MovieLens Project Report

Eric Mestiza

4/24/2019

Note: Report begins below the R code output.

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────────────────────
──────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.1.1       ✔ purrr   0.3.2
## ✔ tibble  2.1.1       ✔ dplyr   0.8.0.1
## ✔ tidyr   0.8.3       ✔ stringr 1.4.0
## ✔ readr   1.3.1       ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────────────
─────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
## Loading required package: caret
```

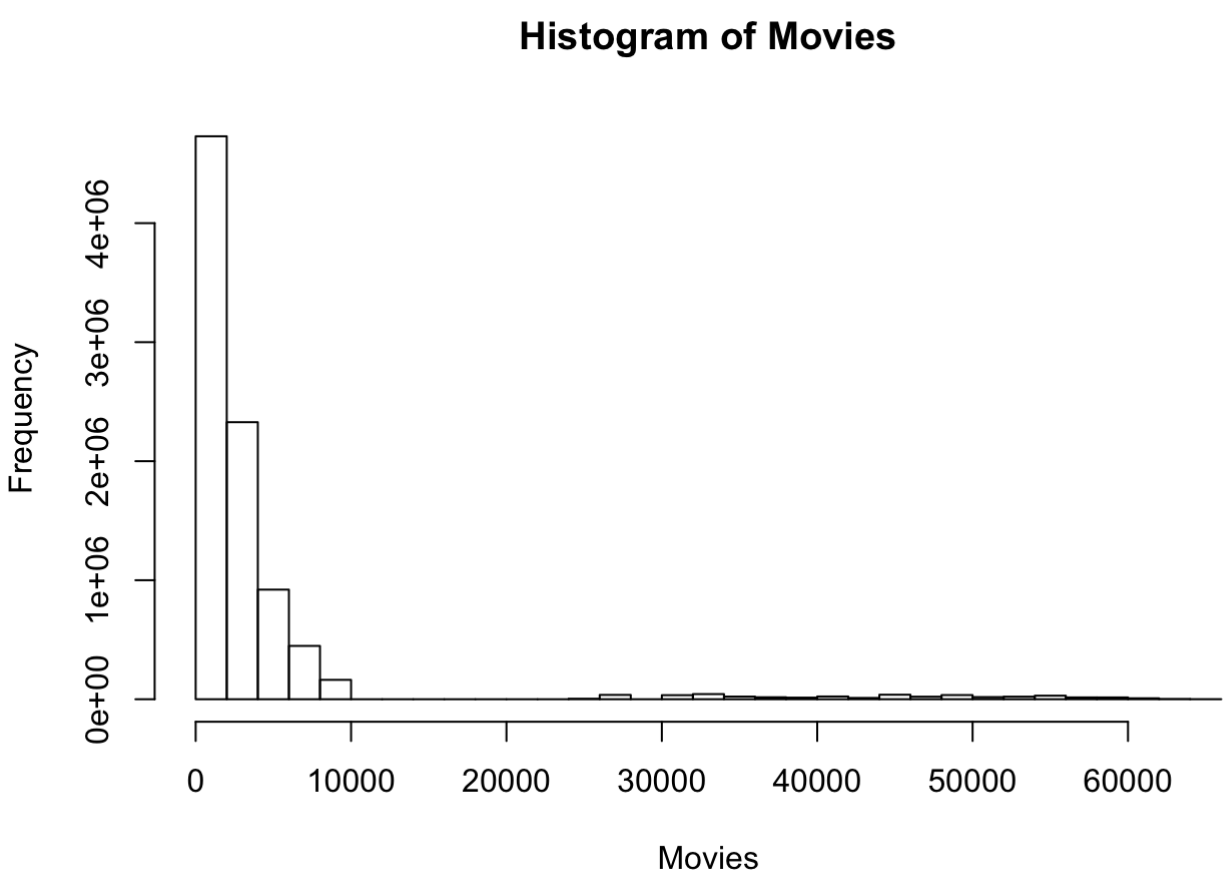
```
## Loading required package: lattice
```

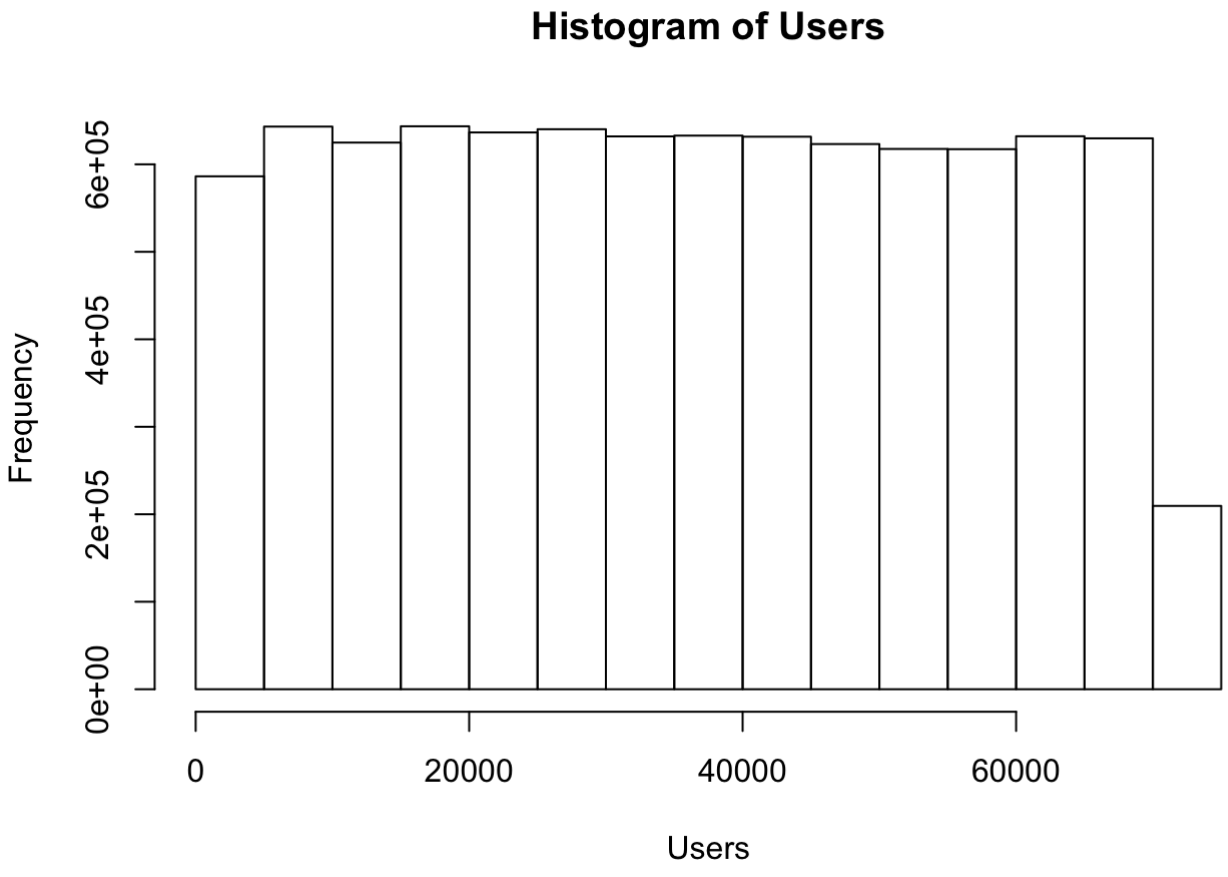
```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror
```

```
## Joining, by = c("userId", "movieId", "rating", "timestamp", "title", "genres")
```


Histogram of Movies
Frequency
0e+00
1e+06
2e+06
3e+06
4e+06
0
10000
20000
30000
40000
50000
60000
Movies


Histogram of Users
Frequency
0e+00
2e+05
4e+05
6e+05
0
20000
40000
60000
Users

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

```
## [1] 1.060331
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

method	RMSE
Edx Dataset	1.0603313
Movie Effect Model	0.9423475

method	RMSE
Edx Dataset	1.0603313
Movie Effect Model	0.9423475
Movie + User Effects Model	0.8567039

```
## [1] 3.512033
```

```
## [1] 1.061202
```

method	RMSE
Validation Dataset	1.0612017
Movie Effect Model	0.9383091

method	RMSE
Validation Dataset	1.0612017
Movie Effect Model	0.9383091
Movie + User Effects Model	0.8251770

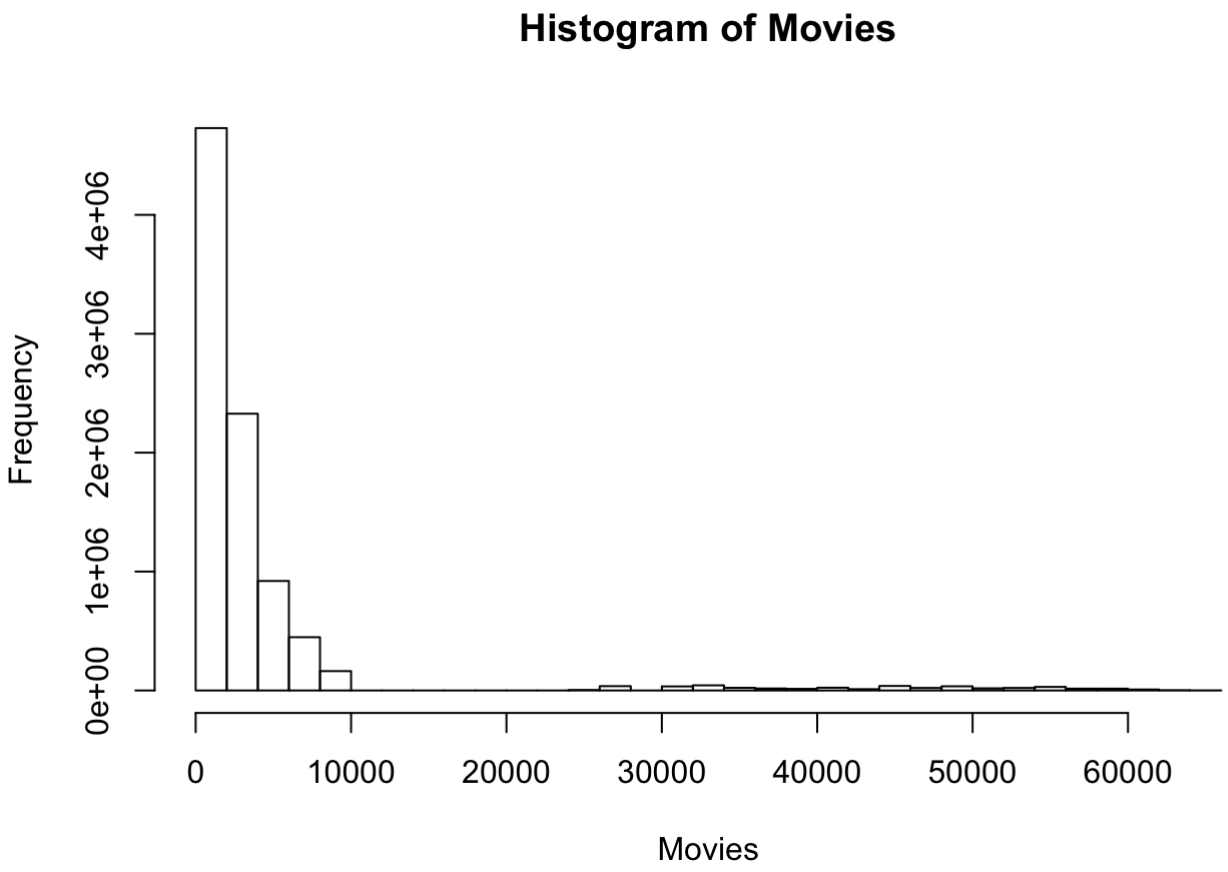
I. Introduction

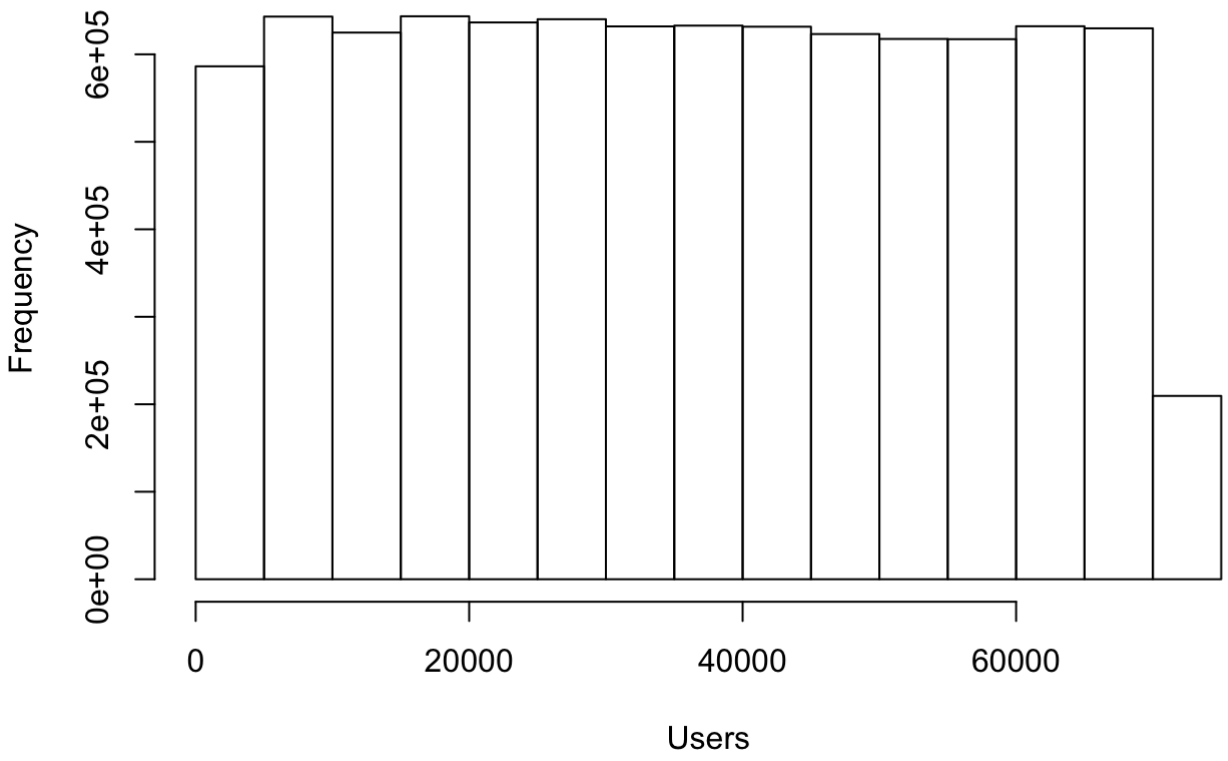
The goal of the MovieLens project is to create a movie recommendation system using the MovieLens dataset. The GroupLens research lab generated their own database with over 20 million ratings for over 27,000 movies by more than 138,000 users. The column headers in the MovieLens dataset are movieId, title, year, genres, userId, rating, and timestamp. The MovieLens table is in tidy format where each row represents a rating given by one user to one movie. There are movies that were not watched or not rated as well. The purpose of the recommendation system is to help fill in this missing information. R code was used to generate the edx and validation datasets where the validation dataset is 10% of MovieLens dataset. An algorithm to predict movie

ratings was developed and trained using the edx dataset. For a final test of the algorithm movie ratings were predicted in the validation set as if they were unknown. The residual mean squared error (RMSE) was used to evaluate how close the predictions are to the true values in the validation set.

II. Methods and Analysis

In this model each outcome y has a different set of predictors. If predicting the rating for movie i by user u, then all other ratings related to movie i and by user u can be used as predictors. However, different users rate a different number of movies and different movies. It is possible to use information from other movies that are similar to movie i or from users similar to user u. Thus, the entire matrix can be used as predictors for each cell. One observation is that some movies get rated more than others. A second observation is that some users are more active than others at rating movies. The movies and users distributions are shown below.





Similar to other machine learning algorithms a training dataset named edx is created to assess the accuracy of the models implemented. To compare different models the residual mean squared error is used. Define $y_{u,i}$ as the rating for movie i by user u and $\hat{y}_{u,i}$ as the prediction. Where N is a number of user movie combinations and the sum is occurring over all these combinations. Then, the residual mean squared error is defined in the following equation: RMSE = $\sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$. The residual mean squared error can be interpreted similar to standard deviation. It is the error made when predicting a movie rating. If this number is much larger than one, then ratings are missing and predictions are not accurate. The following function computes this residual means squared error for a vector of ratings and their corresponding predictors in R: RMSE <- function(true_ratings, predicted_ratings) {sqrt(mean((true_ratings - predicted_ratings)^2))}. This recommendation system is based on an approach called matrix factorization. The simplest recommendation system will predict the same rating for all movies regardless of the user and movie. This model assumes the same rating for all movies and all users, with all the differences explained by random variation. Where ϵ represents independent errors sampled from the same distribution centered at zero, and μ represents the true rating for all movies and users. This is represented in the following model: $y_{u,i} = \mu + \epsilon_{u,i}$.

The estimate that minimizes the residual mean squared error is the least squared estimate of μ. In this case it's just the average of all the ratings which is computed with the mean( ) function in R. The average rating of all movies across all users is 3.51. Predicting all unknown ratings with this average is possible using the RMSE( ) function in R. Then, the residual mean squared error is approximately 1.06. This value is very large and this value needs to be as small as possible. The average minimizes the residual mean squared error when using this model. Some movies are generally rated higher than others. This can be seen making a plot of the average rating that each movie got. The assumption that different movies are rated differently is confirmed by data. The previous model can be improved by adding a term, b_i, to represent the average rating for movie i. The b_i's are called effects. This is represented in the following model: $y_{u,i} = \mu + b_i + \epsilon_{u,i}$.

The least squares can be used again to estimate the b_i's in the following way: fit <- lm(rating ~ as.factor(userId), data = movielens). There are thousands of b's which means each movie gets one parameter and one estimate. The lm function will run very slow and is not recommend to use. The least squared estimate $\hat{b}_i$ is the average of $y_{u,i}$ minus the overall mean for each movie, i. Dropping the hat notation in the code to represent the estimates going forward will make the code cleaner. The following code computes the estimates for the b's: movie_avgs <- edx %>% group_by(movieId) %>% summarize(b_i = mean(rating - mu)). These estimates vary substantially because some movies are good and other movies are bad. The overall average is approximately 3.5. So a b_i of 1.5 implies a perfect five-star rating. The prediction capability improves using this model because the residual mean squared error decreases to 0.94. The model can be improved further by adding more parameters like users. This is because different users vary in terms of how they rate movies. There is substantial variability across users which can be seen when the average rating for user u is computed. A further improvement to the model will include a term b_u, which is the user-specific effect. A negative b_u and a positive b_i can counter each others effects and improve the prediction. This is represented in the following model: $y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$.

To fit this model the function lm can be used. The code would look like this: fit <- lm(rating ~ as.factor(movieId) + as.factor(userId)). This is a big model and running the code will be very slow. Instead an approximation is done by computing the overall mean, $\hat{u}$, the movie effects, $\hat{b}_i$, and then estimating the user effects, $\hat{b}_u$, by taking the average of the residuals obtained after removing the overall mean and the movie effect from the ratings $y_{u,i}$. The following code is used: user_avgs <- edx %>% left_join(movie_avgs, by = "movieId") %>% group_by(userId) %>% summarize(b_u = mean(rating - mu - b_i)). The new Movie + User Effects model will be predicting values and computing the residual mean squared error. There is further improvement in the Movie + User Effects model because the residual mean squared error is approximately 0.85. This is a strong model because with an RMSE value of 0.85 the predictions from this model are reliable. After training and developing the machine learning algorithm with the edx dataset it is applied to the validation dataset. In the validation dataset the average rating of all movies across all users is approximately 3.51 and the residual mean squared error is approximately 1.06. The Movie Effect model has an RMSE of approximately 0.93 and the Movie + User Effects model has an RMSE of approximately 0.82. The tables with all the RMSE values is shown in results section below.

III. Results

Creating a table will help store the results when comparing different approaches. This table is called RMSE results. It's going to be created using this code rmse_results_edx <- data_frame(method = "Edx Dataset", RMSE = naive_rmse). A table is also created for the validation dataset in a similar way. These results validate the models and analysis in Section II. The model with the movie and user effects is the one with the lowest RMSE and accurate for predictions.

The table for the edx dataset is below.

```
## # A tibble: 3 x 2
##   method                      RMSE
##   <chr>                      <dbl>
## 1 Edx Dataset                1.06
## 2 Movie Effect Model         0.942
## 3 Movie + User Effects Model 0.857
```

The table for the validation dataset is below.

```
## # A tibble: 3 x 2
##   method                      RMSE
##   <chr>                      <dbl>
## 1 Validation Dataset         1.06
## 2 Movie Effect Model         0.938
## 3 Movie + User Effects Model 0.825
```

IV. Conclusion

The goal of creating a movie recommendation system using the MovieLens dataset was achieved using matrix factorization. The MovieLens dataset was used to generate the edx and validation datasets. The algorithm to predict movie ratings was developed and trained using the edx dataset. For a final test of the algorithm movie ratings were predicted in the validation set as if they were unknown. The residual mean squared error (RMSE) was used to evaluate how close the predictions are to the true values in the validation set. Models were created with parameters that had a major impact in the RMSE. By analyzing all the parameters in the dataset and their effects, a model that uses both the movie and user parameters was developed and trained using the edx dataset. After the movie and user effects model was trained with the edx dataset it was applied to the validation dataset. The movie and user effects model applied to the edx dataset yields an RMSE of approximately 0.85. The movie and user effects model applied to the validation dataset yields an RMSE of approximately 0.82. With these results the movie recommendation system will have accurate predictions.