“MovieLens Project”

Elena Oskrogo

31/12/2020

1 Introduction

For the project Movielens, I've created a movie recommendation system based on the MovieLens dataset. The dataset size is 10M. Data located at http://files.grouplens.org/datasets/movielens/ml-10m.zip.

I've used the course "PH125.8x: Data Science: Machine Learning by Professor R.Irizarry (Introduction to Data Science book on https://rafalab.github.io/dsbook/)" as the main inspiration for my development work.

The project aims to predict movie ratings based on available features and demonstrate knowledge and skills learned during the HarvardX Professional Certificate in Data Science program.

First, let's describe the dataset. Each line represents the rating of one user for one movie. Dataset movielens has 6 predictors and 10000054 observations.

Here is a preview of the dataset:

```
##    userId movieId rating timestamp                        title
## 1:      1     122      5 838985046             Boomerang (1992)
## 2:      1     185      5 838983525              Net, The (1995)
## 3:      1     231      5 838983392         Dumb & Dumber (1994)
## 4:      1     292      5 838983421              Outbreak (1995)
## 5:      1     316      5 838983392              Stargate (1994)
## 6:      1     329      5 838983392 Star Trek: Generations (1994)
##                           genres
## 1:                Comedy|Romance
## 2:         Action|Crime|Thriller
## 3:                        Comedy
## 4:  Action|Drama|Sci-Fi|Thriller
## 5:       Action|Adventure|Sci-Fi
## 6: Action|Adventure|Drama|Sci-Fi
```

The predictors are:

- **userId** is a unique user identification;
- **movieId** is a unique movie identification;
- **rating** is a value from 0.5 to 5 provided by user U for movie i ;
- **timestamp** is a date and time of the rating;
- **title** is a movie title;
- **genres** is a movie genre.

One of the first observations regarding data is the format of the **timestamp** field. I've applied the function *as_datatime()* to transform it into date format.

The column **title** has two entities: release year and movie name. I've separated the column into two parts - the movie title and release year.

The column **genres** combines several basic types like Comedy, Romance, Action, or Drama, and others. I've used it as it is, without splitting it into basic types.

First, I performed a descriptive data analysis to understood data, research missing values, clean, transform, and identify trends.

Second, I split the original dataset into training and validation sets.

After, I built the model to predict the movie rating based on available predictors. The highest rating suggested that the user will like the movie. I trained my model on the training data set and verified accuracy on the validation dataset. To compare different models, I used root mean squared error (RMSE) as a loss function. The objective was to obtain an RMSE of less than 0.86490.

As I inserted variable values directly in the report text, I did not use standard knit menu to create pdf output, but instead I used command *rmarkdown::render("file_name")* to compile in pdf output.

Lastly, I provided some conclusions about my findings and suggestions regarding future development.

2 Methods/ analysis

2.1 Dataset detailed description

First I've checked missing, null values and data with non expected format for each features. No anomalies were found.

Another way to verify missing value is used *summary()* function:

```
##      userId         movieId          rating        timestamp            title
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08   Length:10000054
##  1st Qu.:18123   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08   Class :character
##  Median :35741   Median : 1834   Median :4.000   Median :1.035e+09   Mode  :character
##  Mean   :35870   Mean   : 4120   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53608   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     genres
##  Length:10000054
##  Class :character
##  Mode  :character
##
##
##
```

I've did followed data transformation:

- field **timestamp** was convert to data format;
- field **title** was split into two columns: movie title (**Title**) and year of movie release (**YearM**);
- add new field **Age**, which is age of movie at the moment of rating;
- add new field **Word_title**, count number of worlds in move title.

The transformed dataset looks like, preview of first 6 lines:

```
##    userId movieId                  Title YearM                         genres rating
## 1:      1     122              Boomerang  1992                 Comedy|Romance      5
## 2:      1     185               Net, The  1995          Action|Crime|Thriller      5
## 3:      1     231           Dumb & Dumber  1994                         Comedy      5
## 4:      1     292               Outbreak  1995   Action|Drama|Sci-Fi|Thriller      5
## 5:      1     316               Stargate  1994        Action|Adventure|Sci-Fi      5
## 6:      1     329 Star Trek: Generations  1994 Action|Adventure|Drama|Sci-Fi      5
##                   date Age Word_title
## 1: 1996-08-02 11:24:06   4          1
## 2: 1996-08-02 10:58:45   1          2
## 3: 1996-08-02 10:56:32   2          2
## 4: 1996-08-02 10:57:01   1          1
## 5: 1996-08-02 10:56:32   2          1
## 6: 1996-08-02 10:56:32   2          3
```

and results of *summary()* function:

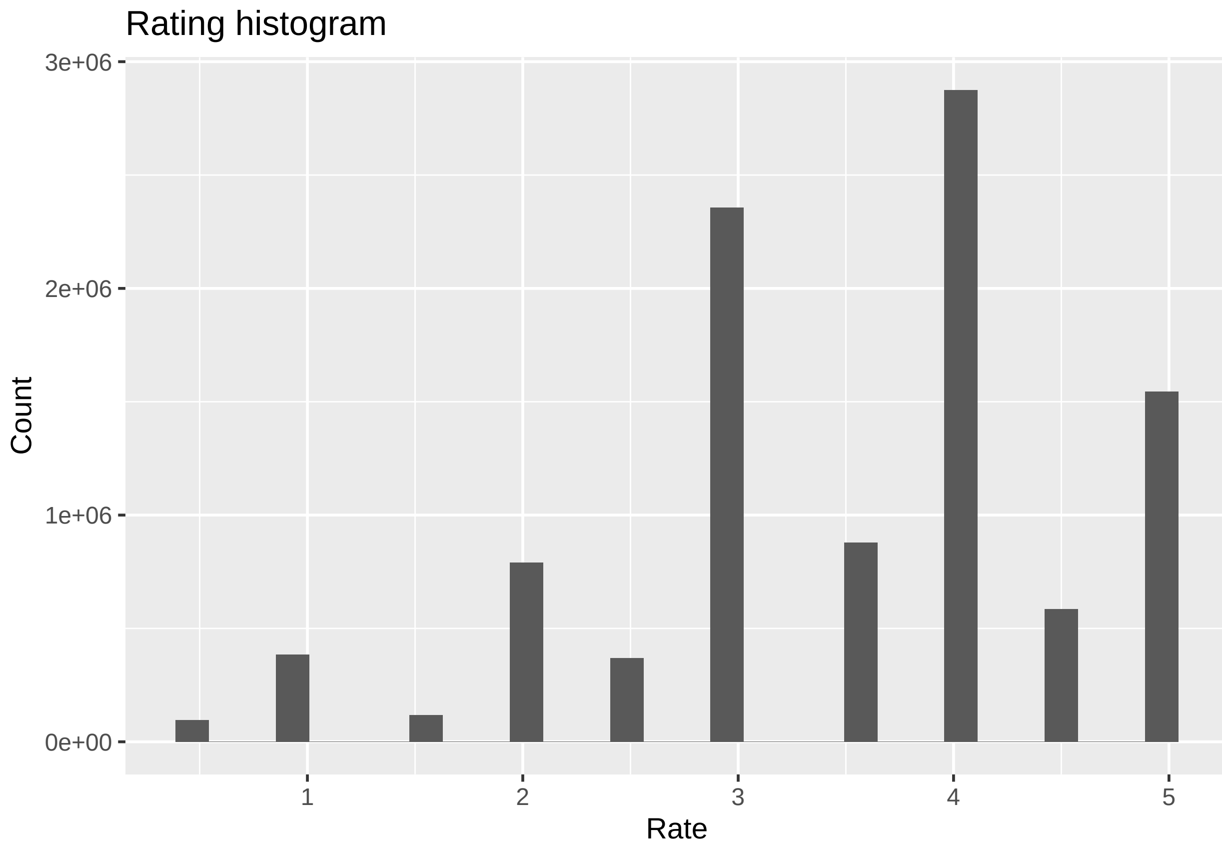
```
##      userId         movieId          Title               YearM         genres         
##  Min.   :    1   Min.   :    1   Length:10000054    Min.   :1915   Length:10000054   
##  1st Qu.:18123   1st Qu.:  648   Class :character   1st Qu.:1987   Class :character  
##  Median :35741   Median : 1834   Mode  :character   Median :1994   Mode  :character  
##  Mean   :35870   Mean   : 4120                      Mean   :1990                     
##  3rd Qu.:53608   3rd Qu.: 3624                      3rd Qu.:1998                     
##  Max.   :71567   Max.   :65133                      Max.   :2008                     
##      rating           date                          Age          Word_title    
##  Min.   :0.500   Min.   :1995-01-09 11:46:49   Min.   :-2.00   Min.   : 1.000  
##  1st Qu.:3.000   1st Qu.:2000-01-01 22:31:20   1st Qu.: 2.00   1st Qu.: 2.000  
##  Median :4.000   Median :2002-10-24 16:21:21   Median : 7.00   Median : 2.000  
##  Mean   :3.512   Mean   :2002-09-21 11:05:54   Mean   :11.98   Mean   : 3.057  
##  3rd Qu.:4.000   3rd Qu.:2005-09-15 01:51:10   3rd Qu.:16.00   3rd Qu.: 4.000  
##  Max.   :5.000   Max.   :2009-01-05 05:02:16   Max.   :93.00   Max.   :28.000  
```

Same verification (missed, null and different format values) was applied to transformed columns. Again, no anomalies were found. For the details please check code source.

Let's review more in details each of predictors.

From *summary()* function we can see that **rating** has values from 0.5 and 5 and 4 is most frequent rating (median value).

Let's build histogram to see the rating distribution.



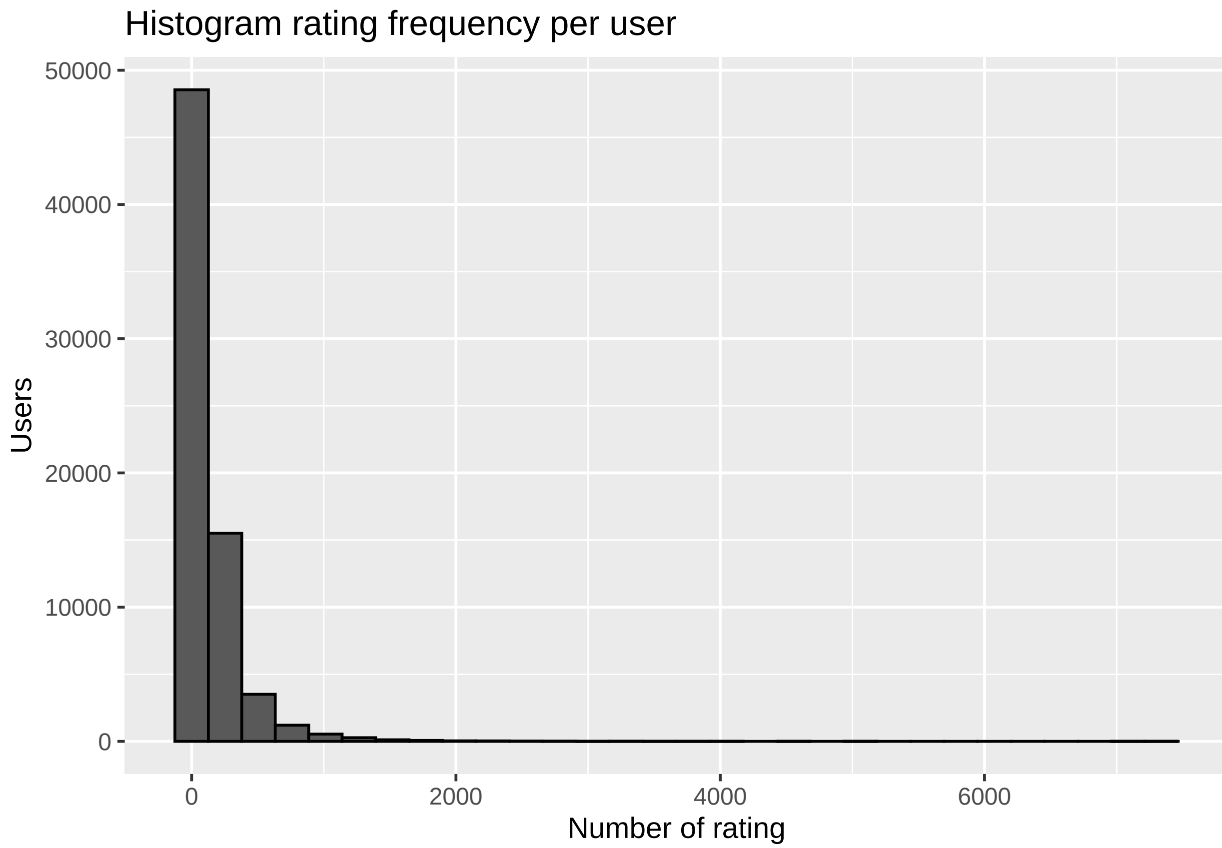
In majority of cases user tend to provide entire number as i.e. 3 then 3.5 . Also we can see that tendency is provide rather positive feedback, with median values 4.

Userid - represent identification of user who provided rating. From *summary()* function results we can see that userid is a number between 1 and 71567. Others information provided by *summary()* for **UserId** is not very useful, as data is identification code.

In Movielens dataset we have 69878 different users.

Not all users have same rating activity, as number of rating per users is vary between min 20 and maximum 7359.

Here is user rating activity:



As we can observe majority of users have small number of ratings, very active users rather exception. We can calculate that 61.53 provided less than 100 ratings and 92.32 provided less than 400 ratings.

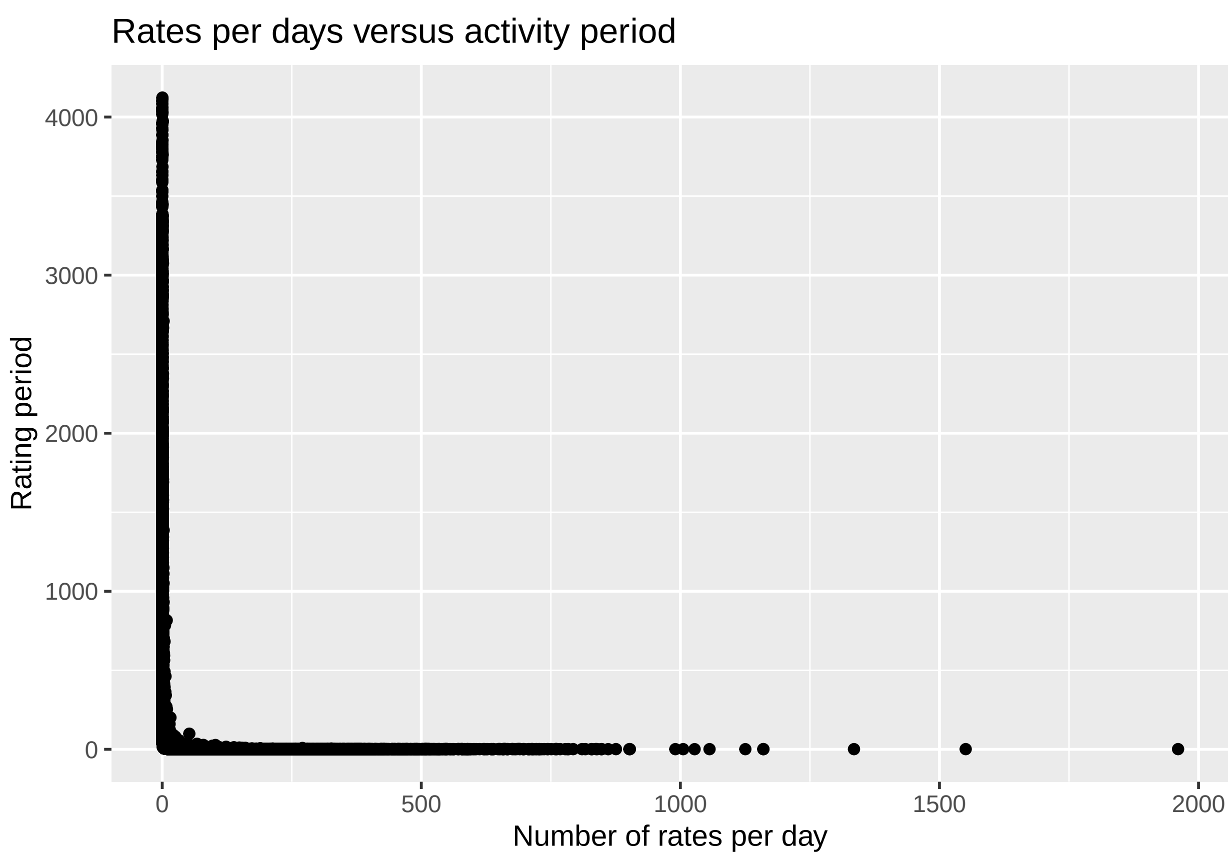
Another question that we can answer is for how long user provide their feedback and what are average feedback (rate) number per specific period of time (i.e. per day). For certain users time period when they provided movie rating is 1.00 days and for others it is 11.30 years.

I've build table to see how many users provided specific number of feedback per day, Here is sample:

From	To	Users
0	1	11008
1	2	3239
2	4	2711
4	6	1362
6	12	2182
12	Max	49376

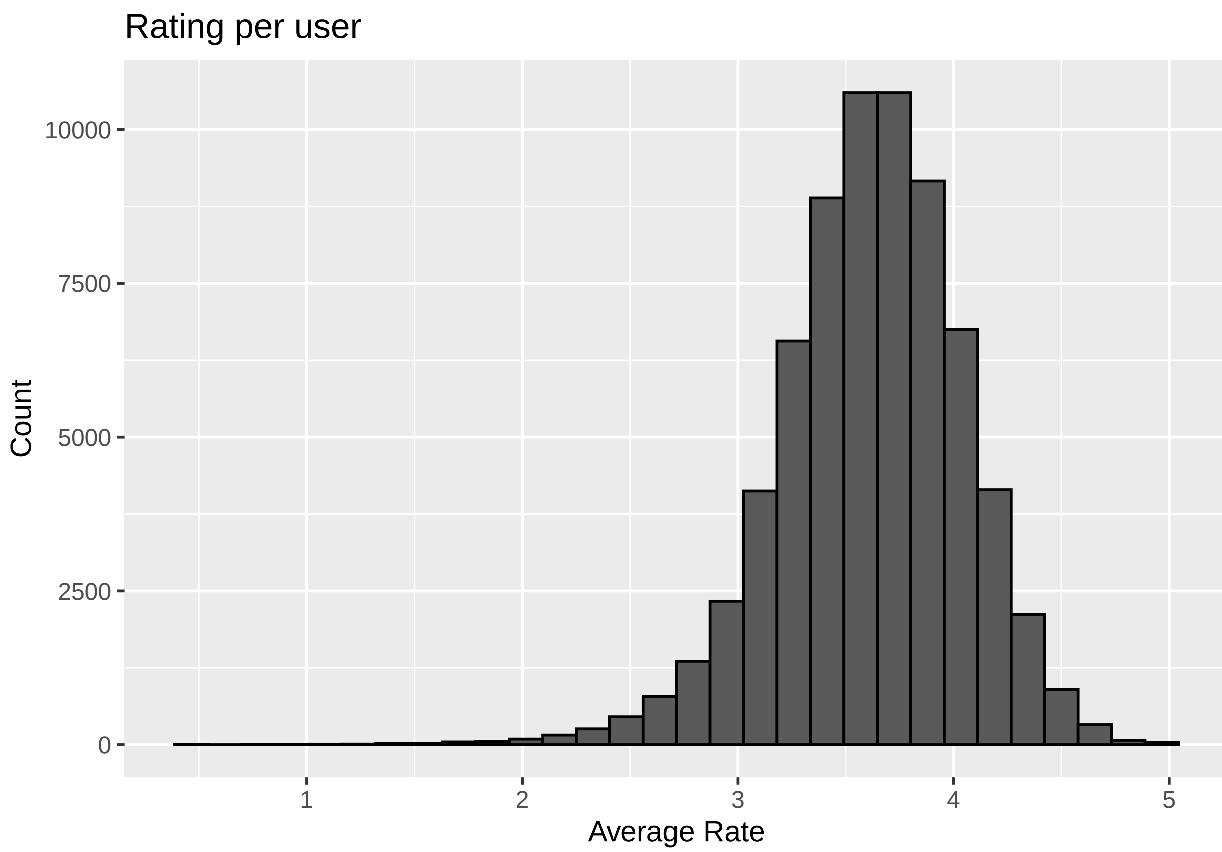
We can classify users based on number of feedback per day as normal, who i.e. provided up to 2 feedback per day in average, or very active movie watcher depending on daily rate. However it is difficult to imagine that person watch more than 12 movies per day. I can just conclude that Movielens is combination of user rating provided after watching movies and i.e. results of different surveys or even results of automated process .

Here is relationship between how long user provide feedback (difference between first rate date and date for latest rate) and number of rates per day:



From this plot we can observe that users with very high number of feedback per day tend to have very short period when they provided feedback. Users with smallest rate per day number tend to provide their feedback on longer periods.

Here is the distribution of average users's ratings:

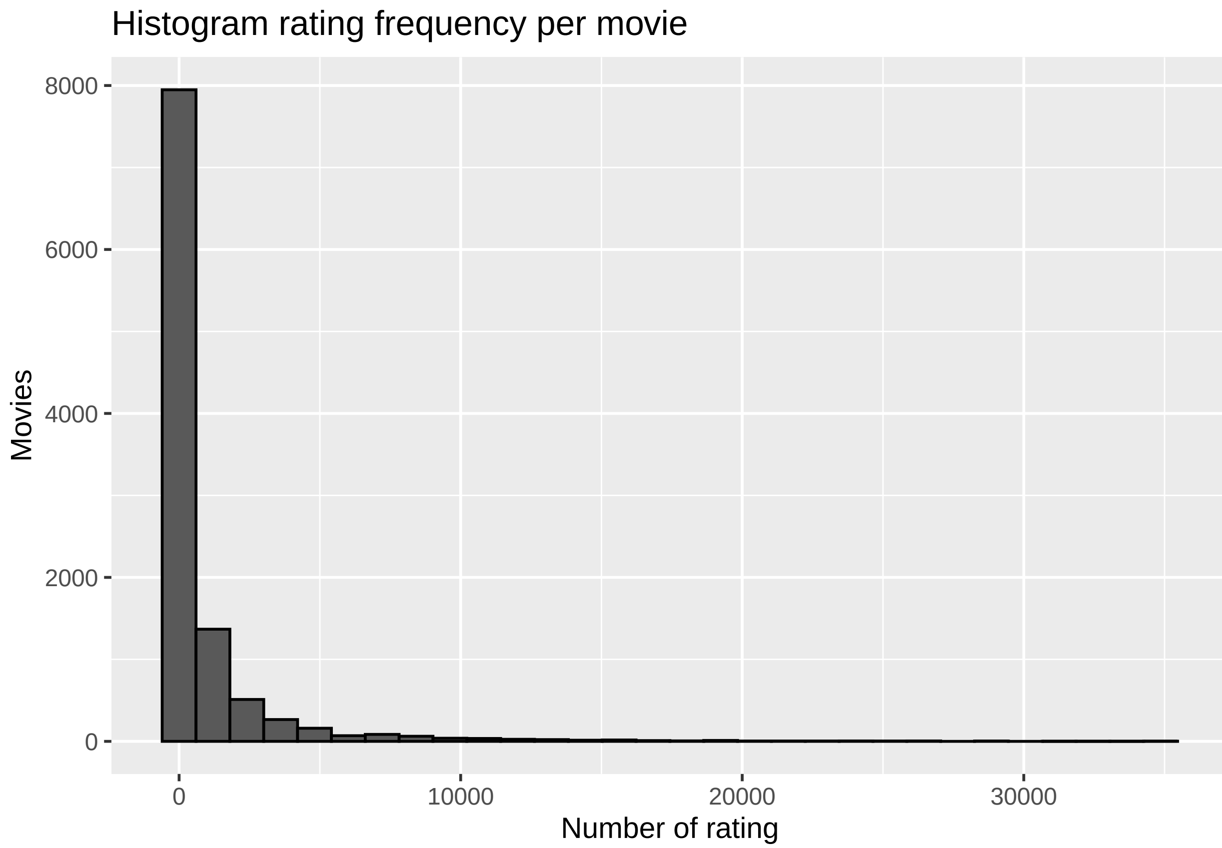


User's rating has slightly left skeed distribution. Maximum of user provide rating between 3 and 4, and we have more users with rating above 4 than less 3.

Rating	Users	Percentage
<= 3	5241	7.500215
> 3 & <= 4	52815	75.581728
> 4	11822	16.918057

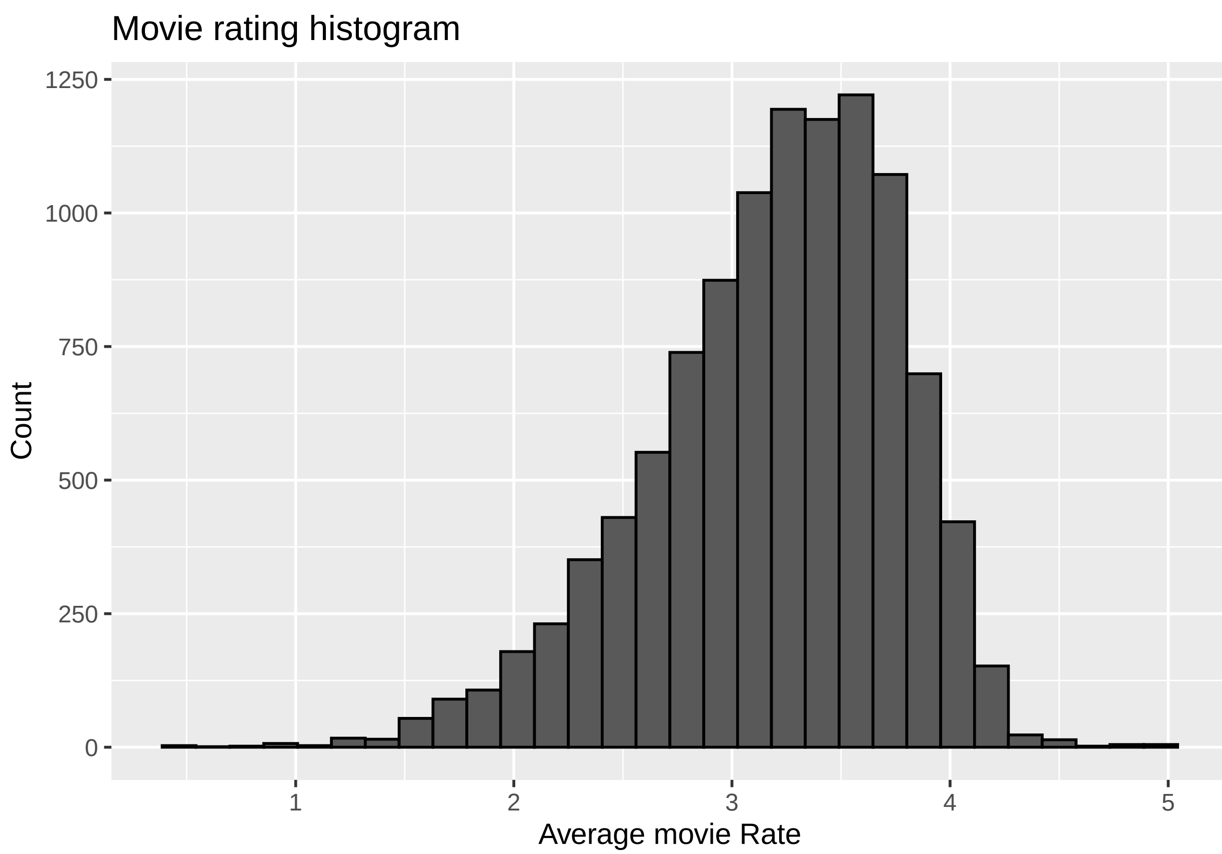
Another observation here or large part of movies is very appreciated, or that user tends to provide rating when then like movie. However we can see that we have important differentiation in each user feedback, some user are more generous in their rate and some are very critical.

MovieId - movie identification varies between 1 and 65133. Similar to user, we can find that dataset has 10677.00 different movies. Each movie has number of rating between min 1.00 and maximum 34864.00. Here is distribution about how many rating receive movie



We can observe that largest part of movies receive small number of rates.

Here is distribution of movie average ratings



We also can clearly see that even if majority of movies have average rating close to total average rating, they have individual impact in the same way as user. Some movies are more appreciated then others.

Heatmap for user- movie rating provide interesting inside on data that available for us:

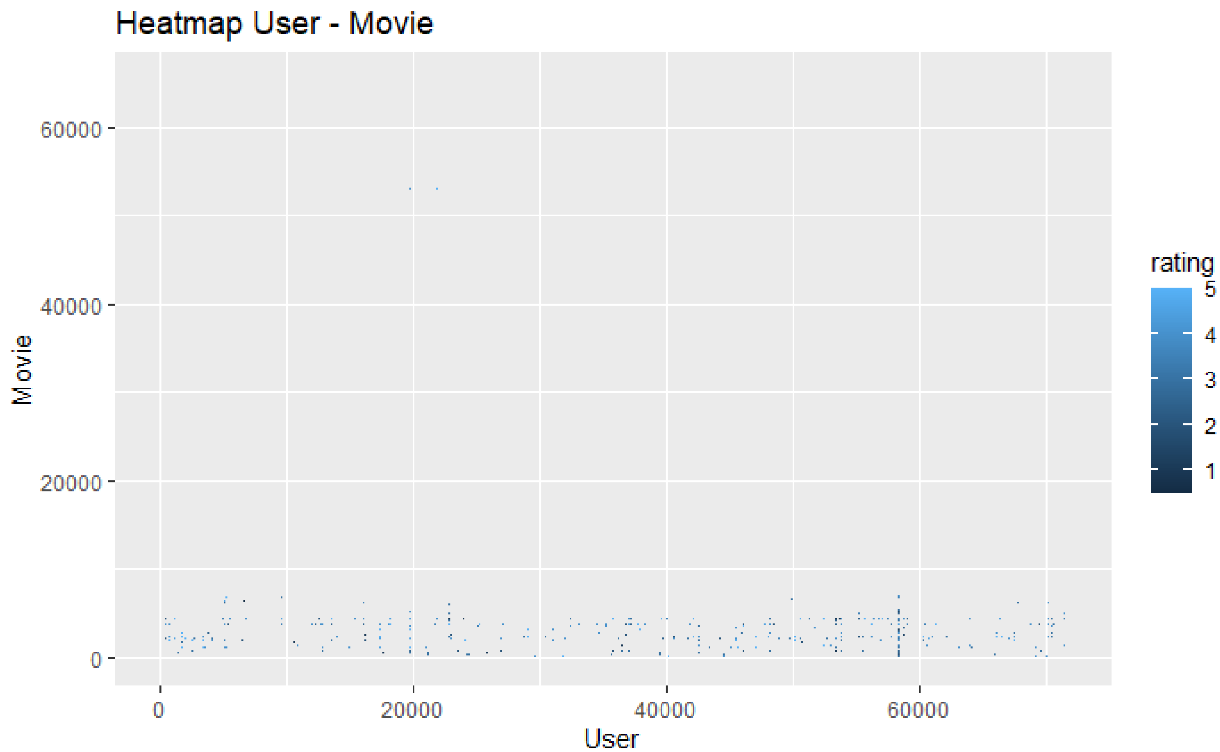
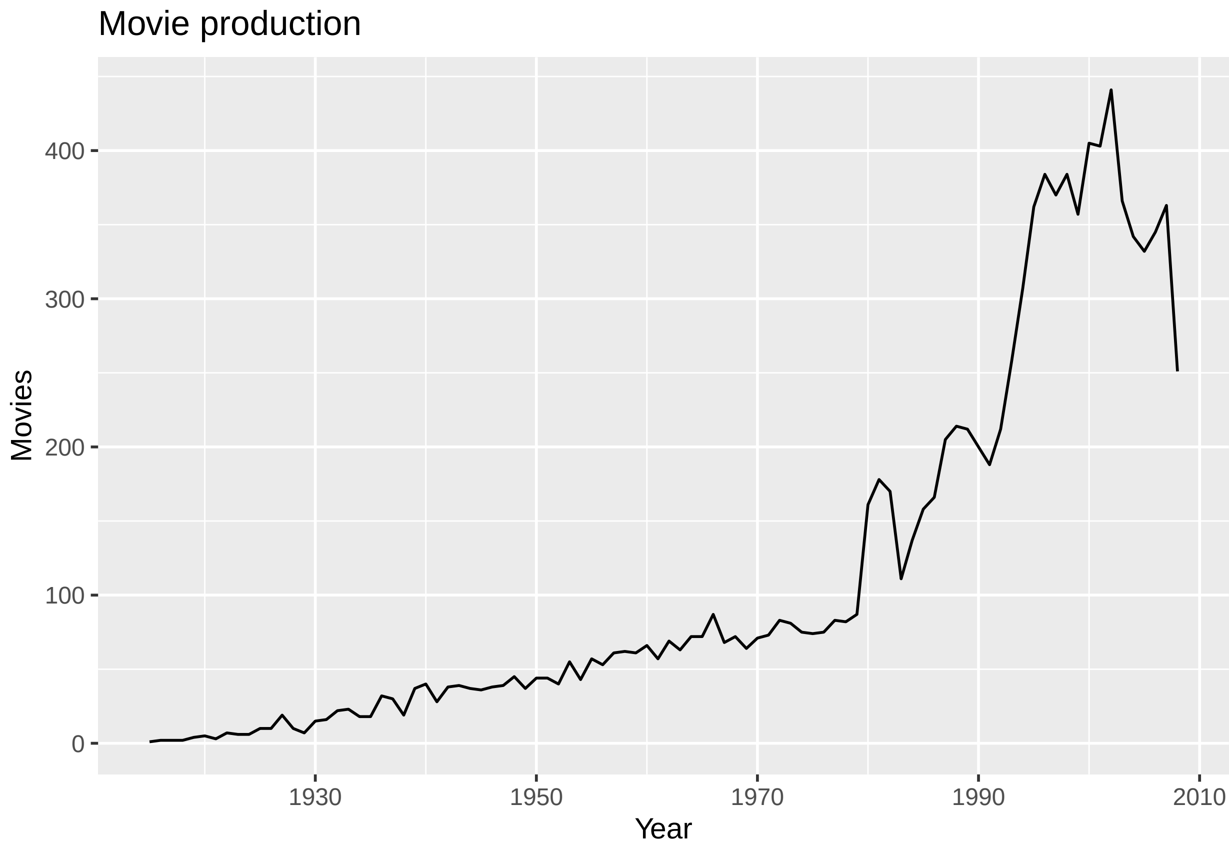


Figure 1: Heatmap User Movie

Heatmap provide interesting view of user-movie rating. We can see that majority of Movie - User missing rating and we do not know if this is due to fact that user does not watch movie or just does not provide feedback. I has technical problems to include heatmap results in R chunk, i.e. very long compilation time and pdf did not displayed result properly (long time prior plot is visible). As workaround I've generate heatmap in my R script, save results in file and add file as image in the report.

YearM - year of movie release. In movielens dataset we have movies between 1915 (the most old) and 2008 (the most recent movies), in total 93 years of movie production. We can also observe that 1994 is year with biggest movie releases and 75% of all movies were produced between 1915 & 1998. Here is movie release distribution:



We can also observe that until 2000 number of movies per year constantly increased and after 2010 this number is declined.

Genres - described movie category, each movie can have several genres combination. In total we have 797.00. Each genre has different number of movies, here is list of 10 genres with highest number of movies:

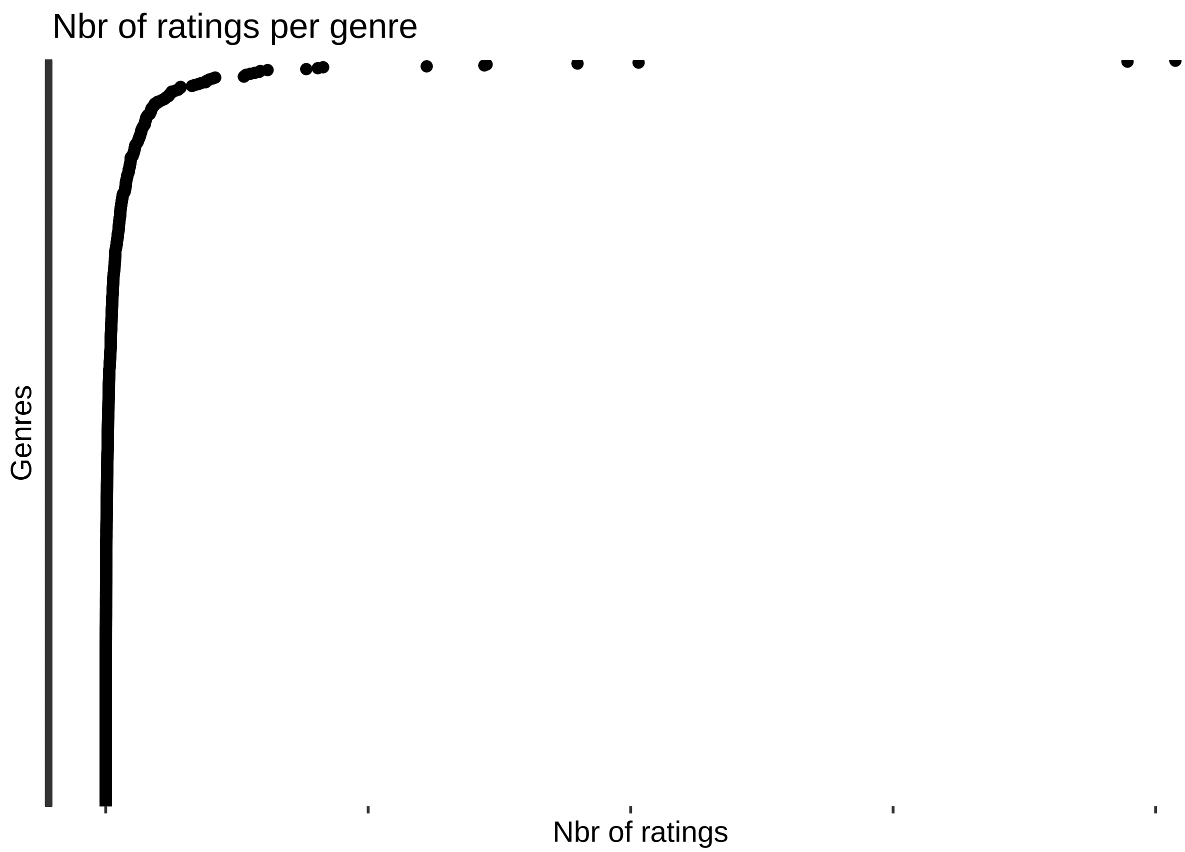
```
## # A tibble: 10 x 2
##    genres                 count
##    <chr>                  <int>
##  1 Drama                   1815
##  2 Comedy                  1047
##  3 Comedy|Drama             551
##  4 Drama|Romance            412
##  5 Comedy|Romance           379
##  6 Documentary              350
##  7 Horror                   267
##  8 Comedy|Drama|Romance     255
##  9 Drama|Thriller           192
## 10 Drama|War                173
```

and 10 genres with minimum number of movies:

```
## # A tibble: 10 x 2
```

```
##    genres                                              count
##    <chr>                                               <int>
##  1 (no genres listed)                                      1
##  2 Action|Adventure|Animation|Children|Comedy|Fantasy      1
##  3 Action|Adventure|Animation|Children|Comedy|IMAX         1
##  4 Action|Adventure|Animation|Children|Fantasy             1
##  5 Action|Adventure|Animation|Children|Sci-Fi              1
##  6 Action|Adventure|Animation|Comedy|Drama                 1
##  7 Action|Adventure|Animation|Comedy|Sci-Fi                1
##  8 Action|Adventure|Animation|Drama|Fantasy|Sci-Fi         1
##  9 Action|Adventure|Animation|Fantasy|Sci-Fi               1
## 10 Action|Adventure|Animation|Horror|Sci-Fi                1
```

Here we can see how frequently user provides rate per movie genre:



This rating numbers various between 2.00 (min value) and 815084.00 (max value). Top 10 genres with highest number of rating:

```
## # A tibble: 10 x 2
##    genres                       count
##    <chr>                        <int>
##  1 Drama                       815084
##  2 Comedy                      778596
```

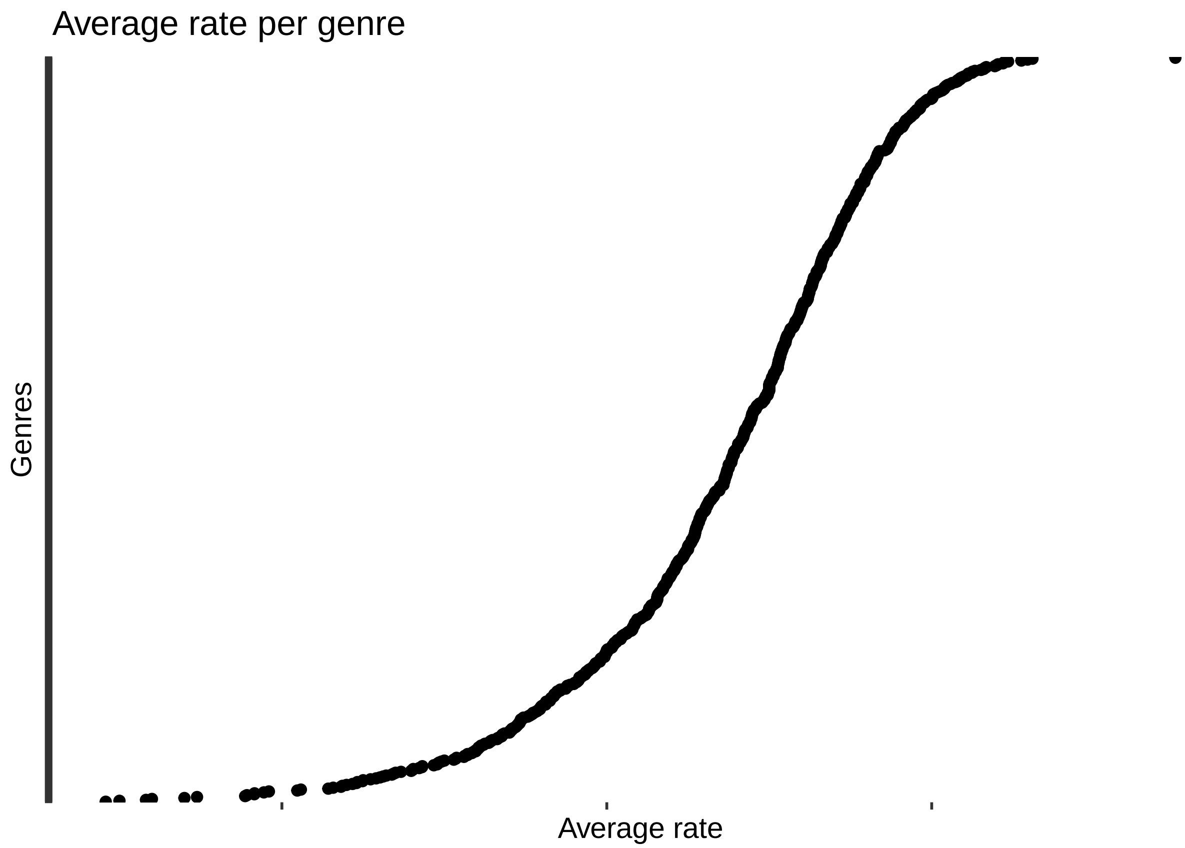
```
##  3 Comedy|Romance            406061
##  4 Comedy|Drama              359494
##  5 Comedy|Drama|Romance      290231
##  6 Drama|Romance             288539
##  7 Action|Adventure|Sci-Fi   244586
##  8 Action|Adventure|Thriller 165671
##  9 Drama|Thriller            161609
## 10 Crime|Drama               152827
```

10 Genres with less number of rating:

```
## # A tibble: 10 x 2
##    genres                                  count
##    <chr>                                   <int>
##  1 Action|Animation|Comedy|Horror              2
##  2 Action|War|Western                          2
##  3 Adventure|Mystery                           2
##  4 Fantasy|Mystery|Sci-Fi|War                  2
##  5 Crime|Drama|Horror|Sci-Fi                   3
##  6 Documentary|Romance                         3
##  7 Drama|Horror|Mystery|Sci-Fi|Thriller        3
##  8 Horror|War|Western                          3
##  9 Action|Adventure|Animation|Comedy|Sci-Fi    4
## 10 Adventure|Animation|Musical|Sci-Fi          4
```

As we can see Drama and Comedy obtain highest number of ratings, which is normal as they also have biggest number of movies in both categories.

Similar analysis on average rating per genres:



As we can see rating will depends on movie genres. With 10 top genres that receive higher rating

```
## # A tibble: 10 x 2
##    genres                                  average_rate
##    <chr>                                          <dbl>
##  1 Animation|IMAX|Sci-Fi                           4.75
##  2 Drama|Film-Noir|Romance                         4.31
##  3 Action|Crime|Drama|IMAX                         4.29
##  4 Animation|Children|Comedy|Crime                 4.28
##  5 Film-Noir|Mystery                               4.24
##  6 Crime|Film-Noir|Mystery                         4.22
##  7 Film-Noir|Romance|Thriller                      4.22
##  8 Crime|Film-Noir|Thriller                        4.20
##  9 Crime|Mystery|Thriller                          4.20
## 10 Action|Adventure|Comedy|Fantasy|Romance         4.19
```

And 10 less rated

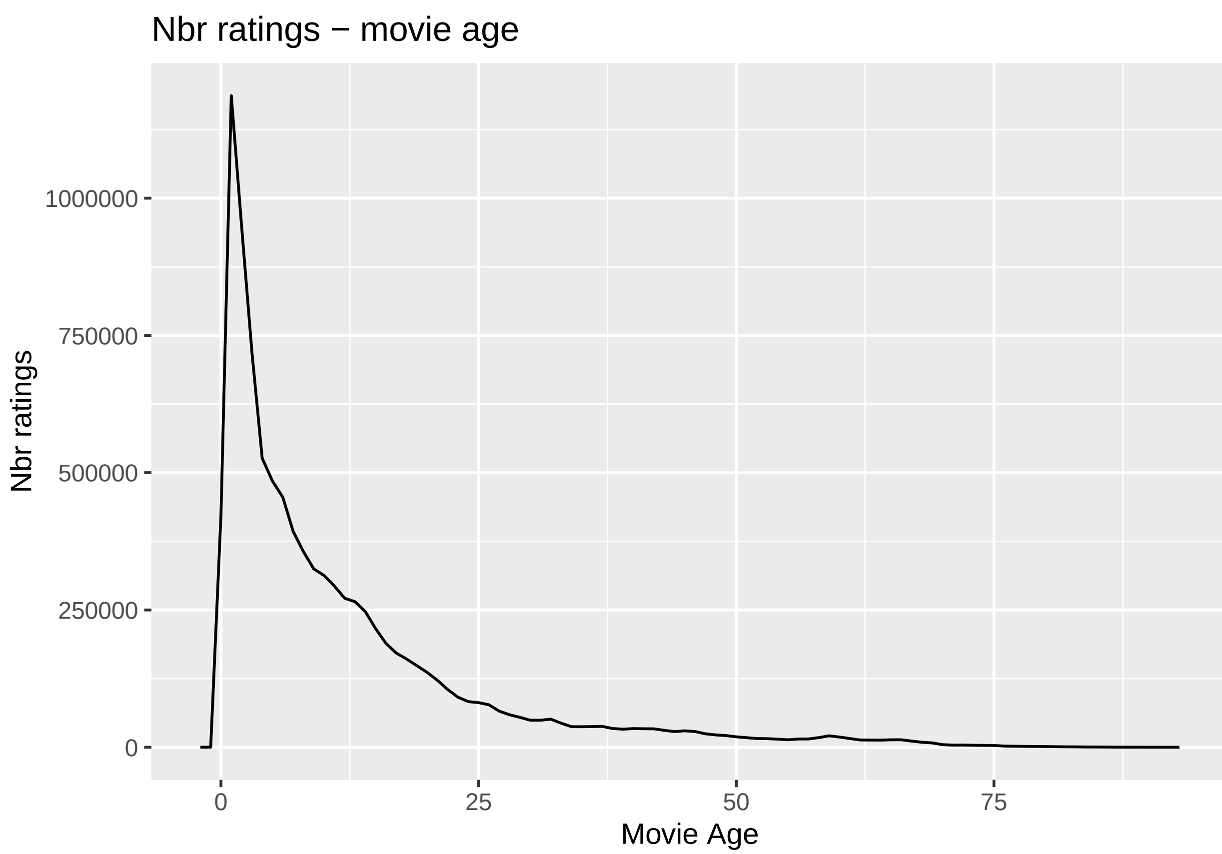
```
## # A tibble: 10 x 2
##    genres                                          average_rate
##    <chr>                                                  <dbl>
##  1 Documentary|Horror                                      1.46
##  2 Action|Animation|Comedy|Horror                          1.5
```

```
##  3 Action|Horror|Mystery|Thriller                          1.58
##  4 Action|Drama|Horror|Sci-Fi                              1.6
##  5 Comedy|Film-Noir|Thriller                               1.7
##  6 Adventure|Drama|Horror|Sci-Fi|Thriller                  1.74
##  7 Action|Children|Comedy                                  1.89
##  8 Action|Adventure|Drama|Fantasy|Sci-Fi                   1.89
##  9 Adventure|Animation|Children|Fantasy|Sci-Fi             1.92
## 10 Action|Adventure|Children                               1.92
```

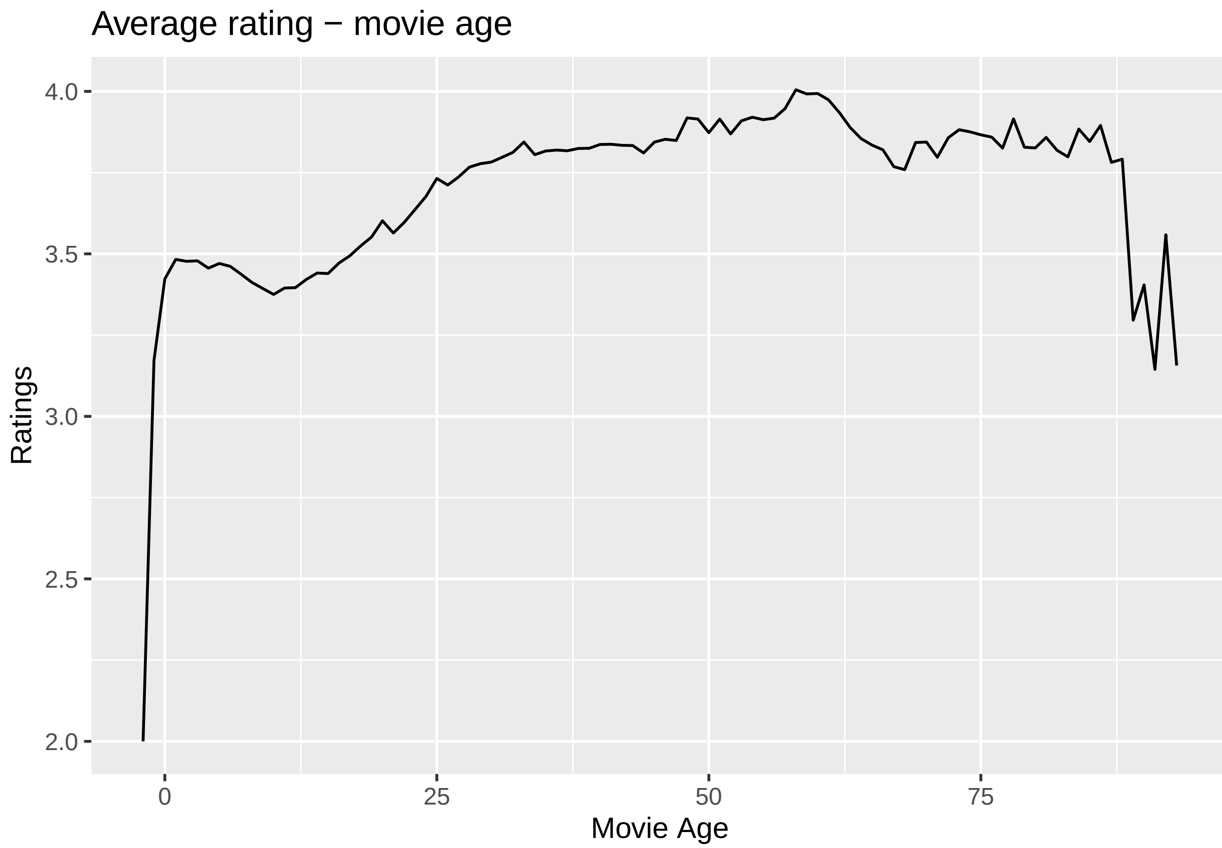
Age - as per *summary()* function we can see that age of movie at the moment of rating various between -2 and 93 years. Ratings with negative movie age or mistake, or rating was provided before movie release.

In total we have 201.00 movies that received feedback before release.

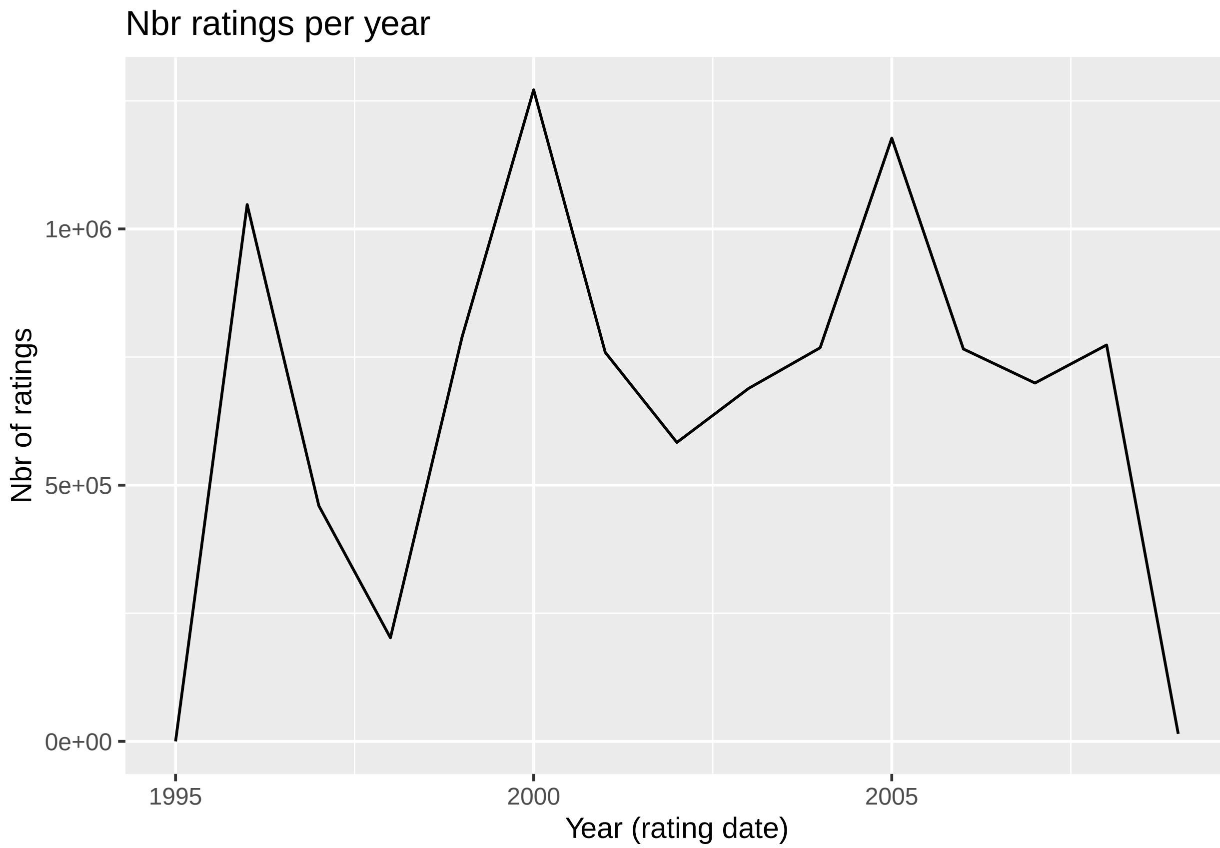
We can see that number of rating is grow up during first year, with maximum at one year after movie release and decrease after one year:



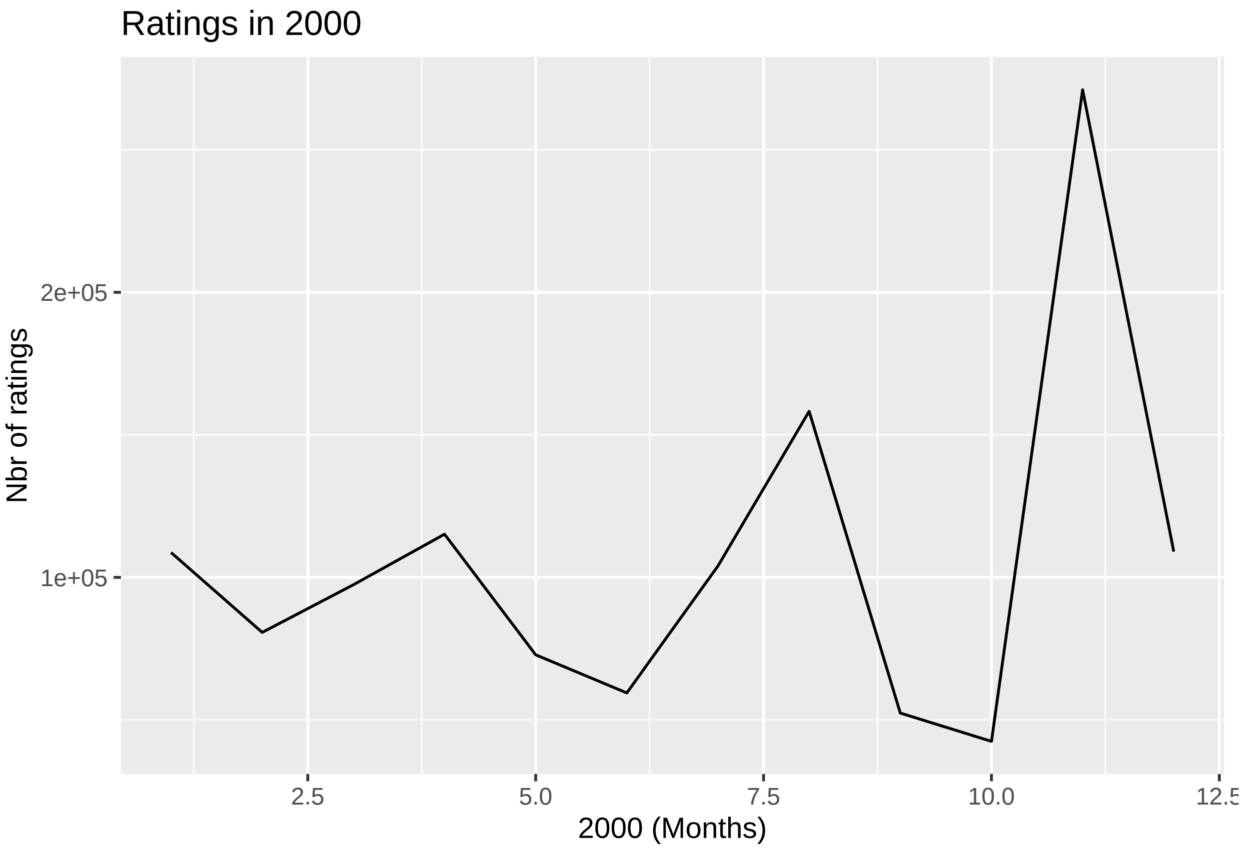
The impact of **AgeM** on ratings:

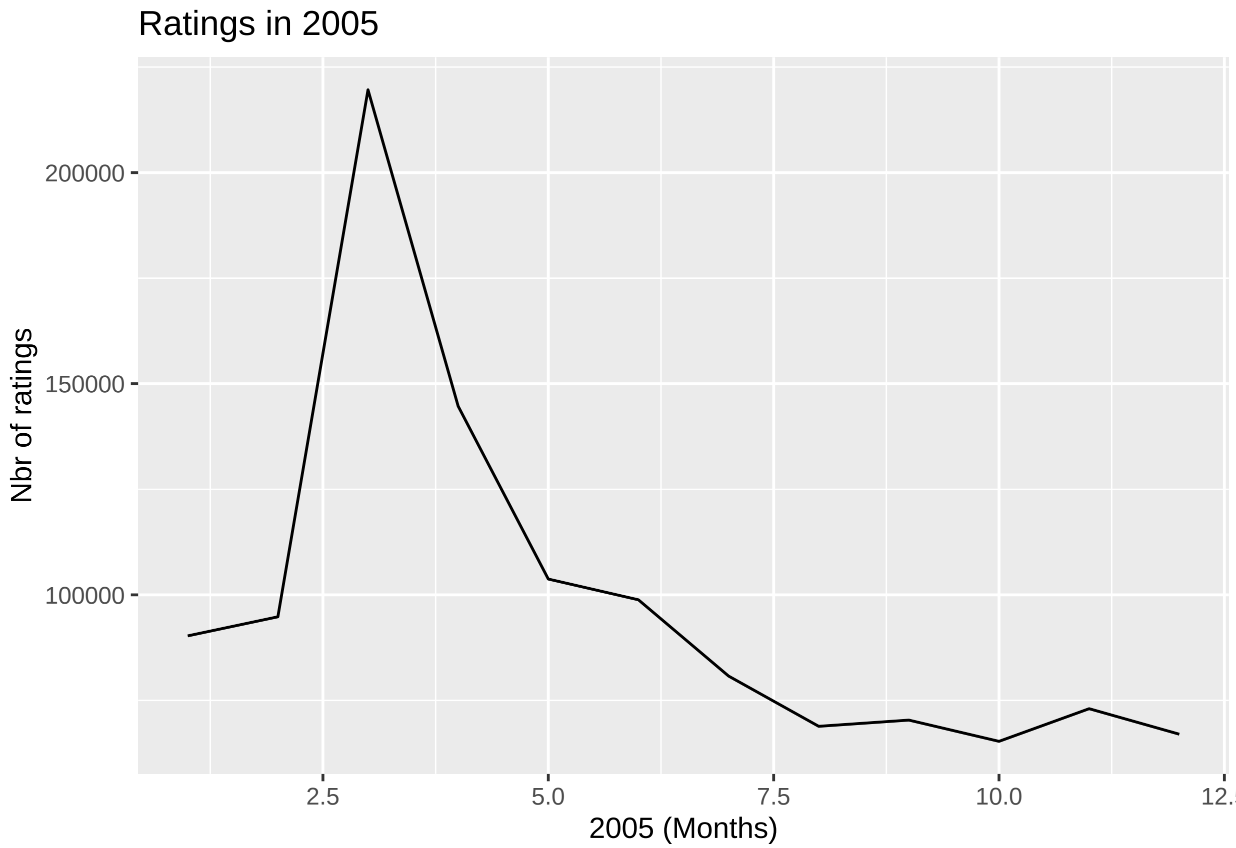


Date - date of rating, from *summary()* function result we can see that first rating was provided in 1995 and latest one in 2009, movielens dataset has 15 years of user observations. We can observe how number of user ratings various from year to year:



Variation during one year does not observe particular pattern, as i.e. if we compare 2000 and 2005 we can see that pick of rating is November for 2000 and March in 2005:

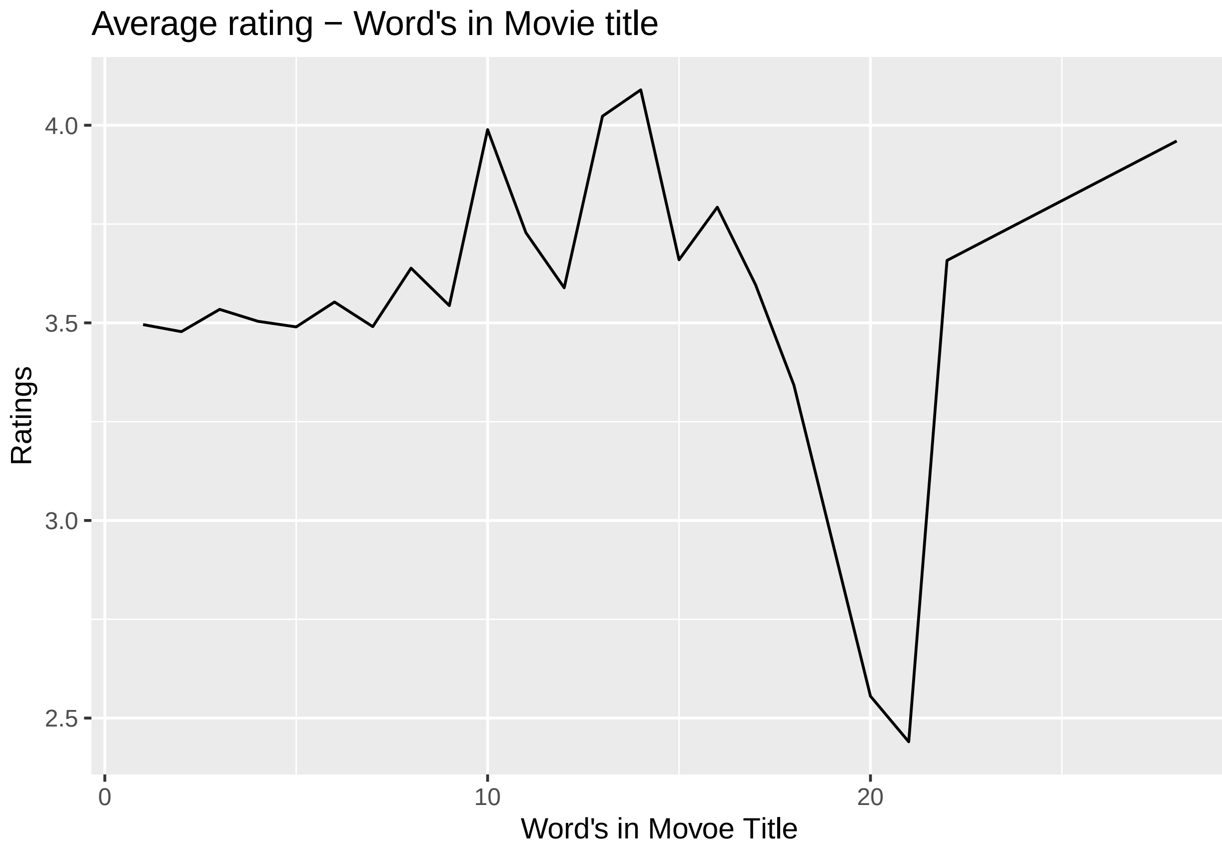

Ratings in 2000
Nbr of ratings
2e+05
1e+05
2.5
5.0
7.5
10.0
12.5
2000 (Months)



Title - as this is character value and individual movie title, *summary()* function does not provide useful info. I've used this column to try to identify sentiment links to movie title, but for majority of movies no sentiment (N/A) was identified.

Word_title - this column was build as number of worlds in movie title. From *summary()* we can see that we have between 1 (min value) and 28 (maximum) words in movie title. In majority movie tend to have short title (with mean of 2 and third quartile of 4 words in title).

Impact on rating of **Word_title** :


Average rating − Word's in Movie title
4.0
3.5
3.0
2.5
Ratings
0
10
20
Word's in Movoe Title

2.2 Model building

Usage of classic methods as i.e., linear regression, the random forest was not possible, as large datasets generate memory issues. The linear regression for one predictor at my laptop took more than 15 min to execute. It was not possible to increase the number of predictors in the model, even with additional memory allocations. I've observed a similar issue with random forest. So, as in Harvard Machine learning courses, I've tried some naive approaches, i.e., using average movie rating as a prediction. At each step, the model was trained at the train set and evaluated at the validation set. A program keep the results of each model evaluation for future comparison.

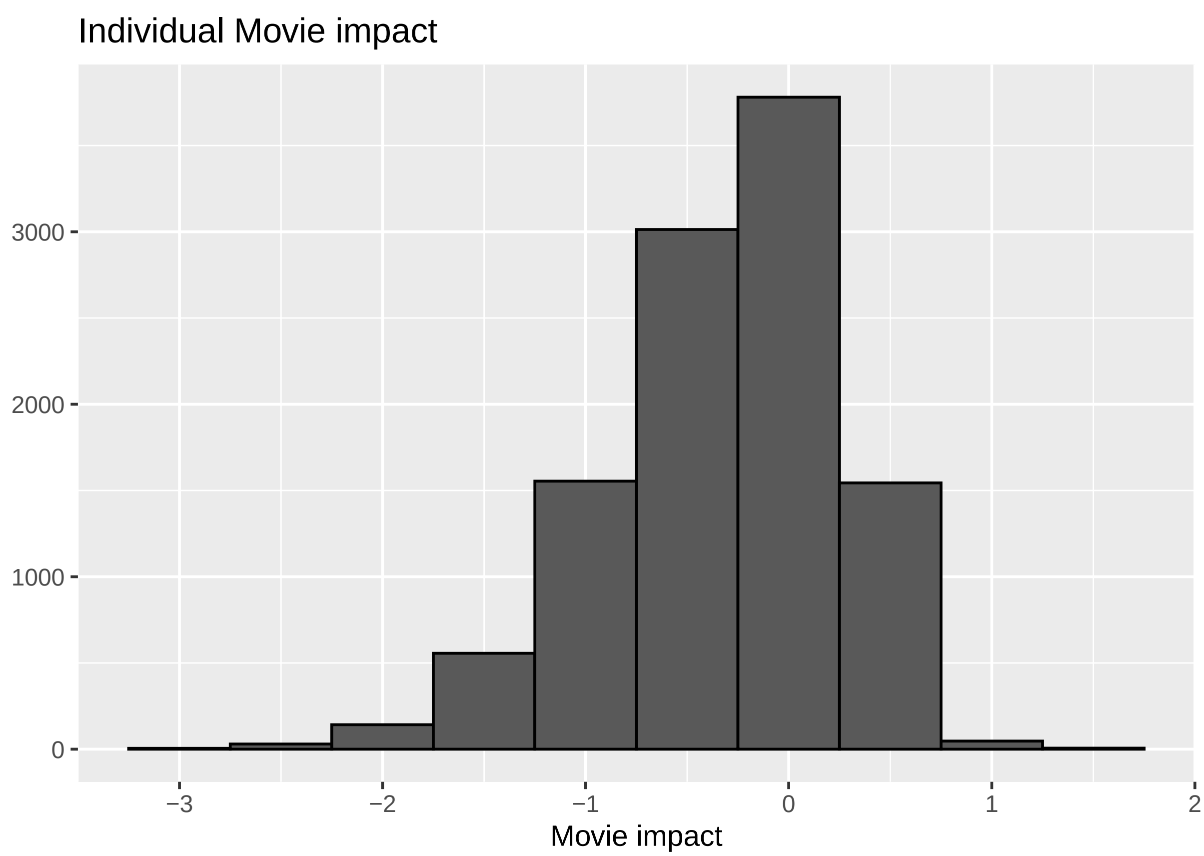
My first very simple model presented as $X_{u,i} = \mu + \epsilon_{u,i}$ where

- $X_{u,i}$ is rating provided by user u for movie i,
- μ is total movies average ratings,
- $\epsilon_{u,i}$ - error for rating for user u, for movie i.

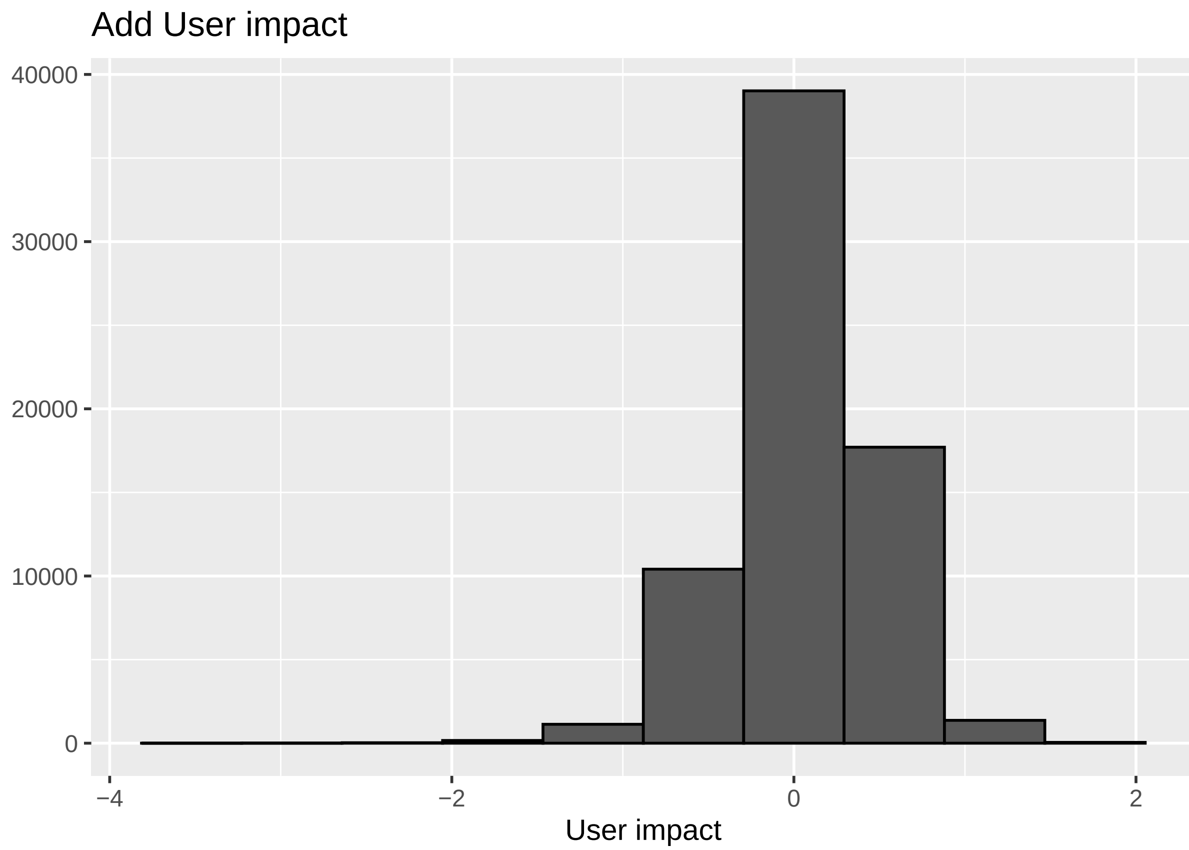
The RMSE at this first step is 1.06120.

In the next two steps, I've reproduced Harvard's course models by including particular movie b_i and users b_u effects. In data analysis step I've saw that user and movie, both has impact on rating results. Same as in the course model, each additional feature brings improvements in model accuracy.

Here is individual movies impact:



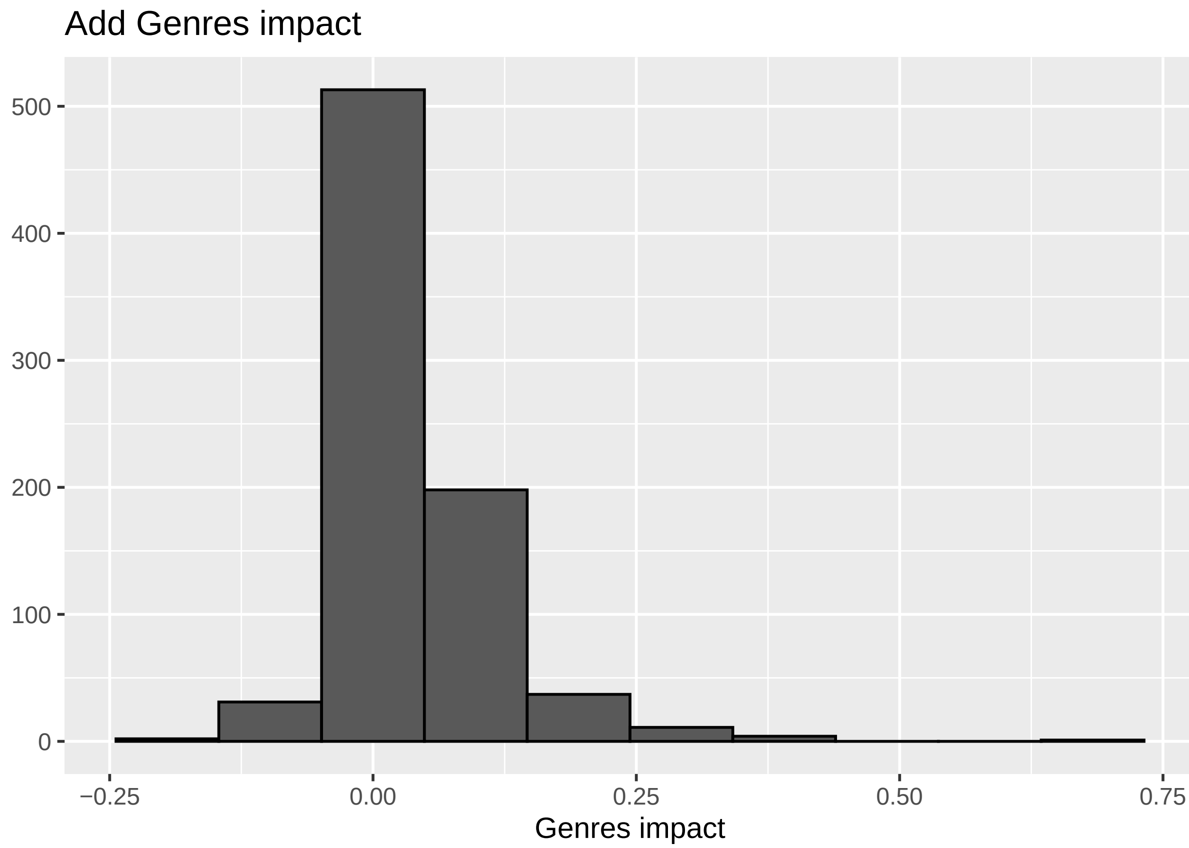
And users impact:



With movie impact RMSE is 0.94391 and with additional user impact RMSE is 0.86535.

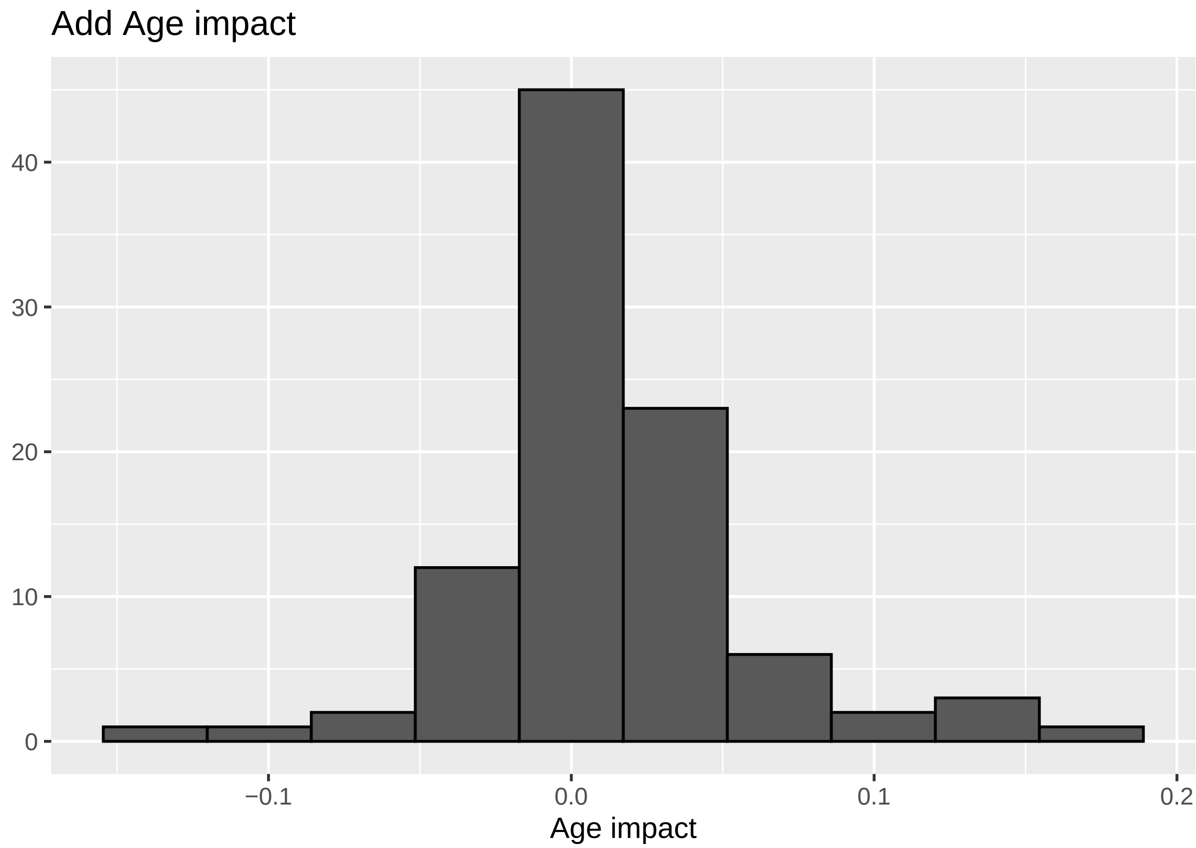
To drive more in deep, I’ve included additional effects such as genres and movie age effects. Those effects also improved the model.

Here is the genres impact:



RMSE after this enhancement is 0.86495.

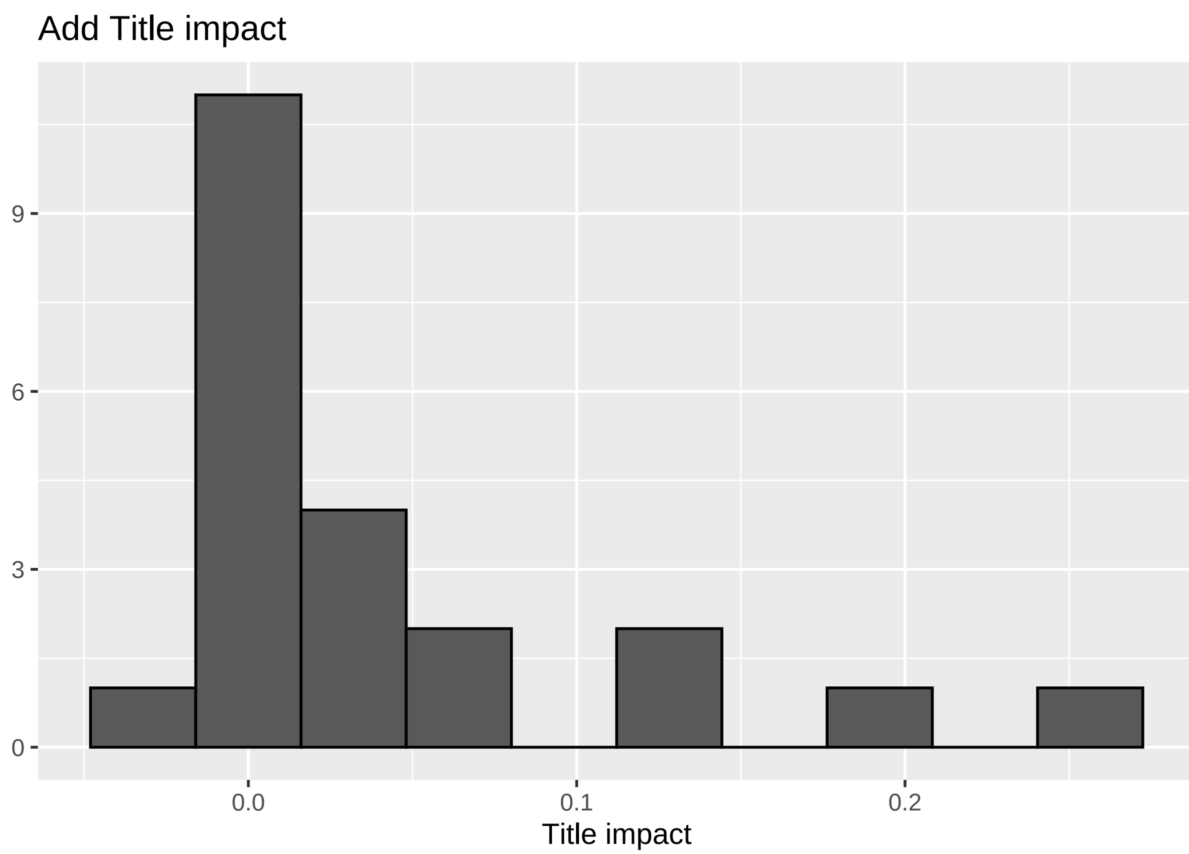
Impact of **AgeM**:



RMSE after age step is 0.86454.

I've tried to perform sentiment analysis on the movie title. Still, I did not identify sentiments for many movies, and the results were not significant to include in the model. Finally, I also explore if Title (i.e., if a short or long title influences rating).

Here is impact of **Word_title** (Nbr of word in title) impact:



RMSE after final step is 0.86453.

My final model is $X_{u,i} = \mu + b_i + b_u + b_{i,g} + b_{i,a} + b_{i,c} + \epsilon_{u,i}$ where

- $X_{u,i}$ is rating provided by user u for movie i,
- μ is total movies average ratings,
- b_i - movie i effect,
- b_u - user u effect,
- $b_{i,g}$ - movie i, genres effect,
- $b_{i,a}$ - movie i, age effect,
- $b_{i,c}$ - movie i, title length effect,
- $\epsilon_{u,i}$ - error for rating for user u, for movie i.

2.3 Results

RMSE results for each model:

method	RMSE
Just the average	1.0612018
Movie Effect Model	0.9439087
Movie User Effect Model	0.8653488
Movie User Genres Effect Model	0.8649469

method	RMSE
Movie User Genres Age Effect Model	0.8645400
Movie User Genres Age Title Effect Model	0.8645256

As we can see final model obtain desired precision. Last model “Movie User Genres Age Title Model” is final model.

4 Conclusion

In this last section, I would like to review the limitations of the current study and mention possible future opportunities.

One of limitation of this method is that we need to have some ratings already provided by user, it will works less well for new users in dataset.

I've used only explicit feedback, which in our case was the user's rating. Data explicitly provided by user limits research. As most users do not rank all movies that they watch, it may be more interesting to evaluate implicit feedback (i.e., instead of rating, collect what movies users watch completely or how long users interact with the film.)

Another point is that some other information will be useful as, i.e., actors or director as if you like the actor you may also be interested in seeing other movies with this actor.

References

- PH125.8x: Data Science: Machine Learning of Professor R.Irizarry (Introduction to Data Science book on https://rafalab.github.io/dsbook/)
- http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/