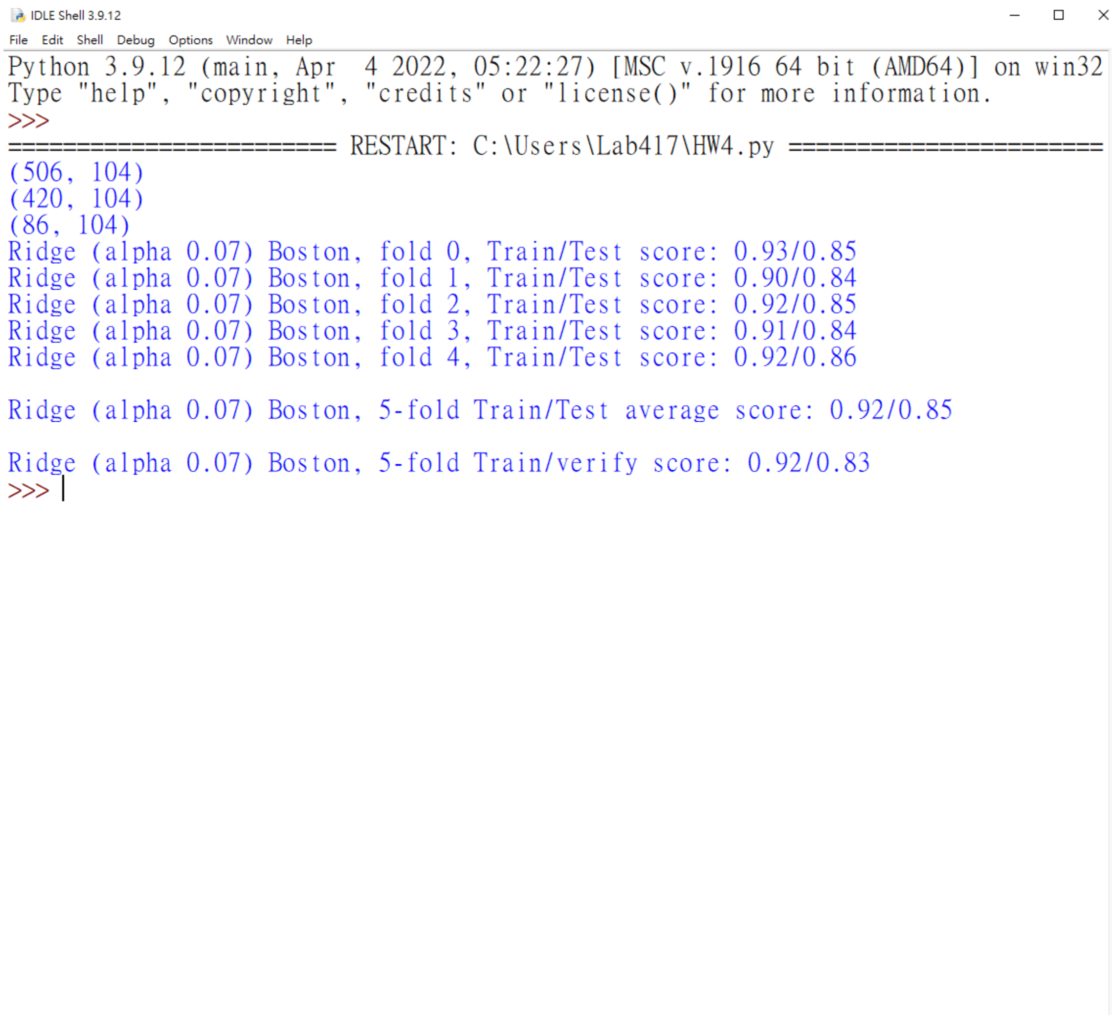
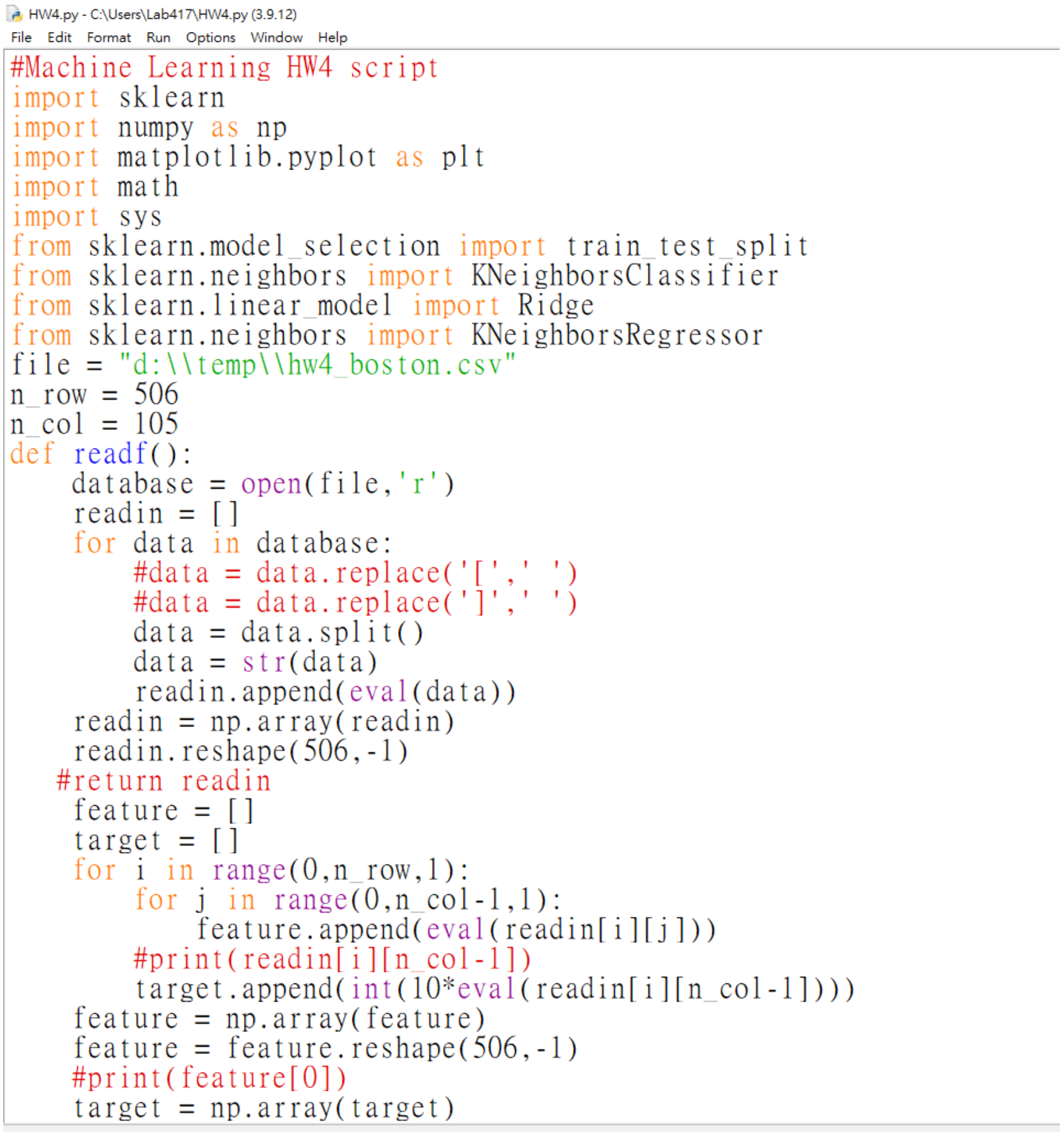
Compile Result



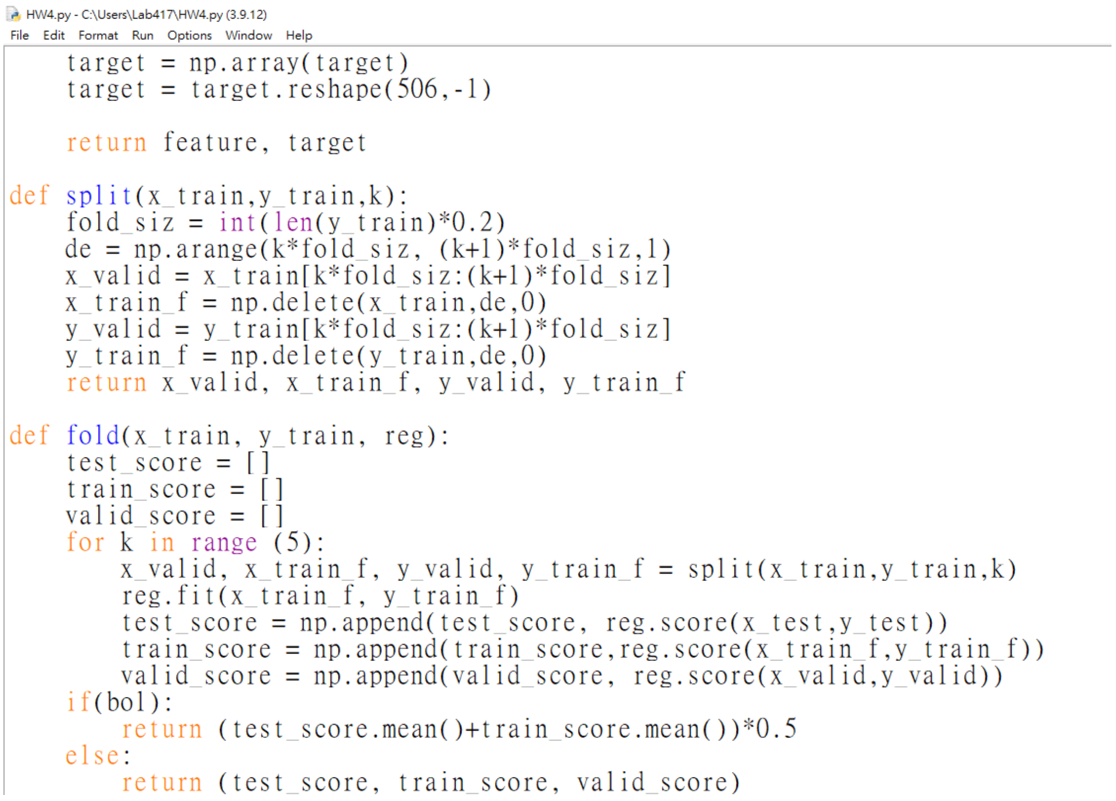
IDLE Shell 3.9.12

```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
======================== RESTART: C:\Users\Lab417\HW4.py =======================
(506, 104)
(420, 104)
(86, 104)
Ridge (alpha 0.07) Boston, fold 0, Train/Test score: 0.93/0.85
Ridge (alpha 0.07) Boston, fold 1, Train/Test score: 0.90/0.84
Ridge (alpha 0.07) Boston, fold 2, Train/Test score: 0.92/0.85
Ridge (alpha 0.07) Boston, fold 3, Train/Test score: 0.91/0.84
Ridge (alpha 0.07) Boston, fold 4, Train/Test score: 0.92/0.86

Ridge (alpha 0.07) Boston, 5-fold Train/Test average score: 0.92/0.85

Ridge (alpha 0.07) Boston, 5-fold Train/verify score: 0.92/0.83
>>> 
```

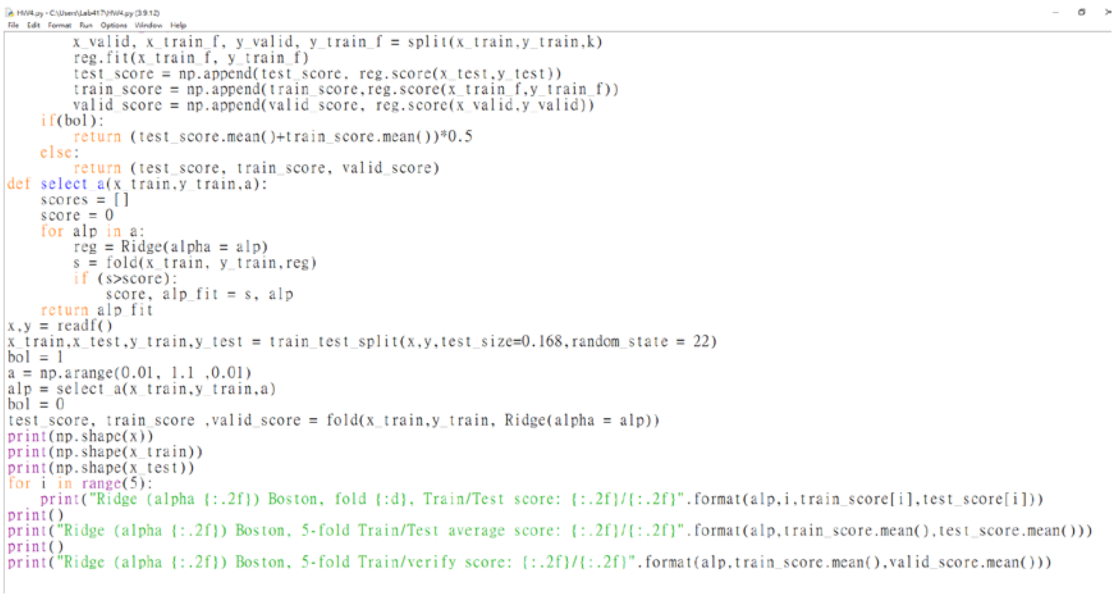
程式碼內容


HW4.py - C:\Users\Lab417\HW4.py (3.9.12)
File Edit Format Run Options Window Help

```
#Machine Learning HW4 script
import sklearn
import numpy as np
import matplotlib.pyplot as plt
import math
import sys
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import Ridge
from sklearn.neighbors import KNeighborsRegressor
file = "d:\\temp\\hw4_boston.csv"
n_row = 506
n_col = 105
def readf():
    database = open(file,'r')
    readin = []
    for data in database:
        #data = data.replace('[',' ')
        #data = data.replace(']',' ')
        data = data.split()
        data = str(data)
        readin.append(eval(data))
    readin = np.array(readin)
    readin.reshape(506,-1)
   #return readin
    feature = []
    target = []
    for i in range(0,n_row,1):
        for j in range(0,n_col-1,1):
            feature.append(eval(readin[i][j]))
        #print(readin[i][n_col-1])
        target.append(int(10*eval(readin[i][n_col-1])))
    feature = np.array(feature)
    feature = feature.reshape(506,-1)
    #print(feature[0])
    target = np.array(target)
```


HW4.py - C:\Users\Lab417\HW4.py (3.9.12)
File Edit Format Run Options Window Help

```python
    target = np.array(target)
    target = target.reshape(506,-1)

    return feature, target

def split(x_train,y_train,k):
    fold_siz = int(len(y_train)*0.2)
    de = np.arange(k*fold_siz, (k+1)*fold_siz,1)
    x_valid = x_train[k*fold_siz:(k+1)*fold_siz]
    x_train_f = np.delete(x_train,de,0)
    y_valid = y_train[k*fold_siz:(k+1)*fold_siz]
    y_train_f = np.delete(y_train,de,0)
    return x_valid, x_train_f, y_valid, y_train_f

def fold(x_train, y_train, reg):
    test_score = []
    train_score = []
    valid_score = []
    for k in range (5):
        x_valid, x_train_f, y_valid, y_train_f = split(x_train,y_train,k)
        reg.fit(x_train_f, y_train_f)
        test_score = np.append(test_score, reg.score(x_test,y_test))
        train_score = np.append(train_score,reg.score(x_train_f,y_train_f))
        valid_score = np.append(valid_score, reg.score(x_valid,y_valid))
    if(bol):
        return (test_score.mean()+train_score.mean())*0.5
    else:
        return (test_score, train_score, valid_score)
```


HW4.py - C:\Users\Lab417\HW4.py (3.9.12)
File Edit Format Run Options Window Help

```python
        x_valid, x_train_f, y_valid, y_train_f = split(x_train,y_train,k)
        reg.fit(x_train_f, y_train_f)
        test_score = np.append(test_score, reg.score(x_test,y_test))
        train_score = np.append(train_score,reg.score(x_train_f,y_train_f))
        valid_score = np.append(valid_score, reg.score(x_valid,y_valid))
    if(bol):
        return (test_score.mean()+train_score.mean())*0.5
    else:
        return (test_score, train_score, valid_score)
def select_a(x_train,y_train,a):
    scores = []
    score = 0
    for alp in a:
        reg = Ridge(alpha = alp)
        s = fold(x_train, y_train,reg)
        if (s>score):
            score, alp_fit = s, alp
    return alp_fit
x,y = readf()
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.168,random_state = 22)
bol = 1
a = np.arange(0.01, 1.1 ,0.01)
alp = select_a(x_train,y_train,a)
bol = 0
test_score, train_score ,valid_score = fold(x_train,y_train, Ridge(alpha = alp))
print(np.shape(x))
print(np.shape(x_train))
print(np.shape(x_test))
for i in range(5):
    print("Ridge (alpha {:.2f}) Boston, fold {:d}, Train/Test score: {:.2f}/{:.2f}".format(alp,i,train_score[i],test_score[i]))
print()
print("Ridge (alpha {:.2f}) Boston, 5-fold Train/Test average score: {:.2f}/{:.2f}".format(alp,train_score.mean(),test_score.mean()))
print()
print("Ridge (alpha {:.2f}) Boston, 5-fold Train/verify score: {:.2f}/{:.2f}".format(alp,train_score.mean(),valid_score.mean()))
```