學號：B04902092 系級： 資工三 姓名：張均銘

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

feature:所有 106 feature 都選

logisitic model : iteration 10000 次，初始 weight:0.0 learning rate = 0.05

	Train data rmse	Public rmse	Private rmse
Generative model	0.842357	0.84606	0.84166
Logistic model	0.853352	0.85417	0.85087

準確率以 Logistic model 比較好

2.請說明你實作的 best model，其訓練方式和準確率為何？

答：

train rmse: 0.859679

public rmse:0.86031

private rmse:0.85603

訓練方式：

將 train_x data 讀出以後自己做 normalized，test_x 讀出做 normalized，是個別做，不是一起做，train 時取所有 data 總共 106 feature 加上 age, sex,capital_gain, capital_loss 這幾個 feature 的 ln 項，所以總共 110 項 feature 做 logistic regression learning rate 設 0.1,iterate 5000 次

3.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。

Unnormalized model	Train data rmse	Public rmse	Private rmse
Generative model	0.842419	0.84582	0.84240
Logistic model	0.617241	0.61916	0.61454

在除了 Generative 在 training 的 rmse 有比較高以外，其他都下降的，但是在 generative 裡面影響其實並沒有那麼明顯，Logistic 反而就差超多，而在 iteration 10000 次的過程中，每 1000 次我有印出他當時的 rmse，發現其實 train 的過程中除了幾次釣到 0.3 左右，大部分次數也都維持在 0.7，但是出來的結果卻只剩 0.6。

Generative model 影響不明顯的原因應該是因為主要是看分佈，即使有的參數只分布在 0-1 間，有的像 age 分不到 20-70 之類的，雖然差距很大，但因為只看分佈，不會說需要拿 weight 來乘，所以比較不會有 age 之類的 feature 比重過重的問題。

Logistic 則是因為 age, fnlwg 的值相對其他大，所以可能這兩個參數沒 normalized 影響整個 model 很多。

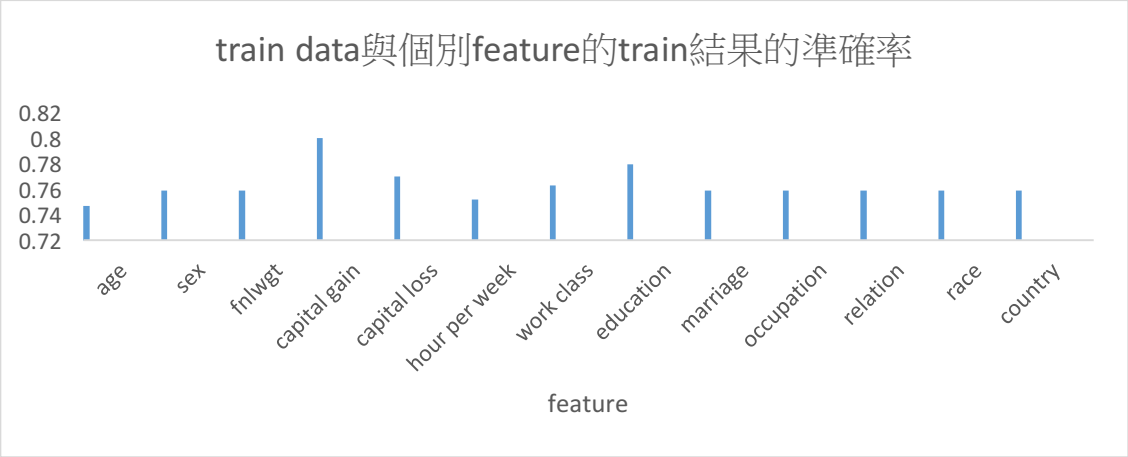
4. 請實作 logistic regression 的正規化(regularization)，並討論其對於你的模型準確率的影響。

答：

Rate/score	Train data	Public data	Private data
0.00001	0.831148	0.83316	0.85100
0.0001	0.85466	0.853997	0.85124
0.001	0.850650	0.85257	0.84756
0.1	0.831148	0.83316	0.82496

模型:logistic regression(iteration:10000)，有 normalized
以 train data set 而言，最低的 regularization rate 有最高的準確率(0.85466),
然後 rate 在 0.01 和 0.1 時準確率其實沒比直接 train 的高，但是 rate 在 0.001 時準確率卻比 train 高一些但是在 0.00001 時又掉下去
而在 public test data 中，普遍不 regularization 的準確率比較高

5.請討論你認為哪個 **attribute** 對結果影響最大？



train 的方式採用 Logistic Regression 並有做 normalized 和 adagrad，
bias = 0.001，weight = 0(初始值)
train 不連續 feature 方式(如 race….):
取 X_train 該 feature 的 one_hot encoding 的部分全部拿來 train，例如國家有 20 分成 20 幾個 one_hot encoding 就把他抓來，等於 model 的參數有 20 幾個每個分配都是 0-1 如果只拿一種 attribute 來 train 的話，capital_gain 擁有最高的準確率，其中很多的輸出其實都是 0.759190, 簡單來說就是 train 的結果在拿 train data 來跑，model 判斷的結果就是所有 sample 都是<50K（全部小於 50K 準確率是 0.759190），尤其集中在後面幾個 feature 更加明顯，其實原因也很明顯，因為後面幾筆資料都是不連續，都只有 0,1 的分佈，而且 0 的機率又遠高於其他，畢竟如國家有在 X_train 裡面拆成 20 幾筆 feature，但一個人只可能屬於一個國家，代表說這 20 幾個 feature 每個人只會有一個是 1，其他都是 0，因此 0 居多的狀況下，train 出來結果當然就都是 0，所以我這種 train 法其實不適合 on_hot encoding 的 train 法，或許比較能準確表現每個 feature 的準確率的方法是：把不連續的 feature 的所有可能值對應到一個整數，讓他變成半連續的 feature 那這樣至少 train 時不會因為有很多 0 或 normalized 後非常接近 0 而導致輸出都是 0