學號：R06944031　系級：　網媒碩一　姓名：林蔚廷

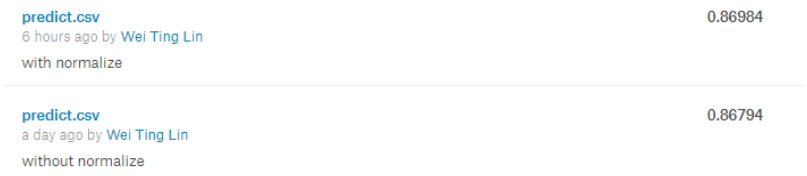
1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize.

   取 training data rating 的 mean 以及 standard deviation，再把 training data 的 rating 做 normalization 再用 normalized 過後的 rating 做為 training 的 target。在 testing 的時候則用同一組 mean 以及 standard deviation 還原並且將結果限制在 1~5 之間。

```
# normalize
if isNormalize:
    mean = np.mean(rating)
    std = np.std(rating)
    rating = (rating-mean) / std
```

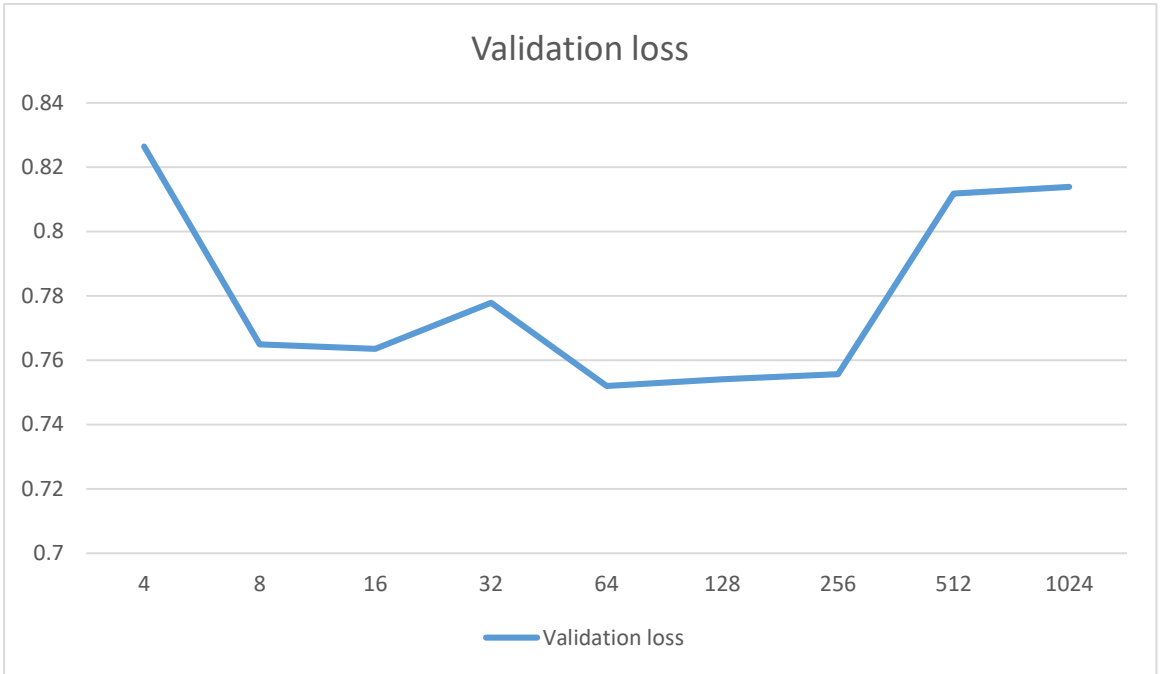
```
if isNormalize:
    prediction = prediction * std + mean
```

```
prediction = np.clip(prediction, 1, 5)
```



   幾乎沒差。

2. (1%)比較不同的 latent dimension 的結果。



   在 latent dimension = 8~256 結果都差不多，有點隨機性，但是可以觀察出太大或是太小的結果都不好。

3. (1%)比較有無 bias 的結果。

with bias:

```
Epoch 8/87
890874/890874 [==============================] - 3s 4us/step - loss: 0.6780 - val_loss: 0.7244
Epoch 9/87
890874/890874 [==============================] - 3s 4us/step - loss: 0.6537 - val_loss: 0.7250
Epoch 00009: early stopping
```

without bias:

```
Epoch 10/87
890874/890874 [==============================] - 3s 3us/step - loss: 0.6447 - val_loss: 0.7566
Epoch 11/87
890874/890874 [==============================] - 3s 3us/step - loss: 0.6255 - val_loss: 0.7586
Epoch 00011: early stopping
```

可以發現加入 bias 表現比較好。

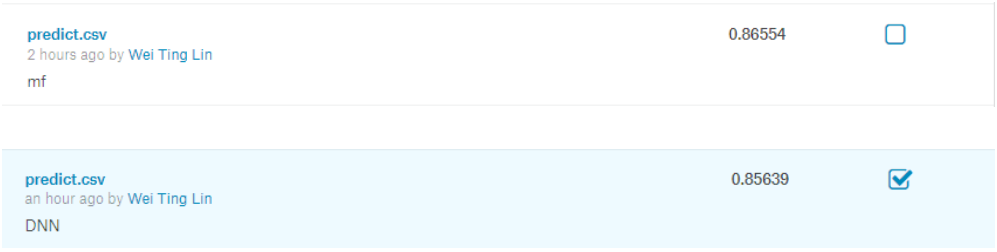
4. (1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。

我用一個 hidden layer(256 個 neuron)，再加上一層 dropout(0.7)。

```
# mf or dnn
if isDNN:
    model = dnn(6040, 3952, 512)
```

```
def dnn(n_users, n_movies, latent_dim = 32):
    user_input = Input(shape=[1])
    movie_input = Input(shape=[1])
    user_vec = Embedding(n_users, latent_dim, embeddings_initializer='random_normal')(user_input)
    user_vec = Flatten()(user_vec)
    movie_vec = Embedding(n_movies, latent_dim, embeddings_initializer='random_normal')(movie_input)
    movie_vec = Flatten()(movie_vec)
    merge_vec = Concatenate()([user_vec, movie_vec])
    hidden = Dense(256, activation='relu')(merge_vec)
    hidden = Dropout(0.7)(hidden)
    output = Dense(1)(hidden)
    model = Model([user_input, movie_input], output)
    model.compile(loss='mse', optimizer='adam')
    return model
```

結果:



DNN 的表現好一些。

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

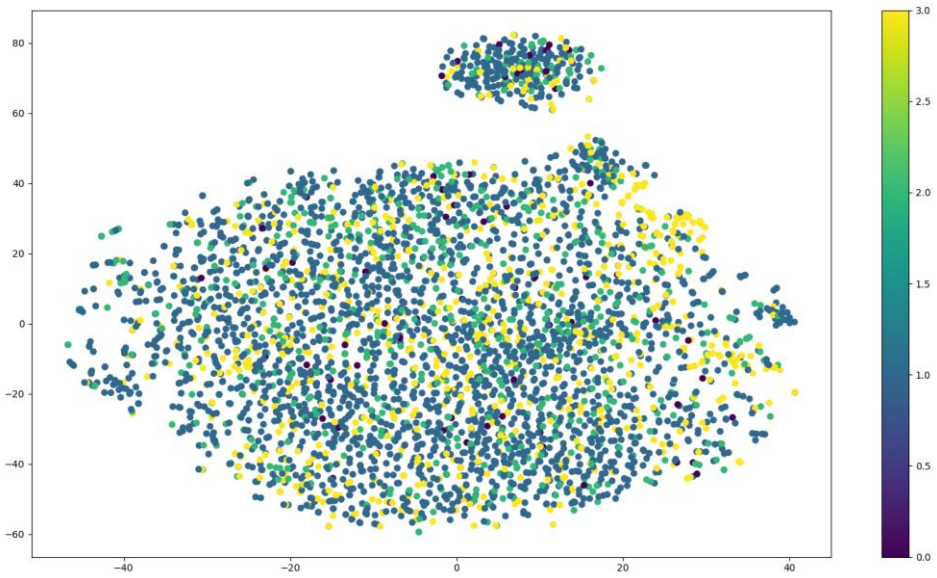
   我把 label 分成三類:

   1: Comedy, Fantasy, Romance, Drama, Musical, Sci-Fi, Animation, Children's

   2: Action, Documentary, Western

   3: Thriller, Horror, Crime, Mystery, War, Adventure, Film-Noir

   0: No label



   可以看到藍色(1)偏左，黃色(3)偏右。

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。

   我將 Age, Occupation 這兩個 feature 加入我的 DNN model，也是都經過 Embedding layer 後，與 userID, movieID 都 concatenate 起來，並經過一樣的 DNN layer。結果表現並沒有比較好，如下圖所示: