MovieLens Project

Reid Hellekson

12/13/2019

Introduction

This is a submission for the MovieLens project performed by Reid Hellekson. The goal of this project was to demonstrate techniques learned in Professor Rafael Irizarry's edx Data Science Program by creating a movie recommendation system using the 10M record version of the MovieLens dataset (https://grouplens.org/datasets/movielens/10m/). Ultimately, we look to create a recommendation system that has a root mean squared error of less than 0.8649.

The provided dataset contains movie ratings by users that also includes a timestamp and a concatenated field of all applicable genres. In the below, details on data wrangling and manipulation are presented. Then we delve into exploration, visualization, and insights. Also, the modeling approach and ultimately the results are presented.

Analysis

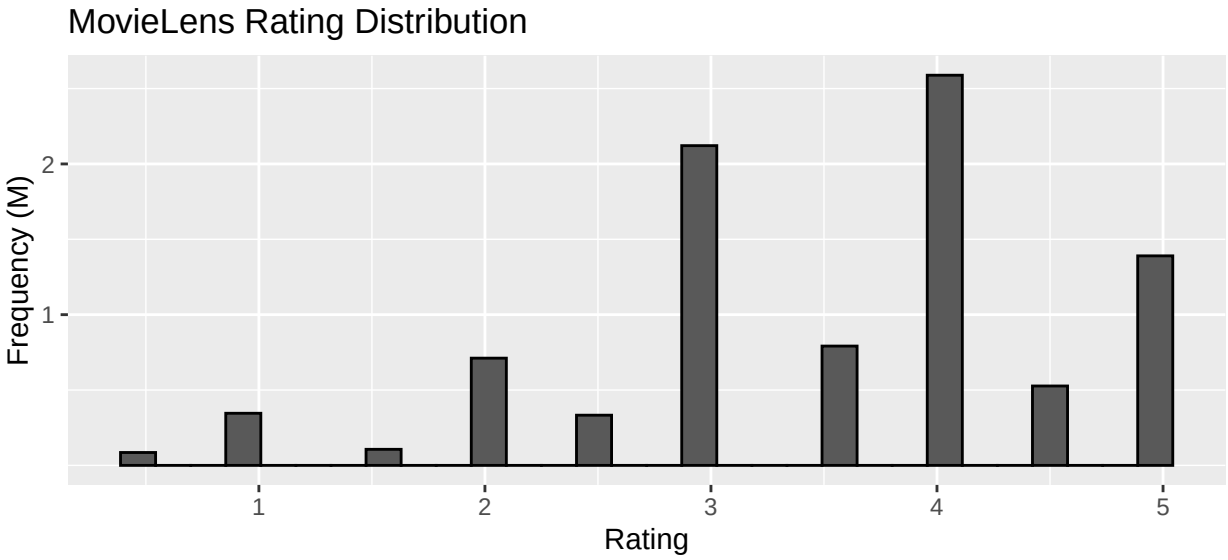
In this project, a lot of the data extraction and transformation is performed by the provided script. By implementing this script, the MovieLens dataset is downloaded and the two files are unzipped - "movies" and "ratings". Delimiters are removed, column names added, and data types are forced on both files. Then, the two files are combined so we have "ratings" and "genres" in one nearly tidy dataset. The data wrangling process is completed by creating a clean test and validation set for the analysis. The resulting dataset looks like this:

userId	movieId	rating	timestamp	title	genres
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
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

It is worth noting that this data is not necessarily tidy. There is a release year concatenated into the title column and the genres column is a pipe delimited combination of all applicable genres. However, the data is sizable and we will forgo the added overhead of extra tidying at this point.

Exploration

To begin, let's take a quick look at the data and how it is distributed.



Number of Movies	Number of Users	Number of Genres	Average Rating
10677	69878	797	3.51

	minimum	q1	median	mean	q3	maximum
Ratings per Movie	1	30	122	842.9386	565	31362
Ratings per User	10	32	62	128.7967	141	6616

Mean Model

Visually, there is an apparent clustering around the mean (i.e. between a rating of 3 and 4). Let's begin with a model based on a constant assumption of the sample mean and get a sense of our error terms. In other words, we look at:

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

And get RMSE = 1.0603313.

Where root mean squared error (RMSE) is defined by:

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

Movie Effect

Let's see if we can improve this by including a consideration for a movie effect or bias. In other words, some movies may be rated differently than others, so let's model:

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

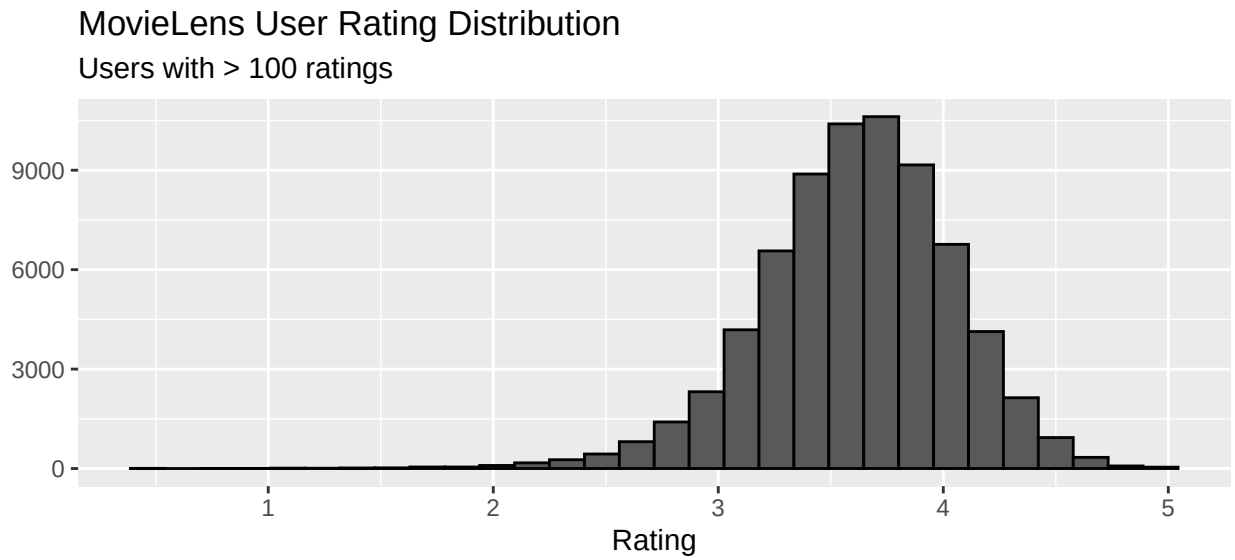
where

$$b_i = Y_{u,i} - \mu$$

By including this movie bias, we get RMSE = 0.9423475. An improvement!

User Effect

Next, let's examine whether there is any explanatory information in the user itself. First, let's see if users are on average different. Here is a distribution of users' average rating for users that have rated more than 100 movies.



We see that there is variation across users, so let's now try a model like:

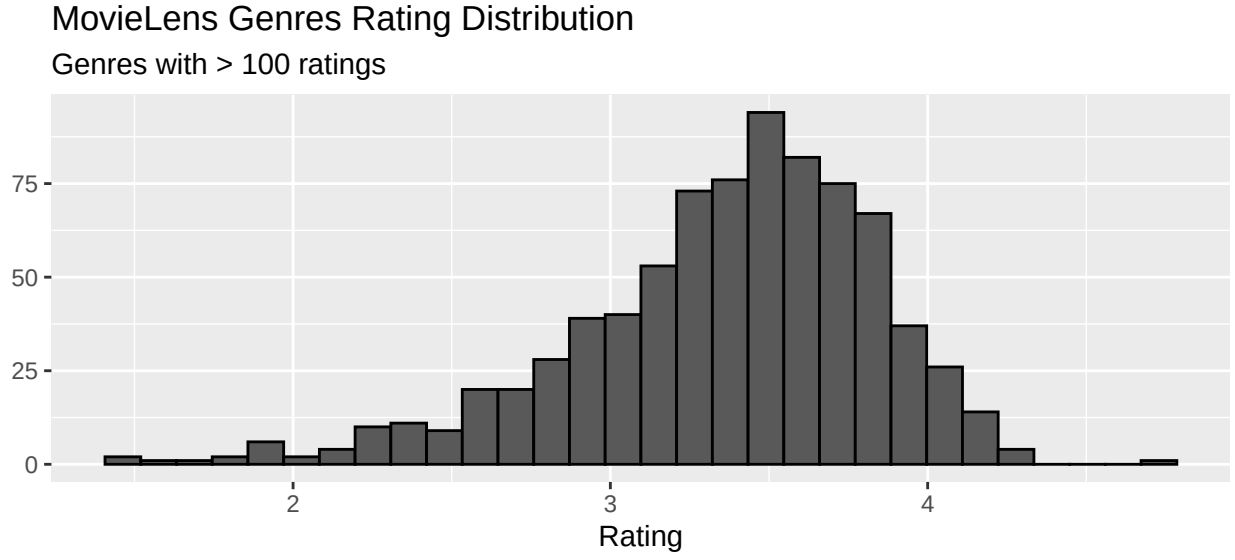
$$Y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$$

where b_u is the effect of the user giving the rating.

By including this user bias, we get RMSE = 0.8567039. More improvement!

Genre Effect

As noted earlier, the MovieLens data includes a genres variable. With more resources, one might be tempted to split this out into a more tidy framework. However, the concatenated field can be treated as a variable itself. The concatenation appears to be consistent, so let's examine the inclusion of a genres bias.



There is notable variation here as well. If we add this into the model as we've done with the previous biases, we get:

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

where b_g is the effect of the genre.

After this decomposition, we have a new RMSE of: 0.8563595. More improvement still!

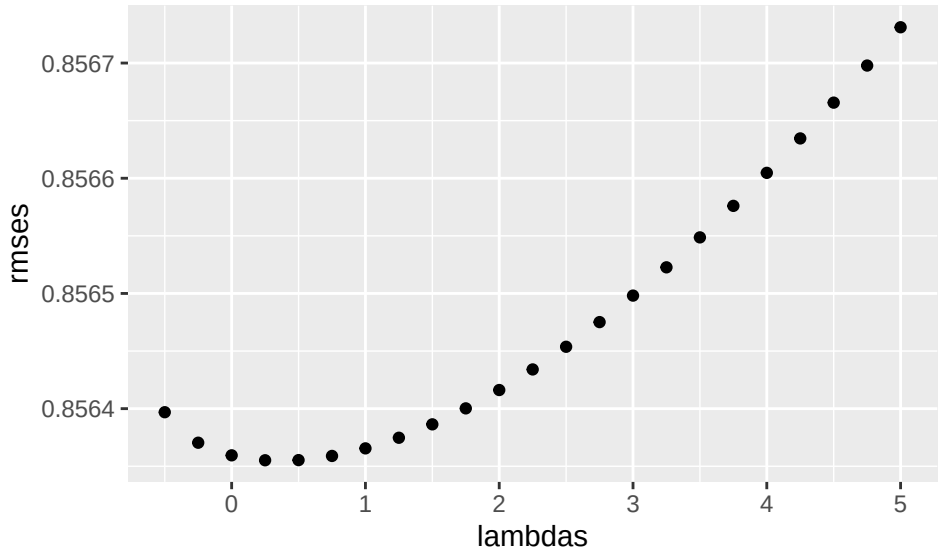
Regularization

Ok, now what about overfitting? Early in our exploration of the data, we noticed that there were users who hadn't rated many movies and numerous movies that hadn't had many ratings. Could our biases be too strong in some cases? Let's look at our best and worst predicted movies (i.e. movies where our movie bias model was most wrong).

title	error	predicted	rating	#Ratings
From Justin to Kelly (2003)	4.097990	0.9020101	5.0	199
PokÃ©mon Heroes (2003)	3.970803	1.0291971	5.0	137
Shawshank Redemption, The (1994)	-3.955131	4.4551312	0.5	28015
Godfather, The (1972)	-3.915366	4.4153660	0.5	17747
Carnosaur 3: Primal Species (1996)	3.911765	1.0882353	5.0	68
Usual Suspects, The (1995)	-3.865854	4.3658537	0.5	21648
Schindler's List (1993)	-3.863493	4.3634933	0.5	23193
Glitter (2001)	3.824484	1.1755162	5.0	339
Pokemon 4 Ever (a.k.a. PokÃ©mon 4: The Movie) (2002)	3.821782	1.1782178	5.0	202
Casablanca (1942)	-3.820424	4.3204238	0.5	11232

There are some well known movies here that had some potentially biased user ratings. However, there are also a number of obscure movies in the above. These movies have relatively few ratings. It appears that there is evidence of overfitting. In each of our biases, we can constrain the total variability of the effect of sizes by employing regularization.

With this form of regularization, there is a tuning parameter, λ. If we plot, possible values for λ against the resulting error term in our test set, we get the following:



From this plot we see that the RMSE is minimized at 0.25. This value of λ results in an error of 0.8563552. This is a pretty good result. We have further converged our error metric. And, in the next section, we will see how all of these models compare and if we have met our threshold on the validation set as well.

Results

This MovieLens dataset is large, so we began with a basic model. As we explored the data, we found ways to increase the complexity without adding too much overhead. To recap our results:

Method	RMSE
Mean Model	1.0603313
Movie Effect Model	0.9423475
Movie and User Effect Model	0.8567039
Movie, User, and Genres Effect Model	0.8563595
Regularization Movie, User, Genres Model	0.8563552

So far we have only looked at our training set.

When we run the finalized model against the validation set, we get an error of: 0.8648817. Bingo! We have an error under our benchmark.

Conclusion

We began with the 10M record MovieLens dataset that was downloaded via Professor Irizarry's script. We cleaned the data, explored the data, and applied the techniques from the "Recommendation Systems" curriculum. Some constraints in the way of memory and processor speed were noticeable and limited our ability to readily apply a multitude of models. Future work could involve employing boosting/bagging techniques and more sophisticated models like neural networks.