**FORMAN CHRISTIAN COLLEGE (A CHARTERED UNIVERSITY)**

****

**COMP 455 – Data Mining and Warehousing**

**FALL 2022**

**Gulraiz Noor Bari (231-525536)**

**Assignment #2**

**CLUSTERING - KMEAN & DBSCAN**

**DATASET:**

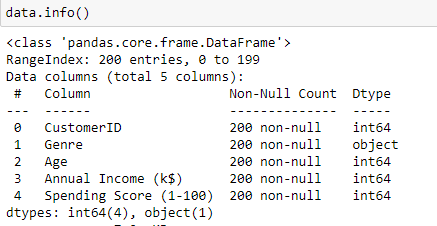
The data set used to implement clustering using K-Mean and DBSCAN algorithm has attributes; CustomerID, gender, age, annual income (k$), spending score on the scale of (1-100).

Table

Description automatically generated

In **head(int n),** n is the number of first n rows of the dataset you want to display. In our case it is 5.

The dataset has 5 attributes and 200 tuples, representing the data of 200 customers or we can also say that it has 5 columns and 200 rows. No attribute has Null value.



**RELATION BETWEEN ANNUAL INCOME AND SPENDING SCORE:**

Plotting a Scatter Plot between the values of both these attributes show that there is no relation between the customers having annual income in the range of 20-40k and 80-140k but there is relation between customers who have annual income in the range of 40-60k.



**FEATURE SCALING ON SPENDING SCORE AND ANNUAL INCOME VALUES ARRAY:**

Feature Scaling is an important step during the pre-processing of data before creating a machine learning model. It is used to shrink the features present in the dataset in a fixed range, which helps to handle highly varying values especially in algorithms that involve distance measurement. There are different ways to perform Feature Scaling, in our case we have used the following methods on our data array containing features of Spending Score and Annual Income:

* **Min-Max Scaler:**

Re-scales the feature values to a specified range. For example, [-1, 1], [-2, 9], etc.

Text

Description automatically generated with medium confidence

Table

Description automatically generatedIn the above code, the range for Min-max scaling is [0,1], which means all the feature values (in data\_array) will be adjusted between 0 and 1.

* **Standardization/Standard Scaler:**

Re-scales the feature values such that the distribution is around 0 and standard deviation is 1

Graphical user interface, text

Description automatically generated

Text

Description automatically generated and so on…

**ELBOW METHOD:**

The elbow method is a graphical representation of finding the optimal 'K' in a K-mean clustering. It works by finding Intra-cluster Sum of Square i.e., the sum of the square distance between points in a cluster and the cluster centroid.

We assume that 10 clusters will be formed and run K-Mean algorithm for each number of clusters till 10. Fit() trains the running K-Mean algorithm on our min max scaled data array. Inertia\_ is calculated by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares across one cluster. It basically tells us how well our clustering was performed.

Graphical user interface, text

Description automatically generated

**OUTPUT**

Chart, line chart

Description automatically generated

**Apply K-Means with the value of K found in last step:**

From the above graph we can observe that the last breakpoint in the curve occurs at 5. So, we can assume number of clusters (K) as 5 and apply the K-mean algorithm on our feature scaled data.

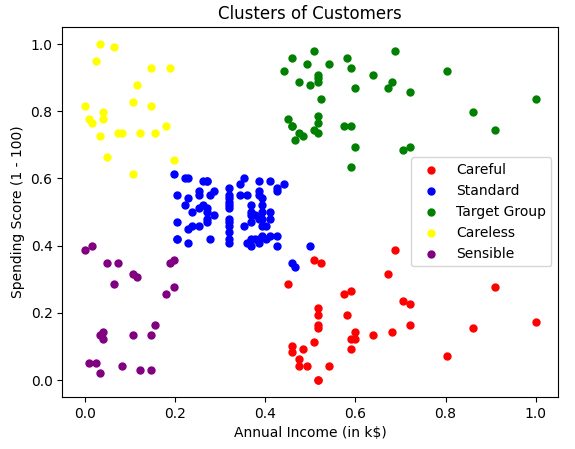
**Fit\_predict(min\_max\_scaled\_data)** will performs clustering on min\_max\_scaled\_data (or any other provided array) and returns cluster labels for each sample.

Text

Description automatically generated

**VISUALIZING THE CLUSTERS:**

Visualizing the data points using a scatter plot:

****

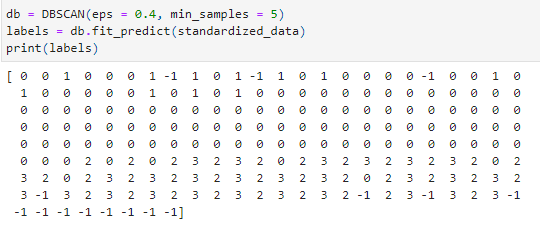
Generated clusters represent the following information:

* Red Color (Careful); means Customers are earning high but spending less
* Blue Color (Standard); means Customers are average in terms of earning and spending
* Green Color (Target Group); means Customers are earning and spending high
* Yellow Color (Careless); means Customers are earning less but spending more
* Purple Color (Sensible); means Customers are earning and spending less

**APPLYING DBSCAN (DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE) ALGORITHM:**

* Epsilon = 0.4
* Min Samples = 5

We apply DBSCAN Algorithm on the standardized data array.

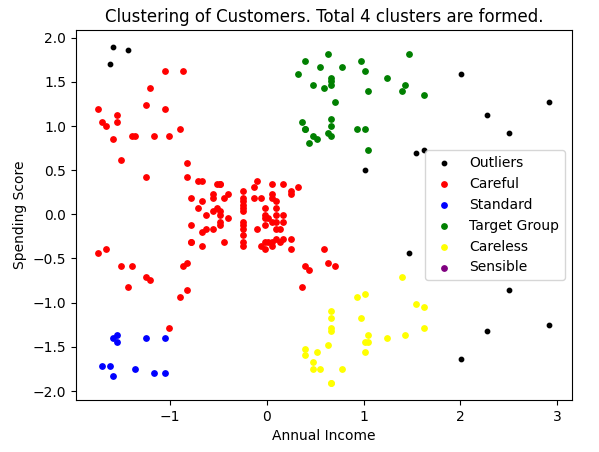


DBSCAN created 4 clusters

Graphical user interface, text, application

Description automatically generated

plus, outlier’s cluster (points that do not belong to any cluster) are represented by black dots. Sizes of clusters vary significantly - some have only 4 or 8 observations. There is total 15 outliers generated by the algorithm.



The graph above shows that there are some outliers - these points do not meet distance and minimum samples requirements to be recognized as a cluster. The cluster are not the most optimal ones as compared to which were formed by Kmean algorithm (As every point is in a cluster), but in DBSCAN there are many points that do not belong to any cluster.

One reason for DBSCAN not giving us optimal results for our dataset is because the density of our data is not that strong. DBSCAN may have performed better if we had had a bigger data.