資料科學導論 HW2

a. 程式說明

1.引用套件:

pandas 對 csv 做讀寫。
從 sklearn 引入 cluster,來使用寫好的分群 model。
引入 StandardScaler 對資料做標準化。

```
from sklearn.preprocessing import StandardScaler
from sklearn import cluster
import pandas as pd
```

2.讀入資料&資料預處理：

讀入 csv 開始做資料預處理。
觀察 csv 檔中資料,並在經過測試後發現'id'、'feature3'、'feature4'、'feature5' 對分群沒幫助,甚至會影響準確度,是多餘的 attribute,所以將他們 drop 掉。
因為資料大部分為平均值和標準差,所以對資料常態分佈,使離群值影響降低

```
#read csv
train = pd.read_csv('data.csv')
test = pd.read_csv('test.csv')
submit = pd.read_csv('submit.csv')

#Drop the redundant data
train=train.drop(['id'],axis=1)
train=train.drop(['feature3'],axis=1)
train=train.drop(['feature4'],axis=1)
train=train.drop(['feature5'],axis=1)

#Data Standardize
train_std = StandardScaler().fit_transform(train)
```

3.進行分群：

我使用 Hierarchical Clustering 來做分群,分成 13 群得到最高的精準度。
Linkage 衡量群間的遠近程度 使用 'ward'離差平方和法
Affinity 用來計算群間距離 使用 'euclidean'歐式距離
針對 test 裡兩個 ID 是否為同群做比較,分完若兩 ID 為同群就為 1,反之為 0。

```
#Clustering to 13 clusters
hclust = cluster.AgglomerativeClustering(linkage = 'ward', affinity = 'euclidean', n_clusters = 13).fit(train_std)
Clustering_result =  hclust.labels_

for i in range(test.shape[0]):
    cluster1 = Clustering_result[test.iloc[i,1]]
    cluster2 = Clustering_result[test.iloc[i,2]]
    if cluster1 == cluster2:
        submit.iloc[i,1] = '1'
    else:
        submit.iloc[i,1] = '0'
```

4.結果:

將結果存入 submit.csv

```
#output the result to csv
submit.to_csv('submit.csv',index=False)
submit
```

	index	ans
0	0	0
1	1	0
2	2	0
3	3	0
4	4	0
...	...	...
395	395	0
396	396	0
397	397	0
398	398	0
399	399	0

400 rows x 2 columns

16	B10715027_邱柏誠		0.92500

b. 演算法介紹

我使用 Hierarchical clustering 做分群,一開始有試過 K-means,後來再試 Hierarchical clustering 發現精準度較高而成為最後的選擇,會想到使用 Hierarchical clustering 是因為資料數不大,計算複雜度太高這項缺點影響不大,他在分群的效果也比 K-means 好,且限制也較少,所以才決定改以這方法嘗試分群,調整參數及分群數,最後在 linkage = 'ward', affinity = 'euclidean', n_clusters = 13 中得到最高的精準度。