MovieLens

Luke Hensley

6/10/2019

I. Introduction

Three models ("Simple Average", "Movie_Effect" and "Movie+User_Effect") were developed and assessed using RMSE for this project. The best model, "Movie + User_Effect Model" (RMSE 0.8426), is run directly against the validation set to predict movie ratings. The RMSE result on the validation dataset (0.8294) is lower than that of the test dataset (0.8426). THis would suggest that the model is likely a good prediction model.

II. Load Data

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

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2
```

```
## -- Attaching packages ---------------------------------------------------------------------
```

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

```
## -- Conflicts ------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x 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
```

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

III. Training and Testing dataset Datasets are derived using edx set: 80% sample for training, 20% sample for testing.

```
set.seed(1)
train_index <- createDataPartition(y = edx$rating, times = 1, p = 0.8, list = FALSE)
train_set <- edx[train_index,]
temp <- edx[-train_index,]
test_set <- temp %>%
semi_join(train_set, by = "movieId") %>%
semi_join(train_set, by = "userId")
removed <- anti_join(temp, test_set)
```

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

```
train_set <- rbind(train_set, removed)
rm(temp, removed)
```

IV. Evaluate Algorithm

The following is used to test the three algorithms.

```
RMSE <- function(true_ratings, predicted_ratings){
sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

1st model: Simple Average Model

```
mu_hat <- mean(train_set$rating)
model_1_rmse <- RMSE(test_set$rating, mu_hat)
rmse_results <- data_frame(Model = "Simple Average", RMSE = model_1_rmse)
```

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

```
rmse_results%>%knitr::kable()
```

Model	RMSE
Simple Average	1.06095
2nd model: Movie_	Effect Model

```
mu <- mean(train_set$rating)
movie_avgs <- train_set %>%
  group_by(movieId) %>%
  summarize(b_i = mean(rating - mu))

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

```
model_2_rmse <- RMSE(predicted_ratings, test_set$rating)
rmse_results <- bind_rows(rmse_results,
data_frame(Model="Movie_Effect",
RMSE = model_2_rmse ))
rmse_results %>% knitr::kable()
```

Model	RMSE
Simple Average	1.0609505
Movie_Effect	0.9441124
The RMSE shows th e 2nd model is an improvement from the 1st.	

3rd model: Movie+User_Effect Model

```
user_avgs <- test_set %>%
left_join(movie_avgs, by='movieId') %>%
group_by(userId) %>%
summarize(b_u = mean(rating - mu - b_i))

predicted_ratings <- test_set %>%
left_join(movie_avgs, by='movieId') %>%
left_join(user_avgs, by='userId') %>%
mutate(pred = mu + b_i + b_u) %>%
.$pred
model_3_rmse <- RMSE(predicted_ratings, test_set$rating)
rmse_results <- bind_rows(rmse_results,
data_frame(Model="Movie + User_Effect",
RMSE = model_3_rmse ))
rmse_results %>% knitr::kable()
```

Model	RMSE
Simple Average	1.0609505
Movie_Effect	0.9441124
Movie + User_Effect	0.8436989
RMSE is further reduce d using the 3rd model.	

V. Evaluate validation set

Based on the above results, the best model, "Movie + User_Effect Model", is selected and run against the validation set. The RMSE of the validation set is 0.8294.

```
user_avgs_validation <- validation %>%
left_join(movie_avgs, by='movieId') %>%
group_by(userId) %>%
summarize(b_u = mean(rating - mu - b_i))
predicted_ratings <- validation %>%
left_join(movie_avgs, by='movieId') %>%
left_join(user_avgs_validation, by='userId') %>%
mutate(pred = mu + b_i + b_u) %>%
.$pred
```

```
model_rmse_validation <- RMSE(predicted_ratings, validation$rating)
model_rmse_validation
```

```
## [1] 0.8294885
```

VI. Conclusion

In this project, three models ("Simple Average", "Movie_Effect" and "Movie+User_Effect") were developed and assessed using their RMSE. The best model, "Movie + User_Effect Model" (RMSE of 0.8426), was run directly against the validation set. The RMSE of the validation dataset was 0.8294, and is lower than that of the test dataset (0.8426). This suggests that the model is likely a good prediction model.