

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

	有 Normalize	無 Normalize
RMSE	0.86091	0.89168

Normalize: `train_Y_Nor = (train_Y - train_Y.mean()) / np.std(train_Y)`

並記錄 `mean`、`std`，`predict`時:`result = predict*std + mean`

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

latent dimension	RMSE
6666	0.86091
16	0.87530
32	0.86830
64	0.86415
128	0.86311

`latent dimension` 越大 `RMSE` 越小，不過像 `dimension` 設到 6666 時，每個 `epoch training` 時間會變很長(約 90~100s)，`dimension` 到 128 為止每個 `epoch` 基本上都可以在 10 秒內跑完

3. (1%)比較有無 `bias` 的結果。
(collaborator:)

	有 bias	無 bias
RMSE	0.86311	0.91252

加上 `bias` 的結果明顯好很多，因為每個 `user`、`movie` 都會有自己的趨勢

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:)

	MF	DNN
RMSE	0.86311	0.88243

參考 `sample code` 的架構，兩種 `model` 的參數量接近，不過 `MF` 的結果較好，`DNN` 如果想要得到較好的結果需要特別調教參數，會比較麻煩，`MF` 原則上比較簡單

DNN:

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 128)	773248	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 128)	773248	input_2[0][0]
flatten_1 (Flatten)	(None, 128)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 128)	0	embedding_2[0][0]
dropout_1 (Dropout)	(None, 128)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 128)	0	flatten_2[0][0]
batch_normalization_1 (BatchNorm	(None, 128)	512	dropout_1[0][0]
batch_normalization_2 (BatchNorm	(None, 128)	512	dropout_2[0][0]
concatenate_1 (Concatenate)	(None, 256)	0	batch_normalization_1[0][0] batch_normalization_2[0][0]
dense_1 (Dense)	(None, 150)	38550	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: 1,593,671			
Trainable params: 1,593,159			
Non-trainable params: 512			

MF:

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding_1 (Embedding)	(None, 1, 128)	773248	input_1[0][0]
embedding_2 (Embedding)	(None, 1, 128)	773248	input_2[0][0]
flatten_1 (Flatten)	(None, 128)	0	embedding_1[0][0]
flatten_2 (Flatten)	(None, 128)	0	embedding_2[0][0]
dropout_1 (Dropout)	(None, 128)	0	flatten_1[0][0]
dropout_2 (Dropout)	(None, 128)	0	flatten_2[0][0]
batch_normalization_1 (BatchNorm	(None, 128)	512	dropout_1[0][0]
batch_normalization_2 (BatchNorm	(None, 128)	512	dropout_2[0][0]
dot_1 (Dot)	(None, 1)	0	batch_normalization_1[0][0] batch_normalization_2[0][0]
Total params: 1,547,520			
Trainable params: 1,547,008			
Non-trainable params: 512			

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

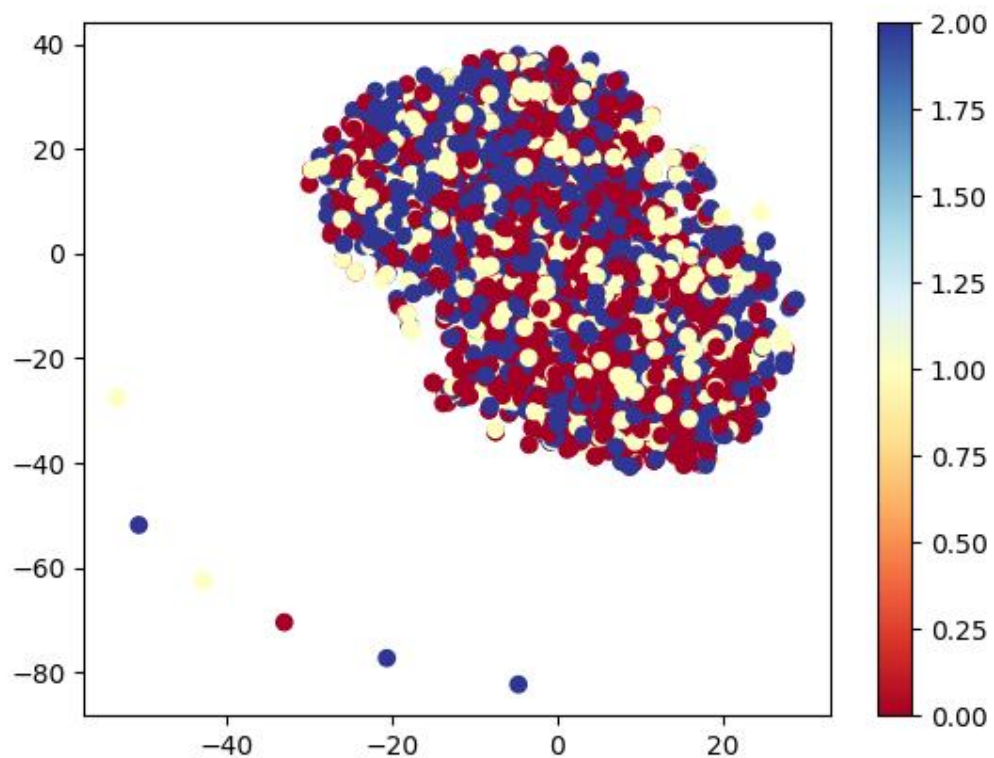
(collaborator:)

將 movie 分為三大類之後作圖結果

Drama:0

Comedy:1

Else:2



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:)