Kmeans Clustering

August 19, 2022

1 kmeans Clustering

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
data=pd.read_csv("shopping_data (1).csv")
data.head()
```

[2]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

[3]:
```
finaldata=data.iloc[:,2:5]
finaldata
```

[3]:

	Age	Annual Income (k$)	Spending Score (1-100)
0	19	15	39
1	21	15	81
2	20	16	6
3	23	16	77
4	31	17	40
..	…	…	…
195	35	120	79
196	45	126	28
197	32	126	74
198	32	137	18
199	30	137	83

[200 rows x 3 columns]

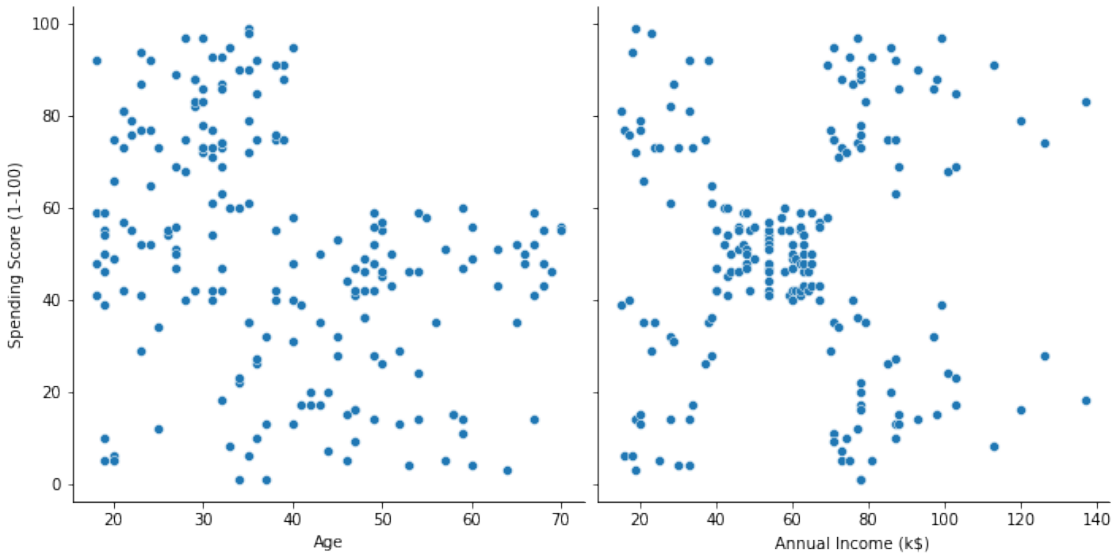
[4]:
```
finaldata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 3 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Age                     200 non-null    int64
 1   Annual Income (k$)      200 non-null    int64
 2   Spending Score (1-100)  200 non-null    int64
dtypes: int64(3)
memory usage: 4.8 KB
```

[5]:
```
sns.pairplot(x_vars=["Age","Annual Income (k$)"],y_vars=["Spending Score␣
↪(1-100)"],data=finaldata,size=5)
```

```
/usr/local/lib/python3.7/site-packages/seaborn/axisgrid.py:2076: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

[5]: <seaborn.axisgrid.PairGrid at 0x7f89ee731c10>



[6]:
```
finaldata=data.iloc[:,3:5]
finaldata
```

[6]:

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

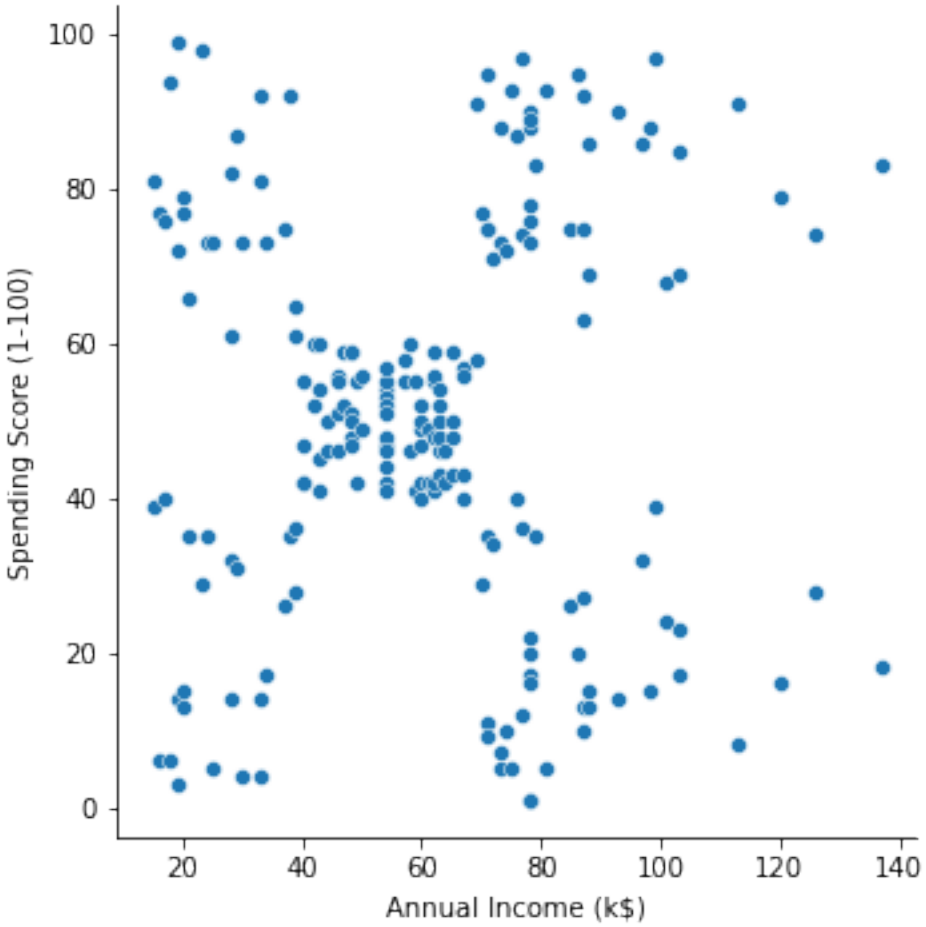
```
..                  ...                 ...
195                 120                  79
196                 126                  28
197                 126                  74
198                 137                  18
199                 137                  83

[200 rows x 2 columns]
```

[7]:
```
sns.pairplot(x_vars=["Annual Income (k$)"],y_vars=["Spending Score␣
↪(1-100)"],data=finaldata,size=5)
```

```
/usr/local/lib/python3.7/site-packages/seaborn/axisgrid.py:2076: UserWarning:
The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

[7]:
```
<seaborn.axisgrid.PairGrid at 0x7f89e8298610>
```



[8]:
```
# Buiding the Kmean model
from sklearn.cluster import KMeans
clust=KMeans(n_clusters=5,random_state=1)
```

[9]:
```
clust.fit_predict(finaldata)
```

[9]:
```
array([4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 2,
       4, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 1, 3, 2, 3, 1, 3, 1, 3,
       2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3], dtype=int32)
```

[10]:
```
clust.labels_ #gives the grouping details
```

[10]:
```
array([4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 2,
       4, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 1, 3, 2, 3, 1, 3, 1, 3,
       2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3], dtype=int32)
```

[11]:
```
# check the centroid
print("the centroid is ")
print(clust.cluster_centers_)
```

```
the centroid is
[[25.72727273 79.36363636]
 [88.2        17.11428571]
 [55.2962963  49.51851852]
 [86.53846154 82.12820513]
 [26.30434783 20.91304348]]
```

[12]:
```
finaldata["clusterlabel"]=clust.labels_
finaldata
```

[12]:

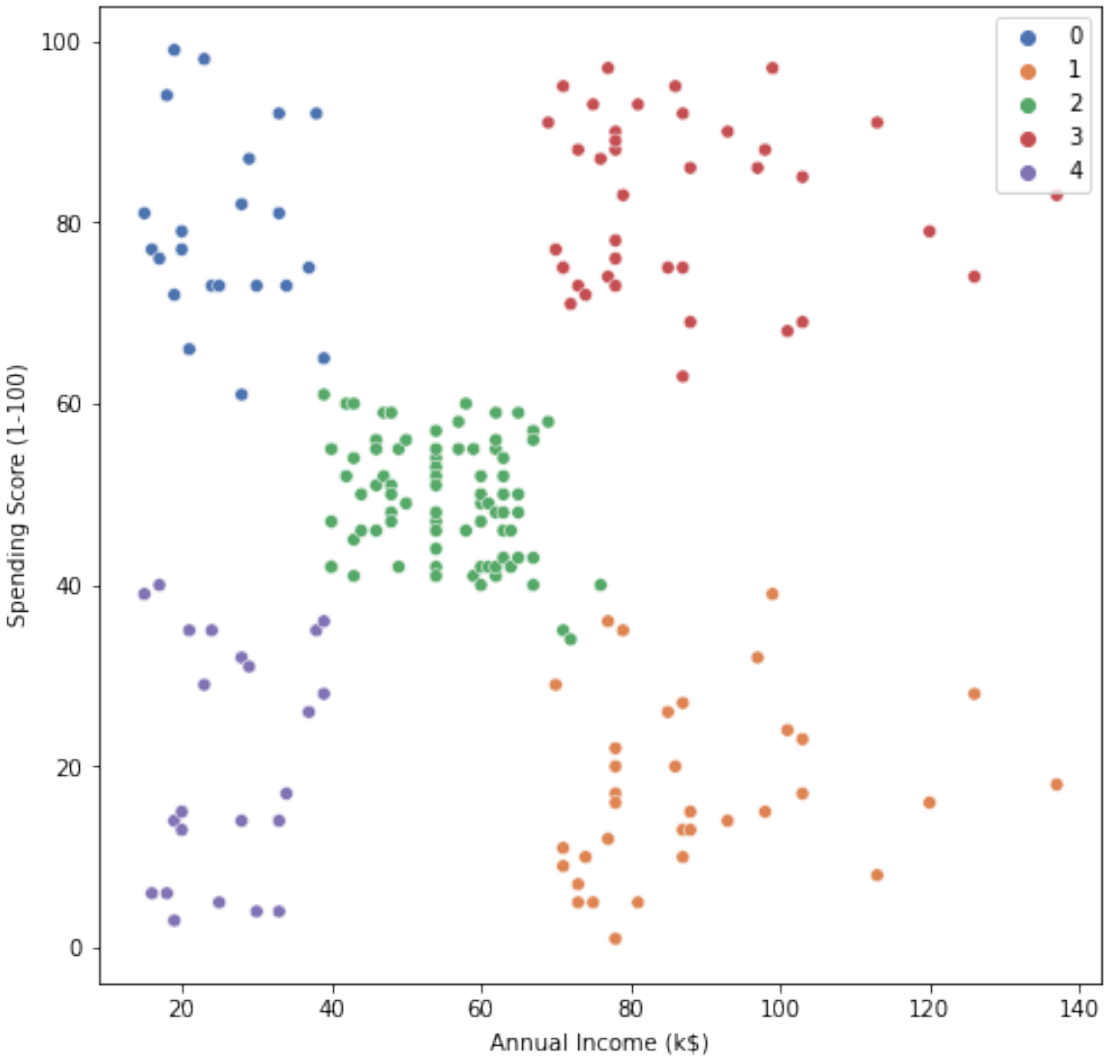
	Annual Income (k$)	Spending Score (1-100)	clusterlabel
0	15	39	4
1	15	81	0

```
2                   16                   6            4
3                   16                  77            0
4                   17                  40            4
..                  …                   …            …
195                 120                 79            3
196                 126                 28            1
197                 126                 74            3
198                 137                 18            1
199                 137                 83            3

[200 rows x 3 columns]
```

[13]:
```
# check the plot with the three groups
plt.figure(figsize=(8,8))
sns.scatterplot(x=finaldata["Annual Income (k$)"],y=finaldata["Spending Score␣
↪(1-100)"],hue=clust.labels_,palette="deep")#to check plot with respect to␣
↪the group labels
#sns.scatterplot(x=finaldata.iloc[:,0],y=finaldata.iloc[:,1],hue=clust.
↪labels_,palette="deep")
```

[13]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



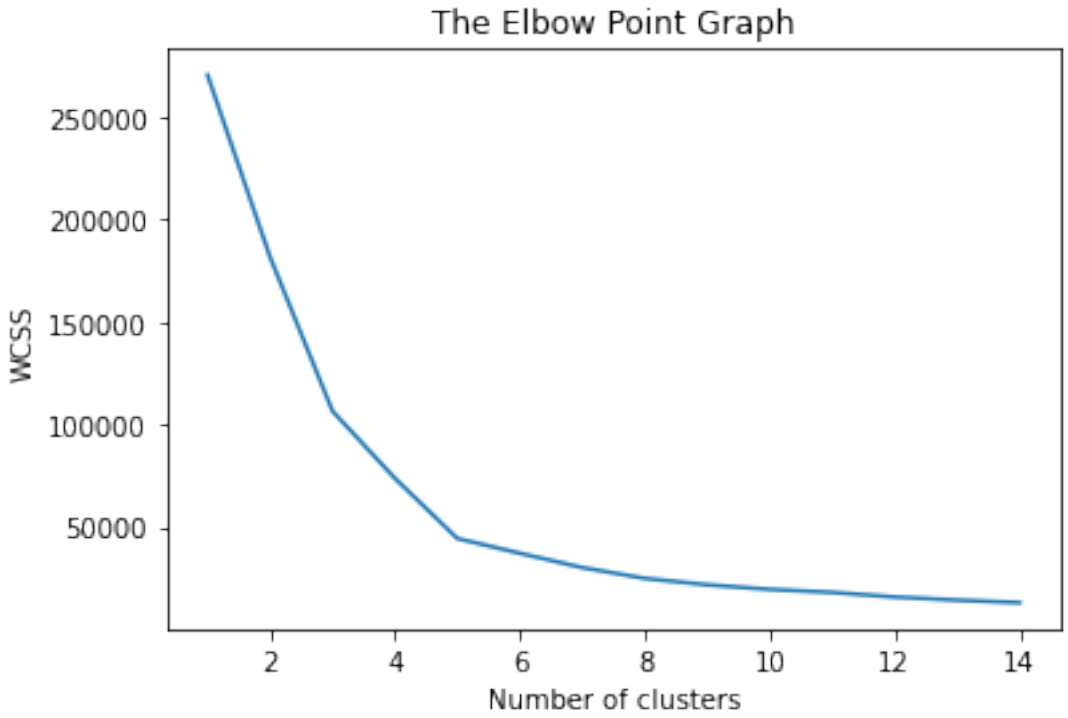
[14]:
```
finaldata["clusterlabel"].value_counts()
```

[14]:
```
2    81
3    39
1    35
4    23
0    22
Name: clusterlabel, dtype: int64
```

[15]:
```
#finding WSS
wcss = []
for i in range(1,15):
    kmeans1 = KMeans(n_clusters = i,random_state=10)
    kmeans1.fit(finaldata)
```

```
   #appending the WCSS to the list (kmeans.inertia_ returns the WCSS value for␣
 ↪an initialized cluster)
    wcss.append(kmeans1.inertia_)


#Plotting The Elbow graph
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Point Graph')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



[16]:
```
from sklearn.metrics import silhouette_score
sc=silhouette_score(finaldata,clust.labels_,metric='euclidean')
sc
```

[16]: 0.5544129978559397

[17]:
```
#help(silhouette_score)
```

[ ]:
```

```