A movie rating prediction model

Fabio A Oliveira

21/09/2020

Contents

Introduction

This report describes the construction of a statistical model that predicts movie ratings, based on the *MovieLens 10M* dataset provided by the GroupLens research lab at the University of Minnesota. This version of the dataset has been released in 2009 and contains over 10 million ratings given by around 70 thousand individual users to over 10 thousand different movies, with each entry containing identification numbers for the user and movie, as well as the movie title and release year, its genres and a timestamp identifying when the rating was given. The dataset has been used regularly by the Data Science community in teaching and in the development of machine learning models.

Before attempting to construct any statistical model, some basic data wrangling is performed, followed by an exploratory data analysis, in order to identify trends in the data that may be later used to construct predictions.

An additive approach has been used to construct the final model, in the sense that a number of increasingly complex models has been created so that, at each step, an additional term accounts for patterns identified in the residuals from the previous step. With this approach, the variable under evaluation at each step can be analyzed individually, and the correlation between different predictors is not a point of concern.

For the purposes of this project, the full 10M dataset has been partitioned into three sets:

1. A *train* set, from where patterns are identified and quantified in order to build the statistical model;
2. A *test* set, where different parameters are tested and tuned in order to find optimal performance and
3. A *validation* set, which will not be used for training or tuning of parameters, and where the models will be evaluated.

The ultimate goal of the project is to create a model capable of predicting ratings in the *validation* set as precisely as practical. The *Root Mean Squared Error* (RMSE) of the predictions compared to the actual ratings is reported as a performance metric. The RMSE obtained after application of the complete model to the *validation* set was 0.843.

A Shiny Web App with a minimalist version of the final model has been created for illustrative purposes and can be accessed at https://fabio-a-oliveira.shinyapps.io/MovieRecommenderApp/.

The GitHub page for this project is https://github.com/fabio-a-oliveira/movie-recommendations.

This analysis is part of the capstone project for the Data Science Professional Certificate offered by HarvardX and hosted on edX. More information can be found at https://www.edx.org/professional-certificate/harvardx-data-science.

Methods/analyses

The project has been developed according to a sequence of steps:

1. Data preparation: during this step, the MovieLens 10M dataset is downloaded programmaticaly from the GroupLens website and partitioned into the *train*, *test*, and *validation* sets. Care is taken to ensure that every user and movie are represented in the *train* set (so that the model is not required to make predictions for users and movies to which it has not been exposed during training). Separate data frames are created with statistics for each individual users, movies and movie genres, so that information gathered during the following steps can be store and accessed easily. Empty data frames are also created to store functions to apply each of the models, its predictions for each of the datasets and the results.

2. Exploratory data analysis: during this step, the *train* set is explored to identify relevant features, how the different variables are related, and which characteristics can potentially be used during the modeling step in order to predict ratings. This step is also used to get familiarity with the data.
3. Modeling: during this step, increasingly complex models are developed, trained and tested to account for the influence of the factors identified during the exploratory data analysis on the ratings given to movies by each user. These models are created so that the simplest connections are accounted for sooner (how does a single predictor relate to the ratings?), followed by more complex phenomenons (how do combinations of predictors relate to the ratings?). The final model includes an *Item-Based Collaborative Filter* (IBCF), in which the predictions for how each individual user rates each movie takes into consideration how the user has previously rated similar movies.

Each of these steps is explained in more details in the corresponding section of this report.

Additionaly, due to the size of the original dataset used in this analysis, significant challenges related to computing performance were tackled. Constant care was taken to avoid using unnecessary memory. Nevertheless, certain sections of the code required a command to increase the amount of memory available to R with the `memory.limit(size = 15000)` command. During the creation of the model with the IBCF in particular, the calculations required the use of the `Matrix` package for dealing with sparse matrices and the implementation of custom functions to identify the similarity between different movies in a memory-efficient manner.

The R code for this project is divided into two different files. All of the data preparation and the modelling are done in the `HX9_MovieLens_Main.R` script. This script prepares the data and does all of the calculations necessary for the development of the models, creates functions to make predictions according to each model, and calculates the performance for each model. It then saves the resulting objects into files that are loaded by the `HX9_MovieLens_Report.RMD` script (the R Markdown script that generates this report) to create all the tables and graphics in this report.

Data preparation

A series of adjustments to the original *MovieLens 10M* dataset is necessary before we begin analysis and modeling. These adjustments are described below:

1. The original data is partitioned into the *train*, *test*, and *validation* datasets. As a first step, the original data is partitioned into the *edx* and *validation* data frames at a 0.9/0.1 proportion. The *edx* object is then partitioned into the *train* and *test* sets at a 0.8/0.2 proportion.

The table below shows how the full set of entries from the *MovieLens 10M* dataset is distributed among the three different sets used in this project:

Table 1: Entries in the different sets

Object	Number of Ratings (% of total)	Number of Users (% of total)	Number of Movies (% of total)
train	7200079 (72%)	7200079 (100%)	7200079 (100%)
test	1799976 (18%)	1799976 (100%)	1799976 (95%)
validation	999999 (10%)	999999 (98%)	999999 (92%)

It is relevant to notice that the *train* set contains entries for all the individual users and movies, so the different models are not required to predict ratings for users and movies with which they have had not previous training.

A glimpse at a sample of these objects reveals what information is available and what further adjustments are necessary:

```
##     userId movieId rating  timestamp                          title              genres
##  1:   8811     410    2.0 1027118151  Addams Family Values (1993)              Comedy
##  2:  57106     831    4.0 1016304071             Stonewall (1995)               Drama
##  3:  57675     454    4.0  978592829             Firm, The (1993)      Drama|Thriller
##  4:  48211    5450    4.0 1107210341      Lovely & Amazing (2001)              Comedy
##  5:  37866      86    3.0  954738947         White Squall (1996)     Adventure|Drama
##  6:  43108    8464    4.5 1102794229         Super Size Me (2004)         Documentary
##  7:  61986    3251    4.0 1018046761          Agnes of God (1985)       Drama|Mystery
##  8:   3340    5060    4.5 1111622359 M*A*S*H (a.k.a. MASH) (1970)    Comedy|Drama|War
##  9:  11514    2408    3.0 1090066896    Cocoon: The Return (1988)       Comedy|Sci-Fi
## 10:  28839    1256    4.0  992201363             Duck Soup (1933)      Comedy|Musical
## 11:   4494    2052    4.5 1216660069           Hocus Pocus (1993)     Children|Comedy
## 12:  66553    2359    3.5 1160368286      Waking Ned Devine (1998)              Comedy
## 13:  60377    5502    2.0 1213631043                 Signs (2002)     Sci-Fi|Thriller
## 14:  52124    1958    4.0  940675530    Terms of Endearment (1983)        Comedy|Drama
## 15:  14675    2064    5.0 1029851647             Roger & Me (1989) Comedy|Documentary
```

After a quick glimpse at the dataset, we see that the `timestamp` column is formatted in a manner that is difficult to read. The `title` column contains both the movie title and its year of release. Additionally, the `genres` column contains all the genres associated with the movie, separated by a "|" character. Some basic data wrangling is necessary to have this information in usable formats. However, in order to preserve these objects as close as possible to the original data, any manipulation and formatting of the data is done in the objects created in the next step.

The choice for doing training and testing with a single split of the *edx* data (as opposed to performing k-fold cross-validation) has been motivated by simplicity and saving time in code execution. With this approach, statistics can be drawn directly from the *train* set and performance for different parameters can be tested directly on the *test* set, without the need to average results over multiple folds. The fact that there are millions of observations in both sets guarantees that the outcomes are robust to random variability and mitigate the risk of overtraining.

2. Objects are created to hold data pertinent to individual entries on the datasets Several R objects are created to hold relevant information that will be used for analysis, modeling and holding results during the project. Basic summary statistics and some necessary data wrangling identified in the analysis of the *train*, *test*, and *validation* sets are also performed and saved to the new objects.

- The `users` object contains a summary with the average rating and number of ratings for each individual user in the form of a data frame;

```
##   userId average rating number of ratings
## 1  56286       3.250000                44
## 2  28452       3.186747               249
## 3  66861       3.905263                95
## 4    547       3.588235                34
## 5  58092       3.395349                43
## 6  37830       3.866667                60
```

- The `movies` object contains a summary with the average rating and number of ratings for each individual movie in the form of a data frame;

```
##   movieId              title year         genres average rating number of ratings
## 1     470      House Party 3 1994         Comedy       2.209677               186
## 2   25947 Unfaithfully Yours 1948         Comedy       4.044118                34
## 3     639             Girl 6 1996   Comedy|Drama       2.942241               580
## 4    5020     Silent Trigger 1996   Action|Drama       2.800000                 5
## 5    5175      Hidden Agenda 1990 Drama|Thriller       3.027778                18
## 6    2957           Sparrows 1926          Drama       3.000000                 9
```

- The `genres` object contains both a data frame with each individual movie genre associated with each movie (in "long" format) and a data frame with columns for each movie genre indicating whether each movie is associated with each particular movie (in "wide" format), in the form of a list;

```
##                          genres    genre
## 1                Comedy|Romance   Comedy
## 2                Comedy|Romance  Romance
## 3         Action|Crime|Thriller   Action
## 4         Action|Crime|Thriller    Crime
## 5         Action|Crime|Thriller Thriller
## 6  Action|Drama|Sci-Fi|Thriller   Action
## 7  Action|Drama|Sci-Fi|Thriller    Drama
## 8  Action|Drama|Sci-Fi|Thriller   Sci-Fi
## 9  Action|Drama|Sci-Fi|Thriller Thriller
## 10      Action|Adventure|Sci-Fi   Action
```

```
##                          genres Comedy Romance Action Crime Thriller Drama Sci-Fi
## 1                Comedy|Romance   TRUE    TRUE  FALSE FALSE    FALSE FALSE  FALSE
## 2         Action|Crime|Thriller  FALSE   FALSE   TRUE  TRUE     TRUE FALSE  FALSE
## 3  Action|Drama|Sci-Fi|Thriller  FALSE   FALSE   TRUE FALSE     TRUE  TRUE   TRUE
## 4       Action|Adventure|Sci-Fi  FALSE   FALSE   TRUE FALSE    FALSE FALSE   TRUE
## 5       Children|Comedy|Fantasy   TRUE   FALSE  FALSE FALSE    FALSE FALSE  FALSE
## 6      Comedy|Drama|Romance|War   TRUE    TRUE  FALSE FALSE    FALSE  TRUE  FALSE
## 7                 Action|Comedy   TRUE   FALSE   TRUE FALSE    FALSE FALSE  FALSE
## 8       Action|Romance|Thriller  FALSE    TRUE   TRUE FALSE     TRUE FALSE  FALSE
## 9  Action|Comedy|Crime|Thriller   TRUE   FALSE   TRUE  TRUE     TRUE FALSE  FALSE
## 10                       Comedy   TRUE   FALSE  FALSE FALSE    FALSE FALSE  FALSE
##    Adventure Children Fantasy   War Animation Musical Western Mystery Horror Film-Noir
## 1      FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 2      FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 3      FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 4       TRUE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 5      FALSE     TRUE    TRUE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 6      FALSE    FALSE   FALSE  TRUE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 7      FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 8      FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 9      FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 10     FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
##    Documentary  IMAX (no genres listed)
## 1        FALSE FALSE              FALSE
## 2        FALSE FALSE              FALSE
## 3        FALSE FALSE              FALSE
## 4        FALSE FALSE              FALSE
## 5        FALSE FALSE              FALSE
## 6        FALSE FALSE              FALSE
## 7        FALSE FALSE              FALSE
## 8        FALSE FALSE              FALSE
## 9        FALSE FALSE              FALSE
## 10       FALSE FALSE              FALSE
```

- The `models` object is an empty list of functions, which will be filled during the modeling phase with functions calculating predictions for each different model;
- The `parameters` object is an empty list, which will contain parameters required for the application of the models calculated during the modeling phase;
- The `results` object is an empty list, which will be filled with results acchieved by each of the models and
- The `predictions` object is a list containing empty data frames that will be filled with the RMSE results obtained by each model on the *train*, *test*, and *validation* sets.

During the creation of the models, additional information pertinent to each entry of these data frames is included.

3. Creation of functions In order to facilitate calculations during the subsequent phases, some R functions are created to perform tasks that are repeated multiple times:

- The `RMSE()` function calculates the Root Mean Squared Error between two supplied vectors;
- The `limitRange()` function introduces a saturation to a supplied vector, ensuring that each entry is within the interval determined by the `min` and `max` arguments and
- The `sparse.colCenter()`, `sparse.colMeans()`, `sparse.colSums()` and `sparse.correlationCommonRows()` functions perform numerous sparse matrix manipulation tasks in a memory-conscious manner and will be described in more details in the section dedicated to the model with collaborative filtering.

Exploratory Data Analysis

Before we tackle the prediction problem, we begin by doing a thorough exploratory data analysis, with the goal of identifying trends and features of the data that might be useful in understanding it and possibly in constructing a prediction algorithm.

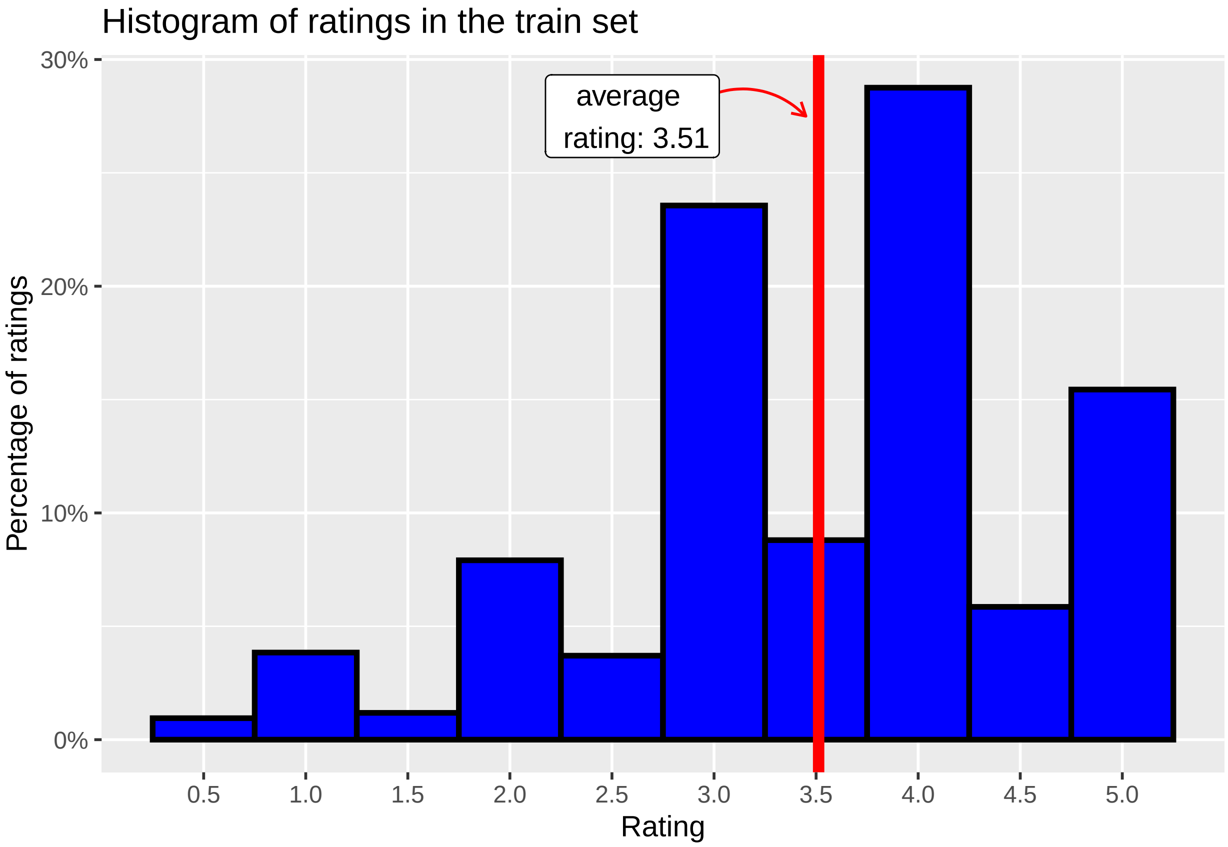
Five aspects will be explored in details in this section:

1. Distribution of ratings
2. Individual movies
3. Individual users
4. Movie genres
5. Movie year of release and date of rating

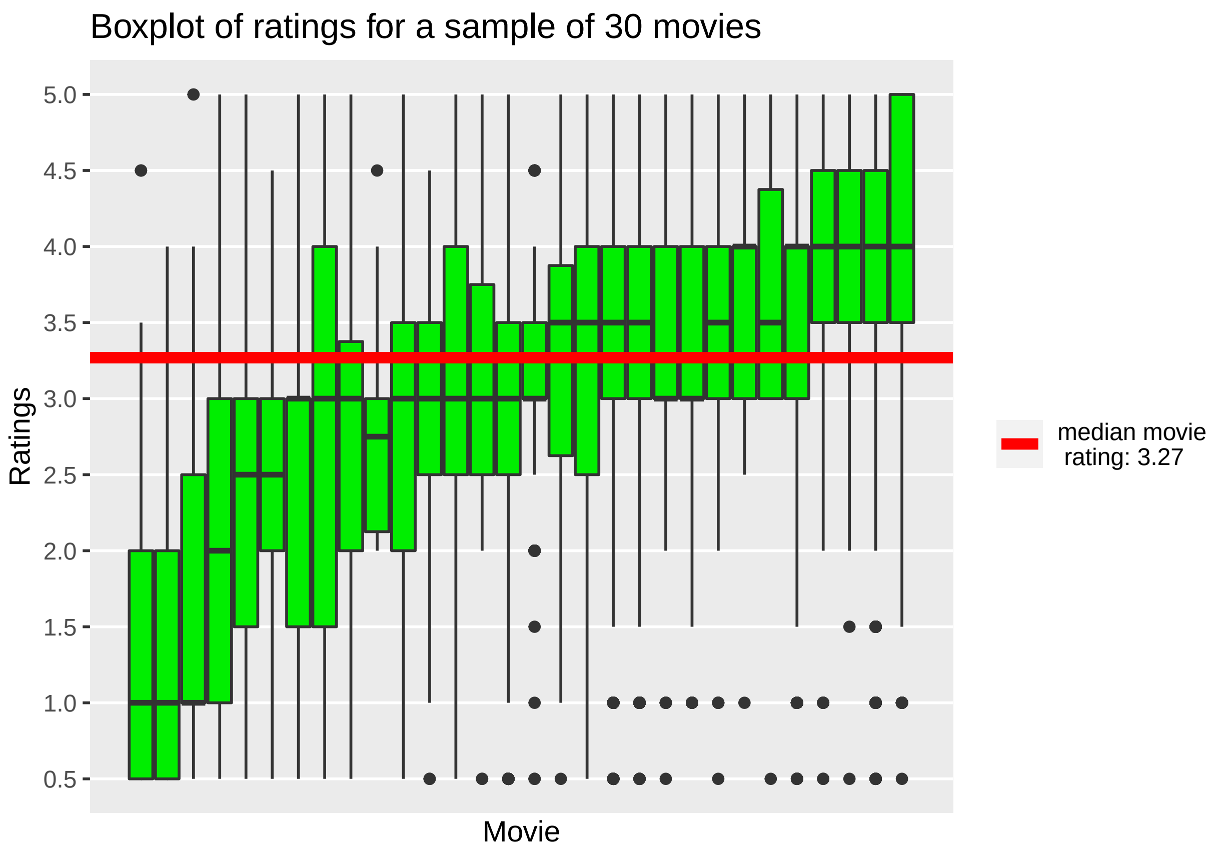
At this stage, no attempt will be made to predict the actual ratings, but trends identified during the exploratory data analysis will be later used as criteria to build prediction models during the modeling section.

Global distribution of ratings

We begin by looking at how the ratings are distributed in the *train* set.



We see that the ratings range from 0.5 to 5.0 in increments of 0.5, which is rather typical of a rating system based on number of stars. The histogram also shows that the most common ratings are 4 and 3, and that most integer ratings are more common than non-integer ratings.



This boxplot of the ratings for 30 randomly selected movies shows that the Inter-Quartile Range (IQR) is typically the interval of +/- 0.5 points around the median. However, the majority of movies do have ratings that cover most of the 0.5 - 5 points scale.

Analysis of individual movies

We begin the exploration of the characteristics of individual movies by looking at the table of the most popular movies, i.e. the ones with the most number of ratings.

Table 2: Top 10 Popular Movies

movieId	Title	Average Rating	Number of Ratings
296	Pulp Fiction	4.16	25005
356	Forrest Gump	4.01	24803
593	Silence of the Lambs, The	4.20	24199
480	Jurassic Park	3.66	23451
318	Shawshank Redemption, The	4.45	22561
110	Braveheart	4.09	20919
589	Terminator 2: Judgment Day	3.93	20866
457	Fugitive, The	4.01	20820
260	Star Wars: Episode IV - A New Hope (a.k.a. Star Wars)	4.22	20581
592	Batman	3.38	19418

It is also interesting to see which movies have the highest average ratings.

Table 3: Top 10 Best Rated Movies

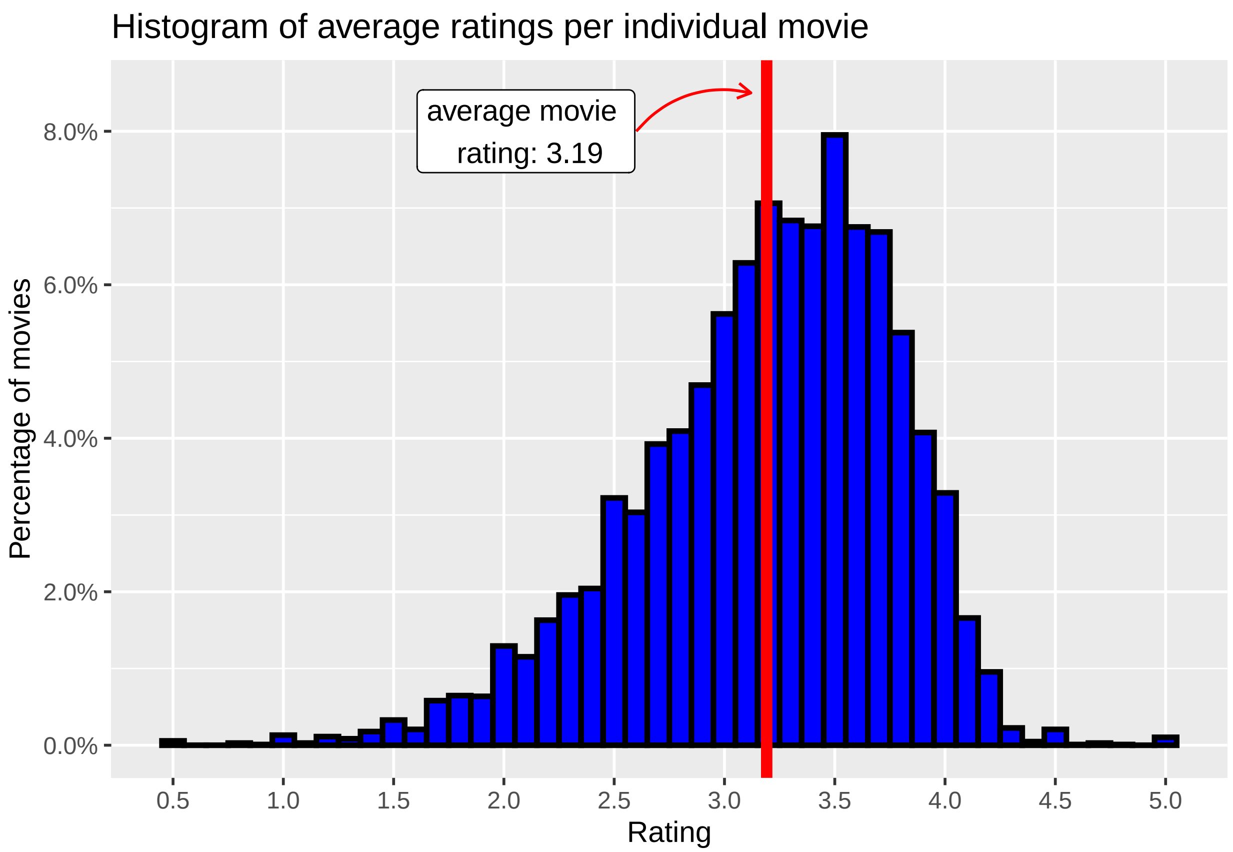
movieId	Title	Average Rating	Number of Ratings
3226	Hellhounds on My Trail	5	1
7447	MC5*: A True Testimonial	5	1
8536	Intended, The	5	1
33264	Satan's Tango (SÃ¡tÃ¡ntangÃ³)	5	1
42783	Shadows of Forgotten Ancestors	5	1
51209	Fighting Elegy (Kenka erejii)	5	1
53355	Sun Alley (Sonnenallee)	5	1
60983	Eve and the Fire Horse	5	1
63808	Class, The (Entre les Murs)	5	2
64197	Hunger	5	1
64275	Blue Light, The (Das Blaue Licht)	5	1

From this previous table, we notice that the movies at the top of the list all have very low number of ratings. That indicates that they high averages are most likely a product of random variability. When we create a list of the top 10 movies arranged by their ratings, but first filter out movies with less then 100 individual ratings, we get a more reasonable result:

Table 4: Top 10 Best Rated Movies (with over 100 ratings)

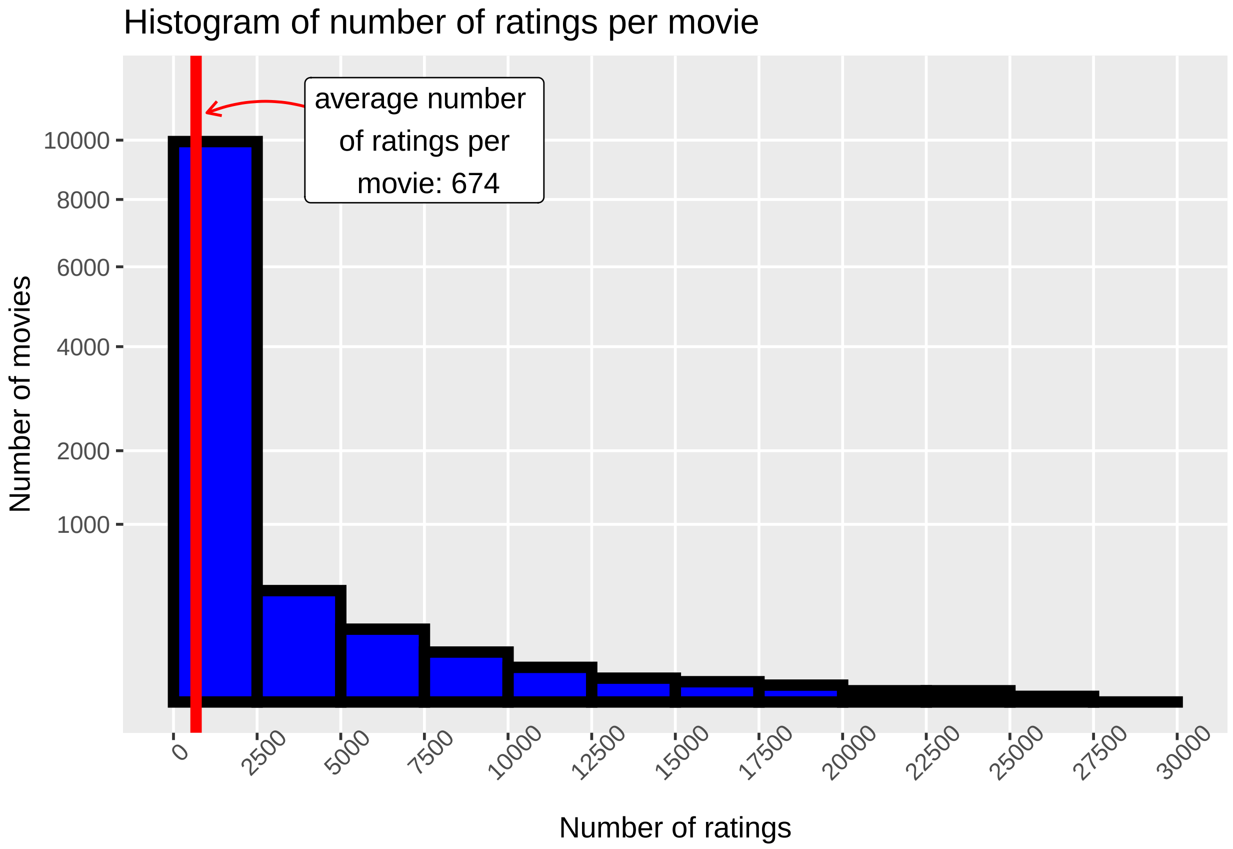
movieId	Title	Average Rating	Number of Ratings
318	Shawshank Redemption, The	4.45	22561
858	Godfather, The	4.41	14122
50	Usual Suspects, The	4.37	17358
527	Schindler's List	4.36	18611
912	Casablanca	4.32	9015
922	Sunset Blvd. (a.k.a. Sunset Boulevard)	4.32	2349
904	Rear Window	4.32	6381
3435	Double Indemnity	4.31	1721
2019	Seven Samurai (Shichinin no samurai)	4.31	4140
1212	Third Man, The	4.31	2390

A histogram of the average movie ratings shows that movie averages are most common in the vicinity of 3.5. However, the overall average is actually around 3.19, which indicates that the data is skewed to the right. This makes sense, since the upper limit of the interval of possible ratings is closer to the average, so movies better rated than the average are more concentrated.

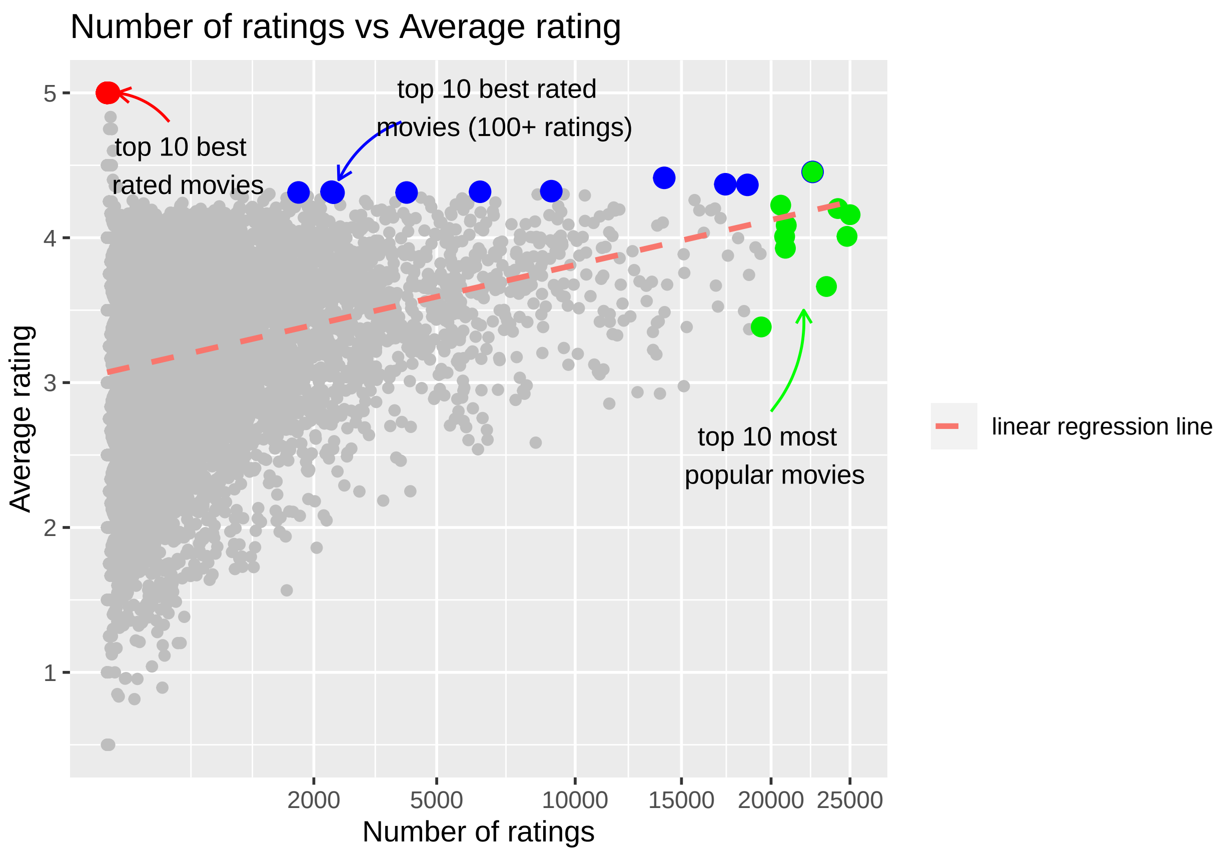


The following histogram shows the distribution of the total number of ratings. We see that there is a large concentration in the low numbers, with an average of 674 reviews per movie, while some have over 20 thousand ratings.

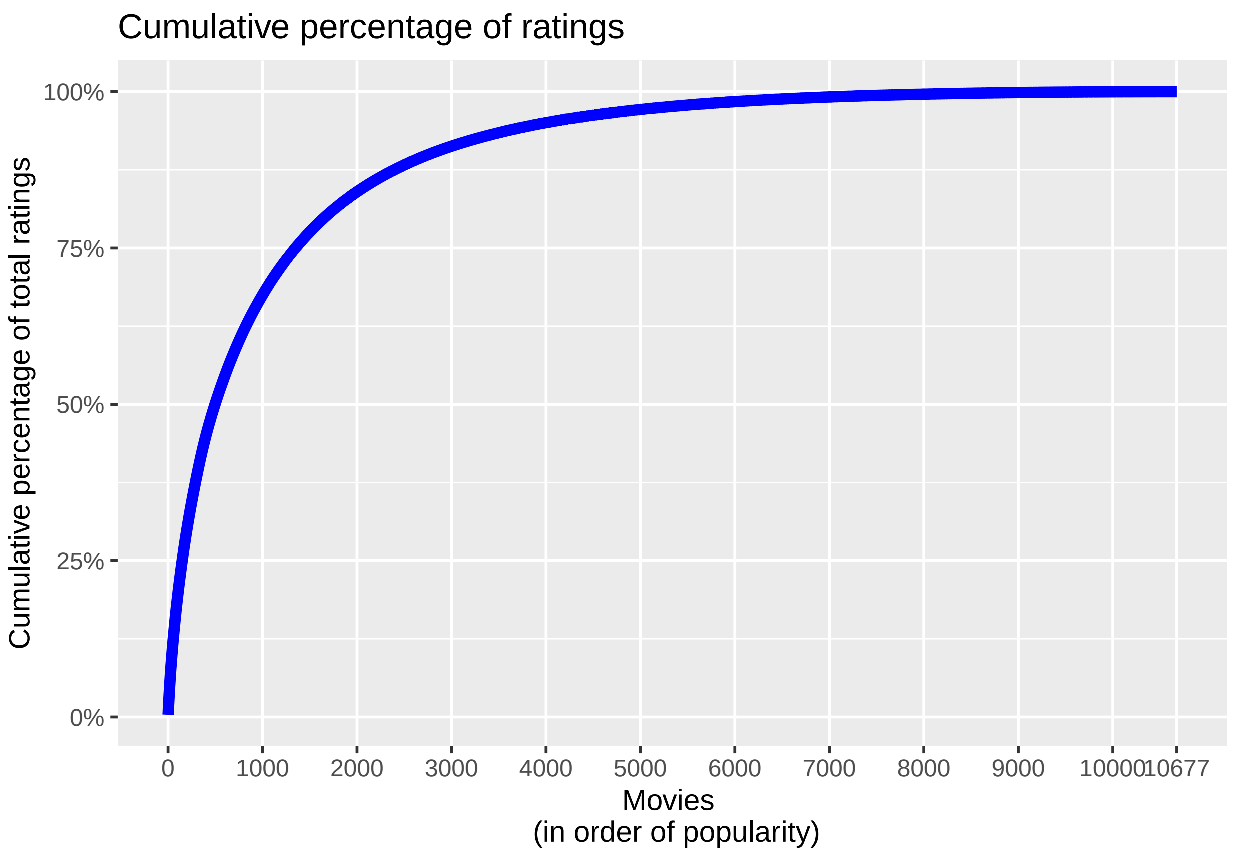
This concentration of ratings into few movies has a dramatic effect on the average. The actual median number of ratings is much lower, at 98, with the IQR of 24 to 453 reviews.



A scatter plot of the average rating per number of ratings for the entire `movies` object illustrates some of these points. We see in the figure below that the top 10 best rated movies are all concentrate into a single region and are obviously outliers. There is a clear positive correlation between movie popularity and rating, and the most popular movies are indeed well rated. However, the top 10 best rated movies with over 100 ratings are somewhat evenly distributed in the range between roughly 2 and 22 thousand ratings.



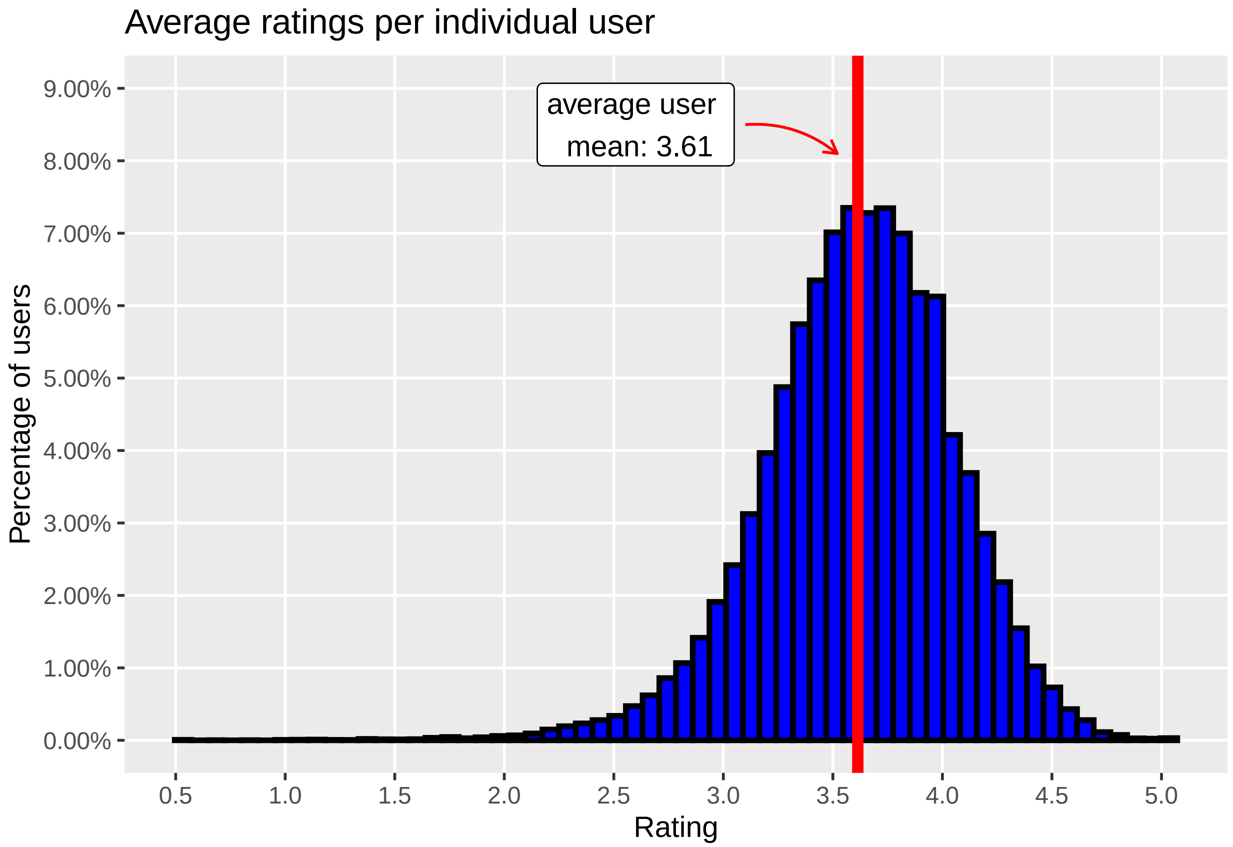
Finally, we look at the cumulative proportion of ratings, with all movies order from most to least popular. The figure below shows that, from a total of 10677 movies, only around 500 are necessary to account for half the individual ratings, while around 2800 movies account for 90% of all the ratings.



From the analysis of individual movies we see that there is a clear connection between popularity and how a movie is rated on average. We also see that there is considerable random noise in the set of movies that are more obscure, with very few ratings. Finally, we see that the ratings are very concentrated in a reduced set of very popular movies.

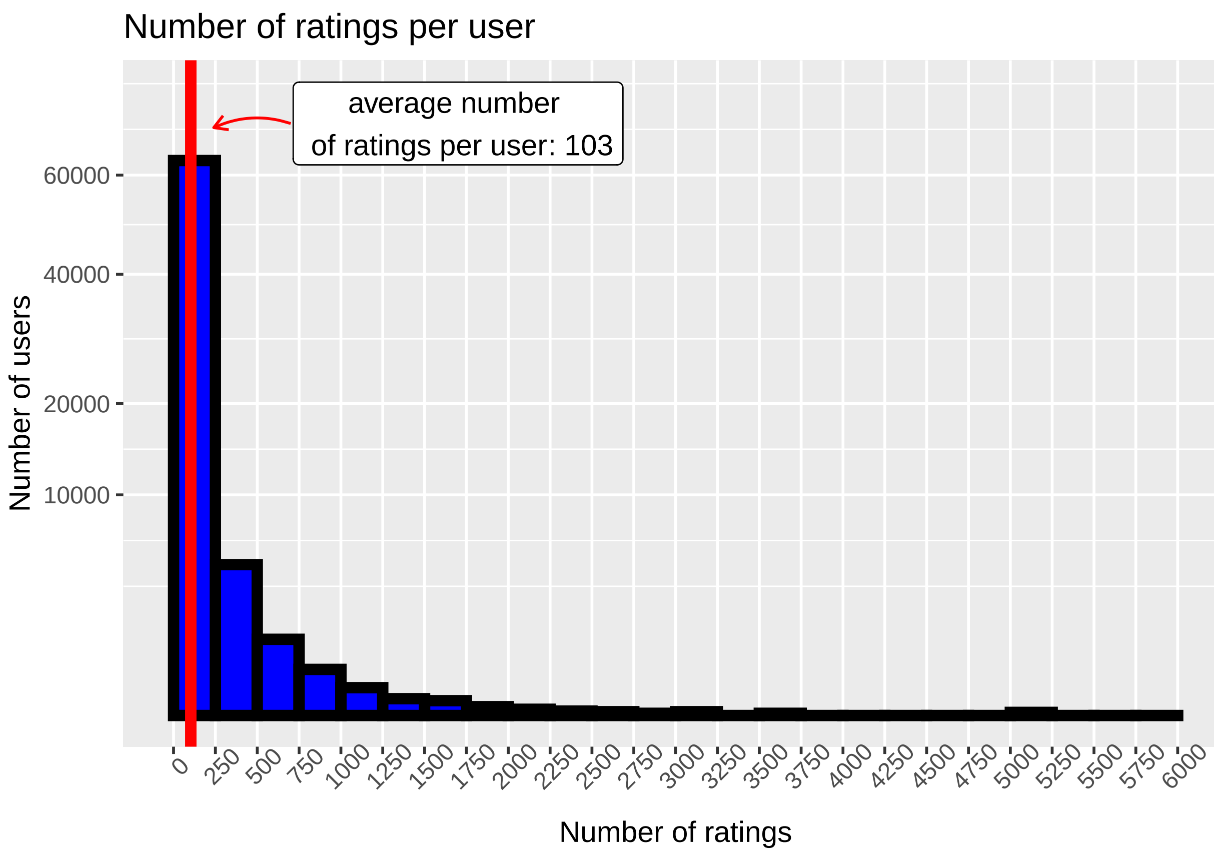
Analysis of individual users

We now turn our attention to the behavior of individual users. First, we look at the distribution of user's individual mean ratings.



The figure shows that average ratings are distributed according to a roughly bell-shaped curve, centered at 3.61. The curve is slightly skewed to the right, which agrees with the fact that the average is closer to the top limit of 5 than the bottom limit of 0.5.

The frequency at which users rate is depicted in the histogram below:



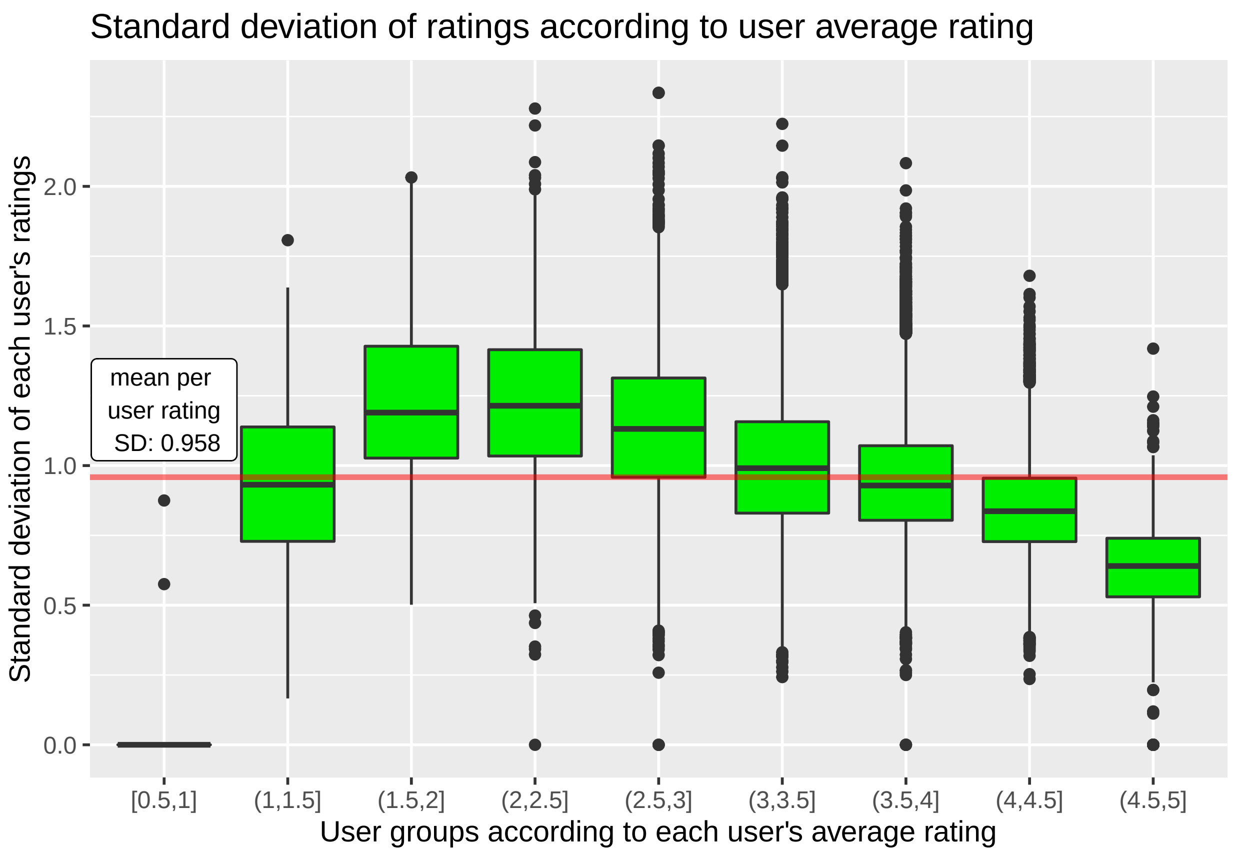
We see from the histogram that the vast majority of users have rated between 0 and 250 movies, with the average at 103. However, some users have rated considerably more often than that, with 300 users having rated more than one thousand movies, and 2 having rated over five thousand.

Some basic statistics for the number of ratings per user are shown in the table below:

Table 5: Mean and quantiles for individual user's number of ratings

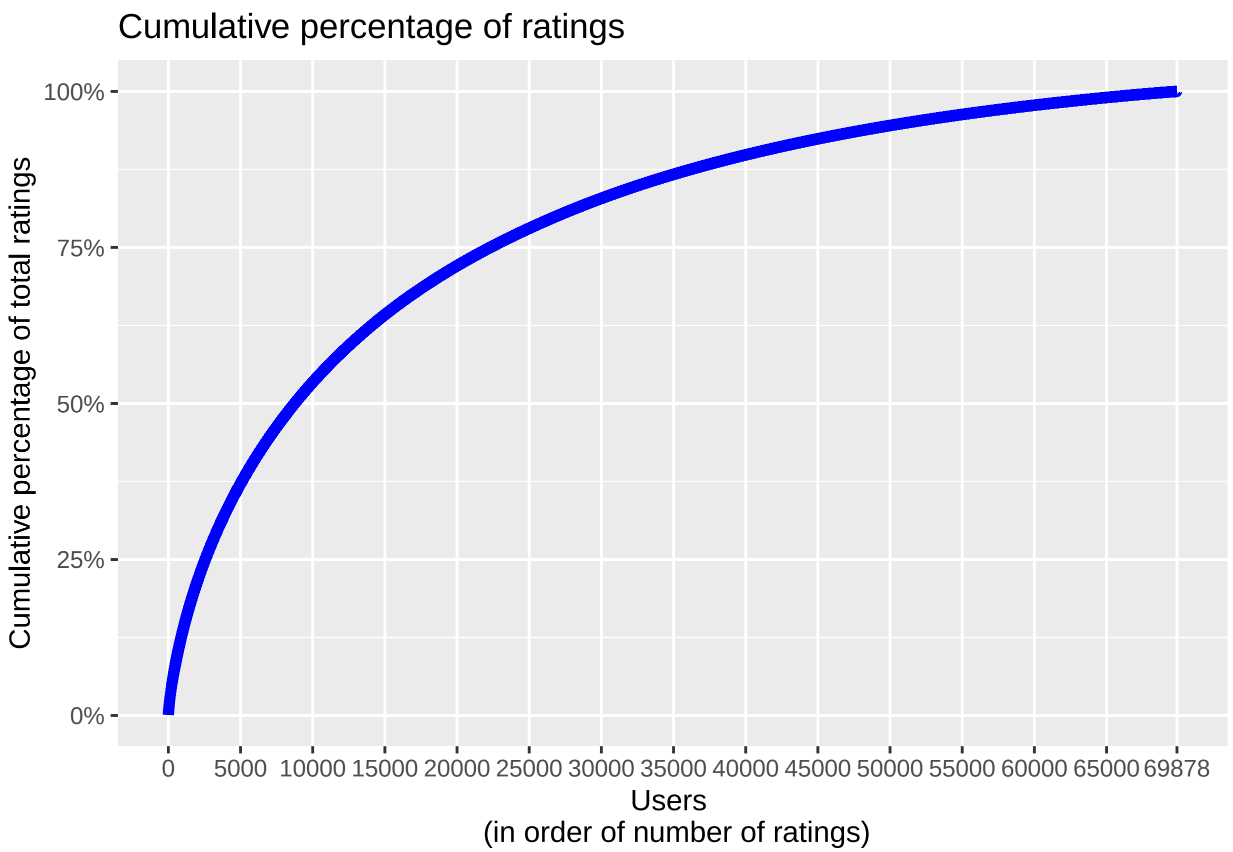
0% (min.)	25% (1st qu.)	50% (median)	Mean	75% (3rd qu.)	100% (max.)
6	26	50	103.0379	112	5222

Next, we investigate whether users rate consistently across different movies. The following plot shows the distribution of each user's rating standard deviation. Users are grouped according to their mean rating.



The figure shows that users tend to be consistent, with the mean standard deviation at 0.958. As one would expect, this standard deviation is lower for users with average ratings closer to the extremes of the scale.

Finally, we look at the cumulative percentage of ratings. The figure below shows this cumulative percentage, with users ordered according to their total number of ratings.



The curve shows that, while there is a concentration of the total number of ratings in the most active users, this concentration is not as pronounced as was observed for the most popular movies. The table below compares how much of the full sets of movies and users are required to account for 50% and 90% of the total ratings.

Table 6: Concentration of total ratings in most popular movies and most active users

Variable	Total	Percentage to account for 50% of all ratings	Percentage to account for 90% of all ratings
movies	10677	4.7%	26%
users	69878	12.5%	58%

The table confirms what was observed in the plots: there is a much higher concentration of the total ratings into the most popular movies than there is into the most active users. This indicates that statistical conclusions drawn from a reduced list of popular movies may prove to have more significance in predicting results for the full set than statistics drawn from a reduced list of most active users.

From this analysis of the individual users, we conclude that the typical user is rather consistent in giving ratings, with the average standard deviation at around 0.95 and the mean user average at 3.61. We also see that the typical user is not very active, with half of the users having less than 50 ratings in the *train* set. Finally, there is a clear concentration of the total number of ratings into the most active users, but it is not nearly as pronounced as observed across movies.

Analysis of movie genres

Upon inspection of the genres column presented in the data, we see that each entry is comprised of a string of characters with all genres in which a movie fits, separated by “|”. Below we see a sample with 12 entries with relevant columns of the movies object:

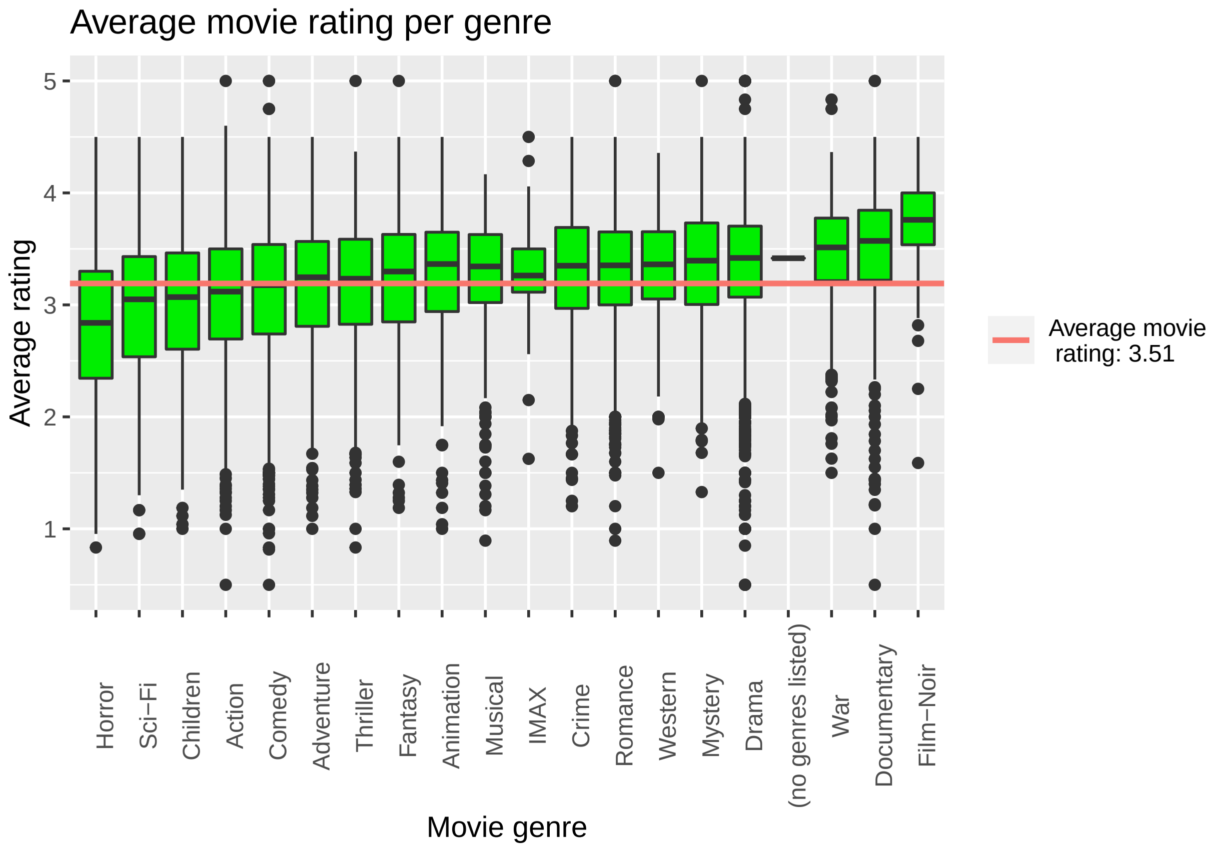
```
## # A tibble: 12 x 3
##    movieId title                 genres
##      <dbl> <chr>                 <chr>
##  1    7250 Out-of-Towners, The   Comedy
##  2    5352 Big Sleep, The        Thriller
##  3    1112 Palookaville          Action|Comedy|Drama
##  4   26544 Heaven Help Us        Comedy|Drama
##  5   59795 What Would Jesus Buy? Comedy|Documentary
##  6    6120 Q: The Winged Serpent Drama|Fantasy|Horror|Sci-Fi|Thriller
##  7    3690 Porky’s Revenge       Comedy
##  8    2629 Love Letter, The      Comedy|Romance
##  9    2112 Grand Canyon          Crime|Drama
## 10    2442 Hilary and Jackie     Drama
## 11    6093 Last Unicorn, The     Animation|Children|Fantasy
## 12    5471 Perfect               Drama|Romance
```

After some data manipulation, we see that the genres variable contains 19 different genres, which are counted and arranged by number of appearances in the table below:

Table 7: Movie genres arranged by number of movies

#	Genre	Number of Movies	Average Rating
1	Drama	5336	3.35
2	Comedy	3703	3.11
3	Thriller	1705	3.18
4	Romance	1685	3.30
5	Action	1473	3.06
6	Crime	1117	3.29
7	Adventure	1025	3.16
8	Horror	1013	2.80
9	Sci-Fi	754	2.97
10	Fantasy	543	3.20
11	Children	528	3.00
12	War	510	3.46
13	Mystery	509	3.34
14	Documentary	481	3.47
15	Musical	436	3.26
16	Animation	286	3.25
17	Western	275	3.32
18	Film-Noir	148	3.72
19	IMAX	29	3.27
20	(no genres listed)	1	3.42

We see that “Drama” is the most common movie genre. The table also shows that there is a relevant difference between average movie ratings across different genres. We can see the distribution of these average ratings in the plot below:



As indicated in the previous table, all genres other than "IMAX" have over 100 individual movies, which suggests that the relation between movie genres and average movie ratings is an actual feature of the data and not due to random variability.

We also note that each movie is not associated with a single genre, but rather a collection of genres. In fact, 4 movies classified with 7 or 8 different genres:

Table 8: Movies with 6+ different genres

Title	# Genres	Genres
Host, The (Gwoemul)	8	Action|Adventure|Comedy|Drama|Fantasy|Horror|Sci-Fi|Thriller
Who Framed Roger Rabbit?	7	Adventure|Animation|Children|Comedy|Crime|Fantasy|Mystery
Monster House	7	Adventure|Animation|Children|Comedy|Drama|Fantasy|Mystery
Enchanted	7	Adventure|Animation|Children|Comedy|Fantasy|Musical|Romance

Some genres are more frequently found together. We calculate the correlations between the columns in the `genres$wide` data frame and show the highest and lowest pairs to demonstrate:

Table 9: Movie genre pairs with highest correlation

Pair	Correlation
Animation | Children	0.3566642
Children | Musical	0.1819614
Crime | Thriller	0.1758849
Horror | Thriller	0.1756856
Mystery | Thriller	0.1750220
Children | Fantasy	0.1668665
Crime | Film-Noir	0.1557878
Adventure | Children	0.1495040
Documentary | IMAX	0.1442926
Action | Adventure	0.1412978

Not surprisingly, there is positive correlation between genres like “animation” and “children”, as well as “mystery” and “thriller” or “action” and “adventure”. Similar results appear when we look at the bottom of the table, with the least correlated genres:

Table 10: Movie genre pairs with lowest correlation

Pair	Correlation
Children | Thriller	-0.2572242
Crime | Fantasy	-0.2019309
Animation | Thriller	-0.1972529
Musical | Thriller	-0.1917192
Action | Musical	-0.1883138
Comedy | Thriller	-0.1803907
Animation | Crime	-0.1641847
Children | Horror	-0.1631741
Romance | Sci-Fi	-0.1542794
Adventure | Horror	-0.1510313

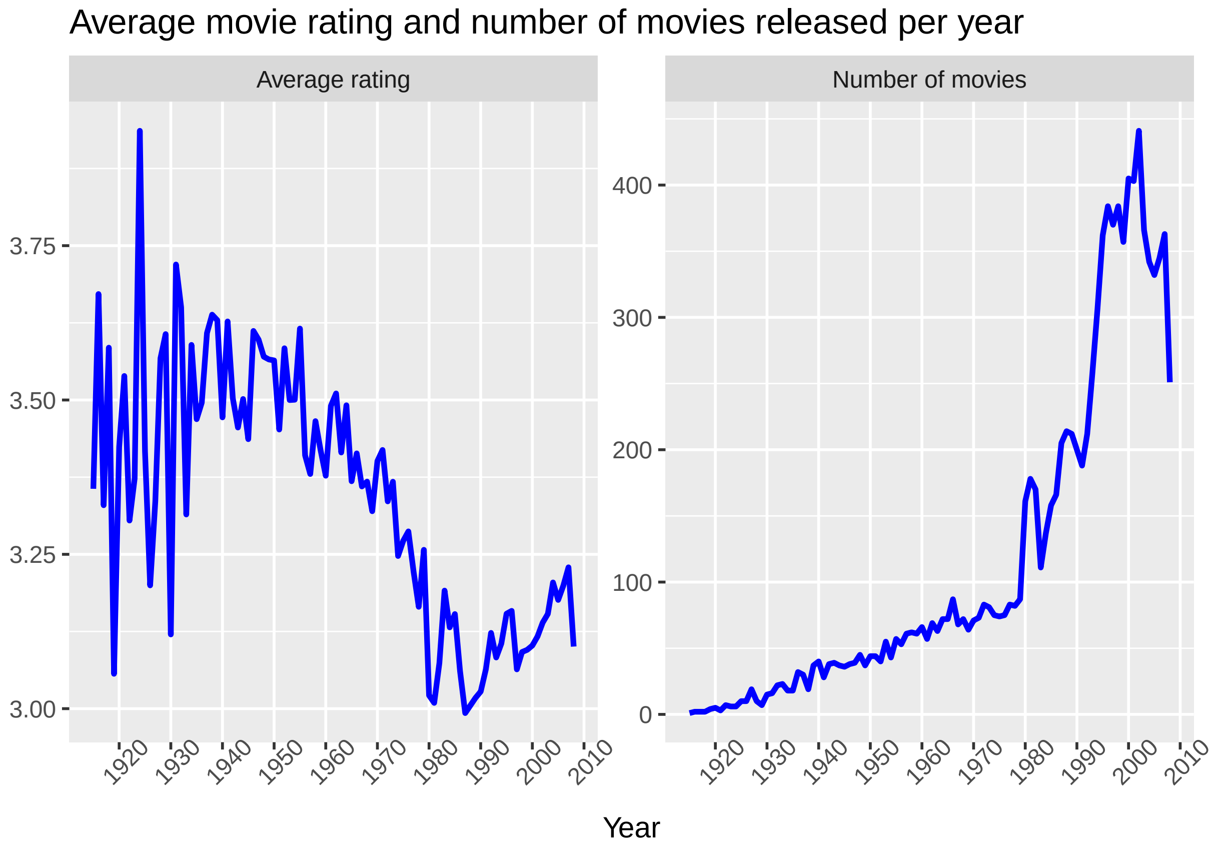
As you would imagine, certain pairs are not found often (though I suspect writing a children’s thriller or an action musical would be an interesting challenge for a movie writer).

From this analysis of the movies genres, we conclude that there is a large difference in the frequency with which individual genres are used to describe movies, with “Drama” and “Comedy” being the most frequent. There is also a connection between the movie genres and the average rating, with movies tagged with “Horror” averaging 2.80, while ones tagged with “Film-Noir” average 3.72. We also note a significant correlation between certain pairs of genres that are often found together.

Analysis of time

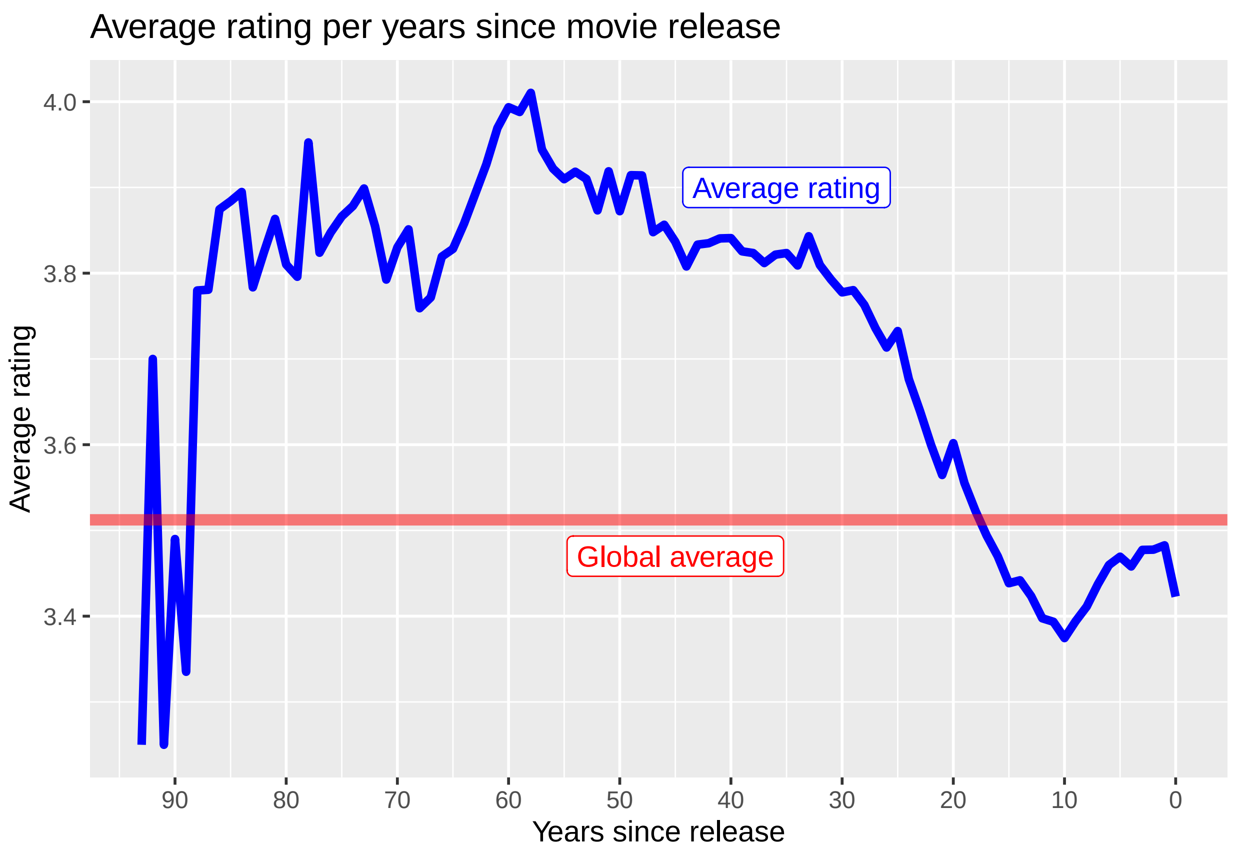
As a final step in the exploratory data analysis, we look at the distribution of the year of release of each movie and how the average ratings are distributed according to the year of release. We also investigate how the number of years since the release of each movie at the time of rating influences the rating.

The figure below contains a depiction of the number of movies released at each year and the average rating for movies released at each year:



We see that the number of movies in the dataset increases dramatically in the late 70s, as well as in the early 90s. We also see that the average movie rating has a strong negative correlation with the year of release.

It is also interesting to look at the average ratings as a function of the number of years passed since the release of the movie at the time of each rating. The plot below shows this information:



Again, we see that there is a strong connection between the movie age at the time of rating and the value of the rating, demonstrating a tendency of good movies to survive the test of time.

From this analysis, we conclude that there is a prevalence of newer movies in the dataset, but that older movies (be it according to year of release of age at the time of rating) tend to be better rated.

Modeling

We now proceed to use the hints provided by the exploratory data analysis to construct models that predict movie ratings given by users. We begin by constructing a reference model in which all predictions are the global average rating calculated on the *train* set. Then, we create increasingly complete models by applying the general approach described below:

1. Choose a previous model as reference
2. Calculate residuals from the previous model applied to the *train* set
3. Create a new model with an added term that explains the residuals
4. Train parameters on the *test* set (if applicable)
5. Evaluate performance on the *test* set

Finally, after development of a complete model resulting from the combination of all the previous steps, we evaluate model performance on the *validation* set in the Results section of the report.

Naive model

We begin by constructing a simple model in which the global average rating calculated in the *train* set (3.5123554) is given uniformly as prediction to all ratings.

The model is depicted in mathematical notation below:

$$Y_{u,m} = \mu + \epsilon$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating
ϵ = random residual

The RMSE result obtained after application of this model to the *test* set is below:

Table 11: Results from application of Naive model (modelId = 1) to test set

modelId	RMSE	description
1	1.060301	Global average (benchmark)

Simple movie effect

We proceed to include the movie effect in the model, representing the fact that each movie has an inherent quality that makes it be qualified, on average, above or below the global mean. The movie effect is calculated for each movie as the mean residual obtained after application of the naive model on the *train* set. The model with this new parameter is this:

$$Y_{u,m} = \mu + e_{movie} + \epsilon$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating
e_{movie} = movie effect for movie m
ϵ = random residue

As e_m is modeled as the average points each movie receives above the *train* set average, $\mu + e_m$ is equal to each individual movie average. This version of the model corresponds to predicting that each movie is rated as the average of previous ratings plus a random term.

Results on the *test* set are displayed below:

Table 12: Results from application of Simple Movie Effect model (modelId = 2) to test set

modelId	RMSE	description
2	0.9440701	model 1 + simple movie effect

Regularized movie effect

Although the results did improve, the fact that some movies are rated very few times (as observed in the Exploratory Data Analysis section) indicates that a prediction based solely on the movie average is not adequate. In these situations, it is normally beneficial to use a prediction that is closer to the global average for movies with few ratings, and converges the the movie average as they receive a more substantial number of ratings. A regularization of each movie's mean rating is appropriate to achieve this correction.

The model with the movie effect regularized according to the number of ratings is represented via the equation below:

$$Y_{u,m} = \mu + e^r_{movie} + \epsilon$$

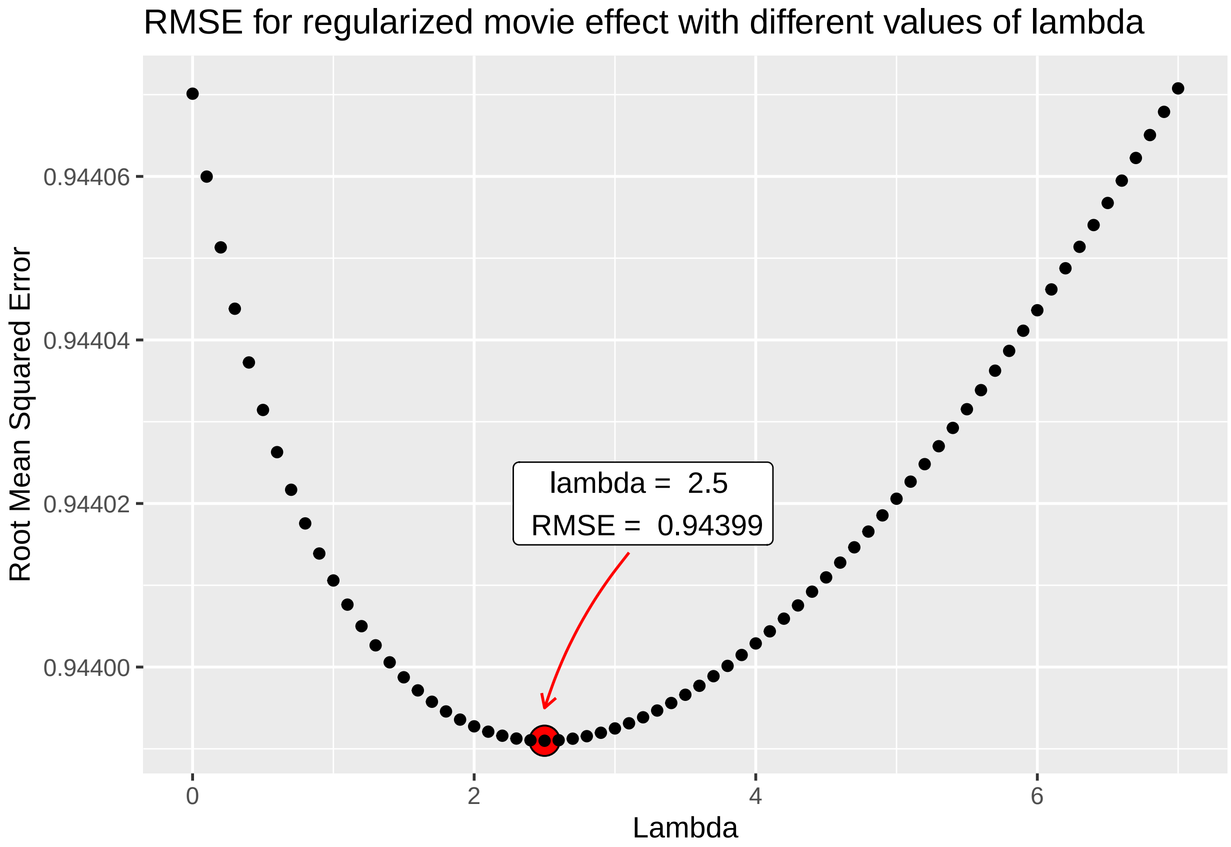
where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e^r_{movie} = regularized movie effect ϵ = random residue

The regularized movie effect corresponding to each individual movie is obtained via a modified version of the average deviation from the global average, according to the equation below:

$$e^r_m(\lambda) = \frac{1}{\lambda + n_i} * \sum_{m=1}^{n_i} (Y_{m,i} - \mu)$$

This indicates that the movie effect e^r_m depends on an arbitrary parameter λ, which can be chosen to minimize the residue.

The figure below depicts the results obtained on the *test* set after calculating the regularized movie effect on the *train* set with different values of lambda, from which we choose the optimal.



The optimal regularization effect parameter λ was found to be 2.5.

From he RMSE results in the table above, it is evident that regularization has not contributed with a substantial improvement. This may be due to the fact that, as the dataset gets larger, instances of movies with very few ratings become rarer and have a lesser effect on the overall results.

The RMSE after application of this model on the *test* set is given below:

Table 13: Results from application of Regularized Movie Effect model (modelId = 3) to test set

modelId	RMSE	description
3	0.943991	model 1 + regularized movie effect

Simple user effect

Similarly to what was done for the movie averages, we proceed to modify our model to account for the fact that different users tend to rate differently. In order to estimate this effect, we calculate the residuals after application of the previous model on the *train* set and calculate the average residual for each user. We then add this term for each user to get a new prediction.

As we include additional terms in the model, it is important to make sure that predictions fall between 0.5 and 5, the minimum and maximum possible ratings. To do so, we introduce the function `limitRange`, defined as below:

$$limitRange_{Y_{min}}^{Y_{max}}(Y) = \begin{cases} Y_{min} & \text{, if } Y < Y_{min} \\ Y_{max} & \text{, if } Y > Y_{max} \\ Y & \text{otherwise} \end{cases}$$

The modified model is depicted below:

$$Y_{u,m} = limitRange_{0.5}^{5}(\mu + e_{movie}^{r} + e_{user} + \epsilon)$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e_{movie}^{r} = regularized movie effect
e_{user} = simple user effect ϵ = random residue

After application of this model, we obtain the results below:

Table 14: Results from application of User Effect model (modelId = 4) to test set

modelId	RMSE	description
4	0.8659097	model 3 + simple user effect

Regularized user effect

As we did for the movie effect, we proceed to modify our model to account for the fact that certain users have a low number of ratings and that their average ratings may be heavily influenced by random variability.

We evaluate several values of λ in the *test* set and select the one that gives the lower value of the RMSE. In this case, the best λ is 4.5.

The modified model with the regularized user effect is depicted below:

$$Y_{u,m} = limitRange_{0.5}^{5}(\mu + e_{movie}^{r} + e_{user}^{r} + \epsilon)$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e_{movie}^{r} = regularized movie effect
e_{user}^{r} = regularized user effect ϵ = random residue

The application of this model in the *test* set gives the following results:

Table 15: Results from application of Regularized User Effect model (modelId = 5) to test set

modelId	RMSE	description
5	0.8656025	model 3 + regularized user effect

Again, we see that regularization does not introduce a great gain in performance, which is explained by the fact the fact that we are using a massive dataset and that ratings from users that are not very active do not account for a significant portion of the total ratings.

Effect of movie genres (Drama movies)

As observed during the Exploratory Data Analysis section, there is a connection between the movie ratings and the movie genres with which each movie is tagged. However, this connection is already accounted for in the previous model: since there is a one-to-one relation between a movie and it's tags (meaning that the tags are constant and do not change across different ratings), this effect is already represented in the movie average.

Regardless of that, we can investigate the residuals from the previous model to determine whether each individual user is partial do a particular genre. Users with a preference for a particular genre will have higher residuals for movies tagged with that genre.

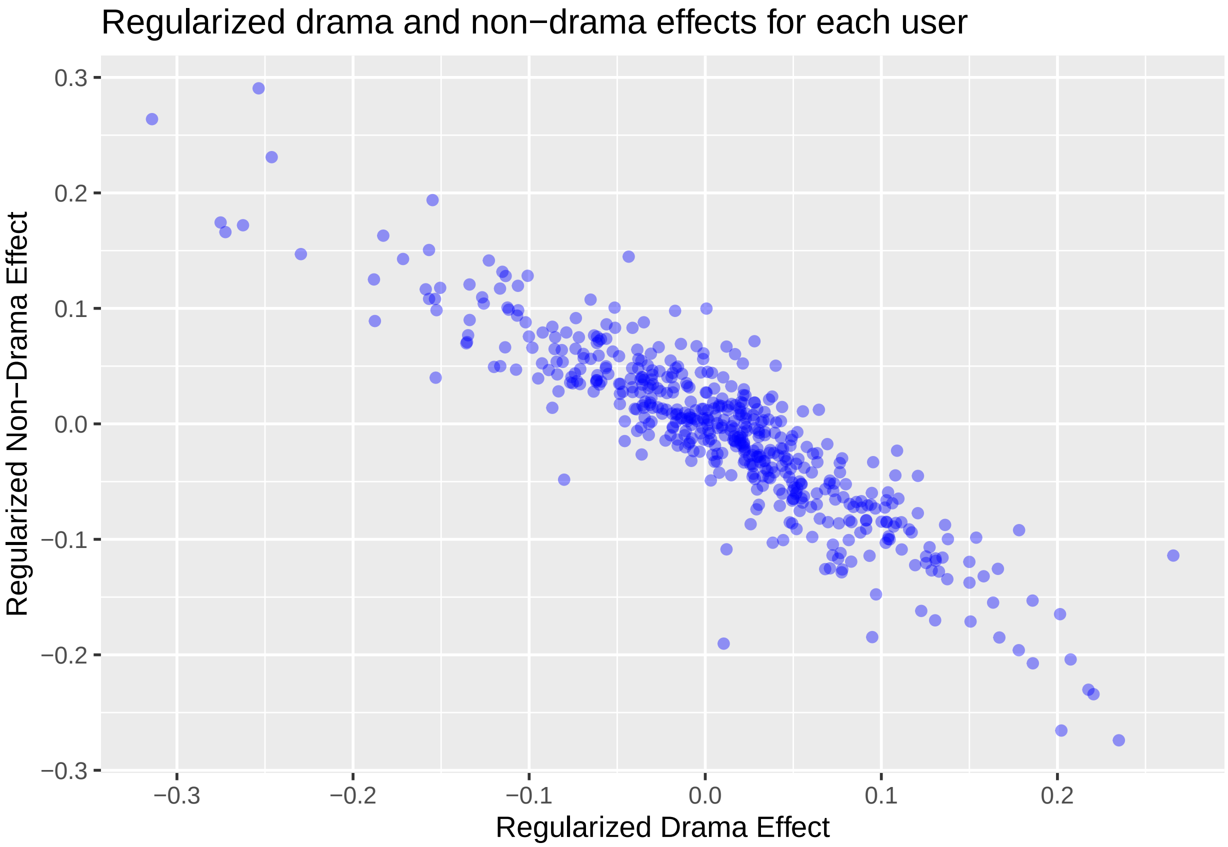
As a proof of concept, we now create a model with an added term corresponding to the mean residual observed for each user for movies either tagged or not tagged with a particular genre. For this, we choose the Drama genre, which was observed to be the most frequent. We will evaluate the residuals in the *train* set, calculate their average for each user by grouping movies that are tagged with Drama and movies that are not, and apply regularization by selecting the value of λ that minimizes RMSE observed in the *test* set.

The corresponding model is represented as:

$$Y_{u,m} = limitRange_{0.5}^{5}(\mu + e_{movie}^{r} + e_{user}^{r} + e_{drama/non-drama}^{r} + \epsilon)$$

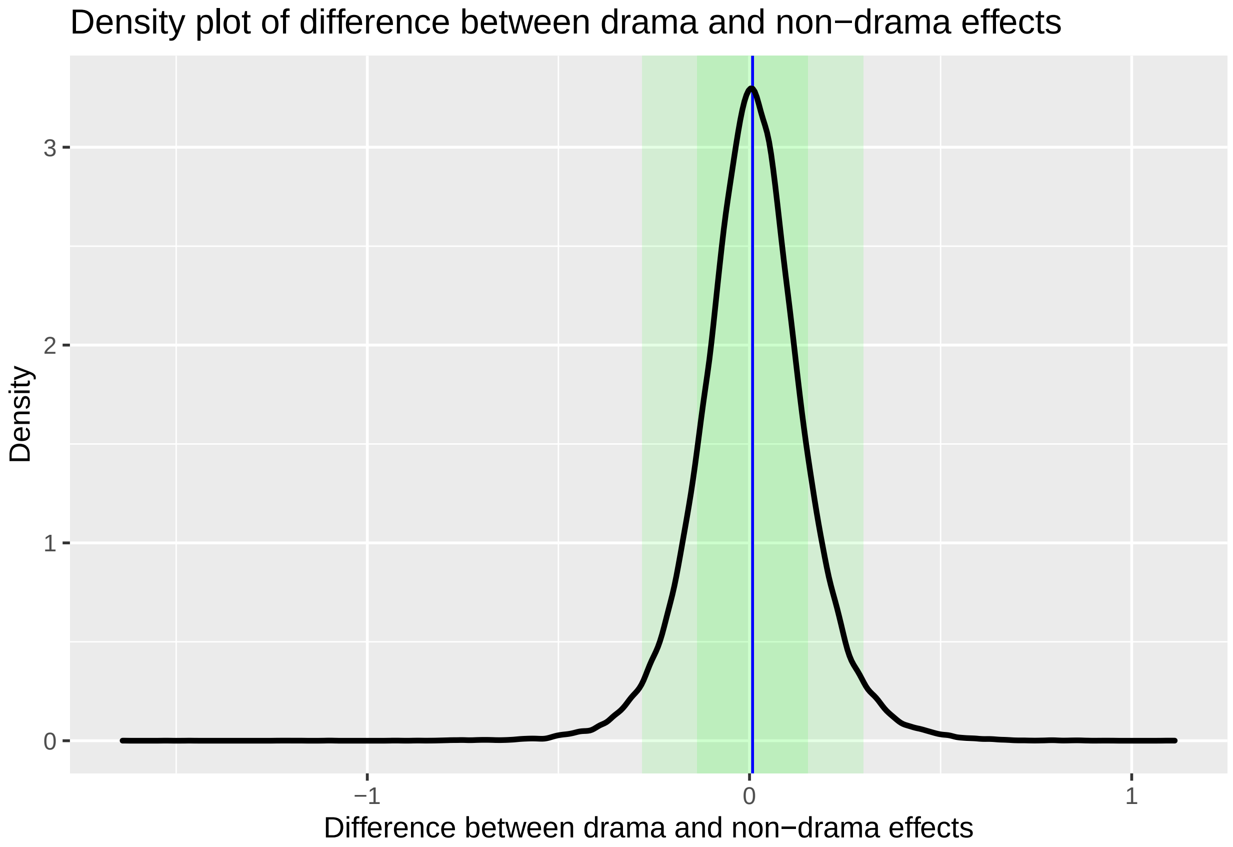
where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e_{movie}^{r} = regularized movie effect
e_{user}^{r} = regularized user effect $e_{drama/non-drama}^{r}$ = regularized drama/non-drama effect
ϵ = random residue

To illustrate how this term is distributed across users, we look at a scatter plot of the drama and non-drama effects for a sample of 500 random users:



The plot shows a clear negative correlation between the drama and non-drama effects for each user. This is to be expected, since by definition their average (weighted by the number of drama and non-drama movies) is equal to each user's mean residual.

In order to get a sense of how pronounced the preference is, look at a plot of the spread between the drama and non-drama effects.



The figure above contains the density plot for the difference between the drama and non-drama effects for each user, with a blue line depicting the average and the different shades of green background depicting the intervals of one and two standard deviations from the mean. The distribution is bell-shaped, centered at 0.008 and with a standard deviation of 0.145. This indicates that, for 95% of users, the difference between the average rating between movies with and without the 'Drama' tag is below 0.29 points.

When we account for this additional piece in the model, we get the following results:

Table 16: Results from application of Drama/Non-Drama Effect model (modelId = 6) to test set

modelId	RMSE	description
6	0.8614645	model 5 + regularized drama/non-drama effects

Genre effect through Principal Component Analysis (PCA)

Even though the results obtained with the previous model were an improvement, we would like to extend to analysis for all of the available movie genres. In theory, we could repeat the process of calculating the residuals from one model and adding an effect for each of the movie genres, but that would be repetitive and cumbersome. The most appropriate way to tackle the effect of the movie genres on the ratings is by calculating effects for all of them at the same time.

When addressing the effect of multiple variables at once, we must consider the fact that there is a correlation between the predictors. As observed during the exploratory data analysis, there is a correlation between the tags associated with each of the genres. Consequently, if we choose to calculate effects for each of the genres separately and add them all together according to each movie's genres, we risk "adding the same thing twice".

To account for this correlation, we take the matrix saved in the `genres$wide` object, which contains columns indicating whether each genre tag is part of each of the different genre combinations in the dataset - and perform a Principal Component Analysis (PCA). The resulting object is saved to the `genres$principal.components` object and contains the principal components for the matrix, representing a rotation to a set of coordinates that makes the columns uncorrelated. We can think of principal components as "equivalent genres", which are independent from one another.

The object resulting from the PCA also contains, among other pieces of information, the rotation matrix used to convert from the original logical values indicating whether a movie is tagged with each of the movie genres to the continuous values indicating how much the movie is associated with each of the principal components.

An excerpt from the original data in the `genres$wide` object is shown below:

```
##                          genres Comedy Romance Action Crime Thriller Drama Sci-Fi
## 1               Comedy|Romance   TRUE    TRUE  FALSE FALSE    FALSE FALSE  FALSE
## 2        Action|Crime|Thriller  FALSE   FALSE   TRUE  TRUE     TRUE FALSE  FALSE
## 3 Action|Drama|Sci-Fi|Thriller  FALSE   FALSE   TRUE FALSE     TRUE  TRUE   TRUE
## 4      Action|Adventure|Sci-Fi  FALSE   FALSE   TRUE FALSE    FALSE FALSE   TRUE
## 5      Children|Comedy|Fantasy   TRUE   FALSE  FALSE FALSE    FALSE FALSE  FALSE
## 6     Comedy|Drama|Romance|War   TRUE    TRUE  FALSE FALSE    FALSE  TRUE  FALSE
##   Adventure Children Fantasy   War Animation Musical Western Mystery Horror Film-Noir
## 1     FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 2     FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 3     FALSE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 4      TRUE    FALSE   FALSE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 5     FALSE     TRUE    TRUE FALSE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
## 6     FALSE    FALSE   FALSE  TRUE     FALSE   FALSE   FALSE   FALSE  FALSE     FALSE
##   Documentary  IMAX (no genres listed)
## 1       FALSE FALSE              FALSE
## 2       FALSE FALSE              FALSE
## 3       FALSE FALSE              FALSE
## 4       FALSE FALSE              FALSE
## 5       FALSE FALSE              FALSE
## 6       FALSE FALSE              FALSE
```

We can also see an excerpt from the `genres$principal.components$x` object, which contains the values of each of the principal components for each movie in the dataset. The values of the components are continuous, as opposed to the original values which where logical.

```
##           PC1          PC2         PC3         PC4         PC5         PC6         PC7
## 1 -1.00349551 -0.003621436  0.49748730  0.61938280 -0.26930415 -0.37886205  1.34364741
## 2 -0.33032637 -0.471594251  0.80738547  0.47828204  0.04356930 -0.09539502  1.15087991
## 3 -0.81422863  0.551347816 -0.24074062  0.06118877  0.64660798  0.09743760 -0.06494836
## 4 -0.06850071  1.012082459 -0.38194737  0.20887547  0.43083447 -0.19287805 -0.06549951
## 5 -0.78925081  0.222587165 -0.03439398 -0.15357760 -0.23793438  0.02523626 -0.07382492
## 6  0.02024799 -1.169497917 -0.88224028  0.35800661 -0.07928572  0.55251752 -0.12176358
##            PC8          PC9        PC10          PC11          PC12        PC13
## 1 -0.192368444  0.277132441 -0.42534751  1.445321e-01 -0.0106085226 0.340021354
## 2 -0.159675452  0.263385855 -0.19135462  6.558184e-02 -0.1541750581 0.382492360
## 3  0.032318203 -0.021048503  0.05922673  5.188830e-04 -0.0009563814 0.017746002
## 4 -0.046212295 -0.004455704  0.03287435 -5.579304e-02 -0.0151889011 0.035608842
## 5  0.108110459 -0.022334703  0.05536344 -9.477939e-05 -0.0240778548 0.001558162
## 6  0.005675863  0.070142334 -0.17523281 -7.843739e-02  0.1387569617 0.099984324
##          PC14        PC15        PC16         PC17          PC18          PC19
## 1 0.18917291 -0.17754277  0.11425865  0.420717408 -8.216308e-05  0.0190771562
## 2 0.09843175  0.12563786 -0.08213033 -0.458370669  5.880179e-03  0.0022170929
## 3 0.09800892  0.02594786 -0.05584165  0.006248834  3.974770e-03 -0.0001822967
## 4 0.03684341 -0.13628102  0.07711716  0.006848232 -1.728377e-02 -0.0014354283
## 5 0.04636914  0.09177617 -0.09060276  0.012174524  1.294167e-02 -0.0003711257
## 6 0.03688075  0.05730162 -0.05143720 -0.009178221  6.703405e-02  0.0009304504
##             PC20
## 1 -0.0002708846
## 2  0.0002805121
## 3  0.0001723632
## 4 -0.0003263937
## 5  0.0002995218
## 6  0.0002151488
```

Below we see the output of the `summary()` function applied to the `genres$principal.componenents` object, which has been created using the R function `prcomp()`. The summary depicts how much of the total variance is explained by each of the principal components.

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8
## Standard deviation     0.5657 0.4936 0.37274 0.36291 0.34528 0.29156 0.28098 0.25651
## Proportion of Variance 0.2091 0.1592 0.09077 0.08604 0.07789 0.05554 0.05158 0.04299
## Cumulative Proportion  0.2091 0.3682 0.45900 0.54505 0.62293 0.67847 0.73005 0.77304
##                            PC9    PC10    PC11   PC12    PC13    PC14    PC15    PC16
## Standard deviation     0.23430 0.21974 0.20632 0.2033 0.19506 0.18766 0.17921 0.14922
## Proportion of Variance 0.03586 0.03155 0.02781 0.0270 0.02486 0.02301 0.02098 0.01455
## Cumulative Proportion  0.80890 0.84045 0.86826 0.8952 0.92011 0.94312 0.96410 0.97865
##                           PC17    PC18    PC19     PC20
## Standard deviation     0.13030 0.11397 0.05120 0.009675
## Proportion of Variance 0.01109 0.00849 0.00171 0.000060
## Cumulative Proportion  0.98974 0.99823 0.99994 1.000000
```

In order to include this information in the model, we begin by grouping all the ratings in the *train* set by users and performing weighted averages of the residuals of every rating given by each user. These averages are weighted by the values of each of the principal components associated with each movie, so that every user ends up with 20 scores indicating how partial they are to each of the principal components. These values are then regularized, so that results for users with a lower number of ratings have reduced importance. The resulting model follows the equation below:

$$Y_{u,m} = limitRange_{0.5}^{5}(\mu + e^{r}_{movie} + e^{r}_{user} + \sum_{n=1}^{N_{PC}} e^{r}_{genre(PC)} + \epsilon)$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e^{r}_{movie} = regularized movie effect
e^{r}_{user} = regularized user effect N_{PC} = number of principal components $e^{r}_{genre(PC)}$ = regularized genre effect
ϵ = random residue

Here, we use all of the 20 principal components of the movie genres tagging system, so $N_{PC} = 20$. The value of λ that minimizes the residuals after application of this model in the *test* set is 35, and the RMSE in the *test* set is displayed below:

Table 17: Results from application of Genre Effect model (modelId = 7) to test set

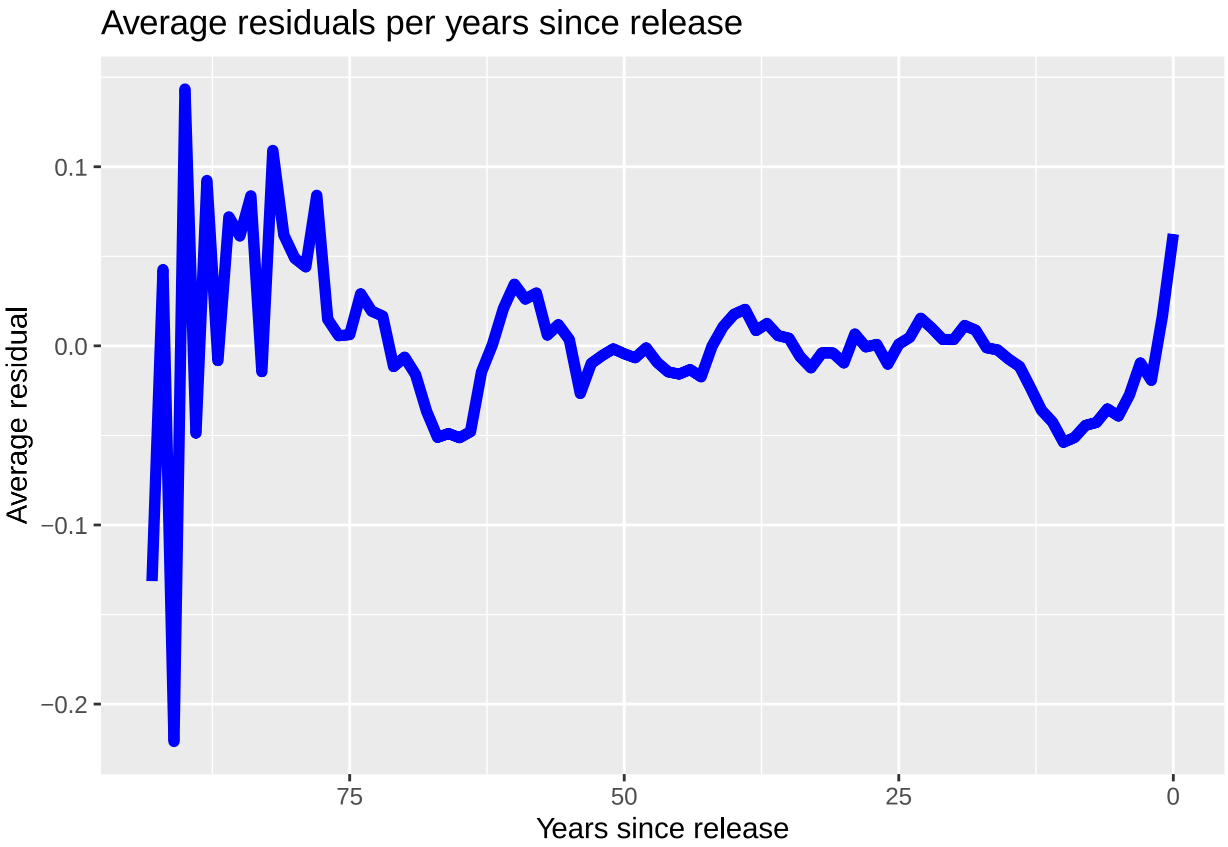
modelId	RMSE	description
7	0.8496471	model 5 + regularized genre effect through PCA

Movie age effect

We now turn our attention to the effect that the year of release (extracted from the movie title in the original dataset) and the date of the rating have on the ratings. As was the case with the movie genres, looking at the year of release in an isolated manner brings no new information, since there is a one-to-one connection between the movie and its age of release. Any effect strictly from the year of release would have already been captured in the movie average rating, already accounted for in previous models.

However, there is one aspect that can still be explored, which is the age of the movie at the time of each rating. We get that by subtracting each movie's year of release from the year each rating was given.

A plot of the average residuals as a function of the movie age at time of rating shows interesting features:



We can see that the average residual is positive for the first two years after a movie has been release, which seems to indicate some enthusiasm from avid users who like to watch the movies as soon as possible. There is then a period of negative residuals up to around 15 years after release. Finally, the average residual increases until it starts to oscillate around zero, when it appears there is a renewed interest for them (or at least they are considered good enough to be worth watching and rating). For longer periods, the curve gets very noisy, since there are considerably less movies and ratings for these periods.

Because there is a long interval of movie age for which there are few data points, this effect benefits a lot from regularization. The value of λ that minimizes the residuals in the *test* set is 400, considerably larger than for the other phenomenons.

The model that incorporates this feature of the data is represented below, followed by the results obtained after application on the *test* set.

$$Y_{u,m} = limitRange_{0.5}^{5}(\mu + e_{movie}^{r} + e_{user}^{r} + \sum_{n=1}^{N_{PC}} e_{genre(PC)}^{r} + e_{age}^{r} + \epsilon)$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e_{movie}^{r} = regularized movie effect
e_{user}^{r} = regularized user effect N_{PC} = number of principal components $e_{genre(PC)}^{r}$ = regularized genre effect
e_{age}^{r} = regularized movie age effect ϵ = random residue

Table 18: Results from application of Movie Age Effect model (modelId = 8) to test set

modelId	RMSE	description
8	0.8491858	model 7 + regularized movie age effect

Collaborative filtering

As a final step in the modeling phase, we implement a collaborative filtering technique to improve the predictions.

In previous sections, we investigated how each of the predictors related directly with the ratings. Collaborative filtering, on the other hand, looks at how predictors are indirectly related to the ratings. An User-Based Collaborative Filter (UBCF) looks at how similar users have rated a particular movie to come to a prediction. An Item-Based Collaborative Filter (IBCF) looks at how the user has rated similar movies to come to a prediction.

Both approaches required identifying similarities between either different users or different items (movies, in our case).

Here, we prefer to use a IBCF due to the fact that there is a higher concentration of ratings in popular movies than there is in active users, as revealed during the exploratory data analysis phase. This permits some simplifications - instead of calculating similarities between every pair of movies in the dataset, we only calculate similarities between each movie and a selection of the 1000 most popular movies. This approach drastically reduces the dimensions of the matrices containing these similarities.

As a measure of similarity between movies, we choose to calculate the correlation between the residuals obtained after application of the previous model. We calculate the correlation based solely on the users that have rated both movies.

The corrections according to the user's rating to a particular movie is given by the equation below:

$$e_{IBCF}^{u,m} = \frac{M_{u,m_R} * C_{m_R,m}}{\sum_{m_R} C_{m_R,m}}$$

where: $e_{IBCF}^{u,m}$ = adjustment to prediction of rating of user u to movie m according to ratings given by user u to similar movies
M_{u,m_R} = ratings given by user u to set of most popular movies
$C_{m_R,m}$ = correlation matrix between each movie in the complete dataset and each of the most popular

movies, calculated from the set of common users between each pair of movies
$\sum_{m_R} C_{m_R,m}$ = row sums of $C_{m_R,m}$

Because the value of $e^{u,m}_{IBCF}$ is unique for each user/movie combination, it is impracticable to represent the full set of values in usual matrix form. The package `Matrix` is used to represent these values in sparse matrix form.

In order to further reduce the need for computational resources, the sets of users and movies are partitioned and the values of $e^{u,m}_{IBCF}$ are calculated sequentially for each combination of this partitions. After some experimentation, we choose to create 7 partitions of users and 10 partitions of movies, so that the full corrections matrix is completed in 70 steps.

With the inclusion of this features, we reach our complete model for predicting movie ratings. It is expressed mathematically as:

$$Y_{u,m} = limitRange^{5}_{0.5}(\mu + e^r_{movie} + e^r_{user} + \sum_{n=1}^{N_{PC}} e^r_{genre(PC)} + e^r_{age} + e^{u,m}_{IBCF} + \epsilon)$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e^r_{movie} = regularized movie effect
e^r_{user} = regularized user effect N_{PC} = number of principal components $e^r_{genre(PC)}$ = regularized genre effect
e^r_{age} = regularized movie age effect $e^{u,m}_{IBCF}$ = adjustment to prediction of rating of user u to movie m according to IBCF ϵ = random residue

The results after application of this model on the *test* set are depicted below:

Table 19: Results from application of Item-Based Collaborative Filter model (modelId = 9) to test set

modelId	RMSE	description
9	0.8432272	model 8 + similar movies effect

Results

The final model has been constructed by considering the rating averages for each particular movie and user. It also considers each user's individual preference for the different movie genres and the effect that the number of years since movie release has on the ratings. Finally, ratings are adjusted according to how each user has rated similar movies (from a list of the 1000 most popular movies). The mathematical representation of the full model is repeated here for convenience:

$$Y_{u,m} = limitRange^{5}_{0.5}(\mu + e^r_{movie} + e^r_{user} + \sum_{n=1}^{N_{PC}} e^r_{genre(PC)} + e^r_{age} + \epsilon)$$

where:
$Y_{u,m}$ = rating given by user u to movie m
μ = mean global rating e^r_{movie} = regularized movie effect
e^r_{user} = regularized user effect N_{PC} = number of principal components $e^r_{genre(PC)}$ = regularized genre effect
e^r_{age} = regularized movie age effect ϵ = random residue

The results obtained as we applied each step in the modeling phase to the *test* set were presented in their respective sections. Here, we aggregate the results from all versions of model after application to the *train*, *test*, and *validation* sets:

modelId	RMSE (train set)	RMSE (test set)	RMSE (validation set)	Description
1	1.0603389	1.0603010	1.0612018	Global average (benchmark)
2	0.9421131	0.9440701	0.9440994	model 1 + simple movie effect
3	0.9421828	0.9439910	0.9440041	model 1 + regularized movie effect
4	0.8554965	0.8659097	0.8661594	model 3 + simple user effect
5	0.8560914	0.8656025	0.8657915	model 3 + regularized user effect
6	0.8477618	0.8614645	0.8615420	model 5 + regularized drama/non-drama effects
7	0.8182396	0.8496471	0.8496989	model 5 + regularized genre effect through PCA
8	0.8177938	0.8491858	0.8492470	model 7 + regularized movie age effect
9	0.7880624	0.8432272	0.8432228	model 8 + similar movies effect

We see from the table that the results obtained on the *validation* set are very close to the ones obtained on the *test* set, which indicates that the validation strategy (partitioninig the `edx` dataset into a *train* and a *test* sets) was appropriate and that the tuning parameters defined for each aspect of the model are robust and not subject to overtraining. The results on the *train* set are considerably better, which does indicate that they are not representative of what should be expected when exposing the model to new datasets.

We also see from the results that the improvement to the RMSE as new effects are incorporated into the model are very subtle and seem to converge to around 0.84. This reflects the fact the ratings have an intrinsic level of random variability and that there is a limit to the level of precision that any model can attain.

However, this subtle improvements to the RMSE should not be interpreted as insignificant performance improvements in practice. Ultimately, the goal of these models is not to predict ratings, but rather to create movie recommendations for each particular user. Because the objective is to use the models to create an ordered list of movie recommendations, a small change in the predicted ratings can have a big impact in the order at which movies are recommended.

The table contains the top 20 recommendations given to a randomly selected user by 3 of the models and illustrates this feature:

Table 21: Movie recommendations from 3 models to a random user

Model 5: regularized user effect	Model 7: regularized genre effect	Model 9 (complete model): similar movies effect
Schindler's List	Rashomon (RashÃ´mon)	Rashomon (RashÃ´mon)
Casablanca	Chinatown	Chinatown
Sunset Blvd. (a.k.a. Sunset Boulevard)	Rear Window	Maltese Falcon, The (a.k.a. Dangerous Female)
Rear Window	Third Man, The	Third Man, The
Seven Samurai (Shichinin no samurai)	Vertigo	Rear Window
Double Indemnity	Big Sleep, The	Vertigo
Third Man, The	Rebecca	Big Sleep, The
Paths of Glory	Witness for the Prosecution	Vanishing, The (Spoorloos)
General, The	Vanishing, The (Spoorloos)	Witness for the Prosecution
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb	Maltese Falcon, The (a.k.a. Dangerous Female)	In a Lonely Place
One Flew Over the Cuckoo's Nest	Diabolique (Les Diaboliques)	Rebecca

Model 5: regularized user effect	Model 7: regularized genre effect	Model 9 (complete model): similar movies effect
Yojimbo	Testament of Dr. Mabuse, The (Das Testament des Dr. Mabuse)	Lady Vanishes, The
Dark Knight, The	Oldboy	Diabolique (Les Diaboliques)
Lives of Others, The (Das Leben der Anderen)	Orphanage, The (El Orfanato)	Diva
Wallace & Gromit: A Close Shave	L.A. Confidential	39 Steps, The
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva)	Lady Vanishes, The	L.A. Confidential
Human Condition III, The (Ningen no joken III)	Yojimbo	Oldboy
M	In a Lonely Place	Dark Knight, The
Wallace & Gromit: The Wrong Trousers	Diva	Yojimbo
Big Sleep, The	To Catch a Thief	Strangers on a Train

Finally, the result of the RMSE after application of the complete model to the *validation* set is approximately 0.843.

Conclusion

This project covered the construction of a statistical model to predict movie ratings. A *train* set of approximately 7.2 million ratings was used to create predictions with parameters tuned with a *test* set of approximately 1.8 million ratings. Finally, model performance was measured on a *validation* set with approximately 1 million ratings. The Root Mean Squared Error (RMSE) was used as a metric, and a RMSE of 0.843 was reached with the complete model in the *validation* set.

Models were created in the Modeling section based on trends identified in the Exploratory Data Analysis section. Increasingly complex models were created by including terms that accounted for specific trends present in the residuals from the previous model.

In a practical setting, the results of the prediction models would be used to predict ratings to movies that have not already been watched/rated by a particular user. Movies would be ordered according to their predicted ratings and recommended to users. In this sense, the prediction and its RMSE are not the actual outcome of the model. Instead, a proper outcome would be a list of recommended movies.

Considering this fact, a list of recommendations generated by some of the models for a random user was also presented in the Results section. We notice that, even though the RMSE decreases very slightly with each step in the modeling process, the list of recommended movies changes more drastically.

One point not taken into consideration during the project is the fact that, in a practical setting, there are sources of "voting" other than the ratings given by users. Many times, users do not go through the trouble of rating a movie they've just watched. User are also not very consistent in givin ratings, and there is an intrinsic random component to a rating. A potentially more illustrative indication of an user's preferences may be the list of movies he or she have watched, regardless of ratings.

We also note that the strategy for partitioning the full set of ratings may be somewhat misleading. We randomly selected proportions of the original *MovieLens 10M* dataset, but a more relevant manner of partitioning the dataset would be to consider the date of rating. For instance, we could have set apart ratings from the last year in the dataset for validation. With this approach, we wouldn't be using future ratings to predict past ratings.

A potential point of improvement would be to perform further training on the collaborative filter portion of the model. The number of movies in the set of popular movies, used as reference for calculating a matrix of correlations and identifying similar movies was chosen from limited experimentation with a few set of values. The sizes of the users and movies partitions were also not fully explored to produce results with the lowest RMSE.

This was mostly due to the high demand for computational power. Calculating the predictions for one single set of values took nearly one hour in a home computer, so performing training with a large set of parameters would have been impractical. One possible solution would be to make use of the parallel computing capabilities in the `caret` package, which allows for the evaluation of performance for different combinations of parameters to be evaluated in parallel using additional processor cores.