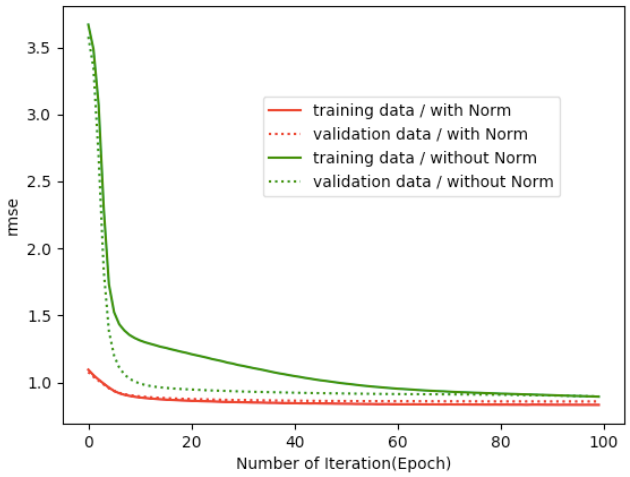
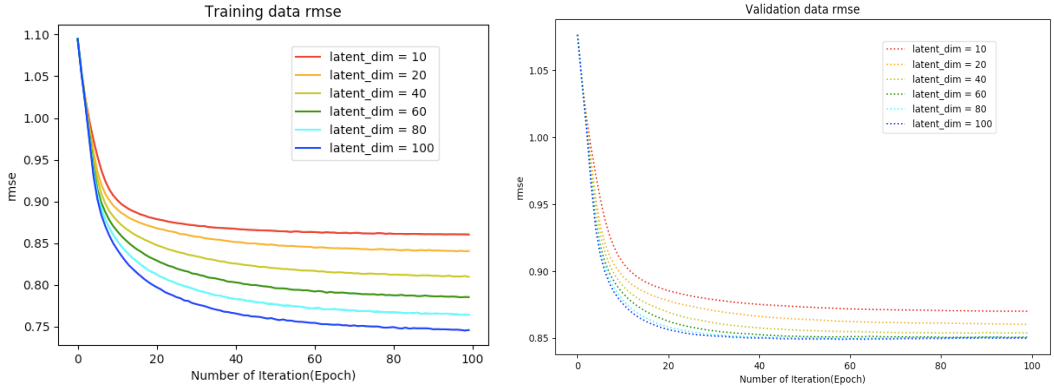
學號：R06922048 系級： 資工碩一 姓名：陳柏堯

1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize.
(collaborator:)
normalize 的方式是將每個 rating 的值減掉均值再除以標準差。Testing 的時候要乘上之前存好的標準差、加上之前的均值。
比較有無 normalize 的兩個 MF latent dim 我都設定為 25。
如下圖 training 過程的 rmse，發現 normalize 過後的 data training 較快，在 500 個 epoch 的時候，val rmse with norm 和 without norm 基本差不多，但是 norm 過後的 performance 略好一些。



epochs	rmse train	rmse val	rmse train (without norm)	rmse val (without norm)
50	0.840339383508	0.861209869438	0.99050607352	0.916963340651
100	0.83229477625	0.857391056108	0.892687377739	0.894344991899
500	0.828001326398	0.857171080478	0.828126540495	0.858051353753

2. (1%)比較不同的 latent dimension 的結果。
(collaborator:)
我分別比較 dim=10, 20, 40, 60, 80, 100 的 mf，他們的 data 都有 normalize 過。比較發現，dim 越高 training 越快收斂，經過 500 個 epoch 比較發現，dim 越高其 performance 會越好。但是，當 dim 大到 100 左右時，能提升的 performance 就有限了。



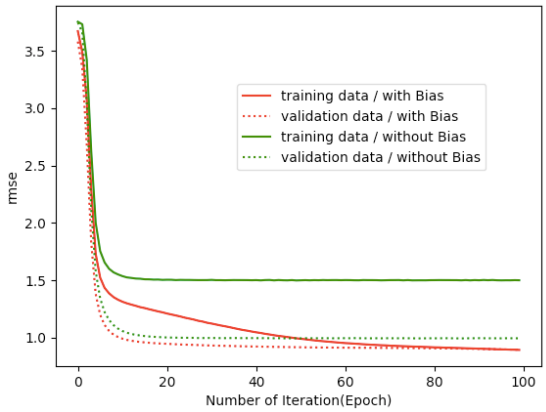
100epochs	rmse (training)	rmse(validation)
latent_dim=10	0.860438135	0.870045758
latent_dim=20	0.840536174	0.86041391
latent_dim=40	0.81026557	0.853824166
latent_dim=60	0.785594829	0.850756407
latent_dim=80	0.764185992	0.850519824
latent_dim=100	0.745714655	0.849947802

3. (1%)比較有無 bias 的結果。

   (collaborator:)

   有無 bias 比較的兩個 mf 我都沒有對 data normalize 過，無 bias 的 model 我是同時去掉 movie 和 user 的 bias。

   下圖 training 過程發現，有 bias 的 model 收斂比較快，從 validation rmse 來看，500 個 epoch 之後，有 bias 可以到 0.85 左右，而無 bias 只能到 0.9 多。這表示 bias 對於 performance 非常重要。



epochs	rmse train	rmse val	rmse train (without bias)	rmse val (without bias)
50	0.990506074	0.916963341	1.499865793	0.994658277
100	0.892687378	0.894344992	1.502169075	0.994767141
500	0.82812654	0.858051354	1.500741916	0.994261664

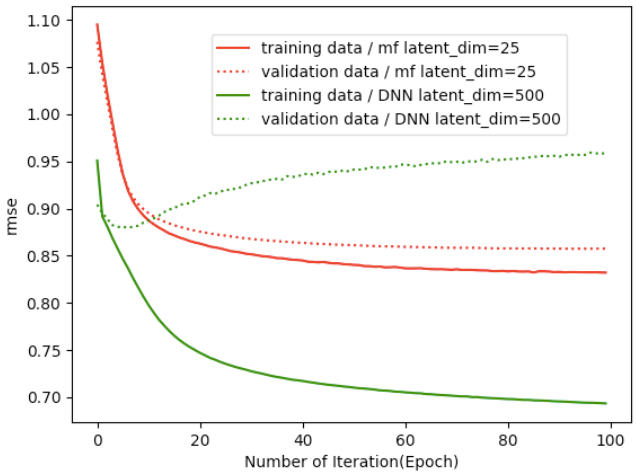
4. (1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。

   (collaborator:)

   MF 和 DNN 的比較，兩個 model 我都有做 normalize。

   DNN 的 model 是分別將 user 和 movie 做 500dim 的 embedding，然後 concatenate 在一起再依序接到 100、20、10、1 的 dense 做 linear regression。

   由於 DNN 的 latent dim 較大，因此收斂速度比 mf 快，其 validation 的 rmse 0.879 略比 mf 差一點。但這表示，DNN 可以做到接近 MF 的 performance，調整參數和 model 應可再改進 DNN 的 performance。



	Best val rmse	Best epochs
Mf latent_dim=25	0.856731646775	235
DNN latent_dim=500	0.879712448921	6

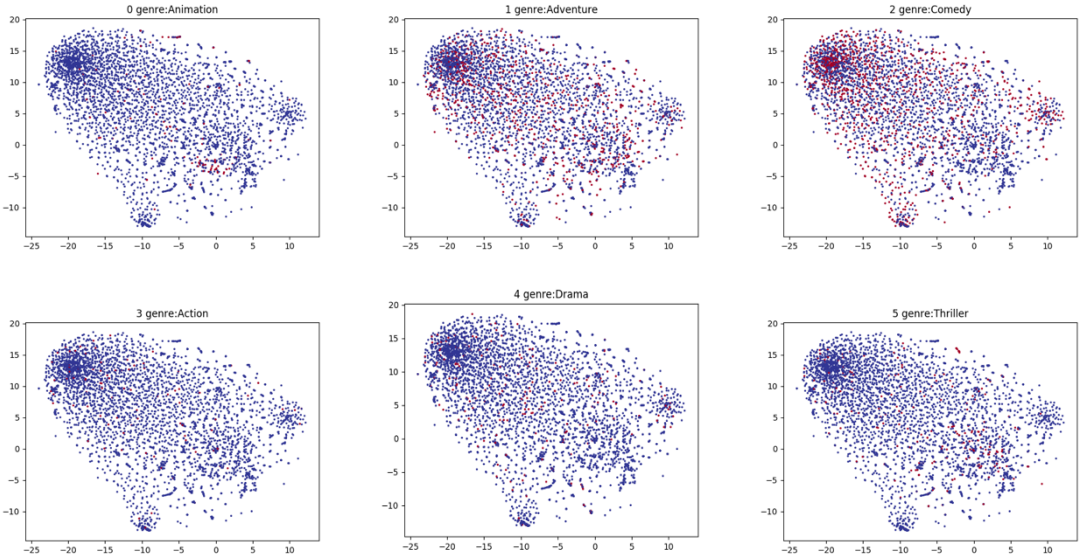
5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。

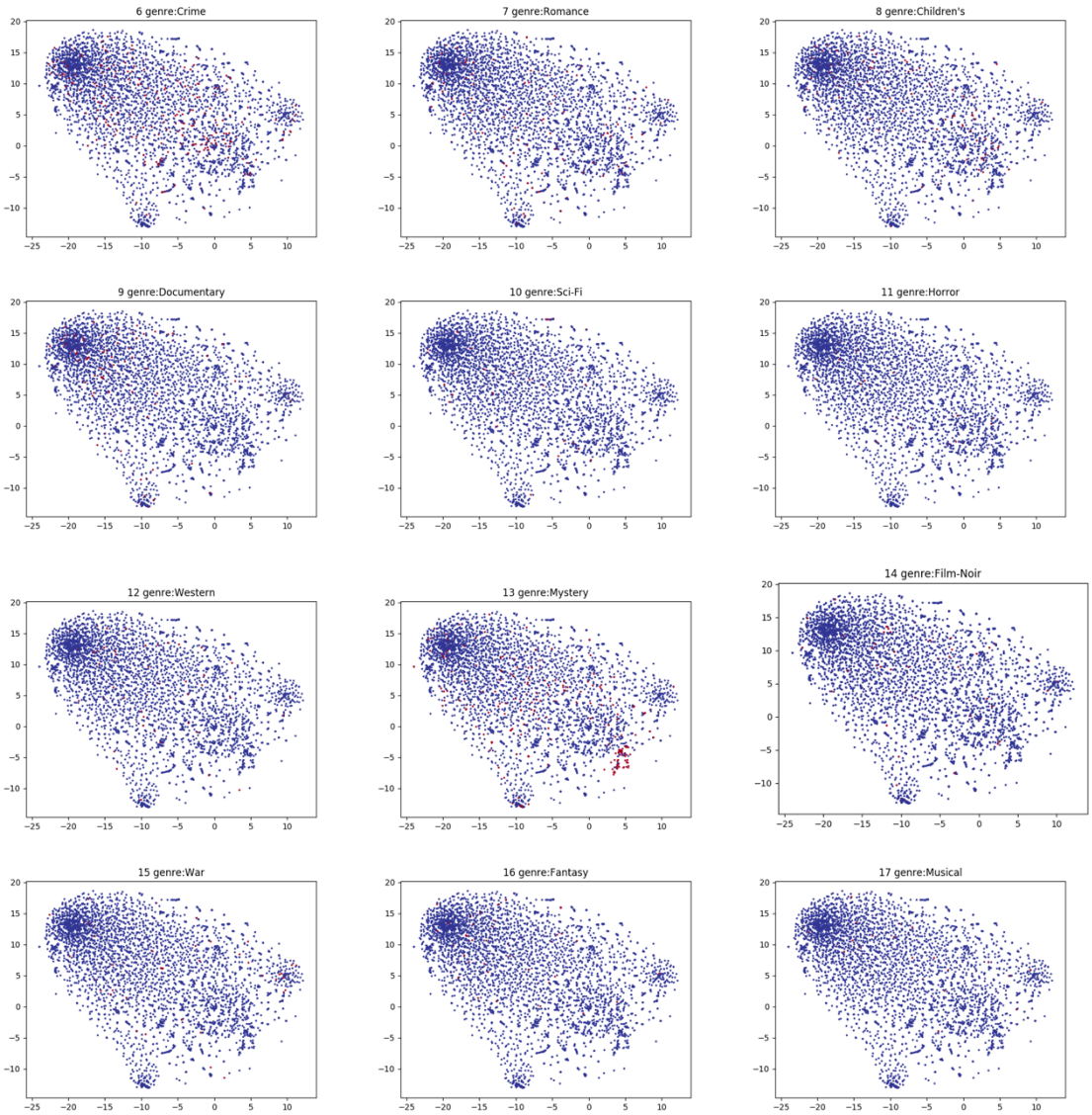
   (collaborator:)

   先取出 MF dim100 model 的 movie embedding。

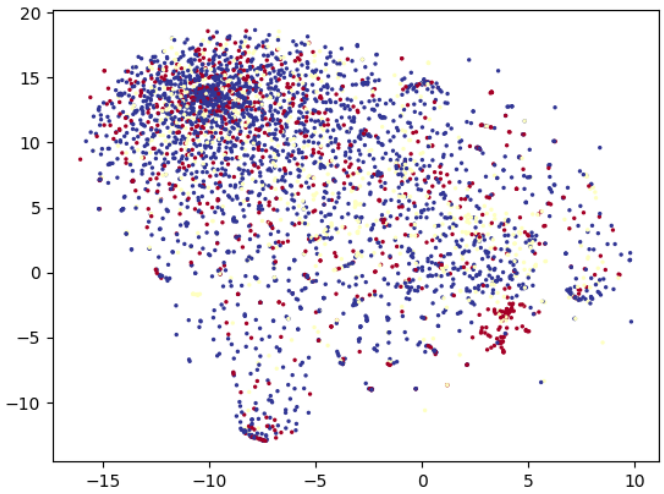
   對每個 movie 的 genre 做 random 取其中 1 個作為 category，再用 sklearn 的 T-SNE 對 embedding 層降維，對每一個 category 作圖如下：

   （紅色是該 genre，藍色是其他 genre）





再根據上方個別的分佈圖以及自己對於電影種類的認知，將一些不同的 genre 歸為同一個 category, 作圖如下：



紅色：Action, Drama, Documentary, Romance, Mystery, Sci-fi
黃色: Thriller, Crime, Horror, War, Film-Noir, Western
藍色： Musical, Animation, Children, Adventure, Fantasy, Comedy

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:)
作法：
latent_dim = 100 並且如第 1 題所述作 normalization。
原始 model 是 movie 和 user 的 embedding 層作 dot 加上 user bias 和 movie bias,
本題新的 model 是在原始 model 的基礎上，再加上 gender bias, age bias 和 occupation bias。Gender 和 occupation 是利用 embedding 算出 bias，age 則是接 1 層 dense 算出 bias.

結果：
沒有加上 gender, age, occupation 的 model 在一開始 training 的速度比較慢（如下圖綠線），而在 100 個 epoch 之後，其 performance 和原來的 model 基本差不多。原始的 model 在第 58 個 epoch，validation rmse 達到最低點 0.849183，上傳到 Kaggle 得到 public Score 0.85129；而加了新 feature 的 model 在第 160 個 epoch 達到 validation rmse 最低點 0.849136，上傳到 Kaggle 得到 public score 0.84979。 比較發現，增加這些 feature 對於最後的 performance 仍然有一點幫助。