Bangabandhu Sheikh Mujibur Rahman
Digital University

Course Title: Data Science

Course Code: ICT 1343

Assignment 02:Clustering

SUBMITTED TO:

Nurjahan Nipa

Lecturer

Department of IRE, BDU

SUBMITTED BY:

Sharika Khan

ID: 1901013

Session: 2019-2020

Third Year Second Semester,

Department of IRE.

Date of Submission:14th October, 2023.

K_Means:

K-means is a popular clustering algorithm used in data science and machine learning to group data points into clusters based on their similarity. It is an unsupervised learning technique, which means it doesn't require labeled data; instead, it tries to find patterns or structure in the data on its own.

Here's how the K-means algorithm works:

Initialization: It begins by randomly selecting K initial cluster centers (where K is a user-defined parameter).

Assignment: Each data point is assigned to the nearest cluster center, typically based on the Euclidean distance between data points and cluster centers.

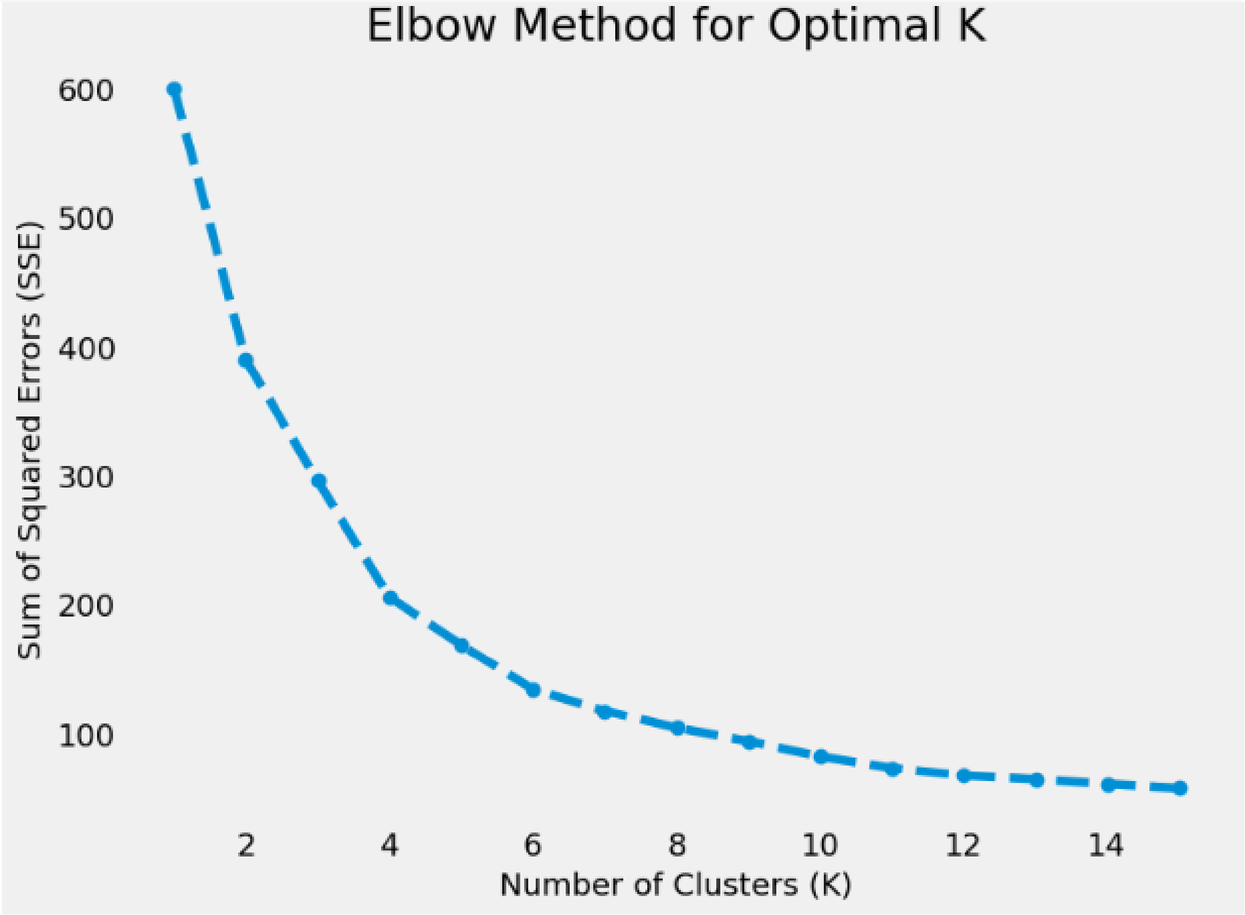
Update: After all data points have been assigned to clusters, the cluster centers are updated by computing the mean of all the data points assigned to each cluster.

Repeat: Steps 2 and 3 are repeated iteratively until convergence, which is typically defined by a certain stopping criterion, such as no or very minimal change in cluster assignments or cluster centers.

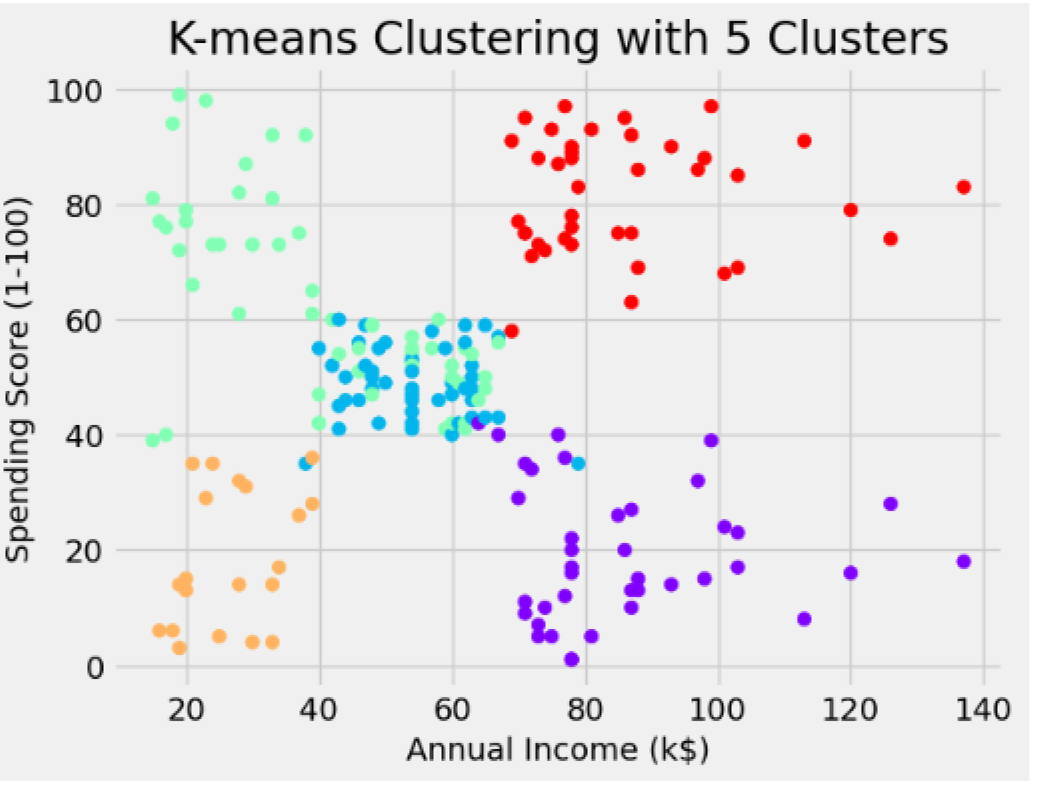
Final Clustering: The algorithm converges to a solution, and data points are grouped into K clusters, with each cluster represented by its center.

Output:

1.


Elbow Method for Optimal K
Sum of Squared Errors (SSE)
600
500
400
300
200
100
2
4
6
8
10
12
14
Number of Clusters (K)

2.


K-means Clustering with 5 Clusters
Spending Score (1-100)
0
20
40
60
80
100
Annual Income (k$)
20
40
60
80
100
120
140

Hierarchical Clustering

Hierarchical clustering is a popular clustering technique in data science and machine learning that builds a hierarchy of clusters. Unlike K-means, which requires the user to specify the number of clusters (K) in advance, hierarchical clustering doesn't require that parameter. Instead, it organizes data points into a tree-like structure, known as a dendrogram, which visually represents the clustering process and allows you to choose the number of clusters later. Here's how hierarchical clustering works:

Initialization: Each data point is initially treated as a single cluster, so you start with as many clusters as data points.

Agglomeration (bottom-up): The algorithm proceeds by iteratively merging the closest clusters to create larger clusters. The distance between clusters is typically based on a linkage criterion, which can be one of the following:

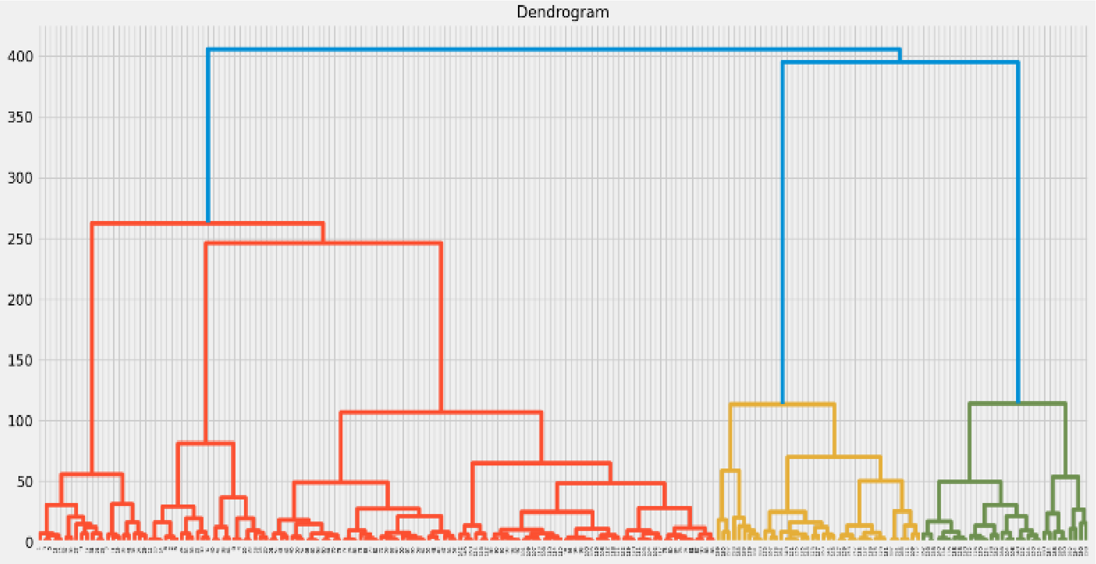
- **Single Linkage:** The distance between two clusters is defined as the minimum distance between any two data points in the two clusters.
- **Complete Linkage:** The distance between two clusters is defined as the maximum distance between any two data points in the two clusters.
- **Average Linkage:** The distance between two clusters is defined as the average distance between all pairs of data points in the two clusters.
- **Ward's Method:** It combines clusters that minimize the increase in the within-cluster sum of squares.

Dendrogram Formation: As clusters are merged, a dendrogram is constructed. The dendrogram shows the hierarchy of clusters and the order in which they were merged. It's a tree-like structure where the root represents all data points, and the leaves represent individual data points.

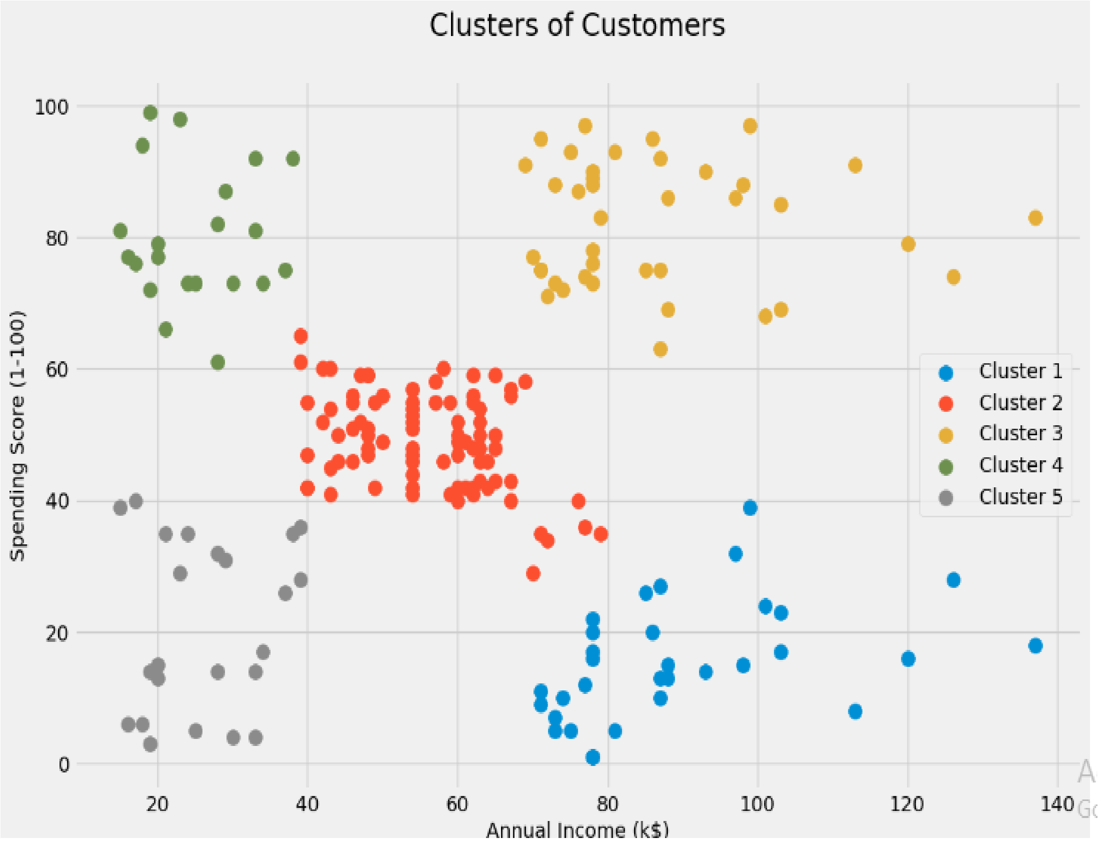
Cutting the Dendrogram: To determine the number of clusters, you can "cut" the dendrogram at a certain level. The height or distance at which you cut the dendrogram corresponds to the number of clusters you want. If you cut it at a higher level, you'll get fewer, larger clusters. If you cut it at a lower level, you'll get more, smaller clusters.

Output:

1.



2.


Clusters of Customers
Spending Score (1-100)
0
20
40
60
80
100
Annual Income (k$)
20
40
60
80
100
120
140
● Cluster 1
● Cluster 2
● Cluster 3
● Cluster 4
● Cluster 5

DBSCAN

DBSCAN, which stands for Density-Based Spatial Clustering of Applications with Noise, is a density-based clustering algorithm used in data science and machine learning. DBSCAN is particularly useful for identifying clusters of arbitrary shapes and handling noisy data points. It doesn't require the user to specify the number of clusters in advance, making it advantageous in scenarios where the number of clusters is not known or when clusters have different shapes and densities.

Here's how DBSCAN works:

Core Points: DBSCAN defines three types of data points:

- **Core Point:** A data point is considered a core point if there are at least "MinPts" data points (a user-defined parameter) within a specified radius (epsilon, ε) around it.
- **Border Point:** A data point is a border point if it is within ε distance of a core point but does not have MinPts data points within its ε radius.
- **Noise Point (Outlier):** Data points that are neither core points nor border points are considered noise points or outliers.

Cluster Formation: DBSCAN starts with an arbitrary, unvisited data point. If the point is a core point, a new cluster is created, and all reachable data points (directly or indirectly) from the core point are assigned to this cluster. This process continues until no more core points can be added to the cluster.

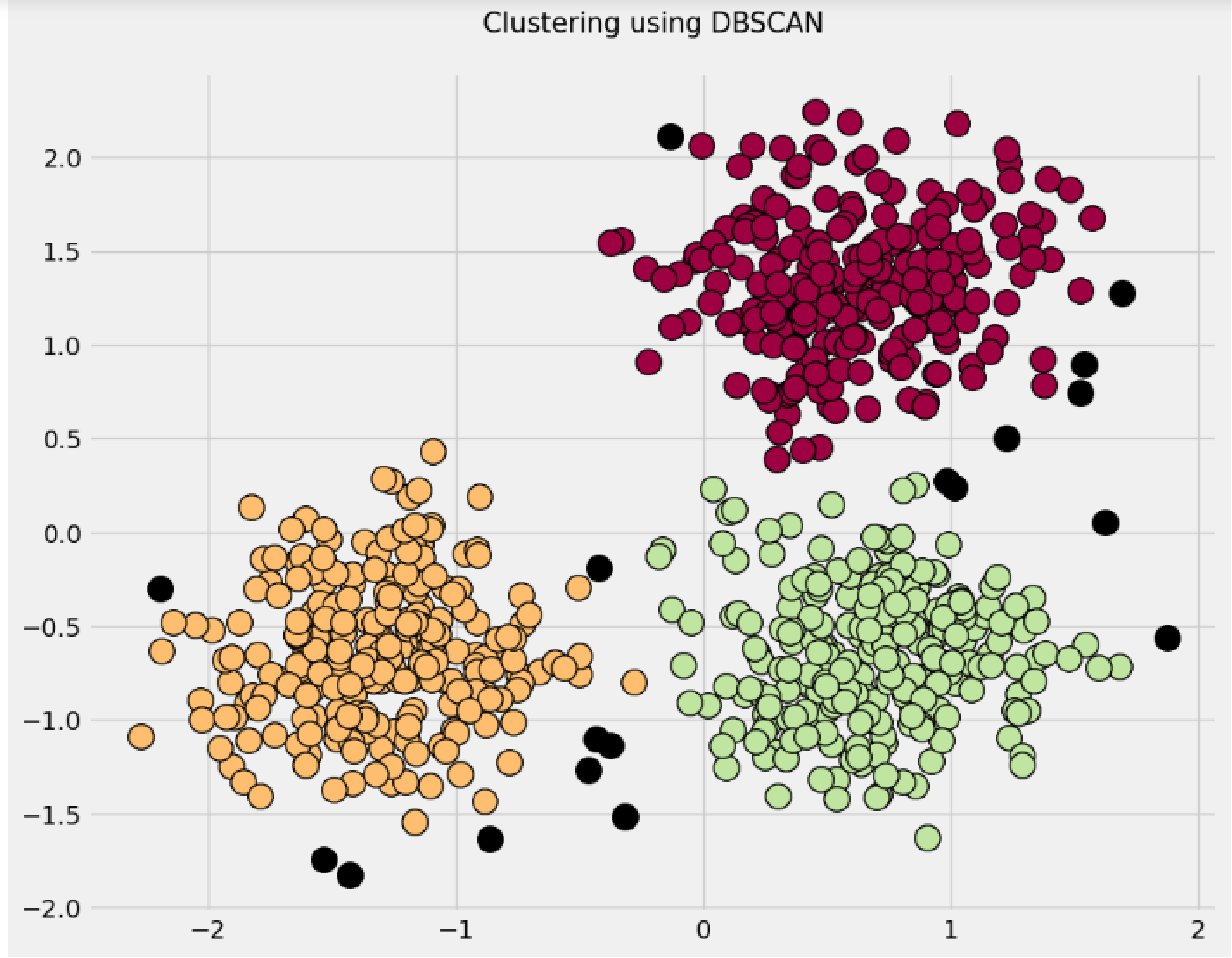
Exploring Neighbors: For each core point, DBSCAN explores its ε-neighborhood, and if it finds other core points in this neighborhood, it merges these clusters into a single cluster.

Border Points: Border points are assigned to the cluster of their corresponding core point.

Noise Points: Data points that are neither core points nor border points are treated as noise points and do not belong to any cluster**.**

Output:

1.


Clustering using DBSCAN