

```
In [1]: !pip install sklearn

Requirement already satisfied: sklearn in c:\users\pranav\continuum\ppp\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\pranav\continuum\ppp\lib\site-packages (from sklearn) (1.1.2)
Requirement already satisfied: joblib>=1.0.0 in c:\users\pranav\continuum\ppp\lib\site-packages (from scikit-learn->sklearn) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\pranav\continuum\ppp\lib\site-packages (from scikit-learn->sklearn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\pranav\continuum\ppp\lib\site-packages (from scikit-learn->sklearn) (1.21.5)
Requirement already satisfied: scipy>=1.3.2 in c:\users\pranav\continuum\ppp\lib\site-packages (from scikit-learn->sklearn) (1.7.3)
```

```
In [2]: from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
```

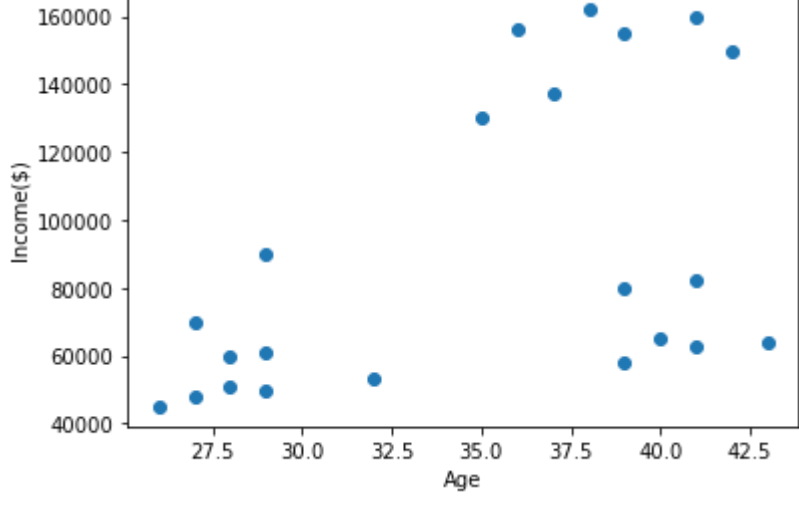
```
df=pd.read_excel('C:\Users\NITANAV\Desktop\DATA SCIENCE DATA\excel file\INCOME.XLSX')
```

```
In [4]: df.head()
```

```
Out[4]:
```

	Name	Age	Income(\$)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000

```
In [5]: plt.scatter(df['Age'], df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
plt.show()
```



```
in ['0']: km = km + 1
```

```
In [7]: kn
```

```
Out[7]: KMeans
KMeans(n_clusters=3)
```

```
In [8]: y_predict=km.fit_predict(df[['Age', 'Income($)']])
```

variable

```
In [9]: y_predict
```

```
out[0]: array([0, 0, 2, 2, 1, 1,
```

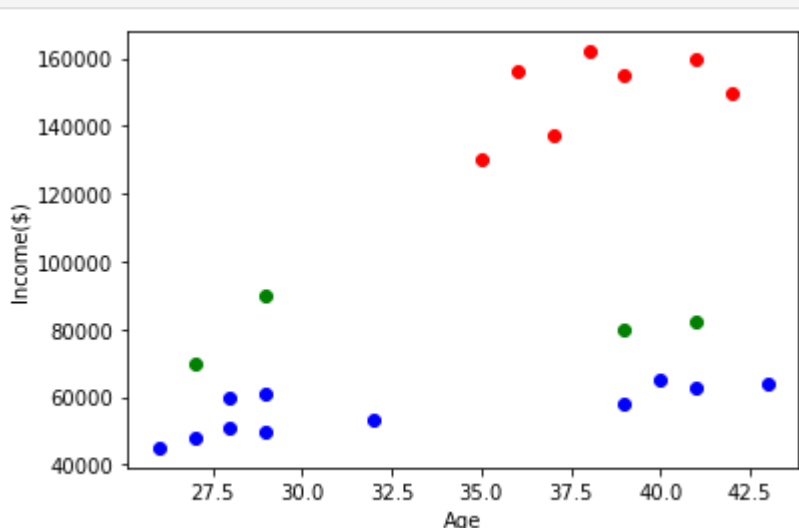
In [11]:	df				
Out[11]:		Name	Age	Income(\$)	clusters
	0	Rob	27	70000	0
	1	Michael	29	90000	0
	2	Mohan	29	61000	2
	3	Ismail	28	60000	2
	4	Kory	42	150000	1
	5	Gautam	39	155000	1
	6	David	41	160000	1
	7	Andrea	38	162000	1
	8	Brad	36	156000	1
	9	Angelina	35	130000	1
	10	Donald	37	137000	1
	11	Tom	26	45000	2
	12	Arnold	27	48000	2
	13	Jared	28	51000	2
	14	Stark	29	49500	2
	15	Ranbir	32	53000	2
	16	Dipika	40	65000	2
	17	Priyanka	41	63000	2
	18	Nick	43	64000	2
	19	Alia	39	80000	0
	20	Sid	41	82000	0
	21	Abdul	39	58000	2

```
Out[12]: array([[3.40000000e+01, 8.05000000e+04],
        [3.82857143e+01, 1.50000000e+05],
```

```
In [14]: df1=df[df.clusters==0]
```

```
df2=df[df.clusters==1]
df3=df[df.clusters==2]
```

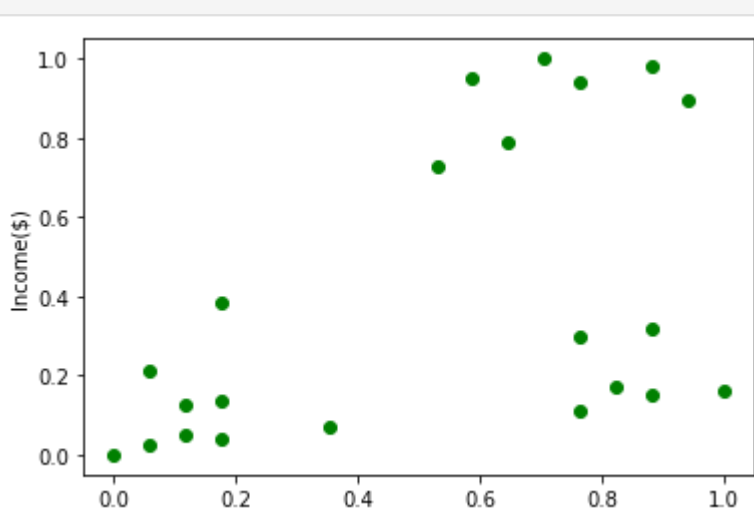
```
[21]: plt.scatter(df1['Age'], df1['Income($')', color='g')
plt.scatter(df2['Age'], df2['Income($')', color='r')
plt.scatter(df3['Age'], df3['Income($')', color='b')
plt.xlabel('Age')
plt.ylabel('Income($')
```



```
In [28]: scaler.fit(df[['Income($)']])
df['Income($)']=scaler.transform(df[['Income($)']])
scaler.fit(df[['Age']])
df['Age']=scaler.transform(df[['Age']])
```

	Name	Age	Income(\$)	clusters
0	Rob	0.058824	0.213675	0
1	Michael	0.176471	0.384615	0
2	Mohan	0.176471	0.136752	2
3	Ismail	0.117647	0.128205	2
4	Kegon	0.041176	0.093426	1

```
In [30]: plt.scatter(df['Age'],df['Income($)',color='g')
plt.ylabel('Income($)')
```



```
In [33]: km=KMeans(n_clusters=3)
```

```
C:\Users\Pranav\continuum\ppp\lib\site-packages\sklearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
```

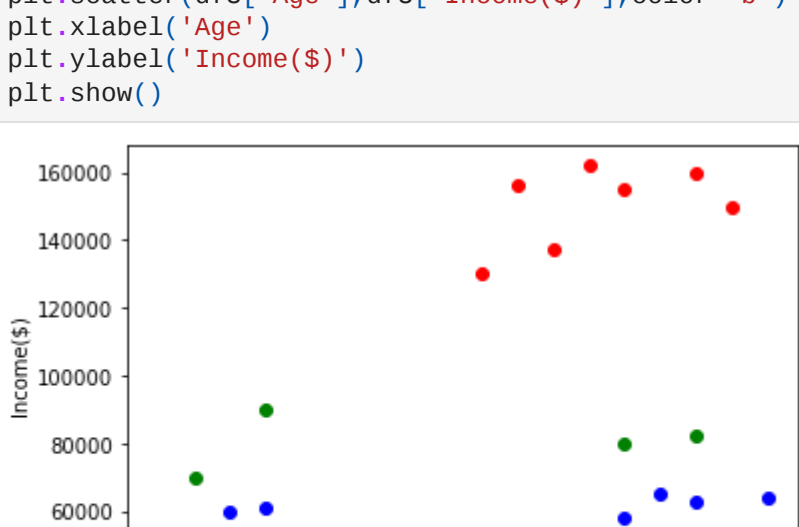
```
fn [34]: y predicted
```

```
Out[34]: array([0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1])
```

```
df["category"] = "mediated"
```

```
[0.85294118, 0.2022792 ],
 [0.72268908, 0.8974359 ]])

In [38]: plt.scatter(df1['Age'],df1['Income($)',color='g'])
plt.scatter(df2['Age'],df2['Income($)',color='r'])
```



```

rng = range(1,10)
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[['Age','Income($)']])
    sse.append(km.inertia_)

C:\Users\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, 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\Pranav\.continuum\ppp\lib\site-packages\skslearn\cluster\_kmeans.py:1332: UserWarning: KMeans is known to have a mem
ory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(

```

```
Out[42]: [5.434011511988176,
          2.0911363886990766,
          0.47507834985530933,
          0.3401047004410564]
```

0.20040301240084145,
0.21055478995472487,
0.16858512236029757,
0.1378188013376402,
0.10497488680620908]

