

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？

使用助教提供的範例程式 `itr = 10, batch_size = 8, learning_rate = 0.1`

```
Logistic Training accuracy: 0.8836166291214418
Generative Training accuracy: 0.8728066941905043
```

logistic regression 是對於原數據的忠實呈現，並不像 generative model 會事先假設機率分佈，相較之下，為了達到相同的準確率，generative model 需要較多的 train data，因此在相同且有限的樣本數下，logistic regression 的準確率會高於 generative model。

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。

我自己的模型是把沒有做 one-hot encoding 的資料全部做 one-hot encoding

```
lambda = 0 Training accuracy: 0.8899179799096857
lambda = 0.001 Training accuracy: 0.8898949405584738
lambda = 0.01 Training accuracy: 0.8866463920376002
lambda = 0.1 Training accuracy: 0.8740669062759192
lambda = 1 Training accuracy: 0.8174361809971431
lambda = 10 Training accuracy: 0.7943737904340613
lambda = 100 Training accuracy: 0.7943737904340613
```

而下面這張圖是用助教的 code 來測試看看會造成什麼影響

`itr = 100, batch_size = 15, learning_rate = 2e-2`

```
lambda = 0 Training accuracy: 0.8873375725739563
lambda = 0.001 Training accuracy: 0.8872454151691088
lambda = 0.01 Training accuracy: 0.8872454151691088
lambda = 0.1 Training accuracy: 0.8841351027555064
lambda = 1 Training accuracy: 0.8737213160077413
lambda = 10 Training accuracy: 0.8031287438945719
lambda = 100 Training accuracy: 0.7943737904340613
```

由上述數據可知，超過 0.1 後大致上會隨著 lambda 提高而降低，因為要考慮 w 的影響 所以沒辦法把 model fit 的很好。另外猜測也許是 lambda 太高時會使得函數沒辦法完美的呈現最佳的分類函數。而且並不是所有的係數都是越小越好，看過資料發現某些 feature 有非常大的影響（例如說 age 小於 18，幾乎可以確定年薪不會超過 50000）

3. (1%) 請說明你實作的 **best model**，其訓練方式和準確率為何？
1. 使用 **adagrad**
  2. 將某些資料做 **one-hot encoding**（比如說 **age**, **work week per year** 等）總共 7 筆
  3. 將一些 **features** 刪除（比如說 **not in universe**, ? 等）
  4. 調整 **max\_iter = 500**, **batch\_size = 15**, **learning rate = 2e-2**
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。

由於我自己的模型是把沒有做 **one-hot encoding** 的資料全部做 **one-hot encoding**，因此無法應用此技巧

然而我有拿助教的 **code** 來做比對

**No feature engineering** 是完全沒有做標準化

**Feature engineering** 則是將[0,126,210,211,212,358,507]這七筆做標準化

```
No Feature Engineering Training accuracy: 0.8284046692607003  
Feature Engineering Training accuracy: 0.8850205929870071
```

標準化後的準確率較高，推測是因為這些項[0,126,210,211,212,358,507]數值遠大於 1，所以如果沒有做標準化會使得算該維 **w** 時，一旦數值有所變化會造成很大的差異，進而造成準確率下降。