

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

(collaborator: 陳代穎 r06922095)

normalization 的方式是對所有 rating 作 min-max scale。

加上 normalization 之後，training 的速度會比較快。

	Non-normalization (50 epochs)	Normalization (50 epochs)
Private RMSE	0.88499	0.87676
Public RMSE	0.89077	0.88422

不過 normalization 後繼續 training 下去的結果並不一定比較準確。

2. 比較不同的 *latent dimension* 的結果。

(collaborator: 陳代穎 r06922095)

	Latent dimension=3	Latent dimension=5	Latent dimension=10	Latent dimension=20
Private RMSE	0.87892	0.87393	0.88223	0.92024
Public RMSE	0.88440	0.88197	0.88971	0.92513

3. 比較有無 *bias* 的結果。

(collaborator: 陳代穎 r06922095)

	Bias	Non-bias
Private RMSE	0.87676	0.88408
Public RMSE	0.88422	0.89143

4. 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: 陳代穎 r06922095)

	NN	MF
Private RMSE	0.85349	0.86813
Public RMSE	0.86084	0.87729

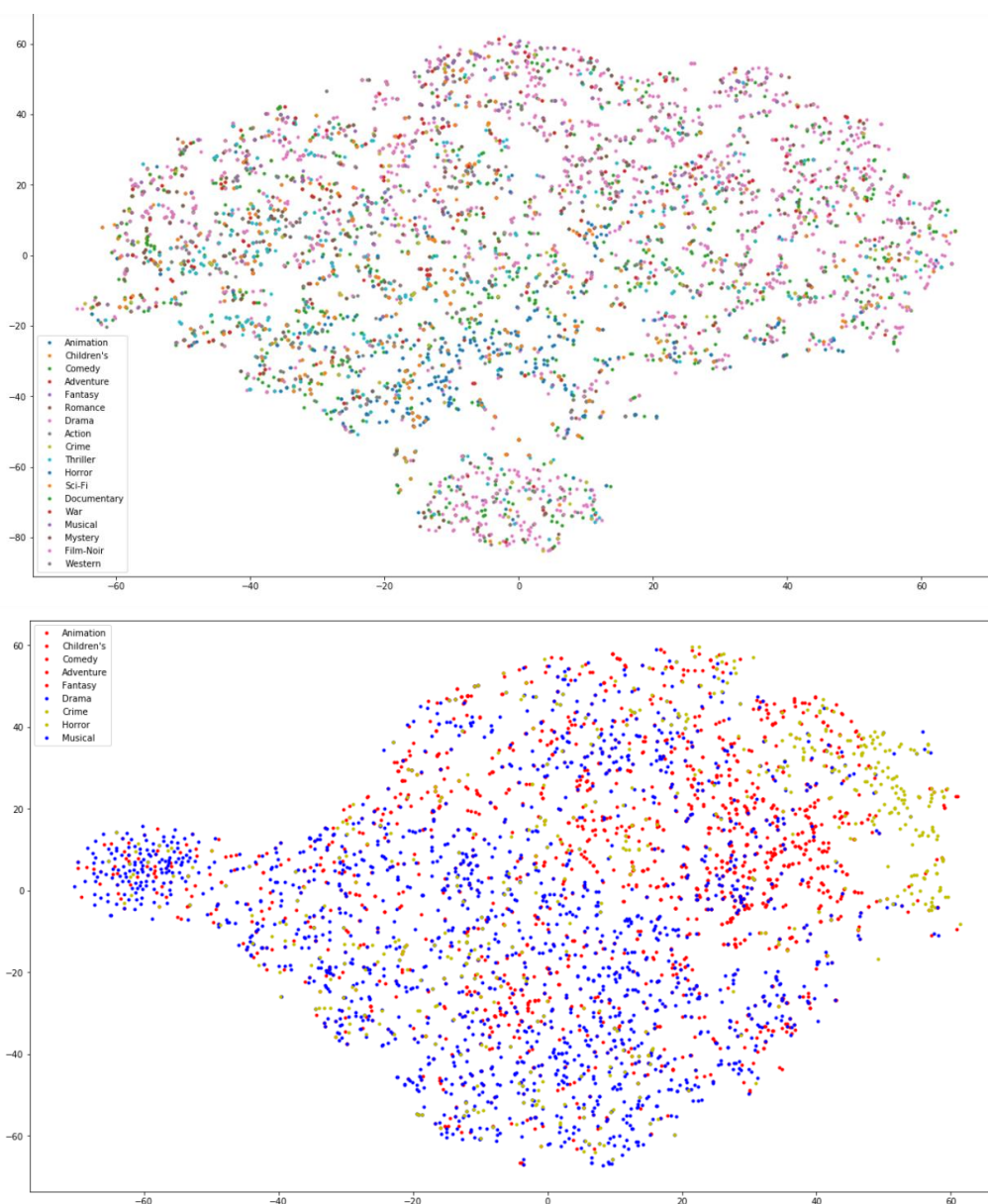
將 user_id, age, gender, occupation, movie_id 分別經過一層 embedding layer，再將所有 embedding vector concatenate 起來後丟入 3 個 Dense

Layer，其 units 分別為 512 256 128 。

兩者的結果比較後，可以發現 MF 所得到的 Performance 比較好，猜測是由於 MF Model 可以找出影響使用者偏好的因素(內積的運算)，而 DNN 只是單純地將 input 放入 embedding vector 後並接做 training。

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

(collaborator: 陳代穎 r06922095)



6. 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: 陳代穎 r06922095)

將 user_id, age, gender, occupation 各自經過一層 embedding layer 後，將得到 embedding vector 將加得到一個加總後的 embedding vector 。另外再將 movie_id 經過一層 embedding layer 後與上述加總後的 embedding vector 做內積。

將內積結果再加上 user bias 與 movie bias 後再接一個小型的 NN 便是整個 model structure 的設計。

Kaggle' s private / public score: 0.85506 / 0.86021

其表現比沒有考慮其他 feature 的 model 好。