Statistical Modeling and Data Analysis

Emilie Krutnes Engen
Clara Cabañas Pujadas

Statistical Learning
Carlos III University of Madrid

Universidad
Carlos III de Madrid

Contents

List of Tables

List of Figures

1 Introduction

You may have experienced the difficulty related to how to decide what movie you would like to watch. IMDB[1] is a website that contains information about thousands of movies and is considered a reliable source for movie related information, such as ratings, reviews, and technical details about every movie. The score of a movie in IMDB comes from a user based rating mechanism where all users can score the movies they have seen. But what exactly is influencing the users' rating of a movie? In this study the factors that may influence movie rating will be investigated. Based on a dataset with information for about 5000 movies scrapped from IMDB this study focuses on determining which variables are important in the rating of movies and attempts to discover patterns in the IMDB movie dataset.

In this study several Statistical Learning methods will be tested, and their performance will be evaluated by assessing their ability to provide a strong estimation of the IMDB rating for a movie. In the development of a predictive regression model a multiple linear regression model will be tested, along with stepwise regression, Ridge and Lasso models, a decision tree and Random Forest. Furthermore, the movie rating prediction problem will be analyzed by treating it as a classification problem, by binning the continuous response variable into three categories: good, average and bad. Based on this categorization of the continuous output variable, the K-Nearest Neighbours (KNN) algorithm is tested, in addition to a logistic regression model. Finally, Unsupervised Classification is performed on the data by using K-Means, a clustering algorithm that has the goal of discovering underlying patterns in the data.

The remainder of this paper is organized in seven sections. Section 1 introduces the variables used in this study and explains how the data was collected. It further presents a univariate analysis of the variables, to provide important insights in how the different variables are distributed. In Section 2 the relationships between the different variables are investigated by performing a correlation analysis to discover possible collinearities in the data set. Besides, the identified discrepancies in the dataset are presented and the initial cleaning process is explained. Section 3 presents the different prediction models developed and tested in this study. The results from these models are discussed and the different models are evaluated based on their prediction strength. In Section 4 the prediction problem is treated as a classification problem and solved using K-Nearest Neighbours and Logistic regression. In Section 5 unsupervised classification is conducted using the K-Means clustering algorithm, in order to discover undefined patterns in the dataset. Finally, Section 6 summarizes the analysis performed and presents the conclusion extracted from the study, along with ideas for further research.

[1] `http://www.imdb.com/`

2 Descriptive Statistics

2.1 Variable Description

The dataset subject to this study is the IMDB Movie Metadata dataset, a collection of data from 5043 unique movies collected from Kaggle[2] by Sun (2016). It contains scraped information from IMDB, including movie metadata as title, director, actor names, total budget, genre, rating, country, number of Facebook likes for the director, number of Facebook likes for the actors, etc. The scraping was conducted using the Python library *Scrapy*[3]. In addition to this, a human face detection algorithm was used to extract the number of faces in the movie poster. The face detection was conducted using the Python library *dlib*[4]. In total, the dataset is composed of 28 different variables, presented in Table 1.

#	Variable name	Data type	Description
1	color	String	color or black and white movie
2	director_name	String	name of movie director
3	num_critic_for_reviews	Numeric	number of critics with review of movie
4	duration	Numeric	duration of movie in minutes
5	director_facebook_likes	Numeric	number of facebook likes for director
6	actor_3_facebook_likes	Numeric	number of facebook likes for third actor
7	actor_2_name	String	name of second actor
8	actor_1_facebook_likes	Numeric	number of facebook likes for first actor
9	gross	Numeric	total gross in local currency
10	genres	String	genres related to the movie
11	actor_1_name	String	name of first actor
12	movie_title	String	title of movie
13	num_voted_users	Numeric	number of users having rated the movie
14	cast_total_facebook_likes	Numeric	number of facebook likes for cast in total
15	actor_3_name	String	name of third actor
16	facenumber_in_poster	Numeric	number of faces in movie poster
17	plot_keywords	String	important keywords from plot
18	movie_imdb_link	String	link to movie on IMDB
19	num_user_for_reviews	Numeric	number of users having reviewed the movie
20	language	String	movie language
21	country	String	country of production
22	content_rating	String	suitability rating for movie
23	budget	Numeric	total budget for movie in local currency
24	title_year	Numeric	releasing year
25	actor_2_facebook_likes	Numeric	number of facebook likes for second actor
26	imdb_score	Numeric	average of user rating on IMDB
27	aspect_ratio	Numeric	supported aspect ratio for movie
28	movie_facebook_likes	Numeric	number of facebook likes for movie

Table 1: Description of all variables in the dataset

[2] `http://www.kaggle.com/`
[3] `https://scrapy.org/`
[4] `https://pypi.python.org/pypi/dlib`

From Table 1 it appears that most variables are either text strings (qualitative) or numerical values. The quantitative numerical variables can be further divided into continuous and discrete variables. Figure 1 shows a diagram with the representation of the variables in the dataset, where roughly half of the variables are related directly to the movie itself, like *movie title*, *title year* or *duration*, while the other half is related to people involved in the production or rating of the movies.

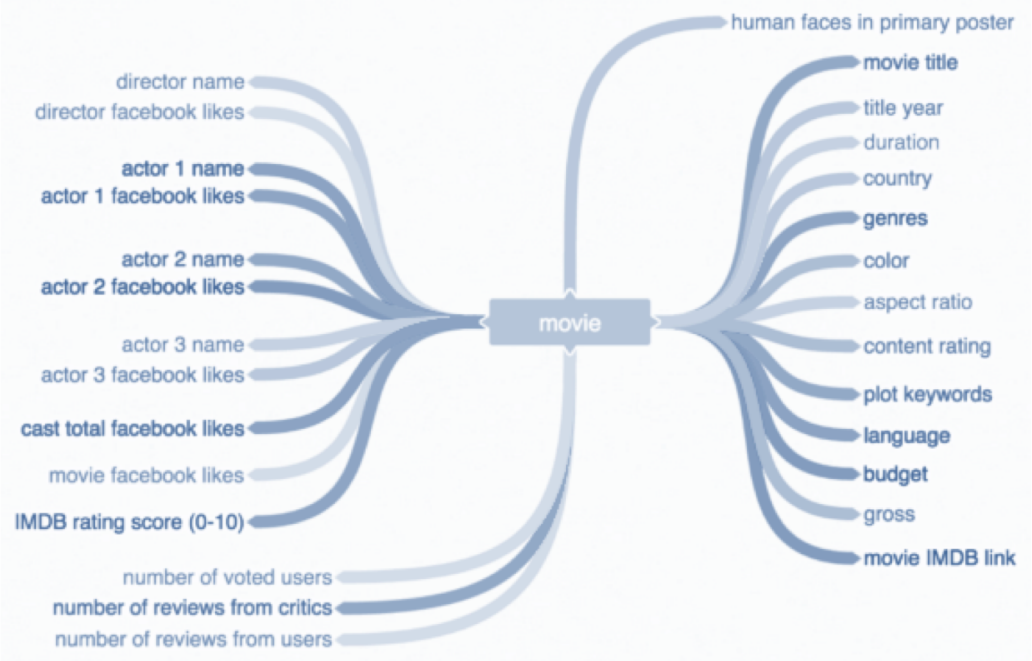

Figure 1: Illustration of the variables in the dataset

Descriptive Statistics: Quantitative Variables					
Statistic	N	Mean	St. Dev.	Min	Max
num_critic_for_reviews	4,993	140.194	121.602	1	813
duration	5,028	107.201	25.197	7	511
director_facebook_likes	4,939	686.509	2,813.329	0	23,000
actor_3_facebook_likes	5,020	645.010	1,665.042	0	23,000
actor_1_facebook_likes	5,036	6,560.047	15,020.760	0	640,000
gross	4,159	48,468,408	68,452,990	162	760,505,847
num_voted_users	5,043	83,668.160	138,485.300	5	1,689,764
cast_total_facebook_likes	5,043	9,699.064	18,163.800	0	656,730
facenumber_in_poster	5,030	1.371	2.014	0	43
num_user_for_reviews	5,022	272.771	377.983	1	5,060
budget	4,551	39,752,620	206,114,898	218	12,215,500,000
title_year	4,935	2,002.471	12.475	1,916	2,016
actor_2_facebook_likes	5,030	1,651.754	4,042.439	0	137,000
imdb_score	5,043	6.442	1.125	1.600	9.500
aspect_ratio	4,714	2.220	1.385	1.180	16.000
movie_facebook_likes	5,043	7,525.965	19,320.440	0	349,000

Table 2: Summary of the quantitative variables in the dataset

For the categorical variables , the number of appearances for each of the categories in each variable is taken into account. Here, the *actor_1_name*, *actor_2_name* and *actor_3_name* variables are combined and grouped by actor names, represented in the variable *actor_name*.

Number of observations						
Variable	Min	Q1	Median	Mean	Q3	Max
color	19	114	209	1680	2510	4820
director_name	1.0	1.0	1.0	2.1	2.0	104.0
actor_name	1.0	1.0	1.0	2.4	2.0	49.0
genres	1.00	1.00	1.00	5.52	4.00	236.00
language	1	1	2	105	8	4700
country	1	1	3	76	11	3810
content_rating	1	8	20	265	114	2120

Table 3: Summary of the frequencies of the qualitative variables in the dataset

2.2 Initial Data Cleaning

Duplicated movies The first problem that arises when looking at the movies in the dataset, is that some movies are replicated. For instance, King Kong appears three times in the original dataset. However, the title is not enough to determine whether a movie is replicated or not, since other movies have the same title, but are not the same movies. As a result, the assumption that movies with the same title and the same director are duplicates is made. The duplicated movies are thus removed from the data set. The total number of duplicated movies removed from the data set is 134.

Variables inappropriate for prediction Initially, variables *movie_title*, *plot_keywords* and *movie_imdb_link* are removed, since they are textual variables that do not provide any useful information for this analysis.

Lack of significance in categorical variables The categorical variable *color* contains only two categories: *Color* and *Black White*, indicating whether the movie was filmed in color or in black and white. However, the frequency of *Black White* is only 209, which is less than the 5% of the data. As a result this variable is not consider in further analysis due to the imbalance of its values.

From Table 3 it can be observed that some of the categorical variables have a high number of levels. This is especially the case of the variables *country*, *language*, *actor_1_names*, *actor_2_names*, *actor_3_names* and *director_names*. From Table 3 it can be seen that even after joining the three actor variables into the variable *actor_name*, there is still a large number of actors with only one appearance. This is also true for the directors. Analysing variables where a large number of categories only appear once in the data, does not make sense as the results would not be statistically significant. One way of handling this problem is to reduce the number of categories by combining those that are closely related. For actors and directors there is no simple way of performing this task. Therefore, the variables have been removed from the dataset.

The *language* variable presents the same problem. One alternative is to group the categories into *English* and *Other*, since *English* is the category with the highest number of appearances. The same could be done for the *country* variables, where the categories would be grouped into continents. However, the number

of observations for English movies is 4700 out of 5043, and the number of movies from the US is 3810 out of 5043. This accounts for 93.20% and 77.55% of the total number of observations for each variable. As a result, the described grouping of these two variables is considered inappropriate. An alternative would be to only consider movies in English or movies from the US. For the purpose of this study, the *language* variable has been eliminated and only movies from the US will be considered. This is due to the fact that the variables *gross* and *budget* are given in local currencies, making the values in the variables inconsistent. Since *gross* and *budget* seem to be be important variables, the problem will be tackled by considering only movies from the US. Converting these variables to one currency is a complex process as the exchange rate shows large variations in time, and the currencies of every country have changed along the years. When only regarding movies from the US, the number of observations is reduced to 3810. The number of observations is still high enough to perform a valid analysis, so this option is considered absolutely appropriate.

The result from the initial cleaning process gives a total number of 3711 observations. The reduced dataset contains a total number of 18 predictors, with two categorical variables. The categorical variables kept are transformed to dummy variables, to enable the usage of these in prediction. The remaining explanatory variables are presented in Table 4.

#	Variable name	Data type
1	num_critic_for_reviews	Numeric
2	duration	Numeric
3	director_facebook_likes	Numeric
4	actor_1_facebook_likes	Numeric
5	actor_2_facebook_likes	Numeric
6	actor_3_facebook_likes	Numeric
7	gross	Numeric
8	budget	Numeric
9	facenumber_in_poster	Numeric
10	title_year	Numeric
11	aspect_ratio	Numeric
12	genres	Categorical
13	num_voted_users	Numeric
14	cast_total_facebook_likes	Numeric
15	num_user_for_reviews	Numeric
16	content_rating	Categorical
17	movie_facebook_likes	Numeric

Table 4: Remaining explanatory variables after initial cleaning process

2.3 Univariate Analysis

After cleaning the data, a univariate analysis of the quantitative variables in the dataset is performed. The main focus is on the response variable, *imdb_score* which is the variable that wants to be predicted. From the histogram in Table 2 it can be seen that the distribution of the IMDB rating for the sample data is approximately normal, with a mean of 6.4. The distribution is slightly left skewed, as the number of movies below the mean is larger than the number of movies above the mean. From the descriptive statistics in Table **??** it can be spotted that the lowest score obtained in this data set is 1.6 and the highest is 9.5. The possible score range a movie can get is between 0 and 10.

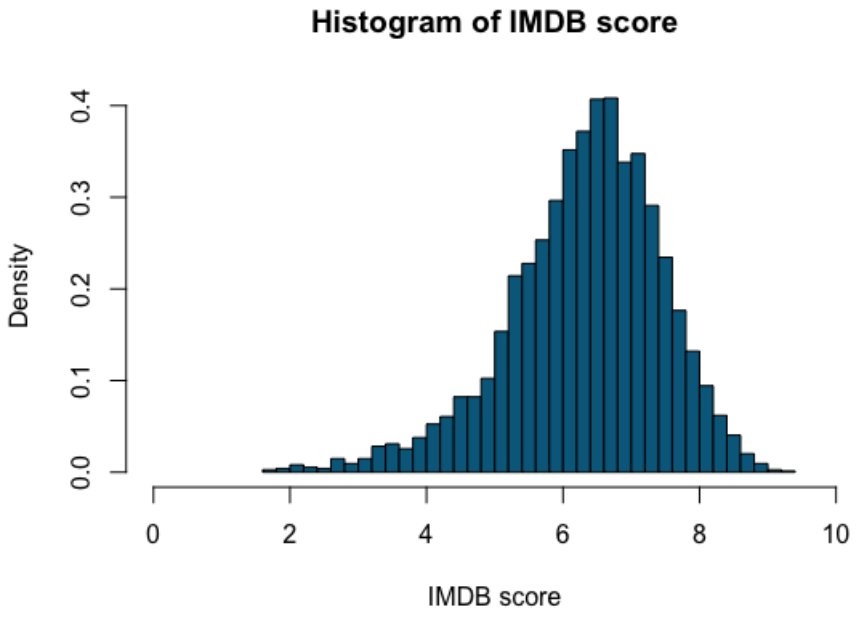

Figure 2: Histogram showing distribution of IMDB score

The next step in the univariate analysis is plotting histograms illustrating the distribution of the remaining quantitative variables. From Figure 3, it can be seen how *duration* shows an approximately normal distribution with a mean of 5. The distribution is slightly right skewed, the long tail suggests the occurrence of some outliers to the right. The histogram of *title_year* shows that a large part of the movies in the dataset were produced after 1990, which may create some bias in this predictor, due to the fact that older movies are not well represented in the dataset. Furthermore, the distribution for the other predictors may indicate that some log-transformation is appropriate.

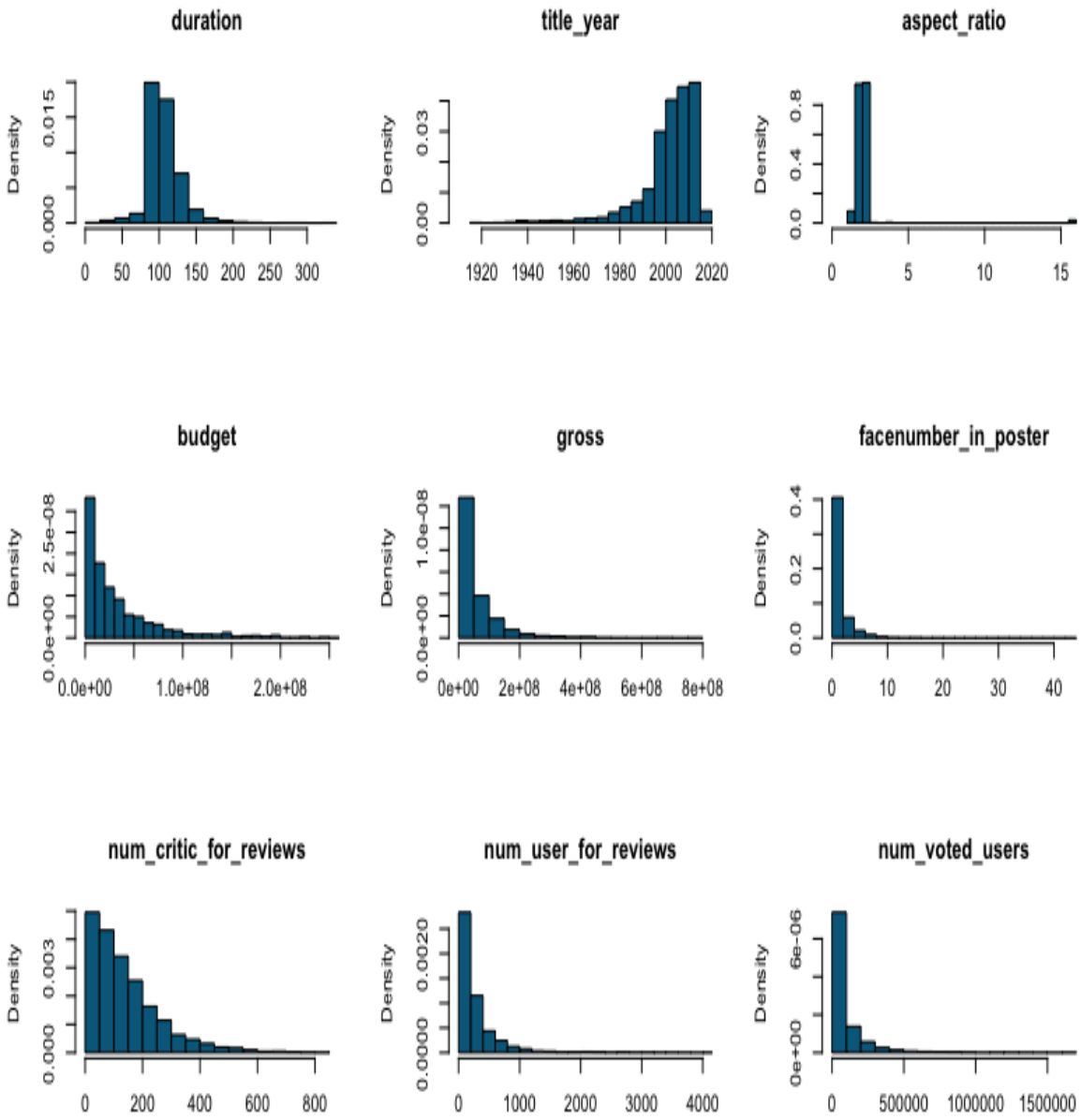

Figure 3: Histograms showing the distribution of the explanatory variables

The histograms illustrating the distribution of the remaining quantitative variables are presented in Figure 4. The distributions of the variables *movie_facebook_likes*, *director_facebook_likes*, *actor_1_facebook_likes*, *actor_2_facebook_likes*, *actor_3_facebook_likes* and *cast_total_facebook_likes* show that a large fraction of the movies in the dataset have a only a few Facebook likes. Thus, it is not expected from these predictors to have a strong impact on the IMDB score.

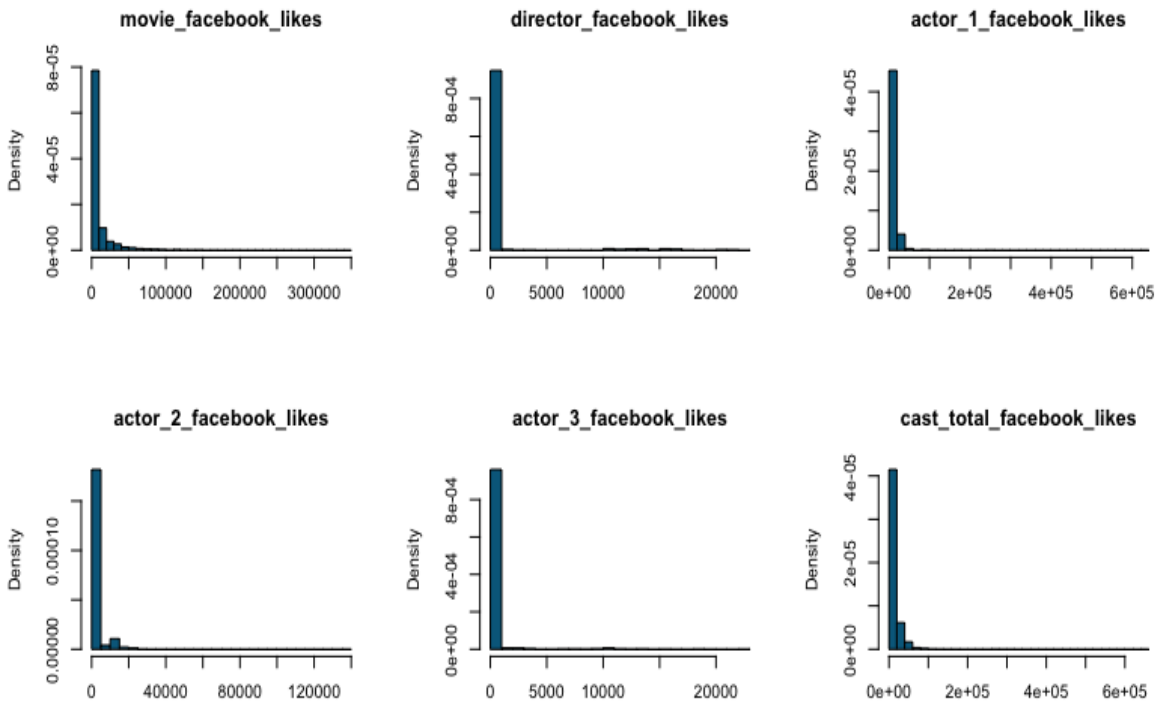

Figure 4: Histograms showing the distribution of the explanatory variables

In Figure 5 the distributions of some of the predictors are described using a log transformation. By taking the logarithm they all show an approximately normal distribution, with a slightly left skew.

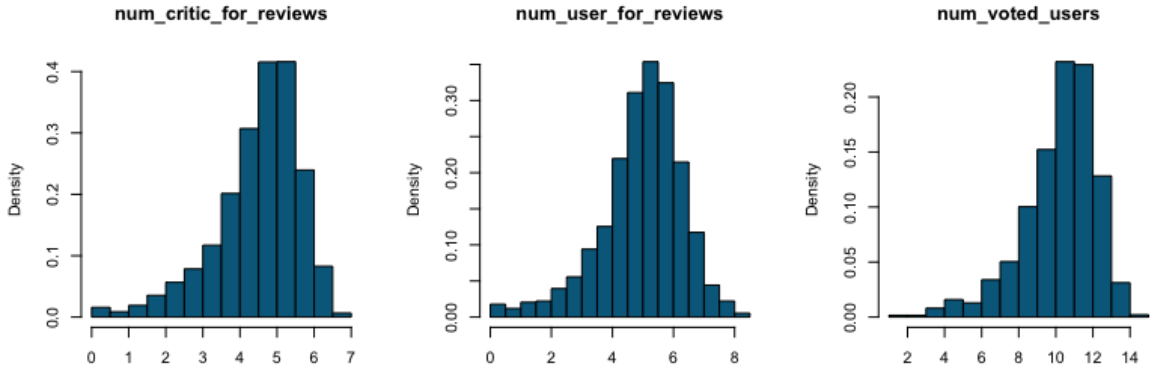

Figure 5: Histograms showing log distribution for explanatory variables

3 Multidimensional Analysis

3.1 Data Discrepancies

Before proceeding with the analysis, it is important to identify discrepancies in the dataset and evaluate different strategies for overcoming these problems.

3.1.1 Handling Missing Data

One thing to look out for when performing an analysis on a data set is missing data. A high number of missing data in the dataset may lead to problems in developing prediction or classification models. Normally a maximum threshold of 5% of the total number of observations is considered safe for large datasets. If the missing data for a certain variable is higher than this threshold, removing the variable from the dataset should be considered, although there are some methods for dealing with missing data that do not imply deleting the variable. The starting point is then investigating if any of the variables in the dataset exceeds the maximum threshold. In Table 5 the 18 remanining variables in the dataset are presented, sorted decreasingly by number of missing values, along with their percentage of missing data. It can be seen that the variable *gross* contains around 15% of missing observations. Variables *budget* and *aspect_ratio* are also above the threshold value, with percentages of missing data of 8.03% and 5.96% respectively. The other variables are all below the maximum threshold in terms of missing data.

Missing data	
Variable	Count (%)
gross	15.04
budget	8.03
aspect_ratio	5.96
director_facebook_likes	2.13
title_year	2.38
num_critic_for_reviews	1.10
actor_3_facebook_likes	0.49
facenumber_in_poster	0.49
num_user_for_reviews	0.46
actor_2_facebook_likes	0.35
actor_1_facebook_likes	0.27
duration	0.16
genres	0.16
num_voted_users	0.16
cast_total_facebook_likes	0.16
content_rating	0.16
imdb_score	0.16
movie_facebook_likes	0.16

Table 5: Sorted list of variables with highest percentage of missing data

Figure 6 gives very valuable information about the missing data in the dataset. The histogram is a graphical representation of the values explained in Table 5. The pattern plot shows the distribution of the missing data accross the variables. From the plot it can be seen that there is a 0.162% of the observations that have missing values for all the variables. In addition, it can be seen that most of the

observations that have missing values for the *gross* variable also have missing missing values for the *budget* variable.

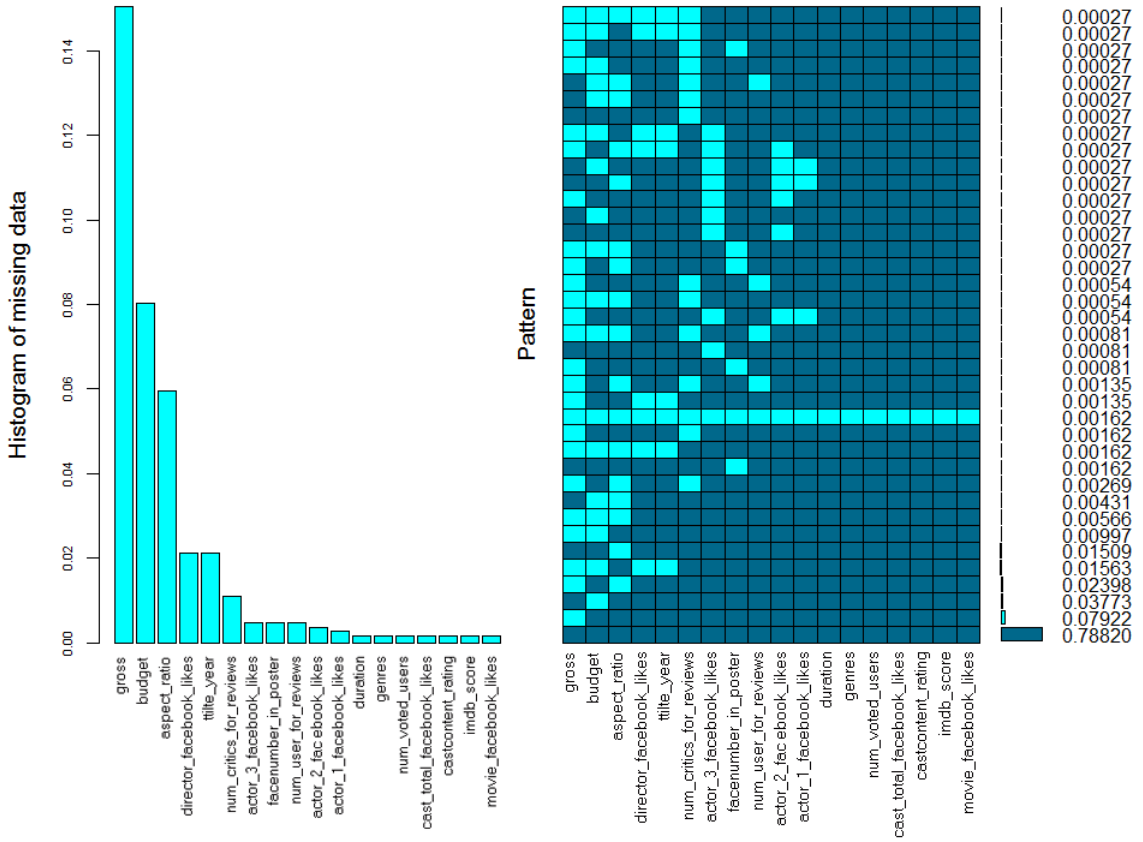

Figure 6: Plots showing the distribution of the missing values in the dataset

After analyzing the results of the missing data exploration, the variable *aspect_ratio* is removed from the dataset, due to having 5.96% of missing values, which is above the maximum threshold set for acceptance. Regarding the *gross* and *budget* variables, despite them having percentages of missing values greater than the threshold, the treatment of the missing data has been different, considering the importance of the variables for explaining movie rating. In this case, the observations with missing values for both variables have been left out of the study. For the rest of the missing data in each of the two variables, a missing data imputation technique has been applied: the missing values have been substituted by the mean value of all the valid values in each variable.

3.1.2 Detection of outliers

During the exploratory analysis of the data, some potential outliers were spotted in variable *duration*. Two observations had a movie duration longer than 330 minutes. Intuitively this is considered quite unlikely for a movie. When looking into this in detail, it was observed that they were in fact TV-series. Thus these observations were removed from the dataset. In addition to this, a high number of movies with duration longer than 200 minutes were spotted and investigated, as 3 hours and 20 minutes is also considered long for a movie. The total number of movies with a duration longer than 200 minutes in the data set is 19. These were compared to data from IMDB (1990-2017) and corrected. In total, 14 movies were corrected. This may indicate that the data from IMDB (1990-2017) was incorrect at the time of the

data scraping or an error in the scraping process.

3.2 Data Cleaning

In order to prepare the dataset for prediction and classification, some further cleaning of the data is necessary. This section provides a description of the cleaning performed to make the dataset appropriate for further analysis.

Handling missing and incorrect data As previously described, variable *aspect_ratio* was deleted due to having a high number of missing data among its values. Variables *gross* and *budget* had a very high number of missing values too, but they were not kept out of the analysis. Instead, the missing values were imputed to the mean value of all the values in each column. In addition to this, both variables presented a problem of data inconsistency, since their values contained money values in different currencies that could not be corrected. The dataset was filtered and only the movies from the US were kept, so the problem of the differences in currencies was solved. However, the *gross* and *budget* variables contained data of currencies accross the years. Thus, the inflation factors across years had to be taken into consideration, and all the US dollars had to be normalized into one basis, which in this case was year 2016. Older movies were therefore adjusted for inflation, assuming the inflation rate is equal to the average US inflation rate between 1913 and 2016, which is 3.018 %.

Correcting for multicollinearity From the correlation analysis carried out for all the numerical variables, some multicollinearity was detected. For instance, the variable *cast_total_facebook_likes* had a high correlation with the variable *actor_1_facebook_likes*, which makes sense since the *cast_total_facebook_likes* variable is dependent on the Facebook likes for all actors. Furthermore, the variable *num_user_for_reviews* presented a high correlation with variable *num_voted_users*, whereas variable *movie_facebook_likes* appeared to be highly correlated with *num_critic_for_reviews*. As a result the following variables were thus removed from the analysis: *cast_total_facebook_likes*, *num_user_for_reviews* and *movie_facebook_likes*.

Removing incorrect variables Although the variable *content_rating* includes many levels it was not removed in the initial cleaning phase, due to possibility of grouping similar content rating together to obtain groups with a significant number of observations. However, when investigating how the variables should be grouped, we discover that the a lot of the ratings did not match the US standard for content rating. In addition a lot of the movies in the dataset did not have any rating. Due to the inconsistency in this variable it is eliminated in further analysis.

3.3 Bivariate Analysis

In this section, the relationship between pairs of values in the dataset will be studied by means of the scatter plots. In particular, the scatter plots between some variables and the output variable, *imdb_score* are obtained. First of all, the relationship between the duration of a movie and its IMDB score is studied by means of a scatter plot.

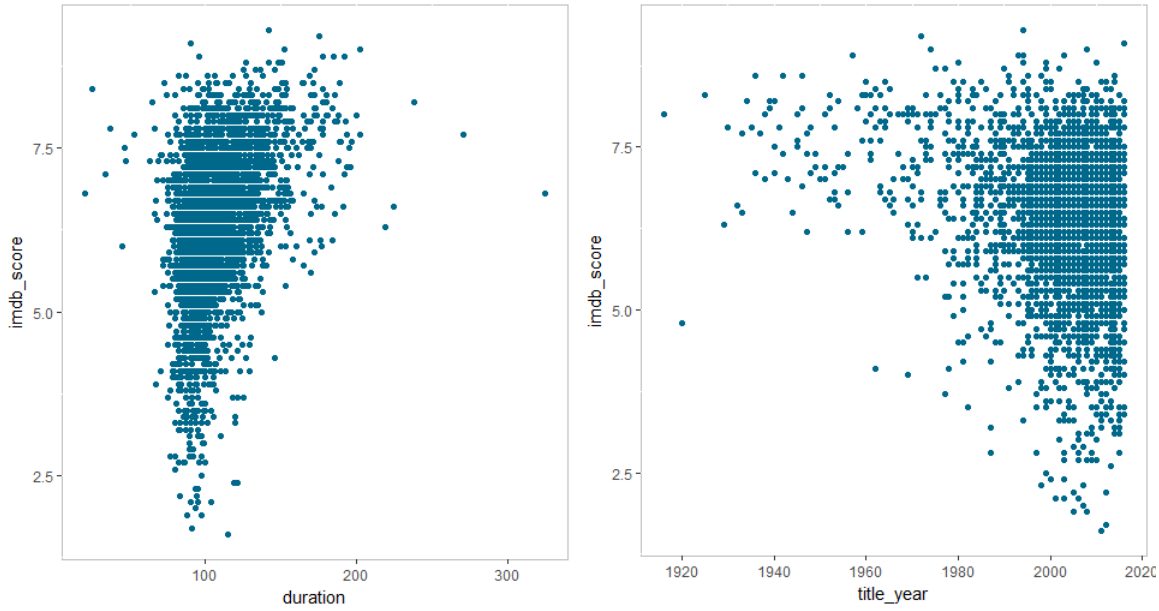

Figure 7: Scatter plots of *duration* and *imdb_score* and $title_year and imdb\_score$

The scatter plot on the left in Figure 7 shows a prevailing movie length of around 100 minutes, along with a slight trend of long movies getting a high IMDB score. Next, a scatter plot representing the relationship between the year a movie was released and its IMDB score is obtained. The scatter plot on the right in Figure 7 shows that most movies in the IMDB collection were released after the 90s, which makes sense due to the advances in technology. In addition, the plot shows how old movies are always very highly rated on IMDB, which brings up the question: were old movies really better than today's movies? To continue with, the relationship between the variables *gross* and *imdb_score* is studied.

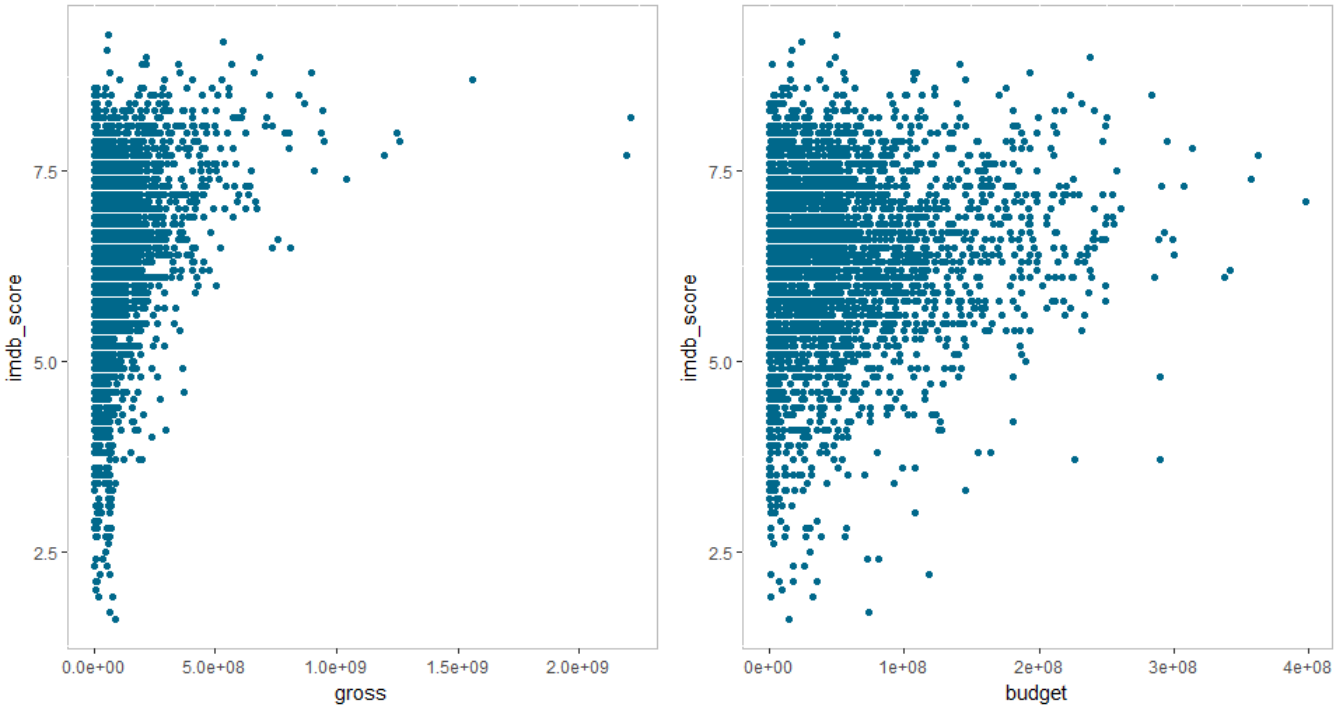

Figure 8: Scatter plots of *gross* and *imdb_score* and *budget* and *imdb_score*

Figure 8 shows on the left the scatter plot between the two variables, in which a majority of low-grossing movies can be detected. In addition, the plot also reveals that high-grossing movies tend to have a high score on IMDB. Finally, the relationship between *budget* and *imdb_score* is analyzed by means of a scatter plot. The scatter plot of *budget* and *imdb_score* can be seen on the right in Figure 8, and it shows a clear majority of movies with budgets lower than 100M$, whereas there exists a tendency of expensive movies, i.e. high budget movies with very high IMDB scores. This leads to thinking that the more expensive, the more valued it is by the public.

3.4 Correlation Analysis

In order to study the relationship between the variables in the dataset, the correlation matrix for all the quantitative variables is obtained. The correlation matrix is presented in Figure 9 , and it contains all the correlations between every pair of variables, ordered by the first principal component (FPC). The strongest correlations (greater than 0.5) happen between *num_voted_users* and *num_critic_for_reviews* whith a correlation of 0.62, and between *actor_2_facebook_likes* and *actor_3_facebook_likes*, with a correlation of 0.55. The high positive correlation between *num_voted_users* and *num_critic_for_reviews* is expectable, since the more popular a movie is, i.e. the more votes a movie gets, the more likely it is for critics to make reviews of it. The high positive correlation between *actor_2_facebook_likes* and *actor_3_facebook_likes* is also intuitive, since in general, movies starring very popular actors as protagonists, also star popular actors as deuteragonists. Besides, the correlation matrix plot also reveals a high positive correlation (0.47) between the output variable, *imdb_score*, and the variable *num_voted_users*, which suggests that users tend to leave a vote for a movie when the movie is good. In addition, the correlation between *gross* and *num_voted_users* is of 0.46, which leads to the conclusion that the more popular a movie is, the more money it makes. On the other hand, a high negative correlation (-0.47) is found between the variables *gross* and *title_year*, which does not seem straightforward, since it means that old movies made more money than new movies (taking into account that the values of the *gross* variable have been adjusted appropriately according to the inflation rate).

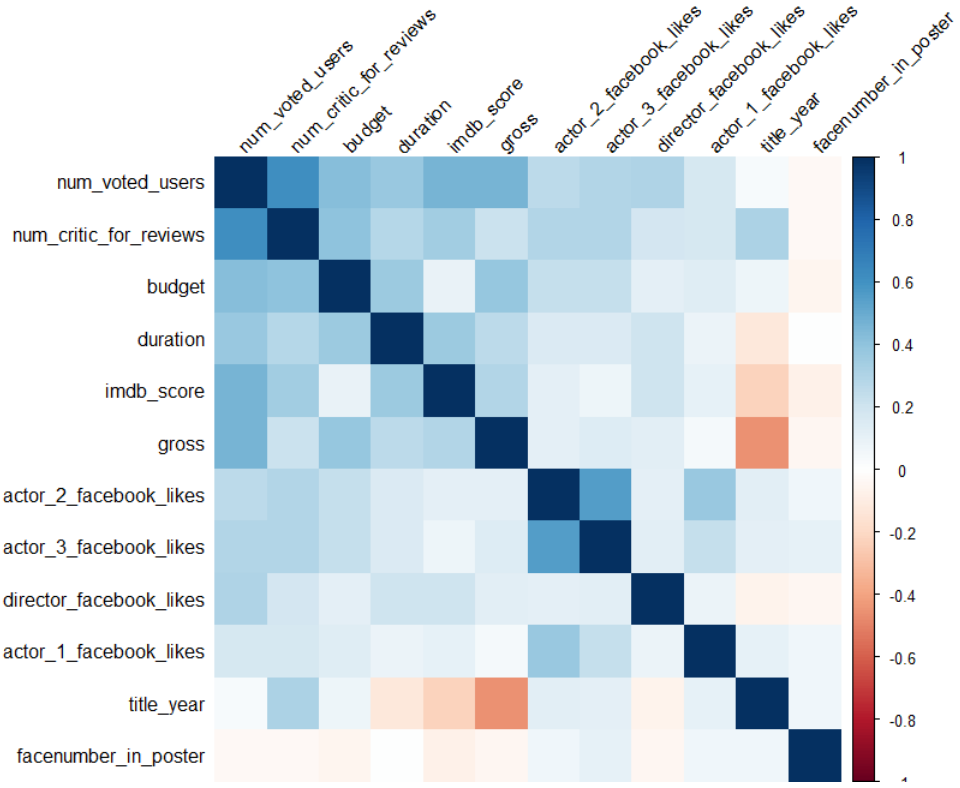

Figure 9: Correlation matrix for all quantitative variables

When regarding the *imdb_score* we see from the correlation matrix that the strongest positive correlation is with *num_voted_users*, followed by *num_critic_for_reviews* and *duration*. It further have a slightly negative correlation with *title_year*. This may indicate that these variables are the ones that impact the *imdb_score* the most. However the correlations between *imdb_score* and the explanatory variables in this data set is not particularly strong, which may reduce the accuracy in the prediction or classification of the IMDB rating, with basis in the predictors from this dataset.

4 Predictive Regression Models

4.1 Multiple Linear Regression

In the simplest setting the IMDB rating can be predicted using a standard linear regression model.

$$Y = \beta_0 + \sum_{j \in p} \beta_j X_j + \varepsilon \tag{1}$$

Here Y is the response vector, corresponding to the IMDB score, while X_j for $j \in p$ is the set of explanatory variables used. This model attempt to fit the regression model using least squares. The least squares procedure estimate the regression coefficients that minimise the residual sum of squares (RSS) (Chatterjee and Hadi, 2015), where

$$RSS = \sum_{i \in n} (y_i - \beta_0 - \sum_{j \in p} \beta_j x_{ij})^2 \tag{2}$$

y_i is the response variable for observation i and x_{ij} is the regressor for predictor j and observation i. We first try to fit the full regression model, including all the explanatory variables. The results from the full regression is given in Table 6.

The full regression model				
	Estimate	Std. Error	t-Test	p-value
intercept	57.0100	3.4588	16.48	0.0000
genres.action	-0.2255	0.0499	-4.52	0.0000
genres.adventure	0.0392	0.0542	0.72	0.4696
genres.crime	0.0443	0.0491	0.90	0.3664
genres.comedy	-0.2025	0.0414	-4.90	0.0000
genres.drama	0.3144	0.0407	7.73	0.0000
genres.fantacy	-0.0614	0.0551	-1.11	0.2656
genres.family	0.0966	0.0614	1.57	0.1159
genres.horror	-0.6054	0.0626	-9.67	0.0000
genres.mystery	0.0092	0.0621	0.15	0.8829
genres.sci.fi	-0.1630	0.0555	-2.94	0.0033
genres.thriller	-0.1398	0.0474	-2.95	0.0032
num_critic_for_reviews	0.0027	0.0002	13.33	0.0000
duration	0.0066	0.0011	6.25	0.0000
director_facebook_likes	0.0000	0.0000	0.25	0.8035
actor_3_facebook_likes	-0.0000	0.0000	-2.92	0.0036
actor_1_facebook_likes	0.0000	0.0000	3.38	0.0007
gross	-0.0000	0.0000	-1.12	0.2613
num_voted_users	0.0000	0.0000	14.24	0.0000
facenumber_in_poster	-0.0252	0.0083	-3.03	0.0024
budget	-0.0000	0.0000	-6.63	0.0000
title_year	-0.0258	0.0017	-14.99	0.0000
actor_2_facebook_likes	-0.0000	0.0000	-0.55	0.5801

Table 6: Results when fitting the full linear regression model on the training data

In Figure 10 the variable importance given by the full regression is plotted. From the full regression model *title_year*, *num_voted_users* and *num_critic_for_reviews* are considered the most important variables. By fitting the linear regression model on the training set, including all predictors, we obtain an adjusted R-squared value of 0.4528. To evaluate the model, we use the mean square error (MSE), where

$$MSE = \frac{RSS}{n} = \frac{1}{n}\sum_{i \in n}(y_i - \hat{y}_i)^2 \tag{3}$$

where n is all the observations in the training or test data set, y_i is the true value of the IMDB score for observation i and $\hat{y}_i$ is the predicted value. When predicting the IMDB score on the test data, we get a mean squared error (MSE) value of 0.6751. From the results obtained some of the predictors are not significant with a significance level of 5 %. We therefore want to try to improve the fit, by reducing the number of predictors.

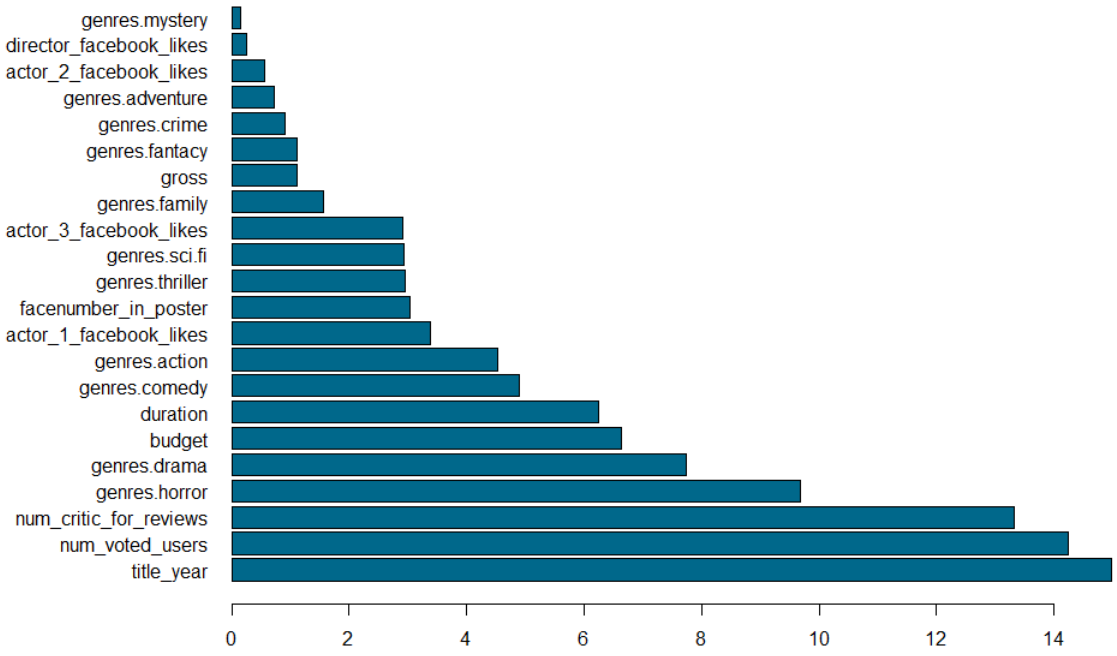

Figure 10: Variable importance plot for the full regression model

4.2 Forward Stepwise Selection

Forward Stepwise Selection is a selection procedure that begins with a regression model with an intercept, but no predictors and then iteratively includes one predictor at a time, until all predictors are included. For each step the Forward Selection algorithm includes the variable that provides the best additional improvement, when added to the model. After all the predictors have been included, the best regression model is selected. There are several statistics that can be used for judging the quality of the models. In this study we have selected the best regression based on the Akaike Information Criterion (AIC) value. When assuming Gaussian errors we have that AIC is given by

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2) \tag{4}$$

where d is the number of predictors, RSS is the residual sum of squares, n is the number of observations and $\hat{\sigma}^2$ is an estimate of the variance of the error ε associated with the response variable Y in Equation 1 (James et al., 2014). AIC adjust the training error for the model size by adding a penalty $2d\hat{\sigma}^2$ to the RSS, where the penalty increases with the number of predictors d included in the model. The results from the regression model obtained with Forward Selection on the training data is presented in Table 7.

The reduced regression model				
	Estimate	Std. Error	t-Test	p-value
intercept	55.3156	2.9683	18.64	0.0000
num_voted_users	0.0000	0.0000	15.00	0.0000
genres.drama	0.3112	0.0399	7.81	0.0000
title_year	-0.0249	0.0015	-16.90	0.0000
num_critic_for_reviews	0.0026	0.0002	13.34	0.0000
genres.horror	-0.6373	0.0597	-10.67	0.0000
budget	-0.0000	0.0000	-7.19	0.0000
duration	0.0063	0.0010	6.20	0.0000
genres.action	-0.2274	0.0474	-4.80	0.0000
genres.comedy	-0.1978	0.0409	-4.83	0.0000
actor_3_facebook_likes	-0.0000	0.0000	-3.60	0.0003
actor_1_facebook_likes	0.0000	0.0000	3.38	0.0007
facenumber_in_poster	-0.0263	0.0083	-3.18	0.0015
genres.thriller	-0.1266	0.0422	-3.00	0.0028
genres.sci.fi	-0.1596	0.0540	-2.95	0.0032

Table 7: Results when fitting the linear regression model on the training data using forward stepwise regression

From the results we have that the number of predictors is reduced from 22 to 14. All predictors are now significant with a 5 % significance level. From the variable importance plot in Figure 11, *title_year*, *num_voted_users* and *num_critic_for_reviews* seems to have the biggest impact on the IMDB score, Y. The plot is similar to the one obtained for the full regression model.

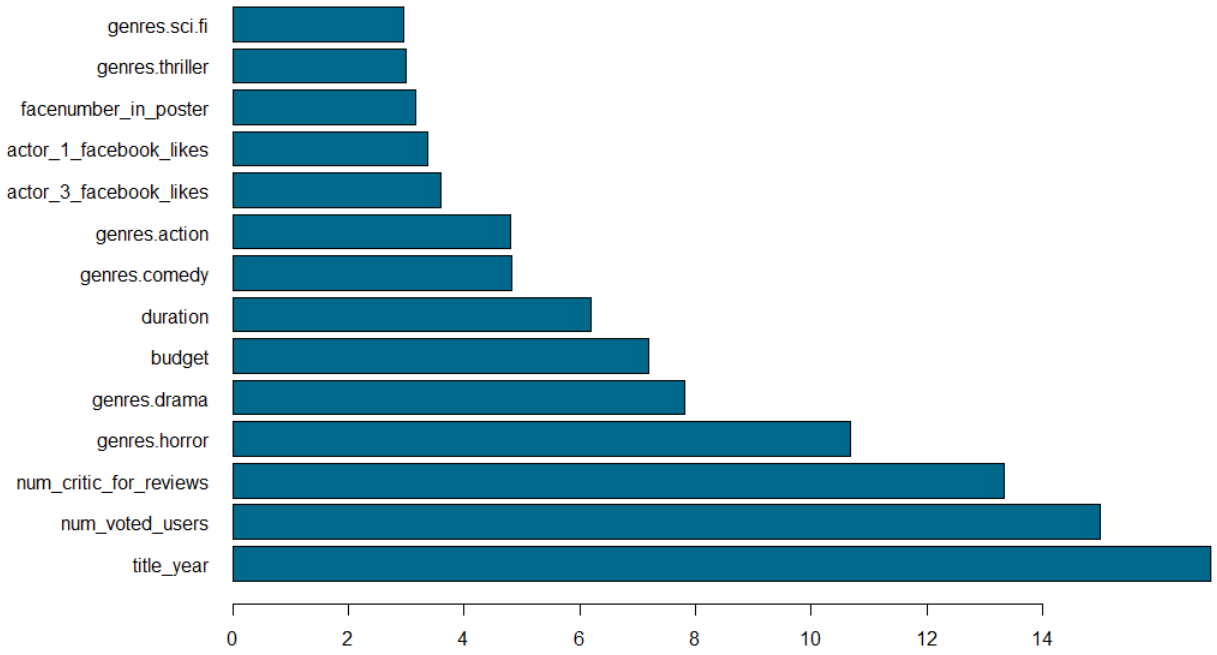

Figure 11: Variable importance plot for the stepwise regression model

When predicting the IMDB score for the test data, using the model obtained from the Forward Selection process on the train data, we get a MSE of 0.6781. This is similar to the MSE obtained from the full regression. We compare the two regression models, using anova. From the anova we obtain a p-value of 0.6201. With a 5 % significance level we cannot reject the reduced model. An alternative to the Forward Selection method is to fit a regression model with p predictors that constrains the coefficient estimates. Here we consider two methods for restricting the coefficient estimates: the Ridge and Lasso regression.

4.3 Ridge Regression

The Ridge regression is a technique that restrict the coefficient estimates by shrinking the estimates to zero. The idea is to shrink the regression estimates in order to reduce the variance in the regression. Ridge is an extension of the least squares procedure, where the coefficients are estimated by minimizing

$$\sum_{i \in n}(y_i - \beta_0 - \sum_{j \in p} \beta_j x_{ij})^2 + \lambda \sum_{j \in p} \beta_j^2 = RSS + \lambda \sum_{j \in p} \beta_j^2 \tag{5}$$

where $\lambda \geq 0$ is a tuning parameter. When the tuning parameter $\lambda = 0$ Ridge is identical to least squares. However when $\lambda > 0$, in addition to minimizing RSS, the Ridge procedure includes a second term, the shrinkage penalty. This is small when the regression coefficients are close to zero, thus it has the effect of shrinking the coefficient estimates (James et al., 2014). For this study we perform 100 estimates of the regression coefficients using a tuning parameter $10^{-2} \leq \lambda \leq 10^5$. In Figure 12 the regression coefficients for different values of λ is presented, illustrating the shrinkage process.

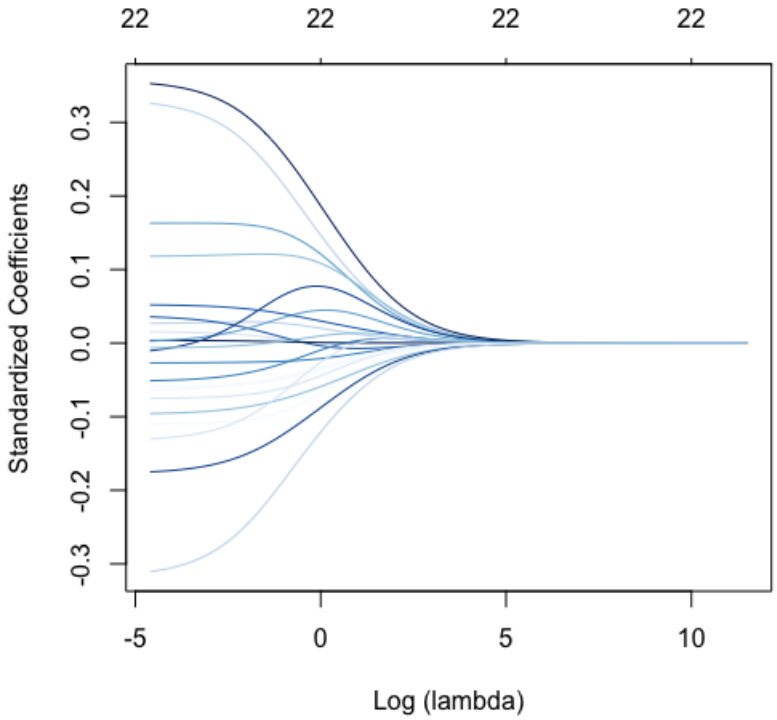

Figure 12: The standardized ridge coefficients as a function of the tuning parameterλ

We further select the λ parameter using a 10-fold cross-validation process on the training set. The model is evaluated on the test data, by using the best tuning parameter $\lambda = 0.0313$, obtained by cross-validation. The MSE obtained for the prediction of Y on the test set is 0.6660. After this, we refit the Ridge regression on the entire data set, using the optimal λ parameter. The estimated regression coefficients for this regression is presented in Table 8.

Estimated coefficients with Ridge	
Variable	Estimate
intercept	6.355710227
genres.action	-0.108315495
genres.adventure	0.013911947
genres.crime	0.028045732
genres.comedy	-0.092292090
genres.drama	0.163024338
genres.fantacy	-0.026491018
genres.family	0.029564644
genres.horror	-0.165628871
genres.mystery	0.003367835
genres.sci.fi	-0.060146864
genres.thriller	-0.073456065
num_critic_for_reviews	0.304621742
duration	0.119448753
director_facebook_likes	0.009680743
actor_3_facebook_likes	-0.047218635
actor_1_facebook_likes	0.050021152
gross	0.008873583
num_voted_users	0.335746631
facenumber_in_poster	-0.058094760
budget	-0.119159502
title_year	-0.284905634
actor_2_facebook_likes	-0.005044970

Table 8: Estimated coefficients when fitting the linear regression model on the training data using ridge regression

From investigating the estimated coefficients in the table, the values are very low. Thus the penalty term in ridge shrink the regression by reducing all coefficients without reducing the number of explanatory variables. In the case where you have a lot of variables that does not necessarily affect the response variable you are likely to get many coefficients close to zero. The MSE obtained for the final model is 0.6667. In Figure 13 the importance of the different predictors, using Ridge regression, is presented.

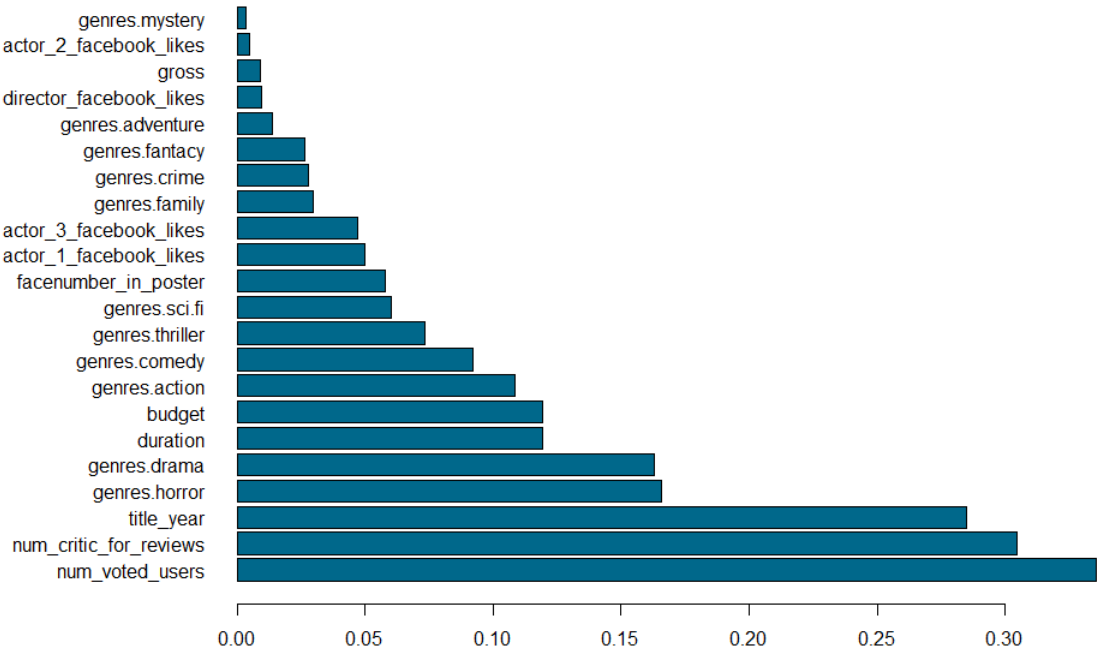

Figure 13: Variable importance plot for the ridge regression model

4.4 Lasso Regression

The Ridge regression have one weakness, compared to the Forward Selection method. While the Forward Selection method generally selects a subset of the predictors, the Ridge regression include all p predictors in the final model. Although the model reduce the coefficients towards zero, none of them will be set exactly to zero. The Lasso regression is an alternative to the Ridge regression that overcomes this weakness. In the Lasso regression the regression coefficients are estimated by minimizing

$$\sum_{i \in n}(y_i - \beta_0 - \sum_{j \in p}\beta_j x_{ij})^2 + \lambda \sum_{j \in p}|\beta_j| = RSS + \lambda \sum_{j \in p}|\beta_j| \tag{6}$$

The β_j^2 term from the Ridge regression is replaced by $|\beta_j|$ in the penalty term. The effect of this replacement is that some of the estimated coefficients are forced to zero, when the tuning parameter λ is sufficiently large. Thus, the Lasso regression, like Forward Selection, is a variable selection procedure. The advantage of this is that the model obtained from the Lasso regression is easier to interpret when the number of predictors are large. Similarly to the Ridge regression we perform 100 estimates of the regression coefficients using a tuning parameter $10^{-2} \leq \lambda \leq 10^5$. In Figure 14 the regression coefficients for different values of λ is presented, illustrating the shrinkage process.

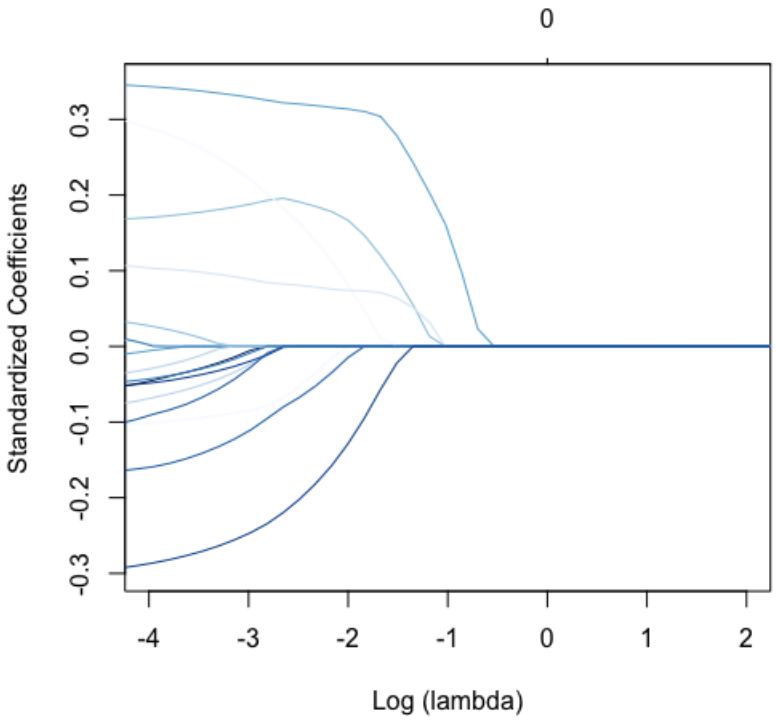

Figure 14: The standardized lasso coefficients as a function of the tuning parameterλ

We further select the λ parameter using a 10-fold cross-validation process on the training set. The model is evaluated on the test data, by using the best tuning parameter $\lambda = 0.0118$, obtained by cross-validation. The MSE obtained for the prediction of Y on the test set is 0.6689. After this, we refit the Lasso regression on the entire data set, using the optimal λ parameter. The estimated regression coefficients for this regression is presented in Table 9.

Estimated coefficients with Lasso	
Variable	Estimate
intercept	6.35571023
genres.action	-0.10367986
genres.adventure	0.00000000
genres.crime	0.01142408
genres.comedy	-0.07927441
genres.drama	0.16730861
genres.fantacy	-0.01287647
genres.family	0.01524058
genres.horror	-0.16651218
genres.mystery	0.00000000
genres.sci.fi	-0.05355725
genres.thriller	-0.05625021
num_critic_for_reviews	0.30405221
duration	0.10859265
director_facebook_likes	0.00000000
actor_3_facebook_likes	-0.03888206
actor_1_facebook_likes	0.03576857
gross	0.00000000
num_voted_users	0.34666399
facenumber_in_poster	-0.04915023
budget	-0.10619359
title_year	-0.29552028
actor_2_facebook_likes	0.00000000

Table 9: Estimated coefficients when fitting the linear regression model on the training data using lasso regression

By comparing the coefficients obtained from the lasso regression with the ones obtain from the ridge regression we can see that fore the lasso regression some of the estimated coefficients are set equal to zero. This implies that these variables are not included in the model obtained from the lasso regression. This proves that the change of the penalty term result in a reduced regression model. The MSE obtained for the final model is 0.6683. This is slightly more than the MSE obtained from ridge. However, the model obtained from lasso have the benefit of being simpler and easier to interpret. In Table 15 the importance of the different predictors, using Lasso, is presented. This plot show similarities to the one plotted for ridge. However here you can see that some of the variables are given zero importance as they are eliminated from the regression.

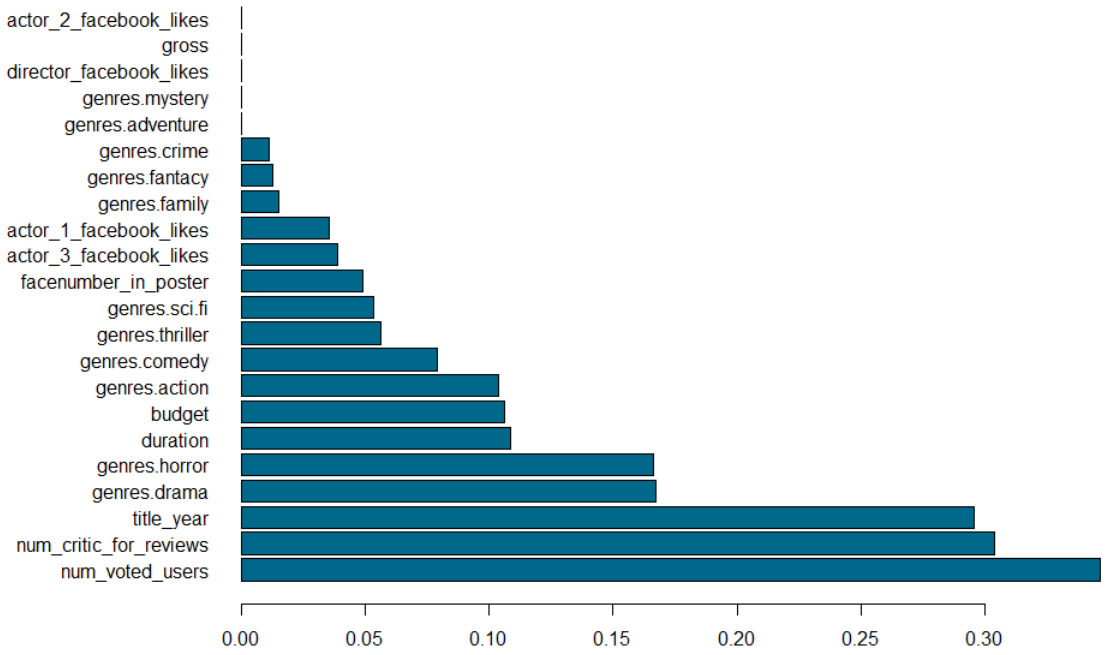

Figure 15: Variable importance plot for the lasso regression model

4.5 Decision Trees

An alternative to the regression methods above, is to construct a decision tree. Here the decision tree is applied to the regression in order to predict the IMDB score. Tree-based methods have the advantage of being simple and useful for interpretation, although they typically are not competitive with the prediction accuracy of other regression methods. Building a regression tree involves segmenting the predictor space into a number of regions. The predictor space is a set of possible values for the predictors $X_1, X_2, \ldots, X_p$. These values are divided into J distinct and non-overlapping regions $R_1, R_2, \ldots, R_J$. All observations in the same region are given the same prediction, which in this gave is the mean IMDb score for all training observation in a given region. The regions are constructed by minimizing the RSS, given by

$$RSS = \sum_{j \in J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{7}$$

where $\hat{y}_{R_j}$ is the mean response for the training observations in region j. Because it is computational infeasible to consider all possible partitions of the predictor space, the regression tree is constructed using a recursive binary splitting. This implies that in every node the partition is branched into two subtrees. The branching process is gready, implying that the current best split is selected in every node. The best split is the split that provides the greatest possible reduction in the RSS. The splitting rules used for segmenting the predictor space is illustrated using a tree structure.

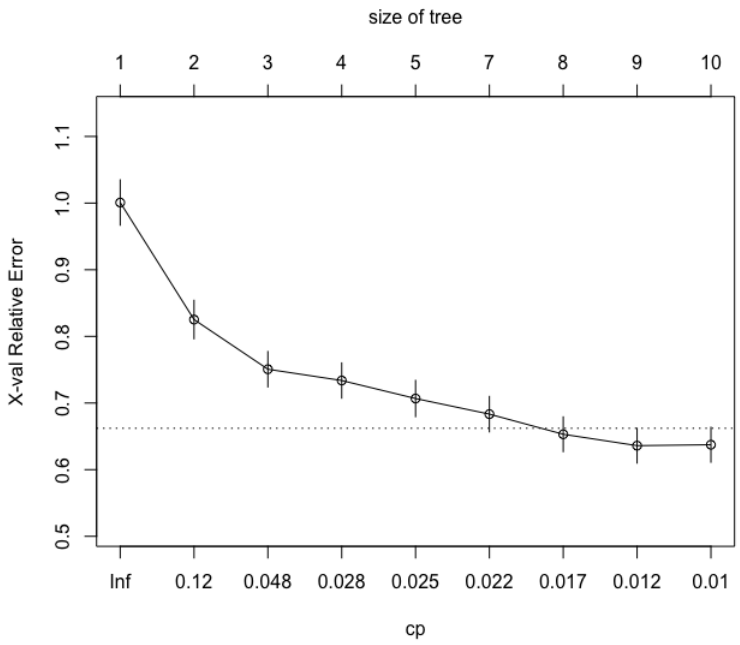

Figure 16: Regression tree analysis showing the cross-validation MSE as a function of the number of terminal nodes in the pruned tree

From Figure 16 we can see the result from cost complexity pruning. The idea behind this approach is to investigate whether the regression tree should be pruned in order to obtain a smaller tree. The pruning process should be performed in such a way that we obtain the lowest possible error rate for the test data. Because considering all possible subtrees is too complex, the cost-complexity pruning method uses cross-validation in order to select a set of subtrees that has the lowest estimated test error. The plot in Figure 16 show that the constructing a tree with 9 and 10 terminal nodes give the same MSE. Therefore we prune the tree to obtain a tree with 9 terminal nodes.

In Figure 17 the tree constructed for the training data set after pruning, is presented. In the top node the splitting rule assigns *num_voted_users* less than 125.000 to the left branch. For this branch the mean IMDB score is 6.11, while the right branch the mean IMDB score is 7.25. The left branch consist of 77.6 % of the total number of observations, while the right branch consist of only 22.4 %. This splitting continues, until we have eight regions, represented by the leave nodes in the tree. The tree structure illustrate the importance of the different predictors in determining the IMDB score. The variable *num_voted_users* is considered the most important, followed by *genres.drama*, *title_year* and *actor_2_facebook_likes* in the left branch and *genres.drama* in the right branch. One normally consider the tree structure as an over-simplification of the true relationship between the predictors and the response variable. However, we recognise that there are some similarities in the results with the ones obtained in previous regression methods. For instance is number of voted users considered the most important predictor in all previous regression methods. However, *actor_2_facebook_likes* has a very low importance level in the previous regression models.

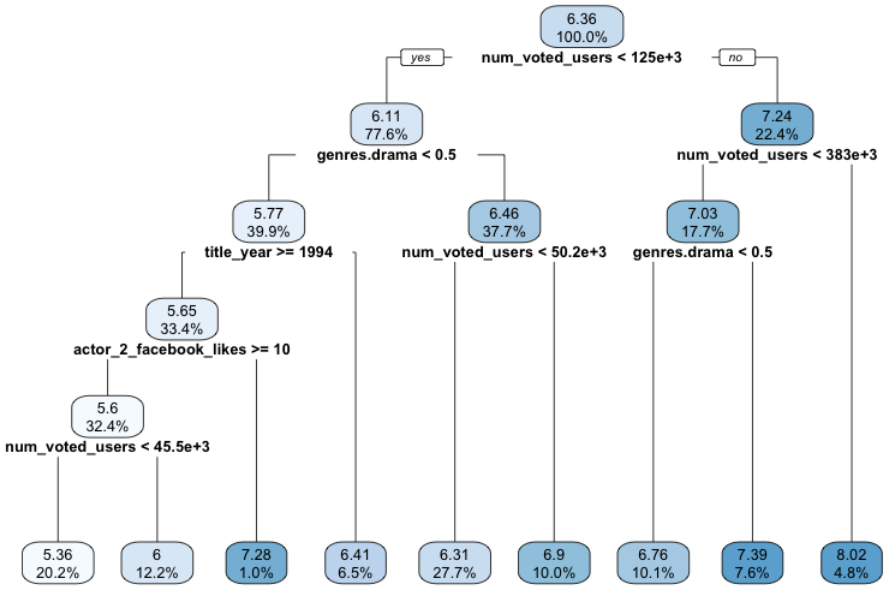

Figure 17: The regression tree for predicting the IMDb score

Once the regression tree is constructed, we predict the IMDB score for the test observations, using the mean of the training observations in the region R_j to which each of the test observations belongs. When predicting the IMDB score using the test data, the MSE associated with the regression tree is 0.8053. By taking the square root of the MSE we obtain 0.8974, which imply that the regression tree model leads to test predictions that are around 0.8974 of the true value of the IMDB score.

4.6 Random Forest

In order to improve the predictive accuracy in regression trees, methods like Random Forest can be applied. Random Forest is a method that use trees as the building block to construct a more powerful prediction model. In this section we apply Random Forest regression to predict the IMDB score. Random forest is an ensemble based learner that add a random feature selection to the decision tree regression model. We perform parameter tuning on the number of trees, ranging between 100 and 1500 trees, with a step length of 100. The computed MSE against the number of trees is illustrated in Figure 18.

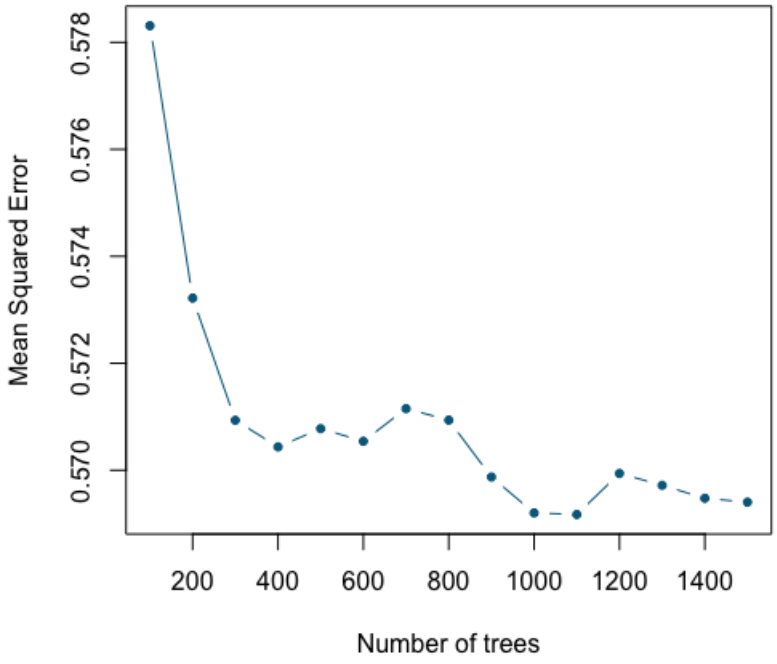

Figure 18: The train MSE as a function of the number of trees

According to this plot, the optimal number of trees are 1100. Using this number of trees we obtain an MSE of 0.5692, which is an improvement from the previous models. Taking the square root of this gives us 0.7545, which imply that the regression tree model leads to test predictions that are around 0.7545 of the true value of the IMDB score. From the Random Forest regression we get that the importance of the explanatory variables are given according to the influence plot in Figure 19.

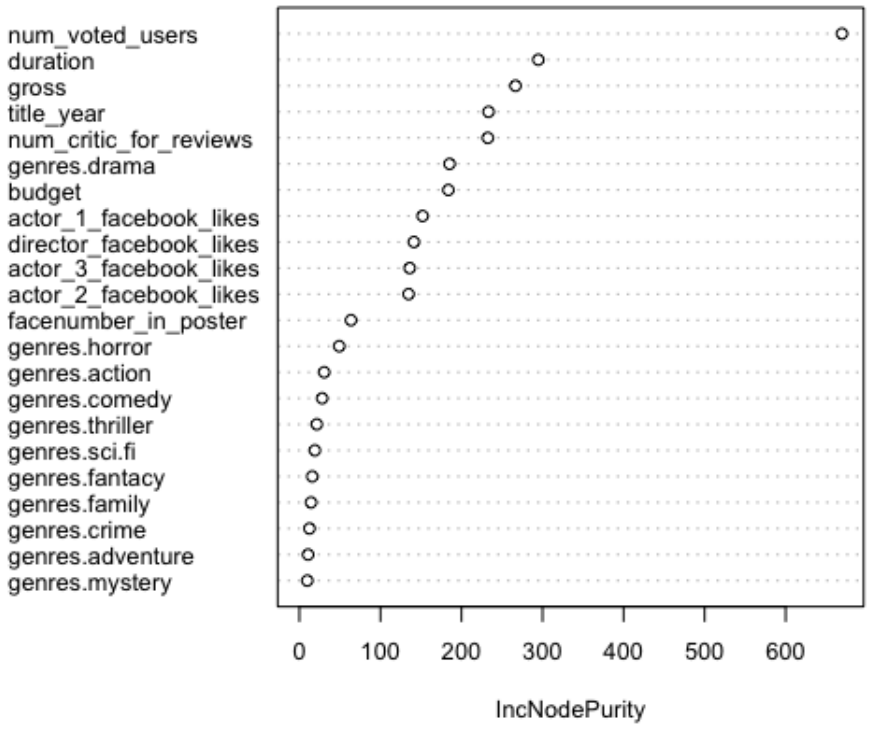

Figure 19: A variable importance plot using random forest with 1500 trees

From this plot we have that *num_voted_users* still have the highest impact on the IMDB score. We further see the influence *duration* and *gross* is higher than for previous models. The dummy variables representing the different genres prove to be of relative low importance, with the exception of of *genres.drama*. The improvement from random forest compared to regression trees is quit strong. Random forest also prove to be the most accurate prediction model in terms of the computed MSE. For investigating the prediction of the *imdb_score* more closely using regression models, one might want to look into whether some transformation would improve the fit of the model. In addition, investigating the regression by including interactions in the dummy variables may give provide more insight.

5 Supervised Classification

From the previous section we had that random forest regression provided the most accurate prediction for the IMDb score. From the variable influence plot, we had that the variable *genres* in general had minimal impact on the response variable. We therefore will continue by only considering the quantitative variables.

In this section we proceed by considering two methods for supervised classification of the IMDB score. Initially the response variable predicted in this study is quantitative. However, more broadly speaking you are in many cases interested in predicting whether a movie is good or bad, based on the characteristics of the movie. Then the exact score becomes of less importance. We therefore wish to study the prediction problem, using supervised classification methods, in order to see if we can gain more information by considering a qualitative response variable. The IMDB score is for this purposed divided into three classes: *good*, *average* and *bad*, according to the following criteria

$$y_i > 7.0 \quad \Rightarrow \quad y_i = \text{good} \tag{8}$$

$$6.0 \leq y_i \leq 7.0 \quad \Rightarrow \quad y_i = \text{average} \tag{9}$$

$$y_i < 6.0 \quad \Rightarrow \quad y_i = \text{bad} \tag{10}$$

When applied to the data set the number of observations for the classes are close to uniformly distributed. The proportions obtained is presented in Table 10. We proceed by constructing a training and set data set, where the training data is a random sample selection of the data. The training data consist of 70 % of the data, while the test set is the remaining 30 %. When investigating the distribution of the different response categories in the two sets, we get that the proportions are approximately the same as in the proportions from the entire data set. The proportions are given in Table 10.

Class	N	Total (%)	Train (%)	Test (%)
Good	1084	30.80 %	30.48 %	31.53 %
Average	1224	34.77 %	35.31 %	33.52 %
Bad	1212	34.43 %	34.21 %	34.94 %

Table 10: Distribution of the response categories in the total data set, the training and test data

5.1 K-Nearest Neighbors

K-nearest neighbors is a classification method that attempts to estimate the conditional distribution of a qualitative response variable given the predictors of a set of test observations. The test observations are then classified according to the probability that each observation belongs to a certain class. Thus the KNN classifier, given a positive integer K, predicts the class of a given test observation by identifying the K observations in the training data that are nearest to it. Then it estimates the conditional probability for each class as the fraction of the points in the neighborhood belonging to that specific class. This can be expressed by the following equation.

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j) \tag{11}$$

where $\mathcal{N}_0$ represent the K-nearest neighbors, K is the number of neighbors to investigate, x_0 a given test observation and j is the response class (James et al., 2014). Finally KNN use Bayes rule to classify the test observations to the class with the highest probability. Because KNN calculates the euclidean distances between all observations, all variables should be scaled. If not, the variables with a higher scale will have a much higher effect on the KNN classifier than the ones with lower a scale. The choice of K is important for the KNN classifier obtained (James et al., 2014). We therefore use cross-validation with K ranging from 1 to 50, to select the best K. The results from cross-validation is presented in Figure 20.

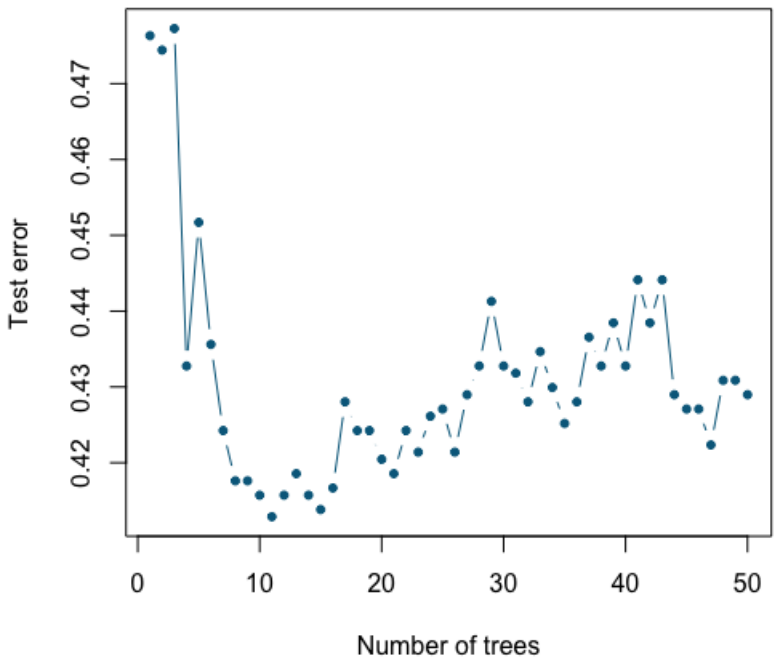

Figure 20: A plot of the test error against K from cross-validation in KNN

In the plot the test error is plotted against the number of K neighbors. From the plot we have that $K = 11$ minimize the test error. Thus $K = 11$ is selected to classify the test observations.

	Predicted		
Actual	Good	Average	Bad
Good	192	100	41
Average	66	161	127
Bad	23	79	267

Table 11: Confusion matrix from KNN

The result from KNN is presented in the confusion matrix in Table 11, where the diagonal elements indicate correct predictions while the off-diagonal elements represent incorrect predictions. From the table we see that quit a lot of the good movies are classified as average and a large part of the average movies are classified as good. The classification error for bad movies is not that sever. The test error is calculating the mean probability of incorrect classifications in the test data, which is the cases where the predicted response does not equal the actual response. The test error from classification using KNN is 0.4129.

5.2 Logistic Regression

Logistic regression is a classification where rather than modeling the response variable Y directly, it models the probability of Y belonging to a certain category. Logistic regression require that the output always is in the range between 0 and 1, for all predictor values. In this case we are considering a multiple logistic regression with three response classes, if a movie is *good*, *average* or *bad*. The probabilities are represented using the log-linear representation (O'Brien, 2016), given by

$$Pr(Y = k|X = x) = \frac{e^{\beta_{0k}+\beta_k^T x}}{\sum_{\ell \in K} e^{\beta_{0l}+\beta_\ell^T x}} \tag{12}$$

where $X_1, \ldots, X_p$ are the p predictors. By performing some manipulation on this equation, we get

$$log\left(\frac{p(X_1, \ldots, X_p)}{1 - p(X_1, \ldots, X_p)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \tag{13}$$

Using the logistic function it directly models the conditional distribution of the response Y belonging to a class k given the predictors X, $Pr(Y = k|X = x)$ (Friedman et al., 2001). In the case of multiclass logistic regression, multinomial likelihood is used to estimate the regression coefficients, where the log-likelihood for N observations is given by

$$\ell(\theta) = \sum_{i \in N} log p_{g_i}(x_i; \theta) \tag{14}$$

where $p_k(x_i; \theta) = Pr(Y = k|X = x_i; \theta)$. In order to construct our classification model, we choose *average* as our reference group for the outcome. The results from classification on the test data can be summaries in a confusion matrix presented in Table 12, where the diagonal elements indicate correct predictions while the off-diagonal elements represent incorrect predictions.

	Predicted		
Actual	Good	Average	Bad
Good	202	100	41
Average	64	171	119
Bad	9	98	262

Table 12: Confusion matrix from KNN

From the table we can see that there are quite a few incorrect classifications. The good thing the number of movies classified as good, that were actually bad and vice versa is not so large. When we calculate the mean probability of making the right classification for each class we get 30.15 % for *good*, 35.93 % for *average* and 33.92 % for *bad*. In the plots in Figure 21 the probabilities of correct and incorrect classifications for each class is plotted, where navy points represent correct classifications and the cyan points represent incorrect classifications.

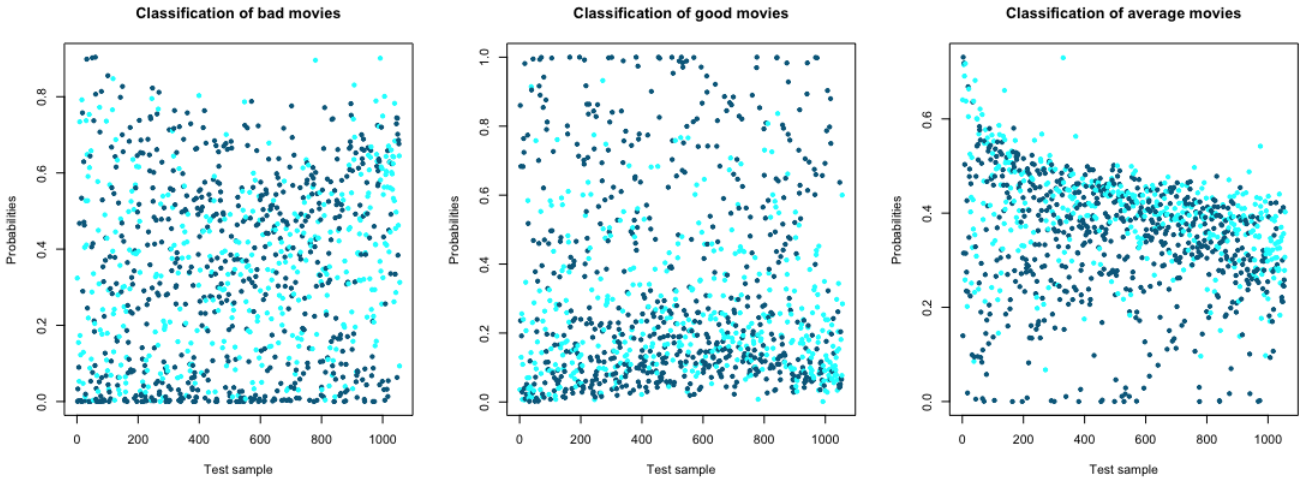

Figure 21: A plot of the probabilities for classifications in each class were blue points represent correct classifications

The plots support the results given in the confusion matrix. The number of bad classifications are quit high. In this plot the incorrect classifications are not separated between the two remaining classes. In this sense the confusion matrix may provide more information. When classifying the rating of a movie, it matters whether a good movie is classified incorrectly as average or bad. The latter should be given a higher penalty than the former, as it is considered more incorrect. To improve the logistic regression, one might consider performing some stepwise regression for reducing the number of predictors used, as the forward selection process described in Section 4. In addition methods for penalizing the incorrect classification according to their severity should also be considered.

6 Unsupervised Classification

The goal of unsupervised classification is to try to extract underlying patterns in the data. For this purpose, the K-means algorithm is used in order to detect groups of movies that have a similar behaviour. The K-means algorithm provides a method for finding clusters and cluster centers in a set of unlabeled data. One chooses the desired number of cluster centers, K, and the K-means procedure iteratively moves the centers to minimize the total within cluster variance. Given an initial set of centers, the K- means algorithm alternates the following two steps:

- for each center we identify the subset of training points (its cluster) that is closer to it than any other center
- the means of each feature for the data points in each cluster are computed, and this mean vector becomes the new center for that cluster

These two steps are iterated until convergence. Typically the initial centers are K randomly chosen observations from the training data. For this study, K-means clustering has been performed on the numerical variables, selecting $K = 2$ to discover two underlying groups of movies in the data. In addition, since the K-means algorithms works with distances, all the variables have been rescaled to values between 0 and 1.

The algorithm returns two very imbalanced clusters: one containing 813 movies and the other one containing 2800 movies. The results of the clustering have been plotted in a parallel coordinates plot that can be seen in Figure 22, where the dark blue observations belong to cluster 1 and the cyan observations belong to cluster 2.

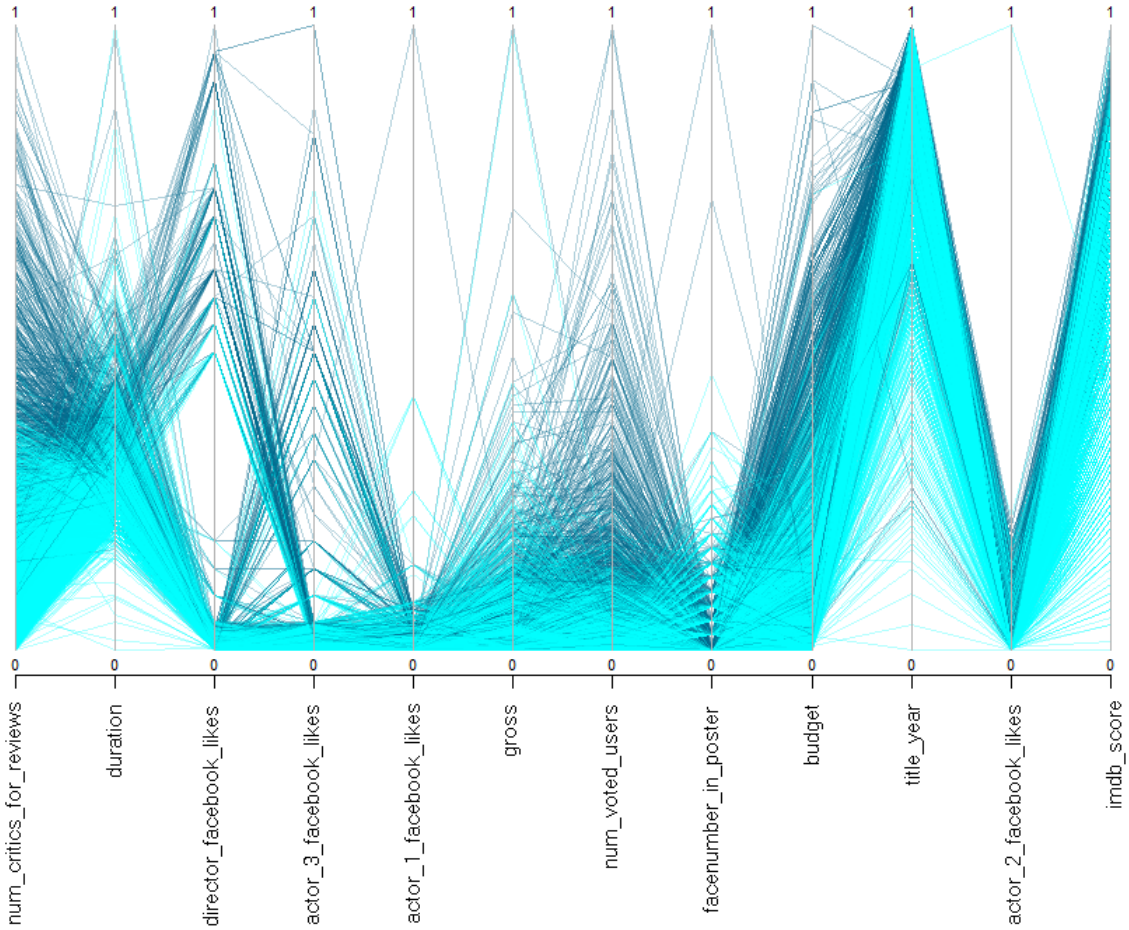

Figure 22: Parallel coordinates plot for K-means clustering with K=2

It can be seen from the plot in Figure 22 that cluster 1 (plotted in dark blue) contains, roughly, movies with a high number of critics' reviews and a high number of Facebook likes for the director, along with a high number of voted users and a high *imdb_score*. Thus, the dark blue cluster can be interpreted as the group of good movies, whereas the cyan cluster contains then the average or bad movies. This gives an intuition of which variables determine whether a movie has a good rating score on IMDB or not.

Next, the centroids of every cluster are plotted so that the general tendency of the clusters can be seen, since in the clustered data plot the high number of observations makes it difficult to see the trends in the data. The parallel coordinates plot representing the centroids of both clusters can be seen in Figure **??**.

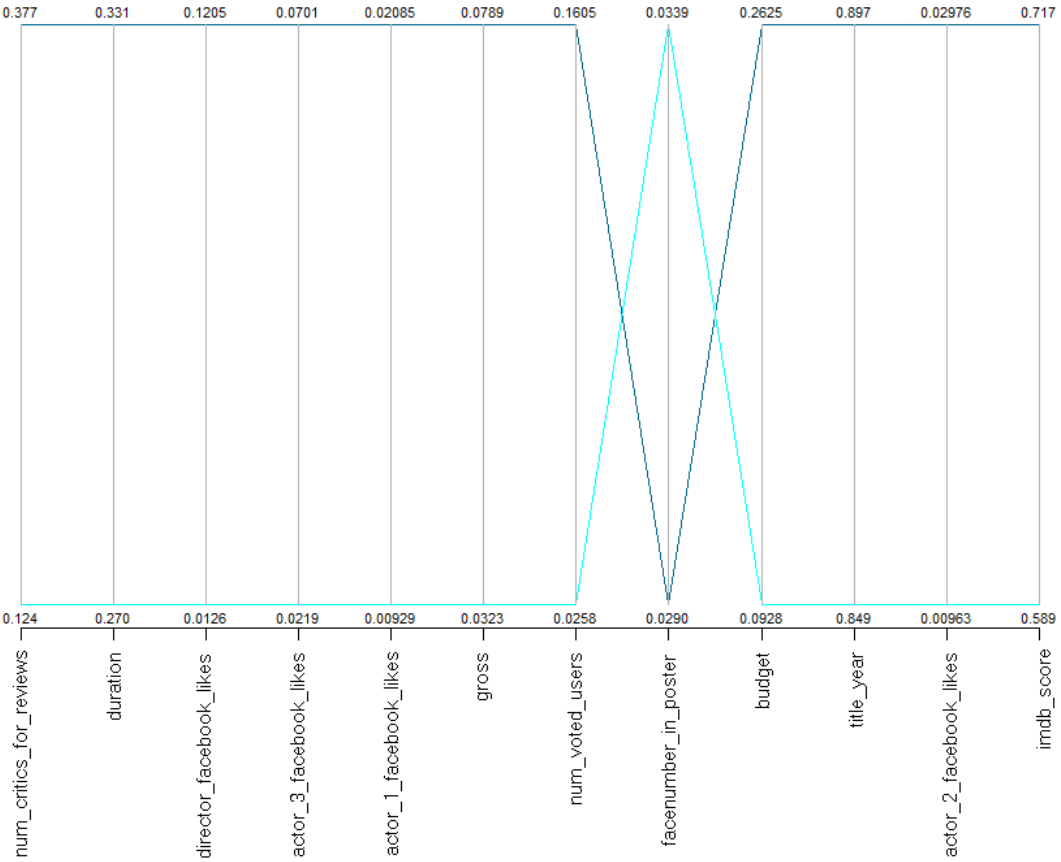

Figure 23: Parallel coordinates plot for the centroids of the two clusters

The plot in Figure 23 shows the two centroids of the clusters, with cluster 1 (represented in dark blue) having higher values for all variables, except for the variable *facenumber_in_poster*. This reinforces the belief of the dark blue cluster containing being a group of good movies, and the cyan cluster containing the *not-so-good* or bad movies.

7 Conclusions and Future Work

There is a wide range of variables that affect the users' rating of a movie. This study has focused on trying to establish which are the key aspects that determine how good a movie is, but the truth is that the reasons why users give high ratings to movies on IMDB are not easy to be determined.

The number of users that have left a rating for a movie, *num_voted_users*, has been proven to be the variable with the highest predictor importance for most of the prediction algorithms. Besides, the categorical variable *num_voted_users* (in its one-hot-encoder version) seemed not to have any relevance for the prediction of the *IMDB_score*.

The Random Forest classifier was the prediction algorithm with the best accuracy, with a Mean Squared Error (MSE) of 0.5692. In this algorithm, the duration of the movie and the gross seemed to be the most important factors in terms of the prediction strength. This leads to the conclusion that the longer and higher-grossing the movie, the better it is. On the other hand, the year of the movie release and the number of reviews from critics did not seem to have a significant influence on the rating that users give to a movie.

In general, it can be said that the information in the dataset was not completely correct or inconsistency-free, and therefore a thorough data cleansing process had to be carried out in order to be able to perform an accurate analysis of the data.

As future lines of work on the topic, several ideas are proposed. The analysis could be extended to movies from every country. For this purpose, the currency details of the textitgross and textitbudget variables should be taken into account, dealing with the currency conversions and with the inflation rates across the years. The information could be completed and crossed-referenced with data from other sources, to create a consistent and bigger dataset for analysis.

The prediction of IMDB rating could be extended by investigating if any transformation, such as the Box-Cox regression or other improve the fit of the model. In this study only a linear regression have been investigated, without including any interaction in the dummy variables. From this study the dummy variables for movie genres showed limited impact on the IMDB score. However by including interaction term, one might obtain a different result.

The movie classification did not yield a particularly high accuracy in predicting whether a movie is good, average or bad. The logistic regression method should be investigated further, by including a stepwise variable selection process. This might yield a higher classification accuracy.

References

Chatterjee, S. and Hadi, A. S. (2015). *Regression analysis by example*, John Wiley & Sons.

Friedman, J., Hastie, T. and Tibshirani, R. (2001). *The elements of statistical learning*, Vol. 1, Springer series in statistics Springer, Berlin.

IMDB (1990-2017). Imdb.com, inc. `http://www.imdb.com/`.

James, G., Witten, D. and Hastie, T. (2014). *An Introduction to Statistical Learning: With Applications in R.*

O'Brien, C. M. (2016). *Statistical Learning with Sparsity: The Lasso and Generalizations*, Wiley Online Library.

Sun, C. (2016). Imdb 5000 movie dataset. `https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset`.