

Movielens Rmd File

John Balzani

12/30/2019

Executive Summary:

How well can we predict how well someone will like a certain movie? The purpose of this project is to answer that by predicting the rating on a scale of 0.5 to 5 that users will give to movies. A model is tested which takes into account the impact of the individual movie, the impact of the individual user, and regularizes the ratings that are given to movies.

The dataset which this model is trained and tested on is the Movielens dataset, which is a dataset of 10 million ratings that users have given to movies. This dataset contains the following pieces of data: the userId of the user who rated the movie, the movieId of the move rated, the rating given to the movie, the title of the move, and the genres that pertain to that movie. About 9 million observations were used to train the model (the “edx” dataset), with the remaining 1 million (the “validation” dataset) being used only for model validation.

This project undertook the following key steps. First, the data was cleaned and explored. Second, the model was created. Third, the model was tested with Root Mean Square Error (RMSE) being used as the metric to judge how much the rating predicted by the model deviated from the actual rating given. It was found that the final model had an RMSE of .86511, meaning that the predicted movie ratings different from the actual ones by .86511 on average.

Methods and Analysis:

The edx dataset was briefly reviewed for NA values, outliers, and duplicates. It was found that there are no NA data points. After checking for duplicates, it was found that there are no duplicate values.

```
## [1] userId    movieId    rating    timestamp title      genres
## <0 rows> (or 0-length row.names)
```

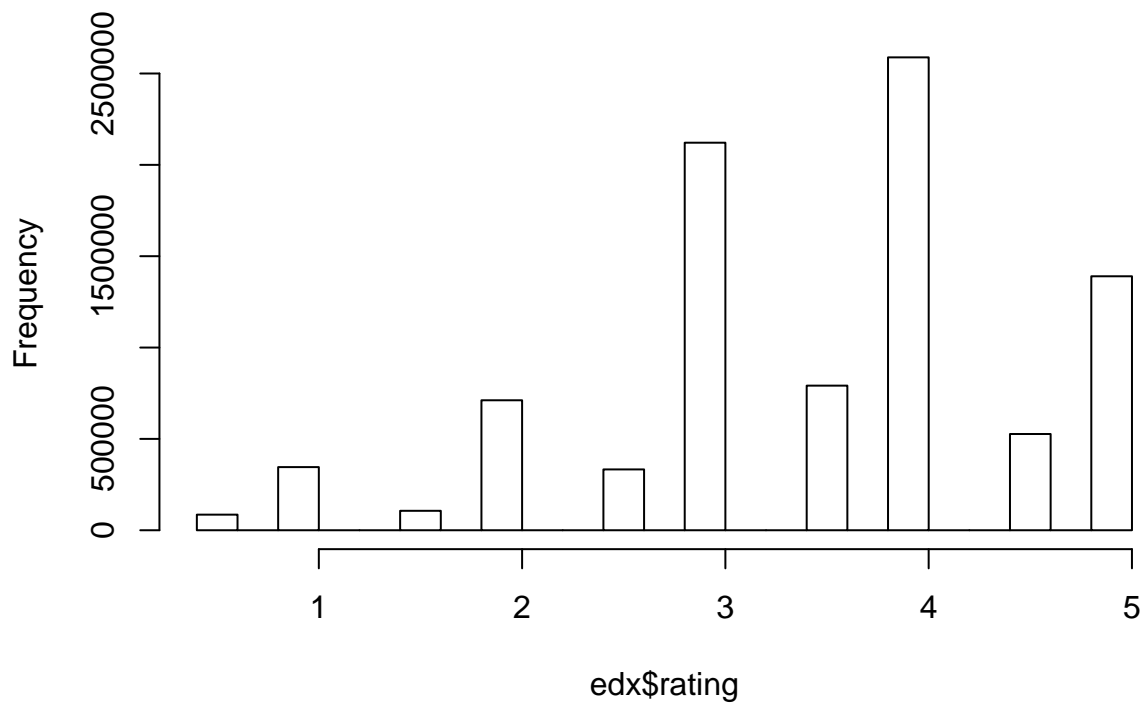
In order to check for outliers and to get a better sense of the data, a boxplot was generated of the ratings (Figure 1). Boxplots were not generated for the other variables as they would not be particularly informative. The maximum and minimum values of rating were also found, to ensure that they were within the limits of what users could rate movies at.

```
#{r boxplot of rating: look for outliers} #check for outliers #boxplot(edx$rating, main
= "Figure 1: Boxplot of Ratings", ylab = "Rating") #
```

When exploring the data, it can be noticed that the ratings vary from 0.5 to 5, with a rating of 4 being most common, followed by a rating of a 3, and then a rating of a 5 (Figure 1). It seems that positive ratings (4 or 5) are more common than very negative ratings such as 2 or below. The average rating stands at 3.5124652, while the median rating is 4. The ratings have a standard deviation of 1.0603314.

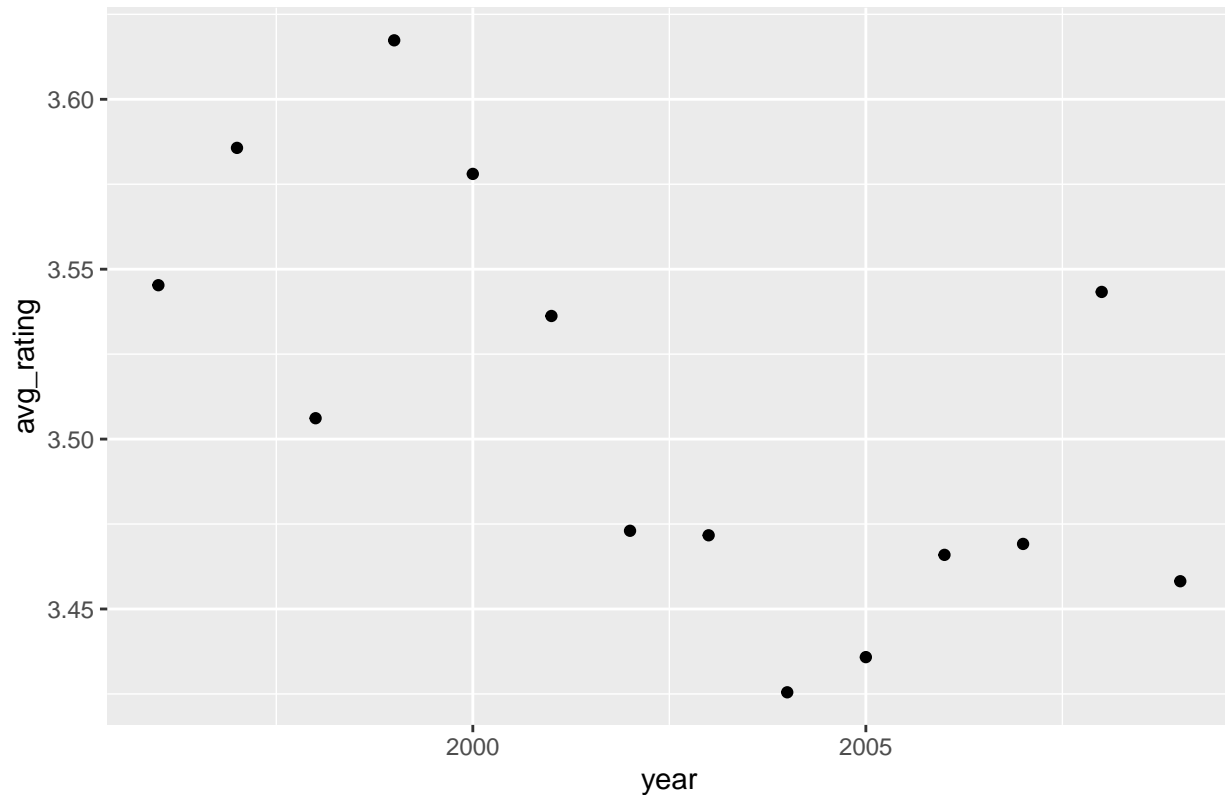
A histogram of the ratings was generated, in order to better visualize the ratings distribution (Figure 2).

Figure 2: Histogram of Ratings



In order to explore the timestamp variable, the timestamp was converted into a date. We can see that the movie ratings were made between January of 1995 and January of 2005, which is a long period of time. In order to see whether there is any relationship between the rating and the time of the rating, as was seen in the Netflix dataset, the date of the movie review was plotted against the rating (Figure 3) (Koren, 2009). 1995 was removed since only 2 reviews were made during this year and the average rating of 4 was much different than the ratings for other years. From this, we can see that the average rating fluctuates by year, with a possible slight decline overall.

Figure 3: Rating vs. Time of Rating



In order to explore whether or not there is any relationship between the genre of the movie and its rating, the average rating for each genre was calculated. It appears that there is a genre impact, as some genres are rated higher than others, with the highest rated genre being Crime|Mystery|Thriller with an average rating of 4.20 and the lowest genre being Adventure|Children|Drama with an average rating of 2.79.

```
## # A tibble: 10 x 3
##   genres                                n avg_rating
##   <chr>                                <int>     <dbl>
## 1 Crime|Mystery|Thriller                26892      4.20
## 2 Action|Adventure|Comedy|Fantasy|Romance 14809      4.20
## 3 Adventure|Drama|Film-Noir|Sci-Fi|Thriller 13957      4.15
## 4 Adventure|Drama|War                    14137      4.08
## 5 Crime|Horror|Thriller                  33757      4.08
## 6 Crime|Film-Noir|Mystery|Thriller        24961      4.06
## 7 Comedy|Crime|Drama|Thriller             24341      4.06
## 8 Adventure|Mystery|Thriller              14712      4.04
## 9 Comedy|Drama|Romance|War                41762      4.01
## 10 Crime|Drama|Sci-Fi|Thriller             10730      4.00
```

```
## # A tibble: 10 x 3
##   genres                                n avg_rating
##   <chr>                                <int>     <dbl>
## 1 Adventure|Children|Drama              10144      2.79
## 2 Action|Crime|Sci-Fi                   17331      2.82
## 3 Comedy|Horror                          37394      2.84
## 4 Comedy|Thriller                        13558      2.87
```

##	5	Children Comedy Fantasy	18784	2.88
##	6	Horror	68738	2.88
##	7	Action Horror Sci-Fi	10364	2.91
##	8	Horror Sci-Fi	31281	2.92
##	9	Children Comedy	63483	2.92
##	10	Sci-Fi	10125	2.93

If someone were to ask me to predict how much a movie rating would differ from the average rating, my first question would be “What is the movie?” For this reason, I thought it made sense to start with a model that looks at the impact that the specific movie has on the rating, since of course good movies will get better reviews than bad ones. The algorithm is: subtract the rating for movies with that particular feature from the mean rating, and take the mean of that difference as the impact of the feature. Then use that feature impact as a predictor. Next, repeat the process with the residuals of the mean and all other feature impacts already calculated from the algorithm. In this case, the feature impact we are looking at is the individual movie impact, so we subtract the rating for the individual movie from the mean rating, and then we take the mean of that.

If I had to predict how a person would rate a movie knowing the movie title, I would then want to know more about the rater and how they rate other movies. For this reason, the expected deviation from the mean rating due to the user was included next in my model.

Next I checked to see if regularization would improve my results. As Dr. Irizarry stated in his lectures, movies with few ratings are more likely to have large deviations from the mean (Irizarry, 2019). These extreme ratings given by only a few people are not representative of the rating that would be given by a larger number of people, and regularization can correct for this by penalizing ratings done by only a few people.

```
## Loading required package: Metrics

##
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':
##
##   precision, recall
```

The RMSE before regularization is 0.8565338. I believe a lower RMSE can be reached.

From observing the vector, it can be seen that 0 is the lowest alpha. I then look closer to see if there is an optimal alpha close to zero.

```
## [1] 0.8565338 0.8566076 0.8568072 0.8570850 0.8574159 0.8577836 0.8581767
## [8] 0.8585878 0.8590113 0.8594429 0.8598796
```

It appears that the RMSE is only slightly reduced by regularization on the training set. This model gives an RMSE on the **training** set of 0.8565324. This is a relatively low RMSE, so it's time to try my luck on the test set. For example, the winners of the Netflix competition, who were tasked with improving Netflix's recommendation algorithm, had an RMSE of .8713 on their winning algorithm (Koren, 2009). The final model is Predicted Rating = mean_edx + regul_spec_movie_dev + regul_user_dev + epsilon, where regul_spec_movie_dev is the regularized specific movie impact, regul_user_dev is the regularized user impact, and epsilon represents the residuals.

Results:

The RMSE of the final model was 0.8651118, meaning that the predicted rating deviated from the actual rating by 0.8651118 on average in the test set.

This is a relatively low RMSE. Again, the winners of the Netflix competition had an RMSE of .8713 on their winning algorithm (Koren, 2009).

Below is a histogram of the predicted ratings, with a histogram of the actual ratings below it.

Figure 4: Histogram of Predicted Ratings

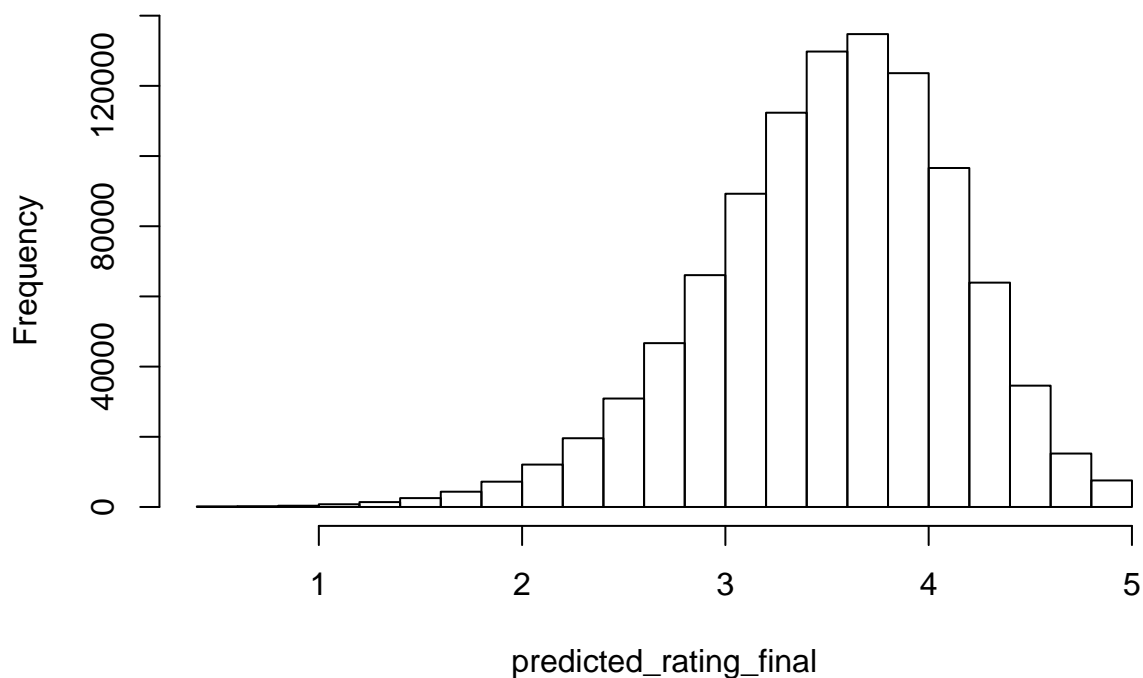
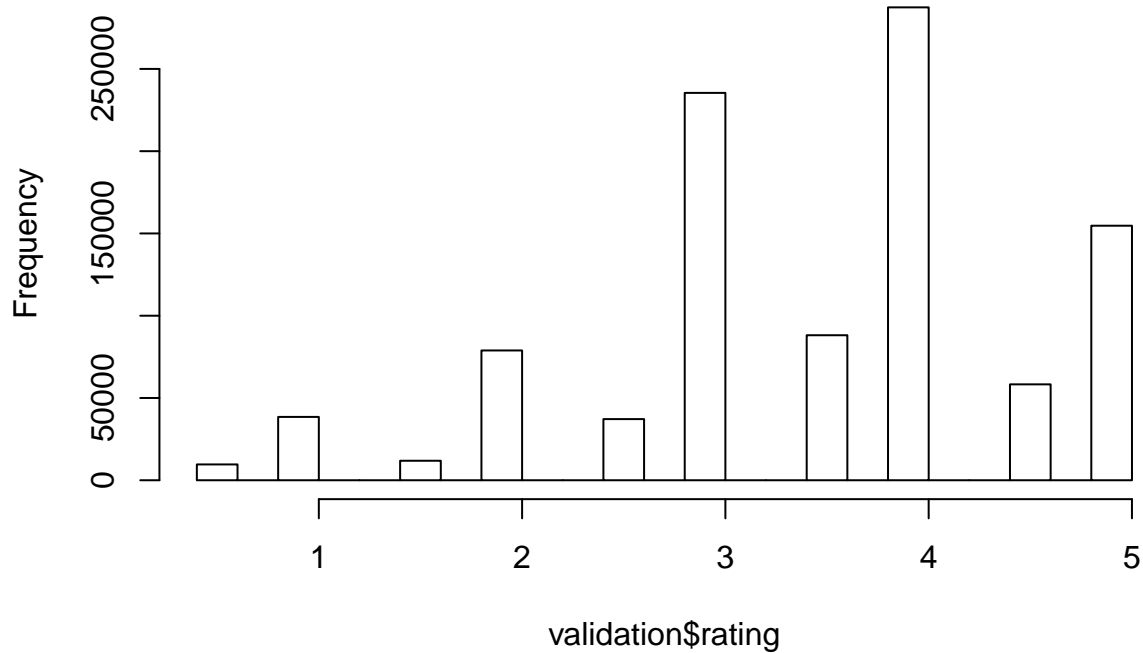


Figure 5: Histogram of Actual Ratings



Conclusion:

A model of rating prediction based on the regularized specific movie impact and the regularized user impact was developed using the edx portion of the Movielens dataset, and this model was tested on the validation data set. This model was found to have an RMSE of 0.8651118.

While this RMSE is acceptable, there are limitations to this model. This model's predictions are not rounded to intervals of 0.5, such as the actual ratings are. As the project was only to provide predictions, and a prediction of 3.8, for example, is still very valuable, the predictions were not rounded. However, this would be an interesting area for future work. Another limitation of this model is that it does not take into account the genre of the movie or the time the rating was made. A potential area for future work involves incorporating these features into a model to further reduce RMSE.

References:

- Irizarry, Rafael. Building the Recommendation System, n.d. ———. Regularization, n.d.
- Koren, Yehuda. "The BellKor Solution to the Netflix Grand Prize," 2009.