```
## K-Means Clustering
#=======================

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# load the customer data into a DataFrame
customer_df = pd.read_csv('customer_data.csv')

# Check the first 5 rows
customer_df.head()

#Explore Data (Warning may appear due to missmatch of virsion)

# Age vs. Spending Score
plt.scatter(customer_df["Age"],
            customer_df["Spending Score (1-100)"])


plt.xlabel("Age")
plt.ylabel("Spending Score (1-100)")

#Age Vs. Annual Income

plt.scatter(customer_df["Age"],
            customer_df["Annual Income (k$)"])

plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")

# Spending Score Vs Annual Income

plt.scatter(customer_df["Spending Score (1-100)"],
            customer_df["Annual Income (k$)"])

plt.xlabel("Spending Score (1-100)")
plt.ylabel("Annual Income (k$)")

# Check for null values
customer_df.isnull().sum()

#Get relevant columns for clustering

relevant_cols = ["Age", "Annual Income (k$)", "Spending Score (1-100)"]

customer_df = customer_df[relevant_cols]

from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()

scaler.fit(customer_df)

scaled_data = scaler.transform(customer_df)


# Determine the best number of cluster

def find_best_clusters(df, maximum_K):

    clusters_centers = []
    k_values = []

    for k in range(1, maximum_K):

        kmeans_model = KMeans(n_clusters = k)
        kmeans_model.fit(df)

        clusters_centers.append(kmeans_model.inertia_)
        k_values.append(k)


    return clusters_centers, k_values

# Elbow plot

def generate_elbow_plot(clusters_centers, k_values):

    figure = plt.subplots(figsize = (12, 6))
    plt.plot(k_values, clusters_centers, 'o-', color = 'orange')
    plt.xlabel("Number of Clusters (K)")
    plt.ylabel("Cluster Inertia")
    plt.title("Elbow Plot of KMeans")
    plt.show()
clusters_centers, k_values = find_best_clusters(scaled_data, 12)
generate_elbow_plot(clusters_centers, k_values)

#Creating final KMeans model

kmeans_model = KMeans(n_clusters = 5)
kmeans_model.fit(scaled_data)

customer_df["clusters"] = kmeans_model.labels_
customer_df.head()

plt.scatter(customer_df["Spending Score (1-100)"],
            customer_df["Annual Income (k$)"],
            c = customer_df["clusters"])
```