Building a Model to Predict Movielens Ratings

Matheus Correia Ferreira

12/18/2020

1. Introduction

The project that is documented here was done as part of the Data Science Professional Certificate course by HarvardX. In it, the construction of a model that seeks to recommend movies to users will be described.

Across its sections, the report intends to take readers step by step through the process that led to the final product. For that purpose, the sections of this paper will include: a short description of the problem; a look into the employed dataset; insightful visual analyses of the records; the translation of those ideas into statistical models; the methodology executed in the experiments; and, at last, the final results obtained.

2. The Problem

The challenge of recommending movies, as well as any other product, to users can be looked at as the task of predicting whether or not the person in question will enjoy the item that is being selected. If we take into consideration that the system where the recommendation model is inserted gives its users the opportunity to rate what is being consumed, then recommendation can be boiled down to accurately predicting what grade a specific user will give to the item in question. Considering our predictions on that front are correct, all the recommendation system will have to do to perform its job rather well is pick the items it thinks the user will rate the highest and deliver them to their screen.

The problem proposed for this work is exactly that. Given a system keeps a large database with how users have rated movies over the years, what other films could be recommended to them. As we go through this paper, the road that leads to building a solution of the kind will be described and the results obtained by the constructed model will be reported.

3. The Data

Movielens [1] is a recommendation system and online community that, over the years, gathered a considerable amount of information on how users rated movies, and this information was promptly made available to the scientific community, especially those who work on data-related fields.

The Movielens dataset [2] comes in various sizes, from one containing 100,000 ratings to another holding 25,000,000. The one used to develop this work has 10,000,000 records, and each one of them has the following format.

userId	movieId	title	genres	timestamp	rating
1	122	Boomerang (1992)	Comedy|Romance	1996-08-02 11:24:06	5
1	292	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller	1996-08-02 10:57:01	5
1	316	Stargate (1994)	Action|Adventure|Sci-Fi	1996-08-02 10:56:32	5

As it's possible to observe, the dataset, as presented for this project, has six columns:

- userId: A number identifying each user. There is a total of 69,878 users in the dataset.
- movieId: A number identifying each movie. There is a total of 10,677 movies in the dataset.
- title: The title of the movie, with its year of release between parentheses.
- genres: The genres of the movie. Movies with multiple genres present those separated by a "|". Considering the groups as unique genres, there is a total of 797 distinct combinations in the dataset.
- timestamp: The date and time when the rating was given.
- rating: The objective of the predictions. The rating given by the user to the movie. Coming in increments of 0.5, it can range between 0.5 and 5. As previously stated, there is a total of 10,000,000 ratings in the dataset.

4. The Datasets and The Methodology

In order to build the models according to good practices, the full dataset was split into three parts that, from this point forward, will be referred to by their names here established.

- Validation Dataset: Containing 999,999 lines, it is made up of roughly 10% of the full dataset. In it are the ratings the model will try to estimate.
- Training Dataset: Originally provided by the staff of HarvardX as the remaining 90% of the dataset, thereby containing about 9,000,000 lines, it was further split into two. 80% of its total (about 7,200,000 lines) was kept as a training dataset, which means it was the data used to not only build the models themselves, but also to support the analyses that were done.
- Test Dataset: The remaining 20% (approximately 1,800,000 records) of the original training dataset was allocated to this test set. With it, the models (constructed using the training dataset) were evaluated according to their performance. Such a step allowed the model that performed the best on the test set to be selected to evaluate the validation dataset.

As such, the methodology employed to build the model for the recommendation system was as follows:

1. Models were built exclusively with data available on the training set.
2. After being built, they were evaluated on the test set.
3. The one with the best performance on the test set was then selected to predict the ratings of the validation set.
4. The predicted ratings were compared to the actual ratings and a final result was obtained.

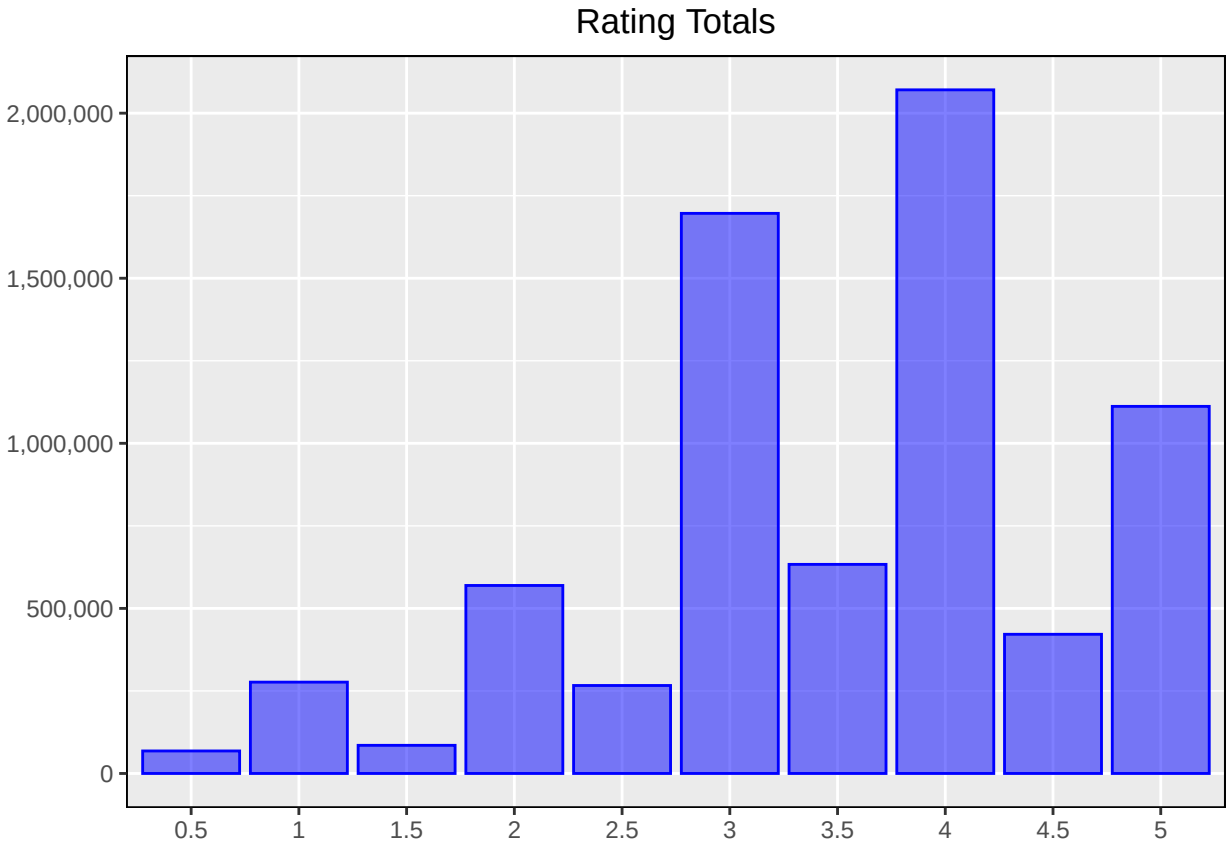
In all cases, models were evaluated according to the RMSE (Root Mean Square Error) metric [3], defined by the formula below. Essentially, it is the squared average of the differences between the predicted and the actual ratings.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(predicted_i - actual_i)}{N}}$$

5. Data Exploration

It is important to note that all plots made and statistics computed in this section were produced exclusively by using the training set. As such, any insights emerging from those were not contaminated with information coming from the test and validation sets.

Firstly, we observe the distribution of the ratings within the dataset through the plot below. As it can be seen, there are far more ratings that lean towards the positive side of the spectrum than to the negative one, with ratings 3 and 4 being the most common. In fact, the average rating found in the training dataset confirms that, as its value is 3.512407. Another point that can be made is that half-ratings tend to be less abundant than full ones.

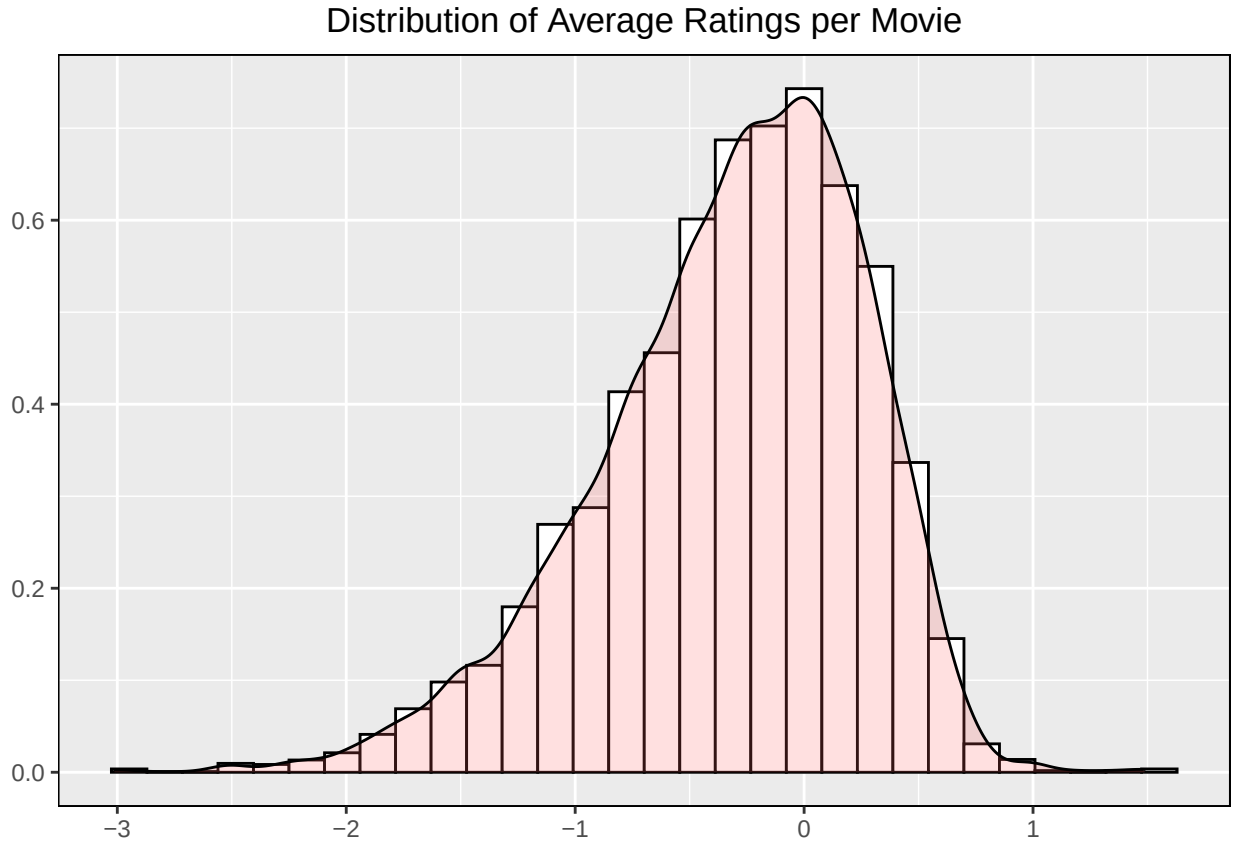


Given we are building a recommendation system, it is essential that we capture the factors that determine how a user will rate a given movie. Within the dataset provided by HarvardX, there are five features that stand out as having the potential to do so, with some having - at least intuitively - a stronger influence than others. These factors are:

- The movie;
- The user;
- The genres of the movie;
- The release year of the movie;
- The time when the rating was given.

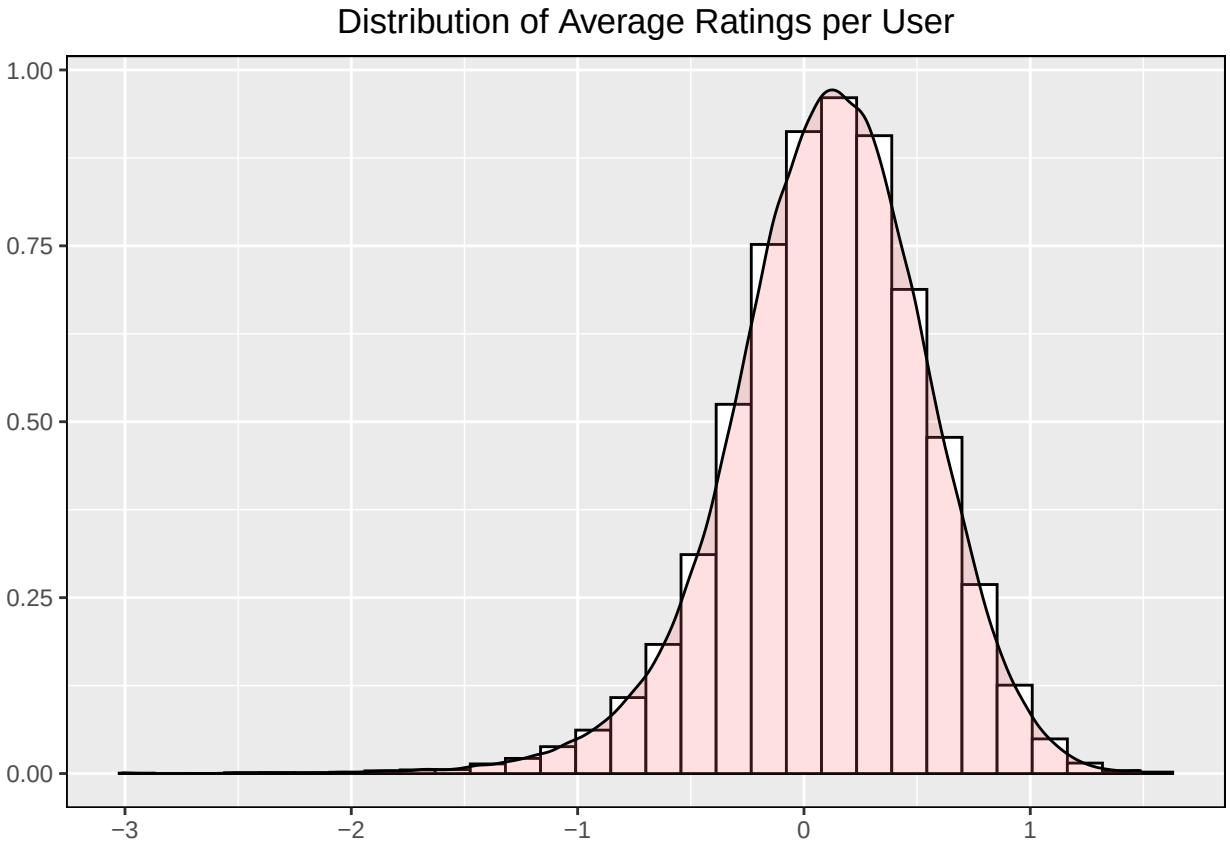
Through the next five charts, we attempt to demonstrate that these features all contribute to the score. For

starters, we look at the most obvious one: the movies themselves. It is only to be expected that, the better a movie is, the higher the average of the ratings it receives will be. Another exepectation is that the average rating of most movies will fluctuate towards the dataset's average, with a smaller amount of them (the ones that are either very good or very bad), breaking that pattern. We see those facts in the following histogram, which displays the distribution of the rating averages for all movies. Note that the values were all centered around the dataset's average.

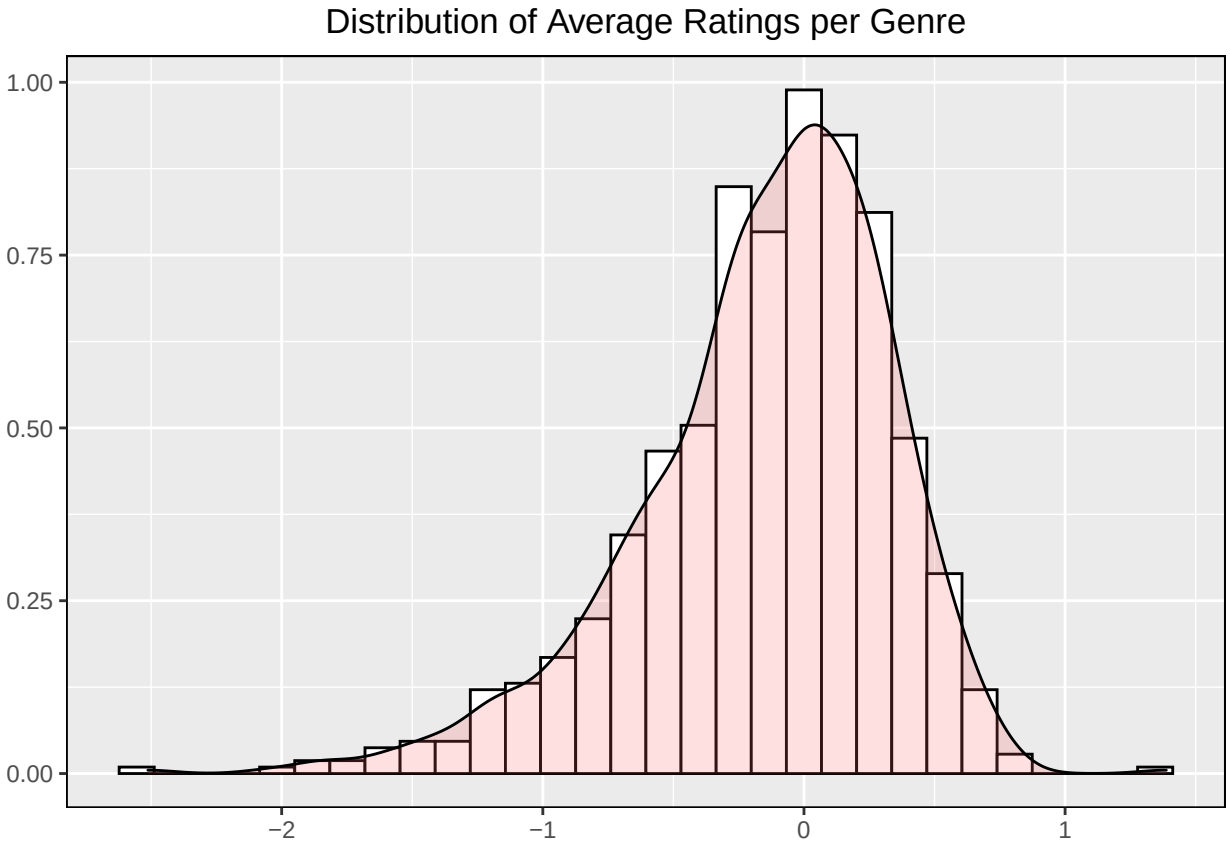


Next, we do the same for users and genres. With the former group, deviations from the average could be explained due to the fact some people are simply harsher than others when giving out ratings, or perhaps they don't have a very good taste when it comes to selecting what they watch. Meanwhile, with the latter, maybe there are some genres that are simply better evaluated than others as a whole: for instance, it is generally said that movie critics tend to love dramas, but are colder towards comedies, horror movies, and romantic comedies.

Does a similar pattern arise when it comes to movie fans giving out ratings? Fist, we look at the users. The distribution appears to be more heavily centered towards the average, which means user bias does not play as considerable of a role as the movies themselves do when it comes to the rating. Yet, there is still a pattern of more forgiving and more strict users.



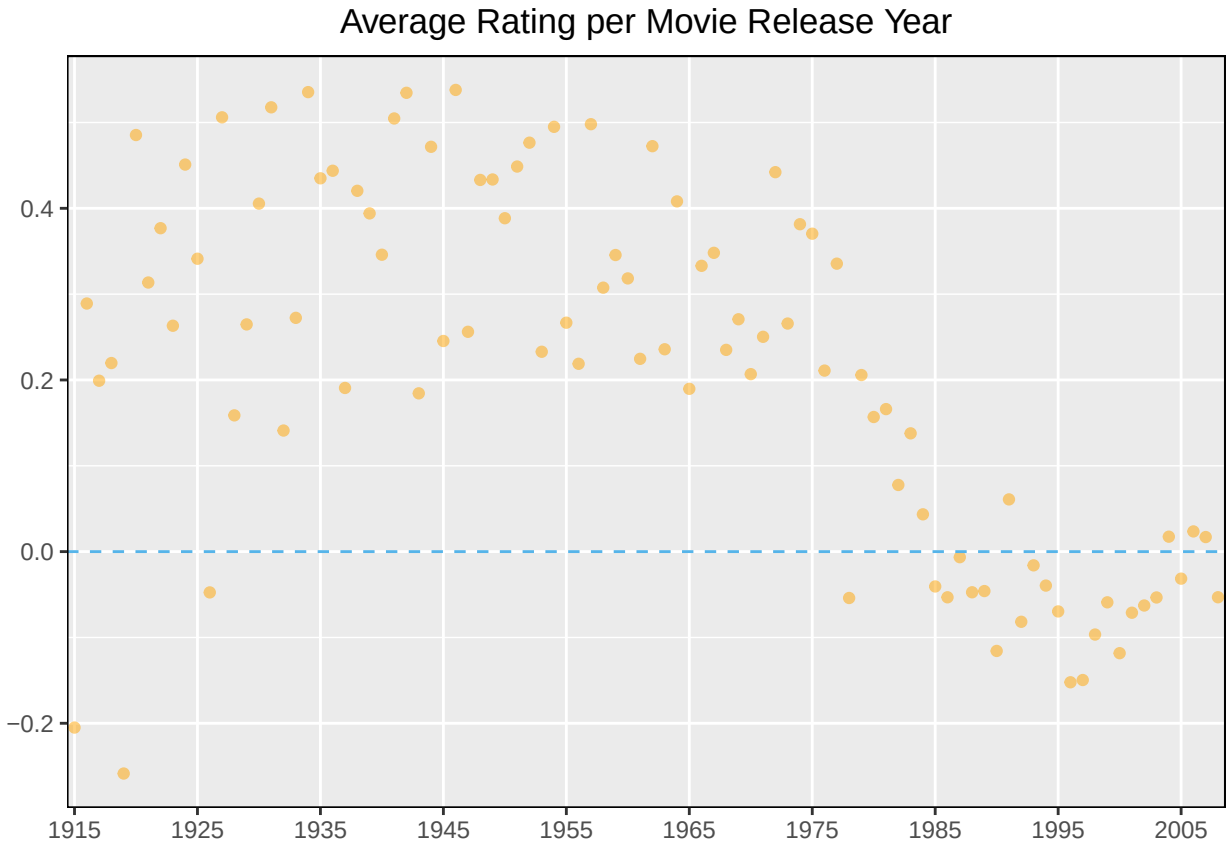
We then get to the genres and, as it happened with users, the curve that appears is not as wide as the one observed for movies, which implies genres are not as determinant to ratings as the films themselves, which is to be expected. However, once again, there is a pattern there, with a few genres tending to deviate from the average.



Next we look at the effects time has over the ratings given. In the case of the Movielens dataset, time emerges in two forms. Firstly, it appears in relation to the year the movie was released, a value that is originally part of the *title* column but that was, for the purpose of this experiment, extracted and placed in a standalone field. Secondly, time is present in the *timestamp* feature, which records when the rating was given.

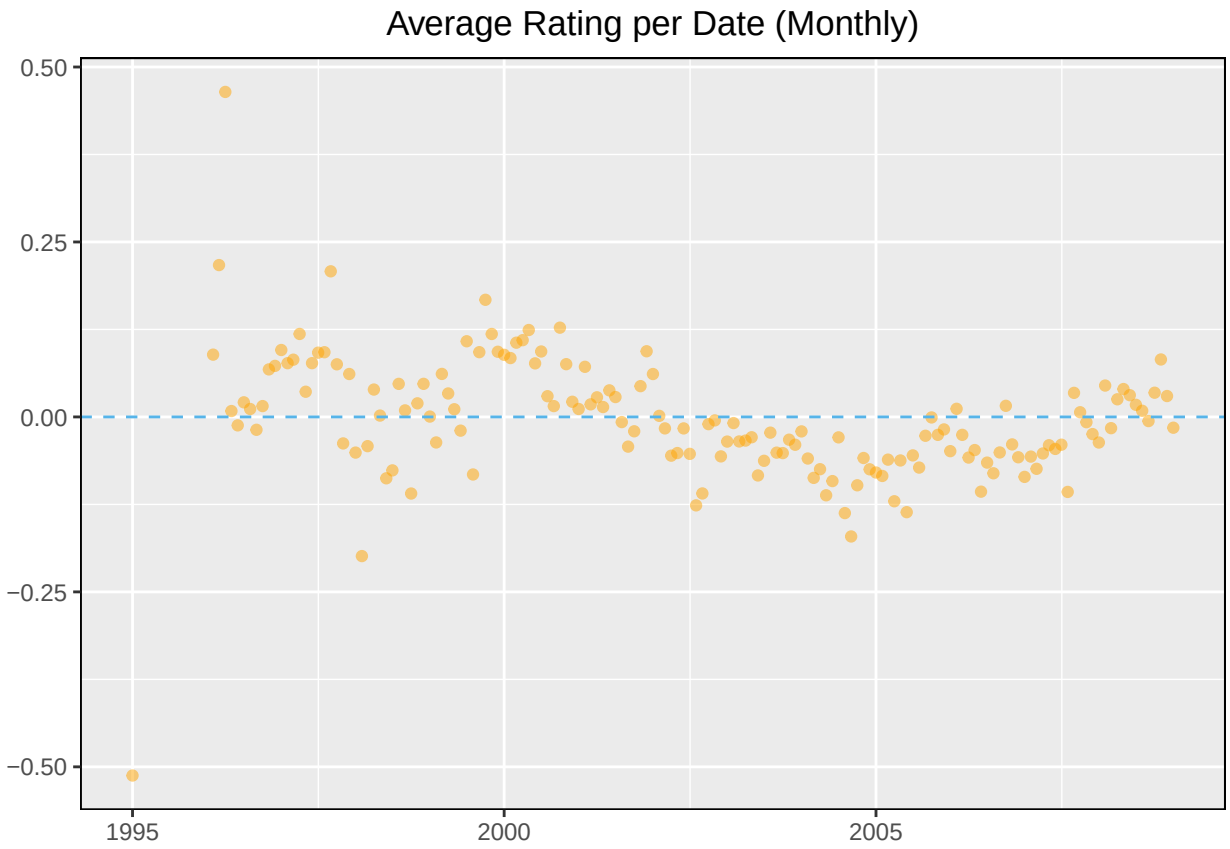
Does the year of a movie's release affect the final rating? In this case, if it does happen, it could be that some years present a higher amount of critically acclaimed titles. Moreover, maybe users are nostalgic and rate older flicks more kindly.

The plot below, with dots representing average ratings (centered around 0) according to the year confirms a tendency, even if the distances from the average are not very considerable. With movies released after 1975, there is a clear progressive approximation towards the total dataset average (represented by the blue dashed line). And before that year, numerous are the cases where the yearly average is 0.4 points above the dataset mean.



Lastly, as far as possible influences towards the rating, there is the moment when it was given. Movielens contains scores recorded between 1995 and 2009. As time passed, did scoring tendencies change? The chart below seeks to determine just that, plotting the monthly average of the ratings.

Here, signs of a tendency are much lighter, as all points fluctuate around the average. Still, there appears to be a somewhat constant shift in giving ratings below and above the average. Note that, for this plot, with the goal of clarity, ratings were grouped according to the month when they were given. For the models, though, the weekly average was considered.

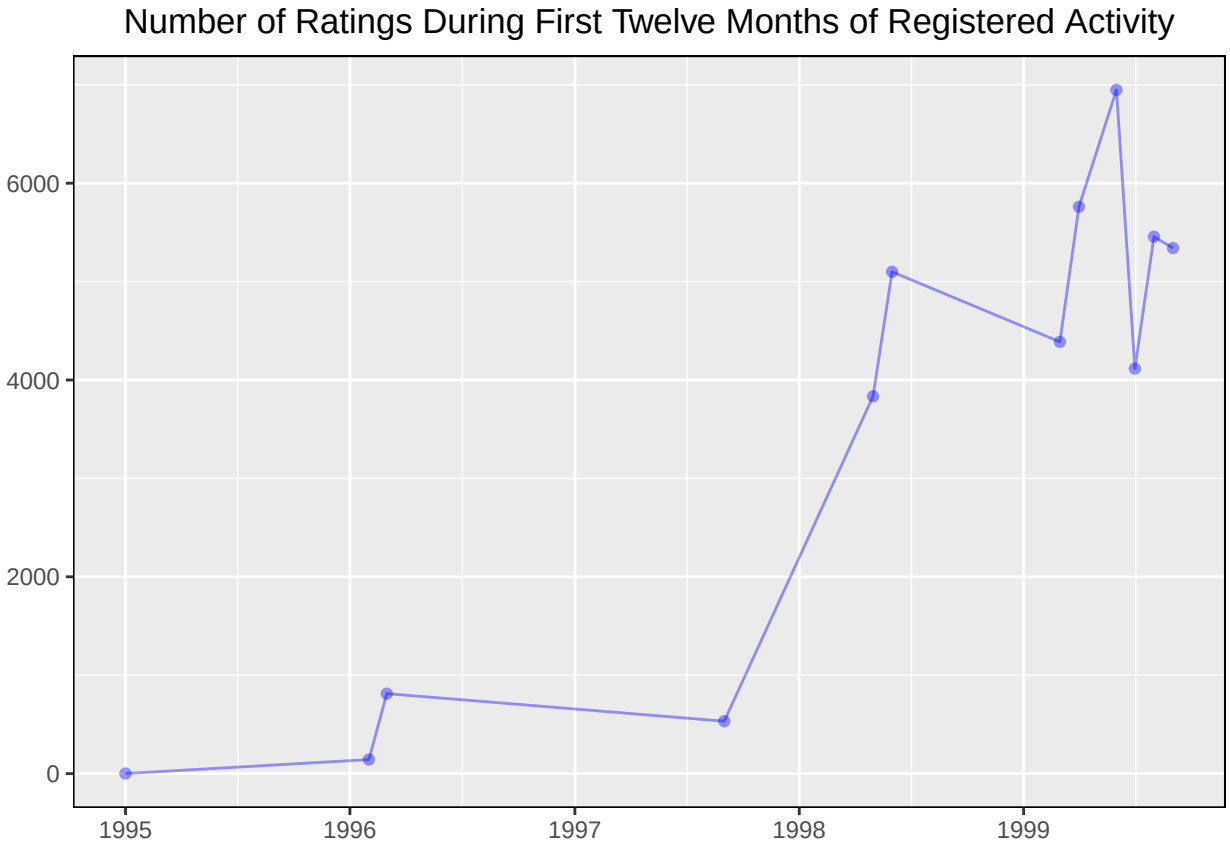


6. The Case For Regularization

The influence exerted by the features at our disposal is there. Therefore, it can be inferred that if we succeed in calculating how much each item pushes the average rating one way or the other, we might be able to build a good recommendation system. A problem, however, arises; one that can be observed in the final plot - *Average Rating per Date (Monthly)* - of Section 5 .

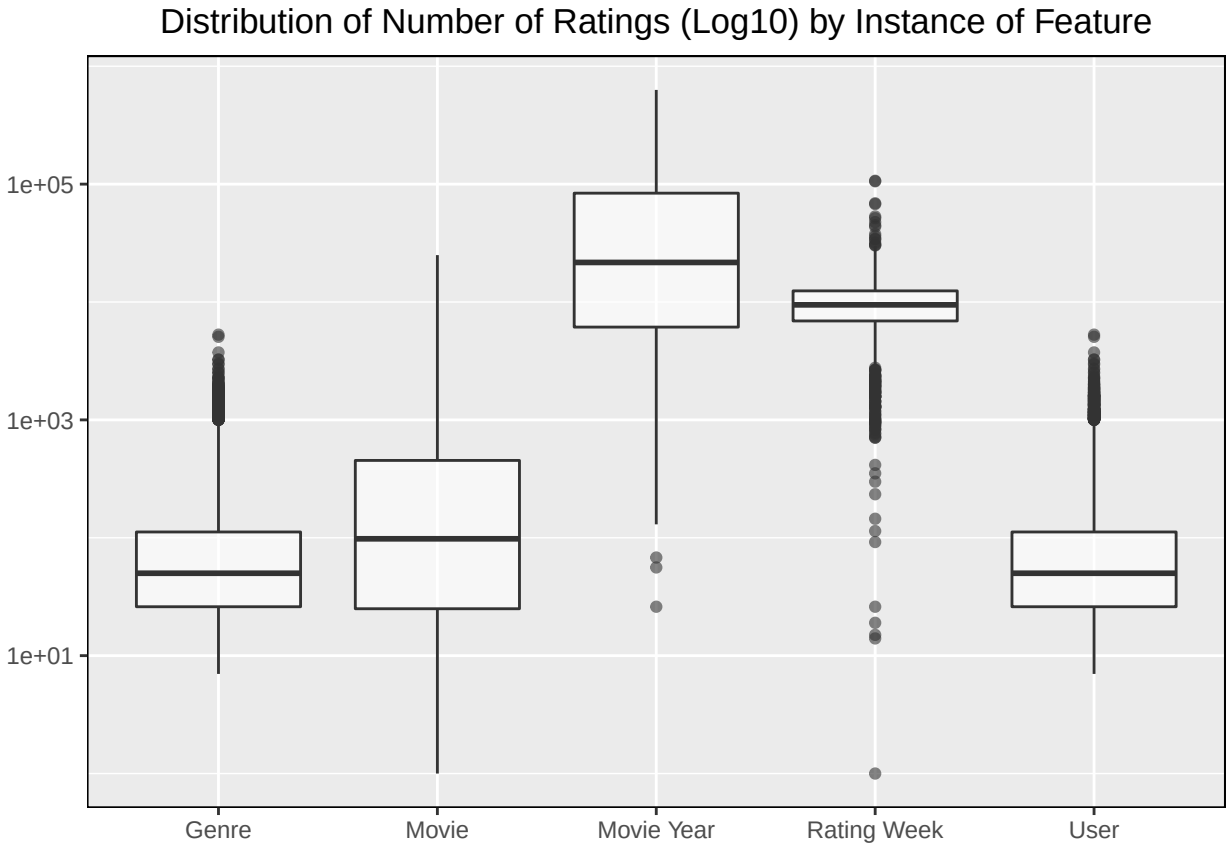
Although most of the average ratings over there are close to the center, a couple lie either far above or below the others. This tendency can be particularly observed during the opening months of the platform. What could be at play? Perhaps those were moments when ratings were not numerous; the website was, after all, in its early days and, consequently, the few ratings that were posted had a huge influence over the numbers.

In fact, the plot below, which shows the number of ratings recorded during the platforms's first twelve months of registered activity confirms that tendency. During the first month (January, 1995), only one rating was posted, and during many of the months that followed, no new activity happened. It was not until May of 1998 that the pace picked up, with that month seeing 3,834 new ratings posted: a lot more than Movielens had seen before it, but still very far away from the total global monthly average of 45,860 ratings recorded.

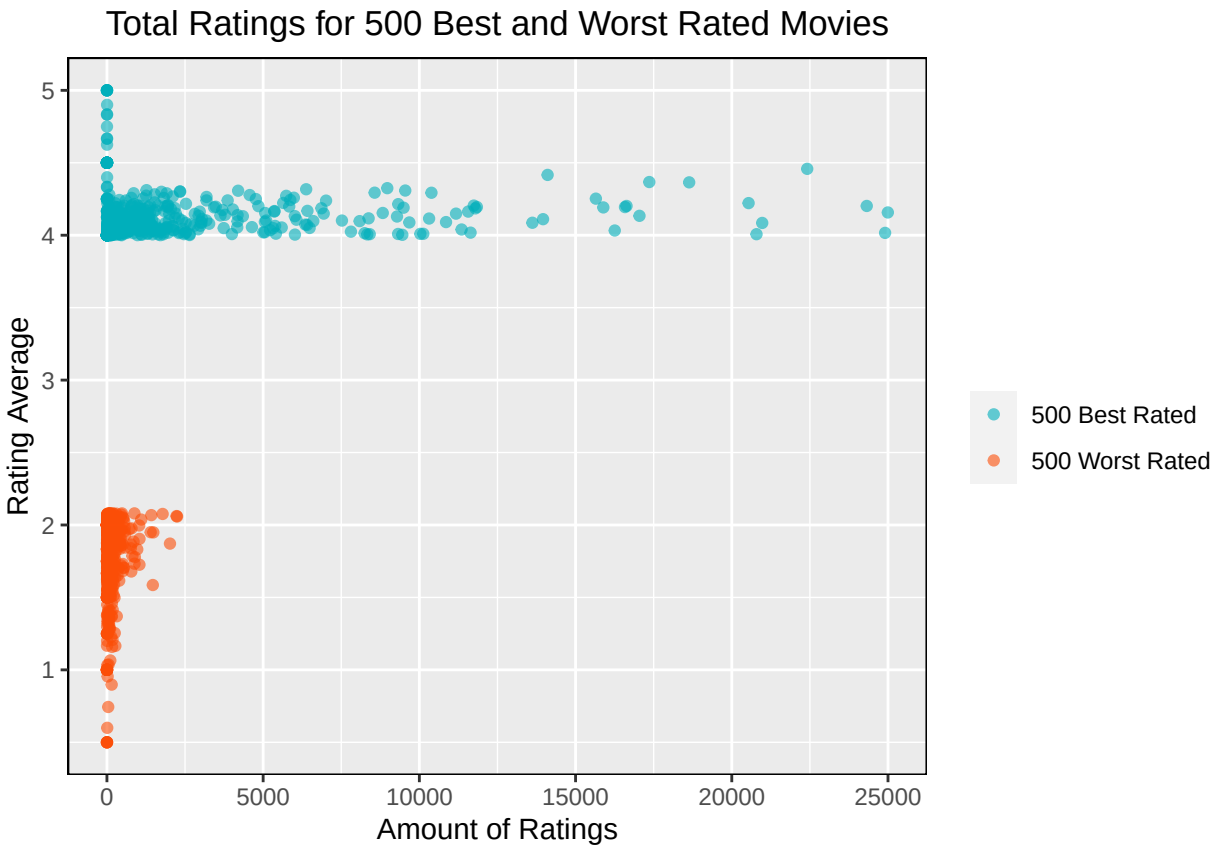


Because of that, during periods with a lower amount of ratings, the few that were registered had an enormous impact on the average. As such, if we were to calculate the influence the date of the rating has over the final score, time spans without too many records would get in the way of correctly assessing that relationship.

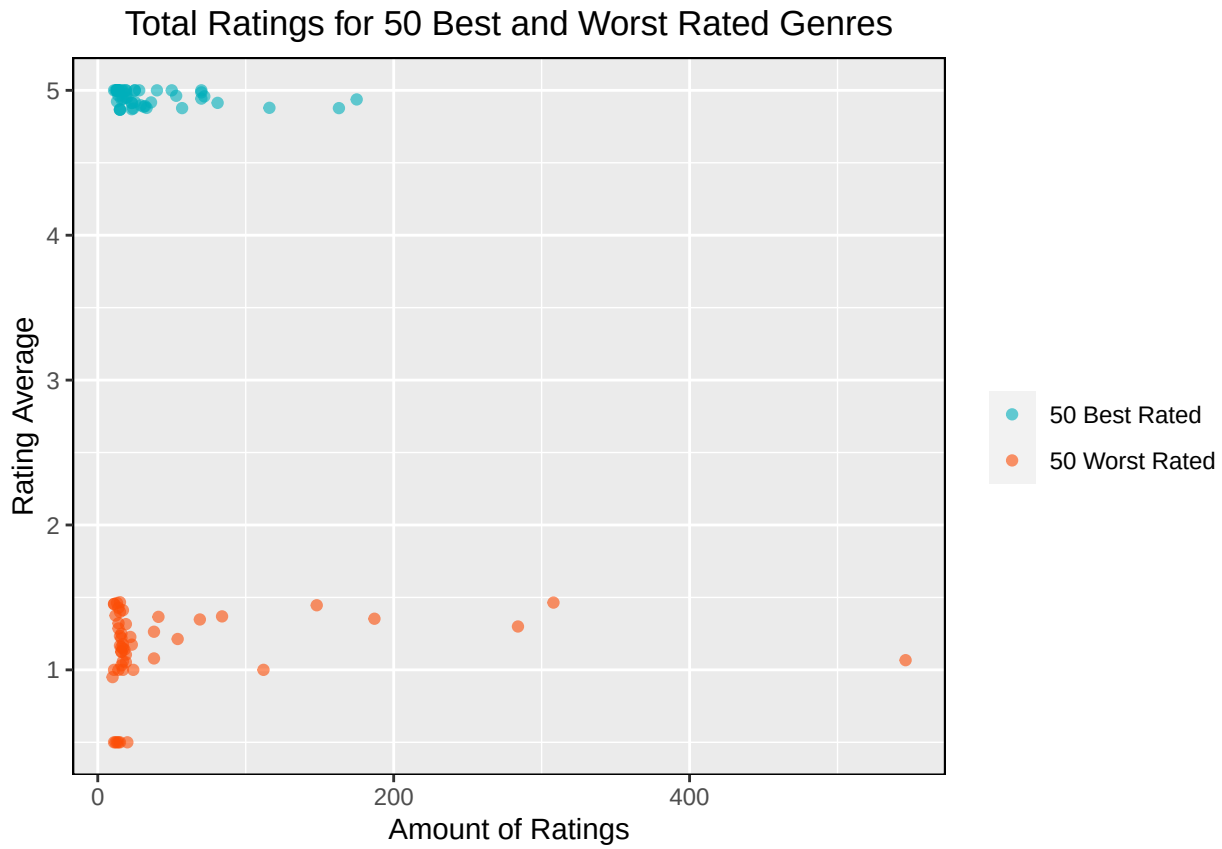
A similar phenomenon, as demonstrated in the plot below, happens throughout the variables we plan to use to predict the score. There are genres, movies, release years, weeks, and users that have an amount of ratings (either average or high) that allows the effect they have over the score to be estimated correctly, but all features also have some instances with low rating quantities, and these are bound to not provide very good insights into the effects they have on the score.



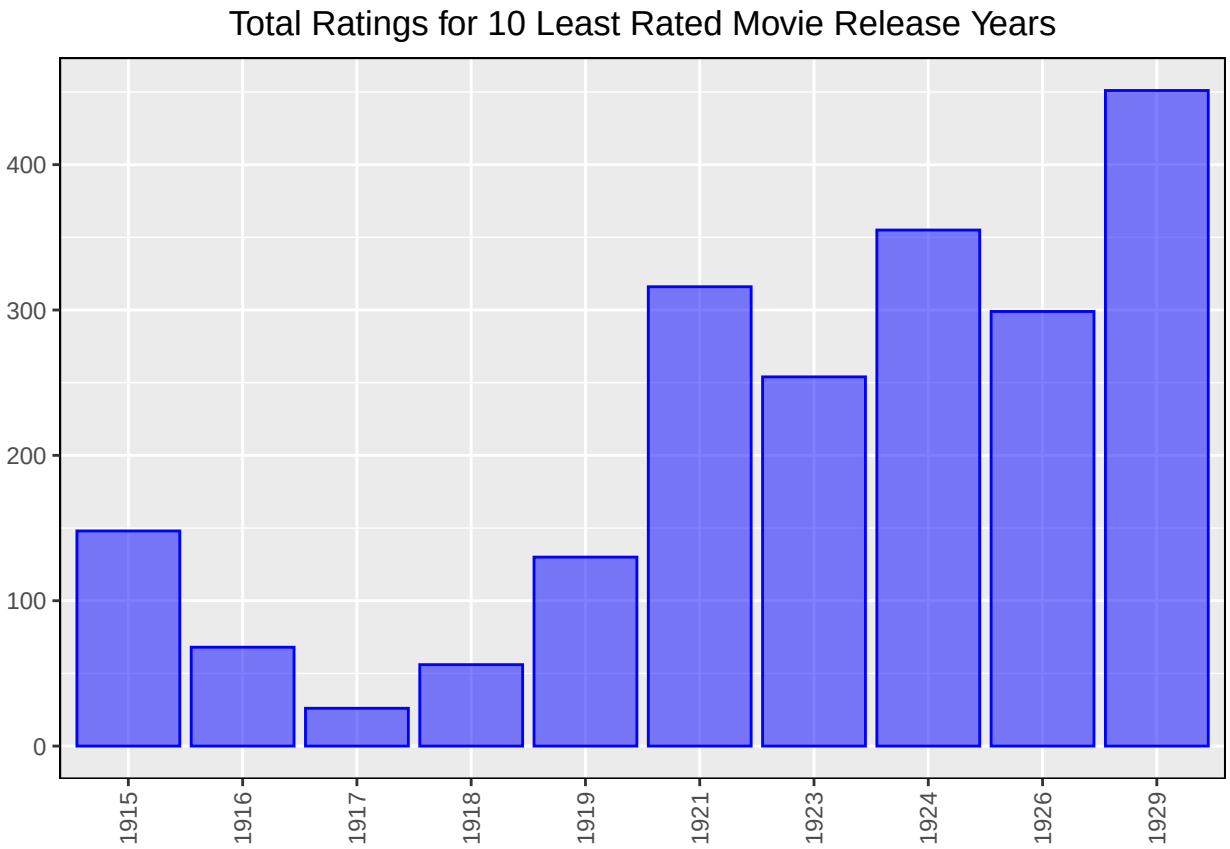
The plot below, which groups together the 500 Best and Worst rated movies also shows that, sometimes, the average score of flicks cannot be trusted. Most of the dataset's films with the highest and lowest scores are also the ones that have received very small amounts of evaluations. In these cases, the fact they have gotten little to no attention means that if we were to measure how these movies influenced their ratings, we would probably walk away with numbers that do not correspond to the truth.



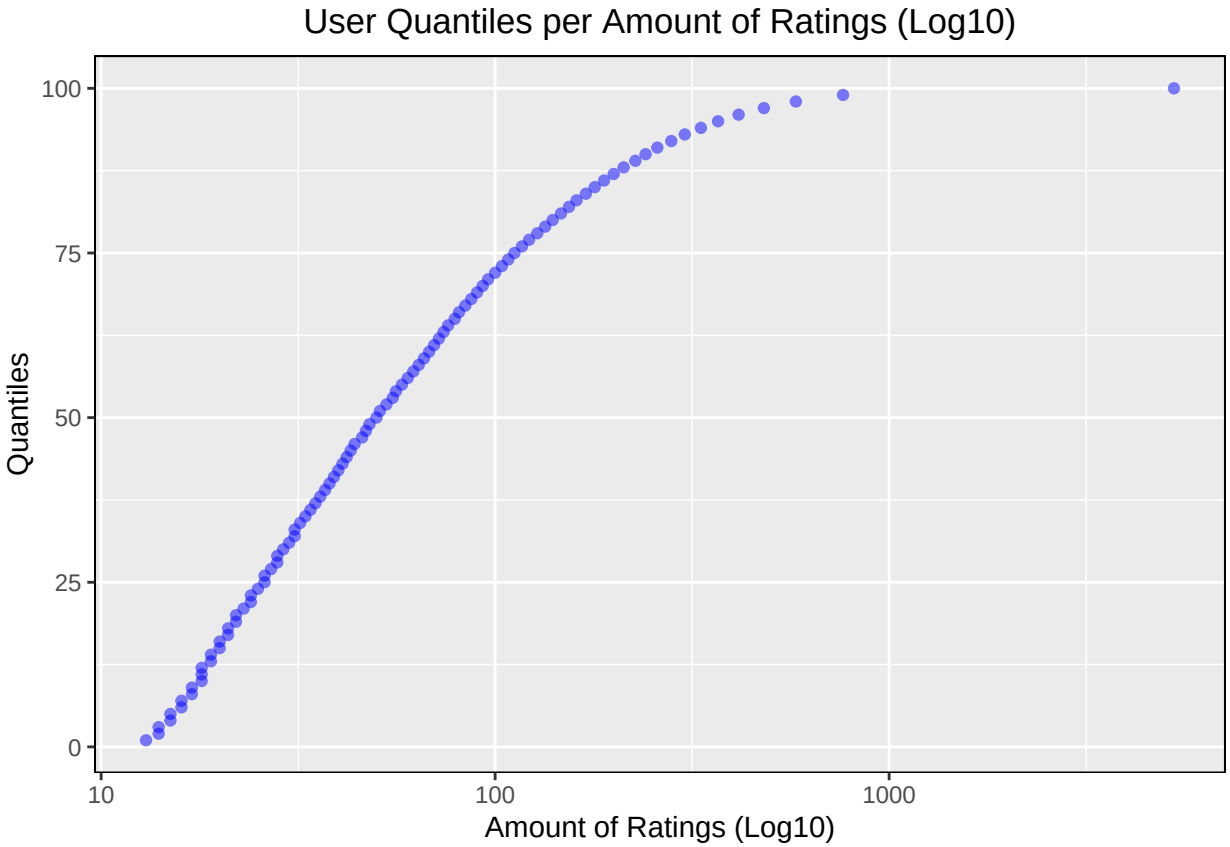
The exact same pattern appears when the genre is considered. This time, by selecting the 50 best and worst rated in the dataset, it is possible to observe movies belonging to them have received a very small amount of ratings. Therefore, trusting that those genres influenced the given rating, be it too positively or negatively, can lead to incorrect predictions. In fact, the effect the low amount of ratings has on these genres is so strong that many of their averages are close to the edges of the scale, 0.5 and 1.



Next, we take a look at the years with the smallest amounts of ratings. Movies from the 1910s and the 1920s tend not to have many evaluations. Can the effect they have on the score, therefore, be trusted?



As far as users go, the next plot shows just how many of them have low amounts of ratings. 15% of them have rated 20 movies or less; 50% have evaluated 50 movies or less; and 71% have not reached the 100-rating mark. Because of that distribution, it is clear the amount of scores they have awarded needs to be taken into account when trying to extract the bias each user has. Perhaps, some users have really low rating averages not because they are rigid evaluators, but because they have not been too active. Similarly, a user whose rating average appears to indicate a more positive outlook towards movies may only have rated the few films he or she loves.



With these facts demonstrated about all of the attributes that will be used to predict the score on the validation set, it is clear that the frequency of the ratings for each feature needs to be taken into account so that the bias we extract from them is not heavily influenced by a few outliers. In other words, movies that have a few ratings cannot be considered very good or very bad just because their averages indicate that could be the case; users cannot be viewed as too strict or forgiving if they are not too active; genres cannot be seen too positively or negatively if movies belonging to them have not been rated frequently; the year when a movie was released cannot upgrade or diminish its predicted score too much if the year in question has not been heavily evaluated; and, finally, the average rating of the week when the score was given cannot play a major role at determining the final score when it wasn't a very active one.

To deal with those effects, regularization comes into play. There are numerous regularization techniques [4], but the one employed in this work is given by the formula below. Our goal will be to obtain the value of b, which is the influence each feature has over the score. The variable y represents the ratings that possess a certain feature, be it those given to a specific movie (for instance, Pulp Fiction), to films of a certain genre (those that are labeled as dramas, for example), and etc. Those ratings are all subtracted by the overall dataset average $\hat{u}$, and the result obtained from that will intuitively tell us how influential such a feature is. These ratings are averaged with ni; that is, how often ratings with a such a feature appear in the dataset. Regularization comes in via the parameter *lambda*: the higher it is set, the more it will affect how the average is calculated.

$$b_i(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i} y_{u,i} - \hat{\mathrm{u}}$$

Because of that, ratings of a kind that is frequent (that is, those given to a specific movie, by a certain user, on a precise date, and etc) will not have their averages heavily hit by regularization: their n is already pretty big. Averages coming from ratings of a rarer kind, however, like those awarded to an obscure movie or to

films of a year that is too far into the past, will be hit by regularization and brought to a more reasonable threshold.

7. The Experiment

In the experiment, a total of ten models were built. With the exception of one, all of them tried to capture either only the effect of one specific feature or the combined effect of multiple features, hence working like ensemble methods. Those effects were calculated on the training dataset, composed of 7,200,043 ratings, and then evaluated (through rating predictions) on the test dataset, composed of 1,799,967 rows. The nine models that attempted to extract the effect features have over the score were all applied using regularization, with the parameter *lambda* varying between 0 and 30 in increments of 0.25. Consequently, a total of 1,090 models were constructed.

The best one was then selected and applied on the validation set provided by HarvardX. The table below summarizes the experiments conducted.

Model Name	Regularization Parameter	Total Models
Average Model	-	1
Movie Model	0-30 (0.25 increments)	121
User Model	0-30 (0.25 increments)	121
Genre Model	0-30 (0.25 increments)	121
Movie Year Model	0-30 (0.25 increments)	121
Rating Week Model	0-30 (0.25 increments)	121
Movie+User Model	0-30 (0.25 increments)	121
Movie+User+Genre Model	0-30 (0.25 increments)	121
Movie+User+Genre+Movie Year Model	0-30 (0.25 increments)	121
Movie+User+Genre+Movie Year+Rating Week Model	0-30 (0.25 increments)	121

The average model was built in the simplest possible way: by calculating the total average of the training set and then predicting all ratings of the test set to have that value. All the others were built according to the regularization formula presented in Section 6, and the predicted score on the test set was given by the average observed on the training set plus the effects calculated by the model. For example, in the Movie Model, predicted ratings were the effect of the movie added to the average; in the Movie + User Model, they were the effect of the movie and the user added to the average, and so forth.

Three comments are important to further clarify the experiment.

Firstly, not all model combinations were tested. For instance, there is no Movie+Genre model. This was done simply because the approach of adding effects to the calculation rather than shuffling them provided results that were satisfactory enough.

Secondly, the calculations for the effects of the models that take into account more than one feature were done already considering the previously extracted effects. In other words, in the Movie + User Effect Model, for example, the movie effect had already been extracted when the user effect was estimated. As such, for this calculation, rather than averaging the ratings subtracted by the total data set average, as it is shown in the formula of Section 5, the movie effect was considered. Therefore, the user effect was obtained via the average of the rating minus the movie effect of the film in question minus the total dataset average.

Such strategy can be seen in the formula below, where b represents the previously extracted effects. Consequently, for the most complex model of the experiment (Movie + User + Genre + Movie Year + Rating Week Effect Model), when obtaining the rating reek effect, b was equal to the movie effect, the user effect, the genre effect, and the movie year effect.

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

Finally, and still on that point, the order in which the effects were extracted could have been any, and - once again - not all combinations were tested. The order selected, however, had to do with the results of the models themselves. Before testing the ensemble strategies, all models of individual effects had been tested, and the Movie Model had shown to be better than the User Model, meaning the movie effect was more significant. Therefore, when combining strategies, the movie effect was the first one to be calculated and extracted from the rating. Such logic is true for all the ensembles: the Genre Model proved to be better than the Movie Year Model, so it was extracted first when combined, and the Movie Year Model was stronger than the Rating Week Model.

The summarized results of the experiment can be seen in the table below. In it, for each model, the best and worst results (in RMSE) are reported (along with the lambda parameter that generated them). The total average for the 121 experiments with that model is also shown. The exception, of course, is the Average Model, which had only ore result and, therefore, has neither an average nor a worst RMSE included.

To allow for better visualization, the model names were abbreviated as listed below:

- Average Model - AM
- Movie Model - MM
- User Model - UM
- Genre Model - GM
- Movie Year Model - YM
- Rating Week Model - RM
- Movie + User Model - MUM
- Movie + User + Genre Model - MUGM
- Movie + User + Genre + Movie Year Model - MUGYM
- Movie + User + Genre + Movie Year + Rating Week Model - MUGYRM

The rows are ordered according to the best result. In other words, the best performance in the test set was achieved by the Movie + User + Genre + Movie Year + Rating Week Effect Model, the one that took into account the effects of all considered features, with the value of *lambda* set to 5.

Model Name	Best RMSE	Best Alpha	Worst RMSE	Worst Alpha	Average
MUGYRM	0.865243215030013	5	0.868974739995228	30	0.866580155887523
MUGYM	0.86548502338188	4.5	0.869535540543643	30	0.866958043723168
MUGM	0.865622735398832	4.75	0.869621236935002	30	0.867068896280895
MUM	0.865910575032523	4.75	0.86996254558533	30	0.867369090551939
MM	0.944018836324081	1.75	0.945125378579896	30	0.944475067723843
UM	0.978177742077132	5.5	0.981248391059729	30	0.979253292272836
GM	1.01817670993685	1.5	1.01820606512066	30	1.01818852377389
YM	1.04944643219089	30	1.04944711165278	0	1.04944672709651
RM	1.0565684248832	30	1.05656995605808	0	1.05656866165435
AM	1.06028923354172	N/A	N/A	N/A	N/A

Two important observations can be made about the results seen in the table. The first is that, for all models, the worst results were achieved when the regularization parameter was either 0, which means no regularization was actually done, or 30, the maximum value tested. Although it is hard to tell what would have happened if values above 30 were tried, it is possible to conclude regularization works, but pushing its parameter to elevated degrees may not yield positive results. It seems the ideal point, at least for this dataset, is somewhere in between.

Secondly, all models confirmed the intuition that the features analyzed play a role when it comes to the rating. Models that took their effect into consideration were uniformly able to improve on the result achieved by the Average Model. Even the worst results for the weakest of the strategies, the Rating Week Effect Model with *lambda* set to 0, surpassed our baseline.

8. The Final Result

With tests concluded and our best model chosen according to the results observed in the test dataset, the champion of our experiment was then applied to the validation set provided by HarvardX.

The Movie + User + Genre + Movie Year + Rating Week Effect Model had its *lambda* set to 5, and built the estimate of the effects on the full training set; that is, the training set itself plus the test set that was created before the experiments began. With the calculations completed, it was executed over the validation set, whose ratings were predicted considering the effects of movies, users, genres, release years, and rating weeks plus the total average of the full training set.

The final RMSE obtained was **0.8640543**, being therefore lower than the expected result of **0.86490**.

9. The Conclusion

In building a model to recommend movies to users based on their preferences as exposed in the Movielens dataset, our challenge was to accurately predict what rating they would give to certain movies if they watched them. To do so, the features provided in the set were explored. At first, intuition said that a few of them played considerable roles in dictating what the ratings would be, such as the movie itself, the rating tendencies of the user, and the genre to which the film belonged. Furthermore, a couple of other traits, namely the year when the movies were released as well as the week when the ratings were given, also seemed like possible influential factors, even if to a smaller level.

Through data exploration, much of those intuitions were confirmed. Furthermore, it was observed that features to which few ratings were attached, such as obscure movies, inactive users, and years without too many rated films, could lead to biased evaluations of how much they affected the final score; after all, they did not have enough ratings to provide a stable estimate. To counter those potential negative points, the models were built with a regularization strategy in mind to diminish the weight of these infrequently rated items.

After settling on that strategy, models were built to extract - out of the training set - the effect movies, users, genres, release years, and rating weeks had over the scores awarded. The process was done by, essentially, estimating how much over or under the average each of those items drove the scores given to them. The evaluation of the models over a test set proved the effects existed and were accurately captured: all models that took into consideration the effect of those features did better than the employed baseline, a model that predicted the average rating to all movies.

To an extent, intuition was again confirmed. The model taking into account the effect movies have over their ratings performed better than the one that calculated the user bias; meanwhile, that one outperformed the model built based on genres, which in turn outdid the model centered around the year of the movie's release, which was better than the one that took into consideration the week when the rating happened.

By combining the calculated effects, ensemble models were built, and the one that aggregated all of them, called the Movie + User + Genre + Movie Year + Rating Week Effect Model, brought the best results in

the test step. Finally, its application over the validation set, held out from any analysis until that point, confirmed the good expectations by having a better performance than the best threshold proposed by the HarvardX course.

This work, therefore, not only highlights the importance of regularization, but also shows that full analysis and consideration of all features at one's disposals can be the path towards building the best possible models. Not all combinations of effects were tried, not all possible time periods were evaluated; and the regularization parameter was only tested within a limited, yet large, threshold. Still, the chosen model brought good results and proved to be a solid recommendation system.

References

1. Movielens - Website
2. Movielens - Datasets
3. Towards Data Science - What does RMSE really mean?
4. Towards Data Science - Regularization in Machine Learning