Movie Ratings Project

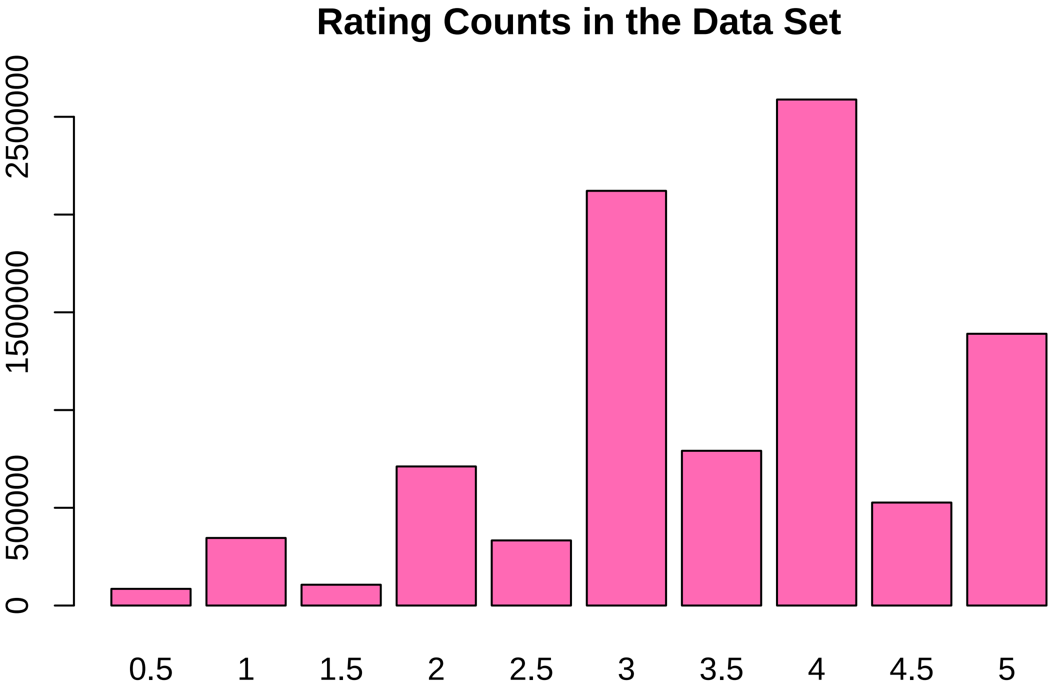
Jason Holland

5/9/2020

Introduction

This project utilizes a data set of approximately 10 million movie ratings. The full data set is available at https://grouplens.org. The goal of the project is to develop a model to predict movie ratings using the available data. The full code for getting the data is omitted from this report but is available in the file MovieRatingHolland.Rmd. The steps involved in obtaining our predictions include splitting the data into training and test sets, modeling user and movie effects using the training set, adding in a parameter to help control the variability of the effects using the training set, and computing the final *Root Mean Squared Error* (RMSE) for the test set (validation set). Before doing so, we look at some summaries of the data.

In the plot below, we look at the number of ratings given for each rating value. Note that the values start at the lowest rating of .5 and end at the highest rating of 5. Note also that ratings of .5, 1.5, 2.5, 3.5 and 4.5 are less likely than integer valued ratings.

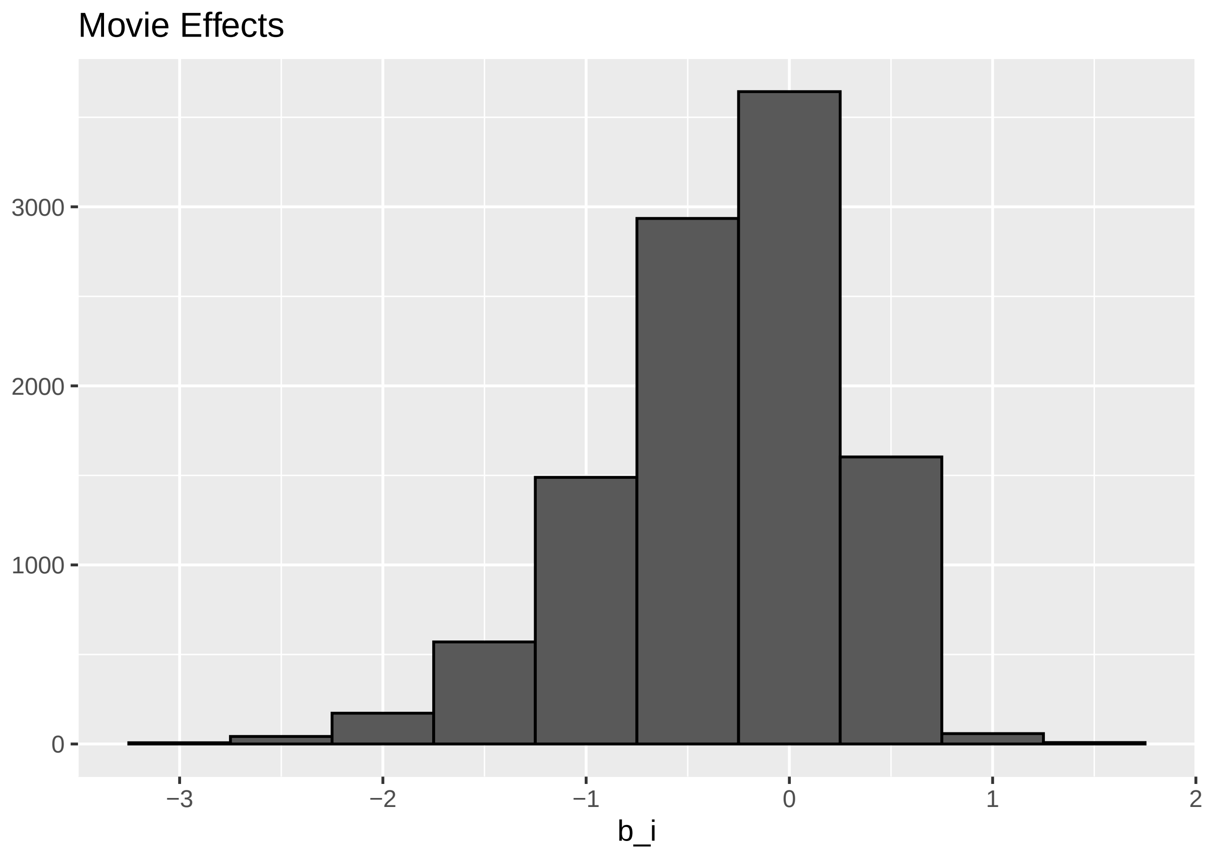


It is interesting to see the top six movies by number of ratings. Pulp Fiction (1994) heads the list with over 31,000 ratings.

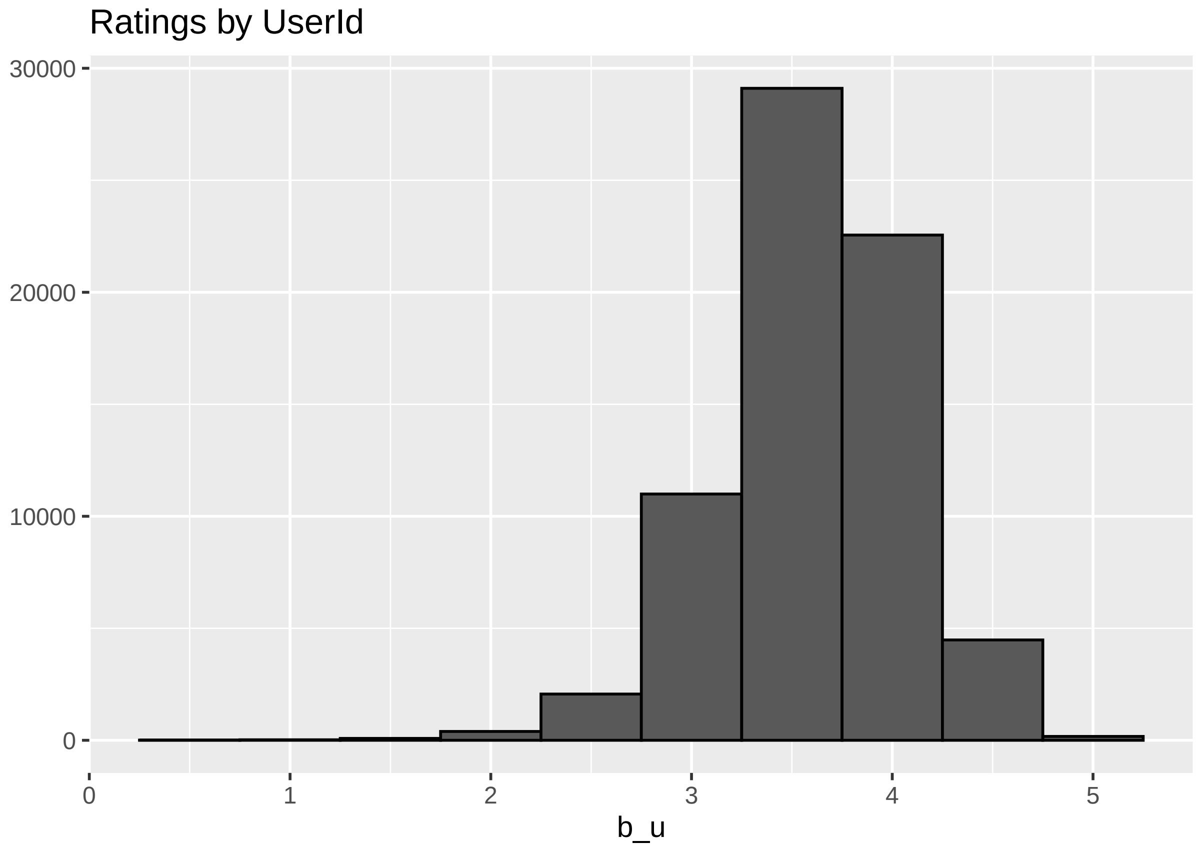
```
## # A tibble: 6 x 3
## # Groups:   movieId [6]
##   movieId title                             count
##     <dbl> <chr>                             <int>
## 1     296 Pulp Fiction (1994)               31362
## 2     356 Forrest Gump (1994)               31079
## 3     593 Silence of the Lambs, The (1991) 30382
## 4     480 Jurassic Park (1993)              29360
## 5     318 Shawshank Redemption, The (1994) 28015
## 6     110 Braveheart (1995)                 26212
```

Methodology

The data is first divided into a training set called edx and a validation set. The training set will be used to build our model and in the end the RMSE will be calculated using the validation set. Our method will be to divide the edx set into two sets: edx_train and edx_test. To do this we use the caret package to partition the sets. The code is not shown here but is included in the file MovieRatingHolland.Rmd. Care must be taken to ensure that movie ID's and user ID's are in both sets so we use appropriate joins to accomplish this. We gain insight by looking at histograms of two predictor variables. The first plot we look at involves the effects of the *movieId* variable.



We see that there is a good bit of variability in the effects of the *movieId* variable. Given that μ is about 3.5, we see that if b_i (the added effect of movie i) approaches its maximum value, then $\mu + b_i$ would approach a perfect rating of 5. In the next plot, we look at the user effects (denoted by b_u) by examining the distribution of ratings grouped by the variable *userId*. We see that there is quite a bit of variability in the following plot also.



Given the variability of b_i for and b_u, our approach will be a model of the type

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

We will do this in four steps:
1. Examine the RMSS with μ.
2. Examine the RMSS with μ plus movie effects.
3. Compute a third model and examine the RMSS using movie effects, user effects, and a parameter λ to account for variability of the effects.
4. Compute the final RMSS value using the validation data set.

Results

In this section we show the RMSE's for our three models and then show the final RMSE calculation using the validation set. We only show the code for the last model for ease of reading. The code for each model is available in the file MovieRatingHolland.Rmd.

Model 1; Predicting the mean.

In Model 1, we predict that the rating will just be the mean of all ratings. The mean of all ratings is approximately $\mu = 3.5$. This results in the following RMSE.

```
## [1] "The RMSE for model 1 is  1.06041 ."
```

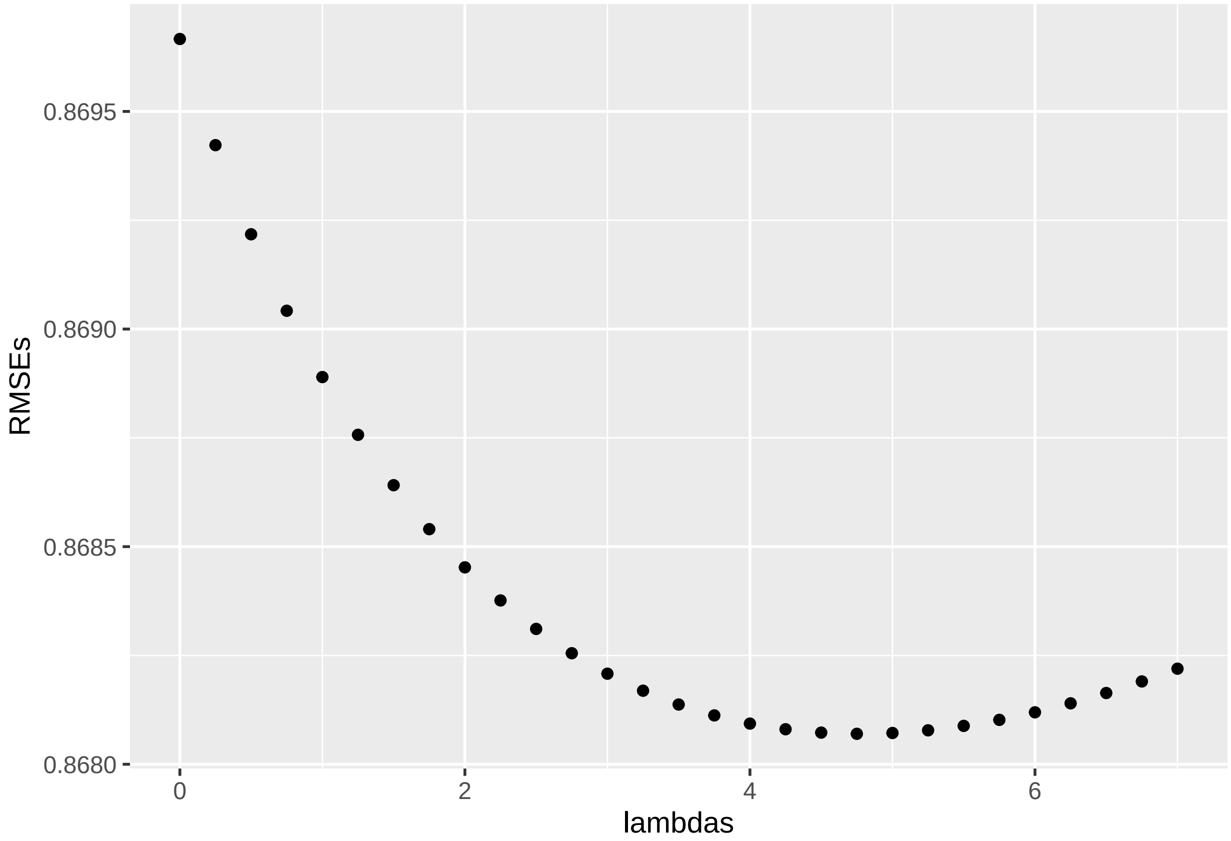
Model 2; Predicting with the Mean and Movie Effects.

In Model 2, we include a correction for the effects of the *movieId* variable. The more movie ratings a particular movie gets, the higher the rating in general.

```
## [1] "The RMSE for model 2 is  0.94413 ."
```

Model 3; Mean, Movie Effects, User Effects, and Correction

Before showing the code and results of the final model, we calculate a tuning parameter λ which corrects for the variability of the effects. We follow closely the techniques covered in section 33.9.2 of `https:` //rafalab.github.io/dsbook. We see in the following plot that the value of λ that minimizes RMSE is around 4.5.



We compute the minimum λ and obtain

```
## [1] "The minimum value is 4.75 ."
```

```
## [1] "The RMSE for this lambda is 0.8681 ."
```

Armed with this value for λ, we compute the final model using the entire test set, then check the RMSE on the validation set. The code is given for this process.

```
mu <- mean(edx$rating)  # We use the full edx set for mu.
# The b_i's are needed for movie effects.
b_i <- edx %>% group_by(movieId) %>% # full edx for b_i
  summarize(b_i = sum(rating-mu)/(n()+4.75))
# The b_u's are needed for user effects. Full edx for b_u
b_u <- edx %>% left_join(b_i, by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - mu)/(n()+4.75))

predicted_ratings <- validation %>%   #Compute predictions and add
  left_join(b_i, by = "movieId") %>%  # to validation set.
  left_join(b_u, by = "userId") %>%
  mutate(pred = mu + b_i + b_u) %>%
  pull(pred)

FINAL_RMSE <- (sqrt(mean((validation$rating - predicted_ratings)^2)))
paste("The validation RMSE is",round(FINAL_RMSE,4))
```

```
## [1] "The validation RMSE is 0.8648"
```

Conclusion

We summarize the results of our models in a data frame reporting RMSE. We use the name *Model1* for the *mean only* model. We use *Model2* for the *mean plus movie effects* model. We will denote the third model with RMSE calculated on the training data by *Model3*. Finally, we use *ModelF* to denote the model that we retuned using the whole training set, and then checked on the validation set.

```
## # A tibble: 4 x 2
##   Model   RMSE
##   <fct>  <dbl>
## 1 Model1 1.06
## 2 Model2 0.944
## 3 Model3 0.868
## 4 ModelF 0.865
```

This model is limited by only using two predictors variables. Future improvements could include incorporating the genre, the year of release, and the year of the rating. One could also use matrix factorization to hopefully achieve more accurate results.