Executive summary

This work presents a novel application of sentiment analysis of user reviews on the well-recognized movie platform IMDb. The purpose of this study is to investigate how natural language, generated from reviews can be used to build models that can create value for businesses across industries.

The study is particularly relevant, due to the emergence of a “connected” world, where everyone has an opinion about everything and companies are trying to transform all this available information into tangible benefits. Furthermore, this paper incorporates an end-to-end classification model, including different strategies on how the language model can be improved with the use of bi-grams, tf-idf and singular value decomposition.

To conduct such an analysis, 24,500 movie related reviews were scraped off the IMDb platform. In order to get the most current and relevant data foundation, the reviews are only related to movies released in the time period from April 2019 to April 2020.

Based on the data foundation the study precedes to dive deeper into traditional data mining approaches and how highly unstructured data can be processed, analysed, improved and evaluated. Through the analysis, the best performing model given the context, was concluded to be the logistic regression yielding an accuracy of 80.8% and thereby outperforming lexicons and other common approaches. The improvement strategies implemented resulted in a 7.5%-point increased performance, compared to the baseline model.

By exploring the sentiment of the reviews, it was found that words related to the actors, cast and directors more often had a negative sentiment, while words related to scenes, action and love had a more positive sentiment.

Limitations of the study, business applications and ethical considerations are reflected upon, and lastly recommendations for further enhancement of our models are proposed. Including how the model could achieve higher generalizability on reviews across different platforms.

Table of Contents

1. Introduction

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

In an evolving digital world as of today, the emergence of vast amounts of data has empowered a new era of data analytics. With the rise of social media and IoT, the internet is home to enormous amounts of unstructured data. This mass of unstructured data has dictated the adaption of new ways of engaging with data and customers across industries. Companies are now more than ever, using analytics as a way of competing and enhancing their competitive edge. However unstructured data has historically been difficult to handle. But in recent years researchers have been making major advancements, which has contributed to the adoption of more sophisticated approaches for businesses to analyse and use unstructured data to improve decision making. One of the areas where advancements have been made is within the field of sentiment analysis.

Sentiment analysis is an umbrella term, covering data mining of emotions and opinions. The concept of opinion mining usually revolves around a combination of natural language processing and machine learning algorithms. Combined, they process unstructured data and classify the opinion of the writer towards a subject of interest. More specifically, we will use opinion mining to classify the sentiment of movie reviews from the IMDb platform.

The internet plays a crucial role in the information stream the customer uses when considering buying new products or engaging in relationships with organizations. Therefore, opinion mining is a very useful tool for organizations to evaluate and improve their products/services based on the customer's feedback. This could for instance be through social media platforms, blog posts, word of mouth and reviews.

One facet of analysing the engagement between consumers and organisations revolves around applying concepts from traditional data mining approaches to unstructured data, more specifically classification algorithms. To develop and train such an algorithm, it requires applications of structure to be applied to the data. This aspect, is referred to as natural language processing and is a big part of this study's emphasis. Furthermore, in order to train such a model using traditional data mining approaches, the data needs to be labelled. With the large amount of user labelled reviews available, IMDb is an obvious choice for inclusion given the scope of this project.

IMDb, short for Internet Movie Database, is an online platform developed in the 90´s, containing information about movies and TV-shows. For each movie and TV-show, IMDb have information about actors, production team, summaries, ratings, reviews and much more. Every month, IMDb has more than 250 million unique visitors across the web and mobile platforms.

This study aims to produce a well performing classification model, that can classify subjective opinions in movie reviews into more homogeneous groups, such as positive and negative, based on the newest and most relevant data from IMDb. Thus producing a model, which can support companies in the movie industry to better understand unstructured data around them.

1.1 Research questions

With the basis in the above mentioned introduction, a research question has been constructed. Moreover, in order to reach a conclusion it is deemed necessary to have consensus, thus two supporting questions are constructed as well:

How can we develop a model based on IMDb movie reviews from April 2019 to April 2020, that can classify sentiment and what improvement strategies are most effective?

- *What are the characteristic words of a positive/negative review on IMDb?*
- *What model is best at classifying reviews?*

2. Literature review

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

Textual data is becoming the most represented form of data across the internet, and therefore a hot topic for research across the globe (Marr, 2018). A great amount of literature regarding sentiment analysis indicates that a lot of studies have been conducted within the document level of various sentimental classification. We have chosen to highlight some similar studies that we find can help and inspire the development of our study.

Classification of Sentiment Reviews using N-gram Machine Learning Approach.
Tripathy et al.(2016), have introduced sentiment analysis with various n-grams techniques, in the form of unigram, bigram and trigram and a combination hereof. They are then applied in combination with several machine learning algorithms such as Naive Bayes, Maximum Entropy, Stochastic Gradient Descent and Support Vector Machines. With a massive amount of computing power, and a combined usage of both uni- and bigrams, the proposed model yields upwards of 95% accuracy.

Pessimists and optimists: Improving collaborative filtering through sentiment analysis.
García-Cumbreras et al.(2013), studies the application of sentiment analysis in recommender systems. First they describe and investigate the relationship between comments and ratings. Their proposed approach results in a binary classification problem, where the users are divided into two groups: pessimist and optimists, and

the data is split into training and test sets. The data is processed using RapidMiner with the Text Processing and Recommender Systems extensions. At last they use collaborative filtering methods such as KNN and SVM, to the classification task of sentiment of the users. The result was a KNN-80 classifier with an accuracy of 80%.

Sentiment analysis using support vector machines with diverse information sources.

Mullen et al.(2004), introduces several methods of using semantic values to words and phrases, and integrating them as features in their SVM model. This allows for weighting the features with regards to their closeness of the topic. In conjunction with that, they are investigating different annotations of the data. The best yielding result is 86% accuracy.

Predicting IMDB Movie Ratings Using Social Media.

Oghina et al.(2012), has another perspective in predicting sentiment by including external input to their models, such as twitter and YouTube features. By combining surface features and textual analysis, the researchers found that with linear regression these external features could reap a prediction accuracy of 89%. The researchers also found that the predictor with the most signal, was the like/dislike ratio of movie trailers on YouTube.

Our literature review has further enhanced our hypothesis, and we have gained a broader understanding of, what research has already been conducted and how it can contribute to our research. Furthermore, the literature review reveals that this research field is in constant development and has sparked our interest in applying similar techniques on different and more current data.

3. Data

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

3.1 Preliminary Theory *(Natural language processing)*

As previously outlined in the introduction, the research in this paper revolves around sentiment analysis, and applying data mining methods to unstructured data. In this case, the unstructured data is in the form of reviews. The extracted reviews are by nature unlabelled, but fortunately the IMDb reviews are associated with a rating from the reviewer, which labels the data. Our aim is to make the reviews structured or at least semi-structured, by vectorizing and transforming character strings into numeric values. These methods are applied, because working with semi-structured and labelled data often yield better results(Yadollahi et. al. 2017). Furthermore, the classification of polarity has to be determined based on granularity and perspective. The focus in this report will be on document level, and thereby makes every review into an observation. Natural language is at its core very hard to comprehend computationally, and is one of the hardest challenges for computers (in the category with computer vision etc.). Therefore in order for the data mining algorithms to work on our dataset, it is necessary to preprocess the data in such a way that it becomes computational feasible.

"The first step in handling text is to break the stream of characters into words or, more precisely, tokens" (Weiss, Indurkhya, Zhang, & Damerau, 2005, s. 20). In short, tokenization is about using separators to split entire documents into words, sentences or paragraphs. Even this step in the process is hard to comprehend for the computer, since delimiters can have many different meanings in natural language. To help resolve this, researchers have developed libraries and packages which eases this, otherwise very complicated task. The next possible preprocessing step towards interpretable input for the models is called lemmatization and stemming. These methods revolve around reducing the number of different tokens, and thereby increasing the frequency in the data frame. Both stemming and lemmatization are text normalisation methods, and they transform inflected words into its root form. More precisely, Stemming a word or sentence can result in words that are not actual words. The stems are obtained by removing suffixes and prefixes in words such as "s", "mis", "ed", "ize", "de".

Once somewhat normalization has been reached through the above mentioned procedures, the document-term matrix(DTM) becomes applicable as a vector generation technique. The DTM is a simple, yet widely used method for applying machine learning algorithms on textual data. Moreover it is basically a representation of a word count of individual terms spread across all documents(reviews). This approach is also called a "bag of words" approach implying that all structure of the sentences are discarded, and the model is only concerned with whether words are represented in the document and not where in the document. As insinuated, this approach is rather simple and thus an adequate stepping stone to further improving model performance, by adding word weighting through a term frequency–inverse document frequency(tf-idf).

A tf-idf is a statistical representation of how "important" a given word is among documents in a corpus. This "importance" weighting is calculated by multiplying the term frequency in a given review, by the inverse document frequency across the entire corpus and can be annotated as: $tf - idf(j) = tf(j) * idf(j)$ where $\mathrm{idf(j)} = \log(\frac{\mathrm{N}}{\mathrm{df(j)}})$ (Weiss, Indurkhya, Zhang, & Damerau, 2005, s. 30). The tf-idf approach has been found to be incorporated in 83% of all textual recommender systems, and has proved to improve prediction power (Breitinger et al.2015). As a final extension to the NLP theory applied in this paper, an addition to the tf-idf approach, would be to incorporate a n-gram model to potentially elevate the prediction power even further. As insinuated, textual data has a high degree of lexical variation which can be encapsulated by introducing a model that takes the sequence and composition of sentences into account. This is where a n-gram model comes in handy, instead of looking at individual words, this approach introduces representations of sequences of words.

N-grams can potentially boost the detection of the true sentiment by capturing a combination of words like "not bad". The 'n' represents the number of words in the sequence being considered, going from unigram, bigram, trigram ...n-gram. Tripathy et al.(2016) proposed utilization of both unigrams, bigrams and trigrams and found that incorporation of a bi-gram model, improved their models prediction power significantly.

3.2 Data collection

The data collection process emphasizes manually scraping movie reviews from the IMDb platform. More specifically, the data extraction is done in R with the Rvest package, in combination with the Google Chrome extension "Selector Gadget". The Selector Gadget is a CSS selector, that eases the scraping process for non-HTML experts. Through our research and literature review, we found that many prior projects have conducted their analysis on either the official IMDb dataset or the well-recognized Stanford dataset(Stanford University, 2018).

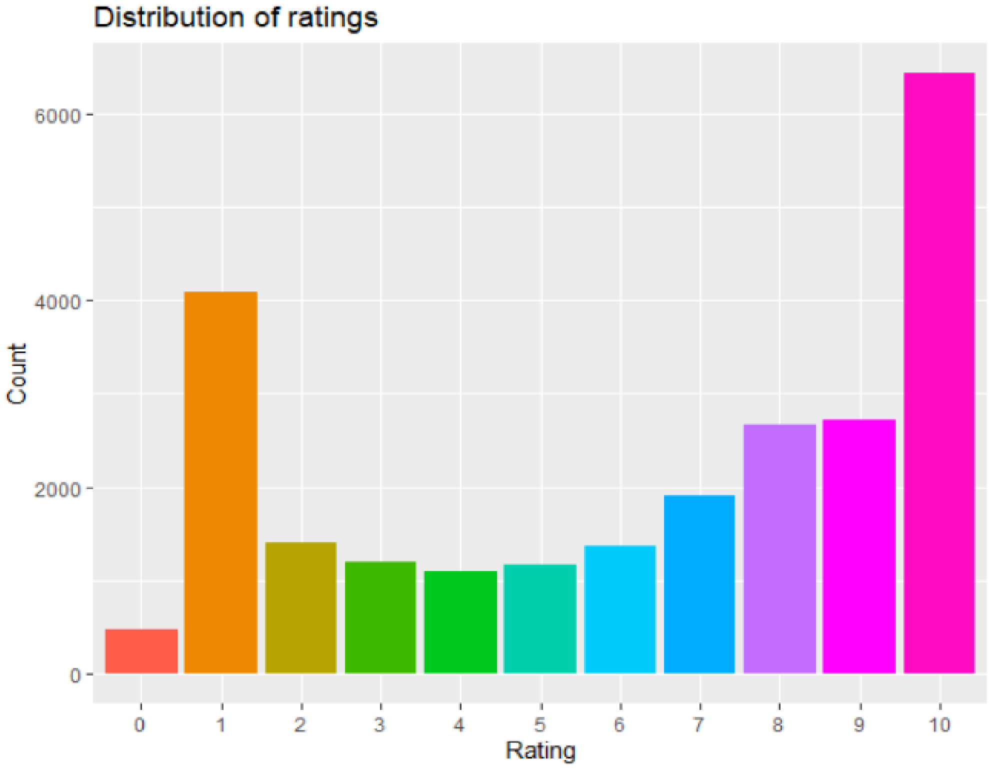
3.2.1 The crawling and scraping process

The first step in the data collection process was to understand the IMDb webpage setup. Luckily, the platform is extremely consistent in its interactions as well as layout. Also the website is without any bot detection or timeout mechanism which spared us a lot of trouble. The platform allows the users to filter, categorise and customise the view of movie information as desired. This has certainly made the process of crawling and scraping easier, since this saved us from having to filter the reviews by date, post scraping. After filtering the movies on the website, we only had movies published from 01-04-19 to 01-04-20, with a simple loop we extracted the movie IDs. Each movie has its own page with the exact same attributes which can be accessed via the movie ID. The main issue we experienced was that not all movies have the same amount of reviews. Some movies had 0 reviews while others had 100's, this meant that we had to incorporate a statement that would check for 0 reviews and maximum include 25 reviews per movie. With simple URL manipulation, we were able to sort reviews so that we would get the reviews with the most "helpfulness", which is a IMDb "like" functionality that allows users to show their agreement with other reviewers. The IMDb platform was very accessible for scraping, and with the help of the GadgetSelector we were able to extract our corpus.

3.3 Data description

The raw data of this project, consists of movie information and reviews from all movies published in the last year (1. April 2019 - 1. April 2020). The proportions of our corpus consists of 24500 reviews including movie ID, review title, review body and rating associated with the review. The movie ID and review title are references to both the movie title and review title, these variables are not considered for any further analysis throughout the study. The relevant variables are the review body and the rating. The review body represents the written review tied to the rating score. The review body varies substantially in length, some reviews are simply one word, while others are full pages of condensed text. Users are able to structure their reviews however they like, this makes the different reviews follow very different formats. Some reviews are in a list form with pros and cons, while others follow a more essay-like format. These written reviews are labelled by the users on a scale from 1-10, called rating.

As shown in figure 1the largest representation of reviews are either 1 or 10. Furthermore, we find some observations of 0, which is due to users writing a review without accompanying it with a rating.



(figure 1)

Since our data consist of reviews of movies complemented by labelled ratings, we have to define cut-offs for each respective sentiment. This definition of a positive and negative sentiment is as follows:

- Rating ≤ 5 equals a negative sentiment
- Rating ≥ 6 equals a positive sentiment

This definition of the target variable is our subjective denotation of the scale, with an equal split of a rating from 1-5 being negative(0) and 6-10 being positive(1). However, since the rating scale has no clear definition, one could argue that ratings of e.g. 4-6 can be ambiguous, thus in future iterations the sentiment cut-offs could be revised.

3.4 Data preparation

The first step in the preparation procedure is removing the reviews with no rating(0). This filtering reduces our corpus from 24500 to 24024 documents. Then we quantify our target variable, with a simple ifelse + mutate statement that creates a column with a binary variable. 1 for a positive sentiment, and 0 for a negative sentiment. This results in a sentiment distribution of 15084 positive reviews and 8940 negative reviews. To equalize the sentiments and ease the computation process, we have under-sampled our majority class (positive), to balance the reviews and give the algorithms equal amounts of data points, to train and capture the structure of positive and negative reviews.

The corpus ready for preprocessing and analysis includes:

- 16000 rows consisting of the columns review_body(text) and sentiment(0-1) of which 8000 is positive and 8000 negative.

3.5 Data pre-processing

The data preprocessing is carried out in two phases as an improvement iteration. This approach derives two corpora, a "baseline" corpus and an "advanced" corpus. The baseline corpus is handled with standard preprocessing methods without feature engineering, while the improvement iteration includes feature engineering such as N-grams, tf-idf and latent semantic analysis(LSA). The preprocessing steps are mainly carried out using the "tm" package, with addition of the "SnowballC" package for stopword removal, "RWeka" for n-gram tokenizing and "irlba" for singular value decomposition.

Baseline corpus	**Advanced corpus**
Input: Data frame with 2 columns (binary sentiment, review) and 16000 rows	**Input:** Data frame with 2 columns (binary sentiment, review) and 16000 rows
1. Conversion to ascii format thus removing unrecognizable characters and stripping HTML tags 2. Tokenization 3. Case conversion(lower) 4. Remove punctuations 5. Remove numbers 6. Strip whitespace 7. Stem 8. Stopword removal 9. Create DTM (bag of words) 10. Sparseness reduction (0,99)	1. Conversion to ascii format thus removing unrecognizable characters and stripping HTML tags 2. Tokenization 3. Case conversion(lower) 4. Remove punctuations 5. Remove numbers 6. Strip whitespace 7. Stem 8. Stopword removal except **“no”, “not and “nor”** 9. Create DTM (bag of words) 10. Sparseness reduction (0,99) 11. Define n-gram 12. Create a tf-idf with N-gram weighting 13. Apply SVD algorithm
Output: Document term matrix of 794 columns and 16000 rows	**Output:** tf-idf of 51 columns and 16000 rows (blackbox)

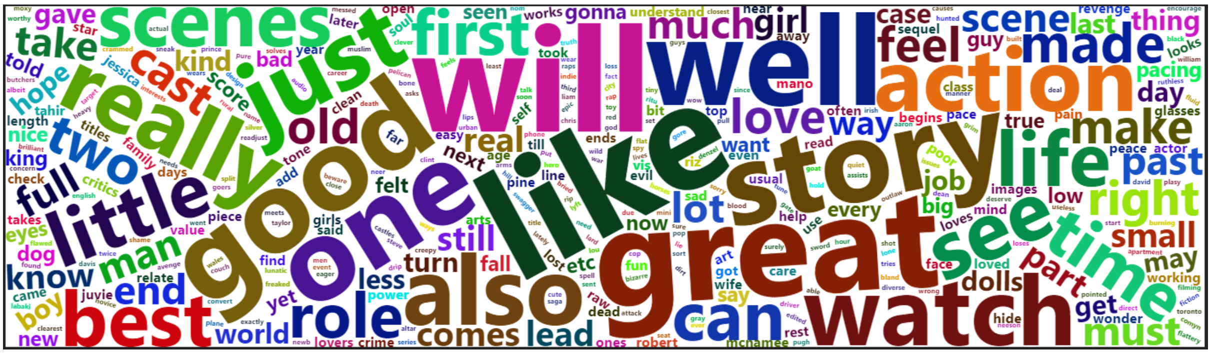
(table 1)

To elaborate further on the output of the “advanced corpus”, we ended up with a bi-gram and trimming the corpus down to 50 predictors, using a singular value decomposition estimation. This estimation is based on the LSA approach, also named a blackbox technique. This technique allows for reducing an extremely large matrix into a significantly smaller matrix, while still preserving the similarity structure between columns.

The bi-gram approach allows us to capture additional semantics of the reviews, this includes word combinations as seen in appendix 1. In this table we get an overview of the top 10 most common combinations. This also includes valence shifters such as combinations with “no”, “not” and “nor”. These valence shifters are further represented in the illustration seen in appendix 2, where we for illustrative purposes have used the AFINN lexicon to visualise the most common words used with “not”.

Furthermore, as a data-exploration extension, word-clouds have been assembled to explore the frequency of words, classified as positive and negative. The word-clouds can be seen below in figure 2 for positive and figure 3 for negative. The word-clouds should be interpreted by the size of the font. The bigger the size of the font, the more represented the word is across reviews.

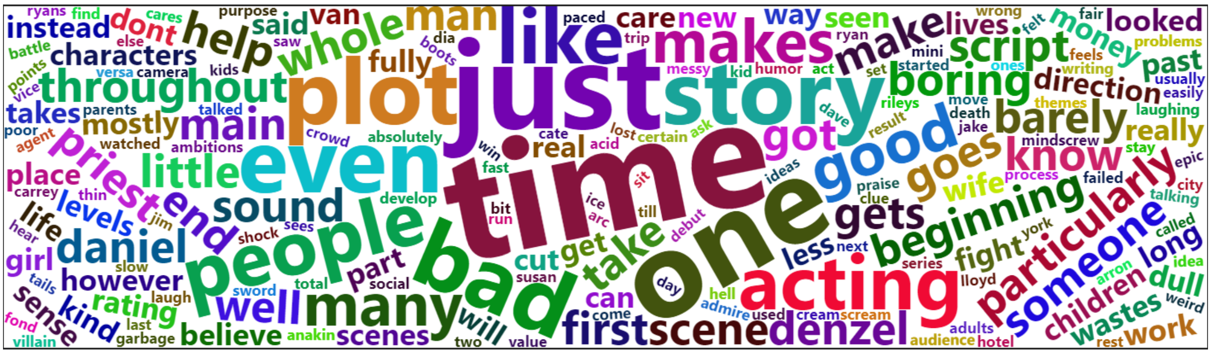
Positive:



(figure 2)

As expected we see that “*good”*, *“like”*, “*great”* and *“best”* are all words that classify as having a positive sentiment. Similarly the same applies for the negative words, though words such as *“acting”*, *“plot”*, *“story”* and *“people”* are weighted with a negative sentiment. Also actor/director names can be found in the negative word cloud. For example we see names like “*Denzel”* and “*Daniel”*, which indicates that Denzel and Daniel has a larger negative influence on experiences rather than positive experiences in our corpus. Another derivation from these visualizations is that most of the represented words are related to either the script(story) or the actors(cast).

Negative:



(figure 3)

4. Applied algorithms

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

4.1 Preliminary theory

When considering machine learning techniques for opinion mining, two types of methods are commonly used for sentiment analysis. These techniques are based on unsupervised and supervised learning. Unsupervised learning is considered when the data is unlabelled and the aim is to discover patterns in the data. On the other hand, supervised learning does require labelled data, and the algorithms are mapping from the input to the output, and are trained to obtain a decent prediction of the target variable. When the preprocessing of the data has been carried out, we become able to quantify our variables, including the target variable and hence apply supervised data mining algorithms to our problem.

Beforehand it should be noted that the Support Vector Machine (SVM) algorithm is anticipated to outperform the other models. This is due to the overall goal of the assignment, is to predict the sentiment of the users based on their reviews. Since SVMs are essentially optimization problems, they will always seek to find a global optimum and thereby have almost always the highest prediction accuracy. Therefore, different models with different characteristics will be included and evaluated in the analysis, in order to verify our preliminary expectation about the performance of our models.

The models included in the analysis:

- *Logistic regression*
- *Random Forest*
- *Support Vector Machines*

4.1.1 Logistic regression

Logistic regression (LR) is a very powerful tool for classification tasks, with binary response variables. LR uses the method of maximum likelihood. The intuition behind using maximum likelihood estimation, is to fit a model that yields a number near one for observations who has a positive sentiment, and a number near zero for observations who have a negative sentiment. The threshold for classification is by default 0.5, meaning that estimated values below 0.5 is classified as a negative sentiment and above is positive. This threshold could be adjusted according to specific goals of predicting, but in our case, 0.5 is sufficient. Furthermore, LR is also very useful for making inference. It is widely used by researchers, and usually yields solid results when considering the principle of parsimony.

4.1.2 Random forest

Random Forest (RF) involves stratifying or segmenting the predictor space into a number of simple regions. The RF approach involves producing multiple decision trees (ensemble of trees), and combining them to yield a single consensus tree. This approach will result in a higher accuracy/flexibility but with the expense of some loss in interpretation of the features. Each tree can consist of a root node, internal nodes and terminal/leaf nodes. The number represented in each leaf, is the mode for categorical data and mean for numeric data, of the observations which fall within that category.

First we build a number of decision trees on bootstrapped training samples. A random tuning parameter of m features is considered as a split candidate out of the full set of p features, and at each split a new sample of m predictors is considered (typically $m \approx \sqrt{p}$ in classification). Compared to bagged trees, RF adds a small tweak of decorrelate the tree, by only considering a subsample of the features, this enhances a quite dramatic reduction in the variance.

4.1.3 Support Vector Machines

SVM will be the most flexible model used in this report, and hence have the lowest interpretability in terms of understanding how the predictors are associated with the response. SVM also implies that they seek maximum generalizability, and are guaranteed to find the global optimum of all possible solutions. This is done by introducing hyper-planes in binary classification tasks, where the hyperplane will separate the observations with the biggest margin possible to avoid overfitting the data. By softening the margins using a budget (Parameter C), and allowing observations to violate the margin and even the decision boundary, it becomes fairly robust to outliers compared to for instance maximal margin classifiers. Furthermore, it’s different from support vector classifiers by extending the feature space, through the use of kernels. They will use the inner product and become flexible enough to capture non-linearity, and accommodate non-linear decision boundaries in the data.

To use the Support Vector Machine algorithm, it requires a vector of numbers as input. Therefore, as earlier mentioned we have vectorized our text and scaled it into numeric values.

4.2 Evaluation criterions

In order to differentiate between the models, multiple evaluation criterions will be performed on the algorithms. Some of the most widely used criterions within supervised machine learning, is based on elements from a contingency table, also known as a confusion matrix. This table can help assess the models in terms of different classifications errors. When talking about the elements in the confusion matrix, it consists of observations classified as “true positive”, “false positive”, “true negative” and “false negative”. True positive classification is when a review is positive and it is classified as positive by the algorithm, and false negative specifies positive reviews but the algorithm does not classify it as positive. Likewise, true negatives reflect negative reviews whereas the algorithm also classifies the review as negative, while false negatives are negative reviews but the algorithm classifies it as positive. Extended from the confusion matrix, it becomes very convenient to examine the sorts of errors the algorithm is producing. The errors produced, could help gain insights into which polarity the algorithms predict well, and if certain classification groups are more difficult to predict than others. Especially precision, recall, f-measure and accuracy are predominant in this area.

- Precision

The precision measure is the ratio of documents that is correctly labelled as positive compared to the total number of positive sentiments classified. Our objective is to maximize this measure meaning that a precision of 1 is desired.

$$Precision\ =\ TP/TP+FP$$

- Recall

The recall measure is the ratio of the truly positive sentiment, compared against the truly positive sentiments and the false negatives, which reflects the positive sentiments classified as negative by the algorithm. In other words, it classifies the number of true positive sentiments against the positive sentiments calculated by the algorithm. As with the precision, the objective here is to achieve a recall close to 1.

$$Recall\ =\ TP/TP+FN$$

- F-measure

F-measure/F1 is reflected in the harmonic mean of precision and recall/sensitivity. The harmonic mean will penalize extreme values. The F-measure is preferred when false negatives and false positives are most important. It measures the accuracy of the model, by balancing the precision and recall, and it is often used when the classes of predicting are imbalanced.

$$F - measure \ = \ 2/((1/precision) + (1/recall))$$

- Accuracy

Accuracy is the measure of all correctly identified cases, and it is usually preferred when the classes are equally important to classify. The null accuracy is the prediction of always classifying the class as the mode, and in this case - 0.5. The null accuracy gives an intuition about how well the algorithm works in accordance with just predicting the most frequent class.

$$Accuracy \ = \ TP + TN/Total$$

5. Analysis

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

As described, our experimental setup is based around classifying polarity by applying a selection of algorithms on different preprocessed corpora. The first experiment in the sentiment analysis that is carried out is the baseline models, without tuning the models. Afterwards we implement the more advanced preprocessed data, and besides that, the algorithms are executed with and without tuning, to really isolate our improvement strategy with regards to our hypothesis. The initial corpus of 16,000 observations of equally distributed sentiment, is divided into a random training and test set (70/30). The algorithms were then trained using the software application R, sequentially on the baseline and then the preprocessed training set, with the binary target variable, regressed on all the other variables. Lastly the prediction of the algorithms was tested on the test sets.

5.1 Initial experiment

The features used in our baseline LR model was generalized linear model (GLM). The GLM relaxes the assumption that response variables should have normal distributed errors. Furthermore, we have chosen the Bernoulli distribution, complemented with the link function “logit” because it’s a binomial outcome. No feature engineering was conducted in order to reveal how much extended preprocessing is contributing to enhanced accuracy, and how much tuning does.

When trying to run the RF algorithm, we did encounter a computational problem on our preprocessing dataset with ~ 800 variables. The problem was discovered very late in our process, but it was still manageable to calculate the score on the extended preprocessed data, due to the SVD algorithm which is compressing the variables. The computational problem is most likely caused by the vast amount of predictors, making the amount of trees to compute, compile to millions.

SVM were carried out with a radial kernel which proved to yield better results than a linear kernel, this indicates that the decision boundary most likely is non-linear. The cost parameter was automatically set to default by R = 1, and the sigma value was also default 1/p = 0.0015.

5.2 Extended preprocessing

The LR model has been slightly transformed besides the extended preprocessing, by having the target variable only regressed on 50 compressed standardized variables. The feature engineering showed no improvement in the models prediction accuracy. As expected the model had none, zero and near zero variance variables, and the model did not improve by normalizing the variables (appendix 3). Otherwise the features remained the same with a GLM method and a Bernoulli distribution with the link function "logit".

In the ensemble of trees "family", RF are shown to have the least variability in their prediction accuracy when tuning, and therefore tuning the algorithm is expected not to improve the model substantially. However, the RF has several tuning parameters, which have been tuned to attain the best possible result. A hyper grid search of 120 different tuning parameter combinations was performed. with the following parameters included: the fraction of features to consider at each split, where 5%, 15%, 25%, 33% and 40% of the original 50 variables were considered, and 8 was optimal. Minimum node size is by default 1 in R for classification tasks, and in the grid search 1, 3, 5 and 10 was included in the search, while 3 as the minimum node size of a tree, performed best. The sampling schemes are by default with replacement, meaning that each bootstrap sample has the same length as the original training data, and FALSE if not. The majority of best performed replacements was "TRUE" which also included the optimal tuning. Lastly, the sample fraction which is a class-specific vector, consisting of a fraction of observation to sample on, included 50%, 65% and 80% and 0.5 performed best in conjunction with the other parameters (appendix 4).

The SVM model is carried out with the same radial kernel as previous, but it is now tuned with a grid search on cost and gamma. The cost parameter was ranging from 1-10 with a 0.5 interval, resulting in C=3.5 as the optimal cost parameter, and gamma was ranging from 0.005 to 0.1 with an optimal gamma of 0.01 (appendix 5).

Initial pre-processing

	Confusion Matrix	Precision	Recall	F-measure	Accuracy
Logistic regression	Observed values **P** **N** **P** 1796 572 **N** 716 1716	**71.49%**	**75.84%**	**73.6%**	**73.17%**
Random Forest	Observed values **P** **N** **P** N/A N/A **N** N/A N/A	**N/A**	**N/A**	**N/A**	**N/A**
Support Vector Machines	Observed values **P** **N** **P** 1816 596 **N** 696 1692	**72.29%**	**75.29%**	**73.75%**	**73.08%**

Extended pre-processing (bigrams, TF-IDF & Tuning)

	Confusion Matrix	Precision	Recall	F-measure	Accuracy
Logistic regression	Observed values **P** **N** **P** 1991 512 **N** 409 1888	**82.95%**	**79.54%**	**81.2%**	**80.81%**
Random Forest	Observed values **P** **N** **P** 1849 551 **N** 590 1810	**75.8%**	**77.04%**	**76.41%**	**76.23%**
Support Vector Machines	Observed values **P** **N** **P** 2006 523 **N** 394 1877	**83.58%**	**79.31%**	**81.38%**	**80.90%**

(table 2)

6. Results

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

The key metrics for evaluating our models and their improvement strategy, are shown in table 2. The performed evaluations are showing indications of the LR and SVM to yield the best performance in terms of accuracy and f-measure, while the LR having an overall accuracy of 80.8% and the SVM with a 80.9% accuracy.

The LR model is performing 73.17% accuracy in predicting sentiment as positive and negative in the baseline model, and the accuracy is improved with 10.4% (7.64%-point) by adding bigrams and TF-IDF. The tuning of the LR model did not show any improvements, most likely because the variables already are normalized, scaled, centred and all variables contain information (variance). All evaluation criterions are improved with precision as the frontrunner, yielding a 16% increase (11.46%-point), while recall has the least improvement with a 4.8% increase (3.7%-point). The improvement of recall and precision enhances the f-measure by 10.32% (7.6%-point) which is almost the same as the best performing model - SVM.

When considering the RF algorithm, it performs poorest on all evaluation parameters and it only yields a 76.23% accuracy, but yet still more accurate than all of the other baseline models. The feature engineering improved the model by 1% (compared to default settings in R), which was somehow as expected, because the variability when tuning RF is proved to be minimal. Furthermore, it is noticeable that the recall metric (77.04%) is almost as accurate as the LR and SVM. This shows that if detecting a high fraction of detected positives compared to actual positive sentiment is desired, then RF could be almost equivalent to the other two models.

Like the LR, comparing the SVM in the two tables, clearly indicates an improvement on all evaluation criterions, based on the extended preprocessing and tuning. The highest increase is a 15.6% (11.29%-point) increase in the precision metric, while the recall only gained 5.3%(4.02%-point), the f-measure achieved a 10.3% (7.63%-point) increase. Another interesting note, is that the SVM algorithm with the extended preprocessing but without tuning parameters, yields an accuracy of 78.83% equivalent to a 7.8% (5.75%-point) increase in accuracy, compared to a 10.7% (7.82%-point) increase if parameter tuning was included. This indicates that the extended preprocessing improves the model with 5.75%-point in terms of accuracy and the tuning itself, only contributes with a 2.07%-point increase in accuracy.

At last, it is also worth noting which kind of errors each algorithm is making. As mentioned earlier RF is almost equivalent to the other models when only considering the recall measure. However, the two best performing models (LR and SVM) are fairly equal in prediction. They both outperforms RF when considering all other metrics and especially the precision, which also indicates the biggest increase after applying extended preprocessing and parameter tuning in the two models. As a final remark, the LR is slightly better at predicting fewer false positives (actually negative) and true negatives - meaning that the model is better at predicting negative sentiment. While the SVM are slightly better at predicting positive sentiment with fewer false negatives (actually positives) and true positives. These classification errors are important to have in mind when choosing the final model.

7. Discussion

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

7.1 Limitations of the study

It is worth noting, that the whole basis of this study is based on a review system that has no clear definitions or guidelines regarding the rating scale. This means that it is up to the user to determine what each of the ratings implicate. *N.S. Koh et al(2010)* studies the issue of bias among "raters" across multiple rating platforms. The study found that the individual perception of a score between 1-10, is highly dependent on aspects such as cultural differences, attitude and social norms. This inconsistency is very visible at a glimpse over the reviews. Some people would give a movie the score of 10, even though they list a number of negatives, while others only have positive things to say and end up with a score of 6. It is also important to take the distribution of ratings into account. As described in the data section, the most represented ratings were 1 and 10. By taking a closer look at some of these ratings, we find that many users simply do not agree with the overall ratings, so they give a radical rating to pull the overall ratings towards their personal verdict. Another aspect worth considering, is that no one knows whether or not a user that has given a review actually watched the movie, and then gave his or her honest opinion. Some people may have incentives to give a specific movie a good or bad review even though they haven't watched the movie. (Hickey, 2016), studied the distributions of ratings among different movie rating websites. One of the findings in the study was, that a movie that wasn't even in the theatres yet, already had over 12.000 reviews, indicating the users are reviewing a movie based on their expectations of the movie and not the movie itself. These findings must be addressed and considered before using the algorithm, since the whole premise of training the models is based on the users reviews and accompanied rating. If the rating and review are contradicting then it would skew the model immensely.

7.2 Further improvements

Since our study is based around users ratings and the assumption that reviews ≤ 5 is considered a negative sentiment and >5 is a positive, we have to state that this is our subjective perception of what determines each sentiment. And this perception most likely differs from some of the reviewers perspective. One way of further improving the model and reducing the ambiguity of mediocre scores skewing the sentiment, is to change the binary separation to e.g. a 1-3 & 8-10 split.

Researchers considered state of the art within NLP yields accuracy upwards of +90%, all utilizing a more sophisticated NLP approach combined with deep learning. Therefore, for us to improve our accuracy further, we could start by expanding our NLP approach, with more advanced self-developed language models, such as a rule based model. Also the incorporation of neural networks could potentially improve our model further. Because this is outside of the scope of our curriculum and project, we instead display a model producing a competitive accuracy using more simplistic preprocessing algorithms, such as bi-grams and SVD. The literature also shows examples of data augmentation used as a preprocessing step, though as the author mentions, we would need 120GB disk space for processing the augmentation (Xie, 2020). In regards to N-grams, the literature review emphasized on studies showing that the accuracy can be improved by combining unigrams, bigrams, and trigrams. We used bigrams in our preprocessing, which took a lot of computational power and run-time. Due to the fact that we do not have access to virtual machines and servers with high computational power, we were not able to incorporate a more complex NLP approach. Language can be complex as sentiments are found in single words, but also the context they appear in. Therefore we are aware that it most likely would have improved our model significantly with a combination of unigrams, bigrams and trigrams.

7.3 Scope expansion

Aligned with the scope of this study, we can only conclude the performance of the models based on the reviews from the IMDb platform. An interesting extension of the scope could be to use the trained models to predict the sentiment of other movie reviews regardless of the platform. This could for instance be platforms like Twitter, Rotten tomatoes etc. Another way to expand the scope of classifying movie reviews further, could be to include classification of reviews from other industries than just movies. For this extension, it would make sense to be able to classify company reviews. In that case, it would be necessary to use data from sources such as Trustpilot, Reddit, Twitter or Facebook in combination with our IMDb data to train the model, which potentially could improve the generalizability across different review types and not just movie reviews. To enable generalizability further it could be relevant to include the use of lexicons such as AFINN. The inclusion of lexicons would allow for a broader extension of the corpus. A lexicon such as AFINN also provides insightful functions such as 7 categories of sentiment. These classes of sentiment are also highly relevant in the context of movie reviews.

7.4 Ethical considerations

We have made sure that it is legally responsible to use crawling and scraping methods on the IMDb website. The terms and conditions stated on the website, outlines that crawling and scraping IMDb data is allowed for personal and academic use, but not commercial use.

“Please note that this data set is strictly for academic use that "stays in the classroom" (IMDb, 2020).

The study has been carried out in accordance with the general GDPR protection principles. Even though our data initially did not consist of any personal information, we still need a consent form IMDb because it is possible to extract personal information on users through their website, and in that case you are obligated to follow the GDPR.

With the consent from IMDb, we have approval to extract the reviews and use it for academic purposes. Because the data did not consist of any personal data, no anonymization or pseudonymization has been carried out. This implies that this report would not be obligated under the GDPR regulation.

8. Conclusion

Simon Stausholm Rasmussen 201902009, Simon Lyngsø 201608547, Nicolai Rosling Andersen 201610867 & Simon Issing Thiesen 201901998.

How can we develop a model based on IMDb movie reviews from April 2019 to April 2020, that can classify sentiment and what improvement strategies are most effective?

In this report, we try to develop a model which is able to classify sentiment using various supervised machine learning algorithms combined with various NLP and parameter tuning techniques. Initially, we have gathered the most current data available, by manually using web crawling and scraping on IMDb. Afterwards, in the exploration phase, numerous NLP steps were carried out in order to evaluate what could improve our model and make it competitive, in a well-researched area of data/computer science. We successfully used a simple and a more complex NLP procedure, to isolate our improvement strategy and the effects. This naturally directs us to the final part of the process; training and tuning. During the final process, we learned about the computational relevance of choosing different data mining models, and we had to stop running a random forest due to the computational requirements. However, all other models seemed to perform at a decent error rate. At last, the model evaluation criterions was calculated and compared and an assessment of the different improvement strategies was performed. That leads into the final part of the question; *"which model improvement strategy was most effective?".* If we look at the outcome of the analysis, we find indications that the advanced NLP improvement strategy produced a remarkable increase in model performance. The NLP approach yielded upwards of 6-7%-point accuracy improvements, while the tuning of the algorithms only yielded around 1-2%-point increase. These increases clearly show that advanced NLP techniques in general seem to be more effective on prediction accuracy when considering sentiment analysis, compared to parameter tuning. But the techniques are definitely not mutually exclusive and the final suggestion must be to use the two methods in combination to obtain the best possible accuracy.

What are the characteristic words of a positive/negative review on IMDb?

The positive and negative word clouds of the reviews revealed no unusual insights and was overall as expected. The most generic words used were words such as *“good”*, *“like”*, *“bad”* and *“boring”*. Interestingly we also find words like *“action”* , *“life”* and *“story”* represented for positive reviews while the negative reviews are more centred around words like *“plot”, “acting”* and *“people”.*

What's more captivating is what may not be represented in these words clouds. We don’t see any positive weighting related to graphics, sound and cinematic experience in general. It could be interesting to take a closer look at what exact movies were released in the last year, and explore the relationship between cinematic expenditures versus user response.

What model is best at classifying reviews?

In order to evaluate our models several evaluation criterions has been used, in this report however, our main focus is on overall accuracy of prediction of sentiment and not specific types of errors. Based on the outcome of the accuracy metric, Support Vector Machines seems to yield the best performance. But the best performance in accuracy is only around 0.09% more accurate than the next best model - Logistic regression. The logistic regression model yields an accuracy of 80.81% which is 30.81% better than the null accuracy of 50%, due to our balanced dataset. Taking implementation and real-world practices into account, the logistic model is in general much more convenient when analysing inference and also a much more parsimonious model, which is of high relevance in an implementation context. So based on the previous results, the recommendation would be to proceed with the combination of the advanced preprocessing and the logistic regression model.

9. Bibliography

(2001). *A maximum entrophy approach to natural language processing.* Yorktown: IBM Watson research center.

Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The Impact of Features Extraction on the Sentiment Analysis. *Elsevier*, 341-348.

Annett, M., & Grzegorz, K. (2008). A Comparison of Sentiment Analysis Techniques: Polarizing Movie Blogs. *Research Gate*, 25-35.

Asghar, M., Khan, A., Ahmad, S., & Kundi, F. (2014). A Review of Feature Extraction in Sentiment Analysis. *Basic and Applied Scientific Research*.

Avinash, M., & Sivasankar, E. (2020). *Efficient Feature Selection techniques for Sentiment Analysis.* Indiana, USA: Samsung R&D department.

Beckman, M., Guerrier, S., Lee, J., Molinari, R., Orso , S., & Rudnytskyi, I. (2020). *An Introduction to Statistical Programming Methods with R.*

Breitinger, C. (2015). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries.* , 305-338.

Clemons, E. K. (2010). *Do online reviews reflect a product's true perceived quality?* Singapore: Singapore management university.

Cumberas, M., Raez, A., & Galiano, M. (2013). Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Elsivir*, 6758-6765.

Feinerer, I. (2019). Introduction to the tm Package. In I. Feinerer, *Text Mining in R* (pp. 1-8).

Gregoire, M., Mikolov, T., Ranzato, M., & Bengio, Y. (2015). *ENSEMBLE OF GENERATIVE AND DISRIMINATIVE TECHNIQUES FOR SENTIMENT ANALYSIS.* Montreal: University of Montreal.

Hickey, W. (2016, 01 01). *Ghostbusters' Is A Perfect Example Of How Internet Movie Ratings Are Broken*. From Ghostbusters' Is A Perfect Example Of How Internet Movie Ratings Are Broken: https://fivethirtyeight.com/features/ghostbusters-is-a-perfect-example-of-how-internet-ratings-are-broken/

IMDb. (2020). *Terms of conditions*. From IMDB: https://www.imdb.com/conditions

Jabeen, H. (2013, 10 23). *Stemming and Lemmatization in Python*. From Datacamp: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Marques, M. (2019, 01 01). *Analyzing Movie Reviews - Sentiment Analysis*. From Kaggle.com: https://www.kaggle.com/mgmarques/analyzing-movie-reviews-sentiment-analysis-i

Marr, B. (2018, 05 21). *Forbes.* From How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#4afd8f4360ba

Mullen, T., & Collier, N. (2004). *Sentiment analysis using support vector machines with diverse information.* Tokyo: National Institute of Informatics.

Oghina, A., Breuss, M., & Tsagias, M. (2012). Predicting IMDB Movie Ratings Using Social Media. *ResearchGate*, 1-9.

Panda, A. K. (2018, 12 02). *NLP - SENTIMENT ANALYSIS ON IMDB MOVIE DATASET*. From Medium: https://medium.com/@GeneAshis/nlp-sentiment-analysis-on-imdb-movie-dataset-fb0c4d346d23

Pang, B., & Lee, L. (2002). *Thumbs up? Sentiment Classification using Machine Learning.* Itchaca: Cornell university.

Papers with code. (2019, 01 01). *Papers with code*. From State of the art sentiment analysis on IMDb: https://paperswithcode.com/sota/sentiment-analysis-on-imdb

Shimada, K., & Endo, T. (2008). *Seeing Several Stars: A Rating Inference Task.* Berlin: Springer.

Socher, R., Perelygin, A., Wu, J. Y., & Chuang, J. (2013). Recursive Deep Models for Semantic Compositionality. *Stanford University*.

Stanford University. (2018, 01 01). *Stanford.edu.* From IMDb Dataset: https://ai.stanford.edu/~amaas/data/sentiment/

Statistical tools for high-throughput data analysis. (2020, 01 01). *STHDA*. From Text mining and word cloud fundamentals in R : 5 simple steps you should know: http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know

Tripathy, A., Agrawal, A., & Rath, S. (2016). Classification of Sentiment Reviews using N-gram Machine Learning Approach. *ResearchGate*, 117-126.

Weiss, S. M., Indurkhya, N., Zhang, T., & Damerau, F. J. (2005). *TEXT MINING: Predictive Methods for Analyzing Unstructured Information.* Yorktown: Springer.

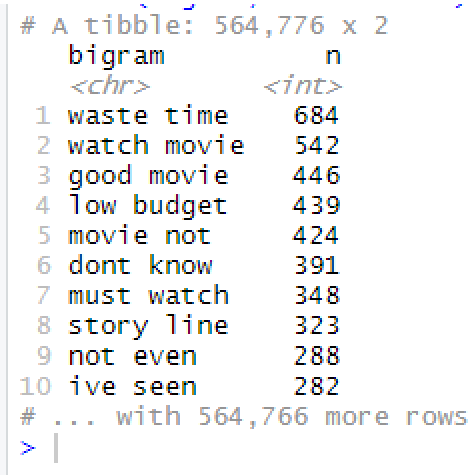
Wickham, H. (2014, 11 14). *rvest: easy web scraping with R*. From RstudioBlog: https://blog.rstudio.com/2014/11/24/rvest-easy-web-scraping-with-r/

Wickham, H. (2019, 11 08). *SelectorGadget*. From Cran.r: https://cran.r-project.org/web/packages/rvest/vignettes/selectorgadget.html

Xie, Q. (2020, 2 3). *Github*. From Unsupervised Data Augmentation: https://github.com/google-research/uda/blob/master/README.md

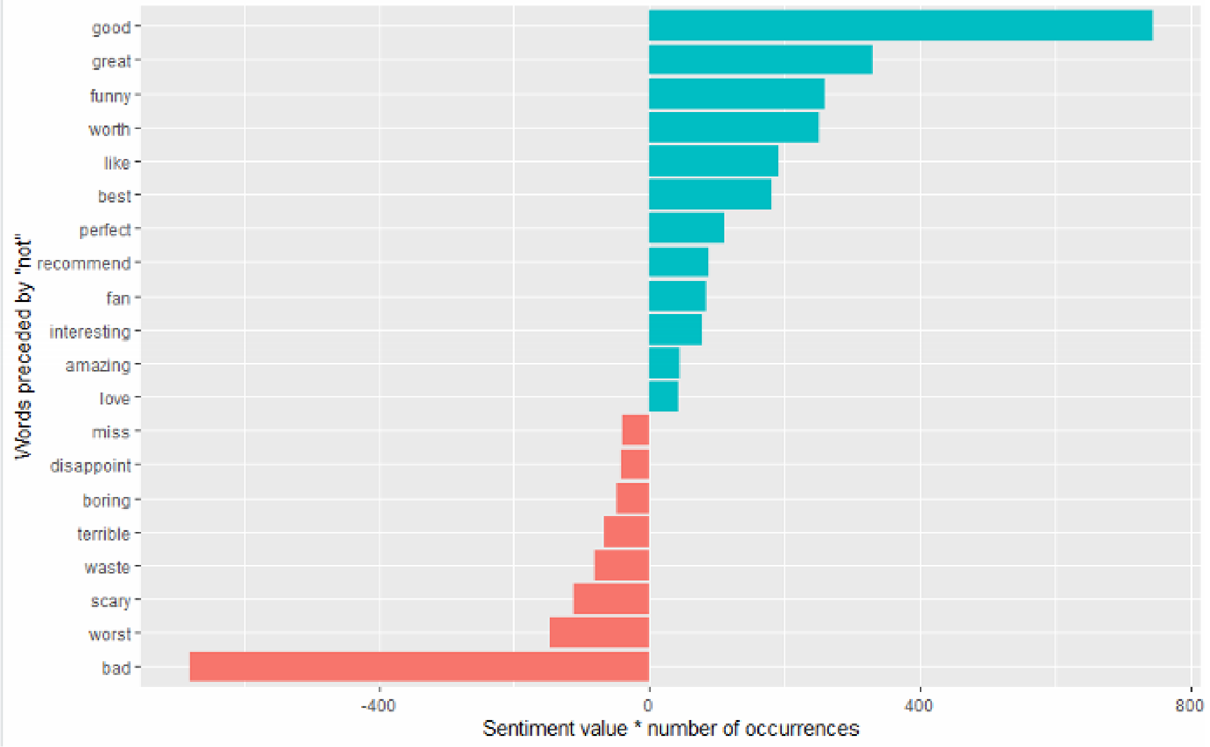
10. Appendix

Appendix 1

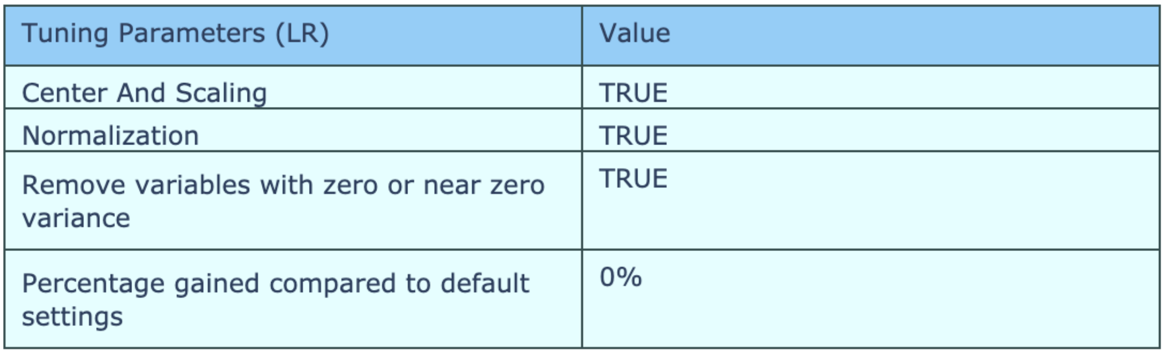


```
# A tibble: 564,776 x 2
   bigram          n
   <chr>       <int>
 1 waste time    684
 2 watch movie   542
 3 good movie    446
 4 low budget    439
 5 movie not     424
 6 dont know     391
 7 must watch    348
 8 story line    323
 9 not even      288
10 ive seen      282
# ... with 564,776 more rows
> |
```

Appendix 2

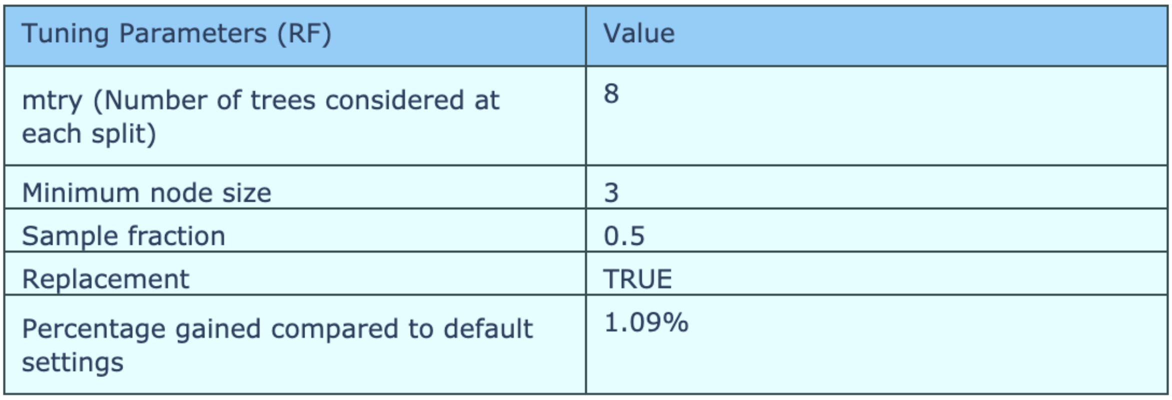


Appendix 3



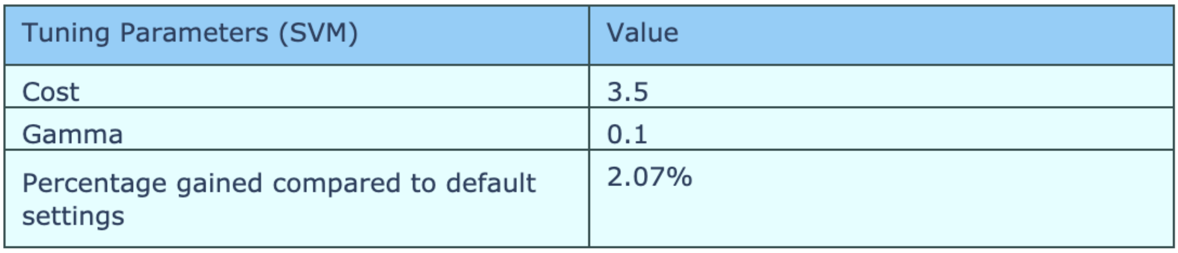
Tuning Parameters (LR)	Value
Center And Scaling	TRUE
Normalization	TRUE
Remove variables with zero or near zero variance	TRUE
Percentage gained compared to default settings	0%

Appendix 4



Tuning Parameters (RF)	Value
mtry (Number of trees considered at each split)	8
Minimum node size	3
Sample fraction	0.5
Replacement	TRUE
Percentage gained compared to default settings	1.09%

Appendix 5



Tuning Parameters (SVM)	Value
Cost	3.5
Gamma	0.1
Percentage gained compared to default settings	2.07%