Develop Movie Recommendation Model Using the MovieLens Dataset - Professional Certificate in Data Science by HarvardX Capstone Project

Velko Kamenov

August 26, 2020

1. Introduction

The aim of this report is to examine the dataset with movies ratings MovieLens and build a recommendation system model based on this dataset. The goals is to find the best model for predicting movie ratings based on the inputs found in the MovieLens dataset and to be able to predict movie ratings with Root Mean Squared Error (RMSE) lower than 0.86490.

In order to achieve this goal the dataset was examined statistically and visually, feature engineering was performed and different linear regression models were built on the modelling sample and tested on the validation sample.

A model with 4 predictor variables satisfied the goal to achieve RMSE bellow 0.86490 on the validation set.

The following 3 sections present the analysis and the final results of the model as well as suggestions for future model improvements.

2. Analysis

In this section of the report are presented the data exploration, data preprocessing, feature engineering, feature relationships analysis as well as the modelling techniques used to generate the final predictive model.

2.1. Initial Data Exploration

Two datasets are provided as starting point for the project. A modelling dataset on which to perform model training and parameter tuning. This dataset consists of 9000055 observations. And a hold-out validation set consisting of 999999 observations. The results of the algorithm and its final predictive power are going to be tested on the validation set.

The following table shows an overview of the 6 variables found in the training set. There are no missing, zero or Inf values among all variables. We have 4 numeric and 2 character type columns.

Table 1: Train Set Variables Summary

variable	q_zeros	p_zeros	q_na	p_na	q_inf	p_inf	type	unique
userId	0	0	0	0	0	0	integer	69878
movieId	0	0	0	0	0	0	numeric	10677
rating	0	0	0	0	0	0	numeric	10
timestamp	0	0	0	0	0	0	integer	6519590
title	0	0	0	0	0	0	character	10676
genres	0	0	0	0	0	0	character	797

And a sample of 10 rows in the Train Set to get better sense of the data:

Table 2: Sample of 10 rows in the Train Set

userId	movieId	rating	timestamp	title	genres
3149	65133	2.0	1231075097	Blackadder Back & Forth (1999)	Comedy
19957	65133	5.0	1231081397	Blackadder Back & Forth (1999)	Comedy
26920	65130	2.5	1231061731	Revolutionary Road (2008)	Drama|Romance
33384	65133	3.0	1231034528	Blackadder Back & Forth (1999)	Comedy
40570	65133	2.0	1231055397	Blackadder Back & Forth (1999)	Comedy
45430	65133	2.5	1231105425	Blackadder Back & Forth (1999)	Comedy
49138	65130	2.0	1231093935	Revolutionary Road (2008)	Drama|Romance
52648	65126	3.0	1231028759	Choke (2008)	Comedy|Drama
63100	65126	4.5	1231130453	Choke (2008)	Comedy|Drama
64621	65126	3.5	1231097168	Choke (2008)	Comedy|Drama
68151	65133	5.0	1231129793	Blackadder Back & Forth (1999)	Comedy

The target variable we aim to predict with this dataset is rating which can take 10 unique values ranging from 0.5 to 5: 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5.

2.2. Feature engineering

We see that in the "title" column is stored the name of the movie as well as the year in which the movie came out. This information should come in two separate columns. We see that the year always comes after the movie name so it is easy just to extract the years as the symbols from the 2nd character going backwards to the 5th character going backwards. We also remove the years from the "title" column since the year is already extracted in a separate column.

Moreover we see that in the genres column for most movies there is more than one genre. It may be useful to extract only the leading genre - which can be found before the first "|" symbol. It is extracted in the column leading_genre.

Here is how the data looks like after these operations:

Table 3: Sample of 10 rows in the Train Set after year is extracted from the title column

userId	movieId	rating	timestamp	title	genres	release_year	leading_genre
3149	65133	2.0	1231075097	Blackadder Back & Forth	Comedy	1999	Comedy
19957	65133	5.0	1231081397	Blackadder Back & Forth	Comedy	1999	Comedy
26920	65130	2.5	1231061731	Revolutionary Road	Drama|Romance	2008	Drama
33384	65133	3.0	1231034528	Blackadder Back & Forth	Comedy	1999	Comedy
40570	65133	2.0	1231055397	Blackadder Back & Forth	Comedy	1999	Comedy
45430	65133	2.5	1231105425	Blackadder Back & Forth	Comedy	1999	Comedy
49138	65130	2.0	1231093935	Revolutionary Road	Drama|Romance	2008	Drama
52648	65126	3.0	1231028759	Choke	Comedy|Drama	2008	Comedy
63100	65126	4.5	1231130453	Choke	Comedy|Drama	2008	Comedy
64621	65126	3.5	1231097168	Choke	Comedy|Drama	2008	Comedy
68151	65133	5.0	1231129793	Blackadder Back & Forth	Comedy	1999	Comedy

We can also observe that the information is the columns movieId and title is the same. The movieId is just a numerical nomenclature of the respective movie title. This is also suggested by the fact that the movieId column has 10677 distinct values while the title column has 10676 distinct values. So we are going to exclude

the title column from the rest of the analysis since we extracted all relevant information from it and it does not give any additional information compared to movieId.

We also see that the timestamp column is formatted in numeric format rather than dates format. We can fix this with the as_datetime() function.

Here is how the data looks after the last transformations:

Table 4: Sample of 10 rows in the Train Set after title is removed and timestamp is converted to date

userId	movieId	rating	timestamp	genres	release_year	leading_genre
3149	65133	2.0	2009-01-04 13:18:17	Comedy	1999	Comedy
19957	65133	5.0	2009-01-04 15:03:17	Comedy	1999	Comedy
26920	65130	2.5	2009-01-04 09:35:31	Drama|Romance	2008	Drama
33384	65133	3.0	2009-01-04 02:02:08	Comedy	1999	Comedy
40570	65133	2.0	2009-01-04 07:49:57	Comedy	1999	Comedy
45430	65133	2.5	2009-01-04 21:43:45	Comedy	1999	Comedy
49138	65130	2.0	2009-01-04 18:32:15	Drama|Romance	2008	Drama
52648	65126	3.0	2009-01-04 00:25:59	Comedy|Drama	2008	Comedy
63100	65126	4.5	2009-01-05 04:40:53	Comedy|Drama	2008	Comedy
64621	65126	3.5	2009-01-04 19:26:08	Comedy|Drama	2008	Comedy
68151	65133	5.0	2009-01-05 04:29:53	Comedy	1999	Comedy

We see that the first rating in the dataset was given on 1995-01-09 and the last rating was given on 2009-01-05.

Knowing this we can create a new variable which may be useful for the further analysis - years_after_release (YAR) which is going to me calculated by subtracting the movie release year from the last year a rating was given for the whole dataset. In this way we can get the variable years_after_release as of the point in time of the data in the MovieLens dataset.

We create also two more variables - Year in which the rating was given (YRG) and Years after the movie release before the rating was given (YRG_AR).

Another variable - number of times a movie is rated (NR) can also be interesting and is extracted.

Table 5: Sample of 10 rows in the Train Set after years since release variable is created

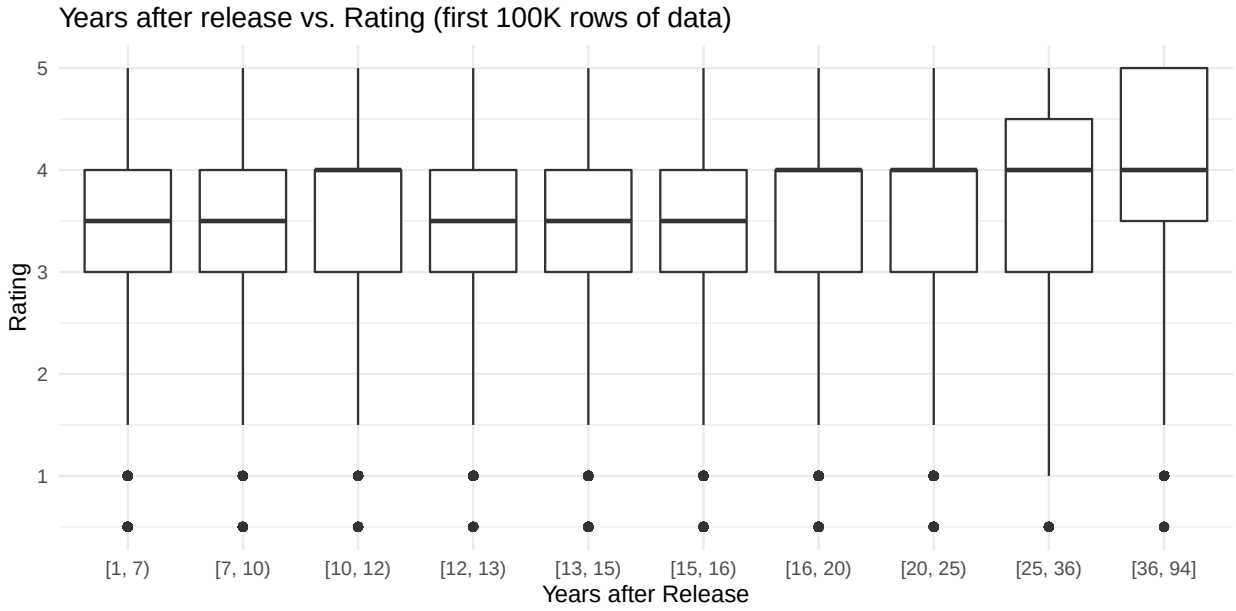
userId	movieId	rating	timestamp	release_year	leading_genre	YRG	YAR	YRG_AR	NR
3149	65133	2.0	2009-01-04	1999	Comedy	2009	10	10	6
19957	65133	5.0	2009-01-04	1999	Comedy	2009	10	10	6
26920	65130	2.5	2009-01-04	2008	Drama	2009	1	1	2
33384	65133	3.0	2009-01-04	1999	Comedy	2009	10	10	6
40570	65133	2.0	2009-01-04	1999	Comedy	2009	10	10	6
45430	65133	2.5	2009-01-04	1999	Comedy	2009	10	10	6
49138	65130	2.0	2009-01-04	2008	Drama	2009	1	1	2
52648	65126	3.0	2009-01-04	2008	Comedy	2009	1	1	3
63100	65126	4.5	2009-01-05	2008	Comedy	2009	1	1	3
64621	65126	3.5	2009-01-04	2008	Comedy	2009	1	1	3
68151	65133	5.0	2009-01-05	1999	Comedy	2009	10	10	6

Because the distinct values a rating can be are only 10 a scatter plot is not the best option to visualize relationships among the numeric variables and the rating. So in order to be able to visualize the data through

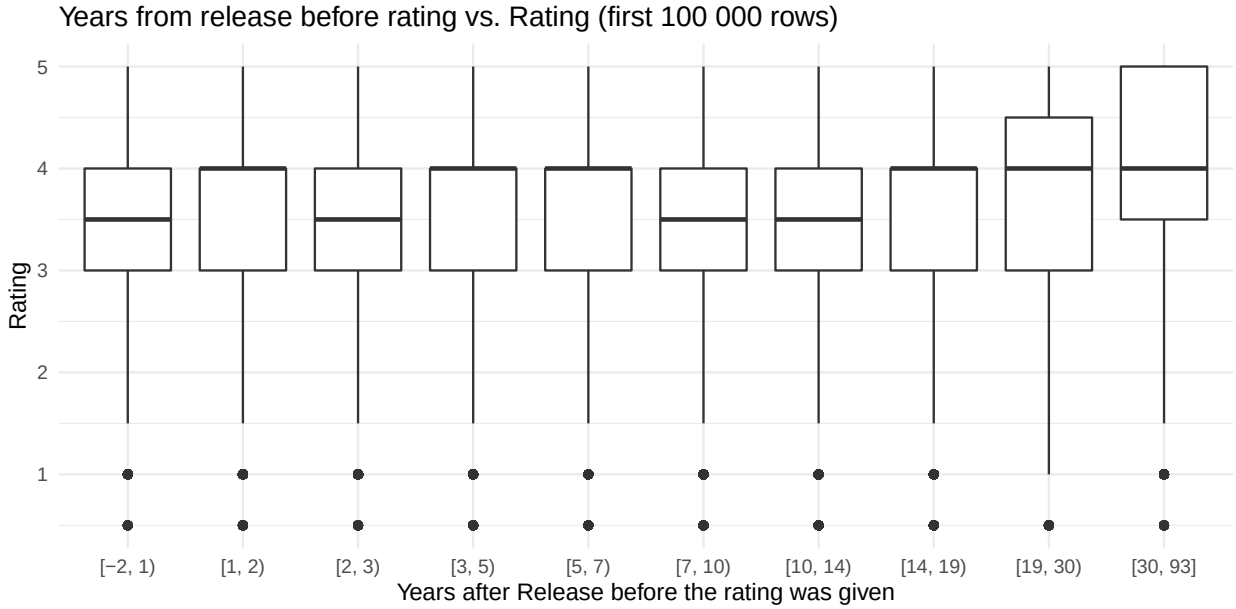
boxplots the numeric features are going to be binned and boxplots created for the binned numeric varialbes vs. the rating.

2.3. Features Relationship to Target

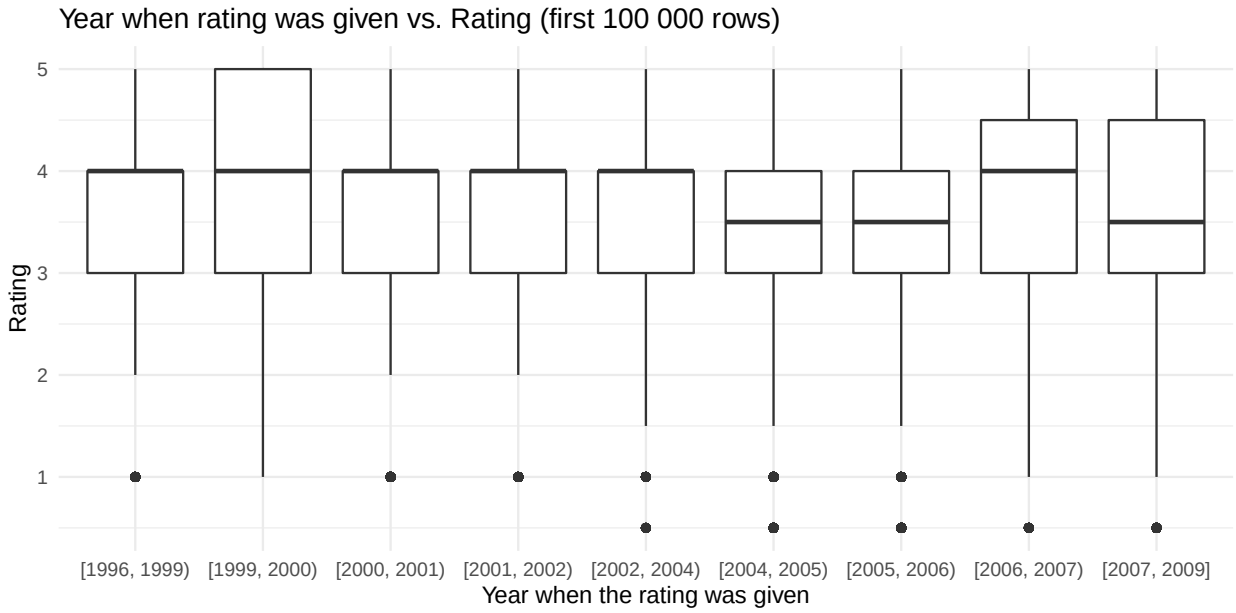
As a next step of the analysis we examine the relationships among the predictor variables and the target variable via boxplots.



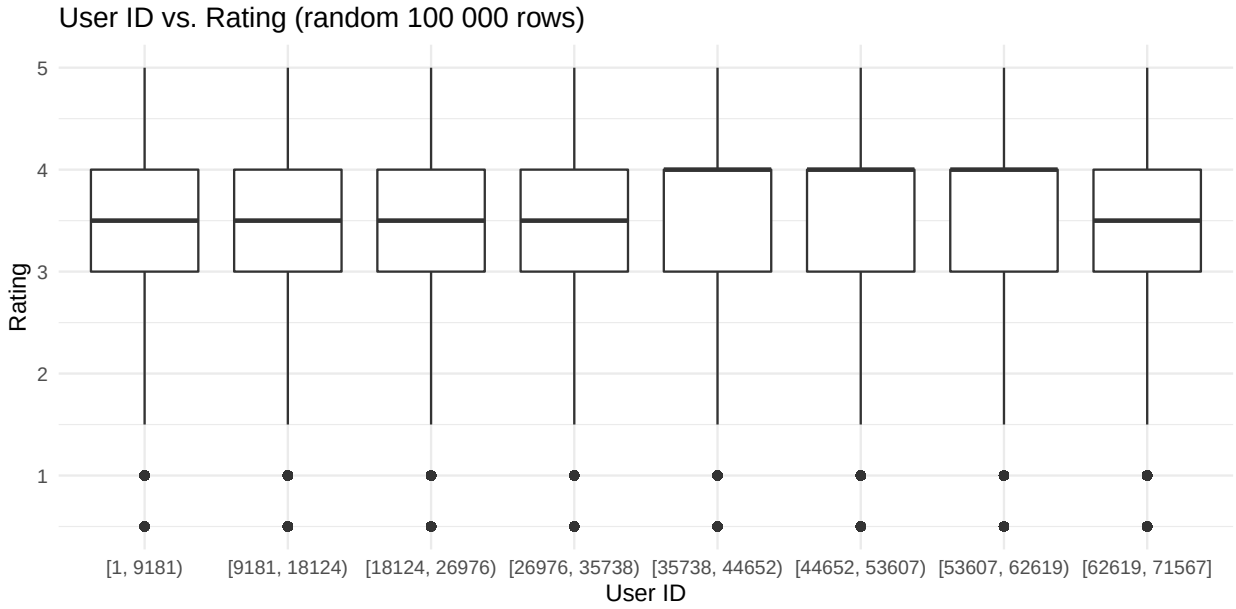
We see that generally movies released before more than 16 years get higher average ratings. The older the movie the higher the rating it receives on average.



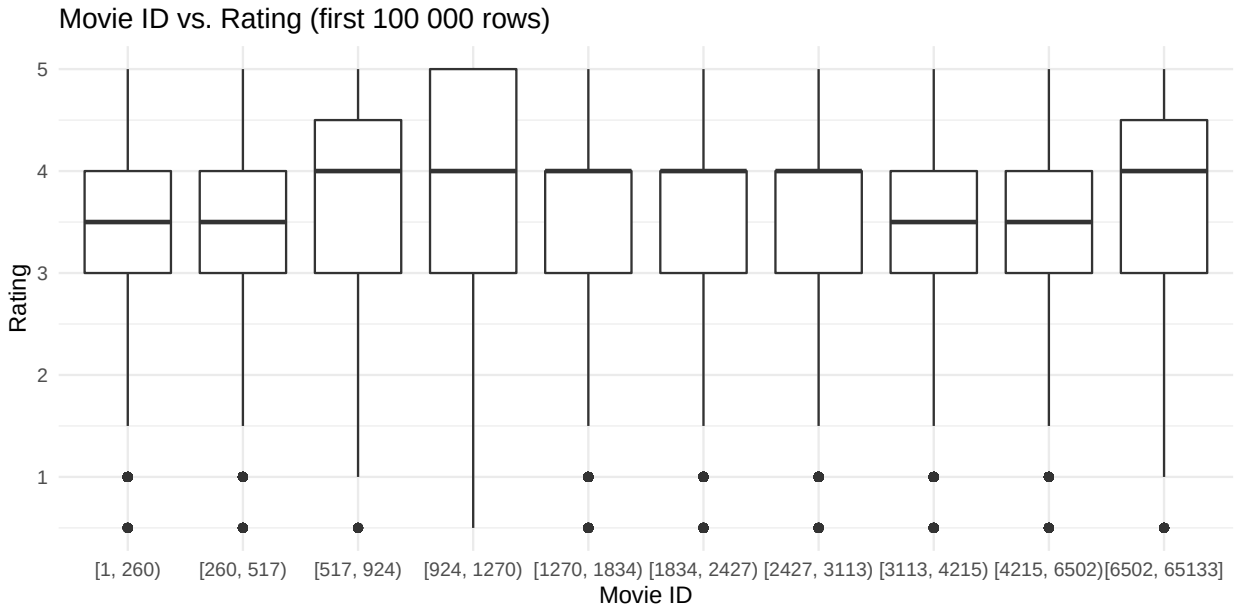
We see that the more time passes between movie release year and the rating moment - the higher the rating the movie receives on average.



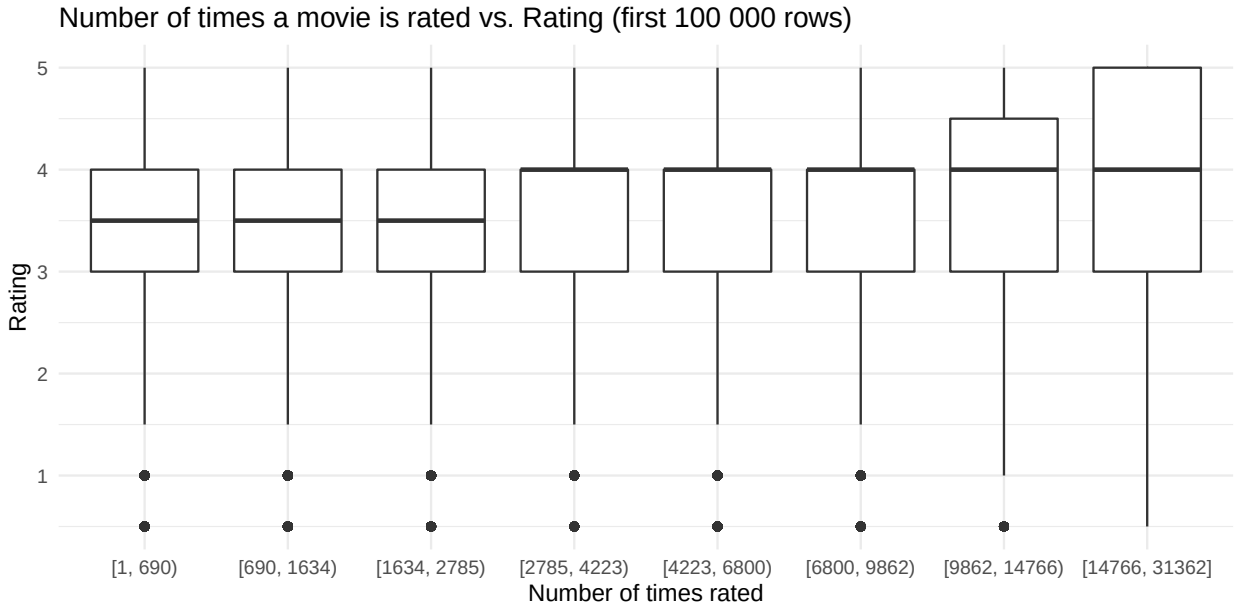
We see that movies rated before 2004 on average receive higher ratings.



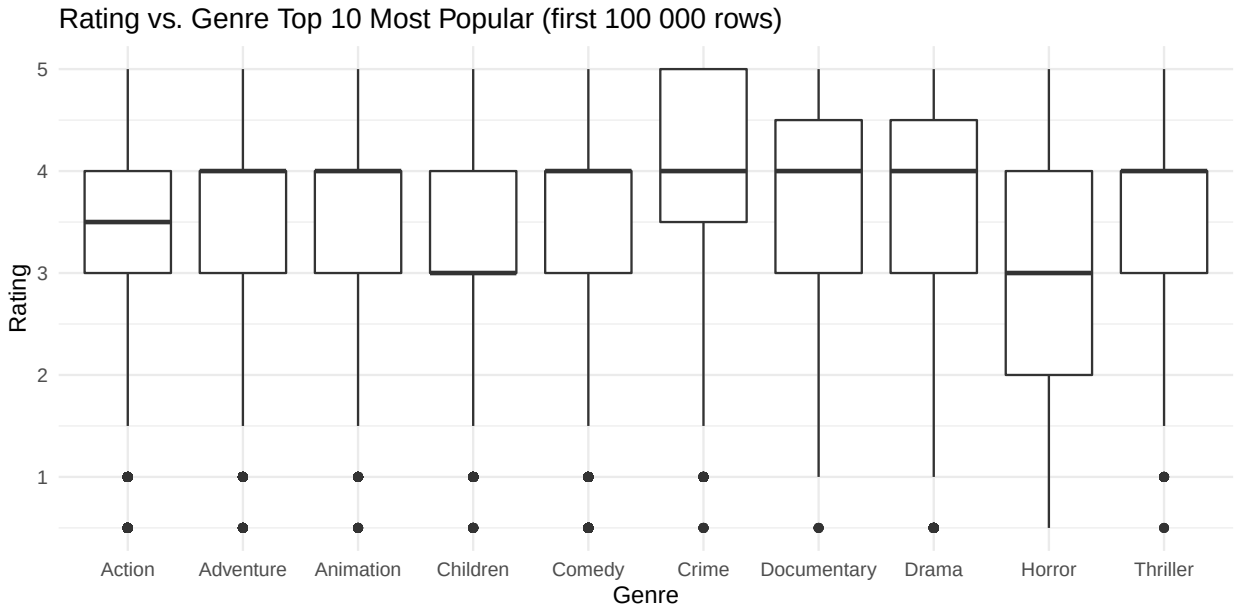
The boxplot shows no strong visible relationship between userId and rating.

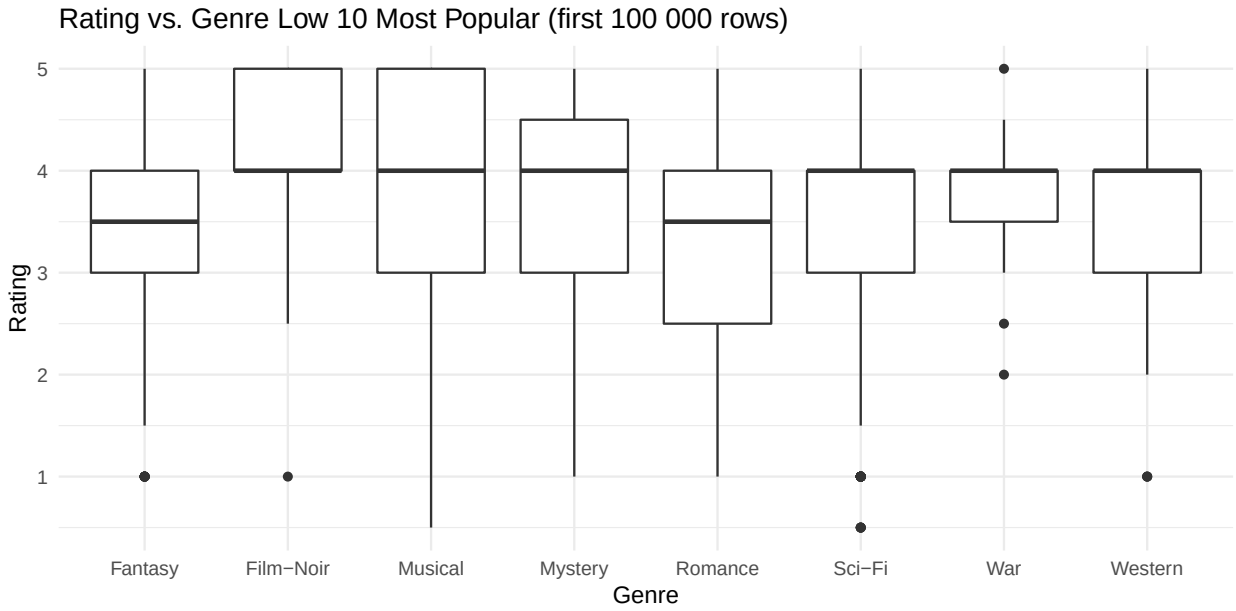


The boxplot shows no strong visible relationship between movieId and rating.



We see that generally movies rated more times get higher average ratings.





We see that some genres like Crime, Documentary, Drama and Musical get higher average ratings than others.

2.4. Correlations

A linear regression modelling technique is going to be used to create a prediction model to forecast the rating. For this model predictor variables have to be uncorrelated. For this reason a correlation matrix to check if this condition holds true is calculated and examined.

Table 6: Pearson correlation coefficient among numeric variables

variable	userId	movieId	release_year	YRG	YAR	YRG_AR	NR
userId	1.0000	0.0049	-0.0001	0.0173	0.0001	0.0047	0.0015
movieId	0.0049	1.0000	0.2400	0.3709	-0.2400	-0.1369	-0.2607
release_year	-0.0001	0.2400	1.0000	0.0888	-1.0000	-0.9633	0.0625

variable	userId	movieId	release_year	YRG	YAR	YRG_AR	NR
YRG	0.0173	0.3709	0.0888	1.0000	-0.0888	0.1820	-0.1790
YAR	0.0001	-0.2400	-1.0000	-0.0888	1.0000	0.9633	-0.0625
YRG_AR	0.0047	-0.1369	-0.9633	0.1820	0.9633	1.0000	-0.1100
NR	0.0015	-0.2607	0.0625	-0.1790	-0.0625	-0.1100	1.0000

We see strong correlations among 3 variables - Release Year and Years Rating Given After Release (-0.96) and Years Rating Given After Release and Years after Release (0.96). This should be kept in mind and only one of these 3 variables should be used in a linear regression model.

Table 7: Spearman correlation coefficient among numeric and categorical variables (top 100 000 rows of data)

variable	userId	movieId	release_year	YRG	YAR	YRG_AR	NR	leading_genre
userId	1.0000	-0.0138	0.0260	-0.0057	-0.0260	-0.0339	0.0231	-0.0093
movieId	-0.0138	1.0000	0.2979	0.5176	-0.2979	-0.0174	-0.5099	0.0190
release_year	0.0260	0.2979	1.0000	0.2256	-1.0000	-0.8345	-0.0687	-0.0373
YRG	-0.0057	0.5176	0.2256	1.0000	-0.2256	0.2678	-0.1850	-0.0257
YAR	-0.0260	-0.2979	-1.0000	-0.2256	1.0000	0.8345	0.0687	0.0373
YRG_AR	-0.0339	-0.0174	-0.8345	0.2678	0.8345	1.0000	-0.0170	0.0326
NR	0.0231	-0.5099	-0.0687	-0.1850	0.0687	-0.0170	1.0000	-0.2494
leading_genre	-0.0093	0.0190	-0.0373	-0.0257	0.0373	0.0326	-0.2494	1.0000

Because the variable leading_genre is not numeric but categorical we need to calculate for it the spearman correlation coefficient with other variables. It shows no strong correlation between leading_genre and any other variable. The only high correlations remain between Release Year and Years Rating Given After Release and Years Rating Given After Release and Years after Release.

2.5. Modelling

The modelling technique used the build the movie recommendation system is linear regression. Its performance is compared to a naive approach of predicting just the average rating. Ireratively one variable at a time is added to the linear regression model and the resulting Mean Squared Errors are compared.

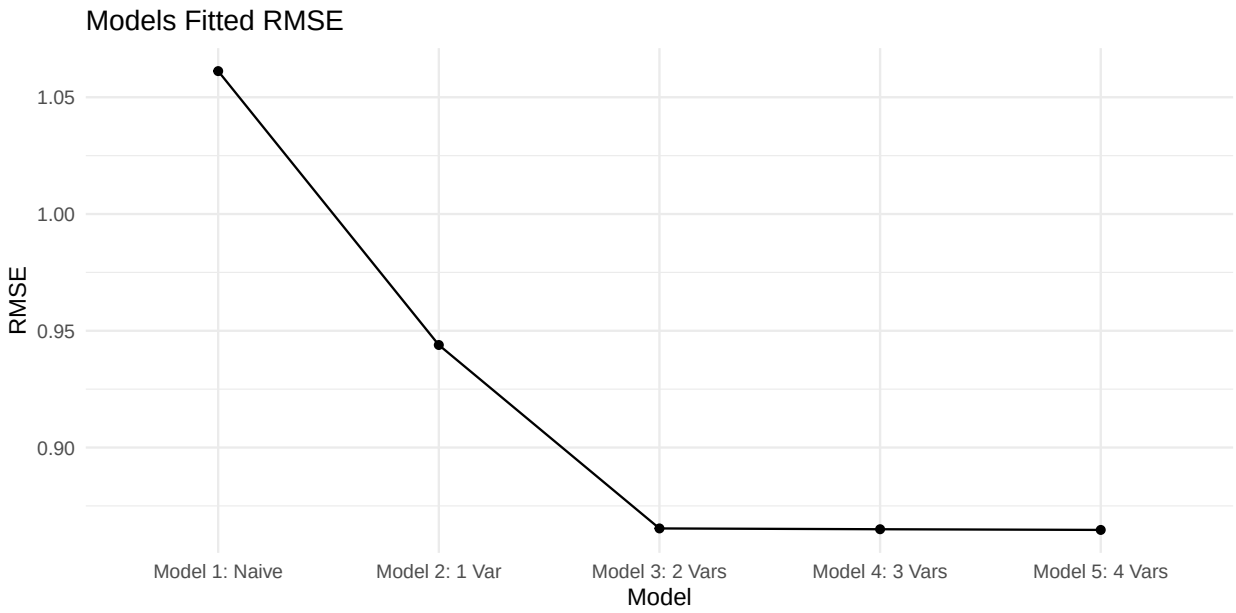
The results of the 4 different models tested are presented in the following table:

Table 8: Linear Regression Models RMSE Comparison

Model	RMSE
Simple average	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8653488
Movie + User + Year after Release Effects Model	0.8650043
Movie + User + Year after Release + Genre Effects Model	0.8647135

The model with the lowest RMSE is the last one which includes as predictor variables 4 predictors - movieId, userId, Years after Release and Genre.

As we see from the graph the first two predictor variables - movieId and userId lead to the biggest drop in RMSE while the next two variables - Years after release and Genre lead to slight improvements in RMSE.



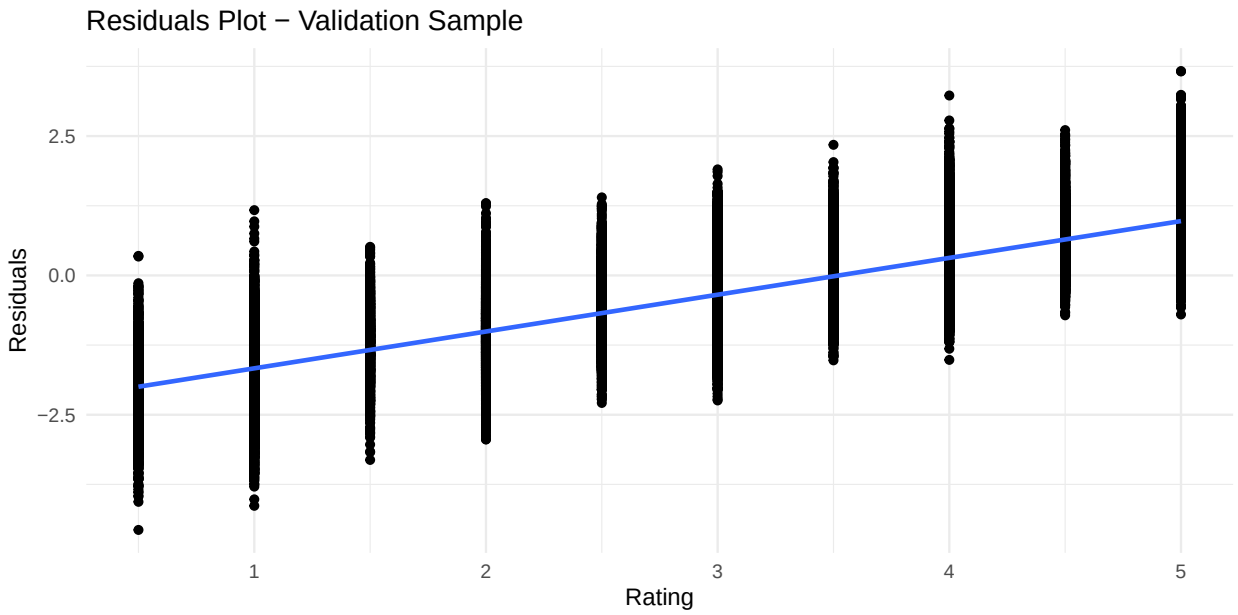
3. Results

This section presents the final results of the analysis and modelling. The best model tested on the validation set was with 4 predictor variables - movieId, userId, Years after Release and Genre. It resulted in RMSE of 0.8647135. This RMSE is bellow 0.86490 which satisfies the project aim.

Table 9: Linear Regression Model with lowest RMSE

Model	RMSE
Movie + User + Year after Release + Genre Effects Model	0.8647135

The residuals plot suggests that lower observed ratings are predicted too pesimistic, while good ratings are predicted too optimistic.



The values of R-Squared(0.336) and Adjusted R-Squared(0.336) also suggest that a lot of the variation is not captured by the model. This means that there are variables influecing the rating that are not present in the dataset.

4. Conclusion

The MovieLens dataset used for the analysis in this paper includes one target variable - rating - and 4 predictor variables. These predictor variables were analysed statistically and visually and also feature engineering to create more variables was performed.

A linear regression model was developed to make forecasts about the rating and a model with 4 predictor variables - movieId, userId, Years after Release and Genre led to the lowest RMSE. The RMSE of this model is 0.8647135 which satisfies the project aim of building a model with RMSE bellow 0.86490.

However the residuals plot and R-Squared estimate (0.336) suggest that the model misses important information which is not found in the dataset and would be useful for further model enhancement.