Movie rating prediction - Report

Mike Miemczok

18 4 2021

Introduction

The Dataset used in this analysis is the MovieLens 10M dataset provided by grouplens.org. It contains two different datasets - one (movies.dat) containing the userID, the movieID, rating, timestamp and the other one (ratings.dat) containing the movieID, the title of the movie and the genres. The movieID column on the ratings data frame is then joined with the movieID column of the movie data frame with the corresponding information.

Afterwards a test and training set is created using the createDataPartition method dividing the data randomly in a 90 - 10 split. The training set named edx contains now 90% of the data. The remaining 10% are stored in validation, the test set which will be used to validate the training method later on.

The structure of the training and test set is as followed:

```
## Classes 'data.table' and 'data.frame':   9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838984885 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Ac
##  - attr(*, ".internal.selfref")=<externalptr>
```

The raw dataset contains ratings by different users for many movies in different genres over a large timespan:

```
##    userId movieId rating timestamp                         title
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
## 3:      1     292      5 838983421               Outbreak (1995)
## 4:      1     316      5 838983392               Stargate (1994)
## 5:      1     329      5 838983392 Star Trek: Generations (1994)
## 6:      1     355      5 838984474       Flintstones, The (1994)
##                           genres
## 1:                Comedy|Romance
## 2:          Action|Crime|Thriller
## 3:  Action|Drama|Sci-Fi|Thriller
## 4:       Action|Adventure|Sci-Fi
## 5: Action|Adventure|Drama|Sci-Fi
## 6:       Children|Comedy|Fantasy
```

The goal of the project is to predict ratings, given by the information of the MovieLens dataset.

Following keysteps were made:

1. Exploring the dataset.
2. Preprocessing the dataset.
3. Visualising the data.
4. Choosing linear regression as the main model.
5. Train the model and choose the right predictors using correlation.
6. Choose the right model to predict the ratings.
7. Present the results.

Analysis

Data preprocessing

The raw dataset has shown that the timestamp is not really readable.

As first step in the data preprocessing it is necessary to transform the timestamp in to a readable date for both datasets (timeformat). Afterwards this information is stripped down to month and year in another column named monthyear.

```
##    timestamp          timeformat
## 1: 838985046 1996-08-02 11:24:06
## 2: 838983525 1996-08-02 10:58:45
## 3: 838983421 1996-08-02 10:57:01
```

```
##    timestamp          timeformat monthyear
## 1: 838985046 1996-08-02 11:24:06    8-1996
## 2: 838983525 1996-08-02 10:58:45    8-1996
## 3: 838983421 1996-08-02 10:57:01    8-1996
```

The genre column is not atomic. That means that a movie can have more genres. It is usefull to seperate the genre in more records.

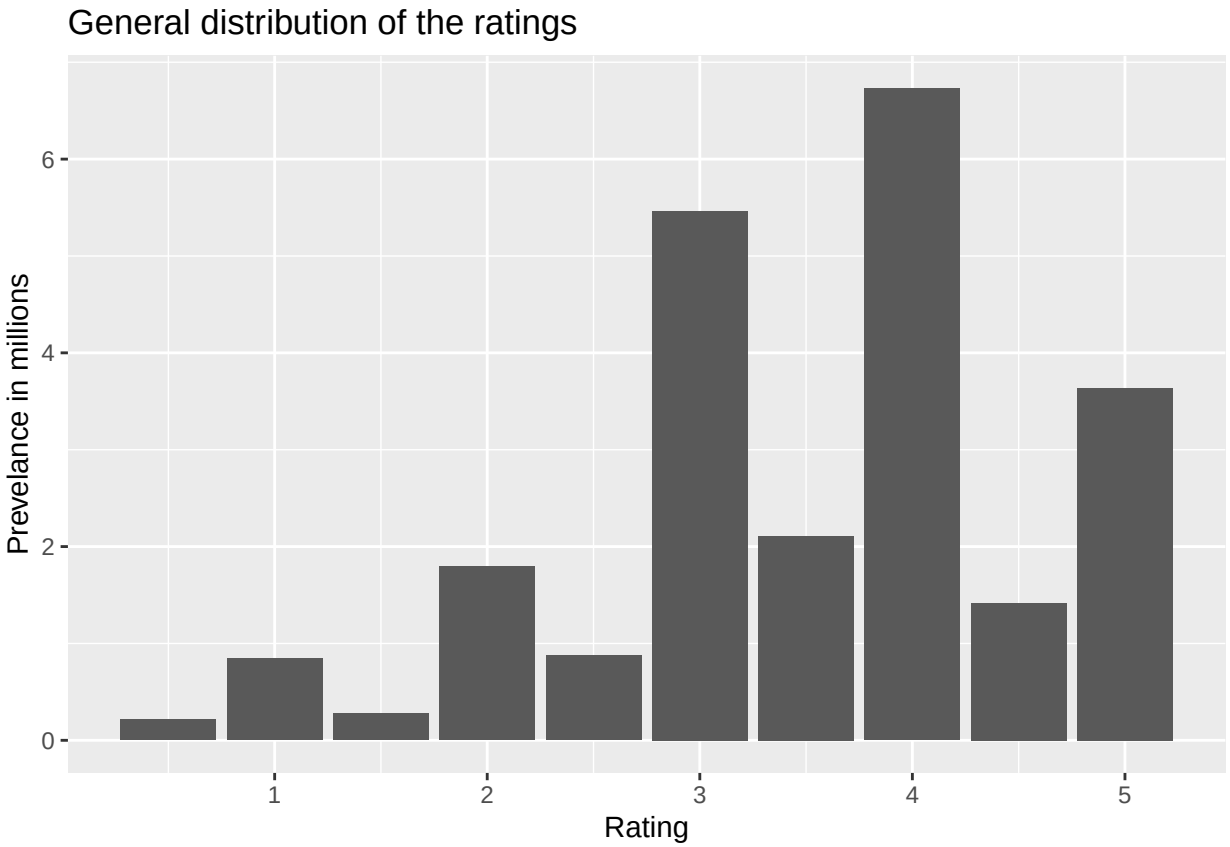
```
##               title         genres
## 1: Boomerang (1992) Comedy|Romance
```

After the seperation the genre looks like:

```
## # A tibble: 2 x 2
##   title            genres
##   <chr>            <chr>
## 1 Boomerang (1992) Comedy
## 2 Boomerang (1992) Romance
```

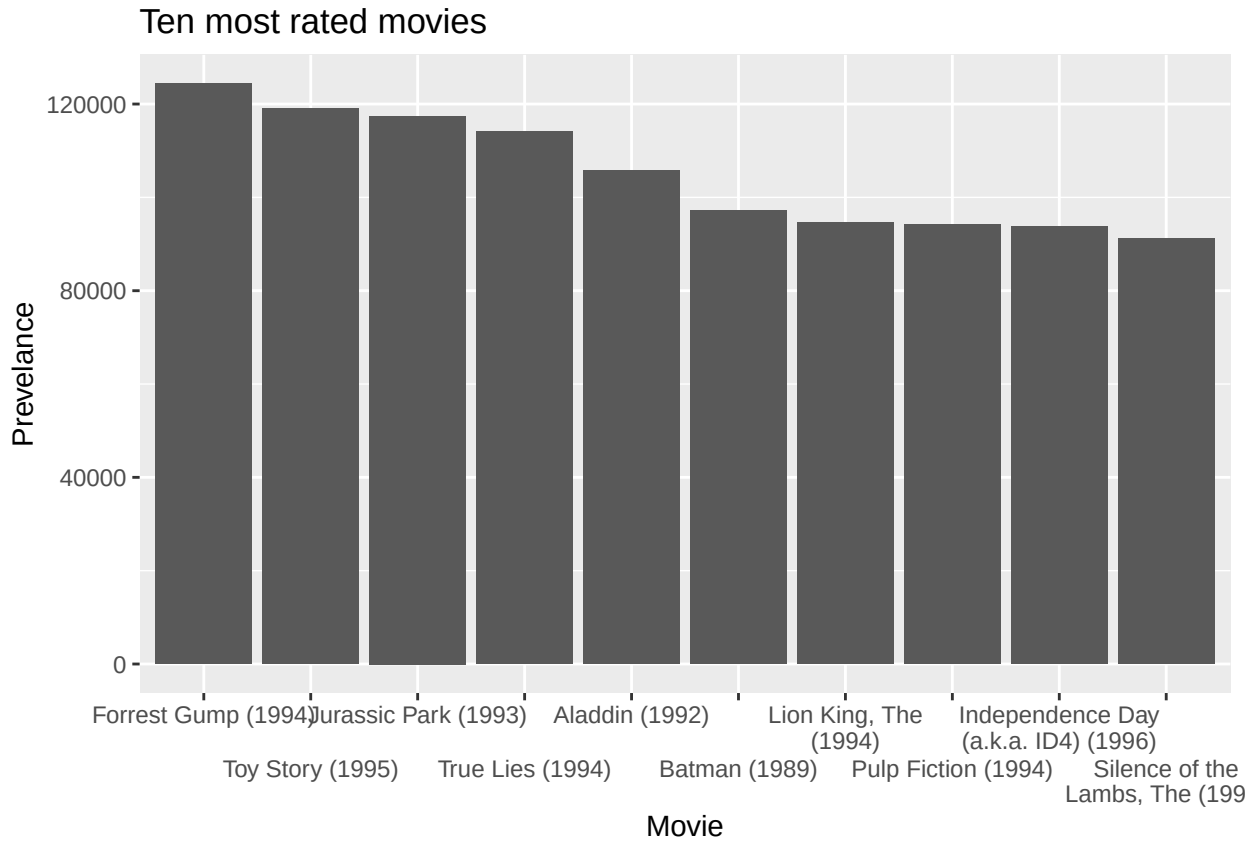
Visualization

To gain a better insight into the dataset, the first step was to look a little closer at the information given in the dataset.



As we can see from the plot shown above the most given rating is 4, followed by 3 and 5. Half star ratings seem to be less likely than whole star ratings. The mean rating seems to be above the mean of the scale 2.5 around 3.5.

The most rated movies are:



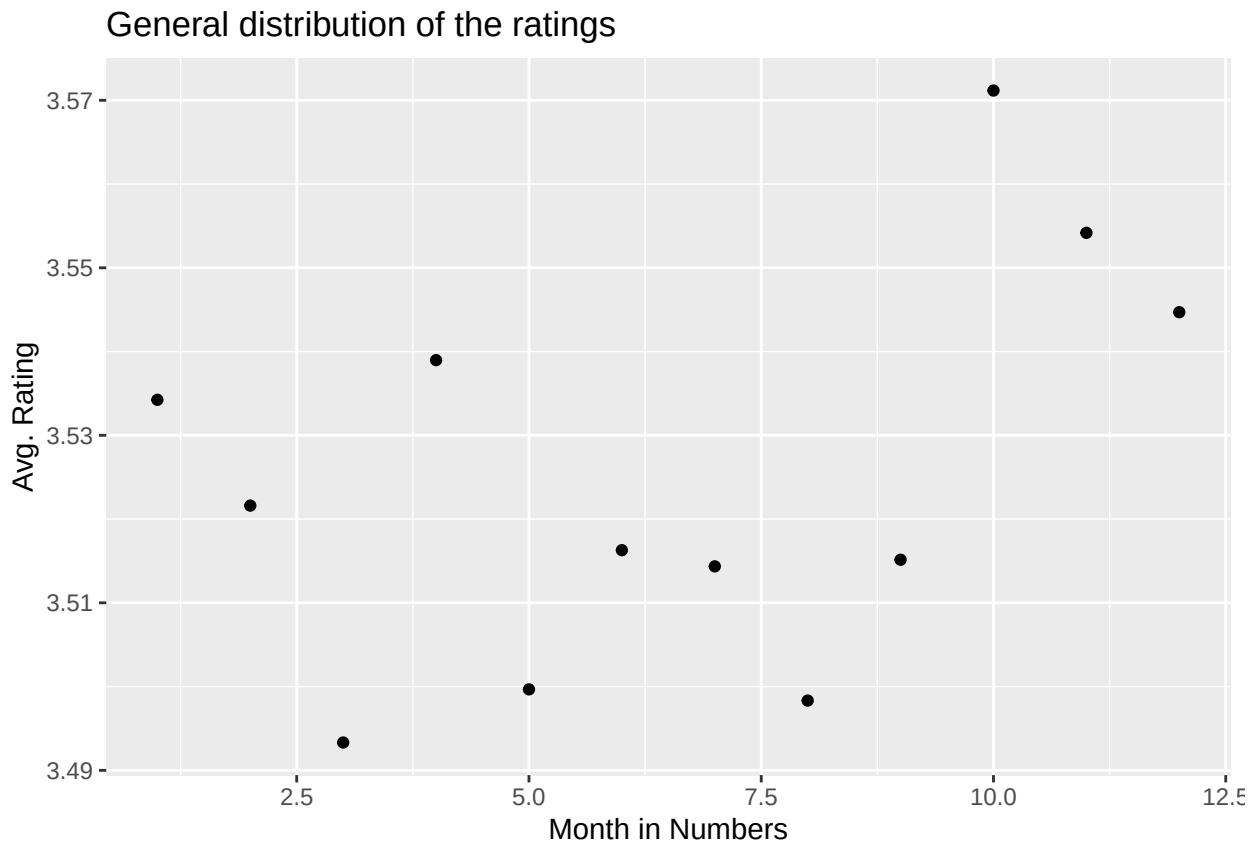
As we can see the most rated movies are all from the 90s. This is most likely due to the fact that movielens ratings start from 1995, so movies from the 90s had a lot more time to accumulate ratings than recent movies.

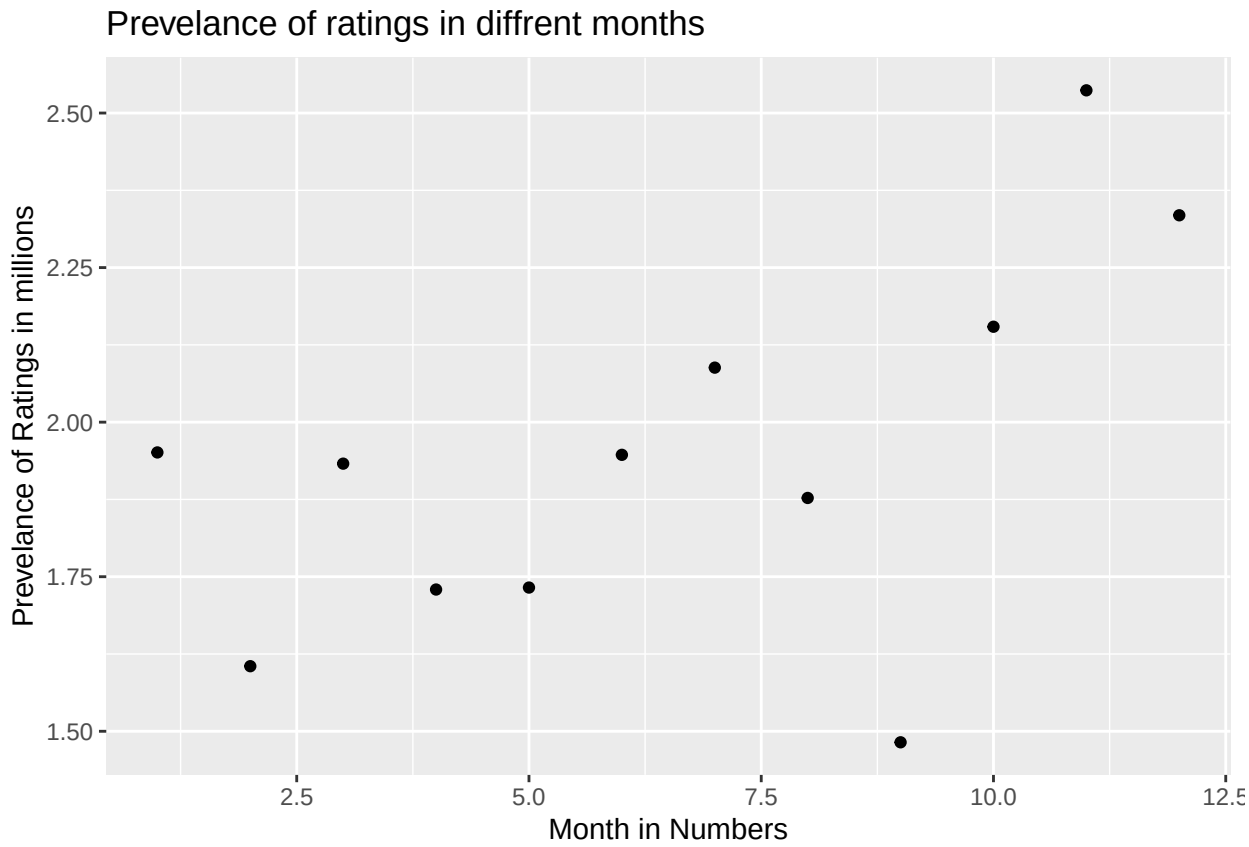
The less rated movies with only one rating each are:

```
## # A tibble: 10 x 1
##    title
##    <chr>
##  1 1, 2, 3, Sun (Un, deuz, trois, soleil) (1993)
##  2 4 (2005)
##  3 Accused (Anklaget) (2005)
##  4 Adios, Sabata (Indio Black, sai che ti dico: Sei un gran figlio di...) (1971)
##  5 Africa addio (1966)
##  6 Bellissima (1951)
##  7 Blind Shaft (Mang jing) (2003)
##  8 Brothers of the Head (2005)
##  9 Condo Painting (2000)
## 10 Confessions of a Superhero (2007)
```

There are over ten thousand movies with only one rating. Due to this occurrence these film tend to be rated more extreme - more positive or negative - compared to movies with thousands of ratings. This effect will be taken into account during regularization with the usage of lambda.

With the next two visualizations the goal is to show if there is a correlation between the rating and the time of the rating.





We will look at the correlation in detail later, but we can already see there is most likely very little to no correlation between the time of the rating and the rating itself.

Building the model

Model: Mean(Rating)

Next up the edx set is divided into a training set and a test set using a 80 - 20 split (80% train set, 20% test set). To test the model later without problems all the entries in the test set, that are not in the training set, are removed and added to the training set.

The naive approach would be to guess the rating based on the mean rating over all movies. The following formula represents this approach, where $Y_{i,u,g,my}$ represents the predicted rating, μ the mean rating and $\epsilon_{i,u,g,my}$ the independent errors:

$$Y_{i,u,g,my} = \mu + \epsilon_{i,u,g,my}$$

The mean rating for all movies for the train set is:

```
## [1] 3.527021
```

Based on this model we can check with the test set how well the model is currently performing.

method	RMSE
Mean(Rating)	1.052367

Using the naive model using only the mean(rating) the obtained RMSE using the test set is: 1.052367

Model: Mean(Rating) + Movie

To gain an insight on the relationship between the predictors and the final rating we first look at the correlation between each predictor and the final rating.

Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102

As we can see in the table movie has the strongest correlation and therefore impact on the rating. The second strongest predictor is User followed by genre and monthyear. Therefore we choose the movie predictor as the first variable used in the linear regression model.

The approach is to guess the rating based on the mean rating over all movies and the movie effect. The movie effect is the influence the movie has on the rating, obtained by the mean rating of this movie in relation to the mean(rating). The following formula represents this approach, where $Y_{i,u,g,my}$ represents the predicted rating, μ the mean rating, b_i the movie effect and $\epsilon_{i,u,g,my}$ the independent errors:

$$Y_{i,u,g,my} = \mu + b_i + \epsilon_{i,u,g,my}$$

Here is an example of 5 values that show the effect the movie has on the rating and the mean influence the movie effect has on the final rating.

```
## # A tibble: 5 x 2
##   movieId    b_i
##     <dbl>  <dbl>
## 1       1  0.398
## 2       2 -0.321
## 3       3 -0.373
## 4       4 -0.650
## 5       5 -0.455
```

```
## [1] 0.6699785
```

Based on this model we can check with the test set how well the adjusted model is currently performing.

method	RMSE	Predictorweight
Mean(Rating)	1.0523666	NA
Movie	0.9411548	0.6699785

Using the adjusted model using only the mean(rating) and the movie effect the obtained RMSE using the test set is: 0.9411548

Model: Mean(Rating) + Movie + User

Based on the independent correlation between user and rating we choose user as the next variable for the regression model. But first we need to make sure movie and user do not correlate much, as user would not improve our model much then.

Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102
Movie ~ User (Independent)	0.1315807

As we can see in the table movie and user do not correlate much, so we include user as the next variable in our model.

The approach is to guess the rating based on the mean rating over all movies and the movie effect + the user effect. The user effect is the influence the user has on the rating, obtained by the mean rating of this user. The following formula represents this approach, where $Y_{i,u,g,my}$ represents the predicted rating, μ the mean rating, b_i the movie effect, b_u the user effect and $\epsilon_{i,u,g,my}$ the independent errors:

$$Y_{i,u,g,my} = \mu + b_i + b_u + \epsilon_{i,u,g,my}$$

Here is an example of 5 values that show the effect the user has on the rating and the mean influence the user effect has on the final rating.

```
## # A tibble: 5 x 2
##   userId     b_u
##    <int>   <dbl>
## 1      1  1.61
## 2      2 -0.326
## 3      3  0.314
## 4      4  0.824
## 5      5 -0.0415
```

```
## [1] 0.4223855
```

Based on this model we can check with the test set how well the adjusted model is currently performing.

method	RMSE	Predictorweight
Mean(Rating)	1.0523666	NA
Movie	0.9411548	0.6699785
Movie + User	0.8578850	0.4223855

Using the adjusted model using only the mean(rating) and the movie effect + user effect the obtained RMSE using the test set is: 0.8578850

Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102
Movie ~ User (Independent)	0.1315807
Rating ~ User (Movie)	0.3596069

As we can see in the correlation table the adjusted user variable (dependent on movie) still correlates with the rating even after the information of the movie is subtracted from it. Therefore there is still a high information gain from the user variable and we can use it to predict the rating more accurately.

Model: Mean (Rating) + Movie + User + Genre

Based on the independent correlation between genre and rating we choose genre as the next variable for the regression model. But first we need to make sure movie and genre or user and genre do not correlate much, as genre would not improve our model much then.

Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102
Movie ~ User (Independent)	0.1315807
Rating ~ User (Movie)	0.3596069
Movie ~ Genre (Independent)	0.2529662
User ~ Genre (Independent)	0.0276370

As we can see in the table user and genre do not correlate much, but movie and genre do. This makes sense as a movie passively already includes the information of the genres it belongs to. We still include the genre variable in our model to see if there would be an improvement, but expect to see not much improvement due to the correlation with movie.

The approach is to guess the rating based on the mean rating over all movies and the movie effect + the user effect + the genre effect. The genre effect is the influence the genre has on the rating, obtained by the mean rating of this genre. The following formula represents this approach, where $Y_{i,u,g,my}$ represents the predicted rating, μ the mean rating, b_i the movie effect, b_u the user effect, b_g the genre effect and $\epsilon_{i,u,g,my}$ the independent errors:

$$Y_{i,u,g,my} = \mu + b_i + b_u + b_g + \epsilon_{i,u,g,my}$$

Here is an example of 5 values that show the effect the genre has on the rating and the mean influence the genre effect has on the final rating.

```
## # A tibble: 5 x 2
##   genres                b_g
##   <chr>               <dbl>
## 1 (no genres listed)  0.336
## 2 Action             -0.0121
## 3 Adventure          -0.0145
## 4 Animation          -0.0146
## 5 Children           -0.0238
```

```
## [1] 0.07811307
```

Based on this model we can check with the test set how well the adjusted model is currently performing.

method	RMSE	Predictorweight
Mean(Rating)	1.0523666	NA
Movie	0.9411548	0.6699785
Movie + User	0.8578850	0.4223855
Movie + User + Genres	0.8578009	0.0781131

Using the adjusted model using only the mean(rating) and the movie effect + user effect + genre effect the obtained RMSE using the test set is: 0.8578009

Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102
Movie ~ User (Independent)	0.1315807
Rating ~ User (Movie)	0.3596069
Movie ~ Genre (Independent)	0.2529662
User ~ Genre (Independent)	0.0276370
Rating ~ Genre (Movie + User)	0.0746336

As we can see in the correlation table the adjusted genre variable (dependent on movie and user) does not correlate with the rating even after the information of the movie and user is subtracted from it. Therefore genre does not provide enough new information to our model to make it worth including. We do not continue to use the genre predictor in our prediction model.

Model: Mean (Rating) + Movie + User + Monthyear

Based on the independent correlation between monthyear and rating we choose monthyear as the next variable for the regression model. But first we need to make sure movie and monthyear or user and monthyear do not correlate much, as monthyear would not improve our model much then. Also monthyear has a low correlation with rating to begin with, so it might not be worth including it in the first place.

Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102
Movie ~ User (Independent)	0.1315807
Rating ~ User (Movie)	0.3596069
Movie ~ Genre (Independent)	0.2529662
User ~ Genre (Independent)	0.0276370
Rating ~ Genre (Movie + User)	0.0746336
Movie ~ Monthyear (Independent)	0.0185652
User ~ Monthyear (Independent)	0.1594020

As we can see in the table movie and monthyear do not correlate much and user and monthyear only correlate a little bit. We include the monthyear variable in our model to see if there would be an improvement, but expect to see not much improvement due to the low correlation with the rating to begin with.

The approach is to guess the rating based on the mean rating over all movies and the movie effect + the user effect + the monthyear effect. The monthyear effect is the influence the timing - meaning the month and year of the rating - has on the rating, obtained by the mean rating for a specific month and year. The following formula represents this approach, where $Y_{i,u,g,my}$ represents the predicted rating, μ the mean rating, b_i the movie effect, b_u the user effect, b_{my} the monthyear effect and $\epsilon_{i,u,g,my}$ the independent errors:

$$Y_{i,u,g,my} = \mu + b_i + b_u + b_{my} + \epsilon_{i,u,g,my}$$

Here is an example of 5 values that show the effect the timing - month and year - has on the rating and the mean influence the monthyear effect has on the final rating.

```
## # A tibble: 5 x 2
##   monthyear     b_my
##   <chr>        <dbl>
## 1 1-1995     0.368
## 2 1-1996     0.335
## 3 1-1997    -0.00292
## 4 1-1998    -0.00780
## 5 1-1999     0.0101
```

```
## [1] 0.04199936
```

Based on this model we can check with the test set how well the adjusted model is currently performing.

method	RMSE	Predictorweight
Mean(Rating)	1.0523666	NA
Movie	0.9411548	0.6699785

method	RMSE	Predictorweight
Movie + User	0.8578850	0.4223855
Movie + User + Genres	0.8578009	0.0781131
Movie + User + MonthYear	0.8578550	0.0419994

Using the adjusted model using only the mean(rating) and the movie effect + user effect + genre effect + monthyear effect the obtained RMSE using the test set is: 0.8578550

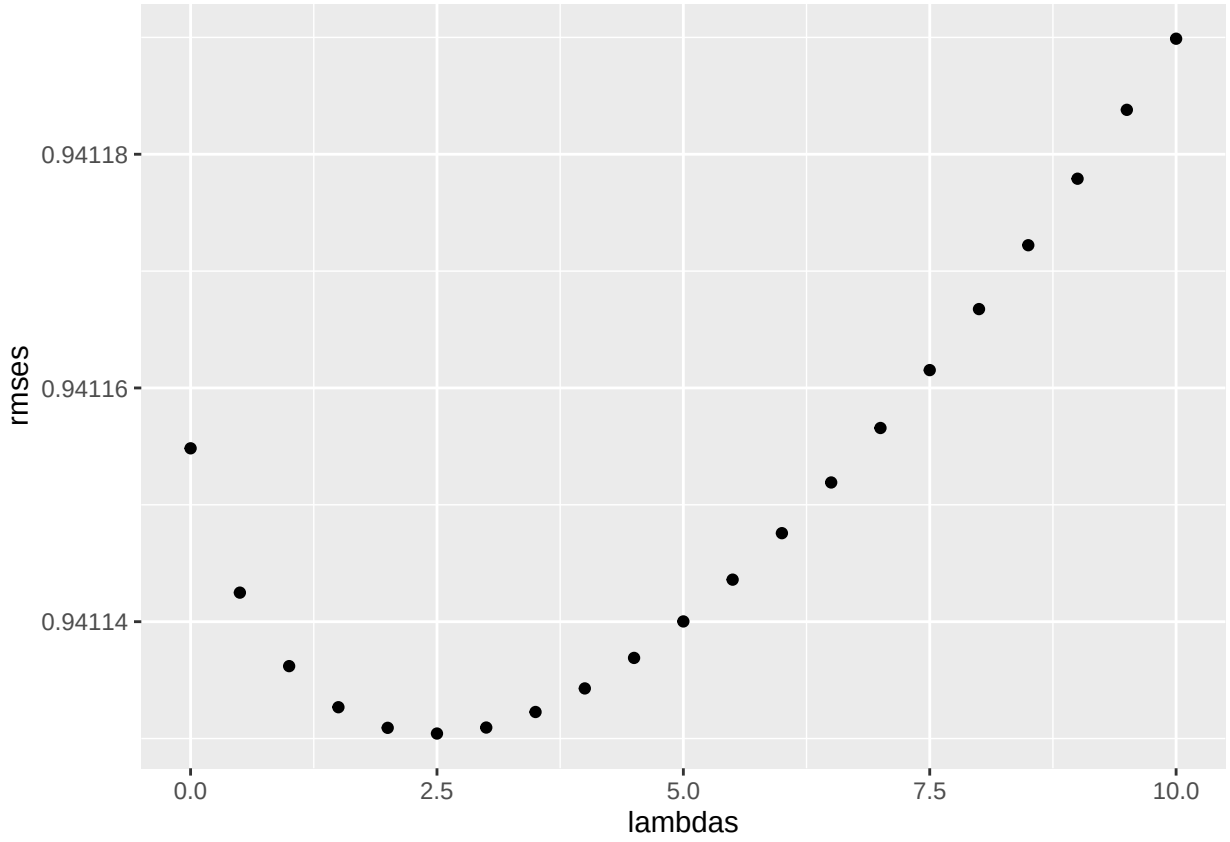
Variables	Correlation
Rating ~ Movie (Independent)	0.4474253
Rating ~ User (Independent)	0.3941398
Rating ~ Genre (Independent)	0.1128976
Rating ~ Monthyear (Independent)	0.0684102
Movie ~ User (Independent)	0.1315807
Rating ~ User (Movie)	0.3596069
Movie ~ Genre (Independent)	0.2529662
User ~ Genre (Independent)	0.0276370
Rating ~ Genre (Movie + User)	0.0746336
Movie ~ Monthyear (Independent)	0.0185652
User ~ Monthyear (Independent)	0.1594020
Rating ~ Monthyear (Movie + User)	0.0149289

As we can see in the correlation table the adjusted monthyear variable (dependent on movie and user) still does not correlate with the rating. Therefore genre does not provide enough new information to our model to make it worth including. We do not continue to use the monthyear predictor in our prediction model.

Regularization

Regularization is a technique that can be used to optimize the weight of the predictors. In the example of ratings for movies it needs a minimum ammount of ratings that the calculated avg. rating is meaningful. To regularize the parameters such as b_i the movie effect and b_u the user effect we are going to calculate lambda for each of them.

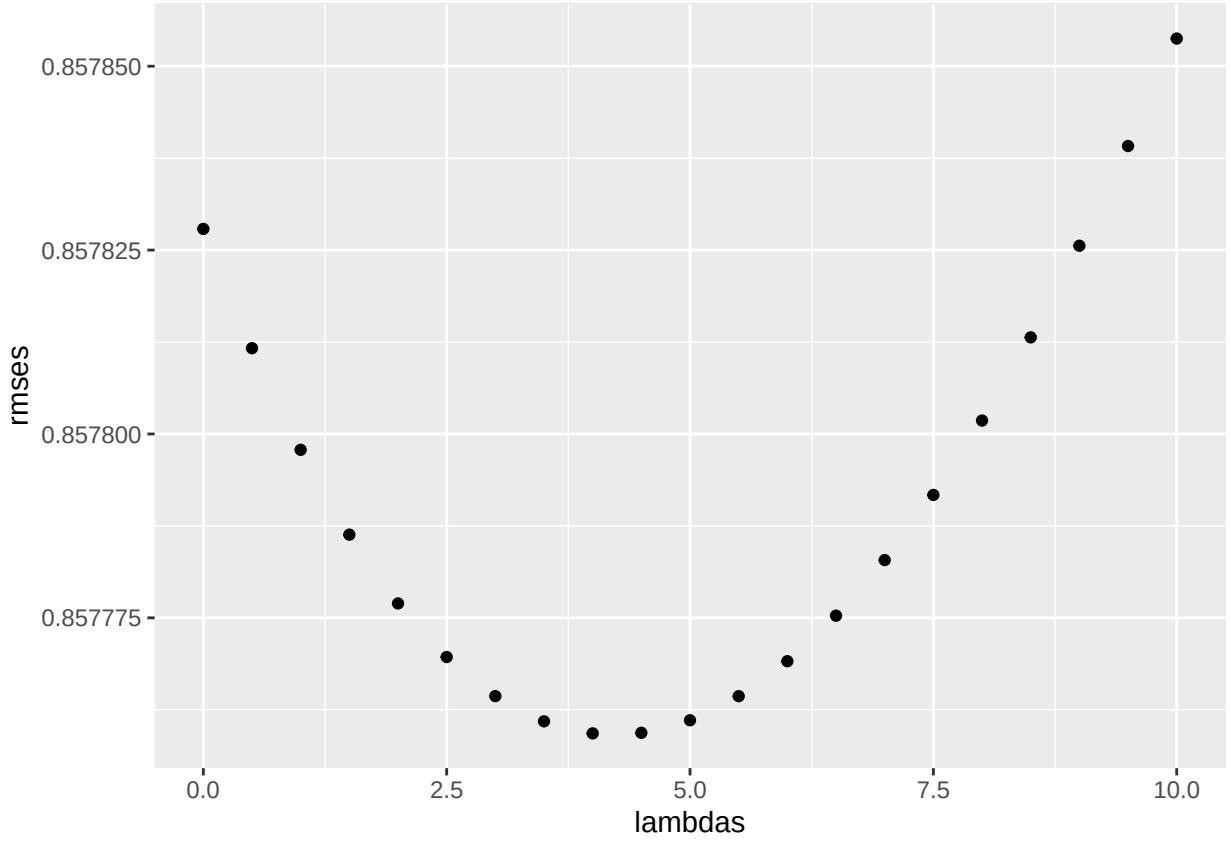
Crossvalidation is used to get the optimal lambda for b_i the movie effect. The effect the lambda has on the RMSE is displayed in the following plot:



As we see the optimal lamda for b_i the movie effect is:

```
## [1] 2.5
```

Crossvalidation is used to get the optimal lambda for b_u the user effect. The effect the lambda has on the RMSE is displayed in the following plot:



As we see the optimal lamda for b_u the user effect is:

```
## [1] 4
```

Once the lambdas are set for each of the biased effects such as b_i the movie effect, b_u the user effect, b_g the genre effect and b_{my} the monthyear effect we can predict the ratings using generalized effect values.

We are going to apply each of the lambdas step by step.

First we are going to use only the generalized $b_i(\lambda)$ movie effect:

$$Y_{i,u,g,my} = \mu + b_i(\lambda) + \epsilon_{i,u,g,my}$$

method	RMSE	Predictorweight
Mean(Rating)	1.0523666	NA
Movie	0.9411548	0.6699785
Movie + User	0.8578850	0.4223855
Movie + User + Genres	0.8578009	0.0781131
Movie + User + MonthYear	0.8578550	0.0419994
Generalized Movie	0.9411304	0.6089229

Then we are going to add $b_u(\lambda)$ the movie effect:

$$Y_{i,u,g,my} = \mu + b_i(\lambda) + b_u(\lambda) + \epsilon_{i,u,g,my}$$

method	RMSE	Predictorweight
Mean(Rating)	1.0523666	NA
Movie	0.9411548	0.6699785
Movie + User	0.8578850	0.4223855
Movie + User + Genres	0.8578009	0.0781131
Movie + User + MonthYear	0.8578550	0.0419994
Generalized Movie	0.9411304	0.6089229
Generalized Movie + User	0.8577593	0.4048624

Results

The final rating will be using $b_i(\lambda)$ the generalized movie effect and $b_u(\lambda)$ the generalized user effect.

Final Model:

$$Y_{i,u,g,my} = \mu + b_i(\lambda) + b_u(\lambda) + \epsilon_{i,u,g,my}$$

When training the final model on the edx data set we get a RMSE of 0.8630447 using the validation set to calculate the RMSE.

method	RMSE
Generalized Movie + User	0.8630447

Conclusion

Based on different movie ratings from different years the model can predict ratings for upcoming movies with a RMSE of 0.8630447. That means that true prediction value is on average in a range between -0.8630447 and 0.8630447. One of the limitations in this dataset is that there aren't enough attributes that can be used for more accurate predictions. Either the other attributes do not give us more information (monthyear) or the attribute correlates with an attribute already used in the prediction (genre ~ movie), so including does not lead to a more accurate prediction.

Feel free to network with me on LinkedIn: https://www.linkedin.com/in/mike-miemczok-432206209/

or Xing: https://www.xing.com/profile/Mike_Miemczok

Also check out my github for future work: https://github.com/mikemiemczok