**Script:**

\*\* have a look at the other data files given

* **Union and Missing Values**

This solution first computes the union of items rated by either user ‘a’ or ‘u’, rather than taking the intersect. This allows for a greater number of samples in the training data. As we are finding the union and it is not confirmed that both users have predicted that item, therefore it is likely that there will be missing values. These missing values are replaced with the user’s mean rating. Calculating the union is helpful as the number of movies a user may have rated could be extremely small, and finding movies that both users rated could be even smaller. By calculating the union and including movies both users have rated allows the sample size to be increased. This allows for a potentially more accurate model, therefore improving the performance of the user KNN-based Collaborative Filtering method. The missing values being replaced with the user’s mean rating also improves the rating. This is because if the missing values are not replaced, then a rating of 0 would be given. This would suggest to the model that the user did not like the movie, however, a user not watching the movie does not mean that the user doesn’t like the movie. Therefore, by imputing the mean, this problem does get solved.

* **Weight Considering Item’s Popularity**

When doing the similarity calculation, the solution provided in the report considers the popularity of the different items unlike the original method. This is done by calculating the log of the total number of users in the database and dividing it by the number of users that rated that item. Considering the weight of each item is important as it allows the method to differentiate between popular and less popular items. This is important as unlike the original model, this considers the impact of the long-tail effect. The long-tail theory. The long-tail theory suggests that the is a small selection of items with extremely high popularity and demand, and most items are in the long tail and are not that popular. The solution in the report does this by squaring the weight of the popularity in the formula to find similarity. The weight is calculated by finding the log of the total number of users in the database divided the total number of users that rated that item. Here, a larger number of users that rated the item would result in a smaller value for the weight, and therefore hold less significance. Items that are less popular and have a smaller number of users that rated the items would have a higher weight and therefore have greater significance. This means that the solution provided in the report is less likely to just predict super popular movies and is more likely to predict movies in the long tail, and therefore less popular movies.

* **Code**

First the value for ‘P’ was calculated. This was done by finding the sum of all non-zero values in each column. Non-zero values suggest that this item was rated.

Then the union set was found. This was by finding the union between the vector for user ‘a’ and user ‘u’. This was where either user has ‘true’ meaning they had rated that item.

**– Explain how the solution in the provided report can improve the performance**

**of the user KNN-based Collaborative Filtering method by using your own**

**language clearly and completely.**

Union and filling in the missing values.

The formula.

This solution first computes the union of items rated by either user ‘a’ or ‘u’, rather than taking the intersect. This allows for a greater number of samples in the training data. As we are finding the union and it is not confirmed that both users have predicted that item, any missing values are replaced using the user’s mean rating. When doing the similarity calculation, the solution provided in the report considers the popularity of the different items unlike the original method. This is done by calculating the log of the total number of users in the database and dividing it by the number of users that rated that item.

**– Explain why the solution in the provided report can improve the performance**

**of the user KNN-based Collaborative Filtering method by using your own**

**language clearly and completely.**

Union and filling in the missing values. (Why?)

The formula (Why?)

Calculating the union is helpful as the number of movies a user may have rated could be extremely small. By calculating the union and including movies both users have rated allows the sample size to be increased. This allows for a potentially more accurate/precise model, therefore improving the performance of the user KNN-based Collaborative Filtering method. The missing values being replaced with the user’s mean rating also improves the rating. This is because if the missing values are not replaced, then a rating of 0 would be given. This would suggest to the model that the user did not like the movie, however, a user not watching the movie does not mean that the user doesn’t like the movie. Therefore, by imputing the mean, this problem does get solved. Considering the weight of each item is important as it allows the method to differentiate between popular and less popular items. This means that more popular items have less significance compared to less popular items. This considers the long-tail effect which the original model does not. So, instead of just recommending popular movies, these movies have a lower significance. This models allows for the movies to be recommended in the long-tail which is made up of less popular movies, and these movies have a higher significance in the model. This is because when dividing the total number of users by the number of users who have rated the movie, this number is higher with a lower value for P.

**– Explain how you implement the solution clearly and completely.**

Explain the code.

ComRV is not found as it does not have anything to do with the MAE and RMSE.

Maybe, in each slide have the how and the why for that one part on the same slide.