Hello this is @Ranjeet_Kumbhar, Enjoy the Notebook

GitHub:https://github.com/RanjeetKumbhar01/TE_IT_ML_ASSIGNMENTS_SPPU

Question

Assignment on Clustering Techniques Download the following customer dataset from below link: Data Set: https://www.kaggle.com/shwetabh123/mall-customers This dataset givesthe data of Income and money spent by the customers visiting a Shopping Mall. The data set contains Customer ID, Gender, Age, Annual Income, Spending Score. Therefore, as a mall owner you need to find the group of people who are the profitable customers for the mall owner. Apply at least two clustering algorithms (based on Spending Score) to find the group of customers.

a. Apply Data pre-processing (Label Encoding , Data Transformation….) techniques if necessary.

b. Perform data-preparation(Train-Test Split)

c. Apply Machine Learning Algorithm

d. Evaluate Model.

e. Apply Cross-Validation and Evaluate Model

```
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

```
/kaggle/input/mall-customers/Mall_Customers.csv
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('../input/mall-customers/Mall_Customers.csv')

df
```

```
     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
..          ...     ...  ...                 ...                    .
..
195         196  Female   35                 120
79
196         197  Female   45                 126
28
197         198    Male   32                 126
74
198         199    Male   32                 137
18
199         200    Male   30                 137
83

[200 rows x 5 columns]
```

```
x = df.iloc[:,3:]
x
```

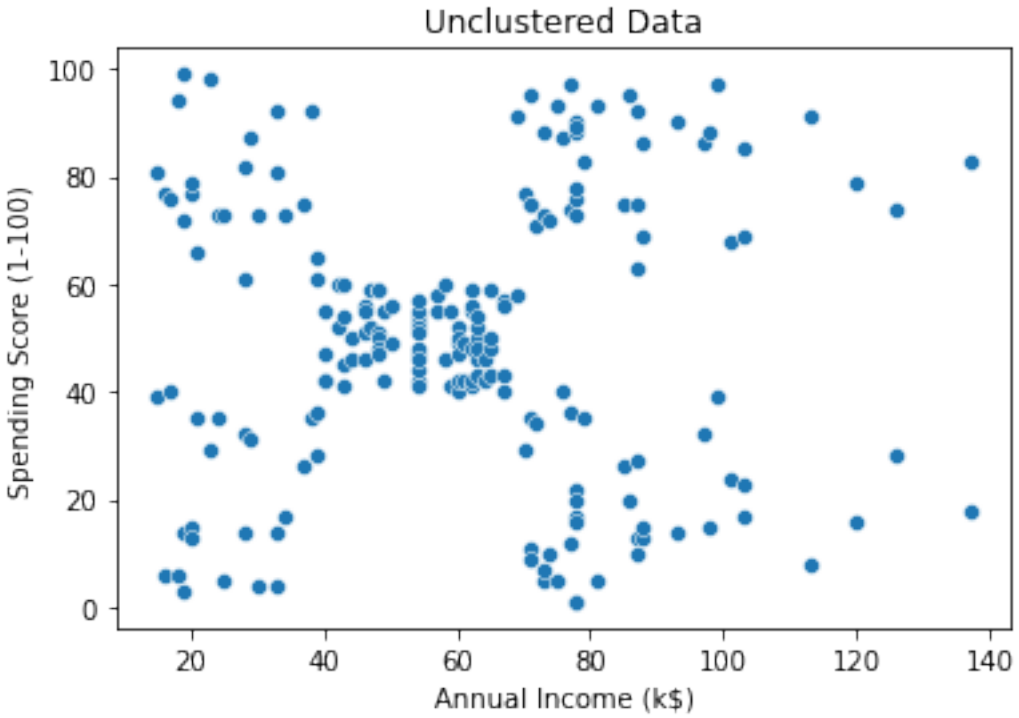
```
     Annual Income (k$)  Spending Score (1-100)
0                    15                      39
1                    15                      81
2                    16                       6
3                    16                      77
4                    17                      40
..                  ...                     ...
195                 120                      79
196                 126                      28
197                 126                      74
```

```
198                 137                 18
199                 137                 83

[200 rows x 2 columns]
```

You can split data using train_test_split

```
plt.title('Unclustered Data')
sns.scatterplot(x=x['Annual Income (k$)'],y=x['Spending Score (1-
100)'])
```

```
<AxesSubplot:title={'center':'Unclustered Data'}, xlabel='Annual
Income (k$)', ylabel='Spending Score (1-100)'>
```




```
from sklearn.cluster import KMeans, AgglomerativeClustering
```

AgglomerativeClustering is hierarchical Clustering

```
km = KMeans(n_clusters=4)

km.fit_predict(x)
```

```
array([2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
1,
       2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
1,
```

```
       2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 0, 3, 0, 3, 0, 3, 0,
3,
       0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
3,
       0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
3,
       0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
3,
       0, 3], dtype=int32)
```

```
#sse
km.inertia_
```

```
73679.78903948834
```

```
sse =[]
for k in range(1,16):
    km = KMeans(n_clusters=k)
    km.fit_predict(x)
    sse.append(km.inertia_)
```

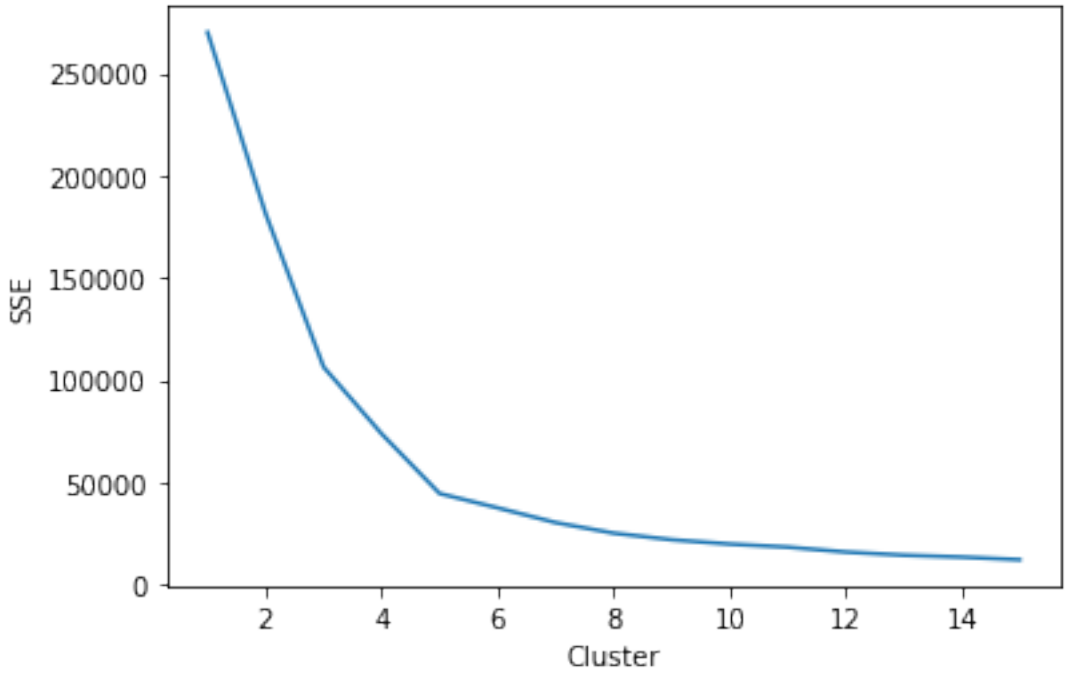
```
sse
```

```
[269981.28,
 181363.59595959596,
 106348.37306211119,
 73679.78903948834,
 44448.45544793371,
 37442.24745037571,
 30241.343617936585,
 25062.433792653785,
 21850.165282585636,
 19740.010370359305,
 18248.58456228341,
 15845.619372815674,
 14292.543823365124,
 13374.273322189787,
 12087.99592074592]
```

Elbow Method

```
sns.lineplot(range(1,16),y = sse)
plt.xlabel('Cluster')
plt.ylabel('SSE')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning

Text(0, 0.5, 'SSE')
```



So at 5th cluster

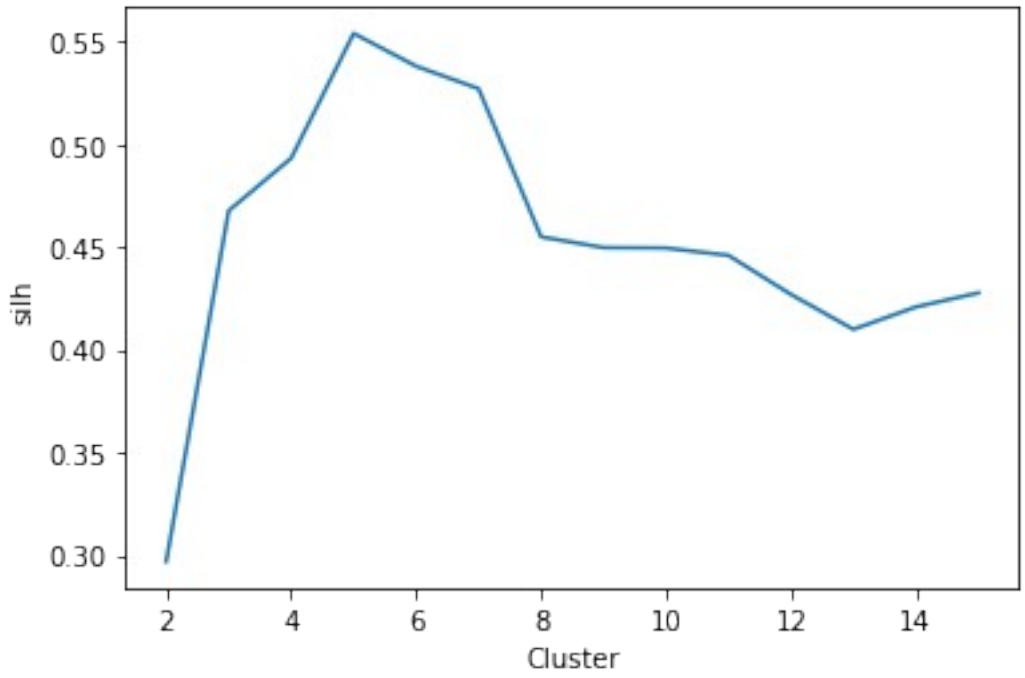
```
#Method second or alternative for elbow method
from sklearn.metrics import silhouette_score

silh =[]
for k in range(2,16):
    km = KMeans(n_clusters=k)
    labels = km.fit_predict(x)
    score = silhouette_score(x, labels)
    silh.append(score)

silh

[0.2968969162503008,
 0.46761358158775435,
 0.4931963109249047,
 0.553931997444648,
 0.5379675585622219,
 0.5270287298101395,
```

```
 0.45492755850983463,
 0.44966289417722194,
 0.4494755585987857,
 0.44588401462331617,
 0.4269031451249127,
 0.4099045071135656,
 0.4207994576595669,
 0.42764015819358164]
```

```
sns.lineplot(range(2,16),y = silh)
plt.xlabel('Cluster')
plt.ylabel('silh')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
```

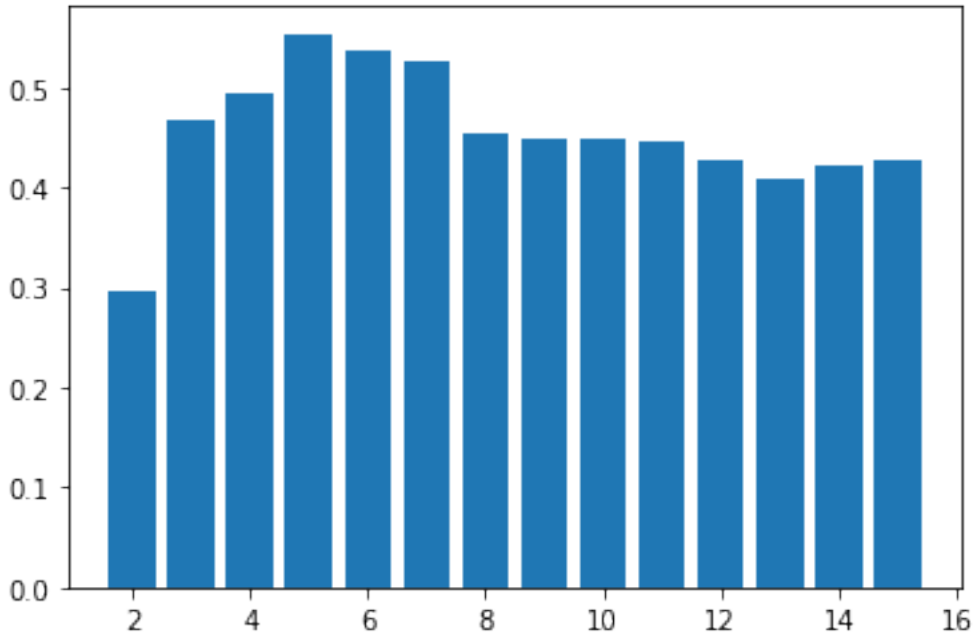
```
Text(0, 0.5, 'silh')
```



same at 5th cluster we are getting highest silhouette_score, this is efficient cluster

```
plt.bar(range(2,16,1),silh)
```

```
<BarContainer object of 14 artists>
```




```
km = KMeans(n_clusters=5,random_state=1)

labels = km.fit_predict(x)

km.labels_
```

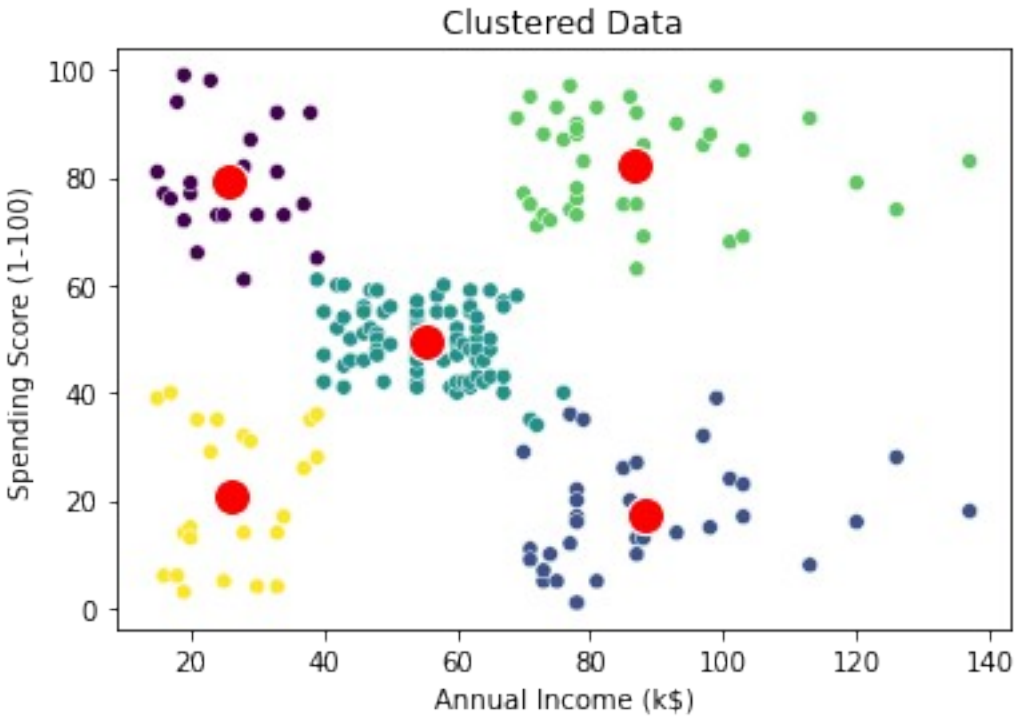
```
array([4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4,
0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4,
2,
       4, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2,
       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, 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)
```

```
cent = km.cluster_centers_

plt.title('Clustered Data')
sns.scatterplot(x=x['Annual Income (k$)'],y=x['Spending Score (1-
```

```
100)'],c=labels )
sns.scatterplot(cent[:,0],cent[:,1], s=200, color='red')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

```
<AxesSubplot:title={'center':'Clustered Data'}, xlabel='Annual Income
(k$)', ylabel='Spending Score (1-100)'>
```




```
df[labels==0]
```

```
    CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-
100)
1            2    Male   21                  15
81
3            4  Female   23                  16
77
5            6  Female   22                  17
76
7            8  Female   23                  18
94
9           10  Female   30                  19
72
```

```
11          12  Female   35                  19
99
13          14  Female   24                  20
77
15          16    Male   22                  20
79
17          18    Male   20                  21
66
19          20  Female   35                  23
98
21          22    Male   25                  24
73
23          24    Male   31                  25
73
25          26    Male   29                  28
82
27          28    Male   35                  28
61
29          30  Female   23                  29
87
31          32  Female   21                  30
73
33          34    Male   18                  33
92
35          36  Female   21                  33
81
37          38  Female   30                  34
73
39          40  Female   20                  37
75
41          42    Male   24                  38
92
45          46  Female   24                  39
65
```

```
agl = AgglomerativeClustering(n_clusters=5)
```

```
alabels = agl.fit_predict(x)
```

```
alabels
```

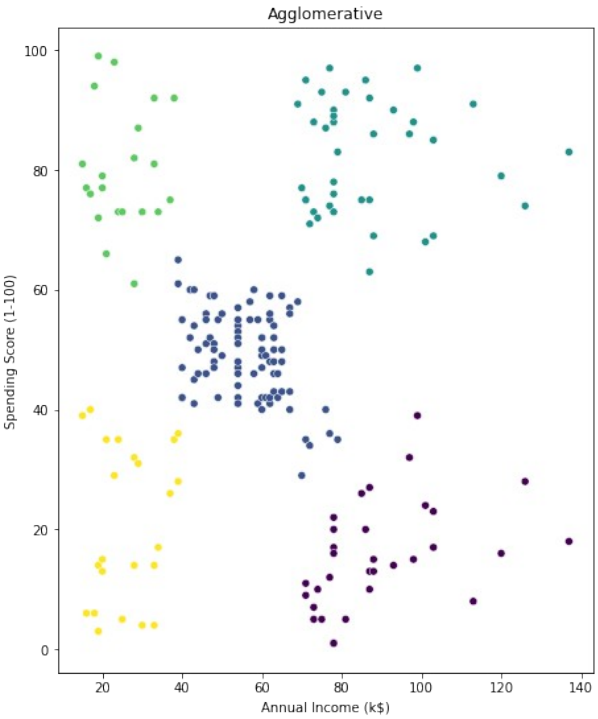
```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4,
3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4,
1,
       4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

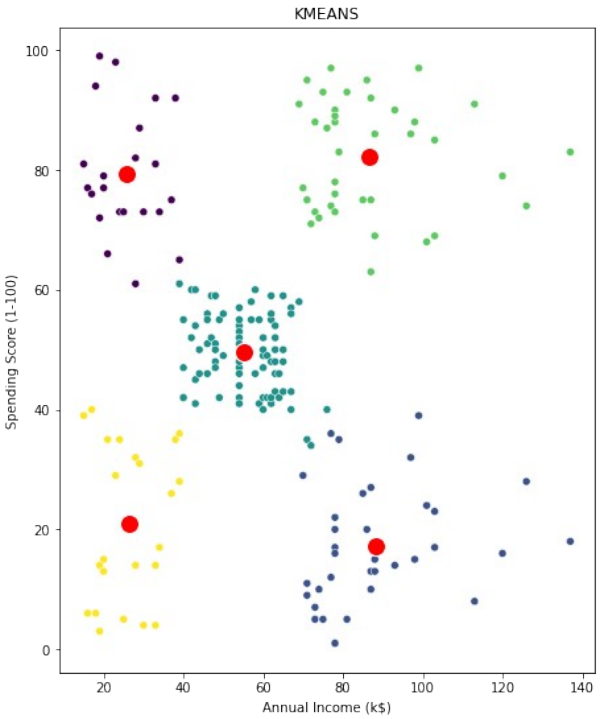
```
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2, 0, 2, 0,
2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0,
2,
       0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
2,
       0, 2])
```

```
plt.figure(figsize=(16,9))
plt.subplot(1,2,1)
plt.title('Agglomerative')
sns.scatterplot(x=x['Annual Income (k$)'],y=x['Spending Score (1-
100)'], c= alabels)

plt.subplot(1,2,2)
plt.title('KMEANS')
sns.scatterplot(x=x['Annual Income (k$)'],y=x['Spending Score (1-
100)'],c=labels )
sns.scatterplot(cent[:,0],cent[:,1], s=200, color='red')
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  FutureWarning
```

```
<AxesSubplot:title={'center':'KMEANS'}, xlabel='Annual Income (k$)',
ylabel='Spending Score (1-100)'>
```


Agglomerative
Spending Score (1-100)
Annual Income (k$)
0
20
40
60
80
100
120
140


KMEANS
Spending Score (1-100)
Annual Income (k$)
0
20
40
60
80
100
120
140