

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

Ans:

將 training data 中 rating 的部分做 normalize。normalize 的方法為，先計算出 rating 的 mean 及 standard deviation，最後以 $\text{prediction} = \text{prediction} \times \text{std} + \text{mean}$ 來當作輸出結果。

	Public Score	Private Score
有 normalize	0.92216	0.91555
無 normalize	0.87076	0.86094

發現有 normalize 的結果反而比較差。

2. (1 %)比較不同的 embedding dimension 的結果。

Ans:

embedding dimension	Validation loss	Validation RMSE
16	0.7495	0.8638
32	0.7510	0.8647
64	0.7545	0.8666
128	0.7520	0.8654

取 training data 的 0.1% 當作 validation data。在 embedding dimension 為 16, 32, 64, 128 中，發現 validation RMSE 的結果是差不多的。

3. (1 %)比較有無 bias 的結果。

Ans:

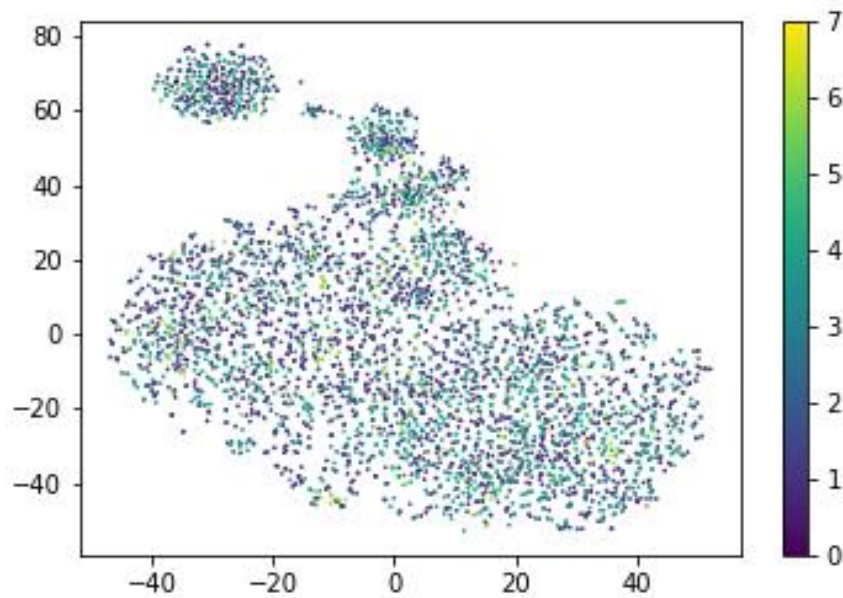
	Public Score	Private Score
有 bias	0.87076	0.86094
無 bias	0.87084	0.86031

有加 bias 的結果會比較好。

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

Ans:

我將 Comedy, Action, Romance, Adventure, Western 分一類，Thriller, Horror, Mystery 分一類，Crime, Film-Noir 分一類，Drama, Musical 分一類，Documentary, War 分一類，Animation, Children's 分一類，最後將 Sci-Fi, Fantasy 分一類。



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。

Ans:

我將 users.csv 中 age, occupation 加入我的 model。

使 $r_{i,j}$ 變為 $r_{i,j} = U_i \cdot V_j + A_i \cdot V_j + O_i \cdot V_j + b_i^{user} + b_i^{age} + b_i^{occupation} + b_j^{movie}$

U: user

A: age

O: occupation

V: movies

	Public Score	Private Score
使用額外 feature	0.87468	0.86784
無使用額外 feature	0.87076	0.86094

發現沒有使用額外 feature 的結果反而沒有比較好。

Reference:

1. <https://github.com/WaitingLin/ML2017FALL>