CUSTOME SEGMENTATION WITH DATA SCIENCE

TEAM 1

PHASE 3- LOADING AND PREPROCESSING DATASET

PROBLEM DEFINITION:

In today's competitive market, it is essential for businesses to understand and target customers with similar characteristics and behaviors. Aims to improve business strategy, improve customer service and increase profitability. Customer segmentation is a data-based method that divides a company's customers into different groups based on their characteristics or behavior, allowing for targeted advertising, business and products or services.

The goal of this data exploration is to create powerful customer solutions that enable companies to make informed decisions and align business processes, copy, sales and customer experience around customer experience.

DATA COLLECTION AND GATHERING:

Collecting relevant data from various sources such as kaggle. Ensuring data quality and accuracy by addressing missing values, outliers and inconsistencies.

I. DATA SOURCE:

Dataset link:
(https://www.kaggle.com/datasets/vedavyasv/usa)

1	CustomerI	Genre	Age	Annual Inc	Spending Score (1-10
2	1	Male	19	15	39
3	2	Male	21	15	81
4	3	Female	20	16	6
5	4	Female	23	16	77
6	5	Female	31	17	40
7	6	Female	22	17	76
8	7	Female	35	18	6
9	8	Female	23	18	94
10	9	Male	64	19	3
11	10	Female	30	19	72
12	11	Male	67	19	14
13	12	Female	35	19	99
14	13	Female	58	20	15
15	14	Female	24	20	77
16	15	Male	37	20	13
17	16	Male	22	20	79
18	17	Female	35	21	35
19	18	Male	20	21	66
20	19	Male	52	23	29
21	20	Female	35	23	98
22	21	Male	35	24	35
23	22	Male	25	24	73
24	23	Female	46	25	5
25	24	Male	31	25	73
26	25	Female	54	28	14
27	26	Male	29	28	82

DATA PREPROCESSING:

Data preprocessing is a critical step in customer segmentation using data science. Properly preparing your data ensures that the results of your customer segmentation analysis are accurate and meaningful.

CODE:

```
import numpy as np

 import pandas as pd

 import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

 from sklearn.decomposition import PCA

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette_score


# Importing the dataset
dataset = pd.read_csv('../input/mall-
customers/Mall_Customers.csv',index_col='CustomerID')


dataset.head()
```

O/P:

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

I/P:

dataset.describe()

O/P:

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

FEATURE ENGINEERING:

Creating meaningful features that can help in customer segmentation such as customer lifetime value, purchase frequency, etc.

EXPLORATORY DATA ANALYSIS (EDA):

Conducting exploratory data analysis to gain insights into the data. Visualizing and summarizing key statistics and trends.

MODEL SELECTION:

Choosing an appropriate segmentation technique or algorithm.

Using **K-MEANS CLUSTERING** algorithm for this project.

MODEL TRAINING:

Training the selected segmentation model on the preprocessed data.

CODE:

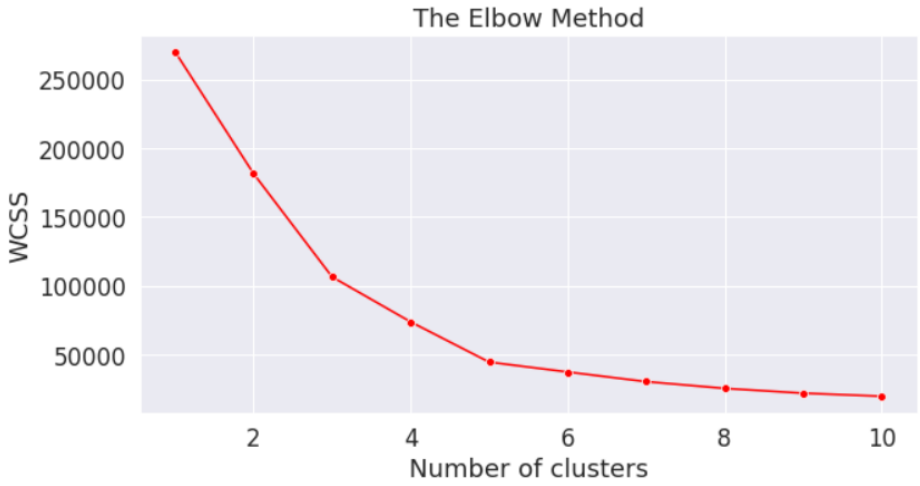
I/P:

```
# Apply PCA for dimensionality reduction

n_components = 2

pca = PCA(n_components=n_components)

reduced_data = pca.fit_transform(scaled_data)
```

```
# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    # inertia method returns wcss for that model
    wcss.append(kmeans.inertia_)
```

```
plt.figure(figsize=(10,5))
sns.lineplot(range(1, 11), wcss,marker='o',color='red')
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

O/P:



I/P:

```
# Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X)


# Visualising the clusters
plt.figure(figsize=(15,7))
sns.scatterplot(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], color = 'yellow', label =
'Cluster 1',s=50)
```

```
sns.scatterplot(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], color = 'blue', label = 'Cluster 2',s=50)
sns.scatterplot(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], color = 'green', label = 'Cluster 3',s=50)
sns.scatterplot(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], color = 'grey', label = 'Cluster 4',s=50)
sns.scatterplot(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], color = 'orange', label = 'Cluster 5',s=50)
sns.scatterplot(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], color = 'red',
            label = 'Centroids',s=300,marker=',')
plt.grid(False)
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

O/P: