Experiment 7: Understanding Clustering - II

Objective: Develop an understanding of how to perform hierarchical clustering on a data set.
Time Required: 3 hrs

Programming Language: Python

Software Required: Anaconda

Introduction

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other. Hierarchical clustering is further divided into two types: Agglomerative and Divisive. In this lab, we ll be applying agglomerative clustering in which objects are grouped in clusters based on their similarity. 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*.

Task:

You have to solve the wholesale customer segmentation problem using hierarchical clustering. You can download the dataset using **this link**. The data is hosted on the UCI Machine Learning repository. The aim of this problem is to segment the clients of a wholesale distributor based on their annual spending on diverse product categories, like milk, grocery, region, etc.

Steps to follow:

1. Import the important libraries
2. Load and view the dataset
3. Normalize the data so that the scale of each variable is the same. If the scale of the variables is not the same, the model might become biased towards the variables with a higher magnitude like Fresh or Milk. To normalize the data, you can use the following code:

```
#Normalize data
from sklearn.preprocessing import normalize
data_scaled = normalize(data)
data_scaled = pd.DataFrame(data_scaled, columns=data.columns)
data_scaled.head()
```

4. Draw the dendrogram to help you decide the number of clusters for this particular problem. You can use the following code for it:

```
#Draw dendogram
import scipy.cluster.hierarchy as sch
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
```

```
dend = sch.dendrogram(sch.linkage(data_scaled, method='ward'))
```

After drawing the dendogram, you will see that the x-axis contains the samples and y-axis represents the distance between these samples. The vertical line with maximum distance is the blue line and hence you can decide a threshold of 6 and cut the dendrogram with the following code:

```
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
plt.axhline(y=6, color='r', linestyle='--')
```

After running the above code, you will get two clusters as this line cuts the dendrogram at two points.

5. Apply hierarchical clustering for 2 clusters. You can use the following code for it

```
#Apply hierarchical Clustering
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage='ward')
cluster.fit_predict(data_scaled)
```

After executing the above code, you will see the values of 0s and 1s in the output since you defined 2 clusters. 0 represents the points that belong to the first cluster and 1 represents points in the second cluster.

6. Plot the clusters to visualize them by using following code:

```
#Plotting clusters
plt.figure(figsize=(10, 7))
plt.scatter(data_scaled['Milk'], data_scaled['Grocery'], c=cluster.labels_)
```