ML HW6-

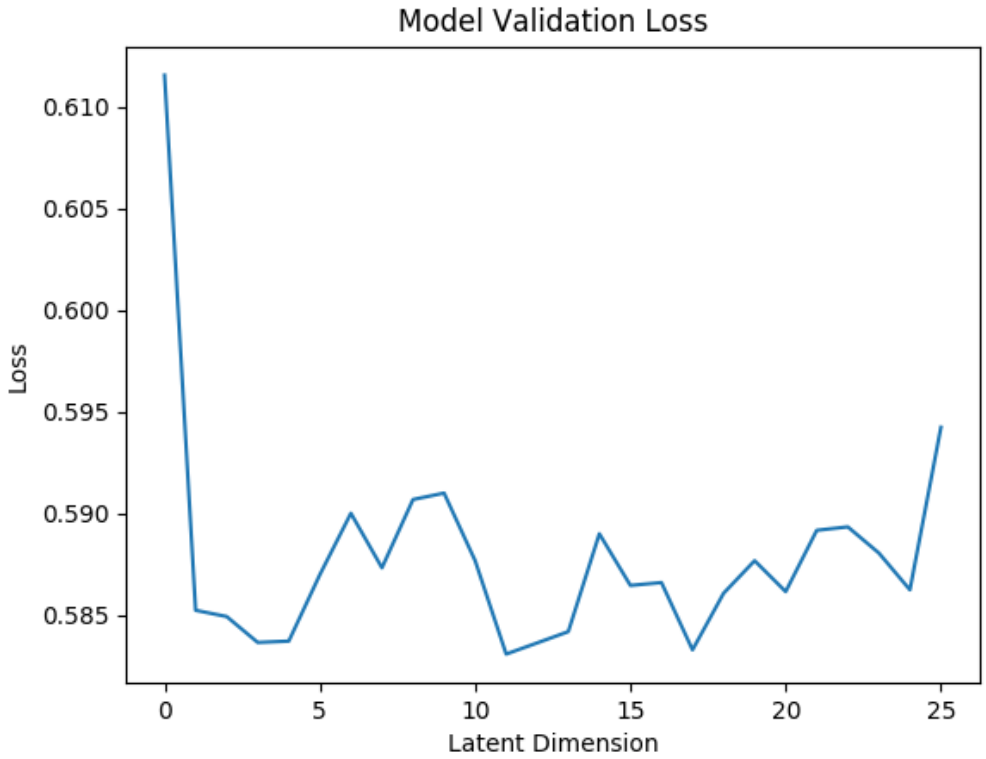
學號：R05943005 系級： 電子所碩一 姓名：呂丞勛

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

Without normalized	With normalized
0.87350	0.86213

用全部的 training 來取 mean 和 Std 後 normalized，送進 Training 後的 model 來預測 test ratings，在乘上 std 加上 mean 後得到最終結果，normalized 可以減少評分的小差異，因此通常 normalized 後訓練出來的 model 較好。

2. (1%)比較不同的 latent dimension 的結果。



利用 lantent 測試[0:130]，以五個距離當做 sample，所以上圖應該要把每一點乘上 5，從 validation 的 loss 看來，可以發現在大約 4(x5)或是 11(x5)的地方可以得到最低的 loss，也就是 embedding 後的結果的 matrix，會將結果轉為一個 20 或 55 維度的 vector 可以得到較好的 Validation 結果。

3. (1%)比較有無 bias 的結果。

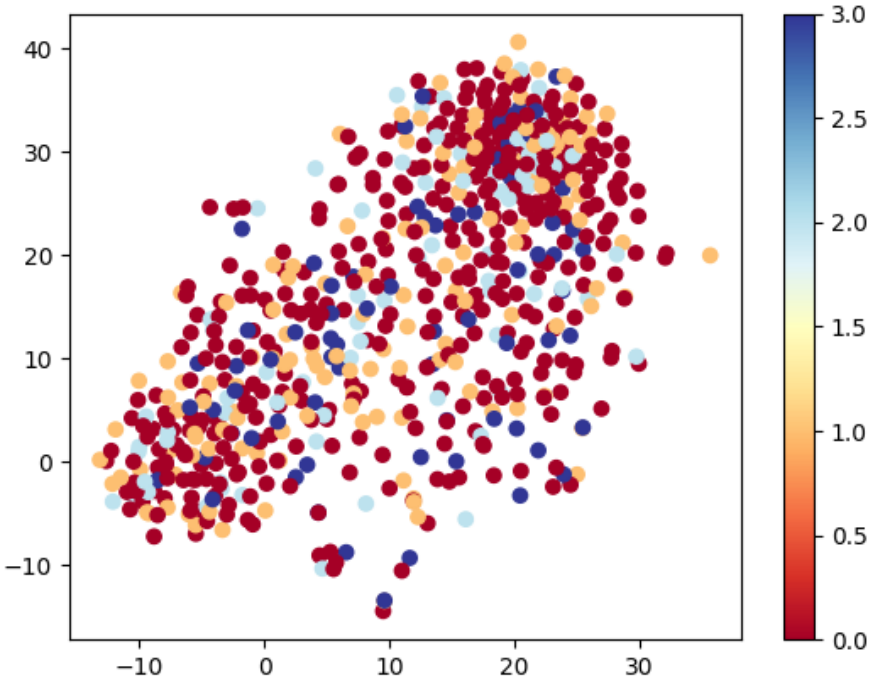
With bias	Without bias

0.87350	0.87917

有 bias 的結果較好，由於每個 user 的評分機制不太一樣，有人的評分可能偏向於較高的結果，如果加入 user bias 可以減少這個誤差，而 movie bias 的則可能由於較賣作的電影會有相對比較高的評分，也可以由此減少 bias 的影響。

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。
   使用 MF 主要只能根據拆解的維度來 optimize 每個矩陣的 elements，若使用 NN 則可以除了利用拆解矩陣之外，還可以拿拆解完的矩陣當做 feature 來丟入 NN 內部訓練，或是拿拆解完的矩陣做不同的運算後(dot, concatenate, add)來做為 feature，這內部可能包含了不同的資訊，可以幫助 NN 來做更準確的判斷，因此，我使用 NN 的結果在 kaggle 上的成績較好。

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



與投影片相同，我將 Drama, Musical 分為一類(y=1)，Thriller, Horror, Crime 分為一類(y=2)，Adventure, Animation, Children's 分為一類(y=3)，其餘分為一類(y=0)，畫出 666 個資料，可以看出大致上有分開，但不算明顯。

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。