Final_Project_1_Markdown

Meher Mankikar

June 20, 2019

R Markdown

First, I loaded all the data using the pre-given code for this project.

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-proje
ct.org")
```

```
## 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_confli
cts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

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

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

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

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

```
library(dplyr)

dl <- tempfile()
download.file("https://grouplens.org/datasets/movielens/10m/", dl)

ratings <- read.table(text = gsub("::", "\t", readLines("/Users/mehermankikar/Downloa
ds/ml-10M100K/ratings.dat")),
                      col.names = c("userId", "movieId", "rating", "timestamp"))

movies <- str_split_fixed(readLines("/Users/mehermankikar/Downloads/ml-10M100K/movies
.dat"), "\\::", 3)
colnames(movies) <- c("movieId", "title", "genres")
movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(levels(movieId))[movi
eId],
                                           title = as.character(title),
                                           genres = as.character(genres))

movielens <- left_join(ratings, movies, by = "movieId")

set.seed(1) # if using R 3.6.0: set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FA
LSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]

validation <- temp %>%
  semi_join(edx, by = "movieId") %>%
  semi_join(edx, by = "userId")

removed <- anti_join(temp, validation)
```

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

```
edx <- rbind(edx, removed)

rm(dl, ratings, movies, test_index, temp, movielens, removed)
```

Then, I created a preliminary algorithm that predicts the ratings using just the average of the sample.

```
mu_hat <- mean(edx$rating)
 mu_hat
```

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

```
rmse_1 <- RMSE(edx$rating, mu_hat)
 rmse_1
```

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

```
 predictions <- rep(2.5, nrow(edx))
 RMSE(edx$rating, predictions)
```

```
## [1] 1.466079
```

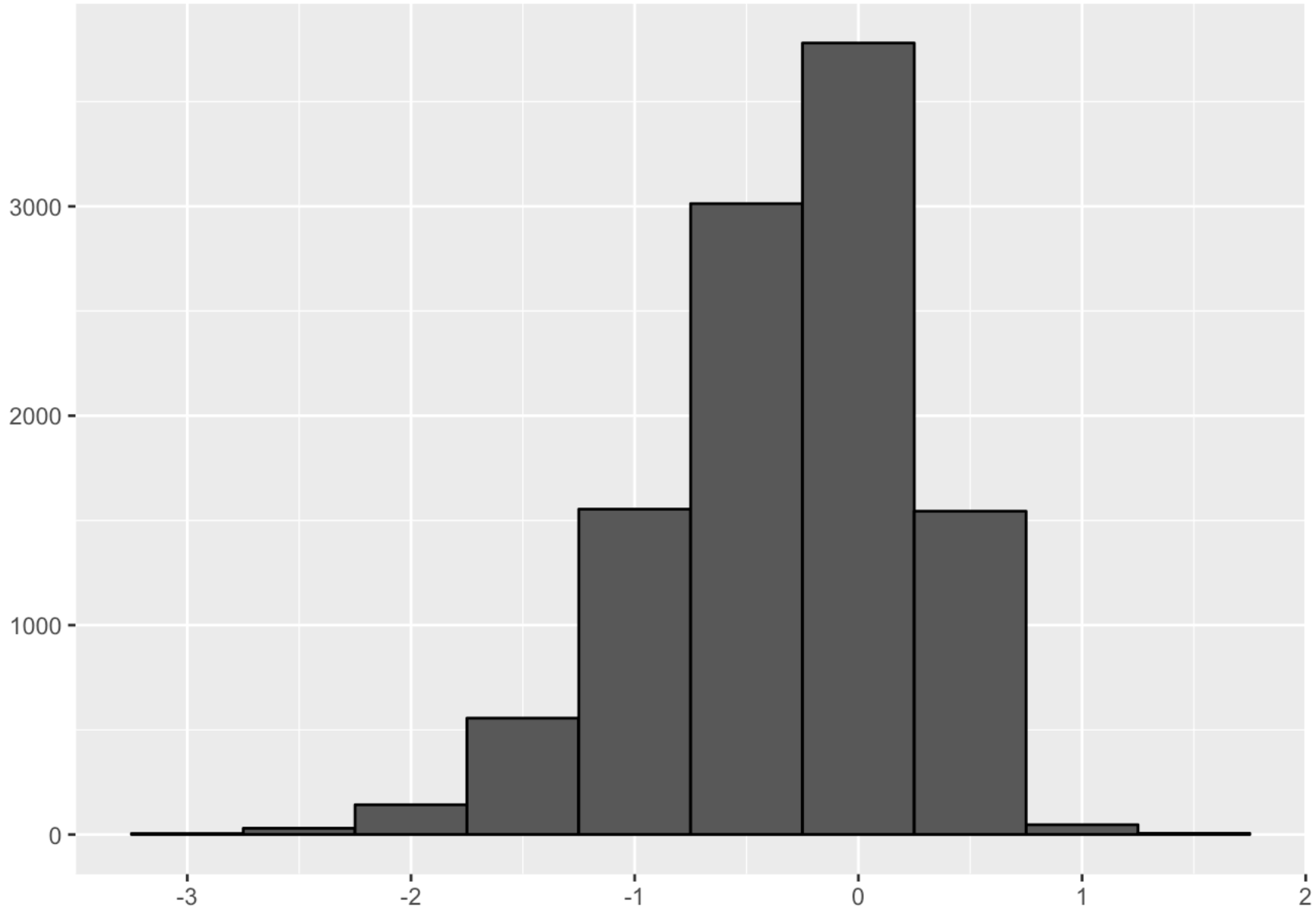
```
 rmse_results <- data_frame(method = "Just the average", RMSE = rmse_1)
```

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

Next, an algorithm was made that uses userID as the only factor.

```
 # fit <- lm(rating ~ as.factor(userId), data = movielens)
 mu <- mean(edx$rating)
 movie_avgs <- edx %>%
   group_by(movieId) %>%
   summarize(b_i = mean(rating - mu))

 movie_avgs %>% qplot(b_i, geom ="histogram", bins = 10, data = ., color = I("black")
)
```




```
predicted_ratings <- mu + edx %>%
  left_join(movie_avgs, by='movieId') %>%
  .$b_i

model_1_rmse <- RMSE(predicted_ratings, edx$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie Effect Model",
                                     RMSE = model_1_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	1.0603313
Movie Effect Model	0.9423475

Next, an algorithm was made that takes into account movieId and userId, which decreased the RMS.

```
# lm(rating ~ as.factor(movieId) + as.factor(userId))
user_avgs <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_i))

predicted_ratings <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_i + b_u) %>%
  .$pred

model_2_rmse <- RMSE(predicted_ratings, edx$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie + User Effects Model",
                                     RMSE = model_2_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	1.0603313
Movie Effect Model	0.9423475
Movie + User Effects Model	0.8567039

Finally, an algorithm was made that takes into account movieId, userId, and genre

```
#lm(rating ~as.factor(movieId) + as.factor(rating) + as.factor(genre))
genre_avgs  <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  group_by(genres) %>%
  summarize(b_y = mean(rating - mu - b_i - b_u))

predicted_ratings <- validation %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  left_join(genre_avgs, by='genres') %>%
  mutate(pred = mu + b_i + b_u + b_y) %>%
  .$pred

model_3_rmse <- RMSE(predicted_ratings, validation$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie + User + Genre Effects Model",
                                     RMSE = model_3_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE

Just the average	1.0603313
Movie Effect Model	0.9423475
Movie + User Effects Model	0.8567039
Movie + User + Genre Effects Model	0.8649469