

# hierarchical-clustering-4

August 28, 2023

###CH.Mahesh #####21X05A6712 #####CSE-DS #####NRCM

##**Project Title:-** #####Analysis and prediction of “Mall\_Customers” of american mall market called PHONIX Mall.To find out on the basics requirements of dendrogram using scipy graphics library with the help of “scipy.cluster.hierarchy”,to ace the no\_of linkage of a clustering to predict.

## 0.0.1 Problem statement:-

**The American Finance Market Clients as per the rate of GDP of 2011 who has highest no\_of growth in their business market.**

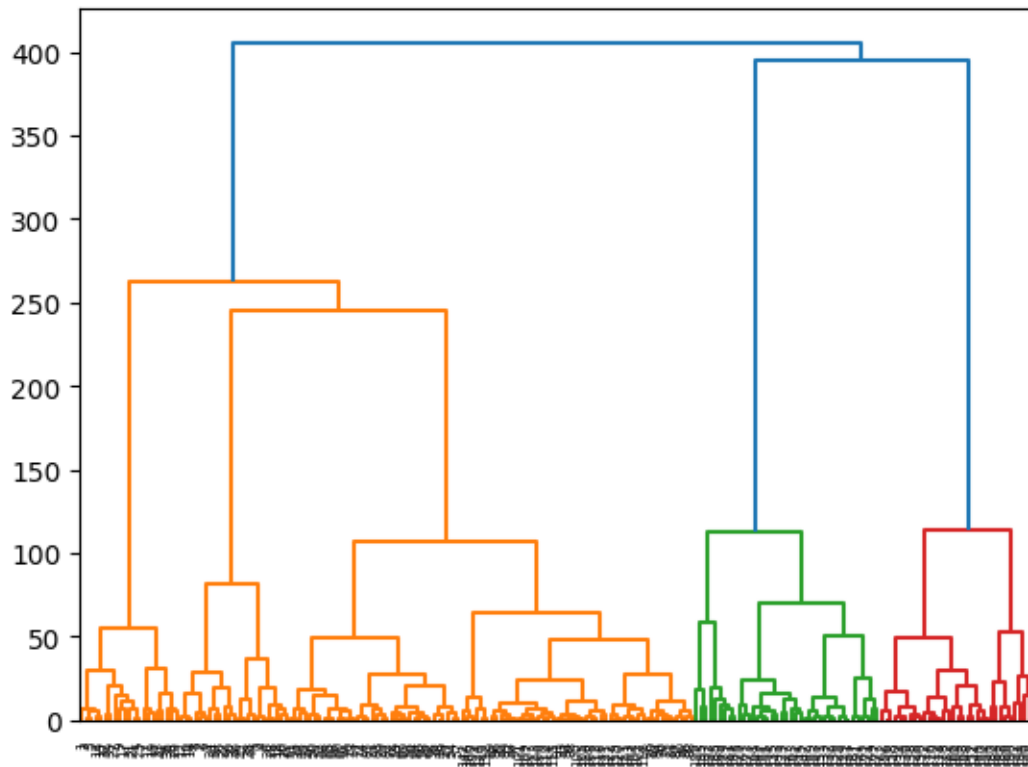
**As a data science engineer find out which hierachy cluster use maximum linkage in upcoming future** #####TASK-1: ##### Import the library and dataset #####TASK-2: ##### Using the dendrogram to find theoptimal no\_of clusters #####TASK-3: ##### Create a hierachy model and visualize the clusterwith help of matplot.lib library

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

```
[ ]: #Assign variable name "dataset" and the input variable as "X" indcludong select
    ↪all the row and index columns which you want [colum_index, Column_index].
dataset=pd.read_csv("Mall_Customers.csv")
dataset
X=dataset.iloc[:,[3,4]].values
```

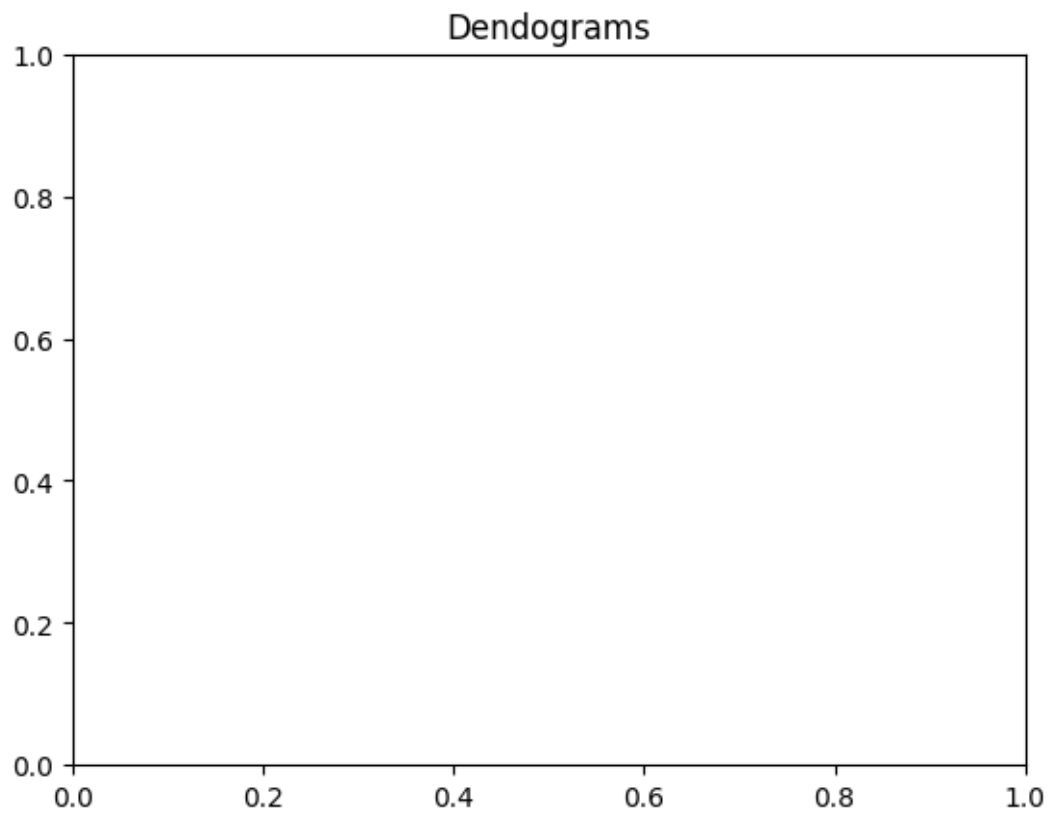
```
[ ]: #import scipy cluster using attribute "scipy.cluster.hierarchy" as sch alias
from scipy.cluster import hierarchy as sch
```

```
[ ]: #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'))"
denograph = sch.dendrogram(sch.linkage(X, method = 'ward'))
```



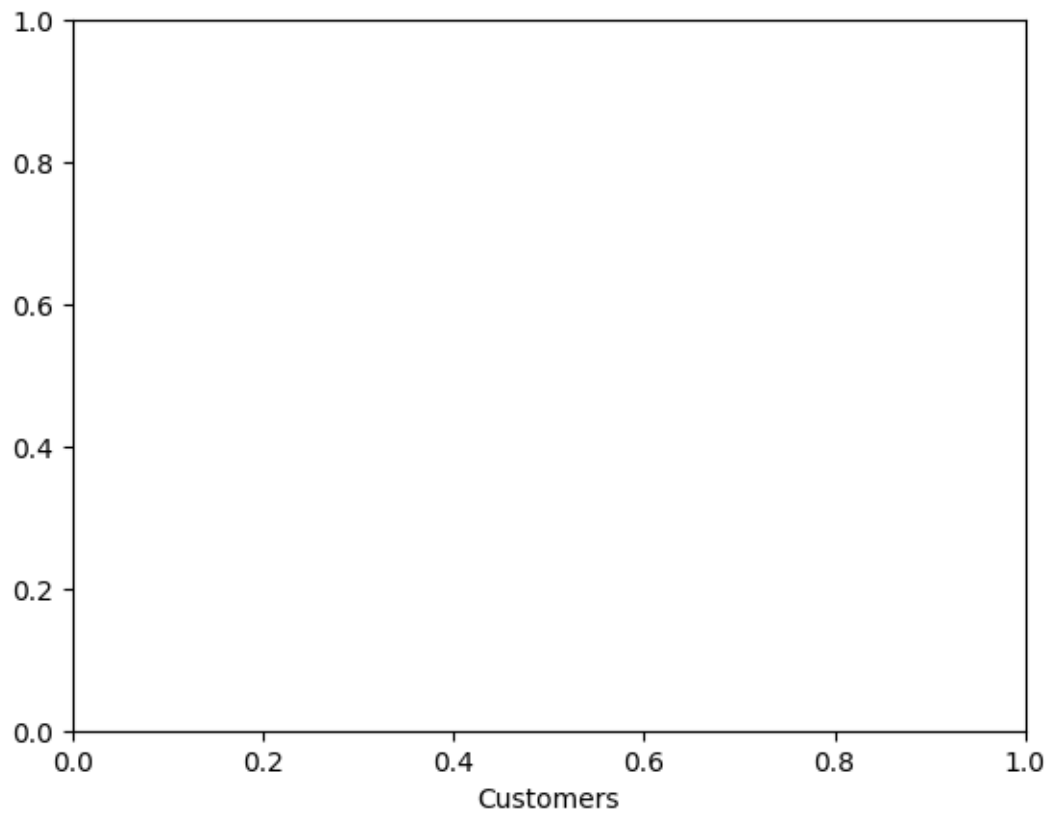
```
[ ]: #Assign the title as "Dendograms"
plt.title("Dendograms")
```

```
[ ]: Text(0.5, 1.0, 'Dendograms')
```



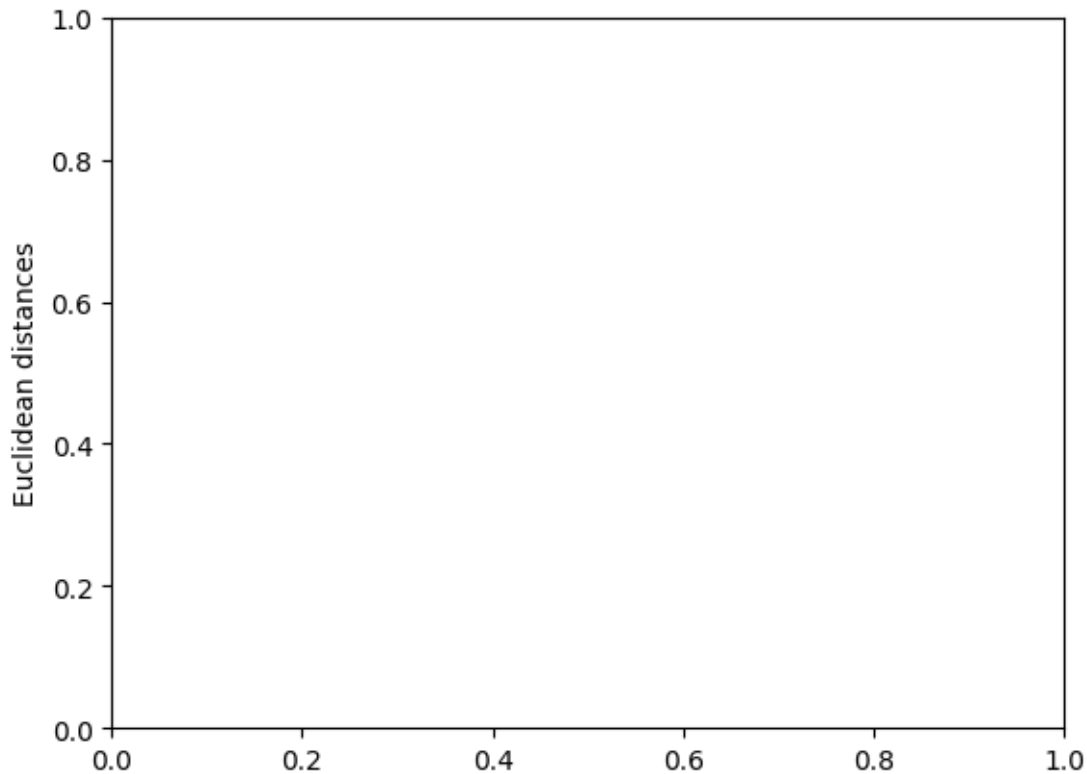
```
[ ]: #Label X axis as "Customers"  
plt.xlabel("Customers")
```

```
[ ]: Text(0.5, 0, 'Customers')
```



```
[ ]: #Label Y axis as 'Euclidean distances'  
plt.ylabel("Euclidean distances")
```

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



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

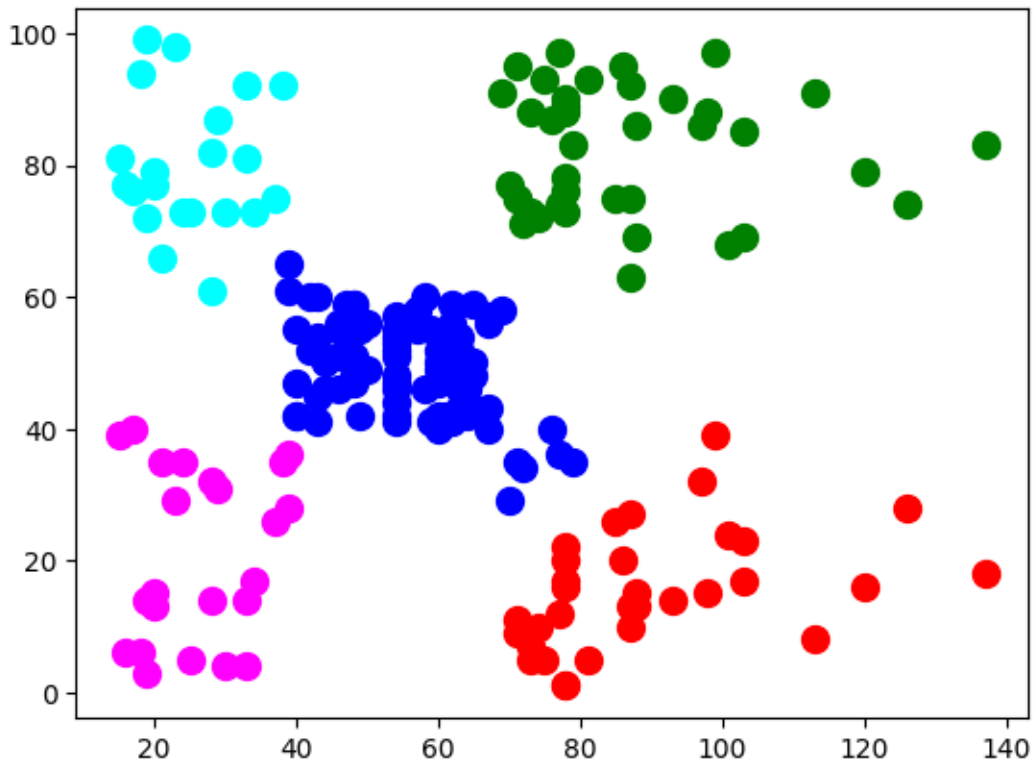
```
[ ]: #Create a cluster for five or nth cluster which you want.
      hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage =
      ↪ 'ward')
      y_hc = hc.fit_predict(X)
```

```
/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(
```

```
[ ]: #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')
```

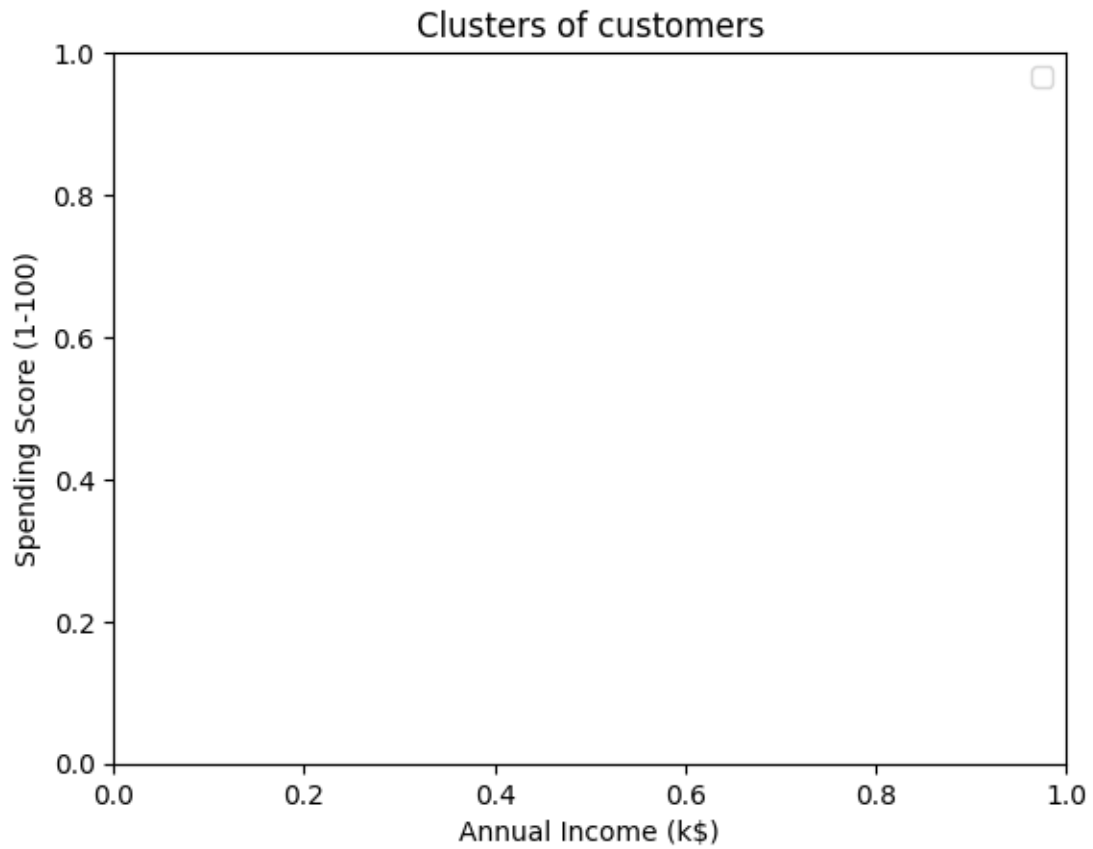
```
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label =_
↳'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta', label =_
↳'Cluster 5')
```

```
[ ]: <matplotlib.collections.PathCollection at 0x7ab6ac04b6d0>
```



```
[ ]: plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

WARNING:matplotlib.legend:No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



**##CONCLUSION #####**According to the model building as a engineer my prediction is cluster no-3 has the highest no\_of linkage

[ ]: