MovieLens Project Report

Lee Hagendoorn

March 7, 2019

Introduction

The primary objective of this project is to create a movie recommendation algorithm to predict a movie rating a person would select based on existing data. We utilized an existing data set of movie ratings created by a research lab at the University of Minnesota called GroupLens that compiled more than 10 million movie ratings. The data was utilized to generate machine learning algorithms to best predict a person's rating for a movie. To rate the project as a success, we must create an algorithm that could predict at an accurate enough level to produce a Residual Mean Square Error (RMSE) of less than or equal to 0.87750.

Analysis

The data set used for this project was the MovieLens database created by GroupLens, a research lab in the Department of Computer Science and Engineering at the University of Minnesota. The database includes 10 million ratings for more than 10 thousand movies and 72 thousand users. Ratings are calculated in ½ star increments from 0 to 5 stars.

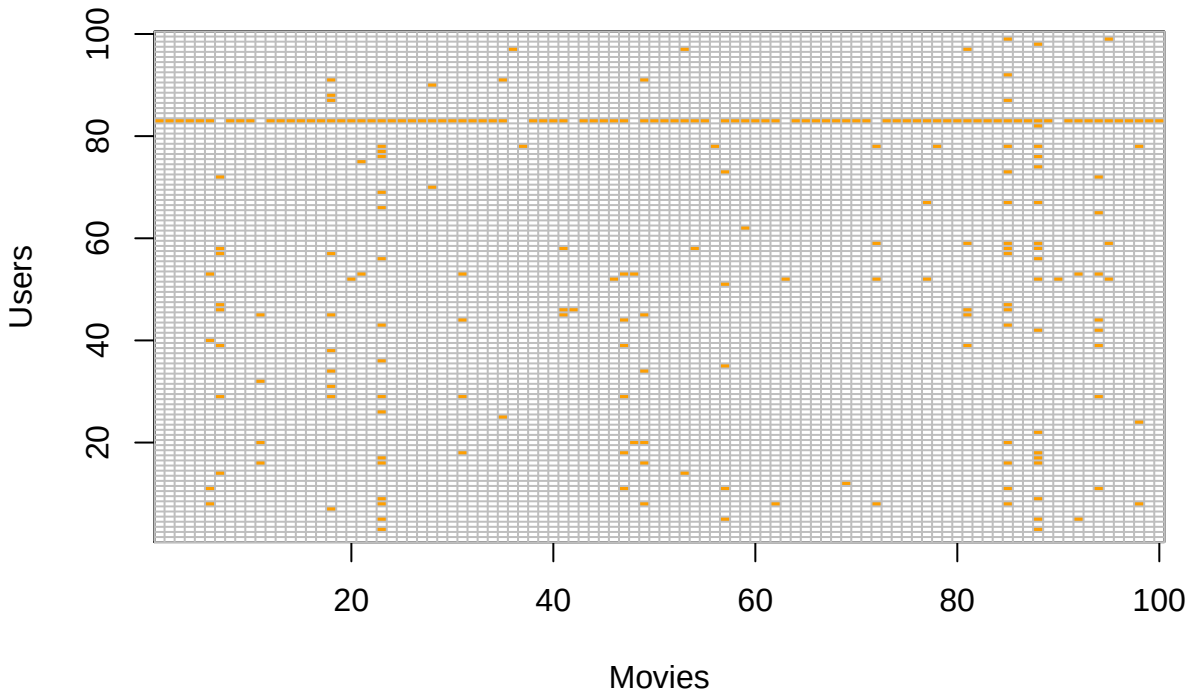
The task of the recommendation algorithm is to fill in the values for each movie that a user has not rated.

Initially we created a training data set to be used to create our rating prediction algorithm and a test data set that will be used to determine how well our algorithm predicts ratings in this data set. The training set included 90% of the data, with the remaining 10% in the test set. We only kept movies and users in the test set that were also in the training set.

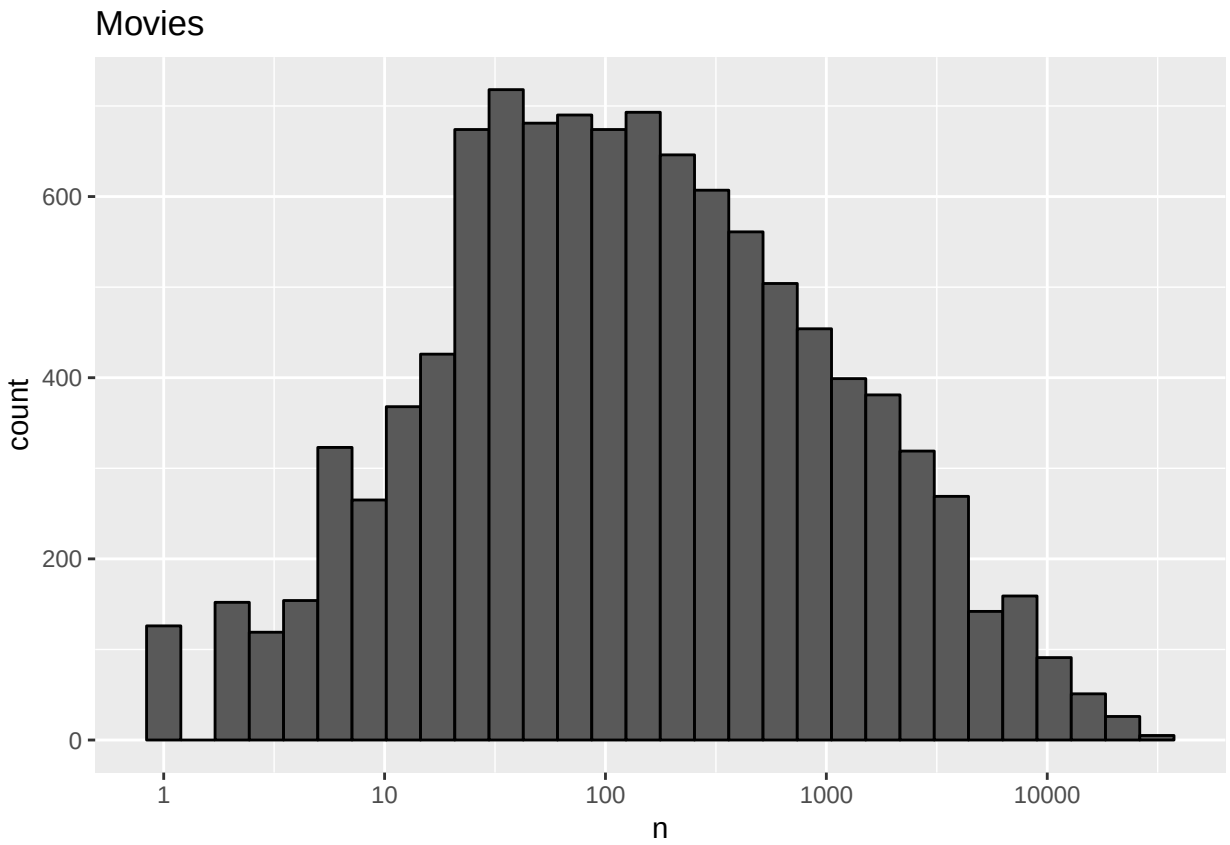
The training data set includes 9,000,055 rows in which each row represents a rating given by one user to one movie. Within this dataset there are 10,677 distinct movies rated and 69,878 users. If each user rated all movies, we would have 746,087,406 rows in the data set. However, we only have roughly 9 million rows in our training set, which indicates not all users rated every movie.

The test data set included 999,999 ratings, 9,809 distinct movies rated and 68,534 users.

We could think of the data sets as very large matrices with users on the rows and movies on the columns with many empty cells. The chart below shows a matrix with a random sample of 100 users and 100 movies from the training data set. The orange spaces show the movies that were rated by each user. You can see that most users only have only rated a small subset of movies.



You can see in the following chart that some movies get rated much more than others. There is somewhat of bell-shaped curve showing an approximately normal distribution of the number of ratings per movie.



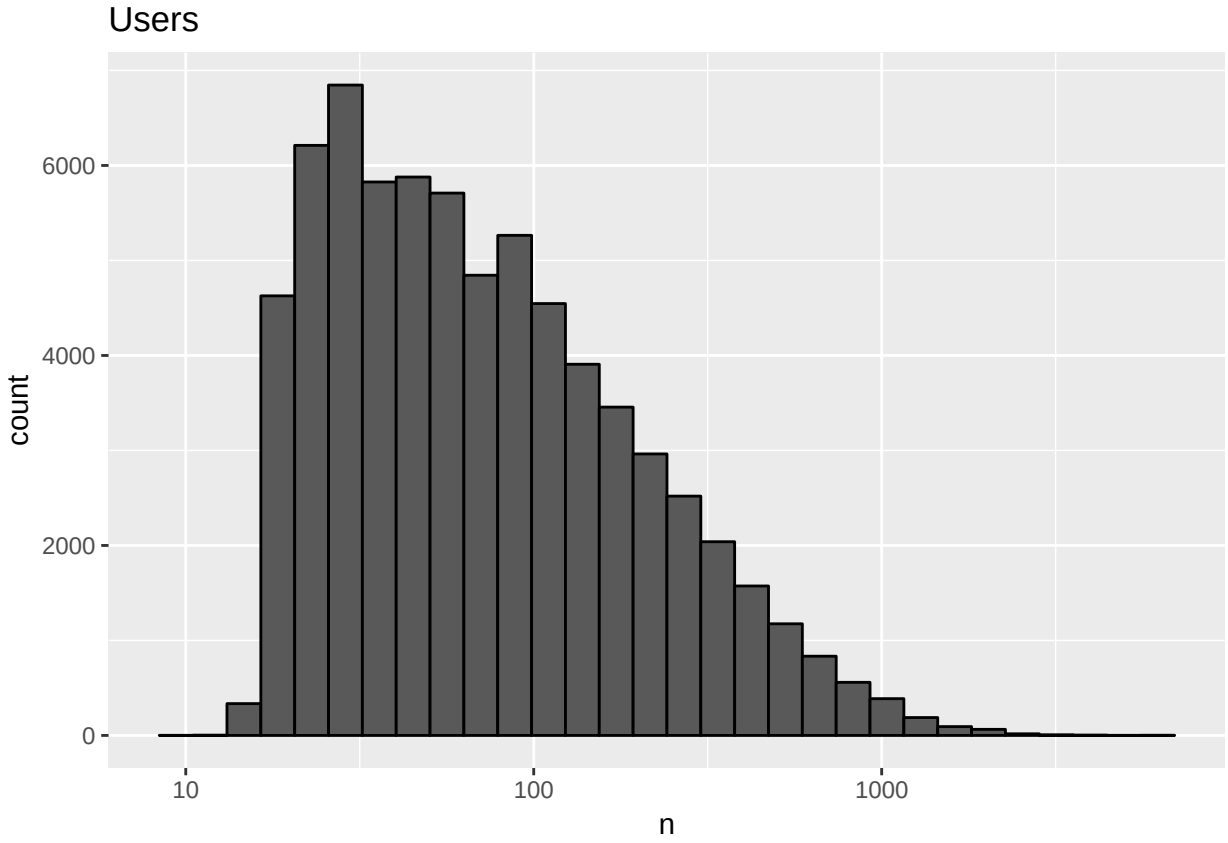
Below is a table of movies that have received the most ratings. You may notice many of these are well-known, blockbuster-type movies. You can also see that some of the movies with the fewest ratings are those with less popular names.

title	n
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015
Braveheart (1995)	26212
Fugitive, The (1993)	25998
Terminator 2: Judgment Day (1991)	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
Apollo 13 (1995)	24284

title	n
1, 2, 3, Sun (Un, deuz, trois, soleil) (1993)	1
100 Feet (2008)	1
4 (2005)	1
Accused (Anklaget) (2005)	1
Ace of Hearts (2008)	1
Ace of Hearts, The (1921)	1
Adios, Sabata (Indio Black, sai che ti dico: Sei un gran figlio di...) (1971)	1
Africa addio (1966)	1

title	n
Aleksandra (2007)	1
Bad Blood (Mauvais sang) (1986)	1

Also, the number of ratings per user can vary significantly. Some users rate movies much more than others as indicated on the chart below.

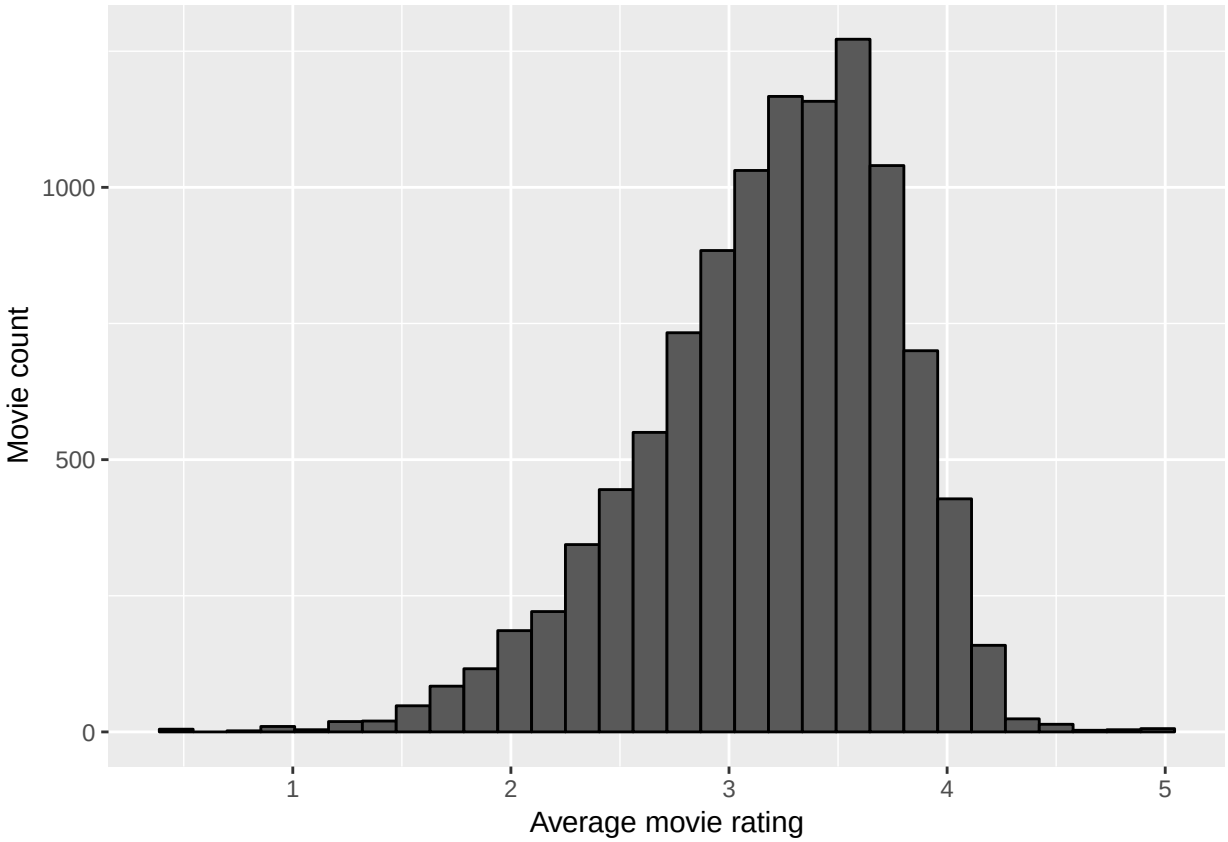


To measure our prediction results, we developed a loss, RMSE, function to see how close our prediction models are to predicting the results in our test data set. The RMSE function looks at the difference between the actual star rating in the test data set and our prediction for that star rating. An RMSE>1 would indicate our predictions were not very good as they would be more than 1 star away of the actual rating. Thus, our goal was to achieve an RMSE <= 0.87750 stars away from the actual star rating.

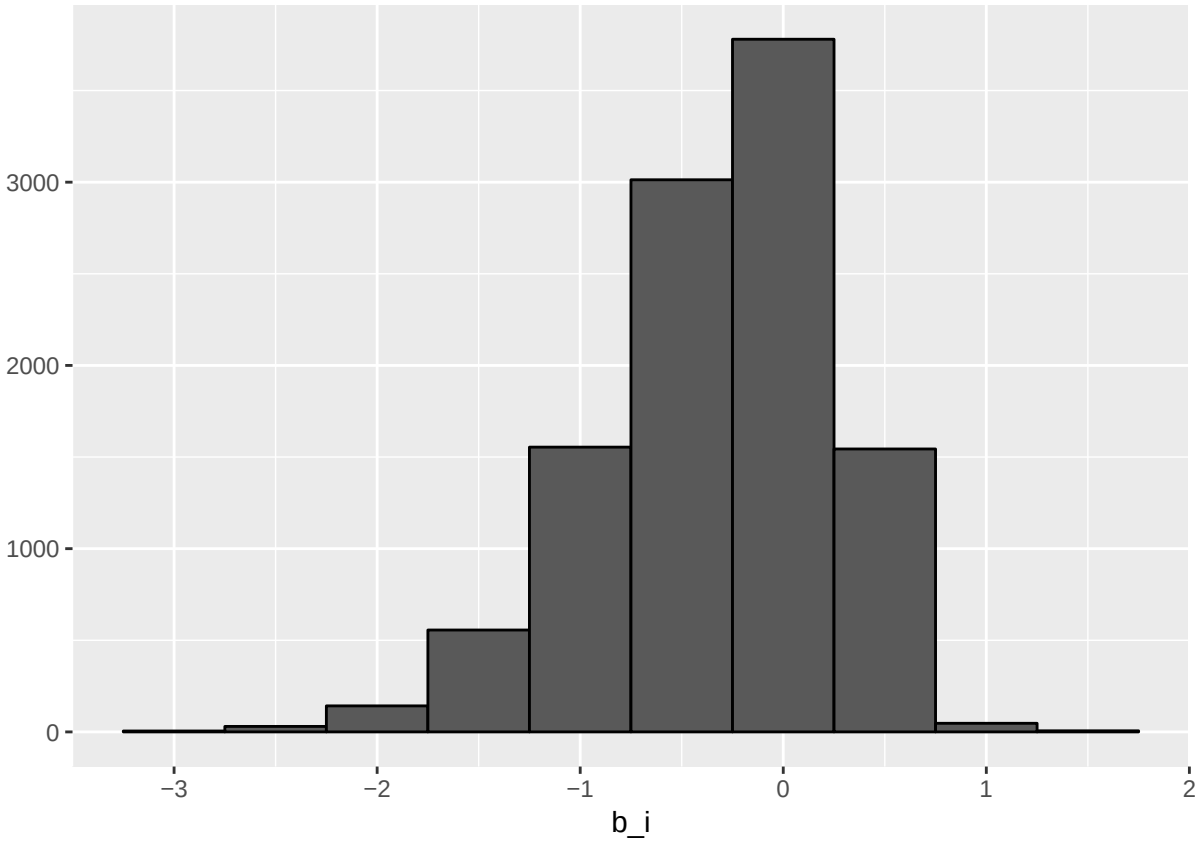
Results

For the first machine learning algorithm, we decided to use the same predictions for all ratings with the differences explained by random variation. Thus, the average rating was taken for the entire training set, which was 3.512 stars, and used to predict the ratings within the test data set. The formula is y_i_u = mu + e_i_u, where mu is the average star rating and e_i_u is the error for movie i and user u. Computing the RMSE using this model on the test data set resulted in an RMSE of 1.06. This is still far from our goal. However, if you plug in any other single, random, number as the predicted rating, the RSME was even higher. For example, plugging in 4 as the prediction for every rating, we get an RMSE of 1.17.

For our next model approach, we looked at the movie effect on ratings. You can see in the chart below that movies have varied average ratings.

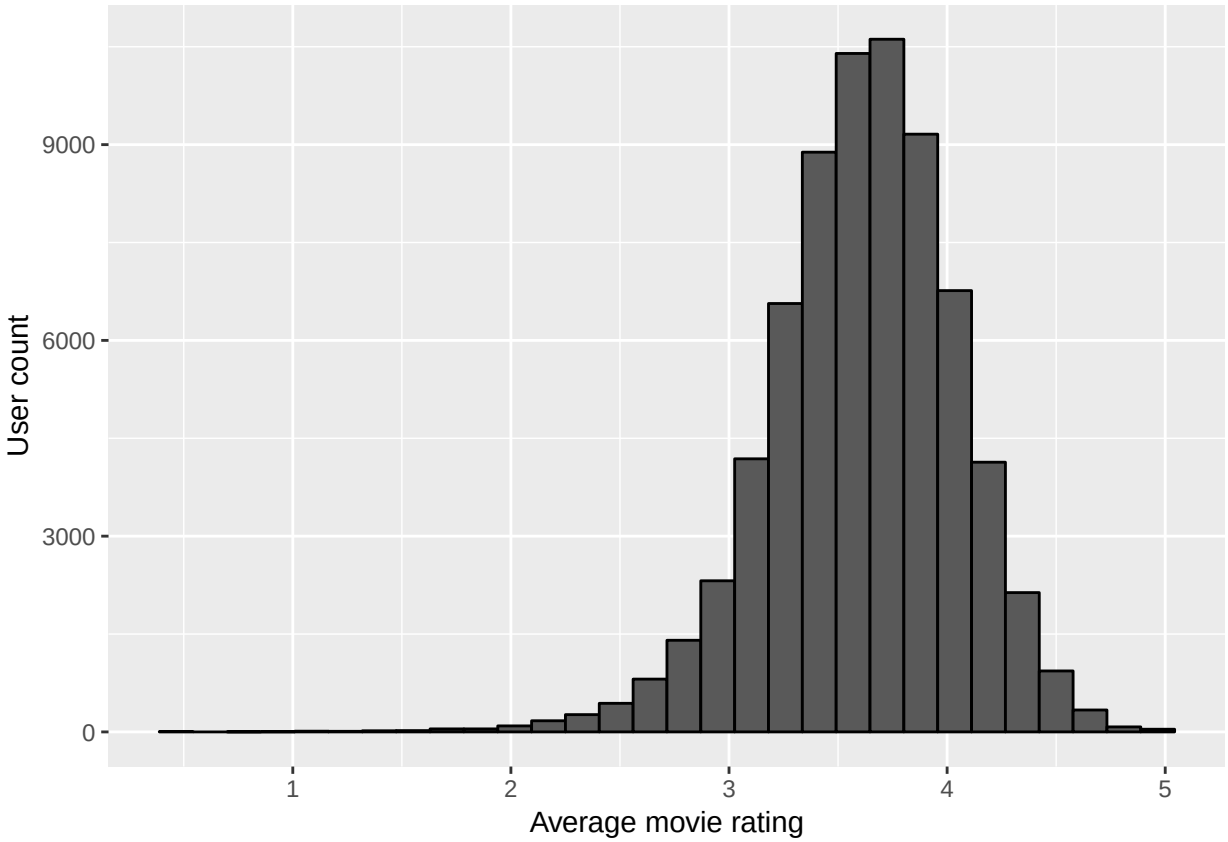


So, we now add a term to our model that represents the average rating for a movie, b_i. The new model is y_i_u = mu + b_i + e_i_u. The estimates for b_i differ substantially as some movies are seen, in general, better than others accross users as you can see in the chart below.



When we test our new model that includes the movie effect, the RMSE drops to 0.944395.

Now we look to improve our model by looking closer at users and their individual effect on the ratings. We see that some users rate differently on average than others. You can see the variation in average ratings by users within the chart below.



Thus, a user-specific effect, b_u, was added to the model. Our new model is y_i_u = mu + b_i + b_u + e_i_u. The results of including a movie and user effect yields an RMSE of 0.86535, which is a nice improvement and meets our goal of an RMSE <= 0.87750.

We then investigated if we could produce an even better model by digging deeper into the movie effect. When reviewing we found that the movies in which we saw the biggest error in rating prediction had very few user's rate the movie. Thus, we applied regularization to the model in which we shrunk our movie effect towards zero, if the number of user ratings for the movie was low. We used cross-validation to determine the optimal value used to shrink the movie effect, lambda, which was 2.5 and produced an RMSE of 0.94389. This is slightly better than the non-regularized movie effect model, but not as good as the model in which the movie and user effects were both taken into account.

Next, we decided to use regularization on the user effect as well to shrink the effect towards zero if a user had not rated many movies. Using cross-validation to find an optimal lambda for both the movie and user effects, we found the best lambda to be 5.25, which produced an RMSE of 0.86481. This result only slightly improves the original, unregularized, movie and user effect model by 0.00054.

Note we also looked at a model with genre and user effects, but this did not yield better results than our movie and user effect model at 0.94021.

Conclusion

Overall, we can produce a model that meets our objective to produce an RMSE of <= 0.87750 by including both movie and user effects. If we adjust the model to penalize estimates for a low number of ratings for a movie as well as a low number of ratings completed by a user, we can make the model even better.