學號：**B03202006** 系級： 物理四 姓名：陳昱蓁

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

答：

- 兩模型均採用助教抽的 feature：X_train, Y_train, X_test
- 連續的 feature (index/name)：**0/age, 1/fnlwgt, 3/capital_gain, 4/capital_loss, 5/hours_per_week**
- 不連續的 feature：其他

	Generative model	Logistic regression
feature	採用所有的 feature，未做任何的 normalization (猜想 generative model 就是在描述變數的分佈了，做 normalization 也只是線性伸縮+平移，似乎沒有特別的必要)	採用所有的 feature，對於 X_train 中所有連續的 feature 做 normalization（標準化=>$\mu' = 0$, $\sigma' = 1$）
Model discription	✓ Naive Bayes Classifier 連續的 feature： Gaussian distribution 不連續的 feature：直接計算 P(X|C1), P(X|C2)(用出現個數算機率，其中 X 可為 0, 1) ✓ training data：所有	✓ Batch_size = 100 ✓ Gradient descent ✓ Loss function = cross entropy ✓ training data：所有
Training set accuracy	0.80857468	0.85252295691164282
Testing set (Public) accuracy	0.81105	0.85565
Testing set (Private) accuracy	0.80604	0.85210

可以見得 **logistic regression 的準確率較佳**，但我猜想是 generative model 不能把所有 feature 當作獨立，亦即用 Naive Bayes Classifier 可能會有缺陷。像是有些 features 是 label 來自不同國家（for ex: 來自美國，來自中國），很顯然就是相依的 feature。

2.請說明你實作的 **best model**，其訓練方式和準確率為何？

答：

	Logistic regression
feature	✓ 採用所有的 feature，對於 X_train 中所有連續的 feature **做 normalization**（標準化=>$\mu' = 0$, $\sigma = 1$） ✓ 增加 **0/age, 5/hours_per_week** 兩個 feature 的**二次項**（normalized） ✓ 增加 **0/age** 此 feature 的**三次項**（normalized）
Model discription	✓ Batch_size = 100 ✓ Gradient descent + **momentum** ✓ Loss function = cross entropy

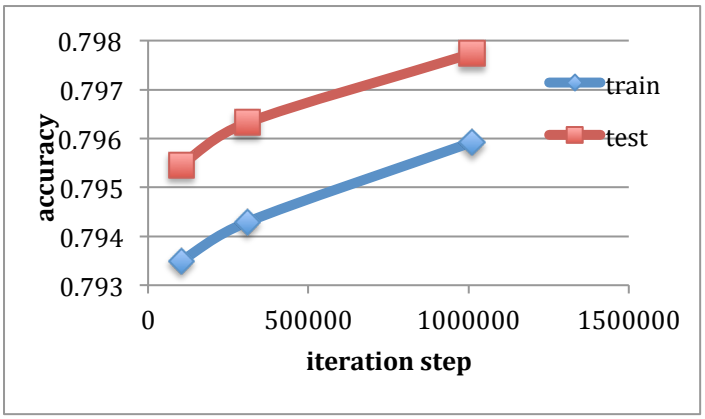
	✓ training data：所有
Training set accuracy	0.8583274469457326
Testing set (Public) accuracy	0.85859
Testing set (Private) accuracy	085493

加入了高次項，在 training set 以及 testing set 上面都有好的表現，因為 logistic regression 實作，相當於 NN 的 one layer，如果不手動加入高次項，整個 function 就是線性的。
加入了 momentum，improvement 並不顯著，和沒有加的狀況再差不多的地方收斂。也許原本的地方就是 global minimum，或是加了 momentum 還是沒有跳出 local minimum。

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：
對於 logistic regression 的模型
（都是選 X_train 的一次項作為 feature）

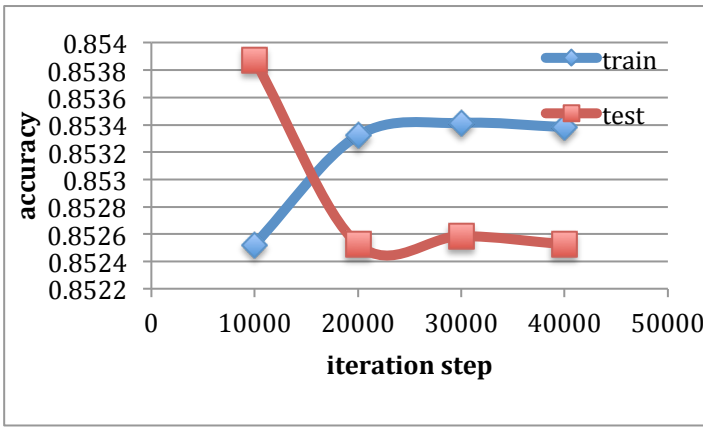
沒有做 normalization

iter	train	public	private	test (avg)
103000	0.793495286	0.79791	0.79302	0.795465
310000	0.794293787	0.79926	0.79339	0.796325
1010000	0.795921501	0.80024	0.79523	0.797735



做了 normalization

iter	train	public	private	test (avg)
10000	0.852522957	0.8521	0.85565	0.853875
20000	0.853321458	0.85356	0.85149	0.852525
30000	0.853413593	0.85393	0.85124	0.852585
40000	0.853382881	0.85393	0.85112	0.852525



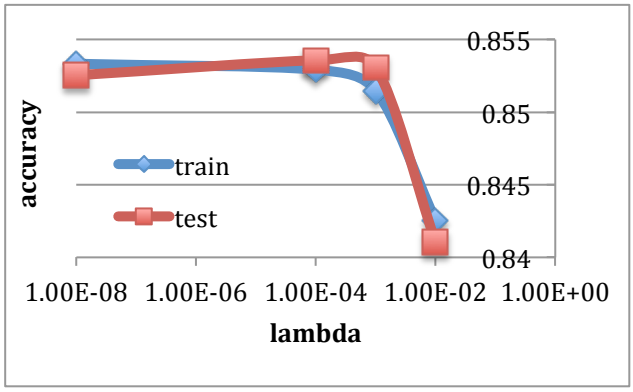
明顯可以觀察到做了 normalization 的模型，**正確率較高**（testing set accuracy(avg)沒有做 ~ 0.79，有做 ~ 0.85）
而且做了 normalization **收斂的也比較快**，在 iteration = 30000 處，沒做 normalization 的模型尚未收斂，做了 normalization 的模型卻已經 overfitting 了。

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：

以 logistic regression 的模型（選 X_train 的一次項作為 feature, 做 normalization, iteration = 20000），討論加入 regularization 的影響：

lambda	train	public	private	test (avg)
0	0.853321458	0.85356	0.85149	0.852525
0.0001	0.852922208	0.85565	0.85149	0.85357
0.001	0.851478763	0.85442	0.85173	0.853075
0.01	0.842541691	0.84324	0.83884	0.84104



可以發現當 lambda = 0.0001, 0.001 時，testing set 上的 performance 都比沒有 regularization 時好。可見加入了 regularization 可以修正 overfitting。

5.請討論你認為哪個 **attribute** 對結果影響最大？

答：

檢驗所有 features 與 Y_train 的相關程度(correlation)，發現 **0/age** (0.2340), **33/Married_civ_spouse**(0.4447), **35/Never_married**(-0.3184), **53/Husband**(0.4010)的相關程度均非常大。

若實驗抽掉其中一個 feature，會發現少了 **0/age**， **3/capital_gain**， **4/capital_loss**， **5/hours_per_week** 的 correctness 會大幅減少。(設定 iteration=100)

總地來說，我認為 **0/age** 這個 feature 對結果影響最大，所以我有再做增加 **0/age** 高次項的模型。