學號：R06946009 系級： 資料科學碩一 姓名：林庭宇

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

每次實驗的 epoch 值都是 EarlyStopping 前，val_loss 最小的那次為準，並取其 epoch 值來重新訓練
Normalize 的方法：rating 減掉平均，除以標準差

	無 normalize	有 normalize
loss	0.5152	0.3624
val_loss	0.7317	0.5891
kaggle score	0.85414	0.85674

做完 normalize 後整體 loss 都下降，但是 kaggle 的結果並沒有比較好

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

每次實驗的 epoch 值都是 EarlyStopping 前，val_loss 最小的那次為準，並取其 epoch 值來重新訓練

latent dimension	64	128	256	512
loss	0.5771	0.5626	0.5152	0.5090
val_loss	0.7373	0.7341	0.7317	0.7340
kaggle score	0.85797	0.85526	0.85414	0.85462

latent dimension 越大時 loss 與 val_loss 會越小，在 latent dimension=256 時表現最好，之後再繼續增加 latent dimension 會使 val_loss 變差

3. (1%)比較有無 bias 的結果。
(collaborator:)

每次實驗的 epoch 值都是 EarlyStopping 前，val_loss 最小的那次為準，並取其 epoch 值來重新訓練

	無 bias	有 bias
loss	0.4867	0.5152
val_loss	0.7358	0.7317
kaggle score	0.85605	0.85414

有加 bias 時，val_loss 比較低，結果比較好

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。

DNN 參數設定：
Dropout=0.5，latent dimension=256
epochs=25，batch_size=512，validation_split=0.05

DNN 實作結構：

```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_1 (InputLayer)         (None, 1)                 0
__________________________________________________________________________________________________
input_2 (InputLayer)         (None, 1)                 0
__________________________________________________________________________________________________
embedding_1 (Embedding)      (None, 1, 256)            1546240     input_1[0][0]
__________________________________________________________________________________________________
embedding_2 (Embedding)      (None, 1, 256)            1011712     input_2[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)          (None, 256)               0           embedding_1[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)          (None, 256)               0           embedding_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)  (None, 512)               0           flatten_1[0][0]
                                                                   flatten_2[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)              (None, 256)               131328      concatenate_1[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0           dense_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)              (None, 1)                 257         dropout_1[0][0]
==================================================================================================
Total params: 2,689,537
Trainable params: 2,689,537
Non-trainable params: 0
__________________________________________________________________________________________________
Train on 854879 samples, validate on 44994 samples
```

	DNN	MF
loss	0.6190	0.5152
val_loss	0.7473	0.7317
kaggle score	0.86334	0.85414

實驗得到的 DNN 結果沒有比 MF 好，但是依然有辦法突破 Strong Baseline，有嘗試多加幾層網路，但是結果反而越來越差，其中 DNN 結果比較差的原因，個人猜測可能是 DNN 把兩個 embedding layers 給 Concatenate 在一起了，每個值都單獨丟進神經網路中訓練，不像 MF 有 User 與 Movie 兩個矩陣相乘的對應關係

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

將 18 類電影重新分成 7 類後作圖

分類如下：

0：Animation、Children's、Comedy

1：Action、Adventure

2：Romance

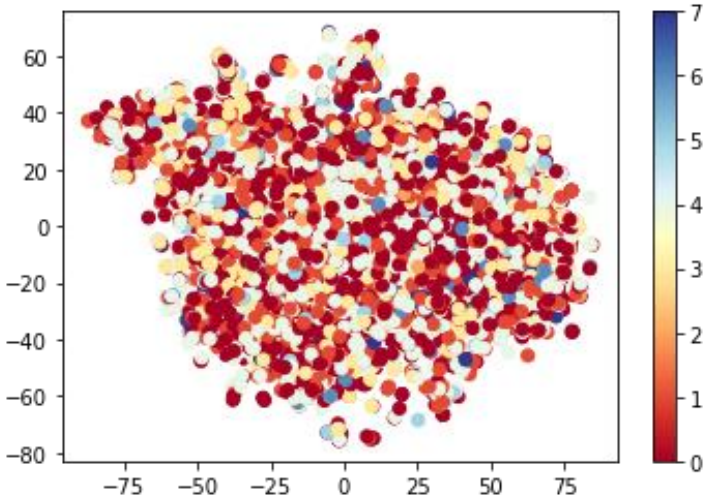
3：Crime、Film-Noir、Horror、Thriller

4：Drama、Musical

5：Documentary、War

6：Fantasy、Mystery、Sci-Fi

7：Western



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。

(collaborator:)

取 users.csv 中的 Gender、Age、Occupation 這三個比較有鑑別度的屬性當作特徵，各用一個輸出為 1 神經元的 Dense 接入，在 MF 加 Bias 時，一起把這三個特徵當作 Bias 加起來，最後得到 loss: 0.5645 - val_loss: 0.7915 的結果，但是 kaggle 分數只有 1.07127，結果並不理想