HW2 R06921034
林東農

作法：

根據講義上要求下命令的形式
python classification.py [R, D, S, N] train.csv test.csv
就可以執行想要使用的方法
執行結束時，會在當前資料夾產生一個 prediction.csv 的 file
這個 prediction.csv file 是根據 test.csv 檔透過 train 好的 model 產生的預測
透過實驗，發現很多 model 都可以達到 90%以上的準確率

Regression
這部份我是採用 Logistic Regression 的方式

```
if command=="R":
        #print("Regression")
        logr = linear_model.LogisticRegression(penalty='l2',solver='liblinear',multi_class='ovr',verbose=0,n_jobs=1)
        logr.fit(xtrain,ytrain)
        predictY=logr.predict(xtest)
        #pd.DataFrame(predictY).to_csv('prediction.csv')
```

採用講義上的參數，執行的時候準確率可以達到 94%左右
非常高的準確率

Decision Tree
這部份也是採用講義上的參數

```
elif command=="D":
        #print("DT")
        sklearn.tree.DecisionTreeClassifier(
        criterion='gini',
        splitter='best',
        max_depth=None,
        min_samples_split=2,
        max_features=None,
        max_leaf_nodes=None,
        min_impurity_decrease=0.0)

        #xtrain, xtest, ytrain, ytest =sklearn.model_selectio
        dct = sklearn.tree.DecisionTreeClassifier()
        dct.fit(xtrain, ytrain)
        predictY=dct.predict(xtest)
        #pd.DataFrame(predictY).to_csv("prediction.csv")
```

準確率達到 92%
SVM
這個開始有實驗多種不同的參數。一開始直接讓 svc_model=sklearn.svm.SVC()，結果在用 cross validation 測試的時候只有 83%，仍不及助教要求的 85% 。
最後用

```
parameter_candidates = [{'C': [1,10,100,1000], 'gamma': [0.001,0.0001],'kernel': ['rbf']}]
clf = GridSearchCV(estimator=svm.SVC(),param_grid=parameter_candidates, n_jobs=-1)
```

```
clf.fit(xtrain,ytrain)

predictY=clf.predict(xtest)
```

準確率可以達到 93%，有時甚至是 95%的高準確率

這個方法需要比較長的時間，但不會超過 1 分鐘

Neural Network
這個方法採用的策略是助教上課講的，因為 Multi-Layer Perceptron 對於 feature 的 scaling 很敏感，因此採用的是 scaling 後的 feature 去 train

```
scalar=StandardScaler()

#Xtrain, Xtest, ytrain, ytest = sklearn.model_sel
scalar.fit(xtrain)
xtrain2 = scalar.transform(xtrain)
xtest2 = scalar.transform(xtest)
mlp = MLPClassifier(solver="adam",max_iter=200)
mlp.fit(xtrain2, ytrain)
predictY=mlp.predict(xtest2)
```

用 cross validation 測試的時候，在未 scaling 的 feature 可以達到 93% ，scaling 後的準確率可以高達 96%
因此用 scaling 的 feature 這個方法去產生的 prediction.csv 。

測試： Cross Validation
我是用 train.csv 檔部份當作 test case,

```
traincx,testcx,traincy,testcy = sklearn.model_selection.train_test_split(xtrain,ytrain,test_size=0.15)
```

如圖所示，85%當作 train data, 15%當作 test data 。

發現問題:
在這次作業中，常用到的一些 function 在 0.18 存在，但在 0.20 就會被刪除
在執行的時候會出現 warning

另外，在執行 SVM 時，總需要最多的時間 。試驗的時候似乎是 kernel='linear'才會出現花很多時間，因此後來採用 kernel='rbf'