學號：R06922113 系級： 資工所碩一 姓名：陳宣伯

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

Normalization	RMSE
True	0.86870
False	0.87335

標準化流程：

a. 算出資料的標準差與平均值

b. 將每筆資料減去平均再除以標準差$\frac{x-\mu}{\sigma}$ μ：*平均值*; σ： 標準差

c. 將預測出來的值經過如下處理：

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

2. (1%)比較不同的 latent dimension 的結果。

Latent dimension	RMSE
64	0.87841
128	0.87602
256	0.86760
500	0.86637
666	0.86589

可以發現 latent dimension 要到達 666 才會超越 strong baseline

3. (1%)比較有無 bias 的結果。

Bias	RMSE
True	0.86589
False	0.89134

可以很明顯地看出，有 bias 的處理下可以得到較好的表現

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 的架構如下圖：

```
def nn_model(n_users, n_items, latent_dim=666):
    user_input = Input(shape=[1])
    item_input = Input(shape=[1])

    user_vec = Embedding(n_users, latent_dim,
                         embeddings_initializer='uniform')(user_input)
    user_vec = Flatten()(user_vec)

    item_vec = Embedding(n_items, latent_dim,
                         embeddings_initializer='uniform')(item_input)
    item_vec = Flatten()(item_vec)

    merge_vec = Concatenate()([user_vec, item_vec])
    hidden = Dense(256, activation='relu')(merge_vec)
    hidden = Dropout(0.3)(hidden)
    hidden = Dense(128, activation='relu')(hidden)
    hidden = Dropout(0.3)(hidden)
    output = Dense(1)(hidden)

    model = Model([user_input, item_input], output)
    # sgd = optimizers.SGD(lr=0.001, decay=1e-6, momentum=0.9, nesterov=True)
    model.compile(loss='mse', optimizer='adamax', metrics=[rmse])
    model.summary()
    return model
```

基本上我就是根據助教 TA Hour 提供的 Sample code 去改的，我把 Embedding 那邊的 initializer 改成了 uniform，並加了兩層的 Dropout 防止 overfitting

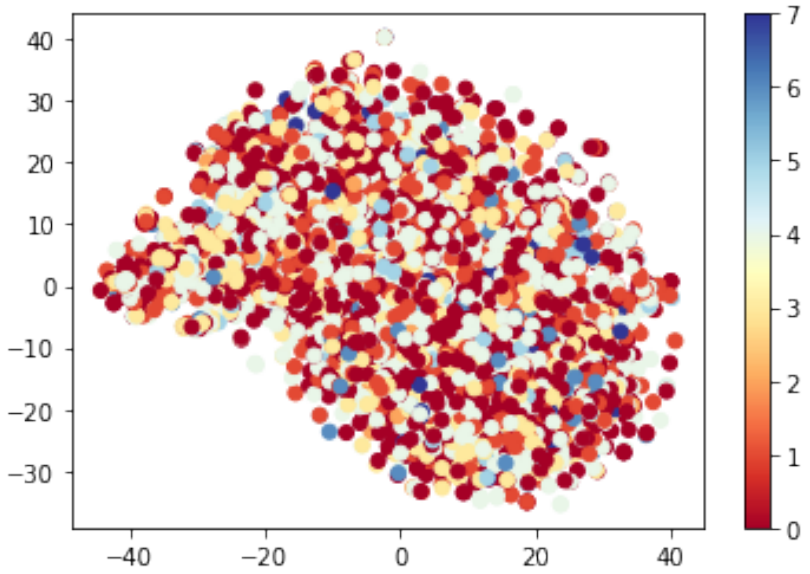
MF 與 NN 的比較：

我固定 latent dimension = 666 下，去比較兩個模型的結果如下：

方法	RMSE
MF	0.88892
NN	0.86589

單純用 MF 來算的話很受 latent dimension 的值影響，而且 MF 的演算機制比較難學習到使用者間的關係，這也是為什麼我覺得 NN 調好參數後一定能比 MF 好的關鍵

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。

這次助教有多給我們兩個檔案分別是 user.csv 跟 movie.csv，所以除了 rating 以外，每個 user 在 train 的時候我都會額外再加上 Gender 跟 Age，因為我認為同一個性別或同一個年齡層喜歡的電影會大致相同，例如說年輕男生喜歡動作爽片、中年婦女喜歡看愛情片等等，以下是我選擇用 NN 架構與 latent dimension = 6666 之結果比較，雖然結果沒有改善但是我認為繼續調參數應該會比原本只選用 rating 來當 features 來的好

Feature	RMSE
Rating	0.86589
Rating+Gender+Age	0.86389