

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

把 training data rating 做 normalize(減掉平均再除標準差)，然後直接下去 train，預測 test data 的時候將結果乘以 training data rating 的標準差，再加上 training data rating 的平均。

我的實驗結果是做 normalize 會比較差 (0.86065, 0.88901)

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

50 -> 0.87904

100 -> 0.88089

150 -> 0.86904

200 -> 0.86065

3. (1%)比較有無 **bias** 的結果。

通常加上 bias 會比較好，因為每部電影跟每個使用者都有自己獨立的特性(普遍評價很高的電影 or 看電影很挑的人 等等)

同樣是 200 dim

有加電影的 bias -> 0.89013

有加使用者的 bias -> 0.87201

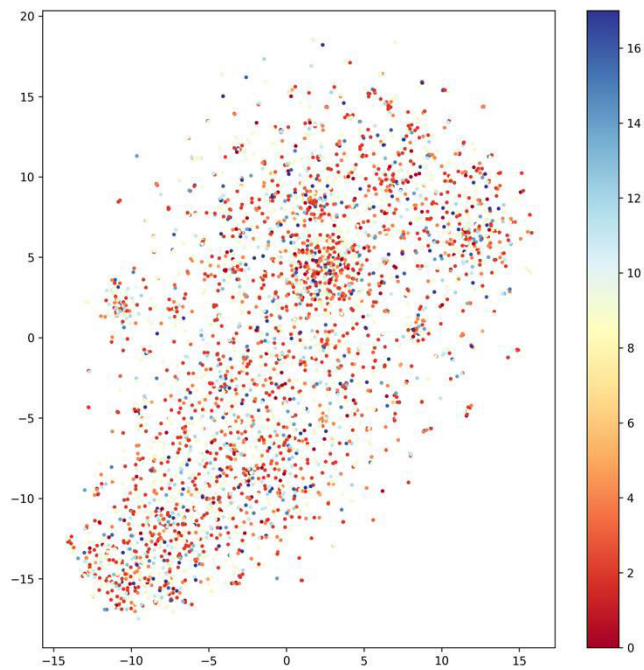
有加使用者跟電影的 bias -> 0.86065

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。

我實作的方法是把兩個 embedding layer 跟把 embedding layer dot 起來的 layer(總共三個)concatenate 起來，接到三層 dnn(512, 256, 1)

我做出來的結果是上面的方法會表現的比純 MF 好上一點 (0.85801, 0.86065)

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



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。