

以下的模型皆為同模型 只有改題目問的參數

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

(collaborator:)

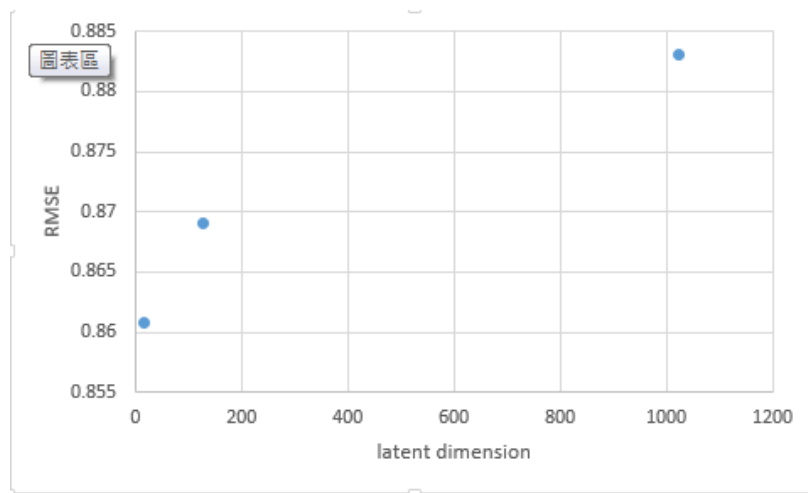
我是把 `Rating` 全部除以 5，限縮到 0~1 之間。

有 `normalize` 的模型上傳到 Kaggle 的 RMSE 為 0.92446

無 `normalize` 的模型上傳到 Kaggle 的 RMSE 為 0.86085

2. (1%)比較不同的 `latent dimension` 的結果。

(collaborator:)



我做了三種 `dimension` 上傳到 Kaggle 依序是 16,128,1024，RMSE 分別是 0.86085,0.86902,0.88302，發現 `dimension` 越小 RMSE 越小

3. (1%)比較有無 `bias` 的結果。

(collaborator:)

	RMSE
有bias	0.86085
無bias	0.91183

以上為上傳到 Kaggle 的結果，明顯有 `bias` 的模型 RMSE 會小很多

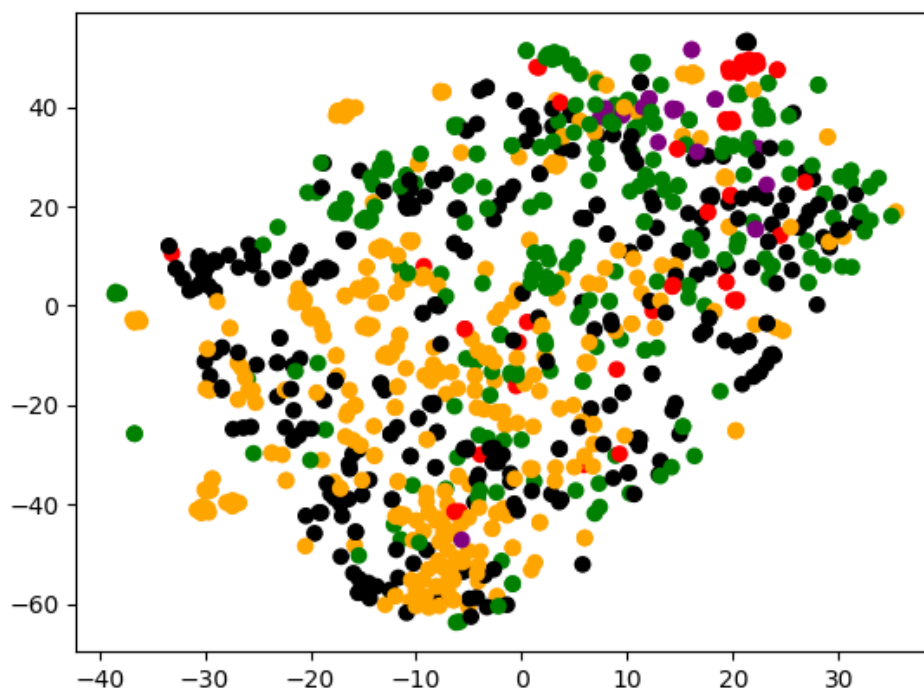
4. (1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。

(collaborator:)

將 `userID` 和 `movieID` 都 `embed` 後串起來做 DNN(疊五層，從 512 一直除二往下)，上傳到 Kaggle 的 RMSE 為 0.87902，`embed` 的地方都一樣，可看出用 MF 結果較好，且訓練速度較快

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

(collaborator: r05942148 鄭立晟)



上圖是將 moviedID 進行 embedding 後，利用 TSNE 降維來做圖，(由於不想太多顏色故只分成 5 個種類)(黑色:

['Animation','Comedy','Children's','Sci-Fi'], 橘色

['Adventure','Action','Romance','Fantasy'],綠色:

['Thriller','Horror','Drama','Mystery'],紅色: ['Crime','War','Musical'],紫色:

['Film-Noir','Documentary','Western'])

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:)

	RMSE
MF	0.86085
DNN	1.11931

左圖為 MF 和 DNN 上傳到 Kaggle 的比較

DNN 是把電影的標題和種類做成 word vector 總共 59 維，把使用者的性別、職業、年紀、職業、zip-code 做成一個 33 維的向量，並起來共 92 維。

MF 是只用 movieID 和 userID 做 embedding 成 16 維、兩者內積後加各自的 bias，和 rating 做 RMSE

明顯後者在處理資料上輕鬆很多，且 training 的速度快很多，Train 的結果更是比 DNN 好很多