In [1]:

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:

```
df=pd.read_csv(r"C:\Users\sruth\OneDrive\Desktop\Income.csv")
df
```

Out[2]:

	Gender	Age	Income($)
0	Male	19	15
1	Male	21	15
2	Female	20	16
3	Female	23	16
4	Female	31	17
...	...	...	...
195	Female	35	120
196	Female	45	126
197	Male	32	126
198	Male	32	137
199	Male	30	137

200 rows × 3 columns

In [3]:

```
df.head()
```

Out[3]:

	Gender	Age	Income($)
0	Male	19	15
1	Male	21	15
2	Female	20	16
3	Female	23	16
4	Female	31	17

In [4]:

```
df.tail()
```

Out[4]:

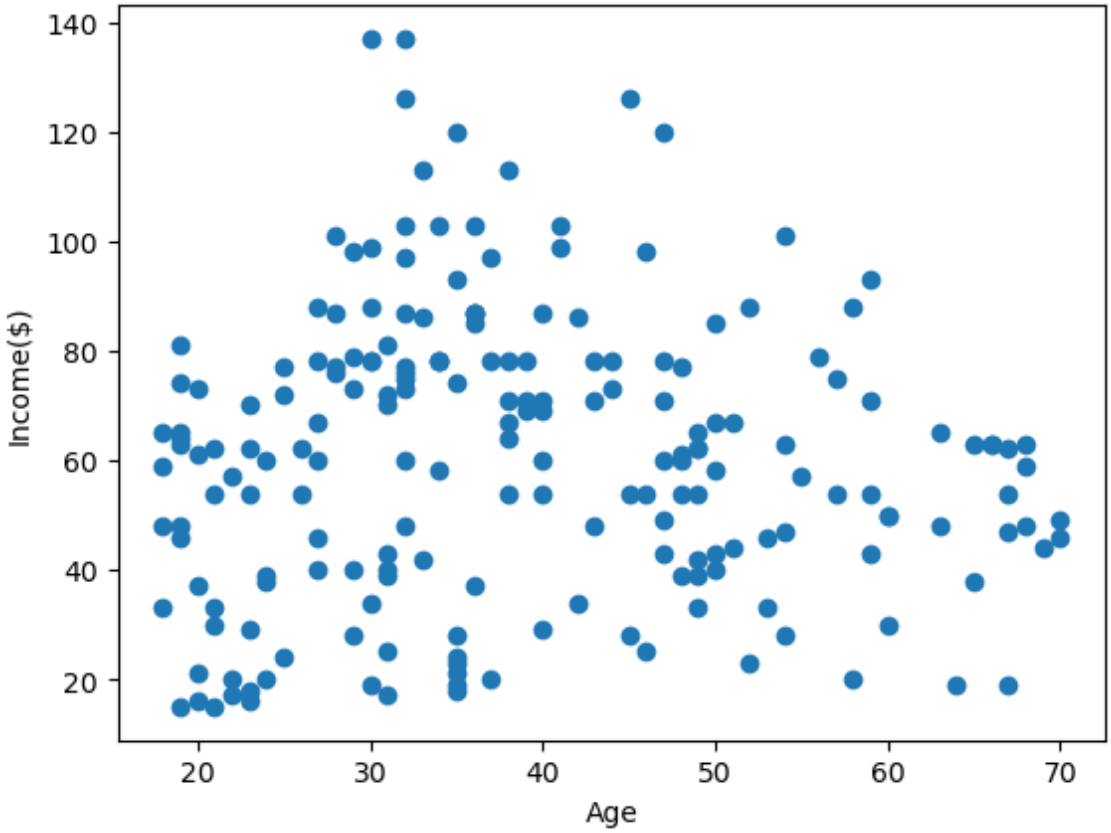
	Gender	Age	Income($)
195	Female	35	120
196	Female	45	126
197	Male	32	126
198	Male	32	137
199	Male	30	137

In [5]:

```
plt.scatter(df["Age"],df["Income($)"])
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[5]:

```
Text(0, 0.5, 'Income($)')
```



In [6]:

```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[6]:

KMeans()

In [7]:

```
y_predicted=km.fit_predict(df[["Age","Income($)"]])
y_predicted
```

```
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[7]:

```
array([4, 4, 4, 4, 4, 4, 4, 4, 2, 4, 2, 4, 2, 4, 4, 4, 4, 4, 2, 4, 4, 4,
       2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 4, 4, 0, 4, 2, 4,
       2, 4, 2, 4, 4, 4, 2, 4, 6, 0, 2, 2, 2, 0, 6, 0, 0, 6, 0, 0, 0, 6,
       2, 0, 6, 6, 0, 0, 0, 0, 0, 6, 1, 1, 6, 1, 0, 1, 0, 1, 6, 1, 0, 6,
       6, 1, 0, 6, 1, 1, 6, 6, 1, 6, 1, 6, 6, 1, 0, 6, 1, 6, 0, 1, 0, 0,
       0, 6, 1, 6, 6, 6, 0, 1, 1, 1, 7, 1, 1, 1, 7, 7, 1, 1, 1, 7, 1, 1,
       7, 7, 7, 7, 1, 7, 7, 7, 1, 7, 7, 7, 7, 7, 1, 7, 7, 7, 7, 7, 1, 7,
       1, 7, 7, 7, 7, 7, 1, 7, 7, 7, 3, 3, 3, 3, 3, 3, 3, 7, 3, 3, 3, 3,
       3, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5,
       5, 5])
```

In [8]:

```
df["cluster"]=y_predicted
df.head()
```

Out[8]:

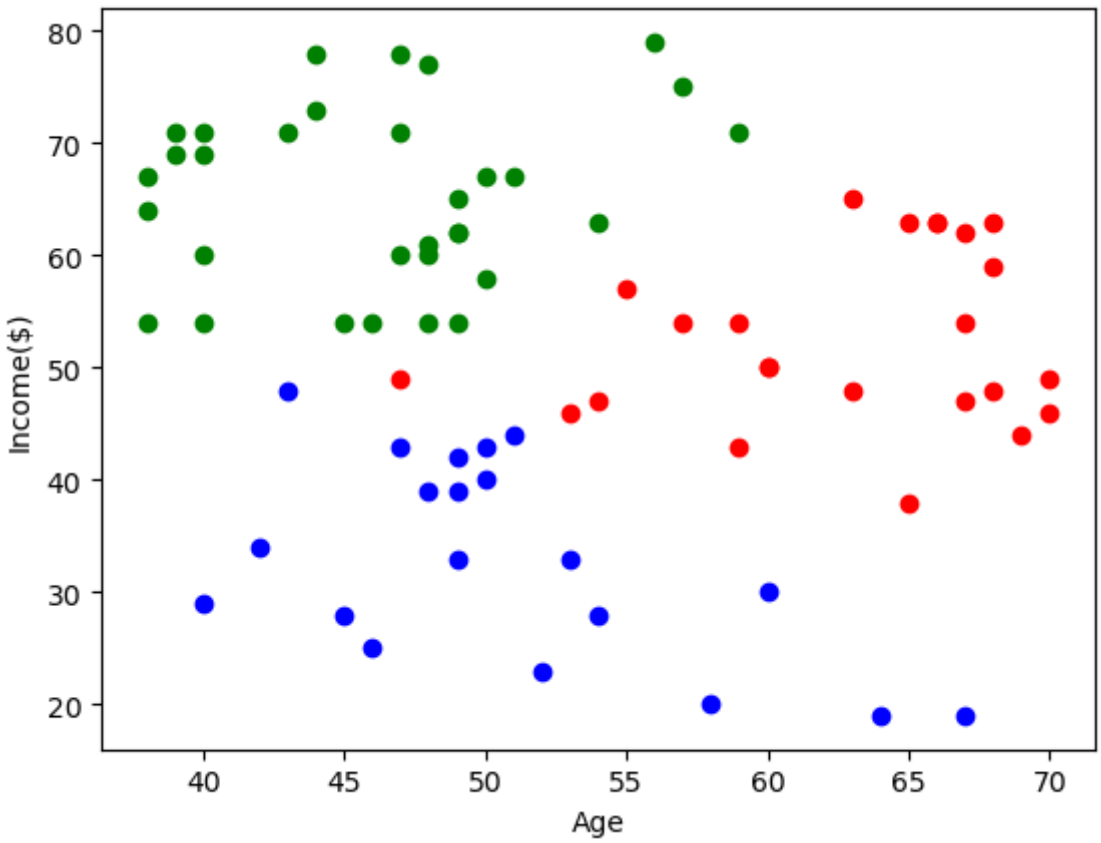
	Gender	Age	Income($)	cluster
0	Male	19	15	4
1	Male	21	15	4
2	Female	20	16	4
3	Female	23	16	4
4	Female	31	17	4

In [9]:

```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["Age"],df1["Income($)"],color="red")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[9]:

```
Text(0, 0.5, 'Income($)')
```



In [10]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["Income($)"]])
df["Income($)"]=scaler.transform(df[["Income($)"]])
df.head()
```

Out[10]:

	Gender	Age	Income($)	cluster
0	Male	19	0.000000	4
1	Male	21	0.000000	4
2	Female	20	0.008197	4
3	Female	23	0.008197	4
4	Female	31	0.016393	4

In [11]:

```
scaler.fit(df[["Age"]])
df["Age"]=scaler.transform(df[["Age"]])
df.head()
```

Out[11]:

	Gender	Age	Income($)	cluster
0	Male	0.019231	0.000000	4
1	Male	0.057692	0.000000	4
2	Female	0.038462	0.008197	4
3	Female	0.096154	0.008197	4
4	Female	0.250000	0.016393	4

In [12]:

```
km=KMeans()
```

In [13]:

```
y_predicted=km.fit_predict(df[["Age","Income($)"]])
y_predicted
```

```
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[13]:

```
array([2, 2, 2, 2, 6, 2, 6, 2, 3, 6, 3, 6, 5, 2, 6, 2, 6, 2, 5, 6, 6, 2,
       5, 6, 5, 6, 5, 6, 6, 2, 3, 2, 5, 2, 5, 2, 5, 6, 6, 2, 3, 2, 5, 6,
       5, 2, 5, 6, 6, 6, 5, 6, 6, 3, 5, 5, 5, 3, 4, 5, 3, 4, 3, 5, 3, 4,
       5, 3, 4, 6, 3, 5, 3, 3, 3, 4, 5, 5, 4, 5, 3, 7, 3, 5, 4, 5, 1, 4,
       7, 1, 3, 4, 1, 7, 7, 4, 1, 4, 1, 4, 4, 1, 3, 4, 1, 4, 3, 1, 3, 3,
       3, 4, 7, 4, 4, 4, 3, 1, 1, 1, 4, 7, 7, 7, 4, 7, 1, 7, 1, 7, 1, 7,
       4, 7, 4, 7, 1, 7, 4, 7, 1, 7, 7, 7, 4, 7, 1, 7, 7, 7, 1, 7, 1, 7,
       1, 7, 7, 7, 7, 7, 1, 7, 4, 7, 1, 7, 7, 7, 7, 7, 7, 7, 7, 7, 1, 7,
       1, 7, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0])
```

In [14]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="red")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker="+")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

```
---------------------------------------------------------------------
--
KeyError                                  Traceback (most recent call las
t)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3802, in I
ndex.get_loc(self, key, method, tolerance)
   3801 try:
-> 3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:138, in pandas.
_libs.index.IndexEngine.get_loc()

File ~\anaconda3\lib\site-packages\pandas\_libs\index.pyx:165, in pandas.
_libs.index.IndexEngine.get_loc()

File pandas\_libs\hashtable_class_helper.pxi:5745, in pandas._libs.hashta
ble.PyObjectHashTable.get_item()

File pandas\_libs\hashtable_class_helper.pxi:5753, in pandas._libs.hashta
ble.PyObjectHashTable.get_item()

KeyError: 'New Cluster'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call las
t)
Cell In[14], line 1
----> 1 df1=df[df["New Cluster"]==0]
      2 df2=df[df["New Cluster"]==1]
      3 df3=df[df["New Cluster"]==2]

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:3807, in DataFram
e.__getitem__(self, key)
   3805 if self.columns.nlevels > 1:
   3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
   3808 if is_integer(indexer):
   3809     indexer = [indexer]

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:3804, in I
ndex.get_loc(self, key, method, tolerance)
   3802     return self._engine.get_loc(casted_key)
   3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
   3805 except TypeError:
   3806     # If we have a listlike key, _check_indexing_error will raise
   3807     #  InvalidIndexError. Otherwise we fall through and re-raise
   3808     #  the TypeError.
   3809     self._check_indexing_error(key)

KeyError: 'New Cluster'
```

In [15]:

```
k_rng=range(1,10)
sse=[]
```

In [*]:

```
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["Age","Income($)"]])
sse.append(km.inertia_)
#km.inertia_ will give you the value of sum of square error
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
```

```
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:87
0: FutureWarning: The default value of `n_init` will change from 10 to 'a
uto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\sruth\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:138
2: UserWarning: KMeans is known to have a memory leak on Windows with MK
L, when there are less chunks than available threads. You can avoid it by
setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

```
[2.34881665014649]
```

In [ ]: