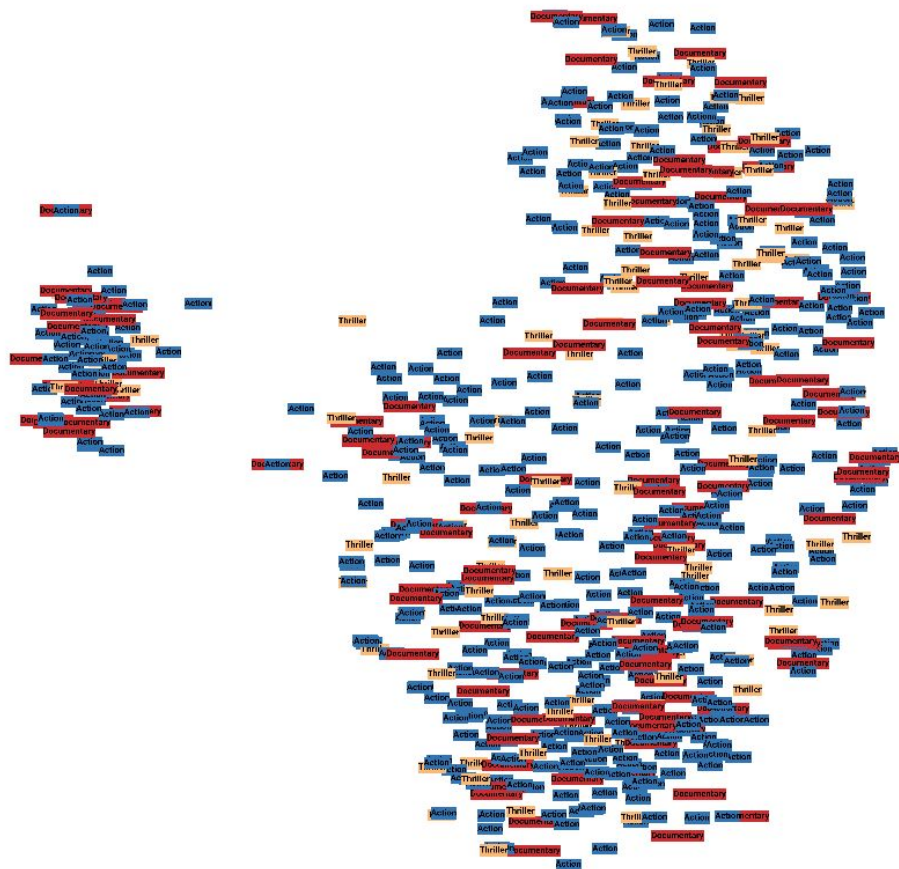


1. (1%)請比較有無normalize(rating)的差別。並說明如何normalize。
無normalize training rmse: 0.7783, kaggle rmse: 0.8722
有normalize training rmse: 0.7304, kaggle rmse: 0.9176 (把rating scale到[0, 1])
normalize後明顯看到overfitting的現象。
2. (1%)比較不同的latent dimension的結果。
dim=25 training rmse: 0.8360, kaggle rmse: 0.8941
dim=50 training rmse: 0.7783, kaggle rmse: 0.8722
dim=75 training rmse: 0.7402, kaggle rmse: 0.8791
dimension越大，training loss越小，但是testing上則不一定，要取一個適中的值，才有比較好的結果。
3. (1%)比較有無bias的結果。
無bias training rmse: 0.7783, kaggle rmse: 0.8722
有bias training rmse: 0.7329, kaggle rmse: 0.8772
加bias後training loss比較小，但是testing則幾乎沒有改變。
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。
把原本的mf中的embedding各自加深，從本來的2個矩陣相乘變成4個，相當於是各多了1層fully connected layer + relu，但是因為這樣參數過多導致嚴重的overfitting，training rmse可以到0.70，validation rmse卻是0.91多，因此沒有上傳kaggle。
5. (1%)請試著將movie的embedding用tsne降維後，將movie category當作label來作圖



用Tensorboard做tsne，藍色是Action 紅色是Documentary 米色是Thriller。

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。

承第4.題，如果把userid, movieid經過轉換後的embedding跟movie的屬性當作feature(binary feature)，加上users的(age, occupancy, gender) concat一起train，就可以達到kaggle上的成績0.8641