MovieLens

Michael Strenge - https://github.com/mistrenge/MovieLens.git (https://github.com/mistrenge/MovieLens.git)

15 5 2019

Introduction

The purpose of this project is to create a movie recommendation system using the MovieLens dataset. In general, recommendation systems use ratings that users have given items to make specific recommendations. Since many organizations such as Amazon, Alibaba or Netflix permit their customers to rate their products and services, they are able to collect a massive amount of data that can be used to predict what rating a particular user will give a specific item. Items with a high predicted rating are then recommended to users. For example, Netflix uses a recommendation system that predicts how many stars a user will give a particular movie, with one star representing a poor movie and five stars an excellent movie.

As the Netflix database is not publicly available so far, the GroupLens research lab generated their own dataset with over 20 million ratings for over 27,000 movies by more than 138,000 users. However, the MovieLens dataset which will be used in this project is only a small subset of the much larger original version. In order to make the computation easier, this project will use the 10M version of the MovieLens dataset. Each row in the data represents a rating given by one user to one movie, including information such as user id, movie id, genre, title, and rating. The main goal of this project is to train a machine learning algorithm using the inputs in one subset to predict movie ratings in the validation set. Building on user, genre and movie inputs, this project constructs a model that identifies the drivers of movie ratings and improves the prediction performance of the recommendation system. The residual mean squared error (RMSE) will be used to evaluate how close predictions are to the true values in the validation set.

The report is structured according the performed key project steps and therefore proceeds as follows: In the next section, the methods and analysis procedures will be described by highlighting the data preparation, exploration and visualization techniques and outcomes. Based on these insights, the modelling approach will be described and explained. The presentation and discussion of the modelling results (incl. RMSEs) follow. The report concludes with general learnings, outlines the limitations of the applied and tested approach, and provides directions for future modelling endeavors.

Methods & Analysis

Data Preparation

After the corresponding dataset has been loaded, the first steps is to define the respective columns (e.g., movieId, title, etc.) and transform them into the appropriate format. This is followed by a separation into training (“edx”) and test set (“validation”), with the validation set representing 10% of the MovieLens data. In addition, the project ensures that the userId and movieId data in the validation set are also in the training set. An excerpt of the training set after data preparation is shown in the following table, consisting of 6 columns and 9,000,055 rows:

userId	movieId	rating	timestamp	title	genres
<int>	<dbl>	<dbl>	<int>	<chr>	<chr>
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

3 rows

Data Exploration

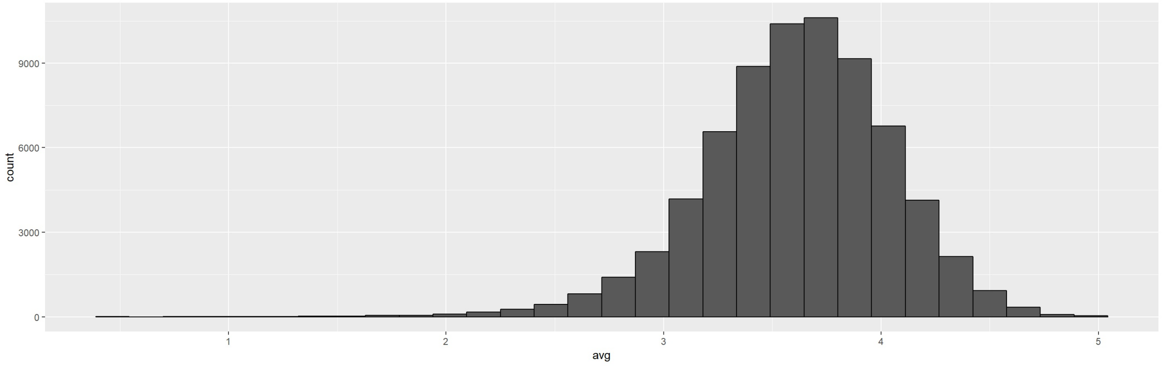
The data exploration starts by analyzing how many different movies, users and genres are in the training set. 10,677 movies, 69,878 users and 797 genres could be identified.

n_users	n_movies	n_genres
<int>	<int>	<int>
69878	10677	797

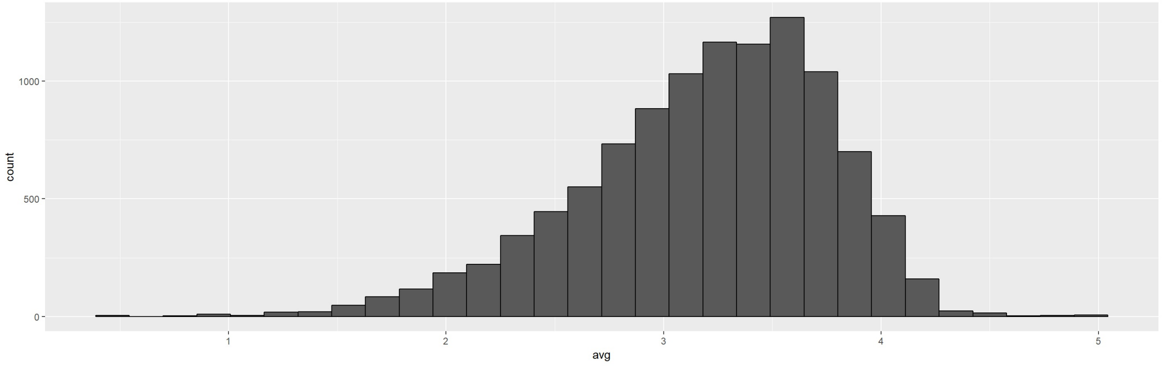
1 row

Note that the goal is to predict the rating for movies by users and in principle all other ratings related to movies and by users can be used as predictors, but different users rate different movies and a different number of movies. A machine learning algorithm seems to be there quite complex. In order to get a better understanding on the distribution of the data and potential drivers of ratings, the average ratings and number of ratings across movies, users and genres are computed and graphical illustrated.

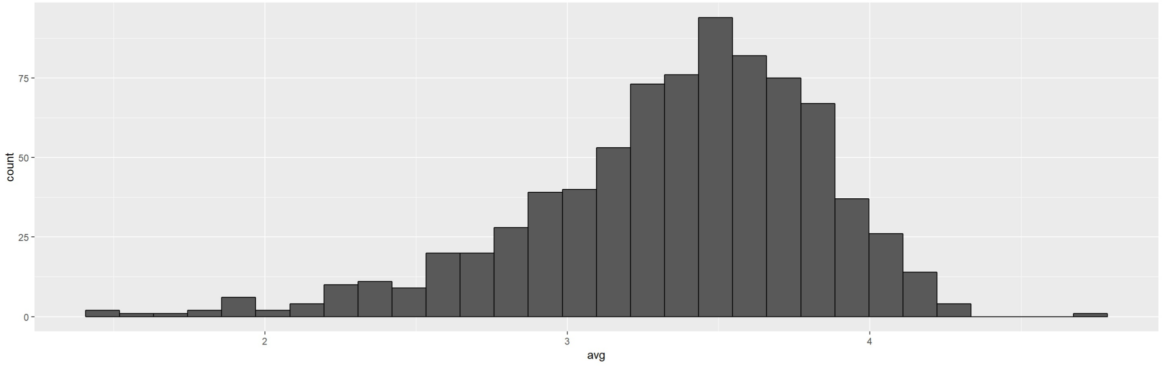
Average rating across users



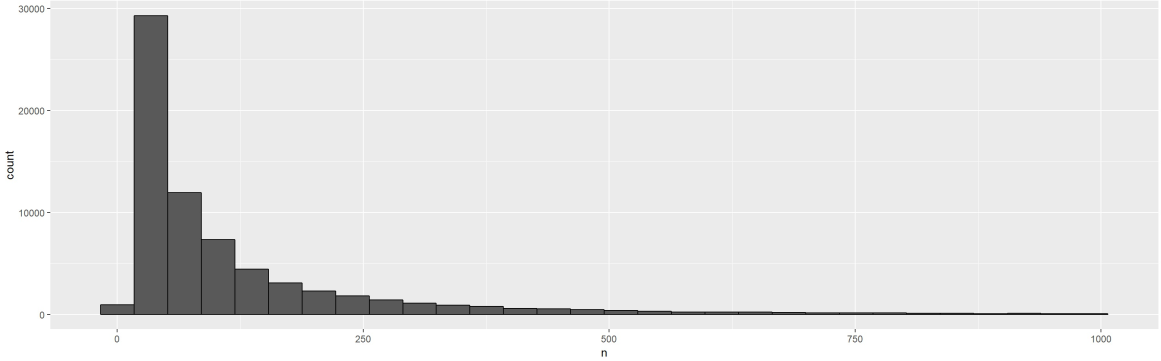
Average rating across movies



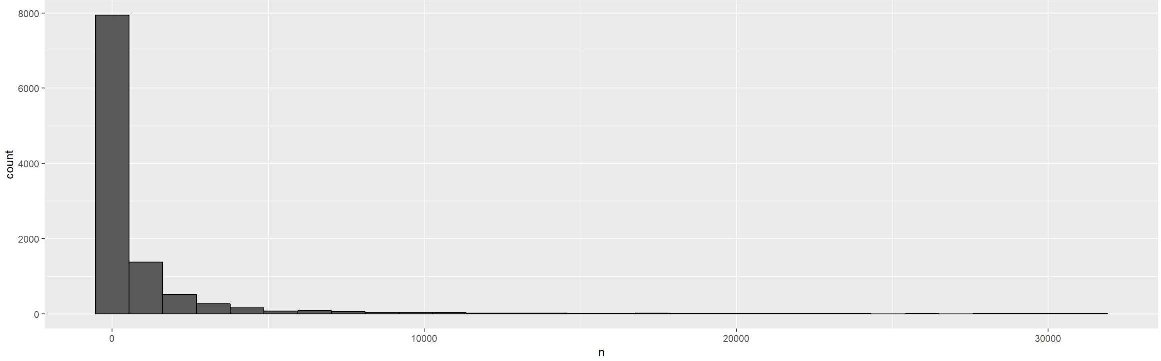
Average rating across genres



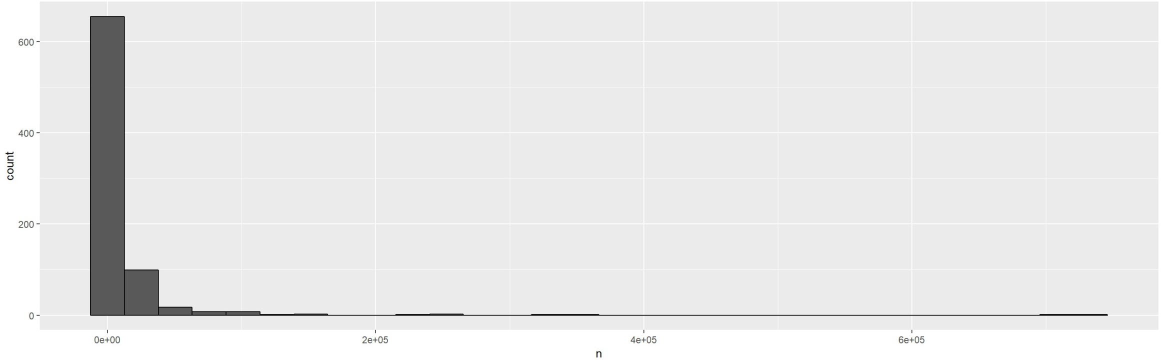
Number of ratings across users



Number of ratings across movies



Number of ratings across genres



The histograms indicate that the distribution of average ratings appears to be normally distributed. The first thing the histograms show is that some movies get rated more and higher than others. This is no surprise because there are blockbuster movies watched by millions and are very popular, independent movies watched by just a few. A second observation is that some users are more active than others at rating movies. There is substantial variability across users` average rating as well: some users are very cranky and others love every movie. These insights suggest that the movie, user, and genre data could potentially affect the rating. However, before proceeding with the description of the modeling approach, the missing values are analyzed. Overall, no missing values could be identified in the training and validation set.

Modeling Approach

As already mentioned, the project uses the RMSE to evaluate how close predictions are to the true values in the validation set. The projects defines $y_{u,i}$ as the rating for movie i by user u and denote the prediction with $\hat{y}_{u,i}$. Thus, the RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

with N being the number of user or movie combinations and the sum occurring over all these combinations. The RMSE can be interpret similarly to a standard deviation. The project uses a stepwise procedure to construct the model and compares the different approaches based on the RMSE results. Please note that the calculation will not be based on the lm function because of computational power issues.

“Just the average”: The project starts by building the simplest possible recommendation system: predicting the same rating for all movies regardless of user. This model assumes the same rating for all movies and users with all the differences explained by random variation and looks like this:

$$Y_{u,i} = \mu + \epsilon_{u,i}$$

With $\epsilon_{u,i}$ as independent errors sampled the same distribution centered at 0 and μ the true rating for all movies. The estimate that minimizes the RMSE is the least squares estimate of μ and, in this case, is the average of all ratings.

Modeling movie effects: The histograms show that some movies are rated higher than others. The model can be therefore augment by adding the term b_i to present the average movie ranking:

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

Note that the least square estimate $\hat{b}_i$ is the average of $Y_{u,i} - \hat{\mu}$ for each movie i.

Modeling genre effect. Although the histograms do not clearly show that the genres might also have an effect on the ratings, it seems plausible to control for genre-specific effects b_g und extent the movie effect model accordingly. The equation is described as follows:

$$Y_{u,i} = \mu + b_i + b_g + \epsilon_{u,i}$$

To fit this model, an approximation will be computed by computing $\hat{\mu}$ and $\hat{b}_i$ and estimating $\hat{b}_g$ as the average of $Y_{u,i} - \hat{\mu} - \hat{b}_i$. The result section will later show that the inclusion of genre effects does not improve the RMSE and therefore will be excluded in the further model design.

Modeling user effects: Based on the data visualization results, the movie effect model could be further improved by including a user-specific effect b_u:

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

To fit this model, an approximation will be computed by computing $\hat{\mu}$ and $\hat{b}_i$ and estimating $\hat{b}_u$ as the average of $Y_{u,i} - \hat{\mu} - \hat{b}_i$.

Regularization of movie and user effects: The project computes standard error and constructed confidence intervals to account for different levels of uncertainty. However, one number, one prediction and not an interval is needed when making predictions. That is why it makes sense to consider the concept of regularization. The general idea of the regularization concept is to control the total variability of the movie and user effects. More precisely, instead of minimizing the least square equation, an equation that adds a penalty will be minimized:

$$\frac{1}{N}\sum_{u,i}(y_{u,i} - \mu - b_i)^2 + \lambda \sum_i b_i^2$$

The first term represents the least squares and the second is a penalty that gets larger when many b_i are large. The values of b_i that minimize this equation are:

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

Where n_i is the number of ratings made for movie i. λ is a tuning parameter and can be chosen by using cross-validation.

Results

The table below reports the results of the modeling approach and the RMSE for each model. As described in the previous section, the project proceeds stepwise to construct the model and compares the different approaches based on the RMSE results. Note that missing values are replaced by the mean value, where necessary in the context of the respective steps of the modeling approach.

method	RMSE
Just The Average	1.0612018
Movie Effect Model	0.9383091
Movie + Genre Effect Model	1.0239819
Movie + User Effect Model	0.9046108
Regularized Movie + User Effect Model	0.8648170

In the first model, only the average of the ratings for all movies is calculated regardless of users. As a starting point, this model achieves a RMSE of 1.06. In the second step, the movie effects are tested, whereby the RMSE improves to 0.94. However, considering genre effects does not contribute to a better RMSE. Therefore, the genre variable is not included in the further model development process. In contrast, the table shows that the RMSE is reduced to 0.90 by adding user effects. Before describing the results of the regularization approach, the tables below show that many movies were rated by very few users and larger estimates are more likely. Large errors can increase RMSE, so it make sense to be conservative when unsure. Since the regularization of movie and user effects controls the total variability of effects, the results indeed show that in the final model the RMSE can be improved to 0.86.

10 best movies

title	b_i	n
Adam & Steve (2005)	1.487535	1
Atragon (Kaitei Gunkan) (1963)	1.487535	1
August Evening (2007)	1.487535	1
Bigger Than the Sky (2005)	1.487535	1
Bridge, The (2006)	1.487535	2
Charm's Incidents (Charms ZwischenfÃ¤lle) (1996)	1.487535	1
City of Ember (2008)	1.487535	1
Close-Up (Nema-ye Nazdik) (1990)	1.487535	1
Defying Gravity (1997)	1.487535	1
Dolemite (1975)	1.487535	1

10 worst movies

title	b_i	n
Across the Pacific (1942)	-3.012465	1
Amy (1998)	-3.012465	1
Attack Force Z (a.k.a. The Z Men) (Z-tzu te kung tui) (1982)	-3.012465	1
Beginning of the End (1957)	-3.012465	1
Beverly Hills Chihuahua (2008)	-3.012465	1
Blood and Black Lace (Sei donne per l'assassino) (1964)	-3.012465	2
Cinderella (1997)	-3.012465	1
Cookout, The (2004)	-3.012465	2
Daltry Calhoun (2005)	-3.012465	1
Darfur Now (2007)	-3.012465	1

Conclusion

The purpose of this project is to create a movie recommendation system based on the MovieLens dataset. Applying machine learning techniques, the results suggest that, in particular, movie and user-specific variables are suitable drivers for predicting movie ratings. The final model achieves a RMSE of 0.86. The results can be used to help organizations such as Netflix or Amazon to provide their customers better services by recommending movies closer to actual customer preferences. Particularly in a highly competitive market such as movie platforms, high customer satisfaction and customer loyalty are significant competitive factors.

This project has some limitations that point to interesting avenues for future research and modelling endeavors. Although genre effects do not contribute to a reduction of the RMSE, it would be interesting to further study this variable. For example, better effects may be achieved if the variable is further split into individual genres such as action, comedy, etc. and thus does not contain a combination of multiple genres per movie. Furthermore, when testing the machine learning algorithm only 10% of the original MovieLens data set is used. Future modeling efforts could use a larger proportion of the dataset and test different machine learning techniques (e.g., random forest) or ensemble models.