**# Import necessary libraries**

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**from sklearn.cluster import KMeans**

**# Load the dataset**

**mydata = pd.read\_csv("Mall\_Customers.csv")**

**print(mydata.head()) # Display the first few rows of the dataset**

**# Select features and create a copy to avoid warnings**

**iv = mydata[['Annual Income (k$)', 'Spending Score (1-100)']].copy()**

**# Perform K-Means clustering**

**kmeans = KMeans(n\_clusters=5, n\_init=10, random\_state=0)**

**kmeans.fit(iv)**

**# Add the cluster assignments to the DataFrame**

**iv['cluster'] = kmeans.predict(iv)**

**# Visualize the Clusters**

**plt.scatter(iv.loc[iv['cluster'] == 0, 'Annual Income (k$)'],**

**iv.loc[iv['cluster'] == 0, 'Spending Score (1-100)'],**

**s=100, c='red', label='Careful')**

**plt.scatter(iv.loc[iv['cluster'] == 1, 'Annual Income (k$)'],**

**iv.loc[iv['cluster'] == 1, 'Spending Score (1-100)'],**

**s=100, c='green', label='Standard')**

**plt.scatter(iv.loc[iv['cluster'] == 2, 'Annual Income (k$)'],**

**iv.loc[iv['cluster'] == 2, 'Spending Score (1-100)'],**

**s=100, c='blue', label='Target')**

**plt.scatter(iv.loc[iv['cluster'] == 3, 'Annual Income (k$)'],**

**iv.loc[iv['cluster'] == 3, 'Spending Score (1-100)'],**

**s=100, c='grey', label='Careless')**

**plt.scatter(iv.loc[iv['cluster'] == 4, 'Annual Income (k$)'],**

**iv.loc[iv['cluster'] == 4, 'Spending Score (1-100)'],**

**s=100, c='brown', label='Sensible')**

**# Uncomment the following lines to plot cluster centroids if needed**

**# plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1],**

**# s=200, c='yellow', label='Centers')**

**plt.title("Results of K Means Clustering")**

**plt.xlabel("Annual Income (k$)")**

**plt.ylabel("Spending Score (1-100)")**

**plt.legend()**

**plt.show()**

**# Displaying data for all clusters**

**for cluster\_num in range(5): # Assuming there are 5 clusters**

**print(f"Cluster {cluster\_num} Data:")**

**cluster\_data = iv[iv['cluster'] == cluster\_num]**

**print(cluster\_data)**

**print("\n") # Add a newline for better readability**