PREDICTING FINANCIAL DISTRESS

MASTER'S THESIS

Frederik Winther Nielsen & Johan Dybkjaer-Knudsen

Study numbers	**101284 | 93335**
Education	**Cand.Merc.IT (Data Science)**
Submission date	**May 15, 2020**
Supervisor	**Nicholas Skar-Gislinge**
No of characters	**144,855**
No of pages	**66**

PREDICTING FINANCIAL DISTRESS

By Frederik Winther Nielsen & Johan Dybkjær-Knudsen

ABSTRACT

Financial Distress Prediction (FDP) models largely revolve around the utilization of financial information to predict the probability of financial distress of companies. Accurate financial distress predictions are relevant for stakeholders as financial distress can have lasting impacts on both internal stakeholders and external stakeholders. Despite the widely studied area of FDP that has seen recent developments from machine learning, the academic literature has primarily focused on financial information, leaving the potential impact of quantitative non-financial ownership information sparsely studied. The potentially underdeveloped aspect of including non-financial ownership information as a predictor in FDP, the latest development of high-performance models using machine learning, and a considerable amount of data on limited Danish companies, leads to the research question: *"How does the inclusion of non-financial ownership information affect the performance of financial distress prediction models on Danish companies?"* Using public data from the Danish Business Authority, linear discriminant analysis (LDA), logistic regression (LR), and gradient boosted trees (GBT) models are trained on reduced (dense) data using cross-validation and randomized grid search – first trained without the proxy for non-financial ownership information, i.e., company ownership default risk (CODR), and then trained similarly with CODR. Additional GBT-models were trained on the complete (sparse) data for better generalization with and without CODR. The results show that the *sparse-GBT-CODR* is the best-performing model ($AUC = 0.8409$) over other models. Following a discussion on limitations, implications, operationalization approaches, and statistical tests, the thesis concludes that there presumably are potential positive impacts of using non-financial ownership information for FDP on Danish companies but calls for further research.

Keywords: Financial Distress Prediction, Machine Learning, Linear Discriminant Analysis, Logistic Regression, Gradient Boosted Trees

Table of Contents

1 TERMINOLOGY

The following contains the most common list of acronyms and terminology used throughout this thesis. While the first occurrence of each term in the thesis is followed by an explanation, the below list provides the reader with a collective terminology as a point of reference.

Term	Description
CODR	Company ownership default risk, used as a proxy for non-financial ownership information
Dense	We define data as being dense if *all* the data elements are non-empty. However, the usual definition is that a majority of the elements are *non-zero*
FDP	Financial distress prediction
GBT	Gradient boosted trees
LDA	Linear discriminant analysis. Also known as Multiple Discriminant Analysis – however, this is simply a generalized form of LDA for *N* possible classes
LR	Logistic regression
ODR	Ownership default risk
Serial failers	People that are repeatedly involved in company bankruptcies
Sparse	We define data as being sparse if the majority of the data elements are *empty*. However, the usual definition is that a majority of elements are *zero*
Sparse-GBT-CODR	A GBT model trained on sparse data and a CODR feature
UDA	Univariate discriminative analysis

2 INTRODUCTION

The global economy is an intertwined web of transactions, relationships, and complex ripple effects. The performance of any company is undoubtably connected to several stakeholders, both directly and indirectly. This is true both for companies in good periods, but also for companies during subpar periods that might lead to *financially distressed* companies characterized by loan defaulting and potentially bankruptcy.

A company in financial distress often affects both the company itself and all its stakeholders negatively; internal stakeholders such as employees, shareholders, and managers, but also external stakeholders such as business partners, suppliers, customers, regulators, creditors, etc. Situations of financial distress in one company can further exacerbate the financial situation of related companies with subpar financial performance, which could create a ripple effect of bankruptcies in the (global or local) economy.

To some extent, financial distress is a natural part of the economy. Regardless, they can have lasting, but potentially avoidable, negative effects. Hence, the ability to predict these could alleviate some of the negative effects by suppliers of credit, e.g., business partners, banks, etc. Here, for new relationships, the suppliers of credit can accurately risk-assess and price the provision of credit – or deny credit. For existing credit relationships, the negative impact can be lessened by discouraging further provision of credit or disbanding existing relationships prior to a potential financial distress.

Due to the economic impact of financially distressed companies, the ability to anticipate these is highly relevant for a wide variety of industries and stakeholders. Consequently, several data-driven models have been developed over the years to predict financial distress. Many scholars and practitioners have investigated the feasibility of *financial distress prediction* (FDP) for the reasons outlined above and to better assess the risk of providing credit (Schuermann, 2005).

FDP as an academic field has developed considerably since its inception more than half a century ago with *univariate discriminant analysis*, then *linear discriminant analysis*, followed by *conditional probability models* (e.g., logistic regression), and in the latter years with various machine learning (ML) implementations giving rise to promising solutions and increased predictive performance that utilize companies' publicly available financial information. The increased academic and practical focus on ML for FDP, specifically, is driven by a multitude of factors, such as predictive superiority over traditional statistical methods, the ability to identify highly complex patterns in datasets, and due to a less restrictive set of assumptions compared to traditional statistical models (Tang et al., 2020). The proliferation of ML in FDP has also been partly driven by advances in computer processing power. While ML generally has been driven by an abundance of data, much of the academic literature focus exclusively on a limited number of financial statements, e.g., annual reports, often with estimation

samples of less than 500 companies, and with a frequent exclusive focus on public limited companies (Aziz & Dar, 2006).[1]

In opposition to this "narrow" scope, two contemporary academic articles have investigated Danish limited companies (*A/S* and *ApS*), including more than 250,000 financial statements on more than 100,000 unique companies extracted from the Danish Business Authority's elaborate company database, and find that it is possible to create "broad" state-of-the-art FDP-models using ML on financial statements (Christoffersen et al., 2018; Matin et al., 2019).

Despite the methodological and theoretical developments in the field of FDP and the explosion of data availability, most studies only focus on the utilization of purely financial information from financial statements.[2] While it has been shown that financial information carries considerable predictive power, the inclusion of non-financial information external to financial statements and its impact on FDP-models, is sparsely studied in the literature. This includes information relating to the company owners, e.g., the ability of owners to grow companies (proven growth track-record), the experience of owners (number of years owning healthy companies), information on whether owners have been involved in previous financial distresses (a default risk of owners), etc. These three pieces of ownership information all potentially contain relevant information that can be used in FDP-models. The latter point is presumably especially relevant as it includes information on owners' previous financial distresses, which could directly influence the likelihood of future distresses, e.g., the impact of *serial failers* (people that are repeatedly involved in company bankruptcies).

To the best of the authors' knowledge, there have been no studies that incorporate the impact of previous financial distresses of owners on FDP. Some serial bankruptcy studies investigate *serial failers* on a company-level, e.g., Hotchkiss (1995) investigate the post-bankruptcy performance of reorganized companies.[3] Similarly, most literature on ownership influences on financial distress relate to large companies, including corporate governance, ownership concentration, absolute and relative power of shareholders, agency theory, etc. (Daily & Dalton, 1994sa, 1994b; Deng & Wang, 2006; Donker et al., 2009; Lajili & Zéghal, 2010; Mangena & Chamisa, 2008; Manzaneque et al., 2016).

The potential underdeveloped aspect of including non-financial ownership information as a predictor in FDP, the development of high-performance models using ML, and the considerable amount of data

[1] For more recent examples, see Alexandropoulos et al., (2019), Huang & Tserng (2018), Tang et al. (2020), Mai et al. (2019)
[2] However, there have been several promising developments in the area of including textual information from financial statements as predictors of financial distress (see e.g., Mai et al., 2019; Matin et al., 2019; Tang et al., 2020). Further, various FDP-models have also included stock prices (Câmara et al., 2012) and macroeconomic variables (Christoffersen et al., 2018).
[3] See also Denning et al. (2001) on factors for a successful reorganization.

on limited Danish companies from the Danish Business Authority leads us to the following research question:

How does the inclusion of non-financial ownership information affect the performance of financial distress prediction models on Danish companies?

Specifically, this thesis seeks to answer this research question by first investigating the predictive power of linear discriminant analysis, logistic regression, and gradient boosted trees models without the inclusion of non-financial ownership information. Following this, two additional logistic regression and gradient boosted trees models are trained with the addition of ownership information to investigate the potential effect on predictive power. As a proxy for the inclusion of *non-financial ownership information*, this thesis uses *company ownership default risk* (CODR), which is a quantification of the risk to a given company that might arise from the current owners' previous company defaults.[4]

2.1 DELIMITATIONS

In the investigation of this research question, the following delimitations apply. The scope is limited to financial statements from non-financial and non-holding Danish limited (*ApS* and *A/S*) companies covering the period from 2012 to 2018.

Non-financial companies and non-holding companies are excluded for their differing asset structure (Christoffersen et al., 2018; Jackson & Wood, 2013; Matin et al., 2019). Denmark is chosen as a case study due to the elaborate database on Danish companies from the Danish Business Authority. The focus on limited companies (*ApS* and *A/S*) primarily stems from the limited availability of financial information on other legal company structures such as sole proprietorships. The delimitation of the period from 2013 to 2018 is partly limited by data availability where the lower boundary signifies the general introduction of digitized financial statements in 2013 and the upper boundary is limited by the methodological choice of categorizing companies as financially distressed if they declare bankruptcy within a period of two years. Logically, we cannot categorize companies as *not financially distressed* before the two-year period has passed, which excludes financial statements from parts of 2018, all of 2019 and 2020.[5]

[4] An introduction to the formal definition with examples can be found in section 6.2.3.2.
[5] For a more elaborate explanation of the delimitations, see Section 6, *Methodology*.

2.2 STRUCTURE OF THE THESIS

As a guide to the reader, the following provides an overview of the structure of the thesis and the main topics covered in each section.

Immediately following the introduction, the *Literature Review* presents definitions, a brief history of the academic literature on financial distress prediction and the models developed historically, an introduction to academic literature on FDP in Denmark, and an introduction to the contemporary practical FDP-approach of the largest bank in the Nordics, Nordea. Lastly, the discussed academic and practical approaches are discussed in relation to the methodology of this thesis.

The section *Theory* gives a brief introduction to the field of *machine learning* and provides the reader with a foundational introduction to the models. Specifically, it introduces the models: linear discriminant analysis, logistic regression, and gradient boosted trees. It also introduces concepts such as boosting, gradient boosting, metrics for evaluating predictive models, etc.

The *Data* is briefly introduced, outlining the two databases employed in this thesis, the *permanent* and the *financial statements* (FS) databases. The first contains "fundamental" company information such as *name*, *address*, *foundation date*, and, most importantly, a potential *cessation date*. The latter contains all financial statements in an .xml-format for machine-readability.

The subsequent section, *Methodology*, provides an overview of the following methodological considerations: First, the philosophy of science-foundation and research design is described followed by an outline of the data pipeline, including the acquisition, cleansing, and general preparation of data from the *permanent* and *FS* databases, and then the merging of these two data sources. Lastly, once the data has been prepared for analysis, the data analytics section outlines the methodological steps in the application of the models, splitting, model training, grid search, cross-validation, and model evaluation.

The *Results* section presents the AUC-scores of the seven models and visualizes the model ROC-curves.

Discussion presents the various data and model limitations such as erroneous data and inter-dataset comparisons. Then, two sparse models are operationalized using optimized thresholds, and the costs of using these models are calculated. Following this, the McNemar test is performed to test whether the two sparse models are significantly similar. Lastly, areas for future work are discussed.

Lastly, the thesis is wrapped up in the Conclusion, presenting the main findings, answering the research question, and presenting potential impacts.

3 LITERATURE REVIEW

The following literature review outlines notable existing literature on *financial distress prediction* (FDP). First, it includes some definitions relating to the area of FDP. Then, it outlines a brief history of the data-driven methodologies undertaken to predict financial distress. Following this, it presents two notable contemporary papers on financial distress in Denmark on which this thesis draws inspiration from and uses the same data foundation. Then, a brief introduction to the practical implementation of FDP-models in Nordea, the largest Nordic bank. Lastly, the academic literature and its relation to the thesis are presented.

3.1 DEFINITIONS

3.1.1 FINANCIAL DISTRESS

Financial distress can generally be understood as something that degrades a company's profitability considerably. However, since different countries have different accounting procedures and sometimes vastly different legal frameworks, to date there is no unified definition of what constitutes financial distress (Tang et al., 2020, p. 4). Despite a lack of unified definition of financial distress, several country-specific studies make use of the legal status *bankrupt* as the outcome of financial distress – however, what *exactly* must be triggered in a company to declare bankruptcy also differs from country to country, but it generally relates to the inability of companies to meet their financial obligations (Bhimani et al., 2014; Charitou et al., 2008, p. 154). This paper uses the *declaration of bankruptcy* as a proxy for having been in financial distress. More precisely, a company is considered *financially distressed* in the period spanning from two years prior to the act of declaring bankruptcy to the act itself, similar to Christoffersen (2018) and Matin et al. (2019).[6]

In the Danish context, a company is financially distressed if the company has one of the following legal states within a period of two years: *Bankrupt*, *in bankruptcy*, *compulsory dissolved*, or *under compulsory dissolvement.*[7] This classification is in accordance with other academic literature on financial distress (prediction) of Danish companies, e.g., Christoffersen (2018) and Matin et al. (2019).

3.1.2 FINANCIAL DISTRESS PREDICTION

Financial distress in companies have serious ramifications, not only for the business itself, its owners, and its employees, but also for its business environment such as creditors, partner companies, the supply chain in which the company is located, the customers, etc. For creditors, such as banks, business partners, and other parties in the supply chain, a loan default entails that the debtor is unable to make

[6] The chosen window size differs among scholars for different reasons, with prediction windows ranging from 1 to 5 years. This thesis chooses two years specifically to follow the scholarly approaches in Denmark.
[7] In Danish: *Konkurs*, *under konkurs*, *tvangsopløst*, and *under tvangsopløsning.*

its payments on time – which then might lead to insolvency and then start the legal process of declaring for bankruptcy, which can lead to deteriorating liquidity of affected creditors, and in the worst case start a bankruptcy ripple effect. For more than half a century, scholars have researched this topic,[8] and specifically the ability to predict the financial distress of companies, known as *Financial Distress Prediction* (FDP) (Sabela et al., 2018; Sun et al., 2017; Tang et al., 2020; Xin & Xiong, 2011; Zmijewski, 1984, etc.).

3.2 A Brief History of Financial Distress Prediction

The field of financial distress prediction encompasses many different approaches developed over the years. The following provides a brief history of the academic literature on financial distress prediction that are based on quantitative methodologies, which excludes theoretical models of financial distress where the academic focus is on the causes of bankruptcy. Interested readers are referred to Crouhy et al. (2000) for an introduction to the most prominent historical theoretical models.

The field of (quantitative) financial distress prediction has developed considerably over the past half-century since Beaver (1966) – who is generally considered the pioneer within the field of FDP (Charitou et al., 2008; Jones et al., 2017; Mai et al., 2019) – performed univariate financial ratio analyses on financial statements (Beaver, 1966; Jackson & Wood, 2013). Methodologically, Beaver calculated the mean value, dispersion around the mean, and skewness of different financial ratios for both failed and non-failed companies, to investigate the predictive power of *univariate discriminant analysis* (UDA). A univariate discriminatory model uses a single value – here a financial ratio – to categorize companies into either *non-failed* or *failed* in a discriminative manner, i.e., a dichotomous univariate t-test (Gottardo & Moisello, 2019).

As outlined in Figure 1a, Beaver's (1966) *cash flow to total debt* ratio illustrates the discriminatory power of a single financial ratio one year prior to bankruptcy. Specifically, he identifies a certain *cut-off* point on the *cash flow to total debt* dimension. All companies below this threshold are classified as *failed* while the companies above the threshold are labeled *non-failed*. The ability to discriminate between the two classes lessens as the prediction window increases, which is outlined in Figure 1b by the large overlap between non-failed and failed firms using a five year window. This methodology builds on the work of Paul FitzPatrick (1932) who found that there are significant ratio differences at least three years prior to failure and Smith & Winakor (1935) who found "a marked deterioration in the mean values with the rate of deterioration increasing as failure approached" (Beaver, 1966, p. 81).

[8] See Aziz & Dar (2006) for a review of the historical literature.

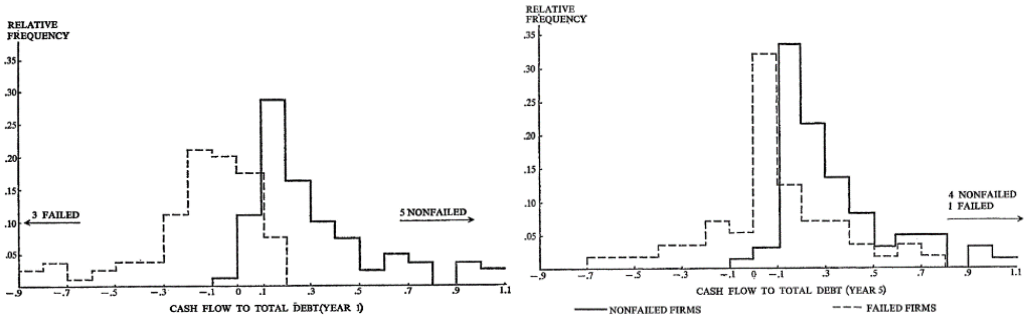

Figure 1 – One of Beaver's (1966) univariate discriminatory models that displays the relative frequency of failed companies (dotted line) and non-failed companies (solid line) on the vertical axis, for *all cash flow to total* debt ratios (horizontal axis). Figure 1a (left) shows the predictions of failed companies when predicting one year ahead, Figure 1b (right) predicts five years ahead (p. 92)

Following the seminal work of Beaver (1966) on univariate discriminant analysis (UDA) using financial ratios, several scholars turned to *linear discriminant analysis* (LDA), which employs more than a single financial ratio.[9] One of the best known examples of LDA in the academic literature is the *Z-score* developed by Altman (1968) based on 91 American manufacturing corporations (Jones et al., 2017), which followed Fisher's (1936) formulation of the linear discriminant that attempts to find a linear combination of features that separates two or more classes of objects or events. Specifically, Altman's Z-score relies on five financial ratios: *Working Capital/Total assets* (x_1), *Retained Earnings/Total Assets* (x_2), *Earnings Before Interest and Taxes/Total Assets* (x_3), *Market Value Equity/Book Value of Total Liabilities* (x_4), and *Sales/Total Assets* (x_5). Altman's (1968) original estimated discriminant on American manufacturing companies is

$$Z = 0.012x_1 + 0.014x_2 + 0.033x_3 + 0.006x_4 + 0.999x_5 \quad (1)$$

For both UDA and LDA, the *non-failed* or *failed* classification is based on thresholds. While UDA utilizes the threshold of a single financial ratio, LDA (such as the Z-score) utilizes several ratios. However, where the UDA approach undertaken by Beaver (1966) specifies a single cut-off point that classifies companies into one of two categories, Altman's (1968) Z-score categorizes into three categories. For the estimated model in equation 1 above, companies with a Z-score greater than 2.99 are categorized as *non-bankrupt*, companies with a Z-score below 1.81 as *bankrupt*, while the interval from 1.81 to 2.99 denote the *zone of ignorance* or a so-called *gray area*. Due to its simplicity, ease of interpretability, and its seemingly good predictive power, the Z-score model gained proponents both inside and outside the academic field, e.g., from financial institutions.

Following the introduction of LDA-models in FDP, of which the Z-score is prototypical, several scholars focused their attention to conditional probability models, e.g., linear probability models, probit,

[9] Often, literature uses the term *multiple discriminant analysis*. However, this is simply a generalized form of LDA for *N* possible classes.

and logit (Aziz & Dar, 2006) – of which the latter has been prevalent in the literature (Aziz & Dar, 2006; Charitou et al., 2008; Hamer, 1983). Ohlson (1980) was the pioneer of using the logit model (logistic regression) for FDP while Zmijewski (1984) was the pioneer of the probit model for FDP (Balcaen & Ooghe, 2006). These new methodological developments partly arose from criticism of the Z-score model (Johnson, 1970; Joy & Tollefson, 1975; Moyer, 1977), including using information for bankruptcy prediction that did not become available until after the event of bankruptcy (Ohlson, 1980, p. 113),[10] and partly from a violation of the underlying statistical assumptions in LDA when predicting financial distress (Balcaen & Ooghe, 2006, p. 86; Tang et al., 2020),[11] e.g., assumptions of multivariate normality, homoscedasticity, linearity, no outliers, etc.

In addition to the purely statistical models in FDP outlined above, scholars increasingly started to focus on *artificially intelligent expert systems* (Aziz & Dar, 2006; Suntraruk, 2010), the first of which was introduced in 1977 by Jerome Friedman (1977) to perform FDP using *recursively partitioned decision trees*. Later, scholars have also utilized *neural networks* and many other types of machine learning (ML) algorithms to perform FDP (Aziz & Dar, 2006, p. 21). Recent studies have shown high performance of *deep learning* models in FDP, e.g., *deep neural networks* and *deep dense multilayer perceptron* (Alexandropoulos et al., 2019; Mai et al., 2019 as cited in Tang et al., 2020). Tsai et al. (2014) further find that ensembles (a collection of models) of ML classifiers tasked with FDP outperform other approaches. They observe that *boosted decision tree ensembles* both outperform other classifier ensembles such as both *boosted* and *bagged support vector machines* and *neural networks* and outperform single ML-classifiers (p. 983).

A considerable amount of the academic literature presents empirical evidence that artificially intelligent expert systems (AI) – or more accurately ML, the subset of AI that deals with how AI-systems "learn" – to be superior to traditional statistical models in the task of FDP (Aziz & Dar, 2006; Jabeur & Fahmi, 2018; Jones et al., 2017; Kuldeep & Sukanto, 2006; Tang et al., 2020). Specifically, Jones (2017) finds that *new age* statistical learning models, i.e., ML-models, are better on three factors: (1) they are better predictors of financial distress than other classifiers both on cross-sectional and longitudinal test sets; (2) they are relatively easy to estimate and implement, e.g., requiring minimal work for data preparation, variable selection, and model architecture specification; and (3) that while the model architecture itself can be relatively complex there is a good level of interpretability through metrics such as *relative variable importances*.

While several other scholars have found ML-models to be good predictors generally, there is still a push in the academic literature to enhance ML-model interpretability (Hall & Gill, 2019; Lipton, 2018),

[10] Known as *information leakage* (David, 2019).
[11] See Büyüköztürk & Çokluk-Bökeoğlu (2008) and Tabachnick & Fidell (2000).

which – for certain models – can be unclear. However, as argued by Hyndman & Athanasopoulos (2018) on forecasting: depending on the circumstances, "the main concern may be only to predict what will happen, not to know why it happens". Similarly, Jones (2015) argue that the benefit of using complex nonlinear (and non-interpretable) classifiers should be improved predictive powers over simpler models (p. 73). Both Jones (2015), Hyndman & Athanasopoulos (2018), and other scholars propose that if easily-interpretable models have comparable results to more complex models, the simpler and more parsimonious method should be utilized.

3.3 Financial Distress Prediction in Denmark

Scholars throughout the world have successfully applied various forms of ML-models due to their seemingly predictive superiority over traditional statistical models (see Aziz & Dar, 2006 for a historical overview), notable examples include Tang et al. (2020) and Sun et al. (2014, 2017) on Chinese companies, Jones et al. (2017) on American companies, Zięba et al. (2016) on Polish companies, and Christoffersen et al. (2018) and Matin et al. (2019) on Danish companies.

Compared to many other FDP-studies throughout the world that focus on large publicly traded companies, the Danish Business Authority provides the general public with access to a large database of financial statements from both listed and non-listed companies through the Danish Business Authority API (Virk.dk, 2020a).[12] In Denmark, both Christoffersen et al. (2018) and Matin et al. (2019) use this database[13] and prepare a dataset of financial statements from non-financial and non-holding companies, which includes 50 numerical financial ratios. In addition, Matin et al. (2019) use textual data from auditors' reports and managements' statements available in financial statements.

Both Christoffersen et al. (2018) and Matin et al. (2019) utilize the same dataset. However, due to the different methodological deliberations on the inclusion of textual data, Christoffersen et al. (2018) use a dataset spanning from 2003 to 2016, encompassing 1.3 million financial statements from 198,929 unique companies, of which 43,674 entered into a distress period at least once (p. 12). Matin et al. (2019) use financial statements from Danish non-financial and non-holding companies, but filter the data on the period from 2013 to 2016 to include text data from auditors and management, which is not available digitally prior to 2013. The latter dataset then encompasses 278,047 financial statements from 112,974 unique companies with 8,033 distresses (p. 201). Both find that the ML-model *gradient boosted trees* perform better than benchmarks. Matin et al. (2019) additionally find that a neural network that includes auditor reports has better prediction power than gradient boosted trees with purely financial ratios.

[12] See Section 5.1 Dataset Description and https://datacvr.virk.dk/data/
[13] However, rather than using the public API, both papers use cleansed and extracted data provided by Bisnode and Experian.

3.4 Company Ownership Default Risk

The predictive value of incorporating companies' current owners' previous company bankruptcies in FDP-models seem to be an underdeveloped point in the academic literature, e.g., the impact of *serial failers* that are repeatedly involved in (or perhaps even cause) company bankruptcies. To the best of the authors knowledge, there have not been any studies on this area. However, there have been "serial" bankruptcy studies on a company-level, e.g., the post-bankruptcy performance of reorganized companies (Hotchkiss, 1995). There have similarly been numerous studies on corporate governance and the impact of ownership concentration on company performance (Daily & Dalton, 1994a, 1994b; Deng & Wang, 2006; Donker et al., 2009; Lajili & Zéghal, 2010; Mangena & Chamisa, 2008; Manzaneque et al., 2016).

3.5 Financial Distress Prediction in Practice

From a collaboration between the authors and Nordea, it is clear that financial distress prediction (and more generally credit scoring) are used extensively in the practical world as well. Nordea – and presumably banks overall – have credit scoring as an integral part of their business and, as a result, developed it as an integrated process. The following briefly and superficially[14] outlines the workings of an in-house credit analysis tool used at Nordea, which to some extent is assumed to generalize to other banks.

Nordea's in-house credit analysis tools for assessing the risk of companies going into financial destress uses publicly available financial company data provided by a vendor. Most of this data is equivalent to the information contained in the database from the Danish Business Authority (Appendix 11.1, 13:20). Despite the fact that most of the data acquired comes from financial statements, Nordea relies on qualitative data as well, which could potentially include information on whether the borrowers have defaulted before, years of experience of the board or owners, previous success stories, attitude, or any other type of qualitative information. However, the content of the qualitative information provided is unknown to the authors and could encompass other aspects entirely.

Regardless, this qualitative aspect indicates that non-financial data is used in a qualitative manner to assign credit scores and calculate the probability of financial distress – or more specifically, loan defaulting. However, Nordea stresses that the qualitative aspect only constitutes a small part of the full risk score, and that the primary focus is put on key quantitative financial ratios. In the case of Nordea, the model does not produce direct probabilities[15] like several models developed in the academic literature, but instead provides a credit grade ranging from zero to seven. Here, it is indicated that the

[14] Superficial largely due to proprietary information that Nordea could not disclose.
[15] At least not for the end-user.

qualitative data at most impact the quantitative score by one grade (Appendix 11.1, 13:20; 16:50). In the case of Nordea, the credit scoring process appears relatively streamlined. Consequently, the development of better FDP-models could potentially be easily implemented in banks in general, showing a certain transferability of academically developed models to practical processes.

3.6 Relation to the Thesis

As outlined above, financial distress, and FDP specifically, has received much interest for more than half a century. This thesis includes three benchmark FDP-models from the academic literature: an LDA-model, a conditional probability model (logistic regression), and an ML-model. Specifically, a re-trained Altman Z-model is included due to its long-standing popularity both academically and from practitioners. Despite its later decrease in popularity (Dimitras et al., 1996), it is frequently used a baseline-model (Altman & Narayanan, 1997; Balcaen & Ooghe, 2006, p. 64). The logistic regression (LR) model is included as the conditional probability model due to its (general) predictive superiority over LDA-models, its general usage in banks and financial institutions today,[16] and its historically high academic interest (Aziz & Dar, 2006). Lastly, this paper incorporates an ML-model with gradient boosted trees due to both its novelty and general predictive superiority over previous FDP-models (Tsai et al., 2014). Additionally, gradient boosted trees have been successfully applied in a Danish context for FDP in Christoffersen et al. (2018) and Matin et al. (2019).[17]

As this paper investigates the potential increased predictive ability of including a co*mpany ownership default risk* (CODR)-variable[18] in FDP of Danish companies, it uses the methodological FDP-considerations in Christoffersen et al. (2018) as the academic foundation – and to some extent Matin et al. (2019). Specifically, this paper uses a similar dataset with a similar set of financial ratios as Christoffersen et al. (2018), but with CODR added as a feature.

[16] Nordea alluded to the use of logistic regression for credit scoring, but it was not be confirmed as it is proprietary information.
[17] While Christoffersen et al. (2018) compare gradient boosted trees to statistical models, Matin et al. (2019) use gradient boosted trees as a benchmark to their convolutional recurrent neural network.
[18] See section 6.2.3.1 on page 42 for an introduction to CODR.

4 THEORY

This section introduces the theories that form for foundation for the later sections. Specifically, three models are introduced, i.e., *linear discriminant analysis* (LDA), *logistic regression* (LR), and *gradient boosted trees* (GBT). Then, the process of hyper-parameter tuning is introduced, followed by a section on model scoring.

4.1 A BRIEF INTRODUCTION TO MACHINE LEARNING

Geron (2017) defines machine learning as "the science (and art) of programming computers so they can learn from data". This very simple definition gives the notion and idea of where the name "machine learning" origins. While this provides a general introduction to machine learning, this thesis uses a more specific definition given by Borovcnik et al. (2012), i.e., "a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data", since this definition better explains the inner workings of ML and its applications.

Machine learning can generally be categorized into four distinct subsets: *supervised*, *unsupervised*, *semi-supervised*, and *reinforcement* learning. In the following, we introduce supervised learning based on its relevancy to the thesis and refer to Geron (2017) for an introduction to the other approaches (for a brief introduction to unsupervised learning, see Appendix 2).

4.1.1 SUPERVISED LEARNING

Supervised machine learning is the subset of machine learning in which models are trained using *known* outcomes. In machine learning, this *outcome* is known as the *target* or the *label*. The target can be both continuous (as in predicting the revenue of a company) or categorical (as in *financial distress* or *no financial distress*). As an example, Figure 2 outlines a snippet of the dataset used in this thesis, showing six predictor features (columns) for five annual statements and a feature with the target values, where 1 represents *financial distress* and 0 *no financial distress*. Thus, the company represented in the third row went bankrupt within two years from the date of publication of this financial statement.

ProfitLoss	NoncurrentAssets	CurrentAssets	Assets	ContributedCapital	RetainedEarnings	Target
-10499000.00	115242000.00	5690000.00	120932000.00	250000.00	17330000.00	0
-5069228.00	117850520.00	956125.00	118806645.00	250000.00	12260979.00	0
2464711.00	117004425.00	1676221.00	118680646.00	250000.00	14725690.00	1
-10665079.00	115938617.00	937368.00	116875985.00	250000.00	4060611.00	0
-1714640.00	42467709.00	5908175.00	48375884.00	250000.00	2345971.00	0

Figure 2 – Example data for supervised machine learning

Supervised machine learning is either a *classification* task or a *regression* task. Classification is the task of classifying a data point into exactly one pre-defined *class*,[19] e.g., *financial distress/no financial distress*, but can also be expanded to multi-class classifications e.g. in the case of classifying an industry, e.g., *retail/insurance/agriculture*. Regression is the task of predicting a continuous value. This thesis uses classification as the target variable belongs to exactly one of two classes.

Supervised models learn by tuning their parameters according a given *objective function*. A model's parameters are the internal variables of a model that, when adjusted, will change the behavior of the model. Typically, the objective function holds a *loss function* and a *regularization term*, although the latter is often not used. The loss function determines the penalty that is given to an instance when fitting a model, based on the errors that the fit creates. In order not to *overfit* the model, a regularization term can be added in the objective function, which penalizes complex models and lead to the creation of simpler models (Fawcett & Provost, 2013). The goal of the machine learning model is then to optimize (maximize or minimize) the objective function by changing the internal parameters, known as the *training* phase. Once a model has been trained, i.e., once the objective function is optimized, the trained model can then use the "learned" patterns to predict the label of a new set of data. To ensure proper training and test the ability of a model to generalize, the model is *tested* on new and unseen data, and its performance measured by comparing the predictions to the actual labels. Figure 3 shows the training and the test phase.

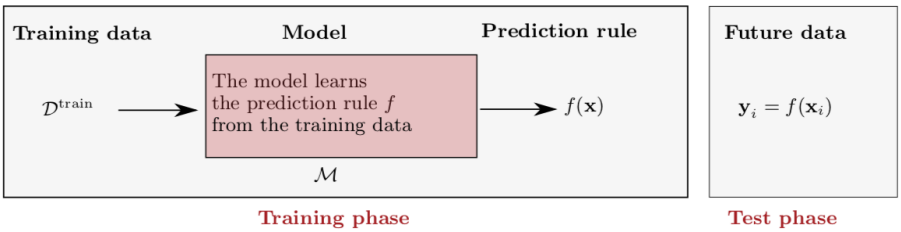

Figure 3 – A visualization of the *training* and the *test* phase that supervised machine learning models undergo, from Herlau et al. (2018).

In the training phase, the model takes training data as input, it trains the model using the objective function, and then it returns a fitted model. In the test phase, the model is then tested on new unseen data and is then evaluated using the preferred scoring metric.

[19] M*ulti-label* classification enables data points to be classified into more than one class.

4.2 MODELS

4.2.1 LINEAR DISCRIMINANT ANALYSIS

Linear discriminant analysis (LDA) is a method historically used for classification, which the Altman Z-score for financial distress prediction is developed on (Altman, 1968; Aziz & Dar, 2006). For two classes, LDA reduces the feature space of a dataset into a single line. On this line, a threshold can be specified where data points above the threshold are classified into one particular class and data points below the threshold to another class. As put by Tan et al. (2006), the purpose of LDA is to find "a linear projection of the data that produces the greatest discrimination between objects that belong to different classes". As an example, imagine a dataset consisting of two classes as outlined by the two circles in Figure 4 below.

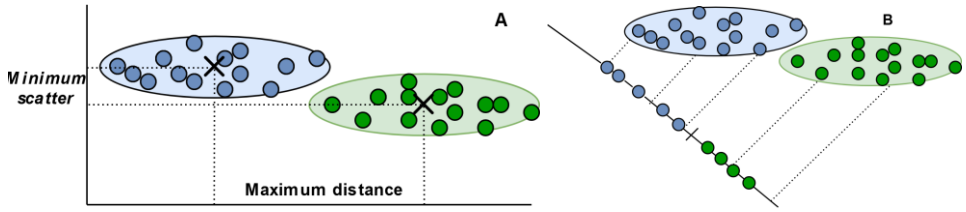

Figure 4 – LDA example

The two classes are projected following the dashed lines. Figure 4a shows two projections: onto the line with maximum distance between the means and onto the line with minimum scatter. Figure 4b shows a better projection with discriminatory power that minimizes scatter while maximizing the distance between the class means. Note the threshold on the prediction line

In Figure 4, there are two classes indicated by the two oval circles plotted using two features from the dataset. The objective of LDA is to find a line that, when all the data points are projected directly onto it, maximizes the distance between the *means* of the two classes and minimizes the *scatter* within each class. This *discriminant* is visualized in Figure 4b. More formally, LDA seeks to maximize the following for classes i and j:

$$\text{maximize } \frac{(\mu_i - \mu_j)^2}{s_i^2 + s_j^2} \tag{2}$$

Where μ_i is the mean of class i, and s_i^2 is the scatter of class i, i.e. for a given class:

$$\text{scatter} = \sum_{i=1}^{N} (x_i - \mu)^2 \tag{3}$$

Where N is the number of samples, μ is the class mean, and x_i is the projected value of data point i. Once the line that maximizes equation 2 has been found, the projected line can be described formulaically using the original features (like the Altman Z-score formulation), then a threshold can be specified for classification purposes, which is represented by smaller solid line perpendicular to the linear discriminant in Figure 4b above.

4.2.2 LOGISTIC REGRESSION

Logistic regression (LR) is a conditional probability model and is one of the best-known classifiers. It is widely used due to its simplicity and interpretability. LR is an extended version of linear regression that produces probabilities, which can be used for classification. To explain the relation and benefits of using LR for financial distress classification over linear regression, consider the *linear probability model* in Figure 5 below (a linear probability model is a linear regression where the dependent variable takes the value 0 or 1).

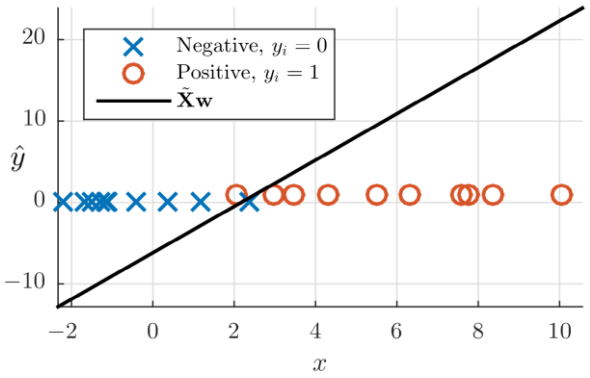

Figure 5 – Linear probability model, from Herlau et al. (2018)

As visualized above, the linear probability model seems to be able to differentiate between the two classes, *negative* and *positive*. However, the regression line far exceeds the range from 0 to 1, which entails that it cannot be used for probabilities since probabilities should range from 0 to 1. In fact, the linear probability model can produce results from $-\infty$ to ∞ linearly which is undesired. In comparison, LR produces values between 0 and 1 (see Figure 6).

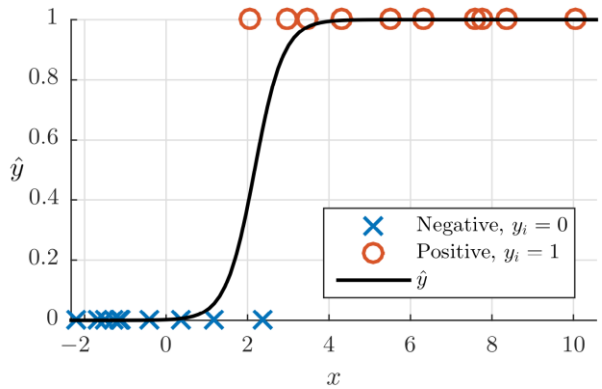

Figure 6 – Logistic regression, from Herlau et al. (2018)

In order to "squeeze" the output range from $[-\infty, \infty]$ to $[0,1]$, LR uses the following *sigmoid* function.

$$p(y = 1|x) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

Where $p(y = 1|x)$ is the output probability that x belongs to class $y = 1$. Thus, the sigmoid function, $\sigma(z)$, converts any input z in the range $[-\infty, \infty]$ to $[0,1]$. Here z is defined as

$$z = \log \frac{p(x|y = 1)p(y = 1)}{p(x|y = 0)p(y = 0)} \tag{5}$$

Which should be read as the *log-odds* that a data point x belongs to class 1. The above relies on Bayes Theorem, which is outside the scope of this section (interested readers are referred to Herlau et al. (2018)). Since the sigmoid function $\sigma(z)$ converts the log odds that a data point, x, belongs to class 1, $y = 1$, into a probability between 0 and 1, a classification threshold can be used to classify data points. For a threshold t, the model will classify data point x as the predicted class $\hat{y}$ using the following logic.

$$\hat{y} = \begin{cases} 1 \text{ if } p(y = 1|x) \geq t \\ 0 \text{ otherwise} \end{cases}$$

However, in order to classify different data points, the model must be trained first. The LR is trained by minimizing the cost c based on the model weights θ. For one sample the cost is defined as

$$c(\theta) = \begin{cases} -\log\big(p(y = 1|x)\big) & y = 1 \\ -\log\big(1 - p(y = 1|x)\big) & y = 0 \end{cases}$$

As an example, if the true label of x_1 is 1 but the predicted probability $p(y = 1|x_1) = 0.2$. The cost of this prediction is $c(\theta) = -\log(0.2) \approx 0.7$. The cost is then averaged over all instances to find the overall cost of the weights. This cost is then calculated for different sets of weights, and the weights that minimize the cost are chosen.[20]

4.2.3 Gradient Boosted Trees

Compared to LR and LDA that are *single-model* classifiers, *gradient boosted trees* (GBT) is an *ensemble* of *decision tree* classifiers. Before introducing the GBT-model itself, important parts that make up GBT are introduced, including decision trees, ensemble learning, boosting, and lastly the variant of GBT used in this thesis, XGBoost.

4.2.3.1 Decision trees

A decision tree follows a *divide and conquer* approach in a tree-like structure with the objective to maximize *class purity* in *leaf nodes* for classification purposes.[21] To illustrate the model, consider Figure 7 below.

[20] In statistics this is known as the maximum likelihood estimate
[21] Decision trees can also be used for regression tasks, we refer to Han et al. (2012).

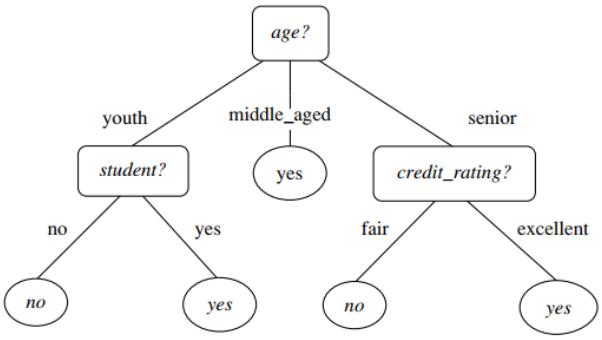

Figure 7 – Decision Tree Example, from Han et al. (2012)

Decision tree on whether a customer is likely to purchase a computer at a retail store.

Here the objective is to classify whether a customer in a retail store will purchase a new computer or not, based on Boolean logic (yes/no answers), e.g., whether the customer is *youth/middle-aged/senior*, *student/non-student*, or has an *excellent/fair credit rating*. In this decision tree, all customers start at the *root node*, i.e., age (the later paragraphs outline how the structure of the tree is established). At the root node each customer is evaluated based on this single criterion. Since all customers that are *middle aged* purchase computers (meaning that the resultant node is *pure*), the tree terminates at the *leaf node* and all customers that followed this decision path, are classified as *yes* (likely to purchase a new computer). For the other customers, however, the path continues until a potential pure leaf is reached. All paths result in a leaf node (a classification), but it is quite likely that not all leaf nodes are pure.

Decision trees are constructed such that any given *split* seeks to maximize the *purity gain* of the resulting nodes. First, both the feature of the root node and the corresponding split of this feature is decided. This decision is based on two factors: (1) how pure the resulting classes are (maximizing purity of the resultant nodes) and (2) how balanced the question is (maintaining balanced subsets, so the split is not too specific). Then each subsequent node is decided on the next-best split, third-best split, etc. in a recursive manner until a stopping condition is reached or when the purity of the resulting nodes cannot be improved anymore.

More formally, the impurity, I, of the dataset at the root, r, is calculated, I_r. Then, the impurity, I, of a split on feature k and threshold t_k is calculated as

$$I(k, t_k) = \frac{m_{left}}{m} I_{left} + \frac{m_{right}}{m} I_{right} \tag{6}$$

Where m is the total instances used for the current split, $m_{left/right}$ is the instances in the left/right nodes after the split, and $I_{left/right}$ is the impurity of the left/right nodes. Comparing the impurity before the split with the impurity after the split enables the calculation of the *purity gain* Δ, which decision trees seek to maximize. It can be formulated as

$$\text{maximize } \Delta = I_r - I(k, t_k) \tag{7}$$

Then, once the purity gain has been maximized, the decision tree is split into the corresponding nodes, where each node now acts as root nodes from which a new purity gain is considered.

There are several impurity measures that can be employed for different purposes. The most common methods for measuring the impurity are *Gini* and *Entropy*, of which the following outlines the former. Formally, the Gini of a node i is formulated as

$$G_i = 1 - \sum_{c=1}^{n} p_{i,c}^2 \tag{8}$$

Where $p_{i,c}$ is the ratio of instances of class c in node i. As an example, consider the following (left) node with a total of seven instances, with six instances of the class *financially distressed* and one instance of the class *not financially distressed.* The Gini impurity of this node is calculated as

$$G_{left} = I_{left} = 1 - \left(\frac{6}{7}^2 + \frac{1}{7}^2\right) \approx 0.24 \tag{9}$$

If the other (right) resultant node included five *financially distressed* and five *non financially distressed* companies, a total of 17 instances have been split into the left and right nodes. Then, calculating the Gini impurity, $G_{right} = I_{right}$ gives the following resultant Gini impurity of the overall split

$$I(k, t_k) = \frac{7}{17} * 0.24 + \frac{10}{17} * 0.50 \approx 0.39 \tag{10}$$

Calculating the Gini purity gain, Δ using the root Gini impurity, I_r gives the following

$$\Delta = I_r - I(k, t_k) = \left[1 - \left(\frac{11}{17}\right)^2 + \left(\frac{6}{17}\right)^2\right] - 0.39 \approx 0.71 - 0.39 \approx 0.31 \tag{11}$$

Thus, the purity gain for this split is $\Delta \approx 0.31$. If this split maximizes the Gini purity gain considering all features and thresholds, the split is created and recursively done so for the subsequent nodes.

Following the above logic, a decision tree can be built, trained, and used for prediction of new data. Compared to other ML-models, decision trees are considered *white box models* as the level of interpretability is high (Pedregosa et al., 2011). Specifically, the prediction of new data samples is based on Boolean logic in splits that clearly indicate how the label for a given data sample is predicted.

4.2.3.2 Ensemble Learning

The concept of ensemble learning comes from the idea that a group of predictors, called an *ensemble*, performs better than single predictors. One example of an ensemble is a *random forest*, which is a

collection of decision trees – each trained on random subsets of the training data. The decision trees are then combined into one predictor such that the majority vote of the individual decision trees is predicted. While random forest is a combination of the same type of classifier, ensembles can also be a combination of different types of models.

4.2.3.3 GRADIENT BOOSTING

One powerful technique within ensemble learning is the concept of *boosting*, where several *weak* learners (model that predict just slightly better than random guessing) are combined into one strong learner by training them *sequentially*. Here, sequential learning is the process of training a weak model, after which a subsequent predictor attempts to weakly adjust the incorrect predictions made by the first predictor, then a third predictor is added that adjusts errors made by the first two models, etc., until a sequential ensemble of weak learners is created. Sequential weak learning is computationally easy and therefore enables training many models.

There are different approaches to boosting. Two of the more common approaches are *AdaBoost and gradient boosting*, of which a variant of the latter is used in the GBT-model. Specifically, gradient boosting boosts the residual errors of the previous predictor (compared to AdaBoost that boosts weights). Specifically, for each model after the first weak learner, a predictor is fitted to the residual errors of all the previous models, and then added to the ensemble. Gradient boosting is often performed using decision trees as weak learners since they are computationally efficient, known as *boosted trees*.

To exemplify the process of boosted trees, consider Figure 8 below. Here, the top left corner illustrates the original data points and overlaid with the single trained decision tree on the green line. The predictions of this decision tree are then shown in the top right side on the red line (which are the same as the green fitted line to the left). Following this, on the middle left, a new weak decision tree is fitted on the residuals of the first tree, and when combined with the previous learner on the original data they produce the predictions on the middle right. Lastly, a third weak learner on the bottom left is fitted to the residual errors of the two previous sequential models, combined, and finally predicts the output on the bottom right.

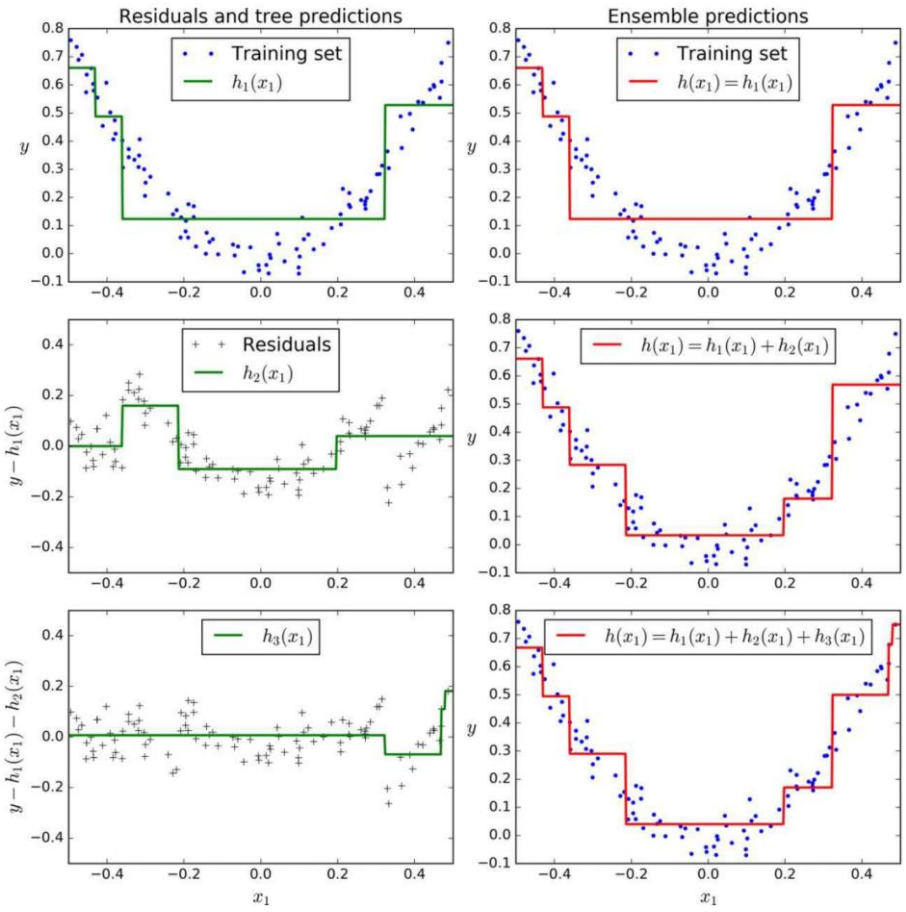

Figure 8 - Example of gradient boosting on decision tree regressors, from Geron (2017)

4.2.3.4 XGBOOST

XGBoost (XGB) is an acronym for *Extreme Gradient Boosting* developed by Chen & Guestrin (2016). As the name implies, XGB is a gradient boosted model that uses decision trees. The primary advantages of XGB over other models is its high execution speed and excellent performance, with over a factor 4 performance gain over comparable gradient boosted models (Chen & Guestrin, 2016). Consequently, it has been a top performer of various data science competitions for these reasons and due to a support for sparse datasets (missing values) and good imbalance handling (Brownlee, 2018). These features provide a solid foundation for XGB as an FDP-model due to large sparse datasets (which LDA and LR cannot handle) and an accented class imbalance between financially distressed companies and non-financially distressed companies. Further, there are several technical implementations in XGB that speed up computational performance, e.g., cache access patterns, data compression, sharding, etc., which are quite technical areas that are outside the scope of this thesis.

4.3 HYPER-PARAMETER OPTIMIZATION

Hyper-parameters are model parameters that are not directly learnt from data. Instead, hyper-parameters are specified prior to model estimation and decide *how* an ML-model should learn. For example, the hyper-parameters on decision trees include the maximum depth of a tree, the minimum number of

samples required to split an internal node, the maximum number of features to consider for a split, the function for measuring the quality of a split, etc. Hyper-parameters can have significant impact on the performance of a model, and it is therefore important to correctly *tune* these.

While hyper-parameter tuning is an important task, it is also non-trivial and usually requires a mixture of rules-of-thumb and trial-and-error approaches (Brownlee, 2019). Due to the (sometimes quite large) number of model configurations manual trial-and-error is infeasible and is instead usually performed using a (standard or random) *grid search* of the model parameters to find the best hyper-parameters.

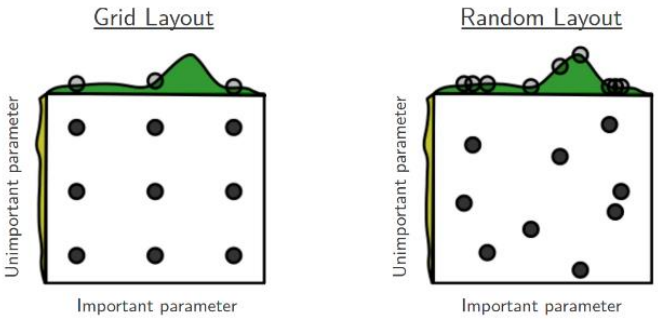

Figure 9 – Illustration of both standard (left) and random (right) grid search, from Bergstra & Bengio (2012)

Figure 9 (left) illustrates the concept a grid search over a two-dimensional hyper-parameter space. The green (top) and yellow (left) curves each illustrate the value of each hyper-parameter individually. In this illustration, the "green" parameter is considerably more important than the "yellow" parameter – however, the grid must be searched to find the peak of these curves as they are not known in advance. The figure also illustrates some of the drawbacks of using a standard grid search compared to a random search, i.e., for a standard grid search only a small subset of the individual hyper-parameter spaces is searched compared to a random search, as illustrated by the points on the curves in Figure 9.

4.4 K-FOLD CROSS VALIDATION

Before introducing cross-validation, the concept of train and test *splitting* is introduced.

Once a model has been trained with data, its performance should be tested on unseen data since model training and testing on the same data might lead to the model simply "repeating" the labels which it has already seen from the training phase while being unable to predict anything useful on new and unseen data (Pedregosa et al., 2011). This is known as *over-fitting*, which partly arises from the fact that some machine learning implementations can capture highly complex and non-linear patterns, which might lead to modelling of random noise in the training data. Instead, datasets are split into training and test partitions as illustrated in Figure 10 below to performance-test estimated models.

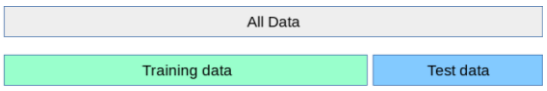

Figure 10 – Train and test split illustration, from Pedregosa et al. (2011)

However, due to the fact that a model's hyper-parameters must be tuned prior to estimation and later performance-tested on a test set, as outlined in section 4.3 above, there is a risk that the hyper-parameter tuning leads to over-fitting on the *test set*, since the hyper-parameters can be tweaked until optimal performance on the test set is reached (Pedregosa et al., 2011). In other words, the information from the test set is said to "leak" to the training phase, violating the requirements of testing model performance on new and unseen data. To combat this, the training set can be further split into *training* and *validation* sets to enable hyper-parameter tuning without information leakage. Once training has finished and hyper-parameters have been optimized on the validation set, performance can be evaluated on the unseen test data.

While this approach is valid, partitioning the dataset into three distinct sets does not allow for full training utilization of the data as the training data points are severely reduced. *K-fold cross-validation* combats this drastic sample reduction and removes the need for a distinct validation set. Instead, after a dataset has been split into train and test splits, the training data is used for cross-validation which entails splitting the training data into k smaller sets (see Figure 11), where 1 of the k folds is used as a validation set and the remaining $k - 1$ sets are used as training sets. This is repeated in *splits*, where each fold iteratively is used as a validation set while the remaining $k - 1$ folds are used as training sets. The model performance can then be averaged over all k parts to estimate how well the model will perform in the future. This process can then be repeated for every combination of hyper-parameters, i.e., combining (random) grid search with cross-validation.[22] Lastly, the best performing model with specified hyper-parameters is then usually re-estimated on the entire training set without cross-validation and subsequently tested on the unseen *test data* for the final model evaluation (Daume, 2017, p. 65).

[22] In sci-kit learn, this is implemented through *RandomizedSearchCV* and *GridSearchCV*.

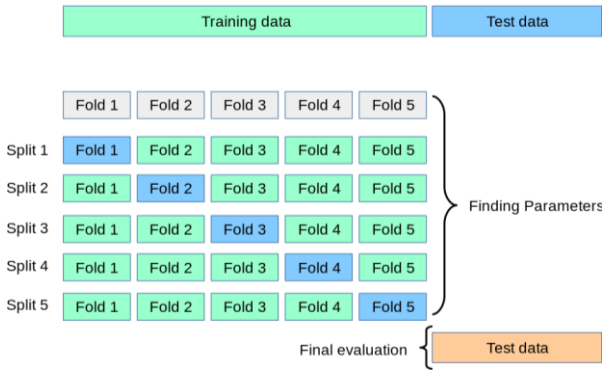

Figure 11 – Train and test split illustration, from Pedregosa et al. (2011)

4.5 SCORING

The following introduces the concept of *scoring*. While the above introduced the various models and their intricacies, their performances need to be evaluated using a suitable metric. There are many different metrics for evaluating classification (and regression) performance, which largely depend on the goal of the evaluation and whether the classes are balanced. This section visits the *confusion matrix* and the evaluation metrics *accuracy*, *F1-score*, *Receiver Operating Characteristics* (ROC), and *Area Under the Curve* (AUC).

4.5.1 CONFUSION MATRIX

While the confusion matrix itself is not an evaluation metrics, it is a useful tool for understanding the performance of a classification model, which provides a foundation for the later sections on evaluation metrics. It is built on four *building blocks* as illustrated in Figure 12 below (Han et al., 2012).

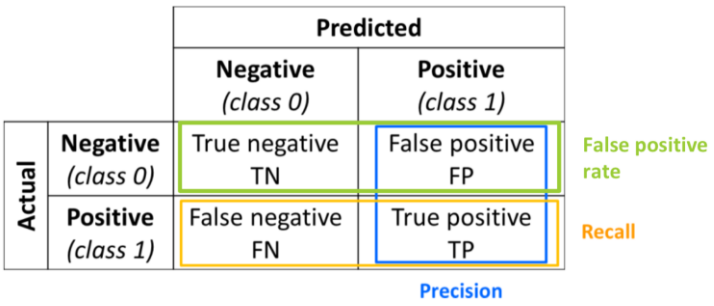

Figure 12 - Confusion Matrix

The four building blocks of the confusion matrix are the following:

- **True positive** *(TP)*: The number of observations that are classified as positive, and truly are positive.
- **False positive** *(FP)*: The number of observations that are classified as positive, but in fact are negative. Also known as a *type I error*.

- **True negative** *(TN)*: The number of observations that are classified as negative, and truly are negative.
- **False negative** *(FN)*: The number of observations that are classified as negative, but in fact are positive. Also known as a *type II error*.

The confusion matrix is commonly used as a tool for analyzing how well the model classifies the observations. A perfect model would have values only in the diagonal from the top left to the bottom right, with values only in the *true positive* and *true negative* cells. The confusion matrix provides a simple way to gauge the way in which a model misclassifies. Some of the more common metrics built from the confusion matrix are the following:

$$\textbf{Precision} = \frac{TP}{TP + FP} \qquad \textbf{Accuracy} = \frac{TN + TP}{TP + FP + TN + FN}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \qquad \textbf{Error rate} = \frac{FP + FN}{TP + FP + TN + FN}$$

Briefly, *precision* is the proportion of positive samples that are correctly classified, *accuracy* is defined below, *recall* is the proportion of positive samples that are classified as positive, and *error rate* is the misclassification rate, i.e., the proportion of the samples that have been classified incorrectly. Both the *accuracy* and *error rate* suffer from the same issues when dealing with imbalanced datasets, which are outlined below.

4.5.2 ACCURACY

One of the simpler evaluation metrics is *accuracy*, which simply is defined as the proportion of correctly classified samples. As outlined in the equation below, *accuracy* is the number of correctly classified samples (true positives and true negatives) divided by the total number of samples (both true and false positives and negatives), i.e.,

$$Accuracy = \frac{\#\ of\ correctly\ classified\ samples}{\#\ of\ samples} = \frac{TP + TN}{TP + FP + TN + FN} \qquad (12)$$

However, accuracy as a measure of model performance has several important limitations. First, for highly imbalanced data where one class is severely underrepresented, e.g., only 1% of all cases, a model that *always* predicts the majority class, has an accuracy of 99% despite being completely unable to classify the minority class. Consequently, it is a poor metric for imbalanced data. Second, the importance of correctly classifying one (e.g., the minority class) of the classes might be higher than correctly classifying the majority class, which accuracy does not consider. This is true for many cases, e.g., credit fraud, tumor classification, identification of financially distressed companies, etc. For these cases, respectively, it is presumably more important to capture all cases of fraudulent activity, malignant tumors, and financially distressed companies (*recall*) than it is to incorrectly categorize non-fraudulent activity as fraudulent, malignant tumors as benign, or financially distressed as healthy (*false positive*).

4.5.3 F-SCORE

The *F-score* (or F_1) is a measure that takes the harmonic mean of the precision and recall score and combines them into a single score.

$$\text{F-score} = \frac{2}{\frac{1}{\text{precesion}} + \frac{1}{\text{recall}}} \tag{13}$$

Since there tends to be an inverse relationship between precision and recall, where the increase in one normally results in a decrease in the other, the F-score is a commonly used substitute to the accuracy metric (Han et al., 2012).

4.5.4 ROC AND AUC

While both the *F-score* and *accuracy* are commonly used metrics for classification evaluation, they both suffer from the same limitation, i.e., they only evaluate models at a single threshold. What this means in practice is that samples with a probability higher than a pre-defined threshold (usually 0.5) are categorized as positive and negative otherwise. From these classification, accuracy, F-score, and other metrics can be calculated. However, as with the case of financially distressed companies, it could be more important to classify one specific class correctly compared to the other class, e.g., classifying a high number of financially distressed companies as financially distressed (high recall) despite potentially misclassifying a higher number of healthy companies as financially distressed (false positive rate, $FPR = \frac{FP}{FP+TN}$). To achieve this, the threshold can be lowered, which results in higher recall (and usually a higher FPR). To visualize the general performance of a classification model, the Receiver Operating Characteristic (ROC) is used to show the false positive rate and recall (also known as true positive rate) for all possible thresholds, as visualized in Figure 13 below.

Figure 13 – ROC Curve

The advantage of the ROC curve is that it can be used for a cost/benefit analysis to assign the appropriate threshold, where each type of classification (TP, TN, FP, and FN) can be assigned either a financial cost or benefit, e.g., the financial benefit identifying financially distressed companies (TP), the cost of misclassifying

healthy companies as financially distressed (FP), etc. (Han et al., 2012, p. 374). We can thus say that the ROC depicts the relative trade-off between cost and benefits of a given model at different thresholds. The closer the ROC curve is to the upper left corner the better, whereas a curve that follows the diagonal line represents a model that does no better than random guessing. The ROC curve is thus an intuitive visualization of the performance of a model at a range of thresholds, but it is not a scoring metric.

To convert the informative ROC curve into a scoring metrics, the *AUC* (Area Under the ROC Curve) can be calculated. As the name suggests, the AUC is simply the geometric area under the ROC curve. Visualized in Figure 13 above, it is calculated as the area under the ROC divided by the unit square ranging from 0-1, where 1 represents a perfect model (that touches the top left corner), 0.5 represents a random model (the diagonal), and 0 represents a model that never predicts the true label (a ROC that would touch the bottom right corner). Although AUC does not provide as much information as the ROC graph used for deciding on a suitable threshold, it is a useful metric to evaluate the overall performance of a classification model (Fawcett & Provost, 2013). Furthermore, another interpretation of AUC is the probability that a classifier ranks a randomly chosen positive observation higher than a randomly chosen negative observation, e.g., two financial distress prediction models with AUC scores of 0.71 and 0.74 indicate that the second model is 3 percentage points more likely to predict a higher distress probability for a random distressed firm than for a random non-distressed firm on average (Christoffersen et al., 2018).

5 DATA

5.1 DATASET DESCRIPTION

This paper utilizes two distinct databases from www.virk.dk that contain public data of all Danish companies. We have dubbed the two databases, the *permanent* database and the *financial statements* (FS) database to differentiate the two.

The permanent database contains fundamental company data, i.e., name, address, starting date, status (active, bankrupt, etc.), potential cessation date, legal corporate form (A/S, ApS sole proprietorships, etc.), industry code (NACE-codes[23]), contact information, number of employees, associated persons/businesses (owners, auditors, management, etc.), signing rules, registered capital, etc. Furthermore, the database contains all companies' full historical record of any changes in the above-mentioned features and the date of these changes, i.e., all registered changes in both dissolved and active companies (Virk.dk, 2020a).

The FS-database contains financial statements of all Danish companies legally required to disclose such information. The financial information is published in one or more of three formats, i.e., .pdf, scanned paper reports (usually .tiff), or .xml. The latter format, .xml, follows an XBRL-structure[24] and is the digitized and machine-readable format that this thesis uses. There is a legal requirement for companies to submit financial statements in both .pdf and .xml format. However, in the case of discrepancies between the file formats, the contents of the former prevail (Mygind, 2018b; Virk.dk, 2020b).

[23] See https://www.dst.dk/da/Statistik/dokumentation/nomenklaturer/dansk-branchekode-db07
[24] XBRL (eXtensible Business Reporting Language) is a standard for business and financial data (XBRL, 2020).

6 METHODOLOGY

6.1 PHILOSOPHY OF SCIENCE

The research design of this thesis follows the positivistic research paradigm, specifically *critical realism*. Using the hypothetico-deductive method proposed by Popper (1935), a research question is specified with a partially implicit hypothesis, i.e., that the inclusion of non-financial company information can impact the predictive power of FDP-models, which then is tested through empiricism. According to Guba (1991), this is a common way of conducting research in the positivistic paradigm where the ontological belief is that there is an objective truth and that it exists independently of the researchers studying it (Egholm, 2014; Guba, 1991). In the investigation of predictors for financial distress, the authors believe that there exist causal relations and patterns in quantitative data that may explain the nature of why companies get financially distressed, and that these relations exist independently of our research into the subject. Thus, this thesis seeks to investigate the objective truth of companies' financial behavior through machine learning. Following the epistemological method of critical realism, the authors believe that evidence can support *a priori* hypothesis, but cannot fully confirm them.

With the thoughts of Guba (1991) on the importance of experiments, this thesis seeks to conduct research in an objective manner and with reproducible experiments. In the research for this thesis, large quantities of financial and non-financial data are gathered, and machine learning experiments are created to find relations in the data. We attempt to conduct value free research such as to distance our subjective observations from the research, limiting human bias in the results. Thus, we seek to let our results derive from data provided by limited Danish companies without introducing biased manipulations into the data, which could invalidate the findings. One example of attempting to conduct value free research and limit human bias is seen when randomly searching the hyper-parameter space for optimal hyper-parameters. In so doing, we ensure that the findings are as objective and reproducible as possible. Despite the effort to conduct value free research, human bias cannot be completely eliminated due to normative choices of models, parameters, and variables – all subjective decisions of the researcher.

6.2 DATA PIPELINE

As outlined in the data section, this paper utilizes two different data sources from The Danish Business Authority (Virk.dk, 2020b, 2020a); the *permanent* database containing fundamental business information and the *FS* database comprising financial statements.

Section 6.2 outlines the data pipeline, i.e., how the data is acquired, filtered, cleansed, parsed, stored, changed, and finally yields the proper data format ready for analysis. Specifically, it describes how the permanent data is retrieved, filtered, parsed, and output to the financial statements process, which then retrieves and parses the

relevant financial statements. Finally, this section explains how the outputs of each database are combined to prepare it for analysis. The data pipeline is illustrated in the flowchart in Figure 14 below.

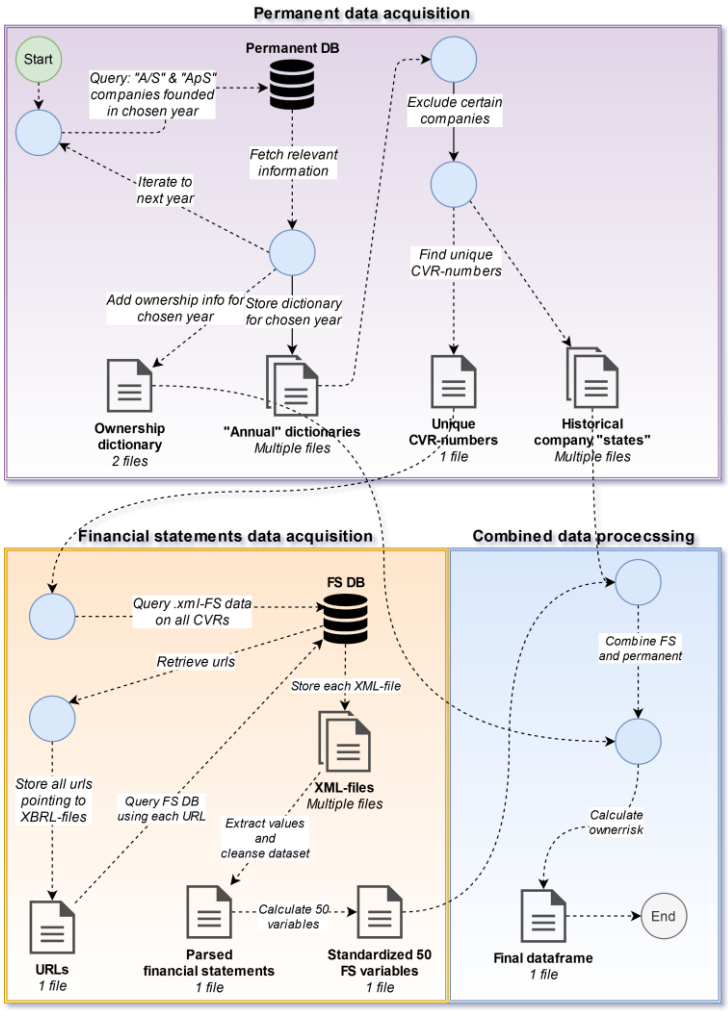

Figure 14 – Flowchart of the data pipeline from the databases (DB) to the final dataframe.

6.2.1 PERMANENT DATA

6.2.1.1 DATA ACQUISITION

To acquire the historical changes of all limited companies in Denmark, a script is created to query the permanent database from the Danish Business Authority (Virk.dk, 2020a) for information on all Danish limited companies, i.e., *A/S* and *ApS* companies. However, rather than querying and fetching *all* the data in the permanent database, the needed features are specified, including the CVR-number, owners, ownership shares, industry codes, municipality, legal form (A/S/ApS), status of the company (such as active and bankrupt), etc.[25] – including retrieving the changes in any of the specified features and the date of change.

[25] A full list of the features can be found in Appendix 3

To ensure speediness and smaller-scale testing, every iteration of a search only queries and fetches information on companies founded in a chosen year (starting from the first registered company in 1798 [26]). Once the information is acquired and stored, the script moves to the subsequent year, fetches the same information (if any), stores it, repeating this process until terminating after fetching information on all limited companies founded in 2020. Once completed, the script stores all information in dictionary-structures (json-files) in one file for each year on the hard-drive for parsing.

While creating and populating these "annual" dictionaries, another large dictionary is created that contains information on *all* owner-company relations from 1798 until today, i.e., a dictionary that comprises all ownership stakes held by any immediate owner (holding company, person, parent company, etc.). In so doing, it is possible to identify all current and past owner-company relationships for any given owner (usually represented by a holding company). As an illustration, Figure 15 below provides a snippet of the dictionary structure where we identify that owner "10000874" (PVC Holding[27]) owned 50 % of "31472512" (PC Ejendomme Hvalsø) from 2009 until 2016, from which point the owner took full ownership control of the company. Furthermore, we identify that PVC Holding acquired a 10 %-stake in "35380337" (Skjoldenæsholm Golfcenter) in 2013 and still holds that position (gyldigTil is null).


```
"10000874": {
        "CVR": {
            "31472512": [
                {
                    "Andel": "0.5",
                    "Periode": {
                        "gyldigFra": "2009-10-15",
                        "gyldigTil": "2016-04-30"
                    }
                },
                {
                    "Andel": "1.0",
                    "Periode": {
                        "gyldigFra": "2016-05-01",
                        "gyldigTil": null
                    }
                }
            ],
            "35380337": [
                {
                    "Andel": "0.1",
                    "Periode": {
                        "gyldigFra": "2013-10-01",
                        "gyldigTil": null
                    }
                }
            ]
        }
    }
```


Figure 15 – Snippet of the *owner-company relations* dictonary

[26] The company is *Aktieselskabet. Det kongelige octroierede almindelige. Brandassurance-Compagni* founded by Christian VII of Denmark. CVR: 63095818. History of the company: https://dis-danmark.dk/bibliotek/907080.pdf

[27] As a legal requirement, the authors must inform that this company is protected against unsolicited advertising, which we ask the reader to observe (see https://datacvr.virk.dk/data/node/178)

6.2.1.2 DATA PROCESSING

This section outlines the processing steps taking place after acquiring the historical information on all Danish limited companies and their owner-company relations.

First, certain companies are excluded. Similar to existing literature on financial distress prediction, financial companies and holding companies are excluded due to their differing asset structure (Christoffersen et al., 2018; Jackson & Wood, 2013; Matin et al., 2019). Furthermore, financial companies have different accounting standards (Christoffersen et al., 2018). We further exclude companies that have unknown industry codes as spot tests reveal that these companies have a much higher likelihood of erroneous numbers when cross-referencing the scanned annual reports. They are also often simply mislabeled holding companies. The exclusion method is outlined in Table 1 below.

Exclusion type	**Exclude if one of the following is true**
Unknown industry code	[Industry code] is 999999
Financial companies	[Industry code] begins with either "64", "65", or "66"[28]
Holding companies	[Company name] contain the name "holding"

Table 1 – Company exclusion table

Once these companies are filtered out, the remaining companies are parsed. For each json-file that contains information on companies founded in one specific year, the entire historical record of those companies is parsed from a raw dictionary format (see Appendix 4), which resembles how the data is stored in the permanent database, to a tabular format as seen in Table 2 below. Every row in the matrix thus represents a *state* that the company has been in and the date of that state change. As such, all rows for a given company represent *all* states that this specific company has ever been in with respect to the queried features.[29]

Municipality	**Industry Code**	**CVR**	**Name**	**Status**	**Legal Form**	**Date**
657	511900	53399428	H. Pries-Jensen A/S	Normal	A/S	26/02/1930
657	511900	53399428	H. Pries-Jensen A/S	Under bankruptcy	A/S	08/10/1999
657	511900	53399428	H. Pries-Jensen A/S	Bankrupt	A/S	02/09/2003

Table 2 – Example of fundamental company information in a tabular format

Following these transformations, a list of all unique CVR-numbers is provided for the acquisition of financial statements.

[28] Following the NACE-codes outlined in the code repository: /Extras/From The Danish Business Authorities/CVR-Branchekoder.xlsx
[29] Thus, changes in company features that are out of scope for this paper (e.g., company name changes, changes in the board of directors, etc.) are not represented in this matrix.

6.2.2 FINANCIAL DATA

6.2.2.1 DATA ACQUISITION

As outlined in the flowchart above, the acquisition of the financial information occurs *after* the completion of the permanent data acquisition. The processes could run simultaneously, and unneeded companies could then simply be excluded post-merging. However, the stepwise approach is done to ensure that unnecessary financial statements (e.g., from holding companies) are not fetched, increase data acquisition speed, and reduce strain on the FS database.

The financial statements are acquired through a two-step process, interacting with the FS database on two separate occasions, explained in the following paragraphs. In the first step, the database is queried using the list of unique CVR-numbers. Here all URLs that point to .xml financial statements relating to a specific batch of CVR-numbers are saved. Once completed, the process is repeated for a new batch of CVR-numbers until all financial statements have been acquired. In the second step, all URLs acquired in the previous step are used to download all financial reports followed by a data extraction process of the key financial figures.

The database stores metadata on each company's financial statements and the financial statements themselves. The metadata includes information such as the period of reporting, a timestamp of when the information was last edited by The Danish Business Authority, the date of publishing the financial report, the type of report (annual report, quarterly report, final report of liquidation, etc.), the file format (.tiff (images), .pdf, .xml), if the report required revision (True, False), the CVR number of the company, and finally a URL pointing to the financial statement in an .xml format. An example of the data is shown in Figure 16 below.


```
'_source': {
    'indlaesningsId': None,
    'sagsNummer': '14-390.980',
    'regnskab': {
        'regnskabsperiode': {
            'slutDato': '2014-06-30',
            'startDato': '2013-07-01'
        }
    },
    'sidstOpdateret': '2014-12-18T23:00:00.000Z',
    'cvrNummer': 62816414,
    'dokumenter': [{
            'dokumentType': 'AARSRAPPORT',
            'dokumentMimeType': 'application/xml',
            'dokumentUrl': 'http://regnskaber.virk.dk/41461826/Y
3ZyLmRrOi8veGJybHMvWC03MDEyMTQ4OS0yMDE0MTAwOV8yMTE3MTFfMzIw.xml'
        }
    ],
    'regNummer': None,
    'indlaesningsTidspunkt': '2018-04-01T06:06:34.932Z',
    'offentliggoerelsesTidspunkt': '2014-12-18T23:00:00.000Z',
    'omgoerelse': False,
    'offentliggoerelsestype': 'regnskab'
}
```


Figure 16 - Example of financial statement metadata and URL to an .xml file

Several filters are applied when querying the database for the URLs to ensure that only relevant data is returned. First, the query is set to only return URLs pointing to .xml files as .pdf files and other formats are unreadable for the script. In August 2012, the Danish Business Authority submitted new guidelines on the digitalization of financial statements, resulting in a requirement for companies to upload financial statements electronically in an XBRL format (Erhvervsstyrelsen, 2015). Consequently, the acquired financial statements generally cover full accounting years from 2012 until today. However, since most financial statements also include key financial figures from past years, data from 2010 and 2011 is often acquired as well. Excluding .pdf files and solely acquiring digitized financial statements entails a significant data exclusion, which could be partly circumvented by using optical character recognition (OCR). However, this requires setting up a robust data-retrieval pipeline for .pdf files with its own validation system, which is outside the scope of this thesis.

The second filter is the exclusion of any financial statement that is not an annual report, e.g., quarterly reports and reports of liquidation. While the inclusion of quarterly reports might contain information relevant to financial distress prediction, solely using annual reports provides a certain standardized framework of managing financial data. Thus, each instance in the dataset covers an entire accounting period, which for specific industries and companies negates the impact of seasonality.

Once the list of URLs containing all relevant financial statements has been consolidated, the process of acquiring each .xml file is initiated using *asynchronous processing*[30] rather than fetching each single financial statement in a sequential manner. This allows for great speed efficiency when interacting with the FS database. In so doing, a significant speed improvement over a standard synchronous process is achieved, which reduces the time needed for fetching of financial statements from more than a week to approximately 30 hours.

6.2.2.2 Data Processing

Once all necessary data is acquired, an extensive feature extraction process of turning the .xml files into a format suitable for analysis is undertaken. Specifically, the values of each .xml file are extracted, cleansed, tested for errors, and parsed into a tabular format. The following describes the process in detail.

6.2.2.2.1 Feature extraction

Extracting the feature and value pairs requires two important steps, i.e., defining the features to extract and then extracting the feature-value pairs using a reference map to assign the extracted values to the relevant year. The first step is done prior to the data extraction, the second step during.

[30] The concept of asynchronous processing can be rather technical and might be out of scope for this thesis, so we will not dive into further detail about this. Instead, we refer to the code base and to https://realpython.com/async-io-python/

First, the required features are defined from the *International Financial Reporting Standards* (IFRS), which the XBRL-standard observes. In 2019, this standard covered 613 numerical features and 6,571 text features (International Accounting Standards Board, 2020). A list of selected variables can be found in Appendix 5.

Second, the feature-value pairs are extracted, and a reference map is created for each .xml file. Each .xml file is structured in the format shown in Figure 17 below. For the first row, `<` indicates the beginning of the metadata of the feature, `TaxExpense`, in the `contextRef` (reporting year), `c4` with currency, `u5`[31], followed by a `decimals` indicator on whether the value is stored in thousands, millions, or actual numbers. Then the metadata indicators end with `>`, followed by the actual value, `-30000`, and then ended by `</d:TaxExpense>` indicating the end of the feature-value pair and its metadata.

```
<d:TaxExpense contextRef="c4" unitRef="u5" decimals="0">-30000</d:TaxExpense>
<d:ProfitLoss contextRef="c1" unitRef="u5" decimals="0">200951</d:ProfitLoss>
<d:ProfitLoss contextRef="c4" unitRef="u5" decimals="0">401620</d:ProfitLoss>
```

Figure 17 - Snippet of .xml code with feature-value pairs and metadata

Once all feature-value pairs are extracted, the values are transformed using the metadata. For `decimals`, the value is simply multiplied accordingly.[32] If `unitRef` is different from `u5`, entailing usage of other currencies than *DKK*, the entire financial statement is discarded. This is done to ensure a coherence of Danish companies in the dataset without influence from foreign accounting standards. Consequently, companies whose operations are solely based outside of Denmark (and reported in any other currency) are excluded.

Each financial statement utilizes `contextRef` as a reference to an accounting year. However, the year-context mapping is not consistent between financial statements. Thus, for each financial statement, a unique context-year mapping is created using the reference mapping at the end of each .xml file illustrated in Figure 18 below (see Appendix 6 for an example of an context-year mapping). For this instance, all values referencing the context `c4` are coded as the period from *2010-07-01* to *2011-06-30*.

```
<context id="c4">
 <entity>
  <identifier scheme="http://www.dcca.dk/cvr">56208410</identifier>
 </entity>
<period>
 <startDate>2010-07-01</startDate>
 <endDate>2011-06-30</endDate>
</period>
```

Figure 18 – Snippet of context metadata in .xml code

[31] *u5* references DKK following the ISO 4217 currency codes: https://www.iso.org/iso-4217-currency-codes.html
[32] However, as the later sections will outline, several companies have misreported their financial figures by factors of thousands, millions, and sometimes billions.

The process of extracting feature-value pairs and mapping these to corresponding accounting periods is then repeated for each .xml file and stored in a large .json file.

6.2.2.2.2 DATA CLEANSING

Once fetched, the parsed financial statements must be cleansed considerably due to a large proportion of various errors in the .xml files. According to two Danish credit rating agencies, 25-33% of all Danish electronic financial statements might be erroneous (Bisnode, 2017; Mygind, 2018b, 2018a), e.g., incorrect CVR-numbers, more than one feature-value pair for *profit/loss for the year* that are conflicting, values off by several orders of magnitude, etc.

The data investigation reveals wrong usage of the decimal tag as evidenced by cross-checking .xml files and the corresponding .pdf files. These errors are assumed to arise from the interaction with the reporting software used to generate the .xml files using the XBRL-standard. To alleviate the issues arising from misreported data, several implementations are made to cleanse the data, which is described in more detail below. One example is financial values three orders of magnitude away (off by a factor 10^3) from the reported .pdf values. This can be seen in Table 3 below where the accounting year of 2017 suddenly saw a considerable increase in *Assets* and other financial information from the previous year and compared to the subsequent year.

Year	cvrNumber	OtherFinanceIncome	ProfitLoss	NoncurrentAssets	CurrentAssets	Assets	ContributedCapital	RetainedEarnings	Equity
2015	89998719	0.0	149488.0	6383118.0	22862280.0	29245398.0	1500000.0	9709709.0	11750287.(
2016	89998719	377.0	20103364.0	5101070.0	31064653.0	36165723.0	1500000.0	9678476.0	31853650.(
2017	89998719	2000000.0	19430000000.0	4612000000.0	88322000000.0	92934000000.0	1500000000.0	9551000000.0	31180000000.(
2018	89998719	3572000.0	2187000.0	215000.0	2936564.0	3151564.0	1500000.0	-116253.0	1383747.(
2019	89998719	41624.0	-566114.0	375000.0	3002723.0	3377723.0	1500000.0	-682367.0	817633.(

Table 3 – Example of a three orders of magnitude error

These errors are identified iteratively by screening every financial statement. At each row, the value of *Assets* is stored and compared with its value in the previous year and the subsequent year. *Assets* is chosen as the proxy for the decision on whether to de-scale since all companies report this value and since it is one of the values least prone to large yearly fluctuations (except when erroneous). However, as some companies experience extreme growth from year to year, the error detection allows for a growth in *Assets* up to a factor 100 increase. While a factor 100 might seem high, this ensures that novel growth companies are accurately modelled, e.g., a newly founded company could increase its total assets from DKK 50,000 up to 5,000,000 DKK in one year. Consequently, if the current Asset value is off by a (growth limit) factor of more than 100, all financial values are descaled by a factor 1,000, then *Assets* is re-checked, followed by another potential descaling of all values, continuing until the current *Asset* value is within limits of a growth limit factor of 100. While this presumably descales most companies to their true values, a small subset of extreme growth companies might be incorrectly scaled using this approach.

While the growth limit disallows more than a 100 factor increase in asset value, there is also a check on whether the asset value is divisible by 1,000 as this provides further evidence that the asset value has been increased by three orders of magnitude or more. Formally, these two checks are formulated as below, and if both are true, the financial values are divided by 1000.

$$Assets_{t-1} < (growthLimit * Assets_t) > Assets_{t+1} \textbf{ AND } Assets_t \bmod 1{,}000 = 0 \qquad (14)$$

While the financial values are most often multiplied by 1,000, there are also errors of much higher magnitudes, as illustrated in Table 4 with more than 15,000 financial statements with magnitude errors.

# of financial reports with magnitude errors		
Magnitude 9 (billion)	**Magnitude 6 (million)**	**Magnitude 3 (thousand)**
2	346	14,484

Table 4 - Distribution of magnitude errors in *Assets*

In addition to the magnitude errors outlined above, there are several other errors, e.g., when the parsed data does not contain essential financial information such as either *Assets* or *Liabilities*. These two values are essential for calculating the *company size* (the process is explained in the next paragraph). In total, this excludes 69,213 financial statements. Further, there are some values that are reported as negative that should be positive and vice-versa. However, the inconsistency of these within financial statements disallowed proper cleansing. As such, many of these errors still exist in the dataset, but spot tests indicate that the proportion of these errors is much smaller than magnitude error.

6.2.2.2.3 FINANCIAL RATIO CALCULATIONS

After the extraction and cleansing phase of the financial data, the data is restructured into 46 financial ratios[33], most of which are scaled by the company size. Furthermore, the ratios are winsorized at 5% and 95% quantiles to remove the impact of extreme outliers. Similar to Christoffersen et al. (2018) and Matin et al. (2019), company size is defined as the total debt of the firm when equity is negative (in absolute numbers) and total assets otherwise. In so doing, each financial report is standardized by the size of the company, which ensures that each financial statement can be generalized. Otherwise, the models could be heavily impacted by unstandardized financial data. As an example, consider the *debt*/*size* ratio with high predictive power: large corporations would create noise in the (unscaled) *debt* variable when their debt only constitutes a small

[33] This list of variables closely mirrors both Christoffersen et al. (2018) and Matin et al. (2019). However, some individual values could not be computed, e.g., due to data inconsistencies.

percentage of the capital structure, which lessens the generalizability of the model. The selected variables are listed in Appendix 7.

6.2.3 COMBINED DATA STRUCTURE

Once the fundamental historical company information, the financial ratios, and the owner-company relationships are acquired, they are all combined into one dataset. An illustration of the combined data structure is shown in Figure 19, which outlines how the financial ratios form the data foundation on which the fundamental company data is added, followed by the *Company Ownership Default Risk* (CODR) feature.

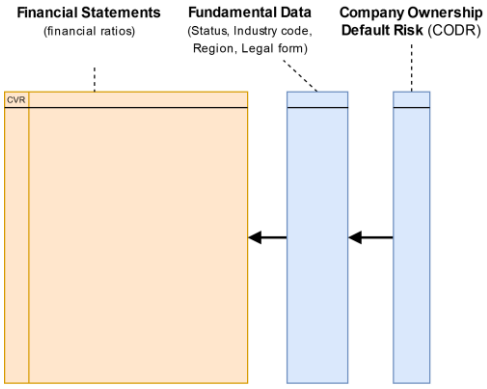

Figure 19 – Combined data structure

6.2.3.1 PERMANENT DATA

The information on industry code, municipality code, and legal form are added to each financial statement using the publication date of each financial statement. In the cases where the publication date is not available, six months are added to the end of the accounting period similar to the approach of Christoffersen et al. (2018) and Matin et al. (2019). Thus, the latest available information at the time of publication[34] is appended as features to the dataset. The same exercise is performed for the target variable, *financially distressed* or *not financially distressed*. However, rather than simply appending the *status* information at the time of publication, a window of two years from the date of publication to exactly two years later is created. Thus, if the given company has had one of the statuses: *Under compulsory dissolution*, *Dissolved after bankruptcy*, *Under bankruptcy*, *Compulsory dissolved* within this two-year period, the company receives the label 1 (financially distressed), otherwise it receives the label 0 (not financially distressed).

6.2.3.2 COMPANY OWNERSHIP DEFAULT RISK

Following the addition of the fundamental company information, including the target variable, the *Company Ownership Default Risk* (CODR) feature is added. To introduce the acronym of CODR, it is a method of

[34] This is done in order not to create information leakage from the future (David, 2019).

quantifying the risk (R) to a given company (C) that might arise from the current owners' (O) previous company defaults (D). These four aspects are then combined in the CODR-variable.

To calculate the CODR of company c at time t, the *ownership default risk* (ODR) of each individual owner o of company c at time t must be calculated first. Once the ODR of each owner o at time t has been calculated, these are then weighted by the owner's share in company c. In other words, CODR is a weighted average of each owner's ODR, weighted by that owner's ownership share (percentage) in company c. More formally,

$$\text{CODR}_{ct} = \sum_{o=1}^{N} \text{ownership share}_{cto} * \text{ODR}_{to} \tag{15}$$

Where the subscripts o, c, and t denote the 'owner', 'company', and 'time'. Thus, $\text{ownership share}_{cto}$ denotes the ownership share of owner o in company c at time t. Similarly, ODR_{to} denotes the ownership default risk of owner o at time t.

ODR can be thought of as the number of company defaults that a person (owner) has had up until now weighted by the ownership share of those companies, all divided by the total ownership shares held up till this point. More formally,

$$\text{ODR}_{to} = \frac{\sum_{s=1}^{N} \text{latest ownership share}_{tos} * \text{isDefaulted}_{tos}}{\sum_{s=1}^{N} \text{latest ownership share}_{tos}} \tag{16}$$

Where s denotes 'subsidiary'. Thus, $\text{latest ownership share}_{tos}$ represents the latest percentage ownership of held by owner o in subsidiary s at time t. isDefaulted_{tos} is a Boolean flag denoting whether subsidiary s has defaulted while owned (wholly or partially) by owner o at or any time before time t, i.e., the value is 1 if the subsidiary s defaulted and 0 otherwise. *Subsidiary* relates to any company other than c owned at or before time t by any of the owners at time t.

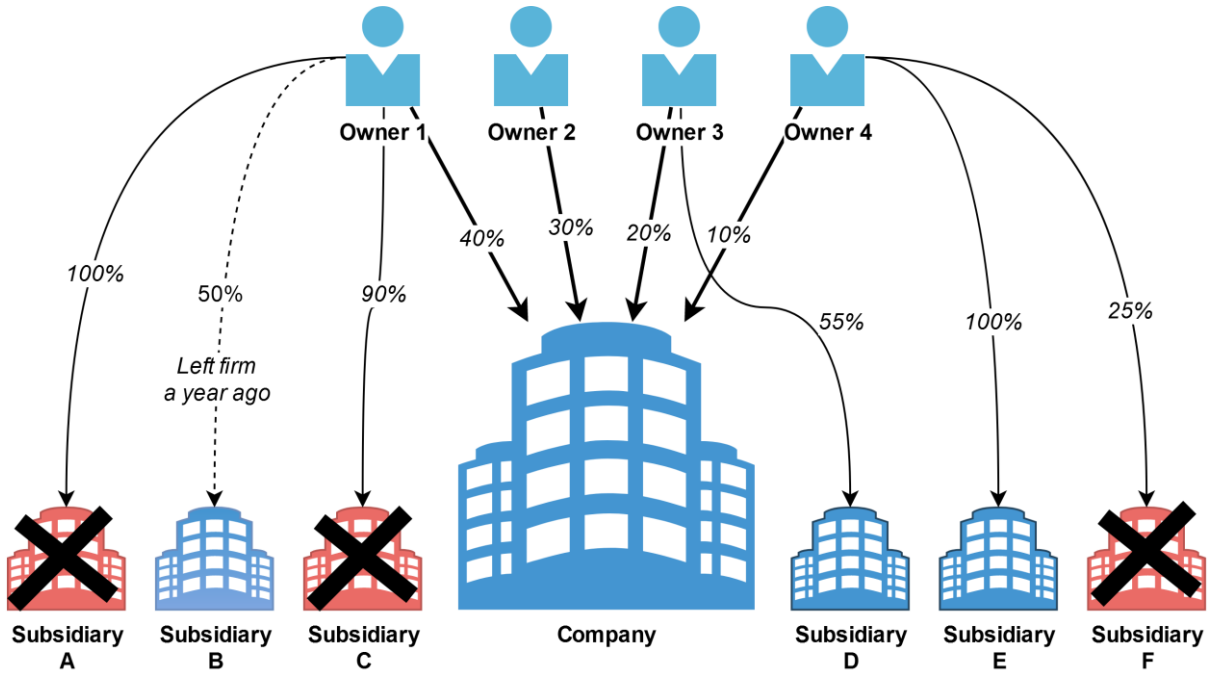

Figure 20 – Overview of company-ownership and owner-subsidiary relations

To give a practical example on CODR, consider the company in Figure 20 above. To calculate the CODR of this company today, the ODR of all owners (*owners 1-4*) must be calculated. To calculate the ODR of *Owner 1*, the number of defaults owner 1 has been a part of is counted, which is 2 (as subsidiary A and C defaulted when owner 1 was part of the organization), weigh each of these defaults by owner 2's ownership shares, which is $100\% * 1 + 90\% * 1 = 1.9$. Then divide the sum of the weighted defaults (1.9) by the latest held sum of ownership shares that owner 1 currently controls and has controlled, which is $A_{share} + B_{share} + C_{share} + Company_{share} = 100\% + 50\% + 90\% + 40\% = 2.8$. Note that while owner 1 does not own shares in subsidiary B anymore since he left the firm a year ago, the latest position of 50% is still used in the ODR-calculation as a representation of a "successful" exit.[35] Consequently, we calculate the ODR of owner 1

$$\text{ODR}_{owner\ 1} = \frac{(100\% * 1) + (50\% * 0) + (90\% * 1) + (40\% * 0)}{100\% + 50\% + 90\% + 40\%} = \frac{1.9}{2.8} \approx 0.68$$

As O*wner 2* has no previous or current positions other than the 40% in the company, the ODR of owner 2 is $\frac{40\%*0}{40\%} = 0$. Similarly, the ODR of *Owner 3* is $\frac{(20\%*0)+(55\%*0)}{20\%+55\%} = 0$ and the ODR of O*wner 4* is $\frac{(10\%*0)+(100\%*0)+(25\%*1)}{10\%+100\%+25\%} \approx 0.19$.

[35] "Success" should be understood quite narrowly as it simply refers to the absence of a company in financial distress.

Consequently, the current CODR for this company is $(40\% * 0.68) + (30\% * 0) + (20\% * 0) + (10\% * 0.19) \approx 0.29$.

6.2.4 LDA Implementations

As mentioned in Section 3.2, Altman (1968) uses five financial ratios in his analysis: *Working Capital/Total assets* (x_1), *Retained Earnings/Total Assets* (x_2), *Earnings Before Interest and Taxes/Total Assets* (x_3), *Market Value Equity/Book Value of Total Liabilities* (x_4), *and Sales/Total Assets* (x_5). Three of the five variables (x_1, x_2, x_3) can be created directly from the financial reports, but the latter two are only available for publicly listed companies. Altman (2017) instead proposes replacing *Market Value of Equity* (x_4) with *Book Value of Equity* when analyzing private companies. He also mentions the potential difficulty of obtaining *Sales* and therefore suggests replacing this value with just a fixed constant (Altman et al., 2017). These four ratios are then incorporated into the main dataset consisting of the 46 primary features. The LDA-implementation thus only uses these four features, whereas LR and GBT use all the other previously discussed features.

6.3 Data Analytics

As outlined in the previous sections, this thesis creates three financial distress prediction models, i.e., Linear Discriminant Analysis (LDA), Logistic Regression, (LR) and Gradient Boosted Trees (GBT). The methodology consists of three steps: (1) First, creating a baseline model using LDA that mirrors Altman's (1968) traditional linear discriminant using a combination of the original features and new suggestions as described in Section 6.2.4. Then, (2) the LR and GBT models, with and without a CODR-feature, are trained on the dense dataset (missing values excluded) described in Section 6.2.3. Finally, (3) the GBT-models, with and without the CODR-feature, are trained on the full sparse dataset (missing values included). These models are then evaluated in relation to each other. This section describes the process of finding the best models and covers hyper-parameter tuning, standardizing data, and ensuring reliable results using cross-validation. Note that, due to missing values, three distinct datasets will be used. LDA, that has a small feature space, trains on its own dense dataset with four "Altman Z"-variables. LR and GBT similarly train on a dense dataset, but with much more features than LDA. Finally, due to GBT's ability to handle missing values, GBT is trained on the full (sparse) dataset.

To avoid verbosity and be succinct when referring to the specific models, the following terminology is used when discussing the models in relation to each other, i.e., models trained on the large dataset with missing values are denoted as *sparse* whereas models trained on the smaller dataset with no missing values are denoted as *dense*. Further, to accent the difference between the models with a CODR-variable and those without, they are denoted with the suffix *CODR* or no suffix, respectively. As such, the GBT model trained on the large (sparse) dataset with missing values and which includes the CODR-variable is denoted as *sparse-GBT-CODR* and the related model without a CODR-variable is simply denoted *sparse-GBT*.

6.3.1 Linear Discriminant Analysis

The LDA-model is trained using the four variables obtained in Section 6.2.4. Altman's (1968) original Z-score model contained a set of estimated coefficients used for classification. Rather than using Altman's (1968) previously estimated coefficients from a very different business context, the LDA-model is re-trained to better generalize on Danish companies.

Before training the model on the four financial ratios, all data points with missing values are excluded, after which approximately 230,000 of 745,000 instances remain. Then, the data is split into training (75%) and test (25%) sets such that reliable test results can be produced. There is no need for hyper-parameter tuning since LDA offers no hyper-parameters that can be tuned.[36]

6.3.2 Dense Prediction

For LR, data instances with missing values must be excluded and the remaining values must be standardized. The first point on excluding values is simply due to an inability of the model to handle missing values, and the second point on standardization is recommended practice when performing LR (or ML generally) that includes a regularization term. Just prior to standardization, the data is split into training and test samples. After these, the hyper-parameters are tuned using cross validation on the training set to find the optimal model.

For the removal of missing values, i.e., making the sparse dataset dense, removing all missing values poses an issue as several of the features have more than 99% missing values. Consequently, the removal of the corresponding data instances would shrink the dataset to less than 1% of the original size. Thus, features that contain information in less than 60% of the instances (i.e., more than 40% missing values) are removed. Using this approach, 22 features are removed. Following this, the remaining data instances with at least one missing value are similarly removed, which results in a considerable reduction from 743,607 instances to 153,750, i.e., shrinking the vertical size of the dataset by 79%. After converting the dataset from a sparse to a dense dataset, the resultant data is split into a training set, consisting of 75% of the data, and a test set with the remaining 25%.

For the training and test set individually, each of the data features are standardized independently by subtracting each value by the mean and dividing by the feature's variance such that it scales to unit variance (Geron, 2017; Pedregosa et al., 2011). Standardizing is a standard procedure in machine learning that ensures a better input for the models recommended for models with a regularization term. Standardization is

[36] There are several hyper-parameters available, but these are mostly which *solver* to choose etc. It does thus not make sense to implement hyper-parameter tuning methods such as random search.

specifically performed as it is recommended for regularization used in logistic regression (Pedregosa et al., 2011).

6.3.2.1 *Logistic Regression*

For LR, random search is performed to find optimal hyper-parameters and cross-validation for training the model. Two hyper-parameters are specified, i.e., *class_weight* and *C*. The *class_weight* hyper-parameter allows the model to weigh the two target classes, 0 and 1, differently, which is especially useful for imbalanced datasets. This entails that errors are penalized differently, so that errors from the majority class are penalized less than errors from the minority class. Consequently, it is set to *balanced* and serves to re-balance the data. Then the *C* hyper-parameter, which is the inverse of regularization strength where lower values result in a higher regularization, will be found during random hyper-parameter optimization. There are no limit boundaries for on the value, but it is set to 1 by default. A log-uniform distribution ranging from 0.001 to 1,000 is created and the random search will then choose random *C* values from this distribution.

```
Parameters = {class_weight = "balanced",
              C = log-uniform(0.001, 1000)}
```

Figure 21 - Logistic Regression hyper-parameter settings

As a third hyper-parameter that is not model-specific, the scoring metric used to evaluate the models in the random cross-validation is set to *AUC* (*roc_auc*). After the hyper-parameter space is specified, 50 random iterations are run to get the optimal value of *C* and the best performing model[37]. The same steps are then repeated for the *LR-CODR* model, which returns a different model.

6.3.2.2 *Gradient Boosted Trees*

The XGBoost implementation of Gradient Boosted Trees (GBT) has seven hyper-parameters that are relevant for this thesis. Three of the hyper-parameters are specified prior to performing a random search to reduce complexity when finding the optimal set of hyper-parameters. Subsequently, a random search will optimize the remaining hyper-parameters. This task is done for both *dense-GBT* and *dense-GBT-CODR*. First, the *scale_pos_weight* parameter is set to re-balance the two imbalanced classes[38] using the ratio between the negative classes and positive classes, here 22.49. Following this, the optimal number of trees in the ensemble is estimated.

6.3.2.2.1 Number of trees

First, the optimal number of trees in the ensemble, given by the *n_estimators* hyper-parameter, is estimated. Specifically, the estimation is performed by calculating the AUC of different ensemble models using cross-

[37] It is infeasible to find the actual *optimal* value when running random search, but the estimated values could be close.
[38] For an overview of the hyper-parameters, see https://xgboost.readthedocs.io/en/latest/tutorials/param_tuning.html

validation with different numbers of trees on the training set. Once the AUC scores reach a plateau, indicating that no improvements are found by adding additional trees, an optimal number of trees is found. As shown in Figure 22 below, the model performance on both the training set and the test set (which is within the original training set) clearly indicates how the model progressively fits the training set at the expense of generalizability. From the figure, it seems that the model quickly reaches a stage of diminishing returns and converges on 11 trees based on the validation set.

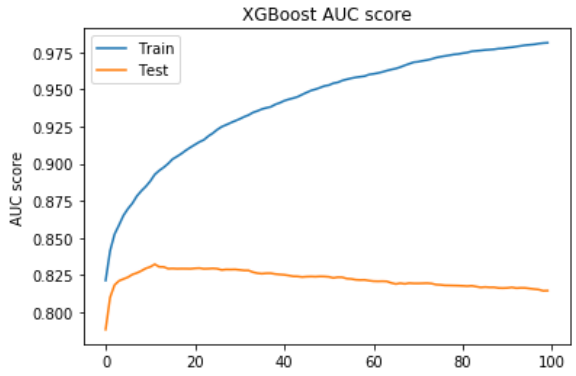

Figure 22 – Finding the optimal number of trees

6.3.2.2.2 MAX DEPTH

Following the optimal number of trees, the size of each tree (hyper-parameter *max_depth*) is found. The size relates to the maximum number of layers for each tree. It is important to find the right depth-balance since too shallow trees will perform too poorly and too deep trees tend to overfit. The optimal tree depth is found in a similar manner to the optimal number of trees, with the AUC scores shown in Figure 23. For each depth-level, the mean AUC-score is plotted along the curved line and vertical lines showing the maximum and minimum scores for each depth level. Here, there are indications that the optimal number of trees is 5.

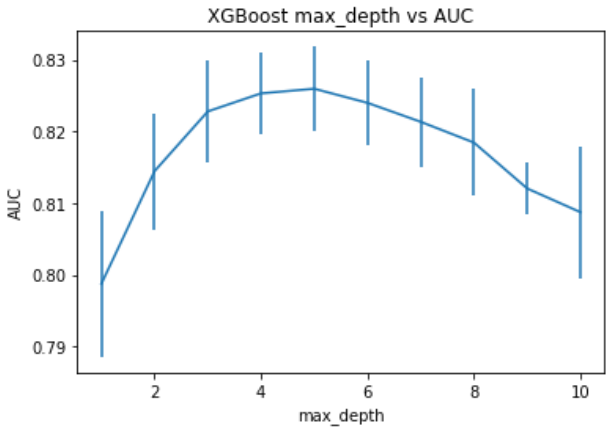

Figure 23 – Finding the optimal depth of the trees

6.3.2.2.3 RANDOM SEARCH

After these three hyper-parameters have been specified, the hyper-parameter search space has decreased considerably in complexity. Consequently, a random search of 20 iterations is conducted to find the remaining four hyper-parameters, i.e., *learning_rate, gamma, subsample, colsample_bytree*. The *learning_rate* (also called the shrinkage factor) specifies the effect of adding one more tree to the ensemble. As previously described, gradient boosted trees iteratively add trees to the ensemble where each tree attempts to correct the residual errors made by the previous trees. The learning rate applies a weighting on the corrections that every new tree makes and specifies the speed at which the model learns: too high and the optimal parameters might not be found, too low and the training will slow down considerably, which increases training time but also lead to a more fine-tuned model. The learning rate is set to range between 0.01-0.1. *Gamma* is a hyper-parameter that specifies the minimum loss (the highest gain from split purity) that is required to create one extra branch in a tree. The range is set to range between 0-5, where 0 is the default. *Subsample* defines the proportion of the training set that any given train is allowed to train on and is given as a ratio 0-1, where a subsample-size of 0.5 entails that each tree will be trained on a random half of the training set to make the model generalize better. It is recommended not to specify *subsample* at the extremes (Brownlee, 2018), thus the range is set to 0.3-0.8. The final hyper-parameter *colsample_bytree* is similar to the *subsample* feature, but rather than subsampling the instances (rows), it subsamples the features (columns) instead. This hyper-parameter is used for an entire tree, meaning one tree is only allowed to use the randomly sampled features, whereas the *subsample* hyper-parameter on instances, randomly subsamples the training data at each node. The feature subsample range is set to 0.8-1.0, which heightens the probability that any given tree always will have important features. The full set of hyper-parameter and their values is shown in Figure 24.

```
Parameters = {scale_pos_weight = 22.49,
              n_estimators = 11,
              max_depth = 5,
              learning_rate = range(0.01, 0.1),
              gamma = range(0, 5)
              subsamble = range(0.3, 0.8),
              coolsample_bytree = range(0.8, 1.0)}
```

Figure 24 - Hyper-parameter settings for the random search

6.3.3 SPARSE PREDICTION

The above section on *dense prediction* outlined the need to shrink the dataset to a dense format, which considerably reduces the available data and information contained within it. Instead, the following performs the same procedure as presented in Section 6.3.2.2 above, but with sparse data. However, only GBT is able to handle sparse data, excluding LR and LDA for this step. The hyper-parameter optimization follows the same approach and the same hyper-parameter space is chosen. However, due to the differences between the dataset, a new random search must be initialized.

6.3.3.1 Gradient Boosted Trees

Similar to above, the optimal values for *scale_pos_weight*, *n_estimators*, and *max_depth*, are found prior to random search. The best ratio for the *scale_pos_weight* hyper-parameter is calculated as 24.29 and is used for the remaining hyper-parameter optimization steps.

6.3.3.1.1 Number of trees

The optimal number of trees is found by iteratively evaluating the performance of the model at different numbers of trees in the ensemble. AUC is again used as the scoring metric. Here, the best performing model appears to include 50 trees. However, the performance on the validation (test) set quickly flattens out and fewer trees could presumably be used. Regardless, *n_estimators* is set to 50 to better enable the search for the best performing model.

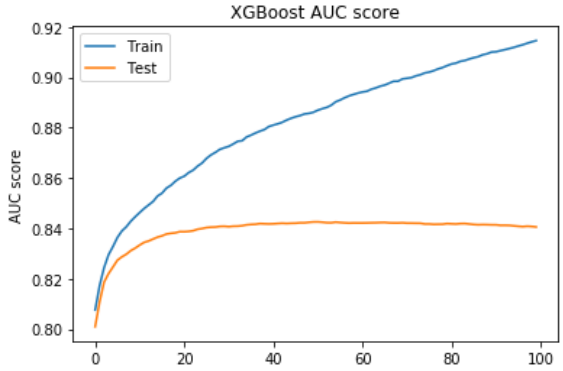

Figure 25 - Optimal numbers of trees (on sparse dataset)

6.3.3.1.2 Max depth

For the depth of each tree, the AUC-scores of five models at max depth levels ranging from 1-10 are investigated. From Figure 26, it appears that the model on average performs better at a max depth of 5 despite the fact that the best performing model had a max depth of 6 (as shown by the vertical lines). Regardless, the max depth is set at 5 as this max depth, on average, performed better.

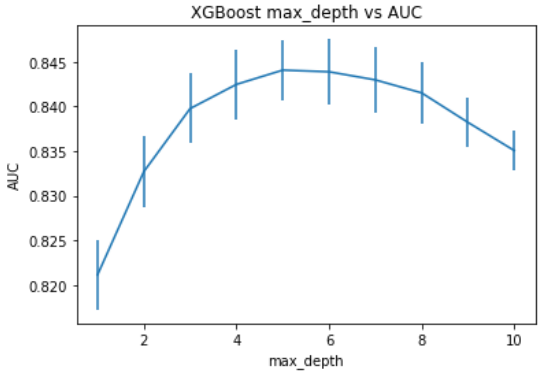

Figure 26 - Optimal tree depth (on sparse dataset)

6.3.3.1.3 RANDOM SEARCH

Following the specification of the above the hyper-parameters, i.e., *scale_pos_weight*, *n_estimators*, and *max_depth*, a random search is conducted for the four remaining hyper-parameters, i.e., *learning_rate*, *gamma*, *subsample*, and *colsample_bytree*. The random search is implemented as explained in Section 6.3.2.2, but with changes to *scale_pos_weight*, *n_estimators*, and *max_depth* as outlined above. The optimal hyper-parameters on the *sparse-GBT* models might be considerably different from the ones found for the *dense-GBT* models due to the different data structure. The overall settings for the random search on the sparse dataset are shown in Figure 27.

```
Parameters = {scale_pos_weight = 24.29,
              n_estimators = 50,
              max_depth = 5,
              learning_rate = range(0.01, 0.1),
              gamma = range(0, 5)
              subsamble = range(0.3, 0.8),
              coolsample bytree = range(0.8, 1.0)}
```

Figure 27 - Hyper-parameter settings for the random search (on the sparse dataset)

7 RESULTS

This section outlines the results of the implemented models. Specifically, the AUC-scores are presented in Table 5 and visualized in Figure 28, followed the ROC-curves of the different models. The estimated hyper-parameters can be found in Appendix 8.

As outlined below, of the five models evaluated on the dense dataset, i.e., *LDA*, *LR*, *LR-CODR*, *dense-GBT*, and *dense-GBT-CODR*, the best performing model is *dense-GBT* with an AUC-score of 0.8347, followed by *dense-GBT-CODR* with a score of 0.8332, then *LR* with 0.8191, *LR-CODR* with 0.8189, and finally *LDA* with 0.7210. For the two sparse models, the better model is the *sparse-GBT-CODR* with an AUC-score of 0.8409, followed by *sparse-GBT* with a score of 0.8236.

Model results *(AUC-scores)*				
		Altman Z-score (LDA)	Logistic Regression (LR)	Gradient Boosted Trees (GBT)
Without missing values (dense)	w/o CODR	*LDA* 0.7210	*LR* 0.8191	*Dense-GBT* 0.8347
	w/ CODR		*LR-CODR* 0.8189	*Dense-GBT-CODR* 0.8332
With missing values (sparse)	w/o CODR			*Sparse-GBT* 0.8236
	w/ CODR			*Sparse-GBT-CODR* 0.8409

Table 5 – AUC-scores of the trained model, evaluated on the test set. Best-performing models are highlighted for each dataset

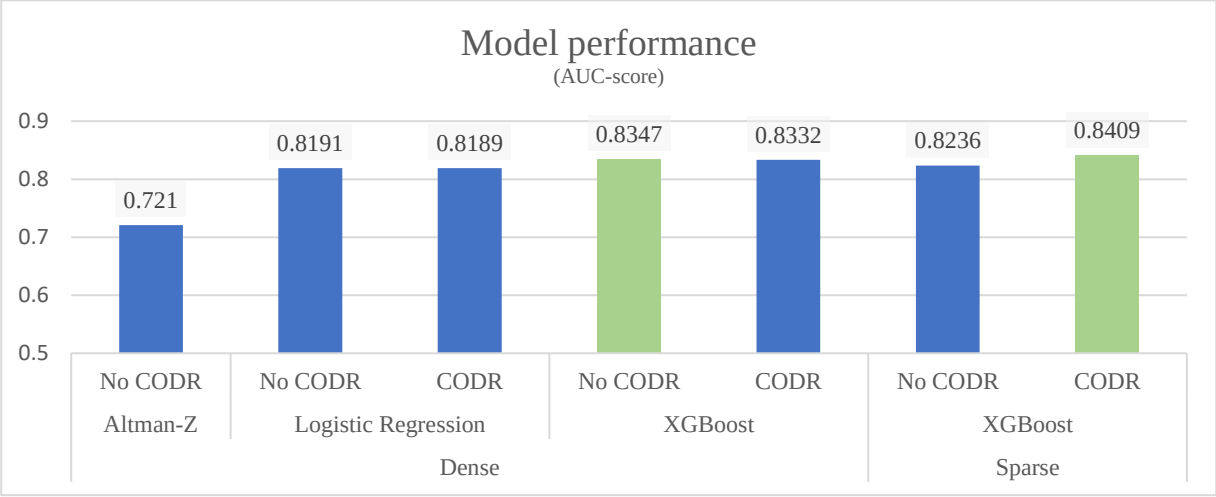

Figure 28 – AUC-scores of the trained model, evaluated on the test set. Best-performing models are highlighted for each dataset. Note that the vertical axis starts at 0.5, which indicates random guessing.

7.1 ROC-CURVES

The following section first outlines the ROC-curves underlying the AUC-scores from Table 5 and Figure 28, then the individual ROC-curves of the dataset-model categories are visualized (LR-models, both *dense-GBT* models, and both *sparse-GBT* models).

Figure 29 visualizes the best performing models in each dataset-model category. Here the *sparse-GBT-CODR* and *dense-GBT* follow a similar trajectory, though *sparse-GBT-CODR* (as evidenced by the AUC-score) is slightly more concave. Following this, *LR* is noticeably similar though with slightly lower true positive rates and higher false positive rates as the threshold increases (as the line moves from bottom-left to top-right). Lastly, the performance of *LDA* appears to be considerably different to the other models.

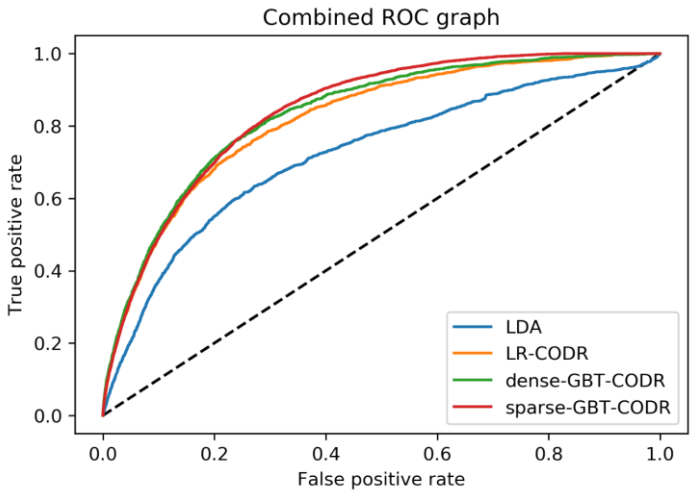

Figure 29 – ROC-curves of the best performing models in each category

For logistic regression, the *LR* and *LR-CODR* ROC-curves are visualized in Figure 30, presenting almost identical ROC-curves.

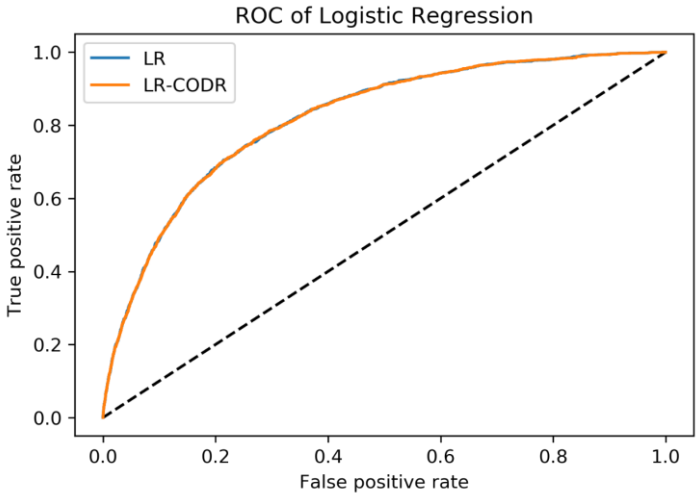

Figure 30 – ROC-curves of the logistic regression models

Similar to logistic regression, the ROC-curves for *dense-GBT* and *dense-GBT-CODR* are considerably similar, though with some more pronounced differences at the top left area of the curve, as evidenced in Figure 30.

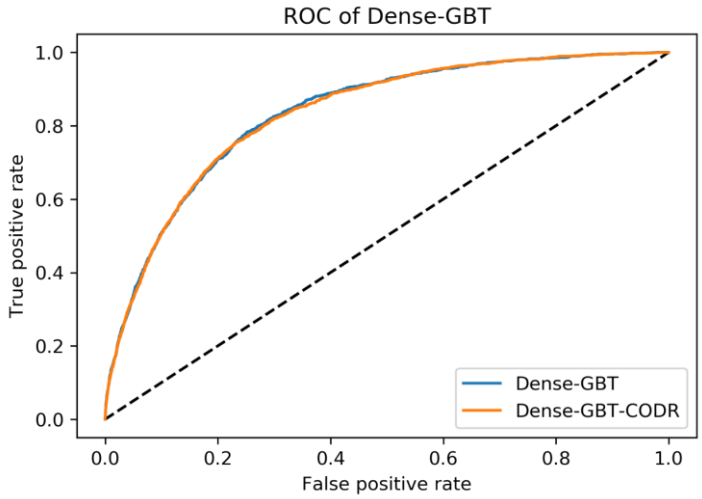

Figure 31 – ROC-curves of the dense gradient boosted trees models

Lastly, the ROC-curves for *sparse-GBT* and *sparse-GBT-CODR* are visualized in Figure 32. Here, the ROC-curves are noticeably different for different thresholds.

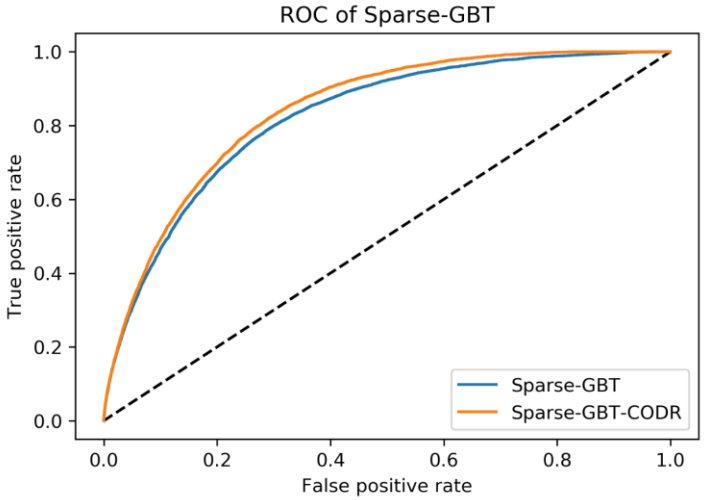

Figure 32 – ROC-curves of the sparse gradient boosted trees models

8 DISCUSSION

For the models trained on the smaller, but denser dataset, CODR does not appear to enhance the predictive power of either *logistic regression* (LR) or *gradient boosted trees* (GBT) when evaluated on the test set. However, for the large dataset with several missing values, GBT performs slightly better with the inclusion of the CODR-variable than if it had not been included. In short (using the terminology outlined Section 6.3), the results presented above indicate that there is an enhanced predictive power of the *sparse-GBT-CODR* over *sparse-GBT* while there appears to be no benefit of having a CODR-variable in the dense models.

When comparing the different models without a CODR-variable with their CODR counterparts, it is not immediately clear why only the *sparse-GBT-CODR* model performed better on the test data. The results could indicate that for companies that do not provide a "complete" financial statement, i.e., companies that provide financial statements with a large proportion of missing values, FDP-models benefit from having additional information when little is available. When comparing the feature importances of the GBT-models (which are discussed later), the fact that the CODR-variable ranks relatively low in the feature importances of the *dense-GBT-CODR* model and high in the *sparse-GBT-CODR* model, might further support this assumption. Since relatively smaller companies probably have a higher propensity to submit sparse financial statements, the sparse dataset likely also contains a larger proportion of smaller companies (or any other type of company likely to submit sparse financial statements) than the dense dataset, which has different impacts on the information contained in the dense and sparse datasets, respectively.

Considering *dense-LR, dense-LR-CODR, dense-GBT,* and *dense-GBT-CODR,* it appears that the inclusion of CODR has worsened the predictive power of the models. As discussed above, the dense dataset might encompass enough information on how to accurately predict financial distress for the given financial statements that adding CODR simply adds noise to the model. More technically, it might be that the regularizer penalizes the general model for creating complexity without any added benefit of higher prediction power, which negatively impacts the other features. Similarly, it might be that CODR simply adds noise to an already complex model, which diverts training time from important parameters to a "noisy" feature. Lastly, it might just be due to chance, indicated by the closeness of the AUC-scores.

The following section discusses some of these areas of interest. First, the data limitations and model limitations are presented for discussion. Then, the model consistency and the feasibility of cross-dataset comparison of the GBT-models are discussed. Following this, a potential approach to operationalizing the *sparse-GBT* models is presented after which the potential added predictive power of *sparse-GBT-CODR* is discussed and partially tested. Lastly, potential future work is presented.

8.1 Data Limitations

A major data challenge is the level of erroneous data, which can negatively impact the model training phase. This is exemplified by the vast number of checks implemented and the need for data modification. One of these (smaller) challenges is the presence of missing values in financial statements. This arises both from incorrect usage of the XBRL-format as evidenced in the data but is most likely also due to many strikingly specific accounting terms that only a handful of companies ever use, creating a sparse dataset, which only the GBT models can handle.

Specifically, a large data challenge is the incorrect usage or the decimal indicator in the .xml files leading to financial values that are off by three, six, and in two cases, nine orders of magnitude. Several of these errors were handled using a relatively simple approach of comparing the current year's *Assets* value to previous and subsequent year and down-scaling if considerably above (approximately 15,000 cases). This simple check does not capture all cases since newly started companies have no future or past reference points to compare with. Similarly, companies that have reported incorrect values throughout all years will not be identified either.

While the implementation of these simple checks has captured some of the errors, it might also have led to incorrect modifications in the case of growth companies. Similarly, several financial statements included negative values when they should have been positive and vice-versa, but these issues were unfortunately too inconsistent to modify in an automatic manner. There are also indications that more than 30% of the .xml files are erroneous (Mygind, 2018a).

8.2 Model Limitations

All models generally rely on the same overall data structure, which should entail that all models are impacted by the same data issues. While this is partly true for the case of erroneous data (negative reported values when they should be positive, errors of magnitude, etc.), the impact of not being able to handle missing values is much larger since only one of the implemented models, GBT, can handle missing values. Consequently, the training and test sets used for sparse and dense models differ. This might seem like a trivial challenge that simply lessens the availability of data for the models unable to handle missing values. However, important limitations and potential impacts arise from the number of missing values.

The limitation that LDA and LR requires dense data considerably limits the effective training space and causes loss of potentially important information, e.g., several financial statements might include important pieces of information that could be used for financial distress prediction, but if a number of other companies did not report these values, the features are excluded. This potentially limits the predictive power of dense models substantially and leads to an inability to predict the probability of financial distress of new data samples with sparse information. Conversely, the ability of GBT to both train on and predict using all available data despite missing values, cannot be understated. This is especially true in financial distress prediction as the simple task

of removing features that have a considerable amount of missing values (only keeping features with values in more than 60% of financial statements) results in a feature space reduction from 193 to 171 (12% reduction). Further, after removing sparse features, individual financial statements with missing values must be removed as well. In so doing, the number of financial statements is reduced from 743,607 to 154,237 – a removal of 80% of the available financial reports.

In order to minimize data shrinkage for the dense models, the threshold that specifies the minimum proportion of values needed in a feature could be changed (here 60%). This threshold has a large impact on the data reduction level as it decides on the number features to exclude, which impacts the data instance space. In this case, the choice of only keeping features that have values in more than 60% of its rows results in a 12% reduction in the feature space but a mighty 80% reduction in the data instance space. Instead, the feature threshold could be programmatically optimized so that it results in the lowest level of data shrinkage, e.g., a threshold of 91% results in a feature reduction of 16.6% and a row reduction of 14.2%. Consequentially, the 60%-threshold has potentially removed important information for the dense models that could have increased (or decreased) the prediction performance drastically.

Another limitation of this research comes from how the models have been trained. While they perform seemingly well, in fact better than what current research has achieved (Christoffersen et al., 2018; Matin et al., 2019), the models are still limited in their hyper-parameter space. Random search was applied to find the optimal combination of hyper-parameters (see Appendix 8). While this method achieves good results in general, especially when time is taken into consideration, it will not find the true optimum in the continuous hyper-parameter space. Only a close-to-optimal hyper-parameter setting will be reached. Due to computational constraints, only 20 different settings of hyper-parameters were tested per model in this study.[39] This suggests that better model performances could have been achieved, with hyper-parameters closer to the optimum.

8.3 Model Consistency and Comparisons over Datasets

Considering the four GBT models, which generally are superior to both LDA and LR models, the produced feature importances show that a similar set of features consistently show up as the ten most important features, i.e., c*ash, accounts payable, age*, *size*, *other short-term debts*, *profit*, and *retained earnings* (see Figure 33 below). Several of these are consistent with the findings of Christoffersen et al. (2018). Feature importance here is simply a score that counts the number of times each feature is split on. For the GBT-CODR models only *sparse-GBT-CODR* consider CODR an important feature, ranking sixth. In fact, in *dense-GBT-CODR* the CODR feature is not important.

[39] Because of a low computational cost, LR was trained on 50 iterations.

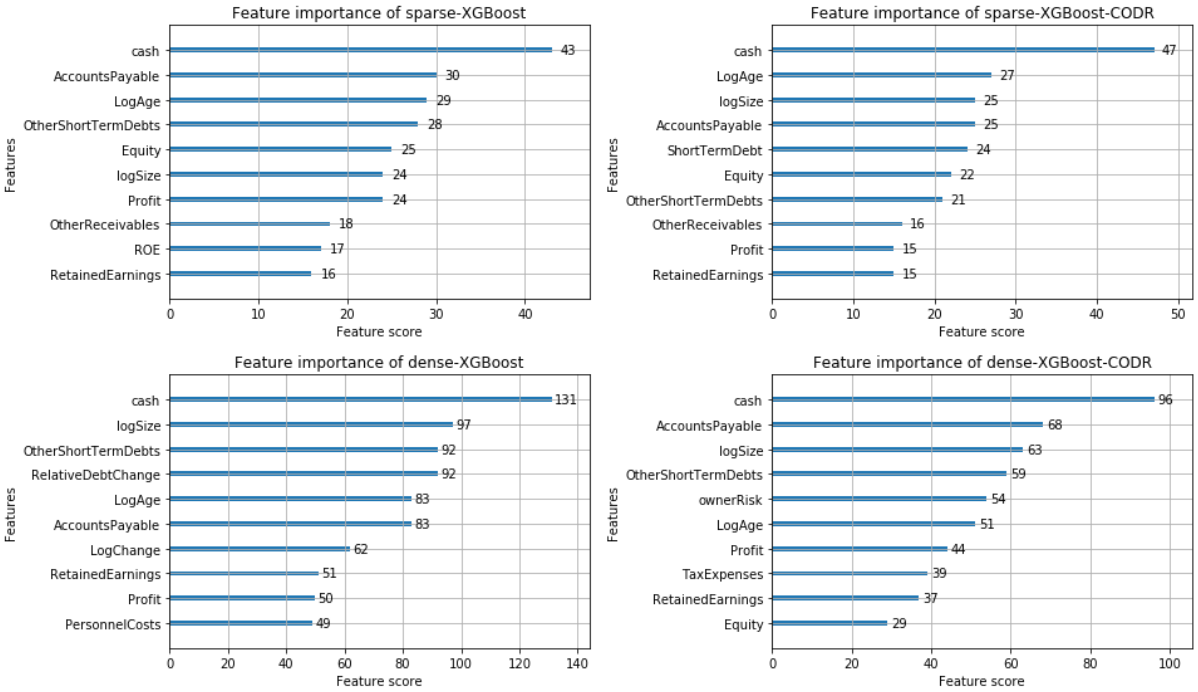

Figure 33 – Feature importances of the four GBT-models

While the type of company likely to submit sparse financial statements probably differs from the ones that submit dense statements, the AUC-scores between the sparse and dense models could also have been impacted by the split of the train and test samples as they are inherently different. This introduces a potential bias where one test set could contain instances that are relatively harder to predict compared to the other test set – meaning that *harder-to-predict* samples largely might appear in the training set for one of the datasets and in the test set for the other.

At first sight, this split issue between datasets could be solved by first performing the train and test split on the sparse dataset that contains all financial statements and features, and then removing the sparse features and sparse financial statements for the models requiring a dense dataset (compared to the procedure employed in this thesis of first removing missing values, and then splitting). This ensures that all training samples used in the dense models are used as training samples in the sparse models as well (and that test samples for dense models are used as test samples for sparse models). However, it does not work in the opposite direction, i.e., that all training samples used in the sparse dataset are used in the dense training set (similarly for the test sets). Since this approach still does not equate the two datasets as the sparse financial statements remain for one of the models and not for the other, it could still lead to *harder-to-learn/predict* samples in the sparse dataset.

Furthermore, removing data instances *after* a split could skew the class balance of the training and test sets. Highly imbalanced datasets are especially sensitive to such post-split processing since small changes in the minority class can lead to large proportional fluctuations. Similarly, this could also change the prior class

probabilities in models that rely on such weights (e.g., logistic regression), which could lead to usage of incorrect probabilities from a training set that does not accurately reflect the reality.

However, while there could be impacts stemming from the above-outlined approaches, it does not seem particularly plausible that two distinct random splits would make such an impact primarily due to the sheer number of financial statements that occur in both datasets. Since the original dataset is of a decent size, the impact of the split is lowered. Thus, a random split on these large datasets probably does not carry a noticeable impact on the AUC-scores.

From the results table it is clearly indicated that the models, in order from best to worst, are *sparse-GBT-CODR*, followed by the two dense-GBT models, then *sparse-GBT*, then the LR-models, and lastly *LDA*. However, model performance cannot (perhaps, should not) be compared across different datasets. The primary issue is the non-identical test samples, which restricts the ability to make fair comparisons. Some of the deliberations outlined above, e.g., on samples that are *harder-to-predict*, potential sample bias, etc. might give an indication as to why this might be the case. Instead, the better approach is to compare models that are tested on the same test set, and for the case of financial distress prediction here, to compare non-CODR models with their CODR counterpart – both using the same sparse or dense dataset.

When comparing models on the same dense dataset, this thesis has produced models that perform better than industry standard models (LR) and historical models (LDA). However, since all these models require a dense data structure to make predictions, it is severely limited and there are clear benefits of utilizing financial distress prediction models that do not rely on strict data input rules, especially since most of the available financial statements from the Danish Business Authority are sparse as a rule rather than as an exception. Consequently, the practical application opportunities of dense models are considerably more limited than that of sparse models. Consider providing a dense and a sparse model the same list of companies to risk-assess, respectively. Even if the scores (AUC, precision, recall, misclassification rate, etc.) were similar (or even if a dense model outperformed a sparse model) – a sparse model would still be preferable for the simple reason that it is more flexible and contains the predictive ability for (almost) any level of sparsity. The following discussion will only discuss the two sparse models: *sparse-GBT* and *sparse-GBT-CODR* because of this crucial inherent superiority of sparse models and further due to the indication that the CODR feature enhances predictive power, which the following seeks to investigate.

8.4 Operationalization of Sparse Models

Throughout this thesis, AUC has been used as an objective scoring metric that compares classifier performance and which is widely used for imbalanced dataset (Hand, 2009). Thus, AUC has been used as an objective comparison scoring method over all possible thresholds. However, to operationalize the FDP-models, the threshold that defines whether a company is classified as *financially distressed* or not must be specified.

Specifying a threshold is largely dependent on the use case of the financial distress prediction and requires some of the following (financial or otherwise relevant) *cost* deliberations. Regardless of the use case of an FDP-model, the cost of misclassifying companies (false positive and false negative) and the benefit of correctly classifying companies (true positive and true negative) should be quantified. Quantifying or otherwise specifying these costs allow for an optimization of the threshold such that the costs of using the model are minimized.

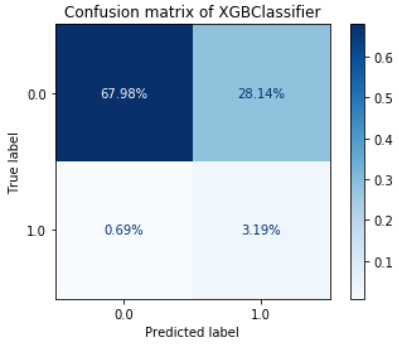

Figure 34 – Confusion matrix of sparse-GBT-CODR on test set using a 0.5 threshold

Consider the confusion matrix in Figure 34 above, which shows the number of correctly classified and incorrectly classified instances with a (standard) threshold of 0.5. Considering only the misclassifications (false positive and false negative), there is a much higher number of false positives than false negatives. If the costs of false positives and false negatives are equal for a certain use case, then the cost impact of using a threshold of 0.5 is suboptimal. In other words, the threshold can be changed such that the total misclassification instances shrink, and the number of false positives and false negatives are more balanced. More formally, the optimal threshold is the one that maximizes the following.

$$\text{maximize } \left[\left(\gamma_{fp} * FP\right) + \left(\gamma_{fn} * FN\right)\right] \tag{17}$$

Where FP and FN are *false positives* and *false negatives*, respectively, and γ_{fp} and γ_{fn} are the individual impacts of these, respectively. Note that if γ is negative, it constitutes a cost – if positive, a benefit. The equation can then be further expanded to include the benefits (or costs) of correct classification (TP and TN).

$$\text{maximize } \left[\left(\gamma_{fp} * FP\right) + \left(\gamma_{fn} * FN\right) + \left(\gamma_{tp} * TP\right) + \left(\gamma_{tn} * TN\right)\right] \tag{18}$$

As previously noted, the cost of misclassification differs between use cases. However, Altman et al. (1977) find that the cost of misclassifying a firm that is financially distressed as not financially distressed (false negative) is 35 times higher than the cost of a false positive in loan-giving situations. They argue that this cost distribution primarily stems from auditor costs, damages to the public brand, legal costs, etc. Although the costs could be a 35-to-1 relationship, the following will assume equal costs as the authors are of the opinion that the cost-distribution is highly situational, following Balcaen & Ooghe (2006) who also find that most

practitioners and academics assume equal costs of false positives and negatives. As such, the optimized threshold in the following should not constitute a truth, rather the following should depict the benefit of finding such a threshold in terms of cost.

First, the trained model, *sparse-GBT-CODR*, predicts the probabilities of financial distress on a training set. Then, using these probabilities and the actual values, the threshold is changed from 0 to 1 using small steps – and at each point, the proportion of false positives and false negatives is calculated. Once the proportion of false positives and false negatives multiplied by their respective costs reaches a balance, an optimal cost threshold has been found. Here, the calculated threshold is $0.8054 \approx 0.81$. Then, the classification performance using optimized threshold, 0.81, is tested on the test set. The results are outlined in the confusion matrix in Figure 35 below, indicating that the optimized threshold for the *sparse-GBT-CODR* model generalize well since the resultant balance between the false positives (2.97%) and the false negatives (3.11%) is approximately equal.

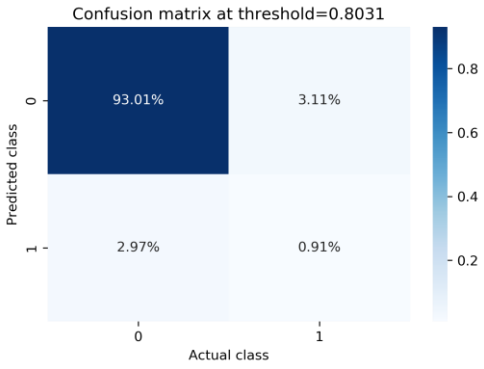

Figure 35 – Confusion matrix of *sparse-GBT-CODR* on test set using a 0.8054 threshold

Comparing the 0.81 threshold to the original 0.50, the misclassification rate on the test set has decreased from $28\% + 0.7\% = 28.7\%$ to $2.97\% + 3.11\% = 6.08\%$. Using a misclassification cost of 1 and a correct classification cost (or benefit) of 0, this provides the following impacts (costs) of using thresholds 0.50 and 0.81, respectively.

$$impact_{0.50} = [(-1 * 0.7\%) + (-1 * 28\%)] = -28.7$$

$$impact_{0.81} = [(-1 * 2.97\%) + (-1 * 3.11\%)] = -6.08$$

Which gives the following improvement of using the 0.8 over the 0.5 threshold.

$$improvement_{0.81-0.50} = -6.08 - (-28.7) = 22.62$$

Thus, if the misclassification cost of 1 represents DKK 1, each usage of the model using 0.5 and 0.81 as thresholds would result in financial costs of -28.7 and -6.08 on average, respectively.

The above operates with a dichotomous approach to classification. Instead, the predicted probabilities can be used to create multiple bins indicating the likelihood that a company is financially distressed, see Figure 36 below for the probability density plots of the two classes (note the distressed curve has been upscaled for visualization purposes). The approach of binning continuous values is presumably an approach utilized by Nordea that credit scores companies on an integer scale from 0 to 7 (see Appendix 1). Similar to Nordea, using the knowledge of the data distribution in Figure 36, bins can be created in a similar manner to the cost-based calculations above or in a more qualitative manner, e.g., a probability over 90% indicating financial distress *highly likely*, over 70% as *likely*, over 50% as *possible*, etc. depending on the use case and the cost of not identifying a financially distressed company and the cost of wrongly identifying a company as financially distressed. Furthermore, the predicted probabilities and the knowledge of the distribution as outlined in Figure 36 can be used to create a *zone of ignorance* similar to Altman (1968) where probabilities in a certain range are classified as *unknowns* or other categories.

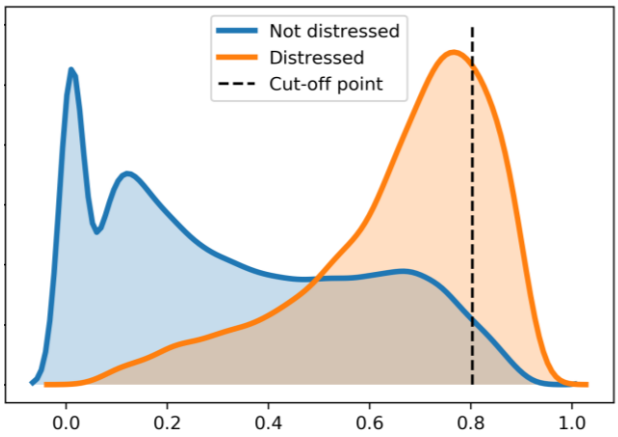

Figure 36 – Probability density plot

Note: The *distressed* density plot has been upscaled considerably.

The *cut-off point* denotes the cost-optimal point ($\approx \mathbf{0.81}$)

8.5 Inclusion of Non-Financial Ownership Information

Several models have been tested in this thesis. However, as outlined in the sections above, the sparse models are generally superior to the dense models with better practical opportunities stemming from their ability to operate on limited data information. As such, the latter parts of the discussion section largely revolved around the two sparse models, i.e., *sparse-GBT* and *sparse-GBT-CODR*. Considering these two models, the inclusion of non-financial ownership information – of which the *company ownership default risk* (CODR) feature is a proxy – appears to improve the power of financial distress prediction with AUC-scores of 0.8236 and 0.8409 for *sparse-GBT* and *sparse-GBT-CODR*, respectively. Despite the seemingly relatively modest improvement, the enhanced predictive power is noticeable on the ROC-curves as illustrated in Figure 37 below.

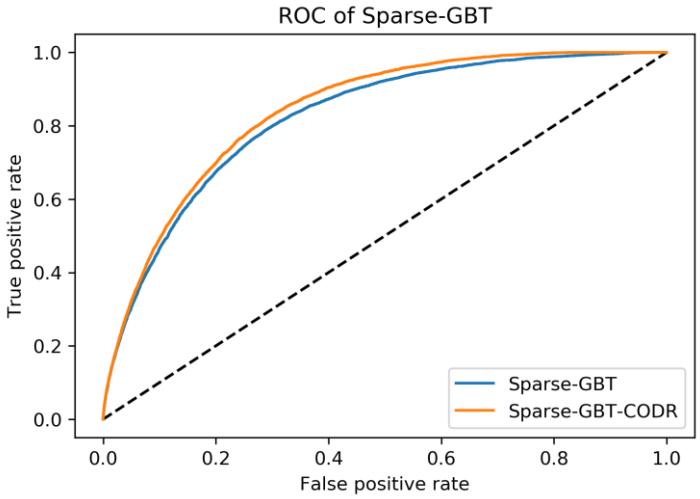

Figure 37 – ROC curves of sparse-GBT and sparse-GBT-CODR

The cross-validation results in Figure 38 below further support the possibility that the two *sparse-GBT* models perform differently and the hypothesis that non-financial ownership information might increase predictive power. Here, the AUC-scores from the 5-fold cross-validation of the fitted hyper-parameters is visualized, which shows no overlap between the five folds between the models. However, it is important to note that the cross-validation for the two models was performed on two different random samples, which entails that the individual folds are not directly comparable to one another.

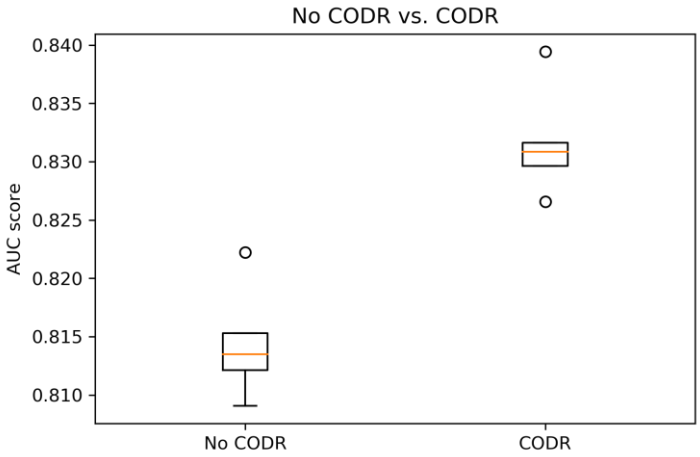

Figure 38 – Comparison of AUC-scores of *sparse-GBT* and *sparse-GBT-CODR* in during 5-fold cross-validation.
Note that the folds are random, meaning they are not directly comparable between the models.

As outlined in the previous sections, the feature importance rank of *sparse-GBT-CODR* similarly indicates that the inclusion of the CODR feature is an important part of the model. However, to more robustly test whether the inclusion of a CODR variable positively enhances predictive power or not, certain tests can be performed to test statistically significant differences between machine learning models. There are a variety of test for such tests. Dietterich (Dietterich, 1998) discusses the implications of using five different statistical tests for different

purposes, depending on how *expensive* the training phase of models is, i.e., the time and computation power needed to train models. He suggests performing *5×2 cv* (five repetitions of a 2-fold cross-validation), which could be employed in the thesis. However, due to the computational power and time needed to perform such tests on top of the already-trained models, this is not be feasible. For non-expensive statistical tests, Dietterich (Dietterich, 1998) proposes performing the McNemar test instead. The McNemar test, tests the null hypothesis that two algorithms have the same error rate, e.g., that the two classifiers disagree the same amount. Consequently, rejecting the null hypothesis suggests that there is evidence that the two classifiers disagree in different ways.

The following will outline the McNemar's test on the *sparse-GBT* and *sparse-GBT-CODR* models with the null hypothesis that the classifiers disagree the same amount. To perform the test on two classifiers, the thresholds for each model must be specified first since the McNemar's requires a 2×2 contingency matrix with dichotomous classification, as outlined in Table 6 below.

	sparse-GBT correct	***sparse-GBT*** incorrect
sparse-GBT-CODR correct	**(a)** No. of times both models classify **correctly**	**(b)** No. of times *sparse-GBT-CODR* is **correct** when *sparse-GBT* is **incorrect**
sparse-GBT-CODR incorrect	**(c)** No. of times *sparse-GBT-CODR* is **incorrect** when *sparse-GBT* is **correct**	**(d)** No. of times both models classify **incorrectly**

Table 6 – 2x2 contingency table for the two sparse-GBT classifiers

Once the contingency table has been filled out, the test statistic can be calculated using cells *b* and *c* above. Since the McNemar's test tests the null hypothesis that the two classifiers disagree the same amount, only the counts of disagreement are included (cells bottom-left and top-right in Table 6). Formally, the statistic is

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{19}$$

Where *b* and *c* denote the counts of disagreements between the models as in Table 6. To perform the actual test, the previously estimated optimal threshold of $0.8054 \approx 0.81$ is used for the *sparse-GBT-CODR* model with the assumption of equal misclassification costs, and when estimating the threshold for the *sparse-GBT* model with the same assumptions, the estimated threshold is $0.790103 \approx 0.79$. These thresholds are then used to classify all instances in the test set, filling the contingency table (see Table 7 below).

	sparse-GBT correct	***sparse-GBT*** incorrect
sparse-GBT-CODR correct	173,030	1,639
sparse-GBT-CODR incorrect	1,599	9,634

Table 7 – 2x2 contingency table (sparse-GBT-CODR threshold ≈ 0.81, sparse-GBT threshold ≈ 0.79)

Leading to the following test statistic

$$\chi^2 = \frac{(b-c)^2}{b+c} \approx 0.49 \tag{20}$$

Which leads to a *p-value* of ≈ 0.482. Consequently, the null hypothesis that the two models are significantly different cannot be rejected on a significance level $\alpha = 5\%$, which suggests that for the specified thresholds, the models appear similar. However, an important limitation to the McNemar test for classification is that it relies on clearly defines thresholds and dichotomous classification. Consequently, the specification of other thresholds could result in different conclusions on the difference between the performance of *sparse-GBT* and *sparse-GBT-CODR* models.

If the above test is repeated using the standard threshold of 0.5 for both models (or assuming a different cost structure), the contingency is considerably different (see Table 8 below).

	sparse-GBT correct	***sparse-GBT*** incorrect
sparse-GBT-CODR correct	125,985	7,399
sparse-GBT-CODR incorrect	6,322	46,196

Table 8 – 2x2 contingency table (threshold = 0.5)

Performing the McNemar's test on these classification outcomes result in $\chi^2 \approx 84.53$ and a *p-value* of ≈ 0.00. Here, the null hypothesis is rejected on the previously set significance level ($\alpha = 5\%$). Thus, the models appear to be significantly different for some thresholds, but not for others, suggesting that the CODR feature, and non-financial ownership information generally, can be included in FDP-models with increased predictive performance over models that do not include non-financial ownership information for certain use cases.

Despite significant differences between the models for some thresholds, the difference in performance is not clear-cut since the performance largely seems to depend on the specification of the threshold, which predominantly is a practical decision that depends on the use case. Consequently, the McNemar test on the difference in performance of two classifiers might not be a suitable test to perform to estimate the general difference in performance between the models. Instead, a *5×2 cv* test proposed by Dietterich (1998), although computationally *expensive*, might be a better approach as it "assesses the effect of both the choice of training set (by running the learning algorithms on several different training sets) and the choice of test set (by measuring the performance on several test sets)" (p. 1919), which allows for more robust comparisons between the models. The indications of differences in model performance with the inclusion of non-financial ownership information also suggest that further research is warranted into the topic with potential benefits for academics in the field of FDP and practitioners alike.

Lastly, the inclusion of non-financial ownership information (or other types of information external to financial statements) open up for the possibility for financial distress predictions to become continuous and less reliant on the publication time of financial statements, which often are published with one year's interval. Consequently, by the time the financial statements are published, the numbers might not accurately reflect the current reality anymore. Specifically, financial distress predictions can now be made whenever information external to the financial statements is updated. For the case of CODR, model predictions can be adjusted instantaneously when a change in the CODR feature occurs rather than having to wait a year for a new financial statement to be released. Including other important features would further enable more dynamic FDP-models for the benefit of stakeholders relying on accurate and timely predictions.

8.6 FUTURE WORK

Doing the ideation, development, de-bugging, and writing stages of this thesis, several discussions arose on other avenues within the area of financial distress prediction. Some of these where implemented and some are introduced as topics for future work.

For this thesis, potential changes include incorporating the number of employees as both a feature, but also as a measure for outlier detection where companies above a certain threshold, e.g., $\frac{Profit}{\#\ of\ employees}$, would be marked as potential outliers. Similarly, since many erroneous financial statements are identified and consequently adjusted, an *erroneous statement* flag could be added as a feature indicating whether the financial statement is erroneous. Likewise, the textual information included in the financial statements could be further included in the models similar to the approach of Matin et al. (2019) that found an improvement of including auditor reports – however, this is largely out of scope for this thesis due to its complexity. Further, a macro-economic feature that captures relevant financial information external to the company could also be utilized. The data scope could also be broadened to include older financial statements by developing a robust OCR-scanner. Assuming a good OCR-model, it can then further be used to classify digitally erroneous financial statements, which heightens data quality.

The CODR feature could also be expanded to include not only the *immediate* owner, but the *ultimate* owner, which presumably would lead to a score that is more tied to the *root-cause*, i.e., the person and not the holding company. Other proxies than CODR for non-financial ownership information could also be integrated, including the (members of the) *board ownership default rate* or inclusion of the *ownership concentration* heavily discussed in the literature (Daily & Dalton, 1994a, 1994b; Deng & Wang, 2006; Donker et al., 2009; Lajili & Zéghal, 2010; Mangena & Chamisa, 2008; Manzaneque et al., 2016). Additionally, rather than having a score like CODR that is calculated based on the proportion of failures, a more "positive" owner score could be developed, including previous ownership performance such as individual growth numbers, the owners'

previous experience (e.g., quantification of the number of years in healthy companies), etc. If a proper data pipeline is setup, social media information could also be integrated, though this likely is resourceful.

The FDP-models developed and described in this thesis heavily rely on financial statements, which excludes many start-ups that presumably are more prone to financial distress. Consequently, it could be beneficial to develop FDP-models that are able to take these considerations into account, e.g., through a focus on non-financial features that are available prior to the publication of the first financial statement, e.g., CODR or other ownership variables outlined above such as the ability of the owner to drive company growth or the ability to maintain the health of these.

9 CONCLUSION

Motivated by the potentially underdeveloped aspect of including non-financial ownership information as a predictor in *financial distress prediction*, the development of high-performance models using *machine learning*, and the considerable amount of data on limited Danish companies from the Danish Business Authority, this thesis set out to investigate the following research question: "How does the inclusion of non-financial ownership information affect the performance of financial distress prediction models on Danish companies?"

To answer the research question above, three types of models were trained and evaluated on *dense* data, i.e., *linear discriminant analysis* (LDA), *logistic regression* (LR), and *gradient boosted trees* (dense-GBT). First, all models were trained on financial ratios without a proxy for *non-financial ownership information*, i.e., *company ownership default risk* (CODR). Then, *LR* and *dense-GBT* were trained using CODR to compare predictive power. Realizing the extent to which the dense dataset limits the predictive ability of models stemming from model needs of removing missing data, GBT was additionally trained on the full *sparse* dataset both with and without CODR, i.e., producing the *sparse-GBT* and *sparse-GBT-CODR* models, creating a total of seven models.

The results from the financial distress predictions of Danish limited companies show an enhanced predictive power of *sparse-GBT-CODR* ($AUC = 0.8409$) over *sparse-GBT* ($AUC = 0.8236$). For dense models, however, the results do not suggest a benefit of including CODR. The McNemar test was then performed on the two sparse models, *sparse-GBT-CODR* and *sparse-GBT*, and found that model performance is significantly different for certain thresholds, though not for others. These findings suggest that the inclusion of CODR – and possibly non-financial ownership information generally – for increased predictive performance is largely business-dependent. In certain contexts, the inclusion of non-financial ownership information might increase the predictive performance over models that do not include non-financial ownership information. In so doing, existing FDP-models could be enhanced, enabling better and more fair credit scoring, more accurate pricing of credit risk, and enable the creation of more dynamic and up-to-date predictions by avoiding the reliance on financial statements published on an annual basis and instead adjust predictions as soon as the non-financial predictors change.

Despite the promising indications, it cannot be said for certain that the inclusion of non-financial ownership information positively affects the performance of financial distress prediction models on Danish companies. However, the authors believe that the above findings clearly suggest that further research into the inclusion of non-financial ownership information is warranted.

10 BIBLIOGRAPHY

Alexandropoulos, S. A. N., Aridas, C. K., Kotsiantis, S. B., & Vrahatis, M. N. (2019). A deep dense neural network for bankruptcy prediction. *Communications in Computer and Information Science*. https://doi.org/10.1007/978-3-030-20257-6_37

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance*. https://doi.org/10.2307/2978933

Altman, E. I., Haldeman, R. G., & Narayanan, P. (1977). ZETATM analysis A new model to identify bankruptcy risk of corporations. *Journal of Banking and Finance*, *1*(1), 29–54. https://doi.org/10.1016/0378-4266(77)90017-6

Altman, E. I., Iwanicz-Drozdowska, M., Laitinen, E. K., & Suvas, A. (2017). Financial Distress Prediction in an International Context: A Review and Empirical Analysis of Altman's Z-Score Model. *Journal of International Financial Management and Accounting*, *28*(2), 131–171. https://doi.org/10.1111/jifm.12053

Altman, E. I., & Narayanan, P. (1997). An International Survey of Business Failure Classification Models. *Financial Markets, Institutions & Instruments*, *6*(2), 1–57. https://doi.org/10.1111/1468-0416.00010

Aziz, M. A., & Dar, H. A. (2006). Predicting corporate bankruptcy: Where we stand? *Corporate Governance*, *6*(1), 18–33. https://doi.org/10.1108/14720700610649436

Balcaen, S., & Ooghe, H. (2006). 35 years of studies on business failure: an overview of the classic statistical methodologies and their related problems. *The British Accounting Review*, *38*(1), 63–93. https://doi.org/10.1016/j.bar.2005.09.001

Beaver, W. H. (1966). Financial Ratios As Predictors of Failure. *Journal of Accounting Research*, *4*, 71. https://doi.org/10.2307/2490171

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*.

Bhimani, A., Gulamhussen, M. A., & da Rocha Lopes, S. (2014). Owner liability and financial reporting information as predictors of firm default in bank loans. *Review of Accounting Studies*, *19*(2), 769–804. https://doi.org/10.1007/s11142-013-9269-0

Bisnode. (2017). *Gratis data kan blive en dyr fornøjelse - Bisnode Danmark*. https://www.bisnode.dk/bliv-klog-paa-data/nyheder/gratis-data-kan-blive-dyrt/

Borovcnik, M., Bentz, H.-J., & Kapadia, R. (2012). A Probabilistic Perspective. In *Chance Encounters: Probability in Education*. The MIT Press. https://doi.org/10.1007/978-94-011-3532-0_2

Brownlee, J. (2018). *XGBoost With Python*.

Brownlee, J. (2019, January). *What is the Difference Between a Parameter and a Hyperparameter?* https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/

Büyüköztürk, Ş., & Çokluk-Bökeoğlu, Ö. (2008). Discriminant function analysis: Concept and application. *Egitim Arastirmalari - Eurasian Journal of Educational Research*.

Câmara, A., Popova, I., & Simkins, B. (2012). A comparative study of the probability of default for global financial firms. *Journal of Banking and Finance*, *36*(3), 717–732. https://doi.org/10.1016/j.jbankfin.2011.02.019

Charitou, A., Lambertides, N., & Trigeorgis, L. (2008). Bankruptcy prediction and structural credit risk models. In S. Jones & D. A. Hensher (Eds.), *Advances in Credit Risk Modelling and Corporate Bankruptcy Prediction* (pp. 154–174). Cambridge University Press. https://doi.org/10.1017/CBO9780511754197.007

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Christoffersen, B., Matin, R., & Mølgaard, P. (2018). Can Machine Learning Models Capture Correlations in Corporate Distresses? In *Danmarks Nationalbank. Working Papers* (No. 128; Danmarks Nationalbank. Working Papers). http://www.nationalbanken.dk/da/publikationer/Sider/2018/10/Working-Paper-No-128.aspx

Crouhy, M., Galai, D., & Mark, R. (2000). A comparative analysis of current credit risk models. *Journal of Banking and Finance*, *24*(1), 59–117. https://doi.org/10.1016/S0378-4266(99)00053-9

Daily, C. M., & Dalton, D. R. (1994a). Bankruptcy and Corporate Governance: The Impact of Board Composition and Structure. *Academy of Management Journal*. https://doi.org/10.5465/256801

Daily, C. M., & Dalton, D. R. (1994b). Corporate governance and the bankrupt firm: An empirical assessment. *Strategic Management Journal*. https://doi.org/10.1002/smj.4250150806

Daume, H. (2017). *A course in machine learning*. http://ciml.info/

David, Z. (2019). Information leakage in financial machine learning research. *Algorithmic Finance*, *8*(1–2),

1–4. https://doi.org/10.3233/AF-190900

Deng, X., & Wang, Z. (2006). Ownership Structure and Financial Distress: Evidence from Public-Listed Companies in China. *International Journal of Management*.

Denning, K. C., Perris, S. P., & Lawless, R. M. (2001). Serial bankruptcy: Plan infeasibility or just bad luck? *Applied Economics Letters*. https://doi.org/10.1080/13504850150204156

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*. https://doi.org/10.1162/089976698300017197

Dimitras, A., Zanakis, A., & Zopoudinis, C. (1996). A survey of business failures with an emphasis on failure prediction methods and industrial applications. *European Journal of Operational Research*.

Donker, H., Santen, B., & Zahir, S. (2009). Ownership structure and the likelihood of financial distress in the Netherlands. *Applied Financial Economics*. https://doi.org/10.1080/09603100802599647

Egholm, L. (2014). *Videnskabsteori : perspektiver på organisationer og samfund*. Hans Reitzel.

Erhvervsstyrelsen. (2015). *Indsendelsesbekendtgørelsen*. 1–51. http://filer.erhvervsstyrelsen.dk/file/268499/vejledning_indsendelsesbekendtgoerelsen.pdf

Fawcett, F., & Provost, T. (2013). Data Science for Business. In *O'Reilly*.

Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, *7*(2), 179–188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

FitzPatrick, P. J. (1932). A Comparison of the Ratios of Successful Industrial Enterprises With Those of Failed Companies. *The Certified Public Accountant*, 727–731.

Friedman, J. (1977). A Recursive Partitioning Decision Rule for Nonparametric Classification. *IEEE Trans. Computers*, *26*, 404–408. https://doi.org/10.1109/TC.1977.1674849

Geron, A. (2017). *Hands–On Machine Learning with Scikit–Learn and TensorFlow 2nd edition* (1st ed.). O'Reilly Media.

Gottardo, P., & Moisello, A. M. (2019). *Capital Structure, Earnings Management, A Comparative Analysis Distress and Risk of Financial of Family and Non-family Firms*. Springer.

Guba, E. (1991). The Paradigm Dialog. *Canadian Journal of Sociology / Cahiers Canadiens de Sociologie*, *16*(4), 19–27. https://doi.org/10.2307/3340973

Hall, P., & Gill, N. (2019). An Introduction to Machine Learning Interpretability. In *Nature Machine Intelligence*. https://doi.org/10.1038/s42256-019-0048-x

Hamer, M. M. (1983). Failure prediction: Sensitivity of classification accuracy to alternative statistical methods and variable sets. *Journal of Accounting and Public Policy*, *2*(4), 289–307. https://doi.org/10.1016/0278-4254(83)90032-7

Han, J., Kamber, M., & Kaufmann, M. (2012). *Data Mining: Concepts and Techniques* (Third).

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, *77*(1), 103–123. https://doi.org/10.1007/s10994-009-5119-5

Herlau, T., Schmidt, M. N., & Mørup, M. (2018). *Introduction to Machine Learning and Data Mining* (1st ed.). Technical University of Denmark.

Hotchkiss, E. S. (1995). Postbankruptcy Performance and Management Turnover. *The Journal of Finance*. https://doi.org/10.2307/2329237

Huang, H. T., & Tserng, H. P. (2018). A Study of Integrating Support-Vector-Machine (SVM) Model and Market-based Model in Predicting Taiwan Construction Contractor Default. *KSCE Journal of Civil Engineering*. https://doi.org/10.1007/s12205-017-2129-x

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting : Principles and Practice* (2nd ed.). Otexts.

International Accounting Standards Board. (2020). *IFRS - IFRS Taxonomy 2020*. https://www.ifrs.org/issued-standards/ifrs-taxonomy/ifrs-taxonomy-2020/

Jabeur, S. Ben, & Fahmi, Y. (2018). Forecasting financial distress for French firms: a comparative study. *Empirical Economics*, *54*(3), 1173–1186. https://doi.org/10.1007/s00181-017-1246-1

Jackson, R. H. G., & Wood, A. (2013). The performance of insolvency prediction and credit risk models in the UK: A comparative study. *British Accounting Review*, *45*(3), 183–202. https://doi.org/10.1016/j.bar.2013.06.009

Johnson, C. G. (1970). Ratio Analysis and the Prediction of Firm Failure. *The Journal of Finance*, *25*(5), 1166–1168. https://doi.org/10.2307/2325590

Jones, S., Johnstone, D., & Wilson, R. (2015). An empirical evaluation of the performance of binary classifiers in the prediction of credit ratings changes. *Journal of Banking and Finance*, *56*(C), 72–85. https://doi.org/10.1016/j.jbankfin.2015.02.006

Jones, S., Johnstone, D., & Wilson, R. (2017). Predicting Corporate Bankruptcy: An Evaluation of Alternative Statistical Frameworks. *Journal of Business Finance and Accounting*, *44*(1–2), 3–34. https://doi.org/10.1111/jbfa.12218

Joy, O. M., & Tollefson, J. O. (1975). On the Financial Applications of Discriminant Analysis. *Journal of Financial and Quantitative Analysis*, *10*(5), 723–739. https://doi.org/10.2307/2330267

Kuldeep, K., & Sukanto, B. (2006). Artificial neural network vs linear discriminant analysis in credit ratings forecast: A comparative study of prediction performances. *Review of Accounting and Finance*, *5*(3), 216–227. https://doi.org/10.1108/14757700610686426

Lajili, K., & Zéghal, D. (2010). Corporate governance and bankruptcy filing decisions. *Journal of General Management*. https://doi.org/10.1177/030630701003500401

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, *16*(3).

Mai, F., Tian, S., Lee, C., & Ma, L. (2019). Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, *274*(2), 743–758. https://doi.org/10.1016/j.ejor.2018.10.024

Mangena, M., & Chamisa, E. (2008). Corporate governance and incidences of listing suspension by the JSE Securities Exchange of South Africa: An empirical analysis. *International Journal of Accounting*. https://doi.org/10.1016/j.intacc.2008.01.002

Manzaneque, M., Merino, E., & Priego, A. M. (2016). The role of institutional shareholders as owners and directors and the financial distress likelihood. Evidence from a concentrated ownership context. *European Management Journal*. https://doi.org/10.1016/j.emj.2016.01.007

Matin, R., Hansen, C., Hansen, C., & Mølgaard, P. (2019). Predicting distresses using deep learning of text segments in annual reports. *Expert Systems with Applications*, *132*, 199–208. https://doi.org/10.1016/j.eswa.2019.04.071

Moyer, R. C. (1977). Forecasting Financial Failure: A Re-Examination. *Financial Management*, *6*(1), 11–17. https://doi.org/10.2307/3665489

Mygind, D. (2018a). *Fejl i regnskabsdata tvinger virksomheder og myndigheder til manuel kontrol*.

Mygind, D. (2018b). *Ringe datavalidering årsag til fejlbehæftede digitale årsregnskaber | Version2*. https://www.version2.dk/artikel/ringe-datavalidering-aarsag-fejlbehaeftede-digitale-aarsregnskaber-

1086915

N., E. Popper, K. (1935). Logik der Forschung. *The Journal of Philosophy*, *32*(4), 107. https://doi.org/10.2307/2016612

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*. https://doi.org/10.2307/2490395

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://scikit-learn.org/

Sabela, S. W., Brummer, L. M., Hall, J. H., & Wolmarans, H. P. (2018). Using fundamental, market and macroeconomic variables to predict financial distress: A study of companies listed on the Johannesburg Stock Exchange. *Journal of Economic and Financial Sciences*, *11*(1), e1–e11. https://doi.org/10.4102/jef.v11i1.168

Schuermann, T. (2005). A review of recent books on credit risk. In *Journal of Applied Econometrics* (Vol. 20, Issue 1, pp. 123–130). https://doi.org/10.1002/jae.827

Smith, R. F., & Winakor, A. H. (1935). Changes in the financial structure of unsuccessful industrial corporations. *University of Illinois. Bureau of Business Research. Bulletin*, *51*.

Sun, J., Fujita, H., Chen, P., & Li, H. (2017). Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2016.12.019

Sun, J., Shang, Z., & Li, H. (2014). Imbalance-oriented SVM methods for financial distress prediction: a comparative study among the new SB-SVM-ensemble method and traditional methods. *The Journal of the Operational Research Society*, *65*(12), 1905–1919.

Suntraruk, P. (2010). A Review of Statistical Methods in the Financial Distress Literature. *AU Journal of Management*, *8*(2), 31–41.

Tabachnick, B. G., & Fidell, L. S. (2000). *Using multivariate statistics* (4. ed.).

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining* (1st ed.). Pearson.

Tang, X., Li, S., Tan, M., & Shi, W. (2020). Incorporating textual and management factors into financial distress prediction: A comparative study of machine learning methods. *Journal of Forecasting*.

https://doi.org/10.1002/for.2661

Tsai, C.-F., Hsu, Y.-F., & Yen, D. C. (2014). A comparative study of classifier ensembles for bankruptcy prediction. *Applied Soft Computing Journal*, *24*(C), 977–984. https://doi.org/10.1016/j.asoc.2014.08.047

Virk.dk. (2020a). *System til system adgang til CVR-data - Virk | Data*. https://data.virk.dk/datakatalog/erhvervsstyrelsen/system-til-system-adgang-til-cvr-data

Virk.dk. (2020b). *System til system adgang til regnskabsdata - Virk | Data*. https://data.virk.dk/datakatalog/erhvervsstyrelsen/system-til-system-adgang-til-regnskabsdata

XBRL. (2020). *The XBRL Standard*. https://specifications.xbrl.org/index.html

Xin, X., & Xiong, X. (2011). Financial Distress Prediction of Chinese-Listed Companies Based on PCA and WNNs. *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)*, *3*(4), 6–14. https://doi.org/10.4018/japuc.2011100102

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems With Applications*, *58*, 93–101. https://doi.org/10.1016/j.eswa.2016.04.001

Zmijewski, M. E. (1984). Methodological Issues Related to the Estimation of Financial Distress Prediction Models. *Journal of Accounting Research*, *22*, 59–82. https://doi.org/10.2307/2490859

11 APPENDICES

11.1 APPENDIX 1 – INTERVIEW TRANSCRIPT HIGHLIGHTS WITH NORDEA

Interviewee – Simon Nissen, Business Developer, 12 years of work experience at Nordea of which the first 9 years were in consumer-facing business units. He is now primarily working on system-technical tasks but is also involved with the Danish credit rating process and is ensuring that it works as intended.

13:20 – **Simon:** "We have our own internal rating models, that we use, and in these, we use quantitative factors. The majority of the factors are pure (ed. financial) numbers. On top of this, we have more soft values, the quantitative values that we use. These concern information on, the owners, the executive board, the industry etc. All this information (ed. quantitative and qualitative) will together form the credit score that goes from 0-7."

14:45 – **Simon:** "We have partnered with companies, that provide data to us, data from financial reports. The data may come from CVR or other data providers, that provide data to us and our (ed. electronic) financial analysis tools. This data will be automatically read by our tools."

15:15 – **Simon:** "So if we have a company that we need to rate, then we can go in (ed. in their systems), and source data from that company. Then I get the external financial report, without typing anything (ed. into the analysis tool)."

16:50 – **Frederik:** "What about the qualitative aspects (ed. in the credit rating process)? How much can they influence the ratings?

Simon: "… They have an influence, but they influence with at most one grade"

11.2 Appendix 2 – Unsupervised Learning

While supervised machine learning models train using labeled data, unsupervised machine learning models train on *unlabeled* data (Fawcett & Provost, 2013). As an example of an *unlabeled* dataset, note Figure 39 below. This figure is identical to Figure 2, but the target variable is removed.

ProfitLoss	NoncurrentAssets	CurrentAssets	Assets	ContributedCapital	RetainedEarnings
-10499000.00	115242000.00	5690000.00	120932000.00	250000.00	17330000.00
-5069228.00	117850520.00	956125.00	118806645.00	250000.00	12260979.00
2464711.00	117004425.00	1676221.00	118680646.00	250000.00	14725690.00
-10665079.00	115938617.00	937368.00	116875985.00	250000.00	4060611.00
-1714640.00	42467709.00	5908175.00	48375884.00	250000.00	2345971.00

Figure 39 – Example data for unsupervised machine learning

Compared to supervised machine learning, which attempts to learn patterns in the data to accurately predict or classify data points, unsupervised learning focuses more on exploratory data analysis. There are five primary uses for unsupervised learning: Clustering, density estimation, anomaly detection, association mining, and dimensionality reduction. Unsupervised learning is not employed in the analysis of financial distress prediction in this thesis; hence the following only briefly outlines one of the five subsets of unsupervised machine learning, i.e., *clustering*, to provide the reader with the two contrasting methodologies of unsupervised and supervised learning.

Clustering is a machine learning technique used to cluster data into groups. There are several algorithms that can do this effectively for different types of data (Geron, 2017). As an example, imagine a different dataset on retail customers, where each customer is visualized as a dot in Figure 40. The left area of the figure illustrates the original uncategorized dataset (no colors). Following a clustering technique, the customers have now been categorized into three classes as indicated by the three colors in the right side of the figure. Following the categorization of the data into these three groups, a retailer might be able to identify that these three types of customers warrant different marketing strategies to be successful. For the data relating to this thesis, a clustering technique could similarly categorize financial statements into groups based on similar traits. While the data visualized below is two-dimensional, clustering can be done for any n-dimensional data.

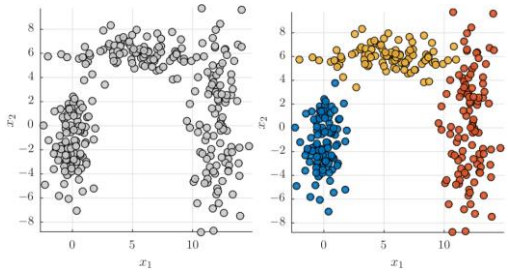

Figure 40 - Example of clustering data into groups from Herlau et al. (2018)

11.3 APPENDIX 3 – QUERIED PERMANENT DATABASE VARIABLES

Type of information	Feature name	Description
Company	CVR	Company registration No.
	enhedsNummer	Unique database ID
	samtID	Revision number
	nyesteNavn	Latest company name
	status, periode	Status (Active, bankrupt) for a given period
	kortBeskrivelse, periode	Legal form (A/S, ApS)
	branchekode, periode	Industry code
	livsforloeb	Foundation date and possible cessation date
	kommunekode, periode	Municipality No.
Participants	enhedsNummer	Unique database ID
	forretningsnoegle	CVR No. if the participant is a company
	enhedsNummerOrganisation	Unique database ID of organization No.
	medlemsData.attributter	The data values contained for the query

Table 9 Permanent DB variables

11.4 APPENDIX 4 – COMPANY INFORMATION (DICTIONARY)


```
"5": {
        "Adresse": [
            {
                "kommune": {
                    "kommuneKode": 661
                },
                "periode": {
                    "gyldigFra": "2008-01-08",
                    "gyldigTil": "2010-08-03"
                }
            }
        ],
        "Branche": [
            {
                "branchekode": "821100",
                "periode": {
                    "gyldigFra": "2008-01-08",
                    "gyldigTil": "2010-08-03"
                }
            }
        ],
        "CVR": 31172772,
        "EnhedsNr": 4000614889,
        "Liv": [
            {
                "periode": {
                    "gyldigFra": "2008-01-08",
                    "gyldigTil": "2010-08-03"
                }
            }
        ],
        "Navn": "KODIF ISLAND EXPRESS ApS",
        "SamtID": 5,
        "Senestestatus": "TVANGSOPL\u00d8ST",
        "Status": [
            {
                "periode": {
                    "gyldigFra": "2008-01-08",
                    "gyldigTil": "2010-07-22"
                },
                "status": "NORMAL"
            },
            {
                "periode": {
                    "gyldigFra": "2010-07-23",
                    "gyldigTil": "2010-08-02"
                },
                "status": "UNDER TVANGSOPL\u00d8SNING"
            },
            {
                "periode": {
                    "gyldigFra": "2010-08-03",
                    "gyldigTil": "2010-08-03"
                },
                "status": "TVANGSOPL\u00d8ST"
            }
        ],
        "Virksomhedsform": [
            {
                "kortBeskrivelse": "APS",
                "periode": {
                    "gyldigFra": "2008-01-08",
                    "gyldigTil": "2010-08-03"
                }
            }
    }
```


Figure 41 – Fundamental company information a dictionary format

11.5 Appendix 5 – List of Initial Selected Financial Features

Financial features	
AccumulatedImpairmentLossesAndAmortisationOfIntangibleAssets	LongtermLiabilitiesOtherThanProvisionsDueInOneYear
AccumulatedImpairmentLossesAndDepreciationOfInvestments	LongtermMortgageDebt
AccumulatedImpairmentLossesAndDepreciationOfPropertyPlantAndEquipment	LongtermReceivablesFromAssociates
AccumulatedRevaluationOfPropertyPlantAndEquipment	LongtermReceivablesFromGroupEnterprises
AccumulatedRevaluationsOfInvestments	ManufacturedGoodsAndGoodsForResale
AcquiredIntangibleAssets	MinorityInterests
AdditionsToIntangibleAssets	NetIncreaseDecreaseInCashAndCashEquivalents
AdditionsToInvestments	NominalValueOfIssuedShares
AdditionsToPropertyPlantAndEquipment	NoncurrentAssets
Adjustments	NoncurrentBankLoans
AdjustmentsForCurrentTaxOfPriorPeriod	NoncurrentLiabilities
AdjustmentsForDecreaseIncreaseInWorkingCapital	NoncurrentReceivables
AdjustmentsForDeferredTax	NoncurrentReceivablesDueFromRelatedParties
AdjustmentsOfHedgingInstrumentsAtFairValue	NumberOfEmployees
AdministrativeExpenses	NumberOfIssuedShares
AmortisationOfGoodwillOfInvestments	OperatingMargin
AmortisationOfIntangibleAssets	OpinionOnAuditedFinancialStatements
Assets	OtherAdjustmentsOfFinanceExpenses
AverageNumberOfEmployees	OtherAdjustmentsOfFinanceIncome
BiologicalAssets	OtherAdjustmentsRelatedToInvestments
CashAndCashEquivalents	OtherCurrentPayables
CashAndCashEquivalentsConcerningCashflowStatement	OtherEmployeeExpense
CashCapitalIncrease	OtherExpenseByNature
CashFlowFromOperatingActivitiesBeforeFinancialItems	OtherExternalExpenses
CashFlowsFromUsedInFinancingActivities	OtherFinanceExpenses
CashFlowsFromUsedInInvestingActivities	OtherFinanceIncome
CashFlowsFromUsedInOperatingActivities	OtherFinanceIncomeFromGroupEnterprises
ContractWorkInProgress	OtherInterestExpenses
ContributedCapital	OtherInterestIncome
CopyrightsPatentsAndOtherIndustrialPropertyRightsServiceAndOperatingRights	OtherLongtermInvestments
CurrentAssets	OtherLongtermPayables
CurrentBankLoans	OtherLongtermReceivables
CurrentDeferredTaxAssets	OtherOperatingExpenses
CurrentLiabilities	OtherRegulationsDevaluations
CurrentReceivablesFromSubsidaries	OtherRegulationsImpairmentLossesAndDepreciations
CurrentTaxExpense	OtherReserves
DateOfApprovalOfReport	OtherShorttermInvestments
DeferredIncomeAssets	OtherShorttermPayables
DepreciationAmortisationExpenseAndImpairmentLossesOfPropertyPlantAndEquipmentAndIntangibleAssetsRecognisedInProfitOrLoss	OtherShorttermReceivables
DepreciationOfPropertyPlantAndEquipment	PaidContributedCapital
DisposalsOfIntangibleAssets	PlantAndMachinery
DisposalsOfInvestments	PostemploymentBenefitExpense
DisposalsOfPropertyPlantAndEquipment	PrepaymentsForPropertyPlantAndEquipment
Dividend	ProceedsFromLongtermLiabilitiesClassifiedAsFinancingActivities
DividendIncomeRelatedToInvestments	ProfitLoss
DividendPaid	ProfitLossAfterAttributableToMinorityInterest
EmployeeBenefitsExpense	ProfitLossAttributableToMinorityInterest
Equity	ProfitLossFromOrdinaryActivitiesAfterTax
EquityRatio	ProfitLossFromOrdinaryActivitiesBeforeTax
EquityTransfersToReserves	ProfitLossFromOrdinaryOperatingActivities
ExchangeRateAdjustmentsOtherFinanceExpenses	ProfitLossRelatedToInvestments

ExchangeRateAdjustmentsOtherFinanceIncome	ProfitLossRelatedToInvestmentsImpairmentLossesAndDepreciation
ExchangeRateLoss	PropertyCost
ExchangeRateProfit	PropertyPlantAndEquipment
ExtraordinaryDividendPaid	PropertyPlantAndEquipmentGross
FeesForAuditorsPerformingStatutoryAudit	ProposedDividendRecognisedInEquity
FeesForAuditorsPerformingTaxConsultancy	Provisions
FeesForOtherServicesPerformedByAuditors	ProvisionsForDeferredTax
FinanceCosts	PurchaseOfIntangibleAssetsClassifiedAsInvestingActivities
FinanceExpensesArisingFromGroupEnterprises	PurchaseOfInvestments
FinanceIncome	PurchaseOfPropertyPlantAndEquipmentClassifiedAsInvestingActivities
FixturesFittingsToolsAndEquipment	RaisingOfDebtToCreditInstitutions
GainsLossesFromCurrentValueAdjustmentsOfOtherInvestmentAssets	RaisingOfLongtermDebt
Goodwill	RawMaterialsAndConsumables
GrossMargin	RepaymentsOfLongtermLiabilitiesClassifiedAsFinancingActivities
GrossProfitLoss	ReportingPeriodEndDate
GrossResult	ReportingPeriodStartDate
IdentificationNumberCvrOfAuditFirm	ReserveAccordingToArticlesOfAssociation
IdentificationNumberCvrOfReportingEntity	ReserveForNetRevaluationAccordingToEquityMethod
IdentificationNumberCvrOfSubmittingEnterprise	RestOfOtherFinanceExpenses
IdentificationNumberCvrOfSubmittingEnterprise	RestOfOtherReserves
IdentificationNumberPnrOfAuditFirm	ResultsFromNetFinancials
ImpairmentLossesOfIntangibleAssets	RetainedEarnings
ImpairmentLossesOfInvestments	ReturnOnCapitalEmployed
ImpairmentOfFinancialAssets	ReturnOnEquity
IncomeFromInvestmentsInAssociates	RevaluationsOfPropertyPlantAndEquipment
IncomeFromInvestmentsInGroupEnterprises	Revenue
IncomeFromOtherLongtermInvestmentsAndReceivables	ReversalsOfImpairmentLossesAndAmortisationOfDisposedIntangibleAssets
IncomeTaxesPaidRefundClassifiedAsOperatingActivities	ReversalsOfImpairmentLossesAndDepreciationOfDisposedPropertyPlantAndEquipment
IncomeTaxExpenseContinuingOperations	SaleOfInvestments
IncreaseDecreaseOfImpairmentLossesAndAmortisationOfIntangibleAssetsThroughNetExchangeDifferences	ShareHeldByEntityOrConsolidatedEnterprisesInRelatedEntity
IncreaseDecreaseOfImpairmentLossesAndDepreciationOfPropertyPlantAndEquipmentThroughNetExchangeDifferences	SharePremium
IncreaseDecreaseOfIntangibleAssetsThroughNetExchangeDifferences	ShorttermDebtToBanks
IncreaseDecreaseOfInvestmentsThroughNetExchangeDifferencesEquity	ShorttermInvestments
IncreaseDecreaseOfPropertyPlantAndEquipmentThroughNetExchangeDifferences	ShorttermLiabilitiesOtherThanProvisions
IncreaseDecreaseOfPropertyPlantAndEquipmentThroughTransfers	ShorttermMortgageDebt
IncreaseOfCapital	ShorttermPartOfLongtermLiabilitiesOtherThanProvisions
IntangibleAssetsGross	ShorttermPayablesToAssociates
InterestExpenseAssignedToGroupEnterprises	ShorttermPayablesToGroupEnterprises
InterestExpensePartOfCostOfAsset	ShorttermPayablesToShareholdersAndManagement
InterestIncomeFromGroupEnterprises	ShorttermPrepaymentsReceivedFromCustomers
InterestPaidClassifiedAsOperatingActivities	ShorttermReceivables
InterestReceivedClassifiedAsOperatingActivities	ShorttermReceivablesFromAssociates
Inventories	ShorttermReceivablesFromGroupEnterprises
InvestmentInPropertyPlantAndEquipment	ShorttermReceivablesFromOwnersAndManagement
InvestmentsGross	ShorttermTaxPayables
InvestmentsWithNegativeEquityDepreciatedOverReceivables	ShorttermTaxReceivables
LandAndBuildings	ShorttermTradePayables
LeaseholdImprovements	ShorttermTradeReceivables

LiabilitiesAndEquity	SocialSecurityContributions
LiabilitiesOtherThanProvisions	TaxExpense
LongtermDebtToBanks	TaxExpenseOnOrdinaryActivities
LongtermDebtToOtherCreditInstitutions	TradeAndOtherCurrentReceivables
LongtermInvestmentsAndReceivables	TradeAndOtherCurrentReceivablesDueFromRelatedParties
LongtermInvestmentsInAssociates	TradeAndOtherReceivables
LongtermInvestmentsInGroupEnterprises	TradeAndOtherReceivablesDueFromRelatedParties
LongtermLiabilitiesOtherThanProvisions	ValueAdjustmentsOfEquity
LongtermLiabilitiesOtherThanProvisionsDueAfterFiveYearsAndMore	ValueOfKeyFigureOrFinancialRatio
LongtermLiabilitiesOtherThanProvisionsDueBetweenOneAndFiveYears	WagesAndSalaries

11.6 Appendix 6 - Example of a Reference Map

{'c1': '2012',
'c4': '2011',
'c5': '2011',
'c7': '2012',
'c33': '2012',
'c49': '2012',
'c50': '2012',
'c51': '2012',
'c72': '2012',
'c76': '2012',
'c106': '2012',
'c157': '2012',
'c158': '2011',
'c179': '2012',
'c180': '2011',
'c208': '2011',
'c209': '2012',
'c210': '2012',
'c214': '2011',
'c215': '2012',
'c216': '2012',
'c217': '2011',
'c218': '2012',
'c219': '2012',
'c247': '2011',
'c248': '2012',
'c249': '2012',
'c286': '2011',
'c287': '2012',
'c288': '2012',
'c292': '2011',
'c293': '2012',
'c294': '2012',
'c362': '2012',
'c364': '2011'}

Figure 42 – Reference map

11.7 APPENDIX 7 – LIST OF SELECTED VARIABLES[40]

	Count		Count
AccountsPayable	557062	ROE	740687
AccountsReceivable	494247	LongTermBankDebt	39596
cash	660890	LongtermMortgageDebt	125389
CorporationTax	743607	LongTermDebt	264831
CurrentAssets	709752	TotalMortgageDebt	29752
Depreciation	5240	QuickRatio	284574
EBIT	627359	InvestedCapital	262905
Equity	742792	EquityInvestedCapital	262313
ExpectedDividends	222790	FinancialAssets	660890
FinanceIncome	926	x1	699966
FinanceCosts	954	x2	723659
Inventories	293106	x3	623675
LandAndBuildings	170351	x4	261244
liquidAssets	660890	Altman_Z	227896
logSize	743607	ownerRisk	632841
Profit	741452	Target	743607
InterestCoverageRatio	924	LogAge	741737
OtherOperatingExpenses	743607	Sector	742797
OtherReceivables	498567	Adress	742753
PersonnelCosts	486764	LogChange	743607
Prepayments	311668	RelativeDebtChange	356033
Provisions	298881	RelativeLiabilityChange	734693
ReceivablesFromRelatedParties	4895		
RetainedEarnings	728851		
TangibleAssets	504109		
TaxExpenses	341947		
TotalReceivables	52		
ShortTermBankDebt	160		
ShorttermMortgageDebt	34314		
OtherShortTermDebts	696563		
ShortTermDebt	704812		

[40] For detailed calculations of each variable, please go to the following in the code repository /Code/Main/2_Add50Variables.ipynb

11.8 APPENDIX 8 – RESULTS OF RANDOM SEARCH

	Dense				Sparse		
LR hyper-parameter	**LR**	**LR-CODR**	**GBT**	**GBT-CODR**	**GBT**	**GBT-CODR**	**GBT hyper-parameters**
C	0.017	0.017	22.56	22.56	24.29	24.29	scale_pos_weight
	-	-	11	11	50	50	n_estimators
	-	-	5	5	5	5	max_depth
	-	-	0.01	0.089	0.092	0.098	learning_rate
	-	-	4.446	1.517	0.457	0.377	gamma
	-	-	0.533	0.351	0.367	0.538	subsample
	-	-	0.965	0.961	0.882	0.895	colsample_bytree