Film Recommendation System based on a Ridge Regression Model

Israel Girón-Palacios

2023-12-05

Abstract

This paper covers the process by which a quasi-stepwise ridge regression machine learning model was developed to perform film recommendations based on rating prediction trained utilizing MovieLens' 10 Million observations dataset.

Introduction

Recommender systems; machine learning models which provide suggestions of particular product or service, are a cornerstone of media platforms and retail. Among the most engaged examples of these being film recommendations. This type of recommender systems have a long history of machine learning competitions, the most prominent being "The Netflix Prize" which was won by the Bellkor's Pragmatic Chaos team on July 26, 2009. The challenge consisted of developing a machine learning model that predicted user rating for films. This paper will cover the development of a model trained using the 10M MovieLens Dataset[1].

MovieLens

MovieLens[2] is a non-comercial site film recommendation site operated and maintained by GroupLens Research at the University of Minnesota.

Methodology

The recommendation system will be developed utilizing using the "MovieLens 10M Dataset". MovieLens is a non-commercial film review site run by GroupLens Research at the University of Minnesota and it maintains publicly available datasets from their review site. A model will be trained using quasi-stepwise ridge regression based on the large data size and given the model's effects in minimizing multicollinearity of predictors, sequential application of analytically solutions will be applied. Numerical Data will be centered, scaled and transformed as required, however the response variable will be prepared in such a manner that allows for the reverting the preparation allowing it to return to an interpretable response.

Model Output and Loss Function

The model will be based on predicted rating and will aim to minimize Root Mean Square Error (RMSE) of the predicted rating, the response variable and model outputs will necessarily be numeric as RMSE requires numeric values. The formula for RMSE is given as:

[1] https://grouplens.org/datasets/movielens/10m/
[2] https://grouplens.org/datasets/movielens/10m/

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

Where $\hat{y}_i$ is the predicted value y_i is the observed value and n is the number of observations.

Source Documentation

As described in the source documentation for the dataset, the data consists of the following 6 variables:

Variables	Variable Class	Unique Observartions	Description
userId	integer	69,878	Randomized User ID
movieId	integer	10,677	Film ID
rating	numeric	10	User Rating, 5-star scale, with half-star increments
timestamp	integer	6,519,590	Timestamp (UTC)
title	character	10,676	Film Title & Year of Release (Manual Entry)
genres	character	797	Film Genre (Pipe separated list)

Data Partition

Model training and testing will consist of a training set, test set and a final holdout set in a 70/20/10 split. The training and test set will be used in the development and selection of the final model to be used for predicting the holdout set.

Data Preparation

The data from the source documentation will be cleaned and extractable features will be extracted. This section will also cover feature engineering tasks stemming from data explorations.

Data Cleaning

The original dataset is cleaned and base features extracted to the following form:

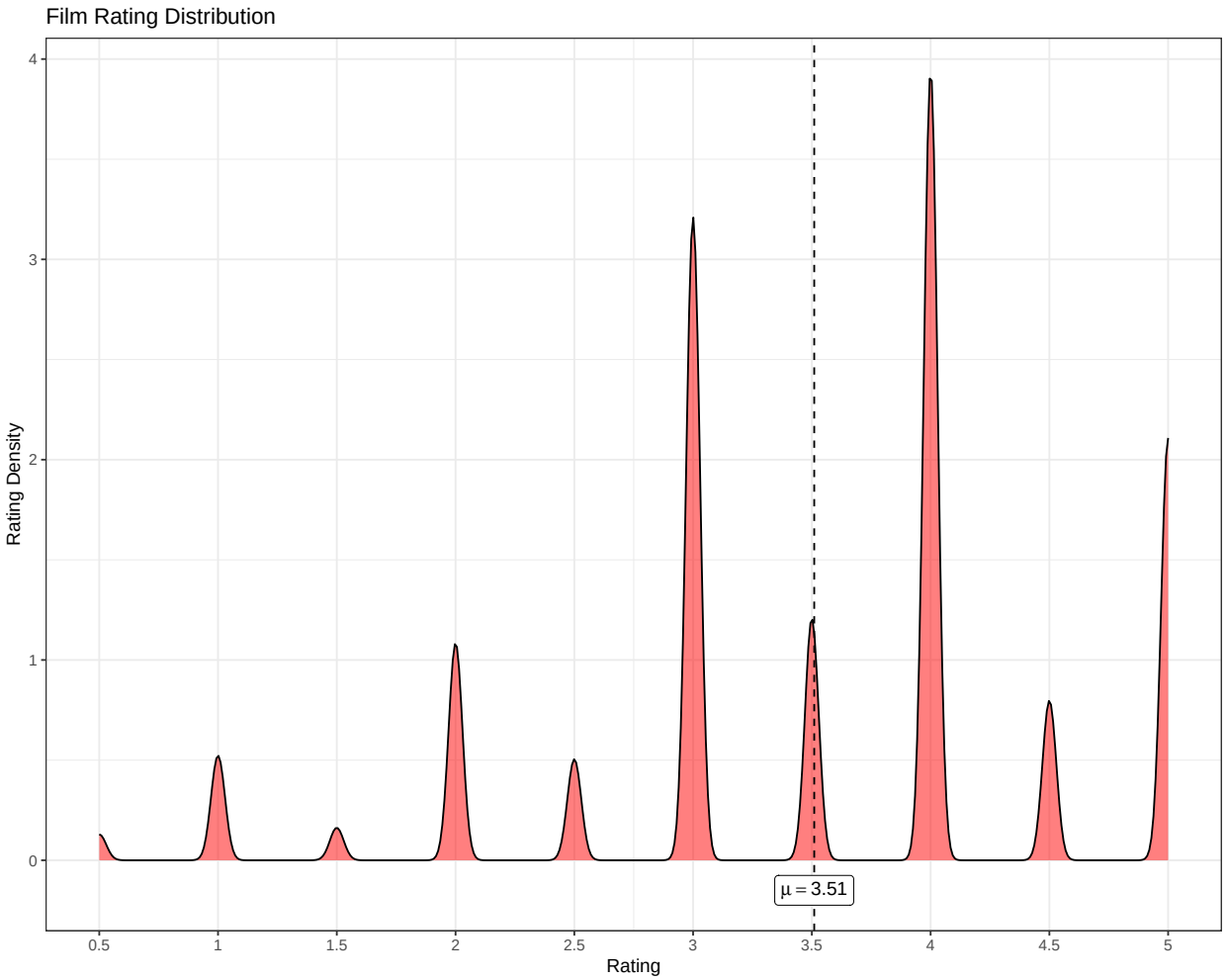
Some of the extracted features have low variance and would require cleaning if necessary, genres for example decrease in unique observations as level increases. Features such as timestamp and title will be ignored in training. While there are models that may use these features in their current for these will be removed prior to training the model.

Numeric Data Analyzing the numeric variables it can be observed that they have low confidence intervals, in particular rating has a confidence interval of 3.51 to 3.51 when rounded to two decimal points.

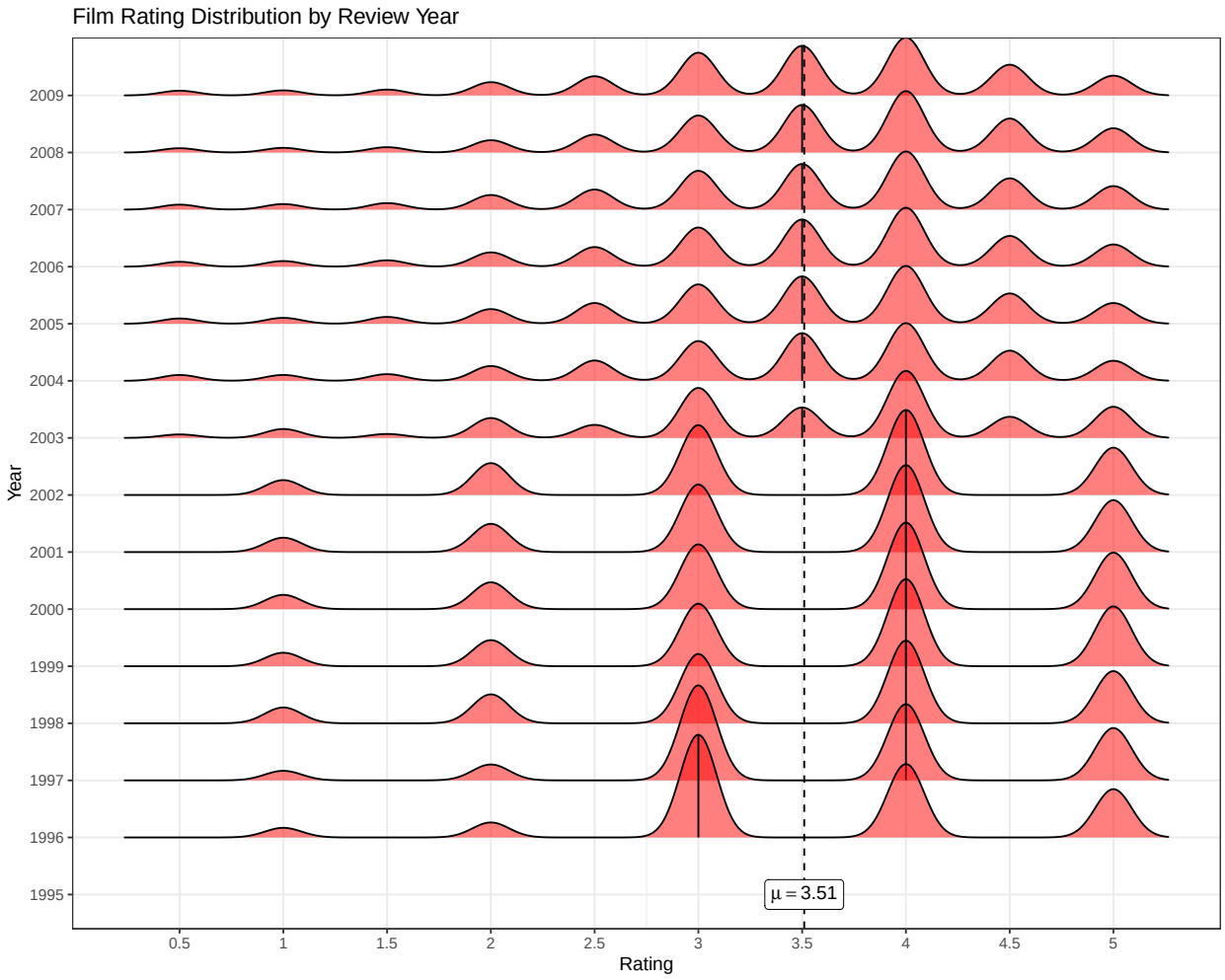
Rating Distribution Per the source documentation ratings have a scale of 0.5 to 5 in $\frac{1}{2}$ star increments. The distribution of these values demonstrates that $\frac{1}{2}$ values are far less common than full-star ratings. However the mean rating, 3.51 is within range of a $\frac{1}{2}$ star rating.

Variables	Variable Class	Unique Observartions
rating	numeric	10
user_id	factor	69,878
movie_id	factor	10,677
film_year_of_release	factor	94
film_age	numeric	96
timestamp	POSIXct, POSIXt	5,438,099
title	character	10,407
genre_1	factor	20
genre_2	factor	19
genre_3	factor	18
genre_4	factor	16
genre_5	factor	14
genre_6	factor	10
genre_7	factor	4
genre_8	factor	2
year	numeric	15
month	ordered, factor	12
day	integer	31
day_of_the_year	numeric	366
day_of_the_quarter	numeric	92
weekday	ordered, factor	7
hour	integer	24
minute	integer	60
second	numeric	60
user_reviews	integer	5,308
movie_reviews	integer	25,037

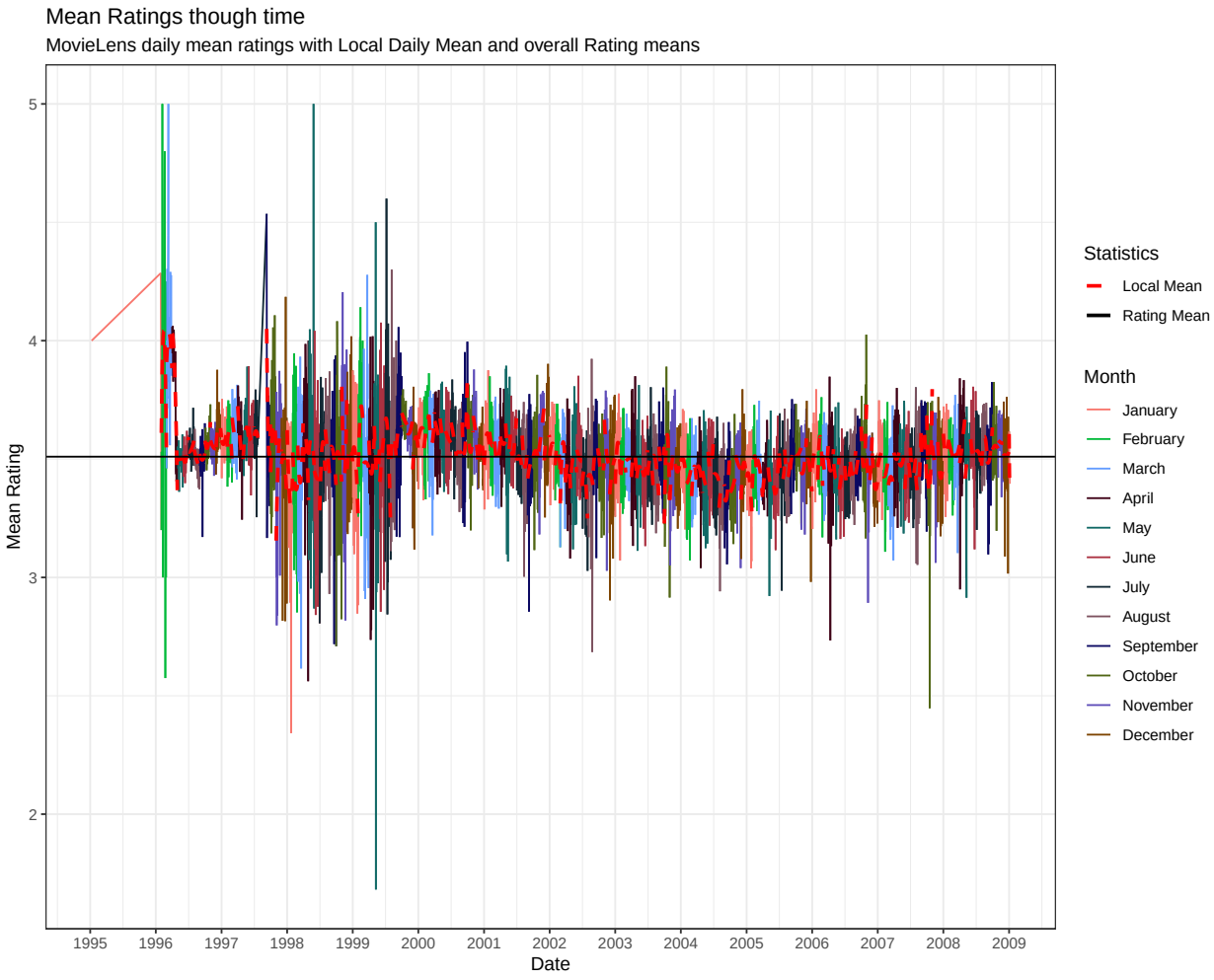
Variable	Estimate	Statistic	P.value	Parameter	Conf.low	Conf.high	Method	Alternative
day	15.60	4,754.17	0	7,200,084	15.59	15.60	One Sample t-test	two.sided
day_of_the_quarter	45.80	4,635.15	0	7,200,084	45.78	45.81	One Sample t-test	two.sided
day_of_the_year	191.18	4,750.39	0	7,200,084	191.11	191.26	One Sample t-test	two.sided
film_age	11.98	2,336.00	0	7,200,084	11.97	11.99	One Sample t-test	two.sided
hour	12.48	4,586.59	0	7,200,084	12.48	12.49	One Sample t-test	two.sided
minute	29.63	4,598.28	0	7,200,084	29.62	29.64	One Sample t-test	two.sided
movie_reviews	2,714.29	2,062.75	0	7,200,084	2,711.71	2,716.87	One Sample t-test	two.sided
rating	3.51	8,889.34	0	7,200,084	3.51	3.51	One Sample t-test	two.sided
second	29.50	4,570.90	0	7,200,084	29.48	29.51	One Sample t-test	two.sided
user_reviews	169.33	1,722.91	0	7,200,084	169.14	169.52	One Sample t-test	two.sided
year	2,002.20	1,448,038.21	0	7,200,084	2,002.20	2,002.20	One Sample t-test	two.sided



Grouping the rating distribution by year we can observe that $\frac{1}{2}$ star ratings are seemingly not available for the first several years. It can also be observed that the mean rating by year differs from the overall mean rating and only centers on the overall mean rating once $\frac{1}{2}$ star ratings are available.

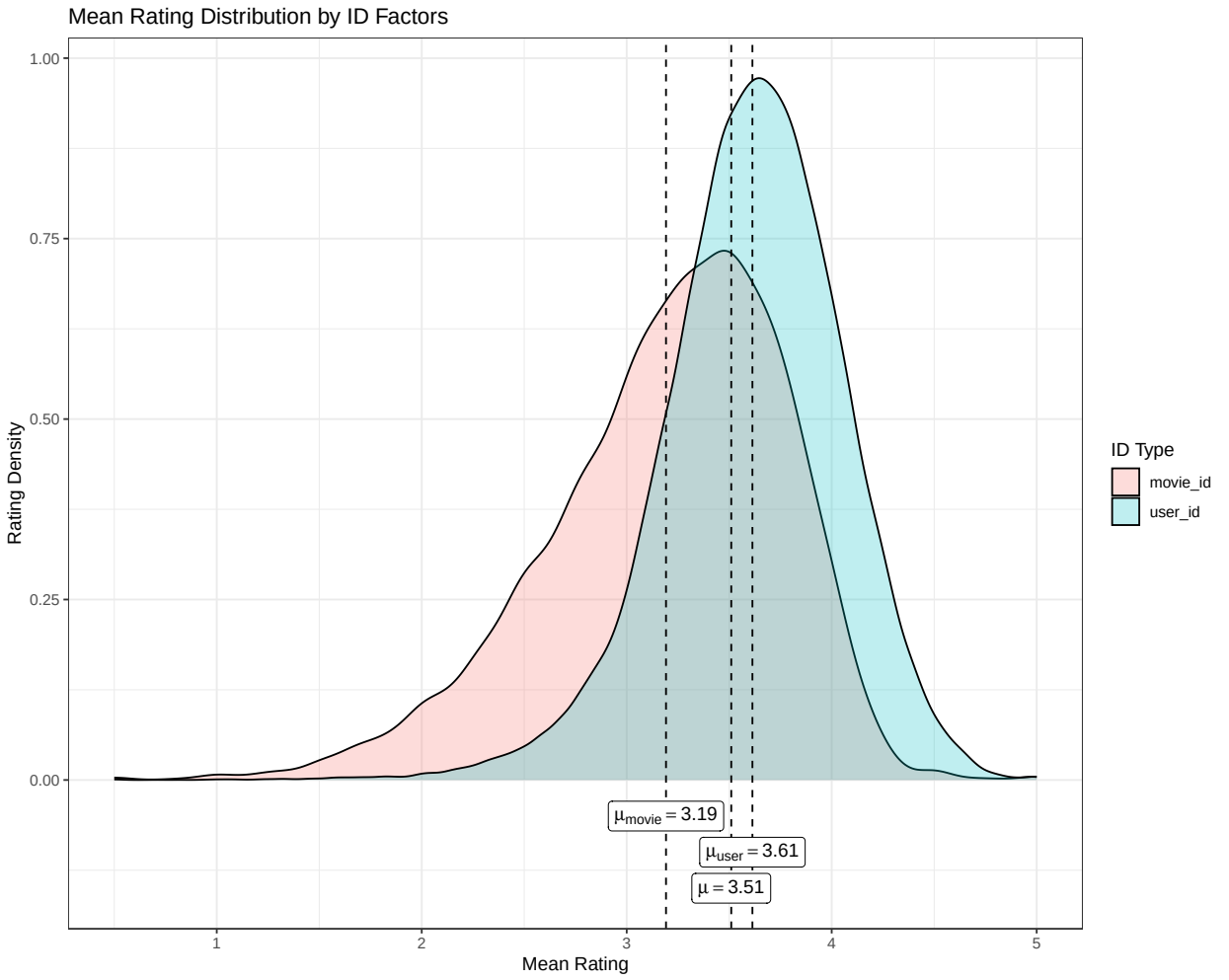


Observing the mean rating by date of review we can observe the similar effects. The daily mean rating tends towards the overall mean rating with larger and more erratic spreads prior to 2002, after which the daily mean ratings stabilize. There also seems to be some effects by month. This could be due to some films exhibiting seasonality and new reviews being performed close to these dates.



From this we can expect year, month and day to have some effects in predictions.

Rating Distribution by Users and Films Grouping mean ratting distributions by users and films we can observe a clear discrepancy.



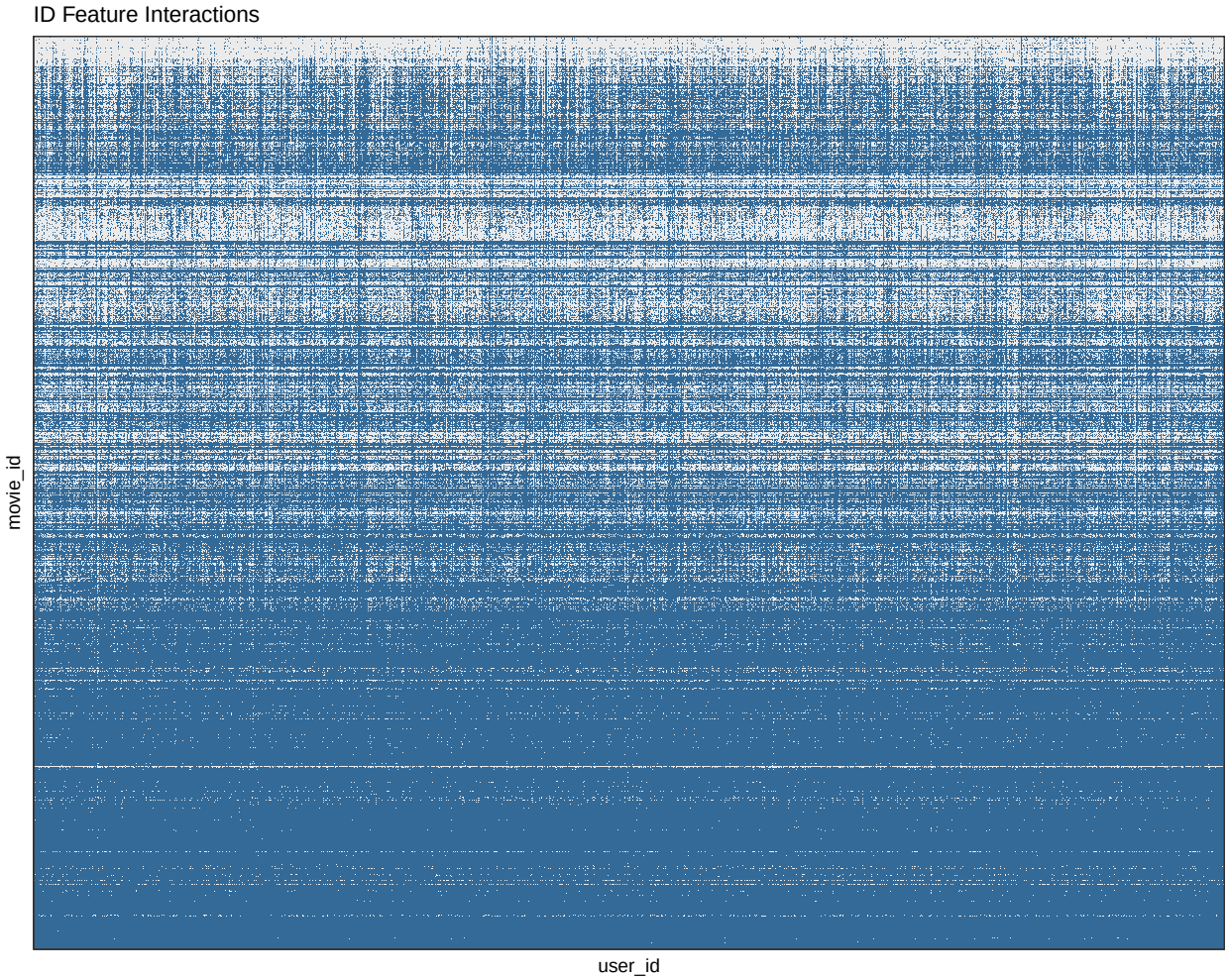
Users have a tendency to have higher mean ratings while films have a tendency to have lower mean ratings.

Estimate	Estimate1	Estimate2	Statistic	P.value	Parameter	Conf.low	Conf.high	Method	Alternative
-0.42	3.19	3.61	-71.91	0	12,583.2	-0.43	-0.41	Welch Two Sample t-test	two.sided

The difference in means are statistically significant and different from zero, we can expect that users and films will have different effects on prediction. These variables have different unique counts and proportions within the dataset, the large discrepancy in proportion requires to analyze for any repeated interactions.

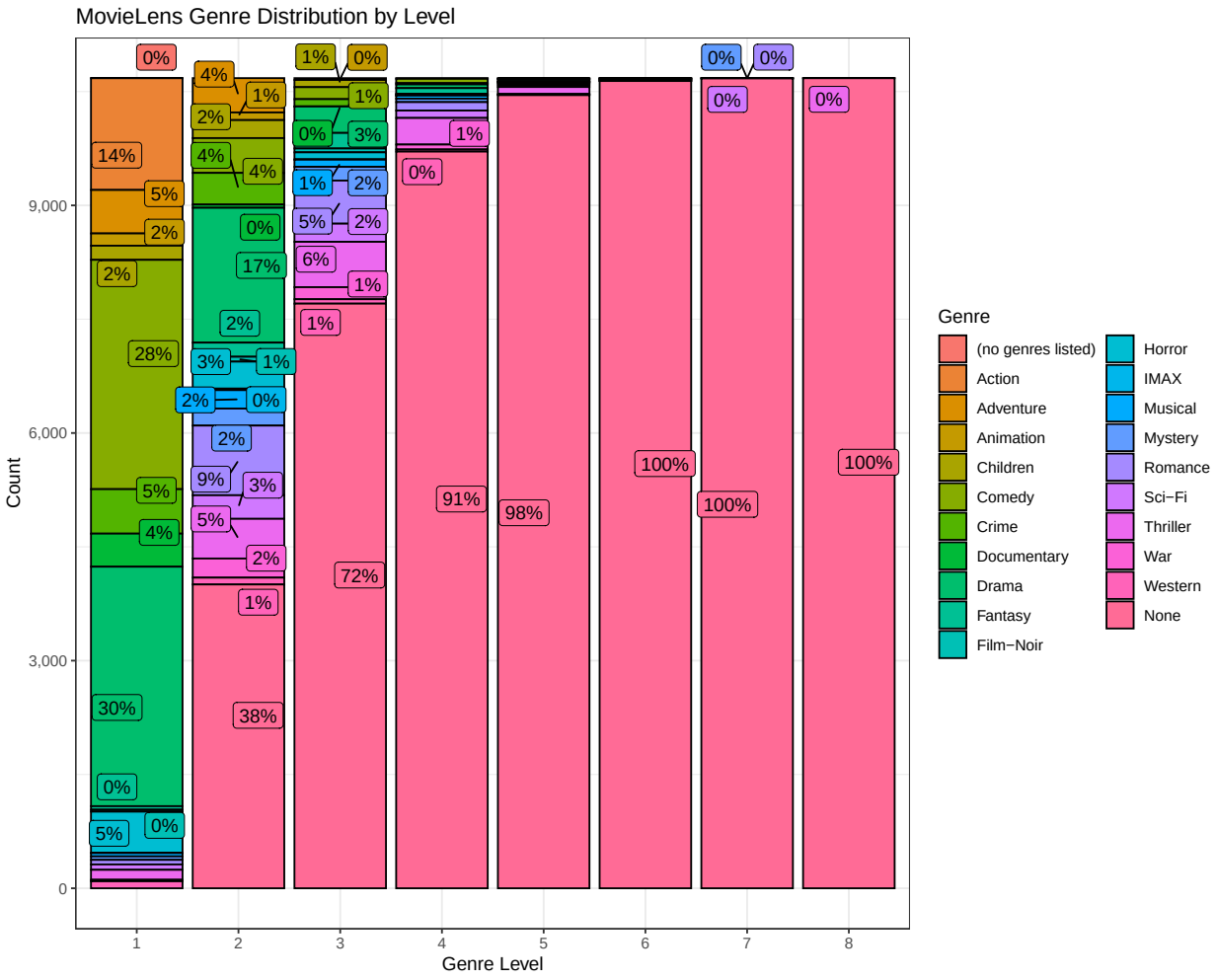
Id Type	N	Proportion
user_id	69,878	87%
movie_id	10,677	13%

The combined effects of these variable will be minimal as them have near unique interactions as observed in the follwing plot.



A base model of $\hat{Y} = User_{effects} + Film_{effects}$ can be expected. We can also determine that there are variable amounts of ratings for both users and films, this can have effects on predictions that will require some added analysis of the model in order to take these balance issues into account.

Genre Rating Distribution There are a total of 8 genre variables extracted from the original pipe separated variable. Out of these only the first two will be considered due to lack of unique observations having near zero variance.



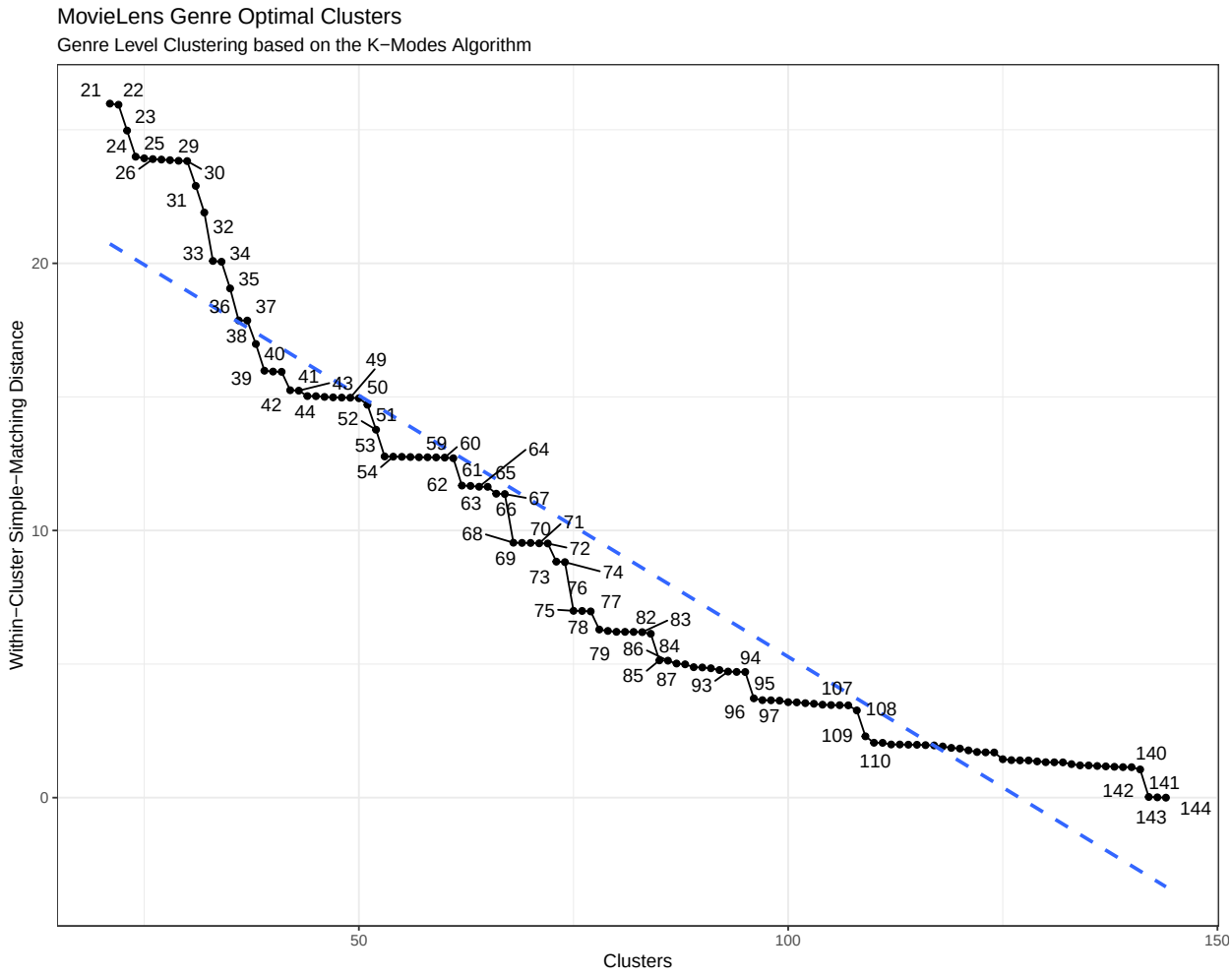
The rating distribution for the interactions of these two variables reveals that there genre combinations can have differing effects on rating.

Film-Noir is the highest rated Genre on both genre levels, however when there are instances where the combination with other genres causes a major dip in mean rating.

Genre Clusters In order to minimize model predictors a feature names *Genre Clusters* will be developed. The development of this feature will be based on the *K-Modes*[3] clustering algorithm. The algorithm is in essence a frequency based variant of the *K-Means* Clustering Algorithm. The generated cluster in this analysis will be based on the statistical mode of concurrence of categorical features. In this case the concurrence of genres by level dis a list of distinct films. The maximum amount of distinct genre clusters for this dataset is 144 , this can be reduced to a more manageable set of clusters based on *K-Modes*.

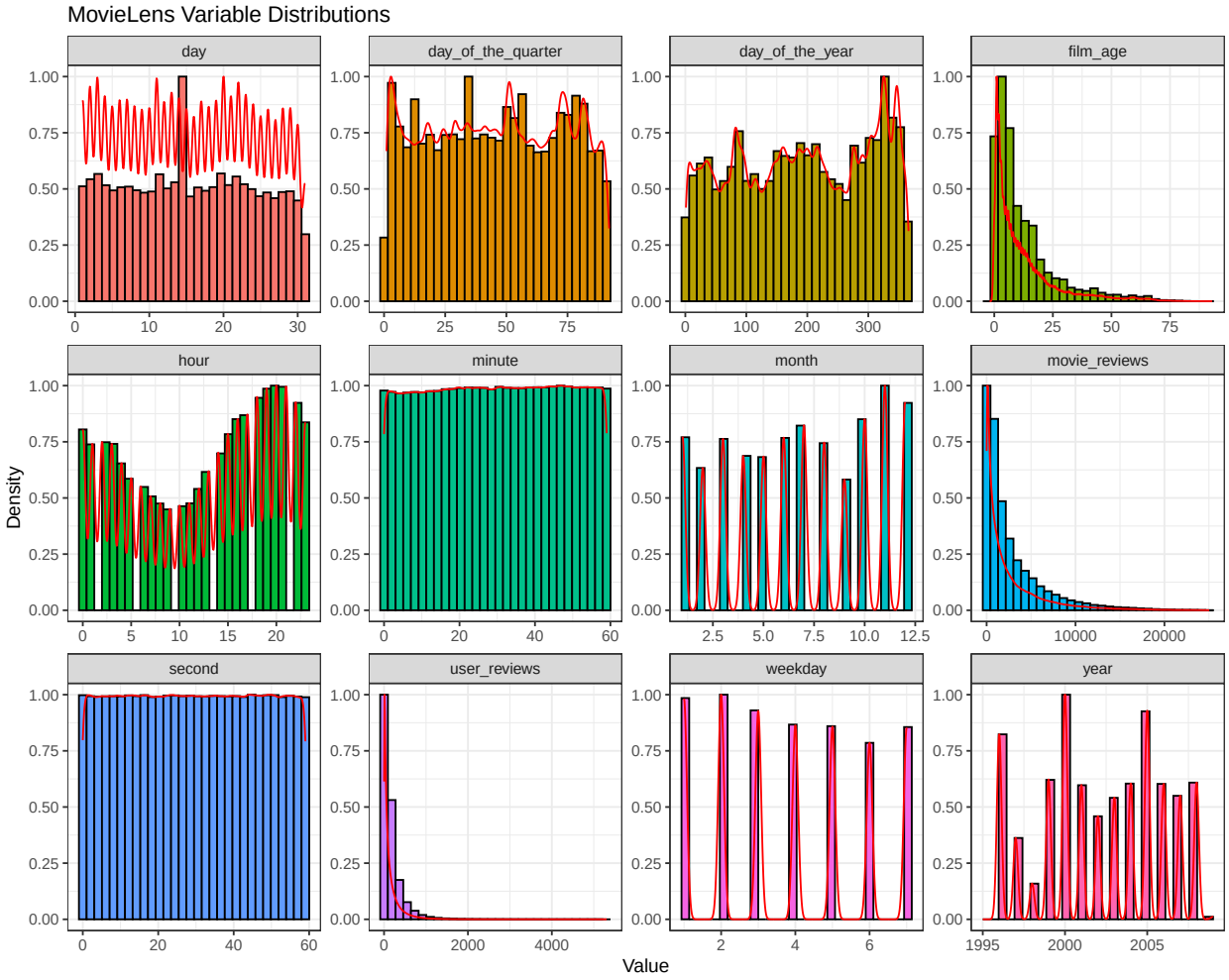
Running the K-Modes algorithm and using the general linear slope of *Within-Cluster Simple-Matching Distance* of we can determine that the optimal amount of genre clusters is 36.

[3]https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/

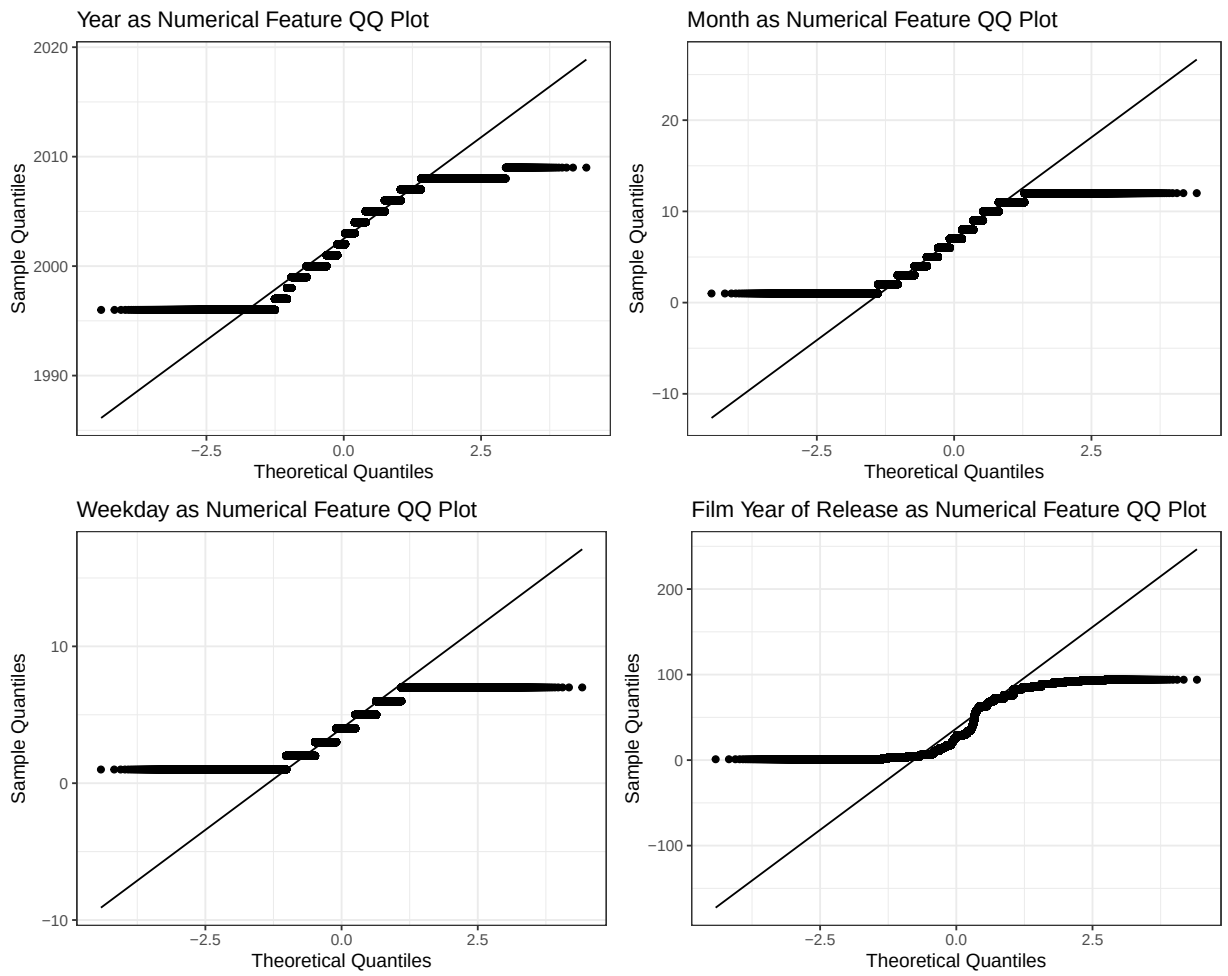


Given how Genre Clusters are defined this new feature will have a 1:1 alignment with film, therefore the usefulness of this feature will be determined only in relation to $User_{effects}$.

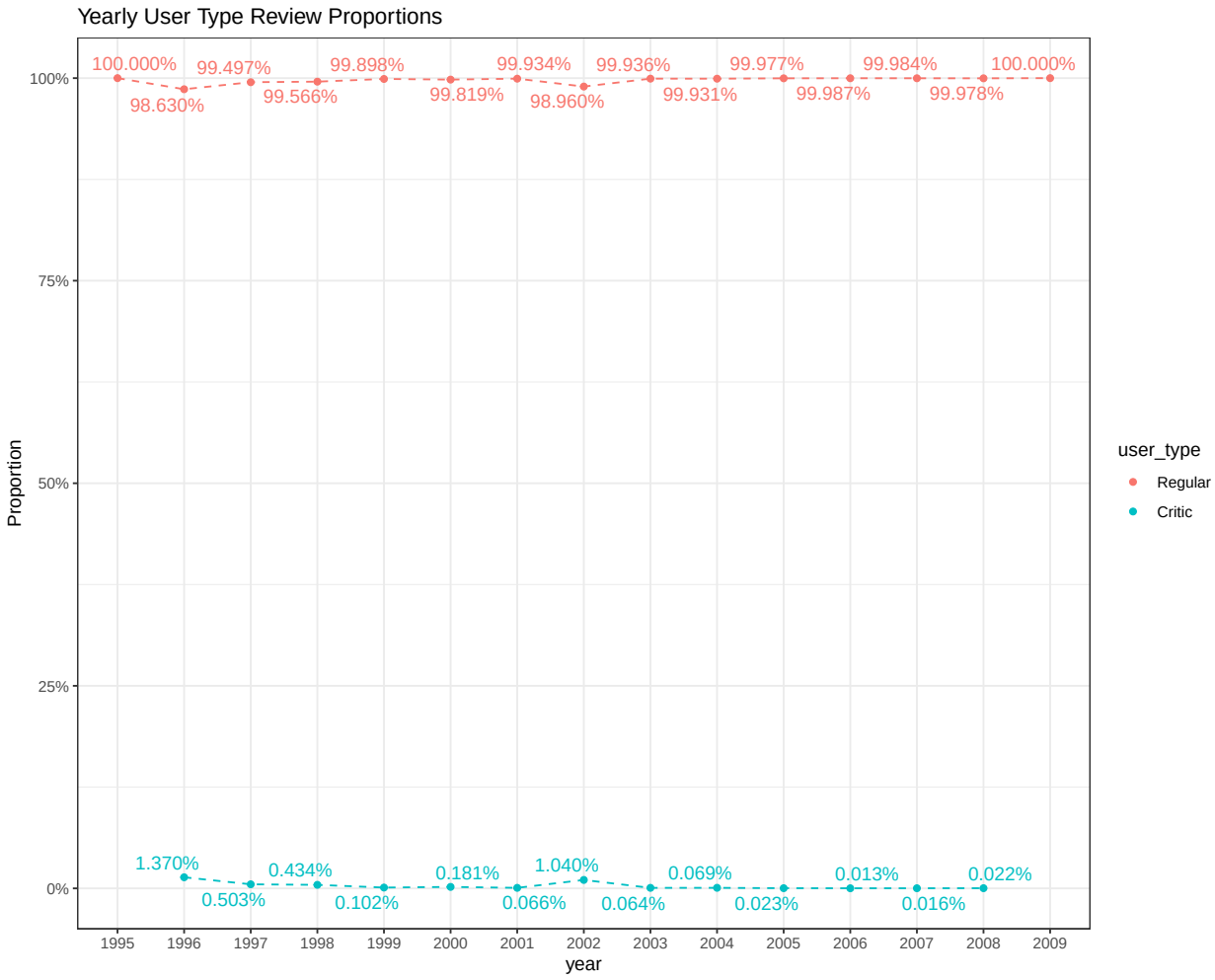
Numeric Predictor Distributions Exploring the distributions of numeric predictors we can observe that Minute, Second and Day of the Quarter are Nearly uniformly distributed and would therefore be ill suited for prediction. Movie and User reviews have an approx Negative Binomial Distribution overall. However users by user and film by film the count should be linear as reviews are expected to increase as time passes and never decrease.



Normality of Numeric Predictors Utilizing QQ-Plots for determining the normality of predictors we can observe a clear tendency for Year, Month and Weekday to act as categorical variables. However given the ordered nature of the predictors these are expected to function better as numerical predictors as both PCA and linear regression slopes may yield improved results to numeric data effects and interactions.

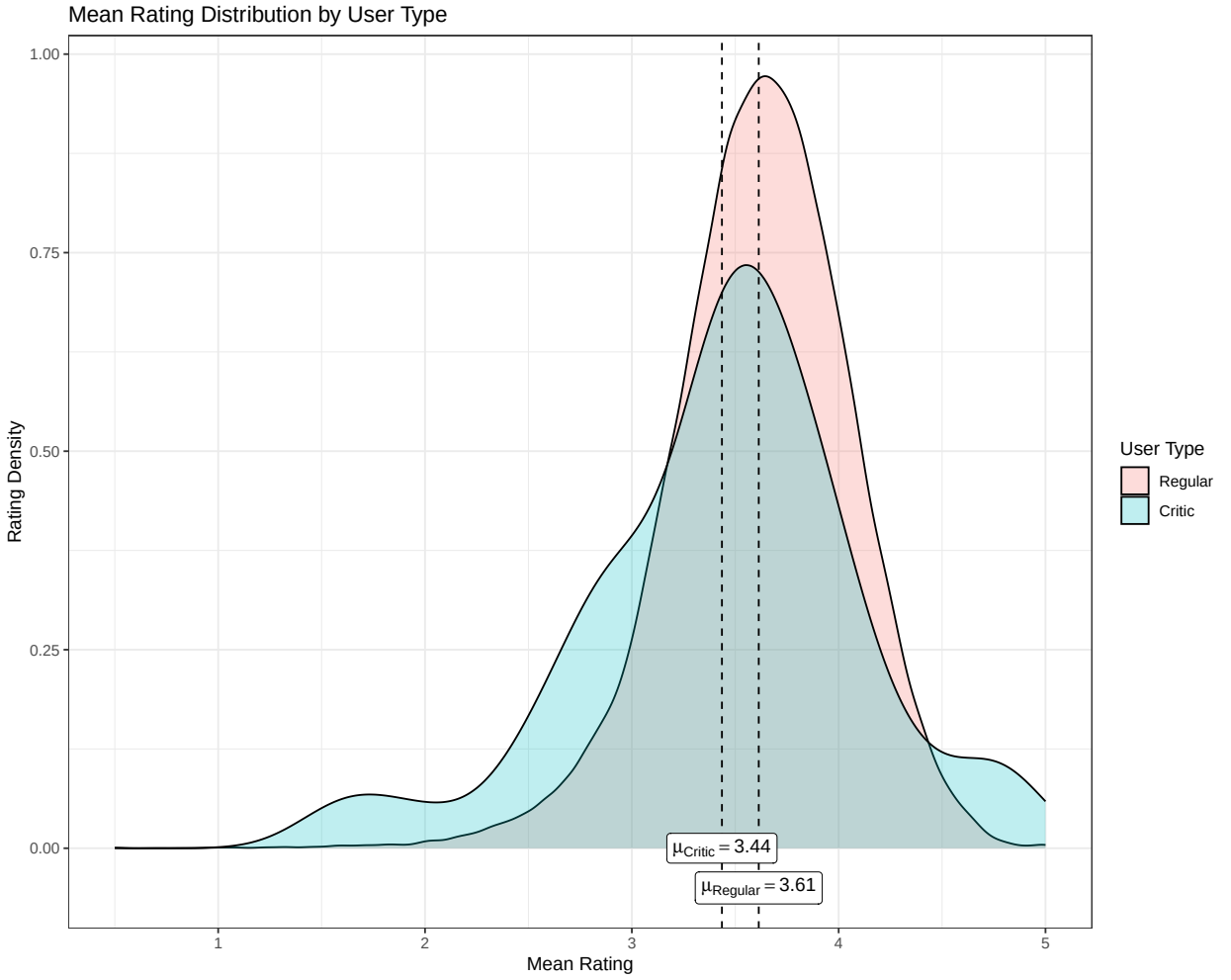


Critics There is an odd tendency in film age where there seems to be negative film ages. Exploring this further reveals that there is as subcategory of users which contain negative film reviews. The proportion of users remains constant through time.



This class of users also exhibits statistically different mean rating than other users.

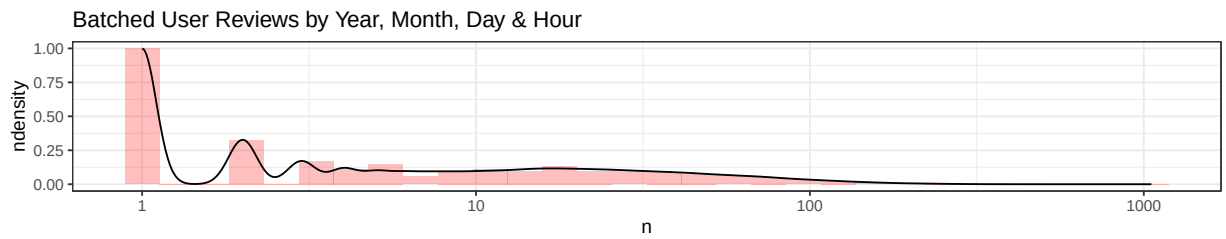
Estimate	Estimate1	Estimate2	Statistic	P.value	Parameter	Conf.low	Conf.high	Method	Alternative
-0.1780412	3.435526	3.613567	-2.857539	0.0050826	113.159	-0.3014783	-0.0546041	Welch Two Sample t-test	two.sided

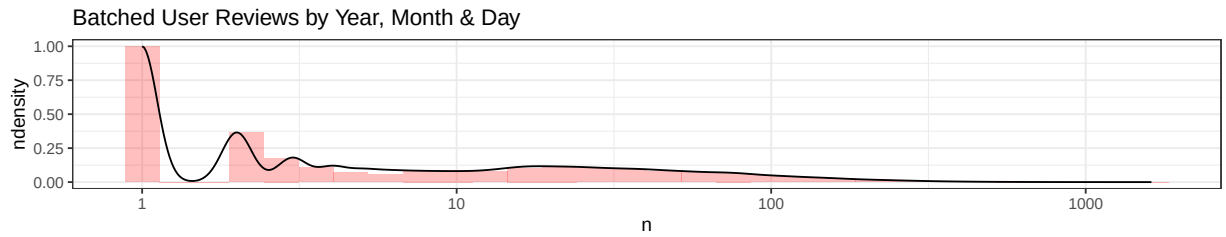


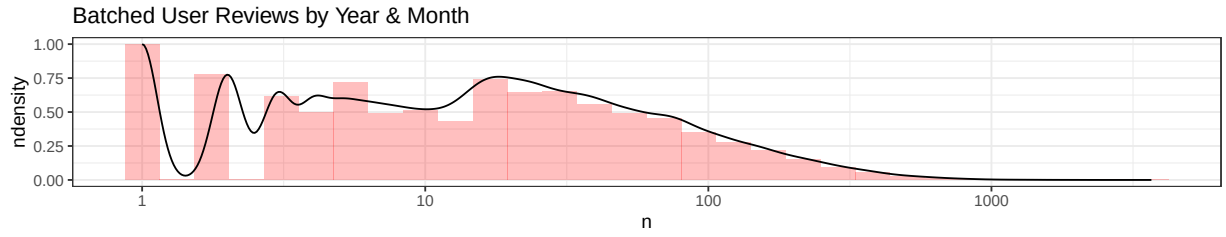
There is insufficient data to determine if any other class of user can be considered a critic, while imperfect and only exhibiting marginal differences this feature may be relevant in some fashion and will be kept in its current state. Considering how the feature is defined this will be used in effect only with $Film_{effects}$.

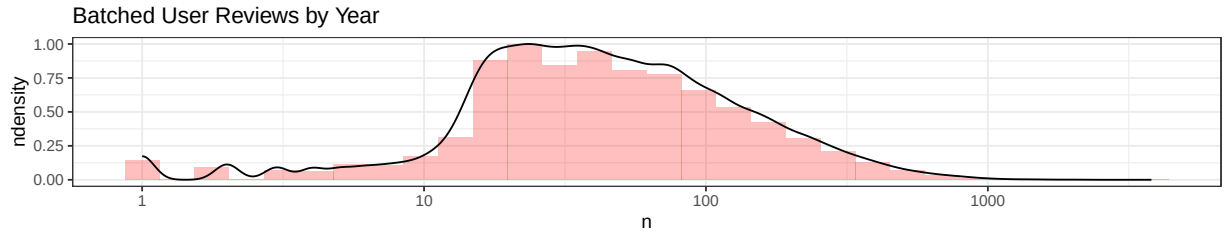
Batched and Accumulated Reviews During the Summer of 2023 there was a film event known as **Barbenheimer**[4], this event celebrated the simultaneous film release of Greta Gerwig's Barbie and Christopher Nolan's Oppenheimer. Participants in this mass event watched both films back-to-back, this event included several noted film critics and reviewers. Using this as a back drop, additional features concerning batched reviews and accumulated reviews by batches are defined.

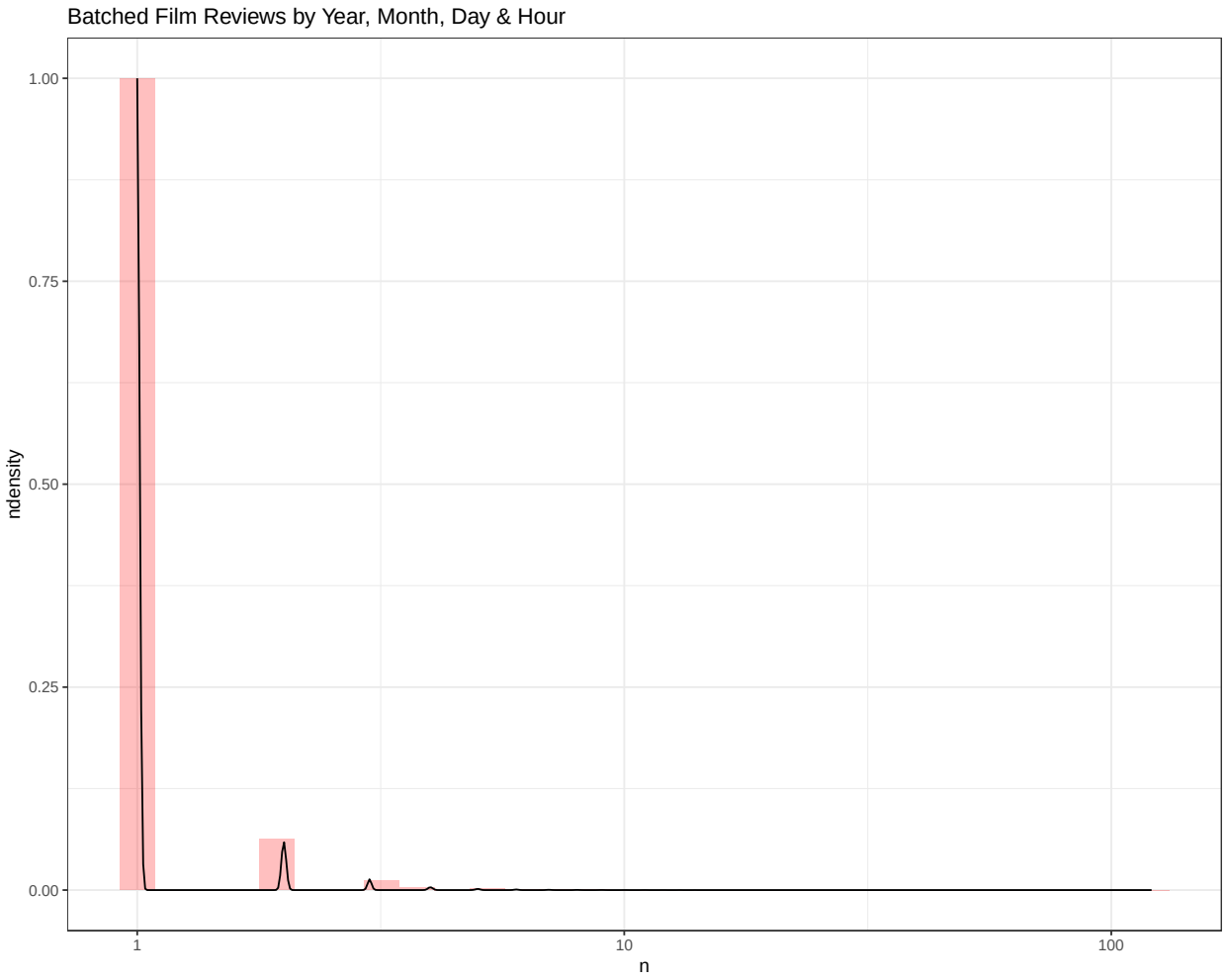
[4]https://www.wikiwand.com/en/Barbenheimer

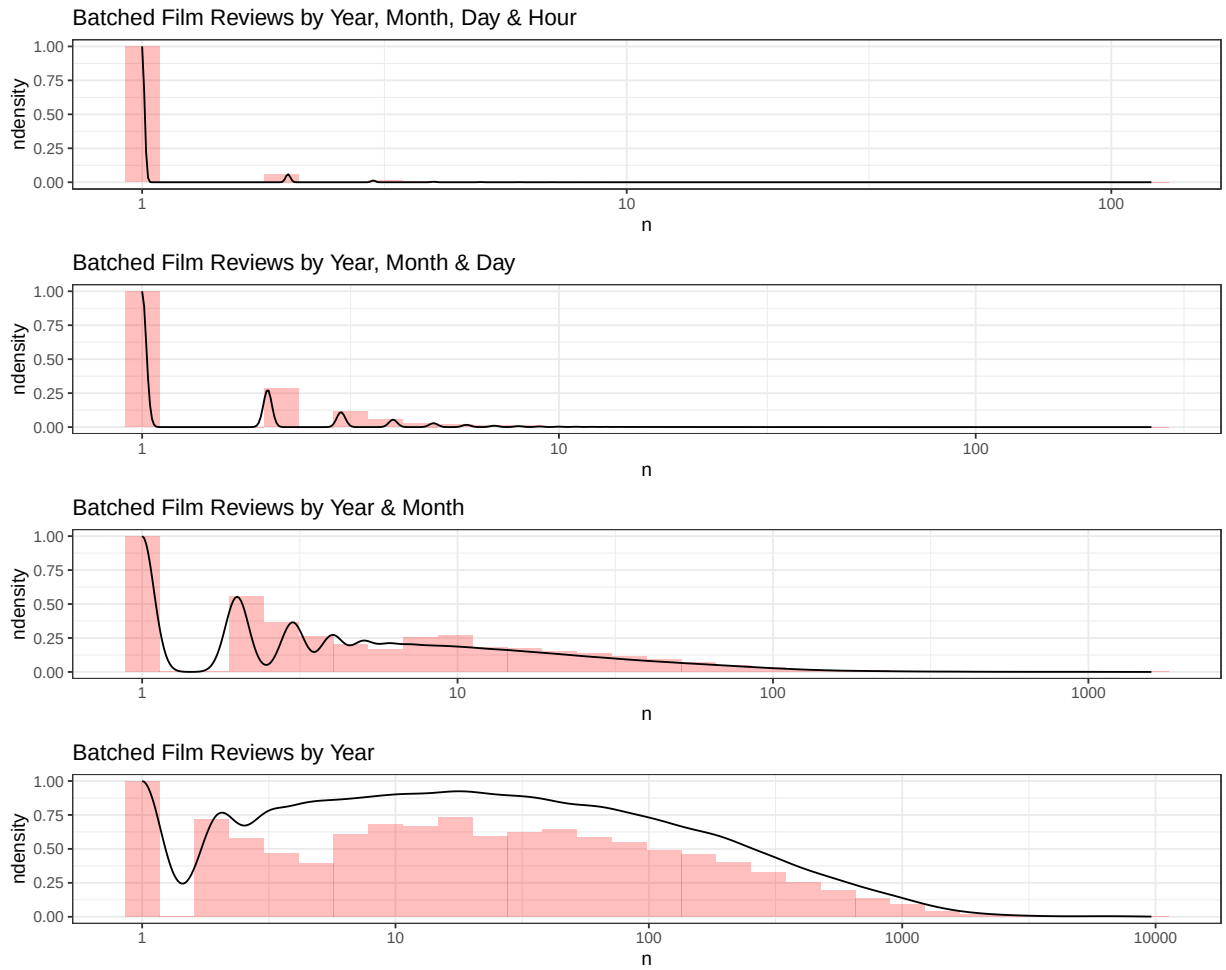

Batched User Reviews by Year, Month, Day & Hour
ndensity
1.00
0.75
0.50
0.25
0.00
1
10
100
1000
n


Batched User Reviews by Year, Month & Day
ndensity
1.00
0.75
0.50
0.25
0.00
1
10
100
1000
n


Batched User Reviews by Year & Month
ndensity
1.00
0.75
0.50
0.25
0.00
1
10
100
1000
n


Batched User Reviews by Year
ndensity
1.00
0.75
0.50
0.25
0.00
1
10
100
1000
n


Batched Film Reviews by Year, Month, Day & Hour
1.00
0.75
0.50
0.25
0.00
ndensity
1
10
100
n



Films are best batched in Year-Month groups while Users were batched as Year-Month-Day groups

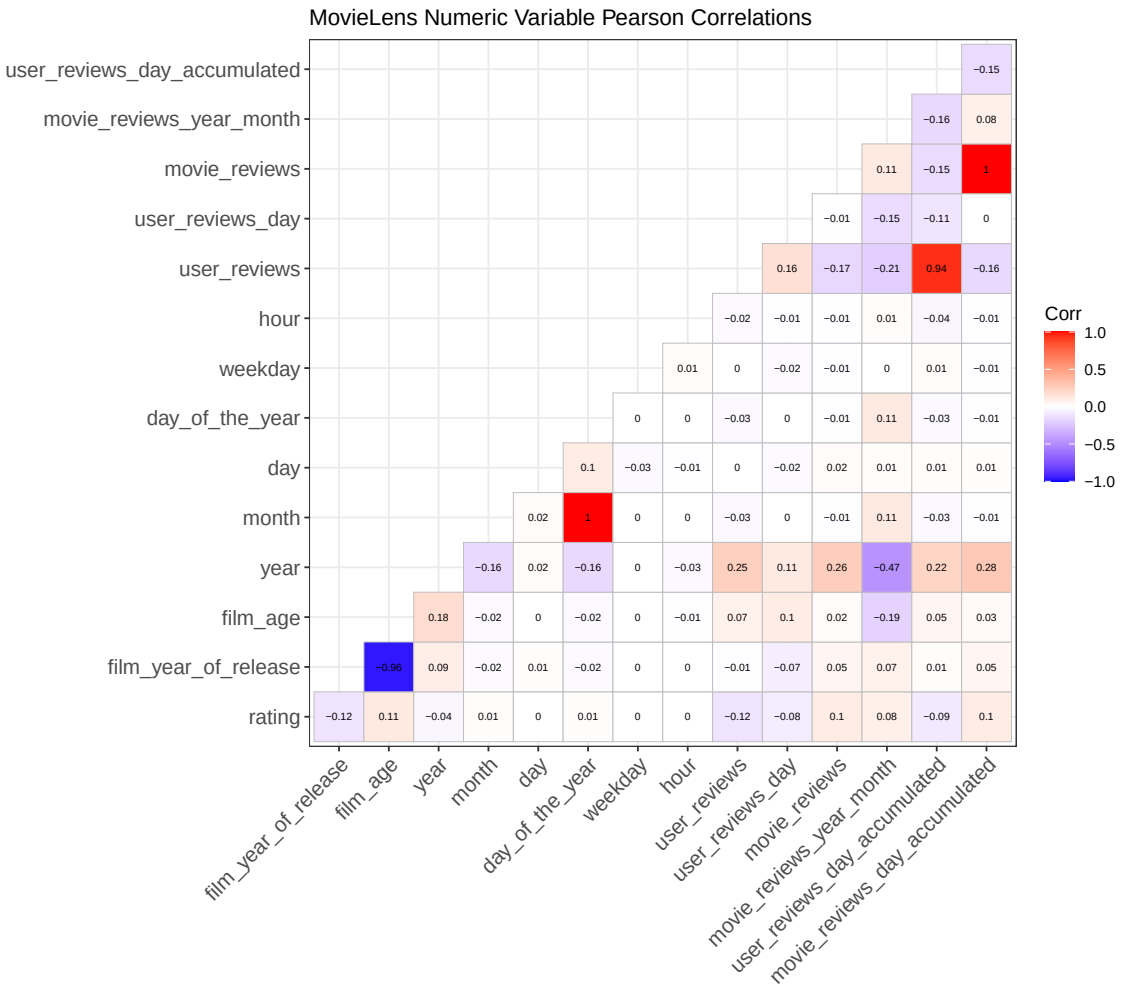
Feature Selection

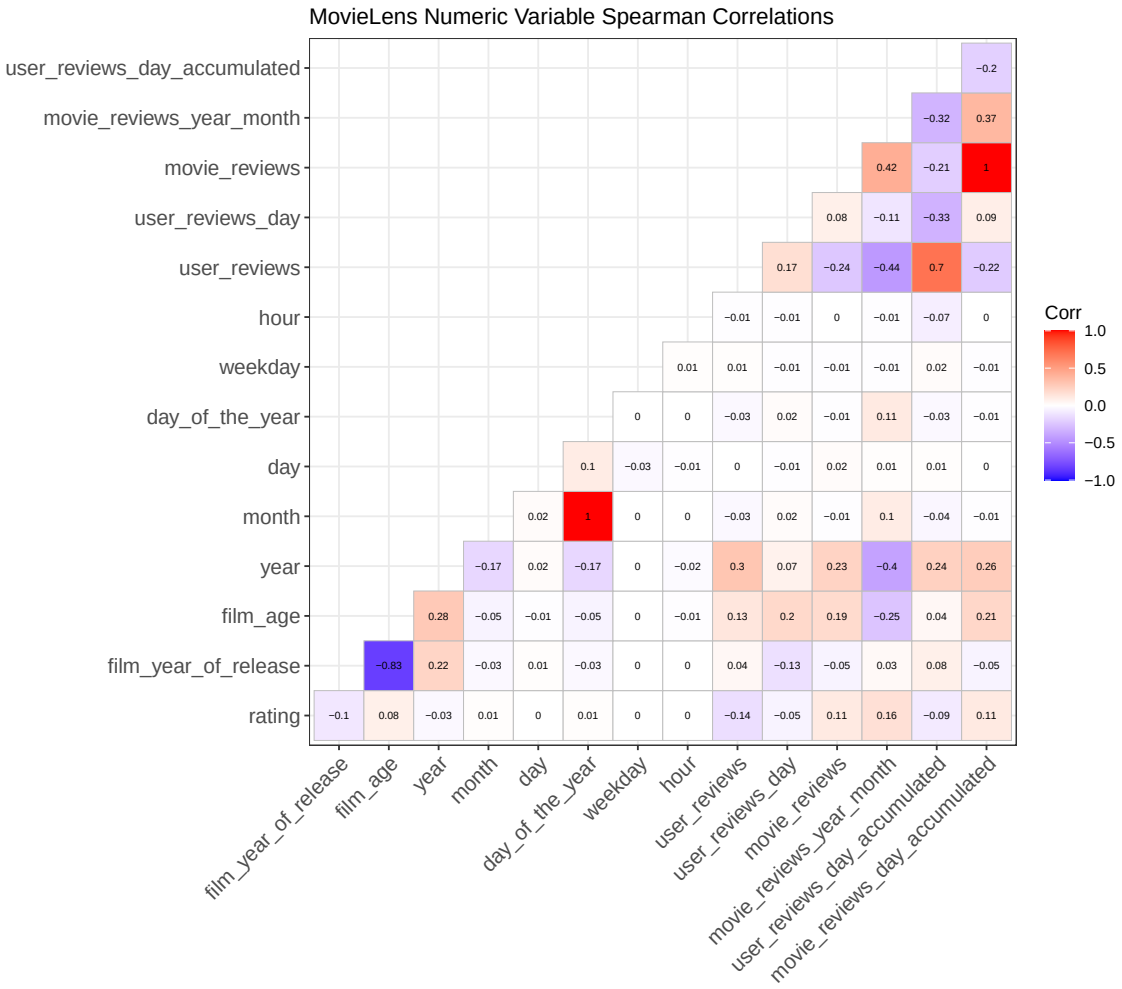
Feature Selection will use a combination of filter methods and wrapper methods. The Filter methods used will be Filtering by Near-Zero Variance, Numeric Feature Correlations (both Pearson and Spearman), Linear Dependencies, and Mutual Information. The sole wrapper method applied will consist of the *Boruta Algorithm* utilizing *XGBoost* for calculating variable importance.

Near Zero Variance Filtering

Near Zero Variance filtering method found that User Type and Accumulated Reviews have near zero variance, however as these are variables of interest with potential interactions they will be left within the feature space at this time

Correlation Filtering



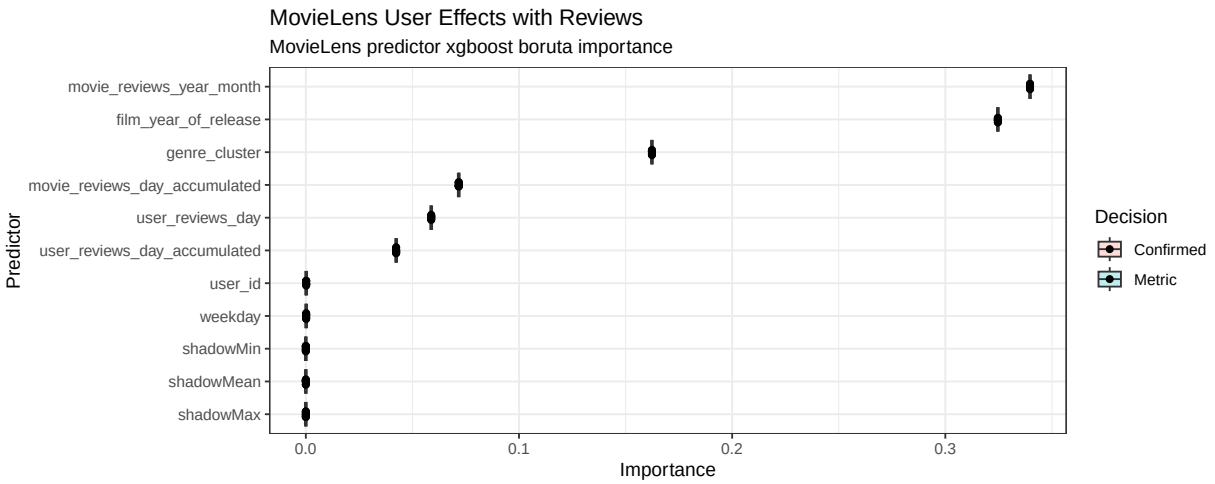


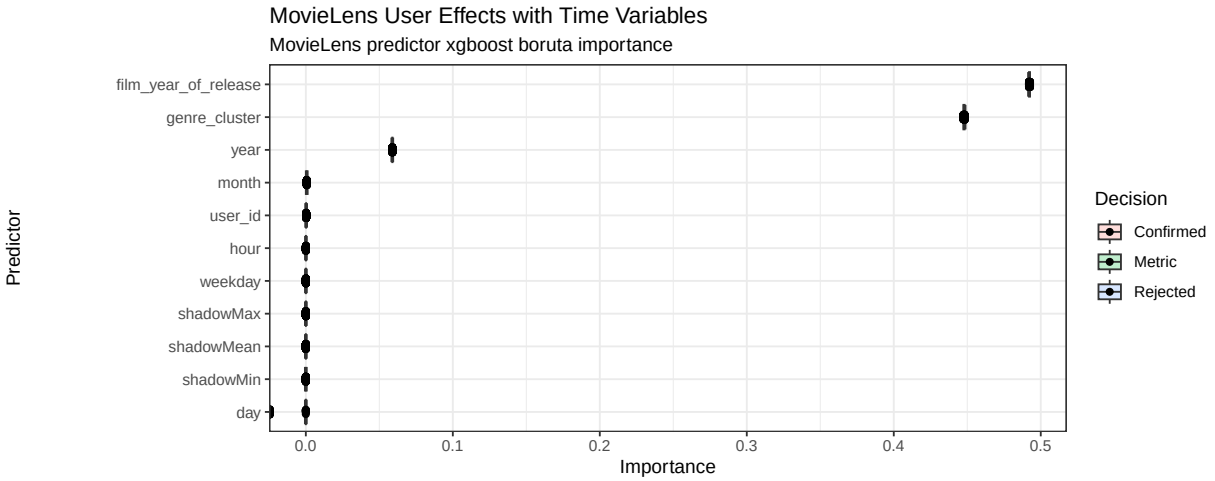
Day of the year, Film age and base reviews were selected for removal. Since PCA can use both year and year of release to encode film age this selection is accepted.

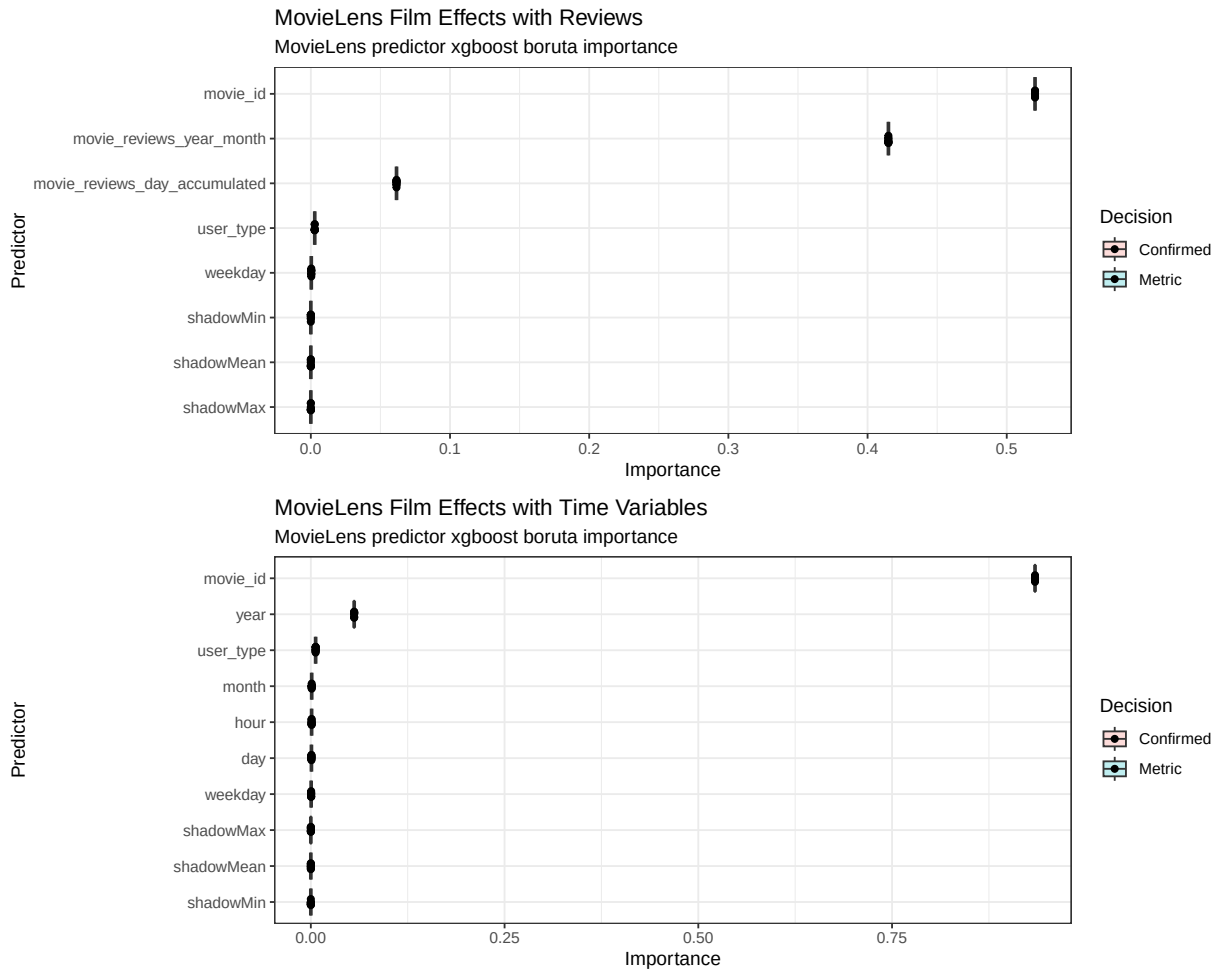
Boruta Feature Selection

The Boruta Feature Selection Algorithm[5] will be applied in such a way that models based on reviews and models based on time features will be compared. This will allow the selection of an expanded model beyond the current expected model $\hat{Y} = User_{effects} + Film_{effects}$. In this instance feature importance will be calculated using XGboost methodologies instead of the typical Random Forrest methods as the large amount of observations will render the method ineffective.

[5]https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/


MovieLens User Effects with Reviews
MovieLens predictor xgboost boruta importance
movie_reviews_year_month
film_year_of_release
genre_cluster
movie_reviews_day_accumulated
user_reviews_day
user_reviews_day_accumulated
user_id
weekday
shadowMin
shadowMean
shadowMax
Predictor
0.0
0.1
0.2
0.3
Importance
Decision
Confirmed
Metric


MovieLens User Effects with Time Variables
MovieLens predictor xgboost boruta importance
film_year_of_release
genre_cluster
year
month
user_id
hour
weekday
shadowMax
shadowMean
shadowMin
day
Predictor
0.0
0.1
0.2
0.3
0.4
0.5
Importance
Decision
Confirmed
Metric
Rejected



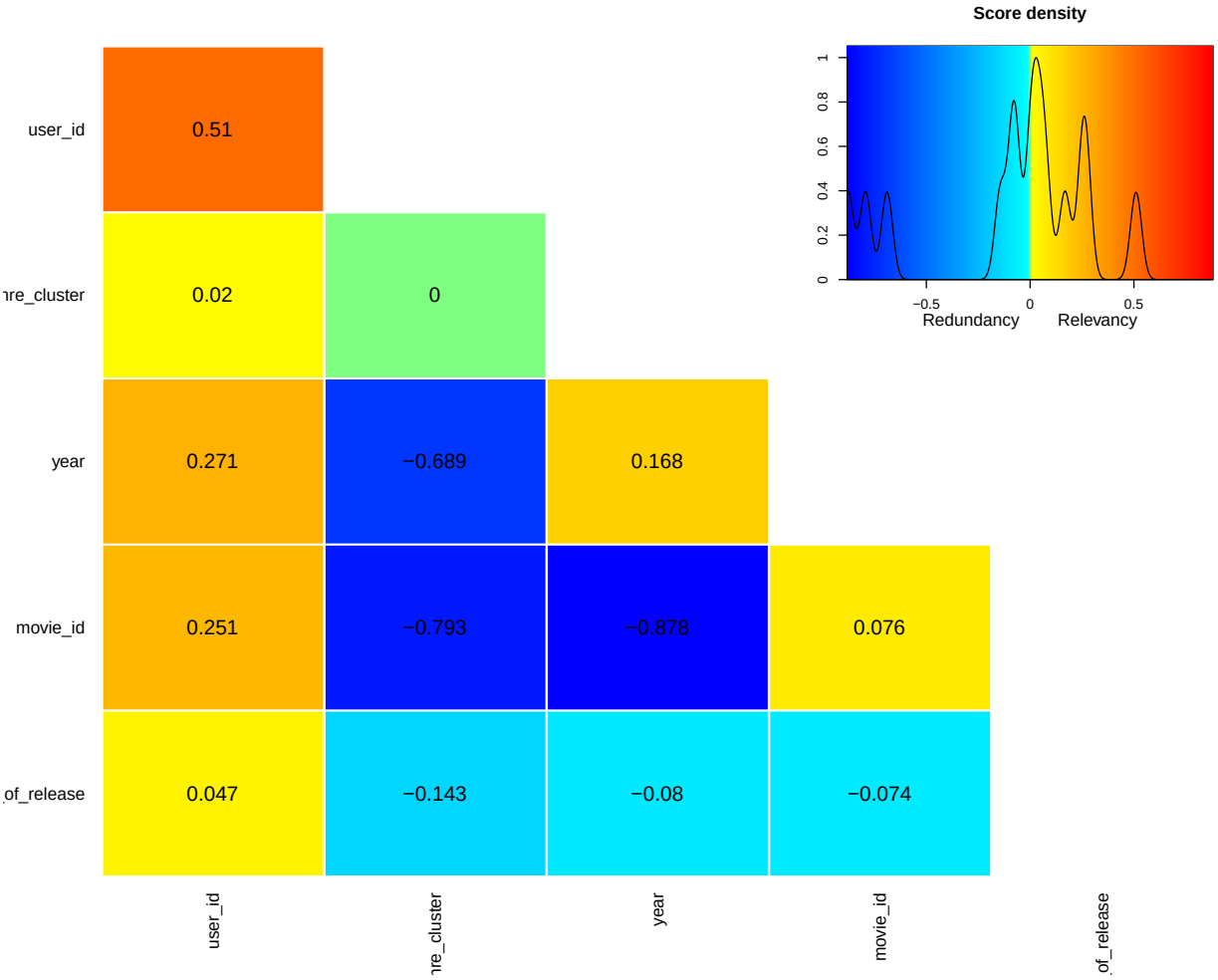
Analyzing the relative performance of the models in can be concluded that the simpler model is derived from the use of time variables as opposed to review variables. The increase in relative importance for remaining features while decreasing their amount will simplify the training process.

Entropy & Mutual Information

As an additional feature selection method Entropy[6] and Mutual Information[7] between the predictors will be calculated. Entropy of as random variable refers to the average level of uncertainty naturally present given the variables outcomes. In simple term it indicated the level of expected surprise in the outcome of an event. Mutual Information is a special case of entropy, where the uncertainty in outcome is calculated for two variables. These calculations will be applied to the current feature space to gauge the relative importance of features in probabilistic terms using the *varrank* package and the *infotheo* package.

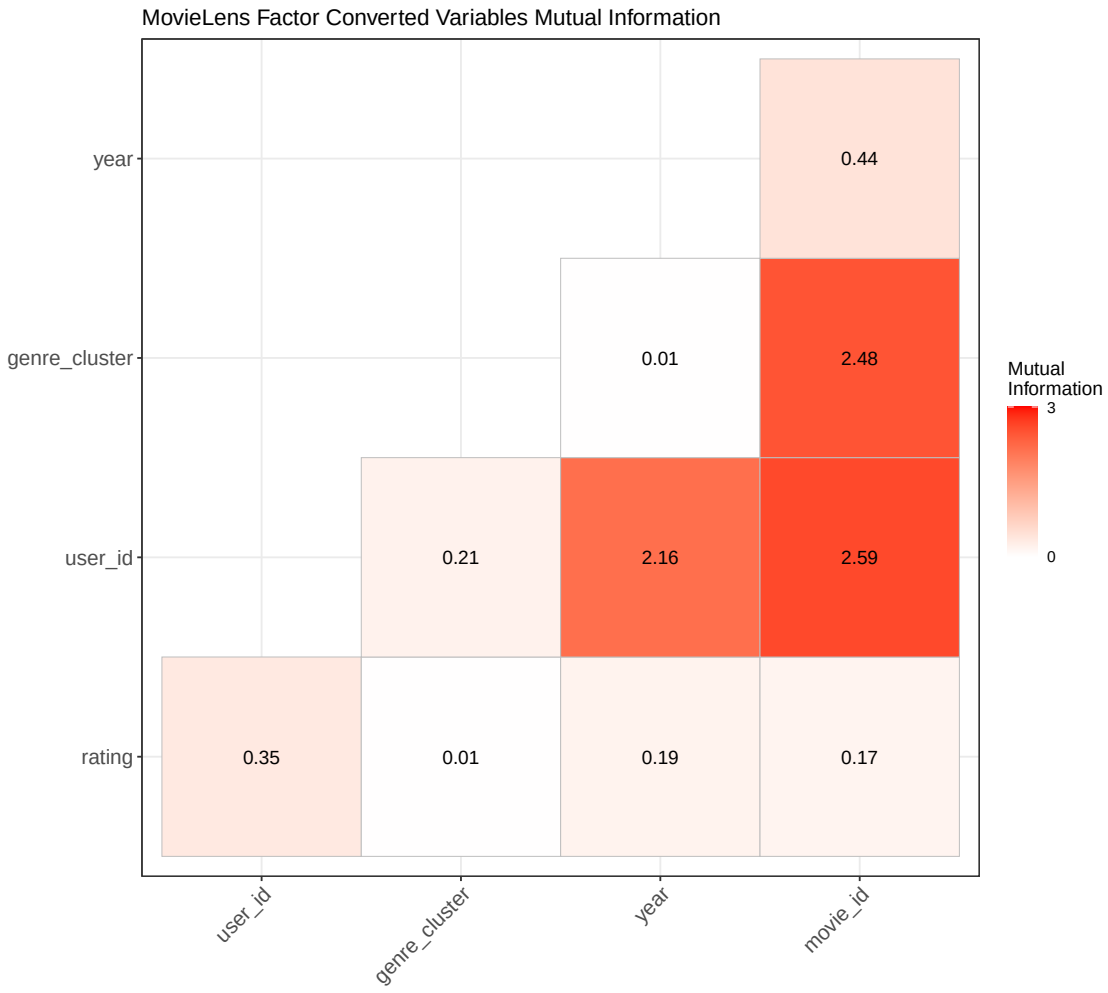
[6]https://youtu.be/YtebGVx-Fxw?si=5O-_vIImiA3uIBLC
[7]https://youtu.be/eJIp_mgVLwE?si=Jj6-n-JYOYVUfjrv

Entropy Variable Rank



We can observe in the above plot that optimal training order given by the entropy of the features is: User Id, Genre Cluster, Year, Movie Id. Considering this outcome Mutual information can be calculated to gauge feature interactions ahead of training.

Mutual Information



Mutual Information calculations reveal that genre clusters has little to no concurrence with rating especially compared to other predictors. It does however share large amounts of information with user so an interaction has high potential. Genre Cluster however is overshadowed by Year in terms of potential user interaction. considering that it is ranked higher the model will train genre ahead of year.

Year has high interaction potential with both but the variable ranking calculations reveal high redundancy, therefore an interaction may have very limited application. This is further demonstrated by the Boruta analysis where year was a far distant feature in the film effects model.

Given these considerations the expected model form is:

$$rating = b_{0,intercept} + b_1 * user + b_2 * genre + b_3 * user * genre + b_4 * year + b_5 * user * year + b_6 * film$$

Model Training

The model will be trained using quasi-stepwise ridge regression. As a starting method the full model will be trained and backwards selection will be applied at the step where RMSE yields small and irrelevant or no improvements.

Analytical Solution

Given the size of the dataset the model will be trained using sequentially applying analytical solutions to Ordinary Least Squares (OLS) with the L2 penalization of ridge regression applied after the coefficients have been calculated.

Numeric predictor OLS closed form solution

The simple application of ordinary least squares aims to minimize the sum of square differences between the predicted and observed values. Mathematically the Loss function is given as:

$$L_{OLS} = \sum (y_i - \alpha * x_i)^2$$

When the derivative of this function is calculated for a variable α and it is equal zero, the slope being equal to zero, the α which minimizes the loss is calculated.

$$L_{OLS} = \sum (y_i - \alpha x_i)^2 L'_{OLS} = \sum 2(y_i - \alpha x_i)(-x_i) L'_{OLS} = -2*(\sum x_i y_i - \alpha \sum x_i^2) \therefore \sum x_i y_i = \sum x_i^2 \therefore \hat{\alpha} = \frac{\sum x_i y_i}{\sum x_i^2}$$

Categorical predictor & Intercept OLS closed form solution

The closed form solution to ordinary least squares is based on a single categorical value application of the matrix approach to OLS with no intercept calculation. The Matrix approach to OLS establishes:

$$Y = X\beta + \epsilon$$

is equivalent to:

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix} \begin{bmatrix} \beta_1 & \beta_2 & ... & \beta_n \end{bmatrix}$$

Where β can be solved by:

$$\hat{\beta} = (X'X)^{-1} X'Y$$

For a categorical predictor X can be defined as:

$$\begin{bmatrix} 1_1 \\ 1_2 \\ . \\ . \\ . \\ 1_n \end{bmatrix}$$

Then $X'X$ can be solved a:

$$X'X = \sum_{n=1}^{n} 1 = \begin{bmatrix} n \end{bmatrix}$$

The Inverse of $X'X$ is given by:

$$\because AB = I \Rightarrow \begin{bmatrix} a \end{bmatrix} \begin{bmatrix} b \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix} \ \begin{bmatrix} ab \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix} ab = 1 \therefore b = \frac{1}{a} \begin{bmatrix} n \end{bmatrix}^{-1} = \frac{1}{n}$$

The operation $X'Y$ is calculated by:

$$\begin{bmatrix} 1_1 & 1_2 & ... & 1_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} = \begin{bmatrix} 1_1 y_1 + 1_2 y_2 + ... + 1_n y_n \end{bmatrix} \Rightarrow \begin{bmatrix} y_1 + y_2 + ... + y_n \end{bmatrix} \Rightarrow \sum_{n=1}^{n} y$$

Finalizing as:

$$\beta = \frac{1}{n} \sum_{i=n}^{n} y = \bar{y}$$

The closed for solution for OLS of a single categorical variable can be calculated as the mean value of the response for that category.

The same methodology can be applied to calculating the intercept as the matrix approach which calculates an intercept implies an X vector where every value is 1.

Sequential Training

The sequential modeling will be applied in the following fashion:

$$\hat{Y_0} = b0\hat{Y_i} = \hat{Y_0} - b_0 - b_{i-1}x_{i-1}$$

Every time a new coefficient is trained and regularized the following coefficient will be trained on a response calculated from subtracting the calculated predicted values.

Ridge Regression L2 Penalty Terms

Ridge regression penalty terms will be applied to the test set in order to determine which value minimizes RMSE.

The L2 term will be applied to a trained coefficient in the form of:

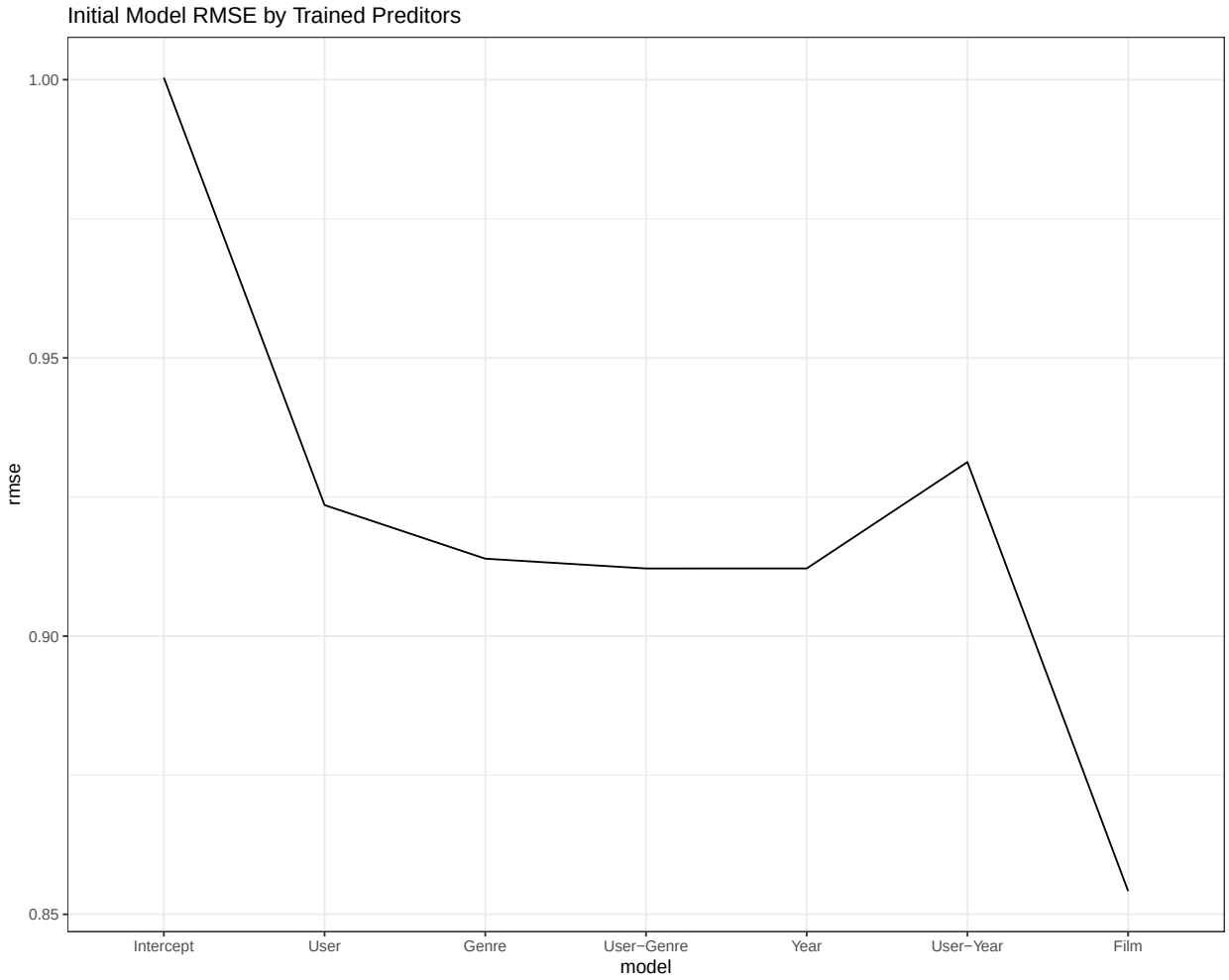
$$L' = min \sum_{u,i} (y_{u,i} - b_i)^2 + \lambda \sum b_i^2 \therefore \hat{b_i}(\lambda) = \frac{1}{n + \lambda} \sum_{u=1}^{n_i} (Y_{u,i} - b_i) \Rightarrow \frac{1}{n + \lambda} \hat{Y_i}$$

Whenever $L2 = 0$ the regression coefficients will remain as estimated via OLS.

Scaling and Centering

Numeric data was onl;y scaled and centered therefore the mean and standard deviation of the training set can be used to revert the process and produce interpretable results of the model output.

Initial Model

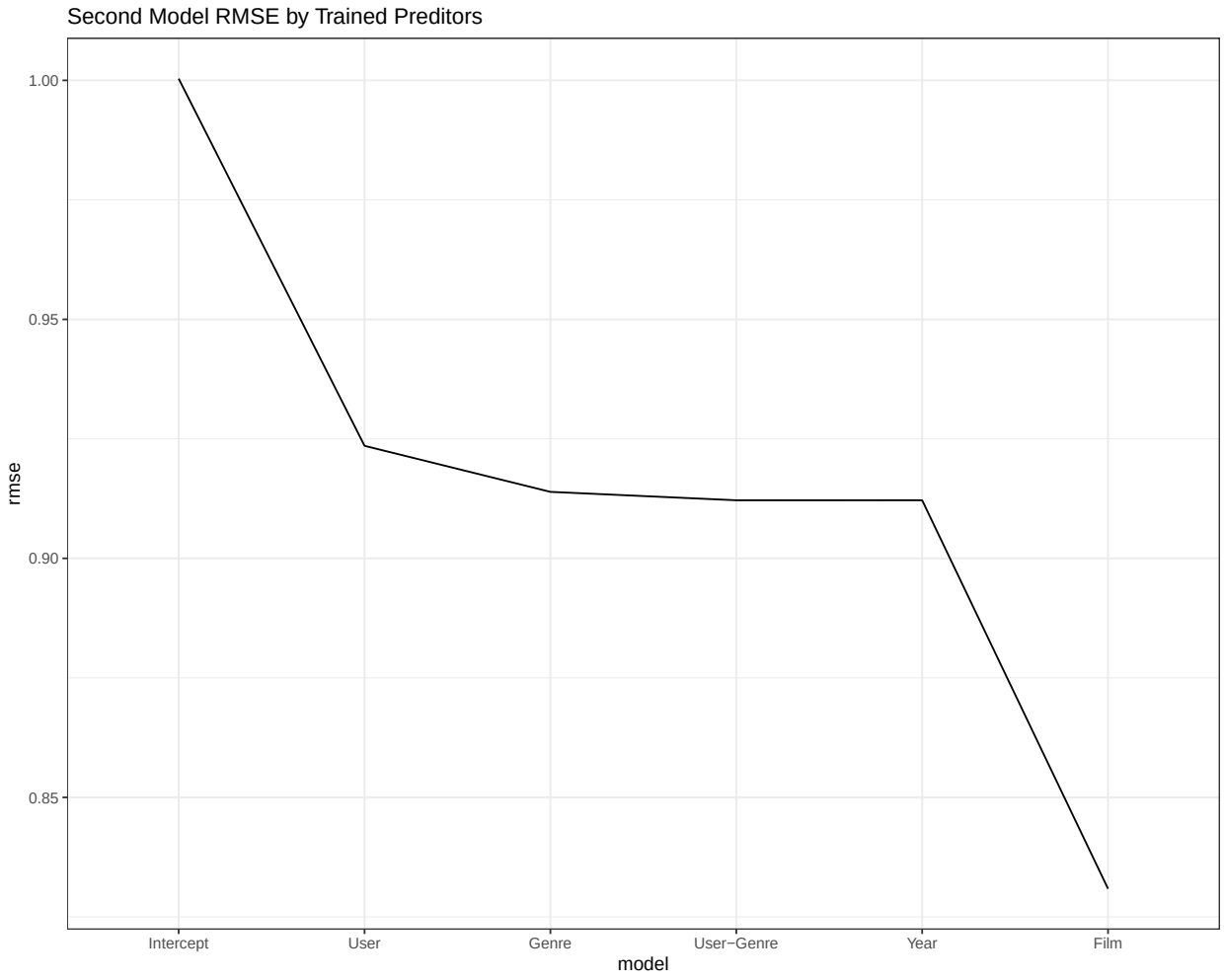


Model	RMSE	RMSE Difference
Intercept	1.0003586	NA
User	0.9235438	0.0768148
Genre	0.9139119	0.0096319
User-Genre	0.9121529	0.0017590
Year	0.9121571	-0.0000042
User-Year	0.9312753	-0.0191182
Film	0.8541669	0.0771084

The ridge regression model progressively exhibits reduction in RMSE up until the model train $b_4 * year + b_5 * user * year$. The model will regress and train $b_6 * film$ without training $b_5 * user|year$, year will be left

as is for the second iteration.

Second Model

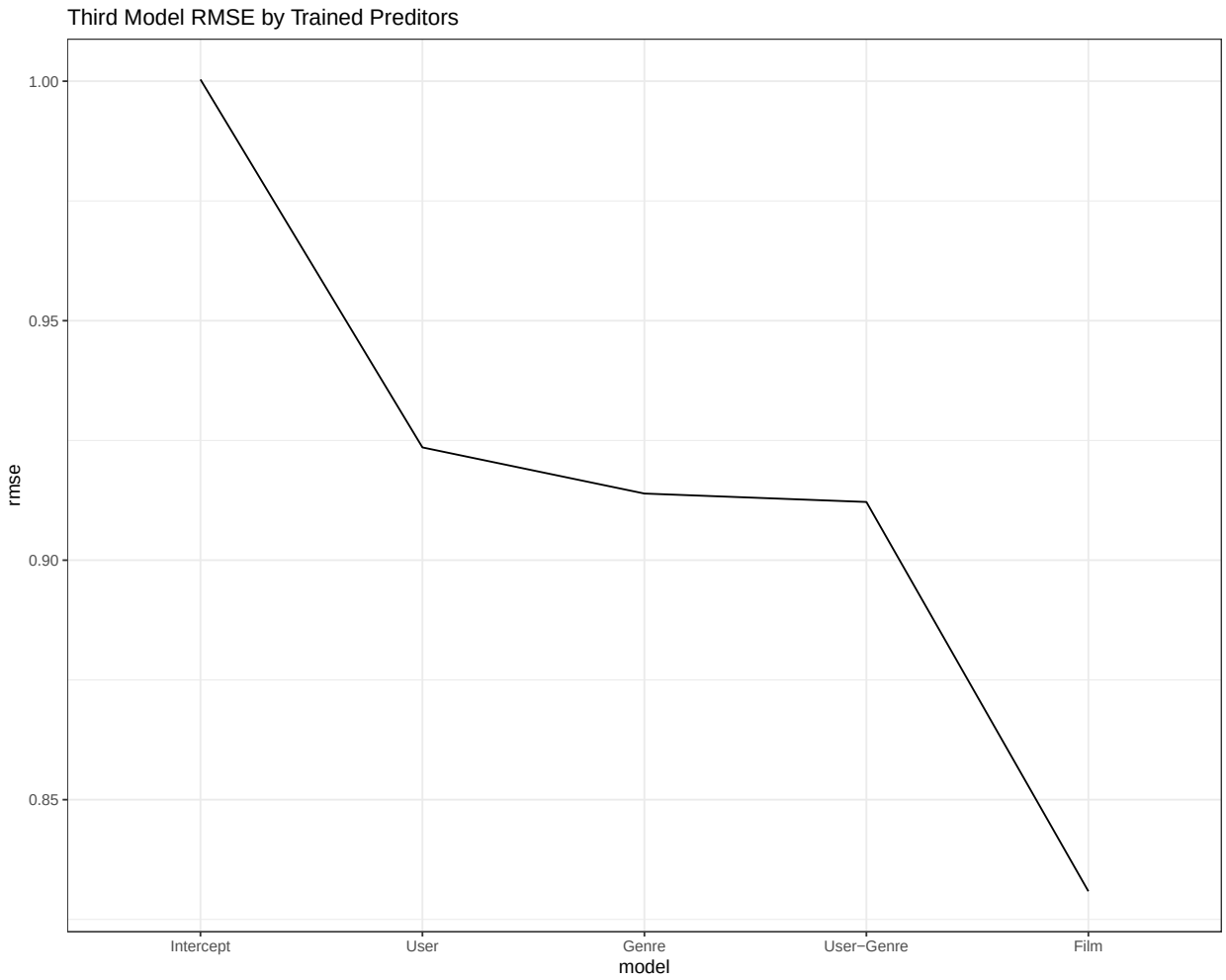


Model	RMSE	RMSE Difference
Intercept	1.0003586	NA
User	0.9235438	0.0768148
Genre	0.9139119	0.0096319
User-Genre	0.9121529	0.0017590
Year	0.9121571	-0.0000042
Film	0.8308783	0.0812788

The second model shows an improvement with a Final RMSE of `r``round(model_b_rmse$rmse[6],2)` . However, Year may does demonstrate slight negative effects in the model. The predictor will be removed and film coefficients will be retrained in order to gauge the effects of this removal on model RMSE.

Model	RMSE	RMSE Difference
Intercept	1.0003586	NA
User	0.9235438	0.0768148
Genre	0.9139119	0.0096319
User-Genre	0.9121529	0.0017590
Film	0.8308783	0.0812746

Third Model



There was no change in final RMSE with the removal of Year as a predictor. The final Test set RMSE stands at 0.83 . This will be considered the final model.

The only predictor to require L2 penalization was $user * genre$ with an L2 of 4.7. This feature exhibited collinearity with the previously trained features and the reduction in coefficients with the application of L2 minimized this effect.

The final model form is:

$$rating = b_{0,intercept} + b_1 * user + b_2 * genre + b_3 * user * genre + b_4 * film$$

Holdout Set Prediction

The RMSE of the Holdout set utilizing the final model is calculated as:

Final RMSE
0.830689

The final model RMSE for both the test set and the final holdout set are essentially equal at 0.83.

Since the numeric variables were scaled and centered the mean and standard deviation of the training set can be utilized to revert the scaling and centering in order to have interpretable results for the model output.

Reversed Rating	Rating
3.523496	-0.4833351
3.995128	0.4598339
3.980486	0.4598339
3.218534	0.4598339
4.857092	1.4030029
2.863712	-1.4265042
3.643882	1.4030029
4.119401	1.4030029
2.945934	0.4598339
3.532608	0.4598339

Conclusions

The final model demonstrates that predicting rating are mostly dependent of both user and user effects with user genre taste adding some improvements to the predictive power of the model. Feature selection methods demonstrated that year and year interaction may have some effects on prediction but model training removed both predictors. This may stem from the year effects being inherently encoded into user effects and film effects due to the date of the review. Additional post-hoc studies may show insights into the actual effects.

Limitations & Future State

The method of prediction is limited in future predictions given the lack of time progression in overall ratings by user. Training based on time slice partitions and the use of running means may yield better predictions than the current model.