BANGABANDHU SHEIKH MUJIBUR RAHMAN DIGITAL UNIVERSITY, BANGLADESH.

ASSIGNMENT-02

Department: IoT and Robotics Engineering
Program: BSc. In IoT and Robotics Engineering
Course Title: Data Science
Course Code: IIoT 4313

Submitted To:	**Submitted By:**
Nurjahan Nipa	Sabrina Shawon
Lecturer	**ID:** 1801002
Department of IoT and Robotics Engineering	Department of IoT and Robotics Engineering
Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh	**Program:** BSc. In IoT and Robotics Engineering

Date of Submission: 14th October, 2023

Assignment 02: Clustering

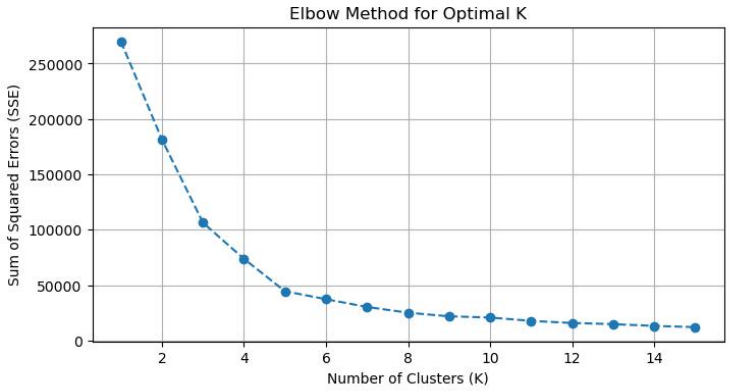
PART (A)

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).

Explanation:
The k-means clustering method is an unsupervised machine learning technique used to identify clusters of data objects in a dataset. There are many different types of clustering methods, but k-means is one of the oldest and most approachable.

- ✓ To perform K-means clustering, at first I select the columns "Annual Income (k$)" and "Spending Score (1-100)" for clustering.
- ✓ Then range of K values to consider is from 1 to 16(as range should be of 0-15).
- ✓ Then I initialize an empty list sse to store the SSE values for different K values.
- ✓ After that I use a loop to perform K-means clustering for each value of K and calculate the SSE using kmeans.inertia_.
- ✓ Finally, I have plotted the SSE values for different K values to identify the elbow point.

Result Obtained:




```
KMeans(n_clusters=15, random_state=0)
SSE:  [269981.28, 181363.59595959596, 106348.37306211119, 73679.78903948834, 44448.45544793371, 37265.86520484346, 30259.65720728547, 25095.703209997548, 21830.041978049438, 20736.679938924128, 17702.59593229628, 15810.838613705502, 14763.330402558204, 13165.329070181626, 12064.939000692291]
```

PART (B)

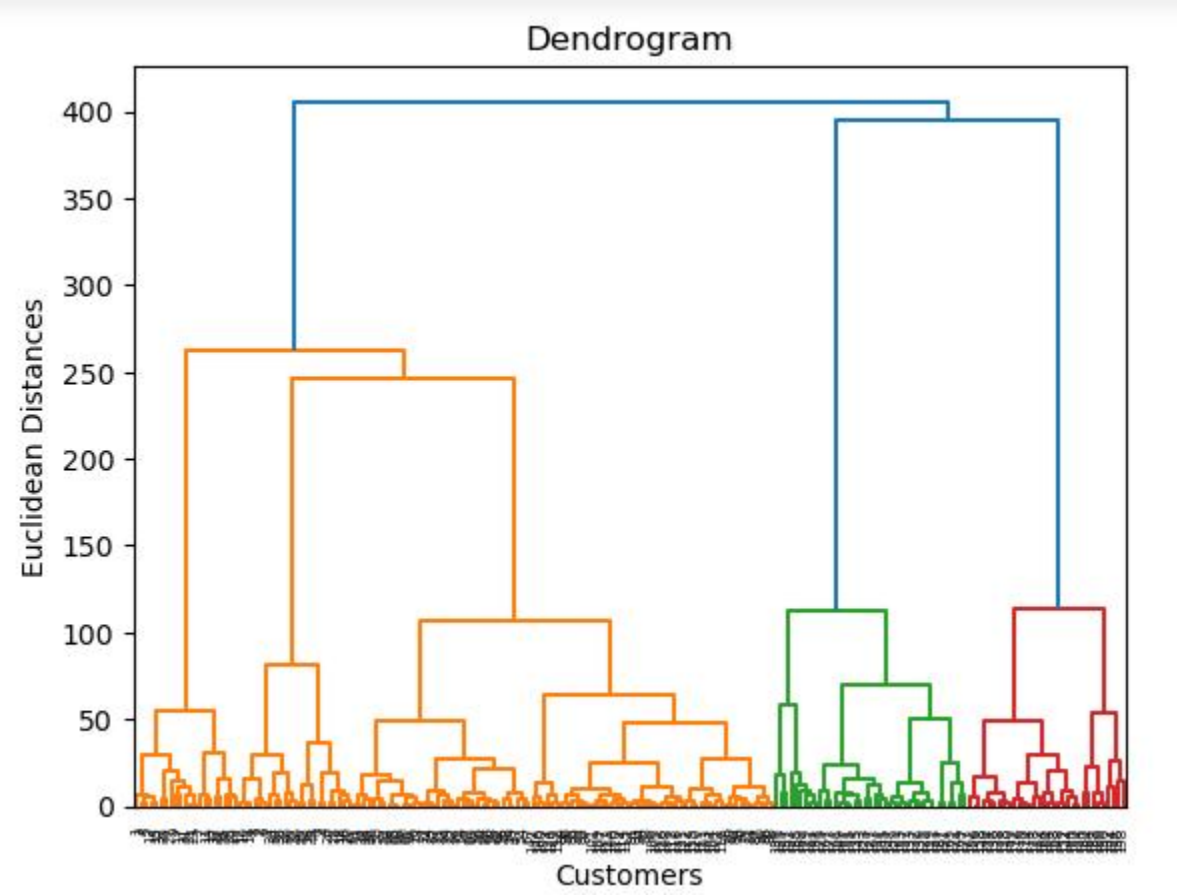
Hierarchical Clustering: In this part, you will apply hierarchical clustering algorithm (agglomerative or divisive) to the provided mall dataset.

Explanation:

Hierarchical clustering is an unsupervised learning method for clustering data points. The algorithm builds clusters by measuring the dissimilarities between data.

- ✓ For the Hierarchical Clustering, firstly I have to install scipy library and import that.
- ✓ Here I have used agglomerative hierarchical clustering.
- ✓ scipy.cluster.hierarchy module is used to perform agglomerative hierarchical clustering.
- ✓ The sch.linkage function computes the linkage matrix based on the "ward" method, which minimizes the variance of distances between clusters.
- ✓ Now we visualize the hierarchical clustering result using a dendrogram.

Result Obtained:



PART (C)

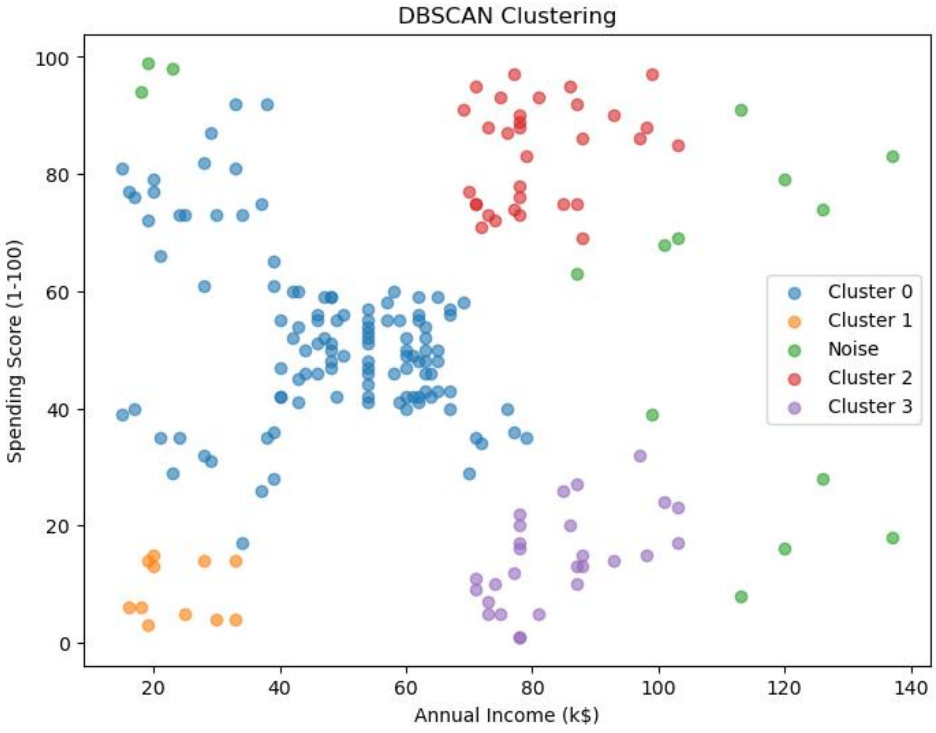
Density-based Clustering: In this part, you will apply density-based clustering algorithm to the provided dataset.

Explanation:

Density-Based Clustering refers to unsupervised machine learning methods that identify distinctive clusters in the data, based on the idea that a cluster/group in a data space is a contiguous region of high point density, separated from other clusters by sparse regions.

- ✓ To apply a density-based clustering algorithm to the "Mall_Customers.csv" dataset, we can use the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm.
- ✓ DBSCAN is effective at identifying clusters of varying shapes and handling noisy data.
- ✓ To apply this clustering, at first I standardized the data using StandardScaler to ensure that both features have similar scales.
- ✓ Then I performed DBSCAN clustering using the DBSCAN class from scikit-learn.
- ✓ Adjust the eps (epsilon) and min_samples parameters to control the density and cluster size.
- ✓ After that I added the cluster labels to the DataFrame and plotted the clusters, with different clusters and noise points represented by different colors.

Result Obtained:



Github Link:

https://github.com/sabrina991/DataScienceClustering.git