Code

109611099_顏彥臣_hw5.py - C:/prog/ML/hw5/109611099_顏彥臣_hw5.py (3.8.3)

File Edit Format Run Options Window Help

```
import numpy as np
import numpy as np
import mglearn
import pandas as pd
import sklearn
import matplotlib.pyplot as plt
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
#############################################################
def readHw5Cancer(inFileName):
    # init
    recArr = []
    clsArr = []

    # open input text data file, format is given
    inFile = open(inFileName, 'r')
    s = inFile.readline() # skip

    row = 0
    while True:
        s = inFile.readline()
        data1 = s.strip() # remove leading and ending blanks
        if (len(data1) <= 0):
            break

        # since we use append, value must be created in the loop
        value = []

        strs31 = data1.split(',') # array of 31 str

        # convert to real
        for ix in range(30):
            value.append( eval(strs31[ix]) )
        # end for

        target = eval(strs31[30])

        recArr.append(value) ;  # add 1 record at end of array
        clsArr.append(target) ; # add 1 record at end of array

        row = row+1 # total read counter
    # end while
    # close input file
    inFile.close()
    # convert list to Numpy array
    npXY = np.array(recArr)
```

```
    # close input file
    inFile.close()
    # convert list to Numpy array
    npXY = np.array(recArr)
    npC  = np.array(clsArr)
    # pass out as Numpy array
    return npXY, npC
# end function

#########################################################
def TrainTestSplit_Fold(X, y, fold, test_size):
    # safety check
    if (fold < 0):
        fold = 0

    # 輸入
    numValue = X.size
    rows = len(X)
    cols = int(numValue/rows)

    # safety check
    rmns = numValue % rows
    if (rmns != 0):
        print("ERROR - missing data in X")

    t0 = fold * test_size
    t1 = t0 + test_size
    # safety check
    if (t1 > rows):
        print("ERROR - out of bound")
        t1 = rows


    fea_test = X[t0:t1,:]
    tar_test = y[t0:t1]


    dr = [t0+x for x in range(test_size)]


    fea_train = np.delete(X, dr, 0)
    tar_train = np.delete(y, dr, 0)
    return fea_train, fea_test, tar_train, tar_test
# end function
```

```
#########################################################
def Train_model_5fold(model,X_5fold,y_5fold,X_check,y_check):
    num_folds=5
    numRec=len(X_5fold)
    testsize=int(numRec/num_folds)#95
############################
    total_train = 0
    total_test = 0
    for fold in range(num_folds):
        X_train, X_test, y_train, y_test = TrainTestSplit_Fold(X_5fold, y_5fold, fold, test_size)
        ir=model.fit(X_train,y_train)
        train_s = ir.score(X_train, y_train)
        test_s  = ir.score(X_test,  y_test)

        total_train += train_s
        total_test  += test_s
        #average
    average_train = total_train/num_folds
    average_test  = total_test/num_folds
    print("Train/Test average score: {:.3f}/{:.3f}".format(average_train, average_test))
        #5 fold
    ir5= model.fit(X_5fold, y_5fold)
    fold5_s=model.score(X_5fold,  y_5fold)
    check_s=model.score(X_check,  y_check)
    return fold5_s,check_s
#########################################################
print("109611099_顏彥臣_HW5_cancer.txt")
X,y=readHw5Cancer("C:\\prog\\ML\\hw5\\hw5_cancer.csv")
X_5fold, X_check, y_5fold, y_check = train_test_split(X, y, test_size = 94, random_state = 0)
#分割
g_numRec = 475
num_folds = 5
test_size = int(g_numRec * (1.0/num_folds))
# loop through folds
total_train = 0
total_test = 0

for fold in range(num_folds):
    X_train, X_test, y_train, y_test = TrainTestSplit_Fold(X_5fold, y_5fold, fold, test_size)
    logreg=LogisticRegression(max_iter=10000).fit(X_train,y_train)
    train_s = logreg.score(X_train, y_train)
    test_s  = logreg.score(X_test,  y_test)
    total_train += train_s
    total_test  += test_s

    #5 fold
    logreg5= LogisticRegression(max_iter=10000).fit(X_5fold, y_5fold)
    fold5_s=logreg5.score(X_5fold,  y_5fold)
    check_s=logreg5.score(X_check,  y_check)
```

109611099_顏彥臣_hw5.py - C:/prog/ML/hw5/109611099_顏彥臣_hw5.py (3.8.3)

File Edit Format Run Options Window Help

```
    print('Train/Test average score: {:.3f}/{:.3f}'.format(average_train, average_test))
        #5 fold
    ir5= model.fit(X_5fold, y_5fold)
    fold5_s=model.score(X_5fold,  y_5fold)
    check_s=model.score(X_check,  y_check)
    return fold5_s,check_s
##########################################################
print("109611099_顏彥臣_HW5_cancer.txt")
X,y=readHw5Cancer("C:\\prog\\ML\\hw5\\hw5_cancer.csv")
X_5fold, X_check, y_5fold, y_check = train_test_split(X, y, test_size = 94, random_state = 0)
#分割
g_numRec = 475
num_folds = 5
test_size = int(g_numRec * (1.0/num_folds))
# loop through folds
total_train = 0
total_test = 0

for fold in range(num_folds):
    X_train, X_test, y_train, y_test = TrainTestSplit_Fold(X_5fold, y_5fold, fold, test_size)
    logreg=LogisticRegression(max_iter=10000).fit(X_train,y_train)
    train_s = logreg.score(X_train, y_train)
    test_s  = logreg.score(X_test,  y_test)
    total_train += train_s
    total_test  += test_s

    #5 fold
    logreg5= LogisticRegression(max_iter=10000).fit(X_5fold, y_5fold)
    fold5_s=logreg5.score(X_5fold,  y_5fold)
    check_s=logreg5.score(X_check,  y_check)
average_train = total_train/num_folds
average_test  = total_test/num_folds
print("(a) LogisticRegression(max_iter=10000)")
print("Train/Test average score: {:.3f}/{:.3f}".format(average_train, average_test))
print("5-fold/verify score: {:.3f}/{:.3f}".format(fold5_s, check_s))
#Randomforest
print("(b)Randomforest (random state=0)")
forest=RandomForestClassifier(random_state=0)
tr1,tr2=Train_model_5fold(forest,X_5fold,y_5fold,X_check,y_check )
print("5-fold/verify score: {:.3f}/{:.3f}".format(tr1,tr2))
#Gradient Boosted Regression Trees
print("(c)Gradient Boosted Regression Trees (random state=0)")

gbrt=GradientBoostingClassifier(random_state=0)
tr1,tr2=Train_model_5fold(gbrt,X_5fold,y_5fold,X_check,y_check )
print("5-fold/verify score: {:.3f}/{:.3f}".format(tr1,tr2))
```

Result:

Python 3.8.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.3 (default, Jul  2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/prog/ML/hw5/109611099_顏彥臣_hw5.py ==========
109611099_顏彥臣_HW5_cancer.txt
(a) LogisticRegression(max_iter=10000)
Train/Test average score: 0.966/0.958
5-fold/verify score: 0.968/0.926
(b)Randomforest (random state=0)
Train/Test average score: 1.000/0.966
5-fold/verify score: 1.000/0.947
(c)Gradient Boosted Regression Trees (random state=0)
Train/Test average score: 1.000/0.973
5-fold/verify score: 1.000/0.936
>>>
```

Ln: 13 Col: 21