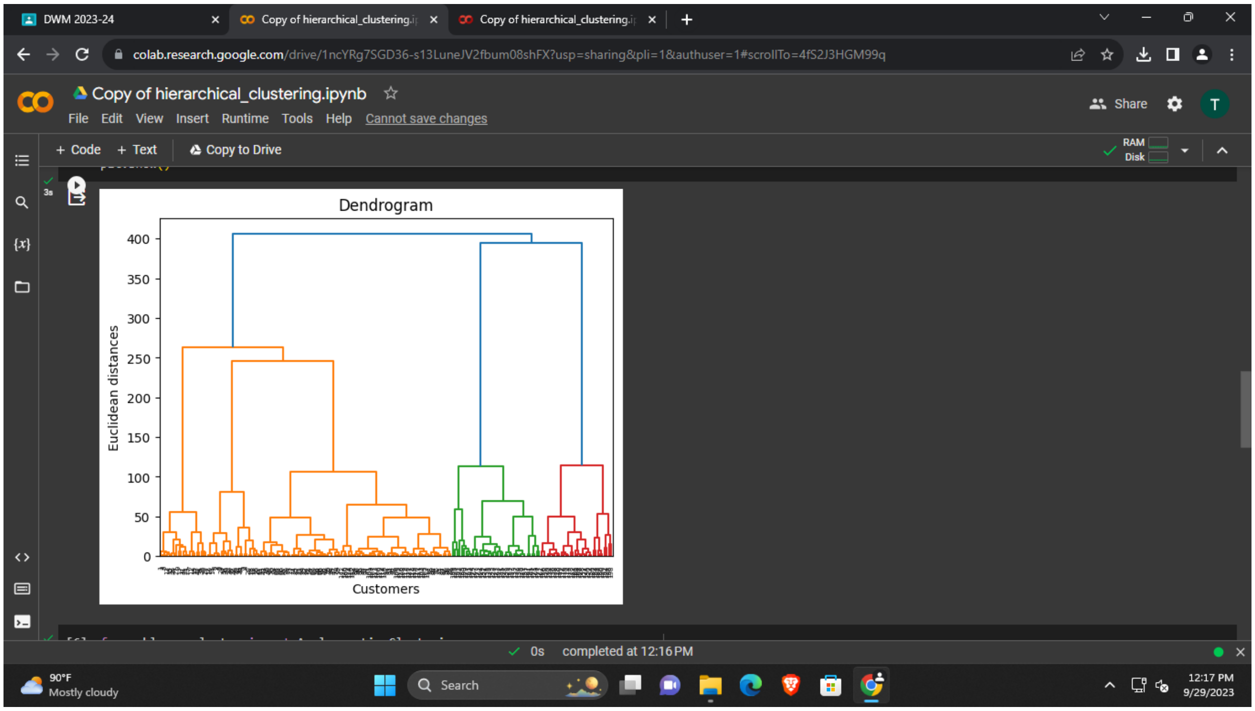

Copy of hierarchical_clustering.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive

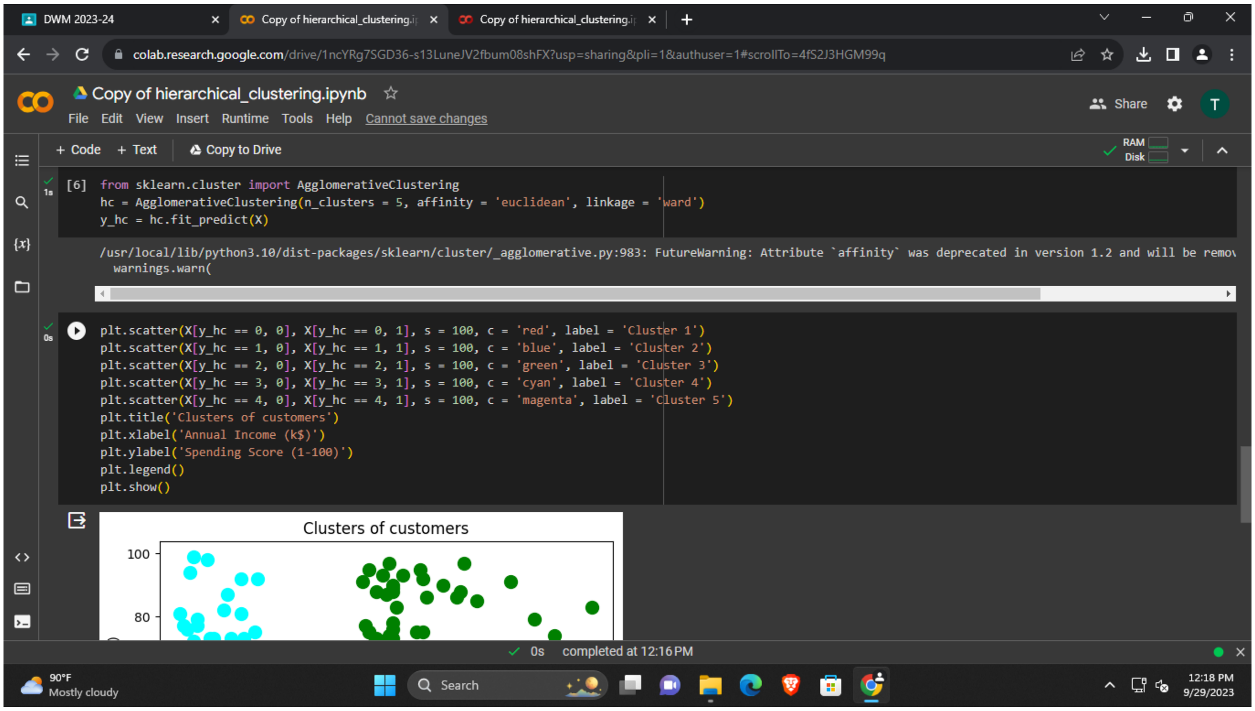
```
        [ 87,  75],
        [ 87,  10],
        [ 87,  92],
        [ 88,  13],
        [ 88,  86],
        [ 88,  15],
        [ 88,  69],
        [ 93,  14],
        [ 93,  90],
        [ 97,  32],
        [ 97,  86],
        [ 98,  15],
        [ 98,  88],
        [ 99,  39],
        [ 99,  97],
        [101,  24],
        [101,  68],
        [103,  17],
        [103,  85],
        [103,  23],
        [103,  69],
        [113,   8],
        [113,  91],
        [120,  16],
        [120,  79],
        [126,  28],
        [126,  74],
        [137,  18],
        [137,  83]])
```


DWM 2023-24
Copy of hierarchical_clustering.ip
Copy of hierarchical_clustering.ip
colab.research.google.com/drive/1ncYRg7SGD36-s13LuneJV2fbum08shFX?usp=sharing&pli=1&authuser=1#scrollTo=4fS2J3HGM99q
Copy of hierarchical_clustering.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM Disk
0s completed at 12:16 PM
90°F Mostly cloudy
Search
12:17 PM 9/29/2023

```
       [ 87,  27],
       [ 87,  63],
       [ 87,  13],
       [ 87,  75],
       [ 87,  10],
       [ 87,  92],
       [ 88,  13],
       [ 88,  86],
       [ 88,  15],
       [ 88,  69],
       [ 93,  14],
       [ 93,  90],
       [ 97,  32],
       [ 97,  86],
       [ 98,  15],
       [ 98,  88],
       [ 99,  39],
       [ 99,  97],
       [101,  24],
       [101,  68],
       [103,  17],
```

```
[5] import scipy.cluster.hierarchy as sch
    dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
    plt.title('Dendrogram')
    plt.xlabel('Customers')
    plt.ylabel('Euclidean distances')
    plt.show()
```

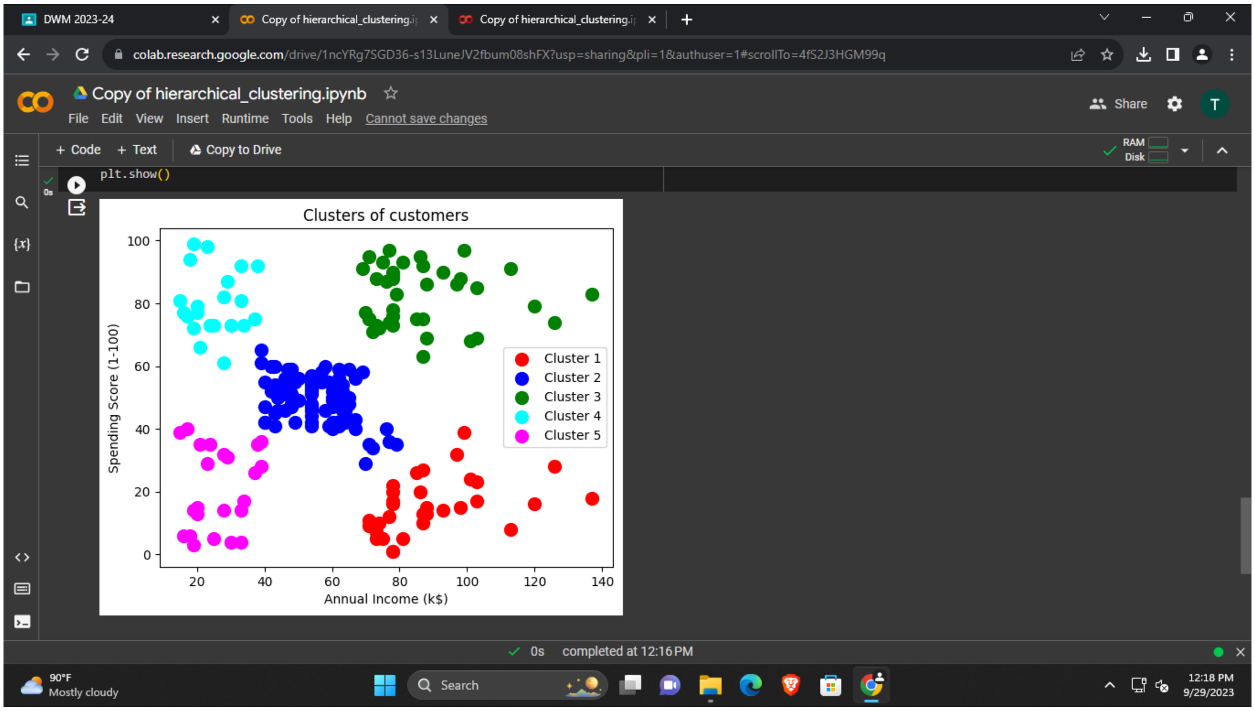

DWM 2023-24
Copy of hierarchical_clustering.ip
Copy of hierarchical_clustering.ip
colab.research.google.com/drive/1ncYRg7SGD36-s13LuneJV2fbum08shFX?usp=sharing&pli=1&authuser=1#scrollTo=4fS2J3HGM99q
Copy of hierarchical_clustering.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM Disk
Dendrogram
Euclidean distances
400
350
300
250
200
150
100
50
0
Customers
0s completed at 12:16 PM
90°F Mostly cloudy
Search
12:17 PM 9/29/2023



DWM 2023-24
Copy of hierarchical_clustering.ip
Copy of hierarchical_clustering.ip
colab.research.google.com/drive/1ncYRg7SGD36-s13LuneJV2fbum08shFX?usp=sharing&pli=1&authuser=1#scrollTo=4fS2J3HGM99q
Copy of hierarchical_clustering.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text
Copy to Drive
RAM
Disk
Clusters of customers
100
80
completed at 12:16 PM
90°F
Mostly cloudy
Search
12:18 PM
9/29/2023

```
[6] from sklearn.cluster import AgglomerativeClustering
    hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
    y_hc = hc.fit_predict(X)

    /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_agglomerative.py:983: FutureWarning: Attribute `affinity` was deprecated in version 1.2 and will be remov
      warnings.warn(
```

```
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



DWM 2023-24
Copy of hierarchical_clustering.ip
Copy of hierarchical_clustering.ip
colab.research.google.com/drive/1ncYRg7SGD36-s13LuneJV2fbum08shFX?usp=sharing&pli=1&authuser=1#scrollTo=4fS2J3HGM99q
Copy of hierarchical_clustering.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text
Copy to Drive
RAM
Disk
Clusters of customers
Spending Score (1-100)
100
80
60
40
20
0
20
40
60
80
100
120
140
Annual Income (k$)
● Cluster 1
● Cluster 2
● Cluster 3
● Cluster 4
● Cluster 5
completed at 12:16 PM
90°F
Mostly cloudy
Search
12:18 PM
9/29/2023

```
plt.show()
```

Conclusion: The quality and interpretation of clusters formed after performing a clustering algorithm can vary widely depending on the data, the algorithm used, and the chosen parameters. The choice of clustering algorithm can significantly impact the resulting clusters. Common algorithms include K-Means, Hierarchical Clustering, DBSCAN, and Gaussian Mixture Models. Each algorithm has its own assumptions and limitations, so the choice should be appropriate for the data and the problem at hand. Ideally, clusters should be well-separated and distinct from each other. A good clustering result will have tight and cohesive clusters with minimal overlap. You can assess cluster separation using metrics like silhouette score, Davies-Bouldin index, or visual inspection of cluster plots.Visualizations like scatter plots, heatmaps, and dendrograms can provide valuable insights into the clustering structure. Visual inspection can help identify any issues or patterns that may not be apparent through numerical metrics alone. Clustering is often an iterative process. If the initial clustering results are not satisfactory, consider refining the process by adjusting parameters, trying different algorithms, or preprocessing the data differently.In summary, when commenting on the clusters formed after performing clustering algorithms, it's essential to assess the quality, interpretability, and appropriateness of the results in the context of your specific problem. A combination of quantitative metrics and visual exploration can provide a comprehensive understanding of the clustering outcome.