Group 15

MGSC 661 Midterm Project

Students: Diwei Zhu, Oleg Kartavtsev, Yichen Wang, Jeewon Kim, Senan Agblonon, Nathan Murstein

November 2021

1 Introduction

The purpose of this project is to develop a predictive model that projects the critic rating (variable name: imdb_score) of the movie based on the IMDb (an online database for all film-related information) dataset. The dataset, with 46 predictors, contains various movie attributes such as the budget of the movie, its duration, and genre, allowing for data analysis and model building. At the completion of the project, the team will use the final model to predict the IMDb rating of 40 blockbusters with the expectation of testing the accuracy and usefulness of the model.

If the accuracy of the model is confirmed, an indicating tool that helps movie producers with decision-making procedure on which metrics should the movie include. For example, if the model indicates that a horror movie with a budget that is higher than 1 million dollars is to be a high-rated movie, the movie producers will be inclined to the specific type of movies.

In order to develop a model with the sufficient predictive power, we explored each variable individually as well as the relationships among the variables, removed outliers, tested and corrected heteroskedacity, introduced non-linearity into the model, and tested linear and polynomial spline models. We compared the models in terms of R^2 scores and MSE scores to find the best performer, then tested out-of-sample performance to reduce the chance of overfitting that would give us a wrong result at the final stage of the project.

As a result, our complete model is able to predict 36% of the observations within the dataset, which gives us a good chance of predicting the IMDb ratings of the 40 "unnamed" movies.

2 Data Description

In order to develop a good predictive model, we first need to analyze each variable individually and then in conjunction with each other, looking at their distribution, outliers and other factors that might be influencing them.

First, we get a feel for the data by identifying the numeric characteristics. In this dataset, we have 11 characteristics that we are going to start with before deciding which ones are going to be excluded from the model regression. The characteristics are: imdb_score (target variable), budget_in_millions, month_of_release,

year_of_release, duration_in_hours, total_number_languages, total_number_of_actors, total_number_of_directors, total_number_of_producers, total_number_of_production_companies, total_number_of_production_countries. After defining the numeric columns that we are going to work with, we develop a boxplot and histogram for each one in order to visualize the distribution and the outliers.(refer to Appendix A - Fig 1) For example, when we look at the distribution of total number of actors in a movie, it becomes clear that the data is heavily skewed to the left and there are many outliers that will have to be eliminated for us to build a good model. This helps us understand the data and identify any possible issues.

2.1 Non-linearity Problem

Before building the model, we need to explore the dataset to find potential problems that could affect the accuracy of our model. In the data exploration part, we will mainly focus on non-linearity, collinearity and outlier problems.

The first problem is non-linearity. By building a simple linear regression model and doing a residual plot, we could observe which numerical predictor is non-linear in shape. After checking all the residual plots, we found that most of our numerical predictors (year_of_release, budget_in_millions, duration_in_hours, and total_number_of_actors) are nonlinear, and this indicates that we should adopt non-linear regression model instead of the simple linear regression to improve reliability. **(Refer to Appendix B - Fig 2)**

2.2 Collinearity Problem

When exploring the dataset, we also investigated the 'collinearity problem' among predictors. To see whether a collinearity problem exists or not, we generated a correlation matrix between given numeric predictors. From the matrix, we found that the highest correlation lies between total_number_of_production_companies and total_number_of_production_countries, and the value of the coefficient is 0.44. For the negative correlation, the lowest correlation lies between genre_thriller and genre_comedy, and the coefficient value is –0.31.

When we consider the rule of thumb to handle the collinearity problem, it can be problematic only the absolute value of coefficients exceeds 0.8. As a result, our team concluded that the collinearity problem does not exist in the given dataset, so no further fix is required. **(Refer to Appendix B - Fig 3)**

2.3 Outlier test

We decided to check if there are any observationsoutliers present in the dataset and remove them. Outliers in data can distort predictions and affect the accuracy in our regression model.

We did a numerical test (Bonferroni Test) using the function oultlierTest from the "car package" which gives the most extreme observations based on the given model.

When we ran the test, for most of the predictors (eg: "duration_in hour, genre_action, genre_animation, genre_family, genre_history, main_actor1_is_female, genre_drama, budget_in_millions", etc..), we got the same output suggesting that observations in rows 633,895 are the most extreme that need to be removed. Similarly, for the predictors "genre_comedy, total_number_of_actors" the R output suggested that observations in rows 633, 895, 2045 are the most extreme; and for the predictor "year of release", the following observations 633,895,2310,2718,2045,526 are the most extreme ones.

As we could observe, most of them have the same row observations in common. Then, moving forward we created a vector containing the row numbers (575, 633, 895, 2310, 2718, 2045,526) and extracted them for the dataset to get a new dataset.

3 Model Selection

3.1 Feature Selection

Before starting to build the model, we have to select the subset of the most relevant features for the given dataset to obtain higher predictive power. Our predictive model can be misled by irrelevant input variables, resulting in more inferior performance. Also, fewer features allow the model to run more efficiently in terms of computational resources and time complexity.

To identify irrelevant features, we utilized the Recursive Feature Elimination tactic. It works by defining 'incremental R-squared' per each predictor by starting with all features in the dataset and removing one predicter from the initial model by one iteration.

Now we will look at using the Recursive Feature Elimination procedure for our

regression model as below:

Step 0: Before conducting analysis, we dummified possibly relevant categorical variables – "main_lang" and "main_production_country" to enhance predictive power.

Step 1: First of all, we figured out the value of 'benchmark R-squared' as 0.354, which refers to the R-squared value of multiple regression model containing all numeric features in the dataset. It will be benchmark R-squared, compared to R-squared values found in the iteration of removing each predictor in the following steps.

Step 2: Then we built the for-loop, and performs 121 iterations which is the number of all numeric predictors we initially incorporated. Per each predictor removed for the iteration, we stored the output value of R-squared from the model summary.

Step 3: Based on the result dataframe of the previous step, we could successfully find out 'incremental R-squared' when we incorporate the predictor in our model, deducting R-squared without the predictor from benchmark R-squared.

Moreover, to validate the result of Recursive Feature Elimination, we additionally conducted feature selection of Random Forest for the given dataset. This analysis found that budget_in_millions and genre_drama are also essential to achieve decent predictive power, which is consistent with the team's intuition.

In conclusion, our team decided to incorporate ten variables found from Recursive Feature Elimination and two variables found from Random Forest Feature Selection to build the predictive model.

(Refer to Appendix B - Table 2)

3.2 Polynomial Regression Model

From the residual plots, we detected non-linearity of 4 numerical variables (year_of_release, budget_in_millions, duration_in_hours, and total_number_of_actors) that are all features with predictive powers as suggested by the feature selection section. Therefore, the dataset reveals the need to use polynomial regression to increase the predictive power of the model. The 4 numerical predictors are transferred into non-linear terms.

To determine the best degree for each predictor, we decided to apply ANOVA test. To ensure the efficiency and accuracy of the ANOVA test, 18 polynomial regression models with various degrees were constructed in order to minimize the effect brought by variable change. We first kept all predictors at degree = 1, and then increment the degree for each numerical predictor. The degrees of other numerical predictors are kept the same and these models were included in the ANOVA function. By checking the decreasing pattern in RSS while keeping our Pr(>F) smaller than 0.05, we could observe the desired degree for our polynomials.

After walking through the 4 predictors, we came up with a polynomial model in which year_of_release has a degree of 3, duration_in_hours has a degree of 4, total_number_of_actors has a degree of 2, and budget_in_millions has a degree of 4. That is,

$$\begin{aligned} Imdb\_score = {} & 7.01 - 6.66 \times year\_of\_release + 1.92 \times year\_of\_release^2 \\ & + 1.51 \times year\_of\_release^3 + 15.74 \times duration\_in\_hours \\ & - 5.50 \times duration\_in\_hours^2 - 0.93 \times duration\_in\_hours^3 \\ & + 2.37 \times duration\_in\_hour^4 + 7.74 \times total\_number\_of\_actors \\ & - 4.98 \times total\_number\_of\_actors^2 - 8.32 \times budget\_in\_millions \\ & + 8.36 \times budget\_in\_millions^2 - 5.00 \times budget\_in\_millions^3 \\ & + 1.99 \times budget\_in\_millions^4 - 0.64 \times genre\_horror \\ & - 0.40 \times genre\_action - 0.29 \times genre\_comedy \\ & + 0.70 \times genre\_animation - 0.30 \times genre\_family \end{aligned}$$

We were able to achieve a MSE of 0.566 using 5-fold cross validation.

(Please refer to Appendix B - Fig 4 for the polynomial regression output)

3.3 Spline Model

Once we obtained the predictors, built a baseline model, and experimented with polynomial models, we wanted to see if we can improve the performance of our model using linear and polynomial spline regression. The first task was to optimize the number of linear splines/knots that minimized MSE. We first decided that we cannot intuitively place splines based on knowledge of the shape of the data, since visualizing so many variables in a hyperplane and visually estimating optimal spline

locations would be impossible. Therefore, we defined a function that outputs the locations of splines by percentile given the number of splines desired and a column of a data frame. We then used this function and a loop to try out between 2 and 5 splines, fitting models and testing them using 5-fold cross validation. We used 5-fold cross validation because in the next step (testing polynomial splines), 10-fold cross validation was taking excessively long, and we wanted a standardized value of K for all cross-validation tests to allow for comparison. We obtained the minimum MSE of 0.592 with 5 splines (4 knots).

Knowing that 4 knots was the optimal number, we next wanted to see if we could further decrease MSE by introducing polynomials. We considered the tradeoff of interpretability that comes with using polynomials - but in this case, since we had already used splines and accuracy was the priority, we decided to try polynomial splines. First, we defined a matrix with all possible combinations of polynomial degrees 2 and 3 for all 13 variables, where each row had 13 columns for polynomial degrees of each predictor. Initially, we wanted to test polynomial degrees between 2-5. However, while the matrix of combinations for degrees 2-3 has 8192 rows, the matrix of combinations for degrees 2-5 has over 67 million, which massively increases computation time. Possible future improvements of this project could use grid search and/or random search algorithms to more efficiently optimize these parameters; however, in this scenario we opted to simply narrow our search to polynomial degrees 2 and 3. We then used a loop to fit a polynomial spline model with 4 knots using each combination of polynomial degrees and evaluated MSE using 5-fold cross validation. We obtained an optimal MSE of 0.586 using the polynomial degrees indicated in the following table and 4 knots, placed evenly by percentile (Table 1).

Because 0.586 was a higher MSE than the MSE achieved using polynomial regression without splines, we settled on the polynomial regression model without splines as the optimal model.

3.4 Heteroskedastic Test for our Optimal Model

To detect heteroskedasticity in our model, we ran a nonconstant variance test and evaluated P-value.

Since p-value from the test is less than 0.05, we assumed that heteroskedasticity is present in our model. Then, we used the "coeftest" function from "lmtest and plm" packages to correct this issue. This transformed the regression standard errors to by

Table 1: Option Polynomial Degrees of Spline Model

Index	Labels	Optimal_degrees
1	best_a	2
2	best_b	3
3	best_c	2
4	best_d	2
5	best_e	3
6	best_f	3
7	best_g	2
8	best_h	3
9	best_i	3
10	best_j	3
11	best_k	3
12	best_l	2
13	best_m	2

eliminating heteroskedasticity from our model. From the output, we could clearly see that the P-values have changed.

(Please refer to Appendix B - Fig 5 and Fig 6)

4 Managerial Implications

To sum up the insights that we derived from this project, we must interpret the predictive model that we built. This carries challenges because we used nonlinear regression and interpreting the coefficients for each polynomial degree in the regression summary is difficult. However, we visualized each variable's effect by creating simple regressions for each numeric predictor, in which there was only one predictor. We then visualized the fit of these regression lines to roughly estimate whether the managerial effects of the predictors were positive or negative to generate recommendations/implications.

In conclusion of our project, we can derive that the key metrics that influenced our model's predictive power were year of release, duration, genre of the movie and the total number of actors (listed in the descending order, using recursive features elimination.) The movie ratings are slightly going down as the year of release increases, but this will not be useful for movie makers. Duration of the movie is highly correlated with the rating, with the longer movies receiving higher ratings. Genre is also very important. For example, horror and action movies are more likely to be rated lower, while drama and history movies are correlated with higher rankings.

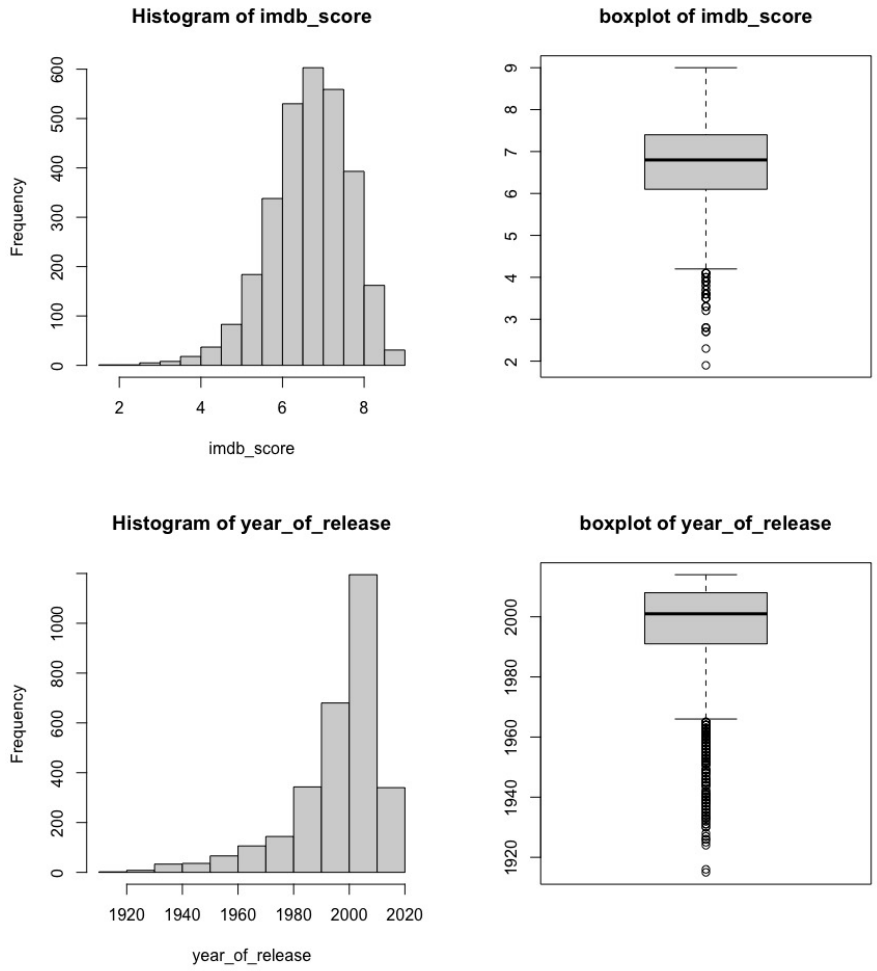
Consequently, movie studios could invest more in those genres in an effort to increase ratings. Finally, total number of actors is also important, with higher number of actors correlating with the higher IMDb scores.

In the future, if given more time to work on this project, we would recommend looking into interaction terms before predictors to determine whether the factors that make a highly scored movie successful are constant across genres. For example, does increasing duration have the same effect on the critic score for action movies that it does on comedy movies? Increasing the level of granularity using interaction terms is an interesting next step for this project.

A This is an appendix

Appendix A - data description

Figure 1: Histograms and Boxplots Examples



Appendix B - model selection

Figure 2: Non-Linearity (Residual Plots)

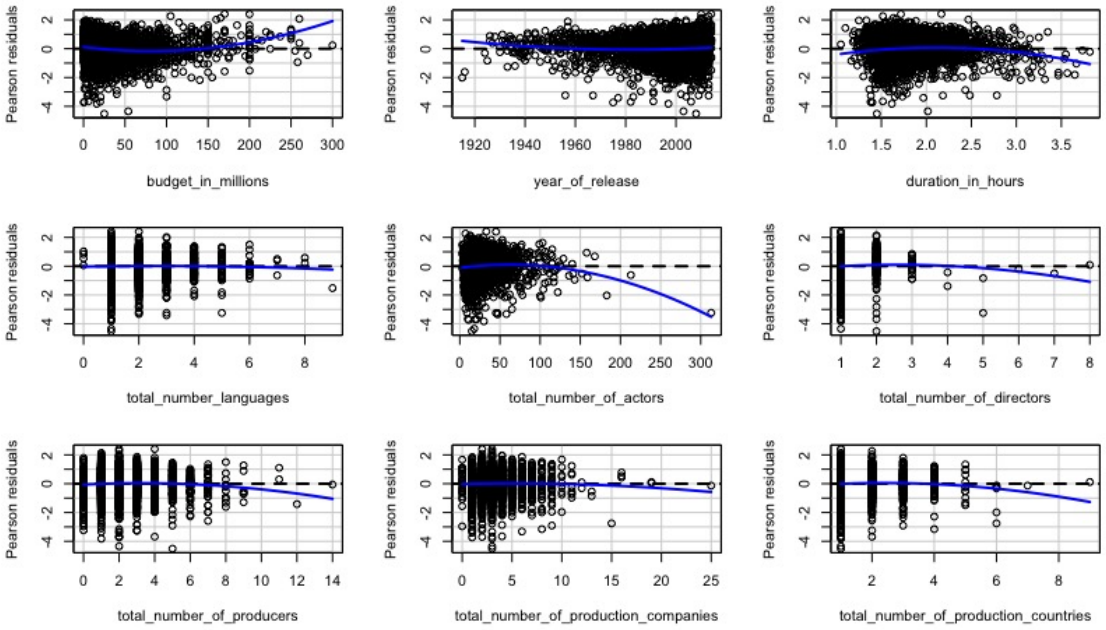


Figure 3: Collinearity Problem (Correlation Matrix)

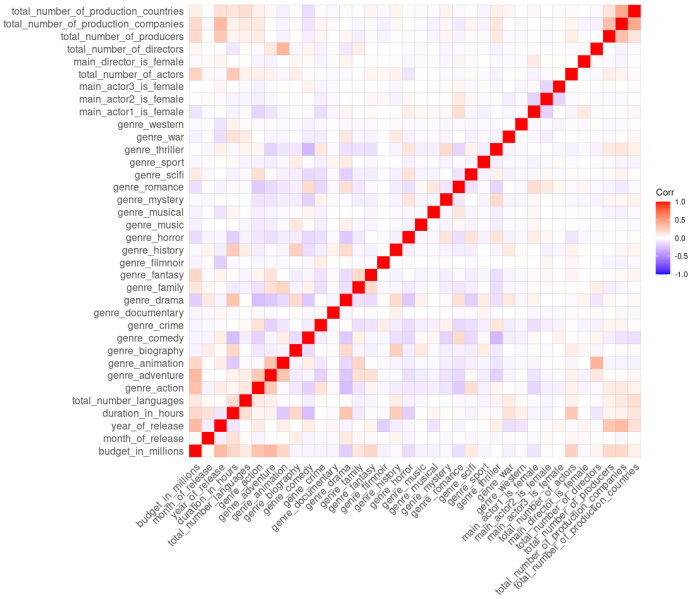


Table 2: Top 10 features selected by Recursive Features Elimination

Index	Predictor removed	R-squared without predictor	Incremental R-squared
3	year_of_release	0.3176912	0.0359775
4	duration_in_hours	0.3282065	0.0254622
18	genre_horror	0.3306731	0.0229956
6	genre_action	0.3327779	0.0208908
33	total_number_of_actors	0.3332562	0.0204125
10	genre_comedy	0.3420516	0.0116171
8	genre_animation	0.3426061	0.0110626
14	genre_family	0.3457035	0.0079652
30	main_actor1_is_female	0.3459894	0.0076793
17	genre_history	0.3506194	0.0030493

Figure 4: Polynomial Regression Model Output

```
Call:
lm(formula = imdb_score ~ poly(year_of_release, 3) + poly(duration_in_hours,
    4) + poly(total_number_of_actors, 2) + poly(budget_in_millions,
    4) + genre_horror + genre_action + genre_comedy + genre_animation +
    genre_family + genre_history + main_actor1_is_female)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2261 -0.4058  0.0629  0.4898  2.1588

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      7.01087    0.02657 263.895  < 2e-16 ***
poly(year_of_release, 3)1       -6.66265    0.93919  -7.094 1.63e-12 ***
poly(year_of_release, 3)2        1.92123    0.80452   2.388   0.0170 *
poly(year_of_release, 3)3        1.51453    0.80058   1.892   0.0586 .
poly(duration_in_hours, 4)1     15.74089    0.97063  16.217  < 2e-16 ***
poly(duration_in_hours, 4)2     -5.49791    0.82852  -6.636 3.83e-11 ***
poly(duration_in_hours, 4)3     -0.93005    0.79557  -1.169   0.2425
poly(duration_in_hours, 4)4      2.36581    0.78479   3.015   0.0026 **
poly(total_number_of_actors, 2)1  7.74034    0.83380   9.283  < 2e-16 ***
poly(total_number_of_actors, 2)2 -4.97904    0.78295  -6.359 2.34e-10 ***
poly(budget_in_millions, 4)1    -8.32904    1.02175  -8.152 5.25e-16 ***
poly(budget_in_millions, 4)2     8.35692    0.84285   9.915  < 2e-16 ***
poly(budget_in_millions, 4)3    -4.99526    0.81412  -6.136 9.61e-10 ***
poly(budget_in_millions, 4)4     1.99161    0.80178   2.484   0.0130 *
genre_horror                    -0.63740    0.05102 -12.494  < 2e-16 ***
genre_action                    -0.39682    0.03693 -10.746  < 2e-16 ***
genre_comedy                    -0.29478    0.03394  -8.685  < 2e-16 ***
genre_animation                  0.70084    0.07771   9.019  < 2e-16 ***
genre_family                    -0.30127    0.06251  -4.819 1.51e-06 ***
genre_history                   -0.10860    0.07796  -1.393   0.1637
main_actor1_is_female           -0.20291    0.03548  -5.718 1.18e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7708 on 2932 degrees of freedom
Multiple R-squared:  0.3674,    Adjusted R-squared:  0.3631
F-statistic: 85.15 on 20 and 2932 DF,  p-value: < 2.2e-16
```

Figure 5: Visual Test of Heteroskedasticity in our final model:

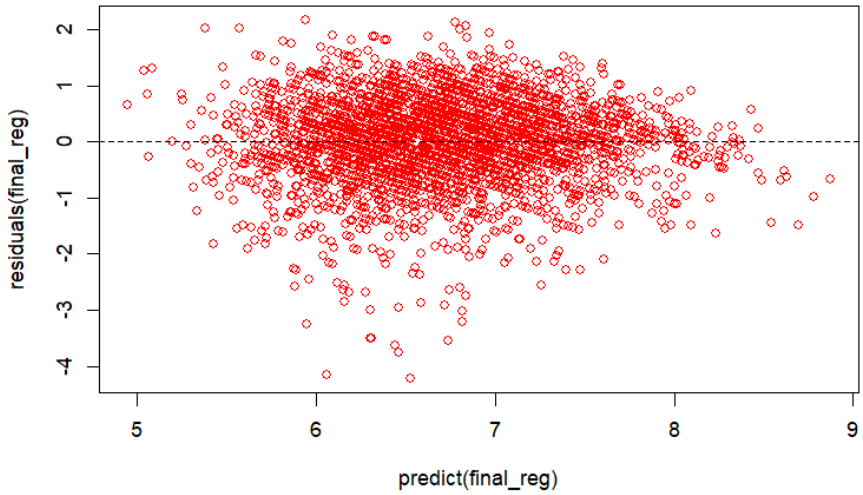


Figure 6: Polynomial model without heteroscedasticity

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	7.010870	0.025427	275.7258	< 2.2e-16	***
poly(year_of_release, 3)1	-6.662652	0.913409	-7.2943	3.846e-13	***
poly(year_of_release, 3)2	1.921231	0.804600	2.3878	0.017012	*
poly(year_of_release, 3)3	1.514527	0.819647	1.8478	0.064735	.
poly(duration_in_hours, 4)1	15.740895	1.066421	14.7605	< 2.2e-16	***
poly(duration_in_hours, 4)2	-5.497913	0.878109	-6.2611	4.382e-10	***
poly(duration_in_hours, 4)3	-0.930049	0.791247	-1.1754	0.239921	
poly(duration_in_hours, 4)4	2.365805	0.764443	3.0948	0.001988	**
poly(total_number_of_actors, 2)1	7.740336	0.863802	8.9608	< 2.2e-16	***
poly(total_number_of_actors, 2)2	-4.979037	0.814522	-6.1128	1.108e-09	***
poly(budget_in_millions, 4)1	-8.329041	1.018103	-8.1809	4.149e-16	***
poly(budget_in_millions, 4)2	8.356918	0.814791	10.2565	< 2.2e-16	***
poly(budget_in_millions, 4)3	-4.995256	0.770778	-6.4808	1.066e-10	***
poly(budget_in_millions, 4)4	1.991605	0.698388	2.8517	0.004379	**
genre_horror	-0.637402	0.060079	-10.6094	< 2.2e-16	***
genre_action	-0.396819	0.037892	-10.4723	< 2.2e-16	***
genre_comedy	-0.294782	0.034976	-8.4281	< 2.2e-16	***
genre_animation	0.700839	0.080227	8.7357	< 2.2e-16	***
genre_family	-0.301274	0.060975	-4.9409	8.212e-07	***