Faculty of Engineering
Department of Information and Communication Technology
B.Sc. (Honors) in Internet of Things

Bangabandhu Sheikh Mujibur Rahman Digital University

Course Title : Data Science
Course Code: IOT- 4313

Assignment-02

SUBMITTED TO
Nurjahan Nipa
Lecturer,
Department of IRE, BDU.

SUBMITTED BY
Name : Nishat Tasnim Shishir
ID :1901030
Department : IRE
Session : 2019-20

Date of Submission: 13th October 2023

Clustering

Let's imagine we're owning a supermarket mall and through membership cards, we have some basic data about were customers like Customer ID, age, gender, annual income and spending score, which is something we assign to the customer based on our defined parameters like customer behavior and purchasing data.
The main aim of this problem is learning the purpose of the customer segmentation concepts, also known as market basket analysis, trying to understand customers and separate them in different groups according to their preferences, and once the division is done, this information can be given to marketing team so they can plan the strategy accordingly.
This Mall_Customer dataset that has been provided to we is composed by the following five features:

- CustomerID: Unique ID assigned to the customer
- Gender: Gender of the customer
- Age: Age of the customer
- Annual Income (k$): Annual Income of the customer
- Spending Score (1-100): Score assigned by the mall based on customer behavior and spending nature.

In this particular dataset we have 200 samples to study.

We have to find out:
Part A: K-means Clustering
Part B: Hierarchical Clustering
Part C: Density-based Clustering

The necessary files are uploaded to GitHub.
GitHub Link: https://github.com/nishat-shishir/DS_Assignment_02.git

Part A

K-means Clustering

In this part, you will be utilizing K-means clustering algorithm to identify the appropriate number of clusters. You may use any language and libraries to implement K-mean clustering algorithm. Your K-mean clustering algorithm should look for appropriate values of K at least in the range of 0 to 15 and show their corresponding sum-of-squared errors (SSE).

Answer:

The process of classifying n observations into K clusters is called K-means clustering. By using vector quantization, it attempts to allocate each observation to the cluster whose prototype is the closest mean or centroid. K-means is a distance- or centroid-based technique that uses distance measurements to determine how to allocate a point to a cluster. Each cluster in K-Means has a centroid attached to it.The K-Means algorithm's primary goal is to reduce the total distances between each location and the cluster centroid.

1. Data Loading

Firstly, we need to import necessary libraries, including pandas, numpy, matplotlib.pyplot, and KMeans from scikit-learn. After that, Using pd.read_csv, I load a dataset from a CSV file ("Mall_Customers.csv") and saves it in a DataFrame called d.

2. Feature Extraction and Standardization

i. Here, I extract relevant features for clustering, which are "Annual Income" and "Spending Score," and stores them in the "a" array.

ii. Then standardizes the data using StandardScaler to have zero mean and unit variance, which is important for K-means clustering.

3. K-Means Clustering

i. Here I initialize an empty list called sse in this section to hold the Sum of Squared Errors (SSE) for various values of K.

ii. K-means clustering algorithm runs for K values ranging from 1 to 15.

iii. For each K value, the code creates a K-means model with the KMeans class, specifying parameters such as the number of clusters (n_clusters), initialization method (init), maximum number of iterations (max_iter), and the number of times the algorithm will be run with different centroid seeds (n_init).

iv. The SSE for each K is calculated and stored in the sse list.

4. Elbow method plotting

To identify the optimal K value, we employed the Elbow Method.

i. K, or the number of clusters, is displayed on the x-axis, while SSE is displayed on the y-axis.

ii. For each K value, we calculated the corresponding SSE, which measures the distance between data points and their assigned cluster centroids.

iii. The "Elbow Method" seeks to locate the plot's "elbow" point. The elbow represents the starting point of a slower SSE decline.

The optimal number of clusters(k) is based on the point where the SSE After a considerable decline, values level out and resemble a "elbow." It strikes good balance between the quality of clustering and the complexity of the model. The optimal K may vary depending on the specific dataset and problem.

Result:

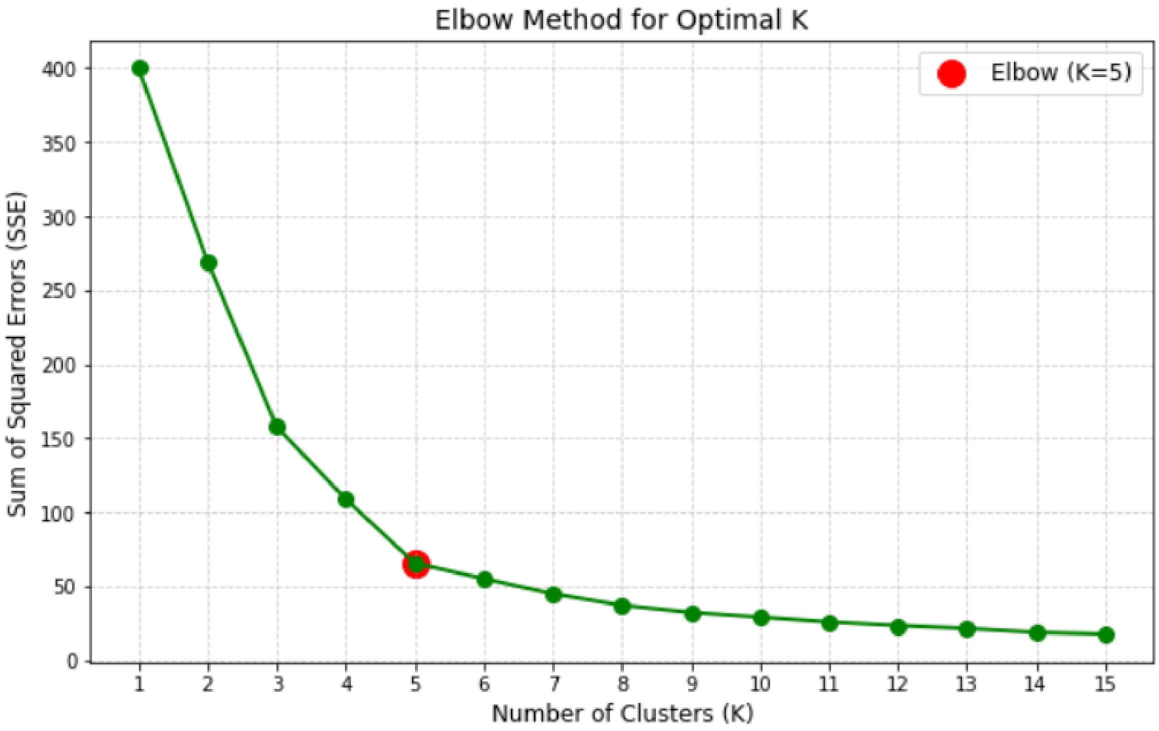

Figure: Elbow method

Part B
Hierarchical Clustering

In this part, you will apply hierarchical clustering algorithm (agglomerative or divisive) to the provided mall dataset.

Answer:

Hierarchical clustering is an unsupervised learning technique used to group similar objects into clusters. It creates a hierarchy of clusters by merging or splitting them based on similarity measures. Hierarchical clustering groups similar objects into a dendrogram. It merges similar clusters iteratively, starting with each data point as a separate cluster. This creates a tree-like structure that shows the relationships between clusters and their hierarchy.

1. Data Loading and Preprocessing

i. Firstly, we have to import necessary libraries, including pandas, numpy, matplotlib.pyplot, and scipy.cluster.hierarchy.
ii. Then I load the dataset from a CSV file("Mall_Customers.csv") and store it in the DataFrame d.
iii. Extract relevant features for clustering ("Annual Income" and "Spending Score") into the array “a”.
iv. Then standardize the data using StandardScaler, which is essential for hierarchical clustering.

2. Hierarchical Clustering with Dendrogram Plotting

i. Hierarchical clustering performs using the "ward" linkage method and the Euclidean distance metric.
ii. The linkage function from scipy.cluster.hierarchy is used to create a linkage matrix linked based on the data in data_scaled. The "ward" method minimizes the variance of distances within clusters.
iii. The dendrogram function generates the dendrogram plot based on the linked matrix.
iv. Parameters such as orientation, distance_sort and show_leaf_counts are set for dendrogram customization.

Agglomerative hierarchical clustering approach starts with individual data points as separate clusters and then merges them into larger clusters iteratively. The merging process is based on the similarity between clusters or data points. In this case, the "ward" linkage method and the Euclidean distance metric are used for clustering.

The dendrogram plot visually represents the hierarchical clustering process, showing how data points or clusters are grouped together based on their similarity. The "color_threshold" parameter is set to 5, indicating that clusters with a Euclidean distance less than 5 are merged together.

Result:

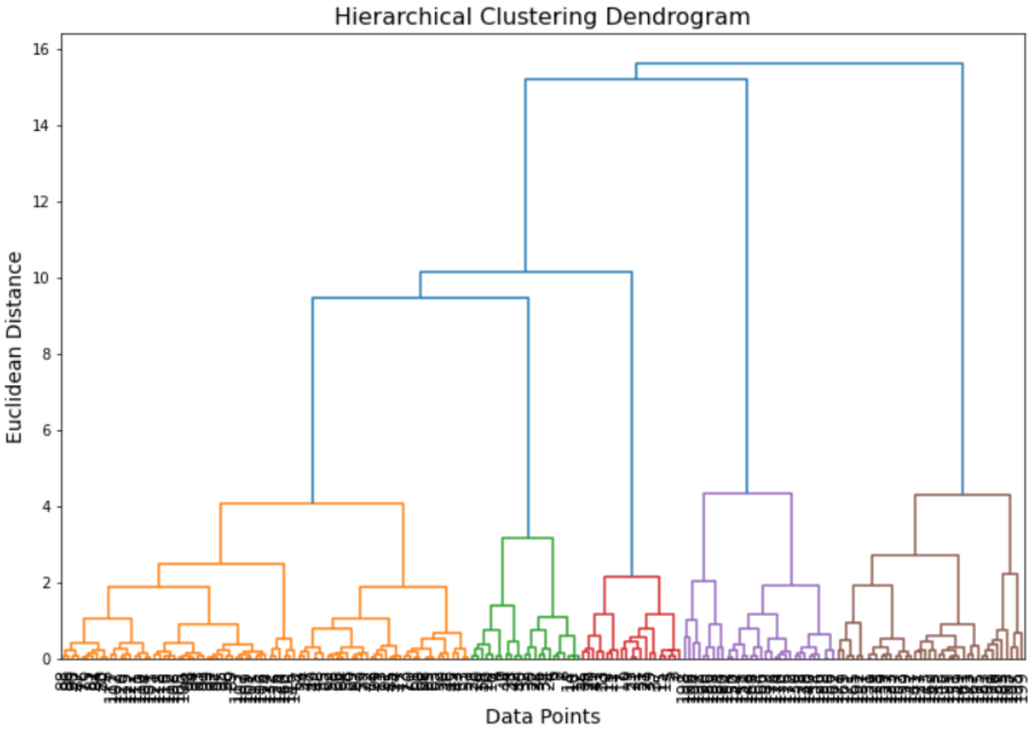

Figure: Hierarchical Clustering with Dendrogram

Part C
Density-based Clustering

In this part, you will apply density-based clustering algorithm to the provided dataset.

Answer:

In this report, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm's outcomes, applied to a dataset. The goal is to discover client clusters based on their density in the feature space, which will provide insights into consumer segmentation.

1. Data Loading and Preprocessing

i. Firstly, we need to import necessary libraries, including pandas, numpy, matplotlib.pyplot, StandardScaler from sklearn, and DBSCAN from sklearn.cluster.
ii. Then load the dataset from a CSV file and store it in the DataFrame d.
iii. Extract relevant features for clustering ("Annual Income" and "Spending Score") into the array a.
iv. Standardize the data using StandardScaler. Standardization is performed to ensure that all features have the same scale.

2. DBSCAN Clustering

i. Here, I create a DBSCAN clustering model. DBSCAN is a density-based clustering algorithm.
ii. The key hyperparameters are set here:
   - eps (epsilon) is the maximum distance between two samples for one to be considered as in the neighborhood of the other.
   - min_samples is the number of samples (data points) in a neighborhood for a point to be considered as a core point.

3. Data Point Labeling and Visualization

i. Here, a scatter plot is created to visualize the clustering. The x-axis represents "Annual Income," the y-axis represents "Spending Score," and data points are colored according to their cluster labels.
ii. The cmap parameter specifies the color map used to represent different clusters.
iii. The colorbar is added to the plot to help interpret the cluster labels.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups data points that are closely packed and separates them from sparse regions. It identifies core points (dense regions), border points, and noise points.
Core points are data points within a specified neighborhood (defined by eps) that have a sufficient number of neighboring data points (defined by min_samples). Border points are data points within the neighborhood of a core point but do not have enough neighbors to be core points.
Noise points are data points that do not belong to any cluster.

Result:

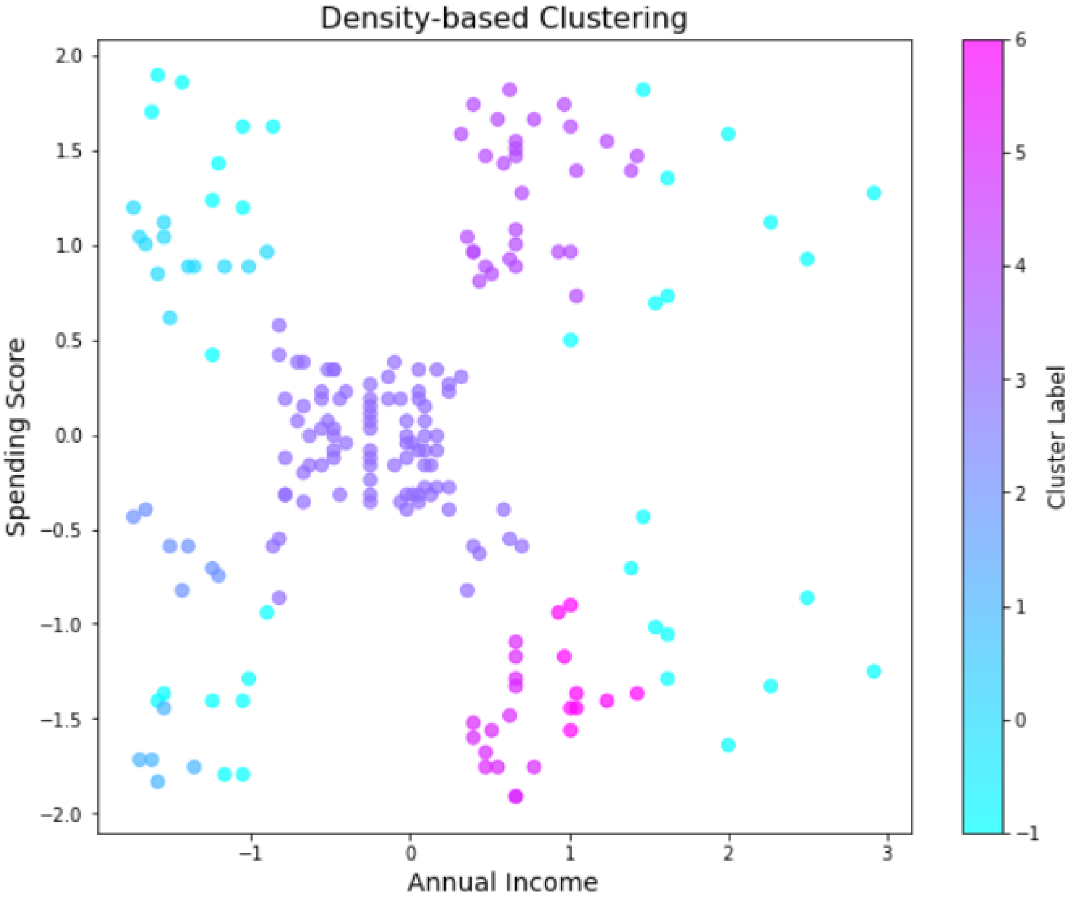

Figure: Density-based Clustering

Conclusion: In tis report, here is the detailed explanation of my approaches against the given three part and the results obtained for all of the clustering algorithms required in Parts A, B, and C. I upload both files in my personal GitHub profile and provide the link in the first page of this report.

The End