London Business School

Movie Recommendation Algorithms

Zervaan Borok

Abstract

This report aims to illustrate the performance of four different recommendation systems on movie and user data. The data is comprised of 283,228 users and their corresponding ratings of 58,098 movies. The recommendation systems explored in this report are: Item Based Collaborative Filtering (IBCF), User Based Collaborative Filtering (UBCF), Single Value Decomposition (SVD), and Model Based Collaborative Filtering using Matrix Factorisation (LIBMF) (Appendix 1). The root-mean-squared-error (RMSE) was used to evaluate model performance (Appendix 1). The results of this analysis indicate that the best recommendation system for this data set is the LIBMF model as it attains the lowest RMSE of all the models and is also one of the least computationally intensive.

Introduction

Recommendation systems play a vital role in day-to-day business operations for some of the world's largest streaming platforms. Firms such as Netflix, Hulu, and Spotify all utilize recommendation systems to enhance user experience (Hinkle, 2021). These algorithms attempt to recommend new relevant items, whether it be music, TV shows, or movies, for any given user via memory-based or model-based algorithms. It has been shown that these algorithms help increase user retention and frequency of user usage (Hinkle, 2021). Therefore, one could argue that these algorithms are necessary for any streaming platform to succeed. The purpose of this report is to investigate which particular type of recommendation system performs the best when given a sufficiently large data set comprised of users and their corresponding ratings of movies. Next, we will examine the data cleaning and exploratory analysis that was conducted for this report.

Data Cleaning & Exploratory Analysis

Prior to beginning any analysis, the data had to be cleaned. Due to the very large size of the data set, I took a random sample of 10,000 users and their corresponding ratings of 21,106 movies. Following this, I removed duplicate movies. Some users gave the same movie different ratings, so in these instances, only the maximum rating was kept. I then created a ratings matrix with this filtered data in which user ID's corresponded to rows and movie ID's corresponded to columns before transforming the ratings matrix into a 'recommenderlab sparse matrix'. To avoid hampering model performance, I further filtered the data to only include users that had rated more than 50 movies and movies that had been rated by more than 50 users. I began my exploratory analysis by creating heatmaps and distributions of the data. The heatmaps of the first 100 users and first 100 films revealed that, in general, the users and movies in this data set are neither very similar nor very dissimilar

(Appendix 2, Figures 1,2). A plot of the distribution of the number of views for each movie showed that the data is very heavily right-skewed and that the vast majority of movies have less than 500 views (Appendix 2, Figure 3). Plotting the distribution of the average rating per user illustrated that over 81% of average user ratings fall within the range of (3.25, 4.5) (Appendix 2, Figure 4). We will now inspect the models constructed for this report.

Model Construction

I created two memory-based algorithms: User Based Collaborative Filtering (UBCF) and Item Based Collaborative Filtering (IBCF), and two model-based algorithms: Single Value Decomposition (SVD) and Model Based Collaborative Filtering using Matrix Factorisation (LIBMF). I split the cleaned data into training and testing sets, which were used for all four models. For each model, 80% of the data was used for training and 20% for testing. The root-mean-squared-error (RMSE) was used to evaluate the goodness-of-fit for each model. Under the UBCF model, the lowest RMSE attained was 1.1163. This occurred when the 'nearest neighbor' parameter was set to 50 (Appendix 2, Table 1). The lowest RMSE attained by the IBCF model was 0.8577 and occurred when the cross-validation parameter (k) was set to 750 (Appendix 2, Table 2). The SVD model achieved an RMSE of 0.8944 when the cross-validation parameter (k) was set to 100 (Appendix 2, Table 3). Finally, the LIBMF model achieved the lowest RMSE of all the models with a value of 0.8206. This was achieved when the latent factors tuning parameter was set to 12. As a result, I determined the best model for this data set was the LIBMF model and set about evaluating its performance on different data sets.

Model Performance & Sensitivity Analysis

To measure the performance and robustness of my chosen model, I evaluated it on five new data sets. The first data set included users that had rated more than 30 movies and movies that had been rated by more than 30 users. For the second data set, these threshold levels were both increased to 70. The third data set saw both of these thresholds increased to 100. In the fourth data set, the thresholds were both increased to 200. Lastly, the fifth data set had these thresholds set to 300. Each of these data sets was split into training and testing sets before being passed to the model. As before, the training proportion was set to 0.8 for each data set split and the RMSE was used to evaluate model performance. The table below contains the RMSE's of the chosen model, which correspond to the five new data sets.

Data Set	RMSE
Threshold of 30	0.8468154
Threshold of 70	0.7910748
Threshold of 100	0.8033744
Threshold of 200	0.8050031
Threshold of 300	0.7835364

As we can see, this model is quite robust. When infrequent users and movies are included in the data, the performance does deteriorate, but remains very reasonable. Naturally, the model performs best under the fifth data set as it is comprised of very frequent users and movies. In general, the performance of recommendation algorithms will suffer when they are used to predict recommendations for infrequent users. We will now examine the chosen model's predicted recommendations for two random users, user 500 and user 600.

To evaluate how well the model performs, we will compare the frequency of genres of the recommended movies with the frequency of genres of the movies actually viewed by these two users. 20 recommendations were predicted for each user. The model recommended movies of the genres 'Action', 'Adventure', 'Comedy', and 'Drama' for user 500, which neatly aligns with the genres of films this user has already watched (Appendix 2, Figure 5). For user 600, the model recommended movies of the genres 'Action', 'Adventure', 'Comedy', 'Romance', and 'Drama'. Again, these recommendations neatly align with the genres of films that the second user has already watched (Appendix2, Figure 6). As evidenced by the results of this exercise, we can see that this model is a relatively good recommendation system.

Conclusion

The results of this analysis indicate that some recommendation systems are better than others with regards to performance on film and user data. While all of the models explored in this report are valid recommendation systems, it is clear that the LIBMF model reigns supreme. When all models are passed the same data set, the RMSE of the LIBMF model is roughly 0.03 lower than the RMSE of the next best model, which is the IBCF model that has the tuning parameter k = 750. While this somewhat small difference in RMSE might seem insignificant, this is not the case. Running this particular IBCF model is very computationally intensive and takes significantly longer to execute than the LIBMF model. Furthermore, the predicted recommendations for the two randomly selected users under the LIBMF model align with these users' observed tastes and preferences. Thus, with regards to efficiency and accuracy, and with respect to this data set, the LIBMF model is unequivocally the best recommendation system out of the four explored in this report. In summary, for streaming platforms such as Netflix and Hulu, I would recommend implementing a recommendation system that is based on the LIBMF model constructed in this report.

Reference List

1. Hinkle, D. (2021) *How Streaming Services Use Algorithms*. Available at: https://amt-lab.org/blog/2021/8/algorithms-in-streaming-services (Accessed: 5 December 2021).

2. Michael Hahsler (2021). recommenderlab: Lab for Developing and Testing Recommender Algorithms. R package version 0.2-7. https://CRAN.R-project.org/package=recommenderlab

3. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

4. Matt Dowle and Arun Srinivasan (2021). data.table: Extension of `data.frame`. R package version 1.14.0. https://CRAN.R-project.org/package=data.table

5. Hadley Wickham (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. URL http://www.jstatsoft.org/v21/i12/.

6. Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2021). dplyr: A Grammar of Data Manipulation. R package version 1.0.7. https://CRAN.R-project.org/package=dplyr

7. Alboukadel Kassambara and Fabian Mundt (2020). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.7. https://CRAN.R-project.org/package=factoextra

8. *Recommender systems*. Available at: https://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/itembased.html (Accessed 5 December 2021).

9. Gupta, R. (2020) *User-Based Collaborative Filtering*. Available at: https://www.geeksforgeeks.org/user-based-collaborative-filtering/ (Accessed: 5 December 2021).

10. Tam, A. (2021) *Using Singular Value Decomposition to Build a Recommender System*. Available at: https://machinelearningmastery.com/using-singular-value-decomposition-to-build-a-recommender-system/ (Accessed: 5 December 2021).

11. Thandapani, S. (2019) *Recommendation Systems: Collaborative Filtering using Matrix Factorization — Simplified*. Available at: https://medium.com/sfu-cspmp/recommendation-systems-collaborative-filtering-using-matrix-factorization-simplified-2118f4ef2cd3 (Accessed: 5 December 2021).

12. Glen, S. (2021) *RMSE: Root Mean Square Error*. Available at: https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/ (Accessed: 5 December 2021).

Appendix 1

Item Based Collaborative Filtering (IBCF): A model-based algorithm that makes recommendations using the similarities found within different items in the data set. The similarities are calculated using any one of the various similarity measures. The resulting similarity values are used to estimate ratings for user-item pairs that are outside of the data set (Recommender systems).

User Based Collaborative Filtering (UBCF): A memory-based algorithm that makes recommendations for a given user (target user) based on the ratings given to said recommendations by other users that have similar tastes and preferences to that of the target user (Gupta, 2020).

Single Value Decomposition (SVD): Breaks down a given matrix into the product of a number of smaller matrices. This is less computationally-intensive than other methods and reveals similarities via eigenvectors and eigenvalues. This is a model-based algorithm (Tam, 2021).

Model Based Collaborative Filtering using Matrix Factorisation (LIBMF): A model-based algorithm that creates a matrix of users and a matrix of items and uses the dot product of these two matrices to create a ratings matrix. The tuning parameter for this model is the number of latent factors in the movie matrix. With respect to movies, the latent factors could be genre, length, actors, etc. (Thandapani, 2019).

Root-Mean-Squared-Error (RMSE): A model evaluation metric that calculates the difference between the actual values in the data set and the predicted values provided by the model (Glen, 2021).

Appendix 2

Figure 1:

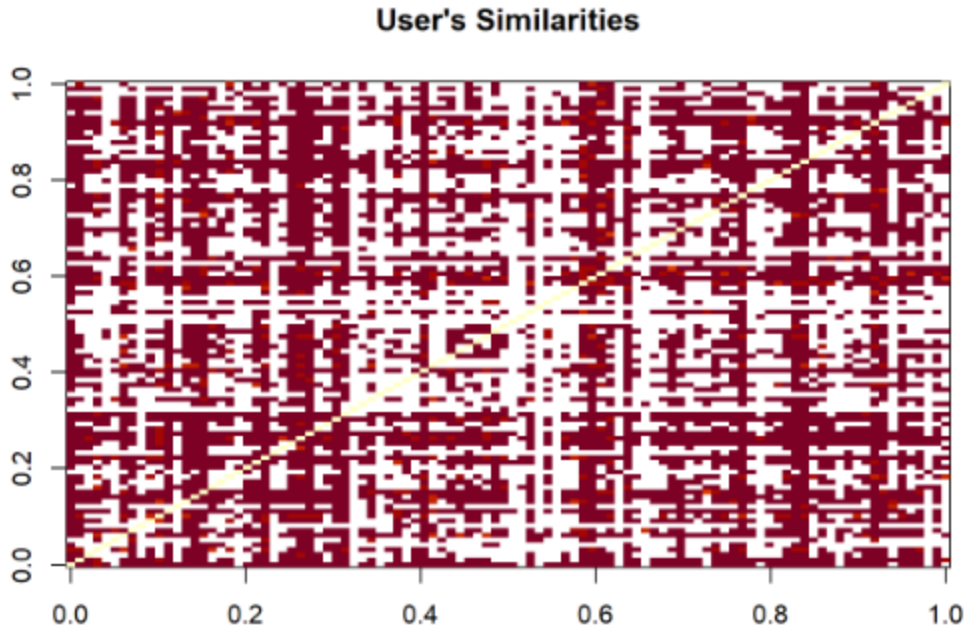

Figure 2:

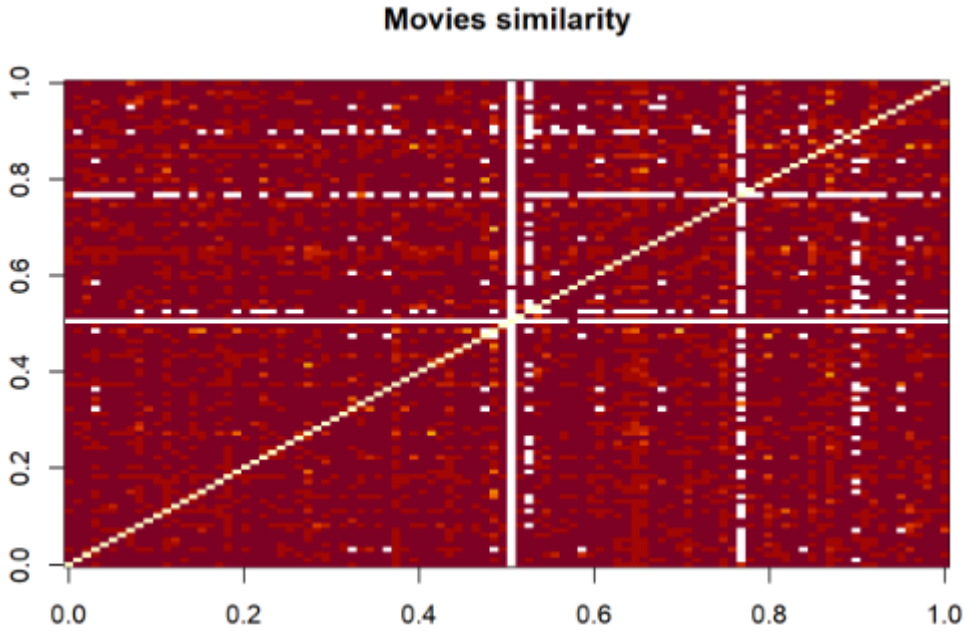

Figure 3:

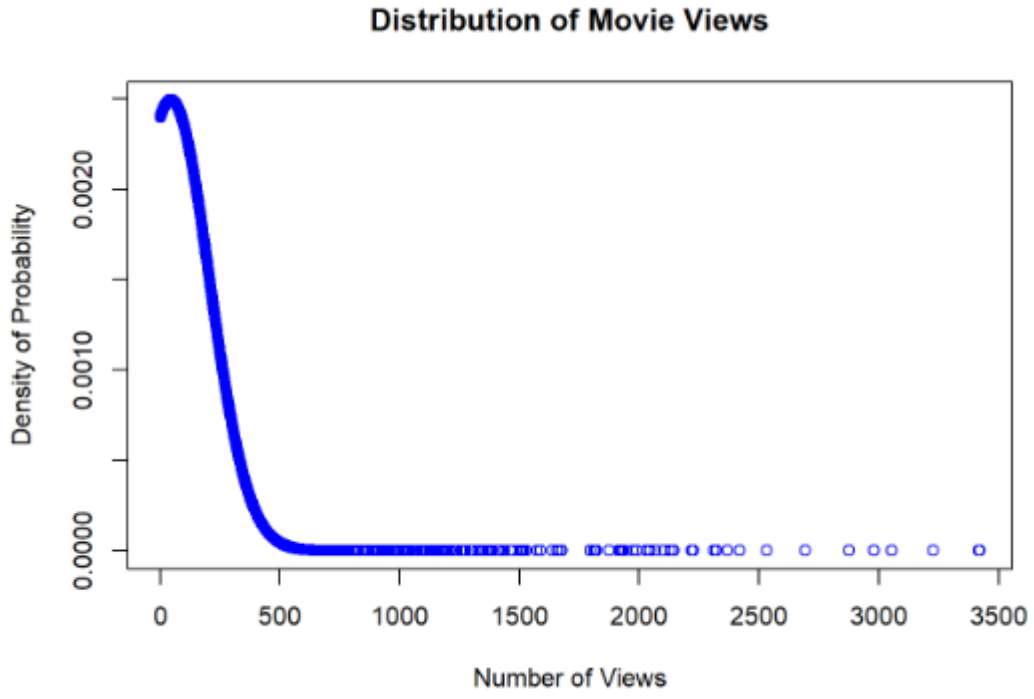

Figure 4:

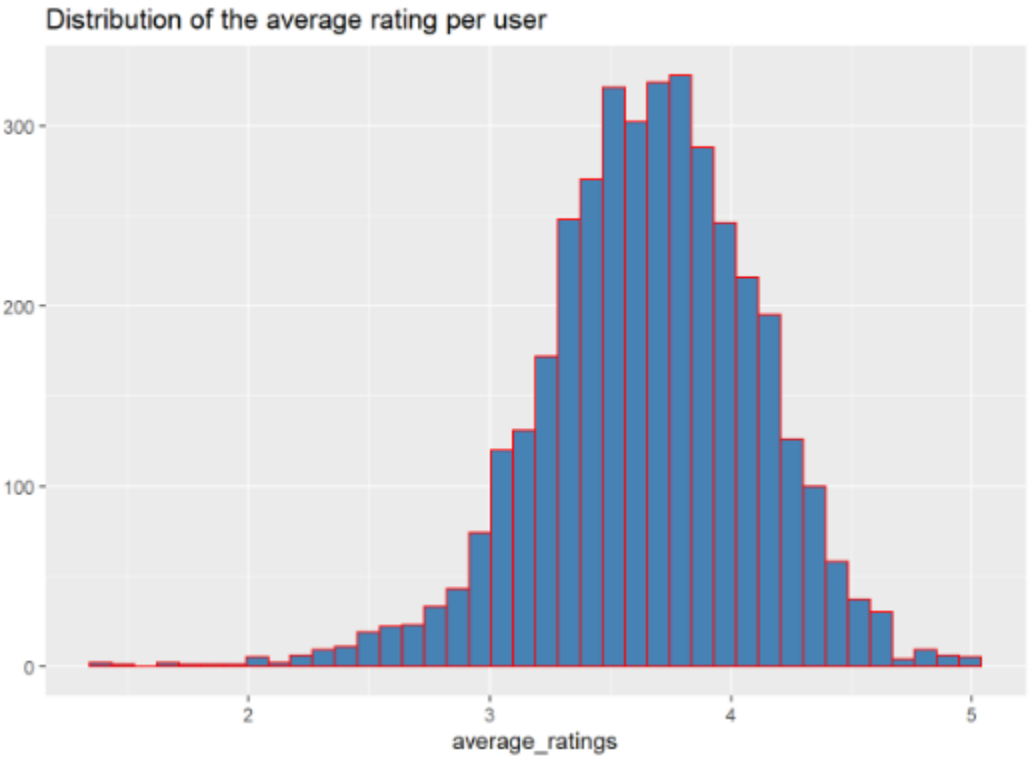

Table 1: UBCF Model

Model	RMSE
UBCF NN = 25	1.161937
UBCF NN = 30	1.152830
UBCF NN = 40	1.130616
UBCF NN = 50	1.116314

Table 2: IBCF Model

Model	RMSE
IBCF K = 150	0.9610675
IBCF K = 350	0.9070720
IBCF K = 550	0.8696669
IBCF K = 750	0.8576683

Table 3: SVD Model

Model	RMSE
SVD K = 25	0.9011910
SVD K = 50	0.8963923
SVD K = 100	0.8944230
SVD K = 150	0.8962056

Figure 5:

User 500 Predicted Recommendations

	Genre	Freq
1	Action|Adventure|Sci-Fi	2
8	Comedy	2
12	Comedy|Drama	2
2	Action|Adventure|Sci-Fi|Thriller	1
3	Action|Comedy	1
4	Action|Comedy|Crime	1
5	Adventure|Animation|Comedy	1
6	Adventure|Comedy	1
7	Animation|Comedy|Romance	1
9	Comedy|Crime|Mystery|Thriller	1
10	Comedy|Crime|Thriller	1
11	Comedy|Documentary	1
13	Comedy|Drama|Romance	1
14	Drama|Fantasy|Romance|Sci-Fi	1
15	Drama|Romance	1
16	Drama|War	1
17	Mystery|Romance|Thriller	1

Movies Viewed by User 500

	Genre	Freq
30	Drama	11
16	Comedy	10
17	Comedy|Drama	7
29	Documentary	3
33	Drama|Romance	3
40	Thriller	3
10	Adventure|Children	2
19	Comedy|Drama|Romance	2
23	Comedy|Thriller	2
32	Drama|Musical|Romance	2
1	Action	1
2	Action|Adventure|Sci-Fi	1
3	Action|Adventure|Sci-Fi|Thriller	1
4	Action|Comedy|Musical	1
5	Action|Crime|Drama	1
6	Action|Crime|Drama|Thriller	1
7	Action|Drama|Romance|War	1
8	Action|Drama|War	1
9	Action|Thriller	1
11	Adventure|Children|Comedy	1
12	Adventure|Children|Drama|Romance	1
13	Animation|Children|Fantasy|Musical	1
14	Animation|Children|Musical	1
15	Children|Comedy	1
18	Comedy|Drama|Film-Noir	1
20	Comedy|Fantasy	1
21	Comedy|Fantasy|Romance|Sci-Fi	1
22	Comedy|Romance	1
24	Crime	1
25	Crime|Drama	1
26	Crime|Drama|Fantasy|Thriller	1
27	Crime|Drama|Film-Noir|Thriller	1
28	Crime|Drama|Musical|Thriller	1
31	Drama|Horror|Thriller	1
34	Drama|Sci-Fi	1
35	Drama|Western	1
36	Fantasy|Horror	1
37	Film-Noir|Mystery|Thriller	1
38	Horror	1
39	Romance	1

Figure 6:

User 600 Predicted Recommendations

	Genre	Freq
14	Drama	3
1	Action|Adventure|Drama	1
2	Adventure|Comedy|Romance	1
3	Adventure|Documentary	1
4	Animation|Drama|War	1
5	Comedy	1
6	Comedy|Drama	1
7	Comedy|Drama|Romance	1
8	Comedy|Drama|War	1
9	Comedy|Romance	1
10	Comedy|War	1
11	Crime|Drama|Film-Noir|Thriller	1
12	Crime|Mystery|Thriller	1
13	Documentary	1
15	Drama|Mystery|Sci-Fi	1
16	Drama|Mystery|Thriller	1
17	Drama|War	1
18	Mystery|Thriller	1

Movies Viewed by User 600

	Genre	Freq
33	Drama	13
20	Comedy	8
40	Horror	8
26	Comedy|Romance	5
23	Comedy|Drama|Romance	4
36	Drama|Romance	4
22	Comedy|Drama	3
2	Action|Adventure|Drama|Thriller	2
3	Action|Adventure|Sci-Fi	2
17	Animation|Children	2
28	Crime|Drama|Romance|Thriller	2
29	Crime|Drama|Thriller	2
31	Documentary	2
34	Drama|Horror|Thriller	2
42	Horror|Sci-Fi	2
43	Horror|Thriller	2
44	Mystery|Thriller	2
45	Romance	2
1	Action|Adventure|Drama	1
4	Action|Children|Romance	1
5	Action|Comedy|Fantasy	1
6	Action|Horror|Sci-Fi	1
7	Action|Sci-Fi	1
8	Action|Sci-Fi|Thriller	1
9	Adventure	1
10	Adventure|Animation|Children|Comedy|Fantasy	1
11	Adventure|Children|Musical	1
12	Adventure|Comedy	1
13	Adventure|Comedy|Romance	1
14	Adventure|Comedy|Sci-Fi	1
15	Adventure|Comedy|Western	1
16	Adventure|Drama|War	1
18	Children|Comedy|Drama	1
19	Children|Comedy|Romance	1
21	Comedy|Crime|Thriller	1
24	Comedy|Drama|Romance|Thriller	1
25	Comedy|Drama|War	1
27	Crime|Drama|Mystery|Thriller	1
30	Crime|Thriller	1
32	Documentary|IMAX	1
35	Drama|Mystery|Romance|Thriller	1
37	Drama|Thriller	1
38	Drama|Thriller|War	1
39	Drama|War	1
41	Horror|Mystery|Sci-Fi|Thriller	1
46	War	1