**SI 671 Assignment 1**
**Name:** Raphael Ku
**Kaggle:** Raphael Ku
**Uniqname:** rapku

**Option:** Option 1 (Use existing libraries)

**Final Approach:**
The approach used was the implementation of singular value decomposition in Surprise, where parameters were optimized to minimize root mean square error using GridSearchCV.

The parameters determined by GridSearchCV to optimize root mean square error in the training data focused on increasing training epochs (set to 10 epochs), a high learning rate to be used (set to 0.01), and a lower regularization factor (set to 0.3). These values are significantly higher than the default settings for the SVD algorithm, and most likely allowed for more aggressive changes in the bias terms in the SVD per iteration, while minimizing overfitting seen in the trained Spotlight recommender systems.

Lastly, the SVD with optimized parameters was chosen due to the algorithm having the lowest RMSE on the development data (RMSE on dev data: 1.015) compared to other algorithms used in both Surprise and Spotlight.

**Tried Approaches:**
Initial attempts used a package called Spotlight, which uses PyTorch to implement both an explicit and implicit recommender system.

Next attempts used the Python implementation on Surprise.

| **System** | **Parameters** | **Comments** |
|---|---|---|
| Package: Spotlight<br>Explicit Recommender System | 128 latent dimensionality<br>4 epochs<br>1024 minibatch size<br>1e-9 L2 regularization<br>1e-3 learning rate | First system that resulted in working recommendations.<br><br>Had to implement upper and lower bounds, as some scores being predicted on Spotlight were negative/above 5<br><br>RMSE on dev data: 1.071 |
| Package: Spotlight<br>Explicit Recommender System | Changes:<br>8 epochs | Same issues.<br><br>May be overfitted to training data<br><br>RMSE on dev data: 1.017 |
| Package: Spotlight<br>Explicit Recommender System | Changes:<br>100 latent dimensionality<br>6 epochs<br>L2 regularization set to 1e-7 | Made adjustments to reduce overfitting<br><br>Note: All Spotlight results returned RMSE of approx. 1.4000 on test data in Kaggle |

| | | |
|---|---|---|
| | | RMSE on dev data: 0.9817 |
| Package: Surprise<br>SVD | Default settings | Attempts to make this work failed previously until file conversion of data to csv<br><br>RMSE on dev data: 1.022 |
| Package: Surprise<br>Non-negative matrix factorization | Change algorithm to prebuilt implementation of Non-negative matrix factorization | RMSE on dev data: 1.1097 |

**Dev Prediction Error Analysis**

| User ID | Item ID | Score (actual) | Score (pred) |
|---|---|---|---|
| AV6QDP8Q0ONK4 | B007IUEDYY | 2.0 | 3.2001 |
| A3BQCZNB97XUNY | B000063EME | 5.0 | 3.9482 |
| A3EBHHCZO6V2A4 | B003BWQEMM | 5.0 | 3.6658 |
| A2RMG94SRG1SD9 | 6301661761 | 3.0 | 4.2649 |
| A27BY97QQS36V3 | 0767814908 | 1.0 | 3.0982 |

In terms of all user-item pairs listed here, it is interesting to note that all user and item IDs here are represented in the training data. Additionally, all predicted values are relatively distant to the mean of all observations in the training data (4.1110). Thus, the difference in the predicted and actual scores are not due to encountering a new item or user.

For user AV6QDP8Q0ONK4 and item B007IUEDYY (1st row), looking at their number of reviews on the training dataset, as well as their style of writing in the review text (provides a parental guide), this user is an experienced user of the website/system. In fact, the user is the 9th most common user in the training dataset. This may mean that the error indicated here may be due to the large number of items reviewed, their user embedding could be similar to a wide range of other users, leading to a rating that represents an average of a wide pool of users that may not match this user's profile and experience.

For user A3BQCZNB97XUNY (2nd row), looking at figure 1 below, we see that the SVD algorithm may not be considering strange voting patterns for a user. For A3BQCZNB97XUNY, they have an extreme voting pattern that seems to have a small positive bias against the average when aggregated.

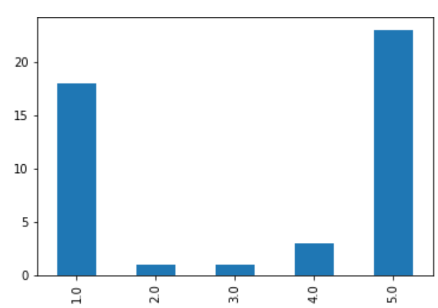


*Figure 1: Distribution of ratings for user A3BQCZNB97XUNY*

Looking at user A2RMG94SRG1SD9, item 6301661761 (4th row) and A27BY97QQS36V3(5th row) share an opposite issue, in which each user provided very few ratings to begin with (8 ratings for user

A2RMG94SRG1SD9 and 14 for user A27BY97QQS36V3). Given how close the rating user A2RMG94SRG1SD9 gave for item 6301661761 to the total mean of all item-pair ratings, we could assume that the sparsity of ratings led to a low user and item bias.

Lastly, for user A3EBHHCZO6V2A4 and item B003BWQEMM, looking at the review details, it seems strange that, despite a generally dismissive tone, with words indicating some weaknesses in the item, the rating is still set to 5.0. Comparatively, this user has a significant range of ratings [1-5], but generally rates items as 5s, which may be behavior not captured by the SVD algorithm.

**Areas for improvement:**
It seems that it may be useful to embed information to group users based on the number of items reviewed in total. It could be possible to run separate SVD algorithms based on user behaviors and compare the resulting values to an SVD algorithm using all users and items, with weights to represent the importance of each model.

Additionally, since the bias is a single number, it fails to reflect the issue we encountered with extreme scoring. It may be useful to implement an algorithm that generates ratings as a function of the probability of a user rating in a certain way and the cosine similarity to other users, so the information on the distribution of their scoring can be maintained.