學號：R06921095 系級： 電機碩一 姓名：吳睿哲

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

   (collaborator:)

   Normalize 的方式主要是取 training data rating 的 mean 以及 standard deviation,再把 training data rating 做 normalization,再用 normalize 過後的 rating 作為 training 的 target,在 keras 上實作很簡單,只要在 Dot 函式中將 normalize 設為 true 即可達到目的。

   至於 normalize 的結果比較:經過 normalize 之後的 model,loss 下降的速度比較快,比較不會遇到卡在 local minimum 的問題,最後也可以收斂到比較低的 loss。

   上傳到 kaggle 上的結果:no normalize:0.88573,normalized:0.86617

2. (1%)比較不同的 latent dimension 的結果。

   (collaborator:)

   在我的測試結果中,我分別取了 3 個不同的 latent dimension 做比較,分別是 222,666,1066 維,我發現如果取到一定的維度以上,結果就沒有什麼差別了,但如果取得太小,loss,會下不去,結果如下表格所示

   結果

	222 維	666 維	1066 維
Kaggle	1.04084	0.85236	0.85239

3. (1%)比較有無 bias 的結果。

   (collaborator:)

   沒有 bias 的 model,loss 下降的速度會非常慢,且一下就會卡在 local minimum,上傳到 kaggle 的結果也非常差,只有 0.92697

	No bias	Bias
Kaggle	0.85236	1.25653

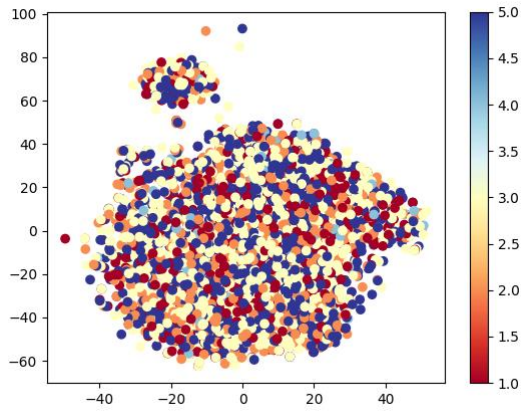
4. (1%)請試著將 movie 的 embedding 用 tsne 降維後, 將 movie category 當作 label 來作圖。

   (collaborator:)

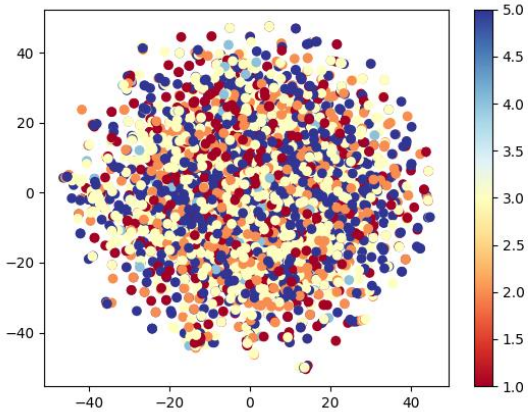
   這題我的作法是先將電影類型分為 5 大類

   Class1:[thriller,horror,crime,mystery]

Class2:[war,adventure,action,western]
Class3:[drama,musical,romance,doucumentary]
Class4:[sci-fi,fantasy,film-noir]
Class5:[children's,comedy,animation]
再從各電影中所佔 class 最多的當作那部電影的 class,如果一樣多的話就拿該電影的第一個 class 當作 class,然後再用 tsne 把 movie 的 embedding 直接降成 2 維結果如下圖所示



結果非常奇怪,我想說應該是跟作業 4 一樣,直接從太高維降到 2 維會分不好,於是我先用 pca 把 movie embedding 降到 50 維,再用 tsne 降到 2 維,結果如下圖所示



結果還是跟預期的差很多,我猜想可能的原因應該是把電影分成 5 大類太狹隘了,造成分得不理想,不然就是我分成 5 大類的方法不對,可能被我分在同一類的關聯性反而不高。

5. (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:)
我使用了 gender 當作我的 extra feature.我先用 user data 把 training data 中的男生以及女生分成 2 個 dataframe,再分別用這兩個 dataframe 用以上的方式(matrix factorization)去訓練出 2 個 model,再用這兩個 model 去 predict testing data,也就是

做 2 次的 matrix factorization,上傳到 kaggle 的結果是 0.86127,比原本的 0.85223 還要低,所以我覺得取 gender 當作 feature 可能不是很有用的方式。