Clustering With K Means - Python Tutorial

In [4]:
```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

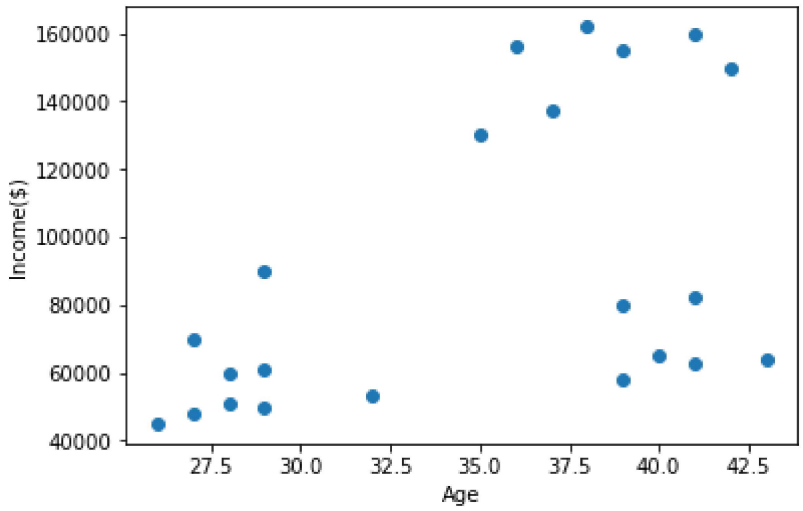
In [12]:
```
df = pd.read_csv(r"D:\abc\abc\abc\ML\13_kmeans\income.csv")
df
```

Out[12]:

	Name	Age	Income($)
0	Rob	27	70000
1	Michael	29	90000
2	Mohan	29	61000
3	Ismail	28	60000
4	Kory	42	150000
5	Gautam	39	155000
6	David	41	160000
7	Andrea	38	162000
8	Brad	36	156000
9	Angelina	35	130000
10	Donald	37	137000

In [198]:
```
plt.scatter(df.Age,df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

Out[198]:
```
<matplotlib.text.Text at 0x159c7655ac8>
```



In [13]:
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

Out[13]:
```
array([2, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0])
```

In [17]:
```
df['cluster']=y_predicted
df
```

Out[17]:

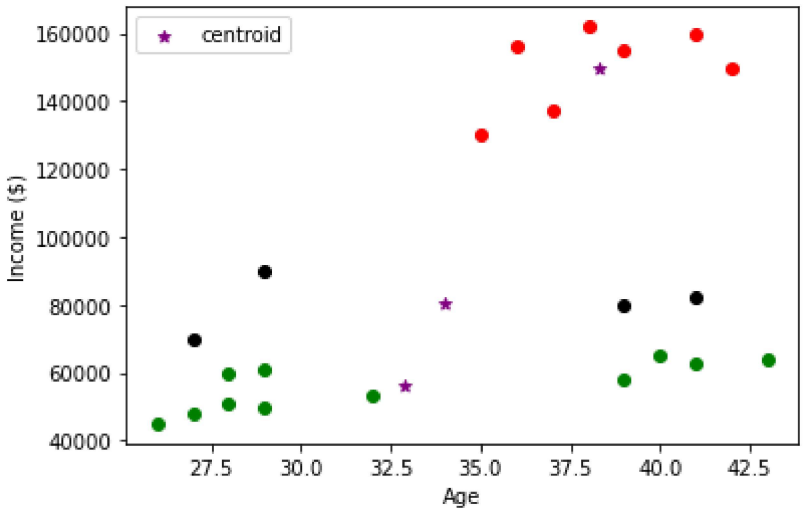
	Name	Age	Income($)	cluster
0	Rob	27	70000	2
1	Michael	29	90000	2
2	Mohan	29	61000	0
3	Ismail	28	60000	0
4	Kory	42	150000	1
5	Gautam	39	155000	1
6	David	41	160000	1
7	Andrea	38	162000	1
8	Brad	36	156000	1
9	Angelina	35	130000	1
10	Donald	37	137000	1
11	Tom	26	45000	0
12	Arnold	27	48000	0
13	Jared	28	51000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

In [9]:
```
km.cluster_centers_
```

Out[9]:
```
array([[3.29090909e+01, 5.61363636e+04],
       [3.82857143e+01, 1.50000000e+05],
       [3.40000000e+01, 8.05000000e+04]])
```

In [10]:
```
1 = df[df.cluster==0]
2 = df[df.cluster==1]
3 = df[df.cluster==2]
t.scatter(df1.Age,df1['Income($)'],color='green')
t.scatter(df2.Age,df2['Income($)'],color='red')
t.scatter(df3.Age,df3['Income($)'],color='black')
t.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',mark
t.xlabel('Age')
t.ylabel('Income ($)')
t.legend()
```

Out[10]: <matplotlib.legend.Legend at 0x20105063b80>



In [20]:
```
df1
```

Out[20]:

	Name	Age	Income($)	cluster
2	Mohan	29	61000	0
3	Ismail	28	60000	0
11	Tom	26	45000	0
12	Arnold	27	48000	0
13	Jared	28	51000	0
14	Stark	29	49500	0
15	Ranbir	32	53000	0
16	Dipika	40	65000	0
17	Priyanka	41	63000	0
18	Nick	43	64000	0
21	Abdul	39	58000	0

In [21]:
```
df2
```

Out[21]:

	Name	Age	Income($)	cluster
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

In [22]:
```
df3
```

Out[22]:

	Name	Age	Income($)	cluster
0	Rob	27	70000	2
1	Michael	29	90000	2
19	Alia	39	80000	2
20	Sid	41	82000	2

Preprocessing using min max scaler

In [203]:
```
scaler = MinMaxScaler()

scaler.fit(df[['Income($)']])
df['Income($)'] = scaler.transform(df[['Income($)']])

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

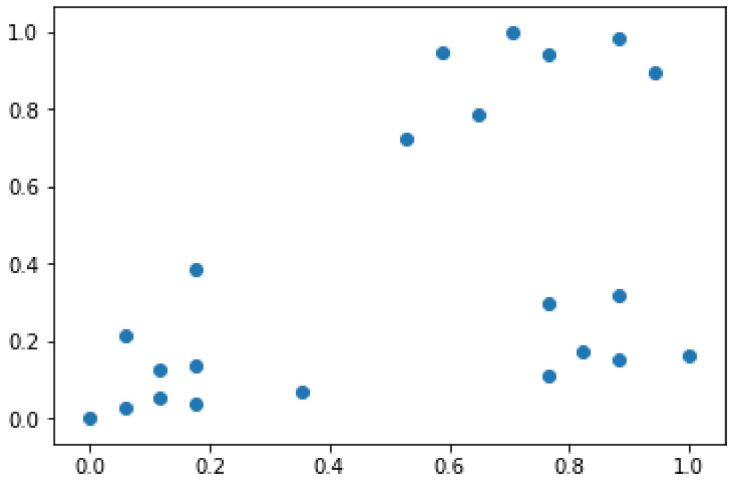
In [204]:
```
df.head()
```

Out[204]:

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	2
1	Michael	0.176471	0.384615	2
2	Mohan	0.176471	0.136752	0
3	Ismail	0.117647	0.128205	0
4	Kory	0.941176	0.897436	1

In [205]:
```
plt.scatter(df.Age,df['Income($)'])
```

Out[205]:
```
<matplotlib.collections.PathCollection at 0x159c78f2358>
```



In [206]:
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

Out[206]:
```
array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2])
```

In [207]:
```
df['cluster']=y_predicted
df.head()
```

Out[207]:

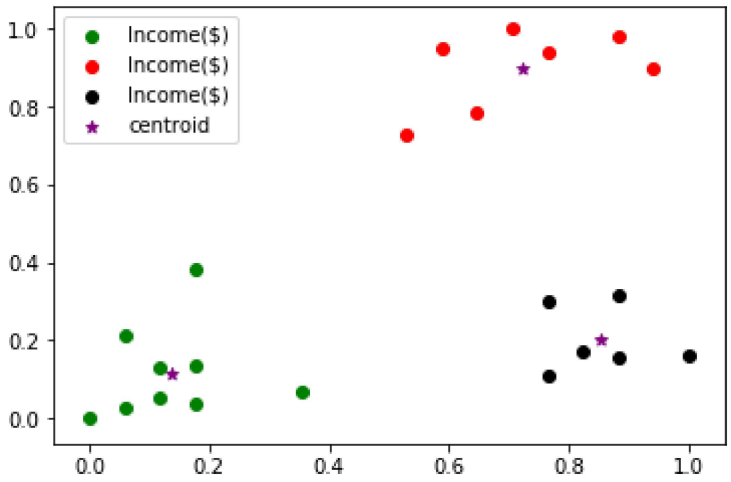
	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	0
1	Michael	0.176471	0.384615	0
2	Mohan	0.176471	0.136752	0
3	Ismail	0.117647	0.128205	0
4	Kory	0.941176	0.897436	1

In [208]:
```
km.cluster_centers_
```

Out[208]:
```
array([[ 0.1372549 ,  0.11633428],
       [ 0.72268908,  0.8974359 ],
       [ 0.85294118,  0.2022792 ]])
```

In [209]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',m
plt.legend()
```

Out[209]: <matplotlib.legend.Legend at 0x159c7982f60>

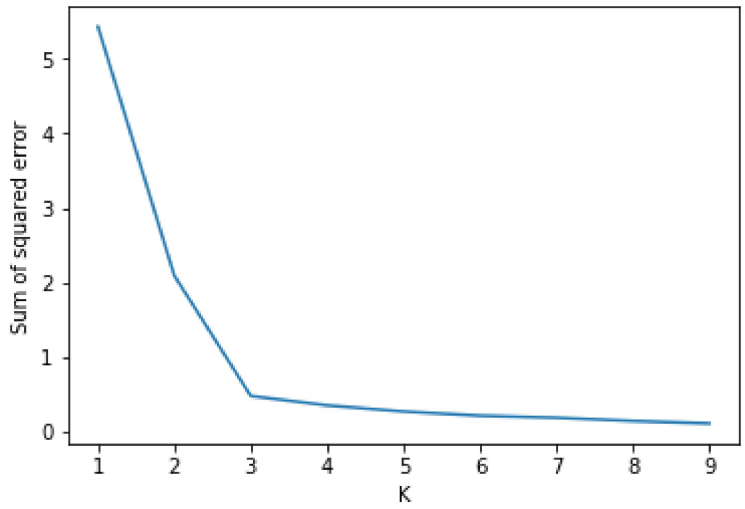


Elbow Plot

In [210]:
```
sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['Age','Income($)']])
    sse.append(km.inertia_)
```

In [211]:
```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

Out[211]:
```
[<matplotlib.lines.Line2D at 0x159c7a34978>]
```



Exercise

1. Use iris flower dataset from sklearn library and try to form clusters of flowers using petal width and length features. Drop other two features for simplicity.
2. Figure out if any preprocessing such as scaling would help here
3. Draw elbow plot and from that figure out optimal value of k