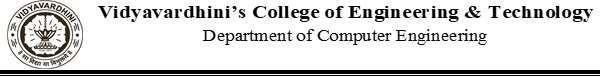


| Experiment No. 6 |
| --- |
| Apply Hierarchical Clustering algorithm on the Wholesale Customers Dataset |
| Date of Performance: 03/09/2024 |
| Date of Submission: 17/09/2024 |

Aim: Apply Hierarchical Clustering algorithm on the Wholesale Customers Dataset

Objective: Able to perform various feature engineering tasks, apply Hierarchical Clustering Algorithm on the given dataset.

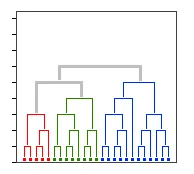
Theory:

It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

Bottom-up: Initially, each point is a cluster, Repeatedly combine the two “nearest” clusters into one.

Top-Down:Start with one cluster and recursively split it, Used for clustering similar things join and make a hierarchical clustering.



Agglomerative with Dendogram:-

Consider only one part (lower triangle part)

Now find the minimum distance

Minimum distance = 1 for E & A

Hence merge EA

So

Now Find the minimum distance from EA to other

C = min [ dist { (E, A), C} ]

= min [ dist(E, C) , dist(A, C) ]

= min [2, 2]

= 2

B = min [ dist { (E, A), B} ]

= min [ dist(E, B) , dist(A, C) ]

= min [2, 5]

= 2

D = min [ dist { (E, A), D} ]

= min [ dist(E, D) , dist(A, D) ]

= min [3, 3]

= 3

In this minimum distance = 1

So for CB

Now again find the minimum distance from CB to another point as we find above So

EA = min [ dist { (E, A), (C, B} ]

= min [ dist { (E, C) , (E, B) , (A, C), (A, B) } ]

= min dist[2, 2, 2, 5]

= 2

D = min [ dist { (C, B), D} ]

= min [ dist{(C, D) , (B, D)} ]

= min dist[6, 3]

= 3

Now minum distance = 2 for EA , and CB from above table

D = min(dist [ ( E, A , C, B ) , D ]

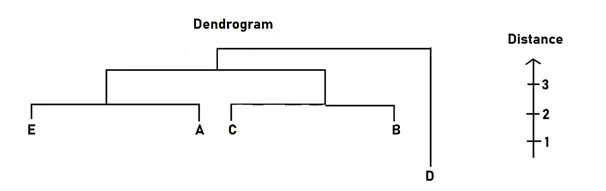
= min dist [ (E, D), (A, D). (C, D) , (B, D)

= min dist [3, 3, 6, 3]

= 3

So

So finally are dendogram are final



Dataset:

This data set refers to clients of a wholesale distributor. It includes the annual spending in monetary units (m.u.) on diverse product categories. The wholesale distributor operating in different regions of Portugal has information on annual spending of several items in their stores across different regions and channels. The dataset consist of 440 large retailers annual spending on 6 different varieties of product in 3 different regions (lisbon , oporto, other) and across different sales channel ( Hotel, channel) Detailed overview of dataset

Records in the dataset = 440 ROWS Columns in the dataset = 8 COLUMNS

FRESH: annual spending (m.u.) on fresh products (Continuous) MILK:- annual spending (m.u.) on milk products (Continuous) GROCERY:- annual spending (m.u.) on grocery products (Continuous) FROZEN:- annual spending (m.u.) on frozen products (Continuous)

DETERGENTS\_PAPER :- annual spending (m.u.) on detergents and paper products (Continuous)

DELICATESSEN:- annual spending (m.u.)on and delicatessen products (Continuous); CHANNEL: - sales channel Hotel and Retailer

REGION:- three regions ( Lisbon, Oporto, Other)

Code:

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import normalize

import scipy.cluster.hierarchy as shc

from sklearn.cluster import AgglomerativeClustering

data = pd.read\_csv('/content/Wholesale customers data.csv')

print(data.head())

#normalize data so the scale of each varirable is same

#if not done, model might become biased towards variables with higher magnitude (in this case fresh or milk)

data\_scaled = normalize(data)

data\_scaled = pd.DataFrame(data\_scaled, columns=data.columns)

print(data\_scaled.head())

#similar scales

#Dendogram to decide the number of clusters

plt.figure(figsize=(10, 7))

plt.title("Dendrograms")

d = shc.dendrogram(shc.linkage(data\_scaled, method='ward'))

#x=samples, y=distance between samples. threshold=6

plt.figure(figsize=(10, 7))

plt.title("Dendrograms")

d = shc.dendrogram(shc.linkage(data\_scaled, method='ward'))

plt.axhline(y=6, color='r', linestyle='--')

#line divides forming 2 clusters

#apply hierarchical clustering for 2 clusters

cluster = AgglomerativeClustering(n\_clusters=2, metric='euclidean', linkage='ward')

print(cluster.fit\_predict(data\_scaled))

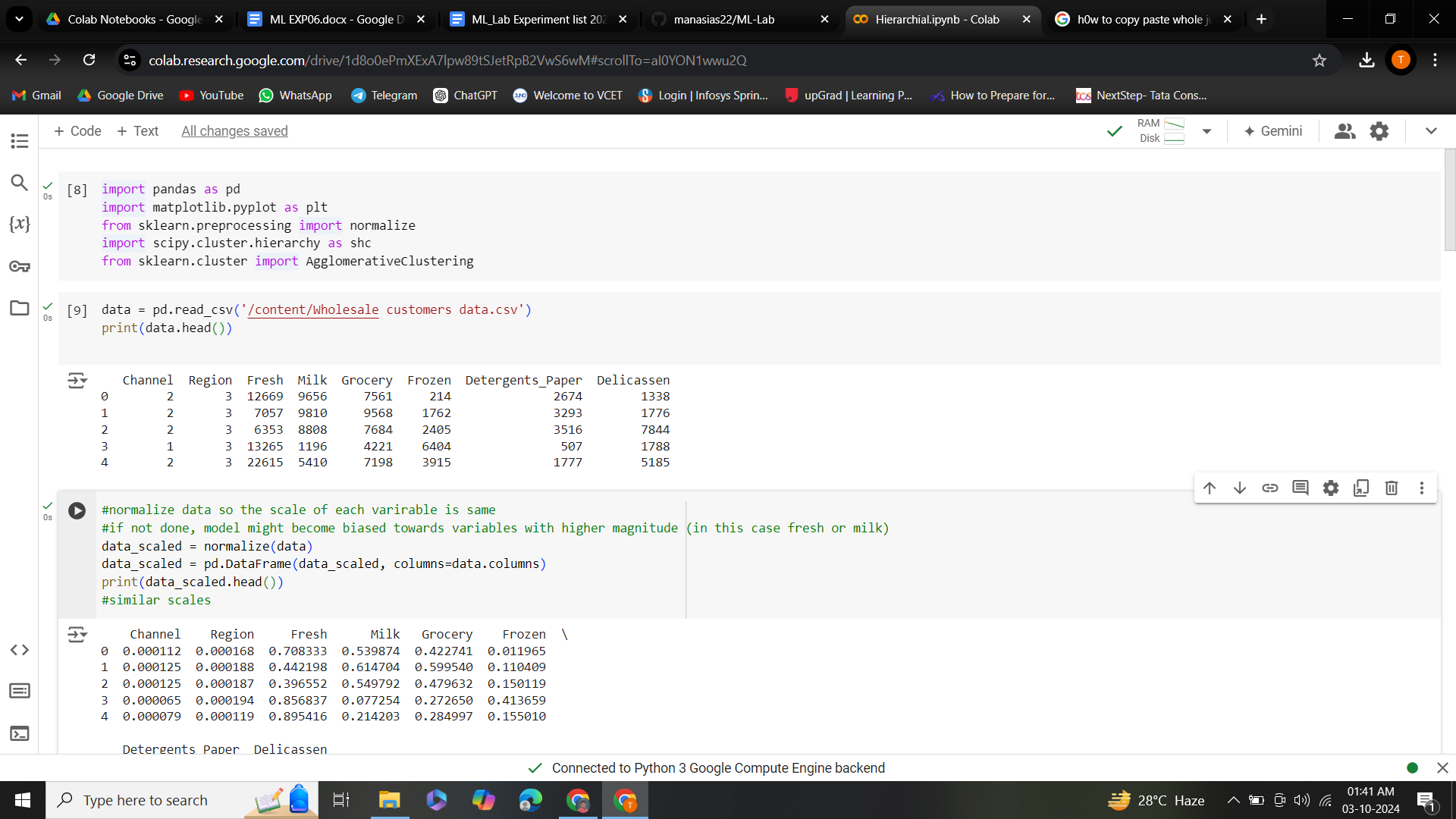
#0=cluster 1, 1=cluster 2

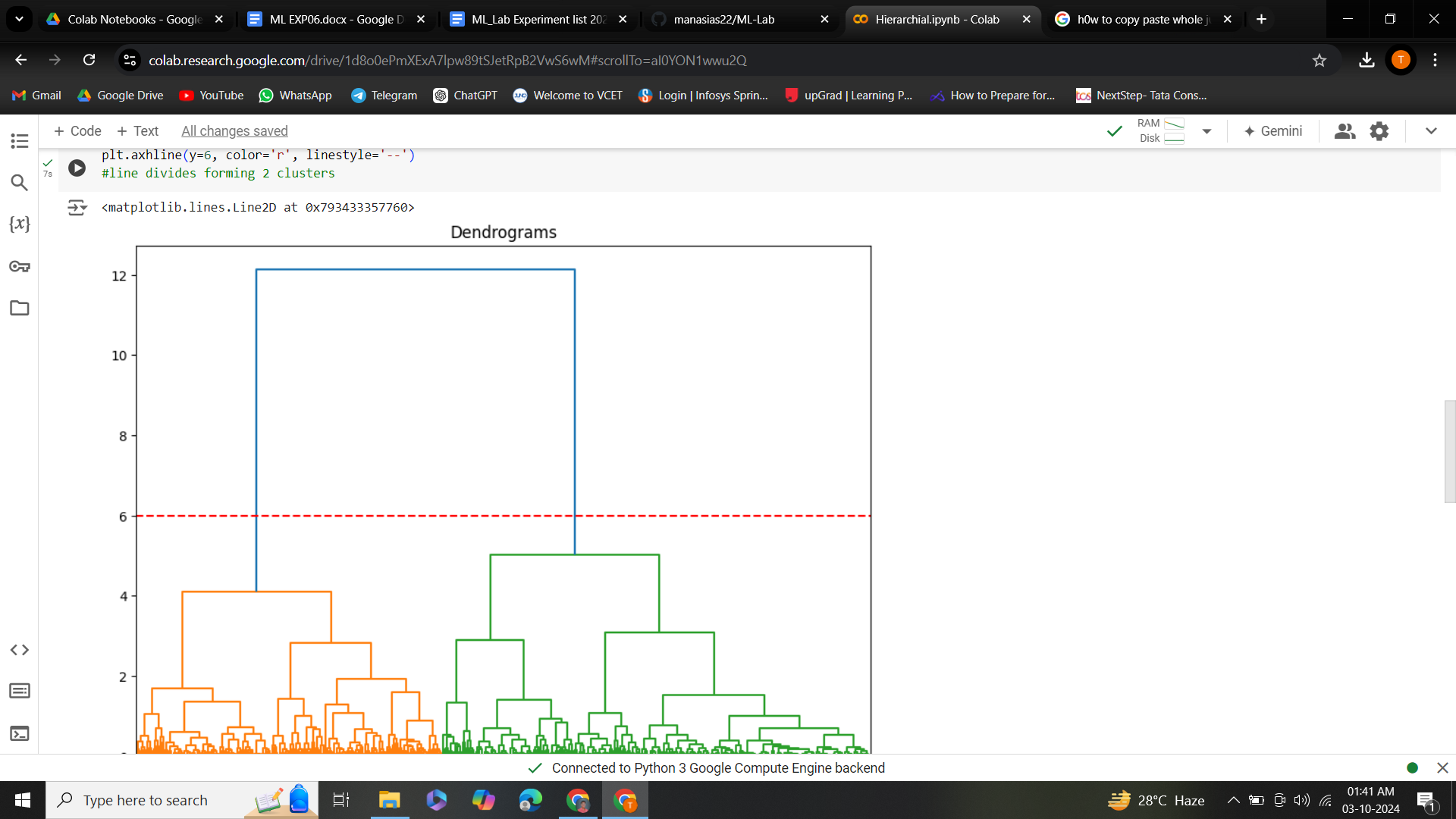
#visualize clusters

plt.figure(figsize=(10, 7))

plt.scatter(data\_scaled['Milk'], data\_scaled['Grocery'], c=cluster.labels\_)

Output:







Conclusion:

In conclusion, the practical implementation of the Hierarchical Clustering algorithm demonstrates its effectiveness in grouping similar data points based on their features. By building a hierarchy of clusters, this method allows for a nuanced understanding of data structure, revealing relationships that may not be apparent with other clustering techniques.

Throughout the process, we observed the impact of different linkage criteria (single, complete, average) on the clustering outcome, highlighting the importance of choosing the right method for the specific dataset and analysis goals. The dendrogram visualization further aids in interpreting the results, providing a clear representation of how clusters are formed and allowing for the identification of an optimal number of clusters based on domain knowledge or desired granularity.

While Hierarchical Clustering is powerful for exploratory data analysis, it is essential to consider its computational complexity, especially with larger datasets. This limitation can often be mitigated by employing techniques such as sampling or dimensionality reduction prior to clustering.

Overall, the practical application of Hierarchical Clustering not only enhances our understanding of data patterns but also serves as a valuable tool for various fields, including bioinformatics, marketing, and social sciences. Future work may involve integrating this method with other algorithms to improve clustering performance and adaptability across different contexts.