**Practical 7**

a. Perform Simple Linear Regression on the data set Salary\_Data.csv.

## **Step 1: Import the required python packages**

# Import libraries  
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

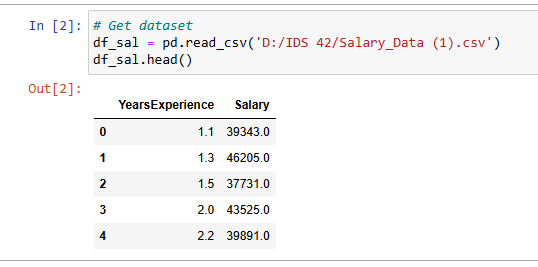
import seaborn as sns

from sklearn.model\_selection import train\_test\_split

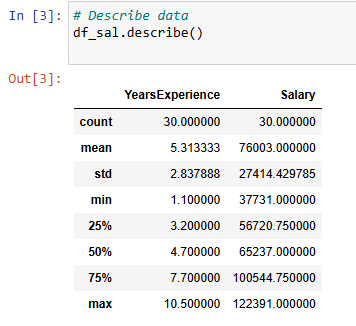
from pandas.core.common import random\_state

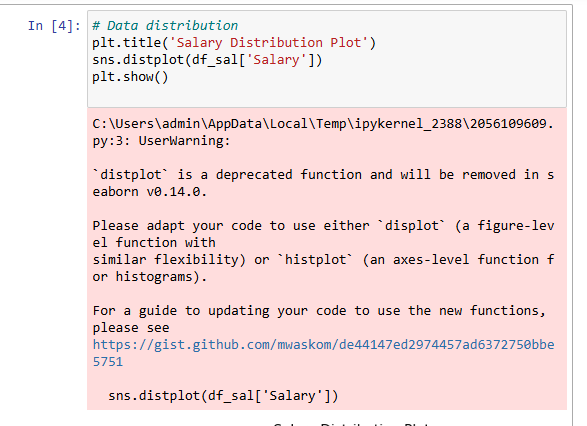
from sklearn.linear\_model import LinearRegression

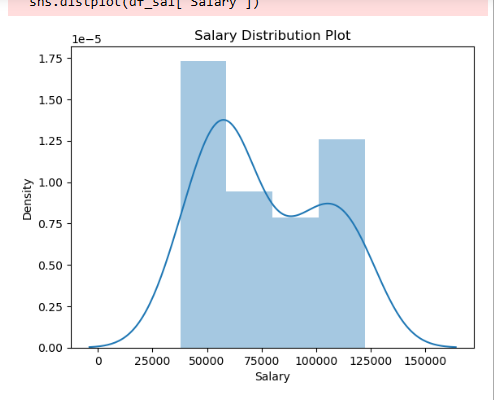
## **Step 2: Load the dataset**

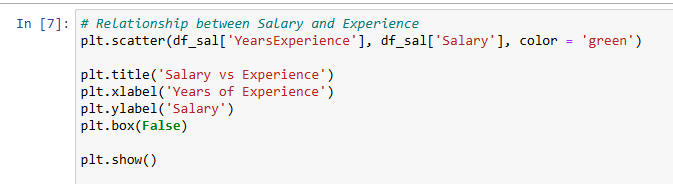


## **Step 3: Data analysis**



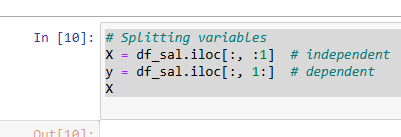


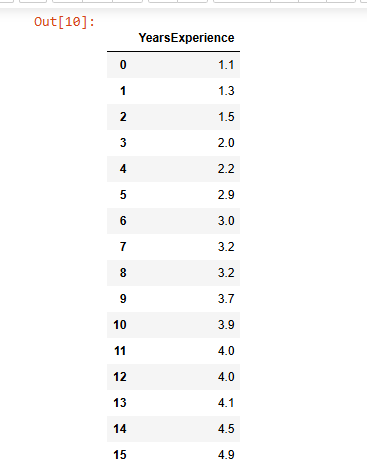


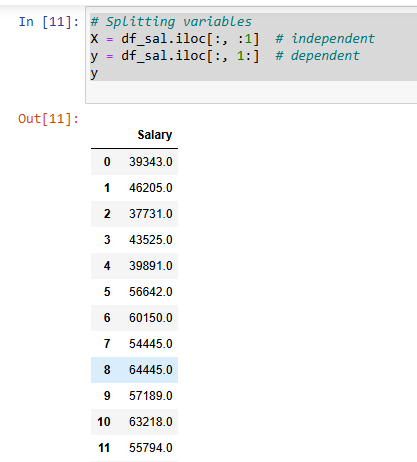




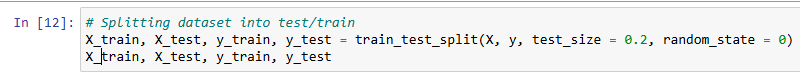
**Step 4: Split the dataset into dependent/independent variables**

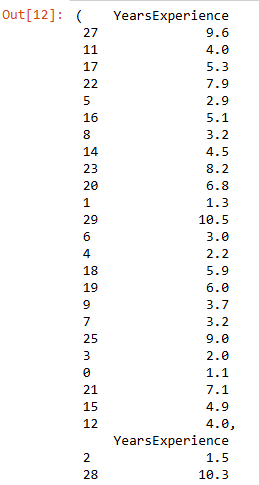




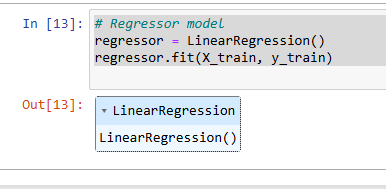


## **Step 4: Split data into Train/Test sets**

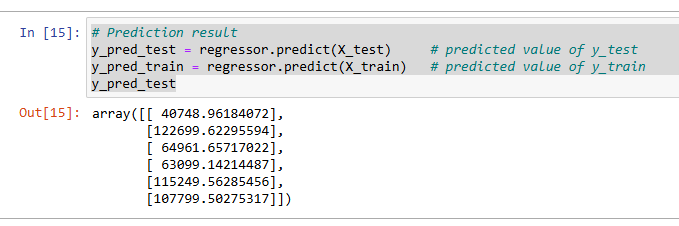


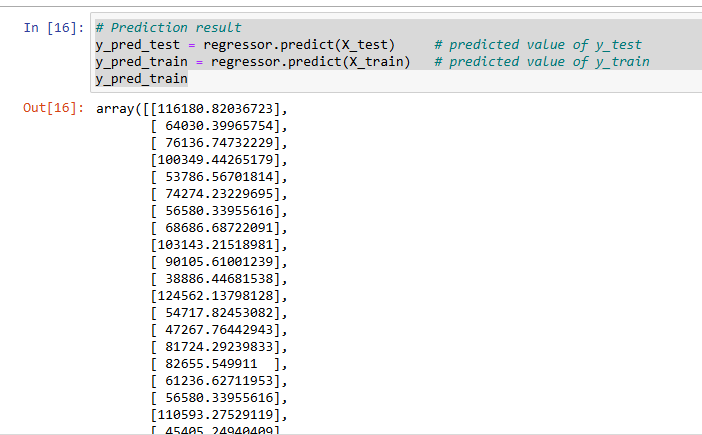


## **Step 5: Train the regression model**

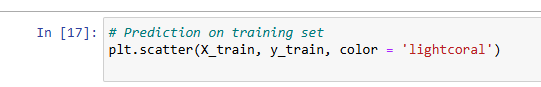


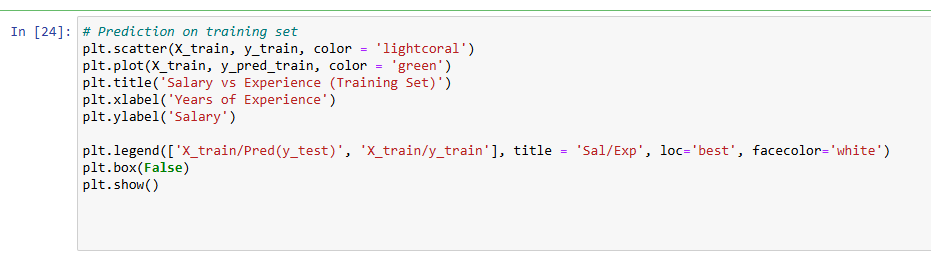
## **Step 6: Predict the result**



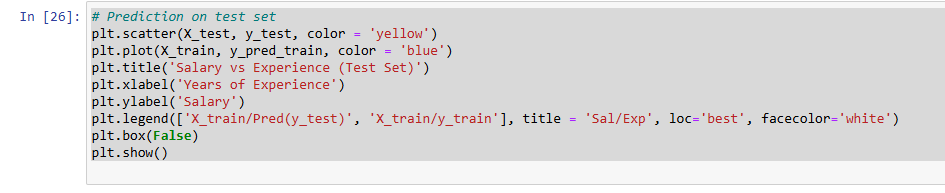


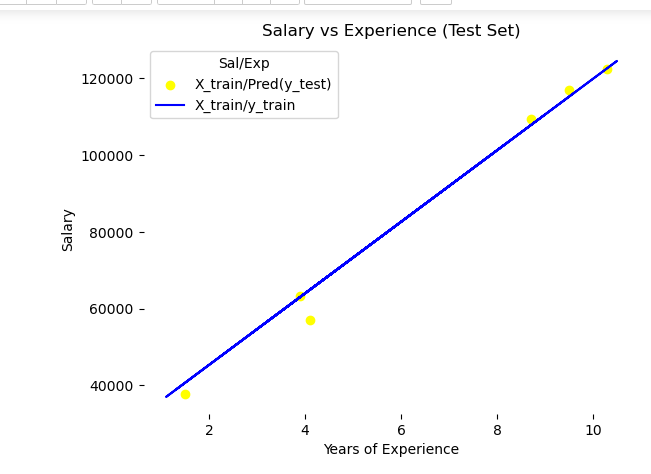
## **Step 7: Plot the training and test results**

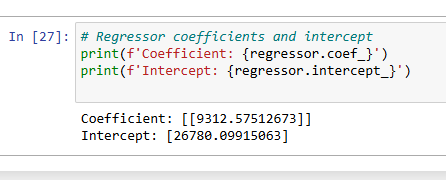




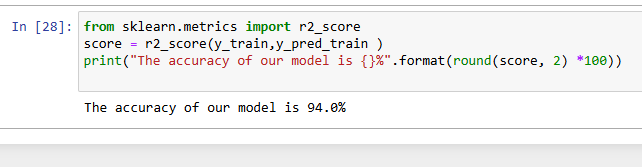








**Testing Accuracy of the Model**



b. .Perform Multiple Linear Regression on the data set Advertising.csv.

**Importing the Libraries**

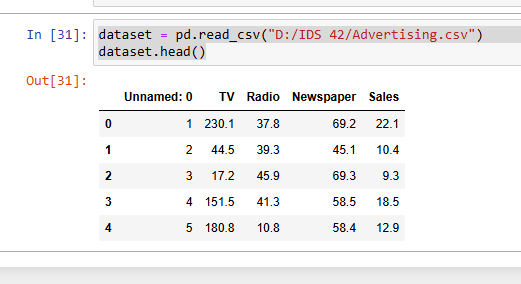
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

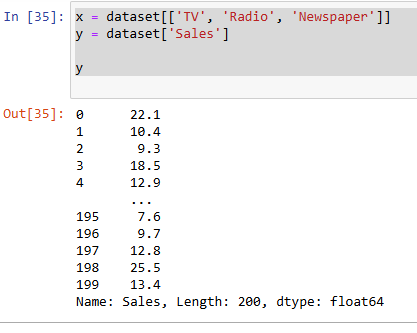
import seaborn as sns

**Reading the Dataset**

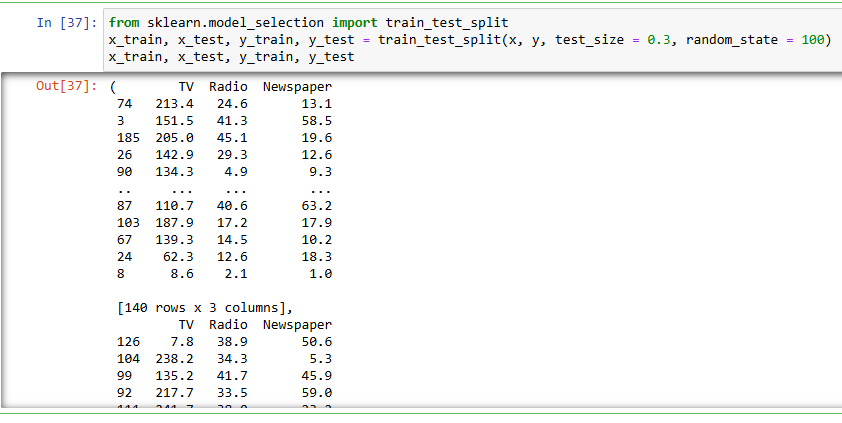


**Setting the values for independent (X) variable and dependent (Y) variable**

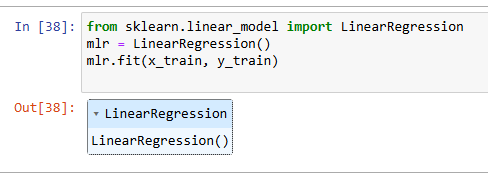




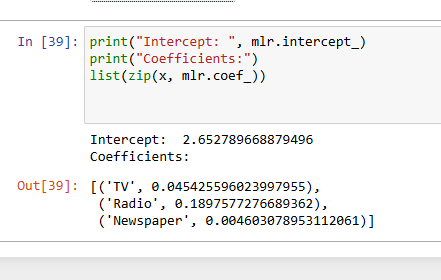
**Splitting the dataset into train and test set**



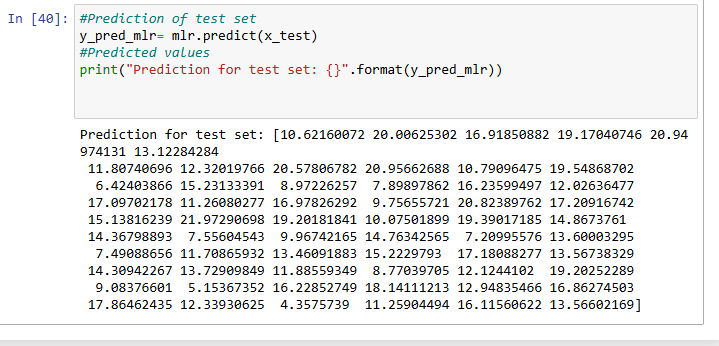
**Implementing the linear model**



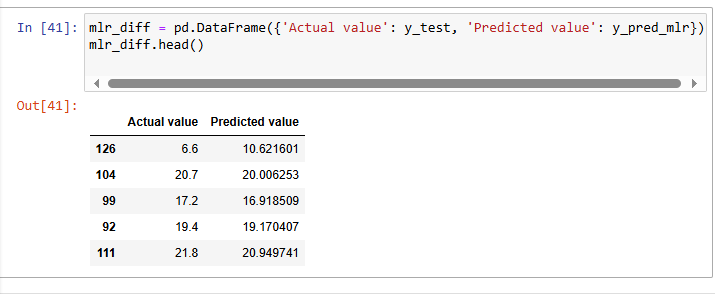
**Model Equation**



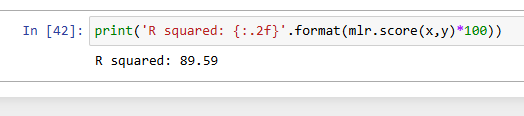
**Prediction on the test set**



**Actual values and the predicted values**



**Evaluating the Model**



Name : Cheruvathoor Abin Anto

Roll No : TIT2425008

**Practical 8 :** **K-means clustering in Action using Python**

**Import libraries and load the data**

import pandas as pd

import matplotlib.pyplot as plt

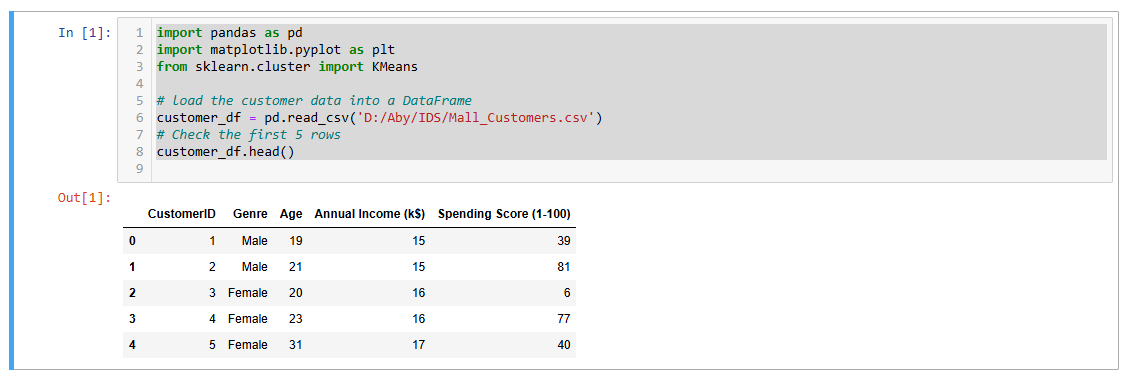
from sklearn.cluster import KMeans

# load the customer data into a DataFrame

customer\_df = pd.read\_csv('D:/Aby/IDS/Mall\_Customers.csv')

# Check the first 5 rows

customer\_df.head()



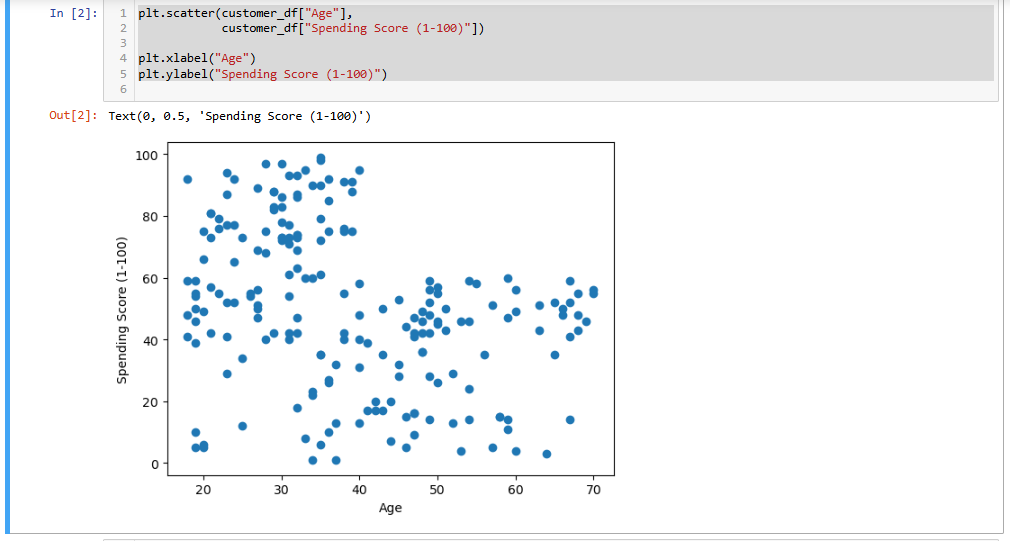
**Explore the data**

plt.scatter(customer\_df["Age"],

customer\_df["Spending Score (1-100)"])

plt.xlabel("Age")

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

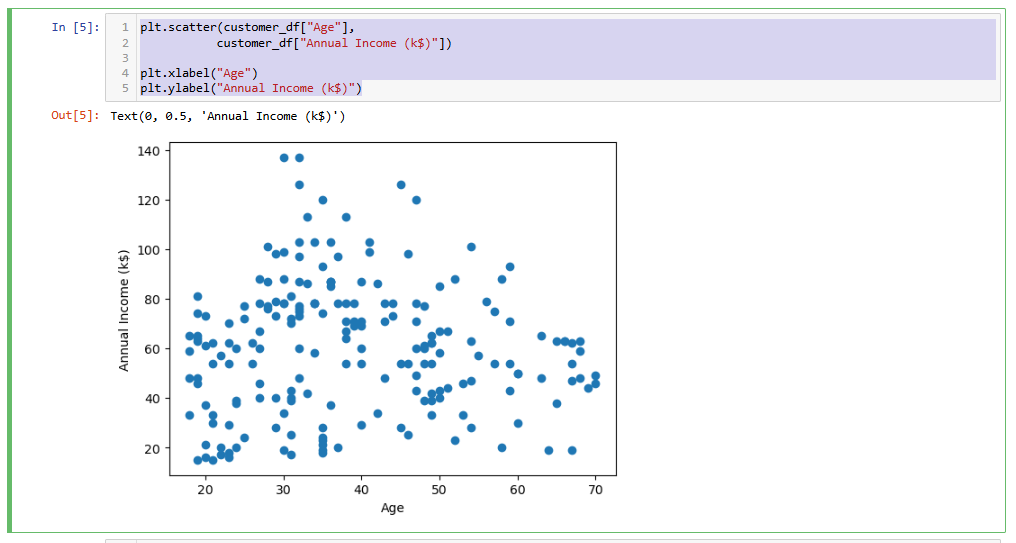


plt.scatter(customer\_df["Age"],

customer\_df["Annual Income (k$)"])

plt.xlabel("Age")

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

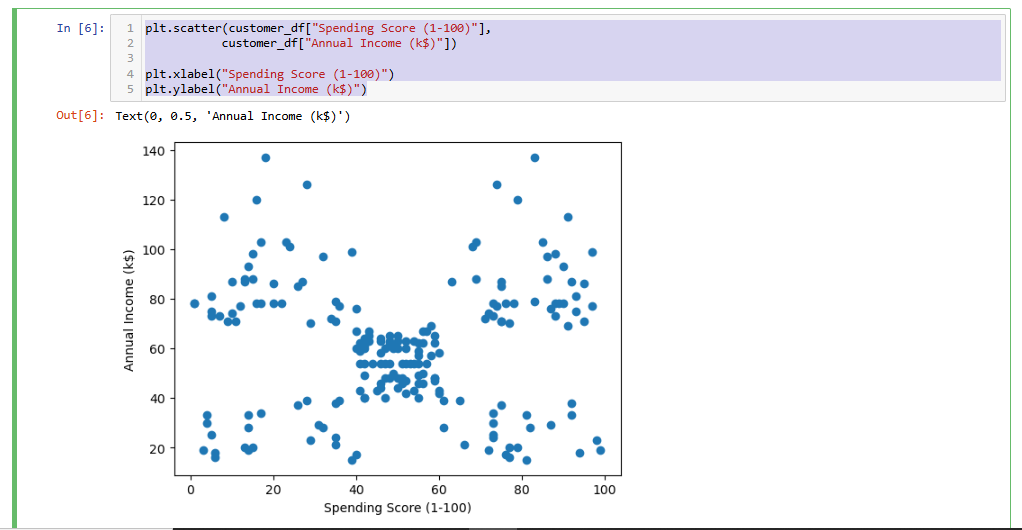


plt.scatter(customer\_df["Spending Score (1-100)"],

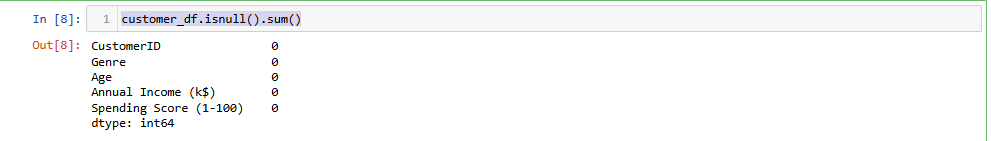
customer\_df["Annual Income (k$)"])

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

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



customer\_df.isnull().sum()



**Get the relevant columns for clustering**

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

customer\_df = customer\_df[relevant\_cols]

**Data Transformation**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(customer\_df)

scaled\_data = scaler.transform(customer\_df)

**Determine the best number of cluster**

Below is the code for the first function:

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

**The second function uses those inertias and K values to generate the final 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")

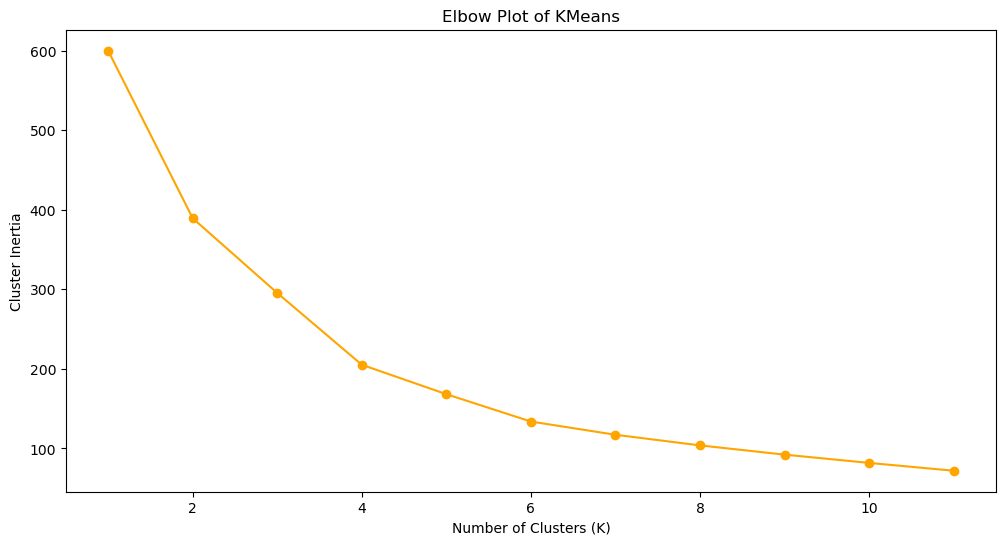
plt.title("Elbow Plot of KMeans")

plt.show()

**Now, we can apply the above 2 functions to the dataset using a maximum K value of 12 and get the final result.**

clusters\_centers, k\_values = find\_best\_clusters(scaled\_data, 12)

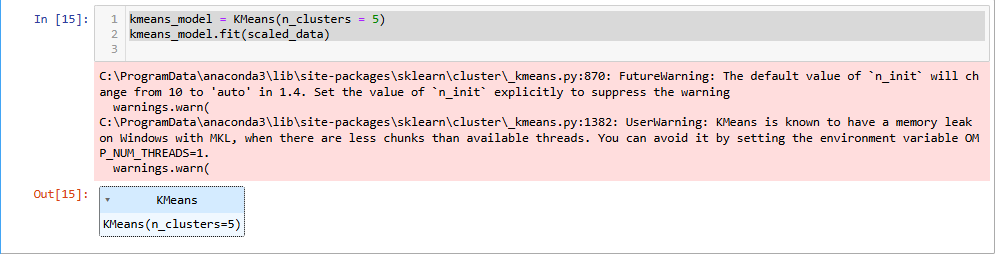
generate\_elbow\_plot(clusters\_centers, k\_values)



**Create the final KMeans model**

kmeans\_model = KMeans(n\_clusters = 5)

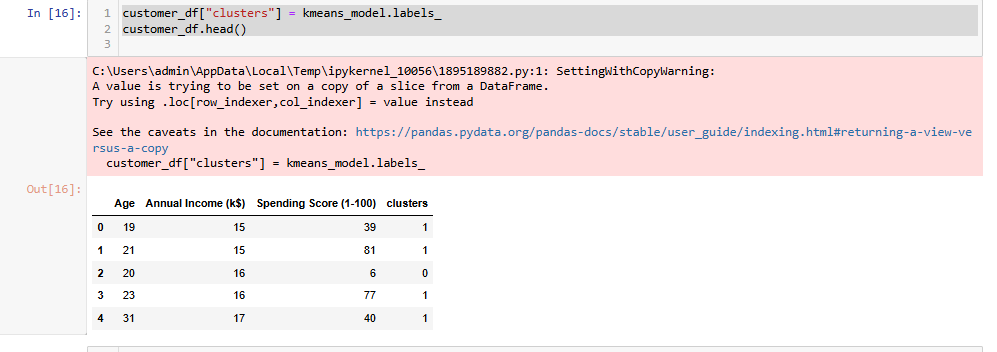
kmeans\_model.fit(scaled\_data)



**The .fit() method is used to train the K-Means clustering model on the given dataset (scaled\_data).**

customer\_df["clusters"] = kmeans\_model.labels\_

customer\_df.head()



**Visualize the clusters**

plt.scatter(customer\_df["Spending Score (1-100)"],

customer\_df["Annual Income (k$)"],

c = customer\_df["clusters"])

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

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

