Movie Recommendation Report

Faneva Tafita REZIKY STEFANA

2022-08-28

Introduction

We are living in an era of data and AI. From the most pressing issues in our daily life to the smallest, atomic habits, most of it are related, or at least getting more related to the use of data. Data is now a big help in improving products both directly (for instance tech industries rely mostly on the use of data to create more accurate contents), and indirectly, as data can be used to improve marketing strategies and make products more appealing. Here, we have a case related both to direct and indirect use of data on a product. Streaming industries like YouTube and Netflix are using algorithm in order to satisfy users by recommending movies that match to the users' preferences. Preferences can be shown in many ways: by looking at the most seen types of movies, the use of like/dislike and/or share button,… Let's get a look at our data:

```
## Classes 'data.table' and 'data.frame':   10000054 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 231 292 316 329 355 356 362 364 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983392 838983421 838983392 838983392 838984474 838983653 8
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Dumb & Dumber (1994)" "Outbreak (1995)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Comedy" "Action|Drama|Sci-Fi|Thriller"
##  - attr(*, ".internal.selfref")=<externalptr>
```

Here, we have a data frame with 10 millions observations and 6 variables: *userId* a chain of number that helps identify the user; *movieId*: an identifier to help identify the movie; *rating* (the outcome variable), which is the rating the user attributed to the movie based on how satisfied they were and ranging from 0 to 5; *timestamp*: an integer expressing the date of rating, using second as unit; *title*, a character vector of the movie title; genres: the genres in which the each movie belongs to.

For the purpose of our data analysis, we decided to split the data into 2 data sets: *edx* data, the data on which we will perform our analysis and train our algorithm, and *validation* data on which we'll test our data. They both correspond respectively to 90% and 10% of the initial data frame *movielens.* Here are the dimensions of the train and test set.

```
## [1] 9000055       6
```

```
## [1] 999999      6
```

Analysis

Before we start our analysis, we need to take a look at both our train and test sets.

Here is our train set:

```
##    userId movieId rating timestamp                         title
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
## 3:      1     292      5 838983421               Outbreak (1995)
## 4:      1     316      5 838983392               Stargate (1994)
## 5:      1     329      5 838983392 Star Trek: Generations (1994)
## 6:      1     355      5 838984474       Flintstones, The (1994)
##                            genres
## 1:                Comedy|Romance
## 2:         Action|Crime|Thriller
## 3:  Action|Drama|Sci-Fi|Thriller
## 4:       Action|Adventure|Sci-Fi
## 5: Action|Adventure|Drama|Sci-Fi
## 6:       Children|Comedy|Fantasy
```

And here's our test set:

```
##    userId movieId rating timestamp
## 1:      1     231      5 838983392
## 2:      1     480      5 838983653
## 3:      1     586      5 838984068
## 4:      2     151      3 868246450
## 5:      2     858      2 868245645
## 6:      2    1544      3 868245920
##                                                       title
## 1:                                     Dumb & Dumber (1994)
## 2:                                     Jurassic Park (1993)
## 3:                                        Home Alone (1990)
## 4:                                           Rob Roy (1995)
## 5:                                    Godfather, The (1972)
## 6: Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##                                      genres
## 1:                                   Comedy
## 2:         Action|Adventure|Sci-Fi|Thriller
## 3:                          Children|Comedy
## 4:                 Action|Drama|Romance|War
## 5:                              Crime|Drama
## 6: Action|Adventure|Horror|Sci-Fi|Thriller
```

We can see that there are further variables that we can extract from the data

Variable extraction

There are 2 variables that can be extracted from the initial train and test sets: the date of rating and the release date of the movie.

Rating date

The date of rating is shown within the *timestamp* variable. The *timestamp* variable is an expression of the rating date in second. There are 2 points which makes it difficult to exploit this variable: the first one is its precision, as such atomic-leveled precision creates too division in the data and therefore makes each observation different: this can be a source of bias. Another point is that this precise tool do not affect the user's preference. Therefore, we need a larger time period. We will therefore choose the year as a time reference. We will name it *rate_year*.

Release date

When we looked at the first lines of *edx* and *validation* data sets, we saw certain pattern within the *title* variable:

```
## [1] "Boomerang (1992)"               "Net, The (1995)"
## [3] "Outbreak (1995)"                "Stargate (1994)"
## [5] "Star Trek: Generations (1994)" "Flintstones, The (1994)"
```

We can see that the *title* variable is a character string and at its ending part, between brackets, there are 4 digits which indicate the release date of the movie as part of the movie title. We will extract this variable and name it *release* variable.

Analysis: statistics and visualization

We will perform our analysis using statistics and visualization approaches.

Basic statistics

The first thing we'll look at the outcome variable statistics. To do so, we'll look at the basic statistical summary. We will summarize the data and save under *avg* column the average rating in the full training data set and under *se* column the standard deviation. Here is the statistical summary of the rating distribution:

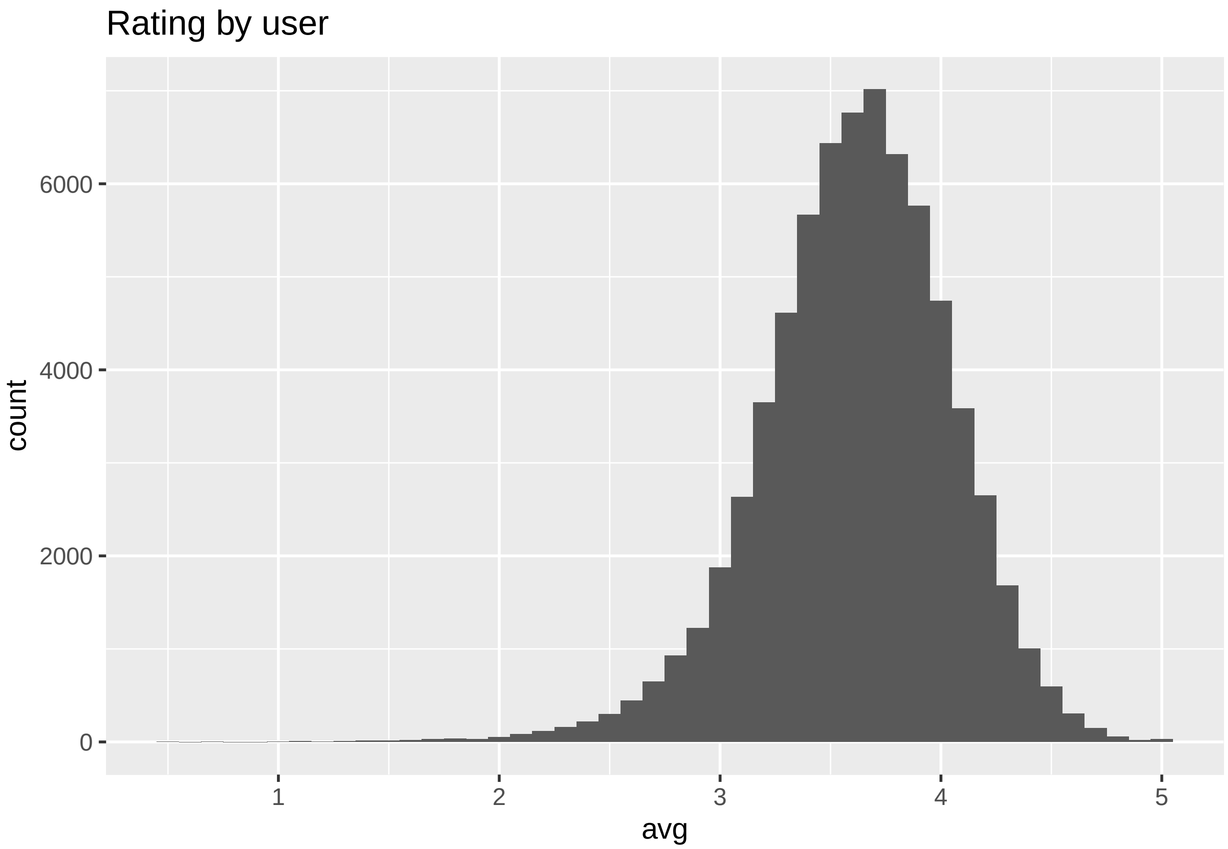
```
##        avg       se
## 1 3.512465 1.060331
```

We can see that the average rating by every user is 3.51 and the average distance to the mean is 1.06.

Starting from this point, we'll see the distribution of the rating given each variable. This follows a Bayesian notation and that's what we'll look in the following paragraphs.

User variable

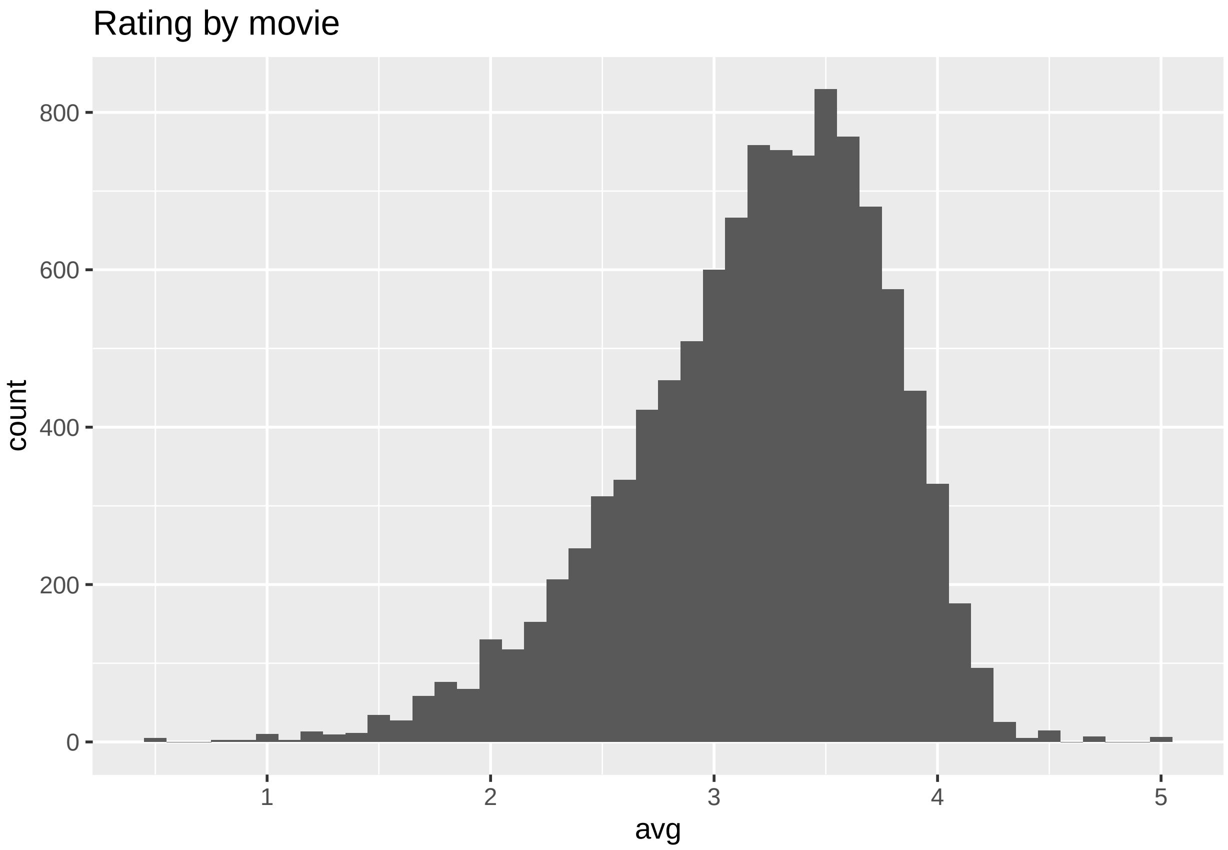
The user is the main source of variation that we'll need to take, as the analysis is mainly made for the user and their preferences is based on their distribution. Here, we will plot the distribution of rating given the user. To do this, we will find the conditional average, **E(Rating|User)**, which is the average rating by each user. Here is the distribution the rating given by every user plotted.



The plot takes the shape of a normal distribution, and we can clearly see that the rating per user are mostly concentrated a bit more than the average rating we found earlier, however, the difference is still quite close (less than .2 standard deviation away from the average).

Movie variable

Apart from the user rating, each movie can also bring variation. The fact is that users can have difference general preference for each movies. We plot here the distribution of the average rating per movie.



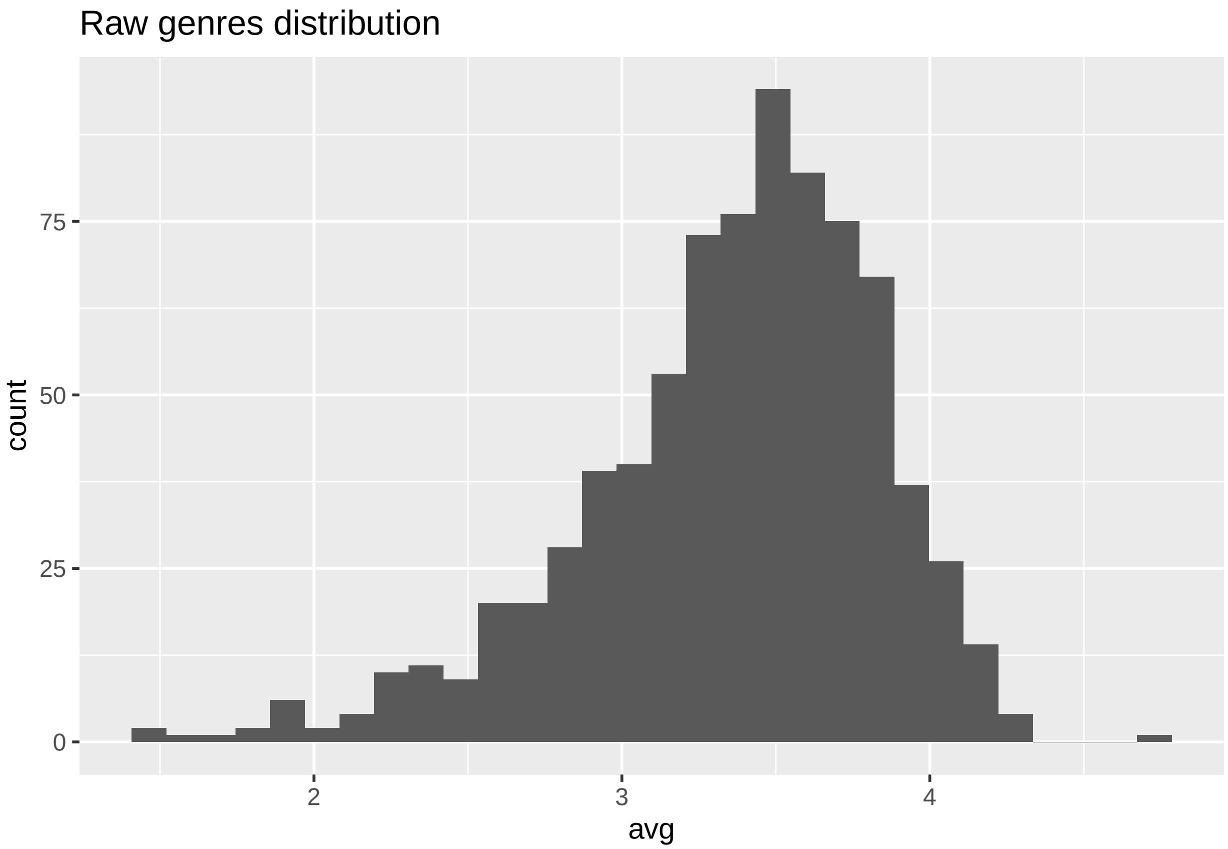
Genre variable

In addition to the movie rating, genres can also reflect the user's preference. Even better, genres may bring more effect to the preference, as it quite determines the taste of each user, as the emotion the user expects from the movie depends on the genre.

```
## [1] "Comedy|Romance"                 "Action|Crime|Thriller"
## [3] "Action|Drama|Sci-Fi|Thriller"   "Action|Adventure|Sci-Fi"
## [5] "Action|Adventure|Drama|Sci-Fi" "Children|Comedy|Fantasy"
```

```
## [1] 797
```

We can see each movie may have different genres (eg: "Comedy|Romance", "Action|Crime|Thriller"…). In this scheme, there are 797 different genres. Here is the distribution among the 797 genres in the initial data set:



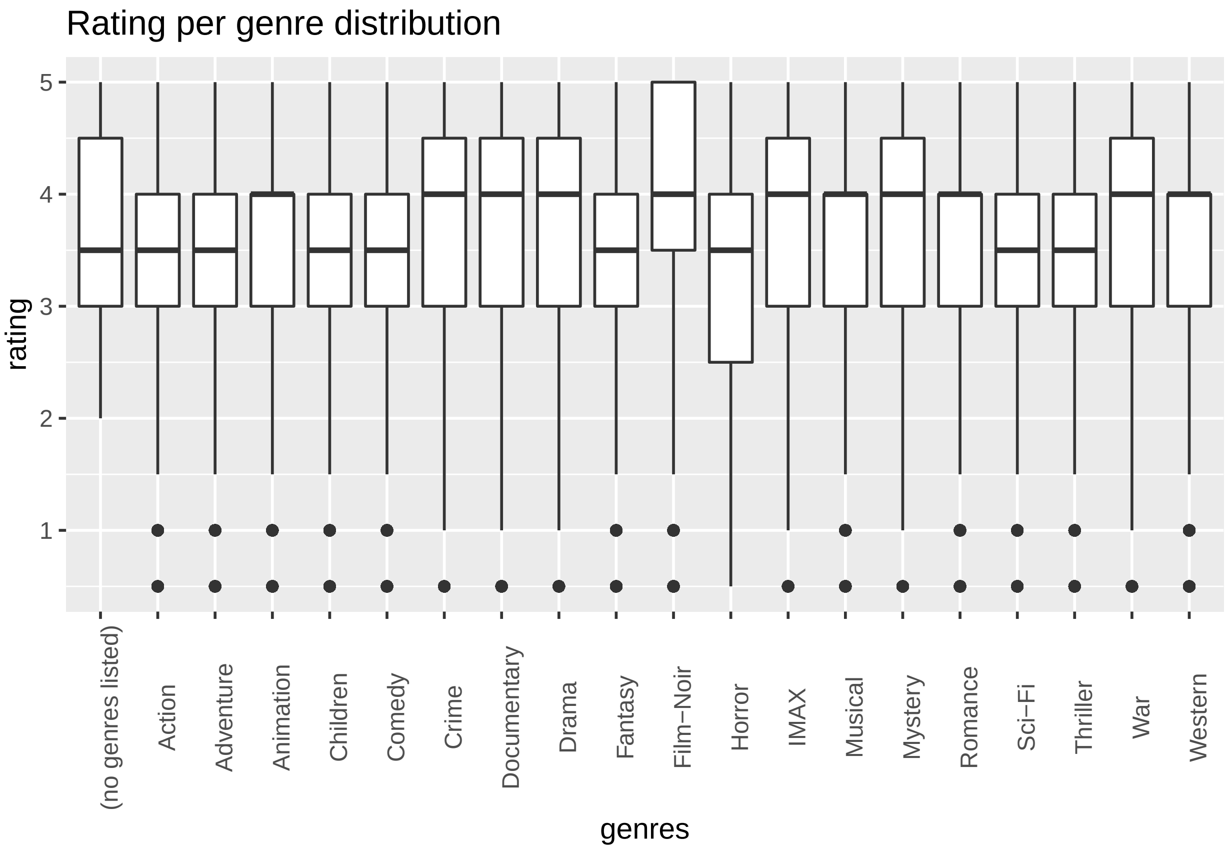
We can see here that there are clear variation related to the genres. The values rotate around the average, but the dispersion is still quite low. One thing with this variable is that it can be quite biased, as there are many genres that do not have that many rating.

```
## # A tibble: 1 x 1
##   `Proportion of genre with less than 100 ratings`
##                                              <dbl>
## 1                                            0.210
```

We can see that 20% of genres have less than 100 ratings, but we can avoid that by breaking *genres* into the actual unique genres. Now, let’s find out how many genres are there actually.

```
## [1] 20
```

There are actually 20 genres, and the combinations genres compose the 797 different genres in the initial data set. Here is the variation among the 20 homogeneous genres.



We can see that there are distinct pattern of distribution: some has bigger interquartile range (the distance between the higher and lower horizontal bar), some medians are higher on the plot (the horizontal bold bar),… But still, there are not many dispersion.

Movie and genre variables

Movie and genre variables are 2 variables that are closely related, as each movie belongs to a specific genre and a genre can have many movies. This means that it will be possible that we will not need both variables for our machine learning process, as they both may bring the same variability. We will need to choose. As we saw from the graphs above, there is not enough variability among genre, and therefore, it may be better to opt for movie variable.

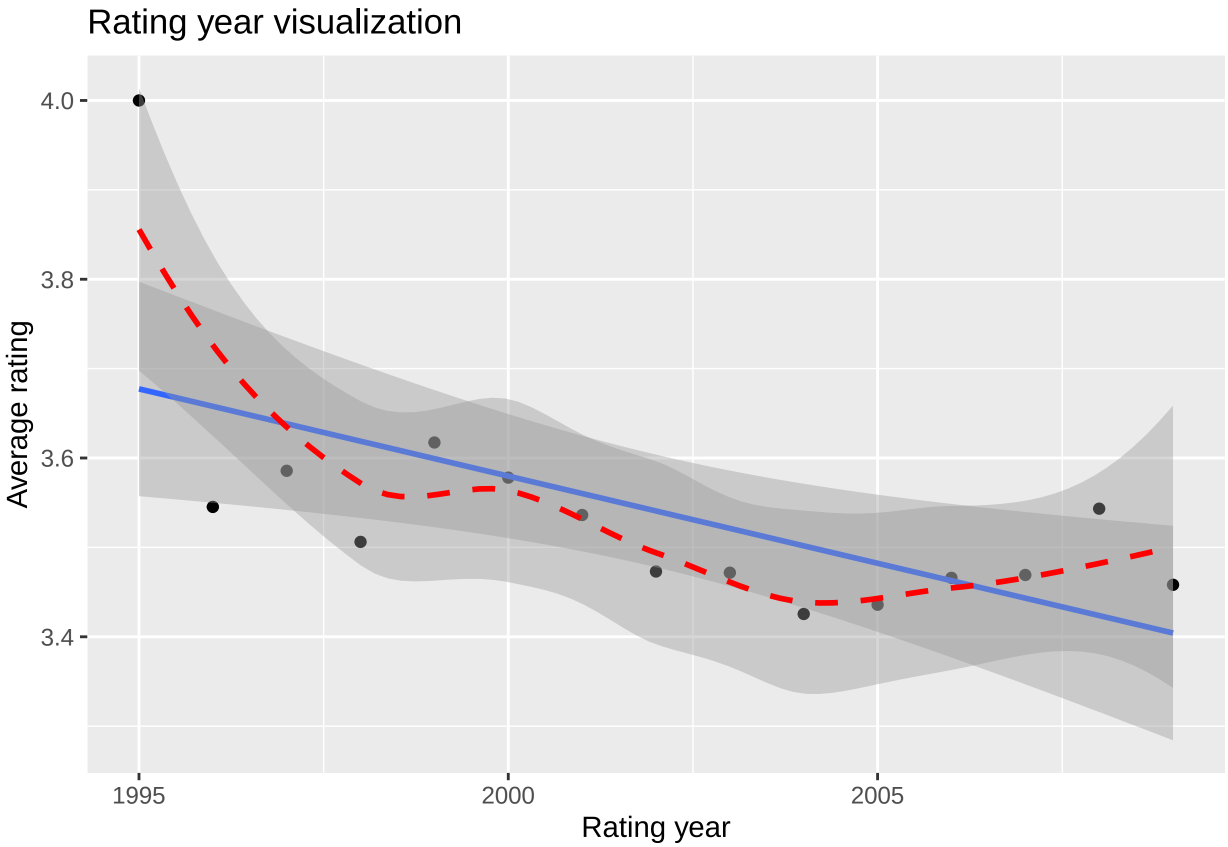
Rating year

The year of rating can be an important variable. Preferences may vary across years and so does the rating attributed. We will visualize the average rating per year. Do appreciate this variable, we will use 3 methods:

- A ***scatter plot***, which is the plot of the average rating per year at their actual level. This permits us to see any patterns on the data.
- A ***linearly smoothed plot***, which plots the linear regression of the average rating per year. This permits us to appreciate the tendency and to appreciate the effects.
- A ***smoothed plot***, which plots a smoothed tendency of the data. This confirms the patterns shown by the scatter plot.

The combination of the linear regression and the scatter plot shows us the difference in balance on the data: some points may be farther than the actual while many points may be concentrated on an unique point. The linear and smooth plot on the other hand permits us to appreciate if there are any fluctuation compared to the general trends.

Here is the plot for the rating year.



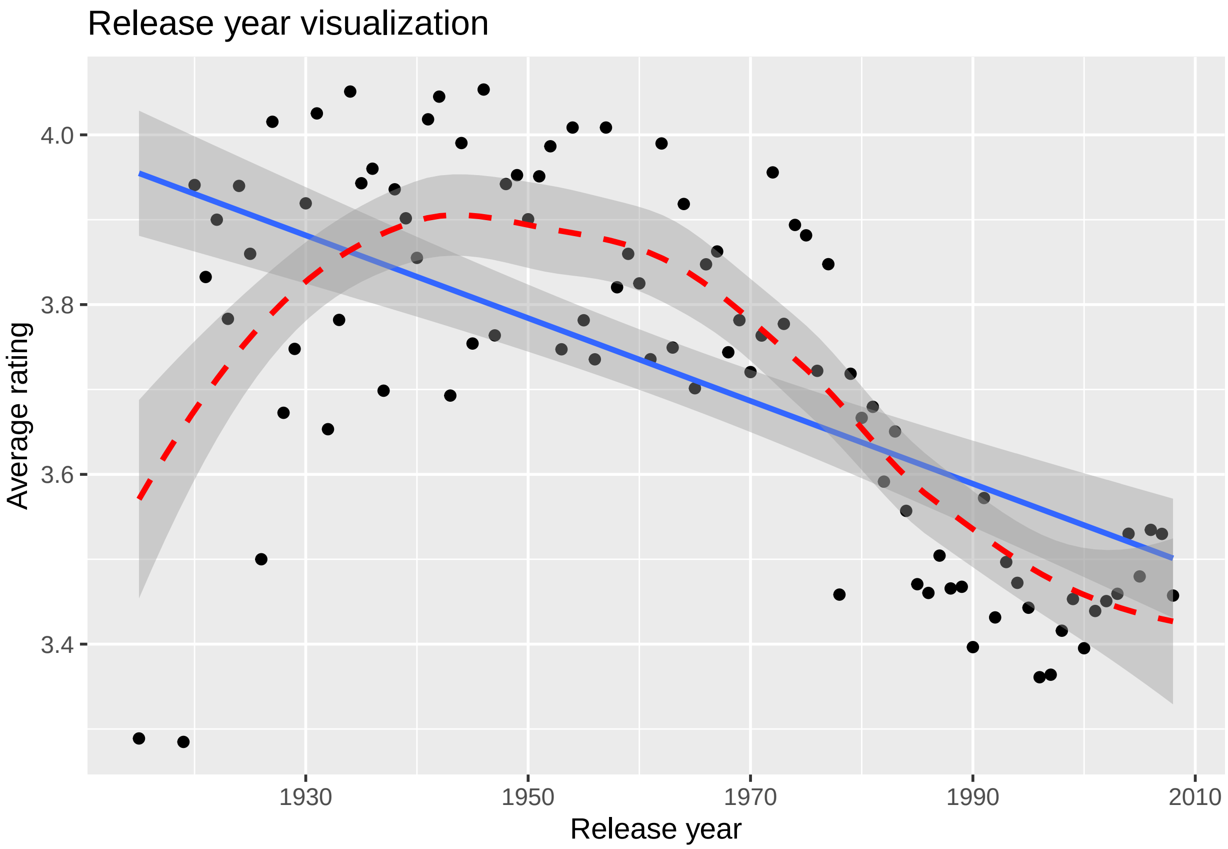
Here, we can see that the smooth shows 3 infliction points. This information is very useful for the machine learning part.

Release year

The release year can be an important factor in the variation of preferences.

1. Some years may have created more blockbuster,
2. The preferences may be related to some attachment, for example, in some cases, it can relate to some people's childhood

We will use the same approach as we did using the rating year.



We can identify 2 inflection points.

After this exploratory analysis, we will move next to the machine learning part using the teachings from this analysis part.

Machine learning

We will divide this part into 3 section: identification of the baseline scenario, regression method using discrete variable and regression method using continuous variable. These parts have been made to create a model that we'll improve, create prediction using discrete variables, and regress using numeric variables.

Baseline scenario

In this part, we will create a model which will be the one we'll need to improve. This model is very simple: we will create an algorithm which guess the average rating every single time. Using this algorithm, we will get the following result.

```
## [1] 1.061202
```

As we train our model, we will try to improve this result.

Note: We will try to improve this model, not the following models, as improvement may sometimes cause bias, called over-training.

Regression using discrete variable

For this part of regression method, as we have a very large data frame and many unique points, we will use regularization. We will use the following formula:

$Rating_i = Average + UserEffects_i + MovieEffects_i + GenreEffect_i$

Each of these effects are called bias, and corresponds to the average of rating given each variables. We will find the bias for each variable.

User bias

As we saw earlier, the user bias is the average effect of the user on the rating. To find the user bias, we will need to subtract the average, also called ***general bias***, from the actual rating.

Here is the result of the new model.

```
## [1] 0.978336
```

Movie bias

We will find the movie bias using the same approach as the user bias, but this time, we will remove from the actual rating the general bias and the user bias. This was performed for us to avoid any sort of confounding, as we may still account for the user bias someway alongside user bias. Here, we have the result accounting for the movie bias.

```
## [1] 0.8816096
```

Genre variable

We mentioned earlier that we will need to choose between *movie* and *genres* variables, but here, we can prove this fact by looking at the remaining effect that genres variable is able to predict. We will show here the variation that genres can bring, by showing the range (minimum and maximum values) of the average genre effect.

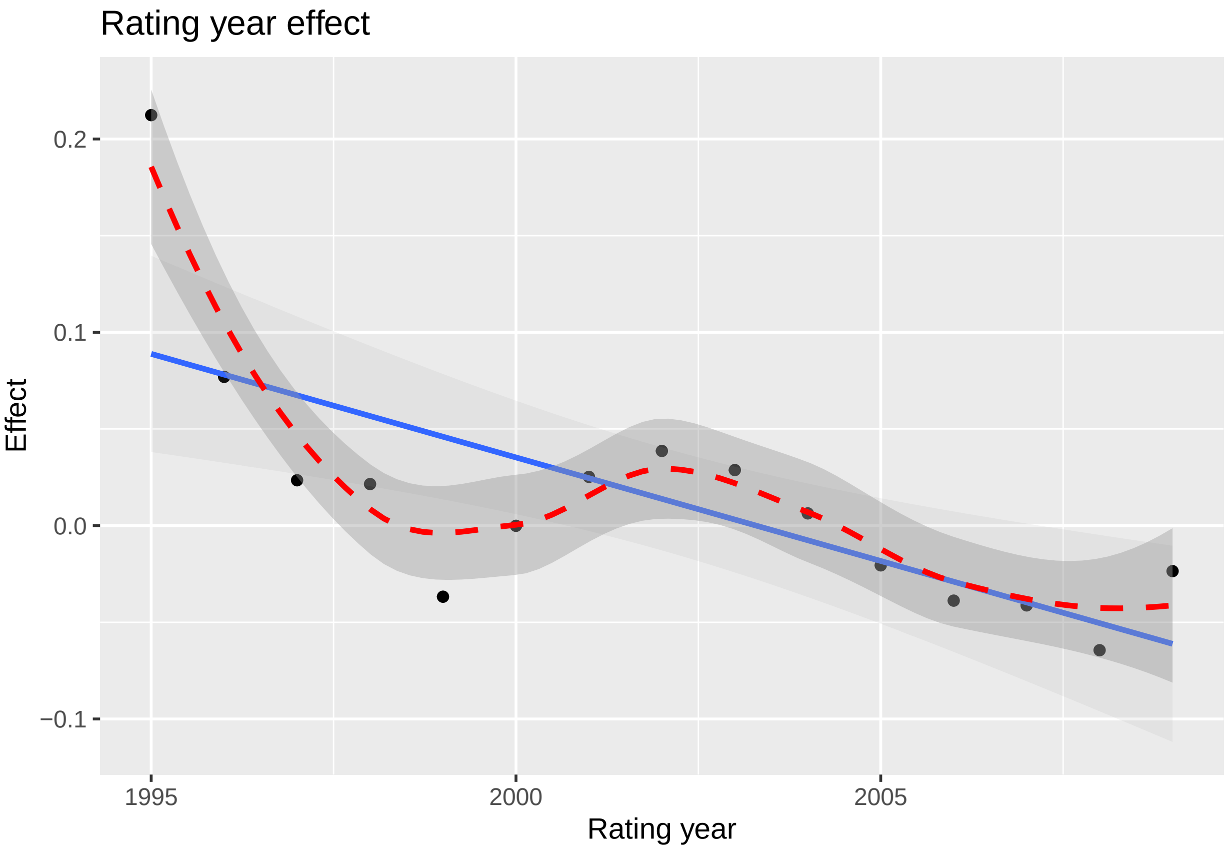
```
## [1] -1.522955e-16  1.776357e-16
```

We can see that the values are very small (close to 0) and that it will only create a small approach to the actual prediction.

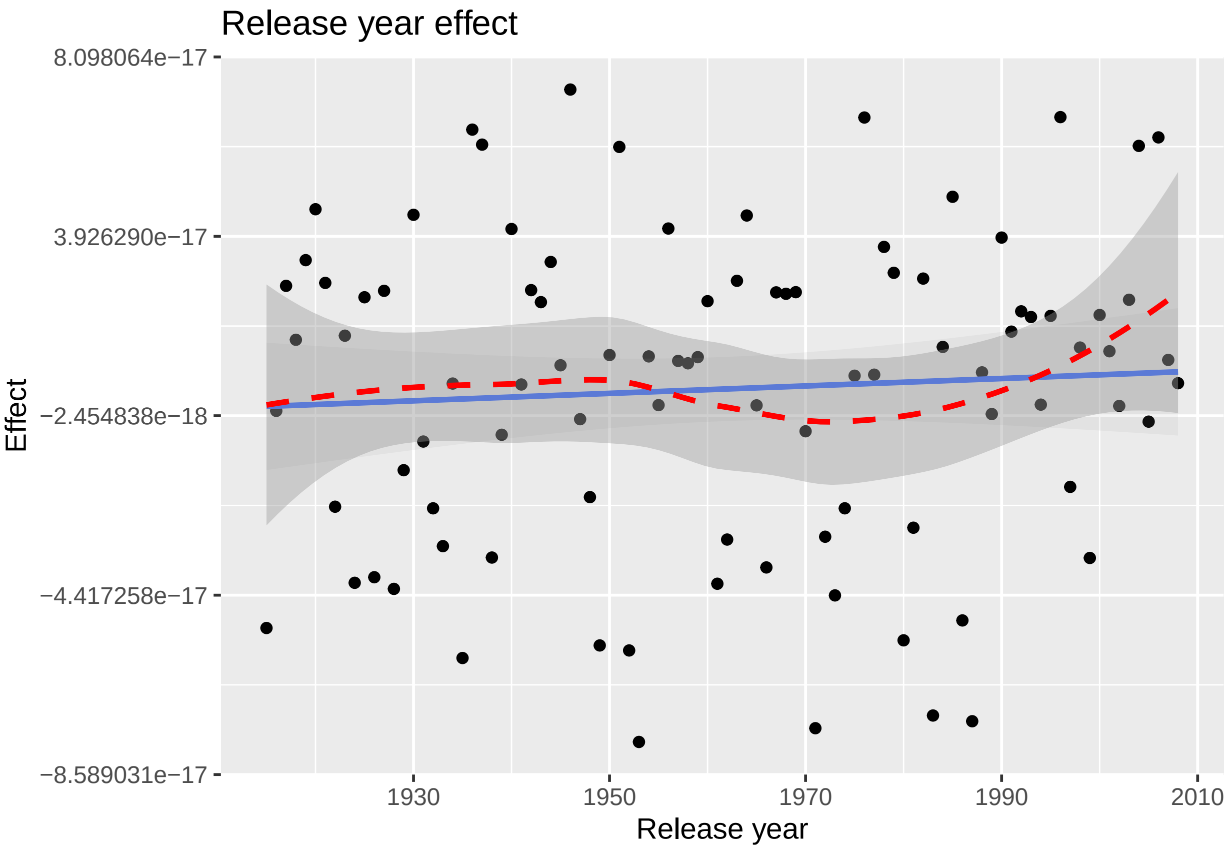
Regression using continuous variables

In this part of regression is the technique needed to predict an outcome using continuous variables. We still have 2 variables for the regression: rating year and release year. Before we start, we need to plot the remaining effect on both variables, as some changes may have appeared after removing values from the previous regression.

Here, we plot the remaining effect explained by the rating year:



And here is the effect of the release on rating:



Here, we can see that there remains only one single variable for the regression, as the fluctuation is very low and quite flat. We will need to perform our regression using the rating year.

For this part, we are going to consider a few method and pick the ones with the best result. Here, we will directly combine the results from the previous and the current regression method and see the results.

Linear regression

A linear regression will be used as our baseline. We will choose algorithm that are able to perform better than the linear regression.

```
## [1] 0.8811339
```

The rating is quite close to the ones using the discrete variable alone. We will pick the methods that can improve this model.

Polynomial regression using lm

As we saw earlier, the rating year had 3 inflection points. This can mean for us that the derivative equals to 0 thrice, therefore, if we want to fit using a polynomial approach, we will need to fit using a fourth degree polynomial.

```
## [1] 0.8808315
```

We have a better result. Let's find other ways to improve our model

Loess

Here, we will try another approach to fit our data, as there may be different way to predict our data. The first model we can try is loess regression: a technique where we use moving average.

```
## [1] 0.8808291
```

We can see using loess that the results are close to the ones with our polynomial approach. This can also mean that our fitting approach using the polynomial was a good one, as it is close to the smoothing method.

Neural network

The next method that we'll consider is neural network. This is a very famous algorithm as it aims at recognizing the underlying relationships. Let see how it performs. We will use bayesian regularized neural network from the *brnn* package.

```
## [1] 0.8807942
```

Neural network improved the prediction closer to the actual outcomes.

Final model

For us to create a stronger algorithm, we will find the average result found using the previous techniques. This way, we can avoid any sort of bias due to training.

```
## [1] 0.8808338
```

Conclusion

To sum up, machine learning and data analysis goes hand in hand: data analysis is an important method required before we make any sort of prediction using machine learning. Before we make any prediction, we need to visualize the data and analyze the statistical distribution among variables, find the best way to approach each variables and also choose among variables.

Most of the time, when we hear the expression machine learning, we tend to think about something complicated, like complex code and techniques, however, as we saw using the discrete variables, it can be a simple approach that can explain most of the variation: the approach using regularization in the first part of the regression made most of the result:

```
## [1] 0.8816096
```

We used the continuous variable as a way to fine tune our prediction and reach our final result:

```
## [1] 0.8808338
```

Another teaching that we found is that we do not necessarily need all of the variables, as we used only the user, movie and rating year and we set aside genre and release year. From our side, we need to pick carefully the variables as was the case with movie and genres variable, as both may explain the same thing but the dispersion among variables makes the one better than the other.

A limitation we may find to this analysis is that rating may not be a better prediction to the user preferences when it comes to movie recommendation, however, browsing history and like/share button may be useful way, as rating may depend on some other factors apart from preferences.