Movielens Project

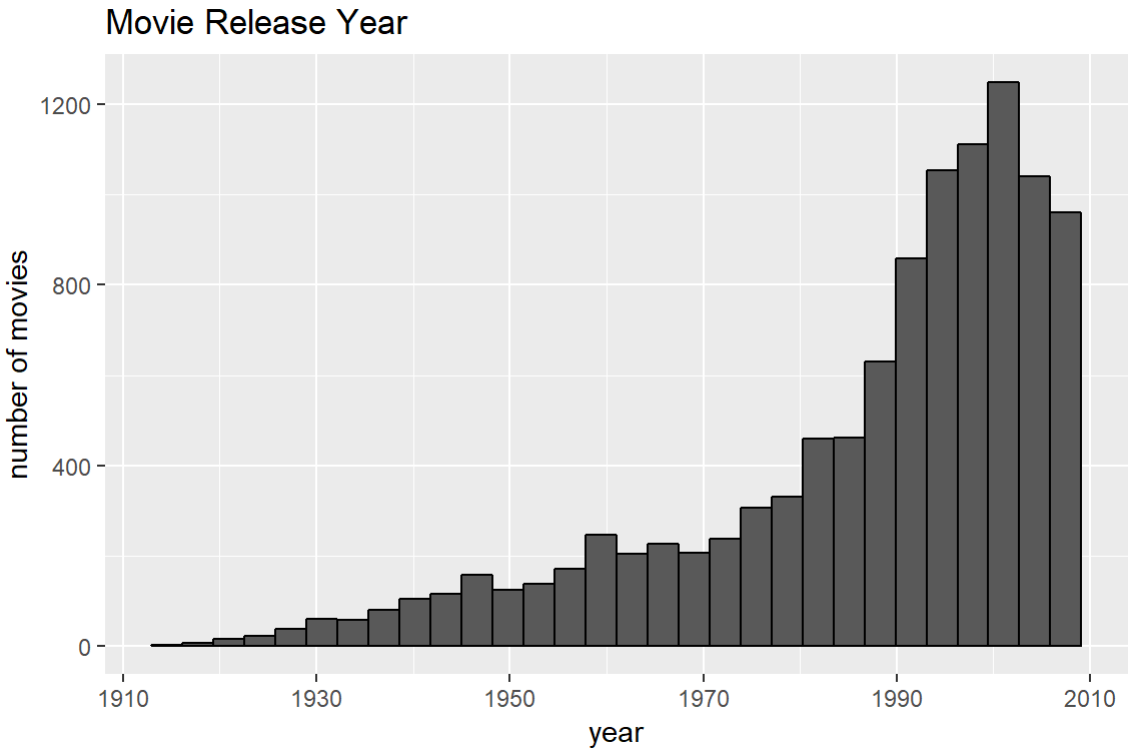
Kyle Karber

April 17, 2019

Introduction

The objective of this project is to predict movie ratings for specific individuals based on other movies that they have rated and other people's ratings for the movie being predicted. This objective is motivated by the need to make accurate movie recommendations on streaming platforms such as Netflix.

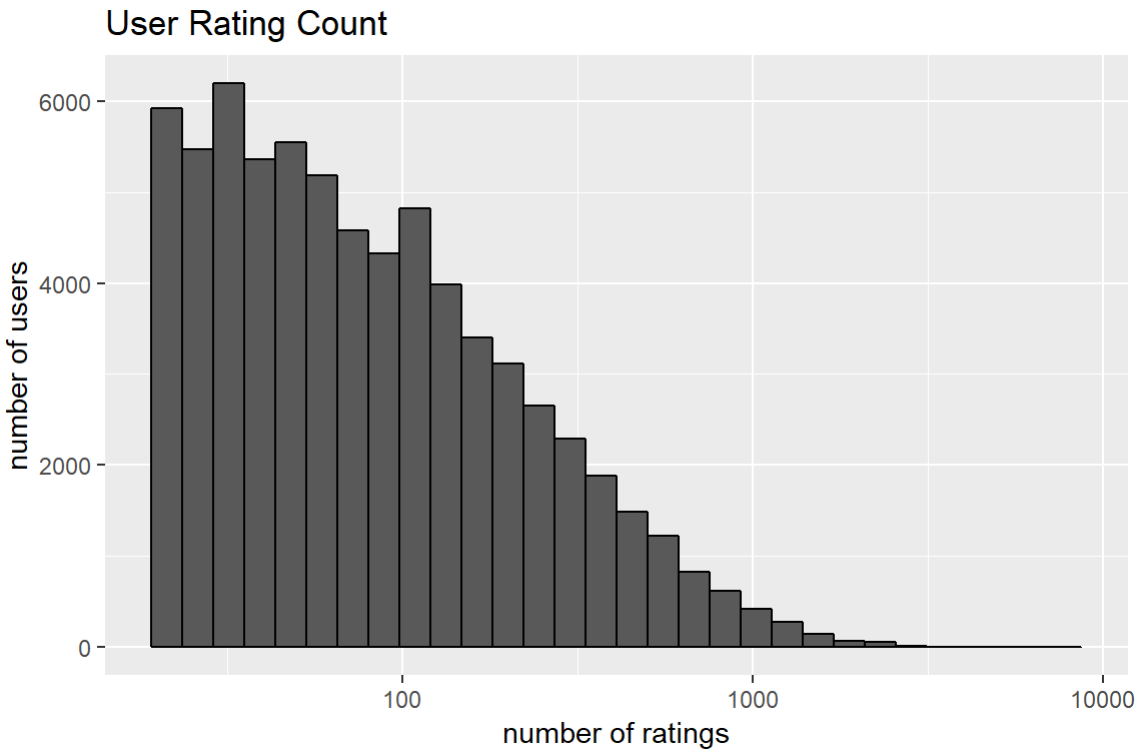
The dataset consists of about 10 million movie ratings and 6 variables: user ID, movie ID, rating, movie title, movie genre, and the date and time of the rating. There are 10681 unique movies with dates ranging from 1915 to 2008, as shown in the histogram below.



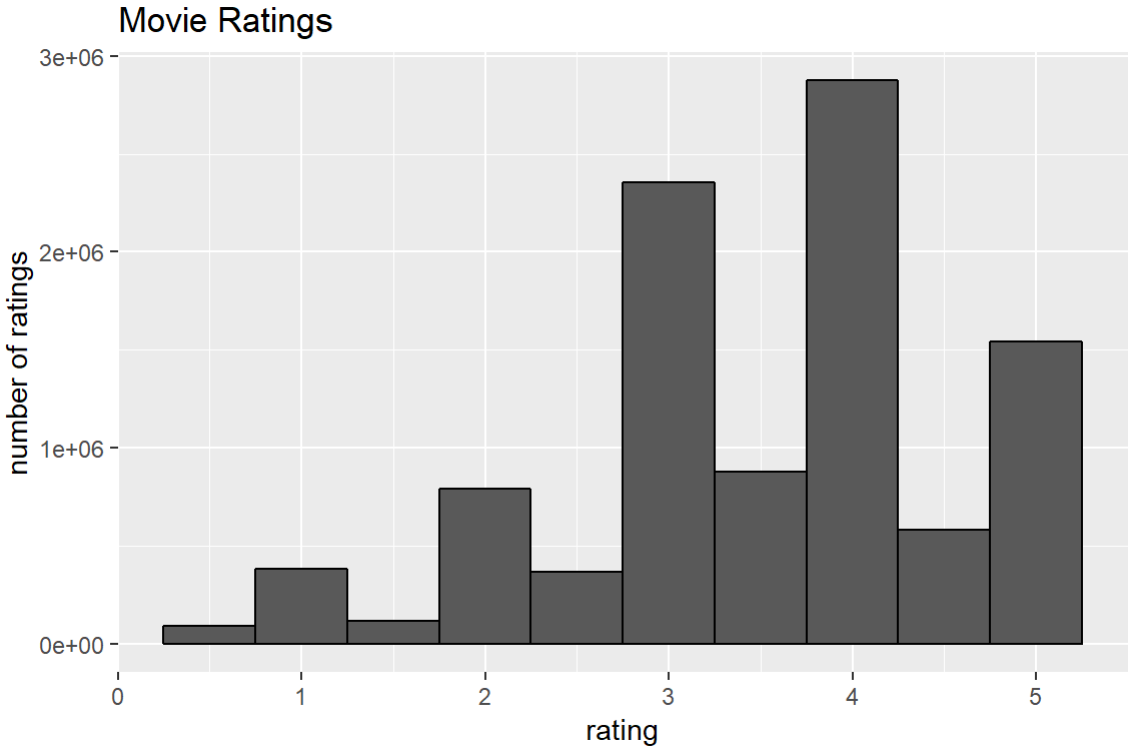
There are 19 non-exclusive movie generes, with the top five most popular genres shown in the table below.

genres	count
Drama	5339
Comedy	3703
Thriller	1706
Romance	1685
Action	1473

There are 69878 unique users, and the number of movies they rated is shown in the histogram below.



Ratings range from 0.5 to 5, with the distribution of ratings shown in the histogram below.



In order to predict movie ratings, I began by downloading the data from http://files.grouplens.org/datasets/movielens/ml-10m.zip (http://files.grouplens.org/datasets/movielens/ml-10m.zip). I cleaned and preprocessed the data as necessary. I developed a simple linear regression model that accounts for movie effects and user effects. I also applied regularization to the model, but it had minimal effect on model error.

Methods/Analysis

The data required minimal cleaning and preparation before analysis. There were no missing data and the datasets were already in tidy format. The movie details and ratings were in seperate datasets, so I merged them into one dataframe. I converted the UTC timestamp into date-time format. The movie title included the release year, so I removed it from the title and created a seperate column for the year.

I created the model piece-by-piece, starting with the naive model: the outcome $Y_{u,i}$ for user u and movie i is the overall average rating for all movies and users (μ) plus some error term ($\varepsilon_{u,i}$).

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

The second model included the movie effects (b_i), which is the average rating of each movie. This term accounts for the movie to movie variability, i.e. some movies are better than others.

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

The third model included the user effects (b_u), which is the average rating a user gives. This term accounts for user to user variability, i.e. some users are harsher critics than others.

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

The final model included regularization in order to penalize large estimates of b_i and b_u that are made with small sample sizes. For example, when estimating b_i with the equation below, a penalty parameter (λ) of five can significantly decrease the magnitude of the estimate when the number of ratings (n_i) is small, but has minimal effect when n_i is large.

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

The models were evaluated based on the residual mean squared error (RMSE), which is defined by the following equation:

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

where $y_{u,i}$ is the rating for movie i by user u, $\hat{y}_{u,i}$ is our prediction, and N is the total number of user and movie combinations that we are predicting/evaluating.

Results

The results of the four models are shown in the table below.

method	RMSE
Naive Model	1.0612
Movie Effect Model	0.9439
Movie + User Effects Model	0.8653

method	RMSE
Regularized Movie + User Effect Model	0.8648

Including the movie effect showed a 19% improvemnt in RMSE over the naive model. Including the user effect showed a significant improvement in RMSE over the movie effect model. Regularization did not have a significant impact on the RMSE. This is probably because this dataset has a high number of ratings for each movie, with less than 10% of movies having fewer than 10 ratings. Regularization would prove more useful on smaller movie rating datasets which have fewer ratings per item.

Conclusion

I created a model that accounted for movie-to-movie rating variability and user-to-user rating variability. This modeling approach resulted in a significant reduction in movie rating prediction error compared to the naive model. The model was easy to interpret and was not computationally intensive. There are a number of other models that could be employed, such as user-user or item-item colaborative filtering using matrix factorization or neural networks, but these methods are very computationally intensive given our large and sparse dataset.