Capstone MovieLens

Adam

2022-03-05

Capstone MovieLens Assaignment

Movie Recommendation system based on MovieLens Data set. The Dataset is altered and loaded as the edx course suggests. Official test set is loaded into "validation" but it only can be used for end result RMSE checking. So in order to see the results through the functions, I partioned the "edx" data set to "train_set" and "test_set".

```
mytest_index <- createDataPartition(y = edx$rating, times = 1, p = 0.2, list = FALSE)
train_set <- edx[-mytest_index,]
test_set <- edx[mytest_index,]
```

RMSE function

RMSE function is used for checking the accuracy of our predictions:

```
RMSE <- function(actual_rating, predicted_rating) {
  sqrt(mean((actual_rating - predicted_rating)^2))
}
```

Baseline, average approach

Creating and testing baseline where the prediction is the general average across the data set and checking its accuracy against our partitioned "test_set". Adding its results to a tibble, that will stand as a summary table.

```
mu_hat <- mean(train_set$rating)
rmse_mu <- RMSE(test_set$rating, mu_hat)

rmse_resTable <- tibble(method = "Predicting by average", RMSE = rmse_mu)
rmse_resTable
```

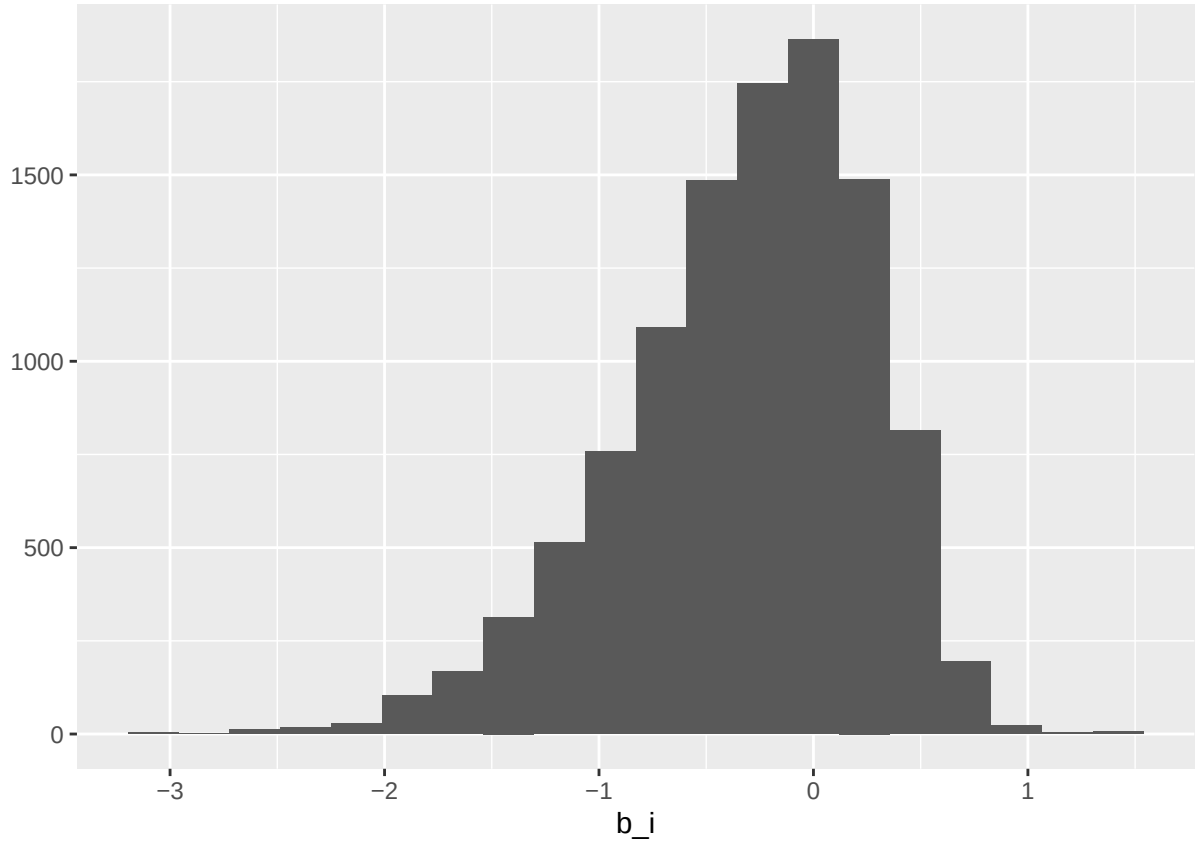
```
## # A tibble: 1 x 2
##   method                RMSE
##   <chr>                <dbl>
## 1 Predicting by average  1.06
```

Movies effect on prediction

Checking the movies effecting the prediction

```
movie_effect <- train_set %>% group_by(movieId) %>% summarise(b_i = mean(rating - mu_hat))
```

Seeing it on a plot:




```
movie_effect %>% summarise(sum = sum(b_i))
```

```
## # A tibble: 1 x 1
##       sum
##     <dbl>
## 1 -3388.
```

As we can see, the movies have an expecting dumping effect on the overall accuracy that will probably bring the value closer to 0.

```
pred_movies <- test_set %>% group_by(movieId) %>% left_join(movie_effect, "movieId") %>% pull(b_i)
sum(is.na(pred_movies))
```

```
## [1] 45
```

```
pred_movies[is.na(pred_movies)] <- mu_hat
pred_mui <- mu_hat + pred_movies
```

Removed 45 NA s from data set and replaced with a baseline prediction. This will have a minimal effect for accuracy since the data set itself is very big.

Checking its RMSE and plugging it into the tibble for comparison

```
rmse_mui <- RMSE(test_set$rating, pred_mui)
rmse_resTable <- rmse_resTable %>% add_row(method = "Movie effect on prediction", RMSE =  rmse_mui)
rmse_resTable
```

```
## # A tibble: 2 x 2
##   method                       RMSE
##   <chr>                       <dbl>
## 1 Predicting by average      1.06
## 2 Movie effect on prediction 0.944
```

Users effect on prediction

Calculating the users effect alone

```
user_effect <- train_set %>% group_by(userId) %>% summarise(b_u =  mean(rating) - mu_hat)
pred_usr <- test_set %>%
  group_by(movieId) %>%
  left_join(user_effect, "userId") %>%
  mutate(prediction = mu_hat + b_u) %>%
  pull(prediction)

rmse_usr <- RMSE(test_set$rating,pred_usr)
```

Adding and comparing to the table:

```
rmse_mov_usr <- RMSE(test_set$rating,pred_usr)
rmse_resTable <- rmse_resTable %>% add_row(method = "Users effect on prediction", RMSE =  rmse_usr)
rmse_resTable
```

```
## # A tibble: 3 x 2
##   method                       RMSE
##   <chr>                       <dbl>
## 1 Predicting by average      1.06
## 2 Movie effect on prediction 0.944
## 3 Users effect on prediction 0.979
```

Similarly as the movies users had a good effect on prediction accuracy

Genres effect on the prediction

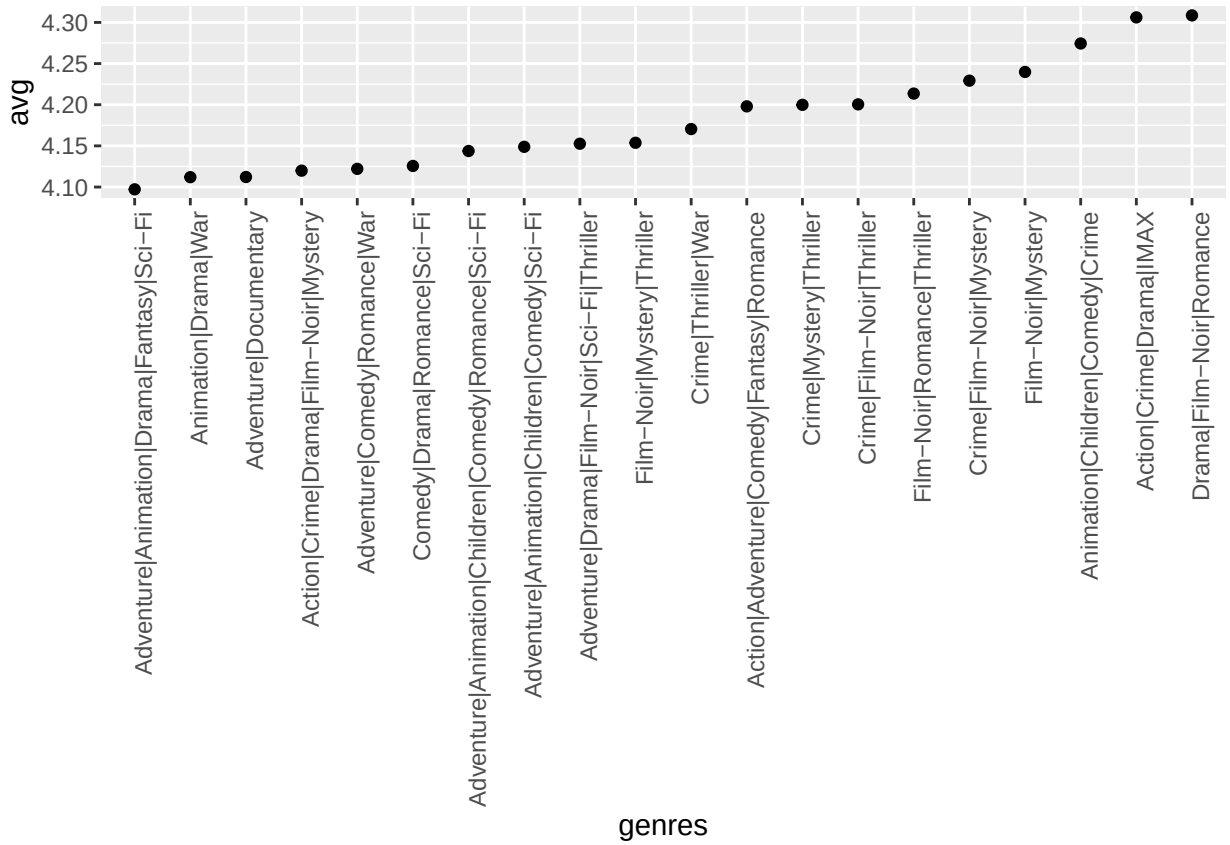
In theory I would expect genres have a small but still visible effect on the accuracy since people often watch movies on genres they did not like because many people watch from friends or relatives' recommendations.

I think many people is not checking the genre before watching a movie. Also many genres are combinations that might diverge from the definition of one of the given genre. This should tell if people generally act as I would guess.

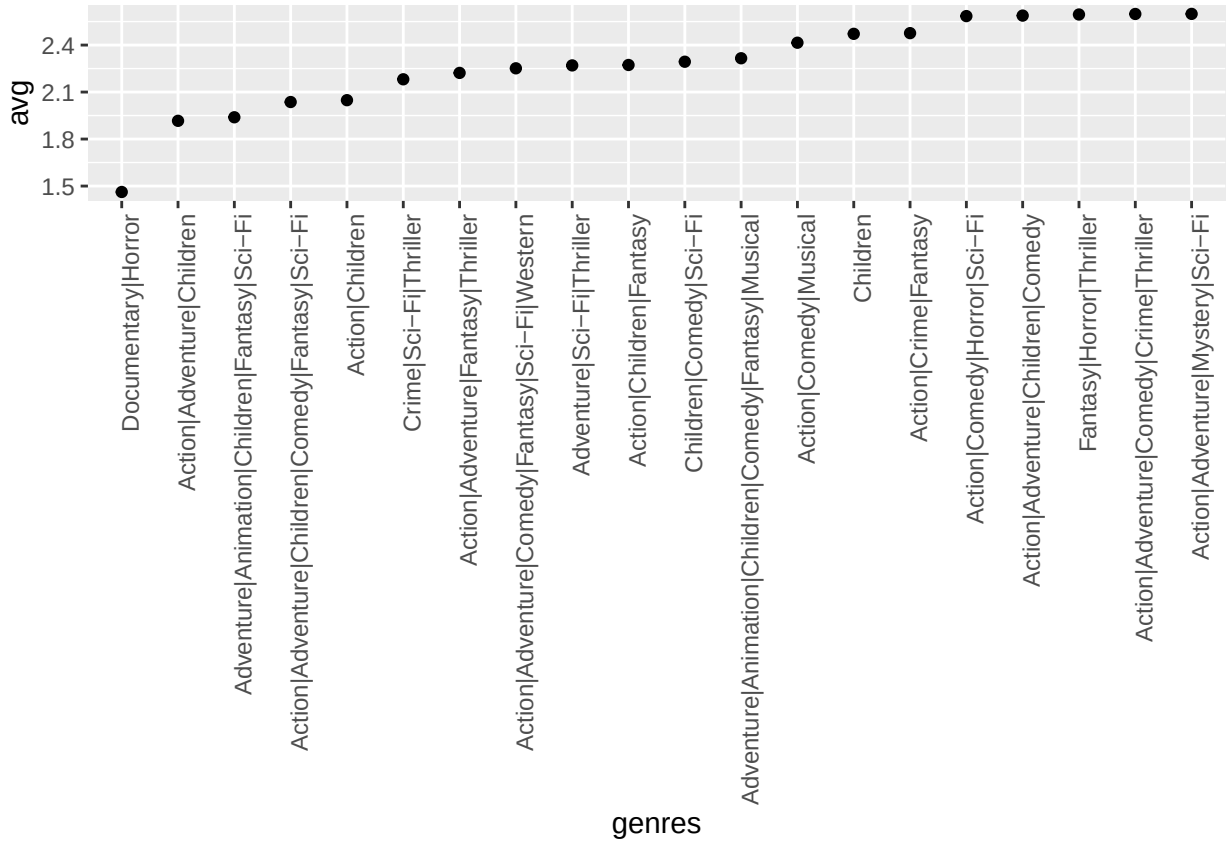
```
genres_table <- train_set %>% group_by(genres) %>%
  summarize(n = n(), avg = mean(rating)) %>%
  filter(n >= 500) %>%
  mutate(genres = reorder(genres, avg))

str(genres_table)
```

```
## tibble [487 x 3] (S3: tbl_df/tbl/data.frame)
##  $ genres: Factor w/ 487 levels "Documentary|Horror",..: 64 330 445 27 249 441 250 190 279 159 ...
##   ..- attr(*, "scores")= num [1:487(1d)] 2.94 3.66 3.96 2.7 3.51 ...
##   .. ..- attr(*, "dimnames")=List of 1
##   .. .. ..$ : chr [1:487] "Action" "Action|Adventure" "Action|Adventure|Animation|Children|Comedy" "
##  $ n     : int [1:487] 19666 54752 6036 590 1508 3417 702 3276 5180 3747 ...
##  $ avg   : num [1:487] 2.94 3.66 3.96 2.7 3.51 ...
```

```
genres_table %>% arrange(desc(avg)) %>%
  head(.,20) %>%
  ggplot(aes(x = genres, y = avg)) +
  geom_point()+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
genres_table %>% arrange(desc(avg)) %>%
  tail(.,20) %>%
  ggplot(aes(x = genres, y = avg)) +
  geom_point()+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



As it is is visible by inspecting the first 20 and last 20 entries it is visible there is some trend across the genre combinations. Let's see its effect.

```
genres_effect <- train_set %>%
  group_by(genres) %>%
  summarise(b_g = mean(mean(rating) - mu_hat))

pred_genres <- test_set %>%
  group_by(movieId) %>%
  left_join(genres_effect, "genres") %>%
  mutate(prediction = mu_hat + b_g) %>%
  pull(prediction)
if(sum(is.na(pred_genres))) pred_genres[is.na(pred_genres)] <- mu_hat
```

Checking RMSE and adding this to the table too.

```
rmse_gen <- RMSE(test_set$rating, pred_genres)
rmse_resTable <- rmse_resTable %>% add_row(method = "Genres effect on prediction", RMSE =  rmse_gen)
rmse_resTable
```

```
## # A tibble: 4 x 2
##   method                        RMSE
##   <chr>                        <dbl>
## 1 Predicting by average       1.06
## 2 Movie effect on prediction  0.944
## 3 Users effect on prediction  0.979
## 4 Genres effect on prediction 1.02
```

I wanted to check if users have some impact on these genre combinations so I checked it but did not add to the table:

```
genres_effect_bad <- train_set %>%
  left_join(user_effect, "userId") %>%
  group_by(genres) %>%
  summarise(b_g = mean(mean(rating) - mu_hat - b_u))
# user related to genres not as punctual but very close to checking only the genres
pred_genres2 <- test_set %>%
  group_by(movieId) %>%
  left_join(genres_effect_bad, "genres") %>%
  mutate(prediction = mu_hat + b_g) %>%
  pull(prediction)
if(sum(is.na(pred_genres2))) pred_genres2[is.na(pred_genres2)] <- mu_hat

rmse_gen2 <- RMSE(test_set$rating,pred_genres2)
rmse_gen2
```

```
## [1] 1.019527
```

The original was:

```
rmse_gen
```

```
## [1] 1.018538
```

So we can see it has a little effect but overall not much. Note this is not the same prediction that predict ratings by genre + user effect

Combining effects for the model

Combining the Movies-, users, and genres effects for the prediction:

```
pred_mov_usr_gen <- test_set %>%
  group_by(movieId) %>%
  left_join(movie_effect, "movieId") %>%
  left_join(user_effect, "userId") %>%
  left_join(genres_effect, "genres") %>%
  mutate(prediction = mu_hat + b_i + b_u + b_g) %>%
  pull(prediction)
if(sum(is.na(pred_mov_usr_gen))) pred_mov_usr_gen[is.na(pred_mov_usr_gen)] <- mu_hat

rmse_gen <- RMSE(test_set$rating,pred_genres)
rmse_gen
```

```
## [1] 1.018538
```

Does not seem to good so I will try to combine only movies and user effects:

```
pred_mov_usr <- test_set %>%
  group_by(movieId) %>%
  left_join(movie_effect, "movieId") %>%
  left_join(user_effect, "userId") %>%
  mutate(prediction = mu_hat + b_i + b_u) %>%
  pull(prediction)
if(sum(is.na(pred_mov_usr))) pred_mov_usr[is.na(pred_mov_usr)] <- mu_hat

rmse_mov_usr <- RMSE(test_set$rating, pred_mov_usr)
rmse_mov_usr
```

```
## [1] 0.8862238
```

Adding results to the table:

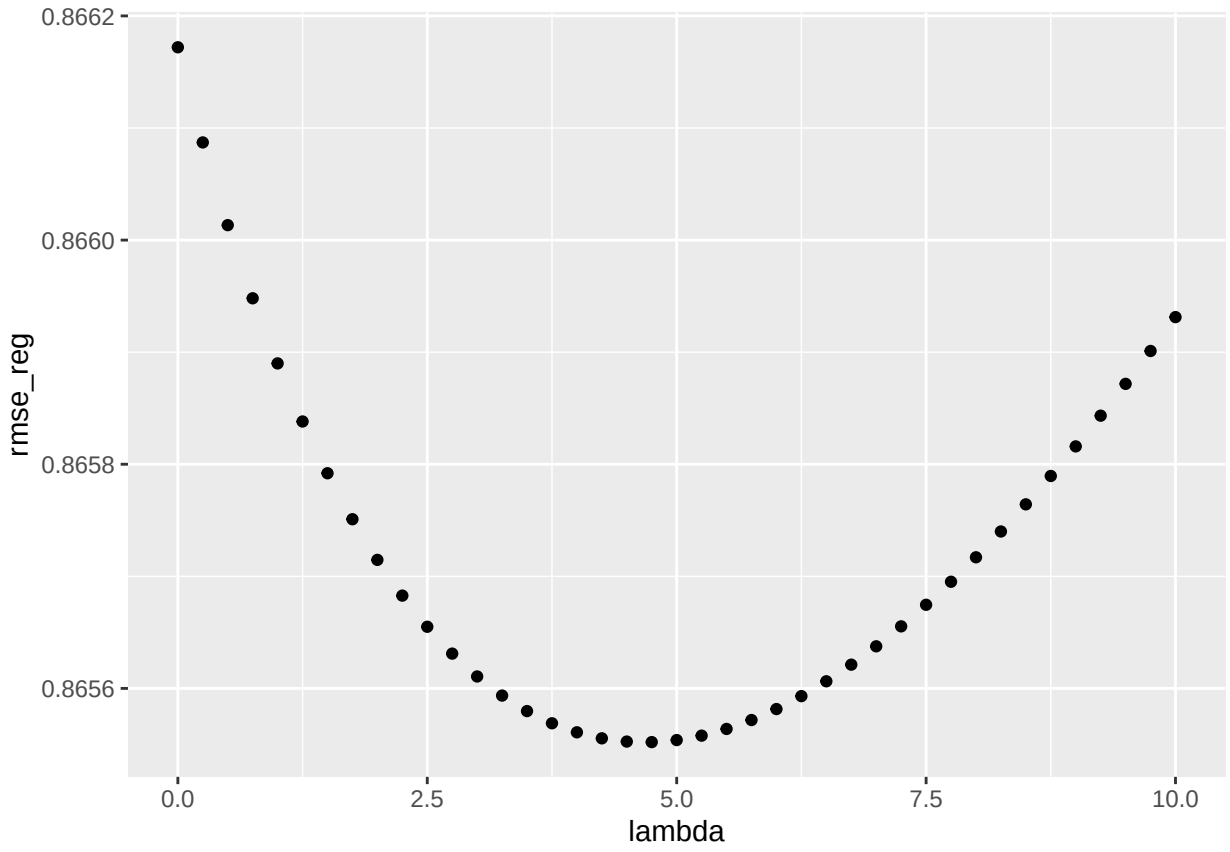
```
rmse_resTable <- rmse_resTable %>% add_row(method = "Movie + User effect on prediction", RMSE =  rmse_m
rmse_resTable
```

```
## # A tibble: 5 x 2
##   method                              RMSE
##   <chr>                              <dbl>
## 1 Predicting by average              1.06
## 2 Movie effect on prediction         0.944
## 3 Users effect on prediction         0.979
## 4 Genres effect on prediction        1.02
## 5 Movie + User effect on prediction 0.886
```

Regularisation

```
lambda <- seq(0, 10, 0.25)

rmse_reg <- sapply(lambda, function(lambda){

  movies_reg <- train_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu_hat)/(n()+lambda))

  users_reg <- train_set %>%
    left_join(movies_reg, by="movieId") %>%
    group_by(userId) %>%
    summarise(b_u = sum(rating - b_i - mu_hat)/(n()+lambda))

  pred_mov_usr_reg <- test_set %>%
    group_by(movieId) %>%
    left_join(movies_reg, "movieId") %>%
    left_join(users_reg, "userId") %>%
```

```
    mutate(prediction = mu_hat + b_i + b_u) %>%
    pull(prediction)
  if(sum(is.na(pred_mov_usr_reg))) pred_mov_usr_reg[is.na(pred_mov_usr_reg)] <- mu_hat

  return(RMSE(test_set$rating, pred_mov_usr_reg ))
})

qplot(lambda, rmse_reg)
```




```
lambda[which.min(rmse_reg)]
```

```
## [1] 4.75
```

RMSE with the chosen 4.75 lambda:

```
rmse_reg[which.min(rmse_reg)]
```

```
## [1] 0.8655521
```

In table:

```
rmse_resTable <- rmse_resTable %>% add_row(method = "Regularized Movie + User effect on prediction", RMS
rmse_resTable
```

```
## # A tibble: 6 x 2
##   method                                          RMSE
##   <chr>                                          <dbl>
## 1 Predicting by average                          1.06
## 2 Movie effect on prediction                     0.944
## 3 Users effect on prediction                     0.979
## 4 Genres effect on prediction                    1.02
## 5 Movie + User effect on prediction              0.886
## 6 Regularized Movie + User effect on prediction 0.866
```

Final model is used on the validation set to see how it performs. Final Root mean square error value is defined by:

```
l <- 4.75

movies_reg <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu_hat)/(n()+l))

users_reg <- edx %>%
  left_join(movies_reg, by="movieId") %>%
  group_by(userId) %>%
  summarise(b_u = sum(rating - b_i - mu_hat)/(n()+l))

pred_mov_usr_reg <- validation %>%
  group_by(movieId) %>%
  left_join(movies_reg, "movieId") %>%
  left_join(users_reg, "userId") %>%
  mutate(prediction = mu_hat + b_i + b_u) %>%
  pull(prediction)
if(sum(is.na(pred_mov_usr_reg))) pred_mov_usr_reg[is.na(pred_mov_usr_reg)] <- mu_hat

RMSE(validation$rating, pred_mov_usr_reg)
```

```
## [1] 0.8648201
```

Conclusion

The most impact was made by combining the movie effect and the user effect to minimize the root mean squared error. Genres proved helpful overall but the final model's accuracy was only decreased by it.