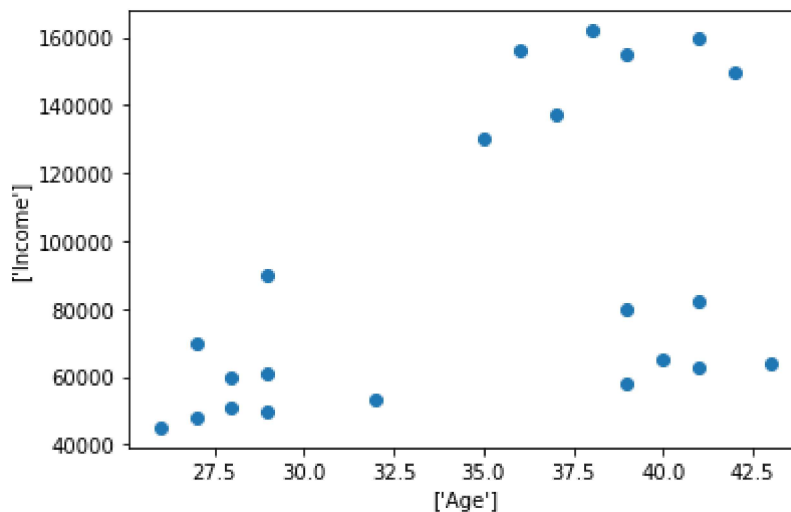```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        data=pd.read_csv('kmeans.csv')
        data.head()
```

Out[1]:

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

```
In [2]: plt.scatter(data['Age'],data['Income($)'])
        plt.xlabel(['Age'])
        plt.ylabel(['Income'])
        plt.show()
```



```
In [3]: from sklearn.cluster import KMeans
        km=KMeans(n_clusters=3)
        y_predict=km.fit_predict(data[['Age','Income($)']])
        y_predict
```

Out[3]: array([2, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0])

```
In [4]: data['clusters']=y_predict
        data.head()
```

Out[4]:

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

```
In [5]: km.cluster_centers_
```

Out[5]: array([[3.29090909e+01, 5.61363636e+04],
               [3.82857143e+01, 1.50000000e+05],
               [3.40000000e+01, 8.05000000e+04]])

```
In [6]: df1=data[data.clusters==0]
        df2=data[data.clusters==1]
        df3=data[data.clusters==2]
        plt.scatter(df1['Age'],df1['Income($)'],color='red')
        plt.scatter(df2['Age'],df2['Income($)'],color='blue')
        plt.scatter(df3['Age'],df3['Income($)'],color='green')
        plt.show()
```
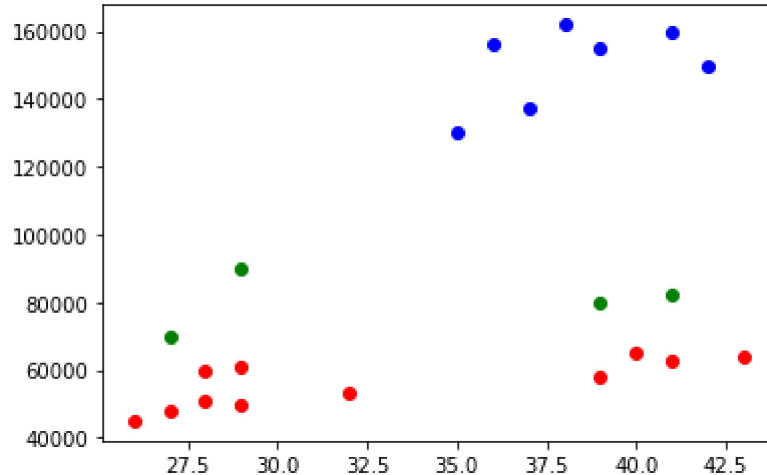
```
In [7]: from sklearn.preprocessing import MinMaxScaler
        s=MinMaxScaler()
        s.fit(data[['Age']])
        data['Age']=s.transform(data[['Age']])

        s.fit(data[['Income($)']])
        data['Income']=s.transform(data[['Income($)']])

        data.head()
```

Out[7]:

	Name	Age	Income($)	clusters	Income
0	Rob	0.058824	70000	2	0.213675
1	Michael	0.176471	90000	2	0.384615
2	Mohan	0.176471	61000	0	0.136752
3	Ismail	0.117647	60000	0	0.128205
4	Kory	0.941176	150000	1	0.897436

```
In [8]: km=KMeans(n_clusters=3)
        y_predict=km.fit_predict(data[['Age','Income']])
        y_predict
```

Out[8]: array([1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2])

```
In [9]: data['clusters']=y_predict
        data.head()
```

Out[9]:

	Name	Age	Income($)	clusters	Income
0	Rob	0.058824	70000	1	0.213675
1	Michael	0.176471	90000	1	0.384615
2	Mohan	0.176471	61000	1	0.136752
3	Ismail	0.117647	60000	1	0.128205
4	Kory	0.941176	150000	0	0.897436

```
In [10]: km.cluster_centers_
```

Out[10]: array([[0.72268908, 0.8974359],
               [0.1372549 , 0.11633428],
               [0.85294118, 0.2022792]])

```python
df1=data[data.clusters==0]
df2=data[data.clusters==1]
df3=data[data.clusters==2]
plt.scatter(df1['Age'],df1['Income($)'],color='red')
plt.scatter(df2['Age'],df2['Income($)'],color='blue')
plt.scatter(df3['Age'],df3['Income($)'],color='green')
plt.show()
```
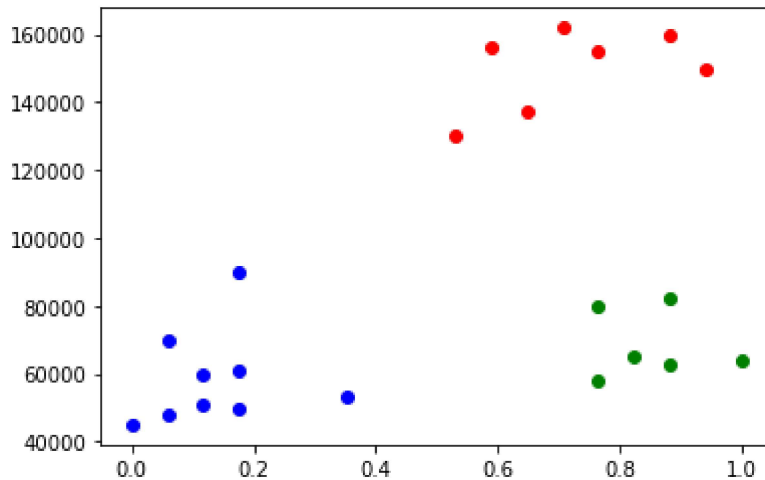
```python
ran=range(1,10)
s=[]
for i in ran:

    km=KMeans(n_clusters=i)
    km.fit(data[['Age','Income']])
    s.append(km.inertia_)

s
```
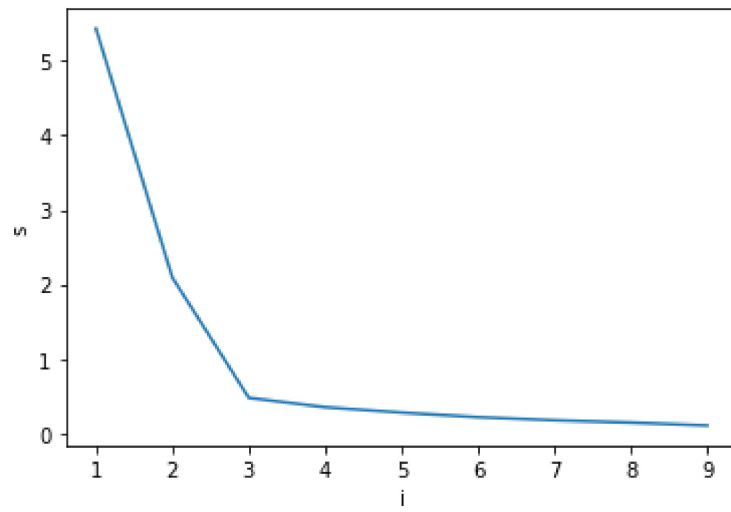
```
C:\Users\admin\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: User
Warning: KMeans is known to have a memory leak on Windows with MKL, when there
are less chunks than available threads. You can avoid it by setting the environ
ment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

```
[5.434011511988179,
 2.091136388699078,
 0.4750783498553097,
 0.3491047094419566,
 0.2766936276300279,
 0.21552563314755377,
 0.17355596555312644,
 0.1425707624969037,
 0.10383752586603562]
```

```
In [17]: plt.xlabel('i')
         plt.ylabel('s')
         plt.plot(ran,s)
         plt.show()
```



```
In [ ]:
```