MovieLens report

Nina Caparros

2019-10-30

Contents

1 Introduction

The following report is the analysis and results of the MovieLens Assessment of the Data Science Program of HarvardX, available on Edx (https://https://courses.edx.org/courses/course-v1:HarvardX+PH125.9x+2T2019/course/).

The purpose of this project was to develop a movie recommendation system using a subset of the MovieLens dataset. MovieLenses datasets were available on grouplens.org and several sizes were at one's disposal. As required in the assessment, one was using the MovieLens 10M Dataset, which provides roughly ten millions (9000055) of movie ratings. The initialization of the dataset was provided at the begining of the assessment by HarvardX. Each row of the dataset represented a rating, of one movie, by one user, at a certain time.

The goal of this assignment was to fit a model that would give a RMSE (Root Mean Square Estimate, see section 6.2. RMSE) of less than 0.8649.

2 Overview

The dataset initialized was made of six columns as follow :

- userId : the identifier of the user relative of the rating in the row
- movieId : the identifier of the movie rated in the row
- rating : the rating given by the user, which can take values from 0 to 5, as whole star ratings (0 to 5) and half star ratings (0.5 to 4.5)
- timestamp : the date and time as a timestamp at which the user left it's rating
- title : the title and year of release of the movie rated
- genres : the genre or genres of the movie rated

Parameter	Class	Distinct values	Minimum value	Maximum value
userId	integer	69878	Not relevant	Not relevant
movieId	numeric	10677	Not relevant	Not relevant
rating	numeric	9000055	0.5	5
timestamp	integer	9000055	789652009	1231131736
title	character	10677	Not relevant	Not relevant
genres	character	797	Not relevant	Not relevant

The edx dataset looked like :

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

A first glance at the current dataset showed that :

- The timestamp information clearly appeared uninterpretable.
- The information of the year of release was contained in the title column.

3 Preparation

Before further investigation, one needed to clean those timestamp values and extract the release's year.

3.1 Data cleaning

The first step one took was converting the `timestamp` column into a `date` (Year-Month-Day) and `time` (Hour:Minute) columns. The original `timestamp` column was removed. This step allowed one to analyse datas by date and hour of the day. The validation dataset had been cleaned in the same way.

One wondered if the time of the day, the day of the month, the month of the year, or even the year were influencing the ratings of the users. Could the date influence the rating of one given user ? Could the rating of that particular user change depending of the period ?

The next step was to extract the year the movie was released and add it to both the `edx` and `validation` datasets. In order to do this, one had to process the `title` column, which contained the title and the year of release between parenthesis. The `title would` contain only the title, and the new column `yearOfRelease` would contain the year previously stored in the title.

As some titles had some string characters between parenthesis, and since the structure of the title (`title (year of release)`) was always the same, one decided to simply use `substring` instead of a `regex`.

The cleaned dataset looked like :

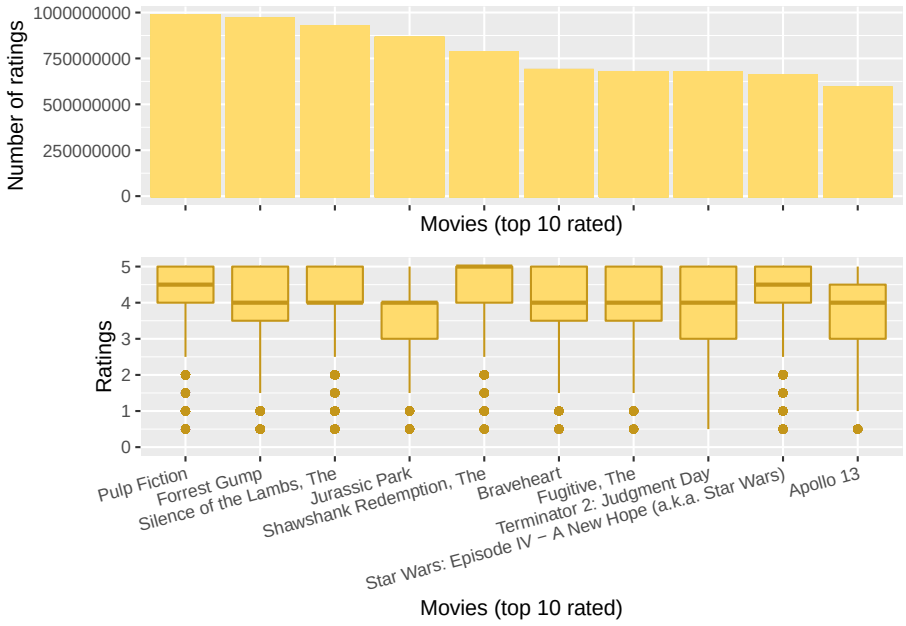
```
##   userId movieId rating                  title
## 1      1     122      5              Boomerang
## 2      1     185      5               Net, The
## 3      1     292      5               Outbreak
## 4      1     316      5               Stargate
## 5      1     329      5 Star Trek: Generations
## 6      1     355      5       Flintstones, The
##                            genres       date  time yearOfRelease
## 1                  Comedy|Romance 1996-08-02 11:24          1992
## 2           Action|Crime|Thriller 1996-08-02 10:58          1995
## 3  Action|Drama|Sci-Fi|Thriller 1996-08-02 10:57          1995
## 4         Action|Adventure|Sci-Fi 1996-08-02 10:56          1994
## 5 Action|Adventure|Drama|Sci-Fi 1996-08-02 10:56          1994
## 6       Children|Comedy|Fantasy 1996-08-02 11:14          1994
```

4 Analysis

This section described the insights one got of the data. It showed correlations and effects which were to be kept for the final model, or the ones rejected.

4.1 Movies

The `edx` dataset provided ratings for 10677 different movies. The following charts showed the number of ratings and their distribution for the ten most reviewed movies.

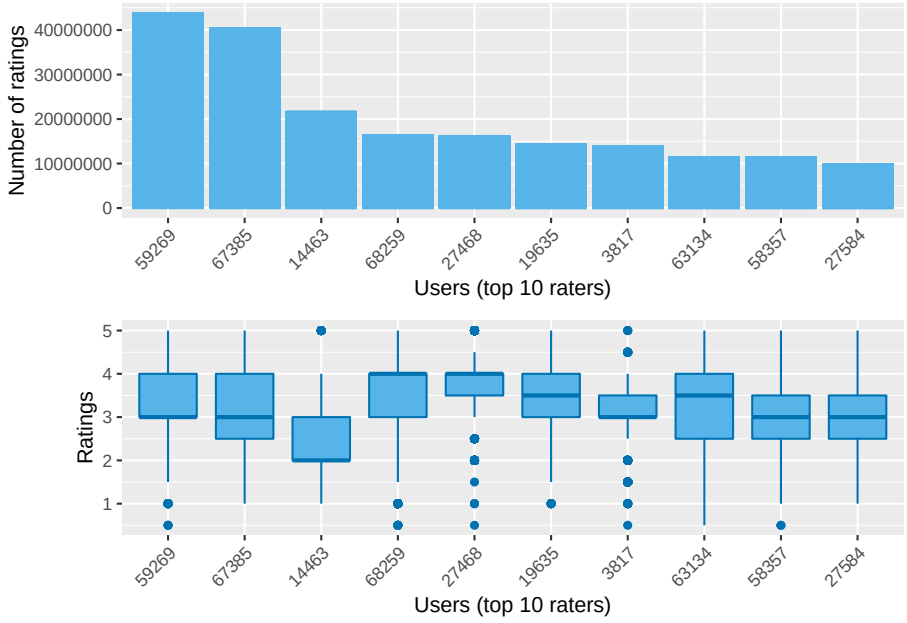


It appeared that the most rated movies were not necessarily the best rated movies, and the distribution of the ratings varied substantially. For instance, in the right plot, the first most rated movie (Pulp Fiction) has 50% of it's ratings between 4 and 5, with a median of 4.5, while Jurassic Park, the fourth most rated movie has 50% of it's ratings between 3 and 4, and 25% of it's ratings equal to 4. The Shawshank Redemption (fifth most rated movie) even has 25% of it's ratings equal to 5.

The movie bias was obvious and was to be introduced later in the 5.2 Effects section.

4.2 Users

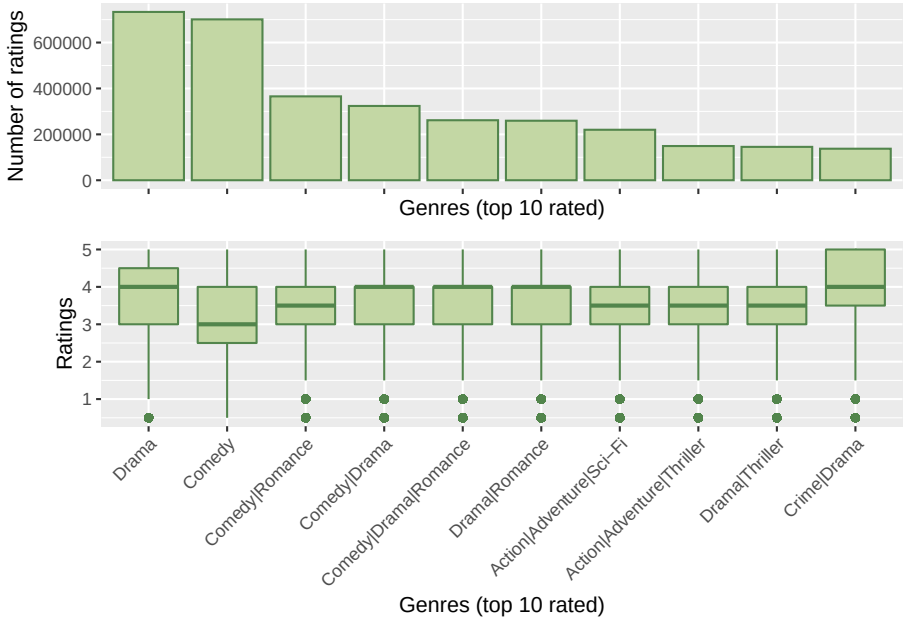
Each movie rating was associated to a user. Users could rate one or several movies, and the following charts shows the number of ratings, for the ten most prolific raters, and the ratings of those users.



It was clear that the number of ratings did not affect the ratings by user. Some users tend to be more severe than others, and some tend to be give high ratings more easily. This induced a user bias that was described and used in the 5.2 Effect section.

4.3 Genre

Since we all tend to appreciate some genres more than others, it was logical to analyse the ratings depending of the genre. The following plots showed respectively the number of ratings versus the genre, and the repartition of the ratings by genre, for the 10 most reviewed genres.



It appeared that there was no clear relationship between the number of ratings and the ratings. The most reviewed genre is Drama (25% of it's ratings between 4 and 4.5), but it seemed to be rated more harshly than the combination Crime/Drama (25% of it's ratings between 4 and 5). Knowing that Drama alone had been reviewed 5.34 times more than Drama/Crime.

Comedy, which was the second most reviewed genre, very close to Drama, has a median rating of only 3, with a rather large interquartile range (25% between 3 and 4), but combined with at least another genre (Drama, or Drama and Romance for example), it's median rating went up (25% of it's rating are equal to 4 for the two said combination).

These observations led one to wonder about two biases : genre and the number of genre of each movie. The genre bias was to be detailled in the 5.2 Effects section.

4.3.1 Primary genres

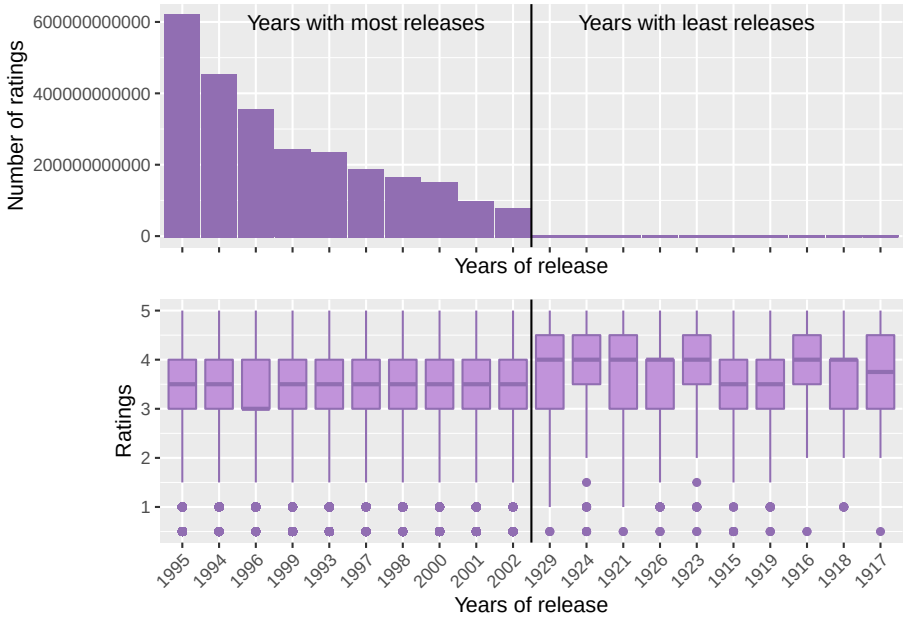
The genres were classified as a combination of 20 primary genres, with a total of 797 distincts combinations. One approach could have been to extract for each movie all the genres related to that movie, as, for instance, 20 columns as `isDrama`, `isComedy`, ... and then grouping the movies 20 times, with each parameter. This solution, though explored, had not been kept, as it required 20 `group_by`. Instead, one chose to use the 20 combinations, as a single `group_by(genres)`. This method was indeed less precise, but required a significantly less amount of

time to compute. If one had to do the more precise prediction, the extraction of the genres would have been chosen.

4.4 Year of release

Intuitively, it was easy to figure that the year the movie was released influenced on the rating of that movie. Some users were 90' movie fans, some prefered the recent ones, some, the old ones, and since trends come and go, time had an effect on the rating of the movie. Some masterpieces became outmoded, and some never grew old.

The following plots showed the number of ratings, for the ten most prolific years in reviewed movies, and the ten least reviewed years, and the repartition of the ratings.



A slight bias, depending of the year of release could be seen, even if it was not as strong as expected. The more ratings, the more the repartion of those ratings seemed to be similar : median of 3.5, and 50% of the ratings between 3 and 4.

4.5 Date and time

Several questions about the date and time were raised :

- Could the ratings of a user for the same movie be different depending of the period ?

- Could the overall ratings be influenced by the year, month or day ?
- Could the ratings for a same movie evolve with time ?
- Could the ratings of a user evolve with time ?

Considering our own experiences, we could intuitively say yes to these interrogations.

- We sometimes liked a specific movie more or less as our age, situation, mood, and tastes evolve.
- Our liking could be influenced by :
  - The day of the week (weekend), or the month (holiday), the season (Christmas movies in winter for example).
  - We could be harsher in our ratings based on exterior events (crisis, war, attack,...), or in contrast, more lenient (social win, end of said crisis or war,...). This was a theory to be tried out (see 4.5.1 Exterior events section).
- Some movies trending at a specific period were liked less and less with time, and on the other hand, some movies needed time to be appreciated.
- We could evolve to become more critical or more lenient with time, after watching enough movies.

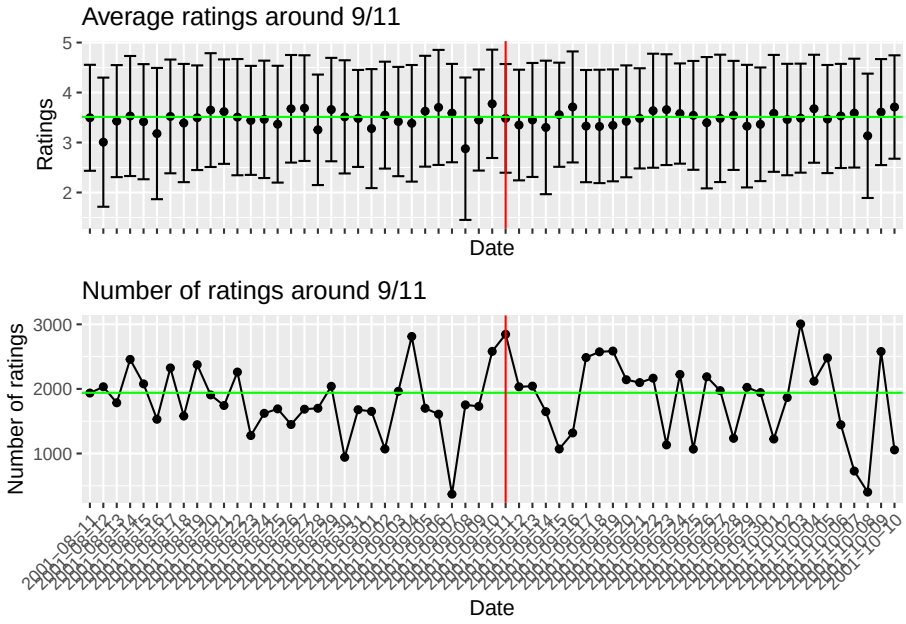
4.5.1 Exterior events

One was considering a comparison between the month preceding a said event, and the following month, trying to find out if that specific event could alter the movie ratings given.

4.5.1.1 9/11

On 11 September 2001, four coordinated terrorist attacks stroke the United States. Two planes crashed into the Twin Towers of the World Trade Center, and two crashed into the Pentagon. Even though terrorist attacks happened all around the world, it was the first one to happen on American soil. The horror of the war deeply affected Americans.

The following plots showed the average ratings, and the number of ratings from 2001-08-11 to 2001-10-10, one month before and after the 9/11 attacks. The vertical red lines represented the 2001-09-11, and the horizontal green lines the overall averages, respectively ratings and number of ratings.



No specific pattern was to be seen, yet it was noticeable that on the day of the attacks, the number of reviews was much higher than the average. This was surprising, it could have been supposed that on a tuesday, whith an event that traumatizing, people would have been less prone to watch movies.

The theories that emerged were :

- The users were not necessarily Americans, therefore less concerned by this event
- The effects of this event were not the one expected (higher number of reviews)
- The events may not influence the ratings
- The timelapse chosen was to short to observe a changing in ratings

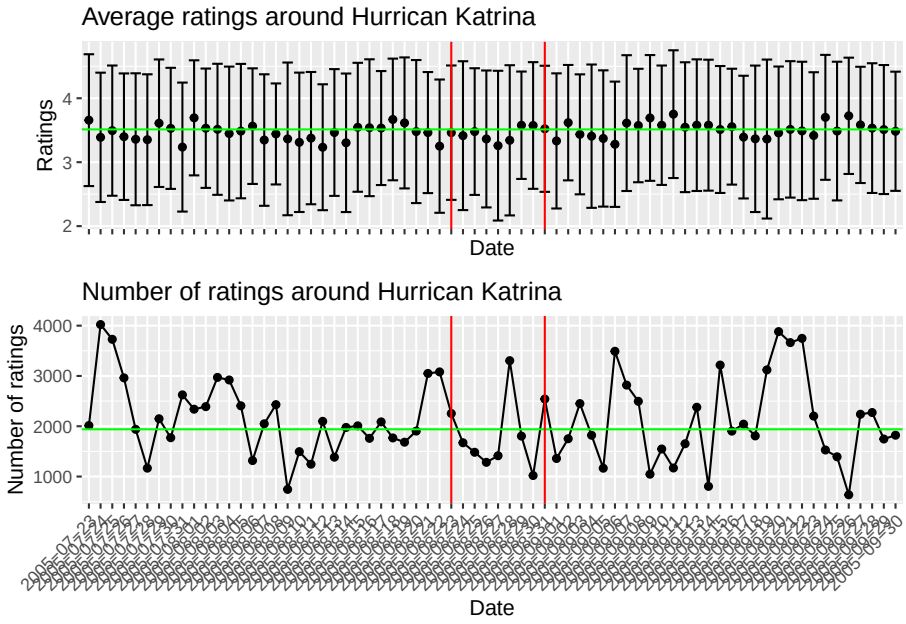
To confirm that events do not affect users ratings, one decided to analyse another event, a natural catastrophe, as the Hurricane Katrina, which devastated several states in America.

4.5.1.2 2005 Hurricane Katrina

From August 23 to August 31 of 2005, a category 5 hurricane made landfall on Florida, Gulf Coast, Louisiana, Mississipi. It was the third-most intense Atlantic hurricane at the time, and the damages reached $125 billion. 1,245 to 1,836 (estimations) people lost their lives to the hurricane, or the aftermath. People were evacuated, power was cut, cities were flooded and houses destroyed. It could have been expected that, in these conditions, the number of ratings would

have drastically diminished.

The following plots showed the average ratings and the number of ratings one month before through one month after hurricane Katrina. The horizontal green lines represented the overall averages, and the two vertical red lines delimited the duration of the hurricane.



Surprinsigly, there was no change in the average ratings following the landfall of the hurricane. The number of ratings did not decrease that much neither, a significant high rating could even be observed during the catastrophe.

It seemed the exterior event did not affect the ratings of the users. Consequently, there was no point trying to use the date of the rating in the prediction algorithm.

5 Methods

This section gathered the method kept for this assignment. The different biases were described, the training and testing sets were created, the tuning parameters determined.

5.1 Determining the model

Considering the rather large dataset at one's disposal, training algorithms as linear models cannot be used. One chose instead to fit the model :

$$Y_{u,i,...} = \mu + \sum_{n=u,i,...} b_n + \varepsilon_{u,i,...}$$

Where Y is the rating prediction knowing all the parameters $u, i, ...$, μ the average rating, and b the effects for each parameter. ε represents the error, or the residual. Since it was not possible to compute it, one was considering it as in the perfect scenario, which meant ε equal to zero.

This meant "the predicted rating for user `u` of the movie `i` is the average rating plus the sum of `n` biases for user `u` of the movie `i` plus the error for user `u` of the movie `i`".

This approach has a major inconvenient : it does not do well for small number of ratings. To regularize the biases (also called effects) `b`, a penalty term λ is introduced.

The penalty term λ minimizes the residual sum of squares plus that penalty term, known as the penalized least squares equation, as:

$$PLSE = \frac{1}{N} \sum_{u,i,...} (y_{u,i,...} - \mu - \sum b_{u,i,...})^2 + \lambda(\sum (b_i)^2 + \sum (b_u)^2 + \sum (b_{...})^2)$$

5.2 Effects

In order to fit the model described, one considered the biases analized previously:

- Movie bias
- User bias
- Genre bias

The year of release, the date and time, the number of genres were not kept as biases for the model, because they were too unsignificant (see 4.4 Year of release and 4.5 Date and time.)

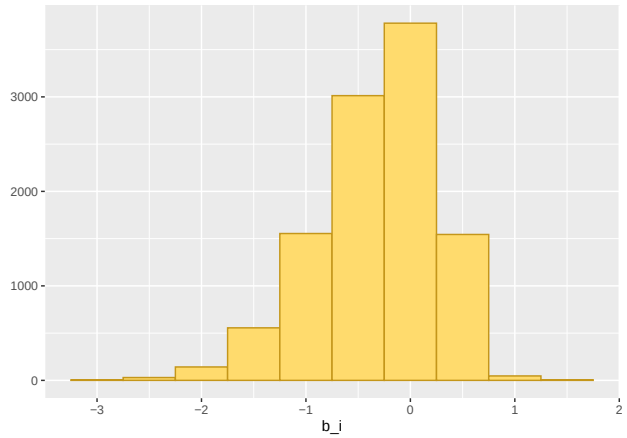
5.2.1 Movie effect

A movie bias `b_i` was defined for each movie, using a `group_by(movieId)`, as follow :

$$b_i = \frac{1}{N} \sum (r_i - \mu)$$

Where N was the number of ratings, r_i was each rating of movie i and μ the overall average rating.

The distribution of the b_i is shown in the following histogram.



This bias allowed the model to take into account the singularity of each movie, not only considering the overall average. The model introduced "good" and "bad" movies.

A positive `b_i` meant the movie `i` has an average rating above the overall average of 3.5124652. A negative `b_i` meant the movie `i` has an average rating below the overall average.

The value of b_i that minimized the penalized least squares equation was :

$$\hat{b}_i(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i} (Y_{u,i} - \hat{\mu})$$

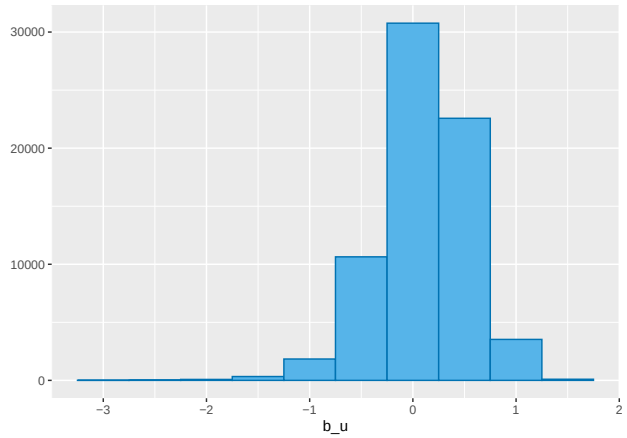
5.2.2 User effect

Simarly user bias `b_u` was defined for each movie, using a `group_by(userId)`, as follow :

$$b_u = \frac{1}{N} \sum (r_u - \mu)$$

Where N was the number of ratings, r_u was each rating from user u and μ the overall average rating.

The distribution of the b_u is shown in the following histogram.



This bias allowed the model to take into account the singularity of each user, not only considering the overall average. The model introduced "cranky" and "lenient" users.

A positive `b_u` meant the movie `u` has an average rating above the overall average of 3.5124652. A negative `b_u` meant the movie `u` has an average rating below the overall average.

The value of b_u that minimized the penalized least squares equation was :

$$\hat{b}_u(\lambda) = \frac{1}{\lambda + n_u} \sum_{u=1}^{n_u} (Y_{u,i} - \hat{\mu} - \hat{b}_i)$$

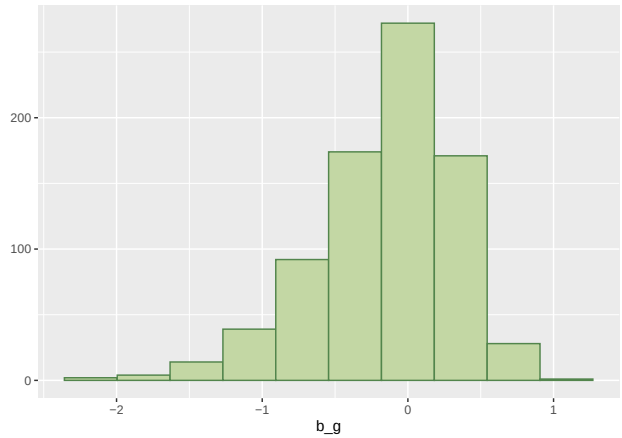
5.2.3 Genre effect

A gender bias `b_g` was defined for each genre, using a `group_by(genres)`, as follow :

$$b_g = \frac{1}{N} \sum (r_g - \mu)$$

Where N was the number of ratings, r_g was each rating for genre g and μ the overall average rating.

The distribution of the b_g is shown in the following histogram.



This bias allowed the model to take into account the singularity of each genre, not only considering the overall average. The model introduced the concept of "popular" and "unpopular" genres.

A positive b_g meant the genre g has an average rating above the overall average of 3.5124652. A negative b_g meant the genre g has an average rating below the overall average.

The value of b_g that minimized the penalized least squares equation was :

$$\hat{b}_g(\lambda) = \frac{1}{\lambda + n_g} \sum_{u=1}^{n_g} (Y_{u,i} - \hat{\mu} - \hat{b}_i - \hat{b}_u)$$

5.2.4 Others

Other bias were analyzed, as the year of release (see section Analysis), the date of the rating, and the number of genres of the movie. The results were not significant enough and thus not reported here.

5.3 Cross-validation

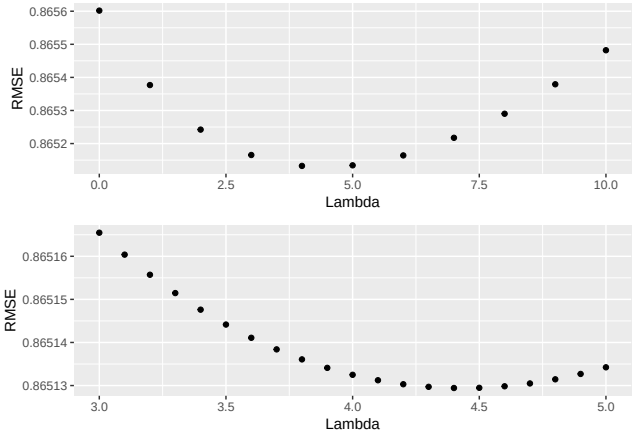
In order to pick the best fitted penalty term λ, one chose to use cross-validation. A training and testing set were created from the edx dataset, using the `createDataPartition` function.

The training set `train_edx` contained 8100067 rows, and the testing set `test_edx`, 899988.

5.4 Picking penalty term

The model described in the previous sections was transcribed into a function `functionRmses` which took a parameter `lambda`, computed the movie effect `b_i`, the user effect `b_u`, and the genre effect `b_g`, all regularized by `lambda` on the training set, and then predicted the ratings on the testing set. The function returned the RMSE (Root Mean Square Error, the standard deviation of the residuals) of the predictions made, and the actual ratings of the testing set.

In order to fasten the computing of the λs, two sets of λs were made. The first one, `lambdasRough`, was a sequence from 0 to 10 with an increment of 1. The function `functionRmses` was then used to compute the RMSEs of these rough lambdas using the function `sapply`. The minimum of the rough lambdas was kept, called `minRoughLambda`, and a second sequence, `lambdasFine`, from `minRoughLambda -1`, to `minRoughLambda +1`, with an increment of 0.1 was applied to the function `functionRmses`. The minimum of this second sequence was kept for the final model.



The λ which minimized the penalized least square equation was : 4.4.

6 Results

This section applied the previously defined model to the edx and validation datasets. The final RMSE was calculated.

6.1 Final model

The final model was :

$$Y_{u,i,g} = \mu + b_i + b_u + b_g$$

$$Y_{u,i,g} = \mu + \frac{1}{N_i + \lambda}\sum(r_i - \mu) + \frac{1}{N_u + \lambda}\sum(r_u - \mu) + \frac{1}{N_g + \lambda}\sum(r_g - \mu)$$

6.2 RMSE

It was time to apply the model to the edx and validation datasets.

```
#calculate mu the overall average of ratings on edx dataset
mu <- mean(edx$rating)

#movie effect
b_i <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n()+lambda), n_i = n())

#user effect
b_u <- edx %>%
  left_join(b_i,
            by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - mu)/(n()+lambda))

#genre effect
b_g <- edx %>%
  left_join(b_i,
            by="movieId") %>%
  left_join(b_u,
            by="userId") %>%
  group_by(genres) %>%
  summarize(b_g = sum(rating - b_u - b_i - mu)/(n()+lambda))

#predict ratings on the validation dataset
predicted_ratings <-
  validation %>%
  left_join(b_i,
            by = "movieId") %>%
  left_join(b_u,
            by = "userId") %>%
  left_join(b_g,
            by="genres") %>%
  mutate(pred = mu + b_i + b_u + b_g) %>%
  pull(pred)

#return the RMSE
rmse <- RMSE(predicted_ratings, validation$rating)
```

The final RMSE was 0.864456, which was less than 0.8649, thus the goal had been reached.

7 Conclusion

It was possible to determine a recommendation system with a simple mathematical model using biases using the formula :

$$Y_{i,u,g} = \mu + \lambda(b_i + b_u + b_g)$$

The time required for running the hole algorithm was decent, and the final RMSE for $\lambda = 4.4$ was 0.864456 was below the required value of 0.8649.

8 Openings

Even though the RMSE goal has been reached, the model described here was far from precise. First of all, the predicted ratings were (almost) never "real" values, as 0.5, 1, 1.5, ..., 4.5, 5. They were approximations implying the accuracy ($\frac{\sum(ratingPredictions == ratingValidation)}{nrow(Validation)}$) is equal to 0. It could have been more demonstrative to round the predictions to fit the possible rating values.

Rounding up the predicted ratings, the accuracy was 0.2485052, which was very low. Yet, since this was a recommendation system, and not a prediction system, the accuracy was not that important. The system had to offer to a user movies he or she could enjoy. It did not matter if the user rates the movie 0.5 more or less.

Another problem in the model was the fact that even if some users appreciated, in general, this genre of movie, it did not necessarilly mean they would enjoy all the movies of that particular genre.

Plus, some users appreciated some movies for their actors and actresses, independently of the genre, and this was not represented in the model.

Other recommendation systems could have been used, as Single Value Decomposition or Principal Component Analysis, but it required additionnal packages as `irlba`, `bigalgebra` and `bigmemory`. Since the RMSE goal had been hit, this approach was not analysed in this report.

9 Sources and references

9/11 : Wikipedia (https://en.wikipedia.org/wiki/September_11_attacks)

Choosing the penalty terms : Pr Irizarry from Harvard, on Edx (https://rafalab.github.io/dsbook/large-datasets.html#choosing-the-penalty-terms)

Generating edx and validation datasets : Pr Irizarry from Harvard, on Edx (https://courses.edx.org/courses/course-v1:HarvardX+PH125.9x+2T2019/courseware/dd9a048b16ca477a8f0aaf1d888f0734/e8800e37aa444297a3a2f35bf84ce452/2?activate_block_id=block-v1%3AHarvardX%2BPH125.9x%2B2T2019%2Btype%40vertical%2Bblock%40e9abcdd945b1416098a15fc95807b5db)

Hurricane Katrina : Wikipedia (https://en.wikipedia.org/wiki/Hurricane_Katrina)

Modeling movie effects : Pr Irizarry from Harvard, on Edx (https://rafalab.github.io/dsbook/large-datasets.html#modeling-movie-effects)

Penalized least squares : Pr Irizarry from Harvard, on Edx (https://rafalab.github.io/dsbook/large-datasets.html#penalized-least-squares)