

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

| 準確度 | generative | logistic |
|----------------------|------------|----------|
| training data | 0.76064 | 0.85884 |
| public testing data | 0.76523 | 0.85909 |
| private testing data | 0.76231 | 0.85542 |

無論是在 training 還是 testing data，logistic model 的表現都優於 generative model，我想是因為在實作 generative model 時我沒有刻意篩選優秀的 feature。

就這次的 data 而言，generative model 優點在於簡單且快速，logistic model 則是需要去不斷調整參數，用較長時間去訓練，才能得到較高的準確率。

2.請說明你實作的 best model，其訓練方式和準確率為何？

使用 training data 的每一個 feature 及(age,capital_gain,capital_loss,hours_per_week)的 0.5~9 次方（所有資料在 training 前都經過 normalization），使用 logistic regression，learning rate: 1.0，1600 epochs，正規化 lambda=0.1。

❖ 我嘗試加入各種 feature 儘量使 training data 的準確度提高，先不考慮 overfitting。

| training data | public testing data | private testing data |
|---------------|---------------------|----------------------|
| 0.85884 | 0.85909 | 0.85542 |

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。

(a) 以我的 logistic regression 來說，若沒有標準化，高次方的 feature 有可能會 overflow，或是各種誤差導致準確率下降。

```
def divide_max(ipt):  
    ipt=np.array(ipt,dtype=float)  
    rec_max=ipt.max(axis=0)  
    for i in range(ipt.shape[0]):  
        for j in range(ipt.shape[1]):  
            ipt[i][j]=float(ipt[i][j]/rec_max[j])
```

| 執行 200 次 epochs | normalize | without normalize |
|----------------------|-----------|----------------------|
| training data | 0.8533 | 0.7657 |
| public testing data | 0.8550 | 0.7651 |
| private testing data | 0.8539 | 0.7671 |

(b) 以我的 generative model 來說，反而是沒有標準化準確度比較高

```
def normalize(b):
    array=np.array(b,dtype=float)
    row_means = np.mean(array, axis=0)
    row_std = np.std(array, axis=1)
    for i in range(array.shape[0]):
        for j in range(array.shape[1]):
            if not row_std[i]== 0 :
                array[i][j] = (array[i][j]- row_means[j]) / row_std[j]
    return array
```

| | normalize | without normalize |
|----------------------|-----------|-------------------|
| training data | 0.7606 | 0.6322(爆炸) |
| public testing data | 0.7652 | 0.8236 |
| private testing data | 0.7623 | 0.8252 |

❖ 由(a)(b)可得知，有沒有標準化沒有絕對的好壞，可能會有精度問題，或是使原始資料某些性質不見，但好處是可以增加訓練的速度。

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。

```
theta -= learning_rate * ( (-1) * add + 2 * lamda*theta)/(tot**0.5)
```

❖ 將 loss function 加上 x^2 項，使 model 參數不會太大

使用 best model 來比較 lambda，準確度（執行 200 次 epochs）：

| lambda | 1 | 0.1 | 0.01 | 0 |
|----------------------|--------|--------|--------|--------|
| training data | 0.8522 | 0.8533 | 0.8536 | 0.8538 |
| public testing data | 0.8542 | 0.8550 | 0.8548 | 0.8548 |
| private testing data | 0.8532 | 0.8539 | 0.8541 | 0.8542 |

❖ 由上表可知正規化對於我這題的 model 的準確度並沒有幫助，有可能是因為本題 training data 和 testing data 相似，noise 不多。

5.請討論你認為哪個 attribute 對結果影響最大？

我透過加一個，少一個 feature 的方式來決定要不要選某一個 feature

以 training data 準確度來觀察（執行 200 次 epochs）：

| | |
|-------------------|--------|
| all feature | 0.8446 |
| no age | 0.8444 |
| no fnlwgt | 0.8451 |
| no sex | 0.8537 |
| no capital_gain | 0.8424 |
| no capital_loss | 0.8442 |
| no hours_per_week | 0.8453 |

❖ 對結果影響最大的 feature 為 capital_gain