

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

答:

Generative model:

hw2_pgm.csv	0.84227	0.84533
2 hours ago by r05922028_yao		
add submission details		

Logistic regression:

hw2_logistic.csv	0.84940	0.85479
26 minutes ago by r05922028_yao		
add submission details		

根據 kaggle 上面的成績，可以看出不管 private data 或 public data 都是 logistic regression 的成績較高。

2.請說明你實作的 best model，其訓練方式和準確率為何？

答:

hw2_best.csv	0.85370	0.85859
23 minutes ago by r05922028_yao		
add submission details		

我的 best model 是利用所有 features 做 normalization 再加上二次式的 training 過程，最後準確率在 private data 為 85.37% 和 public data 為 85.86

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:

我在 logistic model 實作有 normalization 和沒有 normalization 的比較，結果如下圖：

沒有 normalization:

hw2_gd_adagrad_no_norm.csv	0.78184	0.78501
just now by r05922028_yao		
add submission details		

實作 normalization:

hw2_gd_adagrad_norm.csv	0.84940	0.85479
4 hours ago by r05922028_yao		
add submission details		

當我在相同條件下，在我的 logistic model 加上 normalization 之後的結果會比用原本的資料下去 train 的結果還要好。沒有實作 normalization 的 model 在 private data 的表現只有 78%，實作 normalization 之後準確度可以達到 85%左右。

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:

我在 logistic model 實作有 regularization 和沒有 regularization 的比較, 結果如下圖:

沒有 regularization:

hw2_gd_adagrad_norm_quad.csv a day ago by r05922028_yao	0.85370	0.85859
hw2_adagrad_quad_iter_400		

實作 regularization lambda = 0.1:

hw2_adagrad_norm_quad_regu.csv 4 hours ago by r05922028_yao	0.85308	0.85737
add submission details		

實作 regularization lambda = 100:

hw2_adagrad_norm_quad_regu_100.csv a few seconds ago by r05922028_yao	0.85321	0.85675
add submission details		

我認為在這個作業的 logistic model 加上 regularization 效果不大, 在準確度的結果下反而比較差。

5.請討論你認為哪個 attribute 對結果影響最大?

在 **Logistic model weights** 可以看出第四個 attribute 的比重較高, 因此推論'**capital_gain**' 對結果影響最大

3.22025735e-01	7.14914757e-02	3.39833402e-01	1.89990949e+00
2.47275569e-01	3.41025141e-01	1.02039055e-01	-1.20079748e-02
-2.25131218e-02	5.93950852e-02	5.92842018e-02	-8.94165917e-02
-3.32951719e-02	-6.65489683e-02	-7.54323490e-02	-2.03986546e-01
-2.05394574e-01	-8.46225928e-02	-1.18253758e-01	-1.43635175e-01
-2.32979228e-01	-1.75902890e-01	3.47829579e-04	3.12653869e-03
2.12314203e-01	1.81144749e-01	-2.32619711e-01	2.08341875e-01
-2.66406561e-01	1.88350905e-01	-6.88569646e-02	-1.48995025e-01
2.83219107e-02	3.05686979e-01	-4.73749382e-02	-3.87325072e-01
-9.29184369e-02	-5.44305816e-02	-4.71353453e-02	-1.99422210e-02
-3.22733001e-02	1.99645115e-01	-1.86723381e-01	-1.54086100e-01
-1.02974674e-01	-2.53156112e-01	-1.34433321e-01	1.11906928e-01
5.42591589e-02	3.68724616e-02	7.56647850e-02	-5.93431353e-02
-7.50035467e-02	1.61565753e-01	-1.03775516e-01	-9.75864342e-02
-3.16599111e-01	-1.04094294e-01	3.23748400e-01	-4.20073754e-02
2.74435350e-02	-2.51556441e-02	-2.52562049e-02	2.67856935e-02
3.10020532e-02	2.18568292e-02	-2.78820433e-02	-7.60761706e-02
1.83947452e-02	-6.52243541e-02	-5.53800672e-03	-2.40193720e-02
1.87855857e-02	1.89321493e-02	2.89966729e-02	-2.48937606e-02
-3.71703306e-03	-3.82880968e-04	-8.06394039e-03	-1.63919306e-02
-1.67615208e-03	-9.28497868e-04	-1.51088121e-02	2.20346618e-03
1.56397867e-02	3.73086170e-02	3.89114322e-03	1.98054150e-02
-1.28842358e-02	-5.96626679e-02	-2.31402059e-02	-5.62109638e-02
-2.21424370e-02	3.50711779e-02	1.56461878e-03	2.37220253e-05
-1.54315064e-02	9.30137891e-04	-4.71512855e-02	2.08314094e-03
-1.21086891e-02	-8.15811534e-03	7.10935378e-02	-4.45806121e-02
1.48401131e-02	-1.83935716e-02]		