學號：R06922117 系級： 資工碩一 姓名：李岳庭

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

   我使用 kaggle 上最好的 MF model，加上 normalize 的結果如下：

normalized	public	private	total
yes	0.87710	0.87511	0.87611
no	0.87491	0.87275	0.87383

   比較之下，有做 normalize 的結果反而較差。

   normalize 的方法是算出 training data 的 rating 平均值和標準差，將所有 rating 值以以下公式更新，train 出一個 model。

$$x' = \frac{x - \bar{x}}{\sigma}, \quad \bar{x} : mean \ ; \ \sigma : standard\ deviation$$

   test 時先將 model predict 好的值輸出，再以 raining data 的 rating 平均值(mean_val)和標準差(std_val)還原，得出正確的結果。

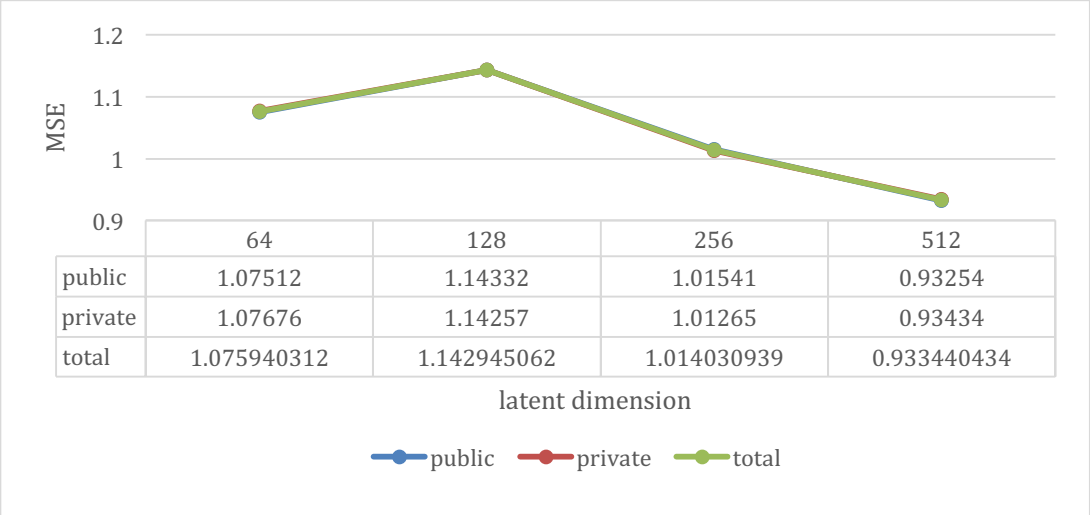
   mean_val = 3.58171208604

   std_val = 1.11689766115

$$pred_{test} = pred^{*}_{test} \times train_{std} + train_{mean}$$

2. **(1%)比較不同的 latent dimension 的結果。**

   使用 MF 方法架構與助教相同，將 latent dimension 設為 64、128、256、512，train 好 model 下去預測的結果如下：



	64	128	256	512
public	1.07512	1.14332	1.01541	0.93254
private	1.07676	1.14257	1.01265	0.93434
total	1.075940312	1.142945062	1.014030939	0.933440434

   經過測試的結果，latent dimension 有越高越好的趨勢，其中 128 達到最大的 MSE，512 有最小的 MSE。

3. **(1%)比較有無 bias 的結果。**

   我使用上一題 latent dimension 為 512、不加 bias 的 model，得到的結果如下：

bias	public	private	total
yes	0.93254	0.93434	0.93344
no	0.93574	0.93716	0.93645

   可看出沒加 bias 的 MSE 略高一點點，這點也很 make sence，雖然差異不大就是了。

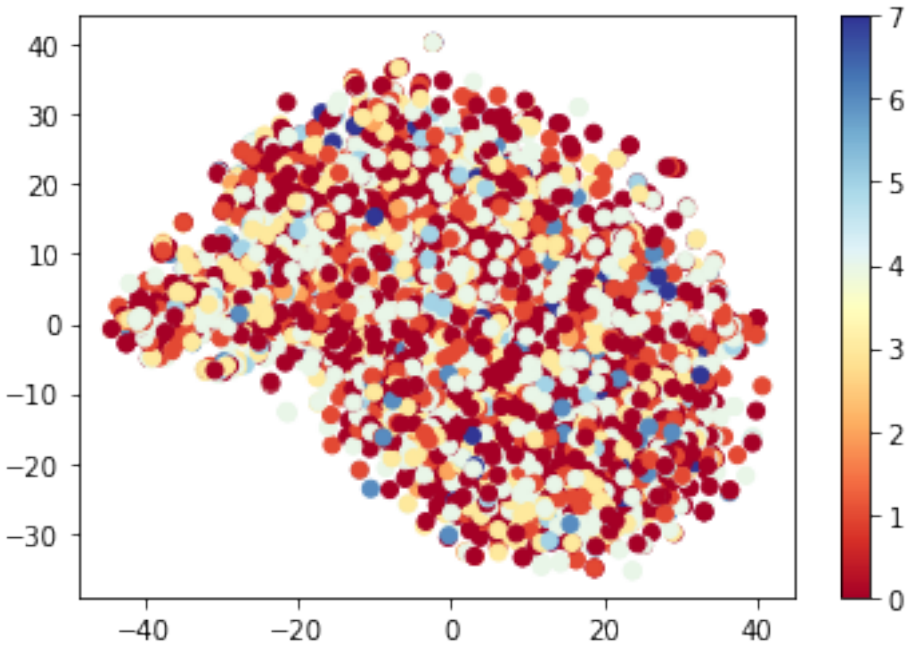
4. **(1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。**

   我使用 DNN 的架構如下，embedding 的 latent dimension 設為 689，只接兩層 Dense，一層是 128 個神經元，最後一層是一個輸出。

```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_1 (InputLayer)         (None, 1)                 0
__________________________________________________________________________________________________
input_2 (InputLayer)         (None, 1)                 0
__________________________________________________________________________________________________
embedding_1 (Embedding)      (None, 1, 689)            4161560     input_1[0][0]
__________________________________________________________________________________________________
embedding_2 (Embedding)      (None, 1, 689)            2722928     input_2[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)          (None, 689)               0           embedding_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)          (None, 689)               0           embedding_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)  (None, 1378)              0           flatten_1[0][0]
                                                                   flatten_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)          (None, 1378)              0           concatenate_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)              (None, 128)               176512      dropout_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0           dense_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)              (None, 1)                 129         dropout_2[0][0]
==================================================================================================
Total params: 7,061,129
Trainable params: 7,061,129
Non-trainable params: 0
__________________________________________________________________________________________________
```

   在 train 的過程中 DNN 收斂得較慢，但 validation 也有漸漸收斂，MF 則很容易 overfitting，故使用 DNN 的效果較佳。另外有發現 optmizer 也有差別，我原先使用'Adam'來 train，結果 MSE 一直比 strong baseline 高一點，後來改成'adamax'，才突破 strong baseline 來到 0.85672。

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



6. **(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。**

   由於 movies 的 category 難以量化，我只將每個 user 的 features 加入，concatenate 在每個 user 後，下去 train 完預測的結果並沒有顯著提升，MSE 為 0.85879。