Estimating Credit Risk Premiums with Gradient Boosted Regression Trees and Neural Networks

René Alby

MSc Computer Science

The University of Bath

May 13, 2023

1. Abstract

An axiom in the world of finance is the concept of the risk-return trade-off. That is, a risky investment should offer a higher potential return than a less risky investment. This trade-off is referred to as the risk premium. Machine learning may be an excellent tool for estimating risk premiums. We test this by performing a comparative analysis of neural networks and gradient boosted regression trees to assess how well these models can estimate risk premiums on corporate bonds. We define the risk premium as the Z-spread on vanilla corporate bonds. We selected 22 estimators ranging from financial ratios, stock market performance and volatility data, economic data, and bond idiosyncratic data. We collected a data set of 22,161 rows spanning the 20-year period from January 2003 to January 2023 across 197 corporate bonds and 71 bond issuers. We then split this into 80% model training data and 20% testing data. We found that both models have significant predictive capability to estimate credit risk premiums on the test set. The gradient boosted regression tree model predictions resulted in an adjusted R^2 of 0.95, outperforming the neural network model with an adjusted R^2 of 0.823. We further assess the resultant models by decomposing a selection of individual predictions. We find that both models have learned sensible interpretations of the risks implied by the financial data.

Table of Contents

2. Introduction

Economic theory dictates that a rational investor should demand a higher return on a risky investment compared to an investment with a lower level of perceived risk, a concept known as the risk-return trade-off. While some exceptions to this rule have been observed, this is generally a widely accepted phenomenon in financial markets (Ghysels, Santa-Clara and Valkanov, 2005). For example, the interest earned on a bank account is extremely low because the depositor has access to the funds immediately and the deposits are guaranteed by government-backed insurance providers. The interest earned on a corporate bond, however, would generally be significantly higher than interest on a bank account. This is because there is a risk that the corporation may find itself unable to pay back the bond and because unlike bank deposits, bonds cannot easily be converted into cash.

The difference in return between an investment that is risk-free versus an investment that is risky is referred to as the risk premium. For corporate bonds, there are various sources of risk that contribute to the risk premium. Macroeconomic factors such as inflation and central bank interest rates, as well as microeconomic factors such as the risk of corporate default. In his seminal 1959 study on risk premiums, Lawrence Fisher identified the following factors that contribute to microeconomic risk premiums: earnings variability, period of solvency, equity/debt ratio, bonds outstanding, volume of trading and ratio of standard deviation in earnings to equity (Fisher, 1959). These variables capture two sources of risk: the risk that a corporate borrow will not be able to repay their loans, known as credit risk, and the risk that an investor will not be able to sell or redeem a bond for cash easily if needed, known as liquidity risk. Fisher found that approximately 75% of the risk premium on corporate bonds could be estimated using a simple linear regression model on the variables mentioned above.

Risk premiums are fundamental to pricing or valuing investments. The role of an investment analyst is to estimate the risk premium for any given investment. However, there is no model that can perfectly estimate the appropriate risk premium for a particular investment. The unpredictability of economic outcomes ensures that estimating risk premiums remains as much an art as it is a science. Due to the empirical nature of investment valuation, machine learning algorithms may lend themselves nicely to the problem of estimating risk premiums, as there are vast amounts of financial data readily available and numerous predictors that can be fed into a machine learning model.

2.1 Motivation

Much of the literature investigating machine learning for investment valuation has focused on estimating stock returns or picking stock outperformers. As far back as 1993, Kryzanowsi et al. showed how a simple neural network trained on company fundamental data and macroeconomic data can perform well when selecting stocks that will outperform (Kryzanowski, Galler and Wright, 1993). More recently, machine learning has been applied to the problem of valuation and determining equity risk premiums (i.e., the risk premiums on stocks/shares). Gu et al. found that machine learning models significantly outperformed traditional ordinary least squares regression models in estimating equity risk premiums using several fundamental valuation metrics (Gu, Kelly and Xiu, 2018). They attributed this outperformance to the machine learning models' ability to handle nonlinear interactions between the predictive variables. Comparatively little research has been conducted on the application of machine learning in the corporate bond realm.

This study seeks to expand upon the work of Gu, Kelly and Xiu in empirical asset valuation, except with a focus on corporate bond risk premiums rather than equity risk premiums. To investigate this, an Artificial Neural Network model and a Gradient Boosted Regression Tree model will be trained using fundamental valuation metrics such as corporate leverage and solvency ratios, bond terms and conditions data, and observable market data including yields on short-term and long-term treasury bonds. The goal of this study is to assess how well machine learning models can predict risk premiums on corporate bonds as measured using Z-spreads for vanilla corporate bonds. In financial industry jargon, this is referred to as the credit spread.

Z-spreads on vanilla corporate bonds were selected because they are widely observable measures of the credit, liquidity, and payment timing risk of a bond. Vanilla corporate bonds refer to bonds without any embedded options. Embedded options may have potentially distorting effects on a bond's risk profile and price. A significant amount of technical adjustment would be required to compensate for this. By sticking to vanilla bonds, we avoid these pitfalls.

The application of a machine learning model to estimate credit spreads on corporate bonds could be useful for asset managers for two reasons. First, if the model is found to be accurate in estimating appropriate risk premiums, it could potentially be applied to assess relative value and to identify overpriced or underpriced bonds in the market. Second, it can serve as a tool to assess new bond issues that do not yet have observable market prices.

Much of the research in estimating credit risk employs market-focused unsystematic risk variables such as the volatility of a bond issuer's stock price, with many derivatives of this methodology referred to eponymously as Merton Models (Merton, 1973). However, many bond issuers are private corporations that do not have publicly listed stocks and therefore no stock price volatility to measure. This means that these models cannot be applied to a segment of the outstanding universe of corporate bonds. This research applies a model that is valid for the entire universe of outstanding bonds, employing only systematic market risk measures such as general market volatility or interest rates.

2.2 Research questions

This research seeks to answer the following questions:

1. How accurately can machine learning algorithms estimate credit risk premiums when trained on market data, financial ratios, and other traditional credit analysis data?
2. Which machine learning models perform better at this task?
3. What features of the training data do machine learning algorithms learn, and are these sensible in the context of traditional credit analysis?

3. Literature review

The topic of this research intersects the fields of financial economics and computer science, specifically machine learning and artificial intelligence. This review will investigate the literature around credit risk premiums, how they are estimated and how they are measured. Then, it delves into the topic of machine learning in financial applications and the state of the art around machine learning in credit analysis.

3.1 Credit analysis and predicting bankruptcy

Arguably the first published investigation into the factors that contribute to bond risk premiums began with the seminal work of Lawrence Fisher (Fisher, 1959), who postulated that bond risk premiums, as measured by the yield on a bond less the yield on the pure rate of interest, depend on whether the firm will default and the bond marketability and that this risk premium can be estimated using a multiple linear regression (ordinary least squares) of the logarithm of four variables: 1. The variance of the firm's earnings over 9 years. 2. the length of time the firm has been operating without any default. 3. the market value of the firm's equity relative to the size of its debt. 4. The market value of all publicly traded bonds the firm has outstanding, which fisher uses as a proxy for marketability. Fisher found that this model accounted for 81% of the variance in the log of risk premiums.

William Beaver posited that the use of accounting ratios could be useful as predictors of firm failure (Beaver, 1966). Accounting ratios standardise the financial results of companies, allowing for performance comparisons to be made. To test this hypothesis, Beaver took a sample of failed and nonfailed firms from Moody's Industrial Manual, matching the firms, and split them by their industry type using Standard Industry Classification (SIC) codes and asset size as measured from their most recent financial statement before failure. The data were aligned to ensure that failed and nonfailed firms were in each industry and were similar in asset size in a pairwise fashion. Beaver assessed 30 financial ratios related to firm cashflow, net income, debt levels, liquidity, and inventory turnover and selected one from each: Cash Flow to Total Debt, Net Income to Total Assets, Total Debt to Total Assets, Working Capital to Total Assets, Current Ratio and No Credit Interval. Through comparisons of these ratios between failed and nonfailed firms, Beaver identified clear patterns that could be used as predictors of firm failure.

Edward Altman formalised the use of financial ratios for credit risk assessment using multiple discriminant analysis (MDA) in what is now eponymously known as the Altman Z Score (Altman, 1968). Similar to Beaver's research design, Altman took a split of failed and nonfailed firms. Altman assessed 22 financial ratios using correlation analysis to weed out ratios that did not add additional information to the model. Altman landed on a selection of 5 ratios: Working Capital/Total Assets, Retained Earnings/Total Assets, EBIT/Total Assets, Market Value of Equity/Book Value of Total Debt, and Sales/Total Assets. Altman's MDA model correctly classified 95 percent of the total initial sample of 33 firms; however, the model's predictive capability decreases significantly as the time to bankruptcy increases. Edward Deakin combined Altman and Beaver's models using an MDA model applied to Beaver's set of financial ratios and achieved high predictive capability over a longer time to bankruptcy measures by calculating the discriminant analysis in each of the five years to bankruptcy (Deakin, 1972).

James Ohlson (1980) extended the research, both by using much larger sample sizes than his predecessors, 2163 firms, and by employing logistic regression, which gives a probability of firm failure as opposed to the binary outcome of MDA. Ohlson finds that the statistically significant factors that can be used to determine the probability of failure are firm size, measured as total assets/GNP price-level index (i.e., adjusted for inflation), Total Liabilities/Total Assets, OENEG which is a dummy variable indicating whether total liabilities exceeds total assets and zero otherwise, Net income/Total Assets, Funds from operations/total liabilities, Change in net income measures as $NI_{t0} – NI_{t-1}/ABS(NI_{t0}) + ABS(NI_{t-1})$

Zmijewski (1984) added to the accountancy-based probabilistic models of bankruptcy by addressing some of the potential biases in bankruptcy data sets and by introducing a model based on the probit function. We use all listed companies on the NYSE from 1972-78, totalling thousands. The model results in a score that classifies firms above 0.5 as bankrupt and vice versa. The accuracy of his model was measured at above 95%.

Wu, Gaunt, Grey (2010) compare various bankruptcy probability models: Altman’s MDA method, Ohlson’s Logit method, Zmijewski’s Probit model, as well as the Shumway Hazard model and A model based on Black-Scholes option pricing. The research indicates that the Altman MDA model did not perform as well as the probit and logit models, which in turn worked well on older data sets but not as well on more modern data sets. The Hazard model had a higher predictive capability than the others, and this was attributed to its use of nonsystematic market data (i.e., the company share price). As noted earlier, this type of model is only useful for companies with shares listed on the stock market, and therefore, it is out of scope for the research conducted in this dissertation. The same is true for Black-Scholes models. Wu et al. also construct a new model consisting of key elements from the other 5 models assessed and find that it has a higher predictive capability.

While it has generally been considered axiomatic that investors will demand a premium on bonds that increases with risk of bankruptcy, a model for determining whether the bond risk premium actually exists was only proposed in 1999 (Elton, Gruber, Agarwal, Mann, 1999). They hypothesised that the risk premium (i.e., the ‘credit spread’) could be related to different tax treatments or other nonrisk-related variables. The model worked by calculating spot rates across the corporate bond spot curve, effectively the yields at various cashflow payments on a bond. The model then priced the bond on a risk-neutral model. By additively factoring tax and liquidity differentials, any remaining differential can only be explained by a risk premium. The result is that they found that a substantial risk premium exists in corporate bonds.

Much of the literature investigating credit spreads looks broadly at economic factors that impact general (i.e., systematic) levels of spreads for specific credit ratings categories. Davies found that credit spreads on AAA and BAA Moody’s rated bonds were positively correlated with the S&P 500 index. (Davies, 2008). Davies also found that credit spreads are inversely related to the risk-free rate; however, this result is considered more controversial (see Li, X.-L., Li, X., and Si, D.-K., 2020 for a summary of contradicting evidence). Another finding from Davie’s econometric model was that differing inflation regimes impacted high grade and low-grade credit in differing ways, with the latter more affected by high inflation regimes. Similarly, spreads are found to be inversely related to periods of economic expansion and periods of economic recession (Kao, 2000). Employing the multifactor Vasicek model, Bhar and Handzic (2010) find that the systematic variability in US credit spreads can be mostly explained by 3 factors: 1. the stock market volatility as measured by the volatility index (VIX), 2. the stock market return as measured by the S&P 500 index, and 3. the so called ‘long bond rate’, i.e., the yield to maturity on the 10-year benchmark government bond.

In summary, there are numerous firm-level and macroeconomic variables that are relevant to bond risk premiums and corporate bond valuation. However, there does not appear to be a single valuation model that combines all these variables to estimate the expected risk premium on a corporate bond.

3.2 Machine learning applications to financial analysis

A great deal of research has been conducted in the field of machine learning applications to financial problems. This is unsurprising given that financial analysis and machine learning are both data-driven processes. The majority of the research focuses on stock market prediction and trading. For a review of the literature in this field, see Henrique, Sobreiro, Kimura (2019). To a lesser extent, machine learning applications to financial analysis have investigated foreign exchange rate prediction, commodity price prediction, macroeconomic prediction, portfolio management and credit risk. For a summary of the literature in these fields of research, see Huang, Chai, and Cho (2020).

Investigations into the use of artificial neural networks for stock selection can be traced back to the early nineties when Wong et al. used a set of 32 company financial variables while applying a layer of rules based on 'expert knowledge' (i.e., an 'expert system') to the data (Wong, Wang, Goh, 1992). After the expert rules were applied, the data were input into an artificial neural network for predictions about stock prices for 24 stocks. Their model was highly accurate in predicting the direction of stock price movement and, to a lesser extent, the actual percent performance. The name they gave to their model was FuzzNet for fuzzy networks.

Swales and Yoon (1992) compared the performance of multiple discriminant analysis (MDA) to that of an artificial neural network (ANN) on two samples of 58 companies and 40 companies from Fortune 500 and BusinessWeek, respectively. For each group, the study analysed the content of the president's letter to shareholders, identifying themes of economic importance. The researchers used the frequency of theme and the percentage of the letter that was dedicated to the theme to feed both their MDA and ANN to produce a binary output of 'good' vs 'bad' as relates to the company share price performance. The ANN model without any hidden layers underperformed the MDA model at approximately a 52% success rate, but with additional hidden layers, it outperformed the MDA model, with a 77% success rate compared to a 65% success rate. They concluded that nonlinear relationships between predictors contributed valuable information to the model.

Kryanowski et al. (1993) applied an ANN to predict stock price returns on 120 companies over a five-year period from 1984-89. They trained the model on the stock return 1 year in the future, using their most recent 4 years trailing financial data, calculating ratios similar to Altman and Deakin, and trained on a set of seven macroeconomic indicators, which include industrial production, gross domestic product, the McLeod Young Weir corporate long bond index, the 90-day treasury bill rate, the government of Canada long bond index, the consumer price index and the Montreal Exchange 25 index. Due to the small sample size, they directly encoded feature data into the model rather than attempting to obtain the model to generalise the features. To do this, they encoded 3-bit combinations that translated to trends in the data, uptrend, downtrend or stable. Their tests achieved a 66.4% overall accuracy in predicting whether a stock's return would be positive or negative in the following year.

More recently, Gu et al. (2018) compared the success of various machine learning algorithms against a large universe of stocks and predictors to measure equity risk premiums. Their sample included nearly 30,000 individual stocks over a 60-year period, employing over 900 variables based on 94 predictors for each stock, 8 predictors calculated as interactions with the 94 independent variables, and 74 dummy variables reflecting industry sector. Employing such a large set of independent variables is novel to the economic literature, which largely focuses on linear models that cannot consume so many inputs. The research is also novel in that they investigate and compare which predictors are most influential amongst the different models employed. They find that the models are mostly in agreement about the set of predictors found to be the strongest. This adds to the robustness of the findings. Finally, they demonstrate significant economic gain in the use of machine learning algorithms compared to liner models in stock selection.

While the above focuses on investment valuation in equity markets, relatively less research has investigated machine learning applications in the bond market.

Bianchi, Buchner, and Temoni (2019) investigated boosted regression trees, random forests, extremely random randomised regression trees and shallow and deep neural networks to predict bond risk premiums, focusing narrowly on the treasury market, first by using yield curves as predictors and then by adding several time series of macroeconomic variables to the model. They found that neural networks performed well in the yield-only case and even better when employing macroeconomic variables. The following macroeconomic variables were studied: real output and income, employment and hours, real retail, manufacturing and sales data, international trade, consumer spending, housing starts, inventories and inventory sales ratios, orders and unfilled orders, compensation and labour costs, capacity utilisation, price indices, interest rates and rate spreads, stock market indicators and foreign exchange rates.

Credit ratings are an area where machine learning has enjoyed a significant amount of focus and success. This is not surprising given the categorical nature of credit ratings, which is a problem well suited to machine algorithms. For a review of the literature in this field, see Golbayani, Florescu and Chatterjee (2020). Golbayani et al. also contributed to the literature with a comparative study employing bagged decision trees, support vector machines and multilayer perceptrons and found that decision trees had the best performance.

Credit spreads (i.e., risk premiums on corporate bonds) are an area that has received relatively little attention in the field of machine learning. Kim, Kim and Jung (2021) compared various machine learning methods against linear regression models to predict 1-year and 2-year corporate bond yield spreads. Yields were calculated by taking the volume-weighted average price of trades from the American TRACE reporting system. The independent variables they chose were credit ratings, interest rate volatility, equity index volatility, bond maturity and coupon rate, and liquidity (measured as number of days bond traded in past 12 months). The models they employed were ridge, multivariate adaptive regression splines, neural networks, random forest and support vector machines. They found neural networks to outperform all other methods.

Liu, Zhou and Xiong (2021) proposed a novel depth-gated recurrent neural network with a self-attention mechanism for the prediction of credit spreads. Their research investigated the Long-Short-Term memory (LSTM) neural network architecture, the Gated Recurrent Unit (GRU) Neural Network architecture, and the Just Another NETwork (JANET) neural network architecture compared against linear models. They also implemented a self-attention mechanism that dynamically generates weights for different connections within neural network layers. The adaptive momentum estimation (Adam) optimiser and extreme gradient boosting (XGBoost) were employed for training, and the mean squared errors were selected as a loss function. Predictors included the Risk-Free interest rate, credit spread term structure, stock index return, stock index volatility, exchange rates, Oil Prices, TED Spread, swap spread and the commodity price index. Credit spreads in this study were taken on aggregate by using ICE BofAML credit indices. Their findings suggest that the self-attention mechanism can improve results in neural network architectures.

4. Model selection

The scope of this research is to see whether we can arrive at a reasonable valuation for a given corporate bond as measured by credit spreads by using machine learning algorithms. This research is not meant to predict future spreads based on a time series of current spreads and other data. Therefore, recurrent neural network models were deemed out scope. Instead, based on the review of the literature, two models were selected as being highly performant for financial applications.

1. Artificial Neural Network model (aka multilayer perceptron)
2. Gradient Boosted Regression Tree model

4.1 Artificial neural network model

An artificial neural network model is composed of layers of interconnected perceptrons, as depicted in Figure 1 below. Perceptrons are a form of binary classifier composed of an input variable (X), a weight (W) and a bias/intercept (t).

$$W_1X_1 + W_2X_2 + b = 0$$

In an artificial neural network, perceptrons are layered and transformed in such a way that the output of one perceptron passes through a nonlinear ‘activation function’ and is fed to all the perceptrons in the next layer. This is why the model is also referred to as a multilayer perceptron model. The unit whereby the output of a perceptron is fed through an activation function is referred to as a neuron or a node in the artificial neural network model.

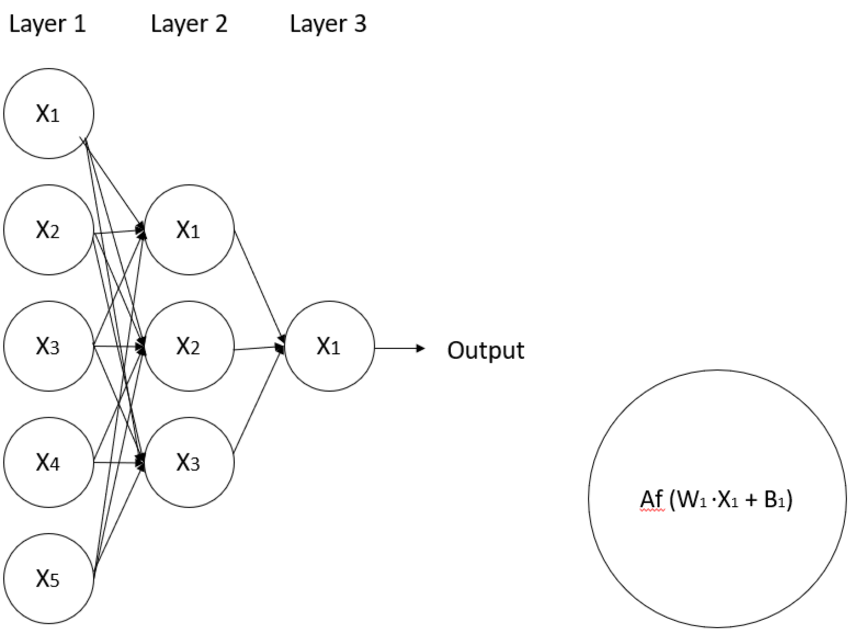


Figure 1: Diagram of a simple artificial neural network (left) and a single node in the network (right).

where each node is composed of a weight (W) multiplied by the input data (X) plus a bias value (B), and then the result is passed through the activation function (Af)

At the final layer, a single perceptron outputs a value that could be a classifier of some sort, a probability, or, for a regression problem, a single estimated value.

All artificial neural networks are composed of the following:

- **Input layer**: A layer that receives the set of estimators or X-Variables for the model (also referred to as Features).
- **Hidden layers**: Stacked layers of nodes where the output of each node is fed to all nodes of the next layer
- **Output Layer**: A single node that outputs the result of the calculation.
- **Activation Function**: Usually, either a rectified linear unit (ReLU) function, a logistic function, or a linear function. The output of each node is fed through this function, which, in the hidden network layers, serves to transform a linear function into a nonlinear function, and at the output layer, distinguishes the model from a regression, probabilistic, or categorical problem.
- **Weights and Biases**: Each note has a weight between 0-1, which allows the model to be adjusted so that a single node feeds more or less data to the following nodes. The bias is added to the weighted value prior to the activation function being applied.
- **Loss function**: A function that measures the output or prediction of the neural network against the actual expected value. The purpose of the model is to minimise the loss for any single estimate produced by the neural network. Therefore, the artificial neural network is ultimately a complex optimisation problem.
- **Backpropagation** refers to the process of feeding the calculated loss backwards through the neural network, at which point the weights of each node are adjusted using differential equations so that the model will produce a closer estimate to the actual value given the same inputs. This is ultimately the model's learning process.
- **Optimisation algorithms**: As previously described, the backpropagation algorithm relies on differential equations (or gradient descent) to optimise the weights of the model. There are various algorithms that can improve this learning process. One is ADAM (Adaptive Momentum Estimation), which smooths the data by applying moving averages, ultimately facilitating the process of gradient descent.

The implementation of the neural network algorithm for this research was performed using TensorFlow 2.10.0. TensorFlow was selected due to its widespread adoption in the financial services industry and because it is open source.

For more detail on this model, refer to Appendix II.

4.2 Gradient boosted regression tree model

A gradient boosted regression tree model combines multiple decision trees to arrive at a predicted value in a regression or classification problem. Decision trees work by taking input data and repeatedly splitting it into subsets based on the values of each different feature (X-Variable) until it arrives at a 'leaf' node, referring to a final value of data or the final prediction. Each split attempts to minimise an impurity function, usually an entropy function for classification problems or statistical variance for a regression problem. Once several trees are created, predictions can be made by taking the average value produced by many trees (Ng, Bagul, Shyu, Ladwig, 2023).

The gradient boosted regression tree model enhances the random forest model by applying a gradient boosting algorithm. The model creates new trees by fitting the trees to the negative gradient of the loss function (or objective function), therefore optimising each tree to arrive at a closer estimate of the final prediction. The process continues until a maximum number of trees has been produced, referred to as the number of estimators. The process of creating new optimised trees in this way generally outperforms other decision tree models.

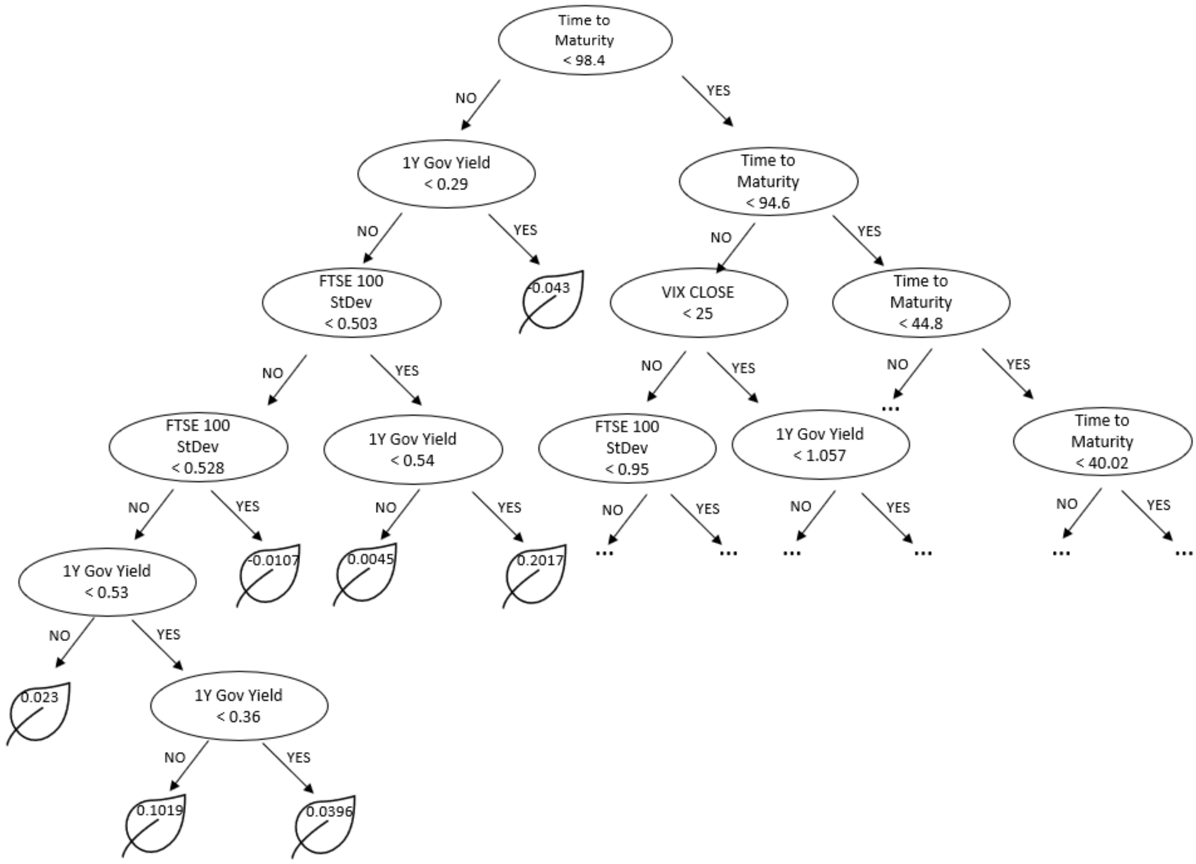


Figure 2: Diagram of a portion of the decision tree taken from the model. Leaf nodes reflect a terminal value in the tree but are not meaningful in themselves, as they are summed across all trees and then transformed to produce a final estimate. The full tree is available in APPENDIX IV.

This research uses the XGBoost (Chen and Guestrin, 2016) Python library version 1.7.3 to implement the gradient boosted regression tree model. XGBoost was selected due to being open-source and widely implemented in financial services as well as a popular algorithm in machine learning competitions such as Kaggle.

More detail on this model is contained within Appendix III.

5. Requirements

The following section will outline the requirements to achieve the objectives of this research.

5.1 Data sources

All data used in this research were sourced from one of the following sources:

1. Refinitiv, a London Stock Exchange-owned data platform
2. The Bank of England official data published on their website[1]
3. Office for National Statistics (ONS) official website[2]
4. Yahoo Finance[3]

5.2 Credit risk premiums as measured by Z-spreads

The Z-spread (zero volatility spread) is a fixed amount that, when added to the benchmark government spot rate curve, will make a bond's discounted future cashflows equal to the observed market price of the bond. It is a single numeric value that captures the excess risk and return investors currently demand to hold the bond compared to 'risk-free' government bonds of the same currency. The formula for calculating the Z-spread is as follows:

$$0 = \mathrm{MP} - \sum \frac{CF_x}{(1+Z+Y_x)^T}$$

where:

- MP refers to the market price of the bond on the date you are calculating the Z-spread
- CFx refers to the future bond cash flow at time x
- Y refers to the corresponding treasury spot yield at time x
- T refers to the T time to X expressed as an annual fraction
- Z refers to the Z-spread we are solving for

Therefore, Z is the number that makes the market price of the bond, minus the sum of the series of future discounted cash flows, equal to zero.

Therefore, to calculate the bond Z-spreads, we require:

1. Market Prices of Bonds
2. An interpolated Treasury Spot Curve

[1] https://www.bankofengland.co.uk/
[2] https://www.ons.gov.uk/
[3] https://uk.finance.yahoo.com/

Throughout this research, we will refer to Z-spreads by the industry standard measure: Basis Points (bps). A basis point is one-hundredth of a percent expressed as an integer. Therefore, 2.2% is 220 bps.

5.3 Feature variables used to estimate Z-spreads

A selection of company financial ratios, bond terms and conditions information and general market data were selected as estimators or features (to use machine learning jargon) for credit spread prediction. The list of items is presented in detail below. Note that the order of these items reflects the order (or column number) that they were fed into the machine learning algorithms and therefore the order required to make predictions from the machine learning algorithms:

5.3.1 Summary of feature variables

Data Item	Data Source
1. Bond Time to Maturity (Years as Decimal)	Refinitiv
2. 1 Year Benchmark Government Bond Yield	Bank of England
3. 10 Year Government Yield	Bank of England
4. 5 Year Breakeven Inflation	Bank of England
5. ONS GDP Growth Estimate	ONS
6. FTSE 100 Standard Deviation (22 day rolling)	Refinitiv
7. FTSE 100 Return (22 day rolling)	Refinitiv
8. VIX Daily Close Price	Yahoo Finance
9. Senior Subordination (Y/N)	Refinitiv
10. Unsecured (Y/N)	Refinitiv
11. Secured (Y/N)	Refinitiv
12. Quick Ratio	Refinitiv
13. Current Ratio	Refinitiv
14. Interest Coverage Ratio	Refinitiv
15. Working Capital to Total Assets	Refinitiv
16. Debt to Asset Ratio	Refinitiv
17. Debt to Capital Ratio	Refinitiv
18. Debt to Equity Ratio	Refinitiv
19. Net Profit Margin	Refinitiv
20. Operating Cashflow to Debt Ratio	Refinitiv
21. Return on Equity (ROE) using EBIT	Refinitiv
22. Return on Assets (ROA) using EBIT	Refinitiv

Table 1: Summary of feature variables used for model training.

5.3.2 Detailed information regarding feature variables[4]

1. **Bond Time to Maturity (years)**: The time to maturity is the remaining time until a bond reaches its maturity date. The maturity date is when the loan must be repaid to investors in full. The time to maturity is an important factor in determining the risk of a bond because longer maturity periods expose investors to more risks, such as inflation risk, credit risk, and interest rate risk. A longer time to maturity is associated with higher credit risk and vice versa. Therefore, longer times to maturity are associated with higher credit risk and therefore higher credit spreads.

2. **One Year Government Benchmark Yield**: The one-year government yield is the interest rate that the government currently pays to borrow money for a period of one year. It is used as a benchmark for the pricing of short-term bonds and as an indicator of the general direction of interest rates. This is because government rates are viewed as 'risk-free' and therefore serve as the theoretical minimum rate of return that an investor should demand from an investment. They are risk free because governments can theoretically print money to pay off debts, so their bonds should theoretically have no default risk.

3. **Ten-Year Government Benchmark Yield**: Like the one-year benchmark, the ten-year government benchmark yield is the interest rate that the government pays to borrow money for a period of 10 years. As discussed above, the longer the time to maturity of a bond, the higher perceived risk the bond poses to an investor, so the yield on the government 10-year bond is typically higher than the yield 1-year government yield. The difference between short-term interest rates and longer-term interest rates is referred to as the 'Yield Curve'. The shape of the yield curve can also be important in credit analysis. A normal yield curve, where long-term yields are higher than short-term yields, can indicate a healthy economic environment and therefore stable credit risk for corporate bonds with longer maturities. However, an inverted yield curve, where short-term yields are higher than long-term yields, can indicate a potential economic recession and increased credit risk for corporate bonds with longer maturities. By having both 1-year and 10-year government benchmark yields, the hope is to capture that the ML algorithm can discover the predictive feature of the yield curve in estimating credit spreads.

4. **Five Year Breakeven Inflation:** The five-year breakeven inflation is the difference between the yield on a five-year inflation-linked government bond and the standard five-year government benchmark. It reflects the market's expectations for inflation over the next five years and is used to gauge the inflation risk associated with a corporate bond. Inflation is an important factor when considering a bond investment because inflation will erode the returns on a bond. Because bonds pay a fixed rate of interest over a period, an investor will want that amount to be above the inflation level over that time frame. This is referred to as the 'real' rate of return, i.e., the return achieved after subtracting inflation. Therefore, as inflation expectations rise, so should yields, and consequently also spreads. An interesting element here is whether the inflation expectations are fully captured within the yield of government benchmark yields.

[4] OpenAI ChatGPT was used to assist writing with this section.

Recall that the credit spread reflects only the amount of yield above the government yield. If inflation expectations are fully reflected in government yields, then breakeven inflation should have no impact in predicting credit spreads because inflation should be fully factored into the model. However, to the extent that inflation impacts a company’s profitability and therefore its ability to service debt, inflation expectations should be idiosyncratic to the company’s risk and therefore should not be fully captured in government yields.

5. **The Office for National Statistics GDP Growth Estimate**: A measure of the expected growth rate of the United Kingdom's economy as estimated by the economists at the Office for National Statistics. A company's creditworthiness is largely determined by its ability to generate revenue and cash to service its debts. These in turn are influenced by broader economic conditions. If the economy is growing, companies may have more opportunities to increase sales and profits, which can improve their creditworthiness and reduce the credit risk of their bonds. Conversely, if the economy is contracting or experiencing slow growth, companies may struggle to generate revenue and profits, which can increase their credit risk and make it more difficult to repay debt. Therefore, the GDP growth estimate provides insight into broader economic conditions in the UK, which in turn may be associated with higher or lower credit spreads.

6. **FTSE 100 22-Day Rolling Standard Deviation of Daily Returns:** The standard deviation is a measure of the volatility of the FTSE 100 index. It is used to assess the overall level of market risk associated with a corporate bond. Market volatility may be an important predictor of credit spreads because market volatility is associated with moments of financial distress for companies. Therefore, one could expect a positive correlation between stock market volatility and credit spreads. Note that if the bond issuer had a publicly listed stock price, this would almost certainly be a much better predictor of credit spreads than the broad FTSE 100 stock market. However, since not all bond issuers have publicly listed stocks, this estimator would not be available in all situations. To make the results of this research as general as possible, no stock-specific data were used as a predictor. This was calculated by taking the Refinitiv daily FTSE 100 values and calculating the daily return in Excel. Then, the rolling 22-day standard deviation was calculated off the daily returns.

7. **FTSE 100 22-Day Rolling Return**: A measure of stock market performance. This may be correlated to credit risk since the performance of the stock market reflects the financial performance of companies and the general economic climate. Both factors should affect the ability of a company to generate revenue and repay debt. Therefore, the assumption here is that positive stock market performance should be associated with lower credit spreads and vice versa.

8. **VIX Daily Close Price**: The Chicago Board Options Exchange (CBOE) Volatility Index, commonly known as the VIX, measures the expected volatility of the S&P 500 index, which is a benchmark index of the US stock market. The VIX is unique in that it extrapolates the forward-looking volatility that is priced into stock options on the underlying index. This estimator is similar to

item 7, except it is more forward-looking and it is calculated on US listed companies rather than UK companies. The VIX should theoretically serve as a barometer for global economic conditions.

9. **Senior Subordination**: Senior debt refers to the level of subordination for a particular debt instrument. A bond with senior subordination means it has priority over other forms of debt in the event of default or bankruptcy. It is considered less risky than other forms of debt and is therefore associated with a lower credit risk. Here, a flag was used with 1 indicating that a bond has senior subordination, whereas a 0 was used to reflect a bond that did not.

10. **Unsecured**: Unsecured debt is a type of bond that is not backed by collateral, i.e., assets pledged to secure the bond. This means that in the event of default, the bondholders do not have a specific claim on any assets of the company to be repaid. Instead, these bonds rely solely on the company's ability to generate sufficient cash flows to repay the bond. It is considered riskier than secured debt and should therefore be associated with a higher credit risk and higher credit spreads.

11. **Secured**: Secured debt is backed by collateral. It is therefore considered less risky than unsecured debt and should be associated with a lower credit risk and lower credit spreads.

12. **Quick Ratio**: The quick ratio, also known as the acid-test ratio, is a financial ratio that measures a company's ability to meet its short-term obligations using its most liquid assets, such as cash, marketable securities, and accounts receivable. The quick ratio is calculated by dividing the company's current assets minus its inventory by its current liabilities. This ratio is used to assess a company's ability to pay off its short-term debts without having to sell its inventory or other long-term assets. The ratio is a potential estimator to consider when assessing a company's ability to meet its debt obligations. A company with a high Quick Ratio should generally be considered to have a stronger ability to meet its short-term obligations, which can reduce the credit risk of its bonds. or the purpose of this research, the Quick Ratio was taken directly as calculated by Refinitiv.

13. **Current Ratio**: The current ratio is also a measure of a company's ability to meet its short-term obligations. It is calculated by dividing a company's current assets by its current liabilities. A company with a high current ratio is generally considered to have a stronger ability to pay off its short-term debts, which may reduce the credit risk of its bonds and therefore credit spreads. For this research, the current ratio was taken directly as calculated by Refinitiv.

14. **Interest Coverage Ratio**: The interest coverage ratio is a financial ratio that measures a company's ability to pay off its debt interest expenses using its earnings before interest and taxes (EBIT). It is calculated by dividing a company's EBIT by its interest expenses. A company with a high interest coverage ratio is generally considered to have a stronger ability to pay off its interest expenses, which can reduce the credit risk of its bonds and should therefore be

associated with lower credit spreads. For this research, the interest coverage ratio was taken directly as calculated by Refinitiv.

15. **Working Capital to Total Assets**: The working capital to total assets ratio is a measure of a company's ability to meet its short-term obligations. It is calculated by dividing a company's working capital (current assets minus current liabilities) by its total assets. A company with a high working capital to total assets ratio is generally considered to have a stronger ability to meet its short-term debt obligations, which can reduce the credit risk of its bonds. This is because a higher working capital to total assets ratio suggests that the company has more liquid assets available to meet its short-term obligations. For this research, the ratio was calculated manually from Refinitiv Data as follows:

$$\frac{Total\ Current\ Assets - Total\ Current\ Liabilities}{Total\ Assets}$$

16. **Debt to Asset Ratio**: The Debt-to-Asset ratio is a financial ratio that measures the proportion of a company's total assets that are financed by its debt. Conversely, the Debt-to-Asset ratio can be seen as a measure of the burden of a company's debt relative to its asset base. It is a measure of financial leverage, which is financial industry jargon meaning the debt load a company bears or the amount by which a company is financed by debt. In general, a lower debt to asset ratio may suggest that a company is less risky because it has a lower level of debt relative to its assets, which means it may have a greater ability to repay its debts in case of financial distress. However, this may be complicated by the fact that a company that uses debt to generate new business may be more profitable than a similar company that does not. Therefore, measures of financial leverage should theoretically have nonlinear relationships to credit risk and credit spreads. For the purpose of this research, the ratio was calculated manually from Refinitiv data as follows:

$$\frac{Total\ Debt}{Total\ Assets}$$

17. **Debt to Capital Ratio**: The Debt-to-Capital ratio is also a measure of a company's leverage. It is calculated by dividing a company's total debt by its total debt and equity (assets – liabilities). A higher ratio indicates a higher degree of financial risk and is therefore associated with a higher credit risk. However, like the Debt-to-Asset ratio, it could also suggest that the company has been able to leverage its capital to generate higher returns, so it may not entirely be credit negative. For this research, the ratio was calculated manually from Refinitiv data as follows:

$$\frac{Total\ Debt}{Total\ Debt\ +\ (Total\ Assets\ -\ Total\ Liabilities)}$$

18. **Debt to Equity Ratio**: The Debt-to-Equity ratio is also a measure of a company's leverage. It is calculated by dividing a company's total debt by its total equity similar to the above two ratios. For this research, the ratio was calculated manually from Refinitiv data as follows:

$$\frac{Total\ Debt}{Total\ Assets\ -\ Total\ Liabilities}$$

19. **Profit Margin**: The profit margin is a measure of a company's profitability. It is calculated by dividing a company's income by its revenue. A higher net profit margin indicates a greater ability to generate profits and is therefore associated with a lower credit risk. A company with a high net profit margin is generally considered to be more profitable and financially healthy, which can reduce the credit risk of its bonds. Therefore, a higher margin should theoretically be associated with lower spreads. Because taxes are idiosyncratic and depend on factors external to the company, the profit margin here was calculated on earnings before tax. For this research, the ratio was calculated manually from Refinitiv data as follows:

$$\frac{Earnings\ Before\ Tax}{Total\ \ Revenue\ From\ Business\ Activities}$$

20. **Operating Cashflow to Debt Ratio**: A financial ratio that measures a company's ability to generate enough cash flow to cover its debt obligations. It is calculated by dividing a company's operating cash flow by its total debt. Cash flow is more important than profitability to a credit investor because creditors are only concerned with the company's ability to pay debt. A firm that generates much operating cash flow but is not profitable may still be viewed as low risk to a creditor. A high operating cashflow to debt ratio suggests that a company has sufficient cash flow to cover its debt obligations, which can reduce the credit risk of its bonds and therefore should be associated with lower credit spreads. For this research, the ratio was calculated manually from Refinitiv data as follows:

$$\frac{Net\ Cash\ Flow\ From\ Operating\ Activities}{Total\ Debt}$$

21. **Return on Equity using EBT**: Return on Equity (ROE) is a financial ratio that measures a company's profitability by expressing its income as a percentage of shareholders' equity. ROE using EBT (Earnings Before Taxes) is potentially more relevant to a bond investor than net income. As previously mentioned, taxes are idiosyncratic and not easily comparable between companies in an international context, so they should be removed for the purpose of this analysis. Additionally, credit investors are concerned with evaluating a company's ability to generate earnings to service debt, and EBT indicates how much income remains available after having paid interest but prior to paying taxes. Unlike dividends to shareholders, interest payments are made from pretax income, so EBT is a logical measure for a company's earnings from the perspective of a bond investor. For this research, the ratio was calculated manually from Refinitiv data as follows:

$$\frac{Earnings\ Before\ Taxes}{Total\ Assets\ -\ Total\ Liabilities}$$

22. **Return on Assets using EBIT**: Similar to the above except using the company's assets as a denominator, calculated manually from Refinitiv data as follows:

$$\frac{Earnings\ Before\ Taxes}{Total\ Assets}$$

6. Methodology

This section will outline what methodology was employed to accomplish the goals of this research.

6.1 Data collection and compilation

The data requirements for this research posed a complex problem. Bond prices, and therefore credit spreads, change daily depending on economic conditions and supply and demand. The underlying company financial data used as predictors in this research are generally reported on a quarterly or semiannual basis. A method to collect bond prices, calculate the Z-spreads and then link these data with the underlying bond issuer's financial data at a correct point-in-time posed a considerable challenge.

First, the universe of corporate bonds is massive and therefore would need to be shrunk to an appropriate scale for this research. Therefore, the universe of bonds was limited to only those that are GBP Sterling denominated. Note, however, that these GBP-denominated bonds could be issued by any corporation worldwide, not necessarily just those operating in the United Kingdom.

As previously mentioned, only vanilla corporate bonds were in scope for this research to avoid issues around bonds with embedded options or other exotic features such as inflation-linking. Therefore, using the Refinitiv platform, a screen for bonds was created with these conditions:

1. Universe = Corporate Bonds
2. Currency = GBP
3. Excluded coupon types: variable, floating, zero/coupon, resettable, range, to-be priced
4. Exclude Convertibles bonds
5. Exclude Green Bonds
6. Exclude Putable bonds
7. Exclude Callable bonds
8. Exclude bonds with Sinking Fund
9. Exclude inflation-linked bonds
10. Include inactive/nontrading bonds
11. Exclude bonds from Financial Companies

Because banks and financial firms are major issuers of bonds and banks tend to have very different financial operations from nonbanks, bonds from these types of companies were excluded. This is consistent with the approach taken by Altman (1968)

The result was a list of 596 bonds that was downloaded with corresponding terms and conditions information and is referred to as the 'Master' list.

Each bond was individually loaded into the Refinitiv Workspace analysis application, and the entire available monthly price series for the bond was downloaded into Excel format. The downloaded prices corresponded to the last day of each month for the date range selected. The available date range for prices was between Jan 31, 2003, and Jan 31, 2023, so the research spans the economic period from 2003 to 2023 or 20 years. In total, 30,036 monthly bond prices were collected.

For each bond price time series downloaded, the corresponding parent company's financial statements were also manually downloaded from Refinitiv. These were downloaded for as long a timeframe as available and with a financial reporting frequency that provided the most information, so first Quarterly if available, then Semi Annually if available, then Annually if available.

Once the data were downloaded, two complex tasks remained:

1. Calculate the Z-spreads from all the bond prices.
2. Create table data that associates the bond Z-spread to the corresponding company financial data and the required bond terms and conditions from the 'Master' list so it can be fed into the machine learning algorithms.

To perform these tasks, a set of software tools was developed to facilitate data aggregation, manipulation and calculation, model training and evaluation, including a GUI to navigate it. This is available here[5]. The details of this tool are available in Appendix I.

6.2 Calculating Z-spreads

The main input required in calculating the Z-spread is the Government Spot Curve that corresponds to the date for which you are calculating the Z-spread, i.e., the date that corresponds to the market price of the bond in question. The monthly government spot curve data were sourced from the Bank of England website with data going as far back as 1970. However, to calculate Z-spreads, we require a spot curve with points (referred to as 'tenors' in industry jargon) that correspond to each future cash-flow date of the bond. The bank of England data provided monthly data points on the curve for the first few years and then semiannual data points up to 25 years. Beginning in 2016, the spot curves were available with semiannual data up to the 40-year point. Bonds may pay cash flows for periods much longer than this, so this potentially posed a problem. Furthermore, the semiannual data points available do not perfectly correspond to the dates of the cash flows of the bonds. Therefore, a method of interpolation and extrapolation was needed. The chosen method was a cubic spline, which has been found to be a reasonable method of yield curve interpolation (McCulloch, 1975). Cubic splines are piecewise polynomial functions used for interpolating smooth surfaces. Here, the spline was implemented using an in-built method from the Python Pandas 1.5.3 library.

No bonds had cash flows extending beyond 40 years; however, a handful had cash flows extending beyond 25 years, for which there was no yield curve point prior to 2016, and therefore, extrapolation was needed. To do this, the average difference between the 25-year curve tenor and the 40-year curve tenor was taken for all the data available, i.e., all the months from 2016 onwards. This average was found to be -21 basis points. The distributions of these points are presented in Figure 3 and show that they generally stay within a band, so using the average value for extrapolation seemed a reasonable approach.

[5] https://github.com/Reneige/EstimatingCreditSpreadsViaMachineLearning

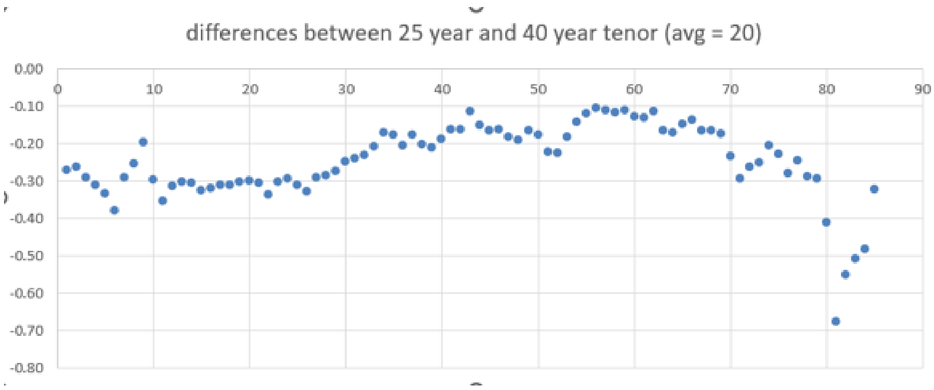


Figure 3: Differences in percent yield between the 25- and 40-year tenor points for all GBP Government Spot Curves (85 in total presented on the Y-Axis) and expressed in percent (X-Axis)

To extrapolate the curve up to the 40-year point, an amount equal to the 25-year tenor minus 21 basis points was inserted into the data at the 40-year tenor point, and the cubic spline was performed up to that new point. The python code written to perform curve interpolation code is available here[6] and forms part of the wider application developed for this research. The resulting curves are referred to as the 'Valuation Curves', as they are used to value the bonds to calculate Z-spreads.

Figure 4 shows the unaltered government spot curve on 31 October 2022 spanning the full 40-year period. Figure 5 below presents an example spot curve where the daily data have been interpolated but have not been extrapolated beyond the 25-year point. Figure 6 below presents the same curve after extrapolation, generating data from the 25-year point to the 40-year point. The similarity between the extrapolated points in Figure 6 and the curve in Figure 4 suggests that the extrapolation methodology generates a natural curve structure.

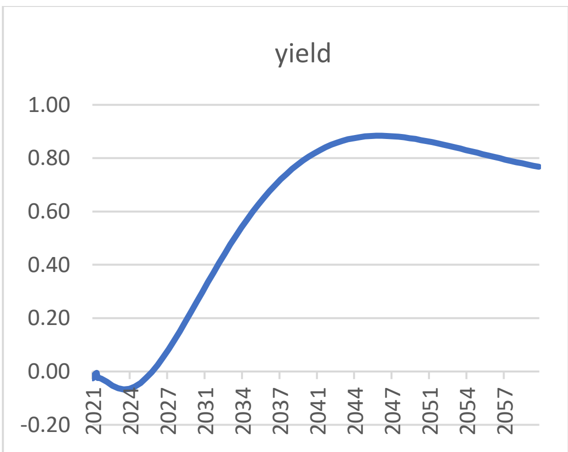


Figure 4: Government spot curve on October 31, 2020. Yield-to-maturity % (Y-axis), year (X-axis)

[6]https://github.com/Reneige/EstimatingCreditSpreadsViaMachineLearning/blob/master/valuation_curve_builder.py

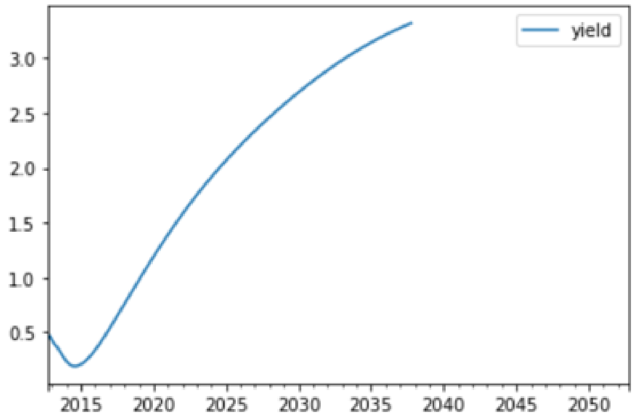


Figure 5: Daily interpolated spot curve before extrapolation of the 40-year tenor point. Yield-to-maturity % (Y-axis), year (X-axis)

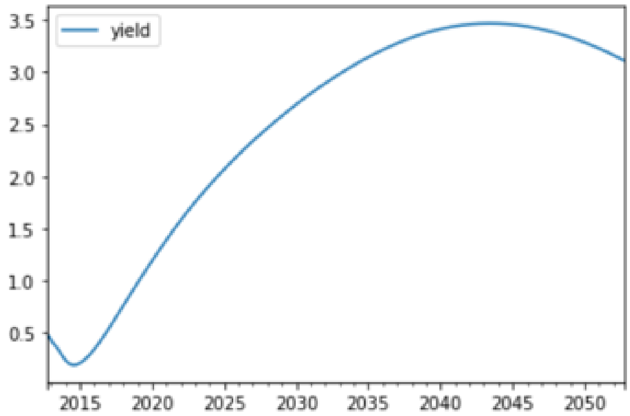


Figure 6: Daily interpolated spot curve after extrapolation of the 40-year tenor. Yield-to-maturity % (Y-axis), year (X-axis)

Once the valuation curve is constructed, we can continue to calculate the Z-spread. To do this, for each bond and for each day that the bond has a price available, we project all the bonds' interest payments (referred to as 'coupons' in industry jargon) and the final repayment of the loan. This is performed by taking the maturity date, the first coupon date, the issue date and the coupon frequency information from the Master list of bond data. The valuation curve that corresponds to the price date of the bond is merged into the data, giving us the corresponding government benchmark yields on the cash flow dates. Then, an iterative algorithm is applied to progressively find the Z-spread that equates the discounted cash flow to the current market price of the bond. Once this value is found, the process continues for each bond and for each date. The resulting data are stored within the 'Prices' table of the database in the Python tool. The specific module that calculates this is available here[7]

[7] https://github.com/Reneige/EstimatingCreditSpreadsViaMachineLearning/blob/master/ZSpreadCalc.py

6.3 Distribution of Z-spread data

The distribution of calculated Z-spreads resulting from the above methodology is depicted here. The majority of observations fall between 50 bps and 250 bps, which is reasonable given the very small share of the overall bond market that is inhabited by so-called 'high yield'.

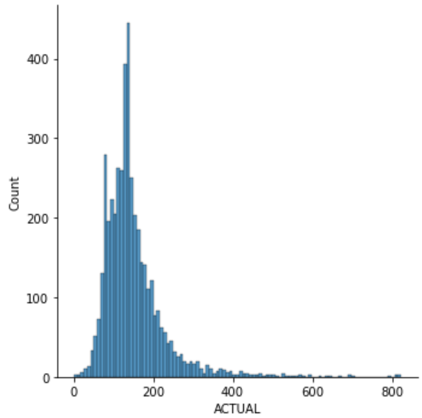


Figure 7: Distributions of Z-spreads - # observations (x-axis), Z-spread in basis points (y-axis)

6.4 Compiling the research data

With Z-spreads now calculated, the table of research data to train the machine learning algorithms must be compiled. This is complex because it requires combining data from various sources and linking them by date. The methodology employed to do this was as follows. Software was written to aggregate all the bond price data, all company financial data, and all the economic data, and the master bond list, which includes the bond static data, using Python Pandas (available here[8]). These data were inserted into an SQLite database with a separate table for Prices, Financials, Master Bond List, and for each of the Economic Series: VIX, FTSE 100, ONS GDP Estimates, Yield Curves, Inflation Curves. Bond Prices and Z-spreads were linked to the corresponding company financial data by using the Master List as a mapping table (Bond → ISIN → Company Name → Company Financial Data). All the required data were then queried from the database by 'joining' the data sets on date. Crucially, however, because financial data are reported with a lag, the data were transformed so that the reporting dates were pushed forward by 3 months to help avoid a look-ahead bias. This means that the Price Data/Z-spread that was 'joined' between the prices table and the financial data table was for 3 months after the reporting date. Three months seemed to be a reasonable amount of time to lag the reporting data, given that many companies report financial data quarterly.

The resultant data set was then adjusted as follows:

1. The data were trimmed to remove each line where there was a missing feature.
2. Lines of data where the calculated Z-spread was less than one year from the bond maturity date were removed. This is due to annualisation of yields and spreads that below one year result in distorted values, which would negatively impact the training process.

[8] https://github.com/Reneige/EstimatingCreditSpreadsViaMachineLearning/blob/master/data_aggregator.py

3. Some extreme values of financial ratios were removed because they could negatively impact the training of the neural network. Specifically, the rows of data where the Operating Cashflow to Debt ratio was above 50 were removed.
4. Rows with extreme and suspicious Z-spread values were removed. Specifically, all Z-spreads above 1000 or below -10 were removed.

The final data set had 22,167 rows of training data spanning 197 bonds and 71 bond issuers.

6.5 Relationship between predictors and Z-spreads

It can be useful to visualise estimators (X Variables) with respect to Labels (Y variables) to see if there are any clear relationships. Note that some of the outliers have been removed from the charts below to allow the images to render properly.

6.5.1 Market data variables

Plotting the Z-spreads against feature variables can help identify any linear relationships between these variables. This is performed below in Figure 8. Here, there does appear to be a slight negative relationship between Z-spreads and nominal yields and inflation. Low values of the VIX and FTSE Volatility are associated with lower credit spreads, which seems logical.

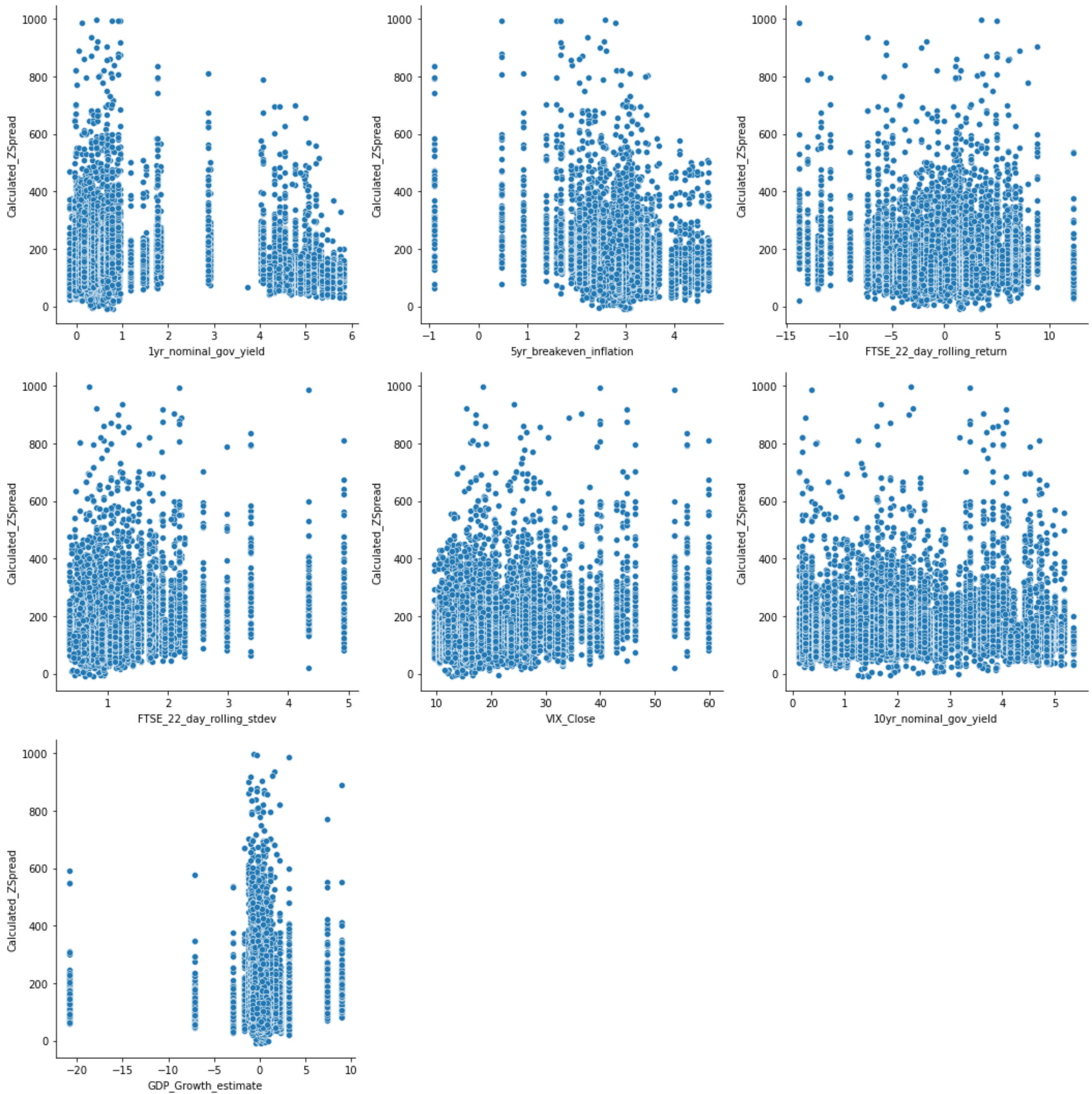


Figure 8: Z-spreads (bps Y-axis) plotted against various market data feature variables (Y-axis)

6.5.2 Bond static data

Senior (SR) and Secured debt (SEC) exhibit much lower Z-spreads. This is logical and is expected given the lower risk profile of those debt types. The unsecured category (UN) is less intuitive, exhibiting lower spreads. This category should theoretically be mutually exclusive from the secured category. This suggests that the data collected from Refinitiv may not be completely reliable. The time to maturity exhibits a very strong negative relationship with Z-spreads, which is unintuitive. This could be due to long-term bonds being restricted to issuers deemed to be the most creditworthy.

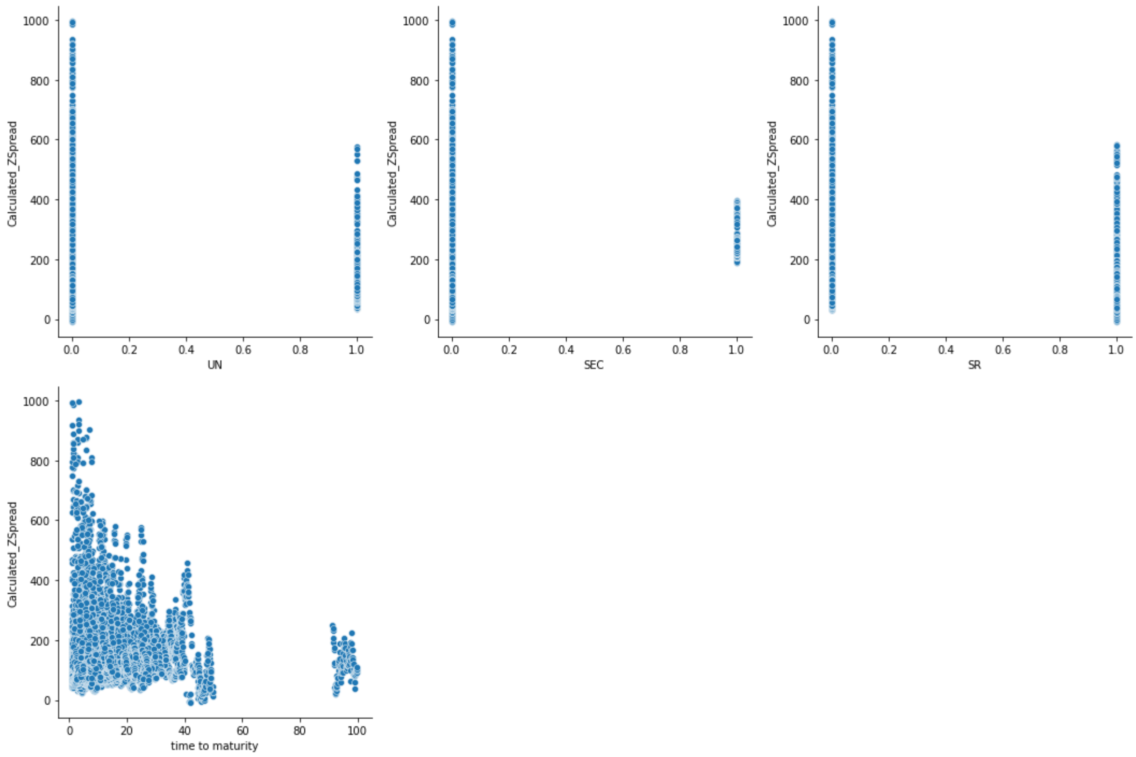


Figure 9: Z-spreads (bps Y-axis) plotted against various bond static data feature variables (Y-axis).

6.5.3 Financial ratios

Many of the financial ratios exhibit clear inverse relationships to credit spreads, as would be expected, given that most of the ratios indicate financial distress at low levels.

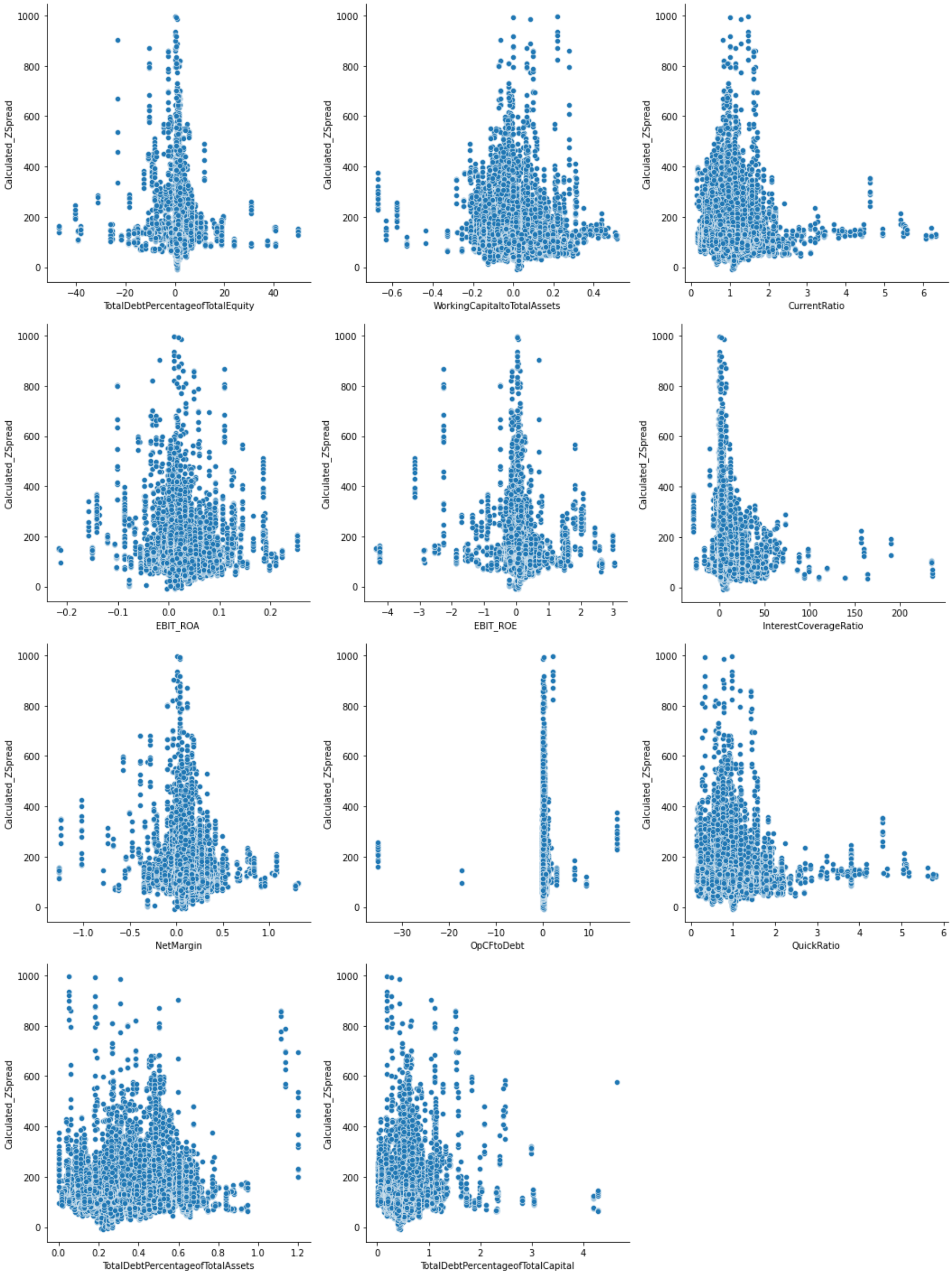


Figure 10: Z-spreads (bps Y-Axis) plotted against various financial ratios (Y-Axis).

7. Training the machine learning algorithms

For both models, the data were split into an 80% training and 20% testing set. For the neural network model, 80% of the training data were further split into 25% of the validation data.

7.1 Training the neural network

There is no rule book for training neural networks. The process is empirical and is best suited to trial and error, so it is therefore as much an art as it is a science. Given the 22 variables that were being used as estimators, the process began with a network with 3 hidden layers that shrunk progressively from 22 nodes down to one, along with dropout nodes to prevent overtraining. Through trial and error, it quickly became apparent that dropout nodes were not necessary. The noisiness of the data ensured that no overtraining occurred. In fact, the data could be trained for very long periods of time, and the loss and validation (Figure 11) would eventually flatten but never increase, which is the characteristic of overtraining. It also became clear that a much deeper and wider network structure produced better results. Through trial and error, the final neural network model contained 8 hidden layers shrinking progressively with the following numbers of nodes: 576, 288, 144, 72, 36, 18, 9, and 3, for a total number of trainable parameters of 234,979.

For all hidden layers, a ReLU activation function was used. Given that this is a regression problem, the output layer used a linear activation function to predict the single value of the bond's Z-spread. The loss function was set as the mean squared error for computational efficiency; however, the results were measured by the mean absolute error for the sake of interpretability. While it is standard practice to normalise the training data, I did not perform this step for the following reasons:

1. I wanted to preserve the values of the test data for model interpretation and exploration.
2. Most of the input data were in % or Ratio form which is already somewhat standardised.
3. The Boolean data for Senior/Secured/Unsecured were not suitable for standardisation.
4. Standardisation is meant to help prevent training sessions that initialise badly, but it is clear from the slow learning rate when this occurs. Therefore, as a workaround, when the learning rate is visibly too slow, the process can just be restarted until a better learning rate is observed.

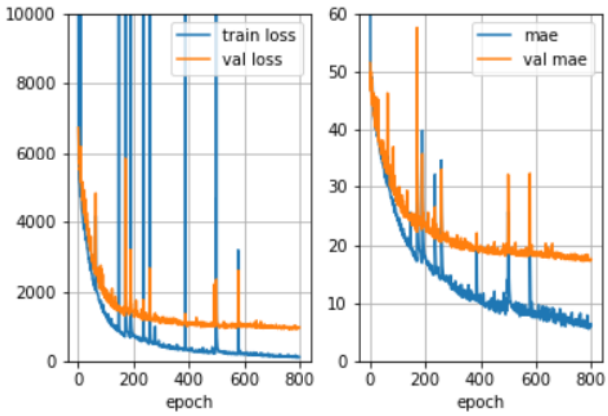


Figure 11: Left: Training set and validation set loss measured as the mean squared estimate error of the Z-spread in basis points (Y-axis) per epoch. Right: Training set and validation set accuracy measured as the mean absolute error of the Z-spread in basis points (Y-axis) per epoch.

7.2 Training the gradient boosted regression tree model

The exact same data were used except without a separate validation set, so we used only the 80/20 train/test split. The metric used was the mean absolute error. The number of estimators/trees was set to 8000 given that the data do not appear to be prone to overtraining. However, above 1000, the training displayed extremely diminished returns, as presented in the learning curve chart below. The learning rate was set to 0.05.

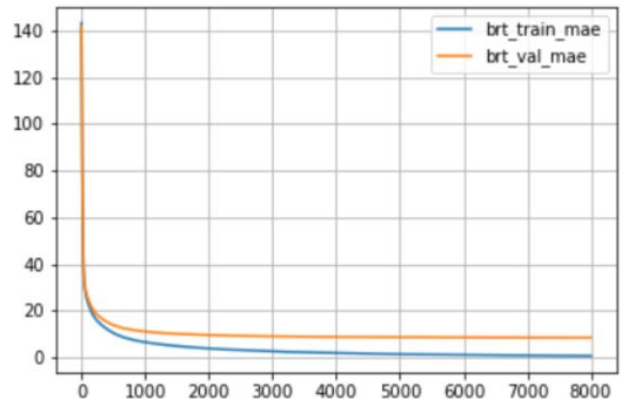


Figure 12: Training set and validation set accuracy measured as the mean absolute error of the Z-spread in basis point (y-axis) achieved with each additional regression tree added to model (x-axis)

8. Results

Twenty percent of the 22,167 lines of training data collected were held out from the training data set to test the accuracy of the trained machine learning models. This left 4,435 rows of testing data. The mean value of the Z-spreads in the test set was 149 bps. The standard deviation of the Z-spreads in the test set is 76 bps.

Consistent with the method used to evaluate the accuracy when training the models, the mean absolute error was used on the testing set. The results for each model are presented in sections 8.1 and 8.2 immediately below. The sections immediately after will explore how the models performed their predictions by analysing which variables were most important to either model. Finally, an analysis of some individual estimates will be performed to gain insight into how the models are arriving at their estimates.

8.1 Neural network results

The mean absolute error across all estimates on the test data using the Neural Network model was 16.97 bps. The standard deviation of the mean absolute error was 24.42 bps. The adjusted R^2 value calculated was 0.823, indicating a good predictive ability of the model. The distribution of model estimates versus actual values is depicted in Figure 13 below (left). For context, if the model had perfect prediction capability, the chart would be a single horizontal line from bottom left to upper right. The clustering across the vertical line gives an indication of how closely the model is performing.

Overall, the Neural Network model clearly has good predictive capability in estimating the credit spreads.

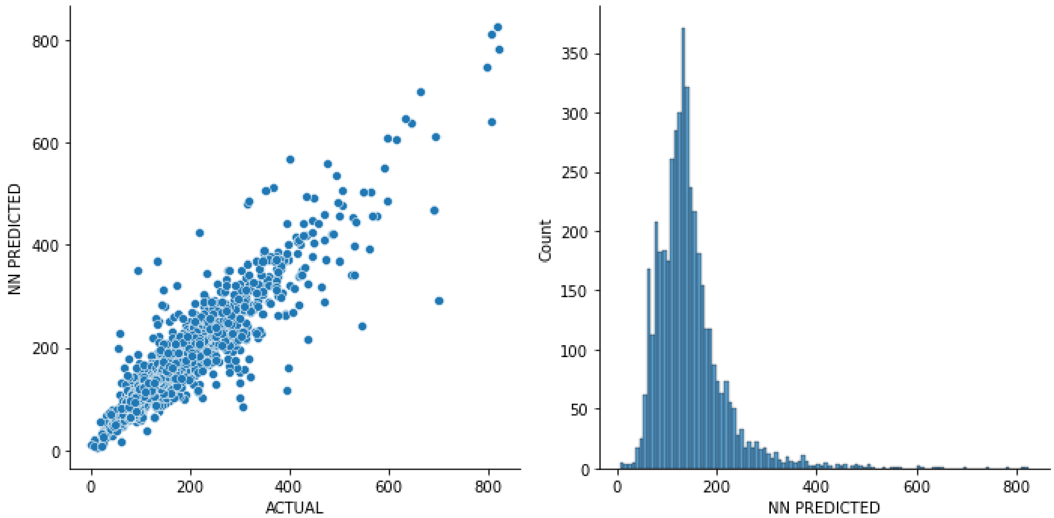


Figure 13: Left: Neural Network predicted Z-spread (y-axis) plotted against actual Z-spread (x-axis). Right: The distribution or frequency of neural network predicted Z-spreads (x-axis).

8.2 Gradient boosted regression tree results

The mean absolute error across all estimates on the test data using the gradient boosted regression tree (GBRT) model was 8.38 bps. The standard deviation of the mean absolute error was 14.01 bps. The adjusted R^2 value calculated was 0.95, indicating very strong predictive power of the model. The distribution of model estimates versus actual values is depicted in Figure 13 below (left). The tight clustering across the vertical line shows how capable the model is at estimating credit spreads.

This is an impressive result that shows that the gradient boosted regression tree model clearly has strong predictive capability in estimating credit spreads and significantly outperforms the neural network model.

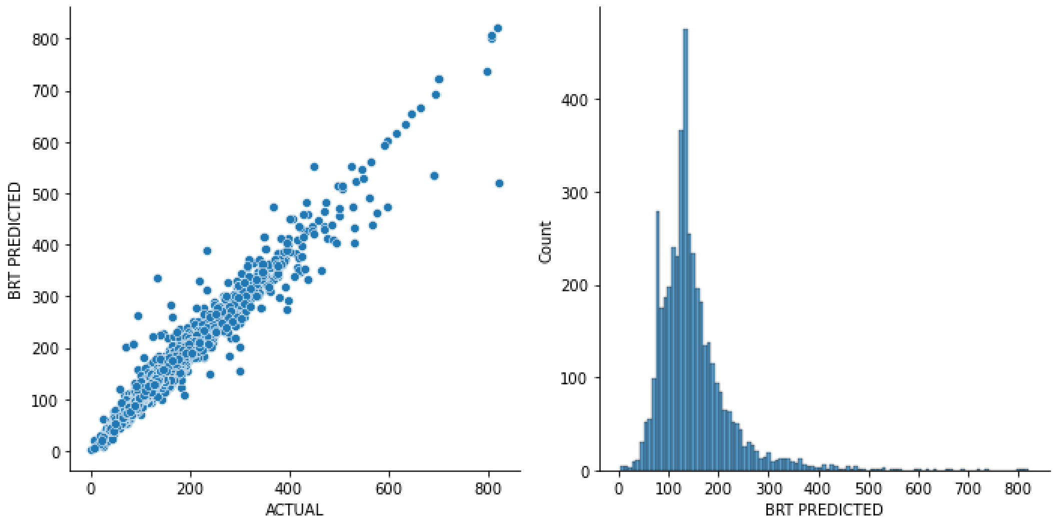


Figure 14: Left: boosted regression tree predicted Z-spread (y-axis) plotted against actual Z-spread in testing data (x-axis). Right: The distribution or frequency of GBRT predicted Z-spreads (x-axis).

8.3 Feature importance within the gradient boosted regression tree model

The XGBoost Python library has an inbuilt capability to inspect the feature importance of the trained model. Feature Importance is the average number of splits in the trees attributed to a particular feature (X-Variable in the model). However, the splits in themselves do not necessarily mean a feature is a strong predictor.

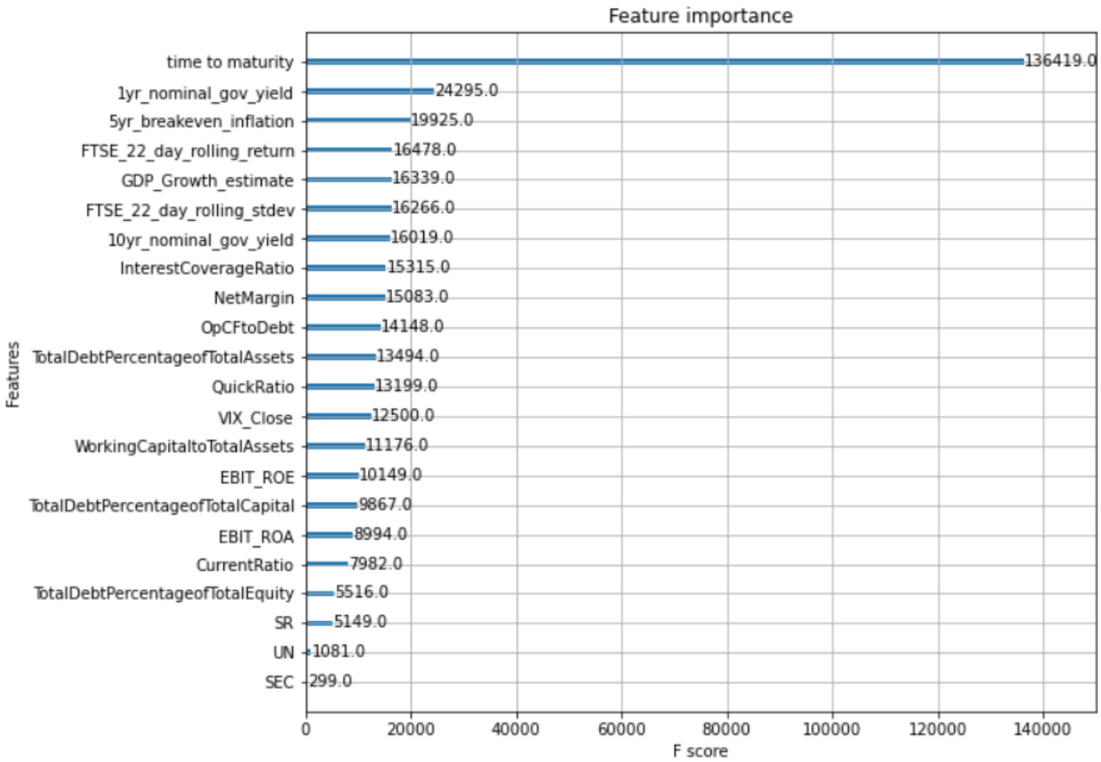


Figure 15: The length of the bar displays the frequency with which a feature was split into producing branches of the decision trees within the GBRT model.

8.4 Weights of the gradient boosted regression tree model

Using the Eli5 Library (Korobov M., Lopuhin K, 2021), we can inspect the model weights of the gradient boosted regression tree model. These weights depict the average gain that results from including a particular feature in the model. The list is presented from highest to lowest. A higher number suggests that a feature provides greater predictive power relative to a lower number. Interestingly, the weights are quite different from the importance presented above.

Data Item	Weight
1. Secured (Y/N)	0.3077
2. Debt to Capital Ratio	0.0802
3. Interest Coverage Ratio	0.0603
4. Senior Subordination (Y/N)	0.0555
5. Return on Equity (ROE) using EBIT	0.0550
6. Debt to Capital Ratio	0.0468
7. Return on Assets (ROA) using EBIT	0.0460
8. FTSE 100 Standard Deviation (22 day rolling)	0.0445
9. Operating Cashflow to Debt Ratio	0.0443
10. Quick Ratio	0.0375
11. Debt to Asset Ratio	0.0343
12. 5 Year Breakeven Inflation	0.0335
13. Net Profit Margin	0.0329
14. Current Ratio	0.0317
15. Working Capital to Total Assets	0.0223
16. Unsecured (Y/N)	0.0153
17. VIX Daily Close Price	0.0138
18. 10 Year Benchmark Government Bond Yield	0.0136
19. 1 Year Benchmark Government Bond Yield	0.0121
20. Time to Bond Maturity (Years as Decimal)	0.0057
21. FTSE 100 Return (22 day rolling)	0.0037
22. ONS GDP Growth Estimate	0.0034

Table 2: Model weights of the gradient boosted regression tree model. (Eli5 Python Library)

8.5 Feature Importance in the neural network model

Using the Eli5 library, we can inspect the feature importance of a neural network model using permutation importance or mean decrease accuracy (MDA). This works by measuring the model's predictive capability when any one feature is not available to the model.

Weight	Feature
3745.2195 ± 110.2713	InterestCoverageRatio
3285.1614 ± 132.3526	time to maturity
3193.6148 ± 68.9465	SR
2770.6619 ± 110.7310	VIX_Close
2656.0179 ± 91.3770	10yr_nominal_gov_yield
2532.6134 ± 149.8877	TotalDebtPercentageofTotalEquity
1581.3750 ± 103.0874	CurrentRatio
1497.3258 ± 36.4628	1yr_nominal_gov_yield
1335.3927 ± 125.6018	QuickRatio
834.5083 ± 61.6248	WorkingCapitaltoTotalAssets
682.4694 ± 37.3161	5yr_breakeven_inflation
666.0549 ± 15.6881	TotalDebtPercentageofTotalAssets
637.2681 ± 49.2752	FTSE_22_day_rolling_return
625.5861 ± 38.9173	UN
616.4917 ± 61.1880	FTSE_22_day_rolling_stdev
609.3422 ± 72.6969	GDP_Growth_estimate
515.0343 ± 31.6653	EBIT_ROE
474.2814 ± 42.2941	NetMargin
380.1742 ± 31.4339	TotalDebtPercentageofTotalCapital
214.7334 ± 18.1496	OpCFtoDebt
135.0073 ± 16.4848	SEC
19.0167 ± 2.5352	EBIT_ROA

Figure 16: Inspection of Neural Network model feature importance using Eli5 Python Library

While the models agree on certain estimators, such as Interest Coverage Ratio, Debt-to-Equity and senior subordination, there are some significant divergences in the feature importance of each model. For example, the profitability ratios, ROE and ROA and Net Margin are all very low on the Neural Network model, whereas they are quite high on the Regression Tree model. The models agree on

8.6 Inspecting individual predictions

Using the LIME Library (Ribeiro, M., Singh, S., and Guestrin, C., 2016) for the Neural Network (NN) Model and using the Eli5 Library for the Gradient Boosted Regression Tree Model (GBRT), we can analyse individual predictions of each model on the same set of data. This gives us an idea of how the algorithms are arriving at their predictions and which estimators are the most important for their predictions.

Below, we will inspect 3 different model predictions. The first is an example where the models both predict accurately. The second is an example where the GBRT model predicted accurately and the NN model did not. The third example is a prediction where the GBRT model predicted inaccurately and the NN model predicted accurately. The purpose of this analysis is to obtain a sense as to which features drive the model predictions. Whether the models agree or disagree on feature importance can inform us as to whether the models are learning real economic variables or simply producing highly optimised but ultimately non economically driven results.

8.6.1 Result sample 1: both models predict accurately

Feature/Estimator	Value
Time to Bond Maturity	1.05
1-Year Benchmark Government Bond Yield	0.10
10-Year Benchmark Government Bond Yield	0.80
5-Year Breakeven Inflation	2.68
ONS GDP Growth Estimate	0.21
FTSE 100 22-day rolling standard deviation	0.89
FTSE 100 22-day rolling return	1.74
VIX Close Price	13.29
Senior	0.00
Unsecured	0.00
Secured	0.00
Quick Ratio	0.86
Current Ratio	1.09
Interest Coverage Ratio	5.00
Working Capital to Total Assets	-1.55
Total Debt Percentage of Total Assets	5.04
Total Debt Percentage of Total Capital	0.97
Total Debt Percentage of Total Equity	31.13
Net Margin	0.01
Operating Cashflow to Debt	0.01
Return on Equity measured with EBIT	0.19
Return on Assets measured with EBIT	0.03
Actual Z-spread	**79.60**
Neural Network Predicted Z-spread	**78.30**
Gradient Boosted Tree Predicted Z-spread	**78.98**

Table 3: Analysis of actual versus predicted values of Z-spread based on a single set of estimators.

Comparing Figure 17 and Figure 18, we see that negative working capital, low net margin, and lack of senior subordination were strong positive contributors to spread predictions in both models. Interestingly, the current ratio, with a value near 1, was the largest positive contributor to the regression tree model, whereas for the neural network model, it was slightly negative. From the perspective of a credit analyst, current assets that only just cover current liabilities could be seen as risky, so the Regression Trees interpretation appears more sensible. The Debt-to-Capital, Debt-to-Equity and Debt-to-Assets ratios had the same direction of influence in predicted spreads between both models, but with varying intensity. For the neural network model, the debt-to-capital ratio strongly increased the predicted credit spread, whereas for the GBRT model, it only slightly increased the prediction. For both models, the short time to maturity had a strong negative impact on predicted credit spreads, which is a sensible assumption. In general, the models seem to agree over what factors influence their highly accurate prediction.

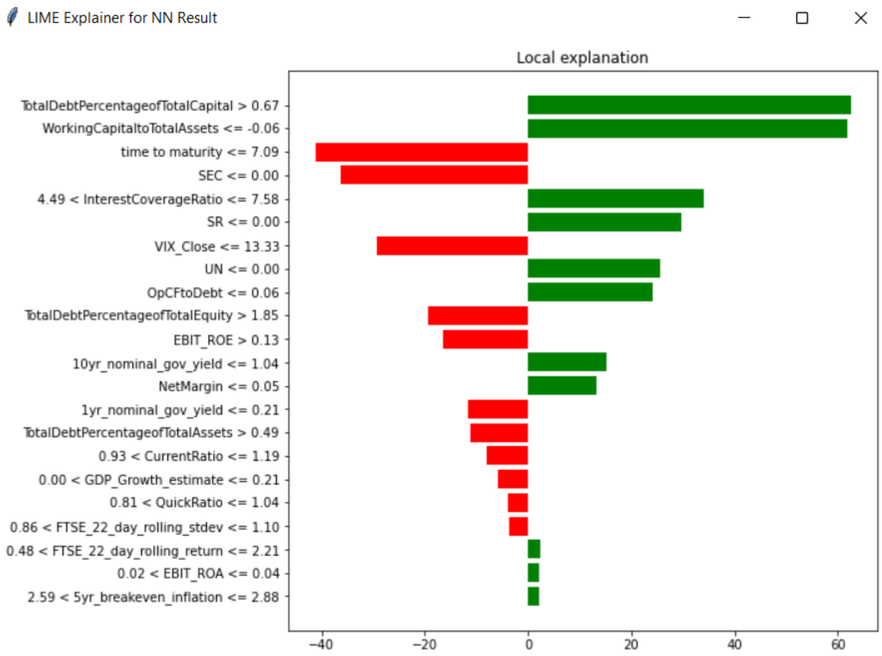


Figure 17: Breakdown of which estimators from Table 3 had the largest contribution to the predicted Z-spread value of 78.30 bps in the NN model. (Produced using the LIME Python Library and captured from the Research Tool developed and outlined in Appendix I)

View Decision Tree

y (score **78.463**) top features

Contribution?	Feature	Value
+150.777	<BIAS>	1.000
+5.200	CurrentRatio	1.090
+3.761	WorkingCapitaltoTotalAssets	-1.545
+2.772	NetMargin	0.009
+2.726	InterestCoverageRatio	5.000
+2.676	SR	0.000
+2.440	QuickRatio	0.860
+2.201	1yr_nominal_gov_yield	0.100
+1.430	EBIT_ROE	0.192
+0.437	UN	0.000
+0.219	TotalDebtPercentageofTotalCapital	0.969
-0.226	GDP_Growth_estimate	0.210
-0.573	SEC	0.000
-1.111	FTSE_22_day_rolling_return	1.737
-2.676	VIX_Close	13.290
-3.374	TotalDebtPercentageofTotalEquity	31.130
-4.212	EBIT_ROA	0.031
-4.631	OpCFtoDebt	0.013
-6.249	5yr_breakeven_inflation	2.675
-7.238	FTSE_22_day_rolling_stdev	0.887
-8.229	10yr_nominal_gov_yield	0.797
-8.398	TotalDebtPercentageofTotalAssets	5.044
-49.258	time to maturity	1.055

Figure 18: Breakdown of which estimators from Table 3 had the largest contribution to the predicted Z-spread value of 78.98 bps in the gradient boosted regression tree model. (Produced using the Eli5 Python Library and captured from the Research Tool developed and outlined in Appendix I)

8.6.2 Result sample 2: gradient boosted regression tree is accurate but neural network is not accurate:

Feature/Estimator	Value
Time to Bond Maturity	7.16
1-Year Benchmark Government Bond Yield	0.56
10-Year Benchmark Government Bond Yield	2.51
5-Year Breakeven Inflation	2.50
ONS GDP Growth Estimate	1.19
FTSE 100 22-day rolling standard deviation	2.14
FTSE 100 22-day rolling return	-4.93
VIX Close Price	42.96
Senior	0.00
Unsecured	0.00
Secured	0.00
Quick Ratio	0.61
Current Ratio	0.63
Interest Coverage Ratio	4.47
Working Capital to Total Assets	-0.07
Total Debt Percentage of Total Assets	0.07
Total Debt Percentage of Total Capital	0.10
Total Debt Percentage of Total Equity	0.11
Net Margin	0.05
Operating Cashflow to Debt	0.67
Return on Equity measured with EBIT	0.01
Return on Assets measured with EBIT	0.01
Actual Z-spread	**164.10**
Neural Network Predicted Z-spread	**230.55**
Gradient Boosted Tree Predicted Z-spread	**170.60**

Table 4: Analysis of actual versus predicted values of Z-spread based on a single set of estimators.

Referencing Figure 19 and Figure 20, we see that both models are strongly positively influenced by market volatility. Both are also positively influenced by the lack of senior subordination. The NN model, however, is strongly positively influenced by the slightly negative working capital to total assets, whereas the regression tree model is strongly negatively influenced by this factor. Interestingly, working capital to total assets is the strongest positive influencing factor in the NN model and the strongest negative influencing factor in the GBRT model, so clearly the models' learning has significantly diverged here. Negative working capital would be associated with low liquidity and therefore higher risk and higher credit spreads. Therefore, oddly, the NN model evaluation of this variable appears to be more correct, despite the NN model having the incorrect prediction overall. The models also disagree on the influence of breakeven inflation at 2.5%, with the NN model being positively influenced by it (approximately +12 bps contribution) and the GBRT model being negatively influenced by it (-12.82 bps contribution). The interest coverage ratio of 4.47 strongly positively influences the NN model prediction but has very little impact on the GBRT model.

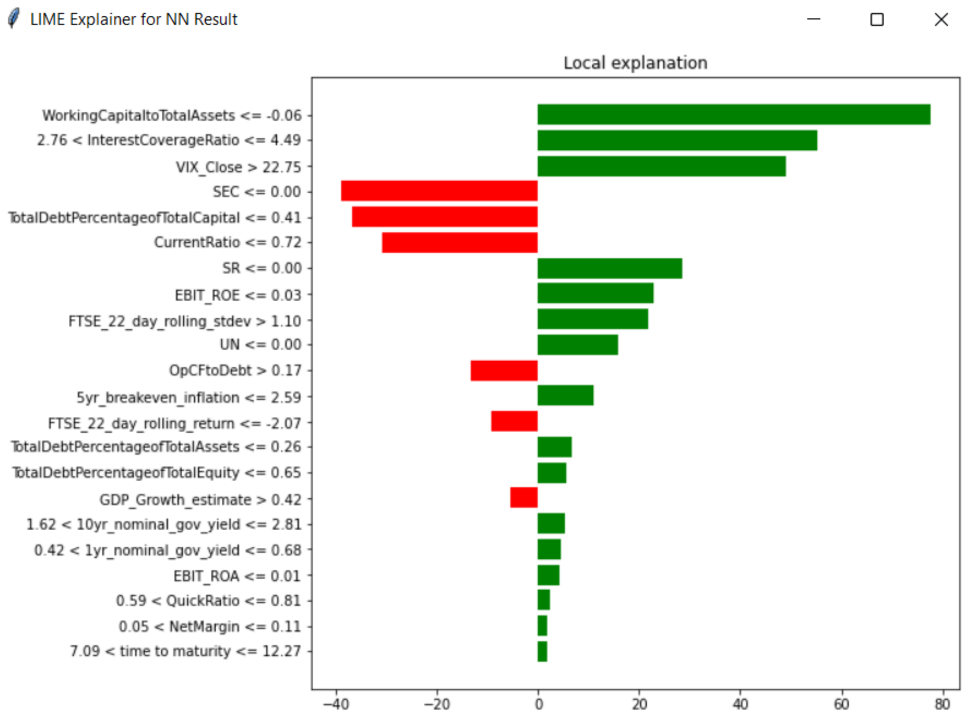


Figure 19: Breakdown of which estimators from Table 4 had the largest contribution to the predicted Z-spread value of 164.10 bps in the NN model. (Produced using the LIME Python Library and captured from the Research Tool developed and outlined in Appendix I)

View Decision Tree

y (score **170.100**) top features

Contribution?	**Feature**	**Value**
+150.777	<BIAS>	1.000
+42.692	FTSE_22_day_rolling_stdev	2.136
+16.510	VIX_Close	42.960
+13.557	SR	0.000
+9.981	EBIT_ROE	0.015
+4.948	GDP_Growth_estimate	1.190
+1.702	10yr_nominal_gov_yield	2.510
+0.766	UN	0.000
+0.601	NetMargin	0.054
+0.080	OpCFtoDebt	0.674
+0.047	time to maturity	7.162
-0.092	InterestCoverageRatio	4.470
-0.251	SEC	0.000
-0.624	QuickRatio	0.610
-2.958	FTSE_22_day_rolling_return	-4.932
-4.294	EBIT_ROA	0.009
-4.390	TotalDebtPercentageofTotalEquity	0.113
-6.530	1yr_nominal_gov_yield	0.559
-7.377	TotalDebtPercentageofTotalCapital	0.102
-9.133	CurrentRatio	0.630
-9.759	TotalDebtPercentageofTotalAssets	0.066
-12.824	5yr_breakeven_inflation	2.502
-13.330	WorkingCapitaltoTotalAssets	-0.067

Figure 20: Breakdown of which estimators from Table 3 had the largest contribution to the predicted Z-spread value of 170.60 bps in the GBRT model. (Produced using the Eli5 Python Library and captured from the Research Tool developed and outlined in Appendix I)

8.6.3 Result sample 3: neural network is accurate but gradient boosted tree model is not:

Feature/Estimator	Value
Time to Bond Maturity	1.30
1-Year Benchmark Government Bond Yield	0.59
10-Year Benchmark Government Bond Yield	3.18
5-Year Breakeven Inflation	2.93
ONS GDP Growth Estimate	0.36
FTSE 100 22-day rolling standard deviation	0.88
FTSE 100 22-day rolling return	-0.70
VIX Close Price	25.25
Senior	0.00
Unsecured	0.00
Secured	0.00
Quick Ratio	0.97
Current Ratio	1.48
Interest Coverage Ratio	0.04
Working Capital to Total Assets	0.22
Total Debt Percentage of Total Assets	0.05
Total Debt Percentage of Total Capital	0.18
Total Debt Percentage of Total Equity	0.22
Net Margin	0.00
Operating Cashflow to Debt	2.17
Return on Equity measured with EBIT	0.04
Return on Assets measured with EBIT	0.01
Actual Z-spread	**822.58**
Neural Network Predicted Z-spread	**781.22**
Gradient Boosted Tree Predicted Z-spread	**521.55**

Table 5: Analysis of actual versus predicted values of Z-spread based on a single set of estimators.

Referencing Figure 20 and Figure 21, we see that the operating cash flow to debt ratio was a strongly positive influencer in the GBRT model (+255.84 bps), whereas it was slightly negative in the NN model (approximately -10 bps). Here, as well, the models have strongly different interpretations of the working capital to total assets ratio of 0.22, with the NN model being strongly negatively influenced by this factor (approximately -40 bps) and the GBRT model being strongly positively influenced by it (+33 bps contribution).

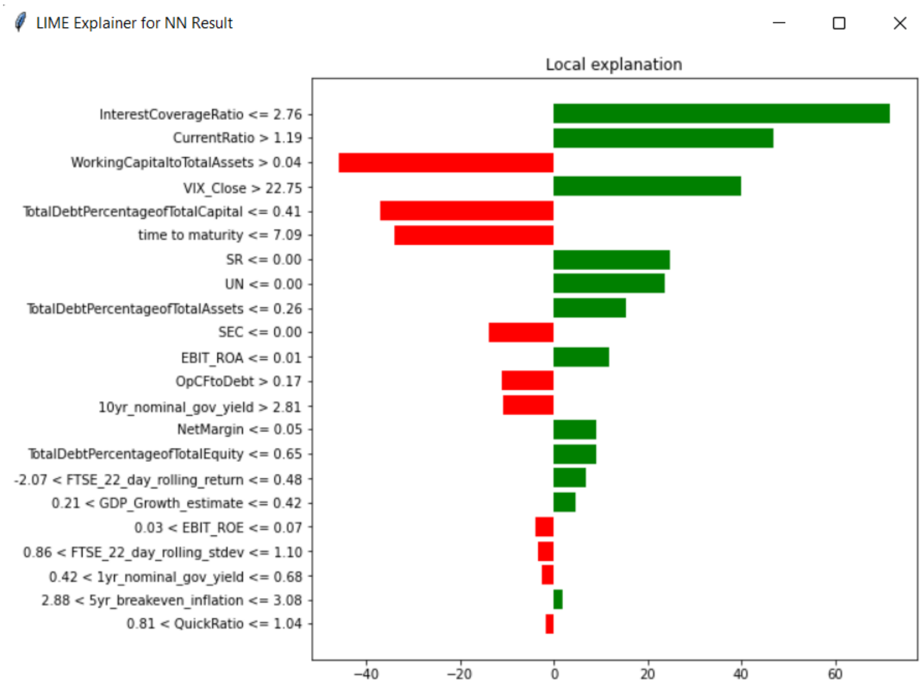
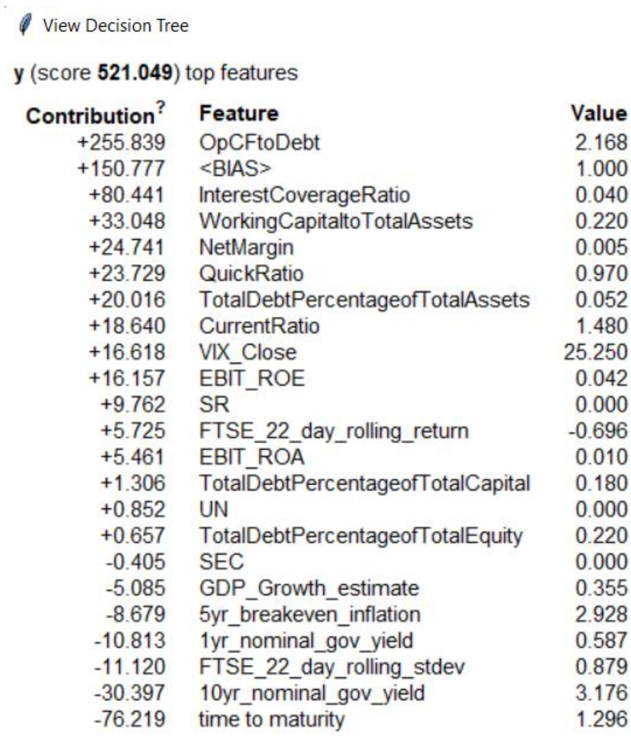


Figure 20: Breakdown of which estimators from Table 5 had the largest contribution to the predicted Z-spread value of 781.22 bps in the Neural Network model. (Produced using the LIME Python Library and captured from the Research Tool developed and outlined in Appendix I)



View Decision Tree

y (score **521.049**) top features

Contribution?	Feature	Value
+255.839	OpCFtoDebt	2.168
+150.777	<BIAS>	1.000
+80.441	InterestCoverageRatio	0.040
+33.048	WorkingCapitaltoTotalAssets	0.220
+24.741	NetMargin	0.005
+23.729	QuickRatio	0.970
+20.016	TotalDebtPercentageofTotalAssets	0.052
+18.640	CurrentRatio	1.480
+16.618	VIX_Close	25.250
+16.157	EBIT_ROE	0.042
+9.762	SR	0.000
+5.725	FTSE_22_day_rolling_return	-0.696
+5.461	EBIT_ROA	0.010
+1.306	TotalDebtPercentageofTotalCapital	0.180
+0.852	UN	0.000
+0.657	TotalDebtPercentageofTotalEquity	0.220
-0.405	SEC	0.000
-5.085	GDP_Growth_estimate	0.355
-8.679	5yr_breakeven_inflation	2.928
-10.813	1yr_nominal_gov_yield	0.587
-11.120	FTSE_22_day_rolling_stdev	0.879
-30.397	10yr_nominal_gov_yield	3.176
-76.219	time to maturity	1.296

Figure 21: Breakdown of which estimators from Table 5 had the largest contribution to the predicted Z-spread value of 521.55 bps in the Gradient Boosted Regression Tree model. (Produced using the Eli5 Python Library and captured from the Research Tool developed and outlined in Appendix I)

8.7 Observations

Interestingly, the time to maturity variable in the three previous predictions always contributes positively to the overall credit spread prediction, as one would expect. That is, when the time to maturity is lower, the predicted credit spread is lower. Therefore, the interpretation is that a short time to maturity is associated with lower risk. This, however, differs from the apparent relationship depicted in the scatter plot in Figure 9 from section 6.5.2 of this report. This suggests that the models may be learning an interaction between the time to maturity variable and other estimators. An interaction between variables would suggest that the model is learning a nonlinear relationship.

8.8 Out-of-sample test results

While the above results were conducted with data that were held out of the ML training data sets, it is useful to test the models on fully out-of-sample data. From an economic perspective, it is useful to see whether the economic conditions that existed during the time frame of the model training still have predictive capability on data outside those economic conditions and timeframes. From a machine learning perspective, it is also useful to ensure that one did not accidentally feed test data into the training data. To conduct this test, we screened for bonds using all the conditions in the methodology section, looking specifically at bonds that were issued after the 2003-2023 sample period. Unfortunately, due to a short research timeframe and a dearth of new bond issues in the GBP market, only one suitable test case was found. Volvo 4.75% (ISIN code XS2606449390). The results below show that the GBRT model outperforms once again in the prediction.

Feature/Estimator	Value
Time to Bond Maturity	3.21
1-Year Benchmark Government Bond Yield	3.93
10-Year Benchmark Government Bond Yield	3.48
5-Year Breakeven Inflation	3.46
ONS GDP Growth Estimate	0.30
FTSE 100 22-day rolling standard deviation	1.32
FTSE 100 22-day rolling return	-3.58
VIX Close Price	18.70
Senior	0.00
Unsecured	0.00
Secured	0.00
Quick Ratio	0.98
Current Ratio	1.28
Interest Coverage Ratio	35.33
Working Capital to Total Assets	0.11
Total Debt Percentage of Total Assets	0.34
Total Debt Percentage of Total Capital	0.56
Total Debt Percentage of Total Equity	1.27
Net Margin	0.08
Operating Cashflow to Debt	0.07
Return on Equity measured with EBIT	0.06
Return on Assets measured with EBIT	0.02
Actual Z-spread	**127.2**
Neural Network Predicted Z-spread	**206.8**
Gradient Boosted Tree Predicted Z-spread	**156.1**

Table 6: Analysis of actual versus predicted values of Z-spread based on a single set of estimators.

Referencing Figure 21 and Figure 22, we see that the interest coverage ratio at 35.55 was the largest negative influencer in both the GBRT model (-29.023 bps) and the NN model (approximately -175 bps). The GBRT model was positively influenced by market data factors such as government yields and breakeven inflation, whereas the NN was not. Both models were positively influenced by senior subordination and the current ratio value of 1.28, which are both sensible interpretations of the data.

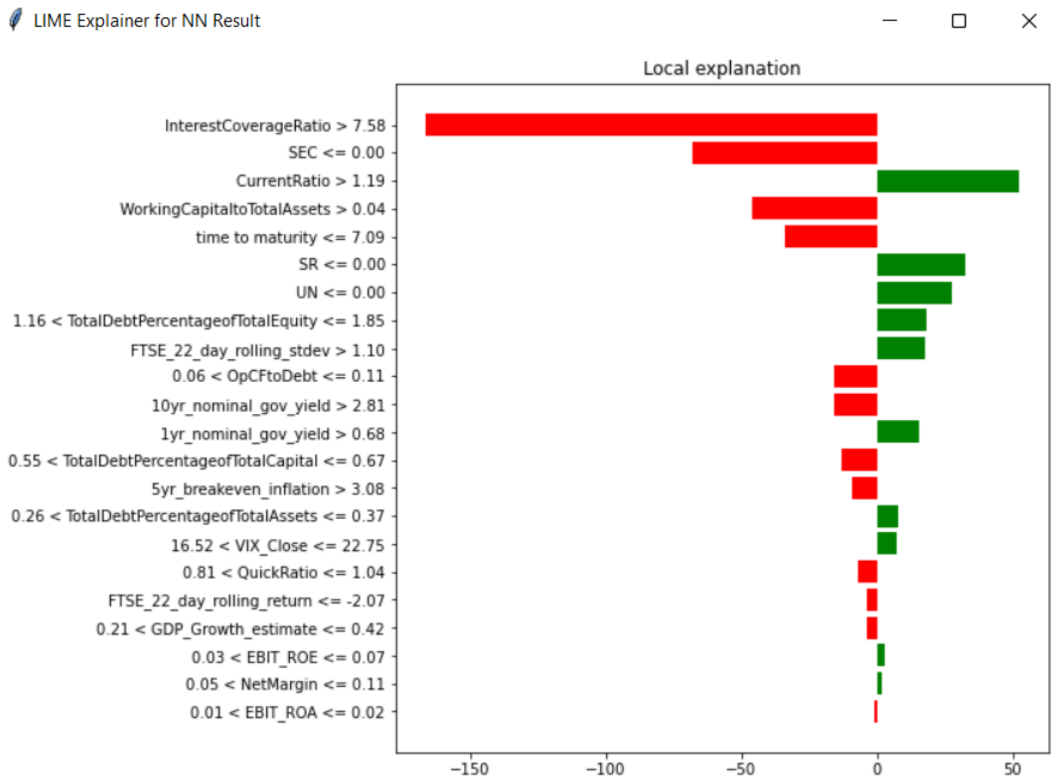


Figure 21: Breakdown of which estimators from Table 6 had the largest contribution to the predicted Z-spread value of 206.8 bps in the Neural Network model. (Produced using the LIME Python Library and captured from the Research Tool developed and outlined in Appendix I)

View Decision Tree

y (score **155.559**) top features

Contribution?	Feature	Value
+150.777	<BIAS>	1.000
+19.267	1yr_nominal_gov_yield	3.929
+13.247	SR	0.000
+12.203	10yr_nominal_gov_yield	3.478
+12.076	OpCFtoDebt	0.069
+10.139	5yr_breakeven_inflation	3.464
+7.407	FTSE_22_day_rolling_stdev	1.319
+2.810	CurrentRatio	1.280
+1.443	VIX_Close	18.700
+0.328	UN	0.000
-0.410	WorkingCapitaltoTotalAssets	0.114
-0.606	FTSE_22_day_rolling_return	-3.578
-0.613	SEC	0.000
-0.718	GDP_Growth_estimate	0.301
-0.913	TotalDebtPercentageofTotalEquity	1.269
-1.372	TotalDebtPercentageofTotalAssets	0.335
-2.479	NetMargin	0.078
-2.603	TotalDebtPercentageofTotalCapital	0.559
-3.596	EBIT_ROE	0.063
-5.174	QuickRatio	0.980
-5.251	EBIT_ROA	0.017
-21.383	time to maturity	3.211
-29.023	InterestCoverageRatio	35.330

Figure 22: Breakdown of which estimators from Table 6 had the largest contribution to the predicted Z-spread value of 156.1 bps in the Gradient Boosted Regression Tree model. (Produced using the Eli5 Python Library and captured from the Research Tool developed and outlined in Appendix I)

9. Conclusion

We set out to assess whether machine learning algorithms can estimate credit risk premiums when trained on market data, financial ratios, and other traditional credit analysis data. We constructed a relatively small set of data consisting of 17,736 rows of training data and 4,434 rows of testing data, composed of 22 estimators selected from financial ratios, stock market data, economic data, and bond idiosyncratic data. We found that both models have impressive predictive capability; however, the gradient boosted regression tree model outperformed the neural network quite significantly, as measured by adjusted R^2. We then explore the trained models to see which estimators were the most valuable in arriving at predictions. To do this, we examine the model weights and feature importance, and we perform a comparison of 4 individual predictions of each model. We find that idiosyncratic bond data such as time to maturity and bond subordination and collateralisation were significant influencers in both models. We also find that individual predictions by each model were making sensible interpretations of financial ratios. The overall outcome suggests that both models can be useful tools for assessing bond risk premiums.

APPENDIX I – Software developed to support research

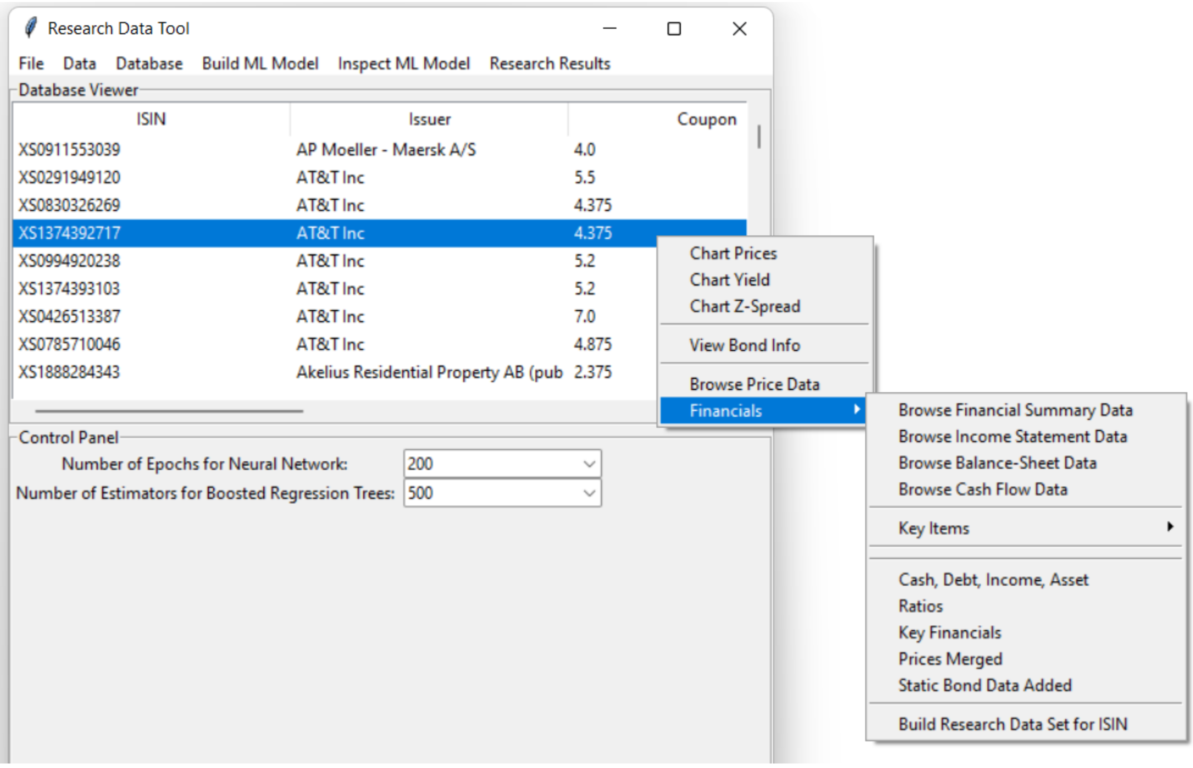
To facilitate this research, a software tool was developed in Python. This tool is available here on GitHub[9]. The tool provides the following functionality:

- Aggregate data in bulk and insert into database for the following data sets:
  - Bond prices
  - Company financial data
  - Bond Master List with bond static data
  - Yield curve data
  - Economic data
- Data exploration tools were used to ensure that data were captured and associated correctly.
- Generate a machine learning training data set on the fly using the data above.
- Import Training data via the clipboard.
- Train A Neural Network or Gradient Boosted Regression Tree model as specified in this dissertation.
- Browse the results of the testing data for either model.
- Save both ML models for later analysis.
- Analyse regression tree predictions with the Eli5 library
- Analysis of Neural Network predictions with the LIME library
- Analyse Regression Tree model F Scores and Weights

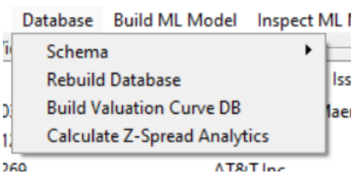
Graphical user interface (GUI)

The interface of the tool presents the list of bonds aggregated from the raw Excel data that were downloaded during the data collection process. The data can be browsed by right-clicking on a single bond and selecting data items. This was created to test whether the SQL Queries and associated Python code were linking bond price data/Z-spreads to the correct underlying financial data on the correct date.

[9] https://github.com/Reneige/EstimatingCreditSpreadsViaMachineLearning



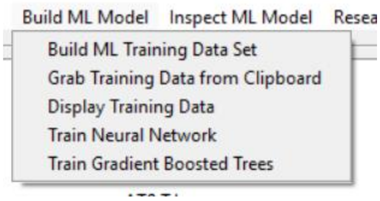
New data can quickly be added to the database by downloading files into the/data/folder and selecting Database > Rebuild Database



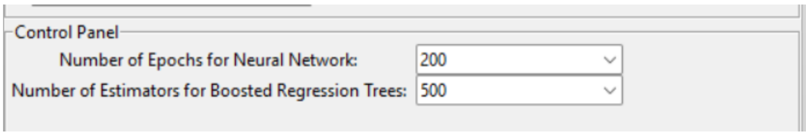
When first running the tool, the valuation curve must be interpolated using the cubic spline algorithm as described in the Methodology section of this report. This is performed by selecting *Database > Build Valuation Curve DB* menu item as depicted above. Afterwards, the bond Z-spreads can be calculated using the *Database > Calculate Z-spread Analytics* menu item.

Building and training a machine learning model

To generate a full research set of data based on the features described in the Requirements section of this paper, click on *Build ML Model > Build ML Training Data Set.* This will take several minutes to run and will perform all the SQL queries required to link the bond prices and static data to the corresponding time-shifted company financial data and economic time series. Once complete, these data will appear in a new Excel document. The data can then be cleaned and processed, however, the analyst would like. The final data set must be in this format: All the features should be on the left (X-Variables), and the labels (Y-Variables) should be the rightmost column of data. In this format, the user must capture the data from Excel in the clipboard by pressing Ctrl-C. Then, we select *Build ML Model > Grab Training Data from Clipboard.* The tool will automatically adjust the input layers of the neural network and the gradient boosted tree model to the number of X-variables captured when doing this. You can also confirm that the X-Variables are correct by selecting *Build ML Model > Display Training Data.* This will not display Labels/Y-Variables, however, as they have already been sliced out of the data set at this point.



You can then select Number of Epochs or Number of Estimators depending on whether you are training a Neural Network or a Boosted Regression Tree model, respectively.

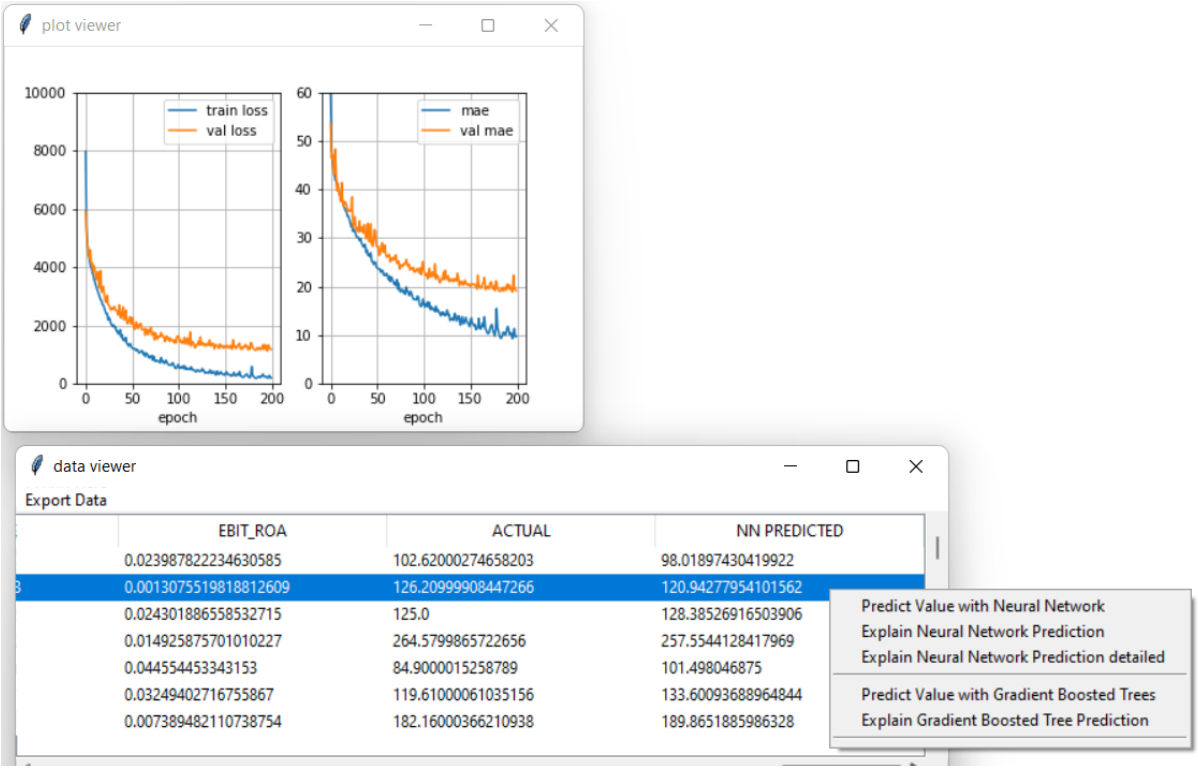


Once set, select *Build ML Model > Train …* to train your respective model. The training status should be displayed in the console or shell window.

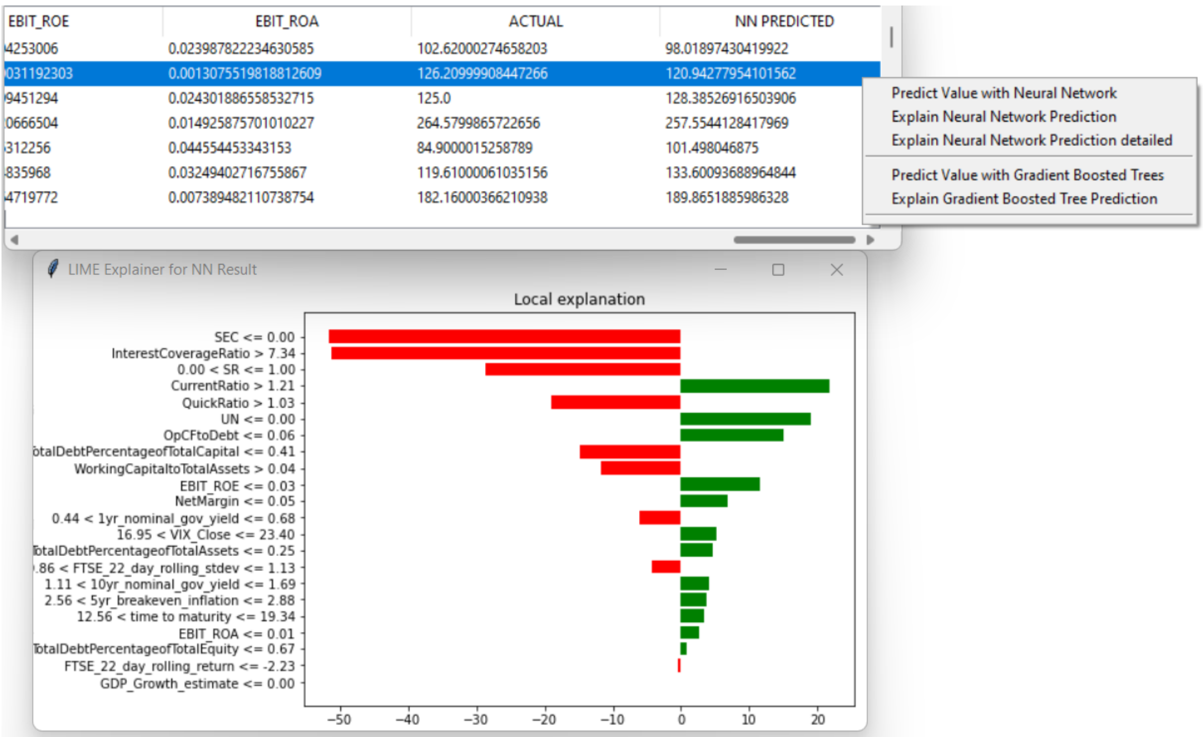
Making predictions and exploring results of trained models

Once the training is complete, the tool will automatically pop up both a chart with the model learning curve and a data viewer showing the results of the test set of data. The final two columns of the data viewer show the actual versus predicted Y-Variables (Labels), whereas all the other columns are the Estimators/X-Variables.

Here, the user can now right-click and explore individual predictions using LIME for neural networks or Eli5 for gradient boosted trees:

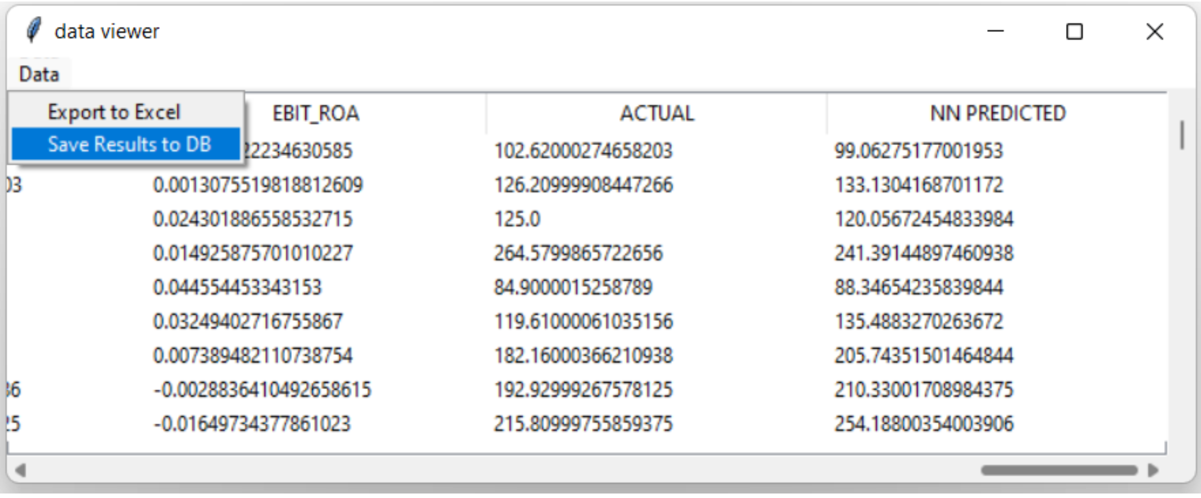


Here, for example, we have selected the explanation for the 120 bps Z-spread predicted (against 127 bps Actual) with the trained Neural Network. The results window shows that the major element contributing negatively to the Z-spread (decreasing credit risk/return) is that it is a senior unsecured bond with an interest coverage ratio above 7.35. The major items contributing positively to Z-spread (increasing credit risk/return) are the low current ratio and the low operating cash flow to debt ratio.

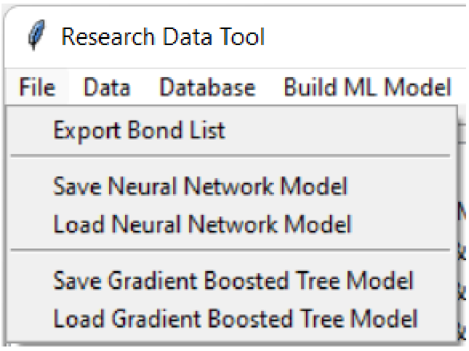


Exporting / saving results and models

The results can also be saved to the database for future exploration or exported to Excel for further analysis.



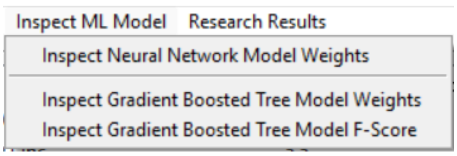
If you must also save the trained ML models if you would like to use them in the future to make predictions or to explore predictions.



Note that neural networks are saved in a folder and contain various files and subfolders. When saving a neural network, you must create a folder first to save all these data. Gradient boosted trees, on the other hand, are saved as a single JSON file.

Inspecting model weights

Once you have a trained model or have loaded a saved model, you can inspect the weights and F scores to get an idea as to what features the model is finding most important when making predictions:



APPENDIX II – Artificial neural network model[10]

Let X be the input data, with shape (n, d), where n is the number of samples and d is the number of input features.

Let Y be the output data, with shape (n, k), where k is the number of output classes or continuous values.

Let W be a set of weights with shape (m, d), where m is the number of nodes in the current layer and d is the number of nodes in the previous layer.

Let b be a set of biases with shape (m), where m is the number of nodes in the current layer.

Let f be the activation function applied to the weighted sum of the inputs to a node.

Let L be the loss function used to measure the difference between the predicted output and the true output.

Let η be the learning rate, which determines the step size of the weight and bias updates during training.

Step 1: Forward propagation:

The output of the first hidden layer can be calculated as follows:

$$Z_1 = XW_1 + b_1$$

where Z1 has shape (n, m1), where m1 is the number of nodes in the first hidden layer. We then apply the activation function f to Z1 to obtain the output A1:

$$A_1 = f(Z_1)$$

We repeat this process for each subsequent hidden layer and the output layer. The output of the final layer is the predicted output Yhat:

$$Z_L = A_{L-1} W_L + b_L$$

$$Yhat = f(ZL)$$

Step 2: Calculate Loss:

We use the loss function L to calculate the difference between the predicted output Yhat and the true output Y:

[10] OpenAI ChatGPT was used to assist writing with this section.

L(Y,Yhat)

A typical loss function is the mean squared error:

*MSE = (1/N) * sum[(yi - fi)^2]*

Step 3: Backpropagation:

We now use backpropagation to update the weights and biases in the MLP. We start by calculating the gradient of the loss function with respect to the weights and biases in the output layer:

We repeat this process for each subsequent hidden layer and the output layer. The output of the final layer is the predicted output Yhat

$$\frac{\partial L}{\partial W_L} = A_{L-1}^T \frac{\partial L}{\partial Z_L}$$

$$\frac{\partial L}{\partial b_L} = \sum_{i=1}^{n} \frac{\partial L}{\partial Z_L^{(i)}}$$

where $\partial Z_L^{(i)}$ is the ith row of ∂L.

We can then use these gradients to update the weights and biases in the output layer:

$$W_L = W_L - \eta \frac{\partial L}{\partial W_L}$$

$$b_L = b_L - \eta \frac{\partial L}{\partial b_L}$$

Apply the chain rule calculating the gradient of the loss function with respect to the weights and biases for each previous layer:

$$\frac{\partial L}{\partial Z_{L-1}} = \frac{\partial L}{\partial A_{L-1}} \frac{\partial A_{L-1}}{\partial Z_{L-1}} = \frac{\partial L}{\partial Z_L} W_L^T \frac{\partial A_{L-1}}{\partial Z_{L-1}}$$

$$\frac{\partial L}{\partial W_{L-1}} = A_{L-2}^T \frac{\partial L}{\partial Z_{L-1}}$$

APPENDIX III – Gradient boosted regression trees[11]

The process of calculating gradient boosted regression trees is as follows.

Begin by defining an objective function (loss function). The objective function measures the accuracy of the model in fitting the training data. For regression problems, the most common objective function is the MSE (mean squared error), defined as:

$$\frac{1}{N}\sum(yi - fi)^2$$

N = the number of training instances

yi = the i-th instance of the actual value (i.e., label)

fi = the i-th instance of the predicted value (i.e., label)

Steps of the model:

1. Start by Initialising the model with a constant value. Usually, the mean of the target variable.
2. Fit a regression tree to the residuals of the current model. The residuals are the differences between the actual labels and the predictions of the current model. The goal of the tree is to learn the pattern in the residuals that the current model has not captured.
3. Add the new tree to the current model by updating the predicted values. Each new tree provides a correction to the previous predictions, which improves the overall performance of the model.
4. Repeat steps 2 and 3 until the objective function reaches a minimum or a maximum number of trees have been added.

Residual are defined as follows:

r = y - f

y = the vector of actual labels and

f = the vector of predicted labels from the current XBGR model.

We want to fit a regression tree to the residuals, which means finding the optimal splits that minimise the MSE. Define the MSE of a leaf node as:

$$\frac{1}{Nnode}\sum(yi - fi)^2$$

where N_node is the number of instances in the leaf node.

[11] OpenAI ChatGPT was used to assist writing with this section.

We can then define the total MSE of the tree as the sum of the MSEs of all the leaf nodes:

MSE_tree = sum[MSE_node]

The goal of fitting the tree is to find the optimal splits that minimise the MSE_tree. The algorithm searches for the optimal splits by trying all possible splits for each feature and selecting the one that results in the greatest reduction in MSE_tree. The reduction in MSE_tree from a split can be computed as:

MSE_reduction = MSE_parent - (MSE_left + MSE_right)

where MSE_parent is the MSE of the parent node before the split, MSE_left is the MSE of the left child node after the split, and MSE_right is the MSE of the right child node after the split.

Once we have found the optimal splits for all the features, we can build the tree by recursively partitioning the data based on the splits until we reach the maximum depth or the minimum number of instances per leaf node.

Finally, to add the new tree to the current model, we multiply its predictions by a learning rate (gamma) and add it to the previous predictions:

f_new = f + gamma * f_tree

f_new = the new vector of predicted labels

f_tree = the vector of predicted labels from the new tree

gamma = the learning rate.

The learning rate controls the contribution of each new tree to the final prediction and can be tuned to balance the bias-variance trade-off. A smaller learning rate will result in a more conservative model with lower variance but higher bias, while a larger learning rate will result in a more aggressive model with higher variance but lower bias.

APPENDIX IV – Sample tree from gradient boosted regression tree model

The below is booster 999 from the trained final model packaged within the Research Software outlined in Appendix I.

```
0: [Time to maturity<98.4219208] yes=1, no=2, missing=1
   1:[Time to maturity<94.6712341] yes=3,no=4,missing=3
      3:[Time to maturity<44.8342476] yes=7,no=8,missing=7
         7:[Time to maturity<40.0232849] yes=13,no=14,missing=13
            13:[Time To Maturity<39.436985] yes=25,no=26,missing=25
               25:[Time to maturity<38.4095879] yes=43,no=44,missing=43
                  43:leaf=0.00039431543
                  44:leaf=-0.058167398
               26:[1Year Gov Yield5<0.254609704] yes=45,no=46,missing=45
                  45:leaf=0.249500379
                  46:leaf=0.0150101231
            14:[1Year Gov Yield1<0.480000019] yes=27,no=28,missing=27
               27:[VIX Price<16.4500008] yes=47,no=48,missing=47
                  47:leaf=0.0645071417
                  48:leaf=-0.0922083482
               28:[Inflation<3.01444864] yes=49,no=50,missing=49
                  49:leaf=-0.213276654
                  50:leaf=-0.0572059937
         8:[VIX Price<25.0400009] yes=15,no=16,missing=15
            15:[1Year Gov Yield<1.05748391] yes=29,no=30,missing=29
               29:[FTSE Return<1.83365107] yes=51,no=52,missing=51
                  51:leaf=0.0615691543
                  52:leaf=0.158199087
               30:[FTSE Volatility<1.00538588] yes=53,no=54,missing=53
                  53:leaf=0.13994132
                  54:leaf=-0.119572476
            16:[FTSE Volatility<0.953621507] yes=31,no=32,missing=31
               31:[1Year Gov Yield<0.62545526] yes=55,no=56,missing=55
                  55:leaf=-0.343266755
                  56:leaf=-0.0778465271
               32:[1Year Gov Yield8<0.0587370694] yes=57,no=58,missing=57
                  57:leaf=-0.0825040415
                  58:leaf=0.0560428984
      4:[VIX Price<13.3099995] yes=9,no=10,missing=9
         9:[FTSE Return<-1.02018023] yes=17,no=18,missing=17
            17:[Inflation<2.89665937] yes=33,no=34,missing=33
               33:leaf=0.00735702505
               34:[Time to maturity<96.7136993] yes=59,no=60,missing=59
```

```
                    59:leaf=-0.0427884422
                    60:leaf=-0.134996042
              18:[1Year Gov Yield<0.185819268] yes=35,no=36,missing=35
                 35:leaf=0.0331206657
                 36:leaf=0.109819032
           10:[Inflation<2.30027294] yes=19,no=20,missing=19
              19:[Time To Maturity<97.838356] yes=37,no=38,missing=37
                 37:leaf=0.116821289
                 38:leaf=-0.0767501816
              20:[Inflation<3.15124989] yes=39,no=40,missing=39
                 39:[FTSE Volatility<0.860798717] yes=61,no=62,missing=61
                    61:leaf=-0.205633193
                    62:leaf=-0.0840486884
                 40:[Time to maturity<95.0452118] yes=63,no=64,missing=63
                    63:leaf=-0.0341316238
                    64:leaf=0.019428635
     2:[1Year Gov Yield<0.291954875] yes=5,no=6,missing=5
        5:leaf=-0.0439495109
        6:[FTSE Volatility<0.503516257] yes=11,no=12,missing=11
           11:[1Year Gov Yield<0.540818393] yes=21,no=22,missing=21
              21:leaf=0.20170784
              22:leaf=0.0444509499
           12:[FTSE Volatility<0.528873682] yes=23,no=24,missing=23
              23:leaf=-0.0107620237
              24:[1Year Gov Yield<0.532905877] yes=41,no=42,missing=41
                 41:[1Year Gov Yield<0.369861037] yes=65,no=66,missing=65
                    65:leaf=0.0395910889
                    66:leaf=0.101934157
                 42:leaf=0.023293877
```

APPENDIX V – Ethics checklist

UNIVERSITY OF BATH

Department of Computer Science

12-Point Ethics Checklist for UG and MSc Projects

Student	René Alby
Academic Year or Project Title	2023
Supervisor	Raghubir Singh

This form must be attached to the dissertation as an appendix.

Does your project involve people for the collection of data other than you and your supervisor(s)? NO

If the answer to the previous question is YES, you need to answer the following questions, otherwise you can ignore them.

This document describes the 12 issues that need to be considered carefully before students or staff involve other people ('participants' or 'volunteers') for the collection of information as part of their project or research. Replace the text beneath each question with a statement of how you address the issue in your project.

1. *Will you prepare a Participant Information Sheet for volunteers?* YES / NO

   This means telling someone enough in advance so that they can understand what is involved and why – it is what makes informed consent informed.

2. *Will the participants be informed that they could withdraw at any time?* YES / NO

   All participants have the right to withdraw at any time during the investigation, and to withdraw their data up to the point at which it is anonymised. They should be told this in the briefing script.

3. *Will there be any intentional deception of the participants?* YES / NO

   Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.

4. *Will participants be de-briefed?* YES / NO

   The investigator must provide the participants with sufficient information in the debriefing to enable them to understand the nature of the investigation. This phase might wait until after the study is completed where this is necessary to protect the integrity of the study.

5. Will participants voluntarily give informed consent? YES / NO

Participants MUST consent before taking part in the study, informed by the briefing sheet. Participants should give their consent explicitly and in a form that is persistent –e.g. signing a form or sending an email. Signed consent forms should be kept by the supervisor after the study is complete. If your data collection is entirely anonymous and does not include collection of personal data you do not need to collect a signature. Instead, you should include a checkbox, which must be checked by the participant to indicate that informed consent has been given.

6. Will the participants be exposed to any risks greater than those encountered in their normal work life (e.g., through the use of non-standard equipment)? YES / NO

Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life.

7. Will you be offering any incentive to the participants? YES / NO

The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.

8. Will you be in a position of authority or influence over any of your participants? YES / NO

A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.

9. Will any of your participants be under the age of 16? YES / NO

Parental consent is required for participants under the age of 16.

10. Will any of your participants have an impairment that will limit Their understanding or communication? YES / NO

Additional consent is required for participants with impairments.

11. Will the participants be informed of your contact details? YES / NO

All participants must be able to contact the investigator after the investigation. They should be given the details of the Supervisor as part of the debriefing.

12. Will you have a data management plan for all recorded data? YES / NO

Personal data is anything which could be used to identify a person, or which can be related to an identifiable person. All personal data (hard copy and/or soft copy) should be anonymized (with the exception of consent forms) and stored securely on university servers (not the cloud).

GLOSSARY OF FINACIAL TERMS

Asset: two meanings: 1. A synonym for any investment. 2. Valuable property owned by a company.

Asset Valuation: Estimating the amount of return an investment should provide an investor in order to compensate for the perceived risks of holding that investment.

Credit: Here used as a synonym for corporate bond and the riskiness of corporate debt.

Credit Spread: The difference in yield between a risky bond and a risk-free government bond of equal maturity and of the same currency.

EBIT: Earnings before interest and taxes. A measure of business profitability before costs of servicing debt and taxes.

EBIDTA: Earnings before interest, depreciation, taxes, and amortisation. A measure of business profitability before costs of servicing debt and taxes, and prior to some of the potential distortions of modern accounting methods.

Enterprise Value: The sum of the market value of a company's debt and equity. Viewed as the total value of a business, or the cost of purchasing an entire business and therefore taking ownership of the company's debt obligations.

FFO: Funds from operations. This represents the cash a business generates from conducting its primary business operations.

Fundamental Valuation: Analysing the financial health of a company to arrive at a valuation for the investments they issue.

Government Spot Curve: Sometimes referred to as the Zero-Coupon Curve, a spot curve plots the yield to maturity across government issued zero-coupon bonds of various maturities, with the y-axis depicting the yield and the x-axis depicting the maturity date.

Liquidity: two meanings: 1. The ease with which an investor can sell (liquidate) an investment. 2. The amount of cash or cash-like assets a company has at its disposal to pay its liabilities.

Leverage: The amount of debt a company holds.

Maturity: The expected date a bond will be repaid to investors

Options: A financial instrument or a feature of a bond that allows the holder to buy or sell an investment at a given price in the future.

Option Adjusted Spread (OAS): A single discount rate added to each point of the risk-free term structure, which when used to discount a risky bond's cash flows, makes the discounted cash flows equal to the observed market price of the bond after adjusting the price for any embedded options. The OAS can be viewed as a measure of a bond's credit spread; however, it also includes other risks such as liquidity risks. In the absence of embedded options, the OAS is equivalent to the Z-spread.

Risk Premium: The percent amount an investment will compensate an investor compared to a risk-free investment.

Solvency: The risk that a company will not be able to repay its debts.

Systematic Risk: Risk variables that impact the entire universe of an investment and therefore are inherent to the entire market.

Unsystematic Risk: Risk that is specific to an individual investment or firm.

Vanilla Bonds: Bonds that contain no special price features suck as sinking funds or embedded options.

Yield: Short for Yield-to-Maturity, which is the expected return a bond should provide an investor if held to the maturity date of the bond.

REFERENCES

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

Babbel, D. and Zenios, S., 1992. Pitfalls in the Analysis of Option-Adjusted Spreads. *Financial Analysts Journal*, 48(4), pp.65-69.

Bhar, R., and Handzic, N., 2010. A multifactor model of credit spreads. *Asia-Pacific Financial Markets*, 18(1), pp.105–127.

Bianchi, D., Büchner, M. and Tamoni, A., 2020. Bond Risk Premiums with Machine Learning. *The Review of Financial Studies*, 34(2), pp.1046-1089.

Beaver, W.H., 1966. Financial ratios as predictors of failure. *Journal of Accounting Research*, 4, p.71.

Buczynski, W., Cuzzolin, F. and Sahakian, B., 2021. A review of machine learning experiments in equity investment decision-making: why most published research findings do not live up to their promise in real life. *International Journal of Data Science and Analytics*, 11(3), pp.221-242.

Chen, T., and Guestrin, C., 2016. XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Davies, A., 2008. Credit spread determinants: An 85 year perspective. *Journal of Financial Markets*, 11(2), pp.180–197.

Deakin, E.B., 1972. A discriminant analysis of predictors of business failure. *Journal of Accounting Research*, 10(1), p.167.

Elton, E.J., Gruber, M.J., Agrawal, D., and Mann, C., 2001. Explaining the rate spread on corporate bonds. *The Journal of Finance*, 56(1), pp.247–277.

Fama, E., 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), p.383.

Fisher, L., 1959. Determinants of Risk Premiums on Corporate Bonds. Journal of Political Economy, vol. 67, no. 3, University of Chicago Press, 1959, pp. 217–237.

Ghysels, E., Santa-Clara, P. and Valkanov, R., 2005. There is a risk-return trade-off after all. *Journal of Financial Economics*, 76(3), pp.509-548.

Golbayani, P., Florescu, I., and Chatterjee, R., 2020. A comparative study of forecasting corporate credit ratings using neural networks, support vector machines, and decision trees. *The North American Journal of Economics and Finance*, 54, p.101251.

Gu, S., Kelly, B. and Xiu, D., 2018. Empirical Asset Pricing Via Machine Learning. Chicago Booth Research Paper No. 18-04, 31st Australasian Finance and Banking Conference 2018, Yale ICF Working Paper No. 2018-09, Available at SSRN: https://ssrn.com/abstract=3159577

Harvey, C., 2021. Be Skeptical of Asset Management Research. *SSRN Electronic Journal*,. Available at Available at SSRN: https://ssrn.com/abstract=3906277

Henrique, B.M., Sobreiro, V.A., and Kimura, H., 2019. Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications, 124, pp.226–251.

Huang, J., Chai, J., and Cho, S., 2020. Deep Learning in Finance and banking: A literature review and classification. Frontiers of Business Research in China, 14(1).

Kao, D.-L., 2000. Estimating and pricing credit risk: An overview. *Financial Analysts Journal*, 56(4), pp.50–66.

Kim, J.-M., Kim, D.H., and Jung, H., 2021. Applications of machine learning for corporate bond yield spread forecasting. *The North American Journal of Economics and Finance*, 58, p.101540.

Korobov M., Lopuhin K, 2021, *Eli5* 0.11 [computer program]. Available From https://github.com/eli5-org/eli5.

Kryzanowski, L., Galler, M. and Wright, D., 1993. Using Artificial Neural Networks to Pick Stocks. *Financial Analysts Journal*, 49(4), pp.21-27.

Li, X.-L., Li, X., and Si, D.-K., 2020. Asymmetric determinants of corporate bond credit spreads in China: Evidence from a nonlinear ARDL model. *The North American Journal of Economics and Finance*, 52, p.101109.

Liu, X., Zhou, R., Qi, D., and Xiong, Y., 2022. A novel methodology for credit spread prediction: Depth-gated recurrent neural network with self-attention mechanism. *Mathematical Problems in Engineering*, 2022, pp.1–12.

McCulloch, J.H., 1975. The tax-adjusted yield curve. *The Journal of Finance*, 30(3), pp.811–830.

Merton, R.C, 1973. An Intertemporal Capital Asset Pricing Model. *Econometrica: Journal of the Econometric Society*, 41, 867-887

Ng, A., Bagul, A., Shyu, E., Ladwig, G., n.d. Advanced learning algorithms. [online] *Coursera*. Available from: https://www.coursera.org/learn/advanced-learning-algorithms [Accessed 10 April 2023].

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

OpenAI, ChatGPT, 24 Jan. 2023, https://chat.openai.com/

Ribeiro, M., Singh, S., and Guestrin, C., 2016. “Why should I trust you?”: Explaining the predictions of any classifier. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*: Demonstrations.

Swales, G.S., and Yoon, Y., 1992. Applying artificial neural networks to investment analysis. *Financial Analysts Journal*, 48(5), pp.78–80.

Wong, F.S., Wang, P.Z., Goh, T.H., and Quek, B.K., 1992. Fuzzy Neural Systems for stock selection. *Financial Analysts Journal*, 48(1), pp.47–52.

Zmijewski, M.E., 1984. Methodological issues related to the estimation of financial distress prediction models. *Journal of Accounting Research*, 22, p.59.