Introduction:

Customer segmentation simply means grouping your customers according to various characteristics .

It’s a way for organizations to understand their customers. Knowing the differences between customer groups, it’s easier to make strategic decisions regarding product growth and marketing.

There are different methodologies for customer segmentation, and they depend on four types of parameters:

- geographic
- demographic
- behavioral
- psychological

Geographic customer segmentation is very simple, it’s all about the user’s location. This can be implemented in various ways. You can group by country, state, city, or zip code.

Demographic segmentation is related to the structure, size, and movements of customers over space and time. Many companies use gender differences to create and market products. Parental status is another important feature. You can obtain data like this from customer surveys.

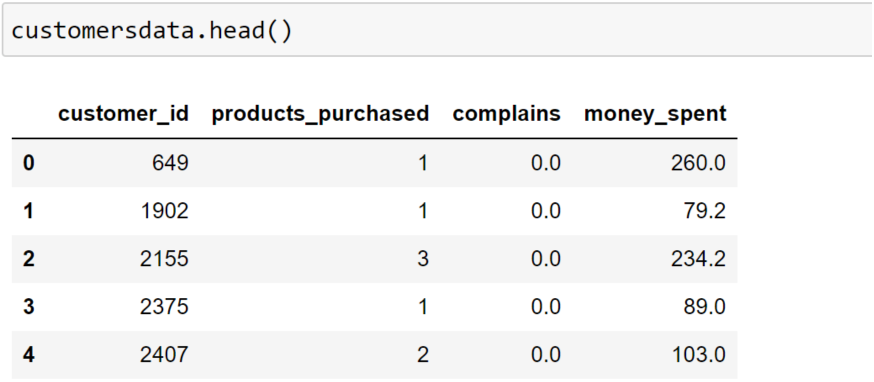
Behavioral customer segmentation is based on past observed behaviors of customers that can be used to predict future actions. For example, brands that customers purchase, or moments when they buy the most. The behavioral aspect of customer segmentation not only tries to understand reasons for purchase but also how those reasons change throughout the year.

Psychological segmentation of customers generally deals with things like personality traits, attitudes, or beliefs. This data is obtained using customer surveys, and it can be used to gauge customer sentiment.

Exploring customer dataset and its features

analyzing a customer dataset. Our dataset has 24,000 data points and four features. The features are:

- Customer ID – This is the id of a customer for a particular business.
- Products Purchased – This feature represents the number of products purchased by a customer in a year.
- Complaints – This column value indicates the number of complaints made by the customer in the last year
- Money Spent – This column value indicates the amount of money paid by the customer over the last year.

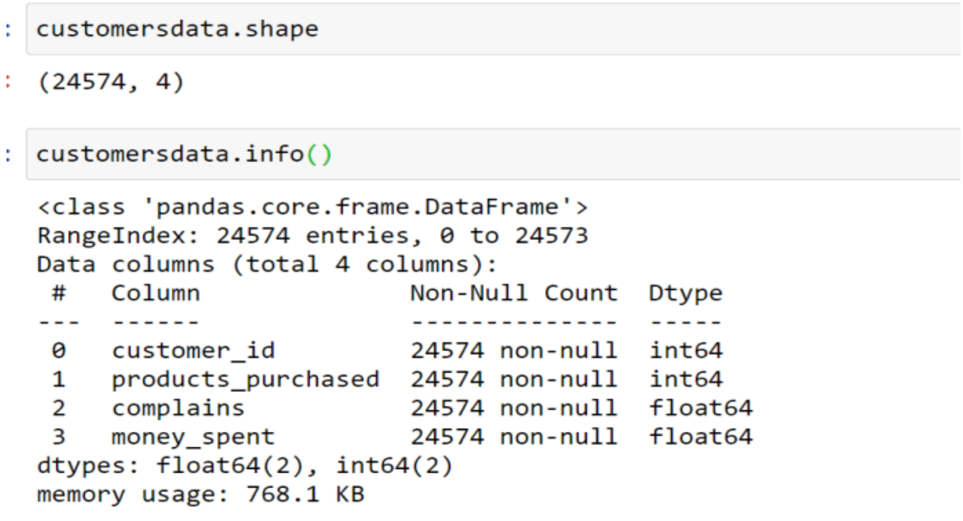


```
customersdata.head()
```

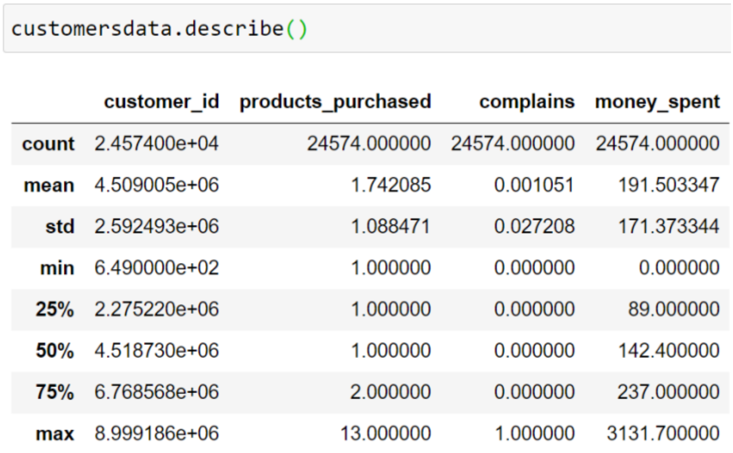
	customer_id	products_purchased	complains	money_spent
0	649	1	0.0	260.0
1	1902	1	0.0	79.2
2	2155	3	0.0	234.2
3	2375	1	0.0	89.0
4	2407	2	0.0	103.0

Pre-processing the dataset

Before feeding the data to the k-means clustering algorithm, we need to pre-process the dataset. By implementing the necessary pre-processing for the customer dataset.



```
: customersdata.shape

: (24574, 4)

: customersdata.info()

  <class 'pandas.core.frame.DataFrame'>
  RangeIndex: 24574 entries, 0 to 24573
  Data columns (total 4 columns):
   #   Column              Non-Null Count  Dtype  
  ---  ------              --------------  -----  
   0   customer_id         24574 non-null  int64  
   1   products_purchased  24574 non-null  int64  
   2   complains           24574 non-null  float64
   3   money_spent         24574 non-null  float64
  dtypes: float64(2), int64(2)
  memory usage: 768.1 KB
```



```
customersdata.describe()
```

	customer_id	products_purchased	complains	money_spent
count	2.457400e+04	24574.000000	24574.000000	24574.000000
mean	4.509005e+06	1.742085	0.001051	191.503347
std	2.592493e+06	1.088471	0.027208	171.373344
min	6.490000e+02	1.000000	0.000000	0.000000
25%	2.275220e+06	1.000000	0.000000	89.000000
50%	4.518730e+06	1.000000	0.000000	142.400000
75%	6.768568e+06	2.000000	0.000000	237.000000
max	8.999186e+06	13.000000	1.000000	3131.700000

Implementing K- means clustering in Python

K-Means clustering is an efficient machine learning algorithm to solve data clustering problems. It’s an unsupervised algorithm that’s quite suitable for solving customer segmentation problems.

Unsupervised Learning

Unsupervised machine learning is quite different from supervised machine learning. It’s a special kind of machine learning algorithm that discovers patterns in the dataset from unlabelled data.

Unsupervised machine learning algorithms can group data points based on similar attributes in the dataset. One of the main types of unsupervised models is clustering models.

The Challenge

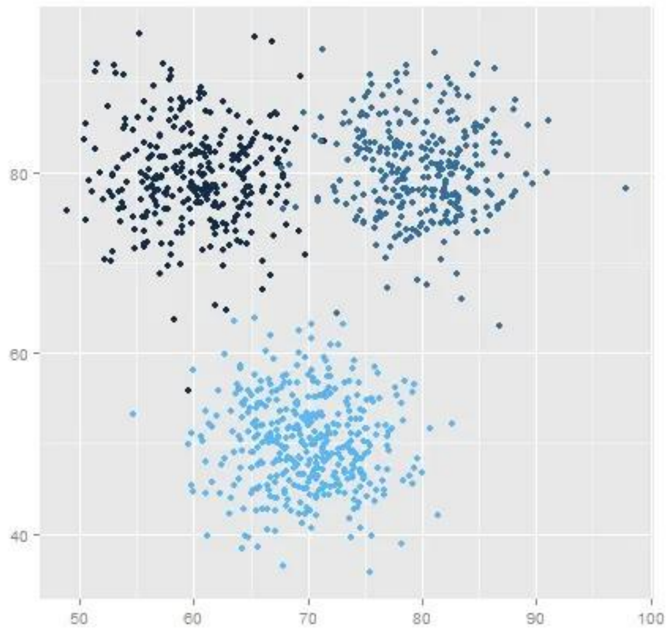
You are owing a supermarket mall and through membership cards, you have some basic data about your customers like Customer ID, age, gender, annual income and spending score. You want to understand the customers like who are the target customers so that the sense can be given to marketing team and plan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



Environment and tools

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

Loading the data

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

```
Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns


Df = pd.read_csv(“../input/customer-segmentation-tutorial-in-python/Mall_Customers.csv”)

Df.head()
```

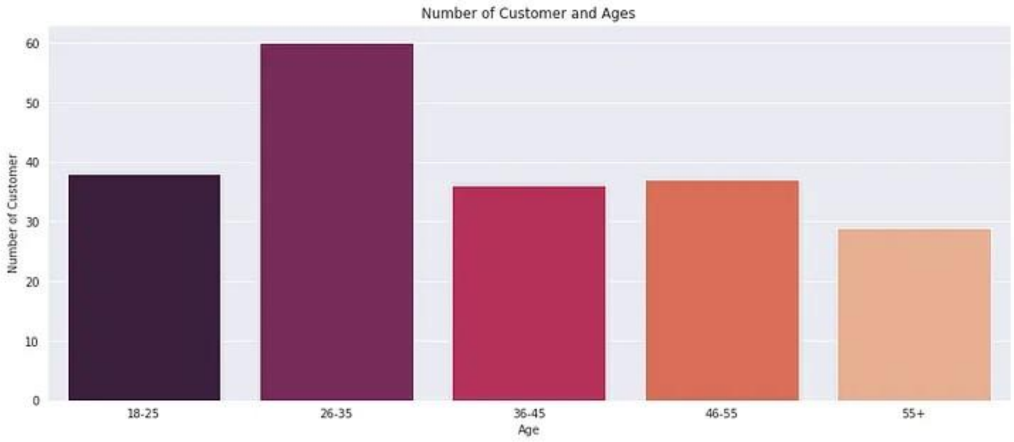
Dropping the id column and assuming the age frequency as customer

```
Df.drop([“CustomerID”], axis = 1, inplace=True)


Plt.figure(figsize=(10,6))

Plt.title(“Ages Frequency”)

Sns.axes_style(“dark”)

Sns.violinplot(y=df[“Age”])

Plt.show()
```

Distribution of customers:

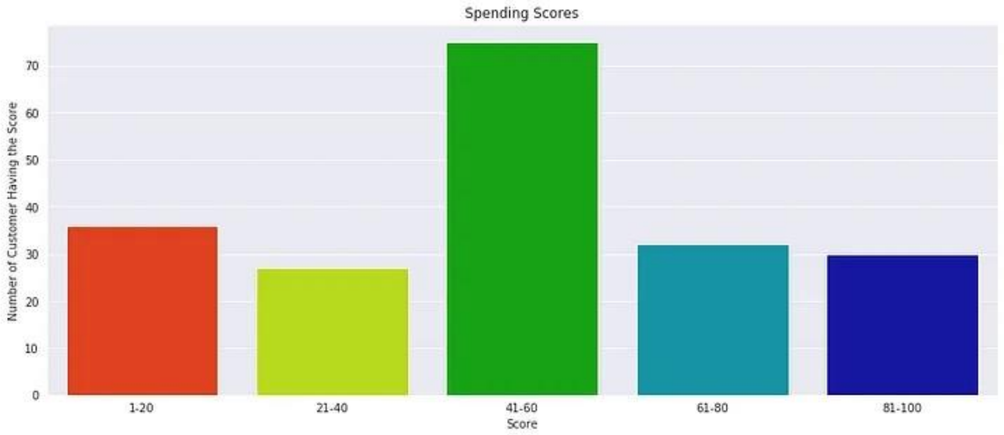
Checking the distribution of the number of customers .

```
Age18_25 = df.Age[(df.Age <= 25) & (df.Age >= 18)]

Age26_35 = df.Age[(df.Age <= 35) & (df.Age >= 26)]

Age36_45 = df.Age[(df.Age <= 45) & (df.Age >= 36)]

Age46_55 = df.Age[(df.Age <= 55) & (df.Age >= 46)]

Age55above = df.Age[df.Age >= 56]


X = [“18-25”,”26-35”,”36-45”,”46-55”,”55+”]

Y =
[len(age18_25.values),len(age26_35.values),len(age36_45.values),len(age46_55.values),len(age55above.
values)]


Plt.figure(figsize=(15,6))

Sns.barplot(x=x, y=y, palette=”rocket”)

Plt.title(“Number of Customer and Ages”)

Plt.xlabel(“Age”)

Plt.ylabel(“Number of Customer”)

Plt.show()
```



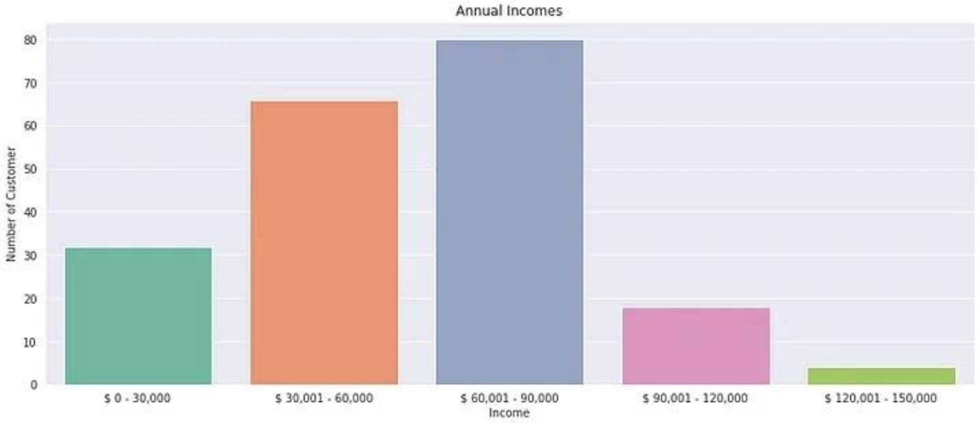
bar plot to visualize the number of customers according to their spending scores. The majority of the customers have spending score in the range 41–60.

```
Ss1_20 = df[“Spending Score (1-100)”][(df[“Spending Score (1-100)”] >= 1) & (df[“Spending Score (1-100)”] <= 20)]

Ss21_40 = df[“Spending Score (1-100)”][(df[“Spending Score (1-100)”] >= 21) & (df[“Spending Score (1-100)”] <= 40)]

Ss41_60 = df[“Spending Score (1-100)”][(df[“Spending Score (1-100)”] >= 41) & (df[“Spending Score (1-100)”] <= 60)]

Ss61_80 = df[“Spending Score (1-100)”][(df[“Spending Score (1-100)”] >= 61) & (df[“Spending Score (1-100)”] <= 80)]

Ss81_100 = df[“Spending Score (1-100)”][(df[“Spending Score (1-100)”] >= 81) & (df[“Spending Score (1-100)”] <= 100)]


Ssx = [“1-20”, “21-40”, “41-60”, “61-80”, “81-100”]

Ssy = [len(ss1_20.values), len(ss21_40.values), len(ss41_60.values), len(ss61_80.values), len(ss81_100.values)]

Plt.figure(figsize=(15,6))

Sns.barplot(x=ssx, y=ssy, palette=”nipy_spectral_r”)

Plt.title(“Spending Scores”)

Plt.xlabel(“Score”)

Plt.ylabel(“Number of Customer Having the Score”)

Plt.show()
```



bar plot to visualize the number of customers according to their annual income. The majority of the customers have annual income in the range 60000 and 90000.

```
Ai0_30 = df[“Annual Income (k$)”][(df[“Annual Income (k$)”] >= 0) & (df[“Annual Income (k$)”] <= 30)]

Ai31_60 = df[“Annual Income (k$)”][(df[“Annual Income (k$)”] >= 31) & (df[“Annual Income (k$)”] <= 60)]

Ai61_90 = df[“Annual Income (k$)”][(df[“Annual Income (k$)”] >= 61) & (df[“Annual Income (k$)”] <= 90)]

Ai91_120 = df[“Annual Income (k$)”][(df[“Annual Income (k$)”] >= 91) & (df[“Annual Income (k$)”] <= 120)]

Ai121_150 = df[“Annual Income (k$)”][(df[“Annual Income (k$)”] >= 121) & (df[“Annual Income (k$)”] <= 150)]



Aix = [“$ 0 – 30,000”, “$ 30,001 – 60,000”, “$ 60,001 – 90,000”, “$ 90,001 – 120,000”, “$ 120,001 – 150,000”]

Aiy = [len(ai0_30.values), len(ai31_60.values), len(ai61_90.values), len(ai91_120.values), len(ai121_150.values)]



Plt.figure(figsize=(15,6))

Sns.barplot(x=aix, y=aiy, palette=”Set2”)

Plt.title(“Annual Incomes”)

Plt.xlabel(“Income”)
Plt.ylabel(“Number of Customer”)

Plt.show()
```



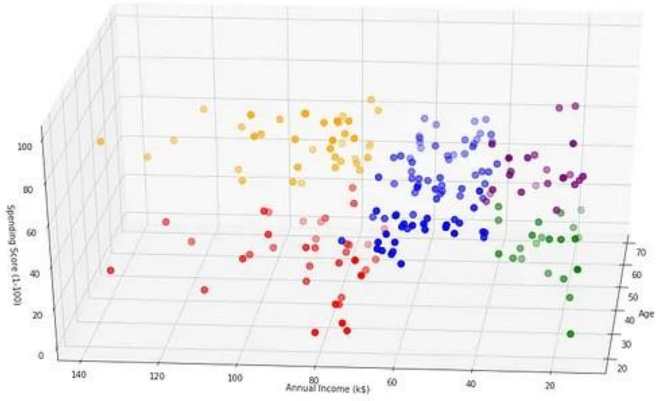
Using k means for clustering and visualization of the 3 plot by importing the k means .

```
From sklearn.cluster import KMeans

Wcss = []

For k in range(1,11):

   Kmeans = KMeans(n_clusters=k, init=”k-means++”)

   Kmeans.fit(df.iloc[:,1:])

   Wcss.append(kmeans.inertia_)

Plt.figure(figsize=(12,6))

Plt.grid()

Plt.plot(range(1,11),wcss, linewidth=2, color=”red”, marker =”8”)

Plt.xlabel(“K Value”)

Plt.xticks(np.arange(1,11,1))

Plt.ylabel(“WCSS”)

Plt.show()
```

Now finding and visualization of the customer segmentation

```
Km = KMeans(n_clusters=5)
Clusters = km.fit_predict(df.iloc[:,1:])
Df[“label”] = clusters

From mpl_toolkits.mplot3d import Axes3D
Import matplotlib.pyplot as plt
Import numpy as np
Import pandas as pd

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$)”][df.label == 4], df[“Spending Score (1-100)”][df.label == 4], c=’purple’, s=60)
Ax.view_init(30, 185)
Plt.xlabel(“Age”)
Plt.ylabel(“Annual Income (k$)”)
Ax.set_zlabel(‘Spending Score (1-100)’)
Plt.show()
```

Result:



Conclusion :

Therefore loading and pre-processing of the customer data is done using the k-means clustering algorithm and the visualization of the customer data is presented using the customer data of distribution and annual spending of the customer .