**12.2**

**Code：**

# 生成数据

import numpy as np

X = np.array([[1,3,2,4,1,5,5,5,5,7,4,9,2,8,3,10]]).reshape(-1,2)

# 展示点分布和计算点距离矩阵

import matplotlib.pyplot as plt

plt.plot(X[:,0], X[:,1], '.r')

for i in range(8):

plt.text(X[i,0]+0.05, X[i,1], i+1)

from scipy.spatial.distance import cdist

distance = cdist(X, X, metric='euclidean')

# 类距离计算函数

def dist\_compute(A, B, method='s'):

try:

if method == 'single':

return distance[np.ix\_(A, B)].min()

if method == 'complete':

return distance[np.ix\_(A, B)].max()

if method == 'average':

return distance[np.ix\_(A, B)].mean()

except TypeError:

return distance[A, B]

# 自定义的Agglomerative分层聚合

for method in ['single', 'complete', 'average']:

cluster = dict(zip(range(8), [[i] for i in range(8)]))

for step in range(1,8):

cluster\_distance = np.ones([8, 8]) \* distance.max()

for k1, v1 in cluster.items():

for k2, v2 in cluster.items():

if k1 != k2:

cluster\_distance[k1, k2] = dist\_compute(v1, v2, method)

A, B = np.unravel\_index(

np.argmin(cluster\_distance), cluster\_distance.shape)

cluster[A].extend(cluster[B])

del cluster[B]

if step==5:

break

print("在", method, " linkage下的自定义三类聚类结果：", cluster)

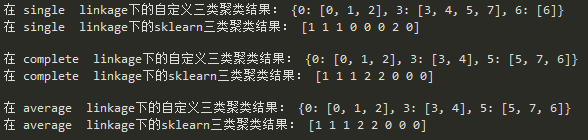
# 对比sklearn的Agglomerative分层聚合

from sklearn.cluster import AgglomerativeClustering

clu = AgglomerativeClustering(n\_clusters=3, linkage=method)

print("在", method, " linkage下的sklearn三类聚类结果：", clu.fit\_predict(X) , "\n")

**Output：**



其中自定义的结果由类别-类内点的数值对形式构成。可以看到结果与sklearn调包结果完全一致。

**12.3**

**Code：**

# 读取数据

import pandas as pd

data = pd.read\_table("primate.scapulae.txt")

X = data.iloc[:, 1:8]

y = data.iloc[:, -1]

# 标准化

X = (X - X.mean()) / X.std()

# 分别进行聚类，并计算各自的误分类率

import numpy as np

from sklearn.cluster import AgglomerativeClustering

from sklearn.metrics import accuracy\_score

for method in ['single', 'complete', 'average']:

y\_pred = np.zeros(len(y))

cluster = AgglomerativeClustering(n\_clusters=5, linkage=method)

clu\_pred = cluster.fit\_predict(X)

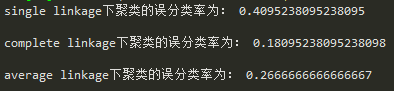
for i in range(5):

pred = y[clu\_pred == i].value\_counts().index[0]

y\_pred[clu\_pred == i] = pred

print(method + " linkage下聚类的误分类率为：",1 - accuracy\_score(y, y\_pred) , "\n")

**Output：**



可以看到，single linkage聚类的误分类率最大，complete linkage聚类的误分类率最小。