Norwegian School of Economics
Bergen, Fall 2020

The Use of Textual Data Analysis and Machine Learning in Bankruptcy Prediction

Evaluating the predictive power of sentiment scores and ratios from news articles for bankruptcy prediction in the Norwegian market using machine learning

Torjer Stuland Bertelsen and Jonas Johansen

Supervisor: Øivind Anti Nilsen

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

This thesis is written during the fall of 2020 at the Norwegian School of Economics (NHH), as part of our Master of Science degree in Economics and Business Administration, majoring in Business Analytics.

It has been a challenging, but very rewarding process working on our thesis. We are thankful for the opportunity to pursue a research question that excites us. We believe and hope that our analysis represents a contribution to the fields of applied textual data analysis and bankruptcy prediction.

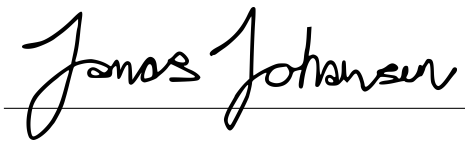
We would like to express our gratitude to several stakeholders contributing and guiding us throughout our research. First, thank you to Paul I. Huse at Enin for providing us with data and also contributing by sharing his knowledge within bankruptcy prediction modelling. We would also like to express our gratitude to Assistant Professor Maximilian Rohrer and Lecturer Christian Langerfeld at NHH for sharing their knowledge within the field of textual data analysis, as well as machine learning. Next, we would also like to thank Senior Engineer Svein Lamvik at NHH for providing access to well-specced virtual machines, enabling us to work with vast amounts of data and run our developed algorithms.

Lastly, we want to express our deepest gratitude to our supervisor, Professor Øivind Anti Nilsen at NHH. With his valuable guidance and support, the quality of our research has greatly improved.

Norwegian School of Economics

Bergen, December 2020

Torjer Stuland Bertelsen



Jonas Johansen

Abstract

In this thesis, we investigate whether there is predictive power in sentiment scores and ratios derived from news articles with regards to bankruptcy prediction of Norwegian private limited companies. Our analysis is based on Norwegian news articles and annual accounts from the Brønnøysund Register Centre. We derive sentiment scores and ratios by performing lexicon-based sentiment analysis on the news articles. The sentiment scores and ratios are averaged for four different time observation periods and are then matched with their belonging companies. Furthermore, we utilize Altman's five financial ratios to form our financial variables. Our models including both Altman's financial ratios and sentiment variables are in our analysis compared to a reference model only including the financial ratios.

In order to assess the problem we develop models using two different techniques, Generalized Linear Modelling and xgboost. Our emphasis is on comparing models with sentiment variables to reference models without sentiment variables in order to examine the potential predictive power of sentiment. We assess different model configurations, taking into account both different news observation periods and bankruptcy prediction horizons. The scores and ratios from the news observations are included on different time lags, ranging from 1 to 12 months prior to the announcement of annual accounts. The performance of the models is measured in AUC and balanced accuracy. In addition, we examine the average marginal effects in the developed GLMs and variable importance in the xgboost models.

The results of the applied methodology indicates that there is no significant improvement when including sentiment variables. The reference models utilizing only financial ratios tend to perform better than the models including sentiment variables in terms of AUC and balanced accuracy. In terms of marginal effects and variable importances, the financial ratios also tend to outperform the sentiment variables. Furthermore, we provide a nuanced discussion based on the presented approach and results, and point to further research approaches that we find promising.

Keywords – Bankruptcy Prediction, Textual Data Analysis, Sentiment Analysis, Predictive Analytics, Machine Learning, Big Data, xgboost, GLM

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and research question

Bankruptcy prediction models have many applications for many different users. Banks, investors and credit firms are all interested in being able to evaluate the healthiness of and risk associated with companies of interest. Public institutions are also interested in these models. Both the Central Bank of Norway and the Financial Supervisory Authority of Norway use bankruptcy models in order to investigate the credit risk of the banks. One could easily argue that bankruptcy models help to improve the financial market's ability to allocate capital to a lowest possible cost, by reducing information asymmetry. Well established models are performing seemingly well already. However, there are some obvious limitations when using financial accounts data. One prominent limitation is the inevitable time lag between the end of the accounting year and the publishing of the annual accounts, which is usually not published until months into the following year. This means the financial information is not reflecting the current situation when accessible.

The covid-19 pandemic has once again made financial distress and bankruptcy very hot topics. In such an uncertain environment, one might argue that established financial modelling is insufficient when quantifying the health of a company. Although market-based bankruptcy prediction models often account for external socioeconomic factors, these factors have traditionally been neglected in the established accounting-based bankruptcy models. As a result of this, experimenting with further development of existing accounting-based models is highly relevant. Particularly interesting is the addition of information that is both external and up to date.

Rapidly increasing computational power has given a foundation for new techniques and methods handling big data. This also includes unstructured data, such as textual data. In other words, there are emerging opportunities that have not yet been fully utilized. Textual data analysis has been a research field for decades, but due to the availability of increasing computational power and new methods being developed, the application of

textual data analysis is becoming increasingly popular.

The amount of accessible data on the internet is rapidly increasing and provides a lot of valuable information, if managed in a proper way. Textual data retrieved from news articles are no exception, as most newspapers today publish all their articles on the internet. Thus, an interesting topic to investigate is whether the inclusion of news data in an accounting-based bankruptcy prediction model is able to improve prediction accuracy. To our knowledge, such an analysis has never been conducted in the Norwegian literature.

The novelty of our thesis compared to previous literature is that we combine quantitative financial data with information extracted from news articles when developing bankruptcy prediction models. By calculating the sentiment polarity of the written news articles, we seek to investigate whether opinion rich textual data can provide predictive power in bankruptcy prediction. We assign individual sentiment values to every private limited company that has been mentioned in our dataset of Norwegian news articles published on the internet. Using different model configurations, prediction horizons and observation periods for news articles, we investigate whether bankruptcy prediction reference models that only utilize financial ratios improve when including sentiment variables.

This leads us to the research question that this thesis aims to answer:

Is there predictive power in sentiment scores and ratios based on news articles with regards to bankruptcy prediction of Norwegian private limited companies?

1.2 Overview of sections

This paper consists of in total 8 sections. Section 2 presents some prominent literature on bankruptcy prediction in addition to a recent study on bankruptcy prediction in combination with textual data. Section 3 presents the relevant methodology that we are using in the analysis. Section 4 introduces the data and furthermore presents how the variables of interest are derived and form the final datasets. Section 5 elaborates on the model development considerations. Section 6 presents the results obtained from the

analysis. Section 7 provides some topics of discussion with regards to limitations and further development. Finally, section 8 concludes upon the research question and presents the main takeaways from the analysis.

2 Literature

2.1 Bankruptcy Prediction Modelling

Beaver (1966)

Beaver is often regarded as a pioneer within bankruptcy prediction. The model he develops is a univariate model[1], individually investigating the predictive power of 30 financial ratios. The 30 financial ratios are split into five different categories, each category reflecting different parts of a company's financial structure. The dataset he uses consists of 79 bankruptcies and 79 non-bankruptcy observations over a five year time period. The main takeout from his study is that increasing reservoir and net liquid asset flow from operations lead to a smaller probability of bankruptcy, while larger amounts of debt and fund expenditures lead to higher probability of bankruptcy.

Although Beaver's study leads to some interesting findings and introduces financial ratios as explanatory variables in bankruptcy prediction, a clear disadvantage is the fact that the model only investigates the predictive power of the variables individually. Financial failures are usually more complex, thus a univariate model using only one explanatory variable at a time, will struggle to capture all the relevant dimensions of a firm. Beaver himself also points this out and mentions a multivariate model as a further development.

Altman's Z-score (1968)

As a further development of Beaver's model, Altman (1968) introduces the well-known Altman Z-score model, often applied and referred to in the literature ever since its publication. The model is a Multivariate Discriminant Analysis (MDA)[2] based on predefined financial ratios. The data sample in the model consists of 66 manufacturing firms. The class distribution is equal, meaning 33 of the firms are considered to be bankruptcy firms while the other 33 firms are considered to be non-bankruptcy firms. The

[1] A univariate analysis investigates the dependency of a single predictor and a response variable (Beaver, 1966)

[2] MDA is a statistical technique that is used to classify an observation into one of several groupings based on the individual characteristics of the observation (Altman, 1968)

bankrupt group consists of manufacturers that file a bankruptcy petition under Chapter X of the National Bankruptcy Act during the time period 1946-1965. Firms in the non-bankrupt group were still in existence in 1966. Furthermore, the non-bankrupt samples are carefully selected in order to match the bankrupt samples in terms of industry and firm size. Based on total asset value, Altman also removes relatively small and relatively large firms from the sample in order to prevent a skewed dataset. The feature foundation of the model is in total 22 financial ratios, either previously used in the literature or introduced by Altman based on intuition. The final model consists of the five ratios yielding the best overall performance.

Altman's final model is the following:

$$Z = 0.012X_1 + 0.014X_2 + 0.033X_3 + 0.006X_4 + 0.999X_5 \tag{2.1}$$

where

X_1 = Working capital/Total assets

X_2 = Retained earnings/Total assets

X_3 = Earnings before interest and taxes/Total assets

X_4 = Market value of equity/Total liabilities

X_5 = Sales/Total assets

The function above yields a Z-score which determines the modelled healthiness of a firm. The higher the Z-score, the smaller the probability of bankruptcy. A lower Z-score indicates a larger probability of bankruptcy. Furthermore, Altman introduces an upper threshold (2,67) and lower threshold (1,81) for the Z-score, aiming to minimize the number of misclassifications. An observation with a Z-score above the upper threshold is classified as non-bankrupt, while an observation with a Z-score below the lower threshold is classified as bankrupt. A Z-score in between these thresholds indicates uncertainty with regards to the classification.

Altman's results show a high predictive power one year before bankruptcy, with an overall

accuracy of 95% [3]. Furthermore, two years prior yields 72% accuracy, 3 years prior yields 48% , 4 years prior yields 29% and 5 years prior yields 36% . In other words, the predictive power is clearly diminishing over time, and when predicting on a horizon above 2 years you are better off guessing the outcome. Worth mentioning, is that these predictions are made in-sample, meaning the model is trained and tested on the same data. Such an approach is limiting the validity of the model since it is never tested on unseen data. When predicting out of sample on a one year horizon, Altman's model yields an accuracy of 79% (Altman, 1968).

Ohlson's O-score (1980)

Another well-known and established early-phase model is Ohlson's O-score model (1980). In his paper, Ohlson points out some shortcomings with Altman's MDA model. He points out the assumption of normal distributed variables in a MDA, which he argues is not realistic. To exemplify, he questions the required equality of the variance-covariance matrices of the predictors for the two classes: bankrupt and non-bankrupt firms. Furthermore, he argues that the output score of the MDA model has little intuitive interpretation. Lastly, Ohlson criticizes the matching of bankrupt and non-bankrupt firms in the MDA models. The use of criteria such as size and industry when matching bankrupt and non-bankrupt firms appears somewhat arbitrary. Instead, Ohlson suggests that the size of a firm should be included as a variable in the model. By using a conditional log-it analysis[4], he argues that the mentioned problems with respect to MDA can be avoided.

The dataset Ohlson uses in the analysis consists of financial information from 105 bankruptcies and 2 058 non-bankruptcies between the years 1970 and 1976. Distinguishable from most other literature, is that Ohlson bases his research on an imbalanced dataset meaning the classes are not evenly distributed. Similarly however, Ohlson is utilizing financial ratios. In addition, he includes binary variables and a company size measure.

[3]Overall accuracy is defined as correct classified observation divided by all observations (Altman, 1968)

[4]Conditional log-it analysis allows for the individual observations to face observation-specific conditions. The coefficients are the same for all observations, but the value of a given independent variable can be observation-specific. Another property of the conditional logit-model is that the output probabilities are constrained to the range of zero to one (Adkins, 2014).

The final model is presented as the following:

$$O - score = -1.32 - 0.407(Size) + 6.03(TLTA) - 1.43(WCTA) + 0.076(CLCA)$$
$$-1.72(OENEG) - 2.37(NITA) - 1.83(FUTL) + 0.285(INTWO) - 0.521(CHIN) \tag{2.2}$$

where

SIZE = Log (Total assets/GNP price-level index)

TLTA = Total liabilities/Total assets

WCTA = Working capital/Total assets

CLCA = Current liabilities/Current assets

OENEG = 1 if total liabilities exceed total assets, 0 otherwise

NITA = Net income/Total assets

FUTL = Funds provided by operations/Total liabilities

INTWO = 1 if net income was negative for the last two years, 0 otherwise

CHIN = Change in net income

The interpretation of the equation is that the higher the O-score, the higher the probability of bankruptcy. An O-score above 0,5 indicates a potential bankruptcy within a predefined year. An O-score below 0,5 indicates that the firm is healthy. Ohlson introduces in total three models with one-, two- and three-year prediction horizons respectively. The three models yield accuracies of 96.12% , 95.55% and 92.84% , based on in-sample predictions. When predicting out-of sample, a one-year horizon yields an accuracy of 85% (Ohlson, 1980).

Bellovary *et al.* (2007)

The *meta-study* from Bellovary *et al.* (2007) collects and examines in total 165 studies on Bankruptcy prediction. The aim of the study is to compare the methodologies and results obtained, and also examine the variables being included in different models, in order to facilitate more productive future research within this area. The paper presents some interesting findings that are highly relevant for this thesis.

To begin with, the paper points out the lack of a generally accepted definition of bankruptcy and the existence of different interpretations of the concept among researchers. These are mentioned as some prominent reasons for the various non-unified models throughout the history of bankruptcy modelling. A diverse set of definitions of the phenomenon have been assessed in different bankruptcy studies. Often, the actual filing for bankruptcy or liquidation is used. However, some studies regard financial stress or inability to pay financial obligations as a bankruptcy trigger. The paper also points out that some studies do not provide a sufficient definition. The varying definitions overall make it more difficult to compare the various models (Bellovary *et al.*, 2007).

Furthermore, an important topic of discussion in the paper is the different methodologies being applied. The paper presents a trend with regards to the techniques being used in the different studies over time. In earlier studies, the Multivariate Discriminant Analysis (MDA) method was often used. Over time however, logit and probit models have experienced an increase in popularity. Both logit and probit models take the probability of bankruptcy into account and the main difference between the two is that probit models require non-linear estimation. Furthermore, Neural Networks increased in popularity in the late 1980s. The Neural Networks models are designed to emulate the human pattern recognition function. In recent years, even more specialized methods are becoming increasingly popular. The authors present the increased availability of computational power as an important contributor to this trend (Bellovary *et al.*, 2007).

Another trend the authors point out is regarding the validation process of the bankruptcy prediction models. In-sample validation has been used in earlier years, while out-of-sample validation has gained more attention in more recent years. When measuring the performance of the different models, the literature has continuously referred to the previously mentioned overall accuracy in addition to Type I and Type II errors. Type I errors refer to misclassification of non-bankrupt firms as bankrupt, while Type II errors refer to bankrupt firms misclassified as non-bankrupt firms. Furthermore, it has generally been agreed upon in the literature that Type II errors are more costly than Type I errors. The mentioned error rates have been increasingly emphasized in the litterature in more

recent years. Type I and Type II errors are also referred to as false positives and false negatives (Bellovary *et al.*, 2007).

A consistent trend throughout the many decades of bankruptcy prediction studies is that the majority of the models are based on balanced datasets. In a case of bankruptcy prediction, a balanced dataset would imply an even distribution of bankruptcies and non-bankruptcies. Since bankruptcy is a rare event in real life, models based on balanced data can potentially perform poorly on real life data. When comparing models based on balanced datasets with models based on imbalanced real-life datasets, one would need to be especially aware of the performance measures being used (Bellovary *et al.*, 2007).

A last important discussion topic presented by the authors is the selection of features and ratios. The paper concludes that the average amount of ratios has been varying over time, but remains around 10 overall. The ratios themselves are also of significant importance. The paper emphasizes that more factors does not necessarily increase accuracy and mentions two-factor models performing as good as 21-factor models. The actual feature selection is far from standardized and can appear as a cherry-picking process trying to capture all financial sides of a company. However, interesting to note is that the five financial ratios from Altman's original model are well-represented. The paper provides an exhausting list of all ratios being used in research and their respective count throughout time. All of Altman's five ratios appear among the most used ratios (Bellovary *et al.*, 2007).

2.2 Textual Data Analysis in Bankruptcy Prediction

Although the literature on textual data analysis in combination with bankruptcy prediction is scarce, some researchers have been investigating the field in recent years and have presented some interesting findings. Particularly interesting for the analysis in this thesis is a paper by Nam-Ok Jo and Kyung-Shik Shin from 2016.

The paper "Bankruptcy Prediction Modelling Using Qualitative Information" by Jo and Shin (2016) points out the numerous academic studies throughout the decades investigating

different techniques and features used in bankruptcy prediction. Furthermore, the authors stress that the use of external qualitative information has been neglected, although financial accounting data has some obvious flaws like the inevitable time lag between the point of closed financial statements and credit evaluation. In addition, the authors argue that the financial ratios do not include environmental considerations, such as the external economic situation.

The authors carry out an experiment investigating the predictive power of sentiment scores. The aim of the study is to analyze the overall aspects of the economic situation in the construction industry. Financial data is gathered from financial statements of in total 916 Korean small and medium sized construction firms. The sample consists of 458 bankruptcy and 458 non-bankruptcy firms from the period 2008 to and including 2012. The horizon of bankruptcy observations is 12 months after the credit evaluation date. Based on univariate analysis and expert opinions, the selected financial data for the study is five different ratios.

The textual data is acquired from in total 81 318 economic news articles, all containing the keyword "construction". By utilizing big data analytics techniques such as sentiment analysis, they are able to process the qualitative information in the economic news articles. The sentiment scores are incorporated at an industry level and they are meant to represent a quantification of the external economic atmosphere as presented in the media. The methodology proposed in the paper is a lexicon-based sentiment analysis[5]. The lexicon of choice is a construction-specific sentiment lexicon, derived by the authors using news articles in order to represent construction companies. It is designed to capture the relationship between an occurring term in a news article and the industry's economic situation as a whole. The news sentiment variables are incorporated in the presented models in the time period in between settlement date and evaluation date of the financial statements. Different time lags are tested, and the predictive power of each news period observation are compared. The time lags tested in the analysis vary from 1 month to 5

[5]A lexicon-based sentiment analysis utilizes pre-defined lexicon as look up tables in order to classify or quantify the polarity of textual data (Langerfeld & Rohrer, 2019a). See section 3.1.1 for further explanation.

months after the settlement date of annual accounts, meaning the end of the accounting year. This way, the sentiment variables are meant to supplement limited accounting information and fill in the time lag where no updated financial information is available.

The experiment concludes that the qualitative information incorporated as sentiment scores, contribute to correctly predicting bankruptcy firms. Furthermore, the authors stress that the obtained results are highly dependent on the lexicon that is applied in the sentiment analysis.

3 Methodology

3.1 Textual Data Analysis

The process of textual data analysis, also referred to as text mining, aims to obtain valuable insights from unstructured text. Extracted high quality information from text can subsequently be used in decision making in different fields. The employment of computers for execution of algorithms, enables faster processing of digital information, detection of high dimensional patterns and structured analysis on textual data (Langerfeld & Rohrer, 2019b). The field of textual data analysis is diverse and there are several different approaches that can be used in order to obtain valuable insights.

3.1.1 Sentiment analysis

Sentiment analysis is an often-used approach within textual data analysis and has been successfully applied within different business contexts. A sentiment analysis seeks to quantify and/or classify the sentiment polarity of a text. Opinion-rich text can be exploited to capture valuable insight. One variation of this method is to assign sentiment scores to words, sentences or complete texts. Another is to classify the textual data as positive, negative or neutral. There are several different approaches within sentiment analysis, with two common ones being the lexicon-based approach and the supervised machine learning approach, where one can utilize methods such as *Artificial Neural Networks (ANN)*[6](Langerfeld & Rohrer, 2019a). In this paper, the lexicon-based approach is to be applied.

Lexicon-based sentiment analysis utilizes either dictionaries or corpuses in order to determine the textual polarity (Jo & Shin, 2016). There are several available predefined dictionaries that can be utilized in order to examine sentiment in texts. Some of the dictionaries are regarded as general-purpose dictionaries, like the Jockers & Rinker-dictionary. Others are domain-specific, developed in order to capture the sentiment

[6]Artificial Neural Networks are simulating the human mind, utilizing interconnected neuron nodes and backward propagation to improve the obtained results (Bellovary *et al.*, 2007). The method has several different variations and we refer to other sources for more comprehensive explanations.

polarity within a domain. The dictionaries contain words and their associated polarity values, usually between -1 and 1. By looking up words in the provided dictionary, the polarity of input text can be calculated or classified. Important to note is that the results of this approach will vary based on the dictionary of choice. In addition, the language input can create problems. It can be hard to capture any value if the text contains slang and misspellings (Langerfeld & Rohrer, 2019a).

The package *sentimentr* in R provides the function *sentiment()* which calculates sentiment scores based on input text and lexicon. The calculations and assumptions within this function will be derived as presented by the author Tyler Rinker (2019). Each paragraph p is divided into sentences s; $p_i = \{s_i, ..., \ s_n\}$. Furthermore, each sentence s is broken into ordered words w; $s_j, j = \{w_i, ..., \ w_n\}$. All punctuations, except for comma words cw, are removed. We will denote each word as $w_{i,j,k}$, word in paragraph i, sentence j and word number k. Every word $w_{i,j,k}$ is searched for and compared to a dictionary consisting of polarized words, e.g. the previously mentioned Jockers & Rinker-dictionary. Furthermore, each word is tagged as either positive $w^+_{i,j,k}$ or negative $w^-_{i,j,k}$, and assigned a value in between -1 and 1. Polarized words will be denoted as pw and form polarity clusters $c_{i,j,l}$, which again are subsets of the sentences, $c_{i,j,l} \subseteq s_i, j$.

Next, the concept of valence shifters will be introduced. Valence shifters are words that alter or intensify the polarity of the words of interest. Each valence shifter is categorized as either a (1) negator, (2) amplifier, (3) de-amplifier or (4) adversative conjunctions. The clusters $c_{i,j,l}$ are used for these calculations and represent the polarized context of each word pw. The default area of polarized context and hence activation of valence shifters, is the four words before (nb) and two words after (na) the pw. The parameters nb and na can be determined by the user in the function as *n.before* and *n.after* respectively. The context clusters can be derived as follows; $c_{i,j,l} = \{ \ pw_{i,j,k-nb}, ... \ , pw_{i,j,k}, ... \ , pw_{i,j,k-na} \ \}$.

Next, the words in these clusters are tagged as one of the four categories; neutral $w^0_{i,j,k}$, negator $w^n_{i,j,k}$, amplifier $w^a_{i,j,k}$ or de-amplifier $w^d_{i,j,k}$. In the cluster equation above, the neutral words will not provide any values, but they will be considered in the total word

count n. All polarized words are individually weighted with weight w based on the weights provided by the input dictionary via the input argument *polarity_ dt*. In addition, the words will be further weighted by the valence shifters surrounding a positive or negative word, pw.

The amplifiers $w^a_{i,j,k}$ will increase the polarity of a given word by a predefined weight z. The default value of z is 0.8. However, if the context cluster of interest contains an odd number of negators $w^n_{i,j,k}$, the amplifier will become a de-amplifier $w^d_{i,j,k}$. An example of a negator is the word "not" . In such a case, the de-amplifier will decrease the polarity of the word. In addition, the negators will flip the polarity of a polarized word, meaning a positive polarity value is flipped to a negative polarity value. The exact negation is derived by raising -1 to the power of the number of negators + 2. The author justifies this determination by pointing out that two negative words yield a positive, while three negative words yield a negative etc.

Furthermore, also the valence shifters categorized as adverse conjunction will influence the polarity. If an adverse conjunction, e.g. "however" or "but" , appears before the polarized word $w_{\text{adversative conjunction}}, ..., w^p_{i,j,k}$, the cluster will be up-weighted by $1 + z_2 * \{|w_{\text{adversative conjunction}}| \ , ..., w^p_{i,j,k}\}$, where z_2 has a default weight equal to 0.85. On the other hand, if the adverse conjunction appears after the polarized word, the cluster will be down-weighted by $1 + \{ \ w^p_{i,j,k}, \ ..., \ |w_{\text{adversative conjunction}}| * -1\} * z_2$. Rinker (2019) argues that adverse conjunctions make the next clause of greater value, while the prior clause is made of lower value.

The author also introduces upper and lower bounds that are implemented in the function. In order to do so, the mentioned comma words cw are considered. Each cw is indexed in order to function as lower and upper bounds for the mentioned polarized context cluster. According to the author of the function, the reason for this is that these cw´s indicate a change of thought and the word before and after a cw are not necessarily connected. The following constraints are thus implemented for the polarized context clusters; upper bound $min\{pw_{i,j,k+na}, w_{i,jn}, min\{cw_{i,j,k} > pw_{i,j,k}\}\}$ and lower bound

$max\{pw_{i,j,k-nb}, 1, max\{cw_{i,j,k} < pw_{i,j,k}\}\}$. $w_{i,jn}$ equals the number of words in the sentence.

Finally, the polarity scores are derived as follows. The weighted context clusters $c_{i,j,l}$ are summed as $c'_{i,j}$ and divided by $\sqrt{w_{i,jn}}$, where w is the word count. This yields an unbounded polarity score, $\delta, persentence$:

$$\delta = \frac{c'_{i,j}}{\sqrt{w_{i,jn}}} \tag{3.1}$$

where

$$\begin{gathered}
\mathrm{c}'_{i,j} = \sum((1 + w_{amp} + w_{deamp}) * w^p_{i,j,k}(-1)^{2+w_{neg}}) \\
\mathrm{w}_{amp} = (w_b > 1) + \sum(w_{neg} * (z * w^a_{i,j,k})) \\
\mathrm{w}_{deamp} = max(w_{deamp'}, -1) \\
\mathrm{w}_{deamp'} = (w_b < 1) + \sum(z(-w_{neg} * w^a_{i,j,k} + w^d_{i,j,k})) \\
\mathrm{w}_b = 1 + z_2 * w_{b'} \\
\mathrm{w}_{b'} = \sum(|w_{adversativeconjunction}, ..., w^p_{i,j,k}, w^p_{i,j,k}, ..., |w_{adversativeconjunction} * -1) \\
\mathrm{w}_{neg} = (\sum w^n_{i,j,k}) mod 2
\end{gathered}$$

3.1.2 Preprocessing of textual data

When applying textual data methods like the mentioned sentiment analysis, there is usually a need for preprocessing of the raw textual data. The reason for this is possible gains from both increased accuracy and decreased computation time. Preprocessing of textual data can include the following: lowercasing, stemming, lemmatization, normalization and removal of digits, stop-words, punctuation and noise (Ganesan, 2019).

Some of the mentioned preprocessing steps are considered “must do” , some are “should do” and some are task dependent. There is no one-size-fits-all approach (Ganesan, 2019). Often applicable during the preprocessing steps are *regular expressions,* functions which filter textual data and prepare it for analysis. An example is the *gsub()*-function in the R base package which operates on patterns found in strings.[7] The *gsub()*-function is useful

[7]A string is an ordered sequence of character data (Ganesan, 2019)

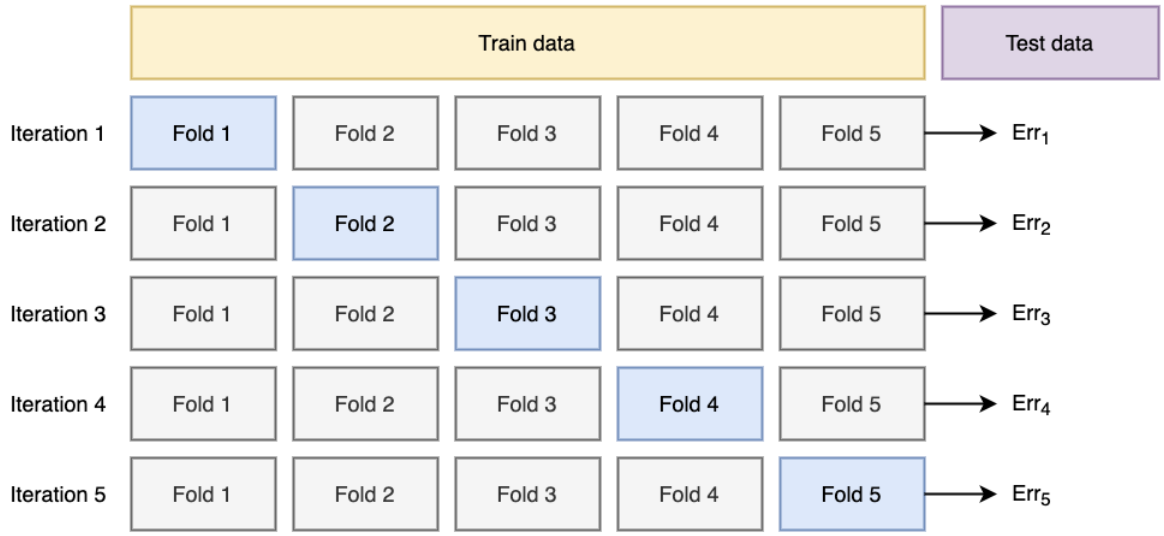
when finding, replacing or removing parts of strings.

3.2 Estimation and validation

The validation set approach involves a random split of the total dataset into train and test data, given a predefined ratio. First, the model is trained using the train dataset. After training the model, the fitted model will make predictions on unseen observations in the test dataset. This way the model can be evaluated based on out of sample data. The method is straightforward, easy to implement and will in some cases yield good results. However, a downside is that the false rate on the test data potentially has high variance, since it is directly dependent on the randomly chosen observations in the estimation train set. In addition, the validation set approach is prone to overfitting, meaning the model is too closely fit to the train dataset and not performing well on new unseen data (James *et al.,* 2013).

Cross-validation is an efficient way of dealing with the mentioned challenges of high variance and overfitting, ensuring good validity of the models. The method is initialized by dividing the already defined train dataset into k folds, k being the predefined number of folds. The folds are equal in size and non-overlapping. Furthermore, the model will be estimated k times, using all but one fold for estimation (k-1) and using the one omitted fold for validation. After each of the k estimations the one omitted validation fold will rotate, meaning all of the k folds will be used as validation fold after all k estimations. This process is summarized in figure 1, using $k = 5$.

Figure 3.1: K-fold cross-validation with 5 folds



In figure 3.1 the blue folds represent the validation fold, while the four grey folds represent the folds used for estimation for each iteration. All folds are derived from the training data, seen in yellow in figure 3.1. Furthermore, the error rate for k folds when validating using the omitted folds, can be derived as follows:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{K} Err_{(i)} \tag{3.2}$$

Where Err_i is the error rate for each k iteration (James *et al.*, 2013). The final estimated model utilizing the output probabilities can be derived by averaging the probability predictions of all k estimated models. The performance of the final estimated model will be determined by its ability to predict unseen data, in other words the hold-out test data from the initial train-test split seen in purple in figure 3.1.

In the context of cross-validation, it is necessary to introduce the concept of the *bias-variance trade-off*. The foundation for this concept is that in order to minimize the expected test error, the statistical method used when estimating the model needs to simultaneously achieve low variance and low bias (James *et al.*, 2013). In this context, the

variance refers to the amount an estimated function $\hat{f}$ will change when estimated using a different training dataset. When varying training dataset, the $\hat{f}$ will vary to some degree. However, ideally this variation will be very small when varying training dataset. Using a method with high variance will result in large changes in $\hat{f}$ when there are only small changes in the training dataset. *Bias* on the other hand, refers to the error introduced in the model as a result of an approximation of a real-life problem. In general, the more flexible the method is, the more variance and the less bias it has. Too much variance leads to a model that finds non-existing patterns for example by fitting a line that goes through every single observation in the training data. Such a model would perform well on training data but yield high errors on hold-out test data. Too much bias, however, leads to a model that oversimplifies the problem and tends to linearity for example by fitting a horizontal line to the training data resulting in high errors both on training data and test data (James *et al.*, 2013).

When using k-fold cross-validation, the bias-variance trade-off can be adjusted by appropriately selecting k. A lower k leads to lower bias and higher variance. A higher k leads to lower variance and higher bias. Empirically it has been proven that choosing $k = 5$ or $k = 10$ results in an acceptable trade-off. In addition, not increasing the k even further has an advantage in less need for computational power. Thus, one of these mentioned k values are typically chosen (James *et al.*, 2013).

3.3 Measures of performance

Confusion Matrix

In a binary classification problem, there are four possible outcomes when hard predicting given a predefined threshold: *True Positive (TP), True Negative (TN), False Positive (FP),* and *False Negative (FN).* Throughout this paper we refer to bankruptcy as the positive outcome with its binary response variable being assigned the value of 1, while we refer to non-bankrupt as the negative outcome with an assigned value of 0. These four possible outcomes can be visualized in a confusion matrix, displayed in table 3.1:

Table 3.1: Confusion matrix

		Actual values	
		Positive (1)	Negative (0)
Predicted values	Positive (1)	TP	FP
	Negative (0)	FN	TN

In a case of bankruptcy prediction, *True positive (TP)* is bankruptcy companies correctly classified as bankrupt. *True negative (TN)* is non-bankrupt companies correctly classified as non-bankrupt. *False positive (FP)* is non-bankrupt companies incorrectly classified as bankrupt. *False negative (FN)* is bankruptcy companies incorrectly classified as non-bankrupt. The confusion matrix is not a performance measure itself, but based on the output of the confusion matrix we can derive several performance measures.

Accuracy and balanced accuracy

Accuracy is often used to evaluate the performance of prediction models. One reason is the good interpretability of the measure. *Accuracy (ACC)* is defined as:

$$ACC = \frac{TP + TN}{(TP + FN) + (TN + FP)} \tag{3.3}$$

However, when dealing with imbalanced data sets accuracy is not a preferred performance measure. In a case of very imbalanced data, the accuracy could be close to 100% just by predicting all observations to be equal to the majority class, which is the binary class with the greatest number of observations. To exemplify using bankruptcy prediction, all companies could be classified as non-bankrupt and the accuracy would be considered very good. This would be very misleading, since the performance measure is not emphasizing the model's ability to correctly classify bankruptcy companies. An additional performance measure derived from accuracy, is the *balanced accuracy.* This performance measure takes class distribution into account. It does this by taking into the rates of true positives and true negatives, referred to as *sensitivity* and *specificity* respectively. These rates are derived as follows:

Sensitivity:

$$TPR = \frac{TP}{TP + FN} = 1 - FNR \tag{3.4}$$

Specificity:

$$TNR = \frac{TN}{TN + FP} = 1 - FPR \tag{3.5}$$

We can further derive the *balanced accuracy (BA)*, defined as:

$$BA = \frac{TPR + TNR}{2} \tag{3.6}$$

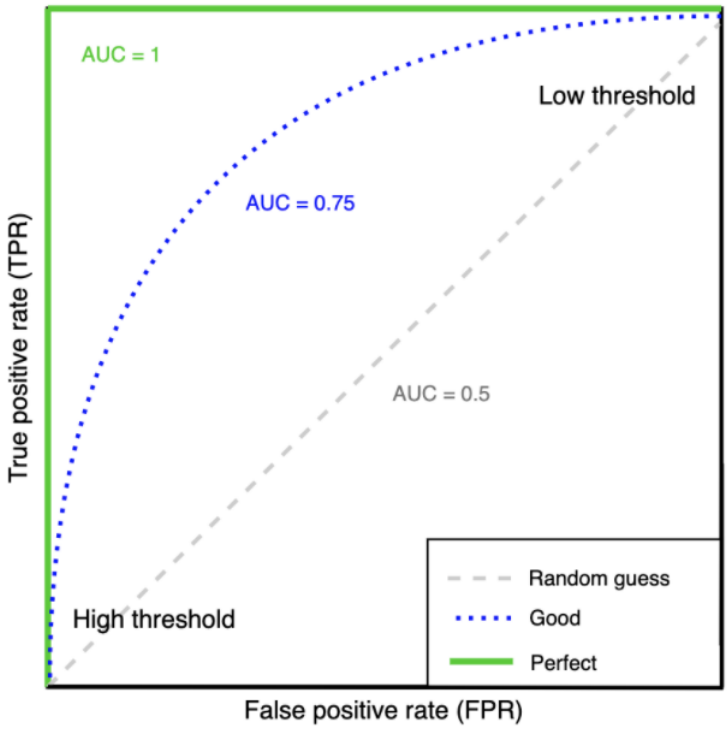
Setting classification thresholds

When applying a model that outputs probabilities assigned to every observation, the threshold for predicting bankruptcy can be adjusted in order to adjust the obtained true and false rates. The threshold should be adjusted to every individual problem based on the cost related to each false rate. Domain knowledge is critical when deciding the best threshold for a given problem (James *et al.*, 2013). In a case of bankruptcy prediction, it is often preferred to avoid incorrectly classifying bankruptcy firms as non-bankrupt, meaning false negatives. One reason is the large costs associated with for example investing in or cooperating with unhealthy firms. In conclusion, the optimization of threshold is dependent on the prediction problem of interest.

Receiver Operating Characteristics

The *Receiver Operating Characteristics (ROC)* curve visualizes the performance of a classification problem by graphing the trade-off between the presented rates; *TPR* and *TNR*. Figure 3.2 illustrates three different ROC curves and their belonging *Area Under the Curve (AUC)*.

Figure 3.2: ROC curves illustration



The *ROC* reflects all possible thresholds, in other words how varying the classification threshold impacts the *TPR* and *FPR*. An advantage of the *ROC* curve is that this trade-off can be chosen based on the aim of the prediction model. If you want to achieve higher sensitivity, you will need to compromise on specificity (James *et al.*, 2013). In addition, the *ROC* curve has useful properties when dealing with skewed distributions, meaning imbalanced data, and unequal classification error costs. *ROC* curves are insensitive to changes in class distribution, meaning that the curve remains constant when changing the proportion of negative and positive observations (Fawcett, 2006).

Area Under the Curve

The *Area Under the Curve (AUC)* is a performance measure derived from the area under the *ROC* curve. The higher the *AUC* value, the better the model is at distinguishing the two classes. An *AUC* of 1 indicates a perfectly performing model, seen as the green solid line in figure 3.2. In such a case, the optimal point in the top left corner will yield 100% *TPR* and 100% *TNR* (1 - *FPR*). An *AUC* of 0.5 indicates a model performance no better

than random guessing, seen as the grey dashed line in figure 3.2. The blue dotted curve represents an AUC of 0.75, reflecting a model that has some ability to distinguish the classes.

The use of AUC as a performance measure within bankruptcy prediction is often preferred both in the literature and when applied in the financial markets. When working with imbalanced data such as the uneven distribution between companies that go bankrupt and those that do not, AUC is a preferable performance measure since the ROC curve is insensitive to changes in the proportions of the two classes. Hence, AUC will be an appropriate supplementary performance measure to the mentioned *balanced accuracy* for imbalanced classification problems.

3.4 Handling imbalanced data

An imbalanced dataset is present when the minority class is very underrepresented compared to the majority class. The presence of an imbalanced dataset could be either a result of the data collection approach or an actual reflection of a real-life scenario. When present, it can affect the reliability and quality of the results of machine learning problems. When there is less information about the one of the classes, it generally becomes harder to accurately predict occurrences of the minority class (Burnaev *et al.*, 2017).

The challenge of an imbalanced dataset can be faced in several ways. Often proposed is resampling using over- and/or under-sampling. Examples of applicable methods are *Synthetic Minority Oversampling Technique (SMOTE)* and *random under-sampling (RUS).* The *SMOTE* involves introduction of new synthetic data based on *k-nearest neighbors*[8], while the *RUS* method randomly removes observations from the majority class. Hence, the techniques lead to observations and class distributions in the datasets that are not reflecting the real-world situation (Burnaev *et al.*, 2017).

[8] A k-nearest neighbor algorithm classifies or sorts observations based on their features. In the context of oversampling, this method will group k nearest neighbors, join them and create synthetic samples in this space (Walimbe, 2017).

Another approach aiming to increase the importance of the minority class is to adapt the probability threshold which separates the classes (Burnaev *et al.*, 2017). In his literature, Berg (2007) is using this approach when handling imbalanced data in bankruptcy prediction research. He argues that resampled data is non-representative for the actual population. Thus, he claims that these techniques will reduce the accuracy and application in the real world. Instead of introducing resampling techniques to balance the data, he suggests lowering the threshold for predicted bankruptcy. Lowering the threshold results in lower overall accuracy since more non-bankruptcy firms will be classified as bankrupt. On the other hand, he proves that the true positive rate (TPR) will increase. In his paper, Berg (2007) lowers the threshold to 10% , meaning a firm with an assigned probability of bankruptcy above 10% is classified as bankrupt.

3.5 Generalized Linear Model

A generalized linear model (GLM) is a further development of linear regression, introducing flexible generalization. The flexible generalization allows the response variable to have an error distribution that is not normally distributed. The model was first introduced by Nelder and Wedderburn (1972), meant to unify several existing statistical methods, such as Poisson regression, linear regression and logistic regression. The generalization of a linear model can be approached in different ways, typically the logistic model is utilized (James *et al.*, 2013). The GLM is a conventional method useful when evaluating the predictive power of the individual explanatory variables. By including a GLM in our analysis, we are able to observe the explanatory variables' individual effects. Furthermore, our analysis applies the generalization using the logistic regression link function with multiple predictors, which is defined as follows (James *et al.*, 2013):

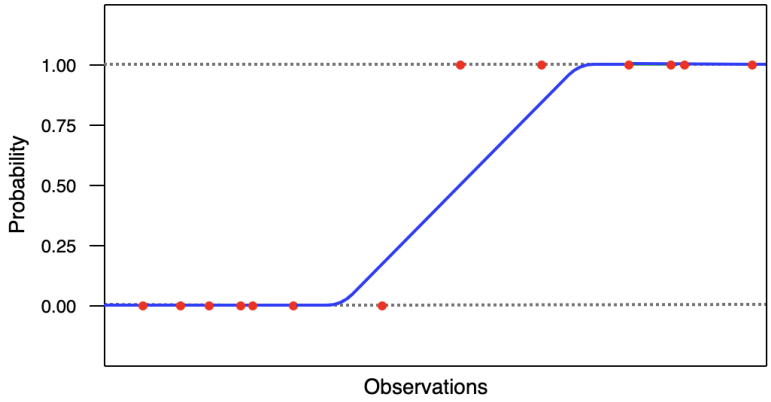
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p \tag{3.7}$$

where $X = (X_1....X_p)$, and p are the p predictors. The β -values $(\beta_0, ..., \beta_p)$ are calculated using *maximum likelihood.* The equation above can be rewritten as follows:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}} \tag{3.8}$$

The output *p(X)* is the distribution of probabilities with values between 0 and 1. A companies assigned $p_i(x_i)$ represents the probability of bankruptcy for the company. These output probabilities form a S-shaped curve with values in the space [0,1]. An illustrative example of a logistic curve in a classification problem is presented in figure 3.3.

Figure 3.3: Probability distribution in a classification problem for logistic regression



The β_i coefficients, estimated using maximum likelihood, are contributing in assigning a probability $p_i(x_i)$ to every observation. The intuition behind the maximum likelihood method is as follows. When estimating the model using the training dataset, the main goal is to assign probabilities that correspond to the actual observation response values. In order to do this, the model is trained to find the optimal β_i values. In a case of bankruptcy prediction, this involves assigning bankrupt companies a $\hat{p}_i(x_i)$ closer to 1 and non-bankrupt companies a $\hat{p}_i(x_i)$ closer to 0. In a simplified case with two β *values (* β_0, β_1 *),* the concept is mathematically formulated as the following *likelihood* function (James *et al.*, 2013):

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})) \tag{3.9}$$

where the estimates of the β -values $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen in order to maximize function displayed above.

The R package *caret* provides a *train()*-function that allows us to train a GLM. The necessary input for the function is a training dataset along with a binary response variable that we seek to predict. Furthermore, the function provides some useful properties when

training a GLM. In the case of imbalanced data, the function provides a sampling input where we can input a resampling method.

Amongst the outputs from a GLM are the coefficients of every independent variable along with a p-value from a test of significance. The test considers every explanatory variable individually and the null hypothesis for the test is that there is no relationship between the response variable and the individual variable. Given that there is no high correlation between the independent variables, the p-values from the test provide information about whether the individual variables are contributing to predicting the response variable at different levels of significance.

Furthermore, the marginal effect of each explanatory variable can be derived from the GLM. In GLMs such as the logistic regression, the marginal effect will vary for each individual observation, meaning there is no single constant marginal effect for the sample as a whole. One commonly used approach is to present the *average marginal effect (AME)* for each explanatory variable. The *AME* is calculated by averaging across all marginal effects at every observed value of the given explanatory variable. The calculated values can be interpreted as the average rate of change that happens instantaneously for the probability when a small change is made to the explanatory variable (Leeper, 2018a). The *AME* values in a developed GLM can be calculated using the *margins()*-function provided in the R-package *margins*. The function outputs the *AME* values assigned to each variable in addition to test statistics obtained from a statistical test of significance (Leeper, 2018b).

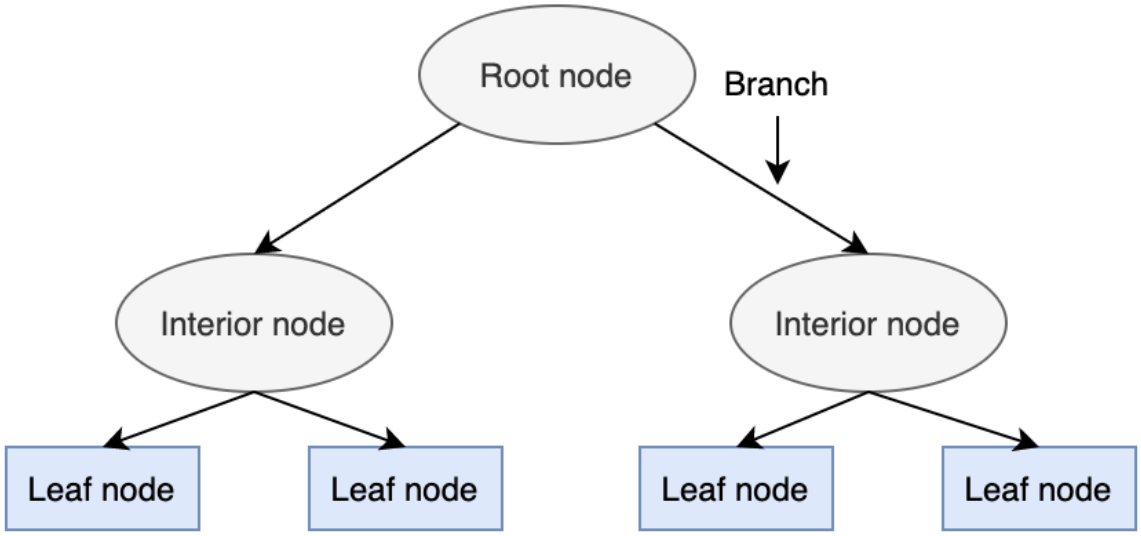
3.6 Extreme gradient boosting with xgboost

Xgboost is a *decision tree-based model*, meaning it utilizes decision trees when training and building the model. Decision trees are simple and intuitive supervised machine learning[9] methods that can be used to solve both classification and regression problems. Every

[9]Supervised machine learning is the algorithmic task of learning a function that maps an input x to an output y based on training input-output observation pairs (Brownlee, 2020).

individual tree is built using *if-else*[10] conditions and seeks, in a classification problem, to classify an observation according to some given conditions. They usually consist of a root node, branches, interior nodes and leaf nodes. Each of the internal nodes denote a test on a given attribute, the branches display the outcome of the test and the leaf nodes display the class label (James *et al.*, 2013). Figure 3.4 displays an example of a decision tree structure.

Figure 3.4: Illustration of a decision tree



Methods utilizing decision trees are proven to often yield high predictive power compared to other machine learning techniques used within the field of predictive analysis. Furthermore, the nature of the decision tree-based model brings several benefits particularly useful for our analysis. First of all, the models handle missing values by default. In decision tree algorithms, branch directions where there are missing values are learned during training. Furthermore, a model's decision tree foundation makes it less prone to multicollinearity. Compared to other methods like a generalized linear model (GLM), where the features are assumed to be uncorrelated, decision tree algorithms are in general more robust to multicollinearity (Benesty *et al.*, 2018). We therefore find it interesting to see how a decision tree-based model is affected when adding sentiment variables that can be

[10]The if-else statement works so that if a specified condition is true a block of code is executed. Otherwise, if it is false, another block of code is executed. (W3Schools, 2020).

somewhat correlated. A downside to the decision tree-based models is the low degree of readability and interpreting the explanatory variables' individual effects is harder.

Xgboost is based on the method of *boosted trees,* which applies the concept of *boosting* when building multiple trees. This means that the method tries to improve the model based on the information from the previously constructed trees. A large number of individual trees are combined to create a single consensus prediction yielding a highly improved accuracy, at the cost of some interpretation of the model. Specifically, the residuals from the previously built tree are utilized when constructing the next three, where each tree is built sequentially. This results in decreasing residuals as an additional three is added. By training the model on variance in the dataset that has not yet been explained, the model will improve in areas where it in earlier steps did not perform well. In short, the combination of several weak learners, meaning individual trees, will result in a unified strong learner (James *et al.,* 2013).

The use of *boosted trees* in classification problems was introduced through implementation of a classification algorithm by Friedman *et al.* (2000) in their paper *Additive logistic regression: A statistical view of boosting.* The paper argued that the method used in classification problems can be regarded as a *forward stepwise additive method* where an exponential loss function is minimized. Based on this knowledge, the authors presented a framework called *gradient boosting machines.* Initially, every observation within a tree is assigned a weight w, initialized as $w = \frac{1}{\text{number of observations}}$. Next, the weight w is updated after every iteration. If the model classifies an observation incorrectly, the observation is assigned a greater weight w in the next iteration. On the other hand, if the model classifies an observation correctly, the assigned weight is reduced in the next iteration. This way the observations that are hard to classify are assigned a greater weight and the model estimation process is forced to focus on these observations (Friedman *et al.*, 2008).

Boosted trees are regarded as slow learners, slightly improving the accuracy of the model by adding trees that provide more information about the training data. Important to note is that by adding trees in the model there is a possibility of overfitting, meaning the

model is too closely fitted to the training data and performs worse on the test data. This challenge can be faced to a certain degree by tuning the parameters introduced by the researchers. The parameter d controls how many splits each individual tree can have. An increase in the parameter d could result in overfitting, since interaction effects between the variables could occur. Furthermore, the parameter λ (*shrinkage*) controls the learning speed of the algorithm. The parameter scales each individual tree's contribution to the model. A small value for λ results in slower learning speed. In order to exploit the effects of slow learning one could increase the number of trees that are being built, controlled by the last parameter B. Due to a risk of overfitting when increasing B, it is necessary to adjust B in context of λ (James *et al.*, 2013). A two-class classification problem utilizing *boosted trees* can be formulated as follows:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x) \tag{3.10}$$

where $\hat{f}^{\,b}(x)$ is the estimated probability that a given observation belongs to one of the classes, λ is the shrinkage parameter controlling the learning speed and B is the total amount of trees referred to as the additive functions (James *et al.*, 2013). The final predictions are calculated by including each of the individual trees in the B additive functions and weighting the output from each leaf by weight w.

The *extreme gradient boosting* (*xgboost*) methodology is a further development of *gradient boosting machines.* The method has experienced increased popularity in recent years for several reasons. The additions to the gradient boosting machines are both a regularization expression which is meant to prevent overfitting and a second-degree approximation which aims to increase the performance compared to *gradient boosting machines.* Similar to gradient boosting machines, the mentioned weights w_i are assigned to every observation i. The sum of these weights w is used in a *L2 norm regularization* (least squares method), meant to penalize complex models proportional to the square root of w. The *second-degree approximation* is simplifying the existing objective function presented by Friedman *et al.* (2000), a simplification that both decreases the calculation time and yields better predictions (Chen & Guestrin, 2016).

Xgboost introduces four additional parameters that can be tuned and optimized for each individual problem. The parameter *sub_ sample* is a ratio that decides how much of the provided dataset each tree will use when constructing the tree. This parameter will to an extent prevent overfitting and decrease computational time. The parameter *col_ sample_ bytree* is a ratio deciding how many of the variables that will be used when constructing trees. The subsampling will occur for every individual tree. The parameter *min_ child_ weight* controls the minimum number of instances needed in each node. Lastly, if the trees that are added do not decrease the objective function sufficiently, the parameter *gamma* γ is meant to stop the algorithm.

The method is summarized in the following regularized objective function:

$$\mathcal{L}(\phi) = \sum_{i=1}^{I} l(\hat{y}_i, y_i) + \sum_{b=1}^{B} \Omega(f_b) \tag{3.11}$$

where

$$\Omega\left(f_b\right) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

$\mathcal{L}(\phi)$ is the loss function we seek to minimize. Furthermore, l is the *second-degree approximation,* measuring the difference between the predicted $\hat{y}$ and target y. The other term, $\Omega\left(f_b\right)$, is penalizing the complexity of the model and contains the *L2 norm regularization* in the last clause of the equation. T is the number of leaves in a given tree and is penalized via γ which refers to *gamma*. Furthermore, I is the number of observations in the dataset and B is the number of trees (Chen & Guestrin, 2016).

Furthermore, the *xgboost* model can be configured to handle imbalanced data when performing binary classification. The input parameter *scale_ pos_ weight* aims to train a class-weighted model. This means that the parameter will adjust the weight that is assigned to classification errors on the minority class relative to the majority class during the process of boosting decision trees. By default, the *scale_ pos_ weight* parameter is

set to 1. However, this value can be adjusted based on the training data used for model development. The official *xgboost* documentation suggests that the parameter is set to the inverse of the class distribution. To exemplify, if the training dataset has a 1 to 50 ratio in the minority to majority class, the parameter value can be set to 50. As a result, the classification error made on the minority class will have 50 times more impact on the model during training. In turn, the model will focus 50 times more on correcting the errors on the minority class compared to errors on the majority class (Brownlee, 2020).

The exact parameter values will vary for each problem and they can be optimized for different predefined performance measures. The R-package *mlr* provides a hyperparameter[11] tuning function called *tuneParams()*, that optimizes the different parameters for a predefined performance measure. The function takes arguments such as learner and task where the method and data need to be specified. Furthermore, it needs a resampling technique, a performance measure, the set of parameters to be tuned and a searching method. The default resampling technique is a 5-fold cross-validation, used for estimation and validation on the training data. In terms of performance measures, the function can for example tune the parameters to optimize the *AUC*. Furthermore, all parameters of the *xgboost* function can be tuned. Lastly, the searching method describes how the function is to search through the combinations of the parameters in order to optimize the performance measure. The two most commonly used are grid searches and random searches. The difference between the two is that a randomized search runs through a given number of models, with the number specified by the user, while a grid search runs through all possible combinations of the parameters.

The *xgboost* model can output a variable importance based on a measure called *gain*. *Gain* is the accuracy improvement that a feature brings to the branches it is on. For a given tree, we can add a new split considering feature x on a branch that has some elements classified wrong. Then, there are two new branches and if each of these is more accurate, the *gain* for the feature will increase. The measure itself is relative between the variables for a given model. This means that it is useful for displaying which features contribute the

[11] A hyperparameter is within statistics a parameter that is from a prior distribution, capturing the prior belief before one is able to observe the data (Prabhu, 2018).

most and which the least, but the numeric value itself has little interpretation. It is also useful for comparison between the ranking of features between models but comparing the numeric values does not make sense. The variable importance is useful when investigating to which degree the variables are contributing to correctly classify companies as bankrupt or not. It is also supposed to be valid although variables are perfectly correlated. In theory, the method of boosted trees will try not to focus on a specific link between the outcome and a feature after the link has been learned by the algorithm (Benesty *et al.*, 2018).

4 Data

4.1 Data sources and preprocessing

4.1.1 Financial data

The first dataset we cover is the *annual accounts* dataset. The dataset consists of accounting data from the year 1992 to and including 2019, for all companies registered in the Brønnøysund Register Centre. All financial information that is registered annually for each company is found in this dataset. The total dataset consists of 4 596 053 observations and 148 features, where the features mostly consist of standard financial figures such as total assets or total sales.

As we are looking into private limited companies in the analysis, we exclude observations which are not registered as private limited companies. This is due to their annual accounts being publicly available and easily accessible. Furthermore, we only keep observations with at least NOK 50 000 in revenue. The reason for this delimitation is that we want to avoid holding companies in our analysis, as well as small companies barely operating. We are extracting the accounting years from year 2014 to 2017. This time period is chosen because the bankruptcy ratio is steady throughout this period, and all our datasets contain sufficient data for this given period. Also due to limited computational power we need to limit the number of observations. Furthermore, we are only including observations that are registered in currency code NOK, which is the vast majority of all observations. As a result of this delimitation, we can disregard the fluctuating exchange rates throughout the time period. We find this delimitation appropriate as the observations registered in other currencies are very few.

The *company details* dataset consists of every registered Norwegian company and their belonging characteristics. This data is also gathered from the Brønnøysund Register Centre. The dataset consists of in total 1 706 858 observations and 48 features. For our analysis, the features of interest in the dataset are the organizational number used for identifying unique companies and industry codes outlining what sector the company

operates in. The industry codes are used to delimit the industries regarded in the textual data analysis. We observe some noise when matching news articles to companies in certain industries. Due to shortcomings of the matching method, some newspaper providers are matched with news articles they have published themselves, although the article is not necessarily focused on the newspaper company itself. To avoid this noise in our dataset, we exclude all companies registered in industries with codes ranging from 58000 to and including 64000. The excluded industries are related to telecom, IT and media[12].

The *company announcements* dataset consists of published events related to companies that are registered in the Brønnøysund Register Centre. The initial dataset contains 11 486 711 observations of 8 features. One of the event categories is a bankruptcy announcement. The announcement contains information on title, summary and content of the event, in addition to the date of publication. In this paper the bankruptcy announcements in the dataset, with their belonging date of publication, are used for deriving the bankruptcy-trigger in terms of a binary variable. We observe that the registered event of bankruptcy in Brønnøysund Register Centre usually is announced at the same time or somewhat earlier than the date registered in the Register of Legal Entities, another governmental source of registered bankruptcies. Thus, these dates are usually highly connected. When we only include bankruptcy observations, there are a total of 25 195 bankruptcy observations in the dataset from the period 2001 to 2020, where we include two variables: organizational number and the date of publication for the bankruptcy filing.

4.1.2 News data

The *news* dataset consists of Norwegian news articles published on the internet during the time period 2008 to 2020. The source of news articles ranges from local newspapers to nationwide economic newspapers. The news articles have been scraped from the internet by the company Infomedia and further distributed by Enin AS. Infomedia offers a broad and deep media monitoring service, across various media and behind payment walls (Infomedia, 2020). The total dataset consists of more than 58 000 000 observations and 7 features. The features of interest are the news title, summary and content. In addition, the exact date of publication will be used to assign the articles to the belonging annual accounts data and

[12]See table A1.1 in appendix for an exhaustive list of industry codes

time period. Furthermore, we perform one initial delimitation. From the total dataset we only include articles from year 2014 to and including 2018. By including 2018 we are able to utilize news data published the months prior to publication of 2017 annual accounts in 2018. We also remove extreme outliers for news title and summary, which we regard as observations containing more characters than 100 and 500 respectively. Furthermore, in order to save computational power, we remove content observations with more than 10 000 characters, which is about 4.7 percent of all observations. After extracting the years of interest, the dataset contains approximately seven million news observations.

By default, the news articles used in this thesis are not assigned to companies. In order to assign the sentiment scores and values to the correct company we need to match every news article to their respective company. The matching algorithm is divided into two steps.

The first step involves using a replacement algorithm that retrieves potential candidates from the news articles. These candidates can for example be versions of company names where "&" is replaced by "og" or versions where "AS" is removed. In addition, candidates consisting of regular words are discarded. This way, we for example avoid matching Rør AS with "rør" in the news article. The regular words are in our paper regarded as words that occur at a relatively high frequency of 0.5% or above in all the news articles, meaning they are not providing any unique information or referring to a specific company. By generating a corpus consisting of all words from all news articles, these high frequency words can be filtered out from the extracted candidates. The output in the end is a list where each article has a number of candidates ranging from zero to multiple candidates. News articles without probable matches are then removed from the dataset.

In step two, the *FlashText*-algorithm is being used to find the most probable match in each article. If there are several possible candidates, the algorithm will choose the candidate with the most characters. The mentioned matching method is optimized on large scale data (Huse, P.I., personal communication, 22.10.2020).

4.1.3 Lexicons

We investigate the lexicon-based sentiment analysis approach. Thus, there is a need for datasets containing appropriate lexicons. We will present two different dictionaries which we use separately and yield two different ratios in the sentiment analysis. Neither of the dictionaries are domain-specific, but more general-purpose dictionaries. One reason for using these general-purpose dictionaries is that the news sources are diverse and not focused on one particular field, such as economics or sports. In addition, the general-purpose dictionaries provide large amounts of words. Furthermore, the two dictionaries are chosen for their complementary properties with regards to their way of classifying the different words.

The *Jockers & Rinker* dictionary derived by the researchers Matthew L. Jockers and Tyler Rinker (2019) consists of 11 710 words. These words are regarded as general-purpose words and each word is assigned its individual polarity score. As a result of this, the dictionary not only distinguishes negative and positive words. It also provides a polarity score that reflects the degree of negativity or positivity. This score ranges from -1 to 1; -1 corresponding to very negative, 1 corresponding to very positive. Furthermore, the dictionary is the default lexicon of the *sentiment*-function provided in the *sentimentr* package in R. The dictionary has been translated from English to Norwegian using Google's Translate API. This API is accessed through the package *translateR*, where we utilize the function *translate()*. As a result of the translation, the total number of words decreases from over 11 710 to 6612, as some of the translations create duplicate words in the lexicon. For example, both "happy" and "glad" translate to the Norwegian word "glad". If there are duplicates when we format the data as a lexicon, the first observation will be the one that is not removed.

The *UiO* dictionary is a Norwegian dictionary that consists of 6103 positive and 14839 negative words. This dictionary is also regarded as a general-purpose dictionary, and it is based on the work of M. Hu and B. Liu (2004). All words are translated to Norwegian and manually inspected and corrected by Barnes *et al.* (2019). Unlike the Jockers & Rinker´s dictionary, this dictionary only divides the words into one of the two classes:

negative or positive. We assign a polarity value of 1 to the positive words and -1 to the negative words.

Furthermore, we are using valence shifters. These are found in a dataset originally consisting of in total 140 valence shifters. Each valence shifter is categorized as either a (1) negator, (2) amplifier, (3) de-amplifier or (4) adversative conjunctions. The valence shifters are derived by Rinker (2019) and provided in the R-package *lexicon*. All valence shifter words are translated from English to Norwegian for the purpose of this thesis. In addition, we remove all words from the valence shifters data table that also occur in the dictionary of choice as this is necessary for the function to operate. This results in 86 and 73 valence shifters for the *Jockers & Rinker* dictionary and *UiO* dictionary respectively.

4.2 Developing the final dataset

4.2.1 Explanatory variables

Financial ratios - Altman's five ratios

In total we derive five financial ratios from the *annual accounts* dataset. These five ratios are consistent with the ratios in the previously presented Altman's Z-score model. The reason for choosing these variables is the easy interpretation and the empirically proven high predictive power on shorter time horizons, one and two years. In addition, a majority of the observations in the *annual accounts* dataset includes the information necessary for deriving these five ratios. Thus, the percentage of missing data is very low for the financial ratios, most ratios averaging well under 0.1 percent. The five financial ratios form our reference models, which we use for comparison to extended models including sentiment variables. After delimiting the annual accounts data and deriving the financial ratios, we extract the financial ratios, the organizational numbers, the accounting year and the announcement date for the accounts. This results in a dataset, we hereby refer to as *financial data*, consisting of 886 739 observations of 8 variables. Table 4.1 displays the variables in the *financial data* dataset.

Table 4.1: The financial data dataset

Variable name	Description	Variable type
org_nr	Organizational number	Identifier
accounting_year	Accounting year	Timestamp
accounting_announcement_date	Announcement date of annual accounts	Timestamp
x1	(total_current_assets - total_current_debt)/ total_assets	Financial ratio
x2	retained_earnings/total_assets	Financial ratio
x3	operating_profit/total_assets	Financial ratio
x4	total_equity/total_liabilities	Financial ratio
x5	total_sales/total_assets	Financial ratio

Sentiment variables

All sentiment variables are calculated using the *sentiment()*-function provided in the *sentimentr* package in R. The presented valence shifters are also included in the derivation of the scores and ratios. The four different methods we apply are:

1. Sentiment scores using the Norwegian translated *Jockers & Rinker* dictionary.
2. Sentiment scores using the *UiO* dictionary.
3. Percentage positive words based on the *UiO* dictionary.
4. Percentage negative words based on the *UiO* dictionary

The sentiment variables are calculated individually based on news title, summary and content from the *news* dataset. By including the different variations, we hope to capture all available information. In total, the methods applied to the three news-related columns result in 12 variables per news article. After deriving the sentiment variables per news article, we extract the 12 sentiment variables, their belonging organizational numbers and a time dimension displaying month and year into a separate dataset, hereby referred to as *news averaged*. From here, we average the sentiment scores for each company on a monthly basis for each year. This means that over the course of our time period from 2014 to 2018 for news data, the maximum number of observations a company can have is 60[13]. The *news averaged* dataset contains 877 978 observations of 14 variables and is described in table 4.2.

[13] 5 years of 12 months each, $5 * 12 = 60$.

Table 4.2: The news averaged dataset

Variable name	Description	Variable type
org_nr	Organizational number	Identifier
yearmonth	Year and month	Timestamp
title_sentiment_JR	Jockers & Rinker sentiment	Score
summary_sentiment_JR	Jockers & Rinker sentiment	Score
content_sentiment_JR	Jockers & Rinker sentiment	Score
title_sentiment_UiO	UiO sentiment	Score
summary_sentiment_UiO	UiO sentiment	Score
content_sentiment_UiO	UiO sentiment	Score
title_positivity	UiO - positive word percent	Ratio
summary_positivity	UiO - positive word percent	Ratio
content_positivity	UiO - positive word percent	Ratio
title_negativity	UiO - negative word percent	Ratio
summary_negativity	UiO - negative word percent	Ratio
content_negativity	UiO - negative word percent	Ratio

4.2.2 Response variables

The lack of a generally accepted definition of bankruptcy and existence of different interpretations of the concept among researchers (Bellovary *et al.*, 2007) has led to various different non-unified models deriving the response variable in different ways. Thus, it is essential to clarify how we derive the response variable, which for two-class classification problems is a binary variable. We are using the announcement of bankruptcy-date from Brønnøysund Register Centre as the trigger for bankruptcy.

Generally, the prediction horizon of choice in a bankruptcy model is dependent on the model's purpose. The developed models in this thesis predict bankruptcy both on a 12-month and a 24-month horizon. There are several reasons for this choice. First of all, we observe from the literature that shorter prediction horizons in general yield the best predictive performance. Furthermore, a credible assumption when including news data is that this information is more valuable in the short term. The use of a 12-month horizon is an often prefered time horizon among actors in the market for bankruptcy models and thus an often-used approach, especially as financial accounts are published annually. However, we observe both in the literature and in our dataset that the majority of the registered bankruptcies is registered two years after the last publication of account details.

The trend in Norway in recent years has been that about 30 percent of bankruptcies are registered in the first year after the last approved accounts, while about 85 percent of bankruptcies accumulated are registered within two years after the last approved accounts (Hjelseth & Raknerud, 2016). This means that of the total bankruptcies filed within two years, about 35 percent[14] are filed within the first year. Although we do not have a total number of bankruptcies filed within all different time horizons for our dataset, we observe that about 30 percent of the bankruptcies filed within two years are filed within the first year. If we assume that we cover 85 percent of the total bankruptcies for our two-year prediction horizon, this corresponds to 25.5 percent of all bankruptcies being filed within the first year.

Given our assumption that news data is more valuable in the short term, a 12 months prediction horizon can appear appropriate in order to detect predictive power of the news data. However, such an approach will lead to a majority of the bankruptcies being omitted. In addition, if a given account observation contains both account and news information indicating a potential bankruptcy, but an official bankruptcy filing is registered after the 12-month horizon, this observation would be assigned a non-bankrupt response variable value of zero. By defining such observations as non-bankrupt, it is likely that we will add some noise to the majority class of non-bankruptcy companies resulting in a dataset where it is harder to distinguish the two classes.

To tackle this potential problem, we are preparing two separate binary bankruptcy response variables, one for each prediction horizon. The first binary response variable we derive represents bankruptcy within 12 months. If a company files for bankruptcy within 12 months after their annual accounts data are published, the bankruptcy variable is set to 1 in the belonging accounting year. Otherwise, it is set to 0. The second binary response variable represents bankruptcy within 24 months. If a company files for bankruptcy within 24 months after their annual accounts data are published, the variable is set to 1. Otherwise, it is set 0. The advantage of predicting from the date of announcement is that we can evaluate the annual accounts once they are announced and available in the Brønnøysund Register Centre.

[14] $0.30 \div 0.85 \approx 0.35$

The response variables are derived by using the *company announcements* dataset in combination with the *financial data* dataset. Here, we utilize *for*-loops[15] and *if*-statements[16]. Two separate loops are created for each of the response variables. These are created so that for each company belonging to an observation in *financial data*, if the company has a registered bankruptcy filing in the *company announcements* data between the published accounts and the loop's respective prediction horizon, the response variable is set to 1. If not, it is set to 0. The two additional variables are this way added to the *financial data* dataset, so that the total number of variables go from 8 to 10.

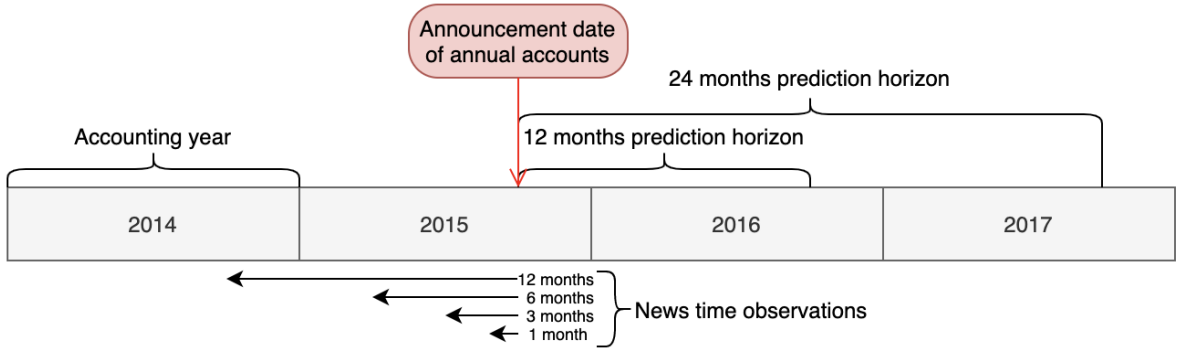
4.2.3 Combining textual and financial data

Next, we start the process of merging our *financial data* dataset including the two response variables with our *news averaged* dataset. Here, we introduce four news observation periods we want to investigate. These are 1 month, 3 months, 6 months and 12 months prior to the annual account announcement date for each company, where the sentiment scores are averaged for each of the described periods. This way, we obtain a total of 48 explanatory sentiment variables. These are derived through four nested *for*-loops in combination with *if* -statements R, one for each news observation period. In short, the two functions are applied so that for each *financial data* observation the loop checks whether the news observations from *news averaged,* belonging to the company the *financial data* observation regards, is within the news observation period. The news observations from *news averaged* that are within the news observation period are then added to a dataset, where they are averaged for the period and then added to the *financial data* dataset. The resulting dataset is called *final data* and consists of 372 540 observations of 55 variables. Figure 4.1 illustrates how a given *final data* observation is handled with regards to prediction horizons and inclusion of sentiment variables.

[15] A for-loop is a control flow statement used for iterating over an object and allows for code to be repeatedly executed (Datamentor, 2020a).

[16] An if-statement is a control flow statement that is executed based on given conditions (Datamentor, 2020b)

Figure 4.1: Overview of a how a given observation is handled

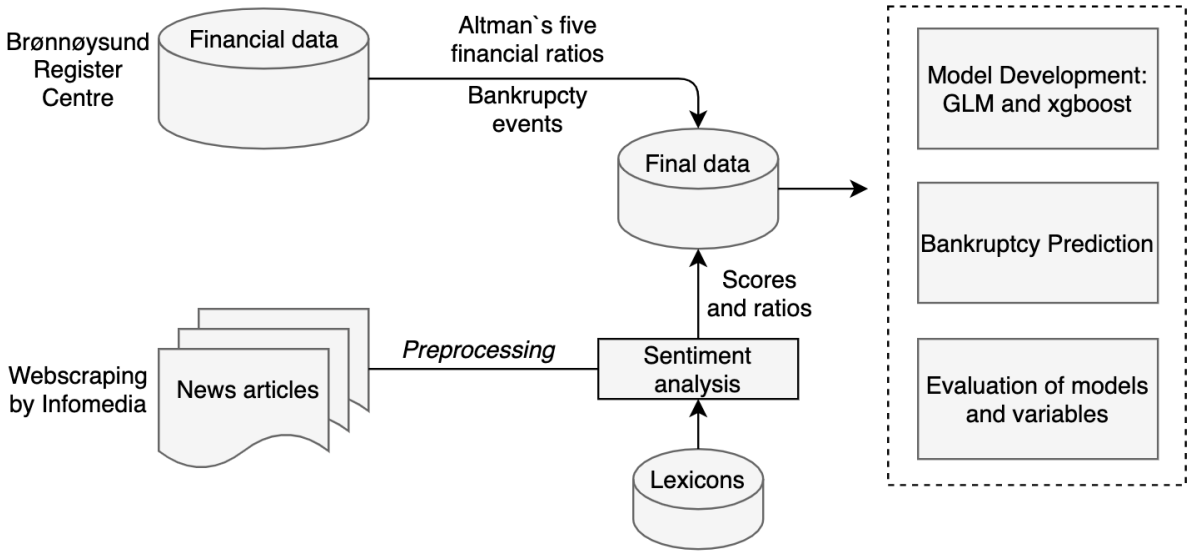


For the given observation in figure 4.1, the annual accounts from accounting year 2014 are announced in the fall of 2015. For this observation we assign sentiment variables for all of the four news observation periods separately. In addition, two separate response variables will be assigned, one for each of the two prediction horizons.

One important aspect to note is that the bankruptcy-ratio falls drastically when merging the financial data with the news data. We go from 886 739 observations in our *financial data* to 374 540 after merging with the *news averaged* dataset, meaning that about 42 percent of the companies in *financial data* are mentioned in the news. The number of bankruptcies decreases from a total of 8 865 out of 886 739 to 481 out of 374 540 for a 1-year horizon, and 17 517 out of 886 739 to 1610 out of 374 540 for a 2-year horizon. This corresponds to a change from about 1 percent and 2 percent in *financial data* to 0.13 percent and 0.43 percent for *final data* for 1 and 2 years respectively. On average, the bankruptcy ratio drops by more than 80 percent when we merge *financial data* with the *news averaged* dataset. The obvious implication here is that amongst companies that are mentioned in the news, the rate of bankruptcy is substantially lower than companies not mentioned. This substantiates the argument for applying balancing mechanisms when developing models, as the *final data* dataset can be seen as highly imbalanced.

A summary of our methodology process is displayed in figure 4.2.

Figure 4.2: Methodology process



The illustration is inspired by Jo and Shin (2016)

4.3 Descriptive analysis of final datasets

4.3.1 Final data subsets

Before modelling, we divide *final data* into several subsets considering the different time horizons of news observations, and different prediction horizons. For example, when considering the 1-month news horizon only about 50 000 observations have registered news for this period. Therefore, we remove the remaining observations that do not have news 1 month prior to publishing their financial accounts. In this manner, we go from 372 540 observations in *final data* to 50 801 observations for this particular subset when we consider this news horizon. This way, all observations in a given subset will have belonging sentiment variables. By only including the observations with sentiment variables, we can better compare the developed models to a reference model based on the same subset data, but *without* sentiment variables. In total, this will result in 8 different subsets. In table 1 we summarize the different subsets used for modelling, including the total amount of observations and percentage of bankruptcy observations.

Table 4.3: Summary of all final data subsets

News period	Prediction horizon	Total observations	Bankruptcy ratio
1 month	12 months	50801	0.001752
3 months	12 months	90110	0.001565
6 months	12 months	126010	0.001444
12 months	12 months	168287	0.001408
1 month	24 months	50801	0.005374
3 months	24 months	90110	0.005060
6 months	24 months	126010	0.005142
12 months	24 months	168287	0.005289

From table 4.3 we observe as expected that the percentage of bankruptcy observations increase when using the 24-month prediction horizon. Although there is some variation in the exact bankruptcy ratios depending on the news observation period and prediction horizon, all subsets are highly imbalanced with regards to bankruptcy and non-bankruptcy observations. Lastly, table 4.4 displays all explanatory variables with belonging descriptions that we are using in the analysis.

Table 4.4: Description of variable names

Variable name	Description
x1	Working capital ÷ Total assets
x2	Retained earnings ÷ Total assets
x3	EBIT ÷ Total assets
x4	Total equity ÷ Total liabilities
x5	Total sales ÷ Total assets
Title UiO	Sentiment score of title using the UiO dictionary
Summary UiO	Sentiment score of summary using the UiO dictionary
Content UiO	Sentiment score of content using the UiO dictionary
Title JR	Sentiment score of title using Jockers & Rinker's dictionary
Summary JR	Sentiment score of summary using Jockers & Rinker's dictionary
Content JR	Sentiment score of content using Jockers & Rinker's dictionary
Title pos.	Percentage positive words in title
Summary pos.	Percentage positive words in summary
Content pos.	Percentage positive words in content
Title neg.	Percentage negative words in title
Summary neg.	Percentage negative words in summary
Content neg.	Percentage negative words in content

4.3.2 Data quality and other considerations

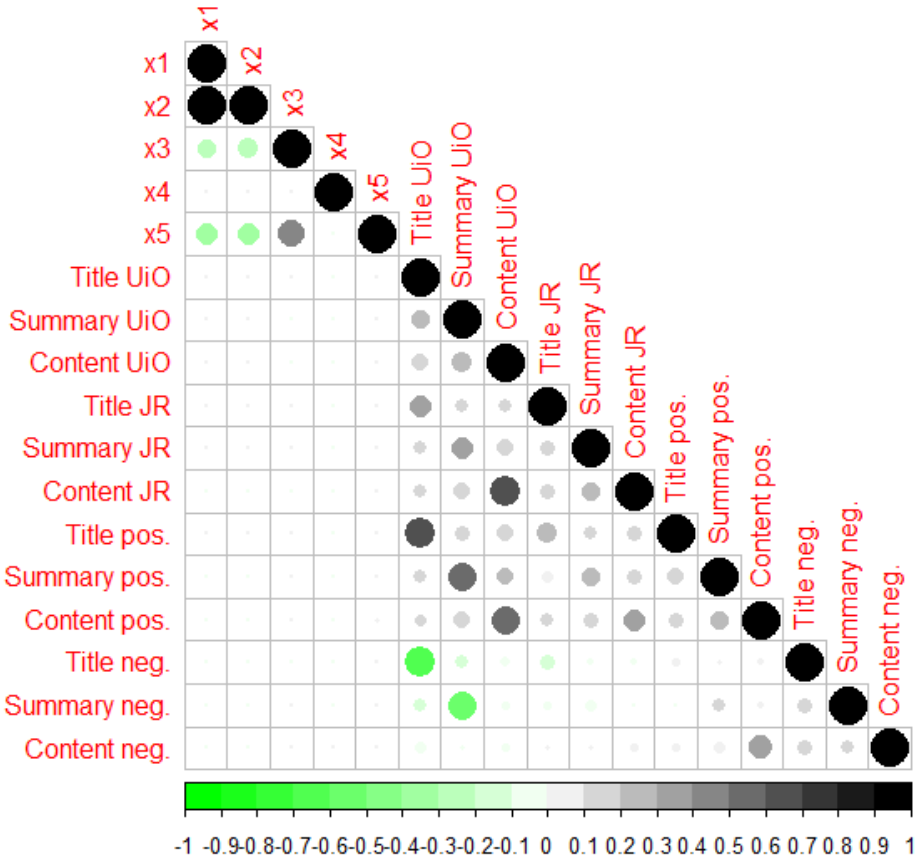
We observe some missing data in the sentiment variables, the reason being that some of the scraped news articles do not contain input for all three features: title, content and summary. Since these observations only make up a very small percentage, we keep all the observations.

Furthermore, we observe an increase in bankruptcy observations in 2017 compared to earlier years. The two probable reasons are both the way we derive response variables and the increased amount of *news article* data in 2018 leading to more observations of bankruptcy companies with mentions in the news. This is problematic if we were to perform validating out of time, meaning validating our models on observations from accounting year 2017 exclusively. For this reason, we disregard the year aspect when dividing our data into a train and test set and perform regular randomized out of sample validation.

Worth noting is that the matching method is prone to error. The method is optimized on a large scale, but especially since the minority class of bankruptcies is very small, errors in the matching of bankruptcy observations could potentially add decisive noise to the models.

In general, multicollinearity can be a challenge when trying to interpret how the individual explanatory variables affect the response variable. In addition, the coefficient estimates in a GLM are sensitive when there is multicollinearity among the explanatory variables. Multicollinearity is present when there is collinearity between three or more explanatory variables (James *et al.*, 2013). To investigate the correlations in our data we plot a correlation matrix based on the *final data* dataset. In figure 4.3, the correlation matrix for one of the *final data* subsets is presented.

Figure 4.3: Correlation matrix



See table 4.4 for variable descriptions.

From figure 4.3, we observe that x1 and x2 are highly correlated, which is somewhat expected based on how the ratios are derived. Since we are not particularly interested in the accurate coefficients of either x1 or x2, we keep both variables in our datasets. Furthermore, several of the sentiment variables are somewhat correlated. For example, we observe moderate correlation between the variables Content UiO and Content JR. Since these two variables only differ with regards to the dictionary being used when calculating the scores, this is expected. In general, we observe that the two different dictionaries are providing somewhat similar information due to the existing correlation.

The *xgboost* method handles multicollinearity well. We are aiming at investigating the predictive power of the sentiment variables compared to a reference model excluding sentiment variables. The goal is not to achieve the best performing models with low

levels of multicollinearity, but to put emphasis on the comparison. Multicollinearity is furthermore not influencing the overall predictive power of our models (Frost, 2020). In addition, the levels of correlation between the sentiment variables are mostly considered to be moderate, with the highest correlation being 0.64[17]. For these reasons, we do not remove any of our explanatory variables before developing models, although multicollinearity is present in our dataset.

[17]See table A2.1 in appendix for the full correlation matrix.

5 Model development

Randomization and reproduction

In order to be able to reproduce the obtained results and compare the different proposed models, we utilize the *set.seed()*-function in R and set the seed-value equal to 1. This ensures the splits performed during 5-fold cross-validation and train-test are equal every time the code runs. The randomly chosen majority class observations during *SMOTE* are also consistent. Furthermore, when the random number stream set by the seed is being held constant, the model training process will be consistent from run to run.

Estimation and validation approach

The initial train-test split is 0.75-0.25, meaning 75% of the final dataset is used for estimating the models and 25% of the final dataset is used for the final validation. Furthermore, when performing cross-validation we set the k value equal to 5, resulting in 5 folds. The number of folds is chosen based on empirically research pointing out five folds as sufficient in order to achieve an acceptable bias-variance trade-off. The average of all the five folds' predictions on the hold-out test data, form our final prediction. The final prediction is used for evaluation and comparison.

Handling missing data

In order to estimate a GLM we need to handle the missing data in the dataset, referred to in R as *NA* for "*Not Available*" . There are missing data both in the financial ratios and in the sentiment variables. We choose to set all *NA*s equal to 0, which is a quick and easy fix for handling missing data. When imputing zero-values bias might occur, which is one of the downsides of using models such as logistic regressions that need missing values to be handled in order to work. For our *xgboost* models we do not handle missing data in any way, as the method handles it on its own.

Hyperparameter tuning

We conduct two hyperparameter tunings, one for *xgboost* models containing Altman's

ratios and one for models Altman's ratios with sentiment variables. Here, the optimization algorithm is applied to a *final data* subset with a 12-month news horizon, and we assume the optimized parameters would be the same regardless of news horizon. In order to perform the tuning in R we utilize the function *tuneParams()* provided in the package *mlr*. We optimize with regards to AUC. All hyperparameters in the *xgboost* function are tuned. Due to the huge computational requirement of such a tuning, we do a randomized search instead of a grid search. Furthermore, the default estimation and validation approach of 5-fold cross-validation is applied. The results of the hyperparameter tuning are displayed in table 5.1 for both *xgboost* models.

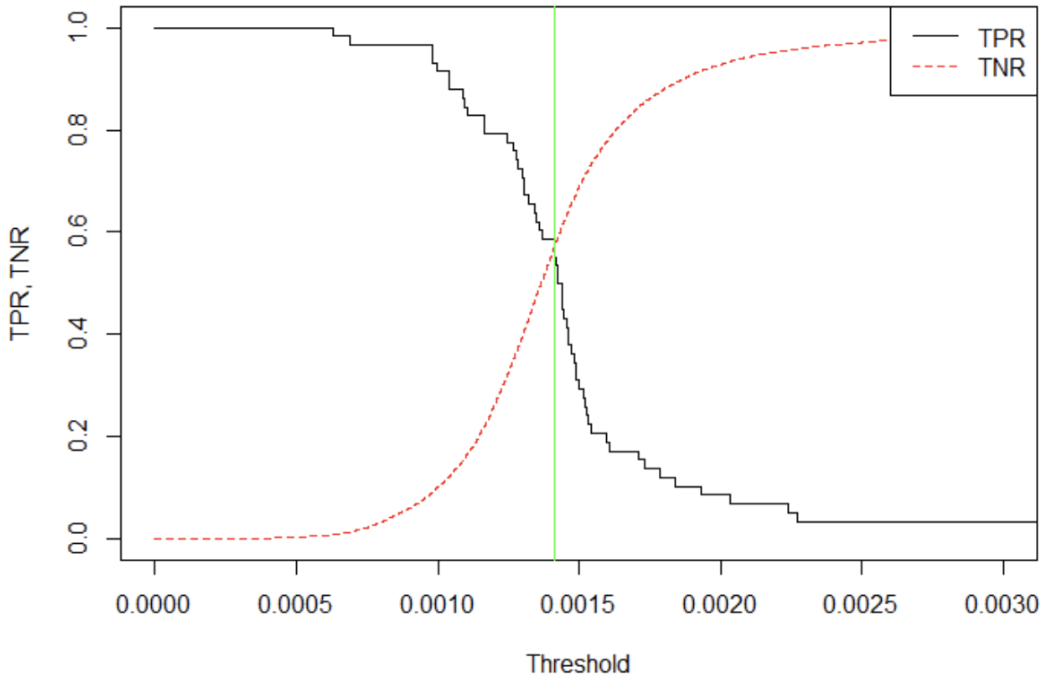
Table 5.1: Optimal parameters for xgboost

	Sentiment	**Reference**
gamma	0.386	0.429
max_depth	8	3
eta	0.0181	0.0524
min_child_weight	8.96	3.15
subsample	0.67	0.978
colsample_bytree	0.744	0.963
nrounds	389	204

Optimizing confusion matrix thresholds

When creating confusion matrices, we set the threshold such that $TPR = TNR$ for each model. This lays a good foundation for comparison and allows us to prioritize the rates equally, although it can easily be argued that prioritizing a good TPR is more important when predicting bankruptcies as misclassifying a bankrupt company as not bankrupt can be costly. All datasets being used in the modelling are highly imbalanced, which means the thresholds are likely to be extremely low in order to obtain a sufficient balanced accuracy. By using the same approach for each model, we are able to compare the balanced accuracy. Figure 5.1 displays where the TPR and TNR are equal for one of our models.

Figure 5.1: Optimized threshold for a GLM with 12-month news horizon



6 Results

6.1 Without rebalancing

6.1.1 GLM - 12-month prediction horizon

Table 6.1: Performance measures - GLM 12-month prediction horizon

	Sentiment			Reference			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.5043	0.5280	0.4606-0.5948	0.5525	0.5660	0.4844-0.6486	0.4342
6 month	0.5135	0.4880	0.3973-0.5792	0.6271	0.6570	0.5561-0.7586	* 0.0120
3 month	0.5298	0.5480	0.4520-0.6442	0.6322	0.6900	0.5980-0.7828	* 0.0216
1 month	0.5834	0.6230	0.5215-0.7248	0.5959	0.6140	0.4831-0.7440	0.9059

The column DeLong provides a p-value obtained when performing a DeLong test comparing the two ROC curves of the Sentiment model and the Reference model. The alternative hypothesis is that the true difference is not equal to 0. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

In table 6.1 we present the results for the GLM using a 12-month prediction horizon. Both when including sentiment variables 6 months prior and 3 months prior to annual accounts announcement date, we observe that the reference model without sentiment variables is performing significantly better. The p-values from DeLong´s test are 0.0120 and 0.0216 respectively. When including sentiment variables 12 months and 1 month prior to the announcement date, we observe that the two models perform more similar. The AUC on the sentiment model with a 1-month news horizon yields a slightly higher AUC, compared to the reference model. However, the difference is not significant, indicated by the high p-value and large confidence intervals for the AUC. Furthermore, we observe that the balanced accuracies are higher for all reference models compared to the sentiment models for all news horizons.

Table 6.2: Average marginal effects - 12-month prediction horizon

Variables/	Sentiment				Reference			
News horizon	12-month	6-month	3-month	1-month	12-month	6-month	3-month	1-month
x1	0.0000	0.0000	0.0000	-0.0001	0.0000	0.0000	0.0000	0.0001
x2	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0001
x3	0.0000	0.0000	0.0000	-0.0001	0.0000	0.0000	0.0000	-0.0001
x4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Title UiO	0.0004	0.0013	-0.0001	0.0007				
Summary UiO	-0.0007	-0.0009	-0.0005	-0.0016				
Content UiO	0.0003	-0.0004	-0.0008	-0.0001				
Title JR	-0.0024	-0.0018	-0.0002	* -0.0031				
Summary JR	0.0005	-0.0016	-0.0022	* -0.0042				
Content JR	0.0005	0.0017	0.0017	0.0003				
Title pos.	-0.0006	-0.0017	0.0004	-0.0016				
Summary pos.	0.0025	0.0027	0.0047	0.0066				
Content pos.	-0.0066	0.0005	-0.0123	0.0003				
Title neg.	0.0023	0.0039	0.0017	0.0021				
Summary neg.	-0.0014	-0.0021	-0.0001	-0.0027				
Content neg.	** 0.0114	* 0.0104	0.0097	0.0012				

Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05. See table 4.4 for variable descriptions.

In table 6.2 we present the Average Marginal Effects (AME) for all explanatory variables in the different GLMs along with significance codes for levels of significance. From the table we observe that there are few significant *AME* values. Content negativity has significant *AME* values of 0.0114 and 0.0104 for 12 and 6 month news horizons respectively. The positive signs indicate that the probability of bankruptcy increases when content negativity increases, which makes sense. Furthermore, both Title JR and Summary JR have significant *AME* values in the 1-month news horizon sentiment models. The negative signs appear intuitive as we expect the probability of bankruptcy to drop when increasing the polarity values.

6.1.2 Xgboost - 12-month prediction horizon

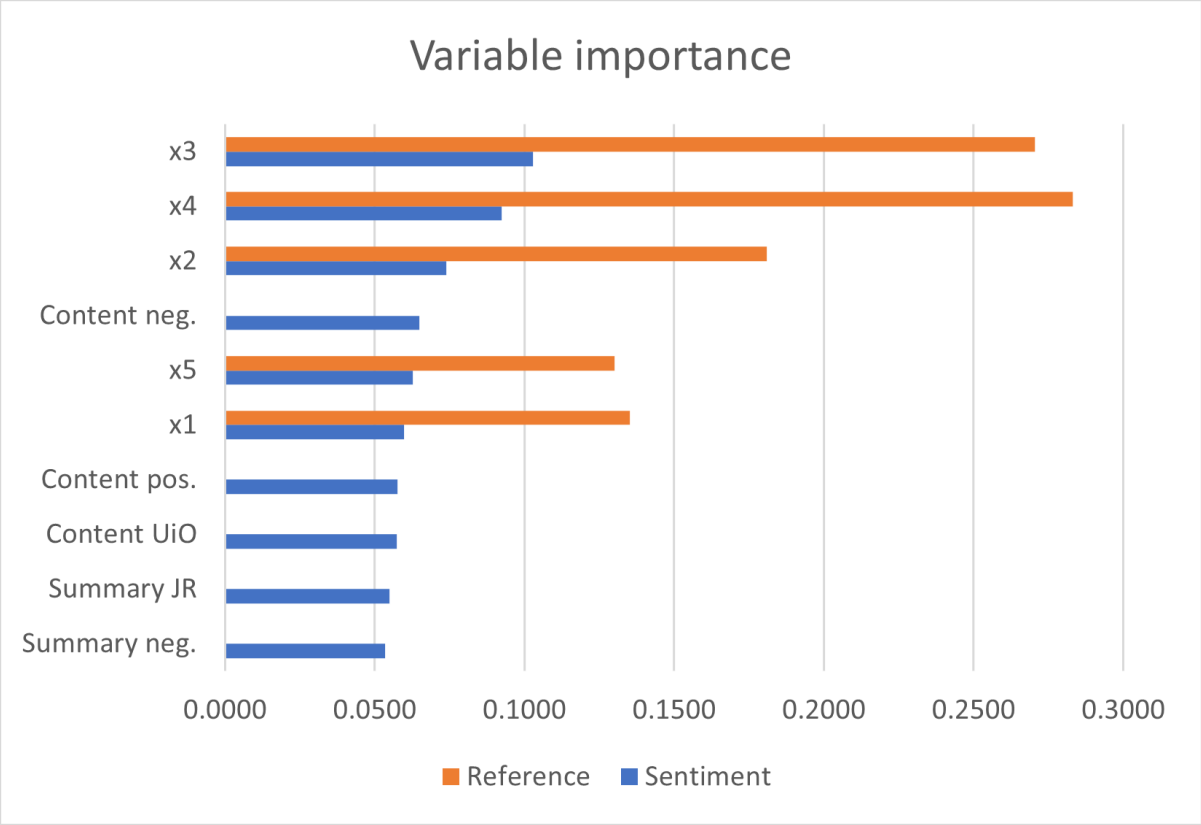
Table 6.3: Performance measures - xgboost 12-month prediction horizon

	Sentiment			Reference			
News	*Bal. Acc.*	*AUC*	*95% CI*	*Bal. Acc.*	*AUC*	*95% CI*	*DeLong*
12 month	0.7333	0.7830	0.7276-0.8498	0.7333	0.7930	0.7390-0.8570	0.4160
6 month	0.7418	0.8000	0.7386-0.8775	0.7297	0.8050	0.7344-0.8846	0.9333
3 month	0.7349	0.7420	0.6576-0.8408	0.7017	0.7450	0.6680-0.8303	0.9968
1 month	0.7574	0.8360	0.7784-0.9110	0.7264	0.8300	0.7776-0.8984	0.8011

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

In table 6.3 we present the results for the *xgboost* model using a 12-month prediction horizon. An initial observation is that all *xgboost* models perform better in terms of balanced accuracy and AUC, compared to the comparable GLMs in table 6.1. Furthermore, in table 6.3 we observe that the AUC in the sentiment models and reference models are very similar for all news horizon observations. None of the sentiment models are performing significantly better than the reference model, as seen in the DeLong p-values. However, we observe that the AUC of the 1-month news horizon sentiment model is slightly higher than the AUC of the reference model, with values equal to 0.836 and 0.830 respectively. Furthermore, also the balanced accuracies are slightly higher in the sentiment models for the news horizons 1, 3, and 6 months.

Figure 6.1: Average variable importance for xgboost, 12-month prediction horizon



Top 10 performing variables, measured in gain. See table 4.4 for variable descriptions.

In figure 6.1 we illustrate the averaged variable importance in terms of gain obtained from the *xgboost* models. The blue bars represent the averaged gain values for the variables

in all sentiment models, while the orange bars represent the averaged gain values for the financial ratios in the reference models. We observe that the financial ratios, x1-x5, are all assigned relatively high variable importance values, indicating that the financial ratios overall are contributing the most in predicting bankruptcy given these model configurations. Furthermore, we observe that content negativity has an assigned average importance higher than both x5 and x1 in the sentiment models.

6.1.3 GLM - 24-month prediction horizon

Table 6.4: Performance measures - GLM 24-month prediction horizon

	Sentiment			Reference			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.5342	0.548	0.5130-0.5840	0.5742	0.617	0.5728-0.6603	* 0.0148
6 month	0.5063	0.519	0.4777-0.5601	0.6460	0.725	0.6899-0.7598	*** 0.0000
3 month	0.5611	0.571	0.5188-0.6224	0.6442	0.699	0.6475-0.7507	*** 0.0002
1 month	0.5313	0.555	0.4869-0.6230	0.6101	0.667	0.5920-0.7424	* 0.0335

See table 6.1 for DeLong explanations. Signif. codes: 0 '∗ ∗ ∗ ' 0.001 '∗ ∗ ' 0.01 '∗ ' 0.05.

In table 6.4 we present the results for the GLM using a 24-month prediction horizon. We observe that the reference models significantly outperform the sentiment models in terms of AUC for all news horizons. Also, the balanced accuracies are higher for all reference models.

Table 6.5: Average marginal effects - 24-month prediction horizon

Variables/	**Sentiment**				**Reference**			
News horizon	**12-month**	**6-month**	**3-month**	**1-month**	**12-month**	**6-month**	**3-month**	**1-month**
x1	-0.0001	0.0000	0.0000	-0.0001	-0.0001	0.0000	0.0000	-0.0001
x2	0.0001	0.0000	0.0000	0.0001	0.0001	0.0000	0.0000	0.0001
x3	-0.0001	0.0000	0.0000	-0.0001	-0.0001	0.0000	0.0000	-0.0001
x4	0.0000	0.0000	0.0000	0.0000	* -0.0000	0.0000	0.0000	0.0000
x5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Title UiO	-0.0015	0.0008	0.0014	0.0010				
Summary UiO	-0.0010	-0.0007	-0.0022	-0.0017				
Content UiO	0.0010	0.0001	0.0000	0.0009				
Title JR	** -0.0050	-0.0031	-0.0016	-0.0047				
Summary JR	-0.0010	-0.0007	0.0005	-0.0025				
Content JR	* 0.0031	* 0.0031	0.0010	-0.0003				
Title pos.	0.0043	0.0018	-0.0020	0.0004				
Summary pos.	0.0053	0.0077	0.0088	0.0117				
Content pos.	-0.0052	-0.0018	0.0138	0.0140				
Title neg.	0.0003	0.0056	0.0070	0.0054				
Summary neg.	0.0047	0.0053	-0.0050	0.0069				
Content neg.	* 0.0282	** 0.0315	0.0199	-0.0155				

Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05. See table 4.4 for variable descriptions.

In table 6.5 we present the *Average Marginal Effects (AME)* obtained from the GLMs. An initial observation is that there are few significant *AME* values. Similar to the GLMs that predict bankruptcy on a 12-month horizon, we observe that content negativity variable has significant *AME* values for both 12- and 6-month news horizons. The positive sign on both *AME* values indicates that the probability of bankruptcy increases when increasing the value of the content negativity variable. Worth mentioning, is that also the Content JR variable has significant *AME* values for 12- and 6-month news horizons, but the positive signs for both values appear counterintuitive as we expect the probability of bankruptcy to decrease when increasing the sentiment ratios.

6.1.4 Xgboost - 24-month prediction horizon

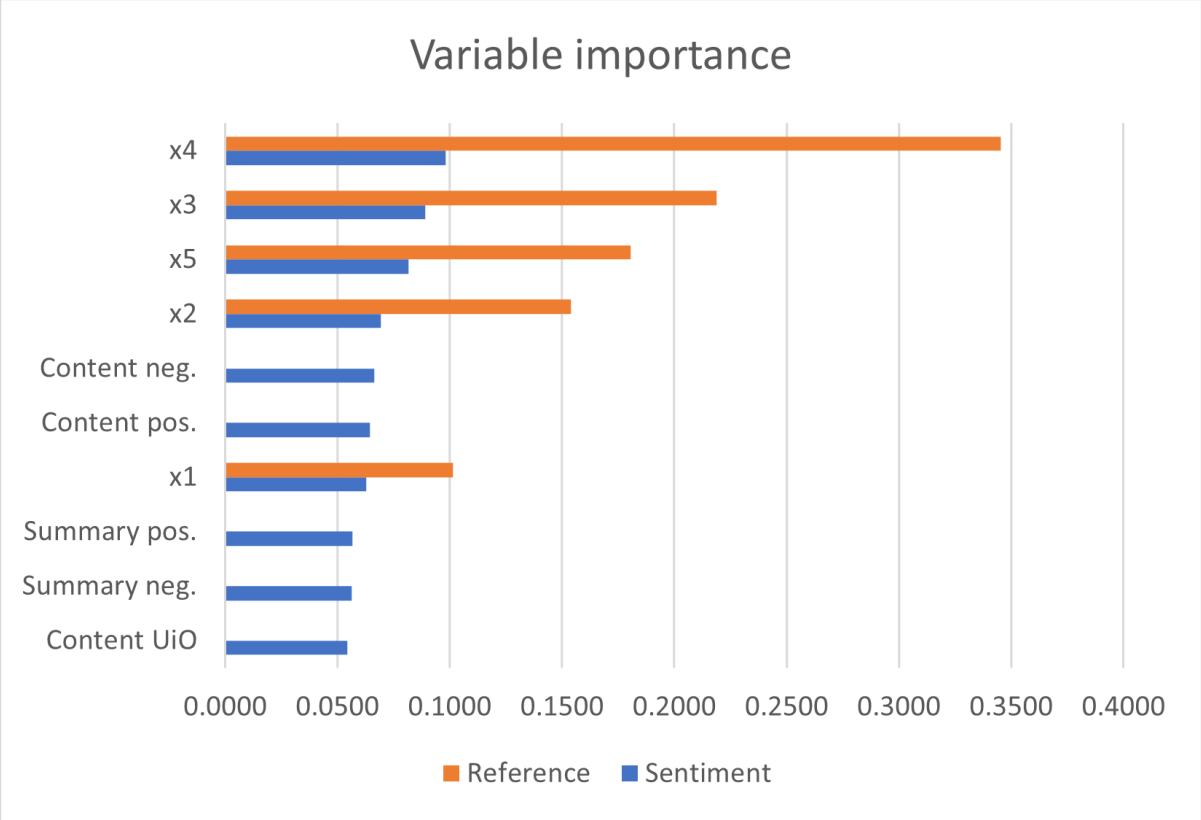
Table 6.6: Performance measures - xgboost 24-month prediction horizon

	Sentiment			**Reference**			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.7467	0.804	0.7821-0.8376	0.7454	0.804	0.7762-0.8346	0.4293
6 month	0.7410	0.796	0.7690-0.8337	0.7439	0.813	0.7857-0.8476	* 0.0317
3 month	0.7308	0.799	0.7667-0.8427	0.7339	0.810	0.7775-0.8480	0.2941
1 month	0.7683	0.844	0.8197-0.8893	0.7656	0.844	0.8089-0.8851	0.5352

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

In table 6.6 we present the results for the *xgboost* model using a 24-month prediction horizon. We once again observe that the performance measures *AUC* and balanced accuracy obtained with *xgboost* are superior to the performance measures of the GLMs presented in table 6.4. However, there is still no evidence of sentiment models significantly improving the predictive performance. For all model configurations in table 6.6, the reference model performs as good as or slightly better than the sentiment model in terms of *AUC*. In terms of balanced accuracy, the sentiment and reference models also obtain very similar results.

Figure 6.2: Average variable importance for xgboost, 24-month prediction horizon



Top 10 performing variables, measured in gain. See table 4.4 for variable descriptions.

In figure 6.2 we illustrate the averaged variable importance in terms of gain obtained from the *xgboost* models. Once more we observe that the financial ratios are present among the variables of the highest average importance. However, both content negativity and content positivity are assigned an average gain value that is slightly higher than the one of x1 in the sentiment models.

6.2 With rebalancing

6.2.1 GLM - 12-month prediction horizon

Table 6.7: Performance measures - Resampled GLM 12-month prediction horizon

	Sentiment			Reference			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.6833	0.722	0.6580-0.7864	0.7036	0.772	0.7157-0.8289	** 0.0079
6 month	0.6757	0.741	0.6617-0.8206	0.7297	0.778	0.7044-0.8518	0.1170
3 month	0.6000	0.656	0.5632-0.7486	0.6750	0.726	0.6406-0.8121	** 0.0078
1 month	0.6844	0.776	0.6861-0.8658	0.6874	0.783	0.6895-0.8766	0.8091

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

In table 6.7 we present the performance measures obtained from the GLMs when resampling the training dataset and using a 12-month prediction horizon. In terms of AUC, we observe that the sentiment models perform worse than the reference models for all news horizons. For the 12- and 3-month news horizons, the AUC in the sentiment models are also significantly worse. Also, the balanced accuracies are higher for the reference models in all model configurations. From the obtained results, it appears that the performed resampling has not improved the predictive power of the sentiment models, relative to the reference models.

Table 6.8: Average marginal effects - Resampled GLM 12-month prediction horizon

Variables/	**Sentiment**				**Reference**			
News horizon	**12-month**	**6-month**	**3-month**	**1-month**	**12-month**	**6-month**	**3-month**	**1-month**
x1	0.0350	-0.0211	* 0.0823	* 0.2097	0.0150	0.0350	0.0673	0.1785
x2	0.0815	* -0.1540	* 0.0282	-0.0396	0.0681	** -0.1936	* 0.0288	-0.0978
x3	*** -0.3877	-0.0851	*** -0.6409	* -0.2220	*** -0.3934	-0.0828	*** -0.5825	-0.1796
x4	*** -0.1798	*** -0.1864	*** -0.1192	*** -0.4279	*** -0.1524	*** -0.1971	*** -0.1370	*** -0.3905
x5	0.0158	0.0039	0.0054	0.0059	0.0173	0.0064	0.0057	0.0094
Title UiO	-0.2448	0.0759	-0.0503	0.0018				
Summary UiO	-0.0855	0.0121	-0.1983	* -0.4592				
Content UiO	** 0.1827	-0.0276	-0.1519	*** -0.3121				
Title JR	*** -0.4993	0.0195	0.1965	** -0.6121				
Summary JR	0.3302	* -0.5152	-0.0978	** -0.6755				
Content JR	0.1064	* 0.2661	* 0.3194	* 0.3907				
Title pos.	0.9502	-0.0650	0.0556	0.3669				
Summary pos.	-0.6563	-0.8804	1.0545	* 2.3704				
Content pos.	-2.1108	-0.3912	-3.0649	* 4.0461				
Title neg.	-0.2236	-0.1216	-0.4026	-0.0199				
Summary neg.	0.0173	-0.3480	-0.6665	** -2.9264				
Content neg.	* 3.6975	* 3.6229	* 4.6980	-3.5508				

Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05. See table 4.4 for variable descriptions.

In table 6.8 we present the Average Marginal Effects (AME) obtained from the GLMs. Contrary to the *AME* values obtained when not resampling the training data, we observe that some of the financial ratios have assigned *AME* values that are both significant and not equal to 0. The *AME* values for both x3 (Earnings before interest and taxes/Total assets) and x4 (Market value of equity/Total liabilities) are highly significant for the majority of news horizons and with a negative sign. The intuition that the probability of bankruptcy decreases when increasing x3 and x4 also concur with the findings in other literature. However, the *AME* values of the other financial ratios, x1, x2 and x5, do not follow the same pattern. Furthermore, we observe that the *AME* values of x5 are insignificant for all model configurations.

We also observe an increase in significant *AME* values among the sentiment variables, compared to the GLM models without resampling. The 1-month news horizon sentiment model has several significant *AME* values assigned to the different sentiment values. However, we observe that there is no consistency regarding the sign of the values. In other words, an increase in polarity score as a result of more positive words does not necessarily reduce the probability of bankruptcy, according to the presented *AME* values table 6.8.

6.2.2 Xgboost - 12-month prediction horizon

Table 6.9: Performance measures - Reweighted xgboost 12-month prediction horizon

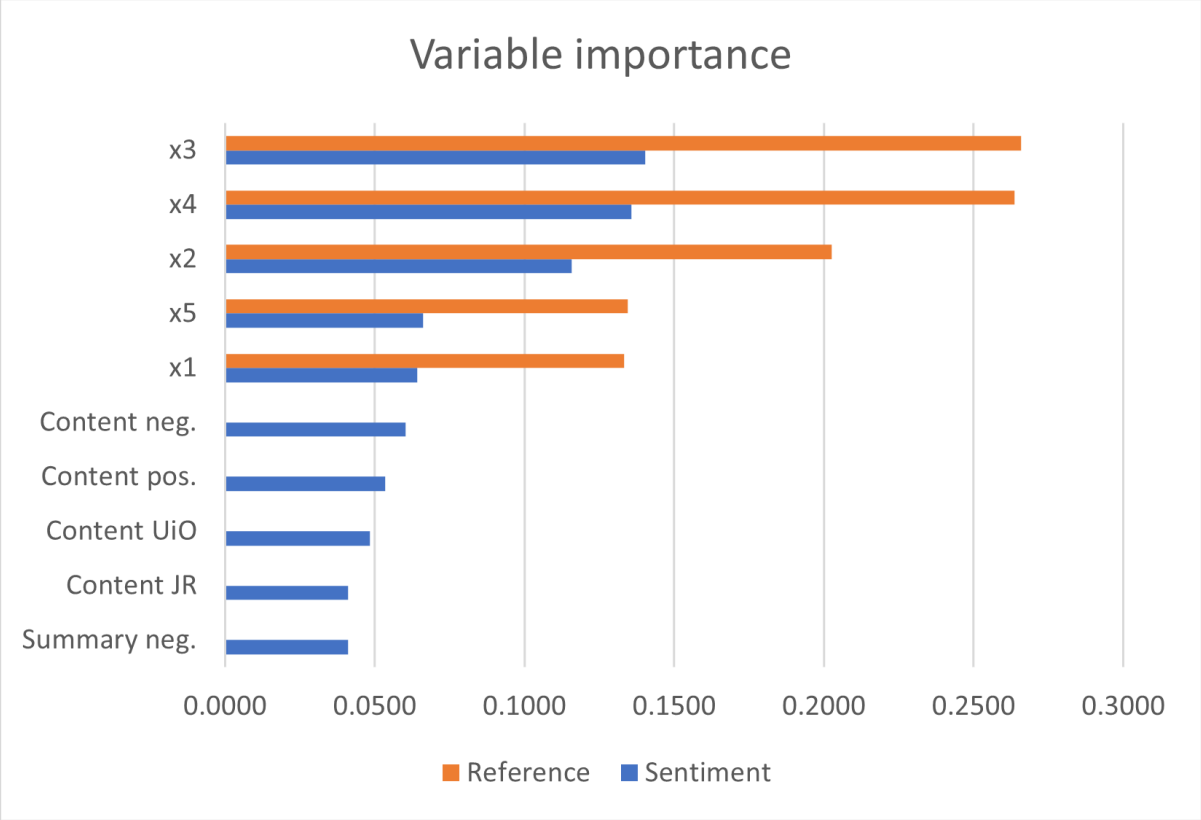
	Sentiment			**Reference**			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.7000	0.793	0.7499-0.8565	0.7500	0.788	0.7398-0.8552	0.8107
6 month	0.6757	0.752	0.6829-0.8411	0.6843	0.777	0.6941-0.8669	0.5750
3 month	0.6599	0.748	0.6801-0.8356	0.7041	0.750	0.6691-0.8398	0.8714
1 month	0.7187	0.805	0.7429-0.9002	0.7083	0.805	0.7489-0.8930	0.9845

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

In table 6.9 we present the performance measures obtained from the *xgboost* models when reweighting the two classes. In terms of *AUC*, the sentiment models and reference models still perform very similarly and there is no significant difference. In terms of balanced accuracy, the reference model performs better for all model configurations, except for the

1-month news horizon. In other words, the reweighting has not significantly proven to improve the sentiment models, relative to the reference models.

Figure 6.3: Average variable importance for reweighted xgboost - 12-month prediction horizon



Top 10 performing variables, measured in gain. See table 4.4 for variable descriptions.

In figure 6.3 we illustrate the averaged variable importance in terms of gain obtained from the *xgboost* models. From the illustrated ranking, it is clear that the financial ratios are assigned the higher gain values.

6.2.3 GLM - 24-month prediction horizon

Table 6.10: Performance measures - Resampled GLM 24-month prediction horizon

	Sentiment			Reference			
News	*Bal. Acc.*	*AUC*	*95% CI*	*Bal. Acc.*	*AUC*	*95% CI*	*DeLong*
12 month	0.6543	0.713	0.6800-0.7465	0.6713	0.728	0.6938-0.7628	0.0692
6 month	0.6446	0.701	0.6632-0.7392	0.6882	0.743	0.7043-0.7807	*** 0.0002
3 month	0.6681	0.753	0.7102-0.7957	0.6735	0.758	0.7151-0.8005	0.6559
1 month	0.7381	0.812	0.7654-0.8578	0.7558	0.816	0.7699-0.8628	0.4710

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

In table 6.10 we present the performance measures obtained from the GLMs when resampling the training dataset and using a 24-month prediction horizon. In terms of *AUC*, we observe that the sentiment models perform either slightly worse or significantly worse, depending on the news horizon. Also, the balanced accuracy is slightly lower for the sentiment models for all news horizons.

Table 6.11: Average marginal effects - Resampled GLM 24-month horizon

Variables/	**Sentiment**				**Reference**			
News horizon	**12-month**	**6-month**	**3-month**	**1-month**	**12-month**	**6-month**	**3-month**	**1-month**
x1	* -0.0319	-0.0363	-0.0079	-0.0536	* -0.0335	-0.0358	-0.0265	-0.0275
x2	*** 0.0549	-0.0003	0.0050	* 0.0671	*** 0.0561	-0.0004	0.0119	0.0432
x3	-0.0235	0.0195	** -0.1086	*** -0.3162	-0.0267	0.0225	*** -0.1067	*** -0.3051
x4	*** -0.1261	*** -0.1496	*** -0.1946	*** -0.2019	*** -0.1226	*** -0.1571	*** -0.1706	*** -0.1782
x5	*** 0.0243	-0.0015	*** 0.0216	*** 0.0426	*** 0.0255	-0.0003	*** 0.0255	*** 0.0431
Title UiO	0.0149	0.0856	-0.0684	0.0684				
Summary UiO	-0.0284	-0.0895	-0.0458	-0.0143				
Content UiO	0.0401	0.0219	0.0277	0.0524				
Title JR	* -0.1966	-0.1672	0.1388	-0.0550				
Summary JR	-0.0712	-0.1201	-0.1165	0.0541				
Content JR	*** 0.2353	0.1086	0.0864	-0.1391				
Title pos.	0.0049	-0.0449	0.1016	-0.0999				
Summary pos.	0.1330	0.5498	0.3058	0.3362				
Content pos.	-0.4989	0.8486	0.7294	-0.6693				
Title neg.	0.1560	0.4655	0.1775	-0.0271				
Summary neg.	0.2098	0.2131	-0.1133	0.6817				
Content neg.	0.9469	0.6070	0.8231	0.2496				

Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05. See table 4.4 for variable descriptions.

In table 6.11 we present the Average Marginal Effects (AME) obtained from the GLMs. Similar to the resampled GLMs that predict on a 12-month horizon, we observe that many of the financial ratios have significant *AME* values. We now also observe that the financial ratio x5 has *AME* values of significance. The interpretation of x3 and x4 still seem to follow our intuition that the probability of bankruptcy decreases when increasing the two ratios. Both x1 and x2 have significant AMEs for the 12-month news horizon, and x1 consistently has a negative sign, which is intuitive in terms of bankruptcy probability decreasing as x1 increases. With regards to x5, the positive marginal effects appear counterintuitive, as we expect the probability of bankruptcy to decrease when improving sales relative to total assets.

6.2.4 Xgboost - 24-month prediction horizon

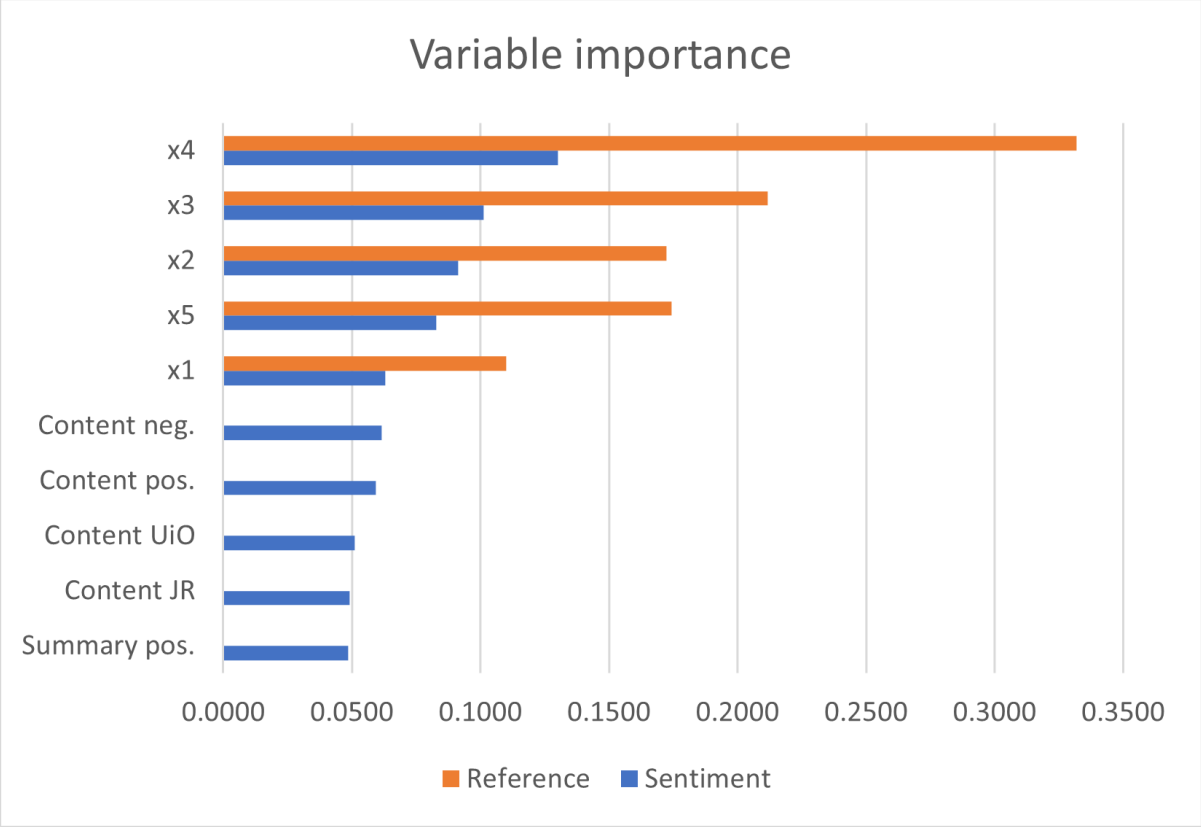
Table 6.12: Performance measures - Reweighted xgboost 24-month prediction horizon

	Sentiment			Reference			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.7222	0.781	0.7578-0.8186	0.7407	0.797	0.7696-0.8305	0.2341
6 month	0.7117	0.780	0.7561-0.8216	0.7378	0.807	0.7809-0.8446	* 0.0138
3 month	0.6923	0.737	0.6981-0.7904	0.7308	0.802	0.7751-0.8445	*** 0.0003
1 month	0.7500	0.806	0.7759-0.8633	0.7813	0.84	0.8104-0.8866	0.1649

See table 6.1 for DeLong explanations. Signif. codes: 0 '* * * ' 0.001 '* * ' 0.01 '* ' 0.05.

In table 6.12 we present the performance measures obtained from the *xgboost* models when reweighting the two classes and using a 24-month prediction horizon. Once again, we observe that the AUC is higher in the reference models, and significantly better for the 3- and 6-month news horizons. The balanced accuracy is also better for all reference models.

Figure 6.4: Average variable importance for reweighted xgboost - 24-month prediction horizon



Top 10 performing variables, measured in gain. See table 4.4 for variable descriptions.

In figure 6.4 we illustrate the averaged variable importance in terms of gain obtained from the *xgboost* models. The ranking is very similar to the one obtained from the 12-month prediction horizon models. In other words, the financial ratios are still assigned the highest gain values.

6.3 Summarized results

Based on the presented performance measures in the results, there is no evidence of significantly improved predictive power when adding sentiment variables. The reference models only including the financial ratios perform either slightly or significantly better with few exceptions. When resampling the dataset, we observe that the GLMs tend to perform better overall in terms of AUC and balanced accuracy, but sentiment variables still do not outperform the reference models. As for the *xgboost* model, the reweighting of the classes does not seem to have any large impact on either the overall performance measures or the predictive power of sentiment variables.

Furthermore, we have presented the average marginal effect values for the explanatory variables in the GLMs. When we do not perform resampling, we observe that very few explanatory variables had significant AME values. We observe that content negativity tends to have significant AME values for the 12- and 6-month news horizons, but other than that there is no clear trend among the sentiment variables in terms of significant AME values. When resampling the dataset, we observe that several of the financial ratios are highly significant, which seems to make sense as the models in general perform better when resampling. This is also expected as the financial ratios have been proven to have predictive power in bankruptcy prediction in previous studies.

For some of the model configurations, the presented significant AME values of sentiment variables seem to make sense intuitively. However, there are also instances of significant AME values that are contradictory to our intuition that an increase in polarity score, in other words more positive words, will decrease the probability of bankruptcy. These

contradictory results might indicate that the *AME* values are somewhat arbitrary for the different models, and therefore not credible sources of information when trying to interpret the effects.

Based on the developed *xgboost* models, we have derived averaged variable importances for each of the prediction horizons and both with and without reweighting the models. In general, a clear trend among all model configurations is that the five financial ratios are among the variables of the highest average gain. However, we do observe that content negativity and content positivity are assigned relatively high values of gain and also surpass the x5 variable in the non-weighted models.

6.3.1 Further analysis of content negativity

As there is a tendency for content negativity to be significant for GLMs and highly important for *xgboost* models, we choose to investigate this further. We therefore develop GLMs with resampling and *xgboost* models without reweighting, both on a 12-month prediction horizon. These are models where content negativity seemingly has some importance in terms of significant *AME* values and relatively high variable importance.

Table 6.13: Performance measures - Resampled GLM 12-month prediction horizon

	Sentiment			**Reference**			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.6863	0.765	0.7092-0.8206	0.7036	0.772	0.7157-0.8289	0.3611
6 month	0.7027	0.770	0.6977-0.8431	0.7297	0.778	0.7044-0.8518	0.3722
3 month	0.6552	0.728	0.6421-0.8142	0.6750	0.726	0.6406-0.8121	0.8740
1 month	0.7083	0.783	0.6894-0.8766	0.6874	0.783	0.6895-0.8766	0.9455

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

Table 6.14: Performance measures - xgboost 12-month prediction horizon

	Sentiment			**Reference**			
News	***Bal. Acc.***	***AUC***	***95% CI***	***Bal. Acc.***	***AUC***	***95% CI***	***DeLong***
12 month	0.7333	0.794	0.7372-0.8581	0.7333	0.794	0.7391-0.8576	0.8834
6 month	0.7161	0.802	0.7340-0.8777	0.7343	0.807	0.7355-0.8865	0.4691
3 month	0.7000	0.753	0.6719-0.8393	0.7030	0.742	0.6631-0.8291	0.1905
1 month	0.7083	0.819	0.7559-0.8936	0.7227	0.825	0.7700-0.8972	0.2609

See table 6.1 for DeLong explanations. Signif. codes: 0 ‘∗ ∗ ∗ ’ 0.001 ‘∗ ∗ ’ 0.01 ‘∗ ’ 0.05.

For our GLMs containing content negativity displayed in table 6.13, we see a clear improvement for all performance measures compared to the models containing all sentiment

variables. However, the reference models still outperform the sentiment models except when considering a 3-month news horizon. Yet, the outperformance of the sentiment model is not significant. The same trend is present for our *xgboost*-model performance results, displayed in table 6.14, where there is an improvement in sentiment models, but reference models still perform the best. Also here the 3-month news horizon model performs slightly better than the reference model, but again there is no significance.

7 Discussion

7.1 Limitations

This thesis has exclusively focused on the lexicon-based approach when performing sentiment analysis. As a result of this, the dictionaries of choice play a significant role. One possible limitation in this context is that the dictionaries are not fully capturing the sentiment of news articles presenting information that indicates presence of for example liquidity problems or financial uncertainty. Certain words that indicate such situations for a person reading the news, might not be weighted sufficiently, or captured at all by the dictionary. In addition, a short mention of such happenings can potentially be outweighed by noise in the news article that is not providing any value with regards to probability of bankruptcy.

Specifically for our paper, we have worked with news articles in Norwegian. Jockers & Rinker's dictionary and the valence shifters used in our analysis are both developed for the English language. Therefore, we might encounter poorly translated words in both of the text lists. For valence shifters, we have manually inspected the translated lifts, which we find sufficient for analytical purposes. On the other hand, the Jockers & Rinker-dictionary has thousands of words, making it harder to manually inspect. Thus, the sentiment scores derived from this dictionary might be more error-prone due to the imperfect translation method.

It was necessary to develop a matching algorithm in order to assign the news to the mentioned companies. The method proposed is optimized on large scale data, meaning that the correctness will be sufficient when performing larger amounts of matchings. The intuition behind the matching algorithm is also coinciding with the methodology used by Enin when working with textual data analysis. However, as we observe in all the *final data* subsets, the amounts of bankruptcy observations are relatively few compared to the non-bankruptcy observations. Hence, if the matching algorithm is mismatching a sufficient amount of the bankruptcy observations, the validity of the assigned sentiment

variables will weaken.

In this paper we have stressed the presence of highly imbalanced datasets. Since we have only included companies mentioned in the news the already imbalanced datasets with regards to bankruptcy and non-bankruptcy have become even more imbalanced. In order to tackle this problem, we have performed resampling and reweighting in our analysis. However, there is no guarantee that these techniques are sufficient or optimal for this given case. Different rebalancing methods will yield different results, which makes the results highly dependent on the chosen resampling method. When we rebalance the data used for training the model, the data is also less depictive of reality.

7.2 Further research

Although the research in this paper could not prove any significance of news articles, we still believe that the information provided in textual data can be utilized in bankruptcy prediction. Due to the time and computational power constraint in our research, not all possible approaches have been fully investigated. In the following we present some thoughts for further research within the field that we find interesting and promising, both regarding methodology, other approaches and data sources.

The lexicon-based sentiment analysis approach is dependent on the lexicon of choice. A further development of our research is thus to experiment with other dictionaries. Especially the use of domain-specific dictionaries is interesting. One approach could be to look into more finance-related dictionaries and investigate how this affects sentiment ratios and predictive power. It is also possible to develop dictionaries that relate directly to bankruptcy, weighting words such as bankrupt as particularly negative. However, the lexicon-based method still includes some sort of dependency on how the dictionary is developed and the words are categorized.

Another interesting sentiment analysis approach would be to utilize a supervised machine learning technique, such as *Artificial Neural Networks (ANN).* Such methods have not been applied in this thesis, but it would be interesting to investigate whether the methods

would yield other results on the same data. By training a model on a news set and testing on an out-of-sample test-set, one could for example classify news articles as negative or positive, either as a binary classification or a multi-class classification.

Furthermore, we find the approach of searching for specific keywords that are related to financial distress and bankruptcy interesting. This is a relatively straightforward method and could potentially yield interesting results. When performing such an analysis, there would also need to be a defined list of words containing words that are related to financial distress and bankruptcy.

The decrease in bankruptcy rate that happens when companies are merged with news articles is another element of this thesis that could be interesting to investigate further. The decrease from 1 percent to 0.1 percent is an indicator that companies mentioned in the news have a lower bankruptcy rate, at least for the companies and news articles included in this paper. One interesting aspect could be to investigate why the rate changes. Another interesting aspect would be to see whether being mentioned in the news has some form of predictive value when it comes to the probability of going bankrupt.

This paper has considered Norwegian private limited companies operating in all sectors except for telecom, media and IT. As we have had a limited number of bankruptcies, we decided not to look into specific industries. However, should one be able to obtain a more balanced dataset with a greater number of bankruptcy observations, looking into how sentiment variables affect bankruptcy probabilities within specific industries could be an interesting further development.

Furthermore, instead of including news articles from all different news providers, it would be interesting to only regard the news articles categorized as economic papers. By combining this delimitation with a domain-specific dictionary, it could lead to some interesting findings. A potential downside, however, could be that small and medium sized companies would not have any mentions if only nation-wide economic papers are included in the analysis. Therefore, many observations could potentially be excluded.

Another approach could be to perform a similar sentiment analysis as introduced in this thesis, but instead of news articles one could look into annual reports. The annual reports contain sections where the company's financial situation is presented in writing. The annual reports are also connected to their respective companies, meaning there would be no need for matching. The potential source of error when matching textual data to companies would thus be omitted. However, the external perspective introduced by news articles and the mentioned time lag would not be captured when basing the analysis on internal information from the annual reports.

8 Conclusion

The main objective in this paper is to investigate whether sentiment variables derived from news articles have predictive power with regards to bankruptcy prediction. In order to assess the problem, we have developed sentiment variables applying textual data analysis and developed multiple models estimated using two different techniques; Generalized Linear Modelling and *xgboost*. The developed models including sentiment variables have all been compared to reference models only including Altman's five financial ratios in order to examine whether the performance improves. The performance is measured in AUC and balanced accuracy.

The source of the financial annual accounts is the registry of Brønnøysund Register Centre. The textual data is gathered from Norwegian news articles published on the internet. Textual data preprocessing methods and a lexicon-based sentiment analysis is then applied on the retrieved textual data. Furthermore, the sentiment variables derived from news articles have been added to the financial ratios on different time lags ranging from 1 to 12 months. By doing so, we both fill the time-lag between account year and published accounts and add external information to the annual accounts observations. Both sentiment and reference models have been estimated using 5-fold cross validation and validated on hold-out test data. Furthermore, the *xgboost* models have been hyperparameter tuned in order to optimize the output AUC. The bankruptcy event has been predicted on both a 12-month horizon and a 24-month horizon, capturing the majority of registered bankruptcy events. As a result of the final datasets being highly imbalanced, we have also performed resampling for the GLMs and reweighting for the *xgboost* models.

Our research indicates that there is no significant predictive power in the derived sentiment variables. This is the case for all models, both with and without rebalancing measures. All model configurations without exception substantiate this conclusion. This is also the case when further investigating whether the best performing sentiment variable alone can improve the models. The financial ratios tend to outperform the sentiment variables, both in terms of marginal effects and belonging significance from the GLMs, and in terms of

variable importance from the *xgboost* models. Furthermore, we point to several possible shortcomings. These range from error-prone matching of news articles to companies, to highly generalized dictionaries utilized in the sentiment analysis instead of domain-specific. The sentiment variables utilized are simply adding noise to the reference models, in most cases worsening the predictive ability of the models. Regardless, given our approach there is seemingly no significant improvement in predictive performance from including sentiment variables.

On a final note, although our approach yields no significant improvement, we believe there is great potential in combining the fields of textual data analysis and machine learning for predictive purposes within the Norwegian market. We have presented many potential further developments and would like to emphasize that other approaches to textual data analysis may yield different results. We would also like to substantiate the fact that textual data analysis, especially with regards to the Norwegian language, is still in its early phases of adoption. Potential contributions such as domain-specific lexicons in Norwegian may yield different results than the general-purpose lexicons utilized in our sentiment analysis. This paper is only one contribution, to what will hopefully be many more, within the combined field of textual data analysis and machine learning for predictive analysis in the Norwegian market.

References

Adkins, L. (2014). Using gretl for principles of econometrics. *Economics Working Paper Series. 1412*, 4:370–384. Retrieved from: http://EconPapers.repec.org/RePEc:okl:wpaper:1412.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(3):589–609. Retrieved from: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1968.tb00843.x.

Barnes, J., Touileb, S., Øvrelid, L., and Velldal, E. (2019). *Lexicon information in neural sentiment analysis: a multi-task learning approach*. Proceedings of the 22nd Nordic Conference on Computational Linguistics, Turku, Finland. Retrieved from: https://www.aclweb.org/anthology/W19-6119/.

Beaver, W. H. (1966). Financial ratios as predictors of failure. *The Journal of Accounting*, (4):71–111.

Bellovary, J. L., Giacomino, D. E., and Akers, M. D. (2007). A review of bankruptcy prediction studies: 1930 to present. journal of financial education. *The Journal of Finance*, 33:1–42. Retrieved from: https://epublications.marquette.edu/cgi/viewcontent.cgi?article=1025&context=account.

Benesty, M., Chen, T., He, T., and Tang, Y. (2018). Understand your dataset with xgboost. Retrieved from: https://cran.r-project.org/web/packages/xgboost/vignettes/discoverYourData.html#numeric-v.s.-categorical-variables.

Berg, D. (2007). Bankruptcy prediction by generalized additive models. *Applied Stochastic Models in Business and Industry*, 23(2):129–143. Retrieved from: https://onlinelibrary.wiley.com/doi/abs/10.1002/asmb.658/.

Brownlee, J. (2020). How to configure xgboost for imbalanced classification. Retrieved from: https://machinelearningmastery.com/xgboost-for-imbalanced-classification/.

Brønnøysund (2020). Kunngjøringer fra konkursregisteret. Retrieved from: https://www.brreg.no/produkter-og-tjenester/kunngjoringer-fra-konkursregisteret/.

Burnaev, E., Erofeev, P., and Papanov, A. (2017). Influence of resampling on accuracy of imbalanced classification. Retrieved from: https://arxiv.org/pdf/1707.03905.pdf.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Retrieved from: https://arxiv.org/pdf/1603.02754.pdf.

Chowdary, D. H. (2020). Decision trees explained with a practical example. Retrieved from: https://towardsai.net/p/programming/decision-trees-explained-with-a-practical-example-fe47872d3b53.

Datamentor (2020a). R for loop. Retrieved from: https://www.datamentor.io/r-programming/for-loop/.

Datamentor (2020b). R if...else statement. Retrieved from: https://www.datamentor.io/r-programming/if-else-statement/.

Fawcett, T. (2006). An introduction to roc analysis. Retrieved from: https://people.inf.elte.hu/kiss/13dwhdm/roc.pdf.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–407. Retrieved from: https://projecteuclid.org/download/pdf_1/euclid.aos/1016218223.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). *The elements of statistical learning*, volume 2. Springer series in statistics, New York, USA. Retrieved from: https://web.stanford.edu/~hastie/Papers/ESLII.pdf.

Frost, J. (2020). Multicollinearity in regression analysis: Problems, detection, and solutions. Retrieved from: https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/.

Ganesan, K. (2019). All you need to know about text preprocessing for nlp and machine learning. Retrieved from: https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html.

Hjelseth, I. and Raknerud, A. (2016). A model of credit risk in the corporate sector based on bankruptcy prediction. Retrieved from: https://static.norges-bank.no/contentassets/3da7332610b74bdeacfd208e1a1a76f2/staff_memo_20_2016.pdf?v=03/09/2017123537&ft=.pdf.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. Retrieved from: https://github.com/ltgoslo/norsentlex.

Infomedia (2020). Media monitoring with maximum precision. Retrieved from: https://infomedia.org/media-monitoring/.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Jo, N.-o. and Shin, K.-s. (2016). Bankruptcy prediction modeling using qualitative information based on big data analytics. Retrieved from: http://jiisonline.evehost.co.kr/files/DLA/20160704142343_03-%EC%A1%B0%EB%82%A8%EC%98%A5%C2%B7%EC%8B%A0%EA%B2%BD%EC%8B%9D.pdf.

Kuhn, M. and Johnson, K. (2013). *Applied predictive modeling*, volume 26. Springer.

Langerfeld, C. and Rohrer, M. (2019a). Applied textual data analysis for business and finance. Lecture 12.

Langerfeld, C. and Rohrer, M. (2019b). Applied textual data analysis for business and finance. Retrieved from: https://www.nhh.no/en/courses/applied-textual-data-analysis-for-business-and-finance/.

Leeper, T. (2018a). Interpreting regression results using average marginal effects with r's margins. Retrieved from: https://rdrr.io/cran/margins/f/inst/doc/TechnicalDetails.pdf.

Leeper, T. (2018b). Package 'margins'. Retrieved from: https://cran.r-project.org/web/packages/margins/margins.pdf.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society*, 135(3):370–384. Retrieved from: https://www.jstor.org/stable/2344614?seq=1.

Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, 18(1):109–131.

Prabhu, R. (2018). Understanding hyperparameters and its optimisation techniques. Retrieved from: https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568.

Rinker, T. (2019a). Package 'lexicon'. Retrieved from: https://cran.r-project.org/web/packages/lexicon/lexicon.pdf.

Rinker, T. (2019b). Package 'sentimentr'. Retrieved from: https://cran.r-project.org/web/packages/sentimentr/sentimentr.pdf.

SSB (2008). Næringsstandard og næringskoder. Retrieved from: https://www.ssb.no/virksomheter-foretak-og-regnskap/naeringsstandard-og-naeringskoder.

W3Schools (2020). Javascript if/else statement. Retrieved from: https://www.w3schools.com/jsref/jsref_if.asp.

Walimbe, R. (2017). Handling imbalanced dataset in supervised learning using family of smote algorithm. Retrieved from: https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family.

Appendix

A1 Industry Sector Codes

Table A1.1: Industry sector classification from Statistics Norway (SSB)

Number	Sector industry	From	To
1	Primary industries	0	5000
2	Oil/Gas/Mining	5000	10000
3	Manufacturing industries	10000	35000
4	Energy/Water/Sewage/Util.	35000	40000
5	Construction & Property Development	40000	45000
6	Trade	45000	49000
7	Shipping	50000	51000
8	Transport, Tourismm (excl. Shipping)	49000	58000
9	Telecom/IT/Media	58000	64000
10	Finance, Insurance	64000	68000
11	Real Estate, Services	68000	69000
12	General services (ecxl. R&D)	69000	84000
13	Research & Development	72000	73000
14	Public sector/Culture	84000	-

Source: https://www.ssb.no/virksomheter-foretak-og-regnskap/naeringsstandard-og-naeringskoder

A2 Correlation matrix for final data

Table A2.1: Correlation matrix for final data subset regarding 12-month news horizon

	x1	x2	x3	x4	x5	Tit. UiO	Sum. UiO	Con. UiO	Tit. JR	Sum. JR	Con. JR	Tit. pos.	Sum. pos.	Con. pos.	Tit. neg.	Sum. neg.	Con. neg.
x1	*1.00*	*0.84*	*-0.27*	*0.00*	*-0.39*	*0.00*	*0.00*	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
x2	0.84	**1.00**	-0.27	0.00	-0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.00
x3	-0.27	-0.27	**1.00**	0.00	0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
x4	0.00	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
x5	-0.39	-0.34	0.46	0.00	**1.00**	0.00	0.00	0.01	0.00	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.01
Title UiO	0.00	0.00	0.00	0.00	0.00	**1.00**	0.21	0.20	0.33	0.11	0.13	0.64	0.12	0.10	-0.61	-0.11	-0.08
Summary UiO	0.00	0.00	0.00	0.00	0.00	0.21	**1.00**	0.27	0.11	0.33	0.17	0.15	0.55	0.17	-0.12	-0.51	-0.02
Content UiO	0.00	0.00	0.00	0.00	0.01	0.20	0.27	**1.00**	0.13	0.19	0.62	0.18	0.21	0.51	-0.06	-0.06	-0.04
Title JR	0.00	0.00	0.00	0.00	0.00	0.33	0.11	0.13	**1.00**	0.17	0.16	0.28	0.09	0.12	-0.15	-0.04	0.01
Summary JR	0.00	0.00	0.00	0.00	0.01	0.11	0.33	0.19	0.17	**1.00**	0.22	0.11	0.25	0.14	-0.04	-0.09	0.02
Content JR	0.00	0.00	0.00	0.00	0.01	0.13	0.17	0.62	0.16	0.22	**1.00**	0.14	0.16	0.34	-0.02	0.00	0.05
Title pos.	0.00	0.00	0.00	0.00	0.00	0.64	0.15	0.18	0.28	0.11	0.14	**1.00**	0.17	0.16	0.05	0.00	0.05
Summary pos.	0.00	0.00	0.00	0.00	0.00	0.12	0.55	0.21	0.09	0.25	0.16	0.17	**1.00**	0.21	0.01	0.13	0.09
Content pos.	0.00	0.00	0.00	0.00	0.01	0.10	0.17	0.51	0.12	0.14	0.34	0.16	0.21	**1.00**	0.03	0.03	0.38
Title neg.	0.00	0.00	0.00	0.00	0.00	-0.61	-0.12	-0.06	-0.15	-0.04	-0.02	0.05	0.01	0.03	**1.00**	0.16	0.17
Summary neg.	0.00	-0.01	0.00	0.00	0.00	-0.11	-0.51	-0.06	-0.04	-0.09	0.00	0.00	0.13	0.03	0.16	**1.00**	0.13
Content neg.	0.00	0.00	0.00	0.00	0.01	-0.08	-0.02	-0.04	0.01	0.02	0.05	0.05	0.09	0.38	0.17	0.13	**1.00**