1. 以 income 作為 label / 2-a. 資料前處理(dummy, normalize)

```
data = pd.read_csv('D://data.csv')
data = pd.get_dummies(data,
        columns= ['workclass','education', 'marital_status','occupation',
        'relationship','race','sex','native_country'])
for i in ['age','fnlwgt','education_num','capital_gain', 'capital_loss',
        'hours_per_week']:
    temp = [data[i].values]
    data[i] = preprocessing.normalize(temp)[0]

data['income'] = data['income'].replace({' <=50K':0," >50K":1})
```

Index	age	fnlwgt	ducation_nur	capital_gain	capital_loss	ours_per_wee	income	Workclass_ ?	lass_ Federa
0	0.00528154	0.00197821	0.00692473	0.00161426	0	0.00524289	0	0	0
1	0.0067712	0.0021261	0.00692473	0	0	0.00170394	0	0	0
2	0.00514611	0.00550329	0.00479405	0	0	0.00524289	0	0	0
3	0.00717747	0.00599009	0.0037287	0	0	0.00524289	0	0	0
4	0.00379187	0.00863621	0.00692473	0	0	0.00524289	0	0	0
5	0.00501069	0.00726254	0.0074574	0	0	0.00524289	0	0	0
6	0.00663578	0.00408798	0.00266336	0	0	0.00209716	0	0	0

2-b. 請使用 Gradient Boosting 進行分類 / 2-c. 請寫自行撰寫 function 進行 k-fold cross-validation(不可使用套件)並計算 Accuracy

```
def K_fold_CV(k, data):
    size = data.shape[0]//k
    acc=[]
    for i in range(k):
        test_set = data[i*size:(i+1)*size]
        train_set =
pd.concat([data[0:i*size],data[(i+1)*size:]],ignore_index=True)
        X_train = train_set.drop(['income'],axis=1)
        Y_train = train_set['income']
        X_test = test_set.drop(['income'],axis=1)
        Y_test = test_set['income']
        GDBT = GradientBoostingClassifier()
        GDBT.fit(X_train, Y_train)
        acc.append(GDBT.score(X_test,Y_test))
    print(acc)
    return np.mean(acc)
```

```
[0.8568796068796068, 0.8676289926289926, 0.870085995085995,
0.8584152334152334, 0.8688574938574939, 0.8682432432432432,
0.8660933660933661, 0.871007371007371, 0.8691646191646192,
0.8627149877149877]
```

3. 請計算 k=10 的 Accuracy，並上傳程式碼與報告

```
print ('Mean accuracy of 10-fold CrossValidation：',K_fold_CV(10, data))
```

```
Mean accuracy of 10-fold CrossValidation：  0.8659090909090909
```