

1.請說明你實作的 **generative model**，其訓練方式和準確率為何？

答：

這次我實作的 **generative model**，主要是利用老師投影片上的數學推導式，算出 weight 和 bias。

首先先將 training\_data 分為  $\leq 50K$  和  $> 50K$  兩個 class，再分別算出兩個 class 的 mean 和 variance，又因為將兩個 class 的 covariance matrix 共用的話要訓練的參數可以少很多，進而改善訓練的效果，所以將 covariance 分別乘上 training data 的比例。

$$\text{weight} = (\mu^1 - \mu^2)^T \Sigma^{-1} x$$

$$\text{bias} = -0.5 * (\mu^1)^T \Sigma^{-1} \mu^1 + 0.5 * (\mu^2)^T \Sigma^{-1} \mu^2 + \ln \frac{N_1}{N_2}$$

讓上面式子算出來的 weight, bias 當作 model 的最後參數，在 Kaggle 上面所得到的準確率為 0.83796。

2.請說明你實作的 **discriminative model**，其訓練方式和準確率為何？

答：

實作 **discriminative model** 則是類似 hw1 實作 linear regression model，利用 gradient descent 的方法，去找到最後訓練完的 weight 和 bias，比較不同於 linear regression 的是 logistic regression 要通過一次 sigmoid function。

在 **discriminative model**，我初始的 learning rate 是 0.5，weight 和 bias 都是初始值為 0，最後在 Kaggle 上面所得到的準確率為 0.85663。

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。

答：

最剛開始沒有做 feature scaling，有些 feature 因為數值較大的緣故，在做 exponential 的時候會 overflow，而在 Kaggle 上的準確率為 0.78305，做完 feature scaling 後，cross entropy 的收斂速度快了好幾倍，我想是因為我 weight 對於每個 feature 的初始值都一樣，所以如果沒做 feature scaling 的話，有些 weight 收斂的速度會很慢，另外做了 feature scaling 就不會發生 overflow 的問題，且在 Kaggle 上的準確率為 0.85663。

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：

在 logistic regression model 裡加了 regularization 後，我發現對於模型準確率的幫助並沒有很大，且若選取三次方、四次方的 feature，加了 regularization 後反而會讓結果稍微的變差，我想這是因為 regularization 本身是希望讓所有的 weight 平均的移動，

所以反而一些比較特別的 feature 需要比較大的 weight 的時候，regularization 會造成反效果。

在我最後選定的 feature 中，如果做了 regularization，在 Kaggle 上的準確率為 0.85491，若不選擇 regularization，在 Kaggle 上的準確率為 0.85663。

5.請討論你認為哪個 **attribute** 對結果影響最大？

經過了好幾次試驗之後，我發現是 data 裡的”capital\_gain”這項 attribute 對結果影響最大，如果將這個 feature 做二次方、三次方，分別變為另一個 feature，這樣對結果有很顯著的幫助，另外我是透過剛開始都是一次方的 feature，去找出最後收斂的時候 weight 的分布，發現”capital\_gain”這項 feature 所對應到的 weight 最大，所以我認為應該是對結果影響最大的一項。