學號：R06942128 系級： 電信碩一 姓名：許祐銘

1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize.

Latend_dim = 256	Private loss	Public loss
Normalize_rating	0.86914	0.86928
Non_normalize	0.86103	0.85843

Normalize 的方法是取 Training data Rating 的平均以及標準差，將每個 Rating 扣掉平均除以標準差當做新的 Rating，最後 Testing 的時候再將 Predict_Rating 乘上 Training data Rating 的標準差再加平均。結果兩者差別並沒有很大，大約是 0.0085 loss。

2. (1%)比較不同的latent dimension的結果。

Latent_dim	Private loss	Public loss
16	0.85982	0.85923
32	0.85320	0.85397
64	0.85517	0.85323
128	0.85726	0.85510
256	0.86103	0.85843
512	0.86185	0.86033
1024	0.86214	0.86161

可以很明顯發現， Latent_dim = 32 的時候 Loss 最低，增加或減少 dimension 都會導致 Loss 上升。

3. (1%)比較有無bias的結果。

Latend_dim = 32	Private loss	Public loss
No_bias	0.85227	0.85189
With_ bias	0.85320	0.85397

可以發現沒有加上 Bias 的結果稍好一些些，Loss 大約是減少了 0.001 ~ 0.002 。

4. (1%)請試著用DNN來解決這個問題，並且說明實做的方法(方法不限)。並比較MF和NN的結果，討論結果的差異。

```
Train users: 6040
Train movies: 3952
Latent dim: 32
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 1)            0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 1, 32)        193280      input_1[0][0]
__________________________________________________________________________________________________
embedding_2 (Embedding)         (None, 1, 32)        126464      input_2[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 32)           0           embedding_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 32)           0           embedding_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 32)           0           flatten_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 32)           0           flatten_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 64)           0           dropout_1[0][0]
                                                                 dropout_2[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 128)          8320        concatenate_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 64)           8256        dense_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1)            65          dense_2[0][0]
==================================================================================================
Total params: 336,385
Trainable params: 336,385
Non-trainable params: 0
```

Model	Private loss	Public loss
MF	0.85320	0.85397
DNN (128,64)	0.87769	0.87693
DNN (256,128,64,32)	0.88994	0.88856

有試著多疊了兩層 hidden layer，但是 Loss 並沒有變好，可能還要詳細調參數才有辦法達到比MF更好的訓練。

5. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。

(BONUS)(1%)試著使用除了rating以外的feature, 並說明你的作法和結果，結果好壞不會影響評分。