Predicting Corporate Bankruptcy using Ensemble Learning

Ishan Rohan Parikh

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2020

Abstract

Bankruptcy prediction has been a subject of interest for almost a century and leading to intensive research from academics and practitioners. The aim of predicting financial bankruptcy is to develop a predictive model that combines various econometric measures and allows us to foresee the financial condition of a firm.

In this study, we use ensemble machine learning models to predict bankruptcy on a one-year horizon. We use Wharton Research Data Services to provide financial data from 4000 listed Japanese corporations. Using financial data from 2000 to 2018, our study uses 64 financial ratios chosen from seminal research done in the past. Comparing results with the Altman's seminal study [5], our research leads to a substantial improvement in prediction performance using different imputation techniques to estimate missing data and by oversampling minority class data using the Synthetic Minority Oversampling Technique (SMOTE). We compare the performance of machine learning and ensemble learning techniques, specifically Decision Trees, Random Forests, Bagging, and Extreme Gradient Boosting (XGBoost).

We find that using Multiple Imputation by Chained Equations as an imputation technique along with XGBoost as an ensemble model, outperforms the other models by achieving the highest AUC and F1 score of 0.856 and 0.798 respectively. Attaining these results substantiates our contribution to the ongoing discussions and debates about the superiority of computational methods over traditional statistical techniques. We also put forward a novel contribution towards prediction of bankruptcies of Japanese corporations as combining a ten-year rolling window model along with MICE imputations and XGBoost has not been done in the past.

Acknowledgements

I would like to take this opportunity to thank my supervisor, Dr. Tiejun Ma, for providing me with the necessary financial data for this project and for guiding me through the course of this dissertation. Dr. Ma encouraged me to do a very thorough study of the existing research, which helped me in exploring newer ideas.

I also want to acknowledge the contribution of Mr. Luo Chang, for his limitless patience and mentorship during my project. His advice on how to do effective research saved me countless hours, which was then invested in documenting this project.

My dissertation submission is amidst a worldwide COVID-19 pandemic being suffered by millions across the world. At the time of a nationwide lockdown, away from University, I am eternally grateful to my friend Liam and his parents Mair and Peter for extending their home and heart to me.

Finally, to my family, for giving me this opportunity to study at the University of Edinburgh and for their constant support and encouragement to pursue all the opportunities in front of me and always inspiring me to do my best by going beyond the defined lines to raise the bar.

Contents

Chapter 1

Introduction

Predicting corporate bankruptcy is important because business failures have wide ranging repercussions. It affects internal and external stakeholders of the company such as management, employees, shareholders, creditors, suppliers, clients, governments and global economies [95]. Corporate bankruptcy can destabilise the economic system by increasing the unemployment rate, depriving investors and creditors of livelihood and contributing to a higher the crime rate [72]. On a large scale, it can lead to negative micro and macro-economic consequences affecting multiple stakeholders and potentially cause social dislocation, economic downturns and recessions [54].

Beaver [11] and Altman [5] were pioneers in predicting bankruptcies and their work is still referenced and valid today. As there are no mature theories of corporate bankruptcy, most studies in corporate bankruptcy are largely been based on iterative, trial and error processes that involve selecting features (such as $\frac{Working\ Capital}{Total\ Capital}$, $\frac{Retained\ Earnings}{Total\ Assets}$, etc.) and predictive models [117].

Early statistical methods applied in corporate bankruptcy prediction utilised Linear Discriminant Analysis (LDA) and Logistic Regression Analysis (LRA) [65]. The problem with applying these statistical techniques to bankruptcy prediction is that the assumptions for independent variables are frequently violated in the practice, which makes these techniques theoretically invalid for finite samples [92].

To overcome these limitations, intelligent techniques that do not assume certain data distributions and automatically extract knowledge from training samples have been developed actively in the field of machine learning [3, 64]. Machine learning models like Support Vector Machines (SVMs), Neural Networks (NNs) and ensemble models have been reported to be more accurate than the traditional statistical methods as shown in [9].

Although numerous previous studies concluded that machine learning techniques are superior to statistical models, it has been argued that no ’single’ classifier can produce the best results. Results also tend to vary when models are tested on data from different countries and economies.

Recently, integrating multiple predictors into an aggregated output, i.e., ensemble methods, have been demonstrated to be an efficient strategy in predicting bankruptcies, especially when the component predictors have different structures that lead to unique prediction errors [17]. Moreover, latest studies have shown that such ensemble techniques are better performers than single intelligent techniques in financial distress prediction [33, 100].

With the discovery of Extreme Gradient Boosting (XGBoost) in 2016, ensemble learning has made major advances in solving real world data driven problems that have resisted the determined attempts of the artificial intelligence community for many years. It has turned out to be very successful at discovering intricate structures in high-dimensional data and is, therefore applicable to many domains of science, business and government. In addition, ensemble learning has been predicted to have many more successes in the near future, largely because they require very little engineering by hand and can be run very efficiently and achieve credible performance [30].

Financial ratios are relationships determined from a company's financial information and are used for comparison purposes. Some examples include measures such as return on investment (ROI), return on assets (ROA), and debt-to-equity, etc. These ratios are the result of dividing one account balance or financial measurement with another. They describe the financial health of companies and measure different aspects of a company's performance. Since these ratios are often characterised by high variance, they often tend to pose as a problem for machine learning algorithms. This problem is also difficult to overcome when data is normalised or standardised. Ensemble methods that use tree-based learners take into account the order of feature values, not the values themselves. Therefore, they are resistant to the large variance observed in the economic indicators.

Ensemble models are among the most effective methods for improving recognition of the minority class. This makes ensembles robust to class imbalance and reduces overfitting. They can also employ preprocessing methods before learning component classifiers or embed a cost-sensitive framework in the ensemble learning process [14].

1.1 Motivation and Goals

For financial institutions, the ability to accurately predict in advance any possible business failures is crucial, as incorrect decisions can have direct and severe economical consequences. Our objective is to help address this problem by building a highly effective bankruptcy prediction model and test its performance on financial data from listed Japanese companies. We propose the use of ensemble learning to build these bankruptcy prediction models as ensembles have emerged as a powerful tool that leverage a pool of individual (base) learners and produce highly accurate classification models. Practical investigations have demonstrated that ensembles with tree-based base learners generally outperform stand-alone prediction methods in most credit risk and bankruptcy prediction problems [112, 36, 4, 100]. Research has also shown that using ensemble models to predict bankruptcy has been able to outperform complex models like neural networks [56].

1.2 Work Completed and Personal Contributions

The Asian economy is often ignored in financial research - the paucity of available resources is representative of this. As Japan has the third-highest market capitalisation in the world, we chose to carry out our research on the Japanese economy. Consequently, our paper has two main contributions: First is the preparation of a cleaned, aligned and pre-processed Japanese financial dataset that can be made available for future studies to carry out further research. Second, to the best of our knowledge, this paper puts forth a novel contribution to the field of bankruptcy prediction by using a ten-year rolling window on Japanese financial data and then predict bankruptcy on a one-year horizon. We compare three different imputation techniques along with oversampling the minority class and then use ensemble learning to produce results that are in line with similar research on North American companies [9, 61]. The novelty lies in combining a ten-year rolling window along with MICE imputation and XGBoost, as this has never been done on financial data from Japan.

We aim to address the gap in literature by comparing the performance of different data imputation[1] techniques that could be used to estimate missing data. Additionally, efforts have been made to account for the class imbalance by including a stage of over sampling of the minority class. Machine learning models have been implemented and their results have been critiqued and compared to produce the best prediction model for the available dataset. A case has been made for the ability of ensemble models to prevent overfitting by achieving high F1-Scores and AUC values under the ROC curve.

Henceforth the paper is organised as follows: In chapter 2, we conduct a background review of related financial and machine learning research. In chapter 3, we describe and the Japanese financial dataset, its relevance in being representative of the modern markets and also summarise some key statistics and features that provide a general overview on the data. We also discuss the data pre-processing steps that have been used. In chapter 4, we outline our experiment methodologies and describe the evaluation metrics that have been used. In chapter 5, we present our empirical results. In chapter 6 we discuss the results obtained in the light of existing literature and put forward our best performing model. We conclude in chapter 7 with a summary of the main findings, the implications of this work and suggested areas for further research.

[1] Imputed value is an assumed value given to an item when the actual value is not known or available. Imputed values are a logical or implicit value for an item or time set, wherein a ”true” value has yet to be ascertained. It would be a best guess estimate, to accurately estimate a larger set of values or series of data points.

Chapter 2

Background and Literature Review

At the time of writing in 2020, with a US recession[1] on the horizon (as shown by data produced by Bloomberg in Figure 2.1), the need for risk assessment is increasingly important to avoid repeating the carnage caused by the 2008 financial crisis. While in the past the small and medium (and micro) companies had higher propensity of going bankrupt, more recently we have also seen an uptick in bankruptcies of large global firms. A few examples are the big retail companies such as Toys“R”Us, Debenhams, Forever 21, JC Penney and Sears. The travel agency Thomas Cook and the low-cost airline Flybe have also filed for bankruptcy in the past two years.

The global recession in 2008 proved that companies were inappropriately evaluated by credit rating agencies and banks. Governments around the world were forced to implement trillion-dollar rescue packages to keep the plumbing behind banking systems running. Given the devastating effects of the financial crisis on firms, now, more than ever, there is a need to identify (and anticipate) upcoming bankruptcies. The surge in the number of research papers around bankruptcy is increasingly evident from Figure 2.2.

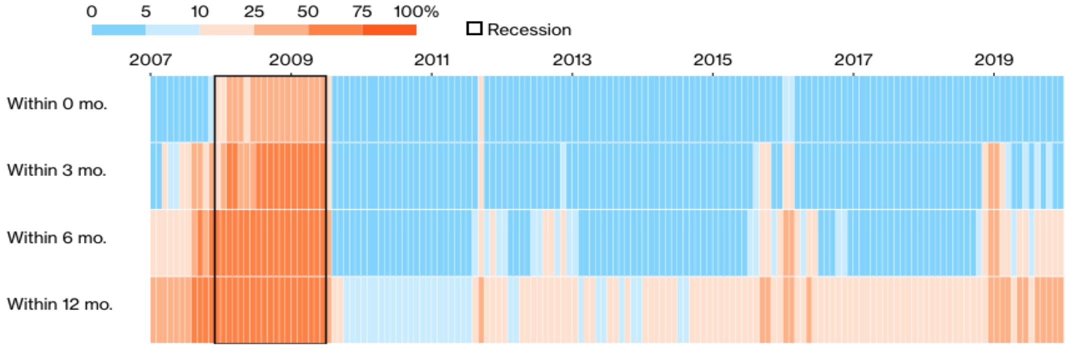


Figure 2.1: Probability of recession within 0, 3, 6 and 12 months
Source:Bloomberg

Situations may arise where a company can become distressed and continue to operate in that condition for many years and can even (at times) recover from their distress. On

[1] This study has not been updated to account for the market crash due to the COVID-19 pandemic.

the other hand, some companies enter bankruptcy immediately after a highly distressing event such as a financial fraud or accounting issues. Several factors influence these outcomes. Lensberg's paper [63] investigates related work and categorises numerous factors affecting bankruptcy. Broadly speaking they are audit, financial ratios and fraud indicators, which are measured by qualitative or quantitative variables. Bankruptcy occurs if a company cannot operate in circumstances, including force majeure events or due to government regulations, or due to its inability to pay off its debt and earn profits for an extended time.

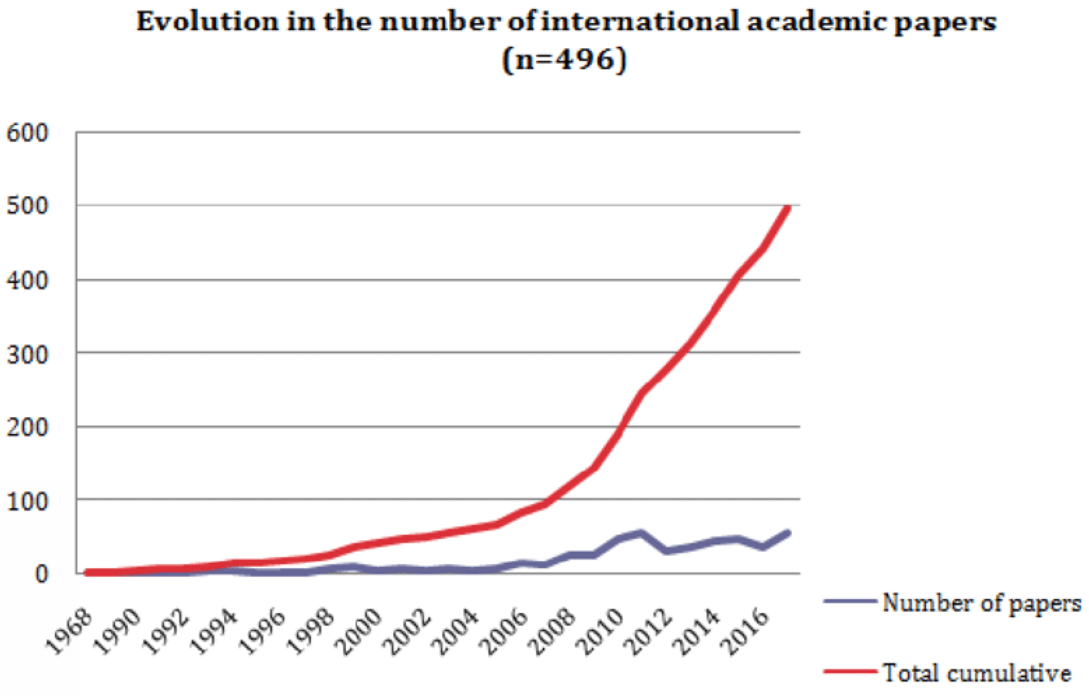


Figure 2.2: Rise in the number of international academic articles during the years 1968-2017. This graph has been taken from Shi and Xiaoni's work, published in 2019 [91].

Initial models used in bankruptcy prediction were mainly statistical, such as Multiple Discriminant Analysis [5] or Logistic Analysis [77]. Although these models have been widely used in academia and industry, statistical models are constrained in their ability to increase predictive power. [13]. Many recent studies suggest the use of intelligent data mining techniques, including neural networks (NNs), decision trees(DT), case-based reasoning (CBR), support vector machines (SVM), and soft computing [60]. Neural Networks have an excellent ability to treat non-linear data making them one of the most actively used models in bankruptcy prediction [23, 8, 24].

Recent efforts have shown that the performance of predictive models can be significantly enhanced through hybrid and ensemble computing [106]. A hybrid system exploits several approaches (e.g., heuristic techniques and classification algorithms) aiming for optimizing the prediction performance. Following this direction, evolutionary algorithms such as genetic algorithm (GA), annealing simulation (AS), particle swarm optimization (PSO), ant colony optimization (ACO), tabu search (TS) are extensively employed in conjunction with machine learning methods [68]. The typical usages include tuning the architecture of a particular model (such as the connected weights of MLP [50] and the parameters of SVM [73]), selecting relevant features, and refining the samples for learning. From another viewpoint, the ensemble approach combines several models and aggregates the output in some rules. It has been shown that a well designed ensemble-based system can outperform a single predictor by inheriting advantages of its base learners.

2.1 Early Statistical Techniques Used in Bankruptcy Prediction

The first work using data to predict bankruptcies of companies is from Beaver (1966)[11] and Altman (1968)[5]. It appears to be the genesis and benchmark for several empirical studies published henceforth. Beaver [11] developed a one-dimensional dichotomous classification, i.e., based upon a single ratio. Subsequently, Deakin (1972)[32] and Edmister (1972)[42], have shown that the predictive power of financial ratios is additive and that individual ratios have less predictive power than a small number of independent ratios used simultaneously. The multivariate analysis allows a richer description of the situation of the company and was used systematically.

Shirata (1998) [93] and Taffler (1983) [101] tried to determine the financial ratios that predict bankruptcy of the company by using only discriminant analysis (DA). However, the new model for predicting the failure inspired by DA such as multi-criteria discriminant analysis was applied by Zopounidis and Doumpos (2002)[119]. Subsequently, due to the restrictive statistical requirement of normality for the explanatory variables and quality of the variance-covariance group matrices, logit and probit models were also applied. Among these, the pioneering studies in logistic regression were carried out by Ohlson (1980) [77]. It is the first in this area to look at the prediction of corporate failure.

West (1985)[113] used the factor analysis to create composite variables to describe a bank's financial and operating characteristics. Experimental results demonstrated that the combined method of factor analysis and logit was promising for evaluating the bank's condition. However, these conventional statistical techniques have some restrictive assumptions, such as the linearity, normality, independence among predictor variables and pre-existing functional form relating the criterion variable and the predictor variable.

In more current literature, when predicting corporate bankruptcy, researchers have routinely used accounting-based variables (e.g., profitability ratio and liability ratios) and market-based variables (e.g., stock market returns and volatility) as a gauge of default risk. We refer readers to Kumar and Ravi's paper [60] that provides a comprehensive literature review on the studies before 2008. In Table 2.1, we curate a list of more recent studies on bankruptcy prediction.

Study	Source of data	Sample size	Models	Time period	Variables type
Sueyoshi and Goto (2009)[99]	Japanese Construction Industry	1K	DEA-DA, PCA	2000–2005	Accounting
Ioannidis, Pasiouras, and Zopounidis (2010)[52] [52]	78 countries, Bankscope, World Bank	1K	UTADIS, MLP, CART, KNN, Ordered logit, stacked models	2007–2008	Accounting, country-level variables
Chen et al. (2011)[26]	France, Diane database	1K	GA + LVQ	2006–2007	Accounting
Olson, Delen, and Meng (2012)[78]	USA, Compustat	1K	DT, logit, MLP, RBFN, SVM	2005–2009	Accounting
Ding et al. (2012)[35]	USA, Compustat, CRSP	1M+	Transformation Survival Model	1981–2006	Accounting & Market
Sánchez-Lasheras et al. (2012)[85]	Spain, Bureau van Dijk	63K	SOM + MARS	2007–2008	Accounting (5 Altman variables)
Cinca and Nieto (2013)[88]	USA, FDIC	8K	PLS-DA	2008–2011	Accounting
Geng et al. (2015)[46]	China, CSMAR	200	NN, DT, SVM, MV	2001–2008	Accounting
Wanke et al. (2015)[111]	Brazilian, Economatica	600	DEA+DSBM	1996–2011	Accounting
du Jardin (2015)[38]	France, Bureau van Dijk – Diane database	18K	DA, logit, MLP, SA	2003–2012	Accounting
Tian et al. (2015)[104]	USA, Compustat, CRSP	1.5M	Discrete Hazard Model, Logit	1980–2009	Accounting & Market
du Jardin (2016)[39]	France, Bureau van Dijk – Diane database	17K	Bagging, boosting, random subspace, PBM	2003–2012	Accounting
Liang et al. (2016)[67]	Taiwan Economic Journal (TEJ)	500	SVM, KNN, NB, CART, MLP	1999–2009	Accounting, market, corporate governance
Doumpos et al. (2017)[37]	18 EU countries, Bureau van Dijk, Eurostat, IEA, OECD, and UNECE	13K	MCDA	2012–2016	Accounting, Macroeconomic, energy markets
Calabrese et al. (2017)[20]	U.S. Department of the Treasury, FDIC, Call Reports	10K	LOBGEV(GEV model and D-vine copula)	2008–2013	Combination of variables

Table 2.1: Recent studies on bankruptcy prediction.

2.2 Recent Intelligent Techniques in Bankruptcy Prediction

In recent years, many studies have demonstrated that intelligent techniques can be alternate methodologies to predict corporate bankruptcy. Intelligent techniques automatically extract knowledge from a dataset and construct different model representations to explain the data set. The major difference between intelligent techniques and statistical techniques is that statistical techniques usually need researchers to impose structures to different models, such as linearity in the multiple regression analysis. Statistical techniques also require researches to construct the model by estimating parameters to fit the data or observation, while intelligent techniques allow learning the particular structure of the model from the data [108].

Most of the studies in Table 2.1 fall into the intelligent technique category. The goal of these studies was to develop accurate models using artificial intelligence and operations research techniques. Also, models that allow non-linear decision boundaries (e.g., neural networks, SVM with non-linear kernels) quickly gained popularity and are now widely applied. These features provide better model flexibility and improved classification performance. A trend in recent literature is studying the combinations of models. Several studies demonstrate how to combine various models horizontally using ensemble techniques as shown by Geng et al (2015) [46] and Kim & Kang, (2010)[56], or vertically as shown by du Jardin (2016) [39]. These hybrid models can capture greater variations in the decision space and result in more stable and accurate predictions.

Second, we notice a wide diversification of data sources in recent studies. As noted before, theoretical and empirical studies have long established that accounting-based ratios and market-based variables are the main indicators of future bankruptcy. More recent studies have started to evaluate the predictive power of data sources beyond the two types of variables. For example, Liang, Lu, Tsai, and Shih (2016) [67] examines the discriminatory power of a broad array of corporate governance indicators (discussed in detail in section 2.3), including board structure, ownership structure, leadership personnel, and others. Doumpos, Andriosopoulos, Galariotis, Makridou, and Zopounidis (2017)'s model [37] takes country characteristics into account. They show that country-level data on the economic and business environment, energy efficiency policies, as well as characteristics of markets, can add value to corporate failure prediction models. Calabrese, Degl'Innocenti, and Osmetti (2017) [20] study how the U.S. government's Troubled Asset Relief Program (TARP) impacted the probability of failure among commercial banks.

Table 2.2 provides an overview of some of the other machine learning algorithms used in past studies. The models used in our research are discussed in greater detail in chapter 4.

S/N	Variable	Description	Advantages	Disadvantages
1.	Neural Networks (NN)	Learn from examples using several constructs and algorithms just like a human being learns new things	Good at function approximation, forecasting, classification, clustering and optimization tasks depending on the neural network architecture	The determination of various parameters associated with training algorithms is not straightforward. Many neural network architectures need a lot of training data and training cycles (iterations)
2.	Genetic Algorithms (GA)	Mimics Darwinian principles of evolution to solve highly nonlinear, non-convex global optimization problems	Good at finding the global optimum of a highly nonlinear, non-convex function without getting trapped in local minima	Does take a long time to converge; May did not yield global optimal solution always unless it is augmented by a suitable direct search method
3.	Case-based reasoning (CBR)	Learns from examples using the euclidean distance and k-nearest neighbour method	Good for small data sets and when the data appears as cases; similar to the human-like decision-making	Cannot be applied to large data sets; poor in generalization
4.	SVM	It uses statistical learning theory to perform classification and regression tasks	It yields global optimal solution as the problem gets converted to a quadratic programming problem; It can work well with few samples	Selection of the kernel and its parameters is a tricky issue. It is abysmally slow in the test phase. It has high algorithmic complexity and requires extensive memory
5.	Rough sets	They use a lower and upper approximation of a concept to model uncertainty in the data	They yield ‘if-then’ rules involving ordinal values to perform classification tasks	It can be (a) sometimes impractical to apply as it may lead to an empty set (b) sensitive to changes in data and (c) inaccurate

Table 2.2: Merits and Demerits of Intelligent Techniques.

2.3 Corporate Governance Indicators

The general definition of corporate governance includes the mechanisms, processes and relations by which corporations are controlled and directed [90].

Many corporate governance indicators (CGIs) have been identified in the literature which has been used for solving bankruptcy or financial crisis problems. These can be broadly classified into five categories including board structure, ownership structure, cash flow rights, key persons retained, and others.

However, the prediction performance obtained by combining CGIs and financial ratios has not been fully examined. Only some selected CGIs and financial ratios have been used in related studies and the chosen features may differ from study to study. Lee (2004) [62] uses six financial ratios belonging to solvency, profitability, and other categories use. It also uses ten CGIs in the board structure and ownership categories. They found that model performance was enhanced by using a combination of CGIs and financial ratios. However, these findings may not be applicable in some markets where the definition of distressed companies is unclear and the characteristics of corporate governance indicators are not obvious, such as in the Chinese market.

Note: Although including CGI data in our research would be very promising, the paucity of such data hinders our ability to incorporate CGIs into our bankruptcy prediction models.

2.4 Ensemble Classifiers in Bankruptcy Prediction

Ensemble methods have been known to be used as tools to improve the accuracy of learning algorithms by combining a set of weak classifiers (otherwise called weak learners), each of which needs only moderate performance on the training set [82, 86]. Two popular methods for creating accurate ensembles are Bagging [17] and Boosting [44]. Both theoretical and empirical studies have demonstrated remarkable improvements in the generalisation behaviour [10, 45]. Literature suggests that ensemble methods decrease the generalisation error of CART decision trees [17], C4.5 decision trees [84], and Neural Networks (NNs) [79].

Several studies on the bankruptcy prediction have applied AdaBoost, a popularly used boosting algorithm, to bankruptcy classification. Results have shown that AdaBoost decreases the generalisation error and improves the accuracy [31] of these ensemble classifiers. An empirical comparison has shown that AdaBoost with classification trees decreases the generalisation error by around 30% with respect to an error produced with a NNs [4]. Previous studies have suggested that ensembles with classification trees are very effective for bankruptcy prediction, however, there has been little empirical testing of an ensemble with NNs in bankruptcy prediction literature. The primary reason is that ensembles with decision trees provide fast training speed and well-established default parameter settings, while NNs has the difficulties for testing both in terms of the significant processing time required and in selecting training parameters [79]. The significant overhead in training NNs and its hyper-parameter optimisation is the main reason that we exclude NNs from our comparative study.

We choose to use ensemble methods as they are expected to provide the following advantages over the traditional NNs (as detailed in [56]); First, ensembles have been shown to produce results with greater prediction performance compared to any of the individual NNs classifiers. Second, the classification approaches that use error minimisation are prone to overfitting when a classifier is highly adjusted to learn the training set. This causes the classifier's generalisation error to increase significantly when it is applied to previously unseen data. Ensemble models like Random Forests, Bagging and XGBoost are robust to overfitting and thus reduce generalisation error.

Against these backgrounds, we propose three ensemble methods to improve the performance of bankruptcy prediction. The three popular methods, Random Forests, Bagging and Extreme Gradient Boosting (XGBoost), are used for creating high-performance ensembles by combining the predictions of multiple tree-based (weak) classifiers. This paper presents a comprehensive evaluation of applying these ensemble models on predicting bankruptcy of listed Japanese firms.

Our review of the bankruptcy prediction literature has highlighted three key gaps. First, there is a lack of seminal work done on Japanese financial data. This gives an opportunity to put forth our analysis on this seldomly studied economy. Second and to the best of our knowledge, a comprehensive comparison regarding data imputation models on financial data and their effect on the performance of bankruptcy prediction models have not been undertaken on financial data from Asia. Third, a holistic performance evaluation of tree-based ensemble models, specifically Random Forests, Bagging and XGBoost on the Japanese Data has not been conducted before, but similar studies on leading economies such as USA, France and Germany have yielded positive performance results. We aim to help address these important gaps in the literature.

Chapter 3

Our Japanese Dataset

3.1 Why we chose the Japanese Economy

Past research shows a major focus on bankruptcy prediction models for the American and European market. In an attempt to contribute to research focused on the Asian market, we focus our attention to Japan - one of the largest economies in the world. As of 2018, the market capitalisation[1] of Japan is the third-largest in the world at **$5.296 trillion** (Source: World Federation of Exchanges database), with the US and China having the largest global markets values respectively. Countries like the United Kingdom, Germany and France are similar to most capital markets but are significantly different from the countries in Asia.

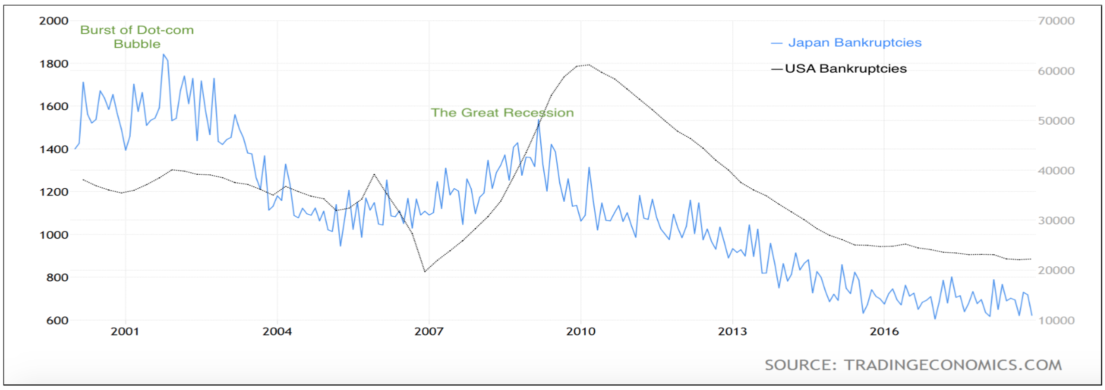


Figure 3.1: Comparing the number of bankruptcies in Japan and USA between 2000-2018. Source:Trading Economies

A comparison between Japan and the US in the number of corporate bankruptcies between 2000-2018 can be seen in Figure 3.1. The figure summarises the number of bankruptcy filings each year and it is clear that the bankruptcy filing events display a strong cyclical trend. In specific, we can note the two big jumps around the most

[1] Market capitalisation (also known as market value) is the share price times the number of shares outstanding (including their several classes) for listed domestic companies.

recent global recession periods, namely, the early 2000s and late 2000s. Such business-related fluctuation confirms the existence of the "domino-effect" financial distress at an international level.

In Japan, the two most prominent legal routes for filing for bankruptcy are: filing for court protection from creditors under the Corporation Reorganisation Law or under the Civil Rehabilitation Law.

The Corporation Reorganisation Law is aimed at firms whose failure would have a big impact on society, the management must resign and a new team is brought in according to a reorganisation plan devised by a court-appointed administrator. Although this procedure allows the company to sever ties with executives deemed responsible for the collapse, recovery often takes longer under this process than via the Civil Rehabilitation Law. The Civil Rehabilitation Law, similar to America's Chapter 11, is often preferred because it allows the company to rebuild under its management team, speeding up the reconstruction process.

3.2 Data-source

We obtained the financial data of 4000 Japanese firms from an international financial database, provided by Wharton Research Data Services (Website: WRDS). WRDS ensures convenient comparability among different countries without loss of the data consistency and reliability, regardless of differences in geometric location, business regulations etc. The database has information on listed, and major unlisted/delisted, companies across the globe. Japanese corporations also have well documented financial data as they adopt GAAP[2] for consolidated financial statements.

To formally construct the international bankruptcy database, we collected each firm's annual financial information and its bankruptcy status from WRDS' annual files, this includes data from the Tokyo Stock Exchange and Osaka Securities Exchange.

To maintain comparability with other studies, we have chosen to adopt the following definition of bankruptcy - a firm is classified as "bankrupt" if it files either Chapter 7 (liquidation) or Chapter 11 (using the Civil Rehabilitation Law). In particular, the bankruptcy indicator for the firm 'i' at a time 't' is set to 1 if the firm was delisted due to filing for Chapter 7 or Chapter 11, at the time 't'. There are very few cases that firms who were delisted may re-enter the database later, but we do not consider any firm's observation after their first delisting in our analysis.

3.2.1 Choosing 2000 to 2018 as a Sampling Period

In this study, we have chosen to limit our sampling period to begin from 2000 to 2018 as the Japanese economy went underwent a series of changes in bankruptcy regulations in the years leading up to 2000. In 1998, the Japanese financial regulatory standard was tightened by the government [76]. Doing so greatly weakened the original corporate

[2] Generally accepted accounting principles (GAAP) refer to a common set of accounting principles, standards, and procedures issued by the Financial Accounting Standards Board (FASB).

structure of Japanese market - a system that mainly relied on the connection to the main bank and/or the major Keiretsu[3] groups for financing; The research in [116] showcases this in further detail. Before 1998, a firm that was unable to pay off the debt from its creditors could seek financial support from the ”main bank” or a certain Keiretsu group, if the company was associated to any, to avoid stepping in any further default filing process. However, after the government tightened the regulatory standard, neither the main bank nor the Keiretsu group was allowed to save a firm from default. This institutional structure change leads to an increased number of bankruptcy filings. Therefore, to avoid skewed bankruptcy figures, we begin our sampling only from 2000.

3.3 Dataset Reorganisation

Re-organising the dataset was the most time-consuming task and the hardest challenge that we faced. The initial WRDS data set consisted of financial data from 1990 to 2018 of 4000 Japanese firms. Accounting for the reasons mentioned in the above sub-section, we first removed all the data that predated 2000. Our next big challenge was to combine all the available information on one timescale, as not all companies followed GAAP standards of reporting data. Careful and precise efforts needed to be undertaken to ensure that all financial indicators were reported in the same month/year so that further examinations could be made.

Acknowledging that financial data suffers from class imbalance[4] and concept drift[5] that results in poor and degrading predictive performance in predictive models. This paper proposes the use of rolling time window (as recommended in [74, 51, 98]) with a fixed window size.

Building on the successes seen in [71, 66, 74], we have decided to incorporate a ten-year rolling window, to predict bankruptcy in a one-year horizon. This required us to reorganise the data to represent five forecasting years, i.e, Year 1, Year 2, Year 3, Year 4 and Year 5. This approach has been explored to outperform conventional bankruptcy models that otherwise use data from a fixed period to predict bankruptcy within a fixed horizon (number of years). Table 3.1 showcases the steps we have taken to reorganise the dataset, to carry out a prediction task.

3.4 Dataset Description

3.4.1 Feature Extraction: Using Financial Ratios

Studies in the past have aimed at proposing novel machine learning techniques to enhance the models’ prediction performances. However, research focused on the effect

[3] A keiretsu is a set of companies with interlocking business relationships and shareholdings. In the legal sense, it is a type of informal business group that are loosely organised alliances within the social world of Japan’s business community.

[4] Our dataset consists of bankrupt and non-bankrupt/operational companies. The number of companies that go bankrupt is far fewer than the number of companies that remain operational, hence a company that goes bankrupt is part of the minority class

[5] Concept drift has been explained in detail in Appendix A.

Dataset Name	Prediction Year	Window Length	Number of Firms in Dataset
Year 1	2013	2003 - 2012	2342
Year 2	2014	2004 - 2013	3391
Year 3	2015	2005 - 2014	3501
Year 4	2016	2006 - 2015	3264
Year 5	2017	2007 - 2016	1970

Table 3.1: Using rolling windows to predict the bankruptcy in the next financial year.

of the input variables (or features) on prediction performance is scarce. In general, financial ratios (FRs) have been recognised as one of the most important factors affecting bankruptcy prediction and are used to develop prediction models [5, 11, 77].

Our feature selection is based on the seminal papers [5, 12, 48, 103, 104, 35]. A comprehensive use of financial ratios in bankruptcy prediction can be seen in the study [67]. Financial ratios generally cover seven categories, which reflect the company's solvency, profitability, cash flow ratios, capital structure ratios, turnover ratios, company growth and others.

Our feature selection for this project was done by taking a tailored combination of the top eighty, most frequently occurring financial ratios that have been used in the above-mentioned papers. For this purpose, our Japanese dataset was re-structured to create these eighty financial ratios. In doing so, we realised that our dataset had a few limitations as it did not contain data related to the ownership structures[6], turnover of 'C-level' executives and compensation given to board members.

Discarding these financial ratios, we were able to create sixty-four financial ratios that could be used as predictive variables. These synthetic features were generated by a selection of two or more existing features and applying an arithmetical operation on them. The arithmetic operation included the following set of possible values: $\{+, -, \times, \div\}$.

The central idea we have used in our approach is to leverage financial ratios as they may have a better influence on prediction than typical economic factors [67]. Combining financial ratios with the use of tree-based ensemble models has resulted in more effective learning from the data as shown in [117]. We have taken advantage of this property and further propose a model that uses an ensemble of boosted trees, dedicated to solving the problem of bankruptcy prediction.

The research done by Beaver, McNichols, and Rhie in [12], on global markets, concluded that over a long period of time, the performance of the bankruptcy prediction model with both financial statement data and market data is essentially similar to the one with financial ratios only. Therefore we have considered only FRs in our model.

A description of the sixty-four features is presented in Table 3.2. All variables used for calculation of the financial indicators are obtained from the balance sheets, income

[6] Ownership structure data mainly involves data related to shareholders i.e., the shareholding ratio of the board, shareholding ratio of directors, shareholding ratio of an outside person, etc.

statements or cash flow statements of the companies.

3.4.2 Data Range and Correlations

Appendix B contains the data ranges including the minimum, maximum and mean values of the financial ratios. We present the data ranges for each window (Year 1, Year 2, Year 3, Year 4 and Year5). We have also included correlation heatmaps for the FRs in Appendix C, examining these heatmaps, we conclude that we have a good degree of variability in our features as the heatmaps show a low correlation between features.

3.4.3 Handling Missing Data

There are three typical mechanisms causing missing data [69, 97, 41]:

1. Missing completely at random (MCAR)
2. Missing at random (MAR)
3. Missing not at random (MNAR)

If the mechanism causing missing data is not dependent on observed data nor on the missing data, then data is said to be missing completely at random (MCAR) [97, 41]. MCAR causes enlarged standard errors due to the reduced sample size but do not cause bias ('systematic error' that is an overestimation of benefits and underestimation of harms). The MAR and MNAR conditions cannot be distinguished based on the observed data since, by definition missing data are unknown and it can therefore not be assessed if the observed data can predict the unknown data [97, 41].

As justified in [49], missing values in Japanese financial data could be due to the following reasons:

1. The Japanese accounting standards for the net assets section changed in 2006.
2. The notation for accounting items can differ depending on the industry, even if they have similar meanings.
3. Items that have zero value are missing. This might cause erroneous calculations when computing financial ratios.

Conducting further exploratory data analysis (after having created our financial ratios), we found that the dataset still contained several missing values. Examining this more closely, the dataset was tested to observe the extent of data loss if listwise deletion (an entire record is excluded from analysis if any single value is missing) is used. We observe that the data loss would be severe as we lose over 50% of our available data. This is detailed in Table 3.3

Furthermore, the python library **missingno** was used to visualise the features that had missing values. Nullity matrices were created for each forecasting period to find visual patterns of missing values; white spaces in the graph can be inferred as missing values. An example of the Year 2 dataset can be seen in Figure 3.2

ID	Description	ID	Description
X1	Net Profit / Total Assets	X33	Operating Expenses / Short-Term Liabilities
X2	Total Liabilities / Total Assets	X34	Operating Expenses / Total Liabilities
X3	Working Capital / Total Assets	X35	Profit on Sales / Total Assets
X4	Current Assets / Short-Term Liabilities	X36	Total Sales / Total Assets
X5	[(Cash + Short-Term Securities + Receivables - Short-Term Liabilities) / (Operating Expenses - Depreciation)] * 365	X37	(Current Assets - Inventories) / Long-Term Liabilities
X6	Retained Earnings / Total Assets	X38	Constant Capital / Total Assets
X7	EBIT / Total Assets	X39	Profit on Sales / Sales
X8	Book Value of Equity / Total Liabilities	X40	(Current Assets - Inventory - Receivables) / Short-Term Liabilities
X9	Sales / Total Assets	X41	Total Liabilities / ((Profit on Operating Activities + Depreciation) * (12/365))
X10	Equity / Total Assets	X42	Profit on Operating Activities / Sales
X11	(Gross Profit + Extraordinary Items + Financial Expenses) / Total Assets	X43	rotation Receivables + Inventory Turnover in Days
X12	Gross Profit / Short-Term Liabilities	X44	(Receivables * 365) / Sales
X13	(Gross Profit + Depreciation) / Sales	X45	Net Profit / Inventory
X14	(Gross Profit + Interest) / Total Assets	X46	(Current Assets - Inventory) / Short-Term Liabilities
X15	(Total Liabilities * 365) / (Gross Profit + Depreciation)	X47	(Inventory * 365) / Cost of Products Sold
X16	(Gross Profit + Depreciation) / Total Liabilities	X48	EBITDA (Profit on Operating Activities - Depreciation) / Total Assets
X17	Total Assets / Total Liabilities	X49	EBITDA (Profit on Operating Activities - Depreciation) / Sales
X18	Gross Profit / Total Assets	X50	Current Assets / Total Liabilities
X19	Gross Profit / Sales	X51	Short-Term Liabilities / Total Assets
X20	(Inventory * 365) / Sales	X52	(Short-Term Liabilities * 365) / Cost of Products Sold)
X21	Sales (n) / Sales (n-1)	X53	Equity / Fixed Assets
X22	Profit on Operating Activities / Total Assets	X54	Constant Capital / Fixed Assets
X23	Net Profit / Sales	X55	Working Capital
X24	Gross Profit (in 3 years) / Total Assets	X56	(Sales - Cost of Products Sold) / Sales
X25	(Equity - Share Capital) / Total Assets	X57	(Current Assets - Inventory - Short-Term Liabilities) / (Sales - Gross Profit - Depreciation)
X26	(Net Profit + Depreciation) / Total Liabilities	X58	Total Costs /Total Sales
X27	Profit on Operating Activities / Financial Expenses	X59	Long-Term Liabilities / Equity
X28	Working Capital / Fixed Assets	X60	Sales / Inventory
X29	Logarithm of Total Assets	X61	Sales / Receivables
X30	(Total Liabilities - Cash) / Sales	X62	(Short-Term Liabilities *365) / Sales
X31	(Gross Profit + Interest) / Sales	X63	Sales / Short-Term Liabilities
X32	(Current Liabilities * 365) / Cost of Products Sold	X64	Sales / Fixed Assets

Table 3.2: Summary of features used

Dataset	Total number of instances	Number of instances with missing data	Number of instances with no missing data	Data loss if listwise deletion is used
Year 1	2342	1278	1064	54.54%
Year 2	3391	2029	1362	59.81%
Year 3	3501	1879	1628	53.48%
Year 4	3264	1675	1589	51.29%
Year 5	1970	960	1010	48.71%

Table 3.3: Data Quality Assessment: Missing Data

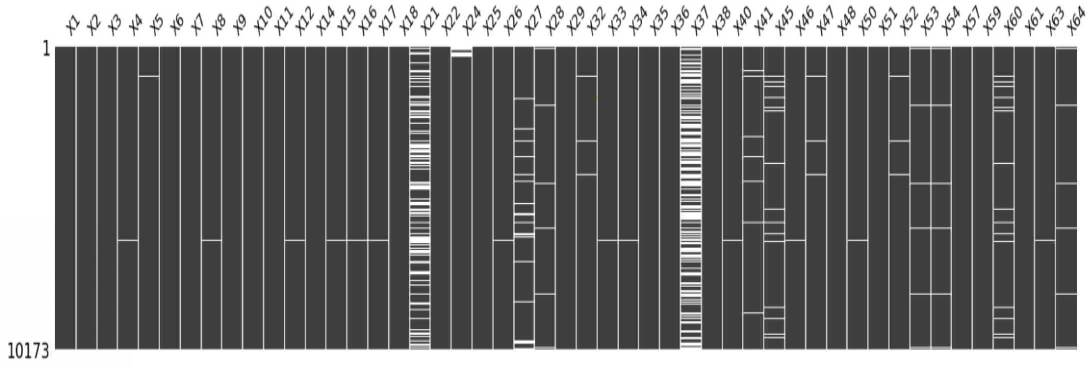


Figure 3.2: Nullity Matrix for Year 2. White space along a feature column represents missing data.

In an attempt to make the data richer, we have removed all those instances that have over 50% of their attributes as either ‘n.a.’ or blank/empty - this resulted in our dataset shrinking by 10.3 %.

Missing data can potentially cause three major problems [55]:

1. The missing data can cause bias in the estimation of parameters.
2. It can reduce the representativeness of the samples and may complicate the analysis of the study.
3. The absence of data reduces statistical power, which refers to the probability that the test will reject the null hypothesis when it is false.

Dropping all the rows with missing values or listwise deletion introduces bias and affects representativeness of the results. A viable alternative to listwise deletion is by estimating missing data by using imputation techniques. In our project, we explored three techniques of imputation, and we will see them in the subsequent sections.

1. Expectation-Maximisation Imputation
2. k-Nearest Neighbours Imputation
3. Multivariate Imputation Using Chained Equations

3.4.4 Expectation-Maximisation Imputation

In statistics, Expectation–Maximisation (EM) is an iterative method to find maximum likelihood estimates of parameters in statistical models. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters. This is followed by a maximisation (M) step, which computes parameters maximising the expected log-likelihood found on the E step [75]. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step. EM Imputation is, therefore, the process of imputing missing values using Expectation-Maximisation. Missing values of quantitative variables are replaced by their expected value computed using the Expectation-Maximisation (EM) algorithm. In practice, a Multivariate Gaussian distribution is assumed. In general, EM imputation is better than mean imputations because they preserve the relationship with other variables [6].

We carried out EM Imputation using python's *impyute* library and used 50 as the number of EM iterations to run before breaking.

3.4.5 k-Nearest Neighbours Imputation

The k-nearest neighbour's algorithm or kNN is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. It can also be used as a data imputation technique where kNN imputation replaces 'n.a.' and empty values in the data with the corresponding value from the nearest-neighbour row. The nearest-neighbour row is the closest row by Euclidean distance. If the corresponding value from the nearest-neighbour is also 'n.a.', the next nearest neighbour is used. After finding k nearest neighbours, the weighted average of them is returned.

We carried out kNN Imputation using python's *fancyimpute* library and used 100 nearest neighbours for the process.

3.4.6 Multivariate Imputation using Chained Equations

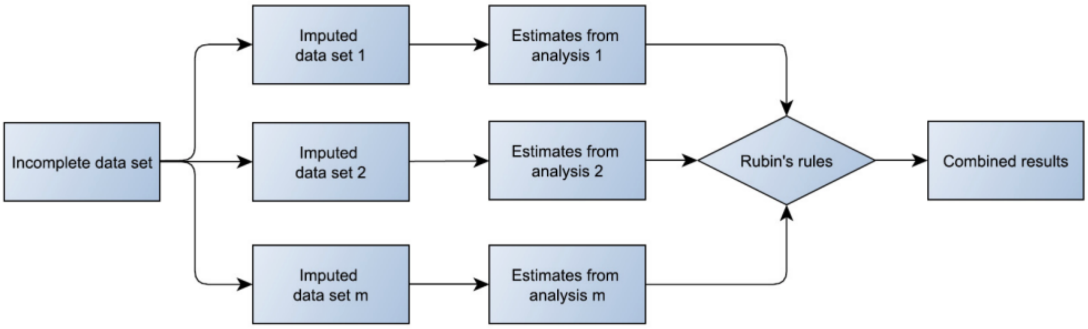


Figure 3.3: The MICE-process [115]

Multiple imputations using chained equations or MICE is an imputation technique that uses multiple imputations to tackle missing data. MICE has become one of the principal methods of addressing missing data [7]. Creating multiple imputations, as opposed to single imputations, accounts for the statistical uncertainty in the imputations.

Since each variable is imputed using its own imputation model, MICE can handle different variable types (for example, continuous, binary, categorical), as well as complexities such as varying bounds [115]. Multiple imputations (MI) was originally designed to handle missing data in public-use large datasets. Hence this is an ideal imputation technique for our data. Because multiple imputations involve creating multiple predictions for each missing value, the analysis of multiple imputed data take into account the uncertainty in the imputations and yield accurate standard errors [115].

We outline the MICE algorithm for a set of variables, $x_1 \ldots x_k$, some or all of which have missing values. Initially, all missing values are filled in at random. The first variable with at least one missing value, x_1 say, is then regressed on the other variables, $x_2 \ldots x_k$. The estimation is restricted to individuals with observed x_1. Missing values in x_1 are replaced by simulated draws from the posterior predictive distribution of x_1, an important step known as proper imputation. The next variable with missing values, say x_2, is regressed on all the other variables, $x_1, x_3, ..., x_k$. Estimation is restricted to individuals with observed x_2 and uses the imputed values of x_1. Again, missing values in x_2 are replaced by draws from the posterior predictive distribution of x_2. The process is repeated for all other variables with missing values in turn: one such round is called a cycle. To stabilize the results, the procedure (similar to a Gibbs sampler) is usually repeated for about ten cycles to produce a single imputed dataset. The entire procedure is repeated independently 'm' times, yielding 'm' imputed data-sets. Standard texts on MI suggest that small numbers of imputed data-sets (m = 3 to 5) are adequate.

According to [94], which does an extensive comparison between different imputation methods, multiple imputations is the most accurate method for dealing with missing data in most of the missing data scenarios. The paper recommends using no more than M=5 imputations and sometimes as small number as 2 or 3 to generate useful statistical inferences.

We carried out MICE imputation using python's *fancyimpute* library and used 5 imputations in our study.

3.4.7 Dealing with Data Imbalance

As bankruptcy is an uncommon event, it would be reasonable to note that there are only a few companies that file bankruptcy every year. We showcase this class imbalance on our own data in Table 3.4.

Having an imbalanced dataset can cause our machine learning models to overfit to the majority class, i.e., companies that are still operational, i.e. non-bankrupt companies. Since the data is skewed towards one class, machine learning models can struggle to learn to correctly classify a minority class (bankrupt) instance. Despite this class imbalance, most models would still produce a reasonably high accuracy as most instances fall under the non-bankrupt majority class. Therefore, we disregard accuracy as a valid metric and use AUC, F1-Score, precision and recall, to evaluate the performance of our model.

To tackle this data imbalance, we use the over-sampling technique that has been proposed in [61], to generate synthetic data that adds instances from the minority class to the dataset. This technique will raise the percentage of minority class data from around 5% to 15% of the dataset, as recommended by extensive work done in [21]. This will allow the models to be exposed to a greater number of instances with the minority class, thus improving the chances for the model to predict the correct classification when an instance is labelled as bankrupt.

Synthetic Minority Over-sampling Technique (SMOTE) is an algorithm has been used to over-sample the dataset as it is one of the most well-known techniques for oversampling. SMOTE [25] works by selecting/sampling similar instances of the minority class, by finding the nearest neighbours k (using Euclidean distance) and changing each attribute one at a time by multiplying each x by a random number (between 0 to 1), therefore creating a new synthetic instance. We generate these minority class samples for our dataset after estimating the missing values using the previously mentioned imputation techniques. SMOTE has been implemented using the *imblearn.over_sampling* library.

Dataset	**Total number of instances**	**Number of bankrupt instances**	**Number of non-bankrupt instances**	**Percentage of bankrupt instances**
Year 1	2342	90	2252	3.93%
Year 2	3391	134	3257	4.71%
Year 3	3501	165	3336	5.25 %
Year 4	3264	172	3092	6.93%
Year 5	1970	137	1833	3.93%

Table 3.4: Data organisation and Instances

3.4.8 Dimensionality Reduction and Feature Selection

Dimensionality reduction is a crucial component of financial analysis and has received a lot of interest in recent studies [22]. Many dimensionality reduction methods have been proposed, such as t-testing, correlation matrices, factor analysis, principal component analysis (PCA), independent component analysis (ICA), etc. In a linear pre-processing stage, PCA and ICA are capable of improving the discriminating power of classifiers [27]. However, nonlinear projection methods are particularly applicable to solve high-dimensional financial data. Reducing the number of variables was found to be one of the key components in the successful prediction of bankruptcy, not only simplifying the model structure but also by improving the discriminative power [22]. The dimensionality reduction method we used is described in subsubsection 3.4.8.1

Feature selection techniques are methods used to eliminate features which are redundant or irrelevant. This is achieved using different methods; the most common is to find those features that are correlated to each other (or correlation with the outcome) thus removing redundant and unwanted features. Reducing the number of irrelevant or redundant features drastically reduces the running time of a learning algorithm and yields a more general concept. There are many potential benefits of feature selection, some of which are:

1. Facilitating data visualisation and data understanding
2. Reducing the measurement and storage requirements
3. Reducing training and utilisation times
4. Defying the curse of dimensionality to improve prediction performances by simplifying the hypothesis in a model

In addition, this helps in getting a better insight into the underlying concept of a real-world classification. There are various techniques to achieve this, but for the purpose of this project, Recursive Feature Elimination (RFE) and Chi-Square feature selection will be described in subsubsection 3.4.8.2 and subsubsection 3.4.8.3 respectively.

3.4.8.1 Principal Component Analysis (PCA)

Dimensionality Reduction in our project was carried out using principal component analysis (PCA). PCA re-orients a data set in the direction of the eigenvectors, which are ordered according to the extent of variance they capture from the data. The more eigenvectors we use, the higher the variance captured. PCA is a distributed representation wherein raw variables collaborate to generate a principle component. In predictive machine learning, principle components can replace the original variables. The objective is to learn the functional relationship between the target variable and the principle component. This can simplify the learning task, increase predictive accuracy, and facilitate feature reduction.

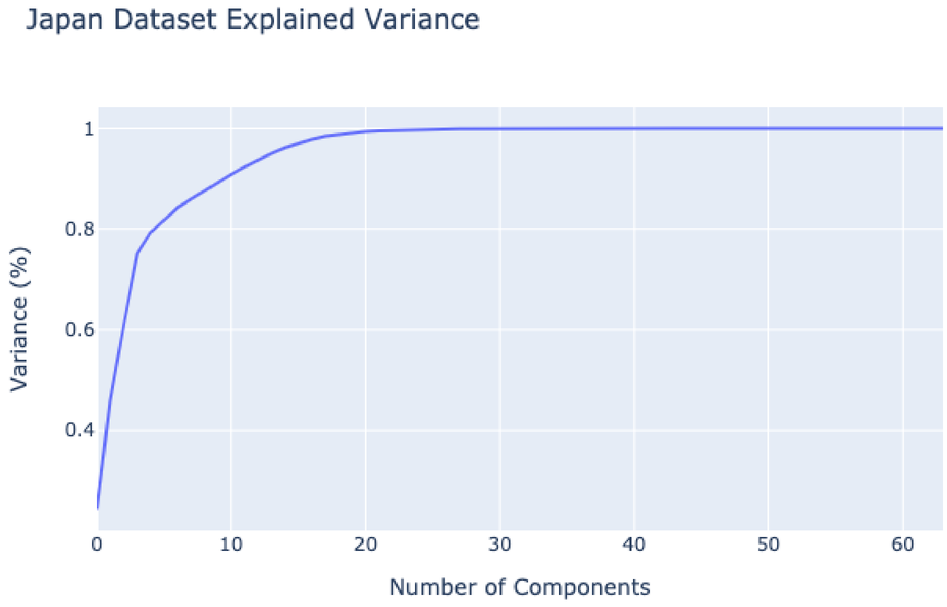


Figure 3.4: PCA Carried out on MICE Imputed and Oversampled Year-1 dataset.

Figure 3.4 shows the results of PCA being carried out on our dataset. After examining the graph, we decided to choose twenty, as the number of components as this seemed as the elbow point in the graph. Using just twenty components, we were able to attain a variance of 99.35%.

3.4.8.2 Recursive Feature Elimination

The Recursive Feature Elimination [47], as the name suggests, recursively fits the model and eliminates the least important feature/s with every iteration. This is done by ranking the features according to their coefficient weight and eliminating the least weighted features. After each iteration, the model has fitted again, and the least weighted feature/s are eliminated again until the specified number of maximum iterations or features to be eliminated is reached. In this project, the Logistic Regression classifier is used as an estimator.

The number feature selected from Recursive Feature Elimination was limited to 20, to maintain comparability with the dimensionality reduction methods.

The top 20 features selected by Recursive Feature Elimination are as follows:
’x11’, ’x33’, ’x52’, ’x53’, ’x61’, ’x63’, ’x13’, ’x17’, ’x19’, ’x36’, ’x54’, ’x64’, ’x6’, ’x9’, ’x23’, ’x31’, ’x32’, ’x50’, ’x8’, ’x20’.

3.4.8.3 Chi-Square Feature Selection

The Chi-Square [53] is used to determine if the feature/s and outcome are dependent or not. The features and classes must not contain non-negative values (re-scaling is used to change negative values to positive values). It is often used in machine learning to rank features based on their Chi-Square statistic (score). Features that are found to be irrelevant for classification, are discarded depending on a specified number of features to be selected so that only the top-ranked features are selected.

Having normalised and re-scaled the data, the number feature selected from Chi-Square was limited to 20, to maintain comparability with the dimensionality reduction methods.

The top 20 features selected by Chi-Square are as follows:
’x21’, ’x27’, ’x51’, ’x32’, ’x2’, ’x44’, ’x62’, ’x43’, ’x64’, ’x30’, ’x61’, ’x52’, ’x54’, ’x53’, ’x22’, ’x58’, ’x35’, ’x20’, ’x60’, ’x47’.

COVID19-Note: The experimentation process required us to use Edinburgh University GPU’s and work from DICE computers. Due to the closure of Appleton Tower on 17^{th} March 2020 and the inability to access University servers for computational needs, ongoing experiments using PCA, Recursive Feature Elimination and Chi-Square were halted abruptly. Moving away from Edinburgh lead to further disruptions with connectivity and the change of environment demanded the need to prioritise writing the report. All further experiments were stopped due to the lack of adequate computational capacity.

Chapter 4

Experiment Methodologies and Design

4.1 Data Partitioning - Cross Validation

In machine learning, model validation is referred to as the process where a trained model is evaluated with an unseen testing data set [109]. The main concept behind the validation process is to partition the dataset into a training set, which is used to train the model, and testing set, which is used to test and evaluate the model. In validation, the training set is utilised only one time, i.e., to train the model.

The problem with splitting the dataset into two partitions is that the model will only perform based on the data that was used to train it. This could lead to overfitting or underfitting hence weakening the model's ability to perform in a more generalised way (i.e., evaluate unforeseen data outside these two partitions). Validation also tends to induce some testing bias as we reserved a piece of the dataset just for testing. To mitigate such problems and fully utilise our dataset, we use cross-validation to re-enforce the predictive ability of our model. There are various techniques which try to ensure low bias and low variances such as K-Fold Cross Validation, Stratified K-Fold Cross-Validation and Leave-P-Out Cross-Validation.

Stratified K-Fold cross-validation has been implemented in our experiments using the python library *sklearn.model_selection.StratifiedKFold*. It is common to use this method for cross-validation as the folds are made by preserving the percentage of samples for each class. The paper [96] discusses the benefits of using Stratified K-Fold cross-validation where there is a high disparity in the number of instances of the majority and minority class. We ensured that each class is (approximately) equally represented across each test fold. These classes are combined in a complementary way to form training folds.

The intuition behind this relates to the bias of most classification algorithms. They tend to weight each instance equally which means over-represented classes get too much weight (e.g. optimising F1-measure, Accuracy or a complementary form of error). One specific issue that is important across even unbiased or balanced algorithms, is that they cannot learn or test a class that isn't represented at all in a fold [96].

To carry out Stratified K-Fold cross-validation, the dataset is split into *k* partitions, of which $k-1$ partitions are used as a training set and the one remaining partition is used as a testing set. Here we ensure that each partition has the same percentage of minority class samples as seen in that year's data. The process is iterated k number of times so that at the end of the iterations every partition will be used once as a testing set. At each iteration/fold performance metrics (like AUC, F1-score, precision, recall, etc) are measured. By the end of the k iterations, these metrics are averaged. This reduces both bias and variance as the original dataset is used for both training and testing sets. Therefore the model is neither overfitted by one large training set nor is it being under fitted by having a larger test set than the traditional validation method.

This technique is applied to the different machine learning models to help in the selection of the best performing model - by using the same folds for each model at each iteration. This technique enables the facility to tune hyperparameters by using this cross-validation technique with the same model but different hyperparameters and selecting the best at the end of the iteration (using some performance metrics). We have used 10-Fold cross-validation throughout this project, similar to the approach taken in [67].

4.2 Baseline with Logistic Regression

The foremost method of bankruptcy prediction, proposed by Altman [5] used logistic regression (LR). Logistic regression [1] is a binary classification method which assigns one class as '1' and the second as '0' and discriminates them using a linear decision boundary. It is easy to implement and can be used as an effective baseline for binary classification problems. Being quick and effective on high-dimensional data, we use LR as a baseline classifier in this study. This model was implemented by using the python library *sklearn.linear_model.LogisticRegression*.

Logistic regression makes data assumptions such as independence and constant variance between the output and all values of the inputs. This simplicity and interpretability can occasionally lead to LR outperforming other sophisticated nonlinear models such as support vector machines [57], making it an effective baseline model.

In this project, LR will be utilised to classify a binary outcome (bankrupt or non-bankrupt). In this model the sigmoid function (Equation 4.1) is used as a Hypothesis Function (Equation 4.2) to map an instance/s denoted as x, with some given weights denoted as θ, to its Estimated Probability (Equation 4.3) of the discrete outcome. Here, x represents a feature vector for observation in our data.

$$g(z) = \frac{1}{1+\exp^{-z}} \tag{4.1}$$

$$\begin{aligned} h_\theta &= g\left(\theta^T x\right) \\ h_\theta &= \frac{1}{1+e^{-g(\theta^T x)}} \end{aligned} \tag{4.2}$$

The Hypothesis Function will classify an instance/s to its predicted outcome. Since the function will output the estimated probability of the discrete outcome, the model needs to interpret this output to be able to classify into '0' or '1'. To be able to compute this, a decision boundary is needed to predict $\hat{y} = 0$ or $\hat{y} = 1$, this can be seen in Equation 4.4.

$$\begin{aligned} P = (\hat{y} = 0|x;\theta) + P = (\hat{y} = 1|x;\theta) = 1 \\ P = (\hat{y} = 0|x;\theta) = 1 - P = (\hat{y} = 1|x;\theta) \end{aligned} \tag{4.3}$$

$$\hat{y} = \begin{cases} 1 & \sigma\left(\theta^T x\right) \geq 0.5; \theta^T x \geq 0 \\ 0 & \sigma\left(\theta^T x\right) < 0.5; \theta^T x < 0 \end{cases} \tag{4.4}$$

Now that the hypothesis is defined, and the classification function is explained the model must be trained to adjust its weights θ to minimise the cost. The cost function $J(\theta)$ utilised in Logistic Regression is shown in Equation 4.5 and this cost function is used over the squared cost function to be able to find the global minimum when applying gradient descent. A desirable property of $J(\theta)$ is that it greatly penalises wrong predictions which have high probability with a high cost.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y_i \log h_\theta\left(x_i\right) + \left(1 - y_i\right)\log\left(1 - h_\theta\left(x_i\right)\right) \tag{4.5}$$

The objective is to find a set of weights such that the negative log-likelihood is minimised over the defined training set using optimisation techniques like gradient descent. The loss function (Equation 4.5) measures the difference between the ground truth label and the predicted class label. If the prediction is very close to the ground truth label, the loss value is low. Alternatively, if the prediction is far from the true label, the resulting log loss will be higher.

4.3 Decision Trees

A decision tree [83] is a learning model based on a binary tree structure. Each internal node in the tree represents a yes/no question based on a single model covariate, with the data split based on the answer. The predicted outcome of each leaf node based on the majority class of the associated samples. Decisions trees are typically learnt by considering multiple candidate covariates and split points at each internal node and selecting the one that provides the biggest increase in the homogeneity of classes in the resulting subsets.

Various algorithms can grow a tree, they differ in the possible structure of the tree (e.g.the number of splits per node), the criteria how to find the splits, when to stop splitting and how to estimate the simple models within the leaf nodes. In this study, we have used the classification and regression trees (CART) algorithm for tree induction.

CART is a binary tree which is built by taking a set of instances from the training set and passing them to the root node, then a decision is made by using simple 'if-statements', based on the features. This is done to partition the instances into two subsets and pass them as an input to child nodes. This procedure is then applied to each node until it reaches a subset with the purest (no mixed classes) distributions of the classes, known as the leaf node. The most important aspect of this procedure is to know which decision should be taken at each node, in order to decrease the uncertainty (minimising the mixing of classes) of the subset. Firstly, a metric to quantify the uncertainty/purity at each node is needed and in this implementation Gini Impurity [18] is used as shown in Equation 4.6. The formula takes the node's class of each instance in the subset, for that node t, and computes the probability of obtaining two different outputs from all the possible classes k. If the subset only contains instances with the same label, the subset is pure, meaning $I(t) = 0$.

$$Gini\text{-}Index(t) = 1 - \sum_{j=1}^{k} p^2(j|t) \tag{4.6}$$

A smaller Gini index value of node n represents purity, which implies that the node contains more observations from a single class. Hence, a decreasing Gini index is an important criterion for node splitting.

Additionally, another metric is needed to quantify how much a certain decision reduces uncertainty, and for this Information Gain is used. A decision is simply a condition based on a specific feature and it is usually a $\geq$ check when it's a numeric value. Equation 4.7 shows how Information Gain is computed by taking the $I(t)$ of the current node and subtracting it by the total number of samples in the right node over the total number of samples in the parent node $\frac{N_1^2}{N_1}$ multiplied by the right node uncertainty $I(t_R)$ and subtracting again with weighted average of left node uncertainty $\frac{N_t L}{N_t} I(t_L)$ based on a specific decision.

$$I_g(t) = I(t) - \frac{N_{tR}}{N_t} I(t_R) - \frac{N_{tL}}{N_t} I(t_L) \tag{4.7}$$

So, at each node, every feature of each instance is checked using the Information Gain to find the best split. Once this is found the node is split into two child nodes taking as input the subset which met the criteria and the subset which did not meet the criteria. This is done recursively until there is no more further splits and the leaf nodes are reached. Once the tree is built, the tree can predict the class of unseen instances by passing the test dataset to the tree. These test instances follow down the tree taking the left or right nodes based on the criteria of the decision. Once a leaf node is reached the tree will predict the probability of that instance belonging to a specific class.

Decision Trees are better at handling situations where the relationship between features and outcome is nonlinear or where features interact with each other (logistic regression models fail in this regard). The performance of Decision Trees although higher than LR is relatively lower compared to other ensemble techniques (as shown in chapter 5). This is mainly due to two reasons [34]:

1. Noise present in the data.
2. Redundant attributes of data.

This model was implemented using the python library *sklearn.tree.DecisionTreeClassifier*, with the following hyper-parameters:

1. The strategy used to choose the split at each node: Best
2. The function to measure the quality of a split: Gini
3. The number of features to consider when looking for the best split: Number of Features

4.4 Random Forests

Random forests are ensembles of Decision Trees, the technique utilises multiple Decision Trees (weak learners) known as estimators to average out the predictions made by each tree, and this is done to reduce overfitting and to reduce the low bias, high variance trade-off found in a Decision Trees. Therefore Random forests are applied in various areas, including computer vision [15] and credit-scoring [19],

While random forests are not at the leading edge for classification tasks such as the deep learning variants technique [58, 102], we choose random forests as they have several desirable features [59], such as:

1. Random forests provide higher discriminatory ability because they assemble the decisions of a large number of trees, instead of a single tree.
2. They are more generalisable and are robust to over-fitting; hence, they may have better out-of-sample accuracy and be more robust to noise.
3. They are better at handling large datasets because they enable researchers to efficiently train multiple trees in parallel and do not require complex hyper-parameter settings. The researchers have to choose the number of decision trees to build a model.
4. They provide a measure of each variable's relative contribution to the prediction, which helps researchers identify and assess the influence of each variable while distinguishing between active and inactive companies. This can help us predict bankruptcy.

Random Forests are very simple to use and provide high performing models without loss of interpretability, which makes them suitable for real-world data [59].

Predictions from random forests are made using the following process:

1. Draw a subset of training data with random sampling by replacement (bootstrap).
2. Train a decision tree with the subset of training data. At each node of the tree, choose the best split of a variable from only the randomly selected m variables rather than from all the variables.
3. Repeat steps 1 and 2 to produce d decision trees.
4. Make predictions for new data by voting for the most popular class from among all of the output of the d decision trees.

This model was implemented using the python library *sklearn.ensemble.RandomForestClassifier*, with the following hyper-parameters mentioned below:

1. The number of trees in the forest = 100
2. The function to measure the quality of a split: Gini
3. The number of features to consider when looking for the best split: $\sqrt{Number\ of\ Features}$

4.5 Bagging

Breiman's bagging [17], also known as bootstrap aggregating, is one of the earliest ensemble learning algorithms. It is a technique involving independent classifiers that uses portions of the data and then combines them through model averaging, providing efficient results concerning a collection. It is designed to improve the stability and accuracy of classification [84] while simultaneously reducing variance to avoid overfitting.

Diversity in bagging is obtained by using bootstrapped replicas of the training dataset, i.e., different training data subsets are randomly drawn—with replacement—from the entire training dataset. Each training data subset is used to train a different base learner of the same type. The base learners' combination strategy for bagging is a majority vote. This simple strategy can reduce variance (hence prevent overfitting) when combined with the base learner generation strategies [43].

We have implemented the bagging algorithm as follows:

1. A random bootstrap set, t, is selected from the parent dataset.
2. Classifiers C_t are configured on the dataset from step 1.
3. Steps 1 and 2 are repeated for $t = 1, \ldots, T$
4. Each classifier determines a vote based on : $C(x) = T^{-1} \sum_{t=1}^{T} C_t(x)$
   where x is the data of each element from the training set. In the last step, the class that receives the largest number of votes is chosen as the classifier for the dataset.

This model was implemented using the python library *imblearn.ensemble.BalancedBaggingClassifier*, with the following hyper-parameters mentioned below:

1. The number of base estimators in the ensemble = 5
2. Base Learner = Random Forests
3. Bootstrap = True

4.6 Extreme Gradient Boosting Classifier (XGBoost)

XGBoost was chosen as a suitable model as it provides a strong regularisation framework that constrains overfitting [89]. The algorithm was developed to efficiently reduce computing time and allocate an optimal usage of memory resources. As mentioned in [28], important features of implementation include:

1. Handling of missing values (Sparse Aware).
2. Block Structure to support parallelisation in tree construction.
3. The ability to fit and boost new data added to a trained model (Continued Training).

As an ensemble tree-boosting method, XGBoost predicts a new classification membership after each iteration. This is done in an additive way, i.e., that predictions are made from weak classifiers, that constantly improve over the previous classifier's error. Incorrectly classified samples receive higher weights at the next step, forcing the classifier to focus on their performance in the following iterations. The final classification is vigorous as it includes the combined improvement of all the previous modelled trees. The learning of these classifiers is based on defining an objective function [29]. This function represents training loss and regularisation. The former describes the predictive accuracy of the model, while the latter describes the complexity. The working of XGBoost is now discussed in detail.

Let us denote by $\mathbf{x} \in X$ a vector of features describing an enterprise, where $X \subseteq \mathbb{R}^D$ and by $y \in \{0,1\}$ a label representing whether the enterprise is bankrupt, $y = 1$, or not, $y = 0$. Further, we utilise Decision Trees as discriminative models, specifically, we use CART. A CART tree can be represented by the weights associated with the leaves in the tree structure

$$f_k(\mathbf{x}_n) = w_{q(\mathbf{x})} \tag{4.8}$$

where $q(\mathbf{x}_n)$ is the function that takes an example $\mathbf{x}$ and returns the path id in the structure of the tree, $q : \mathbb{R}^D \to \{1, \cdots, T\}$, T is the number of paths (leaves). A path is ended with a leaf that contains weight w_i

Our algorithm aims at learning an ensemble of K decision trees [29] as shown below.

$$h_K(\mathbf{x}) = \sum_{k=1}^{K} f_k(\mathbf{x}) \tag{4.9}$$

where $f_k \in F$, for $k = 1, \ldots, K$, and F is a space of all possible decision trees (CART). To obtain a decision for new x, the conditional probability of a class for h_K is calculated as follows:

$$p(y = 1|\mathbf{x}) = \sigma(h_K(\mathbf{x})) \tag{4.10}$$

where sigma is the sigmoid function (seen in Equation 4.1). For given training data $D = \{\mathbf{x}_n, y_n\}_{n=1}^{N}$, the model is trained by minimising the following criterion:

$$\begin{aligned} L_{\Omega}(\theta) &= L(\theta) + \Omega(\theta) \\ &= \sum_{n=1}^{N} l\left(y_n, h_K\left(\mathbf{x}_n\right)\right) + \sum_{k=1}^{K} \Omega\left(f_k\right) \end{aligned} \tag{4.11}$$

The ensemble model for this loss function is known as the Logit-Boost model [29].

The problem of learning such model can be solved iteratively by adding a new weak learner $f_k(.)$ in the k^{th} training iteration assuming that models $f_1(\cdot), \cdots, f_{k-1}(\cdot)$ are already trained. We can present the loss function for single example $l\left(y_n, h_k\left(\mathbf{x}_n\right)\right)$ in the following manner:

$$l\left(y_n, h_k\left(\mathbf{x}_n\right)\right) = l\left(y_n, h_{k-1}\left(\mathbf{x}_n\right) + f_k\left(\mathbf{x}_n\right)\right) \tag{4.12}$$

We assumed an additive regularisation term, therefore we can represent it in the following form:

$$L_{\Omega}(\theta) = \sum_{n=1}^{N} l\left(y_n, h_{k-1}\left(\mathbf{x}_n\right) + f_k\left(\mathbf{x}_n\right)\right) + \Omega\left(f_k\right) + \text{ constant} \tag{4.13}$$

Therefore, we can represent the general learning criterion (Equation 4.11) as:

$$L_{\Omega}(\theta) = \sum_{n=1}^{N} l\left(y_n, h_{k-1}\left(\mathbf{x}_n\right) + f_k\left(\mathbf{x}_n\right)\right) + \Omega\left(f_k\right) + \text{ constant} \tag{4.14}$$

Further, approximating the objective function using the Taylor expansion with respect to $h_{k-1}\left(\mathbf{x}_n\right)$ yields the following:

$$\begin{aligned} L_{\Omega}(\theta) \simeq \sum_{n=1}^{N} &\left[l\left(y_n, h_{k-1}\left(\mathbf{x}_n\right)\right) + g_n \cdot f_k\left(\mathbf{x}_n\right) + \frac{1}{2} \cdot h_n \cdot f_k^2\left(\mathbf{x}_n\right) \right] \\ &+ \Omega\left(f_k\right) + \text{ constant} \end{aligned} \tag{4.15}$$

where g_n is the first derivative with respect to $h_{k-1}\left(\mathbf{x}_n\right)$

$$g_n = \frac{\partial l\left(y_n, h_{k-1}\left(\mathbf{x}_n\right)\right)}{\partial h_{k-1}\left(\mathbf{x}_n\right)} \tag{4.16}$$

and h_n is the second derivative with respect to $h_{k-1}\left(\mathbf{x}_n\right)$

$$h_n = \frac{\partial^2 l\left(y_n, h_{k-1}\left(\mathbf{x}_n\right)\right)}{\partial h_{k-1}^2\left(\mathbf{x}_n\right)} \tag{4.17}$$

Considering the logistic loss (Equation 4.11), g_n can be re-written as:

$$\begin{aligned} g_n &= -y_n \frac{\exp\{-h_{k-1}(\mathbf{x}_n)\}}{1+\exp\{-h_{k-1}(\mathbf{x}_n)\}} + (1-y_n)\frac{\exp\{h_{k-1}(\mathbf{x}_n)\}}{1+\exp\{h_{k-1}(\mathbf{x}_n)\}} \\ &= -y_n \frac{1}{1+\exp\{h_{k-1}(\mathbf{x}_n)\}} + (1-y_n)\frac{1}{1+\exp\{-h_{k-1}(\mathbf{x}_n)\}} \\ &= -y_n(1-\sigma(h_{k-1}(\mathbf{x}_n))) + (1-y_n)\sigma(h_{k-1}(\mathbf{x}_n)) \\ &= \sigma(h_{k-1}(\mathbf{x}_n)) - y_n \end{aligned} \tag{4.18}$$

In calculating the first derivative we took advantage of the sigmoid function property, namely, $\sigma(-a) = 1-\sigma(a)$. It can be observed, that $\sigma(h_{k-1}(\mathbf{x}_n))$ has interpretation of the probability of observing the class indexed by 1 for the example $\mathbf{x}_n$

We can make use of $\sigma'(a) = \sigma(a)(1-\sigma(a))$ property to calculate the second derivative, h_n.

$$h_n = \sigma(h_{k-1}(\mathbf{x}_n))(1-\sigma(h_{k-1}(\mathbf{x}_n))) \tag{4.19}$$

There are different possible regularisation terms. However, in our considerations we focus on the regulariser in the following form:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{t=1}^{T} w_t^2 \tag{4.20}$$

where λ and γ are the parameters of the regularisation term. For the tree representation with weights the objective function given in (Equation 4.14) can be presented in the following manner:

$$\begin{aligned} L_\Omega(\theta) &\simeq \sum_{n=1}^{N}\left[g_n w_{q(\mathbf{x}_n)} + \frac{1}{2}h_n \cdot w_{q(\mathbf{x}_n)}^2\right] + \gamma T \\ &+ \frac{1}{2}\lambda\sum_{t=1}^{T} w_t^2 + \text{ constant} \\ &= \sum_{t=1}^{T}\left[\left(\sum_{j\in I_I} g_j\right) w_t + \frac{1}{2}\left(\sum_{j\in I_t} h_j + \lambda\right) w_t^2\right] + \gamma T + \text{ constant} \\ &= \sum_{t=1}^{T}\left[G_t w_t + \frac{1}{2}(H_t+\lambda) w_t^2\right] + \gamma T + \text{ constant} \end{aligned} \tag{4.21}$$

where $I_t = \{n|q(\mathbf{x}_n) = t\}$ is the set of indexes of instances associated with the t -th leaf in the tree, $G_t = \sum_{j\in I_t} g_j$ and $H_t = \sum_{j\in I_t} h_j$. Assuming the known structure of the tree, the optimal value of the weight in the $t-$ th leaf is as follows:

$$w_t^* = -\frac{G_t}{H_t+\lambda} \tag{4.22}$$

The optimal value of the approximated objective function is given by

$$L_{\Omega}(\theta) \simeq -\frac{1}{2}\sum_{t=1}^{T}\frac{G_t^2}{H_t+\lambda}+\gamma T+\text{ constant} \tag{4.23}$$

The key problem in the above consideration is that the structure of the tree is not given in advanced and searching all possible structures is computationally not feasible. To overcome this issue the tree is being constructed starting from the root. Further, the best attribute located in the root is selected and the best split point for the attribute is chosen. The splitting process is performed until the quality of the model is improved. As the splitting criterion we take the Information Gain:

$$Gain = \frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_R+H_L+\lambda}-\gamma \tag{4.24}$$

where $\frac{G_L^2}{H_L+\lambda}$ is the score value calculated for the left child, $\frac{G_R^2}{H_R+\lambda}$ for is the score value for the right ancestor and $\frac{(G_L+G_R)^2}{H_R+H_L+\lambda}$ is the score value if splitting is not performed. Parameter γ penalizes addition of more leaves to the tree structure. The model can be also regularised by setting a minimum number of examples combined with each of the leaves, by setting the maximal depth of the tree, by setting the percentage of features randomised for each iteration of constructing the tree or by adding the new tree with the corrected influence of the trees in the committee

$$h_k(\mathbf{x}_n) = h_{k-1}(\mathbf{x}_n)+\varepsilon f_k(\mathbf{x}_n) \tag{4.25}$$

where $\varepsilon \in [0,1]$ is called step-size or shrinkage.

Parameter	**Domain**	**Default Parameter**	**After Tuning**
n_estimators	(100,500)	100.0	100
max_depth	(5,30)	3.0	9.0
learning_rate	(0.005,0.2)	0.1	0.2
min_child_weight	(1,50)	1.0	14.0
subsample	(0.8,1.0)	1.0	1.0
colsample_bytree	(0.8,1.0)	1.0	1.0
colsample_bylevel	(0.8,1.0)	1.0	1.0
gamma	(0,0.02)	0.0	0.0

Table 4.1: XGBoost Hyper-Parameters

This model was implemented using the python library *xgboost.XGBClassifier*. The domain of hyper-parameters was taken from those suggested in [110]. Starting with the default, we carried out a grid search of the parameters in that domain to obtain the best hyper-parameters for our task. This can be seen in Table 4.1.

4.7 Model Evaluation

In accordance with 2019's Basel III[1] regulations for evaluating bankruptcies and default models, it is crucial to choose a bankruptcy prediction model that scores well on a range of different performance metrics.

The accuracy measure of a model is not well-suited for imbalanced datasets and they can be largely ignored in a dataset where a large proportion of the observations belong to one class. The issue with the accuracy measure is that it does not look at class breakdown precision, nor does it provide evidence of true positives or true negatives values.

To conduct a credible performance evaluation that accounts for overfitting and generalisability, we measure the following metrics on each model:

1. Precision
2. Recall
3. F1-Score
4. Area under the Receiver Operating Characteristics (ROC) curve

4.7.1 Precision

Precision is as defined as the proportion of positive identifications that are actually correct (Equation 4.26). High Precision indicates an instance when labelled as positive is indeed positive (a small number of false positives). In our task, a high precision value indicates an instance that is classified to be bankrupt, is actually bankrupt.

$$Precision = \frac{True\text{-}Positive}{True\text{-}Positive + False\text{-}Positive} \tag{4.26}$$

4.7.2 Recall

Recall (Sensitivity) is defined as the proportion of actual positives that have been identified correctly (Equation 4.28). High Recall indicates that the class is correctly recognised (a small number of false negatives). The question recall address is: Of all the companies that have gone bankrupt, how many did the model label?

$$Recall = \frac{True\text{-}Positive}{True\text{-}Positive + False\text{-}Negative} \tag{4.27}$$

4.7.3 F1-Score

F1-score is a good performance metrics that leverages both precision and recall metrics. F1-score is obtained by taking the harmonic mean in place of arithmetic mean as it

[1] Basel III is an international regulatory accord that introduced a set of reforms designed to improve the regulation, supervision and risk management within the banking sector.

punishes the extreme values more. Unlike precision which mostly focuses on false-positive and recall which mostly focuses on false-negative, F1-score is a feasible metric that accounts for both false positive and false negative rates.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.28}$$

4.7.4 Area Under the Curve (AUC)

The receiver operating characteristic curve (ROC) is a plot of the true positive rate (TPR) versus the false positive rate (FPR) for the predictions of a binary classifier at multiple thresholds. The integrated area under the curve (AUC ROC) provides a summary measure of the discriminative ability of the model across all evaluated thresholds. AUC is desirable for the following two reasons:

1. AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values.
2. AUC is classification-threshold-invariant. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

Since performance is evaluated at multiple thresholds, AUC ROC is less sensitive to imbalanced class distributions compared to other commonly reported metrics (e.g. Accuracy, TPR, True Negative Rate (TNR)) making it suitable for this task [16]. The AUC of the ROC curve corresponds to the value of the WilcoxonMann-Whitney test, it is used as a measure of 'goodness' for predictions [107]. The range of AUC ROC values is between 0.5 and 1.0 with a value of 0.5 representing a classifier that is no better than randomly guessing the class and a value of 1.0 signifying a classifier with perfect discriminative ability.

Chapter 5

Empirical Results

We repeated 10-fold cross-validations for five times with different random seeds as is conducted by [56] in order to ensure that the comparison among three different classifiers does not happen by chance. For each of 10-fold cross-validation, the entire data set is first partitioned into 10 equal-sized sets, and then each set is in turn used as the test set while the classifier trains on the other nine sets. The cross-validated folds were tested independently of each algorithm. This way we obtained the results for four classifiers (and the baseline) on each of the fifty experiments. The empirical results reported in this section are the averages of the evaluation metrics used.

5.1 Baseline Results

We began our initial experiments by setting a valid baseline. We followed Altman's [5] footsteps and used Logistic Regression (LR) as our baseline classifier. Table 5.1 shows the results obtained. Examining these results, we make the following observations; first when data is not imputed LR performs worse than a random classifier. This is because LR is not good at handling missing data and performs poorly when it is not accounted for. Second, incorporating imputation techniques but leaving out the oversampling step brings about a significant increase (over 10%) in the AUC score. These models suffer from a low precision score suggesting that there are many false positives in the model. Third, the ideal baseline results are produced when both imputation and oversampling are used, particularly when kNN imputation is used. **This is our best performing baseline with an AUC score of 0.628 and an F1-score of 0.564.**

5.2 Results from Machine Learning Models

Table 5.2 shows the results we obtained, solely by applying our classifiers on the raw data, i.e., without imputing the data nor oversampling the minority class. As expected, the performance of our classifiers is very poor, this is evident from the low F1 and AUC scores. As the AUC scores are just over 0.5, these models are only marginally better than a random classification.

Model	Imputation	Oversampled	Precision	Recall	F1-Score	AUC
Logistic Regression	None	No	0.044	0.030	0.036	0.448
Logistic Regression	None	Yes	0.053	0.035	0.042	0.473
Logistic Regression	EM	No	0.064	0.360	0.108	0.583
Logistic Regression	KNN	No	0.068	0.370	0.115	0.592
Logistic Regression	MICE	No	0.068	0.370	0.115	0.592
Logistic Regression	EM	Yes	0.515	0.490	0.503	0.580
Logistic Regression	**KNN**	**Yes**	**0.570**	**0.559**	**0.564**	**0.628**
Logistic Regression	MICE	Yes	0.504	0.647	0.567	0.595

Table 5.1: Baseline Results

Model	Imputation	Precision	Recall	F1-Score	AUC
Decision Tree	None	0.070	0.120	0.088	0.507
Random Forest	None	0.102	0.251	0.145	0.512
Bagging	None	0.017	0.556	0.033	0.563
XGBoost	None	0.050	0.032	0.039	0.527

Table 5.2: Experiment results without data imputation nor minority class oversampling.

Following this, we incorporate the imputation techniques to estimate the missing values present in our data. Table 5.3 shows the results obtained by the classification models. We see that all the classifiers seem to be involved in a 'tug-of-war' match between precision and recall values.

A high recall, the low precision value indicates that most of the positive examples are correctly recognised (low false negatives) but there are a lot of false positives. This is seen in the Bagging ensemble model that achieves an AUC score of 0.746 with MICE imputation. Whereas a low recall value, high precision indicates that the model misses a lot of positive examples (high false negatives) but those we predicted as positive are indeed positive (low false positives). This can be observed in the Random Forests with MICE imputation model that achieves an AUC score of 0.739.

Next, we used SMOTE to oversample the minority class and to examine the effect that this process would have without performing data imputations. From Table 5.4 we note that the results are an improvement when compared to the experiment setup in Table 5.2; but the overall performance of these classifiers is still poor. The low precision

Model	Imputation	Precision	Recall	F1-Score	AUC
Decision Tree	EM	0.361	0.520	0.426	0.701
Decision Tree	KNN	0.414	0.550	0.472	0.721
Decision Tree	MICE	0.384	0.530	0.445	0.709
Random Forest	EM	0.800	0.080	0.145	0.540
Random Forest	KNN	0.650	0.130	0.217	0.563
Random Forest	MICE	0.894	0.420	0.572	0.739
Bagging	EM	0.121	0.670	0.205	0.705
Bagging	KNN	0.149	0.660	0.243	0.721
Bagging	MICE	0.171	0.770	0.280	0.746
XGBoost	EM	0.881	0.370	0.521	0.684
XGBoost	KNN	0.933	0.420	0.579	0.709
XGBoost	MICE	0.945	0.520	0.671	0.759

Table 5.3: Experiment results when data is imputed but not oversampled.

and recall values indicate that the model still suffers from high false-positive and high false-negative rates.

Model	Imputation	Precision	Recall	F1-Score	AUC
Decision Tree	None	0.084	0.138	0.104	0.684
Random Forest	None	0.122	0.289	0.172	0.691
Bagging	None	0.020	0.639	0.040	0.760
XGBoost	None	0.060	0.037	0.046	0.711

Table 5.4: Experiment results from oversampling the minority class to 15% of the data.

5.3 The Best Model

Finally, we combined the use of imputation techniques and oversampling of the minority class. Table 5.5 showcases a significant improvement across all the performance metrics. The combination of MICE imputation and oversampling achieves the highest performance amongst all the classifiers used. We conclude our experiments by reporting that XGBoost, when combined with MICE imputation, produces the best performing model to predict bankruptcies. **The model has the highest discriminative ability with an AUC score of 0.856. It achieves high precision and recall value of 0.774 and 0.804 respectively, resulting in a strong F1-Score of 0.798.**

5.4 Evaluating Feature Impact

Accepting the success of XGBoost, we proceed to further explore the most commonly used financial ratios in the forests of XGBoost.

Model	**Imputation**	**Precision**	**Recall**	**F1-Score**	**AUC**
Decision Tree	EM	0.650	0.784	0.710	0.740
Decision Tree	KNN	0.680	0.813	0.740	0.770
Decision Tree	MICE	0.767	0.775	0.771	0.803
Random Forest	EM	0.651	0.676	0.663	0.707
Random Forest	KNN	0.640	0.627	0.634	0.687
Random Forest	MICE	0.734	0.734	0.734	0.773
Bagging	EM	0.667	0.784	0.720	0.751
Bagging	KNN	0.694	0.824	0.753	0.781
Bagging	MICE	0.776	0.813	0.794	0.821
XGBoost	EM	0.701	0.784	0.740	0.812
XGBoost	KNN	0.702	0.765	0.731	0.806
XGBoost	**MICE**	**0.774**	**0.804**	**0.798**	**0.856**

Table 5.5: Experiment results after over-sampling minority class to 15% with SMOTE.

We evaluate the popularity of a feature in the XGBoost forest by describing the total number of occurrences in trees that constitutes the forest. Let us denote the total number of occurrences of the d^{th} feature in the forest structure by m_d. We define the categorical distribution $\theta_F = \left[\theta_F^{(1)}, \cdots, \theta_F^{(d)}, \cdots \theta_F^{(D)}\right]$ for selecting the features to be replicated in the following manner:

$$\theta_F^{(d)} = \frac{m_d}{\sum_{d=1}^{D} m_d} \tag{5.1}$$

Table 5.6 presents the twenty most important features that are considered in each of the classification cases. Analysing these results, we can ratiocinate that there are three key features in predicting bankruptcy, as they appeared in every research year. These features are:

1. **X25** - the adjusted share of equity in the financing of assets.
2. **X40** - current ratio, the most frequently used ratio in the integrated models. [118]
3. **X52** - liabilities turnover ratio.

Additionally, it is worth noting that the following features appeared in four out of five research years.

1. Profitability Ratios: X13, X22, X31, X42.
2. Leverage Ratios: X15.
3. Operating Performance Ratios: X9, X36, X48, X52.
4. Other Ratios: X5, X27, X58.

Further detail on this can be found in Table 3.2.

	Year 1		**Year 2**		**Year 3**		**Year 4**		**Year 5**	
Rank	**ID**	$\theta_F^{(d)}$	**ID**	$\theta_F^{(d)}$	**ID**	$\theta_F^{(d)}$	**ID**	$\theta_F^{(d)}$	**ID**	$\theta_F^{(d)}$
1.0	X16	0.051	X40	0.047	X15	0.050	X22	0.041	X25	0.062
2.0	X52	0.038	X15	0.044	X22	0.038	X52	0.044	X22	0.048
3.0	X32	0.037	X27	0.040	X52	0.036	X15	0.041	X27	0.037
4.0	X28	0.035	X5	0.034	X27	0.033	X25	0.038	X15	0.035
5.0	X5	0.034	X25	0.034	X40	0.032	X27	0.034	X52	0.032
6.0	X40	0.033	X36	0.033	X5	0.030	X40	0.032	X53	0.028
7.0	X9	0.031	X22	0.027	X25	0.026	X58	0.025	X14	0.024
8.0	X11	0.030	X42	0.027	X31	0.025	X42	0.025	X40	0.024
9.0	X59	0.030	X31	0.026	X12	0.025	X13	0.025	X42	0.023
10.0	X23	0.026	X13	0.026	X42	0.023	X36	0.023	X36	0.023
11.0	X25	0.024	X12	0.022	X13	0.023	X31	0.023	X54	0.023
12.0	X55	0.024	X35	0.021	X53	0.023	X5	0.023	X12	0.026
13.0	X17	0.023	X9	0.021	X57	0.022	X53	0.022	X58	0.021
14.0	X14	0.022	X58	0.021	X37	0.021	X6	0.021	X41	0.021
15.0	X29	0.021	X11	0.020	X48	0.020	X35	0.020	X44	0.019
16.0	X13	0.021	X48	0.020	X6	0.020	X48	0.020	X48	0.019
17.0	X58	0.021	X52	0.020	X35	0.018	X9	0.020	X9	0.019
18.0	X30	0.019	X57	0.020	X41	0.018	X24	0.019	X31	0.019
19.0	X57	0.019	X55	0.018	X32	0.018	X38	0.019	X32	0.019
20.0	X56	0.017	X6	0.0179	X36	0.018	X29	0.018	X16	0.018

Table 5.6: Ranking the top 20 features for each research year.

Chapter 6

Discussion

6.1 Success of MICE Imputation

The results obtained ascertain that all machine learning and ensemble models perform optimally under MICE imputation. MICE imputation is designed with the focus of handling missing data in large, public-use datasets. Under the missing-at-random (MAR) assumption [70], MICE enables more efficient and less biased estimation of model parameters. The imputation also works well on our data as the missing completely at random (MCAR) and missing not at random (MNAR) assumptions are not plausible as there is no randomness involved in the missing entries of this financial data i.e., the blank entries in the data are because the corporations did not record those indicators. The success of MICE imputation can additionally be attributed to the algorithm having a separate imputation model for each variable that needs to be estimated. MICE can also handle different variable types as well as complexities such as varying bounds [115]. The best performing model - XGBoost is unusable without MICE as its F1-Score is only 0.046 meaning that the model has very poor generalisability and a low discriminant power (to distinguish between the two classes). **After using MICE to estimate missing data, the F1-Score achieved by XGBoost is the highest amongst all experiments and the 20% improvement in AUC score is also noteworthy.**

6.2 Success of Oversampling the Minority Class

The oversampling technique has been used to overcome the data imbalance problem seen in our financial dataset. As the number of non-bankrupt instances exceeds that of the bankrupt instances, the non-bankrupt class is likely to invade the territory of the bankrupt class so that the class boundary is vulnerable to distortions. Therefore, this study uses oversampling as a method to generate more instances of the bankrupt class. This increases generalisability in our models and significantly improves performance. Table 5.3 reports XGBoost with an AUC and F1-score of 0.759 and 0.671 respectively. **After incorporating oversampling, these values are boosted to 0.856 and 0.798 respectively (as seen in Table 5.5) leading to the best performing model in our study.**

6.3 Success of Extreme Gradient Boosting (XGBoost)

The results are shown in Table 5.5 show that XGBoost proves to be the best ensemble classifier when applied to the bankruptcy prediction problem on the Japanese financial data. These results are in line with the success seen in past studies like [9, 61] that compare various machine learning models and conclusively report XGBoost as the most successful model on bankruptcy and credit default predictions.

The success of XGBoost can be attributed to its many advantages that include: being robust to outliers present in the input data, its ability to capture non-linear relationships in data and most importantly the use of L1 and L2 regularisation that the algorithm introduces to combat overfitting and improve generalisation. The paper [110] reviews this property to be "one of the most crucial aspects to the success of XGBoost". XGBoost also implements the 'Netwon-Rhapson Method' to optimise its loss. Instead of solely computing the gradient and following it, XGBoost uses the second-order derivative to firstly gather more information and secondly to make a better approximation of the direction of the maximum decrease and the step size required. Using this higher-order approximation (calculated by the Hessian matrix), XGBoost can achieve a better tree structure.

6.4 Most Influential Financial Ratios

Finally, to understand the most useful and impactful ratios for predicting bankruptcy, we analyse the popularity of features present in the forests of XGBoost. We note that X25 - the adjusted share of equity in the financing of assets, X40 - the current ratio and X52 - the liabilities turnover ratio appear in each research year (as shown in Table 5.6). These financial ratios are therefore presented as the most influential financial ratios, for the Japanese financial data.

6.5 Limitations

Although the methods used in this study to predict bankruptcies are manifold, there are still a few limitations that are worthy of being discussed.

First and foremost, the credibility of our prediction is based solely on the authenticity and integrity of the data provided to WRDS. Manipulating financial statements is relatively easy to do [2] as the GAAP standards afford a significant amount of flexibility, making it feasible for corporate management to falsely report the financial condition of the company. Also, the compensation of corporate executives is directly tied to the financial performance of the company. As a result, they have a direct incentive to report positive performance every year. Since we do not have the resources to validate the authenticity of the data ourselves, we assume that the data provided is indeed accurate.

Second, the use of 'feature selection' as a further pre-processing method could not be effectively incorporated in this study. Feature selection is employed to select and extract more valuable information from the vast amounts of data. That is, it aims at filtering out redundant or irrelevant information, and consequently can improve the model's performance as well as reduce the effort of training the model.

In this paper, we attempted to use PCA, Chi-Square, and Recursive Feature Elimination to improve the model's performance, but due to the unforeseen circumstances (the COVID-19 pandemic and the nation-wide lockdown) that lead to the closure of Appleton Tower and the restricted access to Edinburgh University servers, we were unable to conduct detailed and conclusive experiments within the set time-frame.

Third, using oversampling methods such as SMOTE can make the problem harder by duplicating the existing outliers and noisy examples. Further noise is added when minority samples around the boundary (between bankrupt and non-bankrupt) are duplicated in their neighbourhood. This study suffers this limitation as we have not accounted for outliers that are synthetically generated.

Fourth, there are three main strategies used for finding the proper setting of hyperparameters in tree-based algorithms: grid search (GS), random search (RS), and Bayesian Tree-structured Parzen estimators (TPE). They have demonstrated a substantial influence on classification performance. As our research has had a time-bound constraint, we have not been able to explore these strategies to their fullest extent. Exploring these strategies could potentially lead to discovering models that have better performance.

Finally, our current study is limited by only using financial ratios as feature vectors. We acknowledge that recent studies ([40, 49, 105]) have incorporated various combinations of accounting data, stock market information, corporate governance indicators and sentiment analysis of articles published on social media platforms and other textual disclosure data. I believe that incorporating more data would give us an additional edge to predict bankruptcies within a shorter horizon as a company's health would be represented comprehensively.

Chapter 7

Conclusion

Bankruptcy prediction has long been regarded as a critical topic and has been studied extensively in the accounting and finance literature and now more recently in data science and artificial intelligence communities.

The Asian economy is often ignored in financial research - the paucity of published research is representative of this. As Japan has the third-highest market capitalisation in the world, we chose to carry out our research on the Japanese economy. To the best of our knowledge, this paper puts forth a novel contribution to the field of bankruptcy prediction by using a ten-year rolling window on Japanese financial data and then predicting bankruptcy on a one-year horizon.

We began by conducting a meticulous data analysis on the financial health of 4000 listed Japanese companies from 2000 to 2018. The most difficult challenge we faced was cleaning and re-structuring this dataset. Since all the companies being evaluated for bankruptcy don't follow GAAP standards, the frequency and type of data reported varied significantly. We had to be very precise while gathering data from different files and had to ensure that a particular indicator was consistently reported for the same month/year.

The features on which we ultimately modelled our bankruptcy prediction were not directly available in our dataset. We generated the financial ratios for our model after extensively studying the previous research on bankruptcy prediction. Consequently, we contribute a cleaned, aligned and pre-processed Japanese financial dataset that can be made available for future studies to carry out further research.

To overcome the two biggest shortcomings of the dataset - missing data and class imbalance, we introduced Expectation-Maximisation, kNN and MICE as three different imputation techniques along with Synthetic Minority Oversampling Technique (SMOTE) as a technique to oversample minority class data.

Research from [81, 19] suggests that using a single classifier cannot solve all problems effectively whereas ensemble models have been revealed to be promising in previous credit risk and bankruptcy forecasting studies. Taking this into account, we began our research by establishing a valid baseline using Logistic Regression and then proceeded to use tree-based machine learning and ensemble models. We started with CART Decision Trees and then explored ensemble models such as Random Forests, Bagging and XGBoost.

A leading contribution of our research is a detailed comparison between imputation techniques to estimate missing financial data. We submit a case to use Multiple Imputation by Chained Equations for data estimation, as it outperformed all other imputation methods used in this study. The rigorous experiments conducted, lead to a considerable improvement in the quality of prediction performance. We conclude by reporting that the best model was achieved by using MICE to estimate missing data, SMOTE to oversample the minority class data and XGBoost as the ensemble learning model. The model achieves an AUC score of 0.856 and an F1-Score of 0.798 and the results are in line with the research seen in [9, 61], where the authors report XGBoost as the best model to predict bankruptcy among North American companies.

We also report that the most influential financial ratios in predicting bankruptcies of listed Japanese companies are, the adjusted share of equity in the financing assets, the current ratio and the liabilities turnover ratio.

Since our model uses financial ratios that can be created from data reported by most listed companies, the solution we propose is not limited to listed Japanese companies but can represent a general framework that can be applied to any economic and financial dataset.

As bankruptcy prediction encapsulates a vast domain, we would like to focus our attention towards researching the applicability of neural networks. In particular, we would like to explore the potential of ensembles models that use neural networks as base learners, to test whether they can achieve higher performance as compared to the current tree-based algorithms we used in our research. Results on listed firms in Korea [56] indicates that the bagged and the boosted neural networks show the improved performance over traditional neural networks. Additionally, the use of shallow convolutional neural nets presented in Dr Tiejun Ma's paper [80] have achieved high accuracy on predicting stock prices and we would like to see whether that hypothesis could be extended to predicting bankruptcy.

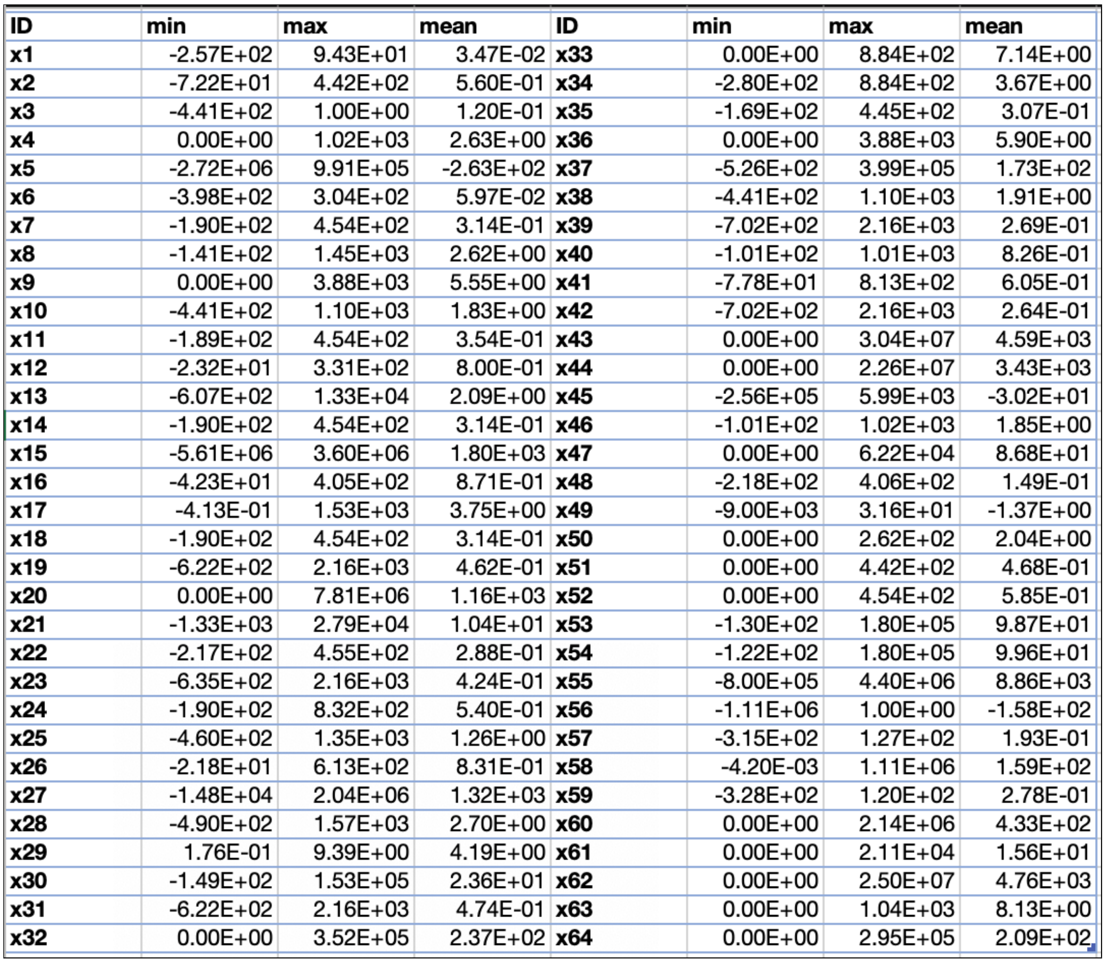
Appendix A

Concept Drift

Data is subject to concept drift when our interpretation of the data changes with time even while the general distribution of the data does not. This change led to complications in machine learning tasks. Schlimmer and Granger [87] defined concept drift as the process changes of hidden environment that led to changes of target concept. Widmer and Kubat [114] called real concept drift as the process in which recessive changes in the environment led to changes of target concept, and called virtual concept drift as the process in which environmental changes led to changes of distribution of dataset. These two types of concept drifts may occur simultaneously, and virtual concept drift may occur in isolation. No matter the concept drift was real or virtual, they would make the model based on old data inconsistent with new data. Thus, it is necessary to account for concept drift when designing machine learning models on historic data.

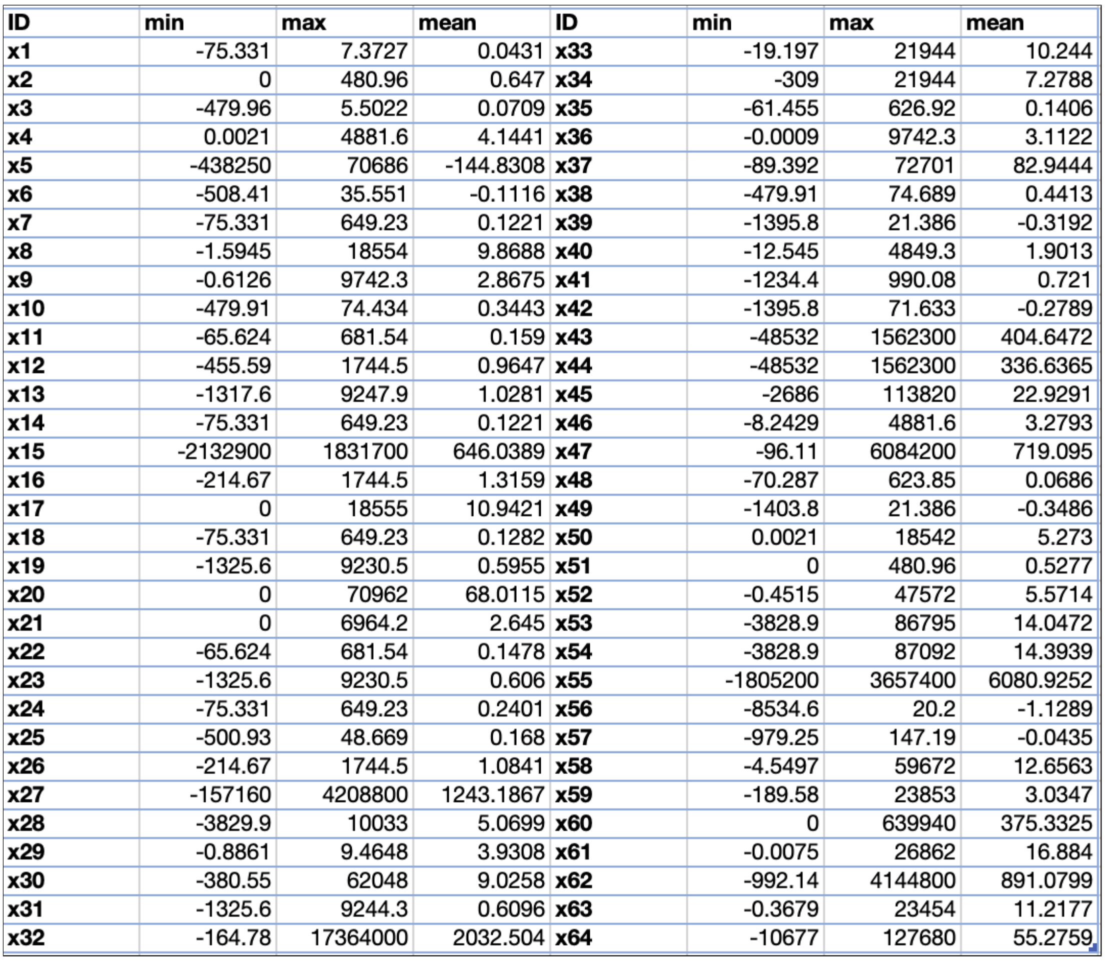
Appendix B

Data Ranges



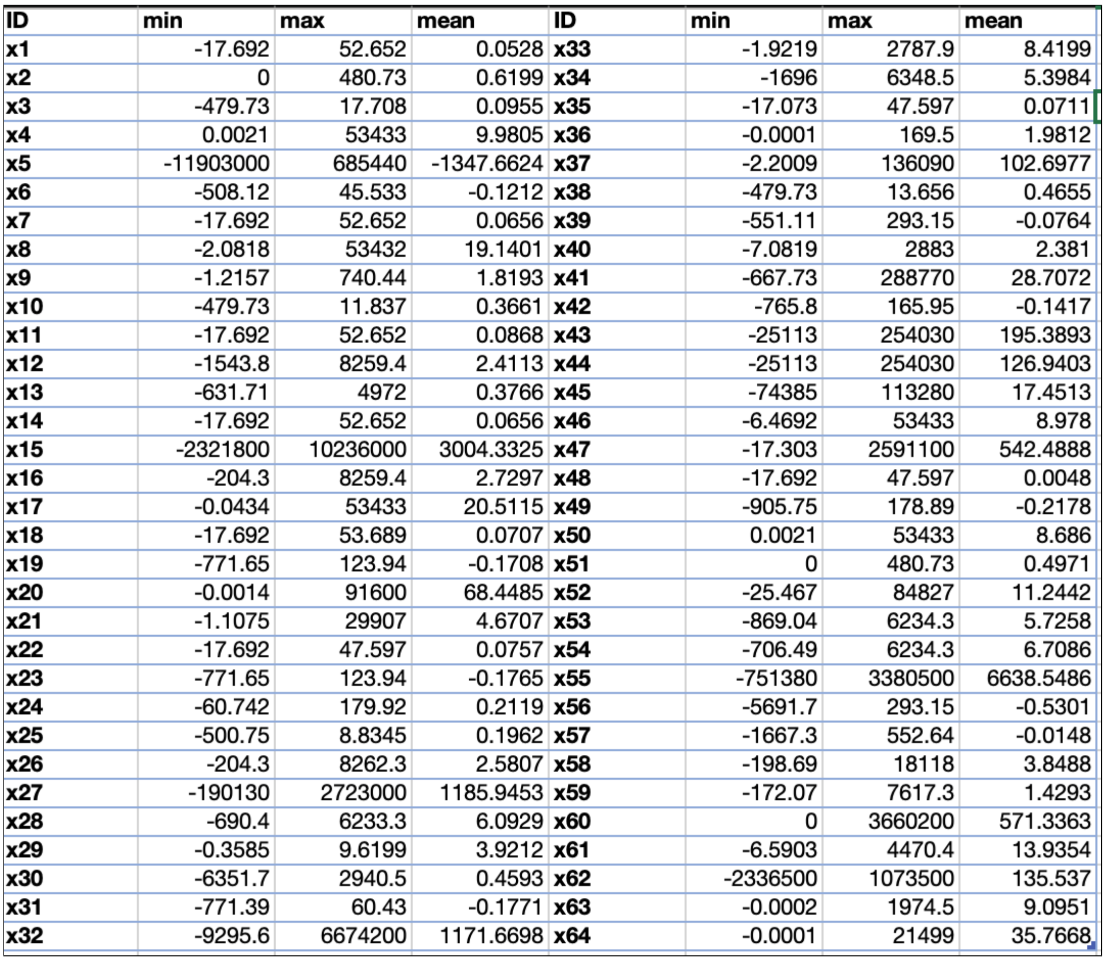
ID	min	max	mean	ID	min	max	mean
x1	-2.57E+02	9.43E+01	3.47E-02	**x33**	0.00E+00	8.84E+02	7.14E+00
x2	-7.22E+01	4.42E+02	5.60E-01	**x34**	-2.80E+02	8.84E+02	3.67E+00
x3	-4.41E+02	1.00E+00	1.20E-01	**x35**	-1.69E+02	4.45E+02	3.07E-01
x4	0.00E+00	1.02E+03	2.63E+00	**x36**	0.00E+00	3.88E+03	5.90E+00
x5	-2.72E+06	9.91E+05	-2.63E+02	**x37**	-5.26E+02	3.99E+05	1.73E+02
x6	-3.98E+02	3.04E+02	5.97E-02	**x38**	-4.41E+02	1.10E+03	1.91E+00
x7	-1.90E+02	4.54E+02	3.14E-01	**x39**	-7.02E+02	2.16E+03	2.69E-01
x8	-1.41E+02	1.45E+03	2.62E+00	**x40**	-1.01E+02	1.01E+03	8.26E-01
x9	0.00E+00	3.88E+03	5.55E+00	**x41**	-7.78E+01	8.13E+02	6.05E-01
x10	-4.41E+02	1.10E+03	1.83E+00	**x42**	-7.02E+02	2.16E+03	2.64E-01
x11	-1.89E+02	4.54E+02	3.54E-01	**x43**	0.00E+00	3.04E+07	4.59E+03
x12	-2.32E+01	3.31E+02	8.00E-01	**x44**	0.00E+00	2.26E+07	3.43E+03
x13	-6.07E+02	1.33E+04	2.09E+00	**x45**	-2.56E+05	5.99E+03	-3.02E+01
x14	-1.90E+02	4.54E+02	3.14E-01	**x46**	-1.01E+02	1.02E+03	1.85E+00
x15	-5.61E+06	3.60E+06	1.80E+03	**x47**	0.00E+00	6.22E+04	8.68E+01
x16	-4.23E+01	4.05E+02	8.71E-01	**x48**	-2.18E+02	4.06E+02	1.49E-01
x17	-4.13E-01	1.53E+03	3.75E+00	**x49**	-9.00E+03	3.16E+01	-1.37E+00
x18	-1.90E+02	4.54E+02	3.14E-01	**x50**	0.00E+00	2.62E+02	2.04E+00
x19	-6.22E+02	2.16E+03	4.62E-01	**x51**	0.00E+00	4.42E+02	4.68E-01
x20	0.00E+00	7.81E+06	1.16E+03	**x52**	0.00E+00	4.54E+02	5.85E-01
x21	-1.33E+03	2.79E+04	1.04E+01	**x53**	-1.30E+02	1.80E+05	9.87E+01
x22	-2.17E+02	4.55E+02	2.88E-01	**x54**	-1.22E+02	1.80E+05	9.96E+01
x23	-6.35E+02	2.16E+03	4.24E-01	**x55**	-8.00E+05	4.40E+06	8.86E+03
x24	-1.90E+02	8.32E+02	5.40E-01	**x56**	-1.11E+06	1.00E+00	-1.58E+02
x25	-4.60E+02	1.35E+03	1.26E+00	**x57**	-3.15E+02	1.27E+02	1.93E-01
x26	-2.18E+01	6.13E+02	8.31E-01	**x58**	-4.20E-03	1.11E+06	1.59E+02
x27	-1.48E+04	2.04E+06	1.32E+03	**x59**	-3.28E+02	1.20E+02	2.78E-01
x28	-4.90E+02	1.57E+03	2.70E+00	**x60**	0.00E+00	2.14E+06	4.33E+02
x29	1.76E-01	9.39E+00	4.19E+00	**x61**	0.00E+00	2.11E+04	1.56E+01
x30	-1.49E+02	1.53E+05	2.36E+01	**x62**	0.00E+00	2.50E+07	4.76E+03
x31	-6.22E+02	2.16E+03	4.74E-01	**x63**	0.00E+00	1.04E+03	8.13E+00
x32	0.00E+00	3.52E+05	2.37E+02	**x64**	0.00E+00	2.95E+05	2.09E+02

Figure B.1: Min, Max and Mean for Year 1



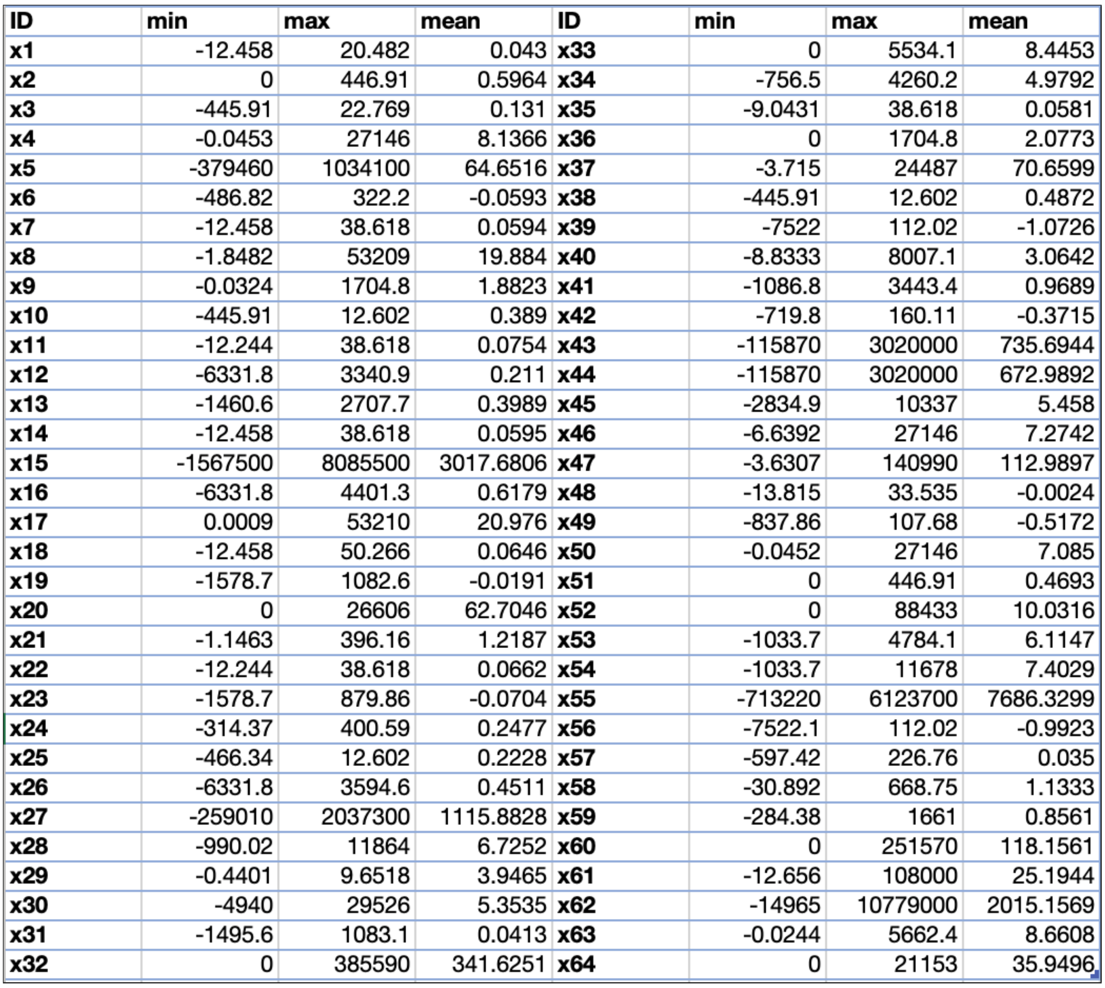
ID	min	max	mean	ID	min	max	mean
x1	-75.331	7.3727	0.0431	x33	-19.197	21944	10.244
x2	0	480.96	0.647	x34	-309	21944	7.2788
x3	-479.96	5.5022	0.0709	x35	-61.455	626.92	0.1406
x4	0.0021	4881.6	4.1441	x36	-0.0009	9742.3	3.1122
x5	-438250	70686	-144.8308	x37	-89.392	72701	82.9444
x6	-508.41	35.551	-0.1116	x38	-479.91	74.689	0.4413
x7	-75.331	649.23	0.1221	x39	-1395.8	21.386	-0.3192
x8	-1.5945	18554	9.8688	x40	-12.545	4849.3	1.9013
x9	-0.6126	9742.3	2.8675	x41	-1234.4	990.08	0.721
x10	-479.91	74.434	0.3443	x42	-1395.8	71.633	-0.2789
x11	-65.624	681.54	0.159	x43	-48532	1562300	404.6472
x12	-455.59	1744.5	0.9647	x44	-48532	1562300	336.6365
x13	-1317.6	9247.9	1.0281	x45	-2686	113820	22.9291
x14	-75.331	649.23	0.1221	x46	-8.2429	4881.6	3.2793
x15	-2132900	1831700	646.0389	x47	-96.11	6084200	719.095
x16	-214.67	1744.5	1.3159	x48	-70.287	623.85	0.0686
x17	0	18555	10.9421	x49	-1403.8	21.386	-0.3486
x18	-75.331	649.23	0.1282	x50	0.0021	18542	5.273
x19	-1325.6	9230.5	0.5955	x51	0	480.96	0.5277
x20	0	70962	68.0115	x52	-0.4515	47572	5.5714
x21	0	6964.2	2.645	x53	-3828.9	86795	14.0472
x22	-65.624	681.54	0.1478	x54	-3828.9	87092	14.3939
x23	-1325.6	9230.5	0.606	x55	-1805200	3657400	6080.9252
x24	-75.331	649.23	0.2401	x56	-8534.6	20.2	-1.1289
x25	-500.93	48.669	0.168	x57	-979.25	147.19	-0.0435
x26	-214.67	1744.5	1.0841	x58	-4.5497	59672	12.6563
x27	-157160	4208800	1243.1867	x59	-189.58	23853	3.0347
x28	-3829.9	10033	5.0699	x60	0	639940	375.3325
x29	-0.8861	9.4648	3.9308	x61	-0.0075	26862	16.884
x30	-380.55	62048	9.0258	x62	-992.14	4144800	891.0799
x31	-1325.6	9244.3	0.6096	x63	-0.3679	23454	11.2177
x32	-164.78	17364000	2032.504	x64	-10677	127680	55.2759

Figure B.2: Min, Max and Mean for Year 2



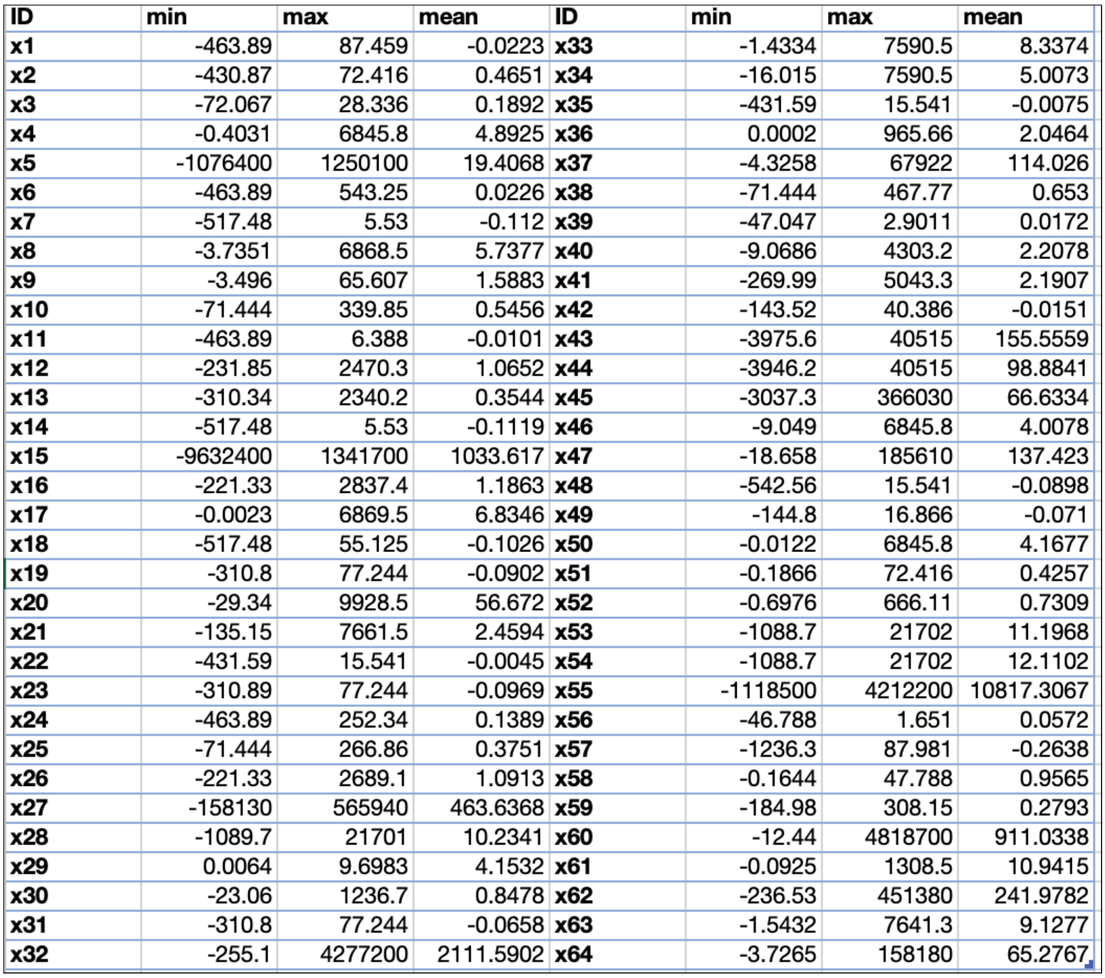
ID	min	max	mean	ID	min	max	mean
x1	-17.692	52.652	0.0528	x33	-1.9219	2787.9	8.4199
x2	0	480.73	0.6199	x34	-1696	6348.5	5.3984
x3	-479.73	17.708	0.0955	x35	-17.073	47.597	0.0711
x4	0.0021	53433	9.9805	x36	-0.0001	169.5	1.9812
x5	-11903000	685440	-1347.6624	x37	-2.2009	136090	102.6977
x6	-508.12	45.533	-0.1212	x38	-479.73	13.656	0.4655
x7	-17.692	52.652	0.0656	x39	-551.11	293.15	-0.0764
x8	-2.0818	53432	19.1401	x40	-7.0819	2883	2.381
x9	-1.2157	740.44	1.8193	x41	-667.73	288770	28.7072
x10	-479.73	11.837	0.3661	x42	-765.8	165.95	-0.1417
x11	-17.692	52.652	0.0868	x43	-25113	254030	195.3893
x12	-1543.8	8259.4	2.4113	x44	-25113	254030	126.9403
x13	-631.71	4972	0.3766	x45	-74385	113280	17.4513
x14	-17.692	52.652	0.0656	x46	-6.4692	53433	8.978
x15	-2321800	10236000	3004.3325	x47	-17.303	2591100	542.4888
x16	-204.3	8259.4	2.7297	x48	-17.692	47.597	0.0048
x17	-0.0434	53433	20.5115	x49	-905.75	178.89	-0.2178
x18	-17.692	53.689	0.0707	x50	0.0021	53433	8.686
x19	-771.65	123.94	-0.1708	x51	0	480.73	0.4971
x20	-0.0014	91600	68.4485	x52	-25.467	84827	11.2442
x21	-1.1075	29907	4.6707	x53	-869.04	6234.3	5.7258
x22	-17.692	47.597	0.0757	x54	-706.49	6234.3	6.7086
x23	-771.65	123.94	-0.1765	x55	-751380	3380500	6638.5486
x24	-60.742	179.92	0.2119	x56	-5691.7	293.15	-0.5301
x25	-500.75	8.8345	0.1962	x57	-1667.3	552.64	-0.0148
x26	-204.3	8262.3	2.5807	x58	-198.69	18118	3.8488
x27	-190130	2723000	1185.9453	x59	-172.07	7617.3	1.4293
x28	-690.4	6233.3	6.0929	x60	0	3660200	571.3363
x29	-0.3585	9.6199	3.9212	x61	-6.5903	4470.4	13.9354
x30	-6351.7	2940.5	0.4593	x62	-2336500	1073500	135.537
x31	-771.39	60.43	-0.1771	x63	-0.0002	1974.5	9.0951
x32	-9295.6	6674200	1171.6698	x64	-0.0001	21499	35.7668

Figure B.3: Min, Max and Mean for Year 3



ID	min	max	mean	ID	min	max	mean
x1	-12.458	20.482	0.043	x33	0	5534.1	8.4453
x2	0	446.91	0.5964	x34	-756.5	4260.2	4.9792
x3	-445.91	22.769	0.131	x35	-9.0431	38.618	0.0581
x4	-0.0453	27146	8.1366	x36	0	1704.8	2.0773
x5	-379460	1034100	64.6516	x37	-3.715	24487	70.6599
x6	-486.82	322.2	-0.0593	x38	-445.91	12.602	0.4872
x7	-12.458	38.618	0.0594	x39	-7522	112.02	-1.0726
x8	-1.8482	53209	19.884	x40	-8.8333	8007.1	3.0642
x9	-0.0324	1704.8	1.8823	x41	-1086.8	3443.4	0.9689
x10	-445.91	12.602	0.389	x42	-719.8	160.11	-0.3715
x11	-12.244	38.618	0.0754	x43	-115870	3020000	735.6944
x12	-6331.8	3340.9	0.211	x44	-115870	3020000	672.9892
x13	-1460.6	2707.7	0.3989	x45	-2834.9	10337	5.458
x14	-12.458	38.618	0.0595	x46	-6.6392	27146	7.2742
x15	-1567500	8085500	3017.6806	x47	-3.6307	140990	112.9897
x16	-6331.8	4401.3	0.6179	x48	-13.815	33.535	-0.0024
x17	0.0009	53210	20.976	x49	-837.86	107.68	-0.5172
x18	-12.458	50.266	0.0646	x50	-0.0452	27146	7.085
x19	-1578.7	1082.6	-0.0191	x51	0	446.91	0.4693
x20	0	26606	62.7046	x52	0	88433	10.0316
x21	-1.1463	396.16	1.2187	x53	-1033.7	4784.1	6.1147
x22	-12.244	38.618	0.0662	x54	-1033.7	11678	7.4029
x23	-1578.7	879.86	-0.0704	x55	-713220	6123700	7686.3299
x24	-314.37	400.59	0.2477	x56	-7522.1	112.02	-0.9923
x25	-466.34	12.602	0.2228	x57	-597.42	226.76	0.035
x26	-6331.8	3594.6	0.4511	x58	-30.892	668.75	1.1333
x27	-259010	2037300	1115.8828	x59	-284.38	1661	0.8561
x28	-990.02	11864	6.7252	x60	0	251570	118.1561
x29	-0.4401	9.6518	3.9465	x61	-12.656	108000	25.1944
x30	-4940	29526	5.3535	x62	-14965	10779000	2015.1569
x31	-1495.6	1083.1	0.0413	x63	-0.0244	5662.4	8.6608
x32	0	385590	341.6251	x64	0	21153	35.9496

Figure B.4: Min, Max and Mean for Year 4



ID	min	max	mean	ID	min	max	mean
x1	-463.89	87.459	-0.0223	**x33**	-1.4334	7590.5	8.3374
x2	-430.87	72.416	0.4651	**x34**	-16.015	7590.5	5.0073
x3	-72.067	28.336	0.1892	**x35**	-431.59	15.541	-0.0075
x4	-0.4031	6845.8	4.8925	**x36**	0.0002	965.66	2.0464
x5	-1076400	1250100	19.4068	**x37**	-4.3258	67922	114.026
x6	-463.89	543.25	0.0226	**x38**	-71.444	467.77	0.653
x7	-517.48	5.53	-0.112	**x39**	-47.047	2.9011	0.0172
x8	-3.7351	6868.5	5.7377	**x40**	-9.0686	4303.2	2.2078
x9	-3.496	65.607	1.5883	**x41**	-269.99	5043.3	2.1907
x10	-71.444	339.85	0.5456	**x42**	-143.52	40.386	-0.0151
x11	-463.89	6.388	-0.0101	**x43**	-3975.6	40515	155.5559
x12	-231.85	2470.3	1.0652	**x44**	-3946.2	40515	98.8841
x13	-310.34	2340.2	0.3544	**x45**	-3037.3	366030	66.6334
x14	-517.48	5.53	-0.1119	**x46**	-9.049	6845.8	4.0078
x15	-9632400	1341700	1033.617	**x47**	-18.658	185610	137.423
x16	-221.33	2837.4	1.1863	**x48**	-542.56	15.541	-0.0898
x17	-0.0023	6869.5	6.8346	**x49**	-144.8	16.866	-0.071
x18	-517.48	55.125	-0.1026	**x50**	-0.0122	6845.8	4.1677
x19	-310.8	77.244	-0.0902	**x51**	-0.1866	72.416	0.4257
x20	-29.34	9928.5	56.672	**x52**	-0.6976	666.11	0.7309
x21	-135.15	7661.5	2.4594	**x53**	-1088.7	21702	11.1968
x22	-431.59	15.541	-0.0045	**x54**	-1088.7	21702	12.1102
x23	-310.89	77.244	-0.0969	**x55**	-1118500	4212200	10817.3067
x24	-463.89	252.34	0.1389	**x56**	-46.788	1.651	0.0572
x25	-71.444	266.86	0.3751	**x57**	-1236.3	87.981	-0.2638
x26	-221.33	2689.1	1.0913	**x58**	-0.1644	47.788	0.9565
x27	-158130	565940	463.6368	**x59**	-184.98	308.15	0.2793
x28	-1089.7	21701	10.2341	**x60**	-12.44	4818700	911.0338
x29	0.0064	9.6983	4.1532	**x61**	-0.0925	1308.5	10.9415
x30	-23.06	1236.7	0.8478	**x62**	-236.53	451380	241.9782
x31	-310.8	77.244	-0.0658	**x63**	-1.5432	7641.3	9.1277
x32	-255.1	4277200	2111.5902	**x64**	-3.7265	158180	65.2767

Figure B.5: Min, Max and Mean for Year 5

Appendix C

Feature Distributions

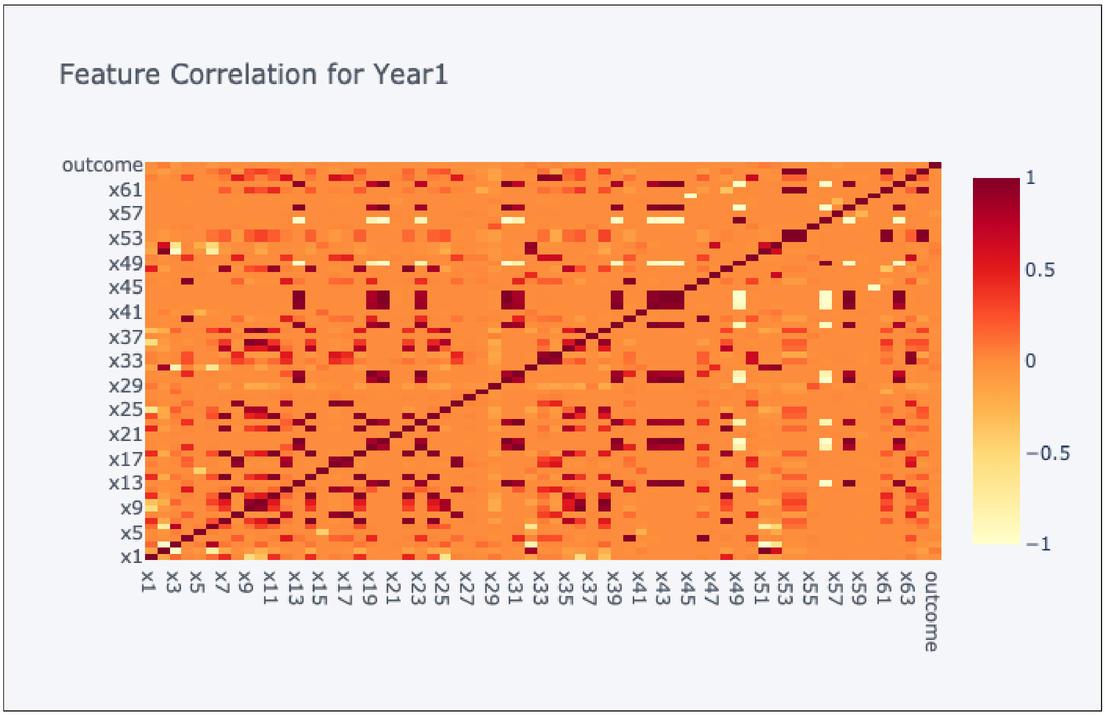


Figure C.1: Correlation Heatmap for Year 1

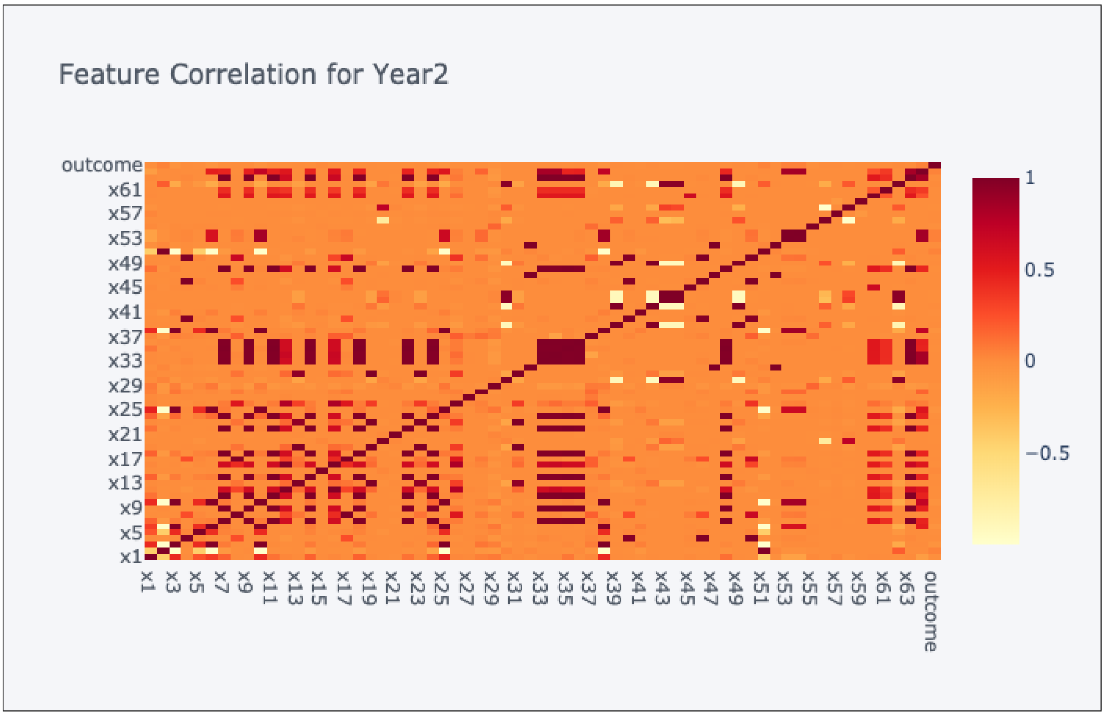


Figure C.2: Correlation Heatmap for Year 2

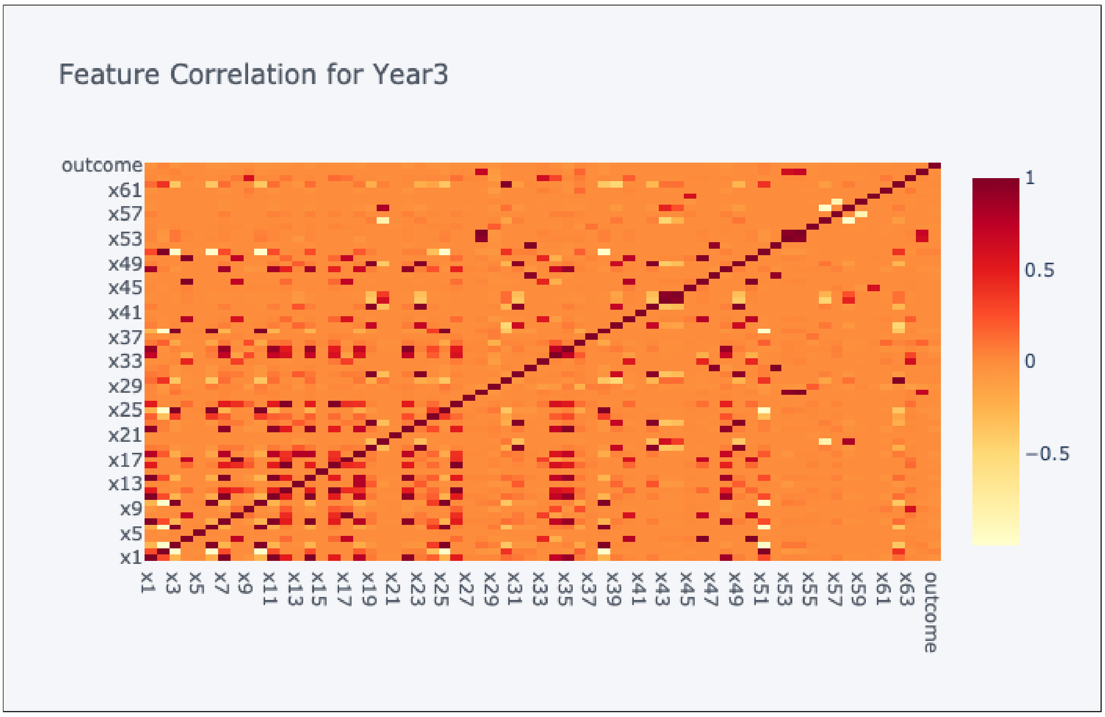


Figure C.3: Correlation Heatmap for Year 3

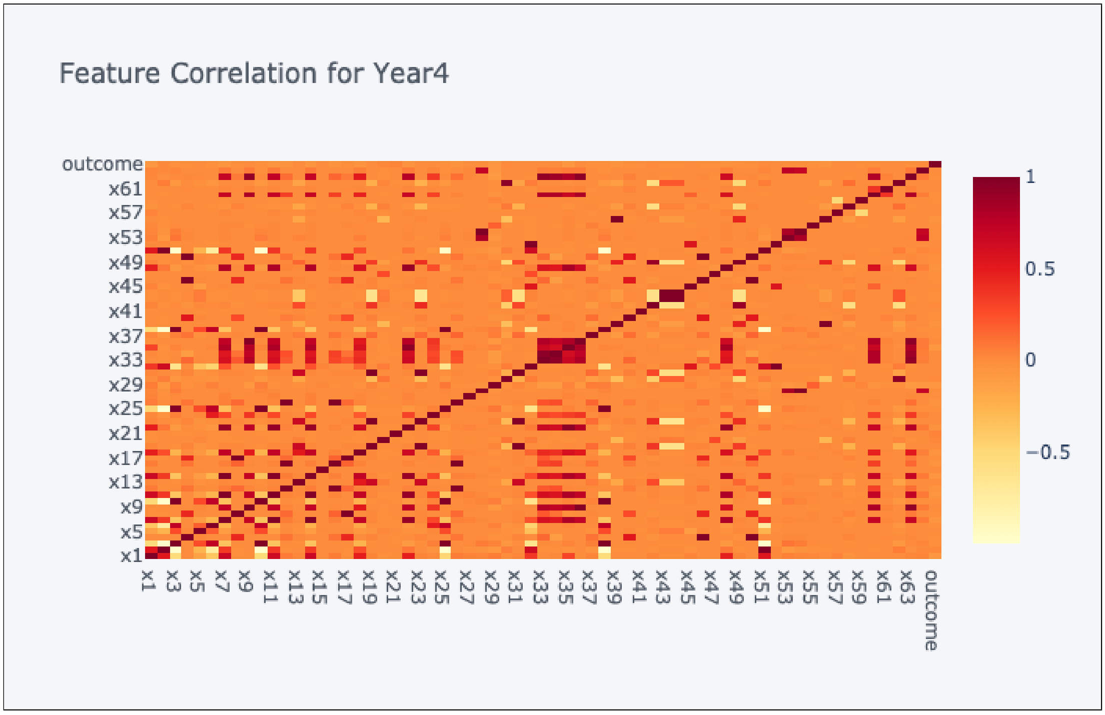


Figure C.4: Correlation Heatmap for Year 4

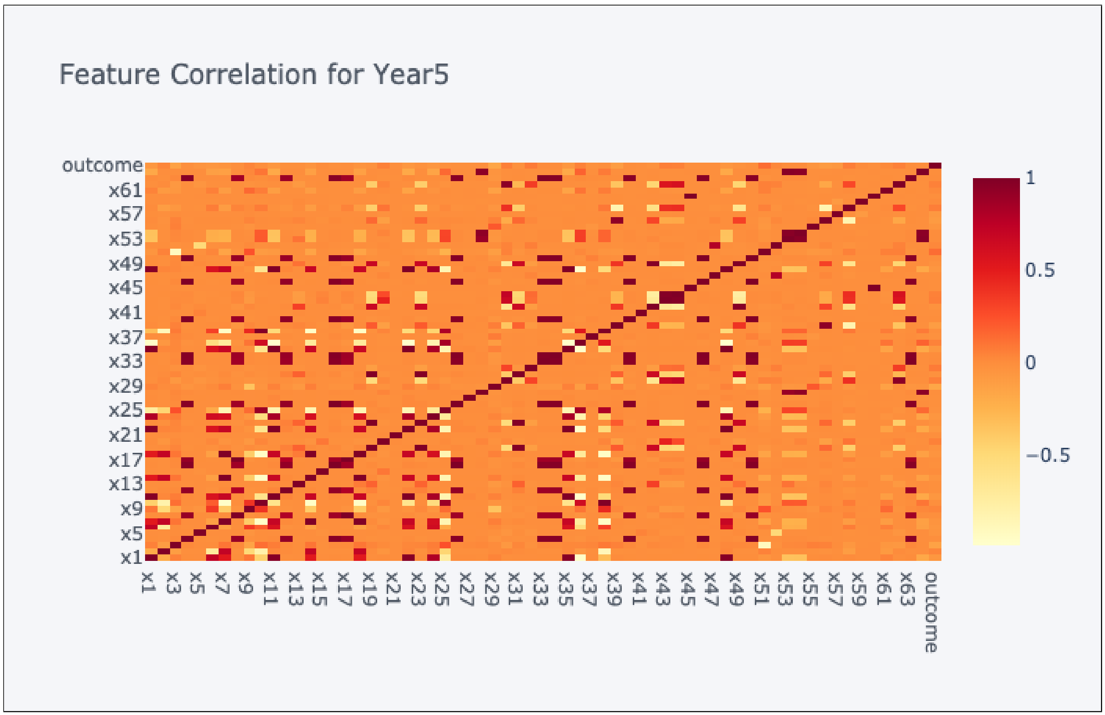


Figure C.5: Correlation Heatmap for Year 5

Appendix D

Python Libraries used for Project

Library	**Description**
numpy	Data organisation and statistical operations.
pandas	Data manipulation and analysis. Storing and manipulating numerical tables.
matplotlib, plotly and seaborn	Plotting libraries
missingno	Generate nullity matrices and correlation heatmaps for missing data.
fancyimpute	Perform k-NN and MICE imputation
impyute	Perform EM imputation
sklearn.model_selection.StratifiedKFold	Perform Stratified K-Fold Cross Validation
imblearn.over_sampling.SMOTE	Perform SMOTE Oversampling
sklearn.linear_model.LogisticRegression	Logistic Regression Classifier
sklearn.tree.DecisionTreeClassifier	Decision Tree Classifier
sklearn.ensemble.RandomForestClassifier	Random Forest Classifier
imblearn.ensemble.BalancedBaggingClassifer	Balanced Bagging Classifier
xgboost.XGBClassifier	Extreme Gradient Boosting classifier
sklearn.metrics	Performance evaluation metrics like accuracy score, recall, precision, ROC curve, etc.

Table D.1: Libraries used for the project.

Bibliography

[1] Alan Agresti and David B Hitchcock. Bayesian inference for categorical data analysis. *Statistical Methods and Applications*, 14(3):297–330, 2005.

[2] Cletus Agyenim-Boateng, Anne Stafford, and Pamela Stapleton. The role of structure in manipulating ppp accountability. *Accounting, Auditing & Accountability Journal*, 2017.

[3] Hafiz A Alaka, Lukumon O Oyedele, Hakeem A Owolabi, Vikas Kumar, Saheed O Ajayi, Olugbenga O Akinade, and Muhammad Bilal. Systematic review of bankruptcy prediction models: Towards a framework for tool selection. *Expert Systems with Applications*, 94:164–184, 2018.

[4] Esteban Alfaro, Noelia García, Matías Gámez, and David Elizondo. Bankruptcy forecasting: An empirical comparison of adaboost and neural networks. *Decision Support Systems*, 45(1):110–122, 2008.

[5] Edward I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609, 1968.

[6] Gareth Ambler, Rumana Z Omar, and Patrick Royston. A comparison of imputation techniques for handling missing predictor values in a risk model with a binary outcome. *Statistical methods in medical research*, 16(3):277–298, 2007.

[7] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

[8] Jinwoo Baek and Sungzoon Cho. Bankruptcy prediction for credit risk using an auto-associative neural network in korean firms. In *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.*, pages 25–29. IEEE, 2003.

[9] Flavio Barboza, Herbert Kimura, and Edward Altman. Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, 83:405–417, 2017.

[10] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999.

[11] W Beaver. Financial ratios predictors of financial distress. *Journal of Accounting Research*, pages 70–112, 1967.

[12] William H Beaver, Maureen F McNichols, and Jung-Wu Rhie. Have financial statements become less informative? evidence from the ability of financial ratios to predict bankruptcy. *Review of Accounting studies*, 10(1):93–122, 2005.

[13] Joy Begley, Jin Ming, and Susan Watts. Bankruptcy classification errors in the 1980s: An empirical analysis of altman's and ohlson's models. *Review of accounting Studies*, 1(4):267–284, 1996.

[14] Jerzy Błaszczyński and Jerzy Stefanowski. Actively balanced bagging for imbalanced data. In *International Symposium on Methodologies for Intelligent Systems*, pages 271–281. Springer, 2017.

[15] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee, 2007.

[16] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[17] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[18] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[19] Iain Brown and Christophe Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3):3446–3453, 2012.

[20] Raffaella Calabrese, Marta Degl'Innocenti, and Silvia Angela Osmetti. The effectiveness of tarp-cpp on the us banking industry: A new copula-based approach. *European Journal of Operational Research*, 256(3):1029–1037, 2017.

[21] Hong Cao, Xiao-Li Li, David Yew-Kwong Woon, and See-Kiong Ng. Integrated oversampling for imbalanced time series classification. *IEEE Transactions on Knowledge and Data Engineering*, 25(12):2809–2822, 2013.

[22] Yu Cao. Aggregating multiple classification results using choquet integral for financial distress early warning. *Expert Systems with Applications*, 39(2):1830–1836, 2012.

[23] Devulapalli Karthik Chandra, Vadlamani Ravi, and Pediredla Ravisankar. Support vector machine and wavelet neural network hybrid: application to bankruptcy prediction in banks. *International Journal of Data Mining, Modelling and Management*, 2(1):1–21, 2010.

[24] Chris Charalambous, Andreas Charitou, and Froso Kaourou. Application of feature extractive algorithm to bankruptcy prediction. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 5, pages 303–308. IEEE, 2000.

[25] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[26] Ning Chen, Bernardete Ribeiro, Armando S Vieira, João Duarte, and João C Neves. A genetic algorithm-based approach to cost-sensitive bankruptcy prediction. *Expert Systems with Applications*, 38(10):12939–12945, 2011.

[27] Ning Chen and Armando Vieira. Bankruptcy prediction based on independent component analysis. In *ICAART*, pages 150–155, 2009.

[28] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[29] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4, 2015.

[30] Zhensong Chen, Wei Chen, and Yong Shi. Ensemble learning with label proportions for bankruptcy prediction. *Expert Systems with Applications*, 146:113155, 2020.

[31] Esteban Alfaro Cortes, Matias Gamez Martinez, and Noelia García Rubio. Multiclass corporate failure prediction by adaboost. m1. *International Advances in Economic Research*, 13(3):301–312, 2007.

[32] Edward B Deakin. A discriminant analysis of predictors of business failure. *Journal of accounting research*, pages 167–179, 1972.

[33] Despina Deligianni and Sotiris Kotsiantis. Forecasting corporate bankruptcy with an ensemble of classifiers. In *Hellenic Conference on Artificial Intelligence*, pages 65–72. Springer, 2012.

[34] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

[35] A Adam Ding, Shaonan Tian, Yan Yu, and Hui Guo. A class of discrete transformation survival models with application to default probability prediction. *Journal of the American Statistical Association*, 107(499):990–1003, 2012.

[36] Michael Doumpos and Constantin Zopounidis. Model combination for credit risk assessment: A stacked generalization approach. *Annals of Operations Research*, 151(1):289–306, 2007.

[37] Michalis Doumpos, Kostas Andriosopoulos, Emilios Galariotis, Georgia Makridou, and Constantin Zopounidis. Corporate failure prediction in the european energy sector: A multicriteria approach and the effect of country characteristics. *European Journal of Operational Research*, 262(1):347–360, 2017.

[38] Philippe Du Jardin. Bankruptcy prediction using terminal failure processes. *European Journal of Operational Research*, 242(1):286–303, 2015.

[39] Philippe du Jardin. A two-stage classification technique for bankruptcy prediction. *European Journal of Operational Research*, 254(1):236–252, 2016.

[40] Travis Dyer, Mark Lang, and Lorien Stice-Lawrence. The evolution of 10-k textual disclosure: Evidence from latent dirichlet allocation. *Journal of Accounting and Economics*, 64(2-3):221–245, 2017.

[41] James D Dziura, Lori A Post, Qing Zhao, Zhixuan Fu, and Peter Peduzzi. Strategies for dealing with missing data in clinical trials: from design to analysis. *The Yale journal of biology and medicine*, 86(3):343, 2013.

[42] Robert O Edmister. An empirical test of financial ratio analysis for small business failure prediction. *Journal of Financial and Quantitative analysis*, 7(2):1477–1493, 1972.

[43] Silvia Figini, Roberto Savona, and Marika Vezzoli. Corporate default prediction model averaging: a normative linear pooling approach. *Intelligent Systems in Accounting, Finance and Management*, 23(1-2):6–20, 2016.

[44] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[45] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[46] Ruibin Geng, Indranil Bose, and Xi Chen. Prediction of financial distress: An empirical study of listed chinese companies using data mining. *European Journal of Operational Research*, 241(1):236–247, 2015.

[47] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[48] Wolfgang Härdle, Yuh-Jye Lee, Dorothea Schäfer, and Yi-Ren Yeh. Variable selection and oversampling in the use of smooth support vector machines for predicting the default risk of companies. *Journal of Forecasting*, 28(6):512–534, 2009.

[49] Tadaaki Hosaka. Bankruptcy prediction using imaged financial ratios and convolutional neural networks. *Expert systems with Applications*, 117:287–299, 2019.

[50] Kuan-Chieh Huang, Yau-Hwang Kuo, and I-Cheng Yeh. A novel fitness function in genetic algorithms to optimize neural networks for imbalanced data sets. In *2008 Eighth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 647–650. IEEE, 2008.

[51] Qing-Hua Huang, Jie Sun, and Wei-Dong Mao. Dynamic modeling on credit risk evaluation with fixed time window and imbalanced ensemble of support vector machine. *Recent Patents on Computer Science*, 5(1):51–58, 2012.

[52] Christos Ioannidis, Fotios Pasiouras, and Constantin Zopounidis. Assessing bank soundness with classification techniques. *Omega*, 38(5):345–357, 2010.

[53] Xin Jin, Anbang Xu, Rongfang Bie, and Ping Guo. Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. In *International Workshop on Data Mining for Biomedical Applications*, pages 106–115. Springer, 2006.

[54] Stewart Jones and David Hensher. Advances in credit risk modelling and corporate bankruptcy prediction. Technical report, Cambridge University Press, 2008.

[55] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64(5):402, 2013.

[56] Myoung-Jong Kim and Dae-Ki Kang. Ensemble with neural networks for bankruptcy prediction. *Expert systems with applications*, 37(4):3373–3379, 2010.

[57] Kaitlin Kirasich, Trace Smith, and Bivin Sadler. Random forest vs logistic regression: Binary classification for heterogeneous datasets. *SMU Data Science Review*, 1(3):9, 2018.

[58] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[59] Max Kuhn and Kjell Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[60] P Ravi Kumar and Vadlamani Ravi. Bankruptcy prediction in banks and firms via statistical and intelligent techniques–a review. *European journal of operational research*, 180(1):1–28, 2007.

[61] Tuong Le, Hoang Le Son, Minh Thanh Vo, Mi Young Lee, Sung Wook Baik, et al. A cluster-based boosting algorithm for bankruptcy prediction in a highly imbalanced dataset. *Symmetry*, 10(7):250, 2018.

[62] Tsun-Siou Lee and Yin-Hua Yeh. Corporate governance and financial distress: Evidence from taiwan. *Corporate governance: An international review*, 12(3):378–388, 2004.

[63] Terje Lensberg, Aasmund Eilifsen, and Thomas E McKee. Bankruptcy theory development and classification via genetic programming. *European Journal of operational research*, 169(2):677–697, 2006.

[64] Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.

[65] Hui Li and Jie Sun. Gaussian case-based reasoning for business failure prediction with empirical data in china. *Information Sciences*, 179(1-2):89–108, 2009.

[66] Xinyi Li, Yinchuan Li, Hongyang Yang, Liuqing Yang, and Xiao-Yang Liu. Dp-lstm: Differential privacy-inspired lstm for stock prediction using financial news. *arXiv preprint arXiv:1912.10806*, 2019.

[67] Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research*, 252(2):561–572, 2016.

[68] Shih-Wei Lin, Yeou-Ren Shiue, Shih-Chi Chen, and Hui-Miao Cheng. Applying enhanced data mining approaches in predicting bank performance: A case of taiwanese commercial banks. *Expert Systems with Applications*, 36(9):11543–11551, 2009.

[69] Roderick J Little, Ralph D'Agostino, Michael L Cohen, Kay Dickersin, Scott S Emerson, John T Farrar, Constantine Frangakis, Joseph W Hogan, Geert Molenberghs, Susan A Murphy, et al. The prevention and treatment of missing data in clinical trials. *New England Journal of Medicine*, 367(14):1355–1360, 2012.

[70] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[71] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. Exploring graph neural networks for stock market predictions with rolling window analysis. *arXiv preprint arXiv:1909.10660*, 2019.

[72] David O Mbat and Eyo I Eyo. Corporate failure: causes and remedies. *Business and Management Research*, 2(4):19–24, 2013.

[73] Jae-H Min, Chul-Woo Jeong, and Myung-Suk Kim. Tuning the architecture of support vector machine: the case of bankruptcy prediction. *Management Science and Financial Engineering*, 17(1):19–43, 2011.

[74] Tanya Molodtsova and David H Papell. Out-of-sample exchange rate predictability with taylor rule fundamentals. *Journal of international economics*, 77(2):167–180, 2009.

[75] Carol M Musil, Camille B Warner, Piyanee Klainin Yobas, and Susan L Jones. A comparison of imputation techniques for handling missing data. *Western Journal of Nursing Research*, 24(7):815–829, 2002.

[76] Masao Nakamura. Japanese corporate governance practices in the post-bubble era: Implications of institutional and legal reforms in the 1990s and early 2000s. *International Journal of Disclosure and Governance*, 3(3):233–261, 2006.

[77] James A Ohlson. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of accounting research*, pages 109–131, 1980.

[78] David L Olson, Dursun Delen, and Yanyan Meng. Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, 52(2):464–473, 2012.

[79] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

[80] Larry Olanrewaju Orimoloye, Ming-Chien Sung, Tiejun Ma, and Johnnie EV Johnson. Comparing the effectiveness of deep feedforward neural networks and

shallow architectures for predicting stock price indices. *Expert Systems with Applications*, 139:112828, 2020.

[81] Giuseppe Paleologo, André Elisseeff, and Gianluca Antonini. Subagging for credit scoring models. *European Journal of Operational Research*, 201(2):490–499, 2010.

[82] Michael P Perrone. Putting it all together: Methods for combining neural networks. In *Advances in neural information processing systems*, pages 1188–1189, 1994.

[83] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[84] J Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[85] Fernando Sánchez-Lasheras, Javier de Andrés, Pedro Lorca, and Francisco Javier de Cos Juez. A hybrid device for the solution of sampling bias problems in the forecasting of firms' bankruptcy. *Expert Systems with Applications*, 39(8):7512–7523, 2012.

[86] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[87] Jeffrey C Schlimmer and Richard H Granger. Incremental learning from noisy data. *Machine learning*, 1(3):317–354, 1986.

[88] Carlos Serrano-Cinca and BegoñA GutiéRrez-Nieto. Partial least square discriminant analysis for bankruptcy prediction. *Decision support systems*, 54(3):1245–1255, 2013.

[89] Martin Sewell. Ensemble learning. *RN*, 11(02), 2008.

[90] Gregory EP Shailer. *An introduction to corporate governance in Australia*. Pearson Education Australia, 2004.

[91] Yin Shi and Xiaoni Li. An overview of bankruptcy prediction models for corporate firms: A systematic literature review. *Intangible Capital*, 15(2):114–127, 2019.

[92] Kyung-Shik Shin and Yong-Joo Lee. A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications*, 23(3):321–328, 2002.

[93] Cindy Yoshiko Shirata. Financial ratios as predictors of bankruptcy in japan: an empirical research. In *Proceedings of the second Asian Pacific interdisciplinary research in accounting conference*, pages 437–445. Citeseer, 1998.

[94] Fiona M Shrive, Heather Stuart, Hude Quan, and William A Ghali. Dealing with missing data in a multi-question depression scale: a comparison of imputation methods. *BMC medical research methodology*, 6(1):57, 2006.

[95] Tyler Shumway. Forecasting bankruptcy more accurately: A simple hazard model. *The journal of business*, 74(1):101–124, 2001.

[96] Joseph F Sinkey Jr. A multivariate statistical analysis of the characteristics of problem banks. *The Journal of Finance*, 30(1):21–36, 1975.

[97] Jonathan AC Sterne, Ian R White, John B Carlin, Michael Spratt, Patrick Royston, Michael G Kenward, Angela M Wood, and James R Carpenter. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338:b2393, 2009.

[98] James H Stock and Mark W Watson. Why has us inflation become harder to forecast? *Journal of Money, Credit and banking*, 39:3–33, 2007.

[99] Toshiyuki Sueyoshi and Mika Goto. Dea–da for bankruptcy-based performance assessment: Misclassification analysis of japanese construction industry. *European Journal of Operational Research*, 199(2):576–594, 2009.

[100] Jie Sun and Hui Li. Financial distress prediction using support vector machines: Ensemble vs. individual. *Applied Soft Computing*, 12(8):2254–2265, 2012.

[101] Richard J Taffler. The assessment of company solvency and performance using a statistical model. *Accounting and Business Research*, 13(52):295–308, 1983.

[102] Katsuyuki Tanaka, Takuo Higashide, Takuji Kinkyo, and Shigeyuki Hamori. Forecasting the vulnerability of industrial economic activities: Predicting the bankruptcy of companies. *Journal of Management Information and Decision Sciences*, 2017.

[103] Shaonan Tian and Yan Yu. Financial ratios and bankruptcy predictions: An international evidence. *International Review of Economics & Finance*, 51:510–526, 2017.

[104] Shaonan Tian, Yan Yu, and Hui Guo. Variable selection and corporate bankruptcy forecasts. *Journal of Banking & Finance*, 52:89–100, 2015.

[105] Mario Hernandez Tinoco and Nick Wilson. Financial distress and bankruptcy prediction among listed companies using accounting, market and macroeconomic variables. *International Review of Financial Analysis*, 30:394–419, 2013.

[106] Antanas Verikas, Zivile Kalsyte, Marija Bacauskiene, and Adas Gelzinis. Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey. *Soft Computing*, 14(9):995–1010, 2010.

[107] Mauno Vihinen. How to evaluate performance of prediction methods? measures and their interpretation in variation effect analysis. In *BMC genomics*, volume 13, page S2. BioMed Central, 2012.

[108] Gang Wang, Jinxing Hao, Jian Ma, and Hongbing Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*, 38(1):223–230, 2011.

[109] Haiying Wang and Huiru Zheng. *Model Validation, Machine Learning*, pages 1406–1407. Springer New York, New York, NY, 2013.

[110] Yan Wang and Xuelei Sherry Ni. A xgboost risk model via feature selection and bayesian hyper-parameter optimization. *arXiv preprint arXiv:1901.08433*, 2019.

[111] Peter Wanke, Carlos P Barros, and João R Faria. Financial distress drivers in brazilian banks: A dynamic slacks approach. *European Journal of Operational Research*, 240(1):258–268, 2015.

[112] David West, Scott Dellana, and Jingxia Qian. Neural network ensemble strategies for financial decision applications. *Computers & operations research*, 32(10):2543–2559, 2005.

[113] Robert Craig West. A factor-analytic approach to bank condition. *Journal of Banking & Finance*, 9(2):253–266, 1985.

[114] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[115] Jesper N Wulff and Linda Ejlskov. Multiple imputation by chained equations in praxis: Guidelines and review. *Electronic Journal of Business Research Methods*, 15(1), 2017.

[116] Ming Xu and Chu Zhang. Bankruptcy prediction: the case of japanese listed companies. *Review of accounting studies*, 14(4):534–558, 2009.

[117] Ligang Zhou, Kin Keung Lai, and Jerome Yen. Bankruptcy prediction using svm models with a new approach to combine features selection and parameter optimisation. *International Journal of Systems Science*, 45(3):241–253, 2014.

[118] Maciej Zieba, Sebastian K Tomczak, and Jakub M Tomczak. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, 58:93–101, 2016.

[119] Constantin Zopounidis and Michael Doumpos. Multi-group discrimination using multi-criteria analysis: Illustrations from the field of finance. *European Journal of Operational Research*, 139(2):371–389, 2002.