Quantitative Finance - Master Thesis

The effect of sampling on bankruptcy prediction

Student:
Matthijs Poot (497721)
In collaboration with:
Accenture: Data & AI

Committee:
Dr. Onno Kleen
Dr. Andreas Pick
External:
Marco de Vree

Abstract

Bankruptcy data sets are notoriously imbalanced, such that the minority class of defaults is less than 5%. To train statistical methods and Machine Learning models, one has to sample the training set to create a equal class size. This thesis analyses the effect of six sampling methods on multivariate discriminant analysis, logistic regression, artificial neural network, support vector machine, boosting, XGBoost, and Random Forest. The sampling techniques include the industry standard under-sampling. Contrary to previous research, I compare random over-sampling and four techniques based on Synthetic Minority Oversampling Technique (SMOTE) to the performance of under-sampling. The thesis adds a new extension, which I call Borderline-3. This algorithm creates synthetic points only for observations that have more nearest neighbours for minority class than majority class observations. The first result shows that under-sampling is the most consistent technique across the models. In contrast to previous literature, logistic regression can outperform most sampling and model combinations with a SMOTE-based extension. Furthermore, I test feature importance using Shapley Additive Explanations. The results show that feature importance is significantly affected. Thus the sampling methods one chooses can impact the decisions of company executives or loan providers. This paper concludes that sampling has a significant influence on the predictive performance in bankruptcy prediction despite no attention in the current literature.

June 20, 2022

No amount of (apparent) statistical
evidence will make a statement
invulnerable to common sense.

Robert Merton Solow

Acknowledgements

This thesis is the final graduation requirement for obtaining the Master's degree in Quantitative Finance at the Erasmus University of Rotterdam. This research would not have been possible without the guidance of my supervisor Dr. Onno Kleen. Onno's feedback gave invaluable help towards analysing and structuring this to a coherent narrative. I also want thank Dr. Andreas Pick as part of the graduation committee. I would like to thank Accenture for the opportunity to write and learn so much. In particular the *Agile Release Train* group and the *Talent Factory* for allowing me to present and receive helpful feedback. Finally, I am profoundly grateful for the love and support of my family; My father Jan, my mother Céline, my brother Lucas and my sister Fleur. Finally I thank my roommates and friends for making the last few months so fun and memorable.

Matthijs Poot
June 20, 2022, Rotterdam

List of Algorithms

List of Figures

List of Tables

Contents

1. Introduction

Bankruptcy prediction is a critical classification challenge for banks and other loan providers. Economically, if a bank provides a loan to a company but it does not default than this is a missed capital gain opportunity. If however, a bank loans money to a company that defaults, it can potentially lose the whole sum of the loan and future interest earnings. The costs of the latter are arguably greater than missed capital gains because defaulting loans can do substantial damage to the balance sheet of the loan provider. Statistically, the missclassifcation of 'active' and soon to be 'default' companies are false positives and false negatives. Since the seminal paper of Altman (1968), academics have tried to minimise the number of false positives (type I error) and false positives (type II error). Bankruptcy literature includes statistical techniques, such as discriminant analysis from Altman (1968) and later logistic regression (LR) from Ohlson (1980). In the last few decades Machine Learning models, including support vector machines (SVM), boosting and bagging, have shown superior performance to the classical methods (Wilson and Sharda, 1994; Sun et al., 2014). However before one trains a model, the data needs to be balanced. This means that in the dataset the binary target variable, denoted by 'active' and 'default', has an equal number of observations for each target class. Bankruptcy data is actually imbalanced where the minority class is less than 5% of the total number of observations (Veganzones and Séverin, 2018). The current research only considers under-sampling, such that one randomly deletes observations of the majority class to create an equal number of observations. Other techniques to balance a data set exist such as random over-sampling, which duplicates observations of the minority class but it has the tendency to over-fit. In the last twenty years an over-sampling technique has shown significant improved of predictive performance and is widely applied in the academic fields of medical diagnosis and fraud detection (Fernandez et al., 2018). The techniques is known as Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002). In short, SMOTE creates synthetic observations between minority class observations. This thesis is the first comparative analysis on the performance of SMOTE to under-sampling for the classical statistical techniques and Machine Learning models. Therefore I ask:

Does sampling significantly affect bankruptcy prediction?

The motivation for this question is two-fold. Firstly, it extends the work done by Barboza et al. (2017). The authors compared the the industry standard statistical techniques, discriminant analysis by Altman (1968) and LR by Ohlson (1980), to more recent models. I include the following machine learning models: Artificial Neural Networks (ANN), SVM, Boosting, Extreme Gradient Boosting (XGBoost), and Random Forest (RF). I compare the seven models for an unbalanced data set and six sampling techniques. These include: random under-sampling, random-oversampling, SMOTE, Adative Synthetic (ADASYN), Borderline-1, and Borderline-3. The latter three are SMOTE-based extensions. I choose ADASYN and Borderline-1 because these are widely the most applied extensions for a binary target variable in other academic fields (Fernandez et al., 2018). These extensions have additional rules to over-sample the dataset to create 'smarter' over-sampling. In other words, the algorithm creates more synthetic observations for one observation than another.

For example, for Borderline-1 only creates synthetic observations for points that have more 'active' companies than 'defaulting' companies near it (based on K Nearest Neighbours). These observations are referred to as 'danger' points. If all neighbours are only 'active' then it generates no additional observations. I programmed a new extension, referred to as Borderline-3, to only over-sample observations that have more 'defaulting' neighbours than 'active' neighbours. These observation are referred to as 'safe' points. The motivation is that instead of oversampling on extreme observations, like in Borderline-1, the algorithm focuses on observations that are more equal. Such that the observations that are similar are more informative than the extreme points (i.e. 'danger'). This reduces the risk of generating false positives in a test environment. Based on the methodology of Barboza et al. (2017) with different sampling techniques, I test the following hypothesis:

Hypothesis I: If sampling is applied to one-year ahead bankruptcy dataset, then SMOTE-based sampling outperforms undersampling.

Secondly, I test the effect of sampling on the predictive performance of the different sampling methods. These sampling methods include under-sampling, SMOTE, and Borderline-1. In the case of under-sampling, potentially relevant information is deleted. Whereas SMOTE generates new observations such that the data set is twice the size due to synthetic data. What sampling technique a bank chooses can potentially affect what financial data it bases it decisions on. To analyse the effect of sampling on feature performance, I use Shapley Additive Explanations (SHAP) (Lundberg and Lee, 2017). This a a game theory centric approach that quantifies the contribution of each individual variable. It shows how important a feature is to the trained model. I use SHAP for the machine learning model XGBoost to quantify feature importance. It is beyond the scope of the thesis to test all machine learning models. This test is used to test the second hypothesis:

Hypothesis II: Sampling methods affect feature importance for bankruptcy data.

To answer the main research question, I use the Wharton Research Data Services, access provided by the Erasmus University of Rotterdam, to download publicly listed companies in the United States over the period 1980Q1 until 2022Q1. From this I construct a dataset with a target variable for a company to default within one year. Here '0' is for a company that is active next year and '1' for a company that defaults in the next year. For variable construction I follow Barboza et al. (2017) to create eleven variables.

The first hypothesis expects SMOTE-based sampling to outperform under-sampling. The results show that under-sampling is the most consistent sampling technique for all models. The performance is not always the highest, but it consistently beats SMOTE and ADASYN for all models except linear SVM. Under-sampling the best sampling technique for Boosting, XGBoost, and RF and support existing literature that it outperforms Altman's and Ohlson's techniques (Barboza et al., 2017; Heo and Yang, 2014; Wang et al., 2014). However, Borderline-1 is the superior sampling technique in combination with LR and linear SVM. The results show that LR (with Borderline-1) outperforms under-sampling for boosting and RF. This contradicts the view of Barboza et al. (2017) that machine learning models outperform classical statistical methods. Therefore the statistical method by Ohlson (1980) is still relevant, and simply using machine learning models does not guarantee the highest predictive performance.

The new SMOTE-based extension Borderline-3 slightly outperforms an unbalanced data set, but is arguably useless. In combination with the performance of Borderline-3, the 'danger' points hold more predictive power than the 'safe' points. The Borderline-3 sampling technique omits too much relevant data. The algorithm did minimise the number of false positives but at the expense of over-fitting on the majority class.

The second hypothesis expects sampling to significantly affect feature importance, and indeed the results show that for XGBoost feature importance can significantly change. For under-sampling 'Growth of Sales' has the second highest feature importance but its fifth for SMOTE and tenth for Borderline-1. This implies that if one bank that uses under-sampling for it predictive model and another bank uses Borderline-1, then the risk of providing a loan is equal but the decision is dependent on the sampling decision of the engineer. The statistical evidence of hypothesis one and two show that under-sampling is the most universal technique, but still requires bankers to use common sense for approving loans.

The contributions of this thesis are three-fold:

- Develop a novel extension to SMOTE, based on the algorithm of Borderline-1 that minimises false positives, called Borderline-3. The technique helps to minimise false positives but fails to minimise false negatives.
- Apply six sampling techniques on for seven models on bankruptcy data. Directly contrasting under-sampling with (SMOTE-based) over-sampling techniques in bankruptcy prediction. The results show that the sampling technique one uses affect the outcome and certainty of the chosen model.
- Investigate feature importance and characteristics using SHAP. The tool helps banks and other loan providers to visualise why a Machine Learning model predicts a default or not.

The thesis is structured as follows: Chapter 2 presents the current state of the literature on bankruptcy prediction; in Chapter 3, I present the raw data set, adjustments, and variable construction; then, sampling techniques, statistical methods and machine learning models, feature analysis, and performance analysis are detailed in Chapter 4; Chapter 5 discusses the results; finally, Chapter 6 concludes the thesis with a discussion on limitations, implications, and future research.

2. Literature Review

In the United States, the Securities and Exchange Commission (SEC) defines public corporate bankruptcy as a company that goes out of business from a crippling amount of debt (SEC, 2009). There exist two types of bankruptcy for publicly listed companies. Firstly, a company declares **Chapter 7** if a company is liquidated. Assets are divided over three types of investors (in order of the first claim): Secured creditors, unsecured creditors, and stockholders. The secured creditors, i.e. banks, have collateral and therefore are the first to get paid. The unsecured creditors including banks, suppliers, and bondholders, are next in line to claim the outstanding debt. Finally, the stockholders receive the remaining assets. It is likely at this point a stockholder gets no value whatsoever. Secondly, a company uses **Chapter 11** if management hopes to 'reorganise' the business to one day be profitable again. But this is not certain and still poses a significant risk for credit suppliers. Because the bankruptcy court must approve all significant business decisions. Although the company is labelled bankrupt, it still trades on the stock exchange and must oblige to the SEC filings. If the company is unable to restructure it is declared insolvent and liquidated in the same way as chapter 7. Balcaen and Ooghe (2006) argue the term bankruptcy is a poorly defined dichotomy. In the case of chapter 11, it can be a strategic decision by the board. This human element is however not reflected in the accounting data typically used in the financial distress literature. In this thesis, I refer to bankruptcy and default as both chapter 7 and chapter 11.

The academic literature of bankruptcy prediction was pioneered in the 1960s with Beaver (1966) who conducted a univariate discriminant analysis on thirty financial ratios. The main contribution is that accounting data "can be evaluated in terms of their utility and that utility can be defined in terms of predictive ability" (Beaver, 1966). This was closely followed by the seminal paper of Altman (1968), who used multiple discriminant analysis (MDA) to assess the financial distress of manufacturing companies. His model estimated a so-called Z-score classifying companies into three groups: safe-zone, grey-zone, and distress-zone. The lower the score, the more likely a company was to experience financial trouble in the next one to five years. Later, Altman et al. (1977) improved the original Z model with new variables and showed superior performance. Today, Altman's Z-score is still used in comparative studies as a baseline 'to-beat' (Balcaen and Ooghe, 2006). The Z-score is widely used in the industry and forms the foundation of analysis for a significant body of literature. These variables are based on: liquidity, profitability, productivity, leverage and asset turn-over. However, one significant disadvantage of MDA analysis is it assumes a linear relationship between a variable and the predictive probability (Balcaen and Ooghe, 2006). This led Ohlson (1980) to use conditional probability models, such that variables and failure probability are non linearly distributed. In a comparative study, Begley et al. (1996) found Ohlson's model is superior to Altman's.

As Shin and Lee (2002) point out, these conventional statistical techniques assume normality and independence for the input variables. This is not the case in financial accounting data. Modern techniques, however, do not rely on the same assumptions as classical methods. A long list of literature show the predictive superiority of machine learning models. Firstly, using the five variables of

Altman's Z-score from the 1968 model, Wilson and Sharda (1994) showed a significantly increased performance using Artificial Neural Networks over multivariate discriminant analysis. Other notable machine learning papers are shown in Table 1. This literature has shown that for accounting ratios and data sets (both country and period) show an increase in the predictive performance due to machine learning algorithms. Most notably, ensemble methods show the highest performance (Barboza et al., 2017; Heo and Yang, 2014; Wang et al., 2014).

Table 1: Notable financial distress and bankruptcy prediction papers.

Author(s)	**Sampling**	**Models**
Altman (1968)	Unbalanced and Under-sampling	Multivariate Discriminant Analysis
Altman et al. (1977)	Unbalanced and Over-sampling	Multivariate Discriminant Analysis
Ohlson (1980)	Under-sampling	Logistic Regression
Wilson and Sharda (1994)	Undersampling	Multivariate Discriminant Analysis, and Artificial Neural Networks
Sun et al. (2014)	Under-sampling	Logistic Regression, Artificial Neural Networks, Support Vector Machines, Decision Tree, Random Forest
Ligang et al. (2014)	Under-sampling	Multivariate Discriminant Analysis, Logistic Regression, Decision Trees, and Support Vector Machines.
Heo and Yang (2014)	Under-sampling	Discriminant Analysis, Decision Trees, Boosting, Artificial Neural Networks, and Support Vector Machines
Wang et al. (2014)	Under-sampling	Boosting, Bagging, Logistic Regression, Naive Bayes, Decision Trees, Artificial Neural Networks, Support Vector Machines
Kim et al. (2015)	SMOTE	Boosting
Barboza et al. (2017)	Under-sampling	Multivariate Discriminant Analysis, Logistic Regression, Artificial Neural Network, Support Vector Machines, Boosting, Bagging, Random Forest
Le et al. (2018)	Unbalanced, SMOTE and extensions	Artificial Neural Networks, Support Vector Machines, Decision Tree, Random Forest

Bankruptcy prediction is a binary classification problem, where there is severe class imbalance. The class imbalance problem is common in many real-world data sets such as cancer diagnosis or fraud detection (Fernandez et al., 2018). An imbalanced data set results in poorly trained classifiers such that those that maximise accuracy but neglect false positives (Type I error) and false negatives (Type II error). To address imbalanced training data set the current literature, shown in Table 1, overwhelmingly uses under-sampling. The main disadvantage is the loss of data. Nevertheless, under-sampling has been shown to build significantly better classifiers than over-sampling (Chawla et al., 2002). More recent papers consider over-sampling techniques such as Synthetic Minority Over-sampling TEchnique (SMOTE). Instead of replicating observations, SMOTE creates synthetic points on a vector between a set number of nearest neighbours. More recent papers, such as Kim et al. (2015) and Le et al. (2018), have used SMOTE in a bankruptcy setting, but they have not compared the performance for equal data set to the industry standard under-sampling.

3. Data

This chapter discusses the data set and variable construction for the tests discussed in the introduction and the methodology. The data sets is available from the Wharton Research Data Services [1], WRDS hereafter, and contain US-listed companies from the following data providers: *CRSP Delist* and *CRSP / Compustat Merged*. The following section describes the data set and the decisions to prepare it for the tests.

3.1. CRSP Delist

The *CRSP / Compustat Merged* data set contains company info and financial ratios, but does not have any status information. Therefore I need a crosswalk file that links a company to its current status. I use *CRSP Delist* that contains four variables:

- The PERMNO label; A company's unique code and remains unchanged over time. This code is the identifier to merge the *CRSP Delist* and *CRSP / Compustat Merged* data sets.
- Company Name; Over time, a company's name can change, which I verified by comparing it to the PERMNO label.
- The Status; A company can take any of the following values: Active (100-199), Mergers (200-299), Exchanges (300-399), Liquidations (400-499), Dropped (500-599), Expirations (600-699), Domestics that became Foreign (900-999). A full overview of all codes can be found on the website [2]. In this research, I use Liquidations (400-499), and bankruptcy declared due to insolvency (574) as companies that default. Active companies are set to 100, because other codes between 100 and 199 are active but stopped trading. Active and default companies are set to binary values, 0 and 1, respectively.
- The Delisting Date; When a company is insolvent or liquidated. If a company remains active, this variable is equal to the last date of the database, here 2021-12-31.

Let this data set be called `df_delist`. The data set has 10165 rows × 4 columns for 10,165 individual companies. Here 877 companies have value defaulted. Figure 1 shows a histogram of the number of bankruptcies in the data set for each year between 1980-2021.

3.2. CRSP / Compustat Merged

CRSP, or Center for Research in Security Prices, provides market data for publicly listed companies in North America. Compustat provides additional financial statement data. The *CRSP / Compustat Merged* data set is a file that links fundamental data of Compustat to the PERMNO label of CRSP. I require this for the crosswalk file `df_delist` discussed in the previous section.

[1] https://wrds-www.wharton.upenn.edu

[2] View website using the link https://www.crsp.org/products/documentation/delisting-codes

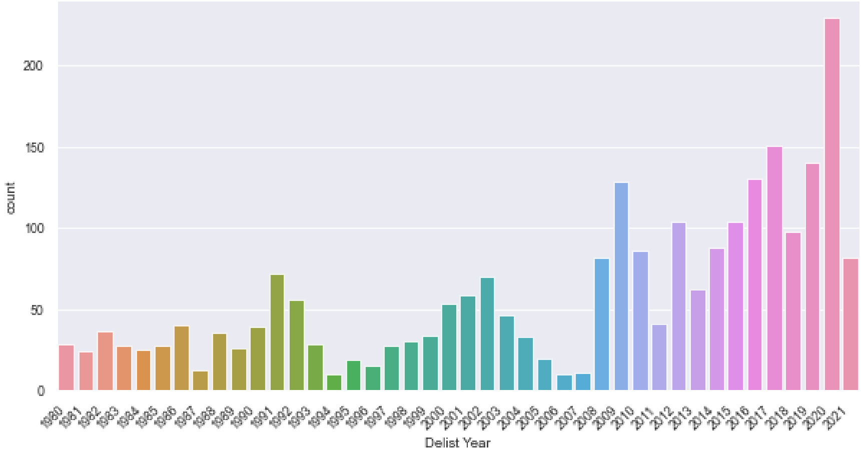


Figure 1: The number of defaults per year in CRSP Delist data set.

Next, I construct eleven variables following Barboza et al. (2017), BKA-17 hereafter. That paper is a comparative study of multiple statistical methods and machine learning models. BKA-17 use five variables of Altman (1968) and six variables of Carton and Hofer (2006). I considered using the Altman et al. (1977) ratios over Altman (1968). However, the supposed superior variables rely on normalisation, which is uncommon in later financial distress literature. Furthermore, due to the popularity of the Altman-1968 variables in later papers, I chose those. The latter variables, by Carton and Hofer (2006), require me to construct some variables based on two years. For this reason, I only keep companies that have two or more observations in the data set. Table 2 shows the mathematical construction of these variables.

Table 2: Variables from Barboza et al. (2017) constructed from the data set.

Variable	Description	Formula
X1	Liquidity	$\frac{\text{Net Working Capital}}{\text{Total Assets}}$
X2	Profitability	$\frac{\text{Retained Earning}}{\text{Total Assets}}$
X3	Productivity	$\frac{\text{Earnings before interest and taxes}}{\text{Total Assets}}$
X4	Leverage	$\frac{\text{Market Value of share * number of shares}}{\text{Total Debt}}$
X5	Asset turnover	$\frac{\text{Sales}}{\text{Total Assets}}$
OM	Operational Margin	$\frac{\text{Earnings before interest and taxes}}{\text{Sales}}$
GA	Growth of Assets	$\frac{\text{Total Assets}_t - \text{Total Assets}_{t-1}}{\text{Total Assets}_t}$
GS	Growth in Sales	$\frac{\text{Sales}_t - \text{Sales}_{t-1}}{\text{Sales}_t}$
GE	Growth in number of Employees	$\frac{\text{Employees}_t - \text{Employees}_{t-1}}{\text{Employees}_t}$
CROE	Change in Return on Equity	$\text{ROE}_t - \text{ROE}_{t-1}$, where $\text{ROE} = \frac{\text{Net Income}}{\text{Common Stockholder's equity}}$
CPB	Change in Price-to-Book ratio	$\text{PB}_t - \text{PB}_{t-1}$, where $\text{PB} = \frac{\text{Market Value per share}}{\text{Book Value per share}}$

Notes: Table 11 in Appendix B on page 36 shows the identifier information for the Compustat variables to construct the variables of table 2. I follow the variable names and construction of BKA-17.

The pre-processed dataset, referred to hereafter as `df_pre`, is constructed using the 'CRSP Delist' and 'CRSP Compustat Merged'. The two data sets are merged on the identifier 'PERMNO'. I am interested in using the data of each company only once. To get a randomised selection, I use a function to get the final dataset `df_sim1` and consists of the following steps:

1. Create a one-year time lag for each ratio for each company.
2. Drop all missing values. This step is second because the shift function creates new missing values.
3. Replace all values equal to positive or negative infinite with the next largest value.
4. Split the data set in 'active' and 'defaulted' companies.
5. For the defaulted companies, keep only the observation of one-year-prior to delisting date.
6. For the active companies, randomly select an observation of a company based on PERMNO.
7. Concat the data frames that result from the previous two steps.

Some comments on these steps. First, to accurately predict one-year ahead, I need to consider the publication date. If a company is declared insolvent, it can no longer meet the financial obligations to lenders as debts become due. This is reflected in the variables. Therefore, I introduce a one-year lag between the delist date and the eleven financial ratios based on all companies at all observations.

Second, I drop all missing values because I want to make as few assumptions on the data set possible. However, the downside is that if one ratio is missing and the other ten available, the whole company for a given year is dropped. Due to the abundance of data, I argue for this solution instead of other solutions, such as imputing missing values, thereby randomly generating observations. Because I will over-sample the dataset at a later stage, I run the risk of modifying the data frame so much that it is unrealistic for real-world experiments. A second consideration is to use a missing values threshold. Meaning, that if the percentage of missing values in a column surpasses 30%, then the column is dropped. The idea is to reduce the number of dropped companies (and thus information). For 30% missing values, the function drops X1 (Liquidity). For a stronger threshold of 20% only five variables remain. I argue this loss of variables is worse then deleting company information because of two reasons: Most data that is dropped is for earlier observations (before the year 2000). Due to changing capital structures (Altman, 2019), such as companies taking on more debt. The second reason is that other papers do not consider deleting variables because there are too few observations. Therefore I delete the rows where one of the columns or more has missing data.

Third, to mitigate trouble with techniques and models, infinite values are set to the maximum / minimum value recorded. I considered using windsorisation, thereby setting outliers to a specified percentile. However, one of the interest of this thesis is how outliers are dealt with in sampling.

Finally, steps 4-7 show an important stylistic choice on how to deal with companies that are active since listing. For example semiconductor producer Intel is active for the whole period 1980-2021. But what year is optimal for training a classifier if a company is still active? To mitigate this I split the data set in Active and Defaulted companies. For all active companies, I randomly selected the year of financial data. In the case of defaulted companies, I keep only the financial ratios one year to default. Then I concat the two data frames back together. This is the last step to make sure I do not delete information because a company has an empty cell for certain variables. This approach follows Veganzones and Séverin (2018) and Brown and Mues (2012).

Table 3 on page 10 presents a summary of the full sample pre- and post- data processing and it shows summary statistics of the active and defaulted companies of the dataset. Figure 2 shows the correlations between the eleven financial ratios and the target variable for the final dataset `df_sim1`. Almost all correlations are irrelevant, except X2 (Profitability) and X3 (Productivity), X3 and OM (Operational Margin). As a double check, I analyse the Variance Inflation Factor (VIF) of `df_sim1` to test the multicolliniearity between variables in Table 3. A VIF smaller than one means variables are uncorrelated, VIF between one and five is moderately correlated, and above means high correlation. I find X2 (Profitability) and X3 (Productivity) show the highest VIF slightly surpassing two. I argue the correlational relationships coupled with low VIF are weak enough to use for the sampling tests and SHAP analysis. A perfectly uncorrelated set of financial ratios is impossible to create (Balcaen and Ooghe, 2006).

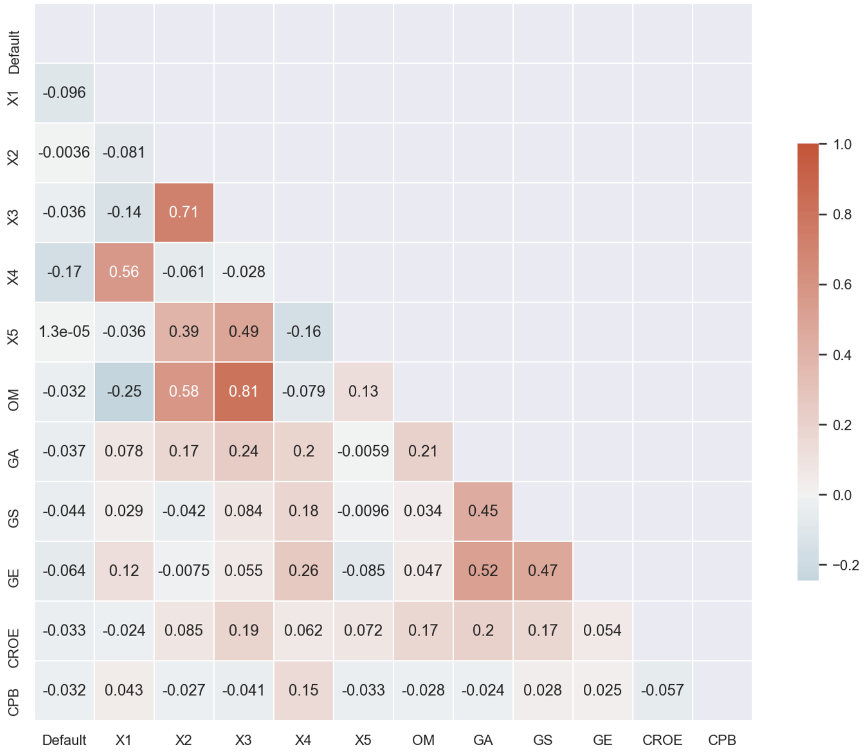


Figure 2: Correlations between financial ratios.

Table 3: Descriptive statistics for the data set

df_pre	**X1**	**X2**	**X3**	**X4**	**X5**	**OM**	**GA**	**GS**	**GE**	**CROE**	**CPB**
mean	-inf	-inf	-inf	inf	0.91	NaN	inf	inf	inf	NaN	NaN
std	NaN	NaN	NaN	NaN	1.54	NaN	NaN	NaN	NaN	NaN	NaN
min	-inf	-inf	-inf	0.00	-2.18	-inf	-1.00	-3115.30	-1.00	-inf	-inf
25%	0.07	-0.07	0.01	0.57	0.25	0.02	-0.03	-0.03	-0.05	-0.06	-0.45
50%	0.24	0.09	0.06	1.59	0.72	0.09	0.06	0.08	0.03	-0.00	0.01
75%	0.43	0.30	0.12	4.52	1.27	0.19	0.19	0.22	0.14	0.04	0.42
max	1.00	6.67	226.31	inf	227.45	inf	inf	inf	inf	inf	inf

df_sim1	**X1**	**X2**	**X3**	**X4**	**X5**	**OM**	**GA**	**GS**	**GE**	**CROE**	**CPB**
mean	0.25	-1.07	-0.06	7.29	0.89	-16.82	0.23	232.09	3.85	0.86	-1.94
std	0.68	5.19	0.68	21.41	0.88	328.66	1.36	1697.89	105.45	45.23	61.45
min	-34.61	-156.44	-28.46	0.01	0.00	-15265.30	-1.00	-1.00	-1.00	-126.18	-2310.56
25%	0.06	-0.67	-0.06	1.02	0.35	-0.07	-0.04	-0.03	-0.03	-0.11	-0.79
50%	0.21	0.02	0.05	2.51	0.70	0.06	0.06	0.08	0.04	-0.01	-0.02
75%	0.44	0.27	0.11	6.35	1.21	0.15	0.22	0.27	0.18	0.07	0.66
max	0.95	1.50	10.70	533.15	15.31	10.81	48.04	12739.00	3957.33	2510.10	659.65
VIF	1.82	2.14	2.88	1.14	1.1	1.02	1.05	1.08	1.01	1.15	1.16

Active	**X1**	**X2**	**X3**	**X4**	**X5**	**OM**	**GA**	**GS**	**GE**	**CROE**	**CPB**
mean	0.25	-1.09	-0.06	7.40	0.89	-17.11	0.23	236.27	3.92	0.89	-1.98
std	0.69	5.24	0.69	21.58	0.87	331.62	1.37	1712.90	106.40	45.64	62.00
min	-34.61	-156.44	-28.46	0.03	0.00	-15265.30	-1.00	-1.00	-1.00	-126.18	-2310.56
25%	0.06	-0.68	-0.06	1.06	0.35	-0.07	-0.04	-0.02	-0.03	-0.11	-0.79
50%	0.22	0.02	0.05	2.57	0.69	0.06	0.06	0.08	0.04	-0.01	-0.00
75%	0.44	0.27	0.11	6.48	1.20	0.15	0.22	0.27	0.18	0.07	0.67
max	0.95	1.50	10.70	533.15	15.31	10.81	48.04	12739.00	3957.33	2510.10	659.65

Default	**X1**	**X2**	**X3**	**X4**	**X5**	**OM**	**GA**	**GS**	**GE**	**CROE**	**CPB**
mean	0.11	-0.57	-0.01	2.33	1.02	-0.73	0.24	0.63	0.22	-0.18	-0.33
std	0.31	2.61	0.15	6.63	1.06	3.32	1.16	3.67	1.54	2.59	6.96
min	-1.67	-21.58	-0.68	0.01	0.00	-20.87	-0.38	-0.99	-0.88	-12.78	-55.36
25%	-0.01	-0.32	-0.04	0.18	0.29	-0.06	-0.14	-0.11	-0.13	-0.21	-0.67
50%	0.08	0.00	0.02	0.46	0.71	0.03	0.01	0.01	0.00	-0.05	-0.15
75%	0.18	0.14	0.07	1.42	1.34	0.10	0.15	0.24	0.08	0.08	0.16
max	0.98	0.91	0.21	41.82	4.65	0.66	9.85	32.32	13.71	16.32	23.36

Notes: Descriptive statistics (mean, standard deviation, minimum, 25%-,50%-,75%-percentile and the maximum). VIF is the Variance Inflation Factor of each variable. The first data set is after merging the CRSP Delist and CRSP / Compustat Merged data sets before data cleaning. The second data set is after the seven pre-processing data steps. The third and fourth data sets are is the data before concating in step 7 for the defaulted companies (step 5) and the active companies (step 6).

4. Methodology

This chapter consists of four sections. Figure 3 is an illustration of the steps used to construct this research. I cover the first two steps, the Database, and Data Pre-processing, in chapter 3. In that chapter, I describe the data source and the process of generating the variables from the raw data. The following four (sub-) steps are discussed in this chapter. Firstly, I describe the different sampling techniques. Secondly, I describe the statistical methods and Machine Learning models used to predict bankruptcy. Thirdly, I discuss SHapley Additive exPlanations (SHAP) to study the effect of sampling on feature importance. Finally, I describe the techniques to evaluate the model performance.

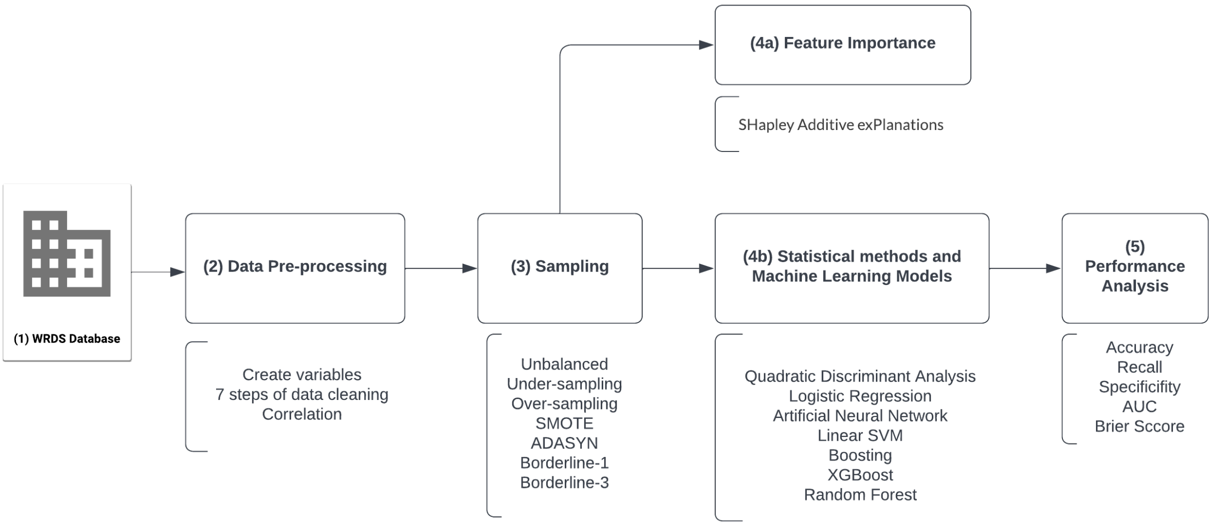


Figure 3: Illustration steps that are detailed in the methodology and data chapters.

4.1. Sampling

A data set is unbalanced if one value of the target variable has more observations than another value of the target variable. The value with more observations is the *majority class*, and in this research is the value ‘0’ for companies that do not default within one year. The *minority class* here is the value ‘1’ for companies that default within the next year. The higher the imbalance, the harder it becomes for the model to correctly identify the minority class. For example, if one trains a model to only predict ‘active’ companies for a data set with a 95/5 imbalance, then it achieves a 95% accuracy. However, it is incapable of identifying the bankruptcy cases that one is interested in. Imbalanced data sets are a prevalent problem in data science (Fernandez et al., 2018), and in bankruptcy prediction the minority class is less than 5% (Veganzones and Séverin, 2018). Therefore, *sampling*, or balancing data sets to create equal an equal number of data points for the majority and minority classes, is essential for programming reliable models. Reliable, in the sense that, the number of missclassified bankruptcies is minimised. Unbalanced data sets prioritise accuracy and create higher number of missclassified bankruptcies. However, as discussed in section

2, there is no extensive research which sampling technique best fits the bankruptcy classification challenge. Re-sampling techniques are divided into four categories: under-sampling the majority class, over-sampling the minority class, combining over- and under-sampling, and creating ensemble balanced sets. Due to the scope of a thesis, I only consider the first two categories. In the following paragraphs, I discuss different ways to sample a data set.

To illustrate the effect of sampling I create a randomly generated dataset with two features and a target variable 'Active' and 'Default'. This is the benchmark, and shows how the different sampling techniques affect the majority (Active) and minority class (Default). The two class have 4800 and 200 observations respectively. Figure 4 shows blue squares for minority class and only present 2% of the dataset. The orange dots present 98% of the data set. I re-sample this dataset using the following techniques to illustrate and elaborate on how sampling techniques work, including the benefits and the weaknesses.

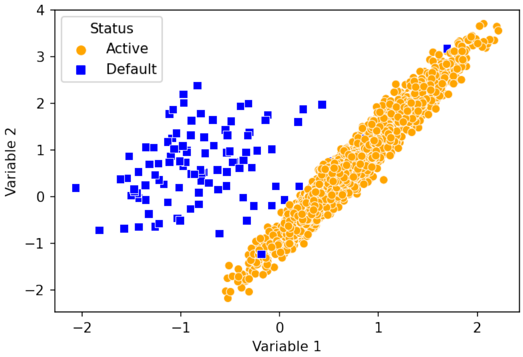


Figure 4: Benchmark dataset, this an unbalanced randomly generated data set.

In bankruptcy prediction random **under-sampling** is the standard method to sample an unbalanced data set. Here one randomly deletes observations of the majority class until there is an equal ratio between the two instances. Figure 5 shows how the majority class, the orange dots, are reduced. Thereby the remaining data set is only 4% of the size of the original data set. The decrease in data points makes the computation time faster. A disadvantage is the loss of potentially important information. Furthermore, the remaining dataset can be an inaccurate representation of the population because only only a random subset remains. There are several ways to combat this, such as stratified sampling, cross-validation, and other options. This is outside the scope of the thesis.

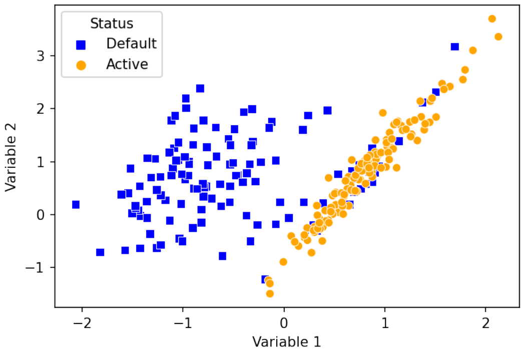


Figure 5: Randomly under-sampled benchmark data set.

Random **over-sampling** is duplicates observations based on the minority class. The main advantage is no information loss, but at the risk of over-fitting on the minority class. Figure 6 plots the benchmark data with new observations to total 9800. Here another disadvantage is apparent, random over-sampling is a naive sampling method, as it uses no heuristics. In other words, all observations, even outliers, are oversampled. As discussed in section 2, random-undersampling is superior at building classifiers than random over-sampling (Chawla et al., 2002).

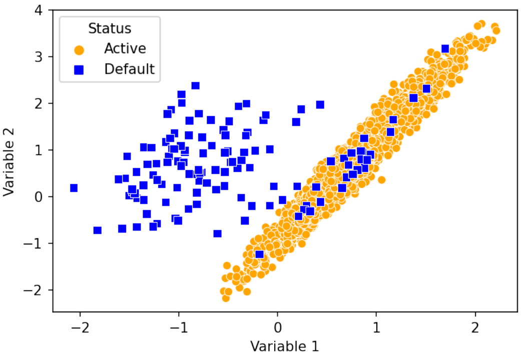
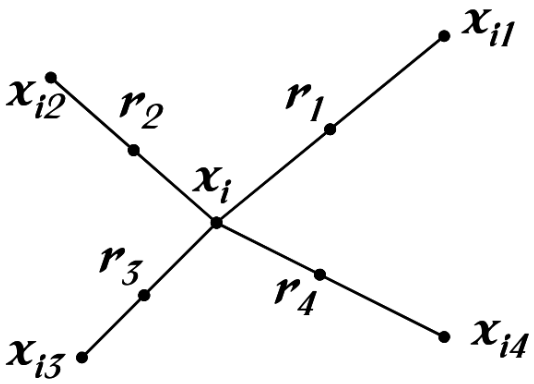


Figure 6: Randomly over-sampled benchmark data set.

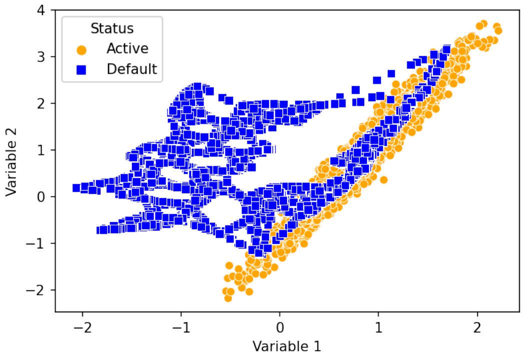
To address the short-coming of under-sampling and over-sampling Chawla et al. (2002) published a seminal paper that introduced the Synthetic Minority Over-sampling Technique, or **SMOTE**. In fraud detection and credit risk, the algorithm is common practice, except in bankruptcy prediction. Despite that, all applications are notoriously unbalanced. In contrast to over-sampling, SMOTE algorithm creates 'synthetic' data points. These synthetic data points are newly generated data, whereas, in random over-sampling, the data is duplicated. Thus SMOTE addresses the main disadvantage of random over-sampling because it reduces overfitting. It works as follows; consider Figure 7a, depicting 'real' data point x_i and four 'real' neighbours $x_{i1}, ..., x_{i4}$. SMOTE uses K-nearest neighbours and identifies the K data points with the shortest distance to x_i. For this example $K = 4$, the default setting is $K = 5$. Next, SMOTE randomly assigns new synthetic points $r_1, ..., r_4$ along the line between x_i and its neighbours. SMOTE generates a synthetic data point with equation 1.

$$r_j = x_i + \text{rand}(0, 1) \cdot (x_i - x_{ij}) \tag{1}$$

Where $\text{rand}(0, 1)$ generates number generator between 0 and 1 using a uniform distribution and j is the nearest neighbours between $[1, ..., K]$. Figure 7b shows how the benchmark data set of Figure 4 is re-sampled. Notice the 'bridges' between the blue cloud and the orange cloud. The lines result from the K nearest neighbours approach to generating data points on the line between two existing data points. However, generating new observations makes no consideration of how many majority data points surround a minority data points. Again, consider Figure 4 with one blue square in the right upper corner in a cloud of blue points. SMOTE constructs 'bridges' to outlier as shown in Figure 7b. This is the main disadvantage of SMOTE, where all minority class observations are over-sampled equally. In the years following the publication of this algorithm, academics created over 80 SMOTE-based extensions (Fernandez et al., 2018). I consider two of the most popular extensions that address the main disadvantage: Adaptive Synthetic and Borderline-1. I chose these two additions because these are the most well-known extensions and regularly used in other academic fields (Fernandez et al., 2018).

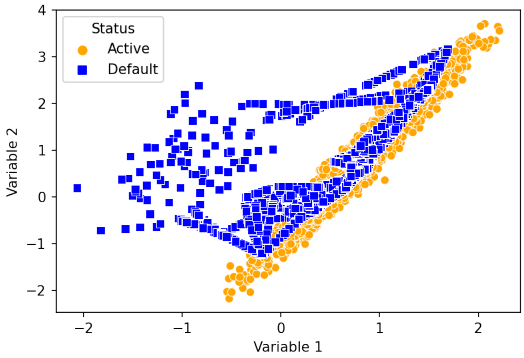


(a) Illustration of how to create the synthetic data points in the SMOTE algorithm. Image from Fernandez et al. (2018).

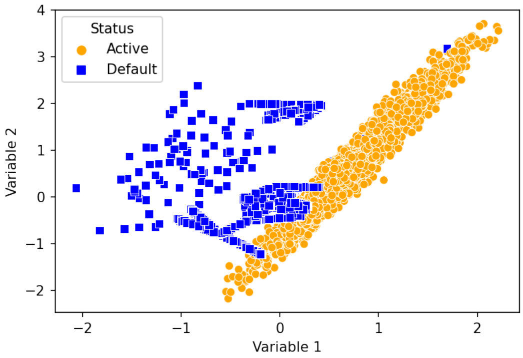


(b) SMOTE benchmark

ADASYN (He et al., 2008), short for Adaptive Synthetic, generates new samples based on the local distribution of majority and minority classes. The main difference between SMOTE and ADASYN, is that the latter uses an impurity ratio for the minority class to create synthetic points. Impurity (imp_i) is found for observation x_i by dividing majority neighbours (Δ_i) by K-nearest neighbours (K). The higher the ratio of majority to minority class observations, thus higher impurity, the more additional points are generated. The number of additional synthetic points for observation x_i is based on the normalised impurity ratio $\hat{imp_i}$. A full derivation is available in appendix C on page 37. The idea is to create synthetic hard-to-learn samples, i.e. neighbourhoods dominated by the majority class. Therefore, it is adaptive because the classification decision boundary is shifted towards the impure samples. Figure 8a shows that ADASYN indeed prioritises the impure samples, as the outliers closer to the blue cloud are oversampled. Whereas the inliers of the orange cloud are minimally oversampled. Minority samples with high number of majority classes get more synthetic points. The synthetic points therefore are similar to the majority class and make it harder for the algorithm to distinguish majority and minority classes and potentially increase the number of false positives.



(a) ADASYN over-sampled benchmark data set.



(b) Borderline over-sampled benchmark data set benchmark

In **Borderline**-SMOTE (Han et al., 2005), synthetic data points are only made along the borders of the majority and minority class. There exist two Borderline SMOTE: '1' and '2'. Number '1' only over-samples the minority class, whereas number '2' also under-samples the majority class if misclassification can occur. As stated earlier, I only consider under-sampling and over-sampling and no combinations. Therefore I limit this analysis to Borderline-1. It is outside the scope of the thesis to study all extensions. Each minority observation is classified in one of three ways:

- Noise: all nearest neighbours are from the majority class. These observations are *not* over-sampled.
- Danger: when the attribute K neighbours (default $= 5$) over M neighbours (default $= 10$) exceeds a predetermined value. These observations are oversampled.
- Safe: K neighbours (default $= 5$) over M neighbours does not exceed threshold. These observations are *not* oversampled.

In Figure 8b, one can see that only the border observation are oversampled. The number of bridges appears significantly smaller. Furthermore, observations in the blue cloud, are correctly classified as 'Noise'. However, like ADASYN, the main disadvantage of this algorithm is that it over-samples observations minority classes with a high number of majority neighbours. This increases the risk of generating false positives in a test set because a model has trouble distinguishing minority from majority classes.

To mitigate that disadvantage of SMOTE and specifically Borderline-1, I created a new extension: **Borderline-3**. This extension is over-samples the minority classes with a higher number of minority neighbours instead of majority neighbours (like ADASYN and Borderline-1 do). Thus it over-samples 'safe' observations instead of 'danger'. The motivation is to address the disadvantage of Borderline-1 that 'danger' are harder to classify for a model such that it creates more false positives. Algorithm 1 shows how it generates synthetic samples. Suppose that the whole training set is T, the minority class is P and the majority class is N. Here $P = \{p_1, p_2, \ldots, p_{pnum}\}$ and $N = \{n_1, n_2, \ldots, n_{nnum}\}$, where *pnum* and *nnum* are the number of minority and majority observations. The procedure of Borderline-3 works as shown on page 16.

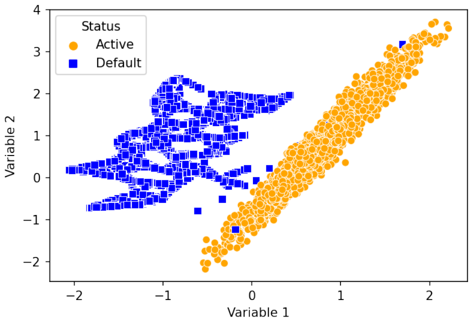


Figure 9: Borderline-3 oversampled benchmark data set.

```
Algorithm 1: Borderline-3 algorithm
Step 1. For every $p_i$ in the minority class $P$, calculate the $m$ nearest neighbours for the
  training set $T$. The number of majority examples among the $m$ nearest neighbours is denoted
  by $m'$ where $0 <= m' <= m$.
Step 2. Classify the observations as NOISE, DANGER, and SAFE.
if $m' = m$ then
    All nearest neighbours of $m'$ are majority class, the observations is labelled as NOISE.
else if $\frac{m}{2} <= m' < m$ then
    The number of $p_i$'s majority neighbours exceeds the number of minority neighbours, the
      observations is labelled as DANGER.
else
    The number of $p_i$'s minority neighbours exceeds the number of majority neighbours, the
      observations is labelled as SAFE.
end
Step 3. For each observations in set SAFE, calculate it's $K$ nearest neighbours from minority
  class $P$.
$SAFE \in \{p'_1, p'_2, ..., p'_{snum}\}, 0 <= snum <= pnum$
Where $snum$ is the number of safe points and $pnum$ is the number of minority class
  observations.
Step 4. Generate $N - P$ synthetic observations
for $p'_i$ in $SAFE$ do
    $s$ is an integer between 1 and $k$.
    randomly select $s$ nearest neighbours from the $k$ nearest neighbours in $P$
    for $j = 1, 2, ..., s$ do
        (a) Calculate the difference $dif_j$ between $p'_i$ and it's nearest neighbour from $P$.
        (b) Multiply $dif_j$ with a random number $r_j$ from a uniform distribution between 0 and
          1.
        (c) Generate a new synthetic minority between $p'_i$ and its nearest neighbours:
        $synthetic_j = p'_i + r_j \cdot dif_j$
    end
end
```

Table 4 is a summary table of the sampling techniques. The main conclusions from the discussion are this. Unbalanced training sets are not adapt in predicting minority classes; therefore one considers sampling. The most popular technique in bankruptcy prediction literature is undersampling. This disadvantage is the loss of information of the majority class. One can over-sample to mitigate this by duplicating observations, however, this leads to significant overfitting. Therefore random under-sampling is better adapt at creating classifiers than random over-sampling (Chawla et al., 2002). SMOTE solves the overfitting problem by creating synthetic data points between observations, and therefore correlating. However SMOTE handles all observations as equal and is therefore prone to outliers. Newer SMOTE-based extensions aim to solve this issue, including the extensions ADASYN and Borderline-1. The thesis' extension, Borderline-3, focuses on over-sampling the 'safe' observations. I expect Borderline-1 to outperform unbalanced and over-sampling because the majority and minority classes are evenly distributed, and the synthetic points should reduce overfitting. Furthermore, Borderline-1 prioritises extreme observations and should therefore see better performance of recall due to a lower number of false negatives. Finally, all sampling techniques rely on randomness. Therefore all models and programs used in this thesis use a global random seed. This seed sets the random state to '1996' to make the results reproducible. The exact number is chosen arbitrarily, but necessary to treat it as an immutable variable to make the test replicable and the results definitive.

Table 4: Summary of sampling techniques.

Sampling Technique	**How**
Unbalanced	Do nothing, initial data set
Random under-sampling	Randomly delete observations
Random over-sampling	Randomly duplicate observations
SMOTE	Create synthetic data points using equation 1
ADASYN	Create synthetic data points based on the impurity density distribution of observation i. See appendix C for derivation.
Borderline-1	Generate synthetic data points if the number of nearest neighbours of observation i does crosses the threshold and if it not equal to K.
Borderline-3	Generate synthetic data points if the number of nearest neighbours of observation i does not cross the threshold. See algorithm 1.

4.2. Statistical Methods and Machine Learning Models

Barboza et al. (2017) evaluated bankruptcy prediction using industry-standard statistical methods and compared the performance to more recent Machine Learning models. I replicate their variables and compare seven classifiers. I additionally sample the data according to the techniques in section 4.1. I discuss the following categories of classifications, in order of appearance: statistical models, artificial neural networks, support vector machines, and ensemble techniques.

4.2.1. Statistical Methods

For completeness follows Barboza et al. (2017), by what they define as statistical techniques. Specifically, multiple discriminant analysis (MDA) from Altman (1968) and Logistic Regression from Ohlson (1980).

Balcaen and Ooghe (2006) argue MDA has restrictive assumptions. Firstly, MDA assumes that any linear combination of the features is normally distributed. However as noted by Shin and Lee (2002), financial data is inherently non-normal. Secondly, it has equal variance-covariance matrices between active and bankrupt groups such that $Cov(x, y) = Cov(y, x)$. However, Balcaen and Ooghe (2006) argue financial data rarely satisfies this assumption. This leads to significant biased tests. Furthermore, the authors note that one should use Quadratic MDA models over Linear MDA models because this addresses the unequal dispersion matrices. However, Quadratic MDA models rarely outperform Linear MDA (as used by Altman (1968)). Taking this into consideration, I use Quadratic MDA, **QDA** henceforth for the analysis. Thirdly, MDA assumes the absence of multicollinearity between features. As shown by Table 3 the highest VIF values are for X2 (Profitability) and X3 (Productivity) in `df_sim1`. Thus there is moderate multicollinearity.

I use Logistic Regression, **LR** hereafter, as a classifier because it is easy to implement and efficient to train. A derivation is in appendix D on page 37. Balcaen and Ooghe (2006) identify the following two restrictive assumptions for LR: multicollinearity and outliers. Furthermore, bankruptcy is by definition outliers because they happen infrequently. To address (some) of the issues, I considered normalising the features and windsorising the dataset. For the research purposes, this is omitted because I want to make as few changes to the data as possible following Barboza et al. (2017).

4.2.2. Artificial Neural Networks

Previous literature that analysed bankruptcy prediction with Artificial Neural Networks (ANN) include Wilson and Sharda (1994), Kim and Kang (2010), and Barboza et al. (2017). The analogy of neural networks mimicking human neural processing is frequently made. This semi-parametric approach makes recursive use of linear combinations and non-linear transformations. ANN exists of input layers, multiple hidden layers, and an output layer. Using back-propagation, the weights of layers are changed by the difference between the calculated output values and the true output values.

For the analysis, I use the MLP Classifier of the python package scikit learn. Most function are set to default, expect activation function $h(a)$, where a is the scalar. The following functions are considered:

- **Logistic**, A logistics sigmoid function that returns $h(a) = 1/(1 + e^{-a})$. Return values are between 0 and 1.
- **Tanh**, A hyperbolic tangent function that returns $f(a) = tanh(a)$. Returns values are between -1 and 1.

Because the target variable is binary and the financial data is not normalised and non-linear, these two options are the most suitable options. A disadvantage of the logistic activation function is that it can slow down the gradient descent of the back-propagation. For example, if the training input is a large negative number, the weights are less regularly updated. In contrast, the hyperbolic tangent function maps this to -1. Kim and Kang (2010) used logistic as activation function, whereas Barboza et al. (2017) used the hyperbolic tangent as activation function. Wilson and Sharda (1994) does not specify the activation function. Due to the slower processing time and the relatively large data set, I set the activation function to the tangent hyperbolic.

4.2.3. Support Vector Machines

A Support Vector Machine (SVM) creates a hyperplane such that the data points are classified according to the side of the hyperplane that they reside (Cortes and Vapnik, 1995). For SVM, I only consider binary classification. However, financial data is not perfectly separable. Therefore hyperplanes are constructed such that there is a 'soft' margin of error. To optimise for SVM, one uses a mapping function that creates higher dimensions kernels to get better data separation. The disadvantage of higher dimensionality kernels is higher complexity and, thus slower computing time. For the thesis, I considered linear and radial basis function kernels following the methodology of Barboza et al. (2017). I opt for linear SVM because radial basis function SVM shows a long computation time and disappointing initial results.

4.2.4. Ensemble Methods

The idea of ensemble methods is to combine a multitude of weak learners such that the combined model outperforms the individual learners (Hastie et al., 2017). These learners can be ensembles sequentially, known as boosting, and in parallel, known as Bagging. In **Boosting**, a random sample of the training set is created and fitted to a shallow decision tree (Freund and Schapire, 1997). Then it updates the weight of the data samples based on the inaccuracy of the previous decision tree. In recent years eXtreme Gradient Boosting, or **XGBoost**, became popular on the Machine Learning competition website Kaggle. This model uses regularization to improve gradient descent towards the target outcome.

In **Bagging**, formally known Bootstrap aggregating, new data sets are created from the original training data set. The algorithm randomly selects observations, sometimes multiple times, and for each data set a classifier is built. Using averaging methods or voting methods a universal classifier is built. **Random Forest**, RF, is a type of Bagging. The critical difference is that RF uses only some of the variables for creating new data samples. In other words, Bagging uses all columns whereas RF randomly selects the columns for its newly created training sets. The main advantage is that each decision tree is de-correlated, therefore more independent and improving the ensemble prediction.

Barboza et al. (2017) found Boosting, Bagging and RF provided the highest predictive capacity. I add XGBoost, due to its popularity and compare it to the performance of Boosting in this bankruptcy setting. I omit Bagging because RF generally outperforms it.

4.3. SHapley Additive Explanations

The inherent nature of most ML models is similar to a black box. The complex and non-linear characteristics make it very difficult to understand the models' underlying decisions. Several methods allow for explore the inner workings of Machine Learning models. One approach is to quantify feature importance using Spearman's correlation coefficient between the features and the target variable. Other techniques allow for a deeper analysis and are known as explainable Artificial Intelligence. These techniques include Partial Dependence (Friedman, 2001), Accumulated Local Effects (Apley and Zhu, 2020), and SHapley Additive exPlanation (SHAP) (Lundberg and Lee, 2017). The latter method offers the following benefits over the former methods, namely: global interpretability and local interpretability.

This framework is based on cooperative game theory where features form subsets. The framework identifies the marginal contribution for each feature in each subset. The Shapley value then represents the weighted average contribution of each feature for all subsets. From Lundberg and Lee (2017), I use equation 2 to approximate Shapley value ϕ_i for machine learning model f for a specific input x for feature i:

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!} \left[f_x(z') - f_x(z' \backslash i)\right] \tag{2}$$

Where z' is a subset of features, and x' is all possible subset not containing the simplified input x'. For the exact mathematical deduction, please refer to Lundberg and Lee (2017) on the property of Local Accuracy. It translates to x' is equal to x. Here M is the total number of subset. The fraction represents the weighting according to how many features are in the subset. The deduction of f_x represents the output of the black box model with and without feature i. The difference is the contribution of the feature to the prediction in this subset.

Figure 10 is an illustration of how to interpret the values of each feature. A trained model has a base value $E[f(z)]$, which shows the expected value of the output if no feature values are known, this is the mean prediction of the model. For each input x, the value shifts towards a new estimated value. For example, ϕ_1 is the shift for feature i with input x_1. After all inputs, the value is equal to the model output $f(x)$, where all ϕ's show the Shapley values to arrive at that output.

This example is for a single set of inputs and provides an explainable way as to why a particular observation (here a specific company with a set of financial ratios) defaults or not. Furthermore, Shapley values are useful to find the mean absolute Shapley value for each feature j:

$$I_j = \frac{1}{n} \sum_{i=1}^{n} |\phi_j^{(i)}| \tag{3}$$

Where n is the number of observations. This allows for analysis on the magnitude of feature attribu-

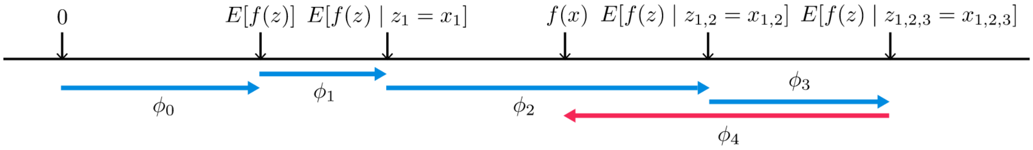


Figure 10: Illustration of how Shapley values work from Lundberg and Lee (2017)

Notes: $E[f(z)]$ is the mean predicted output of the model. For each additional feature i with input x the expected output shifts with Shapley value ϕ.

tions to attain global interpretability. Such that one can analyse how important a feature is to the model. This is superior to classical feature importance, where one uses the correlation coefficient between the features and the target variable. Then one studies the data. However, the Shapley value is the average expected marginal contribution of one feature after all other combinations are considered.

For the analysis I use use the SHAP python package [3]. I chose the XGBoost as Machine Learning model because the documentation is more extensive than other models [4]. Furthermore, the Shapley package allows for ‘beeswarm' summary plots. Each dot represents the Shapley value and the feature value for each feature. This is useful to find relationships between model output and individual features. Moreover, the Shapley package can create dependence plots as an alternative to partial dependence plots and accumulated local effects. SHAP allows for additional information to the other methods. It can show the variance of observations instead of only the average effects. Moreover, it can highlight feature interactions of a chosen variable.

However, Shapley a potential disadvantage is correlation bias. This occurs when during the training of the model, and Shapley has no method of correcting for it. Consider two highly correlated features A and B. XGBoost assigns the highest weight to one of the two features, here A. The trained model assigns high Shapley values to A and not to B to the way the model is constructed. Thus even if feature B is informative, the model neglects its explanatory power because the model is trained on feature A. Therefore, it is vital to use a data set with low multicollinearity.

Finally, the benefits of global interpretation, with feature attribution and summary plots, and local interpretation, with dependence plots make this framework a powerful tool for explaining machine learning algorithms. These visualisations and the better interpretability make it a powerfull tool for loan providers to explain why a model labels a company as default or not. The only financial prediction literature that uses SHAP today is for mortgage defaults prediction in a working paper by Bracke et al. (2019).

[3] Documentation available at `https://shap.readthedocs.io/en/latest/index.html` accessed on June 1st, 2022
[4] `https://christophm.github.io/interpretable-ml-book/shap.html` accessed on June 1st, 2022

4.4. Performance Analysis

To evaluate the performance of the models for different sampling techniques, I follow the relevant literature and use the following classification metrics: Accuracy, Recall, Specificity, and Area under ROC Curve (AUC). Additionally, as a forecasting measure, I use the Brier score. Machine Learning algorithms are typically analysed using a confusion matrix that assigns classification to four possible cases, summarised in Table 5.

Table 5: Confusion Matrix.

		Predicted Category	
		Positive	Negative
True class	Default (0)	True Positive (TP)	False Negative (FN)
	Active (1)	False Positive (FP)	True Negative (TN)

- True Positives (TP): Classifies a company defaults and does so within the next year.
- True Negative (TN): Classifies a company that remains active and defaults within the next year.
- False Positive (FP): Type I error. Classifies a company as defaulting within the next year, but does not. This presents lost opportunities. Companies might fail to get additional loans to grow and expand. Furthermore, lenders lose an opportunity to safely sell loans.
- False Negatives (FN): Type II error. Classifies a company to remain active within the next year when in facts it defaults. This creates lost equity for shareholders, lost jobs for workers, and defaulting loans for lenders.

I measure the performance using these four cases as follows. Firstly, accuracy (AC), as shown in equation 4, is the total number of correctly labelled companies divided by the total number of companies in the dataset. Accuracy is a great measure when one has a balanced data set, but this is not the case for bankruptcy data sets. In the case of unbalanced data sets, this leads to a deceptive interpretation of a models performance. Considering 1% imbalance and all companies are classified to remain active in the next year, our model is 99% accurate. However, the companies that default are of interest here, so I consider additional measures of performance, following the performance measures of Chawla et al. (2002). Specificity (SP), or False Positive Rate (FPR), is shown in equation 5. This measures the ratio of misclassified active companies over all active companies in the data set. Recall (RE), or True Positive Rate (TPR), measures ratio of misclassified defaulted companies over the total number of defaulting companies. Recall is also referred to as sensitivity. Specificity is a threshold for potential gain, whereas recall is a threshold for potential loss. In the earlier example, specificity is 100% and Recall 0%. A perfect classifiers attains 100% for both.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$\text{Specificity} = 1 - \text{Type I Error} = \frac{TN}{TN + FP} = 1 - FPR \tag{5}$$

$$\text{Recall} = 1 - \text{Type II error} = \frac{TP}{TP + FN} = TPR = \text{Sensitivity} \tag{6}$$

Moreover, I can combine the two measures, specificity and sensitivity, for different thresholds to plot the Receiver Operating Characteristic (ROC). Here the X-axis represents the specificity, and the Y-axis the recall. Figure 11 shows three hypothetical curves. Plot A is a perfect classifier with no False Positives or False Negatives. Plot C, when $1 - FPR = TPR$, is when one randomly guesses classes. Plot B is then between a perfect and random classifier; the closer it gets to A the better. The Area under ROC Curve (AUC) allows for comparing classification performance independent of decision criteria for specificity and recall (Fawcett, 2006). The higher the AUC the better the model can separate classes. Therefore, it is a useful tool for comparing models, but it is sub-optimal if one prefers to minimise Type I or Type II error. That is, AUC can be high for low recall, such that the risk of type II error for loan providers it too high to accept.

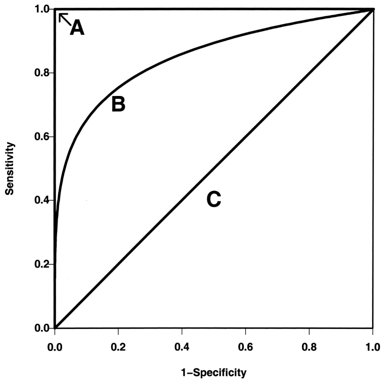


Figure 11: Illustration of a ROC curve.

Finally, I test the prediction performance of the models using the Brier Score (Brier et al., 1950). Previous literature on bankruptcy prediction and sampling omits measures of probabilistic forecast performance. I include this measure to identify the statistical certainty of prediction. The Brier score is calculated as

$$\text{Brier} = \frac{1}{N} \sum_{t=1}^{N} (f_t - o_t)^2 \tag{7}$$

Where N is the total number of observations in the data set, f_t is the forecast probability of default, and o_t the true outcome of the event for observation t. For example, if a model predicts that a company defaults with either 60% or 80% certainty, then the Brier score identifies which prediction is more 'correct'. The Brier score is between 0.0 and 1.0, with 0.0 being a perfect score and 1.0 being the worst possible outcome. If that company defaults then the forecast with 80% has a lower, and thus better, Brier score then the forecast with 60%.

5. Results

The current state of the literature has not addressed the effect of sampling on the predictive performance of models for bankruptcy. This section is addresses this following the methodology of section 4. First, I test the effect of sampling on eight statistical techniques and Machine Learning models. The sampling methods are as discussed in section 4.1 and the performance measures in section 4.4. Second, I analyse the effect of sampling on feature importance. This is two-fold, the global and local interpretability, using Shapley Additive Explanations as explained in section 4.3.

5.1. Simulation 1: The effect of sampling on bankruptcy prediction

I split the data set of chapter 3, `df_sim1`, into two sets: training and test. The split is randomised with a ratio of 67:33. This split omits the element of time, such that observations in the train and test split are from observations across the whole time line. As described in section 3.2, I randomly select one observation per 'active' company over the whole timeline. The motivation is to guarantee that a defaulting company (within two or more years year) does not interfere with training the models. Such that a defaulting company in two years is dropped from the data set rather than labelled as 'active'. The assumption here is that a company with poor financial accounting data is observed in the years prior to bankruptcy. How many years prior to bankruptcy this is measurable is outside of the scope of the thesis. I keep one observation of companies that default within one years, I do the same for companies that remain active (over the whole timeline). This randomised approach already neglects the time dimension and therefore it makes no sense to do a time split.

Additionally, I test the performance of the sampling techniques for XGBoost after hyperparameter tuning. That is, I uses grid search cross validation for a split of train-validation-test of 60:20:20. With only 86 defaults in the whole data set, the validation set only has 17 cases of bankruptcies. Furthermore, the total dataset for under-sampling for the validation set then only contains 34 observations. For over-sampling, all new replicated, or synthetic points are based on a tiny set of companies.

Table 6: Overview of data `df_1` before sampling.

Data frame	**Total Observations**	**Number of Defaults**	**Imbalance (%)**
`df_sim1`	3182	86	2.72%
Training (Unbalanced)	2131	58	2.72%
Training (Undersampling)	116	58	50.00%
Training (Oversampling Methods)	4146	2131	50.00%
Training (ADASYN)	4132	2059	49.83 %
Test	1051	28	1.76%

5.1.1. Does sampling affect the performance of bankruptcy prediction?

Table 6 shows the size of the splits with different sampling techniques. The training set of under-sampling contains only 116 observations compared to 4132 or 4146 observations for over-sampling based techniques. The test set remains unbalanced because this is out-of-sample; therefore sampling makes no sense.

The under-sampling results for the seven classifiers are consistent with Barboza et al. (2017). ANN, Boosting, XGBoost, and RF demonstrate the higher accuracy, AUC, and specificity than QDA and LR. Figure 12 shows the ROC curve for all under-sampling methods. Furthermore under-sampling is the most consistent sampling technique across all models except Linear SVM. Consistent because under-sampling demonstrates values for AUC (higher or equal to 0.625) and recall (higher or equal to 75%). Table 7 shows random over-sampling is too naive (Chawla et al., 2002), and a 'smarter' method is required if one chooses to over-sample. Random over-sampling over-fits by duplicating minority classes. SMOTE-based sampling techniques are adapt at improving recall over over-sampling techniques. However recall for SMOTE and extensions are close to or below 50%, making this an inferior option for parties that want to minimise the type II error. For example, if credit supplier uses SMOTE with Random Forest then he or she accepts that nearly four out of five companies go bankrupt, which is too high. The low values for recall for SMOTE-based sampling techniques, especially for Boosting, XGBoost, and RF make under-sampling the most consistent under-sampling technique.

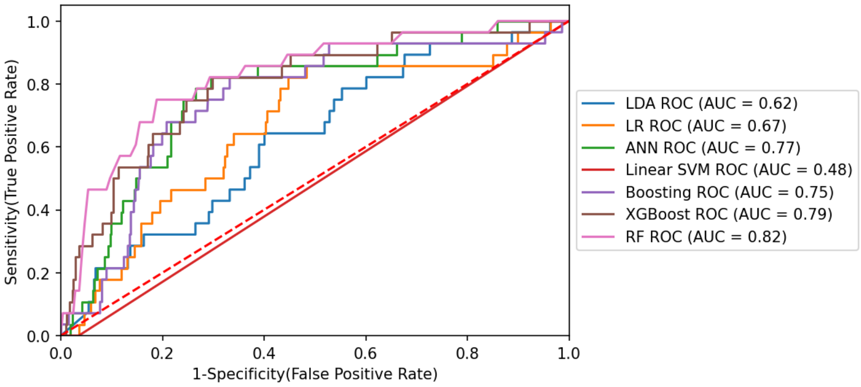


Figure 12: ROC Curve for under-sampling for seven classifiers.

Overall, QDA is a poor predictor for bankruptcy. Although recall shows all data sets correctly predicts bankruptcy over 82% of the time, accuracy and specificity suffer for it because the statistical technique classifies over half of the companies in the test set as bankrupt. For a loan provider translates to missing many loan interest opportunities. These observations are consistent with Shin and Lee (2002) who argue that statistical methods suffer from non-normal data and outliers. Notably, Borderline-1 with LR shows outstanding performance and beats under-sampling with Random Forest based on accuracy, AUC and Brier score. The performance is also seen in figure 16 on page 39. This shows that classical techniques can outperform machine learning models with a machine learning approach for sampling. Borderline-1 is built with K-nearest neighbours and uses outlier detection. Furthermore, Figure 18 on page 40 shows that AUC of the Linear SVM for the unbalanced and under-sampling data set perform worse than a random classifier (dotted red line). Again Borderline-1 is the best performing sampling technique with high recall (76.7%) and specificity (82.1%).

It is striking that the extensions, ADASYN and Borderline-1, behave differently for other models. For example, Borderline-1 improves on SMOTE in both specificity and recall for QDA, LR, Boosting and linear SVM. However, it deteriorates recall for ANN, XGBoost, and Random Forests. Borderline-1 can identify three types of observations, creating only synthetic points between 'danger' points. In contrast, ADASYSN creates synthetic points for changing impurity levels. ADASYN shows higher recall than Borderline-1 for all models except boosting. But creates more false positives because it over-samples on outliers with high impurity ratio. unlike Borderline-1 which does not over-sample 'noise' observations.

The Borderline-3 algorithm show recall of sub 15% for all classifiers but QDA. The prioritisation of 'safe' points of Borderline-3 actually harms the performance. A possible explanation is that synthetic points are only created for a small cluster of data points. Put differently, only a small number of observations are classified as safe with little variance. This 'safe' cluster omits too much information of more extreme data points such as the 'danger' observations in Borderline-1. A future iteration of Borderline-3 could combine the strengths of Borderline-1 and ADASYN. The program identifies noise points where minority class nearest neighbours are exclusively majority class. It uses an impurity ratio for the remaining observations to construct more synthetic observations for more impure observations. In borderline-1, danger points get an equal number of synthetic points, and ADASYN creates new observations based on the impurity for all observations. Thereby this new extension creates only synthetic observations for 'safe' and 'danger' points but the number of each observation is determined by the impurity (like in ADASYN). Instead of all creating an equal number of observations for all 'safe' and 'danger' points (like in Borderline-3 or -1).

Finally, the Brier score is uninformative because it behave very similar to accuracy. The lowest Brier scores are in general for the Unbalanced data set and Borderline-3 sampled data set. These also show the highest accuracy. This shows why measuring AUC and Recall are more informative, because if a model (almost) exclusively predicts that a company remains 'active' then accuracy and Brier show stellar results, but a loan provider risks blindly providing loans to company that default in the next year.

Table 7: Classification results for sampling methods using 11 variables of Barboza et al. (2017).

Model	**Data**	**TP**	**TN**	**FP**	**FN**	**AC (%)**	**AUC**	**SP (%)**	**RE (%)**	**Brier**
QDA	Unbalanced	25	166	857	3	18.2	0.621	16.2	89.3	0.816
	Under-sampling	**27**	101	922	**1**	12.2	0.625	9.9	**96.4**	0.874
	Over-sampling	25	149	874	3	16.6	0.619	14.6	89.3	0.831
	SMOTE	24	203	820	4	21.6	0.591	19.8	85.7	0.783
	ADASYN	24	190	833	4	20.4	0.596	18.6	85.7	0.794
	Borderline-1	23	**332**	**691**	5	**33.8**	**0.633**	**32.5**	82.1	**0.659**
	Borderline-3	25	166	857	3	18.2	0.621	16.2	89.3	0.816
LR	Unbalanced	0	**1023**	**0**	28	**97.3**	0.714	**100.0**	0.0	**0.026**
	Under-sampling	**24**	513	510	**4**	51.1	0.666	50.1	**85.7**	0.250
	Over-sampling	0	1016	7	28	96.7	0.686	99.3	0.0	0.116
	SMOTE	21	490	533	7	48.6	0.682	47.9	75.0	0.230
	ADASYN	22	487	536	6	48.4	0.699	47.6	78.6	0.230
	Borderline-1	21	771	252	7	75.4	**0.841**	75.4	75.0	0.141
	Borderline-3	0	**1023**	**0**	28	**97.3**	0.714	**100.0**	0.0	**0.026**
ANN	Unbalanced	0	**1020**	**3**	28	97.0	**0.789**	**99.7**	0.0	**0.027**
	Under-sampling	**23**	698	325	**5**	68.6	0.765	68.2	**82.1**	0.215
	Over-sampling	12	947	76	16	91.2	0.743	92.6	42.9	0.066
	SMOTE	11	918	105	17	88.4	0.725	89.7	39.3	0.089
	ADASYN	12	905	118	16	87.2	0.715	88.5	42.9	0.099
	Borderline-1	8	963	60	20	92.4	0.730	94.1	28.6	0.062
	Borderline-3	0	**1020**	**3**	28	97.0	0.772	**99.7**	0.0	**0.027**
Linear SVM	Unbalanced	0	**1023**	**0**	28	**97.3**	0.591	**100.0**	0.0	**0.026**
	Under-sampling	**25**	425	598	**3**	42.8	0.451	41.5	**89.3**	0.253
	Over-sampling	0	**1023**	**0**	28	97.3	0.632	100.0	0.0	0.121
	SMOTE	24	390	633	4	39.4	0.722	38.1	85.7	0.230
	ADASYN	24	377	646	4	38.2	0.739	36.9	85.7	0.231
	Borderline-1	23	785	238	5	76.9	**0.853**	76.7	82.1	0.146
	Borderline-3	0	**1023**	**0**	28	**97.3**	0.591	**100.0**	0.0	**0.026**
Boosting	Unbalanced	4	**1011**	**12**	24	**96.6**	0.661	**98.8**	14.3	**0.203**
	Under-sampling	**22**	692	331	**6**	67.9	**0.748**	67.6	**78.6**	0.249
	Over-sampling	6	963	60	22	92.2	0.639	94.1	21.4	0.213
	SMOTE	14	859	164	14	83.1	0.680	84.0	50.0	0.210
	ADASYN	11	847	176	17	81.6	0.642	82.8	39.3	0.211
	Borderline-1	12	930	93	16	89.6	0.664	90.9	42.9	0.217
	Borderline-3	4	**1011**	**12**	24	**96.6**	0.661	**98.8**	14.3	**0.203**
XGBoost	Unbalanced	2	**1020**	**3**	26	**97.2**	0.791	**99.7**	7.1	**0.026**
	Under-sampling	**21**	728	295	**7**	71.3	**0.794**	71.2	**75.0**	0.209
	Over-sampling	6	993	30	22	95.1	0.742	97.1	21.4	0.042
	SMOTE	13	906	117	15	87.4	0.711	88.6	46.4	0.095
	ADASYN	12	895	128	16	86.3	0.732	87.5	42.9	0.098
	Borderline-1	10	961	62	18	92.4	0.777	93.9	35.7	0.058
	Borderline-3	2	**1020**	**3**	26	**97.2**	0.791	**99.7**	7.1	**0.026**
RF	Unbalanced	0	**1023**	**0**	28	**97.3**	0.774	**100.0**	0.0	**0.025**
	Under-sampling	**23**	723	300	**5**	71.0	**0.823**	70.7	**82.1**	0.195
	Over-sampling	0	1020	3	28	97.0	0.777	99.7	0.0	0.027
	SMOTE	6	982	41	22	94.0	0.781	96.0	21.4	0.055
	ADASYN	7	976	47	21	93.5	0.779	95.4	25.0	0.058
	Borderline-1	5	1008	15	23	96.4	0.786	98.5	17.9	0.036
	Borderline-3	0	**1023**	**0**	28	**97.3**	0.774	**100.0**	0.0	**0.025**

Notes: For all sampling, random_state was set equal to 1996. More notes for this table are on the top of the next page.

Notes of Table 7: The Table shows the seven different models discussed in section 4.2 for the unbalanced training set and after six different sampling methods of section 4.1. The results are bold and underlined showing the best performing sampling technique for each model. For True Positives (TP) and True Negatives (TN) the higher the number the more correctly labelled companies. For False Positives (FP) and False Negatives (FN) the lower the number the fewer companies are missclassified. For Accuracy (AC), Area under the Receiver Operating Characteristic Curve (AUC), Specificity (SP), and Recall (RE) the higher the number the better. Finally for the Brier score the lower the higher the predictive performance.

5.1.2. Does recall improve for a hyperparameter tuned model?

From Table 7 it is evident that simply plugging in the classifiers, even for tried and true features, can generate poor results. It is likely that the SMOTE-based sampling methods require further optimisation to outperform under-sampling. So here I continue with XGBoost to see if GridSearch Cross-Validation and hyperparameter tuning can improve the performance.

Table 8: Classification results for XGBOOST after GridSearch Cross Validation.

Model	Data	TP	TN	FP	FN	AC (%)	AUC	SP (%)	RE (%)	Brier
XGBoost	Unbalanced	2	1017	6	26	**<u>97.0</u>**	0.747	**<u>99.4</u>**	7.1	**<u>0.027</u>**
	Under-sampling	**<u>19</u>**	714	309	**<u>9</u>**	69.7	**<u>0.768</u>**	69.8	**<u>67.9</u>**	0.230
	Over-sampling	2	**<u>1020</u>**	**<u>3</u>**	26	97.2	0.731	99.7	7.1	0.029
	SMOTE	6	987	36	22	94.5	0.755	96.5	21.4	0.046
	ADASYN	11	866	157	17	83.4	0.763	84.7	39.3	0.123
	Borderline-1	7	998	25	21	95.6	0.761	97.6	25.0	0.038
	Borderline-3	2	1017	6	26	**<u>97.0</u>**	0.747	**<u>99.4</u>**	7.1	**<u>0.027</u>**

Notes: GridSearch Cross Validation for the following set of combinations of parameters for XGBoost. Maximum tree depth (max_depth): [3,6,10]. Step size shrinkage (learning_rate) to prevent over-fitting) : [0.01, 0.05, 0.1]. Number of trees (n_estimators): [100, 500, 1000]. Subsample of ratio of columns for constructing a tree (colsample_bytree): [0.3, 0.7]. K-fold equal to five, thus total 270 folds. Random_state was set equal to 1996. The results are bold and underlined showing the best performing sampling technique for each model. For True Positives (TP) and True Negatives (TN) the higher the number the more correctly labelled companies. For False Positives (FP) and False Negatives (FN) the lower the number the fewer companies are missclassified. For Accuracy (AC), Area under the Receiver Operating Characteristic Curve (AUC), Specificity (SP), and Recall (RE) the higher the number the better. Finally for the Brier score the lower the higher the predictive performance.

The results of Table 8 show that for bankruptcy, one cannot simply use SMOTE and expect to get good results. The data tends to over-fit even after tuning with cross-validation. I argue this can be due to two reasons. Firstly, SMOTE based sampling deteriorates because the minority class (default) has high variance, and it is very similar to the majority class (active). Table 3 on page 10 shows that the minimum and maximum values of the 'active' class are almost always larger than the 'default' class. Furthermore, the 25th and 75th percentile values are roughly the same size for the 'active' and 'default' class.

Secondly, bankruptcy prediction can be affected by seasonality and changing trends. SMOTE based algorithms on rule-based algorithms have shown superior performance in fraud detection and medical diagnosis (Fernandez et al., 2018). However, in those setting variables remain constant over time. Whereas the size of leverage of a company, for example, has changed significantly over time, due to changing trends in the market (Platt, 2016). Furthermore, as shown in Figure 1 on page 7 defaults are more frequent during recessions. As argued by Richardson et al. (1998), the accounting-based model fails to control for changing macro conditions. A company with equal financial ratios has a higher likelihood of default in distressed market conditions. However, this is not part of the model I considered, as I only control for accounting data and no macro-variables.

5.2. Simulation 2: The effect of sampling on features

The existing literature shows specific characteristics of sampling methods. Under-sampling is prone to estimation bias because it only takes a subset of the majority class. SMOTE, meanwhile, mitigates the main disadvantage of over-sampling, that is overfitting by creating synthetic points. This raises the question: what is the effect on feature importance? New points are generated on the 'bridges' between existing observations of the minority class. To test the effect of sampling on feature importance, I further analyse the XGBoost classifier using Shapley values. The train and set are the same as in the previous sections. I also analyse Borderline-1 because it ignores noise observations.

5.2.1. Does SMOTE affect feature importance?

Table 9 shows the mean Shap value ($mean|S|$) for the eleven financial ratios. The mean Shap plots for the four sampling techniques are shown in Figure 22 in Appendix F on page 42. From Table 9 several interesting results appear. Leverage (X4) is consistently the most important feature for the model, although the relative importance varies across the sampling methods. Among the top five features, most features are relatively consistent except for 'Operational Margin' and 'Growth in number of Employees' (GE).

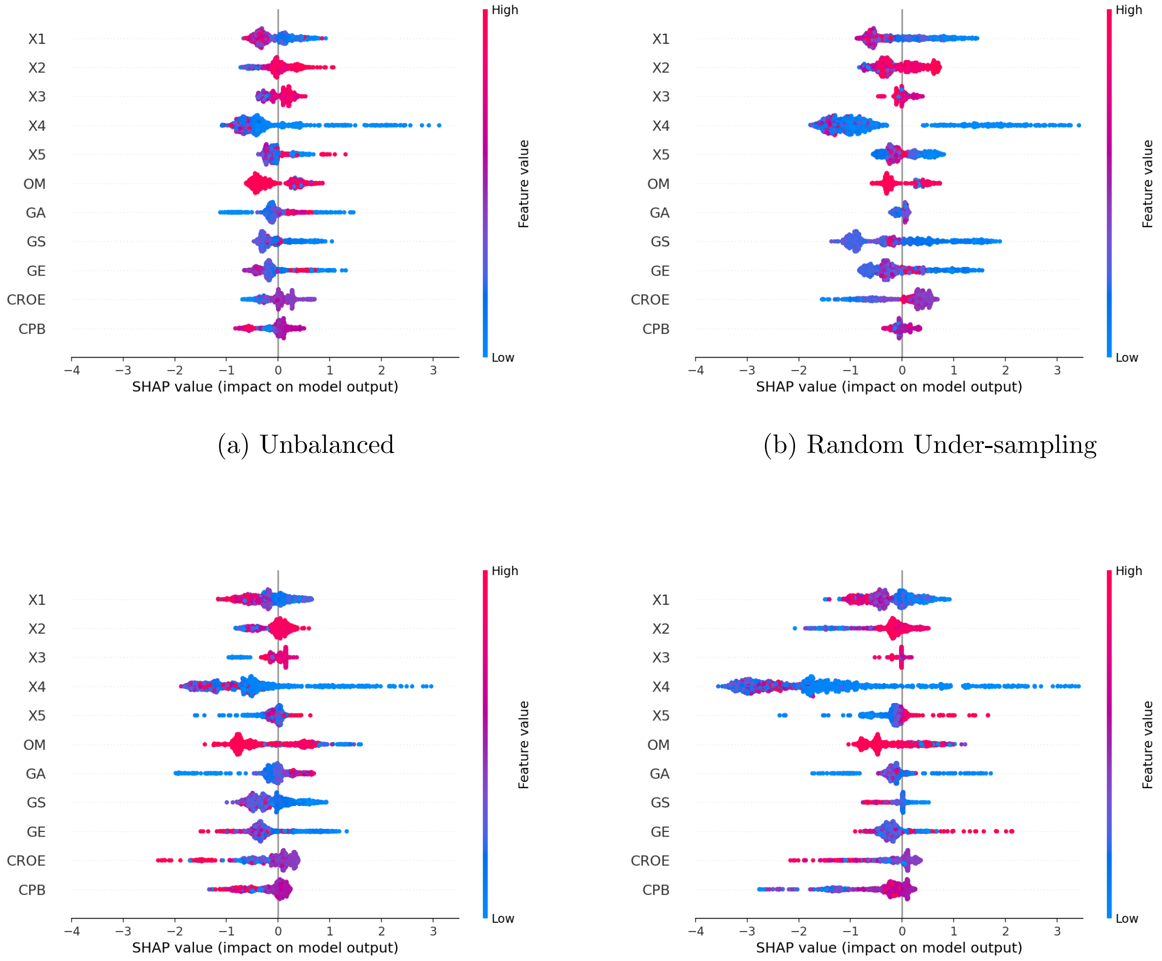
For under-sampling 'Growth in Sales' (GS) is the second most important feature and appears much lower in the ranking of other sampling methods. In the previous section, I concluded for XGBoost that undersampling showed higher recall than the oversampling methods. This suggests that some variables are over-fitted, such as OM, whereas others, such as GS, are under-fitted.

To better understand the effect of sampling, I constructed a beeswarm summary plot in Figure 13. To read this plot, consider GE for SMOTE in figure 13c feature GE. Higher GE values, in red, show lower Shapley values. In other words, the higher the values growth of employees, the less likely a company is to go bankrupt (as the value approaches 0). This third most important feature for SMOTE, GE behaves very differently from the other sampling methods. In under-sampling, in Figure 13b, low values for GE can have both positive and negative Shapley values. For Borderline-1, in Figure 13d, GE drops to the seventh most important feature and has the opposite effect to under-sampling. Higher Growth of Employees can have large positive Shapley values and small negative Shapley values.

Table 9: Absolute mean SHAP values.

Variable	Description	Unbalanced	Under-sampling	SMOTE	Borderline-1
X1	Liquidity	0.29 (3)	0.51 (3)	0.33 (4)	0.41 (4)
X2	Profitability	0.20 (7)	0.35 (6)	0.20 (9)	0.32 (5)
X3	Productivity	0.20 (8)	0.08 (11)	0.15 (10)	0.04 (11)
X4	Leverage	0.58 (1)	1.20 (1)	0.90 (1)	1.95 (1)
X5	Asset turnover	0.17 (11)	0.28 (8)	0.13 (11)	0.22 (9)
OM	Operational Margin	0.37 (2)	**0.31 (7)**	0.62 (2)	0.47 (2)
GA	Growth of Assets	0.21 (6)	0.08 (10)	0.23 (8)	0.25 (8)
GS	Growth in Sales	0.26 (5)	**0.71 (2)**	0.33 (5)	**0.12 (10)**
GE	Growth in number of Employees	0.28 (4)	0.44 (4)	0.41 (3)	**0.28 (7)**
CROE	Change in Return on Equity	0.19 (9)	0.43 (5)	0.32 (6)	0.29 (6)
CPB	Change in Price-to-Book ratio	0.18 (10)	0.10 (9)	0.26 (7)	0.43 (3)

Notes: For all sampling, random_state was set equal to 1996. The number between brackets is the order of feature importance, where number one is the most important and eleven is the least important feature. The underlined values have deviating feature importance to other sampling techniques..



(a) Unbalanced

(b) Random Under-sampling

(c) SMOTE

(d) Borderline-1

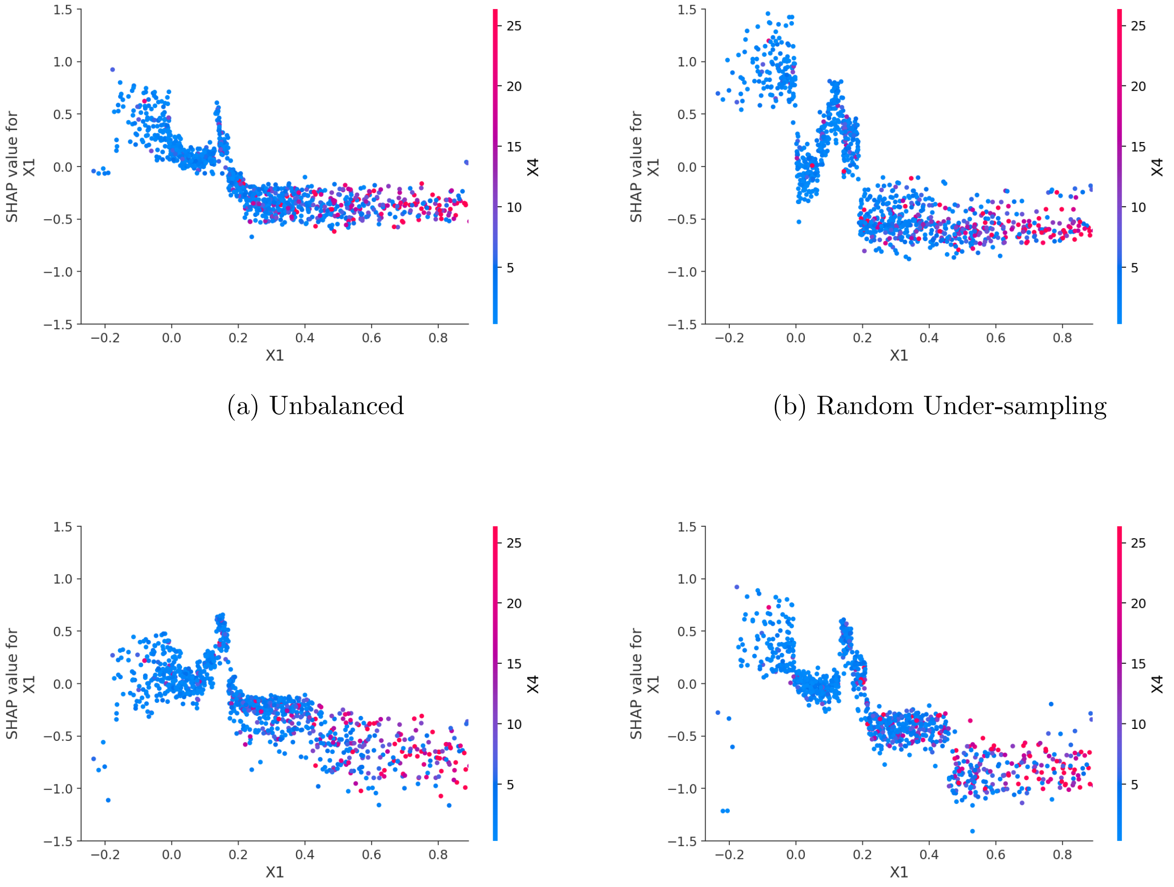
Figure 13: SHAP Beeswarm plots for four different sampling techniques.

Notes: This plot shows how individual data points affect the output of the model. The colours indicate the relative feature value of an observation. The X-axis shows the size of Shapley values, and the Y-axis are the features

5.2.2. SHAP dependence Plots

As a final test, I analyse the effect a single financial ratio has on the predictions made by the model. Thus I only investigate the dependence plot of the ratio Liquidity (X1). The SHAP package allows for deeper analysis by colour plotting the feature value of a different feature. Here I plot the same plot with X4 value colours in Figure 14. I removed the one percentile outliers to improve the plot readability. The percentiles are included in Figure 23 in Appendix G on page 43 shows the dependence plots.

Three characteristics become apparent. Firstly, the sampling methods shift the mean SHAP value of feature contribution. This is also clear from Table 9 and Figure 13, as one could read that the different sampling techniques affected the size of the Shapley values for each individual feature. Moreover, the variance of mean Shapley values between the maximum and minimum is higher for undersampling [+1.5,1.0] versus for SMOTE [+0.6,-1]. This further implicates the feature importance as identical observations have different size of effect on the outcome. Secondly, X1 acts as a 'step' function with threshold around value 0.2. Thus for values higher than 0.2 the Shapley value for X1 decreases roughly 0.5. But under-sampling and borderline-1 show an additional step at 0.0 (going up for under-sampling) and 0.45 (going down for Borderline-1). Thirdly, there is an interaction with X4 (Leverage), such that higher values of X1 have higher values for X4, such that higher liquidity is paired with higher leverage. This is fairly consistent for all sampling methods. I conclude that sampling significantly affects the feature importance in bankruptcy data. The decision of sampling shows different behaviour even for individual features.



(a) Unbalanced

(b) Random Under-sampling

(c) SMOTE

(d) Borderline-1

Figure 14: Dependence plot of X1 (Liquidity) with feature value for X4 (Leverage)

Notes: The X-axis is the true value of an observation, and the Y-axis shows what that value did to the prediction. The colours red and blue are the corresponding Leverage (X4) values for the same observation of Liquidity (X1). The higher the value for X4, the redder the observation becomes.

6. Conclusion

This is the final chapter of the thesis and consists of three sections. The first section syntheses the results and how these relate to the hypotheses. The second section argues the implications the results of the thesis have. The last section discusses the limitations of the thesis and suggests future research to mitigate these.

This thesis analyses the effect of under- and over-sampling techniques in a default data setting. The current state of the literature almost exclusively focuses on identifying better financial ratios and newer machine learning models. However, none directly compare the effect of different sampling techniques. This thesis is a comparative study between the de facto under-sampling method in financial distress prediction and SMOTE based sampling techniques popular in other fields such as fraud detection.

The first hypothesis expects SMOTE-based sampling techniques to outperform under-sampling in one-year-ahead of default prediction. In section 5.1 I compared seven sampling techniques for seven different models. The conclusion for under-sampled models is consistent with the results of Barboza et al. (2017). Ensemble techniques show increased accuracy and AUC, in particular Random Forest. Over-sampling, as expected, shows unacceptably low recall. The number of misclassified defaulted companies is too high for potential users, such as a bank, to trust the results due to the high potential costs. SMOTE sampling, especially Borderline-1, shows the highest results measured in accuracy and AUC. The SMOTE-based sampling methods improve recall over random over-sampling significantly but fail to classify half of the bankrupt companies correctly in Machine Learning models (Boosting, XGBoost, RF). I conclude that under-sampling is the most consistent method as a general model. However, the Borderline-1 for Logistic Regression outperforms all under-sampling methods measured in AUC. This result contradicts Barboza et al. (2017), in that statistical techniques can outperform machine learning models, in particular ensemble methods, by sampling the data differently. Even if one further optimises the data set by using grid search, the results for over-sampling can even deteriorate. Borderline-3 did limit the number of false positives, however showed poor results in detecting 'defaulting' companies. The over-sampling technique performed roughly equal to an unbalanced data set.

The second hypothesis expects that sampling affects features' importance. In section 5.2 I test this assumption with SHapley Additive exPlanations. With this cooperative game theory approach I can analyse the feature contributions of a Machine Learning model. Here I further investigate XGBoost because it is one of the ensemble methods that suffer from SMOTE-based sampling, unlike under-sampling. Among the top four features, the top contributing ones remain fairly consistent except Growth of Sales (GS) for Under-sampling. GS has significantly smaller Shapley values in the other sampling methods, especially for Borderline-1. Thus, both global and local feature analysis shows sampling techniques affect features such that the importance is based on the technique one chooses.

To answer the main research question: Does sampling significantly affect bankruptcy prediction? The results discussed above show that the performance of statistical techniques and machine learning models are dependent on the sampling methods. Furthermore, the feature importance is significantly affected in machine learning such a features' importance depends on the sampling technique. Thus credit suppliers such as bank should be aware of the effect sampling has an the output and use common sense for approving loans.

These results build on the existing evidence that machine learning outperforms statistical techniques *if* one uses under-sampling. The results for Borderline-1 show that the algorithm can significantly boost the performance of statistical techniques (Logistic Regression). Borderline-1, and other SMOTE based techniques, is also a machine learning algorithm because it is built on K-nearest neighbours. So the results are in line with previous works such as Barboza et al. (2017), but such that sampling can significantly affect the performance of classical approaches such as Ohlson (1980).

Moreover, I test the feature importance of default prediction for the first time with Shapley Additive Explanations. This tool allows one to for explain a machine learning models. I analysed the feature importance and feature dependence, but this model is especially helpful for credit suppliers. The package allows for example, a bank to directly show why a model predicts that a certain company defaults or not.

The reliability of this data set is impacted by three factors. Firstly, I constructed the variables Altman (1968) and Carton and Hofer (2006), following Barboza et al. (2017). In particular, the Altman variables (X1 through X5) are widely applied throughout the literature and financial industry. As Beaver (1966) argues, the primary concern is not the validity of the ratios but the accounting behind them. Companies sometimes manipulate the financial data, aware of the key metrics used in finance, to achieve more funding or higher market value. This is known as 'window dressing', and a famous example is the 'cooking of the books' of Enron.

Secondly, the dataset includes both liquidation (chapter 7) and bankruptcy (chapter 11). In both cases, the company is unable to repay its debt, but sometimes board members and executives choose to initiate bankruptcies to restructure a company (Balcaen and Ooghe, 2006). This suggest there is a human element, not represented in the accounting data. Furthermore, if companies use the ratios to detect 'danger' of defaulting, companies can act to mitigate potential financial distress (Beaver, 1966). Therefore the observations can be misleading because active companies could have gone bankrupt if no action was taken.

Thirdly, I only included accounting data as is standard in the financial distress literature. Therefore I omitted external factors, including macro data. These include the seasonality of industries, the cyclicality of the markets such as GDP growth or decline, ESG ratings and more. As argued earlier, accounting-based data fails to control for macro conditions (Richardson et al., 1998). Therefore future research can extend the analysis by including relevant non-accounting features.

As a final note on future research, I considered using earning estimates. Due to the sparsity of the data from the WRDS data set, this was neglected. Other data sets would be interesting but are time-consuming to match and were therefore outside the scope of the thesis. Earning estimates can be better predictors because earnings have no publication lag. This means the estimated earnings could have higher predictive power than the earnings of yesteryear. Estimated earnings, namely, look into the future instead of looking back. This includes analyst predictions of the economy, such as GDP growth rates or the likelihood of a recession.

Appendices

A. Abbreviations

Abbreviation	Definition
AC	Accuracy
ADASYN	Adaptive Synthetic, a SMOTE-based extension
ANN	Artificial Neural Networks
AUC	Area Under ROC Curve
CRSP	Center for Research in Security Prices
FP	False Positives
FN	False Negatives
MDA	Multivariate Discriminant Analysis
QDA	Quadratic Multivariate Discriminant Analysis
RE	Recall, Sensitivity
RF	Random Forest
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Oversampling Technique
SEC	Securities and Exchange Commission
SHAP	Shapley Additive Explanations
SVM	Support Vector Machine
TN	True Negatives
TP	True Positives
XGBoost	Extreme Gradient Boosting
WRDS	Wharton Research Data Services

Table 10: List of Abbreviations

B. Variable Construction

Table 11: CRSP / Compustat Merged input variables for constructing variables.

Variable	Compustat Code	Description
Net Working Capital	WCAP	Working Capital (Balance Sheet)
Total Assets	AT	Assets - Total
Retained Earnings	RE	Retained Earnings
Earnings before interest and taxes	EBIT	Earnings Before Interest and Taxes
Market Value of share	PRCC_C	Price Close - Annual - Calendar
Number of shares	CSHO	Common Shares Outstanding
Total Debt	LT	Liabilities - Total
Sales	SALE	Sales/Turnover (Net)
Employees	EMP	Employees
Net Income	NI	Net Income (Loss)
Book Value per share	BKVLPS	Book Value Per Share

C. ADASYN derivation

This section describes the mathematical derivation of the SMOTE-based extension Adaptive Synthetic (ADASYN) following the original paper He et al. (2008). First, ADASYN determines the number synthetic observations to generate (G):

$$G = (|nnum - pnum|) * \beta \tag{8}$$

Where $nnum$ is the number of majority observations and $pnum$ is the number of minority observations. $\beta \in [0, 1]$ is a parameter to set the balance level after sampling using ADASYN. In this thesis, β is set to 0.5. Next for each observation x_i, ADASYN finds the number of K-nearest neighbours, like in SMOTE, and the number of majority observations Δ_i. Then it calculates the impurity ration imp_i:

$$imp_i = \frac{\Delta_i}{K} \tag{9}$$

Then all impurity ratios are normalised according to $i\hat{m}p_i$:

$$i\hat{m}p_i = \frac{r_i}{\sum_{k=1}^{pnum} imp_j} \tag{10}$$

Then the total number of synthetic points generated (g_i) for observation x_i is determined as:

$$g_i = i\hat{m}p_i * G \tag{11}$$

Finally, the algorithm loops for g_i times for each observation x_i to create new synthetic variables using the following formula:

$$r_j = x_i + rand(0, 1) \cdot (x_i - x_{ij}) \tag{12}$$

Where r_j is a synthetic variables, and rand(0,1) is a random number generators between 0 and 1 using a uniform distribution. x_{ij} is one of k-neighbours of x_i and j is between 1 and K. Therefore the main difference between SMOTE and ADASYN is that $i\hat{m}p_i$ and g_i determine how many observations are created for each minority observation.

D. Logistic Regression

Logistic Regression to estimate the likelihood of a company to default in the next year. The target variable is binary, and a relationship is estimated with the following logistic regression:

$$PD = \text{PR}[\text{Default next year} = 1|X] = F(z) \tag{13}$$

Where PD is the probability to default and $F(z)$ logistic cumulative distribution function evaluated at z, which is expressed as:

$$z = \beta_0 + \beta_1 x_1 + ... + \beta_k x^k \tag{14}$$

Here x are the explanatory variables and β are the coefficients.

$$F(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}} \tag{15}$$

E. Simulation 1: ROC Plots

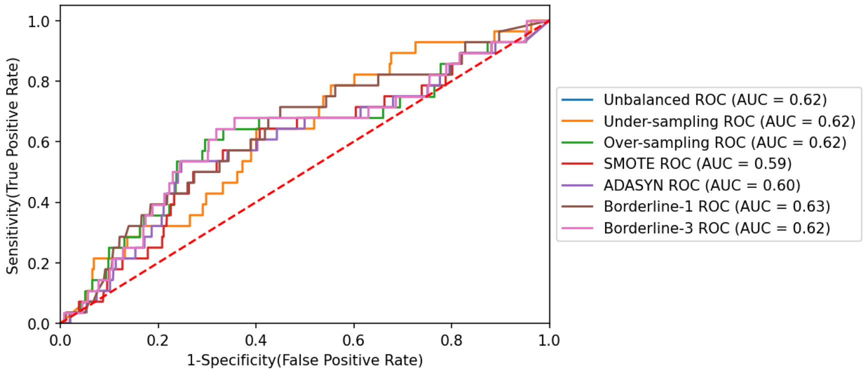


Figure 15: QDA ROC Curve for sampling techniques

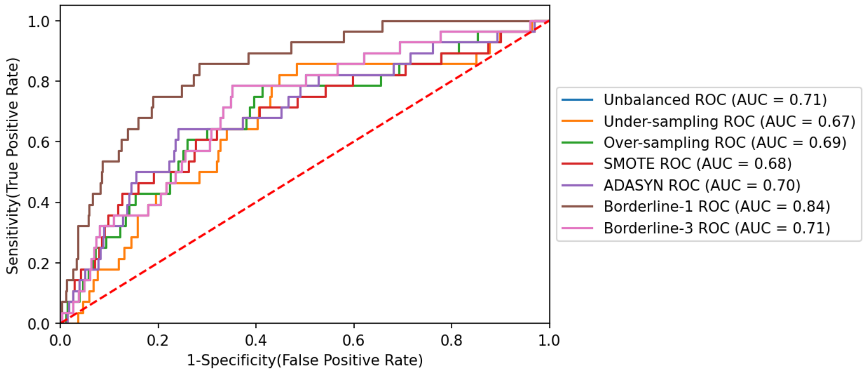


Figure 16: Logistic Regression ROC Curve for sampling techniques

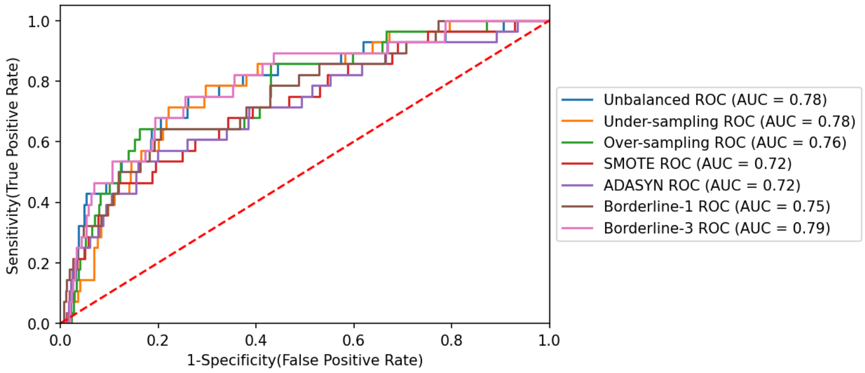


Figure 17: Artificial Neural Network ROC Curve for sampling techniques

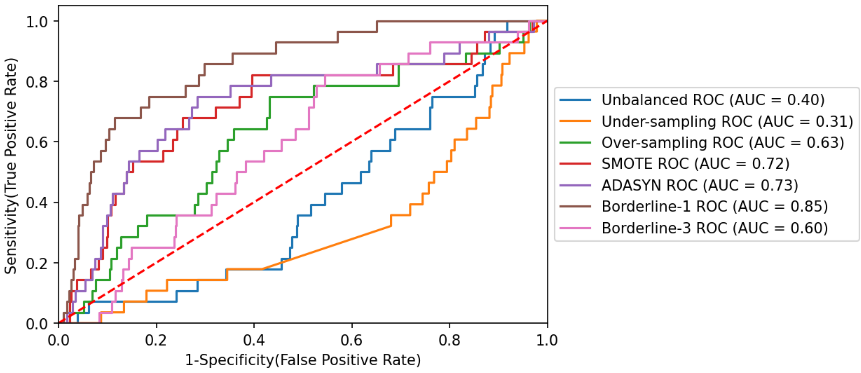


Figure 18: Linear SVM ROC Curve for sampling techniques

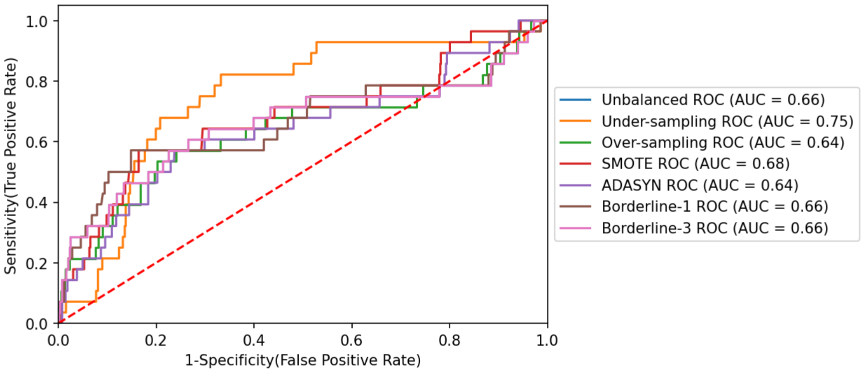


Figure 19: Boosting ROC Curve for sampling techniques

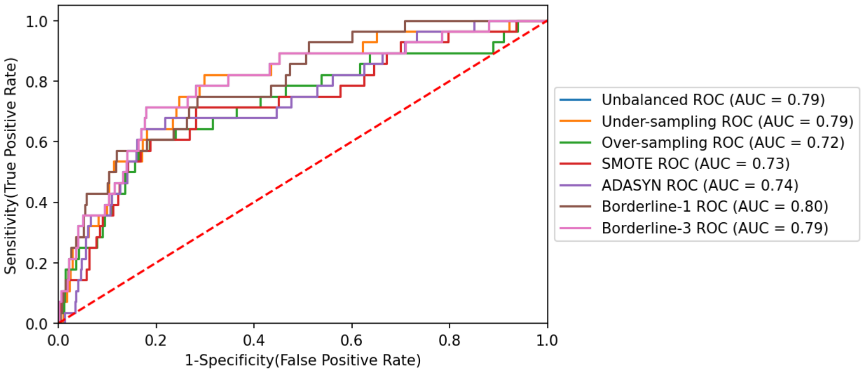


Figure 20: XGBoost ROC Curve for sampling techniques

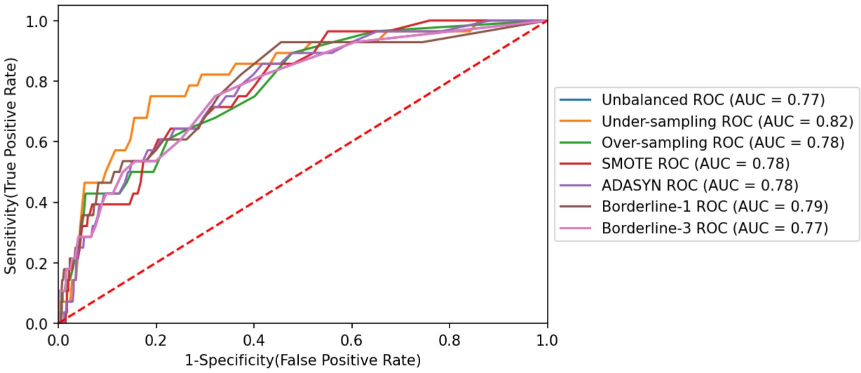
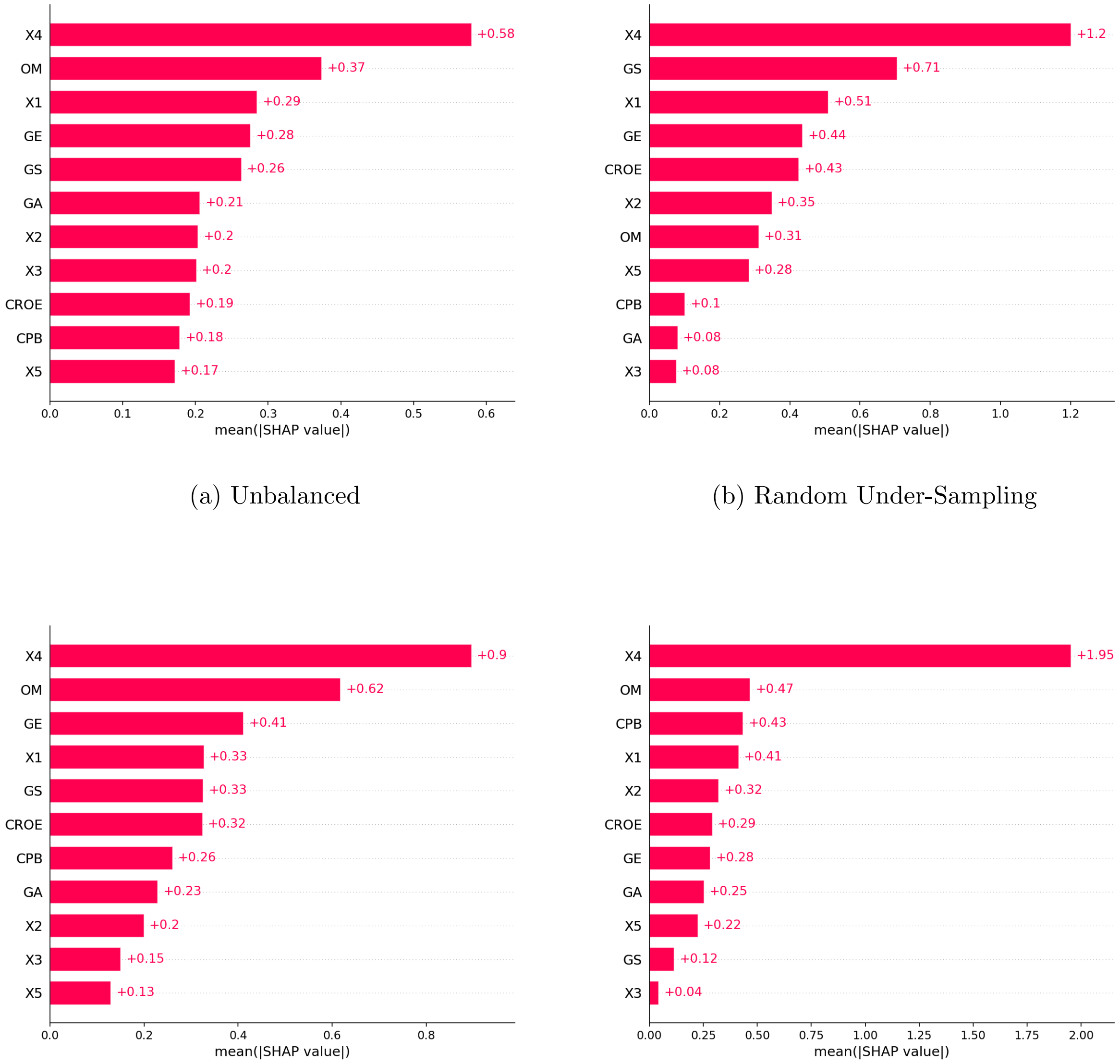


Figure 21: Random Forest ROC Curve for sampling techniques

F. Simulation 2: Feature Importance



(a) Unbalanced

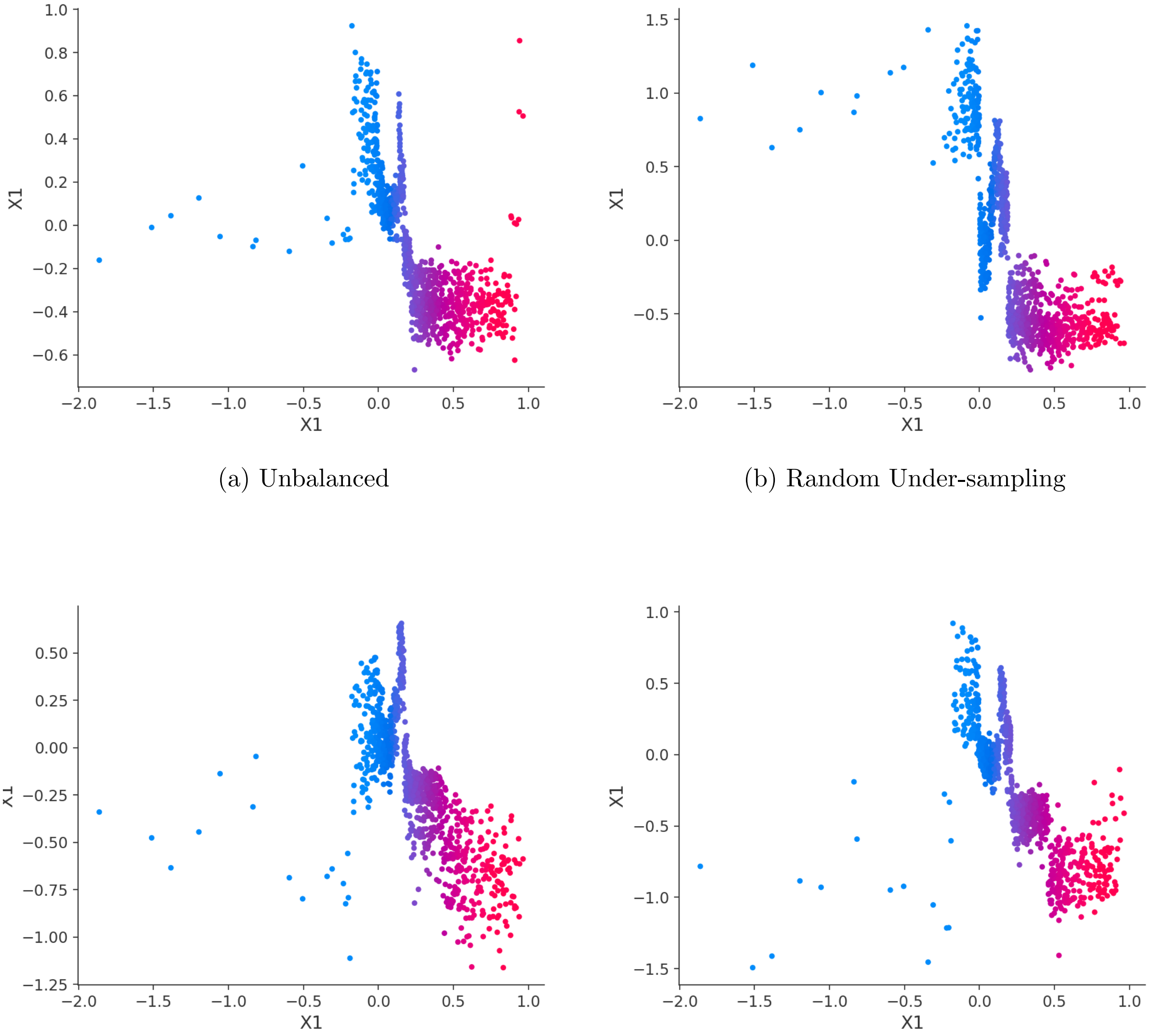
(b) Random Under-Sampling

(c) SMOTE

(d) Borderline-1

Figure 22: Illustration of various images

G. Simulation 2: Dependence Plots



(a) Unbalanced

(b) Random Under-sampling

(c) SMOTE

(d) Borderline-1

Figure 23: Dependence plot of X1 (Liquidity)

References

Altman, E. (1968). Financial Ratios, Discriminant Analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, (23):589–609.

Altman, E. (2019). 50 Years of Altman Z-Score.

Altman, E. I., Haldeman, R. G., and Narayanan, P. (1977). ZETA analysis A new model to identify bankruptcy risk of corporations. *Journal of Banking & Finance*, 1(1):29–54.

Apley, D. and Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82.

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

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

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

Begley, J., Ming, J., and Watts, S. (1996). Bankruptcy Classification Errors in the 1980s: An Empirical Analysis of Altman's and Ohlson's Models. *Review of Accounting Studies*, 1:267–284.

Bracke, P., Datta, A., Jung, C., and Sen, S. (2019). Machine learning explainability in finance: an application to default risk analysis.

Brier, G. W. et al. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

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

Carton, R. and Hofer, C. (2006). Measuring organizational performance: Metrics for entrepreneurship and strategic management research. *Measuring Organizational Performance: Metrics for Entrepreneurship and Strategic Management Research.*

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Fernandez, A., Garcia, S., Herrera, F., and Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, 61:863–905.

Freund, Y. and Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

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

Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In Huang, D.-S., Zhang, X.-P., and Huang, G.-B., editors, *Advances in Intelligent Computing*, Lecture Notes in Computer Science, pages 878–887, Berlin, Heidelberg. Springer.

Hastie, T., Tibshirani, R., and Friedman, J. (2017). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer-Verlag New York Inc.

He, H., Bai, Y., Garcia, E., and Li, S. (2008). ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. IEEE.

Heo, J. and Yang, J. Y. (2014). AdaBoost based bankruptcy forecasting of Korean construction companies. *Applied Soft Computing*, 24:494–499.

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

Kim, M.-J., Kang, D.-K., and Kim, H. B. (2015). Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction. *Expert Systems with Applications*, 42(3):1074–1082.

Le, T., Lee, M., Park, J., and Baik, S. (2018). Oversampling Techniques for Bankruptcy Prediction: Novel Features from a Transaction Dataset. *Symmetry*, 10:79.

Ligang, Z., Lai, K. K., and Yen, J. (2014). Bankruptcy prediction using SVM models with a new approach to combine features selection and parameter optimisation. *International Journal of Systems Science*, 45:241–253.

Lundberg, S. and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Neural Information Processing Systems*.

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, 18(1):109–131.

Platt, E. (2016). Triple A quality fades as companies embrace debt. *Financial Times*.

Richardson, F. M., Kane, G. D., and Lobingier, P. (1998). The Impact of Recession on the Prediction of Corporate Failure. *Journal of Business Finance & Accounting*, 25(1-2):167–186.

SEC (2009). Bankruptcy: What Happens When Public Companies Go Bankrupt.

Shin, K.-S. and Lee, Y.-J. (2002). A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications*, 23(3):321–328.

Sun, J., Li, H., Huang, Q.-H., and He, K.-Y. (2014). Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches. *Knowledge-Based Systems*, 57:41–56.

Veganzones, D. and Séverin, E. (2018). An investigation of bankruptcy prediction in imbalanced datasets. *Decision Support Systems*, 112:111–124.

Wang, G., Ma, J., and Yang, S. (2014). An improved boosting based on feature selection for corporate bankruptcy prediction. *Expert Systems with Applications*, 41(5):2353–2361.

Wilson, R. L. and Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems*, 11(5):545–557.