DMW_Assign2

September 2, 2020

[2]:
```
!pwd
```

```
/home/sumit/DMW
```

[3]:
```
import numpy as np
import pandas as pd
from pandas import plotting

import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')

from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
import scipy.cluster.hierarchy as sch
```

[4]:
```
data = pd.read_csv('~/Downloads/Mall_Customers.csv')
print(data)
```

```
     CustomerID  Gender  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]
```

[5]:
```
data.isnull().any().any()
```

[5]:
```
False
```

[6]:
```python
x = data.iloc[:, [3, 4]].values
print(x)
```

```
[[ 15  39]
 [ 15  81]
 [ 16   6]
 [ 16  77]
 [ 17  40]
 [ 17  76]
 [ 18   6]
 [ 18  94]
 [ 19   3]
 [ 19  72]
 [ 19  14]
 [ 19  99]
 [ 20  15]
 [ 20  77]
 [ 20  13]
 [ 20  79]
 [ 21  35]
 [ 21  66]
 [ 23  29]
 [ 23  98]
 [ 24  35]
 [ 24  73]
 [ 25   5]
 [ 25  73]
 [ 28  14]
 [ 28  82]
 [ 28  32]
 [ 28  61]
 [ 29  31]
 [ 29  87]
 [ 30   4]
 [ 30  73]
 [ 33   4]
 [ 33  92]
 [ 33  14]
 [ 33  81]
 [ 34  17]
 [ 34  73]
 [ 37  26]
 [ 37  75]
 [ 38  35]
 [ 38  92]
 [ 39  36]
 [ 39  61]
```

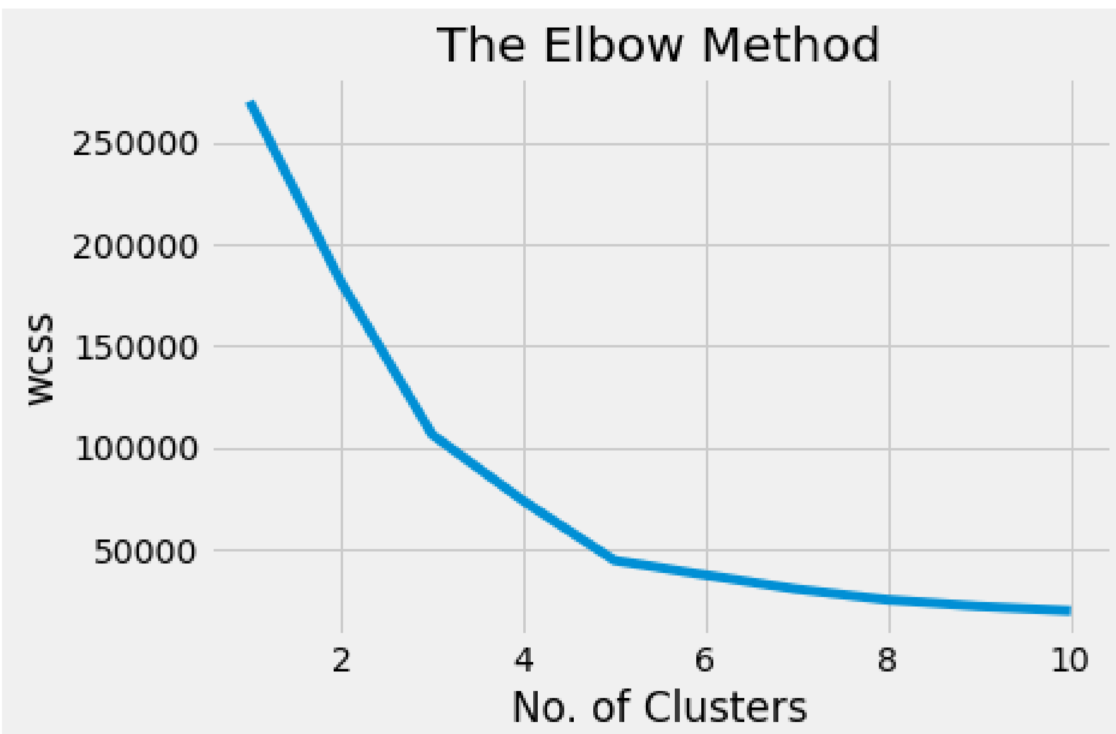
```
[ 39  28]
[ 39  65]
[ 40  55]
[ 40  47]
[ 40  42]
[ 40  42]
[ 42  52]
[ 42  60]
[ 43  54]
[ 43  60]
[ 43  45]
[ 43  41]
[ 44  50]
[ 44  46]
[ 46  51]
[ 46  46]
[ 46  56]
[ 46  55]
[ 47  52]
[ 47  59]
[ 48  51]
[ 48  59]
[ 48  50]
[ 48  48]
[ 48  59]
[ 48  47]
[ 49  55]
[ 49  42]
[ 50  49]
[ 50  56]
[ 54  47]
[ 54  54]
[ 54  53]
[ 54  48]
[ 54  52]
[ 54  42]
[ 54  51]
[ 54  55]
[ 54  41]
[ 54  44]
[ 54  57]
[ 54  46]
[ 57  58]
[ 57  55]
[ 58  60]
[ 58  46]
[ 59  55]
[ 59  41]
```

```
[ 60  49]
[ 60  40]
[ 60  42]
[ 60  52]
[ 60  47]
[ 60  50]
[ 61  42]
[ 61  49]
[ 62  41]
[ 62  48]
[ 62  59]
[ 62  55]
[ 62  56]
[ 62  42]
[ 63  50]
[ 63  46]
[ 63  43]
[ 63  48]
[ 63  52]
[ 63  54]
[ 64  42]
[ 64  46]
[ 65  48]
[ 65  50]
[ 65  43]
[ 65  59]
[ 67  43]
[ 67  57]
[ 67  56]
[ 67  40]
[ 69  58]
[ 69  91]
[ 70  29]
[ 70  77]
[ 71  35]
[ 71  95]
[ 71  11]
[ 71  75]
[ 71   9]
[ 71  75]
[ 72  34]
[ 72  71]
[ 73   5]
[ 73  88]
[ 73   7]
[ 73  73]
[ 74  10]
[ 74  72]
```

```
[ 75   5]
[ 75  93]
[ 76  40]
[ 76  87]
[ 77  12]
[ 77  97]
[ 77  36]
[ 77  74]
[ 78  22]
[ 78  90]
[ 78  17]
[ 78  88]
[ 78  20]
[ 78  76]
[ 78  16]
[ 78  89]
[ 78   1]
[ 78  78]
[ 78   1]
[ 78  73]
[ 79  35]
[ 79  83]
[ 81   5]
[ 81  93]
[ 85  26]
[ 85  75]
[ 86  20]
[ 86  95]
[ 87  27]
[ 87  63]
[ 87  13]
[ 87  75]
[ 87  10]
[ 87  92]
[ 88  13]
[ 88  86]
[ 88  15]
[ 88  69]
[ 93  14]
[ 93  90]
[ 97  32]
[ 97  86]
[ 98  15]
[ 98  88]
[ 99  39]
[ 99  97]
[101  24]
[101  68]
```

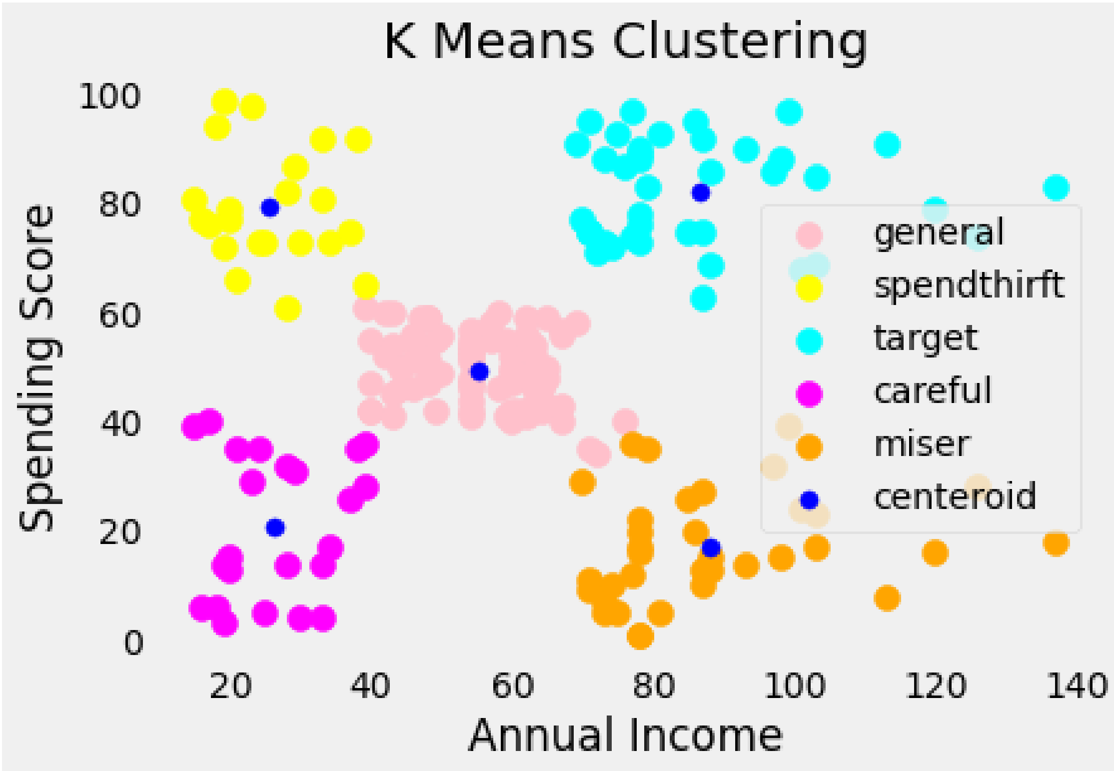
```
 [103  17]
 [103  85]
 [103  23]
 [103  69]
 [113   8]
 [113  91]
 [120  16]
 [120  79]
 [126  28]
 [126  74]
 [137  18]
 [137  83]]
```

[7]:
```
wcss = []
for i in range(1, 11):
    km = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init =␣
 ↪10, random_state = 0)
    km.fit(x)
    wcss.append(km.inertia_)

plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method', fontsize = 20)
plt.xlabel('No. of Clusters')
plt.ylabel('wcss')
plt.show()
```



[8]:
```
km = KMeans(n_clusters = 5, init = 'k-means++', max_iter = 300, n_init = 10,␣
↪random_state = 0)
y_means = km.fit_predict(x)

# print(x[y_means == 0, 0], x[y_means == 0, 1])
print(km.cluster_centers_)
plt.scatter(x[y_means == 0, 0], x[y_means == 0, 1], s = 100, c = 'pink', label␣
↪= 'general')
plt.scatter(x[y_means == 1, 0], x[y_means == 1, 1], s = 100, c = 'yellow',␣
↪label = 'spendthirft')
plt.scatter(x[y_means == 2, 0], x[y_means == 2, 1], s = 100, c = 'cyan', label␣
↪= 'target')
plt.scatter(x[y_means == 3, 0], x[y_means == 3, 1], s = 100, c = 'magenta',␣
↪label = 'careful')
plt.scatter(x[y_means == 4, 0], x[y_means == 4, 1], s = 100, c = 'orange',␣
↪label = 'miser')
plt.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:, 1], s = 50, c =␣
↪'blue' , label = 'centeroid')

# plt.style.use('fivethirtyeight')
plt.title('K Means Clustering', fontsize = 20)
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.legend()
plt.grid()
plt.show()
```

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




```
[9]: y = data.iloc[:, [2, 4]].values
     print(y)
```

```
[[19 39]
 [21 81]
 [20  6]
 [23 77]
 [31 40]
 [22 76]
 [35  6]
 [23 94]
 [64  3]
 [30 72]
 [67 14]
 [35 99]
 [58 15]
 [24 77]
 [37 13]
 [22 79]
 [35 35]
 [20 66]
 [52 29]
 [35 98]
```

```
[35 35]
[25 73]
[46  5]
[31 73]
[54 14]
[29 82]
[45 32]
[35 61]
[40 31]
[23 87]
[60  4]
[21 73]
[53  4]
[18 92]
[49 14]
[21 81]
[42 17]
[30 73]
[36 26]
[20 75]
[65 35]
[24 92]
[48 36]
[31 61]
[49 28]
[24 65]
[50 55]
[27 47]
[29 42]
[31 42]
[49 52]
[33 60]
[31 54]
[59 60]
[50 45]
[47 41]
[51 50]
[69 46]
[27 51]
[53 46]
[70 56]
[19 55]
[67 52]
[54 59]
[63 51]
[18 59]
[43 50]
[68 48]
```

```
[19 59]
[32 47]
[70 55]
[47 42]
[60 49]
[60 56]
[59 47]
[26 54]
[45 53]
[40 48]
[23 52]
[49 42]
[57 51]
[38 55]
[67 41]
[46 44]
[21 57]
[48 46]
[55 58]
[22 55]
[34 60]
[50 46]
[68 55]
[18 41]
[48 49]
[40 40]
[32 42]
[24 52]
[47 47]
[27 50]
[48 42]
[20 49]
[23 41]
[49 48]
[67 59]
[26 55]
[49 56]
[21 42]
[66 50]
[54 46]
[68 43]
[66 48]
[65 52]
[19 54]
[38 42]
[19 46]
[18 48]
[19 50]
```

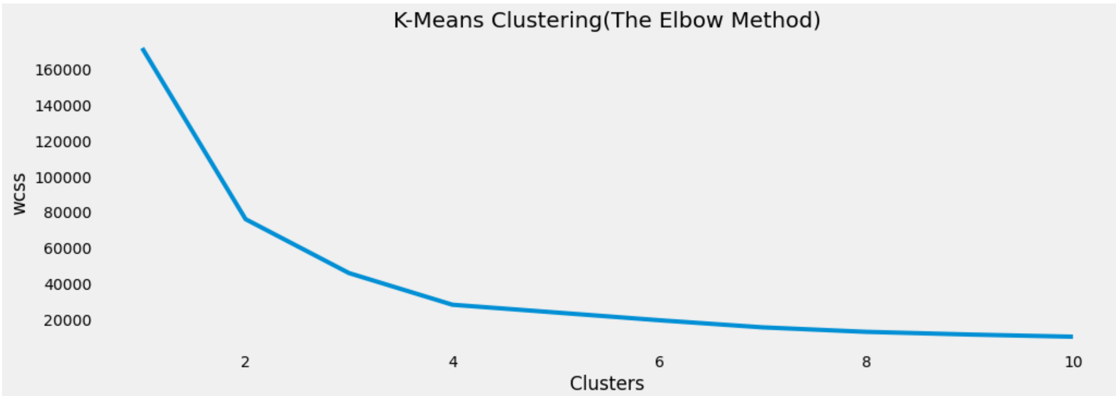
```
[63 43]
[49 59]
[51 43]
[50 57]
[27 56]
[38 40]
[40 58]
[39 91]
[23 29]
[31 77]
[43 35]
[40 95]
[59 11]
[38 75]
[47  9]
[39 75]
[25 34]
[31 71]
[20  5]
[29 88]
[44  7]
[32 73]
[19 10]
[35 72]
[57  5]
[32 93]
[28 40]
[32 87]
[25 12]
[28 97]
[48 36]
[32 74]
[34 22]
[34 90]
[43 17]
[39 88]
[44 20]
[38 76]
[47 16]
[27 89]
[37  1]
[30 78]
[34  1]
[30 73]
[56 35]
[29 83]
[19  5]
[31 93]
```

```
 [50 26]
 [36 75]
 [42 20]
 [33 95]
 [36 27]
 [32 63]
 [40 13]
 [28 75]
 [36 10]
 [36 92]
 [52 13]
 [30 86]
 [58 15]
 [27 69]
 [59 14]
 [35 90]
 [37 32]
 [32 86]
 [46 15]
 [29 88]
 [41 39]
 [30 97]
 [54 24]
 [28 68]
 [41 17]
 [36 85]
 [34 23]
 [32 69]
 [33  8]
 [38 91]
 [47 16]
 [35 79]
 [45 28]
 [32 74]
 [32 18]
 [30 83]]
```

[10]:

```
inertia = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', max_iter = 300, n_init␣
 ↪= 10, random_state = 0)
    kmeans.fit(y)
    inertia.append(kmeans.inertia_)

plt.rcParams['figure.figsize'] = (15, 5)
plt.plot(range(1, 11), inertia)
plt.title('K-Means Clustering(The Elbow Method)', fontsize = 20)
```

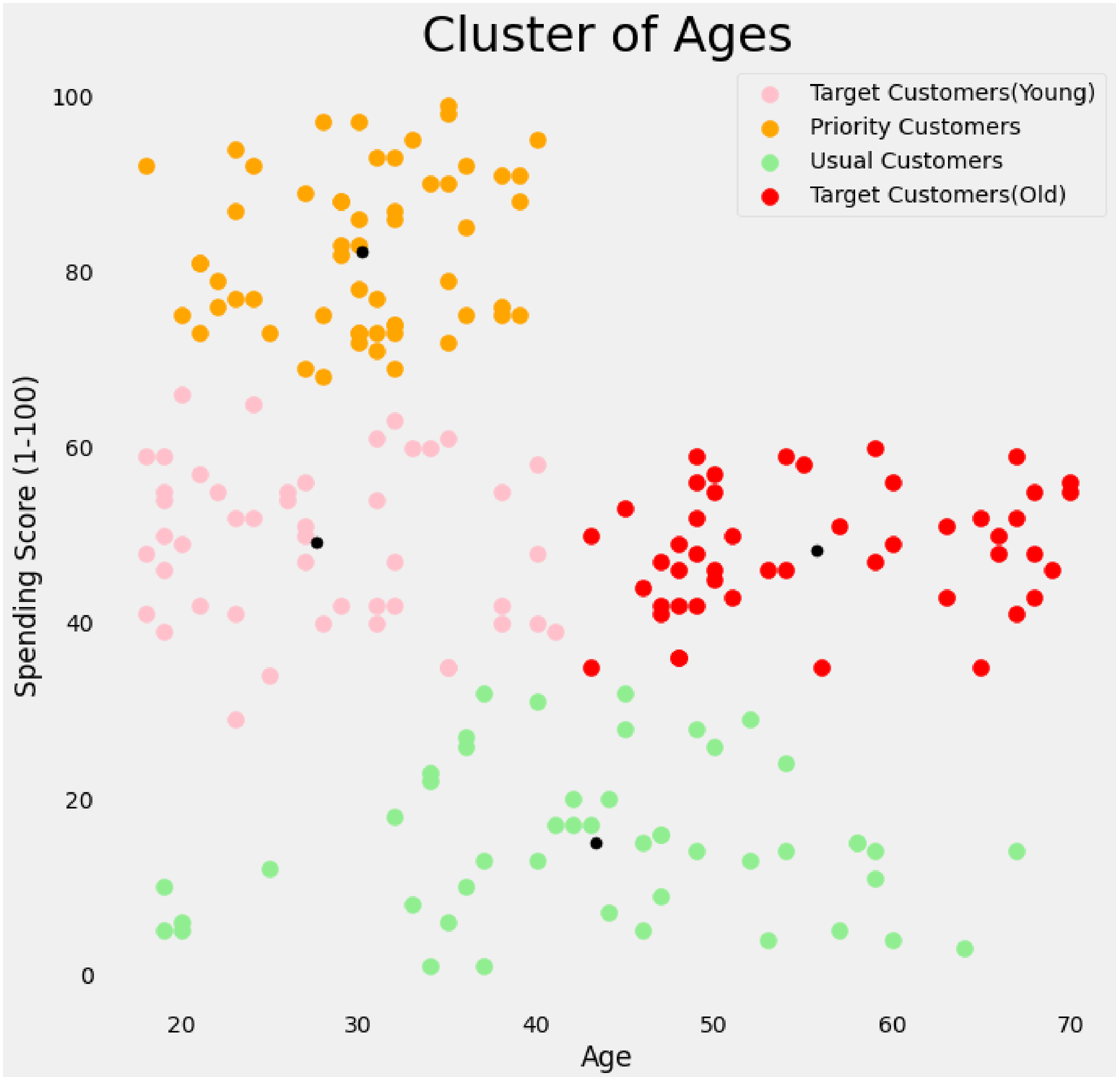
```
plt.xlabel('Clusters')
plt.ylabel('wcss')
plt.grid()
plt.show()
```



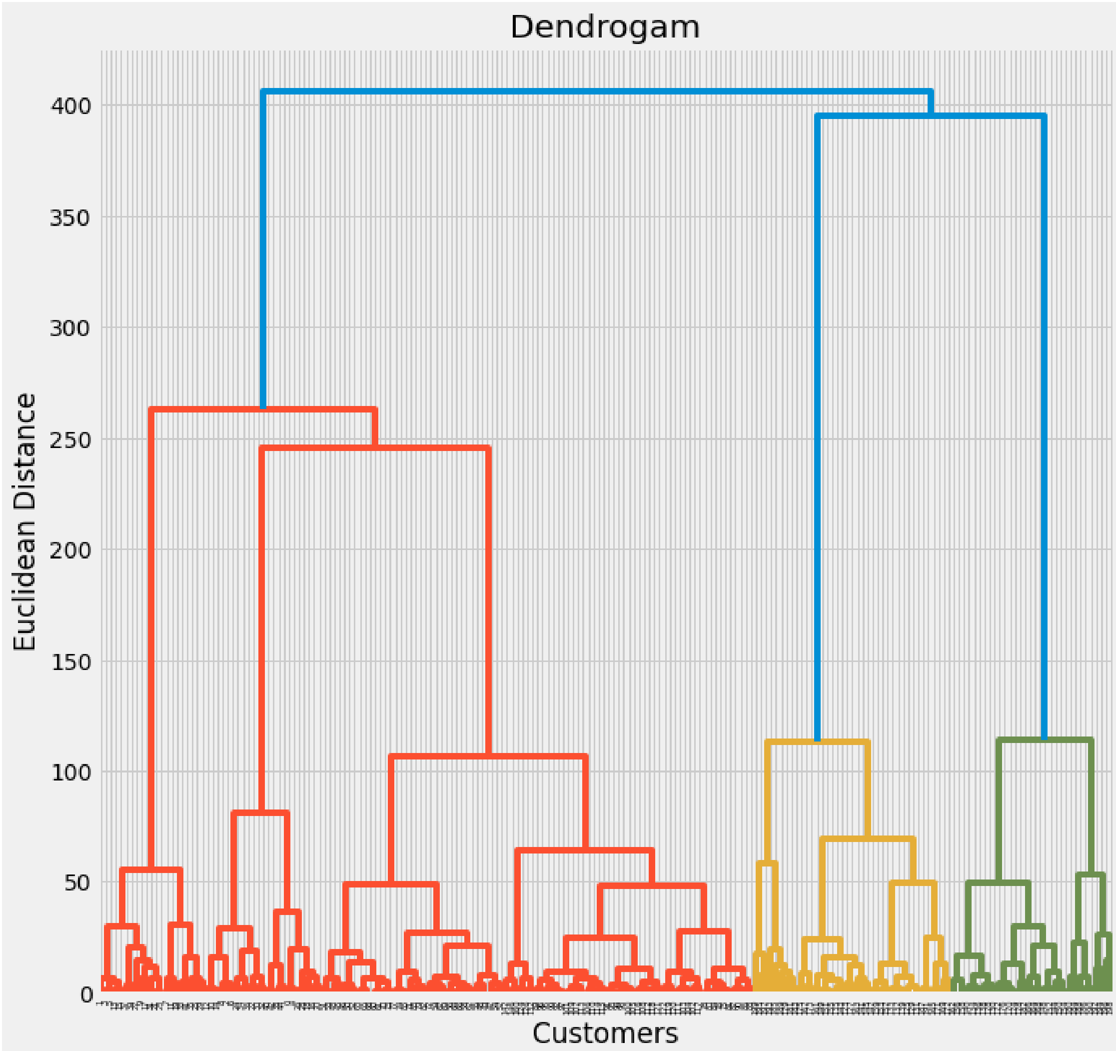
[11]:
```
kmeans = KMeans(n_clusters = 4, init = 'k-means++', max_iter = 300, n_init =␣
↪10, random_state = 0)
ymeans = kmeans.fit_predict(y)


print(kmeans.cluster_centers_)
plt.rcParams['figure.figsize'] = (10, 10)
plt.title('Cluster of Ages', fontsize = 30)

plt.scatter(y[ymeans == 0, 0], y[ymeans == 0, 1], s = 100, c = 'pink', label =␣
↪'Target Customers(Young)' )
plt.scatter(y[ymeans == 1, 0], y[ymeans == 1, 1], s = 100, c = 'orange', label␣
↪= 'Priority Customers')
plt.scatter(y[ymeans == 2, 0], y[ymeans == 2, 1], s = 100, c = 'lightgreen',␣
↪label = 'Usual Customers')
plt.scatter(y[ymeans == 3, 0], y[ymeans == 3, 1], s = 100, c = 'red', label =␣
↪'Target Customers(Old)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s =␣
↪50, c = 'black')

# plt.style.use('fivethirtyeight')
plt.xlabel('Age')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.grid()
plt.show()
```

```
[[27.61702128 49.14893617]
```

```
[30.1754386  82.35087719]
[43.29166667 15.02083333]
[55.70833333 48.22916667]]
```

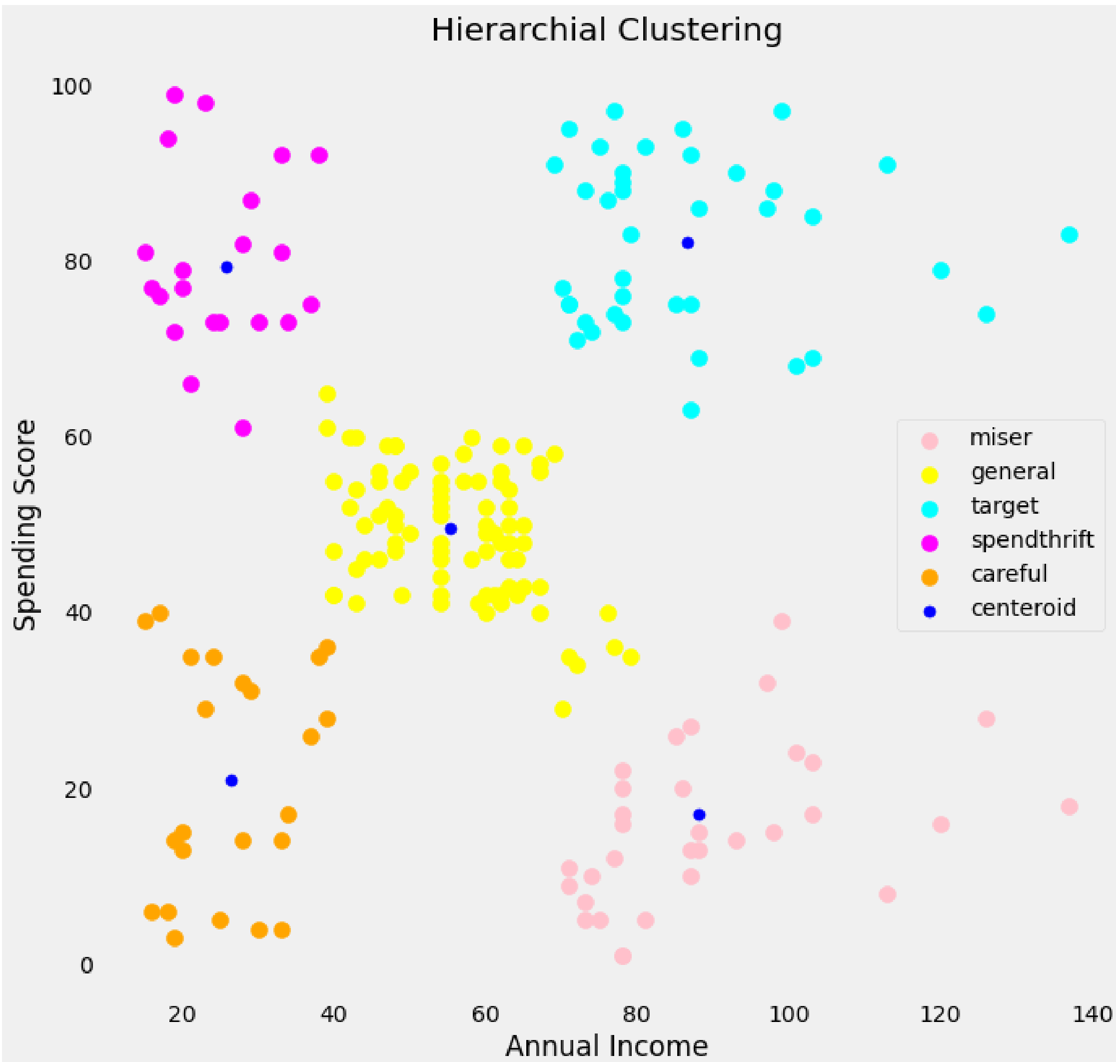


[13]:
```
dendrogram = sch.dendrogram(sch.linkage(x, method = 'ward'))
plt.title('Dendrogam', fontsize = 20)
plt.xlabel('Customers')
plt.ylabel('Euclidean Distance')
plt.show()
```



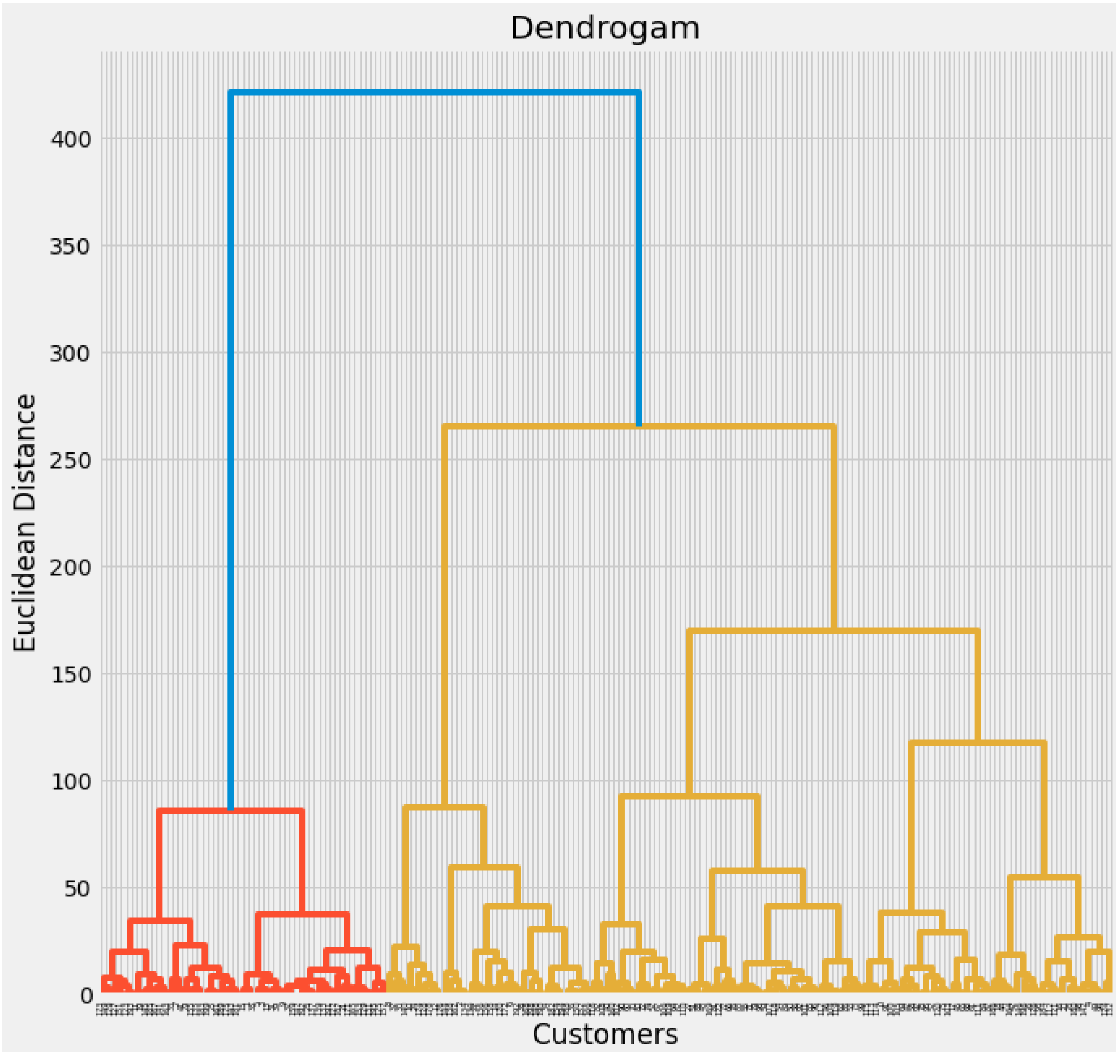
[53]:
```
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage =␣
 ↪'ward')
y_hc = hc.fit_predict(x)

plt.scatter(x[y_hc == 0, 0], x[y_hc == 0, 1], s = 100, c = 'pink', label =␣
 ↪'miser')
plt.scatter(x[y_hc == 1, 0], x[y_hc == 1, 1], s = 100, c = 'yellow', label =␣
 ↪'general')
plt.scatter(x[y_hc == 2, 0], x[y_hc == 2, 1], s = 100, c = 'cyan', label =␣
 ↪'target')
plt.scatter(x[y_hc == 3, 0], x[y_hc == 3, 1], s = 100, c = 'magenta', label =␣
 ↪'spendthrift')
plt.scatter(x[y_hc == 4, 0], x[y_hc == 4, 1], s = 100, c = 'orange', label =␣
 ↪'careful')
```

```
plt.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:, 1], s = 50, c =␣
↪'blue' , label = 'centeroid')

plt.style.use('fivethirtyeight')
plt.title('Hierarchial Clustering', fontsize = 20)
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.legend()
plt.grid()
plt.show()
```



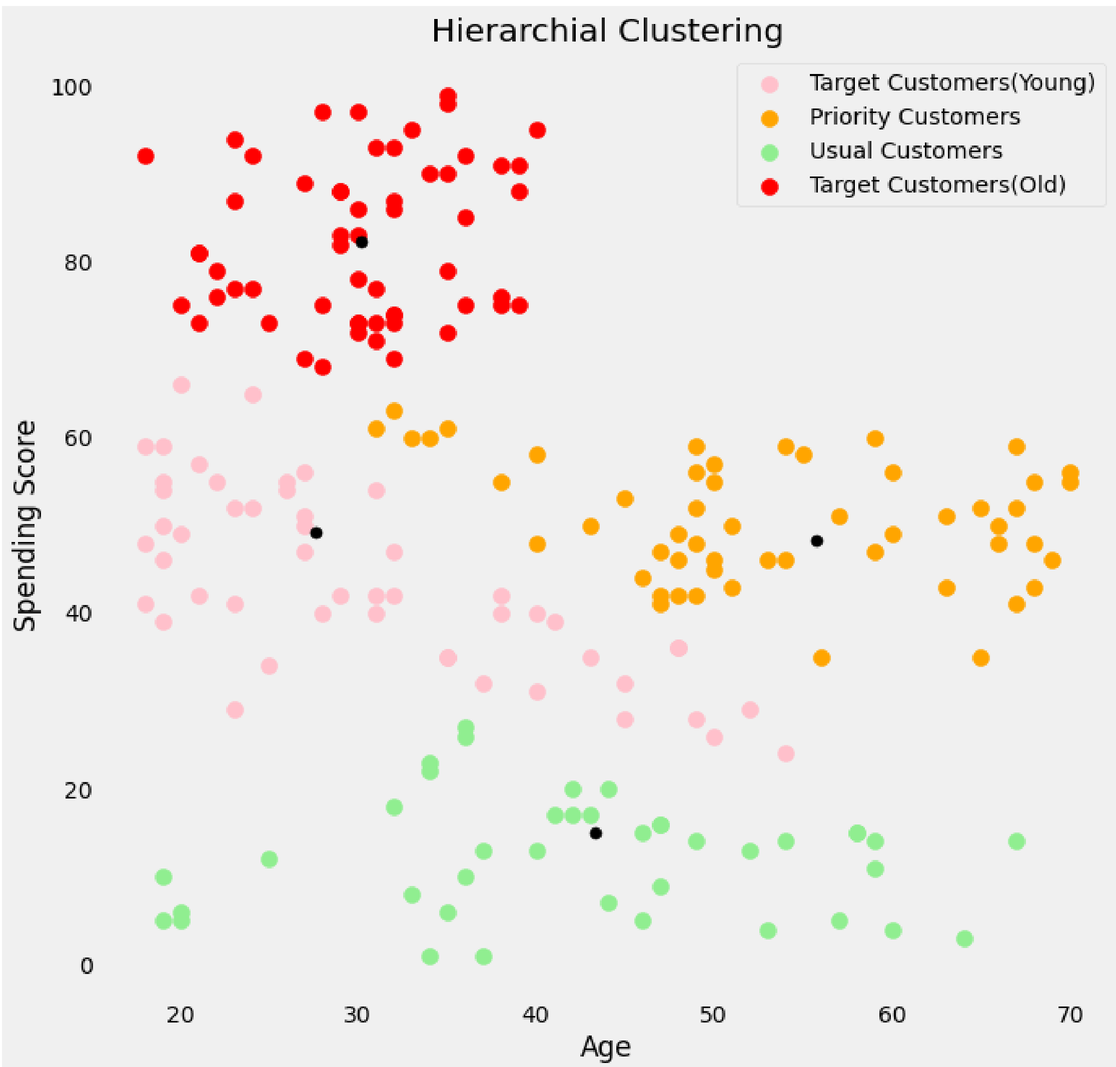
[14]:
```
dendrogram = sch.dendrogram(sch.linkage(y, method = 'ward'))
plt.title('Dendrogam', fontsize = 20)
plt.xlabel('Customers')
```

```
plt.ylabel('Euclidean Distance')
plt.show()
```



[16]:
```
hc = AgglomerativeClustering(n_clusters = 4, affinity = 'euclidean', linkage =␣
 ↪'ward')
y_hc = hc.fit_predict(y)

plt.scatter(y[y_hc == 0, 0], y[y_hc == 0, 1], s = 100, c = 'pink', label =␣
 ↪'Target Customers(Young)' )
plt.scatter(y[y_hc == 1, 0], y[y_hc == 1, 1], s = 100, c = 'orange', label =␣
 ↪'Priority Customers')
plt.scatter(y[y_hc == 2, 0], y[y_hc == 2, 1], s = 100, c = 'lightgreen', label␣
 ↪= 'Usual Customers')
```

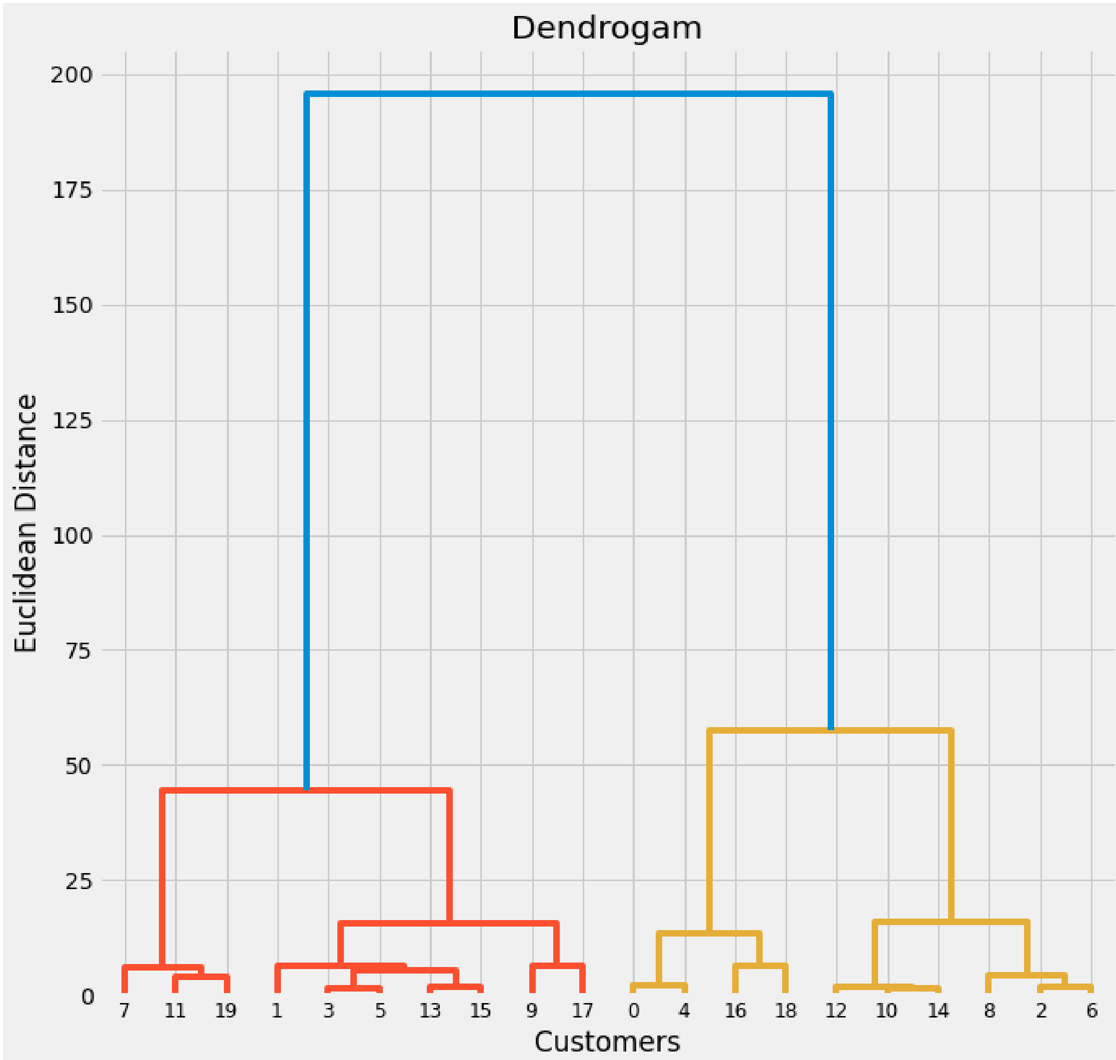
```
plt.scatter(y[y_hc == 3, 0], y[y_hc == 3, 1], s = 100, c = 'red', label =␣
↪'Target Customers(Old)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s =␣
↪50, c = 'black')

# plt.scatter(km.cluster_centers_[:,0], km.cluster_centers_[:, 1], s = 50, c =␣
↪'blue' , label = 'centeroid')

plt.style.use('fivethirtyeight')
plt.title('Hierarchial Clustering', fontsize = 20)
plt.xlabel('Age')
plt.ylabel('Spending Score')
plt.legend()
plt.grid()
plt.show()
```



[17]:
```
x = data.iloc[:20, [3, 4]].values
print(x)
```

```
[[15 39]
 [15 81]
 [16  6]
 [16 77]
 [17 40]
 [17 76]
 [18  6]
 [18 94]
 [19  3]
 [19 72]
 [19 14]
 [19 99]
 [20 15]
 [20 77]
 [20 13]
 [20 79]
 [21 35]
 [21 66]
 [23 29]
 [23 98]]
```

[18]:
```
dendrogram = sch.dendrogram(sch.linkage(x, method = 'ward'))
plt.title('Dendrogam', fontsize = 20)
plt.xlabel('Customers')
plt.ylabel('Euclidean Distance')
plt.show()
```


Dendrogam
Euclidean Distance
0
25
50
75
100
125
150
175
200
7 11 19 1 3 5 13 15 9 17 0 4 16 18 12 10 14 8 2 6
Customers

```
[ ]:
```