KMC (income)

December 18, 2022

[1]:

```python
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline
```

[3]:

```python
df = pd.read_csv("income.csv")
df.head()
```

[3]:

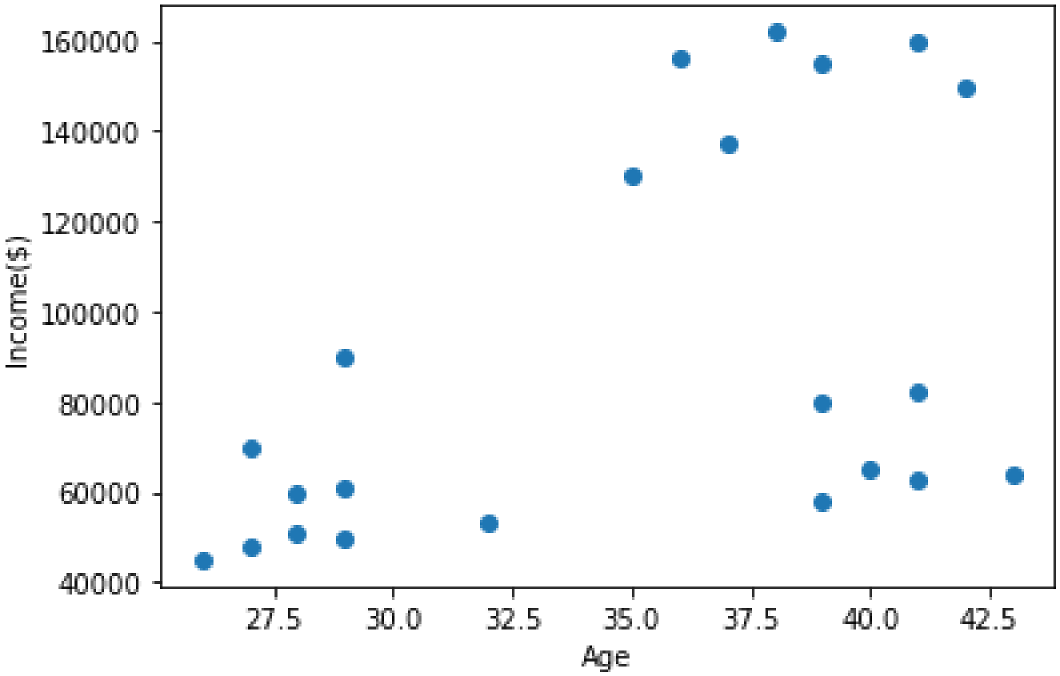
```
      Name  Age  Income($)
0      Rob   27      70000
1  Michael   29      90000
2    Mohan   29      61000
3   Ismail   28      60000
4     Kory   42     150000
```

[4]:

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

[4]:

```
Text(0, 0.5, 'Income($)')
```



[14]:

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

[14]:

```
array([2, 2, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0])
```

[ ]:

```

```

[7]:

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

[7]:

```
      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
```

[8]:

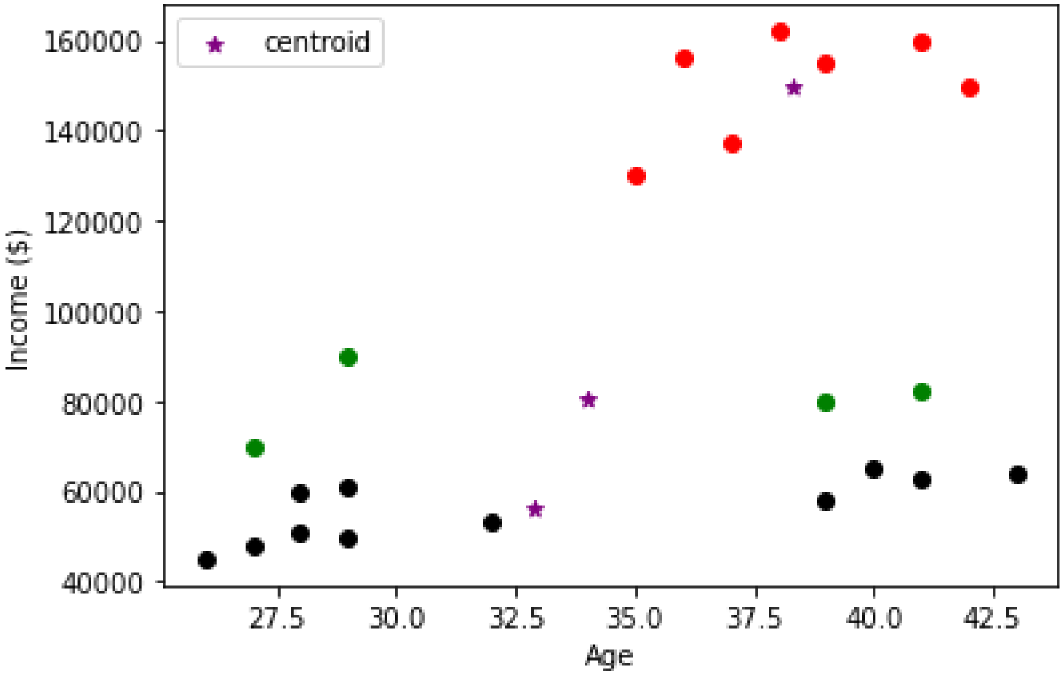
```
km.cluster_centers_
```

[8]:

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

[15]:
```
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($)'],color='black')
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:
 ↪,1],color='purple',marker='*',label='centroid')
plt.xlabel('Age')
plt.ylabel('Income ($)')
plt.legend()
```

[15]:
```
<matplotlib.legend.Legend at 0x11c7eb0>
```



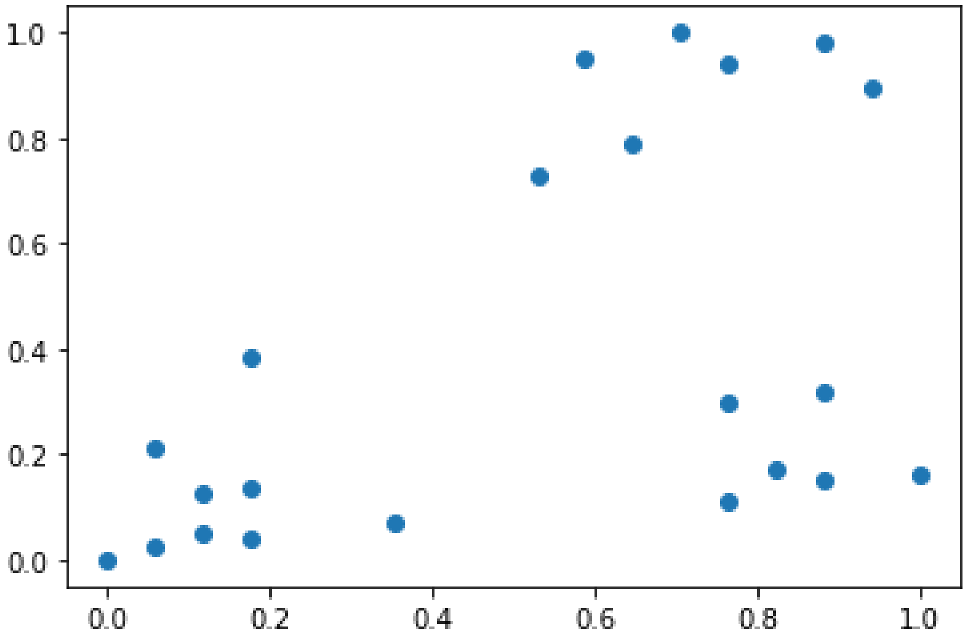
[16]:
```
scaler = MinMaxScaler()

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

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

[17]:
```
df.head()
```

```
[17]:       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
```

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

```
[18]: <matplotlib.collections.PathCollection at 0x1215e98>
```



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

```
[19]: array([0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2])
```

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

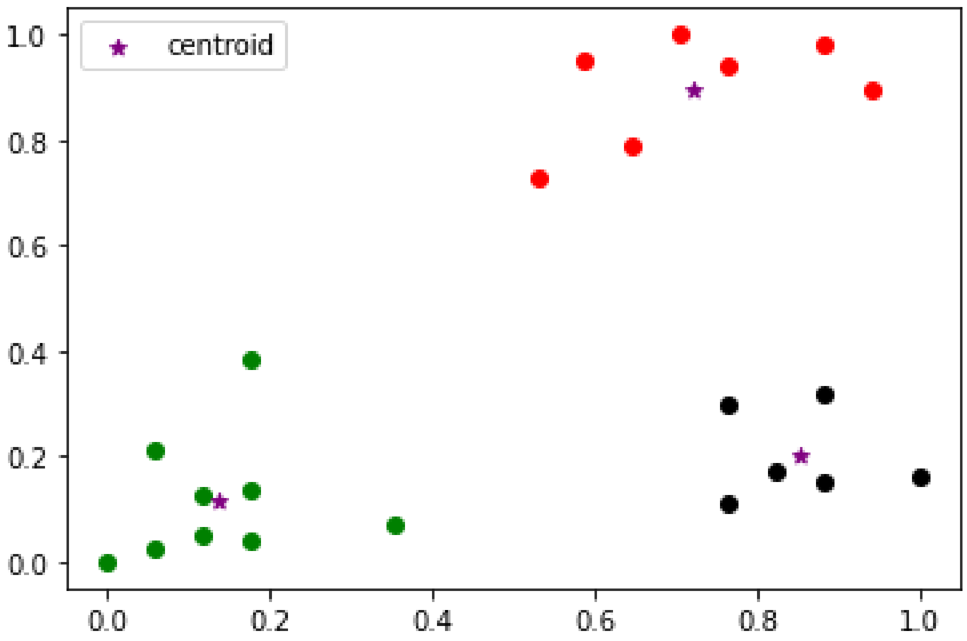
```
[20]:       Name       Age  Income($)  cluster
      0      Rob  0.058824   0.213675        0
      1  Michael  0.176471   0.384615        0
      2    Mohan  0.176471   0.136752        0
      3   Ismail  0.117647   0.128205        0
      4     Kory  0.941176   0.897436        1
```

```
[21]: km.cluster_centers_
```

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

```
[22]: 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',marker='*',label='centroid')
      plt.legend()
```

```
[22]: <matplotlib.legend.Legend at 0x13cd448>
```



```
[23]: 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_)
```

```
C:\Users\Deepak\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881:
```

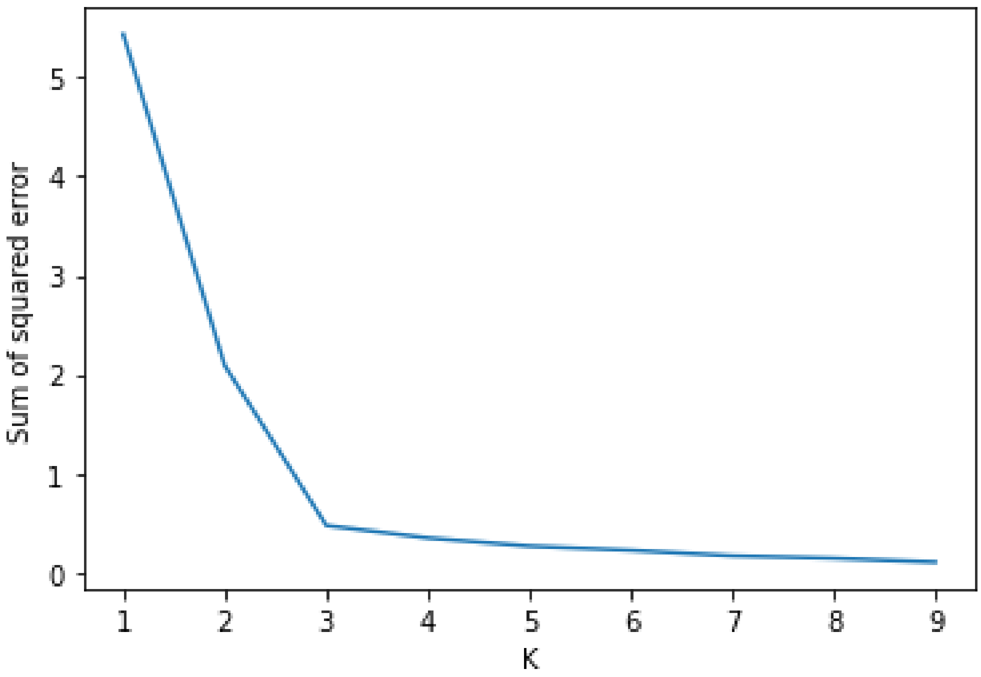
```
UserWarning: KMeans is known to have a memory leak on Windows with MKL, when
there are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[24]:

```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

[24]:

```
[<matplotlib.lines.Line2D at 0x1449250>]
```



[ ]: