MovieLens Capstone Project

Alyse Carter Schultz
9/7/2021

For Harvard EdX Data Science Certificate.

OVERVIEW/ EXECUTIVE SUMMARY

The goal of this project is to create a movie recommendation system. In this project, I will create a machine learning algorithm to predict what any given user will give a movie based on a data set of users ratings of movies. I will compare the results of my algorithm's predictions to the actual predictions using an RMSE score. To evaluate the accuracy of the recommendation system, we will use RMSE and consider anything lower than 0.86 to be a highly accurate estimate.

This project uses the Movielens Data source. The data contained in the movielens data source includes unique userIds for each individual doing the ratings, as well as a unique numerical IDs for each movie. The movies also include information such as title, genre, etc.

The data was divided in to two sets - the "edx" set to be used for testing and training purposes, and the "validation" set to test the final algorithm. The two predictors used in this analysis were UserId and MovieId. The RMSE methods were first tuned on the edx set and then applied to the validation set to achieve the final prediction.

METHODS

Project Steps:

I followed these steps to create and tune the prediction algorithm: 1. Identify a possible predictor 2. Test that predictor for significance (how much bias it creates) 3. Add that predictor to the model. 4. Add Regularization to predictors with more uncertainty. 4. Repeat until an RMSE of <0.85 is achieved.

To generate an RMSE, I had to first define the function to predict the RMSE. This function computes the residual means squared error for a vector of ratings and their corresponding predictors.

I defined RMSE using this code:

```
#this function computes the residual means squared error for a vector of rati
ngs and their corresponding predictors

RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

To begin the tuning process, I computed the most basic predictor: to simply predict the outcome as the average of all ratings. The average rating is:

```
## [1] 3.512465
```

So, if I predict that a given user will rate any given move 3.51, and test this using RMSE, I get the below result:

```
average_only_model <- RMSE(edx$rating, mu)
average_only_model

## [1] 1.060331
```

I can see that this result is very poor and is far away from the target value of <0.86.

To keep track of the model progress, I added the results into a results table, starting with the average only model:

```
##                model     RMSE
## 1 Average Only Model 1.060331
```

I will use this average as the undesirable baseline to compare the results of my models to.

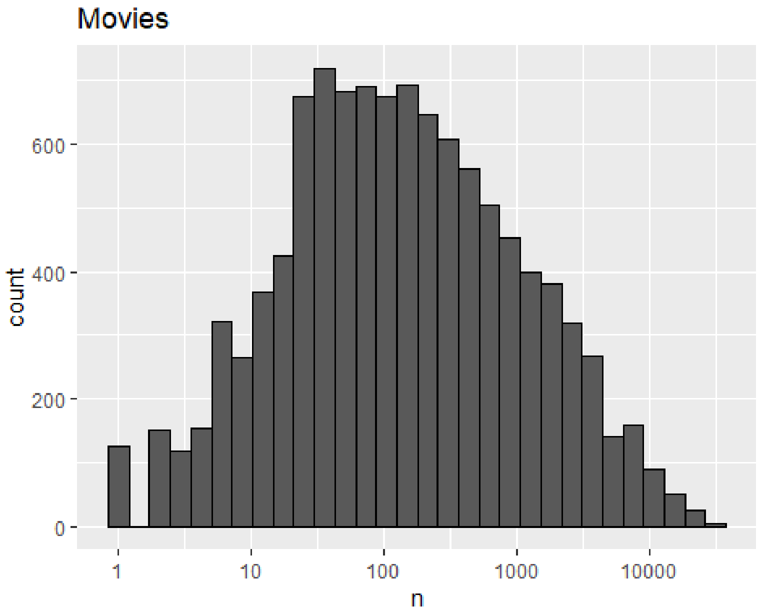
ANALYSIS/RESULTS

To create a better recommendation system, the first step is to identify predictors that have significant bias. Accounting for a biased predictor should increase the accuracy of the prediction model.

Upon examination of the data, I was able to identify several possible predictors for how a given user will rate a movie.

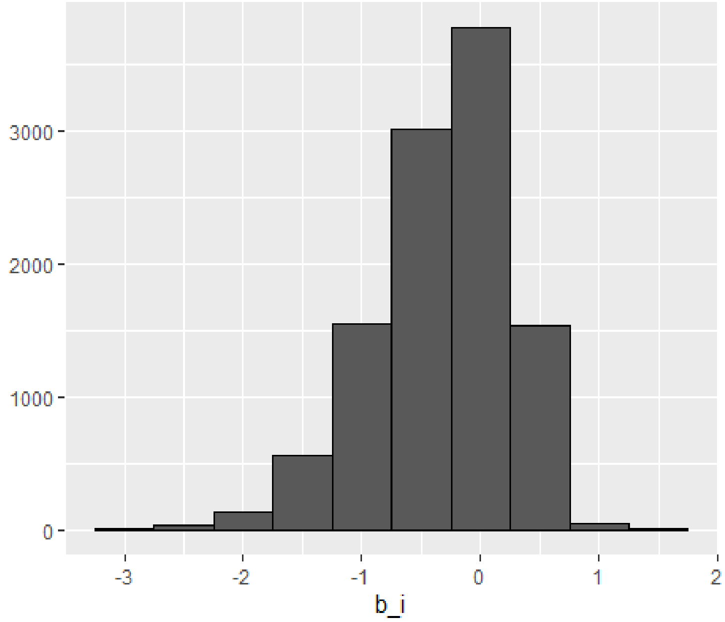
First, I explored if some movies are rated higher than other movies in general. This may be that some movies are more popular or better advertised, and so more users have seen and rated the movie.

I graphed movieId vs the total movies to see if this was the case:



Clearly, there are some movieIds that are being rated more often than others, and some movies with almost no ratings.

First, I estimate the bias of this predictor (movieId), using the term “b_i” to indicate this bias, and graph the results:



From this graph, it is clear that there is a lot of variation in these estimates, so this should be a good predictor.

Using this single predictor, I fit a prediction model and ran it on the validation set using this code:

```
predicted_ratings <- mu + validation %>%
  left_join(b_i, by='movieId') %>%
  .$b_i

movie_model_rmse <- RMSE(predicted_ratings, validation$rating)
movie_model_rmse

## [1] 0.9439087
```

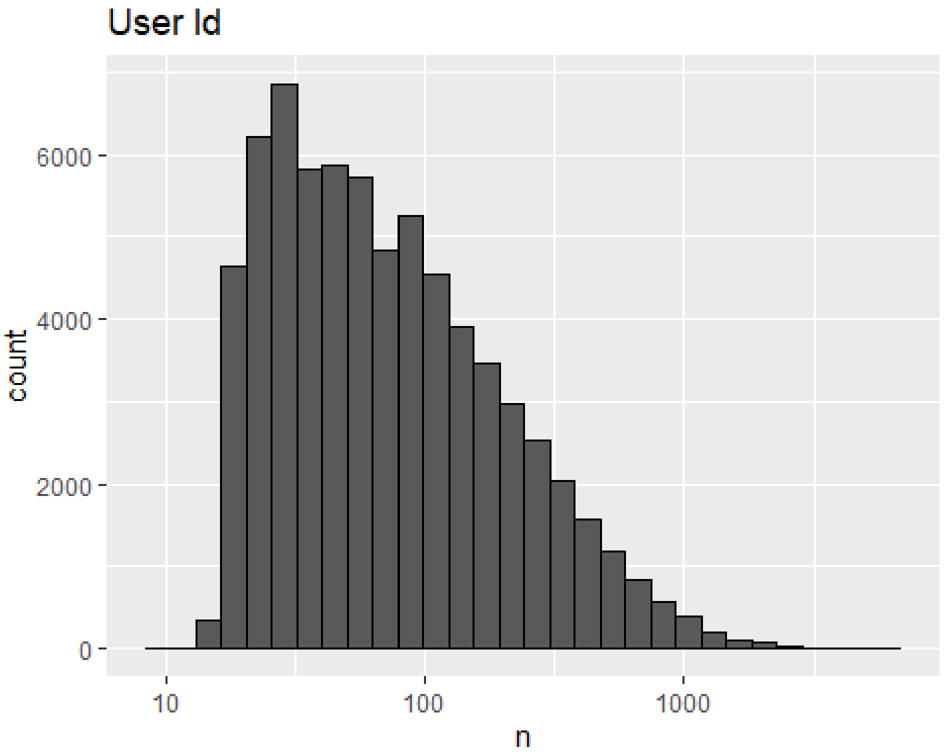
This gave me a slightly better result than just the average.

```
##                 model      RMSE
## 1 Average Only Model 1.0603313
## 2  Movie-Based Model 0.9439087
```

This result is still too far away from the target result. Clearly this one predictor is not enough to accurately predict what a user will rate a movie.

Next, I explored if some users rate more movies than others. Perhaps some movie aficionados spend a lot of time rating movies, and others may have only rated one or two movies they particularly liked or hated.

I graphed number of movies rated per user to see if there is a discrepancy:



This graph shows me that indeed some users are rating movies more often, so this should be a good predictor as well. I computed this bias using the "userId" column and assigning it the variable b_u. I then added this as a second predictor to the previous model:

```
#computes bias of user averages
b_u <- edx %>%
  left_join(b_i, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_i))

#adds user averages to the previous model.
predicted_ratings <- validation %>%
  left_join(b_i, by='movieId') %>%
  left_join(b_u, by='userId') %>%
  mutate(prediction = mu + b_i + b_u) %>%
  .$prediction

movie_user_model <- RMSE(predicted_ratings, validation$rating)
movie_user_model

## [1] 0.8653488
```

Using these two predictors, we have again slightly improved our accuracy, as we can see now in the table:

```
##                    model      RMSE
## 1     Average Only Model 1.0603313
## 2      Movie-Based Model 0.9439087
## 3 Movie/User Based Model 0.8653488
```

While 0.86 is a good RMSE it is not quite below the threshold for success, so I added one additional part to the model to try to increase the accuracy a bit more.

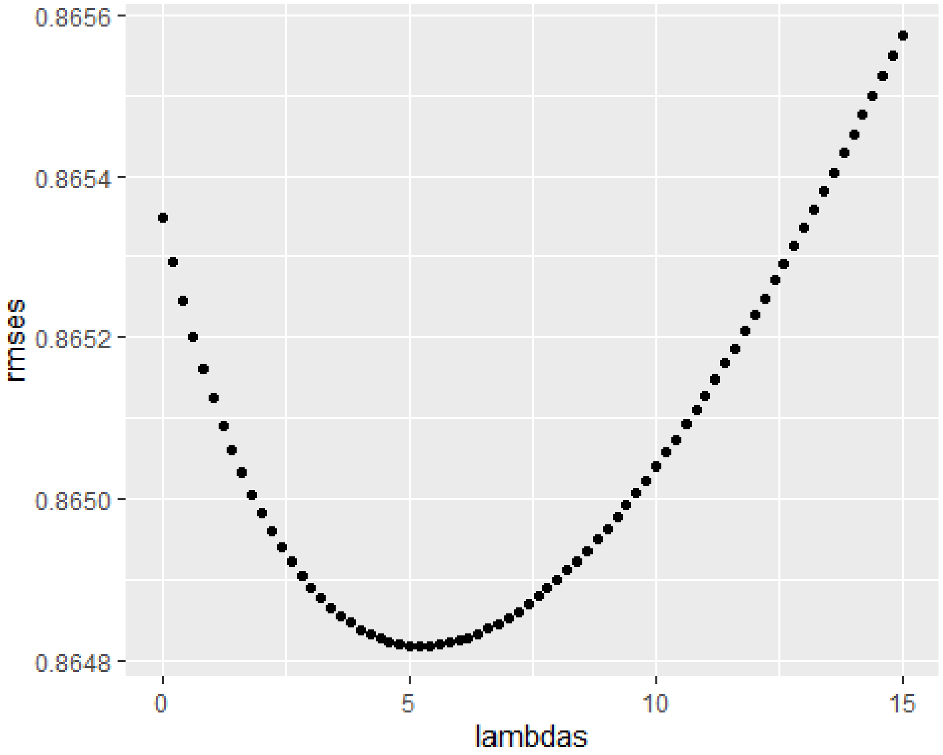
It appears that some movies are rated very high or low but have very few ratings. These are “noisy” estimates that should not be trusted for prediction. Because of this, it is necessary to use Regularization so that we can penalize the large estimates that come from small sample sizes. We will use regularization to penalize the noisy estimates. In practice, this means if there is a large number of ratings for a specific movie or from a specific user, the equation will not be penalized. However, if a movie or user has very few ratings (a small n), the penalty value will shrink that estimate towards Zero to balance out the effect of that rating on the averages.

To apply regularization, I first must select the lambda value. This is the value that will penalize the equation and shrink estimates the farther away an estimate is from zero. I used the below code to graph and choose the appropriate lambda value to minimize the RMSE of the movieId field:

```
#define a sequence of lambdas
lambdas <- seq(0, 15, 0.2)

#calculate the predicted ratings using all above values of lambda
rmses <- sapply(lambdas, function(lambda){
  mu <- mean(edx$rating)
  #movie average
  b_i <- edx %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+lambda))
  #user average
  b_u <- edx %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+lambda))
  #predict ratings for validation set
  predicted_ratings <- validation %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    mutate(pred = mu + b_i + b_u) %>%
    pull(pred)
  #predict RMSE
  return(RMSE(validation$rating, predicted_ratings))
})

#plot lambdas
qplot(lambdas, rmses)
```



```
#lambda which minimizes RMSE
lambdas[which.min(rmses)]

## [1] 5.2
```

I can see from this plot that a lambda value of about 5.2 will best minimize the equation. I now use this to predict the RMSE on the validation data set, with this result:

```
## [1] 0.864817
```

by Regularizing the MovieIds and UserIds, I was able to slightly improve the model, and have now achieved an acceptable RMSE for this project.

FINAL RESULTS AND CONCLUSION

As can be seen in the below table, each model used slightly improved the RMSE until I achieved the target RMSE of <0.865

```
##                                 model      RMSE
## 1                  Average Only Model 1.0603313
## 2                   Movie-Based Model 0.9439087
## 3              Movie/User Based Model 0.8653488
## 4 Regularized Movie/User Based Model 0.8648170
```

The goal of this project was to create a movie recommendation system. From analysis of the data source, I determined that two predictors were essential to predicting user ratings: the movie itself and the frequency of user ratings. I first applied an RMSE model to these two predictors and then applied regularization to further tune the algorithm. As the goal was to achieve an RMSE of >0.86, my model has successfully met the criteria set forth by this project.

Additional study and modeling could potentially further improve results. It is likely that users may rate certain genres more highly than others, so if I were to continue this analysis I would explore this possible addition to the model. Although it is not included in the given data set, my assumption is that demographic factors may affect user ratings, such as age, education, and gender. A model that incorporated more user information would likely be able to increase the accuracy of the model further.