

# CSE 3024 LAB ASSIGNMENT

REG NO - 19BCE1327

Kartikey Gautam

LAB11

## Hierarchical Clustering

### QUESTION 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],) Implement Hierarchical Clustering

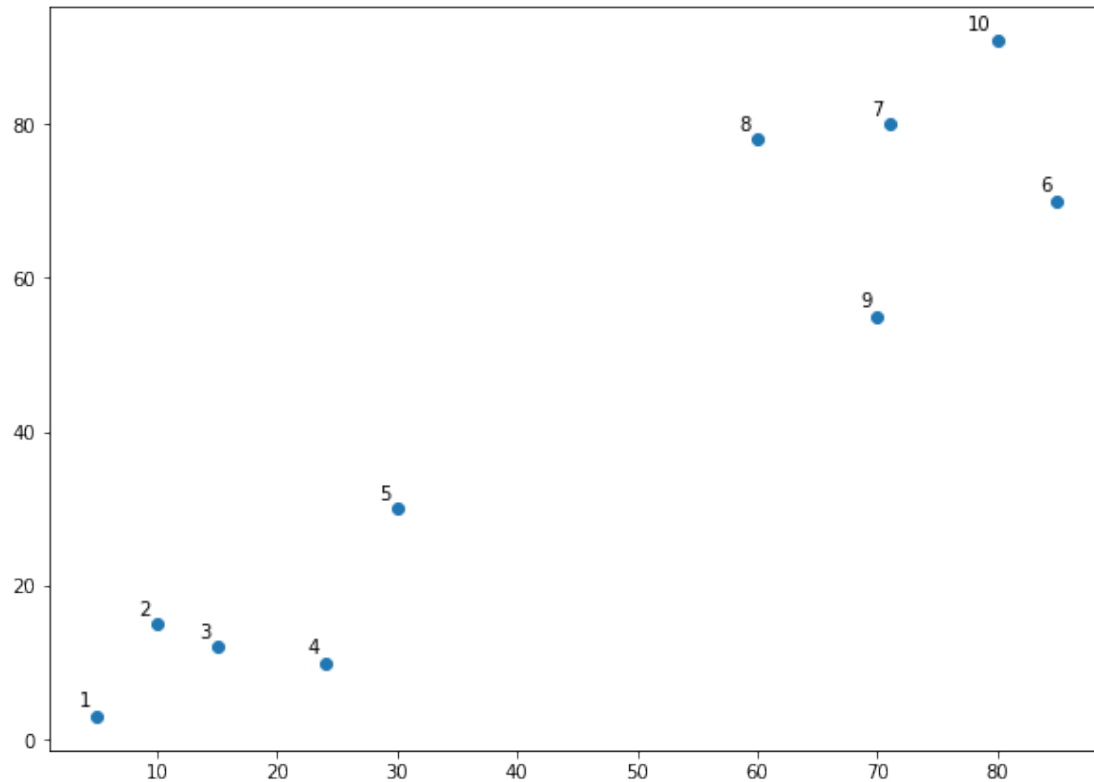
```
import numpy as np

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 in zip(labels, X[:, 0], X[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()
```



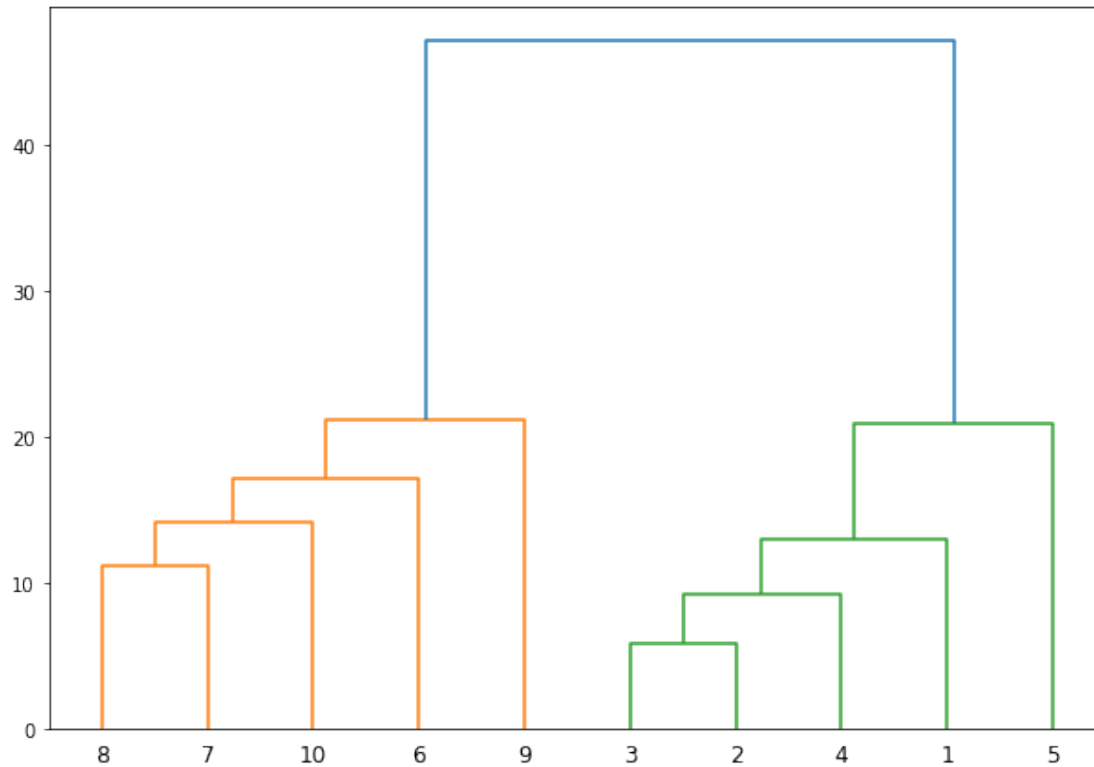
```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
```

```
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()
```



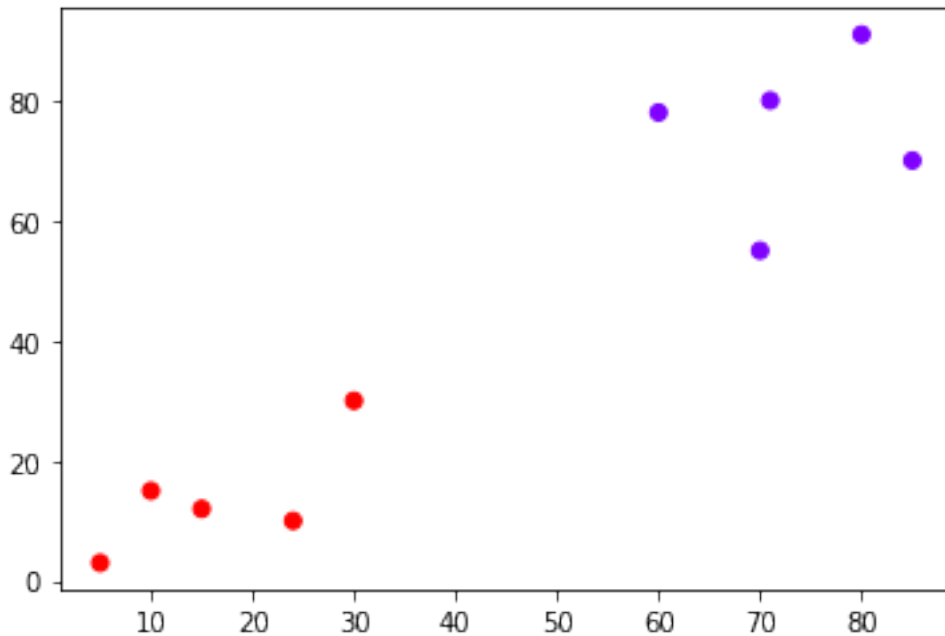
```
from sklearn.cluster import AgglomerativeClustering

cluster = AgglomerativeClustering(n_clusters=2, affinity='euclidean',
linkage='ward')
cluster.fit_predict(X)

array([1, 1, 1, 1, 1, 0, 0, 0, 0, 0])
print(cluster.labels_)

[1 1 1 1 1 0 0 0 0 0]

plt.scatter(X[:,0],X[:,1], c=cluster.labels_, cmap='rainbow')
<matplotlib.collections.PathCollection at 0x7faca88644c0>
```



Q2. Clustering on random 100 randomly generated points

```
import random
data_points = []
for i in range(100):
    x1 = round(random.random()*100)
    y1 = round(random.random()*100)
    data_points.append([x1,y1])
dp=np.array(data_points)
print(dp)
```

```
[[36 98]
 [13 92]
 [82  9]
 [ 7 59]
 [18 52]
 [65 39]
 [ 2 52]
 [51 43]
 [28 43]
 [82 78]
 [10 14]
 [35 73]
 [ 3 84]
 [62 98]
 [62 19]
 [91 93]
 [ 5 31]
 [ 1 21]
 [41 79]
```

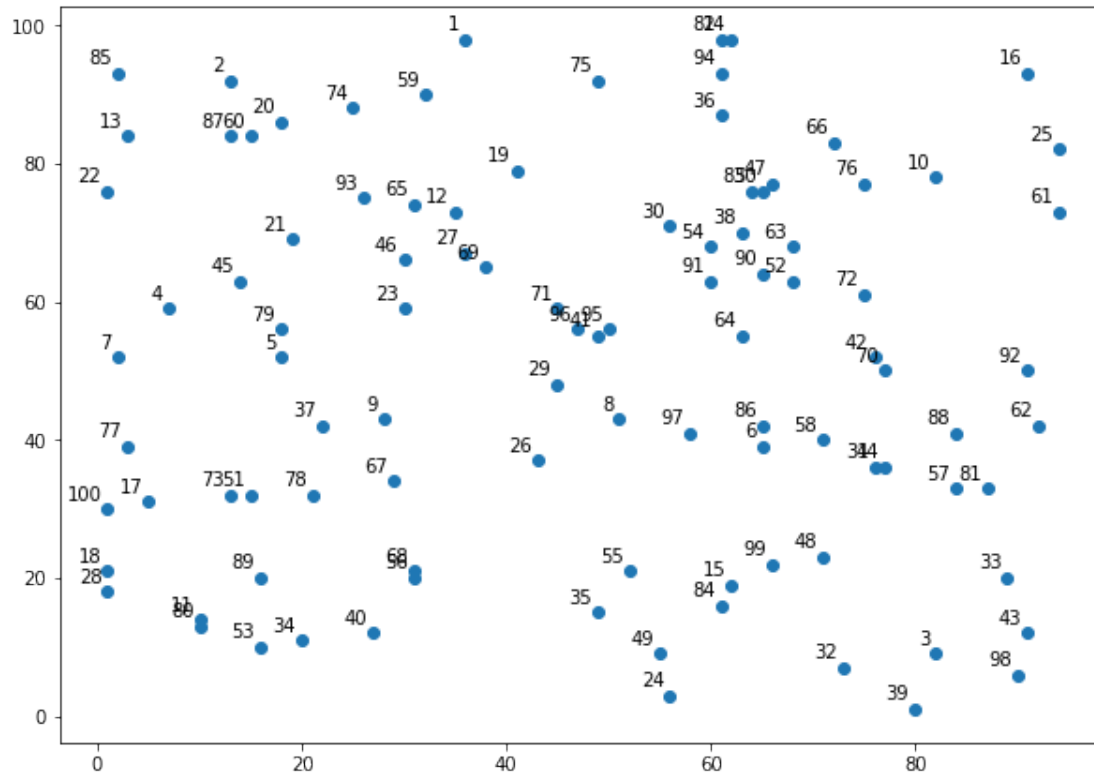
[18 86]  
[19 69]  
[ 1 76]  
[30 59]  
[56 3]  
[94 82]  
[43 37]  
[36 67]  
[ 1 18]  
[45 48]  
[56 71]  
[76 36]  
[73 7]  
[89 20]  
[20 11]  
[49 15]  
[61 87]  
[22 42]  
[63 70]  
[80 1]  
[27 12]  
[49 55]  
[76 52]  
[91 12]  
[77 36]  
[14 63]  
[30 66]  
[66 77]  
[71 23]  
[55 9]  
[65 76]  
[15 32]  
[68 63]  
[16 10]  
[60 68]  
[52 21]  
[31 20]  
[84 33]  
[71 40]  
[32 90]  
[15 84]  
[94 73]  
[92 42]  
[68 68]  
[63 55]  
[31 74]  
[72 83]  
[29 34]  
[31 21]  
[38 65]

```
[77 50]
[45 59]
[75 61]
[13 32]
[25 88]
[49 92]
[75 77]
[ 3 39]
[21 32]
[18 56]
[10 13]
[87 33]
[61 98]
[64 76]
[61 16]
[ 2 93]
[65 42]
[13 84]
[84 41]
[16 20]
[65 64]
[60 63]
[91 50]
[26 75]
[61 93]
[50 56]
[47 56]
[58 41]
[90  6]
[66 22]
[ 1 30]]
```

```
import matplotlib.pyplot as plt
```

```
labels = range(1, 101)
plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(dp[:,0],dp[:,1], label='True Position')

for label, x, y in zip(labels, dp[:, 0], dp[:, 1]):
    plt.annotate(
        label,
        xy=(x, y), xytext=(-3, 3),
        textcoords='offset points', ha='right', va='bottom')
plt.show()
```



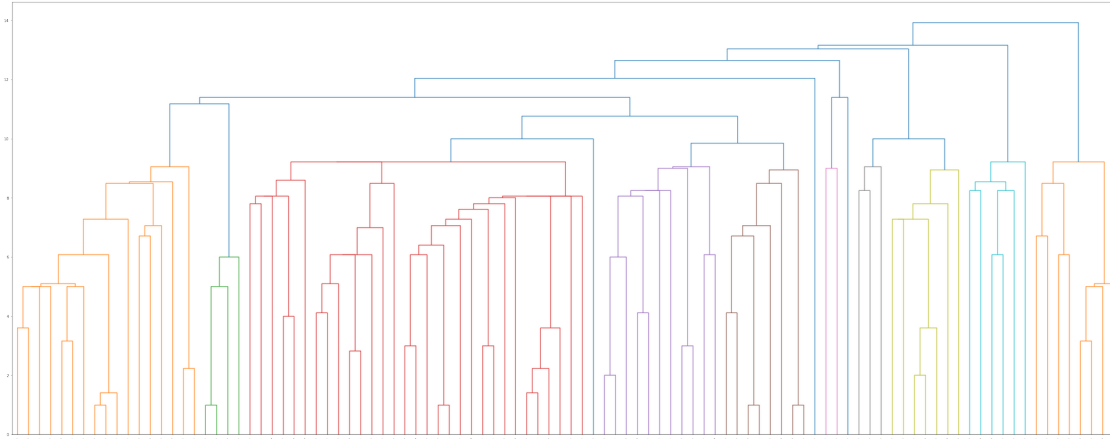
```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
```

```
linked = linkage(dp, 'single') #based on min distance b/w points
```

```
labelList = range(1, 101)
```

```
plt.figure(figsize=(50, 20))
dendrogram(linked,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
```

```
plt.show()
```



```

from sklearn.cluster import AgglomerativeClustering

cluster_q21 = AgglomerativeClustering(n_clusters=5,
affinity='euclidean', linkage='ward')
cluster_q21.fit_predict(dp)

array([1, 1, 4, 1, 1, 0, 1, 0, 3, 2, 3, 1, 1, 2, 4, 2, 3, 3, 1, 1, 1,
1,
      1, 4, 2, 0, 1, 3, 0, 2, 0, 4, 4, 3, 4, 2, 3, 2, 4, 3, 0, 0, 4,
0,
      1, 1, 2, 4, 4, 2, 3, 2, 3, 2, 4, 3, 0, 0, 1, 1, 2, 0, 2, 2, 1,
2,
      3, 3, 1, 0, 0, 2, 3, 1, 2, 2, 3, 3, 1, 3, 0, 2, 2, 4, 1, 0, 1,
0,
      3, 2, 2, 0, 1, 2, 0, 0, 0, 4, 4, 3])

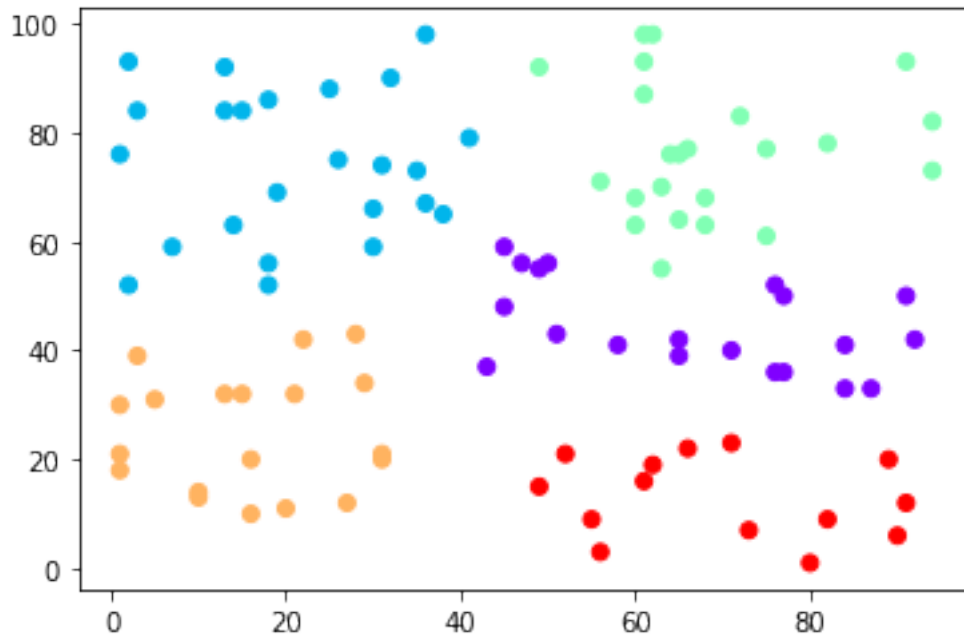
print(cluster_q21.labels_)

[1 1 4 1 1 0 1 0 3 2 3 1 1 2 4 2 3 3 1 1 1 1 1 4 2 0 1 3 0 2 0 4 4 3 4
2 3
 2 4 3 0 0 4 0 1 1 2 4 4 2 3 2 3 2 4 3 0 0 1 1 2 0 2 2 1 2 3 3 1 0 0 2
3 1
 2 2 3 3 1 3 0 2 2 4 1 0 1 0 3 2 2 0 1 2 0 0 0 4 4 3]

plt.scatter(dp[:,0],dp[:,1], c=cluster_q21.labels_, cmap='rainbow')
<matplotlib.collections.PathCollection at 0x7faca87c3580>

```





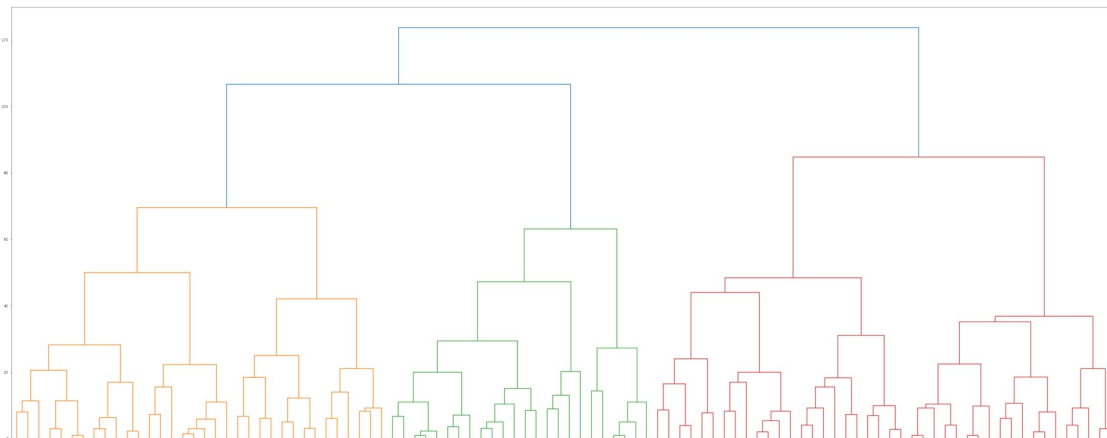
```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
```

```
linked2 = linkage(dp, 'complete') #using complete linkage (max distance)
```

```
labelList = range(1, 101)
```

```
plt.figure(figsize=(50, 20))
dendrogram(linked2,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
```

```
plt.show()
```



```

from sklearn.cluster import AgglomerativeClustering

cluster_q22 = AgglomerativeClustering(n_clusters=5,
affinity='euclidean', linkage='ward')
cluster_q22.fit_predict(dp)

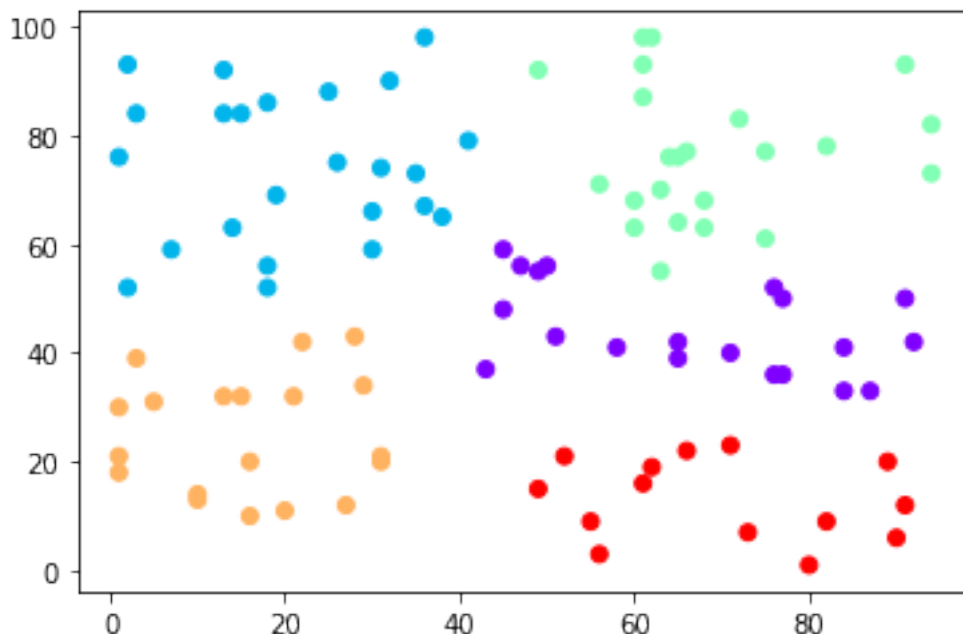
array([1, 1, 4, 1, 1, 0, 1, 0, 3, 2, 3, 1, 1, 2, 4, 2, 3, 3, 1, 1, 1,
1,
      1, 4, 2, 0, 1, 3, 0, 2, 0, 4, 4, 3, 4, 2, 3, 2, 4, 3, 0, 0, 4,
0,
      1, 1, 2, 4, 4, 2, 3, 2, 3, 2, 4, 3, 0, 0, 1, 1, 2, 0, 2, 2, 1,
2,
      3, 3, 1, 0, 0, 2, 3, 1, 2, 2, 3, 3, 1, 3, 0, 2, 2, 4, 1, 0, 1,
0,
      3, 2, 2, 0, 1, 2, 0, 0, 0, 4, 4, 3])

print(cluster_q22.labels_)

[1 1 4 1 1 0 1 0 3 2 3 1 1 2 4 2 3 3 1 1 1 1 1 4 2 0 1 3 0 2 0 4 4 3 4
2 3
 2 4 3 0 0 4 0 1 1 2 4 4 2 3 2 3 2 4 3 0 0 1 1 2 0 2 2 1 2 3 3 1 0 0 2
3 1
 2 2 3 3 1 3 0 2 2 4 1 0 1 0 3 2 2 0 1 2 0 0 0 4 4 3]

plt.scatter(dp[:,0],dp[:,1], c=cluster_q22.labels_, cmap='rainbow')
<matplotlib.collections.PathCollection at 0x7faca6a0b430>

```



```

from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

```

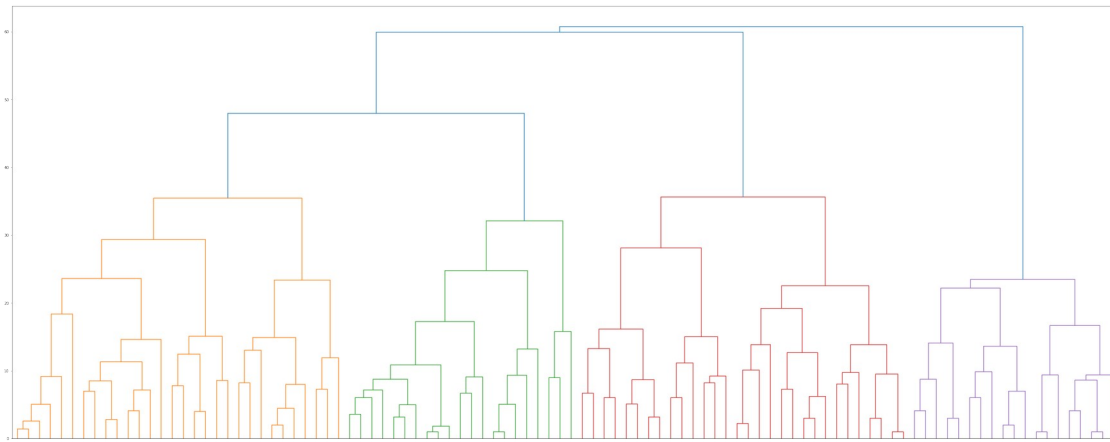
```

linked3 = linkage(dp, 'average') #avg distance b/w points

labelList = range(1, 101)

plt.figure(figsize=(50, 20))
dendrogram(linked3,
            orientation='top',
            labels=labelList,
            distance_sort='descending',
            show_leaf_counts=True)
plt.show()

```



```

cluster_q23 = AgglomerativeClustering(n_clusters=5,
affinity='euclidean', linkage='ward')
cluster_q23.fit_predict(dp)

array([1, 1, 4, 1, 1, 0, 1, 0, 3, 2, 3, 1, 1, 2, 4, 2, 3, 3, 1, 1, 1,
1,
      1, 4, 2, 0, 1, 3, 0, 2, 0, 4, 4, 3, 4, 2, 3, 2, 4, 3, 0, 0, 4,
0,
      1, 1, 2, 4, 4, 2, 3, 2, 3, 2, 4, 3, 0, 0, 1, 1, 2, 0, 2, 2, 1,
2,
      3, 3, 1, 0, 0, 2, 3, 1, 2, 2, 3, 3, 1, 3, 0, 2, 2, 4, 1, 0, 1,
0,
      3, 2, 2, 0, 1, 2, 0, 0, 0, 4, 4, 3])

print(cluster_q23.labels_)

[1 1 4 1 1 0 1 0 3 2 3 1 1 2 4 2 3 3 1 1 1 1 1 4 2 0 1 3 0 2 0 4 4 3 4
2 3
 2 4 3 0 0 4 0 1 1 2 4 4 2 3 2 3 2 4 3 0 0 1 1 2 0 2 2 1 2 3 3 1 0 0 2
3 1
 2 2 3 3 1 3 0 2 2 4 1 0 1 0 3 2 2 0 1 2 0 0 0 4 4 3]

plt.scatter(dp[:,0],dp[:,1], c=cluster_q23.labels_, cmap='rainbow')
<matplotlib.collections.PathCollection at 0x7faca6819b20>

```

