Homework 2 Report - Income Prediction

學號：r06921038 系級：電機所碩一 姓名：謝宗宏

1. (1%) 請比較你實作的 generative model、logistic regression 的準確率，何者較佳？

   logistic regression 得到的 public score 為 0.85761 、private score 為 0.84842。

   generative model 得到的 public score 為 0.84557、 private score 為 0.84191。

   由 logistic regression 得到的準確率較佳，可能原因是，generative model 對於 data distribution 的假設其實並不準確 (Gaussian distribution)，而且由於這次的實作我讓不同的 classes 共用了 covariance matrix 的參數，有可能限制了 generative model 的表達能力。

2. (1%) 請說明你實作的 best model，其訓練方式和準確率為何？

   使用 feature normalization (mean=0, std=1) , L2 regularization (lambda=32), 使用 adagrad。 比較不同的是我總共 train 了兩次，第一次的 weight 初始化成 0, learning rate=0.15, iteration=10000。 第二次從第一次 training 得到的 weight 開始, learning rate=0.0001, iteration=30000。

   準確率 (public, private) = (0.85761, 0.84842)

3. (1%) 請實作輸入特徵標準化(feature normalization)，並討論其對於你的模型準確率的影響。(有關 normalization 請參考：https://goo.gl/XBM3aE)

   共同設定： lambda=32, error tolerance=0.0001, stochastic average gradient descent, max iteration=10000。

   沒有使用 feature normalization 時：

   10-fold cross validation 的平均 accuracy 為 0.79576。

   kaggle (public, private) = (0.80061 , 0.79511)

   使用 feature normalization 時：

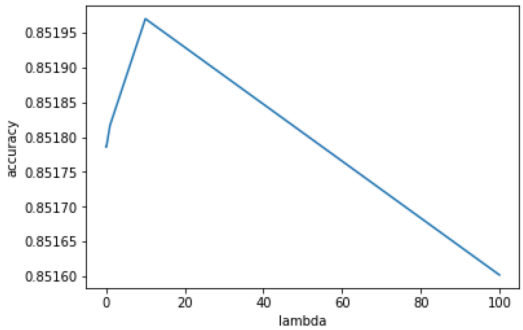
   10-fold cross validation 的平均 accuracy 為 0.852523

   kaggle (public, private) = (0.85761, 0.84842)

   可推測使用 feature normalization 對於準確率的提升確實有幫助。

4. (1%) 請實作 logistic regression 的正規化(regularization), 並討論其對於你的模型準確率的影響。(有關 regularization 請參考：https://goo.gl/SSWGhf P.35)

使用 L2 regularization 。以下為 lambda (regularization 參數) 對於 10-fold cross validation 的 accuracy 作圖 。lambda = [0, 0.1, 1, 10, 100] 。可看出 lambda 大約在 10 這個數量級附近時的準確率最高，也比不使用 regularization (lambda=0) 時的準確率高。此外 regularization 過小或過大 performance 都會下降 。



5. (1%) 請討論你認為哪個 attribute 對結果影響最大？

從兩個面向來考慮：

(a.)我使用了 recursive feature elimination ， 逐步刪掉影響力較低的 features ， 而當決定最後只留下一個 feature 時，留下的 feature 為 ‘capital_gain’ 因此該 feature 應為影響最大的 feature 。

(b.)此外 ， 若是從 training 的參數來看 ， weight (代表此參數的影響力)最大的 feature 在 index 78，對應到的參數也確實是 ‘capital_gain’ 。

```
import pandas as pd
df = pd.read_csv('dataset/train_X')
df.columns[np.argmax(lrcv.coef_)]
```

```
'capital_gain'
```