Movielens Project: predictig a movie’s rating

DATA SCIENCE PROFESSIONAL CERTIFICATE

Movielens Project: predicting a movie’s rating

Marta Acedos Serrano

20/2/2021

1. Introduction

The dataset Movielens is downloaded from the website “http://files.grouplens.org/datasets/movielens/ml-10m.zip” and it contains information about different movies. More precisely, it contains information about the rating given to each movie of the dataset, as well as information such as the title, genre, movie Id and release year of the movie, the rating date and the user Id. The purpose of this project is to predict the rating of a movie given its characteristics. In order to predict the rating, firstly an exploratory data analysis is conducted to obtain an insight on how the different characteristics affect the rating of the different movies. Once this analysis is conducted, different models are constructed to predict the rating given the movie’s characteristics and the best model is selected. Afterwards the validation dataset is used to predict the movies’ ratings with the selected model and using those predictions the root-mean-square error (RMSE) is computed.

To develop this analysis the dataset Movielens is downloaded and splitted into edx (training set) -which contains 90% of the data- and validation (test set) -which contains 10% of the data. Although not included in the report, this code is available on the script. Afterwards the data is cleaned (remove NAs and modify class of some variables) and some variables are added to enhance the analysis.

2. Analysis Section

2.1. Data Management

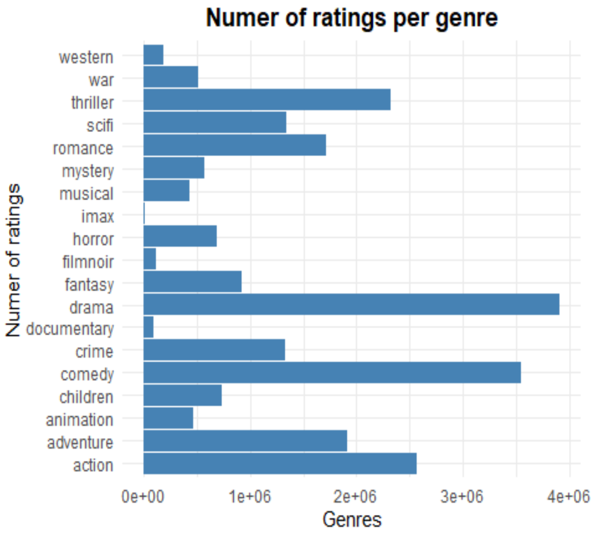
Once the dataset has been downloaded and splitted, some variables are modified and some variable are added in order to better analyze the data. First of all, since movies can have more than one genre, boolean variables for each genre are added to identify separately the genres of a movie. This is, if a movie has the genres “Action” and “Comedy”, then the corresponding variables g_action and g_comedy will have a value of 1, and the rest of the genre variables will have a value of 0. Furthermore, the release year was originally included in the title, so title and release year are separated and stored in 2 different variables. Once the release year is extracted, the rating date (timestamp) is given a date format and the difference between the release year and the rating year is computed (variable years_time, which indicates the number of years between the release date of the movie and the date in which the movie is rated). These variables are also added in the validation set, although the code is not included -but it is available in the script. Furthermore, the NAs are removed by selecting only the rows that don’t have any missing values.

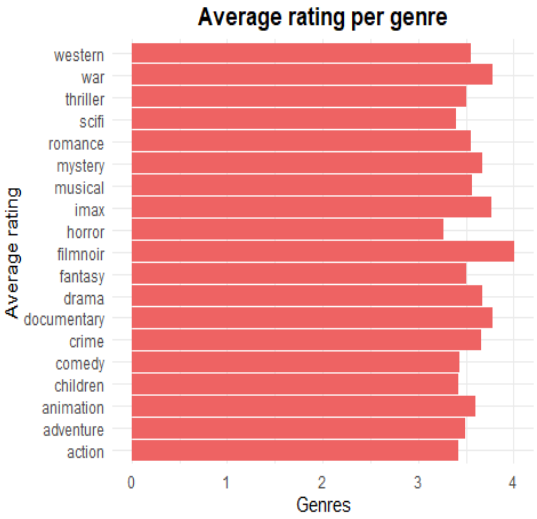
```r
#add genres variables
edx <- edx %>% mutate(g_adventure = ifelse(str_detect(genres, "Adventure")==TRUE
,1,0))
edx <- edx %>% mutate(g_comedy = ifelse(str_detect(genres, "Comedy")==TRUE,1,0))
edx <- edx %>% mutate(g_action = ifelse(str_detect(genres, "Action")==TRUE,1,0))
edx <- edx %>% mutate(g_animation = ifelse(str_detect(genres, "Animation")==TRUE
,1,0))
edx <- edx %>% mutate(g_children = ifelse(str_detect(genres, "Children")==TRUE,1
,0))
edx <- edx %>% mutate(g_fantasy = ifelse(str_detect(genres, "Fantasy")==TRUE,1,0
))
edx <- edx %>% mutate(g_scifi = ifelse(str_detect(genres, "Sci-Fi")==TRUE,1,0))
edx <- edx %>% mutate(g_drama = ifelse(str_detect(genres, "Drama")==TRUE,1,0))
edx <- edx %>% mutate(g_romance = ifelse(str_detect(genres, "Romance")==TRUE,1,0
))
edx <- edx %>% mutate(g_thriller = ifelse(str_detect(genres, "Thriller")==TRUE,1
,0))
edx <- edx %>% mutate(g_crime = ifelse(str_detect(genres, "Crime")==TRUE,1,0))
edx <- edx %>% mutate(g_horror = ifelse(str_detect(genres, "Horror")==TRUE,1,0))
edx <- edx %>% mutate(g_war = ifelse(str_detect(genres, "War")==TRUE,1,0))
edx <- edx %>% mutate(g_mystery = ifelse(str_detect(genres, "Mystery")==TRUE,1,0
))
edx <- edx %>% mutate(g_musical = ifelse(str_detect(genres, "Musical")==TRUE,1,0
))
edx <- edx %>% mutate(g_documentary = ifelse(str_detect(genres, "Documentary")==
TRUE,1,0))
edx <- edx %>% mutate(g_western = ifelse(str_detect(genres, "Western")==TRUE, 1,
0))
edx <- edx %>% mutate(g_filmnoir = ifelse(str_detect(genres, "Film-Noir")==TRUE,
1,0))
edx <- edx %>% mutate(g_imax = ifelse(str_detect(genres, "IMAX")==TRUE, 1,0))


#extract years, add the age of movies when rated and add the number of genres pe
r movie
edx <- edx %>% mutate(g_count = str_count(genres, "\\|"))
edx <- edx %>% mutate(g_count = g_count + 1)
edx <- edx %>% mutate(title = str_replace(title,"^(.+)\\s\\((\\d{4})\\)$","\\1_\
\2" )) %>%
  separate(title,c("title","release_year"),"_")

## Warning: Expected 2 pieces. Additional pieces discarded in 8 rows [632408,
## 2106925, 3103368, 3888247, 4654285, 7646811, 8527410, 8607753].

edx$release_year <- as.numeric(edx$release_year)

## Warning: NAs introducidos por coerción

edx$timestamp <- as_datetime(edx$timestamp)
edx$timestamp <- year(edx$timestamp)
edx <- edx %>% mutate(years_time = timestamp - release_year)
#filter so that there are not movies that were rated before they were released a
nd there aren’t NAs
edx <- edx %>% filter(years_time>=0)
edx <- edx[complete.cases(edx), ]

## Warning: Expected 2 pieces. Additional pieces discarded in 1 rows [421803].
```

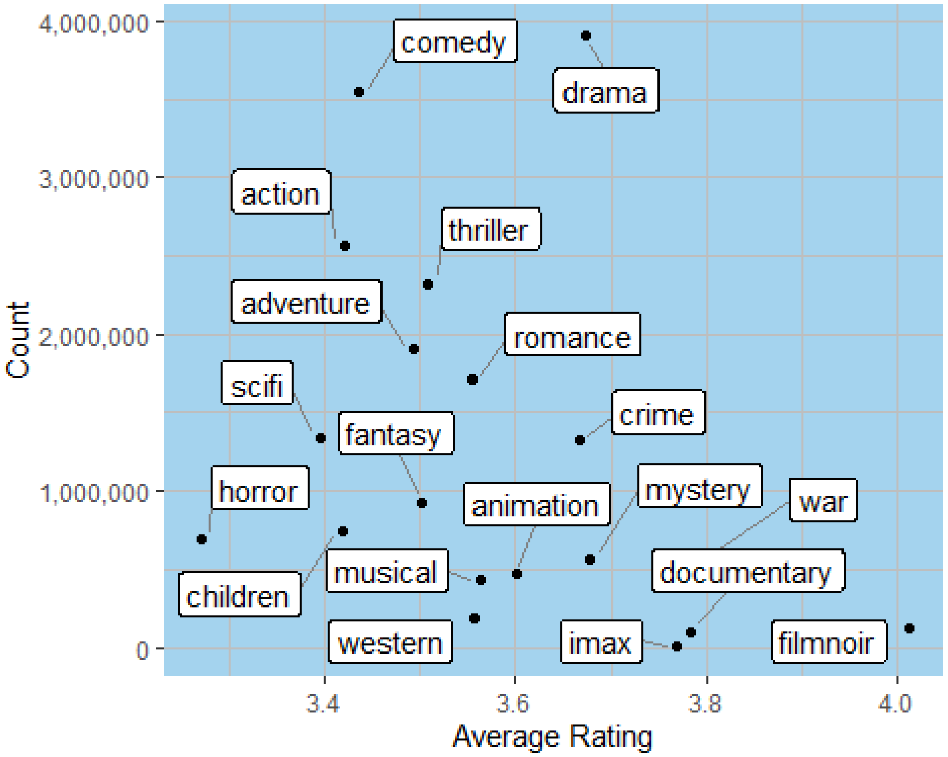
2.2. Exploratory Data Analysis

Once the variables have been added to both datasets, an exploratory analysis is conducted using the edx dataset. It is conducted using the edx dataset because the model will be trained using this dataset, and therefore the exploratory analysis that will justify which variables are used in the model is conducted in the dataset that will be used to train the model. First of all, a table summarizing the average rating and the number of ratings per genre is constructed:

Genre	Count	Average Rating
adventure	1908892	3.493544
comedy	3540907	3.436911
action	2560541	3.421406
animation	467168	3.600644
children	737994	3.418715
fantasy	925635	3.501949
scifi	1341183	3.395743
drama	3910003	3.673154
romance	1712071	3.553819
thriller	2325841	3.507679
crime	1327695	3.665938
horror	691451	3.269801
war	511144	3.780823
mystery	568329	3.677008
musical	433080	3.563305
documentary	93066	3.783487
western	189393	3.555921
filmnoir	118541	4.011625
imax	8181	3.767693



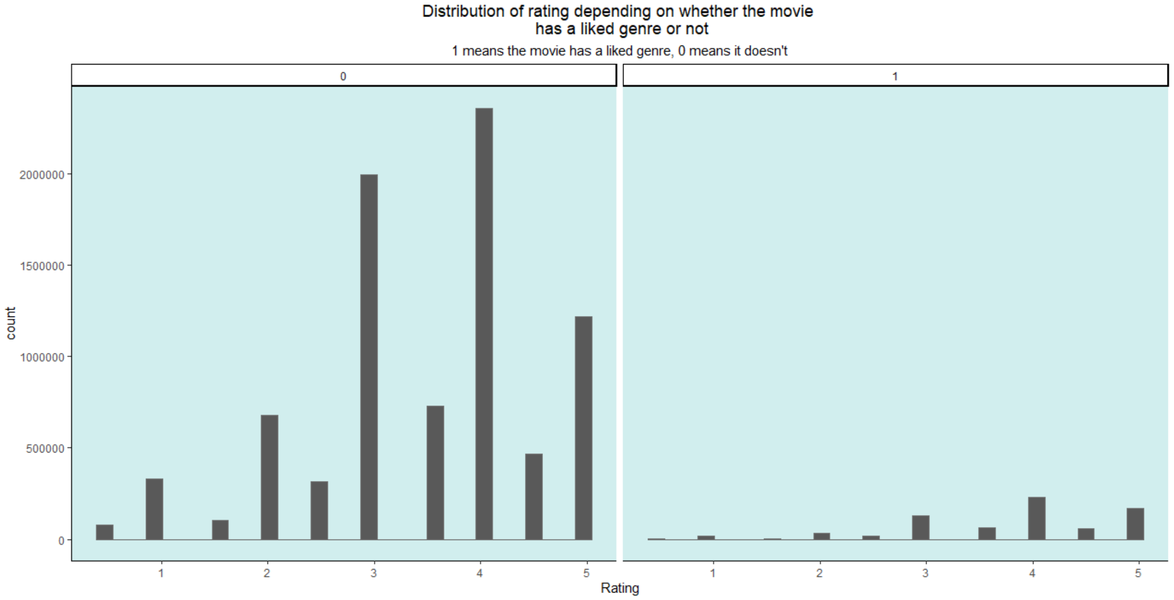




As it can be seen in the plot, some genres are particularly liked (they have high average ratings). This is, the genres "Film Noir", "War", "Documentary" and "IMAX" tend to have higher average ratings. Thus, a variable is constructed indicating whether the movie contains one of the gernes which are particularly liked or not. The purpose of this variable is to study whether a movie that has one of those gernes will tend to systematically have a higher rating.

```
#create boolean variables to represent whether the genre is a liked genre or not
edx <- edx %>% mutate(liked_genre = ifelse(g_filmnoir==1| g_documentary==1| g_wa
r==1| g_imax==1 ,1,0))


edx %>% ggplot(aes(rating)) + geom_histogram(color="gray43") + facet_wrap(~liked
_genre) + labs(title="Distribution of rating depending on whether the movie \n h
as a popular genre or not" ,x="Rating",  subtitle="1 means the movie has a popul
ar (liked) genre, 0 means it doesn't")  + theme_classic() + theme(plot.title = e
lement_text(hjust=0.5), plot.subtitle = element_text(hjust=0.5),panel.grid.major
= element_blank(), panel.grid.minor = element_blank(), panel.background = elemen
t_rect(fill="lightcyan2"))
```

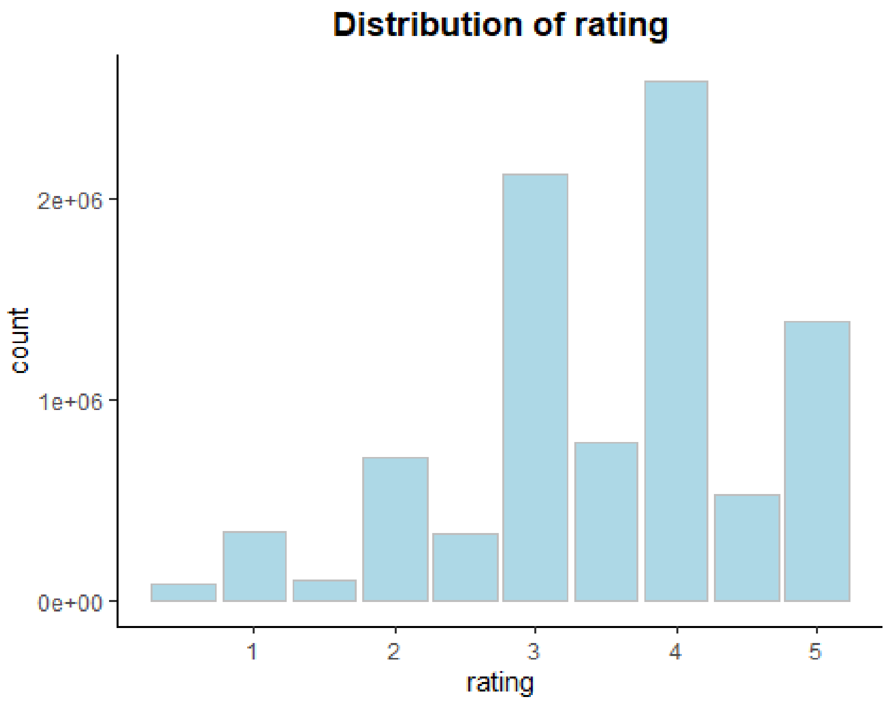
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



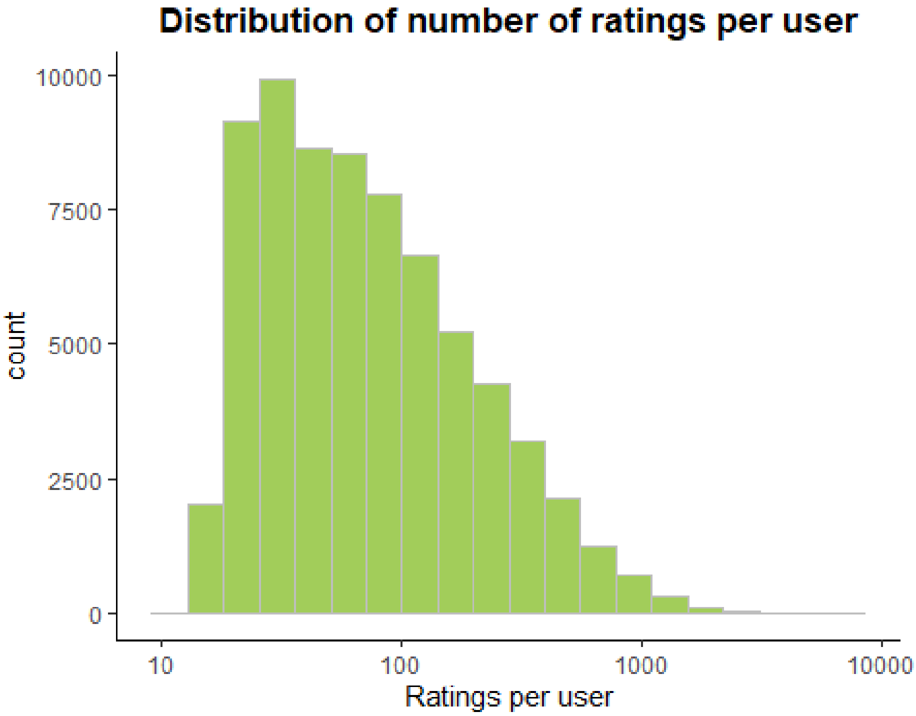
As it can be observed in the plots, the distribution of the ratings of movies containing one of the 4 most liked genres (Film Noir, War, Documentary or IMAX) seems to me more concentrated in ratings above 3.

Once the genre effect has been analyzed, the rest of the variables are explored. First of all, the distribution of each variable is studied through histograms:

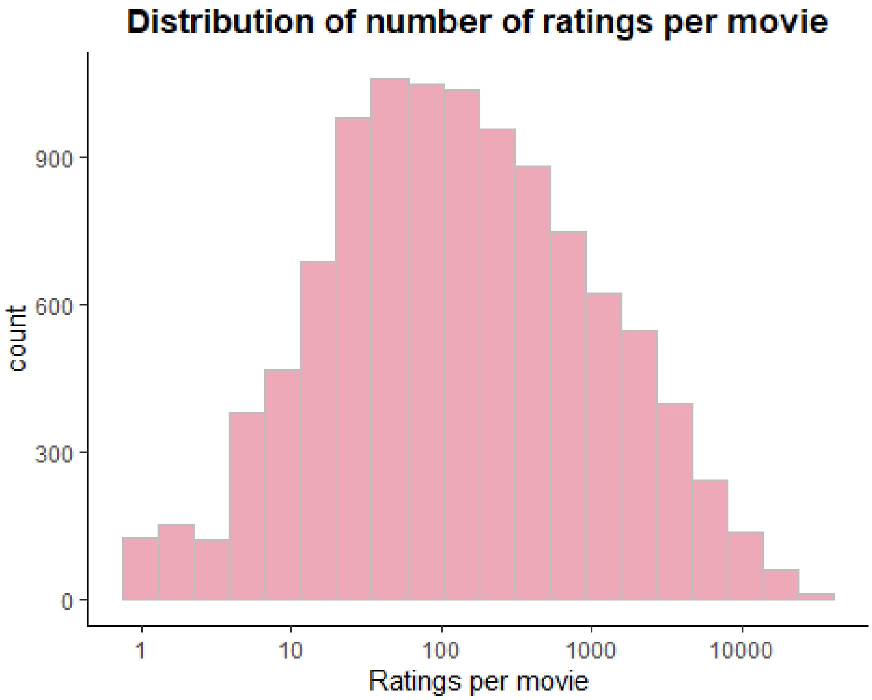
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



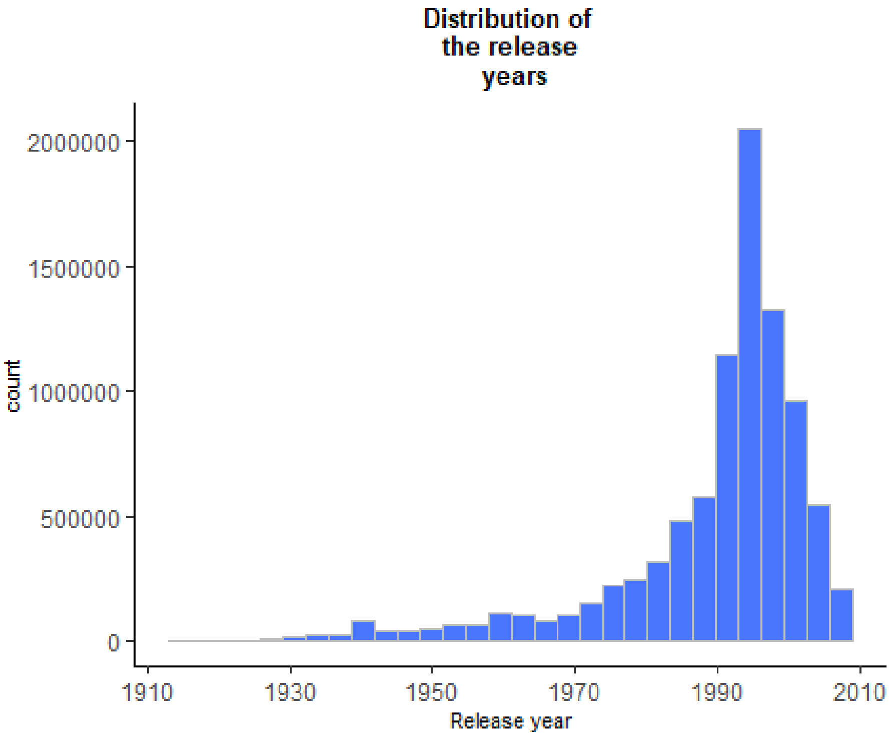
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

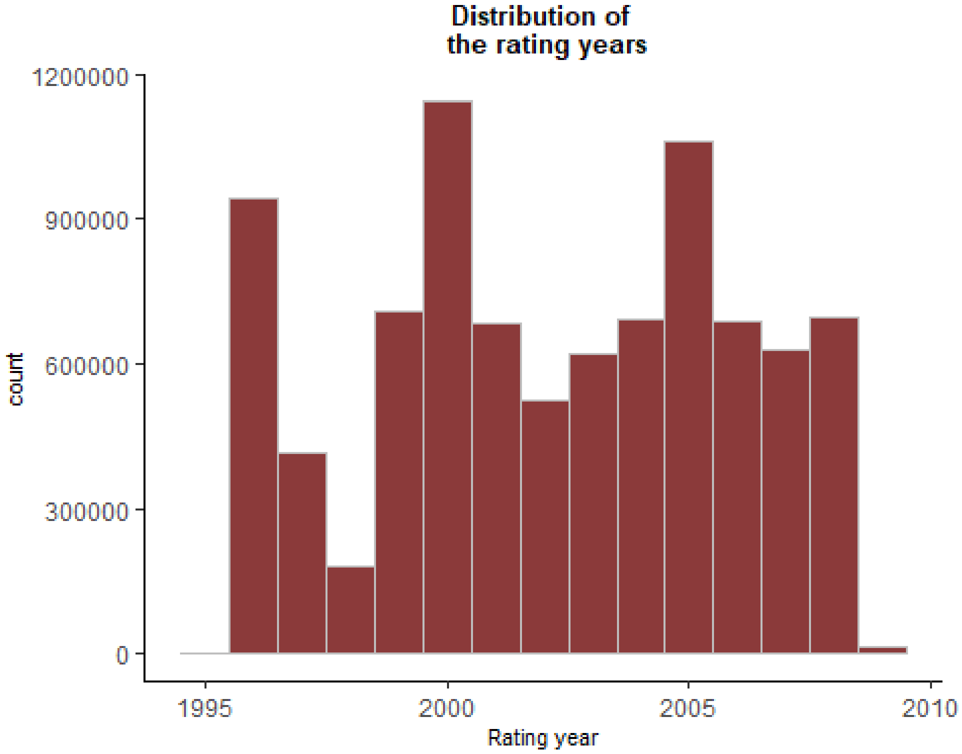


```
## `summarise()` ungrouping output (override with `.groups` argument)
```

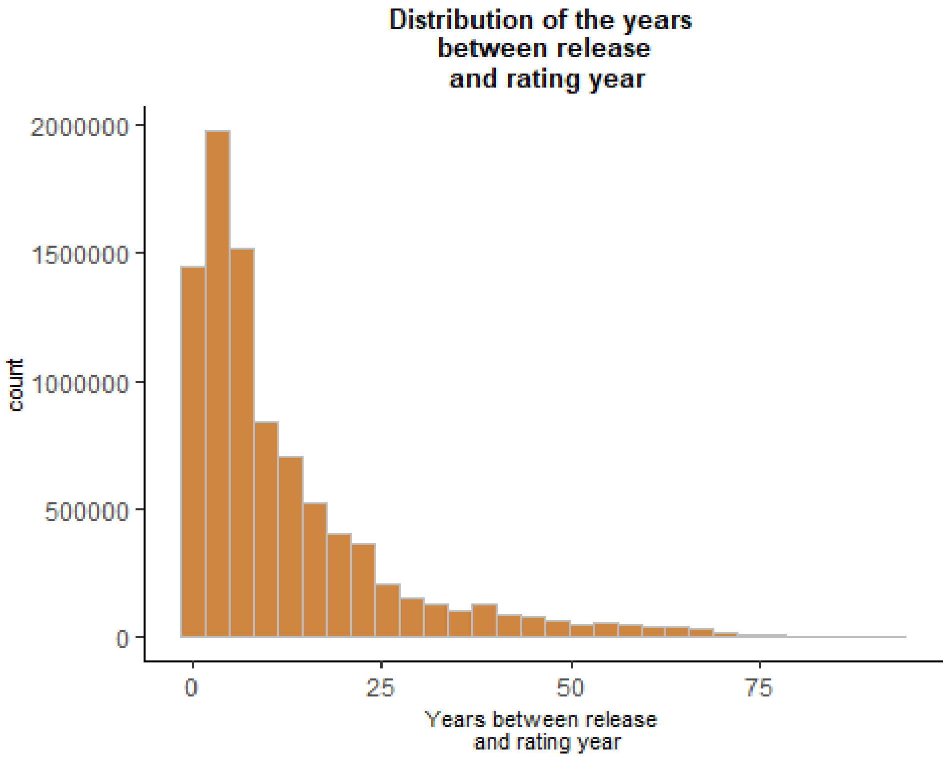


```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



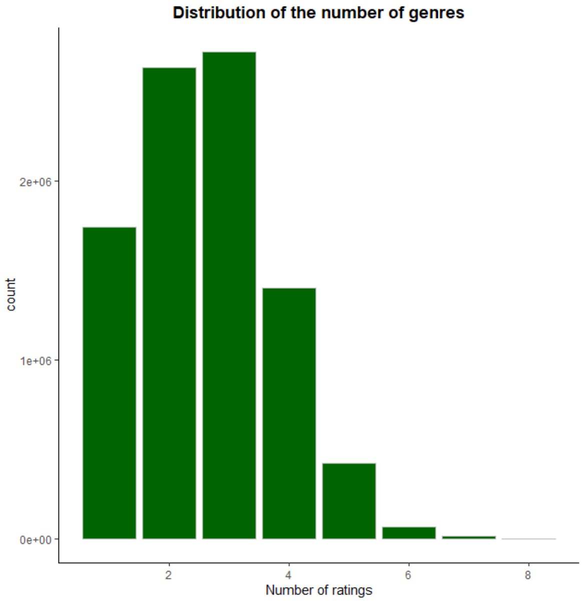


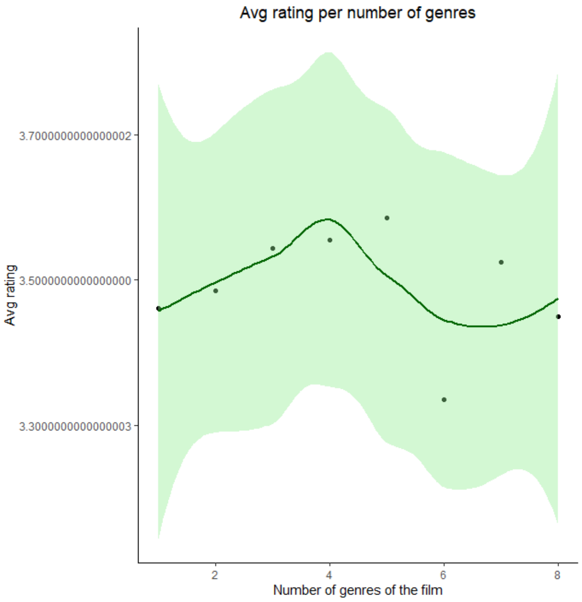
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In this next section the relationship between the different variables and the variable “rating” is explored:

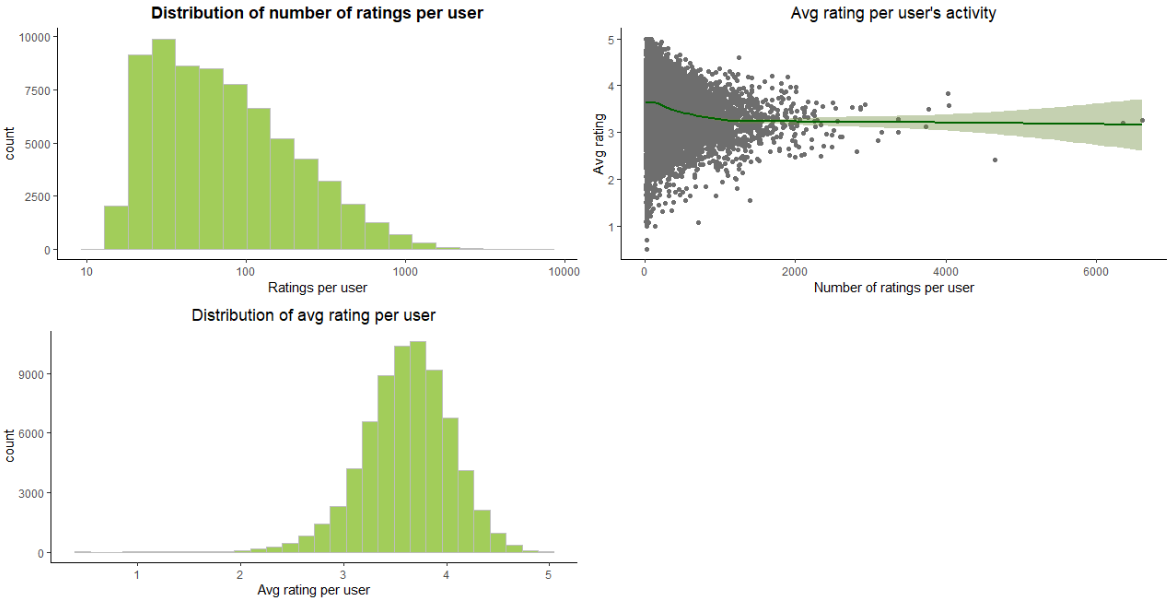
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





As it can be seen in the graph, up to 4 genres the number of genres of a movie seems to be positively correlated with the rating. However, having more than 4 genres seems to have a negative impact on the movie's rating. As a result, it is not clear from the graphs whether this variable is positively or negatively correlated with "rating", and this relationship is furtherly explored in the following sections.

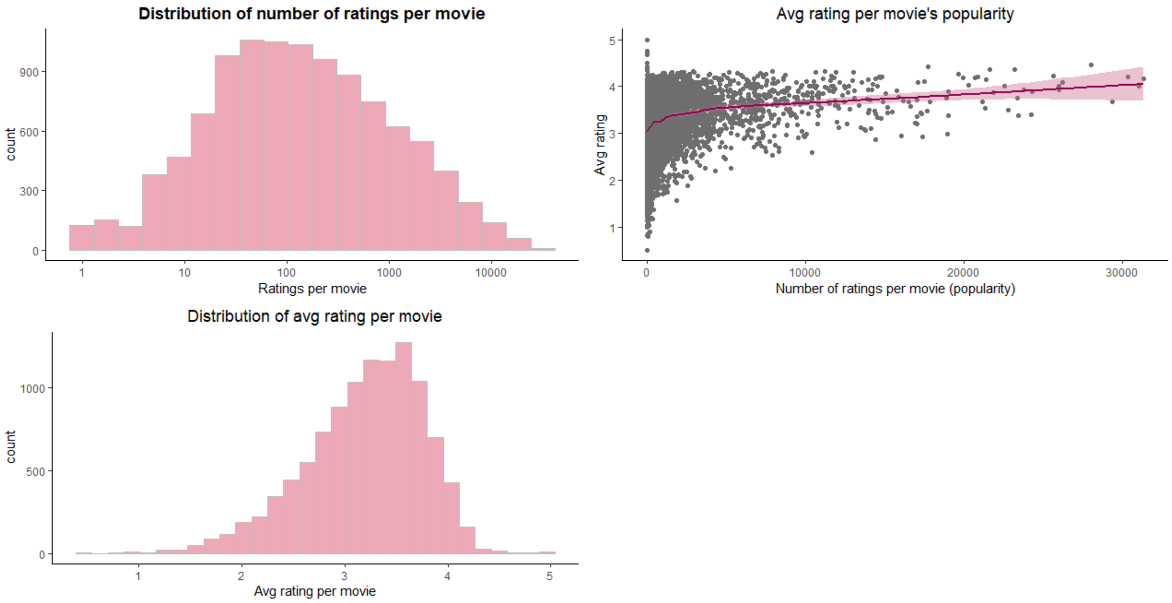
```
## `summarise()` ungrouping output (override with `.groups` argument)
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The previous graphs imply that there isn't a relationship between the user's activity and the rating given to the movies. This is, it seems that the more movies the user has rated is not related to the rating, so more activity does not lead to higher or lower ratings. On the other hand, the histogram of the distribution of the average rating per user shows that some users tend to give, on average, higher ratings. Therefore, the variable "average user", which depicts the average rating given by the user, is added to capture this user effect. Additionally, the variable "user activity", which depicts the number of ratings given by the user, is added to explore in the following sections if this variable is indeed not related to rating -as the plot suggests.

```
edx_user_act <- edx %>% group_by(userId) %>% summarize(user_activity = n())

## `summarise()` ungrouping output (override with `.groups` argument)

edx <- left_join(edx, edx_user_act, by="userId")

validation_user_act <- validation %>% group_by(userId) %>% summarize(user_activi
ty = n())

## `summarise()` ungrouping output (override with `.groups` argument)

validation <- left_join(validation, validation_user_act, by="userId")
avg_user <- edx %>% group_by(userId) %>% summarize(avg_user = mean(rating))
```

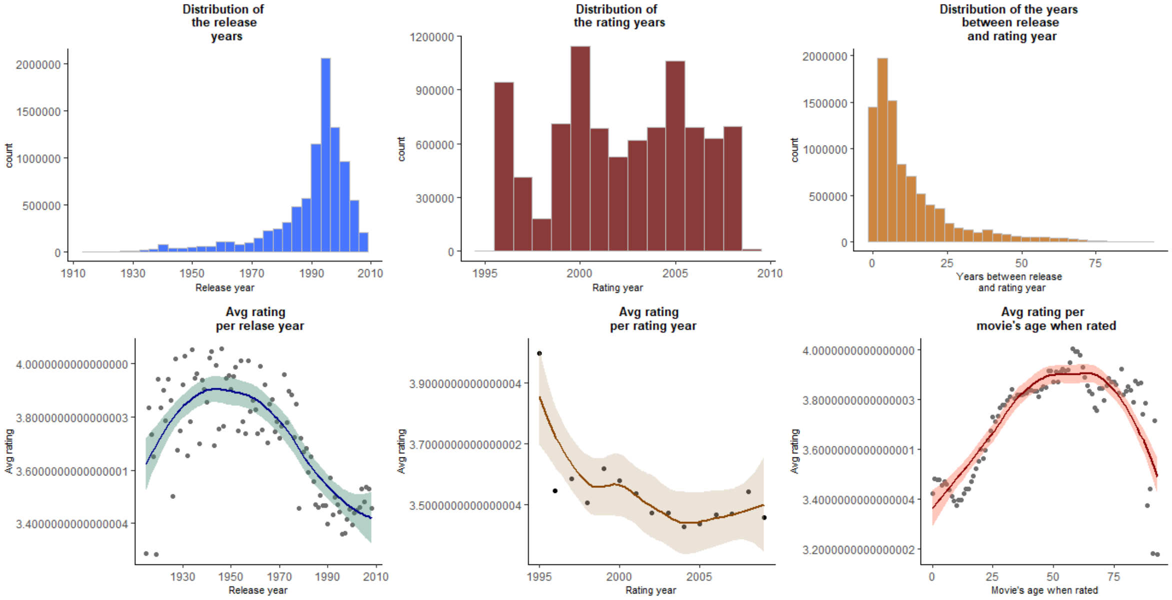
```
## `summarise()` ungrouping output (override with `.groups` argument)

edx <- edx %>% left_join(avg_user, by="userId")
avg_user <- validation %>% group_by(userId) %>% summarize(avg_user = mean(rating
))

## `summarise()` ungrouping output (override with `.groups` argument)

validation <- validation %>% left_join(avg_user, by="userId")

## `summarise()` ungrouping output (override with `.groups` argument)
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



As it can be seen in the graphs, the movie's popularity seems to be positively correlated with the movie's rating. This is, the more often a movie is rated the more likely it is that the movie will have a higher average rating. Thus, the variable popularity, which depicts the number of ratings a movie has received, is added to capture the effect of a movie's popularity on its rating. Additionally, the histogram of the distribution of the average rating per movie shows that some movies tend to have, on average, higher ratings. Consequently, the variable average movie - which depicts the average rating of the movie- is added to capture the movie effect:

```
edx_movie_pop <- edx %>% group_by(movieId) %>% summarize(popularity = n())

## `summarise()` ungrouping output (override with `.groups` argument)

edx <- left_join(edx, edx_movie_pop, by="movieId")

validation_movie_pop <- validation %>% group_by(movieId) %>% summarize(popularit
y = n())

## `summarise()` ungrouping output (override with `.groups` argument)
```

```
validation <- left_join(validation, validation_movie_pop, by="movieId")
avg_movie <- edx %>% group_by(movieId) %>% summarize(avg_movie = mean(rating))

## `summarise()` ungrouping output (override with `.groups` argument)

edx <- edx %>% left_join(avg_movie, by="movieId")
avg_movie <- validation %>% group_by(movieId) %>% summarize(avg_movie = mean(rat
ing))

## `summarise()` ungrouping output (override with `.groups` argument)

validation <- validation %>% left_join(avg_movie, by="movieId")

## `summarise()` ungrouping output (override with `.groups` argument)
## `summarise()` ungrouping output (override with `.groups` argument)
## `summarise()` ungrouping output (override with `.groups` argument)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



When it comes to the release year, it seems that up to 1940-1950 a more recent release year leads to a higher average rating, but after 1940-1950 a more recent release year implies a worse average rating. As it is not clear whether the relationship between rating and the release year is positive or negative this relationship is furtherly studied in the next sections. Regarding the rating year, up to 2005 the average rating is worse per year, but after 2005 the average rating seems to be better each year. This implies that there is a relationship between the rating year and the average rating: older rating years tend to have higher ratings. However, after 2005 this relationship seems to be different: more recent rating years tend to have better ratings. Thus, this relationship needs to be furtherly explored in the following sections in order to identify if the relationship between the rating year and the rating is positive or negative. Finally, when it comes to the years between the release date and the rating date -movie's age when rated or "years_time"-, it seems that there is a positive relationship between the variable "years_time" and "rating": more years between the release date and the rating date tend to lead to higher ratings. However, the graph also shows that when the difference between the release date and the rating date is too big (more than 75 years) the rating tends to be lower. Thus, this relationship also needs to be furtherly examined in the following sections in order to conclude whether the relationship between years_time and rating is actually positive.

2.3. Correlations

Once the relationships of the different variables with "rating" have been visually explored, the correlation of each variable with the variable "rating" is studied. Additionally, the test of Perason's correlation is computed.

$$H_0 : \rho_{x,y} = 0$$

$$H_1 : \rho_{x,y} \neq 0$$

The following table summarizes the correlations of each variable with "rating" and the results of the Pearson correlation test:

Variables	Correlation	Statistic	Results
Number of genres	0.03084744	92.585709	Reject Null Hypotesis
User Activity	-0.10780577	-325.310900	Reject Null Hypotesis
Movie's Popularity	0.17478983	532.564105	Reject Null Hypotesis
Release Year	-0.12071678	-364.815637	Reject Null Hypotesis
Rating Year	-0.03537536	-106.191791	Reject Null Hypotesis
Movie's Age When Rated	0.10963451	330.895806	Reject Null Hypotesis

The graphs from the previous section showed that up to 4 genres the number of genres of a movie seemed to be positively correlated with the rating. However, having more than 4 genres seemed to have a negative impact on the movie's rating. The correlations table shows that the there is a positive relationship between the number of genres, but the correlation is small. Nonetheless, according to the test of Pearson's correlation we can reject the null hypothesis and conclude that the correlation is statistically significant.

Regarding the user activity, according to the graphs this variable didn't seem to be correlated with "rating". However, we obtain a negative correlation that is, in comparison with g_count, stronger. Additionally, we can reject the null hypothesis that the correlation between "rating" and "user's activity" is 0 and conclude that the correlation is statistically significant.

On the other hand, "movie's popularity" is the variable with the strongest correlation. In the graphs this variable seemed to be the most correlated one with rating. Likewise, according with the result from the test of Pearson's correlation we can conclude that this correlation is statistically significant.

When it comes to the variables release year, rating year and movie's age when rated, only release year and movie's age when rated seemed correlated with rating according to the graphs.

This is consistent with the results obtained for the correlations as rating year has the lowest correlation. Additionally, in the previous section it was observed that older movies tended to get higher ratings except when the age of the movie when rated was too big. Nonetheless, in this section it is observed that the correlation between rating and the movie's age when rated is positive. Similarly, in the graphs it was not clear whether release year and rating were positively or negatively correlated. In this section it is obtained that this correlation is negative: the more recent the release year the lower the rating. Furthermore, for the 3 variables (release year, rating year and movie's age when rated) the null hypothesis of the test of Pearson's correlation can be rejected and it can be concluded that the correlation of each variable with rating is statistically significant.

In the case of the boolean variable "liked_genre" the difference in the average of "rating" for each category of this variable is studied. Additionally, the two-samples t test is computed to test whether this difference is statistically significant or not.

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

The following tables summarize the average differences in "rating" for each category of the boolean variable "liked_genre" as well as the results for the two-samples t test:

Genre not liked	**Liked genre**
3.4855102	3.8189594

Variables	**Statistic**	**Results**
Liked Genre	-272.47474	Reject Null Hypotesis

As it has been explained previously, some genres (Film Noir, Documentary, War, Imax) tend to have higher average ratings. The tables show that movies that contain those genres have, on average, higher ratings that movies that don't have those genres, and according to the two-samples t test, this difference is statistically significant -as we can reject the null hypothesis that the average of the two groups is equal. Thus, including this variable can help us predict the movie's rating since the fact of having one of those genres implies that the movie will be likely to have a higher rating.

2.4. Model Development

The previous section analyzed the different variables and their relationship with the variable "rating". In this section different models are built in order to predict the rating of a movie. First of all, since the "validation" set shall only be used to test the selected model, the "edx" dataset is splitted into "train_edx" and "test_edx". This partition is made so that in the model development section the different models can be tested without using the "validation" dataset. This is, "train_edx" is used to train the different models and "test_edx" is used to test each model. Thus, each model trained with the "train_edx" dataset is used to predict the rating of the movies of the "test_edx" dataset and the RMSE is computed. Additionally, the model is also tested on the "train_edx" dataset.

We start using a Generalized Linear Model (GLM) model with only one predictor: the variable g_count (number of genres of the movie).

```
fit_glm1 <- glm(rating~g_count, data=edx)
pred_glm1 <- predict(fit_glm1, validation)
rmse_glm1 <-sqrt(mean((pred_glm1-validation$rating)^2))
pred_train_glm1 <- predict(fit_glm1, edx)
rmse_train_glm1 <-sqrt(mean((pred_train_glm1-edx$rating)^2))
rmse_glm1

## [1] 1.060715
```

Since only one variable is used to predict the rating the model does not have much predictive power: the predictions are not very accurate and the RMSE is very high. Furthermore, in the previous section it was analyzed that the variables number of genres, user activity, movie's popularity, release year, rating year and movie's age when rated were correlated with rating. Furthermore, it was concluded that this correlation is statistically significant. Therefore, the variables number of genres, user activity and movie's popularity are subsequently added to the model. On the other hand, if the variables release year, rating year and movie's age when rated were all included in the model there would be a multicollinearity problem because "movie's age when rated" is defined through the variables "release year" and "rating year" -there is a linear relationship between these variables. Therefore, one of the three variables must be excluded. Rating year is excluded since it is the variable with the lowest correlation with rating.

On the other hand, the code used to construct the models resulting from subsequently adding the variables is not included in the report because otherwise it would be too long. However, the code is included in the script. The following models include the following variables:

-GLM 2: number of genres + user activity

-GLM 3: number of genres + user activity + movie's popularity (number of ratings)

-GLM 4: number of genres + user activity + movie's popularity (number of ratings) + release year

-GLM 5: number of genres + user activity + movie's popularity (number of ratings) + release year + movie's age when rated

-GLM 6: number of genres + user activity + movie's popularity (number of ratings) + release year + movie's age when rated + liked_genre (indicator whether the genre is Film-Noir, Documentary, War or IMAX)

Model	RMSE.validation	RMSE.edx
GLM 1	1.0607734	1.0597129
GLM 2	1.0547165	1.0537745
GLM 3	1.0420543	1.0414074
GLM 4	1.0323314	1.0317494
GLM 5	1.032041	1.0314403
GLM 6	1.0297049	1.0290386

In the previous section it was identified that some users tend to give higher ratings and that some movies tend to have higher ratings. Thus, the variables average user and average movie were added to capture the movie and user effects. Those variables are added to the model to take into account that certain movies have higher ratings and that certain users give higher ratings.

GLM 7:

$$Rating_i \ = \ Number\ of\ Genres_i \ + \ Average\ Rating\ User_i \ + \ User\ Activity_i \ + \ Movie's\ Popularity_i \ + \ Release\ Year_i \ + Movie's\ Age\ When\ Rated_i \ + \ Liked\ Genre_i \ + \ e_i$$

GLM 8:

$$Rating_i \ = \ Number\ of\ Genres_i \ + \ Average\ Rating\ User_i \ + Average\ Rating\ Movie_i \ + \ User\ Activity_i \ + \ Movie's\ Popularity_i \ + \ Release\ Year_i \ + Movie's\ Age\ When\ Rated_i \ + \ Liked\ Genre_i \ + \ e_i$$

```r
#user effect
fit_glm7 <- glm(rating~g_count + avg_user + user_activity + popularity + release
_year + years_time +liked_genre, data= train_edx)
pred_glm7 <- predict(fit_glm7, test_edx)
rmse_glm7 <-sqrt(mean((pred_glm7- test_edx$rating)^2))
pred_train_glm7 <- predict(fit_glm7, train_edx)
rmse_train_glm7 <-sqrt(mean((pred_train_glm7- train_edx$rating)^2))
#movie effect
fit_glm8 <- glm(rating~g_count + avg_user + avg_movie + user_activity + populari
ty + release_year + years_time + liked_genre, data= train_edx)
pred_glm8 <- predict(fit_glm8, test_edx)
rmse_glm8 <-sqrt(mean((pred_glm8- test_edx$rating)^2))
pred_train_glm8 <- predict(fit_glm8, train_edx)
rmse_train_glm8 <-sqrt(mean((pred_train_glm8- train_edx$rating)^2))
rmse_glm7

## [1] 0.9489009

rmse_glm8

## [1] 0.870019
```

Once the user and movie effects have been taken into account the RMSE is significantly reduced. Some users tend to give higher ratings. Thus, a movie rated by one of those users is likely to receive a higher rating. Likewise, some movies tend to have higher ratings than other, so those movies are likely to receive higher ratings just because of the movie itself. As a result, when both effects (user effect and movie effect) are taken into account the predictive power of the model is significantly increased.

Model	RMSE.validation	RMSE.edx
GLM 1	1.0607734	1.0597129
GLM 2	1.0547165	1.0537745
GLM 3	1.0420543	1.0414074
GLM 4	1.0323314	1.0317494
GLM 5	1.032041	1.0314403
GLM 6	1.0297049	1.0290386
GLM 7	0.9489009	0.9493591
GLM 8	0.870019	0.870236

The model that yields the best results is the last one (GLM 8). Using this model and the "edx" dataset a cross-validation algorithm is deployed. However, due to computing limitations only a five-fold cross-validation is used:

```
control <- trainControl(method="cv", number=5, p =0.9)
train_glm <- train(rating~g_count + avg_user + avg_movie + user_activity + popul
arity + release_year + years_time + liked_genre, method="glm", data=edx, trContr
ol=control)
train_glm

## Generalized Linear Model
##
## 8999872 samples
##       8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 7199897, 7199898, 7199899, 7199897, 7199897
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.870215  0.3264404  0.6745684
```

The RMSE obtained with the cross-validation algorithm is very similar to the RMSE obtained testing the model GLM 8 with the "test_edx" dataset.

On the other hand, the development of other type of models was also attempted. More precisely, a random forest model, a decision tree (CART) model and a K-nearest neighbors model were tried, but the lack of computing power made impossible to fit any of those models. Thus they haven't been included neither in the report nor the script since it was not possible to run the code with those models.

3. Results

As it has been argued in the previous section, the model that yields the best results is the GLM 8:

$$Rating_i \ = \ Number\ of\ Genres_i + \ Average\ Rating\ User_i \ + Average\ Rating\ Movie_i \ + \\ User\ Activity_i + \ Movie's\ Popularity_i \ + \ Release\ Year_i \ + Movie's\ Age\ When\ Rated_i \ + \\ Liked\ Genre_i \ + \ e_i$$

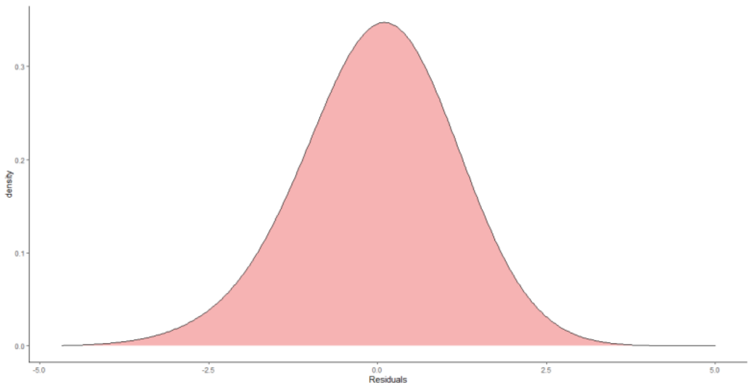
Consequently, this is the model selected, and this model is furtherly examined in this section.

First of all, the summary of the model shows that all the variables or predictors are statistically significant. However, the β coefficient estimated is very small for all variables. This means that the effect of all variables in the rating of the movie is small. This is particularly surprising in the case of the movie's popularity, since this variable was the one that had the strongest correlation with rating. Nonetheless, although this variable had the strongest correlation, this correlation was not very high.

```
##
## Call:
```

```
## glm(formula = rating ~ g_count + avg_user + avg_movie + user_activity +
##     popularity + release_year + years_time + liked_genre, data = train_edx)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.6738  -0.4993   0.0673   0.5841   4.9938
##
## Coefficients:
##                 Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    1.587e+01  1.741e-01   91.166  < 2e-16 ***
## g_count        3.346e-03  2.750e-04   12.167  < 2e-16 ***
## avg_user       8.646e-01  7.478e-04 1156.183  < 2e-16 ***
## avg_movie      9.212e-01  7.423e-04 1240.914  < 2e-16 ***
## user_activity  8.216e-05  6.264e-07  131.167  < 2e-16 ***
## popularity    -2.994e-06  5.261e-08  -56.914  < 2e-16 ***
## release_year  -9.309e-03  8.702e-05 -106.969  < 2e-16 ***
## years_time    -1.043e-02  8.745e-05 -119.210  < 2e-16 ***
## liked_genre   -3.493e-03  1.147e-03   -3.045  0.00233 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.7573115)
##
##     Null deviance: 9104785  on 8099900  degrees of freedom
## Residual deviance: 6134141  on 8099892  degrees of freedom
## AIC: 20734919
##
## Number of Fisher Scoring iterations: 2
```

Additionally, according to the following plot, the residuals seem to follow a Normal distribution with an average close to 0, which is desirable because it implies that the errors when predicting a movie's rating tend to be small and close to 0.



Finally, the predictive power of the selected model is tested using the "validation" data set: this model is employed to predict the rating of the movies of the "validation" dataset and using those predictions the RMSE is computed. Additionally, the model is also tested using the "edx" dataset in order to have an additional RMSE value. However, the concluding points are argued based on the RMSE computed using the "validation" dataset. Since the "validation" data set wasn't used at all in the model development section, the RMSE obtained using this dataset is obtained making out-of-sample predictions and therefore this RSME is the one that best measures the predictive power of the model.

The RMSE obtained testing the model on the validation dataset is 0.8453810, while the RMSE obtained using the "edx" data set is 0.8702143.

Dataset	RMSE
Validation	0.8453810

4. Conclusions

In the previous section the RMSE obtained using the "validation" dataset was 0.8453810. As it was argued previously, since this RMSE is obtained making out-of-sample predictions -because the "validation" dataset wasn't used at all in the model development section- it constitutes a good measure of the model's predictive power. In order to build the selected model an exploratory data analysis was conducted to gain an insight of the relationship between each variable and the variable "rating". In this analysis three main effects were identified: the genre, the movie and the user effects. In order to capture those effects 3 variables were added to the dataset – "liked_genre" to capture the genre effect, "avg_user" to capture the user effect and "avg_movie" to capture the movie effect. Furthermore, those variables were also added to the model in the model development section. The genre effect didn't yield a significant improvement in the RMSE, but the variables capturing the user effect and the movie effect led to an important improvement in the RMSE.

On the other hand, the selected model is a Generalized Linear Model. The RMSE obtained with this model using the "validation" dataset can be regarded to be good to some extent, but it could be improved using more sophisticated machine learning algorithms. However, the use of those algorithms is precisely one of the limitations and challenges I have encountered in this analysis: those algorithms could not be used due to computing limitations. The size of the dataset has been a great challenge, since due to this size my computer lacked enough power to implement models such as a random forest model. As it was argued before, I tried to develop a random forest, a decision tree and a K-nearest neighbors model, but I could not fit any of them due to computing limitations. Therefore, there are future work opportunities based on this project: develop those models (random forest, decision tree and K-nearest neighbors) and other models such as a Loess regression, Quadratic Discriminant Analysis (QDA)… Nonetheless, the employment of those models would require either a more powerful computer or a smaller dataset. In the case of a smaller dataset it is important to take into account that less data to train the model can lead to a loss of precision when training the model. Thus, it is important to analyze the trade-off between less available data to train the model and a more sophisticated machine learning algorithm. This analysis should answer the following question: overall, using a more sophisticated machine learning model with significantly less data leads to a better RMSE?