Recommendation System for Movielens

rene_hiroki

2019/05/27

1. Introduction

In this report, we will build a recommendation system for the MovieLens 10M dataset that is provided by Grouplens(https://grouplens.org). You can download the dataset we use from this *link*. **edx** is train set, **validation** is test set. Then, glance at the dataset structure:

```
glimpse(edx)
```

```
## Observations: 9,000,055
## Variables: 6
## $ userId    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ movieId   <dbl> 122, 185, 292, 316, 329, 355, 356, 362, 364, 370, 37...
## $ rating    <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5...
## $ timestamp <int> 838985046, 838983525, 838983421, 838983392, 83898339...
## $ title     <chr> "Boomerang (1992)", "Net, The (1995)", "Outbreak (19...
## $ genres    <chr> "Comedy|Romance", "Action|Crime|Thriller", "Action|D...
```

```
glimpse(validation)
```

```
## Observations: 999,999
## Variables: 6
## $ userId    <int> 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5...
## $ movieId   <dbl> 231, 480, 586, 151, 858, 1544, 590, 4995, 34, 432, 4...
## $ rating    <dbl> 5.0, 5.0, 5.0, 3.0, 2.0, 3.0, 3.5, 4.5, 5.0, 3.0, 3....
## $ timestamp <int> 838983392, 838983653, 838984068, 868246450, 86824564...
## $ title     <chr> "Dumb & Dumber (1994)", "Jurassic Park (1993)", "Hom...
## $ genres    <chr> "Comedy", "Action|Adventure|Sci-Fi|Thriller", "Child...
```

We can see that how many observations and variables are in the datasets, and also see that what data types they are.

The purpose of our recommendation system is to predict what rating a particular user will give a specific movie. Because, movies for which a high rating is predicted for a given user are then recommended to that user. We define a loss function by RMSE to evaluate our models. Our goal is minimizing RMSE as possible as we can.

In Analysis section, we analyze the data and build machine learning models by following processes,

1. Define a loss function by RMSE
2. Build a Simplest Model
3. Build a Movie Effects Model
4. Build a Movie and User Effects Model

Then, We will evaluate our models in Result section and choose the best model for recommendation system in Conclusion section.

2. Analysis

2.1. Define loss function by RMSE

We use Root Mean Squared Error (RMSE) as a loss function. We define $y_{u,i}$ as the rating for movie i by user u and denote our prediction with $\hat{y}_{u,i}$. The RMSE is then defined as:

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

with N being the number of user/movie combinations and the sum occurring over all these combinations.

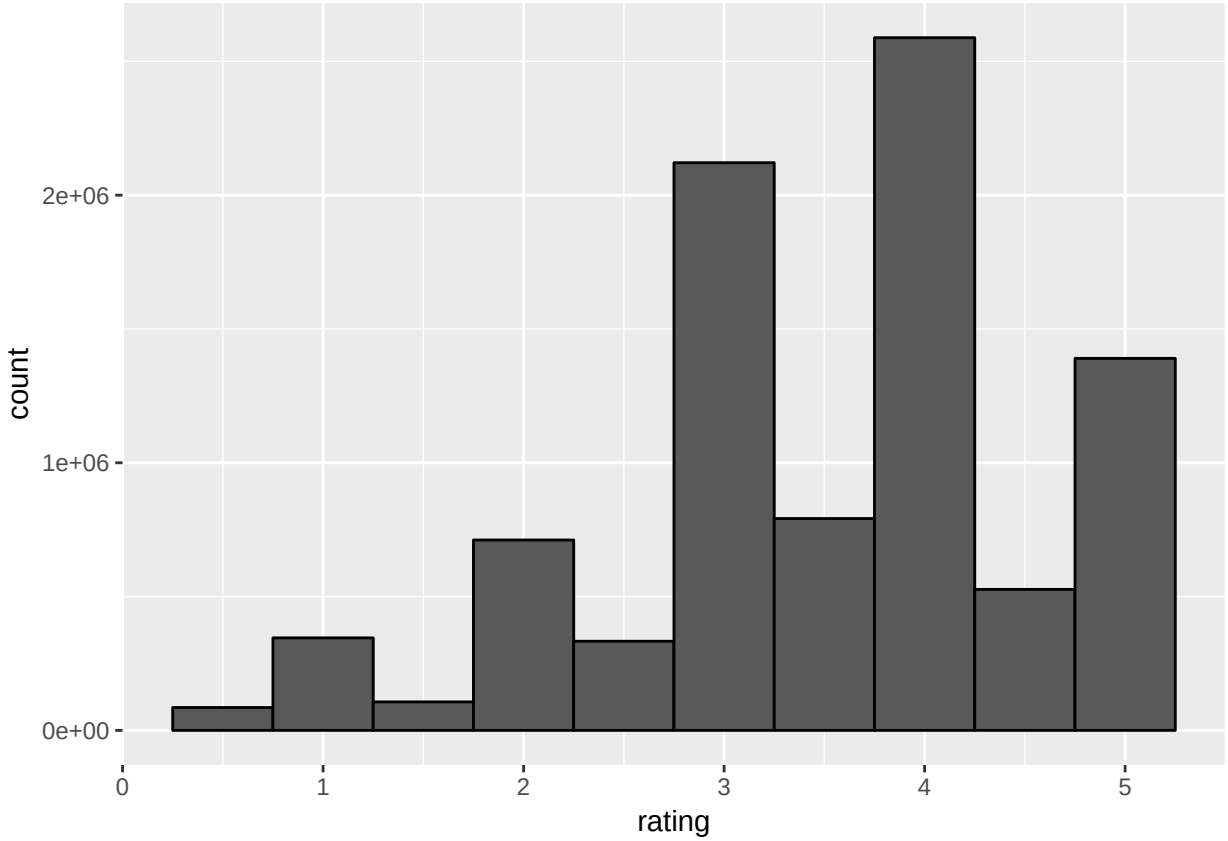
2.2. Build a Simplest Model

Our first model is the simplest. We predict the same rating for all movies regardless of user. A model we assume is like this:

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

with $\epsilon_{u,i}$ independent errors sampled from the same distribution centered at 0 and μ the "true" rating for all movies. We estimate $\hat{\mu}$ by taking average of all rating. Then, calculate the RMSE of this model.

Let's look at the distribution of rating.



We can see that 4 is most counted and 3 is the second. Form histogram, average rating would fall on the between 3 and 4.

Then, estimate average rating $\hat{\mu}$ by taking average.

$\hat{\mu} = 3.5124652$

Using this estimate $\hat{\mu}$, we predict $y_{u,i}$ and calculate RMSE.

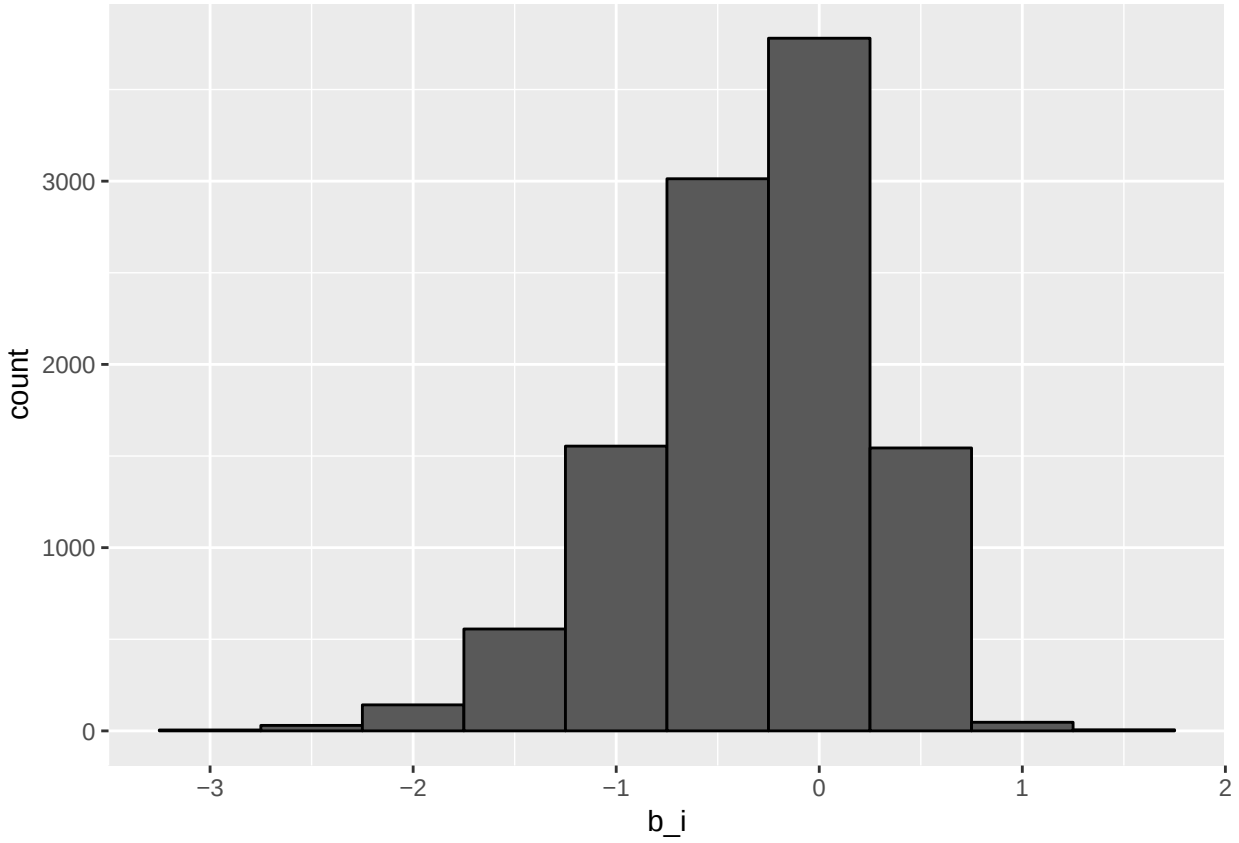
RMSE = 1.0612018

2.3. Build a Movie Effects Model

We know from experience that each movie's rating has propensity for high or low rating. We can augment our previous model by adding the term b_i to represent average rating for movie i:

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

We estimate $\hat{b_i}$ by taking average of $Y_{u,i} - \mu$ for each movie i. Let's see the histogram of b_i.

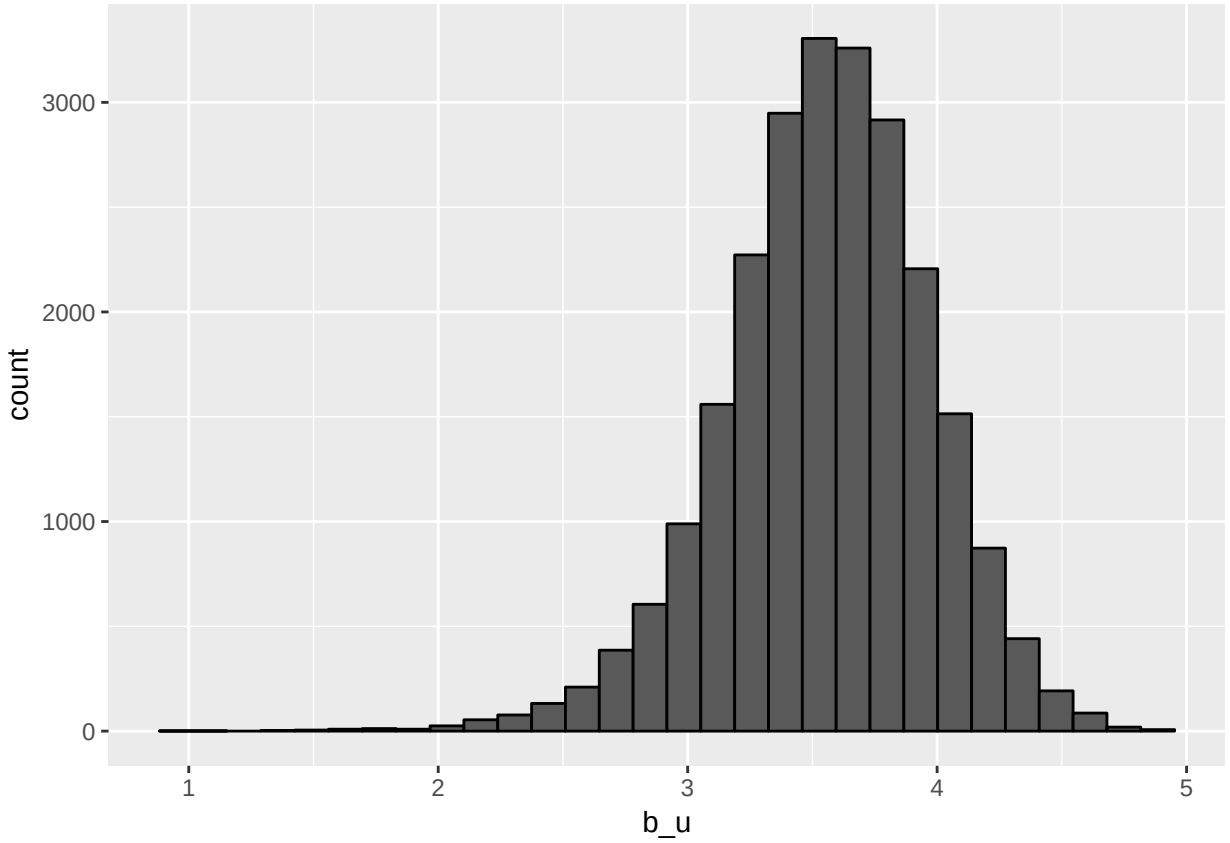


We can confirm that movie bias is exist from this histogram. Thus, our movie effects model should work. Let's predict rating and calculate RMSE.

RMSE = 0.9439087

2.4. Build a Movie and User Effects Model

As well as movie effects, there would be user effects. To confirm that, let's see the histogram of the average rating for user u for those that have rated over 100 movies:



We can see that some users give salty rate and others love every movie. This implies that we can augment our previous model farther. We put b_u on the model to represent average rating for user u:

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

We estimate $\hat{b_u}$ as the average of $Y_{u,i} - \mu - \hat{b_i}$. Then, predict rating and calculate RMSE.

RMSE = 0.8653488

3. Result

method	RMSE
Simplest Model	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8653488

We built three models “simplest model”, “Movie Effect Model”, and “Movie and User Effects Model”. The performances of these models are shown above table. As you can see, “Movie and User Effects Model” has the best performance, **RMSE = 0.8653488** .

4. conclusion

We built the recommendation system by using “Movie and User Effects Model”.

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

RMSE = 0.8653488 means that, on average, we predict rating with the error 0.8653488, that’s not bad.