

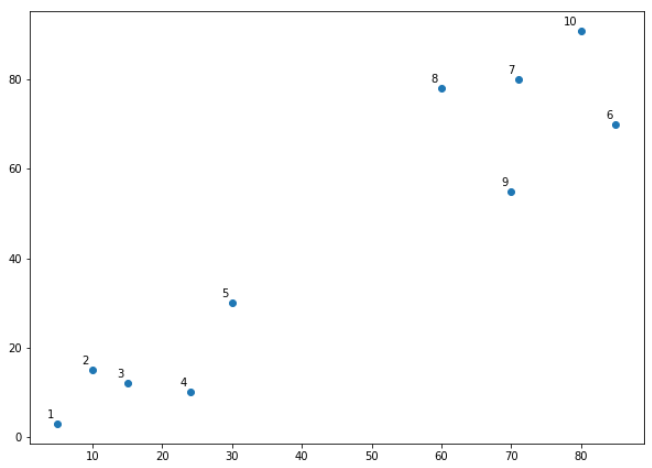
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programme | : | **B.Tech** | Semester | : | **Win Sem 21-22** |
| Course | : | **Web Mining Lab** | Code | : | **CSE3024** |
| Faculty | : | **Dr.Bhuvaneswari A** | Slot | : | **L7+L8** |
| Date | : | **22-03-2022** | Marks | : | **10 Points** |

**GOOGLE COLAB LINK: https://colab.research.google.com/drive/1YXHM6BnMBQrJ-vIm-3TcTgXbhEIZd7JR?usp=sharing**

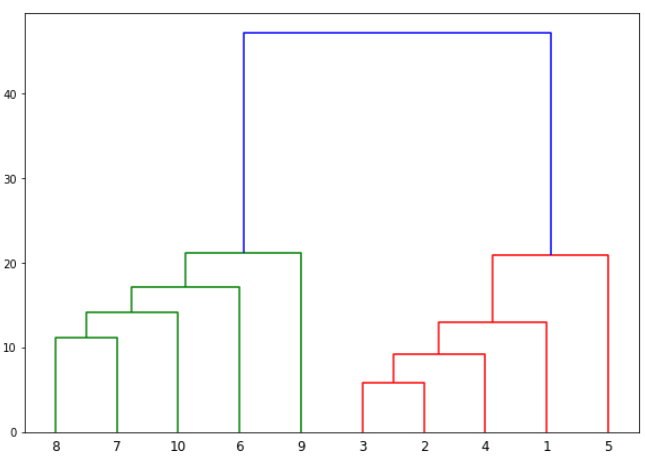
**Exercise11: Hierarchical Clustering**

1. Consider these data points. ([[5,3], [10,15], [15,12], [24,10], [30,30], [85,70], [71,80], [60,78], [70,55], [80,91],])

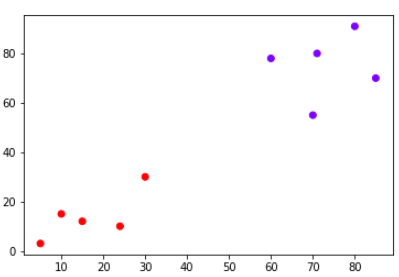
**STEP 1 :** Plot the data points



**STEP 2 :** Apply Hierarchical Clustering with use of dendrograms to get output



**STEP 3 : Show the output clusters**



**CODE AND OUTPUT:**

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

%matplotlib inline

X = np.array([[5,3],

[10,15],

[15,12],

[24,10],

[30,30],

[85,70],

[71,80],

[60,78],

[70,55],

[80,91],])

import matplotlib.pyplot as plt

labels = range(1, 11)

plt.figure(figsize=(10, 7))

plt.subplots\_adjust(bottom=0.1)

plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y inzip(labels, X[:, 0], X[:, 1]):

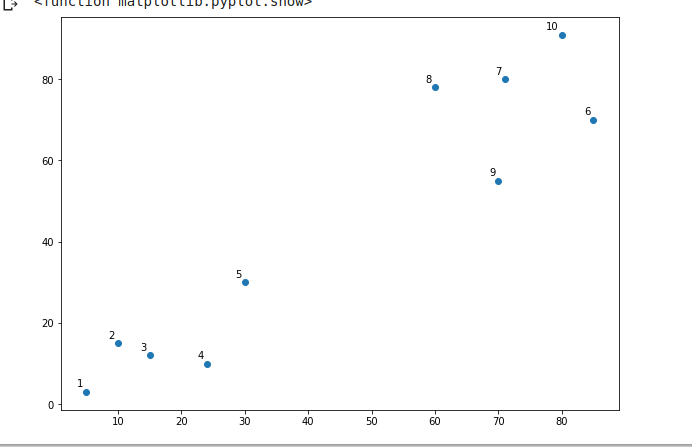
plt.annotate(

label,

xy=(x, y), xytext=(-3, 3),

textcoords='offset points', ha='right', va='bottom')

plt.show



from scipy.cluster.hierarchy import dendrogram, linkage

from matplotlib import pyplot as plt

linked = linkage(X, 'single')

labelList = range(1, 11)

plt.figure(figsize=(10, 7))

dendrogram(linked,

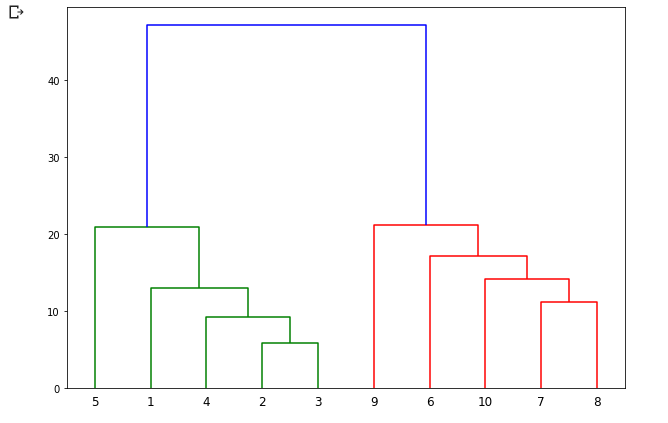
orientation='top',

labels=labelList,

distance\_sort='descending',

show\_leaf\_counts=True)

plt.show()



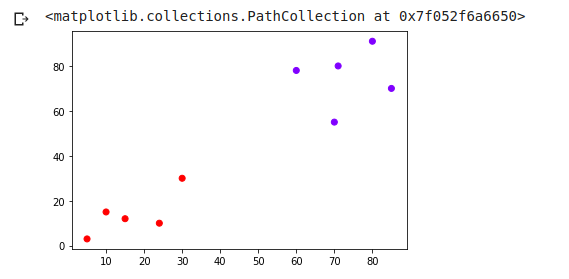
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n\_clusters=2, affinity='euclidean', linkage='ward')

cluster.fit\_predict(X)



plt.scatter(X[:,0],X[:,1], c=cluster.labels\_, cmap='rainbow')



2. Randomly generate 100 datapoints (x,y) and cluster them using hierarchial clustering.

CODE AND OUTPUT:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import random

data = []

for i inrange(100):

t1=round(random.random()\*100)

t2=round(random.random()\*100)

data.append([t1,t2])

X=np.array(data)

labels = range(1, 101)

plt.figure(figsize=(10, 7))

plt.subplots\_adjust(bottom=0.1)

plt.scatter(X[:,0],X[:,1], label='True Position')

for label, x, y inzip(labels, X[:, 0], X[:, 1]):

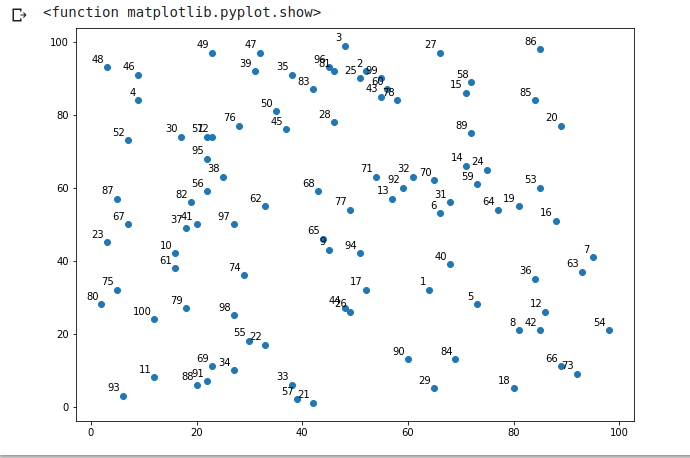
plt.annotate(

label,

xy=(x, y), xytext=(-3, 3),

textcoords='offset points', ha='right', va='bottom')

plt.show



from scipy.cluster.hierarchy import dendrogram, linkage

from matplotlib import pyplot as plt

linked = linkage(X, 'single')

labelList = range(1, 101)

plt.figure(figsize=(10, 7))

dendrogram(linked,

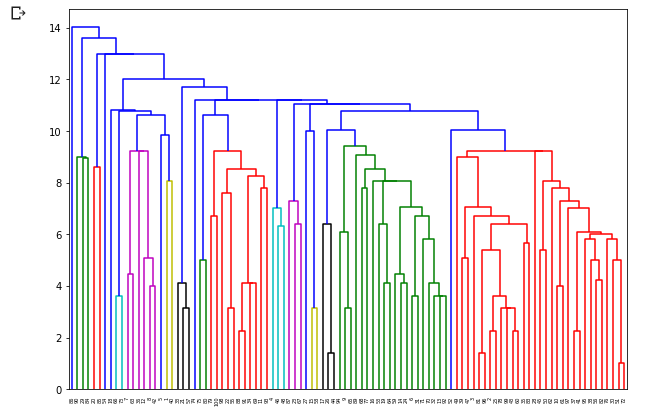
orientation='top',

labels=labelList,

distance\_sort='descending',

show\_leaf\_counts=True)

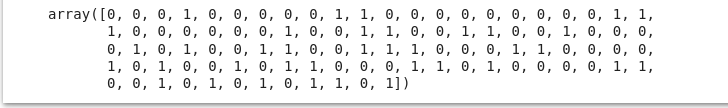
plt.show()



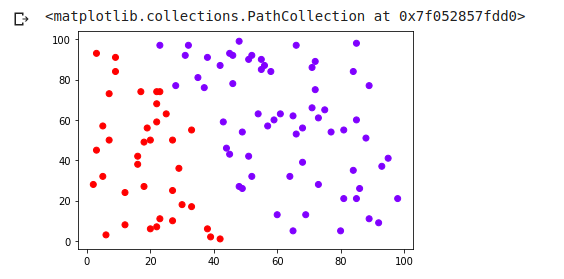
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n\_clusters=2, affinity='euclidean', linkage='ward')

cluster.fit\_predict(X)

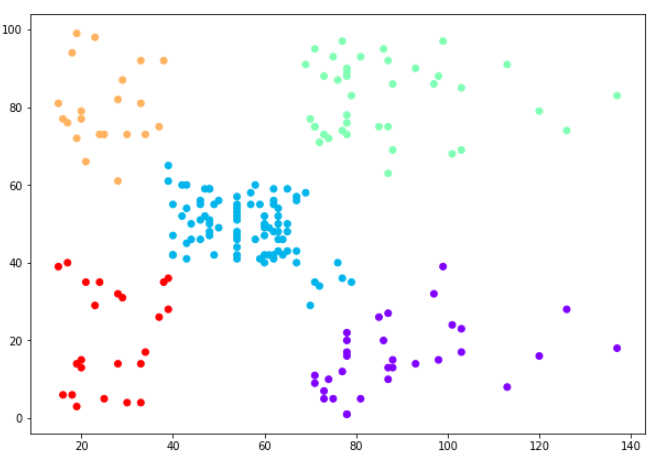


plt.scatter(X[:,0],X[:,1], c=cluster.labels\_, cmap='rainbow')



3. The problem that we are going to solve in this section is to segment customers into different groups based on their shopping trends. The dataset for this problem can be downloaded from the file folder: shopping-data.csv. To cluster this data into groups we will follow the same steps that we performed in the previous section.

**FINAL OUTPUT EXPECTED : Show the output clusters**



**CODE AND OUTPUT:**

import matplotlib.pyplot as plt

import pandas as pd

%matplotlib inline

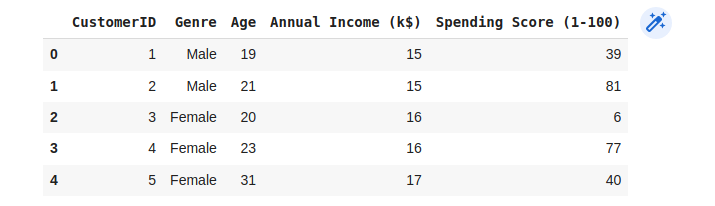
import numpy as np

customer\_data = pd.read\_csv('/content/shopping dataset.csv')

customer\_data.shape



customer\_data.head()



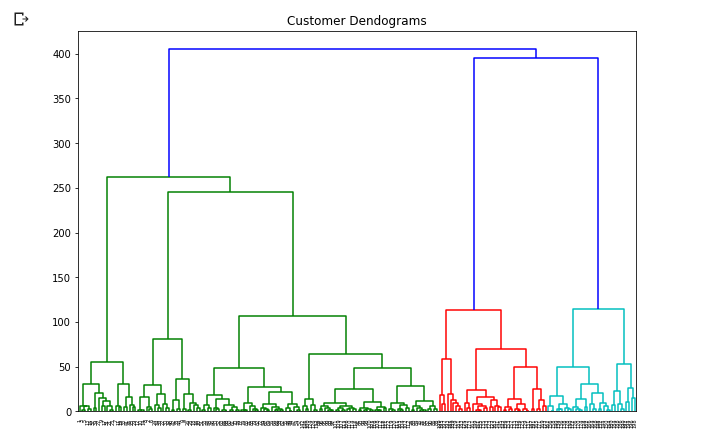
data = customer\_data.iloc[:, 3:5].values

import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))

plt.title("Customer Dendograms")

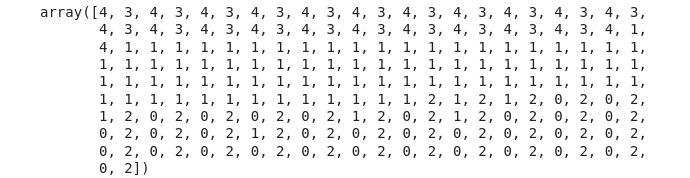
dend = shc.dendrogram(shc.linkage(data, method='ward')



from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n\_clusters=5, affinity='euclidean', linkage='ward')

cluster.fit\_predict(data)



plt.figure(figsize=(10, 7))

plt.scatter(data[:,0], data[:,1], c=cluster.labels\_, cmap='rainbow')

