import pandas as pd # Importing pandas for data handling

import numpy as np # Importing numpy for numerical operations

import matplotlib.pyplot as plt # Importing matplotlib for visualization

from sklearn.cluster import KMeans # Importing KMeans for clustering

from sklearn.metrics import silhouette\_score # Importing silhouette score for evaluation

df = pd.read\_csv("customer\_segmentation.csv") # Reading the dataset from CSV file

X = df[["Age", "Annual\_Income", "Spending\_Score"]]

kmeans = KMeans(n\_clusters=3, random\_state=42) # Initializing K-Means with 3 clusters

df["cluster"] = kmeans.fit\_predict(X)

inertia = kmeans.inertia\_ # Inertia (sum of squared distances to closest centroid)

silhouette = silhouette\_score(X, df["cluster"]) # Silhouette score for clustering quality

# Visualization

plt.figure(figsize=(8,6)) # Setting figure size for the plot

plt.scatter(df["Annual\_Income"], df["Spending\_Score"], c=df["cluster"], cmap='viridis', alpha=0.6) # Plotting clustered data points

plt.xlabel("Annual Income") # Setting x-axis label

plt.ylabel("Spending Score") # Setting y-axis label

plt.title("K-Means Clustering - Customer Segmentation") # Setting plot title

plt.colorbar(label="Cluster") # Adding color bar to indicate clusters

plt.show() # Displaying the plot

# Print evaluation metrics

print(f"Total Samples: {len(df)}") # Printing total number of samples in dataset

print(f"Cluster counts:\n{df['cluster'].value\_counts()}") # Printing number of samples in each cluster

print(f"Inertia: {inertia}") # Printing inertia value

print(f"Silhouette Score: {silhouette}") # Printing silhouette score