

```

# Customer Segmentation Using K-Means Clustering

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Step 1: Load Data
data = pd.read_csv("Mall_Customers.csv")
print(data.head())

# Step 2: Select Relevant Features
X = data[["Age", "Annual Income (k$)", "Spending Score (1-100)"]]

# Step 3: Standardize Features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 4: Find Optimal Number of Clusters (Elbow Method)
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(8,5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title("Elbow Method for Optimal Clusters")
plt.xlabel("Number of clusters")
plt.ylabel("WCSS")
plt.grid()
plt.show()

# Step 5: Apply K-Means
k = 5
kmeans = KMeans(n_clusters=k, init='k-means++', random_state=42)
y_kmeans = kmeans.fit_predict(X_scaled)

# Step 6: Add Cluster Labels
data['Cluster'] = y_kmeans

# Step 7: Visualize the Clusters
plt.figure(figsize=(10,6))
sns.scatterplot(data=data, x="Annual Income (k$)", y="Spending Score (1-100)",
                hue="Cluster", palette='Set2', s=100)
plt.title("Customer Segments")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.legend()
plt.show()

```