# *MovieLens Recommendation System Report*

## ***HarvardX Data Science Professional Certificate: PH125.9x Capstone Project 1***

Mateus Spencer

*07 January, 2024*

# 1. Introduction

Recommendation systems have the goal of predicting a user's rating or preference towards a certain item. They can be applied on a wide range of domains such as, music, books, search queries, movies, restaurants and products in general. They offer great value to companies that handle big amounts of data, as they are able to improve a users experience and consequently drive up the companies revenues.

In this project we will use the MovieLens dataset, generated by the GroupLens research lab, to train a Model for predicting movie ratings and evaluating the RMSE of the evolution of the model to asses its performance.

## 1.1 The Dataset

The full MovieLens dataset can be found here: https://grouplens.org/datasets/movielens/latest/. The Full dataset now has approximately 33,000,000 entries (07/01/2024), but we will be using a subset of only 10,000,000 entries to be able to train the models on a laptop, it can be downlodade at: https://grouplens.org/datasets/movielens/10m/. Running the code provided by HarvardX the dataset is split into two datasets, edx having 90% used to train the model and final_holdout_test with 10% of the data used only for final evaluation.

By analyzing the structure of the dataset we observe that it has 6 variables that are described in the table below:

| Name | Format | Description |
|---|---|---|
| `userId` | Integer | Unique numerical identifier for each user |
| `movieId` | Integer | Unique numerical identifier |
| `rating` | Numerical | Rating given to a movie by a user, starting at 0 and going up to 5 in steps of 0.5 |
| `timestamp` | Integer | Unix epoch of the date/time of the rating (number of seconds since 1-Jan-1970. |
| `title` | Character string | Name of the Movie & year of release |
| `genres` | Character string | Genres the movie belongs to, separated by \| |

Here are some of the first entries of the edx dataset:

| | 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 |

Below are some metrics of each of the datasets that show that they are evenly distributed

Table 3: edx dataset summary

| rows_number | users_number | movies_number | average_rating | first_rating_Date | last_rating_date |
|---|---|---|---|---|---|
| 9000055 | 69878 | 10677 | 3.512 | 1995-01-09 | 2009-01-05 |

Table 4: final_holdout_test dataset summary

| rows_number | users_number | movies_number | average_rating | first_rating_Date | last_rating_date |
|---|---|---|---|---|---|
| 999999 | 68534 | 9809 | 3.512 | 1995-01-09 | 2009-01-05 |

## 1.2 Project methodology

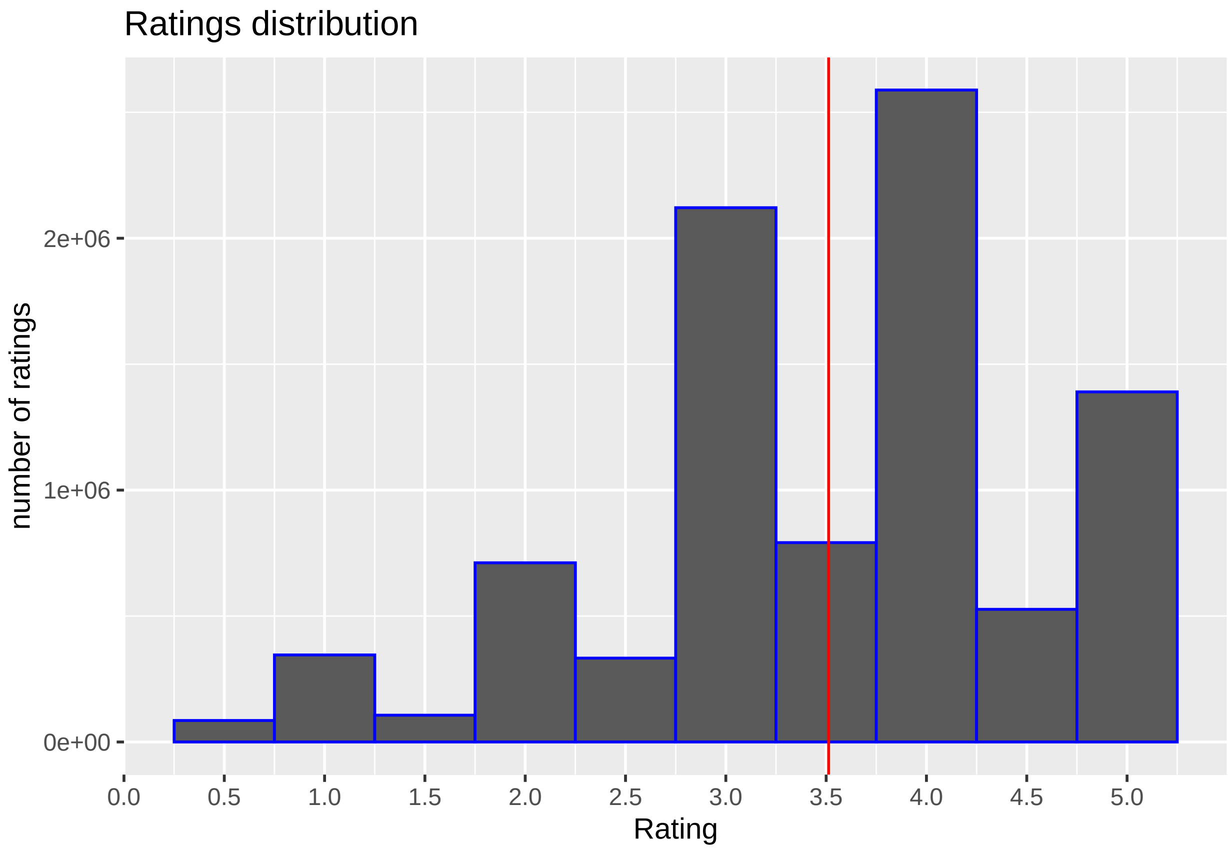
This work will start by firstly doing an exploratory analysis of the dataset in order to gain a better understanding of the data and figure out what variables are more valuable and should be used in the models. After that we will start to develop the model by starting with a simple model and iterating over it to improve it by lowering the RMSE to evaluate the model with the final_holdout_test dataset.The target RMSE to reach will be 0.86490.

# 2. Analysis and Exploration

in the following sections we will analyze each feature individually in order to decide which ones provide the most information and should be firstly implemented in the model.

## 2.1 Ratings Exploration

The rating is an ordinal scale of number from 0.5 to 5 in steps of 0.5 given by the users who watched the movie. Here we can see a histogram of the distribution of all the ratings with the red line representing the mean.

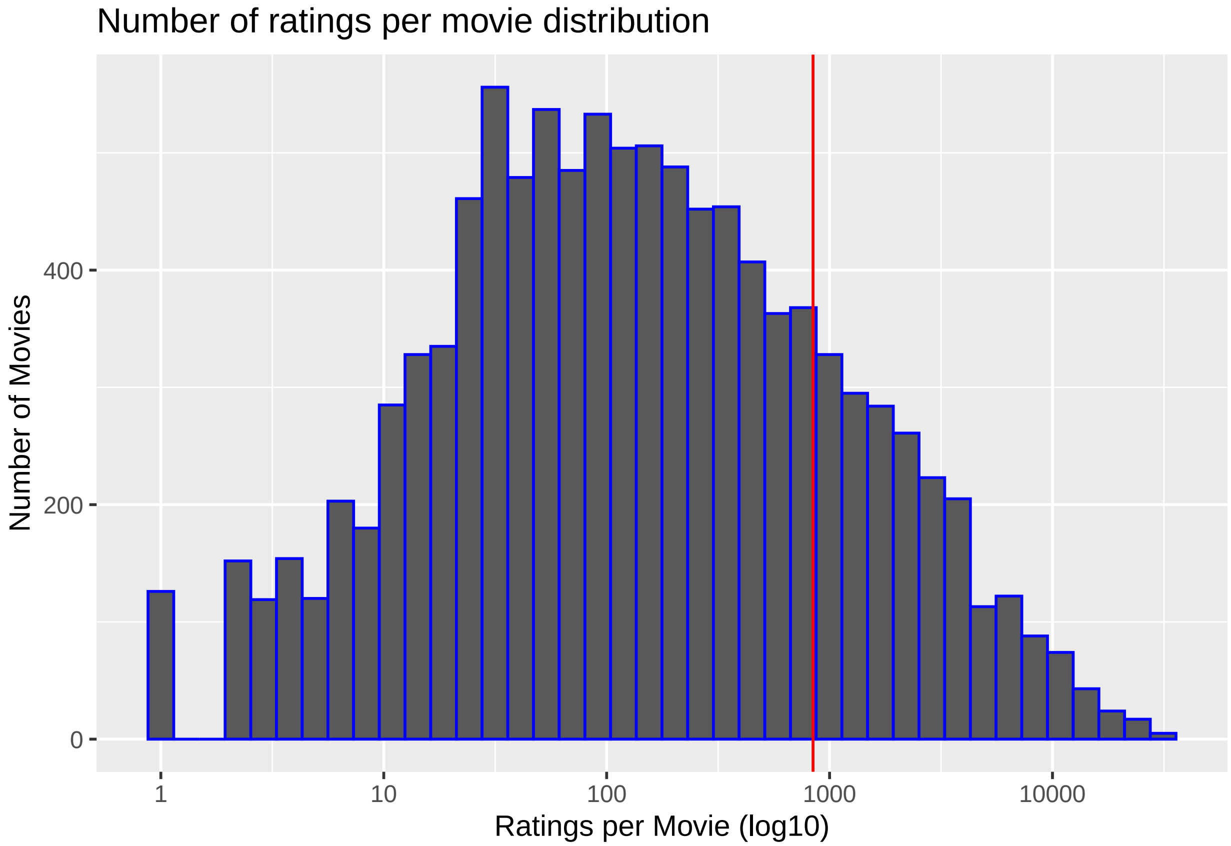


We can conclude:

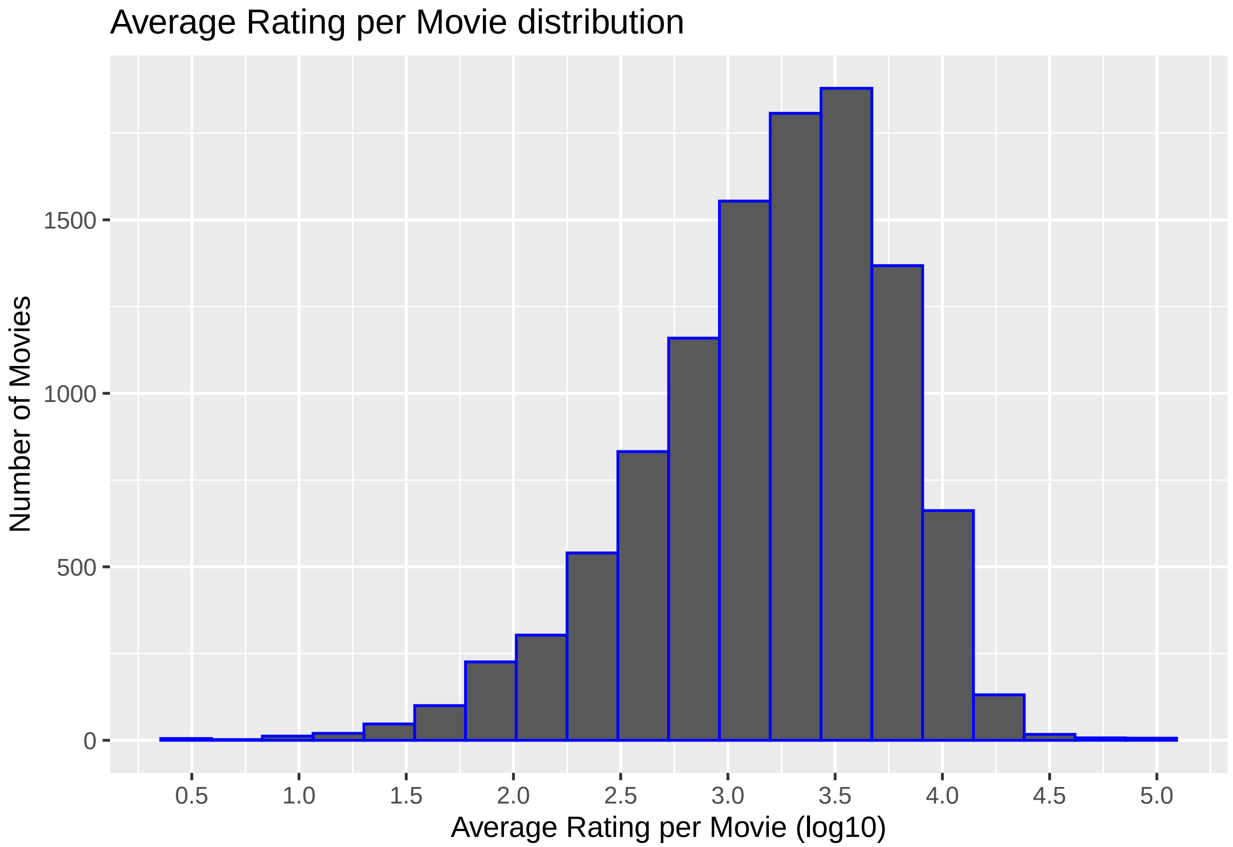
- The overall average rating in the edx dataset was 3.51
- The top 3 ratings from most to least are : 4, 3, 5.
- There is a propensity for users to give higher ratings to movies.
- It is also clear that the ratings between full numbers are less popular than their counterparts.

### 2.2 Movies Exploration

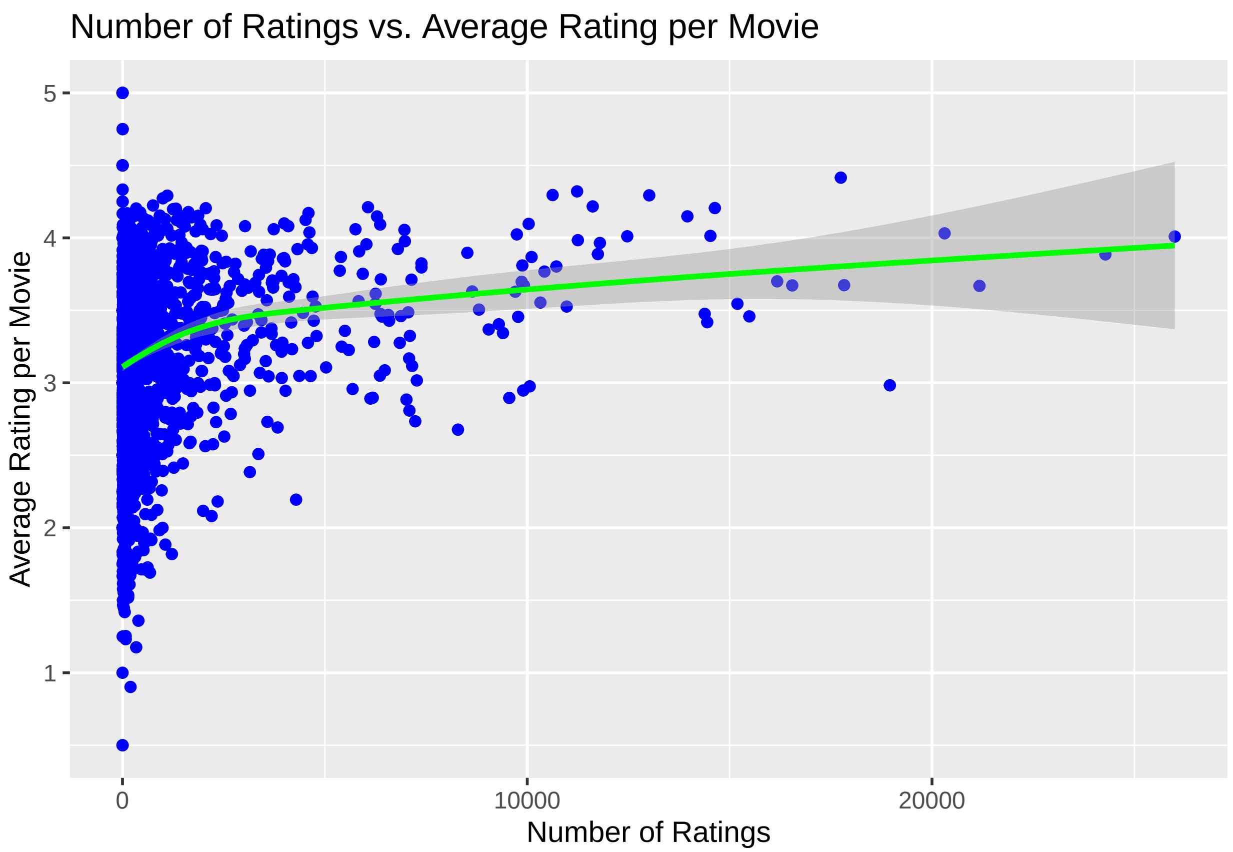
The edx dataset has total of 10677 movies. Below is a histogram of the distribution of number of ratings per movies. The graph if presented in a logarithmic scale becacuse it is much more common for movies to have fewer ratings, but some outliers with more that 10,000 ratings stretch out the histogram too much making the bins for less rated per movies too big for their variation.



The following histogram of the distribution of the average rating per movie is in accordance with the ratings histogram, but the mean eliminated the preference for whole numbers.

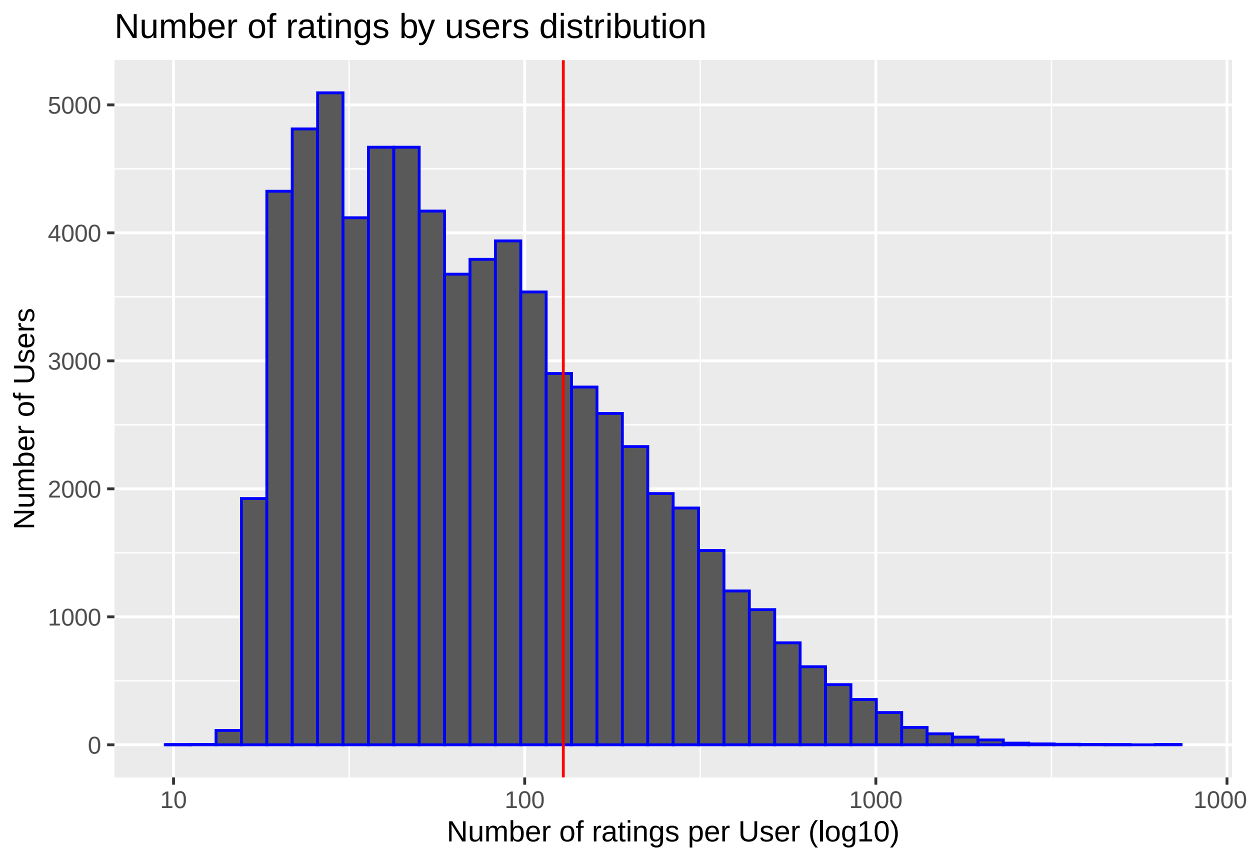


Here, we can see that there is a bigger variation of the average rating among movies with a lower number of ratings. We also see that the average rating of movies increases with the number of ratings given to a movie, which would make sense since people prefer to watch good movies and will prefer to watch those, and there fore giving them more ratings.

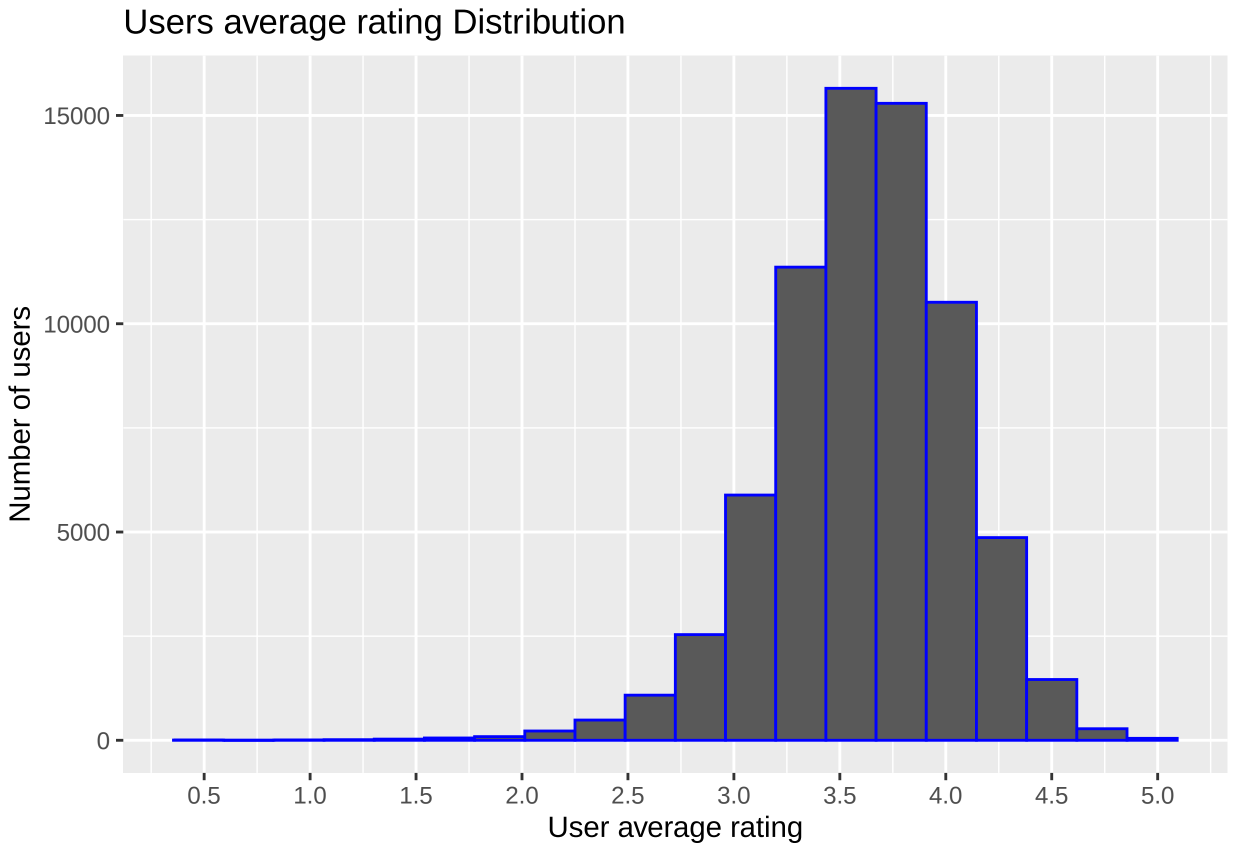


## 2.3 User Exploration

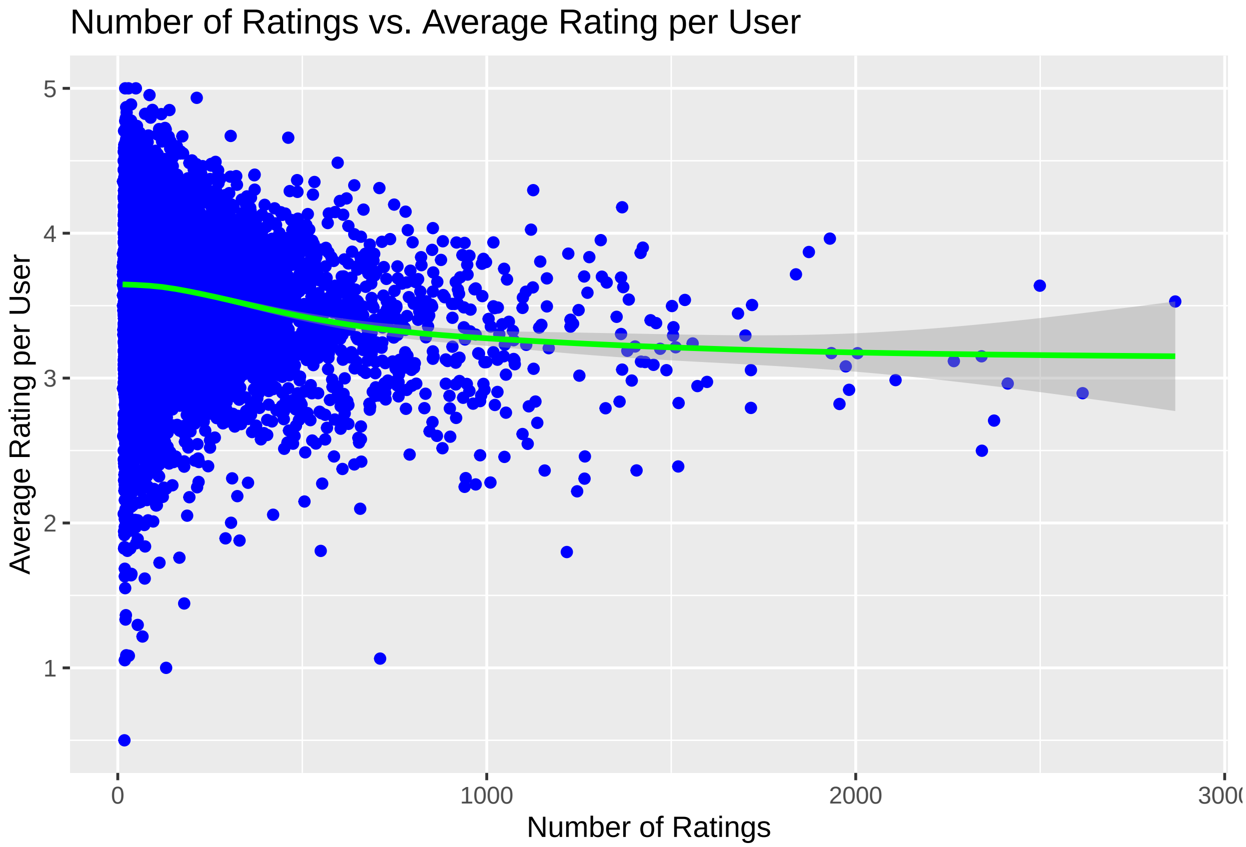
The edx dataset has 69878 users represented by userid. The following histogram represent the number of ratings by user with the x axis again in logarithmic scale to mitigate the stretching out of the histogram by outliers wit a lot of ratings:



As in the movies average rating histogram here we also see a similarity to the ratings histogram without the bias towards whole numbers.

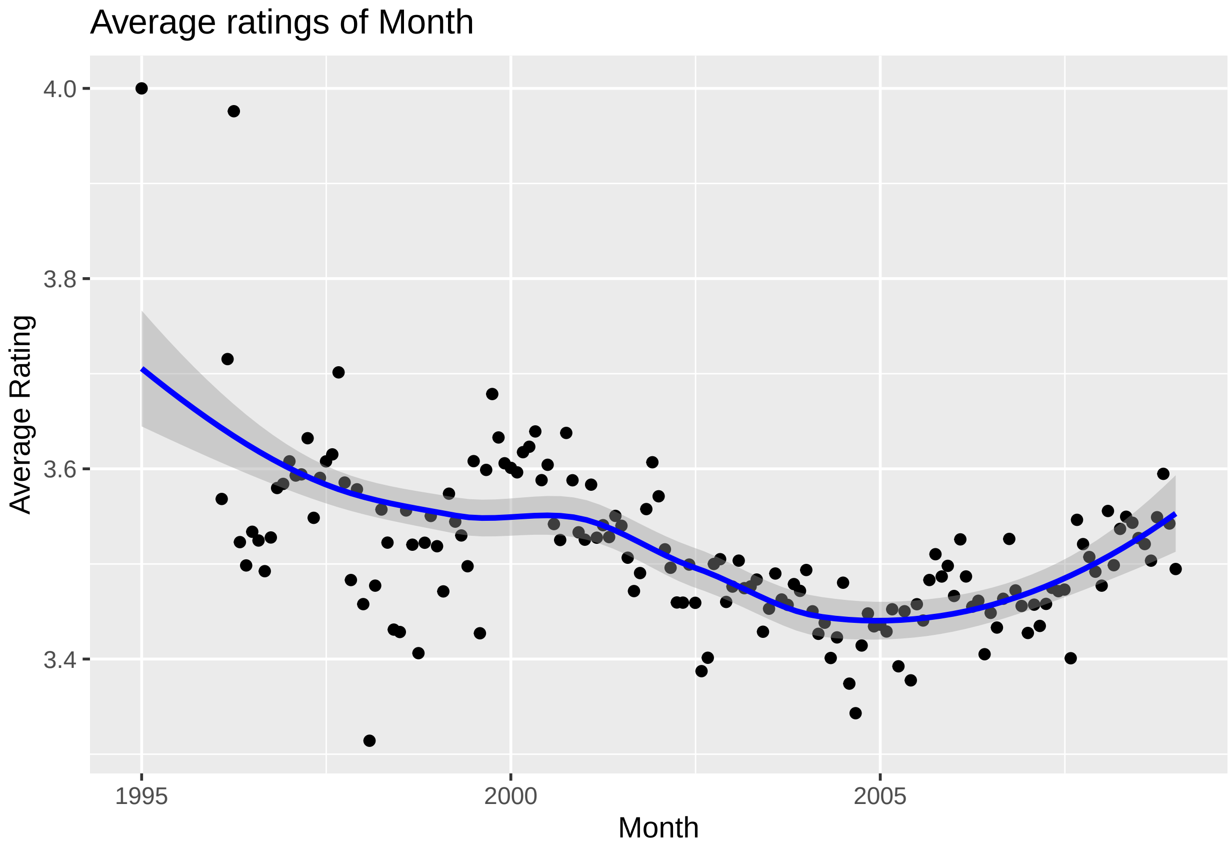


To conclude, in this plot we see again a larger diversity in average rating for movies for users with fewer ratings, nut in this case there is no increase in average rating with the number of ratings.

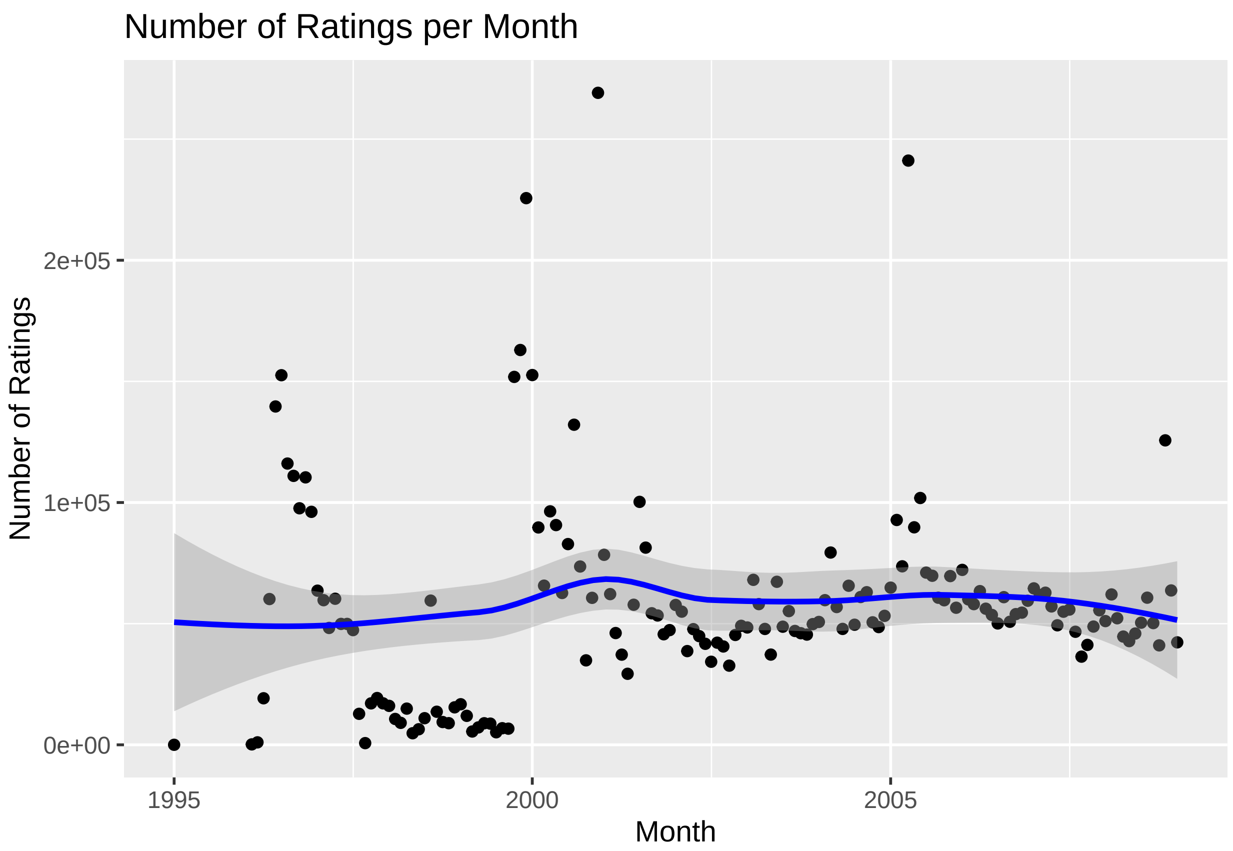


## 2.4 Time effect Exploration

By exploring the time at which a rating was given we can explore if the average rating per movie has a correlation to the time at which it was given. From the plot bellow we see a slight decrease over the years with a small climb since 2005 in the average of ratings given out in a certain month, however, this variation happens in a very narrow window between 3.7 and 3.4 rating therefore not being very relevant. We see however a noticeable increase in consistency of ratings between months.



As for the number of ratings per month overtime we see a general consistency marked by very agresive spikes in certain periods. This coulb be explained due to the release of very popular movies that gather a lot of ratings, which people would watch on the relsease nonth and the following months.



## 2.5 Genres Exploration

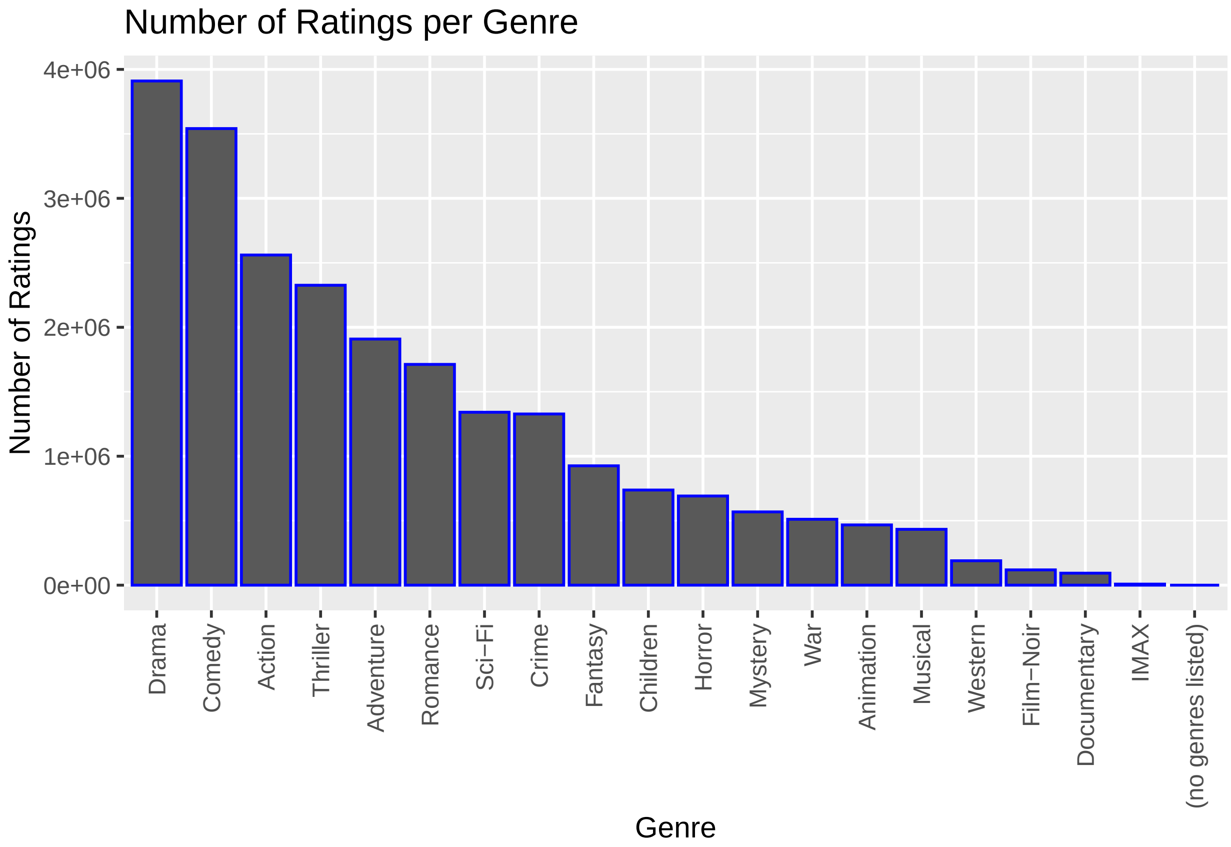
A movie could be classified to one or more genres. There are 797 genre combinations for the movies in the dataset.

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

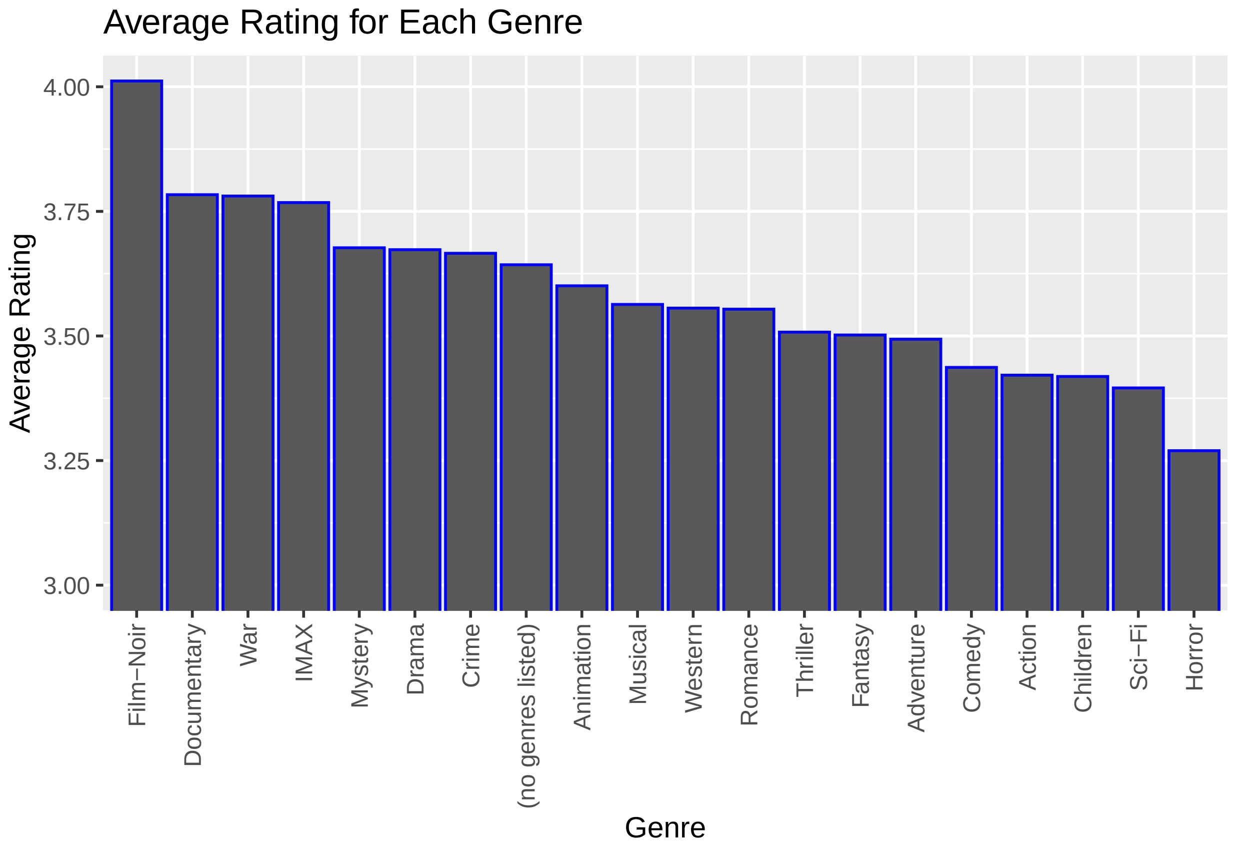
```
##  [1] "Comedy"             "Romance"            "Action"
##  [4] "Crime"              "Thriller"           "Drama"
##  [7] "Sci-Fi"             "Adventure"          "Children"
## [10] "Fantasy"            "War"                "Animation"
## [13] "Musical"            "Western"            "Mystery"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

We will be analyzing each genre independentluy to asses if there is any preference for certinn genres.

In this bar plot we see that there is a general preference for certain genres over others.



By looking at both plots we can see that some of the genres with the lowest amounts of ratings have some of the highest average ratings, this could be due to the fact of them being more niche genres whose movies are mostly seen by people who enjoy those genres and will naturally give out higher ratings.



Although there is some significant variation in the average ratings between genres it is not as significant as the variation in other features such as movies and users average ratings therefore we will not prioritize the inclusion of this feature in our model as much. We should also note that due to the complexity of all of the combinations of genres for each movies building an effective model with the genres would require more advanced methods and bigger computing power.

# 3. Methodology

## 3.1 Model performance evaluation

To evaluate the evolution of performance of our model, we will use root mean squared error (RMSE) as the loss function. The RMSE represent the error loss between the predicted ratings derived from applying the model and actual ratings in the test set.

In the formula shown below, $y_{u,i}$ is defined as the actual rating provided by user $i$ for movie $u$, $\hat{y} * u, i$ is the predicted rating for the same, and N is the total number of user/movie combinations.

$$RMSE = \sqrt{\frac{1}{N} \sum *u, i \left(\hat{y} * u, i - y * u, i\right)^2}$$

## 3.2 Modeling Approach

Due to the size of our training model, conventional statistical methods would take too long to compute, therefore we will be using a machine learning approach as discussed in the course.

# 4. Results

Based on the exploration of the data described above, we will firstly implement user and movie effects in our model since time and genre effect are more marginal.

## 4.1 Model 1: BaseLine Model

Starting with the Simplest Model as a Baseline: predicting the same rating regardless of the user, movie or genre.

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

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

| Model | RMSE |
|---|---|
| Just the Average | 1.061202 |

As expected the RMSE is not great, but it gives us a good starting point from where to build our model.

## 4.2 Model 2: Movie effects Model

In this model, the effects of individual movies are considered. The bias for each movie (b_i) is calculated, representing the difference between the average rating of the movie and the overall average rating. This introduces a level of personalization based on movie preferences. While an improvement over the baseline model, it still has limitations, especially in accounting for user-specific preferences.

| | Model | RMSE |
|---|---|---|
| 2 | Movie Effect Model | 0.9439 |

## 4.3 Model 3: Movies and Users effects Model

Building upon the movie effect model, this approach incorporates user-specific biases (b_u). Now, the model considers both movie and user effects to predict ratings. The RMSE is further reduced compared to the previous models, showcasing the importance of accounting for individual user preferences in recommendation systems.

|   | Model | RMSE |
|---|---|---|
| 3 | Movie and User Effect model | 0.8653 |

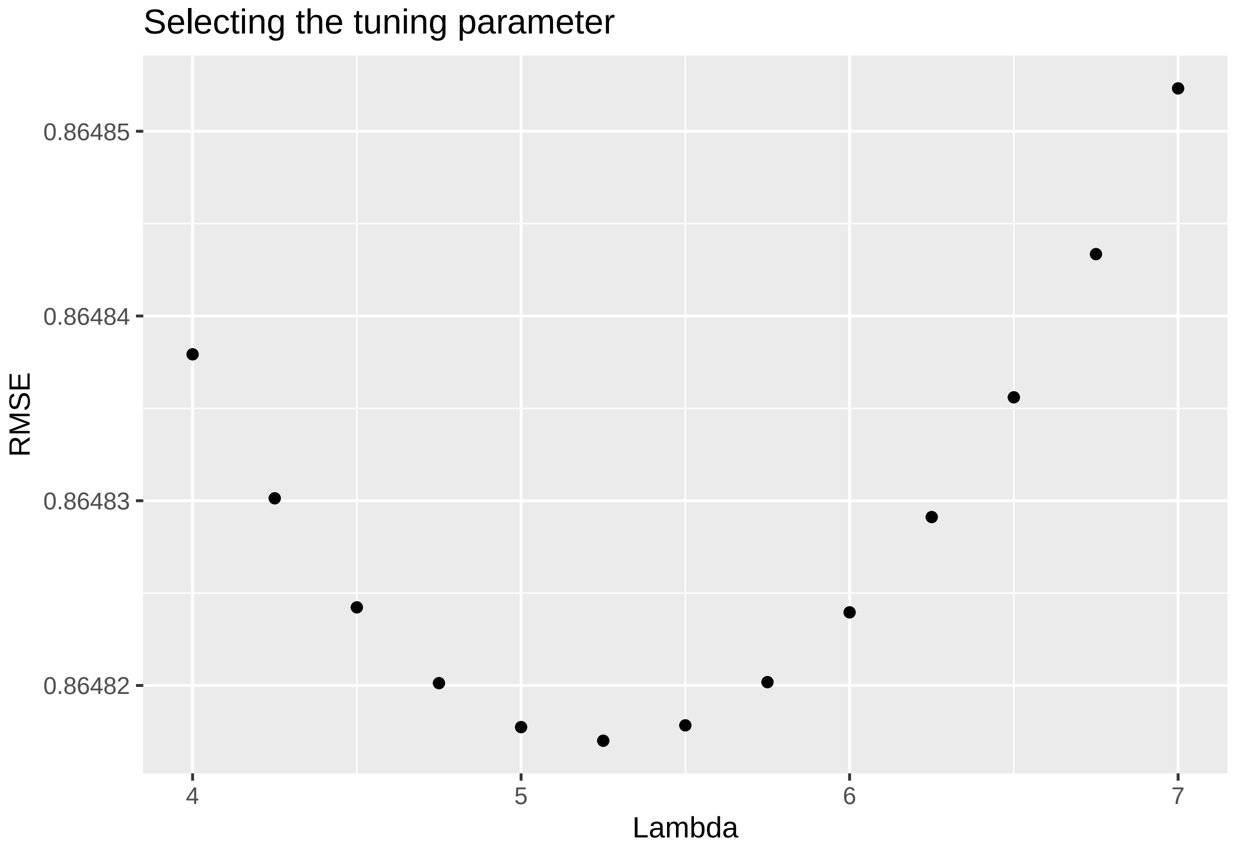
## 4.4 Model 4: Regularization of Movies and Users effects Model

To enhance the predictive performance of our recommendation model, we introduce regularization. Regularization is a technique used to prevent overfitting by adding a penalty term to the model parameters. In our case, regularization is applied to both movie and user effects, represented by b_i and b_u, respectively.

The regularization term imposes a constraint on the magnitude of the biases (b_i and b_u). It discourages extreme values, preventing the model from becoming overly complex and tailored to the training data. This is crucial for generalizing well to unseen data, ensuring that our model doesn't capture noise in the training set.

Lambda, the regularization parameter, controls the strength of the regularization. Choosing an appropriate lambda is a critical step. We systematically explore a range of lambda values and select the one that minimizes the Root Mean Squared Error (RMSE) on our validation set. The plot above displays the relationship between different lambda values and their corresponding RMSE scores. The lambda with the minimum RMSE serves as our optimal regularization parameter.

Benefits of Regularization Regularization not only improves the model's ability to generalize but also helps in managing bias-variance trade-off. By preventing the model from fitting the training data too closely, it encourages a more balanced and robust model. The regularization technique contributes to the overall effectiveness of our recommendation system by striking a better balance between complexity and simplicity.



Lambda, the regularization parameter, controls the strength of the regularization. Choosing an appropriate lambda is a critical step. We systematically explore a range of lambda values and select the one that minimizes the Root Mean Squared Error (RMSE) on our validation set. The plot above displays the relationship between different lambda values and their corresponding RMSE scores. The lambda with the minimum RMSE (5.25) serves as our optimal regularization parameter.

| Model | RMSE |
|---|---:|
| Just the Average | 1.061202 |
| Movie Effect Model | 0.943900 |
| Movie and User Effect model | 0.865300 |
| Regularized Movie and User Effect Model | 0.864800 |

Since we have surpassed the target RMSE of 0.8649 we will not be improving the model any further.

# 5. Conculusions

In summary, our exploration and analysis of the MovieLens dataset, coupled with the iterative development of recommendation models, have yielded valuable insights into user behavior and the dynamics of the movie ratings landscape. Our final model, the Regularized Movie and User Effect Model, successfully surpassed the target RMSE, underscoring its proficiency in predicting user ratings.

To further enhance the performance of our recommendation model, several techniques could be explored, matrix factorization techniques, such as Singular Value Decomposition (SVD) or Alternating Least Squares (ALS). Matrix factorization allows the model to capture latent features and relationships between users and movies, providing a more nuanced understanding of preferences.

# 6. References

[1] "Introduction to Data Science - Data Analysis and Prediction Algorithms with R", Dr. Rafael A. Irizarry link

[2] "R Markdown: The Definitive Guide", Yihui Xie, J. J. Allaire, Garrett Grolemund, 2019-06-03 link