Financial Event Prediction using Machine Learning

Derek Snow

Department of Accounting and Finance

University of Auckland

Owen G. Glenn Building

12 Grafton Road, Auckland, 1010

+6422 3922 056

d.snow@auckland.ac.nz

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD) in Finance

The University of Auckland, 2020

Abstract

Financial machine learning (FinML) has in recent years developed into a subdiscipline in its own right with enthusiastic experimenters at its helm. Since investigating the subject five years ago, there has been an almost exponential progress in academic interest. When I first started thinking about machine learning in finance, formal research was sparse, to say the least. As a result, I have had the luxury to pick from a broad range of topics and decided to investigate financial event prediction using machine learning, giving rise to the eponymous title.

FinML research can loosely be divided into four streams. The first concerns asset price prediction where researchers attempt to predict the future value of securities using machine learning methodologies. The second stream involves predicting hard or soft financial events like earnings surprises, corporate defaults, and mergers and acquisitions. The third stream entails the prediction and or estimation of values not directly related to the price of a security, such as future revenue, firm valuation, credit ratings, etc. The fourth and last stream comprises the use of machine learning techniques to solve traditional optimisation problems in finance like optimal execution and the optimal construction of portfolios.

This thesis, in particular, focuses on the use of machine learning in financial event prediction. In the past, finance academics had to be content with mostly linear models that could only ingest a small number of variables of a particular type. Now we can use non-linear models with a larger number of variables and more versatile data types. In this thesis, I show how machine learning can lead to significant improvements in financial event prediction, more specifically, in earnings surprise, bankruptcy and facility closure predictions, all of which have significant financial implications for the businesses and stakeholders alike.

Contents

Tables

Figures

Introduction

Professional forecasters play an integral role in financial markets. They gather, analyse, and transmit information to market participants, who proceed to use the information when making investment decisions (Barber, Lehavy, McNichols, & Trueman, 2001; Stickel, 1995; Womack, 1996). Successful financial predictions are highly consequential; researchers and financial institutions commonly develop algorithms to predict financial events for speculation and risk management purposes. Hedge funds, such as Two Sigma and Renaissance Technologies, are well known for their use of data and statistics to predict financial event outcomes (Baird, 2017).

Research firms also offer products for this purpose. Zacks Research has proprietary prediction tools such as their Earnings Surprise Prediction (ESP) tool that predicts positive earnings surprises by analysing the dynamics of analyst revisions (Mian, 2013). Similarly, Thomson Reuters provides SmartEstimate, a tool that applies cluster analysis to analyst recommendations in order to predict earnings surprises (Stauth, 2013). I posit that machine learning models trained on publicly available data can be used to beat random-choice benchmarks and human agents. Throughout this thesis, I make use of gradient boosting machines (GBMs). A GBM is a prediction model that sequentially builds multiple decision tree models from which the final outcome is predicted using the full ensemble of trees.

In the first chapter, I use an open source GBM model (XGBoost) and compare its performance to a random choice benchmark in predicting earnings surprises[1]. Similar to Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan (2018), who investigate judges' performance against machine learning algorithms, I show how machine learning, and more specifically, GBM[2] models, can be used to predict earnings surprises by taking advantage of analyst biases and mistakes. The results show that the market, comprising of investors and analysts, underappreciates a range of pricing and earnings patterns when making predictions, paving the way for profitable trading opportunities. Inherent in all the prediction tasks is the potential to use this information to construct trading strategies.

The second chapter involves the prediction of corporate bankruptcy and bankruptcy outcomes. This chapter contends that past research's black-and-white view of simply

[1] The random choice benchmark randomly allocates observations to the underlying test distribution. A random choice benchmark is the most appropriate in this scenario, as the general theory is that models are not able to beat analysts in the estimation process (Brown, 1987).
[2] GBM models are currently in use at Uber, Microsoft, Google, and even CERN, among others.

predicting the occurrence of a legal bankruptcy is not sufficient because the economic effect of the outcome is largely determined by the characteristics associated with the bankruptcy, such as whether the firm would survive the bankruptcy proceedings. As a result, this study extends the prediction to include important filing outcomes, such as how long the bankruptcy process will take, whether the firm will successfully emerge after the bankruptcy period, whether the bankruptcy is tortious, and whether it will involve an asset sale. All of this is done by evaluating a wide set of standardised dollar accounting values and ratios before the bankruptcy filing. The overall GBM prediction model predicts bankruptcy with an accuracy above 97%. The subsequent survival of a firm after entering into bankruptcy can be predicted with 70% accuracy and the type of bankruptcy filing (e.g. Chapter 7 or Chapter 11) can be predicted with 95% accuracy.

In the third chapter, I used a Yelp dataset with 430 unique restaurant variables to predict restaurant closures globally. This model, trained on more than 20,000 individual restaurants, has an accuracy just above 96%. Previous work showed that Yelp data can be used to predict the local economic outlook (Bialik, 2017). A related study shows that this type of digital data can also be useful in guiding labour and economic policy (Glaeser, Kim & Luca, 2017). Restaurant ratings have micro-economic implications. Taylor and Aday (2016) show that better ratings of restaurants command increased prices. A recent publication from Harvard Business School showed that a one-star increase in rating can lead to a 5 to 9 percent increase in revenue (Luca, 2016).

Knowledge of the likelihood of future restaurant closures can inform management actions. Knowing which restaurants are likely to close can help management to decide which locations to retain and which locations to abandon. The model can be expanded to predict more years in advance so as to assist management in intervening before the probable closure. Moreover, a deeper understanding of the non-linear relationship between predictive variables can assist management in improving not just struggling, but also well-run restaurants.

In summary, the third chapter demonstrates a promising method to predict restaurant closures. The implications of restaurant closures are likely to be economically consequential to multiple stakeholders. Customer-sourced restaurant data from Yelp and subsequent closure predictions can be used by parent restaurants and private equity firms to decide which restaurants should receive further funding and which restaurants should be closed or be subject to management intervention. Knowledge about potential failure can significantly aid resource allocation strategies and enhance overall firm performance.

A brief introduction to machine learning

Machine learning tasks can largely be divided into data processing, supervised learning, model validation, unsupervised learning, and reinforcement learning themes. In this thesis, special attention has been given to data processing, supervised learning, and model validation steps. In the data processing step, for example, the subtasks can be further divided into feature[3] cleaning, feature generation, and feature preprocessing methods. Continuing with this pattern, feature generation task can further be broken into manual, automated, and semiautomated feature engineering methods. Each component brings something new to the field of finance and I discuss these innovations throughout the thesis.

Machine learning is 'limitless' in the sense that you can tweak it endlessly to achieve some converging performance ceiling. Some of these tweaks include the use of different methods to perform model validation, hyperparameter selection, up-and down-sampling, outlier removal, and data replacement. Features can also be transformed in myriad ways; the dimensions of features can be reduced or inflated; variables can be generated through numerous unsupervised methods and variables can also be combined, added, or removed.

How do we know if any of these adjustments would lead to a better model? Most of the time we can use proxies for potential performance like the Akaike information criterion (AIC), or feature-target correlation. These approaches get us halfway towards a good outcome. Another approach is to re-test the model each time a new adjustment is introduced. This sort of testing should not be performed on the same data that would be used to test the performance of the model, i.e. the holdout set; instead separate validation sets should be specified for this purpose. It is also preferable that after each new test, the validating data is changed or swapped to ensure that these adjustments do not overfit a single validation set; one such approach is known as K-Fold cross-validation where the validation set is randomly partitioned into K equal-sized subsamples for each test.

When working with classification problems one can look at the mean increase in, for example, accuracy or an increase in the ROC (AUC) score[4] to decide whether the additional

[3] In machine learning the term 'feature' is synonymous with the concept of 'variable' in statistics.

[4] ROC (AUC) (receiver operating characteristics area under curve) plots the true positive relative to the false positive rate with respect to all decision probability thresholds (the threshold is a value from 0%-100% used to classify an observation as 1 as opposed to 0). When Type 1 errors (FP) and Type 2 errors (FN) are minimised across all decision thresholds, this value is maximised. The ROC (AUC) score therefore provides an integral based performance measure of the quality of the classifier. A value of 50% is expected for random noisy predictions. Generally, values from 80%-100% are considered as great classifiers. It is arguably the best single

adjustments should be accepted. If it is expensive to retrain a full model one can create smaller models to test. If it is still a problem, one can resort to other metrics like the AIC. It is important to note that the more tests you perform on a validation set, the more you are 'abusing' the data and increasing the likelihood of overfitting, leading to poor out-of-sample performance. Throughout this thesis, I have automated this form of experimenting using open-source Bayesian optimisation techniques.

In this thesis, I apply innovations finance and machine learning to conduct research that is robust from both perspectives. Each model performance section is followed with an extensive predictor analysis section. For feature importance, I identify, among others, Permutation Importance values using ROC (AUC) and SHAP values. The Permutation Importance is the mean decrease in ROC (AUC) after permuting the variable and restraining the model. Another way of defining it is the increase in the prediction error of the model after the feature's values are permuted. SHAP values are game theoretically computed values with local explanations to accurately and consistently estimate variables' overall contribution to the output of the prediction model.

One of the disadvantages of SHAP values is its compute time which is $O(2^n)$ order of magnitude. Generally, one wants to avoid algorithms with running times where *n* is an exponent. The benefit of SHAP values is that they provide both local and global insight, albeit this is at the expense of being slightly less interpretable. The disadvantage of Permutation Importance is that it only provides global insight but on the other side are very factual and intuitive. These measures play complimentary roles in my analysis. When you are investigating global feature importance, use Permutation Importance; and when you are interested in local feature importance and interaction effects use SHAP values.

I further ensure that the models are interpretable by using model-agnostic variable interpretation techniques like Partial Dependence Plots (PDPs), Individual Conditional Expectation (ICEs), Accumulated Local Effects (ALEs), and feature interaction figures and tables. When appropriate, I also investigate the statistical significance of each variable's contribution to the final prediction as well as the significance of prediction models as a whole by introducing noise and retraining the models. Similar to leading financial research, I test the models on various out-of-sample periods, some of which covers large economic events

---

metric machine learning researchers have in measuring the performance of a classifier (Bradley, 1997; Fawcett, 2006; Powers, 2011).

like the GFC (global financial crisis). In addition, I develop a unique walk-forward validation technique to guard against model leakage and overfitting.

This thesis presents a suite of metrics for all the classification tasks. I use three different forms of confusion matrix/contingency tables, namely the standard quadrant table for binary prediction, a random guessing table, and a percentage composition table with proportions, recall, and precision metrics. I further extended the precision analysis by pitting the various classes against each other and against benchmarks. I used a binary or multiclass receiver operating characteristic (ROC) and the associated area under the curve (AUC) as a universal score for classification performance. At points it was also helpful to show accuracy scores, cross entropy scores, and false positive, and false negative rates.

For each prediction task, I follows the same general process: (1) First, I determine the best model for the prediction task; (2) Then, I describe the steps required to develop and structure the data; (3) Thereafter, I perform the prediction task and report the relevant performance metrics; (4) Finally, I perform an extensive predictor analysis to identify the most important variables, after which I discuss the implications. From the problems I consider, gradient boosting machines always came out on top. The first two chapters use the XGBoost implementation and the last chapter uses the LightGBM implementation of gradient boosting machines. These models are known to perform particularly well on structured or tabular data, whereas deep learning models perform well with unstructured data such as audio, images, and text.

A Surprising Thing: The Application of Machine Learning Ensembles and Signal Theory to Predict Earnings Surprises

Abstract

Nonlinear classification models can predict future earnings surprises with a high accuracy by using pricing and earnings input data. Surprises of 15% or more can be predicted with 71% accuracy. These predictions can be used to form profitable trading strategies. Additional variables have been created using signal-processing and handcrafted feature-engineering methods. Some of these variables have in the past been known to be related to analyst bias. The machine learning model in effect corrects for analyst mistakes and biases by incorporating these variables into a nonlinear prediction model to predict future earnings surprises.

I. Introduction and Motivation

Zacks Research and Thomson research both have proprietary tools to predict earnings surprises that deconstruct individual analysts past performance. In this chapter I predict earnings surprises without deconstructing individual analysts' performance and instead focus on improving on the aggregated analyst consensus which is then used as one of the inputs in the surprise model. I allow the additional inputs and a machine learning model to identify and correct the mistakes and biases identified in consensus analyst forecasts. The earnings surprise classification model[5] shows an increase in accuracy compared to a random choice benchmark. The predictions are made with minimal effort by utilising a narrow set of data and incorporating carefully engineered variables as inputs to a gradient boosting machine (GBM) for a wide cross-section of earnings quarters.

Following from this, this chapter seeks to answer three research questions. Can machine learning be used to predict earnings surprises better than a random choice benchmark? If so, is this evidence of analyst bias and analyst mistakes? And, is it possible to form a simple trading strategy based on earnings surprise predictions? To address these questions, I use a classification task to predict the occurrence of earnings surprises and a trading strategy to identify whether the results produced by the classification model are economically significant. I show that past earnings, current analyst forecasts, and differences between the two are the most important variables for predicting future earnings surprises.

The specific task is to predict future earnings surprises for listed US firms using non-linear machine learning models and extensive variable engineering. I perform validation exercises for two model building stages. In the first stage, the model and variables are selected based on a validation set containing 15% of the data. This model and variables are subsequently used in the second modelling stage to dynamically adjust the model hyperparameters for four different periods preceding the test sets[6] to avoid information leakage. The methodology in the second stage consists of chronologically evaluating the performance of the model by keeping the size of the test set fixed while walking-forward and increasing the size of the training data for each successive split. In short, it is a walk-forward expanding-window validation process. Additional information on this methodology can be found on page 27.

[5] Machine learning terminology for predicting a discrete outcome variable.
[6] 1998-2003, 2004-2008, 2009-2012, and 2013-2016

I label three classification surprise buckets (negative, neutral and positive) for three different classification models of earnings that deviate more than 5%, 10%, and 15% from the expected value. Some notable results show that a classification model can predict a 15%+ positive earnings surprise, 53% of the time, while random guessing yields only 24%. Negative surprises of 15%+ can correctly be predicted 40% of the time, while random guessing achieves a meagre 11%. And the neutral class can be predicted 76% of the time compared to the 65% of random guessing. Further robustness checks include the use of different surprise thresholds to design a trading strategy in order to test the economic significance of predicted surprises.

Each day, I form a long (short) portfolio on stocks with a predicted positive (negative) surprise that deviates between -50% and +50% from analyst forecasted earnings. In the process, I identify the optimal surprise deviation parameter for the best trading strategy as tested on a validation set. The results show that the best trading strategies exist between 5% and 20%, with 15% surprise deviation being optimal.[7] I, show that an event-driven trading strategy that takes long positions in stocks with 15% positive surprise predictions, while earning the market rate of return over non-earnings surprise days, produces an alpha of 8% relative to a five-factor asset-pricing model on an out-of-sample test set (Fama & French, 2015). There is no good reason to form a long-short portfolio with negative and positive earnings surprises, as the constituent firms in the portfolio are not comparable and positive and negative surprise predictions do not necessarily fall on the same dates. For further tests see the trading strategy section on page 55.

The models used in this study incorporate readily available inputs in the form of historical earnings, analyst forecasts, and pricing data. I subsequently show that a handful of variables account for the majority of the prediction success. In predicting earnings surprises, I always incorporate analyst forecasts as an input. In chess, at least currently, human-machine collaborators are the most difficult opponents to beat (Kasparov, 2010). Similarly, the inclusion of analysts' estimates in a machine learning model leads to better performance than just relying on analyst estimates or a machine learning model without analyst estimates. Research by Kogan, Levin, Routledge, Sagi, and Smith (2009) showed that using historical volatility was better at predicting future volatility than using textual analysis scans over financial reports, but that when they combined both predictions in a single model, it significantly outperformed a model purely based on historical volatility. An important reason

[7] See the trading strategy section on page 56 for an elaborate explanation of the methodology used.

why human-machine collaborators tend to outperform has to do with the uncorrelated or dissimilar approaches they take to solve the same problem.

Accuracy and out-of-sample performance are two important concepts in building prediction models. The model can be made more generalisable by following certain procedural steps. Accuracy, especially "base" accuracy, is achieved through domain knowledge in choosing and obtaining relevant variables. Accuracy is enhanced by model selection, parameter adjustments, and data improvements. For this study, I have followed standard approaches for promoting generalisability, such as the creation of train-validate-test data splits in time-series to ensure that the model is tested only against unseen holdout-sets. In this chapter, base accuracy is achieved by selecting variables from three data sources, namely prices, analyst forecasts and earnings data. The set of variables is then expanded using feature engineering, which consists of interacting and transforming existing variables to create new variables.

The use of algorithmic models to improve prediction accuracy is only useful if researchers or practitioners can identify and understand domain specific biases. Machine learning techniques allow users to swiftly correct for analyst biases by engineering variables and feeding them into a high dimensional model in the attempt to 'reveal' these systematic biases to the model. Mullainathan and Spiess (2017) refer to three use-cases for machine learning in economics and finance: (1) To utilise new kinds of data; (2) To find new ways to analyse data; (3) To ask new questions. In this chapter I consider new ways to analyse old problems.

This chapter investigates two types of biases. I first identify earnings forecast biases from the literature, i.e., the systematic deviations of actual realizations from forecasts as reflected by past research (*Table A15*). Subsequently, I speculate on possible unobservable biases through the identification of the most relevant variables and variable groupings for beating analyst forecasts. I incorporate variables that appear to relate to forecast bias as additional inputs into the machine learning models. These variables are contained in the earnings, technical and signal processed variable sets, and are identified in the predictor analysis section. Similar to linear coefficients, the majority of the models identify the contributing factor of all variables. This allows us to theorise about relationships and associations in a multi-dimensional domain.

Researchers have yet to successfully incorporate contemporary machine learning methods in earnings prediction research. This study innovates in the field of finance and machine learning by mapping signal processing algorithms over existing time series variables

to carve out deeper patterns for 'machines' to analyse and learn to predict event outcomes. Furthermore, this paper finds a unique use for technical trading indicators to predict changes in earnings, rather than changes in future returns. It is the first machine learning research model to improve analysts' forecasts and predict earnings surprises by developing a model aware of potential biases. It is also the first paper to show that one can earn profits by predicting future earnings surprises. Lastly, this model is dynamic, in that the most important variables tend to change over time, possibly related to the process of analyst learning.

II. Literature

A. Earnings Surprise Prediction

Earnings surprises have been shown to influence stock prices (Graham & Dodd, 1934). The response of the stock price has been demonstrated to be statistically related to earnings announcements (Bartov, Givoly, & Hayn, 2002; Kasznik & McNichols, 2002). Managers also believe that they have to meet or exceed the market's earnings expectation to increase or maintain the share price (Graham, Harvey, & Rajgopal, 2005). From this, we can infer that the successful predictions of earnings surprises can be used to develop profitable trading strategies.

Ball and Brown (1968) provide compelling evidence that there is information content in earnings announcements. Strategies have been developed to take advantage of the difference between forecasts and surprises (Latane & Jones, 1977). The economic effects of surprises are not necessarily immediately noticeable as the market may take some time to reflect the perceived economic impact of the surprise (Bernard & Thomas, 1990). In contrast to the studies mentioned above, the purpose of this paper is not to prove the profitability of a strategy by exploiting the post-earnings announcement drift (PEAD); rather, the focus is on the short-term, same-day price reaction of the stock[8].

Numerous trading strategies can be fathomed that relies on having information on the likelihood of future earnings surprises. Brown, Han, Keon, and Quinn (1996) have developed earnings surprise prediction strategies using multi-factor regressions, focusing on variables such as stock returns, price-to-earnings ratios, book-to-market ratios, and firm size to predict future earnings, finding that past earnings surprises are an important variable driving future

[8] The next trading day is used for after-hours announcements.

surprises. My study is different in that I do not include non-earnings-related fundamental variables, and that I make use of nonlinear prediction methods. A study by Dhar and Chou (2001) makes use of genetic algorithms with moving averages and a comprehensive list of fundamental information as inputs for 12,164 observations from 1986 to 1997. In contrast, I show that better results can be achieved with a larger sample and a narrow set of easily accessible earnings and pricing data while using modern machine learning techniques.

The first part of the study focuses on prediction rather than hypothesis testing. This paper does, however, attempt to explain variable importance and their respective directions. Due to the nonlinear nature of the prediction methods, this analysis is more involved than simply running linear models and identifying their respective coefficients. In traditional finance, we mostly add interactions and combinations of variables manually when using conventional linear models; here the dimensions can be measured by kn. In ML (Machine Learning) the dimension is a result of the function one chooses, such as the number of nodes (ex. n) and the number of alternative choices to each node (ex. k) in a decision tree. For these tree models, k^n is a more accurate representation of the order of dimension. Further transformations of these trees into ensembles/meta-models lead to an even higher dimensional space (Joret et al., 2016). Using an out-of-sample data set allows a researcher to minimise over-fitting and improve generalisability by adjusting the complexity of a model; for econometric regression, this can be done using L1 or L2 regularisation; for a decision tree, this can be done by adjusting the depth of the trees.

There are a few notable points to consider when constructing an earnings surprise trading strategy, such as the possibility of stale earnings per share (EPS) forecasts (Lys & Sohn, 1990), the likelihood of low analyst coverage (Kinney, Burgstahler, & Martin, 2002), analyst forecast dispersion (Freeman & Tse, 1992), I/B/E/S exclusions (Abarbanell & Lehavy, 2002) and earnings preannouncement (Anilowski, Feng, & Skinner, 2007), all of which can have a big effect as to whether the market actually experiences an earnings surprise. If these factors excessively warp the analyst forecasts, then the defined surprise may merely be a surprise on paper and not be perceived as one by the market. Johnson and Zhao (2012) also remark that a large portion of returns are in the opposite direction of the earnings surprise. They do, however, say that the majority of such occurrences are observed in interior deciles. For that reason, this study incorporates various surprise thresholds to identify this effect on profitability.

B. Feature Engineering and Biases

Feature engineering is a machine learning term for the creation of variables from a raw set of data. For example, rolling average price features can be created from a series of historical prices. The first set of engineered features are generated from price data; in this study, I use the opening, closing, high, low, and volume data for each stock. With this data, I create a further 57 variables using a wide range of technical indicators. Many studies have attempted, and have to some extent succeeded, in making use of technical indicators in combination with machine learning models to predict future stock price movement (Kim, 2003; Patel, Shah, Thakkar, & Kotecha, 2015). In the 1960s, trading rules, based on technical indicators, were said not to be profitable (Fama & Blume, 1966). However, these indicators were never used in combination with learning algorithms and, more specifically, never applied for event prediction. In prior literature, no study makes use of technical indicators to predict financial event outcomes as opposed to changes in the stock price.

In this chapter, I apply signal process mappings over all price and volume containing variables to calculate, among others, Langevin fixed points and Fast Fourier transform coefficients. Signal processing algorithms have been used in finance before. Ramsey and Zhang (1997) used waveform dictionaries to decompose signals contained within the foreign exchange market, and a Langevin approach was used to describe stock market fluctuations and crashes (Bouchaud & Cont, 1998). To my knowledge, there is also no academic study that creates variables from such signal processing decompositions on pricing data to predict financial events like earnings announcements. Making use of signal processing techniques allows us to characterise price and return patterns before announcements by mapping hundreds of functions over a time-series of price and technical variables and testing the relevance of each variable on a holdout set. This gives the machine learning model an added advantage in uncovering patterns and associations for an enhanced understanding of the stock price before abnormal events.

The variable engineering part of the paper takes inspiration from control system engineering to try to model and describe the shape of pre-reaction curves before earnings announcements. This study shows that second-order system parameters such as magnitude and peak time, for historical and technical price series, provide unique insights into data that is not necessarily revealed by traditional pricing and technical indicators alone, as technical indicators are constrained by design to facilitate a specific financial use case. To calculate the parameters, a range of algorithms is applied to price and technical time-series for the 30

days leading up to the announcement date. For a list of these algorithms, please see Appendix *Table A14*.

After consulting past research, I include additional variables deemed most useful for counteracting analyst biases. The first is a running forecast error to identify systematic over- and under-prediction (Butler & Lang, 1991; Fried & Givoly, 1982). Another measure is the rolling average percentage difference in forecasts to track the 'stickiness' in the forecast of earnings (Givoly & Lakonishok, 1984). Barron, Harris, and Stanford (2005) provide empirical evidence that private information inferred at the time of an earnings announcement is correlated with greater trading volume; I, therefore, include volume as a model input. Furthermore, the variance of volume before an announcement may also be a promising input (Beaver, 1968). Two additional types of measures include a skewness measure of past earnings from research by Butler and Lang (1991) and a range of dummies to identify whether or not the last earnings were above surprise thresholds of 5, 10 or 15 percent, due to the tendency of earnings surprises to repeat (Brown et al., 1996). Other variables include the number of analysts covering the firm (Lys & Soo, 1995); this measure also serves as a means to quantify the level of public information available for each stock (Das, Levine, & Sivaramakrishnan, 1998). Das et al. (1998) argue that when earnings are less predictable, analysts have a stronger incentive to issue optimistic forecasts; I will include a standard deviation of earnings measure to proxy for this uncertainty. For a full list of these biases and whether they form part of the top ten most important features in subsequent results, see *Table A15* in the Appendix.

C. Machine Learning

Ensemble learning refers to the weighted voting of multiple models (Dietterich, 2000). In this study, the models are constructed from decision trees. There are two traditional ways to execute an ensemble strategy, namely bagging or boosting. This study will use a versatile boosting model very similar to Gradient Boosting Decision Trees defined by Friedman (2001) known as XGBoost (Chen & Guestrin, 2016). Boosting is the process of fitting an initial model to predict a target value after which new models are subsequently fitted on the errors of the previous step to improve the final prediction model. Gradient boosting for classification models takes the additional step to fit the iterative models on the gradient of a log-loss (cross-entropy) function in order to minimise a differentiable function.

In the past, more research has been conducted in the univariate category than in the multivariate category, but that has slowly changed over the years. The use of machine learning in finance is also becoming more common as researchers slowly uncover the nonlinearity of financial data, as has shown to be the case for quarterly earnings per share data (Callen, Kwan, Yip, & Yuan, 1996). Callen et al. (1996) showed that machine learning models have for a long time been able to beat time-series models in forecasting. Xiao, Xiao, Lu, and Wang (2013) demonstrate the power of ensembles in financial market forecasting; they show that the flexibility of the ensemble approach is key to their ability to capture complex nonlinear relationships to predict future stock prices. And finally, Gu, Kelly, Xu (2018) shows how machine learning methods can be used in empirical asset pricing.

As outlined by Kuhn and Johnson (2013), I select the best model by starting with the most flexible and best-performing models as disclosed in past research, and I analyse their performance on a subsection of data to establish a performance ceiling, after which I select the best model. I further compare the performance of different types of Neural Networks, Support Vector Machines, Random Forest, Naïve Bayes, Adaptive Boosting, and Extreme Gradient Boosting Decision Trees models. In traditional finance research the acceptability of empirical results generally hangs on the requirement for interpretable causality, in this study, accuracy, associations, profitable trading strategies, and measures of variable importance trump the interpretability of the model. The study does not attempt to demonstrate causal relationships but rather the relevance of previously unstudied variables and the performance of machine learning to predict earnings surprise outcomes.

Applied machine learning has gradually made its way into finance. One of the first quality papers came from Teixeira and de Oliveira (2010) who used economic and financial theory in combination with fundamental, technical, and time-series analysis to predict price behaviour using artificial neural networks. Other researchers like Bagheri, Peyhani, and Akbari (2014) used an adaptive networked-based fuzzy inference system to forecast financial time-series for currencies while Hu, Feng, Zhang, Ngai, and Liu (2015) used a hybrid evolutionary trend following algorithms to introduce a trading algorithm that selected stocks based on different indicators. The majority of research in this field has focused on the price movements of stocks, indices, and currencies. These studies are limited in sample size, most opting to analyse a small number of stocks, making it difficult to rule out random results. Very few studies look at financial outcomes; additionally, no study has used modern machine learning techniques to investigate the probability of financial event outcomes in order to apply it in a trading strategy. The study as presented in this paper serves as the foundation for

a novel system that aids investors and market makers in managing their stock ownership before earnings announcements, not just for profit maximisation but also for risk management purposes.

III. Data

All available quarterly earnings per share measures were obtained from the Thomson Reuters' I/B/E/S Detailed File for publicly traded firms in the US, starting from the first available date in 1983 to 2016. I make use of the detailed file due to known rounding errors in the I/B/E/S summary file (Payne & Thomas, 2003). As per Claus and Thomas (2001), I dropped the quarters before 1984 from the I/B/E/S database; before this date the database provided too few firms with complete data to represent the overall market. The starting number of observations are 455,142 firm quarters. Daily stock information was sourced from CRSP's Daily stock file. I exclude observations flagged as "Excluded" or "Stopped Coverage" by IBES. Analyst ratings had to be published two months before the actual announcement, following Behn, Choi, and Kang (2008). This establishes a fair method to compare the machine learning model's performance against timely analyst forecasts. This resulted in 313,416 firm quarters.

I also include only the most recent forecast of each analyst for every brokerage house for each earnings period; this further decreased the number of firm quarters used in the preceding calculation to 175,176. Other data cleaning operations include removing all analyst forecasts that appear after the announcement date and entries where the ticker or forecasted value is null, removing about 3% of the observations. There is no explanation other than the entries being in error. A further requirement is for the firm to have 6 years of prior financial information to create rolling earnings-related variables, leaving 158,224 quarters for the main prediction algorithm. The amount of trailing data is similar to O'Brien (1998) and Kross, Ro, and Schroeder (1990), who have used around seven years of trailing data for their time-series models. Cutting away 6 years leaves us with 26 years of data, a period from 1990-2016, to perform the machine learning operations on.

In this study, 15% of the data is used for model validation, hyper-parameter tuning, and feature selection process after which I disregard the data to uphold prediction integrity, leaving 134,490 observations.[9] This validation process is used to select the model, reveal the

[9] A smaller size of 15% of the data is used for validation because with this particular validation method the data ultimately gets dropped and decreases the size of future training, validation and test data.

starting hyperparameters, and select the features. It is unrelated to the validation process used to develop the models that report the final metrics using a chronological, walk-forward, and expanding-window validation method to dynamically adjust the hyperparameters and test the performance of the model throughout time. Additional details are provided in the next section.

IV. Methods

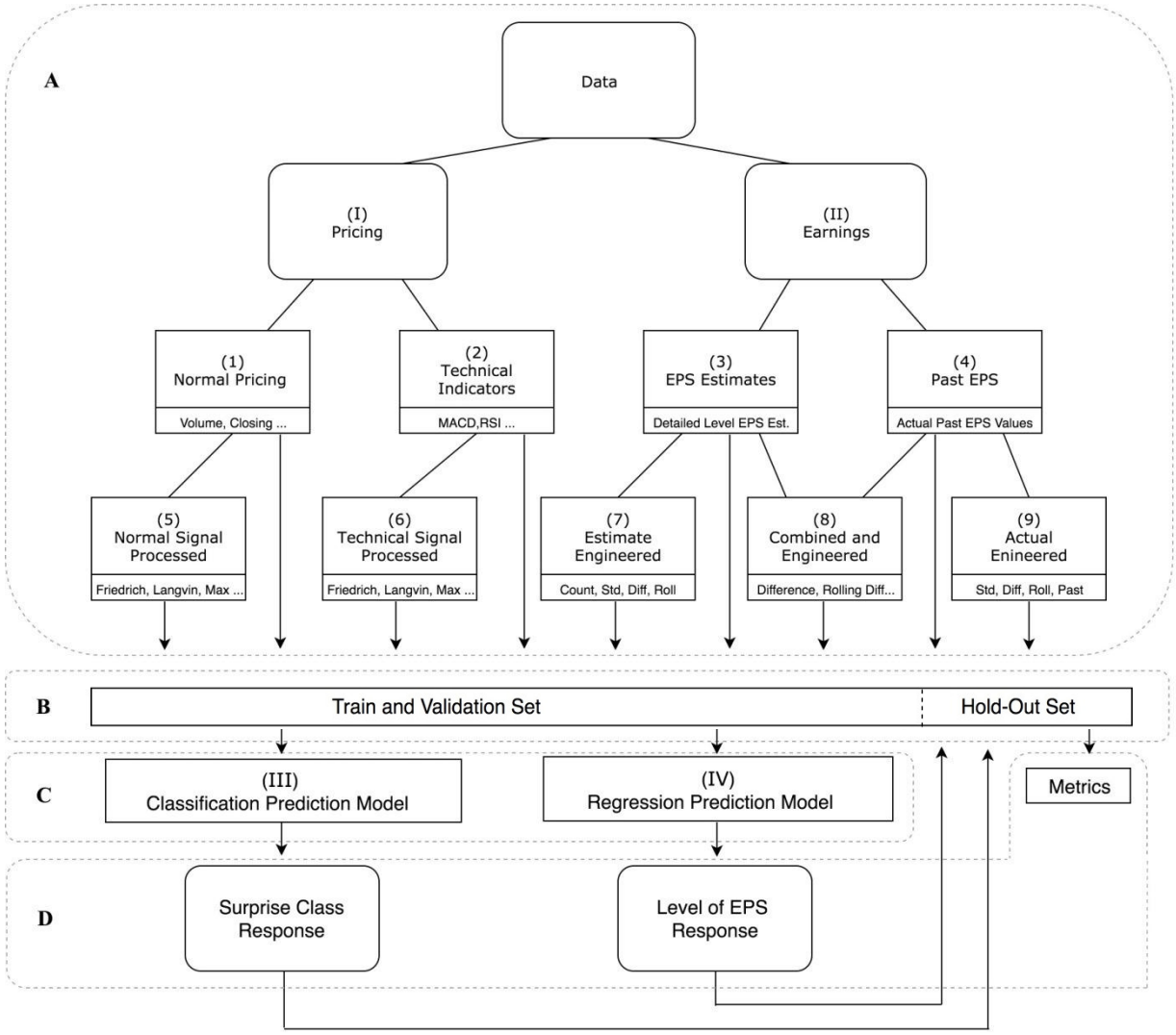
This section develops the methodology of the machine learning classification task. Readers who are interested in the results of this study rather than the methods used should skip to Classification *Section V* or straight to the table on page 35. The empirical part of the study consists of a variable creation stage after which a training dataset is used to train the prediction algorithm. Each quarterly surprise classification is described according to a set of variables and a response value. The trained model is used to make predictions against a test set to, among others, assess its accuracy. This study goes beyond simple machine learning; it does not just report on how well the prediction fits the test set but also sheds light on what the predictions tell us about the quality of analysts' predictions, and whether the algorithmic predictions can help to uncover analyst biases. The first stage will follow the standard machine learning process; the second part will focus on what these predictions tell us about analysts, and how these predictions can be used in practice, such as in establishing trading strategies and improving forecasts. *Figure 1* is a critical diagram revealing the process of obtaining data, creating variables, and training and testing a model. The headings and associated labelling in this section correspond to that of the Figure.

A. Variables

In this study, the black-box prediction model incorporates information leading up to the day before a firm's official earnings announcement date to make a prediction of the forthcoming earnings event. To make these predictions, variables are constructed from a wide range of datasets by applying multiple transformations over time-series data so as to convert them to a series of cross-sectional values. The majority of these transformations are derived from signal processing literature. The final set of single value inputs can then be fed into the machine learning model. *Figure 1* shows the type of variables used in the model. There are

multiple input variables and the relations between these variables play an integral part in the models' prediction success.

Figure 1: Process Tree



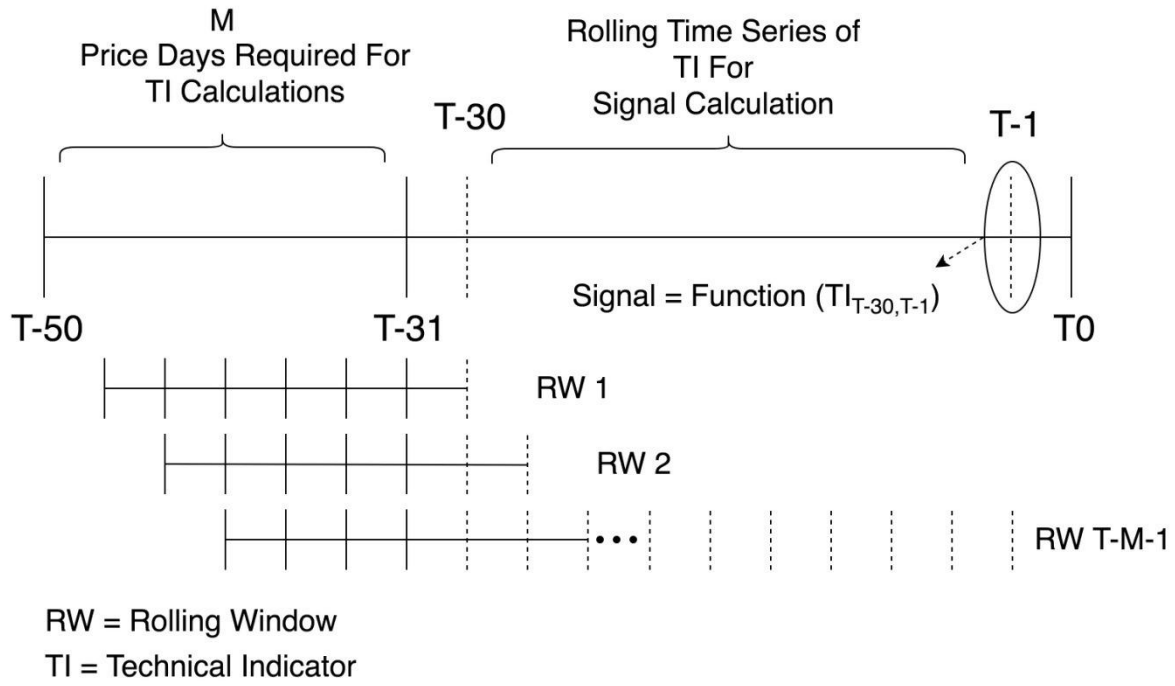
(A) Items (1) - (9) constitute the final groups of variables that would be used in this study. All of these variables are derived from pricing and earnings-related datasets. The earnings-related data can further be deconstructed into (3) EPS forecasts and (4) Past EPS values and hand engineered variables (7) - (9). (1) In total, there are no more than 5 normal pricing variables, (2) 57 technical indicators, (3) 1 EPS forecast, (4) 1 EPS value, (5) 20-50 normal signal processed measures, depending on the task, (6) more than 500 technical signal processed measures, (7) 22 analyst-only variables, (8) 22 combination variables, (9) and 23 actual-only variables. (B) identifies a generic train-validation-test split. (C) identifies the black-box prediction models. In this chapter we will only look at classification models. (D) is the response variables and calculation of performance metrics on a hold-out set classification metrics.

1. Pricing Variables

Of the average 62 trading days between announcements, (1) 50 trailing days of pricing data are needed to create a standard set of technical variables. (2) Of these 50 days, 20

trailing days are "consumed" by a technical indicator transformations process. In (5) - (6), the last 30 days of the bundle of pricing and technical variables transform down to a single number by undergoing a range of signal processing transformations. Such a deep analysis will likely pick up on insider trading as well as other pre-patterns associated with earnings surprises. *Figure 2* illustrates the process from (1) to (6). The appendix item *Method A 1* describes this process in greater depth and includes an explanation of the variable selection process (the process of removing variables which seem irrelevant for modelling). Also, for a comprehensive list of the signal processes mappers, see *Table A14*.

Figure 2: Transforming Price to Technical and Technical to Signal



In this study, I use 50 pricing days to calculate the final signal. The first step is to calculate a time-series of the technical indicator. Depending on the indicator, the function can incorporate between 3 to 20 past pricing days. The resulting time-series covers 30 days for all technical indicators in the study. The calculations do not include the price at *T* as this information is not available to us at the time of prediction, *T-1*. The next trading day is time *T*, for days where the earnings announcement occurs after-hours. The time-series of technical indicators gets fed into a signal processing function, which calculates a singular value from the time-series. Following is an example to better understand what variables get included in the learning and prediction algorithm. The model will include as inputs the closing price at *T-1*, the last rolling value of the technical indicator at *T-1*, and the singular signal processed value as calculated at *T-1*. The above figure only describes the process for technical indicators to signal values, but normal pricing data also gets signal processed as shown in *Figure 1* (5).

2. Earnings Variables

The earnings variables are constructed from past earnings per share (EPS) forecasts and actual EPS values. The machine learning phrase for the process is "feature engineering." For the variables (7) - (9), the calculation involves multiple timeframes of rolling averages, weighted averages, lagged values, past differences, the standard deviation of forecasts and a count of the number of analysts per forecast. A lot of the inspiration for the above variables' calculation selection was drawn from analyst biases and mistakes as noted in past literature. A table identifying some of these variables can be found in the Appendix, *Table A15.*

3. Response Variable

For the classification task[10] (*Figure 1* (C)) the response variable for the machine learning model is the occurrence of an earnings surprise. An earnings surprise is simply defined as a percentage deviation from the analyst's EPS expectation, as described in the data section, and the actual EPS as reported by the firm. In this study, we include percentage thresholds, *S,* as a means of expressing the magnitude of a surprise so as to construct various tests.

$$X = \frac{EPSAC_{it} - EPSAN_{it}}{EPSAN_{it}} - 1$$

$$SURP_{itsx} = 0, \ where \quad X < -S, \ \text{Negative}$$

$$SURP_{itsx} = 1, \ where - S \leq X \leq S, \ \text{Neutral}$$

$$SURP_{itsx} = 2, \ where \quad X > S, \ \text{Positive}$$

i = Firms in the sample

t = Time of the quarterly earnings announcement

S = A constant surprise threshold, 5%, 10%, or 15%

x = A constant percentage of samples sorted by date of earnings announcement

[10] The task predicting a binary dependent variable.

$EPSAN$ = Mean analyst EPS forecast

$EPSAC$ = The actual EPS measure as reported by the firm

This surprise measure is simply the difference between the actual and expected EPS scaled by the expected EPS. This measure is similar to Foster, Olsen, & Shevlin (1984), the difference being that they used the absolute actual earnings as the denominator instead of expected actual earnings. I have tested three other variants where the actual EPS is the denominator and is absolute in value; this led to a small but non-significant improvement (Foster, Olsen, & Shevlin, 1984). I also looked at the level of earnings using the same formula, which produced the same results as EPS, and finally a standardized unexpected earnings (SUE) measure that also led to a small but non-significant improvement in returns (Brown et al., 1996). I found that all these measures are highly correlated from 85% upwards so I only report the most obvious solution. What is important is whether the prediction of surprises as defined leads to positive abnormal returns. If it does, then one is said to have a definition of earnings surprise similar to that of the market. It is also worth noting that I set three thresholds and that these thresholds mostly eliminate issues regarding highly positive or highly negative values when it is close to zero.

B. Train, Validation and Test Sets

The model building in this chapter occurs in two stages. The first stage lays the initial groundwork by choosing the model type and variables that would be used in all future model iterations. In this first stage, various types of models with multiple sets of hyperparameters are compared to each other and the best model is used to perform feature selection. The second stage uses the first model to dynamically adjust the hyperparameters over time. In the second stage, no additional model selection and variable selection procedures are performed; the models in this second iteration are used to report the prediction results on the test data. The methodology of the second stage are presented in *Figure 3;* it consists of chronologically evaluating the performance of the model by gradually increasing the window size while keeping the size of the test data fixed.

To clarify quantitatively, in the first stage, the data is sorted by time, and the first 20% is used for training while a random selection of 15% of the remaining data is used for validation. The first modelling stage uses a fixed but chronological train-validation split to perform model, hyperparameter, and variable selection, after which the validation data are

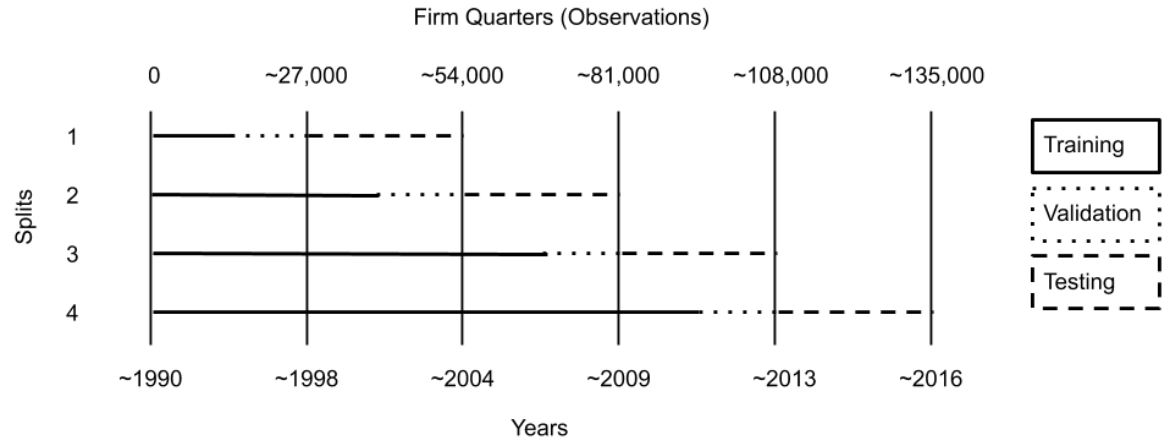
dropped indefinitely. This percentage of data used for validation is kept small because this particular data cannot be used for future training, validation, and testing; the data ultimately gets dropped after use.

This is the safest possible approach to build an applied machine learning model in time-series. The model type and variables selected in stage one remains constant in stage two while only the internal model parameters and hyperparameters are allowed to adjust freely in each of the five successive time-splits. As more data becomes available, the training set increases allowing for improved machine learning prediction. Although the test set stays constant in size, it shifts forward to test distinct non-overlapping periods. The testing data never contains data that is younger than the training data, which is a sensible step for preserving prediction integrity; the validation data never gets recycled as testing data. This particular walk-forward model is also helpful in that the model can easily be tested for robustness over time without a large gap in the data that would normally result from alternative validation procedures in time-series.

The first model in the second development stage starts out with the same 20% of the original time-ordered data, but this time it is used for both training and validation purpose. The data is chronologically split into five chunks each containing 20% of the data (~27,000 observations). The size of the validation set for each iteration is 10% of the original size of the data. For each iteration, the test data is fixed at ~27,000 observations and the validation data is fixed at ~13,500 while the training data increases with each step forward. The size of the validation data has been set based on the objective to have the most up to date hyperparameters for the test data. The validation data is not dropped in this model development stage but ingested by a second model that is trained on both the initial training and validation data. This is done to ensure that the timeliest training data is included in the model to more accurately adjust internal model parameters. This final model is then applied against a test set.

Once the model has been selected and specifications made, it is necessary to measure the out-of-sample performance of the model. This method is well suited for time-series evaluation where there is an expectation of structural changes in pattern change over time. This technique can be used for both classification and regression tasks (Bergmeir & Bentez, 2012). Although the test set stays constant in size, it shifts forward to evaluate distinct non-overlapping periods. To calculate the final result, I compute an average value across all four equally sized test sets and calculate a confidence interval.

Figure 3: Expanding Window Train-Validation-Test Splits



To ensure consistent performance measurements on the test splits, they should be the same size. In this study, the data is split into five equal-sized sections. And the model is trained on four of the five sections and tested on four of the five sections. For each additional section, the model trains on an increasing number of samples ordered by date. This study reports both the overall accuracy and breaks the accuracy down for each period and surprise threshold in question. This table does not show a separate process used to do feature and hyperparameter selection (the process of removing variables that seem irrelevant for modelling), which appears in the first model building stage. The feature selection is done on a small validation set constituting 15% of the data to ensure that during the development stage there is no 'double dipping' into the data; therefore, the model always gets tested on a fresh out-of-sample dataset.

C. Machine Learning

1. Black Box Understanding

The prediction algorithm used in this study can be viewed as a 'black box.' Below is high-level pseudo code to provide a better understanding of some of the core concepts of the black-box model such as its relationship with the training set, test set, validation set, predicted values, and metrics.

(1) $TrainedModel = ModelChoice(TrainInputs, TrainTarget, Valid, Param)$
(2) $Predictions = TrainedModel(TestInputs)$
(3) $Metrics = Functions(TestTarget, Predictions)$

$ModelChoice$ is the model used to approximate a function that closely resembles the target (response) function. I use an XGBoost model (Extreme Gradient Boosting Decision Tree) developed by Chen & Guestrin (2016). $TrainInput$s are the inputs of the training data, meaning all the data except the target variable. $TrainTarget$ are the target values we want to

train the model on, also known as the response variable in social sciences. $Valid$ is the validation data that will be used to adjust the hyperparameters. $Param$ is a list of 23 hyperparameters that can be tweaked to improve the model's performance. The parameters mostly relate to adjustments in the complexity of the model, these values are not learned by the model, but enough techniques exist so that external models can approximate the best parameter values for the prediction problem at hand. $Prediction$ is the predicted target value obtained by running the inputs of the test set through a trained machine learning model. $TestTarget$ is the actual target variable that the model tries to predict. $Functions$ are the numerous metrics that can be applied to evaluate the success of matching the target variables with the predictors on the test set. For the classification task, the main metrics used are accuracy and ROC (AUC) scores. What follows is an example of the pseudocode for the classification task.

(1) $Classifier = XGBoostTreeClassifier\big(Train_X, SURP_{its(0\,:\,x)}, Valid, Param\big)$

(2) $PredSURP_{its} = Classifier(Test_X)$

(3) $Metrics = Functions\big(SURP_{its(x\,:\,1)}, PredSURP_{its}\big)$

The prediction values, $PredSURP_{its}$, of the classifier are a categorical variable that falls within the values $\{0,1,2\} \rightarrow \{\text{Negative Surprise}, \text{No Surprise}, \text{Positive Surprise}\}$, for different surprise thresholds s $\{5\%, 10\%, 15\%\}$ of firms *i* at time *t*. If we assume that the training set is 60% of the original data set, then the training set's target value is $SURP_{its\ (0\%\,:\,60\%)}$, being the first 60% of the dataset ordered by date.[11] A section of 15 percentage points in the 60% of training data are used as validation data ($Valid$) to select hyperparameters ($Param$). The test set's target values are $SURP_{its\ (60\%\,:\,100\%)}$, i.e., the last 40% of the dataset. The metrics for a classification task comprises of accuracy (proportion of correctly classified observations), precision (positive predictive value), recall (true positive rate), as well as confusion matrices/contingency tables.

As a result of defining the surprise thresholds and discretising it into three buckets, there is a possibility that some information is lost. Instead of developing a multi-classification

[11] In this example the validation data is included in $Train_X$ because after the validation is performed and parameters selected on a section of the training data, a new model is formed to include the validation data as training data to get up to data information for future predictions; similar to the second stage of model development in this chapter.

model, one can predict the level of earnings using a machine learning regression model[12] and then convert the predicted dollar earnings into a predicted surprise percentage after which you allocate it to the respective surprise buckets. It could be that a method that predicts the level of earnings and discretise post-model is more accurate than a method that discretises before the model is trained, but this has to be empirically tested. Another alternative is to perform ordinal classification as opposed to multiclass classification as the surprise categories are ordinal in nature. All of these methods have been tested on validation data, and the multiclass classification model performed the best. I suspect it is due to the minimisation of noisy predictions as a result of using a small number of categories and also due to distinct nature and distribution of the defined earnings surprise outcomes.

2. Model of Choice

Machine learning is defined as the study of inductive algorithms that 'learn' (Provost & Kohavi, 1998). For the purpose of this study, it is valuable to have an intuitive grasp of the XGBoost machine learning model. XGBoost is short for Extreme Gradient Boosting. It is a nonlinear inductive algorithm used to approximate the function between inputs and outputs. The idea behind Gradient Boosting is to 'boost' many weak learners or predictive models so as to create a stronger overall model. The training process iteratively adds additional trees to reduce the errors of prior trees that are then combined with previous trees to produce the final prediction.

To create the overall ensemble model, such as the *Classifier* model described in the pseudocode above, we have to establish a loss function, L to minimise in order to optimise the structure and performance of the model. This function has to be differentiable as we want to perform a process of steepest descent, which is an iterative process of attempting to reach the global minimum of a loss function by going down the slope until there is no more room to move closer to the minimum. We, therefore, solve the optimisation by minimizing a loss function, $f(x)$, numerically via the process of steepest descent. For our classification task, we use logistic regression to obtain probabilistic outputs of the target (response) classes. In normal gradient descent one updates model parameters like the coefficients in a linear regression or the weights in a neural net, however, gradient descent in XGBoost essentially "updates" the model by adding new trees instead of updating coefficients or weights.

[12] Machine learning terminology for a model predicting a continuous value.

Furthermore, it is necessary to minimise the loss over all the points in the sample, (x_i, y_i). For a more detailed description of this process and other more involved formulae see the appendix *Method A 2.* The minimisation is done in a few phases. The first process starts with adding the first and then successive trees. Adding a tree emulates adding a gradient-based correction. Making use of trees ensures that the generation of the gradient expression is successful, as we need the gradient for an unseen test point at each iteration, as part of the calculation $f(x)$. Finally, this process will return $f(x)$ with weighted parameters. The detailed design of the predictor, $f(x)$, is outside the purpose of the study, but again for more extensive computational workings see the appendix *Method A 3.* At this point of the study, all the steps in *Figure 1* from (A) - (C) have been dealt with; the next step (D) evolves the evaluation of performance metrics, detailed prediction analyses, and the description of prospective trading strategies.

V. Classification

A classification task involves a classifier that assigns an instance (observation) to every class (category) based on the learned patterns of a training set. The training consists of past observations in which the classes are known. The model, therefore, learns class associations from the past patterns of explanatory variables, commonly called features, and maps this input data into a class according to a newly learned, weighted, and approximated function. The XGBoost classifier used in this study is a probabilistic classifier which simply outputs a probability of an instance belonging to one of the specified classes.

A probabilistic classifier is especially useful because the magnitude of the probability can itself be seen as a confidence value associated with the class choice. For example, if the probability of both a positive and negative earnings surprise is high for a single instance (earnings quarter), you may not want to follow through with a trade on that particular stock, but if the difference in probability is positive, you can be more at ease. The class probabilities can be used as parameters in a trading strategy.

A positive and negative surprise is defined as an occurrence in which actual observed earnings deviate 5%, 10% or 15% more than analysts' consensus forecasts. In this study, the focus is on a multiclass classification problem, which includes the above-mentioned surprise classes as well as a neutral class that lives between the positive and negative surprise thresholds. Past researchers such as Johnson and Zhao (2012) have noted that reaction in the share price can be in the opposite direction, especially in the inner deciles of surprises,

making it necessary for us to investigate surprises that are not just positive or negative, but positive and negative with more than, for example, a 5% deviation. The surprise threshold can also be viewed as a parameter that can be adjusted to suit the prediction task at hand. For a certain type of trading strategy, you might prefer a 10% or 15% surprise as it minimises false positives.

I present the classification results in four main ways. I first present an accuracy measure and make use of an inductive technique to identify the importance of groups of variables that explain the model success. Thereafter, I move into alternative metrics such as precision and recall by means of illustrative confusion-matrices/contingency tables. For the last-mentioned steps, I produce benchmark scores based on random choice to easily compare against the accuracy and precision metrics of the model. After this first form of analysis, under the "prediction analysis" section, I present the important predictors of the model in tabled and in graph form so that the reader can appreciate the vast range of predictors and the nonlinearity of the model. The last method of analysis involves testing different periods of the sample period and different thresholds of surprise. I finally end this section with a trading strategy to show the application value of the model and to confirm that an 'earnings surprise,' as defined in this model, is not just a surprise on paper, but also a surprise to the overall market.

A. Evaluation

The accuracy can be defined as the percentage of correctly classified instances (observations) by the model. It is the number of correctly predicted surprises (true positives) and correctly predicted non-surprises (true negatives) in proportion to all predicted values. It incorporates all the classes into its measure $(TP + TN)/(TP + TN + FP + FN)$, where *TP*, *FN*, *FP* and *TN* are the respective true positives, false negatives, false positives and true negatives values for all classes. The measure can otherwise be represented as follows:

$$acc(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \qquad (1)$$

Table 1 represents the accuracy measures for three surprise thresholds. The percentage surprise represents the threshold of the surprise. The overall accuracy of selecting

the correct class out of three classes is 63%; this number is reported in the last row of the 4th column of the table. By always selecting the highest populated class, a benchmark accuracy of just below 50% can be achieved. This can only be done if the benchmark is made 'knowledgeable' about the underlying distribution. The model leads to a 15-percentage point improvement over this benchmark. This benchmark is a reported figure in *Table 1* (5). In past literature, this benchmark normally 'peeks' at the distribution of the *training* data to make its predictions accordingly; however, for added robustness and to create a higher-level watermark for the forthcoming trading strategy, I allow the benchmark to peek at the class distribution of the *testing* data. It would use this information to select for the percentage of the most occurring class to calculate an accuracy benchmark.

After *Table 4,* I further deconstruct benchmarks, and show, by way of example, the difference between a benchmark focused on accuracy and a benchmark focused on precision. The above-reported aggregate scores can be broken down to calculate the accuracy of predicting surprises of certain predefined magnitudes, for example, 5% or more. This number incorporates both sides of the deviations - i.e., a 5% or more upwards deviation, being a positive surprise, and a 5% or more downwards deviation, a negative surprise. The confidence interval reports the 95% confidence interval derived from four different train-validate-test splits calculated for each surprise threshold.

In *Table 1,* I have included the accuracy results as more groups of variables are introduced to the model, (1) - (4) in the table. These variable groups are presented in *Figure 1*. This process is done to identify the additional accuracy each group of variables deliver to the final outcome. From left to right, the model includes actual earnings variables, analyst variables, basic pricing and technical variables, and lastly, all signal-processed variables whilst calculating accuracy metrics at each step. This method is similar to an inductive theory testing method promoted by Mullainathan and Spiess (2017). The difference is they started with a completed model and worked their way backwards.

The approach is noteworthy as it goes beyond the task of simple prediction by also decomposing the importance of the type of variables in the study, providing added value to the literature. The results show that price and price-derived variables (3 and 4 in table) contribute to 42%[13] of the improvement over the benchmark; past realised (1) earnings variables contribute to about 19% of the improvement; (2) and the analysts and interaction variables with analysts and actual reported earnings variables provide for the further 39% of

[13] $(0.032 + 0.031)/0.150 \approx 42\%$

the improvement. *Table 1* shows, empirically, the importance of the different groups of variables. The table also demonstrates that the accuracy increases as ever higher surprise threshold are tested. This is likely due to a bigger 'neutral' surprise class becoming a more obvious prediction but could also be due to the identification of more distinct characteristics and patterns associated with firm earnings quarters at larger levels of surprises.

Table 1: Accumulative Accuracy Comparison Table - Surprise & Non-surprise

Surprise	(1) Act.	(2) Frc.	(3) Price.	(4) Signal		(5) Bench		(6) Improved	(7) Confidence
5%	0.433	0.508	0.531	0.550	-	0.398	=	0.152	+/- 0.018
10%	0.490	0.566	0.601	0.633	-	0.472	=	0.162	+/- 0.024
15%	0.603	0.630	0.667	0.711	-	0.573	=	0.137	+/- 0.033
Average	0.509	0.568	0.600	*0.631*	-	*0.481*	=	*0.150*	*+/- 0.025*
(8) Accu.	0.028	+ 0.059	+ 0.032	+ 0.031			=	0.150	

This table compares the various surprise thresholds' accumulative accuracy for different variable sets. (1) are the contribution of actual earnings-related variables. (2) the analyst forecast variables and analyst interaction variables with historical earnings. (3) the original price and technical indicator variables. (4) includes the signal processed variables over price variables and is also the final model. (5) is the benchmark as the most frequently occurring class. (6) is the average percentage point improvement of the full model over the benchmark for all surprise thresholds. (7) is the confidence at 95% level. (8) The accumulative improvement starts by deducting the benchmark average (0.481) from the model earnings related variables (0.509), i.e., 0.028; and each successive improvement is the additional accuracy contribution from adding more data (0.568-0.509; 0.600-0.568; 0.631-0.600). This is only one of many methods to measure variable importance, other methods simply permute the features of interest and retrain the model. The problem with permutation methods is that it does not give accurate importance scores with collinear data, whereas the problem with this method is that the order of introduction matters because of interaction effects.

The accuracy score is not always very informative in an imbalanced classification study, as it does not look at class breakdown precision, nor does it provide evidence of true positive or true negative values. Accuracy measures work better for balanced than imbalanced datasets; the problem is that the 'neutral' class comprises the bulk of the data — hence the relatively high benchmark value obtained above by simply predicting the same category. The issue is that the accuracy for the neutral class is not useful for a trading strategy. In finance, especially when constructing trading strategies, you would generally prefer improvements in precision, True Positives / (True Positive + False Positives), as opposed to accuracy. Similarly, in most strategies you prefer precision above recall, True Positive / (True Positive + False Negative). A low recall means lost opportunities, but as long as there is a diversified portfolio or enough observations, a low recall is not a problem. A

trader would rather be concerned with the precision metric, which is the category accuracy of the predictions made.

The next section explores various confusion matrices/contingency tables. Each category has its own recall and precision fraction. All confusion matrix tables are formed by running models over four serial timeframes for 3 different surprise thresholds, being 5%, 10%, and 15%. For each model, 26,895 class predictions get tested against the true classes. This equates to 322,740 predictions in total. This should not be confused with the original sample size, as these are aggregated across three different surprise thresholds.

Table 2: Aggregated Surprise vs Non-Surprise Confusion Matrix

Confusion Matrix		**Predicted**		Sample Distribution
		Non-surprise	Surprise	
Actual	Non-surprise	**144,417**	54,534	0.62
	Surprise	64,428	**59,361**	0.38
Precision		0.69	0.52	
Improvement		0.08	0.14	322,740

This confusion matrix creates a summary by aggregating all the surprises, positive or negative, together, and separately all the non-surprises. Non-surprises consist of neutral or wrongfully predicted positive or negative surprises. The purpose of this matrix is to gain an understanding of the model's overall performance without having to discriminate in the direction of the surprise (+/-) or the threshold of the surprise (5%, 10%, 15%) or the period over which the test was done (4 splits). The *sample distribution* is equal to all the true observations of a certain classification divided by all the observations; an example along the first row: (144+54)/(144+54+64+59) ≈ 62%. The *precision* is calculated by dividing the true positives (Surprises) with the sum of itself and the false negatives (Not non-surprises). An example along the first column: $144/(144+64) = 69\%$

Table 2 shows a breakdown of predicted and actual classes for observations of surprises and non-surprises. Surprises incorporate both negative and positive surprises. This table, by means of the precision measure, digs deeper than the overall accuracy measure that has been reported in *Table 1*. "Improvement" in the above results show preliminary evidence that the model is better at predicting surprises than non-surprises; improvement is calculated by deducting the underlying sample distribution from the precision scores. This again highlights the potential of a trading opportunity. This improvement can also be expressed by drawing up a confusion matrix from random guessing. *Table 3* shows that for both of the classification groups, the model performs better than random guessing based on the underlying distribution. The purpose of *Table 3* is to highlight the difference in performance compared to model's in *Table 2*. It clearly shows the true positives decreasing and the false positives increasing. True positives went from $144{,}417 + 59{,}361\,(203{,}778)$ to

122,642 + 47,480 (170,122). False positives went from 54,534 + 64,428 (118,962) to 76,309 + 76,309 (152,618).

Table 3: Random Guessing Aggregate Confusion Matrix

Random Confusion Matrix		**Random Guess**		Marginal Sum of Actual Values
		Non-surprise	Surprise	
Actual	Non-surprise	**122,642**	76,309	198951
	Surprise	76,309	**47,480**	123789
Marginal Sum of Predictions		198951	123789	322740

This table is formed by 'randomly choosing the observations' by the allocation of observations according to the underlying test distribution, as presented by Sample Proportion in *Table 2*. A random choice benchmark is the most appropriate in this scenario, as the general theory is that models are not able to beat analysts in the estimation process (Brown, 1987).

The original confusion matrix can further be expanded so that we zoom into the type of surprise. In *Table 4*, I will deconstruct the results of *Table 2* into categories and directions of surprises. The next three tables look at, neutral, i.e., non-surprises, negative and positive surprises. I again produce a breakdown of random guessing. By comparing the first two tables, it can be seen that even without controlling for the recall, as identified by the marginal sum of predictions, the model outperforms random guessing in each category. In a later table, I show more definitely how the model outperforms the benchmark.

By using *Table 4* and *Table 5*, we can get a good indication of what is meant by naïve accuracy and precision benchmarks. A rational, uninformed person, trying to establish the best accuracy measure, would consistently predict the most frequent occurring class. In this scenario, they would consistently predict the neutral class and achieve an overall accuracy of 52% (Marginal Sum Neutral/All Observations, i.e., 169/322), close to the average "Bench" accuracy (5) of 48% in *Table 1*[14]. The problem with the benchmark model is that the precision would be very low at 52% and recall high at 100%. This shows the importance of deconstructing the accuracy measure to identify the precision across classes. If a person was to create a benchmark to enhance precision for all classes, the best strategy would be to use the distribution of the *training* data when allocating observations. In this study, I have created a more robust precision benchmark, in that I allow the benchmark to 'peek' at the underlying *test* distribution. Therefore, a person with this knowledge will randomly allocate 169500

[14] The "Bench" accuracy is slightly different because it was equal weighted across three groups and not value weighted as is the case for the confusion matrices.

observations to Neutral, 46821 observations to Negative and 106419 to Positive. This is indeed what *Table 5* illustrates; this can be appreciated by appreciating that the actual marginal sum of values matches the marginal sum of predictions. The reported accuracy of this multi-class prediction is the sum of the bolded figures divided by the total sum of observations. For *Table 4* this is 63% and for *Table 5* it is 41%.

Table 4: Surprise Breakdown Confusion Matrix

Confusion Matrix		**Predicted**			Marginal Sum of Actual Values
		Neutral	Negative	Positive	
Actual	Neutral	**144417**	4134	20949	169500
	Negative	16860	**7878**	22083	46821
	Positive	47568	7368	**51483**	106419
Marginal Sum of Predictions		208845	19380	94515	322740

This table expands on *Table 2* by splitting the prediction in three distinct groups. It is clear that the model produces many more positive than negative surprise predictions. To see whether the low number of predictions is warranted, we can look at the recall score in *Table 6*.

Table 5: Surprise Breakdown Random Guessing Confusion Matrix

Random Confusion Matrix		**Random Guessing**			Marginal Sum of Actual Values
		Neutral	Negative	Positive	
Actual	Neutral	**89020**	24590	55890	169500
	Negative	24590	**6792**	15439	46821
	Positive	55890	15439	**35090**	106419
Marginal Sum of Predictions		169500	46821	106419	322740

This table is formed by "randomly choosing the observations" by allocating the observations according to the underlying distribution. A random choice benchmark is the most appropriate in this scenario, as the general theory is that models are not able to beat analysts in the estimation process (Brown, 1987).

By looking at the proportions and precision in *Table 6* below, we can gain a better understanding of the model's performance results. All classes provide far superior precision scores than the sample proportions, i.e., random selection. The Negative surprise class' precision experienced the greatest improvement over and above random selection; an improvement of 26 percentage points, followed in order by the Positive and Negative class, with 21 and 12 percentage points respectively. Further, by looking at *Table 4* to *Table 6* together, we gain a better overall presentation of the model's performance. Of an aggregated total of 322,740 tested observations, 153,240 (48%) are surprise instances (observations). The model correctly recalled 59,361 (39%) of all the surprise instances. The model predicted

94,515 instances of positive surprises, 51,483 of which were correct, 43,032 of which were wrongly classified as either neutral or negative surprise, giving a precision score of 54%. This is far better than a random choice of 33%. The model predicted 19,380 instances of negative surprises, 7,878 were correctly predicted, and 11,502 were wrongly classified; this gives a precision score of 41%, which is once more much better than the random choice of 15%.

Table 6: Surprise Breakdown Percentage Composition, Proportions, Recall and Precision Measures

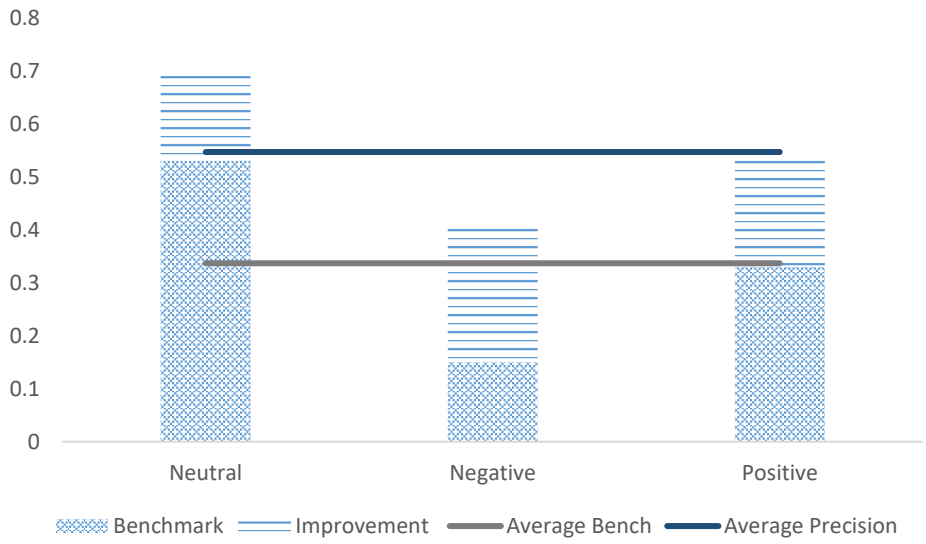
Percentage Composition Confusion Matrix		**Predicted**			Sample Distribution	Recall
		Neutral	Negative	Positive		
Actual	Neutral	**0.45**	0.01	0.06	0.53	0.85
	Negative	0.05	**0.02**	0.07	0.15	0.17
	Positive	0.15	0.02	**0.16**	0.33	0.48
Predicted Proportions		0.65	0.06	0.29		
Precision		0.69	0.41	0.54		
+ive to -ive Outcome Ratio		N/A	1.07	2.33		
Improvement		0.17	0.26	0.21		
Average Accuracy			0.63			

This table reports the same information as is reported in *Table 4* just in another way and with some new calculations over the results. In this table, we report the proportions instead of the number of observations. If you look at the correctly predicted negative observations, they are 2% of the overall proportion, but 15% of the population, so it is clear the model sacrifices recall for precision. This recall score can be improved by changing the decision threshold at the expense of precision. The Recall reports the proportion of the predicted category observations to the underlying sample of that category. The Precision reports the proportion of the predictions that are correct. To use an example above for the Negative Prediction, Recall = Correctly Predicted/Sample Proportion = 0.02/0.15 = 0.17. Precision = Correctly Predicted/Predicted Proportions = 0.02/0/07 = 0.41. More formally, Recall: TP/(TP+FN), Precision: TP/(TP+FP). The average is calculated by multiplying the precision scores with the underlying sample distribution. The average equal weighted percentage point improvement over all classes is 0.20 as can be seen in Figure 5.

This information can be further broken down by investigating the effect of differently sized surprise thresholds. *Table 7*, near the end of this section, shows, that on average, the higher the surprise thresholds are, the greater the improvement over random selection. For surprises 15%+, the average percentage point improvement is about 30%. This improvement is much more consistent for positive than negative surprises as showcased by a 0.03 as opposed to 0.12 confidence interval. The average improvement across the different surprise

thresholds (5%, 10%, 15%) is not that dissimilar from each other and falls between 0.14 - 0.16 with similar sized confidence intervals.

Figure 4: Precision Score Figure Accompanying *Table 6*



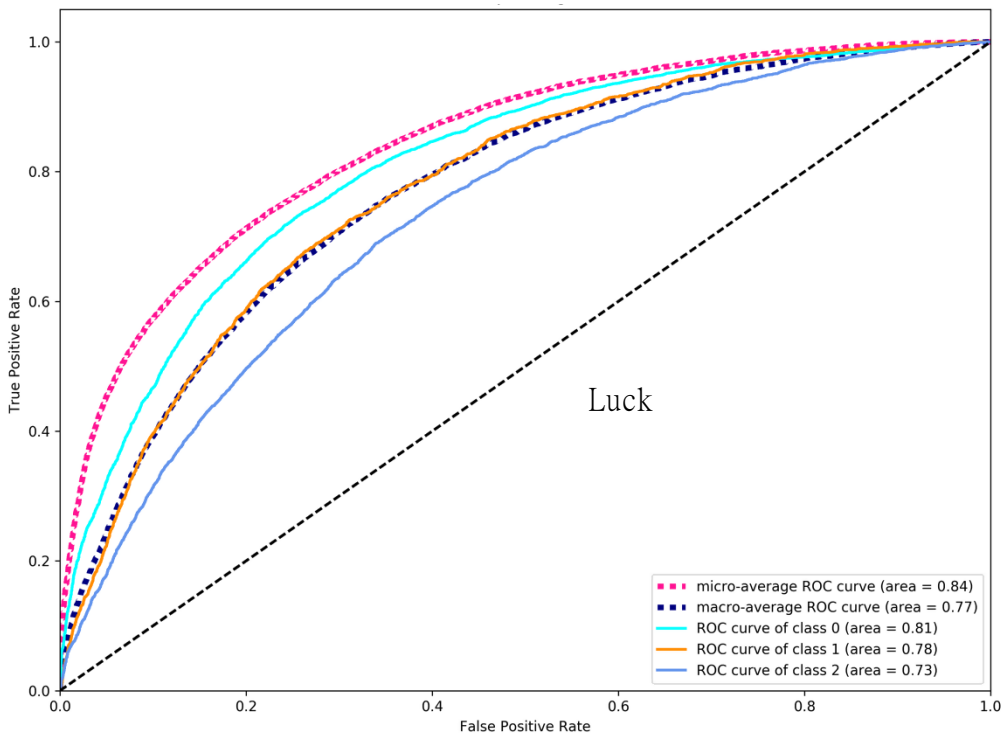
This figure presents the precision scores, i.e., the accuracy of correctly predicting neutral as neutral, negative as negative and, positive as positive, in an easy format. The lines above the benchmark are the improvement the model makes over and above the benchmark precision. Finally, the two average precision scores represent the average precision score vs. the average benchmark score. It is useful to note that the average precision score is different compared to the average. The simple average weights the different classes according to their underlying sample distribution, whereas the average precision presented here equal weights the classes.

The best model for predicting surprises is the 15%+ threshold, likely due to the patterns leading up to the announcement being more pronounced than that of lower thresholds. The higher surprise thresholds performance is slightly being offset by a decrease in the improvement for predicting the ever-increasing neutral class. If we ignore the neutral class, i.e., isolate surprises, we can clearly see an increase in the overall surprise prediction performance. This is an interesting finding as it shows that the model is picking up distinct and definitive patterns. The growing improvement also makes it much easier to create trading strategies for bigger deviating surprises.

The multiclass ROC is a universal way to identify the performance of a classification model. The AUC (area under curve) score provides an integral-based performance measure of the quality of the classifier. It is arguably the best single-number metric that machine learning researchers have to measure the performance of a classifier. The middle line is a line of random ordering. Therefore, the tighter the ROC-curves fit to the left corner of the plot, the better the performance. Two other measures included in the graph are a macro-average

measure that gives equal weight to each class (category) and a micro-average measure that looks at each observation. AUC values of 0.70+ are generally expected to show strong predictive effects. The ROC test adds additional evidence in favour of the model's performance being substantially different from null. In subsequent tables, *Table 7* and *Table 8*, I will also test the statistical significance of this outperformance.

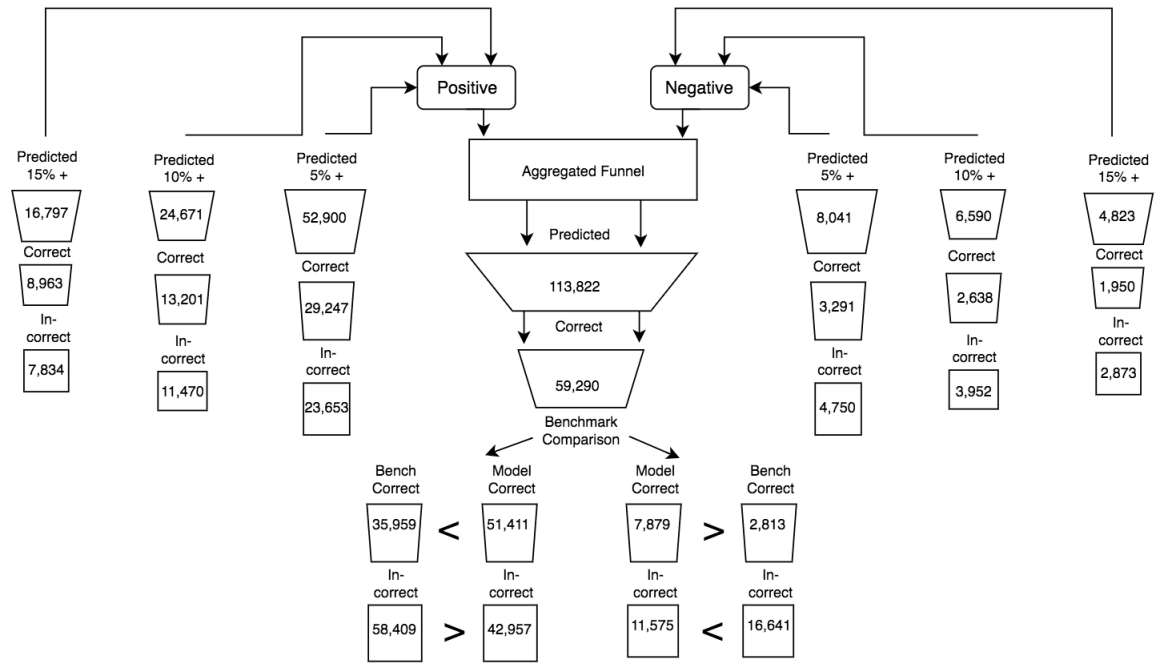
Figure 5: Multiclass Receiver Operating Characteristic (ROC) for a 15% Surprises Strategy



This figure reports the ROC and the associated area under the curve (AUC). The ROC (AUC) is measured for three different classes: class 0 is the negative surprise class, 1 is the neutral class, and 2 is the positive surprise class. The macro-average measure is the equal weighted AUC for all classes and the micro-average measure looks at each class's observation weight. The random ordering or luck line is plotted diagonally through the chart.

Figure 6 is the visual representation of the results presented in *Table 7*. This figure singles out positive and negative surprises and helps to establish an overall outlook of surprises by aggregating all results in a funnel. Random choice benchmark models are also included so that we can compare the significance of the results. This provides a further robustness test to see if, in aggregate, both types of surprises outperform the benchmark.

Figure 6: Model Surprise Prediction Funnel



This figure reports the quantitative performance of the aggregate performance of three different classifier. The left side is the positive surprises and respective thresholds and the right side is the negative surprises. This figure is slightly more involved than the confusion matrix, but it comes down to the same principle. The direction of this figure is top-to-bottom. The benefit of this prediction funnel is that it brings the predictions into an aggregate funnel to be compared against a random choice benchmark. This figure visualises the results of *Table 7*.

Table 8 finally provides interesting results of the performance of the model over different time intervals and with different levels of training data. The table identifies two important generalisations of machine learning prediction in time-series. The first is that with the inclusion of more data, the model tends to perform better (Domingos, 2012). Nevertheless, this does not always hold true for time-series data because of the potential of differences in distributions over time. Something as simple as seasonal trends can lead to bad predictions. There is, of course, ways to mitigate this, such as incorporating seasonal indicators as variables in the model. We can also difference away seasonality, but longer-term changes in analyst behaviour, such as shifts in the median analyst forecast over longer periods would remain an issue for prediction success (Brown, 2001). In machine learning, it is desirable that the distribution of the train and test data is the same, but this is not always possible, especially with financial data (Montas, Quevedo, Prieto, & Menndez, 2002). The expectation is that the results will improve if the distribution remains unchanged. This is especially true for the recall metric; as an example, we can look at the low recall rate of

negative surprises in *Table 6,* where the training set had many more "negative" surprises than the testing sets on average, hence the lower recall rate.

In *Table 8*, it can be seen that the second time-split performs worse than the first time-split, even though it has more data in its training set. This could be because of the characteristics of a particularly noisy training period, 1990-2003, that incorporates the tech bubble, which might have caused some systematic changes across the variables and change how they relate to surprises, leading to worse predictions in the future period. Before the inclusion of the period 1998-2003 in the training set, the model performed much better compared to the benchmark. A further explanation for this drop in improvement is that time-split two applies its learnings to a test set for the years 2004-2008, which in itself was a particularly noisy period containing the housing bubble and the start of the GFC (global financial crisis). The model performs better in the next two periods where both the tech bubble and the GFC are incorporated into the training set, and where the training set increased from 53,796 to 80,694 observations, further helping to improve the prediction outcomes.

Overall the predictions show that the model beats the benchmark for all the periods tested but to different extents; this is ascertained in two ways, first with ROC curves and then with statistical tests. Furthermore, it seems that the portion of correctly classified observations remains relatively stable. When looking at the inclusion of all time periods (the *All* row, last column), we yet again note only a small amount of improvement discrepancy across surprise thresholds. The results of the sub-analysis in table *Table 7* Panel B indicate that if we isolate the surprises from the neutral firms, they significantly outperform the benchmark precision scores.

In summary, splitting training and test sets by time intervals and checking for parameter stability over time is a very useful exercise in building a robust model and understanding how the model learns and predicts. In this part of the study, I have shown that a classification model can predict a surprise with much better precision than a naïve benchmark. I further revealed that this outperformance holds for various surprise thresholds and holds true over multiple periods, making the results robust. Next, I dig deeper into the significant predictors/variables driving the predictions and utilise the results to construct a profitable trading strategy.

Table 7: Class Surprises Count Statistics

Class	Surprise Deviation	Number of Predictions	(1) Model		(2) Random Guessing		(3) Improvement		Student's t-test
Panel A: Class Precision Analysis			Count	Precision	Count	Precision	Count	p.p.	
Positive	5% +	52900	29247	0.55	24030	0.45	5217	0.10	13.14
	10% +	24671	13201	0.54	7861	0.32	5340	0.22	37.49
	15% +	16797	8963	0.53	4068	0.24	4895	0.29	24.04
		94368	51411	0.54	35959	0.34	15452	0.20	24.89
Negative	5% +	8041	3291	0.41	1412	0.18	1879	0.23	4.65
	10% +	6590	2638	0.40	879	0.13	1759	0.27	5.78
	15% +	4823	1950	0.40	522	0.11	1428	0.30	6.57
		19454	7879	0.41	2813	0.14	5066	0.27	5.67
Neutral	5% +	47055	26782	0.57	18066	0.38	8716	0.19	29.17
	10% +	76132	52280	0.69	42197	0.55	10083	0.13	6.75
	15% +	85694	65173	0.76	56095	0.65	9078	0.11	5.40
		208881	144235	0.69	116358	0.53	27877	0.14	13.77
Sum/Avg (4)		322776	203525	0.63	155130	0.34	68913	0.20	14.78
Panel B: All Classes and Surprises Precision Analysis									
All Classes	5% +	107996	59320	0.55	43508	0.40	15812	0.15	15.65
	10% +	107393	68119	0.63	50937	0.47	17182	0.16	16.67
	15% +	107314	76086	0.71	60685	0.57	15401	0.14	12.00
Surprise Classes	5% +	60941	32538	0.53	25442	0.42	7096	0.33	15.28
	10% +	31261	15839	0.51	8740	0.28	7099	0.48	16.91
	15% +	21620	10913	0.50	4590	0.21	6323	0.59	6.26

(1) represents the correctly predicted count and precision of the model; precision is the percentage of correctly predicted observations for each test, while count is the number of successful predictions. (2) produces the results for random choice. (3) the improvement is presented in count and in percentage point improvement, i.e., (1) model precision minus (2) random precision. This improvement has been tested over an unequal variance paired t-test from which confidence intervals were calculated and are reported. I also produce sub-

calculations for each category separately, (4) as well as sub-calculations for all models combined, found beside 'Sum/Avg'. Panel B, All Classes, expands on (4) by deconstructing it into the thresholds (surprise deviation), giving us an indication of each threshold's performance. Since one purpose of this study is to create trading strategies, it is useful to know what the surprise performance is over and above random choice; the Surprise Classes provides the necessary precision and improvement scores of surprises. It is clear that the model outperforms random choice not only at lower surprise thresholds, but also it increasingly outperforms random choice at higher thresholds.

Table 8: Test Intervals Surprises Count Statistics

	Train Size	Train Dates	Test Dates	Train Size	Test Size	Surprise Deviation	(1) Model		(2) Random Guessing		(3) Improvement		t-test
							Count	Precision	Count	Precision	Count	p.p.	
1	20%	90-97	98-03	26898	26898	5% +	15249	0.57	10411	0.39	4838	0.18	5.26
						10% +	17866	0.66	13527	0.50	4339	0.16	4.75
						15% +	19752	0.73	15904	0.59	3848	0.14	2.64
							52867	0.66	39842	0.49	13025	0.16	4.21
2	40%	90-03	04-08	53796	26898	5% +	14447	0.54	10854	0.40	3593	0.13	3.90
						10% +	16785	0.62	13307	0.49	3478	0.13	3.81
						15% +	19012	0.71	16034	0.60	2978	0.11	2.04
							50244	0.62	40195	0.50	10049	0.12	3.25
3	60%	90-08	09-12	80694	26898	5% +	14712	0.55	11500	0.43	3212	0.12	3.49
						10% +	15840	0.59	11176	0.42	4664	0.17	5.10
						15% +	17768	0.66	13191	0.49	4577	0.17	3.14
							48320	0.60	35867	0.44	12453	0.15	3.91
4	80%	90-12	13-16	107592	26898	5% +	14912	0.55	10743	0.40	4169	0.15	4.53
						10% +	17628	0.66	12927	0.48	4701	0.17	5.15
						15% +	19554	0.73	15556	0.58	3998	0.15	2.74
							52094	0.65	39226	0.49	12868	0.16	4.14

Continued on next page

Table 10 - Continued from previous page

Total	22	18	806940	322776		203525	0.63	155130	0.48	48395	0.15	3.88
All	22	18	268980	107592	5% +	59320	0.55	43508	0.40	15812	0.15	4.30
					10% +	68119	0.63	50937	0.47	17182	0.16	4.70
					15% +	76086	0.71	60685	0.56	15401	0.14	2.64

Precision is the percentage of correctly predicted observations for each test; count is the number of successful predictions. This table identifies the correctly (1) classified surprise categories over different test periods. The table also provides for some aggregate measures over all the time-splits; these measures are found beside *Total* and *All*. *Total* is the average and or summation across all tests, while *All* still leaves the percentage surprise buckets intact. In machine learning, these measures are commonly referred to as validated metrics. (3) the improvement is presented in count and in percentage point improvements, i.e., (1) precision - (2) precision.

B. Prediction Analysis

A large part of this study is dedicated to the creation of an improved prediction model. Finance researchers are often interested in understanding the predictors. Machine learning does not make this an easy task. In this section of the study, the focus is on gaining a better grasp on the predictors to the classification model. Here the machine learning model helps us to determine biases by identifying variables that are important in predicting surprises over and above the analysts' forecast. There are 19 different varieties of technical indicators used in this study, with three different parameter timeframes, resulting in 57 technical indicators overall. The indicators are split into 1 - 4, 5 - 9, and 10 - 20-day lookback periods. After mapping signal processing and other algorithms over the indicators and doing the first and second phase of variable selection on validation sets, the process generated 677 relevant variables[15]. Of the original 57 technical indicators, 55 remained relevant. Of the 69 earnings-related variables, 62 were relevant. The overall number of relevant variables amounted to 794.

Table 9 shows a list of the five most important variables to identify surprises. This section specifically focuses on the most telling variables for predicting large positive surprises. The average score is collectively calculated from three tree models, XGBoost, AdaBoost, and Random Forest, to uncover an intersection of important variables for all tested surprise thresholds and test-train splits. This lessens the likelihood of the variables appearing by chance so that the selected variables are agnostic to the type of tree model. This process is known as tree-based variable selection. These variables have been normalised across the models, and the intersection of all variables revealed the most important subgroup.

Multicollinearity has a big influence on the importance measure. The variable importance measure used in this study is called Gini Importance. This measure is based on the number of splits each variable undergoes, weighted by the squared improvement that results from each split that gets averaged over all trees (Elith, Leathwick, & Hastie, 2008). In layman's terms, the more an attribute is used to make key decisions with decision trees, the higher its relative importance. This measure can be thought of as a 'significance' score for decision tree models. The reality is that a single variable's importance is reduced if there are a multitude of variables of similar character and correlation to the response variable. In this study, there are 732 pricing variables and 62 earnings variables. Thus, the importance of the

[15] Relevancy means that the variable has a non-zero effect on the predicted outcome.

pricing variables is widely distributed among relatively homogenous variables. Therefore, it is more useful to look at the cumulative importance of the pricing variable, which further results show to be around 20.9%. The cumulative performance of each group of variables is reported at the bottom of *Table 9* and *Table 10.* In simple terms, more than 20% of the decisions get made as a consequence of using price-related variables.[16]

The issue with many machine learning models is that their nonlinearity makes it hard to enforce monotonicity constraints to identify in which direction the relationship is between the independent variable and the machine-learned response function. In ML, the response can change in a positive or negative direction and at varying rates for changes in an independent variable, making the interpretation of variable importance much harder than simply looking at the coefficients of a linear model. Although the singular importance value of a variable can be very helpful for understanding the average direction and size of a variable to the response, it does not explain the potential nonlinear relationship of the variable with the response. Information about this relationship are of great interest to researchers and industry experts alike. To identify the relationship, we can make use of a technique called partial dependence.

Partial Dependence is a means of identifying the marginal dependence between the predictors and the outcome (Hastie, Tibshirani, & Friedman, 2009). The basic premise of this technique is to obtain a prediction for all unique values of a variable while accounting for the effects of all the other variables. Breaking this development process down, for every unique value of the variable of interest, a new dataset is created where all the observations of the variable is set equal to that unique value and all other variables are left unchanged; the new dataset then gets ingested in a decision tree where all the prediction are averaged and plotted. This process is repeated for all the values of the variable of interest to get a range of outputs for inputs, and similarly, this can be repeated for variable value pairs. This approach, as a result of incorporating all the information from other variables, has the ability to detect nonlinear relationships without the need to pre-specify them, and it allows us to visualise the relationship between an input and a response variable. See Appendix, *Method A 4* for an expanded explanation and the mathematical formulae driving this concept.

The nonlinear nature of these relationships can be seen in *Table 9*, where the direction can change signs as the variable value increases. *D* represents the direction in which an increase in output would drive output. For a classification task such as in *Table 9*, a higher

[16] Apart from the correlated variable problem, the Gini Importance measure is also biased towards variables with more categories. This variable importance measure can be inconsistent for tree ensembles as higher importance can be assigned to variables with a lower impact on the model's output.

output means a higher likelihood of a positive surprise. For both *Table 9* and *Table 10*, 1 means a 100% chance of a positive surprise, and 0, a 0% chance of a positive surprise. A better visualisation of the output value and the relationship of these values can be found in *Figure A14* for earnings-related variables and *Figure A15* for price related variables. Partial dependence between two independent variables and the outcome variable can also be interesting, especially for visualising more complex relationships. The most important of these combined interactions have been included in a graph after the analyses of each set of variables. Moreover, all the important variables have partial dependence plots, see *Figure A14* in the Appendix.

Table 9: Earnings Related Variable Importance and Response Direction for Classification

Name	**Short Description**	**Score**	**D**
est_avg_t	This time period's analyst EPS forecast	0.247	-
$diff_{-1}$	The difference between the past actual EPS, p_{-1} and p_{-2}	0.119	+/-/+
p_{-1}	Actual EPS t_{-1}	0.082	-
$d\_e\_diff_{-4}$	Difference between the past actual, p_{-4} and forecast est_avg_{t-4}	0.073	+
$diff_{-4}$	The difference between actual EPS p_{-4} and actual p_{-1}	0.060	+/-/+
Other	57 other earnings-related variables.	0.212	
Total		**0.794**	

This table identifies the most important earnings-related variables to predict a positive earnings surprise. The Gini importance, which is the average gain in information, is used to represent the variable importance (Score). *D* identifies the direction of the variable as identified by the partial dependence graph. See *Figure A14*, the graphs from which the directions are identified.

The most important earnings-related variable is the analyst consensus forecast itself, est_avg_t; this is expected because the purpose of the model is to identify deviations from this value and the true value to identify surprises. It provides a measure of reference to the other variables of the model. Countless papers have identified the performance of analysts' forecasts in forecasting earnings as recapped by Brown (1987). The lower the forecasted EPS, the more likely a surprise is to occur all else equal; 32% of the outcome is attributable to this value.

The second most important variable is the difference between the actual level of earnings between period t_{-1} and t_{-2}, called $diff_{-1}$ and the fifth most important variable, is the difference in earnings between t_{-1} and t_{-4}, called $diff_{-4}$. These are novel variables not

yet identified by past research. The extent of past increases in earnings are therefore an important variable for predicting future surprises. If the value is very high, surprises become more likely. However, surprises are also more likely if the value gets very low. In the middle range surprises become less likely. The measure is u-shaped, which is indicative of a sort of variance measure.

The next important value is the actual earnings at time t_{-1}, called p_{-1}. Research by Bradshaw, Drake, Myers, and Myers (2012), has shown that the past annual earnings often outperform not just mechanical time-series models but also analysts' forecasts. Past quarterly earnings seem to be an important value in predicting the next quarter's earnings surprise. The relationship between past earnings (p_{-1}) and the current analyst estimates (est_avg_t) shows that where p_{-1} is very large and est_avg_t is very low, then a positive surprise is likely to occur more than 90% of the time, all else being equal. Further, where p_{-1} is low and est_avg_t is high, a surprise is unlikely to occur.

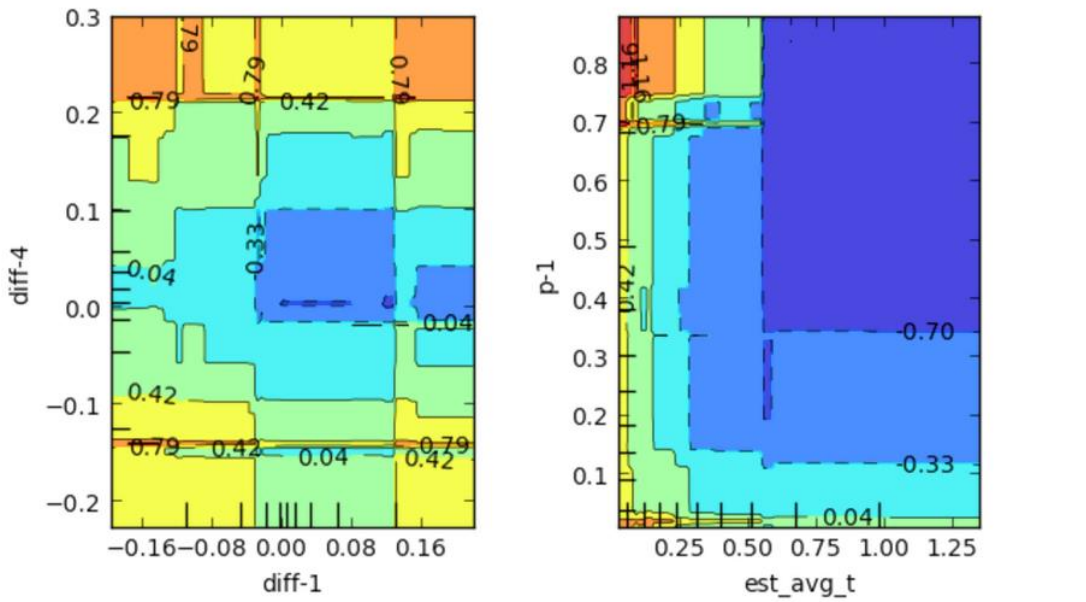
The fourth most important variable is the difference between past and forecasted earnings four quarters ago, i.e., one year between the forecast, est_avg_{t-4} and the actual value, p_{-4}. The importance of this variable was also expected since Easterwood and Nutt (1999) and Fried and Givoly (1982) separately showed that past errors tend to persist. The larger the difference, the higher the likelihood of surprise. Other variables that showed an above 2% importance include rolling averages and weighted rolling averages of the difference between past earnings and analyst forecasts, and the standard deviation of analyst forecasts.[17]

It is not always the best practice to look at the isolated effect of an input variable on the response variable; for that reason I have included the dependence plots in *Figure 7* between two input variables and the output. Out of the above list, there are more than 32 (2^5) ways to conjure up directional relationships. To conceptually understand the web of relationship, due to the nature of nonlinear relationships, it is better to describe a simple tree explanation of the above variables and relationships for a combination of variables that would lead to a perfect surprise prediction. The following explains what would lead to a close-to-

[17] More specifically, $d_{e_{p-1}}$, the difference between p_{-1} and est_avg_{t-1}. And d_e_16, the rolling mean of actual EPS, p from $t_{-16}$to t_{-1}, minus the forecast rolling mean from est_avg_t from $t_{-16}$to t_{-1}. The standard deviation of all individual forecasts, est_std. The higher the standard deviations, est_std, the larger the uncertainty among analysts. This is associated with a decrease to the machine-predicted EPS. The reason is that the base, analyst forecasts, est_avg_t, tend to be excessively positive in periods of uncertainty as has been reported by Das et al. (1998).

perfect surprise prediction. If the current analysts' forecast is low while the past earnings are high, there is a higher likelihood of a positive surprise this period; this is likely due to analysts being conservative. If in the past it has been shown that analysts are conservative and that surprises transpired, i.e., $d\_e\_diff_{-4}$, then the likelihood of surprises increases even more. When there is a large difference in earnings between the last two periods EPS, p_{-1} and p_{-2}, i.e., $diff_{-1}$, the likelihood of surprise increases. The same holds for $diff_{-4}$. Overall, these variables accounted for around 80% of the variable importance. Referring back to *Table 1*, it has also been shown, using an inductive method to theory testing, that these variables accounted for more than half of the total improvement over the benchmark. The remaining importance is distributed between pricing and price-derived variables.

Figure 7: Partial Dependence of Class Probabilities on Earnings Related Feature Combinations for Classification



The figures indicate the probability that an earnings surprise will occur, all else being equal. The dashed lines identify the space where earnings surprises are less likely to occur. The small ticks on the axes are an indication of the underlying distribution. As the colours get warmer, surprises are more likely to occur, the colder the colour the less likely a surprise is to occur. These graphs show the partial dependence relationships between two variables. On the *left*, it can be seen that a surprise is more likely if both $diff_{-1}$ and $diff_{-4}$ are large and that surprises are less likely when both of these values are around the mean. This would indicate that there is a predictable trend. Another interesting observation is that if the longer trend $diff_{-4}$ is large and shorter-term earnings decrease, $diff_{-1}$ is negative, then the small blip is short-lived and likely to be corrected in the next period, as can be seen with the high likelihood of surprise in this area, >79%, i.e., the top left corner. On the *right*, as previously noted, when p_{-1} is large and est_avg_t is low, a surprise is likely, >90%, and where p_{-1} is low and est_avg_t high a surprise is unlikely.

The variables in *Table 10* identify the shapes and patterns associated with the market before the firm makes its official earnings announcement. These variables can potentially

reflect signals of management's position of the forthcoming announcements, or it can simply reflect traders' use of privileged or public information. This information can be firm-related or more broadly economic in nature; either way, these variables reflect information not contained by the smaller scope of earnings-related variables. Many of the mapping transformations identify certain characteristics and shapes associated with technical indicators. One example is a mapper that identifies how often the price crosses the top Bollinger Band in the days leading up to the announcement. A second example is a measure of the serial sum of absolute changes for a Relative Strength Index.

In essence, a mapper is a transformation that helps to identify patterns in time-series data. Incorporating the signal-processed algorithms over the traditional technical indicators provides a significant improvement to the overall variable importance of the pricing variables, as evidenced by the below table, with a total variable importance of 20.9%. This has further been proven in the classification subsection, *Table 1*, where an inductive method to theory testing revealed that these variables account for more than one-third of the total improvement over a naive benchmark. A possible downfall is that the variables presented here are not as easily interpretable as the earnings-related variables in the preceding table.

Table 10: Pricing Related Variable Importance and Response Direction for Classification

Base Feature	**Mapper**	**Parameters**	**Score**	**D**
$Chaikin$	Mean abs change quantiles	qh:1.0, ql:0.4	0.036	+
$Donchian_{10}$	Friedrich Coefficients	m:3, r:30, coeff: 2	0.026	-/+/-
MA_3	Max Langevin Fixed Point	m:3, r:30	0.013	-
$Trix_{10}$	Autocorrelation	lag:8	0.009	-
$KelChU_5$	Max Langevin Fixed Point	m:3, r:30	0.007	-
Other (727)			0.118	
Total			**0.209**	

The above variables have been selected as the 5 most important variables out of all standard pricing, technical and signal processed variables. For many of the variables, zero seems to be an important boundary in deciding in what direction the output 'moves.' See *Figure A15* for the graphs from which the directions are identified. The mapper is a transformation that helps to identify patterns in time-series data. It transforms a series of data into a single number to be fed into the machine learning model. Parameters are the auto-selected parameters that the mapper identified to be most informative in explaining the response function. A mapper has on average 11 different parameter iterations it tests against the output on a validation set.

Looking at *Table 10*, the most important pricing variable is a transformation mapped over a Chaikin technical indicator. The Chaikin Oscillator is an indicator of an indicator, the

latter of which is derived from the stock price. The Chaikin indicator is a third-derivative indicator designed to anticipate directional changes in the Accumulation Distribution Line, a volume-based indicator, by measuring the momentum behind the movements. The purpose of the Chaikin indicator is to predict directional changes in a price trend. The mapper entails a calculation of the average distance between each consecutive value in time-series within a high (1) and low quantile (0.4) range for the $Chaikin$ measure. It, therefore, provides an absolute measure of the amount of variance the Chaikin Oscillator experienced for each subsequent value between a skewed high and low quantile, for the 30 days before the announcement. If the absolute difference is large, a surprise is more likely to occur, ceteris paribus.

The Donchian 10 technical indicator is formed by taking the highest daily high and the lowest daily low of the last k periods and computing the difference. It is a measure of volatility on the one hand but can also provide signals for long and short positions (Patel, 1980). The Friedrich coefficient mapper allows us to fit and calculate the coefficients of a polynomial equation, $h(x)$, that is fitted to the deterministic dynamics of the Langevin Model (Siegert, Friedrich, & Peinke, 1998). Langevin dynamics is essentially an approach to mathematically model the dynamics of molecular systems (Langevin, 1908). In simple terms, it is an equation that describes Brownian motion. This specific extension of the formula has been used in finance-related papers such as those describing the dynamics of market crashes and to quantify random fluctuation in the foreign exchange markets (Bouchaud & Cont, 1998; Friedrich, Peinke, & Renner, 2000). The particular parameter of interest is the Potential coefficient, which is affected by the amount of time the value does not deviate from a harmonic-like motion. If this amount is low, then surprises become less likely; if this value is in the middle ranges, then surprises are much more likely; and lastly, if this value is very high, a surprise is less likely.

MA 3 is a simple moving average indicator that tracks the last three days of pricing history. We will yet again make use of the Langevin model, this time to find the maximum fixed point for a third-degree polynomial fitted to the deterministic dynamics of the Langevin model. In a sense, we are trying to find the maximum fixed point of a rolling momentum average, modelled in the form of Brownian motion. Fixed points are, fundamentally, concepts of stability and equilibria in economics. In physics, it is often a way to predict phase transition. Even though this figure is multidimensional and less interpretable, we can say that the larger the maximum equilibrium in a series of moving averages of a third-degree polynomial, the less likely an earnings surprise will occur.

The triple exponential average, TRIX 10, is a measure used to identify oversold or overbought securities (Hutson, 1983). The measure fluctuates around a zero line. A negative (positive) TRIX value signifies an oversold (overbought) market. This indicator looks at the trend for the last 10 days. The 7-day lag autocorrelation of a rolling time-series of this value over the last 30 days is shown to be an important variable. The higher the autocorrelation of TRIX 10, the less likely you are to experience a positive surprise.

Keltner Channels are trend-following indicators used to identify reversals with channel breakouts. The upper-band of a Keltner Channel indicator, looking at the last five days in combination with a largest fixed-point forecasted from a function which has been fitted to the deterministic dynamics of the Langevin model, shows up as an important variable. A possible interpretation is that the nature of the position and extent of the maximum fixed point of the polynomial is such that a higher value leads to a lower likelihood of surprise and a lower value to a higher likelihood of a surprise.

Machine learning algorithms scour the variable space for patterns to identify associative interactions between input and output values. The relationships and variable directions may completely change when additional variables are added. These associative patterns, such as seen in *Figure A15*, should not be mistaken for causation. They do, however, make for an interesting assessment, especially the variables relating to earnings, as they are easier to interpret; the price variables are slightly less interpretable and somewhat noisy.

C. Trading Strategy

The results reported until now are mutable, in that we can adjust the loss function and surprise thresholds to achieve results that would be better suited to a trading strategy.[18] Precision can be improved but at the expense of recall and overall accuracy; a single class' metrics can be improved at the expense of other classes; even the class probability thresholds can be changed to favour one class over another. It is necessary therefore to identify what is important for a trading strategy and then to optimise for that measure. Every change to the loss function affects another part of the results. In trading, we are particularly focused on improving the precision; this preference can be expressed in the loss function or we can simply alter the sample of firms under observation.

Now that we have a model that can predict earnings surprises on paper with good success, it is necessary to see if it can translate into a trading strategy. It is always possible that the market will not react to what we consider surprises. Earnings surprises are often termed as soft-events in books on event-driven investment strategies. Event-driven investment forms a large part of the hedge fund industry, accounting for about 30% of all $2.9 trillion assets under management according to Hedge Fund Research 2015. Event-driven strategies are a great way to benefit from increased volatility. For the classification results achieved up to this point, it is possible that the consensus forecast is not defined in the same way that the market may perceive the consensus to be. The next section mostly alleviates this concern, as it is shown that the market does indeed react to what this paper defines as surprises. Furthermore, it is possible that the model simply learns on a lot of correlated and tail risk. It is, therefore, worth seeing whether a trading strategy using these surprises can earn long-term cumulative and excess returns.

An important question to ask when combining a classification task and a trading strategy is to know in what categories wrongly classified instances fall to identify the viability of a profitable strategy. For example, if firms experience a negative earnings surprise when a positive surprise is predicted, it can have a significant impact on a strategy's return. To investigate incorrect classification, we make use of a confusion matrix (often called contingency tables). *Table 6* presents a ratio called positive-to-negative outcome, which is an important figure for a potential trading strategy. The ratio can be deconstructed as follows: for positive surprises, it is the ratio of the number of positive surprises correctly

[18] This should only be done on the validation set otherwise it might lead to overfitting.

predicted to the number of negative surprises mistakenly predicted as positive surprises. The negative surprise ratio for example is the ratio of the number of negative surprises correctly predicted to the number of positive surprises mistakenly predicted as negative surprises.

For predicting positive surprises, this number is high at around 2.3, which is reasonably good and means that you would have one critical mistake for every 2.3 positive surprises you predict correctly. For negative surprises, this amount is around 1.1. This amount is too low to form a successful shorting strategy, and this has further been proven by a lacklustre performance of a potential shorting strategy as seen in *Figure 8*. This phenomena of mistakenly predicting both extreme classes are common in machine learning. Although the model does not intend to, it often classifies the surprises according to variables associated with a large variability in earnings, leading to both classes being flagged as potentially correct classifications. The specifications of the model can, however, be changed to penalise false positives more by simply adjusting the loss function thresholds. Doing this decreases the number of trading opportunities available for only a small added benefit. A further step one can take is to sort predictions by the difference in the probabilities of positive and negative surprises and then to only trade on the higher decile predictions. An even simpler approach would be to follow a long-only strategy to take advantage of the large positive-to-negative ratio it offers.

The following expresses a simple strategy by going long (short) on stocks that are expected to experience a positive (negative) surprise tomorrow (t), at closing today (t-1), and liquidating the stocks at closing tomorrow (t). The stocks are equally weighted to maintain well-diversified returns for the day, as there is, on average, only four firms in a portfolio of expected surprises, but there can be as few as one firm in a portfolio.[19] For each day, I form stocks into positive and negative surprise prediction portfolios for surprises that deviate from -50% to 50% in order to select the best performing threshold. The preferred threshold is selected based on tests done against a validation set. The results in the validation set show that the best trading strategies exist between 5%-20%, with 15% being the optimal trading strategy for positive surprises.[20] I, show that an event-driven trading strategy that takes long positions in stocks with 15% positive surprise predictions, while earning the market rate of return over non-earnings surprise days, produces an alpha of 8% relative to a five-factor asset-pricing model on and out-of-sample test set (Fama & French, 2015). There is no good

[19] For robustness, I have also tested for value-weighted returns, which showed a slight increase in improvement.
[20] See the trading strategy section on page 56 for an elaborate explanation of the methodology used.

reason to form a long-short portfolio with negative and positive earnings surprises as the constituent firms in the portfolio are not comparable in type and the positive and negative surprise predictions do not necessarily fall on the same dates.

The strategies developed in this section fully invest all capital in each event. It is therefore important to include some sort of loss minimisation strategy. As a result, one strategy incorporates a stop-loss for stocks that fell more than 10%; 10% is only the trigger, and a conservative loss of 20% is used to simulate slippage. This is done by comparing the opening with the low price of the day. An endless number of opportunities and strategies can be created; it is, therefore, important to select simple strategies not to undermine the robustness of this study. In saying that, the choice of slippage is not backed by any literature and is an arbitrary, albeit conservative, choice for the strategy.

Equation (4) is a simple return calculation for an earnings announcement of firm *i* at time *t*, where the daily low price, Sl_{it}, is not more than 10% lower than the closing price $S_{i,(t-1)}$. If it is, then a slippage loss of -20% is allocated to the return quarter for that firm. The stop-loss is only applied in one strategy in *Table 12*, all other results are reported without any risk mechanisms.

$$R_{it} = \frac{\left(S_{it} - S_{i,(t-1)}\right)}{S_{i,(t-1)}}, \qquad if \ \ \frac{Sl_{it} - S_{i,(t-1)}}{S_{i,(t-1)}} < -10\%, \ \ R_{it} = -20\% \tag{2}$$

In this equation, S is a set of all the firms in the sample, i is the firms that have been predicted to experience an earnings surprise based on preselected thresholds at time *t*.[21]

The equal weighted return of a portfolio of surprise firms is then calculated as such,

$$R_{pt} = \frac{1}{n}\sum_{i=1}^{n_{pt}} R_{it} \tag{3}$$

In this equation, i is all the firms that are predicted to experience a surprise on date t. Therefore, R_{it} is the return on the common stock of firm *i* on date *t* and n_{pt} is the number of

[21] $S_{i,(t-1)}$ is the closing price of the common stock of firm *i* on date *t-1*. Sl_{ti} is the daily low of the common stock price of firm *i* on date *t-1*.

firms in portfolio p at the close of the trading on date *t-1*. R_{Mt} is the value-weighted market return.

The equation below is the five-factor asset-pricing model. In this equation, R_{pt} is the return of portfolio p for period t. R_{Ft} is the risk-free rate. The alpha, a_i is the abnormal return of the trading strategy. R_{Mt} is the value-weighted return of the market portfolio. SMB_t, HML_t, RMW_t and CMA_t are the respective differences between diversified portfolios of small stocks and big stocks, high and low B/M stocks, robust and weak profitability stocks, and low and high investment stocks. To perform the regressions, the respective daily values were obtained from Kenneth French's website.[22]

$$R_{pt} - RF_t = a_i + b_i\,(R_{Mt} - R_{Ft}) + s_i SMB_t + h_i HML_t + r_i RMW_t + c_i CMA_t + e \qquad (4)$$

In *Table 11*, I report the results of three surprise strategies each with a different surprise threshold.[23] The 15% long strategy is the only of these three strategies that shows statistical significance for abnormal profits. The daily abnormal returns generated by the 15% long strategy is 0.037% before transaction costs. The economic viability of these strategies is represented by the cumulative return graph in *Figure 8*. These figures represent a more effective performance measure using Monte Carlo simulation. Instead of simply comparing the portfolio against the market, the performance of the strategy can be compared against a simulated average of the cumulative return from randomly picking from the sample of firms before earnings announcements. These sample firm quarters include surprises and neutral observations. The model shows its superiority at being able to pick out future surprises by beating this simulated average at its 99% significance bands, as identified by the blue channel.

$$R_{pt} = \frac{1}{n}\sum_{i=1}^{n_{pt}} R_{it}\,, \qquad where\ n = 0, R_{pt} = R_{Mt} \qquad (5)$$

[22] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html
[23] In total the parameter searched 100 different strategies, -50% to +50%, only six are reported, -15%, -10%, -5%, and +5%, +10%, +15%. 15%+ was the best performing strategy.

Table 11: Daily Abnormal Returns for Large Firms Trading Strategy

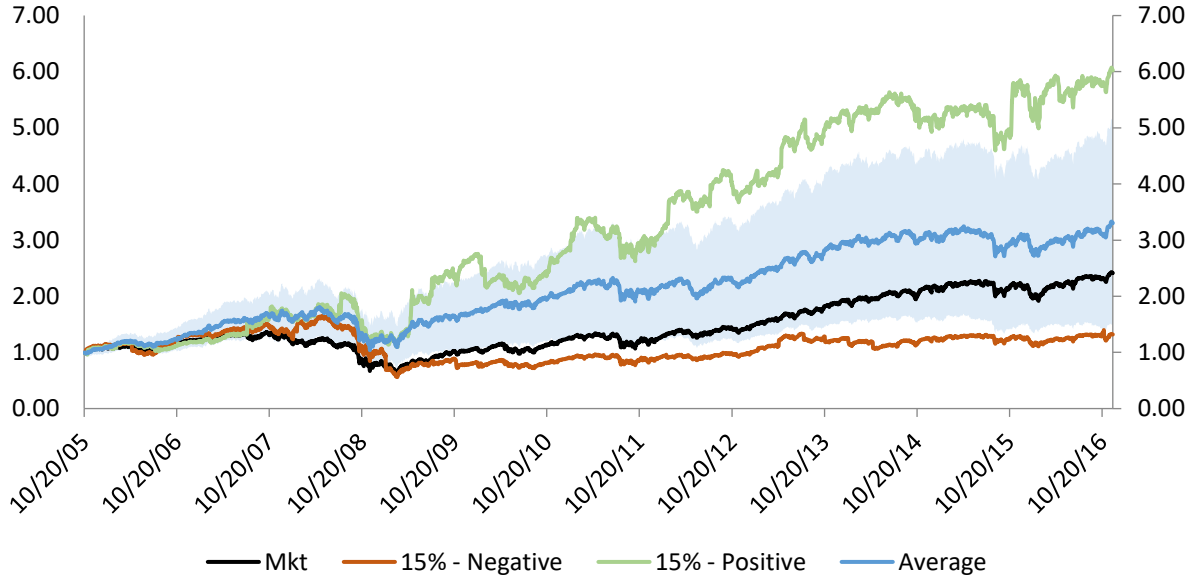
Threshold	Position	Firm Qtrs.	Surprise Days	Abnormal Returns
5%+	Long	5339	1909	0.04%
	Short	267	139	0.00%
10%+	Long	1953	863	0.03%
	Short	111	67	0.01%
15%+	Long	774	220	0.04%**
	Short	234	135	0.02%

Long identifies the positions taken in expected positive surprises portfolios; short is the short-selling of portfolios where negative surprises are expected. Firm quarters are the number of earnings quarters used across all surprise portfolios. Surprise days is the number of days on which surprises occurred and formed part of the portfolio. The table tracks the performance of a strategy over 2994 trading days. All strategies showed significance for the market coefficient. For all the long strategies, HML showed positive significance apart for the 5% long thresholds. For all the short strategies, SMB and CMA showed positive significance. See *Table A17* for a factor table that reports on the size of the positive and negative coefficients and their significance. The alphas with this strategy are somewhat low, mostly as a result of the returns fitting strongly on value firms; however, the plotted cumulative returns still show economic significance over the sample period. The cumulative returns are shown in the tables below. The same study has been performed on smaller firms, but this did not show any strong significance, likely as a result of investors being slower to react to smaller firm surprises and lower analyst coverage leading to inaccurate surprise calculations and stale forecasts. The significance values are calculated from Newey-West adjusted standard errors with 3-month lags. *10% significance **5% significance ***1% significance.

The daily abnormal return of the best performing strategy (15%+ threshold) is around 0.037% daily and 10% annualised. By adjusting for the effect of trading costs at a conservative 50 bps per round trip, both for the surprise and market portfolios, the daily abnormal return decreases to 0.029% and around 8% annualised. This result is especially good as it is driven by only 20 surprise portfolio days on average per year; each surprise portfolio day produces around 0.40% in abnormal return. The abnormal return over the sample period is directly related to the amount of successfully predicted surprises. I also look into the cumulative effects of raw portfolio returns and compare them against the market in a diagram. To calculate the cumulative return over the sample period, I compound the R_{pt} return over *m* days of the sample to yield the total cumulative return of R_{cum}. In the equation below, R_{pt} is the daily return of portfolio *p* on date *t*, and *m* is the number of sample trading days.

$$R_{cum} = \prod_{t=1}^{m}\left(1 + R_{pt}\right) - 1 \qquad (6)$$

Figure 8: Portfolio Value - Large Firms 15% Surprise Prediction Strategies



This portfolio reports the cumulative returns of buying and holding positive and negative surprise portfolios for all firms in the sample with a market value of $10 billion. On average, there are about four firms for each portfolio day. On days where no trading surprises occur, a position in the market is taken. The band in the middle is the significance band obtained from a Monte Carlo simulation by randomly taking a position in 774 firms before an earnings announcement. The chart also reports the cumulative portfolio return of the market as calculated from the market returns obtained from French's website. The chart shows that negative surprises mostly track the market portfolio. It is possible that some returns can be earned by shorting these surprises over certain periods, but on average it is not a very profitable strategy due to the small amount of shorting opportunities. In total, there are 2944 trading days, for the long strategy; 215 of these days are returns from earnings surprises comprising 774 firms, and for the short strategy 62 days comprising 234 firms.

In *Figure 8*, I go a step further than simply reporting the cumulative performance of a market portfolio. I also present another benchmark to simulate the trading performance of a random selection of stocks from the full sample of firm. I run 1000 Monte Carlo Simulations for each of the strategies by randomly selecting *n* number of firms to trade on before earnings surprises.[24] After this, I calculate the average simulated portfolio value and the 99% confidence bands. The simulation is reported in blue with the associated confidence bands. The positive surprise strategy is in green and the negative surprise strategy is in red. From the plot, it is clear that a simple buy-and-hold strategy on the full subsection of firms would also outperform a passive position in the market. This mostly has to do with the sample selection criteria of large firms that have been around for 8 years or longer. However, it is also possible

[24] *n* takes the size of the actual number of predicted positive surprises.

that randomly buying and holding firms over earnings announcement days can itself be a profitable strategy due to the inherent riskiness of these periods that may not be accounted for by the 5-factor asset-pricing model.

Figure 8 shows that the 15% strategy performs very well even though it has less than one-third of the firm quarter observations than that of the 5% strategy. At the end of the trading period, the portfolio value of the 15% strategy finished at a similar portfolio value to that of the 5% strategy. However, the 15% prediction model seems to produce the most consistent results over the sample period. It experienced no period of excessive draw-downs as has been witnessed with the other two strategies. This strategy was profitable for all but one year, with a loss smaller than 7%. Concerning the last third of the sample period, the strategy did not lose 20-40% of its value like the other strategies; instead, it maintained its value over this time. The reason has to do with the selection of firms further away from the lower bound, and it has a better good-to-bad outcome ratio than the other thresholds and a better precision rate.

The overall accuracy score is 82% for the 15% strategy and somewhat lower at 76% and 66% for the 10% and 5% strategies, respectively. This result echoes the machine learning evaluation results above, for which a higher surprise has more easily distinguishable patterns for the machine learning model to train on, leading to better prediction success. Also, because of the increased magnitude of the surprise, it almost directly translates to bigger returns. Larger surprise thresholds were also tested, but they performed poorly due to the lack of available opportunities at that high threshold. For example, when using a 30%+ surprise threshold, it led to only 214 surprise predictions as opposed to 774 for the 15% strategy. Another effect is the S-shaped surprise return curve that leads to lacklustre improvements in return for bigger surprises.

Table 12 shows that there is potential to earn abnormal returns by using a stop-loss strategy in combination with trading on the predictions of the machine learning model. The stop loss is set at the 10% level, but to be conservative, we selected 20% as the slippage stop-loss. There is very little research on an empirically justifiable level of slippage in event trading strategies; for that reason, I arbitrarily double the executed stop-loss level to settle for a 20% loss. This means that a 10% or more decrease, from the previous day's closing versus the current day's low, is the trigger, but that the effective loss is recorded at 20%. Therefore, no firm has experienced a return loss of more than 20% in the sample. Because the stock would always have been sold the next day, it does not significantly worsen the expected transaction cost. The stop-loss strategy significantly limits the downside of the strategy. In

essence, we do not have to hold on to the stocks until the end of the next day and can sell them off throughout the day as we see fit.

A stop-loss strategy is especially important for event-trading when you fully invest in each event. Grossman and Zhou (1993), show that with drawdown constraints of risk control, a continuous stop-loss strategy is optimal. Stop-loss strategies are not very popular in literature, but in practice, one would be hard-pressed to find event trading strategies where they are not used. Lhabitant (2011) notes that event traders such as merger arbitrageurs usually set up strict stop-losses rules for each transaction and that sticking to this discipline is one of only a few ways in which investors can limit their downside risk. Authors such as Kaminski and Lo (2014) show the theoretical underpinning of a stop-loss strategy and the fact that stop-loss policies can increase expected return substantially while reducing volatility. Han, Zhou and Zhu (2016) show an empirical justification of a stop-loss strategy as it is applied to a momentum strategy. They show an almost doubling in abnormal returns using a disciplined stop-loss strategy.

The current slippage assumes that if the stock falls by 10%, then the stop-loss sells at a 20% loss. The break-even slippage for long portfolios for the 5%, 10%, and 15% surprise thresholds are 38%, 45%, 48% respectively, offering a large margin of safety. Further investigation reveals that the reason the stop-loss performs so well is because, near the end of the sample period (2014-2016) there are about 4 portfolio days where a small amount of bad performing observations are the sole constituents to the portfolio each dropping in value between 60% - 80%. As a result, less than 0.25% of firms are accountable for wiping out the abnormal returns.

All the long strategies in *Table 12* show significance at the 99% level. Shorting predicted negative surprise does not seem to be a viable undertaking. This is mostly attributable to the results reported in the contingency tables in the first part of the study, showing that predicted negative surprises are often in reality positive surprises. Recall that *Table 6* specifies that for every two-predicted positive surprise there is less than one mistaken negative surprise of the same deviation, whereas, for every predicted negative surprise there is one mistaken positive surprise of the same deviation. *Table 12* shows that the performance of the long strategy improves as higher thresholds are predicted; the only issue is that the number of trading opportunities decrease from 1909 for the 5% threshold strategy to 252 for the 15% threshold strategy out of about 3000 trading days.

Table 12: Daily Abnormal Returns All Firms Stop-Loss Strategy

Threshold	Position	Firms Qtrs.	% SL	Surprise Days	Abnormal Returns
5%+	Long	34602	0.03	1909	0.26%***
	Short	1722	0.17	431	0.62%*
10%+	Long	15222	0.04	1449	0.32%***
	Short	1170	0.26	313	0.09%
15%+	Long	9996	0.06	1206	0.41%***
	Short	957	0.00	252	0.41%

In this strategy we use all the firms in the sample. Long identifies the positions taken in expected positive surprises days; short is the short-selling of a portfolio of firms where negative surprises are expected. Firm quarters are the number of earnings quarters used across all surprise portfolios. %SL is the percentage of triggered stop loss firms. Surprise days is the number of portfolios that were successfully formed over the sample period. The market coefficient is significant for all strategies. All short strategies show HML significance; all long strategies show SMB significance. *Table A16* further reports on the size of the positive and negative coefficients and their significance. The number of surprise days identifies the successfully formed portfolios where at least one firm experienced an earnings event the following day out of all 2994 possible trading days. *10% significance **5% significance ***1% significance.

VI. Analyst Bias or Something Else?

Kleinberg et al. (2018) use machine learning models to understand judges' mistakes and to offer advice on how judges can improve their decision making in trials. Similar to their study, I investigate observed biases and mistakes made by analysts. Machine learning models help us to identify biases by identifying variables that are important in predicting earnings over and above analysts' consensus forecast. The reason why the machine learning model is able to predict earnings surprises, seems to be the result of a few advantages that models have over analyst consensus forecasts.

One argument is that there is enough public information for analysts to improve their predictions, but that analysts are limited by the amount of information they can process. The machine learning model, on the other hand, can process gigabytes of data without much effort. Other limitations include time restrictions; analysts might recognise important signals before the announcement but do not revise their forecast in time. Studies on analyst inefficiencies show that analysts have a tendency not to update their forecast before earnings announcements even when important revelations have been made (Trueman, 1990). In effect, analysts are limited in resources and have bounded rationality that prohibits them from making suitable forecasts.

Following from the above example, analysts may not want to update their forecast due to conflicts of interest or other biases. A significant body of research reports a large number of biases and conflicts experienced by analysts (Bagnoli, Beneish, & Watts, 1999; Bhattacharya, Sheikh, & Thiagarajan, 2006; Ramnath, Rock, & Shane, 2008). Analysts may purposefully lower their forecasts compared to their actual expectations to keep management content at the expense of forecasting accuracy. Chan, Karceski, and Lakonishok (2007) show that analysts, at least in the years before the dot-com bubble, have had a desire to win investment banking clients, which creates a conflict of interest whereby analysts strategically adjust forecasts to avoid earnings disappointments. The advantage of machine learning models is that they look past the majority of these biases and correct for this systematic pessimism of analysts. A summary of the research and variables that were found to be important in predicting earnings over and above analyst forecasts (i.e., biases) has been listed in *Table A15*.

It is possible that analysts have not sufficiently studied the security's behaviour the days before the announcement, as various signals emanating from a rigorous set of timely technical and signal processed pricing data show a high level of importance in predicting

future earnings surprises (*Table 10*). These signals can be representative of the trading behaviour of insiders before announcements or even market-wide trading as a result of management and other noteworthy announcements. One argument is that these signals have gone unnoticed and that analysts do not include a price-related analysis as part of their estimation process. To identify whether the argument is true, we can look at long-term variables, such as the quarterly earnings measures, to see if the analysts sufficiently incorporate long-term earnings factors when making forecasts. If they do not, then it is unlikely to only be an issue of time and resource constraints.

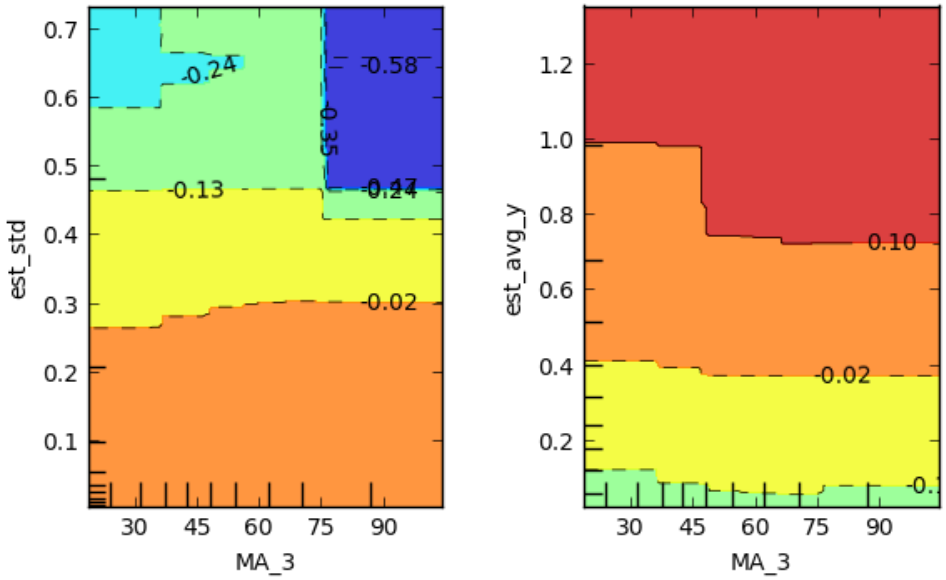
In *Table 9*, it can be seen that the consensus analysts forecast has not sufficiently accounted for the trend of earnings improvement nor changes in the short or long-term earnings ($diff_{-1}, diff_{-4}$). Analysts have also not come to grips with the interesting non-linear relationship between the two, as evidenced by *Figure 7*. Furthermore, analyst estimates do not sufficiently take into account past earnings, p_{-1}, and its interaction with the forecast, est_avg_t, when predicting earnings surprises. Some additional nonlinear relationships show that a large standard deviation among analyst forecasts are associated with lofty analyst consensus forecasts; for example, it is known that analysts show herding behaviour, leading to lower than expected standard deviation (Trueman, 1990). Analyst also do not seem to look at their own errors over time; in fact, the running past analysts forecast errors, $d\_e\_diff_{-4}$, have been shown to be important in predicting the likelihood of surprises. Analysts are very slow to adjust their mind on a company's perceived earnings development.

A growing body of literature has documented the fact that analysts do not fully incorporate or are not motivated to incorporate certain effects in their forecasts, such as the firm's efforts to manage earnings; it is, therefore, possible that these above-mentioned biases fall in the same category (Abarbanell & Lehavy, 2003). A possible outcome of this is that abnormal returns can be realised if the market is unaware of these differences. This is indeed the case; the market does not seem to be fully cognizant of existing analyst biases since abnormal returns can be earned on the predictions of surprises.

The model shows that a low est_std is an indication of herding; as a result, the next analyst would be better off putting more weight on the past earnings, p_{-1} than on the current consensus forecast, est_avg_t, when predicting the target value, p. There are other higher dimensional relationships that are too convoluted to describe and many more that we are not even aware of. As an illustrative example, *Figure 9* presents two difficult price and earnings

relationship at a higher dimension with the purpose of predicting the level of earnings as opposed to occurrence of an earnings surprise.

Figure 9: Partial Dependence of EPS Value on Firm Feature Combinations



Both the est_std and est_avg_t display a nonlinear relationship with *MA_3 (Short for MA_3__max_langevin_fixed_point__m_3__r_30).* To get an idea of the complexity, on top of the above relationship, est_std and est_avg_t also display a strong linear relationship between them as is noted by the 2nd chart in *Figure A15.*

Looking at the plot on the left, when both the standard deviation, *est_std*, and a rolling moving average of the last three days, transformed to fit and find the maximum fixed point on a Brownian motion function, i.e., a *MA_3__max_langevin_fixed_point__m_3__r_30,* are large in value, then the predicted EPS becomes much less than if only one of these variables were large. It can be argued that the combination of these two values provides a valuable measure of market uncertainty, promoting the adjustment of the predicted EPS.

The point of the above description is to show that, as humans, we have bounded rationality and struggle to understand certain relationships, especially relationships in higher dimensions. Thus, without advanced methodologies to uncover these seemingly intractable relationships, analysts would never know how to change their forecasts to become less error-prone, even when they are reported in the literature. It is likely that a large proportion of analysts do not use non-linear techniques to track their biases and that they prefer to make use of models or methodologies that make intuitive sense and are driven by firm policies; and for that reason alone, analysts may be performing worse than the machine learning model.

Although it is hard to test, it is also possible that analyst concern themselves with irrelevant information when making decisions, such as news hype or any other information unobservable to the model, which can lead to worse predictions.

A final possibility is that once analyst forecasts are made public, the target firm may manage earnings upwards in an attempt to earn a small positive surprise (Burgstahler & Dichev, 1997; Burgstahler & Eames, 2006). This argument is difficult to disprove, but this effect is unlikely to explain the significant abnormal profits that can be earned. If analysts' predictions improve over time, while consistently underpredicting earnings, it could be a sign that the firm manages earnings based on public forecasts. See *Table A15* for a summary of which biases identified by past research also show strong predictive power in this study's machine learning models.

VII. Conclusion

A machine learning model with earnings and price variables as inputs performs significantly better at predicting earnings surprises than a random choice benchmark. Surprises that deviate 15% or more can be predicted with 71% accuracy. Exploiting this predictability allows for the construction of profitable trading opportunities. The explanation for this improved performance seems to be three-fold: (1) even though there is enough public information for analysts to improve their predictions, analysts experience an information overload that does not affect machine learning models; (2) the machine learning model corrects for known unobservable biases often experienced by analysts, as evidenced by the list of important variables; (3) the model picks up inside or suspicious trading behaviour by investigating a rigorous set of timely technical and signal-processed variables derived from pricing and volume data.

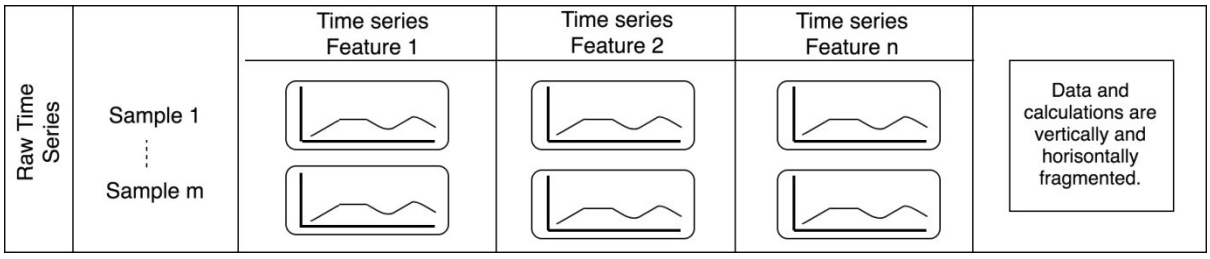
Future research should look at incorporating additional task-relevant variables to improve the prediction model. It is possible that a wider range of fundamental, sentiment, and descriptor variables will further enhance the performance of the current model. Future studies can also attempt to improve the predictions by introducing online machine learning, which in essence means the updating and/or retraining of machine learning models as soon as new data is made available. In finance, new data is created by the second, so one would have to weigh up the performance gain versus the cost of prediction. In the future, the performance can also be improved by developing stacked models where multiple models contribute to the final prediction.

VIII. Appendix

Method A 1: Signal Processing and Feature Selection

The technical indicators that I incorporate in the study include well-known and lesser-known indicators. These technical variables are essentially some decomposition of momentum indicators that quantifies the relationship between recent price changes in a given window and the long-term trend of the instrument. All indicators make use of historical data and are recalculated on a rolling basis for 30 days. Since some of these variables range many orders of magnitude, I will normalise certain variables with their mean and standard deviation merely for data compression reasons. Multiple studies show that the variable space should not just consist of technical indicators and that it can be improved by incorporating other values that are likely to be uncorrelated with the price variables (Dhar & Chou, 2001; Hellstrom & Holmstrom, 1998). On the fundamental front, this study will look at past quarters' reported earnings information such as the EPS forecast, the count, and the standard deviation of analyst forecasts. The next step is to merge all the pricing variables and to present them in a columnar format with exactly 30 trading days' worth of data before all announcements to easily compute the signal processed transformations and create a new set of variables for model inputs.

Figure A10: Columnar Time-series Format

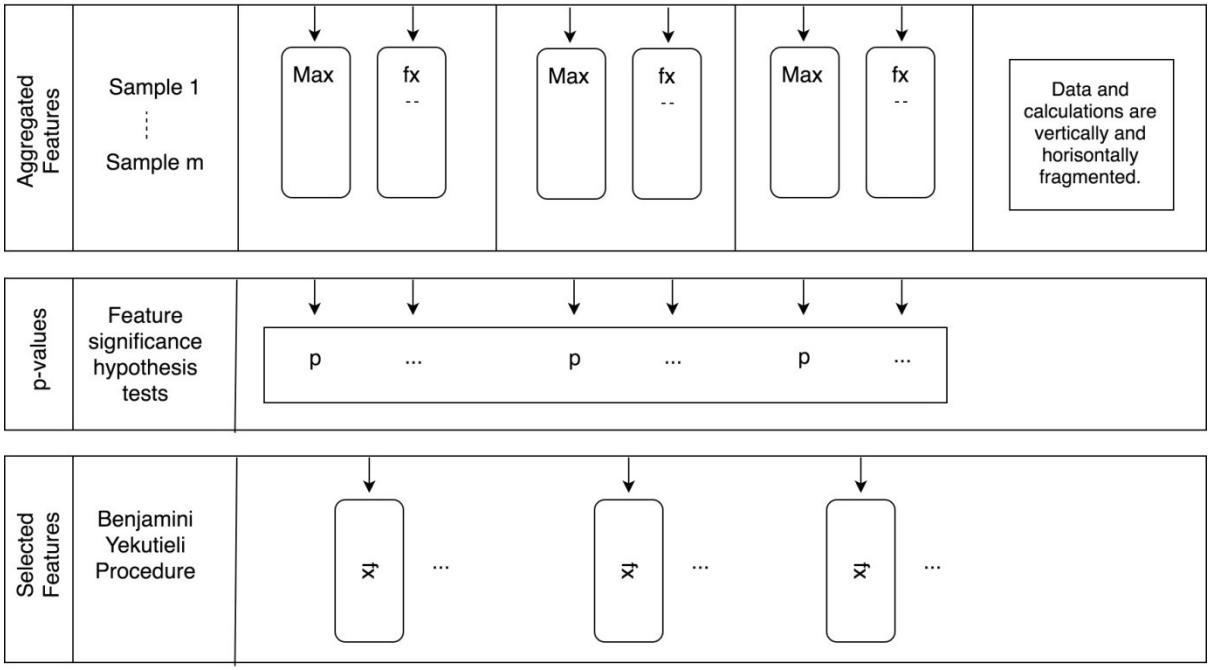


Items (1) and (2) *Figure 1* are put in columnar format, as above, for every firm quarter in the sample period. As shown in *Figure 2*, only data 30 days prior to the official announcement is incorporated. Each time-series constitutes either a series of normal (1) or technical indicators (2).

Once all the price variables have been gathered or computed and are in raw columnar format, then as *Figure A11* illustrates, functions get applied over these chunks of raw data. A final step is the test of significance. This process decides whether or not a newly transformed variable will be kept after observing its effectiveness at predicting the response variable. A p-value of 0.05 has been selected as the significance level. I further use a two-stage solution

that identifies the relevant variables. It is an efficient algorithm that filters the variables in an early stage of the machine learning process with respect to their significance to the classification task, while controlling the expected percentage of selected but non-relevant variables.

Figure A11: Signal Processing Transformations and Feature Selection



The next step is to run all the functions as listed in *Table A14* in which we calculate the significance of each signal value in predicting the target and select the variables based on the Benjamini-Yekutieli Procedure. This is the last step in this variable creation and selection process.

In the first step, all aggregated variables are separately and independently evaluated with respect to their significance for predicting the response variable under investigation using a univariate test. The result of these tests is a vector of p-values. This then quantifies the significance of each variable for predicting the target (response). In *Figure A11* below, this corresponds to the change from aggregated variables to p-values. The vector of p-values is then evaluated on the basis of the Benjamini-Yekutieli-procedure to decide which variables to keep. This is simply a multiple testing procedure that decides which variables to keep and which to cut off based solely on the use of the p-values. This test controls the false discovery rate, which is the ratio of false rejections by all rejections:

$$FDR = \mathbb{E}\left[\frac{|false\ rejections|}{|all\ rejections|}\right] \quad (7)$$

This means that the percentage of irrelevant variables among the extracted variables will be asymptotically controlled by the filtering. This study essentially makes use of classical statistical methods to select variables. This process is unique and better than other variable selection algorithms, such as Boruta, which do not give any insights into how many good or bad variables they filter out (Kursa & Rudnicki, 2010). Often these algorithms 'just work.' Lastly, the process followed in this study is highly scalable: its calculations and the data can be distributed over a cluster and can be performed in parallel.

Once all of the variables have been successfully sculpted and selected, a separate round of variable selection is performed at two critical points, once before executing the classification model and once before executing the model. These procedures incorporate all variables calculated up to this point. The aim of the procedures is simply to separate all the relevant factors from the irrelevant. Feature selection has many benefits; it decreases the computational time, often increases the accuracy of the model, and also simplifies the model to make it easier to understand (Liu & Motoda, 2012).

The second variable selection procedure occurs after running all variables through multiple models to prompt the variable importance values. This technique is different from the first technique and is based on information gain. The approach I have used to uncover the most prominent variables is a tree-based variable selection procedure. I made use of a combination of random forest, gradient boosting, and AdaBoost variable selection procedures. All the variables are calculated as a line vector and aggregated together in a matrix labelled 'variables.' These variables are individually tested by each algorithm on a hold-out set and the relevance scores are summed together after which the top 800 variables are selected for the classification task.

Method A 2: Classifier Learning

To create the overall ensemble model, such as presented by the *Classifier* pseudocode in the methodology section, IV.C.2, we have to establish a loss function, *L* to minimise, so as to optimise the structure and performance of the model. This function has to be differentiable as we want to perform a process of steepest descent, which is an iterative process of

attempting to reach the global minimum of a loss function by going down the slope until there is no more room to move closer to the minimum. We, therefore, solve for by minimizing a loss function numerically via the process of steepest descent. The focus here is on $f(\boldsymbol{x}_i)$ as this is the compressed form of the predictor of each tree *i.*

For our classification task, we use logistic regression to obtain the probabilistic outputs of the target variable.

$$L(\theta) = \sum_i [\boldsymbol{y}_i \, ln(1 + e^{f(x_i)}) + (1 - \boldsymbol{y}_i) ln(1 + e^{f(x_i)})] \tag{8}$$

$$L(\theta) = \sum_i [\boldsymbol{y}_i \, ln(1 + e^{-\hat{y}\, i}) + (1 - \boldsymbol{y}_i) ln(1 + e^{\hat{y}\, i})] \tag{9}$$

Further, it is necessary to minimise the loss over all the points in the sample, $(\boldsymbol{x}_i, y_i)$:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} L(\theta) \tag{10}$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} L(y_i, f(\boldsymbol{x}_i)) \tag{11}$$

At this point, we are in the position to minimise the predictor function, $f(\boldsymbol{x}_i)$, w.r.t. $\boldsymbol{x}$ since we want a predictor that minimises the total loss of $f(\boldsymbol{x})$. Here, we simply apply the iterative process of steepest descent. The minimisation is done in a few phases. These phases are better described in the next appendix section, *Method A 3*, but a short summary follows. The first process starts with adding the first and then successive trees. Adding a tree emulates adding a gradient-based correction. Making use of trees ensures that the generation of the gradient expression is successful, as we need the gradient for an unseen test point at each iteration, as part of the calculation $f(\boldsymbol{x})$. Finally, this process will return $f(\boldsymbol{x})$ with weighted parameters. The detailed design of the predictor, $f(\boldsymbol{x})$, is outside the purpose of the study, but for more extensive computational workings, see the next section.

Method A 3: Detailed XGBoost Design and Supervised Learning

This part can be skipped if you are already familiar with supervised learning and, more specifically, Gradient Boosted Trees. A large part of the detailed workings has been obtained from the official XGBoost documents but have been altered to improve understanding (Chen & Guestrin, 2016). Supervised learning refers to the mathematical structure describing how to make a prediction $\boldsymbol{y_i}$ given $\boldsymbol{x_i}$. In classification task prediction, $\boldsymbol{y_i}$ is the probability of an earnings surprise event of some specified threshold. The inputs, $\boldsymbol{x_i}$, have been selected based on the applied selection procedures. Apart from the different prediction types, in the classification task, the model gets logistic transformed to obtain a vector of probabilities for each observation and associated categories. In supervised learning, parameters play an important role. The parameters are the undetermined part that we are required to learn using the training data. For example, in a linear univariate regression, $\hat{y}_i = \sum_j \theta_j \boldsymbol{x_{ij}}$, the coefficient θ is the parameter.

The task is ultimately to find the best parameters and to choose a computationally efficient way of doing so. To measure a model's performance, given some parameter selections, we are required to define an objective function. The following is a compressed form of the objective function, $Obj(\Theta) = L(\theta) + \Omega(\Theta)$. In this equation, L is the training loss function; the regularisation term is Ω. The training loss function tests the predictive ability of the model using training data. A commonly used method to calculate the training loss is the mean squared error, $L(\theta) = \sum_i (\boldsymbol{y}_i - \hat{y}_i)^2$. Thus, the parameters get passed into a model that calculates, $\boldsymbol{\hat{y}_i}$, a series of predictions, that gets compared against the actual values in a mean squared error function to calculate the loss.

The regularisation term controls the complexity of the model, which helps to avoid overfitting. The Extreme, X, of the XGBoost model, relates to an extreme form of regularisation that controls for over-fitting, leading to improved performance over other models. There are countless ways to regularise models in essence, we constrain a model by giving it fewer degrees of freedom; for example, to regularise a polynomial model, we can transform the model to reduce the number of polynomial degrees. The tree ensemble can either be a set of classification or a set of regression trees. It is usually the case that one tree is not sufficiently predictive, hence the use of a tree ensemble model that sums the predictions of many trees together. Mathematically, the model can be written in the following form $\hat{y}_i = \sum_{k=1}^{K} f_k(\boldsymbol{x_i}), f_k \in F$. Here, K is the number of trees, and f represents one possible function

from the entire functional space *F*. *F* is a set of all possible classification and regression trees (CARTs). This expression then simply adds multiple models together that lives within the allowable CART function space. Therefore, combining the model, the training loss, and the regularisation function, we can gain our objective function and seek to optimise it. The function can be written as follows, $Obj(\theta) = \sum_i^n l\left(\boldsymbol{y_i},\ \hat{y}_i^{(t)}\right) + \sum_{k=1}^K \Omega(f_i)$. Thus far, the model is similar to that of a random forest, the difference being in how the models are trained.

For the next part, we have to let the trees learn, so for each tree, we have to describe and optimise an objective function. We can start off by assuming the following function, $Obj = \sum_i^n l(y_i,) + \sum_{i=1}^t \Omega(f_i)$. By looking at the function, it is important that we identify the parameters of the trees. We want to learn the functions, f_i, each of which contains a tree structure and associated leaf scores. This is more complex than traditional methods where you can simply take the gradient and optimise for it. Instead, Gradient Boosting uses an additive strategy, whereby we learn to adjust and add an extra tree after each iteration. We write our prediction value at step *t* as $\hat{y}_i^{(t)}$, so that we have $\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(\boldsymbol{x_i}) = \hat{y}_i^{(t-1)} + f_t(\boldsymbol{x_i})$. Then we simply choose the tree that optimises our objective, $Obj^{(t)} = \sum_i^n l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^t \Omega(f_i) = \sum_{i=1}^n l\left(\boldsymbol{y_i}, \hat{y}_i^{(t-1)}\right) + f_t(\boldsymbol{x_i}) + \Omega(f_t) + constant$. By using MSE as the loss function, it becomes $Obj^{(t)} = \sum_{i=1}^n \left[2\left(\hat{y}_i^{(t-1)} - \boldsymbol{y_i}\right) f_t(\boldsymbol{x_i}) + f_t(\boldsymbol{x_i})^2\right] + \Omega(f_t) + constant$. The form of MSE is easy to deal with. The Taylor expansion can simply be taken to the second order. $Obj^{(t)} = \sum_{i=1}^n \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(\boldsymbol{x_i}) + \frac{1}{2} h_j {f_t}^2(\boldsymbol{x_i})\right] + \Omega(f_t) + constant$, where g_i and h_i is defined as, $g_i = \partial_{\hat{y}_i^{(t-1)}} l(\boldsymbol{y_i}, \hat{y}_i^{(t-1)})$, $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(\boldsymbol{y_i}, \hat{y}_i^{(t-1)})$. After all the constants are removed, then the objective at *t* get transformed to, $\sum_{i=1}^n \left[g_i f_t(\boldsymbol{x_i}) + \frac{1}{2} h_j {f_t}^2(\boldsymbol{x_i})\right] + \Omega(f_t)$. This then becomes an adjusted optimization function for the new tree. Although we have looked at the training step, we have not looked at regularisation yet. The next step is to specify how complex the tree should be, $\Omega(f_t)$. To do this we can improve the tree definition to *F(x)*, $f_t(\boldsymbol{x}) = w_{q(\boldsymbol{x})}, w \in \mathbb{R}^T, q: \mathbb{R}^m \rightarrow \{1, 2, \dots, T\}$. Here *w* represents the scores of the leaves presented in vector form and *q* represents a function that assigns each point to the appropriate leaf; lastly *T* denotes how many leafs there are. The complexity can be defined as $a\ \Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^T w_j^2$; there are more ways to formulate and define how complex a model is or should be in practice, but this

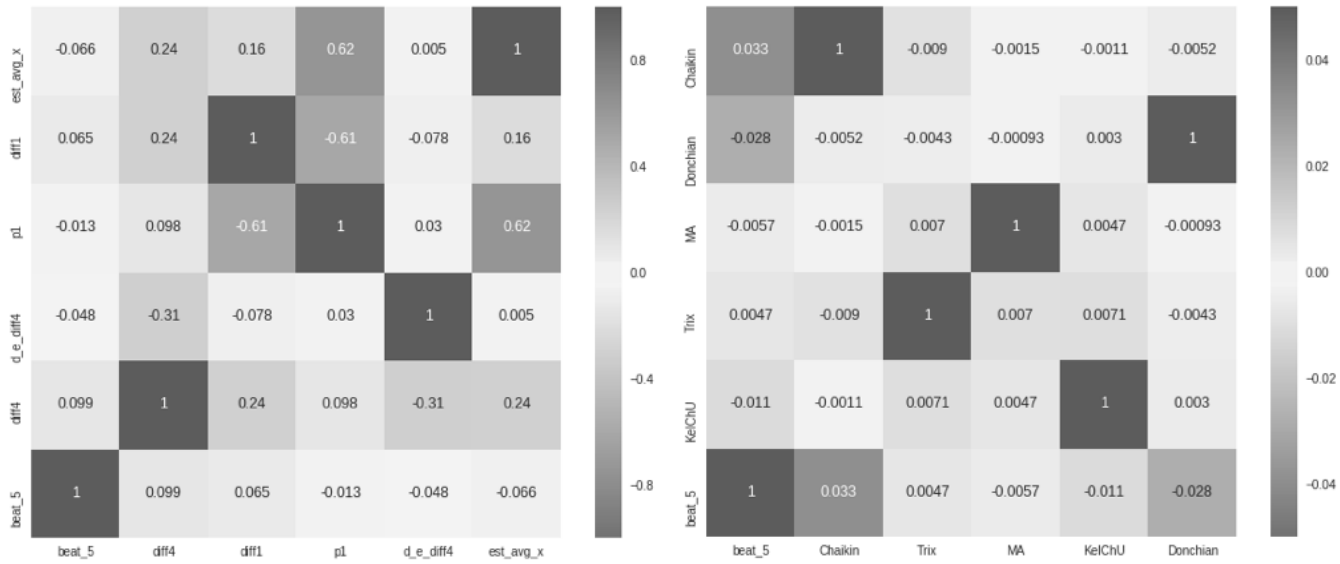
one is quite practical and easy to conceptualise. Once the tree model is described, the objective value w.r.t. the *t-th* tree can be written as follows: $Obj^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(\boldsymbol{x_i}) + \frac{1}{2}h_j {f_t}^2(\boldsymbol{x_i})\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \quad = \quad \sum_{J=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} g_i + \lambda\right)w_j^2\right] + \gamma T$, where $\mathrm{I}_j = \{\mathrm{i}|\mathrm{q}(\mathrm{x}_j) = \mathrm{j}\}$ represents a full set of all the data points as have been assigned to the *j-th* leaf. The equation can then further be compressed by describing $G_j = \sum_{i\in I_j} g_i$ and $H_J = \sum_{i\in I_j} h_i$, then $Obj^{(t)} = \sum_{J=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2\right] + \gamma T$. In the preceding equation the weights w_j are independent w.r.t each other, the form $G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2$ is quadratic, and the best weight for a structure *q(x)* is given by the following expression. $w_j^* = -\frac{G_j}{H_j+\lambda}$, $obj^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$. This equation measures how good a tree structure $q(x)$ is. A lower score is better for the ultimate structure of a tree. Now that we know how to measure the fittingness of a tree, we can identify all the trees and select the best one. It is, however, not possible to approach it this way and instead has to be done for one depth level of a tree at a time. This can be approached by splitting a leaf into two sections and then recording its gain. The following equation represents this process, $Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_l+G_l)^2}{H_L+H_R+\lambda}\right] - \gamma$. If the gain obtained is equal to or smaller than γ, then it would be better if we do not add the branch to the tree; this is often referred to as the pruning technique. We basically search for the ultimate split; if all instances are sorted in order, we simply scan left to right to sufficiently calculate the structure scores of all possible solutions and then identify the most efficient split.

Method A 4: Partial Dependence (D)

For $\boldsymbol{x}_j$, a variable from a vector of variables, sort the unique values $\mathrm{V} = \{\mathbf{x}_\mathrm{j}\}_\mathrm{i} \in \{1,\ldots,\mathrm{n}\}$ resulting in V^*, where $V^* = K$. Create K new matrices $\boldsymbol{X}^k = \left(\boldsymbol{x}_j = V_k^*, \boldsymbol{X}_{-j}\right), \forall \mathrm{k} = (1,\ldots,K)$. Then drop each of the K new datasets, $\boldsymbol{X}^k$**,** down the models' fitted trees predicting a new value for each observation in all *k* datasets: $\widehat{\boldsymbol{y}}^k = \widehat{f(\boldsymbol{X}^k)}, \forall \mathrm{k} = (1,\ldots,K)$. Then average the prediction in each of the K datasets, $\hat{y}_k^* = \frac{1}{n}\sum_{i=1}^{N}\widehat{\boldsymbol{y}}_\mathrm{i}^k, \forall \mathrm{k} = (1,\ldots,K)$. Lastly, simply plot V^* against $\widehat{\boldsymbol{y}}^*$ to visualise the relationship. The above strategy shows the dependence of the target function on a set of target variables by marginalising over the values of other variables.

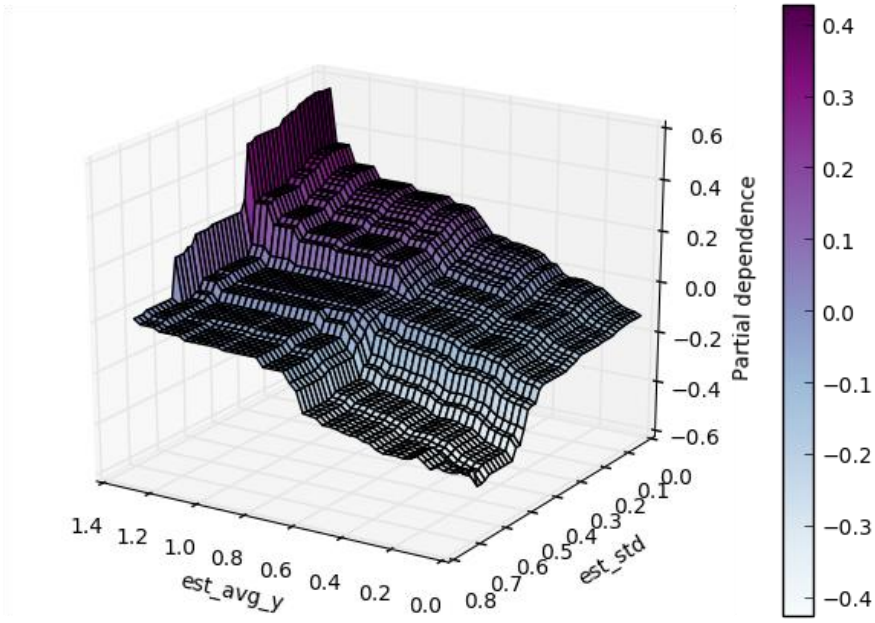
In more simple terms, the above strategy shows the dependence between the target function and a set of target variables by marginalising over the values of other variables

Figure A12: Classification Correlation Matrix for Earnings and Price Variables



This is an example of the correlation matrices for the top variables in the classification model. The left side represents the earnings-related variables and the right side the price-related variables. The purpose of this graph is to show that there is multicollinearity between the variables. This result makes us more cautious about assigning variable significance or importance; the reason is that multiple variables may represent the same dynamic.

Figure A13: Assorted Interaction Charts



This is a three-dimensional way of presenting the interaction of the analyst forecast and standard deviation and the resulting response output using the partial dependence method. This graph clearly shows that to achieve the most accurate EPS prediction, the forecast should be readjusted downward as increased uncertainty, proxied by the standard deviation of forecasts, leads to excessively positive forecasts.

Figure A14: Partial Dependence Classification - Earnings Related

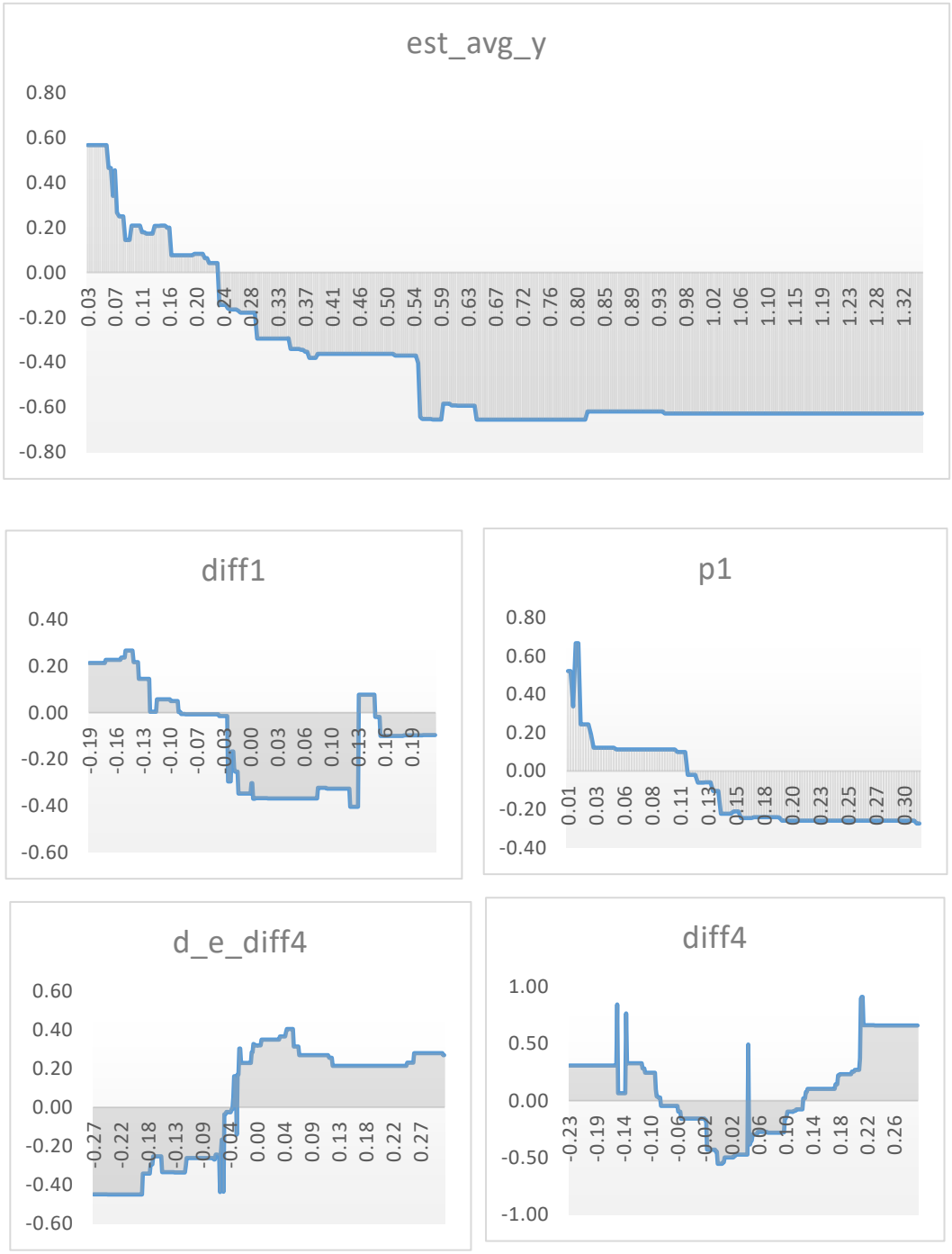


Figure A15: Partial Dependence Classification - Price Related

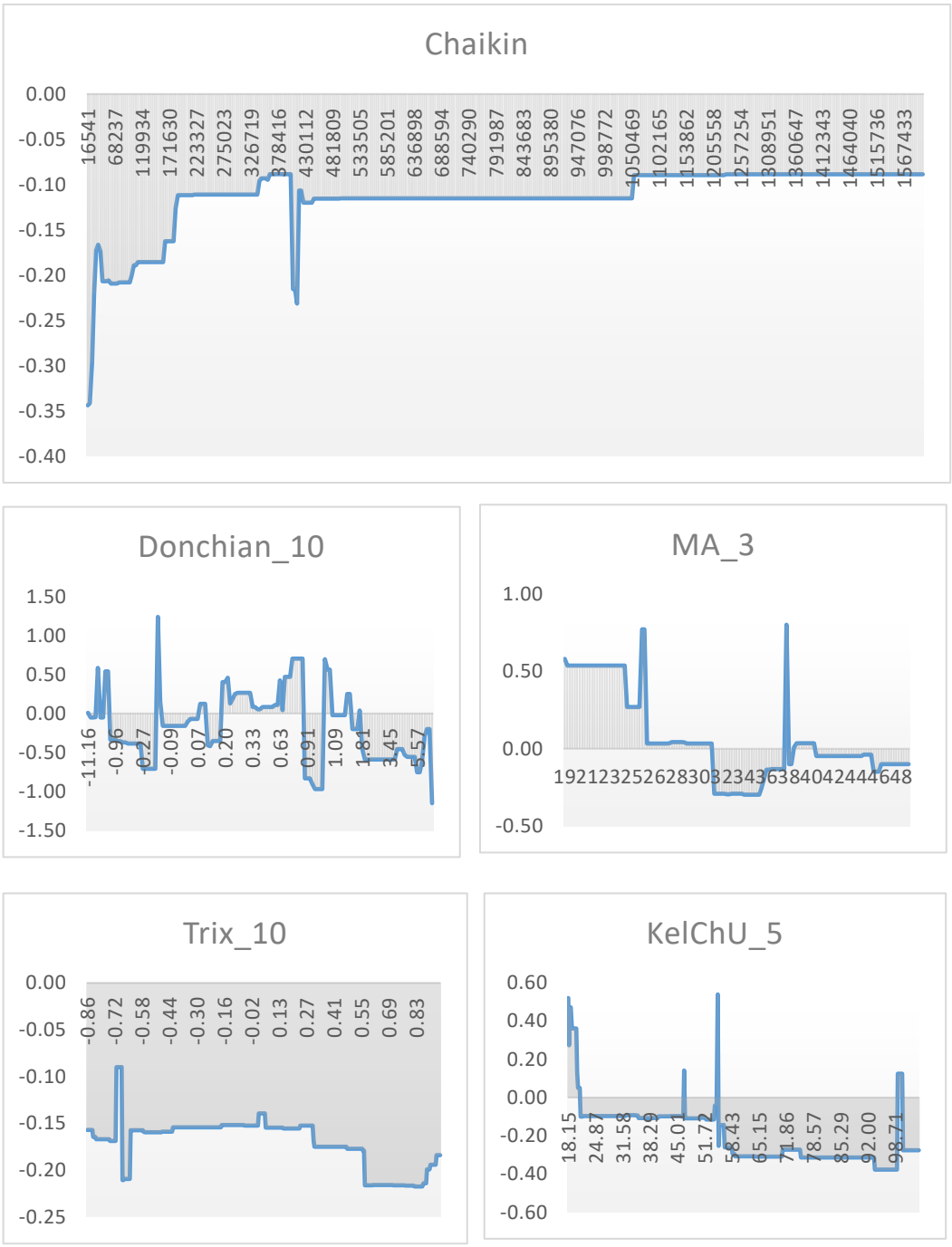


Table A13: Machine Learning for Finance Glossary

Accuracy	The rate of correct predictions made by the model over a data set. Accuracy is usually forecasted by using an independent test set that was not used at any time during the learning process.
Boosting	A technique for combining models based on adaptive resampling - where different data is given to different models. The idea is to successively omit the 'easy' data points, which are already well modelled, so that the later models focus on the left-over 'hard' data points.
Class	The same as category.
Classifier	A mapping from unlabelled instances to classes. Classifiers have a form (e.g., decision tree) plus an interpretation procedure. Some classifiers also provide probability forecasts (scores). Classifiers are used to predict class labels.
Confusion matrix	A matrix showing the predicted and actual classifications. A confusion matrix is of size LxL, where L is the number of different label values.
Cross-validation	A method for estimating the accuracy of an inducer by dividing the data into *k* mutually exclusive subsets of approximately equal size. The inducer is trained and tested *k* times. Each time, it is trained on the data set minus a fold and tested on that fold. The accuracy forecast is the average accuracy for the *k* folds.
Feature selection	The process of removing variables which seem irrelevant for modelling.
Variables	Explanatory and independent variables. It is the measurements and characteristics that represent the data.
Instance	Observations, ex. firm quarters. A single object of the world from which a model will be learned, or on which a model will be used (e.g., for prediction).
Machine learning	The field of scientific study that concentrates on inductive algorithms that can be said to 'learn.'

Mapping	Applying a function to all elements of a list in order and returning a list of results.
Model	Most inductive algorithms generate models that can then be used as Classifiers or Regressors.
Overfitting	A modelling error that occurs when a function is too closely fit to a limited set of data points leading to poor out-of-sample generalisation.
Regression	Predicting the value of random variable *y* from measurement *x*. For example, predicting EPS based on Estimates, Size and P/E. Regression is used to predict continuous values. It does not have the same meaning as in finance; it is concerned with using *regression* to determine the strength of relationships between dependent and independent variables.
Regressor	A mapping from unlabelled instances to a value within a predefined metric space, e.g., a continuous range.
Regularization	Any estimation technique designed to impose a prior assumption of 'smoothness' on the fitted function. Use to alleviate overfitting and model complexity.
Signal Processing	Concerns the analysis, synthesis, and modification of signals, which are broadly defined as functions conveying 'information about the behaviour or attributes of some phenomenon.'
Supervised learning	Techniques used to learn the relationship between independent attributes and a designated dependent attribute (the label). Most induction algorithms fall into the supervised learning category
Target Variable	Response variable, Dependent variable. It is the variable being predicted in supervised learning, whether it be categorical or continuous.

Table A14: Signal Processing and Other Functions

Name and Parameter	**Description**
abs_energy(x)	Returns the absolute energy of the time-series, which is the sum over the squared values.
absolute_sum_of_changes(x)	Returns the sum over the absolute value of consecutive changes in the series *x*.
acf(x[, unbiased, nlags, qstat, fft, alpha])	Autocorrelation function for 1d arrays.
adfuller(x[, maxlag, regression, autolag])	Augmented Dickey-Fuller unit root test.

agg_autocorrelation(x, param)	Calculates the value of an aggregation function.
agg_linear_trend(x, param)	Calculates a linear least-squares regression for values of the time-series that were aggregated over chunks versus the sequence from 0 up to the number of chunks minus one.
approximate_entropy(x, m, r)	Implements a vectorised approximate entropy algorithm.
ar_coefficient(x, param)	This variable calculator fits the unconditional maximum likelihood of an autoregressive $AR(k)$ process.
augmented_dickey_fuller(x, param)	The Augmented Dickey-Fuller test is a hypothesis test that checks whether a unit root is present in a time-series sample.
autocorrelation(x, lag)	Calculates the autocorrelation of the specified lag.
binned_entropy(x, max_bins)	First bins the values of *x* into equidistant bins.
c3(x, lag)	This function calculates the value of nonlinearity in time-series (Schreiber, 1997).
change_quantiles(x, ql, qh, isabs, f_agg)	First fixes a corridor given by the quantiles ql and qh of the distribution of *x*.
count_above_mean(x)	Returns the number of values in *x* that are higher than the mean of *x*.
count_below_mean(x)	Returns the number of values in *x* that are lower than the mean of *x*.
cwt(data, wavelet, widths)	Continuous Wavelet Transform.
cwt_coefficients(x, param)	Calculates a Continuous Wavelet Transform for the Ricker wavelet, also known as the "Mexican hat wavelet."
energy_ratio_by_chunks(x, param)	Calculates the sum of squares of chunk *i* out of *N* chunks expressed as a ratio with the sum of squares over the whole series.
fft_coefficient(x, param)	Calculates the Fourier coefficients of the one-dimensional discrete Fourier Transform.
find_peaks_cwt(vector, widths[wavelet])	Attempts to find the peaks in a 1-D array.
first_location_of_maximum(x)	Returns the first location of the maximum value of x.
first_location_of_minimum(x)	Returns the first location of the minimal value of x.
friedrich_coefficients(x, param)	Coefficients of polynomial, which has been fitted to the deterministic dynamics of Langevin model.
has_duplicate(x)	Checks if any value in *x* occurs more than once.
has_duplicate_max(x)	Checks if the maximum value of *x* is observed more than once.
has_duplicate_min(x)	Checks if the minimal value of *x* is observed

	more than once.
index_mass_quantile(x, param)	Calculates the relative index *i* where *q%* of the mass of the time-series *x* lie left of *i*.
kurtosis(x)	Returns the kurtosis of *x* (calculated with the adjusted Fisher-Pearson standardized moment coefficient *G2*).
large_standard_deviation(x, r)	Boolean variable denoting if the standard dev of *x* is higher than '*r*' times the range = difference between max and min of *x*.
last_location_of_maximum(x)	Returns the relative last location of the maximum value of *x*.
last_location_of_minimum(x)	Returns the last location of the minimal value of *x*.
length(x)	Returns the length of *x*.
linear_trend(x, param)	Calculates a linear least-squares regression for the values of the time-series versus the sequence from 0 to length of the time-series minus one.
linregress(x[, y])	Calculates a linear least-squares regression for two sets of measurements.
longest_strike_above_mean(x)	Returns the length of the longest consecutive subsequence in *x* that is bigger than the mean of *x*
longest_strike_below_mean(x)	Returns the length of the longest consecutive subsequence in *x* that is smaller than the mean of *x*.
max_langevin_fixed_point(x, r, m)	Largest fixed point of Langevin dynamics forecasted from polynomial.
maximum(x)	Calculates the highest value of the time-series *x*.
mean(x)	Returns the mean of *x*.
mean_abs_change(x)	Returns the mean over the absolute differences between subsequent time-series values.
mean_change(x)	Returns the mean over the differences between subsequent time-series values.
mean_second_derivate_central(x)	Returns the mean value of a central approximation of the second derivative.
median(x)	Returns the median of *x*.
minimum(x)	Calculates the lowest value of the time-series *x*.
number_crossing_m(x, m)	Calculates the number of crossings of x on m.
number_cwt_peaks(x, n)	This variable searches for different peaks in x.
number_peaks(x, n)	Calculates the number of peaks of at least support n in the time-series x.
pacf(x[, nlags, method, alpha])	Partial autocorrelation forecasted.
partial_autocorrelation(x, param)	Calculates the value of the partial

	autocorrelation function at the given lag.
percentage_of_reoccurring_datapoints_to_all_datapoints(x)	Returns the percentage of unique values that are present in the time-series more than once.
percentage_of_reoccurring_values_to_all_values(x)	Returns the ratio of unique values that are present in the time-series more than once.
quantile(x, q)	Calculates the *q* quantile of *x*.
range_count(x, min, max)	Count observed values within the interval [*min, max*).
ratio_beyond_r_sigma(x, r)	Ratio of values that are more than $r \times std(x)$ away from the mean of *x*.
ratio_value_number_to_time_series_length(x)	Returns a factor which is 1 if all values in the time-series occur only once, and below one if this is not the case.
ricker(points, a)	Returns a Ricker wavelet, also known as the "Mexican hat wavelet."
sample_entropy(x)	Calculates and returns sample entropy of *x*.
set_property(key, value)	Returns a decorator that sets the property key of the function to value.
skewness(x)	Returns the sample skewness of *x* (calculated with the adjusted Fisher-Pearson standardized moment coefficient *G1*).
spkt_welch_density(x, param)	Variable calculator that forecasts the cross power spectral density of the time-series *x* at different frequencies.
standard_deviation(x)	Returns the standard deviation of *x*.
sum_of_reoccurring_data_points(x)	Returns the sum of all data points that are present in the time-series more than once.
sum_of_reoccurring_values(x)	Returns the sum of all values that are present in the time-series more than once.
sum_values(x)	Calculates the sum over the time-series values.
symmetry_looking(x, param)	Boolean variable denoting if the distribution of *x* looks symmetric.
time_reversal_asymmetry_statistic(x, lag)	This function calculates the value of a complex function shown to be promising variable to extract (Fulcher, Jones, 2014).
value_count(x, value)	Counts occurrences of value in time-series *x*.
variance(x)	Returns the variance of *x*.
variance_larger_than_standard_deviation(x)	Boolean variable denoting if the variance of *x* is greater than its standard deviation.
welch(x[, fs, window, nperseg, noverlap])	Estimates power spectral density using Welch's method.

Table A15: Variables Created Based on Past Literature and Their Appearance In Top 5 Feature Categories for Both The Classification and Regression Task

Reference	**Bias/Observation**	**Illuminating Feature**	**Top 10**
Beaver, 1968; Givoly and Lakonishok, 1984	To identify systematic over and under-prediction.	Multiple running forecast errors.	Yes
Latane and Jones, 1977; Brown et al., 1996	Trading strategy observing that the past difference between forecasts and surprises persists.	Rolling averages of past differences.	Yes
Brown et al., 1996	Prior stock-returns is a significant factor in a Multi-factor regression strategy.	Prior stock returns.	Yes
Lys & Sohn, 1990; Trueman, 1990	Stale forecasts.	Days since last forecast.	-
Butler and Lang, 1991	Skewness in analyst consensus.	The skewness of past forecasts and observed EPS	-
Freeman and Tse, 1992; Das, Levine, and Sivaramakrishnan, 1998	Unregular forecast dispersion between analyst forecasts; analysts more optimistic in times of uncertainty.	A running standard deviation on analysts' forecasts.	Yes
Brown et al., 1996	Earnings surprises tend to repeat.	Dummy to identify past EPS surprise occurrences and threshold levels.	-
Das, Levine, and Sivaramakrishnan, 1998	Analysts more optimistic with increased uncertainty.	The running total count of null values over a range of earnings inputs.	-
Brown, 2001	Median earnings surprise shifts over the years.	Observed and forecasted EPS median.	-
Kinney, Burgstahler, & Martin, 2002	Identified the importance of analyst coverage.	Count of individual analysts per quarter observation.	-
Barron et al. 2005	Insider trading is related to the level of trading volume before announcement dates.	Volume.	-
Anilowski, Feng, and Skinner, 2007; Lys & Soo, 1995	Earnings pre-announcements.	Signals over pricing data.	Yes
Johnson and Zhao, 2012	Surprises tend to reappear in the long run.	Count of past surprises per classification threshold.	-

Table A16: 5 Factor Model Coefficients and Significance for a Stop Loss Surprise Strategy on All Firms

Threshold	Surprise	Intercept	MktRf	SMB	HML	RMW	CMA
5%+	Positive	0.258	0.930	0.978	0.414	0.291	-0.014
		(3.118)	(11.875)	(6.411)	(2.821)	(1.084)	(-0.046)
	Negative	-0.621	0.950	0.238	1.242	-0.124	-0.985
		(-1.988)	(3.426)	(0.465)	(2.23)	(-0.131)	(-0.921)
10%+	Positive	0.317	1.070	0.795	0.189	-0.177	0.265
		(2.769)	(9.902)	(3.762)	(0.907)	(-0.483)	(0.652)
	Negative	-0.086	0.690	0.764	1.834	-1.837	-0.721
		(-0.271)	(2.461)	(1.441)	(3.419)	(-1.858)	(-0.653)
15%+	Positive	0.406	0.000	0.000	0.000	0.000	0.000
		(3.086)	(10.954)	(2.928)	(-0.711)	(0.081)	(2.409)
	Negative	-0.408	0.670	1.763	1.896	-1.053	-1.681
		(-1.187)	(2.237)	(3.028)	(3.396)	(-1.07)	(-1.437)

MktRf is the difference between R_{Mt}, the value-weighted return of the market portfolio and R_{Ft}, the risk-free rate. SMB_t, HML_t, RMW_t and CMA_t are the respective differences between diversified portfolios of small stocks and big stocks, high and low B/M stocks, robust and weak profitability stocks and low and high investment stocks. To perform the regressions, the respective daily values were obtained from Kenneth French's website. The market coefficient is significant for all strategies. All short strategies show HML significance; all long strategies show SMB significance. The regressions on the factors showed that the short strategy showed some significance at the 5% threshold. This strategy includes stop loss limits, robust to slippage. Unlike the strategy in *Table A17*, this portfolio only trades on days where surprises are expected to occur and does not substitute non-trading days by shifting the capital to a market portfolio. The coefficient size of the alpha (intercept) progressively increases with each threshold increase, although the same cannot be said for the significance scores, largely as a result of smaller samples at the higher thresholds and higher weightings on alternative coefficients.

Table A17: 5-Factor Model Coefficients and Significance for a Large Firm Surprise and Market Portfolio Strategy

Threshold	Surprise	Intercept	MktRf	SMB	HML	RMW	CMA
5%+	Positive	0.044	1.005	0.067	0.080	-0.087	-0.090
		(1.553)	(36.065)	(1.233)	(1.511)	(-0.899)	(-0.861)
	Negative	-0.002	1.010	0.083	0.012	0.075	0.181
		(-0.149)	(77.873)	(3.262)	(0.47)	(1.647)	(3.692)
10%+	Positive	0.032	1.007	0.094	0.130	-0.059	-0.040
		(1.221)	(40.146)	(1.914)	(2.703)	(-0.671)	(-0.424)
	Negative	-0.005	1.013	0.069	0.035	0.047	0.146
		(-0.395)	(81.976)	(2.841)	(1.46)	(1.086)	(3.122)
15%+	Positive	0.037	0.998	0.002	0.095	-0.026	-0.032
		(1.979)	(56.22)	(0.051)	(2.807)	(-0.421)	(-0.472)
	Negative	-0.018	1.009	0.098	0.048	0.069	0.125
		(-1.189)	(68.706)	(3.413)	(1.694)	(1.341)	(2.258)

MktRf is the difference between R_{Mt}, the value-weighted return of the market portfolio, and R_{Ft} the risk-free rate. SMB_t, HML_t, RMW_t and CMA_t are the respective differences between diversified portfolios of small stocks and big stocks, high and low B/M stocks, robust and weak profitability stocks, and low and high investment stocks. The following table tracks the performance of a strategy formed over a sample period consisting of 2994 trading days. All strategies showed large significance for the market coefficient, the reason being that a significant part of the portfolio days are formed by taking a position in the market. For all the long strategies, HML showed positive significance apart for the 5% long thresholds. For all the short strategies, SMB and CMA showed positive significance. Unlike *Table A16,* each strategy has the same number of portfolio days; the reason is that the market substitutes for the days where no expected earnings surprises are predicted to occur. But the same portfolio return as presented by *Table A16* is also included in this strategy. The alphas with this strategy are somewhat low, mostly as a result of the returns fitting strongly on value firms; however, the plotted cumulative returns still show economic significance over the sample period. Further tests showed improved significance when stop-loss triggers were incorporated into the strategy.

Table A18: Full Feature-Mapper Combination for top Signal Processed Variables

Base Feature	Mapper	Parameters	Full Name
Chaikin	mean_abs_change_quantiles	qh_1.0__ql_0.4	Chaikin__mean_abs_change_quantiles__qh_1.0__ql_0.4
Donchian_10	friedrich_coefficients	m_3__r_30__coeff_2	Donchian_10__friedrich_coefficients__m_3__r_30__coeff_2
MA_3	max_langevin_fixed_point	m_3__r_30	MA_3__max_langevin_fixed_point__m_3__r_30
Trix_10	Autocorrelation	lag_8	Trix_10__autocorrelation__lag_8
KelChU_5	max_langevin_fixed_point	m_3__r_30	KelChU_5__max_langevin_fixed_point__m_3__r_30
BollingerB_5	mean_abs_change_quantiles	qh_0.8__ql_0.6	BollingerB_5__mean_abs_change_quantiles__qh_0.8__ql_0.6
Momentum_10	friedrich_coefficients	m_3__r_30__coeff_1	Momentum_10__friedrich_coefficients__m_3__r_30__coeff_1
Bollinger%b_5	friedrich_coefficients	m_3__r_30__coeff_3	Bollinger%b_5__friedrich_coefficients__m_3__r_30__coeff_3
RSI_5	friedrich_coefficients	m_3__r_30__coeff_2	RSI_5__friedrich_coefficients__m_3__r_30__coeff_2
Donchian_8	autocorrelation	lag_4	Donchian_8__autocorrelation__lag_4
MA_3	max_langevin_fixed_point	m_3__r_30	MA_3__max_langevin_fixed_point__m_3__r_30
Chaikin	mean_abs_change_quantiles	__qh_1.0__ql_0.4	Chaikin__mean_abs_change_quantiles__qh_1.0__ql...
TSI_3_2	cwt_coefficients	widths_(2, 5, 10, 2)	TSI_3_2__cwt_coefficients__widths_(2, 5, 10, 2)
EMA_3	cwt_coefficients	EMA_3__cwt_coefficients__widths_(2, 5, 10, 20)	EMA_3__cwt_coefficients__widths_(2, 5, 10, 20)
ADX_4_3	fft_coefficient	coeff_6	ADX_4_3__fft_coefficient__coeff_6

Investigating Accounting Patterns for Bankruptcy and Filing Outcome Prediction using Machine Learning Models

Abstract

I study the use of non-linear models and accounting inputs to predict the occurrence of litigated bankruptcies and their associated filing outcomes. The main purpose of this study is to identify the accounting patterns associated with bankruptcies. The filing outcomes include, among others, how long the bankruptcy process will endure, whether the firm will successfully emerge after the bankruptcy period, whether the bankruptcy is tortious, and whether it will involve an asset sale. The study highlights the importance of previously unidentified accounting variables that are useful in predicting bankruptcies and bankruptcy outcomes. The study categorises predictor variables in accounting dimensions to empirically identify the importance of each dimension to the prediction tasks. The high dimensionality of the gradient boosting machine allows us to identify and explain the nonlinear interactions between a wide range of variables.

I. Introduction and Motivation

This study makes use of a modern gradient boosting machine (GBM), XGBoost, to predict litigated bankruptcies and filing outcomes. A GBM sequentially builds multiple decision tree models from which the final outcome is predicted. To ensure that the best model is used to investigate the variable importance scores, I compare the GBM with four state-of-the-art non-linear models and a Logit model. The overall GBM model predicts bankruptcy with an accuracy of 97% and an ROC AUC[25] of close to 96% compared to the 69% accuracy and 71% ROC (AUC) of a standard Logit Model. The selected models use a wide spectrum of dollar accounting values and ratios as inputs, including price ratios embraced under the 'valuation' category. Consistent with past research, this study reports the ROC (AUC) score, accuracy, cross-entropy, and error rates associated with the performance of the prediction model. Further analysis also includes the use of a confusion matrix.

The models used in this study are different from parametric models and do not rely on significance tests; instead, they rely on a data-centric approach that looks at the predictive ability of a parameter based on the variable selection and ranking in the nodes of the trees. In this chapter, I argue that GBMs provide for an improved analysis of accounting and associated bankruptcy patterns compared to that of linear models. First, these models empirically report on the non-linear relationships of variables. This is an important attribute as financial data is likely to exhibit non-linearities. GBMs do not require one to predefine interactions and polynomial transformations to improve model performance. Furthermore, these models are resistant to multicollinearity issues since they simply ignore weaker correlated variables; they are also resistant to parameter clogging, in that redundant variables are simply ignored, which significantly improves on the stability of the model; the hyperparameters of these models can further be adjusted to lessen model complexity and overfitting (also known as regularisation), all of which contribute to more realistic variable importance measures compared to linear models' effect and significance measures.

[25] ROC AUC (receiver operating characteristics area under curve) plots the true positive relative to the false positive rate with respect to all decision probability thresholds (the threshold is a value from 0%-100% used to classify an observation as 1 as opposed to 0). When Type 1 errors (FP) and Type 2 errors (FN) are minimised across all decision thresholds, this value is maximised. The ROC AUC score therefore provides an integral based performance measure of the quality of the classifier. A value of 50% is expected for random noisy predictions. Generally, values from 80%-100% are considered as great classifiers. It is arguably the best single number machine learning researchers have in measuring the performance of a classifier (Bradley, 1997; Fawcett, 2006; Powers, 2011).

The vast majority of past high dimensional bankruptcy studies limit themselves to theoretically identified variables in prior literature (Jones, 2017; Kim & Upneja, 2014). Due to past literature's lack of identifying and isolating important high dimensional interaction pairs, this study does not limit itself to previously identified variables. This study focuses on a comprehensive range of accounting variables and their transformations. The expectation is that these inputs reflect all the necessary information to closely match the current research benchmarks (Jones, Johnstone, & Wilson, 2017; Volkov, Benoit, & Van den Poel, 2017). The simplicity of accounting measures is beneficial to the theoretical discussions associated with the variables. This study is the first to identify and describe high dimensional interactions. It describes the simultaneous interactions and marginal effects of up to three variables on the response variable.

The use of the XGBoost model (GBM) is largely driven by its success in practical domains; for example, it has been shown to be highly effective in data science competitions (Chen & Guestrin, 2016). The GBM model has the benefit of being interpretable, albeit not as well as logit models; however, the identification of non-linear interactions warrants its use. Multiple studies have shown that ensemble techniques can be used to improve financial distress prediction (Deligianni & Kotsiantis, 2012; Sun & Li, 2012). Many effective machine learning approaches have been used to predict default in recent years including artificial neural networks. In this study, I report on the greater usefulness of ensemble models over deep learning and other sophisticated models. I have tested multiple recurrent neural networks (RNNs), feedforward neural networks (FNNs) and convolutional neural networks (CNNs) architectures. I report the results of the best performing neural network, a deep convolutional neural network (DCNN), previously used for large temporal financial datasets (Chen, Chen, Huang, Huang, & Chen, 2016). I show that this particular neural network model performs worse than the GBM model but outperforms other neural network models.

II. Literature

Bankruptcy prediction research can largely be divided into the identification of ‘symptoms’ that lead to bankruptcy (Dambolena & Khoury, 1980; Gombola & Ketz, 1983; Scott, 1981) and studies that compare the performance of different bankruptcy prediction models (Altman, 1968; Ohlson, 1980). These two strains of research remain intact in modern bankruptcy prediction research. However, in recent years the traditional methods and processes have been uprooted by the development of advanced machine learning models.

Traditional statistical models have been largely dropped in favour of high dimensional models (Barboza, Kimura, & Altman, 2017; du Jardin, 2017; Jones, 2017; Liang, Lu, Tsai, & Shih, 2016). These advanced models present numerous advantages in flexibility, efficiency, and most importantly, enhanced prediction quality (Jones, 2017). The purpose of this chapter is to identify the accounting-related symptoms of bankruptcy in higher dimensions, and consequently I will give some attention to model performance to confirm the validity of identified predictor variables.

In recent years, the accuracy measure has been largely replaced by the ROC (AUC) score and other metrics (Bauer & Agarwal, 2014). Traditional significance tests of predictor variable performance have also been substituted by higher dimensional classification tree measures such as Gini Importance, Information Gain, and Split Frequency, as well as their relative measure counterparts like Relative Variable Importance (Behr & Weinblat, 2017; Jones et al., 2017; Mselmi, Lahiani, & Hamza, 2017). These are all data-centric approaches that look at the predictive ability of a parameter based on the variables selected and ranked in the nodes of the trees instead of significance tests. The application of these measures has practical and theoretical implications. Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan (2018), for example showed that machine learning importance measures can be used to understand and improve judges' decision-making in trials.

Recent models perform internal variable selection procedures, mostly removing the need for researchers to prune model inputs before feeding them into an algorithm (Mullainathan & Spiess, 2017). This means that the model can decide from a wide range of variables what it deems to be important without human intervention. With a sufficiently broad set of inputs, researchers can simply copy the model as used in one task and apply it to another, especially when making use of automated freeware to execute the necessary hyperparameter tuning (to optimise the model hyperparameter inputs). This strategy of reusing model architecture and inputs is used in this study to predict filing characteristics such as bankruptcy proceeding durations, survival, filing chapters, asset sales, and tortious claims. The readers mostly interested in the *results* of this study should read the next Contribution and Hypothesis section and, after that, move straight to the first table on page 102. For a further exposition about the research related to the Predictor Variables, Categories, Models, Predictive Power, and Filing Outcomes see the *Literature Addendum* in *Appendix B.*

III. Contribution and Hypothesis

This study contributes to the literature in several dimensions. It is the first study to make use of an XGBoost model to predict bankruptcies and to identify important accounting patterns associated with bankruptcies (Zięba et al., 2016). It is also the first study to implement a DCNN[26] (a biologically inspired variant of multilayer perceptron), which has shown great promise in other domains such as image recognition (Sharif Razavian, Azizpour, Sullivan, & Carlsson, 2014). This study further compares advanced deep learning models with modern decision tree ensemble models. To my knowledge, this is also the first study that uses a stacked model to improve prediction quality. The bankruptcy period spans 37 years (1980 to 2017), which is the longest ranging sample period of all cited literature. Furthermore, few studies acknowledge the theoretical equivalence between dollar-denominated accounting values and accounting ratios in the prediction of bankruptcy using higher dimensional models. In this study, I attempt to show that accounting values can at least be as important as ratios in predicting bankruptcies. It is also the first study to analyse interactions at an interaction depth of three variables.

Furthermore, this is the first attempt to rank the different accounting dimensions according to empirical ranking methods. This study contributes to the literature by using higher dimensional techniques to identify a structural difference in variable and category importance before and after the global financial crises (GFC). Most importantly, it is also the first study that attempts to predict filing outcome responses such as whether the bankruptcy process will endure for longer than a year, whether the firm will successfully emerge after the bankruptcy period, whether the bankruptcy is tortious, and whether or not the bankruptcy will involve asset sales. This is the first step towards successfully using high dimensional models to both improve prediction quality and predictor variable analysis. This study shows, from a categorisation of input variables, that Assets & Liability values, Solvency ratios, and Income values are the most important dimensions unlike past research that emphasises the importance of Profitability, Valuation, and Liquidity values. I put this difference down to the inability of linear models to capture the true reality of high dimensional relationships.

This study is also one of the first to investigate higher dimensional interactions and importance measures. One of the resulting interactions shows that when firms have large R&D programs, they are less likely to become bankrupt, all else equal. Some researchers

[26] Deep Convolutional Neural Network

have historically argued the opposite and said that there is a 'failure-inducement' effect in firms' effort to push for innovation when performance falls (Antonelli, 1989). I find that a handful of variables have a strong association with bankruptcy, many of which have not been noted by past research, such as the level of Stockholder's Equity, Depreciation & Amortization, and the Research & Development to Sales ratio.

This study includes a few additional steps to enhance the robustness of the results. It has the most imbalanced and lowest bankruptcy-to-healthy firm ratio of all decision tree and boosting related bankruptcy studies. It investigates bankruptcy prediction across all industries. The size effects of bankruptcy have been kept to a minimum by establishing minimum constraints on firm size. In addition, more than 10% or 120 of the bankrupt firm-year observations are filed on the premise of tortious claims, 70% of which relates to fraud. Chaudhuri and De (2011) observed that no models have yet been successful in detecting corporate fraud. I similarly find that if the fraud does not go hand in hand with financial distress, it is hard to predict these fraudulent bankruptcies. Therefore, the results reported in this study are much more conservative than those of past research. Following is a summary of key findings:

1. A Gradient Boosting Model (XGBoost) outperforms deep neural networks (DCNN, FFN) in prediction quality as measured by Accuracy and ROC (AUC) scores.
2. In a high dimensional setting, financial ratios have lower aggregate predictive ability over dollar-denominated accounting values as a result of linear constraints imposed on them.[27]
3. Solvency-related accounting-ratios are an important accounting ratio dimension for bankruptcy prediction compared against Profitability, Valuation, Liquidity, and Efficiency ratios.
4. Feature importance changes significantly before and after the GFC.
5. By using that same inputs as the bankruptcy prediction task, the GBM model is able to predict important filing outcomes, such as how long the bankruptcy process will endure, whether the firm will successfully emerge after the bankruptcy period, whether the bankruptcy is tortious, and whether or not it will involve asset sales.

[27] The individual constituents to the ratios are not able to interact with other variables independently leading to a loss of predictive power.

IV. Data

I use a sample of large[28] public firm bankruptcy cases filed under Chapter 11 of the US Bankruptcy Code as obtained from UCLA BRD[29] and a control group of a random sample of large and healthy firms. The vast majority of insolvent companies seek protection under Chapter 11 (Altman, 2002). Although Chapter 11 may be the original filing request, the courts may later decide to do a full asset sale outside of Chapter 11 or ask the company to file under Chapter 7. Those observations are also included in the sample. The sample of firms only comprises publicly listed firms for which financial statements were available. The final sample comprises 33,242 healthy firm years and 1224 bankruptcy firm years from 1977 to 2016, with an average bankruptcy to healthy firm ratio of less than 4%, and a standard yearly deviation of more than 4 percentage points, highlighting the variability of bankruptcies over the sample period.

Large firms have been chosen to limit the noise when identifying the most important accounting value determinants in predicting bankruptcy and filing outcomes at a national level. The purpose of this study is not so much prediction success as it is the identification of important accounting variables and interaction effects. In saying that, good prediction success is necessary to validate the predictive power of variables. Consistent with past literature, firms are considered to be bankrupt if they filed for bankruptcy within one year. The accounting information is obtained from Compustat. In this study, I use simplified and standardised financial information for all firms; it includes accounting information from the Balance Sheet, Income Statement, and Cash Flow Statement. 70% of the bankruptcies occurred after 2000. Half of the bankruptcies emerged after the GFC. The BRD database is unique, in that it includes not just the date of filing, but also the date the case was disposed by the court and information on whether an asset sale transpired, whether the business re-emerged, whether the case has been tried under the law of tort, and lastly, information on chapter of filing. This data in this study therefore allows us to not just predict the occurrence of a bankruptcy but also predict the associated filing outcomes.

To obtain a large enough sample of bankrupt firms without having to deal with small firm bankruptcies, I collect up to three years of data to predict bankruptcies one and two

[28] The BRD database only collects information on large firms. A firm is large if the firm reports assets of more than $250 million as measured in 1979 dollars on the last 10-k filing before the bankruptcy case.
[29] "The BRD contains data on all of the more than one-thousand large public companies that have filed bankruptcy cases since October 1, 1979" http://lopucki.law.ucla.edu/

years in advance. When firms had missing data, I followed longitudinal imputation procedures by comparing multiple methods and selecting and implementing the best method.[30] Consistent with Ohlson (1980) and Jones and Hensher (2004), firms do not get removed from the dataset simply because they are recently or newly listed; as a consequence, a few firms in the sample had only a small amount of data.

Consistent with past research, the bankruptcy event is a binary response. All healthy firms are coded with **0** and failed firms are coded with a **1** in the two preceding years. This is necessary as we want to investigate the performance of the firm before the bankruptcy filing. It is not a requirement for all bankruptcy studies; it is only required for discrete choice models, as it is the only means to incorporate a time dimension. Duration type models such as hazard models are set up to predict time-to-event using survival functions, and for these models it is good practice to only label a firm as bankrupt in the year of the bankruptcy (Beaver, McNichols, & Rhie, 2005; Hillegeist, Keating, Cram, & Lundstedt, 2004). In this study, a firm entering into bankruptcy is labelled as bankrupt for two firm-year observations. This study also identifies the performance of the classifier for data that is coded as bankrupt only in the year before the bankruptcy.

V. Methods

This section provides a short discussion of the bankruptcy prediction methodology used in the study. Readers who are interested in the results of this study should feel free to skip this section and go straight to the first results in *Table 19* on page *102*. I also urge the reader to consult *Figure 1* for each concept not well understood, as this figure highlights the machine learning process and allows the reader to follow the concept to the appendix or other sections for additional explanation. The empirical part of the study consists of steps such as the imputation of missing values, the creation of training, validation, and test sets, and finally the development and training of a model using hyperparameter tuning on validation sets. A trained model can be calibrated on a validation set to adjust the model parameters.[31] Each firm-year observation is described according to a set of variables and a response value. The trained algorithm is then used with new inputs against a pure holdout test set to assess its accuracy and ROC (AUC) score, among other things. This paper performs and reports more than ten robustness tests on XGBoost model performance.

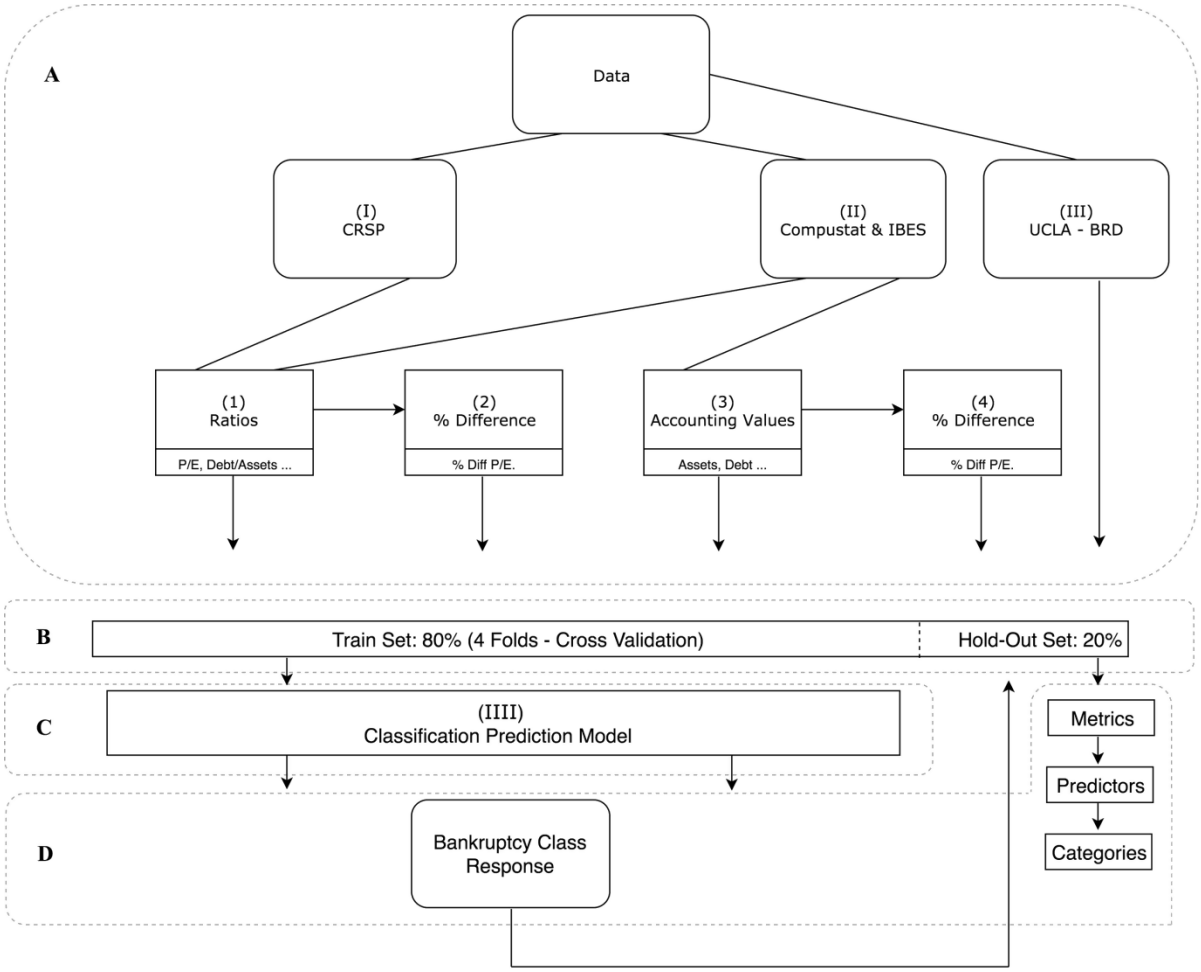
[30] See Appendix XI.D.2 page 182 for a description of the imputation process.
[31] The model parameters include the tree debt, the number of estimators, the learning rate, and many others.

Before delving into the prediction models, it is valuable to understand the cross-validation technique used in this study. The bankruptcy data is sorted by time, and the first 20% of the data is used for training while a random selection of 15% of the remaining data is selected for the validation set. This validation set is used to perform model selection and hyperparameter adjustment after which it gets dropped indefinitely.

Using this approach ensures that the testing data never contains data that is older than the training data, which is a sensible step for preserving prediction integrity. I then use a unique longitudinal blocked form of performance evaluation that suits this form of bankruptcy prediction. This approach creates multiple models in time series to predict the subsequent periods' observation until the last model incorporates all the training data. This chapter uses the same validation approach as the first chapter and like before the final reported metric is the average across all the time splits. For more information on this method, see the cross-validation section XI.D.4 on page 150.

This study goes beyond simple machine learning; it does not only report on how well the prediction fits the test set, but it also sheds light on the most important predictor variables to bankruptcy. The first stage will follow a process similar to standard machine learning and the second part will focus more intensely on what these predictions tell us about bankruptcy and filing outcomes. *Figure 1* is a critical diagram revealing the process of obtaining data and training and testing a model.

Figure 16: Process Tree



(A). I and II are the data sources used in this study; this is dealt with in section III22 (I) CRSP was used to obtain price data, (II) Compustat was used to obtain fundamental accounting information (III) Publicly available information was obtained from the UCLA LoPucki Bankruptcy Database supported with publicly available bankruptcy filings. Items (1) - (4) constitute the final variable groups used in this study. The accounting ratios (1) and values (3) have been transformed to get the respective yearly percentage changes (growth values), (2) and (4). The accounting ratios have been inspired by the WRDS Industry Financial Ratio Manual (2017). (B) identifies the cross-validation split process; details of this process can be found in XI.D.4150 as it is more involved than this figure illustrates. (C) identifies the gradient boosting prediction model. (D) is the use of the bankruptcy response variables in the trained model to identify performance metrics on a hold-out set. These metrics are used to identify important variables and variable categories using partial dependence plots.

VI. Prediction

In this classification task I create a classification model (classifier), that assigns an observation to every class based on the learned patterns of a training set. In this study, the outcome to be predicted is 'bankrupt' or 'healthy' firm-years. The training consists of past observations where the classes are known. The model, therefore, learns class associations from the past patterns of explanatory variables commonly called features and maps this input data into a class outcome according to newly learned, weighted, and approximated functions. The XGBoost classification model used in this study is a probabilistic classifier that outputs a probability not unlike a Probit model. Throughout this study, 50% is selected as the decision rule; therefore, the chosen class is the one with the highest probability.

I use several statistics to report the out-of-sample accuracy. The most important metric is the ROC (AUC) score. The ROC (AUC) score is the benchmark statistic in classification research (Bradley, 1997; Fawcett, 2006; Ferri, Flach, & Hernndez-Orallo, 2002). Its use in bankruptcy research has also picked up; in just the last year alone, more than eight studies within neural network and boosted tree model bankruptcy prediction research have made use of this method (*Table A40, Table A41*). ROC curves plot the true positive relative to the false positive rate with respect to a threshold probability.[32] An area under the curve greater than 0.8 is considered to be a good classifier (classification model). A visual example of the ROC curve can be found in *Figure 17* on page 103.

The ROC curve is simply the relationship of the true positive rate to the false positive rate with respect to a probability threshold. The diagonal line can be described as the "line of luck" and has an AUC of 0.5. Generally classifiers should perform better than 0.5 to be of ay use at all. An AUC of 1 represents the best possible classification score with no Type I and Type II errors, that is, perfectly predictable. Conventionally ROC (AUC) scores above 0.8 and 0.9 indicate "good" and "great" classifiers, although the interpretation is domain dependent. For a visualisation of the ROC curve, the Type I and Type II errors have to be plotted against all threshold values. The primary ROC curve in this study is reported in *Figure 17.*

[32] The threshold probability is ordinarily set at 50%; this means that if the classification model predicts a 51% chance of a future bankruptcy then the observation would be classified as bankrupt. The 50% probability threshold can be adjusted to best fit the task at hand. For example, increasing the threshold would lead to fewer bankruptcy predictions but better-quality predictions.

Although the ROC (AUC) measure appears in a lot of research areas, it is somewhat limited in that it uses different misclassification cost distributions for different classifiers. An alternative measure has been proposed by Hand (2009) to avoid this limitation. Similar to Jones et al. (2017), I found no significant difference in using the H-score as opposed to the ROC (AUC) score. I therefore only report the more commonly known ROC (AUC) score to avoid confusion. I also present the use of ROC (AUC) measures in conjunction with an inductive technique to identify the importance of groups of predictor variables to explain model success. The results of the inductive technique are presented in *Table 1* on page 115. The ROC curve is further discussed in the Evaluation Section.

This study also reports on accuracy, false positive rate, and cross-entropy (negative average log-likelihood) metrics. The accuracy measure is not well-suited for imbalanced sets and can largely be ignored unless the reader assigns equal importance to the correct prediction of both healthy firms and bankrupt firms in a dataset where less than 4% of the actual observations are bankrupt firm years. The issue with the accuracy measure is that it does not look at class breakdown precision, nor does it provide evidence of true positives or true negatives values. The false positive rate similarly serves a somewhat limited role in this study. It is primarily used as validation metric to ensure that the trained model, from which the variable importance measure is derived, does not mistakenly predict healthy firms as being bankrupt (false positives) as this would undermine the validity of the variable importance measures and resulting variable ranking. The last reported metric is the cross-entropy measure; it serves a purpose similar to the ROC measure but stresses a probability interpretation of model prediction. The cross-entropy measure principally serves as a corroborative measure. For model quality and prediction quality, I urge the reader to focus on the ROC measure. Overall, I maintain that the ROC measure is the most relevant measure in the context of bankruptcy prediction

Further analysis also includes the use of a confusion matrix. This study solves for a binary classification problem that produces a 2×2 matrix. The columns of the matrix represent the predicted values, and the rows represent the actual values for bankruptcy and healthy firm predictions. In the cross-section of the rows and columns, we have the True Positive (TP), False Negative (FN - type II error), False Positive (FP - type I error), and True Negative (TN) values. It is useful for a classification study to produce a classification matrix to aid intuition, especially when the dataset is imbalanced, such as in the case of bankruptcy prediction, where a small minority of the observations are bankruptcies. The confusion

matrix is reported in *Table 20* on page 104. Further, I report 13 additional measures for the model as tested in (3), *Table* A45 and *Table A46 on* page 168 and 169.

Accuracy is defined as the percentage of correctly classified instances (observations) by the model. It is the number of correctly predicted bankrupt (true positives) and correctly predicted healthy firm years (true negatives) in proportion to all predicted values. It incorporates all the classes into its measure $(TP + TN)/(TP + TN + FP + FN)$, where *TP*, *FN*, *FP*, and *TN* are the respective true positives, false negatives, false positives and true negatives values for both classes. The measure can otherwise be represented as follows:

$$acc(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}-1} 1(\hat{y}_i = y_i) \qquad (12)$$

In *Table 19*, I empirically compare the performance of a range of binary classification models. I maintain a consistent framework of common sets of inputs as described in the Data Section, III in this paper. The process also involves identical steps for data handling and parameter estimation. From *Table 38* to *Table 37* on page 145 to 144. I select the best model type which is XGBoost to identify its performance under different conditions; this includes adjustments to the testing procedure, parameters, data assumptions, and sample distributions. These tests allow for more robust results of model performance. However, I also include tests that show how the predictive performance can be enhanced by changing the underlying structure and shape of the data.

Two of the classification models in this table make use of neural networks (NNs). These models are designed to identify latent and highly complex nonlinear relationships in the dataset. These are your quintessential "black-boxes." Researchers specify the architecture of the network and the initial inputs to feed into the first layer of these models; apart from that they play a very limited role. Unlike the XGBoost model, neural networks provide no mathematical formalities of the parameters that define relationships apart from their internal mathematics. Neural networks also have issues with handling data of mixed types such as categorical and continuous. Historically, neural networks have been very cumbersome as they are computationally intensive, but recent advances in processing technology have lightened this burden.

The first NN model used in this study is a Deep Feed Forward Network, which is structurally similar to other researchers' use of Multilayer Perceptron Models (Jones et al.,

2017; Mselmi et al., 2017), but it has 5 densely connected hidden layers instead of two. The other NN model is a Deep Convolutional Neural Network (DCNN). This network is a biologically inspired variant of MLP that uses four densely connected hidden layers with the addition of a convolutional layer. The Deep Convolutional Network is traditionally applied to image recognition tasks. Repetitive blocks of neurons are applied across space to learn filters and variables that are associated with the response. The layer structure of the networks has been designed by hand, while the number of neurons has been automatically selected by hyper-parameter search operations. For more information on the construction of these NN variants, see page 147 in the Appendix.

Lastly, as a way to reconcile the performance of these modern models with traditional statistical models in past research, I also include a Logit Model. The XGBoost model outperforms the deep learning models, and the Convolutional Neural Network far outperforms the Feed Forward Network. Given more data, I would expect the gap between the XGBoost model and the DCNN model to progressively decrease. Deep learning models are known to perform especially well with larger datasets. The Logit Model is the worst performing model in this study. Unfortunately, the Logit model does not perform well with too many variables, which often leads to overfitting (Altman, 1968; Ohlson, 1980). Irrelevant variables that enter the global maximum likelihood solution of the Logit Model severely impact the quality of the reduction and model stability, whereas the higher dimensional models are relatively unaffected by noisy variables and outliers. Adding regularisation techniques like L1, L2 and elastic net could also improve to improve the logistic regression prediction performance. A more in-depth theoretical comparison of these models can be found in XI.E on page 154.

Table 19 shows that the XGBoost model, i.e., GBM model performs better than the other models investigated for all metrics considered. It is therefore a valid model to approximate the underlying function and would therefore provide reliable variable importance values. Even without adjusting for the fact that bankrupt firms only account for a very small percentage of the overall firm-years in this sample compared to other studies, this study produced the best accounting-based prediction of all previous studies as measured by both an ROC (AUC) score of 0.958 and a prediction accuracy of 0.976. Furthermore, despite purely making use of accounting data, this study produced the best neural network type model of all previous studies (*Table A40* on page 163) with an AUC of 0.914 and accuracy of 0.95.

Table 19: XGBoost and Deep Learning Model Performance Comparison

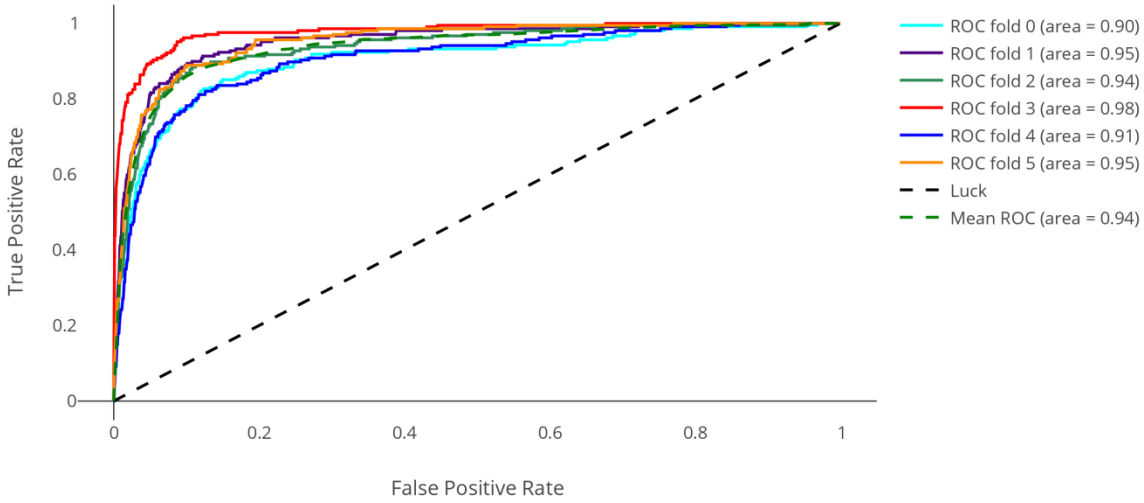
Metrics	XGBoost Model	Deep Feed Forward Network	Deep Convolutional Neural Network	Logit Model
ROC AUC Sore	0.9587	0.8444***	0.9142***	0.7092***
Accuracy Score	0.9755	0.9324	0.9518	0.6856
False Positive Rate	0.0037	0.0666	0.0296	0.2296
Cross-entropy	0.1414	0.5809	0.2996	1.1813

This table illustrates the performance of two deep learning models against the XGBoost Model. The Feed Forward Network is a deep learning network that does not circle back to previous layers. The Convolutional Neural Network is a biologically inspired variant of MLP, popularised by recent image classification studies. The best possible Logit model was established by choosing a selection of the best variables. Further results include the isolation of the 10 best predictor variables (using the Gini Index) in all models; this produced similar results to the above table both in extent and in rank. *p<.1 ** p<.05 *** p<.01. Significance levels are based on a two-tailed Z test to identify the statistically significant difference between all contender models and the best performing model, which is made possible due to the cross-validation process.

The widespread use of the AUC measure in recent studies allows researchers to compare the performance of their model with other studies. The benefit of the metric is that it is somewhat agnostic to different healthy-to-bankrupt firm distributions, at least more so than accuracy measures. In saying that, the problem is that, notwithstanding the recent universal adoption of AUC, it is still hard to compare performance across studies as the sample distribution does have some effect on the performance; other factors include the type of firms, industry, country, sample period, jurisdiction, and the definition of corporate distress.

The average ROC (AUC) of more than ten past decision-tree ensemble studies is around 0.927. The best performing is 0.997 and the worst performing is 0.859. In spite of the conservative sample selection in this chapter, the decision tree ensemble (XGBoost) model used in this study performed better than the average of past reported studies. It is also the best model when compared to other studies that only used accounting values as inputs. The average AUC of eight different neural network studies is 0.850; the best and worst performing past study has an AUC of 0.901 and 0.750 respectively. The DCNN of this model achieved an AUC of 0.9142, making it the best performing neural network of all past research. *Figure 17* below identifies the ROC and the associated AUC of a five-fold cross-validation model. It is the best way to visualise the AUC metric. Both this figure and the aforementioned time-series cross validation table show that there is reasonable amount of variability in the curves for each validation iteration. The dotted green stripes highlight the average AUC and the diagonal line, the line of luck.

Figure 17: The Receiver Operating Characteristic and Area Under the Curve - ROC (AUC)



This figure reports the ROC and the associated area under the curve (AUC). The random ordering or luck line is plotted diagonally through the chart and represents a series of random and noisy predictions. The chart reports five different cross validation folds and the associated performance as well as the mean ROC presented as a dashed green line. The reported mean is much higher than 0.90, which conventionally represents a 'great' classifier.

The results in *Table 20* aggregate multiple predictions to show a contingency table of the different predictions against actual outcomes. *Table 20* presents the precision and improvement score for both healthy and bankrupt firm years. The precision metric is the class-specific accuracy of the predictions made; it is a useful measure to know when there is a class imbalance. The model correctly predicted 258 out of 374 (116+258) predicted bankruptcies. For the purpose of the confusion matrix, the classification threshold is set to minimise the false positives at the expense of a higher false negative rate or recall error. As it happens, a threshold close to 50% is right for this purpose. This decision threshold can, of course, be adjusted to effect change in the table below as the chosen value is wholly up to the researcher. The decision threshold adjustment is possible because the model has a logarithmic loss function that outputs a probability associated with each class, which can simply be adjusted.

Table 20: Healthy and Bankrupt Confusion Matrix

Aggregated Health and Bankrupt Firms Matrix		**Predicted**		Sample Proportion
		Healthy	Bankrupt	
Actual	Healthy	**29041 - TN**	116 - FP	0.96
	Bankrupt	805 - FN	**258 - TP**	0.03
Precision		0.97	0.69	30220
Improvement		0.01	0.66	-

This bankruptcy prediction task solves for a binary classification problem that produces a 2×2 matrix. The columns of the matrix represent the predicted values, and the rows represent the actual values for bankrupt and healthy firm predictions. In the cross-section of the rows and columns, we have the True Positive (TP), False Negative (FN - type II error), False Positive (FP - type I error)), and True Negative (TN) values. The sample proportion on the far right is equal to all the actual observations of a certain classification divided by all the observations. The *precision* is calculated by dividing the true positives (Bankruptcies) with the sum of itself and the false negatives (Healthy). An example along the second column: $258/(116 + 258) = 69\%$. The improvement is the percentage point improvement the prediction model has over a random choice benchmark.

The good performance in *Table 20* can further be highlighted by drawing up a confusion matrix from random guessing. *Table 21* shows the performance of random guessing based on knowledge of the underlying test distribution. There is a big difference between the distribution of bankruptcy selections in this table compared to the model-predicted table. The performance of the table above is much better than random predictions based on the underlying sample distribution. The random guessing model correctly predicted 37 out of 1063 predicted bankruptcies. This equals a precision of just over 3%, which is much worse than the model's 69%.

Table 21: Random Guessing Confusion Matrix

Aggregated Health and Bankrupt Firms Matrix		**Random Guess**		Marginal Sum of Actual Values
		Healthy	Bankrupt	
Actual	Healthy	**28131 - TN**	1026 - FP	29157
	Bankrupt	1026 - FN	**37 - TP**	1063
Marginal Sum of Guesses		29157	1063	1063

This table is formed by 'randomly choosing the observations' by allocating the observations according to the underlying distribution, as presented by Sample Proportion in *Table 20.*

VII. Variable Importance

In bankruptcy prediction, it is useful to know the relative contribution (RVIs) of all variables on the final prediction outcome. Since the variable importance measures, such as the Gain measure and Gini Index, are relative measures, it is conventional to identify the largest value to be labelled with 100 and then to scale all remaining variables according to the most predictive value (Breiman et al., 1984). The final RVI is the frequency of variable splits weighted by the average squared improvement of the model at each split across all trees (Friedman & Meulman, 2003; Hastie, Tibshirani, & Friedman, 2009). *Figure 18* presents an example of a single decision tree used in this study. At each split, there is a re-estimated contribution that I convert into a percentage for simplicity. In *Figure 18*, I only report each subsequent split to easily fit the full tree at a depth of 12. This figure should give the reader an indication of the internal structure of the 'weak learners' of the XGBoost models. However, in practice, instead of one tree, there are multiple trees. This is called an ensemble. A clear benefit of the XGBoost model is that it allows a wide range of variables to contribute to the overall prediction unlike most conventional models; there is evidently a reasonably even distribution of RVIs across multiple predictor variables (*Table 22*). This is the power of high dimensional input space; multicollinearity does not impair the predictive performance of the model to the extent of conventional linear models.

In *Figure 18*, the first variable that gets split is the net income adjusted for common stock equivalent. The inequality equations represent the split in the branch that leads to differently predicted outcomes. The outcome is reported as the probability of bankruptcy. These numbers can be seen in the second line of nodes. As long as a variable has been used once and contributed to a change in the prediction probability, then it has an RVI above zero. This indicates that the variable adds some predictive power to the overall model. The RVIs reported in *Table 22* as mentioned fall between a value of 0 and 100. The most important variable has a value of 100, and the other variables are scaled to match this level of importance. In this table, it is clear that the relative strength of the variable importance measures differs significantly largely across all variables.

Figure 18: Depth 12 - Decision Tree

This figure illustrates a decision tree with depth 12, the original depth used in this paper. The models in this paper have used as many as 4800 different trees to predict every observation. The above output is an example of one tree. On the right-hand side, the tree shows that when income (niadj) is negative (below 50 million), and debt to assets (debt_at) is small, then the likelihood of bankruptcy is less than if the debt to assets ratio was larger (than 0.4955). On the left-hand side, it shows a similar relationship but with debt to invested capital (debt_invcap). The number of splits and overall contribution of the different predictor variables is used to measure the predictive importance or ability of each feature. Further, the interaction pairs are also calculated by identifying the outcome contributing relationships between different predictor variables. This tree is selected for its symmetrical properties, most trees are asymmetrical and have varying branch lengths.

Table 22 presents the predictive power of the top 50 input variables to the prediction model using Gain as a relative variable importance measure. The table further includes the direction the variable has with the response variable. A positive (+) means that bankruptcy is more likely to occur when the variable increases and a negative (-) means that bankruptcy is less likely to occur when the value increases. Therefore, the minus (-) is a good sign. The strongest variables as identified in *Table 22* are pre-tax income (100) and income before extraordinary expenses (91). The strongest two ratios are the EPS (61) and the Price to Sales ratio (55). It is important to remember that the variables get selected for their contribution to the overall model and that these contributions do not occur in isolation from other variables i.e., the predicted values are primarily a consequence of variable interactions.

Table 22: Predictive Power of Variables

Input Variable	RVI	Post GFC RVI	D	Category
Pretax Income (PI)	100	100	-	Income
Income Before Extraordinary Items (IBC)	84	91	-	Income
EPS(Basic) - Exclude Extra. Items ($&c)	61	27	-	Profitability
Price/Sales	55	57	-	Valuation
Liabilities - Total (LT)	50	19	+	Liability
Interest and Related Expense (XINT)	44	8	+	Expense
Long-Term Debt - Total (DLTT)	44	17	+	Liability
Stockholders Equity - Total (SEQ)	35	22	+	Equity
Total Debt/Total Assets	31	6	+	Solvency
Inventories - Total (INVT)	30	18	+	Asset
Net Income - ADJ for Com Stock Equiv	29	42	-	Income
Depreciation and Amortization (DPC)	26	12	+	Equity
Total Debt/Invested Capital	24	8	+	Solvency
Accounts Payable (AP)	22	14	+	Liability
Research and Development/Sales	21	6	-	Other
Property, Plant & Equip. - Total(Net)	21	17	+	Asset
EPS (Diluted) - Excl. Extra. Items ($&c)	21	25	-	Profitability
Price/Book	21	35	-	Valuation
% Change in Price/Sales	20	37	-	Valuation
Capitalization Ratio	20	32	+	Solvency
Free Cash Flow/Operating Cash Flow	18	11	-	Solvency
Cash and Short-Term Investments (CHE)	18	10	-	Asset
Sales/Stockholders Equity	18	8	-	Efficiency
Cash Balance/Total Liabilities	18	15	-	Solvency
Income Taxes - Total (TXT)	17	15	+	Liability
Capital Expenditures (CAPX)	17	8	+	Asset
% Change in Property, Plant & Equip.	16	7	-	Asset
Sale of Common and Preferred Stock	16	6	-	Asset
Price/Cash flow	16	21	-	Valuation
Total Debt/Capital	16	7	+	Solvency
Cost of Goods Sold (COGS)	16	7	+	Expense
Operating Activities - Net Cash Flow	16	8	-	Cash Flow
Long-term Debt/Invested Capital	15	43	-	Solvency
Operating Income Before Deprec.	15	26	-	Income
After-tax Interest Coverage	14	5	-	Solvency
% Change in Income Before Extrao. Items.	14	2	-	Income
Long-term Debt/Total Liabilities	14	17	+	Solvency

Investing Activities - Net Cash Flow	13	10	-	Asset
Sales/Invested Capital	13	12	-	Efficiency
Long-Term Debt - Reduction (DLTR)	13	35	-	Liability
% Change in Common Equity - Total	13	11	+	Equity
Liabilities - Other (LO)	13	16	+	Liability
Current Assets - Other (ACO)	13	12	-	Asset
Book/Market	13	11	+	Valuation
% Change in Liabilities - Total (LT)	13	1	+	Liability
% Change in Interest and Related Expense	13	8	+	Expense
Assets and Liabilities - Other (Net Change)	13	10	+	Asset
Assets - Total (AT)	13	4	-	Asset

This table contains a list of the variables with the most predictive power as measured by the gain statistic. Gain in this paper is defined as the number of splits the variable undergoes, weighted by the squared improvement of the model that resulted from each spilt. The relative variable importance (RVI) is simply the division of subsequent variable gains by the gain of the most contributing variable scaled by 100. It is calculated from 1979-2016. The Post GFC RVI is the relative gain from 2008-2016. D is the direction of the variable with a bankruptcy outcome. The category is the bucket or dimension in which the variable falls. It is used later to analyse which category has the most predictive power.

The model used in the above table near-randomly discriminates between correlated variable pairs; therefore, the predictive performance of a particular accounting dimension will likely be distributed among a few variables. The process of combining variables in predetermined accounting categories can help to highlight the most important dimensions to predict bankruptcy. The most predictive categories and associated variables are the Assets and Liability category, and more specifically, the Total Liabilities, Long-Term Debt, Accounts Payable, PP&E, Cash, Short-Term investments and Inventory variable inputs. The second most predictive category is Income, more specifically, Pre-tax Income and Income Before Extraordinary Activities input variables. In addition, the third most predictive category is Solvency, more specifically, the Total Debt to Total Assets Ratio, Total Debt to Invested Capital Ratio, and the Capitalisation Ratio and Free Cash Flow to Operating Cash Flow Ratio input variables. The fourth category essential for predicting bankruptcies is Valuation and Profitability, more specifically the Price-to-Book, Price-to-Sales, ROE, and EPS input variables. Other not as essential categories and individual value pairs include Equity - Shareholders Equity, Expense - Interest and Related Expense, Efficiency - Sales to Stakeholders Equity, Other - Research and Development to Sales, Liquidity - the % change in the cash ratio and lastly, Cash Flow - Operating Activities Net Cash Flow. I further show that a very competitive prediction model can be created by using only 50 input variables (*Table 39*, Second Column).

In this study, I include multiple accounting-related variables purely because there is no clear consensus as to what variables and interactions are the most important in lower dimensional models, not to mention higher dimensional models. *Table 6* identifies the characteristics of the most predictive variables to the model. The results in this table support my hypothesis that 70 simple accounting values have improved predictive power over 72 ratios due to the high dimensional interactions that remove the requirement to pre-specify ratios. The issue with ratios in high-dimensional studies is that they self-impose linear restrictions in the relationships between the numerator and denominator. In theory, if you feed the model the raw input to the ratio, it should more easily scour patterns for non-linearity between the inputs, as financial measures are reported to be significantly non-linear (Foster, 1986). As a result of this outcome, I will treat each category as a contributor to prediction success rather than favouring the ratios as the only true variables and regarding the fixed values as 'controls' as done in past studies. The results in Panel B further show that fixed variables are more important than growth variables (% change variables).

Table 23: Variable Type Analysis

Panel A

Type	Importance (%)
Simple	0.57
Ratio	0.43

Panel B

Construction	Importance (%)
Fixed Period	0.87
Growth	0.13

Panel C

Type	Construction	Importance (%)
Fixed Period	Simple	0.50
	Ratio	0.38
Growth	Simple	0.07
	Ratio	0.06

This table computes the importance of the variables grouped by certain characteristics. Panel A groups the variables by whether the value is simple or whether it is a ratio. Panel B looks at whether the value is simple or whether it is a calculation of the change of the variables between time *t-1* and *t,* i.e., a growth measure. Only the most predictive growth variables are included in the model. Panel C is the value type in Panel A grouped with the construction type in Panel B.

Due to the multi-dimensional nature of the model, a good approach to study the most important variables is to group them. In *Table 22*, all measures are grouped according to how they would appear on standardised financial reports. That includes *Assets, Liabilities* Expenses, *Income, Equity and Cash Flow*. All other categories are classified according to the following definitions: *Solvency* measures are ratios associated with financial soundness and the ability of the firm to meet its long-term obligations; *Valuation* measures are accounting ratios that identify the firm's attractiveness and whether the stock is under or overpriced; *Profitability* ratios measure the ability of the firm to generate returns; *Efficiency* ratios track the firm's effectiveness in utilizing assets and liabilities; *Liquidity* ratios measure the firm's ability to meet short-term demands; and lastly, *Other* ratios incorporate values such as Research and Development to Sales, and Labour expense to Sales ratios. All predictor variables are exclusively allocated to one of these eleven categories.

Looking back at *Table 22*, it is clear that there is a difference between the variable importance post-GFC (2008) compared to the importance over the full sample period. The best approach would be to identify the categories that show significantly more or significantly less predictive power. After the GFC, the model loading on liability variables is a lot less while valuation measures have increased in importance over equity measures, highlighting the fact that less trust is put on equity as compared to the markets' valuation of the firm. All income measures (PI, IBC, NAIDJ, % IBC) also show increased predictive power after 2008. These differences may be a priori evidence of a structural change in bankruptcy prediction.

In *Table 24*, I empirically investigate the categorical importance post-GFC. I show that Solvency, Income, Valuation, and Profitability measures are more important after the GFC and that Liquidity and Cash-Flow measures are less important after the GFC. These changes make sense, as cash flow and liquidity were not at the heart of the crises. It was largely based on issues of capital structure (Solvency) and unjustified valuations (Valuation). In addition, the prediction algorithm post-GFC gives less attention to the reported valuation (Equity) of firms as it learns that these accounting measures cannot be 'trusted' as much as they could be before the crises. The two measures (3) - (4) in *Table 24* will later be used as three of the nine measures to rank the final categories after the required recategorisation because of strong correlations between some accounting categories.

Table 24: Predictive Power of Categories

Category	(1) R/A	(2) Category Importance (CI) (%)	(3) Relative CI (RCI)	(4) Post GFC RCI
Asset	A	0.18	100	100
Solvency	R	0.17	95	99
Income	A	0.13	70	80
Liability	A	0.11	62	62
Valuation	R	0.11	62	80
Equity	A	0.08	46	32
Profitability	R	0.07	40	44
Expense	A	0.06	31	34
Efficiency	R	0.04	21	21
Other	A	0.02	12	11
Liquidity	R	0.02	12	10
Cash Flow	A	0.01	8	4

This table looks at some predictive metrics of the categorisations used in this study. This table specifically highlights the importance of these categories in predicting bankruptcies before and after the GFC. Here we can compare the difference between the overall Relative Category Importance (RCI) in (3) and the RCI post GFC (4). (1) identifies whether the category constitutes an accounting ratio - R or whether it is a simple accounting value - A. (2) indicates that the category importance (CI) is a summation of the gain measure (PI) of the contained variables. (3) RCI is a relative measure based on the most important category (CI). (4) identifies the respective RCI for the sample period from 2008 onwards.

Decision tree models are mostly robust to multi-collinearity; they will simply choose one of the many correlated variables when deciding upon splitting the tree. This makes sense for a model that is purely interested in boosting prediction performance. Compare this against linear models that rely on all correlated variables for prediction. For decision trees, correlated variables would simply not add to the split process anymore as they do not bring new information that has not already been provided by the first feature. Some boosting models that have several correlated variables as inputs will tend to choose one correlated variable and use it in several trees. The randomised nature of the XGBoost model and associated parameters will, however, randomly select between collinear variables with each subsequent tree and variable selection iteration. This means that across many trees, such as used in this study (4800 trees), the most important correlated variables will on average come out on top. The issue is that the importance still gets distributed across many variables; a prime example is the two top-performing variables, pre-tax income and income before extraordinary expenses, which are both highly correlated (95%+). Therefore, to get a better indication of the overall importance of certain accounting dimensions we can categorise the variable as has

been shown in *Table 24.* One of the benefits of this study is that it uses labelled accounting data that makes it easy to understand what variables share similar characteristics. Finally, there can be correlations across categories; the best way to deal with this is to understand that the components of these categories are similar and that they should be combined to form a single category.

One process to identify the success of the categorisation is to PCA transform the variable sets and simultaneously ensure for low correlation between the categories. For the vast majority of the categories, there seems to be no correlation, which is a good sign. However, there is, as should be expected, large collinearity between liabilities and assets. This has to do with the basic accounting equation, where the majority of the interaction is between Liabilities and Assets over time. Further, the Profitability, and Valuation dimensions also seem to be highly correlated. For that reason, it is better to group Assets & Liabilities, and separately, Profitability & Valuation dimensions together. This logic is often left out in the variable discovery task of machine learning bankruptcy studies. Researchers repeatedly forget to look at the correlation between variables. This is an issue because the reported variable importance is misleading if the importance is divided among various categories.

The PCA deviation score measures the standardised correlation deviation between components of a category. In studies where the variables are unknown, a deviation of more than one can be indicative of a category that can be effectively deconstructed into more categories as long as the constituent variables to a category make theoretical sense. In general, a high deviation in correlation is usually a good thing for each category, as uncorrelated variables lead to improved performance. The high reported value for assets and liabilities is likely due to structural differences between current and non-current assets or liabilities. *Table 25* looks at the correlation between categories, after which a decision is made whether or not to combine certain categories for more informed results. The PCA deviation is further used to decide whether a category should be split into smaller subcategories. The results indicate that a few categories should be combined but that none of them has to be broken into smaller constituents.

Table 25: Correlation and Categorisation Analysis

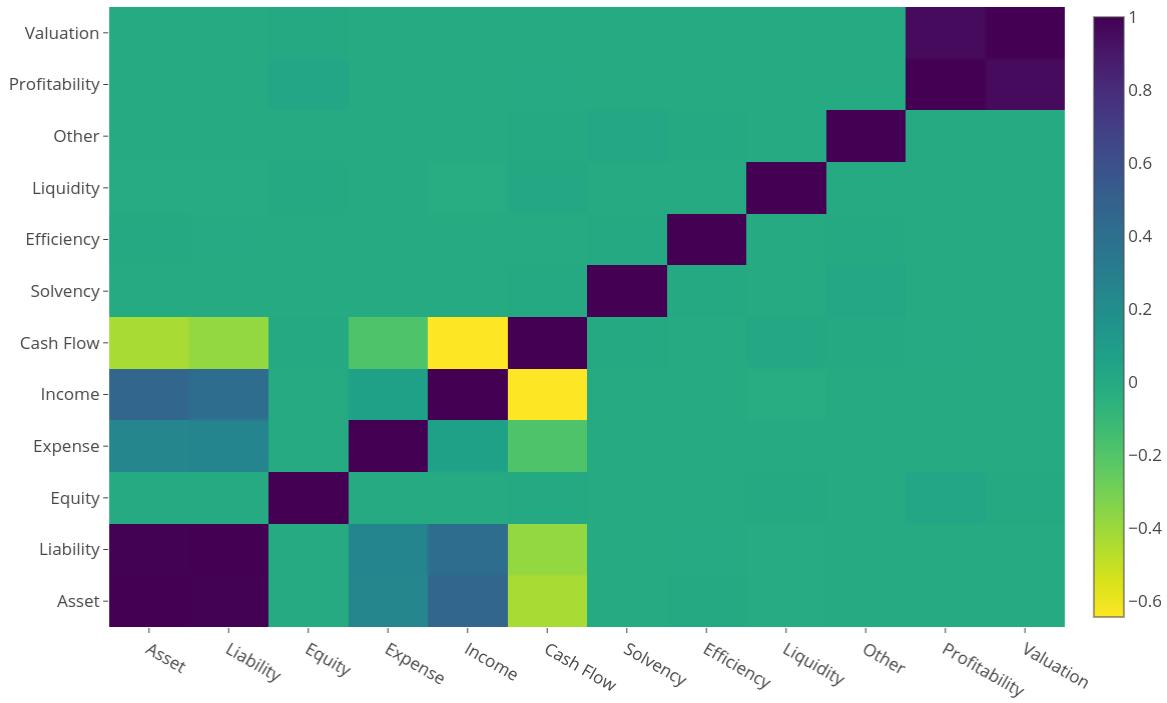
Category	Closely Correlated With	Correlation Score	PCA Category Deviation
Asset	Liability	0.82	1.43
Solvency	Other	0.02	0.49
Income	Cash Flow	-0.56	0.87
Liability	Assets	0.82	1.37
Valuation	Profitability	0.93	0.36
Equity	Profitability	0.03	0.78
Profitability	Valuation	0.93	0.36
Expense	Liability	0.27	0.39
Efficiency	Other	0.01	0.40
Other	Cash Flow	0.03	0.44
Liquidity	Cash Flow	0.01	1.16
Cash Flow	Income	-0.56	0.90

This table identifies the highest correlated category pair and the level of correlation for each category. The process follows a PCA transformation of the first component for each category after which a correlation analysis ensued. The PCA Category deviation is a metric that identifies the variability of the PCA to identify the extent of the diversity inside the category. A measure larger than 1 is indicative of large variability inside a category, bringing the similarity of the variables in the category under question. If all variables in a category can be justified, then a higher PCA Category deviation is usually a good thing, as uncorrelated variables lead to improved performance. A full spectrum of the category correlations is presented in *Figure 19*. Solvency can further be divided into categories such as capital structure, interest coverage, and cash flow ratios. The footnotes of *Table 12* show the ranking order of such a split. The above analysis, however, shows that this is not necessary as Solvency ratios are currently uncorrelated with other categories and such a split may cause new correlation concerns. Furthermore, the variables to the Solvency category are shown to be similar in type as measured by the PCA category deviation score; for that reason, the category should preferably not be split in this study.

If a researcher feels the need to, Assets and Liabilities can be divided into current and long-term categories to lower the deviation. However, for this study, such a grouping is not necessary as both of those groups are indeed rational constituents of the overall assets and liabilities grouping, and instead of slicing the established categories, the reader can refer back to *Table 22* to gain an understanding of what *type* of assets and liability values are the most important variables. The same can be said for liquidity measure that could be further divided into current ratios and conversion cycle type ratios.

Figure 19 expands on the most correlated categories as reported in *Table 25* to include the correlations of all categories to each other. The figure shows that the categories are largely uncorrelated. There is only a small number of correlated categories that will be dealt with from *Table 27* onwards.

Figure 19: Correlation on the PCA Transformation of Categories



I apply a PCA to the variables in each category and select the first principal component to represent that category. Reported above is the correlations between the first principal components of each category. This table shows that there is minimal correlation between the majority of the categories, but that some categories seem to be strongly correlated such as assets and liabilities and valuation and profitability measures.

Table 1 implements an inductive form of hypothesis testing used by Mullainathan and Spiess (2017); the process works this way: include all possible variables in a machine learning algorithm, and then remove the category or variables you believe to be important, retrain the model, readjust the hyperparameters and look at the model performance without that category or variables. From left to right the models in *Table 1* exclude Assets and liabilities, Solvency, Income, Equity and Profitability, and Valuation variables while calculating the AUC score at each point with three different measurement methods. I, therefore, start with a complete model and work my way back to identify each group of variables' relative contributions to the ROC.

The approach is useful as it goes beyond the task of predicting and also decomposes the importance of the type of variables in the study, providing added value to the literature. This method solves the issue of multiple correlated variables leading to prediction success. The results of this analysis show that asset & liability related values contribute the most to the model followed by solvency ratios, income values, profitability & valuation ratios, and equity values. This is one of the best tests to show the importance of the different groups of

input variables empirically. In contrast to *Table 24*, which is the mere categorical summation of variable importance, this table involves whole new tests. It is also interesting to note that the relative contribution (RVI) falls flats much quicker under this method across categories, possibly highlighting that only a few categories are needed to effectively predict bankruptcies.

Table 26: Reverse Induction Test

ROC (AUC)	(1) Full	(2) A&L	(3) Solvency	(4) Income	(5) Equity	(6) P&V
A - All	0.959	0.942	0.944	0.948	0.955	0.952
B - CV	0.947	0.930	0.935	0.937	0.943	0.941
C - Time CV	0.957	0.940	0.942	0.946	0.954	0.950
D - Average	0.954	0.941	0.944	0.947	0.955	0.951
Relative Contribution (R)	-	100	82	63	22	39

This table compares the various sets of variables against each other using an inductive testing technique to identify the importance of groups of variables that explain the model success. (1) is the performance of a model that contains all the variables. (2) - (6) removes variables that fall within the respective asset & liability, solvency, income, equity, and profitability & valuation categories from the model after which the model is retrained and tested to identify the extent to which each category contributes to the model. The following relative contributions are unreported in the table: expense (R 10), efficiency (R 8), liquidity (R 8), other (R 8), cash flow (R 4). The relative contribution is calculated by using the average of three different performance techniques to ensure the robustness of the results. (A) the first validation method is an 80% train and 20% split result in time-series. (B) is a random 10-fold performance validation split. (C) is a variant of the 10-fold performance split but in time series; it is arguably the most robust method. (D) is the average across all three methods, and the value used to calculate the Relative Contribution which is a normalised value out of 100 of the predictive power lost after dropping the category. All these splits are implemented after the validation and model development steps.

Table 27 reports the ranking of variable categories using nine different methods. This analysis is important to the overall study, as it seeks to identify which groupings of accounting dimensions truly come out on top in predicting future bankruptcies. Similar to *Table 1*, the new categories that combines assets with liabilities and profitability with valuation ratios are used. The table orders the categories according to the final ranking. All the italicised categories are accounting ratios. This is done because in *Table A42* it can be seen that past literature has focused on ratio measures; the purpose of the table is to compare the results of this study with that of past literature. If we isolate the *ratios*, as has been done in some past papers, then *Solvency* ratios come out on top followed by *Valuation and Profitability*, *Efficiency*, *Other* and *Liquidity* ratios. This result goes against a lot of low-dimensional and even some higher dimensions prediction analysis studies that stress the importance of *liquidity* measures over *Valuation and Profitability ratios* (Kim & Upneja, 2014; Mselmi et al., 2017).

On the full sample, it can be seen that the Asset & Liabilities and Solvency categories are the most important groups to predict bankruptcies. It is interesting to note that cash flow measures are not that important in predicting bankruptcies. Past researchers have also noted that Cash flow measures do not provide incremental prediction power over accrual-based measures (Casey & Bartczak, 1985). It is clear that ratios and fixed values both have high importance in the predictive model.

Table 27: Category Importance Analysis

Category	(1) CI	(2) RCLI Top 50	(3) Post GFC RCLI	(4) Pot	(5) F	(6) wF	(7) 24 CI	(8) PCA 10	(9) RIS	(10) Avg.	(11) Fin.
Assets & Liabilities	1	1	1	1	1	1	1	1	1	1.0	1
Solvency	2	3	2	2	2	2	3	2	2	2.2	2
Income	3	2	3	3	4	5	2	4	3	3.2	3
Valuation & Profitability	4	4	4	4	3	4	5	3	4	3.9	4
Equity	5	5	6	5	5	3	4	6	5	4.9	5
Expense	6	6	5	6	7	6	6	10	7	6.6	6
Efficiency	7	7	7	8	6	7	7	9	6	7.1	7
Other	8	8	8	9	8	8	10	8	8	8.3	8
Liquidity	9	9	9	10	8	9	9	5	10	8.7	9
Cash Flow	10	10	10	7	10	10	8	7	9	9.0	10

This table is an attempt to regroup categories where there is a strong correlation 80% + and to calculate the rank of the categories according to 9 different predictive importance strategies. This table calculates the equal weighted average of nine ranking methods (10). (1) is the normal importance measure (gain measure) calculated for all variables in every category. (2) is the gain measure for newly created categories using only the top 50 variables. (3) is the gain measure after the GFC. (4) is the ranking according to the potency measure, being the category importance weighted by the number of variables in each category. (5) is a measure (FScore) that tallies the amount of variable splits across all trees within each category. (6) measures the FScore weighted by the probability of the splits to take place. (7) is the overall gain measure for a newly created model that only incorporates the 24 best variables. (8) is the importance of the first PCA component for each category. (9) is the relative importance measured by Shapley value contribution. (10) is the equal-weighted rank average for each category. (11) is the final ranking obtained by ranking the raw average. When percentage growth measures were removed from the categories, all categories remained unchanged apart from a swap between *Other* and *Liquidity*. A further split in category where solvency ratios were split between capital structure and coverage and cash flow ratios resulted in the following rank among categories: (1) asset and liabilities (2) income (3) *valuation and profitability*, (4) *capital structure*, (5) equity, (6) *interest coverage,* (7) expense, (8) *efficiency*, (9) *cash flow* ratios, (10) other ratios (11) *liquidity ratios*, (12) cash flow values. The ratio values are italicised.

The vast majority of past high dimensional research has identified Solvency and Solvency related variables as the most important ratios (*Table A42*). The analysis in *Table 27*

corroborates its importance. Some studies have, according to this study's analysis, underreported the importance of Valuation and Profitability ratios (Kim & Upneja, 2014; Mselmi et al., 2017). As illustrated by the table, there is very little disagreement in the significance of both Solvency and Valuation & Profitability ratios throughout all methods (1) - (9). The same cannot be said for efficiency and liquidity measures. The final ranking of the Efficiency and Liquidity ratios is quite interesting because, to my knowledge, only two studies have noted that efficiency ratios take importance over liquidity ratios, and both are also high dimensional model studies (Jones et al., 2017; Mselmi et al., 2017). Other than these two studies, very few studies incorporate the efficiency dimension as prediction inputs. Furthermore, only three of the high dimensional studies show the same *ratio* ranking as has been reported in this table in column (11) (Behr & Weinblat, 2017; Jones et al., 2017; Volkov et al., 2017).

VIII. Interaction Analysis

The issue with many machine learning models is that their nonlinearity makes it hard to enforce monotonicity constraints to identify the direction of the relationship between independent variables and the machine-learned response function. In ML, the response can change in a positive or negative direction and at varying rates for changes in an independent variable, making the interpretation of feature importance much harder than simply looking at the coefficients of a linear model. Although the importance or contribution of a feature can be very valuable to understand, the measure does not identify the average direction and size of a variable to a response, nor does it attempt to explain nonlinear movements, which is of interest to researchers and industry experts.

To identify the relationship of variables in a machine learning algorithm, we can make use of a technique called partial dependence. Partial dependence allows us to see into the 'black-box.' Plotting the partial dependence, also known as marginal effect plots, produces information associated with both the direction and the strength of the relationship between explanatory and the outcome variables. Partial dependence plots (PDPs) are the visualisation of fitted functions. PDPs show the effect of variables on the response after accounting for the average effects of all other variables in the model (Friedman, 2001; Friedman & Meulman, 2003). PDPs offer the means of identifying the marginal dependence between the outcome and variables (Hastie et al., 2009). The basic premise of this technique is to obtain a prediction for all unique values of a variable while accounting for the effects of all the other variables to detect nonlinear relationships without the need to pre-specify them.

See Appendix *Method A* 4 in the first chapter for an expanded explanation and the mathematical formulae driving this concept and *Figure A27 - Figure A31* for examples of partial plots.

In this first part of this partial dependence analysis, we will first look at the top variables as highlighted in *Table 22*: Pre-tax Income (PI), Income Before Extraordinary Items (IBC), EPS Excluding Extraordinary Items (EPSPX), Price to Sales (PS), and Total Liabilities (LT). These variables have been individually plotted in Appendix Figure A27, *Figure A28, Figure A29, Figure A30* and *Figure A31.* These figures show their marginal association with the likelihood of bankruptcy. All variables in *Table 22* have followed a similar analysis to identify their respective directions (D), but only the first five are graphed. From the analysis of these five, the following directional relationships has been identified. The likelihood of bankruptcy decreases as PI, IB, EPSPX, and PS increase. The PI, IB, and EPSPX plots display a bimodal distribution around the zero bound, and all of these predictor plots show that extreme negative values are associated with failure.

The relationship for IBC is slightly more complex as values that are only slightly negative are not as concerning and tend to be classified as healthy. Furthermore, the partial dependence plots for PS show a monotonous increase in the likelihood of the firm being healthy as the value increases. A firm with a high PS value is likely to have a good profit margin and likely to be at the top of its industry with a lot of prospective growth. For that reason, it makes sense that the model identifies this positive relationship. As well, if we consider liabilities, it seems that the larger the LT, the more likely the firm is to be classified as bankrupt, all else equal. What is especially clear is that for the very low value of LT, the likelihood of being healthy is very high.

The next set of table interactions are noteworthy, *Table 28* looks at interaction pairs. Since the start of bankruptcy prediction, researchers have been interested in identifying the interactions between variables and how they affect the likelihood of bankruptcy. For example, FitzPatrick (1932) reported that less emphasis should be placed on liquidity ratios for firms that have long-term liabilities. And only in recent years have algorithmic techniques caught up to allow researchers to empirically identify the most important interactions. This is the first study to list the top interactions of a high dimensional model. Also, this is the first study to report interactions as far as depth-3, showing the non-linear relationship between three variables and how that relationship extends to the prediction of bankruptcies. It is also known that interaction effects add substantial explanatory power to bankruptcy models (Jones, 2017). Interactions are not given enough attention in bankruptcy prediction literature.

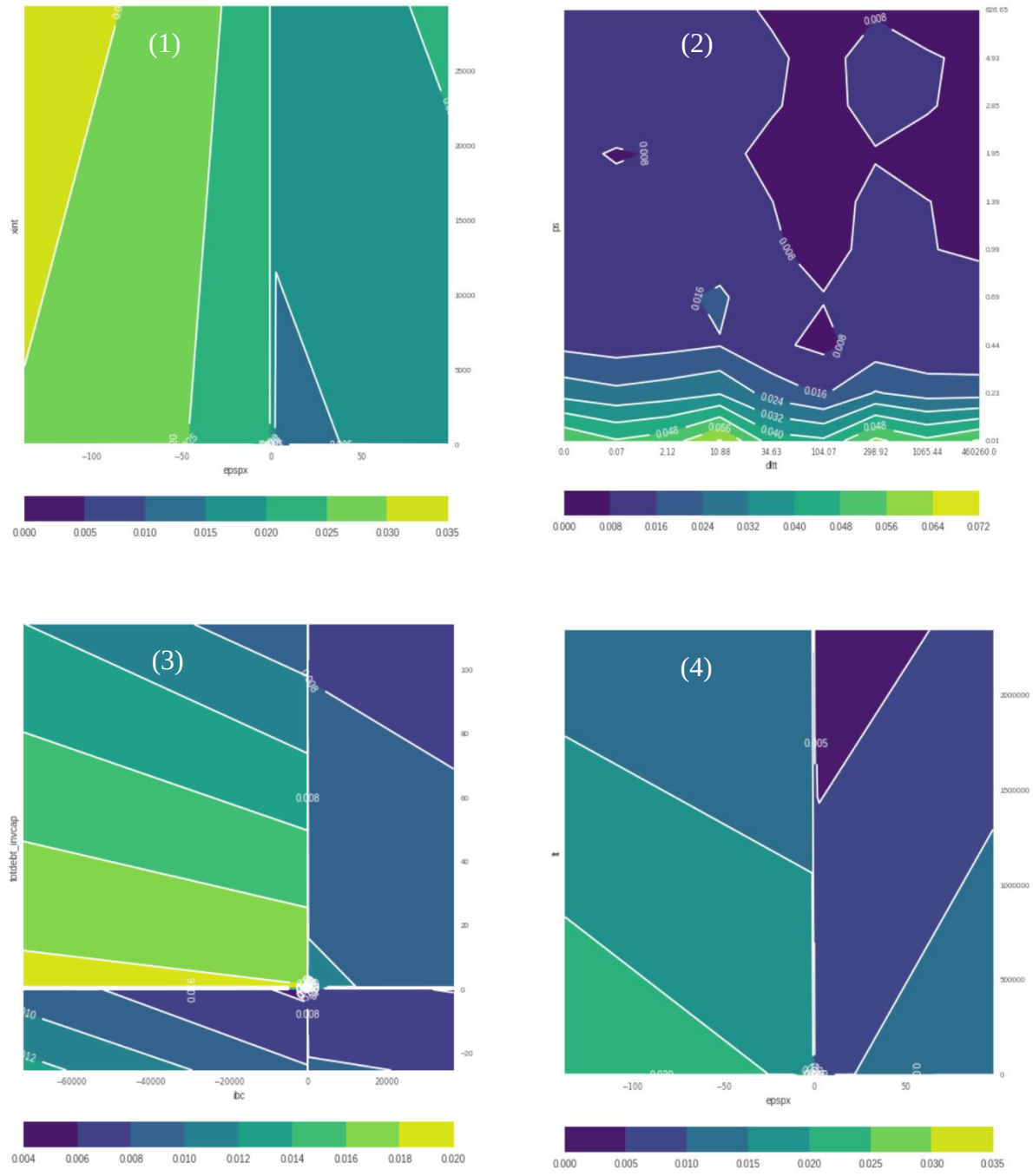
The benefit of the XGBoost model is that it detects important interactions across the full set of explanatory variables. Many of these relationships are relatively self-evident, but some are surprising.

The top left plot in *Figure 20* (1) shows that in situations where EPS is extremely negative and total expenses surpass $5 million, the likelihood for bankruptcy increases significantly all else equal. The relationship between the price-to-sales and long-term debt ratio (2) is also noteworthy as there seems to be a higher dimensional relationship that creates two distinct local maxima. Long-term debt of $10 million and $300 million seem to be two critical points in the prediction model. Values that fall in between $10 million and $300 million combined with a stable PS value are associated with lower rates of bankruptcy. There seems to be a clustered "sweet-spot" that cannot be explained by these variables alone. The plot between Total Debt to Invested Capital (TDIC) and Income Before Extraordinary (IBC).

Items (3) is also interesting as it is quite evident how the variables are playing off each other. Where IBC is larger than zero, it does not matter what TDIC is; for the most part, the likelihood of bankruptcy is not going to increase too much. Value investors have been especially interested in the TDIC ratio. When invested capital is negative, it means that all fixed assets and working capital less interest-free liabilities is negative. This can have both good and bad implications. If a company is growing at its core (e.g., IBC), it means that you do not have to invest any money, and if nothing changes, you can use the extra money from interest-free liabilities to grow your business. Where TDIC is negative you see a smaller likelihood of going out of business, and this shoots up immediately as it becomes positive, just to taper down again as it becomes more positive.

Plot (4) shows that low total liabilities (LT) and low earnings per share (EPS) values are an indication of a future bankruptcy. This is likely due to the failing firm having paid off a lot of debt without being able to pay off the last debt as a result of not achieving good returns (EPS). Firms that can obtain large amounts of LT that have negative EPS are, however, also at higher risk of becoming bankrupt.

Figure 20: Interaction Pair Partial Dependence Plots (Depth Two)



Plot (1) at the top left is the interaction between Interest and Related Expense (xint) and the EPS Excluding Extra. Items (epspx) and resulting response. Top right (2) is the interaction between Price to Sales (ps) and Long-Term Debt (dltt). Bottom left (3) is the interaction between Total Debt to Invested Capital (totdebt_invcap) and Income Before Extraordinary Items (ibc) and the interaction effect on the bankruptcy outcome. Bottom right (4) is the interaction between Total Liabilities (lt) and the EPS Excluding Extra. Items (epspx).

Although there are thousands of relationships, I only present the most predictive interactions in *Table 28*. I further report the direction of the relationship to the response outcome. The results indicate that a small amount of interactions is responsible for the majority of the predictive power of the model.

Table 28: Depth 2 - Interaction Analysis

Term 1	Sign	Term 2	Sign	RII	Gain
ibc	-	totdebt_invcap	+	100	779
epspx	-	lt	+	90	704
epspx	-	xint	+	75	585
dltt	+	ps	-	71	551
debt_assets	+	pi	-	65	509
pi	-	xint	+	54	418
ppent	-	ps	-	50	390
ps	-	ps_prtc	-	43	338
dpc	+	ps	-	36	279

Out of the top 50 variable list, there are millions of ways to conjure up directional relationships. Due to the nature of nonlinear relationships, to conceptually understand the web of relationship, it is best to identify the top interaction pairs. This table represents the most important interaction pairs as measured by the gain statistic at an interaction depth of two. For easier reading, I also report the relative interaction importance (RII). The sign simply indicates the average direction of each variable. The interaction terms are much more informative than single standing variables. Interactions are at the core of what gradient boosting tree models are all about. *Terms 1* are Income Before Extraordinary Items (ibc), EPS Excluding Extra. Items (epspx), Long Term Debt (dltt), Total Debt to Total Assets (debt_at), Pre-tax Income (pi), Property Plant & Equipment (ppent), Price-to-Sales (ps), Depreciation and Amortization (dpc). *Terms 2* are Total Debt-to-Invested Capital (totdebt_invcap), Total Liabilities (lt), Interest and Related Expense (xint), Price-to-Sales (ps), Pre-tax Income (pi), % Change in Price/Sales (ps_prtc). Whether the components to the pairs are in the first or second column is of no consequence; the same value would be reported if the columns swapped, as it is an interaction between values.

In *Table 29*, I further present the interaction results in a more visually appealing way without each variable's directional relationships to bankruptcy. The table also highlights the insignificance of some interactions that are incidental to investigating the top interactions in a cross-tabular fashion.

Table 29: Cross Tab - Top Variable Interactions

	totdebt_invcap	ps	lt	xint	pi
ibc	779	704	63	45	13
pi	66	585	209	338	0
epspx	228	76	551	509	34
dltt	17	418	156	34	14
debt_assets	43	127	239	77	390
ppent	71	279	82	28	61

This table represents the most important interaction pairs as measured by the gain statistic at an interaction depth of two. The table has been constructed to highlight the top ten interactions. For completeness, the surrounding interactions have also been included. Variables vertically follow: Income Before Extraordinary Items (ibc), Pre-tax Income (pi), EPS Excluding Extra. Items (epspx), Long Term Debt (dltt), Total Debt to Total Assets (debt_assets), Property Plant & Equipment (ppent). And horizontally, Total Debt to Invested Capital (totdebt_invcap), Price to Sales (ps), Total Liabilities (lt), Interest and Related Expense (xint).

In *Table 30*, I further highlight the interactions between three variables. Similar to previous tables, for all the interaction signs, + means an increase in the likelihood of becoming bankrupt, - means a decrease in the likelihood of becoming bankrupt. I will describe the most interesting and somewhat unexpected interactions of the table. For the second interaction (2), as income (pi) increases, a higher debt to assets ratio (dept_at) becomes less of a concern. When research and development to sales (rd_sales) is high, the effect of the asset ratio on the outcome is less consequential, whereas the level of income still has a big effect on the outcome. This is very interesting as it shows that firms that have large R&D programs are unlikely to become bankrupt, all else equal. Some researchers have historically argued the opposite and said that there is a 'failure-inducement effect' in a firm's effort to push for innovation efforts when performance falls (Antonelli, 1989). This also makes intuitive sense as management would be less inclined to believe in the future of their company if this value was low. The causal link, like all of these interactions, remains somewhat uncertain. What can be said is that if this relationship with bankruptcy is purely the result of having large amounts of *disposable* income then you would expect other ratios and values in this study to show more importance (advertising/sales, reserves, dividends,

purchase of stock), but they do not even feature in the top 50 most important variables. For that reason, I argue that R&D, as a core activity, is essential to the longevity of a company. It might however, also be possible that R&D, is a much stronger signal that a company has additional cash to spend compared to values like advertising, dividends and purchase of stock and that R&D is the first place they would go to cut back on spending when they are in financial distress.

(5) - (6) tracks the relationship between long-term debt (dltt), income (ibc, pi) and price to sales (ps). When a firm's long-term debt is large while simultaneously having negative income and a low price to sales ratios, it is much more likely to be declared bankrupt in the future. These two interactions are collectively much more important than the next biggest interaction. It should be noted that the combined interactions lead to the final classifications and that they should not be used on their own. For example, if a firm has long-term debt and large sales compared to their valuation but are not profitable in the short-run, then they are more likely to be a failing firm, but a simple out-of-sample screening of these types of firms shows that among 'struggling firms,' it picks up companies like the Ford Motor Company and Hewlett Packard (10 Oct 2017). Only time will tell whether these firms are truly in financial distress using such a simple heuristic as the aforementioned interaction, but it is highly unlikely as the fully built out model considers hundreds of thousands of interactions more before making a prediction.

Table 30: Depth 3 - Interaction Analysis

	Term 1	Sign	Term 2	Sign	Term 3	Sign	RII	Gain
(1)	epsfx	-	ibc	-	totdebt_invcap	+	100	456
(2)	debt_at	+	pi	-	rd_sale	-	95	435
(3)	dpc	-	equity_invcap	+	ps	-	92	419
(4)	ibc	-	ps	+	totdebt_invcap	+	88	402
(5)	dltt	+	ibc	-	ps	-	84	383
(6)	dltt	+	pi	-	ps	-	84	382
(7)	ibc	-	ps	-	ps_prtc	-	83	378
(8)	ibc	-	ibc	-	totdebt_invcap	+	79	362
(9)	dltt	+	ps	-	txt	+	76	348
(10)	ibc	-	ppent	+	ps	-	74	336
(11)	at	+	debt_at	+	epspx	-	68	310

Out of the top 50 variable list, there are millions of ways to conjure up directional relationships. Due to the nature of nonlinear relationships, to conceptually understand the web of relationship it is best to identify the top interaction pairs. This table represents the most important interaction pairs as measured by the gain statistic at an interaction depth of three. For easier reading, I also report the relative interaction importance (RII). The sign purely indicates the average direction of each variable. The interaction terms are much more informative than single standing variables. Interactions are at the core of what gradient boosting tree models

are all about. Unique *Terms 1* are EPS (Diluted) - Excl. Extra. Items (epsfx), Assets - Total (at). Unique *Terms 2* are Common Equity/Invested Capital (equity_invcap). Unique *Terms 3* are Research and Development/Sales (rd_sale) and Income Taxes - Total (txt).

After surveying past literature, it has been noted that no study has made use of PCA transformations to look at high-level interaction effects between different accounting dimensions. Although this abstracts a lot of the minute interactions away, it still offers an important view of the importance of using different accounting dimensions to predict future bankruptcies. In *Table 31,* I present the most important category interactions. The three most important interactions at depth two are Assets & Liability with the Solvency dimensions (934) and Assets & Liability with the Profitability and Valuation dimension (658). This analysis further emphasises the importance of including fixed accounting values in higher-dimensional bankruptcy models. The analysis shows that the most important interactions occur between a fixed value and ratio dimensions.

Table 31: Cross Tab - Category Interactions

	Asset & Liab	Cash Flow	Efficiency	Equity	Income	Liquidity	Other	Profit & Value	Solvency
Asset & Liab	0	446	31	102	603	573	122	658	934
Cash Flow	446	0	143	267	69	241	188	220	34
Efficiency	31	143	0	279	63	95	157	23	59
Equity	102	267	279	0	335	73	211	420	90
Income	603	69	63	335	0	288	59	578	528
Liquidity	573	241	95	73	288	0	116	283	381
Other	122	188	157	211	59	116	0	465	82
Profit & Value	658	220	23	420	578	283	465	0	88
Solvency	934	34	59	90	528	381	82	88	0
Total	3472	1608	851	1779	2547	2050	1400	2740	2196

This table represents all the interaction pairs between the first PCA component of each category dimension as measured by the gain statistic at an interaction depth of two.

IX. Filing Outcomes

The prediction of all filing outcomes is contingent on a correctly predicted bankruptcy outcome. In this section, I use a GBM model with simple accounting value inputs the year before the filing date to predict bankruptcy outcomes. The summary statistics of filing outcomes can be found in *Table A48*. As mentioned in the *Literature Addendum*, filing outcomes have great economic consequence for creditors and shareholders. Stakeholders not only want to know the likelihood of a litigated bankruptcy occurring, but also all the filing outcomes associated with the predicted bankruptcy. In *Table 32*, I present the performance of five different filing outcome models. The first of these five is the chapter prediction model. It involves a prediction task of whether the bankruptcy will finally be filed under Chapter 7 or Chapter 11. The Chapter prediction model performed the best of all other filing outcomes models. It achieved an AUC of 0.88. The survival prediction model that identifies whether the firm would emerge from bankruptcy performed second best with an AUC of 0.73. The task that attempts to predict whether assets will be sold in a 363 Asset sale or by other means came in third with an AUC of 0.64. The duration task, which involves the prediction of whether the bankruptcy proceedings would endure for longer than one year, came in second to last with an AUC of 0.62. And lastly, the tort task had an AUC score of 0.54, which is only slightly higher than random. All prediction tasks performed better than random guessing.

Table 32: Binary Classification Performance for Predicting Bankruptcy Characteristics

Binary Classification Model	ROC AUC Score	Accuracy Score	Average Precision Score	False Positive Rate	False Negative Rate
Duration	0.62	0.56	0.69	0.66	0.26
Survival	0.73	0.69	0.80	0.61	0.12
Chapter	0.88	0.95	0.70	0.05	0.20
Asset Sale	0.64	0.66	0.39	0.27	0.55
Tort	0.54	0.90	0.17	0.05	0.83

This table reports six important metrics for five alternative classification tests to predict the outcome of predicted bankruptcies. *Duration* classification is the first task to predict the binary outcome. This task involves the prediction of whether or not the disposition will take longer than a year after the initial filing. *Survival* predicts a binary outcome as to whether or not a firm will re-emerge out of bankruptcy and remain in business for longer than 5 years. The *Chapter* task predicts whether the bankruptcy filing will be converted to Chapter 7 or whether it will be a Chapter 11 filing. The *Asset Sale* model predicts whether the debtor will sell all or substantially all the assets during the Chapter 11 proceedings. The *Tort* classification task seeks to predict whether the bankruptcy will occur as a result of tortious actions such as product liability, fraud, pension, environmental, and patent infringement claims. The above metrics have been fully defined in table X.

Table 33 identifies the most important variables to each of the prediction tasks, including the most important categories. To obtain the categories, I apply a PCA to the variables in each category and select the first principal component to represent that category. The model column includes the multiple outcomes for which models were created.

Table 33: List of Each Outcome Model's Most Predictive Variables and Categories

Model	Selection	RI - Rank 1	RI - Rank 2	RI - Rank 3
(1) Duration	*Variable*	Cash and Short-Term Investments (CHE)	Net Profit Margin	Inventories - Total (INVT)
		- 100	+ 86	+ 86
	Category	Asset & Liab. + 100	Profit & Value + 48	Solvency + 15
(2) Survival	*Variable*	Current Liabilities/Total Liabilities	Asset Turnover	Investment and Advances - Equity (IVAEQ)
		- 100	+ 50	+/- 50
	Category	Assets + 100	Solvency + 36	Income + 27
(3) Chapter	*Variable*	Receivables Turnover	Payables Turnover	Current Liabilities - Total (LCT)
		+ 100	- 57	+ 43
	Category	Efficiency + 100	Solvency + 26	Asset + 10
(4) Asset Sale	*Variable*	Current Liabilities/Total Liabilities	Depreciation and Amortization (DPC)	Long-term Debt/Total Liabilities
		+ 100	- 93	- 93
	Category	Asset + 100	Profit & Value + 75	Solvency + 55
(5) Tort	*Variable*	Accruals/Average Assets	Minority Interest - Balance Sheet (MIB)	% Change in Forward P/E to 1-year Growth (PEG) ratio
	RI	- 100	+ 67	+ 67
	Category	Solvency + 100	Profit & Value + 60	Asset & Liab + 60

This table reports the top three *Variables* and the top three *Categories* for five different binary classification tasks including the associated relative importance (RI) among the three values. Number (1) - (5) identifies the different filing outcome-classification models.

For each model, the most important variable and accounting category are highlighted. Furthermore, a relative importance measure and the direction to the outcome are also provided. *Table A44* further reports the filing outcome statistics of the most important variables. The results in *Table 33* are interesting for a few different reasons. The first is that these values give the reader some insight into what values the court takes into account or what values are associated with factors the court takes into account before ruling on filing outcomes. The good performance of these models leads me to believe that the inherent characteristics of the firm are important factors that affect the outcome of the bankruptcy. In the next few paragraphs, I will consider the most important variables for each classification model.

(1) Duration: The results of *Table 33* show that firms with increased cash and short-term investments, including increased inventory and a higher net profit margin, will, all else equal, spend more than a year on bankruptcy proceedings. The reason for this is possibly that high net profit firms are more complex to unravel. These firms are also worth saving and spending time on due to the enhanced prospect of creditors reaping the returns in the future if the firm gets sold as a going-concern.

(2) Survival: Firms with low current-to-total liabilities ratios, high investment and advances, as well as a good asset turnover ratio tend to survive the bankruptcy process. If you have proportionally low current liabilities, it means that there are fewer pressing demands in the short-run, which would allow a firm enough time to get back on its feet and re-establish itself as a going-concern. High investment and advances in affiliates, associates, and subsidiaries values indicate a larger interest in the success of the firm. The mere fact that affiliates or subsidiaries exist is enough of an indication that a firm suffering financial pressure can be "rescued" from bankruptcy proceedings by affiliates or subsidiaries. The subsidiaries can be sold to finance the survival of the firm under pressure. A high asset turnover ratio is indicative of a firm generating more revenue per unit of assets. Firms that are efficient, all else equal, will be more successful in emerging from the bankruptcy process; efficiency is likely to be an important factor the court takes into account.

(3) Chapter: If a firm's receivable turnover is low and payables turnover is high while current liabilities are high, then they are more likely to file for Chapter 7 bankruptcy. Firms that pay their average payables more frequently than they collect their average receivables are struggling with short-term liquidity. However, such a pattern or relationship is also indicative of firms giving greater importance to paying back their creditors than to collecting their own debt. Ch. 7 bankruptcy is often referred to as liquidation bankruptcy as these firms are past

the stage of reorganisation. Ch. 7 bankruptcy highlights the process of absolute priority. Trustees are appointed to ensure that the proceeds are paid to specific creditors. It therefore makes sense for these creditors to initiate the Ch. 7 process before the firm remits more money to lower-ranked short-term creditors in the form of payables while ignoring the first priority creditors. Similarly, priority or secured creditors would be concerned with firms that have a disproportionate amount of current liabilities and would initiate action to ensure that they see some of the proceeds coming their way first.

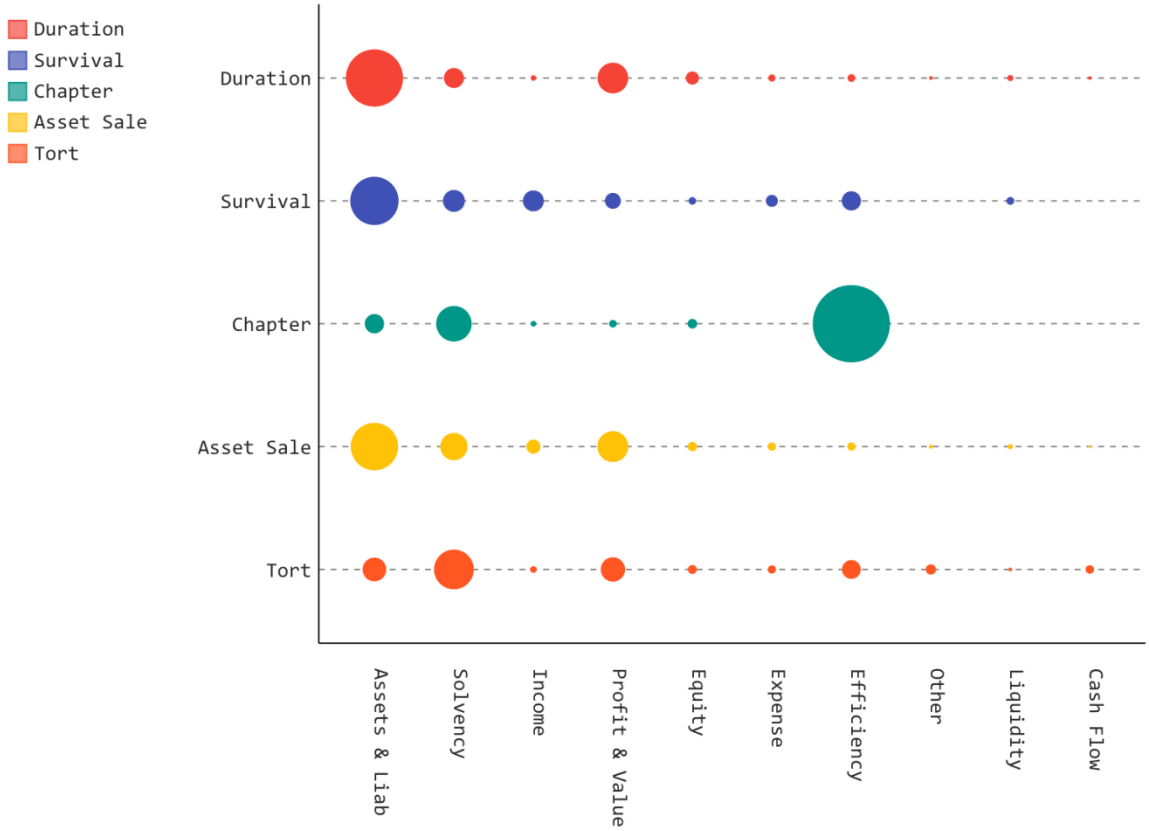
(4) Asset Sale: Firms with high current to total liabilities, and low depreciation and amortisation, and low long-term to total liabilities are likely to sell their assets in the bankruptcy process. The sale of assets is mostly deemed the sale of substantially all the assets of the firm of a Chapter 11 debtor. There are 228 observations of asset sales, and they include the sale of assets under Chapter 7, of which there are 24 observations. These type of asset sales are more commonplace in recent years. Similar to the previous sections, when a firm has a high current liability, the secured creditors put pressure on the company to sell off assets so that they can receive some of the proceeds, as this is the fastest way for them to receive adequate reparations. What is also interesting is that asset sales become more likely when the long-term to total liability ratio is low; this means that when the size of the secured long-term creditors is small, then the small group of creditors have the incentive to sell off the assets to recuperate more money instead of allowing for the reorganisation process to go ahead. This holds especially true when the depreciation and amortisation are low, which is indicative of a lower level of fixed assets making the sale of assets easier than reorganisation.

(5) Tort: Generally speaking, it is not easy to predict tortious events. Chaudhuri and De (2011) note that no models have yet been successful in detecting corporate fraud. More than half of the tortious filings relate to fraud; the other filings relate to environment and product claims. Although tort prediction in this study did not perform well, it showed some interesting associations. When accruals to assets are low and when the total percentage of minority interest is low, then there is a tendency towards a tortious bankruptcy. It could be that management knows that their company is under scrutiny, and for that reason, they underreport their accruals. It could also be the case that potentially tortious companies are selling off subsidiaries to protect them from future claims, increasing their minority interest, but without proper analysis this is mostly speculation.

The last part of this analysis includes a quick categorisation of all the variables to identify which accounting dimensions are the most important for predicting the various filling outcomes. *Figure 21* shows that the Duration of a firm is primarily driven by the

Assets & Liability, and Profitability & Valuation dimensions. Whether a firm would survive or not, is primarily driven by its level of assets. The chapter under filing is driven mostly by Efficiency ratios. And lastly, whether or not an asset sale or a tortious filing would transpire depends on a wide range of categories but primarily by the Assets and Liabilities Dimension.

Figure 21: Bubble Plot and Ranking of each Model's Most Important Categories



This figure reports the relative importance of the five outcome classification models and the associated accounting dimensions. There is a large amount of heterogeneity between the different classification models.

X. Conclusion

This study shows that a Gradient Boosting Tree Model (XGBoost) outperforms some of the latest deep learning networks (DCNN). It also shows that the creation of a meta-model (stacked model using RF, AdaBoost, DCNN, and FNN) outperforms all the individual parts. The study highlights the importance of not only using financial ratios but also including dollar accounting values as inputs to the prediction model. The overall model shows that Assets & Liability values and Solvency ratios are the most important dimensions in

predicting bankruptcy. These categories have been ranked using nine different importance metrics. The models developed in this study make use of accounting variables and a conservative sample. Notwithstanding this fact, the DCNN model used in this chapter is the best performing neural network compared to all past studies. Furthermore, the XGBoost model used in this study has comparable performance to the best models of past studies.

I find, similar to more recent literature, that less restrictive, higher-dimensional models with more variables will outperform most linear models in bankruptcy prediction tasks. A significant number of past bankruptcy studies show a small picture of the larger subset of relationships by using restrictive low-dimensional models and a small set of variables. A range of variables has been found to have a strong association with bankruptcy, many of which have not been noted by past research such as the level of stockholder's equity, inventory, depreciation and amortization, and the research and development to sales ratio.

A pitfall of the vast majority of the past higher-dimensional bankruptcy research is that they do not test the correlations between variables before identifying the most important variables, leading to invalid importance measures. A way to deal with the multicollinearity between variables is to categorise variables and to ensure that there is little to no collinearity between the first component PCAs of the distinct categories. According to the categorisations in this study and the associated ranking metrics, the most important dimension out of ten categories in predicting bankruptcy is Assets & Liabilities.

As a result of using a higher dimensional model, this study further reports on the most important interaction pairs that lead to bankruptcy predictions. This study is the first to list the top interaction *pairs* and is also the first study to list the most important interactions between *three* distinct variables. The study found the most important interaction pairs to be between income and the total debt-to-invested capital ratio, and between earning-per-share and the total-liabilities ratio. The most important depth-three interaction is between the earnings-per-share, income, and total debt-to-invested capital ratio.

A significant contribution of this study is the novel focus on litigated bankruptcies, i.e. Chapter 7 and Chapter 11 bankruptcies, which allows for an extended study to, not only predict the future occurrence of bankruptcy, but also to predict binary filing outcomes one year in advance of the filing event. The findings show that many of the filing outcomes can successfully be predicted. It is shown that the level of cash, short-term investments, and inventory strongly affect how long bankruptcy proceedings would endure; and that the current liability to total asset ratio, asset turnover ratio, and investments and advances significantly affect whether a firm would survive the bankruptcy process. Tests before and

after the GFC also highlight the dynamic nature of variable importance over the years. The performance of the models changes over time; although the average performance is around 0.957, it ranges from as low as 0.917 to as high as 0.984 ROC (AUC) over different periods. The general trend is that the more data that is included, the better the performance. Other effects of the change in performance over the years include differences in the underlying distribution of bankrupt to healthy firms.

Overall, this study has identified new ways to predict bankruptcies (DCNN); it has shed light on the equivalent importance of accounting values and ratios in bankruptcy prediction when high-dimensional models. The study uncovered variable heterogeneity between time intervals and highlighted the importance of interactions for high dimensional models. It is the first study to predict not only bankruptcy as an outcome but also the associated filing outcomes (duration, survival, asset sale, filing chapter, tort). This study is the first stride towards successfully using high dimensional models to improve both prediction quality and variable analysis.

In the future, it would be interesting to know whether a model can predict the potential refiling of a firm that emerged out of a past bankruptcy. In the same breath, it would be fascinating to see whether the long-term survivability of such firms can accurately be predicted. Other noteworthy prediction tasks could include a classification model that can predict whether a plan for bankruptcy is pre-packaged or whether the plan was simply pre-negotiated, as this can have significant implications for creditors. Future bankruptcy research should also seek to create bankruptcy prediction models using causal specifications. This will be very helpful in developing theory within bankruptcy prediction. It would also be good to replicate the framework in this study to extend the model, to not only predict the future occurrence of bankruptcies, but also predict other risky occurrences such as liquidation events, financial distress, and mergers and acquisitions.

XI. Appendix

A. Summaries

Table A34: Bankruptcy Characteristics Over Defined Periods

(1) Interval	(2) Description	(3) Bankruptcies	(4) Survived (%)	(5) Tortious (%)	(6) Long Legal Process (%)	(7) 363 asset Sale (%)	(8) Total Assets (Billions)
1980s	Oil & Metal	48	0.77	0.13	0.94	0.08	257
1990s	Wholesale	143	0.64	0.08	0.64	0.15	242
Early 2000s	Dotcom Bubble	207	0.62	0.09	0.57	0.24	688
Late 2000s	GFC	115	0.54	0.03	0.44	0.40	1,788
Early 2010s	Oil & Electronic	104	0.55	0.02	0.24	0.28	289

(1) The intervals are selected according to periods of themed bankruptcies. (2) indicates the type of bankruptcies that occurred within the interval. (3) is the sample of bankruptcy filings in each region. (4) is the percentage of firms that emerged out of bankruptcy as a result of an agreed plan to re-emerge out of bankruptcy and remain in business indefinitely. (5) is the proportion of bankruptcy cases that resulted from claims for product liability, fraud, pension, environmental, and patent infringement claims. (6) is the percentage of cases in which the disposition took longer than two years after petition filing. (7) The 363 Sale occurs when the debtor sells off substantially all of its assets during the Chapter 11 proceedings; the court can then distribute the proceeds of sale or convert the case to Chapter 7. (8) is the report of the sum of all firm assets as reported on the most recent 10-k filing prior to the bankruptcy petition.

Table A35: Bankruptcy Characteristics Across Industries

Industry	(1) Bank-ruptcies	(2) Sur-vived (%)	(3) Long Legal Process (%)	(4) Tortious Bank-ruptcy (%)	(5) Average Duration (Days)	(6) 363 Asset Sale (%)	(7) Total Assets (Billions)
Agricultural	3	0.67	0.33	0.00	234	0.67	3
Construction	15	0.87	0.60	0.07	662	0.07	14
Finance	99	0.24	0.54	0.05	617	0.29	2,017
Manufacturing	191	0.72	0.58	0.11	618	0.29	524
Mining	47	0.66	0.30	0.02	299	0.17	75
Retail Trade	78	0.58	0.67	0.03	645	0.29	110
Services	72	0.64	0.42	0.08	411	0.18	122
T, C, E & G	90	0.70	0.57	0.02	570	0.17	375
Wholesale Trade	22	0.64	0.50	0.09	550	0.14	25

This table reports the distribution of various bankruptcy outcomes across industries. It reports the legal descriptive statistics as obtained from the firms' bankruptcy case filings. (1) is the sample of bankruptcy filings in each industry. (2) is the percentage of firms that emerged out of bankruptcy as a result of an agreed plan to re-emerge out of bankruptcy and remain in business indefinitely. (3) is the percentage of cases where the disposition took longer than one year after petition filing. (4) is the proportion of bankruptcy cases that resulted from claims for product liability, fraud, pension, environmental and patent infringement claims. (5) reports the average duration for bankruptcies in the industry. (6) is about the 363 sale that occurs when the debtor sold off substantially all of its assets during the chapter 11 proceedings; the court could then distribute the proceeds of sale or convert the case to Chapter 7. (7) is the sum of all firm assets as reported on the most recent 10-k filing prior to the bankruptcy petition.

B. Literature Addendum

a) Variables and Categories.

Gradient boosting (XGBoost) and many other machine learning algorithms allow for a free complexity parameter (Leathwick, Elith, Francis, Hastie, & Taylor, 2006). XGBoost measures model complexity by the depth of the tree as well as the number of trees and the learning rate. Algorithms gain a lot of strength from being able to choose complexity. These added dimensions of complexity are the hidden strength of the gradient boosting and other machine learning models' flexibility and performance. As a result, researchers can start out with a rich set of inputs and allow the model to decide what inputs to use endogenously. A known issue with added complexity is the proclivity of overfitting. Overfitting can be remedied by testing the model on an 'out-of-sample' dataset as far as the model is concerned (Witten, Frank, Hall, & Pal, 2016). Further explanation of these concepts appears in appendix XI.D.4 on page 150. Apart from allowing researchers to start with a richer collection of potential candidate variables, the enhanced model complexity and endogenous variable selection allow researchers to simply allow the data to identify the most important variables without human intervention.

The next step is to identify a blanket set of variable inputs for the model to analyse. The start of bankruptcy literature can be traced back to a report in the 1930s from the Bureau of Business Research (BRB). The BRB compared the ratios of 29 failing firms with the average ratios across the firms to identify the characteristics associated with failure. This rudimentary report showed the importance of ratios such as working capital to assets, reserves to total assets, net worth to fixed assets, sales to assets and cash to total assets in describing bankruptcies. More than 80 years later, the majority of predictor variable identification literature executes similar tasks. The literature did, however, evolve from simple mean comparisons and univariate statistical techniques between healthy and failed

firms to the development of prediction models and statistical significance tests on variables (Altman, 1968; Beaver, 1966; FitzPatrick, 1932). Even in the recent wave of machine learning models, predictor variable identification is still necessary for describing the occurrences of bankruptcies, albeit at a higher dimensional feature space (Behr & Weinblat, 2017; Jones et al., 2017; Mselmi et al., 2017).

After the BRB report, similar exercises soon followed. FitzPatrick (1932) was the first bankruptcy researcher to report that the interactions between variables have important consequences in bankruptcy prediction. He reported that less emphasis should be placed on liquidity ratios for firms that have long-term liabilities. Interactions are notably one of the great strengths of high-dimensional models. Unlike traditional linear models, interactions do not have to be created by hand, nor fathomed by mind like FitzPatrick did (1932). Further pioneering research includes the work of Chudson (1945) who was the first author to do a form of elementary pattern recognition. He showed that specific industry, profitability, and size groups lead to clusters of similar ratios. Future research went on to show that clusters of similar ratios can also be used to predict bankruptcy. These early studies are important theoretical contributors to the recent wave of higher-dimensional studies that use machine learning algorithms to predict bankruptcies.

Although past research has focused on the use of accounting (Zmijewski, 1984; Altman, 1968; Beaver, 1966), market (Duffie, Pedersen, & Singleton, 2003; Ohlson, 1980; Shumway, 2001), corporate governance, industry, analyst, and economic-based variables as inputs (Beaver et al., 2005; Jones & Hensher, 2004; Jones, 2017), this study simply focuses on accounting-based measures for the following reasons. First, additional model variables would consume the variable importance normally associated with account-based variables, due to strong correlations between categories of variables; second, a cornerstone to this study is the identification of symptoms in the US and not prediction accuracy per se; third, the model would also be applied to the prediction of alternative bankruptcy characteristic; and lastly, although past research has shown that models that only focus on accounting information do not provide the best predictive performance (Li, 2012), I believe that this can be attributed to model constraints. More recent high dimensional research shows that a model that incorporates Market Price, Ownership Concentration/Structure Variables, External Rating, Macroeconomic Variables, Executive Compensation, and Accounting Based Variables improves the model quality as measured by the ROC (AUC) score with only 4% as compared to pure Accounting Variables (Jones, 2017). In this study, I argue that all the other values are simply proxies to the pure accounting values. In saying that, the addition of these

variables should be attempted in future bankruptcy pattern studies so long as importance measures can be statistically motivated to discriminate between collinear variables to avoid falling into the aforementioned traps.

Beaver et al. (2005) noted that accounting-related variables have performed without issues over a long period. Many empirical studies have led to a great number of accounting-related variables that have been shown to have power in bankruptcy prediction. Aziz, Emanuel, and Lawson (1988) observe that ratios in bankruptcy studies are based on ad hoc pragmatism rather than sound theoretical work. This study consequently includes a blanket of more than 70 accounting ratios. In line with past research, this study includes financial values and ratios such as (1) Solvency ratios (Cash Balance to Total Liabilities, Total Debt to Assets, Capitalisation Ratio), (2) Liquidity ratios (Current Ratios and Cash Ratios), (3) Profitability ratios (Net Profit Margin, Return on Equity), (4) Efficiency Ratios (Asset Turnover, Payables Turnover), and Valuation Ratios (Price/Sales, Price/Book). Consistent with past research, the expectation is that changes in the above ratios will affect the probability of bankruptcy.

An important concept of high dimensional models is that interactions happen in many more ways than just the numerator-denominator interactions such as predetermined in ratios; for that reason, *control* variables in the form of (5) Assets (Total Assets, Current Assets), (6) Liabilities (Total Liabilities, Current Liabilities), (7) Equity (Stockholders Equity, Depreciation and Amortisation), (8) Cash Flow (Operating Activities - Net Cash Flow), (9) Expenses, and (10) Income are also included. These variables were selected based on the standardised financial accounts on Compustat. These variables do not just facilitate controls; they, in fact, have important interaction effects acting as pseudo-ratios without the restrictive linear constraint of the numerator-denominator relationships. The overall interaction effects produce far more interesting and predictive variable combinations than simple numerator-denominator combinations. The expectation is that the fixed values in aggregate will have more predictive power than the ratios. For that reason, apart from just studying the predictive ability of the ratios, attention should also be given to the interaction effects of both the ratios and 'controls.' For all fixed values and ratios, I also include ten of the most promising annual growth values, i.e. percentage changes in the value over the last year.

As important as individual variables are to this study, the vast majority of learning algorithms cannot discriminate between highly correlated variables. As a result, the categorisation of variables becomes an important step in understanding the true relationship between accounting-based values and bankruptcy events. Although a vast majority of the

papers on high-dimensional models include category labels, no study attempts to show empirically which categories or dimensions of accounting values and ratios are the most important to predict bankruptcies (Behr & Weinblat, 2017; Jones, 2017; Kim & Upneja, 2014). This paper makes use of nine different techniques to rank-order categories to obtain the final rank of the different dimensions. In this study, the categorisation is achieved by including variables by relatedness of theme rather than construction. For example, the liabilities category would include the dollar change in liabilities, current liabilities, and % change in liabilities values. The MDA approach, which pairs samples of failed and non-failed firms using financial ratios, generally shows that solvency, profitability, and liquidity indicators are the most significant indicators (Almamy, Aston, & Ngwa, 2016). However, the order of importance of these categories is unknown as past studies did not use a standardised set of ratios to measure the health of firms (Altman, 1968). In this study, I attempt to solve this problem by starting with a large base of variables classified into Solvency, Profitability, Liquidity, Efficiency and Valuation Ratio categories and allowing the machine learning model to decide what variables and categories are the most important by analysing the patterns in the data. I also establish categories for the fixed accounting values.

2. Models

Later studies from the 1960s to present contributed to the use of LDA and MDA prediction models instead of mean comparison studies. These historical and subsequent traditional statistical models are not of much consequence to this study apart from attempting to highlight the advantages and disadvantages of this group and new high dimensional machine learning models (see the appendix XI.E on page 154). I further compare the performance of the Logit model to other models in this study. The biggest benefit of the past model comparison studies is that they provide a framework for future bankruptcy studies.

In the early years of bankruptcy research, Merwin (1942) revealed that failing firms showed signs of deterioration as early as five years before failure. A few years later, Altman (1968) showed that by the use of an MDA model that he could predict insolvencies one to two years in advance. In this study, I make use of this knowledge and predict bankruptcies one to two years into the future. Within this definition, it is important to predict the year within which the company failed (Ohlson, 1980).

This study, unlike the majority of studies in this domain, predicts bankruptcies across a broad range of industries. The study also uses an XGBoost model that has many advantages

over linear models (Chen & Guestrin, 2016). Recent literature has shown that decision tree ensembles (multiple decision trees) and more specifically boosting ensembles (re-weighing tree importance to assist bad performing trees) almost always come out on top, with the added benefit that they are easier to conceptualise than many black box models such as neural networks (ANN) (Barboza et al., 2017; Jones, Johnstone, & Wilson, 2017b; Olson, Delen, & Meng, 2012; Zięba, Tomczak, & Tomczak, 2016). Olson, Delen and Meng (2012) compared five machine learning models, including ANNs, and found that the decision-tree related models outperformed. Recent evidence by Jones (2017) also highlights this outperformance. In the past, more research has been conducted in the univariate category than in the multivariate category, but that has slowly changed over the years. The structure and internal workings of these models are described in the appendix from page 147 onwards. However, it may be worth the effort to understand some of the concepts.

Due to recent advancements and the re-emergence of artificial neural networks (Barboza et al., 2017; Kim & Kang, 2010; Mselmi et al., 2017; Zhao et al., 2017; Zhou, 2013), I will also provide an alternative model to regular ANNs, in line with popular developments in deep learning[33] research. To do this, I make use of two different models: Feed-forward Neural Networks (Hornik, Stinchcombe, & White, 1989), and Deep Convolutional Neural Networks (DCNN) (Krizhevsky, Sutskever, & Hinton, 2012). I show that DCNNs outperform all past ANNs (refer to literature Table A40). The use of machine learning and deep learning in finance is becoming more common as researchers slowly uncover the nonlinearity of financial data. Callen et al. (1996) showed that machine learning models have long been able to beat time-series models in forecasting. Xiao et al. (2013) demonstrated the power of ensembles in financial market forecasting; they showed that the flexibility of the ensemble approach is key to their ability to capture complex nonlinear-relationships to predict future stock prices. This paper draws inspiration from recent machine learning applications in economics, such as those in papers by Mullainathan et al. (2017) as well as machine learning applications in price behaviour prediction (Bagheri, Peyhani, & Akbari, 2014; Teixeira, De Oliveira, & Adriano Lorena Inacio, 2010) and high-dimensional prediction studies (Jones et al., 2017).

[33] The multi-layered hierarchical representation of data using neural networks.

3. Predictive Power

With high-dimensional decision tree models, two potential candidates for measuring impurity to identify variable performance are Entropy and Gini Index. Gini importance makes use of the Gini Index whereas Information Gain makes use of Entropy as error measures. These values are often used interchangeably. A paper by Raileanu and Stoffel (2004) reports that these measures disagree only 2% of the time due to the similar nature in which they are calculated. Both of these measures are used in machine learning research. Gini importance is often preferred because it is less computationally expensive as it does not require a logarithmic calculation like entropy. Both of these measures attempt to measure the decrease in impurity or uncertainty that each variable provides. Both are data-centric approaches that look at the predictive ability of a parameter based on variable selection ranking in the nodes of the trees.

The benefit of the gain measure is that there are some empirically substantiated alternative measures that have been derived from the original measure; this includes measures such as the information gain ratio (Quinlan, 1986) and the expected gain measure (MacKay, 1992). There has also been a recent development in open source packages like Xgbfi that offer alternatives like average gain, which is the gain divided by the number of possible splits taken on a variable or variable interaction. In this study, I therefore make use of the Gain measure. The Gain measure is equal the number of times a variable is selected for splitting weighted by the squared improvement each split adds to the model average over all the trees (Friedman & Meulman, 2003). Friedman (2001) also established a relative measure that is essentially the contribution of each variable scaled to the performance of the best indicator multiplied by 100 (RVI). In this study, I report both the RVI value and the percentage Gain measure. The larger this number, the greater the effect a variable has on the response. The other measure used in this study is Split Frequency, which is the number of times a variable is selected for splitting in all decision trees. Frequency is presented as a simplified measure of gain.

4. Filing Outcome Prediction

The Bankruptcy Code ("Code") came into effect in 1978. The act afforded the debtor substantial protection against creditors but to different degrees depending on the filing outcome of the bankruptcy. It is a well-known fact that bankruptcy can be costly; LoPucki

and Kalin (2001) show that the direct costs for large public firms can amount to between 1.5-6% of a firm's assets. Given the disparity in cost between bankruptcies, this study argues that it is important to not just predict bankruptcies but also the associated characteristics of the proceedings.

Filing outcomes are important to all creditors and investors. Franks and Torous (1989) note that due to the cost and length of the proceedings in Chapter 11 bankruptcy, the legal and administrative costs may, in many situations, be against the interest of stockholders. Oliver Hart (2000) argues that there should be a greater push towards cash auctions as they are simple and efficient. However, a lot of bankruptcies still occur under Chapter 11 type structured bargaining, which is often costly and time-consuming as a result of the tensions between the parties involved.

The duration of disposition becomes costly for many parties including the executives, stockholders, and creditors. Predicting the duration of a bankruptcy disposition after the filing is therefore of important economic consequence. Results by LoPucki and Doherty (2008) show that case duration is an important determinant of the fees and expenses in large public bankruptcies. Li (1999) asked an important question, "Can this variation in bankruptcy duration be explained by the financial/industrial characteristics of the distressed firms?" In this paper, I seek to answer this question by identifying which accounting-based values have the most predictive power in predicting duration. I similarly ask the same question for all the other filing outcomes.

A 363-asset sale ("asset sale") can enable debtors to expediently and effectively separate a business, which is one of the core goals of the Chapter 11 reorganisation process. An asset sale can be appealing to debtors and creditors alike. In recent years, debtors have increasingly opted to sell their assets rather than restructure under the Chapter 11 process (Baird & Morrison, 2011). A few advantages include the ability of the purchaser to take the assets clear of liens and claims. It is also up to the debtors to pick favourable contracts. It does come with trade-offs such as the negative publicity of selling off assets (Sable, Roeschenthaler, & Blanks, 2006).

The Chapter under which the bankruptcy is filed is also an essential characteristic that shareholders and creditors want to be made aware of as soon as possible. A paper by Bris, Welch and Zhu (2006) shows that Chapter 7 liquidation (cash auctions) can be more expensive in direct costs and as expensive as Chapter 11 bankruptcies in indirect costs. Chapter 7 liquidation does not appear to be more expedient or cheaper than Chapter 11. Moreover, Chapter 11 seems to better preserve assets, allowing creditors to recover more.

The last filing outcome prediction task is whether the firm will file for bankruptcy as a result of tort claims. Torts involve large sums of money that can quickly overwhelm a company (Hardiman, 1985) Tort claimants qualify as creditors under the bankruptcy act and partake in the reorganisation process. Torts in the bankruptcy process are famous for the enormous future liabilities they can entail (Bibler, 1987). It is easy to see how this can undercut commercial creditors and shareholders, and it therefore has significant economic consequence and is worth knowing. This study shows that it is extremely difficult to predict whether a company that is predicted to file bankruptcy will file under tortious Chapter 11 bankruptcy.

A limited number of studies has looked at post-filing resolutions. Barniv, Agarwal, and Leach (2002) showed that knowing the outcome in advance can have immense economic consequences. They noted that significant abnormal returns can be earned if a firm emerges or gets acquired between the filing and disposition resolution, whereas liquidated firms experienced significant negative abnormal returns. The former experienced 155 percent abnormal returns on average, whereas the liquidated firms experienced negative 11 percent returns. Barniv et al. (2002) used a multi-labelled Logit model for predicting firms that will emerge from Chapter 11 bankruptcy. They correctly classified 62% of the firms.

In this study, I show a 70% accuracy using a binary XGBoost classifier to identify firm emergence, which is strictly limited to the use of accounting-related values. I believe that the results obtained in this study can be immensely improved by including additional variables such as the institutional ownership, the filling state, the court and the judge at hand, etc. Barniv et al. (2002) similarly noted that variables such as the resignation of executives, fraud, and other characteristics are important in predicting company survival. More recent research by LoPucki and Doherty (2015) recommended the use of information such as whether firms release press reports regarding bankruptcy and the headquarters' state.

Creditors and investors should not be satisfied with the mere prediction of firm distress and bankruptcy. Stakeholders or prospective stakeholders should have a model whereby they can predict not just the occurrence but also the terms of the bankruptcy in advance of the filing to improve risk management practices. The economic effect of the outcome is largely determined by the characteristics associated with the bankruptcy. As a result, the study extends the prediction to include important bankruptcy outcomes, such as how long the bankruptcy process will endure, whether the firm will successfully emerge after the bankruptcy period, whether the bankruptcy is tortious, and whether it will involve asset

sales; all this is done by solely evaluating the accounting variables before the bankruptcy filing.

C. Robustness

1. Performance-validation

Table 36 below reports some interesting results for this study. It has to do with the differences in model performance as a result of variations in the train-test splits. In summary, (2) time-split is the longitudinal split in the data to ensure that future information does not leak into the past; (3) is the cross-sectional randomised splits between the train and test sets. In this study, I make use of 10-folds or rounds. Lastly, (4) this time split fold is a unique combination of the TS method and the fold method. It is the most robust and accurate measure of true out-of-sample performance. The full design of this method is presented in the appendix, XI.D.4 on page 150. The time-split fold method, as a result of construction, produces metrics for many sub-periods. It is an effective method to look at the consistency of prediction quality over many years. The deconstructed results of this method can be found in the appendix *Table* A45 and *Table A46*.

Time split fold provides evidence of model performance not just over different time intervals but also for different levels of training data. The results of *Table* A45 on page 168 reflect two important generalisations of machine learning prediction in time series. The first is that with the inclusion of more data, the model tends to perform better (Domingos, 2012). But this does not always hold true for time series data; the reason is that that the learning, i.e., pattern recognition, can occur over different seasonal trends leading to worse future predictions. There are, of course, ways to mitigate this, such as incorporating seasonal indicators as variables in the model. In machine learning, it is desirable that the distribution of the train and test data is the same, but this is not always possible, especially with financial data (Montas, Quevedo, Prieto, & Menndez, 2002). The expectation is that the results will improve if the distribution remains unchanged. Splitting training and test sets by time intervals and checking for parameter stability over time is a very useful exercise in building a robust model and understanding how the model learns and predict. The best reported AUC of 0.984 occurred over the last few years of the 2014-2016 sample. The worst AUC occurred over the 2003-2004 period with an AUC of 0.917 (Table A45). Also note that the

performance of the models can be significantly affected by changes in the depth of the tree and adjustments to the underlying sample distribution, see Appendix 2 - Hyper-Parameters.

Table 36: Model Comparison Using Different Performance Validation Procedures

Metrics	(1) All Data	(2) Time-Split (TS)	(3) K-Fold (KF)	(4) Time Split Fold (TSF)	95% Confidence (+/-)
ROC AUC Sore	0.9587	0.9655**	0.9467**	0.9570	0.0142
Accuracy Score	0.9755	0.9837	0.9682	0.9712	0.0163
False Positive Rate (p-value)	0.0037	0.0069	0.0028	0.0039	0.0015
Cross Entropy	0.1414	0.0825	0.1301	0.1052	0.0707

This table compares the performance of the best models that resulted from different out-of-sample performance tests. (1) The original "All Data" model allocates 60% of the observation to the training set, 15% to the development of validation test set and 25% to the test set. The 15% is used to measure and improve the performance of the model. The observations to each of the splits are randomly selected. (2) TS is a simple ordering of the observation in time series and the creation of longitudinal training - 60%, validation - 15% and test set splits -25%; this method ensures that there is no information leakage from the future observations. (3) KF is a randomised cross-sectional method that scrambles the observations and splits them into training and test sets and calculates the average metrics from 10 different iterations or folds. (4) TSF is the most robust method and has also led to the model with the best generalisable performance as evidenced by the battery of metrics - It is a longitudinal blocked form of performance-validation that suits this form of bankruptcy prediction; it uses the strengths of both (2) and (3). All statistical comparisons are made against the model called "All Data." *p<.1 ** p<.05 *** p<.01. Significance levels are based on a two-tailed Z test.

2. Hyper-Parameters and Other Adjustments

Table 5 reports a few adjustments to the model and underlying sample, the first being an adjustment to the XGBoost tree-depth parameter. The model development process shows that the model performance is optimised at a tree-depth of 12. It is therefore expected that any deviation from this number, ceteris paribus, would lead to worse model performance; this has indeed been shown to be the case with a significant reduction in the AUC from 0.9587 to 0.9506. The distribution adjustment test shows that the model performs significantly better when there is an equal amount of bankrupt and healthy firm samples, which is indeed the approach the majority of the bankruptcy studies take.

A branch of literature has for some time been consumed by identifying whether market-based or accounting-based measures are better in predicting bankruptcy. Hillegeist et al. (2004) for example show that option pricing models can provide better estimates of corporate bankruptcy than accounting values. Some studies show that by combining the two

one can achieve more accurate results (Beaver et al., 2005; Shumway, 2001). The problem with market-based measures is that they may not necessarily be efficient, especially considering the fact that small firms form the majority of bankruptcy samples. The multi-dimensional approach in this study may put this argument to rest by identifying the high dimensional contribution of market-based and accounting-based measures.

In the next column, I remove valuation ratios that have a price component as it can be argued that they are market-based even though they have an accounting component. This did not result in any significant reduction in the AUC. The table shows that the model performs only slightly but non-significantly worse when one excludes price-related variables. A further correlation analysis in *Table 25* shows that valuation and profitability ratios are highly correlated. For that reason, as long as accounting-based profitability ratios are included, valuation ratios do not lead to significantly improved prediction performance. The last form of adjustment is the removal of growth or percentage change variables. This leads to an insignificant increase in the model AUC. The small change seen can be due to the additional noise created by potentially irrelevant growth variables.

Table 37: Model Comparison Adjusting the Type of Inputs and Model Parameters

Metrics	All Data	Six Depth	Distribution Adjustment	Sans Value	Sans Percentage Change
ROC AUC Score	0.9587	0.9506**	0.9661***	0.9560	0.9597
Accuracy Score	0.9755	0.9752	0.9339	0.9823	0.9784
False Positive Rate (p-value)	0.0037	0.0051	0.0201	0.0073	0.0075
Cross Entropy	0.1414	0.1518	0.1741	0.1424	0.1398

This table compares the results of the various model and sample adjustments. The first column includes the original model; the second column reports the performance of a model where the tree depth parameter is changed to six from the original 12. The third column reweights the sample distribution, the fourth removes valuation ratios, and the last column removes growth variables. All statistical comparisons are made against the model called "All Data." *p<.1 ** p<.05 *** p<.01. Significance levels are based on a two-tailed Z test.

3. Other Decision Tree Ensembles

Table 38 reports the additional performance of two alternative decision tree ensembles: the AdaBoost model and the Random Forest model. The AdaBoost model performs slightly better than the Random Forest model. The reason for this additional study is to ensure that XGBoost is the best decision tree-based model for the task at hand. The

XGBoost still outperforms these models. By comparing *Table 19* to *Table 38*, it is clear that the decision tree type models perform especially well in bankruptcy prediction. I hypothesize that if larger bankruptcy datasets can be made available, such that the overall sample of bankruptcies reaches the tens of thousands, then the deep learning models will show immense improvement and most likely beat the performance of the decision tree models. For that reason, I included both of these strains of high dimensional models.

The Stacked model in *Table 38* is quite interesting as it is a combination of the AdaBoost, Convolutional Neural Network, Feed Forward Network, and Random Forest Model into one big model that separately weights the importance of each model's predictions using a final Decision Tree model. Therefore, the combination of all models apart from the XGBoost model seems to perform close to the XGBoost model at the expense of being quite inefficient and expensive to run. To my knowledge, this is the first prediction study that has attempted a stacked model in an attempt to improve prediction quality. A further stacked model that includes the XGBoost model as input showed a statistical improvement over the original XGBoost model with an AUC score of 0.9642 and accuracy of 0.9778.

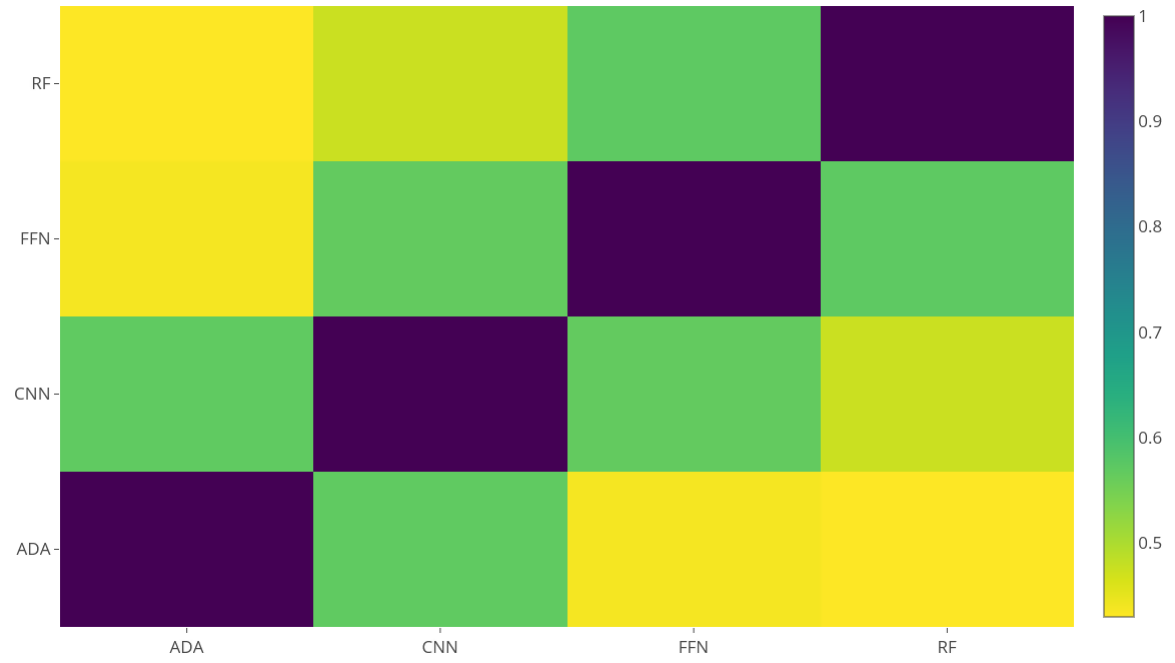
Table 38: XGBoost and Decision Tree Ensemble Model Performance Comparison

Metrics	XGBoost Model	AdaBoost Model	Random Forest Model	Stacked Model
ROC AUC Sore	0.9587	0.9291***	0.9275***	0.9495**
Accuracy Score	0.9755	0.9612	0.9576	0.9681
False Positive Rate	0.0037	0.0185	0.0370	0.0074
Cross Entropy	0.1414	0.1913	0.2409	0.1613

This figure illustrates the performance of the XGBoost model with two other tree ensemble models: AdaBoost and a Random Forest Model. AdaBoost and XGBoost are both ensembles that seek to convert weak learners into a single strong learner. AdaBoost adds weak learners according to performance by changing the sample distribution. In XGBoost, the weak learner trains on the residuals to become a 'strong' learner. Random forests are simply a multitude of decision trees. All three underlying models have decision trees as the base learner. The stacked model is a combination of the AdaBoost, Convolutional Neural Network, Feed Forward Network and Random Forest models into one big model by using the four models' predicted outcomes as inputs to a decision tree model. *p<.1 ** p<.05 *** p<.01. Significance levels are based on a two-tailed Z test.

Stacked models perform especially well when the respective predictions are uncorrelated. *Figure 22* presents a correlation map of these models' predictions. The AdaBoost model (ADA) and Deep Convolutional Neural Network (CNN) is the most correlated model pair. Although similar models in some respects, the AdaBoost and Random Forest models are relatively uncorrelated in their predictions.

Figure 22: Correlation of Predictions Across High-Dimensional Models



This correlation plot shows why the stacked model performed so well as compared to the individual component models. The stacked model is created by combining the AdaBoost, Convolutional Neural Network, Feed Forward Network, and Random Forest models into one big model by using the four models' predicted outcomes as inputs to a decision tree model. Stacking performs well due to its smoothing nature. Stacking is most effective when the based models are less correlated, which is the case for the above models. Stacking is also called meta-ensembles and can be seen as an advanced form of boosting. The results of stacking the models in the correlation map above can be found in *Table 38.*

4. Time and Variable Variants

Table 39 onwards only reports the results of the GBM model. The first column repeats the performance of the GBM models as presented in the previous tables. The second column reports the results of a model constructed out of just 50 of the top variables that have been identified in a later section of this study on a validation set using variable selection methods (*Table 22*). It is clear that the model can predict well even with a small number of variables. Further, although significant, the model performance does not change too drastically when we predict bankruptcies one to two years in advance instead of using all observations from both years.

Table 39: Model Comparison Using Different Inputs

Metrics	All Data	50 Variables Model	One Year Before Bankruptcy	Two Years Before Bankruptcy
ROC AUC Sore	0.9587	0.9408***	0.9666**	0.9434***
Accuracy Score	0.9755	0.9700	0.9860	0.9837
False Positive Rate	0.0037	0.0056	0.0010	0.0002
Cross-entropy	0.1414	0.1795	0.1282	0.2206

This table compares the performance of a model that includes only 50 of the most predictive variables as inputs, a model that only includes bankruptcy observations one or two years before the filing. All statistical comparisons are made against the model called "All Data." *p<.1 ** p<.05 *** p<.01. Significance levels are based on a two-tailed Z test.

D. Model Appendix

1. Deep Learning Specifications

The deep feed forward network is a normal sequential model with an input layer followed by four hidden dense layers. The first one has 450 nodes and a ReLu activation function; the second and third have 260 nodes, and the final hidden layer has 240 nodes. I then use a SoftMax activation function that outputs two classes and categorical cross entropy as the loss function while optimising with the Adam algorithm.

The convolutional neural network model uses a 1D convolutional layer with max pooling applied, followed by three hidden dense layers with ReLu activation function. The dense layers have 340, 200, and 200 nodes respectively. The output block gets flattened and a sigmoid activation function is applied. I use stochastic gradient descent as the optimiser and binary cross-entropy as the loss function.

2. Imputation

Missing values are a common issue of data validity in finance prediction tasks. This study empirically compares multiple methods of imputation and selects the best method as revealed by the trained model's performance on the validation set. This study compares the performance of imputing zeros, mean, median, and SVD, KNN, and MICE imputation. This paper finally made use of a KNN - Nearest Neighbour - imputation method, which weights all the samples using mean squared difference on the variable for which a user-specified number of date-preceding rows have observable data. The imputed value is simply selected

from the nearest observation with missing values based on the distance between that subject and the target.

3. Parameter Tuning

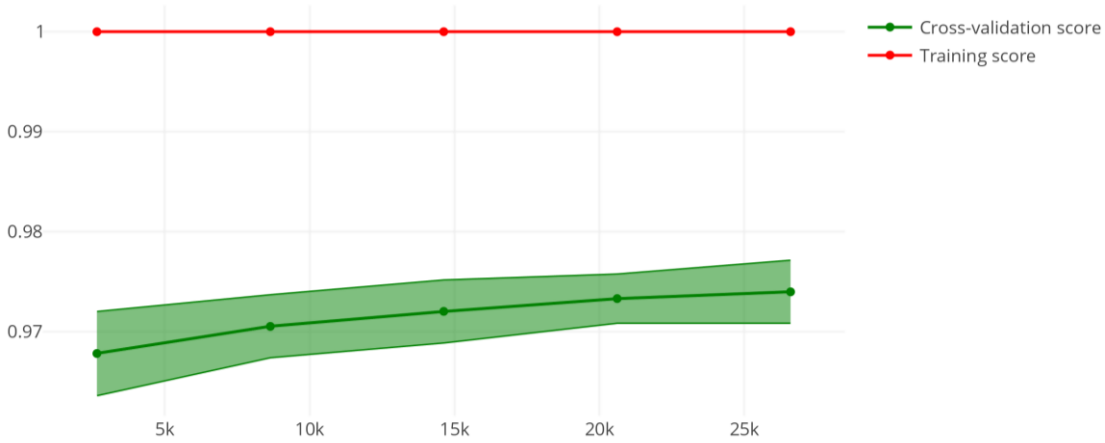
Although the parameter tuning procedures can, to a great extent, be automated, it is still worth understanding the underlying implementation. First, it is important to consider that overfitting models to training data reduces their out-of-sample performance. For that reason, regularisation is an important technique to simplify models so that there is a balance between model fit and predictive performance (Friedman, Hastie, & Tibshirani, 2001). For the majority of models, this simplification (regularisation) is achieved by controlling the number of variables with methods such as a stepwise procedure (Whittingham, Stephens, Bradbury, & Freckleton, 2006) or by creating multiple models and comparing them with information measures such as the Akaike's Information Criterion (Anderson & Burnham, 2002).

Alternatively, shrinkage such as using lasso and ridge methods can be used to add terms and down weight contributions. The concept of shrinkage is similar to what is used in GBMs but is incrementally applied to sequential trees. GBM regularisation jointly involves the optimisation of the number of trees, the learning rate, and tree complexity. *Figure A24* tracks adjustments to these parameters. It shows how an adjustment to the model can be more regularised from left to right. The figure also shows how the parameters can be adjusted to increase the recall of bankruptcy prediction at the expense of precision. There is thus a dimensional trade-off between these parameters. The XGBoost implementation of the GBM has many more parameter inputs than that mentioned above. However, the number of trees, the learning rate, and tree complexity are essential parameters in adjusting the model complexity and reducing overfitting. The approach is then to optimise these parameters by testing many parameter combinations to achieve the minimum prediction error on the validation sets.

The learning curves in *Figure* A23 have an important function in showing researchers whether more data will lead to better cross-validated accuracy. This figure shows that more data will improve the results of this study. This form of analysis is also interesting as it allows one to gauge whether the models overfit the training set. Although a tree ensemble is more likely to overfit than other models, a training score that immediately moves to 100% accuracy could be indicative of overfitting, and by further adjusting the parameters the

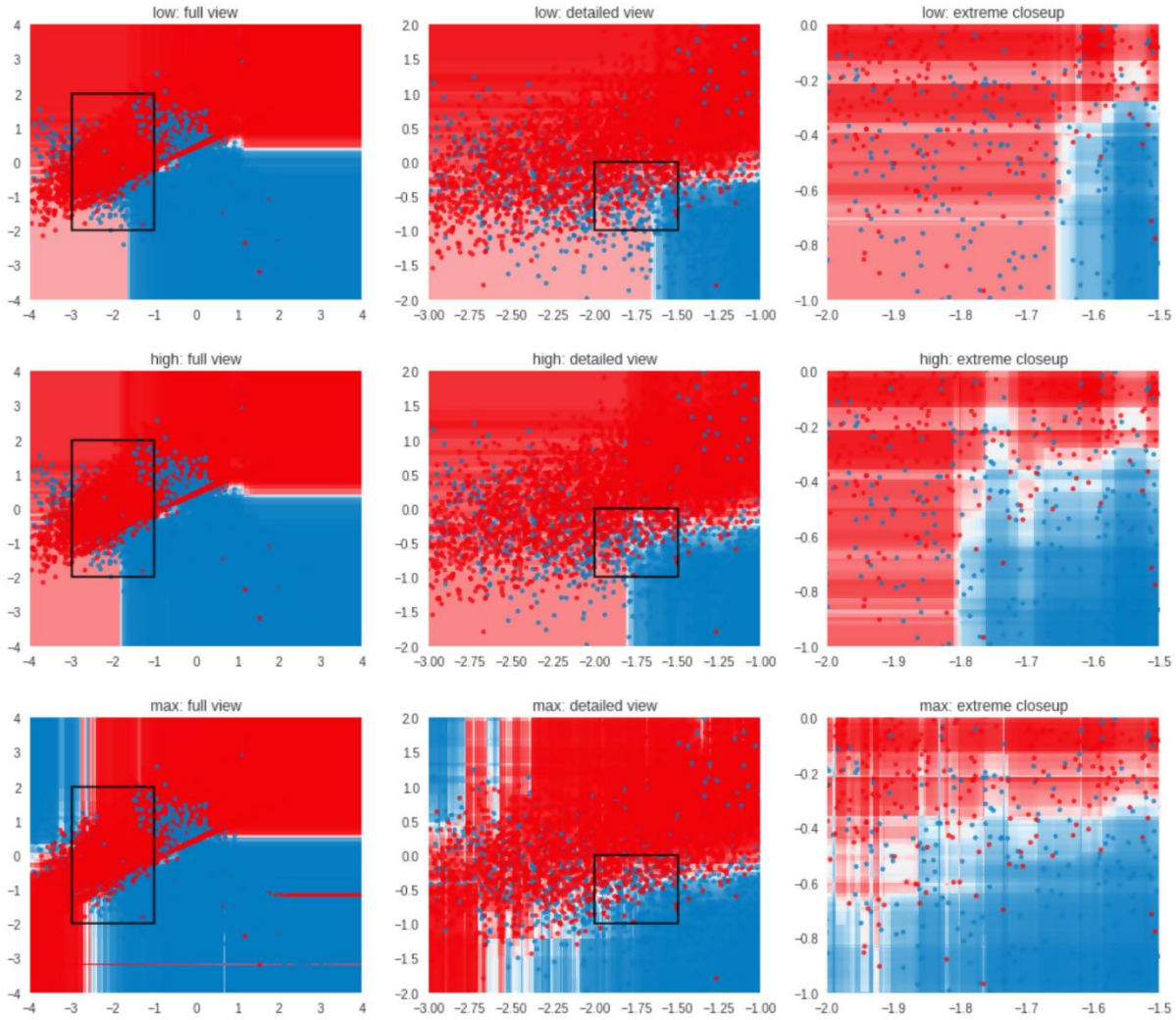
researcher can increase the parameter performance. A more promising curve would be the tangential line starting just above 0.99%.

Figure A23: Learning Curve



This figure illustrates the importance of additional observations in improving the out-of-sample cross-validation (multiple splits to obtain a robust average) accuracy score. The general rule in machine learning is that the more data you have, the better the performance of the classification model. This figure also illustrates the importance of testing on a fresh set of data; simply testing on the training set (the dataset used to infer a function using a machine learning algorithm) biases the results. The above measure shows both the training and cross-validated (out-of-sample) score. The above training score is indicative of overfitting. The model can further be improved with regularisation procedures (discussed in the appendix). After the regularisation, the training score is expected to look more like the curved trajectory presented by the adjacent red line.

Figure A24 Parameter Adjustment Decision Boundaries Between 2 PCA Components



From top to bottom, the parameters of the model get adjusted to increase the recalled bankruptcies. The bankruptcy decision boundary starts with the blued-out area and expands as the model adjusts to include more. Left to right shows the boundary at different resolutions. The boundary in this picture involves the total assets and P/E ratio.

4. Validation

It is valuable to understand the validation technique identified in *Figure 1* before identifying the prediction models. A good illustration of the validation technique on time series data can be seen in *Figure A32*. Once all the input data has been gathered, the sample data has to be split into distinct sets to be able to estimate the generalizable prediction success of both classification models. Following research by Tan, Lee, and Pang (2014), all test splits

in this study are ‘pure hold-out’ sets that are not used by the model at any stage apart from testing the final performance. This dataset remains in a ‘lockbox’ until the testing occurs. This concept is quite important as it allows researchers to get feedback from first testing on the validation data without fearing that they are mistakenly ‘datamining’ the test set.

This study makes use of cross-validated metrics to further improve the robustness of the results (Kohavi, 1995). The cross-validation method simply means that multiple test-train sets are used in evaluating model performance. In this study, I use a unique blocked form of cross-validation that is well-suited for longitudinal evaluation (Bergmeir & Bentez, 2012). Using this approach ensures that the testing data never contains data that is older than the training data. This a sensible step for preserving the integrity of the prediction. As more data becomes available, the training set increases, allowing for an improved prediction. The size of the test set stays constant as the final metric is a simple average over the different splits. Although the test set stays constant in size, it shifts forward to test distinct non-overlapping periods. Each of the training splits can then be fed into the machine learning model to predict a range of target values. This value is compared against the test set’s target values to calculate the prediction success metrics. As mentioned, to calculate the final result, I compute an average value across all the splits and calculate the confidence interval.

5. Classifier Design

Machine Learning is defined as the study of inductive algorithms that ‘learn’(Provost & Kohavi, 1998). For this study, it is valuable to have an intuitive grasp of the XGBoost machine learning model. XGBoost is short for Extreme Gradient Boosting, a nonlinear inductive algorithm used to approximate the function between inputs and outputs. The idea behind Gradient Boosting is to “boost” many weak learners or predictive models to create a stronger overall model. A meta-model gets constructed from a large ensemble of weak models. A weak model simply has to predict slightly better than a random guess. To combine the weak learners, one first trains a weak model, *m*, using data samples drawn from some weight distribution. Then one increases the weight of samples that are misclassified by the model *m* and decreases the weight of those classified correctly, after which one trains the next weak learning using samples drawn according to the updated weight distribution. In this way, the algorithm always uses data samples that were hard to learn in previous rounds to train models. This results in an ensemble that is good at learning a large range of seemingly inscrutable patterns in the training data. In this study, decision trees are used as the weak

learner. After the weighting process, the sum of all the weak learners is taken to produce the overall prediction.

To create the overall ensemble model, such as presented by the *Classifier* pseudocode in the Classifier Design section above, we have to define a loss function, *L,* to minimise. This function has to be differentiable as we want to perform a process of steepest descent, which is an iterative process of attempting to reach the global minimum of a loss function by going down the slope until there is no more room to move closer to the minimum. We, therefore, minimise a loss function numerically via the process of steepest descent. For a classification task, we use logistic regression to obtain the probabilistic outputs of the target variable. The focus here is on $f(\boldsymbol{x}_i)$ as this is the compressed form of the predictor of each tree *i.*

$$L(\theta) = \sum_i [\boldsymbol{y}_i \, ln(1 + e^{f(x_i)}) + (1 - \boldsymbol{y}_i) ln(1 + e^{f(\boldsymbol{x}_i)})] \tag{13}$$

$$L(\theta) = \sum_i [\boldsymbol{y}_i \, ln(1 + e^{-\hat{y}\, i}) + (1 - \boldsymbol{y}_i) ln(1 + e^{\hat{y}\, i})] \tag{14}$$

Further, it is necessary to minimise the loss over all the points in the sample, $(\boldsymbol{x}_i, y_i)$:

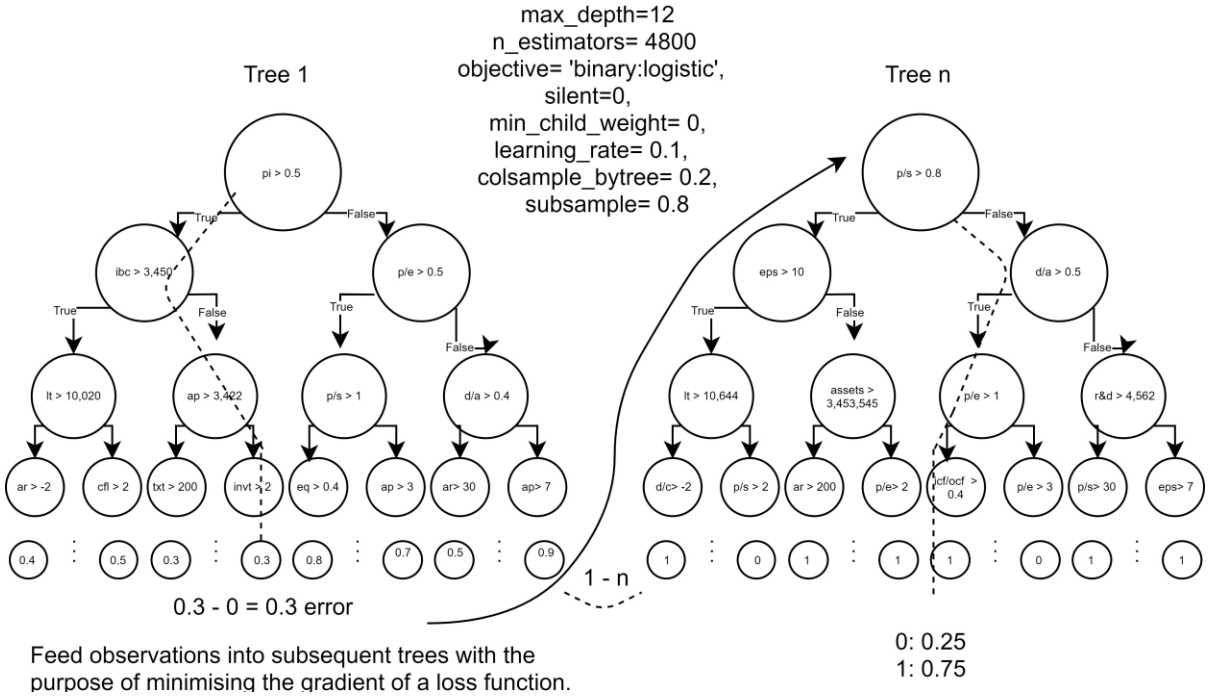
$$f(\boldsymbol{x}) = \sum_{i=1}^{N} L(\theta) \tag{15}$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} L(y_i, f(\boldsymbol{x}_i)) \tag{16}$$

At this point, we are in the position to minimise the predictor function, $f(\boldsymbol{x}_i)$, w.r.t. $\boldsymbol{x}$ since we want a predictor that minimises the total loss of $f(\boldsymbol{x})$. Here we simply apply the iterative process of steepest descent. The minimisation is done in a few phases. These phases are better described in the appendix in chapter 1, page 69, but a short summary follows. The first process starts with adding the first and then successive trees. Adding a tree emulates adding a gradient based correction. Making use of trees ensures that the generation of the gradient expression is successful, as we need the gradient for an unseen test point at each iteration as part of the calculation $f(\boldsymbol{x})$. Finally, this process will return $f(\boldsymbol{x})$ with weighted

parameters. The detailed design of the predictor, $f(\boldsymbol{x})$, is outside the purpose of the this section, but for more extensive computational workings, please see the next section.

Figure A25: XGBoost Decision Tree Ensemble



This illustration provides an example of how a group of decision trees is used to predict a target value. In this example, we follow an observation as it makes its way down two decision trees. In this case, a logistic function transforms the output into a probability for two classes. The XGBoost model used in this study is a little more complex than the above illustration, but the above intuition remains at the core of this model.

As soon as the model is fully trained, testing data can be dropped down the model to identify the predicted response variables. In *Figure A25*: XGBoost Decision Tree Ensemble, I created an illustrative example of how an observation runs through the model and how a prediction is made in the classification task. The response variable is classified as either 0 or 1; a healthy firm-year is designated by a 0 and a bankrupt firm-year with a 1. As a result of the logarithmic loss function, the output is a probability associated with each class for every weak learner. The average of the weak learners establishes the final probability. The regression task follows a similar process; the only difference is that there is only one output per observation and the outputs prediction scores get added together to produce the final prediction.

E. Comparing Traditional, Machine Learning and Deep Learning Models

Hastie et al. (2009) note that one of the most important benefits of Gradient Boosting (GBM) machines is that they require very little research intervention. These models are largely unaffected by missing values, outliers, and monotonic transformation of variables. Apart from being able to easily deal with 'dirty' or 'noisy' data, these models are also much more accurate than the traditional alternatives whose performance even after data cleaning and the pre-processing procedure is substandard at best. Further, these models are not impaired by any heteroscedasticity or multicollinearity issues, which is of serious consequence to parametric models (Probit, LDA & related).

The high dimensionality of GBM models allows them to handle many inputs and to remain largely immune to irrelevant inputs. LDAs and MDA models are low dimensional models that make the mistake of unrealistically assuming linear separability and normality of variables (Chandra, Ravi, & Bose, 2009; Neves & Vieira, 2006). In later years the logit model became the more favoured model (Ohlson, 1980; Pervan, Pervan, & Vukoja, 2011). However, the logit model still has many of the same constraints and disadvantages of the MDA model. Logit and MDA models can only handle a small number of variables as a result of overfitting (Altman, 1968; Ohlson, 1980). For these models, irrelevant variables that enter the global maximum likelihood solution can severely impact the quality of the reduction and model stability. Gradient boosting machines simply classify these inputs as redundant; if the model considers inputs to be irrelevant, then they simply get excluded for the final ensemble.

The GBM does not have as many constraints as other models and can allow for thousands of variables and their interaction effects. Some studies have compared machine learning models such as neural networks with logit and MDA models (Altman, Marco, & Varetto, 1994; Jones et al., 2017). It is difficult to study high dimensional relationships with the traditional models. However, in the past studies that attempted comparison studies, the high dimensional models always come out on top. The GBM model has been identified as one of the strongest models used in prediction research (Hastie et al. 2009). Decision tree ensemble models have consistently been shown to outperform conventional and more sophisticated techniques like support vector machines (SVM) and neural networks (NN). (Hastie et al., 2009; Schapire & Freund, 2012). A multitude of literature outside of finance and accounting has identified this outperformance.

Given that GBM significantly improves the prediction quality of test samples, the use of these models in a practical setting is also important to consider. The first evidence of the

practicality of these models is the ease with which they can be implemented. Structurally GBMs models have minimal architectural requirements; they can easily be developed and executed by popular statistical packages like R and Python. Apart from having the ability to include numerous variables, they can also rank order them based on their predictive power (Friedman, 2001; Hastie et al., 2009). These models are also easily interpretable by using these relative variable importances (RVI) outputs. The use of these models is widespread across many fields such as satellite image recognition, text and speech recognition, biological sciences, credit risk, and cybersecurity.

Another model used in this study is a Convolutional Neural Network (CNN). The main disadvantage of CNN models is that they require large amounts of data; with only thousands of examples in this study, deep learning is unlikely to beat other advanced models. These models also do not have particularly strong theoretical foundations, which means that determining the hyperparameters or topology of deep learning is a black art with no guiding theory. Furthermore, a big drawback is that what is learned cannot be as easily interpreted, as is the case for decision tree models. In saying that, deep learning, via a process known as feature learning, removes the need for manual feature engineering. Lastly, as evidenced in this study, the architecture can easily be adopted for new problem sets.

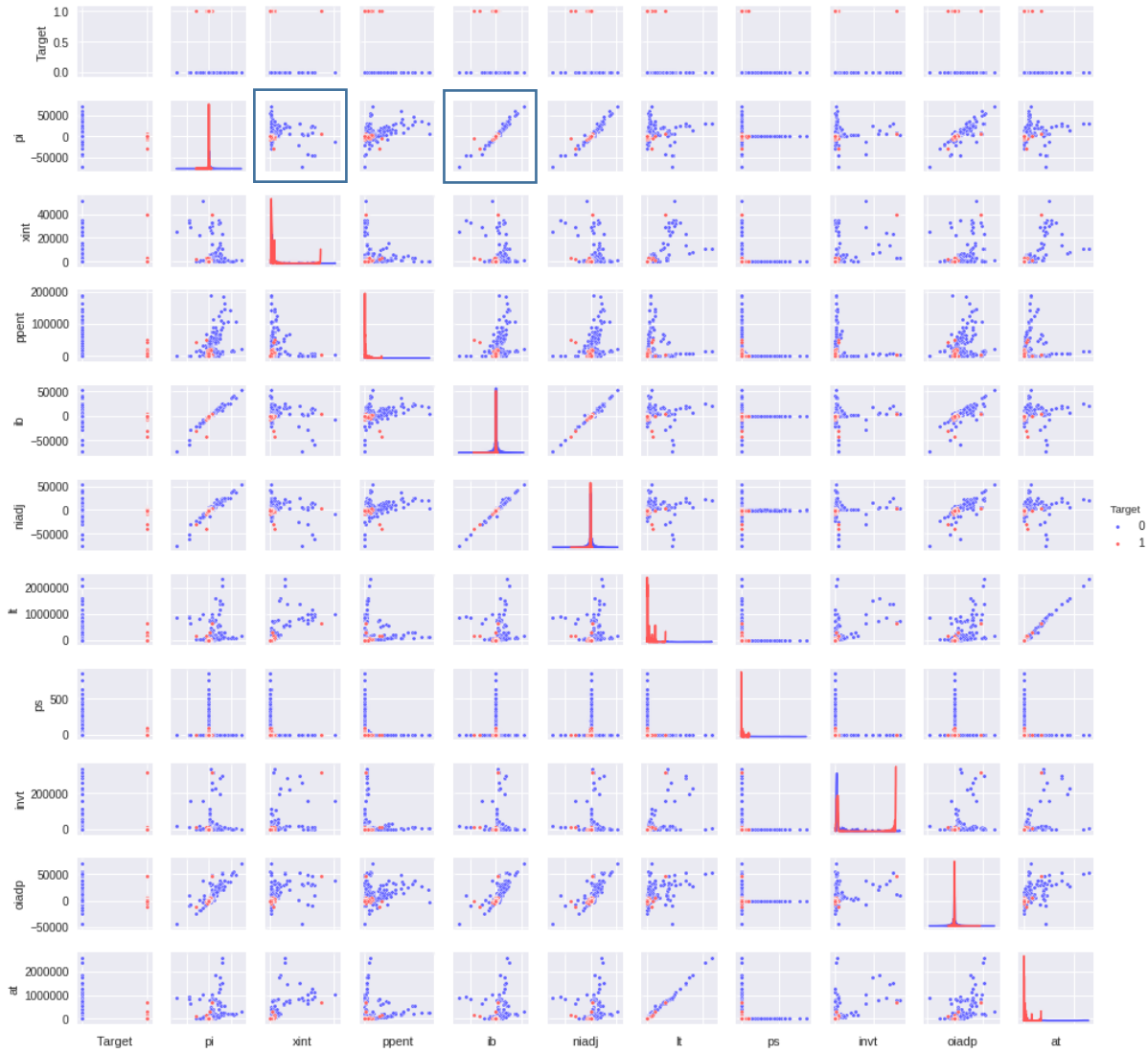
Unlike black-box models, like the CNN model, GBMs are transparent in their methods of inductive learning from data. Further GBMs can be fitted to a small amount of data, whereas the CNN network needs a larger amount of data. Boosting is an interesting feature to the model. It grows the number of trees by sequentially modeling the residuals to include atypical observations that depart from the dominant patterns of the initial trees. In doing this, the algorithm simultaneously reduces the bias and variance of the model. GBMs can also successfully handle different response variables, continuous, discrete, count etc.

Compared with conventional models, there are no p-values to indicate the relative significance of model coefficients; it is also difficult to determine the degrees of freedom in the model. It is questionable as to whether these aspects are a problem or an advantage to the model, as most would be aware there are rigorous debates as to the use of p-value in models (Fidler, Geoff, Mark, & Neil, 2004). Although the GBMs lack simple metrics that can be a disadvantage from a traditional point of view, a large amount of methods of interpretation has developed over the years, with many ongoing developments. These techniques provide equivalent functions to many of the conventional techniques.

F. Extended Analyses

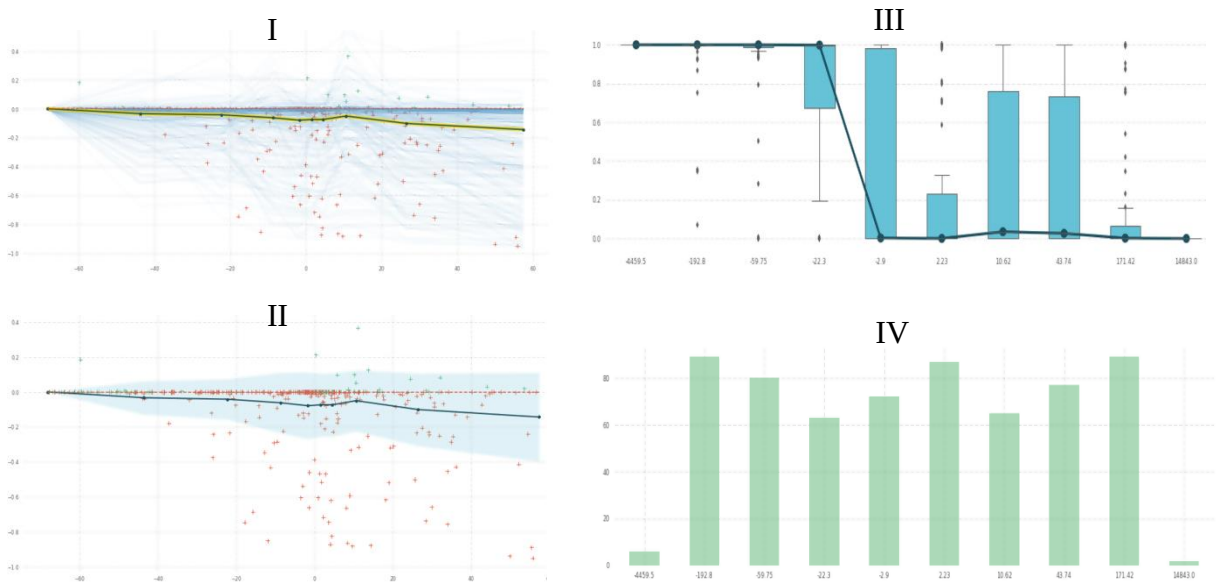
Figure A26: Pair Plots below presents the scatter plots of the most important variables. In this figure, bankruptcies are coded in red and healthy firms in blue. The relationship between pre-tax income (pi) and interest and related expenses (xint) poses as an interesting interaction that would be shown to be one of the most important pairs to predict bankruptcy (See column three, row three in *Table 22*). This pair's predictive power is evident by the dense clustering of bankruptcies around a fixed point, making it easy for a decision tree prediction model to discriminate between bankrupt and non-bankrupt firms. The XGBoost model would seek to sculpt a decision boundary around that point to predict future bankruptcies. This figure illustrates many of these important relationships. The reader should note that this simply shows relationships between pairs in the data, and it does not show the relationship of these variables in a fully trained model. This will be dealt with from *Table 28* onwards. The advantage of the non-linear models used in this study is that they do not just look at the low dimensional relationships of the scatterplots, but they also investigate relationships at extremely high dimensions. In this study, I descriptively report interactions to the depth of three, i.e., up to three variables' non-linear interactions and the predicted response.

Figure A26: Pair Plots



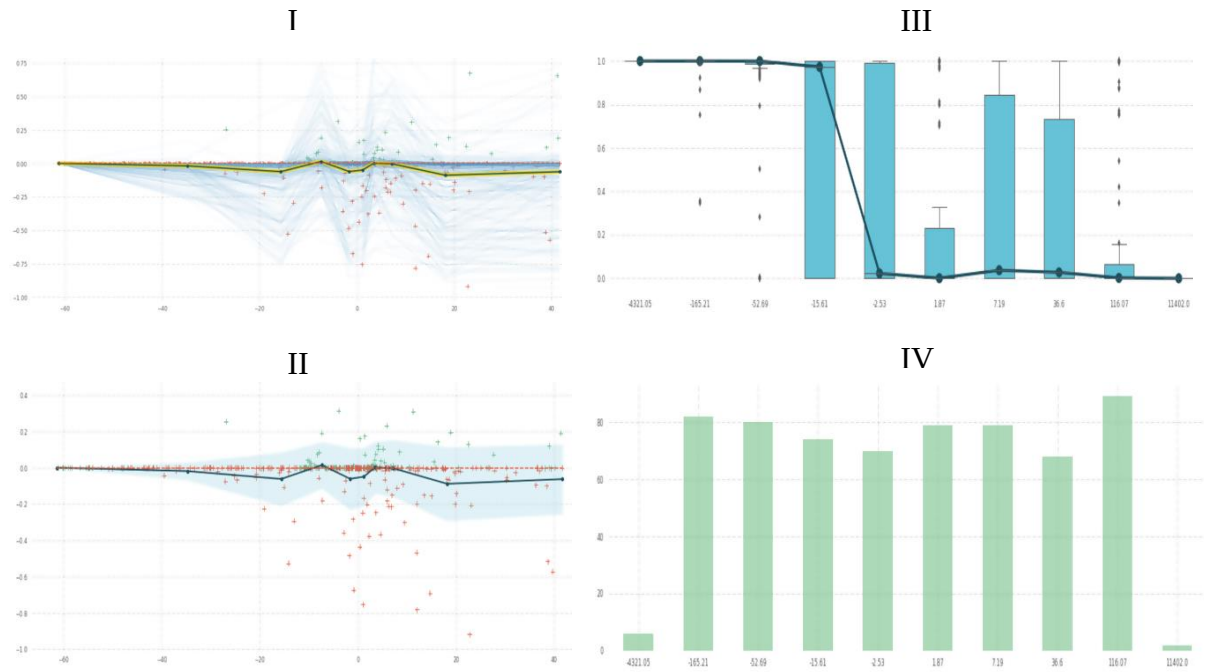
This figure reports the scatterplots of distinct variable relationships that show prominence in their predictive ability. The intersection of the variable with itself plots for the variable's distribution. The red dots are observations labelled as bankruptcies and the blue dots are observations labelled as healthy firms.

Figure A27: Pre-tax Income (PI) Variable Analysis



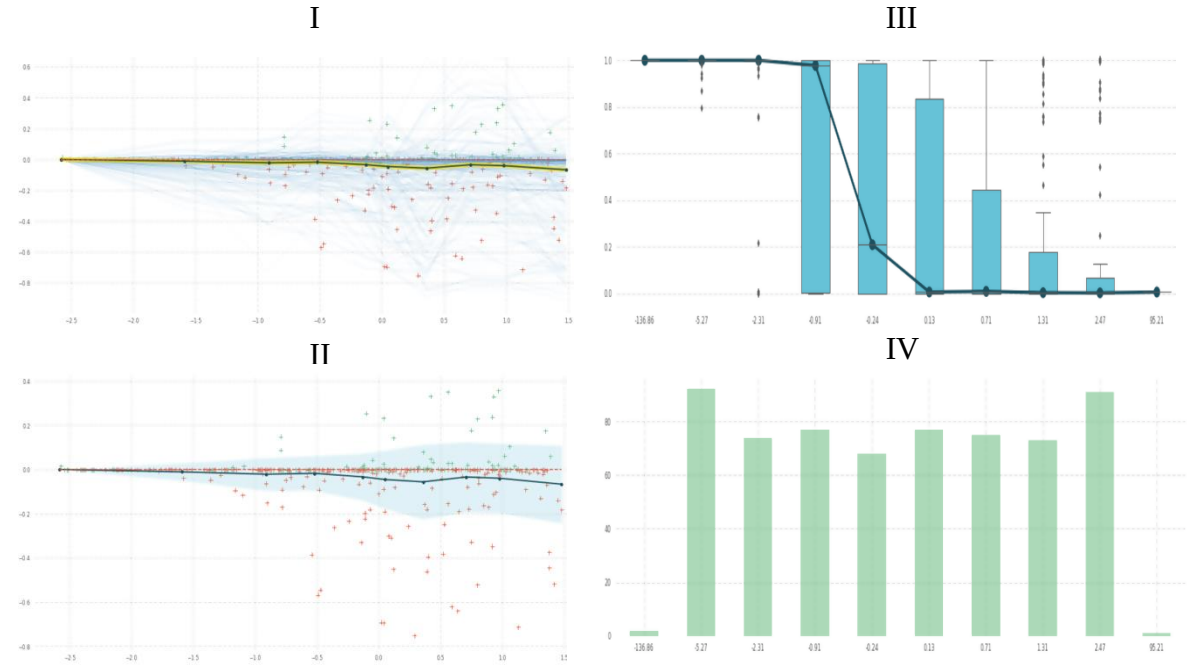
This figure reports the marginal relationship pre-tax income has with bankruptcy prediction. Plot I, II, and III separately show that when the pre-tax-income increases, the likelihood of bankruptcy decreases. Plot I is a simple partial dependence plot that draws lines of all the observational trajectories. Plot II establishes a confidence band and better highlights bankrupt predictions (green dots) and healthy firm predictions (red dots). Plot III is a box plot of an equally balanced bankruptcy and healthy prediction model. Plot IV is a count plot of Plot III. The observations in I and II are Winsorized to improve the look of the plots. Plot III shows that there is a point when the distribution between bankruptcy and healthy firms enlarge at around 10-40 million in income; this is corroborated by a spike in plot I of increased bankruptcy predictions. III shows that negative PI is a potential indicator of future bankruptcy.

Figure A28: Income Before Extraordinary Items (IBC) Variable Analysis



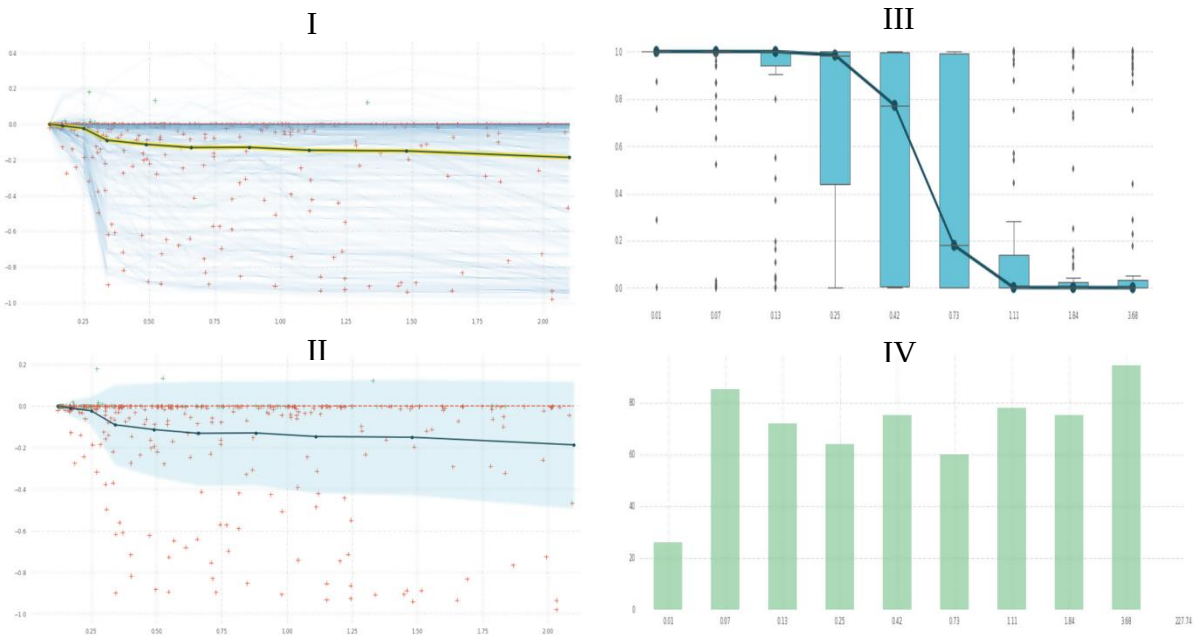
This figure reports the marginal relationship income before extraordinary items (IBC) has with bankruptcy prediction. Plot I, II, and III separately show that when IBC increases the likelihood of bankruptcy decreases. The plot I is a simple partial dependence plot that draws lines of all the observational trajectories. Plot II establishes a confidence band and better highlights bankrupt predictions (green dots) and healthy firm predictions (red dots). Plot III is a box plot of an equally balanced bankruptcy and healthy prediction model. Plot IV is a count plot of Plot III. The observations in I and II are Winsorized to improve the look of the plots. Plot III shows that there is a point when the distribution between bankruptcy and healthy firms enlarges at around 7-35 million in income; this is corroborated by a spike in plot I of increased bankruptcy predictions. III shows that situations of a negative IBC are an indicator of future bankruptcy. This measure seems to be somewhat more volatile than the PI measure. It further highlights in I and II that firms who have small negative IBCs around -10 to -7 are less likely to be bankrupt than those more negative than -10 and more positive than -7 up and till 0.

Figure A29: EPS (Basic) - Exclude Extra. Items (EPSPX) Variable Analysis



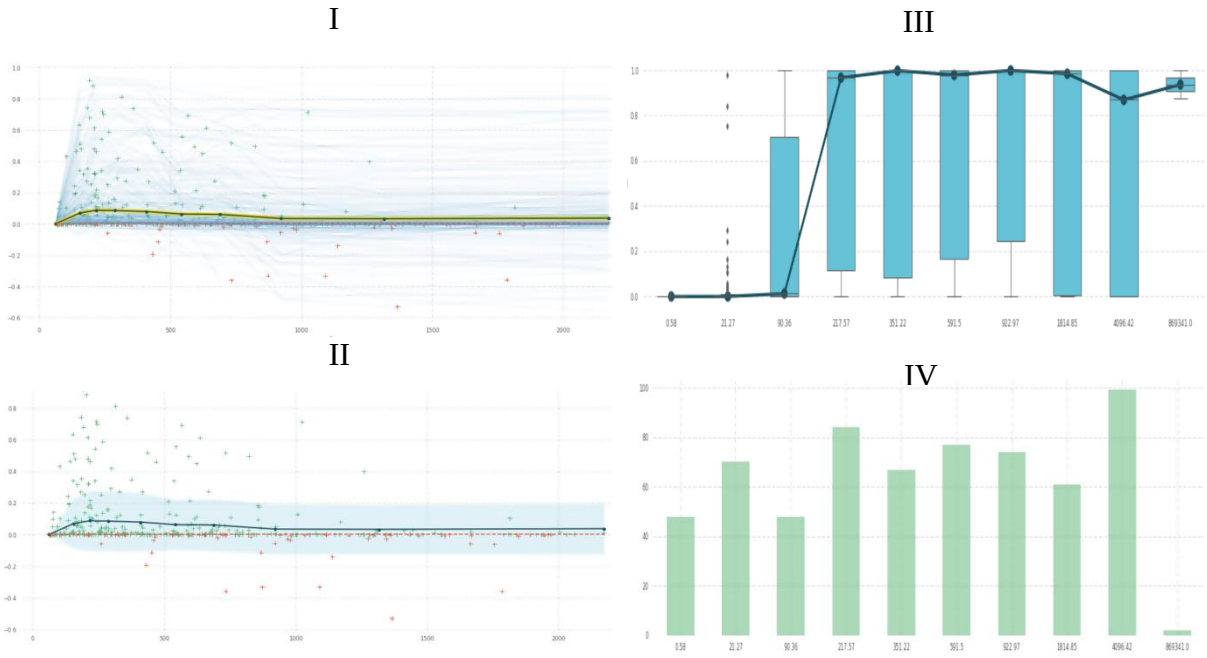
This figure reports the marginal relationship EPS has with bankruptcy prediction. Plot I, II, and III separately show that when EPS increases the likelihood of bankruptcy decreases. The plot I is a simple partial dependence plot that draws lines of all the observational trajectories. Plot II establishes a confidence band and better highlights bankruptcy predictions (green dots) and healthy firm predictions (red dots). Plot III is a box plot of an equally balanced bankruptcy and healthy prediction model. Plot IV is a count plot of Plot III. The observations in I and II are Winsorized to improve the look of the plots. Plot III shows that the distribution gradually centres around a healthy firm prediction as the EPS increases. A negative EPS value seems to be a good indicator of future failure. Although removed from I and II, III shows that situations of increased negative PI are a potential indicator of future bankruptcy.

Figure A30: Price/Sales (PS) Variable Analysis



This figure reports the marginal relationship Price/Sales has with bankruptcy prediction. Plot I, II, and III separately show that when the PS increases the likelihood of bankruptcy decreases. The plot I is a simple partial dependence plot that draws lines of all the observational trajectories. Plot II establishes a confidence band and better highlights bankruptcy predictions (green dots) and healthy firm predictions (red dots). Plot III is a box plot of an equally balanced bankruptcy and healthy prediction model. Plot IV is a count plot of Plot III. The observations in I and II are Winsorized to improve the look of the plots. III shows that situations of increased PI are highly indicative of a healthy firm. A high price to sales firm normally enjoys high-profit margins and are most notably at the top of their respective industries at that point of time, and for that reason, they are unlikely to become bankrupt.

Figure A31: Liabilities - Total (LT) Variable Analysis



This figure reports the marginal relationship liabilities has with bankruptcy prediction. Plot I, II, and III separately show that when LT increases the likelihood of bankruptcy increases. The plot I is a simple partial dependence plot that draws lines of all the observational trajectories. Plot II establishes a confidence band and better highlights bankruptcy predictions (green dots) and healthy firm predictions (red dots). Plot III is a box plot of an equally balanced bankruptcy and healthy prediction model. Plot IV is a count plot of plot III. The observations in I and II are Winsorized to improve the look of the plots. III shows that when a firm has no liabilities, it is extremely unlikely for them to become bankrupt.

Table A40: Neural Network Models Bankruptcy Literature

Reference	Journal	Description	Model	AUC
Kim and Kang (2010)	Expert Systems with Applications 37 (2010) 3373–3379	1458 manufacturing firms (2002–2005), half of which went bankrupt (1:1)	Boosted Neural Network	0.750
du Jardin (2017)	Expert Systems with Applications (75) Pages 25-43.	95,910 French Firms (1996 - 2009) 1,920 failing firms (1:0.020)	Feed Forward Neural Network	0.800
Mselmi et al. (2017)	International Review of Financial Analysis (2017) 50: 67-80	212 French firms, half of which is distressed. (1:1)	ANN (MLP)	0.871
Barboza et al. (2017)	Expert Systems with Applications 83 (2017), Pages 405-417	More than 10,000 firm-year observations. 1,796 failed firms (1:0.22)	ANN (MLP)	0.901
Zhou (2013)	Knowledge-Based Systems (2013) 41: 16-25	86,129 US firm year, 918 (1981-2009) bankruptcies (1:0.011)	ANN (MLP)	0.856
Huang et al. (2016)	Kybernetes (45) 2016	270 Taiwanese companies (2004-2014), 90 failed firms (1:0.5)	GRNN model with FOA optimisation	0.903
Jones, Johnstone, and Wilson (2017)	Journal of Business Finance and Accounting (2017) 44: 3–34	30,129 US firm years, 3960 firm-year bankruptcies. (1:0.15)	ANN (MLP)	0.853
This Study		33,242 US large firm years, 1224 firm-year bankruptcies 1977-2016 (1:0.038)	Deep Convolutional Neural Network	0.914

This table reports the results of past neural network research that reported an ROC (AUC) metric to be used for cross-study comparisons.

Table A41: Boosting and Decision Tree Model Literature

Reference	Journal	Description	Model	AUC
Chandra et al. (2009)	Expert Systems with Applications 36 (2009) 4830–4837, C	240 dot-com companies, half of which went bankrupt (1:1)	Boosting	0.900
Olson et al. (2012)	Decision Support Systems 52 (2012) 464–473, A*	1321 US firm years sampled over the period 2005-2011, 100 firms went bankrupt (1:0.082)	Decision Trees	0.947
Kim and Upneja (2014)	Economic Modelling 36 (2014) 354–362., A	142 Restaurant Firms 1988-2010 (1:1)	AdaBoost	0.988
Karas and Reznakova (2017)	Engineering Economics (2017) 28(2): 145-154, B	1540 Construction Firms, 283 went bankrupt (1:0.23)	CART	0.859
Barboza et al. (2017)	Expert Systems with Applications 83 (2017), Pages 405-417, C	More than 10,000 firm-year observations. 1,796 failed firms (1:0.22)	Boosting	0.901
Zieba et al. (2016)	Expert Systems with Applications 58 (2016) 93-101, C	10,174 emerging market firm years, 400 failed 0.041 (2000-2012)	Ensemble XGBoost	0.944
Jones (2017)	Review Accounting Studies (2017) 22:1366–1422, A*	36,209 US firm years, 4460 firm year bankruptcies 1987 to 2013 (1:0.14)	Proprietary Gradient Boosting Machine - TreeNet	0.997
Volkov et al. (2017)	Decision Support Systems 98 (2017) 59–68, A*	19,380 Belgium and Luxembourg Firms, 1,933 bankrupt firms, 2007-2015 (0.11)	Random Forest	0.859
Jones, Johnstone, and Wilson (2017)	Journal of Business Finance and Accounting (2017) 44: 3–34, A	30,129 US firm years, 3960 firm-year bankruptcies. (1:0.15)	Boosting	0.931
This Study		33,242 US large firm years, 1224 firm year bankruptcies 1977-2016 (1:0.038)	Freeware Gradient Boosting Machine - XGBoost	0.957

This table reports the results of past boosting and decision tree ensemble research that reported an ROC (AUC) metric to be used for cross-study comparisons.

Table A42: Literature on Variable and Category Importance for Decision Tree Ensembles

Reference	Journal	Description	Selector	Category Importance
Kim and Upneja (2014)	Economic Modelling 36 (2014) 354–362., A	142 Restaurant Firms 1988-2010 (1:1)	Splitter's Level	(1) Solvency, (2) Liquidity, (3) Profitability
Beher and Weinblat (2016)	International Journal of the Economics of Business (2017), 24:2, 181-222, B	Default patterns in European Firms 1,964,374 firm observations from 2010-2011.	Random Forest Variable Importance	(1) Solvency, (2) Profitability, (3) Liquidity
Jones (2017)	Accounting Studies Review (2017) 22:1366–1422, A*	36,209 US firm years, 4460 firm year bankruptcies 1987 to 2013 (1:0.14)	Gain Measure	(1) Governance, (2) Valuation
Volkov et al. (2017)	Decision Support Systems 98 (2017) 59–68, C	19,380 Belgium and Luxembourg Firms, 1,933 bankrupt firms, 2007-2015 (0.11)	Random Forest Variable Importance	(1) Solvency, (2) Profitability, (3) Liquidity
Jones, Johnstone, and Wilson (2017)	Journal of Business Finance and Accounting (2017) 44: 3–34, A	30,129 US firm years, 3960 firm-year bankruptcies. (1:0.15)	Random Forest Variable Importance	(1) Solvency, (2) Profitability, (3) Efficiency, (4) Liquidity
Mselmi et al. (2017)	International Review of Financial Analysis (2017) 50: 67-80, A	212 French firms, half of which is distressed. (1:1)	Stepwise Regression	(1) Solvency, (2) Efficiency, (3) Liquidity, (2) Profitability
This Study		33,242 US large firm years, 1224 firm-year bankruptcies 1977-2016 (1:0.038)	Gain Measure	(1) Solvency, (2) Profitability and Valuation, (3) Efficiency (4) Liquidity

This table reports all past studies that ranked the importance of their variables. I matched the respective variables in each study to a category in the attempt to identify which categories these studies deemed to be more important.

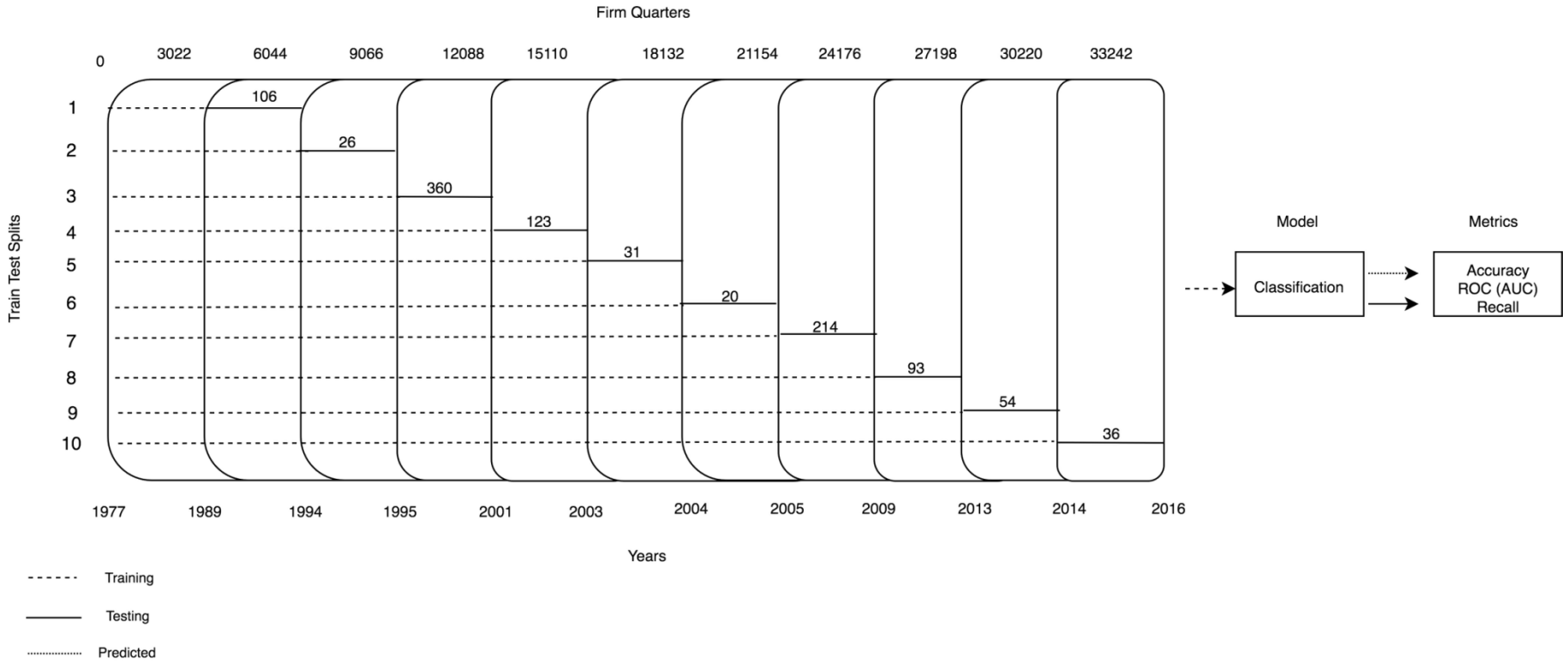
Table A43: Bankruptcy and Healthy Firm Summary Statistics for Important Variables

Table A44: Filing Outcome Summary Statistics

Target	Response	che	npm	lct	at_turn	rect_turn	pay_turn	curr_debt	dpc	accrual	mib
Duration	< 1 Year	163.70	-0.40	510.45	0.89	15.41	12.00	0.42	112.54	0.12	6.36
	> 1 Year	873.82	-1.63	840.72	1.00	17.54	9.31	0.42	120.00	0.06	36.82
Survival	No	1084.19	-0.33	602.56	0.90	15.48	8.24	0.48	71.75	0.07	22.87
	Yes	202.86	-1.53	741.43	0.98	17.23	12.03	0.38	144.90	0.10	22.59
Chapter	11	554.89	-1.08	704.21	0.97	16.95	10.82	0.42	119.17	0.09	23.30
	7	155.81	-0.29	61.10	0.19	1.65	0.75	0.60	17.48	0.05	0.00
Asset Sale	No	262.76	-1.26	718.73	0.93	16.34	11.49	0.40	108.24	0.09	18.88
	Yes	1433.50	-0.43	589.36	1.02	17.22	7.63	0.48	142.71	0.07	34.73
Tort	No	567.78	-1.13	643.65	0.96	17.19	10.59	0.42	113.83	0.09	15.13
	Yes	208.62	-0.11	1323.59	0.84	7.36	10.09	0.41	155.80	0.04	132.19

The following table presents the two most important variables for each prediction task. Instead of reporting the summary statistics for just those two, the summary statistics of the two important variables for all filing outcomes are reported. The full names of the variables in order are: Cash and Short-Term Investments (che), Net Profit Margin (npm), Inventories - Total (invt), Current Liabilities - Total (lct), Asset Turnover (at_turn), Receivables Turnover (rect_turn), Payables Turnover (pay_turn) Current Liabilities/Total Liabilities (curr_debt), Depreciation and Amortization (dpc), Accruals/Average Assets (accrual), Minority Interest - Balance Sheet (mib).

Figure A32: Illustration of Validation in Time Series



Validation is used as an improved means to forecast the accuracy of an inducer by splitting the data into *n* mutually exclusive subsets. To ensure consistent performance measurements on these splits, they should be approximately the same size. In this study, the data splits into four equal-sized sections. And the model is trained and tested on each of these splits. Each time, the model trains on an increasing number of samples ordered by date. This study reports both the overall validated accuracy and breaks the accuracy down to each period and surprise threshold in question. This table does not show a separate process used to do variable selection (the process of removing variables which seem irrelevant for modeling) before the validation. The variable selection is done on a small validation set constituting 15% of the data to ensure that during the development stage there is no "double dipping" into the data. Therefore the model always gets tested on a fresh out-of-sample dataset. Another approach would be to create multiple validation sets and hyperparameter selections for each period.

Table A45: Robustness Table of Validation in Time Series. – Metrics

Train Period	Test Period	Cross-entropy	Brier Score Loss	Accuracy Score	ROC AUC Sore	Average Precision Score	Precision - Bankrupt Firms	Precision - Healthy Firms	False Positive Rate	False Negative Rate	False Discovery Rate
1977 - 1989	1989 - 1994	0.098	0.026	0.967	0.944	0.465	0.563	0.973	0.007	0.745	0.438
1977 - 1994	1994 - 1995	0.023	0.006	0.993	0.983	0.529	0.800	0.994	0.001	0.692	0.200
1977 - 1995	1995 - 2001	0.425	0.089	0.895	0.920	0.672	0.831	0.897	0.004	0.850	0.169
1977 - 2001	2001 - 2003	0.095	0.024	0.971	0.958	0.621	0.688	0.980	0.010	0.480	0.312
1977 - 2003	2003 - 2004	0.042	0.009	0.989	0.917	0.264	0.375	0.991	0.002	0.903	0.625
1977 - 2004	2004 - 2005	0.025	0.005	0.995	0.960	0.465	0.778	0.996	0.001	0.650	0.222
1977 - 2005	2005 - 2009	0.255	0.054	0.937	0.929	0.598	0.780	0.939	0.003	0.850	0.220
1977 - 2009	2009 - 2013	0.085	0.021	0.974	0.953	0.525	0.651	0.978	0.005	0.699	0.349
1977 - 2013	2013 - 2014	0.049	0.011	0.986	0.962	0.566	0.842	0.987	0.001	0.704	0.158
1977 - 2014	2014 - 2016	0.031	0.009	0.988	0.984	0.561	0.500	0.994	0.006	0.472	0.500
1977 - 2014	1989 - 2015	0.105	0.024	0.971	0.957	0.562	0.693	0.975	0.004	0.681	0.307

This table reports an extensive list of model performance metrics over various sample splits. From this table, it is clear that there are large differences between the different periods. The best reported AUC of 0.984 occurred over the last few years of the sample 2014-2016. The worst AUC occurred over the 2003-2004 period. It is worth noting that Congress made amendments to the bankruptcy code in 1994; this could affect the bankruptcy prediction quality from 1995 onwards, at least until new observations are learned (Tabb, 1995). This study shows that the underlying distribution of bankrupt and healthy firms has an impact on the model performance score.

Table A46: Robustness of Validation in Time Series. – Observations

Train Period	Test Period	All Instances	Bankruptcy Sample	Bankrupt Recalled	True Positives	False Positives	Healthy Sample	Healthy Recalled	True Negatives	False Negatives	Bankrupt to Healthy
1977 - 1989	1989 - 1994	3022	106	48	27	21	2916	2974	2895	79	0.04
1977 - 1994	1994 - 1995	3022	26	10	8	2	2996	3012	2994	18	0.01
1977 - 1995	1995 - 2001	3022	360	65	54	11	2662	2957	2651	306	0.14
1977 - 2001	2001 - 2003	3022	123	93	64	29	2899	2929	2870	59	0.04
1977 - 2003	2003 - 2004	3022	31	8	3	5	2991	3014	2986	28	0.01
1977 - 2004	2004 - 2005	3022	20	9	7	2	3002	3013	3000	13	0.01
1977 - 2005	2005 - 2009	3022	214	41	32	9	2808	2981	2799	182	0.08
1977 - 2009	2009 - 2013	3022	93	43	28	15	2929	2979	2914	65	0.03
1977 - 2013	2013 - 2014	3022	54	19	16	3	2968	3003	2965	38	0.02
1977 - 2014	2014 - 2016	3022	36	38	19	19	2986	2984	2967	17	0.01
1977 - 2014	1989 - 2016	30220	1063	374	258	116	29157	29846	29041	805	0.04

This table reports an extensive list of model prediction metrics over various sample splits. From this table, it is clear that there are large differences in the distribution of bankruptcies over the different periods. The least amount of bankruptcies for a validation-period occurred over the last few years of the sample 2014-2016. The most bankruptcies occurred over the period 1995-2001.

Table A47: Table of Financial Ratios and Categorisation

Financial Ratio	Variable Name	Category	Formula
Capitalization Ratio	capital_ratio	Capitalization	Total Long-term Debt as a fraction of the sum of Total Long-term Debt, Common/Ordinary Equity and Preferred Stock
Common Equity/Invested Capital	equity_invcap	Capitalization	Common Equity as a fraction of Invested Capital
Long-term Debt/Invested Capital	debt_invcap	Capitalization	Long-term Debt as a fraction of Invested Capital
Total Debt/Invested Capital	totdebt_invcap	Capitalization	Total Debt (Long-term and Current) as a fraction of Invested Capital
Asset Turnover	at_turn	Efficiency	Sales as a fraction of the average Total Assets based on the most recent two periods
Inventory Turnover	inv_turn	Efficiency	COGS as a fraction of the average Inventories based on the most recent two periods
Payables Turnover	pay_turn	Efficiency	COGS and change in Inventories as a fraction of the average of Accounts Payable based on the most recent two periods
Receivables Turnover	rect_turn	Efficiency	Sales as a fraction of the average of Accounts Receivables based on the most recent two periods
Sales/Stockholders Equity	sale_equity	Efficiency	Sales per dollar of total Stockholders' Equity
Sales/Invested Capital	sale_invcap	Efficiency	Sales per dollar of Invested Capital
Sales/Working Capital	sale_nwc	Efficiency	Sales per dollar of Working Capital, defined as the difference between Current Assets and Current Liabilities
Inventory/Current Assets	invt_act	Financial Soundness	Inventories as a fraction of Current Assets
Receivables/Current Assets	rect_act	Financial Soundness	Accounts Receivables as a fraction of Current Assets
Free Cash Flow/Operating Cash Flow	fcf_ocf	Financial Soundness	Free Cash Flow as a fraction of Operating Cash Flow, where Free Cash Flow is defined as the difference between Operating Cash Flow and Capital Expenditures
Operating CF/Current Liabilities	ocf_lct	Financial Soundness	Operating Cash Flow as a fraction of Current Liabilities
Cash Flow/Total Debt	cash_debt	Financial Soundness	Operating Cash Flow as a fraction of Total Debt
Cash Balance/Total Liabilities	cash_lt	Financial Soundness	Cash Balance as a fraction of Total Liabilities
Cash Flow Margin	cfm	Financial Soundness	Income before Extraordinary Items and Depreciation as a fraction of Sales
Short-Term Debt/Total Debt	short_debt	Financial Soundness	Short-term Debt as a fraction of Total Debt

Financial Ratio	Variable Name	Category	Formula
Profit Before Depreciation/Current Liabilities	profit_lct	Financial Soundness	Operating Income before D&A as a fraction of Current Liabilities
Current Liabilities/Total Liabilities	curr_debt	Financial Soundness	Current Liabilities as a fraction of Total Liabilities
Total Debt/EBITDA	debt_ebitda	Financial Soundness	Gross Debt as a fraction of EBITDA
Long-term Debt/Book Equity	dltt_be	Financial Soundness	Long-term Debt to Book Equity
Interest/Average Long-term Debt	int_debt	Financial Soundness	Interest as a fraction of average Long-term debt based on most recent two periods
Interest/Average Total Debt	int_totdebt	Financial Soundness	Interest as a fraction of average Total Debt based on most recent two periods
Long-term Debt/Total Liabilities	lt_debt	Financial Soundness	Long-term Debt as a fraction of Total Liabilities
Total Liabilities/Total Tangible Assets	lt_ppent	Financial Soundness	Total Liabilities to Total Tangible Assets
Cash Conversion Cycle (Days)	cash_conversion	Liquidity	Inventories per daily COGS plus Account Receivables per daily Sales minus Account Payables per daily COGS
Cash Ratio	cash_ratio	Liquidity	Cash and Short-term Investments as a fraction of Current Liabilities
Current Ratio	curr_ratio	Liquidity	Current Assets as a fraction of Current Liabilities
Quick Ratio (Acid Test)	quick_ratio	Liquidity	Quick Ratio: Current Assets net of Inventories as a fraction of Current Liabilities
Accruals/Average Assets	Accrual	Other	Accruals as a fraction of average Total Assets based on most recent two periods
Research and Development/Sales	RD_SALE	Other	R&D expenses as a fraction of Sales
Avertising Expenses/Sales	adv_sale	Other	Advertising Expenses as a fraction of Sales
Labor Expenses/Sales	staff_sale	Other	Labor Expenses as a fraction of Sales
Effective Tax Rate	efftax	Profitability	Income Tax as a fraction of Pretax Income
Gross Profit/Total Assets	GProf	Profitability	Gross Profitability as a fraction of Total Assets
After-tax Return on Average Common Equity	aftret_eq	Profitability	Net Income as a fraction of average of Common Equity based on most recent two periods
After-tax Return on Total Stockholders' Equity	aftret_equity	Profitability	Net Income as a fraction of average of Total Shareholders' Equity based on most recent two periods

Financial Ratio	Variable Name	Category	Formula
After-tax Return on Invested Capital	aftret_invcapx	Profitability	Net Income plus Interest Expenses as a fraction of Invested Capital
Gross Profit Margin	gpm	Profitability	Gross Profit as a fraction of Sales
Net Profit Margin	npm	Profitability	Net Income as a fraction of Sales
Operating Profit Margin After Depreciation	opmad	Profitability	Operating Income After Depreciation as a fraction of Sales
Operating Profit Margin Before Depreciation	opmbd	Profitability	Operating Income Before Depreciation as a fraction of Sales
Pre-tax Return on Total Earning Assets	pretret_earnat	Profitability	Operating Income After Depreciation as a fraction of average Total Earnings Assets (TEA) based on most recent two periods, where TEA is defined as the sum of Property Plant and Equipment, and Current Assets
Pre-tax return on Net Operating Assets	pretret_noa	Profitability	Operating Income After Depreciation as a fraction of average Net Operating Assets (NOA) based on most recent two periods, where NOA is defined as the sum of Property Plant and Equipment, and Current Assets minus Current Liabilities
Pre-tax Profit Margin	ptpm	Profitability	Pretax Income as a fraction of Sales
Return on Assets	roa	Profitability	Operating Income Before Depreciation as a fraction of average Total Assets based on most recent two periods
Return on Capital Employed	roce	Profitability	Earnings Before Interest and Taxes as a fraction of average Capital Employed based on most recent two periods, where Capital Employed is the sum of Debt in Long-term and Current Liabilities and Common/Ordinary Equity
Return on Equity	roe	Profitability	Net Income as a fraction of average Book Equity based on most recent two periods, where Book Equity is defined as the sum of Total Parent Stockholders' Equity and Deferred Taxes and Investment Tax Credit
Total Debt/Equity	de_ratio	Solvency	Total Liabilities to Shareholders’ Equity (common and preferred)
Total Debt/Total Assets	debt_assets	Solvency	Total Debt as a fraction of Total Assets
Total Debt/Total Assets	debt_at	Solvency	Total Liabilities as a fraction of Total Assets
Total Debt/Capital	debt_capital	Solvency	Total Debt as a fraction of Total Capital, where Total Debt is defined as the sum of Accounts Payable and Total Debt in Current and Long-

Financial Ratio	Variable Name	Category	Formula
			term Liabilities, and Total Capital is defined as the sum of Total Debt and Total Equity (common and preferred)
After-tax Interest Coverage	intcov	Solvency	Multiple of After-tax Income to Interest and Related Expenses
Interest Coverage Ratio	intcov_ratio	Solvency	Multiple of Earnings Before Interest and Taxes to Interest and Related Expenses
Dividend Payout Ratio	dpr	Valuation	Dividends as a fraction of Income Before Extra. Items
Forward P/E to 1-year Growth (PEG) ratio	PEG_1yrforward	Valuation	Price-to-Earnings, excl. Extraordinary Items (diluted) to 1-Year EPS Growth rate
Forward P/E to Long-term Growth (PEG) ratio	PEG_ltgforward	Valuation	Price-to-Earnings, excl. Extraordinary Items (diluted) to Long-term EPS Growth rate
Trailing P/E to Growth (PEG) ratio	PEG_trailing	Valuation	Price-to-Earnings, excl. Extraordinary Items (diluted) to 3-Year past EPS Growth
Book/Market	bm	Valuation	Book Value of Equity as a fraction of Market Value of Equity
Shillers Cyclically Adjusted P/E Ratio	capei	Valuation	Multiple of Market Value of Equity to 5-year moving average of Net Income
Dividend Yield	divyield	Valuation	Indicated Dividend Rate as a fraction of Price
Enterprise Value Multiple	evm	Valuation	Multiple of Enterprise Value to EBITDA
Price/Cash flow	pcf	Valuation	Multiple of Market Value of Equity to Net Cash Flow from Operating Activities
P/E (Diluted, Excl. EI)	pe_exi	Valuation	Price-to-Earnings, excl. Extraordinary Items (diluted)
P/E (Diluted, Incl. EI)	pe_inc	Valuation	Price-to-Earnings, incl. Extraordinary Items (diluted)
Price/Operating Earnings (Basic, Excl. EI)	pe_op_basic	Valuation	Price to Operating EPS, excl. Extraordinary Items (Basic)
Price/Operating Earnings (Diluted, Excl. EI)	pe_op_dil	Valuation	Price to Operating EPS, excl. Extraordinary Items (Diluted)
Price/Sales	ps	Valuation	Multiple of Market Value of Equity to Sales
Price/Book	ptb	Valuation	Multiple of Market Value of Equity to Book Value of Equity

Table A48: Summary Statistics Filing Outcomes

Year	Bankruptcies	Survived	Tortious Bankruptcy	Long Legal Process	Average Duration	363 Asset Sale	Total Assets (Millions)
1980	1	1	0	1	1157	0	514
1981	3	3	0	3	1627	1	4,501
1982	7	5	1	7	920	0	12,212
1983	3	2	0	3	951	0	3,306
1984	5	5	0	5	771	1	8,623
1985	4	4	1	4	830	0	4,515
1986	6	4	2	6	1240	1	19,215
1987	5	5	2	3	597	1	79,864
1988	6	5	0	6	755	0	79,930
1989	8	3	0	7	937	0	44,385
1990	20	15	4	17	845	2	46,403
1991	19	14	2	13	815	1	63,652
1992	16	12	0	10	511	0	44,737
1993	13	10	2	8	523	1	8,683
1994	5	3	0	3	454	0	2,413
1995	9	7	0	7	761	3	13,480
1996	10	4	0	4	356	2	10,216
1997	9	4	0	5	805	3	9,475
1998	16	8	2	10	698	4	13,685
1999	26	15	1	14	519	5	29,450
2000	45	24	2	30	635	12	64,832
2001	56	32	6	33	578	14	221,487
2002	35	23	4	19	546	6	217,744
2003	35	21	5	19	670	13	53,109
2004	14	11	0	5	406	1	21,957
2005	15	11	1	10	612	4	93,852
2006	7	6	0	3	318	0	15,504
2007	10	5	0	4	537	3	70,583
2008	26	13	1	16	726	14	1,245,430
2009	56	33	2	23	398	20	402,871
2010	23	11	0	8	421	9	68,774
2011	10	6	0	4	318	2	57,132
2012	17	7	0	8	344	6	28,226
2013	14	9	0	0	150	5	16,992
2014	10	5	0	4	385	5	53,336
2015	16	6	2	6	348	7	51,247
2016	23	19	0	3	186	4	54,040
2017	14	5	0	0	70	0	28,405

Predicting Global Restaurant Facility Closures

Abstract

This paper predicts the likelihood that a restaurant will close within the next one to two years using a Yelp restaurant dataset and a high dimensional gradient boosting machine called LightGBM (hereafter GBM). This model, trained on more than 20,000 individual restaurants, has an accuracy just above 96% and an ROC (AUC)[34] score of 75%. An ROC (AUC) score above 70% is ordinarily classified as a "fair model" in terms of performance. Using the prediction model, I also quantify the most predictive variables and higher-order variable interactions, both of which produce compelling insights into several non-linear relationships. A model that predicts facility closures has implications for both equity and debt providers. In this chapter, I argue that capital providers should make use of publicly available datasets to aid their capital allocation decision-making process.

[34] The ROC (AUC) measure is necessary because the data set is imbalanced i.e. less closed than open firms, and the AUC measure is not invalidated on imbalanced data like the inflated accuracy score.

I. Introduction and Motivation

In this study, I make use of a comprehensive set of customer-sourced restaurant data that includes, among other things, a large text corpus of reviews and user metadata. In total, 430 unique input variables are used in the prediction model. Previous research has demonstrated the augmenting benefit that text mining plays in restaurant survival prediction, and for that reason sentiment variables are included as part of the matrix of variables in this study (Mejia, Mankad, & Gopal, 2015; Zhang & Luo, 2016). This study not only inspects the direct relationship individual variables have with the outcome, but also identifies and quantifies the higher-order interactions related to restaurant facility closures. It is the first restaurant closure model to study closures across multiple geographies. The prediction model is suitable for multinational chains and is also language agnostic.

This study contends that knowledge of future closures is not just predictive but also prescriptive in nature. For example, Luca (2017), using similar data to that of this chapter, showed that a range of characteristics, many of which also show promise in this study, can influence future revenue; in this sense these models are prescriptive by offering managers added information for resource allocation and other decisions. The variables highlighted in this chapter offer a great starting point to infer causal relationships. Knowledge of which restaurants are most likely to close could help management to 1) identify struggling facilities to provide additional assistance to, or 2) to identify which facilities to let go of. The model can also be extended to predict several years into the future so that management can intervene long before the predicted closure. A deeper understanding of the non-linear relationship of variables sheds light on improving both struggling and well-run facilities.

The performance of the classifier[35] is measured using the ROC (AUC) score and the model's statistical significance by means of a permutation technique. This is the first financial prediction study to use SHAP values to accurately and consistently estimate variables' overall contribution to the output of the prediction model (unlike frequently used Gain/Gini Index measures). In this study, I also implement iterative permutation (mean decrease in the ROC (AUC) score) to identify not just the prediction importance but also the statistical significance of each individual variable to the model. The SHAP values together with the permuted p-values provide us with measures that are akin to linear models' effect

[35] A classification model that seeks to predict whether a restaurant location will be open or close within one to two years.

size and statistical significance values. This study highlights the importance of previously unidentified variables in predicting restaurant facility closures.

Although it is possible to intuit the economic consequences for employees and customers when a restaurant closes, more work should be done to identify the aggregate patterns and economic consequences of restaurant closures. The first part of this chapter will follow a structure similar to standard machine learning papers, and the second part will focus on the variables that show the most predictive power and what they teach us about restaurant closures. I end this study by considering the implications of a successful restaurant closure prediction model.

II. Literature

Many restaurants do not actively take advantage of the large corpus of data generated daily on sites like Yelp. This is likely a result of the difficulty of obtaining and transforming data from these sources to drive operational decision making. The data goes "dark," which is a term used to refer to data that has been generated but that goes unused (Laney, 2017). Recent parsing and extraction advances have allowed for previously inconceivable formats of media such as text, pictures, and video to be transformed into streams of machine-readable data. Review websites are becoming critical for restaurant success because "[consumers have] to make decision[s] with very little information" (Luca, 2011). Yelp had around 83 million monthly visitors in Q4 2017, and Yelpers had written as many as 148 million reviews by the end of Q4 2017. By 2017, more than 80% of restaurants in the US were on Yelp.

Restaurant ratings have micro-economic implications. Findings by Taylor and Aday (2016) show that better ratings of restaurants command increased prices. Some restaurants found that certain words are strong indicators of success and failure (Mejia, Mankad, & Gopal, 2015). A recent publication from Harvard Business School showed that a one-star increase in rating can lead to a 5 to 9 percent increase in revenue (Luca, 2016); this analysis was possible due to the discontinuity of the star rating-system. A self-published report also showed how Yelp data can be used to predict the local economic outlook (Bialik, 2017). A further in-depth study shows that this type of digital data can also be extremely useful for policy analysis (Glaeser, Kim & Luca, 2017).

Kang, Kuznetsova, Luca and Yejin (2013) showed that Yelp reviews can be used to predict the outcome of restaurant health-inspections. Yelp data has further been used to show that health inspection outcomes are related to overall customer satisfaction and that improved

health ratings increase the probability of private equity buyouts (Bernstein & Sheen, 2016). In a similar fashion, I believe that restaurant closures are related to a wide range of factors one of which is likely to be customer sentiment. It is not difficult to see how customer-sourced restaurant data, like that of Yelp, can be used by parent restaurants and private equity firms alike to decide which restaurants should be kept open and which restaurants should be closed down. This paper, as well as the aforementioned research, can be very beneficial to business owners, creditors, treasurers, and investors alike. Knowledge about potential failure can significantly aid resource allocation strategies and enhance overall firm survivability and financial performance.

The focus of this study is on extracting quantitative data from customer reviews to predict the probability of closure. A large range of macro and micro factors can lead to a restaurant closing down. Macro factors like growth, minimum wage, and competition, as well as micro factors like a restaurant's access to capital, location, and overall owner competence, can all play a role. Like previous researchers, we have to question whether these factors and their correlates can be extracted from a reviewer dataset like Yelp (DiPietro, Parsa, & Gregory, 2011).

Bankruptcy prediction is common in finance and management literature (Dambolena & Khoury, 1980; Gombola & Ketz, 1983; Scott, 1981). These studies normally use linear prediction models and financial factor analysis using liquidity, solvency, and profitability measures. Restaurant bankruptcy or financial distress studies focus predominantly on the parent company, rather than the individual locations/facilities (Kim & Upneja, 2014; Youn & Gu, 2010). A lot of parallels in methodology can be drawn between firm bankruptcy and restaurant closure prediction; however, the prediction task is fundamentally different at the level of analysis and the type of data available. In saying that, it is possible that restaurant bankruptcy prediction can significantly be improved by using location-specific restaurant data as is done in this study.

In recent years, the traditional methods and processes in bankruptcy prediction have been uprooted by the development of advanced machine learning models (Barboza, Kimura, & Altman, 2017; du Jardin, 2017; Jones, 2017; Liang, Lu, Tsai, & Shih, 2016). These advanced models present myriad advantages in flexibility, efficiency, and most importantly, enhanced prediction quality (Jones, 2017). Traditional significance tests of predictor performance are also being substituted by higher dimensional classification trees and importance measures such as Gini Importance, Information Gain, and SHAP values (Behr & Weinblat, 2017; Jones, Johnstone, & Wilson, 2017; Lundberg & Lee, 2017; Mselmi, Lahiani,

& Hamza, 2017). These are all data-centric approaches that look at the predictive ability of a parameter based on the variables selected and their ranking in the nodes of the trees instead of significance tests.

III. Evaluation

A handful of variables used in the model are cumulative over time, such as the number of restaurant reviews and the number of reviewer-friends. Therefore, a period has to be selected to develop and measure the performance of a reliable closure prediction model. The years 2016-2017 have experienced especially bad periods of closures and as such provide a useful period to test robustness and generalisability of a model's performance. The National Restaurant Association's Restaurant Performance Index shows that the restaurant industry contracted in 2016 and early 2017 (National Restaurant Association, 2017). For that reason, 2016-2017 have been selected for the purpose of prediction.

The Yelp data for this period only indicates whether a respective restaurant has closed or is open and does not attach a date of closure, so one has to seek guidance from the reviews to estimate the most likely closing date. The first of the last five reviews that contains words or phrases such as "closed," "not open," and their variants are typically selected as the date of closure; otherwise, the last review date of the restaurant is used to approximate the closing date when the restaurant is highlighted as closed without any reviews mentioning the closure.

There were 5,023 restaurant closures over the test period; 2,456 were labelled as closed in 2016, and 2,567 were labelled as closed in 2017. The overall sample size is 36,544, and it includes reviews from 23 cities in the US, UK, Canada and Germany (the majority are from the US). Within this sample, around 7% of restaurants closed annually. Past knowledge on failure ratios is relevant to validate the class distribution of the dataset used in this study. The ratio of closures to failures in this study is smaller and generally more conservative than in past studies.[36] It is, therefore, quite possible that the academic dataset provided by Yelp exhibits some form of survival bias. For that reason, when this model is applied to a random sample of the general population it is likely to underpredict bankruptcies leading to more false negatives in practice. This issue could be mitigated by simply oversampling bankruptcies and retraining the model or by adjusting the decision threshold. But for this

[36] Researchers are of the opinion that the median lifetime of a restaurant is around 4.5 years, suggesting a failure rate of around 22% (Luo & Stark, 2015).

study, a smaller number of closures provides for more conservative performance metrics due to the imbalanced nature of the data.

Supervised machine learning entails the use of variables, X to approximate an outcome y using mostly high dimensional models and testing it against an out-of-sample-dataset. For that reason, traditional statisticians like to refer to it as function approximation. There are a variety of methods one can use to approximate a function. For classification models, one can, among others, use logistic regression, neural networks, and decision trees. The best interpretable machine learning solution used in practice today is gradient boosting machines; for example, it is the primary model used by Uber (Purdy, Chen, & Sumers, 2017). It has gained so much traction that Microsoft has created their own implementation, known as LightGBM (Ke et al., 2017). This is the version used in this study. These models have been widely applied in the last few years; for example, economists have started using gradient boosting machines to better understand important policy and decision-making implications in legal and medical settings (Jung, Concannon, Shroff, Goel, & Goldstein, 2017; Kleinberg, Lakkaraju, Leskovec, Ludwig, & Mullainathan, 2018).

The input variables to the model are formed by an extensive process of extraction and transformation. The final predictor variable count is 430, and it includes geographical measures such as restaurant density, neighbour entropy, number of reviews, and average rating by gender. Each restaurant-year observation is described according to a set of variables and a response value. This trained algorithm is then used with new inputs against a pure holdout test set to assess its accuracy and ROC (AUC) score. A study applied to the 'Yelp of China' achieved an average ROC (AUC) of 72% after combining three different models related to mobility, geographic, and review analysis (Lian, Zhang, Xie, & Sun, 2017). In this study, I include a larger set of variables and I only use a single self-contained model to predict closures globally.

In the process of developing the model, 60% of the observation is dedicated to the training set, 15% to the development (validation) set, and 25% to the test set. The 15% development set is used to measure and improve the performance of the model. As part of the development, I used a 10-fold cross-validation procedure in time series to ensure against information leakage and drifting variables. Cross-validation is implemented to take full use of a somewhat small development set. As part of this process, I implement an automated grid search procedure to identify model hyper-parameters. After the development process, the final model is applied to the test data and the metrics are reported.

Accuracy is defined as the percentage of correctly classified instances by the model. It is the number of correctly predicted bankrupt (true positives) and correctly predicted healthy firm years (true negatives) in proportion to all predicted values. It incorporates all the classes into its measure $(TP + TN)/(TP + TN + FP + FN)$, where *TP*, *FN*, *FP* and *TN* are the respective true positives, false negatives, false positives, and true negatives values for both classes. The measure can otherwise be represented as follows:

$$acc(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i) \qquad (17)$$

The most important metric is the ROC AUC (receiver operating characteristic's area under curve). The ROC curve is a popular statistic in classification research (Bradley, 1997; Fawcett, 2006; Ferri, Flach, & Hernndez-Orallo, 2002). Its use in bankruptcy research has also accelerated; in the last year alone, more than eight studies within neural network and boosted tree model bankruptcy prediction research have made use of this method. This method is widely used because accuracy does not work well on imbalanced data, i.e., where there are far fewer failing firms than healthy firms.

Table 49: Binary Classification Performance for Predicting Restaurant Closures.

Binary Classification Model	ROC AUC Sore	Accuracy Score	Model p-value	False Negative Rate	False Positive Rate	Cross-entropy
Year 17	0.75	0.963	0.008	0.86	0.012	1.29
Years 16-17	0.78	0.964	0.005	0.78	0.010	1.22
20 Variables	0.69	0.961	0.018	0.91	0.015	1.34

This table reports six metrics for three different types of classification tests to predict restaurant failure. The first test only predicts the failure for restaurants one year ahead. Data is gathered until end of 2016 and closures are predicted for the year ending 2017. The next test is somewhat easier; data is gathered up until the end of 2015 after which closures are predicted to occur within the next two years, 2016-2017. The '20 Variables' predicts one year ahead (2017) using only the top 20 variables.

The model performance (ROC) does not change too drastically when we predict closures for only one year as opposed to two years in advance (*Table 32*). 'Year 17' is used to predict whether the firm would fail within the next year, 'Years 16-17' are the default model used in this chapter and are used to predict whether the firm would fail in the next two years. A further model that is included, is one that is developed to use only the top 20 predictor

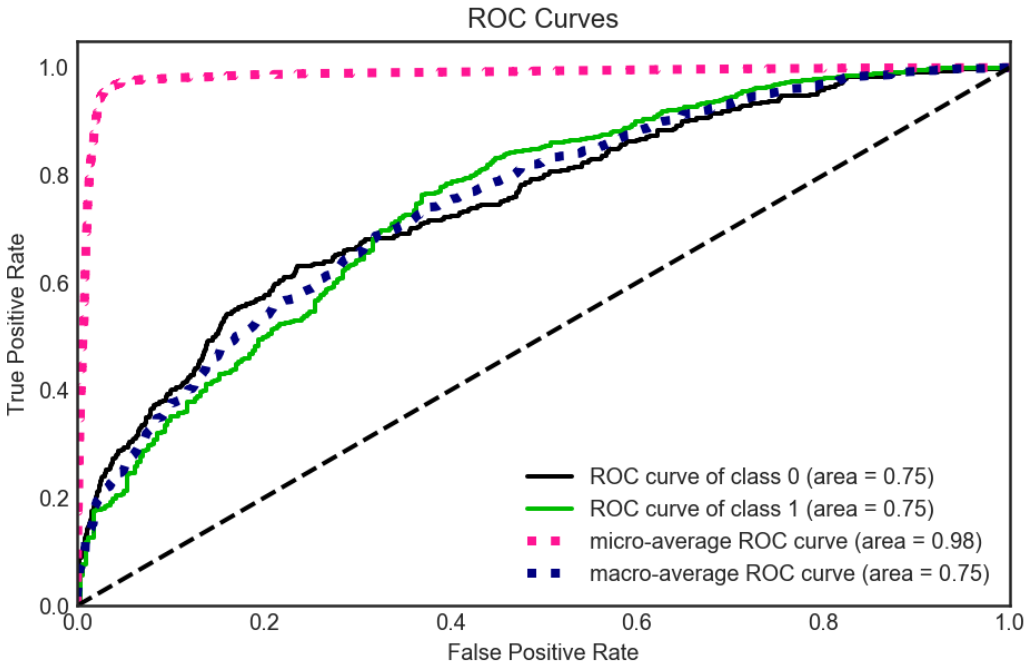
variables as identified by importance scores of the full model. I would ordinarily expect a smaller drop in performance; but in this case, it seems that many variables outside of the top 20, play an important role in the prediction process. This is also illustrated in *Figure 34* and *Figure 35*, which shows the distributed importance of a multitude of variables.

An additional procedure, *Model p-value*, is undertaken to identify the statistical significance of the classification score. This is obtained by repeating the classification exercise after randomising and permuting the labels. The p-value is then given by the percentage of runs for which the score obtained by the random model (noisy model) was greater than the classification score of the original model.[37] I set the number of permutations to 500. Thus, for a p value of 0.008, if the classification model was run 500 times, the noisy model would exceed my initial classification score 4 times.

This study also reports on accuracy, false negative rate, false positive rate, and cross-entropy metrics. The accuracy measure is not well-suited for imbalanced sets and can largely be ignored unless it is used as part of the development of the model for comparative reasons. The issue of the accuracy measure is that it does not look at class-breakdown precision, nor does it provide evidence of true positives or true negatives values. The false positive and negative rates similarly serve a somewhat limited role in this study as it relies on the probability threshold that is manually decided by the researcher. In fact, the ROC score is the combination of these two measures across all levels of the probability thresholds (0-100%), hence it being a more holistic measure. The last reported metric is the cross-entropy measure; it serves a purpose similar to the ROC measure but stresses a probability interpretation of model prediction. The cross-entropy measure principally serves as a corroborative measure. For model quality and prediction quality, I urge the reader to focus on the ROC measure as tabled above and plotted below.

[37] This method was developed by Alexandre Gramfort, the author of Scikit-learn: https://scikit-learn.org/stable/auto_examples/feature_selection/plot_permutation_test_for_classification.html#sphx-glr-download-auto-examples-feature-selection-plot-permutation-test-for-classification-py

Figure 33: ROC (AUC) Curves for Binary Classification Model



The ROC curve is simply the relationship of the true positive rate to the false positive rate with respect to a probability threshold. The horizontal curve can be described as the line of luck and has an AUC of 0.5. Generally classifiers should perform better than 0.5 to be of use at all. An AUC score of 1 represents the best possible classification score with no Type I and Type II errors. Conventionally, AUCs above 0.8 and 0.9 demark good and great classifiers that produce a good balance between true positive and false positive rates across a range of probability thresholds. For a visualisation of the ROC curve, the Type I and Type II errors have to be plotted against all threshold values. The macro-average measure is equal weighted to each class and a micro-average measure looks at each observation weight

The analysis also includes the use of a confusion matrix, *Table 50*. This study solves for a binary classification problem that produces a 2×2 matrix on the out-of-sample test data. The columns of the matrix represent the predicted values, and the rows represent the actual values for closed and open restaurant prediction. In the cross-section of the rows and columns, we have the True Positive (TP), False Negative (FN - type II error), False Positive (FP - type I error) and True Negative (TN) values. It is important for a classification study to produce a classification matrix especially when the dataset is imbalanced, such as is the case in restaurant facility closure prediction, in which a small minority of the observations are closures.

Table 50: Open and Closed Confusion Matrix

Aggregated Open and Closed Firms Matrix		**Predicted**		Sample Proportion
		Open	Closed	
Actual	Open	**8726 - TN**	89 - FP	0.970
	Closed	233 - FN	**37 - TP**	0.03
Precision		0.974	0.29	9085
Improvement		0.004	0.26	-

This restaurant closure prediction task solves for a binary classification problem that produces a 2×2 matrix. The columns of the matrix represent the predicted values, and the rows represent the actual values for closed and open restaurant predictions. In the cross-section of the rows and columns, we have the True Positive (TP), False Negative (FN - type II error), False Positive (FP - type I error) and True Negative (TN) values. The sample proportion on the far right is equal to all the actual observations of a certain classification divided by all the observations. The *precision* is calculated by dividing the true positives (Closures) by the sum of itself and the false negatives (Open). An example along the second column: $37/(37+89) = 29\%$. The improvement is the percentage point improvement the prediction model has over a random choice benchmark.

The good performance in *Table 50* can further be highlighted by drawing up a confusion matrix from random guessing. *Table 51* shows the performance of random guessing based on knowledge of the underlying distribution. There is a big difference between the distribution of closure predictions in this table compared to the model-predicted table. The performance of the table above is much better than random predictions based on the underlying sample distribution. The random guessing model correctly predicted 8 out of 270 predicted failures. This equals a precision of just over 3%, which is much worse than the model's 29%. In general, we want TN to be larger and FN to be lower for all categories predicted.

Table 51: Random Guessing Aggregate Confusion Matrix

Aggregated Open and Closed Firms Matrix		**Random Guess**		Marginal Sum of Actual Values
		Open	Closed	
Actual	Open	**8551 - TN**	262 - FP	8813
	Closed	264 - FN	**8 - TP**	272
Marginal Sum of Guesses		8815	270	9085

This table is formed by 'randomly choosing the observations' by allocating the observations according to the underlying distribution, as presented by Sample Proportion in *Table 50.*

IV. Predictor Variable Analysis

A clear benefit of the GBM model is that a wide range of variables contributes to the overall prediction unlike most conventional models; there is evidently a reasonably even distribution of variable importance across multiple predictor variables (*Table 22*). In *Table 22* we can see the importance meta-type variables take in predicting failures; in many ways they seem to be more important than variables directly derived from reviewers' opinions. For example, the first two values relate to the date of the first review for the individual restaurant and the average first date of the entire chain. Before the advent of modern machine learning techniques, researchers would have had to choose among variables with selection techniques and might have neglected to include these variables, that prima facie, seem unlikely to improve the performance of the model. The benefit of GBM models is that multicollinearity does not impair the predictive performance of the model to the extent of conventional linear models, and for that reason we can include a wide range of variables and leave it to the model to disregards redundant variables.

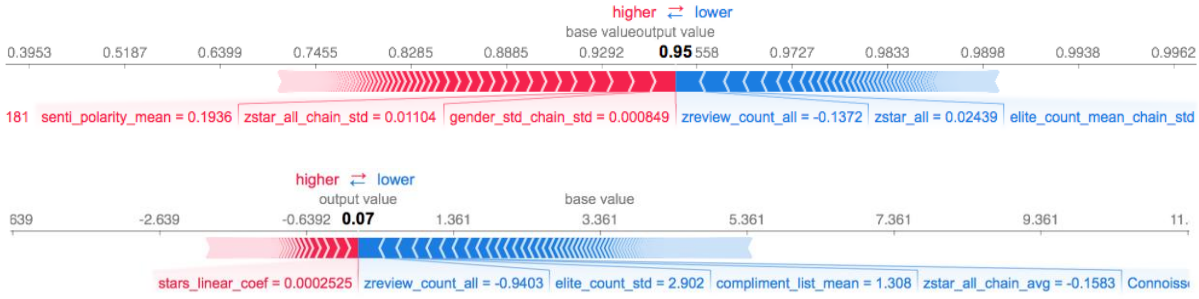
There are a few things to keep in mind when using models based on impurity rankings. Firstly, variable selection based on impurity reduction is biased towards variables with more categories. Secondly, when the dataset has two or more correlated variables, then from the point of view of the model, any of these correlated variables can be used as the predictor variable, with no concrete preference of one over the other. But once one of them is used, the importance of others is significantly reduced since the predictability they ought to account for has already been accounted for by the first feature.[38] As a last step to identify variable importance, the statistical significance of the twenty most important variables is calculated with an iterative permutation technique that shuffles the variable and retrains the model as well as re-calibrate the hyperparameters. In that way, it can be established whether the model, without that particular variable, would be statistically worse off.

Within *Table 22* there is a danger of interpreting the machine learning feature importance outputs incorrectly. These measures should be approached with some nuance. The Gain measure is the average training loss reduction gained when a variable is used for splitting. A good feature importance measure should be accurate and consistent. The Gain

[38] The effect of this phenomenon can be reduced by randomly selecting a subset of variables at each node creation which would allow the overall importance to be equally distributed between two similar variables. This still doesn't get one all the way there because the importance is still diluted, albeit more fairly. The only way to deal with this is to remove correlated features or to group correlated features together, similar to the PCA components in chapter two.

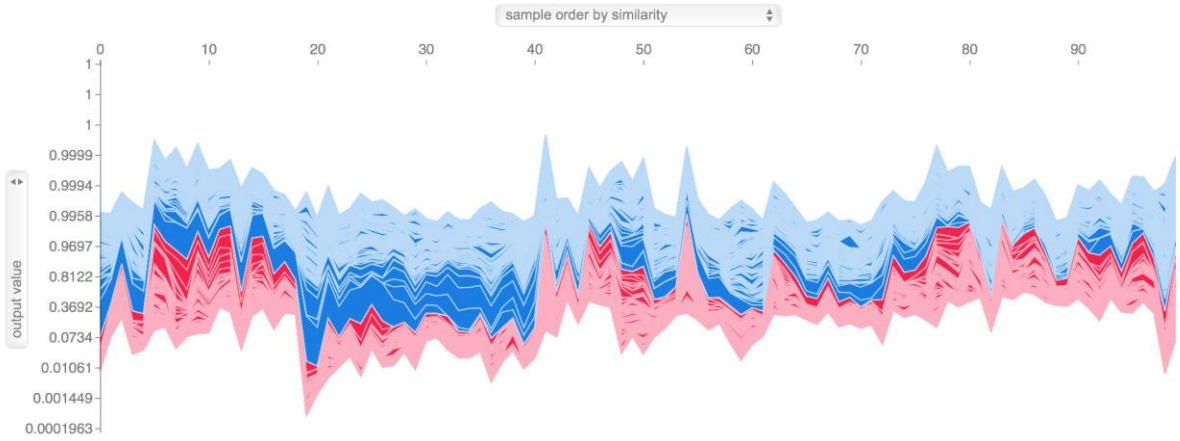
measure is widely used but can often lead to inconsistent results. Inconsistent here refers to the possibility of a large shuffle in importance values occurring when additional data or variables are added to or removed from the model. SHAP values, on the other hand, provide for a good alternative; SHAP is a fast-practical algorithm and it has a solid theoretical underpinning (Lundberg & Lee, 2017). The SHAP values for LightGBM explain the margin output of the model, which is the change in log odds. In *Table 22*, I show the relative contribution of SHAP and Gain values as well as rankings. An important benefit of SHAP is that it provides the aggregate direction a feature has in relation to the response variable. *Figure 34* shows how the SHAP values interact with each other to produce the final output. Although, a few variables cause a large amount of the movement, *Figure 34* also shows that less important variables play a large role in aggregate due to the sheer number of variables present. *Figure 35* shows these predictions vertically while sorting them by similarity to show the effect these different variables have on the final outcome. Here, I have only plotted a subsample of a hundred random observations. The final output is where the red and blue stacks meet. Each vertical slice is a 'DNA strand' of sorts displaying the characteristics driving the predicted outcome of each observation.

Figure 34: Feature Effect on Log-odds and Model Output for Single Observations



The plots above show an example of the predicted outcomes for two different restaurants' observations. In the first figure, the final output is the probability of restaurant closure (0) and the probability of restaurant success (1). The bottom figure plots the log-odds output. As can be seen from the plots, many variables lead to the final predicted outcome. The top observation is predicted to remain open and the bottom observation is predicted to close within the next two years with a decision threshold (rule) of 50%.

Figure 35: Feature Effect on Model Output for a Subsample



This plot replicates the top plot in *Figure 34* but for a subsample of 100 firms that are vertically plotted as opposed to just one horizontal observation. The samples are sorted by similarity and stacked next to each other. The y-axis is the probability of remaining open (1) or closing down (0). In this plot, the predicted outputs are where the blue and red boundaries meet. The variable effects are stacked vertically. Like in the previous figure, red variables push the outcome towards remaining open, and blue variables push the outcome toward closure. This plot is a close analogue to a DNA strand; the intensity and direction of each of the 430 characteristics ('genes') determine the final outcome ('trait').

Table 52: Predictive Power and Significance of Variables

Predictor	SHAP	SHAP Rank	Direction	Gain	Gain Rank	Permutation p-value
oldest_review	100	1	+	54	3	0.001
oldest_review_chain_avg	86	2	+	97	2	0.001
useful_mean	51	3	+	25	9	0.000
stars_linear_coef	50	4	+	44	4	0.001
useful_sum	49	5	-	100	1	0.001
gender_std	43	6	-	9	51	0.002
restaurant_density	42	7	+	24	11	0.006
Connoisseur	39	8	+	21	14	0.005
rating_sum	38	9	-	7	78	0.003
reviews_per_week	37	10	-	16	21	0.006
latitude	33	11	-	19	16	0.008
compliment_plain_mean	33	12	+	14	31	0.002
zreview_count_all	33	13	-	23	12	0.007
average_stars_mean	32	14	-	20	15	0.007
zstar_all_chain_std	32	15	-	17	20	0.004
elite_count_mean	31	16	-	12	38	0.009

first_sent_mean	31	17	-	14	32	0.010
Male to Female	30	18	-	10	45	0.006
first_sent_std_chain_avg	29	19	+	18	17	0.012
zreview_per_week_all	29	20	-	39	5	0.009

The above table lists the twenty most important variables in the prediction task as measured by their SHAP values. See *Table 53* below for the associated definitions. The SHAP value is one of only a few known, accurate, and consistent feature-importance measures. The Gain measure is also included for comparison. Both the SHAP and Gain measure are reported in relative terms to the most predictive feature. The reported SHAP values are absolute; the direction is there to indicate the relationship of the feature with the model output. Finally, the permuted p-value is calculated by identifying the change in model output as a result of randomising and permuting each feature 1000 times and identifying the number of times the permuted model outperformed the original model in terms of the ROC AUC metric. As an example, a p-value of 0.001 means that one out of all the random permutations leads to a better result than the original model. The permuted model simply included the 20 variables listed above as an attempt to minimise the likelihood that correlated variables lower down the importance ranking absorb the effect. From this table, it is clear that the different measures of importance do not disagree all that much for the first five variables, after which there is some divergence.

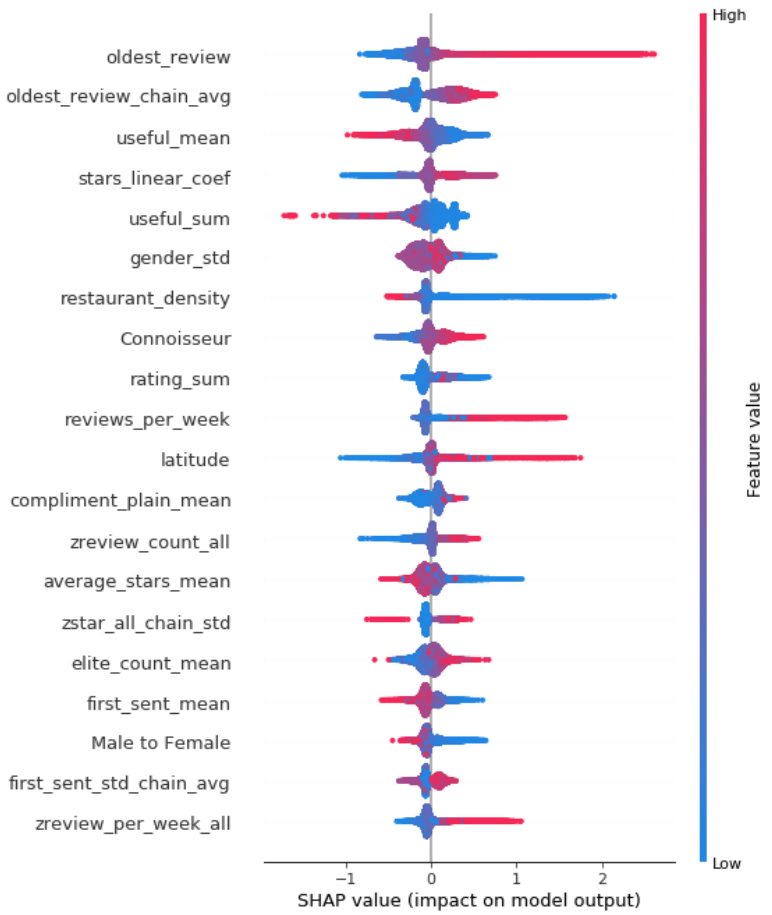
Table 53: Predictor Definitions

Name	**Definition**
oldest_review	Weeks since the first review of the restaurant.
oldest_review_chain_avg	Average weeks since the first review of the restaurant for the entire chain.
useful_mean	The extent to which readers of reviews on average deem information as useful. Reviews deemed “useful” by other members are generally either critical or thrift in nature, offering information about deals and specials.
stars_linear_coef	Measures the extent to which the rating is increasing or decreasing over time in the form of a slope coefficient.
useful_sum	The sum of useful comments.
gender_std	The standard deviation of a binary measure that labels 1 for male and 0 for female.
restaurant_density	A measure of the number of firms in a one-mile radius that are classified according to the same restaurant categories, for example, salad bar, bistro or ethnic.
connoisseur	The rating of 'yelpers' in the top median of contribution in terms of the number of restaurant reviews given.
rating_sum	The aggregate sum of all ratings.
reviews_per_week	The number of reviews the restaurant gets per week.
latitude	The angular distance of a place north or south of the earth's equator.
compliment_plain_mean	The average number of 'compliments' reviewers of the restaurant receive.
zreview_count_all	Difference between total reviews of the restaurant and that of cuisine specific neighbours in a one-mile radius.
average_stars_mean	The average stars all reviewers gave the firm.
zstar_all_chain_std	The standard deviation of the difference in the restaurants average rating as

	compared against surrounding competitors.
elite_count_mean	The average amount of elite customers that reviews the restaurant.
first_sent_mean	The negative sentiment from the first half of the restaurant's existence.
Male to Female	A ratio of the number of males to female reviewers.
first_sent_std_chain_avg	The chain-wide standard deviation of the polarity of customer sentiment.
zreview_per_week_all	The number of reviews per week the restaurant receives compared to the competitive neighbours in a one-mile radius.

Now that the most important variables have been isolated, we can create a type of violin plot that presents the variables in order of importance on the y-axis and the effect the variable has on the model on the x-axis (*Figure 36*). The plot echoes the results as presented in *Table 22.* The coloration (red or blue) of the plot is effective in displaying whether the variable value is increasing or decreasing in relation to the output. Open restaurants are labelled with a 1, and closed restaurants are labelled with a 0. When the colour is red, the value is positive, and when the colour is blue, the value is negative. Thus, when the colour is red on the right-hand side then an increasing variable value leads to an increase in output, i.e., an increase in the likelihood of the firm being open. Moreover, everything to the right of the origin on the x-axis is indicative of an open restaurant prediction and everything on the left is indicative of a closed firm prediction. This figure also provides some evidence of the distribution of output effects as indicted by the vertical thickness. As a result of the non-linear nature of decision trees, you can see that many input variables' effects on the output are not normally distributed; there are long tails both to the left and right, different levels of kurtosis, and sometimes bimodal distributions.

Figure 36: Distribution of Individual Feature Effects on Output



This chart reports all the predicted outcomes for each individual variable. It takes on the form of multiple horizontal violin plots, in that it not only reports the effect on outcome but also reports the variable size based on the continuous colour legend and the distribution of the outcomes though the vertical thickness.

The older the oldest review, *oldest_review*, the more likely the restaurant is to stay in business. This variable likely proxies for how long the restaurant have been in business. It is somewhat intuitive that firms that have lasted for longer will last for longer still. This is often called the Lindy Effect (Eliazar, 2017). Recent literature from China has also found a similar relationship (Zhang & Luo, 2016). Furthermore, it is well documented that restaurants that have started more recently are more likely to fail as opposed to those that have existed for a longer time (Luo & Stark, 2015).

All measures have been transformed to also provide *chain values*; these are values that essentially incorporate all facilities with the same names in an aggregate chain measure (e.g., Subway, Olive Garden, and TGIF). For independent restaurants that do not belong to a chain, the chain value is simply the same as that of the individual restaurant. The second most

important variable is in fact one of these aggregate chain measures; *oldest_review_chain_avg* looks at the average number of days since the first review of each restaurant in the chain. Given the high rank of this variable, it has to be that the aggregate chain measure provides additional information essential to the prediction exercise. Therefore, where other restaurants of the same brand have survived a long time, it is more likely that the individual location itself will last a long time and remain open, in other words, the older the chain the better for the individual facilities.

A very interesting association with failure is a metric that calculates whether readers of reviews on average deem reviews left by other users as useful, *useful_mean*, and a metric that takes the weight or sum of useful reviews, *useful_sum*. On Yelp, readers can rate reviewers' comments as 'useful,' 'funny' or 'cool.' Useful seems to be the most predictive variable among the three. Funny and cool have contribution ranks of 26 and 46 respectively. When reviewers give some advice (i.e., a 'useful' review) they are generally somewhat critical and thrifty in their disposition. These reviews often contain information on what dishes to order and what dishes to avoid, as well as how to take advantage of coupons and deals, probably undercutting the facility's profit.

Another notable variable is compliments, *compliment_plain_mean*. This metric relates to the historical attributes of the user who posted the review and not the review itself. Compliments can be traced to the perceived niceness of reviewers in general on the platform; it can include one of the following seven descriptors: 'Great Photo,' 'Good Writer,' 'Cute Pic,' 'Hot Stuff,' 'Like Your Profile,' 'You're Cool.' All of these compliment measures show a positive association with the likelihood of the firm remaining open, i.e. the firm is more likely to stay open if it attracts quality patrons who receive many compliments. The compliment that shows the best predictive power is, as mentioned initially, the plain compliment that does not contain any of the aforementioned descriptors. The interpretation is that if users who are overrepresented in receiving compliments attend your restaurant (and leave a review), then the restaurant is likely to remain open. Furthermore, users tend to receive compliments when they provide more positive feedback, and where there is criticism it tends to be positive criticism. This type of users may be favourably disposed to attending good establishments.

Stars linear coefficient, *stars_linear_coef*, measures whether the rating is increasing or decreasing over time. When the rating is increasing, the firm is more likely to succeed. A further prominent measure (14^{th}) is the average stars all reviewers gave the restaurant, *average_stars_mean*; this measure shows an inverse relationship to output. To understand

why, it is important to know that restaurants compete for different markets. The average McDonald's Rating (stars) is around 2.9 and TGIF Restaurant is 3.2, while the average Cheesecake Factory rating is 3.7. Additional measures show that the number of stars does not necessarily affect the probability of closure unless it is different to that of the overall chain (*chain*) or close competitors (*z*); without interacting with these variables, the measure acts like more of a proxy of the establishment's business model. For that reason, it is more important to know whether the rating is improving, *stars_linear_coef,* rather than knowing the average number of stars the restaurant has received, *average_stars_mean.*

Gender is a binary measure that labels a one for male and zero for female. Where there is a large amount of deviation in gender, *gender_std*, the firm is more likely to fail. The *gender_std* value decreases when one gender, e.g. female over male, overpowers the other. Firms that are more focused on patrons of a specific gender tends to survive longer, all else equal. A further notable value is the male to female ratio, *Male to Female,* where when the patrons consist of mostly men, the restaurant is slightly more likely to close. Considering these variables together, firms are better off catering to a specific gender and additionally better off catering to women as opposed to men.

Although restaurant density, *restaurant_density,* is not the most important variables globally, it is a very important variable for a large subset of customers as represented in *Figure 36* by its long tail to the right. It tells a very simple story; where the neighbourhood is very dense (competitive), the restaurant is slightly more likely to fail. Density is measured by the number of firms in a one-mile radius that fall within the same category; for example, are there many other salad bars, bistros, or ethnic restaurants in the area. This result corroborates past research that showed that restaurant density and ownership turnover are strongly correlated (Parsa, Self, Njite, & King, 2005).

The connoisseur rating, *connoisseur*, measures the rating of reviewers who are part of the top median in terms of the number of restaurant reviews given. The implication is that these individuals have attended the largest number of restaurants and have a certain expectation as to what constitutes a good dining or food experience. The higher the connoisseur rating, the more likely the restaurant is to remain open.

Similarly, the larger the aggregate sum of restaurant ratings, *rating_sum*, the more likely the firm is to remain open. This variable seems to be important for its interaction effects and seems to be highly non-linear in nature; the variable changes signs twice at the positive end of the model output (right side of the vertical bar). In general, it seems that the smaller the *rating_sum* measure, the more likely the firm is to fail, however, where the value

is extremely low, the restaurant is more likely to succeed (this is hard to see in the figure but is confirmed quantitatively). The obvious answer is that restaurants that have received fewer customers (ratings) are less to likely close shop because of a large initial investment, the assumption being that these are locations that have started very recently, hence they have simply not been around long enough to fail. However, as soon as these firms have a few more ratings next to their name, they are naturally more susceptible to failure as the economics have played itself out, things have equalised, and the owner as well as the customers would know whether or not the restaurant is worthy of staying open.[39]

The more reviews the restaurant receives per week, *reviews_per_week*, the less likely it is to close down. Reviews per week might be a close proxy to the number of customers who prefer your restaurant above others. A restaurant's net profit equals the net profit per customer times the number of customers; hence this measure is essential to restaurant survival. In saying that, without interacting with other variables, the above interpretation may be too simplistic as the number of reviews per week can simply refer to the type of business model the restaurant has i.e. a fast food restaurant will have more reviews than a fine dining restaurant.

The importance of a geographic measure like *latitude* is more obvious. There are structural differences between cities that lead to different rates of closures. According to this variable, restaurants in the north i.e. larger latitude are less likely to fail than restaurants in the south i.e. smaller latitude. My results corroborate past research that shows that location has a significant effect on restaurant success (Parsa, Self, Sydnor-Busso, & Yoon, 2011).

All *z-metrics* account for the average equivalent quantities of neighbouring and competing restaurants in a one-mile radius minus the same quantity from the target restaurant. Therefore, a high z-metric means that the firm's measure is higher than that of surrounding locations. These metrics are interesting and somewhat intuitive, in that when the target restaurant outperforms surrounding restaurants, the target restaurant is more likely to remain open. As an example, when the number of reviews the restaurant receives is higher than the surrounding competitor averages, *zreview_count_all*, the target restaurant is

[39] It is also worth mentioning that *rating_sum* could be a proxy for the size or type of establishment. If that is the case, medium establishments do worse than large establishments and small establishments perform the best. You thus see an inverted-U relationship with restaurant closure. A further feature, outside of the top 20, showed that restaurants with chain affiliation had a greater probability of success than independent restaurants and this corroborates past research (Parsa, van der Rest, Jean-Pierre I, Smith, Parsa, & Bujisic, 2015).

generally less likely to close. Therefore, the more customers you have received over time, as compared to the surrounding competitor restaurants, the better.

The number of reviews per week shows the same relationship, *zreview_per_week_all*. This number is affected by the popularity, business model, and size of the restaurant. A low measure shows a larger likelihood of failure. The issue with having fewer customers is twofold; although it is bad for the bottom line of the restaurant, the quality of food is often rated worse during periods of low customer flow. This can be attributed to a wide range of affects, like the fixed costs to serve a few customers and the employment of less competent employees over these periods (Kreeger, Parsa, Smith, & Kubickova, 2018).

Another measure that seems to be especially non-linear is *zstar_all_chain_std*. When a chain does not have enough restaurants operating in areas where the surrounding restaurants differ in rating from the restaurant itself, then that restaurant is more likely to close down. On the flip side, when restaurants excessively scatter so that there is a large difference in rating compared to surrounding restaurants, then they are also more likely to close down. I believe that this is a measure of diversification. When chains excessively diversify into areas in which they do not belong, they fail; however, when restaurants at least seek out the best opportunities among locations, they tend to perform well. Using this terminology, if diversification is very high or very low, restaurants are more likely to fail compared to when there is a moderate level of diversification.

The average number of elite customers that attend a restaurant, *elite_count_mean*, also shows large predictive power. Elite customers are those whom Yelp has identified as important contributors to their platform. We can assume that these people are highly knowledgeable about the best quality of experience and food, and as such it is a good sign when these customers frequent your establishment.

When the first few years' (first half) negative sentiment, *first_sent_mean,* is low, the firm is more likely to succeed. A further measure shows that when the later half's negative sentiment is higher than earlier years', the restaurant is more likely to close down. It is interesting to observe that although the sentiment score over review text provides some predictive power to the model, its contribution is not that large. This may be due to a few reasons, most notably the existence of fraudulent reviews and the requirement to translate foreign languages to English before the sentiment can be measured. Previous research shows that at least 16% of reviews are fraudulent (Luca, 2016) . These reviews cannot be stopped by the service; instead Yelp relies on other metrics such as how long the profile has been around

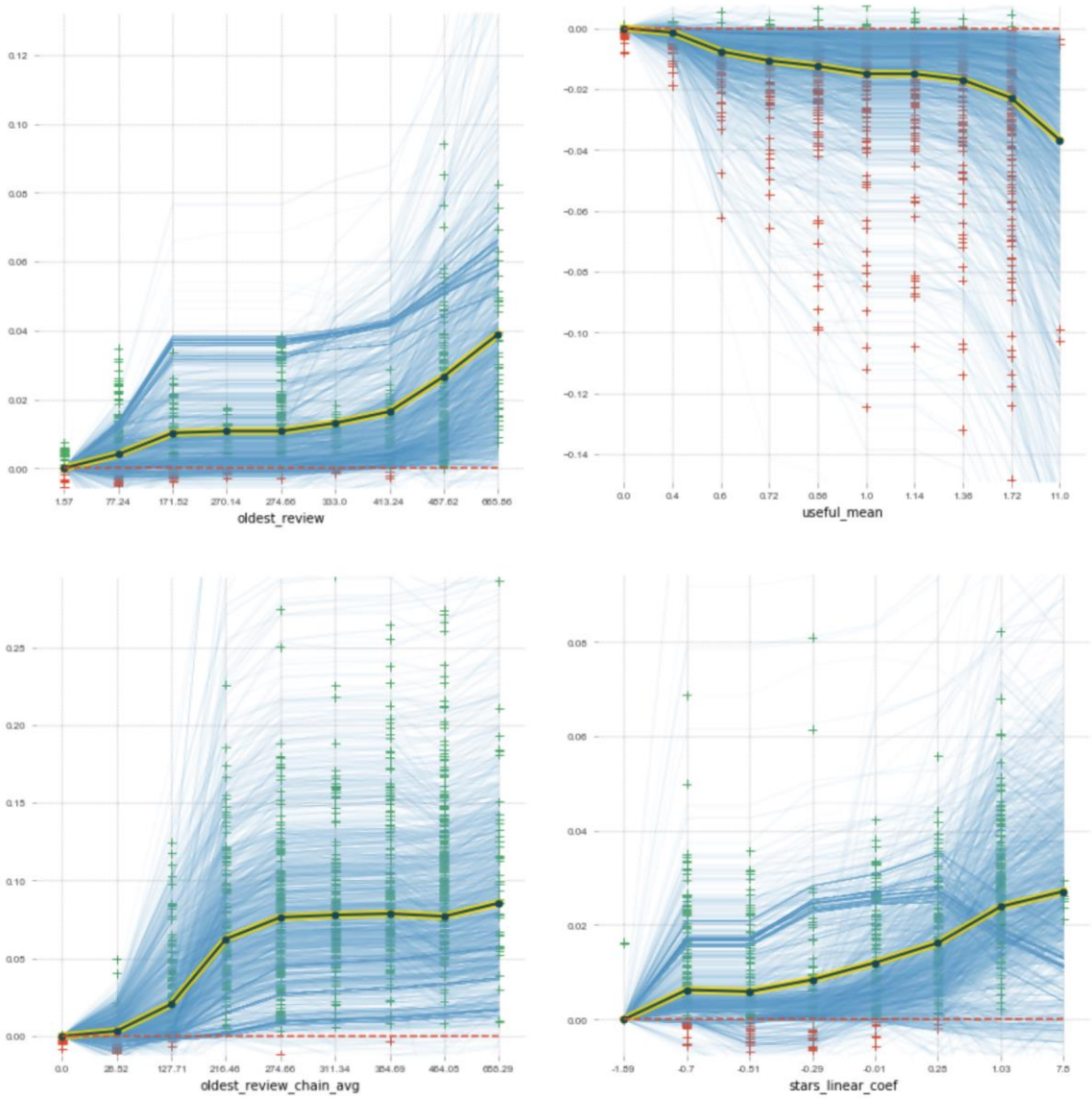
and how many reviews the person has made to score the final rating. It is very likely that this 16% leads to excessively good or bad reviews in review texts.

An analysis of reviewer sentiment shows that the majority of reviews are benign while a small minority are very negative. Fraudulent reviews can take both of these forms. Companies may engage in review warfare against a more successful competitor to digitally tarnish the competitor's reputation. On the other hand, a struggling firm can also fabricate reviews to make themselves look good in the public's eye. As mentioned previously, a further concern with regards to sentiment is the possibility of inaccurate translations from French and German to English, and this could have skewed the sentiment score because English was used as the base language from which to score sentiment. A final measure shows that polarising reviews at the firm level seem good when they are exhibited by the entire chain, *first_sent_std_chain_avg*. When reviews are not polarising, restaurants are more likely to close; again, I feel that this relates to the one-sided nature of fraudulent reviews. Polarising reviews could be a sign that the sentiment score is more trustworthy.

V. Interaction Analysis

The issue with many machine learning models is that their nonlinearity makes it hard to enforce monotonicity constraints to identify the direction of the relationship between independent variables and the machine-learned response function, especially when you use Gain as a feature importance measure. To identify the relationship of predictor variables in a machine learning algorithm, we can make use of a technique called partial dependence. Partial dependence allows us to see into the 'black-box.' Plotting the partial dependence or marginal effect produces information associated with both the direction and the strength of the relationship between explanatory variables. Partial dependence plots are the visualisation of fitted functions, and they show the effect of variables on the response after accounting for the average effects of all other variables in the model (Friedman, 2001; Friedman & Meulman, 2003). In simple terms, it is a method to identify the marginal dependence between the predictor variables and the outcome variable (Hastie, Tibshirani, & Friedman, 2009).

Figure 37: Individual Conditional Expectation Plots (Depth One)



These plots point out the non-linear nature of the top 4 variables. These figures report the marginal relationship of the feature with the predicted outcome for all samples. The green marks are open restaurants; the red marks are closed restaurants. The black line (yellow outline) presents the variables' marginal effect on the predicted outcome for all observations around the central points on the x-axis. The blue lines are an indication of how all other variables further affect the observations to produce the final outcome. (1) Top left is the oldest review of the restaurant in number of days; the older it is, the less likely the restaurant is to close. (2) Top right is the number of useful (critical) reviews; the most critical the review, the more likely the restaurant is to close (3). Bottom left is the average age of the oldest review across the chain in number of days; the older the average first review across the chain, the less likely the individual restaurant is to close. (4) And bottom right measures the slope of historic ratings as measured by stars out of five; the larger the slope, the less likely the restaurant is to close.

Table 54: Interaction Analysis (Depth Two)

Term 1	Sign	Term 2	Sign	RII	Fig.
oldest_review_chain_avg	+	useful_sum	-	100	7 (1)
useful_sum	-	zreview_per_week_all_chain_avg	+	38	7 (2)
Food Aestheticist_chain_avg	+	useful_sum	-	33	7 (3)
oldest_review_chain_avg	+	zstar_all_chain_std	-/+	31	7 (4)
elite_count_std_chain_avg	-	oldest_review_chain_avg	+	30	8 (1)
oldest_review	+	useful_sum	-	23	8 (2)
Number of Reviewers	-	Number of Reviewers_chain_avg	+	22	8 (3)
rating_mean_chain_avg	+	zreview_per_week_all_chain_avg	-/+	20	8 (4)

Out of the 430 variables, there is a near-infinite number of ways to conjure up directional relationships. To understand the web of relationships, it is best to identify the top interaction pairs as they contribute to improved predictions. This table represents the most important interaction pairs as measured by the relative interactive importance (RII) using the gain statistic at an interaction depth of two. The sign indicates the average direction of each predictor variable as read from the partial dependence plots. The interaction terms are much more informative than single standing variables. Interactions are at the core of what gradient boosting tree models are all about.

To understand some of the higher-level interactions, I have isolated the top eight interactions in terms of their predictive importance. It is notably hard to conceptualise interaction; for that reason, I include supporting figures. For the most part, these interactions happen as expected. I will comment on a few of the results, the first being *Figure 38 (2),* page *199*. It is the interaction plot between the sum of useful (generally critical) reviews on the x-axis and the total reviews per week as compared to surrounding locations averaged across the entire chain of restaurants on the y-axis. The larger the number of reviews per week relative to surrounding competitors and the lower the criticality of those reviews for the target restaurant, the better. The number of reviews per week can speak to two effects; first, the more customers the restaurant chain serves, compared to surrounding competitors, the larger the *zreview_per_week_all_chain_avg* value would be; another reason why this value may differ from that of competitors is the size of the establishment; the larger the restaurant, the more capacity they have to serve customers. In summary, a lower probability of closure is associated with more service, larger capacity, and fewer critical reviews.

Research by MacGregor and Lo (2015) showed that when establishments are affiliated with large chains, they are more likely to survive if frequented by transitory customers, whereas establishments of smaller chains endure longer in markets with local customers. In that regard, the *useful_sum* (criticality) measure in *Figure 38 (2)* is an understandably good measure of the extent to which customers are transient. Local respondents would not go on review sites as it is assumed that they already know all there is

to know about their favourite local. And you can see from this plot that when the firm is large the effect of transient clients is less of an issue, corroborating the results of MacGregor and Lo (2015).

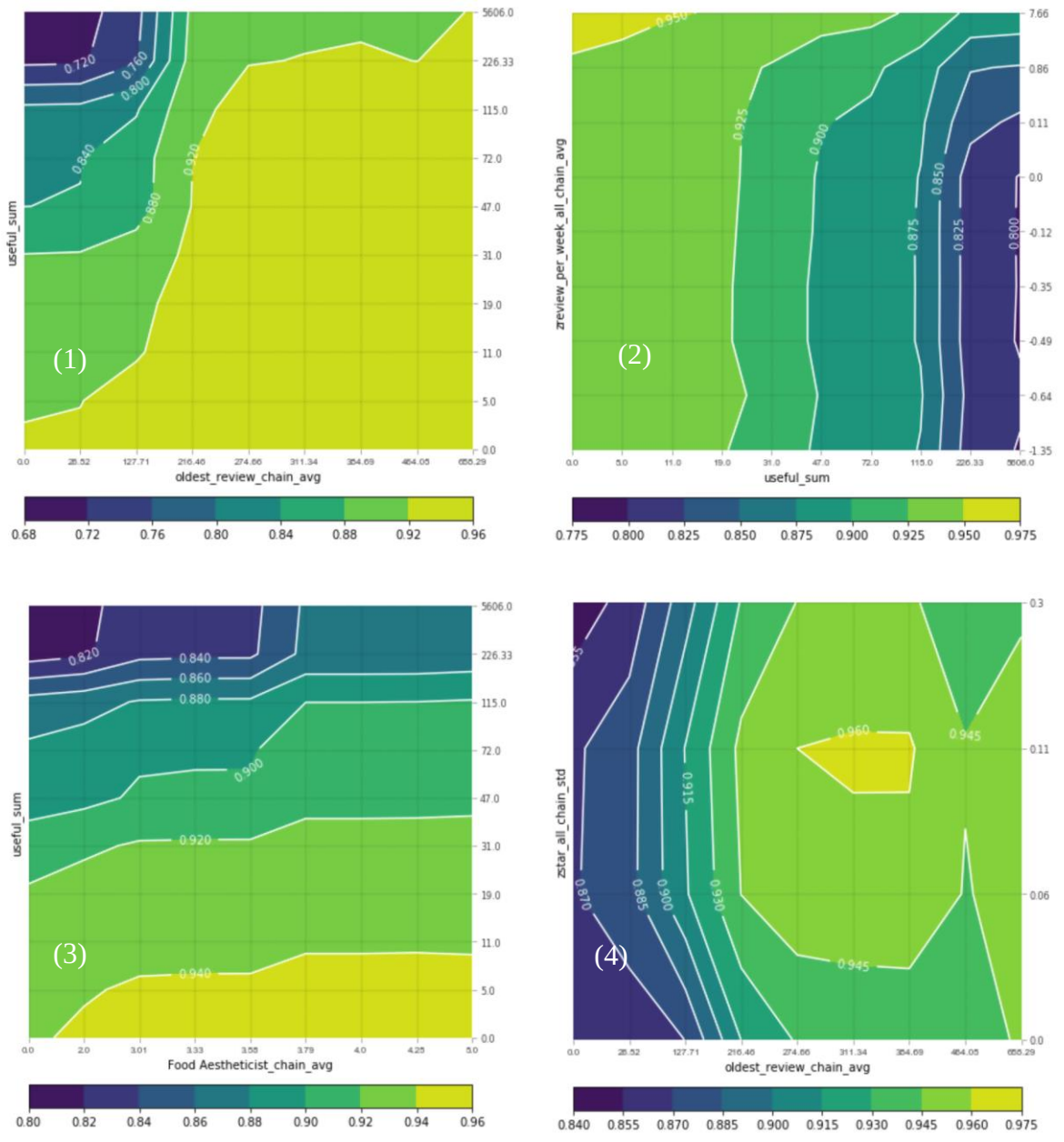
Another interesting plot is *Figure 38 (4).* This interaction plot shows that there is a sweet spot in terms of the age of the average locations among the chain, oldest_review_chain_avg, and the chain-wide deviation of the extent to which the target location outperforms surrounding firms in their star rating, zstar_all_chain_std. I previously referred to this as a measure of diversification of the firm among different area profiles e.g., affluent, middle class, hipster. If an American casual-dining chain only operates in areas that are very receptive to casual-dining clientele and are not willing to risk it in areas where the surrounding ratings may be higher, signifying different clientele and competition, then they are less likely to remain open. The opposite also holds true: when the diversification is too high, the location is more likely to fail. A good level of diversification is where the measure, zstar_all_chain_std approaches 0.1 and where the firm is neither too old nor too young, all else being equal.

A further interesting interaction occurs in *Figure* 39 *(1)*. The plot shows that when there is a lot of deviation related to whether or not patrons are registered as 'elite,' the restaurant is more likely to fail. 'Elite' consensus as to whether or not a restaurant is good is therefore a good measure of a well-functioning restaurant. Where they are not consistent, it may mean that elite patrons' attendance is highly dependent on the individual restaurant's performance rather than the chain brand, therefore making the individual location more susceptible to failure as a result of additional idiosyncratic risk. And like before, the older the chain, oldest_review_chain_avg, the better. The age of the chain starts to matter less when the success of the restaurant is more reliant on restaurant capability than the overall brand. This makes sense; even if your restaurant is old and established, once it consistently produces a bad experience, without the safeguard of brand recognition, it is more likely to fail.

Figure 39 (3) is the interaction between the average number of reviewers across the *chain* and the number of reviewers for the specific location. This plot highlights an important fact; individual locations to a chain have a reasonable time to prove themselves. If they only have a very small number of reviews as compared to other locations of the same chain, then they are unlikely to be closed down; it is the older locations, which have accumulated more reviews, that are more likely to close down. Further investigation also shows that this relationship is not strong for independent restaurants when compared against the sample of

other independent restaurants. My guess would be that when a chain introduces a new location, it subsidises its growth until it is more likely to succeed.

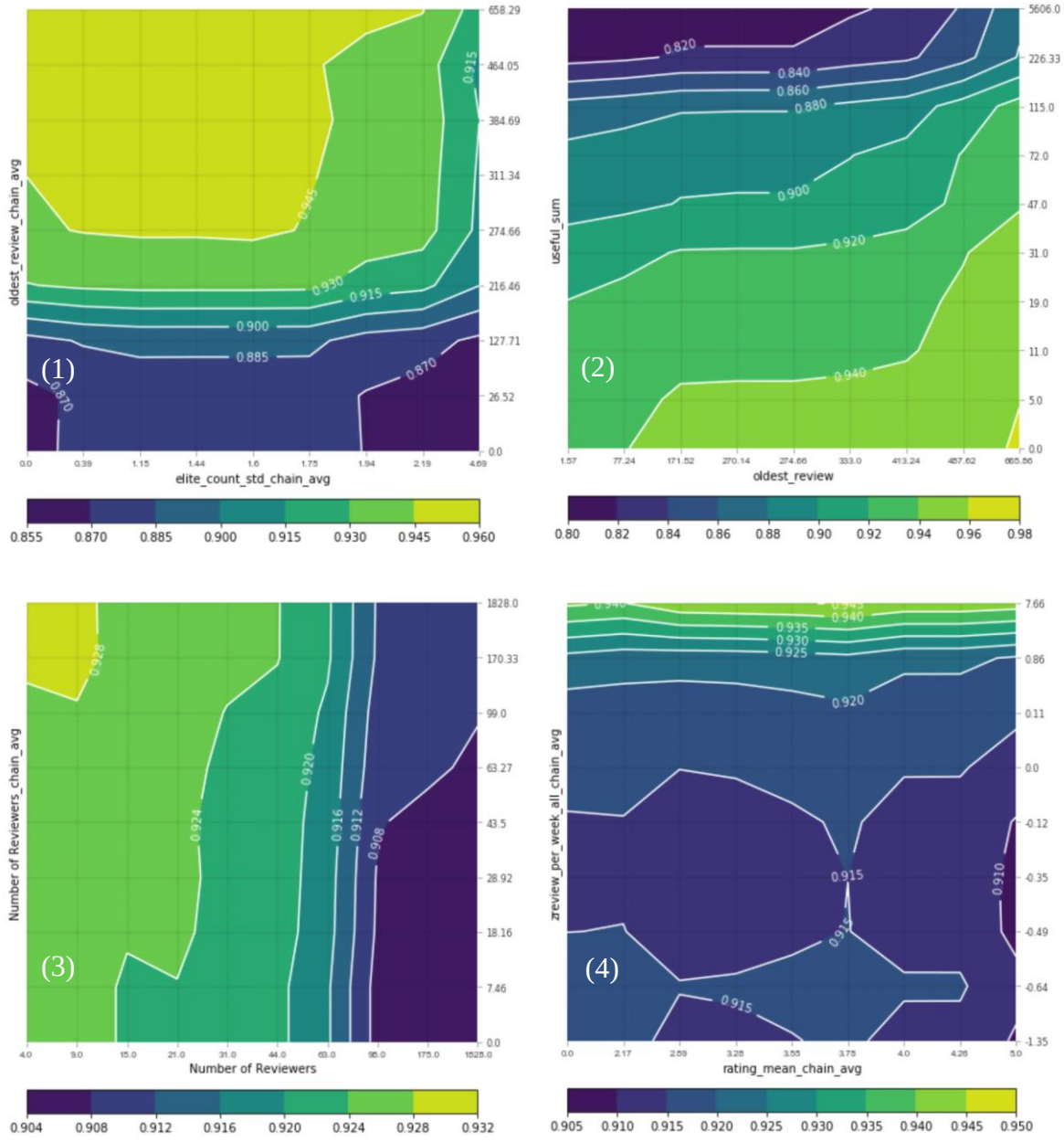
Figure 38: Interaction Pair Partial Dependence Plots (Depth Two) (A)



(1) The top left is the interaction between the aggregate sum of useful reviews and the average oldest reviews across the entire chain as reported in days. Generally, the 'older' the aggregate chain and the less useful the comments, as they relate to the individual restaurant, the smaller the likelihood of closure. (2) Top right is the interaction between the sum of useful (generally critical) reviews and the total reviews per week when compared to surrounding locations averaged across the entire chain of restaurants. Here, more reviews per week by chain compared to the competitor chains and a lower number of useful comments collectively leads to a lower probability of closure. (3) Bottom left is the interaction between the sum of useful (critical) reviews and the average number of food-aestheticists attending the chain; food-aestheticists are defined as the upper median of Yelp reviewers in their extent of taking photos of food. An enlarged attendance of food aestheticists across the chain is a good sign as long as they are not overly critical (useful-sum) of one's

restaurant (4). Bottom right is the interaction between the average oldest review across the chain as reported in days and the standard deviation of all-star ratings relative to that of surrounding competitors. The interaction between these variables highlights a sweet spot of variation in rating compared to surrounding chains and the average chain first-review age.

Figure 39: Interaction Pair Partial Dependence Plots (Depth Two) (B)



(1) The top left is the interaction between the average oldest review across the chain and the average standard deviation of elite count across the chain. Elites are a title given to very active users by Yelp. Generally, the older the individual restaurant and the smaller the variation of the number of elites across the chain, the smaller the likelihood of closure. (2) The top right is the interaction between the sum of useful reviews and the oldest review of the location. If the individual restaurant's first review is long ago and the sum of useful comments over the period is small, then the firm is less likely to close. (3) Bottom left is the interaction

between the average number of reviewers across the chain and the number of reviewers at the location. If the number of reviewers for the individual restaurant is small compared to the chain average, then the individual restaurant is less likely to close. (4) The bottom right is the interaction between the average rating across the chain and the total reviews per week as compared to surrounding locations averaged across the entire chain of restaurants.

VI. Implications and Future Research

This paper demonstrates a novel approach to predicting restaurant closures using data from Yelp, a restaurant review site. As previously mentioned, research suggests that higher-rated firms on Yelp are able to command increased prices leading to higher revenue; a one-star increase can lead to a 9% increase in revenue. Yelp data can also be used to study the local economic outlook. I further maintain that restaurant closures are economically consequential events, the effects of which could be traced to the chain and sector level. I propose that future researchers investigate publicly traded companies' aggregate national exposure to individual location closure risk and study how it changes over time. Such a measure can be used as a risk factor to describe returns. This form of closure risk analysis can further be applied to other hospitality ventures where location success plays a large role, e.g., cinemas, hotels, resorts, and casinos.

This study can further help restaurants redefine what efficient restaurant management means. Corporate efficiency can be loosely defined as how well the resources are managed by a respective organisation. The majority of analysts in the hospitality sector use ratio analysis such as the return on assets/equity, prime cost to total costs, inventory turnover, and others to determine corporate efficiency (Anderson, 2000). I can foresee the use of a new leading measure that identifies the ratio of predicted restaurant location closures to location openings. It has been noted that aggressive growth in this industry can put pressure on a firm's human resources as well as its ability to develop an efficient and effective internal structure (Borde, 1998). For that reason, I consider a ratio between chain closures and openings to be an essential efficiency metric; growth in locations is fine if inefficient locations are undone in the process. Further research can also tease out how this ratio differs among company ownership and franchise ownership models as there is a clear difference in risk appetite. The ability to track individual closures has only become available in recent years, and I urge researchers to pay closer attention to Yelp and its equivalent's more granular data.

This chapter is fundamentally focused on providing good predictions. The truth is that, although it is possible to identify significant predictor variables, these measures are fundamentally derivative in nature and do not express the causal relationships. A score on a review site is only an indication of the restaurant doing something right without the specification of the underlying context. A plausible technique is to organise the good and bad reviews by topic (topic LDA) from which a better indication of possible causes can be sought. Previous studies using similar techniques have, among other things, shown that 5-star rated restaurants have frequent comments related to the cleanliness and friendliness of the staff whereas 1-star rated restaurants have comments more closely related to speed of service and temperature of the food.

Future research can incorporate additional data like health inspection ratings and critical reviews as this could increase the model's precision; it is also possible to tie in rent, menu prices, demographics, psychographics, and location analysis for events or complementary businesses. Lastly, the current model is singularly focused on restaurant closure; a possibly more justifiable economic indicator to track is individual restaurants' profit and revenue, however that would depend on the availability of such data. Extending the analysis in this direction would improve the likelihood of this model being used as a framework to decide whether or not to invest additional resources in a restaurant.

VII. Conclusion

This paper shows that restaurant closures can be predicted with reasonable accuracy using only data and meta-data related to online restaurant reviews. A few surprising variables, not directly attributable to the content of reviews, showed promising predictive power like the number of days since the first review. The model also identified strong interactions between chain and surrounding neighbourhood variables; three chain-interaction variables and three neighbourhood variables rank in the top twenty most predictive variables. In this study, I argue that knowledge of future closures is not just predictive but also prescriptive in nature. The predicted probability of closure gives management an additional metric to help with decision-making. Moreover, a deeper understanding of the non-linear relationship of variables can suggest ways to improve, not just struggling, but also well-run locations.

In summary, the following single-predictor variables are related to a predicted decrease in the likelihood of restaurant closure (i.e. positively related to a restaurant remaining open): 1) higher level of compliments-to-reviewer ratio; 2) lower level of reviews marked as useful by other users (generally critical reviews); 3) increasing star rating over time; 4) restaurants focusing on attracting a specific gender; 5) restaurants attracting more women than men; 6) restaurants positioned in a very low-density area; 7) restaurant chains willing/able to operate in different location profiles – presumably providing a buffer to idiosyncratic risks; however, if the chain is overdiversified the benefit turns negative; 8) restaurants with decreasing negative sentiment; 9) restaurants with higher polarity in sentiment; this measure probably flags untrustworthy reviews, in the sense that when there is not great polarity, the reviews are one-sided and a potential marker of fraudulent reviews; 10) restaurants that have survived in neighbourhoods for longer than surrounding competitors; 11) restaurants that receive more reviews per week than neighbouring peers; and lastly 12) restaurants that receive a high rating from experienced restaurant clientele, i.e. ‘connoisseurs’.

Thesis Conclusion

I provide a model framework that can be used to investigate financial event predictions using machine learning. I investigate the use of machine learning models to predict earnings surprises, corporate defaults, and restaurant facility closures. I show that machine learning can be used to outperform human agents and random choice benchmarks. My work provides a starting point for further research in financial event prediction; researchers could predict initial public offering success, the occurrence of future interest rate adjustments, the outcome of court hearings, changes in unemployment rates, changes in business confidence, changes in credit ratings, and even the occurrence of mergers and acquisitions.

Machine learning is poised to have large economic effects on many financial sectors. We are already witnessing a large portion of asset managers transition to machine learning to find profitable trading opportunities and academia should take up the challenge of providing thought-leadership in the context of the new paradigm enabled by machine learning. For tasks like bankruptcy and earnings surprise prediction, machine learning models substantially outperform their linear counterparts by better modelling the nonlinear nature of financial data. As part of this shift, traditional significance tests of predictor variable performance have largely been replaced by feature importance measures. Advanced machine learning models present numerous advantages in flexibility, efficiency, and most importantly, enhanced prediction quality. Traditional methods in research are now augmented by the power afforded by advanced machine learning tools and I expect this trend to continue.

Bibliography

Abarbanell, J. S., & Lehavy, R. Differences in commercial database reported earnings: Implications for empirical research. *SSRN Electronic Journal,* doi:10.2139/ssrn.228918

Abarbanell, J., & Lehavy, R. (2003). Biased forecasts or biased earnings? the role of reported earnings in explaining apparent bias and over/underreaction in analysts' earnings forecasts. *Journal of Accounting and Economics, 36*(1), 105-146.

Almamy, J., Aston, J., & Ngwa, L. N. (2016). An evaluation of Altman's Z-score using cash flow ratio to predict corporate failure amid the recent financial crisis: Evidence from the UK. *Journal of Corporate Finance, 36*, 278-285.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589-609.

Altman, E. I. (1984). A further empirical investigation of the bankruptcy cost question. *The Journal of Finance, 39*(4), 1067-1089.

Altman, E. I. (2002). *Bankruptcy, credit risk, and high yield junk bonds*. Wiley-Blackwell.

Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). *Journal of Banking & Finance, 18*(3), 505-529.

Anderson, A., Kleinberg, J., & Mullainathan, S. (2017). Assessing human error against a benchmark of perfection. *ACM Transactions on Knowledge Discovery from Data (TKDD), 11*(4), 45.

Anilowski, C., Feng, M., & Skinner, D. J. (2007). Does earnings guidance affect market returns? the nature and information content of aggregate earnings guidance. *Journal of Accounting and Economics, 44*(1), 36-63.

Antonelli, C. (1989). A failure-inducement model of research and development expenditure: Italian evidence from the early 1980s. *Journal of Economic Behavior & Organization, 12*(2), 159-180.

Aziz, A., Emanuel, D. C., & Lawson, G. H. (1988). Bankruptcy prediction-an investigation of cash flow based models. *Journal of Management Studies, 25*(5), 419-437.

Bagheri, A., Peyhani, H. M., & Akbari, M. (2014). Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization. *Expert Systems with Applications, 41*(14), 6235-6250.

Bagnoli, M., Beneish, M. D., & Watts, S. G. (1999). Whisper forecasts of quarterly earnings per share. *Journal of Accounting and Economics, 28*(1), 27-50.

Baird, D. G., & Morrison, E. R. (2011). Dodd-frank for bankruptcy lawyers. *Am.Bankr.Inst.L.Rev., 19*, 287.

Baird, J. (2017). Renaissance technologies: Generating alpha without wall street veterans or MBAs. Retrieved from https://digit.hbs.org/submission/renaissance-technologies-generating-alpha-without-wall-street-veterans-or-mbas/

Ball, R., & Brown, P. (1968). An empirical evaluation of accounting income numbers. *Journal of Accounting Research, 6*(2), 159-178. Retrieved from https://search.proquest.com/docview/1420660825

Barber, B., Lehavy, R., McNichols, M., & Trueman, B. (2001). Can investors profit from the prophets? Security analyst recommendations and stock returns. *The Journal of Finance, 56*(2), 531-563.

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications, 83*, 405-417.

Barniv, R., Agarwal, A., & Leach, R. (2002). Predicting bankruptcy resolution. *Journal of Business Finance & Accounting, 29*(3-4), 497-520.

Barron, O. E., Harris, D. G., & Stanford, M. (2005). Evidence that investors trade on private event-period information around earnings announcements. *The Accounting Review, 80*(2), 403-421.

Barth, M. E., & Hutton, A. P. (2004). Analyst earnings forecast revisions and the pricing of accruals. *Review of Accounting Studies, 9*(1), 59-96.

Bartov, E., Givoly, D., & Hayn, C. (2002). The rewards to meeting or beating earnings expectations. *Journal of Accounting and Economics, 33*(2), 173-204.

Bauer, J., & Agarwal, V. (2014). Are hazard models superior to traditional bankruptcy prediction approaches? A comprehensive test. *Journal of Banking & Finance, 40*, 432-442.

Beaver, W. H. (1968). The information content of annual earnings announcements. *Journal of Accounting Research, 6*, 67-92.

Beaver, W. H., McNichols, M. F., & Rhie, J. (2005). Have financial statements become less informative? evidence from the ability of financial ratios to predict bankruptcy. *Review of Accounting Studies, 10*(1), 93-122.

Behn, B. K., Choi, J., & Kang, T. (2008). Audit quality and properties of analyst earnings forecasts. *The Accounting Review, 83*(2), 327-349.

Behr, A., & Weinblat, J. (2017). Default patterns in seven EU countries: A Random Forest approach. *International Journal of the Economics of Business, 24*(2), 181-222.

Berger, P. G., Ham, C. C., & Kaplan, Z. (2016). Do analysts say anything about earnings without revising their earnings forecasts?

Bergmeir, C., & Bentez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences, 191,* 192-213.

Bernard, V. L., & Thomas, J. K. (1990). Evidence that stock prices do not fully reflect the implications of current earnings for future earnings. *Journal of Accounting and Economics, 13*(4), 305-340.

Bernstein, S., & Sheen, A. (2016). The operational consequences of private equity buyouts: Evidence from the restaurant industry. *The Review of Financial Studies, 29*(9), 2387-2418.

Bhattacharya, N., Sheikh, A., & Thiagarajan, S. R. (2006). Does the market listen to whispers? *The Journal of Investing, 15*(1), 16-24.

Bialik, C. (2017). *Local economic outlook.* Yelp. Retrieved from https://www.yelpblog.com/wp-content/uploads/2017/10/Yelp-Local-Economic-Outlook-Report_October-2017.pdf

Bibler, G. A. (1987). The status of unaccrued tort claims in Chapter 11 bankruptcy proceedings. *Am.Bankr.LJ, 61,* 145.

Booth, A., Gerding, E., & Mcgroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications, 41*(8), 3651-3661.

Bouchaud, J., & Cont, R. (1998). A Langevin approach to stock market fluctuations and crashes. *The European Physical Journal B-Condensed Matter and Complex Systems, 6*(4), 543-550.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7), 1145-1159.

Bradshaw, M. T., Drake, M. S., Myers, J. N., & Myers, L. A. (2012). A re-examination of analysts' superiority over time-series forecasts of annual earnings. *Review of Accounting Studies, 17*(4), 944-968.

Branson, B. C., Lorek, K. S., & Pagach, D. P. (1995). Evidence on the superiority of analysts quarterly earnings forecasts for small capitalization firms. *Decision Sciences, 26*(2), 243-263.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees* CRC press.

Bris, A., Welch, I., & Zhu, N. (2004). The costs of bankruptcy Chapter 7 cash auctions vs. Chapter 11 bargaining.

Bris, A., Welch, I., & Zhu, N. (2006). The costs of bankruptcy: Chapter 7 liquidation versus chapter 11 reorganization. *The Journal of Finance, 61*(3), 1253-1303.

Brown, L. D. (2001). A temporal analysis of earnings surprises: Profits versus losses. *Journal of Accounting Research, 39*(2), 221-241.

Brown, L. D., Han, J. C., Keon Jr, E. F., & Quinn, W. H. (1996). Predicting analysts' earnings surprise. *The Journal of Investing, 5*(1), 17-23.

Burgstahler, D., & Dichev, I. (1997). Earnings management to avoid earnings decreases and losses. *Journal of Accounting and Economics, 24*(1), 99-126.

Burgstahler, D., & Eames, M. (2006). Management of earnings and analysts' forecasts to achieve zero and small positive earnings surprises. *Journal of Business Finance & Accounting, 33*(5-6), 633-652.

Callen, J. L., Kwan, C. C., Yip, P. C., & Yuan, Y. (1996). Neural network forecasting of quarterly accounting earnings. *International Journal of Forecasting, 12*(4), 475-482.

Chan, L. K., Karceski, J., & Lakonishok, J. (2007). Analysts' conflicts of interest and biases in earnings forecasts. *Journal of Financial and Quantitative Analysis, 42*(4), 893-913.

Chandra, D. K., Ravi, V., & Bose, I. (2009). Failure prediction of dotcom companies using hybrid intelligent techniques. *Expert Systems with Applications, 36*(3), 4830-4837.

Chaudhuri, A., & De, K. (2011). Fuzzy support vector machine for bankruptcy prediction. *Applied Soft Computing, 11*(2), 2472-2486.

Chen, J., Chen, W., Huang, C., Huang, S., & Chen, A. (2016). Financial time-series data analysis using deep convolutional neural networks. Paper presented at the *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, 87-92.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Paper presented at the *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785-794.

Chudson, W. A. (1945). The pattern of corporate financial structure: A cross-section view of manufacturing, mining, trade, and construction, 1937. *NBER Books*.

Claus, J., & Thomas, J. (2001). Equity premia as low as three percent? evidence from analysts' earnings forecasts for domestic and international stock markets. *The Journal of Finance, 56*(5), 1629-1666.

Cornelius Casey, & Norman Bartczak. (1985). Using operating cash flow data to predict financial distress: Some extensions. *Journal of Accounting Research, 23*(1), 384-401. doi:10.2307/2490926

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

David R. Anderson, & Kenneth P. Burnham. (2002). Avoiding pitfalls when using information-theoretic methods. *The Journal of Wildlife Management, 66*(3), 912-918. doi:10.2307/3803155

Deligianni, D., & Kotsiantis, S. (2012). Forecasting corporate bankruptcy with an ensemble of classifiers. Paper presented at the *Hellenic Conference on Artificial Intelligence,* 65-72.

Dhar, V., & Chou, D. (2001). A comparison of nonlinear methods for predicting earnings surprises and returns. *IEEE Transactions on Neural Networks, 12*(4), 907-921.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning, 40*(2), 139-157.

DiPietro, R. B., Parsa, H. G., & Gregory, A. (2011). Restaurant QSC inspections and financial performance: An empirical investigation. *International Journal of Contemporary Hospitality Management, 23*(7), 982-999.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87.

du Jardin, P. (2017). Dynamics of firm financial evolution and bankruptcy prediction. *Expert Systems with Applications, 75*, 25-43.

Duffie, D., Pedersen, L. H., & Singleton, K. J. (2003). Modeling sovereign yield spreads: A case study of russian debt. *The Journal of Finance, 58*(1), 119-159.

Easterwood, J. C., & Nutt, S. R. (1999). Inefficiency in analysts' earnings forecasts: Systematic misreaction or systematic optimism? *The Journal of Finance, 54*(5), 1777-1797.

Eliazar, I. (2017). Lindy's law. *Physica A: Statistical Mechanics and its Applications, 486*, 797-805.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology, 77*(4), 802-813.

Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics, 116*(1), 1-22.

Fama, E. F., & Blume, M. E. (1966). Filter rules and stock-market trading. *The Journal of Business, 39*(1), 226-241.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.

Ferri, C., Flach, P., & Hernndez-Orallo, J. (2002). Learning decision trees using the area under the ROC curve. Paper presented at the *Icml, 2*, 139-146.

Fidler, F., Geoff, C., Mark, B., & Neil, T. (2004). Statistical reform in medicine, psychology and ecology. *The Journal of Socio-Economics, 33*(5), 615-630.

FitzPatrick, P. J. (1932). *A comparison of the ratios of successful industrial enterprises with those of failed companies*.

Foster, G. (1973). Stock market reaction to estimates of earnings per share by company officials. *Journal of accounting research, 11*(1), 25-37. Retrieved from http://www.econis.eu/PPNSET?PPN=39273009X

Foster, G. (1986). *Financial statement analysis, 2/e* Pearson Education.

Franks, J. R., & Torous, W. N. (1989). An empirical investigation of US firms in reorganization. *The Journal of Finance, 44*(3), 747-769.

Fried, D., & Givoly, D. (1982). Financial analysts' forecasts of earnings: A better surrogate for market expectations. *Journal of Accounting and Economics, 4*(2), 85-107.

Friedman, J. H., & Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. *Statistics in Medicine, 22*(9), 1365-1381.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* Springer series in statistics New York.

Friedrich, R., Peinke, J., & Renner, C. (2000). How to quantify deterministic and random influences on the statistics of the foreign exchange market. *Physical Review Letters, 84*(22), 5224.

George Foster, Chris Olsen, & Terry Shevlin. (1984). Earnings releases, anomalies, and the behavior of security returns. *The Accounting Review, 59*(4), 574-603. Retrieved from https://www.jstor.org/stable/247321

Givoly, D., & Lakonishok, J. (1984). The quality of analysts' forecasts of earnings. *Financial Analysts Journal, 40*(5), 40-47.

Glaeser, E., Kim, H., & Luca, M. (2017). Nowcasting the Local Economy: Using Yelp Data to Measure Economic Activity (No. 24010). *National Bureau of Economic Research, Inc.*

Gu, S., Kelly, B. T., & Xiu, D.Empirical asset pricing via machine learning. *SSRN Electronic Journal,* doi:10.2139/ssrn.3159577

Graham, B., & Dodd, D. L. (1934). *Security analysis: Principles and technique* McGraw-Hill.

Graham, J. R., Harvey, C. R., & Rajgopal, S. (2005). The economic implications of corporate financial reporting. *Journal of Accounting and Economics, 40*(1), 3-73.

Grossman, S. J., & Zhou, Z. (1993). Optimal investment strategies for controlling drawdowns. *Mathematical Finance, 3*(3), 241-276.

Han, Y., & Zhou, G.Taming momentum crashes: A simple stop-loss strategy. *SSRN Electronic Journal,* doi:10.2139/ssrn.2407199

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning, 77*(1), 103-123.

Hardiman, A. (1985). Toxic torts and chapter 11 reorganization: The problem of future claims. *Vand.L.Rev., 38,* 1369-2013.

Hart, O. (2000). Different approaches to bankruptcy (No. w7921). *National Bureau of Economic Research*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of supervised learning. *The elements of statistical learning* (pp. 9-41). New York: Springer.

Hellstrom, T., & Holmstrom, K. (1998). Predicting the stock market. *Unpublished Thesis, Malardalen University, Department of Mathematics and Physics, Vasteras, Sweden,*

Hillegeist, S. A., Keating, E. K., Cram, D. P., & Lundstedt, K. G. (2004). Assessing the probability of bankruptcy. *Review of Accounting Studies, 9*(1), 5-34.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.

Hu, Y., Feng, B., Zhang, X., Ngai, E., & Liu, M. (2015). Stock trading rule discovery with an evolutionary trend following model. *Expert Systems with Applications, 42*(1), 212-222.

Huang, T., Leu, Y., & Pan, W. (2016). Constructing ZSCORE-based financial crisis warning models using fruit fly optimization algorithm and general regression neural network. *Kybernetes, 45*(4), 650-665.

Hutson, J. K. (1983). TRIX-triple exponential smoothing oscillator. *Technical Analysis of Stocks and Commodities, ,* 105-108.

Jerome H. Friedman. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189-1232. doi:10.2307/2699986

Johnson, W. B., & Zhao, R. (2012). Contrarian share price reactions to earnings surprises. *Journal of Accounting, Auditing & Finance, 27*(2), 236-266.

Jones, C. P., & Litzenberger, R. H. (1970). Quarterly earnings reports and intermediate stock price trends. *The Journal of Finance, 25*(1), 143-148.

Jones, S. (2017). Corporate bankruptcy prediction: A high dimensional analysis. *Review of Accounting Studies, 22*(3), 1366-1422.

Jones, S., & Hensher, D. A. (2004). Predicting firm financial distress: A mixed logit model. *The Accounting Review, 79*(4), 1011-1038.

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

Joret, G., Micek, P., Milans, K. G., Trotter, W. T., Walczak, B., & Wang, R. (2016). Tree-width and dimension. *Combinatorica, 36*(4), 431-450.

Jung, J., Concannon, C., Shroff, R., Goel, S., & Goldstein, D. G. (2017). *Simple rules for complex decisions*

Kaminski, K. M., & Lo, A. W. (2014). *When do stop-loss rules stop losses?* doi://doi-org.ezproxy.auckland.ac.nz/10.1016/j.finmar.2013.07.001

Kang, J. S., Kuznetsova, P., Luca, M., & Choi, Y. (2013). Where not to eat? improving public policy by predicting hygiene inspections using online reviews. Paper presented at the *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1443-1448.

Karas, M., & Reznakova, M. (2017). Predicting the bankruptcy of construction companies: A CART-based model. *Engineering Economics, 28*(2), 145-154. doi:10.5755/j01.ee.28.2.16353

Kasparov, G. (2010). The chess master and the computer. *The New York Review of Books, 57*(2), 16-19.

Kasznik, R., & McNichols, M. F. (2002). Does meeting earnings expectations matter? evidence from analyst forecast revisions and share prices. *Journal of Accounting Research, 40*(3), 727-759.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T. (2017). LightGBM: A highly efficient gradient boosting decision tree. Paper presented at the *Advances in Neural Information Processing Systems*, 3149-3157.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1), 307-319.

Kim, M., & Kang, D. (2010). Ensemble with neural networks for bankruptcy prediction. *Expert Systems with Applications, 37*(4), 3373-3379.

Kim, S. Y., & Upneja, A. (2014). Predicting restaurant financial distress using decision tree and AdaBoosted decision tree models. *Economic Modelling, 36*, 354-362.

Kinney, W., Burgstahler, D., & Martin, R. (2002). Earnings surprise "materiality" as measured by stock returns. *Journal of Accounting Research, 40*(5), 1297-1329.

Kirt C. Butler, & Larry H. P. Lang. (1991). The forecast accuracy of individual analysts: Evidence of systematic optimism and pessimism. *Journal of Accounting Research, 29*(1), 150-156.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., & Mullainathan, S. (2018). Human decisions and machine predictions. *The Quarterly Journal of Economics, 133*(1), 237-293.

Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., & Smith, N. A. (2009). Predicting risk from financial reports with regression. Paper presented at the *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 272-280.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Paper presented at the *Ijcai,, 14*(2) 1137-1145.

Kreeger, J. C., Parsa, H. G., Smith, S. J., & Kubickova, M. (2018). Calendar effect and the role of seasonality in consumer comment behavior: A longitudinal study in the restaurant industry. *Journal of Foodservice Business Research, 21*(3), 342-357.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Paper presented at the *Advances in Neural Information Processing Systems*, 1097-1105.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* Springer.

Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *J Stat Softw, 36*(11), 1-13.

Laney, D. B. (2017). *Infonomics: How to monetize, manage, and measure information as an asset for competitive advantage.* Routledge.

Langevin, P. (1908). Sur la thorie du mouvement brownien. *CR Acad.Sci.Paris, 146*(530-533), 530.

Latane, H. A., & Jones, C. P. (1977). Standardized unexpected earnings—A progress report. *The Journal of Finance, 32*(5), 1457-1465.

Lawrence D. Brown, Gordon D. Richardson, & Steven J. Schwager. (1987). An information interpretation of financial analyst superiority in forecasting earnings. *Journal of Accounting Research, 25*(1), 49-67. doi:10.2307/2491258

Leathwick, J. R., Elith, J., Francis, M. P., Hastie, T., & Taylor, P. (2006). Variation in demersal fish species richness in the oceans surrounding New Zealand: An analysis using boosted regression trees. *Marine Ecology Progress Series, 321*, 267-281.

Lhabitant, F. (2011). *Handbook of hedge funds* John Wiley & Sons.

Li, J. (2012). Prediction of corporate bankruptcy from 2008 through 2011. *Journal of Accounting and Finance, 12*(1), 31-41.

Li, K. (1999). Bayesian analysis of duration models: An application to chapter 11 bankruptcy. *Economics Letters, 63*(3), 305-312.

Lian, J., Zhang, F., Xie, X., & Sun, G. (2017). Restaurant survival analysis with heterogeneous information. Paper presented at the *Proceedings of the 26th International Conference on World Wide Web Companion*, 993-1002.

Liang, D., Lu, C., Tsai, C., & Shih, G. (2016). Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research, 252*(2), 561-572.

Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* Springer Science & Business Media.

LoPucki, L. M., & Doherty, J. W. (2008). Professional overcharging in large bankruptcy reorganization cases. *Journal of Empirical Legal Studies, 5*(4), 983-1017.

LoPucki, L. M., & Doherty, J. W. (2015). Bankruptcy survival. *UCLA L.Rev., 62*, 969.

LoPucki, L. M., & Kalin, S. D. (2001). Failure of public company bankruptcies in Delaware and New York: Empirical evidence of a race to the bottom. *Vand.L.Rev., 54*, 231.

Luca, M. (2011). *Reviews, reputation, and revenue: The case of yelp.com.* Retrieved from http://econpapers.repec.org/paper/hbswpaper/12-016.htm

Luca, M., & Zervas, G. (2016). Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science, 62*(12), 3412-3427.

Lundberg, S. M., & Lee, S. (2017). A unified approach to interpreting model predictions. Paper presented at the *Advances in Neural Information Processing Systems*, 4768-4777.

Luo, T., & Stark, P. B. (2015). Nine out of 10 restaurants fail? check, please. *Significance, 12*(2), 25-29.

Lys, T., & Sohn, S. (1990). The association between revisions of financial analysts' earnings forecasts and security-price changes. *Journal of Accounting and Economics, 13*(4), 341-363.

Lys, T., & Soo, L. G. (1995). Analysts' forecast precision as a response to competition. *Journal of Accounting, Auditing & Finance, 10*(4), 751-765.

MacGregor, N., & Lo, D. (2015). The effect of chain size and customer type on company survival: Evidence from multi-unit restaurants. *Academy of Management Proceedings, 2015*(1), 17991.

MacKay, D. J. (1992). The evidence framework applied to classification networks. *Neural Computation, 4*(5), 720-736.

Matras, K. (2016). Zacks investment research - video: Zacks earnings ESP (expected surprise prediction). Retrieved from https://search.proquest.com/docview/1838754510

Mejia, J., Mankad, S., & Gopal, A. (2015). More than just words: Latent semantic analysis, online reviews and restaurant closure. *Academy of Management Proceedings, 2015*(1), 13912. doi:10.5465/AMBPP.2015.13912abstract

Merwin, C. L. 1. (1942). *Financing small corporations in five manufacturing industries, 1926-36*. United States: Retrieved from http://catalog.hathitrust.org/Record/001128380

Meyer, P. A. (1967). Price discrimination, regional loan rates, and the structure of the banking industry. *The Journal of Finance, 22*(1), 37-48.

Mian, S. (2013). *Zacks earnings ESP*. Zacks. Retrieved from https://staticzacks.net/pdf/EESPreport1V3.pdf

Michael J. Gombola, & J. Edward Ketz. (1983). A note on cash flow and classification patterns of financial ratios. *The Accounting Review, 58*(1), 105-114. Retrieved from https://www.jstor.org/stable/246645

Montas, E., Quevedo, J. R., Prieto, M. M., & Menndez, C. O. (2002). Forecasting time series combining machine learning and Box-Jenkins time series. Paper presented at the *Ibero-American Conference on Artificial Intelligence*, 491-499.

Mselmi, N., Lahiani, A., & Hamza, T. (2017). Financial distress prediction: The case of French small and medium-sized firms. *International Review of Financial Analysis, 50*, 67-80.

Mukherjee, S., Golland, P., & Panchenko, D. (2003). Permutation tests for classification, 11-21

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives, 31*(2), 87-106.

National Restaurant Association. (2017). *Restaurant industry outlook*. Retrieved from http://www.restaurant.org/Downloads/PDFs/News-Research/2017_Restaurant_outlook_summary-FINAL.pdf

Neves, J. C., & Vieira, A. (2006). Improving bankruptcy prediction with hidden layer learning vector quantization. *European Accounting Review, 15*(2), 253-271.

O'Brien, P. C. (1998). Discussion of international variation in accounting measurement rules and analysts' earnings forecast errors. *Journal of Business Finance & Accounting, 25*(9-10), 1249-1254.

Ohlson, J. A. (2009). Financial ratios and the probabilistic prediction of bankruptcy. *Financial Accounting and Investment Management, 18(1),* 363-385. Retrieved from http://www.econis.eu/PPNSET?PPN=603823629

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

Parsa, H. G., Self, J. T., Njite, D., & King, T. (2005). Why restaurants fail. *Cornell Hotel and Restaurant Administration Quarterly, 46*(3), 304-322.

Parsa, H. G., Self, J., Sydnor-Busso, S., & Yoon, H. J. (2011). Why restaurants fail? part II-the impact of affiliation, location, and size on restaurant failures: Results from a survival analysis. *Journal of Foodservice Business Research, 14*(4), 360-379.

Parsa, H. G., van der Rest, Jean-Pierre I, Smith, S. R., Parsa, R. A., & Bujisic, M. (2015). Why restaurants fail? part IV: The relationship between restaurant failures and demographic factors. *Cornell Hospitality Quarterly, 56*(1), 80-90.

Patel, C. (1980). *Technical trading systems for commodities and stocks.* Trading Systems Research.

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*(1), 259-268.

Payne, J. L., & Thomas, W. B. (2003). The implications of using stock-split adjusted I/B/E/S data in empirical research. *The Accounting Review, 78*(4), 1049-1067.

Pervan, I., Pervan, M., & Vukoja, B. (2011). Prediction of company bankruptcy using statistical techniques–Case of croatia. *Croatian Operational Research Review, 2*(1), 158-167.

Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies, 2*, 37-63.

Provost, F., & Kohavi, R. (1998). Glossary of terms. *Journal of Machine Learning, 30*(2-3), 271-274.

Puga, D. (1999). The rise and fall of regional inequalities. *European Economic Review, 43*(2), 303-334.

Purdy, D., Chen, L., & Sumers, T. R. (2017). *Cascaded Boosted Predictive Models.*

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81-106.

Raileanu, L. E., & Stoffel, K. (2004). Theoretical comparison between the Gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence, 41*(1), 77-93.

Ramnath, S., Rock, S., & Shane, P. (2008). The financial analyst forecasting literature: A taxonomy with suggestions for further research. *International Journal of Forecasting, 24*(1), 34-75.

Ramsey, J. B., & Zhang, Z. (1997). The analysis of foreign exchange data using waveform dictionaries. *Journal of Empirical Finance, 4*(4), 341-372.

Robert N. Freeman, & Senyo Y. Tse. (1992). A nonlinear model of security price responses to unexpected earnings. *Journal of Accounting Research, 30*(2), 185-209. doi:10.2307/2491123

Rose, P. S., & Giroux, G. A. (1984). Predicting corporate bankruptcy: An analytical and empirical evaluation. *Review of Financial Economics, 19*(2), 1.

Sable, R. G., Roeschenthaler, M. J., & Blanks, D. F. (2006). When the 363 sale is the best route. *J.Bankr.L.& Prac., 15*, 2.

Schaaf, A. H. (1966). Regional differences in mortgage financing costs. *The Journal of Finance, 21*(1), 85-94.

Schapire, R. E., & Freund, Y. (2012). *Boosting: Foundations and algorithms* MIT press.

Scott, J. (1981). The probability of bankruptcy: A comparison of empirical predictions and theoretical models. *Journal of Banking & Finance, 5*(3), 317-344.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN variables off-the-shelf: An astounding baseline for recognition. Paper presented at the *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 806-813.

Shumway, T. (2001). Forecasting bankruptcy more accurately: A simple hazard model. *The Journal of Business, 74*(1), 101-124.

Siegert, S., Friedrich, R., & Peinke, J. (1998). Analysis of data sets of stochastic systems. *Physics Letters A, 243*(5-6), 275-280.

Soffer, L. C., Thiagarajan, S. R., & Walther, B. R. (2000). Earnings preannouncement strategies. *Review of Accounting Studies, 5*(1), 5-26.

Somnath Das, Carolyn B. Levine, & K. Sivaramakrishnan. (1998). Earnings predictability and bias in analysts' earnings forecasts. *The Accounting Review, 73*(2), 277-294. Retrieved from https://www.jstor.org/stable/248469

Stauth, J. (2013). Trading (and predicting) earnings surprises. *Thomson Reuters*. Retrieved from https://blog.quantopian.com/wp-content/uploads/2013/05/Trading_Earnings_surprises-jess.pdf

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment, 62*(1), 77-89.

Stickel, S. E. (1995). The anatomy of the performance of buy and sell recommendations. *Financial Analysts Journal, 51*(5), 25-39.

Sun, J., & Li, H. (2012). Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing, 12*(8), 2254-2265.

Tan, C., Lee, L., & Pang, B. (2014). The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. *arXiv Preprint arXiv:1405.1438,*

Taylor, D. C., & Aday, J. B. (2016). Consumer generated restaurant ratings: A preliminary look at OpenTable.com. *Journal of New Business Ideas and Trends, 14*(1), 14-23.

Teixeira, L. A., & De Oliveira, Adriano Lorena Inacio. (2010). A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications, 37*(10), 6885-6890.

Trueman, B. (1990). Theories of earnings-announcement timing. *Journal of Accounting and Economics, 13*(3), 285-301.

Volkov, A., Benoit, D. F., & Van den Poel, D. (2017). Incorporating sequential information in bankruptcy prediction with predictors based on markov for discrimination. *Decision Support Systems, 98*, 59-68.

W. S. Hopwood, J. C. Mckeown, & P. Newbold. (1982). The additional information content of quarterly earnings reports: Intertemporal disaggregation. *Journal of Accounting Research, 20*(2), 343-349. doi:10.2307/2490744

Waymire, G. (1986). Additional evidence on the accuracy of analyst forecasts before and after voluntary management earnings forecasts. *Accounting Review, 61*, 129-142.

Weiss, L. A., Bhandari, J. S., & Robins, R. (2000). An analysis of state-wide variation in bankruptcy rates in the united states. *Bankr.Dev.J., 17*, 407.

Whittingham, M. J., Stephens, P. A., Bradbury, R. B., & Freckleton, R. P. (2006). Why do we still use stepwise modelling in ecology and behaviour? *Journal of Animal Ecology, 75*(5), 1182-1189.

William A. Collins, & William S. Hopwood. (1980). A multivariate analysis of annual earnings forecasts generated from quarterly forecasts of financial analysts and univariate time-series models. *Journal of Accounting Research, 18*(2), 390-406. doi:10.2307/2490585

William H. Beaver. (1966). Financial ratios as predictors of failure. *Journal of Accounting Research, 4*, 71-111. doi:10.2307/2490171

William Kross, Byung Ro, & Douglas Schroeder. (1990). Earnings expectations: The analysts' information advantage. *The Accounting Review, 65*(2), 461-476. Retrieved from https://www.jstor.org/stable/247634

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques, *Morgan Kaufmann*.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67-82.

Womack, K. L. (1996). Do brokerage analysts' recommendations have investment value? *The Journal of Finance, 51*(1), 137-167.

Xiao, Y., Xiao, J., Lu, F., & Wang, S. (2013). Ensemble ANNs-PSO-GA approach for day-ahead stock E-exchange prices forecasting. *International Journal of Computational Intelligence Systems, 6*(1), 96-114.

Youn, H., & Gu, Z. (2010). Predict US restaurant firm failures: The artificial neural network model versus logistic regression model. *Tourism and Hospitality Research, 10*(3), 171-187.

Zhang, M., & Luo, L.Can user generated content predict restaurant survival: Deep learning of yelp photos and reviews. *SSRN Electronic Journal*, doi:10.2139/ssrn.3108288

Zhao, D., Huang, C., Wei, Y., Yu, F., Wang, M., & Chen, H. (2017). An effective computational model for bankruptcy prediction using kernel extreme learning machine approach. *Computational Economics, 49*(2), 325-341.

Zhou, L. (2013). Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Systems, 41,* 16-25.

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic variables generation in application to bankruptcy prediction. *Expert Systems with Applications, 58,* 93-101.

Zmijewski, M. E. (1984). Methodological issues related to the estimation of financial distress prediction models. *Journal of Accounting Research, 22,* 59-82.