HarvardX PH125.9x MovieLens Capstone Project

Daniel Middendorf

11/18/2020

1. Introduction

A recommendation system refers to an algorithm to predict the "rating" or "preference" of a user. In contemporary western societies, such methods are deeply integrated into the daily habits of most people. Our contact with such technologies becomes apparent when, e.g., the playlist we hear automatically plays a song we like, or our preferred movie streaming service suggests a movie that fits our interests. These systems even infiltrate areas of our lives that we would not consider being related to machine learning: the scientific articles we get recommended upon our searches are tuned to our research interests, and even the faces/profiles of potential partners on dating websites are adapted to our personal preferences. The past thirty years have seen increasingly rapid advances in machine learning, which allowed many technological advancements in recommendation systems. In 2006, Netflix offered one million dollars to a team able to improve its recommendation system, which was achieved by a team refining their predictive model over months. In the present paper, we will try to build a similar recommendation system. To quantify our success rate, we will try to minimize the root mean squared error (RMSE) between predicted ratings and original ratings to a level below .8649.

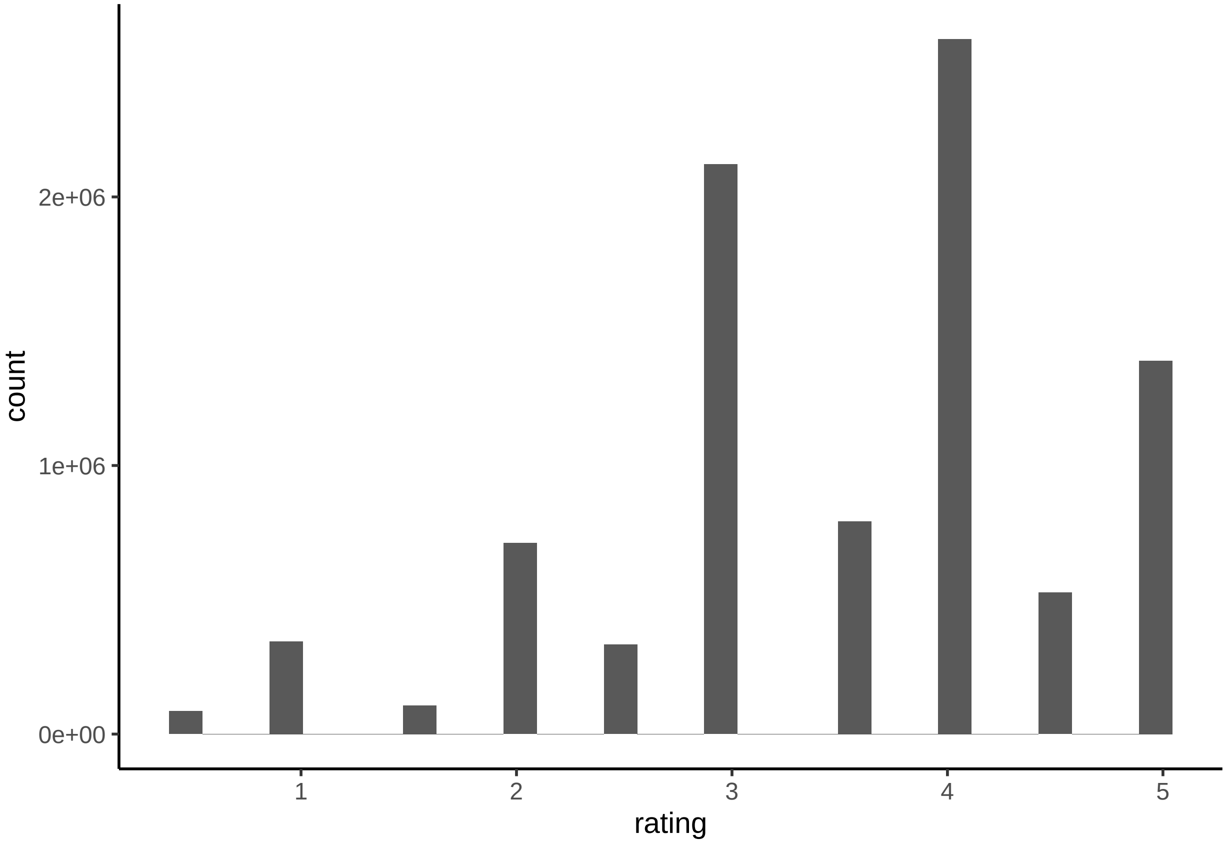
1.1 Description of the Dataset

The dataset we will use for our prediction includes 9000055 rows and six columns. Each row represents a rating given by a user on a movie. The scale for rating each movie ranges from 0 (very bad) to 5 (very good). In this dataset, 10677 movies were rated by 69878 different users. However, the matrix is very sparse as users only watched a part of the films and did not nearly give each movie a rating. The most rated movie genre was Drama and Comedy, letting us infer that such are also most popular. The most given rating is 4.

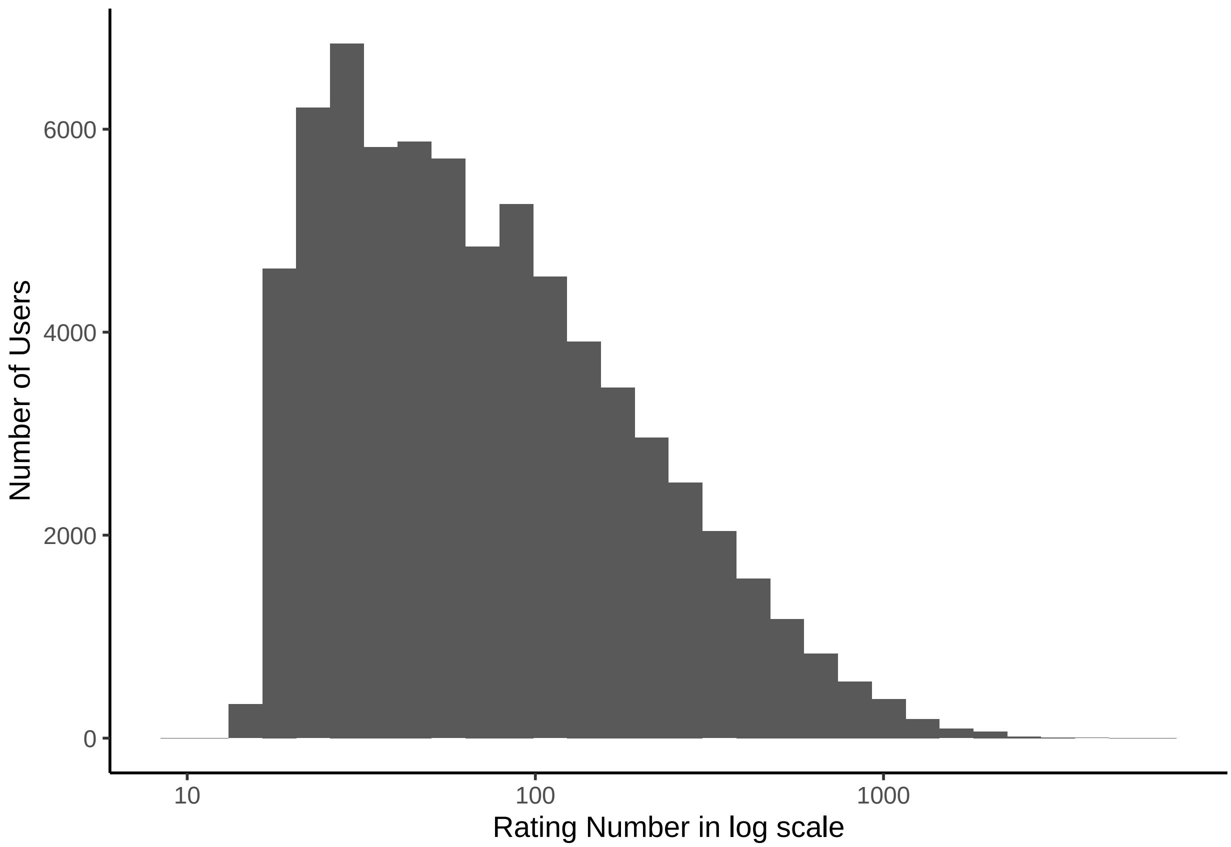
1.2. Visualization of Descriptive Statistics

In the following analysis, the prediction dataset was employed while data from the validation set was excluded. All users were anonymized and given a discrete user identification number.

The mean rating given was 3.512 while the most given rating was 4. From the following histogram the distribution of such ratings can be shown. Interestingly, users also preferred given ratings in whole numbers instead of half numbers.



Most ratings also come from a minority of users, which can be seen in the following histogram. This is important to know as this could bias the effect of movie ratings later employed in our final model. To illustrate this biasing effect: When only a minority of users give most of the ratings, those users' opinions are highly overrepresented. This could be especially problematic if the characteristics of this group do not represent the characteristics of the whole population that is aimed at.



2. Methods and Analysis

2.1 Methods

We employed various statistical methods to create a model that reduces the RMSE between predictions and the obtained data in the validation set. Firstly, we expected that the effect of movie ID plays an essential role in their rating as, from experience, some movies are generally rated higher. Secondly, we also assumed that a user's rating also depends on the user himself. Some users typically give higher ratings than others; hence, including user ID in our model is essential.

The first model would, therefore, looks like this:

$$Y_{u,i} = \mu + b_i + b_u + \varepsilon_{u,i}$$

Although this model seems useful in predicting actual ratings, the model seems not to be optimal. This model does not take into account the number of people who rated a movie. Hence, a movie rated by one person with 5 stars would be seen as a very good movie, even though if rated by other people the movie would have received a much lower rating. To correct for such a problem we used regularization to minimize such errors. We applied a penalty to movies with a low number of ratings. The size of the penalty, thereby, depends on the number of ratings, n. To minimize such errors we try to minimize:

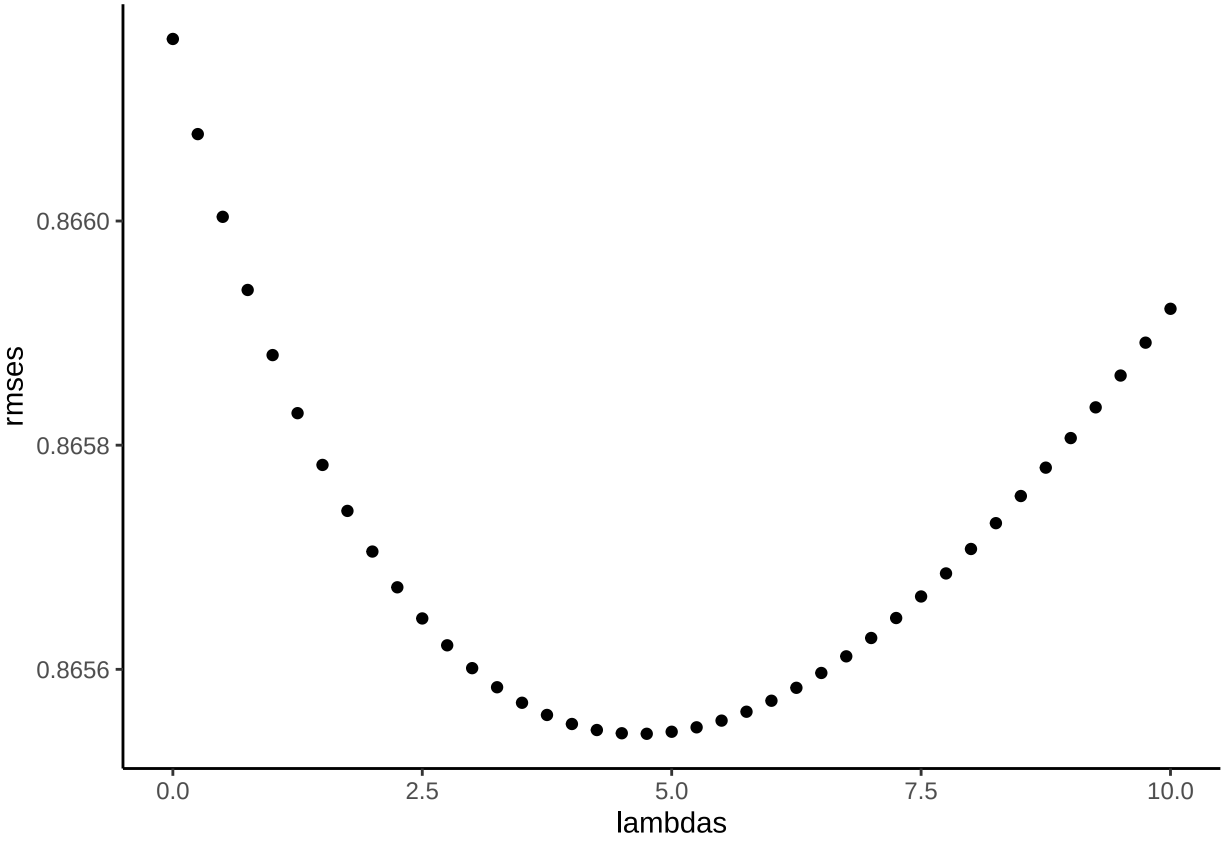
$$\sum_{u,i} \left(y_{u,i} - \mu - b_i - b_u\right)^2 + \lambda \left(\sum_i b_i^2 + \sum_u b_u^2\right)$$

2.2 Analysis

Firstly, we split the original dataset into two distinct datasets: one for training our model and adapting it and one for testing our predictions without consulting the validation dataset. The test dataset includes 20 percent of the original dataset and the train dataset uses 80 percent of the dataset respectively.

To find the best value for the penalty variable λ, we created a sequence of λs from 0 to 10 in .25 steps in order to control the total variability of the movie and user effects. This means that when the sample size is really large n + λ approximately equals n so that λ does not have a great influence. Only when n is very small λ will regulize this effect significantly and will finally shrink it towards 0.

The final model we employed utilizes a λ of 4.75. The results of the visualization of the effects on the RMSE can be shown in the following figure. As a λ of 4.75 produces the smallest effect in the test set, this λ value will also be used when evaluating our model on the validation set.



3. Results

As our project is aimed at creating a RMSE value that lies below .8649, we assessed the theoretical model created by the train and test set by the validation dataset. Although the theoreticl background of our final model was build and tested via the train and test set, we used the entire dataset that is available for building our model to increase the predictive power. The validation set was only used the last final step to calculate the RMSE.

After creating the predicted values for our testset and removing NA values, we achieved the following RMSE:

```
[1] 0.8648201
```

The final RMSE calculated via the validation dataset is lower than the aimed at RMSE.

4. Conclusion

The present project set out to create a statistical model that reduces the error (RMSE) between predictions and obtained ratings. In order to achieve this, we employed the effects of the average rating for each movie and the effect of the user ID. Both effects can be justified by the assumption that various movies get different ratings and that various users have a different rating pattern. Furthermore, we employed regularization techniques in order to balance out few user ratings on each given movie.

Our attempts at finding a model that reaches the threashold of a set RMSE was successful as our model effect lie below the cutoff RMSE:.8649. Hence, our endeavors can be evaluated as successful.

Limitations of the present project mainly concern the computing power of the used machine. Therefore, it was not possible to employ techniques like lm, glm, randomforests or knn as such were very slow in exploratory analyses; also the reason why they are not included in the code provided in the supplementary materials. Future technologies or a more sophisticated setup might solve these limitations.

5. Further Readings

Shah, D., Shah, P., Banerjee, A., & 2017 IEEE Region 10 Conference, TENCON 2017 2017 11 05 - 2017 11 08. (2017). Similarity based regularization for online matrix-factorization problem: an application to course recommender systems. Ieee Region 10 Annual International Conference, Proceedings/Tencon, 2017-december, 1874–1879. https://doi.org/10.1109/TENCON.2017.8228164

Hong, F.-X., Zheng, X.-L., & Chen, C.-C. (2016). Latent space regularization for recommender systems. Information Sciences, 360, 202–216. https://doi.org/10.1016/j.ins.2016.04.042

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. Physics Reports, 519(1), 1–49. https://doi.org/10.1016/j.physrep.2012.02.006

Burke, R., Felfernig, A., & Goker, M. H. (2011). Recommender systems: an overview. Ai Magazine, 32(3), 13–18. https://doi.org/10.1609/aimag.v32i3.2361