學號：R05922096 系級： 資工碩一 姓名：李哲安

1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize.
   std = 1.1168977, mean = 3.581712
   train_rating = (train_rating － mean) / std
   predict_rating = predict_rating * std + mean

   normalize 前 kaggle 分數為 0.86203
   normalize 後 kaggle 分數為 0.85464
   結果有變好，而且 training 收斂的速度變快了許多。

2. (1%)比較不同的 latent dimension 的結果。
   dim = 50, kaggle 0.85823
   dim = 120, kaggle 0.86203
   dim = 200, keggle 0.86398
   latent dimension 提高後，分數反而變差了。

3. (1%)比較有無 bias 的結果。
   沒有 bias 時 kaggle 上分數為 0.85934。
   加入 bias 後 kaggle 上分數為 0.86203。
   分數沒有進步。

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 model 如下, K_FACTORS = 300

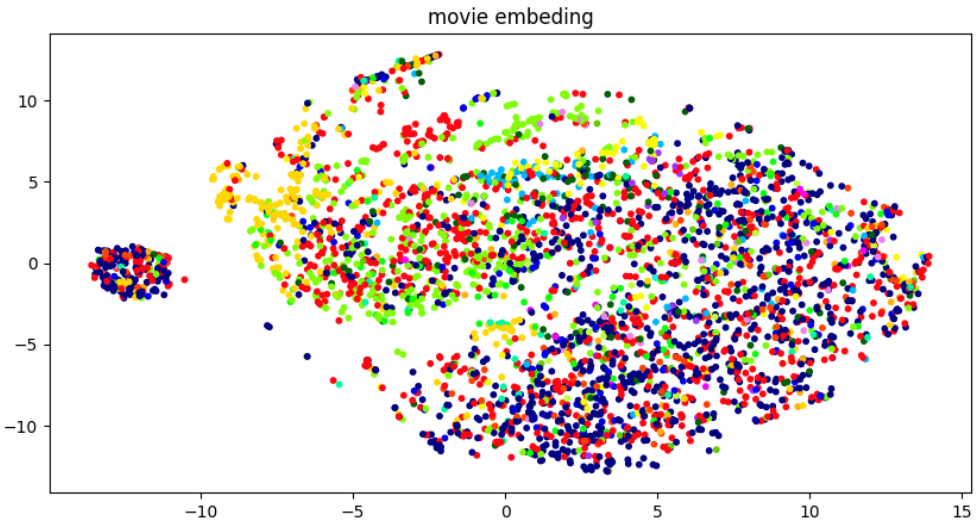
```
# input
input_user = Input(shape=(1,))
input_movie = Input(shape=(1,))

# user, movie ,user bias, movie bias
emb_user = Embedding(max_userid + 1, K_FACTORS, input_length=1)(input_user)
emb_movie = Embedding(max_movieid + 1, K_FACTORS, input_length=1)(input_movie)
emb_user_bias = Embedding(max_userid + 1, 1, input_length=1)(input_user)
emb_movie_bias = Embedding(max_movieid + 1, 1, input_length=1)(input_movie)

# flatten
U = Flatten()(emb_user)
M = Flatten()(emb_movie)
U_b = Flatten()(emb_user_bias)
M_b = Flatten()(emb_movie_bias)

# DNN
out_lay = Concatenate()([U, M])
out_lay = Dense(100, activation='relu')(out_lay)
out_lay = Dropout(0.5)(out_lay)
out_lay = Dense(50, activation='relu')(out_lay)
out_lay = Dropout(0.5)(out_lay)
out_lay = Dense(1)(out_lay)
out_lay = Add()([out_lay, U_b])
model = Model([input_user, input_movie], out_lay)

# compile
model.compile(loss='mean_squared_error', optimizer='adam')
```

   kaggle 上分數為 0.86422，跟 MF 分數 0.85823 比沒有進步。

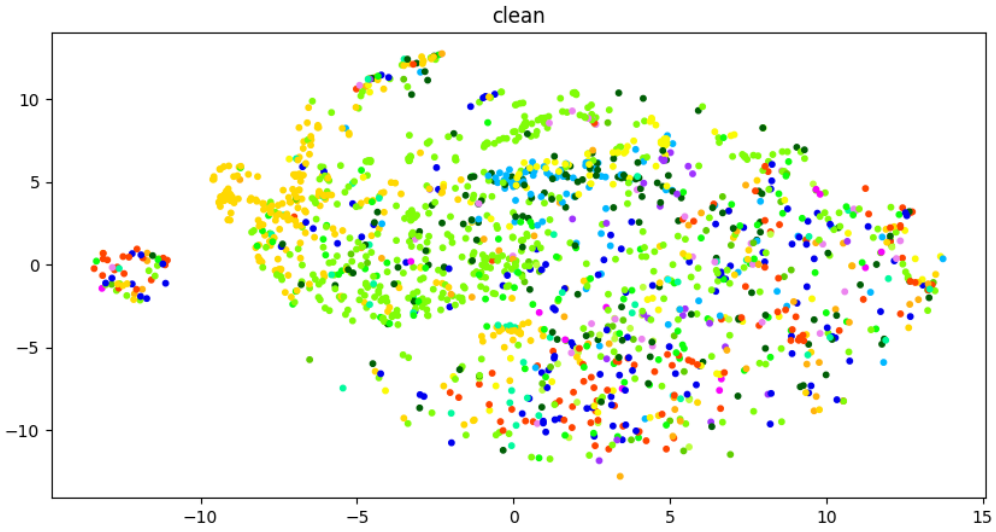
5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。

以下 latent dimension = 10

18 種 category 的圖



去掉 Comedy 和 Drama 後的圖



可以發現同樣的 category 的 movie 傾向聚在一起。