Ca' Foscari
University
of Venice

Master's Degree programme

in Finance

Final Thesis

Machine Learning Models for Bankruptcy Prediction

Supervisor
Ch. Prof. Emanuele Aliverti

Graduand
Leonardo Maritan
Matriculation Number 883654

Academic Year
2022 / 2023

Abstract

After the 2007/2008 financial crisis, bankruptcy prediction has become one of the main and priority assessment topics in credit risk analysis for most financial institutions and intermediaries. In this work, a dataset containing 8262 different companies listed on the American stock market in the period between 1999 and 2018 will be analyzed. This historical series will be analyzed using different Machine Learning models, with the aim of understanding which ones are able to predict at best business failure in different time frames. Before using the models, however, it was necessary to address the aspect relating to the imbalance of the classes due to the rarity of bankruptcy events in the real economy. Finally, in the light of the results obtained, the most interesting results will then be highlighted, the positive aspects and limitations of the various models considered for the analysis will be deepened, and suggestions will be proposed to integrate the work carried out in future studies.

Contents

Introduction

Bankruptcy for companies is one of the most important issues in financial management and investing. Through bankruptcy procedure the firm ceases to exist and it happens when an organisation is unable to honour its financial obligations or make payments to its creditors; there are many causes of bankruptcy, and they are usually related to poor management, marketing and financial capacity.

During its life any business faces different kind of risks, difficulties and changes that should be considered not only by the managers of the firm, which have the main role to pay attention to the liquidity and solvency level of the company, but by all the stakeholders, such as lenders that have to decide whether to give credit to an entity, or investors, who should decide if it is worth investing on a company and assess the risk of such investment.

In the bankruptcy prediction framework, the problem of detecting financial distress in business which could lead to bankruptcy, plays a key role in the assessment of the soundness of companies and as a tool to foresee situations of possible financial distress.

The first literature concerning bankruptcy prediction dates back to the beginning of 1930; the prediction of distressed firm was based on ratio analysis and, till the half of 1960, studies and researches were mainly focused on single factors or on the analysis of specific factors. Altman, in 1968, published the first multivariate study (Altman, 1968). His model, which will be explore more in-depth later, was developed using five common business ratios and it turned out to have excellent results in the prediction of the probability that a firm will go into bankruptcy within one year.

Altman's model represented a crucial turning point in the study of bankruptcy prediction: the number and complexity of the models increased exponentially in the subsequent years; until the development of machine learning models which will be the core of this work.

Therefore, the focus of this thesis will be to analyze and describe the main features of some of these models with the aim of then applying them to "American Bankruptcy Dataset " in which data are collected for 8262 companies in the period between 1999

and 2018. The final purpose of this paper will therefore be to understand what are the most relevant variables to predict bankruptcy and which models are able to better predict this phenomenon.

The structure of the thesis will be the sequent:

- Chapter 1. It will be introduced the definition of bankruptcy and the main reasons why companies face financial distress according to the literature. Moreover, will be described the different procedures to fill for bankruptcy in the American framework.
- Chapter 2. In this chapter will be described the characteristics, the positive aspects and the disadvantages of the Altman Z-score model and the Ohlson O-score which are considered the pioneer techniques for bankruptcy prediction and the basement for future and more advantages models.
- Chapter 3. Machine learning models will be described in this section. Specifically, will be described the main advantages and qualities in the use of lasso regression, classification tree, random forest, bagging, Support Vector Machine and CatBoost and XGBoost. These are the model that will be then used for the empirical analysis.
- Chapter 4. In this chapter will be described the procedures of data manipulation in order to obtain a proper dataset to be used to fit the model. Moreover, will be introduced the performance measures that will be used to understand and compare the performances of the models and establish the best machine learning techniques.
- Chapter 5. Machine learning models described in Chapter 3 will be used and will be provided a description of the main results, the main features of each model and of the best model results.
- Chapter 6. Conclusion and further works.

1 Introduction to Bankruptcy framework

Before going in depth into the description of the machine learning models, of the dataset used and of the main results obtained, in this chapter the definition of bankruptcy will be introduced, some studies and literature about such phenomenon in order to better understand its meaning. To conclude will be described the American legislation and how a company in financial distress can actually fill for bankruptcy.

1.1 What is bankruptcy?

The legal procedure known as bankruptcy involves selling off a debtor's assets to pay creditors and release the debtor from further responsibility. The goals of bankruptcy are to give the debtor a fresh start free from previous obligations and to fairly resolve creditor claims by allocating an equitable portion of the debtor's remaining assets to them.

The bankruptcy process can be initiated by the debtor or by a petition filed by the debtor's creditors, which are called voluntary and involuntary bankruptcies, respectively. Under the U.S federal bankruptcy code, Chapter 7 addresses the complete liquidation of a debtor's assets, while Chapter 11 deals with the reorganization of the debtor business.

For many stakeholders involved with the company, determining whether a business will collapse and predicting potential financial difficulty is a crucial topic:

- Current creditors must consider the possibility that they can lose a part of the money borrowed to the company. The main distinction is between secured and unsecured creditors. Secured creditors are creditors that hold a lien on its debtor’s property, whether that property is real property or personal property. The lien gives the secured creditor an interest in its debtor’s property, entitling it to sell the assets to repay the loan in the event of failure. The secured creditor’s lien can be voluntary, like with a bank or other asset-based lender, or involuntary, like a tax lien.

While, credit card issuers, suppliers, and some cash advance businesses (although this is changing) are examples of unsecured creditors who do not have a claim on the debtor's property to guarantee payment of the debt in the event of a default.

So, priority is given to the secured creditor in the collection of debt from the assets secured by its lien. The easiest way for an unsecured creditor to get paid back by their debtor is by voluntary repayment, as they are not given such protection. Otherwise, short of bankruptcy proceedings, the unsecured creditor must sue and win a judgment to get repaid on a defaulted debt.

• Potential lenders must decide whether to invest, approve the application for a business loan, and then lend money to the borrowers. Additionally, the lenders should choose whether to require collateral, what type of collateral to require, how much money to require, and the right interest rate based on the potential risk of financial distress.

• The shareholders, because they are the ones that are going to lose money first in the case of company failure and they are those that can d have the power to pick new managers if the current ones are performing poorly. The bankruptcy procedure has an impact on shareholders' interests since a company only files for bankruptcy when it has no, or very little equity left and is not making enough money to pay dividends. In the majority of cases, the value of the shareholder investments is completely destroyed; if not, when a firm goes through a reorganization, a significant portion of its debt is typically converted to equity. This newly created equity may or may not have voting rights. It does, however, reduce the value of the present equity because the old shares of a company nearing bankruptcy lose all of their value. A new class of equity shares is introduced in their place. These shares are typically given to creditors who have agreed to accept equity in place of debt. Sometimes, this new equity is also issued to existing shareholders and the number of shares, or the value of shares issued may be reduced. From the perspective of a shareholder, this is a serious loss; nevertheless, the most crucial truth is that the company's stockholders also lose management rights. The management appointed by the shareholders is in

total control of the company up until it declares bankruptcy. However, following the filing of the bankruptcy petition, the responsibility of the management is given to a trust. This trust is responsible for the well-being and interest of the creditors.

• The managers/entrepreneurs as they are capable of coming up with a plan to recover from a challenging circumstance. In fact, a financial distress situation forces management to adopt a series of measures aimed at enhancing the performance of the company, according to Jensen (1989) and Whitaker (1999). It stands to reason that it is too late when a company approaches bankruptcy. The managers should be alert for warning signals of a potential bankruptcy before, just when the company first experiences financial distress situation. These last three words represent "[…] a condition in which a company or individual cannot generate revenue or income because it is unable to meet or cannot pay its financial obligations. This is generally due to high fixed costs, illiquid assets, or revenues sensitive to economic downturns." (Kenton, 2019). From an accounting perspective, it is the company's inability to pay its creditors (such as suppliers or lenders) with its operating results. If this occurs for just a single year, the business will just have a minor crisis; however, if it continues, it may result in bankruptcy.

1.2 Business Life Cycle Theory

Financial crisis, default, and bankruptcy are crucial phases in a company's lifecycle (Wruck, 1990). The four stages of the corporate life cycle are typically: birth, growth, maturity, and decline.

According to the *life cycle theory*, a firm's strategies, access to resources, and expanding ability change over its lifetime (Anthony and Ramesh 1992). The situation, organizational strategy, structure, and decision-making methods alter significantly at each stage and the decisions about restructuring are mostly influenced by specific lifecycle characteristics.

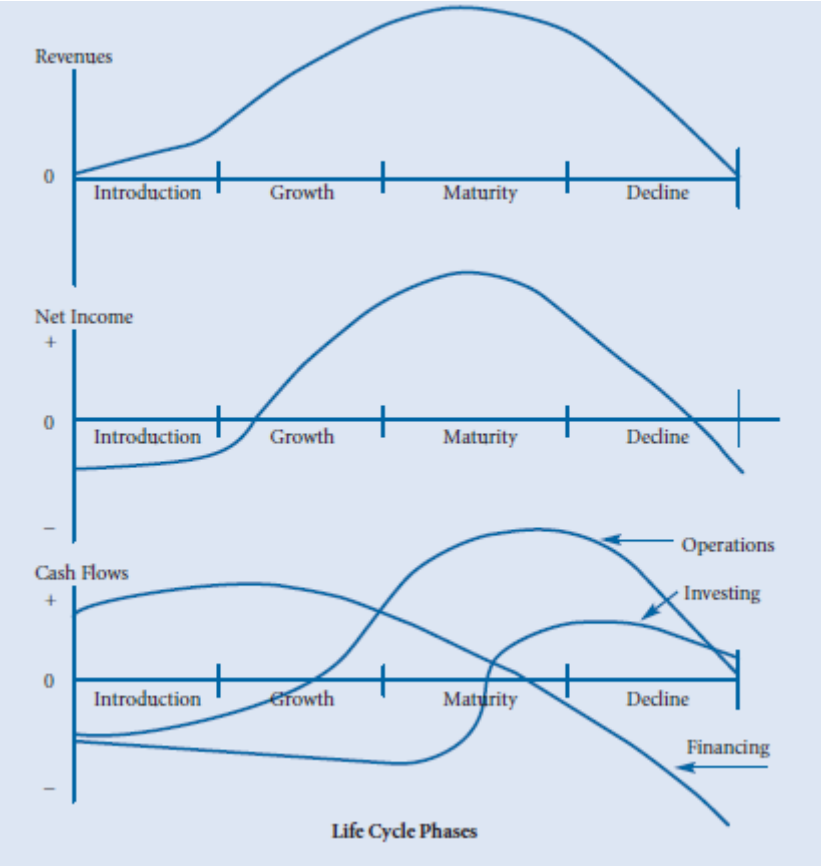

***Figure 1.1**: Business Life Cycles Phases*

Firms are often small, informal, and simple in their early phases, with concentrated power and a main focus on investing on innovation (Adizes 2004; Miller and Friesen 1984; Pashley and Philippatos 1990). The book-to-market ratio and firm-specific risk are higher due to the high level of uncertainty regarding future growth, which makes logical sense (Pastor and Veronesi 2003).

Corporate financial distress at its beginning is typically linked to inadequate liquidity or concurrent cash flow issues (Hasan et al. 2015). Firms may experience rapid expansion at this period, and because it seems necessary to increase capital, corporate financial crisis is typically associated with excessive financial leverage (Agostini 2018). Compared to the earlier stages, organizations typically place less emphasis on innovation and risky strategy during the mature period. They do want to stabilize their company's position in the market.

The last stage of the life cycle also is the phase of decline, during which the business has financial difficulty and suffers a significant drop in performance; if the root causes are not addressed, the deterioration will eventually lead to crises and failures.

According to this theory, the types of corrective actions and restructuring plans used by businesses experiencing corporate financial difficulties can also vary and may be influenced by lifecycle physiognomies (Koh et al. 2015). Each company does, in fact, follow some recovery tactics, such as reducing on investments or dividends, but others are directly connected to the stage at which the troubled company found itself. For instance, reducing the number of staff is a popular restructuring strategy in the early stages, whereas asset reorganization is typically used in a more mature stage.

Sudarsanam and Lai (2001) proposed four main categories of restructuring: management, operational, asset, and financial; managerial restructuring does not directly involve cash, but it does require replacing top management and/or the CEO.

By selling fixed assets, cutting back on expenditures, spending (Capital, R&D), and costs (COGS), operational restructuring decisions seek to maximize output while minimizing costs (Koh et al. 2015). (Lasfer & Remer 2010).

In contrast, asset restructuring entails selling off assets through divestments, spin-offs, and equity carve-outs, merging with another company, and acquiring assets in order to reduce unrelated diversity and increase concentration on core competencies (Shleifer and Vishny 1992); but it is considered to be an extreme solution as it typically urgently requires an increase in the amount of cash needed (Robbins and Pearce 1993). Furthermore, we refer to financial restructuring when a company needs to alter its dividend policy or capital structure. This includes employing strategies like issuing new

securities, exchanging debt for equity, and reducing or neglecting dividends (Koh et al. 2015).

It is evident that a managerial restructuring will not take place in the early stages of a company since power is concentrated and it is more likely that the managers are also the owners. However, as the company expands, the structure becomes more complex and managers are more likely to be replaced in case of poor performances.

1.3 Financial statements

Financial statements are the most significant source of information on the operations and performance of the company and eventually spot and analyse possible causes of financial distress. These documents are updated regularly and reflect the firm's financial situation in several sectors because they are reported on a regular basis:

- The Balance Sheet displays the assets, liabilities, and shareholders' equity of the company.
- The Cash Flow Statement shows all cash inflows and outflows for the reporting period.
- The Income Statement highlights whether the company has made a profit or a loss during the reporting period.
- the Statement of Retained Earnings represents the total revenue earned by the company after dividends have been paid. On each balance sheet, it also shows the variation in the retained earnings account between the beginning and ending periods.

The most important statement to be checked is the Cash Flow Statement: indeed, when cash outflows constantly exceed cash inflows, the company is likely running out of cash and may not have enough to meet all of its obligations. The company may be in severe trouble if it cannot raise some money from equity investors or lenders.

Negative cash flows from operations, in particular, indicate that the company needs external financing since it cannot create enough cash to support and expand its activities. In connection with this, loan interest payments can strain cash flows,

especially for distressed companies that must pay higher interest rates to compensate for their elevated default risk.

1.4 Main Causes of Failure in literature

As has already indicated, the majority of the principal bankruptcy literature has focused on insolvency, or the inability to pay debt (Piesse, Lee, Kuo and Lin 2006).

According to Rees (1990), these are a few of the most prevalent reasons for insolvency:

- Low and declining real profitability.
- Unfitting diversification: moving into unfamiliar businesses or failing to move away from deteriorating ones.
- Import penetration into the firm's domestic markets.
- Worsening financial structures.
- Complications controlling new or geographically isolated processes.
- Over-trading in relation to the investment base.
- Insufficient financial control over contracts.
- Scarce control over working capital.
- Failure to remove actual or potential unprofitable activities.
- Adverse changes in contractual arrangements.

However, this classification solely views the meaning of business failure as equivalent with insolvency and does not take into account any causes that are not financial.

It is the possible to separate internal reasons from external causes, or micro-level and macro-level variables, in order to arrange the factors that influence company financial distress:

- Micro-level determinants and internal causes; At the micro level, factors including the company's size, maturity, sector, and flexibility must be taken into account. The age of the company has a considerable impact on the likelihood of bankruptcy, according to Altman (1971). Considering the corporate life cycle described in the preceding sentence, companies are more likely to fail in the initial stage and in the decline state, although the likelihood of bankruptcy is minimal in the intermediate stages (growth and maturity). The likelihood of a

new, tiny firm failing within its first three years of operation is higher than it is for an established, larger company.

An evidence to support this concept is provided by a study conducted by Thornhill and Amit (2003): it demonstrates that younger companies are riskier and more likely to fail due to a lack of managerial expertise and financial management skills, while older companies may struggle to adapt to environmental changes. Due to the significant initial investments and resources needed, young businesses may not have the money to satisfy financial obligations, or they may not have the industry-specific expertise and competencies to gain a competitive advantage. These businesses might not be greatly impacted by external forces. On the other hand, expanding businesses are frequently at risk of failing because they lack the adaptability to respond to and evolve with the environment. Last but not least, failures for older, more established businesses might result from altered competitive environments coupled with a lack of dedication and motivation or from an overly risky strategy (Ooghe and De Prijcker, 2008).

• Macroeconomic and external causes; Despite the fact that most studies link managerial bad decisions to corporate bankruptcy which depends on a manager's qualities, skills, motivation, and personal characteristics, there may be many external factors that influence a company's policy and performance from both the general and immediate environment.

The nature of the industry (chronically ill industries like agriculture and textiles), deregulation of some important industries (airlines, energy, financial services), which increases the number of entering and leaving firms in the market landscape, high real interest rates in some periods, international competition, overcapacity within an industry, and a high likelihood of new business development are a few of these non-managerial reasons (Altman & Hotchkiss 2010).

One of the pioneering researchers to examine the impact of macroeconomic factors on corporate failure was Altman (1971). He found that the probability of default increases with a tight monetary policy, when investor’s expectations

about economic conditions are negative and finally, when the state of the economy worsens.

Macroeconomic factors, according to Chordia and Shivakumar (2005) and Bonsall et al. (2013), explain around half of the variation in firm earnings and earnings variances: A nation's economic health affects the business environment through changes in its inflation rate, interest rate, employment rate, loan availability, and monetary policy (Liou and Smith, 2007). As a matter of fact, by including country risk factors, bankruptcy prediction models' predictive ability can improve.

1.5 US bankruptcy framework

In the United Stated, bankruptcy is ruled by federal law and the primary source of bankruptcy law is the bankruptcy code. It contains a set of "chapters" that allow for the development of different bankruptcy cases and procedures. Federal courts oversee all bankruptcy cases in the United States, which total six, and handle with liquidation (Chapter 7), Municipality bankruptcy (Chapter 9), Reorganization (Chapter 11), Family Farmer Bankruptcy or Family Fisherman Bankruptcy (Chapter12), individual debt adjustment (Chapter 13), Ancillary and other cross-border Cases (Chapter 15).

Finding a balance between the competing interests of the debtors, creditors, and all other stakeholders is the main common goal of each of those chapters. Despite this, each Chapter has its own procedure. Some chapters seek to liquidate the debtor, while others try to reorganize in order to enable it to continue operating or to reduce the debtor's obligations.

The two primary objectives of American bankruptcy law are to give debtors a "fresh beginning" from their financial struggles and to relieve them of some obligations that they are unable to fulfil. On the other hand, by maximizing the creditor return, the Bankruptcy Code seeks to protect the interests of creditors and other stakeholders.

The focus of this work will be mainly on corporate bankruptcy and therefore the next focus will be on chapters 7 and 11 of the code.

1.6 Filing for Bankruptcy

The bankruptcy procedure starts with the debtor filing a document known as “petition”; most of the debtors have to file also schedules consisting of debtor’s assets, liabilities, current income and expenditures, statements of their financial affairs and other required documents.

These files are required by bankruptcy court in order to have the “whole picture” about the debtor, its financial health and to ensure that there is adequate information available to the debtor’s creditors with the object to facilitate the fair and efficient distribution of the debtor’s income or assets.

The debtor immediately enjoys several benefits after filing for bankruptcy; In fact, the filing of a bankruptcy petition stays the start or continuation of all non-bankruptcy judicial proceedings against the debtor and prevents creditors from taking any action to collect, assess, or recover a claim against the debtor that arose prior to the case's commencement.

These protections are referred to as "automatic stays" since judicial action is not always required to oversee the operations. The primary goal of this process is to safeguard the debtor's estate from any potential disorder brought on by all the conflicting litigation. This process starts immediately after the filing of a bankruptcy petition and its effects stand until the bankruptcy court closes the case, dismisses the case, or grants the debtor a discharge, whichever comes first.

Therefore, to protect the creditor, the stay forbids creditors from acting only in their own self-interest collecting money from a debtor damaging of other creditors.

Substantially in the Bankruptcy Code there are three main ways of declaring bankruptcy: liquidation, reorganization and adjustments of debts. As previously anticipated, when filing a bankruptcy petition, the debtor must choose one of the methods and files under one of the six chapters that will be briefly described below:

-Chapter 7: Liquidation. Liquidation proceeding is the most common form of bankruptcy and through the years it has been applied to half million debtors; it is mainly used by individuals or small businesses. In Chapter 7, firms eliminate most debt through the liquidation of assets. The debtor's assets must be taken control of by a trustee, who

must then sell them and distribute the money to the debtor's creditors. The trustee must be appointed by the court to oversee the case.

The main distinction is between secured creditors, who have a legal right against certain properties known as collateral and are typically entitled to be paid in full before any unsecured creditors, who have no rights on certain properties of the debtor. The Bankruptcy Code establishes a priority framework of expenses and claims that should be paid before others. All non-exempt assets (properties required to maintain minimal standards of life are excluded) are liquidated in order to satisfy debts, starting with unsecured priority debts, moving on to secured debts, and concluding with regular unsecured debts.

-Chapter 11: Reorganization. Differently from Chapter 7, Chapter 11 aims to restructure the debt of the debtor structure to allow to continue operations. Of all the bankruptcy proceedings, Chapter 11 is the most complicated and frequently the most expensive; In this case, the court assists a business in reorganizing its debts and obligations and permits the company's decisions under restructuring, including the sale of assets and inventory, the beginning or termination of a rental agreement, and the stopping of the expansion of a business.
The debtor, often known as the "debtor in possession," is required to present a reorganization plan, which may involve the downsize of the business to cut costs or selling off assets to pay off creditors.

-Chapter 9: Municipality cases. Cities, townships, school districts, etc. that are in financial difficulties can be protected from creditors through this process. It develops a strategy to settle the municipal debt with its creditors.

-Chapter 12: Family farmer/family fisherman. It is intended for "family farmers" and "family fishermen" who are struggling financially. The debtor devises a strategy to repay creditors over a three- to five-year period.

-Chapter 13: Consumer cases. People with regular incomes can restructure their debt through Chapter 13 and pay back some or all their creditors. For that reason, it’s often referred to as “wage-earner’s bankruptcy.”

-Chapter 15: Ancillary and cross-border cases. In the event that international bankruptcy filings have an impact on financial interests in the U.S., Chapter 15 bankruptcy, which was added to the legislation in 2005, enables collaboration between U.S. courts and foreign courts.

2 Classic Bankruptcy Models

The first model analysed will be the Altman Z-scores based on the multivariate discriminant analysis.

Multiple Discriminant Analysis (MDA) aims to group objects into two or more classes by classifying the sample into mutually exclusive and exhaustive groups based on several explanatory variables. So, the purpose of this analysis is to find out the differences between groups; discriminant analysis model can be written as follows:

$$Z = b_0 + b_1 X_{i1} + b_2 X_{i2} + \cdots + b_p X_p,$$

$$Z_i = b_0 + \sum_{j=1}^{p} b_j X_{ij},$$

where:

Z_i : discriminant score of discriminant function I,

X_{ij} : the value of the $i - th$ observation on the j-predictor variable,

i : number of object,

j : $1, \ldots, p$,

p : number of predictors,

b_0 : intercept,

b_j : discriminant coefficient for each j-predictor predictor.

MDA has been widely applied to social and economic research and in the Altman's model the variables used are financial ratios that can predict financial distress of a company.

2.1 Altman's Z-score.

The Z-score formula for predicting bankruptcy was published in 1968 by Edward I. Altman. In his study, published under the name of "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy", he considered a sample of 66 firms. Half of them failed, the other half didn't. In order to forecast the future condition of that company, he applied a model that was based on MDA and it is regarded by many as the pioneering research into the development of a model to predict the probability of failure in corporate entities. To reach the final result, he started using a huge number of ratios, related to the measurement of the liquidity, profitability, leverage, solvency and activity situation of the subjects of the sample. The final model, in order to catch all the aspects of the firm, was supposed to include those ratios which could catch every aspect of the company. The criteria used by Altman to reach his final model have been:

- Weighting of the significance of many created functions, as well as the contribution of each single variable.
- Analysis of possible correlations between the independent variables.
- Evaluation of the success matrix of each model.
- Personal judgement of the author.

The best function found by Altman combines linearly five common business ratios and they are then weighted by coefficients to calculate the Altman Z-score. The coefficients were estimated by selecting a group of companies that had filed for bankruptcy, then selecting a sample of companies that had survived that was matched by industry and approximate size (assets).

The Altman's Z-score formula is written as follows:

$$Z = 1.2X_1 + 1.4X_2 + 3.3X_3 + 0.6X_4 + 1.0X_5,$$

and the following are the key financial ratios that make up the Z-score model:

$X_1 = \frac{Working\ capital}{Total\ Assets}$.

It is a popular measure of liquidity because it compares the net liquid assets with the capitalization. If the company is expecting operational losses, net working capital is likely to decline.

$X_2 = \frac{Retained\ Earnings}{Total\ Assets}$.

This is an indicator of overall profitability because it reflects the profits made in the previous years. This ratio implicitly accounts for the age of the company. The consequence is that a discrimination is performed between the young and the older firms. Despite this, also in the real world there is a discrimination among firms that have a different age and the younger ones are more likely to fail.

$X_3 = \frac{Earnings\ Before\ interest\ and\ Taxes}{Total\ Assets}$.

It is a measurement of the operating profitability of the company (caught by the nominator) and, thanks to the denominator, the productivity is associated to the company amount of assets.

$X_4 = \frac{Market\ Value\ of\ Equity}{Total\ Liabilities}$.

This ratio compares the equity and the total debt and it helps in recognizing which value the equity can reach before the firm becomes insolvent. Its importance is due to the relationship with the other variables. This ratio strongly depends on the industry where the firms operate.

$X_5 = \frac{Sales}{Total\ Assets}$.

It is a financial ratio that show the sales generating ability of the company depending on the assets.

The Z-score rating is as follows:

- If Z-score > 3, it indicates that the company is doing well and is not likely to go bankrupt. When the score is above 3.0 the firm is in the “Safe Zone”, which means that bankruptcy is unlikely in this case. Investors should consider buying

the company's stock since the firm is unlikely to go bankrupt in the two coming years.

- If Z-score < 1.8, it indicates that the company is doing badly and has a high risk to go bankrupt. In this case, the company is in a "Distress Zone," which indicates that it will probably fail in two years. If an investor owns stock of this firm, it is suggested to him to sell before the business declares bankruptcy and he could potentially lose the whole investment amount. Later, Altman clarified that when the Z-score is very near to 0, investors should seriously be concerned about the firm's future.
- If 1.8 < Z < 3, it indicates uncertainty and cannot be easily predicted if the company will go bankrupt or not, it also raised concern for the affected companies. When a company's score is between 1.8 and 3.0, it is considered to be in the "Grey Zone" and has a moderate chance of going bankrupt. This score does not provide particularly clear information about the company's shares, but an investor should think about selling when the value is closer to 1.8, and vice versa when it is closer to 3.0.

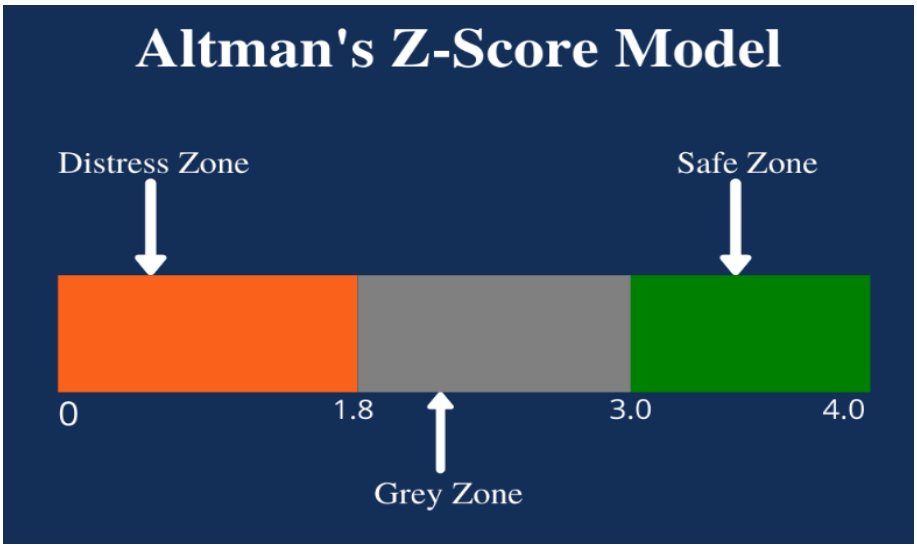

Figure 2.1: *Z-score ranges of financial stability.*

In its initial test the Altman Z-score formula is 72% accurate two years in advance concerning the bankruptcy, with a false negative rate of 6%. In its trial era of 31 years, the accurate rate was in between 80% and 90%, one year in advance concerning the bankruptcy, with a false negative rate in between 15% and 20% (Altman, 1968).

The model has been in use for more than 50 years and it is widely used and accepted. Despite that, the model does not specify the liquidity position of the company since a company cannot pay its debt obligations by profit declared in the books of account but by cash. Moreover, the model only described the company ratios than comparing the probability of a company going bankrupt. Instead of forecasting a company's bankruptcy, the model can serve as a warning indicator. In addition, because the Altman's Z-scores model relies on historical data from the financial statement to forecast bankruptcy, it does not take into account how interest rates and inflation may affect corporate operations in the future.

Altman's Z-score for International Credit Rating Agencies											
Defaulter	Amount of Liabilities	Date of Default	Z-Score					Rating at the time of default			The Consequences
	(in $ billion)		In year of default	1 year prior	2 years prior	3 years prior	4 years prior	S&P	Moody's	Fitch	
Bear Stearns	387	31-July-2007	0.29	-0.79	0.45	0.4	0.36	AA a A	A1 a A2		Acquired by JP Morgan Chase
AIG	807	16-Sep-2008	-1.03	-0.07	-0.02	0.42	0.23	AA- a A-	A1 a A2	AA- a A-	Bailed out by US Government
Lehman Brothers	392	23-Sep-2008	0.06	0.09	0.03	-0.03	0.29	AA, A1	P1 & A1	AA-- & F1+	Bankrupt
Washington Mutual Bank	303	25-Sep-2008	-0.35	-0.3	-0.07	-0.13	-0.3	A- & A2	Baa1 & P2	A- & F2	Acquired by JP Morgan Chase
Ford Motors	132	6-Apr-2009	1.32	1.03	1.23	1	1.29	CC	Caa1, B3	CCC, BB	Revived
MF Global	51	31-Oct-2011	0.23	0.47	0.37	0.41	0.46		Baa2 a Caa	BBB a BB+	Bankrupt

***Figure 3**:Example of Altman's Z-score applied to International Credit Rating Agencies*

2.2 Logistic Model

Since 1968, the early multivariable models were largely using MDA. As was already indicated, multiple discriminant analysis divides businesses according to their characteristics into groups (in this example, bankrupt or non-bankrupt). The products of the ratios and their coefficients are then combined to provide a discriminant score, which enables the classification of the firm. Coefficients are then determined for each ratio.

In the late 1970's logit analysis began to appear, and its main feature is that it considers the probability that the firm will go bankrupt. To better appreciate the functioning of the logistic regression in this section will be described its main characteristic starting from the linear regression.

Regression models play a central role in the study of relationship between variables, and at the same time, they are the basis for more complex models. They allow to estimate how one or more independent variables $X_1, \ldots, X_p$, with $p \geq 1$, influence the distribution of the dependent variable Y.

The easiest regression model is the simple linear regression model ($p = 1$), where the variables involved are two: the dependent variable Y and only one independent variable X. The aim of the simple linear regression model is to estimate the relationship between two quantitative variables.

The formula for a single linear regression is:

$$Y = \beta_0 + \beta_1 X + \varepsilon,$$

where y is the predicted value of the dependent variable (Y) for any given value of the independent variable X. β_0 is the intercept, the predicted value of Y when $X = 0$, and β_1 is the slope, the average increase in Y associated with a one-unit increase in X and ε is the error term.

Simple linear regression is a useful approach for predicting a response on the basis of a single predictor variable. However, in practice we often have more than one predictor; a better approach is to extend the simple linear regression model so that it can directly accommodate multiple predictors. We can do this by giving each predictor a separate

coefficient. In general, suppose that we have p distinct predictors. Then the multiple linear regression model takes the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + X_p \beta_p + \varepsilon,$$

where X_j represents the jth predictor and β_j quantifies the association between that variable and the response. We interpret β_j as the average effect on Y of a one unit increase in X_j, holding all other predictors fixed.

Multiple linear regression, same as simple linear regression, is based on the following assumptions:

1. Linear relationship: There exists a linear relationship between the independent variable, x, and the dependent variable, y.

2. Independence: The residuals are independent.

3. Homoscedasticity: The residuals have constant variance at every level of x.

4. Normality: The residuals of the model are normally distributed.

The linear regression model assumes that the response variable Y is quantitative. But in many cases the dependent variable is qualitative. Especially for a binary response with a commonly use 0 or 1 coding, some of the estimates might lie down outside the $[0,1]$ interval, giving them no meaning as probabilities. So there are two main reasons not to perform classification using a regression model: first of all a regression method cannot fit a qualitative response with more than two classes; secondly a regression method cannot will not provide a meaningfully estimates of $\Pr(Y|X)$, even with just two classes. As anticipated before one most widely used to perform prediction is for sure logistic regression, which is mainly used when the dependent variable is categorial. For the purpose of this thesis, for example, the observations will be classified in Default or Non-Default. As shown in the left panel is possible to predict $p(X) > 1$ or $p(X) < 0$ for some values of X; so, to avoid this issue it is necessary to model the probabilities $p(X)$ by using a function that allows to obtain outputs between 0 and 1 for all values of X. One of these functions is the logistic function, and using only one independent variable for simplicity:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

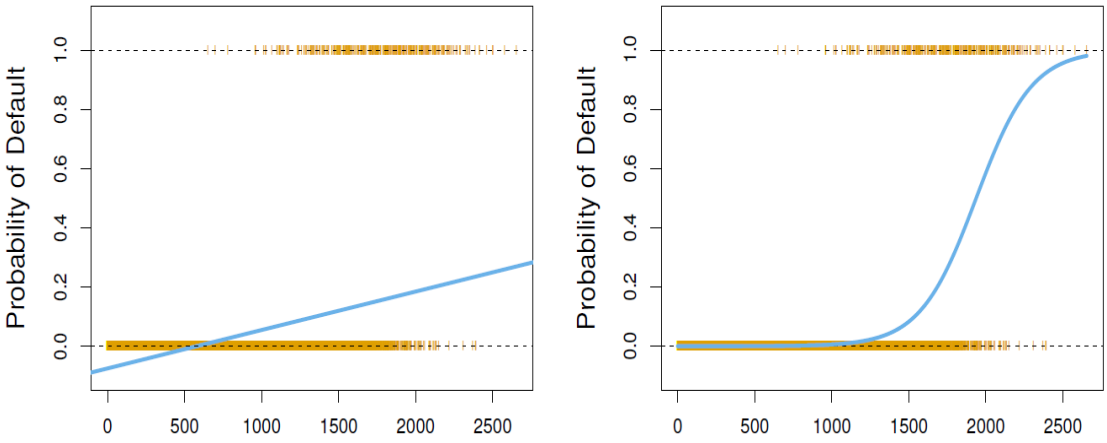

Figure 4: *Comparison between regression model and logistic model.*

The logistic function will always produce an S-shaped curve of this form, so regardless of the value of X it is possible to obtain a sensible prediction. After a bit of manipulation, it is possible to find that

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}.$$

The quantity $p(X)/[1-p(X)]$ is called the odds and it is the ratio between the probability of success and the probability of unsuccess and it can take on any value between 0 and ∞; by taking the logarithm of both sided, we arrive at

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

The left-hand side is called the log odds or logit, so the regression model has a logit that is linear in X. Recalling the interpretation of the parameter β_1, it gives the average change Y associated with a one-unit increase in X while in a logistic regression model, increasing X by one unit changes the log odds by β_1. However, in the logistic function the relationship between $p(X)$ and X is not a straight line, so $\beta_1$does not correspond to the change in $p(X)$ associated with one unit-unit increase in X. The amount that $p(X)$ changes due to a one-unit change in X depends on the current value of X.

Independently of the value X, if β_1 is positive then increasing X will be associated with increasing $p(X)$ and if β_1 is negative then increasing X will be associated with decreasing $p(X)$.

The estimates $\hat{\beta}_0$and $\hat{\beta}_1$are chosen to maximize the likelihood function.

$$\ell(\beta_0, \beta_1) = \prod_{i:yi=1} p(x_i) \prod_{i\prime:yi\prime=0} (1 - p(x_{i\prime})).$$

As in the case of the single and multiple linear regression the logistic regression can be extended and generalize as follow:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X + \cdots + \beta_p X_p,$$

where $X = (X_1, \ldots, X_p)$ are p predictors. The probability can be rewritten as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}.$$

2.3 Ohlson's Logit Model

The first example of the use of logit regression to predict the financial distress of a company is the Ohlson's O-score Model, developed in 1980 as alternative to the Altman's Z-score. The purpose of this model was to remove the following restrictive MDA assumptions by supplying failure probabilities as the function's output:

- The explanatory variables are normally distributed.
- Equal variance and covariance of the explanatory variables for the bankrupt and non-bankrupt firms.
- Bankrupt and non-bankrupt firms are matched according to criteria such as size and industry, and these tend to be somewhat arbitrary.

The Ohlson model consists of nine financial parameters that may be categorized into four primary groups and are useful for determining a company's size, financial structure, performance, and current liquidity. In order to assess the impact of these four variables he uses a logit model which implies the use of 9 financial ratios:

• X1: $log(\frac{Total\ Assets}{GNP})$.

• X2: $\frac{Total\ Liabilities}{Total\ Assets}$.

• X3: $\frac{Working\ Capital}{Total\ Assets}$.

• X4: $\frac{Current\ Liabilities}{Current\ Assets}$.

• X5: 1 if Total Liabilities exceed Total Assets, 0 otherwise

• X6: $\frac{Net\ Income}{Total\ Assets}$.

• X7: $\frac{Operating\ income\ before\ depreciation}{Total\ Liabilities}$

• X8: 1 if Net Income was negative for the last 2 years, 0 otherwise

• X9: $\frac{(NI_t - NI_{t-1})}{(|NI_t| + |NI_{t-1}|)}$

where NI_t is net income for the most recent period. The denominator acts as a level indicator. The variable is thus intended to measure change in net income.

The overall O-Score function is defined as:

O -Score $= -1.3 - 0.4X_1 + 6.0X_2 - 1.4X_3 + 0.8X_4 - 2.4X_5 - 1.8X_6 + 0.3X_7 - 1.7X_8 - 0.5X_9$.

$$p(default) = \frac{e^{O-score}}{1 + e^{O-score}}$$

The logistic function is then used to transform the O-score into probability, with $P > 0.5$ and $P < 0.5$ denoting risky and safe companies, respectively.

According to Ohlson the model was built up by using more than 2000 companies, while Altman used only 66 companies. Therefore, the O-score model is more reliable than the Z-score to forecast bankruptcy, particularly when taking into account a 2-year time frame, where Ohlson's model achieves 90% accuracy as opposed to Altman's model's 70%.

Main Components	Weight (B)	Components Formula	Components Category
X1: Adjusted Size	0,407	Log(Total assets/GNP price-level index)	It determines the size of a company by adjusting the total assets for inflation.
X2: Leverage	6,03	Total Liabilities/Total assets	Solvency: it determines the level of indebtedness. The higher the debt, the higher the risk of bankruptcy.
X3: Working capital measure	-1,43	Net working capital/Total assets	Assets Usage: measures the percentage of liquid assets in a company.
X4: Inverse current ratio	0,0757	Current Liabilities/ Current Assets	Liquidity. It shows the level of liquidity of a company.
X5: Discontinuity correction for leverage measure. (Dummy Variable 1)	-1,72	1 if total liabilities exceed total assets, 0 otherwise	Leverage: it helps to correct the extreme leverage level of a company
X6: Return on assets (ROA)	-2,37	Net Income/Total Assets	Investment profitability: determines the profit level of a company whichis assumed to be negative because of default
X7: Fund to debt ratio	-1,83	Operating income before depreciation/Total liabilities	Liquidity: measures the ability of a company to finance its debt using operating cash flow alone.
X8: Discontinuity correction for ROA (Dummy Variable 2)	0,285	1 if a net loss for the last two years, 0 otherwise	Liquidity: it helps to correct the two-year losses effect of a company.
X9: Change in net Income	-0,521	(Net Income(t) -Net Income(t-1))/(Net Income(t) -Net Income(t+1))	Profitability. It measures possible continuous losses for two continuous for two consecutive years in the history of company life.

Table 1: *Description of the financial parameters used in the Ohlson model.*

3 Machine Learning models

The previously analysed models represent the first, approach to predict bankruptcy; with the main purpose of overstate the main limits of these pioneering models, Machine learning (ML) models started to be developed and gain ground.

Machine learning is a field of study and application within artificial intelligence (AI) that focuses on developing algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. Machine learning is fundamentally about building mathematical models that are trained on data. These models are made to identify patterns, identify underlying links, and predict or decide based on the data's information. The following steps are often included in the process:

1. Data collection: collecting relevant data that is representative of the issue or domain you want the machine learning is intended to understand.
2. Data preprocessing: Getting ready the data for analysis includes cleaning it up and performing tasks like normalizing it, addressing missing values, and removing outliers.
3. Feature extraction and selection: Identifying and selecting the most relevant features or attributes from the data that will be used to train the machine learning model.
4. Model training: The machine learning model is trained using the prepared data, which entails modifying the model's internal parameters to reduce mistakes or discrepancies between the projected outputs and the actual outputs in the training data.
5. Model evaluation: Assessing the performance of the trained model by testing it on a separate set of data called the test set, which was not used during the training phase. This evaluation helps to determine the model's accuracy and generalization capability.

6. Model deployment and prediction: Once the model has been trained and evaluated, it can be deployed to make predictions or decisions on new, unseen data.

There are various types of machine learning algorithms, including supervised learning, unsupervised learning, and reinforcement learning. In this thesis the task is to predict bankruptcy, which is a problem of binary classification; it aims to identify which of a set of categories (bankrupt or non-bankrupt) an observation belongs to.

The algorithm that implements classification is called a classifier, which is the mathematical function implemented by a classification algorithm that sorts input data into a category. The issue belongs to the category of supervised machine learning, where an input dataset is given to the algorithm, which is then tuned to provide a predetermined set of results.

In the next sections will be provided the description of some of supervised machine learning algorithms

3.1 Shrinkage Methods

As already stated, bankruptcy models can be based on a variety of predictors of distress, most of which are built using accounting variables and financial ratios. Variable selection is crucial for identifying a subset of the most important bankruptcy predictors and increasing prediction accuracy, especially in the regression framework. This shrinkage, also referred as regularization, allows also to reduce the variance.

In the Shrinkage approach, a model with all the predictors is fitted. However, the estimated coefficients are shrunken towards zero relative to the least squares estimates. Depending on the shrinkage technique used, some of the coefficients might be precisely zero. As a result, variable selection can be done using shrinkage approaches. Statisticians have developed two main shrinkage methods for this purpose: ridge regression and the lasso.

According to James et al. (2017), Ridge regression is very similar to the OLS method in that it minimizes the residual sum of squares to estimate the linear regression's parameters (RSS).

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{i=1}^{J}\beta_j\, x_{ij})^2.$$

The ridge introduces the complexity parameter λ, that multiplied by the square of the Beta coefficients (intercept excluded) defines the penalty term:

$$\hat{\beta}^{ridge} = argmin_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}$$

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: larger the value of lambda, greater the amount of shrinkage. $\lambda = 0$ means that there are no penalties in the model, so the estimated parameters will be the same as the obtained by OLS.

An alternative way to write the ridge problem is

$$\hat{\beta}^{ridge} = argmin_{\beta}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

$$subject\ to \sum_{j=1}^{p} \beta_j^2 \leq t,$$

which makes explicit the size constraint on the parameters.

When $\lambda \to \infty$ the value of the penalty term is high, which lead many coefficients to be close to zero, but it will not imply their exclusion from the model. Additionally, it should be noted that as the tuning value λ is increased, the model becomes less flexible, resulting in a smaller variance but a higher bias. Therefore, the best bias-variance trade-off can be found with an optimal choice of λ.

The lasso method (least absolute shrinkage and selection operator) fixes the ridge regression's drawback (James et al., 2017). It enables the estimated beta coefficients to be left out of the model when they are zero and avoid challenges in the model interpretation. Ridge and the lasso have similar formulas: the difference is in the structure of the penalty, as it is necessary to calculate the sum of the absolute value of the betas

$$\hat{\beta}^{lasso} = argmin_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

$$subject\ to \sum_{j=1}^{p} |\beta_j| \leq t,$$

As in the ridge regression, the lasso forces the estimated coefficients towards zero but, the absolute value present, forces some of them to be exactly equal to zero. It is possible in this situation to select the variables and remove them from the model. As in the analogous case, when the lambda value is close to zero (λ = 0) it means that there is no penalty in the model and the estimates produced are the same as obtained from the least squares. On the other hand, if $\lambda \to \infty$ the model provided is a null model because all the coefficients are equal to zero.

3.1.1 Beyond Linearity

In the previous section the focus was on linear models. In comparison to other models that will be described later, linear models are easier to understand and use; they offer benefits in terms of interpretation and inference but in terms of predictive power the standard linear regression can have significant limitations; this is because the linearity assumption is almost always an approximation, and sometimes a poor one.

The lasso and the ridge regression can be used to enhance least squares by reducing the complexity of linear models and, as a result, the variance of the estimates. But a linear model is still used, which can be improved.

In the next sections will be relaxed the linearity assumption preserving at the same time as much interpretability as possible.

3.2 Decision Tree

Finding patterns for the prediction of bankrupt and non-bankrupt enterprises in a sample using firms' financial ratio amounts is the major objective of tree-based approaches partition. The space is divided hierarchically using decision trees. Starting with the total space, it is subsequently broken down into smaller areas in a recursive manner. In the end, every region is assigned to a class label. In this section the methods will be described from a mathematical point of view both of regression trees and classification trees.

In a regression tree there are two main steps:

1. The predictor space should be divided into a set of possible values for X_1, X_2,…. X_p, into j distinct and non-overlapping regions, R_1, R_2,…. R_j.
2. For every observation that falls into the region R_j, we make the same prediction, which is simply the mean of the response values for the training observations in R_j.

The goal then is to find boxes R_1, R_2,…. R_j that minimize the RSS, given by:

$$\sum_{j=1}^{J}\sum_{i\in R_j}(y_i-\hat{y}_{Rj})^2,$$

where $\hat{y}_{R_j}$is the mean response for the training observations within the j_{th} box. The procedure used is known as recursive binary splitting. It is a top-down technique since it starts at the top of the tree, where all of the observations are grouped into a single region, and separates the predictors space down the middle, making the best split possible at each stage.

To perform recursive binary splitting, from all the predictors $X_1, X_2,\ldots X_p,$ is necessary to seek for any j, and the split point s among all its the possible values such as the pair of half-planes

$$R_1(j,s)=\{X|X_j<\ s\}\ and\ R_2(j,s)=\{X|X_j\geq\ s\}$$

able to minimize the equation:

$$\sum_{i:x_i\in R_1(j,s)}(y_i-\hat{y}_{R_1})^2+\sum_{i:x_i\in R_2(j,s)}(y_i-\hat{y}_{R_2})^2$$

where $\hat{y}_{R_1}$ is the mean response for the training observations in $R_1(j,s)$ and $\hat{y}_{R_2}$ is the mean response for the training observations in $R_2(j,s)$.

Then the process should be repeated to create all the R_j regions: the response for a specific test observation is determined using the mean of the training observations in the region to which the test observation belongs after they have been created.

A classification tree and a regression tree are quite similar, with the exception that a classification tree is intended to predict a qualitative response instead of a quantitative one. For a classification tree, the prevision is that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. The process of building a classification tree is very similar to the process of building a regression tree, however in the latter, binary splits cannot be made using RSS, so another criterion must be employed; the two most popular are the Gini index and the entropy described as follows:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

which is a measure of the total variance across the K classes. Here $\hat{p}_{mk}$ represents the proportion of training observations in the mth region that are from the $k - th$ class. The index assumes a small value if all the $\hat{p}_{mk}$ are close to zero or one, represents the proportion of training observations in the mth region that are from the k −th class.

An alternative to the Gini index is entropy, given by

$$D = \sum_{k=1}^{K} \hat{p}_{mk} log \hat{p}_{mk}.$$

Since $0 < \hat{p}_{mk} \leq 1$, it follows that $0 \leq \hat{p}_{mk} log \hat{p}_{mk}$. One can show that the entropy will take on a value near zero if the $\hat{p}_{mk}$'s are all near zero or near one.

Therefore, like the Gini index, the entropy will take on a small value if the $m - th$ node is pure. In fact, it turns out that the Gini index and the entropy are quite similar numerically.

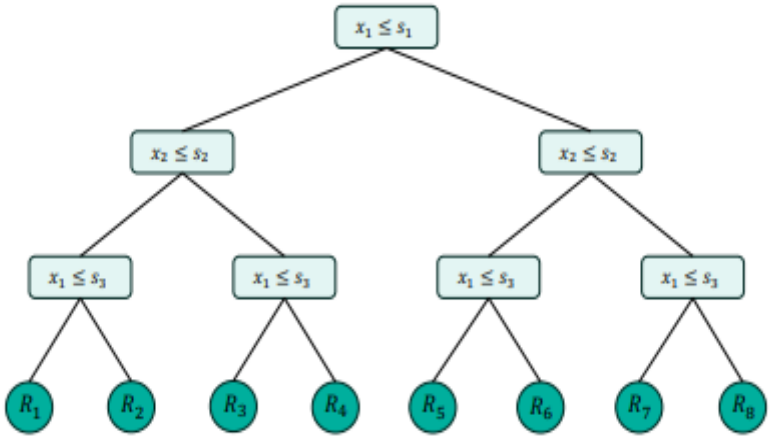

Figure 5: *Graphic representation of Tree algorithm*

Decision trees have some advantages and some disadvantages; among the advantages, trees are even easier to comprehend and more analogous to human decision-making than regression and other classification techniques.

Additionally, trees can handle qualitative predictors without the need of dummy variables, and they can be graphically displayed and easily understood.

In contrast, trees can be relatively non-robust and do not have the same level of prediction accuracy as other regression techniques.

Aggregating trees can solve these issues; this is the goal of techniques like bagging and random forests that enhance the prediction performance of trees.

3.3 Bootstrap

The goal is to utilize the observed sample to estimate the population distribution, and the bootstrap is a broadly applicable and extremely powerful statistical technique that can be used to assess the uncertainty associated with a given estimator. Then, by continuously obtaining random samples with replacement from the observed sample, bootstrapping is a resampling technique that leverages data from one sample to create a sampling distribution. Bootstrap involves repeated random sampling with replacement from the original data, $X = (x_1, x_2, \ldots x_n)$ to produce random samples of the same size n of the original sample, each of which is known as a bootstrap sample, x^* and each provides an estimate $\hat{\theta}^*$ of the parameter of interest; "with replacement" means that every observation can be sampled more than once in each bootstrap sample.

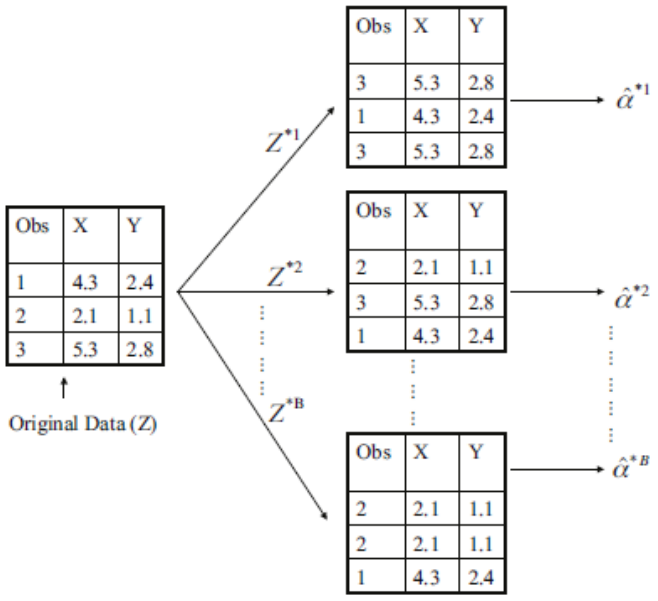

Figure 6: *Graphic representation of bootstrap algorithm*

The objective is to accurately simulate how the sample is obtained from the population in the bootstrap samples from the observed data. Since the standard error predicted from the standard deviation of the statistics is produced from the bootstrap samples, the procedure must be repeated multiple times in order to get the necessary information about the variability of the estimator. The benefit of the bootstrap is that it allows to obtain distinct data sets by repeatedly sampling observations from the original data set instead of obtaining independent data sets from the population.

3.4 Bagging

The bootstrap method can be also applied to statistical procedures in order to improve their features. Bootstrap aggregation, or bagging, is a general-purpose procedure that can be used to reduce bagging variance of a statistical learning method and it especially it is widely used in the context of decision trees, (James et al., 2017).

Recall that given a set of n independent observations $Z_1, \dots, Z_n$ each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by $\sigma^2/\, n$**:** averaging a set of observations reduces the variance. Therefore, selecting a large number of training sets from the population, creating a new prediction model with each training set, and averaging the resulting predictions is a logical technique to decrease variance and increase test set accuracy of a statistical learning method.

Theoretically, should be calculated $\widehat{f^1}(x), \widehat{f^2}(x), \dots, \widehat{f^B}(x)$ using B separate training sets, and average them in order to obtain a single low-variance statistical learning model, given by

$$\widehat{f_{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f^b}(x).$$

Practically, is usually not possible to access to multiple training sets. But through the bootstrap is possible to obtain multiple samples from the single training data set.

The first step is to generate B different training data sets; subsequently the method should be applied to the $b-th$ bootstrapped training set in order to get $\widehat{f^{*b}}(x)$, and then average all the predictors in order to obtain

$$\widehat{f_{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\widehat{f^{*b}}(x).$$

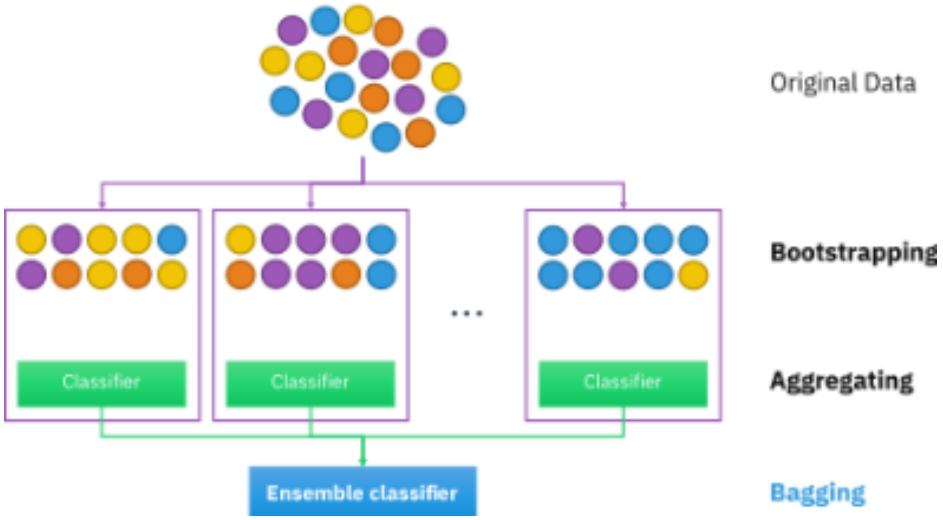

Figure 7: *Graphic representation of Bagging algorithm.*

.

Bagging process offers remarkable advantages, especially when hundreds of trees are processed all at once and it can also be applied to a classification problem in order to predict a qualitative Y. The easiest method is to identify the class predicted by each of the B trees for a certain test observation and then vote by majority. The overall prediction is the class that appears most frequently in the B predictions.

3.5 Random Forest

As already mentioned, bagging is a method for reducing high-variance procedures. A technique called random forests produces a big collection of de-correlated trees, averages them, and is essentially a bagging modification. This method's name comes from the fact that it consists of many decision trees and that its primary objective is to overcome a single decision tree and its shortcomings. On bootstrapped training samples, a number of decision tree forests, similar to bagging are, constructed. But when creating these decision trees, a random sample of m predictors is selected as split candidates from the entire collection of p predictors each time that a split in a tree is taken into account. The split is allowed to use only one of those m predictors: at each split, a new sample of m predictors is collected.

For instance, in the subsequent analysis $m \approx \sqrt{p}$ is used, which means that the number of predictors taken into account at each split is roughly equal to the square root of the total number of predictors.

To put it another way, when creating a random forest, the algorithm is not even permitted to take into account the majority of the predictors that are accessible at each branch in the tree. Assume that the data set contains a few predictors that are moderately strong and a few really strong predictors. The majority or all of the trees in the collection of bagged trees will then use this reliable prediction in the top split: consequently, all of the bagged trees will look quite similar to each other. As a result, there will be a strong correlation between the predictions obtained from the bagged trees. Unfortunately, there is not as much variance reduction from averaging several highly linked quantities as there is from averaging numerous uncorrelated quantities.

This issue is solved by random forests, which require that each split take into account only a portion of the predictors. Therefore, on average $(p - m)/p$ of the splits will not even consider the strong predictor, and so other predictors will have more of a chance. We might conceive of this procedure as decorrelating the trees, which reduces variability and increases reliability of the average of the generated trees.

The size of the predictor subset, measured in m, is the primary distinction between bagging and random forests. This is equivalent to bagging, for instance, if a random forest is constructed using the formula $m = p$.

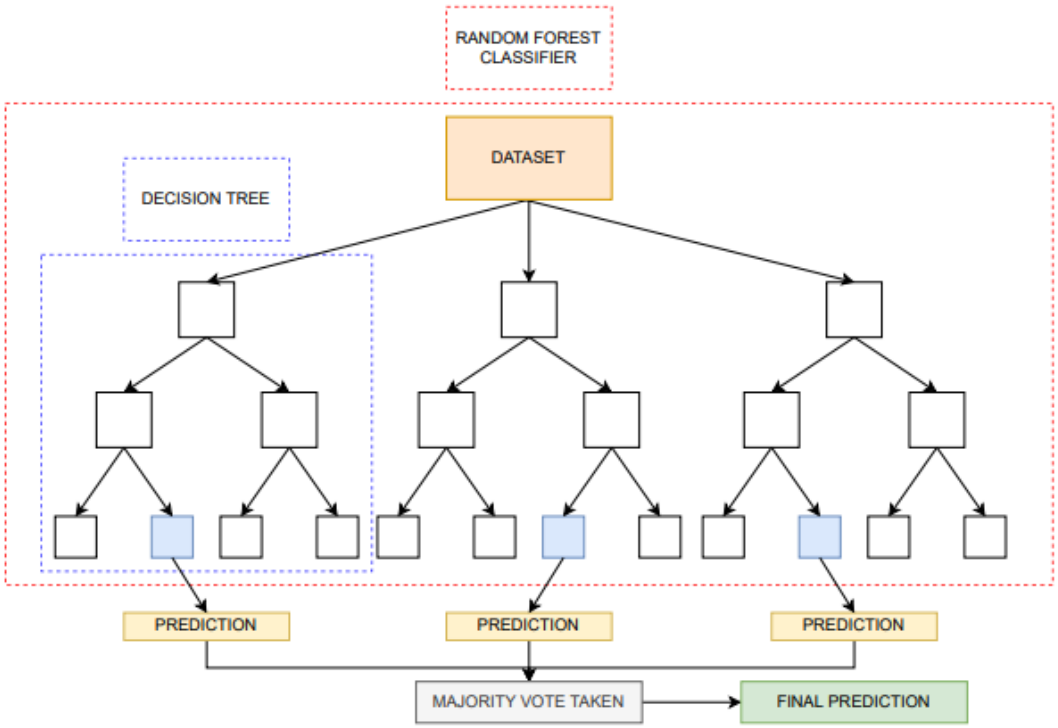

Figure 8: *Graphic representation of Random Forest algorithm for classification.*

In the subsequent analysis of the relevant variables, the feature importance in random forest will be assessed according to two measures of importance given for each variable:

- Mean Decrease Accuracy: which is the average decrease of model accuracy in predicting the outcome of the out-of-bag samples when a specific variable is excluded from the model. The Mean Decrease Accuracy illustrates how much accuracy is lost when a variable is removed from the model; the factors will be listed in order of decreasing relevance and more the accuracy suffers, the more important the variable is for the successful classification

- Mean Decrease Gini: which is the average decrease in node impurity that results from splits over that variable. In other words, the mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. The higher the value of mean decrease accuracy or mean decrease Gini score, the higher the importance of the variable in the model.

3.6 Boosting Procedures

In this section will be discussed boosting, another possible approach for improving the predictions resulting from a decision tree. Boosting is a generic strategy that can be used with a variety of statistical learning techniques both regression and classification. Recall that bagging entails utilizing the bootstrap method to make many copies of the initial training data set, fitting a different decision tree to each copy, and then merging all the trees to produce a single predictive model. Notably, each tree is constructed independently from the other trees using a bootstrap data set. Similar to other tree-growing techniques, boosting grows the trees in a certain order utilizing data from earlier trees but, instead of using bootstrap sampling, boosting fits each tree on a modified version of the initial data set.

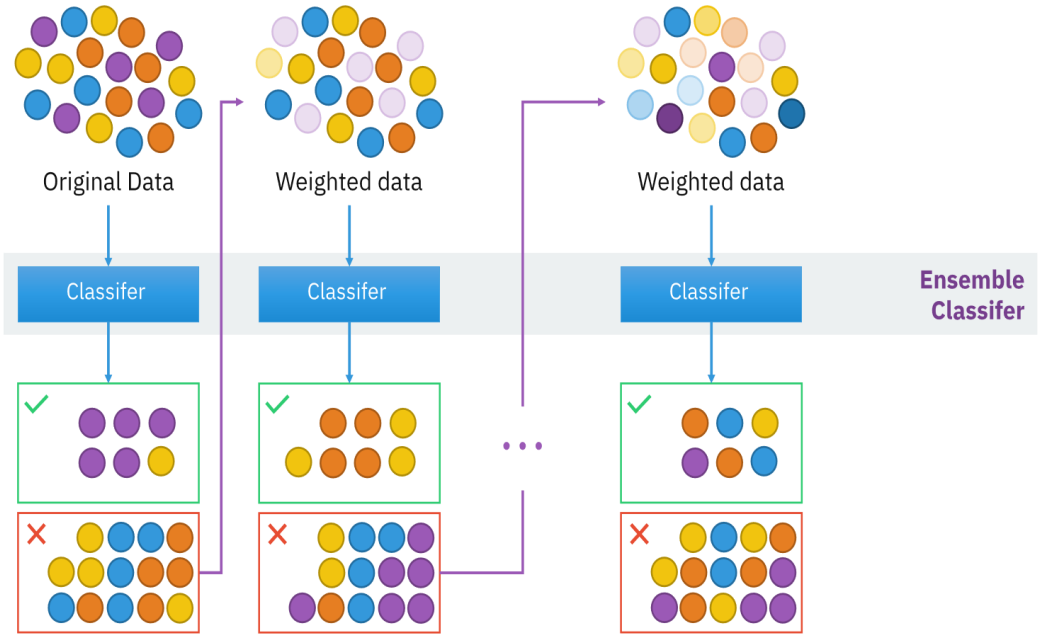

Figure 9: *Graphic representation of Boosting algorithm.*

This logic has been implemented in many different ways; the original one is called AdaBoost (*Adaptive Boosting),* introduced by Yoav Freund and Robert E. Schapire (1995) and presented in the algorithm below. It can be described briefly as follow: a greater weight is assigned to observations that are classified poorly in the early stage. Thus, the goal is to enhance model performance, focusing mostly on the subsets where the first classifier struggled the most. This procedure is iterative; at the beginning a base model

is chosen among the classifier discussed earlier. In the first step, the base classifier is fitted to the data by assigning the same weight to each observation. Then, different weights are used to fit a new classifier at the end of each iteration. At the end of the process, a new classifier is identified through a weighted majority vote among the classifiers fitted in all the iterations.

In AdaBoost as the number of iterations increases, the significance of the base classifier selection tends to diminish because iteration-focused classification becomes more and more closely related to classification decision. This explains why a tree built with one or a maximum of two levels, without pruning, is frequently used as a base classifier. They are commonly referred to as weak classifiers in this situation because their error rate is only marginally better than random guessing. When the weak classifier is a tree, the sequence of interactions permitted by the final model is related to the weak classifier's level count.

For instance, only primary effects are permitted if just one level of trees is present. Indeed, when a tree is fully grown, all its leaves are pure, the classifier makes no errors on the training data, and its error rate is therefore 0. This means that boosting will stop because there are no wrongly classified training units to be boosted. On the other side, the tree will overfit the data if it is extremely large without being fully formed: for this reason, it is usually better not to use very large trees for boosting.

1. Initialize weights $w_i = \frac{1}{n}, \qquad i = 1,2, \dots, n.$

2. Cycle for $b = 1, \dots, B$:

   a. Fit classification model $C_b(x)$ to the training set, with target values 0 or 1, by weighting the observations by w_i.

   b. Obtain:

$$err_b = \frac{\sum_{i=1}^{N} w_i \, I(y_i \neq C_b(x_i))}{\sum_{i=1}^{N} w_i},$$

$$\alpha_b = \log \frac{1 - err_b}{err_b}.$$

c. Assign the new weights:

$$w_i = w_i \exp\{\alpha_b I(y_i \neq C_b(x_i))\}, \quad i = 1,2,\dots,n.$$

3. The new classifier is:

$$C(x) = 1 \; if \; \frac{\sum_{b=1}^{B} \propto_b C_b(X)}{\sum_{b=1}^{B} \propto_b} > \frac{1}{2},$$
$$0 \; otherwise.$$

Another boosting procedure is the Gradient Boosting, developed by Jerome H. Friedman (2001), which works by sequentially adding predictors to an ensemble with each one correcting for the errors of its predecessor. Differently from AdaBoost it does not change the weights of data points, but it trains on the residual errors of the previous predictor. The name, gradient boosting, is used because it combines the gradient descent algorithm and boosting method.

Here will be described the process of the boosting regression tree in order to understand better the idea behind this procedure.

Unlike fitting a single large decision tree to the data, which amounts to fitting the data hard and potentially overfitting, the boosting approach instead learns slowly. A decision tree is fitted to the residuals from the model; in other words, instead of using the outcome as the response, a decision tree is fitted to the residuals from the model and the residuals are then updated by adding this new decision tree to the fitted function. Each of these trees can be rather small, with just a few terminal nodes, determined by the parameter used to determine the number of splits in each tree in the algorithm. By fitting small trees to the residuals, the decision tree is slowly improved in areas where it does not perform well. The process is slowed down even more by the shrinkage parameter, allowing more trees of various shapes to attack the residuals. The performance of statistical learning methods that learn slowly is generally good. Also in this case, unlike in bagging, each tree's construction depends heavily on the trees that have already been produced.

In general, boosting trees has three tuning parameters:

- The number of trees. Unlike bagging and random forests, boosting can overfit if the number of trees is too large, although this overfitting tends to occur slowly if at all. Cross-validation is used to select the proper number of trees.

- The shrinkage parameter, which is a small positive number. This regulates how quickly boosting learns. The ideal choice can vary depending on the task and typical values of 0.01 or 0.001. Very small may necessitate the use of a very high number of trees to obtain good performance.

- The number of splits in each tree, which controls the complexity of the boosted ensemble. When the number of splits is equal to 1, each tree is made from a single split, and this solution is frequently successful. Since each term only contains one variable, the boosted ensemble is in this case fitting an additive model. More generally the number of splits is the interaction depth, and controls the interaction order of the boosted model, since it can involve at most the same number of variables.

As said before boosting procedure has been implemented and improved in many different ways. In this work will be applied to the data wo variants of gradient boosting algorithms: XGBoost (Extreme Gradient Boosting) and CatBoost (Categorical Boosting); which main features will be briefly described below.

3.6.1 XGBoost

XGBoost stands for Extreme Gradient Boosting. It's a parallelized and carefully optimized version of the gradient boosting algorithm presents for the first time in a paper written by Tianqi Chen and Carlos Guestrin. It introduced two new methods to help the model in avoiding overfitting. The first method, originally included in random forest and known as columns or feature subsampling, helps in more effectively training each independent learner on a separate subset of features. Shrinkage is a second method that, like a learning rate in stochastic optimization, lessens the impact of each particular tree by scaling the output weights after each iteration of tree-boosting optimization (Chen and Guestrin, 2016). The training time is significantly increased by parallelizing the entire boosting procedure: thousands of models are trained on diverse subsets of the training dataset instead of developing the best model feasible on the data (as in traditional approaches) and then vote for the best-performing model. In many situations, XGBoost outperforms conventional gradient boosting methods.

More in general, the most important features of XGBoost are:

- *Gradient Boosting*: XGBoost utilizes gradient boosting. Each new model is then trained to predict the residuals (the differences between the actual and predicted values) of the previous model, effectively reducing the error in subsequent iterations.

- *Parallelization:* The model is implemented to train with multiple CPU cores.

- *Regularization:* XGBoost includes different regularization penalties to avoid overfitting. It adds penalty terms to the loss function that control the complexity of the model. Regularization helps in reducing the impact of individual trees and encourages simpler models.

- *Feature importance:* XGBoost provides a measure of feature importance based on the number of times a feature is used across all the trees in the ensemble. It calculates feature importance based on the number of times a feature is used to

split across all the trees in the ensemble. This information can help identify the most relevant features for the prediction task.

- *Sparsity-aware Split Finding*: In many real-world problems it is not uncommon to find sparse data. Sparsity can be caused by the presence of missing values in data, frequent zero entries in the statistics and artifacts of feature engineering as one-hot encoding. XGBoost can handle sparse data effectively. It has built-in techniques that allows to make the algorithm aware of the sparsity pattern in the data to learn the best imputation strategy for handling missing values during training (Chen and Guestrin, 2016).

XGBoost is then a very flexible algorithm: regression, classification, and ranking are just a few of the diverse tasks that XGBoost can be used for. Users can adapt the algorithm to suit particular problem requirements thanks to its support for a variety of goal functions. This adaptability allows XGBoost to be used in a variety of areas.

In chapter 5 the feature importance in XGBoost model will be assessed according to:

- Gain: it implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model. A higher value of this metric when compared to another feature implies it is more important for generating a prediction.
- Cover: it’s a metric of the number of observations related to this feature.
- Frequency; it is the percentage representing the relative number of times a particular feature occurs in the trees of the model.

3.6.2 CatBoost

CatBoost is a free and open-source machine learning technique that combines the terms "Category" and "Boosting.". It was developed by Yandex (the Russian version of Google) in 2017. Yandex claims that CatBoost has been used in a variety of fields, including forecasting, self-driving cars, search ranking, virtual assistants, and recommendation systems.

CatBoost is characterized by some key features that they make it better than the counterparts:

- Symmetric trees: Unlike XGBoost, CatBoost builds symmetric (balanced) trees. The same condition is used to divide leaves from the previous tree at each stage. For each of the level's nodes, the feature-split pair that causes the lowest loss is chosen. This balanced tree architecture helps with effective CPU implementation, cuts down on prediction time, creates quick model implementers, and reduces overfitting because the structure acts as regularization.

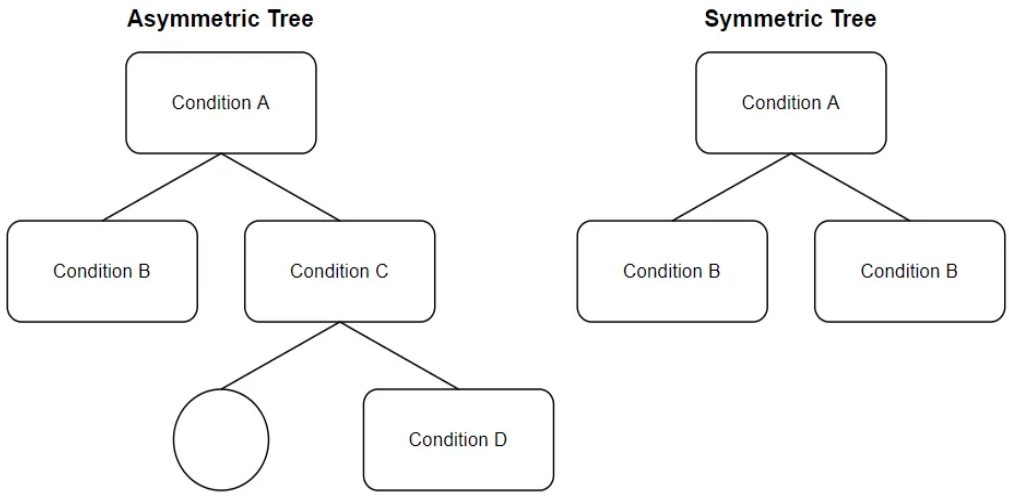

Figure 10: *Graphic representation of asymmetric tree and symmetric tree*

- Ordered boosting: Due to a phenomenon known as prediction shift, traditional boosting algorithms are prone to overfitting on small/noisy datasets. These methods use the same data instances that were used to build the model,

eliminating the possibility of encountering data that hasn't yet been seen. In contrast, CatBoost employs the idea of ordered boosting, a permutation-driven method, to train the model on a portion of data while computing residuals on a different subset, preventing target leakage and overfitting.

CatBoost handles numeric features like other tree-based algorithms, by selecting the best possible split based on the information gain. In the categorical framework decision trees are splitted based on classes rather than a threshold in continuous variables. The split criterion is intuitive as the classes are divided into sub-nodes. All machine learning algorithm requires parsing of input and output variables in numerical form; CatBoost provides the various native strategies to handle categorical variables:

- Category-based statistics CatBoost handles categorical characteristics by using target encoding with random permutation. As it only adds a new feature to account for the category encoding, this approach can be quite effective for columns with high cardinality. To avoid overfitting brought on by feature bias and data leakage, the encoding approach now includes random permutation.

- Greedy search for combination: CatBoost also automatically combines categorical features, most times two or three. To keep possible combinations limited, CatBoost does not enumerate through all the combinations but rather some of the best, using statistics like category frequency. So, for each tree split, CatBoost adds all categorical features (and their combinations) already used for previous splits in the current tree with all categorical features in the dataset.

3.7 Support Vector Machines

Support vectors are power-supervised training machine learning methods for segmentation. Vapnik et al. (1995) first proposed SVMs as one effective algorithm for model pattern recognition. Definition of a hyperspace comes first in this section, followed by a discussion of the support vector classifier and finally the presentation of the support vector machines.

In mathematics and geometry, a hyperplane is a subspace of one dimension less than the ambient space it resides in. A hyperplane, in general, separates two sections of an n-dimensional space. The points on one side of the hyperplane meet one set of requirements, while the points on the opposite side meet another set of requirements.

The fundamental idea of this approach is to place observations on a $p-1$ dimensional hyperplane and utilize those observations to guide vectors across the hyperplane according to the various responses. This can be seen by comparing it with the two-dimensional predictor space in classification trees. According to James et al. (2017) and Hastie et al. (2009), the hyperplane is for p dimensions by definition,

$$\beta_0 + \beta_1 X + \cdots + \beta_p X_p = 0$$

where X_i are points on the hyperplane and β_i are the coefficients. Theoretically, the hyperplane can be divided into areas based on the response value if the data is separable.

This would produce linear vectors across the hyperplane customized to the training data and inherently produce perfect predictions. If this is correct, it means that there are an endless number of points on the hyperplane where the support vector might be drawn because there are an infinite number of deviations that will not cause this separation to fail.

Now, the problem is which of these infinite vectors actually represents the real vector. Since the support vector will be drawn at a same distance from the two classes, the solution is to generate an identical margin on both sides of the vector.

Unfortunately, the data is rarely separatable. This problem is resolved by the support vector classifier by allowing a soft margin of error. This indicates that there may be some misclassification or margin violations on the hyperplane (James et al., 2017).

By performing this simplification, one is able to punish the model for violating the margin, by allocating a cost parameter, c: this hyperparameter is crucial ensuring the optimal bias-variance trade-off.

By extending the feature space with kernels, the support vector machines improve on the solutions the support vector classifier provides. The stretched feature space can be compared to the larger feature space to make things simpler and ensure that the vectors are linear. However, in practice, the vectors would not be linear without this stretch; since this operation is very computationally demanding, the solution is to use kernels to simplify the curvature of the vectors. The method will regardless of position on the hyperplane calculates the inner products. For two observations, x_i and x'_i , it can be calculated as $(x_i , x'_i) = \sum_{j=1}^{p} x_i x_{i',j}$, where p is features. This expression is also known as the linear kernel (James et al., 2017). The kernel is usually defined as a vessel that expresses the similarity of two observations on the hyperplane; the kernel is then a function related to the chosen distribution of the decision boundary.

Although there are many different types of kernels, polynomial and radial kernels are the most popular. For instance, every instance of $\sum_{j=1}^{p} x_i x_{i',j}$ can be replaced with the quantity

$$K(x_{i,}x'_i) = (1 + \sum_{j=1}^{P} x_{ij,}x_{i',j} \quad)^d.$$

This is known as polynomial kernel of degree d, where d is a positive integer. The use of a kernel having d > 1 in the support vector classifier leads to a much more flexible decision boundaries compared to the standard linear kernel. Therefore, the standard procedure is to calculate the Euclidean distance between the training and testing points and afterward rank them.

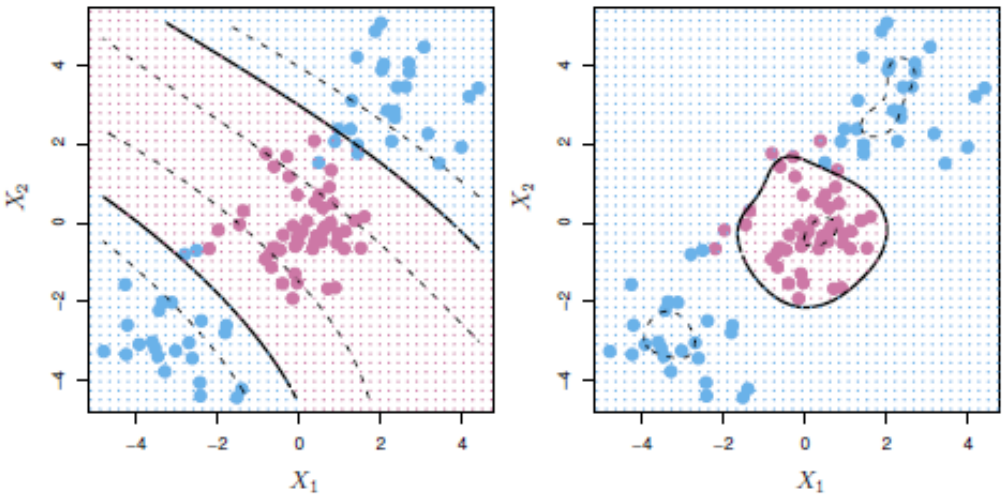

Figure 11: *Graphic representation of polynomial kernel and radial kernel*

A special case for the polynomial kernel occurs if $d = 1$, as the polynomial kernel then is linear, and the method turns into the support vector classifier.

Another popular non-linear kernel is the radial kernel, which takes the form

$$K(x_{i,}x_i') = exp(-\gamma + \sum_{j=1}^{P}(x_{ij,}x_{i',j} \quad))^2,$$

Where γ is a positive constant. For the radial kernel, it is important to calculate the proximity of the observation.

4 Data Analysis

4.1 Data description

In this section will be described the data used for the subsequent empirical analysis and its constituent variables; the dataset has been made public and available on GITHUB (https://github.com/sowide/bankruptcy_dataset).

In the dataset are collected accounting data from 8262 different companies for a total of 78682 observations (the dataset has no missing values or synthetic and imputed added values) in the period between 1999 and 2018 of the American stock market (New York Stock Exchange and NASDAQ); these companies are considered as a good approximation of the health of the American stock market in the time interval. The stock market is dynamic with new companies becoming public every year, changing properties and names, or being removed or suspended from the market as a result of acquisitions or regulatory action. The average number of years available for each company in the dataset is 8 years.

For such organizations have been collected 18 financial variables for each year, usually used for bankruptcy prediction and for the creation of the most important financial ratios. Consideration of accounting data and current market data that may represent the company's liability and profitability is prevalent in bankruptcy prediction.

The variables used are described in the Table 2 below.

Each company has then been labelled (dependent variables) every year depending on its next year's status; as stated in the previous chapters, according to the Security Exchange Commission (SEC) a company in the American market is considered bankrupted in two cases:

- If management of the company files for Chapter 11 of the Bankruptcy Code to "reorganize" its operations, management will still be in charge of day-to-day operations, but all major business decisions will need to be approved by the bankruptcy court.
- If the company's management files for bankruptcy under Chapter 7 of the Bankruptcy Code, all operations are suspended, and the company cease to exist

In both cases, a company is labelled the fiscal year before the chapter filling as "Bankruptcy" (1). Otherwise, the company is considered healthy (0).

Due to this, the dataset allows for the learning of how to anticipate bankruptcy at least a year in advance.

	Variable Name	**Description**
X1	Current assets	All the assets of a company that are expected to be sold or used as a result of standard business operations over the next year
X2	Cost of Goods Sold	The total amount a company paid as a cost directly related to the sale of products
X3	Depreciation and Depreciation	Depreciation refers to the loss of value of a tangible fixed asset over time (such as property. machinery, buildings, and plant). Amortization refers to the loss of value of intangible assets over time.
X4	EBITDA	Earnings before interest, taxes, depreciation and amortization: Measure of a company's overall financial performance alternative to the net income
X5	Inventory	The accounting of items and raw materials that a company either uses in production or sells
X6	Net Income	The overall profitability of a company after all expenses and costs have been deducted from total revenue.
X7	Total Receivables	The balance of money due to a firm for goods or services delivered or used but not yet paid for by customers.
X8	Market Value	The price of an asset in a marketplace. In our dataset it refers to the market capitalization since companies are publicly traded in the stock market
X9	Net Sales	The sum of a company's gross sales minus its returns, allowances, and discounts
X10	Total assets	All the assets, or items of value, a business owns
X11	Total Long-Term Debt	A company's loans and other liabilities that will not become due within one year of the balance sheet date
X12	EBIT	Earnings before interest and taxes
X13	Gross Profit	The profit a business makes after subtracting all the costs that are related to manufacturing and selling its products or services
X14	Total Current Assets	It is the sum of accounts payable, accrued liabilities and taxes such as Bonds payable at the end of the year, salaries and commissions remaining
X15	Retained Earnings	The amount of profit a company has left over after paying all its direct costs, indirect costs, income taxes and its dividends to shareholders

X16	Total Revenue	The amount of income that a business has made from all sales before subtracting expenses. It may include interest and dividends from investments
X17	Total Liabilities	The combined debts and obligations that the company owes to outside parties
X18	Total Operating Expenses	The expense a business incurs through its normal business operations

Table 2: *The 18 numerical bankruptcy features considered in the work.*

The goal of this study is to determine the most precise prediction of a firm's probability of bankruptcy using well-known statistical models and machine learning approaches. Since the number of companies who declare default, each year is frequently a small fraction below the 1% of the firms that are available in the market, there is typically a significant imbalance in bankruptcy datasets. However, there have been some periods when the bankruptcy rate has been greater than usual, such as the Dot-Com Bubble in the early 2000s and the Great Recession in 2007–2008. Table 3 displays the dataset's firm distribution by year.

Year	Total Firms	Bankrupt Firms	Year	Total Firms	Bankrupt Firms
2000	5308	3	2010	3743	23
2001	5226	7	2011	3625	35
2002	4897	10	2012	3513	25
2003	4651	17	2013	3485	26
2004	4417	29	2014	3484	28
2005	4348	46	2015	3504	33
2006	4205	40	2016	3354	33
2007	4128	51	2017	3191	29
2008	4009	59	2018	3014	21
2009	3857	58	2019	2723	36

Table 3: *Firm distribution by year in the dataset.*

4.2 Training and Test Set

The following step in order to apply and use machine learning technique is to split the original dataset in training and test set.

- Training dataset is the initial subset of the original dataset used to teach or train a machine learning algorithm to process information and how to estimate.
- Test dataset is used to evaluate how well the model does with data outside the training set, in never-seen-before data.

A test dataset is a data set that is independent of the training data set, but that follows the same probability distribution as the training set. If a model that fits the training data set likewise fits the test data set well, there hasn't been much overfitting. Over-fitting is typically shown by the training data set fitting the model better than the test data set. Therefore, a test set is a collection of instances used only to evaluate the effectiveness of a fully described classifier.

To do this, the final model is used to predict classifications of examples in the test set. Those predictions are compared to the examples of the true classifications to assess the model's accuracy.

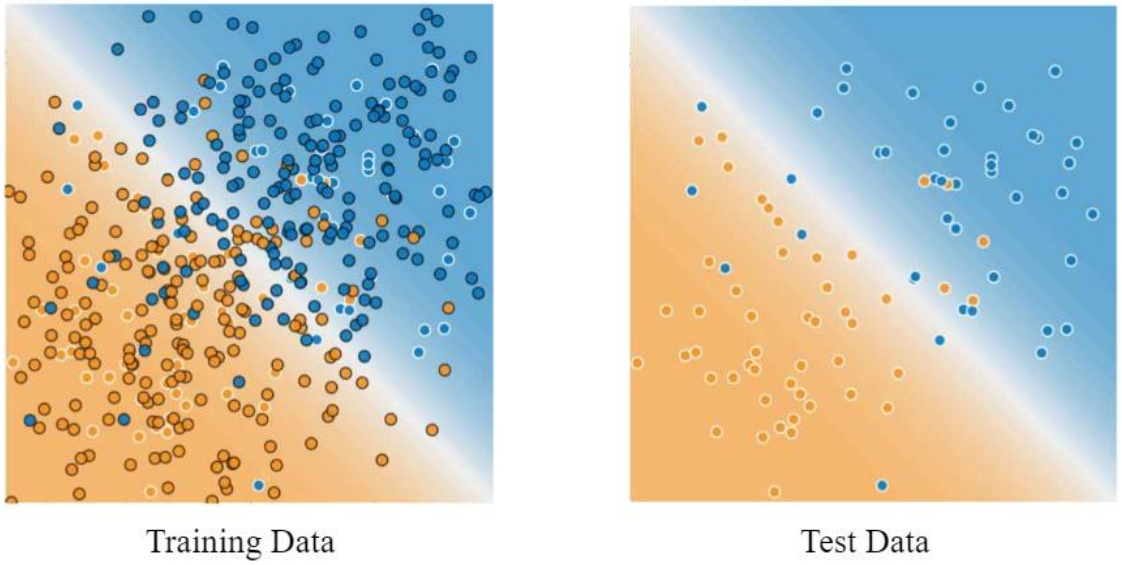

Figure 12: *Graphic representation of a dataset splitted between Training dataset and Test dataset.*

In other words, a common strategy for model validation is data splitting, in which a given dataset is divided into two separate sets for training and testing. The training set is then used to fit the statistical and machine learning models, which are finally tested on the testing set.

The test set, moreover, has two meet two conditions: it has to be large enough to yield statistically meaningful results and it has to be representative of the data as a whole; in other words, the test set must not have different characteristics than the training set.

The next step is to decide how to divide the data into training and test dataset; a commonly used ratio, used in many other studies, is 80:20. Which means that the 80% of the data is randomly selected for training and the remaining 20% for testing. Other ratios as 70:30, 60:40 and even 50:50 are used in literature.

Another topic is that the dataset used for analysis in this work is made of time-series data, so is not possible to randomly splitting. To overcome these issues the data have been separated through time as follows:

- The split is made respecting the temporal order of the observations. “Older” or further data in time have been used to train the models then tested on “younger” or closer in time observations. The main reason is that there are no real-life scenarios where data from the future are used to train a model able to forecast the past. Moreover, data leaking is prevented: leakage would involve making predictions about the future based on the past. In this case all the data before a data point have been taken and verified on the remaining dataset after the data point. This splitting approach allows to consider the changing distribution over time.
- To provide a solution to the literature's unresolved issue of how many years should be taken into account in order to maximize the effectiveness of the bankruptcy prediction model in this work there are three cases. The dataset has been divided in order to obtain different time windows leading to smaller training sets. In the first case (Case 1) observations from 1999 to 2013 represent the training set while the data between 2014 and 2018 are used to test the models. In Case 2 the data point is in 2010, so the models are validated on the data starting from 2011. In the last scenario (Case 3) data between 1999 and

2008 are used for training while data between 2009 and 2018 are used for testing.

The main reason behind these splitting is to recall the main splitting ratios used in the literature: the three cases, indeed, constitute, approximately, the 80:20, 70:30 and 60:40 ratios respectively. However as stated before in the time-series framework is not possible to divide the data with absolute precision: the splits have then been made rounded to the closest data point.

The number of observations for each case are shown in the table below:

	Case 1 80:20	**Case 2 70:30**	**Case 3 60:40**
Training Set	62896	52414	45046
Test Set	15789	26268	33636

Table 4: *Firm distribution after splitting the dataset between training set and test set in the three scenarios.*

4.3 Imbalanced Classification

Another issue in bankruptcy prediction is that the problem belongs to the category of imbalanced classification; in this case the final proportion is of 78073 non-bankrupt and 609 bankrupt observations. So, the dataset is highly imbalanced: the smaller class represents less than the 1% of the total dataset.

The distribution may range in severity according to the percentage of the minority class; therefore, the degree of imbalance can be mild if the proportion of the minority class varies between 20% and 40% of the dataset, moderate between 1% and 20% and extreme if less than 1% of the dataset belongs to the smaller class. The latter is the case for bankruptcy prediction as the smaller class represents less than the 1% of the total assets. Bankruptcy is usually a rare event in a period of normal market conditions and this bias in the training dataset can influence many machine learning algorithms, that will perform poorly and need to be modified to prevent always predicting only the majority class since the distribution of the classes is not balanced: this is problematic because forecasts are often more crucial for the minority class. Furthermore, measures like classification lose their relevance, and alternative techniques, like ROC, area under curve, that will be explained later, are needed to assess predictions on unbalanced examples.

One possible approach to combat this challenge is *Random Sampling*. Since Random Sampling makes no assumptions about the data when it is used, it is characterized as a *naive technique*. To lessen the impact of the data on our machine learning system, it entails developing a new altered version of our data with a new class distribution.

Importantly, the training dataset is the only one to which the class distribution has been altered. The goal is to influence the fit of the models. The test or holdout dataset used to assess a model's effectiveness does not require resampling.

There are two main possibilities to perform random resampling; both techniques can be used for two-class (binary) classification problems and multi-class classification problems with one or more majority or minority classes and both methods have advantages and disadvantages:

- **Random Oversampling**: Random oversampling involves randomly duplicating examples from the minority class and adding them to the training dataset. Examples are randomly selected with replacement from the training dataset.

  This implies that samples from the minority class can be picked from the initial training dataset, added to the new, “more balanced”, training dataset, and then returned or “replaced” in the initial dataset, allowing them to be selected once more. This technique can be effective for those machine learning algorithms that are affected by a skewed distribution and where multiple duplicate examples for a given class can influence the fit of the model. This might include algorithms that iteratively learn coefficients, like artificial neural networks that use stochastic gradient descent. It can also affect models that seek good splits of the data, such as support vector machines and decision trees.

  Adjusting the desired class distribution can be helpful. Some algorithms may overfit the minority class as a result of trying to find a balanced distribution for a dataset with a marked imbalance, which increases generalization error. Better performance on the training dataset may result, however worse performance on the holdout or test dataset may result.
- **Random Undersampling**: Random undersampling involves randomly selecting examples from the majority class to delete from the training dataset.

  This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, for instance, an equal number of samples for each class, is reached; this approach may be more suitable for those datasets where there is a class imbalance although a sufficient number of examples in the minority class, such a useful model can be fit.

  Undersampling has the drawback of removing samples from the majority class that might be essential, useful, or even crucial for fitting a strong decision boundary. There is no method to identify or save "excellent" or more information-rich instances from the majority class since examples are arbitrarily eliminated.

To put it another way, both oversampling and undersampling imply adding a bias by choosing more samples from one class than from another, to make up for an imbalance that is either already present in the data or would likely arise if a completely random sample were obtained.

A third way to overcome imbalanced classification can be obtained by combining random oversampling and undersampling: in some case, combining the two random techniques can lead to overall better performance than when techniques are used alone. The idea is that is possible to apply a modest amount of oversampling to the minority class in order to lessen the bias on the majority class instances, while performing a small amount of oversampling on the minority class in order to improve the bias to the minority class examples.

4.4 SMOTE

Random sampling methods have been described above. One of the most widely used and known technique, that will be used also in the subsequent analysis, is *Synthetic Minority Oversampling Technique* (SMOTE), described for the first time by Nitesh Chawla, et al. in their 2002 paper and inspired by a technique that proved successful in handwritten character recognition (Ha & Bunke, 1997).

It aims to balance the distribution of classes by randomly increasing minority class samples and by replicating them. In other words, SMOTE creates new minority instances by combining minority instances that already exist. For the minority class, it creates virtual training records using linear interpolation. For each example in the minority class, one or more of the $k-$ nearest neighbors are randomly chosen to serve as these synthetic training records. Following the oversampling procedure, the data is rebuilt and can be subjected to several categorization models.

More specifically, SMOTE algorithm follows the following steps:

1. The algorithm takes as input a dataset with a minority class and a majority class.
2. SMOTE identifies the instances belonging to the minority class in the dataset.

3. For each minority instance, SMOTE selects its k − nearest neighbors (that can be selected by the user).
4. SMOTE generates synthetic samples for each chosen minority instance by creating new instances along the line segments joining the minority instance to its k nearest neighbors. SMOTE chooses one or more of the k closest neighbors at random to create a synthetic sample, then computes the difference between the feature values of the minority instance and the chosen neighbor. It then adds the result to the feature values of the minority instance after multiplying this difference by a random number between 0 and 1. For each feature in the dataset, this procedure is repeated.
5. The synthetic samples generated in the previous step are added to the original dataset, resulting in an augmented dataset.
6. Steps from 3 to 5 are repeated until the desired level of balance between the minority and majority class is achieved or a predetermined number of synthetic samples have been generated.

To summarize, SMOTE is an algorithm that adds artificial data points to the actual data points to accomplish data augmentation. SMOTE can be viewed as an improved form of oversampling or as a particular data augmentation procedure. With SMOTE, is possible to avoid producing duplicate data points and instead produce synthetic data points that are marginally different from the original data points.

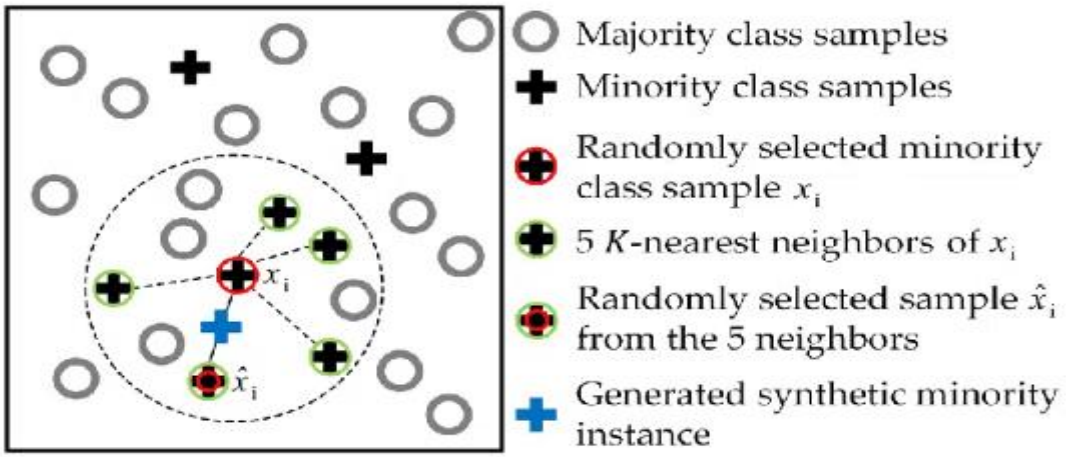

Figure 13: *Graphic representation of SMOTE algorithm.*

4.4.1 SMOTE Results

After splitting the observations into three cases the following step is to analyse the distribution of bankrupt (1) and non-bankrupt (0) companies in every case in order to understand the severity of the imbalance between the minority and the majority class.

The number of failed and alive observation is summarized as follows:

	Case 1 80:20	**Case 2 70:30**	**Case 3 60:40**
Non-Bankrupt (0)	62439	52036	44726
Bankrupt (1)	457	378	320

Table 5: *Bankrupt and Non-Bankrupt firm distribution in the three cases.*

As already stated, bankruptcy is a rare event and in Table 5 are shown the number of Bankrupt and Non-Bankrupt firms in the three cases that will be analysed later.
To overcome this issue SMOTE method has been used to generate new minor class instances based on the dataset used and to reduce the observations in the majority class, still maintaining a sufficient number of samples to properly train the model but reducing, at the same time, class imbalance.
The objective was to obtain an imbalanced classification close to 1:10 ratio. The new training datasets after the use of SMOTE algorithm have the following numerosity.

	Case 1 80:20	**Case 2 70:30**	**Case 3 60:40**
Non-Bankrupt (0)	10968	12852	10880
Bankrupt (1)	914	1134	960

Table 6: *Bankrupt and Non-Bankrupt firm distribution in the three cases after SMOTE resampling.*

In conclusion, the datasets are still imbalanced so it can cause the model to pay more attention to the majority class because it has more instances than the minority class. For

this purpose, all the machine learning models used in this work have been improved by assigning weights to each sample based on its class. So, giving more weight to the minority observations during training the model will pay the same attention also to the minority class; this can help the model to learn the main characteristics of the smaller class and improve its performance in it.

Specifically, the weights were applied inversely proportional to the distribution of the class.

4.5 Performance Measures

4.5.1 Confusion Matrix

The focus of this work is on the prediction accuracy of the methods. In the next chapter the bankruptcy prediction will be implemented as a binary prediction task and when the results on the classification methods are obtained it is crucial to review and compare which methods and which conditions produce the most accurate predictions.

As shown in the previous sections, the dataset has been initially splitted into a train set and a test set. Each model is trained on the train test and then tested using the test set. The performances of machine learning algorithms are then typically evaluated by a confusion matrix and all other metrics that will be covered later are based on this performance statistic.

A confusion matrix is represented as below:

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Figure 14: *Confusion matrix scheme.*

The columns are the Predict class and the rows are the Actual class. In the bankruptcy framework the following quantities can be considered for the default prediction:

- **True Positive (TP)**: The number of actually defaulted companies that have been correctly predicted as bankrupted.
- **True Negative (FN)**: The number of actually healthy companies that have been correctly predicted as healthy.
- **False Positive (FP) (Type I error)**: The number of actually healthy companies that have been wrongly predicted as bankrupted by the model.
- **False Negative (FN) (Type II error)**: The number of actually defaulted companies that have been wrongly predicted as healthy firms.

From this confusion matrix, some important performance measure can be derived [Johnson, 2019]:

- **Accuracy** = $\frac{TP+TN}{TP+TN+FP+FN}$
- **Precision** = $\frac{TP}{TP+FP}$
- **Recall** = $\frac{TP}{TP+FN}$

Accuracy gives the percentage of correctly predicted classifications.
With imbalanced data, this performance metric is not optimal because the negative class will predominate. Since bankruptcy is a rare event, the use of classification accuracy might be misleading to evaluate a model's performance since it is possible to still obtain a high accuracy value from the prediction of the majority class, in this case the "Non-Bankrupt". This means that the model correctly predicts the majority class while fails to predict the minority class most of the time (which is usually the main class to look at) and this creates an illusion of the model's "efficiency" in predicting both classes. Indeed, for a financial institution, the cost of false negatives is significantly higher than the cost of false positives.
Because of this, precision and recall are considered to be more appropriate for this type of data. The level of precision shows the percentage of samples that were accurately identified as positive. Recall, on the other hand, indicates the proportion of positive

samples that were correctly classified positive. Therefore, this is also referred to as the *True Positive Rate (TPR)* or the sensitivity of the model.

Furthermore, another definition to measure the prediction accuracy of these classification methods is the specificity; it is the ratio between the false positives and the actual negatives, which is also known as the *False Positive Rate (FPR).*

The *False Positive Rate* and the *False Negative Rate (FNR)* have the following mathematical definition:

- False Positive Rate = $\frac{FP}{TN+FP}$.
- False Negative Rate = $\frac{FN}{TP+FN}$.

F1 Score is another measure to assess the accuracy of the model. It can be used to measure precision and recall at the same time and for model comparison, since it can be problematic to measure two models with high precision and low recall or vice versa.

The F1 Score is the harmonic mean of the precision and recall defined above:

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either precision or recall are zero.

Below some advantages of F1-score:

- Very small precision or recall will result in lower overall score. Thus, it helps in balancing the two metrics.
- F1-score can assist in balancing the metric between positive and negative samples when the positive class is the one with fewest examples.
- It combines many of the other metrics into a single one, capturing many aspects at once.

The main pitfall, as it will be possible to see subsequently, is that usually F1-score may remain low when the test dataset is highly imbalanced, because in such situation it is difficult to get high recall on the rare class with a reasonable precision.

4.5.2 ROC-Curve AUC Score

The Receiver Operating Characteristic (ROC) curve is a graphical figure that displays how well a binary classifier performs when the threshold is changed.

The curve, which represents the power as a function of the Type I error of the decision rule (based on a sample of the data), is produced by plotting the true positive rate (TPR, also known as sensitivity or recall) against the false positive rate (computed as $1 - TPR$). In the graph X-axis shoes the specificity, and the Y-axis is the sensitivity. When a classification is performed by a method, it provides a number between 0 and 1 (or 0% and 100%).

These samples are assigned to the positive and negative classes according to some threshold. The TPR and the FPR are shown in the ROC-curve at various thresholds.

The frequency of false positives and true positives falls as this threshold rises because fewer objects are classified as positive. Therefore, these ROC-curves can be used to compare the effectiveness of various classification techniques; however, it is not always convenient to just compare these graphs.

The Area Under the Curve (AUC) is an additional metric of this curve that allow to calculate and compare the absolute performance of the methods [Bradley, 1997].

An example of a ROC-curve and of the AUC are shown below:

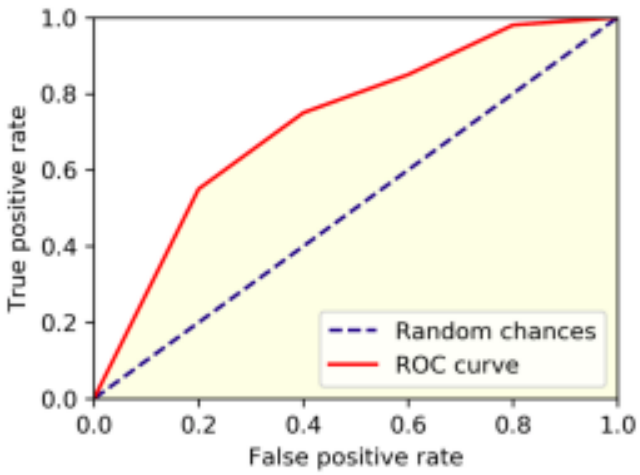

***Figure 15**: Example of ROC curve.*

The ROC-curve always starts in the origin, where all the samples are classified as negative. As a result, there are no samples that are both true positive and false positive. The position (1,1) obviously reflects the situation in which all samples are classified as positive. The point (0,1) is quite interesting since it denotes a circumstance where classification is perfect. Therefore, the performance is better when the ROC-curve is closer to this point. The blue striped line denotes the random guessing approach used to determine a sample's class. The ROC-curves should be at least above this line as a result. As was already established, the model's classification abilities are measured by the area under the curve. This region is exactly equal to one in case of perfect classification.

The ROC-curve shifts in the direction of the random line when the classifier's performance is less than ideal. As a result, the AUC moves toward the appropriate region of 0.5. In conclusion, the AUC ranges from 0.5 to 1. The closer this metric is to one, the better the classification performance of the corresponding method [Fawcett, 2006].

5 Model fitting and Main Results

In this chapter will be shown the result obtained by applying the models described above to the "American Bankruptcy" datasets.

Specifically, will be reported the model fitting and the main performances measures for all the three cases: this will allow to find the best model, compare the different techniques applied to the data and analyse their main features, positive and negative results obtained.

5.1 Case 1

	TP	FN	FP	TN	Accuracy	F1-score
Lasso	82	70	3366	12268	0.7823	0.046
Tree	112	40	4466	11168	0.715	0.047
Random Forest	**134**	**18**	5125	10509	0.6742	0.049
Bagging	125	27	4873	10761	0.6896	0.049
XGBoost	125	27	4943	10691	0.6852	0.047
CatBoost	129	23	5403	10231	0.6563	0.045
SVM	116	36	4606	11028	0.7059	0.048

Table 7: *Confusion Matrix, Accuracy and F1-score of the models in Case 1.*

	AUC Score	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Recall
Lasso	0.662	0.539	0.785	0.024	0.994	0.539
Tree	0.726	0.737	0.714	0.024	0.996	0.737
Random Forest	**0.777**	0.882	0.672	0.025	0.998	**0.882**
Bagging	0.755	0.822	0.688	0.025	0.997	0.822
XGBoost	0.773	0.822	0.684	0.024	0.997	0.822
CatBoost	0.752	0.849	0.849	0.998	0.023	0.849
SVM	0.734	0.763	0.705	0.025	0.997	0.763

Table 8: *Model comparison of the results of default prediction in Case 1.*

5.1.1 Case 1 Lasso

Coefficient	Estimate	Coefficient	Estimate
Intercept	507.183356	**Retained Earnings**	-0.002763
Current Liabilities	0.066556	**Total. OP. Expenses**	-0.00295
Total Liabilities	0.020952	**Total Assets**	-0.0118
Gross Profit	0.009957	**D&A**	-0.0197
Total Revenue	0.0086	**Long Term Debt**	-0.031031
Net Sales	0.00344	**Net Income**	-0.045367
Market Value	0.001782	**EBIT**	-0.050311
EBITDA	0	**Inventory**	-0.103
Current Assets	0	**Total Receivables**	-0.103214

***Table 9**: Coefficients of Lasso Regression in Case 1.*

In Table 9 can be appreciated one of the main features of lasso regression: no coefficients are shown for the variables Current Assets and EBITDA because the lasso regression shrinked the coefficients to zero. This means that such variables are completely dropped from the model because they are not influential enough and those variables are not important in predicting the target variable.

The interpretation of the coefficients is similar to interpreting coefficients in the logistic regression: the magnitude of a non-zero coefficient indicates the strength of the relationship between the predictor and the probability of the binary event. Table 9 suggests that an increase in Current Liabilities and Total Liabilities leads to an increase in the probability of default; while, on the other side, firms with a higher amount of Total Receivables and Inventory are more likely to survive.

From the Table 7 can be noticed that lasso regression provides the better Accuracy; however, such result cannot be considered relevant as the models are fitted to an imbalance dataset and “Non-Bankrupt” is the major class. Indeed, lasso regression has good performance in the prediction of “True Negative” values, but, on the other side it has poor performance, compared to other models, in the most important topic: the prediction of bankrupt firms, which is actually the purpose of this work.

Moreover, its AUC is the lowest among the all the models: proving that lasso regression technique is not efficient for such classification problem. Also, the recall is very low; so for the lasso model there are higher probabilities to commit a II Type error: label a company as “Non-bankrupt” when it actually is.

5.1.2 Classification Tree

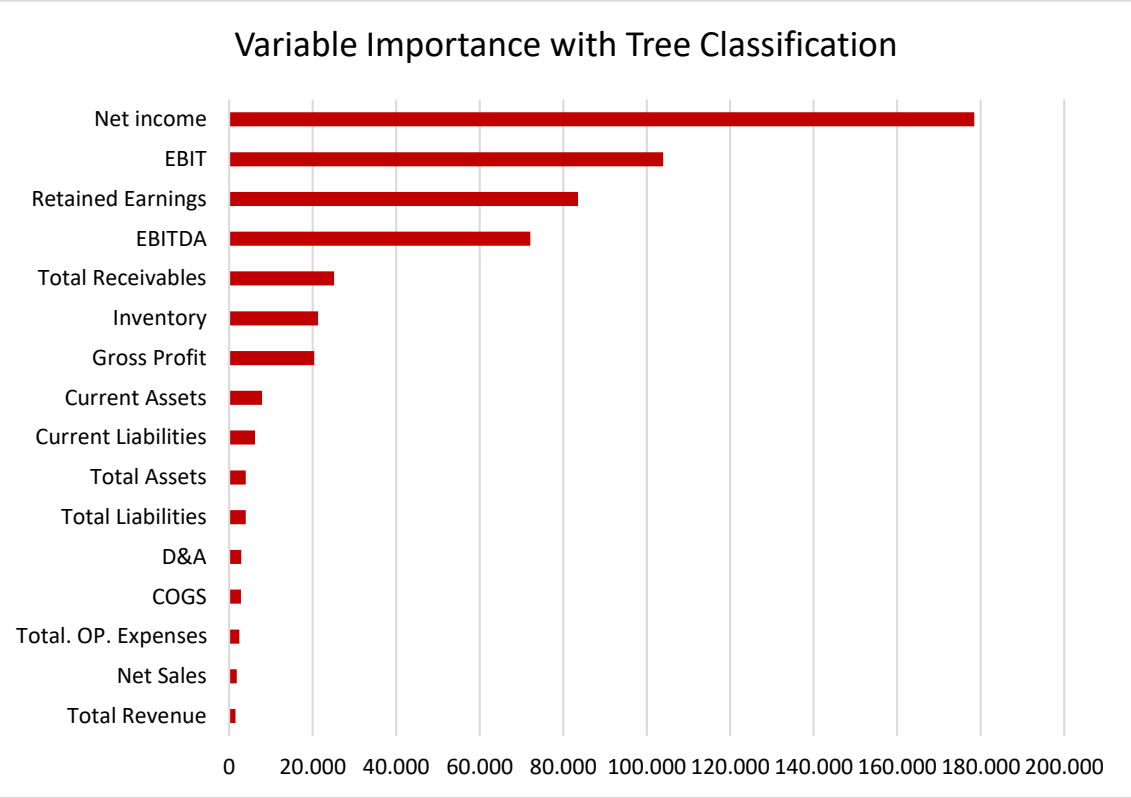

***Figure 16**: Variable Importance of Classification Tree Case 1.*

The second model used for the bankruptcy forecast is the classification tree.

Figure 18 shows the importance of each variable. Variable importance is determined by calculating the relative influence of each variable: whether that variable was selected to split on during the tree building process, and how much the squared error (over all trees) improved (decreased) as a result. The most important features used by the machine learning algorithm are by far Net Income, EBIT, Retained Earnings and EBITDA while the less important are Total Revenues, Net Sales, Total Operating Expenses, COGS, D&A.

The second plot shows the actual three built by the classification tree algorithm.

Starting from the root node (the top of the graph):

- At the top there is the overall probability to fail (1). It also shows the proportion of companies that actually failed: 50% of companies failed.
- This node asks whether the Net Income is greater or equal than "-1.2". If no, the tree goes down to the root's right child node and the probability of default is now 69%.
- The same reasoning can be applied going down to the subsequent nodes in order to understand what features impact the likelihood of survival.

Analysing the performance measures is possible to observe that the classification tree provides general improvements compared to the lasso regression. This technique, indeed, is able to spot a higher number of "True Positive" cases and to reduce the number "False Negative" predictions. Moreover, there is a clear improvement of the AUC-score, which is now equal to 0.726. So, it belongs to the interval 0.7-0.8 for which a value of the AUC score is considered as acceptable.

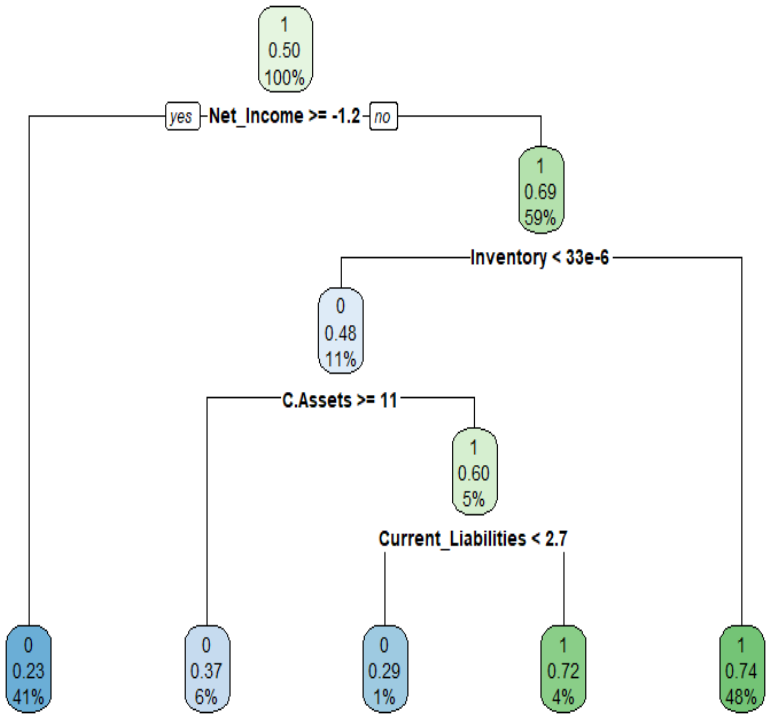

Figure 17: *Graphic representations Classification Tree Case 1.*

5.1.3 Random Forest

According to the performance measures random forest seems to be the powerful machine learning procedure. It provides the highest AUC-score, it maximises the number of "True Positive" predictions (134), and, at the same time, it allows to minimize the number of "False Negative" forecasts (18); for these reasons the random forest has also the highest recall compared to all the other models.

In Figure 20 are shown the Mean Decrease Accuracy and Mean Decrease Gini of Random Forest in Case 1. In the left panel, the most important variable is the Net Income, followed by Current Liabilities, Market Value, and Inventory which have almost the same importance, while the less important variables seem to be Net Sales and Total Revenue.

In the right panel Net Income is again the variable that contributes the most to the homogeneity of the nodes and leaves in the random forest and the second most important one is the Market Value. As in the Mean Decrease Accuracy the less relevant variables are Net Sales and Total Revenue.

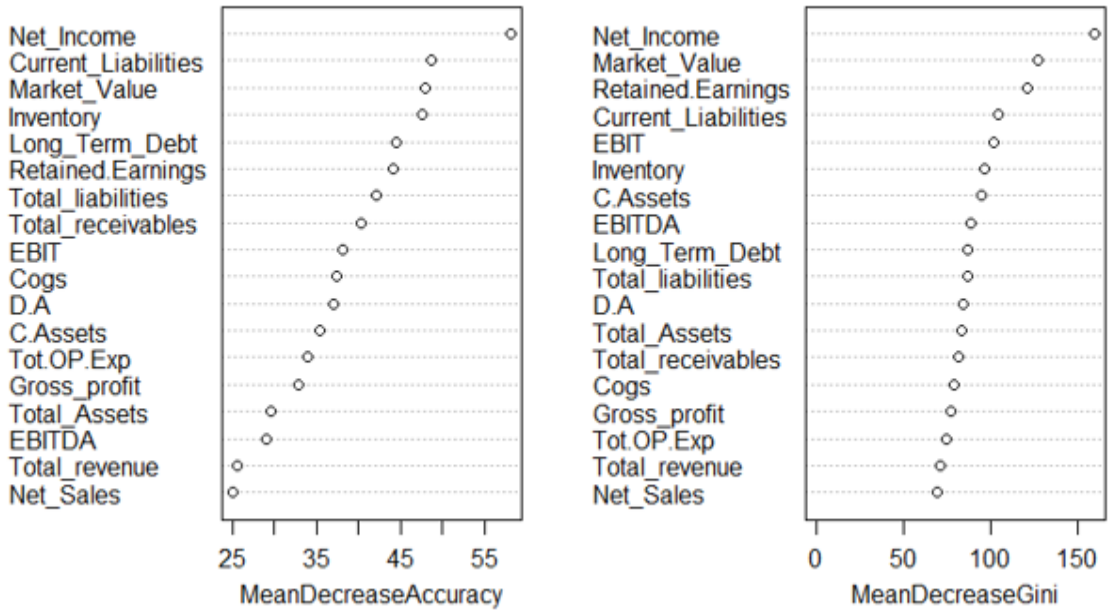

Figure 18: *Graphic representations of Mean Decrease Accuracy and Mean Decrease Gini of Random Forest in Case 1.*

5.1.4 Bagging

The feature interpretation is the same as in the random forest context: the plot related to the Mean Decrease Accuracy and the Mean Decrease Gini are provided. Also, in the bagging framework the three most important variables are Net Income, Market Value and Retained Earnings. While in the first case the less important are by far EBITDA, Net Sales and Total Revenue and, the last two, are the less important also in the determination of the Gini coefficient.

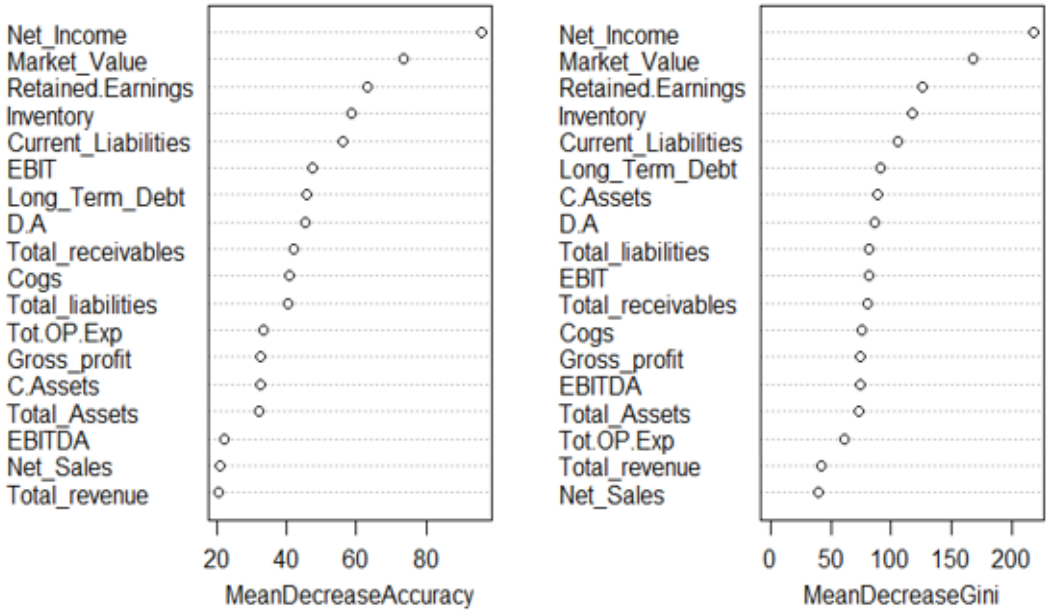

Figure 19: *Graphic representations of Mean Decrease Accuracy and Mean Decrease Gini of Bagging in Case 1.*

As expected, the performance measures provide results which are improved compared to the classification tree but not as good as the random forest; indeed, bagging has a Recall equal to 0.822, which is in the middle between the previous tree-based models, and it is able to catch 125 “TRUE POSITIVE” and only 27 “False Negative” firms.

5.1.5 Boosting

Feature	Gain	Cover	Frequency
Net Income	0.619	0.255720770	0.24
Retained Earnings	0.122	0.146559190	0.11
Current Liabilities	0.072	0.156166149	0.16
Inventory	0.070	0.110417224	0.11
Market Value	0.052	0.180746994	0.19
Long Term Debt	0.045	0.115934235	0.12
EBIT	0.011	0.037892783	0.04
Total Liabilities	0.006	0.019186743	0.02
Net Sales	0.002	0.009272875	0.01

***Table 10**: Importance data table of XGBoost in Case 1.*

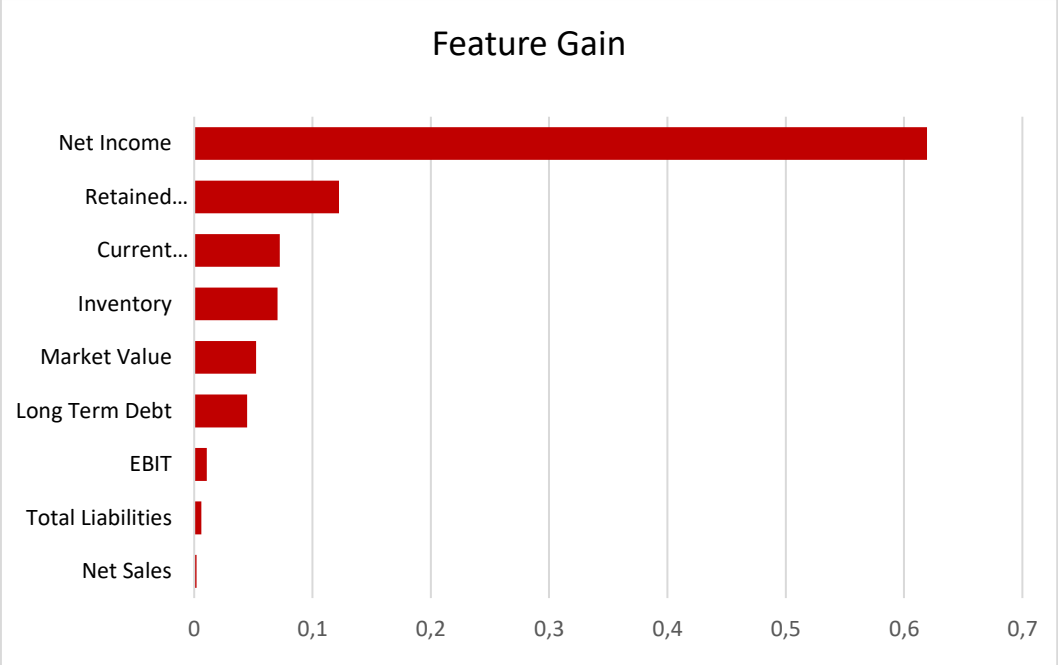

***Figure 20**: Feature importance of XGBoost model in Case 1.*

Boosting procedures provide good general results.

The Gain is the most relevant attribute to interpret the relative importance of each feature and Net Income is the most important variable in building the model; while for

CatBoost procedure the most important variable is Net Income while D&A, Total Revenue, Net Sales, Total Assets, Current Assets, D&A, Gross Profit are the less important (Figure 22).

Looking at Table 7 XGBoost has an AUC-score equal to 0.773, which is second only to the performance of random forest. This machine learning technique is also able to spot a good number of "True Positive" cases and only few "False Negative". Similarly, CatBoost has a good AUC-score (0.752), even though lower compared to the one of XGBoost; but it is able to catch 129 failed company and only 23 false negative cases; also in this case this performance has been overcome only by Random Forest.

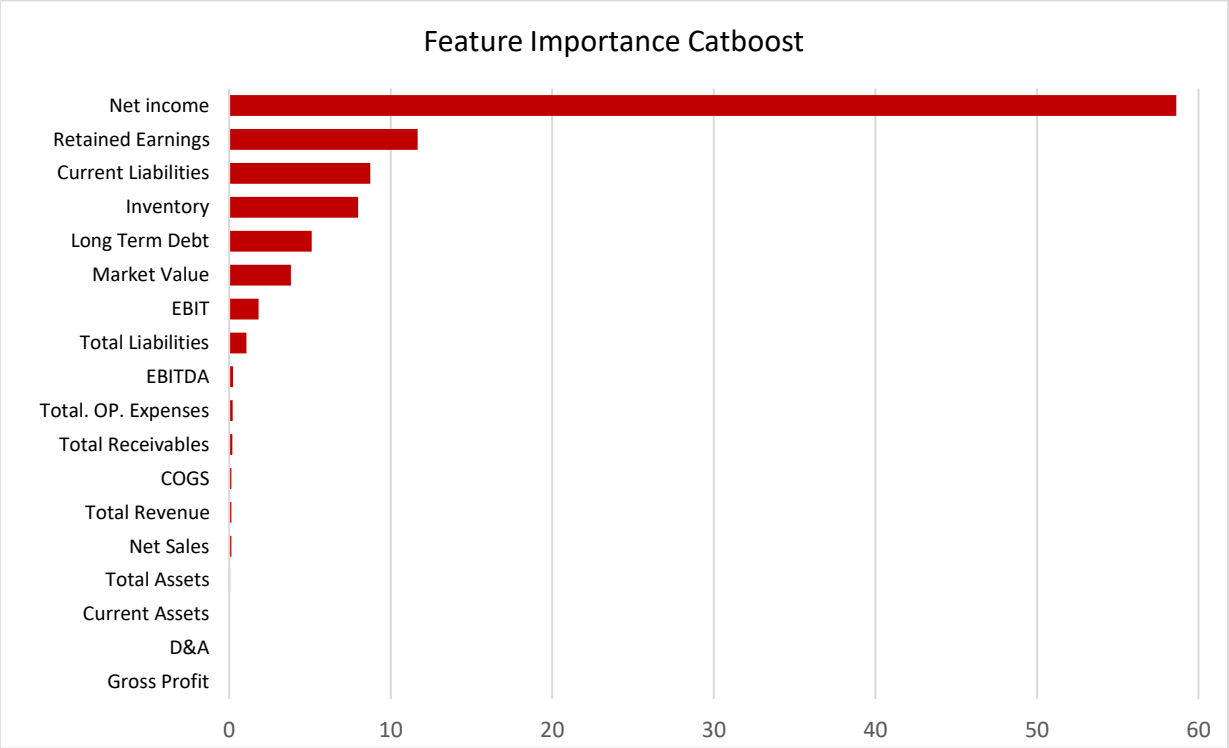

***Figure 21**: Feature importance of CatBoost in Case 1.*

5.1.6 SVM

The last model, to conclude the first case analysis is the Support Vector Machine.

This machine learning technique provides quite good results in terms of AUC-Score and "True Positive" and "False Negative" predictions and so also recall is above 0.75. The number of "True Positive" predictions is equal to 116 against 36 "False Negative". In general, they can be considered as good results, but it is clear that the SVM is not performing as random forest or boosting procedures.

5.2 Case 2 and Case 3

	TP	FN	FP	TN	Accuracy	F1-score
Lasso	114	117	4654	21383	0.8184	0.06
Tree	177	54	6827	19520	0.738	0.049
Random Forest	203	28	7315	18722	0.7205	0.048
Bagging	184	47	6753	19284	0.7411	0.046
XGBoost	206	25	9416	16621	0.6406	0.042
CatBoost	**208**	**23**	9650	16387	0.6318	0.041
SVM	185	46	7023	19014	0.7309	0.048

Table 11: *Confusion Matrix, Accuracy and F1-score of the models in Case 2.*

Colonna1	AUC Score	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Recall
Lasso	0.675	0.49	0.821	0.024	0.995	0.49
Tree	0.752	0.766	0.738	0.025	0.997	0.766
Random Forest	**0.799**	0.879	0.719	0.027	0.998	0.879
Bagging	0.769	0.797	0.741	0.026	0.998	0.797
XGBoost	0.765	0.892	0.638	0.021	0.998	0.892
CatBoost	0.765	0.900	0.9	0.9986	0.021	**0.900**
SVM	0.766	0.8	0.73	0.026	0.998	0.8

Table 12: *Model comparison of the results of default prediction in Case 2*

	TP	FN	FP	TN	Accuracy	F1-score
Lasso	116	173	3445	29902	0.8924	0.06
Tree	222	67	8720	24267	0.7388	0.049
Random Forest	237	52	9518	23829	0.7155	0.048
Bagging	220	69	9042	24305	0.7291	0.046
XGBoost	225	64	10163	23184	0.696	0.042
CatBoost	**253**	**36**	11850	21497	0.6466	0.041
SVM	212	77	8387	26960	0.7484	0.048

Table 13: *Confusion Matrix, Accuracy and F1-score of the models in Case 3.*

	AUC Score	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Recall
Lasso	0.649	0.401	0.897	0.033	0.994	0.401
Tree	0.753	0.769	0.739	0.025	0.997	0.769
Random Forest	**0.767**	0.82	0.715	0.024	0.998	0.82
Bagging	0.745	0.761	0.729	0.024	0.997	0.761
XGBoost	0.738	0.778	0.695	0.022	0.997	0.778
CatBoost	0.76	0.875	0.8754	0.998	0.021	**0.875**
SVM	0.741	0.734	0.784	0.025	0.997	0.734

Table 14: *Model comparison of the results of default prediction in Case 3.*

For the second and the third case the performance measures for each model will be briefly reported.

- **Lasso regression**: in "Case 2" and in "Case 3" the lasso regression is the model with the worst performances. The model is good only in the prediction of the "True Negative", so the companies that are actually not bankrupt but provides a high number of "False Negative" prediction thus proving its poor performance to identify companies in trouble, which is actually the main purpose of the analysis. Moreover, the AUC-score is equal to 0.675 and 0.649 in the second and in the third case respectively: below 0.7 which is the minimum threshold to consider a model as acceptable.
- **Classification Tree**: in the second scenario Net Income, EBIT, Retained Earnings and EBITDA are still the four most important variables. The performance measures show that in the second the classification tree improves its performances: the AUC is higher compared to the first scenario and as it is higher than 0.75 which means that the model provides an acceptable discrimination. Moreover, also the Recall is a bit higher compared to the first case.
- **Random Forest:** as in the first case, it is still the best model. The AUC-score is almost 0.8, which is considered as an excellent classification, and it is the highest rate reached by all the models in the different temporal windows. Above all is important to highlight that random forest predicts the highest number of companies that are bankrupted ("True Positive") but, at the same time, it makes

few “False Negative” mistakes. For these reasons the recall is very high, close to 0.88. About feature importance: The mean decrease Gini provides similar results a Case 1: Net income, Market Value and Retained Earnings are still the most important variables to decrease the node impurity and in the contribution to the homogeneity of the nodes and leaves. In this case Net Income seems less important: its value is closer to the ones of Market Value and Retained Earnings, having then less important impact in each split.

- **Bagging**: as in the first case the most important variables, both in Mean Decrease Accuracy and Mean Decrease Gini, are Net Income, Market Value, Retained Earnings and Inventory; compared to “Case 1” in Mean Decrease Accuracy there is a shorter distance among these variables, while now Market Value has almost the same relevance as Net Income in the contribution of the homogeneity of each node. The less relevant variables seem to be again Total Revenues and Net Sales. Bagging method improves its performances in terms of AUC as it is higher compared to Case 1, while the Recall is a bit lower but still acceptable being close to 0.8. In the last scenario it provides general acceptable performances, but not good as the two best models: random forest and CatBoost.
- **SVM**: reducing the size of the training set and increasing the test set the SVM shows a good improvement. The AUC is greater than 0.75 and the model now outperforms the bagging technique in terms of “True positive”, “False Negative” and so the Recall which is equal to 0.8. Unfortunately, in “Case 3” Support Vector Machine is the worst model after lasso regression.
- **XGBoost:** in second and third scenario the most important variable is again Net Income. In “Case 2” the model provides excellent results and it is second only compared to CatBoost in terms of True Positive, False Negative and then Recall (206, 25, 0.892 respectively). While in Case 3 it performs slightly worse as it is outperformed by random forest and there is a higher difference compared to Catboost.
- **CatBoost:** as the previous boosting procedure the most relevant feature is always Net Income. Table X shows the True Positive, False Negative and Recall of CatBoost are the best results. In the third case CatBoost is providing really good performances in terms of AUC (0.76), and especially for “True Positive” and

“False Negative”, 253 and 36 respectively. These are the best values and Catboost outperforms all the other machine learning techniques, included random forest that is still the best model in terms of AUC-Score. Catboost shows good improvements especially if compared to XGBoost; it predicts 253 True Positive cases against 225 but at the same time, the number of False Negative forecasts is almost the half (36 against 64). To conclude CatBoost provides the best two values of Recall in the second (0.9) and in the third scenario (0.875).

6 Conclusion and further works

6.1 Conclusion

In this thesis the main characteristics related to bankruptcy were initially described.

Although the event is quite uncommon and typically associated with overall poor performance of the company, there may also be other factors that contribute to the event's remarkable impact on the business and the market.

There are several outside factors (like financial crises) that might drastically worsen a company's financial situation. This is why we require efficient models that can generate precise and trustworthy forecasts. According to the investigation, using statistical models with machine learning approaches to predict bankruptcy was successful.

In this work were investigated the performance of several machine-learning techniques concerning predicting bankruptcy in the American stock market. The models were compared over different tasks: the default prediction using time series accounting data and the detection of the variables that can indicate financial distress or poor performance of a company. These tasks were performed using a dataset with 8262 companies in the period between 1999 and 2018 and the machine learning techniques have been trained and tested in three different time windows characterized by a progressive reduction of the train sample and widening of the test set. The first obstacle in the evaluation of the models was the high class imbalance, which required the use of the SMOTE in order to balance the two classes. For this reason, the performances of the models could not be compared using accuracy, hence Area Under the Curve was used as a common metric to evaluate the bankruptcy prediction task supported by the number of “True Positive” and “False Negative” predictions.

In terms of AUC-score random forest offers the best predictions in all the three cases achieving the peak in the second temporal windows with a AUC of 0.799, followed by boosting techniques which are characterized by two opposite behaviours: XGBoost provides really good results in the first case and it gets worse with the decrease of the

train window; on the other hands, CatBoost in the second scenario enhances its performance which remains almost constant also in the third case.

Boosting procedures, and especially Catboost, have the capacity to catch the higher number of bankrupted firms and, at the same time, minimize the “False Negative” predictions. The highest difference with the other models has been detected in the third case where CatBoost shows the best predictions.

6.2 Further works

A wide range of models have been analyzed in this thesis. However, the world of machine learning is large and in continuous expansion and improvement; this aspect, added to the characteristics of the field of study, leave ample room for insights and the development of further analyses. To name a few: in this case only financial variables were considered; but it could be interesting to introduce macro-economic variables that could help explain poor performance or well-being in certain areas; another aspect to be investigated could be the correlation between the maturity in terms of age of the company and its probability of failure.

As for machine learning algorithms; first of all the classes have been rebalanced using SMOTE; however, today's literature offers several and perhaps more advanced rebalancing methods which could then improve the forecasting performance of the models. Furthermore, the models have been trained trying to place them on the same level as much as possible and therefore to compare them on an "equal basis". For example, the same weights were applied to all models to rebalance the classes, but it is not excluded that different weights could have led to an improvement in predictive performance.

To conclude, therefore, to improve the results obtained, future research could focus on the use of different rebalancing methods and on the deepening of the specific parameters of each model.

Bibliography

Agostini M. (2018) The Role of Going Concern Evaluation in Both Prediction and Explanation of Corporate Financial Distress: Concluding Remarks and Future Trends. In: Corporate Financial Distress. Palgrave Pivot, Cham

Altman, E., (1968). "Financial ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy". Journal of Finance, September, Volume 23, pp. 589-609.

Altman E.I, Hotchkiss E. (2010). Corporate Financial Distress and Bankruptcy: Predict and Avoid Bankruptcy, Analyze and Invest in Distressed Debt. 3 Ed. Vol. 289 di Wiley Finance. John Wiley & Sons

Azzalini, A. and Scarpa, B. (2012). Data analysis and data mining: An introduction. OUP USA.

Barboza, F., Kimura, H., and Altman, E. I. (2017). Machine learning models and bankruptcy prediction. Expert Syst. Appl., 83:405–417.

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

Bonsall, S. B., Z. Bozanic, and P. E. Fischer, (2013), What do management earnings forecasts convey about the macroeconomy? Journal of Accounting Research 51, 225–266.

Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. Computational Intelligence, 20(1):18–36.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29:1189–1232.

Hasan, M. M., Hossain, M., & Habib, A. (2015). Corporate life cycle and cost of equity capital. Journal of Contemporary Accounting and Economics, 11(1), 46–60.

Hubert Ooghe, Sofie De Prijcker, (2008) "Failure processes and causes of company bankruptcy: a typology", Management Decision, Vol. 46 Issue: 2, pp.223-242

James, G., Witten, D., Hastie, T. & Tibshirani, R., 2013. "An introduction to statistical learning". New York: Springer.

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

Kassambara, A., 2018. "Machine learning essentials - practical guide in R". 1 ed. s.l.:s.n.

Koh, S., Durand, R. B., Dai, L., & Chang, M. (2015). Financial distress: Lifecycle and corporate restructuring. Journal of Corporate Finance, 33, 19–33.

Li, Y. and Wang, Y. (2018). Machine learning methods of bankruptcy prediction using accounting ratios. Open Journal of Business and Management, 06:1–20.

Liou, D. K., and M. Smith, (2007), Macroeconomic variables and financial distress, Journal of Accounting, Business and Management 14, 17–31.

Lombardo, G., Pellegrino, M., Adosoglou, G., Cagnoni, S., Pardalos, P. M., and Poggi, A. (2022). Machine learning for bankruptcy prediction in the american stock market: Dataset and benchmarks. Future Internet, 14(8).

Luengo, J., Fernandez, A., Garc ´ ´ıa, S., and Herrera, F. (2011). Addressing data complexity for imbalanced data sets: Analysis of smote-based oversampling and evolutionary undersampling. Soft Comput., 15:1909–1936.

Miller, D., & Friesen, P. H. (1984). A longitudinal study of the corporate life cycle. Management Science, 30(10), 1161–1183

Ohlson, J., 1980. Financial ratios and probabilistic prediction of bankruptcy. Journal of Accounting Research, 18(1), pp. 109-131.

Pastor, L., & Veronesi, P. (2003). Stock prices and IPO waves (No. w9858). National Bureau of Economic Research

Piesse J., Lee C.L, Kuo H., Lin L., (2006). Corporate Failure: definition, methods and failure prediction models. Encyclopedia of Finance, chap22, 477-488.

Robbins K., and Pearce J., (1993), Entrepreneurial retrenchment among small manufacturing companies, Journal of Business Venturing, Vol. 8, (4), 301-318

Shleifer, A, Vishny, R., (1992). Liquidation Values and Debt Capacity: A Market Equilibrium Approach. The Journal of Finance. Vol. 47, 1343-1366

Thornhill S., Amit R., (2003) Learning About Failure: Bankruptcy, Firm Age, and the Resource-Based View. Organization Science 14(5):497-509.

Whitaker, R. B., 1999. "The early stage of financial distress". Journal of Economics and Finance, Issue 23, pp. 123-133.

Sitography

Google Machine Learning Education, https://developers.google.com/ machine-learning/crash-course/classification/roc-and-auc

IBM, https://www.ibm.com/cloud/learn/bagging

IBM, https://www.ibm.com/cloud/learn/random-forest

IBM, https://www.ibm.com/cloud/learn/random-forest Imbalanced-learn, https://imbalanced-learn.org/stable/

Imbalanced-learn,https://imbalanced-learn.org/stable/references/generated/imblearn.oversampling.SMOTE.html

Intangible Capital, https://www.intangiblecapital.org/index.php/ic/article/view/1354/756 Investopedia, https://www.investopedia.com/terms/b/bankruptcy.asp.

Investopedia,https://www.investopedia.com/terms/d/debtrestructuring.asp

Investopedia,https://www.investopedia.com/articles/01/120501.asp#::text=Key%20Takeaways,operate%20under%20a%20reorganization%20plan.

Investopedia,https://www.investopedia.com/articles/active-trading/081315/financial-ratios-spot-companies-headed-bankruptcy.asp

Investopedia,https://www.investopedia.com/articles/financial-theory/10/spotting-companies-in-financial-distress.asp

Neptune, https://neptune.ai/blog/xgboost-vs-lightgbm

Neptune, https://neptune.ai/blog/when-to-choose-catboost-over-xgboost-or-lightgbm

Towards Data Science, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Towards Data Science, https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac

Wikipedia, https://en.wikipedia.org/wiki/Machine_learning

Wikipedia, https://en.wikipedia.org/wiki/Bankruptcy