Mall_Customers_K-Means_Clustering

December 7, 2022

0.0.1 Reference: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Dataset : https://www.kaggle.com/code/yudhaykw/market-segmentation-using-k-means/data

[1]:
```
#Importing the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D

%matplotlib inline
```

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

[3]:
```
data.head()
```

[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

Exploring the Dataset

[4]:
```
data.describe()
```

[4]:

	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

The data has 200 entries, that is data from 200 customers

[5]:
```
null_values = data.isnull().sum()
```

[6]:
```
null_values
```

[6]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[7]:
```
corr = data.corr()
corr
```

[7]:
```
                        CustomerID       Age  Annual Income (k$)  \
CustomerID                1.000000 -0.026763            0.977548
Age                      -0.026763  1.000000           -0.012398
Annual Income (k$)        0.977548 -0.012398            1.000000
Spending Score (1-100)    0.013835 -0.327227            0.009903

                        Spending Score (1-100)
CustomerID                            0.013835
Age                                  -0.327227
Annual Income (k$)                    0.009903
Spending Score (1-100)                1.000000
```

Heatmap & Exploratory Data Analysis

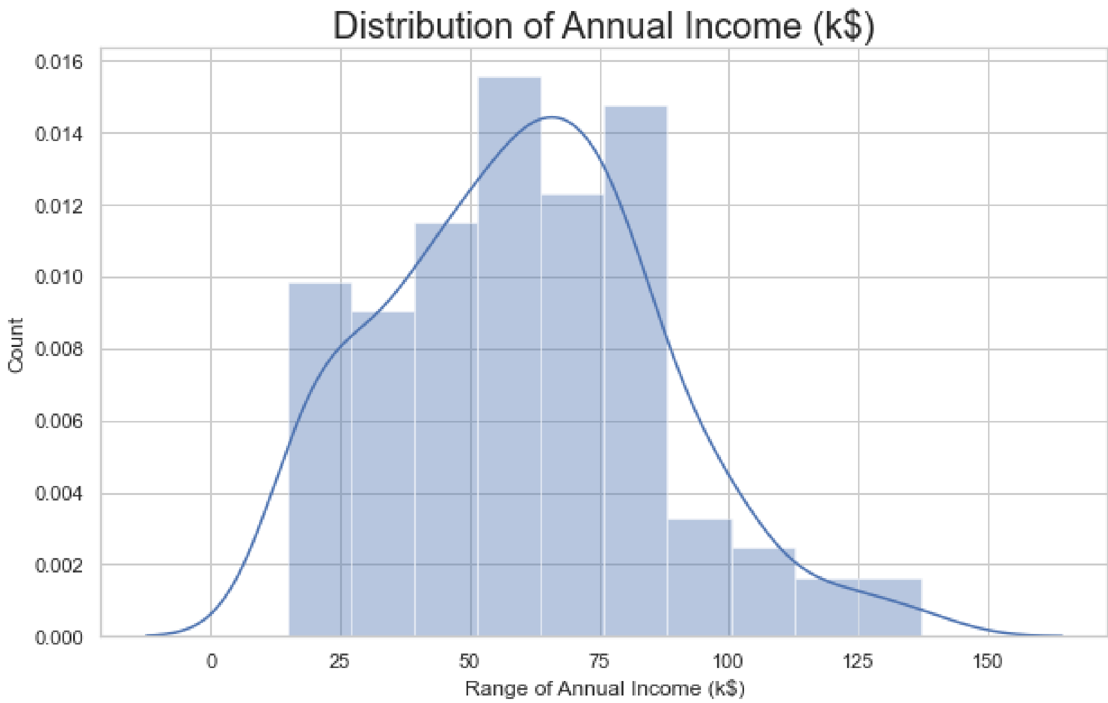
[8]:
```
plt.figure(figsize=(10, 6))
heatmap = sns.heatmap(corr)
heatmap
```

[8]:
```
<AxesSubplot:>
```

[9]:
```
#Distribution of Annnual Income
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(data['Annual Income (k$)'])
plt.title('Distribution of Annual Income (k$)', fontsize = 20)
plt.xlabel('Range of Annual Income (k$)')
plt.ylabel('Count')
```

```
C:\Users\Shobhandeb\AppData\Local\Programs\Python\Python310\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

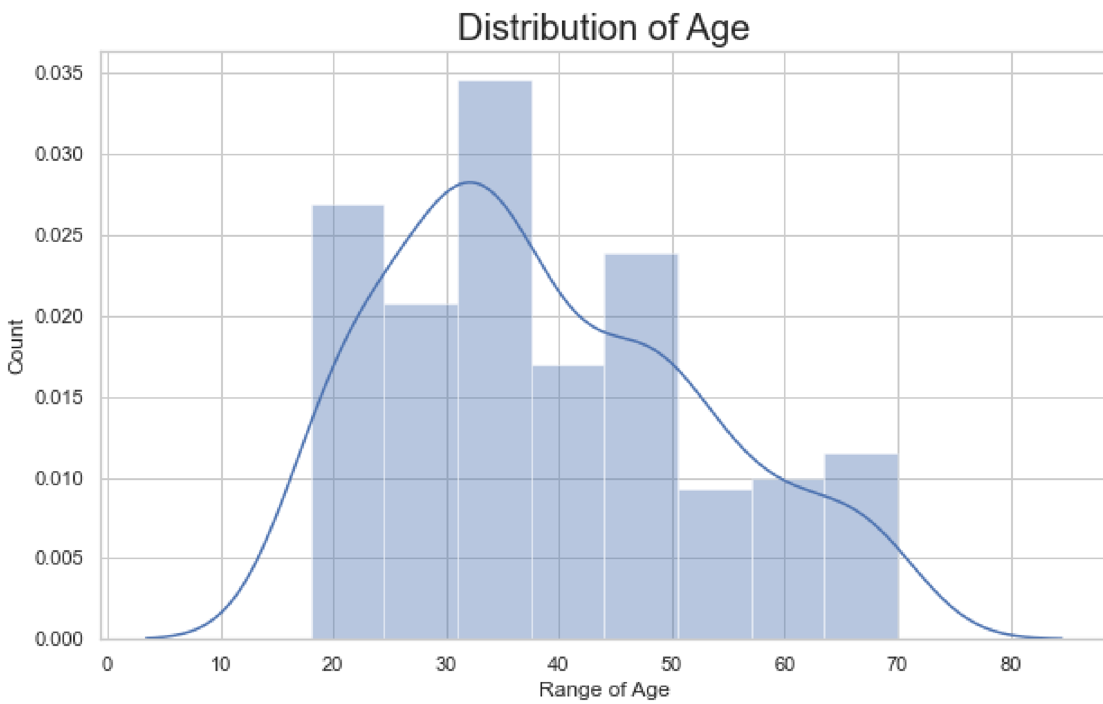
[9]:
```
Text(0, 0.5, 'Count')
```



[10]:
```
#Distribution of age
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(data['Age'])
plt.title('Distribution of Age', fontsize = 20)
plt.xlabel('Range of Age')
plt.ylabel('Count')
```

```
C:\Users\Shobhandeb\AppData\Local\Programs\Python\Python310\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[10]: Text(0, 0.5, 'Count')

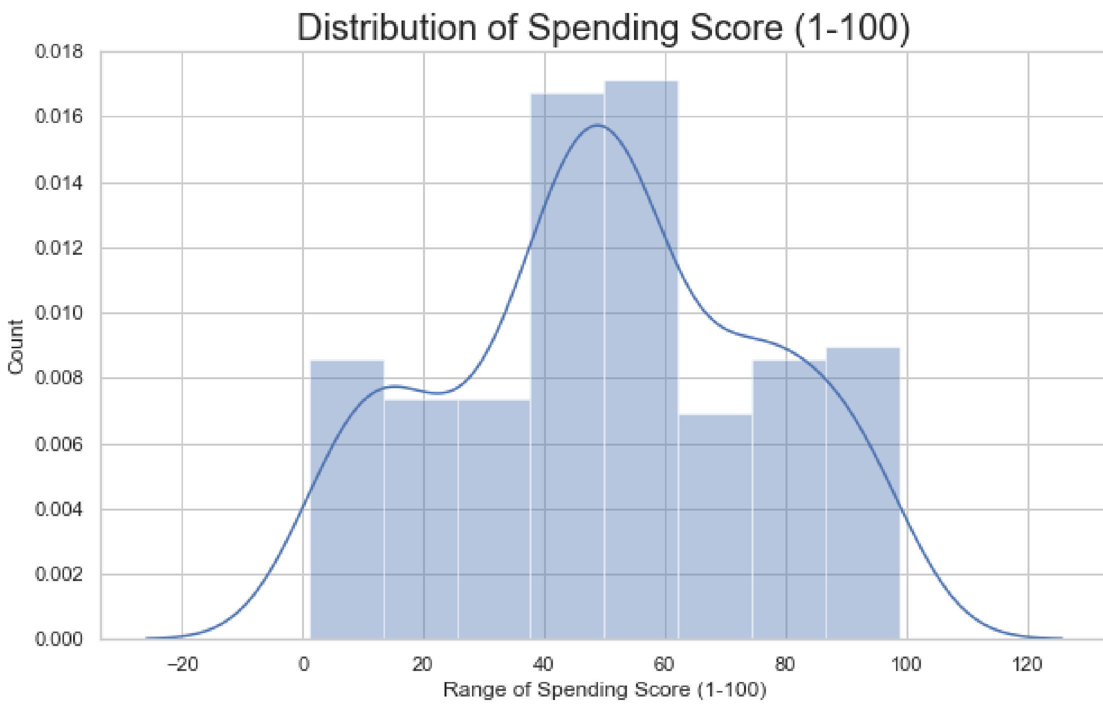


Mainly Annual Income falls between 50K to 85K.

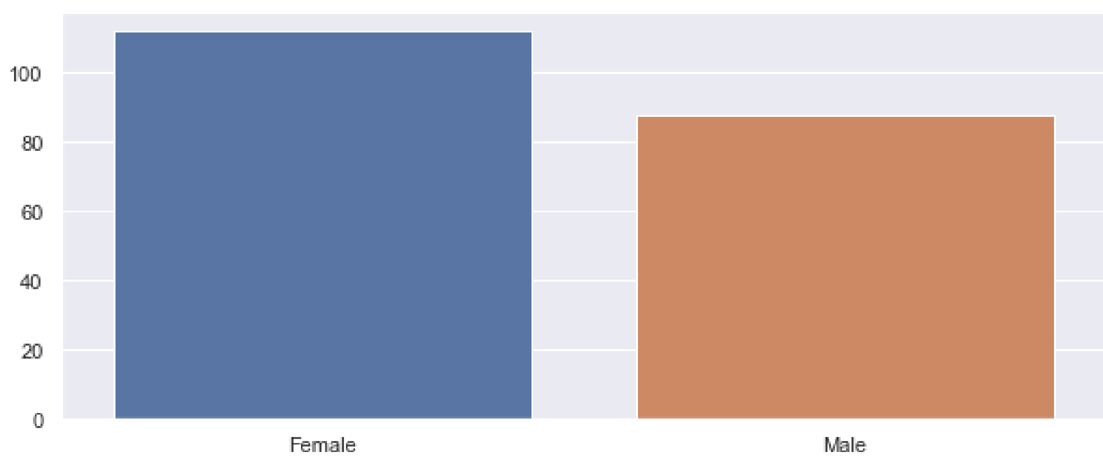
[11]:
```
#Distribution of spending score
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(data['Spending Score (1-100)'])
plt.title('Distribution of Spending Score (1-100)', fontsize = 20)
plt.xlabel('Range of Spending Score (1-100)')
plt.ylabel('Count')
```

```
C:\Users\Shobhandeb\AppData\Local\Programs\Python\Python310\lib\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[11]: Text(0, 0.5, 'Count')



[12]:
```
genders = data.Gender.value_counts()
sns.set_style("darkgrid")
plt.figure(figsize=(10,4))
sns.barplot(x=genders.index, y=genders.values)
plt.show()
```



[13]:
```
#We take just the Annual Income and Spending score
X=data[["Annual Income (k$)","Spending Score (1-100)"]]
```

[14]:
```
X.head()
```

[14]:

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

[15]:
```
#Scatterplot of the input data
plt.figure(figsize=(10,6))
sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',  data =
 ↪X  ,s = 60 )
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```

[16]:
```
#Importing KMeans from sklearn
from sklearn.cluster import KMeans
```

[17]:
```
kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
y = kmeans.labels_
```

[18]:
```
y
```

[18]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0])
```

[19]:
```
#adding the labels to a column named label
data["label"] = y
```

[20]:
```
data.head()
```

[20]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	1	Male	19	15	39	1
1	2	Male	21	15	81	1
2	3	Female	20	16	6	1
3	4	Female	23	16	77	1
4	5	Female	31	17	40	1

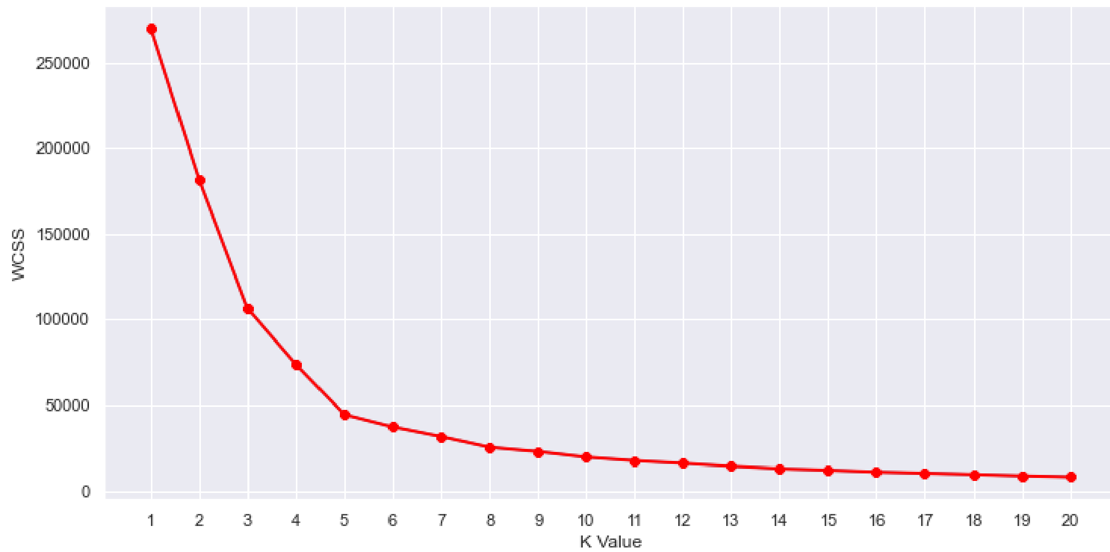
Scatter Plot with Two Clusters

[21]:
```
plt.figure(figsize=(10,6))
sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score␣
↪(1-100)',hue="label",
                 palette=['green','orange'], legend='full',data = data  ,s = 60␣
↪)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```

WCSS to plot the Elbow Curve in order to find the Right value for K (Clusters) Now we calculate the Within Cluster Sum of Squared Errors (WSS) for different values of k. Next, we choose the k for which WSS first starts to diminish. This value of K gives us the best number of clusters to make from the raw data.

[22]:
```
wcss=[]
for i in range(1,21):
    km=KMeans(n_clusters=i)
    km.fit(X)
    wcss.append(km.inertia_)
```

[23]:
```
#The elbow curve
plt.figure(figsize=(12,6))
plt.plot(range(1,21),wcss)
plt.plot(range(1,21),wcss, linewidth=2, color="red", marker ="8")
plt.xlabel("K Value")
plt.xticks(np.arange(1,21,1))
plt.ylabel("WCSS")
plt.show()
```



K-Means with Model Training with 5 Clusters

[24]:
```
#Taking 5 clusters
kmeans_wcss=KMeans(n_clusters=5)
kmeans_wcss.fit(X)
y=kmeans_wcss.predict(X)
data["label"] = y
data.head()
```

[24]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	1	Male	19	15	39	3
1	2	Male	21	15	81	2
2	3	Female	20	16	6	3
3	4	Female	23	16	77	2
4	5	Female	31	17	40	3

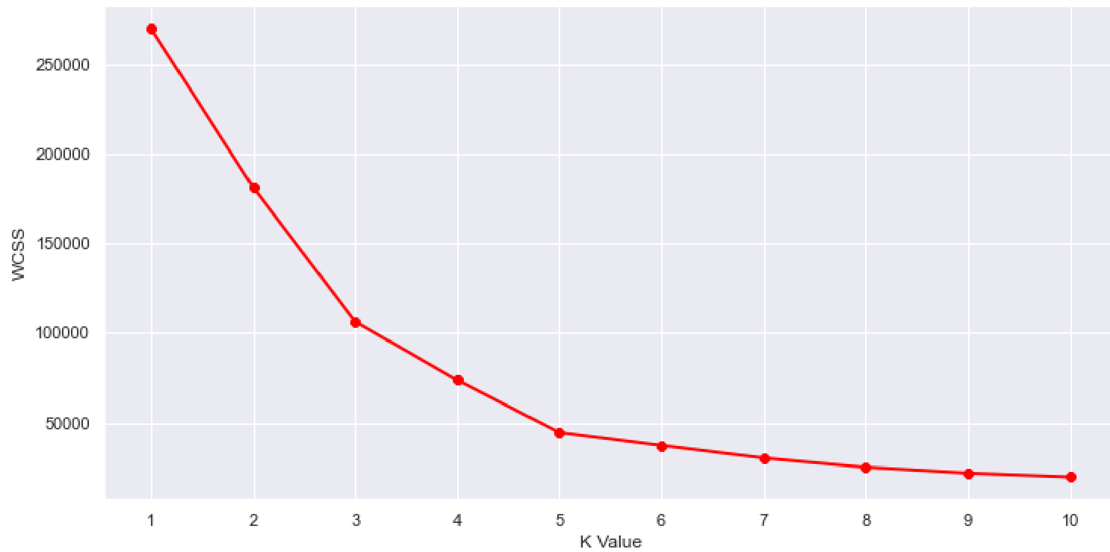
Scatter Plot with 5 Clusters (ref. to Elbow Curve value)

[25]:
```
#Scatterplot of the clusters
plt.figure(figsize=(10,6))
sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score␣
↪(1-100)',hue="label",
                 palette=['green','orange','brown','dodgerblue','red'],␣
↪legend='full',data = data  ,s = 60 )
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```

Using K-Means++ for Elbow Curve & finding out the difference

[26]:
```
#Taking the features
X1=data[["Age","Annual Income (k$)","Spending Score (1-100)"]]
#Now we calculate the Within Cluster Sum of Squared Errors (WSS) for different␣
↪values of k.
wcss = []

for k in range(1,11):
    kmeans = KMeans(n_clusters=k, init="k-means++")
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")
plt.xlabel("K Value")
plt.xticks(np.arange(1,11,1))
plt.ylabel("WCSS")
plt.show()
```



This is known as the elbow graph, the x-axis being the number of clusters, the number of clusters is taken at the elbow joint point. This point is the point where making clusters is most relevant as here the value of WCSS suddenly stops decreasing. Here in the graph, after 5 the drop is minimal, so we take 5 to be the number of clusters.

Here also we choose % clusters as per the Elbow Curve

[27]:
```
#We choose the k for which WSS starts to diminish
kmeans2 = KMeans(n_clusters=5)
y2 = kmeans2.fit_predict(X1)
data["label"] = y2
#The data with labels
data.head()
```

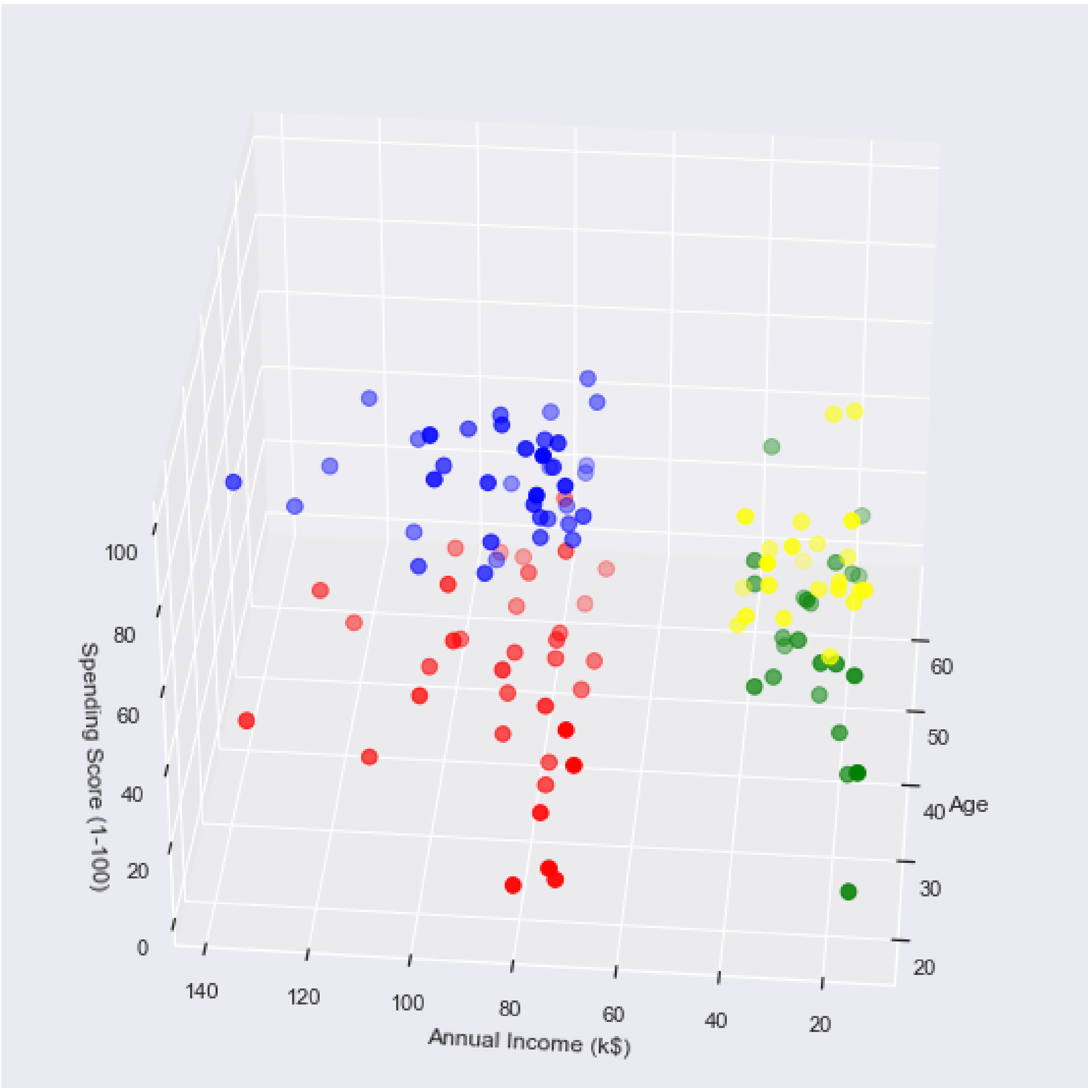
[27]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	1	Male	19	15	39	3
1	2	Male	21	15	81	4
2	3	Female	20	16	6	3
3	4	Female	23	16	77	4
4	5	Female	31	17	40	3

3D Plot for the selected 3 features

[28]:
```
#3D Plot as we did the clustering on the basis of 3 input features
fig = plt.figure(figsize=(10,15))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(data.Age[data.label == 1], data["Annual Income (k$)"][data.label ==␣
 ↪1], data["Spending Score (1-100)"][data.label == 1], c='red', s=60)
ax.scatter(data.Age[data.label == 2], data["Annual Income (k$)"][data.label ==␣
 ↪2], data["Spending Score (1-100)"][data.label == 2], c='blue', s=60)
```

```
ax.scatter(data.Age[data.label == 3], data["Annual Income (k$)"][data.label ==␣
    ↪3], data["Spending Score (1-100)"][data.label == 3], c='green', s=60)
ax.scatter(data.Age[data.label == 4], data["Annual Income (k$)"][data.label ==␣
    ↪4], data["Spending Score (1-100)"][data.label == 4], c='yellow', s=60)
ax.view_init(35, 185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```



Number of customers from each cluster

[29]:

```
cust1=data[data["label"]==1]
print('Number of customer in 1st group=', len(cust1))
print('They are -', cust1["CustomerID"].values)
print("---------------------------------------------")
cust2=data[data["label"]==2]
print('Number of customer in 2nd group=', len(cust2))
print('They are -', cust2["CustomerID"].values)
print("---------------------------------------------")
cust3=data[data["label"]==0]
print('Number of customer in 3rd group=', len(cust3))
print('They are -', cust3["CustomerID"].values)
print("---------------------------------------------")
cust4=data[data["label"]==3]
print('Number of customer in 4th group=', len(cust4))
print('They are -', cust4["CustomerID"].values)
print("---------------------------------------------")
cust5=data[data["label"]==4]
print('Number of customer in 5th group=', len(cust5))
print('They are -', cust5["CustomerID"].values)
print("---------------------------------------------")
```

```
Number of customer in 1st group= 36
They are - [125 129 131 133 135 137 139 141 145 147 149 151 153 155 157 159 161
163
 165 167 169 171 173 175 177 179 181 183 185 187 189 191 193 195 197 199]
---------------------------------------------
Number of customer in 2nd group= 39
They are - [124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156
158
 160 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194
 196 198 200]
---------------------------------------------
Number of customer in 3rd group= 79
They are - [ 47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
64
  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82
  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
 119 120 121 122 123 127 143]
---------------------------------------------
Number of customer in 4th group= 23
They are - [ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43
45]
---------------------------------------------
Number of customer in 5th group= 23
They are - [ 2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44
46]
---------------------------------------------
```

[ ]:

```

```