ML lab2 KNearestNeighbor

0416037 李家安

1. Result

```
resubstation, algo=brute, dist=manhattan
1.0
--- 0.33111858367919922 seconds ---
resubstation, algo=kd_tree, dist=manhattan
1.0
--- 0.045030832290649414 seconds ---
resubstation, algo=ball_tree, dist=manhattan
1.0
--- 0.12276124954223633 seconds ---

resubstation, algo=brute, dist=euclidean
1.0
--- 0.32858037948608398 seconds ---
resubstation, algo=kd_tree, dist=euclidean
1.0
--- 0.03443956375122070 seconds ---
resubstation, algo=ball_tree, dist=euclidean
1.0
--- 0.12823104858398438 seconds ---

kFold, foldNum=12, algo=brute, dist=manhattan
0.623313577672
--- 0.3550472569465637 seconds ---
kFold, foldNum=12, algo=kd_tree, dist=manhattan
0.621873837833
--- 0.10034942626953125 seconds ---
kFold, foldNum=12, algo=ball_tree, dist=manhattan
0.621881588283
--- 0.16851043701171875 seconds ---


kFold, foldNum=12, algo=brute, dist=euclidean
0.612780140828
--- 0.4088349342346191 seconds ---
kFold, foldNum=12, algo=kd_tree, dist=euclidean
0.612202887753
--- 0.10797286033630371 seconds ---
kFold, foldNum=12, algo=ball_tree, dist=euclidean
0.618215858863
--- 0.17551708221435547 seconds ---

resubstation, algo=brute, dist=mahalanobis
1.0
--- 1.7785559797286987 seconds ---
kFold, foldNum=12, algo=brute, dist=mahalanobis
0.677833988625
--- 2.8411589553070068 seconds ---

resubstation, algo=brute, dist=cosDist
0.567782768477
--- 473.0676188468933 seconds ---
kFold, foldNum=12, algo=brute, dist=cosDist
0.456921873238
--- 438.41667986297607 seconds ---
```

2. Environment
Ubuntu 16.04
Python 3.5

3. Language and Library
Python 3.5
Pandas
NumPy
SciPy
SciKit-learn

4. How to use it
$ python3 wine.py

5. Code
在程式中我先對資料做處理，將 dataset 以網路的方式抓下來，用 pandas 轉換為 DataFrame 的形式，再把 quality 這個 column 切出來分成 feature 和 target，同時對 feature 做 normalization。
定義兩個函式 resubstitution 和 kfold 分別做 resubstitution 跟 kfold，分別用不同的距離跟算法當參數輸入。
resubstation 的部分，以距離跟算法建立 KNeighborsClassifier 物件 nbr，nbr.fit() 來訓練，然後用 nbr.predict 來預測，最後用 confusion matrix 測量精準度。過程中我們沒有對 data 做任何的切割，也就是直接拿原始資料做訓練以及預測，符合 resubstation 的採樣方式。
kfold 的部分，以 kFold object 將 dataset 分割為 12 份，用 for loop 分別計算每次預測的精準度，加總後再除以 12 算出平均精準度。
運算時將 ball_tree, kd_tree, brute 與 manhattan, euclidean 交叉丟入兩函式運算，以及將 brute 與 mahalanobis, cosDist 交叉丟入函式運算，就可以得到各類模型的精確度了。