MovieLens Capstone Final Report (Data Science HarvardX)

Daniela Garcia

10/26/2020

Introduction

Dataset Description

The MovieLens dataset used in this project contains data on an extensive range of movies and ratings that users have given them. Each row in the data set contains a movie title, its movie ID, and the movie's genre(s), in addition to the ID of a user who has rated the movie, the rating they gave, and the time they rated it.

Project Goal

The goal of this project is to predict the rating a user will give a movie, given the user's ID, the time they rate the movie, the movie's title and ID, and the movie's genre(s).

Key Steps

The key steps of this project are as follows:

- Download and clean the MovieLens dataset
- Split the MovieLens dataset into a training ("edx") and testing ("validation") set
- Explore the variables in the training set ("edx")
- Split the training set ("edx") into its own training ("edx_train") and testing ("edx_test") sets
- Generate predictive models using only the training ("edx") data
- Select a final model and generate predictions for the original testing ("validation") set

Methods and Analysis

Data Cleaning

The finalized MovieLens dataset is generated by downloading the data on movies (movies.dat) and ratings (ratings.dat) separately, then joining the two datasets by movieId, the common variable between them.

The MovieLens set is then split into a testing ("edx") and training ("validation") set, where 90% of the MovieLens data is used for the training set. The testing set is filtered so that all users and movies present in

the testing set are also present in the training set. This prevents any errors when making predictions on the testing set. Any rows removed from the testing set during this process are then added back into the training set.

In order to build a predictive model, the training ("edx") set is then split into its own training ("edx_train") and testing ("edx_test") set using the same methods as above.

Data Exploration and Visualization

According to my data exploration, the distribution of ratings is left-skewed. 4 was the most common rating (28.76% of the ratings), with 3 and 5 following. More movies were given a rating of 3 or above than less than 3. One can view this distribution in the histogram below.

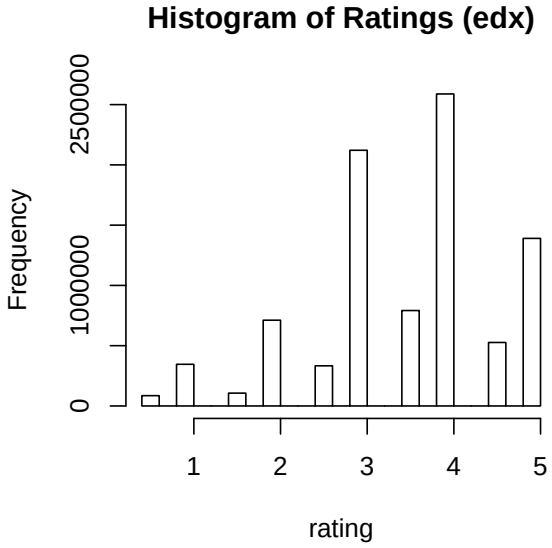
The distribution of movies by the number of times they were rated is not equal. Some movies were rated very frequently (such as Pulp Fiction, which was the most frequently rated with 31,362 ratings), while others were rated only once. The movies that were rated only once may skew the data, indicating that we should use regularization in the model-building process. The distribution appears in the histogram below. As discussed, the data is very right-skewed.

Proportions Table of Ratings (edx):

```
##
##         0.5           1         1.5           2         2.5           3
## 0.009485942 0.038408543 0.011825039 0.079046406 0.037000885 0.235691893
##         3.5           4         4.5           5
## 0.087957685 0.287601576 0.058525865 0.154456167
```

Most frequently rated movie:

```
##                 title movieId number_of_ratings
## 1 Pulp Fiction (1994)     296             31362
```



In addition to exploring the variables on their own, I calculated the correlations between average rating per movie, user, date/timestamp, and genre with overall rating, to see if each variable has an effect on the rating outcome. Based on the table below, average rating per movieId had the strongest correlation with rating (0.458), followed by userId (0.404), genre (0.280), and date/timestamp (0.084).

Based on these results, I decided to account for the effects from all four variables in my predictive model, despite the fact that the correlation between average rating per date/timestamp and overall rating was fairly low.

```
##     r_movie    r_user   r_genre     r_date
## 1 0.4584323 0.4038697 0.2797873 0.08435155
```

Insights Gained

Based on the exploration conducted above, I concluded that the variables "movieId", "userId", "timestamp", and "genres" all have somewhat of an effect on the rating given to a movie. As a result, I will include these four effects in my predictive regularized model.

Modeling Approach

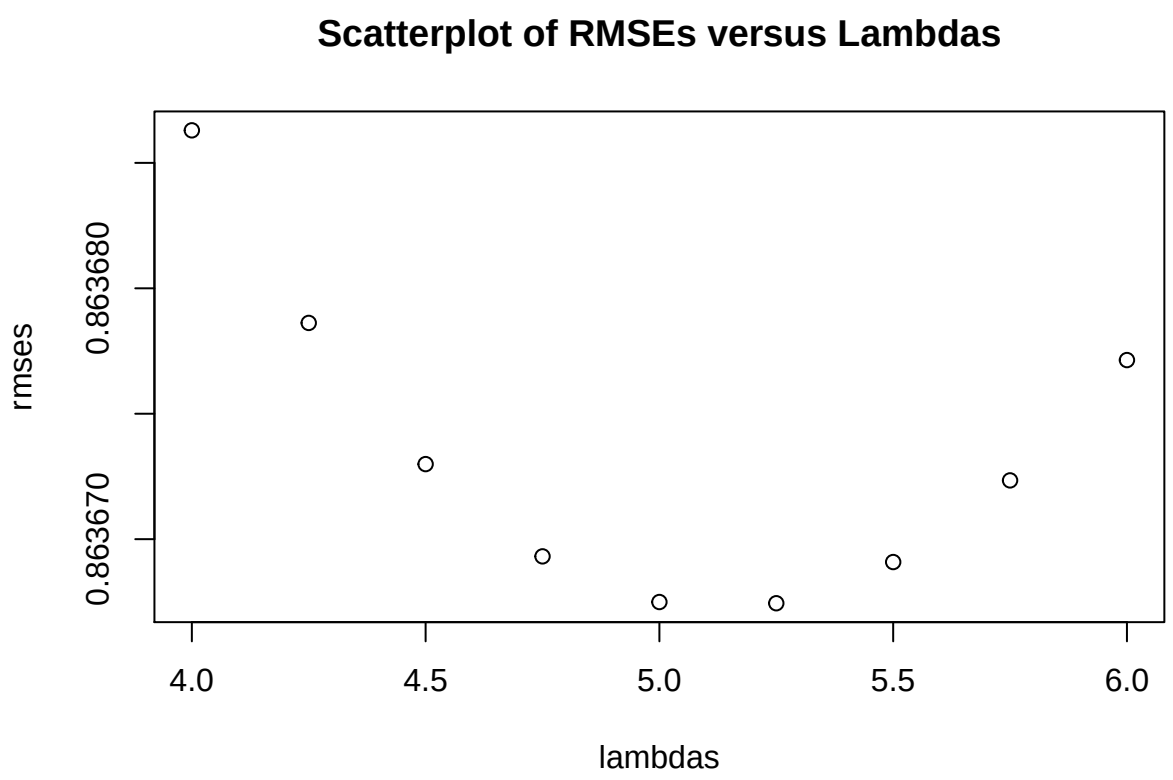
Some initial testing proved that the "edx" dataset is too large to run regression models on using the "train" function in the "caret" package. As a result, I decided to create a Regularized Movie Effects Model taking into account the movie, user, time, and genre effects on rating.

To generate my model, I first calculated the average rating in the "edx_train" dataset. Then, to generate the movie effect, I grouped the "edx_train" set by movieId and calculated the sum of the differences between each given rating and the average rating. I divided this sum by the total number of ratings in that particular movie grouping plus a constant, lambda. This approach ensures that movies with very few ratings do not have as much weight on the model as movies with lots of ratings.

I repeated this process three more times to generate the user, time, and genres effects, grouping by each variable respectively and calculating the sum of the differences between the mean and each rating for each group, divided by the number of ratings per group plus lambda.

Once I generated dataframes for the four different effects, I joined these dataframes onto the testing ("edx_test") set. I then found the average rating in the testing set and added the movie, user, time, and genre effect for each movie to the average rating to generate my predictions. I then calculated the RMSE of the model.

I repeated this process for different values of lambda (4 to 6, increasing in increments of 0.25). This approach, called cross-validation, ensures that I select the lambda the results in the lowest RMSE value for my final model. The lambda I selected for my final model was $\boxed{5.25}$. The plot below shows the RMSE for each lambda value.



Results

After I selected a lambda value of 5.25, I conducted predictions for the final testing ("validation") set using all of the original training ("edx") set. Based on the predictions given, the RMSE for my final model was $\boxed{0.8643044}$.

Conclusions

Brief Summary

This project took data on movies and ratings that users have given them and produced a model that predicts future movie ratings. The predictive model takes into account the effects that the specific movie, the individual user, the genre(s) of the movie, and the time of rating have on the rating of the movie in order to generate more accurate results. The model was trained on training data that provides ratings given different movies and users. After training on this data, a finalized model was selected and predictions were generated on the testing data. These predictions were compared to the true ratings in the testing data to determine the strength of the model.

Limitations and Future Work

The predictive quality of this model is fairly strong but is certainly subject to limitations. For example, the genre effect was calculated by grouping movies with exactly the same listings of genres together, instead of

by individual genres separately. In other words, movies that had exactly the same genres listed (either one or multiple) were grouped together. If I instead split up the genres in the "genres" column into separate words (i.e. split "Action|Adventure" into "Action" and "Adventure"), fewer genre groups would exist with more movies in them, making the predictive quality of the model stronger. This method could be implemented in a future version of the model.

This model can be useful for determining what ratings different movie viewers will leave movies. Given information on the movie IDs, user IDs, movie genre(s), and time of rating, companies can predict how their movies will be rated by their users. This can be useful for determining, for example, which movies will be popular across a wide range of users, which movies will be popular to small groups of users, and the general nature/sentiment of rating for different users (whether they tend to leave good or bad reviews overall).