Project Introduction:

- Provide a brief introduction to the project, explaining the objective of customer segmentation and how it can benefit businesses.

Data Loading and Preprocessing:

- Explain the code you provided for loading the dataset from the 'Mall_Customer.csv' file.

```
# Load the dataset

data = pd.read_csv('Mall_Customers.csv')
```

- Describe the structure of the dataset using the data.head() function.
- Mention the specific features you've selected for clustering, which are "Annual Income" and "Spending Score."
- Explain why you've standardised the features using StandardScaler.

```
# Standardize the features

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)
```

Determining Optimal Clusters:

- Describe the Elbow Method and its purpose in finding the optimal number of clusters.

```
# Plot the Elbow Method to find the optimal number of clusters

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()
```

- Present the plot showing the Within-Cluster Sum of Squares (WCSS) against the number of clusters.
- Explain how you determined that five clusters were appropriate for this analysis.

```
# Based on the Elbow Method, let's choose an appropriate number of clusters (e.g., 5)

num_clusters = 5
```

K-Means Clustering:

- Elaborate on the K-Means clustering algorithm.
- Present the code for applying K-Means clustering with five clusters.
- Show the scatter plot with clustered data points and centroids.

Cluster Analysis:

- Analyze each cluster individually to understand customer segments.
- Present statistics (e.g., mean, standard deviation) for "Annual Income" and "Spending Score" for each cluster.

Program:

```
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler



# Load the dataset

data = pd.read_csv('Mall_Customer.csv')



# Display the first few rows of the dataset to understand its structure
```

```
print(data.head())


# Select relevant features for clustering (Annual Income and Spending Score)
X = data[['Annual Income (k$)', 'Spending Score (1-100)']]


# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# Determine the optimal number of clusters using the Elbow Method
wcss = []  # Within-Cluster Sum of Squares
for i in range(1, 11):
   kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
   kmeans.fit(X_scaled)
   wcss.append(kmeans.inertia_)


# Plot the Elbow Method to find the optimal number of clusters
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
```

```
plt.ylabel('WCSS')

plt.show()



# Based on the Elbow Method, let's choose an appropriate number of clusters (e.g., 5)

num_clusters = 5



# Apply K-Means clustering with the selected number of clusters

kmeans = KMeans(n_clusters=num_clusters, init='k-means++', max_iter=300, n_init=10, random_state=0)

y_kmeans = kmeans.fit_predict(X_scaled)



# Add the cluster labels to the dataset

data['Cluster'] = y_kmeans



# Visualize the clusters

plt.scatter(X_scaled[y_kmeans == 0, 0], X_scaled[y_kmeans == 0, 1], s=100, c='red', label='Cluster 1')

plt.scatter(X_scaled[y_kmeans == 1, 0], X_scaled[y_kmeans == 1, 1], s=100, c='blue', label='Cluster 2')

plt.scatter(X_scaled[y_kmeans == 2, 0], X_scaled[y_kmeans == 2, 1], s=100, c='green', label='Cluster 3')

plt.scatter(X_scaled[y_kmeans == 3, 0], X_scaled[y_kmeans == 3, 1], s=100, c='cyan', label='Cluster 4')
```

```
plt.scatter(X_scaled[y_kmeans == 4, 0], X_scaled[y_kmeans == 4, 1], s=100, c='magenta', label='Cluster 5')

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='yellow', label='Centroids')

plt.title('Customer Segmentation')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.legend()

plt.show()



# Explore and analyze each cluster to understand customer segments

for cluster_num in range(num_clusters):

    cluster_data = data[data['Cluster'] == cluster_num]

    print(f'Cluster {cluster_num} Statistics:')

    print(cluster_data.describe())



# You can save or export the clustered dataset for further analysis or marketing strategies

data.to_csv('Mall_Customer.csv', index=False)
```

Output:

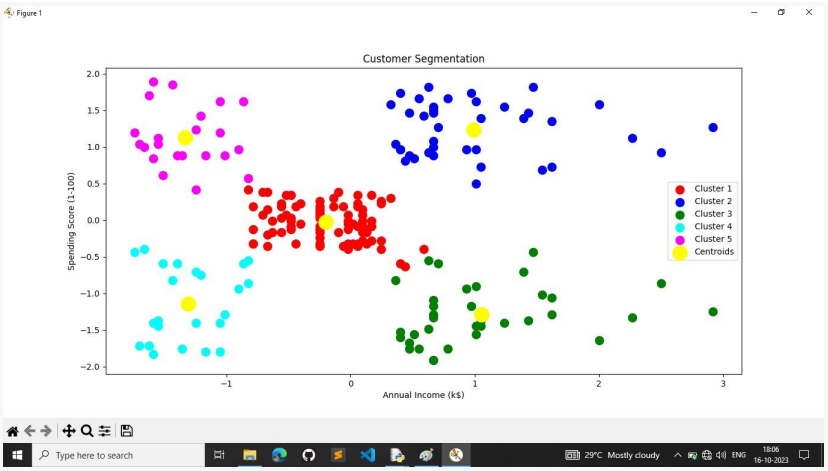


Figure 1
Customer Segmentation
Spending Score (1-100)
Annual Income (k$)
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Centroids
Type here to search
29°C Mostly cloudy
ENG
18:06
16-10-2023

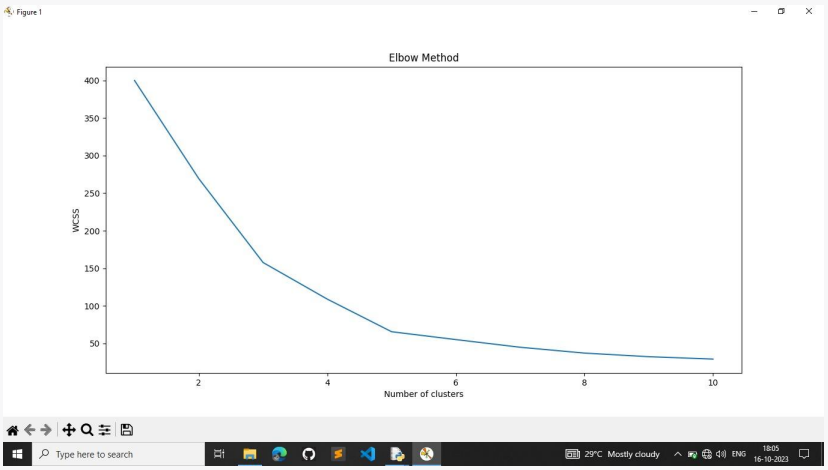
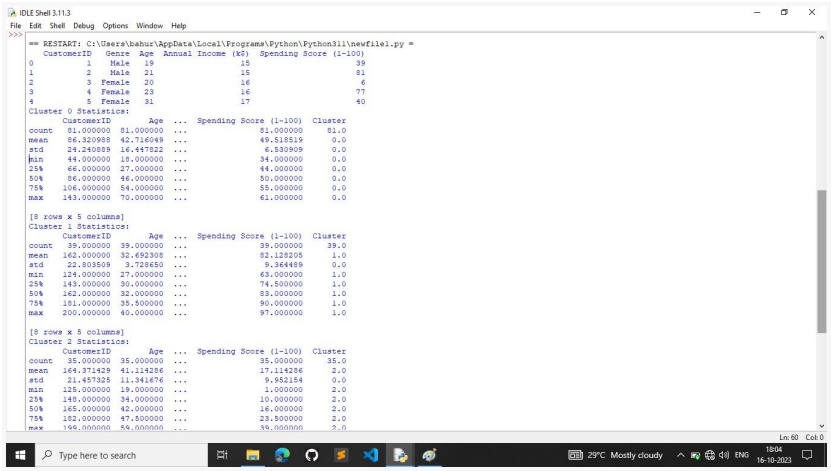


Figure 1
Elbow Method
WCSS
Number of clusters
Type here to search
29°C Mostly cloudy
ENG
18:05
16-10-2023



Conclusion:

- Summarize the findings from the cluster analysis.
- Explain what each cluster represents in terms of customer behavior or characteristics.
- Discuss potential business implications and strategies for each cluster.