學號：R06922118 系級： 資工碩一 姓名：吳政軒

1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize.
(collaborator:R06944049 黃敬庭, R06944032 倪溥辰)

latent dimension 為 300,
有 normalize：0.86584
（將 rating 減掉平均再除以標準差, testing 時將結果乘以標準差再加上平均）
沒有 normalize：0.85907
可以發現對 rating normalize 是沒有比較好的

2. (1%)比較不同的 latent dimension 的結果。
(collaborator: R06944049 黃敬庭, R06944032 倪溥辰)

Latent dimension	200	250	300	350
Score	0.86000	0.85918	0.85907	0.86033

3. (1%)比較有無 bias 的結果。
(collaborator: R06944049 黃敬庭, R06944032 倪溥辰)

latent dimension 為 250,
沒有 bias：0.85978
有 bias：0.85918
可以發現加上 bias 有些微的進步

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: R06944049 黃敬庭, R06944032 倪溥辰)

因為時間來不及，所以這題我使用最簡單的 DNN，架構如下：

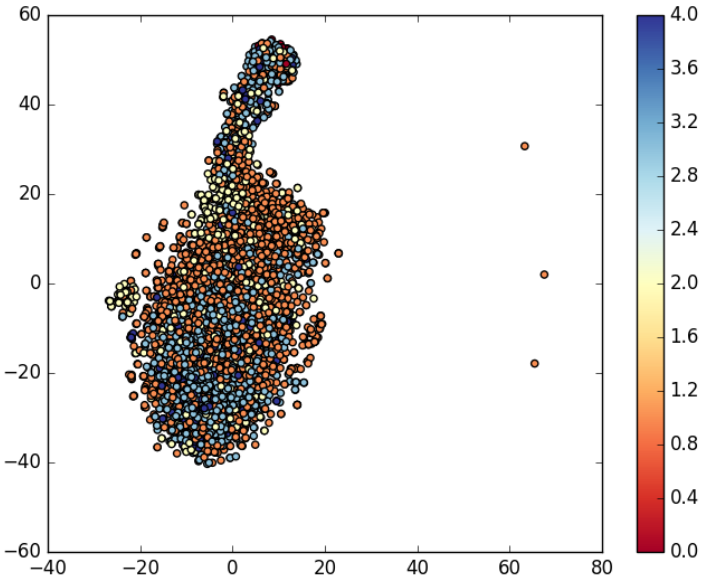
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 300)	1812000	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 300)	1185600	input_2[0][0]
flatten_1 (Flatten)	(None, 300)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 300)	0	embedding_2[0][0]
concatenate_1 (Concatenate)	(None, 600)	0	flatten_1[0][0] flatten_2[0][0]
dense_1 (Dense)	(None, 128)	76928	concatenate_1[0][0]
dense_2 (Dense)	(None, 128)	16512	dense_1[0][0]
dense_3 (Dense)	(None, 1)	129	dense_2[0][0]

Total params: 3,091,169
Trainable params: 3,091,169
Non-trainable params: 0

使用 MF 的分數為 0.85918
使用 DNN 的分數為 0.87699
在我的實驗中 MF 是比較好的，不過或許 DNN 架構和參數調整一下可能會贏過 MF

5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。
(collaborator: R06944049 黃敬庭，R06944032 倪溥辰)

我總共分成五類：
Comedy, Animation, Children's, Action, Adventure 一類
Thriller, War, Horror, Crime 一類
Musical, Drama, Romance, Fantasy, Sci-Fi 一類
Documentary, Film-Noir, Mystery, Western 一類
還有一些 ID 是沒有資料的為一類
最後畫出的圖如下：



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: R06944049 黃敬庭，R06944032 倪溥辰)

我額外多使用了 movie category，一樣 embedding 後和 user 及 movie concatenate，其餘架構都沒有動，架構如下：

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
input_3 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 300)	1812000	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 300)	1185600	input_2[0][0]
embedding_3 (Embedding)	(None, 1, 300)	1185600	input_3[0][0]
flatten_1 (Flatten)	(None, 300)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 300)	0	embedding_2[0][0]
flatten_3 (Flatten)	(None, 300)	0	embedding_3[0][0]
concatenate_1 (Concatenate)	(None, 900)	0	flatten_1[0][0] flatten_2[0][0] flatten_3[0][0]
dense_1 (Dense)	(None, 128)	115328	concatenate_1[0][0]
dense_2 (Dense)	(None, 128)	16512	dense_1[0][0]
dense_3 (Dense)	(None, 1)	129	dense_2[0][0]

Total params: 4,315,169
Trainable params: 4,315,169
Non-trainable params: 0

最後得到的分數由 0.87699 進步到 0.86780。