In [1]:
```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

In [2]:
```
df=pd.read_csv(r"C:\Users\Dell\Downloads\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 [4]:
```
df.head()
```

Out[4]:

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

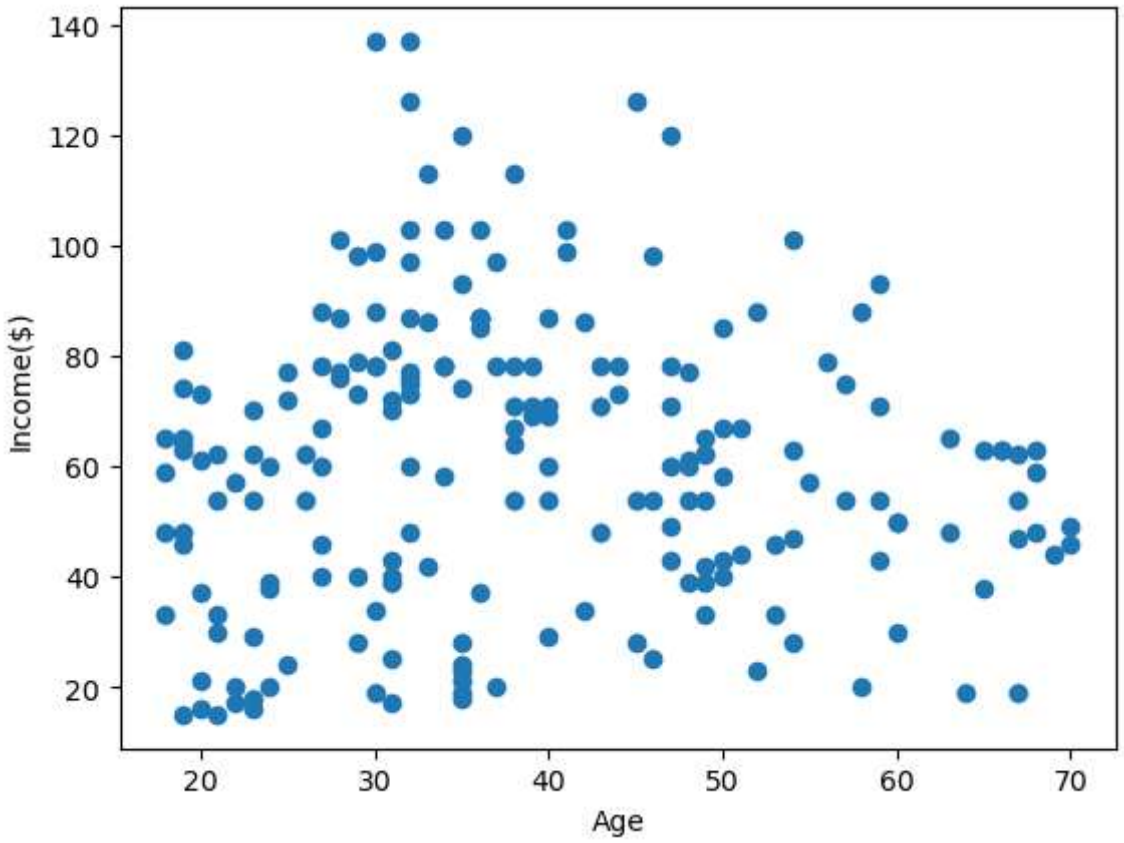
In [5]:
```
df.tail()
```

Out[5]:

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

In [12]:
```
plt.scatter(df["Age"],df["Income($)"])
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[12]:
```
Text(0, 0.5, 'Income($)')
```



In [13]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[13]:
```
KMeans()
```

In [14]:
```
y_predicted=km.fit_predict(df[["Age","Income($)"]])
y_predicted
```

```
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[14]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 7, 1, 7, 1, 7, 1, 1, 1, 1, 1, 7, 1, 1, 1,
       7, 1, 7, 1, 7, 1, 7, 1, 7, 1, 7, 6, 7, 6, 7, 6, 6, 6, 7, 6, 7, 6,
       7, 6, 7, 6, 6, 6, 7, 6, 6, 3, 7, 7, 7, 3, 6, 3, 3, 6, 3, 3, 3, 6,
       7, 3, 6, 6, 3, 3, 3, 3, 3, 6, 3, 0, 6, 3, 3, 6, 3, 3, 6, 3, 3, 4,
       4, 3, 3, 4, 0, 0, 4, 4, 0, 4, 0, 4, 4, 0, 3, 4, 0, 4, 3, 3, 3, 3,
       3, 4, 0, 4, 4, 4, 3, 0, 0, 0, 4, 0, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0,
       4, 4, 4, 4, 0, 4, 4, 4, 0, 4, 4, 4, 4, 4, 0, 4, 4, 4, 0, 0, 0, 0,
       0, 4, 0, 4, 4, 4, 0, 4, 4, 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, 5, 5, 5, 5, 5, 5,
       5, 5])
```

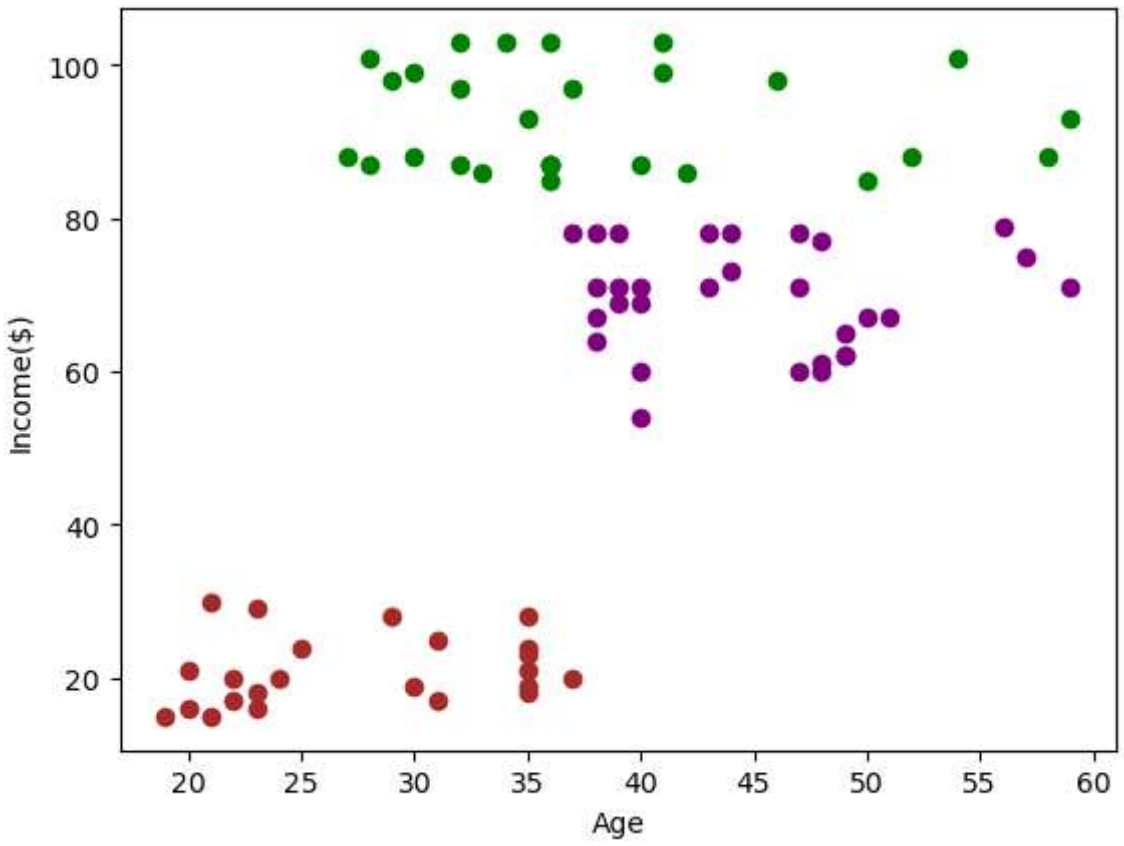
In [15]:
```
df["cluster"]=y_predicted
df.head()
```

Out[15]:

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

In [21]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["Age"],df1["Income($)"],color="purple")
plt.scatter(df2["Age"],df2["Income($)"],color="brown")
plt.scatter(df3["Age"],df3["Income($)"],color="green")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[21]:
```
Text(0, 0.5, 'Income($)')
```



In [22]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["Income($)"]])
df["Income($)"]=scaler.transform(df[["Income($)"]])
df.head()
```

Out[22]:

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

In [23]:
```
scaler.fit(df[["Age"]])
df["Age"]=scaler.transform(df[["Age"]])
df.head()
```

Out[23]:

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

In [24]:
```
km=KMeans()
```

In [25]:
```
y_predicted=km.fit_predict(df[["Age","Income($)"]])
y_predicted
```

```
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[25]:
```
array([7, 7, 7, 7, 3, 7, 3, 7, 0, 3, 0, 3, 0, 7, 3, 7, 3, 7, 6, 3, 3, 7,
       6, 3, 6, 3, 6, 3, 3, 7, 0, 7, 6, 7, 6, 7, 6, 3, 3, 7, 0, 7, 6, 3,
       6, 7, 6, 3, 3, 3, 6, 3, 3, 0, 6, 6, 6, 0, 3, 6, 0, 4, 0, 6, 0, 4,
       6, 0, 4, 3, 0, 6, 0, 0, 0, 4, 6, 6, 4, 6, 0, 1, 0, 6, 4, 6, 6, 4,
       1, 6, 0, 4, 6, 1, 1, 4, 6, 4, 6, 4, 4, 6, 0, 4, 6, 4, 0, 5, 0, 0,
       0, 4, 1, 4, 4, 4, 0, 5, 5, 5, 4, 1, 1, 1, 4, 1, 5, 1, 5, 1, 5, 1,
       4, 1, 4, 1, 5, 1, 4, 1, 5, 1, 1, 1, 4, 1, 5, 1, 1, 1, 5, 1, 5, 1,
       5, 1, 1, 1, 1, 1, 5, 1, 4, 1, 5, 1, 5, 1, 1, 1, 1, 1, 1, 1, 5, 1,
       5, 1, 5, 2, 2, 2, 5, 2, 2, 2, 5, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

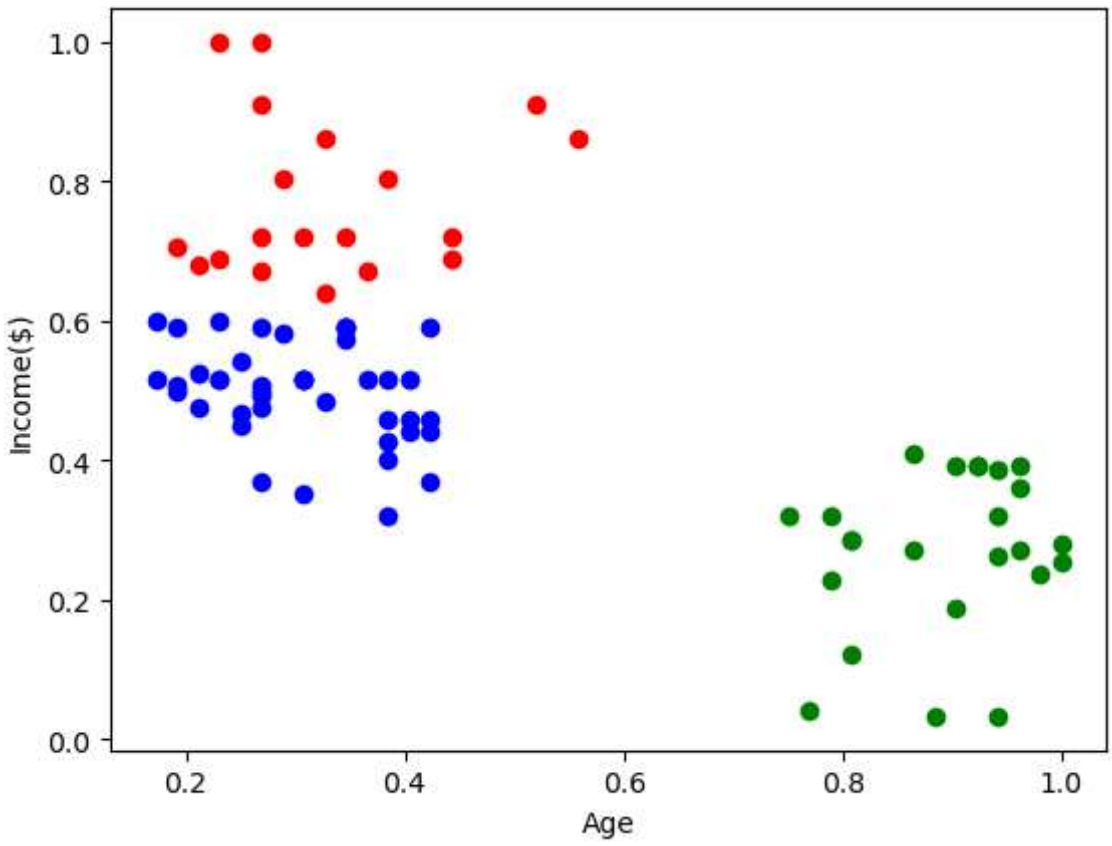
In [26]:
```
df["New Cluster"]=y_predicted
df.head()
```

Out[26]:

	Gender	Age	Income($)	cluster	New Cluster
0	Male	0.019231	0.000000	1	7
1	Male	0.057692	0.000000	1	7
2	Female	0.038462	0.008197	1	7
3	Female	0.096154	0.008197	1	7
4	Female	0.250000	0.016393	1	3

In [28]:
```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="green")
plt.scatter(df2["Age"],df2["Income($)"],color="blue")
plt.scatter(df3["Age"],df3["Income($)"],color="red")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[28]:
```
Text(0, 0.5, 'Income($)')
```

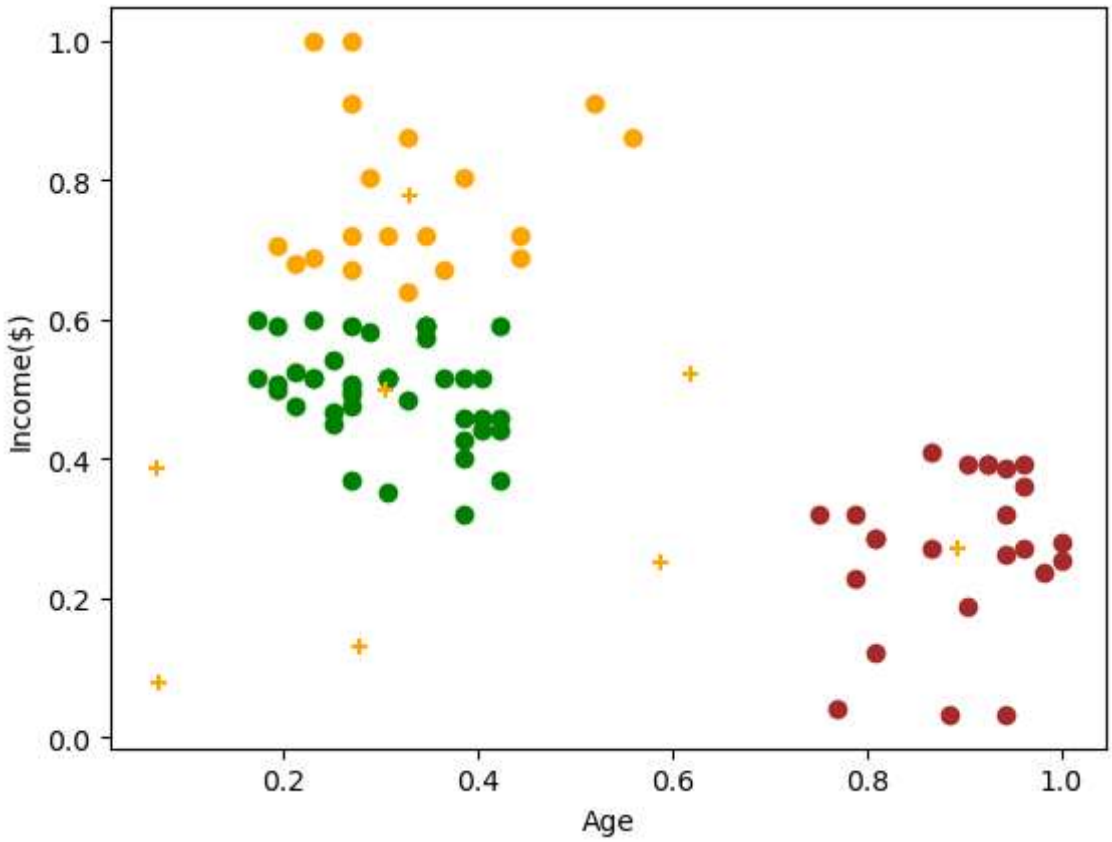


In [29]:
```
km.cluster_centers_
```

Out[29]:
```
array([[0.89262821, 0.27015027],
       [0.30540293, 0.49921936],
       [0.32894737, 0.77782571],
       [0.27884615, 0.13040238],
       [0.06923077, 0.38786885],
       [0.61813187, 0.52185792],
       [0.58717949, 0.25245902],
       [0.07239819, 0.08003857]])
```

In [31]:
```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="brown")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="orange")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",m
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[31]:
```
Text(0, 0.5, 'Income($)')
```



In [32]:
```
k_rng=range(1,10)
sse=[]
```

In [33]:
```
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:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto' in
1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\ProgramData\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting th
e environment variable OMP_NUM_THREADS=1.
  warnings.warn(

[23.583906150363603, 13.028938428018286, 7.492107868586012, 6.05582466759962
3, 4.713416604872824, 3.8710582198144317, 3.054717436369358, 2.6425203435360
72, 2.3135720353543285]
```

Out[33]: Text(0, 0.5, 'Sum of Squared Error')