Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh

Faculty of Cyber Physical System

Dept. of Internet of Things and Robotics Engineering (IRE)

Course Title: Data Science

Course Code: IOT 4313

Assignment 02: Clustering

Submitted to-

Teacher name: Nurjahan Nipa

Designation: Lecturer

Department: IRE

Submitted by-

Md. Shahriar Hossain Apu (1901036)

Date of Submission: 14-10-2023

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).

Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics. Customer Segmentation can be a powerful means to identify unsatisfied customer needs. Using the above data companies can then outperform the competition by developing uniquely appealing products and services.

The most common ways in which businesses segment their customer base are:

1. **Demographic information**, such as gender, age, familial and marital status, income, education, and occupation.
2. **Geographical information**, which differs depending on the scope of the company. For localized businesses, this info might pertain to specific towns or counties. For larger companies, it might mean a customer's city, state, or even country of residence.
3. **Psychographics**, such as social class, lifestyle, and personality traits.
4. **Behavioral data**, such as spending and consumption habits, product/service usage, and desired benefits.

Advantages of Customer Segmentation

1. Determine appropriate product pricing.
2. Develop customized marketing campaigns.
3. Design an optimal distribution strategy.
4. Choose specific product features for deployment.
5. Prioritize new product development efforts.

The Challenge

You own a supermarket mall and through membership cards, you have some basic data about your customers like Customer ID, age, gender, annual income and spending score. You want to understand the customers like who are the target customers so that the sense can be given to marketing team and plan the strategy accordingly.

K Means Clustering Algorithm

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

K Means Clustering where K=3

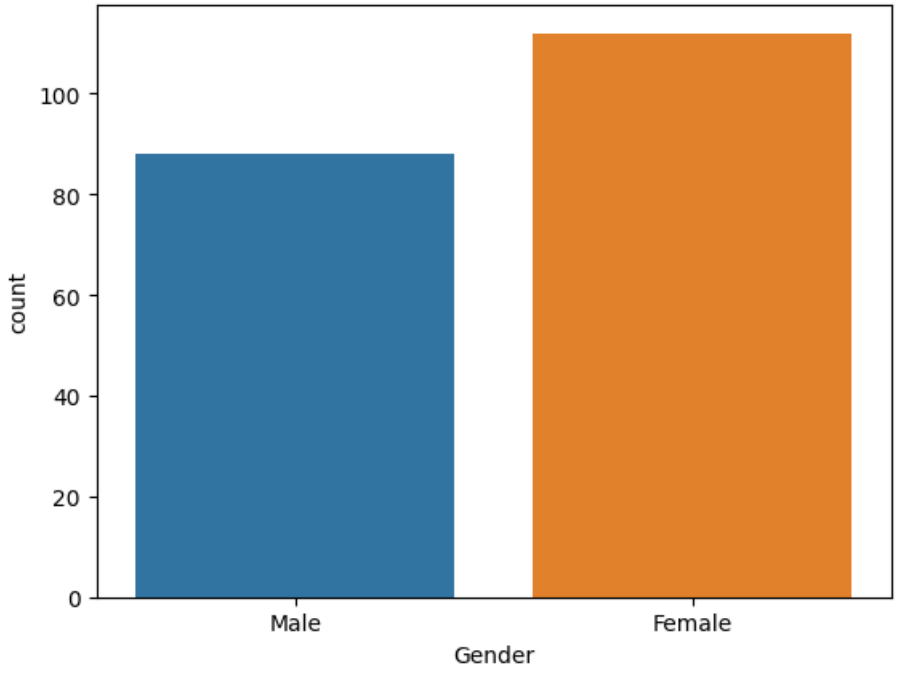
Environment and tools

1. scikit-learn
2. seaborn
3. numpy
4. pandas
5. matplotlib

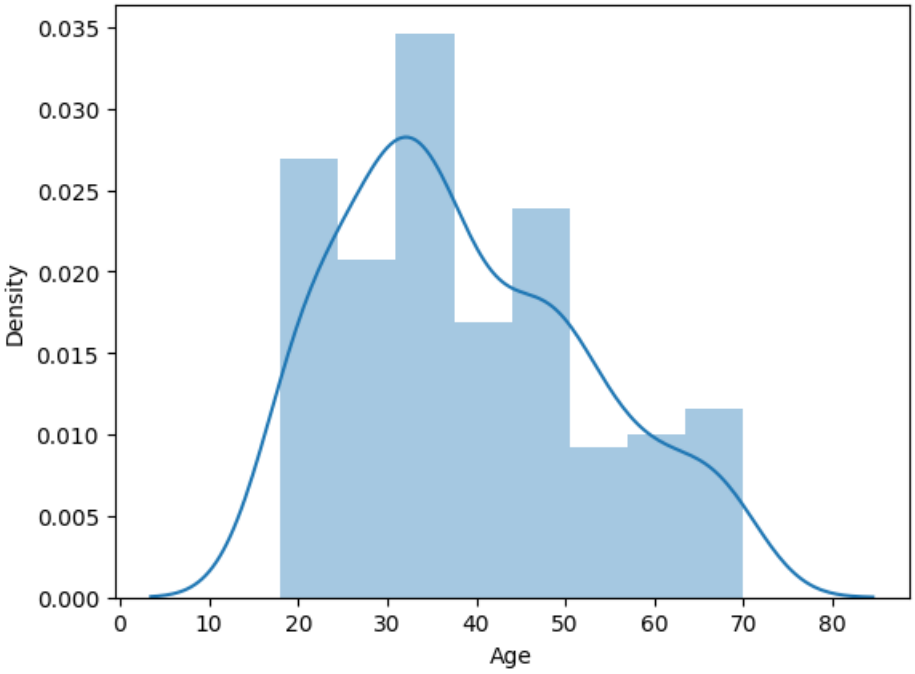
I started with loading all the libraries and dependencies. The columns in the dataset are customer id, gender, age, income and spending score.

	CustomerID	Gender	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

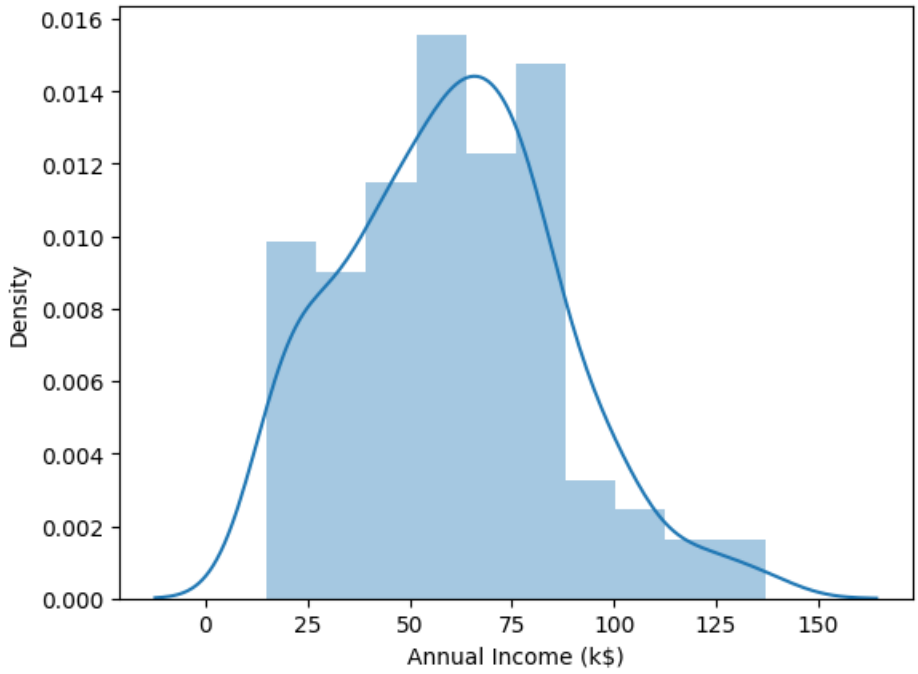
I made a bar plot to check the distribution of male and female population in the dataset. The female population clearly outweighs the male counterpart.



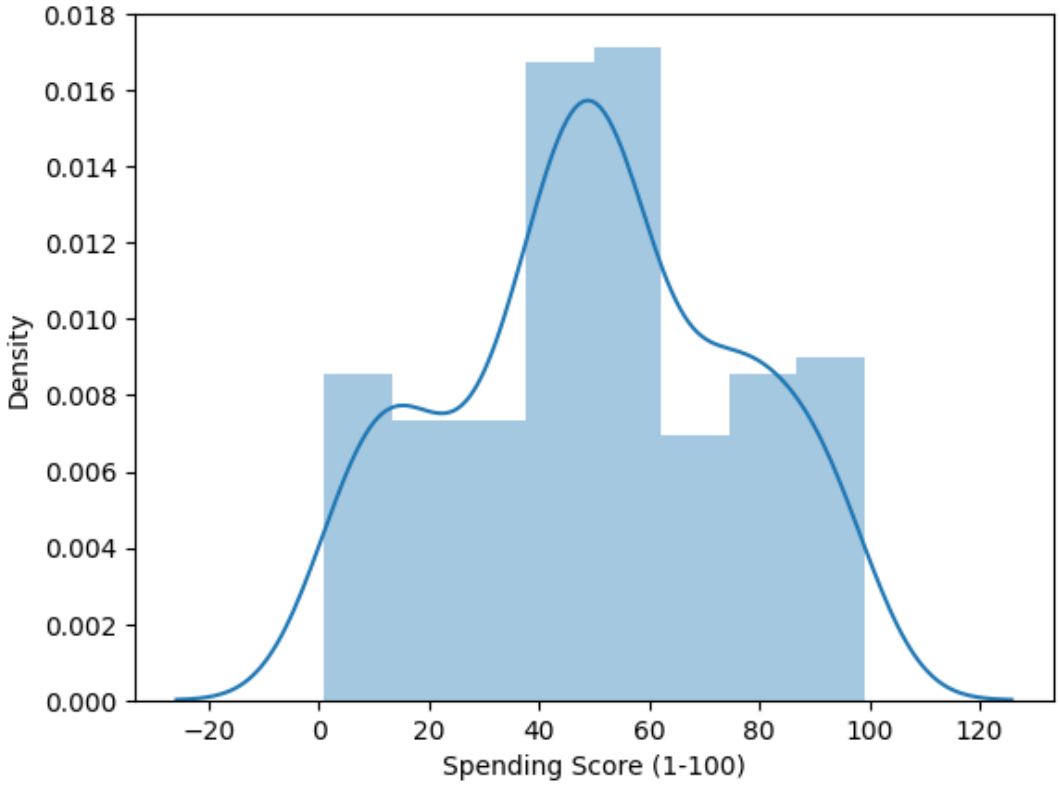
Next, I made create a distribution plot (histogram) for the 'Age' column.



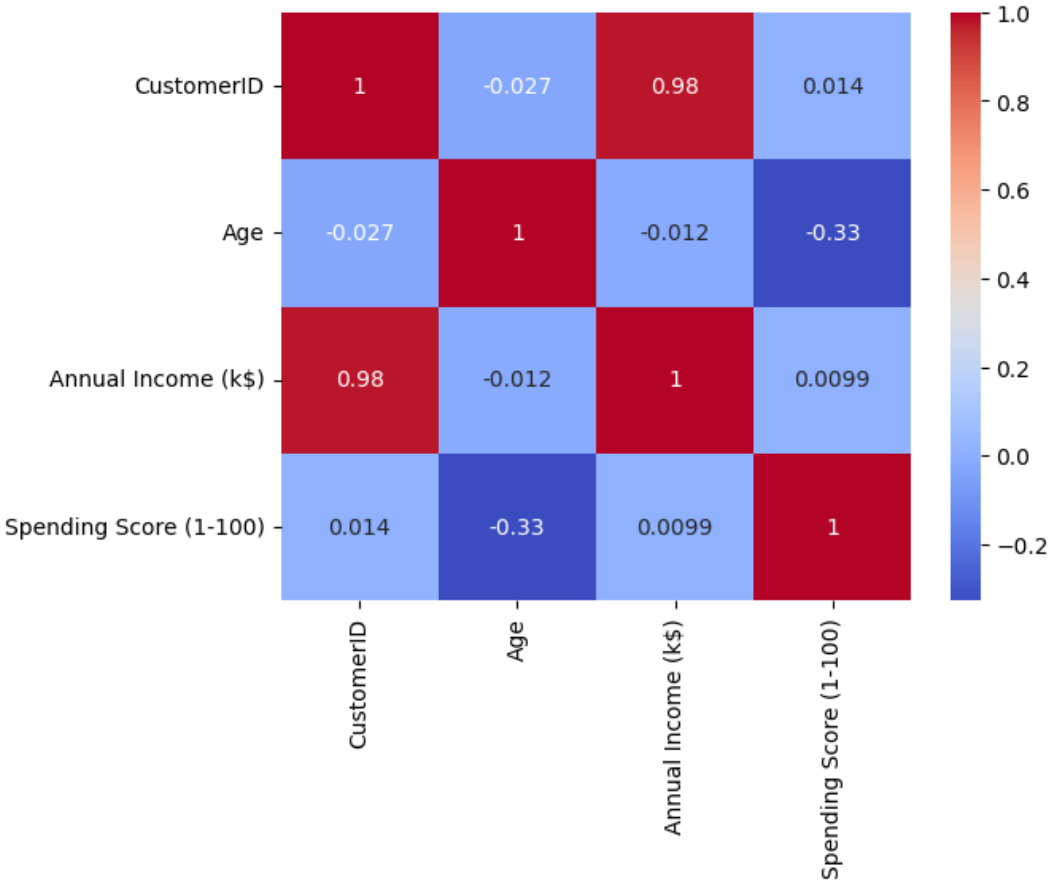
Then, I made create a distribution plot (histogram) for the ' Annual Income' column.

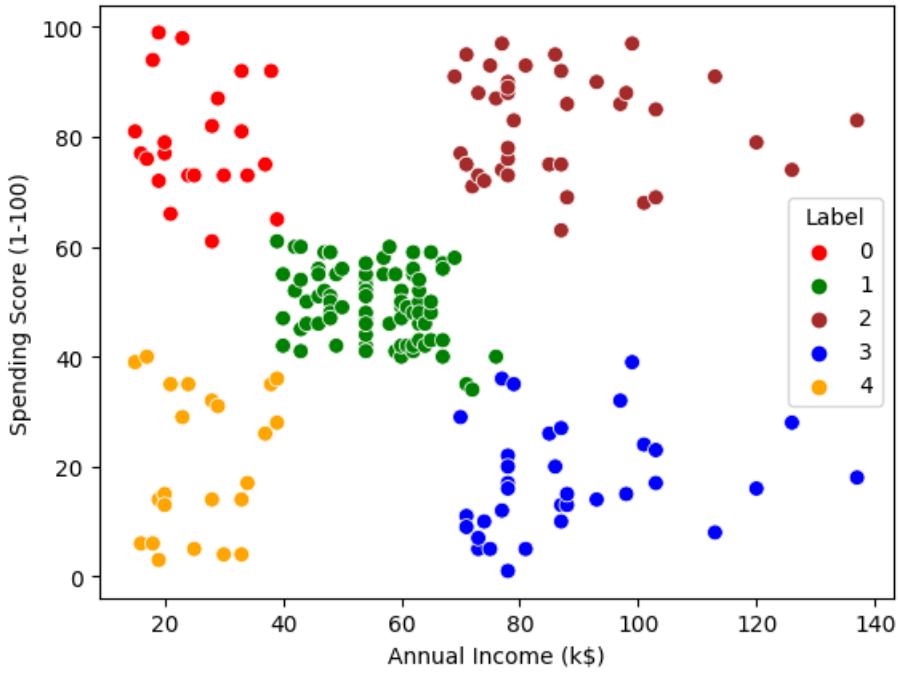


Next, I made create a distribution plot (histogram) for the 'Spending score' column.



Then I create a correlation Matrix to visualize the pairwise correlations between numerical variables and annotates the cells with correlation coefficients. And scatterplot.


CustomerID
Age
Annual Income (k$)
Spending Score (1-100)
1
-0.027
0.98
0.014
-0.027
1
-0.012
-0.33
0.98
-0.012
1
0.0099
0.014
-0.33
0.0099
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Spending Score (1-100)
Annual Income (k$)
Label
0
1
2
3
4
0
20
40
60
80
100
20
40
60
80
100
120
140

Next I plotted Within Cluster Sum Of Squares (WCSS) against the the number of clusters (K Value) to figure out the optimal number of clusters value. WCSS measures sum of distances of observations from their cluster centroids which is given by the below formula.

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$

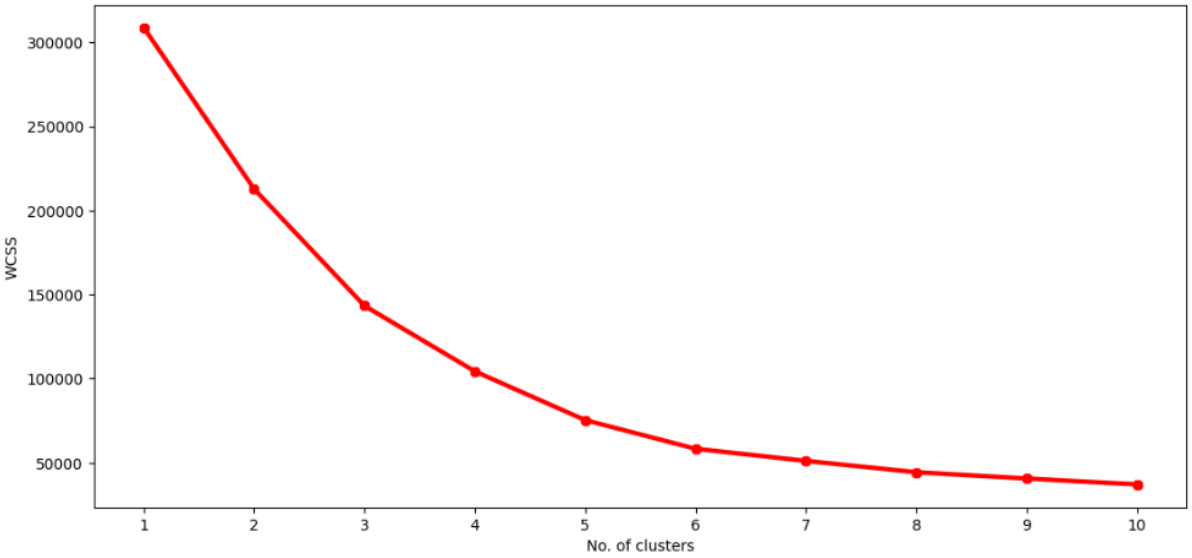
where *Yi* is centroid for observation *Xi*. The main goal is to maximize number of clusters and in limiting case each data point becomes its own cluster centroid.

The Elbow Method

Calculate the Within Cluster Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS first starts to diminish. In the plot of WSS-versus k, this is visible as an elbow.

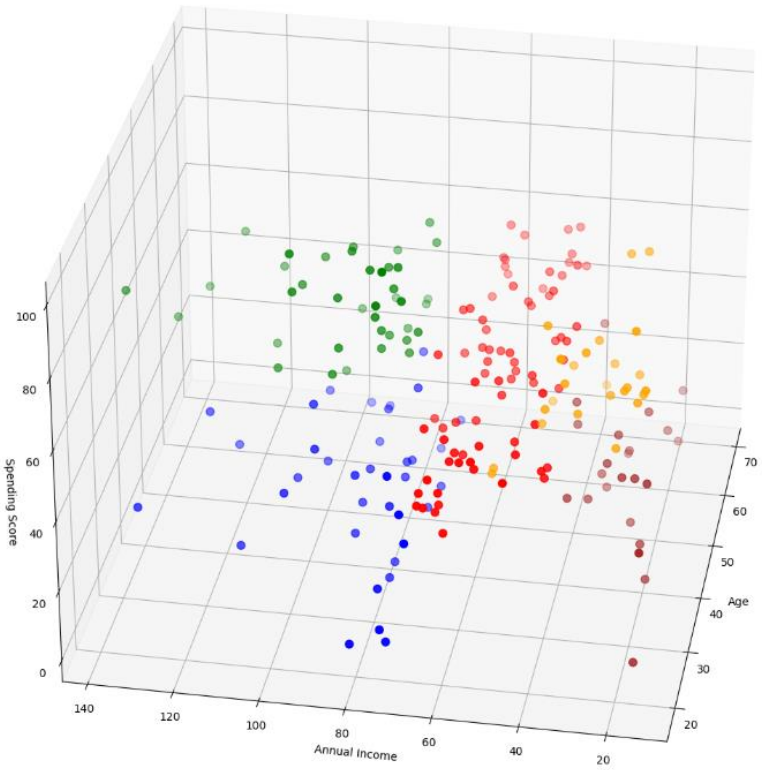
The steps can be summarized in the below steps:

1. Compute K-Means clustering for different values of K by varying K from 1 to 10 clusters.
2. For each K, calculate the total within-cluster sum of square (WCSS).
3. Plot the curve of WCSS vs the number of clusters K.
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.



The optimal K value is found to be 5 using the elbow method. Finally, I made a 3D plot to visualize the spending score of the customers with their annual income. The data points are separated into 5 classes which are represented in different colors as shown in the 3D plot.

Results



Conclusions

K means clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major applications of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.

PART (B)

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

Hierarchical Clustering:

Hierarchical clustering is a unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.

There are mainly two types of hierarchical clustering:

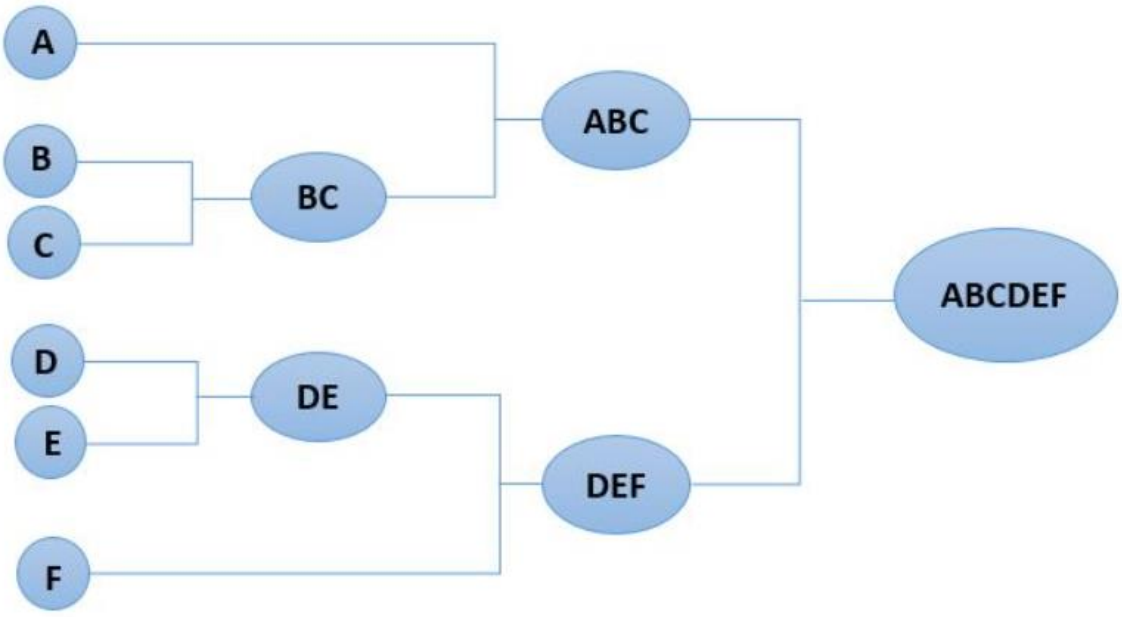
Agglomerative hierarchical clustering

Divisive Hierarchical clustering

Agglomerative hierarchical clustering

It's a Bottom to Up approach clustering technique. In this initially we assign each points to be a individual clusters.

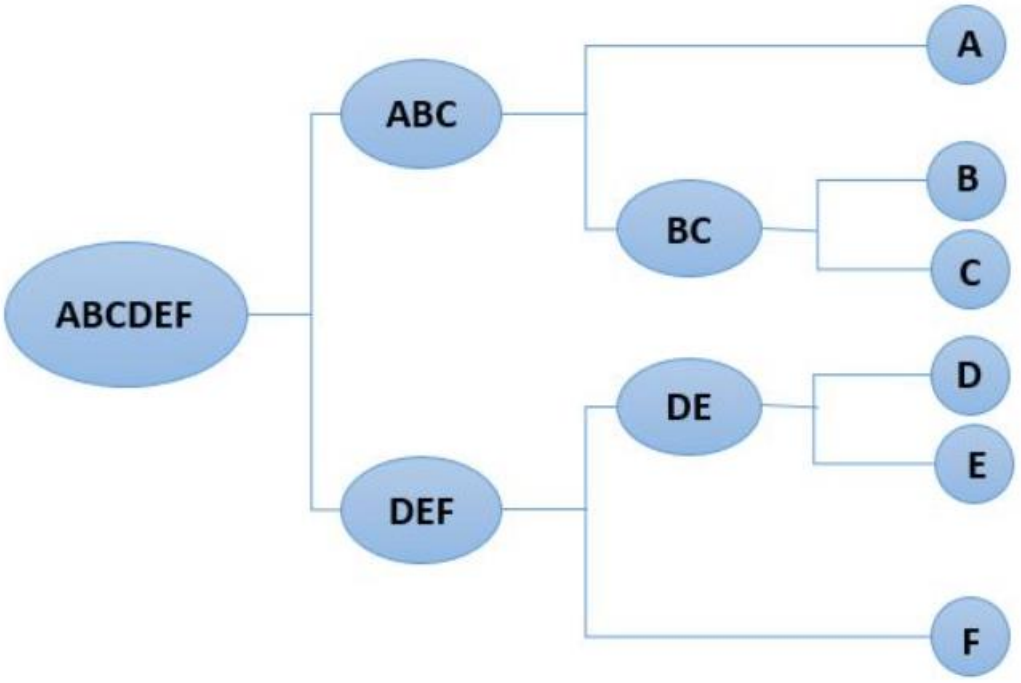
Then, at each iteration, we marge the closest pair of clusters and repeat this step until onlt a single cluster is left. Since at each step we merge the closest clusters. So, it's also known as additive hierarchical clustering.



Divisive Hierarchical clustering

It's a Top to Down Approach clustering technique. It works in opposite way. Instead of starting with 'n' clusters, it starts with a single cluster Containing all points.

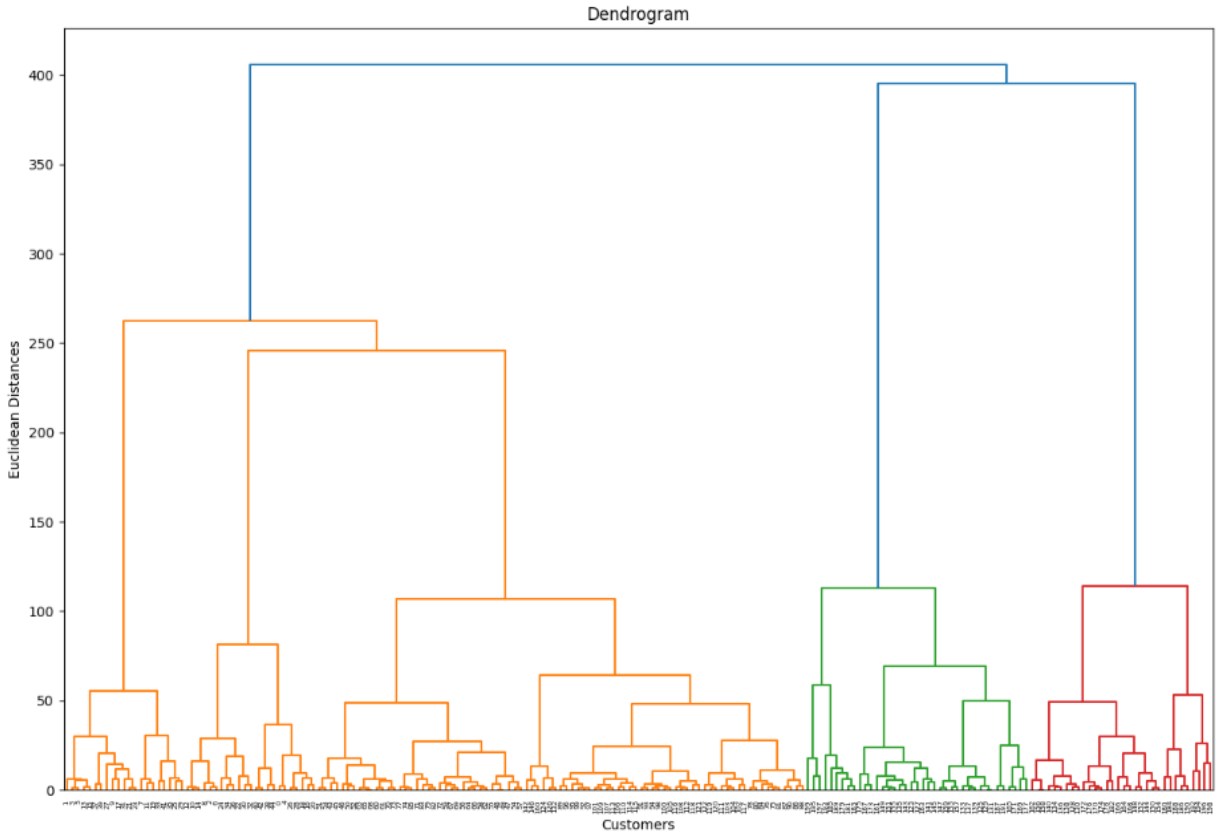
At each iteration, we split the farthest point in the cluster and repeat this process until each cluster only contains a single point. We are splitting (or dividing) the clusters at each step, hence the name divisive hierarchical clustering.

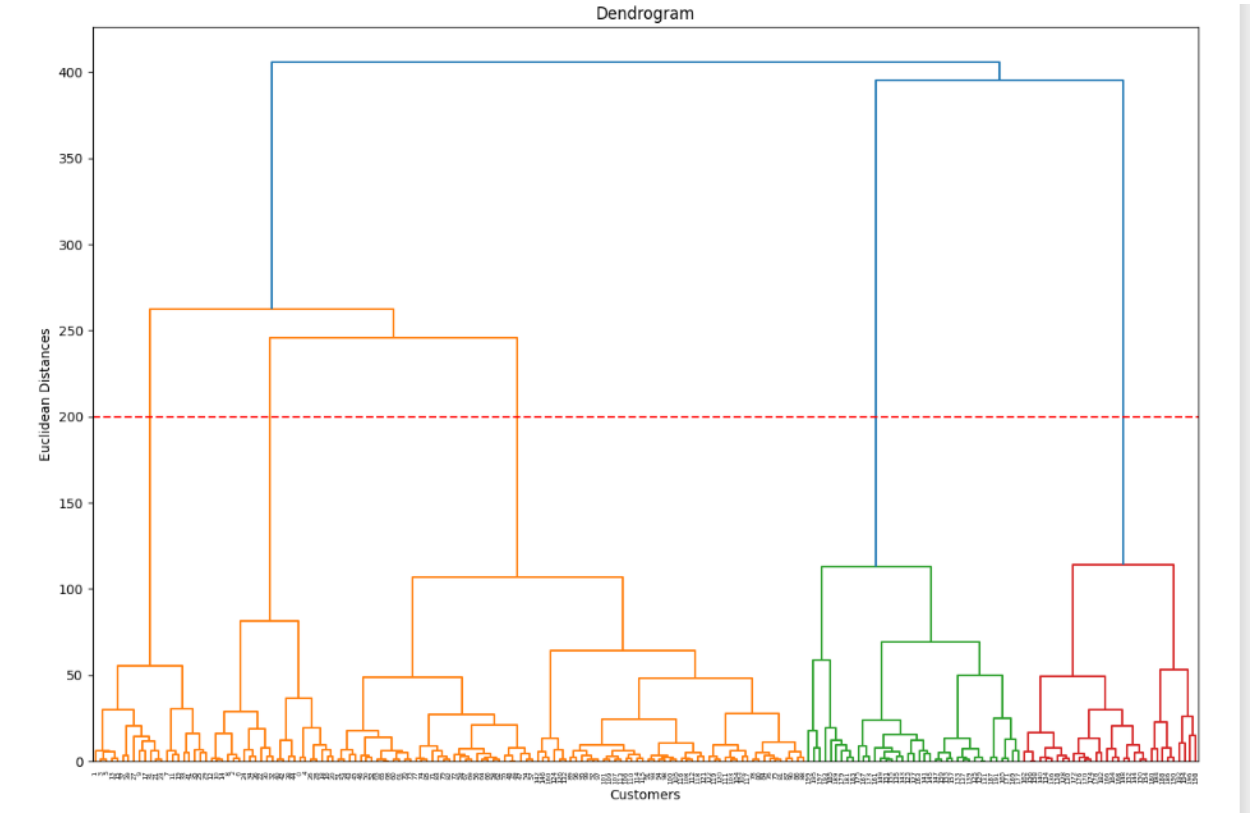


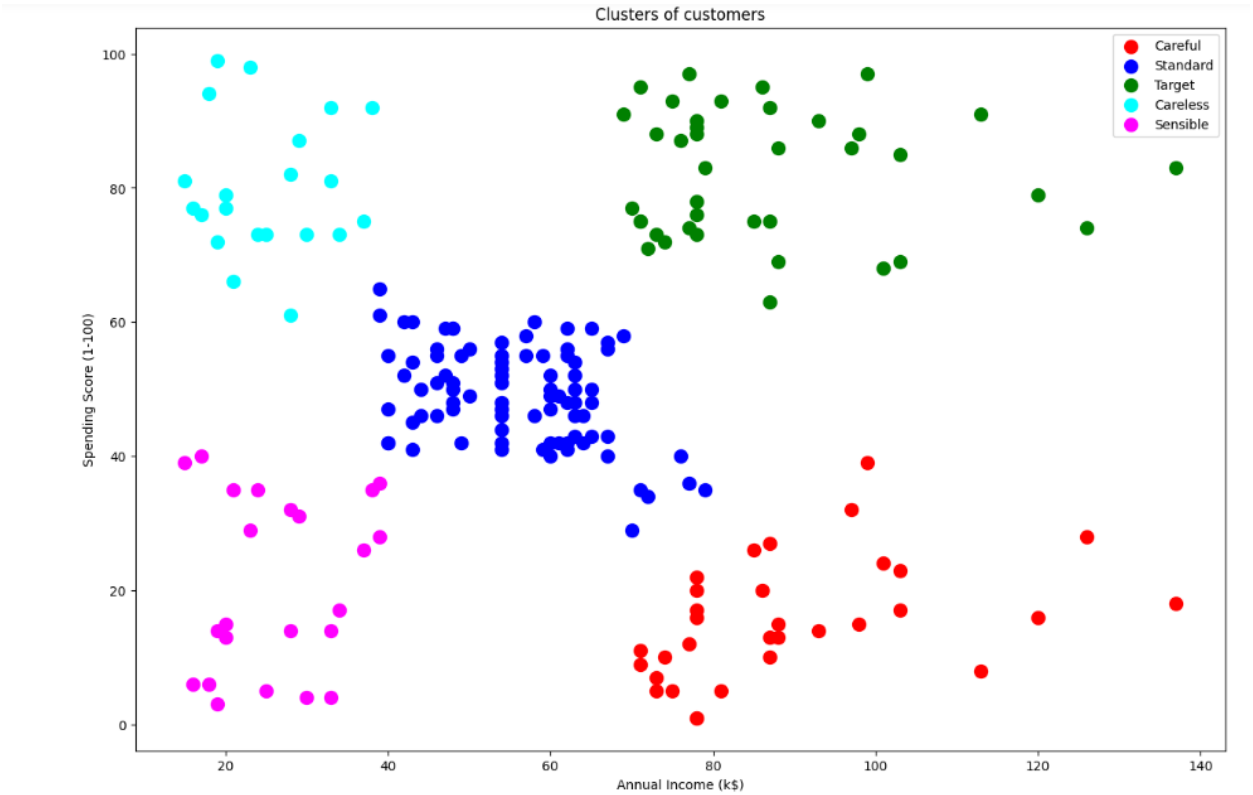
Proximity matrix

Proximity matrix is a square matrix of shape n X n where n is a few observations. It contains the distance of each point from each other points. We use Euclidean distance formula to calculate the distance.

Result:




Dendrogram
Euclidean Distances
Customers
0
50
100
150
200
250
300
350
400


Clusters of customers
Careful
Standard
Target
Careless
Sensible
Spending Score (1-100)
Annual Income (k$)
0
20
40
60
80
100
120
140

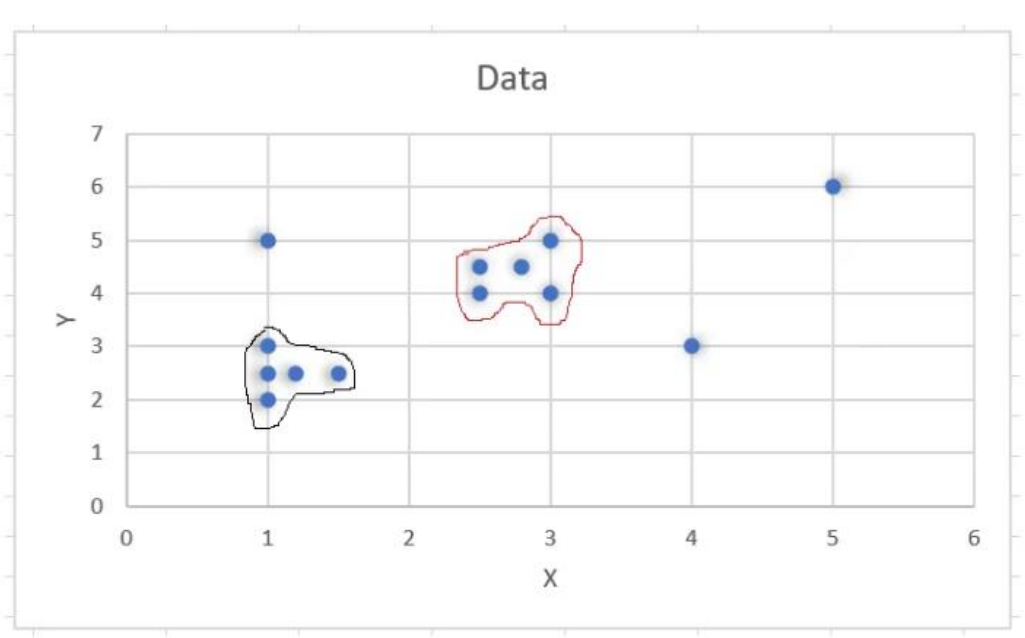
PART (C)

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

DBSCAN stands for Density-Based Spatial Clustering Application with Noise. It is an unsupervised machine learning algorithm that makes clusters based upon the density of the data points or how close the data is. That said, the points which are outside the dense regions are excluded and treated as noise or outliers.

Let's take a dataset of 13 points as shown and plotted below:

X	Y
1	2
3	4
2.5	4
1.5	2.5
3	5
2.8	4.5
2.5	4.5
1.2	2.5
1	3
1	5
1	2.5
5	6
4	3

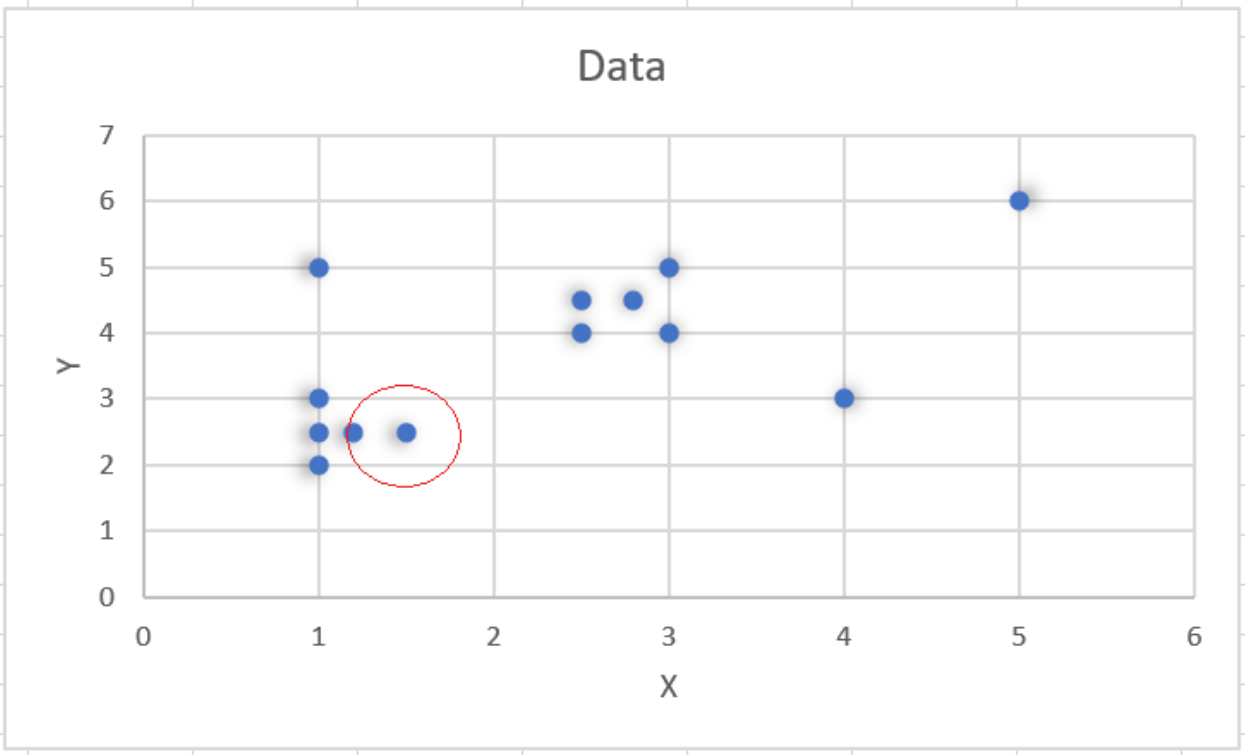


A two-dimensional data is presented for easy visualization and understanding, else DBSCAN can handle multi-dimensional data too. The possible clusters from the data have been marked in the above graph to visualize the clusters that we want. The

points (1,5) (4,3) (5,6) in the above graph fall outside the markings and hence should be treated as outliers. The DBSCAN algorithm should actually make clusters and exclude outliers as we did in the graph. Let's first understand the algorithm and various steps involved in it.

Logic and Steps:

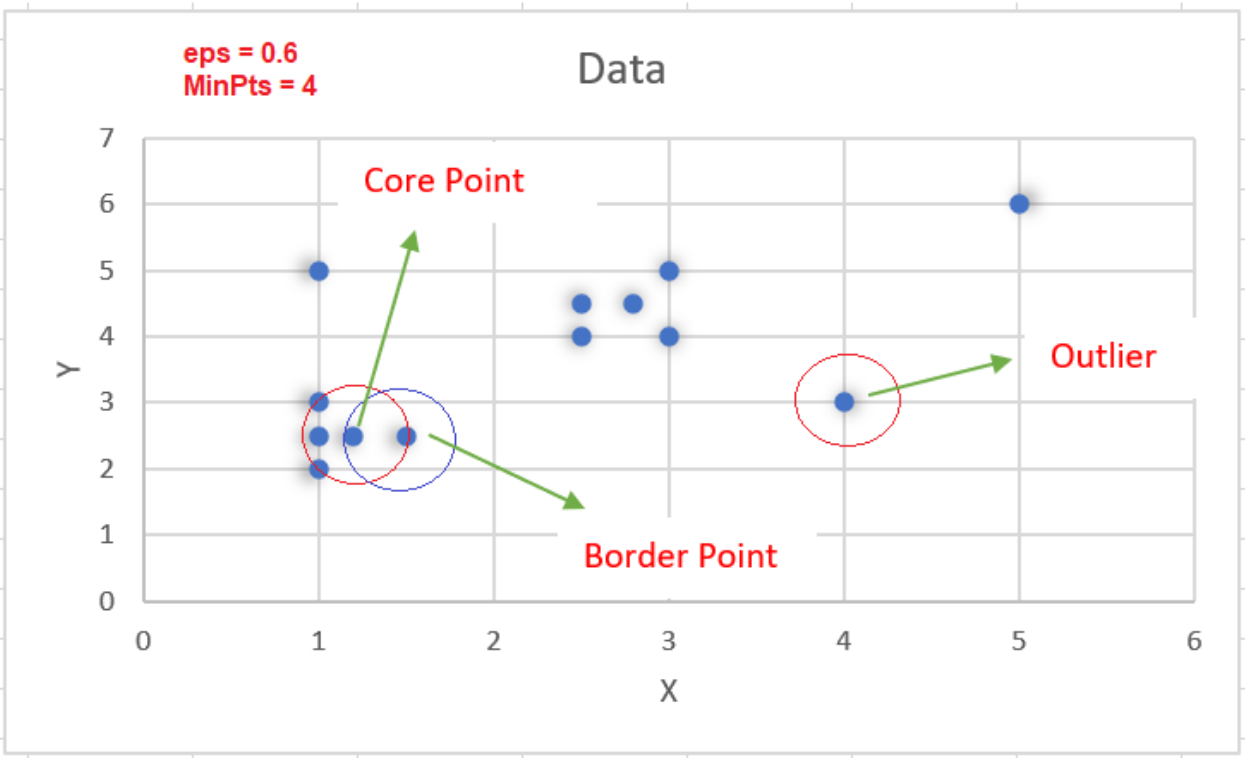
The DBSCAN algorithm takes two input parameters. Radius around each point (*eps*) and the minimum number of data points that should be around that point within that radius (*MinPts*). For example, consider the point (1.5,2.5), if we take *eps* = 0.3, then the circle around the point with radius = 0.3, will contain only one other point inside it (1.2,2.5) as shown below:



Hence for (1.5, 2.5) when *eps* = 0.3, the number of neighbourhood point(s) is just one. In DBSCAN each point is checked for these two parameters and the decision about the clustering is made as described through the below steps:

1. Choose a value for *eps* and *MinPts*
2. For a particular data point (**x**) calculate its distance from every other datapoint.
3. Find all the neighbourhood points of **x** which fall inside the circle of radius (*eps*) or simply whose distance from **x** is smaller than or equal to *eps.*
4. Treat **x** as **visited** and if the number of neighbourhood points around **x** are greater or equal to *MinPts* then treat **x** as a **core point** and if it is not assigned to any cluster, create a new cluster and assign it to that.
5. If the number of neighbourhood points around **x** are less than MinPts and it has a core point in its neighbourhood, treat it as a border point.
6. Include all the **density connected points** as a single cluster. (What density connected points mean is described later)
7. Repeat the above steps for every unvisited point in the data set and find out all core, border and outlier points.

If the number of neighborhood points around **x** is greater or equal to MinPts then **x** is treated as a core point, if the neighbourhood points around **x** are less than MinPts but is close to a core point then **x** is treated as a border point. If **x** is neither core nor border point then **x** is treated as an outlier. The below graph gives an idea about it. We choose eps = 0.6 and MinPts =4, the point tagged as core point has 4 other points (>= MinPts) in its neighbourhood & the one tagged as border point is in the neighbourhood of a core point but has only one point in its neighbourhood (< MinPts). The outlier point is one which is neither a border point nor core point.



Algorithm in action

Let's now apply the DBSCAN algorithm to the above dataset to find out clusters. We have to choose first the values for *eps* and *MinPts*. Let's choose *eps* = 0.6 and *MinPts* = 4. Let's consider the first data point in the dataset (1,2) & calculate its distance from every other data point in the data set. The Calculated values are shown below:

X	Y	Distance from (1,2)
1	2	0
3	4	2.8
2.5	4	2.5
1.5	2.5	0.7
3	5	3.6
2.8	4.5	3.08
2.5	4.5	2.9
1.2	2.5	0.53
1	3	1
1	5	3
1	2.5	0.5
5	6	5.6
4	3	3.1

As evident from the above table, the point (1, 2) has only two other points in its neighbourhood (1, 2.5), (1.2, 2.5) for the assumed value of *eps*, as its less than MinPts, we can't declare it as a core point. Let's repeat the above process for every point in the dataset and find out the neighbourhood of each. The calculations when repeated can be summarized as below:

Point	Neighbourhood Points				
(1,2)	(1.2, 2.5)	(1, 2.5)			
(3, 4)	(2.5, 4)	(2.8, 4.5)			
(2.5, 4)	(3, 4)	(2.8, 4.5)	(2.5, 4.5)		
(1.5, 2.5)	(1.2, 2.5)	(1, 2.5)			
(3, 5)	(2.8, 4.5)				
(2.8, 4.5)	(3, 4)	(2.5, 4)	(3, 5)	(2.5, 4.5)	Cluster 1
(2.5, 4.5)	(2.5, 4)	(2.8, 4.5)			
(1.2, 2.5)	**(1, 2)**	(1.5, 2.5)	(1, 3)	(1, 2.5)	Cluster 2
(1, 3)	(1.2, 2.5)	(1, 2.5)			
(1, 5)					
(1, 2.5)	**(1, 2)**	(1.5, 2.5)	(1.2, 2.5)	(1, 3)	Cluster 2
(5, 6)					
(4, 3)					

Observe the above table carefully, the left-most column contains all the points we have in our data set. To the right of them are the data points which are there in their neighbourhood i.e. the points whose distance from them is less or equal to the *eps* value. There are three points in the data set, (2.8, 4.5) (1.2, 2.5) (1, 2.5) that have 4 neighbourhood points around them, hence they would be called core points and as already mentioned, if the core point is not assigned to any cluster, a new cluster is formed. Hence, (2.8, 4.5) is assigned to a new cluster, Cluster 1 and so is the point (1.2, 2.5), Cluster 2. Also observe that the core points (1.2, 2.5) and (1, 2.5) share at least one common neighbourhood point (1,2) so, they are assigned to the same cluster. The below table shows the categorization of all the data points into core, border and outlier points. Have a look:

Point	Neighbourhood Points					
(1,2)	(1.2, 2.5)	(1, 2.5)			Border Point	
(3, 4)	(2.5, 4)	(2.8, 4.5)			Border Point	
(2.5, 4)	(3, 4)	(2.8, 4.5)	(2.5, 4.5)		Border Point	
(1.5, 2.5)	(1.2, 2.5)	(1, 2.5)			Border Point	
(3, 5)	(2.8, 4.5)				Border Point	
(2.8, 4.5)	(3, 4)	(2.5, 4)	(3, 5)	(2.5, 4.5)	**Core Point**	Cluster 1
(2.5, 4.5)	(2.5, 4)	(2.8, 4.5)			Border Point	
(1.2, 2.5)	**(1, 2)**	(1.5, 2.5)	(1, 3)	(1, 2.5)	**Core Point**	Cluster 2
(1, 3)	(1.2, 2.5)	(1, 2.5)			Border Point	
(1, 5)					Outlier	
(1, 2.5)	**(1, 2)**	(1.5, 2.5)	(1.2, 2.5)	(1, 3)	**Core Point**	Cluster 2
(5, 6)					Outlier	
(4, 3)					Outlier	

There are three types of points in the dataset as detected by the DBSCAN algorithm, core, border and outliers. Every core point will be assigned to a new cluster unless some of the core points share neighbourhood points, they will be included in the same cluster. Every border point will be assigned to the cluster-based upon the core point in its neighbourhood e.g. the first point (1, 2) is a border point and has a core point (1.2, 2.5) in its neighbourhood, which is included in Cluster 2, hence, the point (1,2) will be included in the Cluster 2 too. The whole categorization can be summarized as below:

Cluster 1	Cluster 2	Outliers
(3,4)	(1, 2)	(1, 5)
(2.5, 4)	(1.5, 2.5)	(5, 6)
(3,5)	(1.2, 2.5)	(4, 3)
(2.8, 4.5)	(1, 3)	
(2.5, 4.5)	(1, 2.5)	

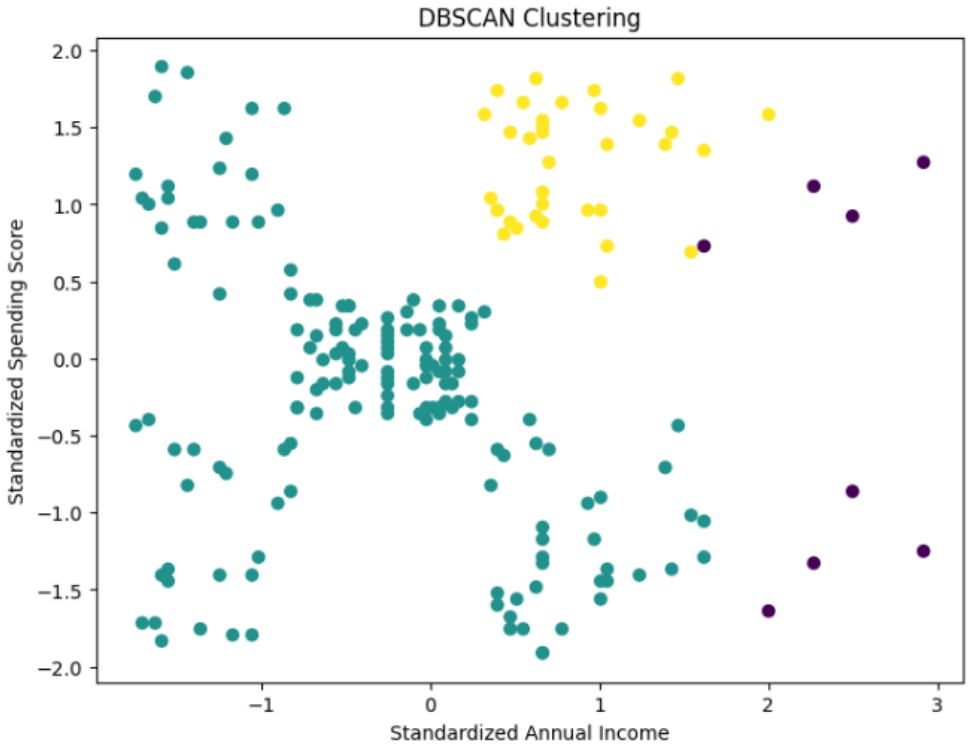
Three terms are necessary to understand in order to understand DBSCAN:

1. **Direct density reachable**: A point is called direct density reachable if it has a core point in its neighbourhood. Consider the point (1, 2), it has a core point (1.2, 2.5) in its neighbourhood, hence, it will be a direct density reachable point.

2. **Density Reachable**: A point is called density reachable from another point if they are connected through a series of core points. For example, consider the points (1, 3) and (1.5, 2.5), since they are connected through a core point (1.2, 2.5), they are called density reachable from each other.

3. **Density Connected**: Two points are called density connected if there is a core point which is density reachable from both the points.

Result:



GitHub Link: https://github.com/Shahriar-Hossain-Opu/Data-Science-Clustering

–End of the Report–