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

   我使用 MIN_MAX scaling 的方式標準化 rating 分數，然而做出來效果很差，同樣的參數下與沒有做標準化的 Kaggle 分數落差達 0.11 (0.97 與 0.86)，可見此題 normalize 之後並不會有幫助。由於 rating 都是固定在 1-5 分的區間，因此標準化資料以邏輯來說確實不需要。

2. 比較不同的 latent dimension 的結果。

   經過實驗之後發現 latent dimension 越大 ，收斂的速度越快，Error 也能做到越小，然而由於 CPU 效能不太足夠，因此只將 iteration 設定 200 次並做到 K = 5， 無法測試當 latent dimension 極大狀況如 K = 100 時的情況。 我個人推測當 latent dimension 極大時會容易產生 overfit。

Dimension	Error
K = 1	733866.12
K = 2	673984.34
K = 3	651721.59
K = 4	635745.21
K = 5	610880.03

3. 比較有無 bias 的結果。

   在 K = 5, iteration = 200 的情況下，有無 bias 的 kaggle 成績只差 0.002 (0.866 與 0.864) 在誤差的範圍內，因此 bias 並不會造成決定性的影響。

4. 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。

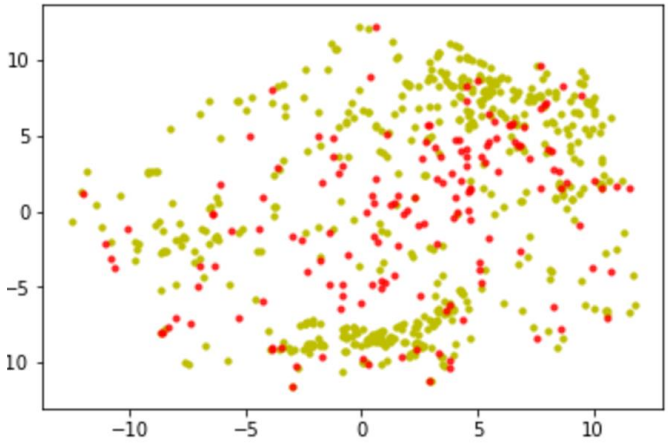
   以下是我的 DNN model，我採用三層 hidden layer，每層 512 個 neuron，根據幾次實驗後發現用 regression 準確率比分類高出許多，因此最後 output 採用 RELU 而非 Softmax。由於能用 Gpu 訓練，因此相比普通 Matrix Factorization 大幅縮訓練時間，一個 epoch 約 2-3 秒。最好的結果 validation error (MSE)能做到 0.697，kaggle 則拿到 0.846 左右的分數。MF 則只做到 0.863。我認為 MF 較像是傳統的的 gradient descent，沒有 DNN 裡面每一個 layer 都做 linear transformation，因此能達到較精準的結果。

```
movie_input = keras.layers.Input(shape=[1])
movie_vec = keras.layers.Flatten()(keras.layers.Embedding(n_movies + 1, 100)(movie_input))
movie_vec = keras.layers.Dropout(0.275)(movie_vec)

user_input = keras.layers.Input(shape=[1])
user_vec = keras.layers.Flatten()(keras.layers.Embedding(n_users + 1, 100)(user_input))
user_vec = keras.layers.Dropout(0.275)(user_vec)
input_vecs = keras.layers.concatenate([movie_vec, user_vec])
nn = keras.layers.Dropout(0.35)(keras.layers.Dense(512, activation='relu')(input_vecs))
nn = keras.layers.normalization.BatchNormalization()(nn)
nn = keras.layers.Dropout(0.35)(keras.layers.Dense(512, activation='relu')(nn))
nn = keras.layers.normalization.BatchNormalization()(nn)
nn = keras.layers.Dropout(0.35)(keras.layers.Dense(512, activation='relu')(nn))
result = keras.layers.Dense(1, activation='relu')(nn)
#result = input_vecs
```

5. 請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。
根據幾次作圖後，我發現幾組畫出來較相近的分類為 :

1. Action & Adventure : 黃色為 Adventure，紅色為 Action，可以看出兩者分布均靠右上方，左上方則較稀疏。



2. Horror & Thriller : 黃色為 Horror，紅色為 Thriller，可以看出兩者的分布較靠近正上方、右下方，左下方則較稀疏。