Movielens Recommendation System

Cecilia Barradas

9/29/2021

Introduction

Overview: This project is the capstone for the course PH125.9x of Harvard X, in order to earn a Data Science Certificate.

Goal: The challenge is to improve on the Recommendation System used by Netflix, which means developing a machine-learning model that achieves a Root Mean Square Error (RMSE) of less than 0.86490. It is understandable that Netflix would want to improve on their recommendation system as it is well known that people sometimes spend more time searching what to watch than actually watching.

Dataset: In order to embrace the challenge, we were provided with a dataset of 10 million ratings, which is a small subset of a much larger dataset.

https://grouplens.org/datasets/movielens/10m/

http://files.grouplens.org/datasets/movielens/ml-10m.zip

Key Steps:

1. Download and load the dataset
2. Create the edx and validation sets. The edx set will serve to train and test the models. The validation set will be used at the end to test the final model.
3. Explore the data: both the edx and validation sets and the variables included in the datasets to see how the data is distributed and the effects in can have in the model. Some tables and graphics will be created to visualize these effects.
4. Data modeling: dividing the edx dataset into a train and test set and creation of a table to record the results of every model.
5. Develop the algorithm starting with a naive model, to then add the variables and their regularization.
6. Test the last model with the validation set.

Method

The following libraries were used: tidyverse, caret, data.table, ggplot2, lubridate, dplyr, knitr, RColorBrewer, rmarkdown, dslabs, pdftools, kableExtra

1. Data preparation

Downloaded, prepared the data and created the edx set and validation set (final hold-out test set), with provided code.

2. Data Exploration, visualization and Insights

Analysis of the basics of the data: size, variables, missing data, main information. There are 9,000,055 observations and 6 variables in the edx dataset. The six variables are userId, movieId, rating, timestamp, title and genres and their class were integers, numeric and character.

```
##      userId         movieId          rating        timestamp        
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08  
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08  
##  Median :35738   Median : 1834   Median :4.000   Median :1.035e+09  
##  Mean   :35870   Mean   : 4122   Mean   :3.512   Mean   :1.033e+09  
##  3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000   3rd Qu.:1.127e+09  
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09  
##     title              genres         
##  Length:9000055     Length:9000055    
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
## 
```

```
## Classes 'data.table' and 'data.frame':   9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838984885 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Ad
##  - attr(*, ".internal.selfref")=<externalptr>
```

Table 1: A glimpse on the edx dataset

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

As for the validation set, there were 999,999 observations.

After this first data overview, we dive deeper into each of the variables, in order to understand the distribution of the ratings.

movieID

For the movieId variable, there were 10,6777 unique movies. 126 movies had only one rating, while 143 had more than 10,000 ratings. The highest number of ratings was concentrated around the 100s.

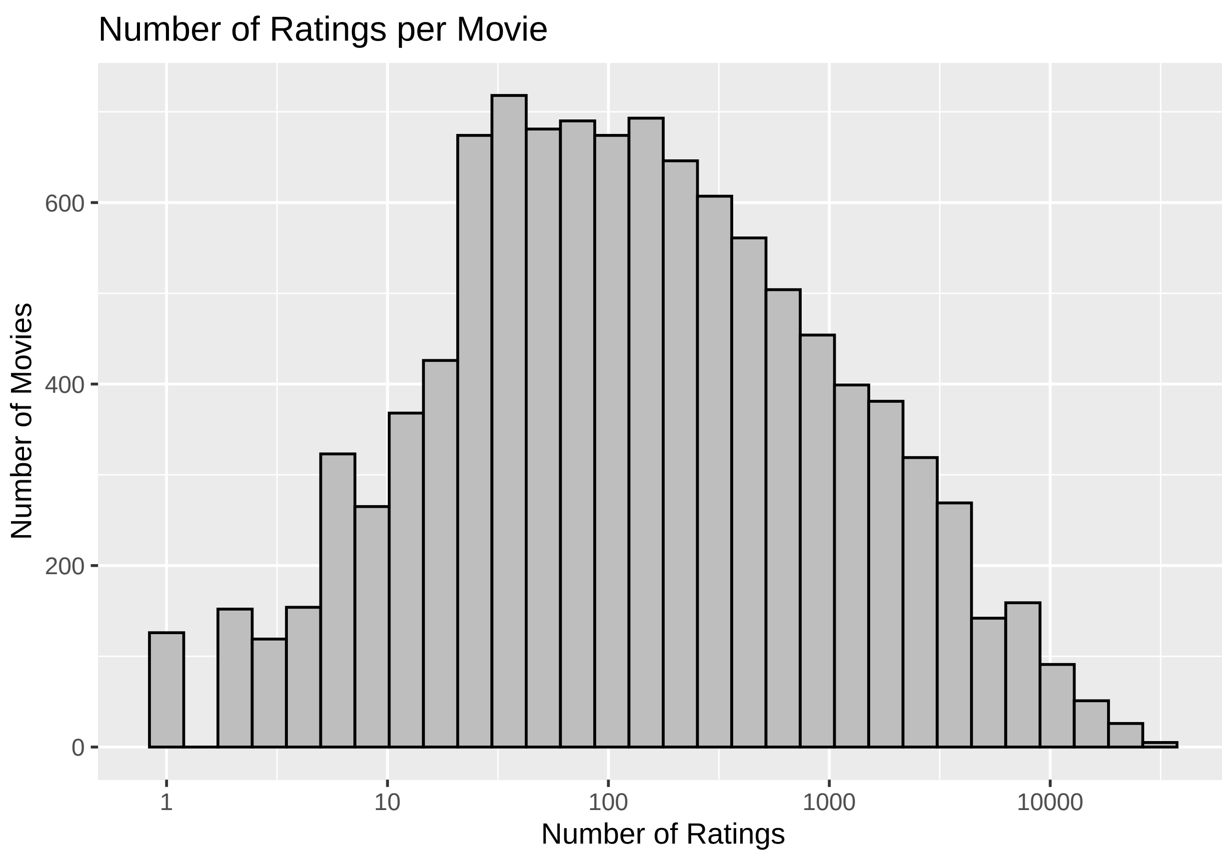


Table 2: Most Rated Movies

movieId	n
296	31362
318	28015
356	31079
480	29360
593	30382

There are three movies with more than 30,000 ratings: Pulp Fiction, The Shawshank Redemption and The Silence of the Lambs. Very popular movies. Among the movies with only one rating are The Quarry, Hexed and Impulse, quite unknown movies. Clearly there is a bias of more ratings on more popular movies, therefore we need to adjust for this in the modeling.

Table 3: Top 3 Most Rated Movies

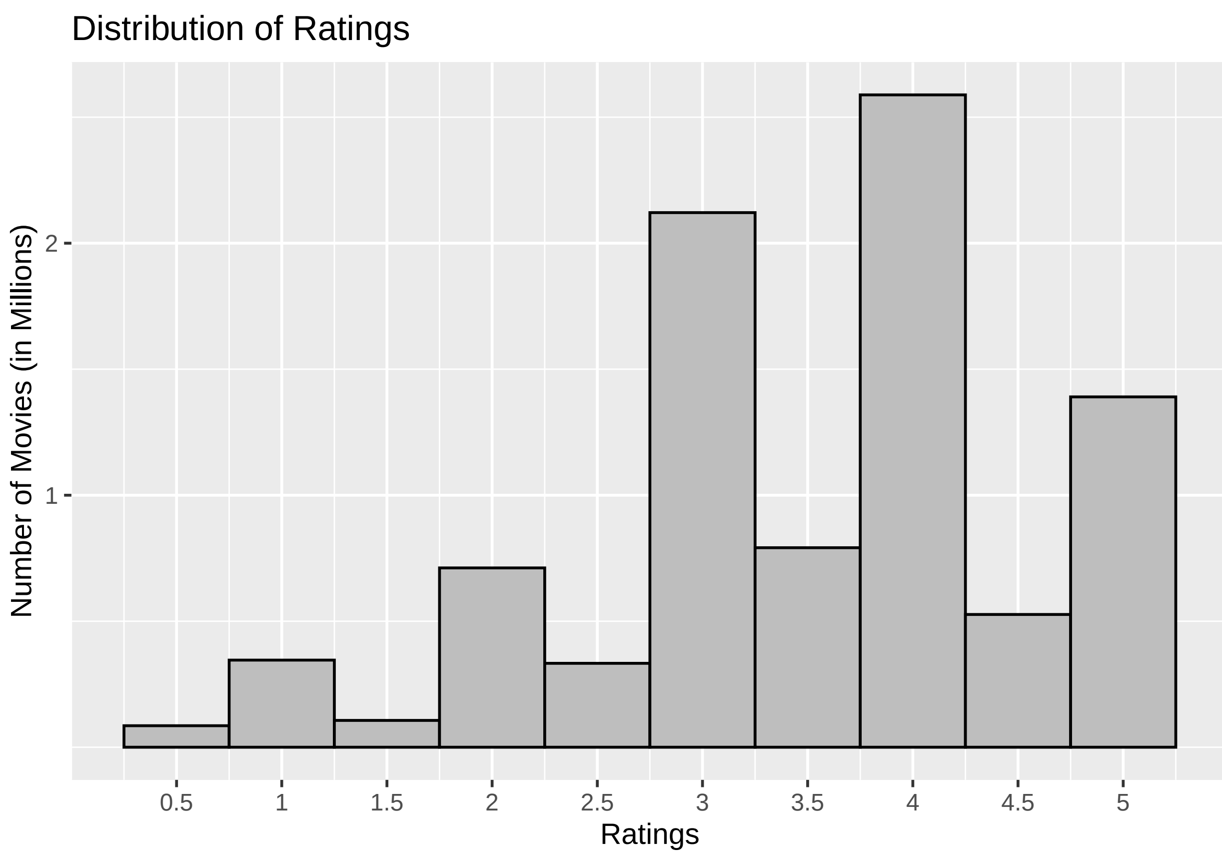
movieId	title	n
296	Pulp Fiction (1994)	31362
356	Forrest Gump (1994)	31079
593	Silence of the Lambs, The (1991)	30382

```
##      movieId                           title n
##   1:    3191             Quarry, The (1998) 1
##   2:    3226  Hellhounds on My Trail (1999) 1
##   3:    3234 Train Ride to Hollywood (1978) 1
##   4:    3356          Condo Painting (2000) 1
##   5:    3383              Big Fella (1937) 1
##  ---
## 122:   64976                  Hexed (1993) 1
## 123:   65006                Impulse (2008) 1
## 124:   65011          Zona Zamfirova (2002) 1
## 125:   65025         Double Dynamite (1951) 1
## 126:   65027         Death Kiss, The (1933) 1
```

We can see that the most common rating is 4, followed by 3 and 5. Half points are less common.

Table 4: Movies Rating

rating	count
4.0	2588430
3.0	2121240
5.0	1390114
3.5	791624
2.0	711422
4.5	526736
1.0	345679
2.5	333010
1.5	106426
0.5	85374



userId

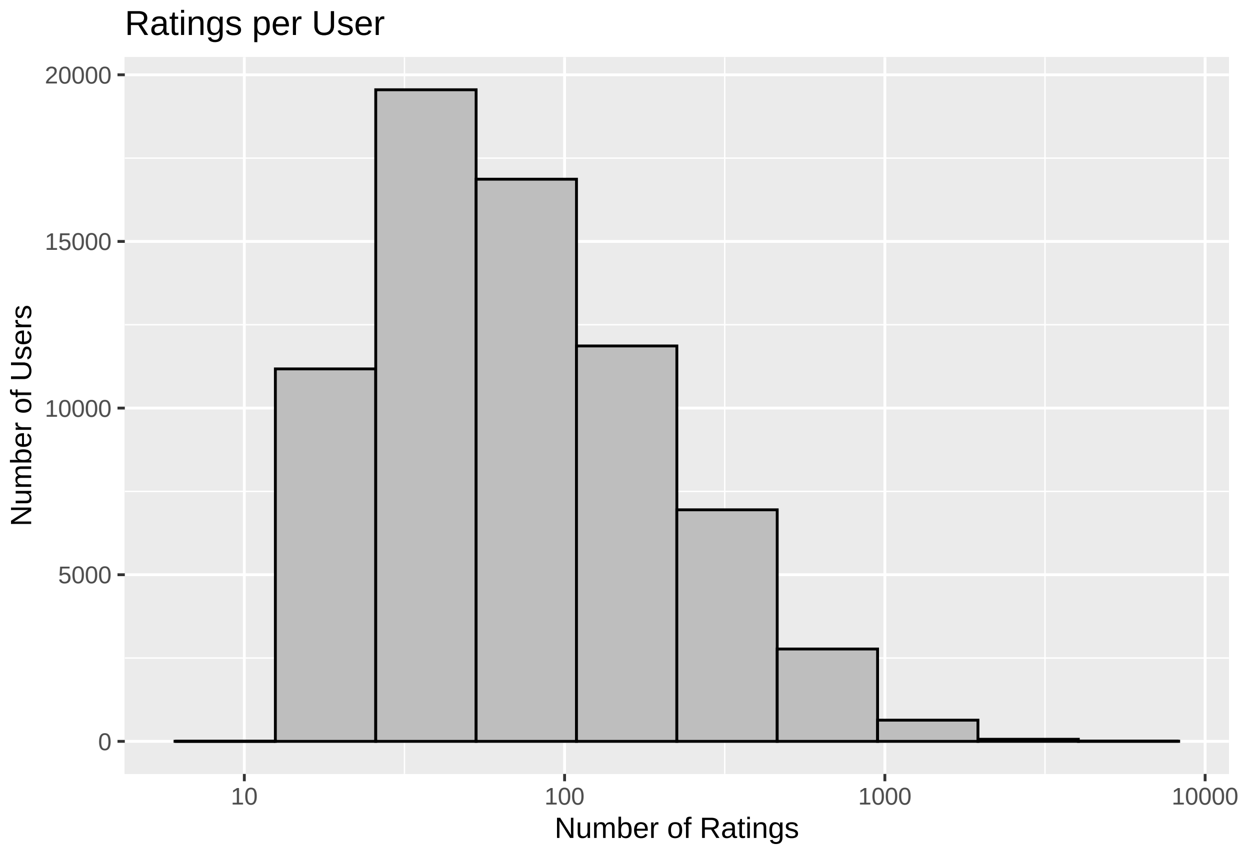
For the userId variable,there were 69,878 unique users. Like in the movie variable, there is a wide spread of rating activity with the users, while there are 610 users that rated more than 1,000 movies; there are 28 that rated less than 15. Most users rated between 20 and 150 movies. After that, the number of ratings declines sharply. As some users are more active than others, these variables also need to be adjusted.

Table 5: Highest number of ratings

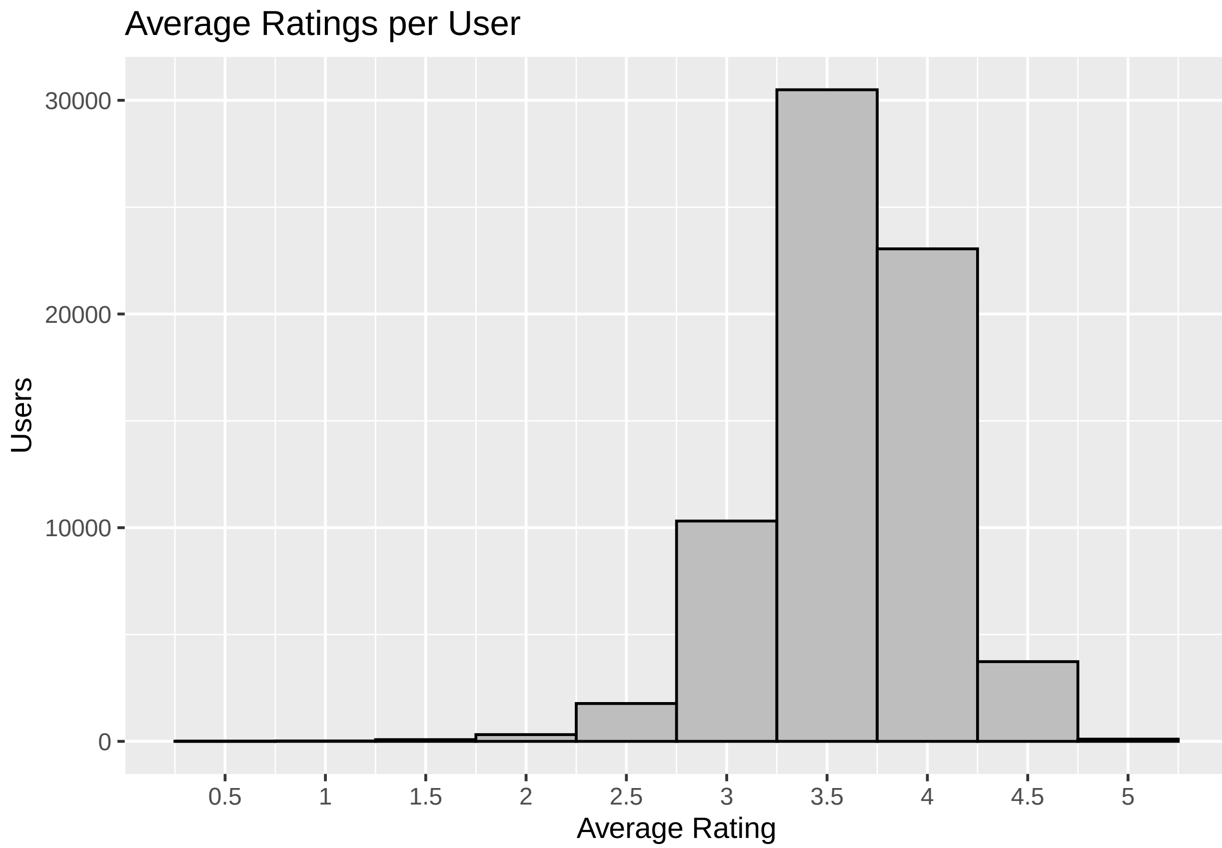
userId	n
14463	4648
59269	6616
67385	6360

Table 6: Lowest number of ratings

userId	n
15719	13
22170	12
50608	13
62516	10



Even though average rating per user range from 2.5 to 4.5, most users stick to the average rating of ~3.5.



It is useful to know the average rating for all movies is 3.51.

genre

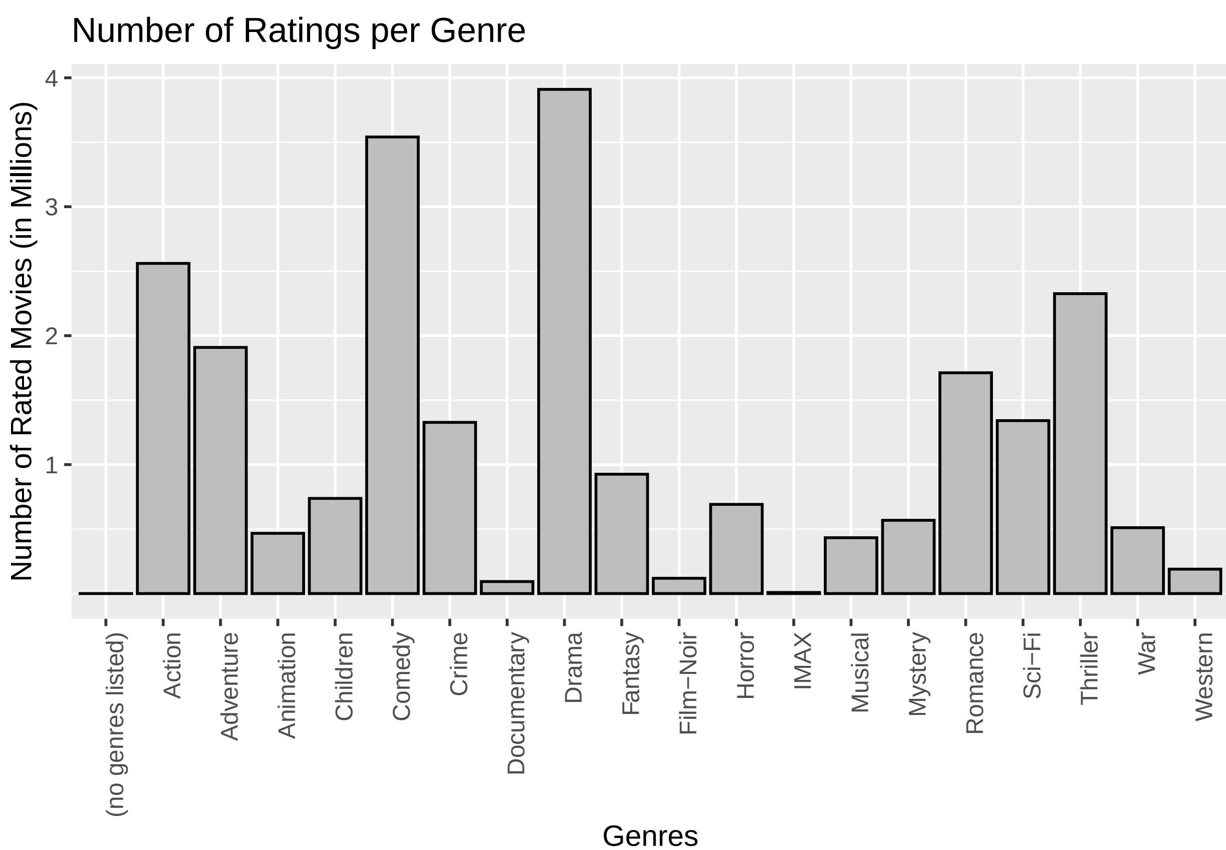
We can see that one single movie belongs to several genres, in order to analyze them, we first need to separate them into individual categories

```
## [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"
```

We can see that the most rated genre is Drama with 3.9 million ratings, followed by Comedy, Action and Thriller, very common movie genres; while the least rated genres are Documentary, Film-Noir and IMAX,somehow less popular genres.

Table 7: Genre Ratings

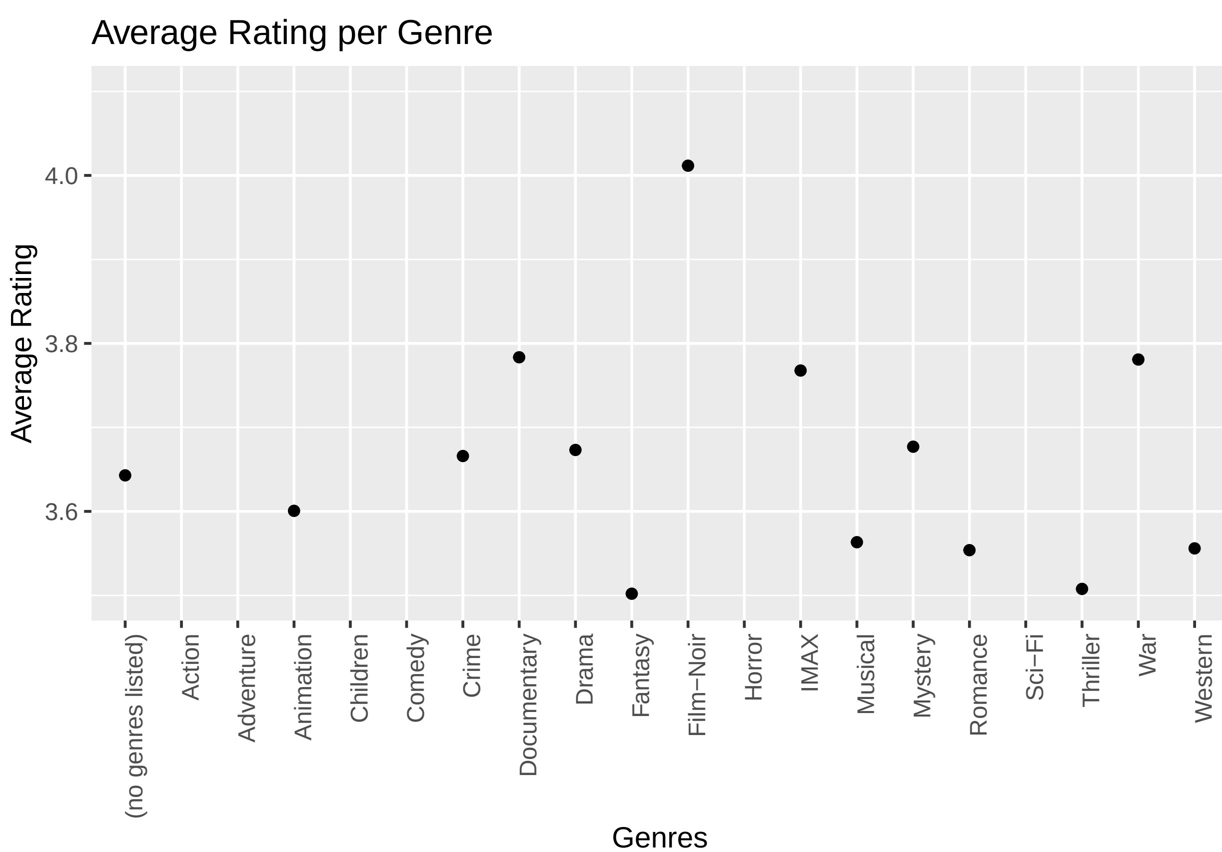
genres	count
Drama	3910127
Comedy	3540930
Action	2560545
Thriller	2325899
Adventure	1908892
Romance	1712100
Sci-Fi	1341183
Crime	1327715
Fantasy	925637
Children	737994
Horror	691485
Mystery	568332
War	511147
Animation	467168
Musical	433080
Western	189394
Film-Noir	118541
Documentary	93066
IMAX	8181
(no genres listed)	7



We can see that the genres least rated have the highest average rating (Film-Noir, Documentary, Imax).

Table 8: Average Rating by Genre

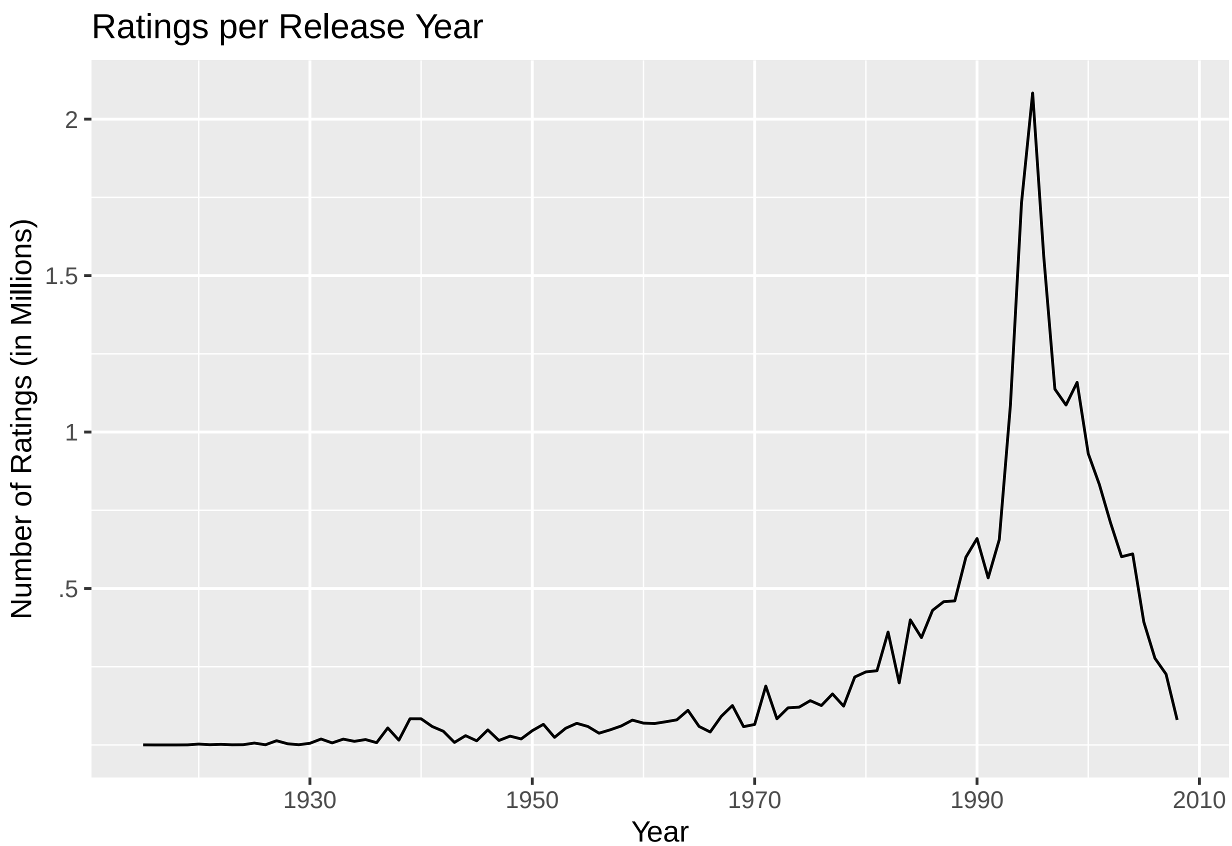
genres	count	mean_rating
Film-Noir	118541	4.011625
Documentary	93066	3.783487
War	511147	3.780813
IMAX	8181	3.767693
Mystery	568332	3.677001
Drama	3910127	3.673131
Crime	1327715	3.665925
(no genres listed)	7	3.642857
Animation	467168	3.600644
Musical	433080	3.563305
Western	189394	3.555918
Romance	1712100	3.553813
Thriller	2325899	3.507676
Fantasy	925637	3.501946
Adventure	1908892	3.493544
Comedy	3540930	3.436908
Action	2560545	3.421405
Children	737994	3.418715
Sci-Fi	1341183	3.395743
Horror	691485	3.269815



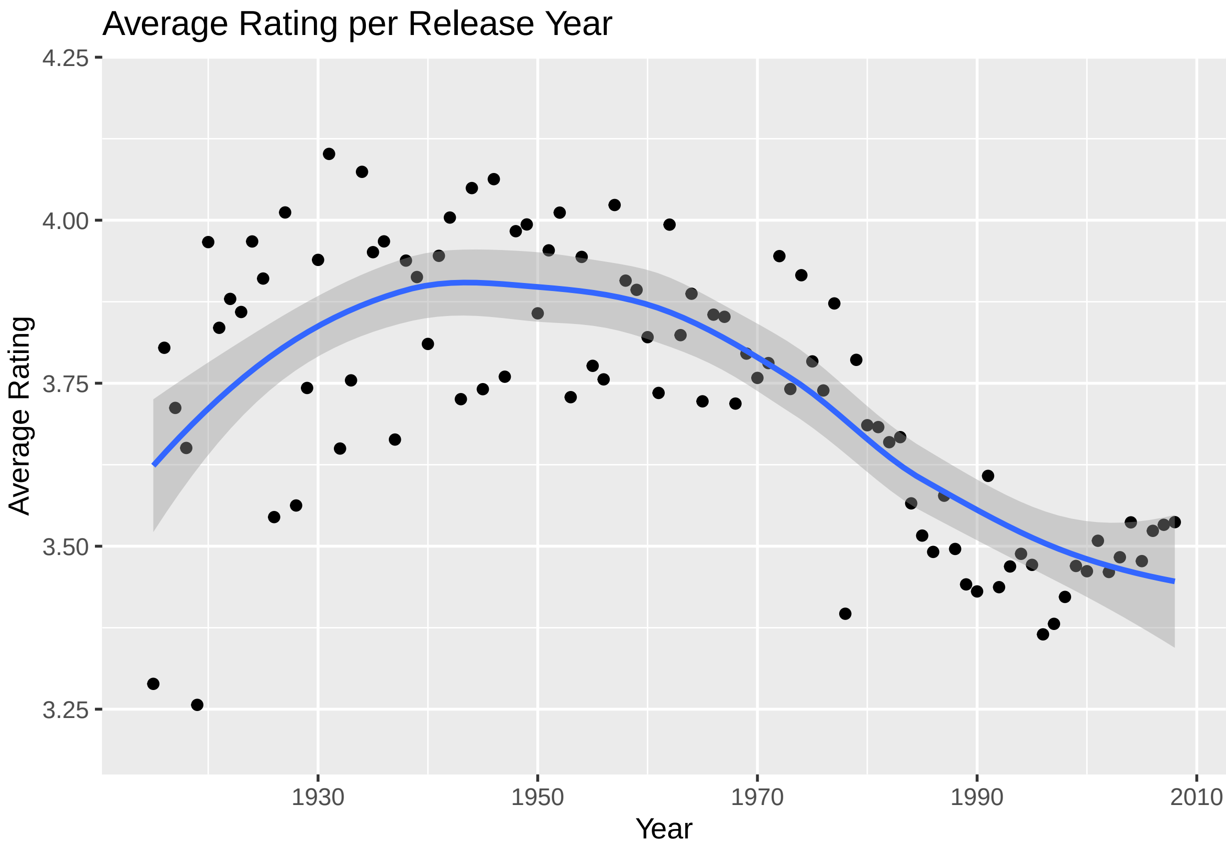
Year

This variable contains the name of the movie and the year of its release. The title is useless for the analysis, but we can extract the release year to check if the age of the movie has an effect on rating. We extract the release year from both the edx and validation sets.

Ratings are not numerous with movies released before 1970, with an upward trend with movies afterwards reaching a peak of over 2 million ratings for movies released in 1995 and then declining all the way to 2007. 2008's very low number of ratings can be due to incomplete data from that year.



There is a higher appreciation for older movies starting in the 1920s and a decline in average ratings from 1980s on.



3. Data Modeling

In the light of the observations given by the variables, we will proceed to the modeling of the algorithm to try to reach the RMSE of less than 0. 86490

First, creating the train and test set and a list to keep record of the results.

Method	RMSE
Objective	0.8649

We will then start the data modeling, first with a naive approach, then including the movie effect and its regularization, after we will add the user effect and its regularization, followed by the genre + user + movie effect and its regularization and finally the year + genre +user + movie effect and its regularization.

Results

1.Naive model = mu + Error

The simplest model possible, we predict the same rating for all movies regardless of the user. It assumes the same rating for all movies and users, where any differences are explained by random variation.

Method	RMSE
Objective	0.864900
Naive Model	1.051984

This naive model returns a RMSE of 1.0519, much higher than the goal. It also means that our ratings will be off by more than 1 point. In order to reach the goal we will try to improve the model by comparing other approaches.

2. MovieEeffect Model = mu + b_i + Error

Data exploration shower that some movies are rated higher than others, we can represent average ranking for movies:

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964

Already a big improvement of 10.55% against the naive model, yet not good enough to reach the goal.

2.1 Regularized Movie Model

Regularization allows us to penalize large estimates that are formed using small sample sizes, like in the case where the best and the worst movies are rated by very few users.

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826

A very small improvement, but there is room for more.

3. User + Movie Effect Model = mu + b_i + b_u + Error

B_u is a user-specific effect that will control for some users giving bad rates to good movies badly and other users giving good rates to bad movies.

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998

Another big improvement of 8.8% reaching 0.8574 and effectively reaching the goal, but let's continue just to see if we can do better.

3.1 Regularized User + Movie Effect

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998
Regularized User + Movie Effect Model	0.8573786

Again, the regularization of the model, only has a slight improvement in the model.

4. Genre + User + Movie Effect Model = mu + b_i + b_u + b_g

Since we saw that some more popular genres were more rated than other more obscure genre and equally some obscure genres were rated higher than others, we add the genre bias to the model.

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998
Regularized User + Movie Effect Model	0.8573786
Genre +User + Movie Effect Model	0.8574106

We see that the Genre+ User+ Movie Effect Model does slightly better than the User + Movie Effect Model but not better than the regularized user+ movie effect. Maybe a regularized version of this combination can do better.

4.1 Regularized Genre + User + Movie Model

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998
Regularized User + Movie Effect Model	0.8573786
Genre +User + Movie Effect Model	0.8574106
Regularized Genre +User + Movie Effect Model	0.8572925

We keep obtaining very small improvements and the computation speed is getting slower, so we will try only one last approach.

5. Year + Genre + User + Movie Model = mu + b__i + b__u + b__g + b__y + Error

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998
Regularized User + Movie Effect Model	0.8573786
Genre +User + Movie Effect Model	0.8574106
Regularized Genre +User + Movie Effect Model	0.8572925
Release Year + Genre +User + Movie Effect Model	0.8570634

We do see an improvement. One last step would be to regularize this last model.

5.1 Regularize Year +Genre +User + Movie Effect Model

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998
Regularized User + Movie Effect Model	0.8573786
Genre +User + Movie Effect Model	0.8574106
Regularized Genre +User + Movie Effect Model	0.8572925
Release Year + Genre +User + Movie Effect Model	0.8570634
Regularized Release Year + Genre + User + Movie Effect Model	0.8569662

6. Validation

Finally: Using RMSE <- function(true_ratings, predicted_ratings){sqrt(mean((true_ratings - predicted_ratings)^2,na.rm = T)), predict ratings on the Validation Set

Method	RMSE
Objective	0.8649000
Naive Model	1.0519843
Movie Effect Model	0.9409964
Regularized Movie Effect Model	0.9409826
User + Movie Effect Model	0.8574998
Regularized User + Movie Effect Model	0.8573786
Genre +User + Movie Effect Model	0.8574106
Regularized Genre +User + Movie Effect Model	0.8572925
Release Year + Genre +User + Movie Effect Model	0.8570634
Regularized Release Year + Genre + User + Movie Effect Model	0.8569662
Validation Set	0.8629447

Conclusion

The use of recommendation systems will only gain in importance because of its usefulness as a marketing scheme (Amazon, Netflix, Spotify), the more systems can catch attention and provoke action, the more demanded their precision will be. In this exercise, we were able to match the Netflix challenge and obtain an RMSE under 0.86490. Yet there is still room for improvement. One way to achieve it is by using linear regression (lm()), but the computing power of personal computers is still limited. Another path to improvement would be to keep adding variables to the model, like the date the movie was reviewed. Overall, we can see that the major improvements were achieved with the Movie Effect over the naive Model and when the User Effect was included in the model.

References

Irizarry, Rafael A., Introduction to Data Science, Data Analysis and Prediction Algorithms with R, 2021-07-03, https://rafalab.github.io/dsbook/

https://www.geeksforgeeks.org/regularization-in-r-programming/