Movie Recommendation System

Moschos Evangelos-Jason

Abstract

In this document, we are presenting our analysis on a movie recommendation system, based on the famous Movielens dataset. The project is conducted for the Harvard Data science professional program.

Contents

Executive summary **2**

1 Introduction **3**

2 Literature review **4**

3 Dataset generation **5**

4 Exploratory Data Analysis **7**

4.1 Timestamp analysis 7

4.2 User-based analysis 8

4.3 Movie analysis 12

4.4 Genre Analysis 17

4.5 Rating analysis 19

5 Analysis: Model building and testing **21**

5.1 Model 1: Naive prediction 21

5.2 Model 2: Adding a time bias to the model 21

5.3 Model 3: Adding a movie bias to the model 21

5.4 Model 4: Adding a user-bias to the model 22

5.5 Model 5: Adding the genre bias to the model 22

6 Results: Model assessment & validation **24**

7 Conclusion **25**

7.1 Discussion of results 25

7.2 Limitations 25

7.3 Further research 25

References **26**

Executive summary

Recommendation systems (RS) are becoming increasingly more prevelant in today's world. They are systems that try to predict or filter preferences according to the user's profile and past choices. From Netflix, to Youtube, to Amazon, Recommendation systems are helping millions of users to find content that is interesting to them.

As part of the Harvard data science program, we will be attempting to create a movie recommendation system, based on the famous MovieLens dataset. The dataset we are using is the one with 10 million rows, downloaded from (https://grouplens.org/datasets/movielens/). The aim of the project is to be able to predict the rating a user might give to a movie (based on some characteristics), in order for us to be able to recommend movies with high predicted ratings to that user.

In our exploratory analysis, we managed to identify four main variables that appear to drive variability in the rating: namely a movie effect, a time effect, a user-based effect and an effect for the different genres. For each of those variables, a different analysis was conducted, but the interaction between them was not studied explicitly.

For the movie effect (or movie bias), we examined how the average rating differed between movies. The analysis shows evidence that movies with a higher number of ratings tend to receive higher absolute ratings from the users, while simultaneously movies with a lower number of ratings tend to have a more wide (sparse) distribution. To verify this assumption, we sampled different movies based on the number of ratings and examined their rating distribution. Based on this analysis, we believe that movies have a strong effect on how a user might rate them, which is in-line with our empirical knowledge.

Regarding the user effect (user bias), we examined the number of times a user rates movies, as well as the cumulative distribution of ratings grouped by user, and discovered that while only a small portion of users is having an extremely high number of ratings, the general Pareto rule (80/20) does not appear to hold true for the dataset. Additionally, when sampling individual users, based on their number of ratings, and examining their rating behaviour, there is strong evidence that a user effect is also present in our dataset.

For the time bias, we studied the effect the time variable has on the rating. Our results indicate that time does have a small effect on the average rating, but it is far too volatile to be used directly in our models. For that reason, we created a smoothened curve for the time variable, which was later used for our predictions.

Finally, we attempted to study the effect that individual genres have on user rating. Our approach was twofold: 1) treat genre combinations as unique genres and 2) Split the genre combinations to the individual genres, they consist of. Based on our analysis, we concluded that the first option appeared to have a larger variability, compared to the second, and thus we selected to treat genre combinations as individual genres in our models.

Based on our EDA, we have developed five main models that consider these factors or a combination of them. The best performing model we developed included all four parameters, and it achieved an

- **RMSE of 0.8631 in the test-set**
- **RMSE of 0.8647 in the validation set**.

The analysis shows significant improvement over simpler methods (20% improvement), but several other factors can be incorporated to further improve the accuracy of the model.

1 Introduction

The rapid growth of the internet has led to the adoption of an e-business model by many traditional (brick-and-mortar) businesses. A basic requirement for all these businesses is to provide the user with an appropriate assortment of products or services, tailored to the specific needs of each user. Recommendation systems are used to fill that gap; their basic idea and goal is to utilize different sources of data and generate meaningful and accurate predictions for the users.

As part of the Harvard data science professional program, we will attempt to create a movie recommendation system. In the next chapters, our analysis is structured as follows:

- Chapter 2: Literature review
- Chapter 3: Dataset generation
- Chapter 4: Exploratory Data Analysis
- Chapter 5: Analysis: Model building & Testing
- Chapter 6: Results: Model assessment & validation
- Chapter 7: Conclusion

In chapter 2, a brief literature review is provided on recommendation systems; the review focuses only on the classification of techniques for recommendation systems, and their advantages and disadvantages.

In chapter 3, the generation of the dataset is explained, and some basic dimensions of the data are explored.

In chapter 4, an Exploratory data analysis is conducted, to identify interesting features in the dataset, and help us select the appropriate variables in our models.

In chapter 5, the basic models are built, based on the characteristics identified in chapter 4. All the models are also evaluated.

In chapter 6 the best model of chapter 5 is selected and its performance is validated on a new dataset.

In chapter 7, an overview of the results is presented, limitations of the project are discussed, and direction for future research is provided.

2 Literature review

Recommendation systems are typically classified into three categories (Isinkaye, Folajimi, and Ojokoh 2015):

- Content-based filtering
- Collaborative filtering
- Hybrid systems

Content-based filtering systems (CBF) recommend content based on the user's profile and the attribute of items (Pazzani and Billsus 2007). An advantage of such system is that it does not require data from other users and it can capture the taste of each specific user (Rafsanjani et al. 2013). However, it is not without its limitations; namely, the "Cold start problem", where a user has to first rate an item for the system to make a prediction. Additionally, the fact that the user will only be recommended items based on his existing profile is another major disadvantage (Adomavicius and Tuzhilin 2005).

On the other hand, **collaborative filtering systems** (CF) recommend items similar to those selected by other people with similar preferences(LYLE 2012). CF has two main advantages: it does not need any domain knowledge, and it can help users identify new interests (Rafsanjani et al. 2013). The disadvantage of such recommendation system is twofold: data sparsity and the cold start problem . Data sparsity refers to the fact that many users and item combinations simply do not exist (LYLE 2012), while the cold start problem is similar to the one found in Content-based systems: new users must rate an item before a recommendation is made(Thorat, Goudar, and Barve 2015).

Collaborative filtering systems can be futher split into Model based and Memory based (LYLE 2012).

The final category are the **hybrid systems**, which use a combination of the other approaches (Sánchez 2013), in order to tackle some of the problems faced by the individual approaches (cold start, data sparsity etc.). Thorat, Goudar, and Barve (2015) describe the four main approaches for hybrid systems as follows:

- CF and CBF are implemented seperately, and their predictions are aggregated.
- Parts of CBF are integrated into CF.
- Parts of CF are integrated into CBF.
- A new generalized "consolidative model" is created, where both CF and CBF characteristics are integrated.

3 Dataset generation

In this chapter, we will describe the dataset generation process, some useful (general) statistics for the dataset and some first insights we have.

The entire dataset is donwloaded from https://grouplens.org/datasets/movielens/10m/.

We begin by examining the different variables

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	231	5	838983392	Dumb & Dumber (1994)	Comedy
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi

There are 6 variables:

- UserId: indicating the user
- movieId: indicating the movie
- rating: the variable we plan to predict: the rating each movie received by a user.
- timestamp: indicating when was the movie rated.
- title: the title of the movie
- genres: indicating the genre

Some initials first observations we have is that the timestamp is in a format we cannot interpret at the moment and thus need to be converted. Furthermore, the genres are all in one row, seperated by "|".

We will now examine how many rows our dataset has, and how many unique users and movies exist.

Number of rows:

```
[1] 10000054
```

Number of unique users:

```
[1] 69878
```

Number of unique movies:

```
[1] 10677
```

The dataset itself is rather large with 10.000.000 datapoints, and almost 70.000 users and 11.000 movies. In fact, the size of the dataset is a prohibiting factor for most Machine learning algorithms (at least in a local setup), as they require immense computational time.

To assess the data completeness, we are going to check the number of missing values in our dataset.

	Number of missing values
userId	0
movieId	0
rating	0
timestamp	0
title	0
genres	0

It appears that our dataset does not contain any missing values.

Before proceeding to the Exploratory data analysis, in order to be able to assess the quality of our algorithms, we are going to split the original dataset (movielens) into two sets: edx and validation. The edx dataset will be the only one used for all our actions, and validation will only be used at the end to calculate the accuracy of our best algorithm in unseen data. The rule used for the splitting is 90/10 (90% of the data in edx, 10% in validation).

Finally, we will further split our dataset (edx) into a training and a test set; the training set will be used for fitting our models, and the test set will be used for evaluating them and comparing with each other. We have used a 90/10 split rule, similar to how we constructed the validation set.

We now have 3 datasets:

- edx: Used for modelling purposes.
- train: Used for model comparison.
- validation: Used, once the best model has been decided, to calculate accuracy in unseen data.

4 Exploratory Data Analysis

In this section, we are presenting our exploratory data analysis (EDA). From the variables in our dataset, described in the previous section, the title does not appear to hold any value in our prediction. Therefore, we will only examine the remaining 5 variables. A dedicated section is devoted to each one of them.

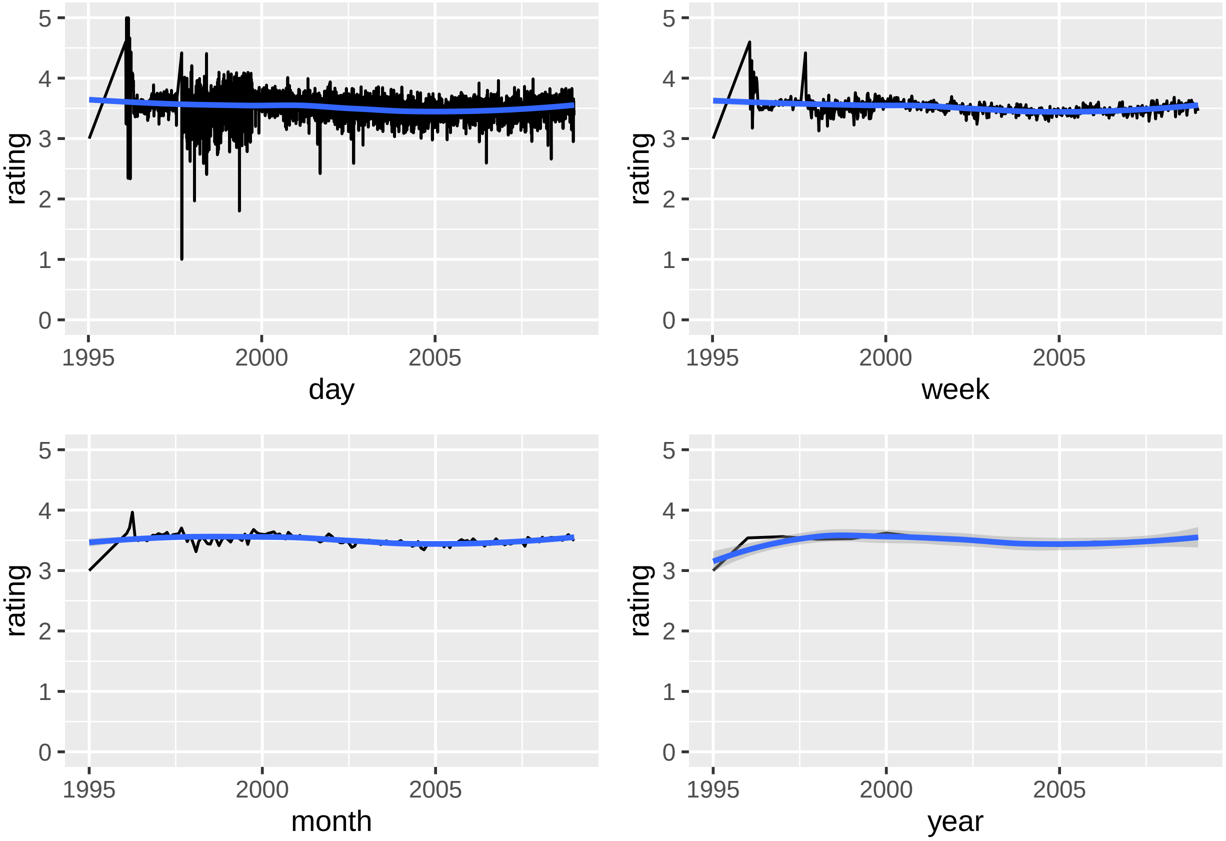
4.1 Timestamp analysis

As part of the analysis, we want to study the relationship between the time (timestamp) variable and the rating. Alternatively, the research question can be formulated as: "Do people seem to rate movies differently in different time periods?". From a logical point of view, some periods are associated with better-quality movies, and thus the rating might differ for those periods. We will be attempting to answer this question based on our available data.

Since our dataset has a timestamp variable, which cannot be used directly in that format, we need to convert it to a date variable. Depending on the unit of choice, we expect a different behaviour; due to the pooling effect, aggregating the date more, will result to a more smooth surface. We will test the distribution at a daily, weekly monthly and yearly level.

For our analysis, firstly we convert the timestamp to a date format and subsequently aggregate our data by date. For each date, we plot the aggregated data and also fit a smoothened curve (loess).

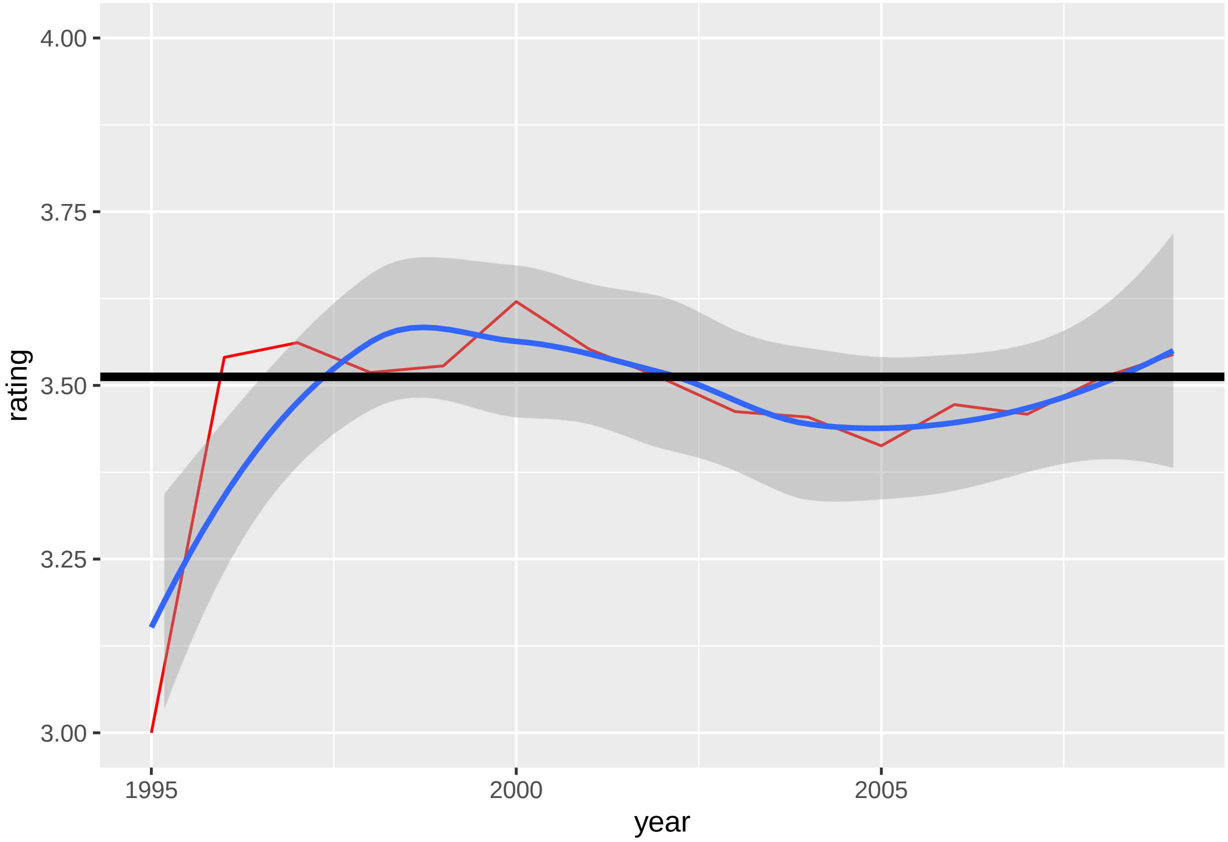
All 4 plots are depicted in the following figure:



From the plots we can clearly see that the average rating is not stable across time and thus it appears that the time does have an impact on the average rating.

We should also note that when the grouping is done at a daily, weekly or monthly level, the resulted time-series object is far too volatile. When however the aggregation is performed at a yearly level, the surface is rather smooth, and the fitted line appears to capture the general trend, while also being close to the original data.

To see this, we are providing the plot with only the yearly level (red), the overall average rating (black) and our fitted line(blue):



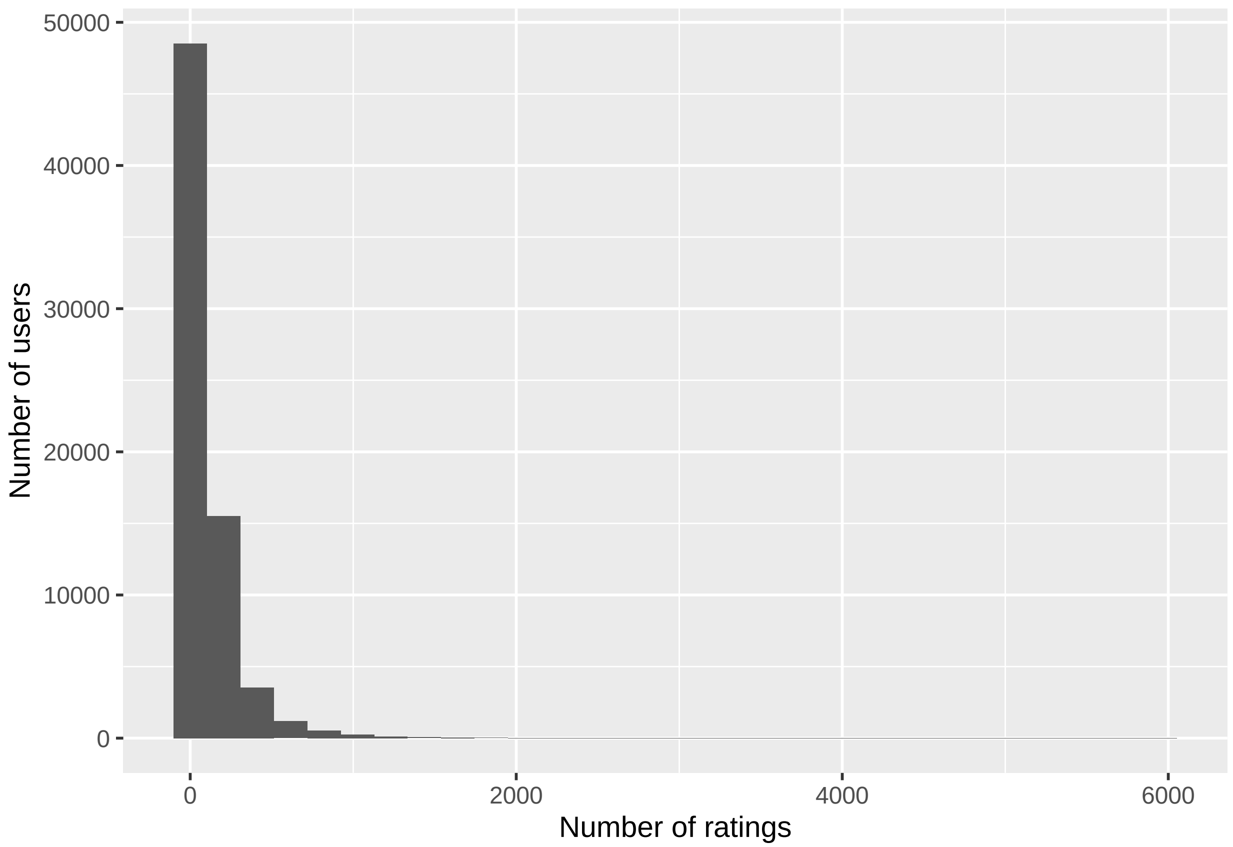
Based on this plot, we observe that the average rating, stratified by date (yearly aggregation), is different than the average of the entire dataset, and the fitted line can be used to approximate that effect.

Therefore, we expect the time variable to have an impact in our rating prediction, and thus its effectiveness will be tested in our models.

4.2 User-based analysis

For this section, we aim to study the user effect or user bias. More specifically, different users have different behaviours when rating a movie; some are more positive and give higher ratings, while others do the opposite.

To begin with, we want to see how many times have different users rated movies. For that reason, we create a histogram with the total number of times users have rated movies, grouped by user.



We observe that most users have a number of ratings less than 2000, but there are some outliers that have rated movies more than 3000 times, as seen in the following table.

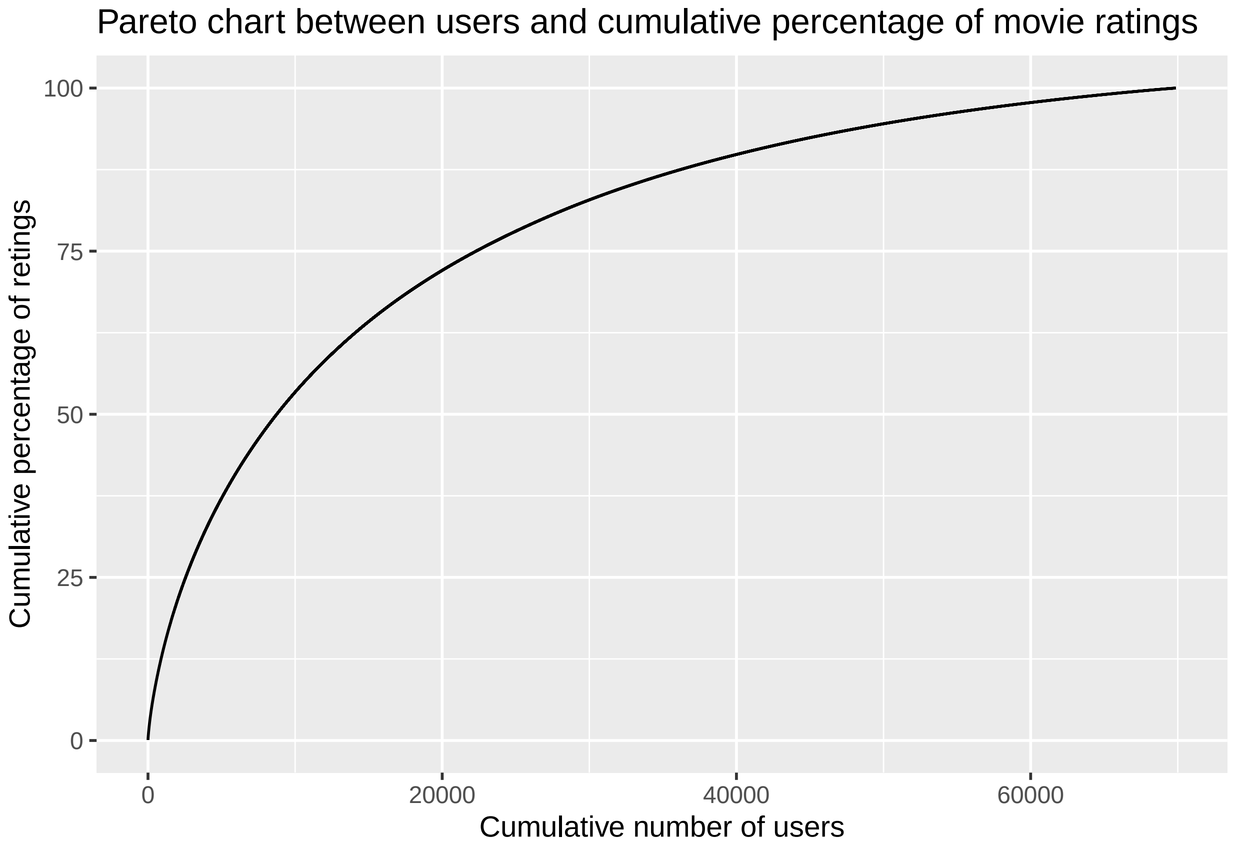
userId	count
59269	5962
67385	5736
14463	4193
68259	3640
27468	3594

In the literature review, we described the cold start problem, in which we have users who have never rated a movie before. We want to examine if in our dataset a similar problem exists. Thus, we calculate the 5 users with the least reviews:

userId	count
13575	11
18119	10
18608	10
39207	10
57911	10

In our dataset, each user has at least 10 reviews, which means that we do not face the cold start problem.

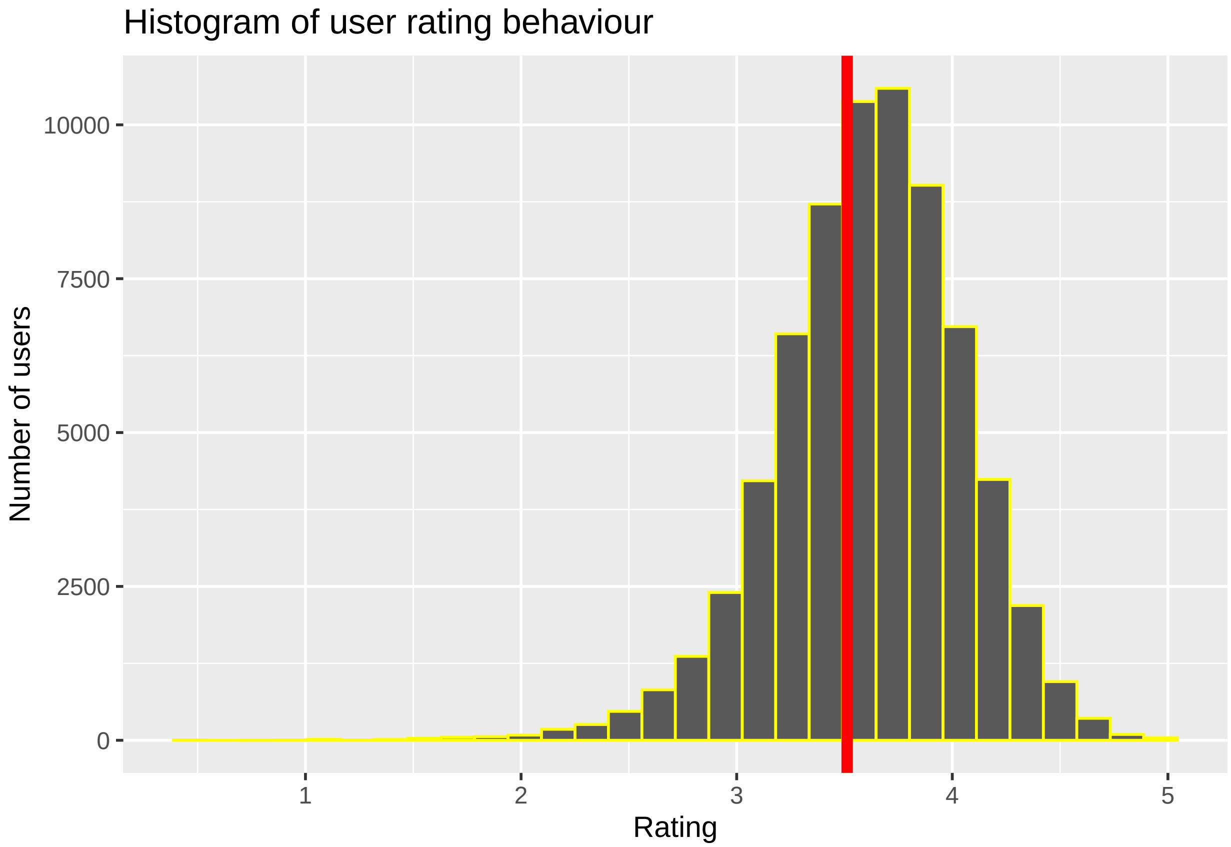
We saw earlier that few users have a very high number of reviews and we should test if these users affect the general dataset. Do they capture a big percentage of the overall ratings (as a whole), or do we have balance in our dataset. To answer this question, we are going to create a pareto chart with the number of users and the total ratings of these users:



From the Pareto chart, we can see that 10.000 users (14% of userbase) are responsible for 50% of the ratings, while 60% of the users are responsible for 80% of the ratings. In theory, Pareto rule is often referred to as thee 80:20 rule (80% of ratings done by 20% of users, and 20% of ratings done by 80% of the users); in our case, the pareto rule does not seem to hold true, and we have a more balanced dataset.

We will now examine how rating is changing for different users.

We begin by plotting a histogram of the average rating grouped by user along with the overall average rating for all users (red line).



Based on the plot, we can safely conclude that different users have different behaviours in terms of rating.

We will now examine how individual users behave, to further illustrate our previous conclusion. We randomly select three users (one with less than 20 ratings (light user), one with more than 100 but less than 500 (medium user) and one with more than 2000 ratings (heavy user)). For each of the user, we plot their unique rating distribution and compare them with each other.

Our first user with more than 2000 ratings has a userId of:

```
[1] 1860
```

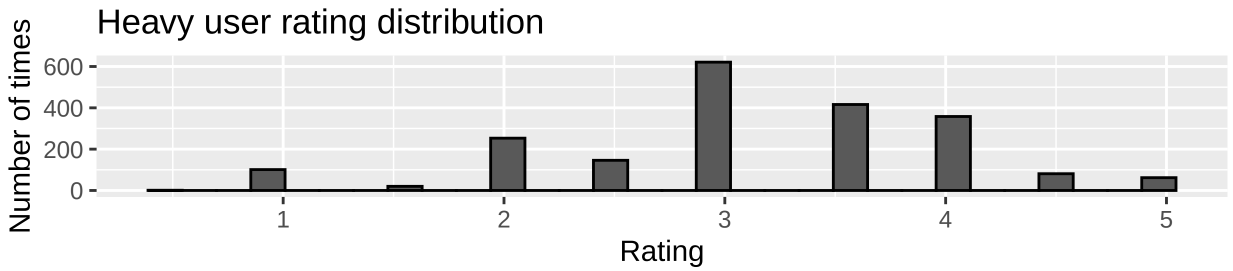
For the medium user, the userId is:

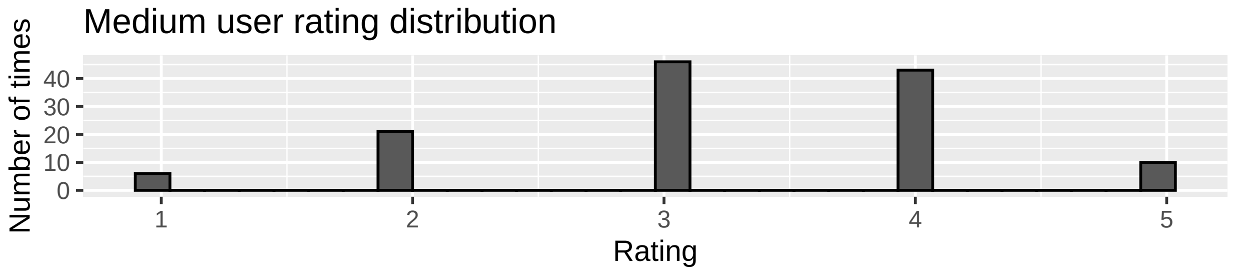
```
[1] 13
```

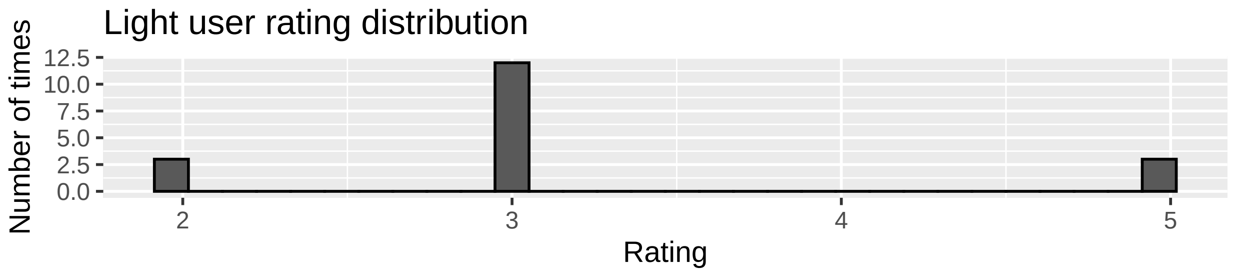
Finally, for the light user (n<20) the userId is:

```
[1] 2
```

Their rating distribution is illustrated in the following plot:







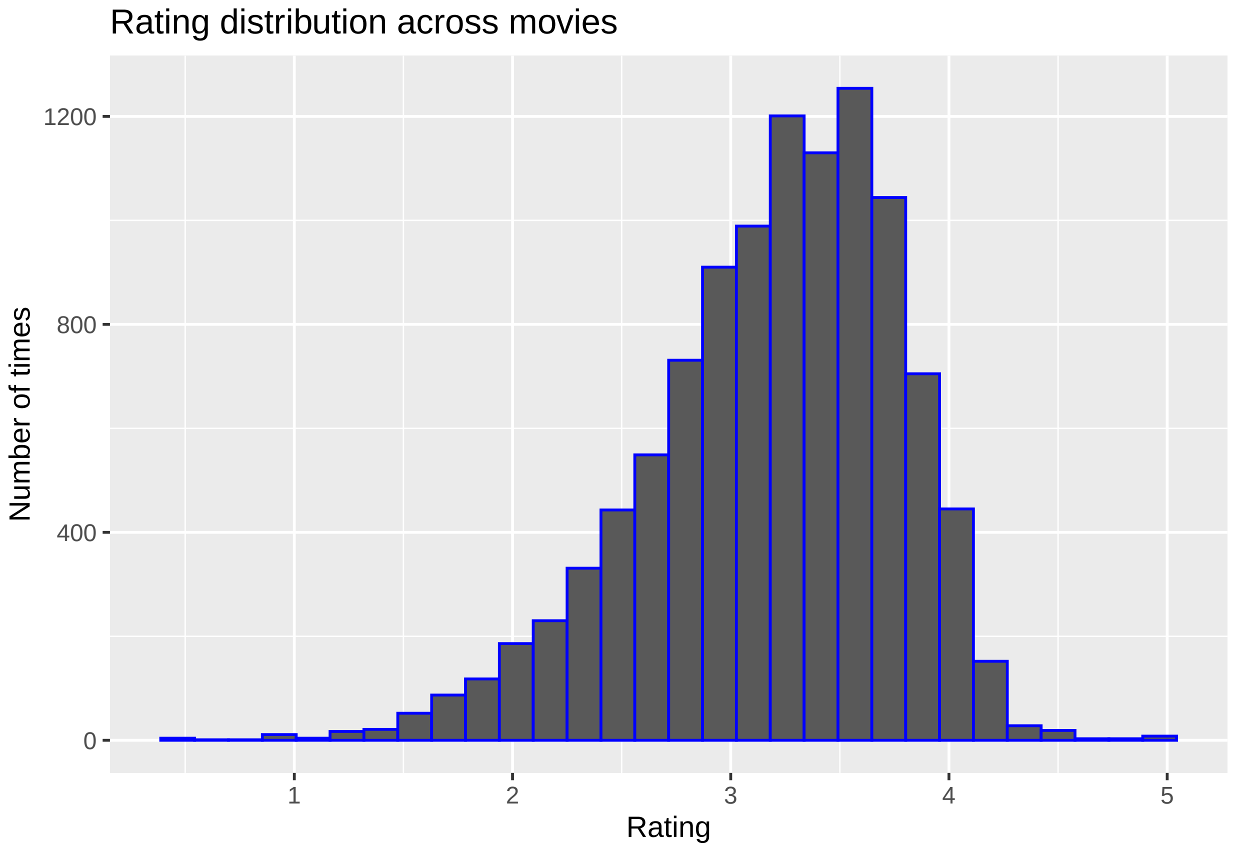
Based on the plots, we observe that these users behave differently (this is based on the seed; we have set the seed to 1 for the dataset creation, for reproducability); the light user appears to be neutral (or slightly positive), with his ratings being mainly 3, while also giving some perfect scores and some low scores. The medium user seems to have a more diverse rating behaviour (although on average he seems more neutral than positive or negative). Finally, the heavy user is positive, with very few ratings below 3.

Therefore, we conclude that there is definately a relationship between the user and the ratings, and thus it will be tested in our models.

4.3 Movie analysis

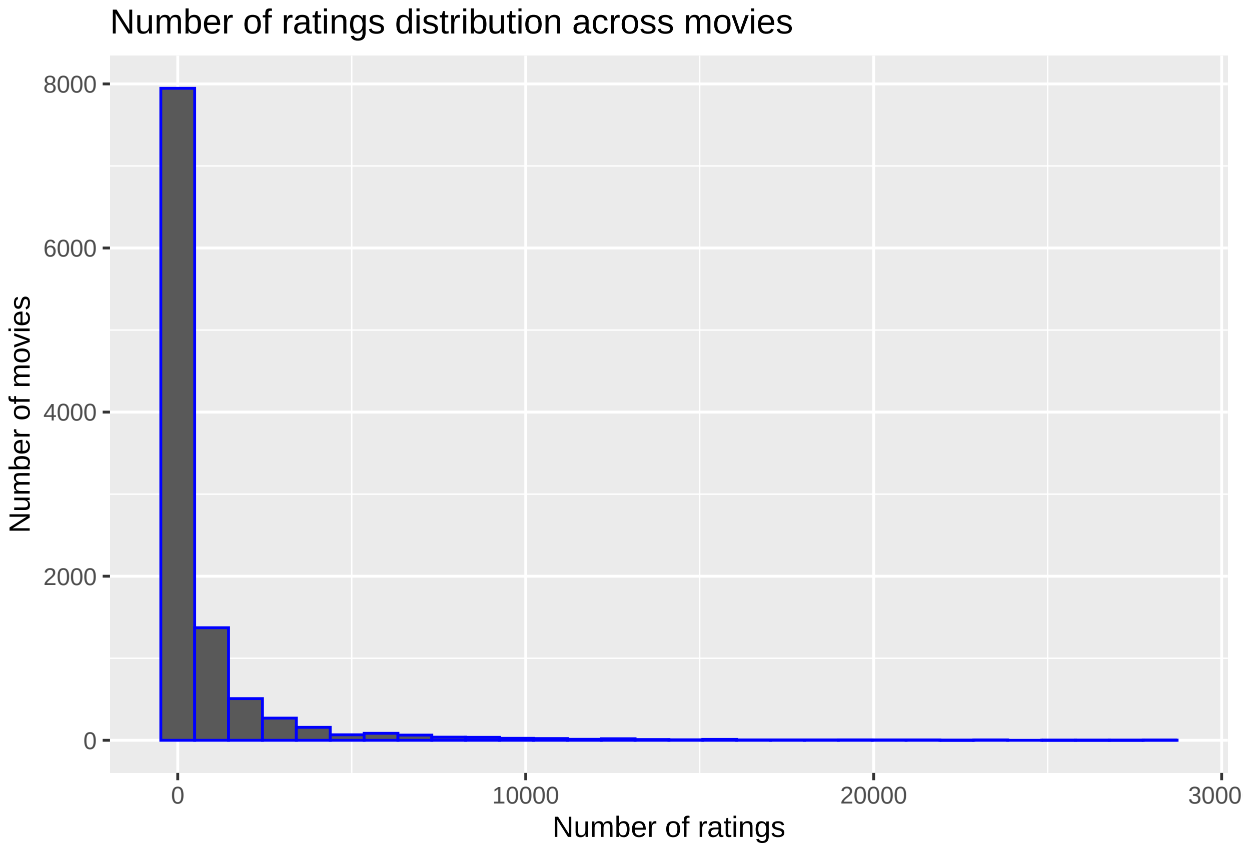
In this section, our objective is to analyze the relationship between movies and rating. From a logical point of view, we expect the movies to have a significant effect on the rating they receive; good quality movies should score higher than low quality movies.

To begin our exploration,we create a histogram of the average rating across movies.



Based on the histogram, we can see a wide distribution of ratings for the different movies. Intuitively, we know that some movies are more popular and some are less popular, so the rating distribution is aligned with our expectactions.

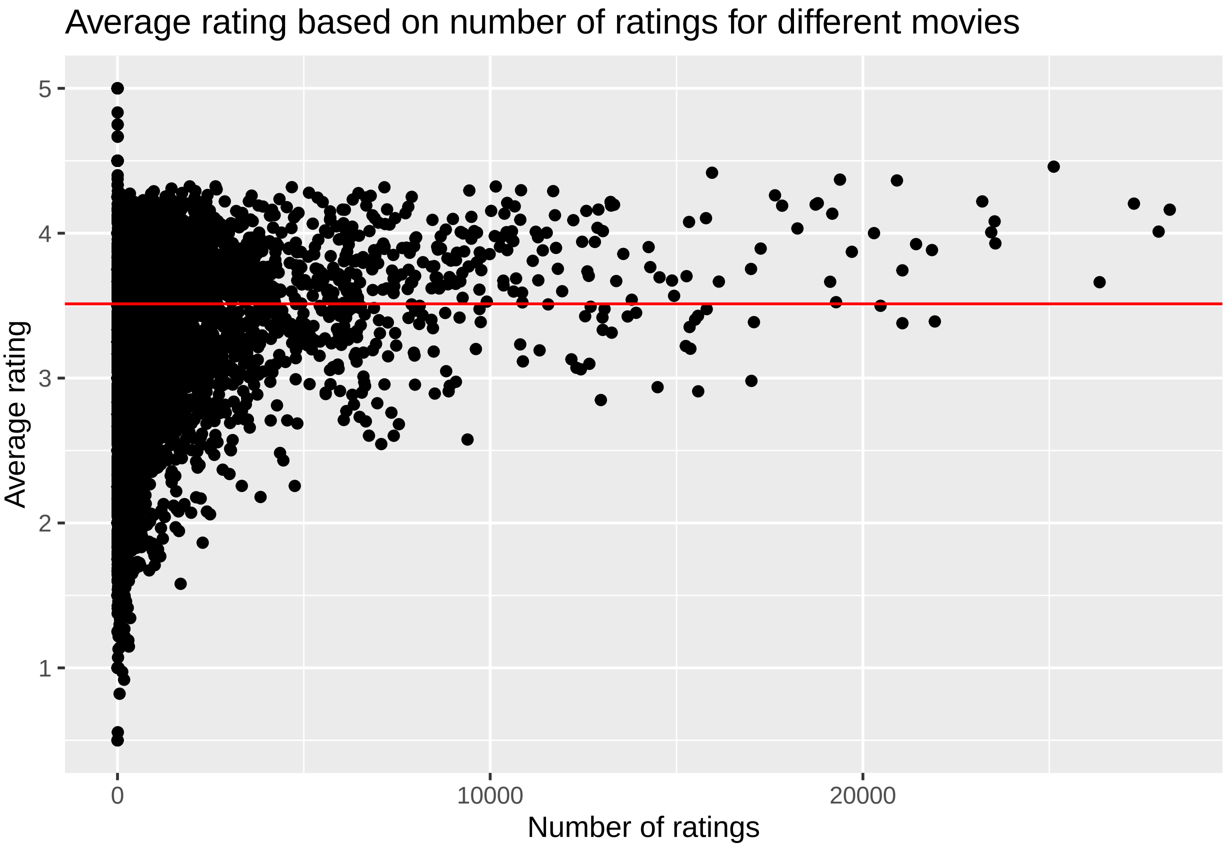
However, we should also examine the impact of the number of ratings to the rating distribution; are movies with a high number of ratings behaving similarly to a less mainstream movie? To answer this question, we start by plotting the histogram of the number of ratings, grouped by movie:



It appears that the majority of movies receive (relatively) few ratings, but there are some movies with more than 20.000 ratings. However, from this histogram, we cannot observe what exactly is happening at movies with a small number of ratings. For that reason, we create a table with the movies with the least number of ratings:

movieId	count
3226	1
3234	1
3290	1
3356	1
3460	1

It seems that there are movies with just 1 rating. Such ratings are unlikely to be reliable. We can expect that such movies have a very narrow and specific target audience and people that watch them are biased in their rating (positively or negatively). To examine this, we are going to plot the average rating every movie has received against the number of ratings it has received. In the same graph, we are also plotting the average rating for all movies (red line).

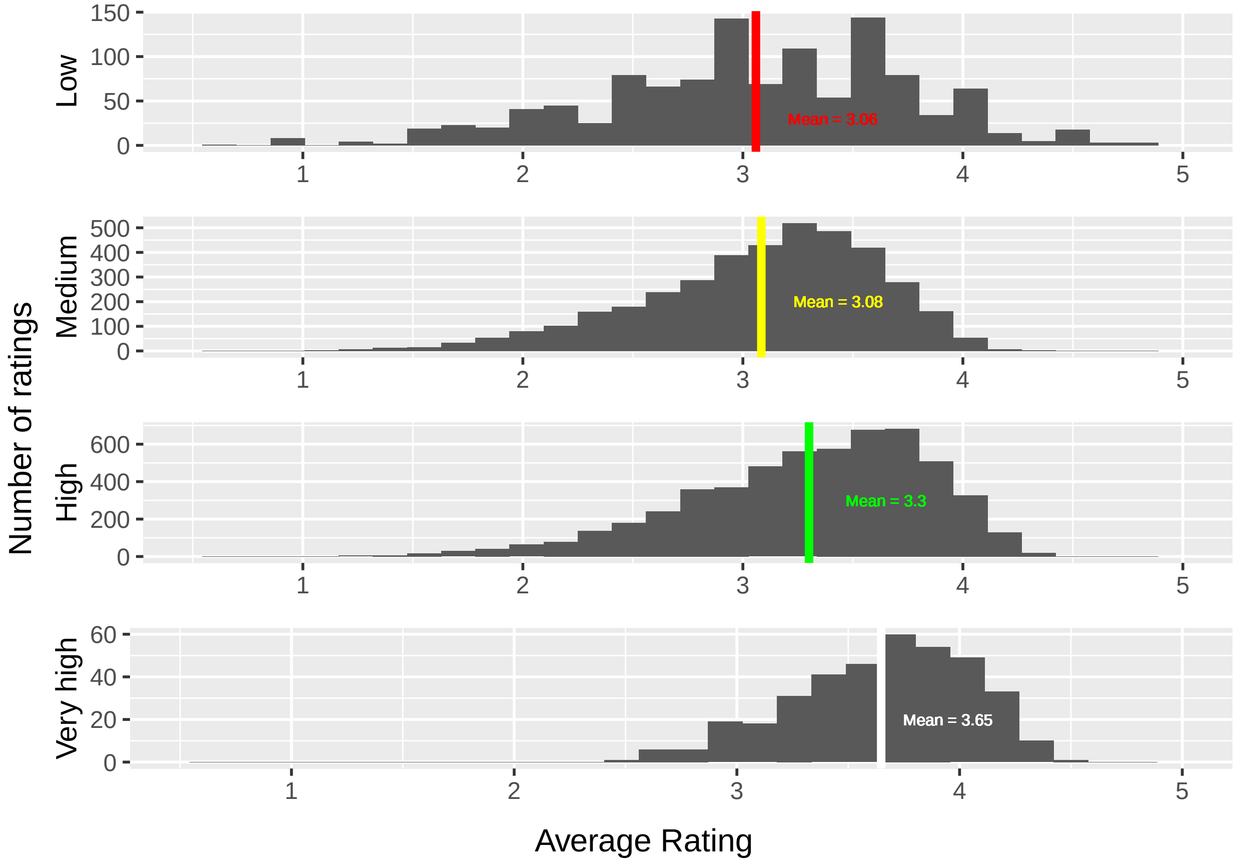


Based on the graph, there is evidence that:

- Movies with **more ratings** (more people have seen them and rated them) appear to have an overall higher rating than the rest.In this case, we are probably talking about blockbuster movies (hence the number of ratings), which is why they are generally scoring higher than average.
- Movies with **very few ratings** can have very good or very bad average ratings (range is [0.5,5]).Additionally, the spread of ratings is much larger in movies with a low number of ratings than it is in those with a higher number.
- The distribution of rating is more wide at low numbers of ratings and becomes more narrow as we move to higher numbers. This is intuitive, as when the number of ratings increases, the average rating of that movie is more representative of the movie's appeal to the public (larger sample).

Continuing our analysis, we are going to explore separately the characteristics of ratings in movies in the low, medium high and very high categories. We define the categories as follows:

- Low: Less than 10 ratings.
- Medium: Less than 100 ratings, more than 10 ratings.
- High: More than 100 ratings.
- Very high: More than 5000 ratings.



In the plot we can clearly observe that when the number of ratings increases the mean value also increases. Simultaneously, the higher the number of ratings, the more close to each other the average ratings are. In the graph, we can see that the distribution is more wide in the upper chart (low,med) and becomes more narrow the further we move down.

Based on this analysis, the movie, as expected, appears to have a strong effect on the rating prediction, and thus it will be included in the model (data analysis section) to test if that is indeed the case.

4.4 Genre Analysis

In this section, we plan to explore the effect of the genres to a movie's rating. Since the genres in our dataset consist of a combination of unique genre categories, they can be either broken down to their individual components or each combination can be treated as a unique category.We will be exploring both options.

For the first part of the analysis, will be treating each genre combination as a unique genre. The total number of unique genres is:

```
[1] 797
```

We begin with some interesting statistics about the genres. Since treating each unique combination as a unique genre seggregates the data to a large extent, we want to see if there are still sufficient data points in every genre.

Firstly, we print the genres with the most data points:

genres	n
Drama	659893
Comedy	630959
Comedy|Romance	329379
Comedy|Drama	291356
Comedy|Drama|Romance	234871

Secondly, we also print the genres with the least data points:

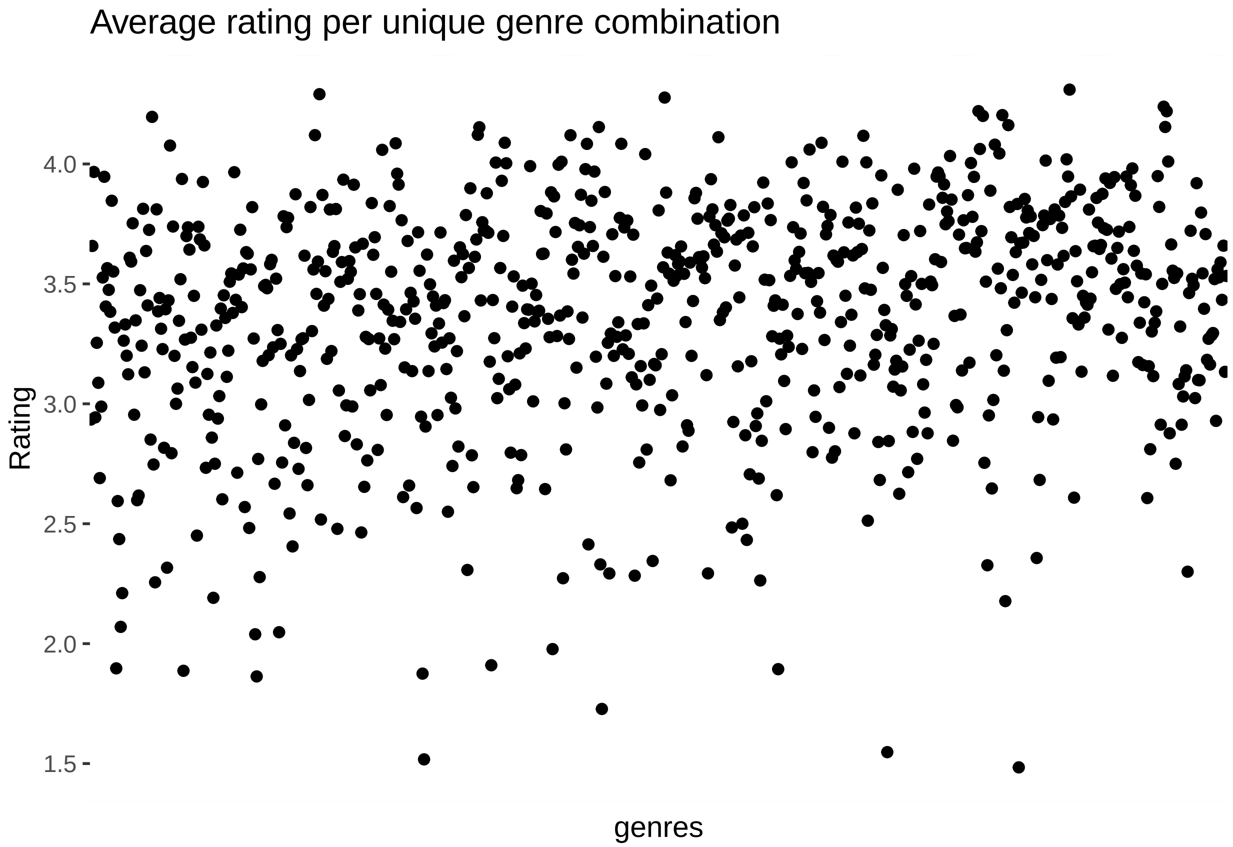
genres	n
Action|Animation|Comedy|Horror	1
Action|War|Western	2
Adventure|Fantasy|Film-Noir|Mystery|Sci-Fi	2
Adventure|Mystery	2
Documentary|Romance	2

It appears that many genres have a sufficient number of data points, but there are also genres that are only appear once or twice. If genre does have an effect on rating, then including genres with very few data points, might affect our analysis. We therefore decided to consider only genres with more than 10 data points.

After removing genres with fewer than 10 data points, the remaining genres are:

```
[1] 761
```

We will now investigate the relationship between genre and rating. We start by plotting the average rating for all the remaining genres (more than 10 data points).



Based on the graph, genres seem to have an impact on the rating given to them. The ones included have a range between 2 and 4.5, for the average rating within the genre, and a very wide distribution amongst different genres.

However, we also want to investigate the actual unique genres and how they affect the rating to a movie. We will attempt to extract the different (individual) components and investigate if we observe similar results in that case too.

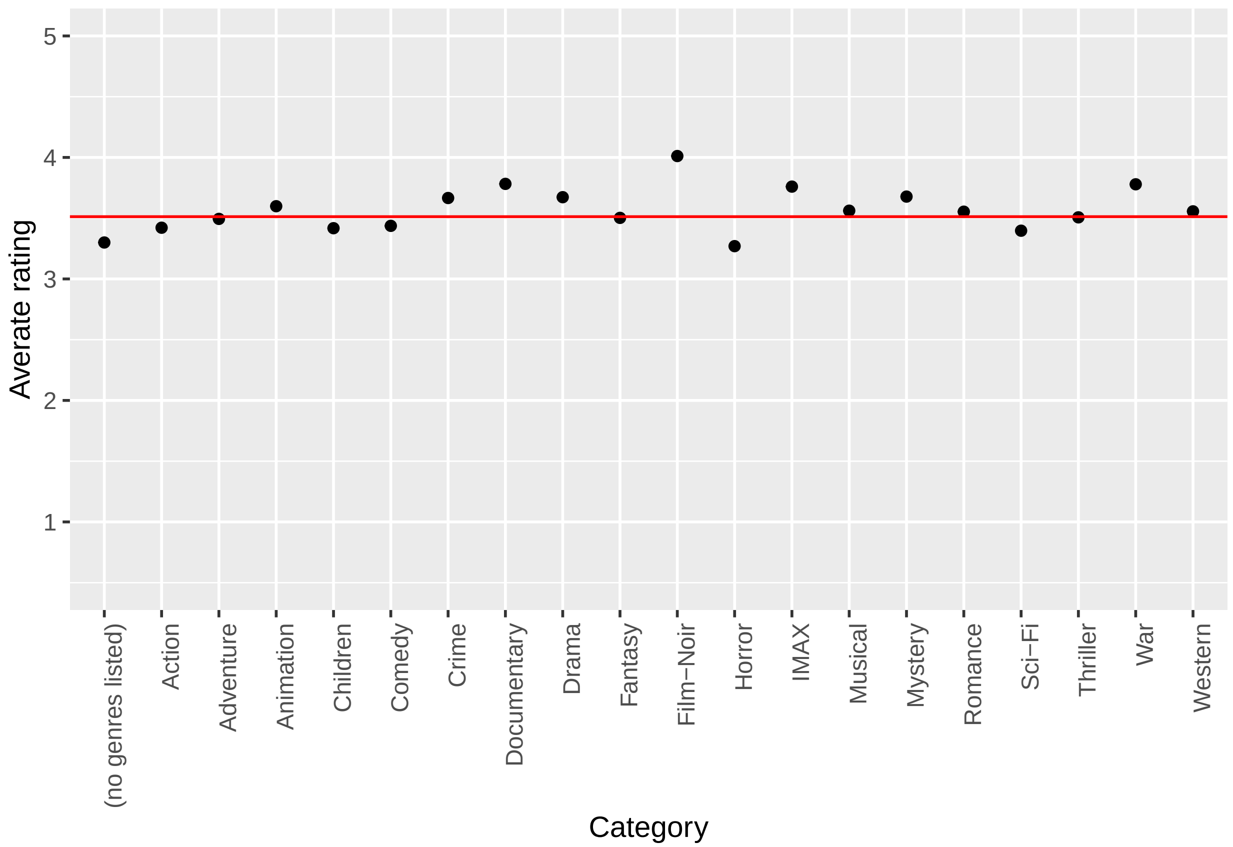
We start by splitting the genres to their individual components and extracting those components:

```
 [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)"
```

The total number of unique genres is:

```
[1] 20
```

We will now calculate the average rating per genre component to see if they are significantly different from each other:

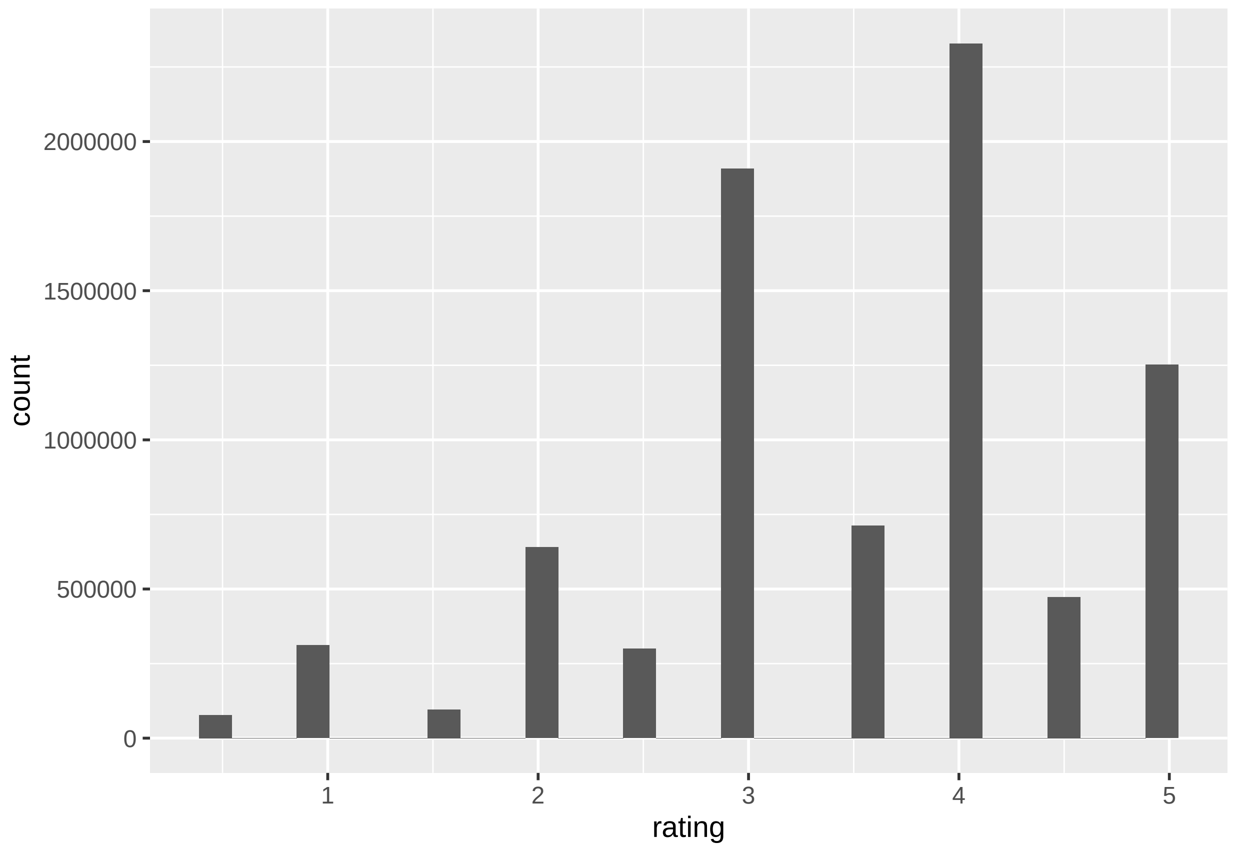


We observe that different genres have different average ratings but the range of those ratings is not as extreme as it was when the combined genre was treated as a unique category, and thus it might not be as effective as the unique genre combinations. Therefore, we will test the genre effect on the movie rating prediction, but we will treat each combination of genres as a unique category (method 1).

4.5 Rating analysis

In this last section of our exploratory data analysis, we will examine how our dependent variable behaves.

For this reason we plot the histogram of rating across all movies and users:



Based on this histogram alone, we can observe two interesting features of the rating:

- The rating given by users is in multiples of 0.5 and not continuous.
- The minimum score is 0.5 (and not 0 as expected), while the maximum is 5. This means that a very easy way to improve predictions in the data analysis chapter is that every predicted value, which is less than 0.5, is set to 0.5 and every predicted value above 5 is set to 5.

5 Analysis: Model building and testing

In this chapter, we build on the results of our exploratory data analysis and start developing models to predict the movie ratings, based on different characteristics.

Firstly, we have removed the title field from our dataset, as we could not identify a clear usecase for including it.

For the accuracy metric, we will be using the root mean squared error (RMSE).

5.1 Model 1: Naive prediction

For our first model, we are going to use a naive rule to generate predictions: every movie gets rated equal to the average rating of all movies. We first calculate the average rating and assign it to the variable mu (will be used in other models too) and then calculate the RMSE of our naive model as:

$\hat{Y} = mu$

The RMSE of our model is:

```
[1] 1.059
```

5.2 Model 2: Adding a time bias to the model

For our second model, we are going to include the time-bias, we observed in our EDA. For this, we first mutate the timestamp into a year variable and then fit a loess line to smoothen it. The time-bias is the smoothened value of the loess line, for each year. To utilize it in our model, we calculate our prediction as follows:

$\hat{Y_t} = mu + f(d_t)$, in which $f(d_t)$ is the smoothened curve fitted in the date data.

In the model, for each prediction we need to generate, we subtract from the mean value the time bias (which is calculated based on the date the rating was assigned).

The RMSE of our 2nd model is:

```
[1] 1.058529
```

Compared to our first model, We observe a very small improvement of 0.5%. We should note that for movies with years that do not appear in the dataset, we are setting the time bias to 0, as we cannot measure it. In the following table, all models and their errors are summarized:

method	RMSE
Naive	1.059000
Time bias	1.058529

5.3 Model 3: Adding a movie bias to the model

For our third model, we are going to build on model 2 and add the movie effect, which appeared to be quite strong in our EDA. Therefore, we begin by calculating the movie bias in our train set. The movie bias for each movie i is the average of all ratings for that movie minus the time bias of each of those ratings.

We calculate our predictions as follows:

$\hat{Y_{t,i}} = mu + f(d_t) + b_i$

The RMSE of our new model is:

```
[1] 0.9428478
```

Compared to our previous model, We observe a substantial improvement of 10.9%. We should note that for new movies (no reviews on those movies) that do not appear in the dataset, we are setting the movie bias to 0, as we cannot measure it. In the following table, all models and their errors are summarized:

method	RMSE
Naive	1.0590002
Time bias	1.0585292
Time bias+Movie bias	0.9428478

5.4 Model 4: Adding a user-bias to the model

For our fourth model, since model 3 was superior to the rest (and therefore the time bias was beneficial for prediction), we are going to build on top of it and add a user bias. To calculate the user bias, for each user we will be computing the average rating after subtracting the time and movie bias that we previously calculated.

We calculate our predictions as follows:

$\hat{Y_{t,i,u}} = mu + f(d_t) + b_i + b_u$

The RMSE of our new model is:

```
[1] 0.8650181
```

Compared to the third model, we observe an improvement of 8.2%. We should note that for new users (no reviews yet) that do not appear in the dataset, we are setting the user bias to 0, as we cannot measure it. In the following table, all models and their errors are summarized:

method	RMSE
Naive	1.0590002
Time bias	1.0585292
Time bias+Movie bias	0.9428478
Time bias + Movie bias + User bias	0.8650181

5.5 Model 5: Adding the genre bias to the model

For our last model, we are again going to build on the best model so far: Model 4. We are now going to add the genre effect to our existing model. At this stage, as a reminder, we have decided at the end of the EDA, to treat each genre combination as a unique category. To calculate the genre bias, first we are filtering only the genres that have more than 10 observations. Subsequently, we are calculating the genre bias as the average rating per user after subtracting the time, movie and user biases.

We calculate our predictions as follows:

$\hat{Y_{t,i,u,g}} = mu + f(d_t) + b_i + b_u + b_g$

The RMSE of our new model is:

```
[1] 0.8631007
```

We observe a small improvement of 0.2%. We should note that for new genres that do not appear in the dataset, we are setting the genre bias to 0, as we cannot measure it. In the following table, all models and their errors are summarized:

method	RMSE
Naive	1.0590002
Time bias	1.0585292
Time bias+Movie bias	0.9428478
Time bias + Movie bias + User bias	0.8650181
Time bias + Movie bias + User bias + Genre bias	0.8631007

6 Results: Model assessment & validation

In this section, we will be using the best model of the previous chapter to validate its performance in the validation set.

The best performing model of chapter 5 is the one with all 4 parameters (time bias, user bias, movie bias and genre bias). To calculate the final RMSE, first will be joining the validation set with the four bias tables we generated in the train set and then predict based on the following formula:

$\hat{Y_{t,i,u,g}} = mu + f(d_t) + b_i + b_u + b_g$

Note that for any unobserved years, users, movies or genres we set the appropriate bias to 0.

Finally, we also improve our predictions with the rule defined in the Rating Analysis of chapter 4: if a rating is below 0.5 we set it to 0.5 and if its above 5, we set it to 5.

The validation RMSE of our model is:

```
[1] 0.8647722
```

The RMSE of our model appears to be in line with what was tested in previous sections and has improved by 20% over our initial estimation (naive model).

7 Conclusion

In this chapter, we will be discussing our methods, models and results, explain some of the limitations of our research and conclude this project with some directions for future research.

7.1 Discussion of results

In chapter 5, we have developed five models that incorporate different factors into the design. More specifically, we have studied the effects of movie-bias, user-bias, time-bias and genre-bias on the rating prediction. Each of the four factors appears to be beneficial in predicting the final rating, and the factors were studied on an additive basis: each factor was added to the best current model and the new model's performance was benchmarked against the original. If an improvement was found, the factor was considered beneficial and it was utilized in all subsequent models.

From our analysis, we can see that the variable, which had the biggest impact in the RMSE reduction, was the movie-bias. From an empirical point of view, we can see why this is the case: the rating of the moving is depending on the actual movie; if a movie is of bad quality, this is reflected on the rating and vice versa. On the other hand, the least effective variable was the time bias. While it is true that some periods are associated with better quality movies, this effect is very minor (slightly better than the naive model). We should note that from a percentage (%) improvement point of view, the weakest variable is the genre-bias; however, this was the last variable added to the model, and thus a lot of variability was already accounted for by the other three factors. For this reason, we are considering the improvement done to RMSE in absolute terms ($Improvement = RMSE_{old} - RMSE_{new}$).

Overall, the best performing model had a substantial improvement (20%) over our first benchmark model and predicts with high accuracy the rating in the validation set (RMSE=0.8647).

7.2 Limitations

The algorithm and research presented, yielded some interesting results but are not without limitations. To begin with, for estimating the different types of biases, we calculated the average (per user/movie/year/genre); however, we could potentially use alternative ways to estimate better values for them. Furthermore, due to the extreme computational times, no sophisticated machine learning algorithms were used. A solution for this problem could be a virtual machine, which was not utilized in this research. Finally, for the genre effects, in the case of individual genres (which was deemed less beneficial for modelling purposes) the explicit relationship between genres was not studied.

7.3 Further research

In this project, several methods for creating a movie recommendation system have been studied, with decent results. In future research, new methods for estimating the different biases should be examined and benchmarked against the original models. Additionally, we propose a more in-depth investigation of the genre effect, to identify some overlaps between genre components. Finally, the utilization of virtual machines and more intelligent ML methods can also be beneficial for building more advanced recommendation systems.

References

Adomavicius, Gediminas, and Alexander Tuzhilin. 2005. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." *IEEE Transactions on Knowledge and Data Engineering* 17 (6). IEEE: 734–49.

Isinkaye, FO, YO Folajimi, and BA Ojokoh. 2015. "Recommendation Systems: Principles, Methods and Evaluation." *Egyptian Informatics Journal* 16 (3). Elsevier: 261–73.

LYLE, BOBBY B. 2012. "Recommender Systems."

Pazzani, Michael J, and Daniel Billsus. 2007. "Content-Based Recommendation Systems." In *The Adaptive Web*, 325–41. Springer.

Rafsanjani, Amir Hossein Nabizadeh, Naomie Salim, Atae Rezaei Aghdam, and Karamollah Bagheri Fard. 2013. "Recommendation Systems: A Review." *International Journal of Computational Engineering Research* 3 (5): 47–52.

Sánchez, José Luis Sánchez. 2013. "Improving Collaborative Filtering Based Recommender Systems Using Pareto Dominance." PhD thesis, Universidad Politécnica de Madrid.

Thorat, Poonam B, RM Goudar, and Sunita Barve. 2015. "Survey on Collaborative Filtering, Content-Based Filtering and Hybrid Recommendation System." *International Journal of Computer Applications* 110 (4). Foundation of Computer Science: 31–36.