MovieLens Project

Harry Terris

2019-03-29

Introduction

In this project, we create a movie recommendation system using the 10M version of the MovieLens dataset (https://grouplens.org/datasets/movielens/10m/).

As noted on the GroupLens site in the page accessed using the link above, the dataset is a stable benchmark comprising 10 million ratings of 10,000 movies by 72,000 users. The dataset was released in January 2009.

The objective is to create an algorithm that predicts the rating that users will give a specific movie. We target a root mean square error (RMSE) of less than or equal to 0.8775.

We will use a training set and a validation set created with code given in the course materials.

The main steps are as follows:

- Load previously wrangled data objects and analyze them
- Train an algorithm and test it against the validation set
- Review results

Methods and Analysis

This project uses the following libraries:

```r
library(tidyverse)
library(caret)
```

(The caret package was used in the creation of the training and test datasets.)

We load the previously wrangled training set and validation set.

```r
load("rdas/edx.rda")
load("rdas/validation.rda")
```

The validation set is about 10% of the original dataset.

We start with a naive model that assumes the same rating for all movies regardless of user, with differences attributed to random variation. The estimate that minimizes RMSE in this model is the mean of all ratings. We compute an estimate using the training set:

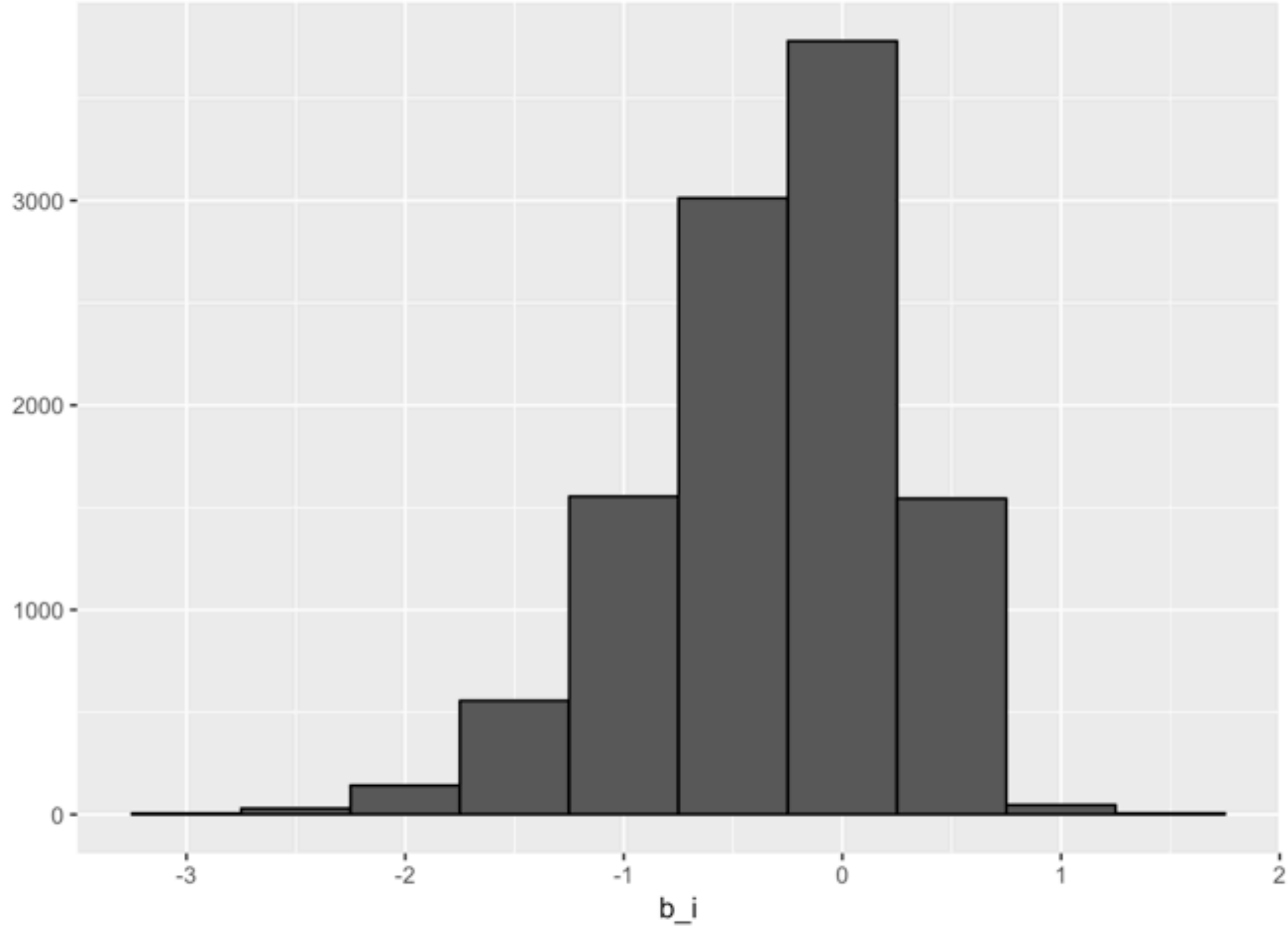
```r
mu <- mean(edx$rating)
mu
```

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

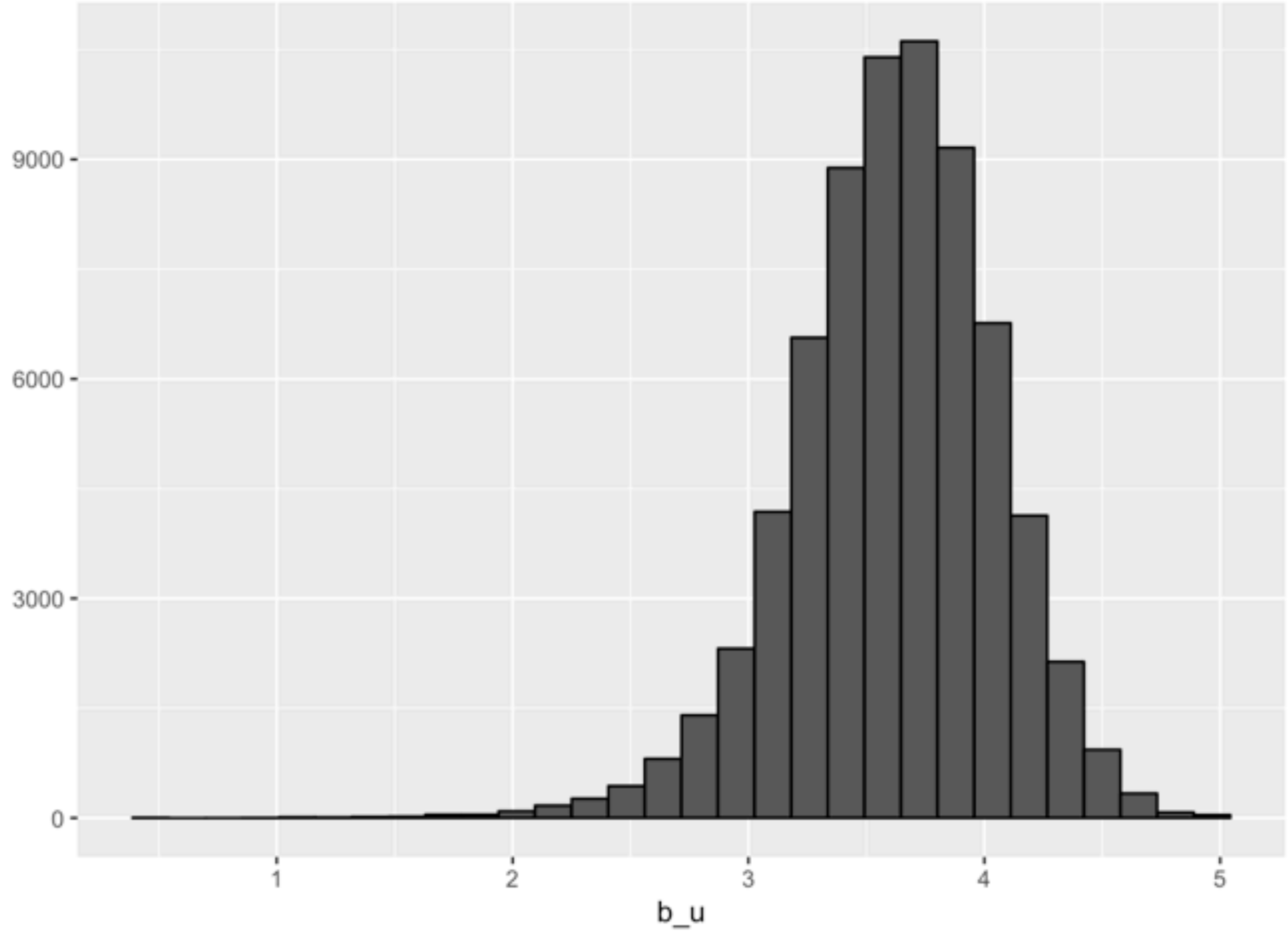
Next, we consider variation attributable to the tendency for some movies to be rated higher than others. We compute the average amount by which ratings for a particular movie in our training set differ from our estimate for the mean:

```
m_effect <- edx %>% group_by(movieId) %>% summarize(b_i = mean(rating - mu))
```

A histogram shows that these average differences vary widely among movies:



We also consider variation attributable to individual users, some of whom tend to give higher ratings generally than others. This variation can be seen in a histogram of the average rating given by each user:



We estimate the user effect for each user with the following code:

```
u_effect <- edx %>% left_join(m_effect, by = "movieId") %>% group_by(userId) %>% summ
arize(b_u = mean(rating - b_i - mu))
```

The foregoing analysis suggests two additional models beyond the naive approach described above. The first augmentation accounts for movie-specific effects by predicting a particular user’s rating for a particular movie as the sum of an estimate for the mean rating across all users and movies, and estimates of the amount by which the average rating for a particular movie varies from that mean. Here is code that generates predictions for the validation set under this model:

```
predicted1 <- validation %>%
  left_join(m_effect, by = "movieId") %>%
  mutate(pred1 = mu + b_i) %>%
  .$pred1
```

A second augmentation accounts for user-specific effects (the tendency for one user to be more generous with movie ratings overall than another user) in addition to movie-specific effects. Here is code that generates predictions for the validation set under this model:

```
predicted2 <- validation %>%
  left_join(m_effect, by = "movieId") %>%
  left_join(u_effect, by =  "userId") %>%
  mutate(pred2 = mu + b_i + b_u) %>%
  .$pred2
```

Results

We are targeting a RMSE of less than or equal to 0.8775. Here is a function that computes RMSE:

```
RMSE <- function(actual, predicted){sqrt(mean((actual-predicted)^2))}
```

The model that accounts for movie effects, but not user effects, does not meet our target:

```
RMSE(validation$rating, predicted1)
```

```
## [1] 0.9439087
```

The model that accounts for movie effects and user effects, does meet our target:

```
RMSE(validation$rating, predicted2)
```

```
## [1] 0.8653488
```

Conclusion

Variation in ratings attributable to the tendency for some movies to generally be rated higher than others, and the tendency for some users to generally be more generous with ratings than other users, is considerable.

Adopting a model designed to capture both effects, and training it using a relatively large sample of about 900,000 ratings, generated predicted ratings for a validation set that achieved a RMSE within our target of less than or equal to 0.8775.

We may be able to improve accuracy further with additional time and computing power by considering genre and time effects, and applying approaches like regularization. But movie effects and user effects are powerful, and enabled us to clear an initial loss function target.