Bangabandhu Sheikh Mujibur Rahman
Digital University

Course Title: Data Science
Course Code: IoT 4313
Assignment No: 02

SUBMITTED TO

Nurjahan Nipa
Lecturer,
Dept. of IRE, BDU.

SUBMITTED BY

Name : Shabit Mahmud
ID : 1901044
Department: Internet of Things and Robotics Engineering.
Session : 2019-20

Date of Submission: 14 Oct 2023

PART (A)

Detailed explanation of K-means clustering approaches:

Data Loading and Exploration:

- First I started by importing necessary libraries such as pandas for data handling, numpy for numerical operations, KMeans from sklearn.cluster for clustering, and matplotlib.pyplot for plotting.
- I read the dataset using pd.read_csv( ) and stored it in the df dataframe.

Data Preprocessing:

- I encoded the "Genre" (Gender) column into numerical values by mapping 'Male' to 0 and 'Female' to 1. This conversion makes the gender feature usable in the K-means algorithm.
- I selected relevant features for clustering. I used Age, Annual Income, Spending Score and the encoded Genre columns.

Determining Optimal K (Number of Clusters):

- I iterated through a range of K values from 1 to 15, creating a K-means model for each K. I fit the model to the data and calculated the SSE values.
- After calculating the SSE for each K, I plotted the SSE values in an elbow plot. This plot helps us to identify the optimal number of clusters based on the "elbow" point, where the SSE starts to level off.

Performing K-means Clustering:

- I set the optimal_k to 5 by visualizing the elbow curve and predicted the cluster labels for each data point using kmeans_optimal.fit_predict(X) and finally added those cluster labels to our dataset as a new column called Cluster.

Final Output Visualization:

- I selected two features Annual income and Spending score and drew a scatter plot to visualize those five clusters.

PART (B)

Detailed explanation of Hierarchical clustering approaches:

Data Loading and Exploration:

- First I started by importing necessary libraries such as pandas, numpy, matplotlib.pyplot, and functions from scipy.cluster.hierarchy for hierarchical clustering.
- I read the dataset using pd.read_csv( ) and stored it in the df dataframe.

Data Preprocessing:

- I encoded the "Genre" (Gender) column into numerical values by mapping 'Male' to 0 and 'Female' to 1. This conversion makes the gender feature usable in the K-means algorithm.
- I selected relevant features for clustering. I used Age, Annual Income, Spending Score and the encoded Genre columns.

Hierarchical Clustering:

- I performed hierarchical clustering using the linkage function from scipy.cluster.hierarchy and I chose the "ward" method and the Euclidean distance metric for it.

Dendrogram Visualization:

- After performing hierarchical clustering, I visualized the results by plotting a dendrogram. The dendrogram displays the hierarchical structure of the clusters and how data points are grouped together at different levels of similarity or dissimilarity.
- The dendrogram function is used to create the dendrogram. This visualization helps us to identify the optimal number of clusters based on the branching patterns and heights in the dendrogram.

Choosing the Number of Clusters:

- I set the num_clusters to 3 by visualizing the dendrogram. This step allows us to control the granularity of the clusters based on the dendrogram's structure.

Cluster Assignment:

- I assigned cluster labels to the data points using the “fcluster” function with the specified number of clusters. This function segments the data into clusters based on the hierarchical structure derived from the dendrogram.
- Cluster labels are added to the data frame under the "Cluster" column.

Cluster Visualization:

- Finally, I drew a scatter plot to visualize those three clusters with different Color.

PART (C)

Data Loading and Exploration:

- First, I started by importing necessary libraries such as pandas for data handling, numpy for numerical operations, scikit-learn's DBSCAN for clustering, and matplotlib for data visualization.
- I read the dataset using pd.read_csv( ) and stored it in the df dataframe.

Feature Selection:

- I selected two specific features for clustering which are Annual Income and Spending Score. These two features are extracted from the dataset and stored in the data frame named x.

Scatter Plot Visualization:

- A scatter plot is created using the 'Annual Income (k$)' on the x-axis and 'Spending Score (1-100)' on the y-axis. This plot is an initial representation of the data.

DBSCAN Clustering:

- The DBSCAN clustering algorithm is applied to the selected features using the following parameters:
    1. eps (epsilon): 5
    2. min_samples: 5

- The eps parameter defines the maximum distance between samples for them to be considered in the same neighborhood. min_samples is the minimum number of data points required to form a dense region (core point).

Cluster Labels:

- The cluster labels are obtained using the fit_predict method of the DBSCAN model, and they are stored in the labels variable.

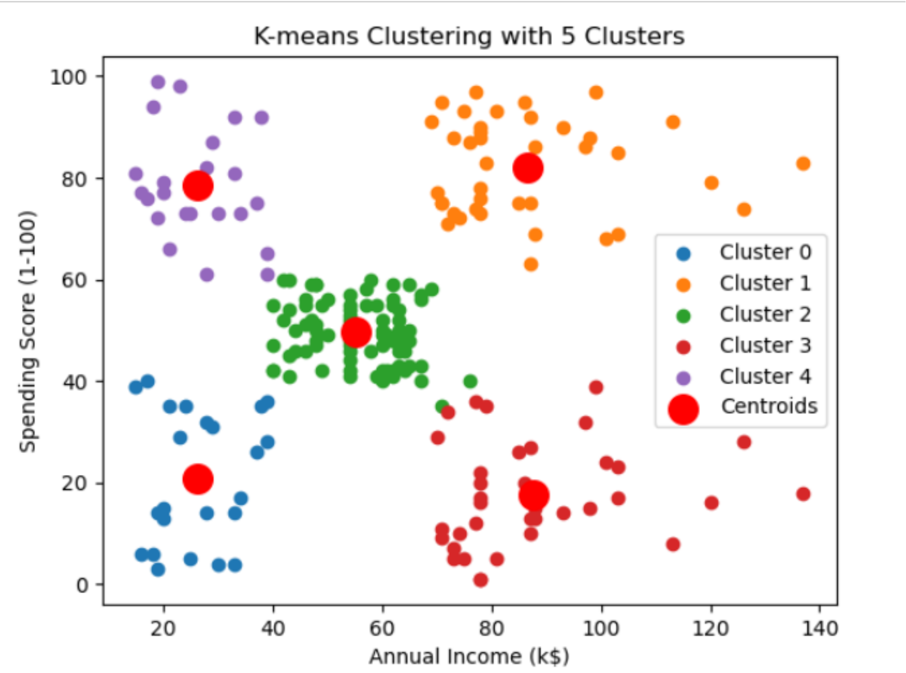
Visualizing Clusters:

- Add the cluster labels with the main data frame.
- Generates a scatter plot which visualizes the clusters as well as noise points (outliers). It separates the data points into different clusters based on the cluster labels obtained from DBSCAN.
- The noise points are shown in the black color and different colors for each cluster.

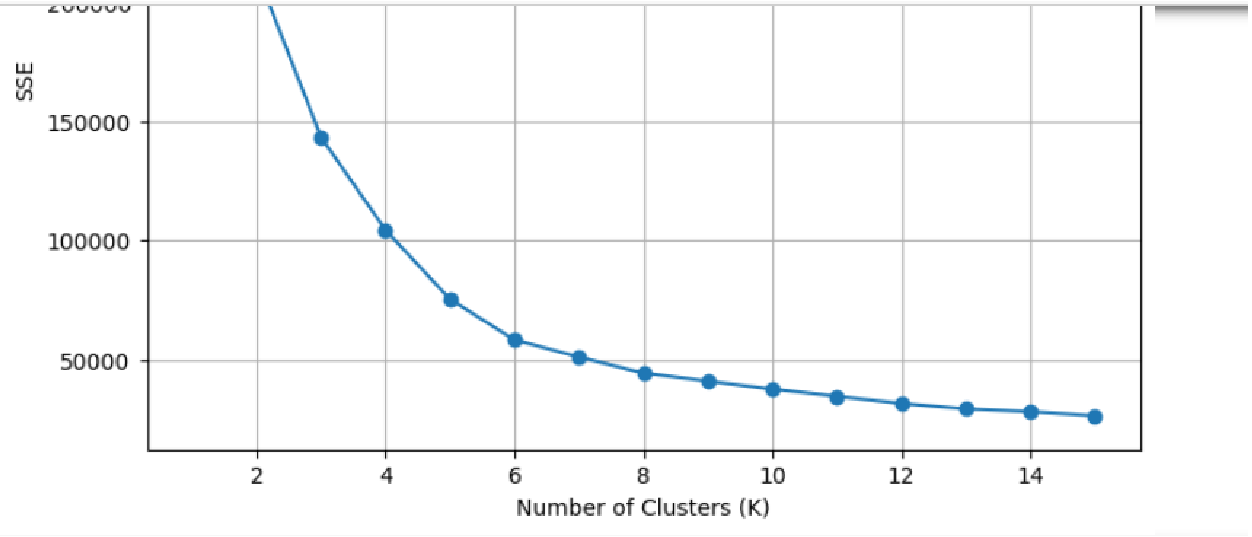
Explanation of the result:

K-means Clustering:

For K-means clustering, we got the optimal number of clusters is 5. It groups data points into K clusters based on their similarity.



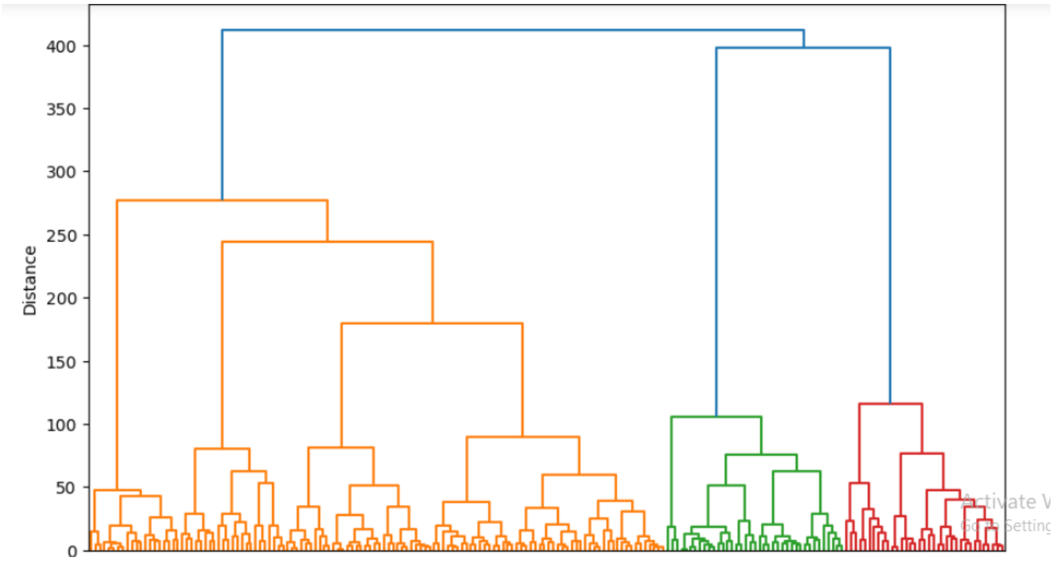
Here, I found the optimal number of clusters from elbow curve and the value for the optimal number of clusters is 4



Hierarchical Clustering:

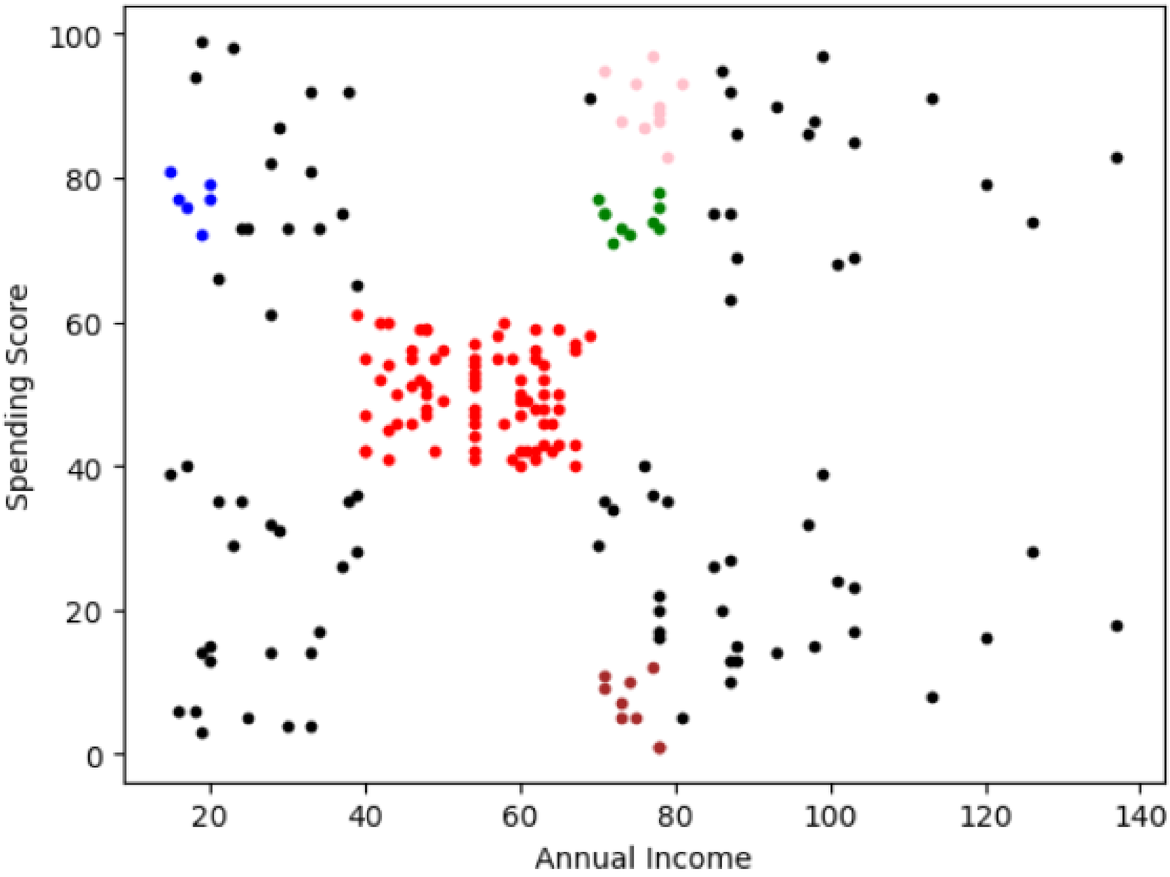
For hierarchical clustering, we got the optimal number of clusters is 5. Hierarchical clustering builds a tree-like structure of clusters, which can be cut at different levels to obtain different numbers of clusters.

Here, we found the number of clusters from dendrogram which is 3.



Density-based Clustering:

DBSCAN identifies clusters based on the density of data points. For density based clustering, the number of clusters and noise points are given below with a scatter plot where the black point indicates the noise point and clustered are represented with other colors.



GitHub link: https://github.com/Shabitmahmud/Clustering