Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Emil Nylén-Forthun, Mats Møller,
Nils-Gunnar Birkeland Abrahamsen

Financial Distress Prediction Using Machine Learning and XAI: Developing an Early Warning Model for Listed Nordic Corporations

TIØ4900 - Financial Engineering, Master's Thesis

Master's thesis in Industrial Economics and Technology Management
Supervisor: Sjur Westgaard
Co-supervisor: Petter Eilif de Lange
June 2022

NTNU
Norwegian University of
Science and Technology

Emil Nylén-Forthun, Mats Møller,
Nils-Gunnar Birkeland Abrahamsen

Financial Distress Prediction Using Machine Learning and XAI: Developing an Early Warning Model for Listed Nordic Corporations

TIØ4900 - Financial Engineering, Master's Thesis

Master's thesis in Industrial Economics and Technology Management
Supervisor: Sjur Westgaard
Co-supervisor: Petter Eilif de Lange
June 2022

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Preface

This paper concludes our Master of Science degree in Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU) in the spring of 2022. The paper proposes an explicable early warning model for financial distress prediction for listed Nordic corporations using machine learning and explainable artificial intelligence. It is a fully independent work conducted by Emil Nylén-Forthun, Mats Møller, and Nils-Gunnar Birkeland Abrahamsen.

The motivation behind this thesis is rooted in our passion for machine learning, data science, and financial risk management. We firmly believe that transparency in financial systems is essential to ensure safe, sustainable, and efficient markets. By combining a series of analytical elements, we provide a novel contribution to the existing financial distress prediction literature. We strongly recognize the work of those preceding our own and hope to contribute with insightful and productive observations to the existing body of research.

We wish to express our gratitude toward Ph.D. candidate Morten Risstad of Sparebank 1 Markets for assistance in collecting data for our analysis and valuable input during discussions. Finally, we thank our supervisors, Professor at NTNU Sjur Westgaard and Associate Professor at NTNU Petter Eilif de Lange, for their guidance.

Abstract

This paper proposes an explicable early warning model for financial distress which generalizes across listed Nordic corporations. A Light Gradient Boosting Machine (LightGBM), an Artificial Neural Network (ANN) and a benchmark Logistic Regression (LR) model are applied to a dataset consisting of quarterly accounting data, information from financial markets, and indicators of macroeconomic trends. After cleaning, the dataset includes 639 listed Nordic companies in the period Q1 2001 to Q2 2022. LightGBM proves to be the superior model, achieving a ROC-AUC score of 0.93 and an F1 score of 0.63, surpassing the benchmark model by a significant margin.

In addition, we propose an end-to-end framework for data collection and pre-processing, providing a transparent data treatment procedure which is replicable for both academic and industrial purposes. We apply a proxy-based definition of financial distress in line with financial intuition and industry practices. This proxy is based on measures of solvency (Interest Coverage Ratio) and liquidity (Current Ratio), in accordance with standard bond and loan covenants.

Feature selection and model explanation are performed using the explainable AI framework SHAP. Results clearly show that features related to liquidity, solvency, and size are highly important to the output. The analysis also uncovers that including seasonality, macro and market information proves advantageous due to interaction effects with other variables.

The combination of the following aspects sets this paper apart: (i) employing quarterly data, allowing us to study the problem on a finer scale and capture seasonality effects, which has rarely been recorded in related literature before, (ii) the geographical area of study (the Nordics), and (iii) the inclusion of macro and market variables which capture the environment surrounding the company. In addition, this paper offers a comparison of state-of-the art machine learning models and a renowned benchmark model, contributing to the body of comparative literature.

Sammendrag

Denne oppgaven lanserer en forklarbar modell for tidlig varsling av økonomisk vanskeligstilthet hos børsnoterte selskaper som er generaliserbar på tvers av landegrenser i Norden. En Light Gradient Boosting Machine (LightGBM) og et kunstig nevralt nettverk måles opp mot en logistisk regresjonsmodell som referanse på et datasett bestående av kvartalsvis regnskapsdata, informasjon fra finansmarkedene og makroøkonomiske indikatorer. Etter datavask omfatter datasettet totalt 639 børsnoterte nordiske selskaper i perioden Q1 2001 til Q2 2022. LightGBM utkonkurrerer de andre modellene, og med en ROC-AUC-poengsum på 0.93 og F1-poengsum på 0.63, overgår referansemodellen med en betydelig margin.

Vi fremlegger også et ende-til-ende-rammeverk for innsamling og prosessering av data som kan gjenskapes for både akademiske og industrielle formål. Vi bruker en proxybasert definisjon på økonomisk vanskeligstilthet som er i tråd med finansiell intuisjon og bransjepraksis. Denne proxyen er basert på forholdstall for solvens (rentedekningsgrad) og likviditet (likviditetsgrad 1), og i samsvar med standard måltall for obligasjons- og låneforpliktelser.

Valg av inputvariabler og modellforklaring gjøres ved bruk av SHAP, et rammeverk for forklarbar kunstig intelligens, og resultatene viser tydelig at variabler relatert til likviditet, solvens og selskapsstørrelse er av stor betydning for modellens output. Analysen avdekker også at inkludering av sesongmessighet, samt makro- og markedsinformasjon viser seg å være fordelaktig grunnet interaksjonseffekter med andre variabler.

Kombinasjonen av følgende elementer skiller denne oppgaven fra tilsvarende arbeid: (i) Observasjon av sesongmessig variasjon på kvartalsbasis, noe som sjeldent belyses i relatert litteratur, (ii) det geografiske fokusområdet (Norden), (iii) inkludering av makro- og markedsvariabler som fanger selskapets omgivelser. I tillegg sammenlikner denne oppgaven to toppmoderne maskinlæringsmodeller og en anerkjent referansemodell, noe som bidrar til den komparative litteraturen.

Table of Contents

List of Figures v

List of Tables v

List of Abbreviations vi

1 Introduction and Background 1

2 Literature Review 3

2.1 Distress Prediction - Seminal Research 3

2.2 An Overview of Methods and Techniques Used in Distress Prediction 4

2.2.1 Traditional Statistical Distress Prediction Models 4

2.2.2 Machine Learning Techniques 4

2.3 Explainable AI 5

2.4 Relevant Features: Financial Ratios and Other Drivers of Distress 6

2.4.1 Financial Variables 6

2.4.2 Market and Macro Variables 7

2.4.3 Feature Selection Using XAI 7

2.5 Related Research and Contribution to Literature 8

3 Dataset 9

3.1 Data Collection 9

3.2 Feature Selection 12

3.3 Definition of Financial Distress 13

3.4 Data Cleaning and Missing Value Handling 14

3.5 Sample and Target Generation 15

3.5.1 Data Analysis and Outlier Handling 16

3.6 Addressing Data Imbalance 18

3.7 Final Dataset 18

4 Models and Methodologies 20

4.1 Models and Hyperparameters 20

4.1.1 Artificial Neural Network 20

4.1.2 Light Gradient Boosting Machine 23

4.1.3 Logistic Regression 24

4.2 SHAP Framework 25

4.3 Measuring Model Performance 25

5 Results 27

5.1 Results from Hyperparameter Tuning 27

5.1.1 Tuning of ANN 27

5.1.2 Tuning of LightGBM 28

5.1.3 Tuning of LR 29

5.2 Model Results on Test Set 29

5.3 Result Interpretation with SHAP 31

5.3.1 Global Explanations with SHAP Beeswarm and Dependence Plots 31

5.3.2 Local Explanations with SHAP Waterfall Plots 34

6 Conclusion and Further Work 36

Reference List 37

Appendix 42

A Logistic Regression 42

B Artificial Neural Networks 44

C Light Gradient Boosting Machine 46

D Frequency of Financial Ratios in Literature 48

E Beta Variable Correlation 49

F Features Omitted Following SHAP Feature Selection 50
G Correlation Heatmap 51
H WandB Hyperparameter Configurations Overview - ANN 52
I WandB Hyperparameter Configurations Overview - LightGBM 53
J Dependence Plots for Categorical, Macro and Market Variables 54

List of Figures

1 SHAP beeswarm plot showing the 34 most important features 13
2 Partial and complete samples before and after handling missing values 15
3 Process for generating training samples 15
4 Boxplots of P/E ratio and WCTA ratio both grouped by target 16
5 Visualization of ELU and ReLU activation functions 21
6 Confusion matrix 26
7 Hyperparameter importance for ANN from WandB 27
8 Hyperparameter importance for LightGBM from WandB 28
9 ROC and Precision-Recall curves for ANN, LightGBM and LR 30
10 SHAP beeswarm plot of model predictions on the test set 32
11 Dependence plots for Working Capital, CR, ICR and ROA 33
12 Waterfall plot for a true positive example 34
13 Waterfall plot for a true negative example 35
14 Conversion process for LR 42
15 Illustration of an ANN 44
16 Level vs. leaf-wise growth for LightGBM 46
17 Frequency of financial ratios applied in literature between 1930 and 2007 48
18 Scatterplot of Market Beta to Industry Beta 49
19 SHAP beeswarm plot displaying the 30 features of least importance 50
20 Correlation heatmap for the final 32 numerical features 51
21 Overview of top 500 different ANN configurations using WandB 52
22 Overview of top 500 different LightGBM configurations using WandB 53
23 Dependence plot for categorical, macro and market variables 54
24 Dependence plot for GDP Growth 54

List of Tables

1 Number of companies per country before and after data cleaning 9
2 Financial variables included in initial data query from Bloomberg 9
3 Explanatory variables selected for this paper 10
4 GICS industry sectors and their corresponding MSCI Europe indices 11
5 Descriptive statistics for the features included in the final dataset 17
6 Distribution of minority and majority classes in final train and test set. 18
7 Hyperparameter values tested during ANN tuning 22
8 Hyperparameter values tested during LightGBM tuning 24
9 Final hyperparameter values for ANN 28
10 Final hyperparameter values for LightGBM. 29
11 Precision, recall, F1 and ROC-AUC scores for all three models on the test set . . . 29

List of Abbreviations

Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under Curve
BEP	Basic Earning Power
CR	Current Ratio
DA	Discriminant Analysis
DART	Dropout meet Multiple Additive Regression Trees
DT	Decision Tree
EBF	Exclusive Feature Bundling
EBIT	Earnings Before Interest and Taxes
EBITA	Earnings Before Interest and Taxes to Total Assets
EBITDA	Earnings Before Interest Taxes Depreciations and Amortizations
ELU	Exponential Linear Unit
EPS	Earnings Per Share
FN	False Negative
FP	False Positive
GBDT	Gradient Boosting Decision Trees
GDP	Gross Domestic Product
GICS	Global Industry Classification Standard
GOSS	Gradient-Based One-Sided Sampling
ICR	Interest Coverage Ratio
kNN	k-Nearest Neighbors
LightGBM	Light Gradient Boosting Machine
LIME	Local Interpretable Model-Agnostic Explanations
LPM	Linear Probability Modeling
LR	Logistic Regression
LSTM	Long Short-Term Memory
MART	Multiple Additive Regression Trees
MDA	Multiple Discriminant Analysis
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron

MSCI	Morgan Stanley Capital International
N/A	Not Applicable/Not Available
OLS	Ordinary Least Squares
P/B	Price to Book
P/E	Price to Earnings
P/S	Price to Sales
PA	Probit Analysis
ReLU	Rectified Linear Unit
RF	Random Forest
RMSprop	Root Mean Square Propagation
ROA	Return On Assets
ROC	Receiver Operating Characteristic
ROE	Return On Equity
SGB	Stochastic Gradient Boosting
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UDA	Univariate Discriminant Analysis
WandB	Weights and Biases
WCTA	Working Capital to Total Assets
XAI	Explainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting

1 Introduction and Background

Since the late 1960s, financial distress prediction has become increasingly popular in research, and as the global economy has grown in complexity and interconnectedness, it has become progressively important (Balcaen and Ooghe, 2006; Xu and Wang, 2009). The financial crisis of 2007-2009 is perhaps the most prominent example of how flawed company evaluations have led to the demise of giants. The need for quantitative models that can interpret patterns in company behaviour beyond what can be recorded by simple and rigid methodologies has prompted the use of various intelligent systems, with Machine Learning (ML) now representing state of the art. The application domain of ML has expanded rapidly since its introduction in finance around the 1980s, and today's applications include algorithmic trading, forensic services, portfolio management, and financial forecasting.

Beaver (1966) authored one of the most prominent papers in the field of financial distress prediction. He defined a company's failure as the inability to meet its financial obligations as they mature, citing bankruptcy, bond default, an overdrawn bank account, and the nonpayment of a preferred stock dividend as examples of failure. In each of these scenarios, the state that is being predicted is linked to a specific credit event. However, rather than predicting specific credit events associated with failure, we recognize the value of being able to identify early-stage signals of enterprises in distress. This question has been examined in the literature, with Balcaen and Ooghe (2006) highlighting concerns with juridical definitions of failure, that is, failure based on legally recorded one-time incidents. Such definitions of distress or failure are inherently prone to noise caused by, for instance, companies filing for bankruptcy despite displaying no real evidence of difficulty, but rather as a strategic move. Financial distress, according to Malakauskas and Lakstutiene (2021), is described as a circumstance in which a company faces a considerable struggle in covering its liabilities, rather than the specific risk of going bankrupt or defaulting on a payment, as has been the focus of previous research. These considerations led to the development of our proxy definition of distress, which is rooted in key financial figures. Further details are disclosed in Section 3.3.

Corporate distress prediction relies on data that reflects a company's true situation. As the extensive body of current literature shows, accounting data derived from financial statements plays a critical role in this regard. However, as pointed out by Jan (2021), relying solely on information from financial statements in distress prediction is risky due to information asymmetry, and the authors draw a parallel to the famous "lemon market" theory by Akerlof (1978). Despite targeting different areas of business in their research, Akerlof and Jan both address the same core issue, namely the difficulty of distinguishing good from bad business in a world of asymmetric information spread. Company management obviously possesses different information about the company than external stakeholders when it comes to predicting corporate distress. As a result, before the release of financial statements, investors may not be aware of a company's true financial situation. Furthermore, as Chava and Jarrow (2004) point out, businesses may use different accounting practices, meaning that the probability of financial distress can differ for companies with otherwise similar financial statements. This encourages the expansion of the feature space of prediction models to include market and macro variables that indicate company relationships other than those available from periodically disclosed balance sheets and income statements.

Artificial Intelligence (AI) is a core technology of the ongoing technological revolution and industrial transformation and is deemed to play a vital role in the future of finance (Zheng et al., 2019). Although ML models have proven quite accurate predictors of distress, often superior to statistical models, they are somewhat limited by their "black box"-nature. The term "black box" refers to a model's inability to reveal internal mechanisms and nuances to their predictions (Kamath and Liu, 2021). Further, certain areas of business, such as finance, healthcare and security, have specific requirements for transparency and explainability of models, so any potential benefit of increased prediction accuracy may not outweigh the disadvantage of lacking transparency. This has given rise to several approaches to provide explainability in complex ML models in recent years, and subsequently, many publications of research targeting the field of finance. Furthermore, in recent years, guidelines and regulations targeting AI and ML, and other technologies used in automated decision making, have become prominent. For instance, the Ethics guidelines for trustworthy AI set by the European Commission (2019) present seven requirements that AI systems must meet to be considered trustworthy, of which several can be linked to the topic of explainable AI (XAI). Hence, adapting model explainability is not only necessary to sustain a broad degree of trustworthiness in ML-predictors among financial users, but also increasingly becoming a legal requirement.

In today's literature, financial distress prediction is often analyzed on a micro or lower macroeconomic scale, evaluating either a set of companies within a single country or a specific industry. We argue, however, that a localized scope may be too narrow given the increasingly globalized economy and the scarcity of distressed companies. In this study, we examine ML-based distress prediction models using a unique, Nordic dataset. Due to their financial structures, financial institutions such as banks, insurance firms, and sovereign funds have been excluded from the analysis. Throughout this section, we have motivated the need for an early warning model for financial distress which generalizes across borders. The objective of this paper is to develop an explainable early warning model for financial distress, examining the financial status of listed Nordic Corporations based on accounting data and information about financial markets and macroeconomic trends. In addition, we propose an end-to-end framework for industrial and academic replicability by thoroughly explaining the process of collecting and processing data from a financial database to create a unique and robust dataset for fully quantitative analyses.

The remainder of this paper is organized as follows: Section 2 provides an overview of relevant literature and research. Section 3 explains the dataset, including pre-processing, feature selection, and analysis. Section 4 elaborates on the models and hyperparameters applied, as well as metrics used to determine the performance of the different models. Section 5 provides results from hyperparameter tuning and final test results, as well as model explanations using the SHAP framework, and Section 6 presents our conclusion and suggestions for further research.

2 Literature Review

The purpose of this literature review is to provide an overview of the background and evolution of corporate distress prediction. We provide context for the work underlying this paper by highlighting seminal papers, state-of-the-art research, and popular methods. The drivers of corporate distress, as well as the features used to identify distressed firms, are also examined. Finally, we discuss how our study contributes to the body of research.

2.1 Distress Prediction - Seminal Research

Beaver (1966) was one of the first to establish a connection between empirical financial ratio analysis and the creditworthiness of companies. He noted the ratios' ability to recognize financial illness in a company but emphasized that their predictive effectiveness varied. In his paper, Beaver classified thirty different ratios into six distinct categories. The selection of ratios was based, in part, on their frequency of occurrence and performance in prior research. The findings of his univariate analysis suggested a correlation between specific financial parameters and the likelihood of future business failure.

Another pioneering paper on failure prediction was published by Altman (1968). His study helped establish the effectiveness of classical ratio analysis as a tool for evaluating business performance at a time when the academic community was moving away from its use. Whereas Beaver (1966) conducted a Univariate Discriminant Analysis (UDA), i.e. analyzing each ratio separately, Altman extended on the principles of classic ratio analysis and utilized Multiple Discriminant Analysis (MDA) to distinguish failing companies from non-failing companies. Altman introduced the Z-score model, which takes numerous financial ratios as inputs and produces a Z-score showing a company's financial strength, specifically its likelihood of bankruptcy. Altman et al. (1977) later evaluated this study and made modifications to the basic Z-score model. However, his original study contributed to reducing the gap between traditional ratio analysis and more rigorous statistical approaches, and it proposed application areas such as corporate credit evaluation, internal control procedures, and investment guidelines.

The two seminal publications listed above are extensively cited in the literature concerning financial distress prediction. Both studies demonstrate how patterns in financial ratios may show symptoms of a company's deterioration and established the groundwork for future failure prediction models and distress research. Nevertheless, both publications suggest relatively straightforward, dichotomous classification techniques. Ohlson (1980) questioned the practical applications of Beaver and Altman's seminal work and claimed that the output from DA could not be intuitively interpreted and that the Z-score had limited relevance in specific financial decision issues. Ohlson argued that his suggested approach, Conditional Logistic Regression, eliminates several MDA-related issues. The output of his models, commonly referred to as O-scores, may be interpreted as a measure of a company's probability of filing for bankruptcy within a specific time period. Although Ohlson's model differed from that of Beaver and Altman, the majority of input variables still consisted of financial ratios derived from accounting data.

It has been shown that observable change in financial ratios over time exhibits predictive power (Luoma and Laitinen, 1991). However, there is also a stochastic element in a company's performance introduced by, for example, industry and market volatility. In recent years, the need for prediction models that can comprehend complex input data and data relations that are unobservable by classic statistical approaches has grown, leading to the development of more intelligent prediction techniques. Intelligent applications in the field of default prediction and risk management became increasingly common in the late 1980s and early 1990s, where methods like Artificial Neural Networks (ANN) (Jensen, 1992; Tam, 1991; Yang et al., 1999) and Decision Trees (DT) (Frydman et al., 1985; Messier Jr and Hansen, 1988) gained popularity in academic research. Since then, there has been tremendous growth in the number of ML approaches created and utilized in distress prediction, many of which are complex and not inherently interpretable. Consequently, the need to explain their logic and way of reasoning has given rise to a number of techniques providing explainability in recent years.

2.2 An Overview of Methods and Techniques Used in Distress Prediction

Distinguishing between classic statistical prediction models and ML models is a common way of classifying the various distress prediction methodologies. Whereas the former focuses on the interpretation of covariate effects between input factors and indicators of financial distress, the latter emphasizes accuracy in predictions and not necessarily the development of explanation (Ogundimu, 2019).

2.2.1 Traditional Statistical Distress Prediction Models

Kim et al. (2020) referred to Discriminant Analysis (DA) as a technique of the "first generation" of literature on corporate default. Beaver (1966) was regarded as a pioneer in the field of corporate distress prediction when he created the UDA model, which was widely regarded as a groundbreaking concept. However, univariate analysis is limited in its practical applications due to the strong assumption of a linear connection between the measures and a company's failure status (Balcaen and Ooghe, 2006). MDA and the Z-score framework developed by Altman (1968), which has been implemented and analyzed in several studies, is much more frequently cited in academic research, see Blum (1974), Dambolena and Khoury (1980), Deakin (1972), Edmister (1972) and Eisenbeis (1977).

In the 1960s and 1970s, DA was the prominent tool for predicting distress, and it dominated the literature on business failure prediction until the 1980s. DA was later superseded by other, statistically less demanding methods (Balcaen and Ooghe, 2006). Among these approaches, some of which are categorized as binary response models (Horowitz and Savin, 2001; Kim et al., 2020) or conditional probability models (Balcaen and Ooghe, 2006; Lin and Piesse, 2004), are Linear Probability Modeling (LPM), Logistic Regression (LR), and Probit Analysis (PA). A binary response model is a regression model with a binary dependent variable, thus characterizing the status of a firm as either normal or distressed (Horowitz and Savin, 2001). LR is by far the most prevalent approach among the three conditional probability models discussed and has become a prominent and highly trusted prediction tool in studies on financial distress (Aziz and Dar, 2006; Balcaen and Ooghe, 2006). Traditional LR is still utilized as a benchmark approach in comparative studies such as Hu and Ansell (2007), Moscatelli et al. (2020), West (2000) and Yeh and Lien (2009), and often performs reasonably well compared to intelligent techniques, which are popularly claimed to be superior to statistical methods.

2.2.2 Machine Learning Techniques

The ability held by machines and intelligent systems within the field of ML to process and model highly complex and large amounts of data has led to their natural occurrence in finance and risk modelling. ANNs are commonly applied in predicting company failure and bankruptcy forecasting and are distinguished for being able to use non-linear equations to develop relationships between input and output variables. ANNs have been applied in multiple papers regarding distress and bankruptcy prediction (Iturriaga and Sanz, 2015; Tam, 1991; Yang et al., 1999) and credit rating/granting (Abiyev, 2014; Falavigna, 2012; Jensen, 1992). ANNs have been criticized in literature due to their "black box" processing approach (Kim et al., 2020), so their usefulness in, for instance, private banking, where the bank needs to provide an elaborate justification for their customer credit assessment, is disputed. As insight into the function it approximates cannot be obtained by simply investigating the network's structure, explanation attempts must be met by targeted techniques and frameworks.

Decision trees (DTs) are ML techniques that use decision rules to recursively partition datasets into simpler subsets. Although it has been argued that DTs are more comprehensible and precise than other ML algorithms (Olson et al., 2012), drawbacks include the risk of overfitting and reliance on large sets of sample data, as pointed out in several review papers on default and bankruptcy prediction (Kim et al., 2020; Kumar and Ravi, 2007). Therefore, ensemble methods that combine the predictive power of several DTs in their decision process have become a highly effective alternative. Two ensemble methods that have become increasingly popular in recent years include Light Gradient Boosting Machine (LightGBM) and Extreme Gradient Boosting (XGBoost), both of which utilize boosting. Boosting is a sequential ensembling technique which combines classifiers by putting more weight on misclassified observations in previous steps and generally outperforms non-boosting ensemble methods such as Random Forest (RF) (Bentejac et al., 2021). Ensemble methods which utilize boosting represent state of the art within corporate financial distress prediction and have proven effective predictors in several studies (Qian et al., 2022; Son et al., 2019). The advantages of XGBoost, proposed by T. Chen and Guestrin (2016), includes relatively low computational complexity, being easily adjustable in terms of hyperparameters and being highly accurate, which makes it suitable for prediction problems (Du et al., 2020). However, XGBoost can be still be quite computationally expensive, so LightGBM was developed by Ke et al. (2017) to further reduce computational complexity.

In a study targeting financial distress prediction for Chinese companies, Qian et al. (2022) use a wrapper-based feature selection process and show that XGBoost and LightGBM both generally outperform other classification methods such as LR, ANN, Support Vector Machine (SVM) and RF. A comparison of the results obtained with and without feature selection imposed on the data testifies to the selection's positive effect on model performance. In a bankruptcy prediction study by Park et al. (2021) targeting Korean companies, LightGBM slightly outperformed XGBoost in terms of ROC-AUC score. Results from feature importance analysis with XAI method Local Interpretable Model-Agnostic Explanations (LIME) showed that LightGBM was more consistent than XGBoost in identifying important features. Further, LightGBM has proven to be superior to other boosting methods in terms of computation speed, which is a great advantage when dealing with large amounts of data, see for instance Shehadeh et al. (2021) and Bentejac et al. (2021), or when training several different model configurations.

Through our literature study, it has become clear that ML methods generally outperform traditional methods in classification and prediction problems. Moscatelli et al. (2020, p. 10) argued that the superiority of ML methods is attributable to their "ability of capturing more precisely the complex relationship between the available firms' indicators and the default outcome [...]". However, they added that additional measures might be needed for accurate assessments, as ML models can be quite non-transparent, and their behaviour can be complex to comprehend. In this paper, we compare the predictive power of three different methods: ANN, LightGBM and LR. The former two represent cutting-edge ML algorithms, and the latter is a trusted traditional classifier used as a benchmark model in this paper. These methods all have a proven track record within the field of financial distress prediction. The exact model versions employed in this study are explained in greater detail in Section 4.

2.3 Explainable AI

Following the various ML methods that have emerged in recent years, research on XAI has gained popularity in the literature. However, interpretability and transparency in prediction models have been discussed long before intelligent prediction systems were first introduced. For example, the simple decision tree, which can be traced back to the 3rd century AD, has intrinsic interpretability from its visual structure and is one of the earliest examples of an explainable prediction model (Kamath and Liu, 2021). More recent studies preceding what is commonly referred to as XAI for intelligent prediction systems today emerged around 1990 (Buchanan and Shortliffe, 1984; Chandrasekaran et al., 1989; Swartout and Moore, 1993; Wick and Thompson, 1992). Later, specialized frameworks and techniques that aim to induce balance in the trade-off between accuracy and explainability for ML prediction models, thus addressing the "black-box" issue, have become popular subjects of study.

The XAI taxonomy is broad, but the different techniques can roughly be sorted by scope, stage, and model, as outlined by Kamath and Liu (2021). Scope refers to whether the method aims to interpret the ML model behaviour on a local or global scale; the interpretation stage may be pre-model, post model, or intrinsic. Finally, the model category refers to whether the technique is model specific or agnostic. For other ways of categorizing explainability techniques, see taxonomies of Minh et al. (2022) or Arrieta et al. (2020).

Within the domain of local, post model, and model-agnostic techniques, we find the Shapley Additive Explanations (SHAP) framework, which has been applied in various forms in financial prediction. SHAP values, proposed by Lundberg and Lee (2017), are based on Shapley values from game theory, first introduced by Shapley (1953). SHAP values are quantitative measures of feature importance which can be used to explain the output of any ML model and have proven to be consistent with human intuition (Demajo et al., 2020; Lundberg and Lee 2017). For a more detailed description of the SHAP framework, see Section 4.2.

Due to difficulties in exact calculation of SHAP values, many methods to approximate their values have been proposed. For instance, Lundberg and Lee propose SHAP value estimation methods such as KernelSHAP, a model-agnostic implementation which combines the local explanation method LIME of Ribeiro et al. (2016) with Shapley values. KernelSHAP has become a common implementation of SHAP. For instance, it was applied by Mokhtari et al. (2019) to explain the classification of several prediction methods for financial time series, including k-Nearest Neighbours (kNN), SVM, RF, XGBoost and Long Short-Term Memory (LSTM). However, KernelSHAP is relatively inefficient due to high computation complexity (Liu et al., 2022). TreeSHAP, a model-specific method for tree-structured ML models, was proposed by Lundberg et al. (2018) to address the issue of inefficiency and has become a popular tool for calculating SHAP values in ML. Bussmann et al. (2021) applied TreeSHAP for explaining an XGBoost model predicting default risk in Southern European SMEs. The results showed that the most important features for non-defaulting companies were profits before taxes plus interests paid and EBITDA, and the most important feature for defaulting companies was total assets. Arguably, this is in line with human intuition, as pointed to previously.

2.4 Relevant Features: Financial Ratios and Other Drivers of Distress

To preserve objectivity in our models, we sought to apply financial ratios that are both broadly used in literature and justifiable from an accounting perspective. In addition, market and macro variables have been included to address the issue of information asymmetry and to improve the models' predictive power. Depending on the object of study, financial ratios chosen to predict distress vary notably across the literature, and indications of performance vary as studies include different combinations of models and ratios. Simply choosing ratios based on their frequency of occurrence in the literature could be *sufficient* but is somewhat unscientific. In addition, data availability constrains the degree of freedom in this regard. A pragmatic approach is to choose ratios that cover main categories, following the idea of sub-categorizing ratios as done by Beaver (1966). Further, a feature selection process including all initially chosen features will help reduce noise and provide insight into the importance of individual features.

2.4.1 Financial Variables

In a comprehensive survey on business failure, Dimitras et al. (1996) reviewed several journals from the period 1932-1994 and found that from 47 papers, the most important financial ratios could be categorized as solvency and profitability ratios. Working Capital to Total Assets (WCTA) was found to be the single most applied ratio, followed by Total Debt to Total Assets, both solvency ratios. A study by Liang et al. (2016) supports these findings, and the authors argued that solvency and profitability ratios held the most important information in predicting bankruptcy when applying several ML models. Other studies, such as Lin and Piesse (2004) and Xu and Wang 2009 have shed light on operation efficiency, or correspondingly, management *inefficiency*, pointing to how well the firm and its assets are managed. Financial ratios in this category include Retained Earnings to Total Assets and Total Assets Turnover. Appendix D shows an overview of the financial ratios applied most commonly in financial distress literature between 1930 and 2007, given that they have been applied in five studies or more. The list, retrieved from Bellovary et al. (2007), shows that the most common ratios include Net Income to Total Assets (NITA), Current Assets to Current Liabilities (Current Ratio), WCTA, EBIT to Total Assets (EBITA), and Sales to Total Assets. In other words, the leading ratios are mostly related to profitability, liquidity, and efficiency.

Accordingly, we have grouped 12 financial ratios into four categories: liquidity, solvency (sometimes referred to as indebtedness), profitability, and operation efficiency (sometimes referred to as activity or just efficiency). The intention behind this choice is to represent a broad spectrum of company characteristics without including ratios excessively, seeking to avoid correlation and misleading results, as Chen and Shimerda (1981) warn against. Throughout our literature review, it has become evident that companies with low performance within the four chosen categories are likely to experience financial distress.

2.4.2 Market and Macro Variables

Recently, it has become increasingly common to expand the feature space from only including accounting ratios, to also include market dynamics and macroeconomic factors. In addition to substantiating the view that the most informative ratios fall within the categories outlined above, Bonfim (2009) combined firm-level information to macroeconomic dynamics such as GDP growth, industrial production, interest rates, and bond spreads. He found that results improved drastically from their inclusion. GDP growth has been applied in several studies concerning corporate distress and has proven its predictive ability (Charalambakis and Garrett, 2018; Jiang and Jones, 2018). Campbell et al. (2008) combined equity market data and accounting data as explanatory variables in their study on corporate distress risk. Chava and Jarrow (2004) argued for the inclusion of variables beyond accounting data. Compared to accounting-based bankruptcy prediction models, they obtained superior results when including market variables related to publicly traded equity. Several other papers have examined the power of market variables in financial distress prediction or similar studies, where valuation multiples (Chen et al., 2006; Jiang and Jones, 2018) and market capitalization (Jones et al., 2017; Ugurlu and Aksoy, 2006) have been applied.

Information asymmetry, as previously mentioned, motivates the use of input features that reflect the market's opinion about a company. Beta, a variable indicating the risk of a security relative to some index, is typically used as a proxy to assess such a relation. Certain studies which investigate the use of beta in distress prediction find that stocks of distressed companies have highly variable returns and high betas, which implies that such stocks are more sensitive to overall market conditions (Campbell et al., 2011; Campbell et al., 2008). How industry-specific factors affect a company's risk of financial distress has received relatively little attention in the literature. Agrawal and Maheshwari (2019) proposed an industry beta for Indian companies, calculated by regressing the monthly stock return on their respective industry sectors, and found that a high sensitivity to industry factors (high betas) led to increased probabilities of default. As the targeted companies in this report are listed at various Nordic stock exchanges, we examined betas linked to both industry and stock exchange returns.

2.4.3 Feature Selection Using XAI

Whereas certain models for prediction and classification are inherently interpretable, many ML models need to be explained using devoted analysis tools. These types of tools and frameworks, such as the aforementioned SHAP, can be helpful in both model explainability and feature importance analysis. As described by Fryer et al. (2021), the core issue of feature selection in ML is to select a subset of feature indices from a set of features so that the model will both minimize cost and maximize some evaluation function linked to a specific goal. Shapley values, implicitly utilized in the SHAP framework, pose an alternative to general feature selection methods. In its simplest form, feature selection using Shapley is performed by computing Shapley values for all features and selecting the highest ranked features. Similar algorithms for feature selection with Shapley values have been applied in the literature, see (Guha et al., 2021; Jothi et al., 2021; Marcilio and Eler, 2020).

Starting with a wide feature space, where the selection of input variables is chiefly based on their frequency of occurrence in literature and financial theory, we aim to develop a parsimonious model with high predictive power by eliminating less relevant inputs through feature selection using SHAP. Further elaboration on chosen features is provided in Section 3.

2.5 Related Research and Contribution to Literature

Our literature review has revealed that most prediction studies have a quite narrow scope in terms of geographical affiliation of data. This observation is corroborated by Kumar and Ravi (2007), who summarized the source of data/country of origin for 71 studies with data between 1969 and 2002. It is clear that most studies target company data from specific countries, where the USA and South Korea appear particularly frequent. Despite the degree of interconnection in the global economy, distress prediction models aiming to generalize across multiple countries have rarely been addressed in the literature. Efforts have been made to generalize across borders in the Nordics in areas of research related to that of our own, where Sormunen et al. (2013) investigated cross-border audit behaviour by comparing reporting patterns for confirmed bankruptcy cases in Norway, Sweden, Denmark and Finland. Although the study found significant differences in financial reporting prior to bankruptcy, the authors emphasize the benefits of such a geographical scope due to similarity in legal systems and audit standards. As pointed out by Tian and Yu (2017), the literature covering international markets is relatively sparse, and the scarcity of studies aimed at developing a generalized distress framework for Nordic countries has motivated our work.

Studies comparing different prediction models rarely address the composition and process of obtaining and augmenting datasets in detail. In most cases, only a brief assessment of the training/test data split, and distribution of the target variable is given, while little attention is paid to data augmentation, data cleaning, and pre-processing. It appears that predictions are somewhat uncritically conducted on readily processed datasets, with rigid definitions of distress. Another factor that is rarely addressed in the literature is data granularity. Although there exist papers using quarterly data, see Cheng et al. (2019) and Huang and Yen (2019), the majority of existing research on the topic is performed on annual data, where single snapshots of each feature reflect a full year of financial activity. The synergies from combining thorough data processing with improved data dynamics from more frequent observation intervals make our paper a distinctive contribution to the literature and are likely to have improved prediction results.

Although the elements described above arguably represent novelty in terms of a unique *combination* of geographical scope, assessment of data, definition of distress, and model comparison, examples from some of the elements can be found in related literature individually. For instance, Tian and Yu (2017) present a study on bankruptcy prediction across international markets, including Japan, the UK, Germany, and France, using a discrete hazard model. Several other studies have also compared ML-based models in bankruptcy, default, or distress prediction using a traditional statistic model as a baseline. However, these seldom go into detail regarding data processing or tuning of model parameters. See for example Iturriaga and Sanz (2015), Moscatelli et al. (2020) and Yeh and Lien (2009). We argue that, by combining a series of analytical elements, we provide novelty to the existing body of research.

3 Dataset

As discussed previously, most related research either utilizes readily processed datasets or omits in-depth descriptions regarding data collection and processing. There are clear advantages of having an end-to-end framework for dataset generation for industrial and academic replicability. Therefore, this section provides an extensive explanation of the process of going from a publicly available professional financial database to a fully trainable dataset. Apart from the initial choice of variables, which is motivated in Section 2.4, the steps include data collection, data cleaning, additional feature generation, feature selection using the SHAP framework and generation of feature-target classification samples. In addition, the issues of missing values, outliers and data imbalance are illuminated and addressed with proposed mitigating steps.

3.1 Data Collection

The dataset collected for this project is comprised of quarterly reported financial key figures from Nordic listed companies (excl. Iceland) for the time interval Q1 2001 to Q2 2022, complemented by macro and market data. The data was chiefly gathered from the Bloomberg database (Bloomberg, 2022). Bloomberg was chosen due to its high quality of data, access to macro and market data, and availability of quarterly financial company records. A data query was made of companies that had non-missing values of Current Ratio (CR) and Interest Coverage Ratio (ICR) during at least 5% of the time interval, since these variables constitute the proxy described in Section 3.3. Table 1 shows the distribution of companies and classification samples with respect to country of listing. It is evident that reporting quality varies somewhat across countries and that a significant portion of the dataset stems from Swedish-listed firms. Table 2 shows the variables that were included in the initial data query to Bloomberg. The motivation behind the choice of these variables is provided in Section 2.

Country	Number of listed companies	Number of companies after data cleaning	Number of samples	Number of distress samples
Sweden	1092	379 (34.7%)	5383	598 (11.1%)
Norway	589	130 (22.1%)	2745	364 (13.3%)
Denmark	357	67 (18.8%)	1572	131 (8.3%)
Finland	279	63 (22.6%)	1344	112 (8.3%)
Total	**2317**	**639 (27.6%)**	**11044**	**1205 (10.9%)**

Table 1: Number of companies per country before and after data cleaning. The final columns display the number of classification samples produced from each country in the final dataset.

Accounting Data	Pre-calculated Ratios	Macro/Market Data	Other Data
EBIT, Working Capital, Fixed Assets, Total Assets, Current Assets, Operating Income, Capital Expenditure, Current Liabilities, Total Sales	EPS, Profit Margin, ROE, ROA, Total Debt to Total Capital, Operating Income to Interest Paid, Total Debt to Total Assets, Current Ratio	Daily Stock Price, P/E, P/B, P/S, Market Capitalization, MSCI Index Development, GDP Growth, 6 Month and 10 Year Bid Yield on Government Bonds*, Stock Exchange Indices*	Company Name, Country Name, Date, GICS Industry Code

Table 2: Financial variables included in initial data query from Bloomberg. *Supplemented with data from financial database Refinitiv Eikon (2022).

A set of features were generated from the raw data pulled from Bloomberg. Table 3 shows the list of initial features and their corresponding category and formula or explanation, if applicable. Some features were unavailable or not in the exact format we needed and thus had to be calculated as part of the data pre-processing. Stock data was used in conjunction with Global Industry

Variable Name	**Category**	**Formula/Explanation**
Current Ratio	Liquidity Ratio	Current Assets / Current Liabilities
WCTA	Liquidity Ratio	Working Capital / Total Assets
Asset Turnover Ratio	Efficiency Ratio	Operating Revenue / Total Assets
Fixed Asset Turnover	Efficiency Ratio	Operating Revenue / Fixed Assets
Fixed BEP Ratio	Efficiency Ratio	EBIT / Fixed Assets
BEP Ratio	Profitability Ratio	EBIT / Total Assets
Profit Margin	Profitability Ratio	EBIT / Operating Revenue
ROA	Profitability Ratio	Net Income / Total Assets
ROE	Profitability Ratio	Net Income / Shareholder Funds
Debt Asset Ratio	Solvency Ratio	Total Debt / Total Assets
Debt to Capital Ratio	Solvency Ratio	Current Liabilities / Capital
Interest Coverage Ratio	Solvency Ratio	EBIT / Interest Paid
Working Capital	Raw Value	N/A
EBIT	Raw Value	N/A
Total Sales	Raw Value	N/A
Total Assets	Raw Value	N/A
Capital Expenditure	Raw Value	N/A
Fixed Assets	Raw Value	N/A
Current Assets	Raw Value	N/A
Current Liabilities	Raw Value	N/A
Government Bond Spread	Macro Variable	10Y Bond Yield - 6M Bond Yield
GDP Growth	Macro Variable	Quarterly GDP Growth
Industry Beta	Macro Variable	See Equation 3.1
Industry Index Return	Macro Variable	Quarterly Log Return on MSCI-Index
Stock Volatility	Market Variable	Annualized Quarterly Stock Volatility
P/B	Market Variable	Price per Share / Book Value per Share
P/E	Market Variable	Price per Share / Earnings per Share
P/S	Market Variable	Price per Share / Sales per Share
Market Capitalization	Market Variable	Share Price · Shares Outstanding
Market Index Return	Market Variable	Log Return on Stock Exchange Index
Market Beta	Market Variable	See Equation 3.1
Stock Return	Market Variable	Log Return of Stock
EPS	Market Variable	Income Available to Common Stockholders / Weighted-Average Number of Common Shares Outstanding*
GICS Code	Categorical Variable	Affiliated Industry
Quarter	Categorical Variable	Seasonal Indicator
Country	Categorical Variable	Affiliated Country

Table 3: Explanatory variables selected for this paper. *Firms with dilutive securities issued must include the effects of these securities in calculation of EPS.

Classification Standard (GICS) codes and corresponding Morgan Stanley Capital International (MSCI) Europe indices in order to calculate stock quarterly log returns, industry quarterly log returns and industry-stock beta for each company. GICS codes are hierarchically structured, and the 11 industry sectors at the top level were deemed sufficient to categorize the different companies. There is little discrepancy in the literature regarding the omission of financial institutions from corporate distress prediction due to their inherently unique structure. Thus, *Financials* (GICS prefix 40), i.e. banks and insurance companies, were excluded from the dataset. Table 4 shows the top-level GICS codes used and their corresponding MSCI index and sector name. In line with common practice, linear interpolation was used to fill gaps in stock data, but extrapolation was avoided to minimize the risk of creating overly synthetic samples. Industry beta values were estimated in a historical trailing twelve months manner using Equation 3.1 (with sample variance and covariance) on log returns from common trading days of MSCI indices and stocks, labelled R_i and R_s, respectively. Conventional market beta values indicating movement of a company's stock price relative to its respective stock exchange index were created similarly. Market betas were assessed relative to the OMXS30 Index, the OSEBX Index, the OMXC20 Index and the OMXH25 Index for Swedish, Norwegian, Danish and Finnish companies, respectively. These are market value-weighted indices representing the main stock exchanges of Stockholm, Oslo, Copenhagen, and Helsinki, respectively.

GICS Prefix	Sector Name	MSCI Index	No. Companies	No. Samples
10	Energy	MXEUEN	46	1202
15	Industrials	MXEUMT	38	739
20	Materials	MXEUIN	153	2755
25	Consumer Discretionary	MXEUCD	57	1130
30	Consumer Staples	MXEUCS	29	651
35	Health Care	MXEUHC	112	1534
40	Financials	MXEUFN	N/A	N/A
45	Information Technology	MXEUIT	100	1467
50	Communication Services	MXEUTC	39	541
55	Utilities	MXEUUT	7	174
60	Real Estate	MXEURE	58	851

Table 4: GICS industry sectors and their corresponding MSCI Europe indices. Number of companies and classification samples are numbers after data cleaning, i.e. from the final dataset.

$$\beta_i^s = \frac{Cov(R_s, R_i)}{Var(R_i)} \tag{1}$$

As shown in Table 3, six-month and ten-year bid yields on government bonds were obtained to calculate each country's yield spread throughout the reference interval. The yield spread is a metric used by investors to assess the relative attractiveness of bond investments. Whether the spread between long-term and short-term government bonds is widening or tightening can reflect changes in the country's underlying economy. Along with GDP growth, industry beta and market beta, this feature aims to capture the macro and market environment surrounding each firm.

For raw values, findings of Zhang and Shi (2018) indicate that the inclusion of variables related to firm size in default prediction models may increase predictive power. Alkhazali and Zoubi (2005) argued that popular proxies for firm size show instability across time and industry and might not be interchangeable. In other words, while Total Assets could serve as an appropriate proxy of firm size in some asset-heavy industries, the same may not be the case for labour-intensive industries. To counteract these instabilities and ensure that information was captured in full, the raw values listed in Table 3 were included in parallel.

As highlighted in Section 2.1, there is evidence in the literature that valuable information may reside in the quarters leading up to the prediction quarter as well. This paper uses a static prediction model approach. To include parts of a company's history in the prediction, we calculated inter-quarter changes of all company-specific numerical features that were not already variables of change (e.g. log returns). Adding quarterly change variables nearly doubled the number of features, which prompted further analysis for selecting a final feature set.

The final type of variables in Table 2 is comprised of the categorical variables GICS Code, Quarter, and Country. As opposed to the other input variables to the model, the categorical variables are not intended to be interpreted directly as numeric values but rather as discrete nominal groupings. By effect of its tree-based structure, the LightGBM is inherently able to handle categorical values. The ANN and LR, on the other hand, require the categorical variables to be transformed into numerical values in order to interpret them. For this task, we used one-hot encoding[1]. In order to avoid the dummy variable trap[2] associated with the one-hot encoded variables, a single binary entry was dropped per categorical variable.

3.2 Feature Selection

It is of utmost importance to mitigate noise when preparing input data for ML models. A primary source of noise is found in non-informative features, which, in addition to complicating the prediction task, also burdens the architect with increased demands of data quality and capacity. Although tree-based methods such as LightGBM are relatively robust to non-informative features, the size of our final dataset was not. In other words, each new accounting variable negatively affected the size of our final dataset since the Bloomberg database contained companies with varying data reporting quality. By excluding non-informative variables from the feature set, we lowered the requirement for reporting quality and thus, fewer samples were dropped during data cleaning. In addition, this step enabled us to choose a parsimonious model, decreasing the risk of overfitting. In this project, a supervised elimination approach was used for feature selection. By examining SHAP values of features that were classified by a default-tuned computationally efficient LightGBM using the TreeSHAP algorithm, we conducted a one-step backward elimination to select our final set of features. A subtle but important detail to address is the issue of data leakage since the feature selection process has the potential to extract information from parts of the future test set and apply it before testing. To avoid this, only 50% of the data was used for feature selection and subsequently excluded from the test set when this was constructed. The reason for not performing our 80/20 train test split at this point was that we expected to see our dataset increase in total size after removing non-informative features, thereby rendering the test set too small.

The beeswarm plot, sometimes referred to as summary plot, in Figure 1 shows the relative importance of the top 34 features in the predictions made on a validation set from the feature selection dataset. A similar plot of the bottom 30 features can be found in Appendix F. Features at the top indicate high explanatory power, while those at the bottom are considered less informative. For further explanation of the SHAP framework and beeswarm plots, see Section 4.2 and 5.3.1, respectively. It is worth noting that some variables are expected to capture more or less the same information and could potentially be interchanged in the event of missing data for future projects. An example of this is shown in Appendix E, where we observe that industry and market betas are highly correlated. As outlined in Section 4.2, although TreeSHAP assumes some degree of independence between variables, it does so to a smaller extent than other SHAP applications, which is why it was deemed appropriate for handling the task of feature selection. Table 5 shows the final set of features used in this paper.

[1] The conversion of categorical variables into binary indicators.

[2] Introducing perfect multicollinearity by redundantly specifying a dummy variable for each category.

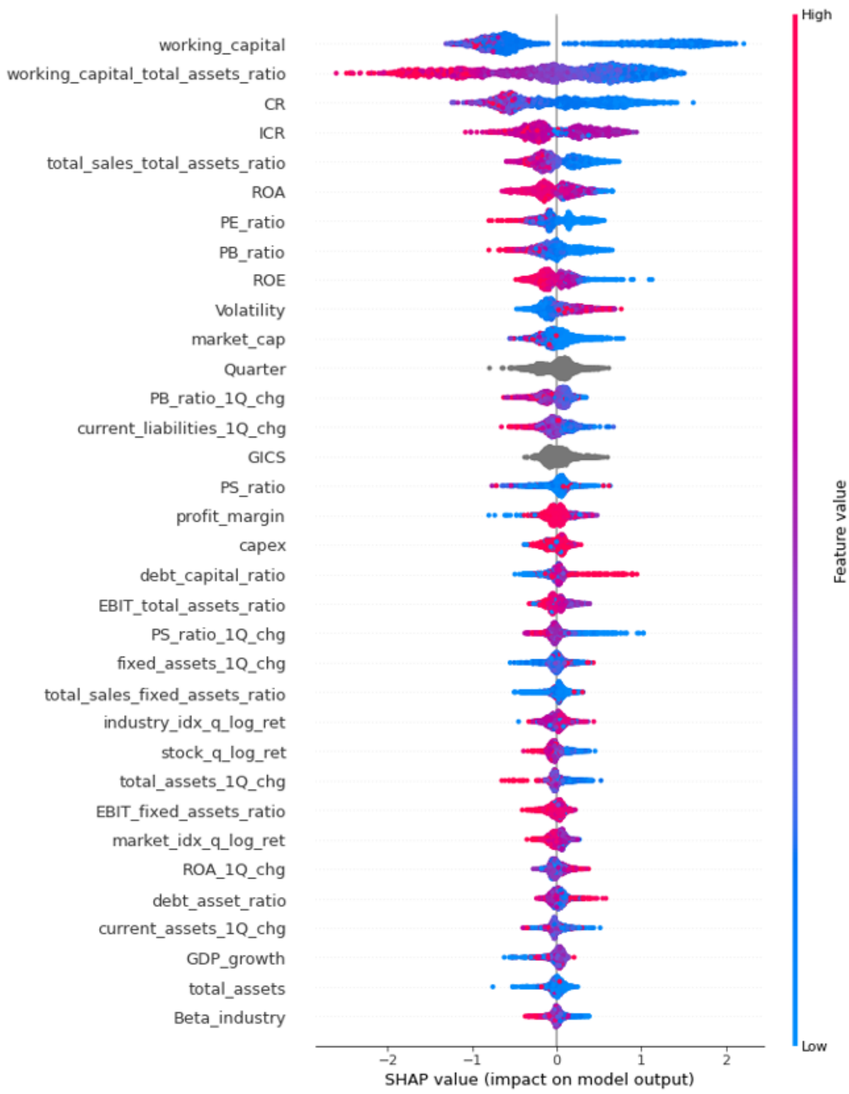


Figure 1: SHAP beeswarm plot showing the 34 most important features. Clusters of SHAP values, represented as dots, left of zero, indicate a feature contribution toward non-distress. Similarly, clusters right of zero indicate feature contribution toward distress. For instance, a blue cluster (low values) of volatility left of zero indicates that having low stock volatility will contribute toward receiving a non-distress prediction.

3.3 Definition of Financial Distress

As discussed in Section 1, using juridical definitions of failure, i.e. documented credit events, to represent failure of a company in prediction studies can be problematic. Firstly, the information surrounding company default is inherently noisy, and the occurrence of certain events, such as bankruptcy as a strategic move, may entirely lack explainability in financial data. Balcaen and Ooghe (2006) bring forth several other reasons why credit events serve as a poor foundation for dichotomous classification. Among these are the fact that it may take several years before failure is formally recorded, making the actual point of distress challenging to determine, and the possibility of other juridical exits such as merger, absorption, dissolution, and liquidation, which act to conceal distress. The issue of time lag between actual failure and recorded failure would be amplified in the context of this study, given the quarterly granularity of our prediction window. Secondly, the relatively rare occurrence of strictly defined negative credit events causes severe class imbalance and omits "grey area" companies, which introduces instability in prediction models. Instead, as a continuation of Malakauskas and Lakstutiene (2021) and similar studies' descriptions of financial distress, we propose a proxy-based definition which allows for expanding the minority class and creating an early warning model to recognize companies in early stages of distress. By doing so, we draw a line between general states of financial distress and the more rigid definition of company failure based on credit events such as bankruptcy and focus on the former. The latter definition, which is not studied in this paper, can be considered a stricter subcategory of our definition of financial distress. For our proxy, we use the ICR in combination with CR.

The ICR is a solvency measure that describes a company's ability to generate sufficient earnings to cover its interest payments, while the CR is a liquidity measure describing the ability to service its short-term debt or, correspondingly, to withstand short term fluctuations in earnings. Assessing a company's ICR relative to some target value is widespread in the credit rating industry. Credit rating agencies such as S&P, Moody's, Fitch Ratings, Morningstar DBRS and Nordic Credit Rating employ ICR (alternatively EBITDA or EBITA to Interest expense) and CR when assessing the financial risk profile of a company. Publicly available credit rating reports and methodology guides from these agencies suggest that an ICR value below 1.5 would typically coincide with non-investment grade entities, and loan and bond covenants often dictate this as a lower threshold (see Fitch Ratings, 2021; Moody's, 2022; Morningstar DBRS, 2022; Nordic Credit Rating, 2022a,b; S&P, 2013). Typically, a financial covenant between a creditor and a debtor dictates that the debtor must maintain a certain level of financial robustness or credit worthiness. A breach of covenant, which, for instance, occurs when certain financial ratios of a debtor fall below a given threshold, will have serious implications for both parties. A creditor will be at risk of not being repaid in full, and a debtor may face higher interest rates, penalty fees, or eventually a technical default. In addition to the ICR, the CR is often included in loan and bond covenants, dictating a minimum level of liquidity. The metric is also supported in the financial literature. Mohammed and Kim-Soon (2012) and Awais et al. (2015) use Altman's Z-score in parallel with a CR value of 1.1 to predict company failure. Both studies find that CR is a useful tool in predicting failure, and the latter even concludes there is no significant difference in using CR compared to the Z-score. Kozlovskyi et al. (2019) show that there is a general agreement among experts that a CR lower than 1 typically indicates a high risk of bankruptcy for a company.

We propose the following thresholds for distress to be used during class labelling of the data: A company that does not generate sufficient earnings to meet bond covenants (ICR < 1.5, i.e. Interest Paid exceeding $\frac{2}{3}$ EBIT), and simultaneously suffers from a low degree of liquidity (CR < 1, i.e. Current Liabilities exceeding Current Assets), is deemed to be in a financially distressed situation. In sum, we believe that our distress proxy with the assigned numerical thresholds finds support in both industry practices and the financial literature.

3.4 Data Cleaning and Missing Value Handling

As can be seen from Table 1, many companies were entirely removed due to incomplete financial records. With the final feature set of 34 features, only about 1/4 of the companies survived data cleaning. If a company was dropped from the dataset, one or more variables from the feature set were missing on that company in the Bloomberg database for the entire duration of the time interval. Other examples of missing data were partial samples, i.e. companies with one or more missing values for a particular quarter. Partial samples that had at most two missing feature values were attempted filled with values from the previous quarter. In other words, samples with more than two missing values were not attempted filled but dropped to avoid creating overly synthetic samples. Figure 2 shows the total number of samples available in the dataset and full samples before and after handling missing values. The total number of samples grows because the number of listed companies has grown over the time interval. A partial sample is counted if a company has reported at least one financial variable signifying its existence at that point in time. Full samples are samples with no missing values, i.e. the samples that end up in the final dataset. In other words, the black line illustrates the maximum amount of usable samples that could have been generated had all companies reported on all financial variables. The gap between the blue and the red line represents the effect of filling in missing values as described above. We note that the financial crisis around 2008 seems to have caused changes in reporting practices since the number of full samples more than doubles over a single quarter, while the total number of samples stays more or less the same. Another note is made toward the end of the time interval, where the number of full samples plummets as companies have yet to publish all of their financial information.

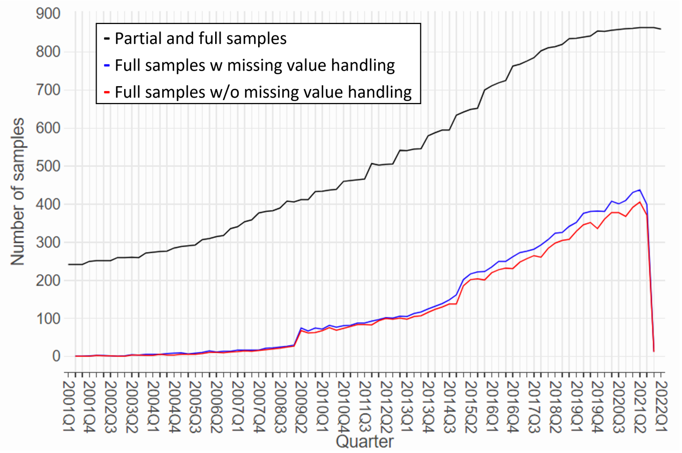


Figure 2: Number of partial and complete samples per quarter before and after handling missing values. The gap between the black and red line illustrates the effects of incomplete reporting.

3.5 Sample and Target Generation

Figure 3 displays the process for generating classification samples from the dataset. A rolling window methodology was used, which entails a window that moves across the time dimension generating feature-target pairs. The window converts financial input from quarters leading up to the prediction point into features which represent the sample and examines the CR and ICR of the ensuing Target quarter to determine the target of the sample, i.e. future state of the company, labelling distressed companies 1 and non-distressed companies 0. This process essentially removes the time dimension from the dataset, yielding a 2D array of samples × features and a 1D target array, which is needed to train an ML classification model.

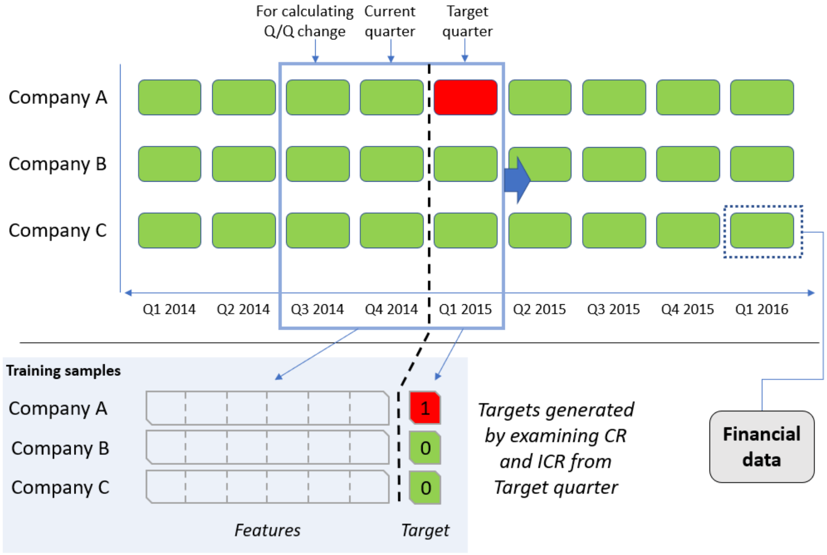


Figure 3: Process for generating training samples: A rolling window uses two consecutive quarters of financial data from a company to generate both static and inter-quarter change features, which constitute the input to the model. Targets are generated by using proxy thresholds and examining CR and ICR from financial data in the ensuing "Target" quarter.

When removing the time dimension, we risk introducing bias by ignoring time-dependent effects. The issue is somewhat mitigated with features that are in the form of ratios. However, some features, such as Market Cap (which is a popular proxy for company size), come in the form of raw values. An issue with this is that one would expect to see larger values toward the end of the interval simply due to inflation and the growth of economic markets. Thus, to provide comparable values across time, raw values were adjusted with their respective country's inflation rates provided by the European Central Bank Statistical Data Warehouse (ECB, 2022). Another source of bias could be seasonal variations or structural breaks caused by, for instance, political or technological shifts that lead to persisting changes in factors such as the capital structures of companies. This type of bias is neither avoided by having features as ratios nor by adjusting for inflation. Seasonality was accounted for by including the quarter as a categorical variable. A way to explicitly account for structural breaks is to add a manually defined indicator variable which highlights significant points in the time interval, such as the financial crisis around 2008. This option puts added responsibility on the authors to correctly identify and include significant breakpoints and potentially introduces a new source of noise to the model. Therefore, an indicator variable was not included, but we argue that the information is sufficiently captured and implicitly represented through the included macro variables. Time-dependent effects could potentially be handled altogether by performing in-quarter transformations. For instance, min-max scaling each feature for each quarter would rank companies with respect to other co-existing companies, as opposed to implicitly ranking them across time. The caveat to such an approach is that some information is lost when scaling with different distributions; the largest company (in terms of, for instance, Market Cap) during a recession suddenly becomes the same as the largest company during a boom. In addition, as shown in Figure 2, several quarters in the first half of the interval contain very few samples, making the transformation highly susceptible to noise and outliers. Finally, since a transformation must also be applied to future samples before feeding data into the model, a single transformer fitted on one of the quarters must be chosen to represent the appropriate distribution with respect to which any new sample should be scaled. As a consequence, we would need to separate entire individual quarters for testing to avoid any data leakage via the transformations. On account of these caveats, in-quarter transformations were avoided in favour of the more straightforward approach of inflation adjustment of raw values and inclusion of macro variables.

3.5.1 Data Analysis and Outlier Handling

Since many of the features are ratios and the denominator in some cases can get close to zero, large values are to be expected. We have not considered these to be outliers of noise since they convey information but have taken steps in terms of model choice and transformations to handle their presence. Figure 4 displays an example of a feature with several extreme values compared to one with no outliers. We also note from the box plot that there is considerable statistical separability in the latter feature, consistent with preliminary SHAP feature importance plots.

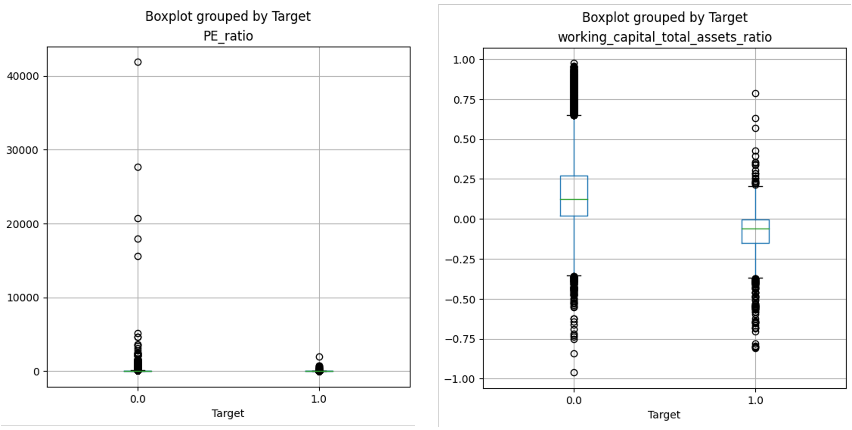


Figure 4: Boxplots of P/E ratio grouped by target, and WCTA ratio grouped by target. P/E ratio shows clear presence of extreme values, which distort the plot but are easily explained by earnings being close to zero. An additional observation is that there is a notable separation between target groups in the WCTA ratio box plot.

Table 5 contains descriptive statistics for the chosen features in the final dataset, and Appendix G shows a heatmap matrix of their correlations. The highest correlations are appearing, as expected, between size-dependent variables such as Market Cap, Total Assets, Working Capital and CapEx, where coefficients range between 0.48 and 0.84 in absolute values, and between naturally related variables such as ROE and ROA, Debt to Capital and Debt to Total Assets, and change in Current Liabilities, Current Assets, and Total Assets, where coefficients are 0.73, 0.85 and around 0.99, respectively. On account of the high correlation between the latter three features, the two inter-quarter change features current_liabilities_1Q_chg and current_assets_1Q_chg were dropped from the final feature set in favor of using total_assets_1Q_chg to capture the information, leaving 32 instead of 34 features in total. We note that quarterly change in the P/B ratio has a relatively large negative correlation with EBIT to Total Assets. This is to be expected since EBIT directly impacts a company's book value. An example is apt to illustrate this: a negative EBIT will lead to a negative EBIT to total assets ratio, which decreases the company's book value, thereby causing a positive change in P/B, given a negligibly correlated share price.

Evident from the minimum and maximum values in Table 5, most ratios have examples where the denominator has likely become very small compared to the numerator. As inter-quarter change variables are dependent on reference level, those too show examples where the level (denominator) has been close to zero. As explained in Section 4, tree-based models such as LightGBM are inherently robust to extreme values in the input and seldom require any transformation to correctly train on the data. For the ANN and LR benchmark model, a transformation which handles outliers was chosen based on the evidence of this analysis. Scikit learn's[3] Quantile Transformer transforms data by ranking and assigning quantiles to the data points. The Quantile transformer offers a normal output distribution and is robust to outliers. The transformer also handles the issue of large differences in the scale of input features since its outputs are in a fixed interval.

Feature	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
Beta_industry	0.53	0.41	-2.88	0.26	0.5	0.77	2.57
CR	2.14	5.63	0.01	0.98	1.4	2.11	461.95
EBIT_fixed_assets_ratio	-2.18	76.73	-4781.08	-0.02	0.04	0.17	3169.67
EBIT_total_assets_ratio	-0.02	1.95	-203.72	-0.01	0.01	0.03	21.98
GDP_growth	0.89	2.49	-9.3	0.1	0.8	1.9	8.9
ICR	-275.27	13405.7	-1261175.0	-1.46	3.34	11.68	88912.0
PB_ratio	11.2	221.95	0.0	1.11	2.6	6.31	15659.98
PB_ratio_1Q_chg	0.41	16.77	-1.0	-0.05	-0.01	0.05	1348.27
PE_ratio	52.73	583.28	0.0	0.0	13.37	36.8	41883.26
PS_ratio	249.3	4982.21	0.02	0.88	2.31	7.83	199054.15
PS_ratio_1Q_chg	0.04	2.1	-1.0	-0.05	-0.01	0.01	177.71
ROA	-2.64	23.18	-271.88	-4.24	3.12	7.26	221.92
ROA_1Q_chg	0.26	22.35	-477.2	-0.16	0.0	0.18	1661.41
ROE	-4.69	59.95	-1653.96	-10.3	7.57	17.88	1059.74
Volatility	0.38	0.26	0.05	0.22	0.3	0.45	4.67
current_assets_1Q_chg	0.2	10.6	-1.0	-0.09	-0.0	0.09	1107.62
current_liabilities_1Q_chg	0.21	9.7	-1.0	-0.08	0.01	0.13	1013.68
debt_asset_ratio	25.99	19.25	0.0	10.08	24.04	38.76	116.23
debt_capital_ratio	34.96	23.43	0.0	15.74	35.59	51.53	214.29
fixed_assets_1Q_chg	0.67	25.53	-1.0	-0.04	-0.0	0.04	1908.94
industry_idx_q_log_ret	0.02	0.1	-0.61	-0.01	0.04	0.08	0.29
market_idx_q_log_ret	0.03	0.09	-0.38	-0.01	0.03	0.08	0.25
profit_margin	-1747.83	36474.2	-2076500.0	-7.72	3.41	10.53	650400.0
stock_q_log_ret	0.01	0.26	-2.66	-0.1	0.02	0.14	1.93
total_assets_1Q_chg	0.15	10.11	-1.0	-0.03	0.0	0.04	1061.27
total_sales_fixed_assets_ratio	20.31	407.45	-0.07	0.24	1.09	3.57	23625.0
total_sales_total_assets_ratio	0.24	3.0	-0.0	0.07	0.19	0.29	304.08
WCTA	0.13	0.23	-0.96	-0.01	0.1	0.24	0.98
capex	-1695777	7675879	-445616793	-555831	-68453	-2892	0
market_cap	172517645	615970230	25086	2885847	16626234	76901804	10033746355
total_assets	124749445	406931326	14030	2050788	14036938	65435701	5149789660
working_capital	11213372	72113719	-320357815	-16567	508127	3947463	1117249711

Table 5: Descriptive statistics for the 32 numerical features included in the final dataset. Ratio and inter-quarter change features are grouped at the top, while inflation-adjusted reported numbers are shown at the bottom. Categorical variables, i.e. Quarter and GICS, were omitted from this statistic.

[3] See Pedregosa et al. (2011)

3.6 Addressing Data Imbalance

Evident from Table 1, the minority class constituted a mere 10.9% of samples in the final dataset. As discussed in Section 4, the LightGBM can be set to handle imbalance with hyperparameters, and LR does not require balanced class distributions in order to train optimally. In the case of the ANN, measures were taken to balance the training sets in order to mitigate the adverse effects that imbalance can have on model performance.

To exploit the entire dataset, over-sampling was chosen to combat class imbalance since it achieves class balance by increasing the size of the minority class instead of decreasing the size of the majority class. Contrary to the somewhat naïve approach of re-sampling, we used a version of the Synthetic Minority Over-sampling Technique (SMOTE), called Borderline SMOTE. SMOTE generates synthetic samples similar to those in the minority class. The algorithm selects a random point among the k-nearest neighbours of an existing minority class datapoint and creates a new datapoint by calculating a randomly weighted linear combination of the two points. Borderline SMOTE is based on the same principle but favours minority class datapoints in proximity to majority class datapoints, generating more samples close to the border between the two classes. The implementation of Borderline SMOTE was imported from the imblearn package[4] and employed with its default value of $k = 5$ nearest neighbours during sampling. The main advantage of the synthetic approaches is that they create new, unseen examples. On the other hand, the approach of simply re-sampling existing datapoints has the disadvantage of creating a smaller decision region which may lead to overfitting during learning (Han et al., 2005). A crucial detail to emphasize here is that the over-sampling was only performed on the train set, so the reasons for avoiding creating synthetic samples in previous data cleaning steps do not apply here. This means that the results presented in Section 5 are from a stratified split-out test set with realistic samples and class distributions (as opposed to synthetic and balanced).

3.7 Final Dataset

Table 6 shows the final dataset after conducting an 80/20 stratified train/test split. It is once again highlighted that the samples involved in the feature selection process were kept out of the final test set to avoid data leakage. As described in the previous section, training data was balanced with Borderline SMOTE as a final step before being fed to the ANN, leading to a synthetically increased training set size of 15 744 samples in that particular case. The transformer, which was applied for both the ANN and LR model, was fitted to unbalanced training data in order to provide a representative distribution for scaling new input samples.

Each sample in the final dataset consisted of 32 features, selected in Section 3.2, and an associated target label, constructed in accordance with the procedure described in Section 3.5. The features include company-specific, industry-specific, and general market and macro information, which strengthens the quality of this paper. Company-specific information and market information are both backward-looking, e.g. accounting variables representing recent history and change variables representing more distant history, and forward-looking, e.g. valuation multiples representing the expected future performance of a company. The environment surrounding the company is captured by both industry-specific, seasonality and macro information. In sum, we argue that the features span the wide spectrum of relevant company-related information that can be thought to influence the future financial state of the company.

	Non-distressed (Label 0)	**Distressed (Label 1)**	**Total**
Train	7871	964	**8835**
Test	1968	241	**2209**
Total	**9839**	**1205**	**11044**

Table 6: Distribution of minority and majority classes in final train and test set.

[4] See Lemaître et al. (2017)

As a final remark, we confirmed that the final dataset was not linearly separable with respect to the target. Formally, two sets $H = \{H^1, \cdots, H^h\} \subseteq \mathbb{R}^d$ and $M = \{M^1, \cdots, M^m\} \subseteq \mathbb{R}^d$ are said to be linearly separable if $\exists a \in \mathbb{R}^n, b \in \mathbb{R} : H \subseteq \{x \in \mathbb{R}^n : a^T x > b\}$ and $M \subseteq \{x \in \mathbb{R}^n : a^T x \leq b\}$ (See Póczos (2013) for further explanation). We applied linear programming through the use of Scipy (Virtanen et al., 2020) and confirmed that the optimal value of the objective function was $\neq 0$, testifying to the linear non-separability of the dataset.

4 Models and Methodologies

This section provides a high-level explanation of how the models deal with important properties of the dataset. Further, the relevant hyperparameters to be tuned for each model are described. Finally, we present the performance metrics that were deemed appropriate for assessing the models given the nature of our prediction task. As discussed in Section 2, the literature indicates that ML approaches tend to outperform classical models in financial classification tasks, and prominent ML models within the field of financial distress prediction include the LightGBM and ANN. To provide a benchmark representative of classical models, the trusted LR model was included due to its frequent appearance and proven track record in the literature. Thus, the models for comparison are LightGBM, ANN and LR. If supplementary intuition behind any of the models is needed, the reader is referred to Appendix A through C.

4.1 Models and Hyperparameters

A hyperparameter is a model parameter that can be adjusted to affect different parts of the learning process. The most important tuneable attributes for each model are outlined below. Due to the extensive number and ranges of the hyperparameters in both the ANN and the LightGBM, the hyperparameter search space within which we seek to find our optimal model configurations quickly becomes intractable. Therefore, the tuning application programming interface (API) Weights and Biases (WandB)[5] was used to tune hyperparameters. WandB provides a dashboard that allows us to track metrics of interest as various model configurations are trained and validated, which gives further insight into the tuning process. As will become evident, certain hyperparameters of the LightGBM have to be tuned in parallel. Consequently, a grid search, where all possible configurations are tested, resulted in 3456 different models. For the ANN, WandB's Bayesian Search option was employed to traverse its slightly more continuous hyperparameter search space intelligently. This entails that configurations were changed based on an informed search through the hyperparameter search space based on results from previous configurations (Biewald, 2020). Table 7 and Table 8 list the hyperparameters that define the Bayesian and grid search space for the ANN and LightGBM, respectively.

4.1.1 Artificial Neural Network

Given the characteristics of our data, we believe that the ANN's ability to model highly complex and non-linear relationships provides an advantage in the prediction task at hand. In addition, ANNs do not make any assumptions about the input distributions. There is no explicit outlier handling available through tuning the hyperparameters of the ANN, but, as noted in Section 3, outliers are, in this case, handled by the Quantile Transformer. ANNs are known to be data-hungry and prone to overfitting, but steps related to data augmentation (i.e. oversampling) and regularization were taken to accommodate this. The implementation of the ANN was conducted using Keras (Chollet, 2015), which is a Python library built on top of TensorFlow[6]. Keras was chosen due to its simplicity when implementing complicated networks. Additionally, it is extensively documented, so that intuition behind each hyperparameter is easily available. When building a neural network, the architect has to make several choices to achieve the best possible performance. The choices include addressing the elements summarized in Table 7, which are explained separately in the following.

Number of Hidden Layers
The architect essentially defines how complex the decision boundary is allowed to be when determining the number of hidden layers. A model with no hidden layers, i.e. just an input and output layer, is only capable of linear separation. As shown in 3.7, our data is not linearly separable, so at least one hidden layer is required. Hornik (1991) showed that given a nonlinear activation function, a single hidden layer can approximate any arbitrary, continuous decision boundary. Consequently,

[5] See Biewald (2020)
[6] See Abadi et al. (2015)

more than one hidden layer is necessary when an arbitrary noncontinuous decision boundary is required. As a result, two and three hidden layers were tested. More than three layers were not explored to limit training time and reduce the risk of overfitting by adopting an overly complex model.

Number of Perceptrons in Each Hidden Layer

The optimal number of perceptrons in each layer is determined by a trade-off between capturing complex connections and overfitting to the training data. According to Xu and Chen (2008), having too few perceptrons leads to high training and generalization error due to underfitting. Too many perceptrons, on the other hand, results in lower training error because more complexity is captured but higher generalization error due to overfitting. Furthermore, more training examples are required to fit a larger network, which may result in a disproportionate learning time. Despite the existence of rules of thumb and research aimed at determining the optimal number of perceptrons, the most common approach is still trial and error (Xu and Chen, 2008). As a result, various network architectures were investigated. The number of perceptrons in each hidden layer was chosen randomly from a uniform distribution ranging from 64 to 512 for the first layer and 32 to 256 for the second layer. If present, the third layer was made up of either 32 or 16 perceptrons to limit the search space.

Activation Function

The activation function of a perceptron determines how the weighted sum of the inputs is transformed into an output of that perceptron. On a higher level, the activation function in the hidden layers determines how well the network learns based on the task at hand, while the activation function in the output layer determines what kind of predictions the network can make. This paper deals with binary classification. As a result, a Sigmoid function was chosen for the output layer. The Sigmoid-function shown in Equation 10 in Appendix B transforms any input to a value between 0 and 1, which can thus be interpreted as a probability when there is only one perceptron in the output layer. Although the traditional Sigmoid function has been used as the internal activation function for many years, Parhi and Nowak (2020) argue that the Rectified Linear Unit (ReLU) activation function is preferable. The primary motivation for creating the ReLU was its ability to promote sparsity by reducing the number of active neurons. Furthermore, because the activation function is computationally cheap, training networks with ReLU tends to be faster. However, ReLU suffers from the dying ReLU-problem, which means that it will always discard negative values by setting them to zero. As a result, the derivative is set to zero, and there is no weight update during backpropagation (Biewald, 2020). Clevert et al. (2015) proposed the Exponential Linear Unit (ELU) to address this issue. As shown in Figure 5, ELU allows for negative output, bringing the mean unit activation closer to zero. This is similar to batch normalization, but the computational complexity is lower (Clevert et al., 2015). Regardless, both activation functions were tested.

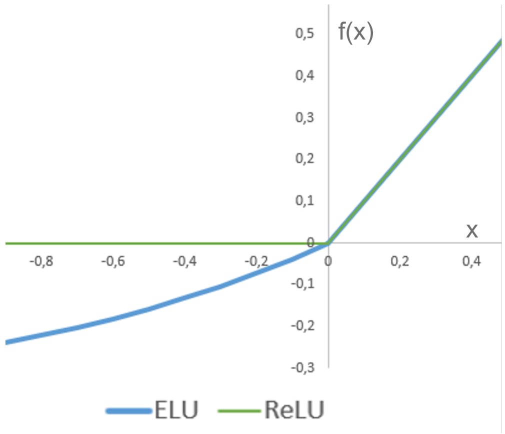


Figure 5: Visualization of ELU and ReLU activation functions. ReLU is linear for positive inputs, but maps all negative inputs to zero. ELU, on the other hand, behaves logarithmically below zero and linearly above zero, bringing the mean unit activation closer to zero compared to ReLU.

Optimizer
The task of the optimizer is to change the network's weights and biases so that the loss function is minimized. According to Choi et al. (2019), no theory adequately explains how this choice should be performed. The update rule, which is further specified by hyperparameters such as learning rate, defines an optimizer. Adaptive Moment Estimation (Adam) has recently been proposed as the preferred optimizer for ANNs (Desai, 2020). However, Desai demonstrated that Stochastic Gradient Descent (SGD) with momentum outperforms Adam and generalizes better under certain conditions. Consequently, we tested both Adam and SGD with momentum, where two common values for momentum were tested, namely 0.9 and 0.98.

Learning Rate
The learning rate is arguably the most critical hyperparameter when training a neural network. Setting the learning rate too low causes slower training and raises the probability of convergence to a local optimum, whereas setting the learning rate too high may lead the network to diverge. There are strategies to prevent both, such as momentum to avoid local optima and the `ReduceLROnPlateau` hyperparameter from Keras, which reduces the learning rate when the network no longer improves. However, these approaches still burden the architect with the task of determining the proper initialization for the network's learning rate (Chollet, 2015). Smith (2017) proposed a cyclical learning rate to accommodate this. Smith clarified that while a higher learning rate may have a short-term negative influence on learning, it can have favourable long-term impacts. To take advantage of this, one can alter the learning rate in cycles between a minimum and a maximum range. As a result, cyclical learning rates for SGD with momentum were implemented, evaluated, and compared to non-cyclical learning rates applied with Adam.

Regularization
Regularization involves strategies to choose a parsimonious model and keep the model from overfitting. While some regularization techniques are included in the optimizers themselves, Srivastava et al. (2014) developed the concept of a dropout layer in the network. Dropout combats overfitting by randomly removing perceptrons, along with their associated weight and bias, during training. Dropout is applied to a neural network by iteratively picking a random subset of the perceptrons and training only this portion of the network. As a result, the network's perceptrons are prevented from co-adapting excessively, decreasing the impact of noise in the dataset. The authors of the original work employed a dropout rate of 0.5 in the hidden dropout layer. Since we believe overfitting to be a significant vulnerability of the ANN, mostly higher dropout rates were tested. Thus, the dropout for hyperparameter tuning was selected from a uniform distribution between 0.4 and 0.8.

Hyperparameter	**Values**
Number of Hidden Layers	(2, 3)
Perceptrons in Layer 1	$U_{64,512}$
Perceptrons in Layer 2	$U_{32,256}$
Perceptrons in Layer 3	(0, 16, 32)
Activation Function	(ReLU, ELU)
Optimizer	(Adam, SGD w./Momentum)
Dropout Rate	$U_{0.4,0.8}$
Learning Rate Adam	$U_{0.001,0.1}$
Learning Rate SGD	Cyclical (0.001, 0.1)
Momentum	SGD (0.9, 0.98)

Table 7: Hyperparameters and the corresponding values that were tested during ANN tuning. Lists define discrete values, while U signifies that values were chosen from a uniform distribution.

4.1.2 Light Gradient Boosting Machine

The LightGBM framework was chosen due to its popularity in literature, its ability to handle unbalanced data and categorical features, and its performance in terms of training speed and accuracy. Due to its tree-based structure, there is no need to scale or transform input data, and the LightGBM does not rely on assumptions about the input distribution to function optimally. Since they do not employ distance metrics, DTs, in general, are insensitive to high variance and outliers in the form of extensively large or small values. As described in Section 3, several features in our dataset exhibit extreme minimum and maximum values, substantiating the choice of this model. What distinguishes, and makes LightGBM exceptionally fast compared to similar gradient boosting algorithms, can be explained through three properties: Gradient-Based One-Sided Sampling (GOSS), Exclusive Feature Bundling (EBF), and histogram-based splitting (binning). These characteristics will be explained in greater detail in the following sections and in Appendix C. Similar to the ANN, LightGBM's effectiveness highly depends on the choice of hyperparameters. There are more than 100 tuneable parameters in LightGBM, so we chose to tune those we believe to be the most influential to model performance. The selected hyperparameters are summarized in Table 8, and further explained below.

Gradient Boosting Methods
LightGBM is inherently a boosting framework, and there are three types of gradient boosting methods that can be varied by specifying the `boosting`-parameter: Gradient Boosting Decision Trees (GBDT), Dropout meet Multiple Additive Regression Trees (DART), and Gradient-Based One-Side Sampling (GOSS)[7].

In the original paper from Friedman (2001), GBDT was the proposed method and is still the default boosting type in LightGBM. In contrast to the pronounced advantages of LightGBM, this method discards the one-sided sampling, which is intended to reduce training time. However, GBDT is known to be both stable and reliable but may overspecialize and use a considerable amount of memory.

DART is essentially an improvement to the traditional Multiple Additive Regression Trees (MART) model. One of the most significant problems for MART, i.e., traditional Gradient Boosted Trees, is over-specialization. The issue arises when trees added in the later iterations impact only a few instances while making vanishing contributions to the rest. This will, in turn, make the model over-sensitive to the initially added trees instead of generalizing smoothly across the ensemble. To counteract this effect, Vinayak and Gilad-Bachrach (2015) proposed an addition of dropout to the MART model, resulting in the DART model. This essentially means dropping trees randomly and can be viewed as a regularization technique.

In conjunction with EBF, the GOSS boosting type is what makes LightGBM stand out compared to similar methods and gives rise to the "Light" part of the framework (Ke et al., 2017). The default boosting parameter, GBDT, is reliable and accurate but does not scale to larger datasets. In the original paper introducing LightGBM, the authors describe a sampling method that involves sorting the instances by gradient and prioritizing the ones with a high gradient to be used for training. To avoid altering the data distribution, a random sample of instances is chosen from the remaining instances (Ke et al., 2017). By utilizing this technique, the training time is reduced while the performance persists. In our case, this allows for training and testing a broad range of model configurations during hyperparameter tuning in a reasonable computational frame.

All three values for the `boosting_type` hyperparameter were tested.

Learning Rate
The `learning_rate` in a gradient boosting classifier determines the magnitude of the modification that each new tree presents to the ensemble. As a rule of thumb, the value defaults to 0.1 and should be lowered for large datasets and increased for small datasets. Evident from Section 3, the size of our dataset is moderate, and therefore values around the default were considered.

[7] to replicate the model proposed by Ke et al. (2017), the `boosting_type` must be set to GOSS.

Regularization
The LightGBM framework builds its weak learners leaf-wise instead of level-wise, and it is established that leaf-wise tree-building algorithms tend to converge faster at the expense of being more prone to overfitting (Zhang and Gong, 2020). Therefore, several regularization techniques have to be applied to counteract this tendency. In addition to testing DART, the regularization hyperparameters `reg_l2`, `early_stopping_rounds`, `num_iterations`, `num_leaves`, and `max_depth` were included in tuning.

There are two main penalty terms used for regularization, L1 and L2, serving slightly different purposes. In general, the L1 penalty term encourages that some features be entirely discarded, which is a desired property when performing feature selection. However, for this project, the feature selection, as described in Section 3.2, was conducted by using the SHAP framework. Therefore, L1 was ignored in favour of experimenting with values for L2. The L2 penalty term, denoted `reg_l2` above, intends to punish less-predictive features without removing them entirely, a property which suits the purpose of our analysis well since we seek to explain financial, macro and market factors influencing company distress.

The hyperparameter `num_iterations` specifies the number of decision trees to build. Training accuracy increases with the number of trees, but so does the chance of overfitting. To counteract this effect, `num_iterations` should be tuned in parallel with `early_stopping_rounds`. Early stopping helps reduce overfitting by monitoring the model's performance on a separate validation set and ends the training once the validation metric has not improved over a specified number of rounds. It is worth mentioning that when `num_iterations` increases, the learning rate should decrease since it essentially defines the contribution of each new tree to the ensemble.

Finally, two closely related hyperparameters that need to be tuned in parallel are the `num_leaves` and `max_depth`. The number of leaves and the maximum depth of each tree are considered to be important hyperparameters for restraining the complexity of the model. A deep tree with a large number of leaf nodes will be able to capture complex relationships in the data at the expense of an increased risk of overfitting. Considering that a leaf-wise built tree with the same number of leaves as a level-wise built tree on average will become deeper sooner in the process, one has to tune these parameters in parallel.

Hyperparameter	**Values**
Gradient Boosting Method	GBDT, DART, GOSS
L2 Regularization	(0, 0.1, 0.3)
Early Stopping Rounds	(25, 50)
Number of Iterations	(50, 100, 200)
Number of Leaves	(8, 16, 31, 50)
Maximum Depth	(-1, 25, 50, 75)
Learning Rate	(0.01, 0.1, 0.05, 0.2)

Table 8: Hyperparameters and the corresponding values that were tested during LightGBM tuning.

4.1.3 Logistic Regression

Unlike ANN and LightGBM, an LR model has very few tuneable hyperparameters. Although the architect can select different regularization methods, early testing revealed no significant improvement when these were changed from their default values. The `C` hyperparameter, which controls the penalty strength, was investigated for potential improvements. `C` indicates how much the model should trust that the training data is representative of the entire population. A low value for `C` instructs the model not to pay attention to outliers, whereas a high value for `C` reassures the model that all data points can be trusted. In most cases, however, the `C` hyperparameter should be close to 1 in order to create a generalized model. As a result, the values 1, 25, and 50 were tested.

4.2 SHAP Framework

Many ML models are not inherently interpretable and need to be explained using an explanation model, which is an interpretable approximation of the model itself. The well renowned XAI framework of Lundberg and Lee (2017), SHAP, which builds on the Shapley values of Shapley (1953), uses a linear function of binary variables as an explanation model. This makes SHAP an additive feature attribution method, and such an explanation model can be expressed as:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2}$$

Where g represents the explanation model, $z' \in \{0,1\}^M$ is the vector of simplified input features, M is the number of simplified input features, and $\phi_i \in \mathbb{R}$ is the attribution effect of some feature i. A Shapley value can be interpreted as the marginal contribution of each feature averaged over the set of all feature combinations. It can thus be used to explain the contribution of individual features for a specific instance. Lundberg and Lee (2017) showed that there exists a unique solution in the class of additive feature attribution methods that, as for Shapley values, holds the three desirable properties 1) local accuracy, 2) missingness, and 3) consistency. This solution can be expressed as follows:

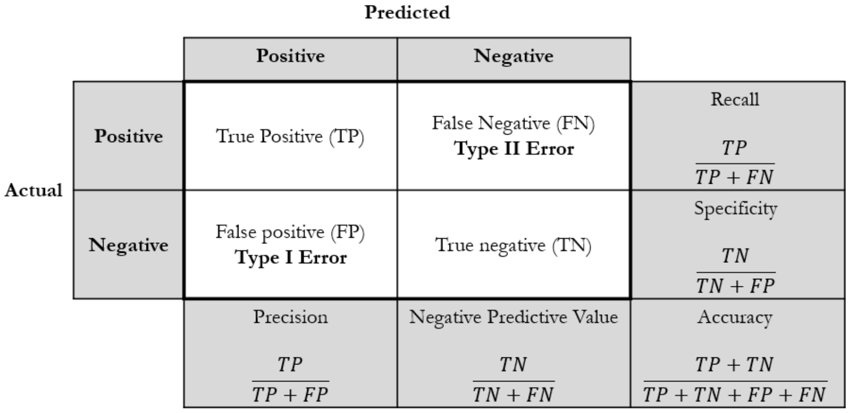
$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z' \setminus i)] \tag{3}$$

Where f is the original model we wish to explain, x is the original input, $|z'|$ is the number of non zero elements in the vector *z'*, $z' \subseteq x'$ denotes all vectors of *z'* where the non-zero elements are a subset of the non-zero elements in *x'*. For further details, the reader is referred to the original paper of Lundberg and Lee. The solution to Equation 3 is referred to as SHAP values. SHAP values are "the Shapley values of a conditional expectation function of the original model" (Lundberg and Lee, 2017, p. 4), meaning that for each feature, they indicate the change in expected prediction when conditioning on that specific feature. Although classified as a local model, as described in Section 2.3, SHAP can be used for both local and global interpretability. Hence, it is able to provide insight both into to how feature values from the entire dataset contribute to individual predictions *and* into how the positive or negative contribution of the features on a single prediction score may be measured. Local explanations are commonly illustrated with waterfall plots, which show feature contributions for individual predictions, and global explanations for full model interpretation can be shown using beeswarm plots. Examples of both can be seen in Section 5.3.

Compared to earlier XAI frameworks, the SHAP framework is advantageous in terms of computation time and consistency between local and global interpretations. However, it is based on an assumption of independence between features, which is rarely the case for features such as our own. Several alternative applications have been suggested to improve and adjust SHAP computation according to the specific models applied. For this study, the TreeSHAP application of Lundberg et al. (2018) was used to explain the results of the superior model, LightGBM. An advantage with TreeSHAP is that it assumes less feature independence than similar methods, meaning it accounts for some but not all dependence (Aas et al., 2021).

4.3 Measuring Model Performance

To compare the performance of different models, it is necessary to apply metrics that evaluate the different techniques on the same premise. After classification, predictions will belong to one of four categories, illustrated by the confusion matrix in Figure 6. Several metrics can be constructed from the four prediction categories. Depending on the structure of the dataset, some metrics are more appropriate than others in assessing a model's predictive ability. For instance, despite being a common metric for classifier evaluation, accuracy could be a deceiving metric when the dataset is unbalanced. This can be illustrated by using our dataset as an example: Given a minority class of 10.9%, an accuracy of $\sim 89.1\%$ is attainable by a model that simply predicts non-distress regardless of input. Despite scoring high on the accuracy metric, such a model would clearly be of no use.



		Predicted		
		Positive	**Negative**	
Actual	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	Recall $\frac{TP}{TP + FN}$
	Negative	False positive (FP) **Type I Error**	True negative (TN)	Specificity $\frac{TN}{TN + FP}$
		Precision $\frac{TP}{TP + FP}$	Negative Predictive Value $\frac{TN}{TN + FN}$	Accuracy $\frac{TP + TN}{TP + TN + FP + FN}$

Figure 6: Confusion matrix showing the four different categories for a prediction. Metrics are described along the bottom and right-hand edge. Illustration adapted from Chakravorty (n.d.).

The example given above motivates our choice of more nuanced performance measures, considering that not all errors are the same. Evident from the confusion matrix, there are two types of errors, type I and type II. In our case, type I errors refer to non-distressed companies classified as distressed, and type II errors are distressed companies classified as non-distressed. The existence of two types of errors naturally poses a trade-off between precision and recall. Exemplified, a model with high recall typically lacks precision and will correctly classify a large share of the actual positives at the cost of including many actual negatives among its positive classifications. The trade-off is best visualized using a precision-recall curve (See Section 5.2 for an example of this). The curve is made by varying the classification threshold between 0 to 1 while plotting the precision and recall achieved at each threshold. Even though such a curve is highly informative, it complicates the process of comparing models since there lacks a straightforward numerical metric. To circumvent this issue, the curve can be reduced to the F1 score, given by Equation 4, which is the harmonic mean of precision and recall. Since models are implicitly trained with a default threshold of 0.5, the F1 score essentially represents a single, realistic point on the precision-recall curve.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

Another popular way of assessing a classifier's performance is the Receiver Operating Characteristic (ROC) curve. The curve is constructed by varying the classification threshold and plotting the true positive rate (recall) against the false positive rate ($= 1 - Specificity$). Put simply, it shows a classifier's ability to sort the data based on their resemblance to the positive class. Optimally, the curve should "hug" the top left corner, indicating that the classifier found all positive examples without erroneously including a negative. Following the equivalent argumentation as given above, the ROC-Area Under Curve (ROC-AUC) score was used to reduce the curve down to a numerical metric for automated comparison during hyperparameter tuning. The ROC-AUC score essentially calculates how close the model's performance is to the optimal curve. As a point of reference: a completely random classifier will receive a score close to 0.5, and a perfect classifier will get a score of 1. Even though the ROC-AUC score works best for balanced datasets, it serves as a comparable summary of a model's performance when evaluated in conjunction with the F1 score.

5 Results

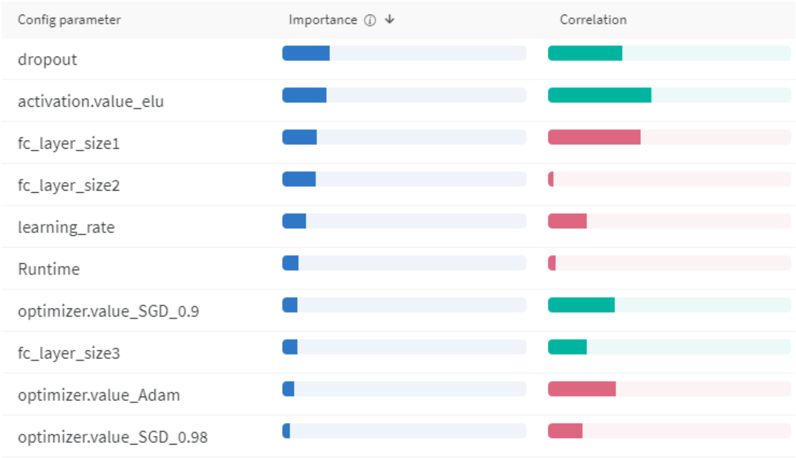
This section is divided into three main parts. The first addresses results from hyperparameter tuning described in Section 4.1, which constitutes the basis for the final choice of parameter configurations. The second part presents and discusses results from running the final models on the test set and compares them based on performance metrics described in Section 4.3. Finally, local and global results of the superior model are interpreted through the SHAP framework.

5.1 Results from Hyperparameter Tuning

5.1.1 Tuning of ANN

Figure 7 shows the importance and correlation of hyperparameters with respect to the validation ROC-AUC score achieved by the different ANN configurations. Importance represents each hyperparameter's contribution to the metric, while correlation relates their values. The statistics are based on the top 1000 different model configurations. Since the search through the hyperparameter space was done using a Bayesian approach, parameter values that yielded higher ROC-AUC scores were favoured by receiving a higher likelihood of showing up in the following configurations. Appendix H displays an illustrative overview of the top 500 model configurations, along with the validation ROC-AUC score achieved after training. The exact parameter values were chosen based on the analysis below and an assessment of the five top-performing models to avoid choosing unstable model configurations that performed well on the validation data by pure chance.

Evident from Figure 7, the `dropout` hyperparameter has the highest importance, and the correlation coefficient being positive (indicated with green colour) suggests that choosing a value in the upper range would be beneficial. Regarding the number of fully connected hidden layers, it is clear that three layers perform the best for this task and that the number of neurons in this layer should equal 32, the largest categorical alternative. The number of neurons in the preceding layers has relatively high importance as well and should be set in the lower and middle parts of their respective ranges for optimal performance. The optimizer that performs best is clearly SGD with momentum equal to 0.9. This optimizer was implemented with a cyclical learning rate and was kept for final testing. Finally, ELU was chosen as the internal activation function since it was favoured by the Bayesian search illustrated in Appendix H.



Config parameter	Importance ⓘ ↓	Correlation
dropout	🟦	🟩
activation.value_elu	🟦	🟩
fc_layer_size1	🟦	🟥
fc_layer_size2	🟦	🟥
learning_rate	🟦	🟥
Runtime	🟦	🟥
optimizer.value_SGD_0.9	🟦	🟩
fc_layer_size3	🟦	🟩
optimizer.value_Adam	🟦	🟥
optimizer.value_SGD_0.98	🟦	🟥

Figure 7: ANN hyperparameter importance with respect to ROC-AUC score on the validation set from WandB. Importance represents the contribution of each hyperparameter to the metric, while correlation relates their values. Green color indicates positive correlation to the ROC-AUC score, while red color indicates negative correlation.

Table 9 shows the hyperparameter configuration chosen for the ANN to be evaluated on the final test set. The batch size of 100 was maintained for final training, but the number of epochs was increased from 100 to 200 to ensure that the final model could find an optimal point of lowest validation loss during training. To save the model at this point of lowest validation loss, we used the ModelCheckpoint callback function provided by Keras (Chollet, 2015).

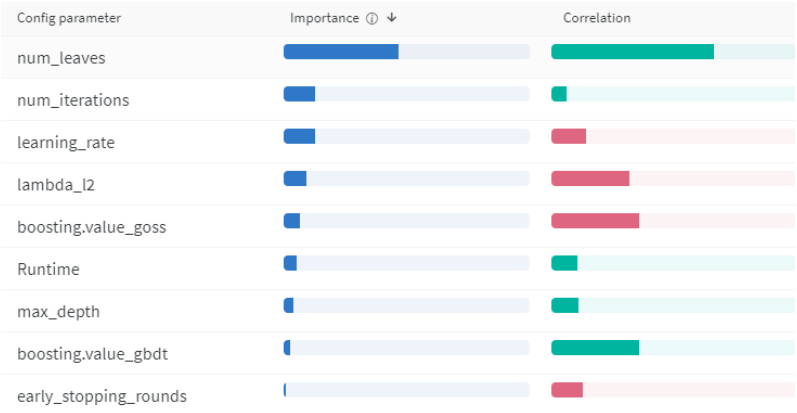
Hyperparameter	Values
Number of Hidden Layers	3
Number of Perceptrons in Each Hidden Layer	(92, 125, 32)
Activation Function	ELU
Optimizer	SGD
Dropout Rate	0.7
Learning Rate SGD	Cyclical (0.001, 0.1)
Momentum	SGD 0.9

Table 9: Final hyperparameter values for ANN.

5.1.2 Tuning of LightGBM

Hyperparameter importance and correlation shown in Figure 8 are based on the top 1000 LightGBM configurations from WandB. Contrary to the ANN, the search through the hyperparameter space was done in a grid-like manner, which entails that all 3456 configurations[8] of hyperparameter values were trained and tested on the validation set. Appendix I shows an overview of the top 500 model configurations, along with individual validation ROC-AUC scores achieved after training. The exact parameter values were chosen in a similar manner to the ANN, based on an analysis of hyperparameter importance and an assessment of the five top-performing models.

The most important hyperparameter for the LightGBM in terms of ROC-AUC score was the number of leaves, evident from Figure 8. The `num_leaves` correlation coefficient is positive, leading us to select the highest value among the top five models, which was the default value of 31. Since the number of iterations showed a slight positive correlation and top-performing models were all in the upper range, we were confident in selecting a high number of iterations for our final model. The learning rate and max depth were chosen as the median values of the top-performing models due to insignificant importance and correlations. The boosting type GOSS was chosen with a lower L2 rate of 0.1 despite a negative correlation due to the fact that all of the top-performing models utilized GOSS and regularization. Finally, choosing too high of a value for `early_stopping_rounds` seemed to be ineffective, so a value of 25 was selected for the final model. Table 10 displays the final hyperparameter configuration for the LightGBM.



Config parameter	Importance ⓘ ↓	Correlation
num_leaves		
num_iterations		
learning_rate		
lambda_l2		
boosting.value_goss		
Runtime		
max_depth		
boosting.value_gbdt		
early_stopping_rounds		

Figure 8: LightGBM hyperparameter importance with respect to ROC-AUC score on the validation set from WandB. Importance represents the contribution of each hyperparameter to the metric, while correlation relates their values. Green color indicates positive correlation to the ROC-AUC score, while red color indicates negative correlation.

[8] $3 \cdot 3 \cdot 2 \cdot 3 \cdot 4 \cdot 4 \cdot 4 \cdot 4 = 3456$ different hyperparameter configurations.

Hyperparameter	**Values**
Gradient Boosting Method	GOSS
L2 regularization	0.1
Early Stopping Rounds	25
Number of Iterations	200
Number of Leaves	31
Maximum Depth	50
Learning Rate	0.05

Table 10: Final hyperparameter values for LightGBM.

5.1.3 Tuning of LR

The LR model showed no improvements when the `C` hyperparameter was set to values larger than the default. Therefore, the default value of 1 from Scikit-learn's implementation was maintained in the final model.

5.2 Model Results on Test Set

Table 11 shows the precision, recall, F1, and ROC-AUC scores obtained by the final tuned models described above. The results show that the ML methods LightGBM and ANN are superior to the benchmark LR model in terms of F1 score, which is in line with findings in related research highlighted in Section 2. The LightGBM outperforms the ANN on both F1 and ROC-AUC scores. Their similar and superior performance compared to the benchmark model indicates that they are both capable of capturing complex relationships that the benchmark model could not recognize, substantiating our belief that there were, in fact, significant non-linear interactions at play between variables in the data.

Model	**Precision**	**Recall**	**F1**	**ROC-AUC**
LightGBM	0.52	0.78	**0.63**	**0.93**
ANN	0.46	0.82	0.59	0.92
LR	0.74	0.46	0.57	0.92

Table 11: Precision, recall, F1 and ROC-AUC scores for all three models on the test set. LightGBM is the superior model in terms of F1 and ROC-AUC score.

As outlined in Section 4.3, precision and recall, the components of the F1 score, are closely related to type I and type II errors, respectively. To reiterate, in the context of this task: a type I error represents the event that an otherwise healthy company is classified as distressed, while a type II error represents the event where a distressed firm is classified as healthy. From a creditor's perspective, one can argue that a type II error is more severe than a type I error since the loss associated with a distressed firm significantly outweighs the potential income from successful interest payments. Therefore, we argue that a realistic measurement of model goodness in the context of this task should put more emphasis on recall compared to precision. It is clear that the LR favours precision over recall, in contrast to the other two models. The LightGBM balances the precision and recall to a higher degree than the ANN, which yields a higher F1 score. However, in light of the severity of type II errors described above, one could argue that the ANN is superior due to a higher recall, indicating fewer "missed" distress cases. Nevertheless, a precision score below 0.5 means that more than half of the positively predicted test samples are erroneous, which could be a deal-breaker for practical applications. Another argument is that the trade-off could be rebalanced by changing the classification threshold and thus that the ANN could have produced a higher F1 score. However, in practice, the models should be used with their training threshold of 0.5 since altering it based on results from the test set would require a new unseen test set to produce valid performance approximations of the models. Thus, the precision, recall and corresponding F1 score from Table 11 represent the most realistic indications of the models' performances, while the Precision-Recall and ROC curves can be used to explain model differences and uncover potential areas of improvements for further research. In other words, even though the ANN shows slightly better performance in terms of minimizing type II errors, it does so at a significant cost of precision.

Figure 9 displays the ROC and Precision-Recall curves for the three models. It is evident from the Precision-Recall plot that a stable, high precision cannot be obtained by any of the models at a level above 0.8. Nevertheless, considering the severity of type II errors, our focus should be on the middle-right part of the curves, where recall is high and type II errors are minimized. Consistent with results from Table 11, the ANN proves to be prioritizing recall, clearly underperforming the others in the range where precision and recall are balanced but performing well in the area where recall is about 0.8 and precision is between 0.4 and 0.5. The LR model exhibits the opposite behaviour, favouring precision over recall. It performs similarly to the LightGBM in the middle-left range but is superseded for recall larger than 0.5. We confirm that the LightGBM exhibits behaviour compatible with our analysis of error types by favouring recall while still maintaining the highest level of precision among models in that area.

The ROC curve in Figure 9 displays the similarity of models when it comes to ranking predictions. All three models perform well, and we only see slight variations in the top left area. We observe that the most considerable discrepancy in behaviour, although small, is between the LR and the ANN, where the ANN has a somewhat greater tendency to generate false positives when the true positive rate is low. In contrast, the LR generates more false positives when the true positive rate is high. This observation adds some nuance to the otherwise equal performance in terms of ROC-AUC score.

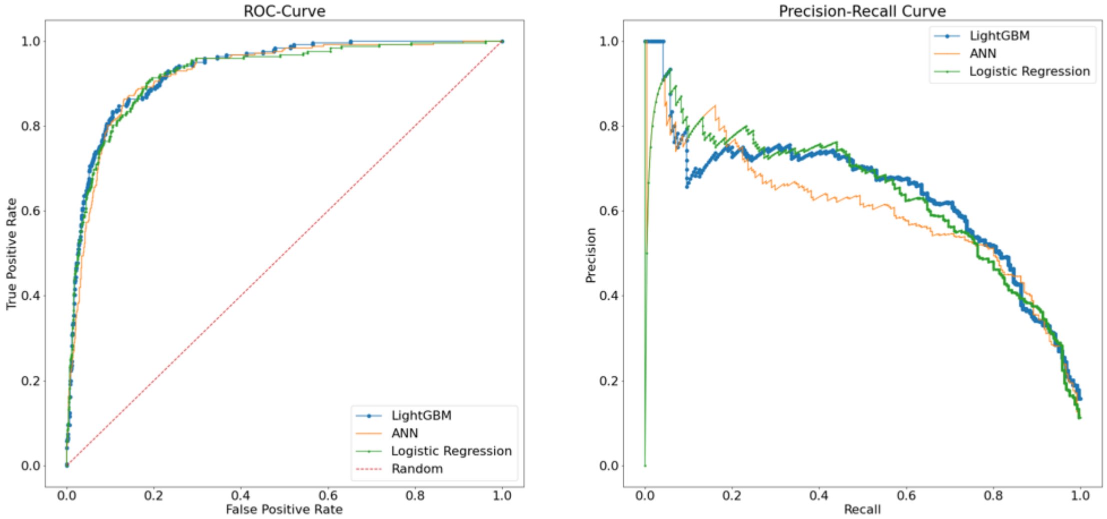


Figure 9: ROC and Precision-Recall curves for all three models on the test set. The red dashed line in the ROC plot represents a random classifier. Models perform more or less similar in terms of ROC but vary in terms of Precision-Recall. A precision score above 0.8 cannot be achieved with stability by any of the models. ANN performs poorly in the balanced Precision-Recall area but surpasses the LR model by a slight margin for recall values around 0.8. LightGBM shows clear superiority in the top right part of the graph, where recall is high (type II errors are minimized), and precision is maintained.

Based on the above results, we argue that the LightGBM proved to be the superior model for this task, and we consequently focus the following section on explaining its behaviour on the test set, using the SHAP framework presented in Section 4.2.

5.3 Result Interpretation with SHAP

This section applies TreeSHAP to the LightGBM model to interpret the behaviour that led to the results presented in the previous section. First, a global explanation is given by a SHAP beeswarm plot, examining the contribution of each feature to the model outputs. Next, interactions between variables are highlighted and explained using SHAP dependence plots. Finally, local explanations are provided by examining SHAP waterfall plots for two examples that were correctly classified by the model.

Although steps were taken to counteract multicollinearity, covered in Section 3.5.1, it is still evident from Figure 20 in Appendix G that several features were correlated. As discussed in Section 4.2, TreeSHAP assumes less feature independence than other XAI approaches and was therefore deemed suitable for explaining the LightGBM in light of the correlation uncovered among features.

5.3.1 Global Explanations with SHAP Beeswarm and Dependence Plots

Beeswarm plots display how the features impact the model's overall output. As explained in Section 3.2, the order of features from top to bottom on the y-axis represents their rank by importance. The x-axis displays the SHAP values generated by feature values of individual instances. Each row/feature holds the same number of dots, although this is not always easily observable due to regions of higher density (signified by vertical stacking). As can be seen from the gradient bar, the colours indicate low and high feature values. The categorical features are not ordinal, meaning their values cannot be ranked in order. Consequently, categorical features receive the colour grey. For further details regarding the beeswarm plot, the reader is referred to the SHAP documentation, see Lundberg (2018).

Evident from Figure 10, the most predictive features are closely related to liquidity, solvency and company size. It is not surprising that Working Capital, CR and ICR show up high on feature importance. This is because current information about the two ratios whose future values will determine the target must be assumed to be of high relevance. Intuitively, when attempting to say something about a state that involves a future value of a variable, an obvious thing to examine is the current value of that variable. However, the mixed colouring of their rows in the beeswarm plot clearly conveys that these features are not able to separate the data independently, and this is substantiated by the dependence plots shown in Figure 11. WCTA, ROA and Volatility, on the other hand, display more apparent signs of independent separation (blue and red dots are mostly on different sides of the middle line); however, some with a lower impact in terms of SHAP values. Notwithstanding, significant additional information is captured by the interaction effects of these latter variables as well, illustrated by, for instance, the ROA - WCTA dependence plot in the bottom right corner of Figure 11.

Several features in Figure 10 exhibit behaviours that coincide with our intuition. For instance, red dots (high values) along volatility generally lead to higher model output, which indicates distress. Furthermore, blue dots (low values) along Working Capital, WCTA, CR, and ICR should lead to a higher predicted probability of distress, which the plot confirms. We note that accounting for seasonality and industry sector was appropriate since both Quarter and GICS appear in the mid to upper range of the beeswarm plot. Dependence plots included in Appendix J further demonstrate the benefit of adding seasonality, macro and market information captured by Quarter, GICS, GDP growth and quarterly log returns of the stock and industry index, some of which provide clear advantageous interaction effects with more important variables.

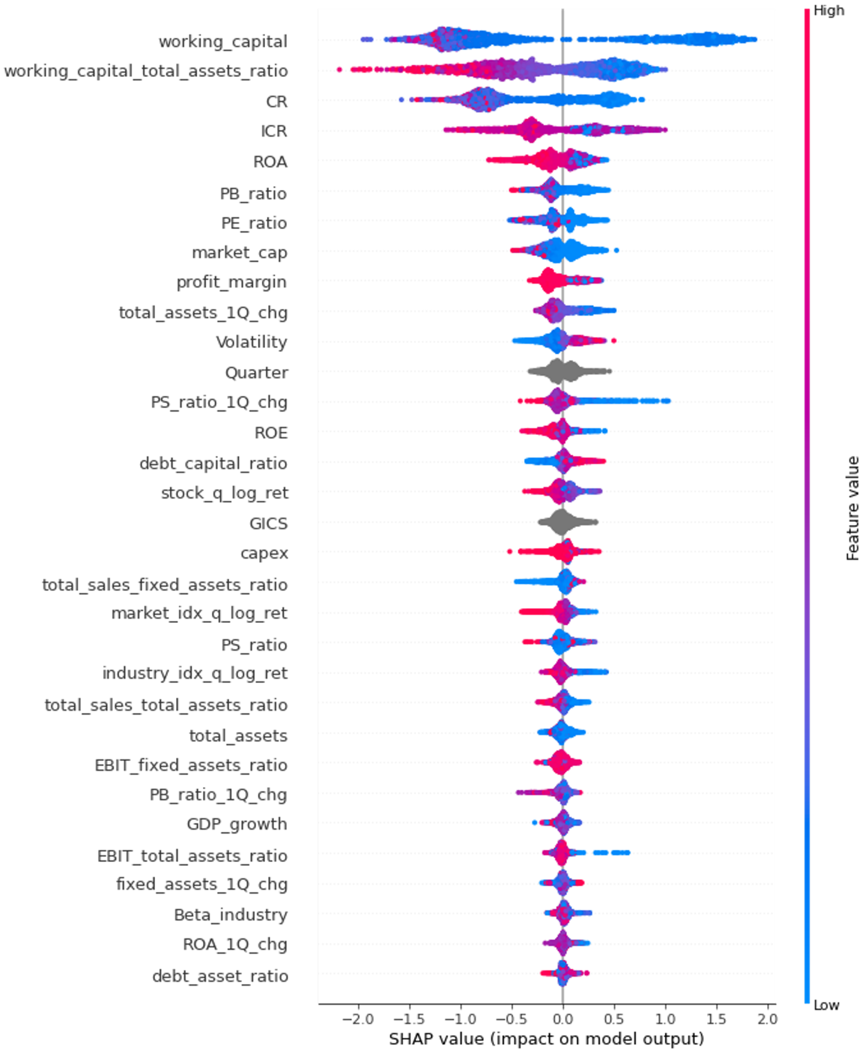


Figure 10: SHAP beeswarm plot of model predictions on the test set. Vertical ordering signifies the predictive power of features, each dot represents a single observed instance, and the colour of each dot indicates the feature value for that instance. Clear colour separation means that the feature independently separates the data - the degree to which it separates is indicated by horizontal spread. Mixed colouring indicates significant feature interaction, i.e. that the SHAP value produced by a single value of that feature varies significantly depending on other feature values.

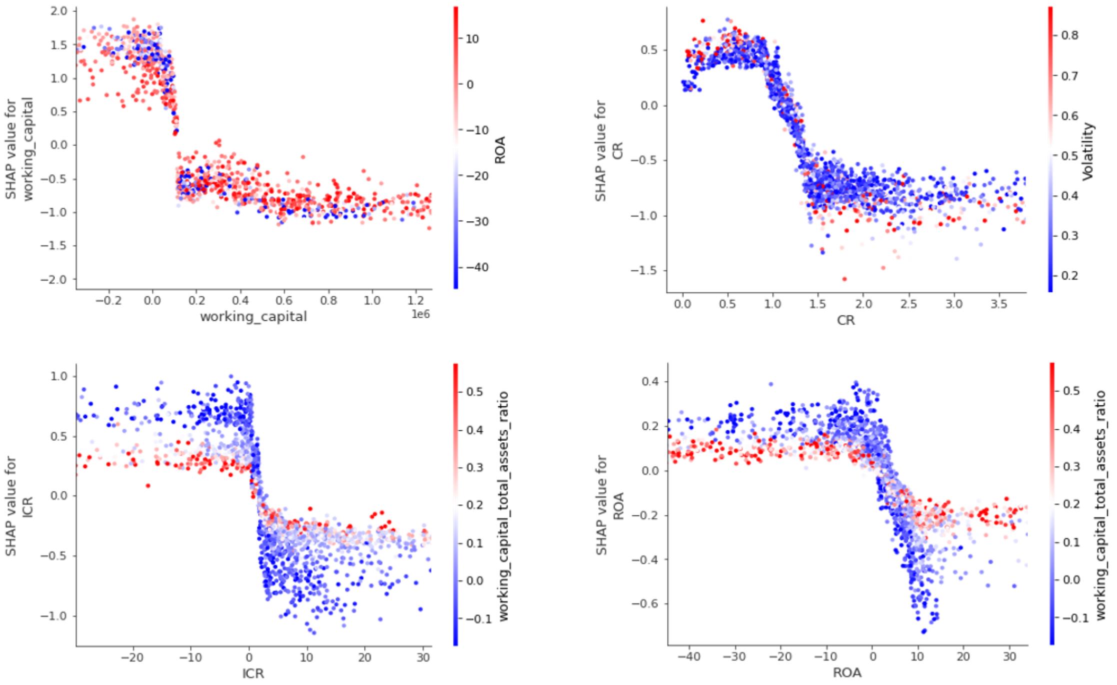


Figure 11: Dependence plots for Working Capital, CR, ICR and ROA. The interaction feature on the right-hand side of each plot is selected by the highest degree of interaction and is automatically chosen by TreeSHAP. The Working Capital plot in the upper left shows that companies with low ROA (blue dots) are given higher output than those with high ROA (red dots) conditional upon Working Capital being lower than 100 000 (left part of graph area). The CR plot shows a similar tendency, i.e. high volatility is more heavily punished (higher SHAP contribution toward distress) when CR is below 0.9. A clear switch is observable in the ICR and ROA plots, where the impact of a low versus a high WCTA is the opposite depending on whether or not ICR/ROA is above or below 0. If a company has a low WCTA (blue dot), it receives a higher SHAP value than if it had a high WCTA (red dot), conditional upon ICR/ROA being negative (left part of graph area). On the other hand, when the condition is changed to ICR/ROA being positive (right part of graph area), we observe that a low WCTA is actually less punished than a high WCTA in terms of SHAP value contribution.

5.3.2 Local Explanations with SHAP Waterfall Plots

Local explanations, meaning interpretability of predictions for individual instances, can be displayed by waterfall plots. Similar to beeswarm plots, feature importance and SHAP value (positive or negative) are indicated by top-to-bottom rank and bar colour, respectively. Starting from the expected model output which is learned during training, $E[f(x)]$ (denoted ϕ_0 in Equation 2), the waterfall plot illustrates how the most important features each exert influence leading to the model's final predicted value, $f(x)$ (denoted $g(z')$ in Equation 2), for an individual sample. By default, the units on the x-axis of the waterfall plot, $E[f(x)]$ and $f(x)$, are given in log odds, and the relation between predicted value, expected value, and SHAP values can be expressed as:

$$f(x) = E[f(x)] + \sum_{i=1}^{M} \phi_i x_i \tag{5}$$

Probability, which has a more intuitive interpretation than log odds, is given by:

$$Probability = \frac{e^{ln(odds)}}{1 + e^{ln(odds)}} \tag{6}$$

Note that Equation 5 expresses the same relation as Equation 2. The reader is referred to the documentation for further details regarding the waterfall plot (Lundberg, 2018). In order to explain model behaviour, we examine examples of predictions that were labelled true positives and true negatives, i.e. correctly classified as distressed and non-distressed, respectively.

Figure 12 shows a waterfall plot of a true positive example from the test set. By using Equation 6, the sample was predicted to be in a state of distress in the ensuing quarter with a probability of 57%. It is evident that CR and ICR offset one another for this specific instance, providing little combined explanation for the prediction. A P/B ratio of less than 1 indicates that the company's assets may be overstated in the market or that the company is struggling with its ROA. The plot displays a negative ROA and negative change in total assets substantiating the low P/B ratio and the impact on the prediction toward a state of distress. By examining the ROA dependence plot from Figure 11, we observe that a negative ROA in conjunction with a low value for WCTA will have a high, positive SHAP contribution to the model output. This explains how the model determines that a company which is currently not in distress could be expected to be in distress in the ensuing quarter.

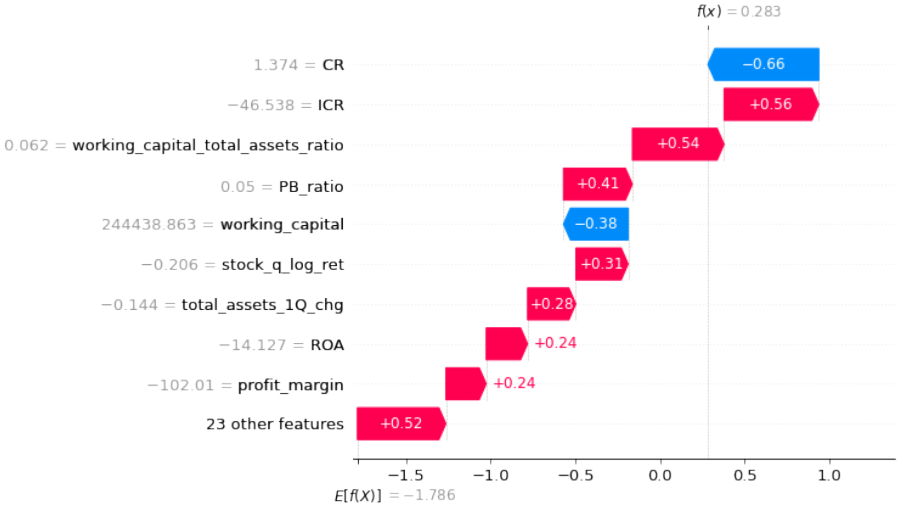


Figure 12: Waterfall plot for a true positive example. The sample was predicted to be in a state of distress in the ensuing quarter with a probability of 57%. Red arrows indicate contribution toward a positive prediction, while blue lines draw the prediction in the opposite direction. The length of the arrows represents the magnitude of the SHAP values contributed to the output by each feature for this sample. Despite a positive CR and a relatively high Working Capital, the company is correctly classified as distressed.

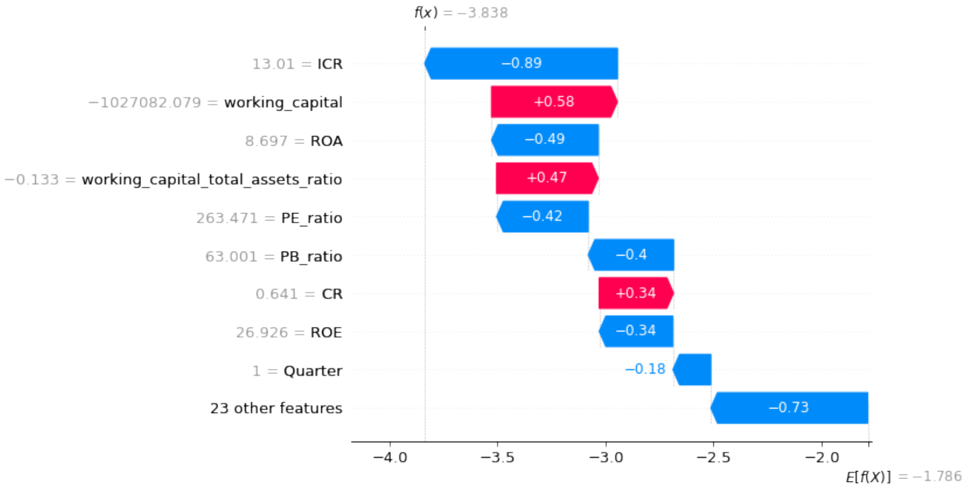


Figure 13: Waterfall plot for a true negative example. The sample was predicted to be in a state of distress in the ensuing quarter with a probability of 2.1%. Red arrows indicate contribution toward a positive prediction, while blue lines draw the prediction in the opposite direction. The length of the arrows represents the magnitude of the SHAP values contributed to the output by each feature for this sample. Despite evidence indicating a low degree of solvency and liquidity, signified by red Working Capital, WCTA and CR, the company is correctly classified as healthy since it performs well in terms of earnings.

Figure 13 shows a waterfall plot of a true negative from the test set. By using Equation 6, the sample was predicted to be in a state of distress in the ensuing quarter with a probability of 2.1%. In addition to having negative working capital and a large negative value of WCTA, the company displays signs of low liquidity, indicated by a low CR. Red arrows signify that these factors contribute toward a positive prediction, but the impact is mitigated by a high degree of liquidity in terms of earnings. Evident from Figure 10, a low feature value of WCTA usually contributes to a higher degree toward a positive prediction, but conditioned upon high ICR, the SHAP value becomes much smaller. This is in line with the trend we observed in the ICR dependence plot in Figure 11, where high ICR (right part of graph area) reduces the SHAP contribution for negative WCTA values (blue dots). This example illustrates how the model extracts and weights relevant information to determine that a company that was not in a state of financial distress could be expected to stay in that state in the ensuing quarter.

To address the core of this section, we found that the LightGBM outperformed the other models in terms of F1 and ROC-AUC scores and further concluded that the LightGBM is the superior model for the task outlined by this paper. By examining the LightGBM's behaviour with TreeSHAP, we explained feature importance and uncovered significant interaction effects that were captured by the model. We note from the analysis of SHAP plots that several explanations are in line with financial intuition, exemplified by examining the decision process behind the predictions on two samples.

6 Conclusion and Further Work

This paper proposes an interpretable early warning model for financial distress in listed Nordic corporations. Predictions are based on insight into their financial situation from accounting data and information about financial markets and macroeconomic trends. By using a proxy-based definition of financial distress rather than relying on juridically recorded credit events, our model proves effective as an early warning tool. Our proxy, which targets measures of solvency and liquidity, is in accordance with standard bond and loan covenants and otherwise in line with financial intuition and industry practices. All three models achieve ROC-AUC scores between 0.92 and 0.93, and the highest F1 score of 0.63 is obtained by the LightGBM, surpassing the remaining models by a notable margin.

Our research suggests that it is possible to generalize firm characteristics and behaviour across Nordic countries. This view finds support in the feature selection step, as the categorical feature intended to capture geographical effects receives very low feature importance. The other categorical features, capturing industry sector and seasonality, prove far more important. Findings from the data cleaning process also imply cross-border generality, with relatively similar class distributions in the four targeted countries. Furthermore, similarities in political governing, legal systems, and audit standards are likely factors that help enhance this effect.

With our proposed end-to-end framework for data collection and processing, we have created a unique and robust international dataset. In combination with model explanations from XAI framework SHAP and insight into hyperparameter importance and correlation with WandB, our paper provides a transparent data treatment procedure replicable for both academic and industrial purposes. Furthermore, we have shown that by applying this approach, our tuned ML models yield strong results, outperforming a trusted benchmark model.

Although LightGBM is not inherently interpretable, we provide insight into how the different input features affect both the model's overall output and predictions for individual instances by applying TreeSHAP. Our findings show that features related to liquidity, solvency and size are highly important to the model output. Further, it turns out that including macro, market and seasonality information, captured by GDP growth, log returns of the stock and industry index, Volatility, GICS Code and Quarter, provide clear advantageous interaction effects with other variables. Seasonality effects on a quarterly basis, captured by the Quarter feature, have rarely been recorded in related literature before. Overall, the results from the SHAP analysis are in line with financial intuition, coinciding with conclusions from related literature.

Throughout the process of creating this paper, we have identified certain limiting factors and areas for future improvement of our research. Three particular areas of interest include: (i) further investigation of characteristics of companies moving between states of distress and non-distress in subsequent time intervals, (ii) measures to improve data quality, and (iii) examining time-dependent effects such as structural breaks. First, to properly observe characteristics of companies transitioning between states in subsequent time intervals, a richer dataset would be preferable, requiring more data points. Secondly, the issue of inconsistent data quality is a core element to improve. Possible solutions to mitigate this issue could be the merger of financial databases, data filling techniques, and more analytical approaches to the trade-off between dropping rows and dropping features. Some of these elements were tested in this paper. However, efforts to improve the dataset are recommended as a first step to further research, as we believe that any additional advancement of the model essentially hinges upon the size and quality of the dataset. Finally, even though our model accounts for time-dependent effects to some extent, structural breaks are not explicitly accounted for. A deeper look into, for instance, structural break indicator variables or in-quarter transformations to handle time-dependent effects is, therefore, a suggested area for further research with access to data with more densely populated quarters.

Reference List

Aas, K., Jullum, M. and Løland, A. (2021). 'Explaining individual predictions when features are dependent: More accurate approximations to Shapley values'. *Artificial Intelligence* 298.

Abadi, M. et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Abiyev, R. H. (2014). 'Credit rating using type-2 fuzzy neural networks'. *Mathematical Problems in Engineering.*

Agrawal, K. and Maheshwari, Y. (2019). 'Efficacy of industry factors for corporate default prediction'. *IIMB Management Review* 31 (1), pp. 71–77.

Akerlof, G. A. (1978). 'The market for "lemons": Quality uncertainty and the market mechanism'. *Uncertainty in economics*, pp. 235–251.

Alkhazali, O. and Zoubi, T. (2005). 'Empirical Testing Of Different Alternative Proxy Measures For Firm Size'. *Journal of Applied Business Research* 21, pp. 79–90.

Altman, E. (1968). 'Financial ratios, discriminant analysis and the prediction of corporate bankruptcy'. *The journal of finance* 23 (4), pp. 589–609.

Altman, E., Haldeman, R. G. and Narayanan, P. (1977). 'ZETATM analysis A new model to identify bankruptcy risk of corporations'. *Journal of banking & finance* 1 (1), pp. 29–54.

Arrieta, A. B. et al. (2020). 'Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI'. *Information fusion* 58, pp. 82–115.

Awais, M. et al. (2015). 'Do Z-Score and Current Ratio have Ability to Predict Bankruptcy'. *Developing Country Studies* 5 (13), pp. 30–36.

Aziz, M. A. and Dar, H. A. (2006). 'Predicting corporate bankruptcy: where we stand?' *Corporate Governance: The international journal of business in society* 6 (1), pp. 18–33.

Balcaen, S. and 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), pp. 63–93.

Beaver, W. H. (1966). 'Financial Ratios As Predictors of Failure'. *Journal of Accounting Research* 4, pp. 71–111.

Bellovary, J. L., Giacomino, D. E. and Akers, M. D. (2007). 'A review of bankruptcy prediction studies: 1930 to present'. *Journal of Financial education*, pp. 1–42.

Bentejac, C., Csorgo, A. and Martinez-Munoz, G. (2021). 'A comparative analysis of gradient boosting algorithms'. *Artificial Intelligence Review* 54 (3), pp. 1937–1967.

Berkson, J. (1944). 'Application of the Logistic Function to Bio-Assay'. *Journal of the American Statistical Association* 39 (227), pp. 357–365.

Biewald, L. (2020). *Experiment Tracking with Weights and Biases.* URL: https://www.wandb.com/.

Bloomberg (2022). *Bloomberg professional.* [Online]. Available at: Subscription service (visited on 6th Apr. 2022).

Blum, M. (1974). 'Failing company discriminant analysis'. *Journal of accounting research*, pp. 1–25.

Bonfim, D. (2009). 'Credit risk drivers: Evaluating the contribution of firm level information and of macroeconomic dynamics'. *Journal of Banking & Finance* 33 (2), pp. 281–299.

Buchanan, B. G. and Shortliffe, E. H. (1984). 'Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project'.

Bussmann, N. et al. (2021). 'Explainable machine learning in credit risk management'. *Computational Economics* 57 (1), pp. 203–216.

Campbell, J. Y., Hilscher, J. and Szilagyi, J. (2008). 'In search of distress risk'. *The Journal of Finance* 63 (6), pp. 2899–2939.

Campbell, J. Y., Hilscher, J. D. and Szilagyi, J. (2011). 'Predicting financial distress and the performance of distressed stocks'. *Journal of Investment Management* 9 (2), pp. 71–77.

Chakravorty, D. (n.d.). *Confusion Matrix.* URL: https://www.debadityachakravorty.com/ai-ml/cmatrix/ (visited on 16th Mar. 2022).

Chandrasekaran, B., Tanner, M. C. and Josephson, J. R. (1989). 'Explaining control strategies in problem solving'. *IEEE Intelligent Systems* 4 (1), pp. 9–15.

Charalambakis, E. C. and Garrett, I. (2018). 'On corporate financial distress prediction: What can we learn from private firms in a developing economy? Evidence from Greece'. *Review of quantitative finance and accounting* 52 (2), pp. 467–491.

Chava, S. and Jarrow, R. A. (2004). 'Bankruptcy prediction with industry effects'. *Review of finance* 8 (4), pp. 537–569.

Chen, J. et al. (2006). 'Financial distress prediction in China'. *Review of Pacific Basin Financial Markets and Policies* 9 (2), pp. 317–336.

Chen, K. H. and Shimerda, T. A. (1981). 'An empirical analysis of useful financial ratios'. *Financial management* 10 (1), pp. 51–60.

Chen, T. and Guestrin, C. (2016). 'Xgboost: A scalable tree boosting system'. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Cheng, C.-H., Chan, C.-P. and Sheu, Y.-J. (2019). 'A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction'. *Engineering Applications of Artificial Intelligence* 81, pp. 283–299.

Choi, D. et al. (2019). 'On Empirical Comparisons of Optimizers for Deep Learning'. *arXiv preprint arXiv:1910.05446*.

Chollet, F. (2015). *Keras*. URL: https://github.com/fchollet/keras.

Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2015). 'Fast and accurate deep network learning by exponential linear units (elus)'. *arXiv preprint arXiv:1511.07289*.

Dambolena, I. G. and Khoury, S. J. (1980). 'Ratio stability and corporate failure'. *The Journal of Finance* 35 (4), pp. 1017–1026.

Deakin, E. B. (1972). 'A discriminant analysis of predictors of business failure'. *Journal of accounting research* 10 (1), pp. 167–179.

Demajo, L. M., Vella, V. and Dingli, A. (2020). 'Explainable ai for interpretable credit scoring'. *arXiv preprint arXiv:2012.03749*.

Desai, C. (2020). 'Comparative Analysis of Optimizers in Deep Neural Networks'. *International Journal of Innovative Science and Research Technology* 5 (10), pp. 959–962.

Dimitras, A. I., Zanakis, S. H. and Zopounidis, C. (1996). 'A survey of business failures with an emphasis on prediction methods and industrial applications'. *European journal of operational research* 90 (3), pp. 487–513.

Du, X. et al. (2020). 'CUS-heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection'. *Applied Soft Computing* 97.

Edmister, R. O. (1972). 'An empirical test of financial ratio analysis for small business failure prediction'. *Journal of Financial and Quantitative analysis* 7 (2), pp. 1477–1493.

Eisenbeis, R. A. (1977). 'Pitfalls in the application of discriminant analysis in business, finance, and economics'. *The Journal of Finance* 32 (3), pp. 875–900.

European Central Bank Statistical Data Warehouse (2022). *Harmonized Index of Consumer Prices - Overall Index (Euro Area)*. URL: https://sdw.ecb.europa.eu/quickview.do?SERIES_KEY=122.ICP.M.U2.N.000000.4.INX (visited on 12th May 2022).

European Commission (2019). *Ethics guidelines for Trustworthy AI*. URL: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai (visited on 9th Feb. 2022).

Falavigna, G. (2012). 'Financial ratings with scarce information: A neural network approach'. *Expert Systems with Applications* 39 (2), pp. 1784–1792.

Fitch Ratings (2021). *Corporate Rating Criteria*. URL: https://www.fitchratings.com/research/corporate-finance/corporate-rating-criteria-15-10-2021 (visited on 13th May 2022).

Friedman, J. H. (2001). 'Greedy Function Approximation: A Gradient Boosting Machine'. *The Annals of Statistics* 29 (5), pp. 1189–1232.

Frydman, H., Altman, E. and Kao, D.-L. (1985). 'Introducing recursive partitioning for financial classification: the case of financial distress'. *The journal of finance* 40 (1), pp. 269–291.

Fryer, D., Strümke, I. and Nguyen, H. (2021). 'Shapley values for feature selection: The good, the bad, and the axioms'. *IEEE Access* 9, pp. 144352–144360.

Guha, R. et al. (2021). 'CGA: A new feature selection model for visual human action recognition'. *Neural Computing and Applications* 33 (10), pp. 5267–5286.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005). 'Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning'. *Advances in Intelligent Computing*. Springer Berlin Heidelberg, pp. 878–887.

Hornik, K. (1991). 'Approximation capabilities of multilayer feedforward networks'. *Neural Networks* 4 (2), pp. 251–257.

Horowitz, J. L. and Savin, N. (2001). 'Binary response models: Logits, probits and semiparametrics'. *Journal of Economic Perspectives* 15 (4), pp. 43–56.

Hu, Y.-C. and Ansell, J. (2007). ‘Measuring retail company performance using credit scoring techniques’. *European Journal of Operational Research* 183 (3), pp. 1595–1606.

Huang, Y.-P. and Yen, M.-F. (2019). ‘A new perspective of performance comparison among machine learning algorithms for financial distress prediction’. *Applied Soft Computing* 83.

Iturriaga, F. J. L. and Sanz, I. P. (2015). ‘Bankruptcy visualization and prediction using neural networks: A study of US commercial banks’. *Expert Systems with applications* 42 (6), pp. 2857–2869.

Jan, C.-L. (2021). ‘Financial information asymmetry: Using deep learning algorithms to predict financial distress’. *Symmetry* 13 (3).

Jensen, H. L. (1992). ‘Using neural networks for credit scoring’. *Managerial finance* 18 (6), pp. 15–26.

Jiang, Y. and Jones, S. (2018). ‘Corporate distress prediction in China: A machine learning approach’. *Accounting & Finance* 58 (4), pp. 1063–1109.

Jones, S., Johnstone, D. and Wilson, R. (2017). ‘Predicting corporate bankruptcy: An evaluation of alternative statistical frameworks’. *Journal of Business Finance & Accounting* 44 (1-2), pp. 3–34.

Jothi, N., Husain, W. and Rashid, N. A. (2021). ‘Predicting generalized anxiety disorder among women using Shapley value’. *Journal of infection and public health* 14 (1), pp. 103–108.

Kamath, U. and Liu, J. (2021). *Explainable Artificial Intelligence: An Introduction to Interpretable Machine Learning.* Springer.

Ke, G. et al. (2017). ‘Lightgbm: A highly efficient gradient boosting decision tree’. *Advances in neural information processing systems* 30, pp. 3146–3154.

Kim, H., Cho, H. and Ryu, D. (2020). ‘Corporate default predictions using machine learning: Literature review’. *Sustainability* 12 (16).

Kozlovskyi, S. et al. (2019). ‘Management and comprehensive assessment of the probability of bankruptcy of Ukrainian enterprises based on the methods of fuzzy sets theory’. *Problems and Perspectives in Management* 17 (3), pp. 370–381.

Kumar, P. R. and Ravi, V. (2007). ‘Bankruptcy prediction in banks and firms via statistical and intelligent techniques–A review’. *European journal of operational research* 180 (1), pp. 1–28.

Lemaître, G., Nogueira, F. and Aridas, C. K. (2017). ‘Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning’. *Journal of Machine Learning Research* 18 (17), pp. 1–5.

Liang, D. et al. (2016). ‘Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study’. *European Journal of Operational Research* 252 (2), pp. 561–572.

Lin, L. and Piesse, J. (2004). ‘Identification of corporate distress in UK industrials: a conditional probability analysis approach’. *Applied Financial Economics* 14 (2), pp. 73–82.

Liu, W., Fan, H. and Xia, M. (2022). ‘Credit scoring based on tree-enhanced gradient boosting decision trees’. *Expert Systems with Applications* 189.

Lundberg, S. (2018). *Welcome to the SHAP documentation.* URL: https://shap.readthedocs.io/en/latest/index.html (visited on 21st Apr. 2022).

Lundberg, S., Erion, G. and Lee, S.-I. (2018). ‘Consistent individualized feature attribution for tree ensembles’. *arXiv preprint arXiv:1802.03888.*

Lundberg, S. M. and Lee, S.-I. (2017). ‘A unified approach to interpreting model predictions’. *Advances in neural information processing systems* 30.

Luoma, M. and Laitinen, E. (1991). ‘Survival analysis as a tool for company failure prediction’. *Omega* 19 (6), pp. 673–678.

Malakauskas, A. and Lakstutiene, A. (2021). ‘Financial distress prediction for small and medium enterprises using machine learning techniques’. *Inžinerinė ekonomika* 32 (1), pp. 4–14.

Marcilio, W. E. and Eler, D. M. (2020). ‘From explanations to feature selection: assessing shap values as feature selection mechanism’. *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 340–347.

McCulloch, W. S. and Pitts, W. (1943). ‘A logical calculus of the ideas immanent in nervous activity’. *The bulletin of mathematical biophysics* 5 (4), pp. 115–133.

Messier Jr, W. F. and Hansen, J. V. (1988). ‘Inducing rules for expert system development: an example using default and bankruptcy data’. *Management Science* 34 (12), pp. 1403–1415.

Microsoft Corporation (2022). *Welcome to LightGBM’s documentation.* URL: https://lightgbm.readthedocs.io/en/latest/index.html# (visited on 6th May 2022).

Minh, D. et al. (2022). 'Explainable artificial intelligence: a comprehensive review'. *Artificial Intelligence Review* 57, pp. 3503–3568.

Mohammed, A. A. E. and Kim-Soon, N. (2012). 'Using Altman's model and current ratio to assess the financial status of companies quoted in the Malaysian stock exchange'. *International Journal of Scientific and Research Publications* 2 (7), pp. 1–11.

Mokhtari, K. E., Higdon, B. P. and Başar, A. (2019). 'Interpreting financial time series with SHAP values'. *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pp. 166–172.

Moody's (2022). *Rating Methodologies.* URL: https://www.moodys.com/researchandratings/methodology/003006001/rating-methodologies/methodology/003006001/003006001/-/0/0/-/0/-/-/en/global/rr (visited on 10th May 2022).

Morningstar DBRS (2022). *General Corporate Methodology.* URL: https://www.dbrsmorningstar.com/research/394214/general-corporate-methodology (visited on 13th May 2022).

Moscatelli, M. et al. (2020). 'Corporate default forecasting with machine learning'. *Expert Systems with Applications* 161.

Nordic Credit Rating (2022a). *Stendörren Fastigheter AB (publ).* URL: https://nordiccreditrating.com/issuer/stendorren-fastigheter-ab-publ?language_content_entity=en (visited on 21st Apr. 2022).

— (2022b). *Studentbostäder i Norden AB (publ).* URL: https://nordiccreditrating.com/issuer/studentbostader-i-norden-ab-publ (visited on 21st Apr. 2022).

Ogundimu, E. O. (2019). 'Prediction of default probability by using statistical models for rare events'. *Journal of the Royal Statistical Society* 182 (4), pp. 1143–1162.

Ohlson, J. A. (1980). 'Financial Ratios and the Probabilistic Prediction of Bankruptcy'. *Journal of accounting research* 18 (1), pp. 109–131.

Olson, D. L., Delen, D. and Meng, Y. (2012). 'Comparative analysis of data mining methods for bankruptcy prediction'. *Decision Support Systems* 52 (2), pp. 464–473.

Parhi, R. and Nowak, R. D. (2020). 'The Role of Neural Network Activation Functions'. *IEEE Signal Processing Letters* 27, pp. 1779–1783.

Park, M. S. et al. (2021). 'Explainability of Machine Learning Models for Bankruptcy Prediction'. *IEEE Access* 9, pp. 124887–124899.

Pedregosa, F. et al. (2011). 'Scikit-learn: Machine Learning in Python'. *Journal of Machine Learning Research* 12, pp. 2825–2830.

Póczos, B. (2013). *10-725 Convex Optimization, Lecture 2.* (Accessed: 24. November 2021). URL: http://www.stat.cmu.edu/~ryantibs/convexopt-F13/scribes/lec2.pdf.

Qian, H. et al. (2022). 'Financial distress prediction using a corrected feature selection measure and gradient boosted decision tree'. *Expert Systems with Applications* 190.

Quinto, B. (2020). *Next-Generation Machine Learning with Spark: Covers XGBoost, LightGBM, Spark NLP, Distributed Deep Learning with Keras, and More.* Springer.

Refinitiv Eikon (2022). *Refinitiv Eikon Company Database.* [Online]. Available at: Subscription service (visited on 10th May 2022).

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016). '" Why should i trust you?" Explaining the predictions of any classifier'. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

S&P (2013). *Corporate Methodology.* URL: https://www.spratings.com/scenario-builder-portlet/pdfs/CorporateMethodology.pdf (visited on 10th May 2022).

Shapley, L. S. (1953). 'Stochastic Games'. *Proceedings of the National Academy of Sciences - PNAS* 39 (10), pp. 1095–1100.

Shehadeh, A. et al. (2021). 'Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression'. *Automation in Construction* 129.

Smith, L. N. (2017). 'Cyclical Learning Rates for Training Neural Networks'. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472.

Son, H. et al. (2019). 'Data analytic approach for bankruptcy prediction'. *Expert Systems with Applications* 138.

Sormunen, N. et al. (2013). 'Harmonisation of Audit Practice: Empirical Evidence from Going-Concern Reporting in the N ordic Countries'. *International journal of auditing* 17 (3), pp. 308–326.

Srivastava, N. et al. (2014). ‘Dropout: a simple way to prevent neural networks from overfitting’. *The journal of machine learning research* 15 (1), pp. 1929–1958.

Swartout, W. R. and Moore, J. D. (1993). ‘Explanation in second generation expert systems’. *Second generation expert systems.* Springer, pp. 543–585.

Tam, K. Y. (1991). ‘Neural network models and the prediction of bank bankruptcy’. *Omega* 19 (5), pp. 429–445.

Tian, S. and Yu, Y. (2017). ‘Financial ratios and bankruptcy predictions: An international evidence’. *International Review of Economics & Finance* 51, pp. 510–526.

Ugurlu, M. and Aksoy, H. (2006). ‘Prediction of corporate financial distress in an emerging market: the case of Turkey’. *Cross Cultural Management: An International Journal* 13 (4).

Vinayak, R. K. and Gilad-Bachrach, R. (2015). ‘Dart: Dropouts meet multiple additive regression trees’. *Artificial Intelligence and Statistics*, pp. 489–497.

Virtanen, P. et al. (2020). ‘SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python’. *Nature Methods* 17, pp. 261–272.

West, D. (2000). ‘Neural network credit scoring models’. *Computers & operations research* 27 (11-12), pp. 1131–1152.

Wick, M. R. and Thompson, W. B. (1992). ‘Reconstructive expert system explanation’. *Artificial Intelligence* 54 (1-2), pp. 33–70.

Xu, S. and Chen, L. (2008). ‘A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN’s and Its Application in Data Mining’. *5th International Conference on Information Technology and Applications (ICITA 2008)*, pp. 683–686.

Xu, X. and Wang, Y. (2009). ‘Financial failure prediction using efficiency as a predictor’. *Expert Systems with Applications* 36 (1), pp. 366–373.

Yang, Z., Platt, M. B. and Platt, H. D. (1999). ‘Probabilistic neural networks in bankruptcy prediction’. *Journal of business research* 44 (2), pp. 67–74.

Yeh, I. and Lien, C. (2009). ‘The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients’. *Expert Systems with Applications* 36 (2), pp. 2473–2480.

Zhang, C. and Ma, Y. (2012). *Ensemble machine learning: methods and applications.* Springer.

Zhang, D. and Gong, Y. (2020). ‘The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure’. *IEEE Access* 8, pp. 220990–221003.

Zhang, Y. and Shi, B. (2018). ‘Does default point vary with firm size?’ *Applied Economics Letters* 25, pp. 1078–1082.

Zheng, X.-l. et al. (2019). ‘FinBrain: when finance meets AI 2.0’. *Frontiers of Information Technology & Electronic Engineering* 20 (7), pp. 914–924.

Appendix

A Logistic Regression

Dichotomous or binary LR was first proposed and popularized by Berkson (1944) and has been popularly applied as a benchmark model for predicting corporate bankruptcy. LR works similarly to linear regression, but instead of providing a continuous value as a prediction, it restricts the output to the range [0, 1], which can be interpreted as a probability, p, that a given input $X = x_1$, x_2, x_3 ... x_n belongs to a certain class. This is achieved by fitting an S-shaped logistic function, the Sigmoid function, to the data by using Maximum Likelihood Estimation (MLE), instead of fitting a straight line with Ordinary Least Squares (OLS).

First, the Logit-function, given in Equation 7, is used to transform the output from probability to the log of the odds of probability, which has advantageous interpretations. Through the logarithmic transformation, the non-linear association can be modeled linearly, and the target axis is extended to ±∞. This is shown in the leftmost plot in Figure 14.

$$Logit(p) = ln(\frac{p}{1-p}) \tag{7}$$

Further, a candidate for the best fitting line is proposed through the coefficients $\beta_0 + \sum_{i=1}^{p} \beta_i X_i$ which determine the line in the leftmost plot in Figure 14. The log of the odds is obtained by projecting the observations onto the candidate line. It becomes evident that estimating the best fitting line through OLS is unattainable as the residuals of the observations would be infinite. Therefore, the log of the odds are converted to probabilities by using the Sigmoid-function (see Equation 8) so that MLE can be used to find the best candidate line. The result of this conversion can be seen in the rightmost plot in Figure 14. The candidate line with the highest log of the probabilities is chosen for the LR model.

$$p(x_1, x_2, ..., x_p) = \frac{e^{\beta_0 + \sum_{i=1}^{p} \beta_i X_i}}{1 + e^{\beta_0 + \sum_{i=1}^{p} \beta_i X_i}} \tag{8}$$

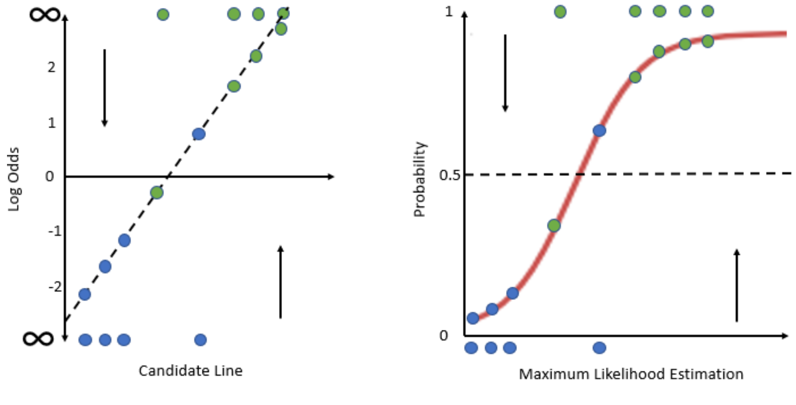


Figure 14: Conversion process for LR from log-odds to probabilities.

Additionally, applying the log has the advantage that it relaxes some of the assumptions that constrain DA like multivariate normality and equal variance-covariance matrices (Moscatelli et al., 2020). The new axes are no longer constrained between 0 and 1, and the output coefficients become interpretable. If the log of the odds is positive, the prediction is true, and vice versa.

Even though the advantages of LR have placed it among the most celebrated classification algorithms, there are still some limitations. Firstly, the consideration of non-linear or complex interactions between input and output is lacking, making it hard to fully exploit large datasets. Secondly, the sensitivity to outliers or missing data demands high-quality datasets as a single outlier will impact the fitting of the Sigmoid-function (Moscatelli et al., 2020).

The implementation used in this paper is the standard LR algorithm from the Python library Scikit-learn (Pedregosa et al., 2011). The built-in algorithm incorporates regularization to counteract overfitting.

B Artificial Neural Networks

The idea behind ANNs was first proposed by McCulloch and Pitts (1943), and has become one of the most prominent ML techniques available. ANN is a supervised ML technique used to perform classification along with several other tasks. Inspired by the structure of the human brain, the ANN consists of numerous nodes, known as perceptrons. Perceptrons are normally organized in a layered manner, where each perceptron is connected to all perceptrons in its neighboring layers. When an input is passed through the network, a weighted sum of the output from perceptrons in the preceding layer is passed through an activation function within each perceptron, yielding the output for that perceptron to the perceptrons in the next layer. There are several possible configurations of the ANN. This paper uses a Multilayer Perceptron (MLP), which organizes the perceptrons in fully connected layers as described above. The network can have any number of layers, but at minimum, an input and an output layer have to be present, as illustrated in Figure 15. When input information is moving through the layers, passing through perceptrons along the way, in one direction, the network is known as a Sequential, or Feed-Forward, network.

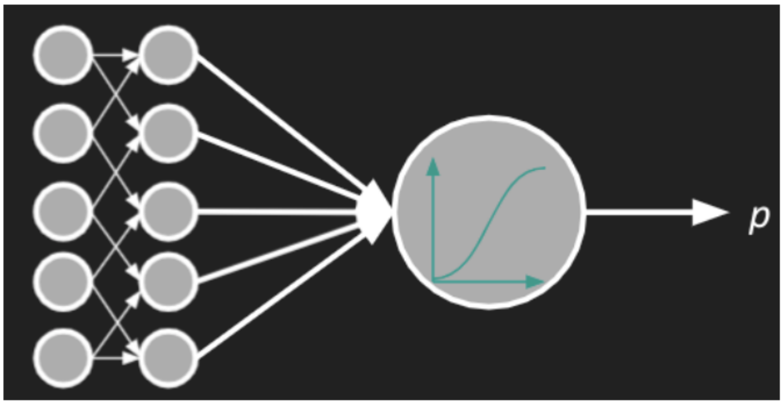


Figure 15: Illustration of an ANN. Nodes in each layer are normally fully connected to the next, but some arrows have been omitted for visualization purposes. Using the Sigmoid-function as the final activation function will map the input to a value between 0 and 1, which can be interpreted as a probability.

When the number inside a perceptron is larger than some threshold or bias, the perceptron is said to be activated. The number in all perceptrons except the input layer is produced by multiplying the output of preceding perceptrons with their associated weight and subtracting a bias as shown in Equation 9. The sum of these products is then passed as an argument, x, into the activation function, such as a Sigmoid function, shown in Equation 10, providing an output for that perceptron which is passed to the next layer. In this manner, the initial input moves from one layer to another until it reaches the output layer.

$$a_0^{(1)} = w_{0,0} * a_0^{(0)} + w_{0,1} * a_1^{(0)} + \cdots + w_{0,n} * a_n^{(0)} + b_0 \tag{9}$$

$$S(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

There are numerous different activation functions, but the Sigmoid function is the most popular for explaining neural networks. The Sigmoid-function maps the input to a value between 0 and 1. Another popular activation function is the ReLU, illustrated in Figure 5, which maps all positive values to themselves and all negative values to 0.

The computation of perceptron values in a forward pass (i.e. as input moves through the network) is conducted using matrix multiplication, shown in Equation 11. The output layer can consist of more than one perceptron, and in that case, the perceptron with the highest value is output as the answer to a classification problem. Essentially, a neural network is a function that, based on some input, provides an educated guess for what class that input belongs to.

$$\mathbf{a}^{(1)} = S(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}) = S(\mathbf{W}\mathbf{a}^0 + \mathbf{b}) \tag{11}$$

The steps outlined above describe how a neural network makes a prediction, but before the network provides a meaningful prediction it has to go through a training process. As mentioned, the ANN is a supervised learner, which means that it requires labeled observations to perform training. Once a prediction is made, a loss function similar to Equation 12 measures how far from the correct prediction the output was, where y_i refers to the correct value, and $f_\theta(x_i)$ refers to the network's prediction. In Equation 12, θ refers to all estimated parameters i.e. weights and biases, while f refers to the neural net as a function.

$$Loss = \sum_{i=1}^{n}(y_i - f_\theta(x_i)) \tag{12}$$

The main goal of neural networks is to minimize the loss, i.e. find the weights and biases that generate predictions close to their true target values. This is achieved by using an optimizer such as SGD, Adam, or Root Mean Square Propagation (RMSprop). Using the SGD as an example, it is a first-order optimization algorithm that utilizes the first-order derivative of the loss function. In essence, it calculates how the weights and biases should be altered to minimize the loss function. The loss-based parameter update is fed backwards through the network and is therefore called backpropagation. When all training examples are sent through the network and changes to the model parameters have been made, the model is said to be fit to the data and can be used to perform predictions on unseen observations.

The implementation of the ANN in this paper uses Keras (Chollet, 2015), which is a Python library that is built on top of Tensorflow (Abadi et al., 2015). A detailed explanation of the exact implementation can be found in Section 4. Note that the implementation in this paper does not entirely correspond to the network described here.

C Light Gradient Boosting Machine

LightGBM of Ke et al. (2017) is a gradient boosting framework used for classification and prediction problems, and has been shown to be superior to similar methods such as XGBoost and Stochastic Gradient Boosting (SGB) in terms of computational speed and memory consumption. Boosting in ML builds on the idea that a combination of simple classifiers obtained by a weak learner can work together to outperform the capability of a simple classifier alone (Zhang and Ma, 2012). Gradient boosting in general is an ensemble method where the weak learners are DTs trained in sequence. The rationale behind such ensemble methods is that it is easier to train many simple classifiers, and combining their results, than training a single complex classifier (Zhang and Ma, 2012).

Whereas most tree-based algorithms grow their trees horizontally (level/depth-wise), LightGBM grows its trees vertically (leaf-wise) (Microsoft Corporation, 2022), as illustrated in Figure 16. The former approach, which is used in XGBoost, may result in many unnecessary nodes and thus a heavier computation process. By using leaf-wise growth, the decision splits are performed only at the most promising branches and leaves, and thus the algorithm may converge faster. The aforementioned property helps optimize model accuracy by reducing loss.

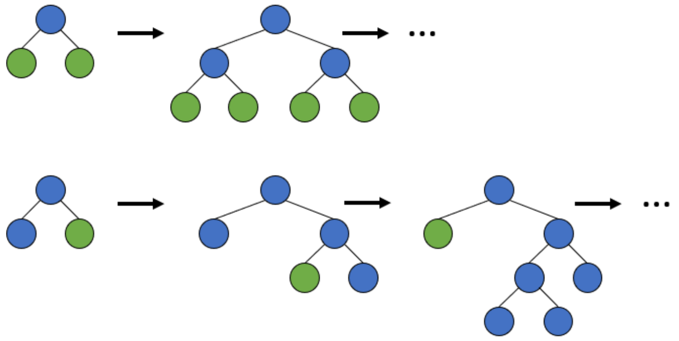


Figure 16: Level (top) compared to leaf-wise (bottom) growth for tree-structured models. Adapted from Quinto (2020, p. 146).

There are mainly three properties that make LightGBM exceptionally fast: GOSS, EFB, and histogram based splitting, referred to as binning. The former two were proposed by Ke et al. (2017), constituting their contribution to the existing GBDT algorithm. GOSS is a sampling method that involves sorting the instances by gradient and prioritizing the ones with a high gradient to be used for training. To avoid altering the data distribution, a random sample of instances is chosen from the remaining instances. Further, EFB is a technique that bundles mutually exclusive features so that the number of features is reduced. Additionally, to find the best split points, which is one of the main costs in gradient boosting algorithms, LightGBM uses a histogram-based algorithm. Such algorithms "buckets continuous feature values into discrete bins and uses these bins to construct feature histograms during training" (Ke et al., 2017, p. 2).

Even though alterations to the original GBDT increases speed and reduces memory usage, the training and predictions performed by the model are still relatively similar to those of GBDT. The procedure, proposed by Friedman (2001), can be simplified to the following steps, where x is an instance, y is the true observation, γ is the log of the odds, and M is the number of trees to build (number of iterations):

- Input: Data on the form $\{(x_i, y_i)\}_{i=1}^{n}$ and a differentiable Loss Function $L(y_i, F(x))$
- Step 1: Initialize model with a constant value $F_0(x) = argmin_\gamma \Sigma_{i=1}^{n} L(y_i, \gamma)$
- Step 2: For $m = 1$ to M:
  - A: Compute $r_{im} = -[\frac{\partial L(y_i,F(x))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$
  - B: Fit a regression tree to the r_{im} values and create terminal regions R_{jm}, for $j = 1, ..., J_m$

  - C: For $j = 1, ..., J_m$ compute $\gamma_{jm} = argmin_\gamma \Sigma_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$
  - D: Update $F_m(x) = F_{m-1}(x) + \nu \Sigma_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

- Step 3: Output: $F_M(x)$

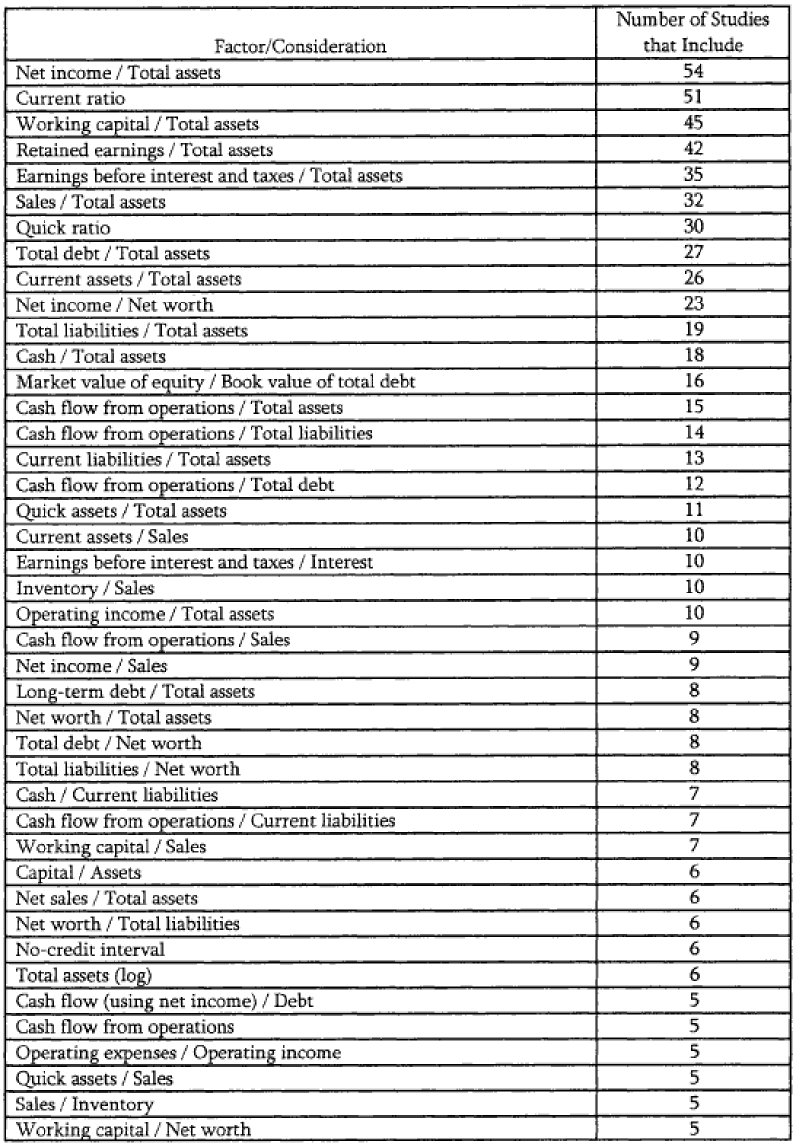
Step 1 involves initializing the $F_0(x)$ that minimizes the loss function (see Equation 13) with respect to the log of the odds. In practice this would be the probability of a positive example drawn at random. The loss function is defined as:

$$L(y_i, F(x)) = -[y \times ln(\gamma) + ln(1 + e^{ln(\gamma)})] \tag{13}$$

Further, step 2A involves calculating the pseudo residuals, i.e. intermediate residuals, r_{im}, for each instance by differentiating the loss function with respect to the most recent predicted log odds. Step 2B fits a regression tree to predict the pseudo residuals, and labels each leaf (terminal region) R_{jm}. Step 2C calculates the new output values for each leaf in this decision tree by finding the value for γ that minimizes the sum. Finally, step 2D makes a new prediction for each sample based on the previous prediction plus the learning rate, ν, and the output values from the previous trees. When $m = M$, the algorithm stops building more trees, and step 3 returns the final model, $F_M(x)$.

The implementation used in this paper uses the LightGBM framework proposed by Microsoft Corporation, 2022.

D Frequency of Financial Ratios in Literature



Factor/Consideration	Number of Studies that Include
Net income / Total assets	54
Current ratio	51
Working capital / Total assets	45
Retained earnings / Total assets	42
Earnings before interest and taxes / Total assets	35
Sales / Total assets	32
Quick ratio	30
Total debt / Total assets	27
Current assets / Total assets	26
Net income / Net worth	23
Total liabilities / Total assets	19
Cash / Total assets	18
Market value of equity / Book value of total debt	16
Cash flow from operations / Total assets	15
Cash flow from operations / Total liabilities	14
Current liabilities / Total assets	13
Cash flow from operations / Total debt	12
Quick assets / Total assets	11
Current assets / Sales	10
Earnings before interest and taxes / Interest	10
Inventory / Sales	10
Operating income / Total assets	10
Cash flow from operations / Sales	9
Net income / Sales	9
Long-term debt / Total assets	8
Net worth / Total assets	8
Total debt / Net worth	8
Total liabilities / Net worth	8
Cash / Current liabilities	7
Cash flow from operations / Current liabilities	7
Working capital / Sales	7
Capital / Assets	6
Net sales / Total assets	6
Net worth / Total liabilities	6
No-credit interval	6
Total assets (log)	6
Cash flow (using net income) / Debt	5
Cash flow from operations	5
Operating expenses / Operating income	5
Quick assets / Sales	5
Sales / Inventory	5
Working capital / Net worth	5

Figure 17: Frequency of financial ratios applied in literature between 1930 and 2007, given their appearance in five or more studies. Retrieved from Bellovary et al. (2007, p.42).

E Beta Variable Correlation

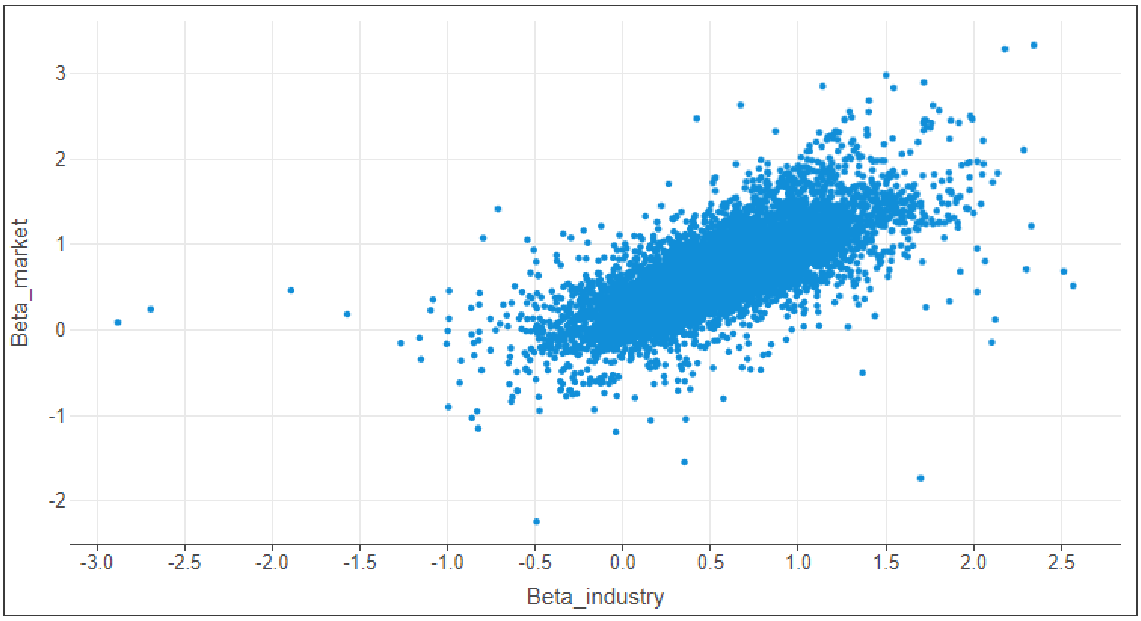


Figure 18: Scatterplot displaying Market Beta to Industry Beta values for 10 000 classification samples.

F Features Omitted Following SHAP Feature Selection

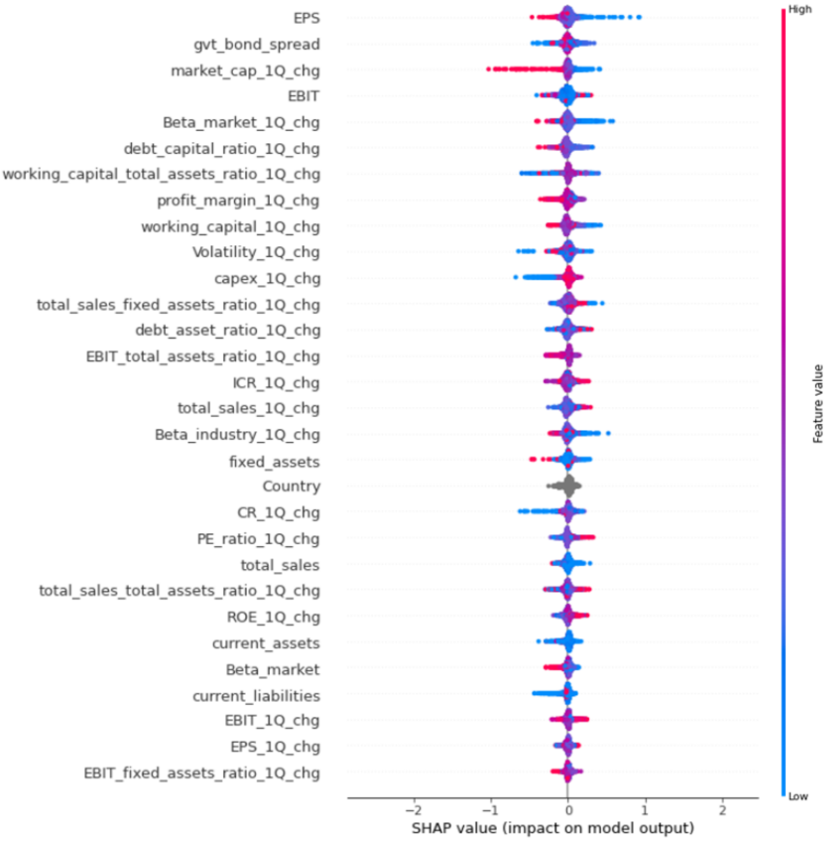


Figure 19: SHAP beeswarm plot displaying the 30 features of least importance. The listed features were omitted from further inclusion in the analysis.

G Correlation Heatmap

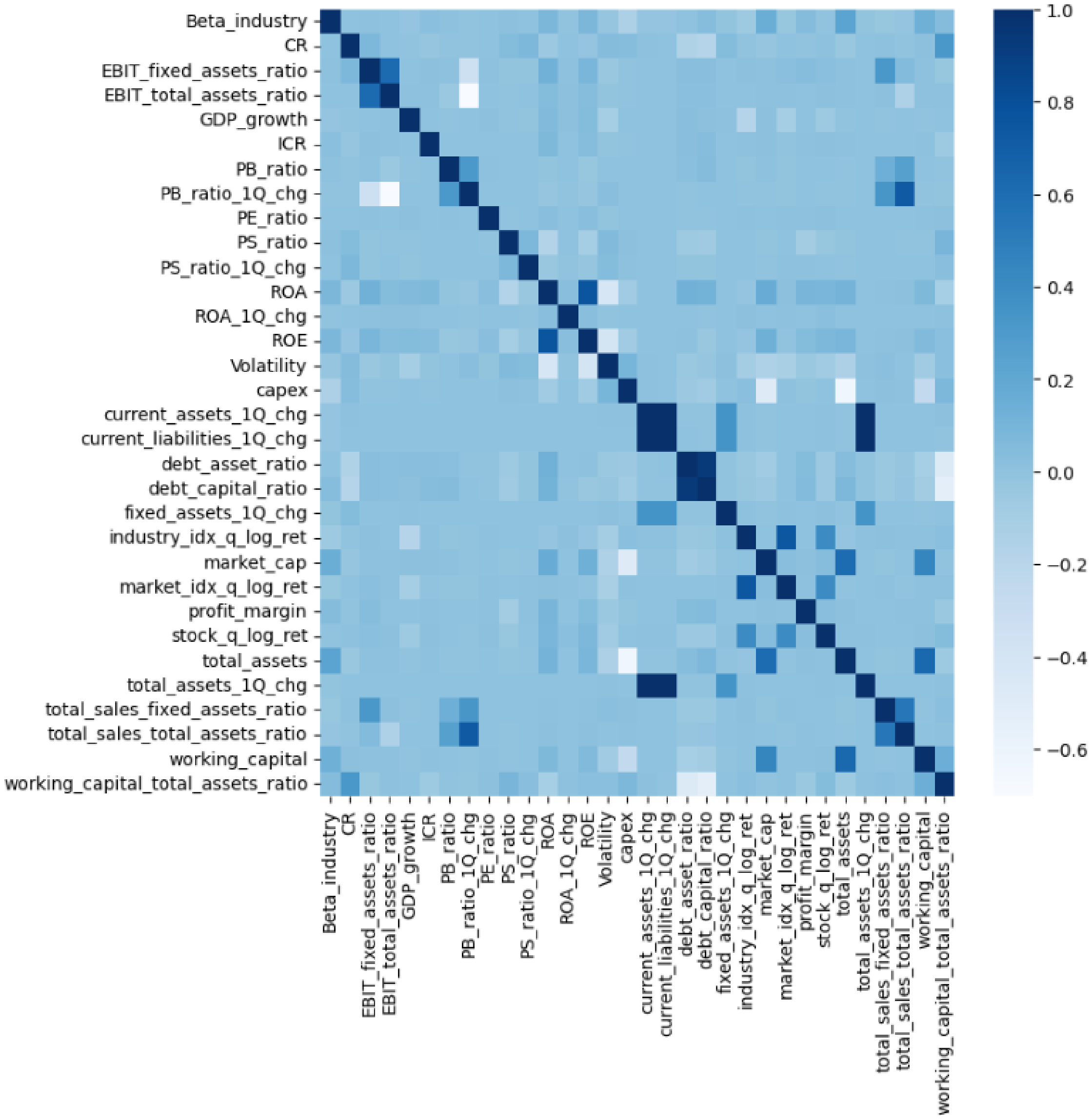


Figure 20: Correlation heatmap for the final 32 numerical features. Note that current_liabilities_1Q_chg and current_assets_1Q_chg were removed due to several high correlations.

H WandB Hyperparameter Configurations Overview - ANN

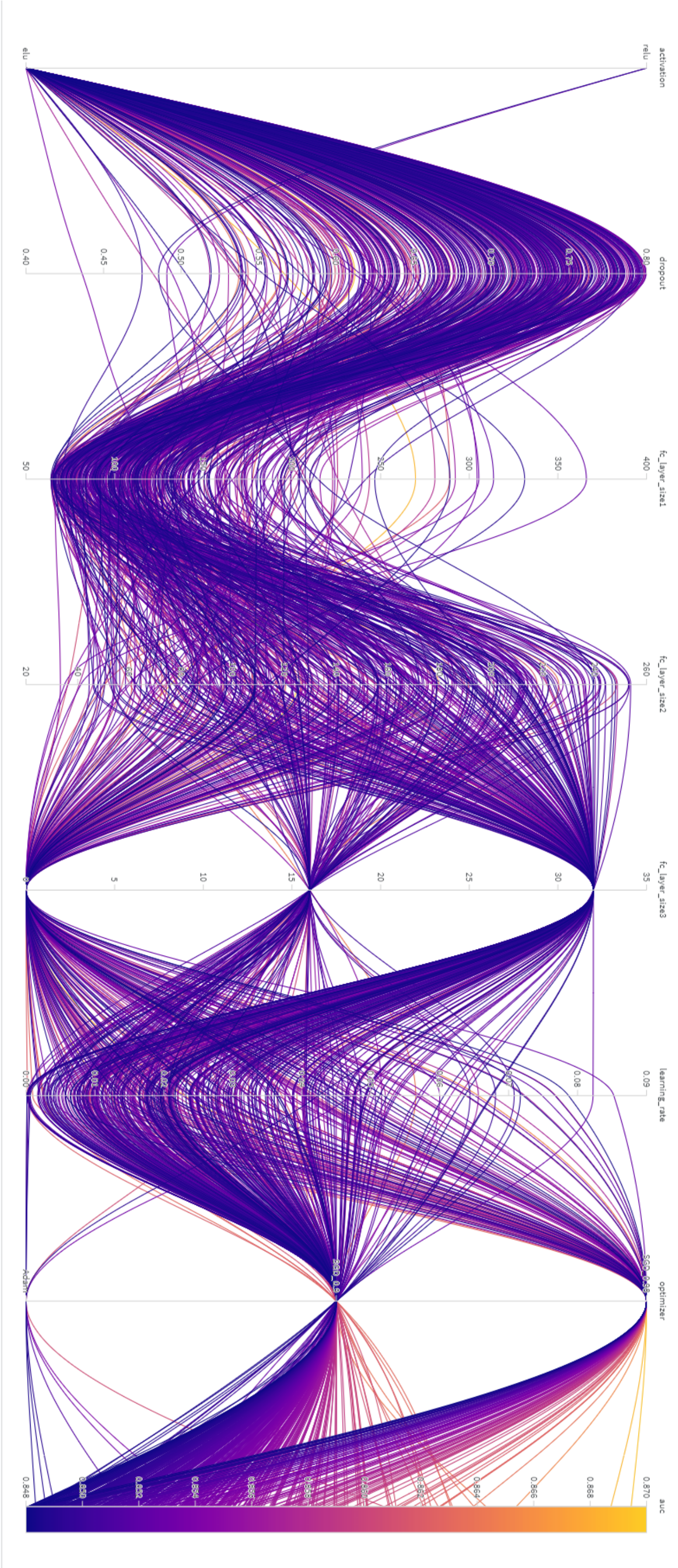


Figure 21: Overview of the top 500 different ANN configurations mapped to their maximum achieved validation ROC-AUC score during training with WandB.

I WandB Hyperparameter Configurations Overview - LightGBM

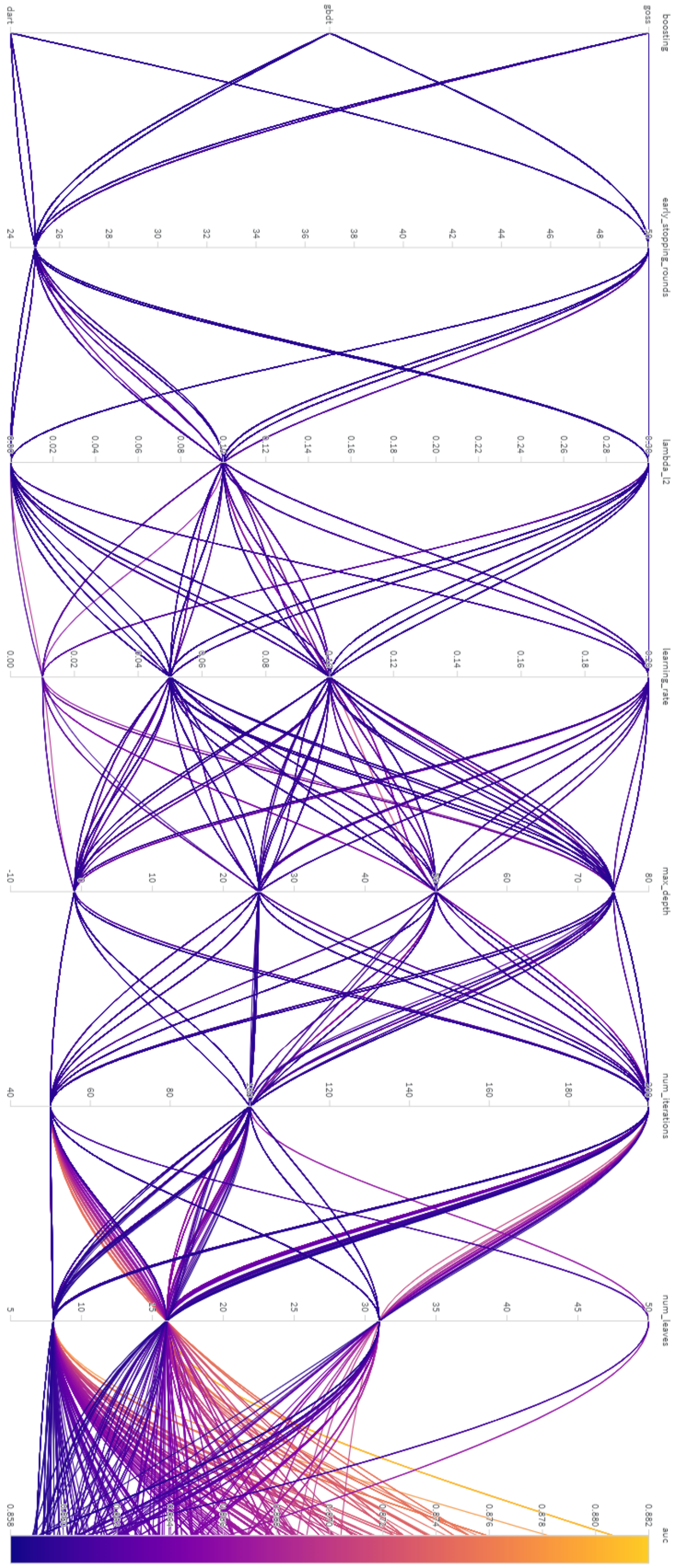


Figure 22: Overview of the top 500 different LightGBM configurations mapped to their maximum achieved validation ROC-AUC score during training with WandB.

J Dependence Plots for Categorical, Macro and Market Variables

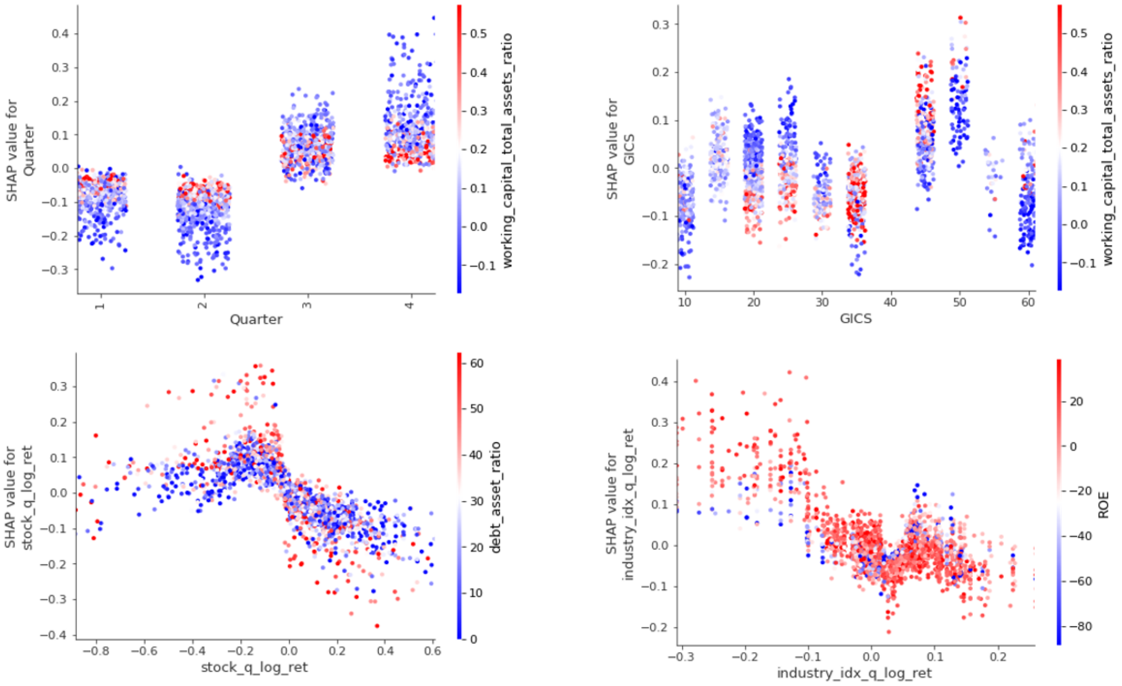


Figure 23: Dependence plot for categorical, macro and market variables. Quarter shows clear interactive effects with WCTA, displaying that WCTA values have opposite impacts on model output depending on whether the observation is from the first half or second half of the year. Similar effects observable for Stock Log Returns, where high values of Debt Asset Ratio have opposite impacts depending on whether the quarterly log returns were positive or negative.

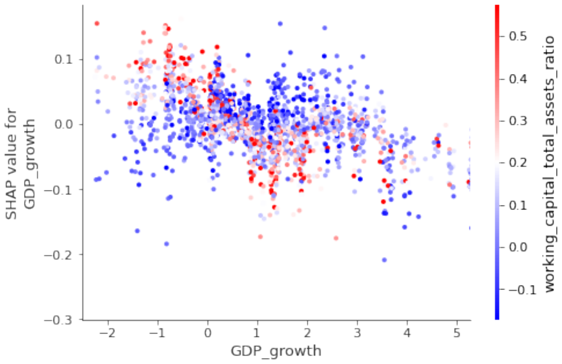


Figure 24: Dependence plot for GDP Growth. Clear interaction effect showing that WCTA yields opposite SHAP impact dependent on whether the economy is experiencing a recession or a boom.

NTNU

Norwegian University of
Science and Technology