MALL CUSTOMER SEGMENTATION

Presented By
Abhishek Sumeet Toppo

TABLE OF CONTENTS

- Abstract
- Introduction
- Existing Model
- Proposed Method
- System Architecture
- Methodology
- Implementation
- Conclusion

ABSTRACT

Effective decisions are mandatory for any company to generate good revenue. In these days competition is huge and all companies are moving forward with their own different strategies. We should use data and take a proper decision. Every person is different from one another and we don't know what he/she buys or what their likes are. But with the help of machine learning technique one can sort out the data and can find the target group by applying several algorithms to the dataset. Without this, It will be very difficult and no better techniques are available to find the group of people with similar character and interests in a large dataset. Here, The customer segmentation using K-Means clustering helps to group the data with same attributes which exactly helps to business the best. We are going to use elbow method to find the number of clusters and at last we visualize the data.

INTRODUCTION

- Nowadays the competition is vast and lot of technologies came into account for effective growth and revenue generation. For every business the most important component is data. With the help of grouped or ungrouped data, we can perform some operations to find customer interests.
- Customer Segmentation is useful to divide the large data from dataset into several groups based on their Gender, Age, Annual Income and Spending Score. These groups are also known as clusters. By this, we can get to know that which age group are purchasing more their gender type, their income and their spending scores. And we can target that age group people the most.

- Initially we are going to work upon the mall customer dataset. Apply the K-means clustering algorithm on the given dataset to us and we have to find the number of clusters first. So, at lastly, we have to visualize the data. One can easily find the potential group of data while observing that visualization.
- The goal is to identify customer segments using the K Means Clustering. The elbow method determines the optimal clusters.

EXISTING MODEL

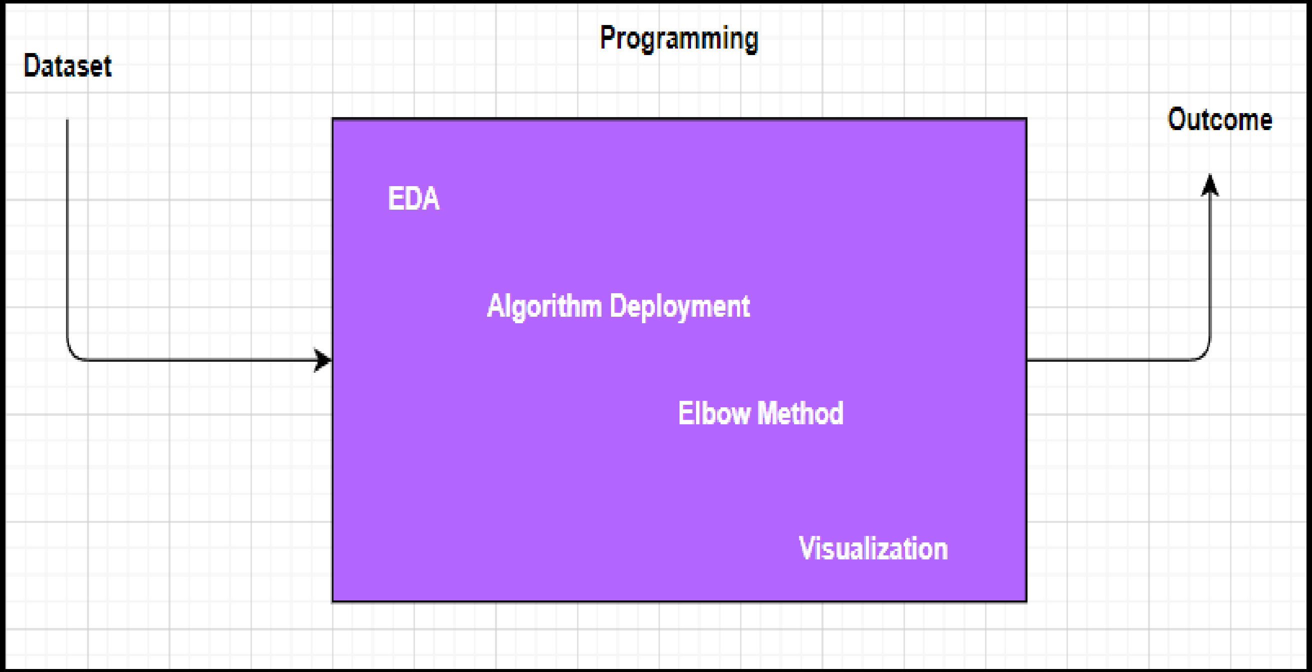
The existing method is storing customer data through paperwork and computer software (digital data) is increasing day by day. At end of the day they will analyze their data as how many things are sold or actual customer count etc. By analyzing the collected data they got to know who is beneficial to their business and increase their sales. It requires more time and more paperwork. Also it is not much effective solution to find the desired customers data.

PROPOSED METHOD

To overcome the traditional method i.e. paper work and computerized digital data this new method will play vital role. As we collect a vast data day by day which requires more paperwork and time to do. As new technologies were emerging in today’s world. Machine Learning which is powerful innovation which is used to predict the final outcome which has many algorithms. So for our problem statement we will use K-Means Clustering which groups the data into different clusters based on their similar characteristics. And then we will visualize the data.

SYSTEM ARCHITECTURE

- Initially we will see the dataset and then we will perform exploratory data analysis which deals with the missing data, duplicates values and null values. And then we will deploy our algorithm k-means clustering which is unsupervised learning in machine learning.



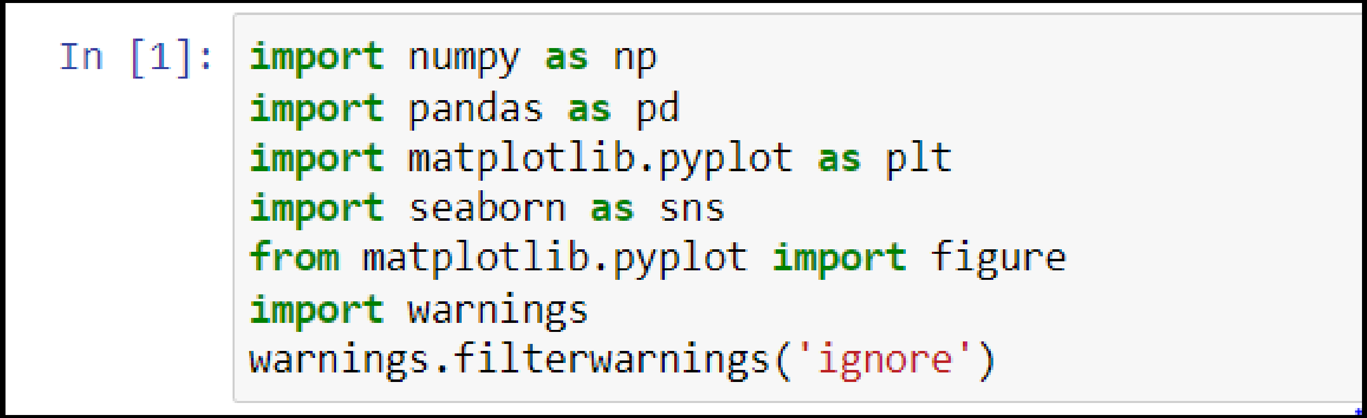
- As in order to find the no of clusters we use elbow method where distance will be calculate through randomly chosen centres and repeat it until there is no change in cluster centres. Thereafter we will analyse the data through data visualization. Finally we will get the outcome.

METHODOLOGY

- First of all we will import all the necessary libraries (Pandas, NumPy, Seaborn, Mathplotlib, Warnings).
- Then we will read dataset and analyze whether it contains any null values, missing values and duplicate values. So we will fix them by dropping or fixing the value with their means, medians etc. which is technically named as Data Preprocessing.
- We will deploy our model algorithm K-Means Clustering, which divides the data into group of clusters based on similar characteristics. To find number of clusters we will use elbow method.
- Finally, we will visualize our data using Matplot, which concludes the customers divided into groups who are similar to each other on their group.

IMPLEMENTATION

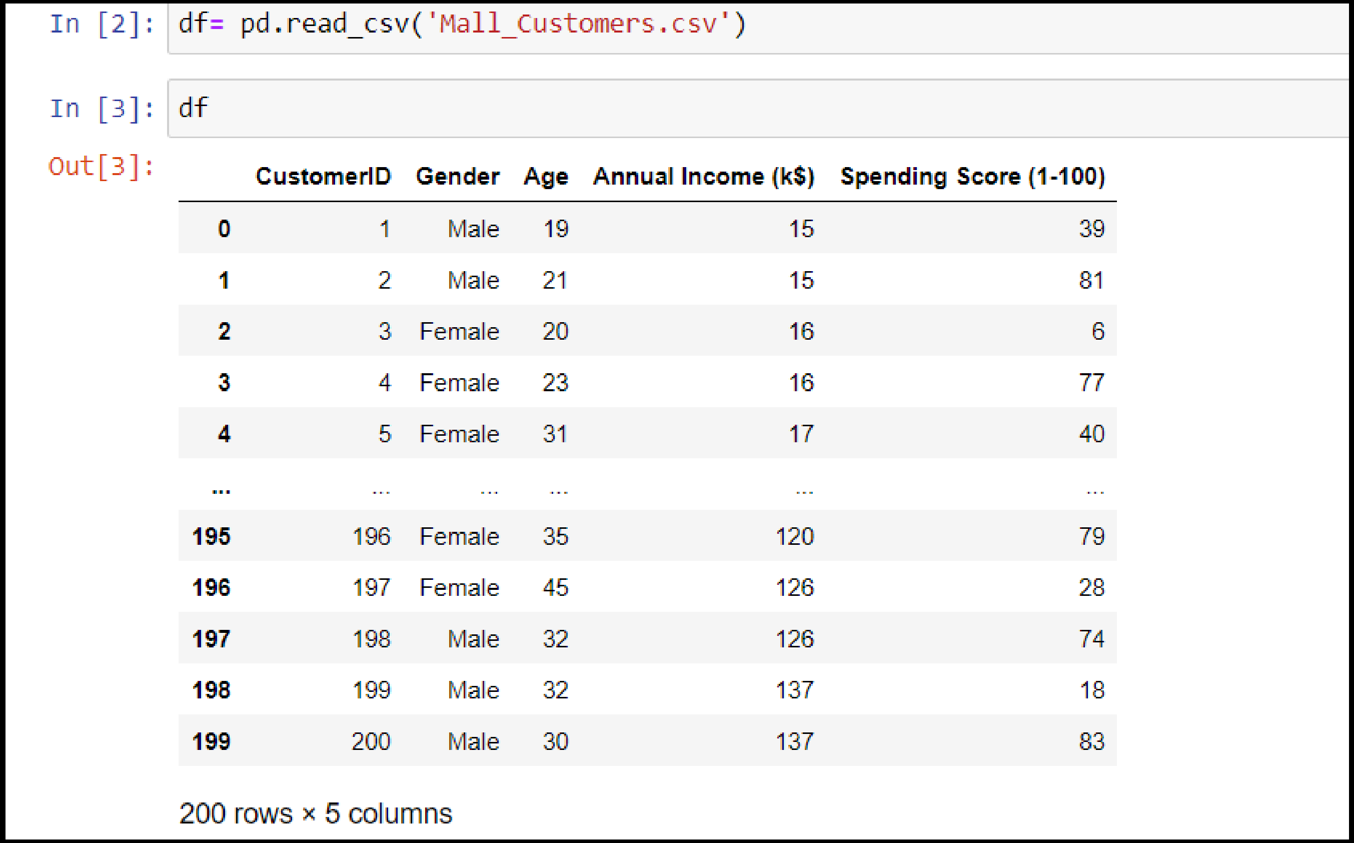
- Importing Libraries



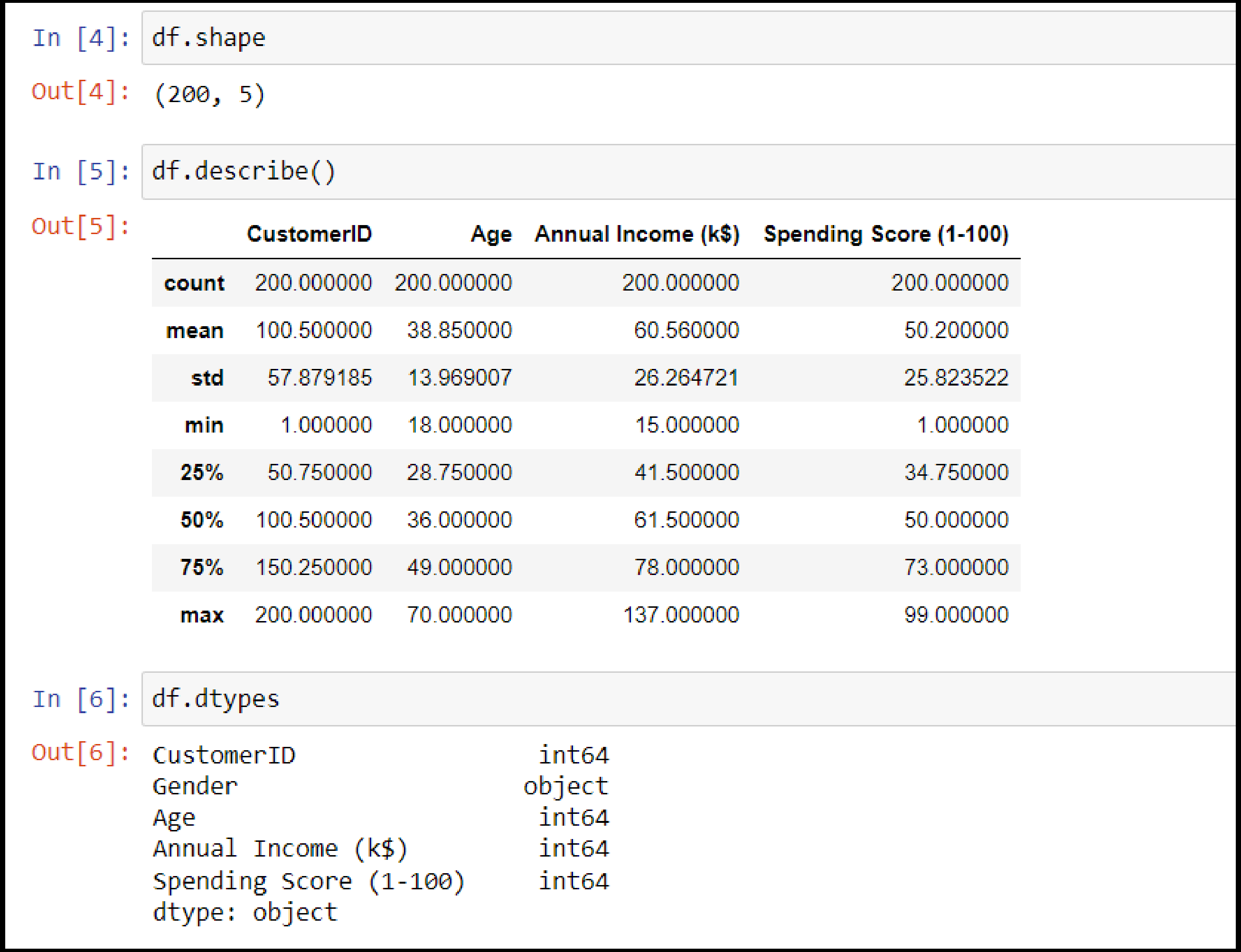
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.pyplot import figure
import warnings
warnings.filterwarnings('ignore')
```

- Overview Of Dataset

  This Mall_Customer dataset contains 200 rows and 5 columns.



```
In [2]: df= pd.read_csv('Mall_Customers.csv')

In [3]: df

Out[3]:
        CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
   0             1    Male   19                  15                      39
   1             2    Male   21                  15                      81
   2             3  Female   20                  16                       6
   3             4  Female   23                  16                      77
   4             5  Female   31                  17                      40
 ...           ...     ...  ...                 ...                     ...
 195           196  Female   35                 120                      79
 196           197  Female   45                 126                      28
 197           198    Male   32                 126                      74
 198           199    Male   32                 137                      18
 199           200    Male   30                 137                      83

200 rows × 5 columns
```



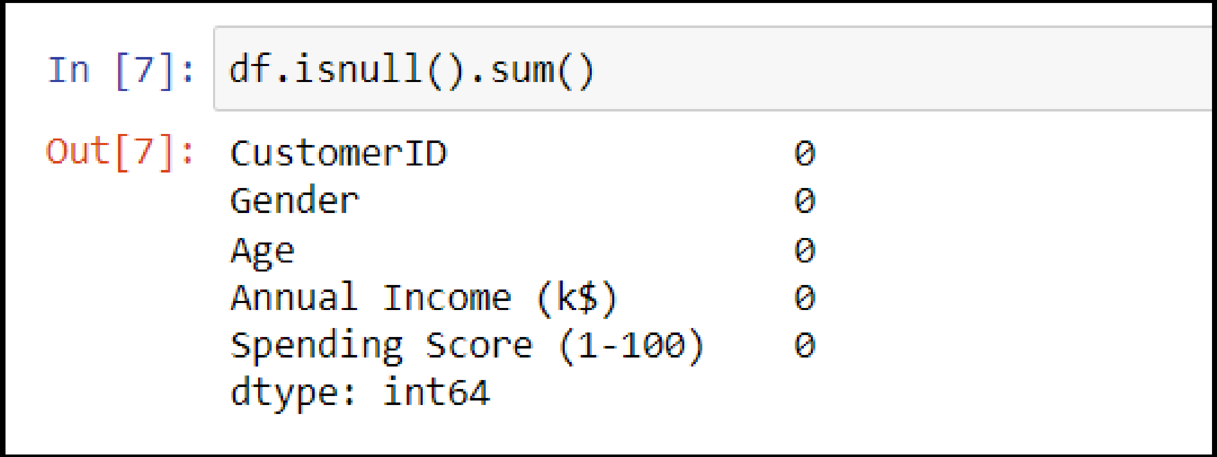
```
In [4]: df.shape
Out[4]: (200, 5)
```

```
In [5]: df.describe()
Out[5]:
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

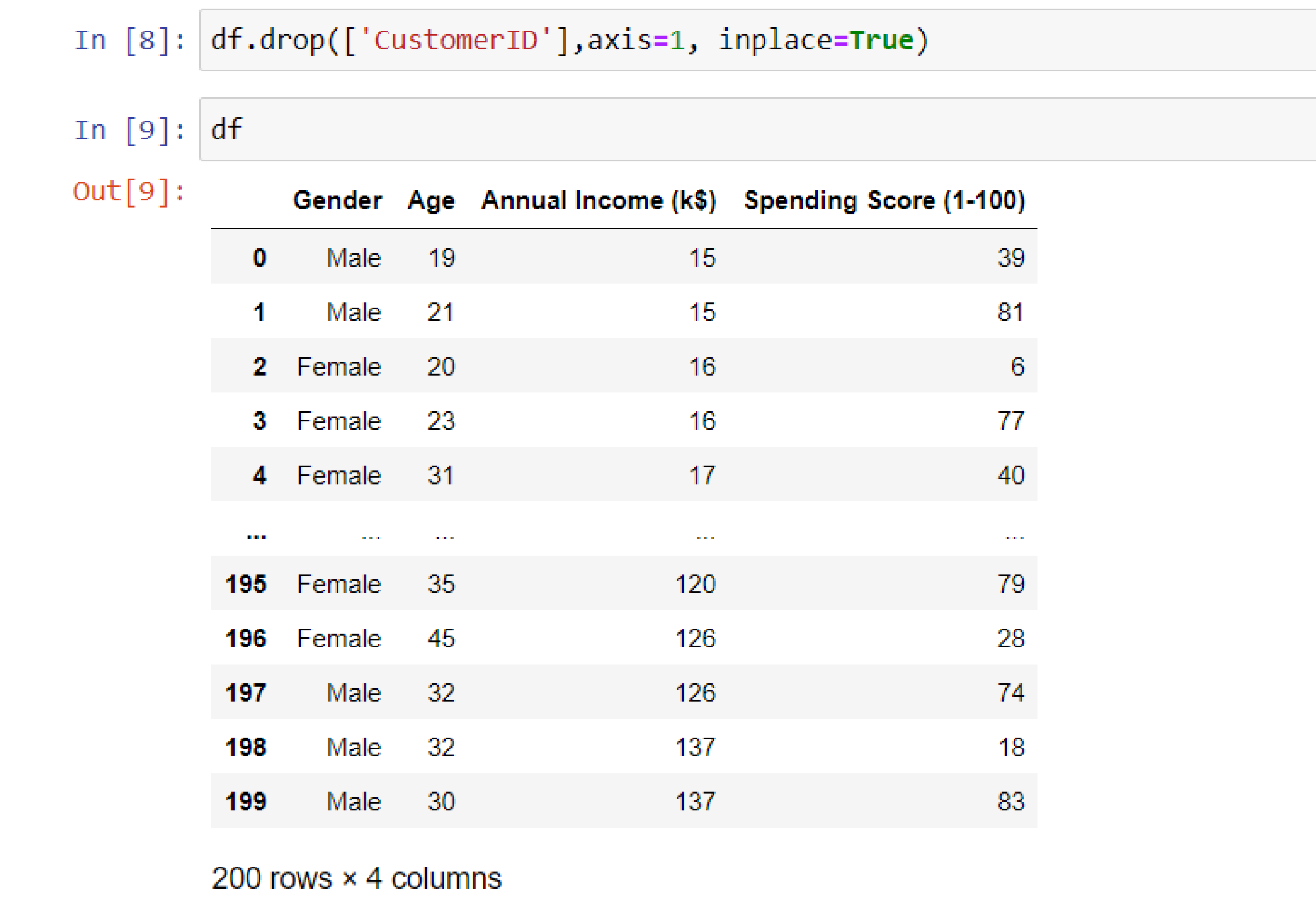
```
In [6]: df.dtypes
Out[6]: CustomerID                 int64
        Gender                    object
        Age                        int64
        Annual Income (k$)         int64
        Spending Score (1-100)     int64
        dtype: object
```

- df.shape() is used to display the size of the dataset
- df.describe() is used to describe the whole dataset
- df.dtypes() is used to show what are the datatypes of all the columns



```
In [7]: df.isnull().sum()

Out[7]: CustomerID                0
        Gender                    0
        Age                       0
        Annual Income (k$)        0
        Spending Score (1-100)    0
        dtype: int64
```

- df.isnull() is used to check whether there are any null values if present in the dataset.
- df.null().sum() is used to count the number of null values. Zero is chosen that no null values are present in the dataset



```
In [8]: df.drop(['CustomerID'],axis=1, inplace=True)
```

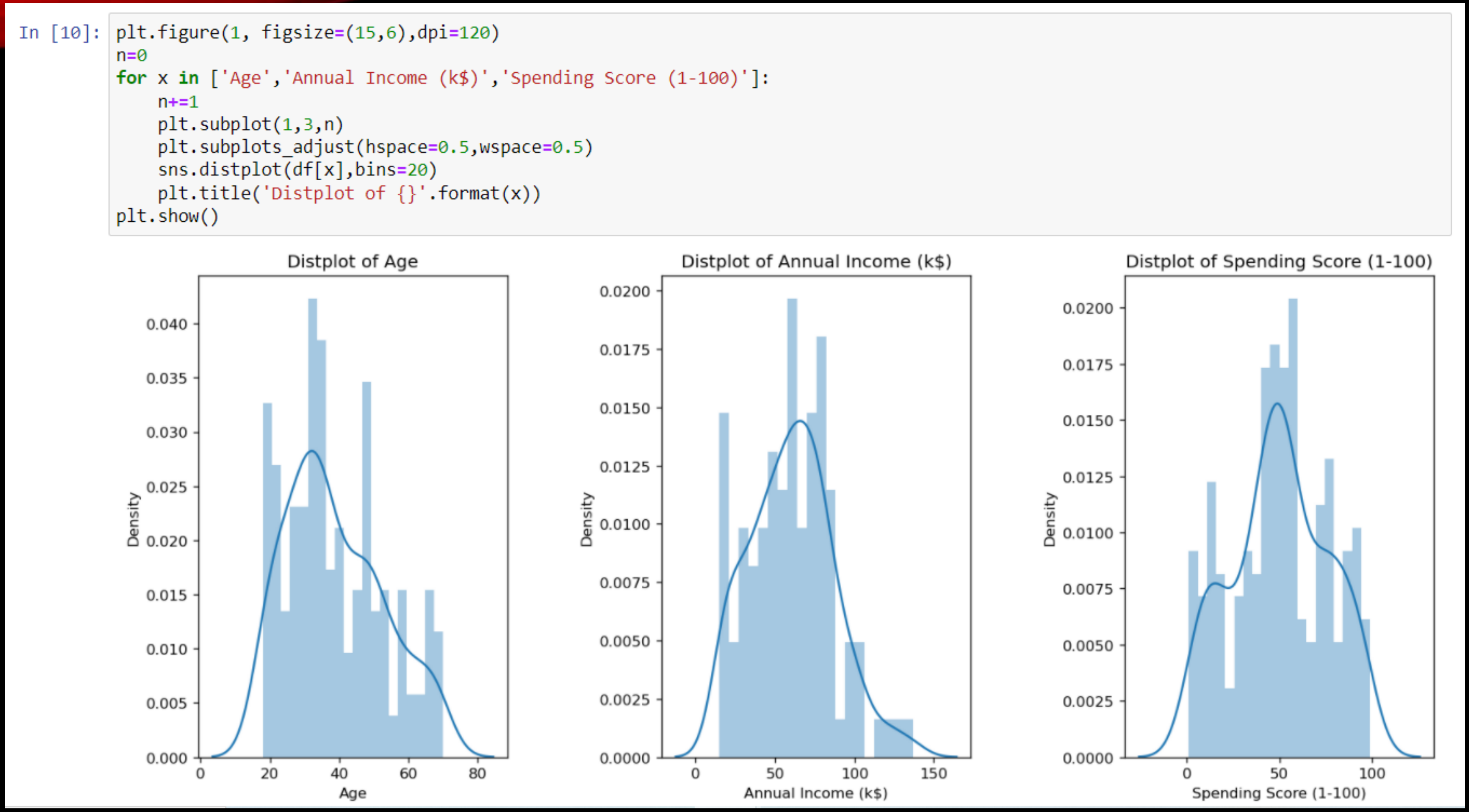
```
In [9]: df
```

Out[9]:

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40
...	...	...	...	...
195	Female	35	120	79
196	Female	45	126	28
197	Male	32	126	74
198	Male	32	137	18
199	Male	30	137	83

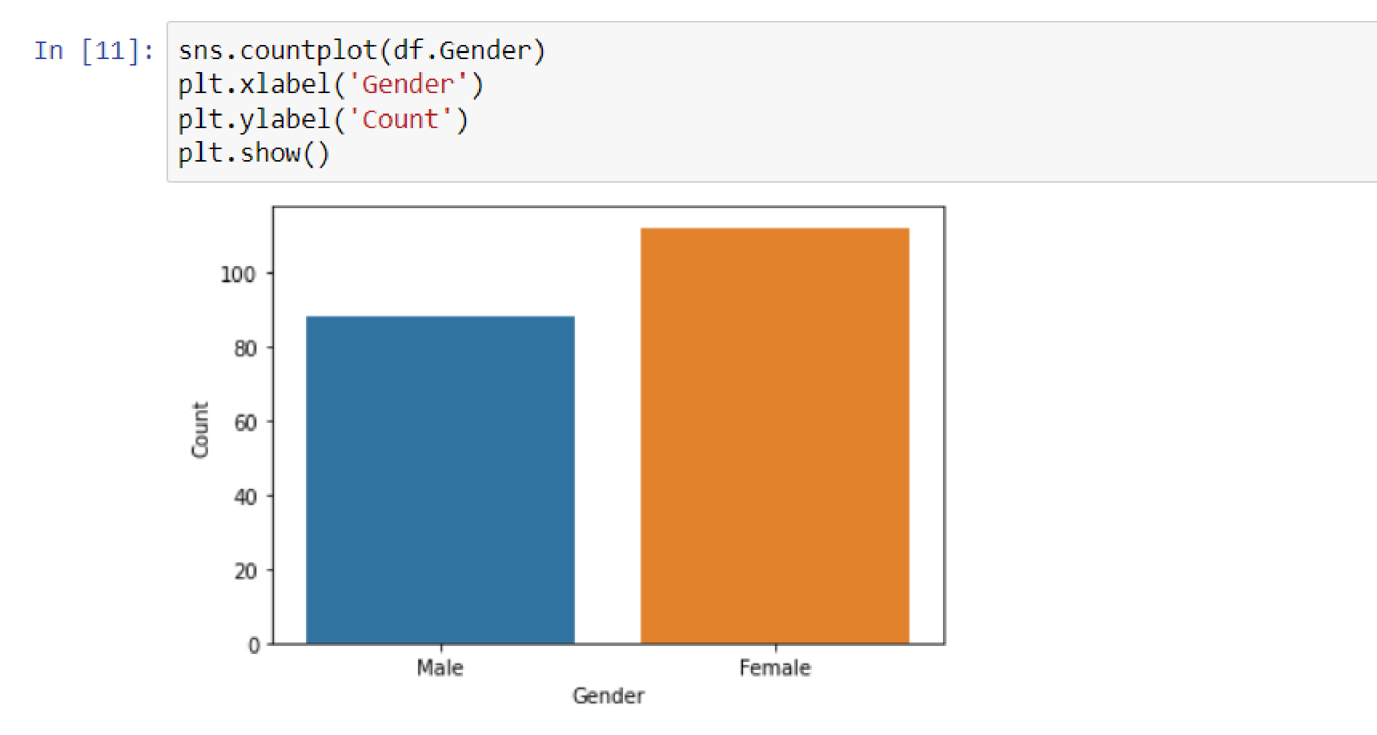
200 rows × 4 columns

- df.drop(['CustomerID'],axis=1, inplace=True) is used to drop the column CustomerID as not necessary to use the column here.



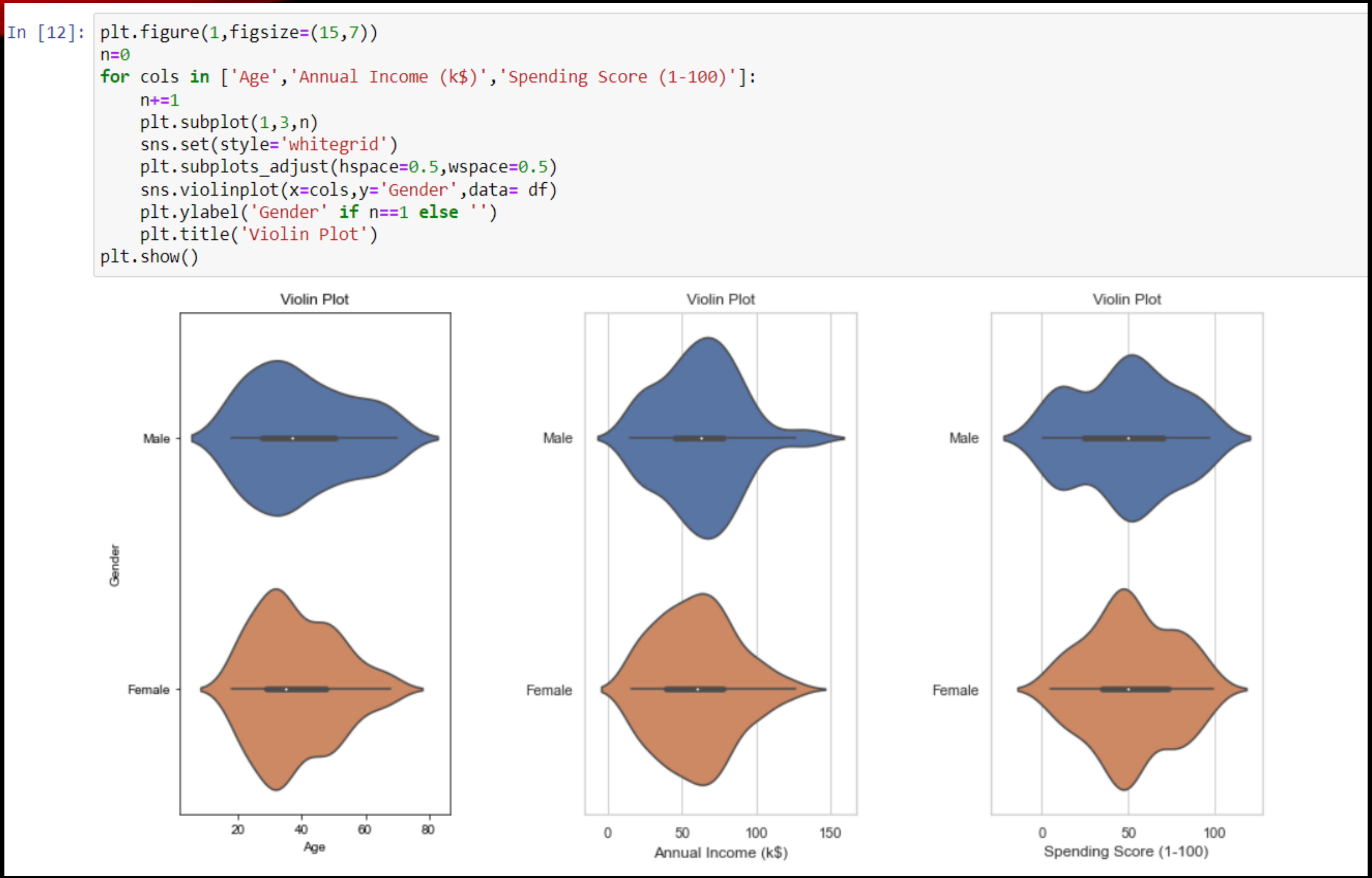
```
In [10]: plt.figure(1, figsize=(15,6),dpi=120)
         n=0
         for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
             n+=1
             plt.subplot(1,3,n)
             plt.subplots_adjust(hspace=0.5,wspace=0.5)
             sns.distplot(df[x],bins=20)
             plt.title('Distplot of {}'.format(x))
         plt.show()
```

- Here a distribution plot is created here for Age, Annual Income(k$) and Spending Score(1-100). From the distribution plot we observe that in 1st Age plot most people in this dataset has between 30 – 35, in the 2nd plot the annual income of the people lies between 60-70 and in 3rd plot the spending score of most people is 50.



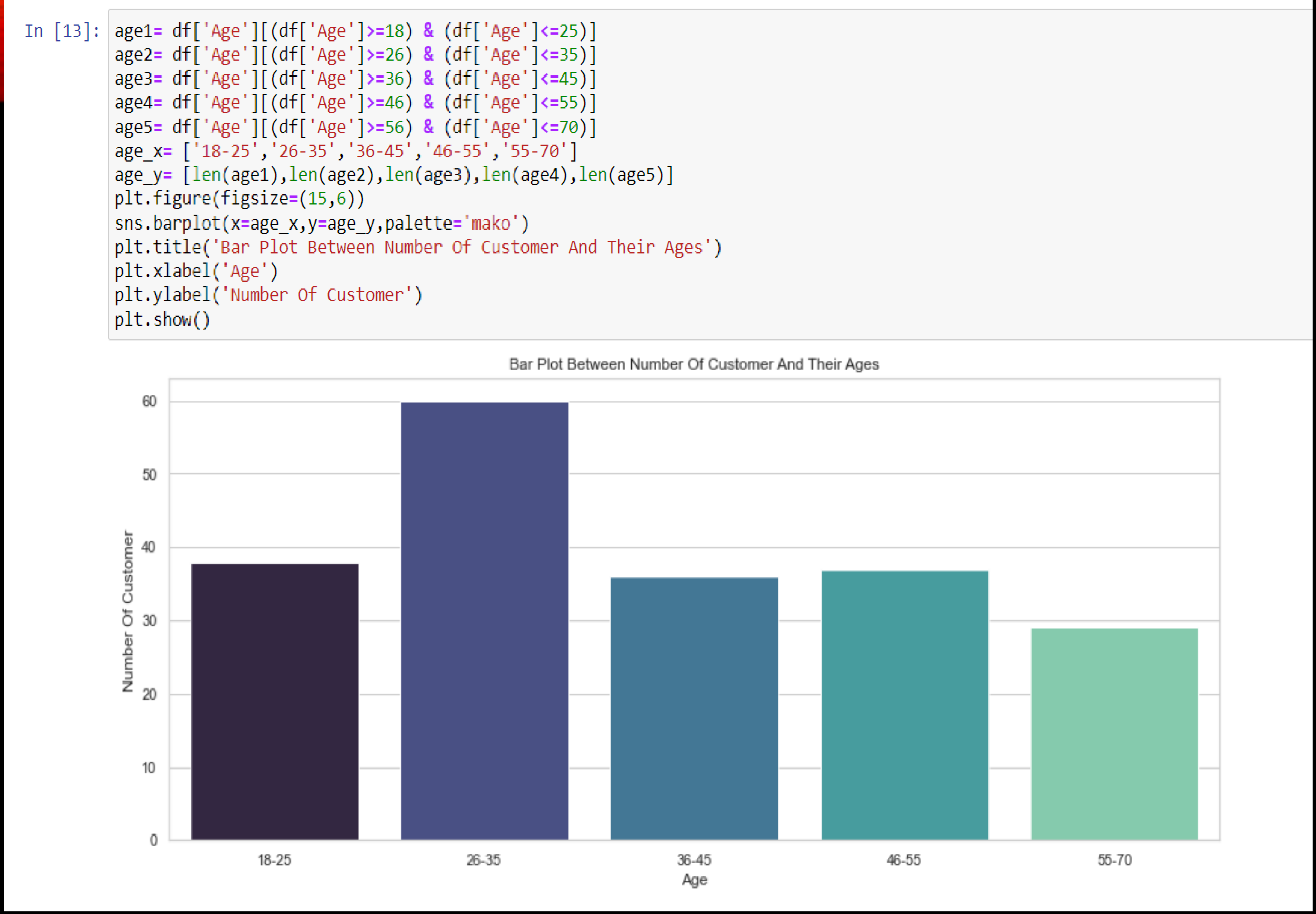
```
In [11]: sns.countplot(df.Gender)
         plt.xlabel('Gender')
         plt.ylabel('Count')
         plt.show()
```

- In terms of gender ratio we observe that females are more in the given dataset than the male.



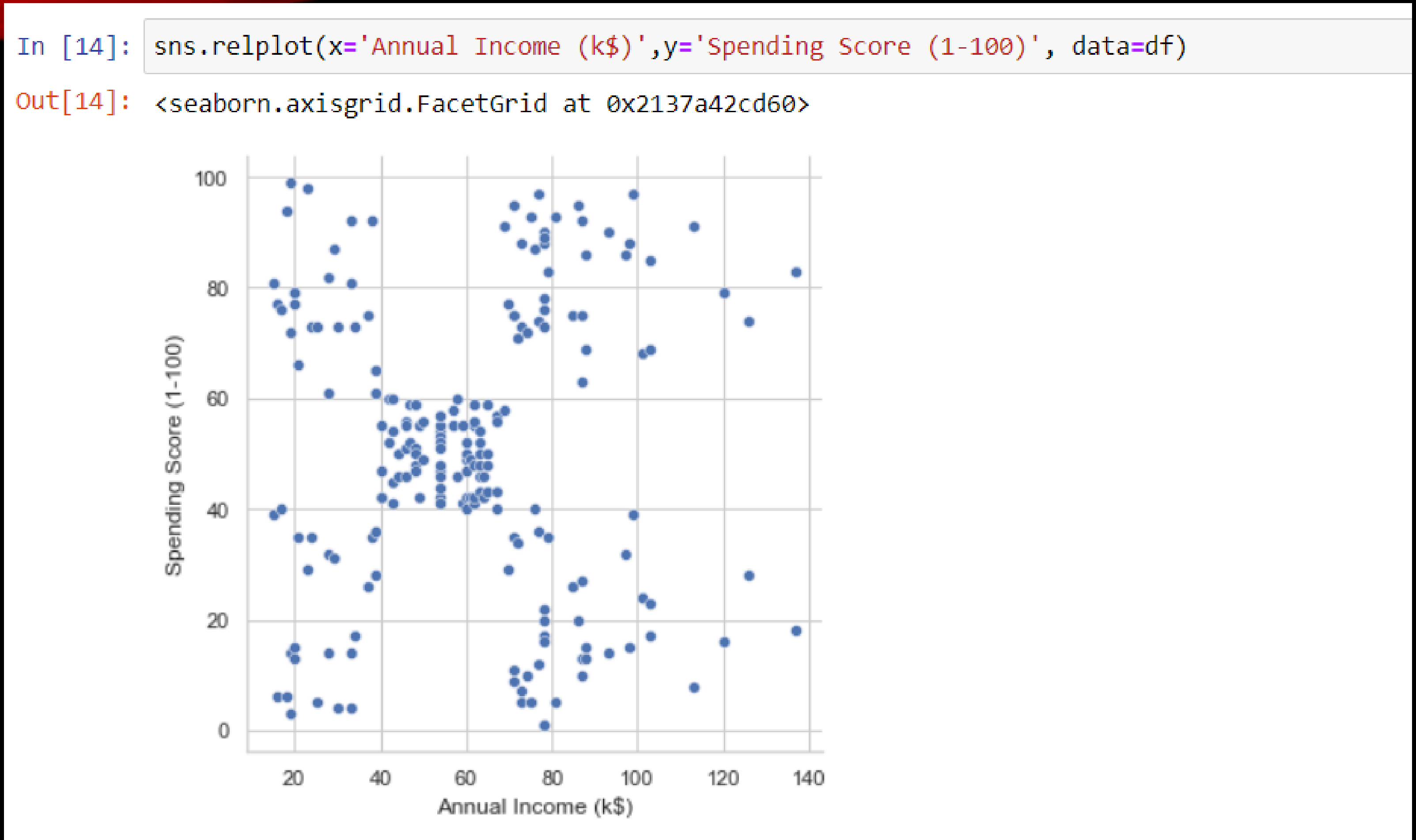
```
In [12]: plt.figure(1,figsize=(15,7))
         n=0
         for cols in ['Age','Annual Income (k$)','Spending Score (1-100)']:
             n+=1
             plt.subplot(1,3,n)
             sns.set(style='whitegrid')
             plt.subplots_adjust(hspace=0.5,wspace=0.5)
             sns.violinplot(x=cols,y='Gender',data= df)
             plt.ylabel('Gender' if n==1 else '')
             plt.title('Violin Plot')
         plt.show()
```

- For better visualization we are representing the Age, Annual Income(k$) and Spending Score(1-100) with respect to gender in a violin plot



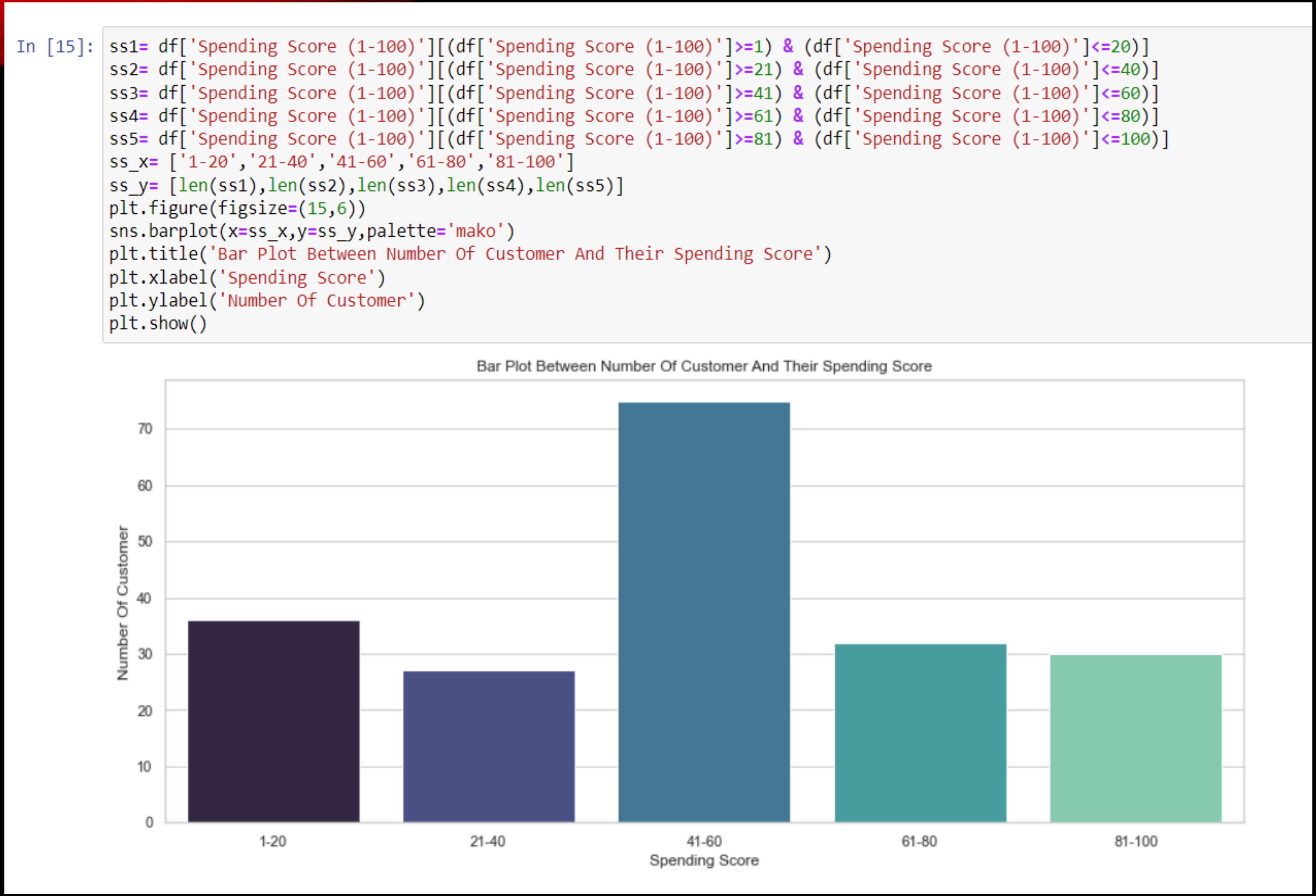
```
In [13]: age1= df['Age'][(df['Age']>=18) & (df['Age']<=25)]
         age2= df['Age'][(df['Age']>=26) & (df['Age']<=35)]
         age3= df['Age'][(df['Age']>=36) & (df['Age']<=45)]
         age4= df['Age'][(df['Age']>=46) & (df['Age']<=55)]
         age5= df['Age'][(df['Age']>=56) & (df['Age']<=70)]
         age_x= ['18-25','26-35','36-45','46-55','55-70']
         age_y= [len(age1),len(age2),len(age3),len(age4),len(age5)]
         plt.figure(figsize=(15,6))
         sns.barplot(x=age_x,y=age_y,palette='mako')
         plt.title('Bar Plot Between Number Of Customer And Their Ages')
         plt.xlabel('Age')
         plt.ylabel('Number Of Customer')
         plt.show()
```

- This bar plot show us the exact information as to how many potential customers are present in the dataset and what are their ages range. From the bar plot we observe that the maximum potential customers ages lies between 26-35.



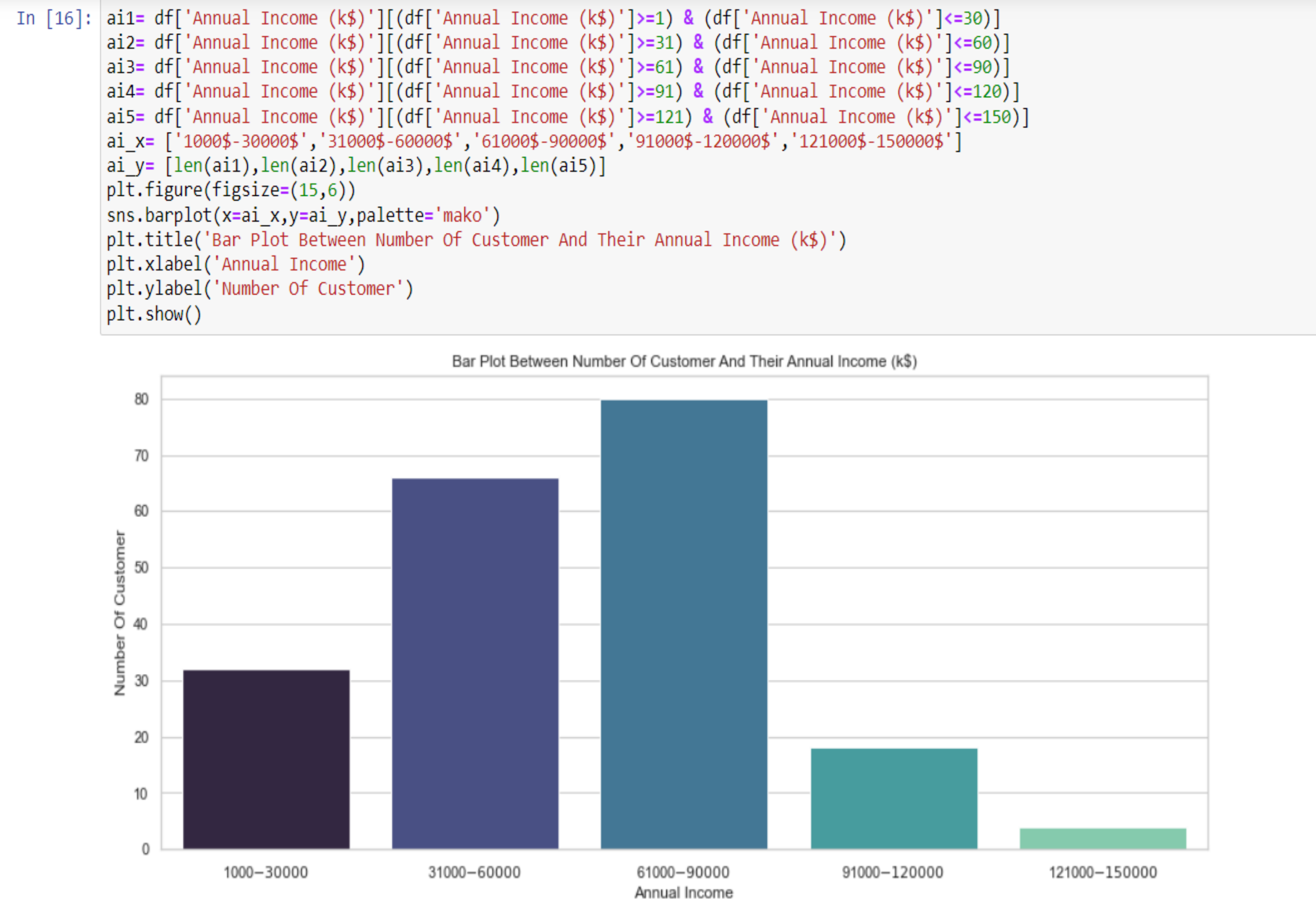
```
In [14]: sns.relplot(x='Annual Income (k$)',y='Spending Score (1-100)', data=df)

Out[14]: <seaborn.axisgrid.FacetGrid at 0x2137a42cd60>
```

- Here we are establishing the relationship between the Annual Income(k$) and the Spending Score(1-100).We observe that there is a relationship that exists between the customers whose annual income lies between 40k to 60k and the Spending Score between 40 to 60.



```
In [15]: ss1= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=1) & (df['Spending Score (1-100)']<=20)]
         ss2= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=21) & (df['Spending Score (1-100)']<=40)]
         ss3= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=41) & (df['Spending Score (1-100)']<=60)]
         ss4= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=61) & (df['Spending Score (1-100)']<=80)]
         ss5= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=81) & (df['Spending Score (1-100)']<=100)]
         ss_x= ['1-20','21-40','41-60','61-80','81-100']
         ss_y= [len(ss1),len(ss2),len(ss3),len(ss4),len(ss5)]
         plt.figure(figsize=(15,6))
         sns.barplot(x=ss_x,y=ss_y,palette='mako')
         plt.title('Bar Plot Between Number Of Customer And Their Spending Score')
         plt.xlabel('Spending Score')
         plt.ylabel('Number Of Customer')
         plt.show()
```

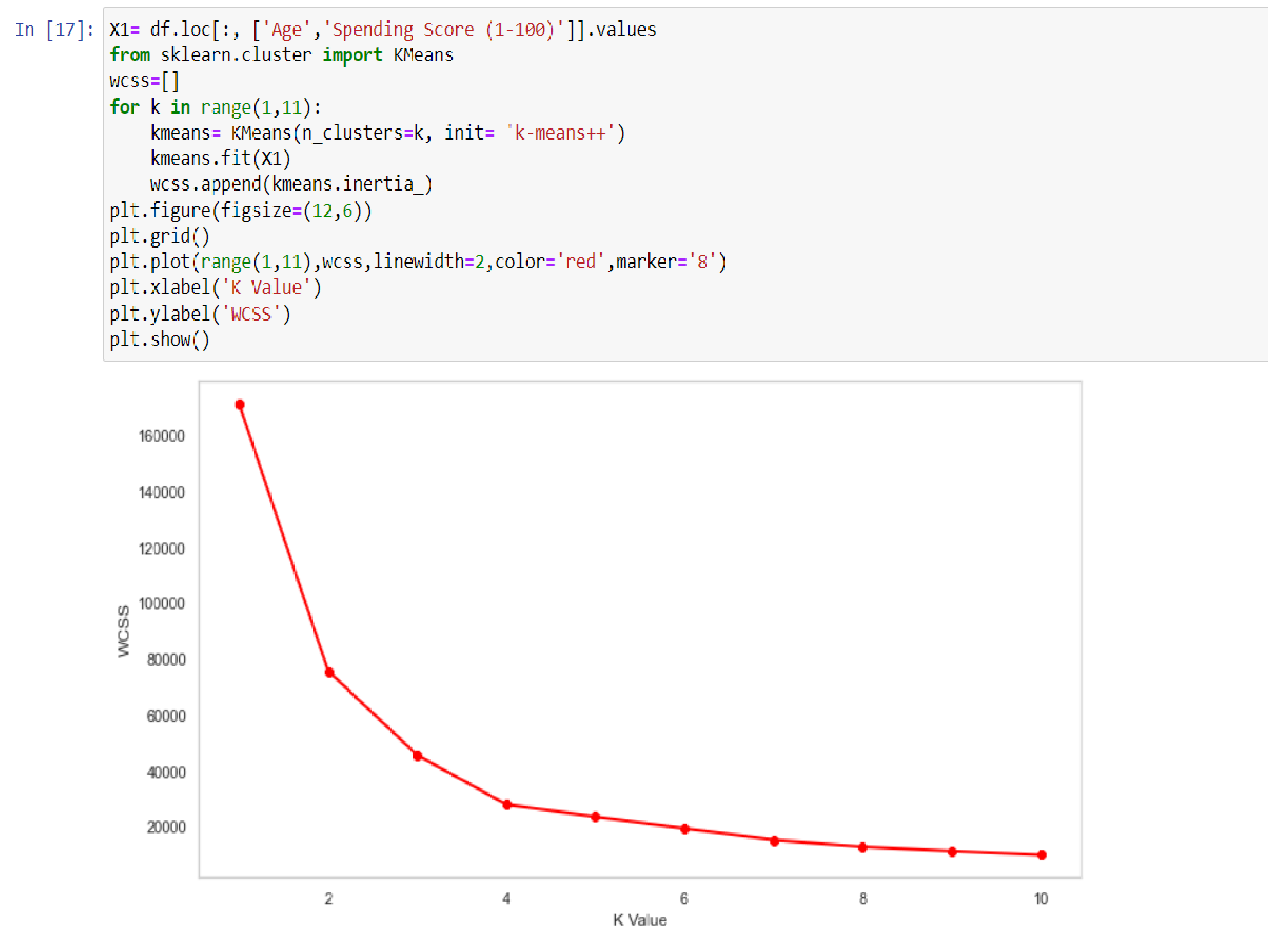
- From the bar plot between the number of customers and their spending score we observer that the maximum spending score lies between 41 to 60.



```
In [16]: ai1= df['Annual Income (k$)'][(df['Annual Income (k$)']>=1) & (df['Annual Income (k$)']<=30)]
         ai2= df['Annual Income (k$)'][(df['Annual Income (k$)']>=31) & (df['Annual Income (k$)']<=60)]
         ai3= df['Annual Income (k$)'][(df['Annual Income (k$)']>=61) & (df['Annual Income (k$)']<=90)]
         ai4= df['Annual Income (k$)'][(df['Annual Income (k$)']>=91) & (df['Annual Income (k$)']<=120)]
         ai5= df['Annual Income (k$)'][(df['Annual Income (k$)']>=121) & (df['Annual Income (k$)']<=150)]
         ai_x= ['1000$-30000$','31000$-60000$','61000$-90000$','91000$-120000$','121000$-150000$']
         ai_y= [len(ai1),len(ai2),len(ai3),len(ai4),len(ai5)]
         plt.figure(figsize=(15,6))
         sns.barplot(x=ai_x,y=ai_y,palette='mako')
         plt.title('Bar Plot Between Number Of Customer And Their Annual Income (k$)')
         plt.xlabel('Annual Income')
         plt.ylabel('Number Of Customer')
         plt.show()
```

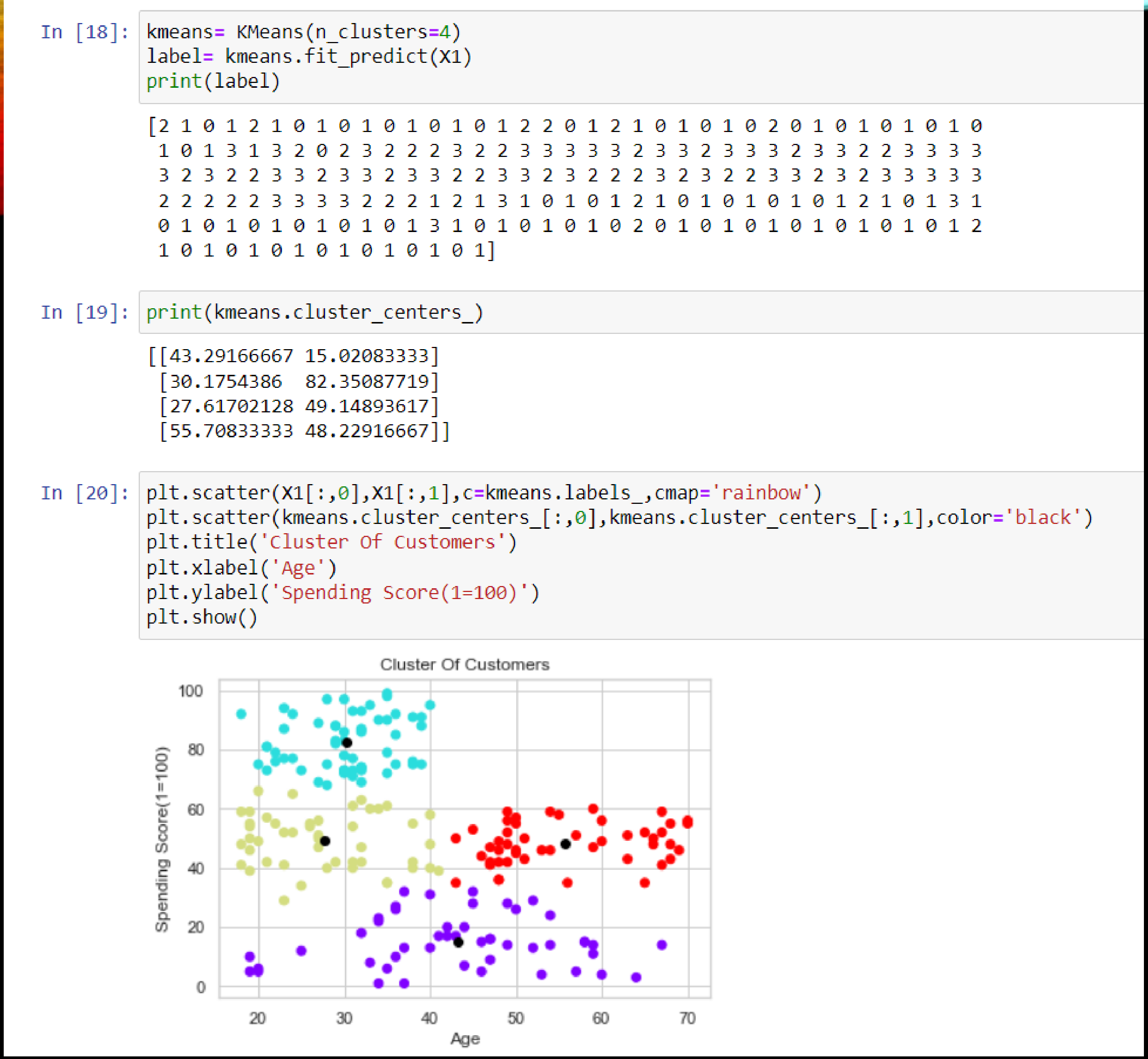
- The bar plot between Number of customer and annual income we visualize that most potential customers have their annual income in between 61000 to 90000.

RELATIONSHIP BETWEEN AGE AND SPENDING SCORE(1-100)



```
In [17]: X1= df.loc[:, ['Age','Spending Score (1-100)']].values
         from sklearn.cluster import KMeans
         wcss=[]
         for k in range(1,11):
             kmeans= KMeans(n_clusters=k, init= 'k-means++')
             kmeans.fit(X1)
             wcss.append(kmeans.inertia_)
         plt.figure(figsize=(12,6))
         plt.grid()
         plt.plot(range(1,11),wcss,linewidth=2,color='red',marker='8')
         plt.xlabel('K Value')
         plt.ylabel('WCSS')
         plt.show()
```

- Here we have used the elbow method to find as to how many clusters are required to perform clustering. From the figure we observe that after k value=4 our graph seems to be constant thus we are creating only 4 clusters as after k value=4 the graph seems pretty constant.



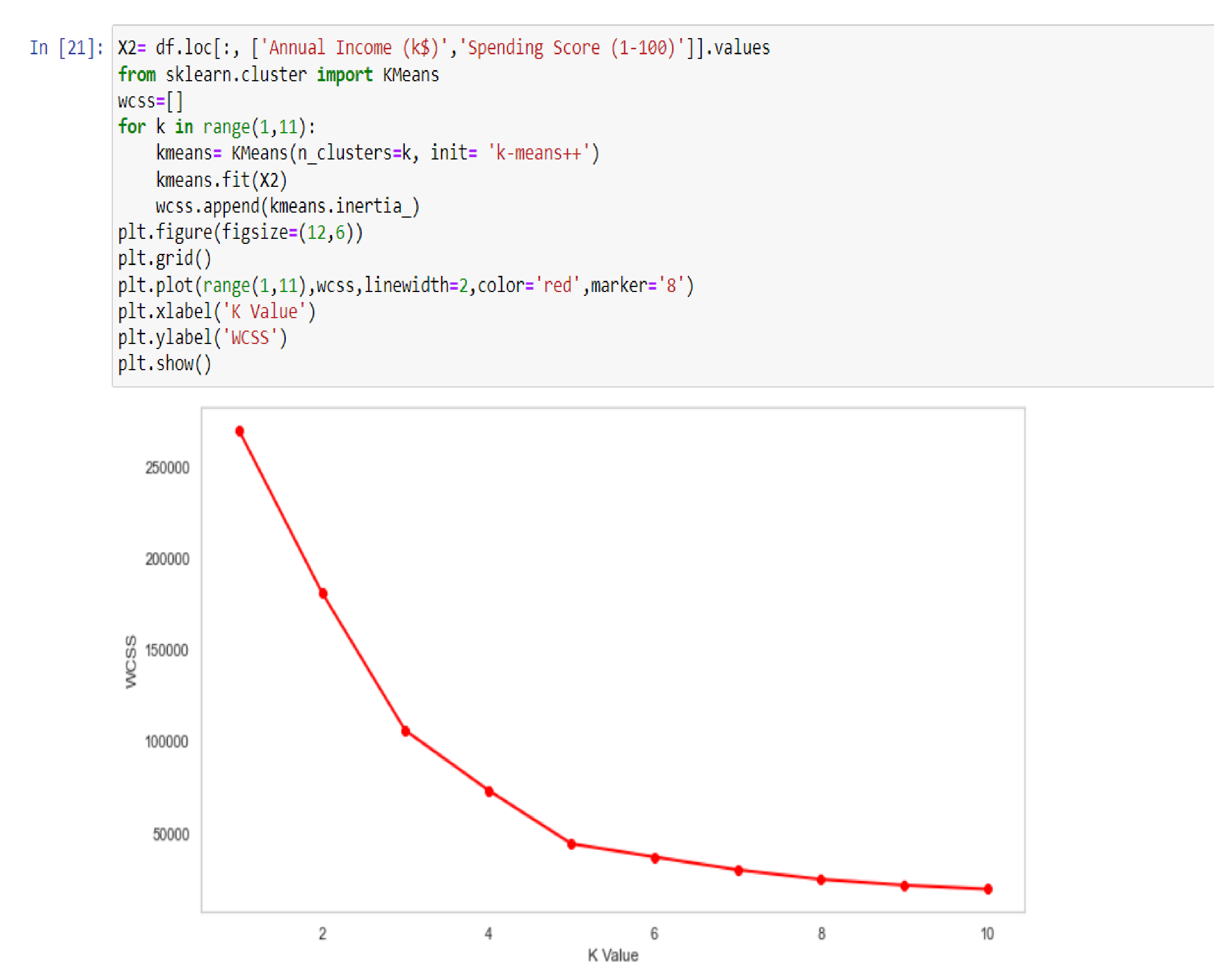
```
In [18]: kmeans= KMeans(n_clusters=4)
         label= kmeans.fit_predict(X1)
         print(label)

[2 1 0 1 2 1 0 1 0 1 0 1 0 1 0 1 2 2 0 1 2 1 0 1 0 1 0 2 0 1 0 1 0 1 0 1 0
 1 0 1 3 1 3 2 0 2 3 2 2 2 3 2 2 3 3 3 3 3 2 3 3 2 3 3 3 2 3 3 2 2 3 3 3 3
 3 2 3 2 2 3 3 2 3 3 2 3 3 2 2 3 3 2 3 2 2 2 3 2 3 2 2 3 3 2 3 2 3 3 3 3 3
 2 2 2 2 2 3 3 3 3 2 2 2 1 2 1 3 1 0 1 0 1 2 1 0 1 0 1 0 1 0 1 2 1 0 1 3 1
 0 1 0 1 0 1 0 1 0 1 0 1 3 1 0 1 0 1 0 1 0 2 0 1 0 1 0 1 0 1 0 1 0 1 0 1 2
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
```

```
In [19]: print(kmeans.cluster_centers_)

[[43.29166667 15.02083333]
 [30.1754386  82.35087719]
 [27.61702128 49.14893617]
 [55.70833333 48.22916667]]
```

```
In [20]: plt.scatter(X1[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
         plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
         plt.title('Cluster Of Customers')
         plt.xlabel('Age')
         plt.ylabel('Spending Score(1=100)')
         plt.show()
```

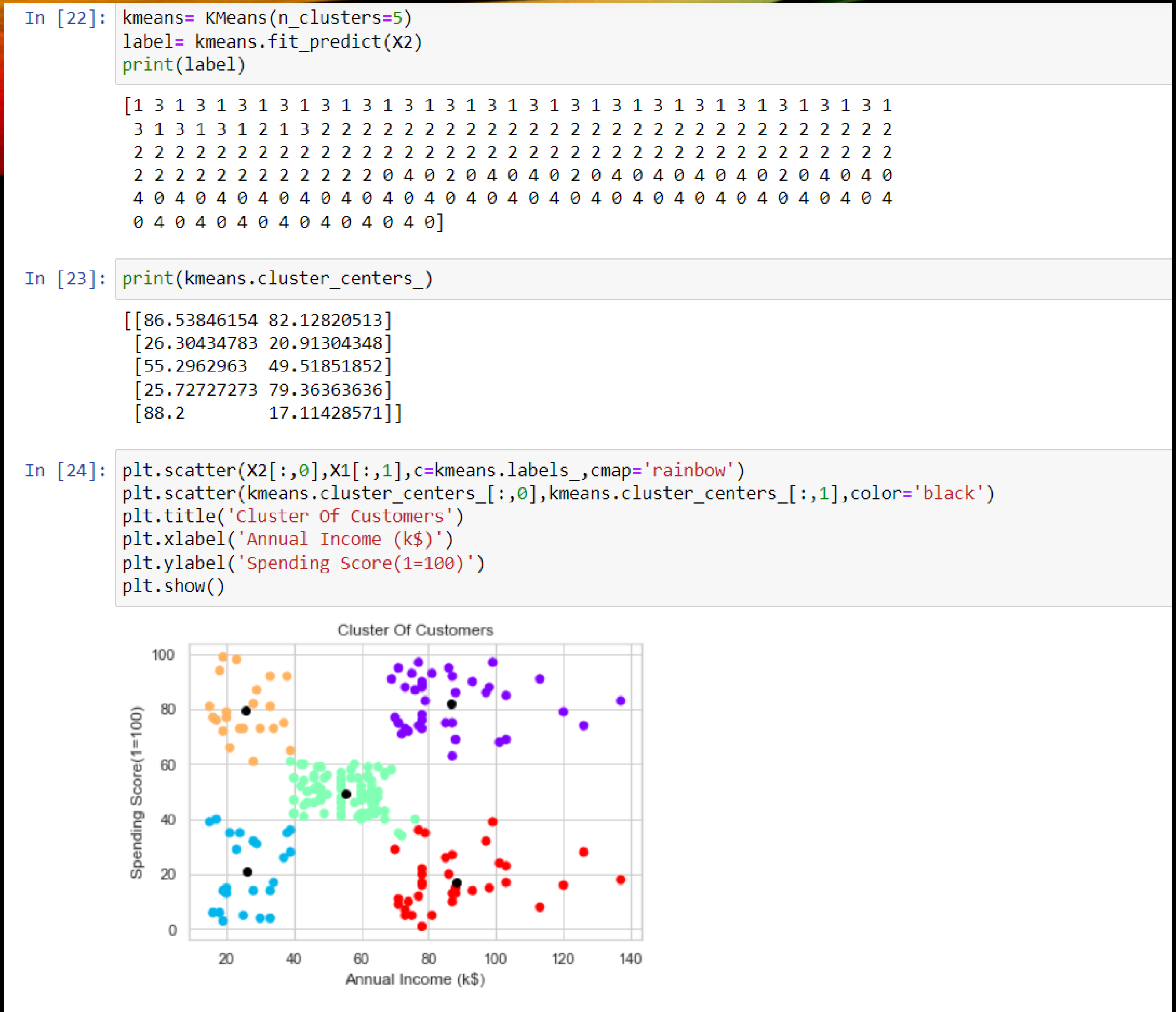
- In the 1st figure our data is divided into 4 clusters (0,1,2,3)
- In the 2nd figure we are finding out the centroid.
- In the 3rd figure we are drawing a scatter plot between spending score(1-100) and age. The point in black color represent the centroid.

RELATIONSHIP BETWEEN ANNUAL INCOME(K$) AND SPENDING SCORE(1-100)



```
In [21]: X2= df.loc[:, ['Annual Income (k$)','Spending Score (1-100)']].values
         from sklearn.cluster import KMeans
         wcss=[]
         for k in range(1,11):
             kmeans= KMeans(n_clusters=k, init= 'k-means++')
             kmeans.fit(X2)
             wcss.append(kmeans.inertia_)
         plt.figure(figsize=(12,6))
         plt.grid()
         plt.plot(range(1,11),wcss,linewidth=2,color='red',marker='8')
         plt.xlabel('K Value')
         plt.ylabel('WCSS')
         plt.show()
```

- Between Annual Income and Spending Score through the Elbow Method the number of cluster that can be made is 5 as after k value=5 the graph seems pretty constant.



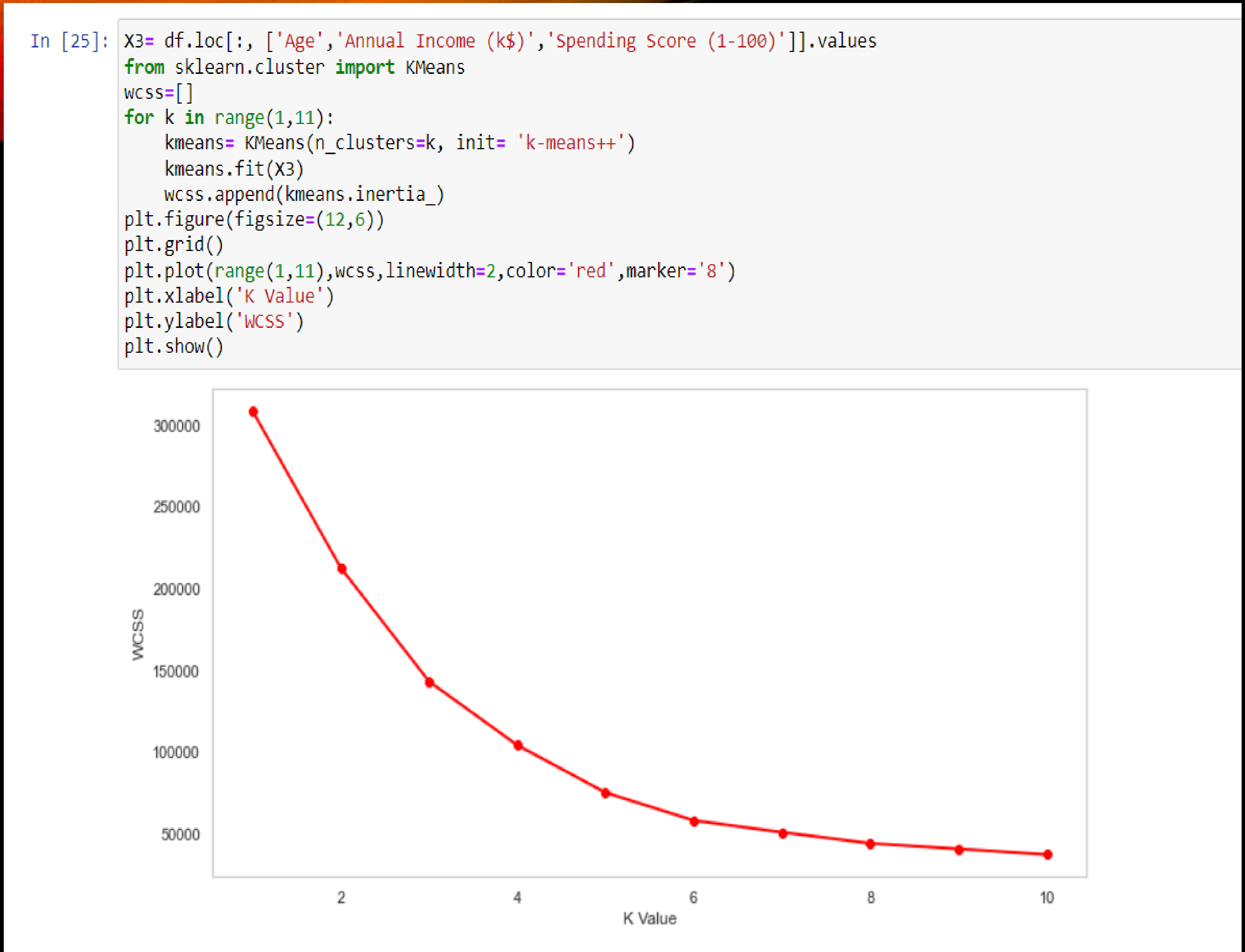
```
In [22]: kmeans= KMeans(n_clusters=5)
         label= kmeans.fit_predict(X2)
         print(label)

[1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
 3 1 3 1 3 1 2 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 0 4 0 2 0 4 0 4 0 2 0 4 0 4 0 4 0 4 0 2 0 4 0 4 0
 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4
 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0]
```

```
In [23]: print(kmeans.cluster_centers_)

[[86.53846154 82.12820513]
 [26.30434783 20.91304348]
 [55.2962963  49.51851852]
 [25.72727273 79.36363636]
 [88.2        17.11428571]]
```

```
In [24]: plt.scatter(X2[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
         plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
         plt.title('Cluster Of Customers')
         plt.xlabel('Annual Income (k$)')
         plt.ylabel('Spending Score(1=100)')
         plt.show()
```

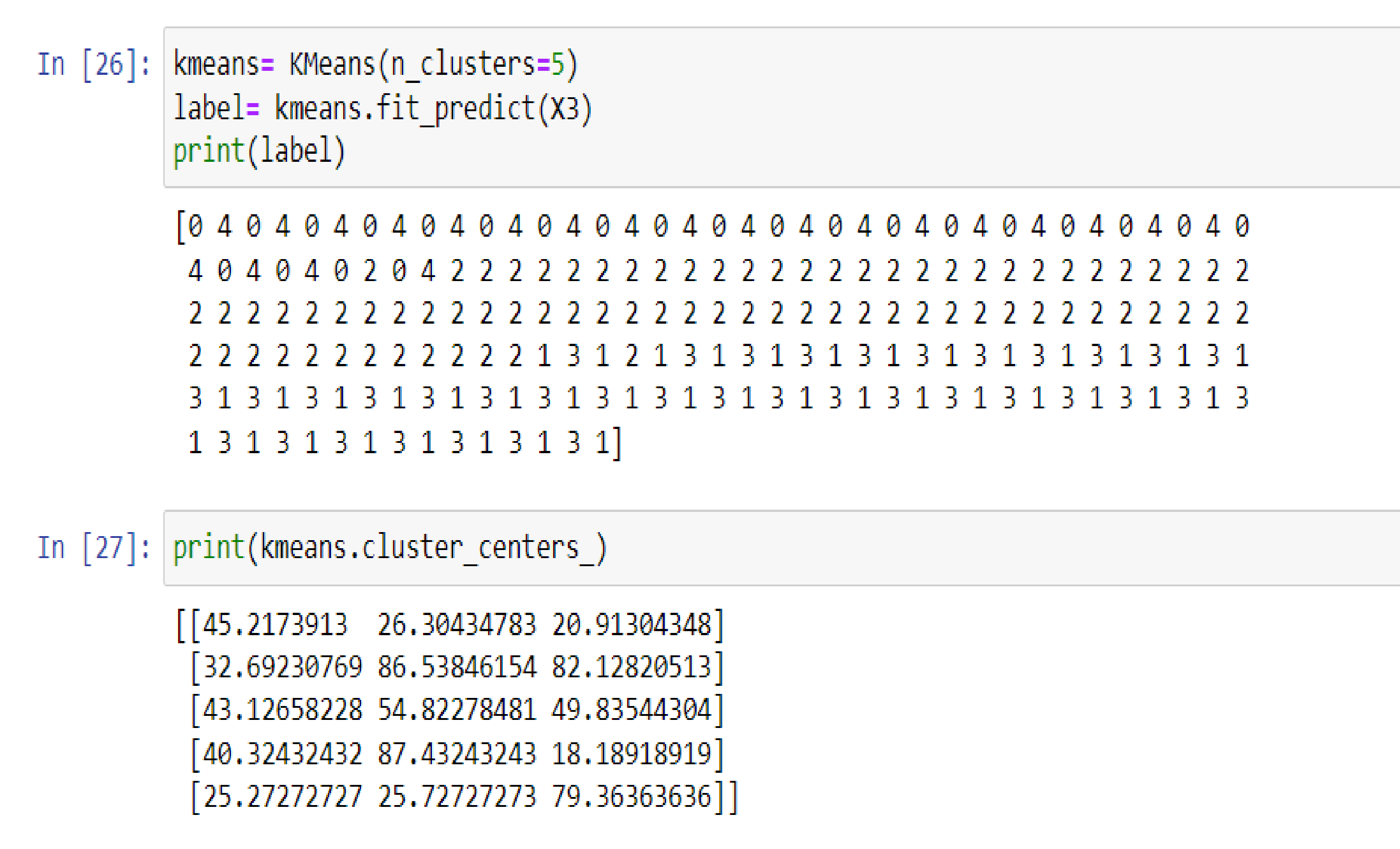
- In the 1st figure our data is divided into 5 clusters (0,1,2,3,4)
- In the 2nd figure we are finding out the centroid.
- In the 3rd figure we are drawing a scatter plot between spending score(1-100) and annual income(k$). The point in black color represent the centroid.

RELATIONSHIP BETWEEN AGE, ANNUAL INCOME(K$) AND SPENDING SCORE(1-100)



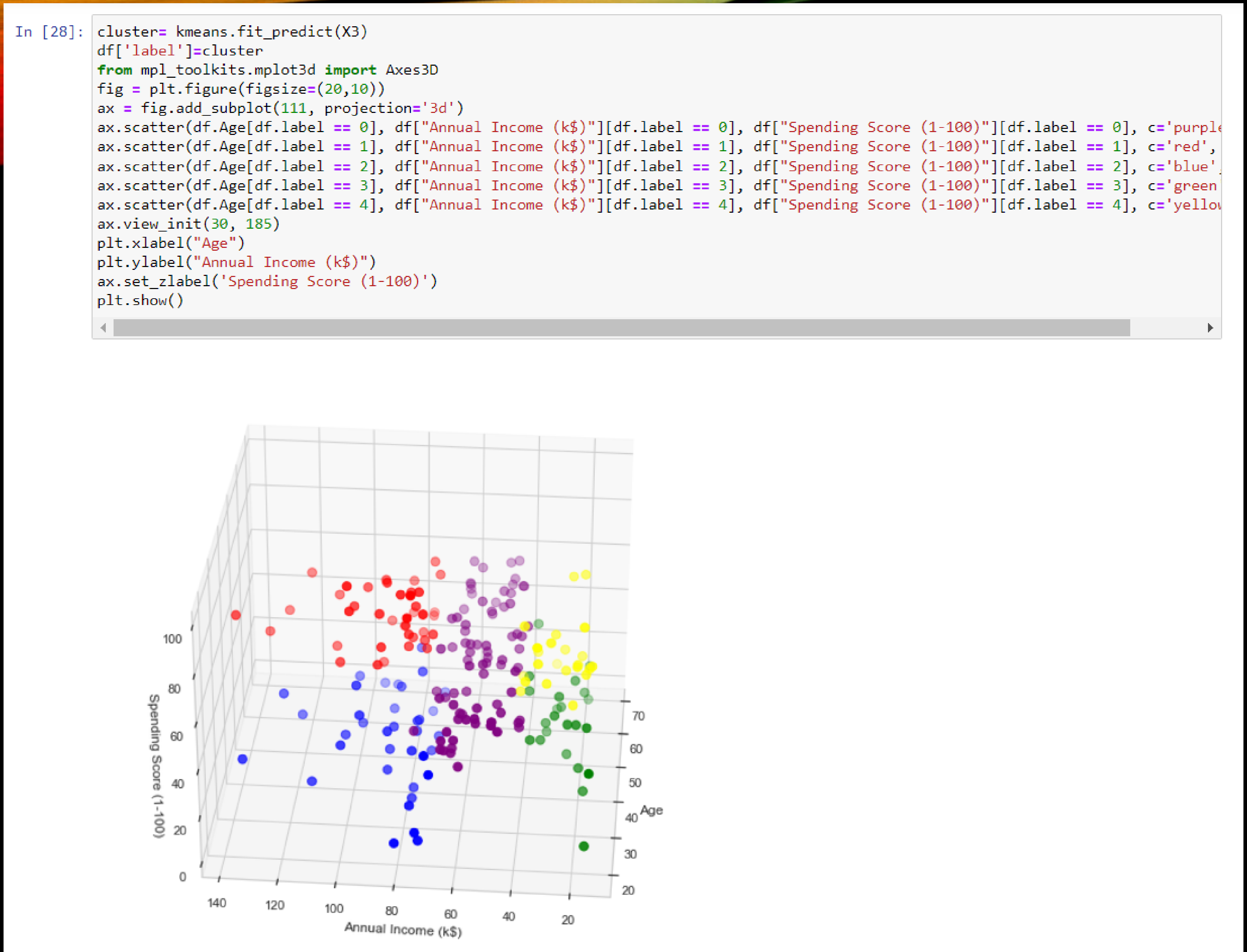
```
In [25]: X3= df.loc[:, ['Age','Annual Income (k$)','Spending Score (1-100)']].values
         from sklearn.cluster import KMeans
         wcss=[]
         for k in range(1,11):
             kmeans= KMeans(n_clusters=k, init= 'k-means++')
             kmeans.fit(X3)
             wcss.append(kmeans.inertia_)
         plt.figure(figsize=(12,6))
         plt.grid()
         plt.plot(range(1,11),wcss,linewidth=2,color='red',marker='8')
         plt.xlabel('K Value')
         plt.ylabel('WCSS')
         plt.show()
```

- Between Age, Annual Income and Spending Score through the Elbow Method the number of cluster that can be made is 5 as after k value=5 the graph seems pretty constant.



```
In [26]: kmeans= KMeans(n_clusters=5)
         label= kmeans.fit_predict(X3)
         print(label)

[0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0 4 0
 4 0 4 0 4 0 2 0 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 1 3 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1
 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3
 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1]

In [27]: print(kmeans.cluster_centers_)

[[45.2173913  26.30434783 20.91304348]
 [32.69230769 86.53846154 82.12820513]
 [43.12658228 54.82278481 49.83544304]
 [40.32432432 87.43243243 18.18918919]
 [25.27272727 25.72727273 79.36363636]]
```

- In the 1st figure our data is divided into 5 clusters (0,1,2,3,4)
- In the 2nd figure we are finding out the centroid.



```
In [28]: cluster= kmeans.fit_predict(X3)
         df['label']=cluster
         from mpl_toolkits.mplot3d import Axes3D
         fig = plt.figure(figsize=(20,10))
         ax = fig.add_subplot(111, projection='3d')
         ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending Score (1-100)"][df.label == 0], c='purple
         ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending Score (1-100)"][df.label == 1], c='red',
         ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending Score (1-100)"][df.label == 2], c='blue',
         ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending Score (1-100)"][df.label == 3], c='green
         ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending Score (1-100)"][df.label == 4], c='yello
         ax.view_init(30, 185)
         plt.xlabel("Age")
         plt.ylabel("Annual Income (k$)")
         ax.set_zlabel('Spending Score (1-100)')
         plt.show()
```

- It’s a 3d representation of age, annual income(k$) and spending score(1-100).

CONCLUSION

- The Highest income , high spending can be target these type of customers as they earn more money and spend as much as they want.
- Highest income, low spending can be target these type of customers by asking feedback and advertising the product in a better way.
- Average income, Average spending may or may not be beneficial to the mall owners of this type of customers.
- Low income, High spending can be target these type of customers by providing them with low-cost EMI's etc.
- Low income, Low spending don't target these type of customers because they earn a bit and spend some amount of money.