

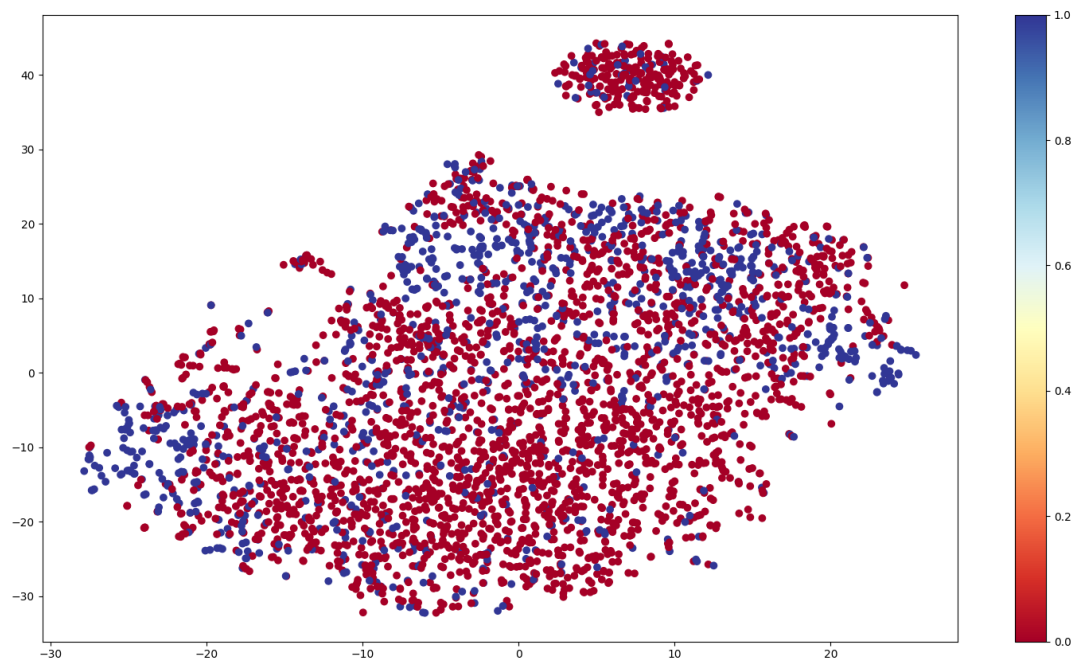
1. (1%)請比較有無 `normalize(rating)` 的差別。並說明如何 `normalize`。
將 `training data` 標準化後拿去學習，之後預測的結果再用 `training data` 原來的平均值跟標準差還原回來。 $(result = pred * training_std + training_mean)$
結果差異：
latent dimension = 25
沒標準化: 0.87448(Kaggle public score)
標準化: 0.90940(Kaggle public score)
標準化的效果看起來並沒有比較好。

2. (1%)比較不同的 latent dimension 的結果。

Latent dimension	Kaggle public score
10	0.87980
25	0.87448
50	0.87577
100	0.87749
200	0.88113
500	0.89091
1000	0.90260

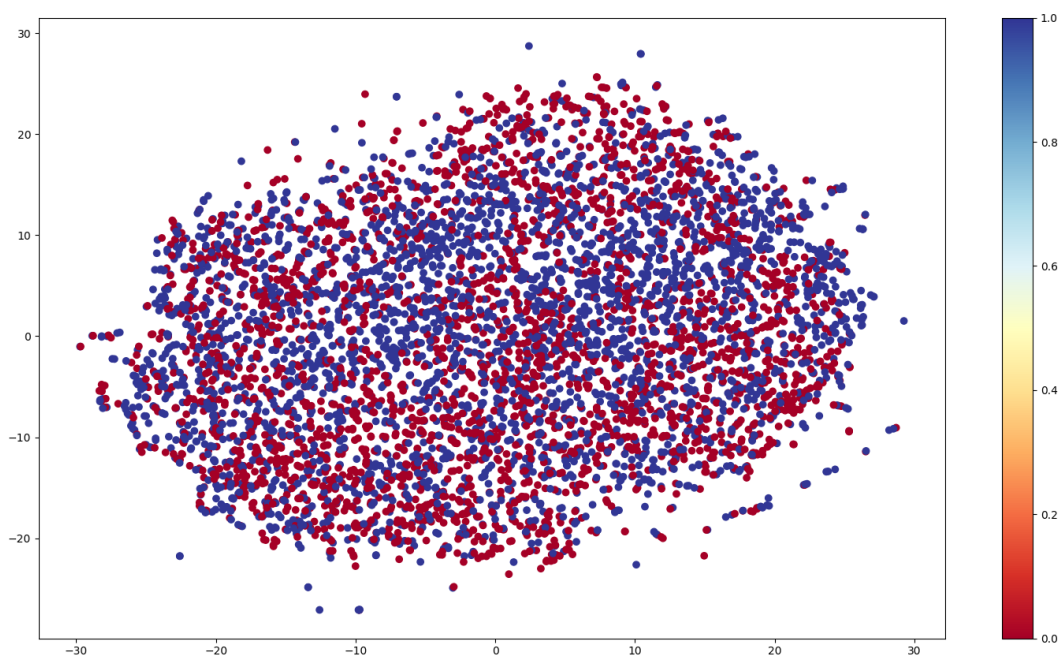
看起來 Latent dimension 在 25~100 之間會有不錯的結果。

3. (1%)比較有無 bias 的結果。
latent dimension = 25
有 bias: 0.87448(Kaggle public score)
沒有 bias: 0.88089(Kaggle public score)
看起來有 bias 的結果好一點。
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。
將 movie embedding 跟 user embedding 後 concatenate 在一起後再通過一個 DNN，output 則是將他視為 regression 問題，用 relu 選出一個數字作為預測的分數。
結果差異：
MF: 0.87448(Kaggle public score)
DNN: 0.86832(Kaggle public score)
DNN 的結果稍微比 MF 的結果好了一點。
5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。



Red = { Adventure, Animation, Children's, Comedy, Documentary, Drama, Film-Noir, Musical, Romance}(猜想: 著重劇情類的電影)
 Blue = {Action, Crime, Fantasy, Horror, Mystery, Sci-Fi, Thriller, War, Western}(猜想: 著重感官刺激類的電影)

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。



Users 資料裡的 age 有七個數字(1, 18, 25, 35, 45, 50, 56)，將他們當成 tag 做圖。
Red = {1,35,45,50,56}(猜想: 中年人(家長)與幼兒)
Blue = {18,25}(猜想: 年輕人)
Embedding 的結果沒有 movie category 好，但還是可以看出部份集中的區域。