**Experiment 7**

**Shashwat Tripathi**

**D15A Batch C  
RollNo: 64**

**AIM : To implement different clustering algorithms.**

**THEORY :**

**a) Clustering algorithm for unsupervised classification (K-means, density based**

**(DBSCAN), Hierarchical clustering)**

Clustering algorithms are unsupervised machine learning techniques used for grouping data points into clusters based on their similarities. These algorithms are widely used in various domains such as data mining, pattern recognition, and image analysis. Here, I'll provide a brief overview of three popular clustering algorithms: K-means, Density-based Spatial Clustering of Applications with Noise (DBSCAN), and Hierarchical clustering.

1. K-means Clustering:

- K-means is one of the simplest and most commonly used clustering algorithms.

- It partitions the dataset into a predetermined number of clusters (K) by minimizing the sum of squared distances between data points and their respective cluster centroids.

- The algorithm starts by randomly initializing K cluster centroids and then iteratively assigns each data point to the nearest centroid and updates the centroids' positions until convergence.

- K-means is sensitive to the initial placement of centroids and may converge to local optima, so multiple initializations and techniques like K-means++ initialization can be employed to mitigate this issue.

2. Density-based Spatial Clustering of Applications with Noise (DBSCAN):

- DBSCAN is a density-based clustering algorithm that can discover clusters of arbitrary shapes and sizes.

- It defines clusters as dense regions of data points separated by regions of lower density.

- DBSCAN requires two parameters: epsilon (ε), which specifies the radius within which to search for neighboring points, and minPts, the minimum number of points required to form a dense region (cluster).

- The algorithm starts by randomly selecting a data point and expanding the cluster by adding neighboring points that satisfy the density criteria. It continues this process recursively until no more points can be added to the cluster, then searches for new points to form additional clusters.

- DBSCAN is robust to noise and outliers and does not require the number of clusters to be specified in advance.

3. Hierarchical Clustering:

- Hierarchical clustering builds a hierarchy of clusters by recursively merging or splitting clusters based on their similarity.

- It can be agglomerative (bottom-up) or divisive (top-down).

- In agglomerative hierarchical clustering, each data point starts as a singleton cluster, and pairs of clusters are iteratively merged based on a specified distance metric until all points belong to a single cluster.

- Divisive hierarchical clustering starts with all data points in one cluster and recursively splits the cluster into smaller clusters until each data point forms its own cluster.

- Hierarchical clustering produces a dendrogram that visualizes the hierarchy of clusters, allowing users to choose the optimal number of clusters based on their domain knowledge and requirements.

Each clustering algorithm has its advantages, limitations, and suitability for different types of datasets and applications. Experimentation and evaluation are essential to selecting the most appropriate algorithm and tuning its parameters for a given clustering task.

**b) Plot the cluster data and show mathematical steps.**

To plot cluster data and show the mathematical steps, let's take the example of K-means clustering, one of the most widely used clustering algorithms. Here's a step-by-step guide:

1. Data Preparation

- First, you need a dataset consisting of data points. For demonstration purposes, let's assume we have a simple dataset with two-dimensional points.

2. Initialization

- Choose the number of clusters (K) you want to identify. Initialize K cluster centroids randomly within the range of the dataset.

3. Assign Points to Clusters

- Calculate the Euclidean distance between each data point and each cluster centroid.

- Assign each data point to the cluster with the nearest centroid.

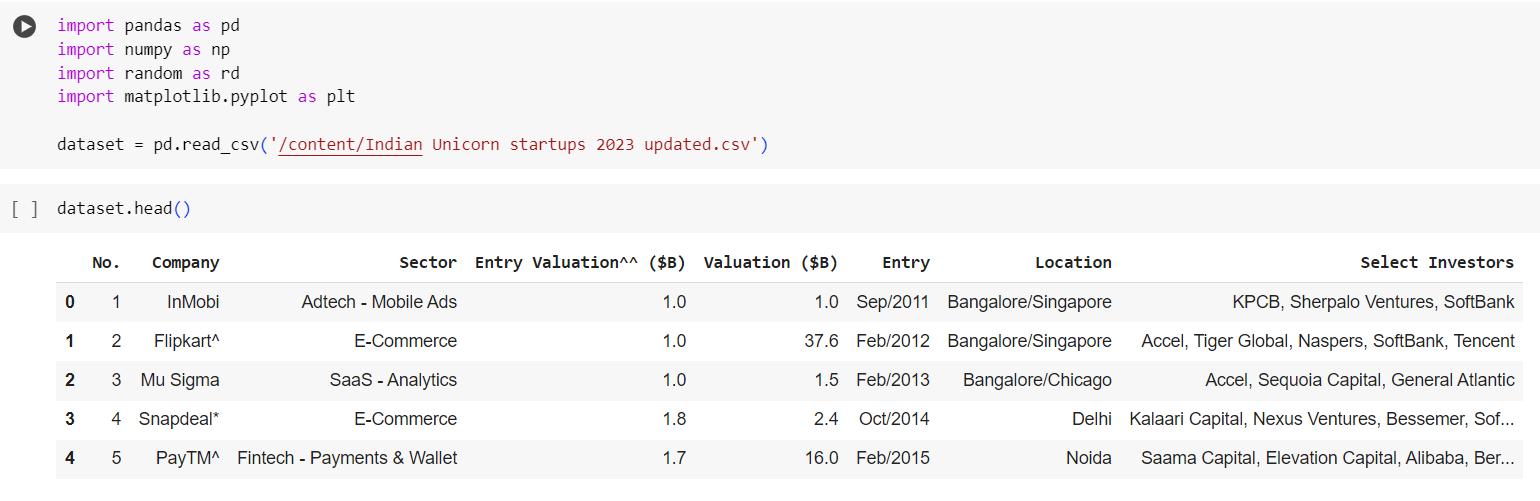
4. Update Cluster Centroids

- Calculate the mean of all data points assigned to each cluster. This becomes the new centroid of that cluster.

5. Repeat

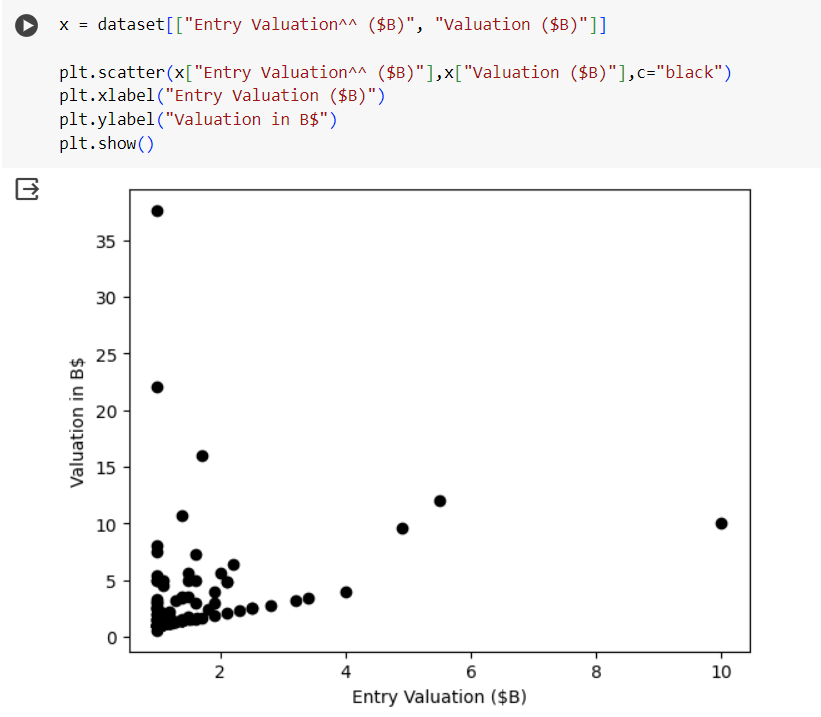
- Repeat steps 3 and 4 until convergence, which occurs when the centroids no longer change significantly or after a fixed number of iterations.

**OUTPUT : Loading the Dataset**

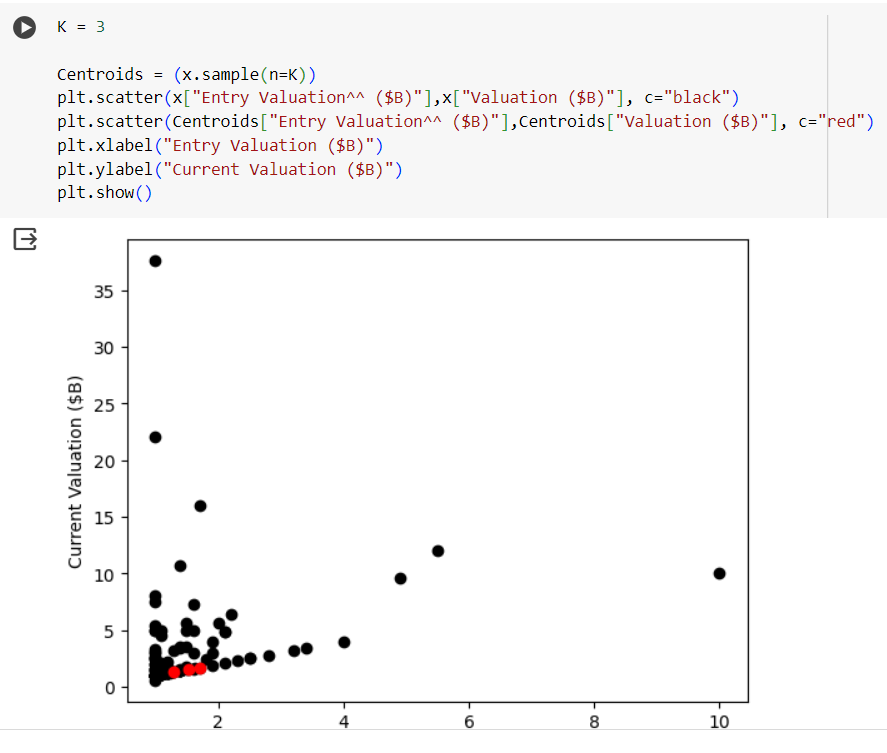


**1. K Means**

**Selecting the attributes to plot a scatter plot**

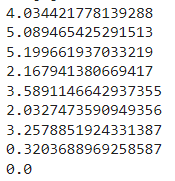


**Defining the number of clusters(k) and selecting the centroid of the cluster**

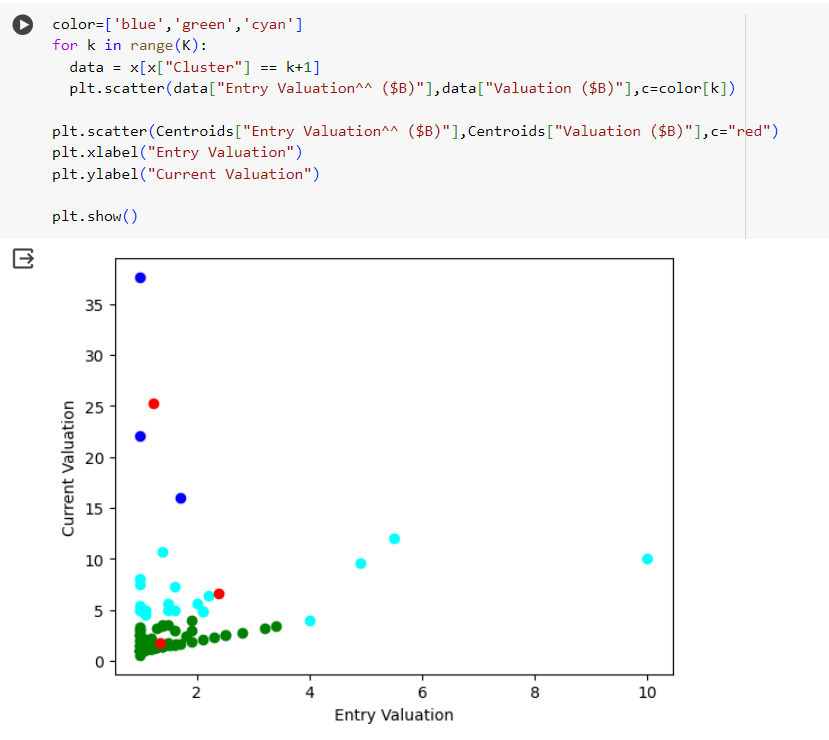


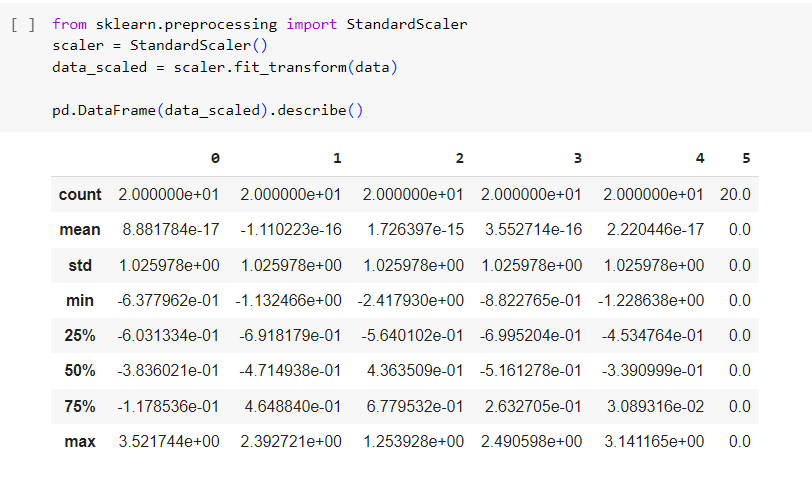
**Assigning each point to the closest cluster and updating the centroid of the cluster**



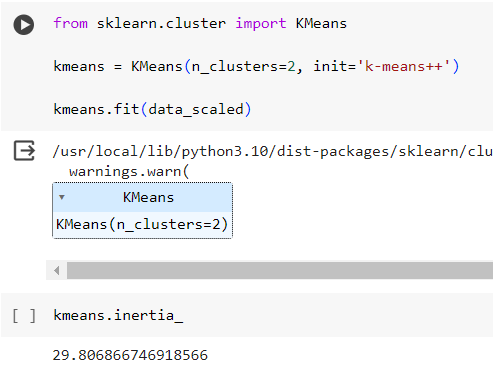


**Visualizing the 3 different clusters as mentioned above with their respective centroids**

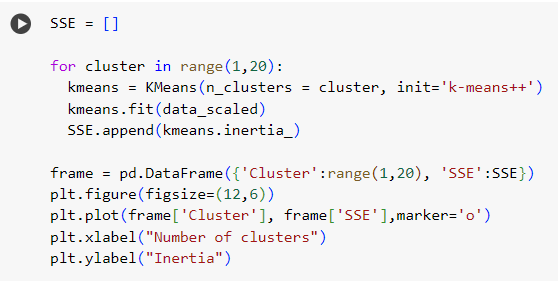


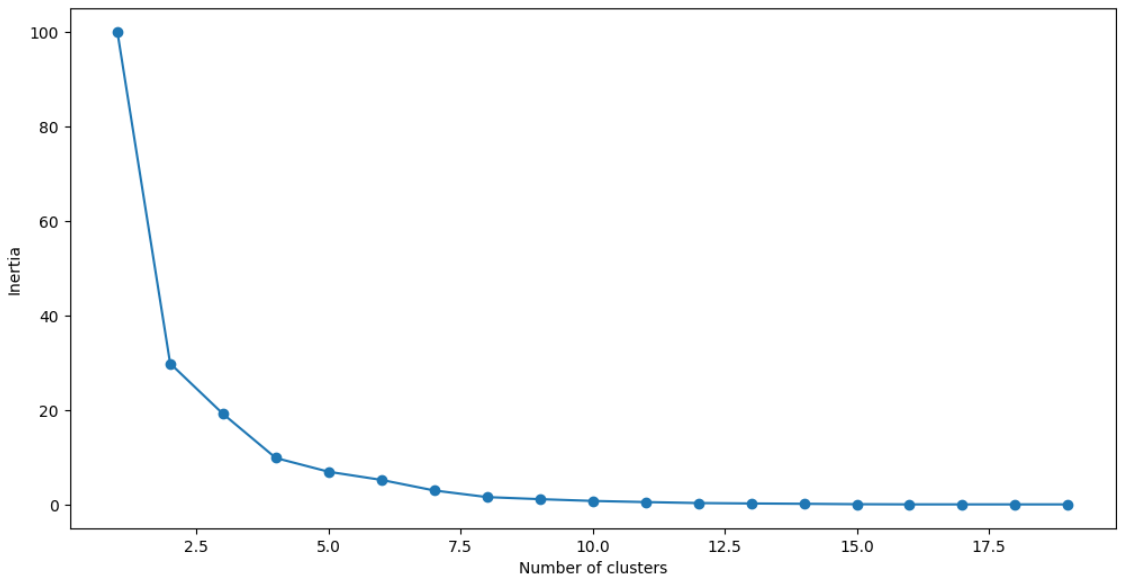
**Standardizing the data**

**Applying K-Means Algorithm on the above data and finding inertia on the fitted data**

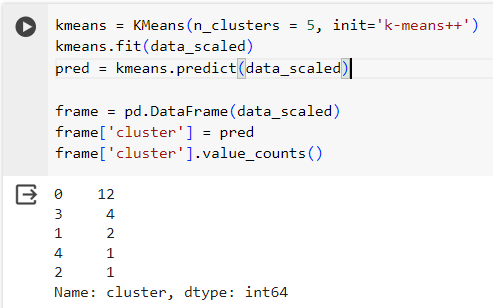


**Storing multiple K-Means algorithm in a list and converting it into dataframe for plotting them**





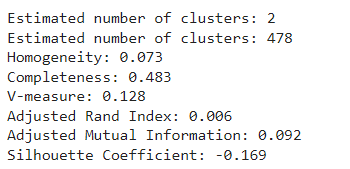
**Finding K-Means for 5 different clusters**

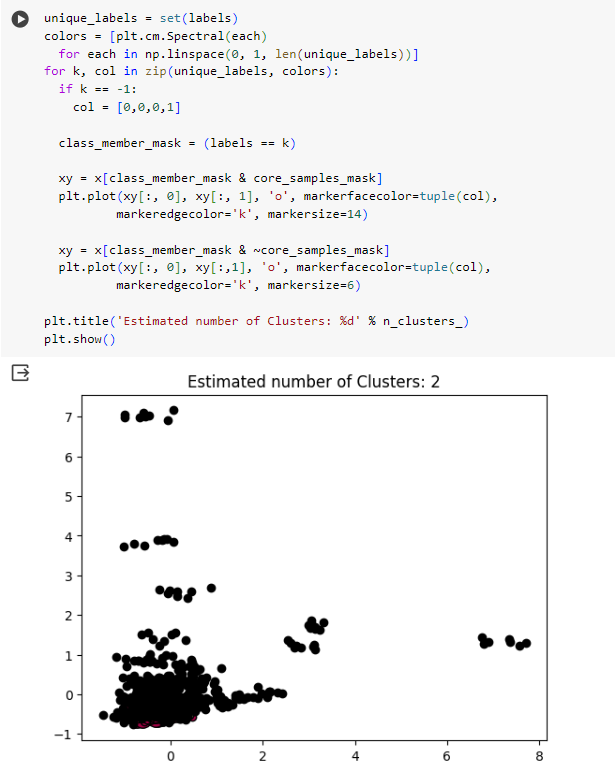


**2. DBSCAN**

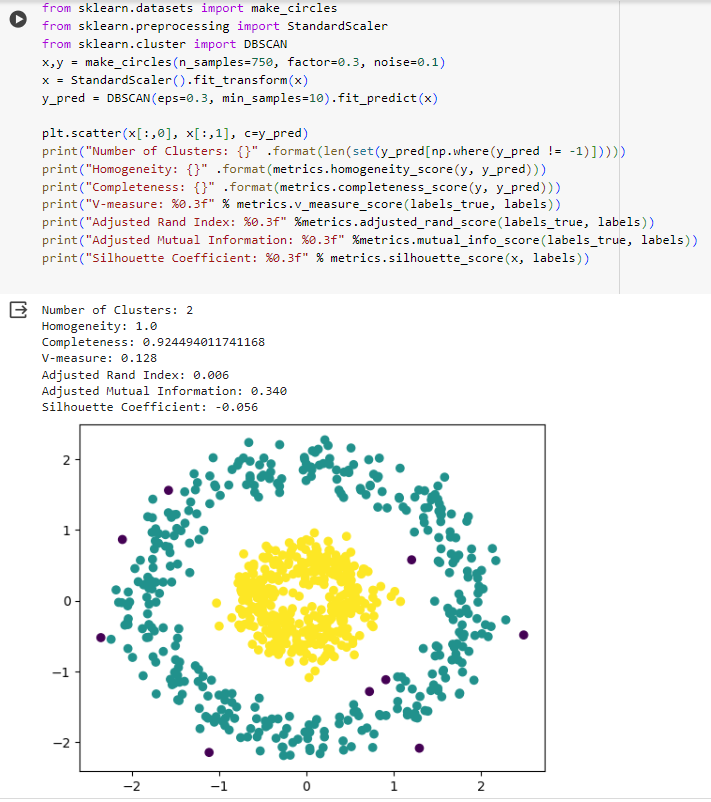
**Plotting the different estimated number of clusters obtained using DBSCAN Algorithm**





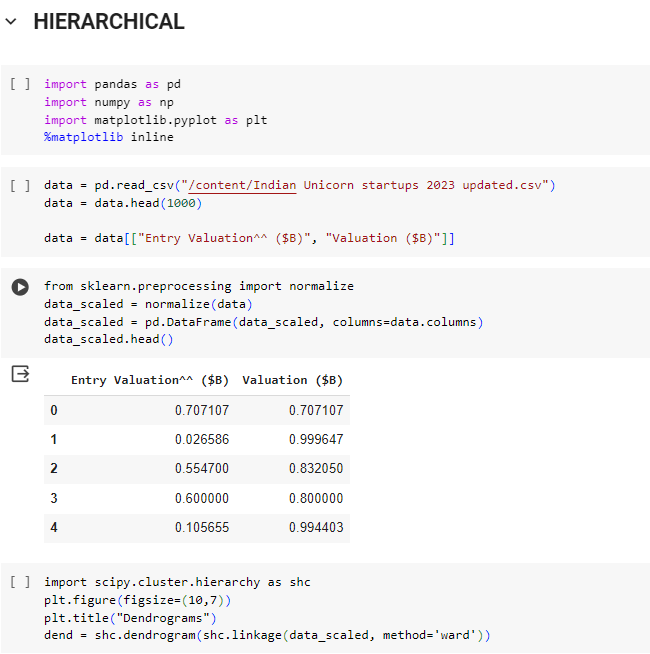


**Visualization of 2 clusters in different format**

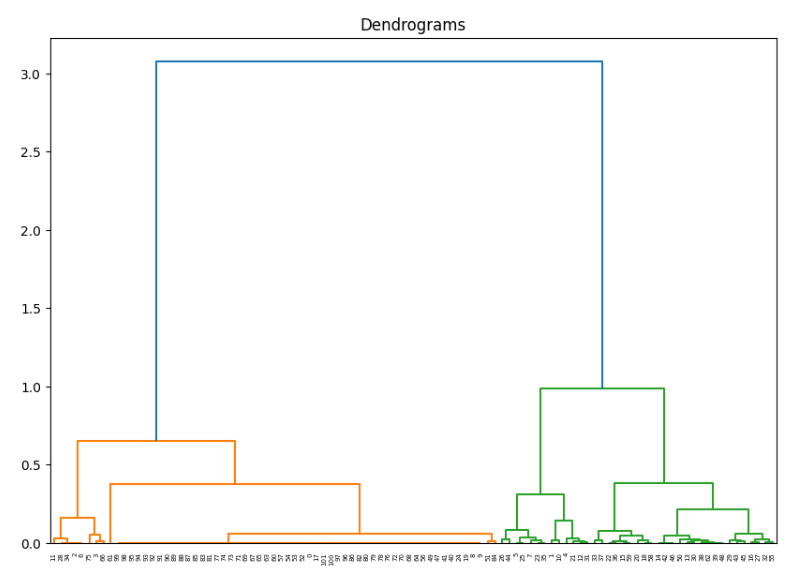


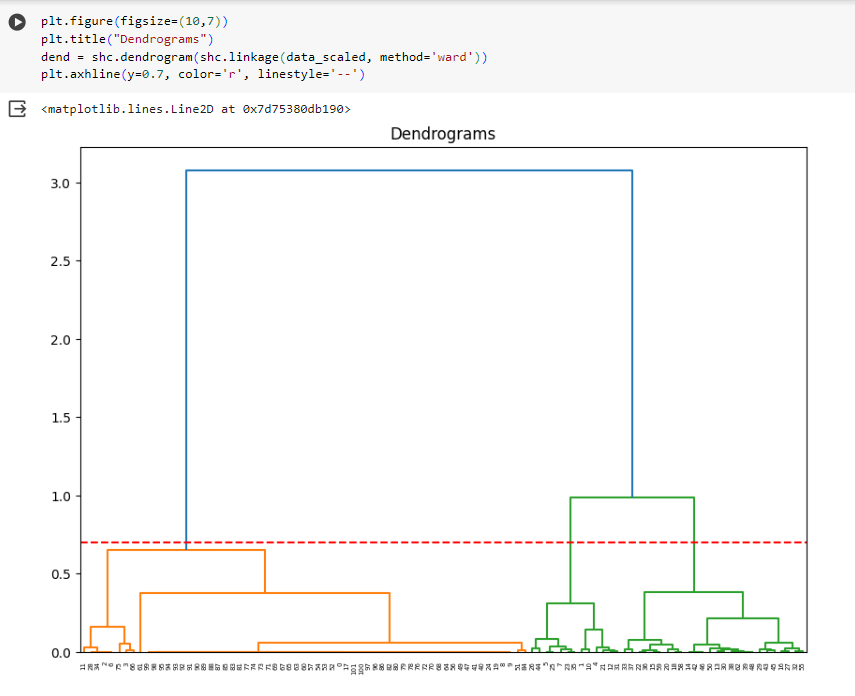
**3. HIERARCHICAL**

**Importing required libraries and using the attributes as per our requirement from the dataset**

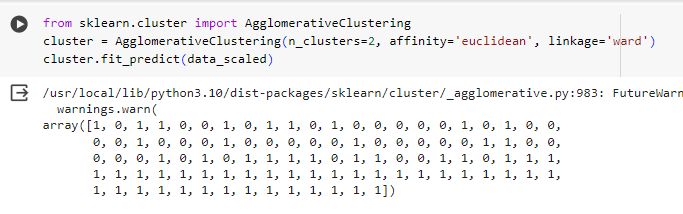


**Plotting the dendogram of the data attributes**

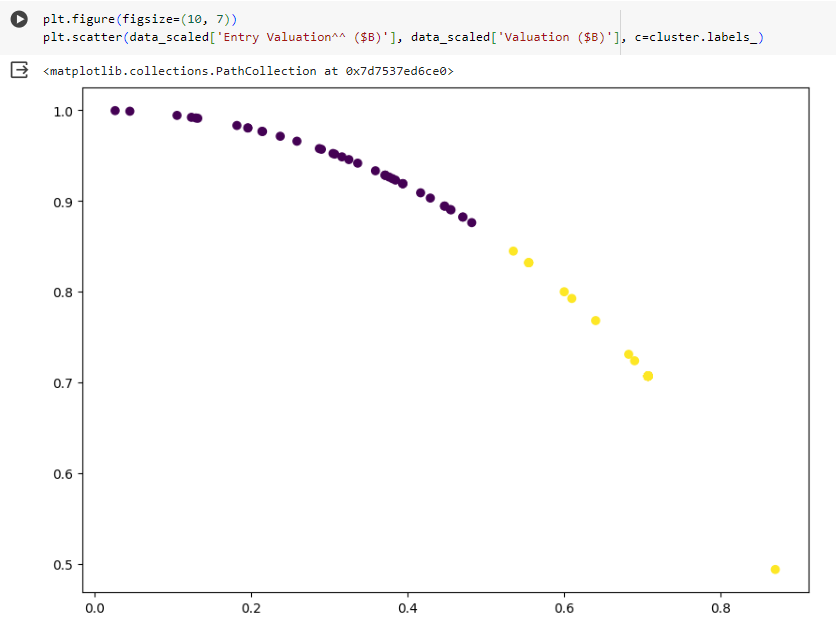




**Applying Agglomerative Hierarchical Clustering Algorithm on our data**



**Visualizing the data attributes in the form of clusters**



**CONCLUSION : Thus, we have implemented different types of clustering algorithms in this experiment.**