Data Science Capstone Project

Harvard University, edx

Pedro J. Llanos

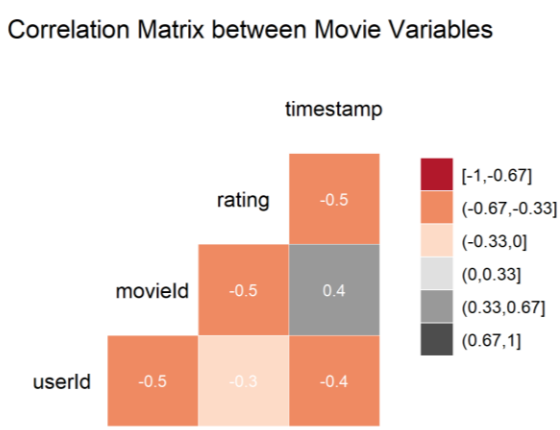
Abstract

This project was part of the final Data Science Capstone course and highlights the usefulness of machine learning for movie ratings applications to improve our understanding how these ratings are made and make proper decisions when providing a movie recommendation. The time period to provide these movie ratings was from 1996 to 2008. Several methods were used to explore this analysis, such as the least square method and the regularization to leverage enhance the performance of the model while reducing the root mean square error (RMSE). The final RMSE achieved with the proposed model using machine learning was 0.8648170, which is near 20% reduction in the RMSE with respect the average method. Comparative analysis between each method is provided, and the estimated movie rating probabilities for various data sorting.

I. INTRODUCTION

For this project, one of the first things it is looked at is the correlation matrix of a data set "edx" (see R code) to find out if there is any strong relationship between variables, and the direction (positive or negative) of such relationship. In this case, we observe that the relationship between the rating and the timestamp is based on a negative correlation, movieId and rating have a negative correlation, movieId and timestamp have a positive correlation, userId have a negative correlation with movieId, rating and timestamps.

There is a total of 10666 movies and 796 different genres. There are 69878 different users. Ratings (1 to 5) are given for all these movies from 1996 to 2008.



Note that the genre does not appear since it is not a numeric value, however we will see that there is a correlation between the movieId and the genre of the movie.

II. METHODOLOGY

Different models are used to analyze the effect various movie factors, such as the ID of the movie, the ID of the user, the genre, the timestamp, a combination of these, and finally a regularized approach which is expected to have a better performance. These models are described below:

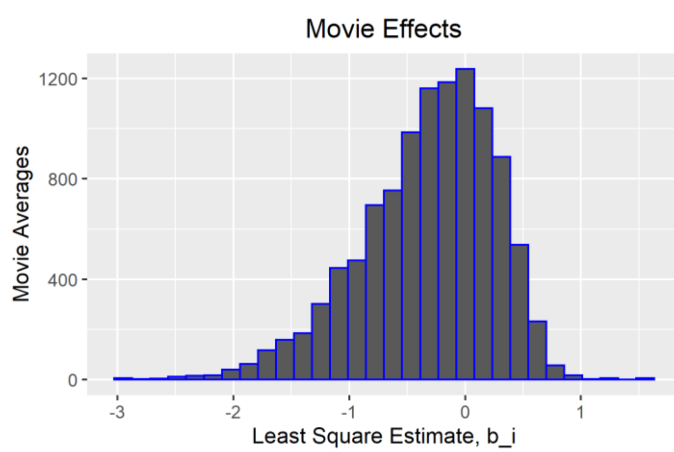
1. **First model**: Analyze the naïve RMSE with just the "Average". This model assumes that the rating is the same for all movies and users, and their differences are associated with random variations:

$$Y_{u,i} = \mu + \varepsilon_{u,i}$$

where μ is the average ("true" rating) for all movies, and $\varepsilon_{u,i}$are the independent errors sampled from the same distribution centered at zero.

2. **Second model**: Analyze the "movieId" effect. This model assumes another extra term, b_i, or movie-specific effect with respect to the previous model, as seen below:

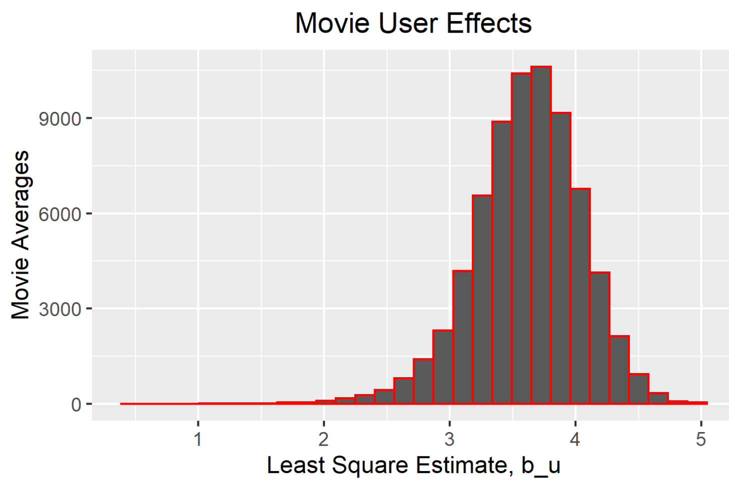
$$Y_{u,i} = \mu + b_i + \varepsilon_{u,i}$$



3. **Third model**: Analyze the "userId" effect. This model assumes another extra term with respect the previous model:

$$Y_{u,i} = \mu + b_i + b_u + \varepsilon_{u,i}$$

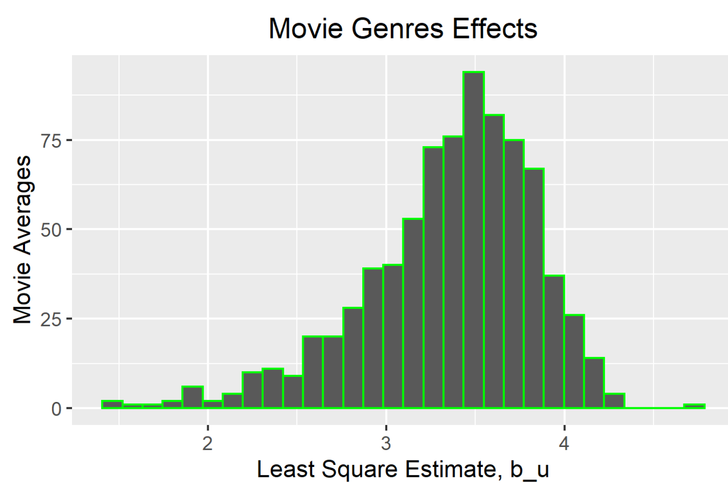
where b_u is the user-specific effect.



4. **Fourth model**: Analyze the "movieId", "userId" and "genres" effects. This model assumes another extra term with respect the previous model:

$$Y_{u,i} = \mu + b_i + b_u + b_g + \varepsilon_{u,i}$$

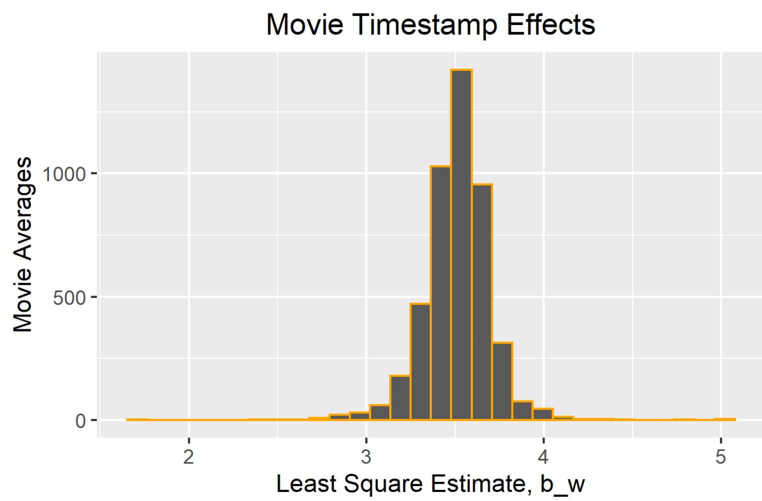
where b_g is the genre-specific effect.



5. **Fifth model**: Analyze the "movieId", "userId", "genres", and "timestamp" effects. This model assumes another extra term with respect the previous model:

$$Y_{u,i} = \mu + b_i + b_u + b_g + b_t + \varepsilon_{u,i}$$

where b_t is the time-specific effect.

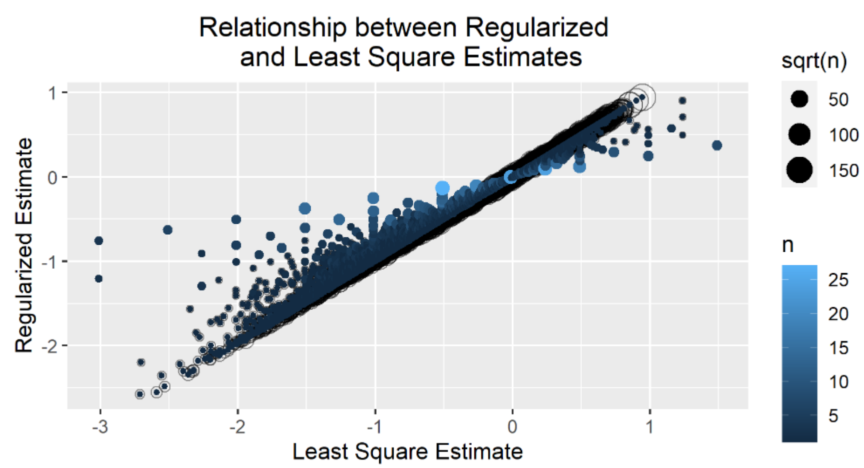


6. **Sixth model**: Analyze the regularization on the "movieId" and "userId" effects. This model is based on the minimization of the equation with a penalty, instead of the minimization of the least squares method:

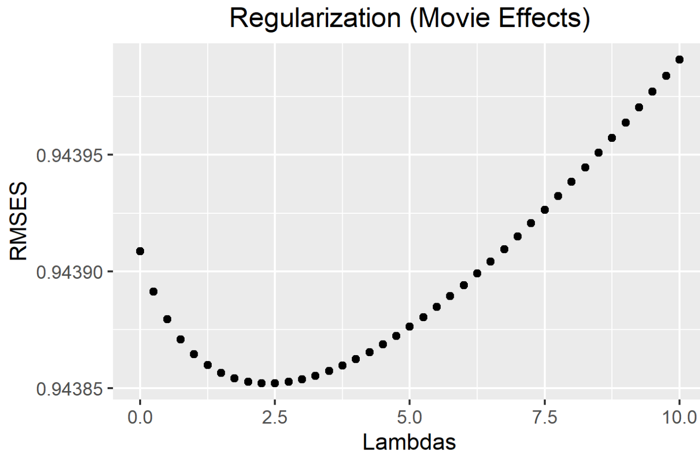
$$\frac{1}{N}\sum_{u,i}\left(y_{u,i}-\mu-b_i\right)^2+\lambda\sum_{i}{b_i}^2$$

where the first term is the well-known least squares method and the second term is the added penalty. The values of b_i that minimize the above equation can be obtained as follows:

$$\widehat{b_\iota}(\lambda)=\frac{1}{\lambda+n_i}\sum_{u,i}^{n_i}\left(Y_{u,i}-\hat{\mu}\right)$$



Next, we can choose the penalty terms, since we know λ a tuning parameter. We can use cross-validation to choose it. In this case we pick the movie effect for the regularization case.

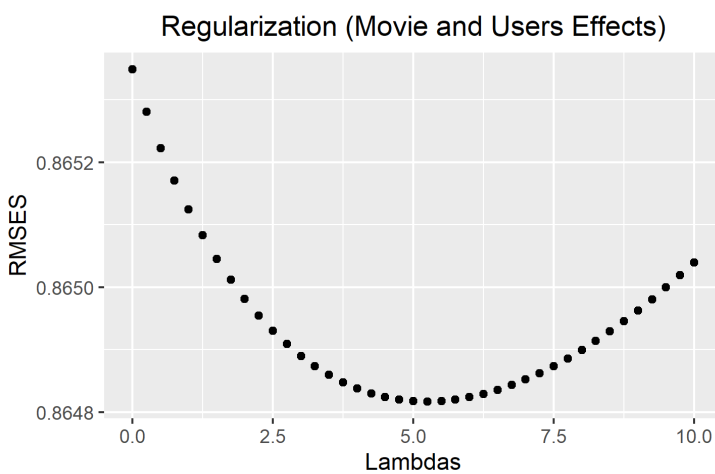


Note, that there is a $\lambda = 2.5$ that minimizes the RMSE. Can we find another λ that minimizes even more the RMSE? For this, we can use the regularization to estimate the user effects too.

$$\frac{1}{N}\sum_{u,i}\left(y_{u,i} - \lambda - b_i - b_u\right)^2 + \lambda\left(\sum_i {b_i}^2 + \sum_u {b_u}^2\right)$$

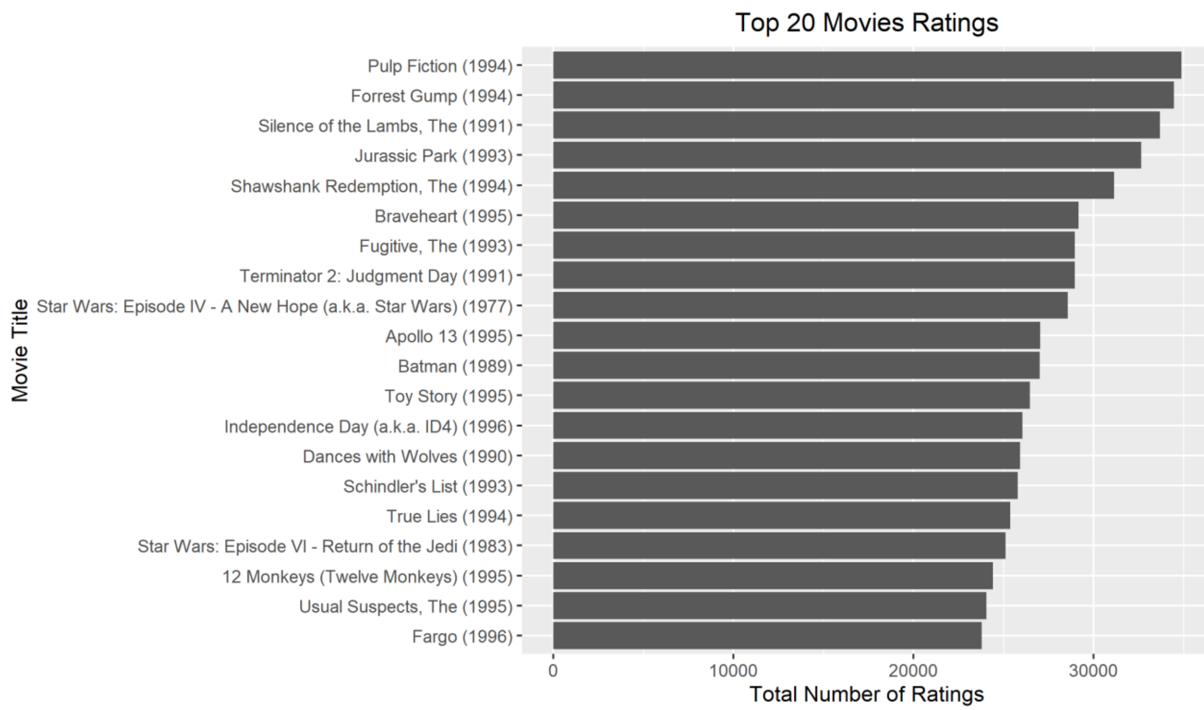
$$\widehat{b_u}(\lambda) = \frac{1}{\lambda + n_i}\sum_{u,i}^{n_i}\left(Y_{u,i} - \hat{\mu} - \widehat{b_\iota}\right)$$

With this improved model, we get a more precise λ that minimizes the RMSE ($\lambda = 5.25$).

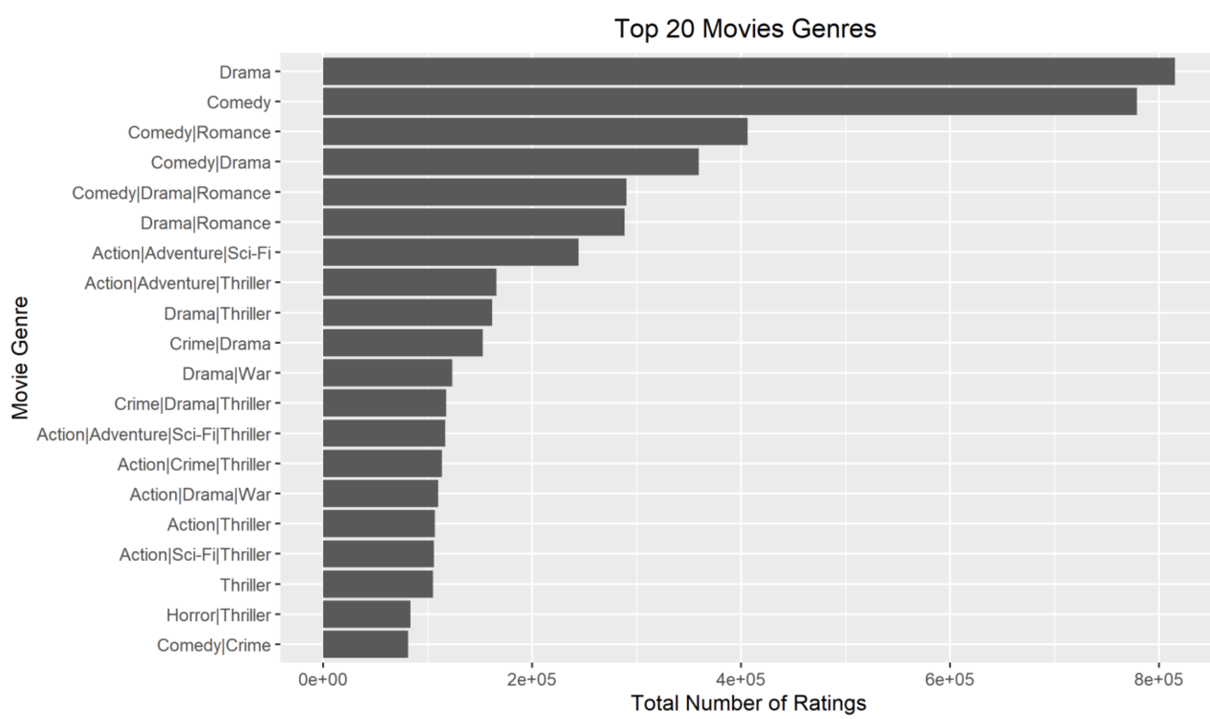


III. RESULTS

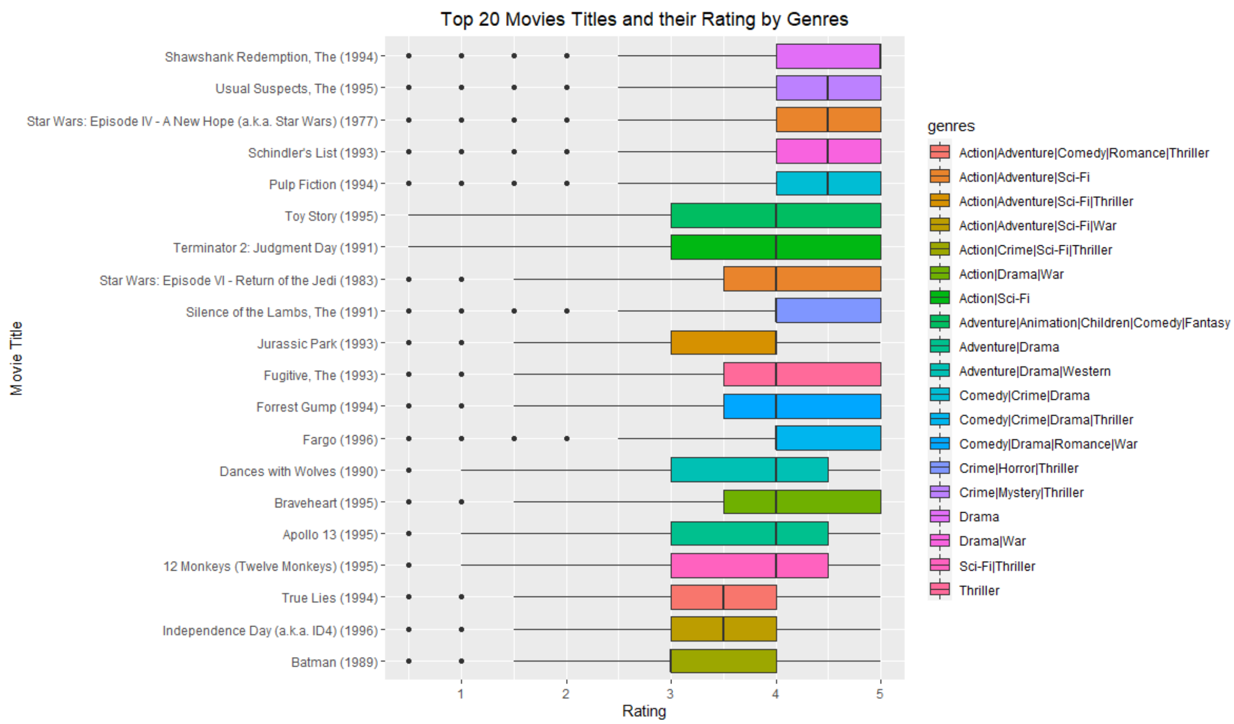
For the preliminary inspection of the data, we looked at the top 20 movies with highest total number of ratings.



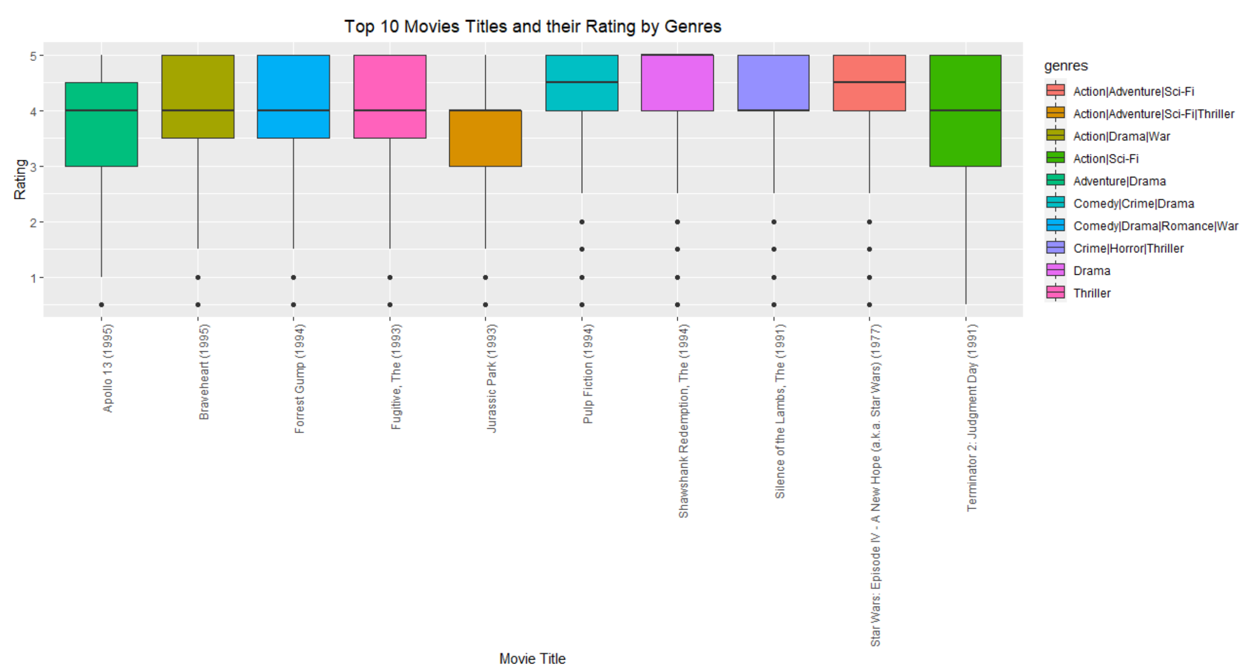
and the top 20 movies by genre:



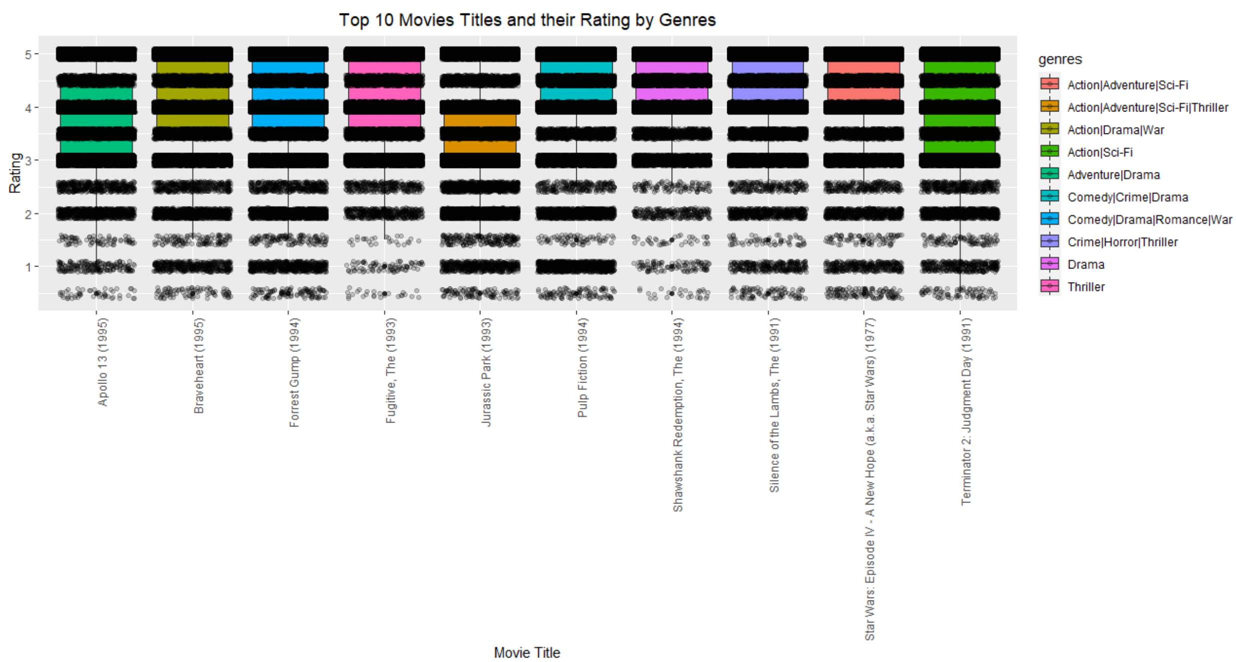
Next, we show a boxplot highlighting the top 20 movies and their rating by genres



and the top 10 movies with their ratings by genre:



The next plot may not appear to be too helpful at first glance, but several interesting observations can be extracted that can be confirmed with our code. It shows a visual of some of the top movies (eg. the Shawshank Redemption) with very few times of ratings for 0.5, 1, 1.5 and even 2.5. Similarly, you can appreciate other top movies and how many few times they got hit for that rating. Another visual observation is that the total number of low ratings for these top 10 movies is for 0.5 and 1.5 ratings, which is confirmed by the code. On the other hand, the most given ratings are 4, 3, 5, 3.5 and 2 from most to least. For these most given ratings it is impossible to visualize them, so we had to compute these as indicated in the code.

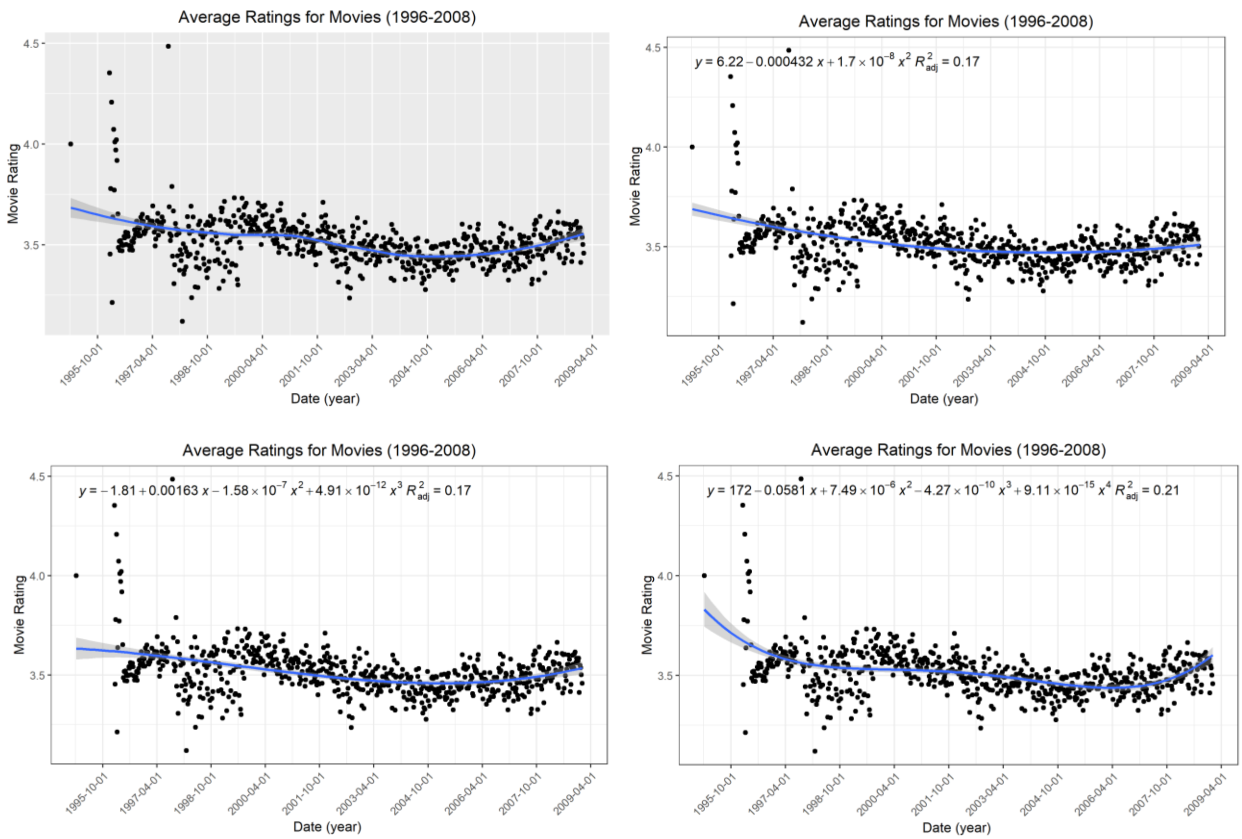


In this machine learning application, we selected 10% of the movielens data to be the validation set. The training set is named “edx” and the test set is named “temp”. Below is a summary for each of the sets:

```
> summary(edx)
     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                                        
 3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000   3rd Qu.:1.127e+09                                        
 Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09                                        
```

```
> summary(validation)
     userId         movieId          rating        timestamp            title              genres         
 Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08   Length:999999      Length:999999     
 1st Qu.:18096   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.467e+08   Class :character   Class :character  
 Median :35768   Median : 1827   Median :4.000   Median :1.035e+09   Mode  :character   Mode  :character  
 Mean   :35870   Mean   : 4108   Mean   :3.512   Mean   :1.033e+09                                        
 3rd Qu.:53621   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09                                        
 Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09                                        
```

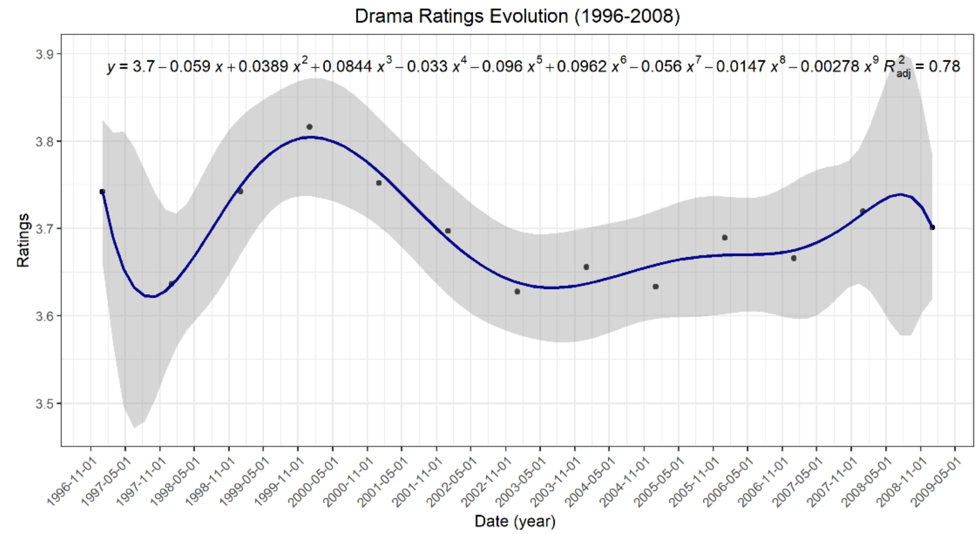
The mean rating for all the movies was found to be $\hat{\mu}$=3.512475 with a RMSE of 1.060959.

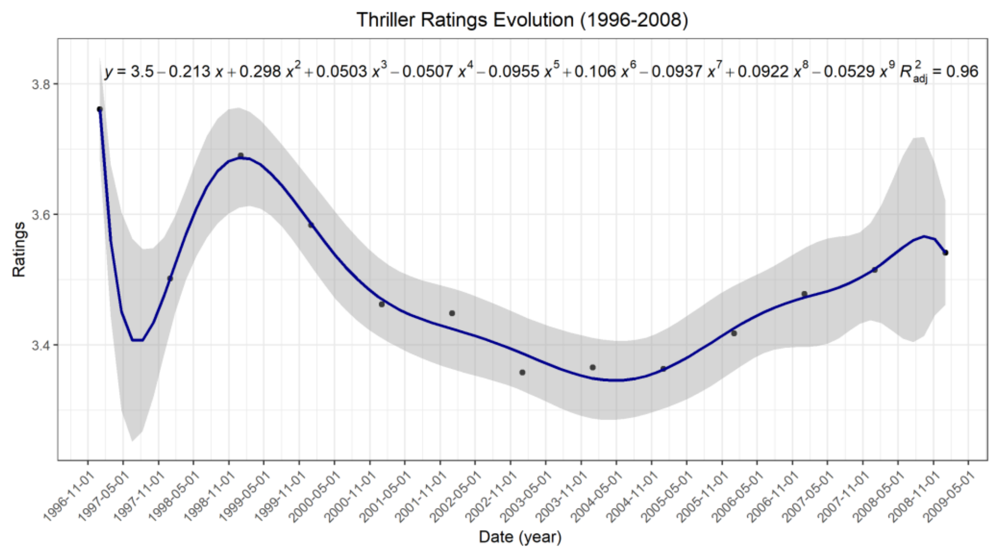


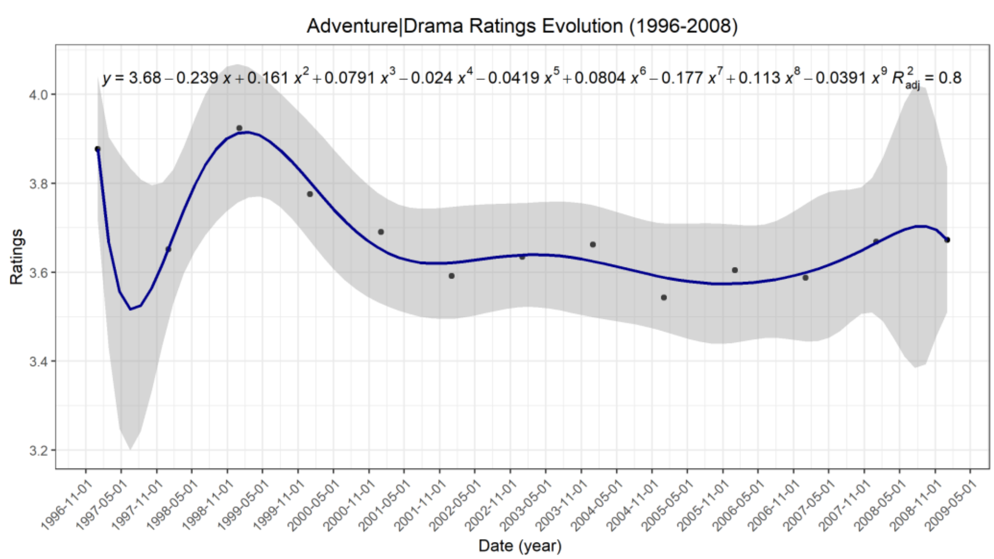
The top left plot was obtained using “geom_smooth()”. The top right plot is for a set polynomial of second order. The bottom left plot was set for a polynomial of third order. The bottom right plot was set for a polynomial of fourth order.

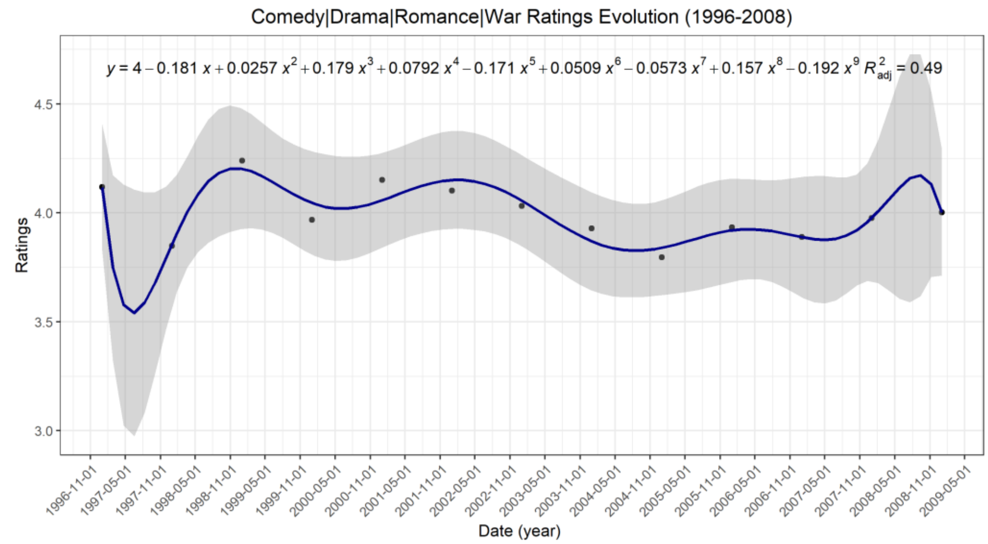
The next series of graphs show the evolution of some movie’s genres with the highest ratings from 1996 to 2008. As an example, we chose a high order polynomial fit (nineth order) for all examples to show consistency across genres. Some observations can be extracted:

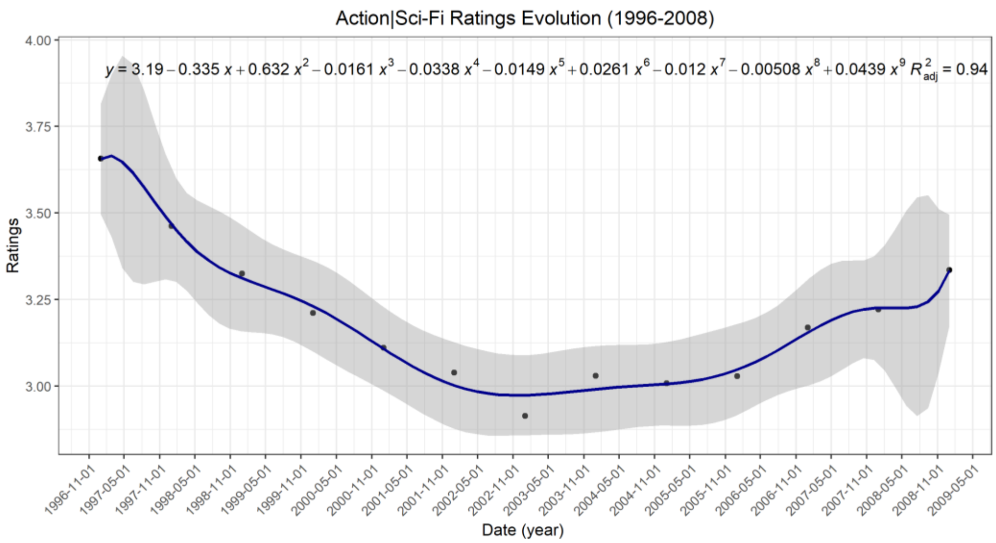
- For all these examples, higher ratings occur during early years.
- Some of these movies’ genres show a notable decreased ratings profile over time, other movies show more variable ratings profiles, also slightly decreasing in time.
- Most genres seemed to have reached a minimum in ratings around 2003-2004 before increasing their respective ratings during the last few years.
- Time units was “years” in these time evolution graphs, but other units (months, weeks, days) can be chose yielding a lower correlation coefficient.
- Different order of polynomial fit can be chosen.

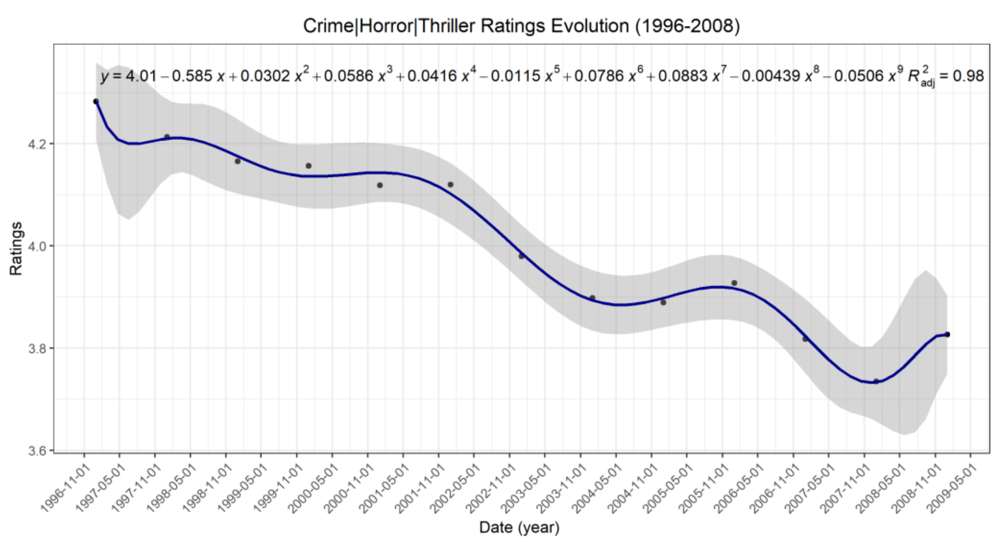

Drama Ratings Evolution (1996-2008)
$y = 3.7 - 0.059\,x + 0.0389\,x^2 + 0.0844\,x^3 - 0.033\,x^4 - 0.096\,x^5 + 0.0962\,x^6 - 0.056\,x^7 - 0.0147\,x^8 - 0.00278\,x^9 \; R^2_{adj} = 0.78$
Ratings
3.9
3.8
3.7
3.6
3.5
Date (year)
1996-11-01 1997-05-01 1997-11-01 1998-05-01 1998-11-01 1999-05-01 1999-11-01 2000-05-01 2000-11-01 2001-05-01 2001-11-01 2002-05-01 2002-11-01 2003-05-01 2003-11-01 2004-05-01 2004-11-01 2005-05-01 2005-11-01 2006-05-01 2006-11-01 2007-05-01 2007-11-01 2008-05-01 2008-11-01 2009-05-01


Thriller Ratings Evolution (1996-2008)
$y = 3.5 - 0.213\,x + 0.298\,x^2 + 0.0503\,x^3 - 0.0507\,x^4 - 0.0955\,x^5 + 0.106\,x^6 - 0.0937\,x^7 + 0.0922\,x^8 - 0.0529\,x^9 \; R^2_{adj} = 0.96$
Ratings
3.8
3.6
3.4
Date (year)
1996-11-01 1997-05-01 1997-11-01 1998-05-01 1998-11-01 1999-05-01 1999-11-01 2000-05-01 2000-11-01 2001-05-01 2001-11-01 2002-05-01 2002-11-01 2003-05-01 2003-11-01 2004-05-01 2004-11-01 2005-05-01 2005-11-01 2006-05-01 2006-11-01 2007-05-01 2007-11-01 2008-05-01 2008-11-01 2009-05-01

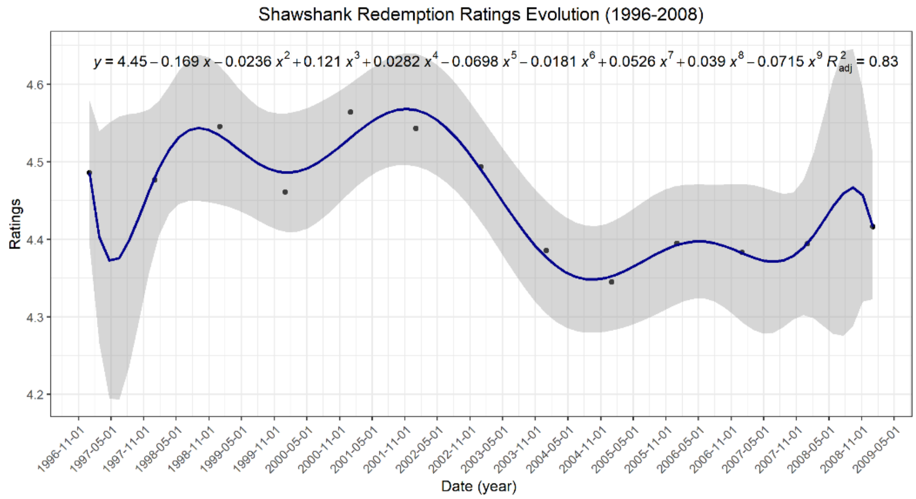

Adventure|Drama Ratings Evolution (1996-2008)
$y = 3.68 - 0.239\,x + 0.161\,x^2 + 0.0791\,x^3 - 0.024\,x^4 - 0.0419\,x^5 + 0.0804\,x^6 - 0.177\,x^7 + 0.113\,x^8 - 0.0391\,x^9 \; R^2_{adj} = 0.8$
Ratings
4.0
3.8
3.6
3.4
3.2
Date (year)
1996-11-01 1997-05-01 1997-11-01 1998-05-01 1998-11-01 1999-05-01 1999-11-01 2000-05-01 2000-11-01 2001-05-01 2001-11-01 2002-05-01 2002-11-01 2003-05-01 2003-11-01 2004-05-01 2004-11-01 2005-05-01 2005-11-01 2006-05-01 2006-11-01 2007-05-01 2007-11-01 2008-05-01 2008-11-01 2009-05-01

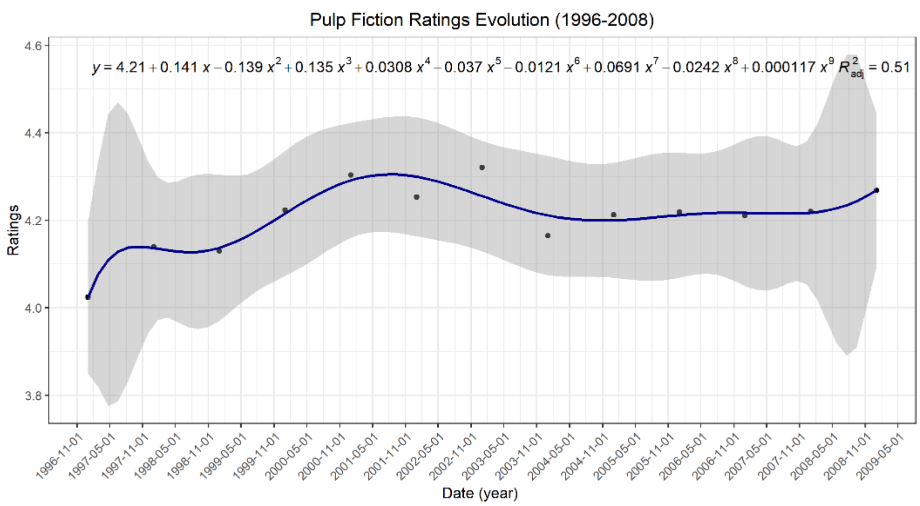

Comedy|Drama|Romance|War Ratings Evolution (1996-2008)
$y = 4 - 0.181\ x + 0.0257\ x^2 + 0.179\ x^3 + 0.0792\ x^4 - 0.171\ x^5 + 0.0509\ x^6 - 0.0573\ x^7 + 0.157\ x^8 - 0.192\ x^9\ R^2_{adj} = 0.49$
Ratings
Date (year)

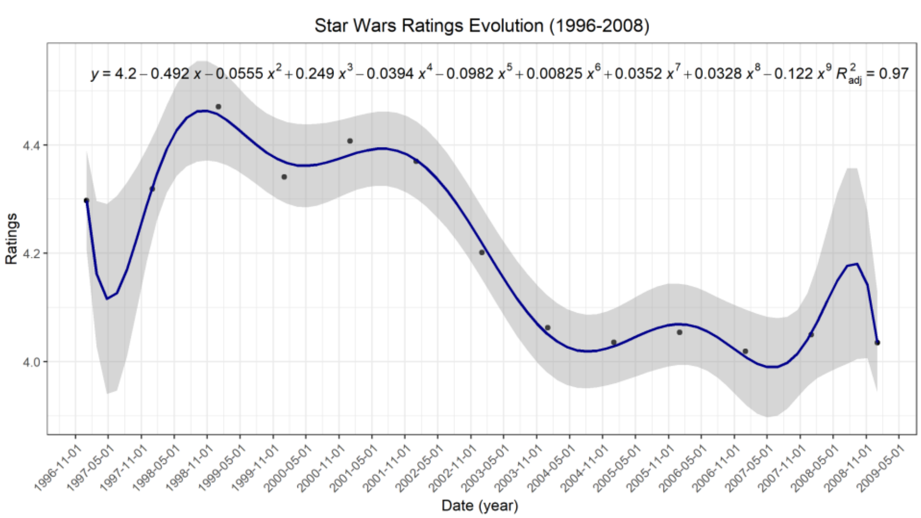

Action|Sci-Fi Ratings Evolution (1996-2008)
$y = 3.19 - 0.335\ x + 0.632\ x^2 - 0.0161\ x^3 - 0.0338\ x^4 - 0.0149\ x^5 + 0.0261\ x^6 - 0.012\ x^7 - 0.00508\ x^8 + 0.0439\ x^9\ R^2_{adj} = 0.94$
Ratings
Date (year)


Crime|Horror|Thriller Ratings Evolution (1996-2008)
$y = 4.01 - 0.585\ x + 0.0302\ x^2 + 0.0586\ x^3 + 0.0416\ x^4 - 0.0115\ x^5 + 0.0786\ x^6 + 0.0883\ x^7 - 0.00439\ x^8 - 0.0506\ x^9\ R^2_{adj} = 0.98$
Ratings
Date (year)

Similarly, we can generate time evolution plots for specific movie titles. As an example, we chose some of the most famous and rated movies of all times: The Shawshank Redemption, Pulp Fiction and Star Wars. Two of these show very similar rating profiles.







After applying some data science and machine learning tools (see code), the algorithm saves the RMSE table as RMSE.csv file, which is the table below:

method	RMSE
Just the average	**1.0612018**
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8653488
Movie + User + Genres Effects Model	0.8649469
Movie + User + Genres + Week Effects Model	0.8648576
Regularized Movie + User Effects Model	0.8648170

Some take away notes:

- We see a decrease of about **11.73%** in the RMSE of “Movie Effect Model” with respect the “Just the average” method.
- The RMSE was decreased by about **7.85%** when using the “Movie + User Effects Model” with respect the “Movie Effect Model”.
- The RMSE was slightly lowered by **0.004%** when using the “Movie + User + Genres Effects Model” with respect to the “Movie + User Effects Model”.
- The RMSE was decreased by **0.0009%** when using the “Movie + User + Genres + Week Effects Model” with respect the “Movie + User + Genres Effects Model”.
- Finally, regularization decreased the RMSE by **0.0004%** with respect to the " Movie + User + Genres + Week Effects Model”.
- Adding genres and timestamp effect in models 3 and 4 add very small gains in terms of model performance and model 5 (regularization) happens to perform better.

Reason for using regularization in the above study:

We first analyzed the top 10 best and 10 worst movies.

We can see that the 10 top movies are only rated between 1 and 4 times only. From the worst 10 movies, six have been rated between 1 and 2 times only, the other four one or two orders of magnitude higher. Very similar statistics can be extracted for the top 20 and worst 20 movies. For this reason, we are going to use regularization to leverage the estimation of the movie rating. We will use lambda parameter, $\lambda = 3$. When using this parameter using regularization, we see that most of the top 10 movies have been rated at least several thousand times or tens of thousands (except one). Similarly, for the worst 10 movies using regularization, about 60% of these have been rated several hundreds of times, the rest several dozen times. Note that we could have used a different $\lambda = 2, 5, 10$ and the RMSE would have been 5E-7 larger only.

Top 10 movies:

title	b_i	n
Hellhounds on My Trail (1999)	1.487534	1
Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)	1.487534	2
Shadows of Forgotten Ancestors (1964)	1.487534	1
Fighting Elegy (Kenka erejii) (1966)	1.487534	1
Sun Alley (Sonnenallee) (1999)	1.487534	1
Blue Light, The (Das Blaue Licht) (1932)	1.487534	1
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	1.237534	4
Human Condition II, The (Ningen no joken II) (1959)	1.237534	4
Human Condition III, The (Ningen no joken III) (1961)	1.237534	4
Constantine's Sword (2007)	1.237534	2

Top worst movies:

title	b_i	n
Besotted (2001)	-3.012466	2
Accused (Anklaget) (2005)	-3.012466	1
Confessions of a Superhero (2007)	-3.012466	1
War of the Worlds 2: The Next Wave (2008)	-3.012466	2
SuperBabies: Baby Geniuses 2 (2004)	-2.717823	56
Hip Hop Witch, Da (2000)	-2.691038	14
Disaster Movie (2008)	-2.653091	32
From Justin to Kelly (2003)	-2.610456	199
Criminals (1996)	-2.512466	2
Mountain Eagle, The (1926)	-2.512466	2

After regularization, the top 10 best movies are:

title	b_i	n
Shawshank Redemption, The (1994)	0.9425642	28015
Godfather, The (1972)	0.9027473	17747
Usual Suspects, The (1995)	0.8532694	21648
Schindler's List (1993)	0.8509172	23193
More (1998)	0.8412738	7
Casablanca (1942)	0.8077420	11232
Rear Window (1954)	0.8058808	7935
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	0.8025895	2922
Third Man, The (1949)	0.7981526	2967
Double Indemnity (1944)	0.7972407	2154

And the worst 10 movies are:

title	b_i	n
SuperBabies: Baby Geniuses 2 (2004)	-2.579629	56
From Justin to Kelly (2003)	-2.571687	199
PokÃ©mon Heroes (2003)	-2.430056	137
Disaster Movie (2008)	-2.425683	32
Carnosaur 3: Primal Species (1996)	-2.321798	68
Glitter (2001)	-2.316450	339
Pokemon 4 Ever (a.k.a. PokÃ©mon 4: The Movie) (2002)	-2.300089	202
Gigli (2003)	-2.297158	313
Barney's Great Adventure (1998)	-2.291910	208
Hip Hop Witch, Da (2000)	-2.216149	14

- **How does the amount of selected data affect the estimated probability for movie rating?**

Assuming a 10% of the data (p=0.1), we computed the estimated probability. The run time is approximately 5 minutes. Note that the polynomial depicted on the plot shows two decimals only so the sixth order is not shown. We keep two decimals for simplicity purposes.

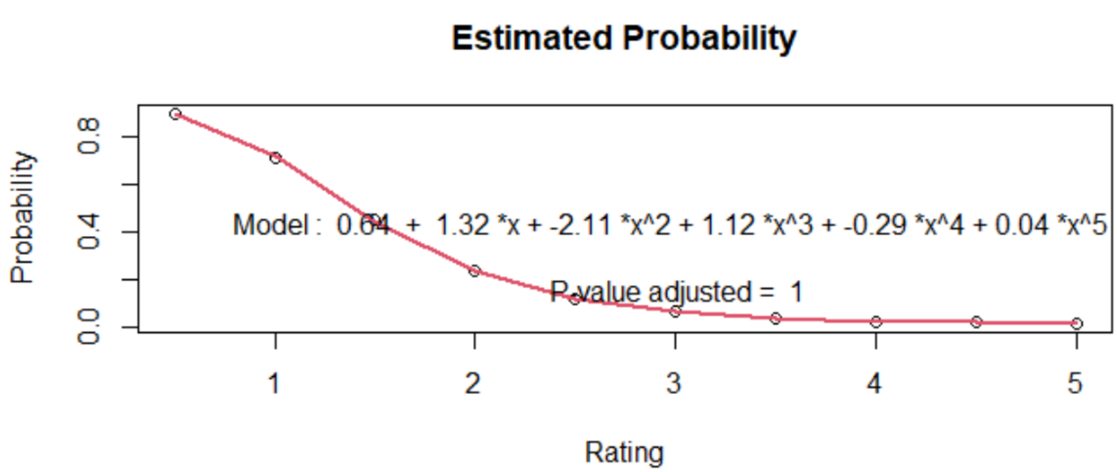
```
> summary(model)

Call:
lm(formula = p_hat_bayes ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +
    I(x^6))

Residuals:
       Min         1Q     Median         3Q        Max
-0.0009200 -0.0002618  0.0001727  0.0002434  0.0050399

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.427e-01  4.630e-05   13881   <2e-16 ***
x            1.318e+00  1.504e-04    8763   <2e-16 ***
I(x^2)      -2.111e+00  1.820e-04  -11596   <2e-16 ***
I(x^3)       1.119e+00  1.070e-04   10456   <2e-16 ***
I(x^4)      -2.869e-01  3.270e-05   -8774   <2e-16 ***
I(x^5)       3.636e-02  4.990e-06    7288   <2e-16 ***
I(x^6)      -1.831e-03  2.998e-07   -6106   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0007022 on 1000000 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 9.274e+09 on 6 and 1e+06 DF,  p-value: < 2.2e-16
```



- Assuming a 20% of the data (p=0.2), we computed the estimated probability. The run time is about 7 minutes. Once again, here the polynomial fit shows only two digits for simplicity.

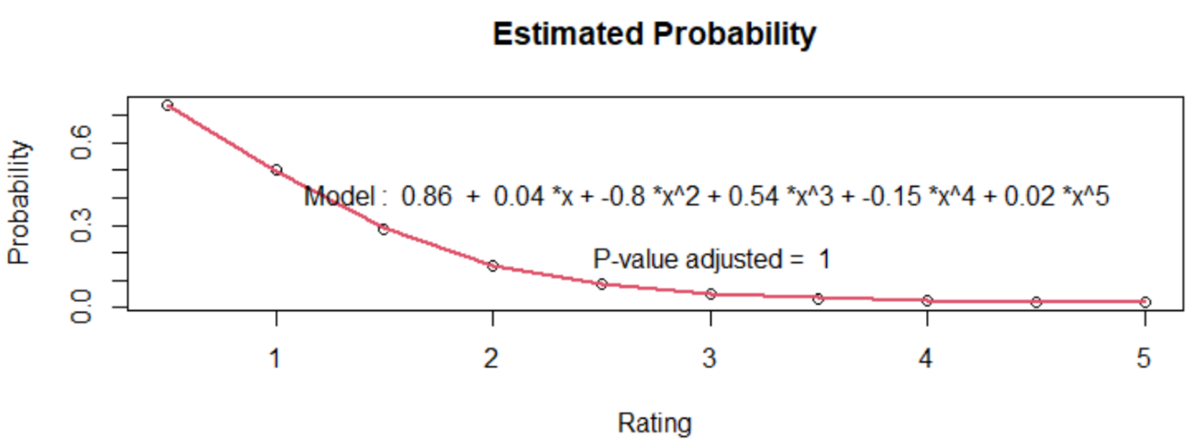
```
> summary(model)

Call:
lm(formula = p_hat_bayes ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +
    I(x^6))

Residuals:
       Min         1Q     Median         3Q        Max
-0.0033837 -0.0001070  0.0000570  0.0003471  0.0018105

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.632e-01  3.039e-05 28407.9   <2e-16 ***
x            3.569e-02  9.866e-05   361.8   <2e-16 ***
I(x^2)      -8.018e-01  1.194e-04 -6714.6   <2e-16 ***
I(x^3)       5.368e-01  7.015e-05  7652.6   <2e-16 ***
I(x^4)      -1.547e-01  2.144e-05 -7214.8   <2e-16 ***
I(x^5)       2.124e-02  3.271e-06  6491.4   <2e-16 ***
I(x^6)      -1.137e-03  1.966e-07 -5786.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0006513 on 2000005 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.084e+10 on 6 and 2000005 DF,  p-value: < 2.2e-16
```



- Assuming a 30% of the data (p=0.3). The run time is about 7 minutes.

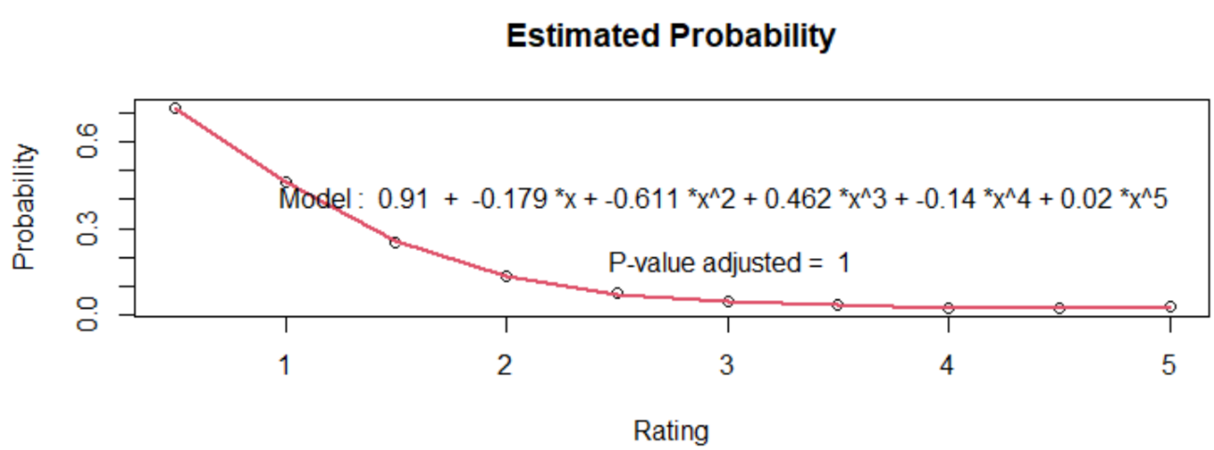
```
> summary(model)

Call:
lm(formula = p_hat_bayes ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +
    I(x^6))

Residuals:
       Min         1Q     Median         3Q        Max
-0.0047275 -0.0001608  0.0000664  0.0003900  0.0020883

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.097e-01  3.053e-05   29797   <2e-16 ***
x           -1.791e-01  9.906e-05   -1808   <2e-16 ***
I(x^2)      -6.105e-01  1.198e-04   -5094   <2e-16 ***
I(x^3)       4.619e-01  7.039e-05    6563   <2e-16 ***
I(x^4)      -1.397e-01  2.151e-05   -6494   <2e-16 ***
I(x^5)       1.973e-02  3.281e-06    6013   <2e-16 ***
I(x^6)      -1.078e-03  1.972e-07   -5465   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0008002 on 3000011 degrees of freedom
Multiple R-squared:  0.9999,    Adjusted R-squared:  0.9999
F-statistic: 9.192e+09 on 6 and 3000011 DF,  p-value: < 2.2e-16
```



- Assuming a 40% of the data (p=0.4). The run time is about 10 minutes.

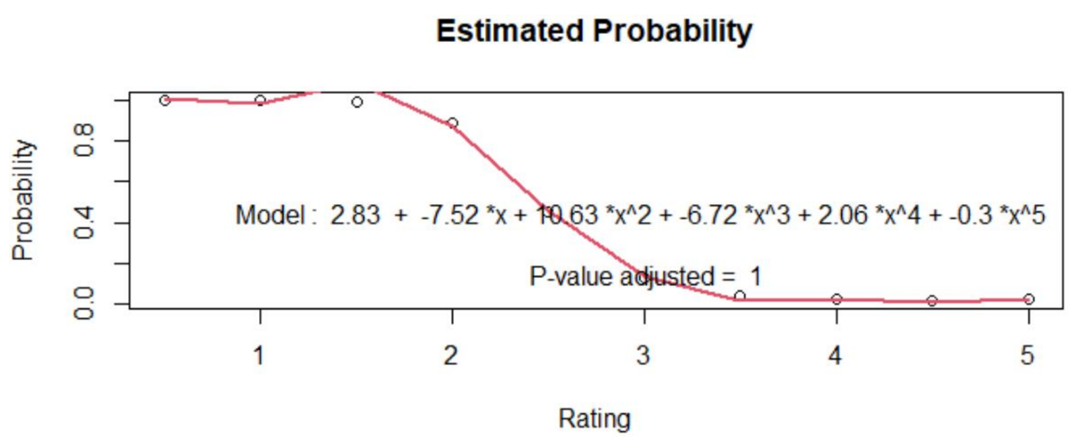
```
> summary(model)

Call:
lm(formula = p_hat_bayes ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +
    I(x^6))

Residuals:
      Min        1Q    Median        3Q       Max
-0.099729 -0.006815 -0.005620  0.009469  0.027297

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.829e+00  5.181e-04    5461   <2e-16 ***
x           -7.517e+00  1.681e-03   -4472   <2e-16 ***
I(x^2)       1.063e+01  2.033e-03    5225   <2e-16 ***
I(x^3)      -6.722e+00  1.194e-03   -5628   <2e-16 ***
I(x^4)       2.058e+00  3.649e-04    5640   <2e-16 ***
I(x^5)      -3.026e-01  5.568e-05   -5434   <2e-16 ***
I(x^6)       1.719e-02  3.346e-06    5138   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01568 on 4000016 degrees of freedom
Multiple R-squared:  0.9975,    Adjusted R-squared:  0.9975
F-statistic: 2.631e+08 on 6 and 4000016 DF,  p-value: < 2.2e-16
```



IV. CONCLUSIONS and FUTURE WORK

Machine learning has proved to be a good promising technique to enhance real-world problems, and in this proposed study, it has been proved to be a great tool to enhance our understanding of the behavior of very large data sets, such as the movielens. In this preliminary study, we have seen that regularization method based on the least square method with penalty terms is a great technique to further reduce the uncertainty of the movie rating by decreasing the RMSE. In this preliminary study, we achieved a RMSE of 0.8648170 using the regularization for the movie and user effects model, which is about 19.5% lower than the average method.

However, other methods such as the matrix factorization, single value decomposition and principal component analysis should be explored to consider tackling this and other problems and further refine the performance of our algorithm.

Time evolution of movie titles and genres can be very useful to predict movie ratings. Although not pursued in detail in this report, ratings for movie genres and movie titles can provide insightful information when analyzing their trends during specific timeframes. It would be suggested to do smaller time frames and perhaps various models for specific polynomial fit to enhance our understanding on how these ratings are given.

Further optimization of the algorithm can be done as future work.

V. REFERENCES

[1] Introduction to Data Science: Data Analysis and Prediction Algorithms with R. Front Cover. Rafael A. Irizarry. CRC Press, Nov 20, 2019.

[2] Social Computing Research at the University of Minnesota, https://grouplens.org/datasets/movielens/10m/

VI. APPENDIX (Code)

The R-algorithm used to generate this report and data can be found in the GitHub repository under https://github.com/PedroLlanos