

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_[0], 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:')

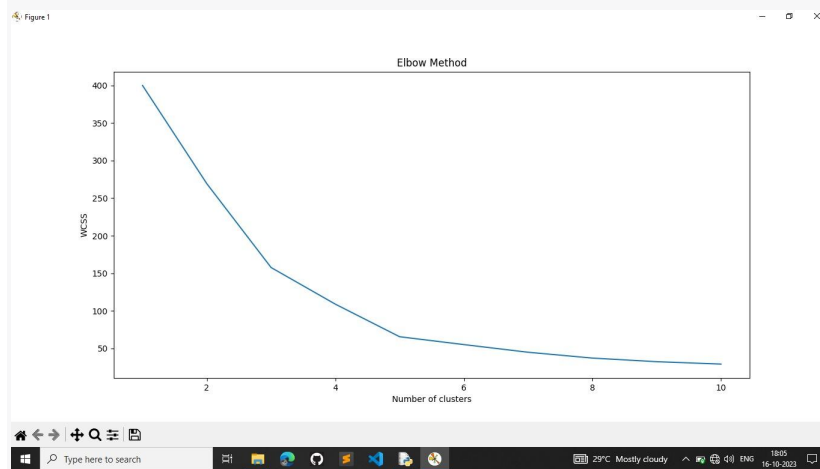
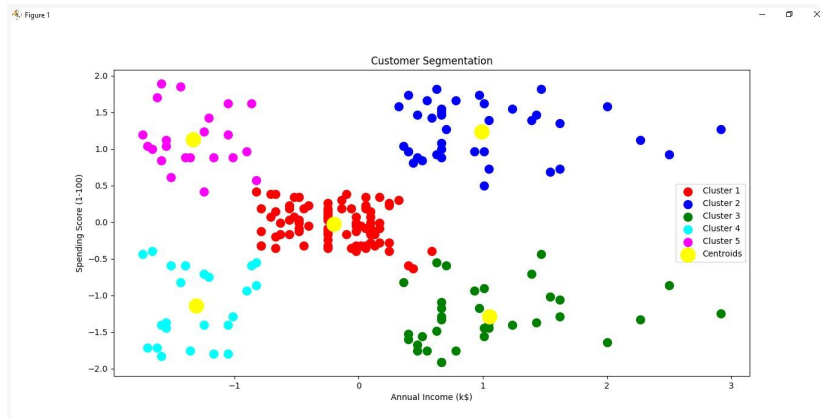
    print(cluster_data.describe())

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

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

```

Output:



```
IDE Shell 3.11.3
File Edit Shell Debug Options Window Help
>>>
== RESTART: C:\Users\abhar\AppData\Local\Programs\Python\Python311\newfile1.py =
CustomerID  Genre  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

Cluster 0 Statistics
CustomerID  Age  ...  Spending Score (1-100)  Cluster
count      61.000000  81.000000  ...  21.000000  81.0
mean       86.320988  42.736049  ...  49.518519  0.0
std        24.240889  16.447822  ...   6.530809  0.0
min       44.000000  15.000000  ...  34.000000  0.0
25%       66.000000  27.000000  ...  44.000000  0.0
50%       86.000000  46.000000  ...  50.000000  0.0
75%      106.000000  54.000000  ...  55.000000  0.0
max      143.000000  70.000000  ...  61.000000  0.0

[8 rows x 5 columns]
Cluster 1 Statistics
CustomerID  Age  ...  Spending Score (1-100)  Cluster
count      39.000000  39.000000  ...  39.000000  39.0
mean       40.000000  32.482308  ...  82.125208  1.0
std        22.803509  3.728450  ...   9.364489  0.0
min       174.000000  27.000000  ...  63.000000  1.0
25%       143.000000  30.000000  ...  74.500000  1.0
50%       162.000000  32.000000  ...  83.000000  1.0
75%       181.000000  35.500000  ...  90.000000  1.0
max       200.000000  40.000000  ...  97.000000  1.0

[8 rows x 5 columns]
Cluster 2 Statistics
CustomerID  Age  ...  Spending Score (1-100)  Cluster
count      35.000000  35.000000  ...  35.000000  35.0
mean      164.371429  41.114286  ...  17.114286  2.0
std        21.457325  11.341676  ...   9.952154  0.0
min       128.000000  19.000000  ...   1.000000  2.0
25%       148.000000  34.000000  ...  10.000000  2.0
50%       145.000000  42.000000  ...  16.000000  2.0
75%       182.000000  47.500000  ...  23.500000  2.0
max       196.000000  55.000000  ...  35.000000  2.0

[8 rows x 5 columns]
Cluster 3 Statistics
CustomerID  Age  ...  Spending Score (1-100)  Cluster
count      23.000000  23.000000  ...  23.000000  23.0
mean       25.000000  45.217391  ...  20.913043  3.0
std        13.564466  13.228407  ...  13.017167  0.0
min         1.000000  15.000000  ...   3.000000  3.0
25%        12.000000  35.500000  ...   9.500000  3.0
50%        23.000000  46.000000  ...  17.000000  3.0
75%        34.000000  53.500000  ...  33.500000  3.0
max        45.000000  67.000000  ...  40.000000  3.0

[8 rows x 5 columns]
Cluster 4 Statistics
CustomerID  Age  ...  Spending Score (1-100)  Cluster
count      22.000000  22.000000  ...  22.000000  22.0
mean       33.090909  25.272727  ...  79.364636  4.0
std        13.147185  5.287030  ...  10.504174  0.0
min         2.000000  18.000000  ...  62.000000  4.0
25%       12.500000  21.250000  ...  73.000000  4.0
50%       23.000000  23.500000  ...  77.000000  4.0
75%       33.500000  25.750000  ...  85.750000  4.0
max        46.000000  35.000000  ...  99.000000  4.0

[8 rows x 5 columns]
>>>
```

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.