

學號：B04701232 系級： 工管三 姓名：陳柔安

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

**normalize 方式：**將 `training data` 的 `rating` 做 `Z-normalize`(減掉平均除以標準差)，`testing data` 預測完之後再將它乘以 `training data` 的標準差、再加上 `training data` 的平均。

無 `normalize`：0.86037      有 `normalize`：0.85973

以此結果看來，有無 `normalize` 的結果並無太顯著的差別，但有 `normalize` 結果稍微好一點。

2. (1%)比較不同的 `latent dimension` 的結果。

latent dimension	32	64	128	256
rmse	0.86274	0.86037	0.86201	0.86159

由結果可發現 64 dimension 的結果是最好的，但不同 `latent dimension` 沒有非常顯著的差異。

3. (1%)比較有無 `bias` 的結果。

有 `bias`：0.86037      無 `bias`：0.86159

從結果看來，有 `bias` 的結果較無 `bias` 的結果好一點，推測因為加入 `bias` 代表有考慮到個別 `movie` 以及 `user` 的喜好，因此在數據上會有較好的結果。

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。

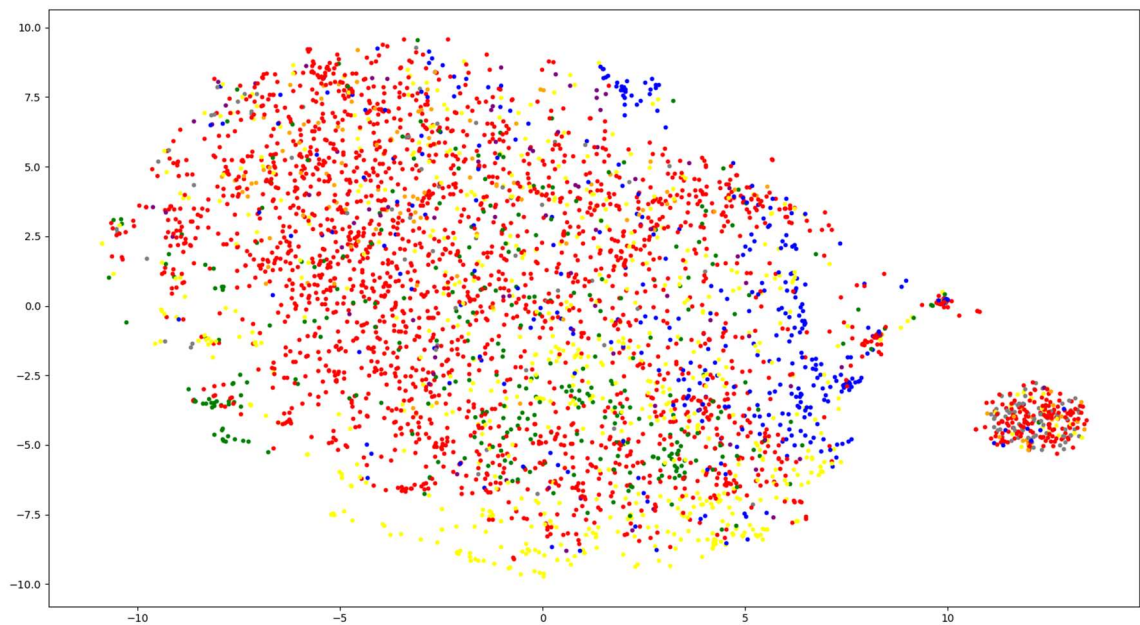
**DNN 作法：**把 `user` 以及 `movie` 做 `embedding` 之後，把兩個 `vector concatenate` 在一起當作 `DNN` 的 `input parameter`，用以下的 `model` 做有 `dropout` 與沒有 `dropout` 兩個版本。

實作方法	MF	DNN(EMBEDDING_DIM = 64 BATCH_SIZE = 1024 EPOCH = 200 DROPOUT = 0.4 )	DNN(EMBEDDING_DIM = 64 BATCH_SIZE = 1024 EPOCH = 200 DROPOUT = 0)
rmse	0.86037	0.91706	0.91026

`MF` 的 `loss score` 明顯較優於 `NN` 所得到的結果，推測可能原因為 `NN` 的參數量較大，因此還沒有試到一組比較合適的參數（疊幾層 `DNN`、每一層的 `Neuron` 數等）。另外推測是因為 `DNN` 在計算時是把 `movie` 和 `user` 的 `embedding` 結果當作是 `input` 的參數，只要是不同的 `movie/user` 組合就會是一組不同的 `input`，但 `MF`

作法是將 user movie 向量內積，所以 train 出來的 embedding vector 會有比較好的結果。

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



紅色：Drama / Musical / Comedy / Romance

藍色：Thriller / Horror

綠色：Adventure / Animation / Children's

黃色：Action / Crime / War

紫色：Sci-Fi / Mystery / Fantasy

橘色：Documentary