Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

Faculty of Cyber Physical System

Department of IoT and Robotics Engineering (IRE)

Course Title: Data Science

Course Code: IOT 4313

Assignment 02

Topic: Clustering

Submitted To,

Nurjahan Nipa

Lecturer

Department of IRE, BDU

Submitted By,

Tousif Mahmud Emon

ID: 1901011

Reg: 201911000114

Session: 2019-20

Date of Submission: 13 October, 2023.

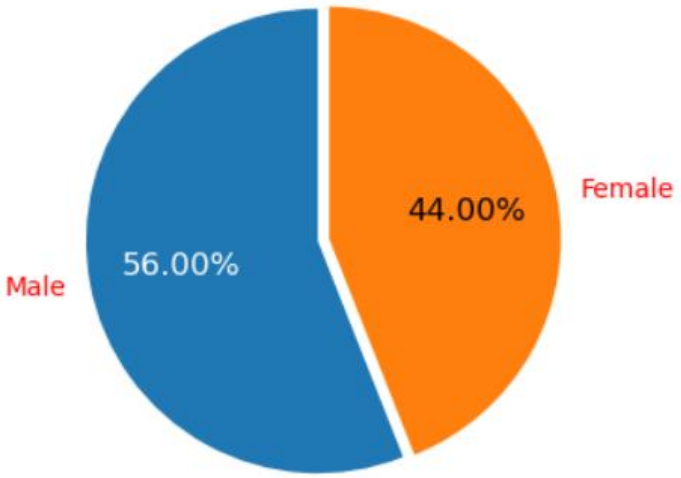
Part A (K-Means Clustering)

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).

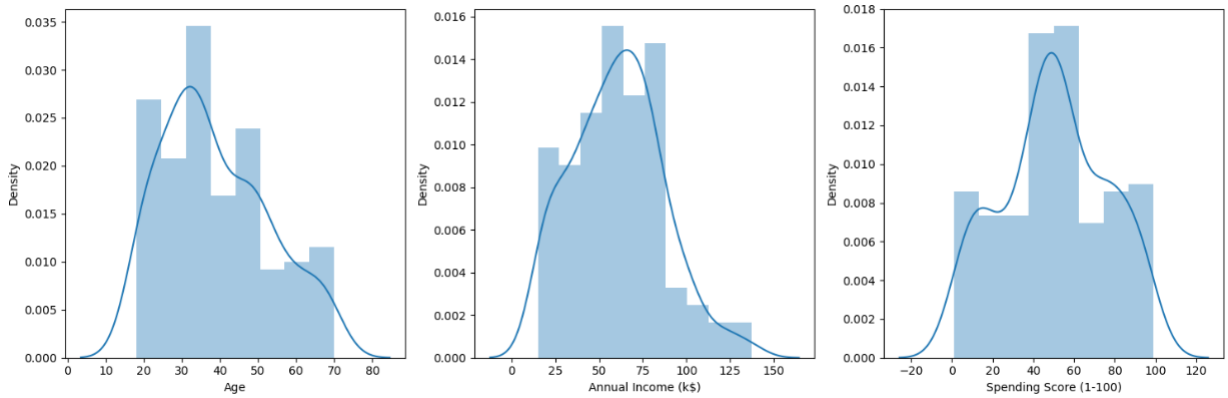
Step 1 (Import Libraries and Load Data): Imported necessary libraries, suppresses warnings, and loads the dataset "Mall_Customers.csv" into a Pandas Data Frame.

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

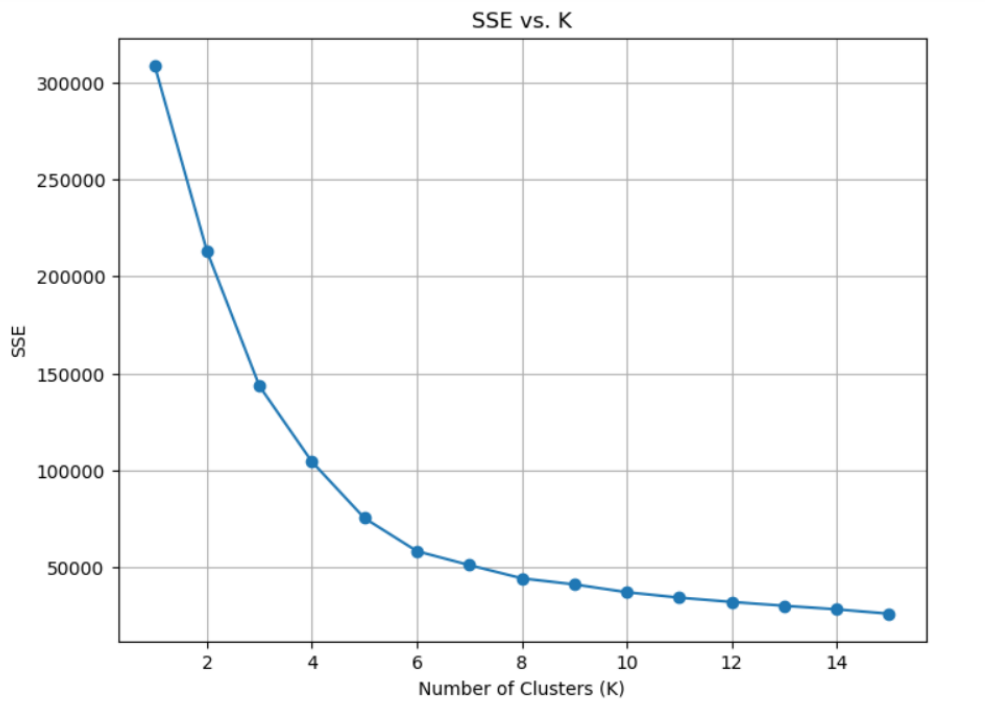
Step 2 (Genre Calculation and Pie Chart): I've calculate the distribution of 'Genre' in the dataset and create a pie chart to visualize the distribution. The pie chart displays the percentage of male and female customers in the dataset.



Step 3 (Distribution Plots): In this step I've created distribution plots (histograms) for 'Age,' 'Annual Income (k$),' and 'Spending Score (1-100).' It visualizes the distribution of these features in the dataset.



Step 4 (K-Means Clustering): In this step, I've performed K-means clustering on the features 'Age,' 'Annual Income (k$),' and 'Spending Score (1-100).' According to the question, I've calculated the values of K at least in the range of 1 to 15 and calculated the sum-of-squared errors (SSE) for each K. I've created an elbow plot to visualize the SSE values against different K values.



Step 5 (K-Means Clustering with Optimal K): Based on the elbow plot, the optimal K value is 5 in this case. Then I performed K-means clustering with K=5, and assigned cluster labels to data points, and add the 'label' column to the Data Frame.

	Age	Annual Income (k$)	Spending Score (1-100)	label
0	19	15	39	4
1	21	15	81	3
2	20	16	6	4
3	23	16	77	3
4	31	17	40	4

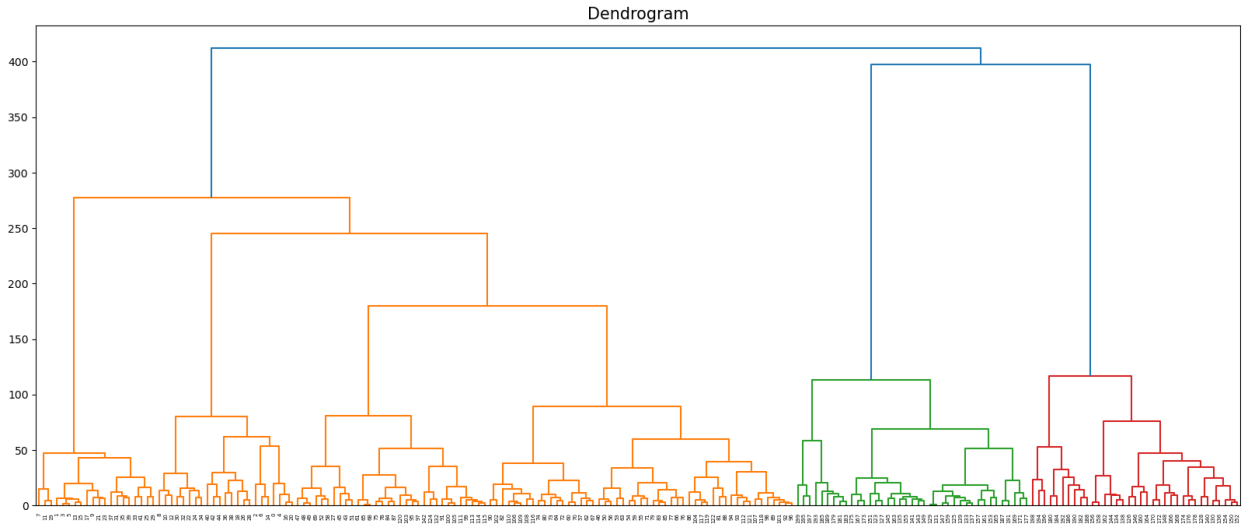
Step 6 (3D Scatter Plot): In the last step I've created a 3D scatter plot using Plotly Express to visualize the data points in three dimensions ('Annual Income (k$),' 'Spending Score (1-100),' and 'Age').

Part B: Hierarchical Cluster

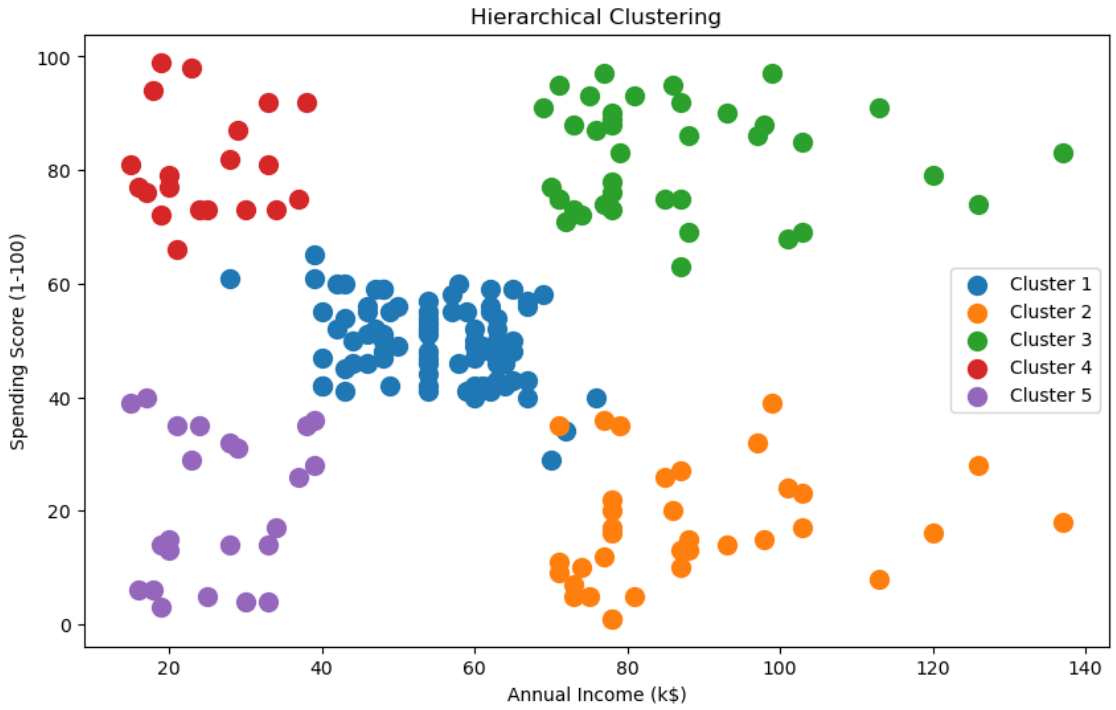
In this part, you will apply hierarchical clustering algorithm (agglomerative or divisive) to the provided mall dataset.

For hierarchical clustering I have used agglomerative algorithm in the provided dataset.

Step 1 (Import Libraries and Create Dendrogram): In this step, I've imported the necessary libraries and created a dendrogram to visualize the hierarchical structure of the data using the 'ward' linkage method. The dendrogram shows how data points are merged into clusters as the algorithm progresses.



Step 2 (Agglomerative Clustering and Scatter Plot Clusters): In the final step I've applied Agglomerative Clustering with specific parameters: 5 clusters, the 'euclidean' distance metric, and the 'ward' linkage method. This assigns cluster labels to data points. And then created a scatter plot to visualize the results of hierarchical clustering. It shows how the data points are grouped into clusters based on the Agglomerative Clustering algorithm.



Part C (Density Based Clustering)

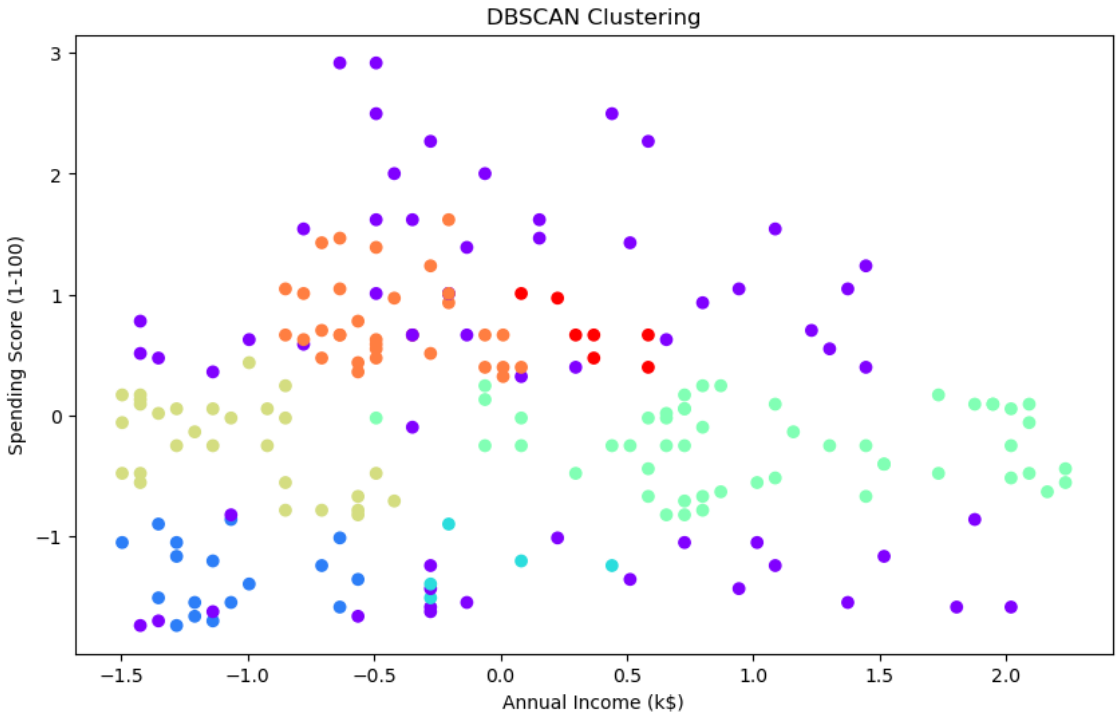
In this part, you will apply density-based clustering algorithm to the provided dataset.

In this part first of all I've imported the necessary libraries and preprocesses the data. I've selected the features ('Age,' 'Annual Income (k$),' and 'Spending Score (1-100)') for clustering and standardize them using the StandardScaler to ensure all features have similar scales.

Then I've initialized the DBSCAN model with hyperparameters such as 'eps' (maximum distance between two samples to be considered in the same neighborhood) and 'min_samples' (the minimum number of samples in a neighborhood for a data point to be considered as a core point).

Then I fit the DBSCAN model to the preprocessed data. The algorithm assigns cluster labels to data points and identifies noise points as -1.

And lastly, I've created a scatter plot to visualize the results of the DBSCAN clustering. It allows you to observe the clusters and the noise points in the data.



Github Link: https://github.com/tousifinity/Clustering/