Movie Recommendation system based on MovieLens 10M data

HarvardX project submission - PH125.9x Data Science: Capstone

Author: Martin Haitzmann (martin.c.haitzmann@gmail.com)

Last Update: 10 August, 2020

Contents

1 Overview

Recommendation systems are one of the most popular data science methodologies coming from the field of machine learning (Irizarry 2019, 499). The main goal of this project is to predict how a user would rate a specific movie, based on how the user rated other movies and how the movie is rated by other users. Several techniques outlined in Irizarry (2019, chap. 33.7) will be applied on the "MovieLens" data-set combined with other findings resulting from the explanatory analysis of the input data.

The movieLens database maintained by GroupLens research lab[1] contains different datasets. As the full data-set is huge, affording quite some computing power for proper processing, this project is based on the "10Mversion"[2] containing around 10 million ratings (~10000 movies rated by ~70000 users - respectively rather userId's).

The 10M data is downloaded and split into an edx data-set (for analysis and training the algorithm) and a validation data-set (for final testing of the developed model)[3].

Dimensions (rows cols) of edx: 9000055 6

Dimensions (rows cols) of validation: 999999 6

[1] https://grouplens.org/
[2] https://grouplens.org/datasets/movielens/10m/
[3] R code provided by the course facilitators

Are there any missings (NA) in the data?: FALSE

The data entries itself look like:

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

So every data-row represents a rating for a specific movie of one user/userId and contains some additional information (timestamp, movie title, genres). If every user had rated every movie, the data-set should contain about 750 Mio. data-rows instead of 10 Mio. This implies that not every user rated every movie. The idea of this recommendation system is to predict what a specific user would rate a movie he has not rated.

As outlined by the course facilitators, one should develop one's own algorithm using the edx data-set only (split the edx data into separate training and test sets). For a final test of the best performing algorithm, movie ratings in the validation set (the final hold-out test set) should be predicted as if they were unknown. RMSE should be used to evaluate how close the predictions are to the true values in the validation set (the final hold-out test set).

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{u,i}\left(\hat{y}_{u,i} - y_{u,i}\right)^2}$$

with $y_{u,i}$ as the rating for movie i by user u and with N being the number of user/movie combinations and the sum occurring over all these combinations.

The **analysis and methods section** includes a descriptive analysis of the downloaded data. The gained insights lead to additional ideas on how different explanatory variables might further influence our response variable (rating) and as a consequence be included in the modeling approach. The development of the models is based solely on the edx data set, which will be split up in a train and test set for this purpose. As plenty of work was already done by many data scientists on possible approaches, this paper will definitely start with introducing the baseline models with the effects described e.g. in Irizarry (2019, chap. 33.7). Even the winners of the Netflix challenge give high importance to the baseline predictors[4]:

> Of the numerous new algorithmic contributions, I would like to highlight one – those humble baseline predictors (or biases), which capture main effects in the data. While the literature mostly concentrates on the more sophisticated algorithmic aspects, we have learned that an accurate treatment of main effects is probably at least as significant as coming up with modeling breakthroughs.

The ***explanatory data analysis*** reveals among others that there might be some *userspecific calender/daytime effect*. One might think of it as: does the daytime (morning, afternoon, evening, night) when people rate films have an influence on the ratings? What about weekdays vs. weekends? It seems obvious that this might not produce a general effect, as people life in different time zones, have different working days, ..., but especially in connection with single userId's that gave enough ratings, there might be a correlation that explains some of the variability.

Finally, the model with the lowest RSME is trained on the whole edx data set and **tested with the validation set**. One of the goals of the this capstone project is to reach an RMSE < 0.86490. ***The final model chosen in this small project reached a RMSE on the validation data-set of 0.85842 and the improvements can therefore be considered to be successful.***

[4] http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/

2 Analysis and methods

This section first focuses on an exploratory analysis of the variables prevalent in the input data-set (edx). Descriptive statistics should ideally reveal patterns in explanatory variables that might have a significant impact on the response variable (rating). The insights are then further used to fine tune the models. The models itself are based on the baseline predictors explained in Irizarry (2019, chap. 33.7), as these predictors are well explained and already show a very good performance. ***With additional information from the exploratory analysis, it is especially tried to improve the "Movie + User Effects Model" and apply regularization on such an improved model.***

2.1 Explore the data

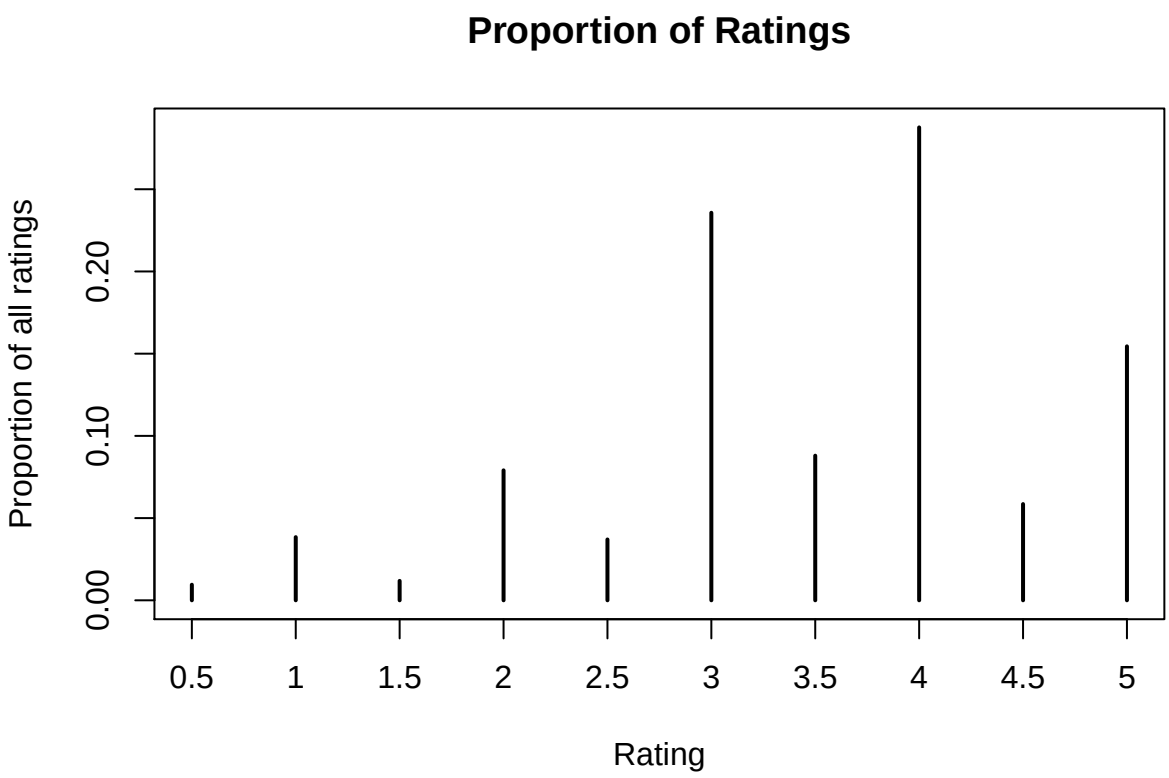
Inspect all variables in detail and think of possible ways these variables might explain part of the value of the target variable "rating".

2.1.1 Response variable (rating)

Ratings are prevalent in 0.5 steps. Most of the ratings refer to 4, 3 and 5 stars. The mean rating is 3.5124652.

```
##
## Frequency of Ratings:
```

```
##
##     0.5       1     1.5       2     2.5       3     3.5       4     4.5       5
##   85374  345679  106426  711422  333010 2121240  791624 2588430  526736 1390114
##     Sum
## 9000055
```



```
##
## Basic summary statistics of ratings:
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   3.000   4.000   3.512   4.000   5.000
```

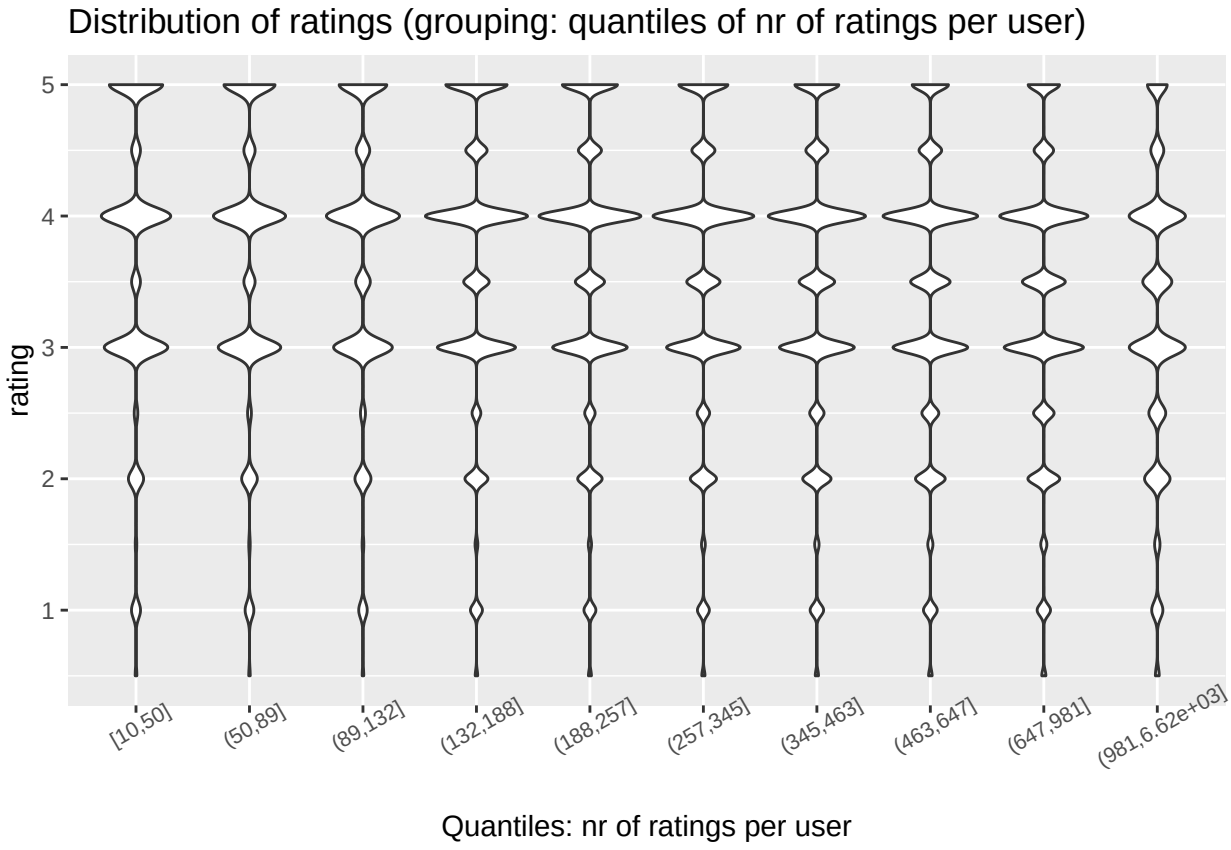
2.1.2 Explanatory variables

2.1.2.1 userId User specific influence should from a theoretical point of view of course be measured in a recommendation system that is based on users' likes and dislikes. To simplify the model and reduce computing time/memory withsuch large datasets, one might be able to reduce complexity by identifying groups of users with similar behavior. E.g. analyse the input data if high frequency raters tend to give other ratings than the "one time user", ...

```
##
## Basic summary: Nr of movieId's rated per user.
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.0    32.0    62.0   128.8   141.0  6616.0
```

```
##
##  69878 users (more precisely userId's) rated the movies in the given data-set.
```



The data-set does not contain "one time raters". The lowest number of movies rated by one user is 10. The median user rated 62 movies. The maximum number of movies rated by a user rated is even 6616[5]. There seems to be no obvious significant difference in the behavior of users being grouped to the number of rated movies per user according to quantiles, as illustrated by the violin plot above.

[5]just a remark: supposing a movie to last one and a half our this is quite some life time this user spent watching - 413.5 complete days nonstop

2.1.2.2 movieId (and title) There should be a movie effect, as some movies get excellent ratings whereas others do not. These effects (blockbuster, obscure movies, old classic films, ...) are exhaustively explored in Irizarry (2019, 638 ff.). That is why here, the analysis regarding movies is not further deepened. However, it is at least worth to see if the variables movieId (and title) have additional information to offer.

10677 are in the given data-set.

The variables movieId and title belong together so first check if these two variables are consistent. Are there duplicates?

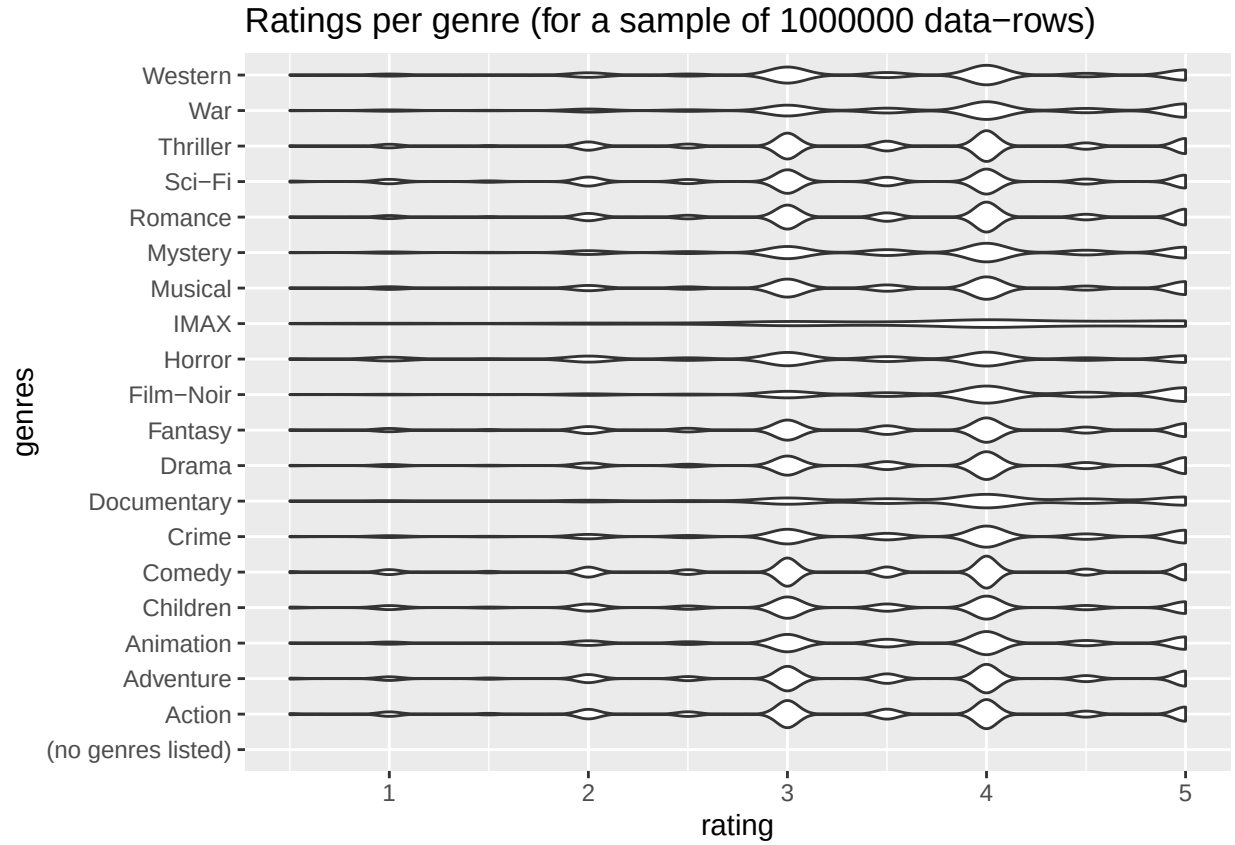
Duplicated title:

movieId	title
34048	War of the Worlds (2005)
64997	War of the Worlds (2005)

Interestingly one movie title has 2 different ID's. Normally one should reconcile that in the basic data. However, as it would affect edx and validation set and comparability between learners doing this capstone project, the data inconsistency is left as it is.

2.1.2.3 genres Movies belonging to different genres are all displayed in one compact variable. However, such data presentation does not allow easy processing. Ideally a column for every genre is created and filled with 1/0 (resp. TRUE/FALSE) to analyse the effect of each genre, e.g. fitting regressions. Due to computing power restrictions, in this case it is only explored if there are obvious differences in the ratings for some genres.

As the violin plot suggests, Film-Noir, IMAX and Documentary tend to have nearly no ratings below three.



That is why movies belonging to one of this 3 genres are subsumed in a category genre_FN_IMAX_DOC.

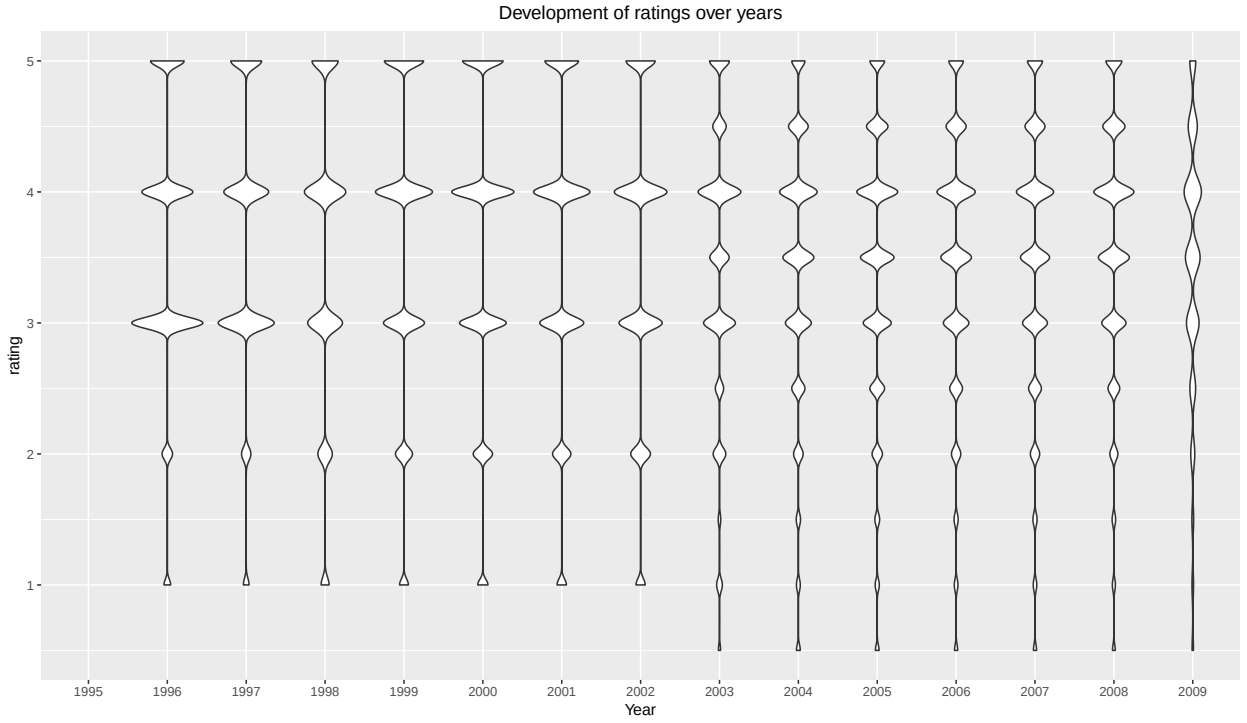
Fitting a simple regression, the effect is with very high significance estimated to be about 0.4 (stars), so the effect of this category is further investigated in the model approach.

```
##
## Call:
## lm(formula = rating ~ genre_FN_IMAX_DOC, data = edx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4083 -0.5026  0.0917  0.4974  1.4974
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.5026419  0.0003572  9806.3   <2e-16 ***
## genre_FN_IMAX_DOC 0.4056112  0.0022952   176.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.058 on 9000053 degrees of freedom
## Multiple R-squared:  0.003458,   Adjusted R-squared:  0.003458
## F-statistic: 3.123e+04 on 1 and 9000053 DF,  p-value: < 2.2e-16
```

2.1.2.4 timestamp The variable timestamp contains information on when the rating was saved in the database. In order to make a useful exploratory analysis, it is first essential to transform the data into user friendly format and extract year, month, day, ... information.

Regarding further analysis one could inspect how ratings developed over the years. Furthermore it could be interesting to see if there is a weekend/weekday effect. Measuring holiday effects in general would be interesting, but as there is a lack of geoinformation where the user lived, this cannot be done within the scope of this analysis.

Interesting information could also be revealed searching for a daytime effect: there will of course be no general effect that ratings in the morning are significantly different to evening ratings (the more as there are different time zones and without geoinformation it is impossible to find out what daytime actually was when a rating was given resp. saved in the database). However, splitting the 24 h day into slices and combining it individually with the userId, could reveal some pattern for specific users - e.g. do frequent raters rate differently when they do the ratings in the morning?.



The above illustration reveals that the half * ratings were only introduced in 2003. So it would make sense to take this in the model building into consideration. Furthermore the violin plot reveals a different pattern of ratings in 2009. However this is just due to the lower number of entries in 2009, as the last rating in the edx data-set was saved in Jan 2009.:

1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
2	17	64667	15908	15208	97902	80153	59906	67995	57081	81193	69536	66336	63386	13123
0	307	39042	8565	8310	72615	49070	46808	52750	79955	85372	58312	62505	54573	0
0	5948	60294	10530	5264	87614	41883	37589	57918	64464	197764	63619	61959	49469	0
0	31478	50179	11016	8869	103708	32877	54281	45520	48817	130131	60777	51058	42434	0
0	110958	55887	5740	8442	65524	30910	42691	54467	58930	93317	47627	49981	40919	0
0	148463	48515	8505	6705	53468	93407	35912	53243	53412	89030	51672	58610	47633	0
0	128074	34441	48537	7049	93586	90203	38970	56382	54828	72727	64160	54715	58935	0
0	128434	0	24614	6601	142430	67826	44205	45037	61381	61887	51420	37485	53178	0
0	87929	11326	11230	19547	47256	56037	35316	46715	57149	63178	47150	37244	47277	0
0	112467	12938	8071	214812	38148	42335	36758	46846	51841	58691	59647	47979	98977	0
0	106610	24057	9835	144230	244024	50964	43223	48010	55203	65716	51716	48955	82970	0
0	82087	12755	19083	264856	98074	47690	49300	45055	48368	60271	63679	52341	56989	0

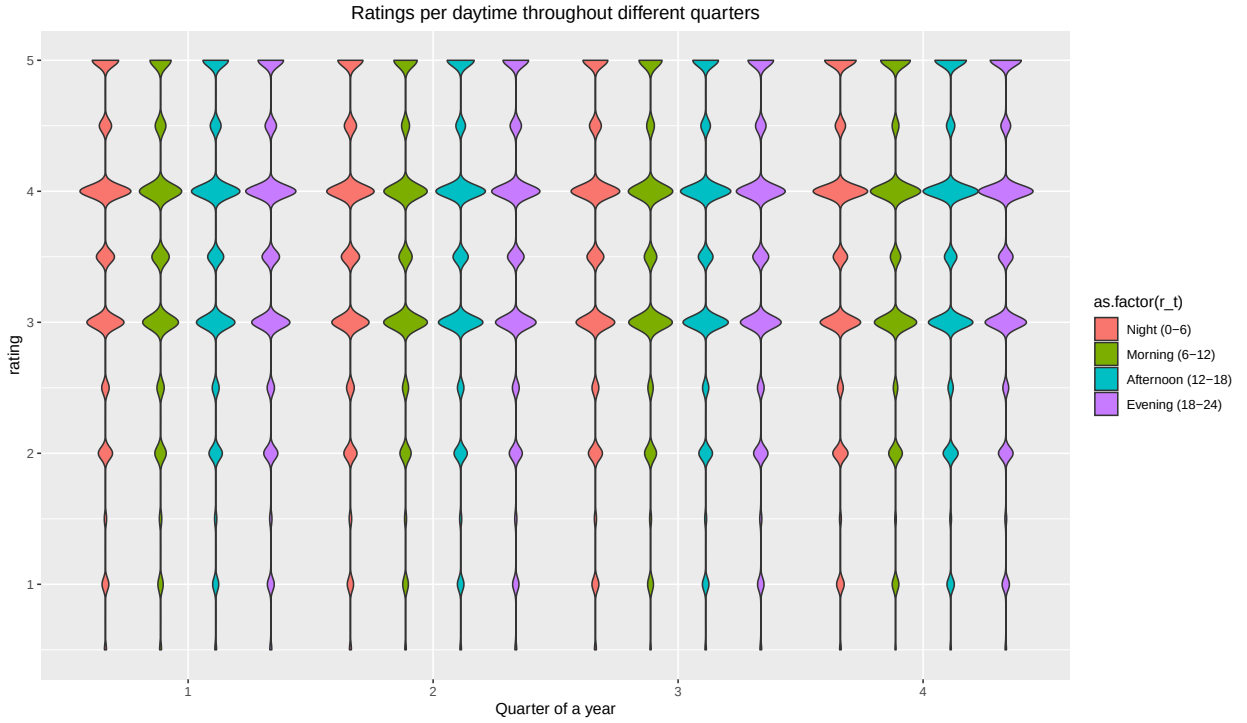
Is there an effect between ratings before 2003 and afterwards, due to the introduction of half stars?

```
##
## Call:
## lm(formula = rating ~ before2003, data = a)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9658 -0.5571  0.4430  0.5342  1.5342
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

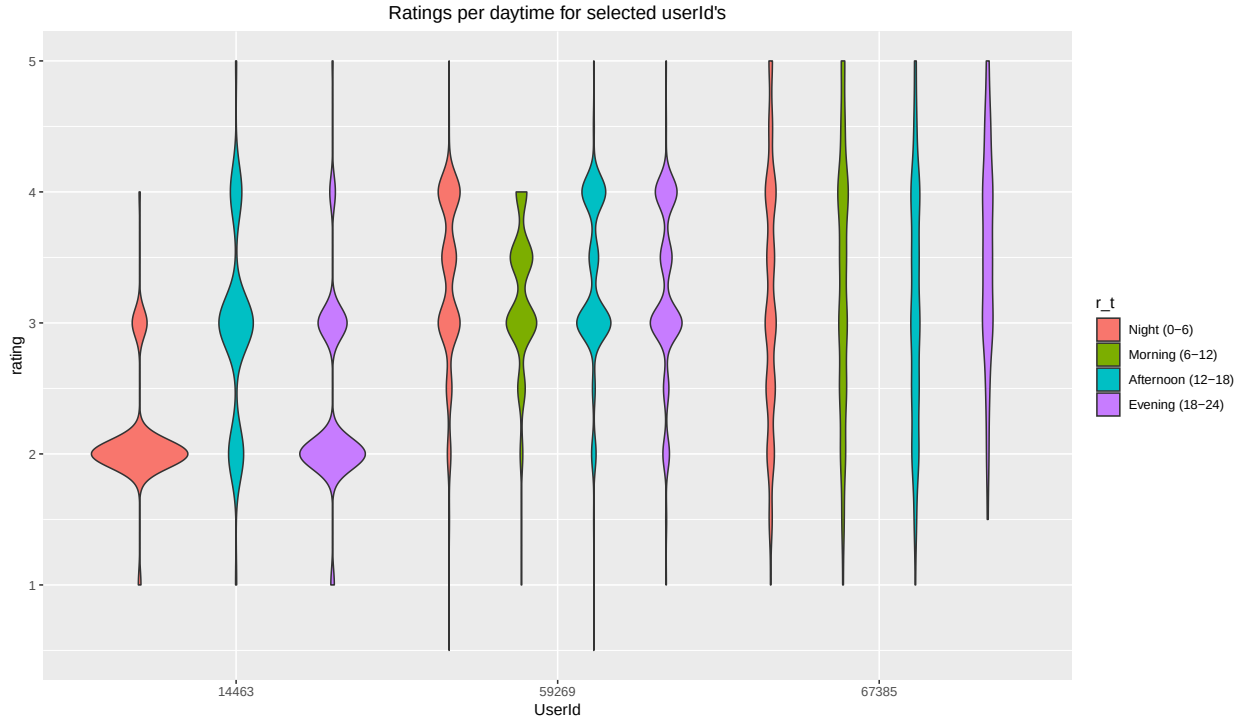
```
## (Intercept) 3.4658283  0.0005051  6861.9   <2e-16 ***
## before2003  0.0912256  0.0007064   129.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.059 on 9000053 degrees of freedom
## Multiple R-squared:  0.00185,    Adjusted R-squared:  0.001849
## F-statistic: 1.668e+04 on 1 and 9000053 DF,  p-value: < 2.2e-16
```

The fitted regression suggests a significant effect but with not even 0.1 (stars) difference it is not further evaluated in the model section.

A view on ***daytimes*** (in the below figure per quarter of a year) generally does not reveal any specifics as expected (timezones, ...):



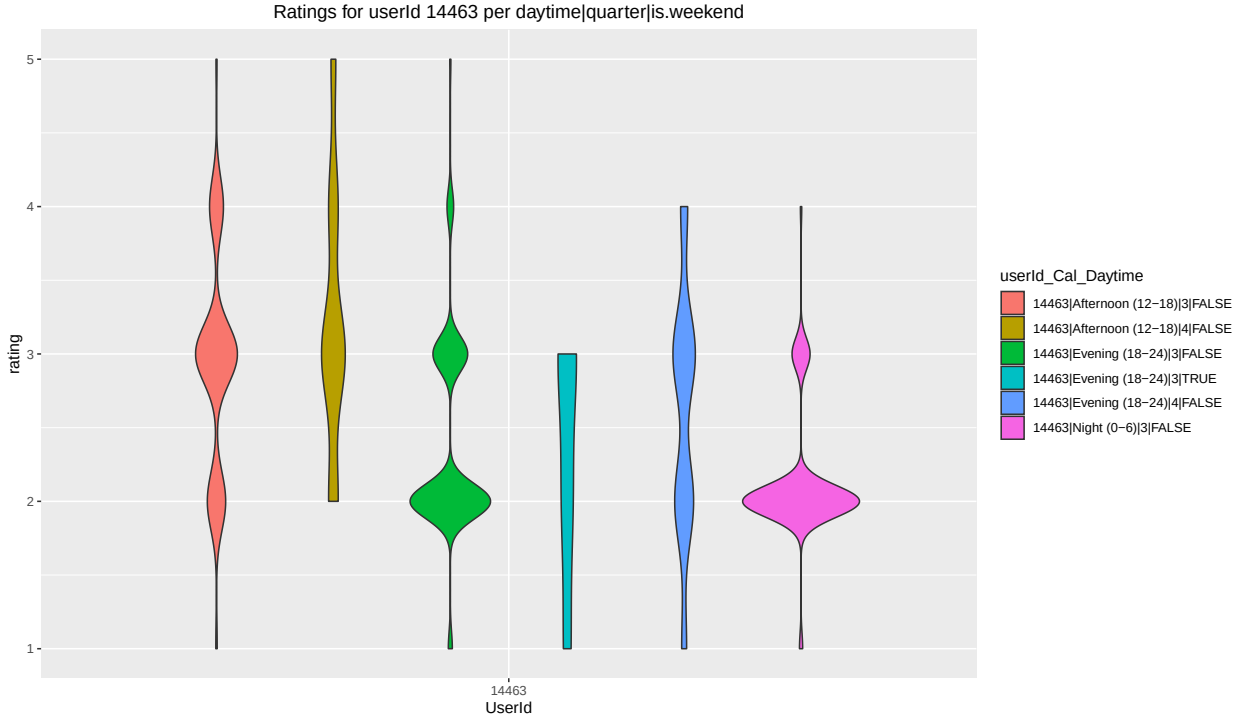
However user specific daytimes (in the below figure for high frequent raters with more than 4000 ratings) reveal some potential influence:



.

User 14463 never gave a 5* when rated in the daytime phase that is defined here as night[6]. Furthermore, this user gives in that daytime substantially more 2* than in the 'afternoon'.

Can the effect be even more deepened when calendar information is added?



Userspecific daytime and calendar information should be introduced in the modeling approach.

[6]based on the conversion of the timestamp variable. Do not take the wording too serious, as it is just useful for making distinctions. Correctly it is probably the daytime of the place where the server recording the ratings is located

2.2 Develop models

The development of the models is based solely on the edx data set, that is split up in a train and test set for this purpose. As plenty of work was already done by many data scientists on possible approaches, this paper starts applying the baseline models, as the effect and impacts are well explored and documented, e.g. in Irizarry (2019, chap. 33.7). Even the winners of the Netflix challenge give high importance to the baseline predictors, as mentioned in the introduction.

Apply models with known effects from Irizarry (2019, chap. 33.7):

method	method_short
Just the average	model_0_rmse
Movie Effect Model	model_1_rmse
Movie + User Effects Model	model_2_rmse

However, further to this basic models two effects that popped up in the exploratory analysis are introduced. One effect covering the genres "Film-Noir", "IMAX" and "Documentary" in one category ***genre_FN_IMAX_DOC***. The second effect deals with a combined calendar/daytime information showing in what quarter and daytime a user saved its rating. Furthermore the information if it was a weekday or a weekend is processed. A combined ***userId_Cal_Daytime*** variable is therefore introduced. Be aware that userId_Cal_Daytime may introduce NA's when applied to the test_set, simply because a specific key representing userId_Cal_Daytime may not be prevalent. In such cases apply only the user Effect measured with the key variable userId.

	method	method_short
4	Movie + User + genre_FN_IMAX_DOC Effects Model	model_2a_rmse
5	Movie + User/Cal_Daytime Effects Model	model_2b_rmse

In case one of the effects brings an improvement, regularization is carried out on that model, to penalize higher deviations.

	method	method_short
6	Regularized Movie + User/Cal_Daytime Effects Model	model_3_rmse

3 Results

The results section first covers the results for the model development. The best model is then chosen to get the final RMSE estimate based on edx as train_set and validation data as test_set.

3.1 Preliminary Modeling (edx split into train_set and test_set)

```
## model development: edx is split into train_set and test_set
##
## Model __Just the average__ processed.
##
## Model __Movie Effect Model__ processed.
##
## Model __Movie + User Effects Model__ processed.
##
## Model __Movie + User + genre_FN_IMAX_DOC Effects Model__ processed.
##
## Model __Movie + User/Cal_Daytime Effects Model__ processed.
##
## Model __Regularized Movie + User/Cal_Daytime Effects Model__ processed.
##
## Iteratively selected minimizing penalty term lambda:  4.2
```

method	method_short	RMSE
Just the average	model_0_rmse	1.06056
Movie Effect Model	model_1_rmse	0.94399
Movie + User Effects Model	model_2_rmse	0.86664
Movie + User + genre_FN_IMAX_DOC Effects Model	model_2a_rmse	0.86827
Movie + User/Cal_Daytime Effects Model	model_2b_rmse	0.86337
Regularized Movie + User/Cal_Daytime Effects Model	model_3a_rmse	0.85984

RMSE shows that the introduction of a *__userspecific calendar and daytime effect__ brings an improvement. The best performing model **Regularized Movie + User/Cal_Daytime Effects Model** is chosen to be applied for the final evaluation.

3.2 Final Modeling (edx is train_set and validation test_set)

As we can see from the result table below, ***Regularized Movie + User/Cal_Daytime Effects Model*** achieved the target RMSE when the complete edx data set is trained and the validation data used as test_set. ***The project goal for earning full points is achieved as the RMSE stays quite far below the target value***[7]***.***

```
## final run: edx is used as train_set, validation as test_set
##
## Model __Regularized Movie + User/Cal_Daytime Effects Model__ processed.
##
## Penalty term lambda was:  4.2
```

method	method_short	RMSE
Regularized Movie + User/Cal_Daytime Effects Model	model_3a_rmse	0.85842

4 Conclusion

It was shown that the introduction of a combined ***user + calender + daytime effect*** could improve the performance of the model. The final RMSE was 0.8584185 and therefore quite below the threshold predefined by the course facilitators of RMSE < 0.86490[8].

The developpment of this recommendation system was done within time and processing power resrtictions to above all achieve the predefined goals of this capstone project. Future improvements should contain approaches using matrix factorisation on residuals of the baseline models, e.g. by experimenting with the recommenderlab package.

5 Appendix

```
print(sessionInfo())
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
```

[7] RMSE < 0.86490 predefined by course facilitators
[8] threshold for earning full points

```
##
## Matrix products: default
##
## Random number generation:
##  RNG:     Mersenne-Twister
##  Normal:  Inversion
##  Sample:  Rounding
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] data.table_1.12.8 caret_6.0-86      lattice_0.20-38   forcats_0.5.0
##  [5] stringr_1.4.0     dplyr_1.0.0       purrr_0.3.4       readr_1.3.1
##  [9] tidyr_1.1.0       tibble_3.0.2      ggplot2_3.3.2     tidyverse_1.3.0
## [13] kableExtra_1.1.0  knitr_1.29        lubridate_1.7.9
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.1           jsonlite_1.7.0       viridisLite_0.3.0
##  [4] splines_3.6.2        foreach_1.5.0        prodlim_2019.11.13
##  [7] modelr_0.1.8         assertthat_0.2.1     stats4_3.6.2
## [10] blob_1.2.1           cellranger_1.1.0     yaml_2.2.1
## [13] ipred_0.9-9          pillar_1.4.6         backports_1.1.8
## [16] glue_1.4.1           chron_2.3-55         pROC_1.16.2
## [19] digest_0.6.25        rvest_0.3.5          colorspace_1.4-1
## [22] recipes_0.1.13       htmltools_0.5.0      Matrix_1.2-18
## [25] plyr_1.8.6           timeDate_3043.102    pkgconfig_2.0.3
## [28] broom_0.7.0          haven_2.3.1          scales_1.1.1
## [31] webshot_0.5.2        gower_0.2.2          lava_1.6.7
## [34] farver_2.0.3         generics_0.0.2       ellipsis_0.3.1
## [37] withr_2.2.0          nnet_7.3-12          cli_2.0.2
## [40] survival_3.1-8       magrittr_1.5         crayon_1.3.4
## [43] readxl_1.3.1         evaluate_0.14        fs_1.4.2
## [46] fansi_0.4.1          nlme_3.1-142         MASS_7.3-51.4
## [49] xml2_1.3.2           class_7.3-15         tools_3.6.2
## [52] hms_0.5.3            lifecycle_0.2.0      munsell_0.5.0
## [55] reprex_0.3.0         compiler_3.6.2       rlang_0.4.7
## [58] grid_3.6.2           iterators_1.0.12     rstudioapi_0.11
## [61] labeling_0.3         rmarkdown_2.3        ModelMetrics_1.2.2.2
## [64] gtable_0.3.0         codetools_0.2-16     DBI_1.1.0
## [67] reshape2_1.4.4       R6_2.4.1             stringi_1.4.6
## [70] Rcpp_1.0.5           vctrs_0.3.1          rpart_4.1-15
## [73] dbplyr_1.4.4         tidyselect_1.1.0     xfun_0.15
```

References

Irizarry, Rafael A. 2019. “Introduction to Data Science.” https://leanpub.com/datasciencebook.