Recommender Systems with R: Model for predicting movie ratings.

Dodema BITENIWE

23 septembre, 2024

Contents

1 Project overview

This project is part of the Data Science professional certification pathway offered by HarvardX on edx. It is the ninth and final course of a multi-part course. In this part, we demonstrate our mastery of the skills acquired during the course by building a machine learning model. The idea is to build a recommendation system on movie data. The system's aim is to suggest movies to users, taking into account their profiles, the movies they have already watched and rated, and similar profiles.

The data used in this project comes from the grouplens website, where we can find a variety of data for Data Science practice. The project data set is a compilation of over 10 million ratings for more than 10,000 movies by 72,000 users. This data is known as the MovieLens 10M Dataset.

In this report, we will start with an organization of the data, followed by an exploratory analysis of the data, then the construction of the model and presentation of the results, and finally the conclusion.

2 Data processing and organization

The data was downloaded from this site, then processed and organized according to the project guidelines, in this case following the code provided for this purpose. For analysis and modelling purposes, the data is divided into two parts. The first part, called edx, contains 90% of the original data and will be used to train and test the various algorithms.Table 1 gives an overview of the edx data.In total, the edx dataset includes 69878 users for 10677 movies.

Table 1: First lines of edx data.

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	1996-08-02 11:24:06	Boomerang (1992)	Comedy|Romance
2	1	185	5	1996-08-02 10:58:45	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	1996-08-02 10:57:01	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	1996-08-02 10:56:32	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	1996-08-02 10:56:32	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi

The second part, called final_holdout_test, will be used to validate the final model or algorithm by evaluating its performance through the RSME. We have ensured that the users and movies included in the validation data are also present in the edx data to avoid possible inconsistencies in predictions. Table 2 gives an overview of the validation data.In total, the validation dataset includes 68534 users for 9809 movies.

Table 2: First lines of validation data.

userId	movieId	rating	timestamp	title	genres
1	231	5	1996-08-02 10:56:32	Dumb & Dumber (1994)	Comedy
1	480	5	1996-08-02 11:00:53	Jurassic Park (1993)	Action|Adventure|Sci-Fi|Thriller
1	586	5	1996-08-02 11:07:48	Home Alone (1990)	Children|Comedy
2	151	3	1997-07-07 03:34:10	Rob Roy (1995)	Action|Drama|Romance|War
2	858	2	1997-07-07 03:20:45	Godfather, The (1972)	Crime|Drama

3 Exploratory data analysis (EDA)

In this section, we propose to extend our understanding of the data through an exploratory analysis. We'd like to mention two references (kaggle and Irizarry (n.d.)) that have inspired us in the following analysis.

3.1 Sparsity of MovieLens Ratings Matrix

Figure 1 shows the matrix for a random sample of 100 movies and 100 users, with yellow indicating a user/movie combination for which we have a rating. In this way, we can better appreciate how sparse the matrix is.

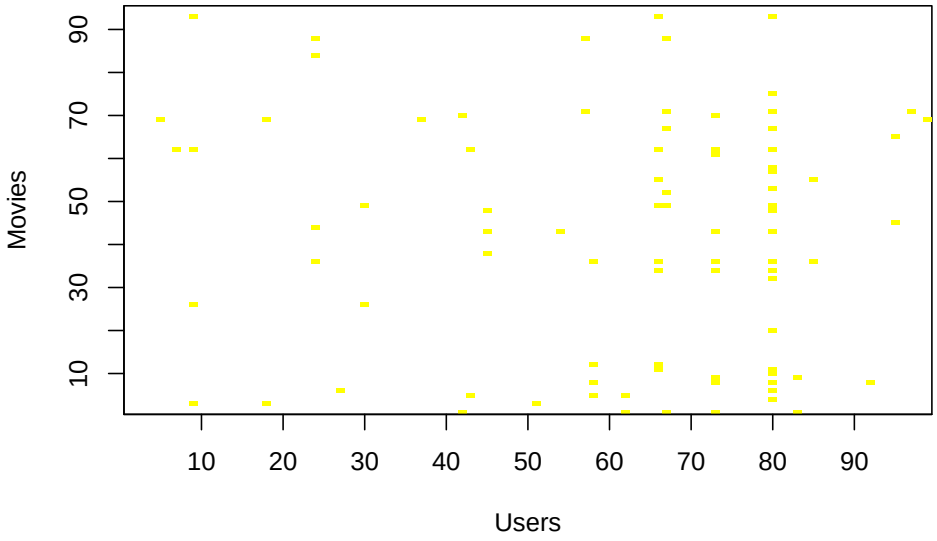

Figure 1: Sparsity of MovieLens Ratings Matrix

3.2 Rating distribution

Figure 2 shows that many users who rated the movies gave them 4 stars, followed by 3 stars, with 5 stars in third place. There are less half-star ratings than full-star ratings

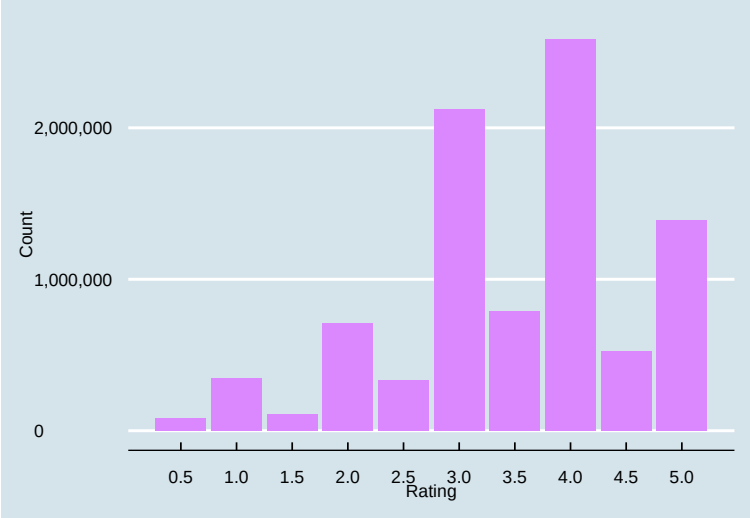

Figure 2: Rating distribution

3.3 Rating per movies and movies' rating histogram

Figure 3 shows that some movies are rated and watched by tens of thousands of users (blockbusters), while others (more numerous) are rated hundreds of times or even less.

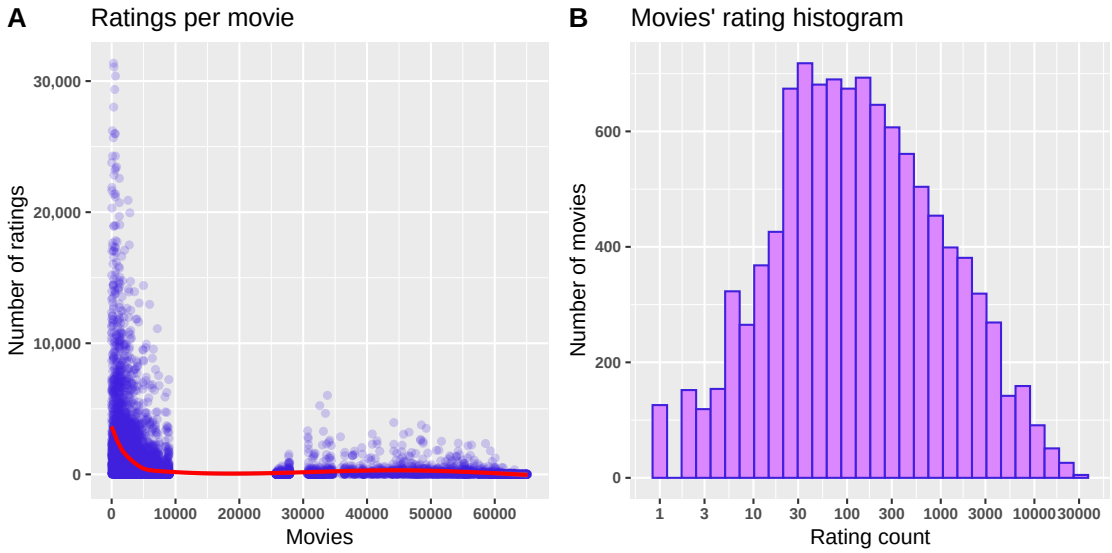

Figure 3: Rating per movies and movies' rating histogram

3.4 Rating per users and users' rating histogram

As Figure 4 shows, some users are very active, rating thousands of movies, while others are less active, rating just a few dozen. We can also see that, on average, users rated very few movies.

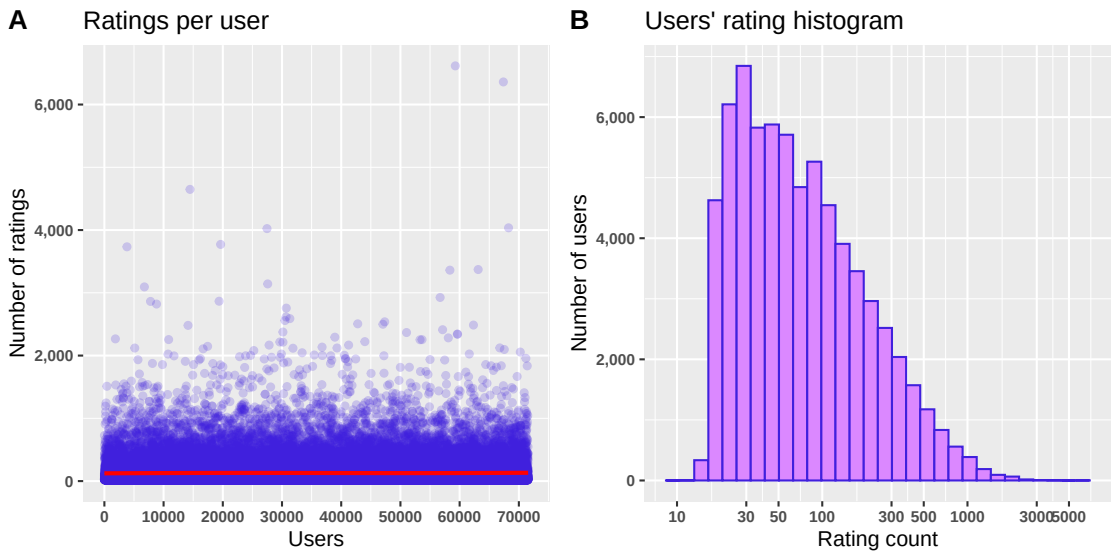

Figure 4: Rating per users and users' rating histogram

4 Analysis

In this section, we intend to train different recommendation models or algorithms and evaluate their performance using well-known statistical regression metrics: mean absolute error and root mean square error. The best performing algorithm will be used for the final prediction.

Mathematically, mean absolute error (MAE) is defined by :

$$MAE = \frac{1}{N} \sum_{u,i} \|y_{u,i} - \hat{y}_{u,i}\| \tag{1}$$

and root mean square error (RMSE) by:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i} (y_{u,i} - \hat{y}_{u,i})^2} \tag{2}$$

We define $y_{u,i}$ as the rating for movie i by user u and denote our prediction with $\hat{y}_{u,i}$.

4.1 Modeling approach A: Popular, random, svd algorithms with recommenderlab

In this first approach, we'll use the recommenderlab package (Hahsler 2011) to implement 3 algorithms:

- RANDOM : Randomly chosen movies. This creates random recommendations which can be used as a baseline for recommender algorithm evaluation.
- POPULAR : Popular movies. This is a non-personalized algorithm which recommends to all users the most popular movies they have not rated yet.
- SVD : This is a latent factor model using singular value decomposition (SVD) to estimate missing ratings

Table 3 compares the performance of the three algorithms and shows that SVD gives better results in terms of RMSE and MAE.

Table 3: Algorithms performance.

Model	MAE	MSE	RMSE
RANDOM	1.5002057	3.3784723	1.8380621
POPULAR	0.6748350	0.7613361	0.8725458
SVD	0.6393417	0.6922737	0.8320299

4.2 Modeling approach B: Linear Regression Model

In order to have a regression model with good performance while avoiding overtraining, and following the procedure in (Irizarry, n.d.), we will make use of a regularization parameter λ in the model. This parameter will be estimated by cross-validation. The minimization equation of the model takes the following mathematical form :

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i}-\mu-b_i-b_u\right)^2+\lambda\left(\sum_i b_i^2+\sum_u b_u^2\right) \tag{3}$$

with μ the average of all ratings, b_i the movie-specific effect, b_u the user-specific effect.

Figure 5 shows the performance of the model for different λ values, allowing us to pick the optimum value.

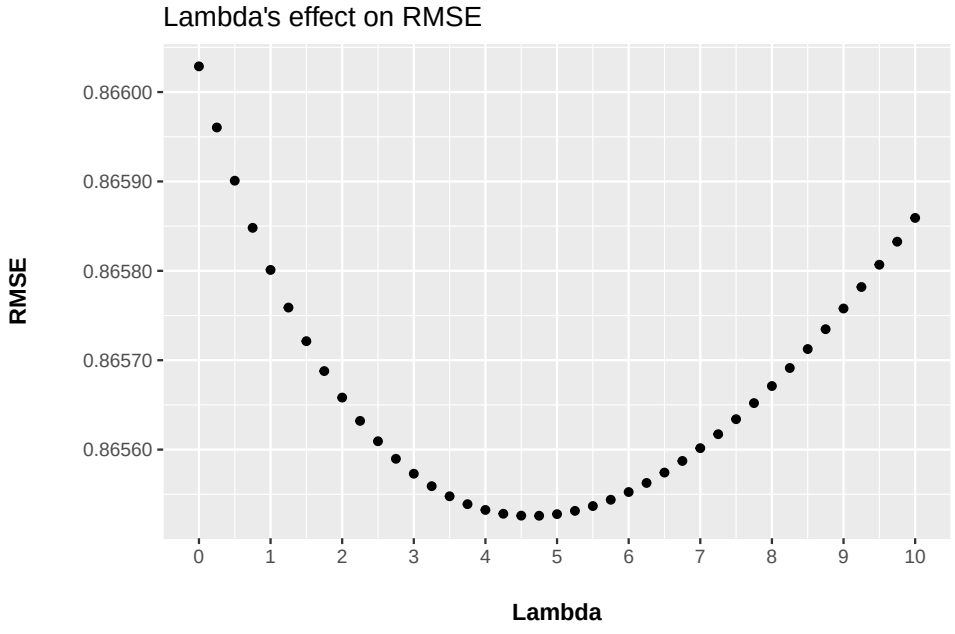

Figure 5: RMSE values as a function of lambda

The optimal λ is: 4.75

It's worth noting that here the lambda value is only optimized on 90% of the edx data, which is the training data for the model. The lambda value found is used to evaluate the model on test data made up of 10% of the edx data. Model performance is reported in Table 4. We can see that the model performs significantly better than the RANDOM and POPULAR algorithms, but not as well as the SVD algorithm.

Table 4: Algorithms performance.

Model	MAE	MSE	RMSE
RANDOM	1.5002057	3.3784723	1.8380621
POPULAR	0.6748350	0.7613361	0.8725458
SVD	0.6393417	0.6922737	0.8320299
Linear Regression Model	0.6699801	0.7491352	0.8655260

4.3 Modeling approach C: Matrix factorization

As described in (Chin et al. 2016) Matrix Factorization (MF) is a process to find two factor matrices, $P \in \mathbb{R}^{k\times m} and Q \in \mathbb{R}^{k\times m}$, to describe a given m-by-n training matrix R in which some entries may be missing. For rating prediction, the entry value $r_{u,i} \in \mathbb{R}$ indicates that the ith item was rated $r_{u,i}$ by the uth user. Once P and Q are learned, a missing rating at the (u', i') entry can be predicted by the inner product of the u'th column of P (i.e., $p_{u'}$) and the i'th column of Q (i.e., $q_{i'}$). It means that we can generate the predicted scores on all movies for a user, and then the one with highest score may be recommended. MF can be formulated as a non-convex optimization problem :

$$\min_{P,Q} \sum_{(u,i)} \left[f(p_u, q_i, r_{u,i}) + \mu_p \|p_u\|_1 + \mu_q \|q_i\|_1 + \frac{\lambda_p}{2}\|p_u\|_2^2 + \frac{\lambda_q}{2}\|q_i\|_2^2 \right] \tag{4}$$

where $r_{u,i}$ is the (u, i) entry of R, $f(.)$ is a non-convex loss function of p_u and q_i, and μ_p, μ_q, λ_p, and λ_q are regularization coefficients.

The results shown in Table 5 demonstrate the high performance of the matrix factorization model.

Table 5: Algorithms performance on test data.

Model	MAE	MSE	RMSE
RANDOM	1.5002057	3.3784723	1.8380621
POPULAR	0.6748350	0.7613361	0.8725458
SVD	0.6393417	0.6922737	0.8320299
Linear Regression Model	0.6699801	0.7491352	0.8655260
Matrix factorization	0.6050095	0.6180481	0.7861603

4.4 Choosing the best-performing model.

A comparative analysis of the results shown in Table 5 leads to the conclusion that the matrix factorization model performs best in terms of RMSE. This model is then chosen as the final model.

4.5 Predictions on the final holdout data

The final model is re-trained on all edx dataset, then evaluated on the validation dataset. The results show an improvement in model performance. This is due to the larger size of the data.

5 Results

Our analysis shows that the matrix factorization algorithm is the best performing of the 5 algorithms trained on edx data. Performance in terms of RMSE on the validation data is **0.7809783**. That said, we also note the good performance of the POPULAR, SVD and linear regression algorithms with a regularization parameter, which are all below 0.87.

The top ten (10) movies recommended by the final model are shown in Table 6 with the average rating and the count of times each movie is recommended.

Table 6: Top 10 movie recommendation by the final model.

Title	Rating	Count
Schindler's List (1993)	4.363390	2584
Lord of the Rings: The Return of the King, The (2003)	4.155627	1253
Pulp Fiction (1994)	4.181039	3502
Shawshank Redemption, The (1994)	4.476213	3111
Matrix, The (1999)	4.228134	2321
Rhyme & Reason (1997)	4.000000	2
Happy Together (1989)	2.875000	4
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	4.210435	2894
Fight Club (1999)	4.191948	1602
Star Wars: Episode V - The Empire Strikes Back (1980)	4.207028	2362

6 Conclusion

In this study, we explored the Movielens-10M data. First, we used various visualizations to understand the data. Then 5 algorithms were chosen and trained on part of the data. The analysis revealed that the matrix factorization model is the best model, with an RMSE performance of **0.7809783**. This model will therefore be ideal for our movie recommendation system.

References

Chin, Wei-Sheng, Bo-Wen Yuan, Meng-Yuan Yang, Yong Zhuang, Yu-Chin Juan, and Chih-Jen Lin. 2016. "LIBMF: A Library for Parallel Matrix Factorization in Shared-Memory Systems." *Journal of Machine Learning Research* 17 (86): 1–5. http://jmlr.org/papers/v17/15-471.html.

Hahsler, Michael. 2011. "Recommenderlab : A Framework for Developing and Testing Recommendation Algorithms." In. https://api.semanticscholar.org/CorpusID:9405286.

Irizarry, Rafael A. n.d. *Introduction to Data Science*. https://rafalab.github.io/dsbook/: HarvardX.