Clustering With K Means - Python Tutorial

In [20]:
```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

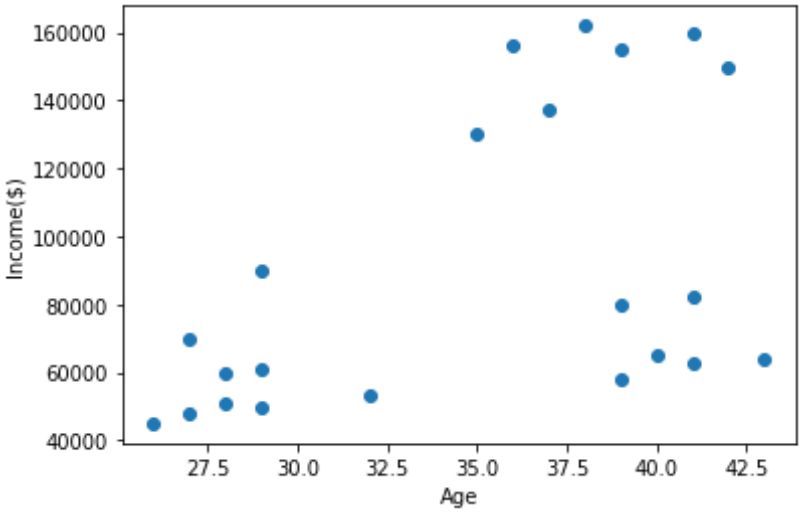
In [24]:
```
df = pd.read_csv("C:/Users/prasa/Desktop/py codes/ds projects/ML/12 kme
ans/income.csv")
df.head()
```

Out[24]:

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

In [25]:
```
plt.scatter(df.Age,df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

Out[25]:
```
Text(0, 0.5, 'Income($)')
```



In [26]:
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

Out[26]:
```
array([0, 0, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0,
2])
```

In [27]:
```
df['cluster'] = y_predicted
df.head()
```

Out[27]:

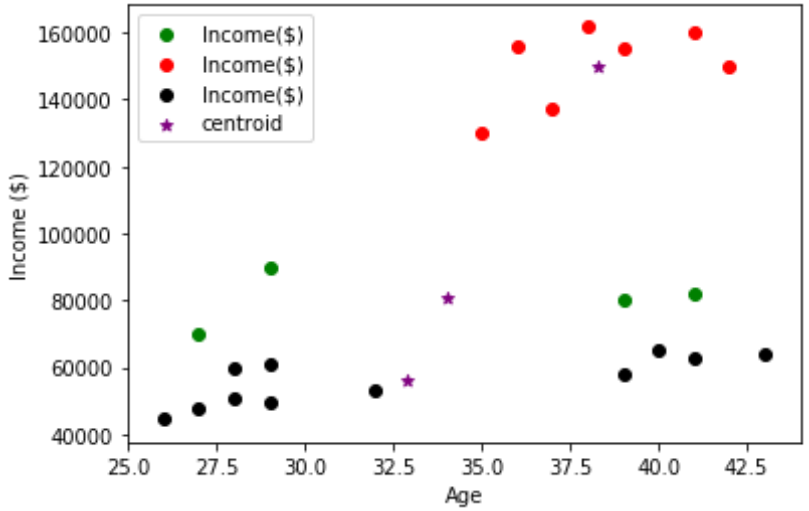
	Name	Age	Income($)	cluster
0	Rob	27	70000	0
1	Michael	29	90000	0
2	Mohan	29	61000	2
3	Ismail	28	60000	2
4	Kory	42	150000	1

In [28]:
```
km.cluster_centers_
```

Out[28]:
```
array([[3.40000000e+01, 8.05000000e+04],
       [3.82857143e+01, 1.50000000e+05],
       [3.29090909e+01, 5.61363636e+04]])
```

In [34]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green',label='Income($)')
plt.scatter(df2.Age,df2['Income($)'],color='red',label='Income($)')
plt.scatter(df3.Age,df3['Income($)'],color='black',label='Income($)')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='pu
rple',marker='*',label='centroid')
plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()
```

Out[34]:
```
<matplotlib.legend.Legend at 0x1cf0737b608>
```



Preprocessing using min max scaler

In [35]:
```
scaler = MinMaxScaler()
```

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

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

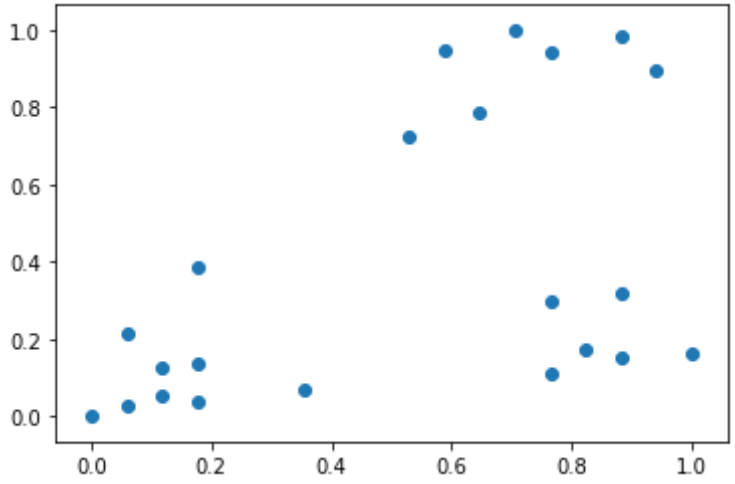
In [36]:
```
df.head()
```

Out[36]:

	Name	Age	Income($)	cluster
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	Kory	0.941176	0.897436	1

In [37]:
```
plt.scatter(df.Age,df['Income($)'])
```

Out[37]:
```
<matplotlib.collections.PathCollection at 0x1cf073f5d48>
```



In [38]:
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

Out[38]:
```
array([0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
1])
```

In [39]:
```
df['cluster']=y_predicted
df.head()
```

Out[39]:

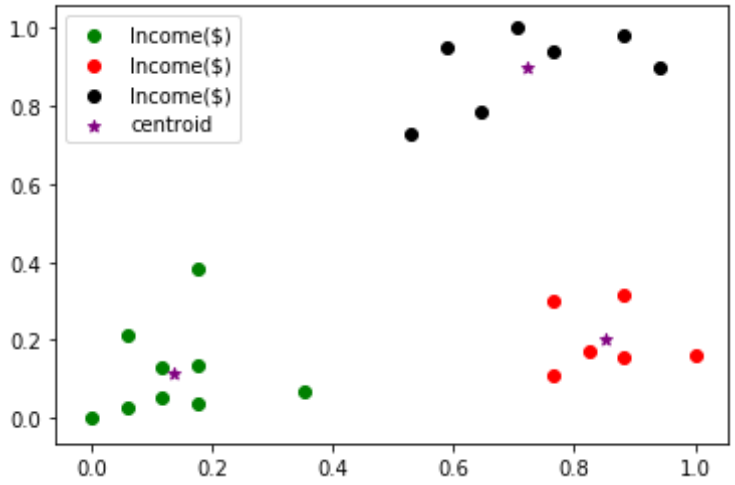
	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	0
1	Michael	0.176471	0.384615	0
2	Mohan	0.176471	0.136752	0
3	Ismail	0.117647	0.128205	0
4	Kory	0.941176	0.897436	2

In [40]:
```
km.cluster_centers_
```

Out[40]:
```
array([[0.1372549 , 0.11633428],
       [0.85294118, 0.2022792 ],
       [0.72268908, 0.8974359 ]])
```

In [42]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green',label='Income($)')
plt.scatter(df2.Age,df2['Income($)'],color='red',label='Income($)')
plt.scatter(df3.Age,df3['Income($)'],color='black',label='Income($)')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='pu
rple',marker='*',label='centroid')
plt.legend()
```

Out[42]:
```
<matplotlib.legend.Legend at 0x1cf074db788>
```



Preprocessing using min max scaler

In [43]:
```
scaler = MinMaxScaler()

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

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

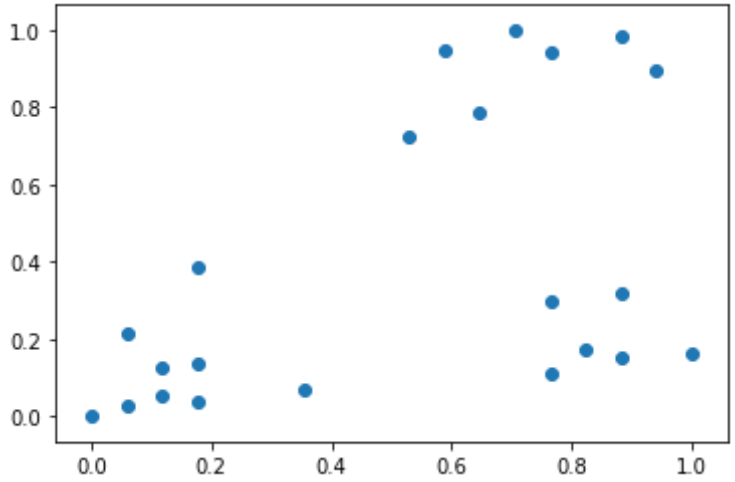
In [44]:
```
df.head()
```

Out[44]:

	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	2

In [45]:
```
plt.scatter(df.Age,df['Income($)'])
```

Out[45]:
```
<matplotlib.collections.PathCollection at 0x1cf08514dc8>
```



In [46]:
```
km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['Age','Income($)']])
y_predicted
```

Out[46]:
```
array([2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0,
0])
```

In [47]:
```
df['cluster']=y_predicted
df.head()
```

Out[47]:

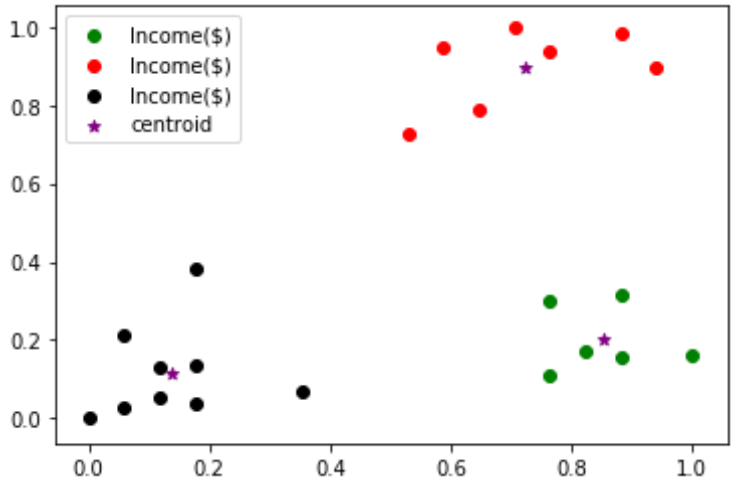
	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	2
1	Michael	0.176471	0.384615	2
2	Mohan	0.176471	0.136752	2
3	Ismail	0.117647	0.128205	2
4	Kory	0.941176	0.897436	1

In [48]:
```
km.cluster_centers_
```

Out[48]:
```
array([[0.85294118, 0.2022792 ],
       [0.72268908, 0.8974359 ],
       [0.1372549 , 0.11633428]])
```

In [50]:
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green',label='Income($)')
plt.scatter(df2.Age,df2['Income($)'],color='red',label='Income($)')
plt.scatter(df3.Age,df3['Income($)'],color='black',label='Income($)')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='pu
rple',marker='*',label='centroid')
plt.legend()
```

Out[50]:
```
<matplotlib.legend.Legend at 0x1cf085d8f88>
```



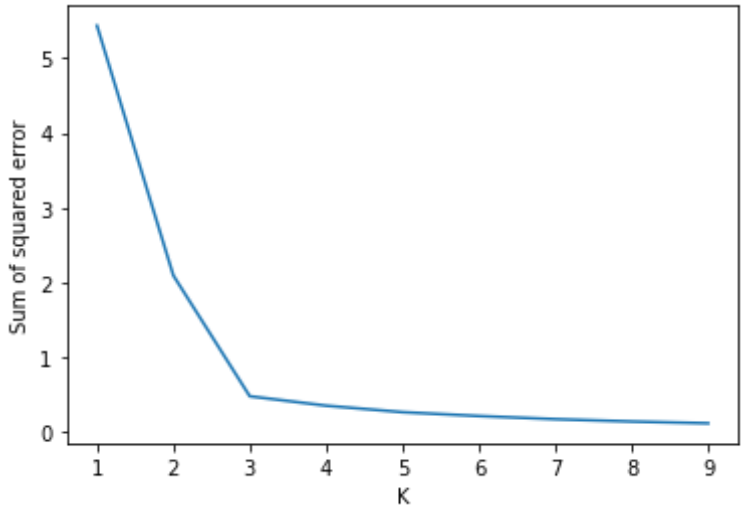
Elbow Plot

In [52]:
```
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 [53]:
```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

Out[53]:
```
[<matplotlib.lines.Line2D at 0x1cf08668808>]
```



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