HarvardX - Data Science
MovieLens Project

Dieter Reinwald

2021

Contents

1 Executive summary

The goal of this project is to predict movie ratings for users based on a set of predictors. The data set used is the publicly available movielens database. Using the loss function root mean square error (RMSE), we compare different models and apply the best one to a validation set in order to compare the predictions with true ratings. The final model found had an RMSE below the target of 0.86490.

2 Overview

The following steps will be performed to reach the goal:

1. Import data

The required data is imported from the publicly available movielens database. Subsequently, the provided code is used, which automatically creates the data sets *edx* and *validation*. The data set *edx* is used to train and test the different models, while the *validation* data set is used to validate the best model due to the its performance.

2. Preprocess, explore and visualize data

We start with general analyses that are the basis for some preprocessing steps. To gain a deeper understanding of the data, it is subsequently analyzed. For visualization we use histograms, other plots and a correlation analysis.

3. Findings

From the presented graphs and smooth plots we draw conclusions for the usability of the variables as predictors.

4. Modeling approach

We take the steps needed for training and testing, i.e. the reduction of the data set by unneeded variables, the split of the data into train and test set, and the definition of the loss function RMSE. Additionally, we create a data frame to gather and compare the individual model results to the target value of 0.86490.

5. Train and compare different models

In total, we use 19 different approaches to identify the best performing model. These are, e.g. linear models, a generalized linear model as well as a loess model. The different performances are also directly discussed.

6. Results

The best model is selected and applied on the validation set to compare with true ratings. Although, we present and discuss the different model results after each analysis, in this section we give an overall result discussion.

7. Conclusion

Finally, we make a brief summary of the report, its limitations and show potential approaches and ideas for future work.

3 Import data

For importing the data, we first load the data from the publicly available movielens database. In the next steps we clean it and transform it into the two data sets *edx* and *validation*. The objects that are no longer required are deleted in order to have a smaller data set for the further steps.

4 Preprocess, explore, and visualize data

In this section we preprocess the data, explore, and visualize it to gain deeper findings about potential relationships between the different variables.

4.1 Preprocessing

Analyzing the dimensions first, we can see that the edx data set contains 9000055 rows and 6 columns.

```
## [1] 9000055       6
```

We further investigate the summary of the edx data first to check for general plausibility. The data quality looks good at first sight in terms of min, max and mean values. Additionally, the table does not contain NA's which makes it easy to continue in the analysis.

userId	movieId	rating	timestamp	title	genres
Min. : 1	Min. : 1	Min. :0.500	Min. :7.897e+08	Length:9000055	Length:9000055
1st Qu.:18124	1st Qu.: 648	1st Qu.:3.000	1st Qu.:9.468e+08	Class :character	Class :character
Median :35738	Median : 1834	Median :4.000	Median :1.035e+09	Mode :character	Mode :character
Mean :35870	Mean : 4122	Mean :3.512	Mean :1.033e+09	NA	NA
3rd Qu.:53607	3rd Qu.: 3626	3rd Qu.:4.000	3rd Qu.:1.127e+09	NA	NA
Max. :71567	Max. :65133	Max. :5.000	Max. :1.231e+09	NA	NA

If we look at the first rows of the imported data

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

we will detect that:

- besides the actual movie title the variable *title* contains the year when the movie was released. We will extract this information as the new column *movie_year* to be able to use it in further analyses.
- the variable *timestamp* is formatted as integer, representing the seconds since 1970-01-01. The data will be mutated to include the new columns (for data sets edx and validation). Hint: Since integer values use only half of the memory compared to numeric the columns with natural numbers will be converted to integer.
- the variable *genres* contains multiple entries which is hard to investigate in this form. We will add a normalized version of this variable for analyzing the genres in detail later on.

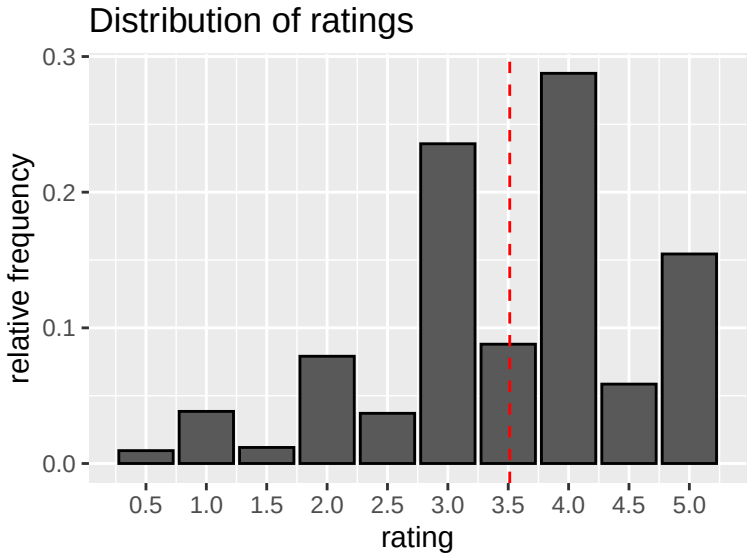
4.2 Data exploration and visualization

In this section, we will perform different analyses that provide an overview of the data. This will be accompanied by supporting data visualizations.

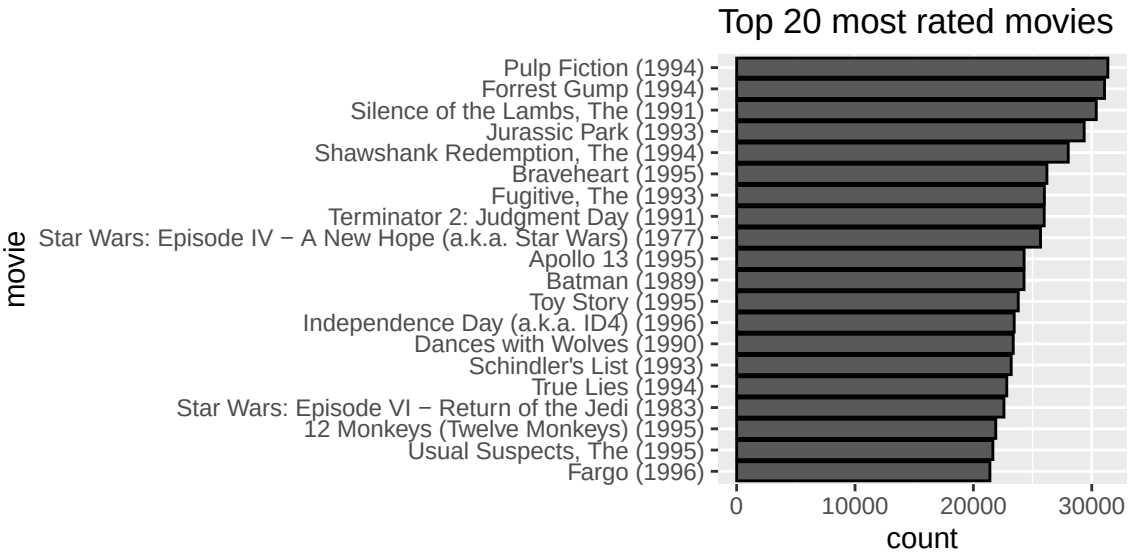
In general, we find out quickly that there are 10677 distinct movies as well as 69878 distinct users.

In terms of the distribution of ratings we see an overall average rating of 3.5124652 which indicates that in the mean the ratings slightly more positive. The standard deviation for the ratings is 1.0603314.

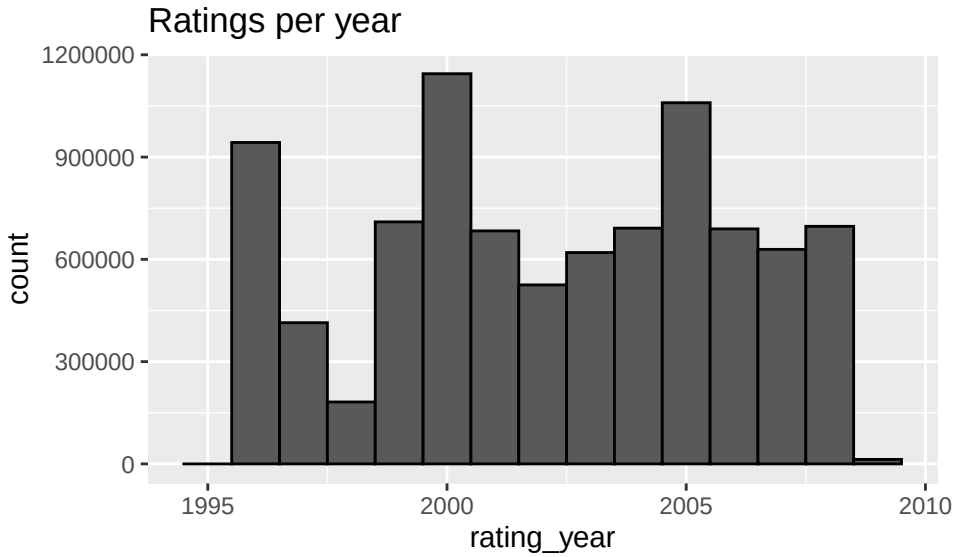
The following histogram *Distribution of ratings* displays the counts of the different ratings. The ratings range from 0.5 to 5 points. A rating of 4 was the most common compared to all other ratings. We can see that in general there are more full ratings compared to the number of half-point ratings. One reason for this could be that it is easier for users to give full ratings or harder to give half-point ratings.



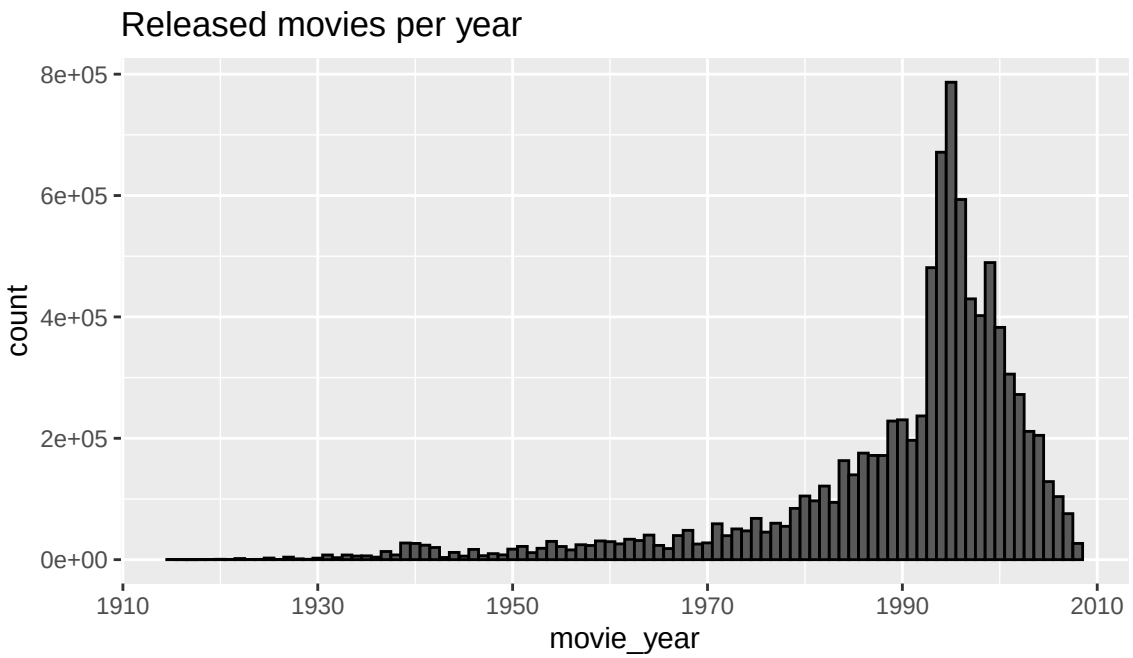
We know that there are blockbusters (which are watched and thus maybe rated with higher probability) and independent movies (which are watch only by a few people). Thus, we could not assume that the films were rated similarly often. In the graph *Top 20 most rated movies* we can see that this is indeed the case. Some movies have been rated quite often (e.g. Pulp Fiction 31362 times). On the other hand, analysis shows that 126 movies were rated only once. We need to consider this for later analyses since they can distort the mean values.



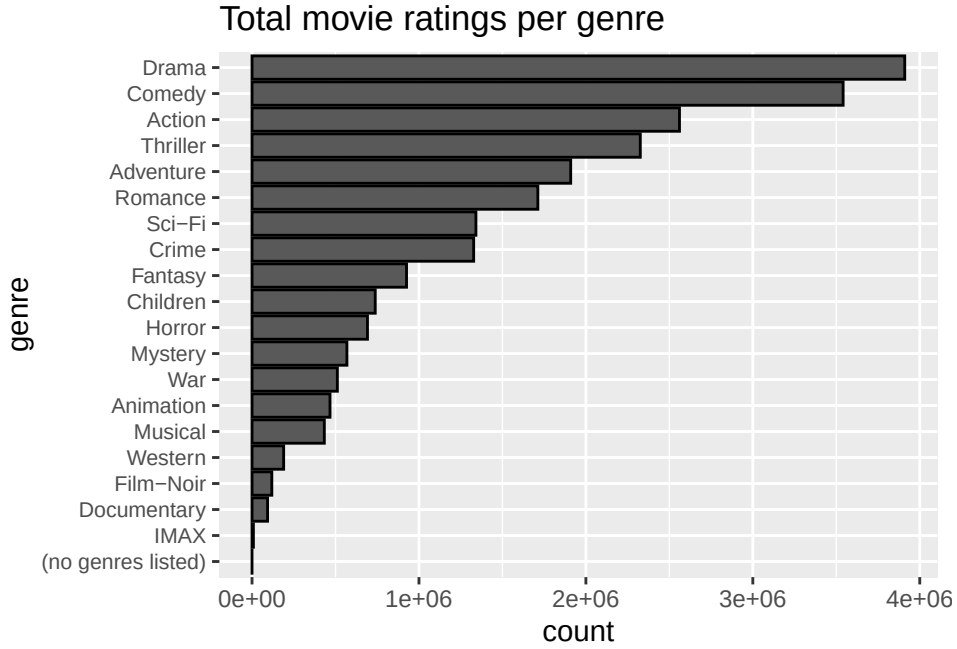
In terms of temporal development the histogram *Ratings per year* shows different things: - the data set contains ratings for the years 1995 to 2009. - users rated movies particularly often in the years 1996, 2000, and 2005. In contrast, there were only very few ratings in the years 1995, 1998, and 2009 compared to other years.



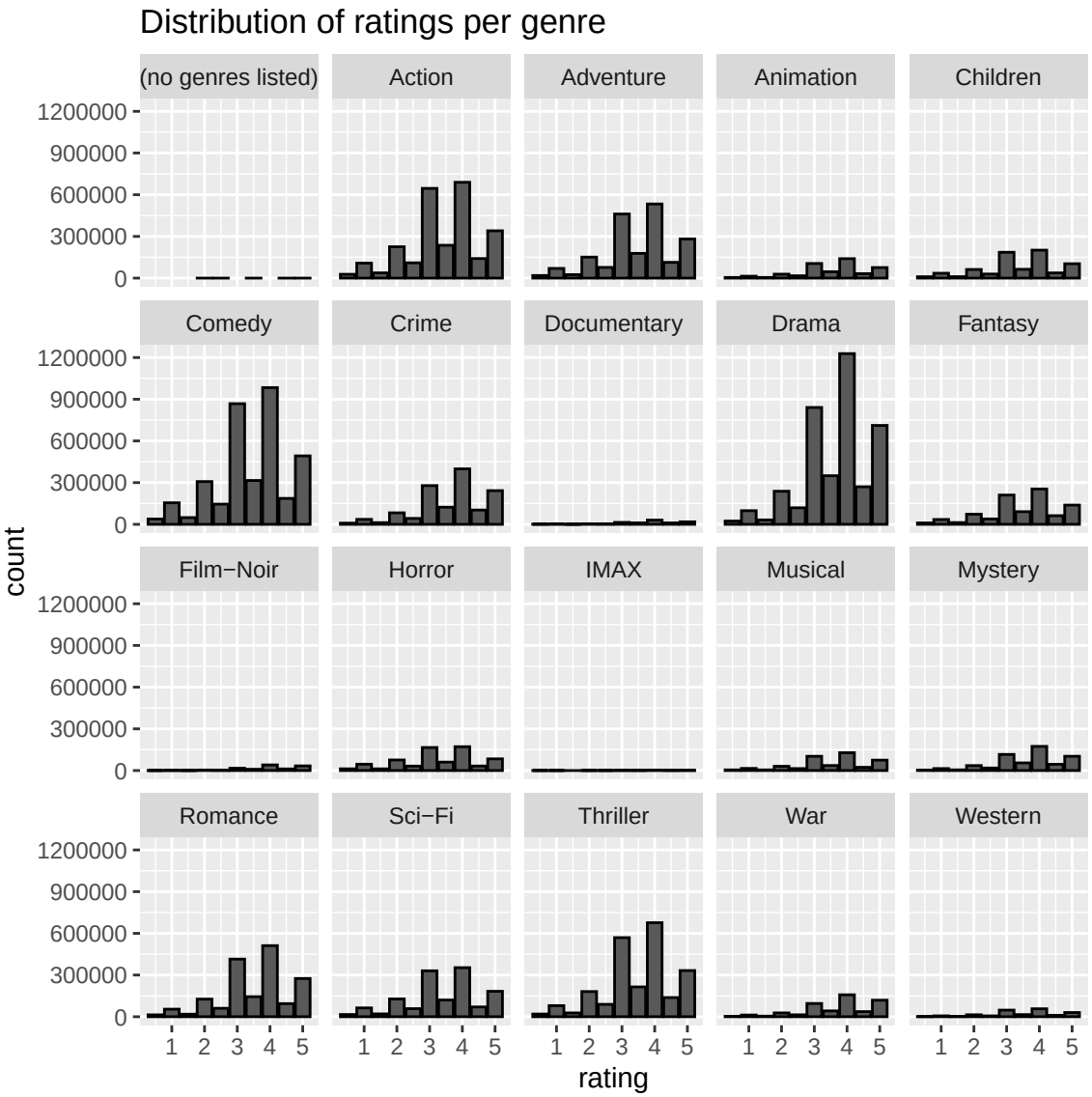
According to the histogram *Released movies per year* we see that around the years 1994 tto 1996 most movies rated have been released.



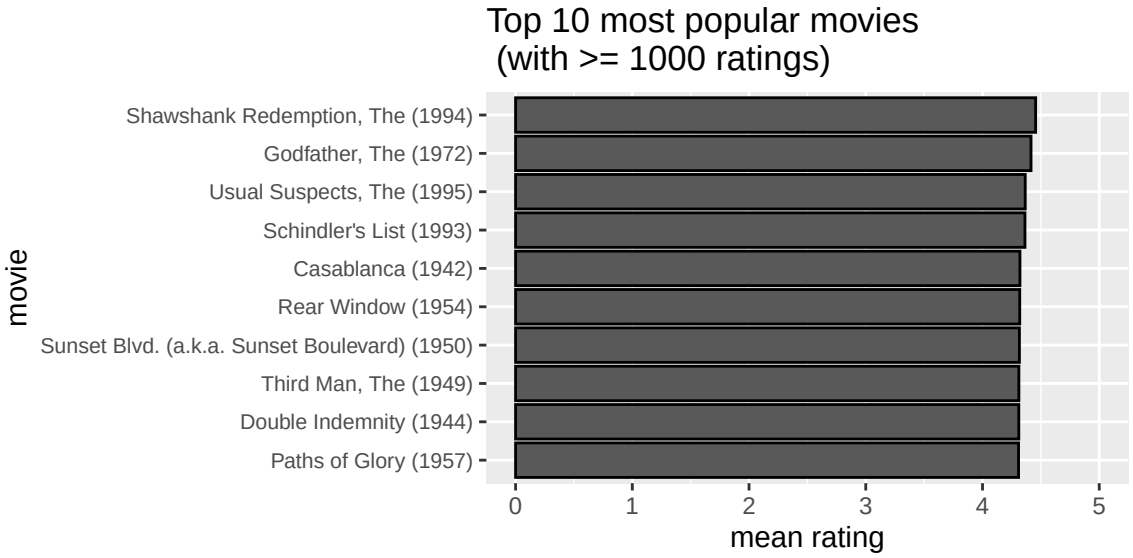
Analyzing the different genres, the histogram *Total movie ratings per genre* reveals that the genres drama, comedy, action, and thriller received the most ratings overall.



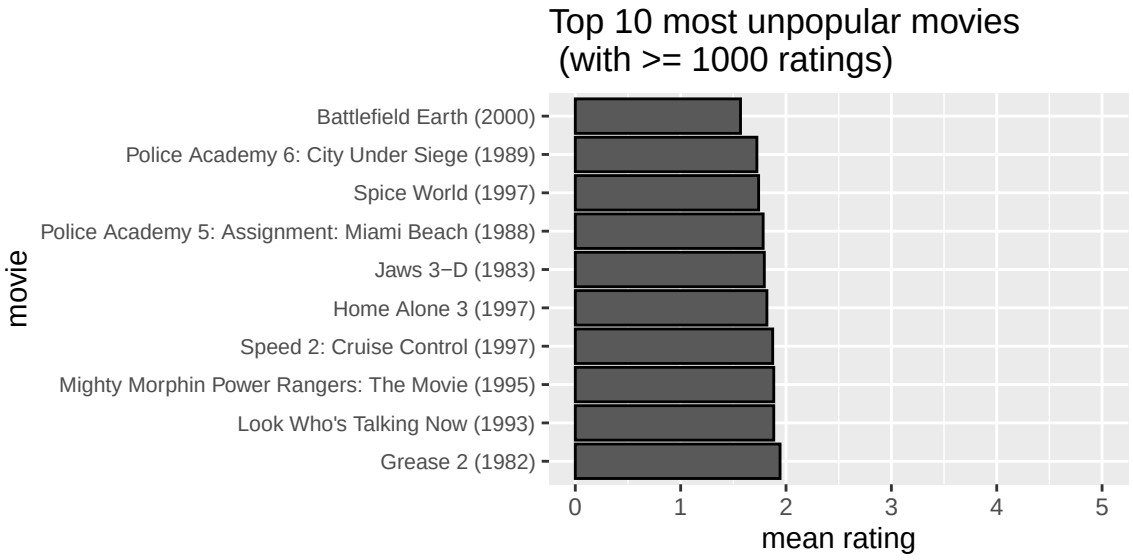
Additionally, the histogram *Distribution of ratings per genre* shows that the rating of 4 is the most common in all these mentioned genres.



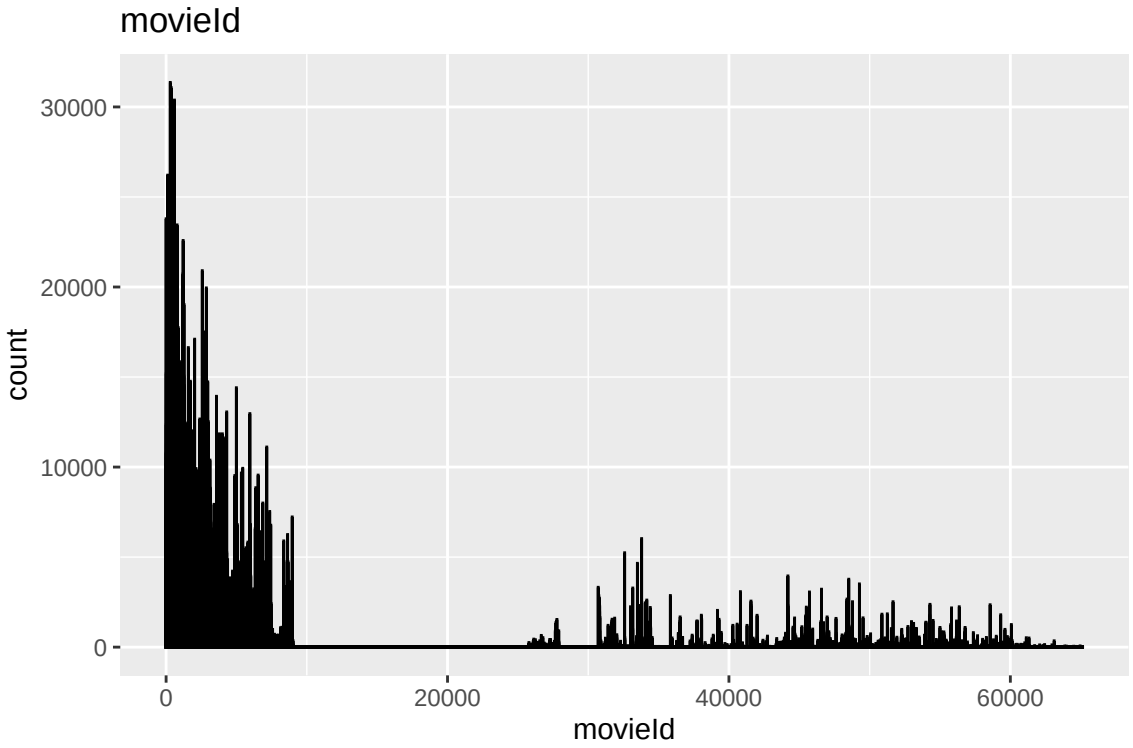
In terms of the top 10 most popular (i.e. best rated) movies, histogram *Top 10 most popular movies (with >= 1000 ratings)* shows the results where movies are only considered if they have at least 1000 ratings.

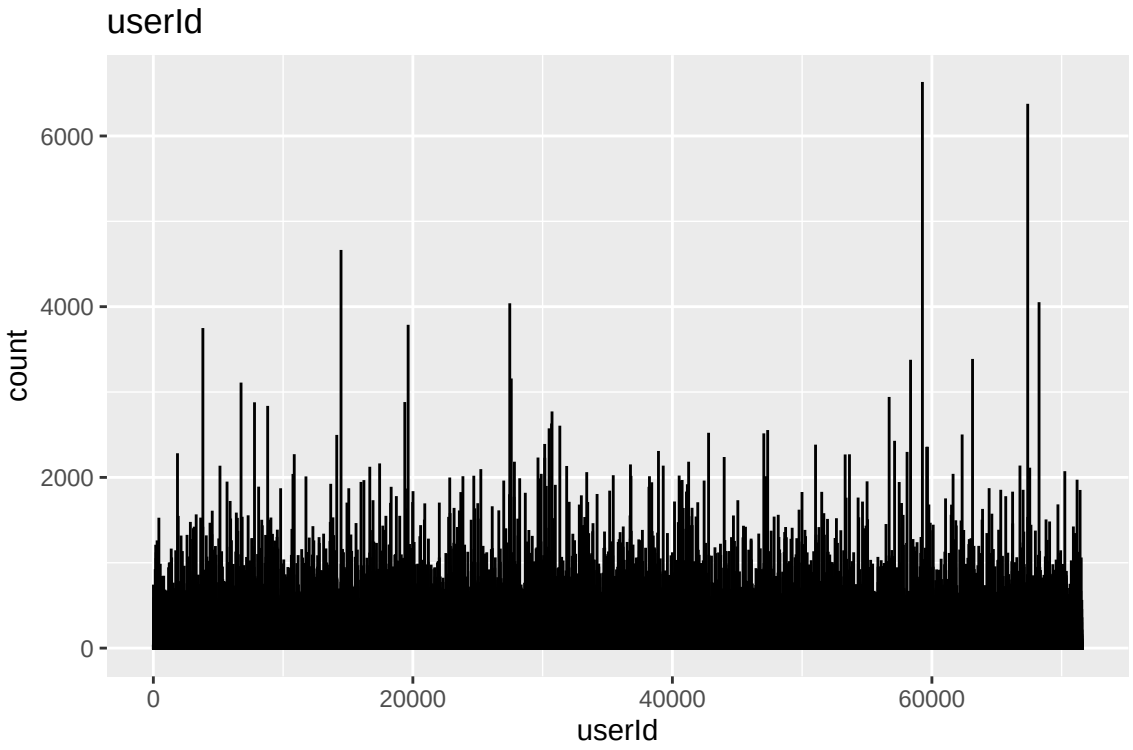


On the other hand, histogram *Top 10 most unpopular movies (worst rated, with at least 1000 ratings)* shows the top 10 most unpopular movies (with at least 1000 ratings).

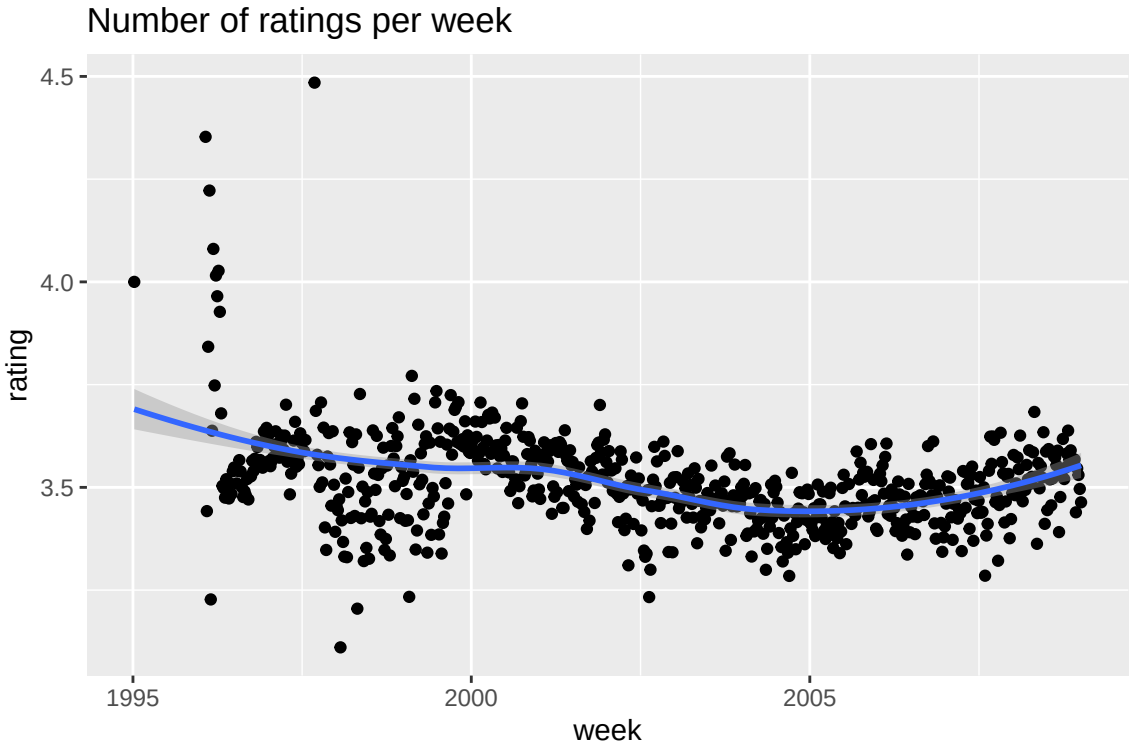


The following two histograms *movieId* and *userId* show that the range of occurrences is quite wide. For our analyses later on, we can observe this as plausible: there are movies and users that have a high number of ratings and, on the other hand, there are movies and users with very few ratings.

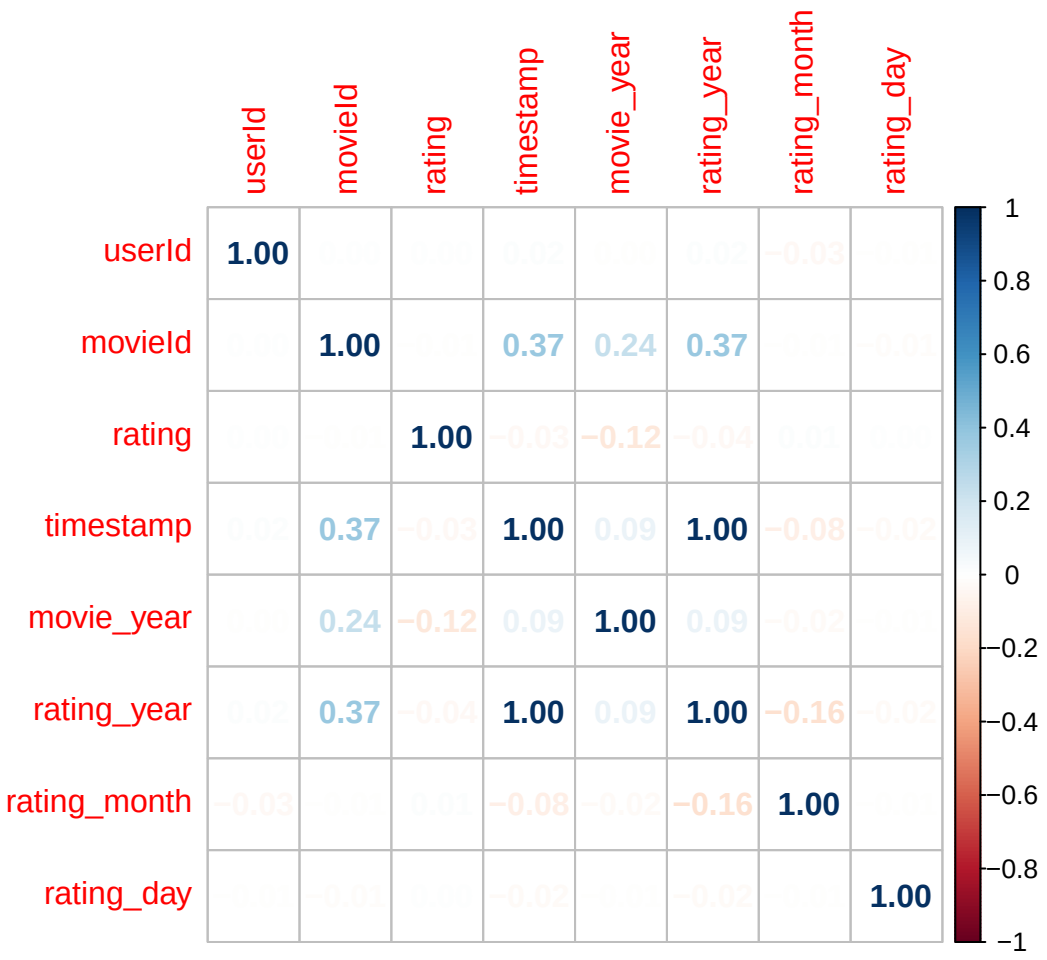




The scatterplot *Number of ratings per week* points out how the ratings develop over the weeks in the analyzed period from 1995 to 2009. The low number of ratings can explain the high outliers between 1995 and 2000 – the means are not that representative compared to the following weeks.



The following correlation analysis helps us to find predictors that are highly correlated. These could be removed (if necessary) for the upcoming section.



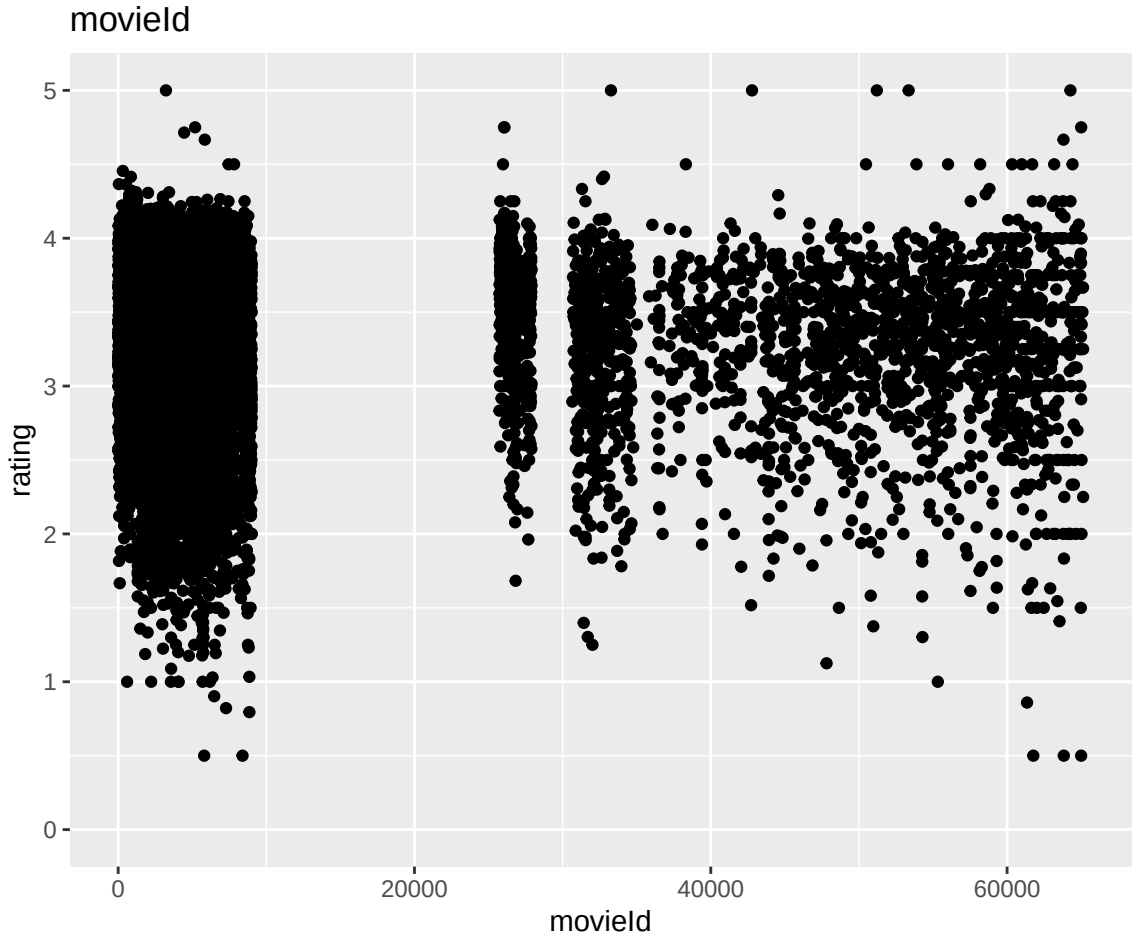
As the correlation matrix shows there are only two perfectly correlated numeric variables: the timestamp and the rating_year variables. This is understandable since rating_year is directly derived from the variable timestamp during the preprocessing step. Thus, we will remove timestamp from both data sets edx and validation to accelerate data processing before training and testing the models. The slight correlation between movieId and timestamp (or rating_year respectively) shows that most ratings tend to happen shortly after the release date of the movie.

5 Findings

The following graphs and smooth plots show the variation of the ratings according to particular variables. They will help to gain deeper insights in terms of the selection of the predictors. In addition, for some variables we also present both mean and standard error to confirm our assumption further.

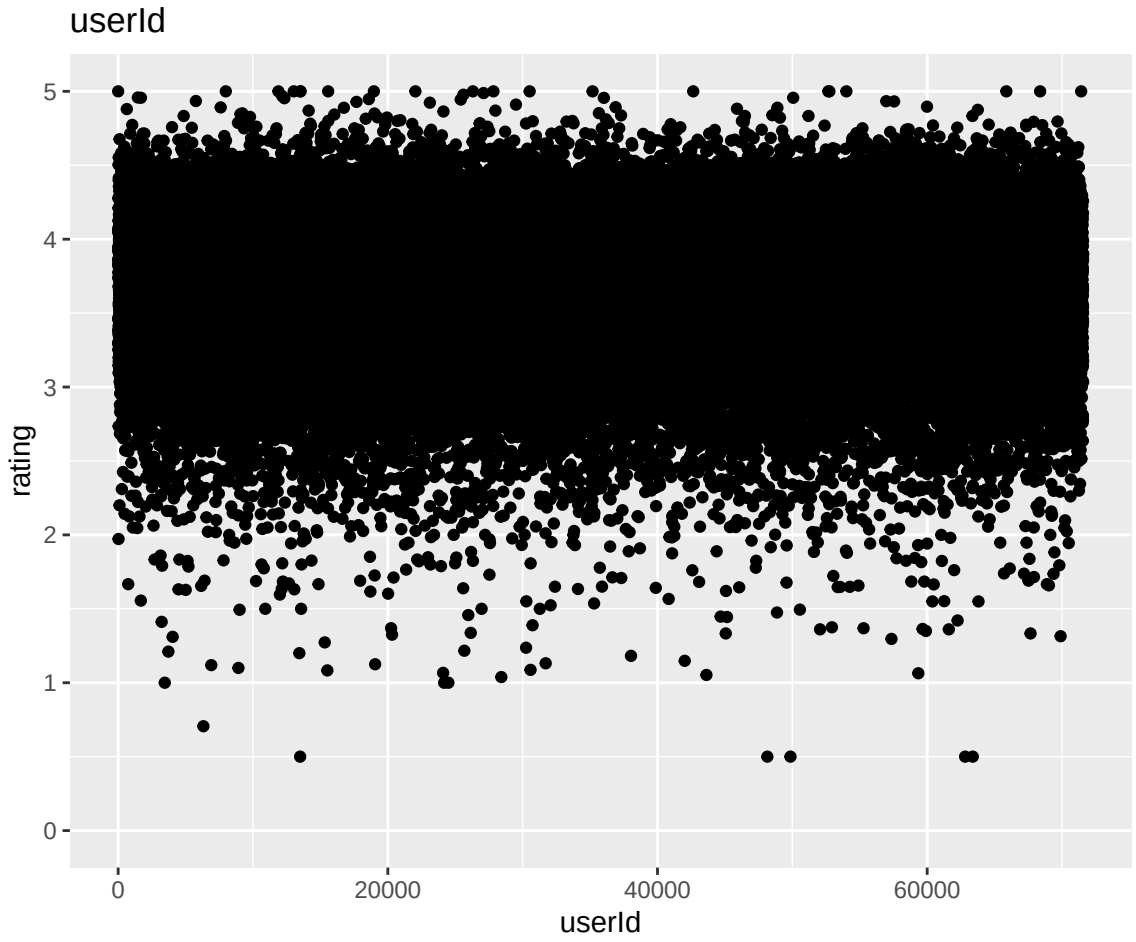
5.1 Variable movieId

The variable movieId shows a particularly high variation in the ratings in graph *movieId*. Here all values from 0.5 to 5 are given. This is a strong indication that it should be used as a prediction due to the high variation.



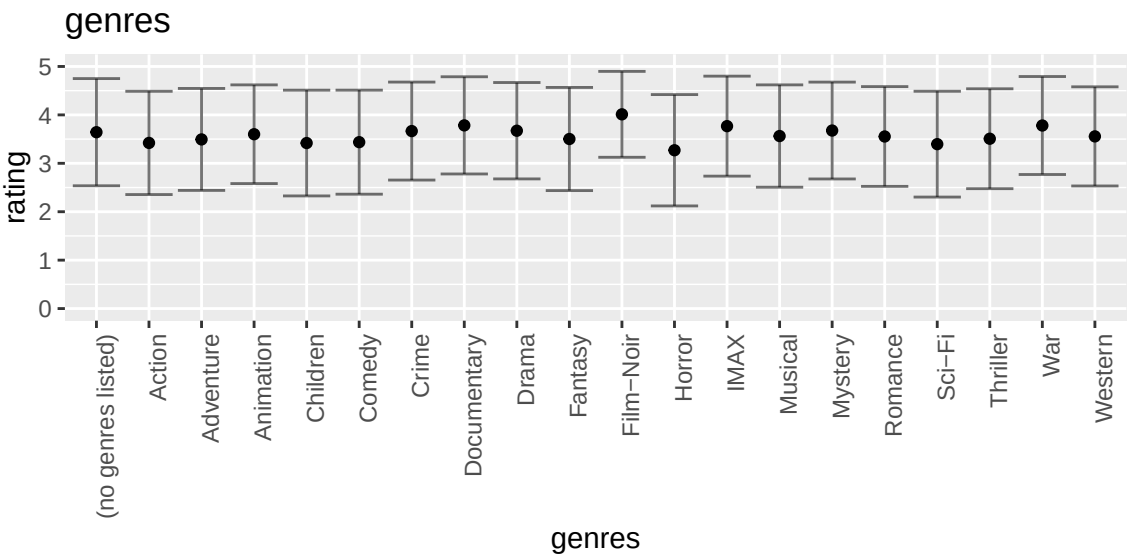
5.2 Variable userId

Although the following graph *userId* looks quite confusing, it shows – similar to the previous representation of movieId – that there is a large variation regarding the ratings for the variable userId as well. Therefore, this variable is also considered as a predictor.



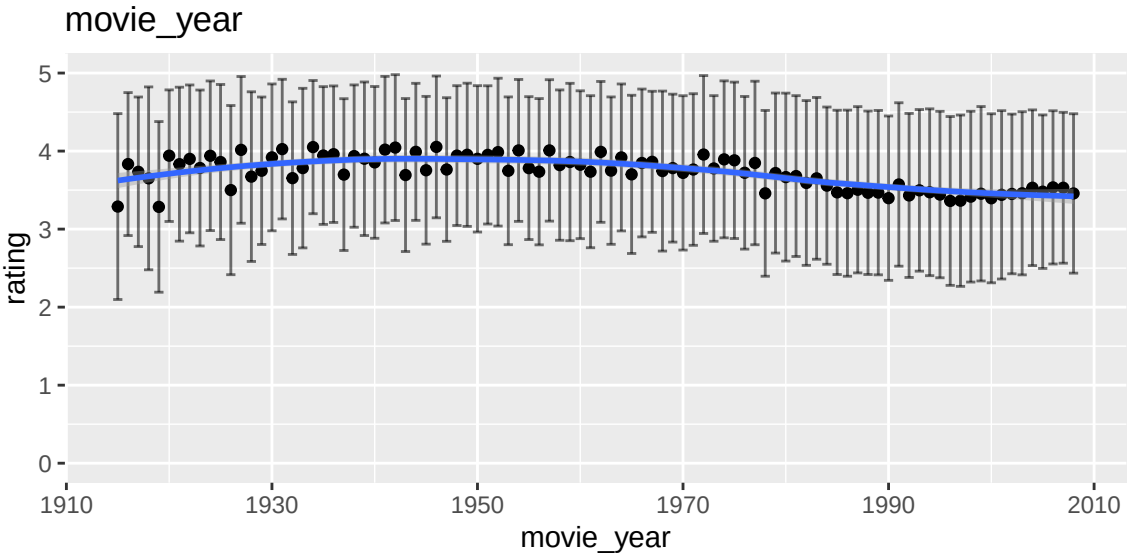
5.3 Variable genres

If we look at the genres (based on the separated genres prepared above), we notice that in the graph *genres* they differ also in the ratings, although this is not as clear as for the previous variables. Nevertheless, this variation makes this variable well suited as a predictor of ratings.



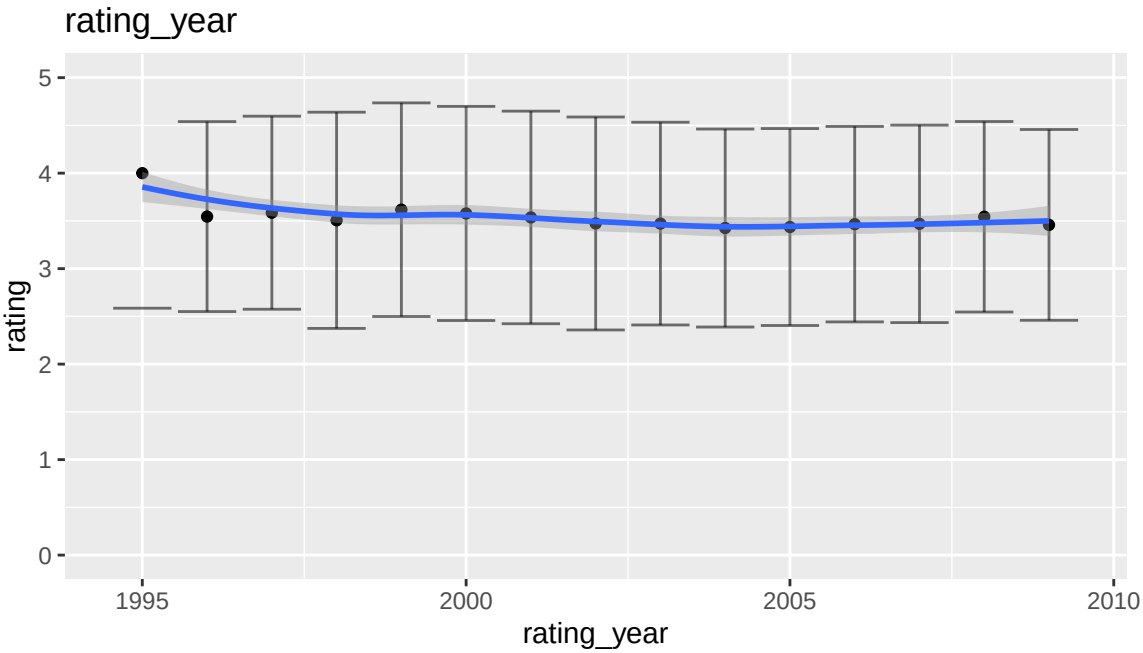
5.4 Variable movie_year

If we look more closely at the variable movie_year we can see that mean and standard error change as well. Furthermore, the smooth plot *movie_year* shows a decline in the mean rating after 1980. According to the findings above, we already know that movies with an early release year (movie year) have less ratings in general. However, these movies have better ratings on average. Thus, we will keep this variable movie_year as predictor.



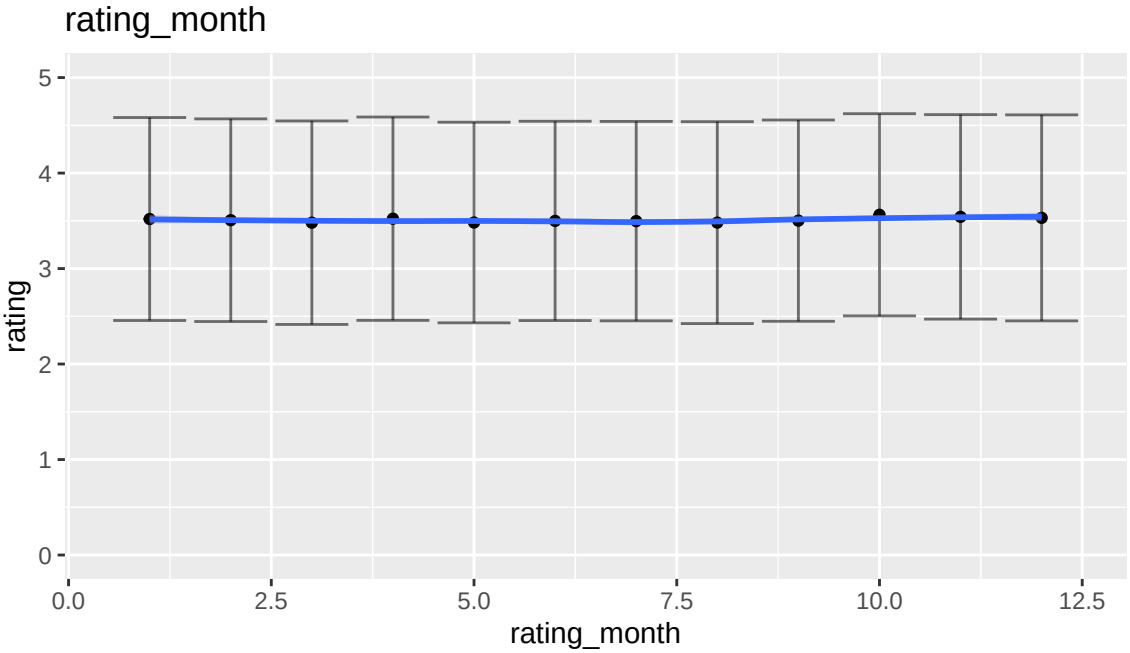
5.5 Variable rating_year

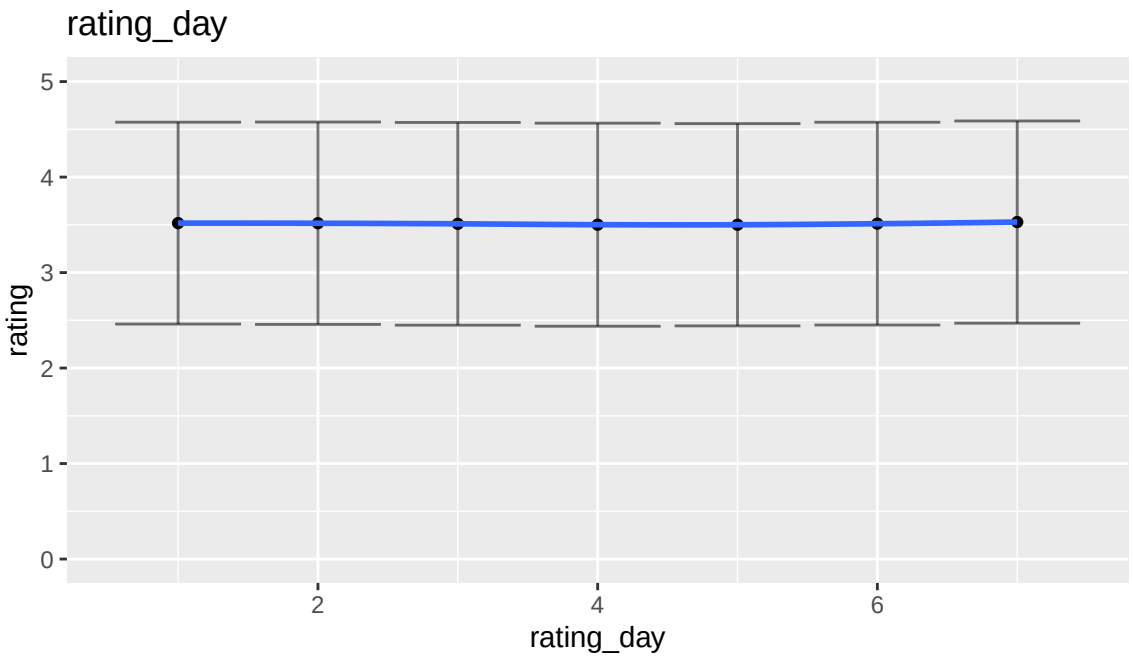
The smooth plot *rating_year* reveals that the ratings slightly change over the different years which could be therefore beneficial to define as predictor.



5.6 Variables rating_month and rating_day

In the last two smooth plots *rating_month* and *rating_day* respectively we can see that ratings on a monthly (left) and daily (right) basis do almost not change.





These variables will not be considered as predictors in the following. Thus, we drop them from the data sets edx and validation in the next section.

6 Modeling approach

After all the preparations and analyses so far, the following section describes the modeling approach. For this purpose, the data set is first reduced by the variables identified in the above section that show almost no variation. Subsequently, the data set is split into train and test set and the loss function RMSE is defined. Finally, we come to the section's core: the training, testing and comparison of the individual models. In order to find the best model we will use the predictors as described previously and apply them individually or in combination.

6.1 Reduce the data set

As mentioned above we attempt to minimize the data set to reduce memory space. According to perfect correlation between timestamp and rating_year, we remove the variables timestamp. Due to the non-existing variation in the two variables rating_month and rating_day, these two variables are also removed.

6.2 Create the train and test set

For the split of the edx data set into train and test set the published project code is used. The train set contains 90% of the data.

6.3 Define the loss function RMSE

As mentioned, for interpreting the performance of the different models we use the loss function root mean square error (RMSE) which is defined as follows:

$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\widehat{y_i} - y_i)^2}{n}}$

which looks like this as a R function:

```
# Define RMSE (i.e. the loss function)
RMSE <- function(true_ratings, predicted_ratings) {
   sqrt(mean((true_ratings - predicted_ratings)^2))}
```

6.4 Create the result data frame and target value

In the following, we create a data frame to gather and compare the individual model results. Next, we will include the target value in the data frame, so that we can directly compare the different models with it.

modelid	method	RMSE	delta_model_target
0	target	0.8649	0

For our further steps, we also need to define the rating mean of the train data set.

```
# Define mean edx_train
mu <- mean(edx_train$rating)
```

7 Train and compare different models

In this section, we will train the different prediction models and compare the outcomes to find the best model for the validation in the next step.

7.1 Overview of the prediction models

The following prediction models are described in the following:

- Model 1 - Average

Linear models with one predictor:

- Model 2 - Linear model with movieId
- Model 3 - Linear model with userId

Linear models with two predictors:

- Model 4 - Linear model with movieId and userId
- Model 5 - Linear model with movieId and rating_year
- Model 6 - Linear model with movieId and movie_year
- Model 7 - Linear model with movieId and genres
- Model 8 - Linear model with userId and rating_year
- Model 9 - Linear model with userId and movie_year
- Model 10 - Linear model with userId and genres

Linear models with three predictors:

- Model 11 - Linear model with movieId, userId, and rating_year
- Model 12 - Linear model with movieId, userId, and movie_year
- Model 13 - Linear model with movieId, userId, and genres

Linear models with four predictors:

- Model 14 - Linear model with movieId, userId, genres, and rating_year
- Model 15 - Linear model with movieId, userId, genres, and movie_year

Linear model with five predictors:

- Model 16 - Linear model with movieId, userId, genres, rating_year, and movie_year

Regularized linear model:

- Model 17 - Regularized linear model with movieId, userId, genres, rating_year, and movie_year

GLM and LOESS models:

- Model 18 - Generalized linear model (GLM)
- Model 19 - gamLOESS

As we can see, several linear models are used below. They apply one to five predictors and calculate the difference between the mean rating of the predictor(s) and the overall mean rating. Then, the outcome is used to calculate the prediction by adding this value to the mean. Finally, the best model is optimized by applying regularization. Furthermore, there are GLM and LOESS models that will be used.

7.2 Model 1 - Average

The simple average is used as the basic prediction model. The RMSE result is well above the target value:

modelid	method	RMSE	delta_model_target
0	target	0.864900	0.0000000
1	average	1.060054	0.1951537

7.3 Model 2 - Linear model with movieId

Since we saw that movieId is one of the most promising predictors we start with this. The RMSE shows that with movieId we get a quite good result compared to model 1: The deviation from the target value is now 0.0780615.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615

7.4 Model 3 - Linear model with userId

If we compare model 2 with the outcome of the linear model with the predictor userId we get the following RMSE:

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091

So far, the results show that model 2 has a slightly lower RMSE compared to model 3. Therefore, movieId is the better predictor compared to userId what we already expected.

7.5 Model 4 - Linear model with movieId and userId

In model 4 we combine both predictors movieId and userId to see if we get a better performance. The results are quite promising: the RMSE for this model are significantly lower than the previous models with isolated predictors - and it is the first model that hits the target value.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156

7.6 Model 5 to 7 - Linear models with movieId and other predictors

To see if there are other two-predictor combinations that show a lower RMSE we use the idea of combining the so far best predictors movieId and userId with the other variables. At first, we combine movieId with rating_year, movie_year and genres respectively.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615

We can see that only the combination of movieId and rating_year reduces the RMSE value slightly compared to use movieId as a single predictor. However, none of the three model beats model 4 - they do not reach the target value.

7.7 Model 8 to 10 - Linear models with userId and other predictors

Next instead of movieId, we use the predictor userId and create the same two-predictor combinations as above.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187

7.8 Model 11 to 13 - Linear models with movieId, userId and other predictors

Since so far the combination of movieId and userId as predictors result still in the best model, we combine those with other variables in models with three predictors: with genres, rating_year, and movie_year respectively.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091

modelid	method	RMSE	delta_model_target
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758

We can see that RMSE results of model 13 - linear model (movieId + userId + genres) - are the best results now. As predictor the variable genres performs better compared to rating_year and movie_year in combination with movieId and userId. The model is beating the target by $-5.7577352 \times 10^{-4}$.

7.9 Model 14 to 15 - Linear models with movieId, userId, genres, and other predictors

To find out if there is a four-predictor combination we go a step further and combine the best model so far (model 13) with rating_year and movie_year respectively.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758
14	linear model (movieId + userId + genres + rating_year)	0.8643142	-0.0005858
15	linear model (movieId + userId + genres + movie_year)	0.8641262	-0.0007738

The RMSE result clearly shows that we have found a new best model: Model 15 - linear model (movieId + userId + genres + movie_year)as predictors has the best value 0.8641262 so far -7.737643×10^{-4} below the target.

7.10 Model 16 - Linear model with movieId, userId, genres, movie_year, and rating_year

Finally, we use all possible predictors that have been in use so far in different combinations - in the form of a model with five predictors.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758
14	linear model (movieId + userId + genres + rating_year)	0.8643142	-0.0005858
15	linear model (movieId + userId + genres + movie_year)	0.8641262	-0.0007738
16	linear model (movieId + userId + genres + movie_year + rating_year)	0.8640467	-0.0008533

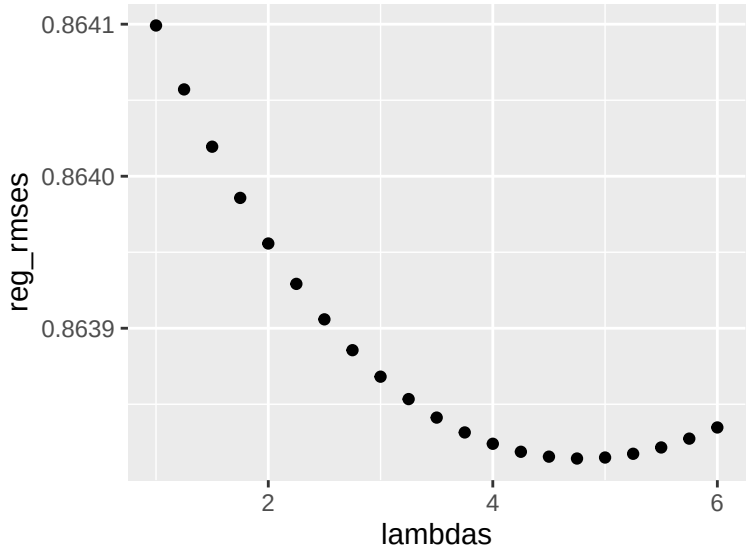
Within the linear models, we now have found the best with model 16. The RMSE result provides the lowest value so far (0.8640467) with a deviation from the target value of $-8.5333119 \times 10^{-4}$.

7.11 Model 17 - Regularized linear model with movieId, userId, genres, rating_year, and movie_year

By regularizing, in terms of the five predictors we will penalize the predictor values that occur very rarely. To do this, we define lambda to result in an overall minimal RMSE by minimizing the following formula:

$\frac{1}{N} \sum_{u,i,g,my,ry} (y_{u,i,g,my,ry} - \mu - b_i - b_u - b_g - b_{my} - b_{ry})^2 + \lambda(\sum_i b_i^2 + \sum_u b_u^2 + \sum_g b_g^2 + \sum_{my} b_{my}^2) + \sum_{ry} b_{ry}^2)$

The following graph shows for which lambda the RMSE becomes minimal.



modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758
14	linear model (movieId + userId + genres + rating_year)	0.8643142	-0.0005858
15	linear model (movieId + userId + genres + movie_year)	0.8641262	-0.0007738
16	linear model (movieId + userId + genres + movie_year + rating_year)	0.8640467	-0.0008533
17	Regularized linear model 16 (movieId + userId + genres + movie_year + rating_year)	0.8638143	-0.0010857

The graph shows gives us the minimum RMSE for a lambda of 4.75. By using this lambda we get a RMSE of 0.8638143 which is -0.0010857 better compared to the target value.

7.12 Model 18 - Generalized linear model (GLM)

The last model we use is the generalized linear model (GLM). We apply that model with the variables from above which a) beats the target and b) has the minimal amount of predictors. In this case, this is model 4 (linear model (movieId + userId). The results show that they are not as good as compared to our regularized model 17.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758
14	linear model (movieId + userId + genres + rating_year)	0.8643142	-0.0005858
15	linear model (movieId + userId + genres + movie_year)	0.8641262	-0.0007738
16	linear model (movieId + userId + genres + movie_year + rating_year)	0.8640467	-0.0008533

modelid	method	RMSE	delta_model_target
17	Regularized linear model 16 (movieId + userId + genres + movie_year + rating_year)	0.8638143	-0.0010857
18	generalized linear model (movieId + userId)	1.0600317	0.1951317

7.13 Model 19 - gamLOESS

We use the local weighted regression (loess) model with the predictor movieId. The results are not significantly better compared to the overall rating average.

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758
14	linear model (movieId + userId + genres + rating_year)	0.8643142	-0.0005858
15	linear model (movieId + userId + genres + movie_year)	0.8641262	-0.0007738
16	linear model (movieId + userId + genres + movie_year + rating_year)	0.8640467	-0.0008533
17	Regularized linear model 16 (movieId + userId + genres + movie_year + rating_year)	0.8638143	-0.0010857
18	generalized linear model (movieId + userId)	1.0600317	0.1951317
19	gamLOESS	1.0586244	0.1937244

8 Results

In summary, we conclude that the best result is achieved by regularized linear model 17, which has the five predictors movieId, userId, genres, movie_year, and raiting_year. Thus, in the next step, this model is validated by the validation set. The regularized model has the lambda value of 4.75.

It is important to check the results of the new predictions for NA's. The summary shows that there are 4 occurrences of NA's.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -0.432   3.138   3.564   3.512   3.942   5.965       4
```

Since these titles are movies that are very rarely watched and rated only once, we use the mean value instead of NA's for the final validation.

title	rating
Rowing with the Wind (Remando al viento) (1988)	3
Young Unknowns, The (2000)	2
Big Fella (1937)	2
Rowing with the Wind (Remando al viento) (1988)	2

modelid	method	RMSE	delta_model_target
0	target	0.8649000	0.0000000
1	average	1.0600537	0.1951537
2	linear model (movieId)	0.9429615	0.0780615
3	linear model (userId)	0.9777091	0.1128091
4	linear model (movieId + userId)	0.8646844	-0.0002156
5	linear model (movieId + rating_year)	0.9408085	0.0759085
6	linear model (movieId + movie_year)	0.9429615	0.0780615
7	linear model (movieId + genres)	0.9429615	0.0780615
8	linear model (userId + rating_year)	0.9776939	0.1127939
9	linear model (userId + movie_year)	0.9688332	0.1039332
10	linear model (userId + genres)	0.9420187	0.0771187
11	linear model (movieId + userId + rating_year)	0.8646812	-0.0002188
12	linear model (movieId + userId + movie_year)	0.8643302	-0.0005698
13	linear model (movieId + userId + genres)	0.8643242	-0.0005758
14	linear model (movieId + userId + genres + rating_year)	0.8643142	-0.0005858
15	linear model (movieId + userId + genres + movie_year)	0.8641262	-0.0007738
16	linear model (movieId + userId + genres + movie_year + rating_year)	0.8640467	-0.0008533
17	Regularized linear model 16 (movieId + userId + genres + movie_year + rating_year)	0.8638143	-0.0010857
18	generalized linear model (movieId + userId)	1.0600317	0.1951317
19	gamLOESS	1.0586244	0.1937244
20	validation (model 17, regularized - movieId + userId + genres + movie_year + rating_year	0.8648559	-0.0000441

The result of the validation is in line with expectations: During the data analysis, we saw that the ratings vary widely. The predictor movieId has turned out to be the one with the greatest influence, followed by userId. Even if the extension by the other variables has only resulted in minor improvements, in the end a regularized linear model with five predictors turns out to be the best. The generalized linear model performs

worse because it assumes a linear relationship for all predictors. The local weighted regression (loess) model assumes that within very small windows the data are linear or parabolic. The different weighting of the data points within a span turns out differently. Looking at the two predictors movieId and userId, we see that these categorical variables are very independent of each other, even when they are right next to each other.

9 Conclusion

In summary, the following results of the project ca n be pointed out: The given target value of 0.86490 could be undercut in several cases with different models. The best model trained was model 17 - Regularized linear model 16 (movieId + userId + genres + movie_year + rating_year). This model reached a RMSE value of 0.8648559 in the validation.

The limitations of this report were the following: First, the evaluations were performed on a limited number of predictors. Second, only a fraction of possible models were elaborated, this also applies to the different combinations of predictors.

For future work, several considerations could apply: On the one hand, the training data could be enlarged. In addition, more models could be used in combination with more predictors to obtain better results. For example, neural networks could be used as further models to enable deep learning algorithms. Further predictors that could help to improve the results would be, for example, actors, film length, production costs, language, but also country-specific characteristics - altogether as a broader database.

Finally, we give a quick summary of the start, end, and elapsed time (in minutes) when running the different sections.

event	start	end	timeElapsed_min
Import data	2021-11-22 19:18:57	2021-11-22 19:19:06	0.15
Preprocess, explore, visualize data	2021-11-22 19:19:06	2021-11-22 19:23:58	4.87
Findings	2021-11-22 19:23:58	2021-11-22 19:24:11	0.22
Modeling approach	2021-11-22 19:24:11	2021-11-22 19:24:14	0.05
Train and compare different models	2021-11-22 19:24:14	2021-11-22 20:09:47	45.55
Results	2021-11-22 20:09:47	2021-11-22 20:09:59	0.20
Conclusion	2021-11-22 20:09:59	2021-11-22 20:09:59	0.00