

nrcm-hierarchical-clustering-1

August 28, 2023

#Hierarchical_Clustering

##Name:K.Saivinay ##Roll No:21X05A6731 ##Branche:Data Science Final Year
##College:Narsimha Reddy Engineering College

#Project Title: ##Analysis and prediction of “Mall_Customer.csv” file of American Mall market called as Phonic Mall, find out on the basis of clients requirements of dendograms using Scipy graphics library with the help of “scipy.cluster.hierarchy”, to ace the No. of linkage as a clustering to predict.

#Problem Statement: ##The American Finance clients as per the rate of GDP of 2011 found as highest number of growth in their business market. ##As a Data Science Engineer find out which hierarchy cluster gives maximum linkage in upcoming future.

#Task-1: ##With the help of scipy library import the libraries and import the dataset #task-2: ##Using the Dendrogram to find the optimal number of clusters. #Task-3: ##Create a hierarchy model and visualize the cluster with the help of matplotlib library.

```
[3]: #Import the numpy, pandas , matplotlib, seaborn library's
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[7]: #Assign variable name "dataset" and the input variable as "X" including select
      ↪ all the row and index columns which you want [column_index, Column_index].
data = pd.read_csv("/content/Mall_Customers.csv")
data.head()
```

```
[7]:
```

	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

```
[28]: #Assign variable name "dataset" and the input variable as "X" including select
      ↪ all the row and index columns which you want [column_index, Column_index].
X = data.iloc[:,[3,4]].values
X
```

```
[28]: array([[ 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],

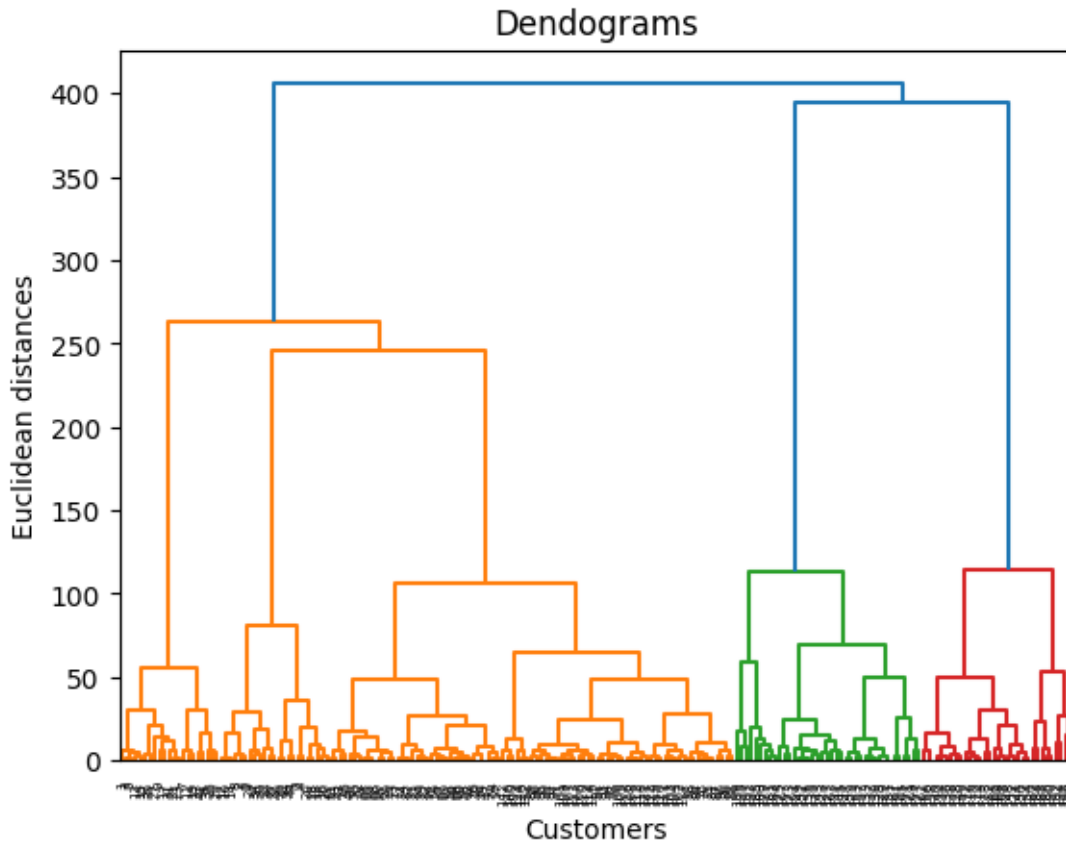
[75, 93],
[76, 40],
[76, 87],
[77, 12],
[77, 97],
[77, 36],
[77, 74],
[78, 22],
[78, 90],
[78, 17],
[78, 88],
[78, 20],
[78, 76],
[78, 16],
[78, 89],
[78, 1],
[78, 78],
[78, 1],
[78, 73],
[79, 35],
[79, 83],
[81, 5],
[81, 93],
[85, 26],
[85, 75],
[86, 20],
[86, 95],
[87, 27],
[87, 63],
[87, 13],
[87, 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]])
```

```
[29]: #import scipy cluster using attribute "scipy.cluster.hierarchy" as sch alias
import scipy.cluster.hierarchy as sch
```

```
[30]: #Using the dendrogram to find the optimal number of clusters
# Assign a variable as dendograph and declers the "sch.dendrogram(sch.
↳linkage(X, method = 'ward'))"
dendograph=sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title("Dendograms")
plt.xlabel("Customers")
plt.ylabel("Euclidean distances")
```

```
[30]: Text(0, 0.5, 'Euclidean distances')
```



```
[35]: # from "sklearn.cluster" attribute import "AgglomerativeClustering" default_
      ↪ argument.
```

```
from sklearn.cluster import AgglomerativeClustering
```

```
[36]: #Create a cluster for five or nth cluster which you want.
```

```
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage =_
      ↪ 'ward')
```

```
y_hc = hc.fit_predict(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_agglomerative.py:983:
FutureWarning: Attribute `affinity` was deprecated in version 1.2 and will be
removed in 1.4. Use `metric` instead
```

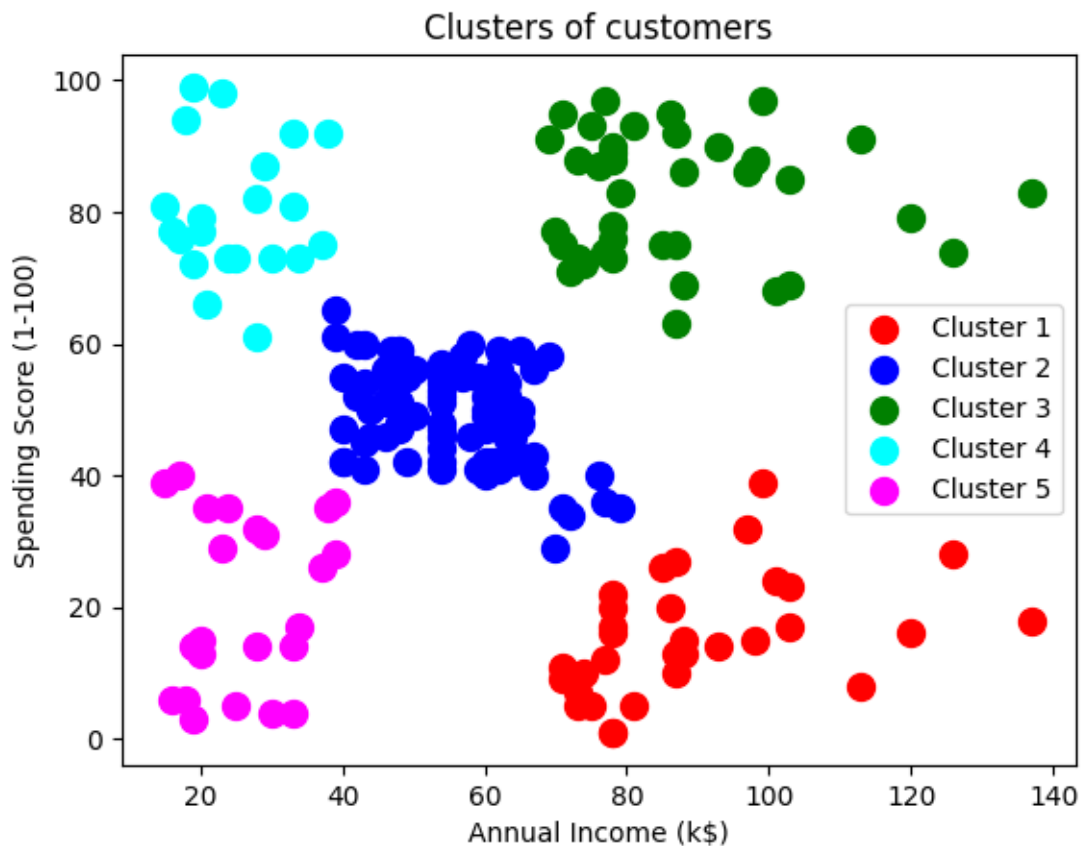
```
warnings.warn(
```

```
[37]: #Plot the scatter plot for scatter visualization.
```

```
plt.scatter(X[y_hc ==0,0], X[y_hc ==0,1], s = 100, c = 'red', label = 'Cluster_
      ↪ 1')
```

```
plt.scatter(X[y_hc ==1,0 ], X[y_hc ==1,1 ], s = 100, c = 'blue', label =_
      ↪ 'Cluster 2')
```

```
plt.scatter(X[y_hc == 2,0], X[y_hc ==2,1 ], s = 100, c = 'green', label =_
↳'Cluster 3')
plt.scatter(X[y_hc ==3,0 ], X[y_hc ==3,1 ], s = 100, c = 'cyan', label =_
↳'Cluster 4')
plt.scatter(X[y_hc ==4,0 ], X[y_hc ==4,1 ], s = 100, c = 'magenta', label =_
↳'Cluster 5')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



#Conclusion: ##According to the model building as a engineer my prediction is Cluster-3 as give highest number of linkage.

[]: