學號：0653440

一、載入的套件：

```
import pandas as pd
import numpy as np
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
```

二、載入資料表：

```
csv = pd.read_csv("data.csv",encoding = "ISO-8859-1")
```

三、資料處理：

1. 決定各欄去留，及是否要轉 Dummy 及非數值的處理：

```
#age
csv = pd.get_dummies(csv, columns=['workclass'])#2
#fnlwgt
csv = pd.get_dummies(csv, columns=['education'])#4
csv = pd.get_dummies(csv, columns=['education_num'])#5
csv = pd.get_dummies(csv, columns=['marital_status'])#6
csv = pd.get_dummies(csv, columns=['occupation'])#7
csv = pd.get_dummies(csv, columns=['relationship'])#8
csv = pd.get_dummies(csv, columns=['race'])#9
csv = pd.get_dummies(csv, columns=['sex'])#10
csv = pd.get_dummies(csv, columns=['capital_gain'])#11
csv = pd.get_dummies(csv, columns=['capital_loss'])#12
#hour_per_week
#csv = csv[csv['native_country'].str.contains(" ?") == False]#14
csv = pd.get_dummies(csv, columns=['native_country'])#12
#csv=csv.drop('native_country',axis = 1)#11
csv['income'] = np.where(csv.income.isin([' <=50K']),'0', csv['income'])# <=50K=0
csv['income'] = np.where(csv.income.isin([' >50K']),'1', csv['income'])#  >50K=1
```

說明：

(1) workclass、education、education_num、marital_status、occupation、relationship、race、sex、capital_gain、capital_loss、native_country: 轉 Dummy。

(2) fnlwgt、hour_per_week、原數值保留。

(3) income 為 target 的部分，將答案為<=50K 的設為 0，>50K 的設為 1。

2. 設定 data 及 target：

```
data=csv.drop('income',axis = 1)
data=np.array(data)

target=csv[['income']]
target=np.array(target)
```

二、hw_K_fold function：

```
def hw_K_fold(k, data,target):
    ksize=data.shape[0]//k
    Accuracy=0
    nowindex=0
    for i in range(k):
        X_train= np.concatenate((data[0:nowindex],data[nowindex+ksize+1:]), axis=0)
        X_test = data[nowindex:nowindex+ksize+1]
        Y_train= np.concatenate((target[0:nowindex],target[nowindex+ksize+1:]), axis=0)
        Y_test = target[nowindex:nowindex+ksize+1]


        regr = GradientBoostingClassifier()
        regr.fit(X_train, Y_train)
        Y_pred = regr.predict(X_test)
        thisacc=accuracy_score(Y_test, Y_pred)
        Accuracy+=thisacc
        print('accuracy',i+1,'=',thisacc)
        nowindex+=ksize

    return Accuracy/k
```

1. ksize：每個 data set 的大小
2. Accuracy：正確率的累加。
3. nowindex：test set 的起始 index。

三、call function： print(hw_K_fold(10, data,target))

四、結果：

```
accuracy 1 = 0.851089960086
accuracy 2 = 0.867669634633
accuracy 3 = 0.85937979736
accuracy 4 = 0.848326680995
accuracy 5 = 0.861836045441
accuracy 6 = 0.86030089039
accuracy 7 = 0.857844642309
accuracy 8 = 0.861836045441
accuracy 9 = 0.864599324532
accuracy 10 = 0.857537611299
0.859042063248
```