

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

(collaborator:)

latent vector dim = 512

normalize1: 將 rating 除 5 normalize 到 0~1 之間

normalize2: $(x - \text{mean}) / \text{std}$

	without normalize	normalize1	normalize2
Public score	0.84555	0.86653	0.85910
Private score	0.84786	0.86627	0.86006

準確率 without normalize > normalize2 > normalize1

2. (1%)比較不同的 latent dimension 的結果。

(collaborator:)

training data 不做 normalize

dimension	128	256	512
Public	0.84551	0.84544	0.84555
private	0.84638	0.84625	0.84786

準確率 latent dim 256 > 128 > 512，但三者差距微小，對準確率影響不大，我認為是在這個 hw5 的 case 中 latent dim 128 已經足夠包含所需的資訊，所以 dim 再增大不會有效率的增加準確率。

3. (1%)比較有無 bias 的結果。

(collaborator:)

latent vector dim = 512

	bias	Without bias
Public	0.84555	0.84662
Private	0.84786	0.84727

可以看到 bias 對這個 model 影響不大，public 變壞、private 變好，兩個平均起來相差不多。

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:)

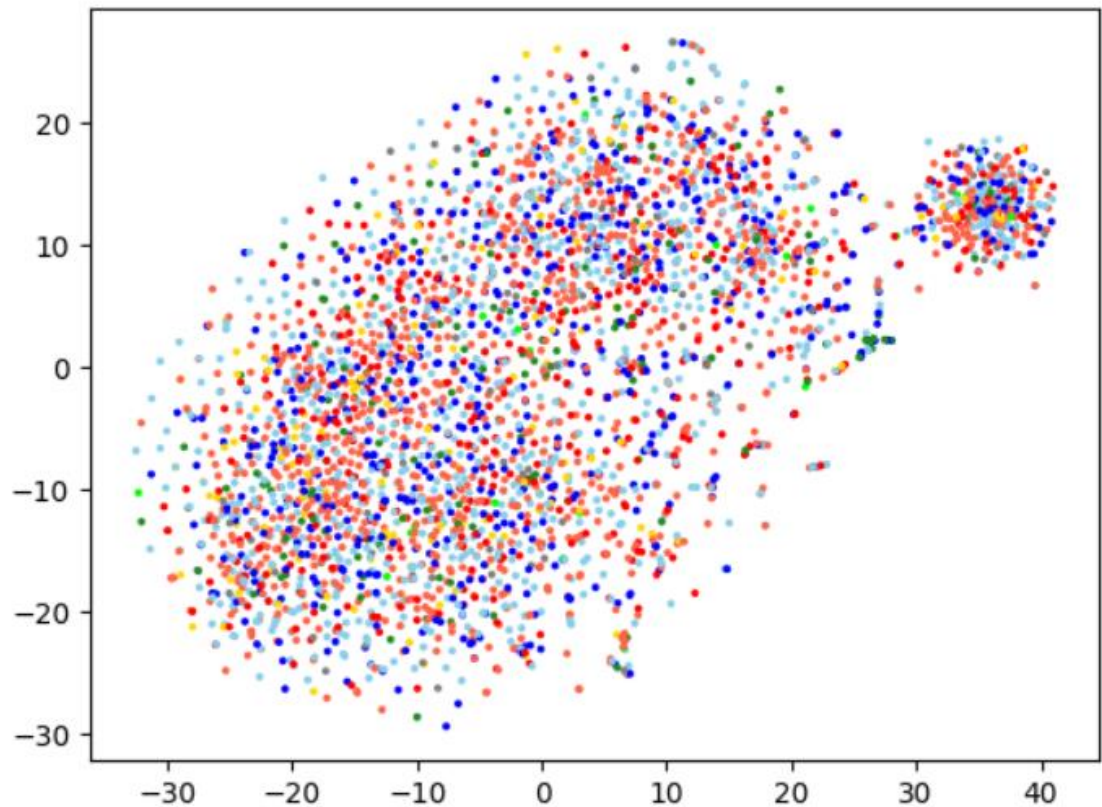
5. latent vector dim = 512

DNN: 把 user 跟 movie 的 latent vector concatenate 在一起，丟進三層 DNN(512, 512, 1)的 DNN。

	MF	DNN
Public	0.84555	0.91857
Private	0.84786	0.91625

可以 DNN 並沒有比 MF 優秀。

6. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。
(collaborator:)



7. (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:)

我利用 (5) DNN model 加入電影分類的 vector 每個 vector 18 維，屬於的分類，vector 的相對應維度就設為 1。

	DNN	DNN + classes
Public	0.91857	0.91120
Private	0.91625	0.90894

可以看到加入電影的分類可以提高預測分數的準確率。