EDX Capstone MovieLens Project

Report and findings

Fernando A. Mitidieri

29/12/2023

1. Introduction

This is a report for the final project of HarvardX Data Science Program to obtain the professional certificate.

The goal of this project is to create a recommendation system to movies based on dataset MovieLens. The dataset will be used to training models that will be validate using RMSE as the loss function to evaluate which one has better performance.

The goal of this project is create a recommendation system of movies for users. It should come up with a movie title recommendation based on the other features presented into dataset.

This project was saved using GitHub to control versions and the source code. Below follow the public address to access all files of this project:

https://github.com/fmitidieri/EDX_Movielens

In the project directory will be found the following files:

1. EDX_Capstone_Final.R - script in R with all development of the model.
2. EDX_Capstone_Final.RMD - RMD file to generate the report in a pdf format.
3. A report in PDF format.

There are also two subdirectories under the main one, that are used to store data and pictures used to generate the report:

1. figures/
2. ml-10M100K/

Movielens is a large and sparse dataset with 10677 movies, 69878 users and 10000054 different rates. Dataset is composed by 6 columns: userId, movieId, rating, timestamp, title, genres.
Below we can verify the type of each feature:

```
## 'data.frame':    10000054 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  122 185 231 292 316 329 355 356 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983392 838983421 838983392 838983392 838984474 838983653 .
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Dumb & Dumber (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Comedy" ...
```

Table: Movielens variable types

To accomplish the goal defined above, will be necessary a bunch of steps that follow:

A. Explonatory Analysis
a) Download complete dataset from http://files.grouplens.org/datasets/movielens/ml-10m.zip. b) Generate edx_original and final_holdout_test sets. c) Pre-processing data. d) From edx_original dataset, generate edx training and validation sets. e) Execute a data exploration and previous analysis of data.

B. Model
a) Loss function: RMSE. b) Develop models. c) Compare results.

C. Test on Final_Holdout_Test
a) Run the final model with final_holdout_test dataset

2. Methods and Analysis

A. Exploratory Analysis

a) Download complete dataset.

Was used the script provided in the assessment to download the movielens data from http://files.grouple ns.org/datasets/movielens/ml-10m.zip, then combine the files ratings.dat and movies.dat to generate the movielens data set.

b) Generate edx_original and final_holdout_test sets.

After that dataset movielens had been downloaded, was created two different datasets:

i) Final_holdout_test - should be used only when the final model had been chose. It is important that final model be tested with a unused dataset. The Final_handout_test set is equivalent of 10% of data from movielens. Total rows in the Final_holdout_test dataset is: 999999

ii) edx_original - From those 90% that wasn't used in Final_handout_test set, it will be used to creat two different sets: edx and validation datasets. Total rows in the edx_original dataset is: 9000055

After created the datasets, they were saved as a Rdata file to be used during the development stage of analysis.

c) Pre-processing data

Before preform the segregation of edx_original dataset, was performed some data transformations on the data. The goal is avoid use strings when running the modeling stage of the work. So, were identified many features and figure out a way to transform them from character type to numerical type. To perform this pre-preocessing and preserve the dataset original, was creates a copy of this and named edx.

c.1) The first data transformation was created a table to relate movieID with title, this will be used to handle only Ids as numerical data instead handle title that are strings. As result, was created a table with movieId and movie title.

c.2) The second pre-processing was identify the year of issuance from the title column for each movie, take out of the string and transform it to be used as a new numerical feature. Was created a new column in the edx dataset.

c.3) In context of this analysis month and day of movie launching do not add as much information and would certainly causes increase in processing time, so a decision was made to keep only the year of movie launching and transform the date column into a year column. And then timestamp column that will no longer be used was deleted from the dataset.

c.4) The genre column was substituted by a reference number. Again, process numbers is easier than process strings. The idea is substitute each combination by a number, For each different combination of genre will be assigned a unique number. Now that was generated the genreNumber table, these numbers will replace the strings of characters of the genre in the edx data set.

```
## 'data.frame':    8100065 obs. of  6 variables:
##  $ userId    : int  1 1 1 1 1 1 1 1 ...
##  $ movieId   : int  122 292 316 329 355 356 362 364 ...
##  $ rating    : int  5 5 5 5 5 5 5 5 ...
##  $ genres    : int  1 4 5 6 3 3 9 10 ...
##  $ year_title: int  1992 1995 1994 1994 1994 1994 1994 1994 ...
##  $ Year      : int  1992 1995 1994 1994 1994 1994 1994 1994 ...
```

Table: edx dataset after data transformation

d) From edx_original dataset, generate edx training and validation sets.

From the edx_original was created two different sets: edx and validation.

d.1) edx - has 90% of data from edx_original dataset and it is to be used to training the models. Total rows in the edx dataset: 8100065

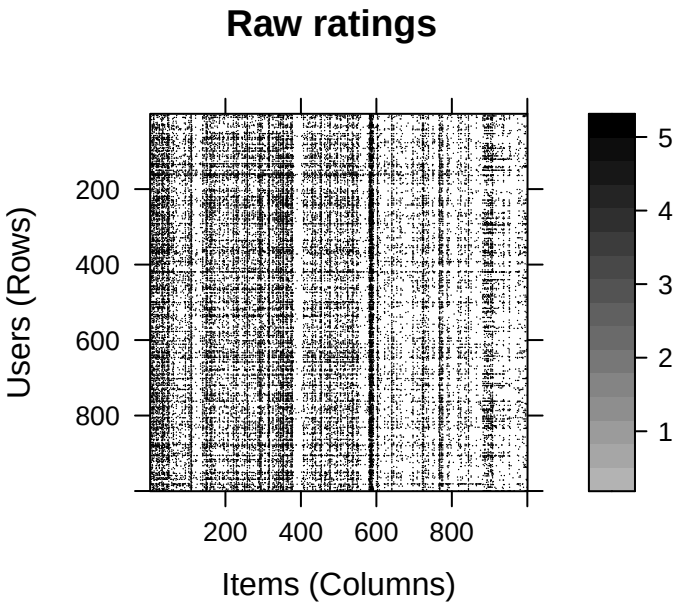
d.2)validation - has 10% of edx_original dataset and it is used to assess the models before the final validation with the Final_holdout_test dataset. Total rows in the validation dataset: 899990

e) Execute a data exploration and previous analysis of data.

First of all, the edx dataset is inspected:

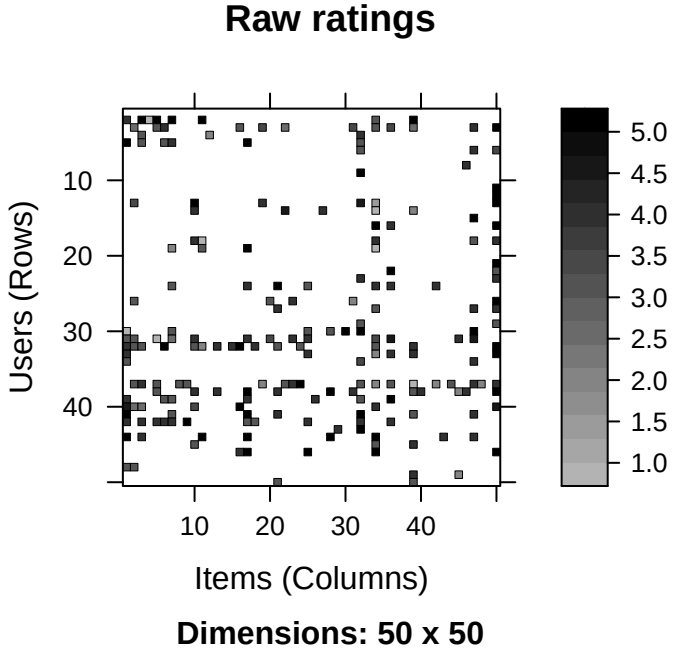
```
## Rows: 8,100,065
## Columns: 6
## $ userId     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,~
## $ movieId    <int> 122, 292, 316, 329, 355, 356, 362, 364, 370, 377, 420, 466,~
## $ rating     <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 3,~
## $ genres     <int> 1, 4, 5, 6, 3, 3, 9, 10, 3, 12, 3, 3, 3, 3, 3, 19, 20, 21, ~
## $ year_title <int> 1992, 1995, 1994, 1994, 1994, 1994, 1994, 1994, 1994, 1994,~
## $ Year       <int> 1992, 1995, 1994, 1994, 1994, 1994, 1994, 1994, 1994, 1994,~
```

One important characteristic of this dataset is that there are many users and many movies, but few rating if we consider all possibilities. Bellow is possible to analysis how sparse the movielens data is. The size of matrix used to generates the plot was constrict into 1000x1000 users by items.



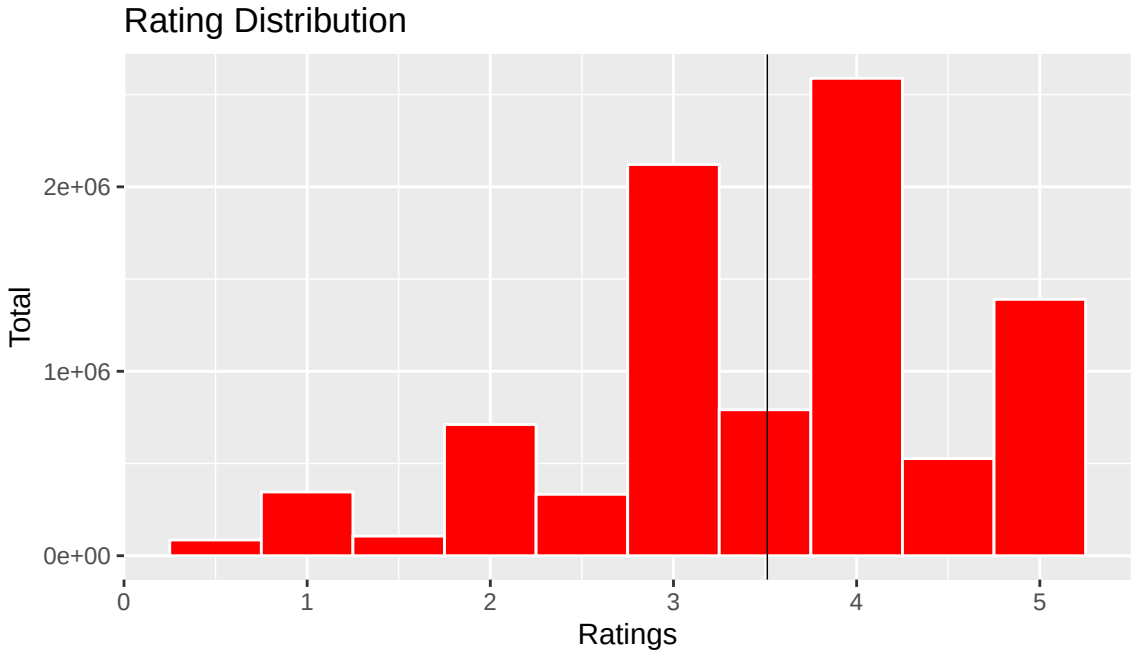
We can be more accurate and calculate the percentage of missing data in this matrix. The dimensions of this matrix are 10677 columns per 69878 rows. It means that we should have 746087406 possible combination between users and movies, but only 8100065 ratings in the dataset. So, the dataset has 98.91% of missing data. Basically this matrix is made of a bunch of zeros.

Just to make it more visual, the plot above is plotted again but now with only 50x50 users and items.



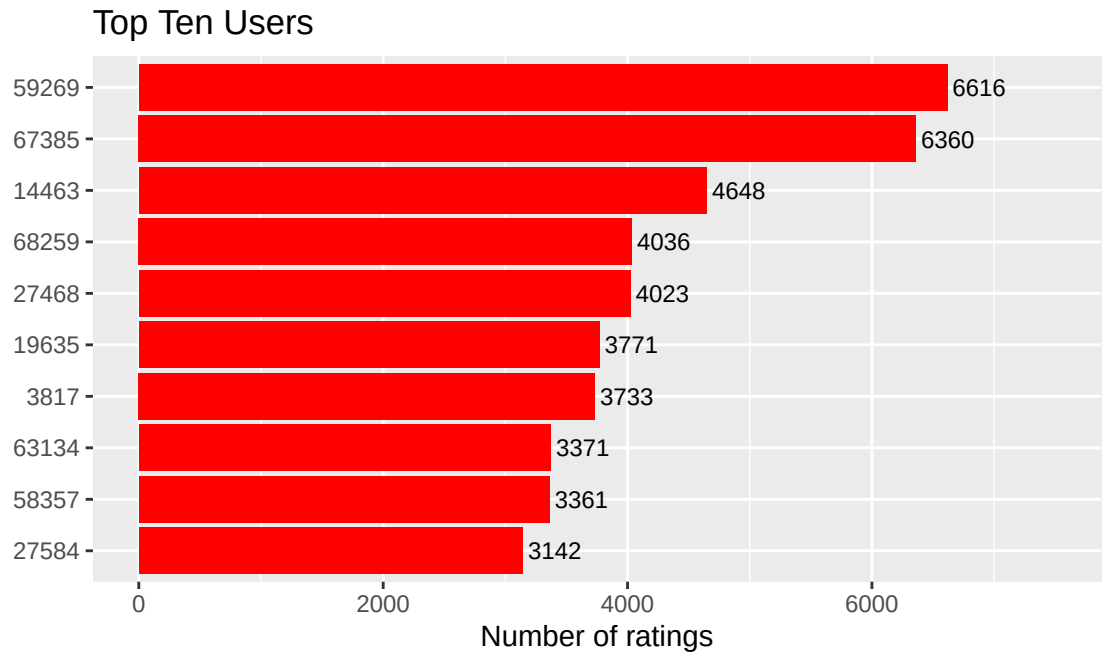
This characteristic of such matrix shows that we could expect two difficulties when working with it: computational complexity and storage necessity.

A natural analysis is find how ratings are distributed along data. It is possible to discover that the mean is around 3.51 stars and that users trend to give higher ratings.



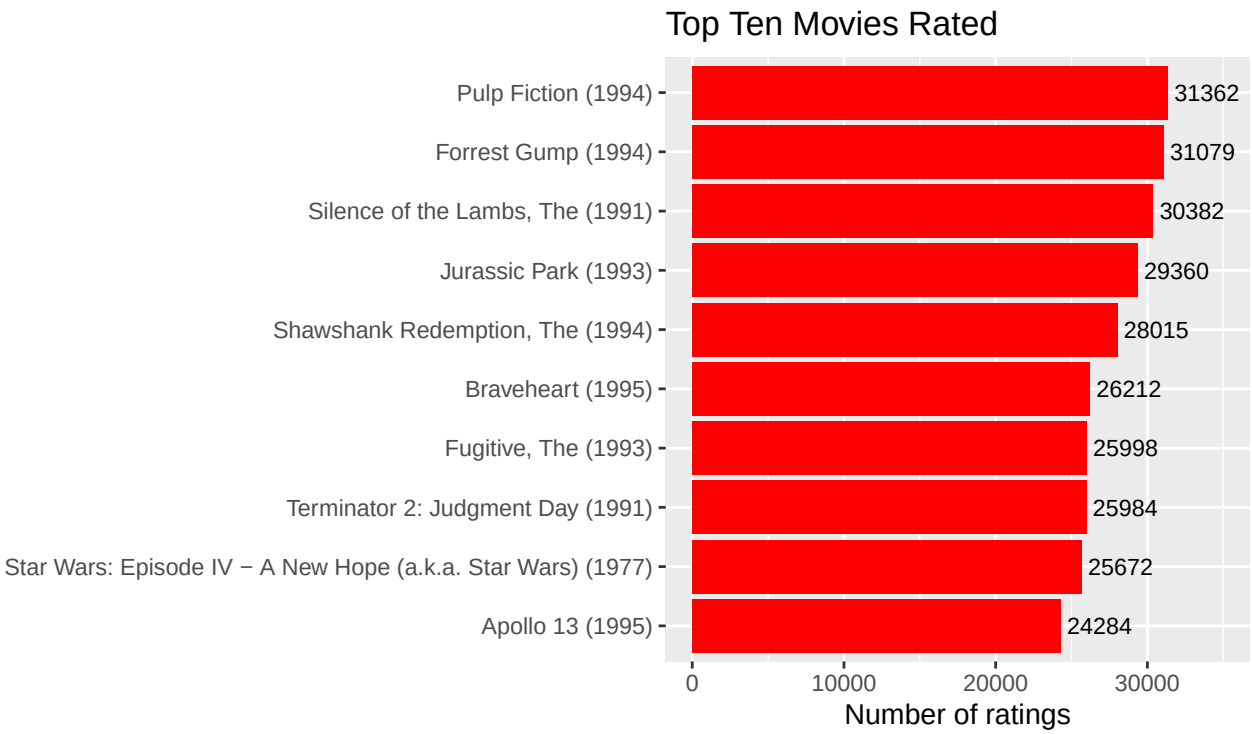
It is clear from the picture above that the users has a trend to give higher ratings and that the rounded ratings are much more common that the rating with half star. it shows a bias from user perspective in relation to the ratings.

We also can see that the Top 10 users could be considered heavy users, with a high number of movies rated.

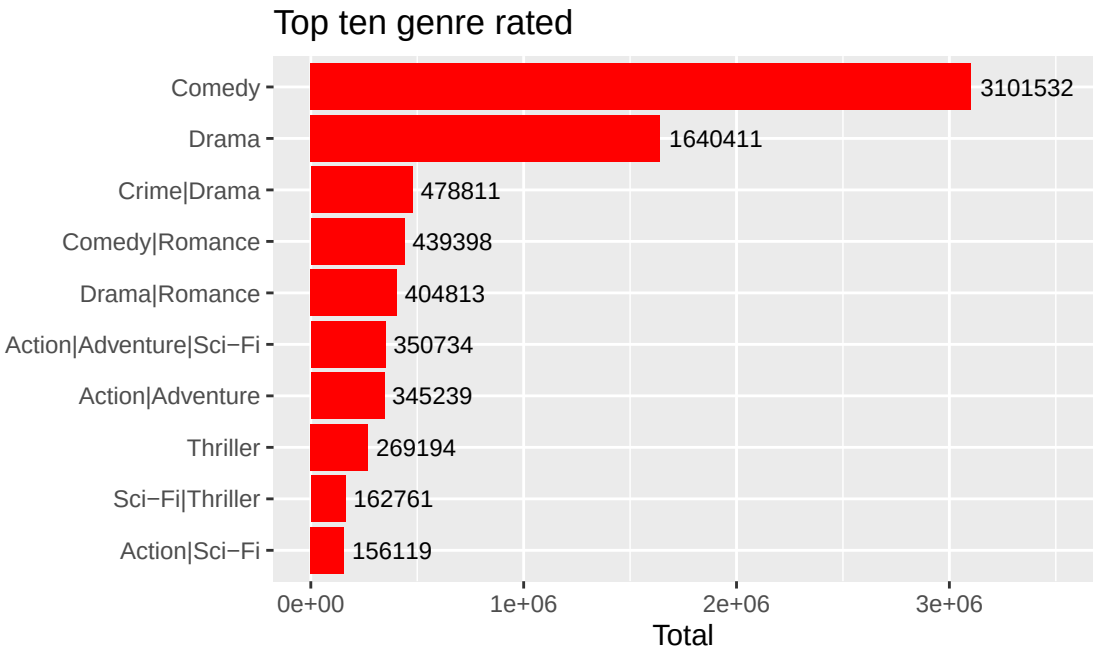


The next analysis is the Top 10 most rated movies in this dataset. We can note that 9 to 10 movies are from

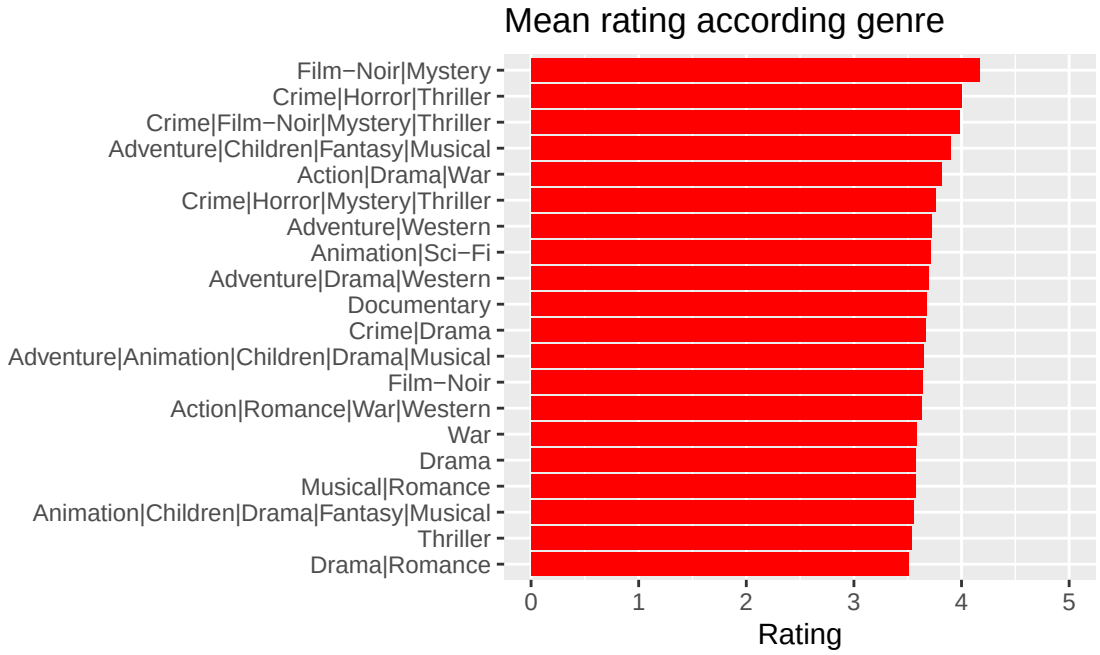
nineties. Maybe because the rating systems was adopted in this decade and naturally the new launches are more rated then the older ones.



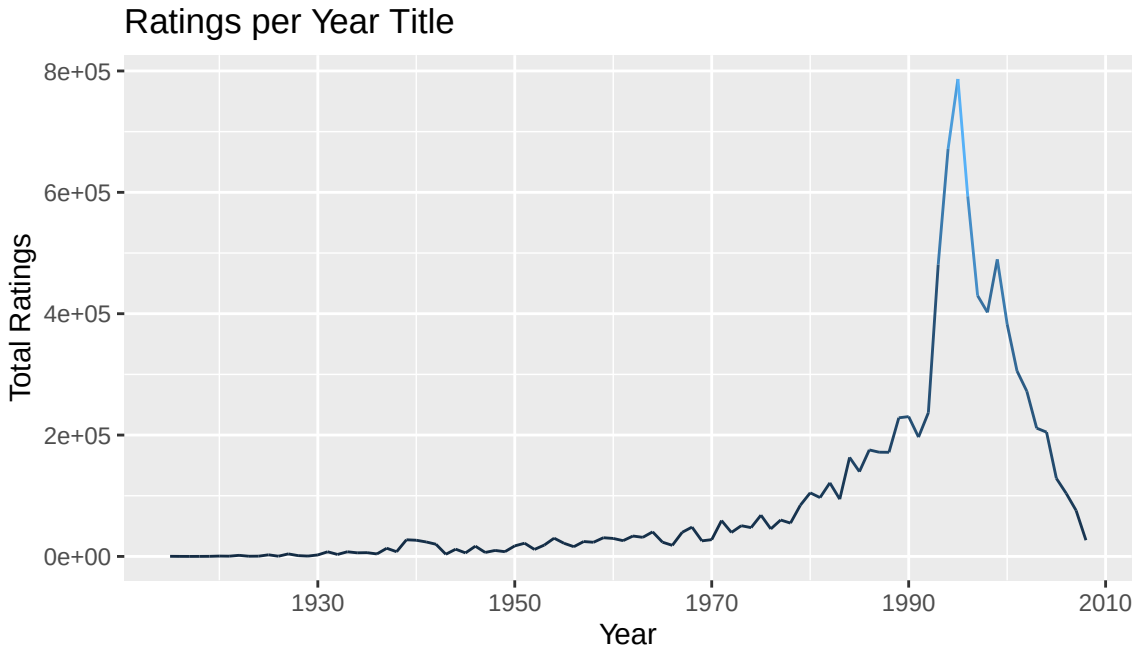
And when we look at the most rated Genres, the highlight is that the first and second genres are so far the most rated. Comedy is approximately double that of the second (Drama) and 6 times that of the third (Crime/Drama). It is probably because these two genres are users' preference and would naturally receive many more reviews than the other. Another piece of information, perhaps, is that there are many more films in these two genres.



A complement of the data view above is the distribution of the ratings among the genres. This figure confirms the bias of the Users rate highest number of stars.



And finally we can see that the distribution of the ratings along the years. Once more it is possible highlight the nineties with a boom in number of ratings. This confirms that user adopted the rating habit in this decade.



B. Model

a) Loss function: RMSE.

The loss function chose for evaluation of the results is RMSE - Root Mean Square Error. This function can be understand as a typical error between the prediction and the true rate.

The goal is make it smaller as possible, which means that our prediction is more accurate.

$$RMSE = (\frac{1}{N} * \sum_{u,i} (\bar{y_u}i - y_u i)^2)^{\frac{1}{2}}$$

b) Develop models.

The approach to solving this problem was to consider the effects of each feature on the rating given to a given film.
The model was constructed feature by feature and the results obtained at each stage were measured using the loss function.
The final model contains all the features designed in the pre-processing phase and also resulted in an RMSE lower than the target proposed for the challenge.

b.1) First factor: mean
In this first step it is calculates the mean for all movies and then used as a prediction of the future ratings. Off course it is a poor approach and the error generated is large.

$$\bar{Y} \sim \mu + \epsilon$$

The first model resulted in the following RMSE:

Table 1: Mean

Method	RMSE
Mean	1.060054

b.2) Second factor: MovieId Bias
Now we will include the effect of movisId in the prediction.

$$\bar{Y} \sim \mu + b_i + \epsilon$$

As expected, the effect of the movieId bias decreased the RMSE:

Table 2: Mean+bi

Method	RMSE
Mean	1.0600537
Movie Bias	0.9429615

But it is still far from the objective set by the challenge. We can do better.

b.3) Third factor: UserId Bias
Now we will include the effect of movisId in the prediction.

$$\bar{Y} \sim \mu + b_i + b_u + \epsilon$$

As expected, the effect of UserId bias contributed to further decreasing the RMSE:

Table 3: Mean+bi+bu

Method	RMSE
Mean	1.0600537
Movie Bias	0.9429615
User Bias	0.8646843

b.4) Fourth factor: Release Year Bias
We will now include the effect of the year the film was released.

$$\bar{Y} \sim \mu + b_i + b_u + b_y + \epsilon$$

Once again we had a drop in RMSE caused by the effect of the film's release year:

Table 4: Mean+bi+bu+by

Method	RMSE
Mean	1.0600537
Movie Bias	0.9429615
User Bias	0.8646843
Year Title Bias	0.8643301

b.5) Fifth factor: Rating Year Bias

Now we will consider the year in which the film was rated.

$$\bar{Y} \sim \mu + b_i + b_u + b_y + b_Y + \epsilon$$

Once again we have a better RMSE result. Already below the target for the challenge. But we can still do better.

Table 5: Mean+bi+bu+by+bY

Method	RMSE
Mean	1.0600537
Movie Bias	0.9429615
User Bias	0.8646843
Year Title Bias	0.8643301
Year Bias	0.8642730

b.6) Sixth factor: Gender bias

Finally, we will include the effect of gender bias in our model.

$$\bar{Y} \sim \mu + b_i + b_u + b_y + b_Y + b_g + \epsilon$$

It seems that we achieved a good result, and at the same time there was no significant decrease in RMSE, as can be seen below:

Table 6: Mean+bi+bu+by+bY+bg

Method	RMSE
Mean	1.0600537
Movie Bias	0.9429615
User Bias	0.8646843
Year Title Bias	0.8643301
Year Bias	0.8642730
Genre Bias	0.8641196

C. Test on Final_Holdout_Test
a) Run the final model with the final_holdout_test dataset

After the model's RMSE results in a value below that established in the proposed challenge, it is time to apply it to the Final_Holdout_test dataset.

The first step is to preprocess the final_holdout_test set for compatibility with the developed process. Basically, the pre-processing done in the edx and validation sets will be repeated: Use MovieId in place of the title, create numeric columns with the year of classification and year of issue of the film, and replace genre strings with codes. The main idea of all processes is to use numbers instead of strings.

Additionally, the model will be retrained using the entire edx_original dataset, i.e., using the full dataset segregated in the first step of the R script.

Table 7: Final_Holdout_test

	Method	RMSE
7	Final Result	0.864778

The result obtained with the final model applied to the Final_Holdout_test dataset resulted in a lower RMSE than defined in the challenge.

3. Results

As is possible to note in the table below the model with only the movieId and userId bias had a result that already was enough to the challenge, but using other factors increased the model performance and helped in the final result. When applied to the Final_Holdout_test dataset the RMSE increased a little, but still became below the target defined in the assessment.

Table 8: Final Results

Method	RMSE
Mean	1.0600537
Movie Bias	0.9429615
User Bias	0.8646843
Year Title Bias	0.8643301
Year Bias	0.8642730
Genre Bias	0.8641196
Final Result	0.8647780

4. Conclusion

Although the objective has been achieved, there is room for much more improvement in the models such as regularization, SVD analysis for feature reduction, combination of other models, feature engineering to discover other relevant variables, among others.

This project was quite challenging and the best opportunity to apply all the knowledge acquired in the previous modules of this course.

Basically we were able to apply what was learned in each module, but it was still necessary to research and look for other sources of information. The internet and R's online documentation were a valuable source of reference and helped with some details of the language.

Finally, this is just the beginning of the Data Science journey. Projects like this motivate us to continue learning and further improve our knowledge.

Even when we achieve a reasonable result, we know we can always do better.

References

[1] Hashler, M. 2018. recommenderlab. A Framework for Developing and Testing Recommendation Algorithms. (Jun.2018).
[2] Hadley W. and Garret G. 4th Edition. R for Data Science.
[3] Chen, E. 2011. Winning the Netflix Prize. A Summary.(Oct. 2011).