Customer Segmentation using K-Means — Full Documentation + Complete Code

This document contains full analysis + ALL code from the notebook exactly as provided (no changes).

1. Overview

This project performs Customer Segmentation using K-Means Clustering.

The notebook includes dataset exploration, scaling, clustering, visualization, and final segmentation.

2. Libraries Used

• pandas — data loading and manipulation

• numpy — numerical operations

• sklearn.cluster — KMeans

• sklearn.preprocessing — StandardScaler

• matplotlib / seaborn — data visualization

3. Workflow Steps

---

1. Load customer dataset

2. Explore distribution of features

3. Standardize the numerical data

4. Use Elbow Method to determine optimal number of clusters

5. Train K-Means clustering algorithm

6. Visualize clustered segments

7. Assign cluster labels to dataset

4. Important Variables

Common variables detected include:

• data / df

• scaler

• scaled_data

• kmeans

• labels

• cluster_centers

5. Notes

• Model saving not included

• Notebook primarily focuses on visualization + clustering

• No missing imports detected

6. Complete Notebook Code

Below is each code cell EXACTLY as it exists in the notebook.

Code Cell 1

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.cluster import KMeans
```

Code Cell 2

```
#loading the file
customer_data = pd.read_csv("/content/drive/MyDrive/Data Science/Projects/13 Customer Segmentation us
customer_data.head()
```

Code Cell 3

```
#Finding the number of rows and cols
```

```
customer_data.shape
```

Code Cell 4

```
#check info for the data
customer_data.info()
```

Code Cell 5

```
#check null values
customer_data.isnull().sum()
```

Code Cell 6

```
customer_data.describe()
```

Code Cell 7

```
x = customer_data.iloc[:,[3,4]].values
```

Code Cell 8

```
#finding the wcss value for different number of cluster
```

Code Cell 9

```
wcss = []

for i in range(1,11):
  kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
  kmeans.fit(x)
  wcss.append(kmeans.inertia_)
```

Code Cell 10

```
#plot an elbow graph
sns.set()
plt.plot(range(1,11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
```

Code Cell 11

```
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=0)

#return a label for each data point based on their cluster
y = kmeans.fit_predict(x)
print(y)
```

Code Cell 12

```
#plotting all the clusters and their centroids
```

Code Cell 13

```
plt.figure(figsize=(8,8))
plt.scatter(x[y==0,0], x[y==0,1], s=50, c='green', label ='Cluster 1')
plt.scatter(x[y==1,0], x[y==1,1], s=50, c='red', label ='Cluster 2')
plt.scatter(x[y==2,0], x[y==2,1], s=50, c='yellow', label ='Cluster 3')
plt.scatter(x[y==3,0], x[y==3,1], s=50, c='violet', label ='Cluster 4')
plt.scatter(x[y==4,0], x[y==4,1], s=50, c='blue', label ='Cluster 5')


# plot the centroids
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=100, c='cyan', label='Centr

plt.title('Customer Groups')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.show()
```

Code Cell 14

```
# (Empty code cell)
```