Movie Recomendation System. A model to predict movie rating

J.Ibarguen

05/08/2020

Contents

1 Introduction

In 2006 Netflix launched a challenge to beat its movie recommendation algorithm by at least 10% of its Root Mean Squared Error (RMSE). This was achieved in 2009 when the BellKor's Pragmatic Chaos surpass the 10% barrier. Since then, the business of movie streaming has expanded greatly, with multiple new emerging platforms, all of which offer thousands of films to their clients. In this new era of streaming platforms offering enormous amount of movie and series, recommendation mechanism become a key stone for the fidelization of clients.

In this scenario, new and more sophisticated movie recommendation algorithms have been developed to improve user experience. The current project aims at humbly contributing to this task by providing a model able to improve a RMSE of 0.8649.

The data used was extracted from http://files.grouplens.org/datasets/movielens/ml-10m.zip and contained 10,000,054 ratings for 10,677 movies with 69,878 users. During this report we referred to *data set* when we speak of the group of data tables. Data table is a single data structure composed of rows and columns which we referred indistinctly as *data* or *data table*. The extracted data was divided into the raw data set, composed of the **Training** (a.k.a *edx*) and **Hold-out** (a.k.a *validation*) data tables to follow a Supervised Learning approach.

The model aimed at the prediction of movie ratings and incorporated five predictors, each of them defined the deviations form the overall average rating produced by the influence of the independent variables. These predictors were further regularized and tuned using bootstrap solutions over the training data set. This allowed to reduce over-training and produce a more smoothed model to fit the hold-out data. The model was mathematically reduced to a linear model consisting on the summation of the average rating (outcome) and its identified effects (predictors).

The final solution to predict the outcome rating reached a RMSE of 0.8631 over the hold-out set, improving in 0.0018 the set objective of 0.8649.

2 Methodology and Analysis

The methodology and analysis section included a first step divided the data ensuring it is split in a manner that allowed relevant training and cross-validation. A second step consisted on exploring the data and the relationships between the independent variables and the outcome (ratings) to guide identification and composition of predictors. Finally, the third step was the description of the modeling approach: definition of predictors and its relationship in the model.

2.1 Data Splitting

The raw data set consisted of a Training data table of 9,000,055 ratings and a Hold-out of 999,999. The training data was further divided through a bootstrap solution with five re-samplings for cross-validation. The bootstrapping produced five data sets each composed of a Training and **Probe** data tables. Each Probe data contained 15% of all ratings.

The bootstrap solution allowed to test over the Probe data without using the Hold-out data, and furthermore it helped to avoid over-training over a unique Probe. As a result of the bootstrap the five re-sampled training data sets contained data of similar characteristics to the original Training data

Although overall the data characteristic is kept between all data sets, exist some slight differences between them, especially in average number of ratings per user, number of movies and average number of rating per movie that prevented to develop an over-training model to unique specific characteristics.

The following piece of code created divided the data through bootstrapping:

Table 1: Basic description of data sets

data	n. ratings	avg. rating	n. Users	avg. ratings/user	n. movies	avg. ratings/movie	avg. genres/movie	perc. commedy	perc. drama
Raw-Hold-out	999,999	3.51	68,534	11.73	9,809	180.42	2.6	0.39	0.43
Raw-Training	9,000,055	3.51	69,878	97.38	10,677	1,611.83	2.6	0.39	0.43
Boots-Training	7,650,044	3.51	69,878	82.95	10,664	1,370.18	2.6	0.39	0.43
Boots-Training	7,650,044	3.51	69,878	82.98	10,646	1,369.80	2.6	0.39	0.43
Boots-Training	7,650,044	3.51	69,878	82.98	10,649	1,370.94	2.6	0.39	0.43
Boots-Training	7,650,044	3.51	69,878	82.91	10,655	1,371.61	2.6	0.39	0.43
Boots-Training	7,650,044	3.51	69,878	82.92	10,650	1,370.09	2.6	0.39	0.43
Boots-Probe	1,349,995	3.51	69,388	15.42	9,994	242.62	2.6	0.39	0.43
Boots-Probe	1,349,972	3.51	69,372	15.39	10,006	243.01	2.6	0.39	0.43
Boots-Probe	1,349,977	3.51	69,357	15.39	10,010	241.93	2.6	0.39	0.43
Boots-Probe	1,349,983	3.51	69,388	15.46	10,013	241.21	2.6	0.39	0.43
Boots-Probe	1,349,976	3.51	69,396	15.45	9,981	242.77	2.6	0.39	0.43

```r
## Bootstrap x5 for cross-validation
set.seed(28032020)
bootstrap <- caret::createResample(edx$rating, times = 5, list = TRUE)

## Partition 15% probe
set.seed(28032020)
probe_i <- lapply(
  1:length(bootstrap), function(x)
  caret::createDataPartition(bootstrap[[x]], p = 0.15, list = TRUE)
)

## Subset partitions from edx and format "Timestamp"
probe_boots <- lapply(1:length(probe_i) , function(i) edx[probe_i[[i]][[1]], ])
train_boots <- lapply(1:length(probe_i) , function(i) edx[-probe_i[[i]][[1]], ])

## match probe to train to remove NA in summaries
probe_boots <- purrr::map2(probe_boots, train_boots, ~ {
  .x %>%
    dplyr::semi_join(.y, by = "movieId") %>%
    dplyr::semi_join(.y, by = "userId") %>%
    dplyr::semi_join(.y, by = "timestamp")
})
```

2.2 Data Exploration and Analysis

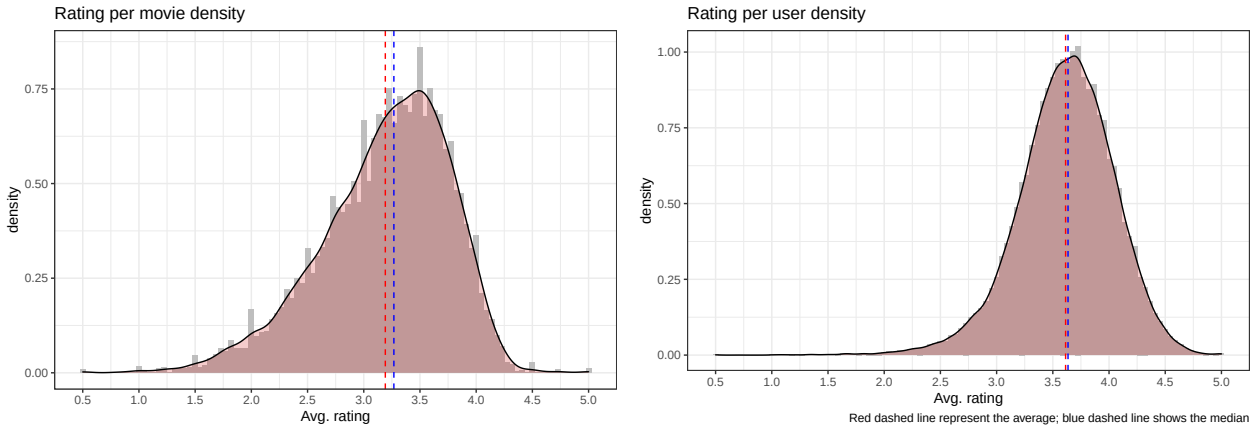
The overall average rating (from no on called indistinctly as μ or *overall average rating*) of the original training data (*edx*) was of 3.51. However, this average rating changed in relationship with other factors. Among them we studied: 1) the movies, 2) the users, 3) the genres, 4) the time (day) and 5) the number of rates given.

2.2.1 Ratings of Movies and Users

Some movies are more popular than other while some users tend to be more critic than others. These differences would reflect on differences of movie and user averages rating when compared to the overall average rating.

As shown in Figure 1a, the distribution of average rating per movie was skewed to the left. Ratings progressively accumulated from 0 to 3.5, and from there fell abruptly. Both, mean and median are below μ suggesting a negative impact of movie titles over the overall average. This could be cause popular movies that get higher rates are not that common.

Otherwise, Figure 1b show the user's average rating followed an assumed normal distribution centered at 3.61. Contrarily to Movie effect, the center of User effect is greater than μ suggesting an overall positive impact on μ. An explanation for this is that the average user might not be specially critic on its rating; critic users are not the norm.



(a) Rating per movie density

(b) Rating per user density

Figure 1: Average Rating per Moview and User (density)

2.2.2 Ratings of Genres

Movies and users shown certain effect over the overall average rating. Similarly genres could also have an effect on overall average rating. Genres allow to classify groups of movies based on categories of superior order which would be linked to user preference: a user would have preference for a specific genre rating high movies of one genre than other, thus deviating from μ.

Acknowledging a single movie can have multiple genres, we still observed the genres independently as to if the movie was classified in that genre (included) or not (excluded). In Figure 2 we observe that only Drama genre seemed to have a positive effect on μ, for movies classified in it have 0.16 rating points than the overall average rating. Conversely movies that are not classified in Drama seem to have a lower average rating (3.39). Otherwise Horror and Sci-Fi genres showed a lower average rating when movies were classified in them, worth noting Horror movies have an average rating of 3.27. Up to less extend yet following the same trend as Horror and Sci-Fi we observed Action and Comedy.

2.2.3 Ratings by Time (Days)

Time was explored at day level as this unit is more likely to capture the user variability on rating than other time units. To explore at day level, the time stamp was converted to date format and round to the nearest day. Furthermore, we cleaned the two only films that were rated in 1995. These were removed from the exploratory analysis to avoid biasing the conclusions.

In Figure 3, we can observe how movies before April 1996 had an average 3.95, significantly higher than the overall average μ. Not as outstanding, we can find other periods of positive deviation from μ between October 1996 and 1997; and April 1999 and 2001. On the other side, between October 1997 and 1998 the deviation from μ turned negative, effect that repeated for the period of April 2002 up to October 2005, and resumed in April 2006. Later we will observe how these variations in seems to be strongly correlated with the number of rates given per day during these periods.

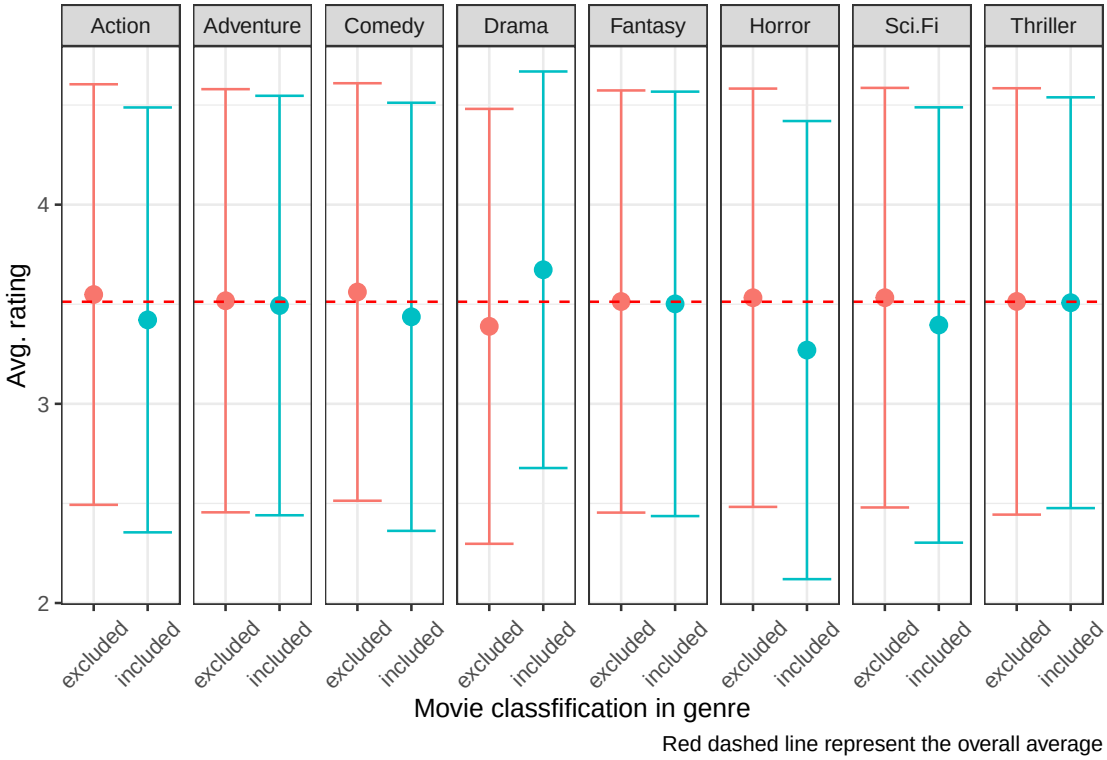


Figure 2: Average rating by movie classified in a genre or not (included/excluded)

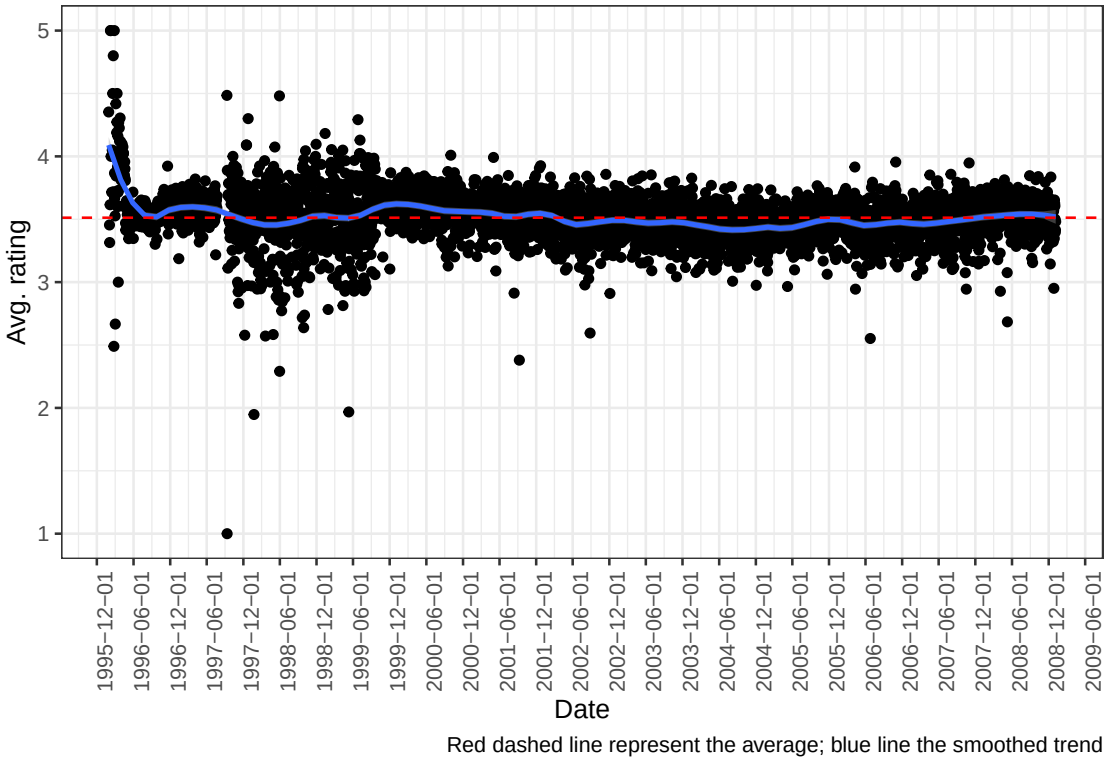
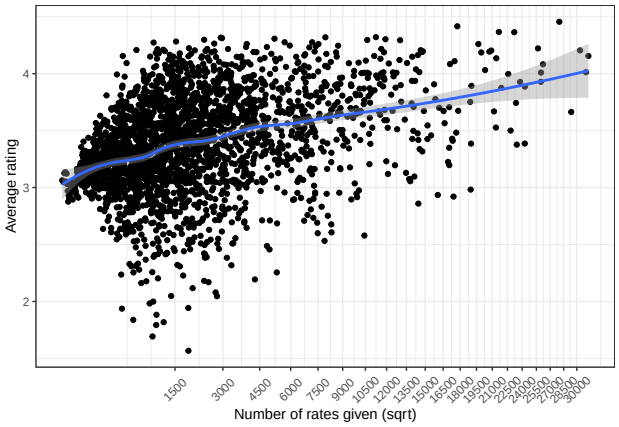


Figure 3: Average rating by number of rates given per day (1994 - 2009)

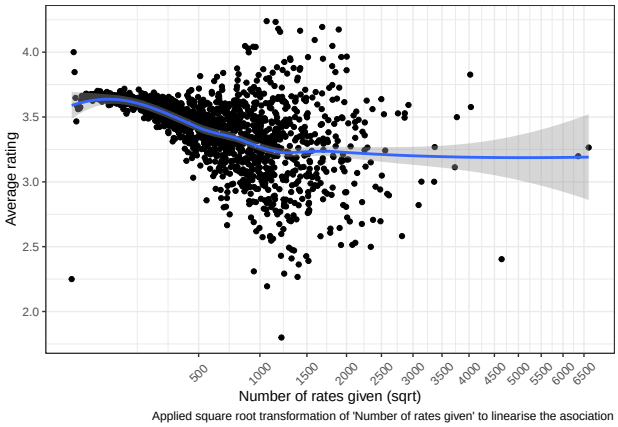
2.2.4 Ratings by Number of Rates Given

Up to now we have seen how average rating by movies, users, genres and time had deviated from the overall average of rating. Observing now the Number of rates given per movie, per user and per day, it was noted that Number of rates given could been an important confound contributing to the previous deviations. However, we did not note any relevant effect when observing Number of rates given per genre.

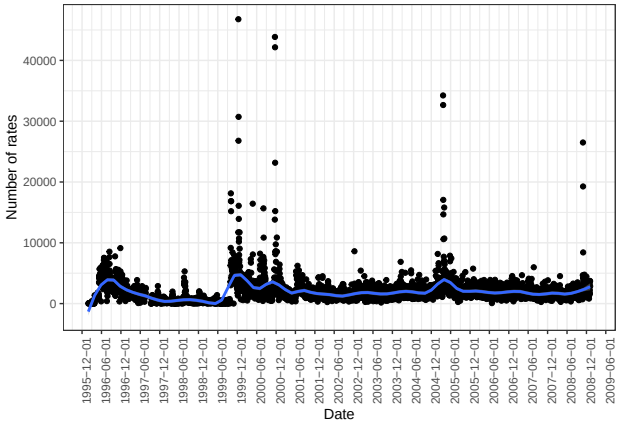
Starting with the effect of Number of rates given per movie, we observed a positive trend in which the more rates a movie has received, the higher the rating average for that movie. Conversely Number of rates given per user have a negative association with regards to user average. These relationships (Figure 4a and 4b) might easily be explained as popular movies, although not very common as we shown before, tend to receive greater number of rates and higher rating; at the same time more critic users, neither very common, tend to be more compromise and committed to provide severe feedback.



(a) Number of rates given per movie



(b) Number of rates given per user



(c) Number of rates given per day

Figure 4: Average rating by Number of rates given per movie, per user and per day

With regards to time, we observe that the periods of time when more rates were given (Figure 4c), mimic the periods of time when the rating deviates positively over μ (Figure 3). We could there assume higher rating on a given day was derived from a greater number of rates given that day. And this most likely could be related and contributed by the release of widely popular movies or specially prolific periods, however, we will not reach this last extend in this work.

2.3 Modelling approach

From the data exploration we observed that overall average rating μ was influenced by a series of independent variables, namely: Movie title, User, Genre, Time and Number of rates given. All these independent factors had an effect on the overall average, they deviate the rating from μ. Therefore, a effect b produced by a factor x was calculated for a given set of ratings ($|R|$) determined by that factor as the average difference between the rating for the factor (r_x) and the overall average rating μ:

$$b_x = \frac{\sum_{x \in R(x)} r_x - mu}{R(x)}$$

All in all we defined 5 effects or predictors that we will tackle in this study: Movie effect (b_m) User effect b_m, Genre effect (b_g), Time effect (b_t) and Number of rates per movie effect (b_n). All these predictors were compiled in a linear model showing the relationship between μ and the effects that determine it to predict the outcome rating (Y). The model was be notated as:

$$\bar{Y} = \mu + b_m + b_u + b_g + b_t + b_n$$

Although from the exploratory analysis we could have easily extract more effects, namely Number of rates per user effect and Number of rates per day effect, we will not tackle them in this study due to computational limitation and memory size challenges. However, as we will see later, we had strong reasons to think that this effects would reduce further the RMSE of our model and improve our prediction.

3 Results

We start out modeling from the most basic model we could draw, and which would serve as a baseline to guide if the inclusion of new predictors. The model consisted uniquely in the overall average rating:

$$\bar{Y} = \mu$$

This baseline model reported an RMSE of 1.06, quite far from our target of $RMSE < 0.8649$.

3.1 Movie Effect

The Movie effect reflect the deviation that the average rating for each movie has from the overall average rating. Having the baseline model, we proceed to incorporate our first identified predictor to account for the effect of movie titles over μ. The model was therefore notated as:

$$\bar{Y} = \mu + b_m$$

The RMSE for the model resulted as 0.9438, with a significant improvement of 0.1166 over the baseline model.

Despite the significant improvement, the predictor was not regularized increasing the risk of over-fitting. For regularizing the predictors, we set a λ parameter in its calculation which allowed to shrink averages towards 0 to produce a more smoothed estimate and reduce fitness to training data noise. This help to avoid over-training and would lie on better estimates when fitting the hold-out data.

Lambda was introduced into the calculations of the effects as follows:

$$b_m = \frac{\sum_{m \in R(m)} r_m - \mu}{\lambda + R(m)}$$

Producing a model that was notated as:

$$\bar{Y} = \mu + \lambda b_m$$

After tuning for λ parameter the best RMSE of 0.9437 was obtained when λ equals 2.25. Although very slightly regularization improved the reported RMSE. However, is its potential effect for reducing over-fitting and improving fitting over the hold-out data for which all predictors were regularized by this mean.

3.2 User Effect

The next predictor, the User effect defined the deviations from mu produced by each user. It was calculated and regularized in a similar fashion as the Movie effect. However, the new predictor, and from here the proceeding predictors, accounted for the accumulated effect of the previous predictor with the following formula:

$$b_u = \frac{\sum_{u \in R(u)} r_u - \mu - b_m}{\lambda + R(u)}$$

The new model was notated as:

$$\bar{Y} = \mu + \lambda b_m + \lambda b_u$$

The original RMSE without regularization was of 0.866. Regularization produce an improvement in RMSE of 5e-04 down to 0.8655. The best λ was 5. All in all the new model reduced the RMSE in 0.0782 with regards to the previous model.

3.3 Genre Effect

The next model incorporated the genre effect which accounted for the effect that genres typology had on the overall average. Note that for the purpose of modeling, and contrarily to the exploratory analysis, genre was taken as the combination of all genres assigned to a movie. This approach allowed to account for all the diversity that the genre's typology offer. Following the same path drafted for the previous predictors:

$$b_g = \frac{\sum_{g \in R(g)} r_g - \mu - b_m - b_u}{\lambda + R(g)}$$

Resulting in the following model:

$$\bar{Y} = \mu + \lambda b_m + \lambda b_u + \lambda b_g$$

The RMSE without regularization was 0.8652. Regularizing the predictor barely improved the RMSE of 0.8652 with the best λ of 13.5. All in all the improvement with regards to the previous model was of 3e-04

3.4 Time Effect

Time effect accounted for the deviations from μ produce by the average rating given for all movies and all user in one day. The time effect was equally incorporated to the model by calculating the predictor according to the following formula:

$$b_t = \frac{\sum_{t \in R(t)} r_t - \mu - b_m - b_u - b_g}{\lambda + R(t)}$$

The new model was notated as:

$$\bar{Y} = \mu + \lambda b_m + \lambda b_u + \lambda b_g + \lambda b_t$$

Without regularization, the RMSE was of 0.8646. With regularization the RMSE improved down to 0.8646, being best λ at 86. This implied an improvement with regard to the previous model of 6e-04.

With the incorporation of Time effect into the model, we reached our objective of producing a RMSE $<$ 0.8649. However, still we haven't included predictor for the effect of the Number of rates given

3.5 Number of Rates per Movie

We have previously seen that Number of rates given confounds with Movies, Users and time (day). This is the Number of rates given per Movie, User and Day all have effect on the overall average rate μ. However, noting that we already have achieved our objective with the previous model and given serious computational and memory size limitations to calculate and model all effects related to Number of rates given, here we will only show the effect for Number of Rates per Movie.

As usual, we include compute the predictor following the usual formula to account for accumulate effects of the previous models

$$b_n = \frac{\sum_{n \in R(n)} r_n - \mu - b_m - b_u - b_g - b_t}{\lambda + R(n)}$$

The accumulated model resulted in:

$$\bar{Y} = \mu + \lambda b_m + \lambda b_u + \lambda b_g + \lambda b_t + \lambda b_n$$

The model without regularization resulted in a RMSE of 0.8639. The regularization allowed to improve the RMSE by 0 with a λ parameter of 7.5. All in all, the model produced a RMSE of 0.8638, with an improvement of 8e-04 over the previous model.

Given the satisfactory results of the inclusion of the new predictor to account for the effect of the Number of rates given per Movie, we could easily infer that the further inclusion of the predictors Number of rates given by User and Number of rates given per day would have had similar and cummulative positive impact on reducing the RMSE of the model, thus better accounting for the deviation of the ratings from the overall average rating μ and improving the predictions.

3.6 Final Test

The model we have developed up to know have been improving the RMSE progressively as they incorporated new predictors to account for the effect of independent variables on overall average rating. As shown in Table 2 the cumulative improvement from our baseline model up to our last and final model was of 0. After the first two predictor that produced the greatest relative improvement, it is significant to point that Genre effect produced the lowest improvement (0).

We have also have seen how regularization technique allowed us to reduce even further the RMSE for all our models. However the real effect of regularization should be noted when fitting the model to the hold-out data.

Table 2: Summary of models

predictors	**rmse**	**relative.improve**	**cumm.improve**
Baseline	1.0604	0.0000	0.0000
Movie effect	0.9437	-0.1167	-0.1167
User effect	0.8655	-0.0782	-0.1949
Genre effect	0.8652	-0.0003	-0.1952
Time effect	0.8646	-0.0006	-0.1958
n. rates / Movie	0.8638	-0.0008	-0.1966

Our final model was then applied to the holdout data to predict rating.

$$\bar{Y} = \mu + \lambda b_m + \lambda b_u + \lambda b_g + \lambda b_t + \lambda b_n$$

The achieved RMSE was of 0.8631. This was 0.0018 below our target of 0.8649.

4 Conclusion

We approached the challenge of predicting rating for a movie recommendation system with a linear model purely based in the cumulative incorporation of predictor to account for deviations for the overall average rating. This approach allowed us to confront a big data set of 10 million rows with a personal computer and achieve our objective of RMSE < 0.8649.

Computational capacity has been the major challenge. As we have seen, our model did not account for all possible predictors and namely we have excluded two: Number of rates given by user and Number of rates given by day which would have reduced further the RMSE improving the predictions.

We have also seen that Genre effect have produced a very small improvement in the RMSE. Is likely genre would need other approached to account for. Given its complexity, as one movie could have multiple genres, it is likely factorization would have been a better approach. It therefore would be advisable in next works on the matter to attempt a factorization of the individual effects of each genre over rating given by each user.

Despite the challenges and limitations, we have seen however, that regularization is a very important technique to improve prediction over the final hold-out. Not only regularization have improved the RMSE over the training data set, but it has likely contributed to have a lower RMSE with the hold-out data than with the training data.

Given weaknesses and strengths on our approach to the challenge, we achieved successfully and with margin our objective of obtaining a RMSE below 0.8649.