

## Hierarchical (Agglomerative) Clustering Algorithm

```
# To perform agglomerative clustering on randomly generated data

import numpy as np
import pandas as pd
from scipy.cluster import hierarchy
from scipy.spatial import distance_matrix
import matplotlib.pyplot as plt
from sklearn import manifold, datasets
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets.samples_generator import make_blobs

# Generating random data #

## n_samples= total data points, centers= no. of centers to generate, cluster_std=standard deviation

X1,y1=make_blobs(n_samples=50,centers=[[4,4],[-2,-1],[1,1],[10,4]],cluster_std=0.9)
#Scatter plot of randomly generated data
plt.scatter(X1[:,0],X1[:,1],marker='o')

# Agglomerative Clustering #
#n_clusters= no. of clusters to form along with their centroids, linkage - criterion:complete

agg=AgglomerativeClustering(n_clusters=4,linkage='average')
agg.fit(X1,y1)

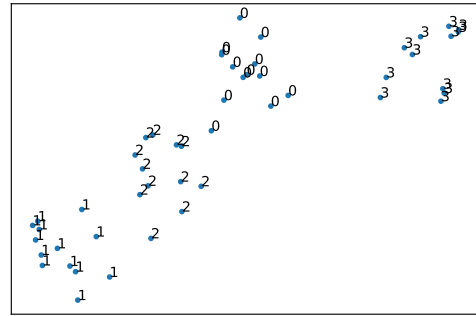
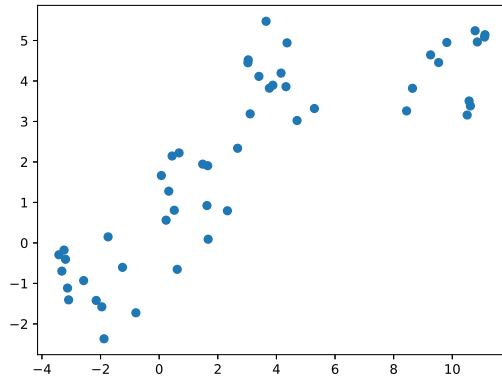
# Plotting after clustering
plt.figure(figsize=(6,4))
x_min,x_max=np.min(X1,axis=0),np.max(X1,axis=0)
X1=(X1-x_min)/(x_max-x_min)

for i in range(X1.shape[0]):
    plt.text(X1[i,0],X1[i,1],str(y1[i])),
    color=plt.cm.nipy_spectral(agg.labels_[i]/10),
    fontdict={'weight':'bold','size':9}

plt.xticks([])
plt.yticks([])

plt.scatter(X1[:,0],X1[:,1],marker='.')

## Dendrogram ##
```



```
dmat=distance_matrix(X1,X1)
with open('Hierarchical.txt','a') as f:
    print(dmat,file=f)
Z = hierarchy.linkage(dmat, 'complete')
plt.figure()
dendro = hierarchy.dendrogram(Z)
# Display plot
plt.show()
```

Solution:

```
[[0.          0.24176836 0.89417683 ... 0.37702876 0.1088064 0.32216237]
 [0.24176836 0.          1.08601781 ... 0.42253887 0.13420902 0.41203647]
 [0.89417683 1.08601781 0.          ... 0.744795 0.96979874 0.70550771]
 ...
 [0.37702876 0.42253887 0.744795 ... 0.          0.36738486 0.08062208]
 [0.1088064 0.13420902 0.96979874 ... 0.36738486 0.          0.33335895]
 [0.32216237 0.41203647 0.70550771 ... 0.08062208 0.33335895 0.          ]]
[[0.          0.35189886 0.47657609 ... 0.28093781 0.69081375 0.66660132]
 [0.35189886 0.          0.81830272 ... 0.58579408 0.40751668 0.34446779]
 [0.47657609 0.81830272 0.          ... 0.25800492 1.08269852 1.09230709]
 ...
 [0.28093781 0.58579408 0.25800492 ... 0.          0.82501021 0.83823636]
 [0.69081375 0.40751668 1.08269852 ... 0.82501021 0.          0.11963145]
 [0.66660132 0.34446779 1.09230709 ... 0.83823636 0.11963145 0.          ]]
```

