1 Introduction

This project utilizes the statistical foundations of predictive data analytics using the R programming language. The goal of this project is to accurately predict the IMDb ratings of the twelve upcoming movies. As a team, our objective is to find a superior model with an optimal selection of predictors, and degree relationships, and deliver a clear, concise report that identifies the model selection which can get the most accurate IMDb score prediction of the 12 movies.

The process involves various examinations like predictor significance, non-linearity, heteroskedasticity, outliers, and collinearity. In addition to grouping selected variables, different regression analyses, and excluding variables to enhance the model's predictive power. Spoiler Alert: Our finalized ratings in Section 4 might tempt you to renew your Netflix subscription, stay home, and get your popcorn!

2 Data Description

In the original dataset, there were 42 attributes including the target variable: IMDb scores for each movie. To downsize the models and improve the efficiency, multiple data pre-processing steps were performed to ensure that we have a comprehensive understanding of the data and that it was concise enough for model building.

2.1 Subset

Before we started to manipulate the dataset, a subset from the original dataset was created upon our agreement to secure an unmodified reference. In our subset, called "our dataset" in the following paragraphs, the label variables such as *movieID* were eliminated since they were identical to each observation and would not be able to provide valuable information for model formulation. The variable *genres* was removed as well as it is already dummified into numerous columns.

2.2 Categorical and Numerical Attribute Segregation

The initial step we took was to segregate categorical and numerical predictors. Columns with alphabetical values are identified as categorical variables. Yet attributes in numbers could still be classified as categorical ones still based on their nature. For instance, *aspectRatio* has numerical values such as 1.85, yet it is actually representing different types of films. After identifying the types of all variables, categorical columns were converted into factors to enable us to plug them into the models.

2.3 Removing Insignificant Predictors

Several variables are excluded as we observed that there was no significant relevance to the target value from the scatter plots, including *releaseDay*, *releaseMonth*, and *releaseYear*. Variables that are highly related to others were also removed. For instance, *productionCompany* and *distributor* are distinctly related to *director*. Thus, the first two were dropped.

2.4 Cardinality

We continued the pre-processing procedure with null value checks in our dataset on all the columns and found that most rows of the predictor *nbFaces* had 0 value. The homogeneity observed in this column might not attribute much to our prediction model and *nbFaces* was subsequently dropped from our final training dataset. Other categorical columns with either relatively high cardinality or uniform values were dropped. For example, the values in *language* column of more than 90% of the movies in the original dataset were English thus this variable was dropped as it gave us neither valuable information nor predictive power and was correlated to the attribute country as well.

2.5 Derived Variables

To utilize the information more efficiently, we decided to aggregate or merge specific variables. Based on our general understanding, directors of movies have notable impacts on movie ratings. Instead of having a categorical variable with hundreds of levels, we calculated the average movie ratings for all directors and ranked them accordingly. Furthermore, we grouped the directors by their ranking. For example, directors rank 1 to rank 10 will be in tier 1, directors rank 11 to rank 20 will be in tier 2 etc. Another example of derived variables is that we computed the average of *actor1_starMeter*,*actor2_starMeter*, and *actor3_starMeter* along with the log of each variable to test for significance, yet the *actor1_starMeter* resulted in the highest significance and was selected in the model configuration.

2.6 Final Adjustment

Our entire dataset was subsetted to only colour films based on the column *ColourFilm* as all movies released in recent decades are mostly coloured. It was also observed that relatively old, black-and-white films generally had higher ratings, and thus will potentially skew the results. This relationship is presented in Plot 1 in the appendix. Once our training dataset was organized after performing all the pre-processing steps mentioned above, multiple models were built based on the filtered predictors and the best fits will be identified.

3 Model Selection

3.1 Regression Analyses

Polynomial regression of varying degrees from 1 to 6 and the ANOVA test was conducted to select the optimal degree relationship (if $p < 0.05$, the next degree model was selected). This process was repeated for each numerical predictor and the optimal degree was recorded, referring to Table 1a to Table 1e.

In addition to polynomial regression, two models of Spline regression were performed with $k = 5$ and $k = 2$, but no significant improvement in the adjusted R^2 was observed, and the observations were prone to overfitting. Therefore, the spline regression was neglected.

3.2 Model Issues

3.2.1 Non-Linearity of Predictors

Residual plots of numerical predictors were created to determine the non-linearity check (if $p > 0.1$, the predictor was considered linear). Only *nbFaces* variable was linear, referring to Table 2 and Plot 2.

3.2.2 Heteroskedasticity

The ncv test was conducted on the optimal model selected. Since $p < 0.5$, the heteroskedasticity issue was identified and resolved using *coeftest*. This allowed the optimal model to be unbiased and that all predictors' significance is not artificially increased or decreased, referring to Table 3.

3.2.3 Outliers

Bonferroni numerical outlier test was conducted separately for each predictor and outliers were removed to enhance predictive power and mitigate the significant influence of certain abnormal observations. Outlier test results are presented in Table 4a to Table 4e in the appendix.

3.2.4 Collinearity

Collinearity was identified using the Variance Inflation Factors (VIF) Test, see Table 5.

3.3 Model Selection

After narrowing the predictor selection process, the model was tested based on interchangeable degrees, predictors, and regression relationships to identify the highest adjusted R^2 value. **Model 11** scored the highest and was compared against the rest of the models with cross-validation testing to identify the MSE margin. Our focus was to achieve a minimal number of predictors while maintaining the highest adjusted R^2 value and minimizing the MSE. This will allow the selected model to achieve high predictive power with fewer overfitting issues and the least number of predictors. Refer to Table 6 for model comparison.

4 Results

4.1 Final Model and Prediction

The final model that we have selected is mathematically represented as,

$$\begin{aligned} y =& \beta_0 + \beta_1 \times movieBudget + \beta_2 \times duration + \beta_3 \times duration^2 \\ &+ \beta_4 \times nbNewsArticles + \beta_5 \times nbNewsArticles^2 + \beta_6 \times nbNewsArticles^3 \\ &+ \beta_7 \times nbFaces + \beta_8 \times western + \beta_9 \times drama + \beta_{10} \times animation + \beta_{11} \times ranking\_bin2 \\ &+ \beta_{12} \times ranking\_bin3 + \beta_{13} \times ranking\_bin4 + \beta_{14} \times ranking\_bin5 + \beta_{15} \times ranking\_bin6 \\ &+ \beta_{16} \times ranking\_bin7 + \beta_{17} \times ranking\_bin8 + \beta_{18} \times ranking\_bin9 + \beta_{19} \times ranking\_bin10 \end{aligned}$$

where *imdbScore* is our dependent variable represented by y.

4.2 Test Set Processing and Predictions

To ensure the test set is compatible with our model, we followed similar data pre-processing as we did with the training set. We got rid of predictors like *colourFilm*, *movieID*, *releaseYear* etc to ensure the number of predictors in the train and test set is the same. For derived columns like *ranking_bin*, we derived the bins that each director should be placed in and then left-joined that data with the test set to recreate the *ranking_bin* column in the test set. We proceeded with the predictions once all the predictors were aligned between the test and train set. Following are our predictions for the 12 movies using the optimal model selected:

Movie Name	**Predicted Score on IMDb**
Falling for Christmas	6.44180
Black Panther: Wakanda Forever	6.19261
Spirited	5.81863
Paradise City	5.47408
Poker Face	6.55002
Que Viva Mexico!	6.97907
Slumberland	5.98879
Blue's Big City Adventure	6.75056
The Menu	5.77393
The Fablemans	7.67115
Devotion	6.42526
Strange World	6.26543

4.3 Performance and Validation

The R^2 of the model comes out to be 0.7481, which implies that 74.81% of the variance in the data can be explained by our model.

The MSE calculated after splitting the dataset by an 8:2 ratio into train and test set respectively is **0.2765225**.

The MSE calculated after performing a LOOCV test is **0.2952443**.

The MSE calculated after performing a K-fold validation test with $k = 5$ is **0.2965276**.

With an average MSE of 0.2894 across the 3 types of tests, our MSEs are stable and without much deviation for each method.

4.4 Parameter Analysis

The following tables show us the coefficients (see Table 7a) and significance (see Table 7b) of each of the independent variables after performing the heteroskedasticity transformation. All predictors selected are significant with p-value less than 0.05 except for,

- β_3 ($p-value = 0.16$) corresponding to $duration^2$ was included since a polynomial function with degree 2 was found out to be the best fit (highest R^2) between *imdbScore* and *duration* when modelled individually.
- β_6 ($p-value = 0.25$) corresponding to $nbNewsArticles^3$ was included since a polynomial function with degree 3 was found out to be the best fit (highest R^2) between *imdbScore* and *nbNewsArticles* when modelled individually.
- β_7 ($p-value = 0.16$) corresponding to *nbFaces* was included since from a business standpoint, the number of faces often has a significant effect on the rating of the movie in terms of having big stars. Furthermore, the p-value is low suggesting there is an 84% chance the predictor is significant.

4.5 Business Insights

Analysing the coefficients for *ranking_bin i* (for *i = 2,3...10*), having directors outside the top 10 percentile (*ranking_bin*1) has a negative effect on the *imdbScore* and it slowly becomes worse with each bucket towards the bottom. For example, $\beta_{11} = -0.34$ means that if a movie has a director from the *ranking_bin*2 it will result in 0.34 decrease in the *imdbScore*. Same applies for β_{12} to β_{19}. Movies belonging to the animation genre have the largest effect on *imdbScore* with $\beta_{10} = 0.37$, implying an animation movie will tend to have a 0.37 increase in their score.

4.6 Future Recommendations and Improvements

Based on the review of the dataset and information collected by watching movies all our lives, there are multiple recommendations that we can think of to improve the overall predictive power of our model.

- Our Model is over reliant on a few predictors: Though *budget* and *duration* are good indicators of movie rating, they fall short of capturing a lot of trends observed in the data and in real life and are biased themselves. For example, budgets of movies released a few decades ago were lower than latest movies as the values are not inflation adjusted.

  Solution
  Supplement the dataset with new predictors that can identify patterns like the one above.

- Most "intuitively" useful predictors had to be dropped: predictors like *cinematographer*, *distributor* etc. had to be dropped from analysis due to high cardinality and collinearity but are intuitively powerful predictors.

  Solution
  Replace or engineer these predictors with alternatives like cinematographer rank or number of distributors that can be used to enhance the predictive power of the model.

5 Appendices

5.1 Tables

Table 1a. Polynomial Regression: *imdbScore* vs. *avgStarMeter*
Adjusted $R^2 = 0.004$ of degree 1 was reported to be the highest and selected.

	Dependent variable:					
	imdbScore					
	(1)	(2)	(3)	(4)	(5)	(6)
Average Star Meter	0.000	1.457	1.457	1.457	1.457	1.457
	(0.000)	(1.094)	(1.094)	(1.094)	(1.094)	(1.094)
Average Star Meter^2		2.701**	2.701**	2.701**	2.701**	2.701**
		(1.094)	(1.094)	(1.094)	(1.094)	(1.094)
Average Star Meter^3			-1.227	-1.227	-1.227	-1.227
			(1.094)	(1.094)	(1.094)	(1.094)
Average Star Meter^4				0.981	0.981	0.981
				(1.094)	(1.094)	(1.094)
Average Star Meter^5					1.256	1.256
					(1.094)	(1.094)
Average Star Meter^6						0.501
						(1.094)
Constant	6.474***	6.479***	6.479***	6.479***	6.479***	6.479***
	(0.026)	(0.025)	(0.025)	(0.025)	(0.025)	(0.025)
Observations	1,867	1,867	1,867	1,867	1,867	1,867
R^2	0.001	0.004	0.005	0.005	0.006	0.006
Adjusted R^2	0.0004	0.003	0.003	0.003	0.003	0.003
Residual Std. Error	1.095 (df = 1865)	1.094 (df = 1864)	1.094 (df = 1863)	1.094 (df = 1862)	1.094 (df = 1861)	1.094 (df = 1860)
F Statistic	1.769 (df = 1; 1865)	3.936** (df = 2; 1864)	3.044** (df = 3; 1863)	2.484** (df = 4; 1862)	2.251** (df = 5; 1861)	1.910* (df = 6; 1860)
Note:						*p<0.1; **p<0.05; ***p<0.01

Table 1b. Polynomial Regression: *imdbScore* vs. *duration*
Adjusted $R^2 = 0.183$ of degree 2 was reported to be the highest and selected.

	Dependent variable:				
	imdbScore				
	(1)	(2)	(3)	(4)	(5)
Duration	0.021***	19.235***	19.235***	19.235***	19.235***
	(0.001)	(0.990)	(0.990)	(0.990)	(0.990)
Duration^2		-6.379***	-6.379***	-6.379***	-6.379***
		(0.990)	(0.990)	(0.990)	(0.990)
Duration^3			-0.856	-0.856	-0.856
			(0.990)	(0.986)	(0.983)
Duration^4				4.092***	4.092***
				(0.983)	(0.983)
Duration^5					-3.531***
					(0.983)
Constant	4.146***	6.479***	6.479***	6.479***	6.479***
	(0.124)	(0.023)	(0.023)	(0.023)	(0.023)
Observations	1,867	1,867	1,867	1,867	1,867
R^2	0.165	0.183	0.184	0.191	0.197
Adjusted R^2	0.165	0.183	0.182	0.189	0.195
Residual Std. Error	1.001 (df = 1865)	0.990 (df = 1864)	0.990 (df = 1863)	0.986 (df = 1862)	0.983 (df = 1861)
F Statistic	369.206*** (df = 1; 1865)	209.358*** (df = 2; 1864)	139.802*** (df = 3; 1863)	110.069*** (df = 4; 1862)	91.200*** (df = 5; 1861)
Note:					*p<0.1; **p<0.05; ***p<0.01

Table 1c. Polynomial Regression: *imdbScore* vs. *movieBudget*
Adjusted $R^2 = 0.005$ of degree 1 was reported to be the highest and selected.

	Dependent variable:				
	imdbScore				
	(1)	(2)	(3)	(4)	(5)
Movie Budget	-0.000*	-3.454*	-3.454*	-3.454*	-3.454*
	(0.000)	(1.093)	(1.093)	(1.093)	(1.093)
Movie Budget^2		0.525	0.525	0.525	0.525
		(1.093)	(1.093)	(1.093)	(1.093)
Movie Budget^3			-0.037	-0.037	-0.037
			(1.093)	(1.093)	(1.093)
Movie Budget^4				1.096	1.096
				(1.093)	(1.093)
Movie Budget^5					0.738
					(1.093)
Constant	6.595	6.479	6.479	6.479	6.479*
	(0.044)	(0.025)	(0.025)	(0.025)	(0.025)
Observations	1,867	1,867	1,867	1,867	1,867
R^2	0.005	0.005	0.005	0.006	0.006
Adjusted R^2	0.005	0.004	0.004	0.004	0.004
Residual Std. Error	1.093 (df = 1865)	1.093 (df = 1864)	1.093 (df = 1863)	1.093 (df = 1862)	1.093 (df = 1861)
F Statistic	9.992 (df = 1; 1865)	5.109 (df = 2; 1864)	3.405 (df = 3; 1863)	2.805 (df = 4; 1862)	2.334 (df = 5; 1861)
Note:					p<0.1; p<0.05; **p<0.01

Table 1d. Polynomial Regression: *imdbScore* vs. *movieMeter_IMDBpro*
Adjusted $R^2 = 0.066$ of degree 4 was reported to be the highest and selected.

	Dependent variable:			
	imdbScore			
	(1)	(2)	(3)	(4)
Movie Meter IMDb Pro	-0.000***	-4.037***	-4.037***	-4.037***
	(0.000)	(1.075)	(1.058)	(1.050)
Movie Meter IMDb Pro^2		8.173***	8.173***	8.173***
		(1.075)	(1.058)	(1.050)
Movie Meter IMDb Pro^3			-8.285***	-8.285***
			(1.058)	(1.050)
Movie Meter IMDb Pro^4				5.942***
				(1.050)
Constant	6.506***	6.479***	6.479***	6.479***
	(0.026)	(0.025)	(0.024)	(0.024)
Observations	1,867	1,867	1,867	1,867
R^2	0.007	0.037	0.068	0.084
Adjusted R^2	0.007	0.036	0.066	0.082
Residual Std. Error	1.092 (df = 1865)	1.075 (df = 1864)	1.058 (df = 1863)	1.050 (df = 1862)
F Statistic	13.674*** (df = 1; 1865)	35.921*** (df = 2; 1864)	45.142*** (df = 3; 1863)	42.430*** (df = 4; 1862)
Note:				*p<0.1; **p<0.05; ***p<0.01

Table 1e. Polynomial Regression: *imdbScore* vs. *nbNewsArticles*

Adjusted $R^2 = 0.005$ of degree 3 was reported to be the highest and selected.

	Dependent variable:					
	imdbScore					
	(1)	(2)	(3)	(4)	(5)	(6)
News Articles	0.0001***	10.693***	10.693***	10.693***	10.693***	10.693***
	(0.00001)	(1.045)	(1.027)	(1.025)	(1.022)	(1.022)
News Articles^2		-9.406***	-9.406***	-9.406***	-9.406***	-9.406***
		(1.045)	(1.027)	(1.025)	(1.022)	(1.022)
News Articles^3			8.504***	8.504***	8.504***	8.504***
			(1.027)	(1.025)	(1.022)	(1.022)
News Articles^4				-2.680***	-2.680***	-2.680***
				(1.025)	(1.022)	(1.022)
News Articles^5					3.782***	3.782***
					(1.022)	(1.022)
News Articles^6						0.068
						(1.022)
Constant	6.378***	6.479***	6.479***	6.479***	6.479***	6.479***
	(0.027)	(0.024)	(0.024)	(0.024)	(0.024)	(0.024)
Observations	1,867	1,867	1,867	1,867	1,867	1,867
R^2	0.051	0.091	0.123	0.126	0.132	0.132
Adjusted R^2	0.051	0.090	0.121	0.124	0.130	0.130
Residual Std. Error	1.067 (df = 1865)	1.045 (df = 1864)	1.027 (df = 1863)	1.025 (df = 1862)	1.022 (df = 1861)	1.022 (df = 1860)
F Statistic	100.377*** (df = 1; 1865)	92.835*** (df = 2; 1864)	87.001*** (df = 3; 1863)	67.164*** (df = 4; 1862)	56.838*** (df = 5; 1861)	47.341*** (df = 6; 1860)
Note:						*p<0.1; **p<0.05; ***p<0.01

Table 2. Non-Linearity Check

Since p-values of *nbFaces* and *avgStarMeter* were larger than 0.1, they were the only ones considered to be linear.

	Test stat	Pr(>|Test stat|)
movieBudget	2.4500	0.01438
duration	-7.0508	2.498e-12
nbNewsArticles	-8.4143	< 2.2e-16
avgStarMeter	1.6231	0.10473
nbFaces	0.5200	0.60315
movieMeter_IMDBpro	6.6297	4.394e-11
Tukey test	-10.4583	< 2.2e-16

Table 3. Heteroskedasticity Check

Since $p < 0.05$, the heteroskedasticity issue exists. Thus, *coeftest* was used to eliminate it.

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 362.4914, Df = 1, p = < 2.22e-16
```

Table 4a. Outliers Check: *movieBudget*

	rstudent	unadjusted p-value	Bonferroni p
954	-4.216131	2.6043e-05	0.048622

Table 4b. Outliers Check: *duration*

	rstudent	unadjusted p-value	Bonferroni p
383	-4.346057	1.4607e-05	0.027272
183	-4.302662	1.7750e-05	0.033140
1748	-4.282114	1.9454e-05	0.036321

Table 4c. Outliers Check: *nbNewsArticles*

	rstudent	unadjusted p-value	Bonferroni p
477	-8.003088	2.1130e-15	3.9449e-12
1529	-4.306017	1.7486e-05	3.2646e-02
954	-4.218356	2.5790e-05	4.8149e-02

Table 4d. Outliers Check: *nbFaces*

	rstudent	unadjusted p-value	Bonferroni p
1529	-4.284663	1.9234e-05	0.035911

Table 4e. Outliers Check: *movieMeter_IMDBpro*

	rstudent	unadjusted p-value	Bonferroni p
1529	-4.231731	2.4316e-05	0.045398
954	-4.215745	2.6087e-05	0.048704

Table 5. Collinearity Check

Since all values are smaller than 4, no collinearity issue was detected between predictors.

	GVIF	Df	GVIF^(1/(2*Df))
movieBudget	1.148362	1	1.071617
poly(duration, 2)	1.579247	2	1.121018
poly(nbNewsArticles, 3)	1.164980	3	1.025777
nbFaces	1.019149	1	1.009529
western	1.010999	1	1.005484
drama	1.280346	1	1.131524
animation	1.063715	1	1.031366
ranking_bin	1.607845	9	1.026734

Table 6. Model Comparison (Before Cross Validation and Ordered by Adjusted R^2)

Model	**Multiple** R^2	**Adjusted** R^2
***model* 1** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + maturityRating + poly(nbNewsArticles, 2) + nbFaces + action + adventure + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin + poly(movieMeter\_IMDBpro, 2) + avgStarMeter)$	0.7536	0.7491
***model* 4** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + maturityRating + poly(nbNewsArticles, 4) + nbFaces + action + adventure + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin + poly(avgStarMeter, 3))$	0.7517	0.7468
***model* 3** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + maturityRating + poly(nbNewsArticles, 4) + nbFaces + action + adventure + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin + poly(movieMeter\_IMDBpro) + poly(avgStarMeter, 3))$	0.7517	0.7466
***model* 2** $= lm(imdbScore \sim movieBudget + poly(duration, 3) + maturityRating + poly(nbNewsArticles, 3) + nbFaces + action + adventure + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin + movieMeter\_IMDBpro + poly(avgStarMeter, 2))$	0.7513	0.7464
***model* 8** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + poly(nbNewsArticles, 3) + nbFaces + action + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin)$	0.7494	0.7456
***model* 9** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + poly(nbNewsArticles, 3) + nbFaces + action + scifi + thriller + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin)$	0.7493	0.7456
***model* 11** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + poly(nbNewsArticles, 3) + nbFaces + western + drama + animation + ranking\_bin)$	0.7481	0.7455
***model* 7** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + poly(nbNewsArticles, 3) + nbFaces + action + adventure + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin)$	0.7494	0.7454
***model* 5** $= lm(imdbScore \sim movieBudget + poly(duration, 3) + poly(nbNewsArticles, 3) + nbFaces + action + adventure + scifi + thriller + musical + romance + western + sport + horror + drama + war + animation + crime + ranking\_bin + poly(avgStarMeter, 2))$	0.7497	0.7453
***model* 10** $= lm(imdbScore \sim movieBudget + poly(duration, 2) + poly(nbNewsArticles, 3) + nbFaces + action + scifi + thriller + romance + western + sport + horror + drama + war + animation + crime)$	0.3757	0.3696

Table 7a. Final Model Coefficients

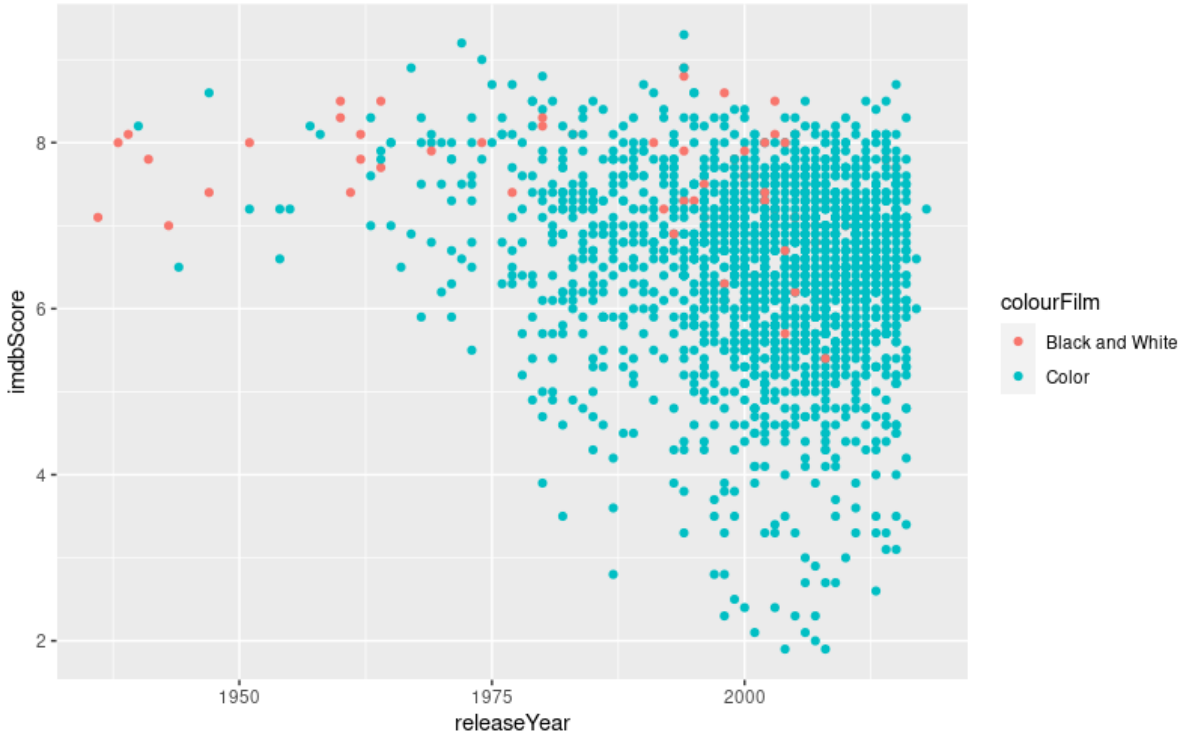
	Dependent variable:
	imdbScore
Movie Budget	-0.00***
	(0.00)
Duration	5.15***
	(0.65)
Duration^2	-0.70
	(0.57)
News Articles	4.90***
	(0.57)
News Articles^2	-2.16***
	(0.55)
News Articles^3	0.52***
	(0.55)
Number of Faces	-0.01
	(0.01)
Western	0.22**
	(0.10)
Drama	0.13***
	(0.03)
Animation	0.37***
	(0.13)
Ranking Bin 2	-0.34***
	(0.06)
Ranking Bin 3	-0.53***
	(0.06)
Ranking Bin 4	-0.70***
	(0.06)
Ranking Bin 5	-0.89***
	(0.06)
Ranking Bin 6	-1.05***
	(0.06)
Ranking Bin 7	-1.24***
	(0.06)
Ranking Bin 8	-1.46***
	(0.06)
Ranking Bin 9	-1.84***
	(0.06)
Ranking Bin 10	-2.94***
	(0.07)
Constant	7.67***
	(0.05)
Observations	1,860
Log Likelihood	-1,486.69
Akaike Inf. Crit.	3,013.38
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 7b. Final Model Predictor Significance

```
                           Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)               7.6718e+00  4.4665e-02 171.7641 < 2.2e-16 ***
movieBudget              -6.2021e-09  9.2953e-10  -6.6723 2.519e-11 ***
poly(duration, 2)1        5.1461e+00  6.4624e-01   7.9631 1.677e-15 ***
poly(duration, 2)2       -7.0206e-01  5.0046e-01  -1.4028  0.160669
poly(nbNewsArticles, 3)1  4.8977e+00  5.3011e-01   9.2389 < 2.2e-16 ***
poly(nbNewsArticles, 3)2 -2.1632e+00  4.1880e-01  -5.1652 2.402e-07 ***
poly(nbNewsArticles, 3)3  5.1630e-01  4.5007e-01   1.1472  0.251314
nbFaces                  -9.6303e-03  6.9472e-03  -1.3862  0.165685
western                   2.2427e-01  7.8165e-02   2.8692  0.004116 **
drama                     1.3140e-01  2.7552e-02   4.7693 1.849e-06 ***
animation                 3.7058e-01  6.4285e-02   5.7645 8.188e-09 ***
ranking_bin2             -3.3732e-01  4.7448e-02  -7.1094 1.166e-12 ***
ranking_bin3             -5.3331e-01  4.5294e-02 -11.7743 < 2.2e-16 ***
ranking_bin4             -6.9696e-01  5.2445e-02 -13.2894 < 2.2e-16 ***
ranking_bin5             -8.9032e-01  5.2267e-02 -17.0341 < 2.2e-16 ***
ranking_bin6             -1.0520e+00  5.3549e-02 -19.6449 < 2.2e-16 ***
ranking_bin7             -1.2397e+00  5.5175e-02 -22.4691 < 2.2e-16 ***
ranking_bin8             -1.4558e+00  5.5300e-02 -26.3256 < 2.2e-16 ***
ranking_bin9             -1.8448e+00  5.8880e-02 -31.3323 < 2.2e-16 ***
ranking_bin10            -2.9422e+00  8.1192e-02 -36.2372 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

5.2 Plots

Plot 1. The relationship between black and white and colour films.



Plot 2. Non-Linearity Check.