學號：R08946006 系級：資料科學學程碩一 姓名：周逸平

1. **(2%) 請比較實作的 generative model 及 logistic regression 的準確率，何者較佳？請解釋為何有這種情況？**

在 Kaggle 上，generative model 以及 logistic regression 所得到的準確率如下 (皆以 tutorial 參數為基準)：

Approach	ACC
Generative model	0.88110
Logistic regression	0.88670

可以看到 Logistic 的表現較 Generative 要好，而我認為的理由為，由於 generative model 需要對樣本的分群作假設，而 tutorial 上假設兩者皆服從常態分配，但我們沒有證據證明這個假設有真的符合實際狀況。又，此次給予的資料 (X_train.csv) 中有多數為 dummy variable，使得變數數量高達 510 項，這樣的高維度空間也會讓分布的預測變得更加困難。

2. **(2%) 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。接著嘗試對正規項使用不同的權重 (lambda)，並討論其影響。(有官 regularization 請參考 https://goo.gl/SSWGhf p.35)**

以下實驗皆使用此組參數:

max_iter = 2000

batch_size = 5000

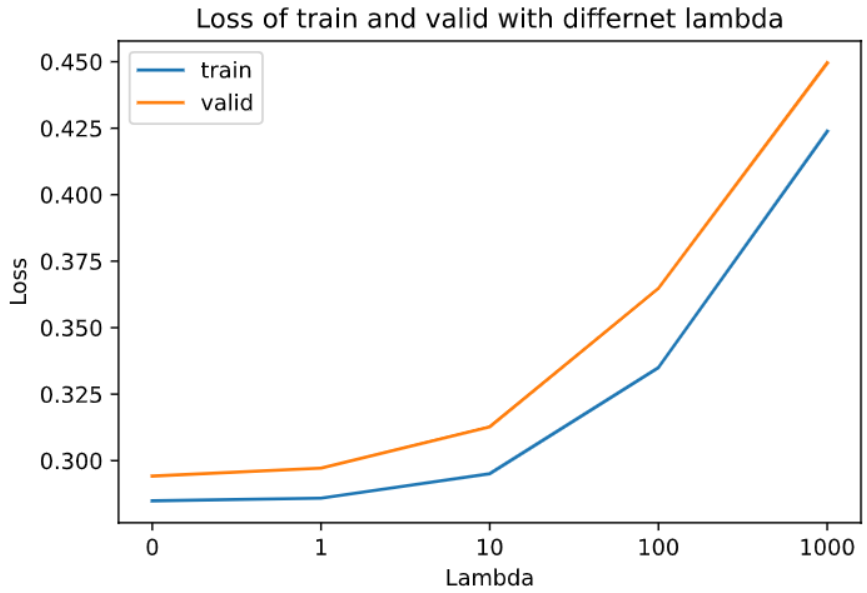
learning_rate = 0.01

公式參考:

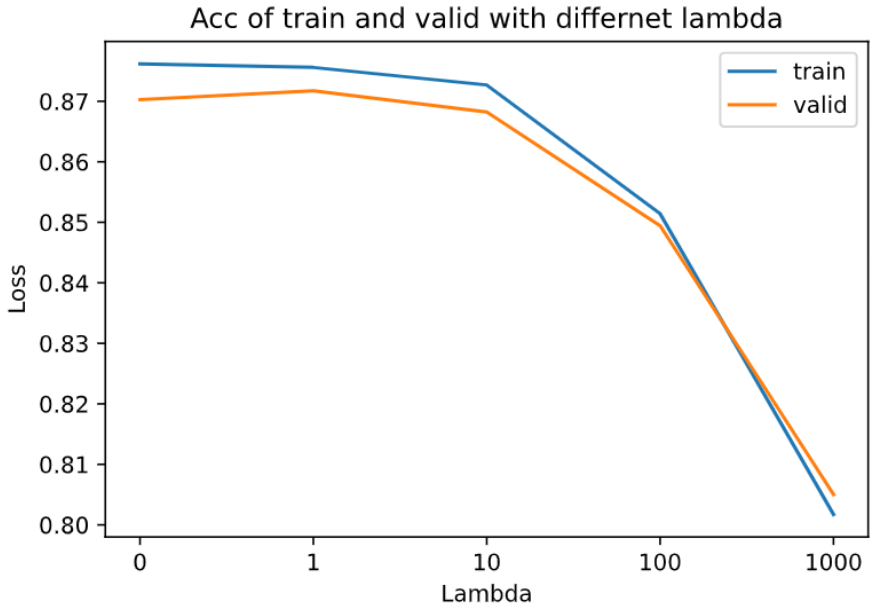
smaller w_i are better

$$L = \sum_n \left(\hat{y}^n - \left(b + \sum w_i x_i \right) \right)^2 + \lambda \sum (w_i)^2$$

Loss



Acc



可以發現以此資料集，使用 regularization 反而造成更高的 loss 以及更低的 Acc，可見 regularization 不見得適用每一種情況，下面兩張圖分別秀出在 lambda = 1000 以及 1 情況下的權重

1000

```
array([ 1.03256847e-01,  1.84062793e-01,  1.31359811e-01,  7.12746907e-02,
        6.35021853e-02,  1.01272620e-01,  9.26374351e-02, -9.38211660e-02,
        1.41854436e-02,  1.32433133e-01,  1.01263726e-01,  4.34981603e-02,
        6.21542715e-02,  3.60027284e-05,  3.74417412e-03,  1.32633243e-02,
       -1.56780804e-02, -5.54327061e-02,  9.83846692e-03,  2.37962803e-02,
        4.12384328e-02,  2.74468789e-03,  2.74468789e-03,  1.70282682e-02,
        4.71659164e-03,  3.04734568e-02,  1.27040806e-02, -2.10100815e-02,
        6.73962858e-02, -4.86131847e-02, -2.83616595e-02, -2.21919689e-02,
       -2.21392480e-02,  7.48505619e-02,  2.86701132e-04,  7.89798046e-03,
        1.19066944e-02,  4.55987985e-03,  1.64167458e-02,  8.18809380e-02,
        1.95475609e-03, -2.91189236e-02, -2.54856940e-03,  1.33997004e-03,
        2.22340115e-02, -4.20923909e-03,  2.26594713e-04,  1.03418736e-02,
        1.64505365e-02,  4.81335823e-03,  1.06620228e-02,  1.07426127e-02,
        5.81110139e-02])
```

1

```
 array([ 0.70743053,  0.79242714,  1.19045091,  0.46980829,  1.64264935,
         0.71628789, -0.64617099, -1.78748921,  0.37225985,  0.35265372,
         0.23649405,  1.31700001,  0.76990379, -1.19152993, -1.20241594,
        -0.26926242,  0.15545677, -0.20071023,  0.99320591,  1.63400954,
         0.098519  , -0.05884352, -0.05884352, -0.04926875,  0.54665542,
         0.51245793,  0.20255851, -0.21764939, -0.73122777, -1.69732898,
        -0.44625347,  0.04011129,  0.26549392,  0.67135666, -0.7869615 ,
         0.43201441, -0.05210365,  0.22971635, -0.1005175 ,  0.34511299,
         0.52922272, -1.190883  , -0.12125716,  0.30844459,  0.34088828,
         0.2199706 ,  1.95521851,  0.23592422,  0.1027962 , -0.46282415,
         0.44945283, -0.1573284 ,  0.9842006 ])
```

可見由於 regularization 的性質，使得幾乎所有權重在高 lambda 之下都變得非常小，使得 logistic regression 的各項參數起到的作用都偏小，甚至將某些參數都縮小至靠近 0，使得 fitting 上較他者高 loss。

3. (1%) 請說明你實作的 best model，其訓練方式和準確率為何？

我使用了 NN 來完成我的 best model，資料切割如 tutorial，Normalization 的部分我只對非 dummy variable 的項目作 (如：年齡)。最後在 Kaggle 上得到了 0.89131 的成績。

我設計了一個十分簡單的 NN，單純使用數個全連階層配合 dropout 以及 ReLU。

Model 資訊如下：

```
hw2_Model(
  (seq): Sequential(
    (0): Linear(in_features=510, out_features=256, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.6, inplace=False)
    (3): Linear(in_features=256, out_features=256, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.6, inplace=False)
    (6): Linear(in_features=256, out_features=128, bias=True)
    (7): ReLU()
    (8): Dropout(p=0.6, inplace=False)
    (9): Linear(in_features=128, out_features=64, bias=True)
    (10): ReLU()
    (11): Dropout(p=0.4, inplace=False)
    (12): Linear(in_features=64, out_features=32, bias=True)
    (13): ReLU()
    (14): Dropout(p=0.4, inplace=False)
    (15): Linear(in_features=32, out_features=1, bias=True)
  )
)
```

4. **(1%) 請實作輸入特徵標準化 (feature normalization)，並比較是否應用此技巧，會對於你的模型有何影響。**

此題使用以下參數為基準：
max_iter = 10000
batch_size = 5000
learning_rate = 0.01
lambda = 0

Normalization	Train	Valid
0	0.7670079868933033	0.7699963140434942
All	0.8862174892484128	0.8767047548838923

從這個結果可以發現，使用 normalization 對模型的預測是有高度幫助的