Recommendation System

Cyrus Cai

3/30/2020

1 Introduction

With the assumption that historical data contains inights for the future, we attempt to predict human behavior. The research is done in various areas, such as retail, movies, polls etc.

1.1 The Chanllenge

Today, I follow many processors and will build a recommendation system for movies. The model will predict what rating the exsiting user will give for a movie that he/she never watched before.

1.2 Dataset: MovieLens

I will use **MovieLens** as our dataset. The dataset is generated by GroupLens reseach lab. You can download the data through the link [(http://files.grouplens.org/datasets/movielens/ml-10m.zip)].

```
##   userId movieId rating timestamp                          title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
##                           genres
## 1                Comedy|Romance
## 2          Action|Crime|Thriller
## 4  Action|Drama|Sci-Fi|Thriller
## 5       Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
```

1.3 Goal

I will find a consice and yet accurate model. The project is aimed at an RMSE that is less than 0.865. The project is also aimed to produce an easy-to-follow and reproducible report.

2 Analysis

Before modeling, I create two dataset: **edx** and **validation**. The **validation** is 10% of the full dataset, and is only used for final assessement of the model. It won't be used for anywhere else in the project. The **edx** data is going to be split into two subset, **train_set** and **test_set**.

2.1 The Simplest Model

The simplest model, y_i=mu+e_i. I use the average of all ratings as the predicted rating for each movie and user.

```
mu<-mean(train_set$rating)
rmse_simplest_model <- RMSE(mu,test_set$rating)
```

Performance

```
rmse_results
```

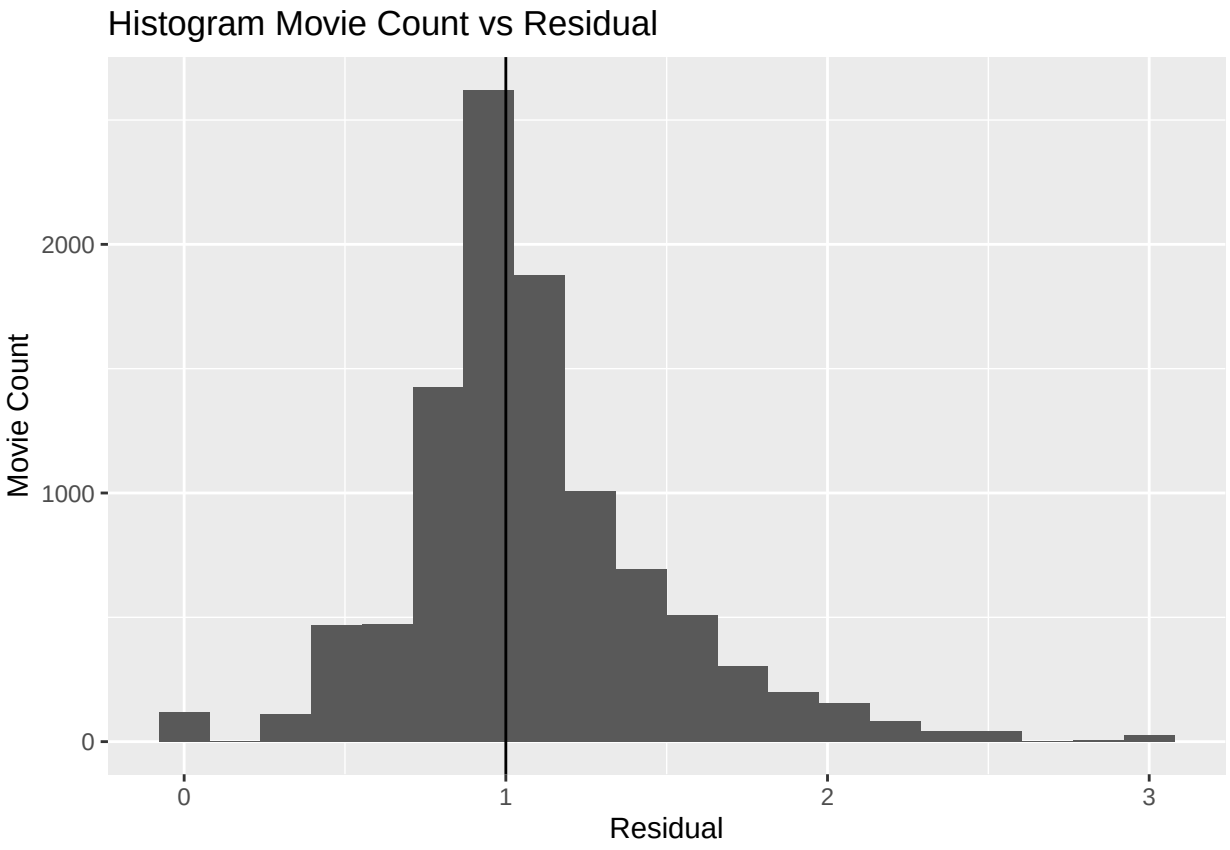
```
##    method    RMSE
## 1 Average 1.06062
```

Pros

This modle is the simplest model.

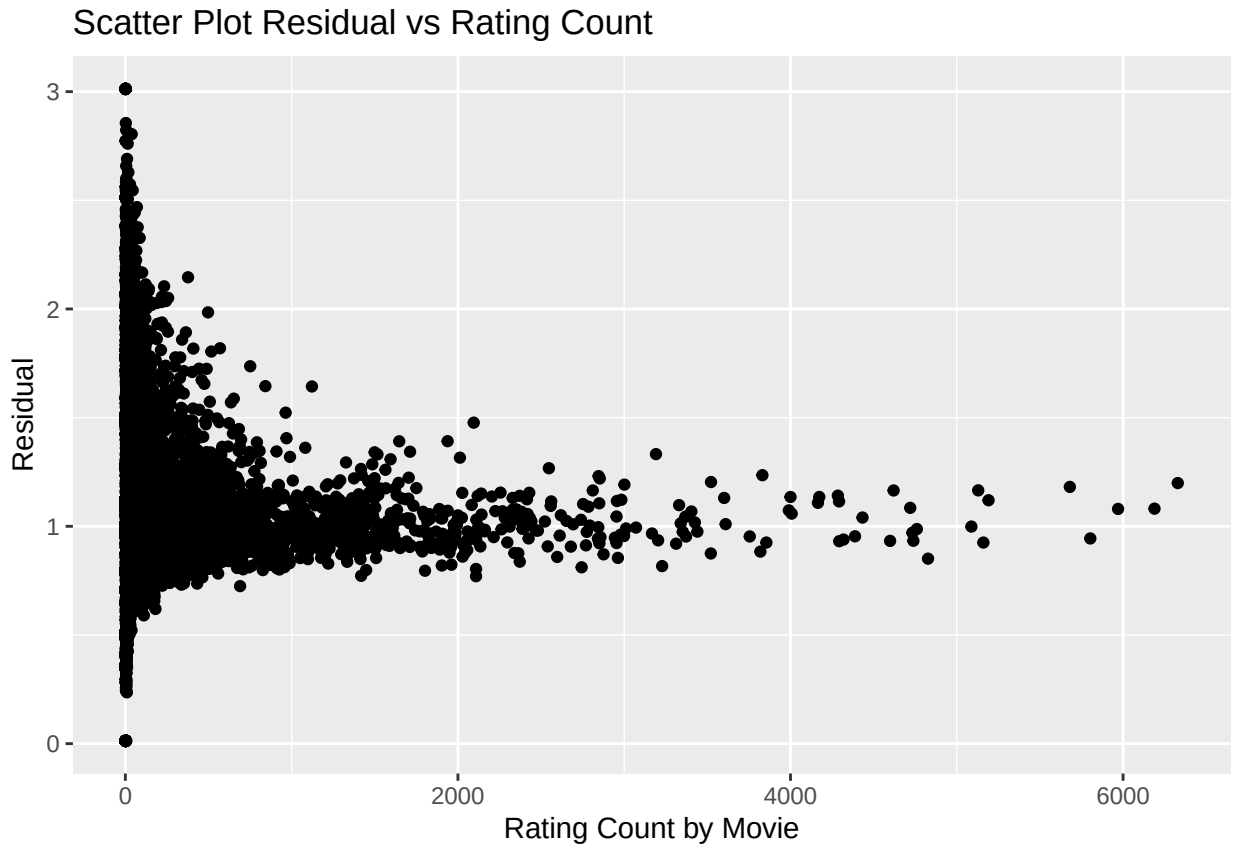
Cons

The histogram shows that majority of the movies have residual more than 1.



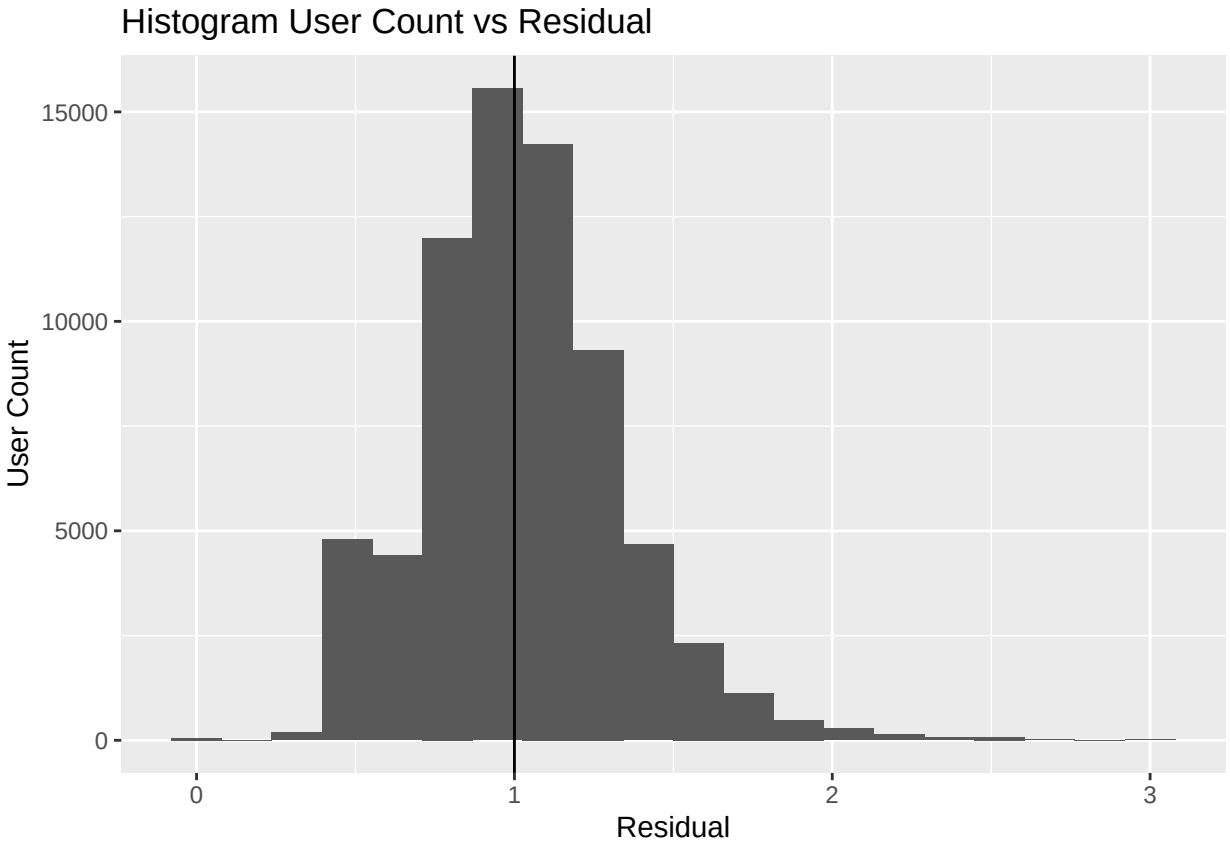
```
## [1] "% of residuals higher than 1" "53.2394643560457"
```

The scatter plot shows that the residual converts to 1 as the rating count increase.In other words, the more rating for a movie doesn't increase the prediction accuracy with this model.



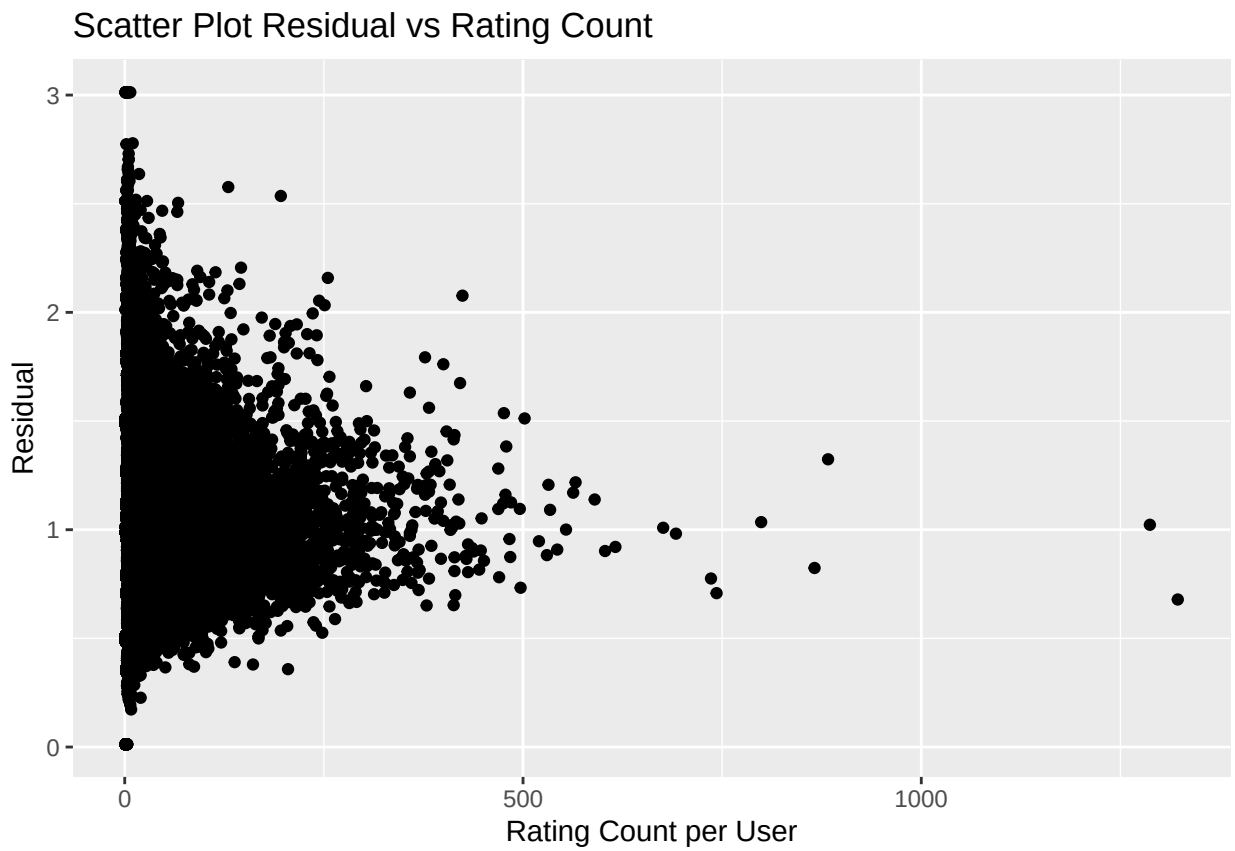
Therefore, I need to consider movie as a factor for the next model.

The histogram shows that majority of the users have residual more than 1




```
## [1] "% of residuals higher than 1" "51.0839331287816"
```

The scatter plot shows that the residual converts to 1 as the rating count increase.In other words, the more ratings from a user don't increase the prediction accuracy with this model.



Therefore, I need to consider user as a factor for the next model.

2.2 Model 2

This model considers movie and user average ratings, y_ij=mu+m_i+u_i+e_ij. m_i is average of the ratings for movie i. u_j is the average of the ratings for movie j. eij is the random error of rating of movie i from user j.

```
movie_avgs <- train_set %>% group_by(movieId) %>%
  summarise(m_i=mean(rating-mu))

user_avgs <- train_set %>% left_join(movie_avgs,by="movieId") %>%
  group_by(userId) %>%
  summarise(u_i=mean(rating-mu-m_i))

predicted_rating <- test_set %>% left_join(movie_avgs,by="movieId") %>%
  left_join(user_avgs,by="userId") %>% mutate(predicted_rating=mu+m_i+u_i) %>%
  .$predicted_rating
rmse_MovieAndUserAverage <- RMSE(predicted_rating,test_set$rating)
```

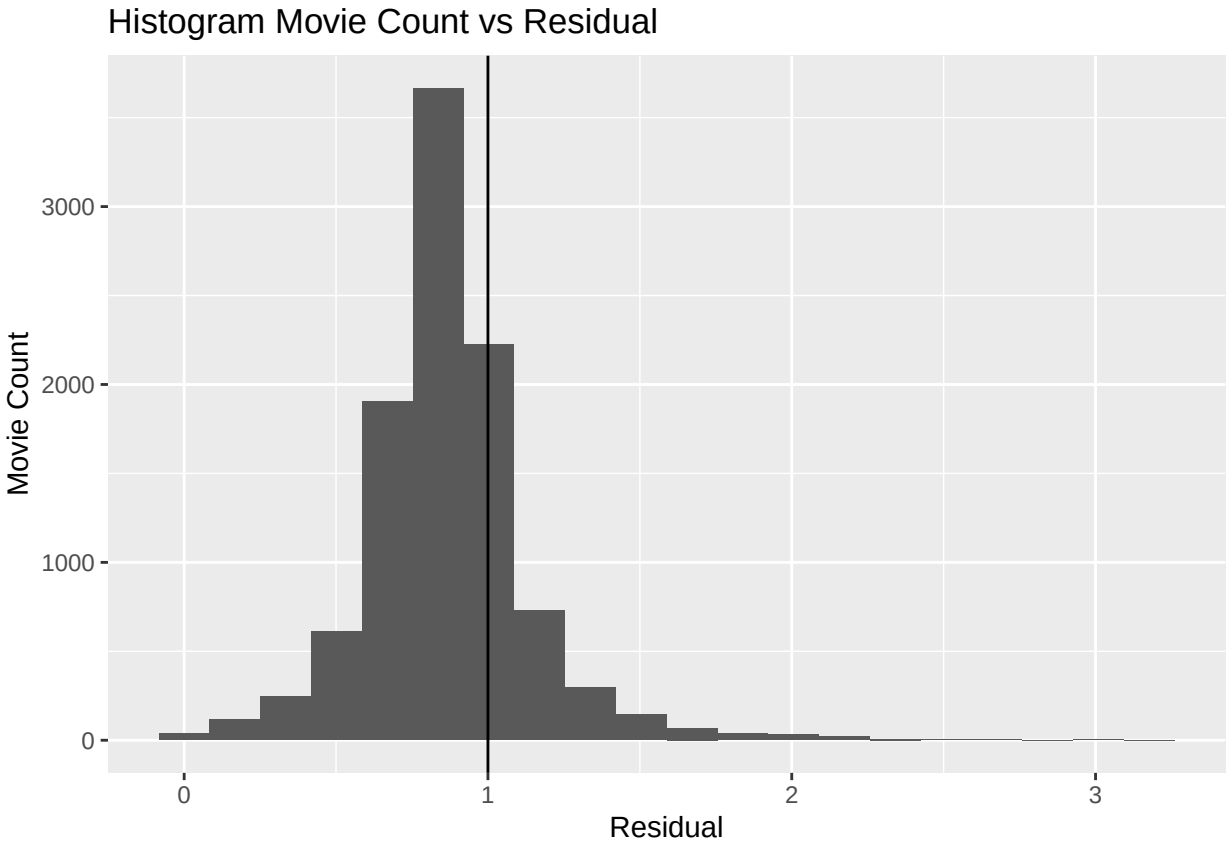
Performance

The RMSE increase from 1.0606 to 0.8660. The accuracy improves siginificantly. But it does not reach my goal yet.

```
##                         method      RMSE
## 1                      Average 1.0606200
## 2 Add Movie and User Average 0.8659711
```

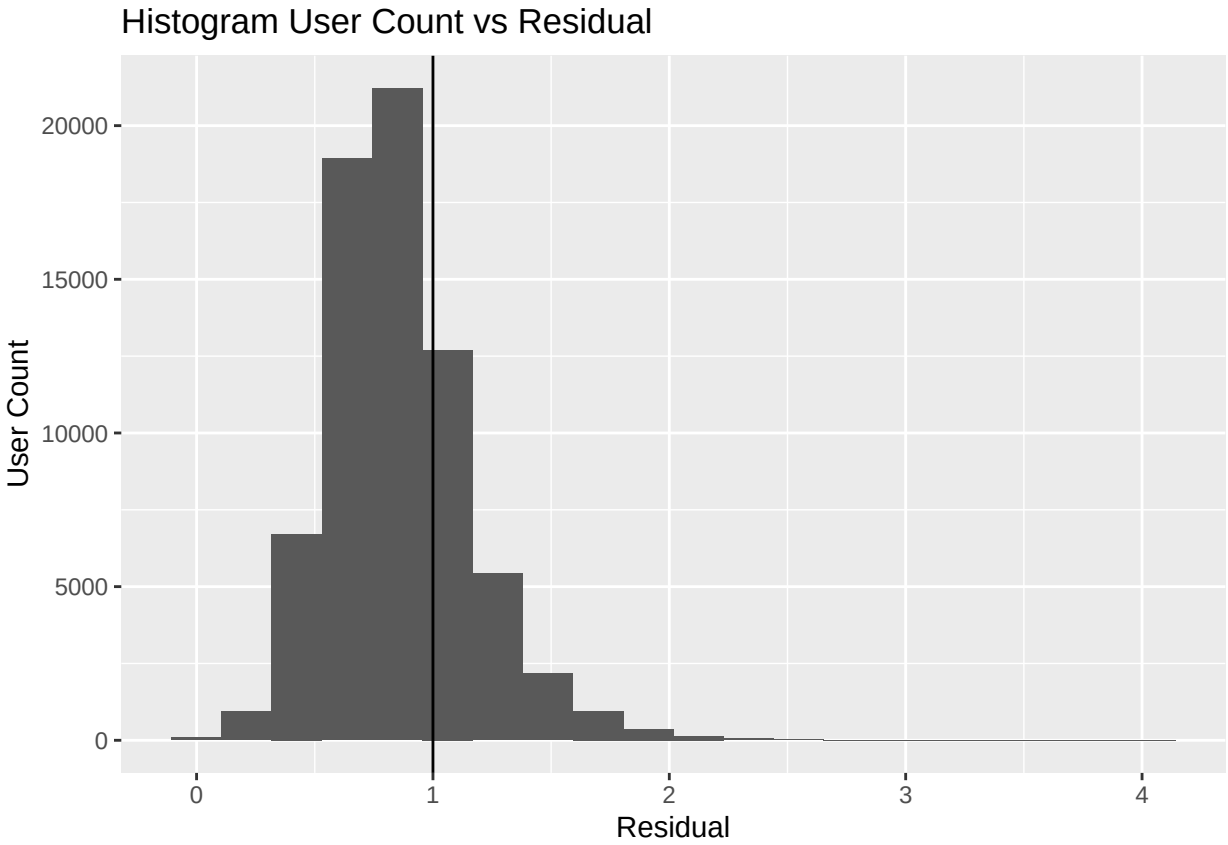
Pros

The histogram shows that majority of the movies have residual less than 1.Thus, adding the movie factor improves the result.




```
## [1] "% of residuals higher than 1" "21.8885387948011"
```

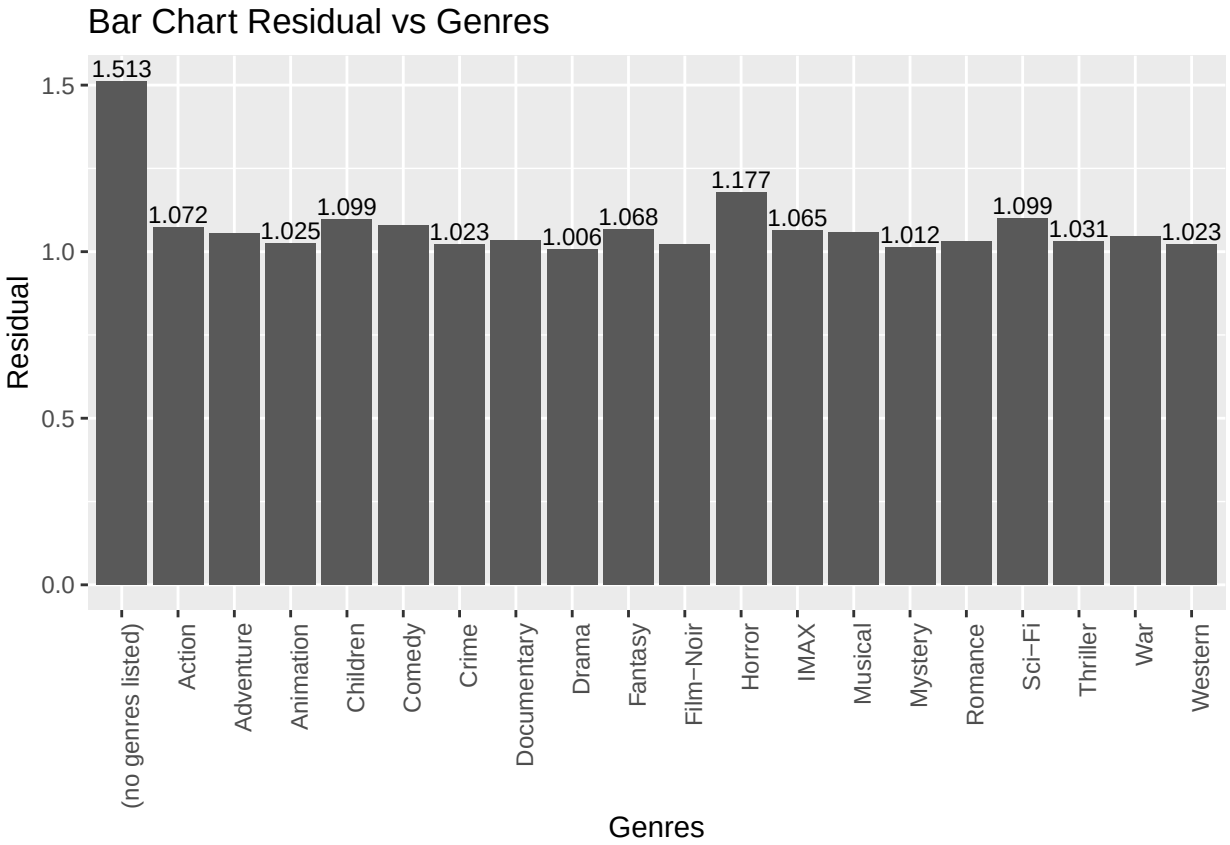
The histogram shows that majority of the users have residual less than 1.Thus, adding the user factor improves the result.



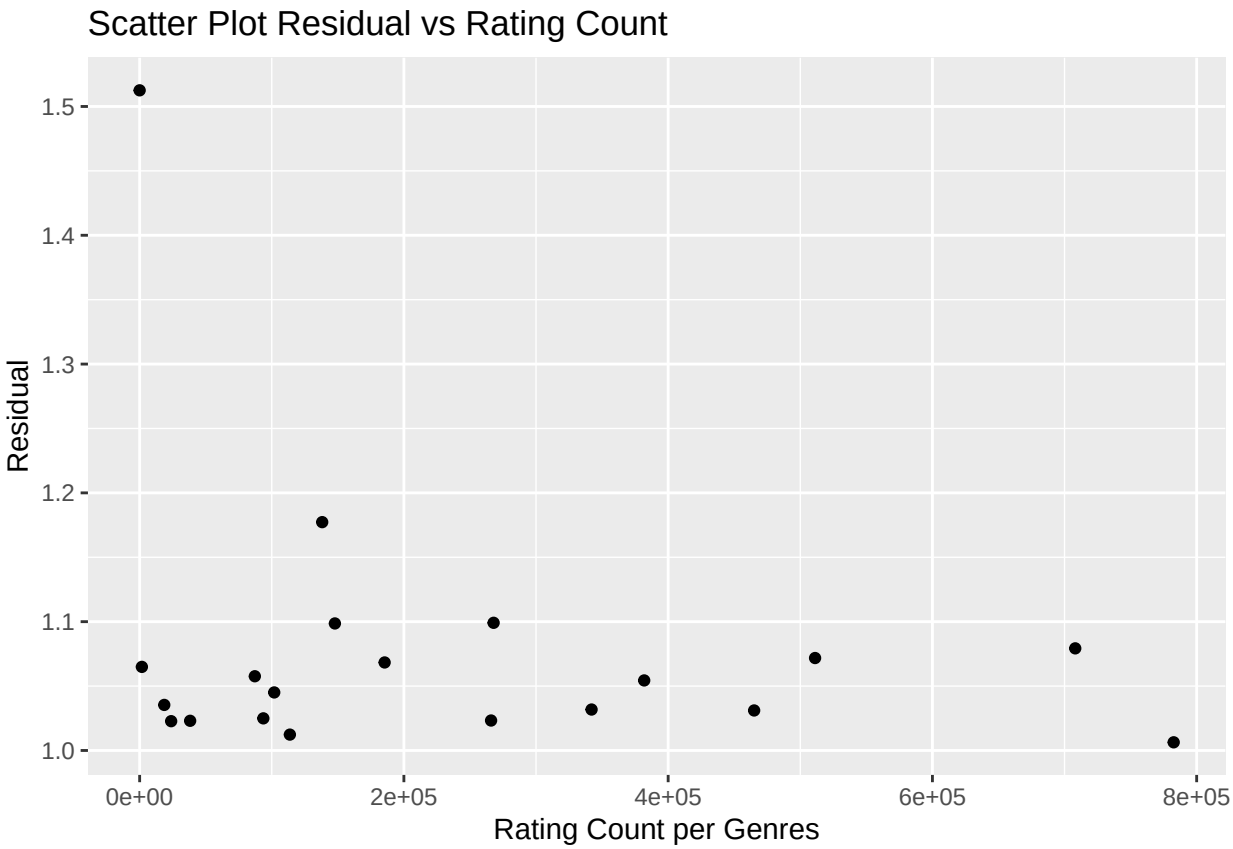

```
## [1] "% of residuals higher than 1" "26.2050870300806"
```

Cons

The bar chat shows that all the genres have residual greater than 1.



The scatter plot shows that the residual doesn't decrease as the rating count increase.In other words, the more ratings for a genre don't increase the prediction accuracy with this model.



Therefore, I need to consider the genre as a factor in the model 3.

2.3 Model 3

I call the third model *User Profile Model*. I will look at the user's historial rating record and identify the most rated genre. The genre will be marked as "Top 1", while any other genres will be marked as "Non-Top 1" for this user.

The underlying reasoning is: the user who like watching movies of certain genre gives an above-average rating to them.

The model can be expressed as, y_ij=mu+m_i+u_j(genres)+e_ij, where u_j is a function of genres.

```
movie_avgs <- train_set %>% group_by(movieId) %>%
  summarise(m_i=mean(rating-mu))

user_profile_avgs <- train_set_with_genres_breakdown %>%
  left_join(movie_avgs,by="movieId") %>%
  left_join(user_profile_rank_by_genres,by=c("userId","genres2")) %>%
  left_join(user_profile_mean_by_rank,by=c("userId","rank")) %>%
  group_by(userId,rank) %>%
  summarise(u_i=mean(rating-mu-m_i))
```

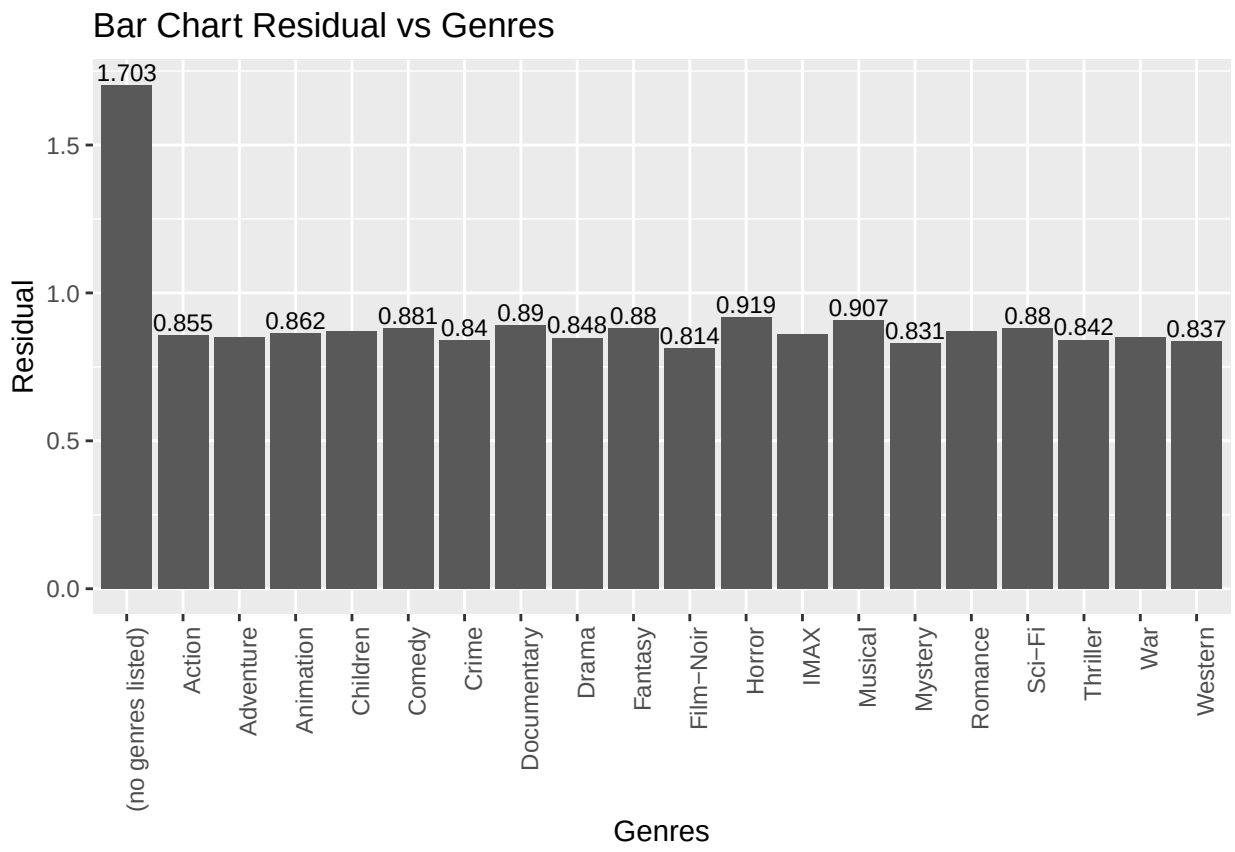
Performance

```
rmse_results
```

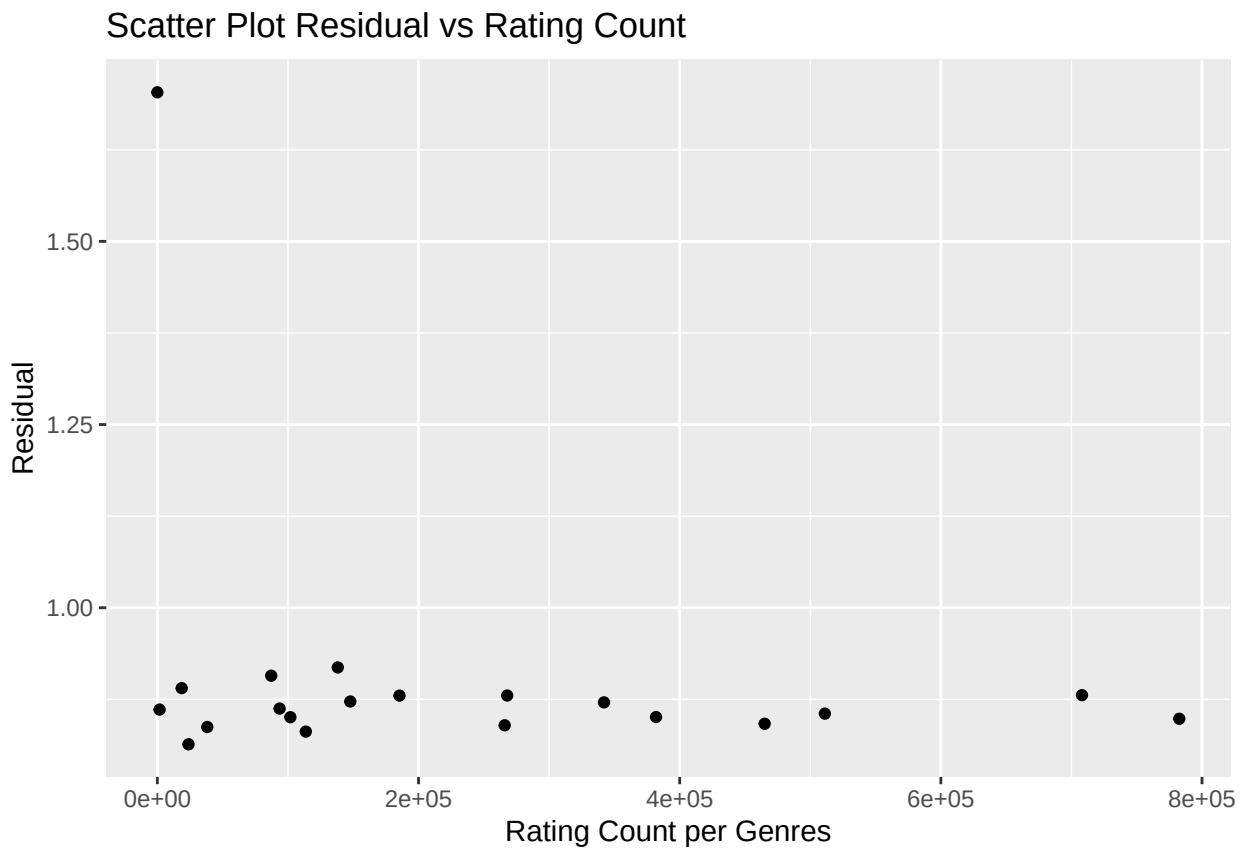
```
##                                 method      RMSE
## 1                              Average 1.0606200
## 2         Add Movie and User Average 0.8659711
## 3 Add Movie and User Profile Average 0.8635038
```

Pros

The RMSE by genres drops below 0.9. There are only two genres with RMSE>0.9.



The scatter plot shows that the accuracy is stable and mostly independent to the rating counts by genre.



Cons

There is noticeable outlier. The RMSE is 1.703 for the movie with no genres listed. Here are the records for those:

```
train_set %>% filter(genres=="(no genres listed)")
```

```
##   userId movieId rating  timestamp                title             genres
## 1   7701    8606    5.0 1190806786 Pull My Daisy (1958) (no genres listed)
## 2  10680    8606    4.5 1171170472 Pull My Daisy (1958) (no genres listed)
## 3  46142    8606    3.5 1226518191 Pull My Daisy (1958) (no genres listed)
## 4  57696    8606    4.5 1230588636 Pull My Daisy (1958) (no genres listed)
## 5  64411    8606    3.5 1096732843 Pull My Daisy (1958) (no genres listed)
## 6  67385    8606    2.5 1188277325 Pull My Daisy (1958) (no genres listed)
```

This model doesn't perform well with the movies if there is no genre listed.

3 Results

```
rmse_validation
```

```
## [1] 0.8638338
```

Model Summary

The simplest model consider no factor and is used as a baseline model. It has an RMSE of 1.06.

The second model takes the movie and user rating averages into consideration. It greatly reduce the RMSE. However, it assumes the user looks at each genres without preference. The assumption is against the common sense. And the bar chart shows that the RMSE by genres is high.

The third model improves upon model 2 by adding the user profile. The user profile analyze the historial ratings of each user. It looks at the average rating by genres for the user, and then find out which genres is most rated by the user. The model significantly improves the RMSE by recoginizing the user's preferable genres.

4 Conclusion

Summary of the report

The report starts from the simpleset model and develop by adding factors to the previous model. And I use the RMSE histogram, scatter polt and bar charts to as a guidance of what factors should be added to the model.

I like the User Profile Model, because it takes into users' preference into consideration.

Limitations

The User Profile Model performs poorly with movies that have no genres listed. Fortunately, there are only very few moveis like that (one movie in our dataset).

It also does not consider difference beween movies with single genres and multiple genres.

Future Works

In order to improve the model, I will group the movies by genres number (how many genres are listed for the movie). Then, I will observe whether the rating is impatcted by this factor. If yes, I will build a model to include the Movie Profile.

5 Reference

1 R Markdown Cheat Sheet [(https://rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf)]

2 Data Science Courses from HarvardX [(https://courses.edx.org/)]