

# Homework 2 Report - Income Prediction


學號：B05902019 系級：資工二 姓名：蔡青邑

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

	Private	Public
Generative Model	0.84191	0.84557
Logistic Regression	0.84571	0.85479

從Kaggle上的分數來看，我們可以發現，Logistic Regression( $\eta = \lambda = 10^{-6}$ )的準確率略佳於Generative Model。我認為其主要原因應該是因為我們的training data夠大夠多，讓比較依賴既有data的discriminative(logistic regression)做法能獲得較佳的參數。且因為Generative的做法是以假設資料背後的高斯分佈為主，在實際準確率上有可能會受到一定的限制。

(1%) 請說明你實作的best model，其訓練方式和準確率為何？

skl3.csv a day ago by B05902019蔡青邑 ^55, sin cos tan arctan, C = 6500	0.87139	0.87751	
--	---------	---------	---

以上是我best的準確率，主要運用的做法是Logistic Regression。事實上，我用了一個很nonlinear的model去跑我的regression，使用的理由主要是藉由不斷配合cross validation後的實做觀察得到的。除了既有的資料，我把原本的資料裡的"age", "fmlwgt", "capital\_gain", "capital\_loss", "hours\_per\_week"等連續實數的項目從2次方到55次方都concatenate進去我的training set裡面，並且我還對他們取了sin, cos, tan, arctan, 並一起concatenate進去。再者為了節省時間和增加準確率，我有使用sklearn相關模組，並事先對資料做normalization，最後得到我的best。

(1%) 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。(有關normalization請參考：<https://goo.gl/XBM3aE>)

以下是我用十次式實作做到收斂的結果：

	Private	Public
No normalization	0.76808	0.77235
Normalization	0.85898	0.86240

正規化後的準確率明顯比較高，我認為主要是因為這次的資料有許多不同數量級的feature(尤其是連續性的數字資料與跟類別性的資料之間)。所以在做regression的同時如果沒有做normalization，數量級較大的feature的權重很容易被過度高估，導致最後的參數變得偏頗。也因此normalization能有效地改善這個問題。

(1%) 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。

	Private	Public	details
regularization	0.84571	0.85479	$\eta = 10^{-5}, \lambda = 10^{-5}$
without regularization	0.83601	0.84299	$\eta = 10^{-5}, \lambda = 0$

從Kaggle上的分數來看，我們可以發現，有regularization的準確率較高，雖然差距沒有到很大。事實上，我在上傳前用本機的training data validate出來的準確率兩者是差不多的( $\pm 0.0002$ )。可能是因為沒有regularization的狀況下的確產生了一點點的overfitting，而regularization改善了overfitting的問題。而關於差距沒有很明顯的原因，應該是因為我用的是一次式，亦或者我們的training set夠大夠好。

(1%) 請討論你認為哪個attribute對結果影響最大？

我認為是occupation。原因是我將一個一個feature從training set裡踢掉下去做logistic regression，發現去除occupation後的accuracy極低。雖然踢掉relationship或者race的準確率也很低，但是踢掉它們仍會得到比較穩定的model參數，而踢掉occupation的w\_length比較大，依照我之前寫作業時的經驗，這樣的參數比較容易overfitting。

j: 31	w_length: 4.701998	Cost: 0.822272
j: 32	w_length: 6.022619	Cost: 0.822272
j: 33	w_length: 6.488020	Cost: 0.821904
j: 34	w_length: 5.364027	Cost: 0.816191
j: 35	w_length: 6.725383	Cost: 0.816222
j: 36	w_length: 6.500604	Cost: 0.816283
j: 37	w_length: 6.527005	Cost: 0.816222
j: 38	w_length: 81.879213	Cost: 0.815853
j: 39	w_length: 81.963832	Cost: 0.812874
j: 40	w_length: 73.882429	Cost: 0.812321
j: 41	w_length: 53.280145	Cost: 0.810295
j: 42	w_length: 38.812286	Cost: 0.809220
j: 43	w_length: 33.459020	Cost: 0.808974
j: 44	w_length: 29.833266	Cost: 0.808728
j: 45	w_length: 25.725826	Cost: 0.808544
j: 46	w_length: 21.798624	Cost: 0.808206
j: 47	w_length: 20.516082	Cost: 0.807960
j: 48	w_length: 17.772410	Cost: 0.807991
j: 49	w_length: 16.814273	Cost: 0.807715
j: 50	w_length: 15.571020	Cost: 0.807193
j: 51	w_length: 15.227802	Cost: 0.807223
j: 52	w_length: 3.271259	Cost: 0.805012
j: 53	w_length: 3.223025	Cost: 0.805012
j: 54	w_length: 3.207606	Cost: 0.804889
j: 55	w_length: 3.188866	Cost: 0.804951
j: 56	w_length: 3.167024	Cost: 0.805012
j: 57	w_length: 3.113910	Cost: 0.804951

(如圖，那幾筆對應的feature是occupation。)