IMDB SCORE PREDICTION

MGSC401 – PROF. SERPA – MIDTERM PROJECT

GROUP #2 – Karen Bou Daou, Radia El Amrani, Zachariya Sow, Alexander Main, Freddy Chen

McGill | DESAUTELS

Introduction

In the modern era, choosing whether or not to watch a movie has become one of the most daunting tasks for individuals. With so many movie and platform choices (entertainment services and cinemas), consumers rely on rating agencies and critics to support their decision-making process. As such, review aggregation websites have substantially gained value with the rise of telecommunications and the Internet. These websites can use a rating system composed of film critics, audience levels, and the public's reviews. This creates pressure on production companies to release movies with better ratings and ensure box office success (depending on their chosen rating system). One of the most popular review websites is IMDb which also serves as a web database of movie information. Our project's aim is to understand the fundamental variables that may affect the rating IMDb reviewers and critics give to a movie and create a model to predict IMDb ratings for subsequent incoming movies. This exercise can create significant value for production companies as they can fund projects that have higher likelihoods of successful ratings, as well as streaming platforms looking to predict which movies to add to their service offering in case of widespread distribution.

Data description

The dataset we received contained a total of 43 variables and 1930 observations. We divided our data analysis into two portions: numerical and categorical data.

Numerical data description

Regarding the numerical variables, we first identified that the following variables were correlated to *imdbScore*: *movieBudget, duration, nbNewsArticles, nbFaces, releaseYear.*

- *imdbScore*: this is our dependent variable (Appendix 1), ranging from 1.9 to 9.3, with a mean of 6.5. These values will be a relevant benchmark to keep in mind when evaluating the performance of our model.
- *movieBudget:* Given the scatter plot we got for the relationship of *movieBudget* with *imdbScore* (Appendix 2), we notice there is heteroskedasticity and that the results are skewed to the right indicating that there is a concentration of movies with a budget between 0-$10,000,000 with other movie budgets skewing to larger values. However,

despite the heteroskedasticity, the variable is significant so we will be looking at residual plots to see if there is non-linearity.

- *Duration:* this is one of the most skewed variables, with a strong concentration of movies being between 100-110 minutes (Appendix 3), which corresponds to the expected average movie times. Considering our short attention spans, people today indeed aren't likely to regularly watch movies for longer than two hours. Looking at the boxplot (Appendix 4), the distribution of the data is inconsistent, and there is a strong concentration of outliers, which can significantly skew our results later on.
- *nbFaces:* The number of faces is positively skewed and displays a funnel shape indicating heteroskedasticity (Appendix 5). This is coherent as movie posters can include up to a certain number of faces before being too crowded or reducing the poster's appeal.
- *releaseYear:* For the release year, the data is highly negatively skewed, with most of the movies recorded on the platform being recent (Appendix 6). However, older movies, although fewer, have higher ratings than newer movies, whose rating range from 2 to 10 compared to 6 to 8 for older movies. As film production becomes more accessible with time, it makes sense that more movies are being produced, without them being necessarily of high quality. Looking into the boxplot (Appendix 7), the average release year is 2001, however, there is a strong concentration of outliers between 1960 and 1980, which might significantly skew results later on.
- *nbNewsArticles:* This variable's distribution is positively skewed, with an average number of 770 articles written per movie (Appendix 8). A striking observation would be some significant outliers, with the max amount being 60,620 articles. This might significantly skew results later on and must be accounted for, and potentially removed.

Categorical data description

Regarding the categorical variables, we created a barplot with the genre variables (including *adventure*, *action*, *scifi*, *thriller*, etc.) (Appendix 9) which highlighted that our dataset mostly contained drama, thriller, and romance movies (top 3 in the respected order) suggesting that these might be the most popular movies pursued by the industry (or found in our sample). While this may not hold any indication in the final predictive rating, we can develop an initial hypothesis that these variables will be significant upon their usage in a fitted model. A final note on this distribution is the potential class imbalance issue we might have: more than half the movies contain the drama label which might considerably skew results. We can also add that the distribution of genres was quite unevenly distributed with a range of 990 movies.

As for the release year, we noted a higher amount of movies coming out in January, October and September in the dataset which could be emphasizing two elements: end of the year (post-Christmas period) and back-to-school and work releases as individuals come back from vacation (since we are assuming that people are busier vacationing in the summer to go to cinemas).

We also looked at the number of unique entries, using the unique(var) function for the factors: *country*, *maturityRating* and *language* which were 34, 12 and 19 respectively. These large unique entry numbers are going to make the model weak as they will need a lot of dummy variables making the model non-parsimonious.

Finally, analyzing the maturity ratings, after dummifying these, we can note that most movies in the dataset are R-rated movies, followed by PG-13 and then PG movies (Appendix 10). These are certainly the most common classifications found in daily life.

While there are other remaining categorical variables, we deem them as irrelevant or too extensive to study (e.g., *colourFilm* or *plotKeywords*).

Model selection

The dataset we received contained a total of 43 variables and 1930 observations. To filter variables, we divided our analysis by type: numerical and categorical. Regarding the analysis of the numerical variables, we identified the variables correlated to *imdbScore* which were non-collinear with each other by creating and analyzing a correlation matrix (Appendix 11). At this point, we ran a simple multi-linear regression on all the numerical data and found the significant quantitative variables left at 99.99% significance which were plugged into a regression:

$$\hat{y} = lm(imdbScore\, movieBudget + releaseDay + releaseYear + duration + aspectRatio + nbNewsArticles + actor1_star Meter + actor2_star Meter + actor3_star Meter + nbFaces + movieMeter_I MDBpro + releaseMonth_n um)$$

The findings were as expected from the correlation matrix.

After performing said regression, the summary returned (Appendix 12) enabled us to keep only *movieBudget*, *duration*, *nbNewsArticles*, *nbFaces* and *releaseYear* in the regression as their

P-value were statistically significant at a 99.99% level (*** threshold). Given the high correlation between *nbNewsArticles* and *moviemeter_IMDBpro*, we decided to only keep the former.

After which, we performed a quantile-quantile plot to evaluate our model's data correlation with a normal distribution. This led us to analyze potential outliers through an outlier test (Appendix 13). As a result, we removed observation 492 from our numeric dataset (given its P-value and distance in the qqplot). This increased the current linear models R-square by around 0.0266%. Observation 492 is Star Wars: Episode IV - A New Hope. This film had 60,620 articles written about it with the next highest observation being 16,092. Needless to say, this ground-breaking film was an outlier in our data set with a high deviation from the mean.

The final step was to run a residual plot to verify any heteroskedasticity issue in the data's shape (Appendix 14). To confirm the trend observed in the plot, we ran an NCV test (Appendix 15) and a coefficient test (Appendix 14) for our regression variables to determine if P-values were significant. Despite the heteroskedasticity in the data, the P-values were indeed significant. To add to this error check, we went back to the correlation matrix (Appendix 11) to make sure none of our numerical variables were too highly correlated.

Regarding the categorical variables, we dummified the different maturity levels and combined these ratings with the film genres. This enabled us to understand which variables were significant and filter through those that would be dropped from our final model after running multiple regressions. First, we found what categories were significant by regressing all categories onto IMDBscore revealing action, romance, horror, drama, and war to be significant.

Then we created a new linear regression with these variables alongside the numerical variables and both *country* and *language*. Given the model summary, we see that *country* and *language* were insignificant, so we removed them. Afterwards, we ran a linear regression with solely the filtered maturity ratings. Then we dummified and kept the most important ones to combine them to the significant genres in an updated linear regression.

Then, we added the maturities to the regression and found none of them to be significant at the 99.99% level. We want to maintain the 99.99% rule to keep the model parsimony as there are a lot of variables to choose from at this level; we can therefore afford to be this picky.

After regressing the numerical and categorical variables together, we were left with only seven significant variables at 99.99% post heteroskedastic coef-test (Appendix 12):

$$\widehat{y} = lm(imdbScore\ movieBudget + duration + nbFaces + releaseYear + action + horror + drama)$$

Our next step was to try multiple combinations by drawing scatter plots, running ANOVA test and drawing residual plots to verify and analyze non-linearity in the remaining factors to determine our final predictive model (Appendix 16 & 17). The tukey test indicates that *movieBudget*, *Duration*, *nbNewsArticles* and maybe *releaseYear* are nonlinear and might require splines. We believed releaseYear might be a spline when looking at its residual plot due to the difference of rating over periods of releases. Through this trial-and-error process, we regularly computed MSEs to generally visualize what kind of polynomials or spline would best fit these three variables and optimize their parameters (e.g., degrees). Thus, all non-linear variables were modeled as splines and polynomials.

After comparing the R-square and ANOVA tables of a variety of regressions we decided to fit *duration* with a quadratic relationship and *nbNewsArticles* and *releaseYear* with splines with degrees of 4 and 3, respectively. The splines were created using 4 quantiles of the data (of equal interval lengths). The decision on using a spline on *releaseYear* was made as the date shows older movies followed a different rating trajectory than newer ones.

At this point we needed to cross validate the degree of the polynomials and splines using k-fold cross validation. This was chosen over LOOC as it is much quicker and almost as accurate. Once calibrated, this model led to some terrible predictions (Appendix 18) that were not in the 1-10 range. After trying to remove a variety of the factors we found *movieBudget* was the issue. This is likely due to it containing missing values in almost half of the observations.

The model was then optimized using the same iterative MSE minimizing method without the *movieBudget* as a factor to reveal our final model (Appendix 19):

$$\widehat{y} = glm(imdbScore\ bs(nbNewsArticles, knots = c(k1, k2, k3, k4), degree = 4) + poly(duration, 2) + bs(releaseYear, knots = c(m1, m2, m3, m4), degree = 3) + nbFaces + action + romance + horror + drama)$$

Results

Predicted Ratings

Based on the model described above, we were able to generate the following results for the 12 upcoming movies:

- Falling For Christmas: 8.08
- Black Panther: Wakanda Forever: 9.61
- Spirited: 8.75
- Paradise City: 7.75
- Poker Face: 7.65
- ¡Que viva México!: 8.74
- Slumberland: 8.62
- Blue’s Big City Adventure: 7.81
- The Menu: 7.71
- The Fabelmans: 9.22
- Devotion: 8.97
- Strange World: 8.11

Our model predicts an average rating of 8.42 for the 12 movies, which is above the average rating for the total data used to train the model (mean = 6.51). Based on these results, the upcoming 12 movies will receive high ratings from film critics. Our model’s MSE was 0.749 using a k fold of 250. The high rating is likely due to the high-profile nature of the films given as data to predict.

Distribution Analysis

The movie with the highest predicted rating is Black Panther: Wakanda Forever: 9.6 followed by Devotion: 8.97 and Spirited: 8.75. On the other end, the expected worst performing movie is Poker Face: 7.65 preceded by Paradise City: 7.75. Leaving the range of our prediction at 9.61 - 7.65 = 1.96. Our data is definitely distributed above the mean, but it is predicting a small sample size of 12 movies, and it is not predicting ridiculous high so we deem it acceptable.

Model Summary

Our model’s r-squared is 38.94% (Appendix 20) highlighting that the 38.94% of the variability in the target *imdbScore* variable can be accounted for by our model’s predictive power. Our model utilized one polynomial of degree 2 for duration and two splines of degree 4 and 3 for nbNewsArticles and releaseYear respectively. It included 4 categories action, romance, horror, and drama as well as the linear factor nbFaces. The model is overall quite parsimonious considering the size of the data it was trained on, although degree of 4 is quite high.

Coefficient & Significance Analysis

Based on our model summary shown in the stargazer's chart in Appendix 21, we can note that all of our predictors are significant except the 1st 6th and 8th spline of nbNewsArticles as well as the 1st and 2cnd spline of releaseYear term whose P-value is higher than the 0.05 threshold leaving it insignificant in this model.

Analyzing the remaining predictors, one striking observation is the coefficient value of *duration* as it returns a value equal to 9.744. This can be explained by the related polynomial term *duration* squared which has a value -3.440.

The *nbNewArticles* is an interesting spline as in spline but 8 it is a positive factor. This is hard to extrapolate as a quadratic spline is a complicated equation but it can roughly suggest that recently there have been more news articles about poorly reviewed movies then there has been in the past.

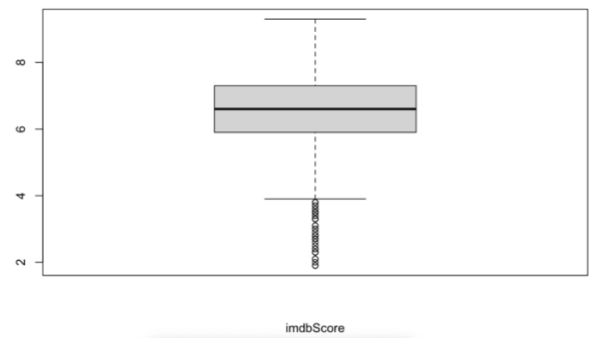
Another interesting observation of the coefficients is that all splines of the releaseYear are negative with the earliest and latest data having the highest negative coefficient. This means recent movies and old movies (bottom 20% and top 20%) are predicted to score the best.

Drama movies are predicted to score 0.460 higher than all other movies category except *action*, *romance*, and *horror* where they will score 0.460 + the negative coefficients of those movies higher. This is because the genre is a binary predictor (1 or 0). This makes sense as drama movies often feature the best acting whereas action, romance or horror are generally tacky, this is statistically shown here.

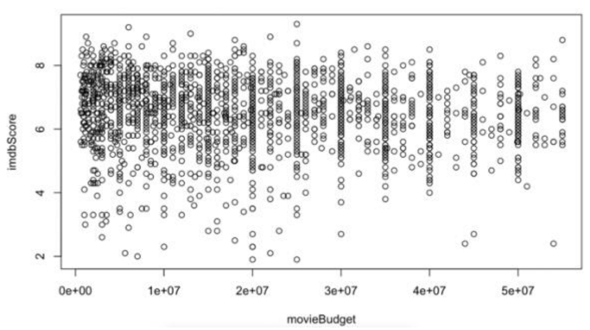
As such, it appears that dramatic movies positively impact the rating of a movie which is consistent with the belief and hypothesis that dramatic movies are more popular (explaining why this segment might be one of the most common ones). This is further instated with its significance at 99%. Using this same logic, *duration squared high* P-value (roughly equal to 0) highlights its significance and predictive power in our model: to that extent, longer movies tend to receive higher ratings which might be true in general, up to a certain limit when the squared term comes in with its negative coefficient. Overall, the coefficients of our model seem rational and have provided predictions for the upcoming films we are happy with.

Appendices

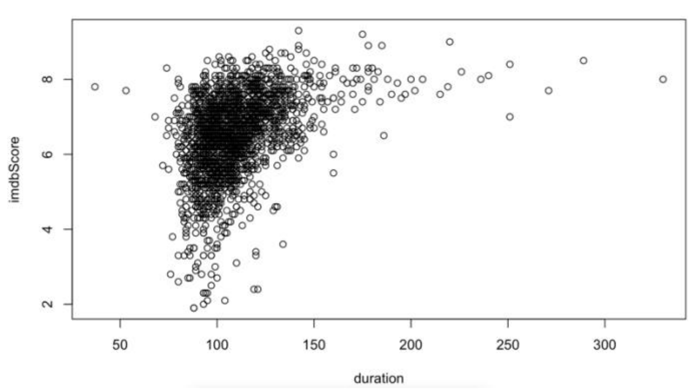
Appendix 1 – IMDB Score variable distribution



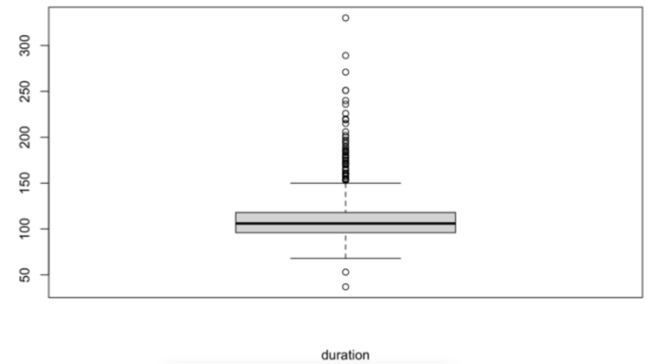
Appendix 2 – Scatter plot between allocated movie budgets and IMDB Scores



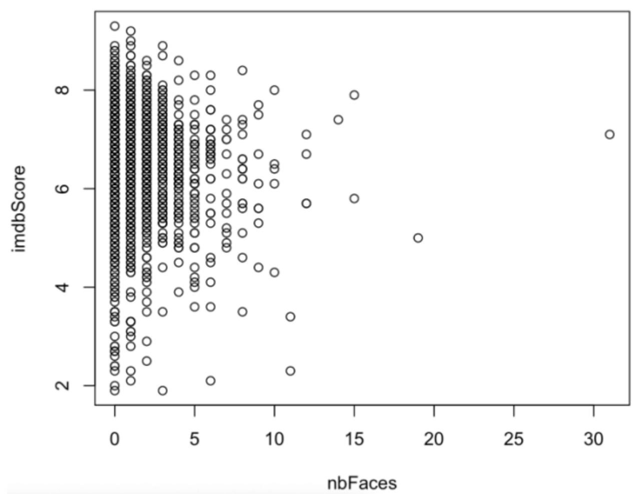
Appendix 3 – Scatter plot of movie by duration



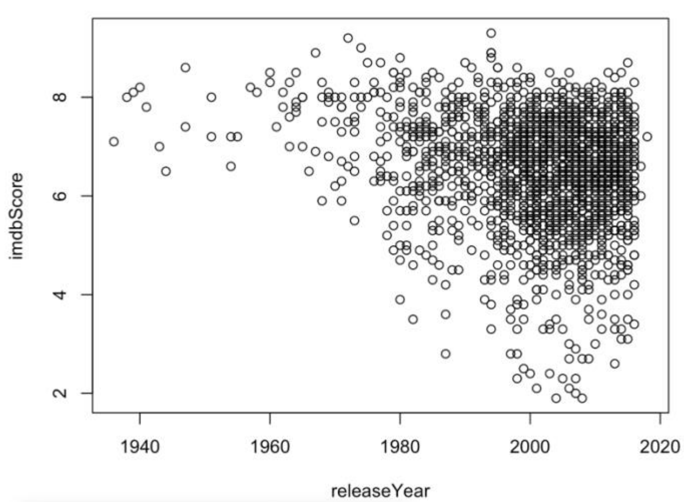
Appendix 4 – Boxplot of movie by duration



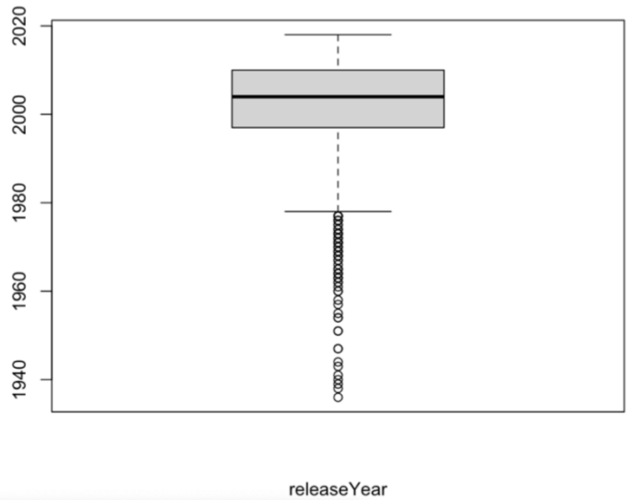
Appendix 5 – Scatter Plot of poster movie faces



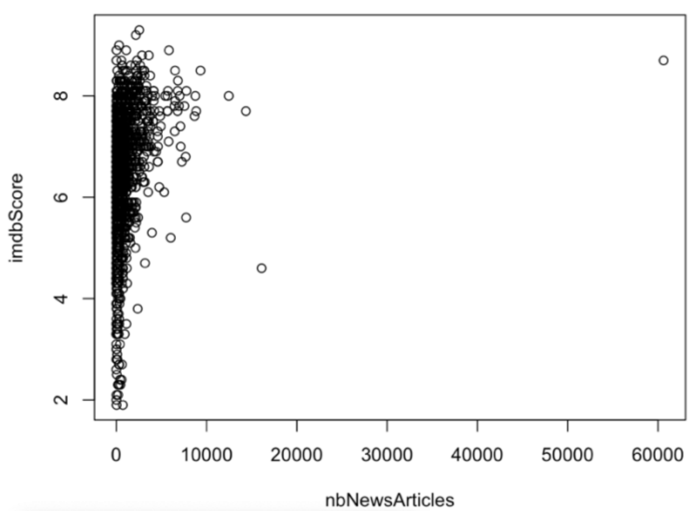
Appendix 6 – Scatter plot of movie by year



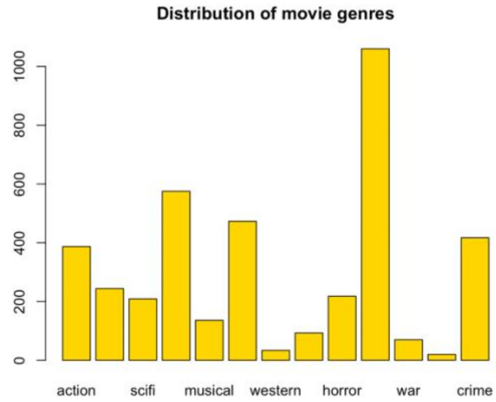
Appendix 7 – Boxplot of movies by year (distribution)



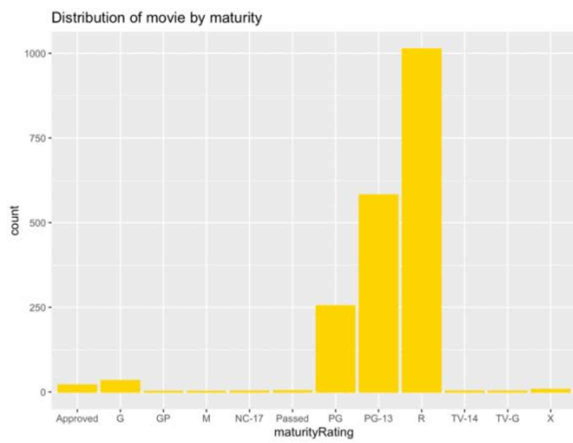
Appendix 8 – Scatter Plot of movies by number of news articles



Appendix 9 – Bar plot for genre



Appendix 10 – Bar Plot for maturity rating



Appendix 11- Correlation matrix of numeric variables

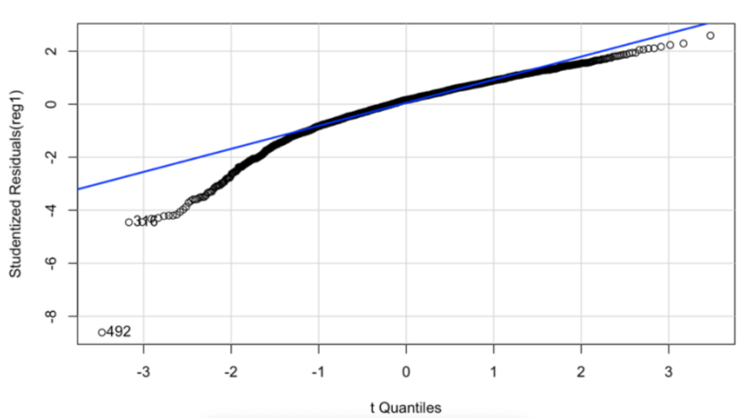
	imdbScore	movieBudget	releaseDay	releaseYear	duration	aspectRatio	nbNewsArticles	actor1_starMeter	actor2_starMeter	actor3_starMeter	nbFaces	movieMeter_IMDBpro	releaseMonth_num
imdbScore	1.000	-0.079	0.021	-0.195	0.411	0.011	0.225	0.029	0.038	-0.004	-0.089	-0.090	0.062
movieBudget	-0.079	1.000	0.021	0.166	0.188	0.232	0.032	-0.019	-0.030	-0.035	0.029	-0.103	0.034
releaseDay	0.021	0.021	1.000	0.004	0.016	-0.025	0.035	0.017	0.007	-0.010	0.021	-0.010	0.026
releaseYear	-0.195	0.166	0.004	1.000	-0.223	0.241	0.062	-0.035	0.017	0.017	0.075	0.041	-0.108
duration	0.411	0.188	0.016	-0.223	1.000	0.099	0.091	-0.004	0.033	-0.008	0.008	-0.058	0.083
aspectRatio	0.011	0.232	-0.025	0.241	0.099	1.000	0.055	-0.051	0.019	-0.013	0.018	-0.003	-0.017
nbNewsArticles	0.225	0.032	0.035	0.062	0.091	0.055	1.000	-0.017	-0.017	-0.029	-0.029	-0.086	0.035
actor1_starMeter	0.029	-0.019	0.017	-0.035	-0.004	-0.051	-0.017	1.000	0.178	0.038	-0.002	0.007	0.007
actor2_starMeter	0.038	-0.030	0.007	0.017	0.033	0.019	-0.017	0.178	1.000	0.299	-0.011	0.042	0.006
actor3_starMeter	-0.004	-0.035	-0.010	0.017	-0.008	-0.013	-0.029	0.038	0.299	1.000	-0.005	0.030	-0.020
nbFaces	-0.089	0.029	0.021	0.075	0.008	0.018	-0.029	-0.002	-0.011	-0.005	1.000	0.002	0.021
movieMeter_IMDBpro	-0.090	-0.103	-0.010	0.041	-0.058	-0.003	-0.086	0.007	0.042	0.030	0.002	1.000	-0.020
releaseMonth_num	0.062	0.034	0.026	-0.108	0.083	-0.017	0.035	0.007	0.006	-0.020	0.021	-0.020	1.000

Appendix 12 – Regression on all numeric data (left), refined factor model (right)

```
Coefficients:
                    Estimate Std. Error t value
(Intercept)        2.063e+01  4.044e+00   5.101
movieBudget       -1.154e-08  1.586e-09  -7.273
releaseDay         1.633e-03  2.626e-03   0.622
releaseYear       -8.178e-03  2.023e-03  -4.042
duration           2.046e-02  1.097e-03  18.646
aspectRatio        8.388e-02  8.452e-02   0.992
nbNewsArticles     1.127e-04  1.186e-05   9.497
actor1_starMeter   9.548e-08  7.749e-08   1.232
actor2_starMeter   1.497e-07  1.340e-07   1.117
actor3_starMeter  -2.123e-08  8.767e-08  -0.242
nbFaces           -4.081e-02  1.061e-02  -3.847
movieMeter_IMDBpro -1.729e-06  5.479e-07  -3.156
releaseMonth_num   5.532e-03  6.262e-03   0.883
                   Pr(>|t|)
(Intercept)        3.71e-07 ***
movieBudget        5.11e-13 ***
releaseDay         0.534122
releaseYear        5.52e-05 ***
duration            < 2e-16 ***
aspectRatio        0.321117
nbNewsArticles      < 2e-16 ***
actor1_starMeter   0.218052
actor2_starMeter   0.263985
actor3_starMeter   0.808687
nbFaces            0.000124 ***
movieMeter_IMDBpro 0.001624 **
releaseMonth_num   0.377116
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9556 on 1917 degrees of freedom
Multiple R-squared:  0.2501,	Adjusted R-squared:  0.2454
F-statistic: 53.26 on 12 and 1917 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = imdbScore ~ movieBudget + duration + nbNewsArticles +
    nbFaces + releaseYear + action + romance + horror + drama)

Residuals:
    Min      1Q  Median      3Q     Max
-5.5416 -0.4177  0.1223  0.5782  2.4736

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.236e+01  3.889e+00   8.320  < 2e-16 ***
movieBudget    -6.473e-09  1.557e-09  -4.158 3.35e-05 ***
duration        1.480e-02  1.148e-03  12.889  < 2e-16 ***
nbNewsArticles  1.251e-04  1.133e-05  11.038  < 2e-16 ***
nbFaces        -4.432e-02  1.032e-02  -4.295 1.84e-05 ***
releaseYear    -1.372e-02  1.929e-03  -7.112 1.61e-12 ***
action         -3.117e-01  5.610e-02  -5.557 3.13e-08 ***
romance        -1.741e-01  5.054e-02  -3.444 0.000584 ***
horror         -3.554e-01  7.143e-02  -4.975 7.11e-07 ***
drama           4.275e-01  4.927e-02   8.677  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9164 on 1920 degrees of freedom
Multiple R-squared:  0.3093,	Adjusted R-squared:  0.3061
F-statistic: 95.54 on 9 and 1920 DF,  p-value: < 2.2e-16

>
```

Appendix 13 – QQPlot of linear model



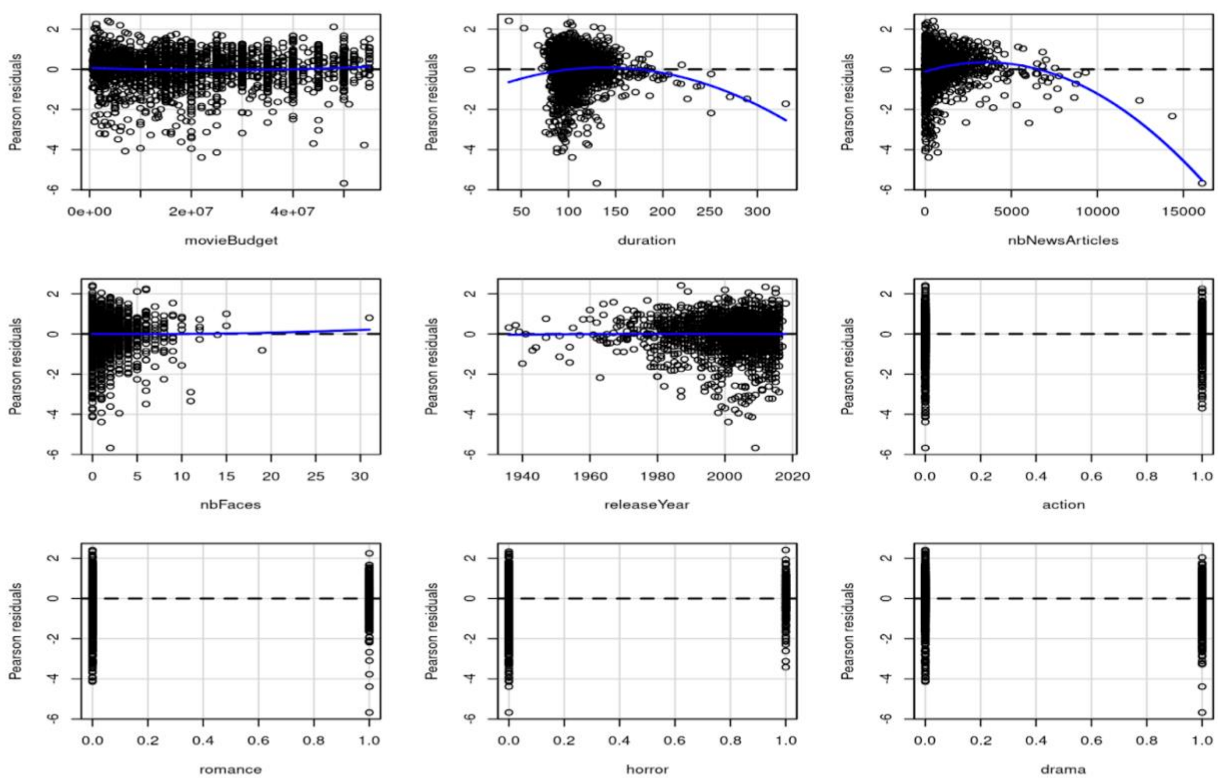
Appendix 14 – Coefficient test results

```
t test of coefficients:

                 Estimate  Std. Error t value  Pr(>|t|)    
(Intercept)    2.6187e+01  3.8541e+00  6.7945 1.444e-11 ***
movieBudget   -1.0912e-08  1.5280e-09 -7.1416 1.303e-12 ***
duration       1.9761e-02  1.4216e-03 13.9008 < 2.2e-16 ***
nbNewsArticles  2.2684e-04  3.2404e-05  7.0004 3.512e-12 ***
nbFaces       -3.7064e-02  1.1060e-02 -3.3511 0.0008204 ***
releaseYear   -1.0857e-02  1.9117e-03 -5.6791 1.560e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Appendix 15 – NCV test

```
> ncvTest(reg1)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.500771, Df = 1, p = 0.22055
```

Appendix 16 - Residual plots for linear model and tukey test



```
                Test stat Pr(>|Test stat|)
movieBudget        2.0628          0.03926 *
duration          -5.3140        1.197e-07 ***
nbNewsArticles   -10.9652        < 2.2e-16 ***
nbFaces            0.2936          0.76911
releaseYear       -0.1865          0.85204
action             0.7524          0.45193
romance            0.5429          0.58726
horror            -0.9667          0.33384
drama              0.8377          0.40231
Tukey test        -9.9517        < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Appendix 17 - Code of test regressions and related ANOVA results.

```
A = lm(imdbScore ~movieBudget +duration +nbNewsArticles +nbFaces+releaseYear +
        action+romance+horror+drama ) #Inital testing using annova of non-liniarity, just trying some ideas out
A2 = lm(imdbScore ~movieBudget +poly(duration,1) + poly(nbNewsArticles,2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +
        action+romance+horror+drama )
A3 = lm(imdbScore ~movieBudget +poly(duration,2) + bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +
       action+romance+horror+drama )
A4 = lm(imdbScore ~movieBudget +poly(duration,2) + poly(nbNewsArticles,2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +
       action+romance+horror+drama )
A5 = lm(imdbScore ~movieBudget +poly(duration,3) + poly(nbNewsArticles,2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +
       action+romance+horror+drama )
A6 = lm(imdbScore ~movieBudget +poly(duration,2) + poly(nbNewsArticles,3) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +
       action+romance+horror+drama )
A7 = lm(imdbScore ~movieBudget +poly(duration,3) + bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +
       action+romance+horror+drama )
```

```
  Res.Df    RSS Df Sum of Sq       F    Pr(>F)
1   1919 1543.5
2   1914 1440.0  5   103.445 28.4884 < 2.2e-16 ***
3   1909 1385.9  5    54.160 14.9154 2.194e-14 ***
4   1913 1421.7 -4   -35.808 12.3269 6.722e-10 ***
5   1912 1421.6  1     0.072  0.0990    0.7531
6   1912 1409.4  0    12.204
7   1908 1385.6  4    23.780  8.1863 1.523e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Appendix 18 - First prediction before factor optimization.

```
> predict(ModelFit,TestData) #test for new sig post removal of Nbnewsarticles
         1           2           3           4           5           6           7           8           9          10          11
  7.744466 2583.044943   33.034558    7.384455    7.218754    8.578563  233.132240    7.305562    7.303085    8.985514   21.462811
        12
 77.470957
```

Appendix 19 - Predictions after factor optimization.

```
       1        2        3        4        5        6        7        8        9       10       11       12
8.077609 9.613422 8.751198 7.753606 7.655581 8.740955 8.621498 7.809884 7.709367 9.217217 8.973521 8.112590
```

Appendix 20: R2 of the final optimized model

```
Residual standard error: 0.86 on 1907 degrees of freedom
Multiple R-squared:  0.3957,    Adjusted R-squared:  0.3887
F-statistic: 56.76 on 22 and 1907 DF,  p-value: < 2.2e-16
```

Appendix 21 - Stargazer charts for Optimized model on left and linear model on right for context

	Dependent variable:
	imdbScore
nbNewsArticles_spline1	0.236
	(0.192)
nbNewsArticles_spline2	0.273**
	(0.136)
nbNewsArticles_spline3	0.344**
	(0.151)
nbNewsArticles_spline4	0.686***
	(0.126)
nbNewsArticles_spline5	2.871***
	(0.414)
nbNewsArticles_spline6	-0.532
	(1.213)
nbNewsArticles_spline7	5.673***
	(1.856)
nbNewsArticles_spline8	-1.242
	(0.815)
duration	9.774***
	(0.992)
$duration^2$	-3.440***
	(0.887)
releaseYear_spline1	-0.182
	(0.783)
releaseYear_spline2	-0.393
	(0.399)
releaseYear_spline3	-1.076**
	(0.436)
releaseYear_spline4	-1.007**
	(0.419)
releaseYear_spline5	-1.607***
	(0.451)
releaseYear_spline6	-1.392***
	(0.463)
releaseYear_spline7	-0.634
	(0.614)
nbFaces	-0.036***
	(0.010)
action	-0.358***
	(0.052)
romance	-0.186***
	(0.048)
horror	-0.411***
	(0.067)
drama	0.460***
	(0.046)
Constant	6.927***
	(0.438)
Observations	1,929
R^2	0.396
Adjusted R^2	0.389
Residual Std. Error	0.859 (df = 1906)
F Statistic	56.885*** (df = 22; 1906)
Note:	*p<0.1; **p<0.05; ***p<0.01

	Dependent variable:
	imdbScore
movieBudget	-0.000***
	(0.000)
duration	0.014***
	(0.001)
nbNewsArticles	0.0002***
	(0.00002)
nbFaces	-0.041***
	(0.010)
releaseYear	-0.016***
	(0.002)
action	-0.295***
	(0.055)
romance	-0.173***
	(0.049)
horror	-0.381***
	(0.070)
drama	0.433***
	(0.048)
Constant	37.713***
	(3.850)
Observations	1,929
R^2	0.337
Adjusted R^2	0.334
Residual Std. Error	0.897 (df = 1919)
F Statistic	108.574*** (df = 9; 1919)
Note:	*p<0.1; **p<0.05; ***p<0.01

Code

```
#Loading the Data

dataDict = read.csv("/Users/kaykaydaou/Desktop/MCGILL U3/FALL 22/MGSC 401 - stat founds of data analytics/Midterm project/data_dictionary_IMDB.csv")

Data = read.csv("/Users/kaykaydaou/Desktop/MCGILL U3/FALL 22/MGSC 401 - stat founds of data analytics/Midterm project/IMDB_data.csv")

TestData = read.csv("/Users/kaykaydaou/Desktop/MCGILL U3/FALL 22/MGSC 401 - stat founds of data analytics/Midterm project/test_data_IMDB.csv")

attach(Data)

temp <- match(releaseMonth, month.abb) # create numeric value for month

Data[['releaseMonth_num']] = temp

attach(Data)

# Looking at the correlation matrix between the numerical variables

quantvars=Data[, c(4,5,6,8,9,13,15,18,20,22,25,40,43)]

corr_matrix=cor(quantvars)

x = (round(corr_matrix,3))

# Running a regression with all the numerical variables

testAll = lm(imdbScore ~ movieBudget + releaseDay + releaseYear + duration + aspectRatio  + nbNewsArticles+
```

```
actor1_starMeter + actor2_starMeter + actor3_starMeter + nbFaces + movieMeter_IMDBpro +releaseMonth_num )

summary(testAll)

# Running a regression with only variables deemed significant at a 99.9% level in testAll

reg1 = lm(imdbScore ~movieBudget +duration +nbNewsArticles +

    nbFaces + releaseYear)

# nbNewsArticles and movieMeter_IMDBpro were very collinear shown by the matrix luckily the regression made this a easy fix as movieMeter_IMDBpro was not *** sig

#Let's plot the numerical variables' relationship to the target variable

plot(movieBudget,imdbScore)

plot(duration,imdbScore) #very skewed

plot(nbNewsArticles,imdbScore)

plot(nbFaces,imdbScore)

plot(releaseYear,imdbScore) #skewed towards newer movies (as one could expect with more releases with technological advances in film)

#Let's get an idea of the numerical variables' distribution to check for interesting patterns (and potential model issues)

boxplot(imdbScore, xlab = "imdbScore")

boxplot(movieBudget,xlab = "movieBudget")

boxplot(duration,xlab = "duration")

boxplot(nbNewsArticles,xlab = "nbNewsArticles")

boxplot(nbFaces,xlab = "nbFaces")
```

```
boxplot(releaseYear,xlab = "releaseYear")


require(car)

require(lmtest)

require(plm)


summary(reg1)

qqPlot(reg1, envelope=list(style="none"))

outlierTest(reg1) #doing outlier test


#Looking over reg1's qqplot, we can remove Observation 492 as it was by far the biggest outlier


NewData = Data[-c(492), ]

detach(Data)

attach(NewData)


# Running again the regression with the same factors (after removing the outlier)

reg1 = lm(imdbScore ~movieBudget +duration +nbNewsArticles +

        nbFaces + releaseYear)

summary(reg1)

# R-squared has increased by 0.0266 with the outlier removed


# Let's look for trends in the residual plots.

residualPlot(reg1,quadratic=FALSE)
```

```
# The plot suggests we will be needing some polynomials - this will be tested later


# Let's check for heteroskedasticity issues in our regression model

ncvTest(reg1)

coeftest(reg1, vcov=vcovHC(reg1, type="HC1")) # after heteroskedasticity test P value still sig



## Now, focusing our attention on categorical data


# Let's run a regression with the genre variables only

reg2 = lm(imdbScore ~ action +adventure+scifi+thriller+

        musical+romance+western+sport+horror+drama+war+

        animation+crime)


summary(reg2)


#Let's look at the distribution of some categorical variables

###Distribution of movie genres

genre = subset(Data, select = c("action", "adventure", "scifi", "thriller", "musical", "romance", "western", "sport", "horror", "drama", "war", "animation", "crime"))

genre_count = c()

for (j in genre) {

  genre_count = append(genre_count, sum(j))

}
```

```
barplot(genre_count, main="Distribution of movie genres", names.arg=names(genre), col = "gold")


require(ggplot2)

###Distribution of movies by month

monthWithoutOrder = table(Data$releaseMonth)

months = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")

monthWithOrder = c()

for (x in 1:12){

  j = monthWithoutOrder[months[x]]

  monthWithOrder = append(monthWithOrder, j)

}

barplot(monthWithOrder, main="Distribution of movies by month", names.arg=months, col = "gold")


###Distribution of movie languages

ggplot(Data,aes(x=language))+ geom_bar(fill="gold",color="gold")+ggtitle("Distribution of movie languages")


### Distribution of movie by country

ggplot(Data,aes(x=country))+ geom_bar(fill="gold",color="gold")+ggtitle("Distribution of movie by country")


###Distribution of movies by maturity

ggplot(Data,aes(x=maturityRating))+ geom_bar(fill="gold",color="gold")+ggtitle("Distribution of movie by maturity")
```

```
# Let's visualize the distribution of genre in our datasets with some simple manipulation

newLi = c(sum(adventure),sum(scifi),sum(thriller),

        sum(musical) ,sum(romance),sum(western),sum(sport),

        sum(horror),sum(drama), sum(war), sum(animation),sum(crime))


label = c('adventure','scifi','thriller','musical','romance',

        'western','sport','horror','drama','war' ,'animation','crime')

names(newLi) = label

newLi

# We can see that the data is very skewed suggesting many of the category are not good predictors


# Running an updated regression with only significant genre variables at a 99.9% level

reg2 = lm(imdbScore ~ action+romance+horror+drama+war)


# Running a regression with the selected significant numerical variables, significant genre variables, country and language

catTest = lm(imdbScore ~movieBudget +duration +nbNewsArticles +

           nbFaces + releaseYear +action+romance+horror+drama+war+

           + country + language)


summary(catTest)

# It appears that both country and language do not matter as they return extremely high P-values (low significance levels)
```

```
# Let's run a regression for the maturity rating factor to test its significance

mattest = lm(imdbScore ~maturityRating)


# Dummifying significant maturity ratings based on the mattest regression

maturityRatingG = ifelse(maturityRating == 'G',1,0)

maturityRatingPG = ifelse(maturityRating == 'PG',1,0)

maturityRatingPG13 = ifelse(maturityRating == 'PG-13',1,0)

maturityRatingR = ifelse(maturityRating == 'R',1,0)

maturityRatingTV14 = ifelse(maturityRating == 'TV-14',1,0)

maturityRatingTVG = ifelse(maturityRating == 'TV-G',1,0)



# Running a regression of relevant genre variables and the maturity ratings created

reg3 = lm(imdbScore ~ action+romance+horror+drama+war+

        maturityRatingG +maturityRatingPG+maturityRatingPG13+

        maturityRatingR+maturityRatingTV14+maturityRatingTVG

)

summary(reg3)


# Running a regression with all the significant numerical variables, significant genre variables and maturity ratings

AllTest = lm(imdbScore ~ movieBudget +duration +nbNewsArticles +

          nbFaces + releaseYear +

          action+romance+horror+drama+war+
```

```
        maturityRatingG +maturityRatingPG+maturityRatingPG13+

        maturityRatingR+maturityRatingTV14+maturityRatingTVG

)


summary(AllTest)


# Based on the AllTest regression, let's run a regression with the remaining significant variables at a 99.9% level

reg4 = lm(imdbScore ~movieBudget +duration +nbNewsArticles +nbFaces+releaseYear +

       action+romance+horror+drama

)

summary(reg4)

# Running a coefficient test to verify reg4 factor significance

coeftest(reg4, vcov=vcovHC(reg4, type="HC1"))

# We can see that all variables are significant at the 99.9% level despite potential heteroskedasticity issues


#To optimize the model and fit the data better, we will be checking for non-linearity issues and potentially integrating splines


residualPlots(reg4)

#Here we see the residual plots for all the variables used in the reg4 model, enabling us to determine if some variables are better modelled using polynomial or spline relationships


#Let's plot the variables to visualize some relationships better (but only those that seemed non-linear in the residual plot)

plot(movieBudget,imdbScore)
```

```
# movieBudget seems relatively linear to the target variable

plot(duration,imdbScore)

# Duration is definitively not linear: let's try another relationship

plot(nbFaces,imdbScore)

# Number of faces is definitively not linear: let's try another relationship

plot(releaseYear,imdbScore)


#After visualizing relationships, let's run an updated regression with polynomials

reg7 = lm(imdbScore ~movieBudget +poly(duration,2) + poly(nbNewsArticles,2) + nbFaces+releaseYear +

        action+romance+horror+drama ) #testing 1 degree of linearity

summary(reg7)

residualPlots(reg7)

# We can see that movie budget gains a significant level of non-linearity in the new model


# We will test both releaseYear and Number of news articles with a spline regression as its relationship might be better accounted with such model

# Creating knots releaseYear

m1= quantile(releaseYear,.20)

m2= quantile(releaseYear,.40)

m3= quantile(releaseYear,.60)

m4= quantile(releaseYear,.80)

# Creating knots nbNewsArticles

k1= quantile(nbNewsArticles,.20)

k2= quantile(nbNewsArticles,.40)
```

k3= quantile(nbNewsArticles,.60)

k4= quantile(nbNewsArticles,.80)

Running different regression scenarios and relationships to compare ANOVA tests, and determine the best relationship

library(splines)

A = lm(imdbScore ~movieBudget +duration +nbNewsArticles +nbFaces+releaseYear +

action+romance+horror+drama) #Inital testing using annova of non-liniarity, just trying some ideas out

A2 = lm(imdbScore ~movieBudget +poly(duration,1) + poly(nbNewsArticles,2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +

action+romance+horror+drama)

A3 = lm(imdbScore ~movieBudget +poly(duration,2) + bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +

action+romance+horror+drama)

A4 = lm(imdbScore ~movieBudget +poly(duration,2) + poly(nbNewsArticles,2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +

action+romance+horror+drama)

A5 = lm(imdbScore ~movieBudget +poly(duration,3) + poly(nbNewsArticles,2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +

action+romance+horror+drama)

A6 = lm(imdbScore ~movieBudget +poly(duration,2) + poly(nbNewsArticles,3) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +

action+romance+horror+drama)

A7 = lm(imdbScore ~movieBudget +poly(duration,3) + bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +nbFaces+bs(releaseYear,knots=c(m1,m2,m3,m4), degree=1) +

action+romance+horror+drama)

```
anova(A,A2,A3,A4,A5,A6,A7)


#Running a spline regression of degree=2 with only the number of Articles as a predictor to test if the relationship is better

reg8 = lm(imdbScore~bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2))

residualPlots(reg8)

summary(reg8)


#Running a quadratic regression with only the number of Articles as a predictor to test if the relationship is better under this relationship

reg9 = lm(imdbScore~poly(nbNewsArticles,2))

residualPlots(reg9)

summary(reg9)

# We can see that the reg8's R-squared is higher: the spline is a better relationship and the clear winnter


#Running a regression with only quadratic relationships

NonlinearTest = lm(imdbScore~ poly(nbNewsArticles,2)+

            poly(movieBudget,2) +poly(duration,2)+nbFaces+releaseYear +

            action+romance+horror+drama )


summary(NonlinearTest)


#Running a regression with both quadratic relationships and a spline regression for nbNewsArticles (as shown better in reg8)

reg9 = lm(imdbScore~ bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +
```

```
        poly(movieBudget,2) +poly(duration,2)+nbFaces+releaseYear +
        action+romance+horror+drama)

summary(reg9)
#We can see that the spline relationship is 1/100 better when combined with the other variables

residualPlots(reg9)

# Let's run some trials testing spline regressions for the other variables (we never know what we'll find)
A1= quantile(duration,.20)
A2= quantile(duration,.40)
A3= quantile(duration,.60)
A4= quantile(duration,.80)
C1= quantile(movieBudget,.20)
C2= quantile(movieBudget,.40)
C3= quantile(movieBudget,.60)
C4= quantile(movieBudget,.80)

#Running a regression with only spline relationships
reg10 = lm(imdbScore~ bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +
         bs(movieBudget,knots=c(C1,C2,C3,C4), degree=2) +
         bs(duration,knots=c(A1,A2,A3,A4), degree=2) +
         +nbFaces+releaseYear +
         action+romance+horror+drama)
```

summary(reg10)

#It appears that our model's adjusted r-squared decreased: fitting spline relationships is not worth it

residualPlots(reg10)

#We can still see that releaseYear is non-linear, let's try to solve that issue

#Running a regression fitting a quadratic relationship to releaseYear

reg11 = lm(imdbScore~ bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +

poly(movieBudget,2) +poly(duration,2)+nbFaces+ poly(releaseYear,2) +

action+romance+horror+drama)

residualPlots(reg11)

summary(reg11)

#Let's try to fit a spline relationship to the releaseYear predictor

m1= quantile(releaseYear,.20)

m2= quantile(releaseYear,.40)

m3= quantile(releaseYear,.60)

m4= quantile(releaseYear,.80)

#Running a regression with the said relationship (at degree=2)

reg12 = lm(imdbScore~ bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +

poly(movieBudget,2) +poly(duration,2)+nbFaces+

bs(releaseYear,knots=c(m1,m2,m3,m4), degree=2)

```
        + action+romance+horror+drama)


residualPlots(reg12)

summary(reg12)

#It appears that fitting a spline relationship is marginally better


#Let's start fitting and creating an optimized model following our tests and trials

library(boot)


#Running a regression based on the best fit relationships as tested above

ModelFit=glm(imdbScore~ bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=2) +

          poly(movieBudget,2) +poly(duration,2)+nbFaces+ poly(releaseYear,2) +

          action+romance+horror+drama)


#Our aim is to minimze MSE as a measure of performance: let's calculate the model's MSE

mse=cv.glm(NewData, ModelFit)$delta[1]

mse

#MSE was determined to be 0.7380479


#Previously, we ran spline fits of only degree 2 and only quadratic relationships: let's determine the best degrees for our optimised model as well as the best number of polynomials


#bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=i) +


Answers = rep(NA,(4))
```

```
BestScore = NA


for(i in 1:5)  {
  for (p in 1:5) {
    for (q in 1:5) {
      for (r in 1:5) {


        ModelFit=glm(imdbScore~
                        bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=i) +
                        poly(movieBudget,p) +
                        poly(duration,q) +
                        bs(releaseYear,knots=c(m1,m2,m3,m4), degree=r) +
                        nbFaces +action+romance+horror+drama)


        cv.error=cv.glm(NewData, ModelFit, K=20)$delta[1]
        if (is.na(BestScore)) {BestScore = cv.error}
        else if (BestScore > cv.error) {
          Answers = c(i,p,q,r)
          BestScore = cv.error
        }}}}}


Answers
BestScore
#Answers we got: 5 4 2 2
```

```
#MSE = 0.73


#Let's run our model with the optimized parameter values (Answers)

ModelFit=glm(imdbScore~
        bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=5) +
        poly(movieBudget,4) +
        poly(duration,2) +
        bs(releaseYear,knots=c(m1,m2,m3,m4), degree=2) +
        nbFaces +action+romance+horror+drama)



mse=cv.glm(NewData, ModelFit,K=100)$delta[1]

mse

#We can see that our model's MSE has gotten better

#MSE = 0.7312127


summary(ModelFit) #wakanda has outlier amount of reviews

predict(ModelFit,TestData) #test for new sig post removal of Nbnewsarticles


#It appears that nbNewsArticles is lacking significance in our optimized model. Let's substitute it with movieMeter_IMDBpro as we've seen earlier that the two factors were collinear.


#Running a regression with movieMete_IMBpro instead

reg15 = lm(imdbScore ~movieBudget +duration +movieMeter_IMDBpro +
       nbFaces + releaseYear +
```

```
        action+romance+horror+drama) #signfiant

summary(reg15)


residualPlots(reg15)

# We can see from the residual plot that movieMeter_IMDBpro could be better fit with a polynomial relationship


Answers = rep(NA,(4))

BestScore = NA


#Let's find the best parameter values for our new regression model

for (i in 1:4) { #iterating to find the degree of the spline and poly

  for (p in 1:4) {

    for (q in 1:4) {

      for (r in 1:4) {


        ModelFit=glm(imdbScore~

                poly(movieMeter_IMDBpro ,i) + poly(movieBudget,p)

               +poly(duration,q)+nbFaces+

                bs(releaseYear,knots=c(m1,m2,m3,m4), degree=r) +

                action+romance+horror+drama)


        cv.error=cv.glm(NewData, ModelFit, K=20)$delta[1]

        if (is.na(BestScore)) {BestScore = cv.error}
```

```
    else if (BestScore > cv.error) {
      Answers = c(i,p,q,r)
      BestScore = cv.error
    }
  }}}}
Answers
BestScore
#Ans = 3 4 2 2
#MSE = 0.8007239


#Running a regression with the optimized parameters for movieMeter_IMDBpro
ModelFit=glm(imdbScore~ poly(ModelFit=glm(imdbScore~poly(movieMeter_IMDBpro ,3) + poly(movieBudget,4)
                    +poly(duration,2)+nbFaces+ poly(releaseYear,1) +
                     action+romance+horror+drama) ,3))


mse=cv.glm(NewData, ModelFit,K=100)$delta[1]
mse


residualPlots(ModelFit)
summary(ModelFit)
predict(ModelFit,TestData)
# Again, this model doesnt work as the min(movieMeter_IMDBpro) = 71 black panther is 14th


#Let's refine model without movieMeter_IMDBpro
```

```
# We are going to take out Movie budget as it adds little MSE reduction

Answers = rep(NA,(2))
BestScore = NA

plot(movieBudget,imdbScore)

m1= quantile(releaseYear,.20)
m2= quantile(releaseYear,.40)
m3= quantile(releaseYear,.60)
m4= quantile(releaseYear,.80)

for (p in 1:3) {
  for (q in 1:3) {

    ModelFit=glm(imdbScore~

                poly(duration,p) +
                bs(releaseYear,knots=c(m1,m2,m3,m4), degree=q) +
                nbFaces +
                action+romance+horror+drama)

    cv.error=cv.glm(NewData, ModelFit, K=15)$delta[1]
    if (is.na(BestScore)) {BestScore = cv.error}
```

```
  else if (BestScore > cv.error) {
    Answers = c(p,q)
    BestScore = cv.error
  }}}

Answers
BestScore
#Ans = 3 2
#MSE = 0.87388

#Running a regression without movieMeter_IMDBscore and movieBudget
ModelFit=glm(imdbScore~
        poly(duration,2) +
        bs(releaseYear,knots=c(m1,m2,m3,m4), degree=3) +
        nbFaces+
        action+romance+horror+drama)

mse=cv.glm(NewData, ModelFit,K=100)$delta[1]
mse
#real MSE

summary(ModelFit)
predict(ModelFit,TestData)
#Our predicted results are much better and more logical sound
```

```
# We have now discovered that the issue might be movieBudget

Answers = rep(NA,(3))
BestScore = NA

for (i in 1:4) { #iterating to find the degree of the spline and poly
 for (q in 1:4) {
  for (r in 1:4) {

   ModelFit=glm(imdbScore~
           bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=i) +
           poly(duration,q) +
           bs(releaseYear,knots=c(m1,m2,m3,m4), degree=r) +
           nbFaces +action+romance+horror+drama)

   cv.error=cv.glm(NewData, ModelFit, K=20)$delta[1]
   if (is.na(BestScore)) {BestScore = cv.error}

   else if (BestScore > cv.error) {
    Answers = c(i,q,r)
    BestScore = cv.error
   }
  }}}
```

Answers

BestScore

```
ModelFit=glm(imdbScore~
        bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=4) +
        poly(duration,2) +
        bs(releaseYear,knots=c(m1,m2,m3,m4), degree=3) +
        nbFaces +action+romance+horror+drama)

mse=cv.glm(NewData, ModelFit,K=250)$delta[1]
mse # real MSE

residualPlots(ModelFit)
summary(ModelFit)
x = predict(ModelFit,TestData)

#Let's create our final model based on the optimized parameters and the best fit variables
finalModel = lm(imdbScore~
          bs(nbNewsArticles,knots=c(k1,k2,k3,k4), degree=4) +
          poly(duration,2) +
          bs(releaseYear,knots=c(m1,m2,m3,m4), degree=3) +
          nbFaces +action+romance+horror+drama)

summary(finalModel)
```

```
#Let's create a starGazer table to present our results

library(stargazer)


#Let's create a non-optimized model to show comparison

NonOptimizedModel = lm(imdbScore ~movieBudget +duration +nbNewsArticles +nbFaces+releaseYear +

                        action+romance+horror+drama)


names(finalModel$coefficients) = c("(Intercept)","nbNewsArticles_spline1","nbNewsArticles_spline2","nbNewsArticles_spline3","nbNewsArticles_spline4",

"nbNewsArticles_spline5","nbNewsArticles_spline6","nbNewsArticles_spline7","nbNewsArticles_spline8",

                            "duration","duration2",

"releaseYear_spline1","releaseYear_spline2","releaseYear_spline3","releaseYear_spline4",

                            "releaseYear_spline5","releaseYear_spline6","releaseYear_spline7", "nbFaces",

                            "action", "romance", "horror", "drama"

)


#Stargazer for our final model

stargazer(finalModel,type="html")
```

```
#Stargazer for our non-optimized model

stargazer(NonOptimizedModel,type="html")


summary(finalModel)
```