Movie Recommendation System Assignment

Jose Ramon Hernandez Galan

8/10/2019

Introduction

This is a report showing the process and results for the creation of a model for movie recommendations.

This assingment is part of the capstone project of the Data Science course pursued by the author in HarvardX.

Dataset

The dataset used is a subset of **movielens**. the subset is composed by 10 millions of movie ratings. Data set cn be found here: **movielens-10m**

Data Wrangling

Several sets are created based on the original dataset:

- edx..... 9.000.055 obs. of 6 variables (9/10)
- validation... 999.999 obs. of 6 variables (1/10)

Model training and result validation will be carried out over these two above sets.

In order to perform parameter tuning we will split edx set into train and test.

- edx_train.... 8.100.067 obs. of 6 variables (9/10)
- edx_test..... 899.988 obs. of 6 variables (1/10)

Note we have included in the edx and edx_train set those users and movies which are alredy present in validation and test_set. So we can properly check the predicted ratings on the test sets.

Let´s observe the structure of the datasets by checking the structure of edx_test:

```
## 'data.frame':    899988 obs. of  6 variables:
##  $ userId   : int  1 1 2 3 3 3 3 3 3 3 ...
##  $ movieId  : num  316 355 1210 213 1408 ...
##  $ rating   : num  5 5 4 5 3.5 2 2 4.5 4 3.5 ...
##  $ timestamp: int  838983392 838984474 868245644 1136075789 1133571145 1133571139 1136075848 1164885
##  $ title    : chr  "Stargate (1994)" "Flintstones, The (1994)" "Star Wars: Episode VI - Return of th
##  $ genres   : chr  "Action|Adventure|Sci-Fi" "Children|Comedy|Fantasy" "Action|Adventure|Sci-Fi" "Dra
```

How results are measured

Error loss will be calculated using the Residual Mean Square Error (RMSE). See formula below.

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

This is how we calculate the error between the predicted rating and the real rating for each movie i provided by each user u.

Proposed models

Several models have been tested. Each model is presented by a formula.

Note that the random error $\epsilon_{u,i}$ is omitted in hereafter formulas for simplification.

Simple Model

This model is based just in the average rating across all movies i and users u.

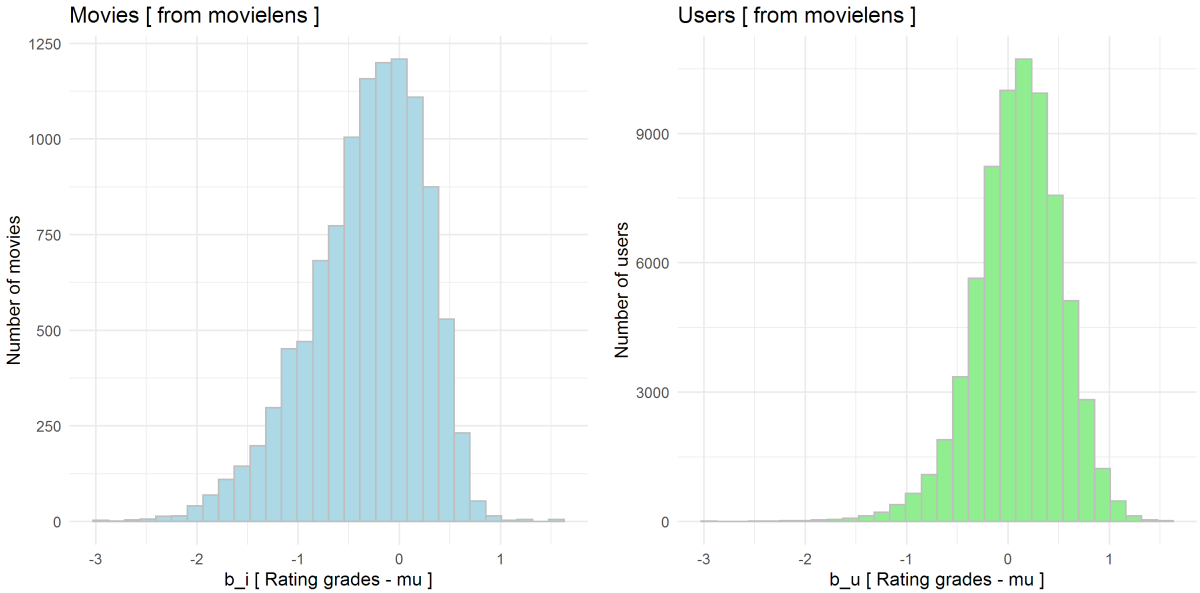
$$Y_{ui} = \mu$$

The prediction based just on the average has the following result:

```
## # A tibble: 1 x 2
##   METHOD             RMSE
##   <chr>             <dbl>
## 1 Just the average 1.0612
```

As expected the value is still not low enough. Remember we are **targeting** a value close to **0.857**, which is value obtained on the Netflix challenge.

If we have a look at the deviation from the average of ratings for each movie and each user we notice there is some variability due to different movies and different users. See charts below.



Most of the predictions are close to the average (those around 0), but there are others which are far from the average. These deviations are due to movie-to-movie variability and user-to-user variability. Hereafter, these effects are know as **movie effect** and **user effect**.

Let´s model these effects.

Modeling Movie and User effects

$$Y_{ui} = \mu + b_i + b_u$$

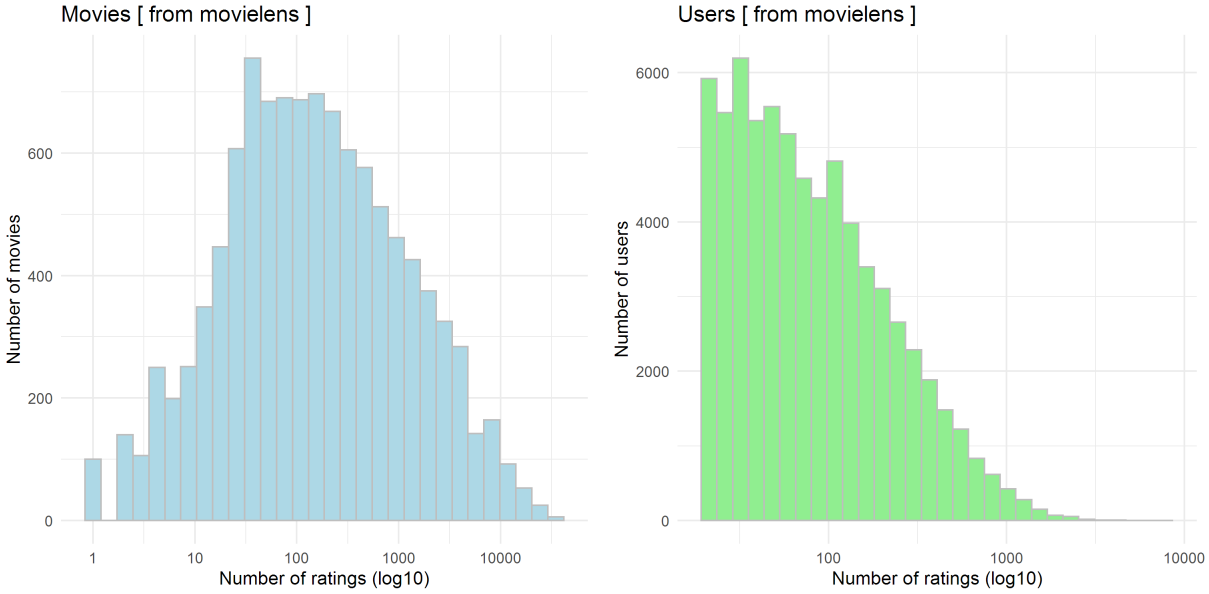
```model
# Predictions  ( Yui = mu + b_i + b_u )
  predicted_ratings <- validation %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_i + b_u) %>%
  .$pred

  model_2_rmse <- RMSE(validation$rating, predicted_ratings)
  rmse_results <- bind_rows(rmse_results,
                            data_frame(method="Movie + User Effect Model",
                                       RMSE = model_2_rmse ))
```

```
## # A tibble: 3 x 2
##   METHOD                      RMSE
##   <chr>                      <dbl>
## 1 Just the average           1.0612
## 2 Movie Effect Model         0.94391
## 3 Movie + User Effect Model 0.86535
```

The results have **improved remarkably**.

Now, let´s assess if **regularization** can help on improving our mmodel. So let´s observe how many ratings each movie and each user is in the movielens dataset. See charts below.



It seems not to be high amount of low rated movies nor users with few ratings.

In spite of the user ratings with less than 20 ratings have been dropped from the original database and the proportion of users with less than 30 ratings is poor (17.5%) we will add some regularation to the model. Furthermore, the movies with **less than 5 ratings represents 4.5% of the movies** and we may have some finding here.

Let´s see if regularization improves our model.

Regularizing Movie and User effects

$$Y_{ui} = \mu + b_i(\lambda) + b_u(\lambda)$$

Now b_i and b_u are calculated being penalized mainly when ratings are low. See r chunk as follows:

```
l <- best_lambda        # Obtained after optimization
mu <- mean(edx$rating)
b_i <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n() + l))

b_u <- edx %>%
  left_join(b_i, by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - mu)/(n() + l))
```

See results after regularization as follows:

```
## # A tibble: 5 x 2
##   METHOD                                    RMSE
##   <chr>                                    <dbl>
## 1 Just the average                        1.0612
## 2 Movie Effect Model                      0.94391
## 3 Movie + User Effect Model               0.86535
## 4 Regularized Movie Effect Model          0.94385
## 5 Regularized Movie + Regularized User Model 0.86482
```

Modeling Genre effects

$$Y_{ui} = \mu + b_i(\lambda) + b_u(\lambda) + b_g(\lambda)$$

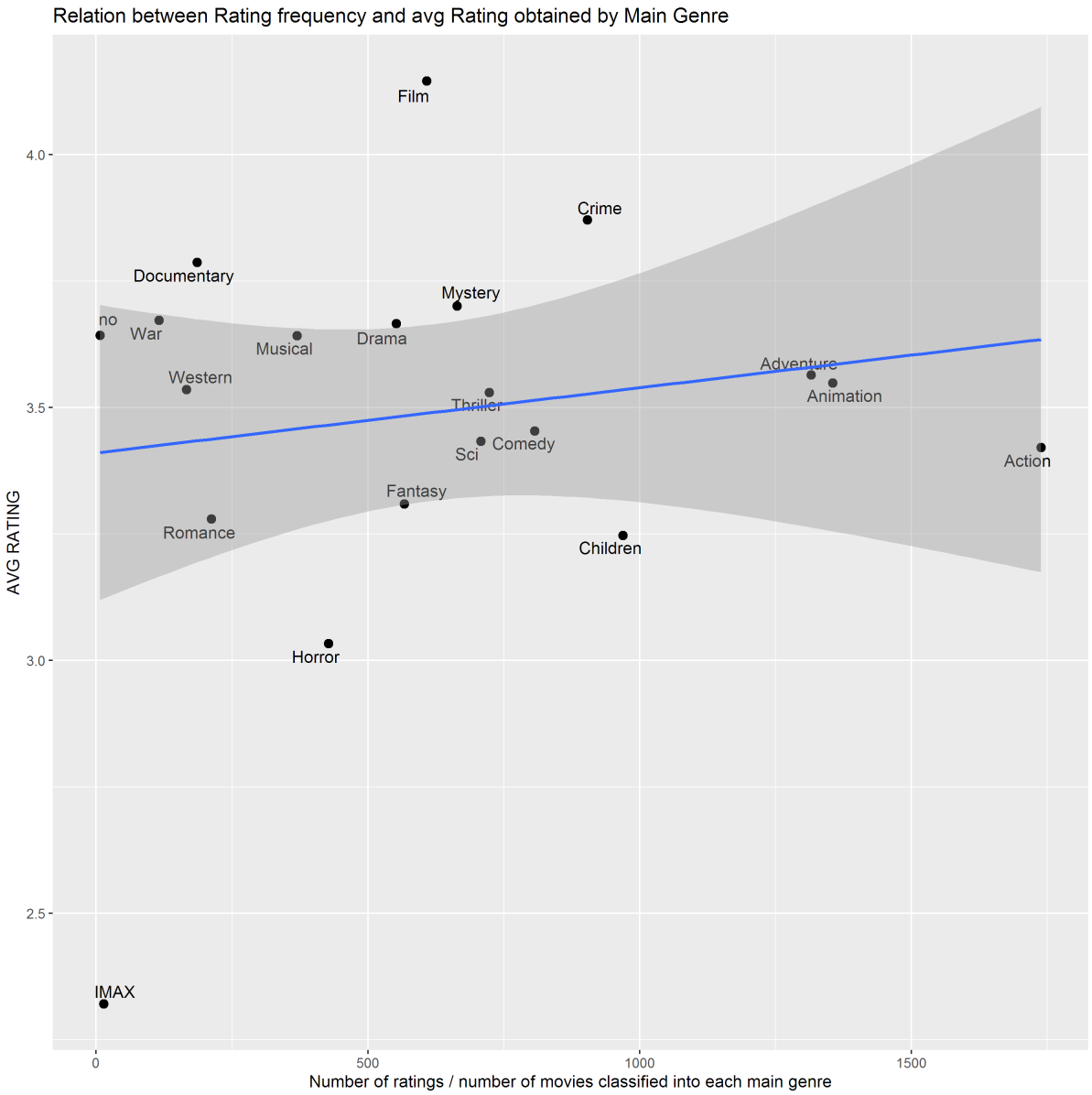
See results after regularization as follows:

```
## # A tibble: 6 x 2
##   METHOD                                                    RMSE
##   <chr>                                                    <dbl>
## 1 Just the average                                        1.0612
## 2 Movie Effect Model                                      0.94391
## 3 Movie + User Effect Model                               0.86535
## 4 Regularized Movie Effect Model                          0.94385
## 5 Regularized Movie + Regularized User Model              0.86482
## 6 Regularized Movie + Regularized User Effect Model + Reg. Genre 0.86445
```

As seen the improvement is not very significant. The genre column has been used as it is: with all the genres together.

I will try to extract the main genre for each movie.

I have done some classification based on the main genre, which was extracted from genres column. The first of the list is observed as the main genre. The number of groups has decresased from 787 to 20 groups.



No strong correlation is observed between number of ratings and avg rating obtained if we group by main genre.

Final Results

The following table shows the summary of the resutls obtained through different models.

```
## # A tibble: 6 x 2
##   METHOD                                                    RMSE
##   <chr>                                                    <dbl>
## 1 Just the average                                        1.0612
## 2 Movie Effect Model                                      0.94391
## 3 Movie + User Effect Model                               0.86535
## 4 Regularized Movie Effect Model                          0.94385
## 5 Regularized Movie + Regularized User Model              0.86482
## 6 Regularized Movie + Regularized User Effect Model + Reg. Genre 0.86445
```

Conclusions

- Although the average model is just too simple the results improves remarkably when the movie and the user effects are added.
- **Regularizaion** barely improves the performance of the model because most of the spurious observations of users with very low ratings have been previously dropped.
- The **Genre effect** seems important but for any reason the results are just slightly better. For model computation the genre column has not been modified and all genres are used as one predictor. Maybe if the model separates the specifics models and properly weighted we may find some improvement.
- **Future work**: Other approach that could be important is the rating time respect to the release of the movie.