

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

(collaborator:葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129、葉孟元 R04921094)

	kaggle Public Score	kaggle Private Score
有normalize	0.85032	0.85386
沒有normalize	0.85280	0.85466

normalize 的做法是，將每筆rate的分數扣掉rate的mean，  
發現沒有做normalize竟然比normalize還要好，突破我原先自己最高的kaggle分數

2. (1%)比較不同的latent dimension的結果。

(collaborator:葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129、葉孟元 R04921094)

(以下資料皆有做user and movie bias、normalize)

latent dimension	kaggle Public Score	kaggle Private Score
20	0.86154	0.86080
200	0.85404	0.85450
2000	0.86013	0.86034

可以由上表發現，dimension=200的效果最好，  
可能是dim = 20時是underfitting(因為train的時候loss也降不太下來)、dim = 2000的時候是overfitting

3. (1%)比較有無bias的結果。

(collaborator:葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129、葉孟元 R04921094)

	kaggle Public Score	kaggle Private Score
有bias	0.85032	0.85386
沒有bias	0.85404	0.85450

有考慮bias後明顯有變好，推測是加入bias後，就代表可以平移，某些 user/movie rating 特高或特低的狀況得以被考量。

#### 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。

(collaborator: 葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129、葉孟元 R04921094)

以下是我的DNN結構：

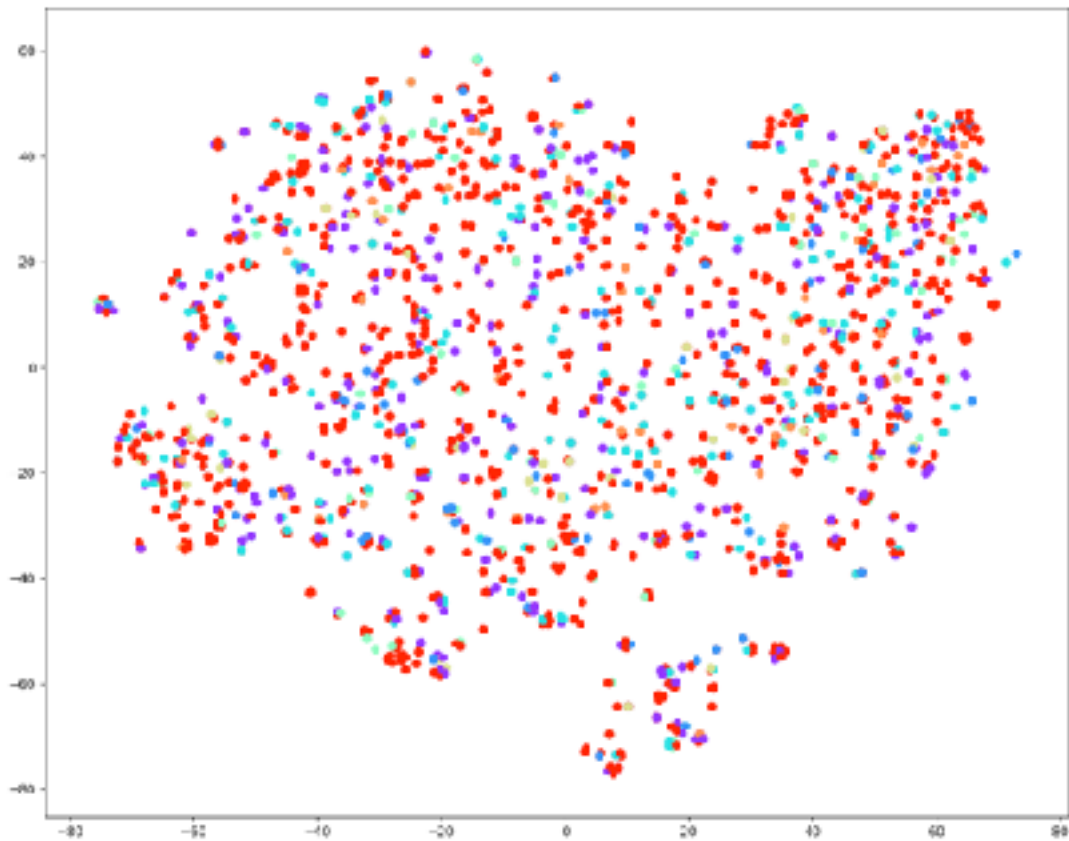
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 200)	1280200	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 200)	792500	input_2[0][0]
flatten_1 (Flatten)	(None, 200)	0	embedding_1[0][0]
flatten_4 (Flatten)	(None, 200)	0	embedding_2[0][0]
concatenate_1 (Concatenate)	(None, 400)	0	flatten_1[0][0] flatten_4[0][0]
dense_1 (Dense)	(None, 150)	60150	concatenate_1[0][0]
dense_2 (Dense)	(None, 50)	7550	dense_1[0][0]
dense_3 (Dense)	(None, 1)	51	dense_2[0][0]
Total params: 2,066,551			
Trainable params: 2,066,551			
Non-trainable params: 0			
Train on 809885 samples, validate on 89988 samples			

先將user 和movie用concatenate接起來，後面再架兩層DNN。

結果非常的差，在kaggle public 中的表現為0.89。

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

(collaborator:葉韋辰 R06922130、鄭克宣 R06921083、丁縉楷 R06922129、蔡孟庭 R05922078)



不知道自己有沒有做錯@@ 圖看起來非常的亂

(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。

沒有做QAQ