MovieLens Project

(Data Science: Capstone" (PH125.9x) by edX and HarvardX)

Jülide Güzin Karagöz

Dec 2022

Table of Contents

1. Introduction

2. Analysis

3. Result

4. Conclusion

Introduction Section

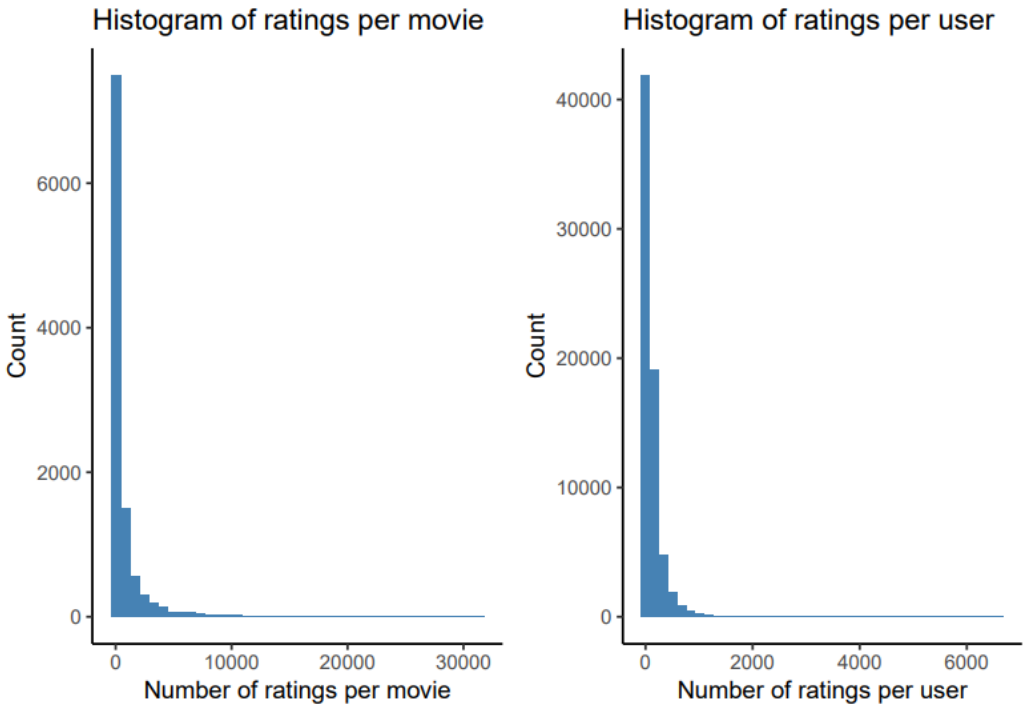
The MovieLens dataset is a database with over 10 million ratings for over 10,000 movies by more than 72,000 users. The dataset includes the identification of the user and the movie, as well as the rating, genre, and the timestamp. No demographic information is included.

The goal of this project is to predict movie ratings. To do that, the dataset was subsetted into two: the train and validation set. The validation set is 10% of the original data and is not used in the construction of the model.

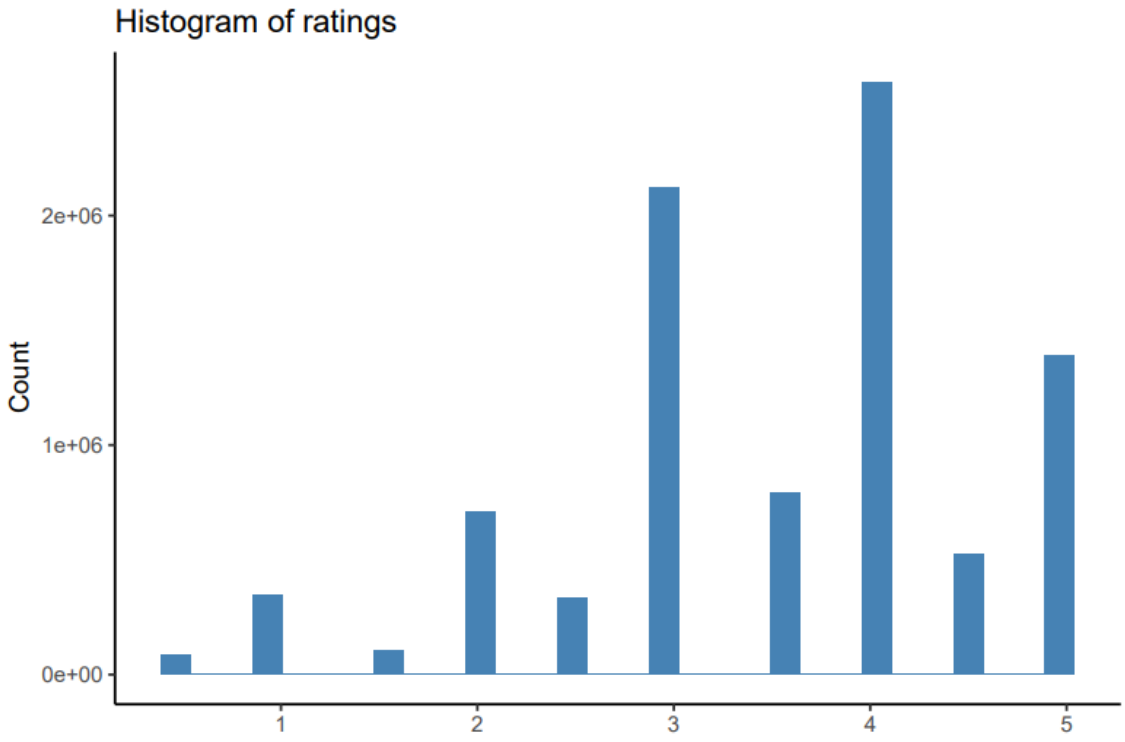
Due to the large size of the dataset, usual data wrangling (for example, the *lm* model) was not possible because of memory allocation. To solve this problem we computed the least square estimates manually. As the dataset is very sparse, we also included regularization in the model.

In total, 69878 unique users provided ratings and 10677 unique movies were rated. If we think about all the possible combinations between users and movies, we would have more than 746 million combinations. Our test set has a little over 9 million rows, which implies that not every user has rated every movie. This number of ratings is only 1.21% of all possible combinations, which designates a sparse matrix.

In addition to not having every movie rated by every user, some movies were rated more than others and some users have rated more than others, as shown in the two histograms below.



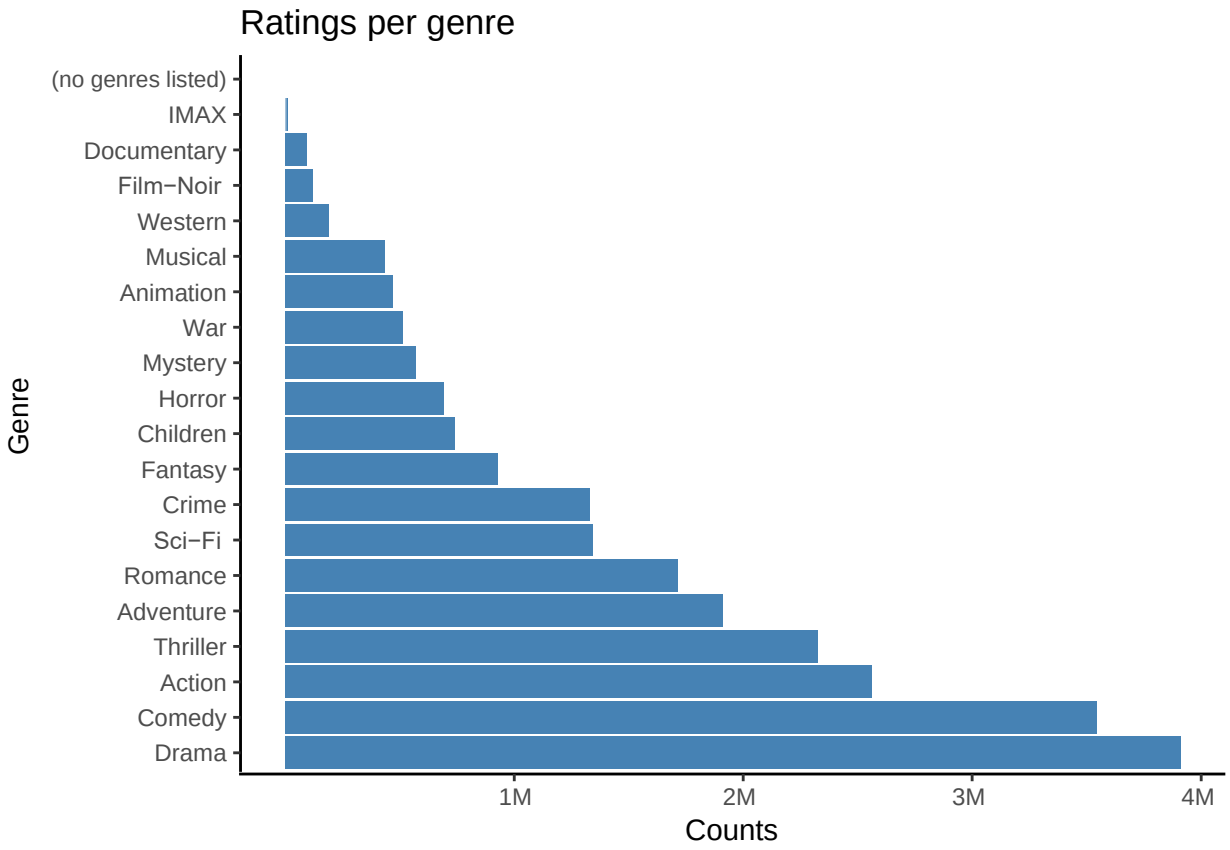
From the histogram of ratings below, we can observe integer ratings were more frequent than half-integers and that the ratings distribution is left-skewed.

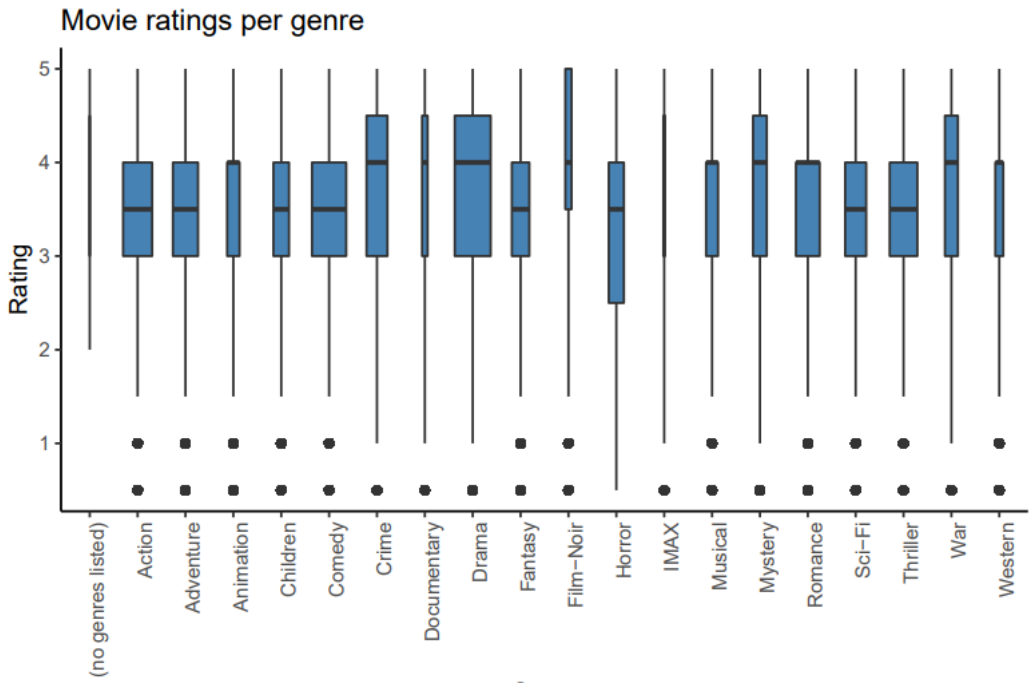


The genre variable contains all the genres the movie is characterized in, within twenty different classifications:

```
##  [1] "Comedy"      "Romance"       "Action"
##  [4] "Crime"       "Thriller"      "Drama"
##  [7] "Sci-Fi"      "Adventure"     "Children"
## [10] "Fantasy"     "War"           "Animation"
## [13] "Musical"     "Western"       "Mystery"
## [16] "Film-Noir"   "Horror"        "Documentary"
## [19] "IMAX"        "(no genres listed)"
```

In respect to genres, we can observe that some genres have a lot more ratings than others and that the ratings appear to be different between genres. The most popular rated genre types are Drama and Comedy. Drama and film-noir are some of the better-rated genre types, while horror is the worst rated.


Ratings per genre
(no genres listed)
IMAX
Documentary
Film−Noir
Western
Musical
Animation
War
Mystery
Horror
Children
Fantasy
Crime
Sci−Fi
Romance
Adventure
Thriller
Action
Comedy
Drama
Genre
1M
2M
3M
4M
Counts



To measure how close the predictions were to the true values in the validation set we will use the Root Mean Square Error (RMSE), defined by the following function:

```
RMSE <- function(true_ratings, predicted_ratings){
    sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

Analysis Section

As explained before, due to the size of the dataset, modeling the data using a function like *lm* is not appropriate. To solve this problem we computed the least square estimates manually. First, we started with the most simple model to have a baseline: predict the same rating regardless of the user, movie or genre. In this model and all the others tested we have limited the predicted ratings to a minimum value of 0.5 and a maximum of 5. This model would look like this:

$$Y_{u,i} = \mu + {}_{u,i}$$

Where u is the index for users, and i for movies. For this, the estimate for μ is the average of all ratings, which is 3.5124652.

Method	**RMSE**
Just the average	1.060331

We assume that some movies have higher ratings than others, so the following model considers the movie effect. We estimate the movie effect as the average of the ratings by a movie. For exemple, the movie indexed as 1 (Toy Story) has a positive effect of 0.4151725 to the average of all ratings (3.5124652), while the

movie number 2 (Jumanji) has a negative effect of -0.3070658. As shown below, we can see that this already improved the model.

Method	RMSE
Movie Effect Model	0.9439087

Besides the movie effect, we also assume that some users rate movies higher than others, so the next model considers both the movie and the user effect. We estimate the user effect as the average of the ratings per user. For example, the user indexed as 1 has a positive effect of 1.6792347 to the average of all ratings (3.5124652), indicating that this person is more generous to rate. In contrast, user number 2 has a negative effect of -0.2364086, suggesting this is a more critical viewer.As shown below, this improves the model.

Method	RMSE
Movie + User Effects Model	0.8651613

As presented before, the movie ratings vary per genre, so the following model will also include the genre effect.

Method	RMSE
Movie + User + Genres Effects Model	0.8647516

The previous model improved the RMSE by only a small amount. This plausibly happened because this model treated the genres together (ie: “Action|Adventure|Animation|Children|Comedy”), holding 797 different combinations. To improve this, the next model treated the genres independently: a movie is or not of a certain genre. We estimate the genre effect as the average of the ratings per genre. The most positive genre effect was from documentaries, while the worst effect was from children movies.

Method	RMSE
Movie + User + Genres Ind. Effects Model	0.8629492

Treating the genre effect independently reduced the RMSE slightly more than before. Now we will include the user-genre effect, as we expect that users rate genres differently.

Method	RMSE
Movie + User + Genres Ind. + Genre_User Effects Model	0.8647424

This new model increased the RMSE. Our final consideration is that the rating estimate for a movie rated many times is more likely to be more precise than the estimate of a movie rated only a handful of times. Regularization is what allows us to penalize those estimates constructed using small sample sizes. When the sample size is very large, the estimate is more stable, but when the sample size is very small, the estimate is shrunken towards 0. The larger the penalty parameter λ, the more the estimate is shrunk. As λ is a tuning parameter, we did a grid search to choose its optimal value.

Method	RMSE
Regularized Movie + User + Genre Ind. + Movie_Genre + Genre_User Effect Model	0.8531658

Result Section

To predict movie ratings we build models that considered the effects of movies, users, genres and interactions of these. The best model considered all, achieving an RMSE of Regularized Movie + User + Genre Ind. + Movie_Genre + Genre_User Effect Model, 0.8531658. The movie effect decreased the RMSE the most, suggesting that the movie in itself is of greatest importance to explain the rating.

Method	RMSE
Just the average	1.0603313
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8651613
Movie + User + Genres Effects Model	0.8647516
Movie + User + Genres Ind. Effects Model	0.8629492
Movie + User + Genres Ind. + Genre_User Effects Model	0.8647424
Regularized Movie + User + Genre Ind. + Movie_Genre + Genre_User Effect Model	0.8531658

Conclusion Section

This project's aim was to predict movie ratings from a database with over 10 million evaluations. To try this, we considered the impact of movies, users and genres on ratings. We divided the dataset into train and validation to avoid overfitting. As the dataset was large, usual data wrangling was no longer feasible in most computers due to memory allocation. To solve this problem, we computed the least square estimates manually.According to the sparsity nature of the dataset, we also included regularization. The best-fitted model achieved an RMSE of Regularized Movie + User + Genre Ind. + Movie_Genre + Genre_User Effect Model, 0.8531658, which is considered very good for the course's standards.

It would have been interesting to have more information about the users (e.g. age and gender) and the movies (e.g. actors, director and language) to try to improve the model.