Movielens Capstone Project

Ismael Leal

2023-01-13

1. Introduction

Recommendation systems are a vital aspect of multimedia providers. Netflix, HBO, and all similar platforms have a "recommended for you" section personalized for every user. In general, algorithms that recommend specific content to specific users are in great demand, now more than ever. The purpose of this project is to create and assess the accuracy of various models used to recommend movies to users.

The dataset used is a Movielens version with around 10 million movie ratings, including ratings for more than 10,000 movies by over 72,000 users. For the creation and testing of the algorithms, two different subsets were created:

- "Edx": contains a 90% of the original dataset
- A final test set: 10% of the original dataset. This will only be used for testing the algorithms, never for their training. For that means, it was created only containing ratings by users and for movies already present in the train set.

Different models will be created, each taking into account more variables.

2. Analysis

The train set must be analysed before the algorithm creation process. It looks like Table 1. It's been assigned the name "edx" and is a data frame with 6 features or columns, of the type displayed in Table 2.

Table 1: First 6 rows

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
2	1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
7	1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

Table 2: Feature types

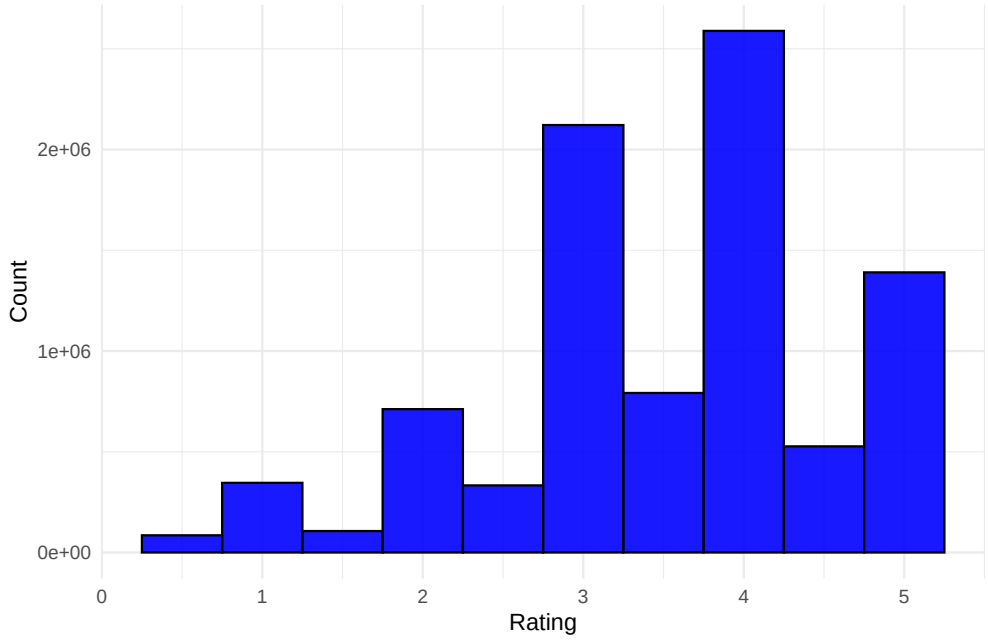
userId	movieId	rating	timestamp	title	genres
integer	integer	numeric	integer	character	character

Table 3: Users, movies, dates

Unique users	Unique Movies	First rating	Last rating
69878	10677	1995-01-09	2009-01-05

From the original ~10 million ratings, our train set consists of 9000055 ratings. Table 3 shows the number of users that participated in those ratings, the total number of movies that were rated, and the dates of the oldest and newest ratings.

Fig. 1 − Frequency of every rating



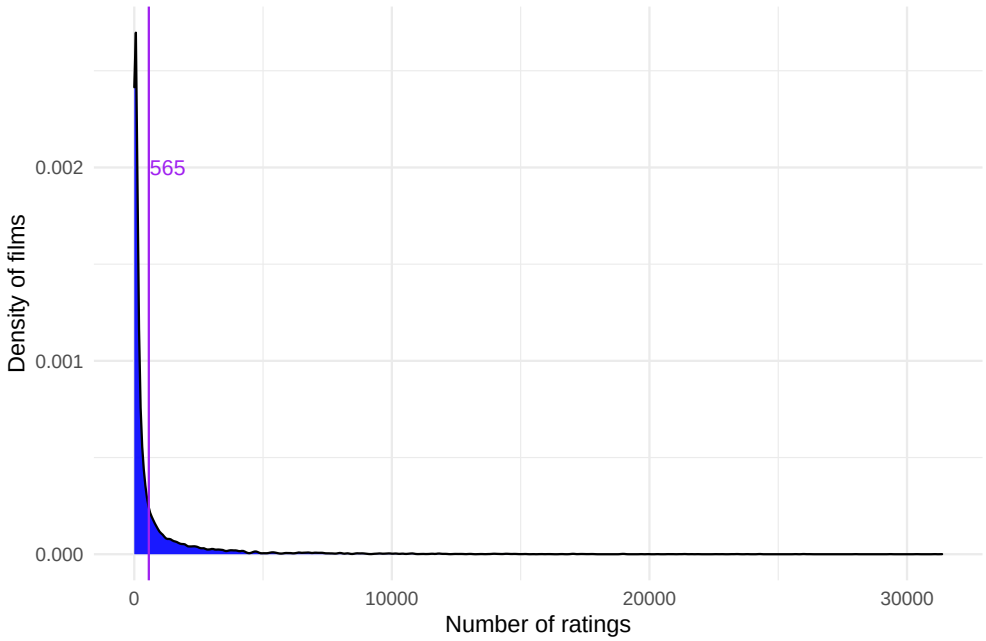
From the histogram in Figure 1 we know that the ratings range from 0.5 to 5, and they can only be integers or half-integers. Integer ratings (i.e. 3, 5) are much more usual than half-integer ratings (i.e. 2.5, 0.5), and most of the ratings are above average. It seems that users are more likely to rate a movie if they have enjoyed it. Let's explore whether there's a clear movie effect or not.

Movie effect

A movie density plot will show how many ratings has each movie received.

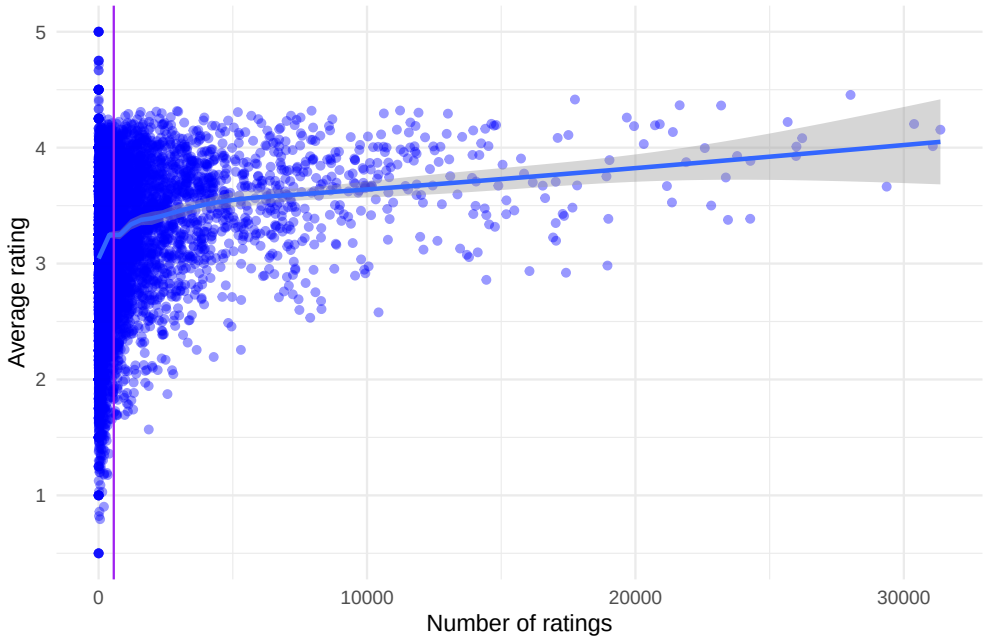
```
## [1] 565
```

Fig. 2 − Density plot of the movies



The density plot of Figure 2 shows a great variation in the number of ratings that films got: it ranges from 1 to 31362. A vertical purple line has been placed in the 3rd quartile of the movies. Thus, a 75% of the movies got less than 565 ratings. However, all the ratings from this 75% of movies just constitute a 10.59% of the total ratings. But will movies with fewer ratings be harder to predict?

Fig. 3 − Scatter plot of movie's average rating vs number of ratings



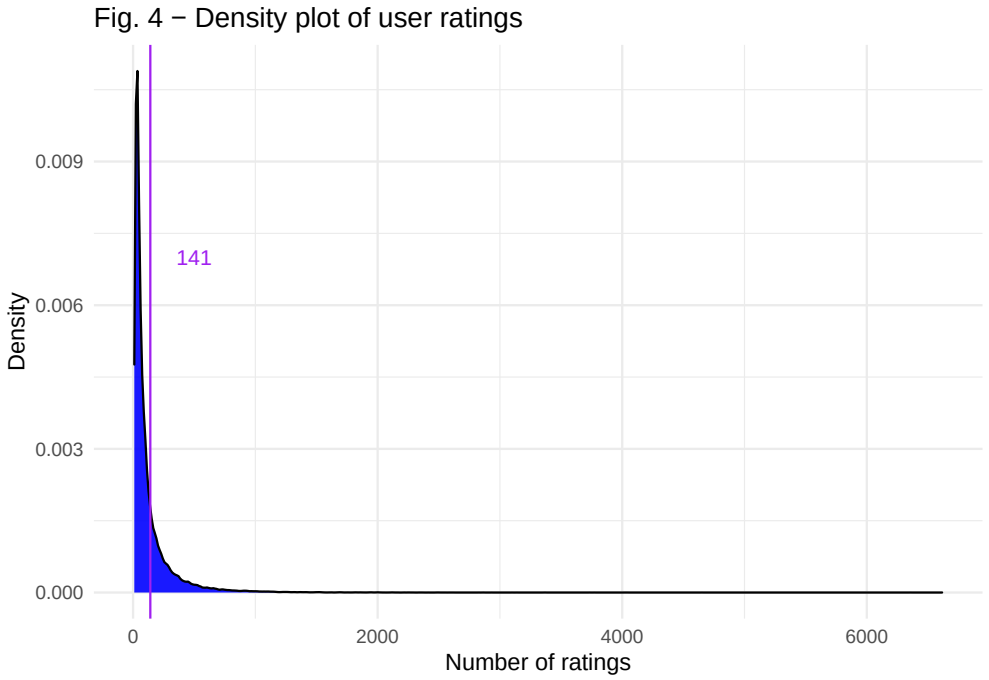
As Figure 3 shows, movies with fewer number of ratings (located at the left side) can have any average rating, ranging from 0.5 to 5, while movies that have been rated many times have a much more clear average, ranging

from near 3 to 4.5. So the more a movie has been rated, the more definite its average rating is. Thus, the predicted ratings will be affected by which movie is being rated and by how many times it has been rated.

User effect

Let's now study another variable: the effect of every user, and how this affects the predicted rating.

```
## [1] 141
```



So again, Figure 4 shows that there are some users with thousands of ratings but a majority of them have less than 1000 ratings. A line representing the 3rd quartile of the users has been drawn: a 75% of the users have submitted less than 141 ratings. This 75% of users only constitute a 31.8% of the total ratings however.

Fig. 5 − Scatter plot of user's average rating vs number of ratings

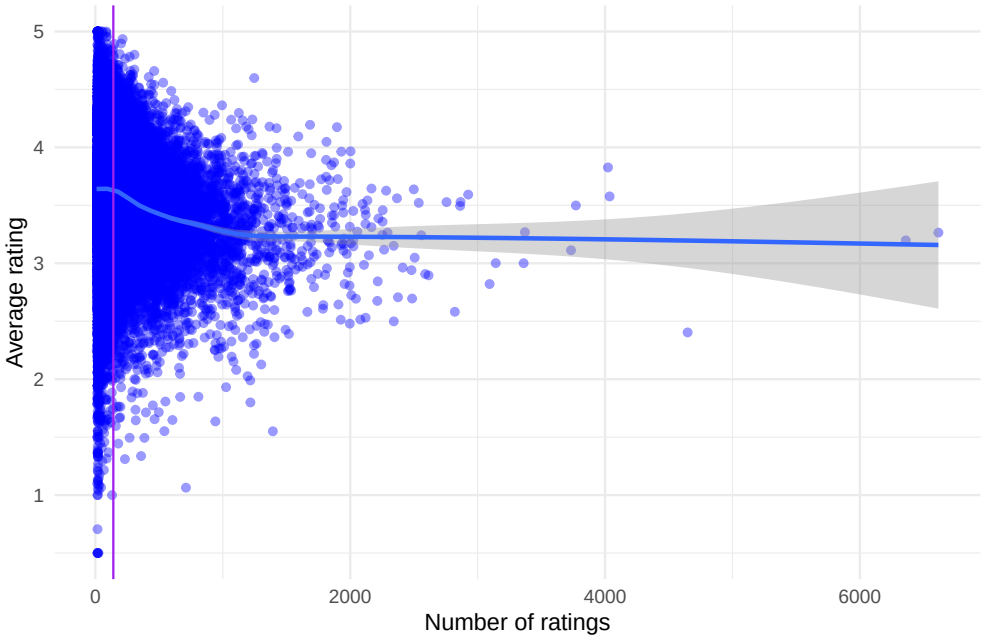


Figure 5 shows an effect similar to the movies with the users: the more ratings a user has submitted, the easier it will be to predict the rating. Users with less than 200 ratings have average ratings ranging from 0.5 to 5, while users with more than 2000 ratings rate movies with an average between 2.5 and 4.

So it can be inferred that there is also a user effect that affects the ratings.

Genre effect

The next variable to study is the genre. As it can be seen in Table 1, the "genres" feature of the train set shows a combination of genres for each film (e.g. Action|Drama). This gives a total of 797 distinct genre combinations. Does the rating change for each of these genre combinations?

Fig. 6 − Column graph of average rating for every genre combination

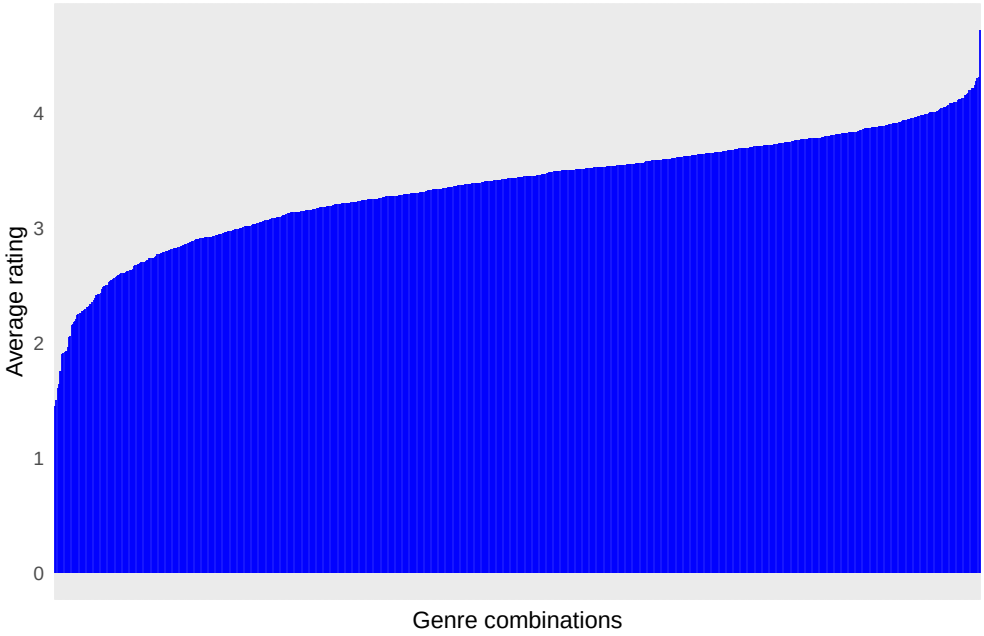
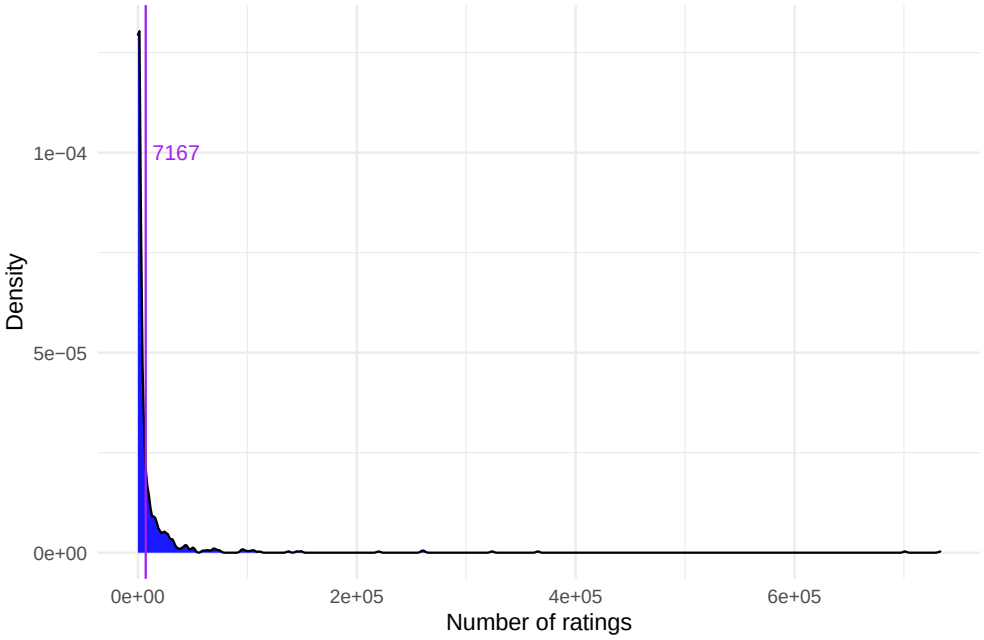


Figure 6 shows a great variability in the average rating given to each of the genre combinations. The lowest ratings are just below 1.5 while the highest are close to 5. Therefore, there is also a genre effect that has to be taken into consideration.

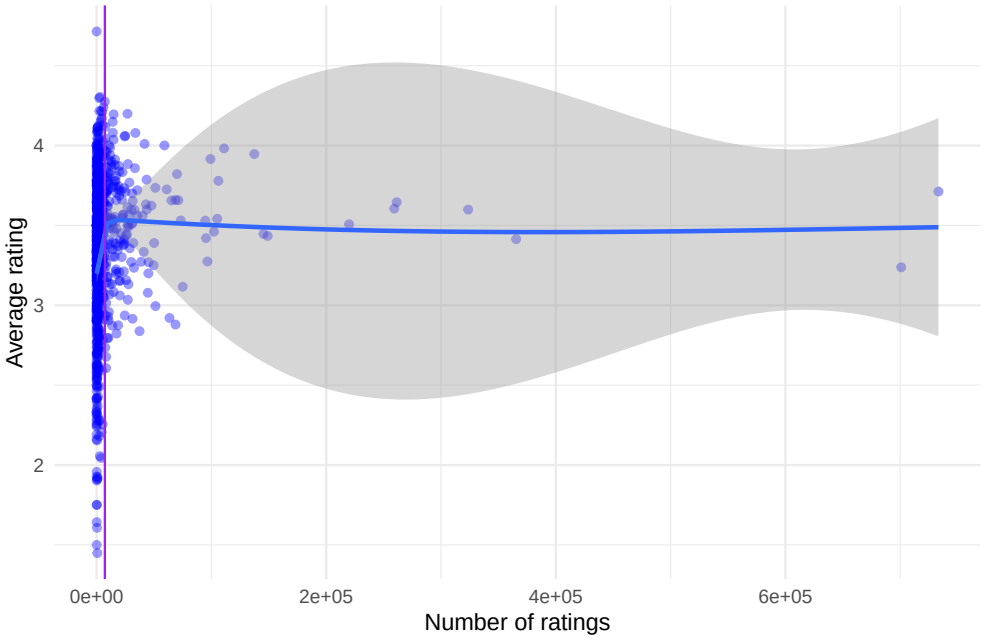
```
## [1] 7167
```

Fig. 7 − Density plot of genre combinations



However, as with the movies and the users, Figure 7 shows that most of the genre combinations have received few ratings, while just a 25% of them have received more than 7167.

Fig. 8 − Average rating versus number of ratings for each genre



And as with the variables already studied, the genres with less number of ratings show a much greater variability than genres with a large number of ratings.

Time effect

Before going into the model-creation, let's study the effect of time on the ratings.

Fig. 9 − Average rating per release year

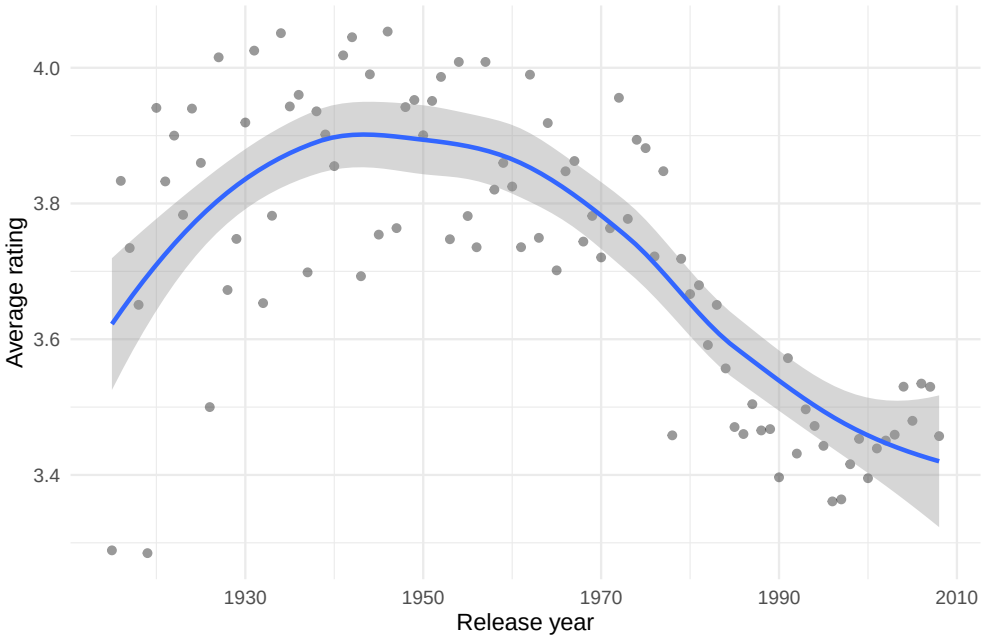
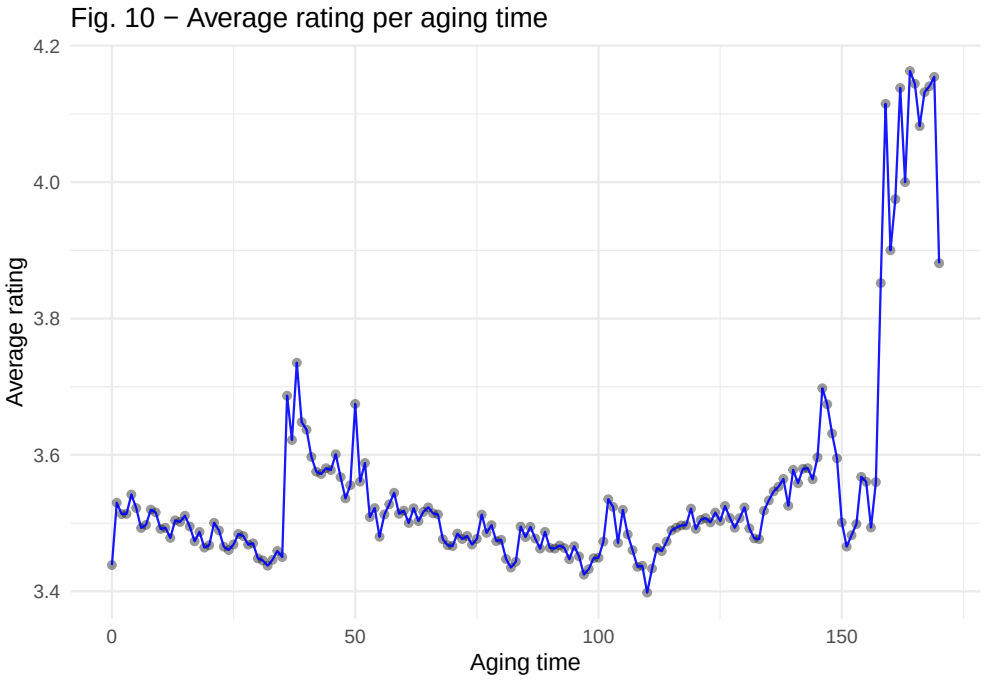


Figure 9 shows how average rating decreases for movies released around 1945 or later. It seems that newer movies are rated lower on average. Maybe, however, it's a movie's aging time what affects its rating more than its release year.



Between a month and two months of aging time, the average rating of movies increases, but then keeps decreasing until almost 10 years later, where it blows up to almost 4.2 of rating average.

RMSE

We have considered movie, user, genre, and time effects on the ratings to be predicted. Before creating or evaluating any algorithm, let's define a function that calculates the Root Mean Squared Error (RMSE) of our predictions. The quality of algorithms will be set out by this RMSE.

```
#This function calculates the RMSE
RMSE <- function(actual_ratings, predictions){
  sqrt(mean((actual_ratings - predictions)^2))
}
```

3. Algorithms

In order to evaluate the RMSE of every model, the "edx" set will be split into train and test sets.

Average

The most simple algorithm that can be thought of is one that just predicts the average rating for every movie and user:

$\hat{Y}_{u,i} = \mu + \epsilon_{u,i},$

where μ is the average rating and has a value of 3.5124556, and $\epsilon_{u,i}$ accounts for the individual error for the user u and the movie i.

Of course, a high RMSE is expected from this model, given that it does no prediction apart from the overall average rating. Indeed, the resultant RMSE is 1.0600537, and we should hope for less. Let's create a table to save the RMSEs.

Method	RMSE
Average	1.060054

Movie effect

We now consider a model that implements the clear effect that every movie has on its rating, seen in Figures 2 & 3. The new model's predictions look like this:

$\hat{Y}_{u,i} = \mu + b_i + \epsilon_{u,i}$

where the new term b_i accounts for the effect of movie i. Thus the ratings of the train set $Y_{u,i}$ can be used to find the average b_i of each film like

$b_i = \overline{Y_{u,i} - \mu} = \frac{1}{N} \sum_{j=1}^{N} (Y_{j,i} - \mu)$

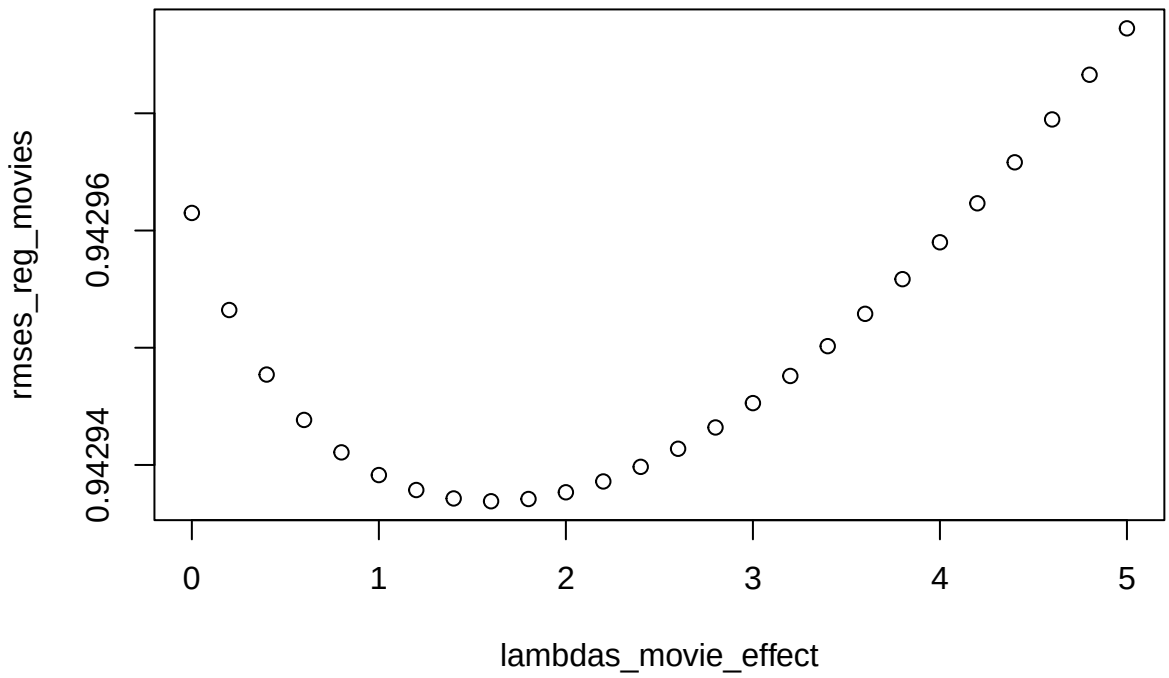
where N is the total number of ratings for the movie i.

Method	RMSE
Average	1.0600537
Movie Effect	0.9429615

However, as seen in Figure 3, the movie effect is not as clear for movies with less number of ratings. Hence, regularization could help alleviate this effect to find a lower RMSE. For this new regularized model, the b_i now look different:

$b_i = \sum_{j=1}^{N} \frac{Y_{j,i} - \mu}{N + \lambda_1}$

where the regularization parameter λ_1 has been included to lower the b_i of movies with fewer ratings.



The best regularization parameter when tried on the test set turns out to be 1.6

Method	RMSE
Average	1.0600537
Movie Effect	0.9429615
Regularized Movie Effect	0.9429369

User effect

A more complete algorithm would take into account the user effect seen in Figures 4 and 5. The predictions $\hat{Y}_{u,i}$ could be modeled as:

$\hat{Y}_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$

where the b_i already have incorporated regularization. The train set ratings $Y_{u,i}$ can be used to find the b_u for each user as:

$b_u = \overline{Y_{u,i} - \mu - b_i}$

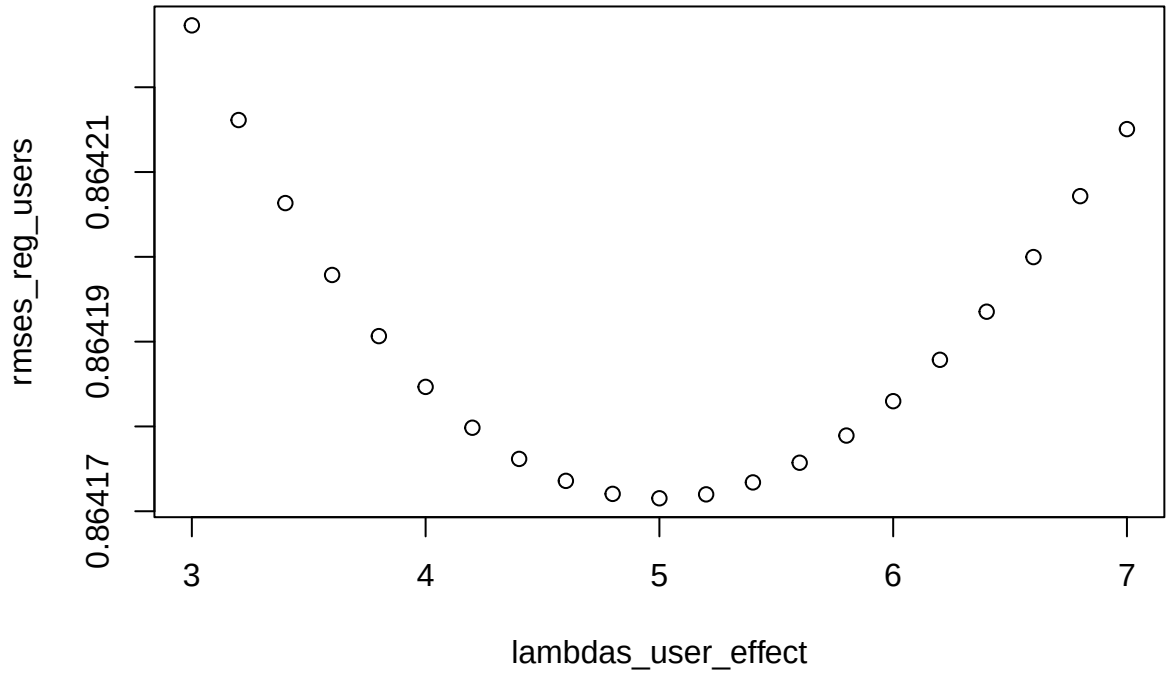
Method	RMSE
Average	1.0600537
Movie Effect	0.9429615
Regularized Movie Effect	0.9429369
User Effect	0.8646032

This model lowers the RMSE in the test set created from the "edx" dataset. However, figure 5 suggests that regularization might be a good idea for the user effect too.

The new b_u would look like:

$b_u = \sum_{j=1}^{N} \frac{Y_{u,j} - \mu - b_j}{N + \lambda_2}$

where the regularization parameter λ_2 has been introduced so that the b_u of users with low number of ratings gets lowered.



Method	RMSE
Average	1.0600537
Movie Effect	0.9429615
Regularized Movie Effect	0.9429369
User Effect	0.8646032
Regularized User Effect	0.8641715

The best regularization parameter λ_2 turns out to be 5. Again, regularization reduced the RMSE, though not too much.

Genre effect

Let's implement the genre effect discovered in Section 2 to the algorithm. Our new model's predictions would look like:

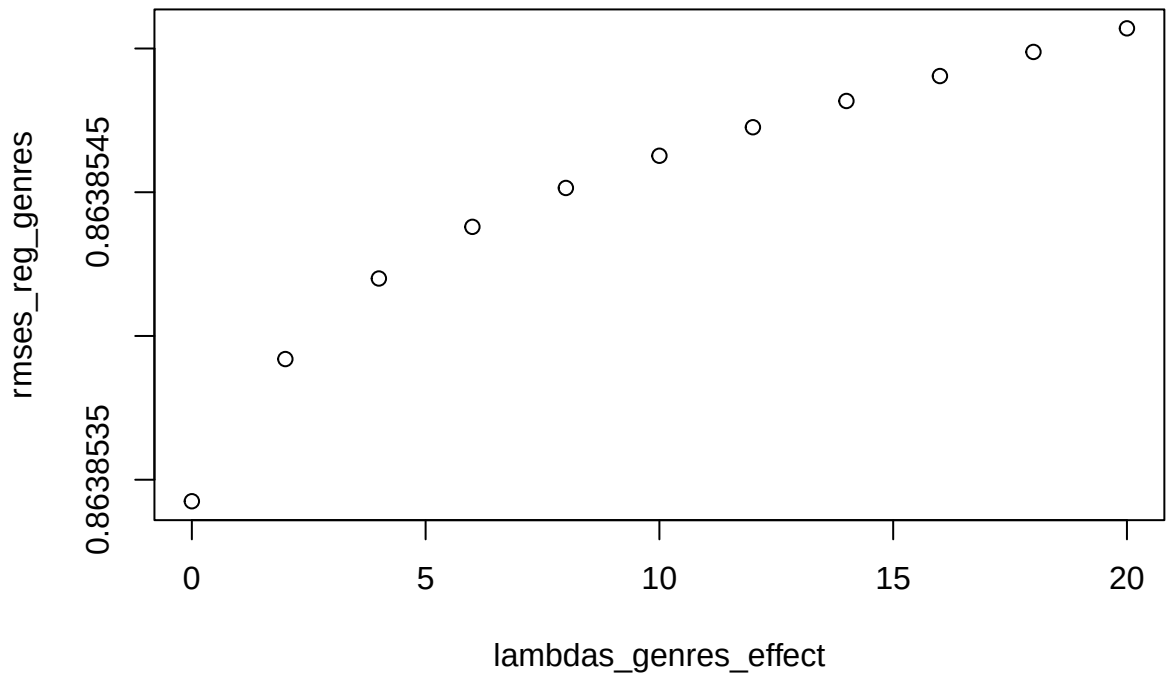
$\hat{Y}_{u,i} = \mu + b_i + b_u + b_g + \epsilon_{u,i}$

where

$b_g = \overline{Y_{u,i} - \mu - b_i - b_u}$

and $Y_{u,i}$ are the ratings from the train set.

Finally, the genre effect will also be regularized, as Figure 8 suggests.



Method	RMSE
Average	1.0600537
Movie Effect	0.9429615
Regularized Movie Effect	0.9429369
User Effect	0.8646032
Regularized User Effect	0.8641715
Genre Effect	0.8638534
Regularized Genre Effect	0.8638534

However, the best regularization parameter λ_3 is equal to 0, so no regularization will take place. Thus, the Regularized Genres Effect model will be taken out of the RMSEs table.

Time effect

The last variable to add to this model is time. The predictions would look like:

$\hat{Y}_{u,i} = \mu + b_i + b_u + b_g + b_t + \epsilon_{u,i}$

where the train set ratings $Y_{u,i}$ can be used to get the b_t:

$b_t = \overline{Y_{u,i} - \mu - b_i - b_u - b_g}$

Method	RMSE
Average	1.0600537
Movie Effect	0.9429615
Regularized Movie Effect	0.9429369
User Effect	0.8646032
Regularized User Effect	0.8641715
Genre Effect	0.8638534
Time Effect	0.8634294

4. Results

Now that the final model has been developed, it can be tested in the original test set, not the one created later for algorithm development. First, the final test set has to be edited so that it has columns for the b_i, b_u, b_g, and b_t. Once this is done, the final prediction can be obtained, with its associated RMSE.

The algorithm developed allowed predictions for the ratings to be made, lowering the RMSE to 0.864474.

5. Conclusion

The predictors used in this algorithm are the average rating, the specific effect by each movie, user, genre, and by time. The predictors with a highest impact in the RMSE are the movie and the user effects. Regularizing the movie and user effects also helped improving our algorithm. This algorithm will probably fail for a movie with few ratings, or for a user that has given few ratings. However, overall it achieves decent results.