

1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize.
(collaborator:r06922130葉韋辰、r06944034黃禹程)

	public score	private score
無normalize	0.84633	0.85107
有normalize	0.85055	0.85016

Normalize之後在private的成績比較好，normalize的方法為助教在投影片所提供減平均除標準差

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

2. (1%)比較不同的latent dimension的結果。
(collaborator:r06922130葉韋辰、r06944034黃禹程)

	public score	private score
Dim = 960	0.85522	0.85732
Dim = 480	0.84780	0.85051
Dim = 240	0.84633	0.85107
Dim = 120	0.84804	0.85020
Dim = 60	0.85027	0.85137

可以看到dimension太高或太低表現都沒有比較好。以private的分數來說dimension=120的時候表現最好。

3. (1%)比較有無bias的結果。
(collaborator:r06922130葉韋辰、r06944034黃禹程)

	public score	private score
無bias	0.84607	0.85061
有bias	0.84633	0.85107

沒有bias的表現比較好，照理說應該有bias表現比較好。

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。

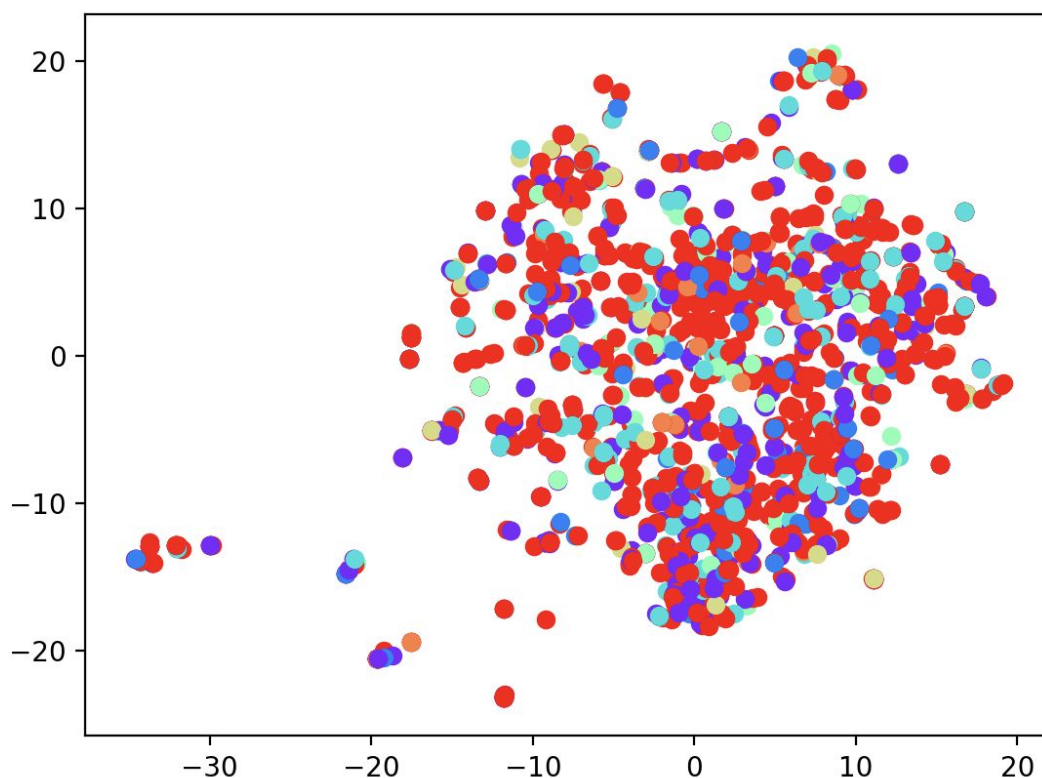
(collaborator:r06922130葉韋辰、r06944034黃禹程)

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 240)	1449840	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 240)	948720	input_2[0][0]
flatten_1 (Flatten)	(None, 240)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 240)	0	embedding_2[0][0]
concatenate_1 (Concatenate)	(None, 480)	0	flatten_1[0][0] flatten_2[0][0]
dense_1 (Dense)	(None, 150)	72150	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: 2,478,311			
Trainable params: 2,478,311			
Non-trainable params: 0			

	public score	private score
MF	0.84633	0.85107
NN	0.88141	0.88069

實作方法為助教投影片兩層dense，雖然助教說dnn可以比MF還好，不過MF比較容易得到好的performance

5. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖。
(collaborator:r06922130葉韋辰、r06944034黃禹程)



先手動將電影類別降至7種，可以看到做tsne時全部都混在一起，效果不是很好

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:)