ABSTRACT

CUSTOMER SEGEMENTATION

Customer Segmentation is the process of division of customer base into several groups of individuals that share a similarity in different ways that are relevant to marketing such as gender, age, interests, and miscellaneous spending habits.

Companies that deploy customer segmentation are under the notion that every customer has different requirements and require a specific marketing effort to address them appropriately. Companies aim to gain a deeper approach of the customer they are targeting. Therefore, their aim has to be specific and should be tailored to address the requirements of each and every individual customer. Furthermore, through the data collected, companies can gain a deeper understanding of customer preferences as well as the requirements for discovering valuable segments that would reap them maximum profit. This way, they can strategize their marketing techniques more efficiently and minimize the possibility of risk to their investment.

The technique of customer segmentation is dependent on several key differentiators that divide customers into groups to be targeted. Data related to demographics, geography, economic status as well as behavioural patterns play a crucial role in determining the company direction towards addressing the various segments.

Segmentation of market is an effective way to define and meet customer needs. Unsupervised Machine Learning Techniques, K-Means Clustering Algorithm, DBSCAN clustering method are used to perform Mall customer Analysis. Mall customer Analysis is carried out to predict the target customers who can be easily converged, among all the customers. In order to allow the marketing team to plan the strategy to market the new products to the target customers which are similar to their interests.

Key words: Target Customers, Clusters, Unsupervised Learning, K-Means, DBSCAN Clustering, Mall customer Analysis

CONTENTS

TABLE OF CONTENTS **PAGE NO.**

Declartion..... iii

Certificate……. iv

Acknowledgement……..... v

Abstract……..... vi

Chapter I Introduction..... 1-3

1.1 Types of Customer Segmentation……. 1

1.2 Purpose of Customer Segmentation……... 2

1.3 Process of Segmenting Customers……. 2

1.4 Using customer Segments……. 2-3

Chapter II Literature review……. 4-5

Chapter III Proposed system……. 6

Chapter IV Technology used……. 7

Chapter V Software required……..... 8

5.1 Software Requirements Specification……. 8

Chapter VI Methodology…… 9-22

6.1 Clustering……..... 9

6.2 K-means Clustering algorithm……..... 9

6.3 DBSCAN Clustering algorithm……... 9

6.4 Elbow method……. 9

6.5 General view of data……... 10

6.6 Reason to use unsupervised learning algorithms…… 10-14

6.7 K-means Clustering…… 14-18

6.8 DBSCAN Clustering algorithm……... 19-21

6.9 Comparison of results……..... 21

Chapter VII Conclusion and future scope…… 22
Chapter VIII References…… 23

FIGURES

FIGURE 1: K means clustering algorithm
FIGURE 2: DBSCAN Clustering Algorithm
FIGURE 3: Elbow Method
FIGURE 4: Importing libraries
FIGURE 5: Read csv file
FIGURE 6: df.head
FIGURE 7: df.shape
FIGURE 8: df.describe()
FIGURE 9: Remove null values
FIGURE 10: Comparison of number male and female in the dataset
FIGURE 11: Violin method for comparison in different attributes
FIGURE 12: Plot between age and customers
FIGURE 13: Plot between annual income and spending score
FIGURE 14: Plot between number of customers and spending score
FIGURE 15: Plot between annual income and spending score
FIGURE 16: K-means number of clusters
FIGURE 17: Scatter plot between Spending Score and Age
FIGURE 18: Optimal number of clusters for X2
FIGURE 19: Labelling of clusters for X2
FIGURE 20: Centroids of clusters for X2
FIGURE 21: Scatter plot between annual income and spending score
FIGURE 22: Optimal number of clusters for X3
FIGURE 23: Labelling of clusters for X3
FIGURE 24: Centroids of clusters for X3

FIGURE 25: Scatter Plot among Age, Annual Income, Spending Score
FIGURE 26: Silhouette score of X2
FIGURE 27: Silhouette score of X1
FIGURE 28: Silhouette score of X3
FIGURE 29: Importing Essential Libraries
FIGURE 30: Reading and visualizing data
FIGURE 31: Scatter plot between annual income and spending score
FIGURE 32: Scatter plot between age and spending score
FIGURE 33: Scatter Plot between age and annual income
FIGURE 34: Silhouette score in DBSCAN Clustering

CHAPTER-I
INTRODUCTION

Management and maintain of customer relationship have always played a vital role to provide business intelligence to organizations to build, manage and develop valuable long term customer relationships. The importance of treating customers as an organizations main asset is increasing in value in present day and era. Organizations have an interest to invest in the development of customer acquisition, maintenance and development strategies. The business intelligence has a vital role to play in allowing companies to use technical expertise to gain better customer knowledge and Programs for outreach. By using clustering techniques like k-means, customers with similar means are clustered together. Customer segmentation helps the marketing team to recognize and expose different customer segments that think differently and follow different purchasing strategies. Customer segmentation helps in figuring out the customers who vary in terms of preferences, expectations, desires and attributes. The main purpose of performing customer segmentation is to group people, who have similar interest so that the marketing team can converge in an effective marketing plan. Clustering is an iterative process of knowledge discovery from vast amounts of raw and unorganized data. Clustering is a type of exploratory data mining that is used in many applications, such as machine learning, classification and pattern recognition.

1.1 Types of Customer Segmentation

There are two types of customer segmentation:

- Business-to-Business
- Business-to-Customer

In business-to-business marketing, a company might segment customers according to a wide range of factors, including

- Industry
- Number of employees
- Products previously purchased from the company
- Location

In business-to-consumer marketing, companies often segment customers according to demographics that include:

- Age
- Gender
- Marital status
- Location (urban, suburban, rural)
- Life stage (single, married, divorced, empty-nester, retired, etc.)

1.2 Purpose of Customer Segmentation

Segmentation allows marketers to better tailor their marketing efforts to various audience subsets. Those efforts can relate to both communications and product development. Specifically, segmentation helps a company:

- Create and communicate targeted marketing messages that will resonate with specific groups of customers, but not with others (who will receive messages tailored to their needs and interests, instead).
- Select the best communication channel for the segment, which might be email, social media posts, radio advertising, or another approach, depending on the segment.
- Identify ways to improve products or new product or service opportunities.
- Establish better customer relationships.
- Test pricing options.
- Focus on the most profitable customers.
- Improve customer service.
- Upsell and cross-sell other products and services.

1.3 Process of Segmenting Customers

Customer segmentation requires a company to gather specific information – data – about customers and analyse it to identify patterns that can be used to create segments. Some of that can be gathered from purchasing information – job title, geography, products purchased, for example. Some of it might be gleaned from how the customer entered your system. An online marketer working from an opt-in email list might segment marketing messages according to the opt-in offer that attracted the customer, for example. Other information, however, including consumer demographics such as age and marital status, will need to be acquired in other ways.

Typical information-gathering methods include:

- Face-to-face or telephone interviews
- Surveys
- General research using published information about market categories
- Focus groups

1.4 Using Customer Segments

Common characteristics in customer segments can guide how a company markets to individual segments and what products or services it promotes to them. A small business selling hand-made guitars, for example, might decide to promote lower-priced products to younger guitarists and higher-priced premium guitars to older musicians based on segment

knowledge that tells them that younger musicians have less disposable income than their older counterparts. Similarly, a meals-by-mail service might emphasize convenience to millennial customers and “tastes-like-mother-used-to-make” benefits to baby boomers.

Customer segmentation can be practiced by all businesses regardless of size or industry and whether they sell online or in person. It begins with gathering and analysing data and ends with acting on the information gathered in a way that is appropriate and effective.

CHAPTER-II
LITERATURE REVIEW

Customer Segmentation

Over the years, as there is very strong competition in the business world, the organizations have to enhance their profits and business by satisfying the demands of their customers and attract new customers according to their needs. The identification of customers and satisfying the demands of each customer is a very complex task. This is because customers may be different according to their demands, desires, preferences and so on. Instead of "one-size-fits-all" approach, customer segmentation clusters the customers into groups sharing the same properties or behavioural characteristics. Customer segmentation is a strategy of dividing the market into homogenous groups. The data used in customer segmentation technique that divides the customers into groups depends on various factors like, demographical conditions, data geographical conditions and economic conditions as well as behavioural patterns. The customer segmentation technique allows the business to make better use of their marketing budgets, gain a competitive edge over their rival companies, demonstrating the better knowledge of the needs of the customer. It also helps an organization in, increasing their marketing efficiency, plan the marketing budget, determining new market opportunities, making better brand strategy, identifying customers retention.

Decision makers use many variables to segment customers. Demographic variables such as age, gender, family, education level and income are the easiest and common variables for segmentation. Socio- cultural, geographic, psychographic and behavioural variables are the other major variables that are used for segmentation. Presented various clustering algorithms taking into account the characteristics of Big Data such as size, noise, dimensionality, algorithm calculations, cluster shape and presented a brief overview of the various clustering algorithms grouped under partitioning, hierarchical, density, grid-based and model-based algorithms.

Explored the necessity of segmentation of the customers using clustering algorithms as the core functionality of CRM. The mostly used K-Means and Hierarchical Clustering were studied and the advantages and disadvantages of these techniques were highlighted. At last, the idea of creating a hybrid approach is addressed by integrating the above two strategies with the potential to surpass the individual designs. Merged clustering of fuzzy c-means and genetic algorithms to cluster, steel industry customers, by using the LRFM variables (length, recency, frequency, monetary value) system, customers were divided into two clusters.

Clustering and K-Means Algorithm

Clustering algorithms generates clusters such that within the clusters are similar based on some characteristics. Similarity is defined in terms of how close the objects are in space.

K-means algorithm in one of the most popular centroid based algorithms. Suppose data set, D, contains n objects in space. Partitioning methods distribute the objects in D into k clusters, C1,……Ck , that is, $Ci \subset D$ and $Ci \cap Cj = \emptyset$ for $(1 \leq i, j \leq k)$. A centroid-based partitioning technique uses the centroid of a cluster, Ci, to represent that cluster. Conceptually, the

centroid of a cluster is its centre point. The difference between an object p ∈ Ci and ci, the representative

of the cluster, is measured by dist(p,ci), where dist(x,y) is the Euclidean distance between two points x and y.

Algorithm: The k-means algorithm for partitioning, where each cluster's centre is represented by the mean value of the objects in the cluster. Input: k: the number of clusters, D: a data set containing n objects. Output: A set of k clusters. Method: Arbitrarily choose k objects from D as the initial cluster centres; repeat (re)assigns each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster; update the cluster means, that is, calculate the mean value of the objects for each cluster; until no change.

CHAPTER-III
PROPOSED SYSTEM

We are going to aim to cluster a data set that is about behaviour of the customers having credit card using many unsupervised algorithms.

Our research question is "How many clusters can we distinguish the customers according to their transactions or behaviours?"

General View of Data

The data set has 8950 transactions or information about account that belong to customers.

Features

CUSTID: Identification of Customer

Age: Age of the customer

Gender: Gender of the Customers

Annual Income: Annual income of the Customers

Spending Score: Spending score of the Customers

CHAPTER-IV
TECHNOLOGY USED

Machine learning: Machine learning models can process customer data and discover recurring patterns across various features. In many cases, machine learning algorithms can **help** marketing analysts find customer segments that would be very difficult to spot through intuition and manual examination of data

- Design A Proper Business Case
- Collect & Prepare the Data
- Performing Segmentation Using k-Means Clustering and DBSCAN clustering
- Tuning The Optimal Hyperparameters for The Model
- Visualization of the Results.

Python: Python coding for

- Data pre-processing for clustering.
- Building a clustering algorithm from scratch.
- The metrics used to evaluate the performance of a clustering model.
- Visualizing clusters built.
- Interpretation and analysis of clusters built.

Google Collab : We used google collab for executing the program

CHAPTER-V
SOFTWARE REQUIRED

5.1 Software Requirements Specification

5.1.1 Python:

- Python 3

5.1.2 Libraries

- NumPy

- Sklearn
- Matplot
- Pandas etc…

5.1.3 Operating System

- Windows or Ubuntu

CHAPTER-V1
METHODOLOGY

6.1 Clustering

Clustering is one of the most common methods used in exploring data to obtain a clear understanding of the data structure. It can be characterized as the task of finding the subtitles and subgroups in the complete dataset. Similar data is clustered in many subgroups. A cluster refers to a collection of aggregated data points due to some similarities. Clustering is used in Market basket analysis used to segment the customers based on their behaviours and transactions.

6.2 K-Means Clustering Algorithm

K Means Clustering is the most common and simplest Machine learning algorithm and it follows an iterative approach which attempts to partition the dataset into different “k” number of predefined and non-overlapping subgroups where each data point belongs to only one subgroup according to their similar qualities.

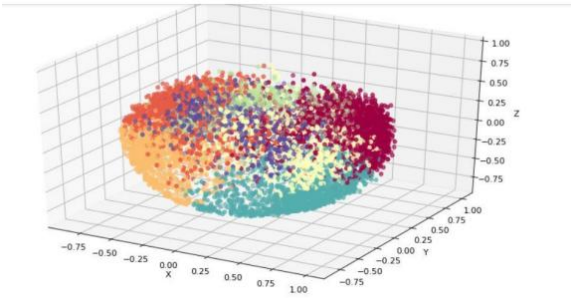


Figure 1 :-k-means clustering

6.3 DBSCAN Clustering Algorithm

DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density. It groups 'densely grouped' data points into a single cluster.

6.4 Elbow Method

Elbow method is a tool used for analysing the clusters formed from our dataset and helps to interpret the appropriate number of optimal clusters in dataset. From this method the optimal number of clusters for our dataset is found to be seven.

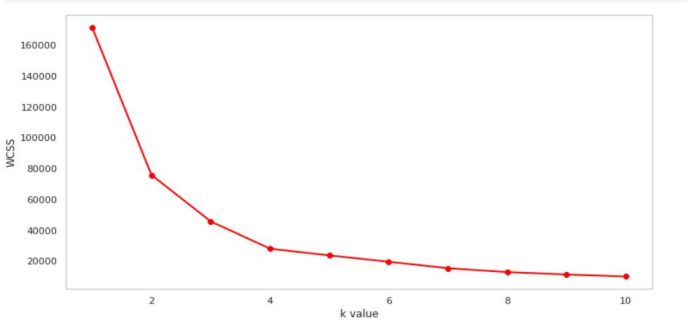


Figure 2:-Elbow method

6.5 General View of Data

The data set has 200 customers information about account that belong to customers.

Features

CUSTID: Identification of Customer

Age: Age of the customer

Gender: Gender of the Customers

Annual Income: Annual income of the Customers

Spending Score: Spending score of the Customers

6.6 Reason to use Unsupervised Learning Algorithms

Unlike Supervised Learning, Unsupervised Learning has only independent variables and no corresponding target variable. The data is unlabelled. The aim of unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.

We are going to examine a dataset that is about mall visitors for segmentation. There is no any feature about label of customers. That is to say, we don't have information about customer's characteristics. We are going to try clustering clients through identifying similarities with machine learning algorithms. Segmentation of customers has a pretty significant position for companies in new marketing disciplines. Firms must reach to the right target audiences with right approaches because of costs.

First of all, we have to import all necessary libraries

```
[45] import pandas as pd
     import numpy as np
     import io
     import matplotlib.pyplot as plt
     import seaborn as sns
     from kneed import KneeLocator
     from sklearn.datasets import make_blobs
     from sklearn.cluster import KMeans
     from sklearn.metrics import silhouette_score
     from sklearn.preprocessing import StandardScaler
     from sklearn.decomposition import PCA
     from mpl_toolkits.mplot3d import Axes3D
     from sklearn.cluster import MiniBatchKMeans
     import scipy.cluster.hierarchy as shc
     from sklearn.cluster import AgglomerativeClustering
     from sklearn import metrics
     # reading the data frame
     from google.colab import files
     uploaded = files.upload()

     Choose Files  Mall_Customers.csv
     • Mall_Customers.csv(text/csv) - 3981 bytes, last modified: 5/9/2022 - 100% done
     Saving Mall_Customers.csv to Mall_Customers (1).csv
```

Figure 3: importing libraries

We can use following code to read data from .csv file.

```
[6] df = pd.read_csv(io.BytesIO(uploaded['Mall_Customers.csv']))
    print(df)

     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
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83

[200 rows x 5 columns]
```

Figure 4: read csv file

We can use df.head() to see first 5 data records from dataset.

```
[47] df.head(  )
```

	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

Figure 5:df.head()

We can use following code to get information regarding dataset.

```
df.shape

(200, 5)
```

Figure 6:df.shape()

To get information regarding each column, we can use df.describe () as follows.

```
[49] df.describe(  )
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Figure 7: df.describe()

df.describe() use to get mathematical information about each columns' data.

Ex: count, mean, std, min,25%,50%,75% of each column.

From now on, we have to prepare dataset for clustering. Before enter dataset as input to the clustering model, we have to clean the dataset. It means that we are fixing if there is any null values or errors.

Following code describes, if there are missing values or not in dataset,

```
[51] df.isnull().sum(  )

CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Figure 8: remove null values

if we want to remove unnecessary columns from dataset, following code can used. Before use data set for clustering, we have to remove Customer Id column.

```
[52] df.drop(["CustomerID"],axis=1,inplace=True)

[53] df.head()
```

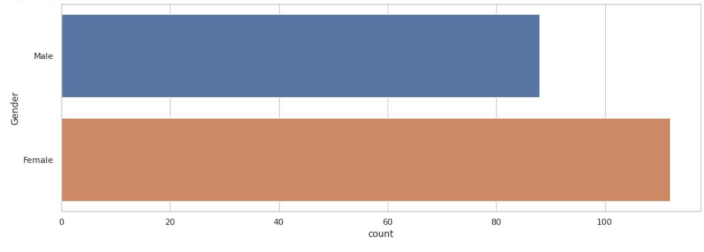
	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

Figure 9: remove unnecessary columns

COMPARISION IN GENDER

```
plt.figure(figsize=(15,5))
sns.countplot(y='Gender',data=df)
plt.show
```

```
<function matplotlib.pyplot.show>
```

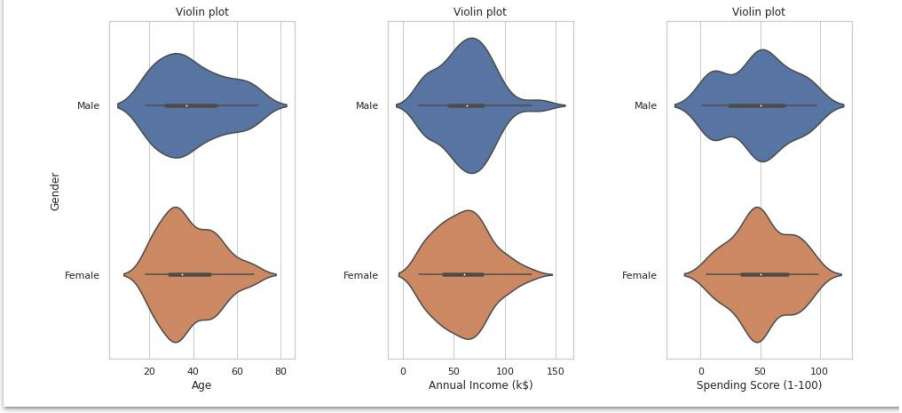


COMPARING GENDER NO.OF FEMALE ARE MORE COMPARED TO MALE

Figure 10: Comparison of number male and female in the dataset

VIOLIN METHOD

```
plt.figure(1, figsize=(15,7))
n=0
for cols in['Age','Annual Income (k$)' ,'Spending Score (1-100)']:
  n+= 1
  plt.subplot(1, 3 , n)
  sns.set(style="whitegrid")
  plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
  sns.violinplot(x = cols , y='Gender', data=df)
  plt.ylabel('Gender' if n== 1 else '')
  plt.title('Violin plot')
plt.show()
```



VIOLIN DISTRIBUTION FOR AGE , ANNUAL INCOME , SPENDING SCORE BASED ON GENDER

Figure 11:-Violin method for comparison in different attributes

```
age_18_25 = df.Age[(df.Age >= 18) & (df.Age <=25)]
age_26_35 = df.Age[(df.Age >= 26) & (df.Age <=35)]
age_36_45 = df.Age[(df.Age >= 36) & (df.Age <=45)]
age_46_55 = df.Age[(df.Age >= 46) & (df.Age <=55)]
age_55above = df.Age[df.Age >= 56]

agex = ["18-25", "26-35", "36-45", "46-55" ,"55+"]
agey = [len(age_18_25.values),len(age_26_35.values),len(age_36_45.values), len(age_46_55.values), len(age_55above.values)]

plt.figure(figsize=(15,6))
sns.barplot(x=agex,y=agey,palette="mako")
plt.title("Number of Customer and Ages")
plt.xlabel("Age")
plt.ylabel("Number of Customer")
plt.show()
```

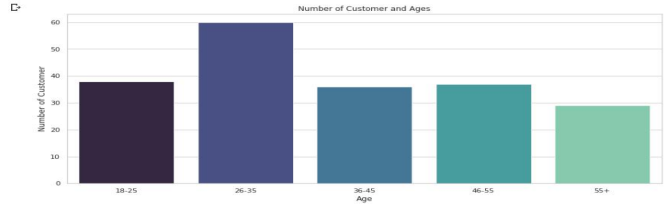


Figure 12:-Plot between age and customers

```
sns.relplot(x="Annual Income (k$)", y="Spending Score (1-100)", data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7fce097b6110>
```

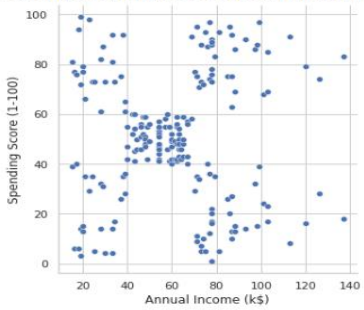


Figure 13:-PLOT BETWEEN ANNUAL INCOME AND SPENDING SCORE

Figure 14:-PLOT BETWEEN NO.OF CUSTOMERS AND SPENDING SCORE

```
ai_0_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>= 0) & (df["Annual Income (k$)"] <=30)]
ai_31_60 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>= 31) & (df["Annual Income (k$)"] <=60)]
ai_61_90 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>= 61) & (df["Annual Income (k$)"] <=90)]
ai_91_120 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>= 91) & (df["Annual Income (k$)"] <=120)]
ai_121_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>= 121) & (df["Annual Income (k$)"] <=150)]

aix = ["0-30", "31-60", "61-90", "91-120" ,"121-150"]
aiy = [len(ai_0_30.values),len(ai_31_60.values),len(ai_61_90.values), len(ai_91_120.values), len(ai_121_150.values)]

plt.figure(figsize=(15,6))
sns.barplot(x=aix,y=aiy,palette="Spectral")
plt.title("Annual Incomes")
plt.xlabel("Income")
plt.ylabel("Number of Customer")
plt.show()
```

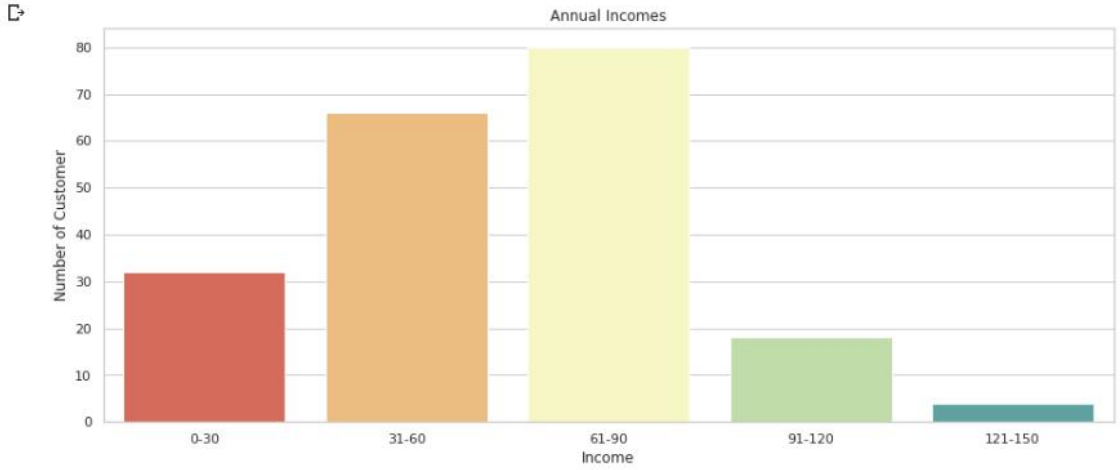


Figure 15:-PLOT BETWEEN ANNUAL INCOME AND NO.OF CUSTOMERS

Now our dataset is ready to use clustering algorithm model.

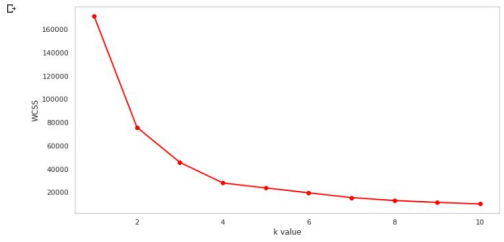
6.7 K-Means Clustering

K Means Clustering is the most common and simplest Machine learning algorithm and it follows an iterative approach which attempts to partition the dataset into different “k” number of predefined and non-overlapping subgroups where each data point belongs to only one subgroup according to their similar qualities.

```
X1=df.loc[:, ["Age", "Spending Score (1-100)"]].values

from sklearn.cluster import KMeans
SSE = []
for k in range(1,11):
    kmeans = KMeans( n_clusters = k, init='k-means++')
    kmeans.fit(X1)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),SSE,linewidth=2, color="red", marker = "8")
plt.xlabel('k value')
plt.ylabel('WCSS')
plt.show()
```



OPTIMAL NO.OF CLUSTERS = 4

Figure 16:kmeans number of clusters

```
[61] kmeans = KMeans(n_clusters=4)
     label = kmeans.fit_predict(X1)

     print(label)
```

```
[2 1 0 1 2 1 0 1 0 1 0 1 0 1 0 1 2 2 0 1 2 1 0 1 0 1 0 2 0 1 0 1 0 1 0 1 0
 1 0 1 3 1 3 2 0 2 3 2 2 2 3 2 2 3 3 3 3 3 2 3 3 2 3 3 3 2 3 3 2 2 3 3 3 3
 3 2 3 2 2 3 3 2 3 3 2 3 3 2 2 3 3 2 3 2 2 2 3 2 3 2 2 3 3 2 3 2 3 3 3 3 3
 2 2 2 2 2 3 3 3 3 2 2 2 1 2 1 3 1 0 1 0 1 2 1 0 1 0 1 0 1 0 1 2 1 0 1 3 1
 0 1 0 1 0 1 0 1 0 1 0 1 3 1 0 1 0 1 0 1 0 2 0 1 0 1 0 1 0 1 0 1 0 1 0 1 2
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
```

```
print(kmeans.cluster_centers_)
```

```
[[43.29166667 15.02083333]
 [30.1754386  82.35087719]
 [27.61702128 49.14893617]
 [55.70833333 48.22916667]]
```

Figure 17:-SCATTER PLOT BETWEEN SPENDING SCORE AND AGE

```
plt.scatter(X1[:,0], X1[:,1], c=kmeans.labels_, cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color = 'black')
plt.title('Cluster of Customers')
plt.xlabel('Age')
plt.ylabel('Spending Score(1-100)')
plt.show()
```

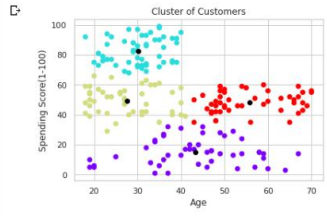
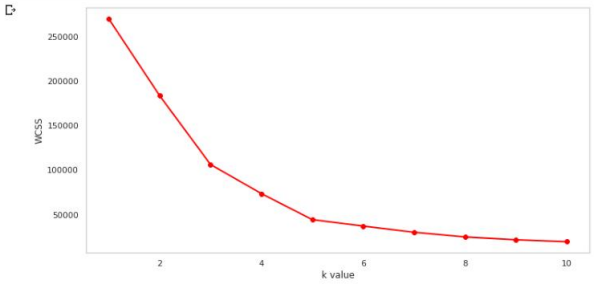


Figure 17:-SCATTER PLOT BETWEEN SPENDING SCORE AND AGE

```
X2=df.loc[:, ["Annual Income (k$)", "Spending Score (1-100)"]].values

from sklearn.cluster import KMeans
SSE = []
for k in range(1,11):
    kmeans = KMeans( n_clusters = k, init='k-means++')
    kmeans.fit(X2)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),SSE,linewidth=2, color="red", marker = "8")
plt.xlabel('k value')
plt.ylabel('WCSS')
plt.show()
```



OPTIMAL NO.OF CLUSTERS = 5

Figure 18:-Optimal no.of clusters for X2

```
[65] kmeans = KMeans(n_clusters=5)
     label = kmeans.fit_predict(X1)

     print(label)
```

```
[1 3 0 3 2 3 0 3 0 3 0 3 0 3 0 3 2 1 2 3 2 3 0 3 0 3 2 1 2 3 0 3 0 3 0 3 0
 3 2 3 4 3 2 1 2 1 4 1 1 2 4 1 1 4 4 2 4 4 1 4 4 1 4 4 4 1 4 4 1 1 4 2 4 4
 4 1 4 2 1 4 4 1 4 2 1 4 4 1 1 4 4 1 4 2 2 1 4 1 2 1 1 4 4 1 4 1 4 4 4 4 4
 1 2 1 1 1 4 4 4 4 1 2 1 3 2 3 2 3 0 3 0 3 2 3 0 3 0 3 0 3 0 3 2 3 0 3 2 3
 2 3 0 3 0 3 0 3 0 3 0 3 4 3 0 3 2 3 0 3 2 1 0 3 0 3 0 3 0 3 0 3 2 3 0 3 2
 3 0 3 0 3 2 3 0 3 0 3 2 3 0 3]
```

Figure 19:-Labelling of clusters

```
[66] print(kmeans.cluster_centers_)
```

```
[[43.78378378 11.27027027]
 [25.32352941 53.05882353]
 [39.41935484 35.03225806]
 [30.1754386  82.35087719]
 [57.24390244 49.73170732]]
```

Figure 20:-Centroid of clusters

```
plt.scatter(X2[:,0], X1[:,1], c=kmeans.labels_, cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color = 'black')
plt.title('Cluster of Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score(1-100)')
plt.show()
```

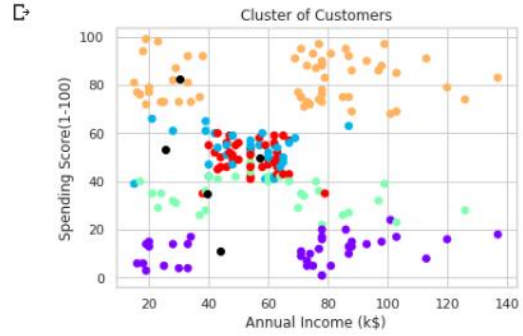


Figure 21:-SCATTER PLOT BETWEEN ANNUAL INCOME AND SPENDING SCORE

```python
X3=df.iloc[:,1:]

SSE = []
for k in range(1,11):
    kmeans = KMeans( n_clusters = k, init='k-means++')
    kmeans.fit(X3)
    SSE.append(kmeans.inertia_)

# converting the results into a dataframe and plotting them
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),SSE,linewidth=2, color="red", marker = "8")
plt.xlabel('k value')
plt.ylabel('WCSS')
plt.show()
```

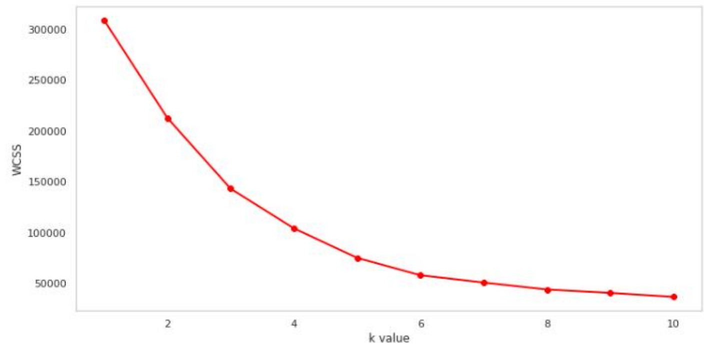


Figure 22:-optimal no.of clusters for X3

Elbow method for X3

Optimal number of clusters = 5

```python
kmeans = KMeans(n_clusters=5)
label = kmeans.fit_predict(X3)

print(label)
```

```
[0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0
 3 0 3 0 3 0 3 0 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 1 4 1 2 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 2 1 4 1 4 1
 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4
 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1]
```

Figure 23:-Labelling of clusters

```python
[70] print(kmeans.cluster_centers_)
```

```
[[45.2173913  26.30434783 20.91304348]
 [32.69230769 86.53846154 82.12820513]
 [43.08860759 55.29113924 49.56962025]
 [25.52173913 26.30434783 78.56521739]
 [40.66666667 87.75       17.58333333]]
```

Figure 24:-Centroid of clusters

```
df["label"] = clusters
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Score (1-100)"][df.label == 0], c='blue', s=60)
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Score (1-100)"][df.label == 1], c='red', s=60)
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Score (1-100)"][df.label == 2], c='green', s=60)
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Score (1-100)"][df.label == 3], c='orange', s=60)
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Score (1-100)"][df.label == 4], c='purple', s=60)
ax.view_init(30,185)
plt.xlabel("Age")
plt.ylabel("Annual Income")
ax.set_zlabel('Spending Score(1-100)')
plt.show
```

```
<function matplotlib.pyplot.show>
```

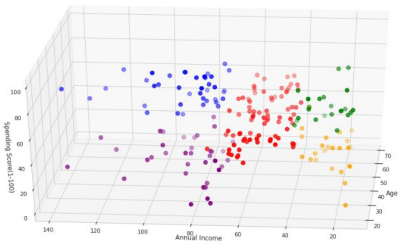


Figure 25:-Scatter Plot among Age, Annual Income , Spending Score

```
[72] # First, build a model with 5 clusters

     kmeans = KMeans(n_clusters = 5 , init='k-means++')
     kmeans.fit(X2)

     # Now, print the silhouette score of this model

     print(silhouette_score(X2, kmeans.labels_, metric='euclidean'))
```

```
0.553931997444648
```

Figure 26:-Silhouette score of X2

```
# First, build a model with 4 clusters

kmeans = KMeans(n_clusters = 4 , init='k-means++')
kmeans.fit(X1)

# Now, print the silhouette score of this model

print(silhouette_score(X1, kmeans.labels_, metric='euclidean'))
```

```
0.49973941540141753
```

Figure 27:-Silhouette score of X1

```
[74] # First, build a model with 5 clusters

     kmeans = KMeans(n_clusters = 5 , init='k-means++')
     kmeans.fit(X3)

     # Now, print the silhouette score of this model

     print(silhouette_score(X3, kmeans.labels_, metric='euclidean'))
```

```
0.44045315045641703
```

Figure 28:-Silhouette score of X3

6.8 DBSCAN Clustering Algorithm

The DBSCAN stands for density based spatial clustering of applications with noise

There are two parameters that play a vital role in the algorithm. 1) Min points and 2) Epsilon.

The algorithm works by processing each and every data point individually in particular for each point. It will construct a kind of a circle with the point being in the center and having the radius equal to the Epsilon.

Min Points:

MinPoints are the number of points that must exist within ε distance from the point.

Epsilon(ε) :

It is the distance or radius around each object.

The DBSCAN will process each and every object/points in this fashion and at the end it will obtain categorization of all the points as either core, border or noise points. Once the categorization of the points is obtained, the next step is to use them to construct the clusters. DBSCAN take up a core point and then look at the points which are inside its Epsilon radius circle and assign a Cluster label to those points, So the key idea is to give the same label to all the points inside the circle of a core point.
Multiple iterations will be run for different core points to assign Cluster label, please note algorithm will not assign new Cluster label to those points which have already be considered in earlier iteration.

```
# DBSCAN Clustering
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
# Importing the dataset
dataset = pd.read_csv("Mall_Customers.csv")
data = dataset.iloc[:, [3, 4]].values
dataset.head()
```

	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

Figure 29:-Importing Essential Libraries

```
[76] X = df.iloc[:,[3,4]].values

# visualizing the dataset
plt.scatter(data[:, 0], data[:, 1], s = 10, c = 'black')
```

```
<matplotlib.collections.PathCollection at 0x7fce0ff5b350>
```

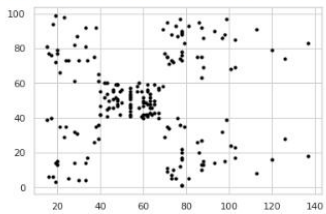


Figure 30:-Reading and visualizing Data

```
[78] # Fitting DBSCAN to the dataset and predict the Cluster label
     from sklearn.cluster import DBSCAN
     dbscan = DBSCAN(eps=5.5, min_samples=4)
     labels = dbscan.fit_predict(data)
     np.unique(labels)

     array([-1,  0,  1,  2,  3,  4,  5])
```

```
# Visualising the clusters
plt.scatter(data[labels == -1, 0], data[labels == -1, 1], s = 10, c = 'black')
plt.scatter(data[labels == 0, 0], data[labels == 0, 1], s = 10, c = 'blue')
plt.scatter(data[labels == 1, 0], data[labels == 1, 1], s = 10, c = 'red')
plt.scatter(data[labels == 2, 0], data[labels == 2, 1], s = 10, c = 'green')
plt.scatter(data[labels == 3, 0], data[labels == 3, 1], s = 10, c = 'brown')
plt.scatter(data[labels == 4, 0], data[labels == 4, 1], s = 10, c = 'pink')
plt.scatter(data[labels == 5, 0], data[labels == 5, 1], s = 10, c = 'yellow')
plt.scatter(data[labels == 6, 0], data[labels == 6, 1], s = 10, c = 'silver')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.show()
```

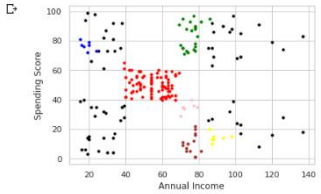


Figure 31:-SCATTER PLOT BETWEEN ANNUAL INCOME AND SPENDING SCORE

```
# Visualising the clusters
plt.scatter(data[labels == -1, 0], data[labels == -1, 1], s = 10, c = 'black')
plt.scatter(data[labels == 0, 0], data[labels == 0, 1], s = 10, c = 'blue')
plt.scatter(data[labels == 1, 0], data[labels == 1, 1], s = 10, c = 'red')
plt.scatter(data[labels == 2, 0], data[labels == 2, 1], s = 10, c = 'green')
plt.scatter(data[labels == 3, 0], data[labels == 3, 1], s = 10, c = 'brown')
plt.scatter(data[labels == 4, 0], data[labels == 4, 1], s = 10, c = 'pink')
plt.scatter(data[labels == 5, 0], data[labels == 5, 1], s = 10, c = 'yellow')
plt.scatter(data[labels == 6, 0], data[labels == 6, 1], s = 10, c = 'silver')
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.show()
```

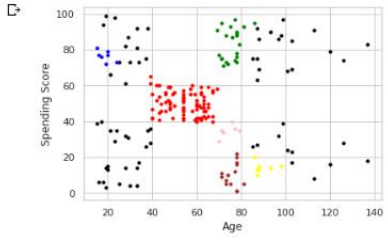


Figure 32:-SCATTER PLOT BETWEEN AGE AND SPENDING SCORE

```
# Visualising the clusters
plt.scatter(data[labels == -1, 0], data[labels == -1, 1], s = 10, c = 'black')
plt.scatter(data[labels == 0, 0], data[labels == 0, 1], s = 10, c = 'blue')
plt.scatter(data[labels == 1, 0], data[labels == 1, 1], s = 10, c = 'red')
plt.scatter(data[labels == 2, 0], data[labels == 2, 1], s = 10, c = 'green')
plt.scatter(data[labels == 3, 0], data[labels == 3, 1], s = 10, c = 'brown')
plt.scatter(data[labels == 4, 0], data[labels == 4, 1], s = 10, c = 'pink')
plt.scatter(data[labels == 5, 0], data[labels == 5, 1], s = 10, c = 'yellow')
plt.scatter(data[labels == 6, 0], data[labels == 6, 1], s = 10, c = 'silver')
plt.xlabel('Annual Income')
plt.ylabel('Age')
plt.show()
```

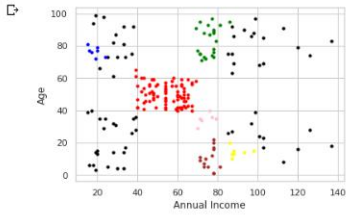


Figure 33:-SCATTER PLOT BETWEEN AGE AND ANNUAL INCOME

```
[ ] # Defining the list of hyperparameters to try
    eps_list=np.arange(start=0.1, stop=0.9, step=0.01)
    min_sample_list=np.arange(start=1, stop=6, step=2)
```

```
# Creating empty data frame to store the silhouette scores for each trials
silhouette_scores_data=pd.DataFrame()

for eps_trial in eps_list:
    for min_sample_trial in min_sample_list:

        # Generating DBSAN clusters
        db = DBSCAN(eps=eps_trial, min_samples=min_sample_trial)

        if("len(np.unique(db.fit_predict(X3)))&gt;1"):
            sil_score=silhouette_score(X3, db.fit_predict(X))
        else:
            continue
        trial_parameters="eps:" + str(eps_trial.round(1)) +" min_sample :" + str(min_sample_trial)

        silhouette_scores_data=silhouette_scores_data.append(pd.DataFrame(data=[[sil_score,trial_parameters]], columns=["score", "parameters"]))

# Finding out the best hyperparameters with highest Score
silhouette_scores_data.sort_values(by='score', ascending=False).head(1)
```

	score	parameters
0	-0.278226	eps:0.9 min_sample :5

Figure 34:-Silhouette score in DBSCAN Clustering

6.9 Comparison of results

KMEANS SILHOUETTE SCORE

X1 = 0.499

X2 = 0.553

X3 = 0.440

DBSCAN CLUSTERING SILHOUETTE SCORE

SCORE = 0.27

Since K-Means have the highest Silhouette score compared to DBSCAN so K-Means Clustering is more appropriate then DBSCAN Clustering for "CUSTOMER SEGMENTATION"

CHAPTER-VII
CONCLUSION AND FUTURE SCOPE

As clustering is unsupervised learning, need to analyse each cluster and have a definition with respect to business data because Clustering is always guided by some business rules. Once clusters are close to business rules, model will make sense.

For identifying, prioritizing, and targeting your best current customer segments, simply following it does not guarantee success. To be effective, you must prepare and plan for the various challenges and hurdles that each step may present, and always make sure to adapt your process to any new information or feedback that might change its output.

Additionally, you cannot force feed this process on your business. If the key stakeholders that will be impacted by the best current customers segmentation process do not fully buy-in, then the outputs produced from it will be relatively meaningless.

If you properly manage the best current customer segmentation process, however, the impact it can have on every part of your organization — sales, marketing, product development, customer service, etc. — is immense. Your business will possess stronger customer focus and market clarity, allowing it to scale in a far more predictable and efficient manner.

Ultimately, that means no longer needing to take on every customer that is willing to pay for your product or service, which will allow you to instead hone in on a specific subset of customers that present the most profitable opportunities and efficient use of resources. That is critical for every business, of course, but at the expansion stage, it can often be the difference between incredible success and certain failure

CHAPTER-VIII

REFERENCES

A. Dr. R. Gardener “The Essential R Reference” (2014),
B. Concepts of customer segmentation http://www.business-science.io
C. https://labs.openviewpartners.com/customer-segmentation/
D. Source data related to our analysis has been collected from https://github.com/mdancho84/orderSimulatoR/tree/master/data
E. https://www.kaggle.com/
F. https://www.r-project.org/
G. D. P. Yash Kushwaha, “Customer Segmentation using K-Means Algorithm,” 8th Semester Student of B.tech in Computer Science and Engineering
H. E. A. Onur DOĞAN1, “CUSTOMER SEGMENTATION BY USING RFM”.
I. C. M. S. R. a. K. V. N. T. Sajana, “A Survey on,” in Indian Journal of Science and Technology, Volume 9, Issue 3, Jan 2016.
J. B. P. E. Shreya Tripathi, “Approaches to,” in International Journal of Engineering and Technology, Volume 7, 2018.
K. R. Azarnoush Ansari, ““Customer Clustering Using a,” in International Journal of Business and Management, Volume 11, Issue 7, 2016.