KMEANS

UNSUPERVISED MACHINE LEARING ALGORITHM

Import Modules

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import warnings
%matplotlib inline
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
from mpl_toolkits.mplot3d import Axes3D
from sklearn.metrics import silhouette_score, calinski_harabasz_score
```

Load the Dataset

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

Out[2]:

	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

In [3]:
```
# statistical info
df.describe()
```

Out[3]:

	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

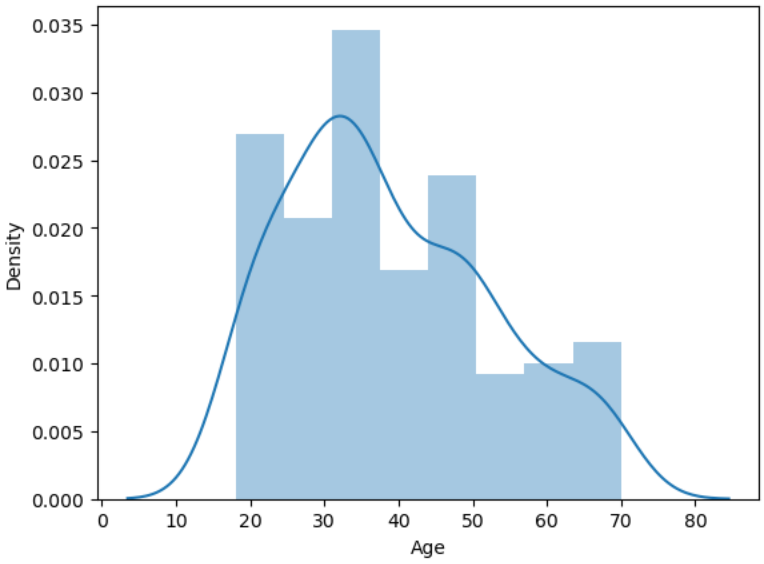
In [4]:
```
# datatype info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

Exploratory Data Analysis

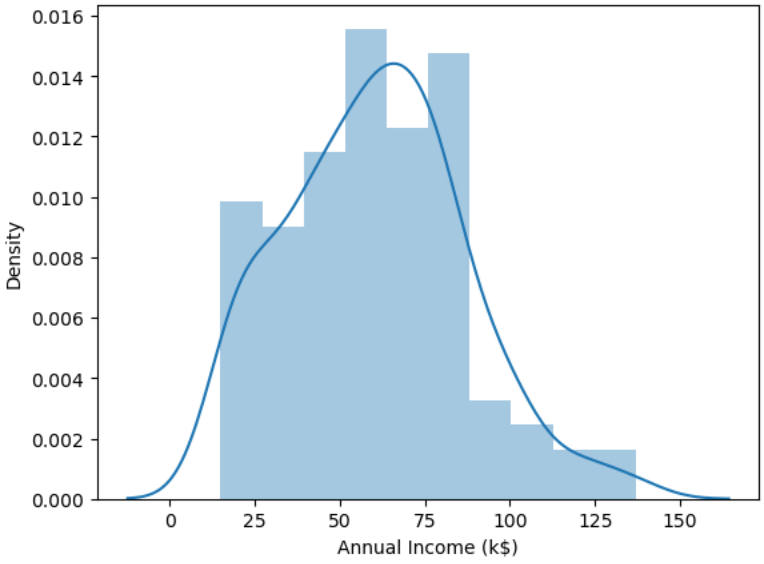
In [5]:
```
sns.distplot(df['Age'])
```

Out[5]:
```
<Axes: xlabel='Age', ylabel='Density'>
```



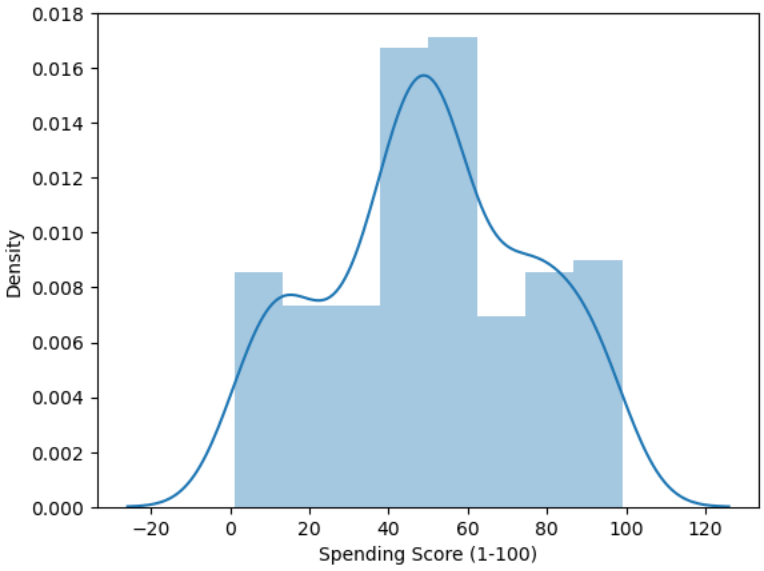
In [6]:
```
sns.distplot(df['Annual Income (k$)'])
```

Out[6]: <Axes: xlabel='Annual Income (k$)', ylabel='Density'>



In [7]:
```
sns.distplot(df['Spending Score (1-100)'])
```

Out[7]: <Axes: xlabel='Spending Score (1-100)', ylabel='Density'>

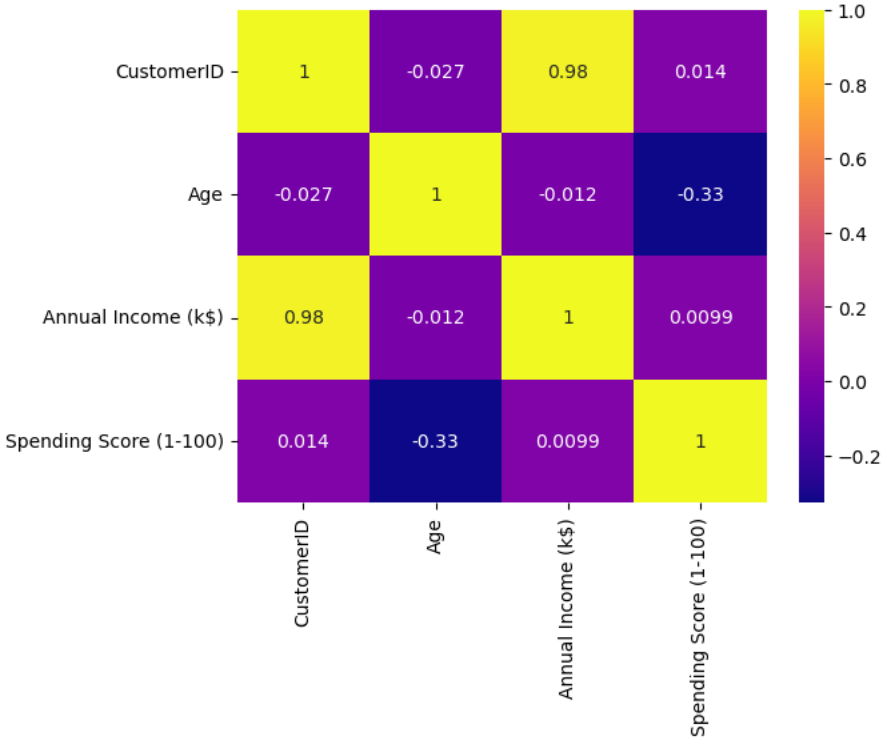


Correlation Matrix

In [8]:
```
corr = df.corr()
```

```
sns.heatmap(corr, annot=True, cmap='plasma')
```

Out[8]: <Axes: >



Clustering

In [9]:
```
df.head()
```

Out[9]:

	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

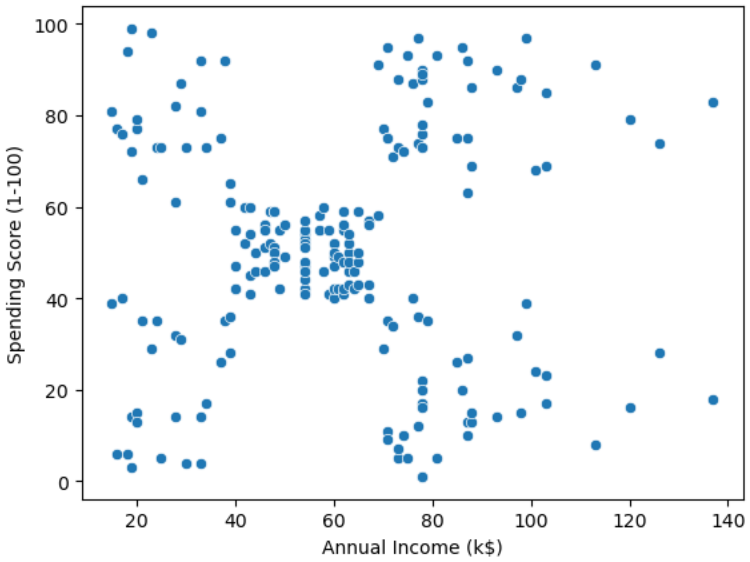
In [10]:
```
# cluster on 2 features
df1 = df[['Annual Income (k$)', 'Spending Score (1-100)']]
df1.head()
```

Out[10]:

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

In [11]:
```
sns.scatterplot(x=df1['Annual Income (k$)'], y=df1['Spending Score (1-100)'])
```

Out[11]: <Axes: xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>

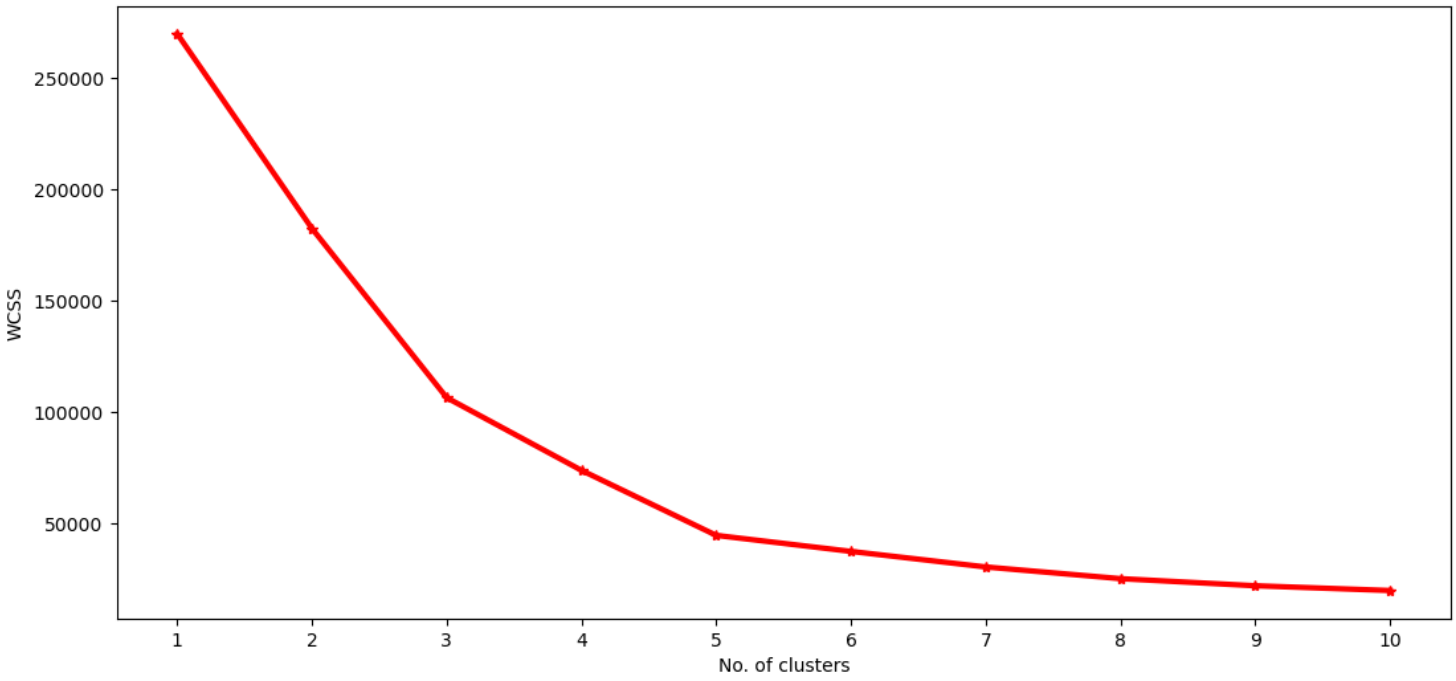


In [12]:

```
from sklearn.cluster import KMeans
wcss= []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(df1)
    wcss.append(kmeans.inertia_)
```

In [13]:

```
# plot the results for elbow method
plt.figure(figsize=(13,6))
plt.plot(range(1,11), wcss)
plt.plot(range(1,11), wcss, linewidth=3, color='red', marker='*')
plt.xlabel('No. of clusters')
plt.ylabel('WCSS')
plt.xticks(np.arange(1,11,1))
plt.show()
```



In [15]:

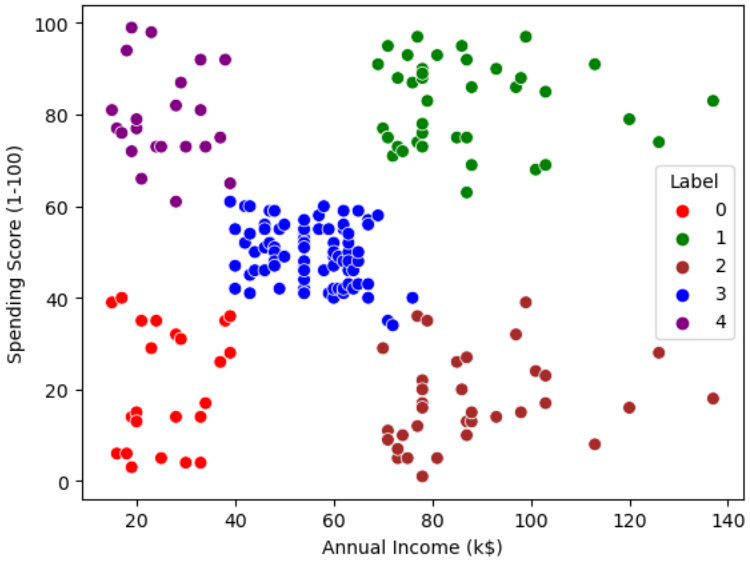
```
km = KMeans(n_clusters=5)
km.fit(df1)
y=km.predict(df1)
df1['Label'] = y
df1.head()
```

Out[15]:

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

In [16]:
```
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=df1, hue='Label', s=50, palette=['red', 'green', 'brown', 'blue', 'pur
```

Out[16]:
```
<Axes: xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



cluster on 3 features

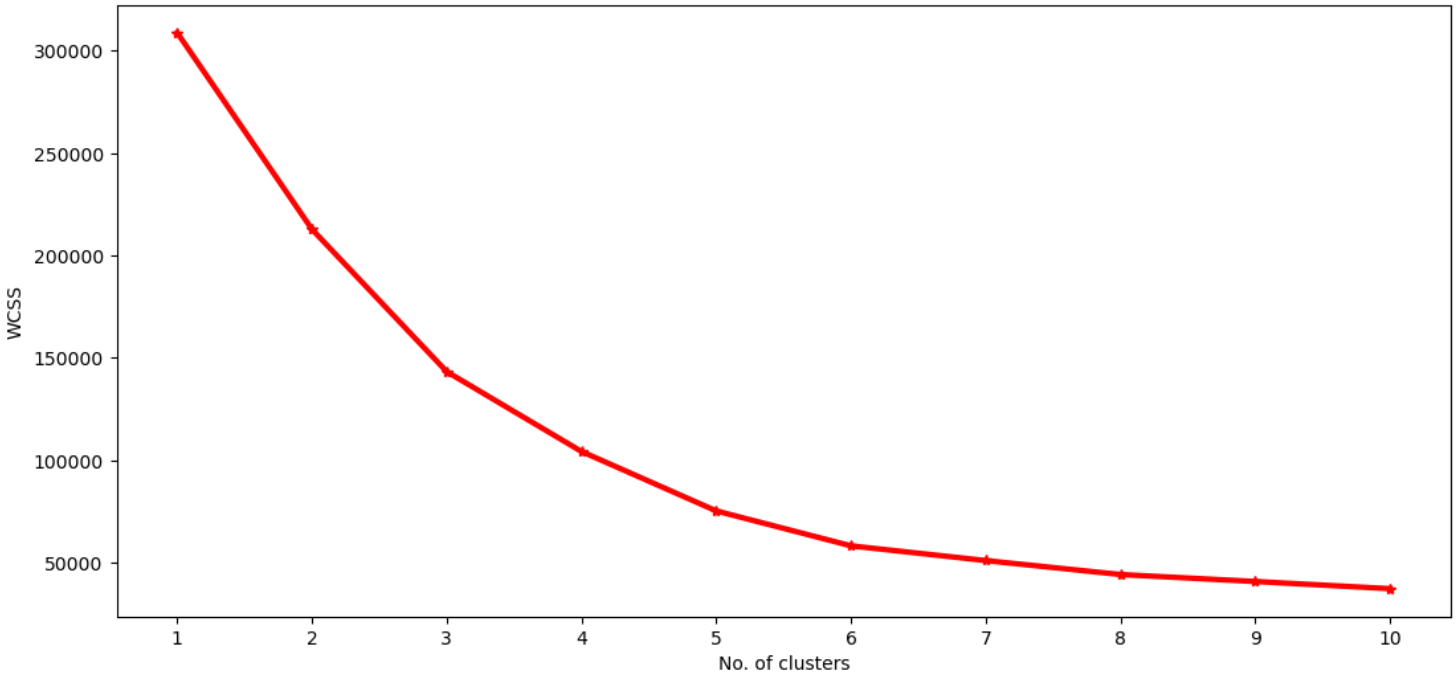
In [17]:
```
df2 = df[['Annual Income (k$)', 'Spending Score (1-100)', 'Age']]
df2.head()
```

Out[17]:

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

In [18]:
```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(df2)
    wcss.append(kmeans.inertia_)
```

In [23]:
```
# plot the results for elbow method
plt.figure(figsize=(13,6))
plt.plot(range(1,11), wcss)
plt.plot(range(1,11), wcss, linewidth=3, color='red', marker='*')
plt.xlabel('No. of clusters')
plt.ylabel('WCSS')
plt.xticks(np.arange(1,11,1))
plt.show()
```



In [20]:

```
km = KMeans(n_clusters=5)
km.fit(df2)
y = km.predict(df2)
df2['Label'] = y
df2.head()
```

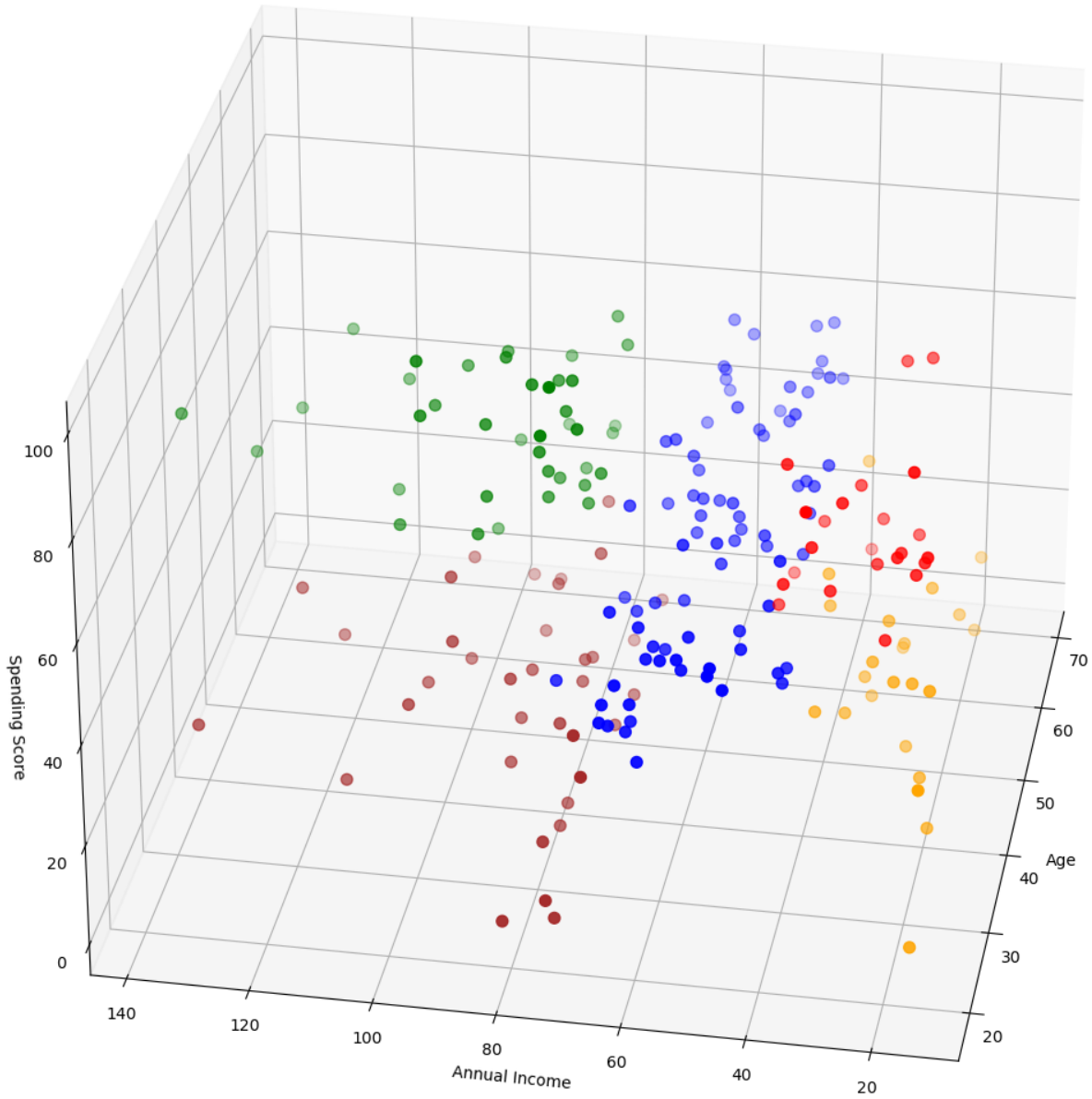
Out[20]:

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

In [21]:

```
# 3d scatter plot
fig = plt.figure(figsize=(20,15))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(df2['Age'][df2['Label']==0], df2['Annual Income (k$)'][df2['Label']==0], df2['Spending Score (1-100)'][df2['Label']==0], c='red', s=
ax.scatter(df2['Age'][df2['Label']==1], df2['Annual Income (k$)'][df2['Label']==1], df2['Spending Score (1-100)'][df2['Label']==1], c='green',
ax.scatter(df2['Age'][df2['Label']==2], df2['Annual Income (k$)'][df2['Label']==2], df2['Spending Score (1-100)'][df2['Label']==2], c='blue', s
ax.scatter(df2['Age'][df2['Label']==3], df2['Annual Income (k$)'][df2['Label']==3], df2['Spending Score (1-100)'][df2['Label']==3], c='brown',
ax.scatter(df2['Age'][df2['Label']==4], df2['Annual Income (k$)'][df2['Label']==4], df2['Spending Score (1-100)'][df2['Label']==4], c='orange',
ax.view_init(30, 190)
ax.set_xlabel('Age')
ax.set_ylabel('Annual Income')
ax.set_zlabel('Spending Score')
plt.show()
```


Spending Score
100
80
60
40
20
0
Annual Income
140
120
100
80
60
40
20
Age
70
60
50
40
30
20