University of Milano-Bicocca

Department of Economics, Management, and Statistics

DOCTOR OF PHILOSOPHY IN ECONOMICS AND STATISTICS

Black-box supervised learning and empirical assessment: new perspectives in credit risk modeling

Doctoral Dissertation of:
Marco Repetto

Advisor:
Prof. Caterina Liberati

Tutor:
Prof. Lisa Crosato

Supervisor of the Doctoral Program:
Prof. Matteo Manera

Preface

This dissertation details my four-year tenure as a Ph.D. candidate at the University of Milan-Bicocca and a Data Scientist at Siemens Italy. Prof. C. Liberati and Prof. L. Crosato have been supervising my research. During this period, I had the unique opportunity to pursue academic research, which fundamentally altered my outlook on life. I was involved in intriguing projects and experiences, the most significant being the year I spent at SKEMA Business School's Artificial Intelligence Institute.

My investigation and this piece of work have both benefited from the contributions of a vast number of individuals, to whom I am incredibly grateful. Moreover, since my first step into academia, I will be eternally grateful to Bruno, who, despite my doubts, persuaded me to apply for the doctoral program.

For trusting me from the first moment, I am thankful to my supervisors for their patience and guidance during this challenging period.

I am also profoundly grateful to the people of Siemens Italy, particularly Enrico Magrin, for his support and encouragement. Prof. D. La Torre, who hosted and supported my research abroad, deserves my gratitude. He was and still is one of the authors I enjoy reading.

In the section of the acknowledgments devoted to one's coworkers, I like to express my gratitude to all of the guys in U6 and Marco, with whom I endured the agony of paper writing together.

As well as my family, who have always supported and encouraged me, I want to thank my girlfriend, Vittoria. She has always been by my side, sharing my research journey's ups and downs. You are the greatest treasure in my life.

Marco Repetto
Milano
January 2023

Abstract

Recent highly performant Machine Learning algorithms are compelling but opaque, so it is often hard to understand how they arrive at their predictions giving rise to interpretability issues. Such issues are particularly relevant in supervised learning, where such black-box models are not easily understandable by the stakeholders involved. A growing body of work focuses on making Machine Learning, particularly Deep Learning models, more interpretable. The currently proposed approaches rely on post-hoc interpretation. Despite these advances, interpretability is still an active area of research, and there is no silver bullet solution. Moreover, in high-stakes decision-making, post-hoc interpretability may be sub-optimal. An example is the field of enterprise credit risk modeling. In such fields, classification models discriminate between good and bad borrowers. As a result, lenders can use these models to deny loan requests. Loan denial can be especially harmful when the borrower cannot appeal or have the decision explained and grounded by fundamentals. Therefore in such cases, it is crucial to understand why these models produce a given output and steer the learning process toward predictions based on fundamentals. This dissertation focuses on the concept of Interpretable Machine Learning, with particular attention to the context of credit risk modeling. In particular, the dissertation revolves around three topics: model agnostic interpretability, post-hoc interpretation in credit risk, and interpretability-driven learning. More specifically, the first chapter is a guided introduction to the model-agnostic techniques shaping today's landscape of Machine Learning and their implementations. The second chapter focuses on an empirical analysis of the credit risk of Italian Small and Medium Enterprises. It proposes an analytical pipeline in which post-hoc interpretability plays a crucial role in finding the relevant underpinnings that drive a firm into bankruptcy. The third and last paper proposes a novel multicriteria knowledge injection methodology. The methodology is based on double backpropagation and can improve model performance, especially in the case of scarce data. The essential advantage of such methodology is that it allows the decision maker to impose his previous knowledge at the beginning of the learning process, making predictions that align with the fundamentals.

Contents

List of Figures

List of Tables

List of Acronyms

ML Machine Learning

DL Deep Learning

AI Artificial Intelligence

XAI eXplainable Artificial Intelligence

ICE Individual Conditional Expectation

SHAP Shapley Additive Explanations

LIME Local Interpretable Model-agnostic Explanations

PD Partial Depencence

ALE Accumulated Local Effects

EU European Union

SME Small and Medium Enterprise

NFBS non-financial business sector

XGBoost eXtreme Gradient Boosting

FANN Feedforward Artificial Neural Network

BGEVA Binary Generalized Extreme Value Additive

LR Logistic Regression

AME Average Marginal Effect

AUC Area Under the Receiver Operating Curve

Introduction 1

ML techniques impact our daily lives, revolutionizing and innovating many businesses. From recommender systems capable of suggesting our next song or movie to scoring models performing loan approval in the blink of an eye. ML systems shift the paradigm of previous AI techniques, allowing knowledge extraction directly from the data instead of having rules explicitly programmed by software engineers. This paradigm shift has become more evident with the advent of DL (LeCun et al., 2015), which allowed for the solution of complex problems like protein folding Wei, 2019. Because of this and several other reasons related to the ease of training and implementing such solutions, ML systems are subject to brand-new interest by companies. According to Wellers et al., 2014, the digital transformation affecting leading organizations is the reason behind such massive adoption.

One way to measure the level of investment in these technologies is to look at the amount of venture capital funding that goes into AI startups. This metric has increased steadily over the past few years, from $2.4 billion in 2016 to $9.3 billion in 2019 (Insights, 2022). The same is happening in academia, where ML papers have been increasing steadily. The combined effect of increased investments and academic activities in ML creates higher expectations about new possible applications and some skepticism (Mitchell, 2021). Nonetheless, in a survey of AI experts, 62% said they believe ML will ”substantially transform” society by 2030, and 18% said ML would ”transform” or ”revolutionize” society (Anderson & Rainie, 2018).

Out of this new AI summer, we can see the manufacturing sector’s role in steering innovation. Companies like Siemens, ABB, and General Electric invests significantly in AI, creating new technologies that can help factories increase their efficiency and productivity while cutting down on costs (Arora et al., 2022; Petri et al., 2021; Zhong et al., 2021). An example is the Siemens Mindsphere, a cloud-based data platform that uses data collected from various sources to help improve the performance of industrial equipment (Annanth et al., 2021). However, this is just one example of how the manufacturing sector uses ML to enhance its operations. ML is also dramatically impacting the financial field (Donepudi, 2017). Banks have been using ML for some time to automate customer support, fraud prevention, and loan approvals (Leo et al., 2019). The pervasiveness of ML in the financial sector will likely increase. There is credit risk modeling among the areas related to the financial sector that saw a surge in interest. Credit risk modeling is the process of assessing the risk of default by a borrower. Such an assessment includes estimating the probability of default and the loss linked to the default event. Lenders can use credit risk models to help make decisions about granting credit, setting credit limits, and pricing loans. In recent years, there has been a surge in the use of ML methods to assess credit risk (Ciampi et al., 2021). A few key factors can attribute the increased use of ML methods in credit risk. First, ML methods can learn from data and make sound predictions. Additionally, ML methods can identify complex patterns in data that may be difficult or impossible to uncover using traditional methods. Finally, ML methods can provide accurate predictions with relatively little data, which is crucial in the world of credit risk. With ML methods, credit risk professionals can make better decisions quickly and accurately.

The focus of this dissertation is specifically on credit risk modeling and, in particular, bankruptcy prediction. Bankruptcy prediction models are mathematical models used to estimate a firm's bankruptcy probability. To model bankruptcy, one must first understand the probability of default. The probability of default is the likelihood that a firm will not be able to meet its financial obligations promptly. It is important to note that the probability of default differs from the probability of loss. The latter is the likelihood that a borrower will default on a loan and the lender will lose money. As set forward by Yu et al., 2014, these types of problems are classification tasks in which the target variable is a binary one, and the features derive from the company's financial statement.

In this work, we also aim to produce interpretable models, that is, models that can explain to humans why they have made specific predictions. Interpretability is a crucial issue in ML. Researchers have proposed many ML models in the past, and these models have demonstrated remarkable performance in numerous tasks. However, most proposed models are not interpretable because their inner workings need to be better understood.

The point of interpretability is essential for many reasons. First, we need to understand why a model produces a given prediction to trust the model. Second, if a model is not interpretable, it is hard to use in decision-making. Third, if a model is not interpretable, it isn't easy to improve it. For example, consider again a bankruptcy model that could be more accurate. If the model is interpretable, it is easier to understand why it is not precise, and it is easier to improve such a model.

In addition to interpretability, another critical issue in ML is the use of expert knowledge. Expert knowledge is a type of knowledge that is not directly available in the data but accepted by experts in the field. Such knowledge can improve the performance of ML models. For example, consider a medical diagnosis task to diagnose a disease from a set of symptoms. In this task, the data may need more information to detect the illness accurately. However, if we have expert knowledge about the disease, then we can use this knowledge to improve the accuracy of the diagnosis.

It is often challenging to use this type of knowledge in ML, as it is hard to encode it in a form that ML models can use. A strategy can be to select a subset of features in advance from financial expertise Yu et al., 2014. Although this strategy seems compelling, feature selection may be suboptimal, especially for complex models.

In the case of credit risk models, knowledge injection is also crucial to avoid possible oddities. Even though the field is still primarily empirical, as pointed out by du Jardin and Séverin, 2011, we can distill some knowledge applicable to some other complex cases. An example is in (Ahelegbey et al., 2019) in which the authors find that a liquidity indicator such as the quick ratio positively impacts the probability of default. This behavior is at odds with all the literature, and the advantage of injecting expert knowledge into model estimation is an attempt to solve this problem. Another worth noting example is in Andreeva et al., 2016. In the study, higher leverage leads to a decrease in the probability of default. As pointed out by the authors, such a fact is at odds with the relevant literature on the subject.

To summarise the previous considerations, this dissertation is about ML, interpretability, and knowledge injection in credit risk modeling. Three chapters constitute this dissertation. The first chapter guides the reader toward model-agnostic approaches and implementations influencing today's ML landscape. Such a chapter has been accepted for publication in World Scientific Publishing's book ”AI and Beyond for Finance” as part of the series Transformations in Banking, Finance, and Regulation. The second chapter focuses on an empirical examination of the credit risk of Italian SMEs. This chapter submitted to ”Applied Stochastic Models in Business and Industry” received the first round of reviews. The final and third chapter introduces a novel multicriteria knowledge injection mechanism. This chapter, submitted to ”Annals of Operations Research,” is available from April 2022.

Interpretability in Machine Learning 2

Marco Repetto

AI and Beyond for Finance. Series of Transformations in Banking, Finance, and Regulation. World Scientific Publishing. *Accepted.*

Given the pervasiveness of complex ML models, it is worth questioning their transparency and interpretability. In essence, interpretability is the ability to understand the rationale behind the predictions of a model. It is a process of making the working of a model understandable to humans. The interpretability of a model is essential because it helps us understand how the model works and why it makes the predictions it does. There are two main types of interpretation methods: model-based and model-agnostic. Model-based interpretability methods try to explain the behavior of a machine learning model by analyzing its structure and parameters. On the other hand, model-agnostic interpretability methods try to explain the behavior of a model without making any assumptions about its internals. This chapter provides an overview of the many different model-agnostic techniques for interpretability. In particular, the focus will be on local and global interpretability methodologies.

2.1 Introduction

From self-driving cars to score credit ratings, ML is already starting to shape our world (Lessmann et al., 2015; Rao & Frtunikj, 2018). Nevertheless, what is ML and how can we explain some models' predictions still sparks much debate in academia. In its simplest form, ML is a way of teaching computers to learn from data without being explicitly programmed by building algorithms that can automatically improve given more data (Jordan & Mitchell, 2015). With good reason, ML is a hot topic in computer science right now. Capable of amazing things, like teaching computers to recognize objects in images or understand spoken language. However, there is a dark side to ML, too. Some ML models can generate fake figures, also called deep fakes, used for all sorts of nefarious purposes, like creating fake news or spreading disinformation (Westerlund, 2019). Another dark side of such methods is their inherent interpretability. In fact, with the rise of ML, a new question has emerged regarding the trustworthiness of such algorithms. The answer, it turns out, is not so simple. A growing body of work shows just how easy it is to fool these models (Szegedy et al., 2013). Moreover, there is evidence that these models can be biased against certain groups of people (Rudin, 2019). So the question: "how can we be sure that the ML models we use are fair and trustworthy?" is crucial, especially in high-stakes decisions. One way to achieve fairness is to ensure that the data we use to train these models is diverse and representative of the population. It is essential for sensitive applications, like healthcare or law enforcement. Another way to ensure fairness is to use interpretable ML models. These models can be explained to humans and are less likely to contain biases. There are many different types of interpretable ML models. One popular type is a decision tree. However, not all ML models are inherently interpretable. Especially in current ML methods, the issue of interpretability is crucial because it can be challenging to understand how these complex algorithms make predictions. In principle, we can define interpretability as the ability to understand the rationale behind the predictions of a model. It is a process of making the working of a model understandable to humans. The interpretability of a model is essential because it helps us understand how the model works and why it makes the predictions it does. There are many ways to make ML algorithms more interpretable, such as using simpler models (i.e., decision trees) or providing explanations of the predictions. However, trade-offs are often necessary, such as sacrificing accuracy for interpretability. One can define interpretability in ML as the ability to explain the behavior of an ML system. It is a relatively new field, with active research beginning in the late 2010s whose goal is to make ML more transparent and accountable. Essentially there are two main approaches to interpretability: model-based and model-agnostic. Model-based interpretability methods try to explain the behavior of an ML model by analyzing its structure and parameters. On the other hand, model-agnostic interpretability methods try to explain the behavior of an ML model without making any assumptions about its internals. Interpretability in complex ML models is vital for many reasons beyond fairness. It can help us understand how ML models work and why they make their decisions. Such knowledge can improve the models and helps build new models that are more interpretable. It is worth noting how interpretable ML differs from XAI. Interpretable ML is a branch of ML that deals with interpreting and explaining the models produced by ML algorithms. Instead, XAI is a subfield of AI that deals with developing methods and techniques to make ML models more interpretable and explicable. Although there is much overlap between the two fields, many approaches designed in one area apply to others. However, there are some crucial differences between the two fields. Interpretable ML focuses on the interpretation of ML models, while XAI focuses on explaining the decisions made by ML models. Interpretable ML algorithms target explanations understandable by humans. XAI solutions instead target machines. Last, interpretable ML methods can help improve the performance of ML models. In contrast, XAI methods aim at fostering the interpretability of ML models.

This chapter aims to provide a broad overview of interpretable ML. It covers why interpretability is essential, the different approaches to interpreting ML models and the challenges involved in making ML models interpretable. It also provides the reader with some knowledge of the recent implementations of such techniques, either in Python (Python Core Team, 2019) or R (R Core Team, 2021). Moreover, using publicly available datasets such as Boston housing or Titanic allow the reader the opportunity to experiment with interpretable ML techniques and gain an understanding of the complexity of the problem.

More specifically, the chapter discusses the following topics:

- Section 2.2: presents the historical roots of interpretability, which dates back to cybernetics;
- Section 2.3: explains the importance of interpretability under different perspectives;
- Section 2.4: gives an overview of the most popular interpretability methods currently used;
- Section 2.5: conveys what are the relevant challenges in ML interpretability.

The remainder of the chapter is a discussion of the field and concludes.

2.2 The historical roots of interpretability

The history of interpretability in ML goes back to work in the early days of AI and cybernetics. In the early days, the field was primarily concerned with methods for analyzing and understanding the behavior of linear models.

In the 1950s, cybernetician Ross Ashby postulated that any system (including an ML system) could be made understandable by reducing its complexity (Ashby, 1956). This principle, known as Ashby's Law of Requisite Variety, suggests that the level of understanding of a system must match the system's complexity to be effective. In practical terms, however, the starting point in the history of interpretable ML can be traced back to the early days of artificial neural networks. One of the earliest examples of interpretation in the field was the work of Marvin Minsky and Seymour Papert on the explanation of the behavior of such structures. In their 1969 book, Perceptrons, Minsky and Papert, showed how the behavior of these networks could be explained by analyzing the connection weights between the neurons (Minsky & Papert, 2017). Other early examples of interpretable ML include the work of D.E. Knuth on the explanation of the behavior of heuristic search algorithms (Knuth & Moore, 1975) and the work of E.H. Shortliffe on the explanation of the behavior of expert systems (Shortliffe et al., 1975). In the 1980s, work in the field of neural networks showed that it is possible to create models that are both accurate and interpretable (Saito & Nakano, 1988). This work demonstrated that neural networks could learn to approximate any function, regardless of its complexity. Furthermore, the structure of a neural network can be interpreted as a set of rules that can be used to make predictions. However, the field began to take off in the 1990s with the development of new techniques for explaining the behavior of AI systems (Nauck & Kruse, 1999; Setiono, 1996; Setiono & Liu, 1995). Since then, there has been a growing body of work, with new techniques being developed and applied to various ML systems. In particular, during this decade, several methods were developed for making decision trees more interpretable (Bredensteiner & Bennett, 1996). Decision trees are a type of ML model which is easy to understand and can be used to make predictions. However, decision trees can be very complex, and it can be difficult to understand why the model made a particular prediction. Nevertheless, in the 2000s and, subsequently, 2010s, the field became particularly renown. At that time, several methods were developed to interpret ML models' predictions (Friedman, 2001; Friedman & Meulman, 2003; Goldstein et al., 2015; Štrumbelj & Kononenko, 2010). One of the most influential works in this area was the paper "Why Should I Trust You?": Explaining the Predictions of Any Classifier by Local Interpretable Model-agnostic Explanations by Ribeiro et al., 2016a, which was published in 2016.

To summarize, interpretable ML has a long history, dating back to the 1950s. This work continued in the 1970s and 1980s, focusing on developing more sophisticated methods for analyzing non-linear models. The field began to gain more mainstream attention in the 1990s as the ML community began to realize the importance of understanding the behavior of complex models. In the 2000s, many researchers focused on developing methods for interpreting black-box models. This work has continued in the 2010s, with a growing focus on developing new methods for understanding the behavior of deep neural networks. It is worth noting that interpretability in ML is still an active area of research. No one approach is universally accepted as the best way to achieve it. However, the methods that have been proposed so far provide a promising start.

2.3 The importance of interpretability

Interpretability is essential in ML for several reasons and can benefit many different stakeholders.

The first stakeholder is the modeler. Having an interpretable model can help the modeler to understand how the model is making predictions. Furthermore, this is clearly among the best practices applied in MLOps and ModelOps frameworks (Tamburri, 2020). Another benefit of making ML models interpretable to the modeler regards the models' overall performance. In accordance with Molnar et al., 2020, an interpretable model is more likely to perform better than a more complex model. In fact, in their work, they found that when a model is not interpretable, it is more difficult to understand why it is not working as expected, making it more challenging to improve. The second crucial stakeholder is the decision-maker. Making crucial decisions trusting an inscrutable model poses serious threats and risks. An interpretability layer may help the decision-maker understand why the model is making specific predictions. Moreover, interpretability can help improve the transparency of the models since it can help to explain how the models work to people who are not experts in ML. Last but not least, the vital stakeholder of any ML model is the end-user. The final user is the one whom the model's predictions will impact, and they must understand how the model is making those predictions. If the model is not interpretable, the user may not trust the predictions and may not use the model. Another benefit to the end used regards the capability of helping to improve the fairness of the models since it can help identify biases. In this sense, interpretable ML is essential for the recent regulations. In the United States, there have been two significant laws passed in the last few years that have increased the importance of making ML models interpretable. The first is the Dodd-Frank Wall Street Reform and Consumer Protection Act. This act requires financial institutions to disclose the rationale behind their automated decision-making. The second is the European Union's General Data Protection Regulation. This regulation gives individuals the right to know why an automated decision was made about them. These regulations have put pressure on organizations to make their ML models interpretable. If an organization cannot explain why a decision was made, it may be subject to fines or other penalties.

2.4 Interpretability methods

This section provides an overview of some of the most common interpretability methods. However, before presenting these methods is worth defining what is intended for an ML model to be interpretable.

Interpretability is the process of understanding the meaning behind the output of an ML model. The goal of interpretability is to provide a model that can be given to a decision-maker to understand how it is making its predictions. Essentially, interpretability provides a human-friendly description of how the model makes its decisions.

A model that is easy for a human to explain is more likely to be used than a model that is difficult to understand. We saw in Section 2.3 how interpretability is essential in decision-making and from a regulatory standpoint. However, nothing was said about how interpretability is measured. There is no one size fits all answer to this question. The most important thing is to make sure that the interpretation is meaningful to the people using the model. Nevertheless, the best interpretability method to use depends on the specific ML model and the specific question that the stakeholder wants to answer. In general, interpretability methods can be used to understand individual predictions, understand the overall behavior of a model, or help design new, more interpretable models. Furthermore, interpretability methods should satisfy some of the properties that make an explanation good. These properties are also known as desiderata in Wickham et al., 2019 and are:

- Causality: an explanation should be able to explain the reason why a model predicts a certain output;
- Contrastive: an explanation should be able to explain the reason why a model predicts a certain output as opposed to a different one;
- Consistency: the explanation should be consistent with the model;
- Faithfulness: the explanation should be faithful to the model;
- Globalness: the explanation should be global in the sense that it should be able to explain the model as a whole;
- Localness: the explanation should be local in the sense that it should be able to explain the model for a single example;
- Illustrativeness: the explanation should explain the model with an example;
- Simplicity: the explanation should be simple to understand;

- Naturalness: the explanation should be natural to understand;
- Generality: the explanation should be generalizable to other examples.

There are many different interpretability methods, each with its strengths and weaknesses. Some interpretability methods are more applicable to certain types of models than others. In general, interpretability methods can be divided into two broad categories:

- Model-based methods; and
- Model-agnostic methods.

Model-based methods are specific to a particular type of ML model. They exploit the structure of the model to provide insights into how the model works. Model-agnostic methods are not specific to any particular type of ML model and will be the ones that will be discussed in the next subsections.

This section will cover model-agnostic methods based on their explanation, which can be divided into two main categories: global and local. Global model interpretation methods are used to understand how the model works. We want to understand how the model works for all data points, not just a single data point. Local model interpretation methods are used to understand how the model works for a specific data point. This means that we want to understand how the model works for a single data point, not all data points. Global model interpretation methods are typically more expensive because they require us to compute the model output for all data points. On the contrary, local model interpretation methods are typically less expensive because they only require us to compute the model output for a single data point.

2.4.1 Local methods

Local explanations expose the reasons why a model predicts a certain output for a given input. Also called instance-level methods, they help to understand how a model yield a prediction for a single observation (Biecek & Burzykowski, 2021). These types of explanations are usually provided to the user in the form of human-readable text or a graphical interface. These methods are of incommensurable importance in high-stakes decisions as for the case of credit scoring (Bücker et al., 2021a).

In this subsection, we will discuss the following local methods:

- Individual Conditional Expectation;
- Shapley Additive Explanations;

- Local Surrogates.

2.4.1.1 Individual Conditional Expectation

The Individual Conditional Expectation is a method that allows for visualizing the effect of a feature change on an instance basis.

Proposed initially by Goldstein et al., 2015, ICE can be seen as a decomposition of Partial Dependence, a method discussed in Section 2.4.2.1. Also called Ceteris Paribus Profiles by Biecek and Burzykowski, 2021, the individual conditional expectations let the modeler and user understand how the forecast would vary if the values of the variables in the model varied. The intuition behind ICE is that the effect of a feature change is the difference between the prediction with the feature change and the prediction without the feature change. In mathematical terms, we observe N data points. For each data point, we keep constant some of the features; we call them $\boldsymbol{x}_C$ and let one feature vary, that is, x_s. The results are $i = 1 \ldots N$ curves, $f_s^{(i)}$. Figure 2.1 shows what an ICE plot looks like. In particular, the plot was obtained using the iml package (Molnar et al., 2018) in R and portrayed the individual conditional expectation of a random forest model for the feature crime in the Boston dataset (Harrison & Rubinfeld, 1978).

2.4.1.2 Shapley Additive Explanations

Shapley Additive Explanations is a local interpretability method that provides information about features' contributions to the outcome.

It is based on game theory, specifically, the Shapley value from cooperative game theory (Shapley, 1953). The Shapley value was developed initially to distribute the payouts for a cooperative game among the game's players. In game theory, a cooperative game is a game where players can form coalitions and work together to achieve a common goal. In mathematical terms, we can define these contributions as:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S)) \tag{2.1}$$

where N is the set of features, v is a function giving the value for any subset of those features, and S is a coalition of features which are a subset of N. Evaluating any coalition is intractable, therefore Štrumbelj and Kononenko, 2013 proposed the following approximation:

$$\hat{\phi}_i = \frac{1}{m} \sum_{j=1}^{m} V_j \tag{2.2}$$

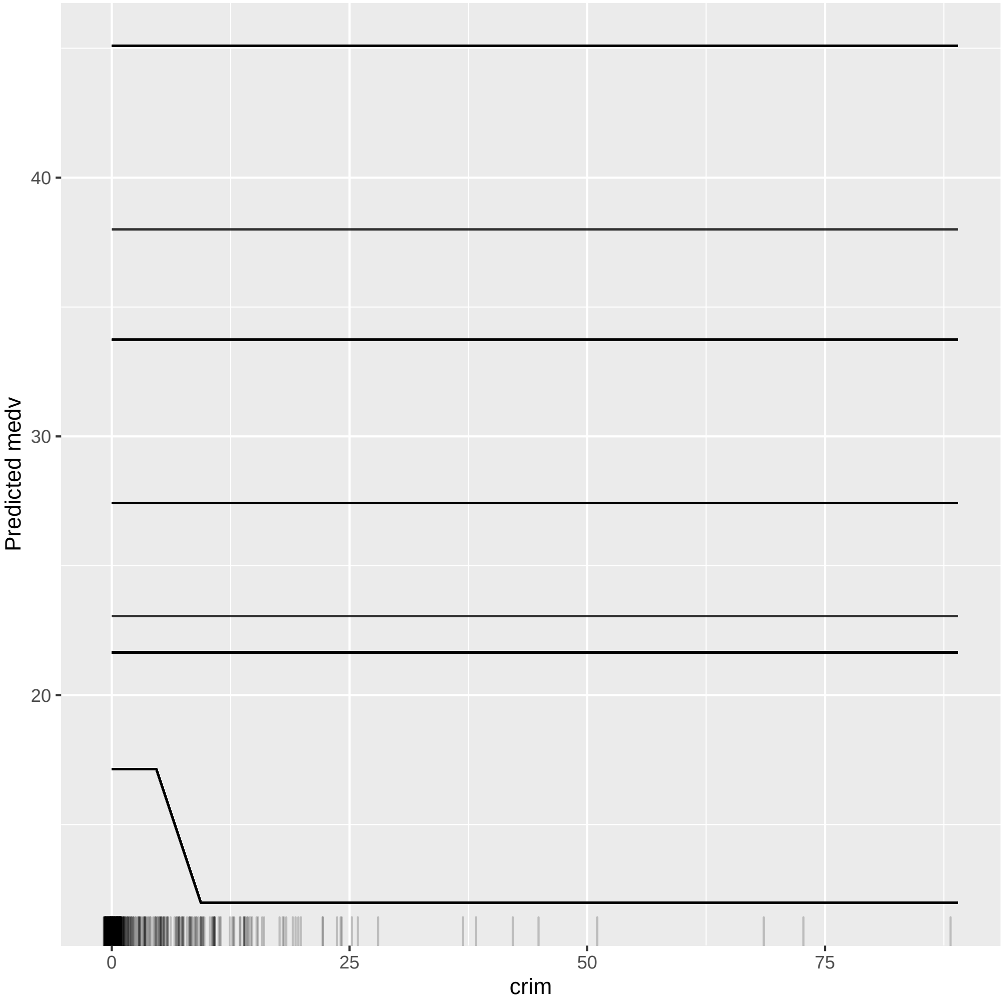


Figure 2.1: Individual Conditional Expectation curves for a Random Forest model trained using the Boston dataset. The flat lines pertain to observations for which the model predicts constant average effects on the ”medv” outcome. The ribbon at the bottom of the plot shows the distribution of the crim feature.

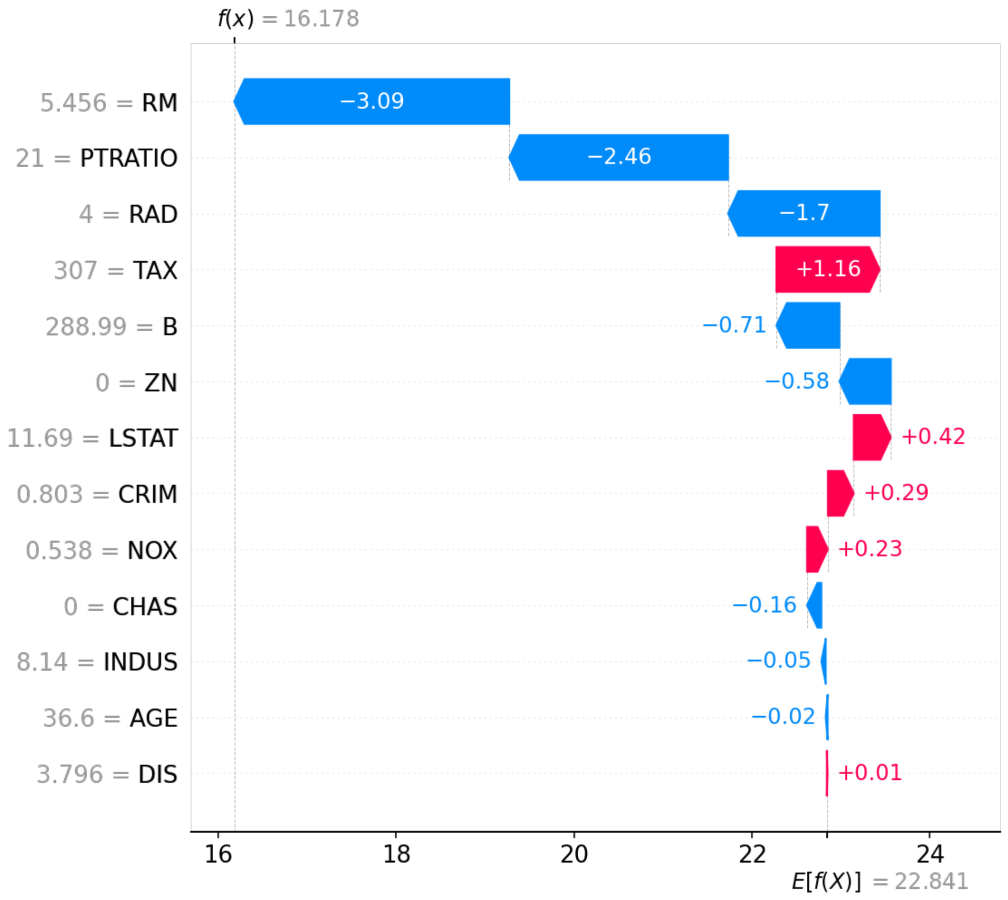


Figure 2.2: Shapley Additive Explanations for a Boosted Trees model of the Boston Dataset. The $f(x)$ tells the stakeholder about the final model outcome, whereas $E[f(x)]$ is the average model response. At the y-axes are reported the values of the observation for which the stakeholder is seeking the explanation. Each SHAP value will add to the outcome. Negative SHAP values are in blue, whereas positive values are in red.

where V_j is a random sample measuring the difference in the contribution by having a specific S coalition in place.

In ML, the Shapley value can be used to determine how much each feature contributes to the model's output. SHAP, as proposed by Lundberg and Lee, 2017, can be seen as further refined this estimator. In the assumption of feature independence, SHAP values can be estimated directly using the formula of Štrumbelj and Kononenko, 2013.

Figure 2.2 shows SHAP explanations of an ML model. In this case, the plot was obtained using the shap package (Lundberg & Lee, 2017) in Python.

2.4.1.3 Local surrogates

Local surrogate interpretability methods are essentially simulation-models trained to approximate the output of a complex ML model.

The idea is that, instead of interpreting a complex model, which can be impossible, a surrogate model can be used to generate results that are close to the complex model's output, with a much lower interpretability burden. Surrogate models are often used in optimization to have a less computationally expensive optimization routine (Kochenderfer & Wheeler, 2019). In this case, the surrogate model is used to evaluate different input values quickly. According to the surrogate model, the inputs that lead to the best output are then used as inputs to the complex model to get the final result. In the case of ML interpretability, the aim is to approximate the complex model locally and then study the behavior of the surrogate. There are many different surrogate models, including regression models and decision trees. The choice of surrogate model depends highly on the type of data and the structure of the model to interpret. The two most known surrogate models are the Local Interpretable Model-agnostic Explanations proposed by Ribeiro et al., 2016b and Anchors, also proposed by Ribeiro et al., 2018. In LIME, a model is explained by learning a locally accurate, interpretable model around the instance being explained. In other words, we say that a LIME explanation $\xi(x)$ should satisfy the following:

$$\xi(x) = \underset{g \in G}{\text{argmax}} \quad \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.3}$$

where f is the model for which we need an explanation, g is a simpler model such as a linear regression π_x is a proximity measure to the observation we want to explain and the second term of the objective function is a regularization measure.

Figure 2.3 shows a LIME model evaluated using the iml package in R.

An Anchor, in contrast to LIME, is a rule that holds the prediction locally, meaning that changes to the rest of the instance's feature values have no effect. Anchors have the advantage of being easy to comprehend because of their specificity and also intuitiveness.

Figure 2.4 taken from the paper of Ribeiro et al., 2018 shows such a difference.

2.4.2 Global methods

We detailed the local methods for the models' interpretation. These methodologies allow the stakeholders to probe a model at the instance level. However, most of the time, the aim is to understand the model behavior on the entire dataset. This holistic view can be used to spot possible biases affecting multiple obser-

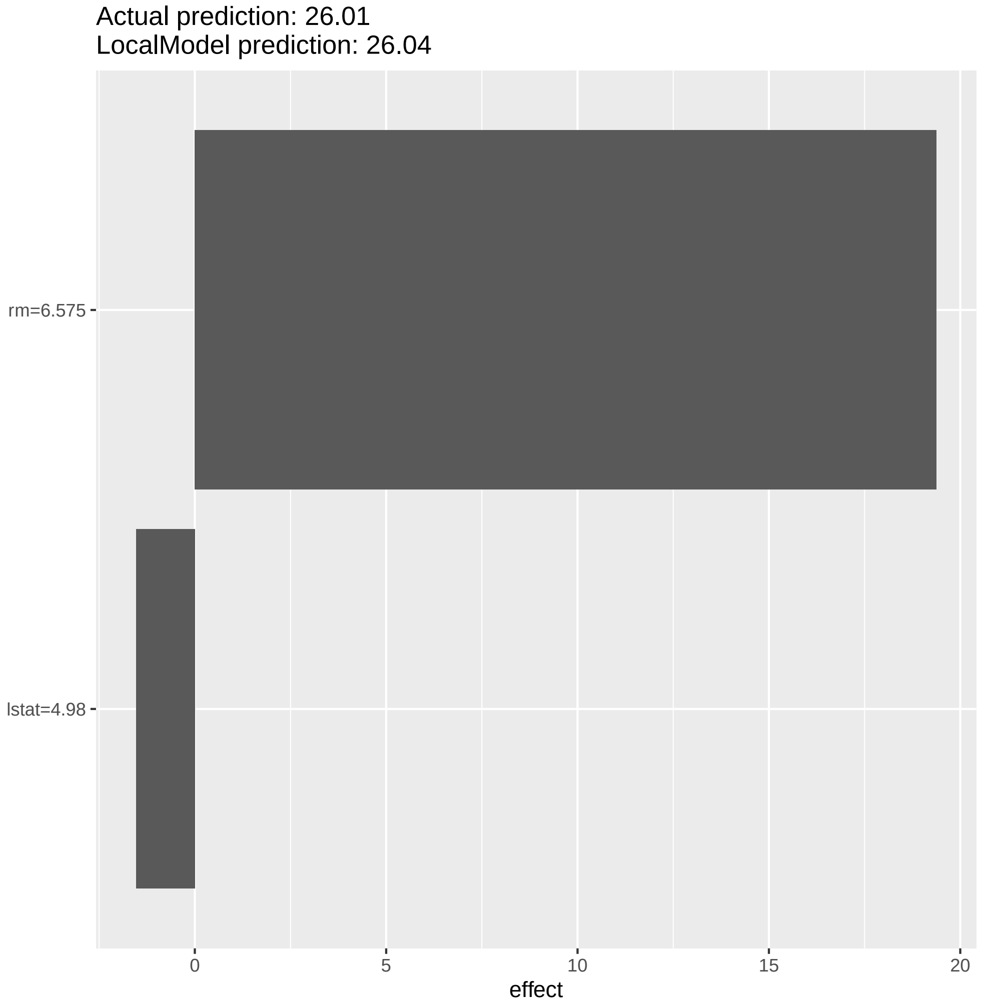


Figure 2.3: Local Interpretable Model-agnostic Explanations for a Random Forest model trained on the Boston dataset. The plot provides the stakeholder with information about the goodness of approximation of the local surrogate, namely actual prediction and LocalModel prediction. Then the plot shows a bar chart of the most relevant effect driving the local surrogate model outcome.

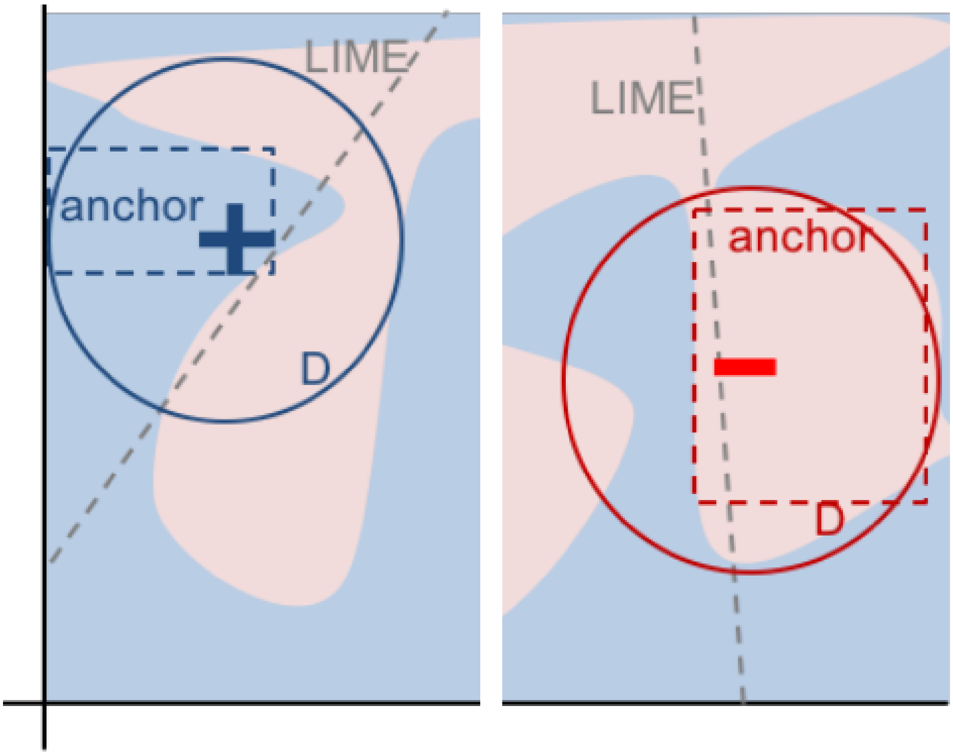


Figure 2.4: Comparison between Anchors and Local Interpretable Model-agnostic Explanations. The plot depicts classification with non-linear decision boundaries and the different behavior of the two local model surrogates methodologies.

vations. Global explanations methods go in this sense as they summarize the model as a whole. Furthermore, global explanations provide additional insights in comparison to local explanations. As mentioned previously, perhaps we want to understand how a specific feature influence the final predictions. As pointed out by Repetto, 2022, many bankruptcy prediction methods may perform well by leveraging dataset biases or spurious correlations. Therefore, a global explanation layer is required to provide robust models, especially in production. Another advantage of global explanations is that they allow measuring the model's feature importance. This is a crucial aspect that allows for parsimonious modeling, especially in high data dimensionality. Last but not least, we may decide to focus on a subset of the dataset and apply these techniques. A clear example is in the case of bankruptcy prediction. We may wonder why certain healthy firms are misclassified or vice versa. By using global explanations, we can uncover the odd model behaviors and provide a solution through feature engineering or modifying the model training process.

The global methods treated in this subsection are:

- Partial Depencence;
- Accumulated Local Effects;
- Feature importance;
- Global surrogates.

2.4.2.1 Partial Dependence

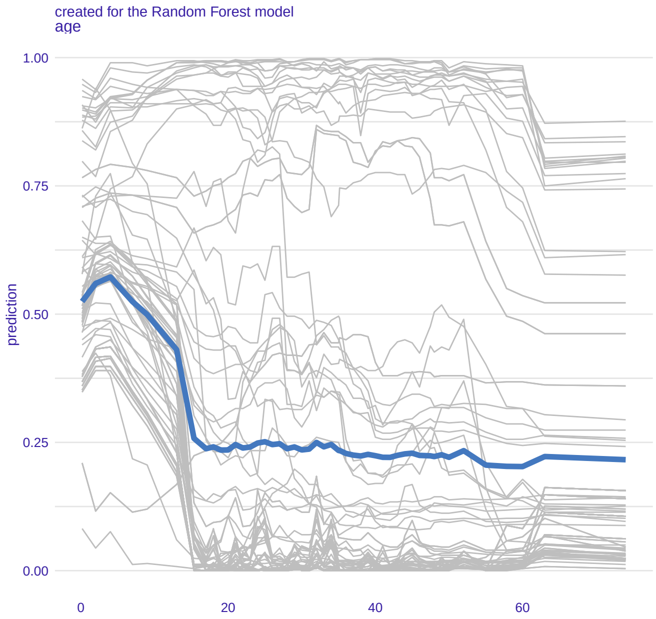
Partial Dependence is a method to understand how the model behaves with respect to a feature change. The idea is to plot the dependence of the model output on the feature while fixing all other features to some baseline values. It can be done one feature at a time or by picking a pair of features. The PD plot for a single feature shows the marginal effect of the feature on the model output. In the case of pairs of features, the resulting plot will uncover possible features interaction driving the model outcome. PD plots are a valuable tool for understanding how an ML model works. They can help us to understand which features are most important to the model and how the model depends on those features. PD plots are model agnostic, meaning that they can be computed for any ML model. Mathematically Partial Dependence can be evaluated with the following equation:

$$\frac{\partial f}{\partial x_i} = \frac{1}{N} \sum_{n=1}^{N} \frac{f(x_i, \hat{x}_{-i}^{(n)}) - f(\hat{x}_i^{(n)}, \hat{x}_{-i}^{(n)})}{x_i^{(n)} - \hat{x}_i^{(n)}} \tag{2.4}$$

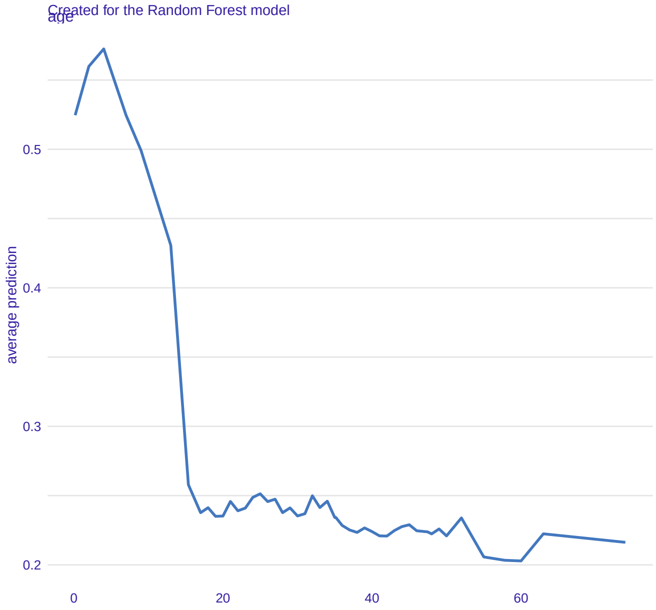
where f is the model prediction, $\hat{x}_i^{(n)}$ and $\hat{x}_{-i}^{(n)}$ are the values of the feature x_i and all the other features, respectively, for the nth observation, and $x_i^{(n)}$ is the original value of feature x_i for the nth observation. The PD concept is related to ICE as the former is essentially the average of all the ICE curves computed for a specific feature. The previous statement is evident by looking at Figure 2.5a obtained using the DALEX (Biecek & Burzykowski, 2021) package in R. In grey are depicted all the ICE curves about each observation. Whereas in blue, it is portrayed the PD of the age feature. Furthermore, the sole PD plot can be obtained with the same package as shown in Figure 2.5b.

2.4.2.2 Accumulated Local Effects

Accumulated Local Effects can be seen as a further refinement of the PD. The idea behind ALE is to compute the effect of a given feature at roughly every value



(a)



(b)

Figure 2.5: Individual Conditional Expectation curves and Partial Dependence. Figure (a) shows the how Partial Depencences (blue line) is the average of Individual Conditional Expectation curves (grey lines). Figure (b) depicts the same Partial Depencence of Figure (a) obtained from a Random Forest using the Titanic dataset.

of the predictor while holding all other predictors at their mean value. First, the feature space is binned to compute the feature's effect at its ith value. Then the effects at each bin border are evaluated by permuting the other features. Last, the effects at each bin's border are subtracted to avoid other features' spurious effects. This results in a value where we can observe the feature effect but without the other features having a relationship with the response. Finally, the ALE response values are plotted against the original values of the features. In this framework, ALEs constitute a further refinement of PD. They avoid the PD plots-drawback of assessing variables' effects outside the data envelope (Apley & Zhu, 2020). Mathematically speaking, computing the ALE implies the evaluation of the following type of function:

$$ALE_{\hat{f},S}(x_S) = \sum_{k=1}^{k_S(x)} \frac{1}{n_S(k)} \sum_{i:x_S^{(i)} \in N_S(k)} \Delta_{z_{k,j},z_{k-1,j},x_{\setminus S}^{(i)}} - C \tag{2.5}$$

where $\hat{f}$ is the ML model itself, S constitutes the subset of variables' index, X is the matrix containing all the features, and$z$ identifies the boundaries of the K partitions such that $z_{0,S} = \min(x_S)$.

The C constant term in the equation is essentially the model average, in other words:

$$C = \frac{1}{n} \sum_{i=1}^{n} ALE_{\hat{f},S}(x_S^{(i)}) \tag{2.6}$$

The only advantage of C is that it centers the plot. Figure 2.6 shows the ALE plot for the RM feature with quantile binning. The plot was obtained using the Alibi package (Klaise et al., 2021) in Python.

2.4.2.3 Feature importance

Feature importance is an immense field of global explanations. In principle, feature importance measures can also be obtained using the two previously seen global methods, PD and ALE. In the case of PD, the intuition is to rank the features in terms of their PD variability (Greenwell et al., 2018). In other words, the authors define the feature importance, say $i(x)$ as:

$$i(x_i) = F\left(\frac{\partial f}{\partial x_i}\right) \tag{2.7}$$

where $F(\cdot)$ is the sample standard deviation for the case of a continuous variable or the range divided by four in the case of categorical variables. The division by four provides an estimate of the standard deviation for a small to moderate sample

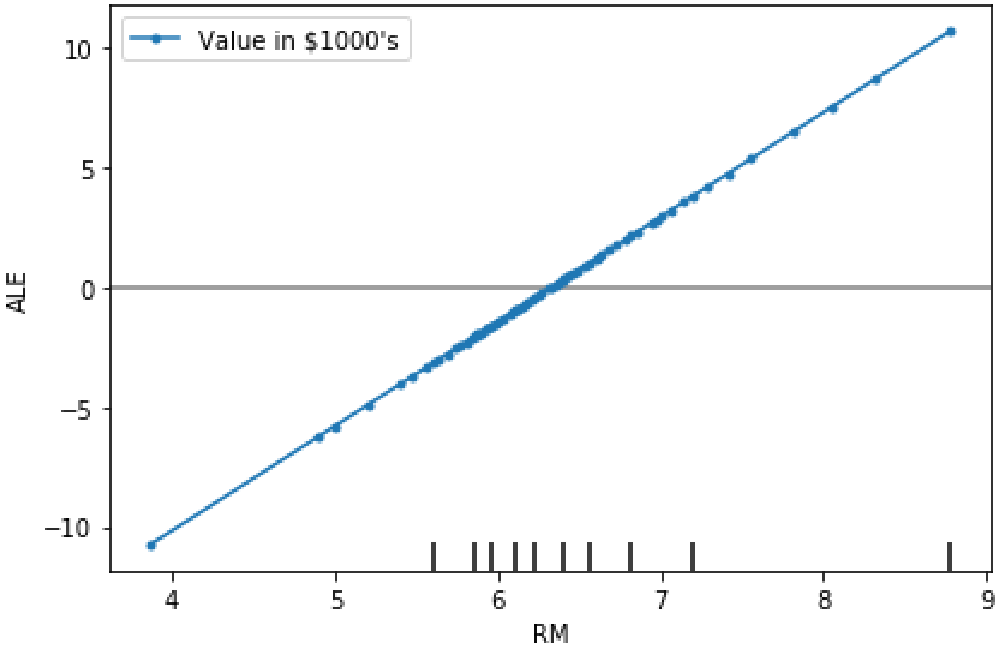


FIGURE 2.6: Accumulated Local Effect plot for a Random Forest model trained on the Boston dataset. The increasing line shows a positive effect on the model's outcome. The ribbon at the bottom of the plot shows the distribution of the RM feature.

size. The same goes for the ALE, as posed by Apley and Zhu, 2020 in which they define ALE range as a measure of feature importance for continuous variables. More commonly, what is intended as feature importance is permutation feature importance. The permutation feature importance is defined as the decrease in the model score when a single feature value is randomly shuffled. A feature is considered "important" if shuffling its values increases the model error. This is calculated for each feature of the data and then normalized before being ranked. More precisely, The permutation feature importance is calculated for each feature in the following steps. First, the model is fitted to the original data. Then, the feature values are permuted for each feature, and the model has fitted again. The difference between the model error on the permuted data and the model error on the original data is recorded. Finally, these differences are normalized so that the sum among all features is equal to 100Figure 2.7 shows an example of the output of feature permutation. In particular, the plot shows a permutation feature importance for a classification task based on accuracy performance. The plot was obtained using the Python package Scikit-learn (Pedregosa et al., 2011).

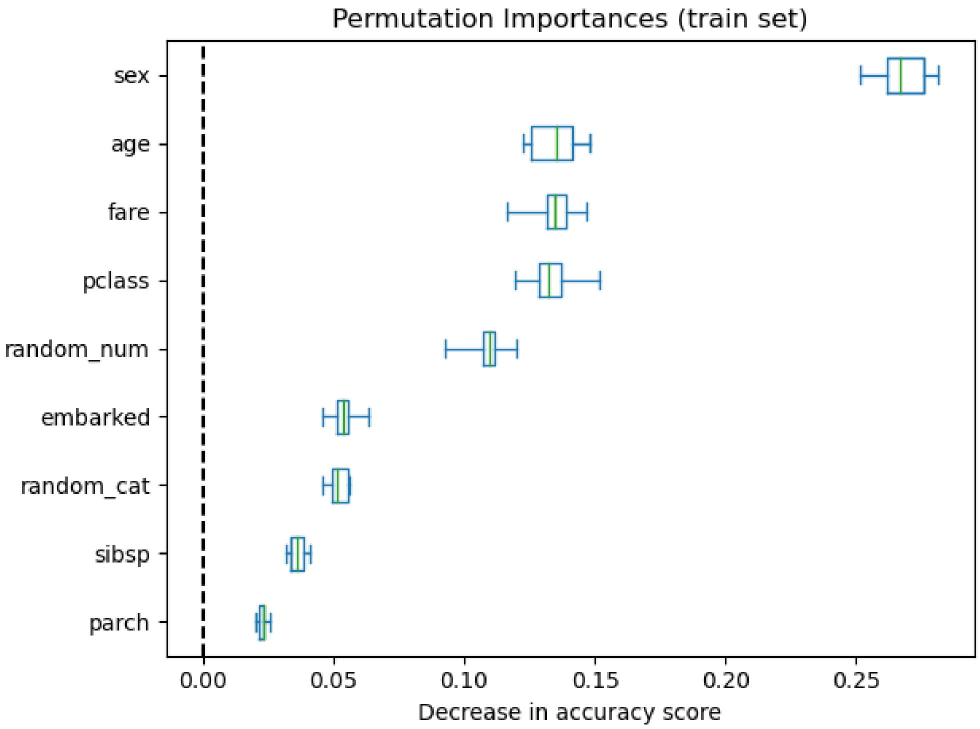


Figure 2.7: Variable importance through permutation of a Gradient Boosted Trees model trained on the Titanic dataset. The plot shows the different decreases in accuracy for each feature as well as its variability as a box plot.

2.4.2.4 Global surrogates

The last of the methods concerning global explanations are global surrogate models. Global surrogates are yet another method of ML interpretability that provides a global explanation for the model. Contrary to local surrogates methods such as LIME, which provide explanations only on an instance basis. Surrogates are trained models similar to the original model but provide more transparency and are interpretable. There are two main types of surrogates: decision trees and rule sets. Decision trees are a predictive model that can be used to model complex relationships between variables. Rule sets are if-then rules that can be used to make predictions. Rule sets are more difficult to interpret than decision trees, but they have the advantage of being more accurate. In general, global surrogates are often used in conjunction with local surrogates. Global surrogates can be used to understand how the model works and determine which input variables are most important to the model. They can also be used to improve the model by making it more transparent. Additionally, surrogate models can be used to

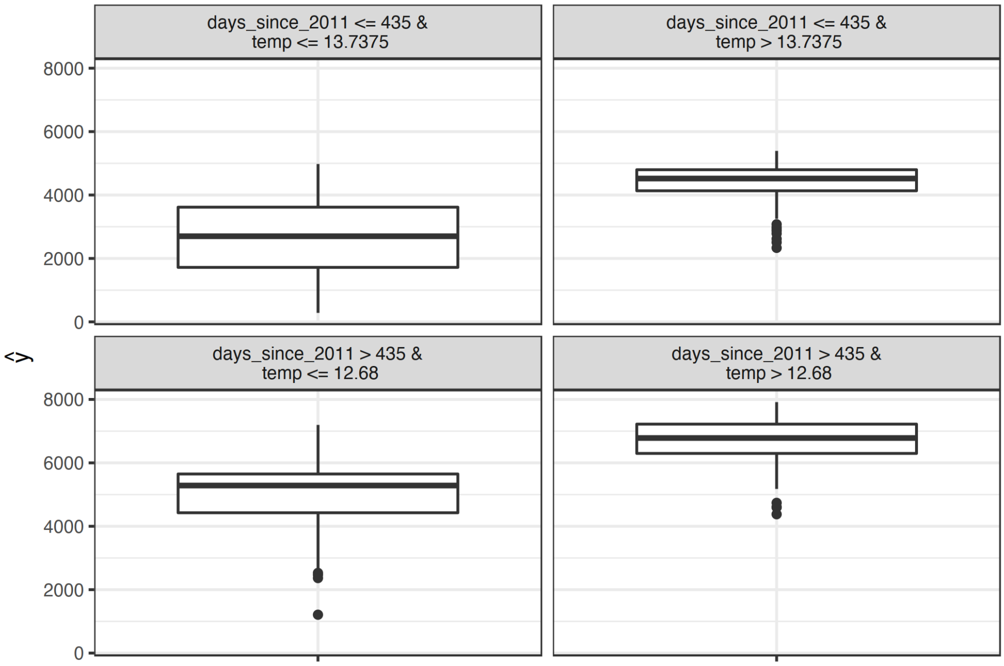


Figure 2.8: Global surrogate model of a Support Vector Machine trained on the Capital-Bikeshare dataset. The global surrogate model chosen is a decision tree. The plot depicts the model outcome for each bin created by the decision tree model.

generate explanations for the actions and decisions of AI systems. Finally, surrogate models can be used to improve the transparency and accountability of AI systems. One way to improve the performance of AI systems is to use surrogate models. Surrogate models are simplified models used to approximate more complex models' behavior. Surrogate models can be used to understand complex models' behavior and optimize AI systems' performance. Global surrogates are generally not provided by any package as they are very simple to implement. The only package implementing them is iml in R. In Figure 2.8, we can observe a global surrogate model made using a decision tree.

2.5 Challenges in interpretability

So far, we have discussed the different perspectives of model interpretability and the techniques used by academics and practitioners to explain complex models. Although these techniques shed some light on explaining the reasons behind models' outcomes, many challenges still need to be addressed. One untackled challenge in making ML models interpretable is that there can be a trade-off between the model's performance and interpretability. This trade-off affects the ML pipeline development in two different stages, namely during model training and during its interpretation. The modeler may enforce some rule simplicity during training resulting in a more interpretable and robust model, as performed by Repetto and La Torre, 2022. Nevertheless, at the same time, this will collide with its capability of capturing highly nonlinear patterns, a feature highlighted by Altman et al., 1994 as crucial. Therefore, in some cases, a more accurate ML model may be less interpretable than a less accurate one. Simultaneously, the modeler must add an interpretability layer capable of providing valuable information to stakeholders. The case of models with many features is an excellent example of how an interpretability approach might be misused. In this instance, a strategy like the PD or ALE will be useless because the stakeholders will have to look at a large number of plots. Feature importance measurements will be more appropriate for this type of assignment in this case. The central aspect of the stakeholder poses another challenge. Namely that there is no single definition of interpretability. In other words, what one person may find to be an interpretable model, another person may find incomprehensible. Another challenge of these techniques is that they are generally computationally expensive since they require multiple data permutations and model fitting. Plus, some of these techniques will not work with categorical data without imposing a particular order, as in the case of ALE. Furthermore, some of them are highly influenced by feature correlations such as PD and permutation feature importance. Last but not least, some ML models are too complex to be easily interpreted. Very complex Artificial Neural Networks, for example, can be extremely difficult to interpret. A well-known example is the usage of the saliency maps in Convolutional Neural Networks, which received many critiques in recent years (Tomsett et al., 2020).

2.6 Conclusion

ML is a field of AI that deals with constructing and studying algorithms that can learn from and make predictions on data. These algorithms are used in various ways, such as detecting fraud, making recommendations, and providing personalized search results. Despite their successes, ML models have several limitations. One is that it can be challenging to understand why a particular algorithm made a specific decision. This lack of interpretability can be a problem when ML is used in fields like medicine, where it is crucial to understand the rationale behind a diagnosis or treatment recommendation. Another limitation of ML models is that they can be biased for several reasons, such as the selection of data used to train the model or the assumptions made by the algorithm. These biases can lead to unfair decisions, such as denying a loan to someone likely to repay it. Or even facial recognition algorithms that are more accurate for white people than for people of color. Despite its limitations, ML is a powerful tool that is increasingly used to automate decision-making and improve the accuracy of predictions.

Many explanation methods have been proposed in the literature to revert interpretability into the modeling pipelines. This chapter discussed the most common model-agnostic methodologies. Model-agnostic methods are generally easier to use and can be applied regardless of the model trained but generally are less accurate and more computationally expensive.

It is essential to be aware of these limitations and use them in conjunction with other methods, such as human expertise, to ensure the best possible results and reliable interpretation.

Lost in a black-box? Interpretable Machine Learning for assessing Italian SMEs default 3

Lisa Crosato, Caterina Liberati, Marco Repetto

Applied Stochastic Models in Business and Industry. Wiley. *Revised.*

Academic research and the financial industry have recently shown great interest in Machine Learning algorithms capable of solving complex learning tasks, although in the field of firms' default prediction the lack of interpretability has prevented an extensive adoption of the black-box type of models. In order to overcome this drawback and maintain the high performances of black-boxes, this paper has chosen a model-agnostic approach. Accumulated Local Effects and Shapley values are used to shape the predictors' impact on the likelihood of default and rank them according to their contribution to the model outcome. Prediction is achieved by two Machine Learning algorithms (eXtreme Gradient Boosting and FeedForward Neural Networks) compared with three standard discriminant models. Results show that our analysis of the Italian Small and Medium Enterprises manufacturing industry benefits from the overall highest classification power by the eXtreme Gradient Boosting algorithm still maintaining a rich interpretation framework to support decisions.

3.1 Introduction

The European Union (EU) economy is deeply grounded in Small and Medium Enterprises (SMEs) which represent about 99.8% of the active enterprises in the EU-28 non-financial business sector (NFBS), accounting for almost 60% of value-added within the NFBS and fostering the EU workforce of the with two out of three jobs (European Commission, 2019).

Consequently, a wide literature has grown covering various economic aspects of SMEs, mainly focused on default prediction (for an up-to-date review see Ciampi et al., 2021), interesting for scholars as well as for practitioners such as financial intermediaries and for policy makers in their effort to support SMEs and to ease credit constraints to which they are naturally exposed (Cornille et al., 2019).

Whether for private credit-risk assessment or for public funding, independently of the type of data imputed to measure a firm health status, prediction of default should succeed in two aspects: maximise correct classification and clarify the role of the variables involved in the process. Most of the times, the contributions based on Machine Learning (ML) techniques neglect the latter aspect, often with better results with respect to standard parametric techniques that provide, on the contrary, a clear framework for interpretation. In other words ML techniques rarely deal with *interpretability* which, according to a recent document released by the European Commission, should be kept ”in mind from the start” (Commission et al., 2019).

Interpretability is central when applying a model in practice, both in terms of managerial decisions and compliance: it is a fundamental requisite to bring a model into production Coussement and Benoit, 2021. Interpretable models allow risk managers and decision makers to understand their outcome and to knowingly take courses of actions. The European Commission itself encourages organizations to build trustworthy Artificial Intelligence (AI) systems (including ML techniques) around several pillars: one of them is transparency, which encompasses traceability, explainability and open communication about the limitations of the AI system (High-Level Expert Group on Artificial Intelligence, European Commission, 2020).

Accordingly, ML models -no matter how good in classifying default- should be made readable to avoid that their inherent uninterpretable nature may prevent their spreading in the literature on firms’ default prediction as well as their use in other contexts regulated by transparency norms.

This work tries to fill this gap by applying two different kind of ML models, FeedForward Neural Networks (Haykin, 1999) and eXtreme Gradient Boosting (T. Chen & Guestrin, 2016), to Italian Manufacturing SMEs’ default prediction, with a special attention to interpretability. Italy represents an ideal testing ground for SMEs default prediction since its economic framework is more extensively configured by firms up to this size than the average of EU countries (European Commission, 2019). Default was assessed on the basis of the firms’ accounting information retrieved from Orbis, a Bureau van Dijk (BvD) dataset.

The main original contribution of the paper is to address ML models’ interpretability in the context of default prediction. Our approach is based on model agnostic-techniques and adds Accumulated Local Effects (ALEs, Apley & Zhu, 2020) to the Shapley values already applied in (Bussmann et al., 2021). Using these techniques we can rank the variables in terms of their contribution to the classification and determine their impact on default prediction.

Robustness of the ML models hyperparameters was taken care of by Montecarlo Cross-Validation and substantial class imbalance between defaulted and survived firms was reduced through undersampling of the latter into the cross-validation training sets. Another contribution of the paper is the benchmarking of the ML models' outcome with Logistic, Probit and with Binary Generalized Extreme Value Additive (BGEVA) classifications, both according to standard performance metrics and to the role played by the input features. Moving a step forward with respect to the current use of ALEs, we fully exploit the tool and supply them also for the parametric models, in order to unfold what is compressed within the single variables coefficients and significance and guarantee a common ground for comparison.

We obtain a few interesting results. First, eXtreme Gradient Boosting (XGBoost) outperformed the other models mainly for total classification accuracy and default prediction rate. Second, the impact of the variables assessed by XGBoost is fully consistent with the economic literature, whereas the same cannot be said for its competitors. Thanks to the ALEs framework for interpretability, risky thresholds, non-linear patterns and other additional insights emerge for predictors even in standard models.

The remainder of the paper is organized as follows. Section 2 gives an overview of the (necessarily) recent literature concerning ML intepretability. Section 3 provides a description of the dataset and of the features we use throughout the modelling. Section 4 discusses our methodology, briefly reviewing the models fundamentals, the techniques employed for interpretability and the research design. Section 5 presents the results and discusses the most relevant findings. Section 6 concludes.

3.2 Literature review

The ability to predict corporate failure has been largely investigated in credit risk literature. On the one hand, the academic interest in the topic has increased after the global financial crisis (2007-2009) and is being renewed today due to the current pandemic impact on the companies of all sizes (Berg, 2007; Ciampi et al., 2021). On the other hand, a good part of the financial industry has shown great attention to statistical algorithms that prioritize the pursuit of predictive power. Such a trend has been registered by recent surveys, showing that credit institutions are gradually embracing ML techniques in different areas of credit risk management, credit scoring and monitoring (Alonso & Carbó, 2020; Bank of England, 2019; Institute of International Finance, 2020). Among all, the biggest annual growth in the adoption of highly performing algorithms has been observed in the SMEs sector (Institute of International Finance, 2019).

For these reasons, new modeling techniques have been successfully employed in predicting SMEs default, including Deep Learning (Mai et al., 2019), Support Vector Machines (Gordini, 2014; L. Zhang et al., 2015), Neural Networks (Ciampi & Gordini, 2013) and Hazards models De Leonardis and Rocci, 2008, 2014, to name only a few. However, they have been applied mainly in order to improve classification accuracy with respect to the standard linear models, supporting decisions through reduced uncertainty but leaving somewhat unsolved the issue of interpretability. But the latter is no longer a negligible aspect, both for academic research and for management of regulated financial services: it has become overriding, since the European Commission and other European Institutions have released a number of regulatory standards on Machine Learning modeling.

The Ethics Guidelines for Trustworthy AI (Commission et al., 2019) and the Report on Big Data and Advanced Analytics (European Banking Authority, 2020) illustrate the principle of explicability of ML algorithms which must be transparent and fully interpretable to the ones directly and indirectly affected. Indeed, as the Commission points out, predictions, even accurate, without explainability measures are unable to foster responsible and sustainable AI innovation. The pillar of transparency (fourth among seven), somewhat combines explainability and interpretability of a model, referring to interpretability as the "concept of comprehensibility, explainability, or understandability" (High-Level Expert Group on Artificial Intelligence, European Commission, 2020).

The difference in meaning between interpretability and explainability, synonymous in the dictionary, has been addressed by the recent ML literature which recognizes the two words a conceptual distinction related to different properties of the model and knowledge aspects (Doran et al., 2017; Lipton, 2018). A clear indication about the distinction is given by Montavon et al., 2018 that defines interpretation as a mapping of an abstract concept into a domain that the human expert can perceive and comprehend and explanation as a collection of features of the interpretable domain that have contributed to produce a decision. Roughly speaking, interpretability is defined as the ability to spell out or to provide the meaning in understandable terms to a human (Doshi-Velez & Kim, 2017; Guidotti et al., 2018), whereas explainability is identified as the capacity of revealing the causes underlying the decision driven by a ML method (Arrieta et al., 2020).

There are several approaches to ML interpretability in literature, classified in two main categories: ante-hoc and post-hoc methods. Ante-hoc methods employ intrinsically interpretable models (e.g., simple decision trees or regression models, also called white-box) characterized by a basic structure. They rely on model-specific interpretations depending on examination of internal model parameters. Post-hoc methods instead provide a reconstructed interpretation of decision rules produced by a black-box model in a reverse engineering logic (Du et al., 2019; Ribeiro et al., 2016b), reckoning on model-agnostic interpretation where internal model parameters are not inspected.

So far, ante-hoc approaches were widely used in the SMEs default prediction literature that counts contributions employing mainly white-box models as Logistic regression (see for example Ciampi, 2015; Lin et al., 2012; Modina & Pietrovito, 2014), Survival analysis (El Kalak & Hudson, 2016; Gupta et al., 2018; Holmes et al., 2010) or Generalised Extreme Value regression (Calabrese et al., 2016). The empirical evidences and the variables' effect on the outcome are interpreted in an inferential testing setting, so that the impact of the predictors and the results' implications are always clear to the reader.

On the contrary, post-hoc methods have been rarely used in this field and comprehend Partial Dependence (PD) plots (Friedman, 2001), Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016a) and the SHAP (Lundberg & Lee, 2017), all of them providing detailed model-agnostic interpretation of the complex ML algorithms employed either focusing on a global or a local scale. Jones and Wang, 2019, Sigrist and Hirnschall, 2019 and Jabeur et al., 2021 used the PD to identify the relevant variables' subset and to measure the change of the average probability of default with respect to the single features. A PD-based framework for making transparent, auditable, and explainable black-box models both at the global level and for single instances was developed in the ambit of credit scoring by Bücker et al., 2021b. LIME and SHAP were applied in Stevenson et al., 2021; Yıldırım et al., 2021 to rank the variables and to provide their impact on the output prediction respectively.

Alternative strategies to enhance interpretability combine the above approaches to get the most out of both. Surrogate models emulate the black-boxes with one or more white-boxes to clarify the output of the former Liberati et al., 2017 Glynn, 2022. Another strand of literature links together complex ML models for feature selection/transformation and white-box models for fitting/interpretation in two-layer frameworks C. Chen et al., 2021; Gosiewska et al., 2021. The rationale under these combinations is to exploit each class of models in what they do better: black-boxes for coping with high-dimensionality and non-linearities and white-boxes for plain explanations, treating all issues within and between data ex-ante and leaving thus space for simpler models ex-post. This approach seems promising, although evidences on its advantages have been so far limited.

This paper contributes to the literature investigating global level interpretability and to the literature on SMEs default: we compare black-box with white-box models on both performance and interpretability domains, thus bridging both sides of the empirical work in the field. We do this by fully exploiting post-hoc methods on all models. Building on a set of features recommended by experts from the well-established literature on firm default, we employ the Accumulated Local Effects (ALEs, Apley & Zhu, 2020), a model-agnostic technique that represents a suitable alternative to PDs when the features are highly correlated, without providing incoherent values Gosiewska et al., 2021. Since ALEs are a newest approach, their usage is still limited and not yet spread in the bankruptcy prediction area. An isolate application to the recovery rate forecasting of non-performing loans can be encountered in the credit risk field (Bellotti et al., 2021).

3.3 Data Description

The data of this study are retrieved from BvD-Orbis database, which provides financial and accounting ratios from balance sheets of the European private companies. We have restricted our focus on Italian manufacturing SMEs for several reasons. Italy is the second-largest manufacturing country in the EU (Bellandi et al., 2020) and this sector generates more than 30% of the Italian GDP (Eurostat, 2018). Differently from SMEs in other EU countries, Italian SMEs trade substantially more than large firms, the manufacturing sector, in particular, driving both imports and exports. Moreover, according to Ciampi and Gordini, 2013, predictive models have better performances when trained for a specific sector in that pooling heterogeneous firms is avoided.

To define our sample, we filtered the database both by country and NACE codes (from 10 to 33) and we employed the European Commission definition (EU, 2003) of Small and Medium Enterprises. We retrieved only firms with an annual turnover of fewer than 50 million euros, a number of employees lower than 250 and a total balance sheet of fewer than 43 million euros. Among those, we classified as defaults all the enterprises that entered bankruptcy or a liquidation procedure, as well as active companies that had not repaid debt (default of payment), active companies in administration or receivership or under a scheme of the arrangement, (insolvency proceedings), which in Orbis are also considered in default. Consistently with the literature, we excluded dissolved firms that no longer exist as a legal entity when the reason for dissolution is not specified (Altman et al., 2010; Andreeva et al., 2016). This category in fact encompasses firms that may not necessarily experience financial difficulties. The resulting dataset contains 105,058 SMEs with a proportion of 1.72% (1,807) failed companies.

The accounting indicators, which refer to 2016 to predict the firm status in 2017, have been selected among the most frequently used in the SMEs default literature and are the following (Altman et al., 2010; Andreeva et al., 2016; Calabrese et al., 2016; Ciampi & Gordini, 2013; Jones et al., 2015; Lin et al., 2012; Michala et al., 2013; Succurro, Mannarino, et al., 2014):

- Cash flow: computed as net income plus depreciations;
- Gearing ratio: computed as the ratio between total debt and total assets;
- Number of employees, as a size measure from an input perspective;
- Profit margin: measured as profit/loss before tax over the operating revenue;
- ROCE: computed as profit/loss before tax over capital employed, which is given as total assets minus current liabilities;
- ROE: computed as profit/loss before tax over shareholders' funds;
- Sales: in thousands Euro, measuring the output side of firm size;
- Solvency ratio: computed as shareholders' funds over total assets;
- Total assets: in thousands Euro, as a measure of total firm resources.

As a quick preview of the expected relationship between the single predictors and the likelihood of default, we have computed the averages and standard deviations of the variables for survived and defaulted firms (see table 3.1).

In line with Andreeva et al., 2016, we can see on average weakest liquidity, smallest size and deficient leverage for defaulted firms.

The Profit margin is higher for surviving firms, whereas the remaining profitability indexes, ROE and ROCE, show a larger median and mean among defaulted firms respectively. They should both be negatively related to default, although some studies found ROCE's impact non-significant coherently with the low-equity dependency of small businesses (Giudici et al., 2020), while others attest its positive effect on default with a caveat for large values (Calabrese et al., 2016). We will get more valuable insights into these profitability indicators when discussing the models' outcome.

3.4 Methodology

3.4.1 White-box versus black-box models

The models we apply can be broadly classified as white-box, or interpretable, and black-box but post-hoc interpretable in the model-agnostic framework.

Table 3.1: Summary statistics by survived and failed firms

	Survived				
Variable	Min	Mean	St. dev.	Median	Max
Cash flow	-43,142.00	236.802	934.877	55.000	89,591.000
Gearing ratio	0.00	24.807	23.093	22.198	99.882
N. of employees	1.00	16.506	24.385	9.000	249.000
Profit margin	-87,700.00	-2.736	610.488	2.673	141,300.000
ROCE	-86,250.00	12.335	516.765	7.955	114,233.333
ROE	-35,961.11	23.020	314.135	17.647	39,500.000
Sales	1.00	3,427.163	6,301.229	1,165.000	49,995.000
Solvency ratio	-99.97	27.101	24.315	22.400	100.000
Total assets	1.00	3,904.129	12,098.087	1,194.000	1,758,577.000
	Failed				
Variable	Min	Mean	St. dev.	Median	Max
Cash flow	-19,497.00	-278.521	1,636.028	-15.000	41,186.000
Gearing ratio	0.00	22.166	26.010	12.594	98.134
N. of employees	1.00	11.080	19.531	5.000	228.000
Profit margin	-87,762.50	-106.845	2,190.012	-9.677	21,700.000
ROCE	-23,600.00	66.367	2,284.001	5.818	90,800.000
ROE	-28,800.00	7.146	971.112	32.692	5,366.667
Sales	1.00	1,259.695	2,940.010	380.000	32,522.000
Solvency ratio	-99.43	-1.044	37.342	3.080	100.000
Total assets	1.00	1,921.689	5,149.559	526.000	110,501.000

In the first category, Logistic Regression (LR) and Probit were selected among the most recurrent models in the economics literature, where the accent on the factors impacting default is certainly of primary importance. These models frequently serve as a benchmark for classification when a new method is proposed. The third model, BGEVA (Calabrese et al., 2016), comes from the Operational Research literature and is based on the quantile function of a Generalized Extreme Value random variable. The main strengths of BGEVA are robustness, accounting for non-linearities and preserving interpretability.

The black-box models we use are XGBoost and FeedForward Neural Networks (FANN). These models are by nature uninterpretable since the explanatory variables pass multiple trees (XGBoost) or layers (FANN), thus generating an output for which an understandable explanation cannot be provided.

The XGBoost algorithm was found to provide the best performance in default prediction with respect to LR, Linear Discriminant Analysis, and Artificial Neural Networks (Bussmann et al., 2021; Petropoulos et al., 2019). The algorithm builds a sequence of shallow decision trees, which are trees with few leaves. Considering a single tree one would get an interpretable model taking the following functional form:

$$f(x) = \sum_{m=1}^{M} \theta_m I(x \in R_m) \tag{3.1}$$

where M covers the whole input space with $R_1, ...R_M$ non-overlapping partitions, $I(\cdot)$ is the indicator function, and θ_m is the coefficient associated with partition R_m.

In this layout, each subsequent tree learns from the previous one, thus improving the prediction (Friedman, 2001).

As a competing black-box model we chose the FANN, which is widely used and well performing in SMEs' default prediction (Ciampi et al., 2021) and in several works on retail credit risk modeling (Baesens et al., 2003; West, 2000; West et al., 2005). FANN consists of a direct acyclic network of nodes organized in densely connected layers, where inputs, weighted and shifted by a bias term, are fed into the node's activation function and influence each subsequent layer until the final output layer. In a binary classification task, the output of a single layer FANN can be described as in Arifovic and Gencay, 2001 by:

$$f(x) = \phi\left(\beta_0 + \sum_{j=1}^{d} \beta_j G\left(\gamma_{j0} + \sum_{i=1}^{p} \gamma_{ji} x_i\right)\right) \tag{3.2}$$

where G is the activation function, in our case $G(x) = \frac{1}{1+e^{-\alpha x}}$, β and γ represent weights and biases at each layer, whereas $\phi(\cdot)$ is the network output function that in our case is also a sigmoid function as for $G(\cdot)$.

3.4.2 Model-agnostic interpretability

To achieve the goal of interpretability, we make use of two different and complementary model-agnostic techniques. First, we use the global Shapley Values (Shapley, 1953) to provide comparable information on the single feature contributions to the model output. Global Shapley Values have been already proposed in the SMEs default prediction literature by Bussmann et al., 2021.They differ from standard feature importance metrics based on feature permutation because of feature attribution evaluation based on possible coalitions capturing feature interactions Covert et al., 2020. Although model-agnostic, they share some of the axioms that characterize gradient-based interpretability methods such as Integrated Gradients Sundararajan et al., 2017.

However, global Shapley Values do not provide any information about the shape of the variable effects, therefore we resort to ALEs (Apley & Zhu, 2020). ALEs, contrary to Shapley Values, offer a visualization of the path according to which the single variables impact on the estimated probability of default.

To further clarify the improvement that ALEs bring to interpretability in our setting, we briefly contextualize the method and outline its fundamentals.

The first model-agnostic approach for ML models' interpretation to appear in the literature was Partial Dependence (PD), proposed by Friedman, 1991 in the early '90s. PD plots evaluate the change in the average predicted value as specified features vary over their marginal distribution (Goldstein et al., 2015). In other words, they measure the dependence of the outcome on a single feature when all of the others are marginalized out. Since their first formulation, PD plots have been used extensively in many fields but seldom in the credit risk literature, with a recent application by Ozgur et al., 2021.

One of the main criticisms moved to PD concerns its managing the relationships within features. The PD evaluation on all the possible feature configurations carries the risk of computing points outside the data envelope: such points, intrinsically artificial, can result in a misleading effect of some features when working on real datasets.

Due to this fallacy, and because of the renewed interest in complex deep learning models as Artificial Neural Networks, many new methodologies have been proposed. With Average Marginal Effects (AMEs), Hechtlinger, 2016 suggested to condition the PD to specified values of the data envelope. Ribeiro et al., 2016a went the opposite direction presenting a local approximation of the model through simpler linear models, the so-called Local Interpretable Model-agnostic Explanations (LIME). In subsequent research, they also worked on rule-based local explanations of complex black-box models (Ribeiro et al., 2018). Shapley Additive exPlanations (SHAP) was introduced by Lundberg and Lee, 2017 to provide a human understandable and local Shapley evaluation.

In this framework, ALEs constitute a further refinement of both PD and AMEs. They avoid the PD plots-drawback of assessing variables' effects outside the data envelope, generally occurring when features are highly correlated (Apley & Zhu, 2020), as in the case of many accounting indicators (Altman et al., 2010; Ciampi, 2015). Furthermore, ALEs do not simply condition on specified values of the data envelope as AMEs do, but take first-order differences conditional on the feature space partitioning, eventually eliminating possible bias derived from features' relationships.

Specifically, computing the ALE implies the evaluation of the following type of function:

$$ALE_{\hat{f},S}(x) = \int_{z_{0,S}}^{x} \left[\int \frac{\delta \hat{f}(z_S, X_{\setminus S})}{\delta z_S} d\mathcal{P}(X_{\setminus S}|z_S) \right] dz_S - constant \qquad (3.3)$$

where $\hat{f}$ is the black-box model, S is the subset of variables' index, X is the matrix containing all the variables, x is the vector containing the feature values and z identifies the boundaries of the K partitions, such that $z_{0,S} = min(x_S)$.

The expression in equation 3.3 is in principle not model-agnostic as it requires accessing the gradient of the model: $\nabla_{z_S} \hat{f} = \frac{\delta \hat{f}(z_S, X_{\setminus S})}{\delta z_S}$ but this is not known or even non existent in certain black-boxes. As a replacement, finite differences are taken to the boundaries of the partitions, z_{k-1} and z_k.

Hence, the resulting formula to evaluate ALEs is:

$$ALE_{\hat{f},S}(x_S) = \sum_{k=1}^{k_S(x)} \frac{1}{n_S(k)} \sum_{i: x_S^{(i)} \in N_S(k)} \left[\hat{f}(z_{k,j}, x_{\setminus S}^{(i)}) - \hat{f}(z_{k-1,j}, x_{\setminus S}^{(i)}) \right] - \frac{1}{n} \sum_{i=1}^{n} ALE_{\hat{f},S}(x_S^{(i)}) \tag{3.4}$$

The replacement of the constant term in equation 3.3 by $-\frac{1}{n}\sum_{i=1}^{n} ALE_{\hat{f},S}(x_S^{(i)})$ in equation 3.4 centers the plot, which is something missing in PD. This makes it clear that, by evaluating predictions' finite differences conditional on S and integrating the derivative over features S, ALEs disentangle the interaction between covariates. This way the main disadvantage of PD is solved.

3.4.3 Research design

Our research design has been carried out according to Lessmann et al., 2015. We split the initial dataset into training (70%) and test (30%) sets (Andreeva et al., 2016; Gordini, 2014; James et al., 2013). Then, through the Monte Carlo Cross-Validation procedure (Xu & Liang, 2001), we estimate the models parameters and validate the estimated rules. More in detail, at each iteration we create a sub-training set and a validation set via random sampling without replacement so that the models learn from the training set whereas the assessment, based on performance metrics, is done on the validation set. This way, we also tune the hyperparameters of the algorithms when necessary.

The training set serves as well to compute the Shapley values, based on the optimal rule, and to calculate the ALEs with corresponding bootstrap non-parametric confidence intervals (Apley & Zhu, 2020; Davison & Kuonen, 2002). Finally, we evaluated the models' performance on the test set.

We took into account also the severe unbalance in favour of survived firms to avoid over-classification of the majority class (Baesens et al., 2021). After testing several techniques for addressing imbalance Veganzones and Séverin, 2018 in the learning phase, we have chosen random undersampling, which consists of sampling randomly among the majority class observations to achieve balancing*.

*Complete results about the resampling schemes are reported in Appendix A

Obviously the undersampling scheme was applied only to the training data, to avoid over-optimistic performance metrics on either the validation or the test set (Gong & Kim, 2017; Santos et al., 2018).

3.5 Results

The results are organized according to the performance and interpretation of the five models. The performance is measured by the proportion of failed and survived firms correctly identified (sensitivity and specificity) together with the Errors of the first and second type (E-I and E-II, respectively) as well as by four global performance metrics: the Area Under the Receiver Operating Curve (AUC), the H-measure, the Brier Score (BS) and the Kolmogorov-Smirnov statistic (KS) (see Table 3.2). We chose these indicators for two reasons: they are popular in credit scoring and evalute three different aspects of the discriminating rule: KS assess the correctness of categorical predictions, the AUC and H-measure assess discriminatory ability, and the BS assesses the accuracy of probability predictions (Lessmann et al., 2015).

Second, we cross-compare the role and weight of the variables among models and contextualize the results within the literature. The post-hoc interpretation of the black-box models is based on the Shapley values and ALEs. We report the ALEs also for interpretable models to exploit a common basis for predictors comparison without incurring in the ”p-value arbitrage” when evaluating white-box models via p-values and ML models via other criteria (Breeden, 2020).

3.5.1 Performance

All competing models offer fair correct classification rates, but the ones that score globally best are black-box models, in terms of all metrics. The FANN reaches the highest H-measure and specificity while it's last as far as correct classification of default is concerned (with a sensitivity not reaching 70%, see Table 3.2). On the contrary, the XGBoost algorithm provides the best default prediction (showing, by far, the largest sensitivity) with a reasonable classification of survivors, as well as the highest global metrics but the H-measure, which ranks it second.

The interpretable models are ranked consistently by AUC and H-measure in the following order: BGEVA, LR and Probit, whereas the Brier score and the KS provide alternative rankings. Anyway, these results confirm the trade-off between performance and interpretability highlighted in previous works on Italian SMEs (Ciampi & Gordini, 2013).

All in all, undersampling the training set has a balancing effect on the rate of correct prediction for either class Veganzones and Séverin, 2018. This improves

Table 3.2: Models' performances on the test set.

Model	Sensitivity	Specificity	E-I	E-II	H	AUC	BS	KS
FANN	0.694	**0.829**	**0.171**	0.306	**0.391**	0.827	0.187	0.501
XGBoost	**0.821**	0.719	0.281	**0.179**	0.383	**0.843**	**0.146**	**0.552**
BGEVA	0.752	0.727	0.273	0.248	0.331	0.819	0.178	0.481
LR	0.745	0.736	0.264	0.246	0.303	0.809	0.151	0.483
Probit	0.738	0.737	0.263	0.262	0.299	0.809	0.190	0.448

global classification not only through FANN, but also when applying Logistic Regression, as compared for instance with the results on the same kind of variables of Ciampi and Gordini, 2013 or Modina and Pietrovito, 2014, the latter for both techniques.

3.5.2 Interpretation

Most of the variables have non-significant effects on the probabilities of default estimated by white-box models, as long as these effects are ascertained by p-values (table 3.3). Three variables display a significant and non-null coefficient, no matter the model: Sales[†], the Solvency Ratio and the Cash flow, all with an adverse effect on the probability of default.

The negative impact exerted by Sales on default, recurrent in many works, is not surprising since Sales is one of the main proxies of a company's size and largest firms tend to overcome demand shocks better than smaller firms (Ciampi, 2015; Psillaki et al., 2010), which is also consistent with the means reported in Table 3.1 for the two groups of firms. Apparently, the size effect is captured exclusively by the output-side variable since the other size proxies, the Number of employees and the Total Assets, both highly correlated with Sales (Jabeur et al., 2021), do not have instead significant effects.

As expected, firms with a strongest leverage (Solvency ratio) and higher liquidity (Cash flow) are less likely to default (Andreeva et al., 2016; Michala et al., 2013).

Notice that profitability measures, rather unexpectedly, do not impact on the probability of default according to significance criteria. BGEVA signals a significant ROCE but the estimated coefficient is zero. To gain additional insights, we can turn to the ALEs: the three common significant variables can be interpreted likewise since they all follow a non-flat path. However, while the models' coefficients for the Solvency ratio and Cash flow describe almost neutral effects on the

[†] In the text we refer to Sales, Total Assets and Number of Employees for readability reasons. However, we have transformed them through logarithms as common in the literature (Altman et al., 2010; Altman & Sabato, 2007; Psillaki et al., 2010)

outcome (with an odds-ratio of 1 for the Cash flow in the Probit model, see Table 3.3), post-hoc interpretation reveals a marked decreasing effect for the former and a clear non-linear pattern for the latter. On the other hand, and contrary to the p-value reading, we can observe that Profit margin and ROE do reduce the probability of default, whereas ROCE increases it according to the LR, Probit (see figure 3, panels (a) and (b) respectively) and to the BGEVA model (figure 3.2).

Another counterintuitive effect is revealed by the ALEs plot of the Profit margin for the Probit (figure 3, panel (b)), which could partially explain the sub-optimal classification performance of the same model.

The picture changes when it comes to black-box models. Global Shapley values indicate (figure 3.3) that both FANN and XGBoost predictions are influenced mainly by Profit margin. This outcome is further clarified by the average change in the model output corresponding to increasing values of the variable, represented by ALEs (figure 3.4).

Table 3.3: Estimates and summary statistics for the Probit, Logistic Regression, and BGEVA models on the test set (Significant variables in bold).

	Probit Model			Logistic Regression			BGEVA Model		
	Odds ratio	Std. error	p-value	Odds ratio	Std error	p-value	Estimate	Std. error	p-value
(Intercept)	6.195	0.134	0.000	21.256	0.233	0.000	2.087	0.137	0.000
Cash flow	**1.000**	0.000	**0.000**	**0.999**	0.000	**0.000**	**-0.001**	0.000	**0.000**
Gearing ratio	1.000	0.001	0.713	1.001	0.002	0.594	0.000	0.001	0.756
Number of employees	1.081	0.078	0.319	1.135	0.131	0.332	0.033	0.081	0.683
Profit margin	1.000	0.000	0.535	1.000	0.000	0.947	0.000	0.000	0.800
ROCE	1.000	0.000	0.256	1.000	0.000	0.302	**0.000**	0.000	**0.027**
ROE	1.000	0.000	0.240	1.000	0.000	0.275	0.000	0.000	0.285
Sales	**0.526**	0.066	**0.000**	**0.316**	0.120	**0.000**	**-0.637**	0.064	**0.000**
Solvency ratio	**0.985**	0.001	**0.000**	**0.973**	0.002	**0.000**	**-0.015**	0.001	**0.000**
Total assets	1.044	0.064	0.503	1.166	0.112	0.172	0.090	0.066	0.174

The ALEs of either model show a downward sharp jump in the probability of default when moving from negative to positive values of Profit margin, with no further decrease in the probability of default as the ratio increases, revealing a clearly decreasing effect of this ratio on the probability of default, as previously found by Altman et al., 2010, Andreeva et al., 2016 and Petropoulos et al., 2019.

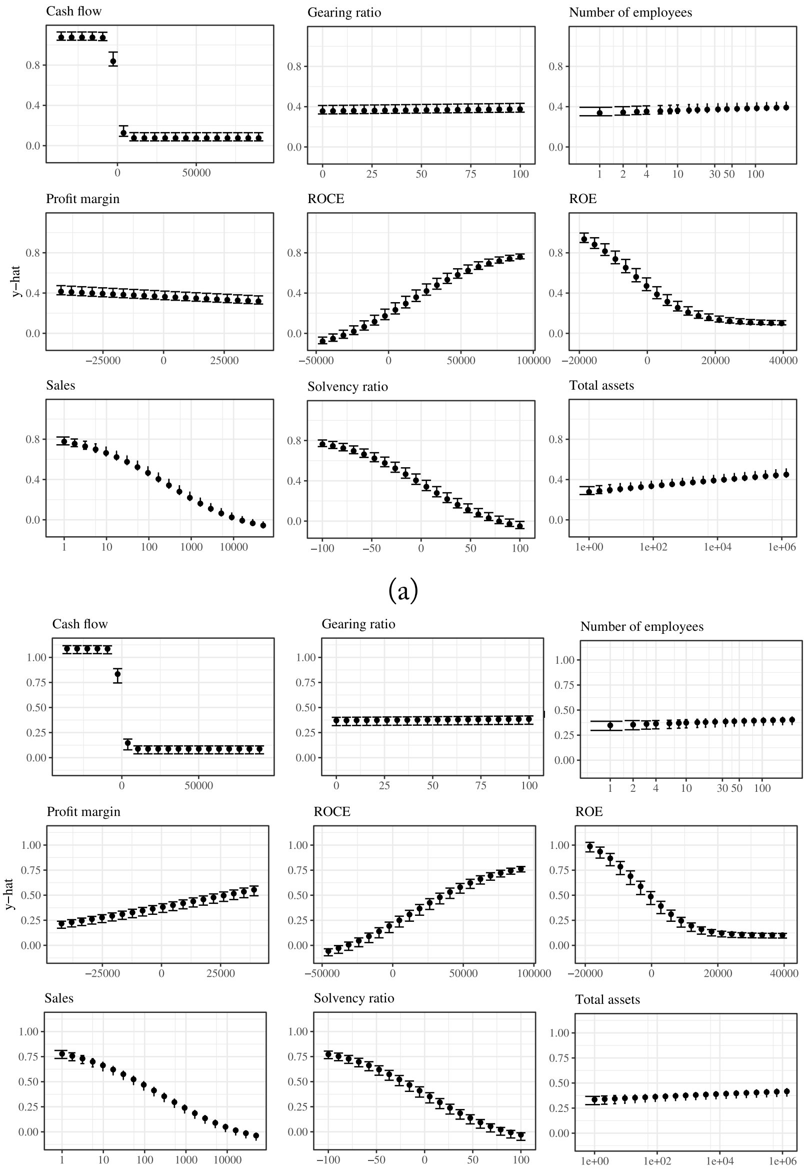
The negative impact of Sales, already emerged in the white-box models, is confirmed to a minor extent by both FANN and XGBoost (second and third important variable respectively according to Shapley values). However, the pattern of the estimated default probabilities for Sales is unlike: a smooth path with no evident plateauing effect in FANN and a first sudden decrease around 100.000 euros and a second drop around 316.000 euros in XGBoost.

A remarkable difference with respect to the white-box models are the sways of Total assets and the Number of employees. Total assets is the third important variable for FANN according to the Shapley values and seems to increase the probability of default judging from ALEs. On the contrary, the variable shows no importance in the prediction by XGBoost (Shapley value close to 0 and flat ALE). A positive impact of Total assets on the probability of default is anomalous, though shared by other scholars (Andreeva et al., 2016), in the light of our descriptive statistics and referring to the literature on firm demography, where exit is usually associated to less tangible assets (Michala et al., 2013). This effect could be associated to the same found by other authors in the credit scoring literature. In that case a non-linear behaviour could be accounted to the fact that creditors do pursue firms with larger assets with the hope to get back the money they have lent, whereas firms with low tangible assets are less worth being pursued (Altman et al., 2010; El Kalak & Hudson, 2016).

A somewhat opposite situation regards the Number of employees: FANN attributes scarce weight to this variable whereas XGBoost highlights its moderate impact (fourth important variable in the Shapley values) and a decrease in the probability of default around 5 employees. The XGBoost algorithm seems therefore able to capture separate and concordant effects of two firm size variables, respectively on the input and the output side, in decreasing the probability of default, contrary to other empirical applications (Andreeva et al., 2016).

The Solvency ratio behaves similarly to Sales, for which the XGBoost shows a plateauing effect after 0 that the FANN does not point out. However, its importance, measured by the Shapley values, differs between the two algorithms since it is the second most relevant variable for XGBoost and the fourth relevant variable in the FANN.

The Cash flow, the third variable impacting on default according to white-box models, maintains a negative sign also in FANN, while it is not relevant in the XGBoost model (as in Michala et al., 2013). The Gearing ratio, ROCE and ROE are of little consequence for XGBoost output and even less for the FANN



(a)

(b)

Figure 3.1: Accumulated Local Effects of the LR (a) and Probit (b) models with superimposed bootstrap 5%-95% confidence intervals. The ALEs for Sales, Total Assets and Number of Employees are calculated on log-transformation of the variables but depicted on anti-log values to enhance readability.

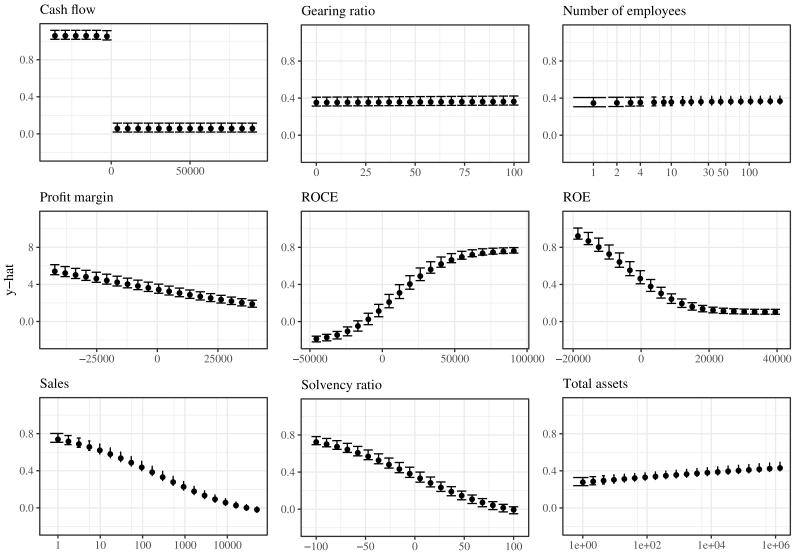


Figure 3.2: Accumulated Local Effects of the BGEVA model with superimposed bootstrap 5%-95% confidence intervals. The ALEs for Sales, Total Assets and Number of Employees are calculated on log-transformation of the variables but depicted on anti-log values to enhance readability.

according to the Shapley values and to overlapping bootstrap confidence intervals in Figure 3.4, except for the FANN's ALEs plot that displays ROCE (however small its importance) as enhancing the probability of default, which is in line with part of the literature (Calabrese et al., 2016 pointed out ROCE's positive effect). Another part of the literature instead found it non-significant (Giudici et al., 2020).

To summarize, blurry effects of one or more variables are encountered for the FANN model (Total assets and ROCE) and for all the white-box models (ROCE for all of them, Profit margin only for the Probit). Considering the prominent roles assigned by FANN to both Sales and Total assets, it seems that these two variables compensate each another in the wrong way, resulting in a the lowest correct classification of defaulted firms among the competing models.

An interesting puzzle remains regarding the completely different ranking in the importance of variables according to white versus black-box models. Keeping performance in mind, we should consider what emerges from the interpretation of the XGBoost output, attributing the highest sensitivity achieved to an evaluation of the interplay among the variables which results more effective in predicting default.

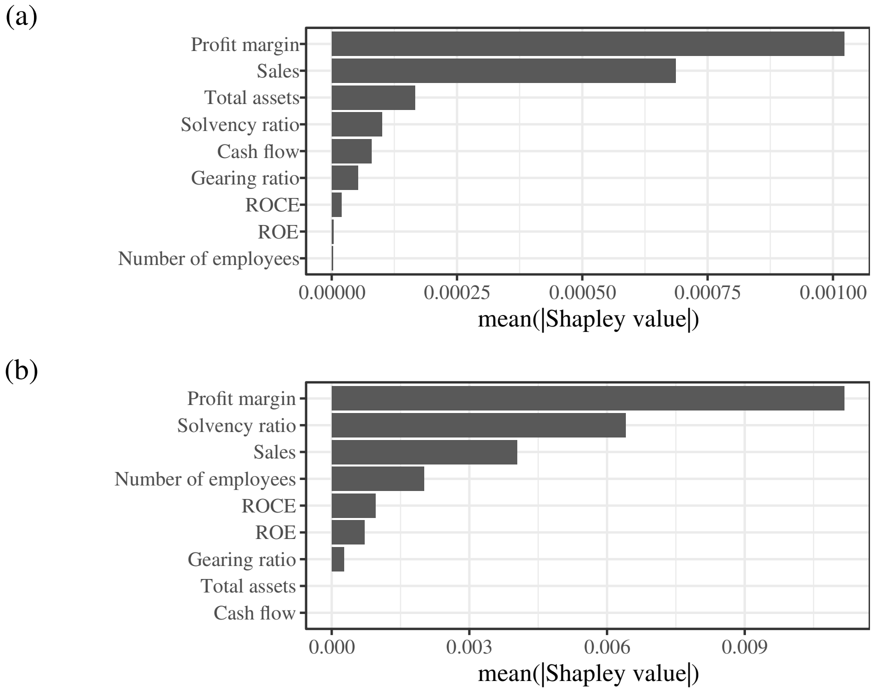
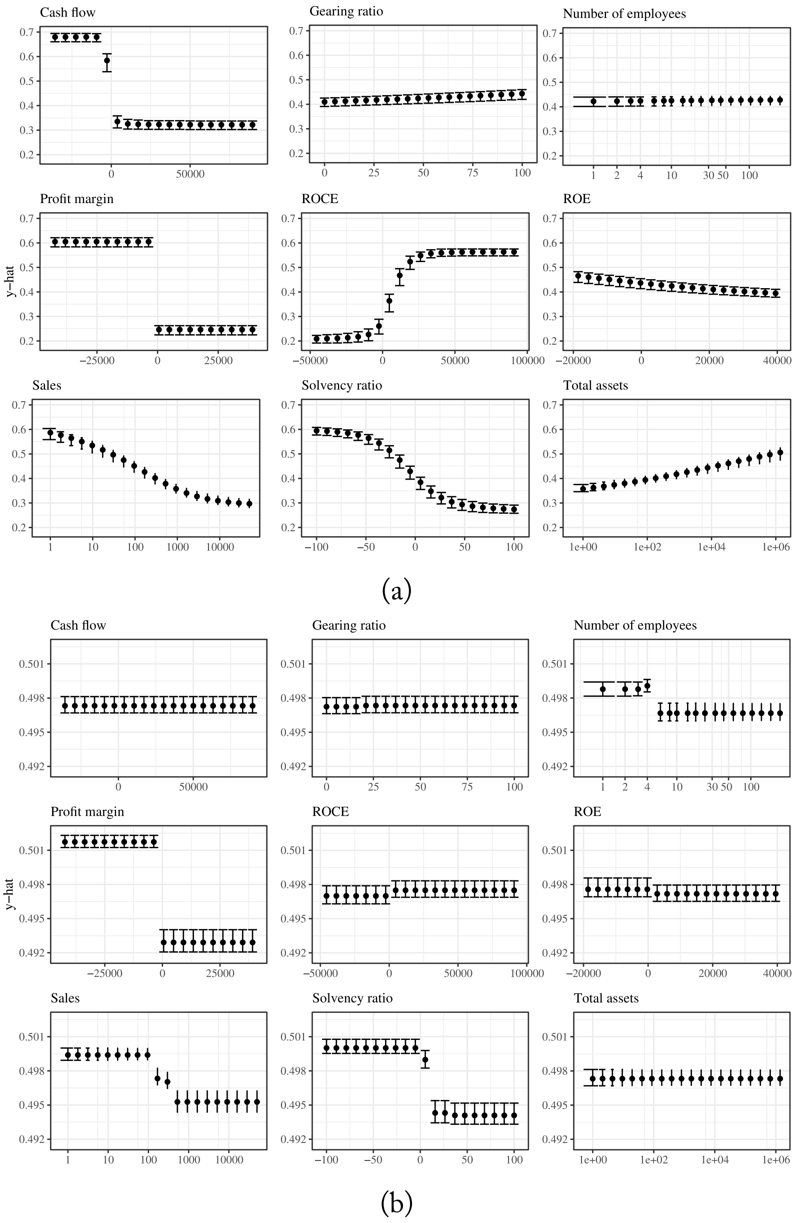


FIGURE 3.3: Global Shapley values for the Feedforward Artificial Neural Network model (a) and the XGBoost model (b).

3.6 Conclusions

Making an AI system interpretable allows external observers to understand its working and meaning, with the non-negligible consequence of making it usable in practice: when a firm (or a customer) applies for a credit line, it has the right to be informed about the possible reasons for a refusal. AI driven decisions must be explained - as much as possible- to and understood by those directly and indirectly affected, in order to allow the contesting of such decisions. This issue has become extremely relevant since both academicians and practitioners have progressively embraced ML modelling of firm default due to excellent performances (Ciampi et al., 2021) and, concurrently, Institutions have started to question the trustworthiness of - and set boundaries for - a safe use of AI in the interest of all involved (Commission et al., 2019). At the same time, using AI methods might grant larger amounts of credit and result in lower default rates (Moscatelli et al., 2020).

Here we contribute to the literature on SMEs default by showing that the good performances in classification tasks obtained through ML models can and should be accompanied by a clear interpretation of the role and type of effect played by the variables involved. We also contribute to the literature on global



(a)

(b)

Figure 3.4: Accumulated Local Effects of the FANN (a) and XGBoost (b) with related bootstrap 5%-95% confidence intervals. The ALEs for Sales, Total Assets and Number of Employees are calculated on log-transformation of the variables but depicted on anti-log values to enhance interpretability.

post-hoc interpretability showing that, differently from the ante-hoc techniques, they enable the comparison among white and black-boxes on a common ground.

Using a collection of relevant accounting indicators, widely employed in the literature, for all the Italian SMEs available in the BvD-Orbis dataset 2016, we have supplied an accurate prediction of default in 2017. Thanks to our research design, caring for imbalance among classes and cross-validation to select the most performing rules, we have achieved fair rates of correct classifications for all the models involved. However, focusing in particular on the correct rate of default classification, the XGBoost algorithm prevails over three white-box models and over the alternative ML model FANN.

Interpretability was provided by means of Shapley values and ALEs, two recent model-agnostic techniques which measure the relative importance of the predictors and shape the predictor-outcome relationship respectively. The analysis of the XGBoost ALEs reveals that such complex models capture highly non-linear patterns as the effects of sales on the probability of default, account for separate effects of correlated measures and suggest also non-trivial risky thresholds: a thing that was not completely grasped by any standard discriminant rule.

We think that the examination of ALEs for models which are already ante-hoc interpretable in the traditional scheme of statistical significance is quite revealing, both methodologically and empirically speaking. The latter models' ALEs permits to add different shades to the variables' effects with respect to the standard parameter-pvalues' paradigm, paradoxically uplifting their a-priori interpretability. Finally, the assessment of ALEs' variability is fundamental to check the output robustness and to evaluate the soundness of results.

With this paper we have shown that, under the assumption that interpretability is crucial to building and maintaining the users' trust in AI systems, their -potential- superiority in classification tasks does no longer need to be an alibi to hide the underlying mechanisms in black-boxes.

The relevancy of this approach could become definitely more important for default prediction based on alternative sources of data, such as web-scraped information Crosato et al., 2021, whose dimensionality and complexity require the power of ML models and whose interpretability is even more puzzling. This, as well as applications to a more extensive basket of traditional predictors, might represent a good ground for further research.

This study has some limitations revolving around three main aspects. The first is given by the post-hoc nature of ALEs, which is common to all the interpretable ML methods. Post-hoc methods restrict the possibility to address any biases and impose some sort of regularization on the interpretations Repetto, 2022. On the user's side, they require some basic knowledge of the methodology to interpret its outcomes. Second, the cross-sectional nature of the data prevented us from including in the analysis standard non-firm specific predictors, such as regional GDP growth, industry-level value added or business confidence indicators, which could have helped to reduce classification errors. Third, our findings, regarding Italian SMEs evaluated in a specific year, should be generalized with caution and would surely benefit from a cross-country comparison and a longitudinal follow-up.

Multicriteria interpretability driven Deep Learning 4

Marco Repetto

Annals of Operations Research. Springer. *Published.*
DOI: https://doi.org/10.1007/s10479-022-04692-6

Deep Learning methods are well-known for their abilities, but their interpretability keeps them out of high-stakes situations. This difficulty is addressed by recent model-agnostic methods that provide explanations after the training process. As a result, the current guidelines' requirement for "interpretability from the start" is not met. As a result, such methods are only useful as a sanity check after the model has been trained. In an abstract scenario, "interpretability from the start," implies imposing a set of soft constraints on the model's behavior by infusing the knowledge and eliminating any biases. By inserting knowledge into the objective function, we present a Multicriteria technique that allows us to control the feature effects on the model's output. To accommodate for more complex effects and local lack of information, we enhance the method by integrating particular knowledge functions. As a result, a Deep Learning training process that is both interpretable and compliant with modern legislation has been developed. Our technique develops performant yet robust models capable of overcoming biases resulting from data scarcity, according to a practical empirical example based on credit risk.

4.1 Introduction

Deep Learning (DL) models are widely employed currently in a variety of industries, including self-driving cars (Rao & Frtunikj, 2018), brain-computer interfaces (D. Zhang et al., 2019), and gaming (Vinyals et al., 2019). DL approaches have become more accessible thanks to recent software and technology, allowing scholars and practitioners to use them in various disciplines. On the software side, current frameworks such as Tensorflow (Abadi et al., 2015) and PyTorch (Paszke et al., 2019) have made it possible to create DL models without the need for ad-hoc compilers, as LeCun et al., 1990 did. On the hardware side, the cost of the necessary gear to train such models has decreased, allowing many people to create and deploy complex Neural Networks for very little money (Q. Zhang et al., 2018). Apart from computer science, the democratization of such strong technology benefited many other disciplines. Economics (Nosratabadi et al., 2020) and Finance (Ozbayoglu et al., 2020) are two of those that benefited the most. Governments are interested in DL applications because they are concerned about potential social consequences. However, when it comes to training, these models demand more attentiveness, especially in high-stakes judgements (Rudin, 2019). To counteract these negative consequences, governments created a number of regulatory requirements, and the law began to expand on the right to explanation concept (Dexe et al., 2020). Scholars have been constructing post-hoc interpretation methods in the aim to build interpretable but DL grounded models. These techniques, on the other hand, are at conflict with newer standards, which demand ”interpretability from the beginning” (Commission et al., 2019). Another concern is that such methods rely solely on interpretation after a model has been trained, preventing the input of prior data or the removal of biases. This research focuses on assuring the interpretability of DL models from the start by injecting knowledge and examining their potential in empirical scenarios such as credit risk prediction. Knowledge is directly infused into the learning algorithm level by our methodology. Knowledge injection, as defined by von Rueden et al., 2021, entails restricting feature relationships and can take place in four ways: (i) on the training data; (ii) on the hypothesis set; (iii) on the learning algorithm; and (iv) on the final hypothesis.

In this regard, we make three relevant contributions to the literature. First, our technique allows the Decision Maker (DM) to inject previous knowledge directly into the model learning processes. Therefore this technique may alleviate the model’s failure to generalize due to scarce data or biased one. Our approach is similar to the Physics-guided Neural Networks (PGNN) proposed by Daw et al., 2021. However, in PGNN, the effects constraints are conditional on the context as in Muralidhar et al., 2018 or applied to non-DL techniques (Kotłowski & Słowiński, 2009; Lauer & Bloch, 2008; von Kurnatowski et al., 2021). A key advantage of our approach is that it can be applied to any DL architecture and is not conditional on features’ context. As a proof of concept, we propose a credit risk empirical assessment as it is a high-stakes context. Moreover, in this field, recent frameworks as proposed by Bücker et al., 2021a do not allow for interpretability from the start. In other words, these techniques can spot model biases but cannot counter them, as their scope sole post hoc explainability. Our methodology can handle both these aspects by leaving a model compliant with the new guidelines on Sustainable AI.

Second, we account for nonlinear effects and local knowledge gaps. Defining ad-hoc knowledge functions on model parameters allows for this constraint. This extra stipulation is required for two reasons. To begin, the credit risk empirical literature argues that the performance of DL models is mostly related to their flexibility (Ciampi & Gordini, 2013). The second reason for introducing nonlinearity is that knowledge in some areas of the feature space may be missing. Third, we investigate the interaction between model-agnostic post hoc interpretability approaches, such as Accumulated Local Effects (Apley & Zhu, 2020). In our plan, these methods serve two important functions. They initially provide graphical visuals to the DM, allowing him to communicate with non-experts. Second, they serve as sanity checks for our technique and explainability-based hyperparameter optimization.

This is how the rest of the paper is organized. Section two covers knowledge injection into the model as well as multicriteria problem formulation. The data sample used to test our technique, software packages, and hardware is shown in Section three. The findings are summarized in Section four and the most important ones are examined. The fifth section draws to an end.

4.2 Methodology

4.2.1 Deep Learning

DL is an AI subfield and type of Machine Learning technique aimed at developing systems that can operate in complex environments (Goodfellow et al., 2016). Deep architectures underpin DL systems which can be defined as:

DL is a subfield of AI and a form of Machine Learning technique focused at producing systems that can operate in complex contexts (Goodfellow et al., 2016). Deep architectures are the foundations of DL systems, which are defined as:

$$\mathcal{F} = \{f(\cdot, w), w \in \mathcal{W}\} \tag{4.1}$$

where $f(\cdot, w)$ denotes a shallow architecture, such as the Perceptron presented by Rosenblatt, 1958. McCulloch and Pitts, 1943 presented a method in which binary neurons grouped in a ring could do simple logic operations before Rosenblatt. In modern Artificial Neural Networks (ANNs) designs, neither the Perceptron nor the system presented by McCulloch and Pitts, 1943 are employed. Gradient-based optimization techniques, notably Stochastic Gradient Descent, are used in modern architectures (SGD).

This research puts our method to the test on two different DL architectures: the Multilayer Perceptron (MLP) and the ResNet (RN). The MLP was chosen because it is an off-the-shelf solution in many use-cases, particularly in credit risk (Ciampi et al., 2021) and various publications on retail credit risk (Lessmann et al., 2015). MLP is made up of a densely connected network of nodes organized in a direct acyclic network. Inputs are supplied into the node's activation function after being weighted and shifted by a bias term and impact each successive layer until the final output layer.

In a binary classification task, the output of an MLP can be described as in Arifovic and Gencay, 2001 by:

$$f(x) = \phi\left(\beta_0 + \sum_{j=1}^{d} \beta_j G\left(\gamma_{j0} + \sum_{i=1}^{p} \gamma_{ji} x_i\right)\right) \tag{4.2}$$

Because it features "shortcut connections," RN differs from the canonical MLP architecture in that it mitigates the problem of degradation in the event of numerous layers He et al., 2015. Although the use of shortcut connections is not new in the literature Venables and Ripley, 1999, He et al., 2015 proposed that identity mapping be used instead of any other nonlinear transformation. The simplest building unit of the ResNet architecture is depicted in Figure 4.1. The shortcut has an impact on both layers in this architecture. And the final output gets both the x inputs and the layers' transformation.

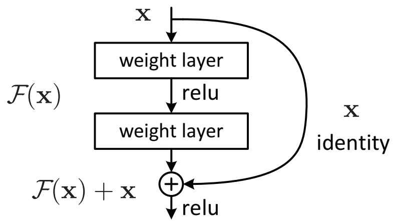

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4.1: Residual Network skip connection block.

4.2.2 Multicriteria optimization

Multiple Criteria Decision-Making (MCDM) is a branch of Operations Research and Management Science. MCDM methods allow the DM to include numerous, possibly conflicting, criteria into the analytical processes. MCDM problems are more common than single criteria ones and have been studied in several fields, including economics, engineering, finance, and management (Colapinto et al., 2015). MCDM techniques are used extensively in DL, especially in the training process and on final model evaluation (Yang et al., 2020). An MCDM problem takes the following form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_k(\boldsymbol{x})\} && (4.3) \\ \text{subject to} \quad & \boldsymbol{x} \in S && (4.4) \end{aligned}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the ith objective and the vector $\boldsymbol{x}$ contains the decision variables that belong to the feasible set S.

When dealing with MCDM problems, scalarization is a frequent strategy. A vector optimization problem is scalarized into a single objective optimization problem. To solve our problem, we start with a weighted sum scalarization:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x}) && (4.5) \\ \text{subject to} \quad & \boldsymbol{x} \in S && (4.6) \end{aligned}$$

where the weights express the relative preference of the DM toward a specific goal. Preferences incorporation can happen in two ways, either a priori or a posteriori. In our approach, we use an a posteriori method as this best suits the DM's lack of knowledge, which may be uncertain about the relative importance of each objective.

4.2.3 Knowledge injection

As posed by von Rueden et al., 2021, knowledge in this paper is validated information about relations between entities in specific contexts. This type of formulation allows for formalization, suggesting that knowledge can be represented mathematically. Let's assume we have a deep architecture such that $\hat{y} = \mathcal{F}(\boldsymbol{x}, \mathcal{W})$. We observe the true label y in a supervised setting, and we can train a model using a differentiable loss function, similar to how we train a model for regression using the mean squared error (MSE):

$$Loss_f(\boldsymbol{x}, \mathcal{W}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.7}$$

or in our case, of a binary classification with the binary cross-entropy:

$$Loss_f(\boldsymbol{x}, \mathcal{W}) = \frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{4.8}$$

The first goal of our loss function, namely data fitting, will be achieved. Instead, the knowledge injection will operate on the effects of the characteristics on the model outcome hence our knowledge-based goal will be:

$$Loss_k(\boldsymbol{x}, \mathcal{W}) = k(\boldsymbol{x}) \odot \frac{\delta \mathcal{F}(\boldsymbol{x}, \mathcal{W})}{\delta \boldsymbol{x}} \tag{4.9}$$

where $k(\boldsymbol{x})$ is a function that penalizes/favorites certain effects of the gradient with range $[-1, 1]$, and the right-hand side of the Hadamard product is the gradient of our DL model at the feature level $\boldsymbol{x}$. Because knowledge does not survive throughout the entire feature space, one option is to define $k(\boldsymbol{x})$, which translates the feature space to the knowledge we expect on that specific feature neighbor.

In its most straightforward formulation, $k(\boldsymbol{x})$ can be a scalar. Three influences on the model may be identified in this situation. If $k(\boldsymbol{x}) = \mathbf{1}$ then all partial derivatives shuold be negative. therefore the result is decreasing monotonicity. The opposite holds for the case when $k(\boldsymbol{x}) = \mathbf{-1}$. When $k(\boldsymbol{x}) = \mathbf{0}$ there is no constraint on the gradient behavior, meaning that knowledge is nonexistent and therefore not injected. Following Daw et al., 2021, we can add a new constraint to our Multicriteria function that gauges network complexity, such as a L_2 regularization on the weights. The following unconstrained minimization emerges as a result:

$$\min_{\mathcal{W}} \boldsymbol{\lambda}^\top \begin{bmatrix} Loss_f(\boldsymbol{x}, \mathcal{W}) \\ ||\mathcal{W}||_2 \\ Loss_k(\boldsymbol{x}, \mathcal{W}) \end{bmatrix} \tag{4.10}$$

The key feature of our technique is that it injects knowledge at the learning phase. As a result, our approach is subject to the same restrictions affecting any differentiable programming problem. A crucial implication is that our approach works with categorical data, provided the embedding.

4.2.4 Interpretability methods

A DL model is essentially a blackbox. The number of parameters and transformations that the inputs have before reaching the output prevents any meaningful understanding by the DM. Model interpretability methods try to address this issue. These techniques are numerous (Molnar et al., 2020) and sortable under different axes. An example is whether a technique provides a global or local explainability measure. A second example is whether a technique has access to the model's parameters or not. The latter are called model-agnostic techniques, the former model-aware. In general model-agnostic methods, as the name suggests, apply to all models. Instead, model-aware techniques are specific to each class of model. Such characteristic means that model-aware models are more efficient and converge faster to the genuine interpretability metric. At least faster than their model-agnostic counterpart. The first interpretability method was a model-agnostic one, the Partial Dependence (PD). Proposed by Friedman, 1991, PD evaluates the change in the average predicted value for a given feature. This metric is computed by varying the features over their marginal distribution (Goldstein et al., 2015). As they rely solely on the marginal distribution, PD can be misleading in the case of feature dependence. Intuitively, given two financial ratios that share the same indicator, it is nonsense to let just one vary over its entire marginal distribution. The resulting synthetic data point used for PD evaluation will be outside the data envelope. Because of this reason, Apley and Zhu, 2020 developed a new metric, the Accumulated Local Effect (ALE). The differences between PD and ALE are two. The first one is that they rely on features' conditional distribution. In practice, this is achieved by binning the feature space. The bin size is an arbitrary parameter. A narrow binning will result in shaky ALE, whereas wide bins can still have the problem of extrapolation outside the data envelope. The second one is that the effects are accumulated rather than averaged, as in the case of PD. In their application, ALE is a model-agnostic technique. However, if the prediction function is differentiable, ALE can be rewritten in model-aware form:

$$ALE_{\hat{f},S}(x) = \int_{z_{0,S}}^{x} \left[\int \frac{\delta \hat{f}(z_S, X_{\setminus S})}{\delta z_S} d\mathcal{P}(X_{\setminus S}|z_S) \right] dz_S - C \tag{4.11}$$

where $\hat{f}$ is the black-box model and $\frac{\delta \hat{f}(z_S, X_{\setminus S})}{\delta z_S}$ its gradient. S identifies the subset of variables' index. X is the matrix containing all the variables, and x is the vector containing the feature values per observation. z identifies the boundaries of the K partitions, such that $z_{0,S} = min(x_S)$, Last, C is a constant term to center the plot.

Having the gradient inside an interpretability measure is not new in the literature. Baehrens et al., 2010 proposed an interpretability technique based on the product of the model's gradient with feature values. Simonyan et al., 2014 proposed Saliency Maps based on the gradient of model output to the input features. Therefore a knowledge injection strategy as proposed in 4.9 will have an impact on these techniques. The empirical experiment provides an analysis of such impact.

4.3 Data, software, hardware

4.3.1 Data

To test the goodness of our approach, we provide an empirical application in the context of credit risk. In particular on the problem of bankruptcy prediction. We used a publicly available dataset of Polish enterprises donated to the UCI Machine Learning Repository by Zikeba et al., 2016. The reason behind the usage of a publicly available dataset is twofold. First, we wanted to have a dataset that can be used and reproduced by anyone, as to not be dependent on the particularities of a private dataset. Second, the dataset has already been used in the literature to test different machine learning algorithms, as to make our results comparable with previous approaches. The data contains information about the financial conditions of Polish companies belonging to the manufacturing sector. The dataset contains 64 financial ratios ranging from liquidity to leverage measures [*]. Moreover, the dataset distinguishes five classification cases that depend on the forecasting period. In our empirical setting, we focus on bankruptcy status after one year. In this subset of data, the total number of observations is 5910, out of which only 410 represents bankrupted firms. It is worth noting that we do not counter the class imbalance in the empirical setting, although this is something done commonly in the literature. We retained class imbalance to test the robustness of our approach even in conditions of scarcity of a particular class and used robust metrics such as the Area Under the Receiving Operating Curve (AU-ROC). Moreover, as our empirical experiment focuses on testing our approach on model interpretability, we restricted the number of features we considered to six. This is due to the fact that ALEs are inspected as plots, and having a plot for each feature increases complexity without providing any additional benefit to the reader or our approach. The choice was to focus on Attr 13, Attr 16, Attr 23, Attr 24, Attr 26, and Attr 27. The attributes were selected by using a ROC-based feature selection (Kuhn & Johnson, 2019).

We give an empirical application in the area of credit risk to test the validity of our technique. In specifically, the challenge of predicting insolvency. Zikeba et al., 2016 provided a publicly available dataset of Polish businesses to the UCI Machine Learning Repository. The report includes information on the financial health of Polish manufacturing enterprises. There are 64 financial ratios in the dataset, spanning from liquidity to leverage measures. Furthermore, the dataset

[*] For a complete description of the indicators, please consider Table 10 in the Appendix.

separates five classification scenarios based on the predicting period. In our empirical setting, we focus on bankruptcy status after one year. The total number of observations in this subset of data is 5910, with only 410 being defunct companies. It's worth emphasizing that we don't correct for class imbalance in the empirical context, despite the fact that this is a frequent practice in the literature. We kept the class imbalance to evaluate the durability of our strategy even when a single class was scarce, and we used robust measures like the Area Under the Receiving Operating Curve to do so (AUROC). Furthermore, we limited the number of attributes we analyzed to six because our empirical experiment focused on assessing our strategy on model interpretability. This is because ALEs are inspected as plots, and having a plot for each feature adds to the complexity without adding any value to the reader or our method. Attr 13, Attr 16, Attr 23, Attr 24, Attr 26, and Attr 27 were chosen as the focus points. The attributes were chosen using the Kuhn and Johnson, 2019 ROC-based feature selection method.

4.3.2 Software and hardware

R is used to manage the entire workflow (R Core Team, 2021). The preprocessing relied on the tidymodels ecosystem (Kuhn & Wickham, 2020). The DL models are developed in Julia (Bezanson et al., 2017) using the Flux framework (Innes et al., 2018; Innes, 2018). The interoperability between the two languages is possible via the JuliaConnectoR library (Lenz et al., 2021). As for the hardware, the pipeline is carried out on a local machine with 12 logical cores (Intel i7-9850H), 16 GB RAM, and a Cuda enabled graphic card (NVIDIA Quadro T2000). Both Julia and R codes are freely available for research reproducibility on GitLab, and an ad-hoc Docker container has been created on DockerHub.

4.4 Results

We apply a typical practice in the field of DL to test the performance of our strategy on the dataset of Polish enterprises. We divided the dataset into two parts: training and testing. The dataset is divided into two parts: one for training the model and the other for testing its performance. A configuration like this will suffice in the event of a model with no hyperparameters. In DL, however, this is seldom the case because these models require extensive hyperparameter calibration. The hyperparameters in our setup are the elements contained in $\boldsymbol{\lambda}$. As a result, using the training set to perform what is known as hyperparameter optimization (Goodfellow et al., 2016) is a frequent method. As a result, the training set is split into training and validation sets, and the model is fitted and validated using various parameters. In our example, we used the bootstrap technique to extract ten samples from the training set, and we trained our model with several hyperparameter combinations. We used grid search, which is also known as full factorial design (Montgomery, 2017), to implement such hyperparameter search. The model was then trained on the whole training set, and the appropriate hyperparameters were used to classify bankruptcy state on the test set.

We divided the study into three stages to ensure a solid understanding of the findings. The MLP and the RN were used to undertake hyperparameter optimization and subsequent hold-out testing, with the former being the best performing model. Second, we used ALE plots to examine the effect of the MLP with knowledge injection on interpretation. Finally, we put our method to the test by reducing the amount of data we used.

4.4.1 Performance review

Because our model validation included ten bootstrap samples, table 4.1 shows the mean AUROC as well as its standard errors. The first remarkable finding is that without regularization and knowledge injection, both the MLP and the RN perform badly. This finding is consistent with that of Zikeba et al., 2016, which indicated that ANN designs suffer from poor generalization. Instead, the gain in performance when both are present is of tremendous interest. The table is set with regularization and knowledge injection. With low amounts of knowledge injection and a moderate level of regularization, the RN appears to perform better. What matters, though, is how the MLP behaves. The model is more responsive to knowledge injection and outperforms the competition in model validation. When $\lambda_3 = 0.3$, this result is obvious. With a little level of standard error, all of the MLP models have an AUROC of 0.8 or above. When regularization starts to ramp up $\lambda_2 = 0.3$ even with inserted information, another key outcome of the MLP is performance degradation. It's worth mentioning that the dataset has a major class imbalance, and nothing has been done to address it in order to test the efficacy of our approach in this setting. Indeed, knowledge injection reduces misclassification and results in a model that is comparable to other reliable classifiers.

Table 4.1: Performances of Multilayer Perceptron and Residual Network on the training set with various hyperparameter settings. The performance is measured as the mean and standard errors of the Area Under the Receiving Operating Curve in each bootstrap sample. Bold values indicate the best performing hyperparametrization for each model.

			Multilayer Perceptron		**Residual Network**	
λ_1	λ_2	λ_3	Mean	Standard error	Mean	Standard error
1.0	0.0	0.0	0.6585	0.0178	0.5061	0.0843
0.9	0.1	0.0	0.6924	0.0111	0.5308	0.0805
0.8	0.2	0.0	0.6302	0.0757	0.6326	0.0180
0.7	0.3	0.0	0.7175	0.0059	0.5584	0.0900
0.9	0.0	0.1	0.7905	0.0087	0.6418	0.0740
0.8	0.1	0.1	0.8286	0.0149	0.5744	0.0769
0.7	0.2	0.1	0.7586	0.0336	0.5059	0.0746
0.6	0.3	0.1	0.6163	0.1219	**0.6604**	0.0879
0.8	0.0	0.2	0.8263	0.0129	0.5124	0.0664
0.7	0.1	0.2	0.8242	0.0170	0.6102	0.0186
0.6	0.2	0.2	0.8206	0.0123	0.5037	0.0443
0.5	0.3	0.2	0.7617	0.0601	0.6249	0.0593
0.7	0.0	0.3	0.8202	0.0119	0.6074	0.0172
0.6	0.1	0.3	0.8289	0.0139	0.5410	0.0417
0.5	0.2	0.3	**0.8306**	0.0135	0.5318	0.0150
0.4	0.3	0.3	0.8178	0.0198	0.5628	0.0378

Table 4.2: Performances of Multilayer Perceptron and Residual Network on the test set with validated and baseline hyperparameter settings. The performance is measured as the Area Under the Receiving Operating Curve in the test sample.

λ_1	λ_2	λ_3	**Multilayer Perceptron**	**Residual Network**
1.0	0.0	0.0	0.577	**0.582**
0.5	0.2	0.3	**0.821**	-
0.4	0.3	0.1	-	0.518

The results in table 4.1 are encouraging. The performances from the test set, on the other hand, must be incorporated for measuring model generalization error. These results are presented in table 4.2 by just considering the ideally parametrized models and their baseline, that is, the model with $\lambda_1 = 1$. This table clearly shows that the MLP generalizes far better than its competitor, with a minor drop in performance that is in accordance with predictions. This finding suggests that knowledge injection combined with modest regularization can improve a DL classifier's generalization performance and make it more resistant to class imbalances.

In the sections that follow, we'll look at how the MLP performs in terms of interpretability and robustness to data scarcity with and without knowledge injection. We'll just concentrate on the MLP because it's the most performant architecture.

4.4.2 Interpretability review

Current interpretability methods, as indicated in the preceding sections, are vital tools for model debugging and inspecting any model bias. As a result, figure 4.2a shows MLP ALEs with and without knowledge injection. The model's ALEs without knowledge injection exhibit a number of misbehaviors that could be related to class imbalance or hidden biases in the training sample. In depth:

- Attr 13: which is also known as the EBITDA-To-Sales ratio, is a profitability indicator. Therefore we should expect to decrease the probability of bankruptcy, especially in cases where the ratio is positive. The opposite occurs instead. An increase of the ratio above zero increases the probability of bankruptcy. This effect is at odds with the literature on the subject as, for example, in Platt and Platt, 2002;

- Attr 16: is the inverse and a proxy of the Debt-To-EBITDA ratio which is leverage ratio. For the inverse of a leverage ratio, we would assume a negative impact on bankruptcy as in Beaver, 1968;

- Attr 23: is the Net profit ratio and is a productivity ratio (Lee & Choi, 2013) which tends to have a negative impact on bankruptcy.

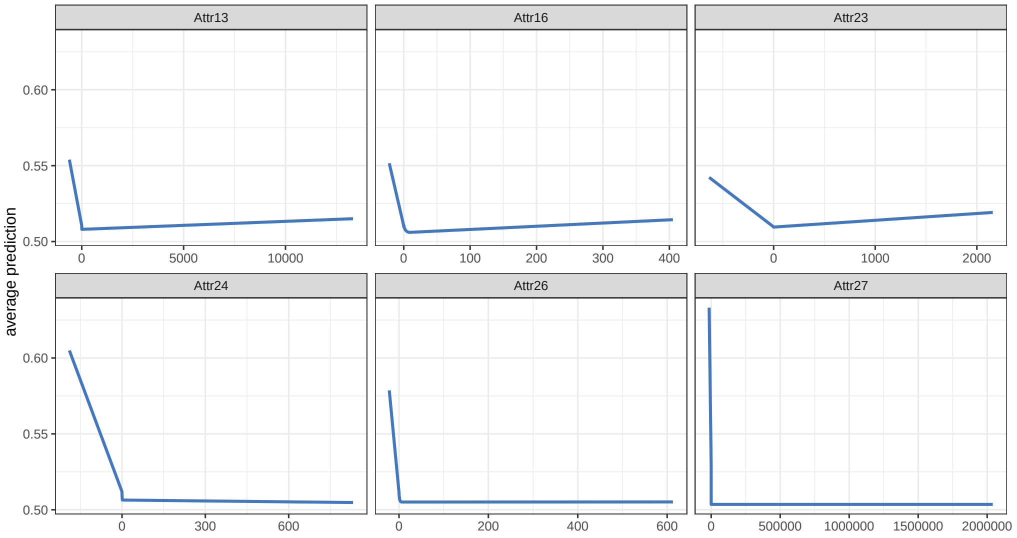
To account for these typical biased effects, we assumed the following logistic form for the knowledge function of all features:

$$k(x) = \frac{1}{1 + e^{-100x}} \tag{4.12}$$

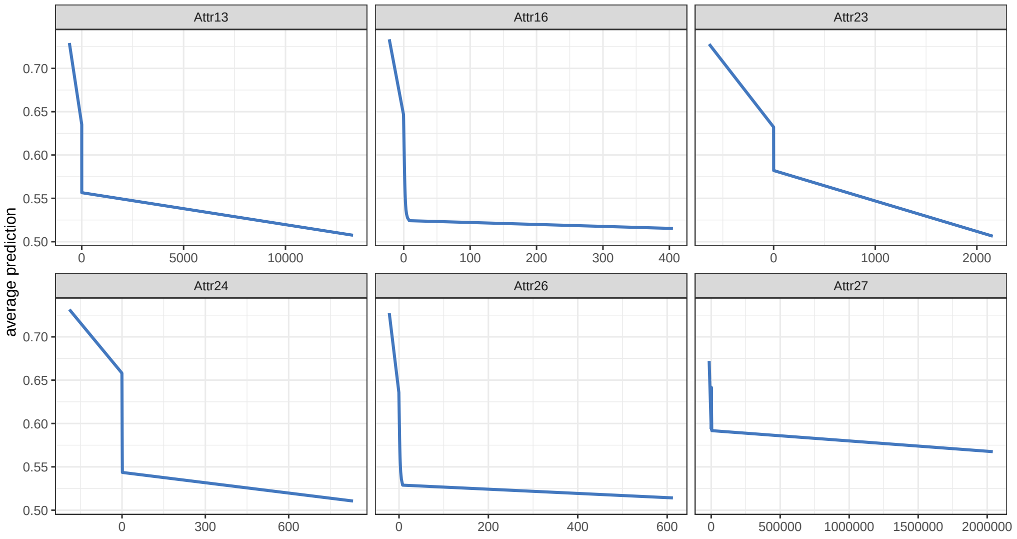
Such a knowledge function penalizes only positive effects above zero and retains the correctly captured effects below it. With this setting, in the case of moderate knowledge injection, the effects align with the literature findings.

4.4.3 Robustness checks

A key topic is how model performance degrades as the amount of training data decreases. Knowledge injection has been used in the past to solve difficulties like those described in von Kurnatowski et al., 2021. We steadily reduced the training set and measured the matching performance on the test set to see how our approach dealt with scarce data. These results are shown in Table 4.3, which shows how the test set performs as the proportion of training data drops. Our strategy, which is in line with the literature on knowledge injection, eliminates performance degradation even when only half of the dataset is used for training.



(a) Accumulated Local Effects plot of the Multilayer Perceptron architecture, without regularization and knowledge injection (i.e. $\lambda_1 = 1, \lambda_2 = 0.0, \lambda_3 = 0.0$).



(b) Accumulated Local Effects plot of the Multilayer Perceptron architecture, with regularisation and knowledge injection optimally selected from the hyperparameter optimization procedure (i.e. $\lambda_1 = 0.5, \lambda_2 = 0.2, \lambda_3 = 0.3$).

Figure 4.2: Accumulated Local Effects plot of the Multilayer Perceptron architecture, with and without regularization and knowledge injection.

Table 4.3: Performances of Multilayer Perceptron on the test set under different proportions of train/test split. The performance is measured as the Area Under the Receiving Operating Curve in the test sample.

Train/Test	**With knowledge** ($\lambda_1 = 0.5, \lambda_2 = 0.2, \lambda_3 = 0.3$)	**Without knowledge** ($\lambda_1 = 0.1, \lambda_2 = 0.0, \lambda_3 = 0.0$)
0.85	0.829	0.615
0.80	0.828	0.543
0.75	0.821	0.577
0.7	0.822	0.605
0.65	0.790	0.613
0.6	0.817	0.646
0.55	0.803	0.505
0.5	0.823	0.641

4.5 Conclusion

We developed a novel method for knowledge injection at the level of feature effects in a DL model in this study. Model interpretability is a very important problem, and new legislation requires it from the start. Recent post-hoc interpretability solutions fall short of this requirement. Our solution solves the problem by allowing model interpretation to be controlled from the beginning. The method entails addressing a multicriteria minimization problem in which greedy data fitting and regularization conflict with knowledge adherence. By constructing ad-hoc knowledge functions on the model's parameters, we were able to account for partial knowledge and nonlinearity. To demonstrate our approach, we presented a use case of bankruptcy prediction using a dataset from a Polish firms. The findings imply that knowledge injection enhances model performance and keeps model interpretation consistent with literature findings, avoiding idiosyncratic effects caused by class imbalance or potential dataset biases. The DM can test the effects of our approach using post-hoc interpretability approaches, which are critical for fine-tuning the model before it goes into production. Another important subject we addressed was model performance degradation in the event of data scarcity. Our findings show that knowledge injection provides the modeler with more flexibility in terms of obtaining the data required for proper model training.

In terms of research, this new paradigm opens up a slew of new possibilities. The examination of increasingly complicated knowledge functions is one avenue for future research. A second line of inquiry would be to ensure knowledge consistency across many contexts, such as time series. Last, in future iterations, we may have the opportunity to rely on newer publicly available datasets to further uncover the Covid-19 implications on firms' default. These are only a few examples of potential new research initiatives that will pave the way for knowledge-informed DL.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems. (Cit. on p. 50).

Ahelegbey, D. F., Giudici, P., & Hadji-Misheva, B. (2019). Latent factor models for credit scoring in P2P systems. *Physica A: Statistical Mechanics and its Applications*, *522*, 112–121 (cit. on p. 3).

Alonso, A., & Carbó, J. M. (2020). On the risk-adjusted performance of Machine Learning models in credit default prediction. *SUERF Policy Note*, *210*, 1–10 (cit. on p. 30).

Altman, E. I., Sabato, G., & Wilson, N. (2010). The value of non-financial information in small and medium-sized enterprise risk management. *J. Credit Risk*, *6*(2), 1–33 (cit. on pp. 32, 33, 36, 39, 42).

Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). *Journal of Banking & Finance*, *18*(3), 505–529 (cit. on p. 24).

Altman, E. I., & Sabato, G. (2007). Modelling Credit Risk for SMEs: Evidence from the U.S. Market. *Abacus*, *43*(3), 332–357 (cit. on p. 39).

Anderson, J., & Rainie, L. (2018). Artificial intelligence and the future of humans. (Cit. on p. 1).

Andreeva, G., Calabrese, R., & Osmetti, S. A. (2016). A comparative analysis of the UK and Italian small businesses using Generalised Extreme Value models. *European Journal of Operational Research*, *249*(2), 506–516 (cit. on pp. 3, 32, 33, 37, 39, 42).

Annanth, V. K., Abinash, M., & Rao, L. B. (2021). Intelligent manufacturing in the context of industry 4.0: A case study of siemens industry. *Journal of Physics: Conference Series*, *1969*(1), 012019 (cit. on p. 2).

Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical*

Society: Series B (Statistical Methodology), *82*(4), 1059–1086 (cit. on pp. 20, 21, 29, 32, 35–37, 51, 55).

Arifovic, J., & Gencay, R. (2001). Using genetic algorithms to select architecture of a feedforward artiÿcial neural network. *Physica A*, 21 (cit. on pp. 35, 52).

Arora, R., Kakkar, P., Dey, B., & Chakraborty, A. (2022). Physics-informed neural networks for modeling rate-and temperature-dependent plasticity. *arXiv preprint arXiv:2201.08363* (cit. on p. 2).

Arrieta, A. B., Diaz-Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115 (cit. on p. 30).

Ashby, W. R. (1956). An introduction to cybernetics (cit. on p. 7).

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K.-R. (2010). How to explain individual classification decisions. *The Journal of Machine Learning Research*, *11*, 1803–1831 (cit. on p. 55).

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 627–635 (cit. on p. 35).

Baesens, B., Höppner, S., Ortner, I., & Verdonck, T. (2021). robROSE: A robust approach for dealing with imbalanced data in fraud detection. *Statistical Methods & Applications* (cit. on p. 37).

Bank of England. (2019). *Machine learning in UK financial services* (tech. rep.). (Cit. on p. 30).

Beaver, W. H. (1968). Alternative accounting measures as predictors of failure. *The accounting review*, *43*(1), 113–122 (cit. on p. 60).

Bellandi, M., Lombardi, S., & Santini, E. (2020). Traditional manufacturing areas and the emergence of product-service systems: The case of Italy. *Journal of Industrial and Business Economics*, *47*(2), 311–331 (cit. on p. 32).

Bellotti, A., Brigo, D., Gambetti, P., & Vrins, F. (2021). Forecasting recovery rates on non-performing loans with machine learning. *International Journal of Forecasting*, *37*(1), 428–444 (cit. on p. 32).

Berg, D. (2007). Bankruptcy prediction by generalized additive models. *Applied Stochastic Models in Business and Industry*, *23*(2), 129–143 (cit. on p. 29).

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, *59*(1), 65–98 (cit. on p. 57).

Biecek, P., & Burzykowski, T. (2021). *Explanatory Model Analysis*. Chapman; Hall/CRC, New York. (Cit. on pp. 11, 12, 18).

Bredensteiner, E. J., & Bennett, K. P. (1996). Feature Minimization within Decision Trees. *Computational Optimization and Applications*, *10*, 10–111 (cit. on p. 8).

Breeden, J. L. (2020). Survey of Machine Learning in Credit Risk (May 30, 2020). (Cit. on p. 38).

Bücker, M., Szepannek, G., Gosiewska, A., & Biecek, P. (2021a). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, 1–21 (cit. on pp. 11, 50).

Bücker, M., Szepannek, G., Gosiewska, A., & Biecek, P. (2021b). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, 1–21 (cit. on p. 31).

Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2021). Explainable machine learning in credit risk management. *Computational Economics*, *57*(1), 203–216 (cit. on pp. 29, 34, 35).

Calabrese, R., Marra, G., & Angela Osmetti, S. (2016). Bankruptcy prediction of small and medium enterprises using a flexible binary generalized extreme value model. *Journal of the Operational Research Society*, *67*(4), 604–615 (cit. on pp. 31, 33, 34, 44).

Chen, C., Lin, K., Rudin, C., Shaposhnik, Y., Wang, S., & Wang, T. (2021). A holistic approach to interpretability in financial lending: Models, visualizations, and summary-explanations. *Decision Support Systems*, 113647 (cit. on p. 31).

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794 (cit. on p. 28).

Ciampi, F. (2015). Corporate governance characteristics and default prediction modeling for small enterprises. An empirical analysis of Italian firms. *Journal of Business Research*, *68*(5), 1012–1025 (cit. on pp. 31, 36, 39).

Ciampi, F., Giannozzi, A., Marzi, G., & Altman, E. I. (2021). Rethinking SME default prediction: A systematic literature review and future perspectives. *Scientometrics*, 1–48 (cit. on pp. 2, 28, 29, 35, 45, 52).

Ciampi, F., & Gordini, N. (2013). Small enterprise default prediction modeling through artificial neural networks: An empirical analysis of Italian small enterprises. *Journal of Small Business Management*, *51*(1), 23–45 (cit. on pp. 30, 32, 33, 38, 39, 51).

Colapinto, C., Jayaraman, R., & Marsiglio, S. (2015). Multi-criteria decision analysis with goal programming in engineering, management and social sciences: A state-of-the art review. *Annals of Operations Research*, *251*(1–2), 7–40 (cit. on p. 53).

Commission, E., Directorate-General for Communications Networks, C., & Technology. (2019). *Ethics guidelines for trustworthy ai*. (Cit. on pp. 28, 30, 45, 50).

Cornille, D., Rycx, F., & Tojerow, I. (2019). Heterogeneous effects of credit constraints on SMEs' employment: Evidence from the European sovereign debt crisis. *Journal of Financial Stability*, *41*, 1–13 (cit. on p. 28).

Coussement, K., & Benoit, D. F. (2021). Interpretable data science for decision making. *Decision Support Systems*, *150*, 113664 (cit. on p. 28).

Covert, I., Lundberg, S., & Lee, S.-I. (2020). Understanding Global Feature Contributions With Additive Importance Measures. (Cit. on p. 35).

Crosato, L., Domenech, J., & Liberati, C. (2021). Predicting SME's default: Are their websites informative? *Economics Letters*, 109888 (cit. on p. 47).

Davison, A. C., & Kuonen, D. (2002). An Introduction to the Bootstrap with Applications in R. *Statistical Computing and Graphics Newsletter*, 6 (cit. on p. 37).

Daw, A., Karpatne, A., Watkins, W., Read, J., & Kumar, V. (2021). Physics-guided Neural Networks (PGNN): An Application in Lake Temperature Modeling. (Cit. on pp. 50, 54).

De Leonardis, D., & Rocci, R. (2008). Assessing the default risk by means of a discrete-time survival analysis approach. *Applied Stochastic Models in Business and Industry*, *24*(4), 291–306 (cit. on p. 30).

De Leonardis, D., & Rocci, R. (2014). Default risk analysis via a discrete-time cure rate model. *Applied Stochastic Models in Business and Industry*, *30*(5), 529–543 (cit. on p. 30).

Dexe, J., Ledendal, J., & Franke, U. (2020). An empirical investigation of the right to explanation under gdpr in insurance. *Lecture Notes in Computer Science*, 125–139 (cit. on p. 50).

Donepudi, P. K. (2017). Machine learning and artificial intelligence in banking. *Engineering International*, *5*(2), 83–86 (cit. on p. 2).

Doran, D., Schulz, S., & Besold, T. R. (2017). What does explainable AI really mean? A new conceptualization of perspectives. (Cit. on p. 30).

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. (Cit. on p. 30).

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77 (cit. on p. 31).

du Jardin, P., & Séverin, E. (2011). Predicting corporate bankruptcy using a self-organizing map: An empirical study to improve the forecasting horizon of a financial failure model. *Decision Support Systems*, *51*(3), 701–711 (cit. on p. 3).

El Kalak, I., & Hudson, R. (2016). The effect of size on the failure probabilities of SMEs: An empirical study on the US market using discrete hazard model. *International Review of Financial Analysis*, *43*, 135–145 (cit. on pp. 31, 42).

EU. (2003). *Commission Recommendation of 6 May 2003 concerning the definition of micro, small and medium-sized enterprises (Text with EEA relevance) (notified under document number C(2003) 1422)* (tech. rep. 32003H0361). (Cit. on p. 32).

European Banking Authority. (2020). *EBA Report on Big Data and Advanced Analytics* (tech. rep.). (Cit. on p. 30).

European Commission. (2019). *Annual Report on European SMEs 2018/2019*. (Cit. on p. 28).

Eurostat. (2018). Relative importance of Manufacturing (NACE Section C), EU, 2018. (Cit. on p. 32).

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, *19*(1), 1–67 (cit. on pp. 36, 55).

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232 (cit. on pp. 8, 31, 35).

Friedman, J. H., & Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. *Statistics in Medicine*, *22*(9), 1365–1381 (cit. on p. 8).

Giudici, P., Hadji-Misheva, B., & Spelta, A. (2020). Network based credit risk models. *Quality Engineering*, *32*(2), 199–211 (cit. on pp. 33, 44).

Glynn, C. (2022). Learning low-dimensional structure in house price indices. *Applied Stochastic Models in Business and Industry*, *38*(1), 151–168 (cit. on p. 31).

Goldstein, A., Kapelner, A., Bleich, J., & Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, *24*(1), 44–65 (cit. on pp. 8, 12, 36, 55).

Gong, J., & Kim, H. (2017). RHSBoost: Improving classification performance in imbalance data. *Computational Statistics & Data Analysis*, *111*, 1–13 (cit. on p. 38).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (Cit. on pp. 51, 57).

Gordini, N. (2014). A genetic algorithm approach for SMEs bankruptcy prediction: Empirical evidence from Italy. *Expert Systems with Applications*, *41*(14), 6433–6445 (cit. on pp. 30, 37).

Gosiewska, A., Kozak, A., & Biecek, P. (2021). Simpler is better: Lifting interpretability-performance trade-off via automated feature engineering. *Decision Support Systems*, *150*, 113556 (cit. on pp. 31, 32).

Greenwell, B. M., Boehmke, B. C., & McCarthy, A. J. (2018). A Simple and Effective Model-Based Variable Importance Measure. (Cit. on p. 20).

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1–42 (cit. on p. 30).

Gupta, J., Gregoriou, A., & Ebrahimi, T. (2018). Empirical comparison of hazard models in predicting SMEs failure. *Quantitative Finance*, *18*(3), 437–466 (cit. on p. 31).

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*(1), 81–102 (cit. on p. 12).

Haykin, S. S. (1999). *Neural networks: A comprehensive foundation* (2nd ed). Prentice Hall. (Cit. on p. 28).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. (Cit. on p. 52).

Hechtlinger, Y. (2016). Interpretation of Prediction Models Using the Input Gradient. (Cit. on p. 36).

High-Level Expert Group on Artificial Intelligence, European Commission. (2020). *The Assessment List for Trustworthy Artificial Intelligence* (tech. rep.). (Cit. on pp. 28, 30).

Holmes, P., Hunt, A., & Stone, I. (2010). An analysis of new firm survival using a hazard function. *Applied Economics*, *42*(2), 185–195 (cit. on p. 31).

Innes, M., Saba, E., Fischer, K., Gandhi, D., Rudilosso, M. C., Joy, N. M., Karmali, T., Pal, A., & Shah, V. (2018). Fashionable modelling with flux. *CoRR*, *abs/1811.01457* (cit. on p. 57).

Innes, M. (2018). Flux: Elegant machine learning with julia. *Journal of Open Source Software* (cit. on p. 57).

Insights, C. (2022). State of ai 2021 report. (Cit. on p. 1).

Institute of International Finance. (2019). *Machine learning in credit risk* (tech. rep.). Institute of International Finance. (Cit. on p. 30).

Institute of International Finance. (2020). *Machine learning governance summary report* (Summary Report). Institute of International Finance. (Cit. on p. 30).

Jabeur, S. B., Gharib, C., Mefteh-Wali, S., & Arfi, W. B. (2021). CatBoost model and artificial intelligence techniques for corporate failure prediction. *Technological Forecasting and Social Change*, *166*, 120658 (cit. on pp. 31, 39).

James, G., Witten, D., Hastie, T., & Tibshirani, R. (Eds.). (2013). *An introduction to statistical learning: With applications in R*. Springer. (Cit. on p. 37).

Jones, S., Johnstone, D., & Wilson, R. (2015). An empirical evaluation of the performance of binary classifiers in the prediction of credit ratings changes. *Journal of Banking & Finance*, *56*, 72–85 (cit. on p. 33).

Jones, S., & Wang, T. (2019). Predicting private company failure: A multi-class analysis. *Journal of International Financial Markets, Institutions and Money*, *61*, 161–188 (cit. on p. 31).

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260 (cit. on p. 5).

Klaise, J., Looveren, A. V., Vacanti, G., & Coca, A. (2021). Alibi explain: Algorithms for explaining machine learning models. *Journal of Machine Learning Research*, *22*(181), 1–7 (cit. on p. 20).

Knuth, D. E., & Moore, R. W. (1975). An analysis of alpha-beta pruning. *Artificial Intelligence*, *6*(4), 293–326 (cit. on p. 8).

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*. The MIT Press. (Cit. on p. 15).

Kotłowski, W., & Słowiński, R. (2009). Rule learning with monotonicity constraints. *Proceedings of the 26th Annual International Conference on Machine Learning*, 537–544 (cit. on p. 50).

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press. (Cit. on pp. 56, 57).

Kuhn, M., & Wickham, H. (2020). *Tidymodels: A collection of packages for modeling and machine learning using tidyverse principles.* (Cit. on p. 57).

Lauer, F., & Bloch, G. (2008). Incorporating Prior Knowledge in Support Vector Machines for Classification: A Review. *Neurocomputing*, *71*(7-9), 1578–1594 (cit. on p. 50).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444 (cit. on p. 1).

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1990). Handwritten Digit Recognition with a Back-Propagation Network. *Advances in Neural Information Processing Systems*, *2* (cit. on p. 50).

Lee, S., & Choi, W. S. (2013). A multi-industry bankruptcy prediction model using back-propagation neural network and multivariate discriminant analysis. *Expert Systems with Applications*, *40*(8), 2941–2946 (cit. on p. 61).

Lenz, S., Hackenberg, M., & Binder, H. (2021). The JuliaConnectoR: A functionally oriented interface for integrating Julia in R. (Cit. on p. 57).

Leo, M., Sharma, S., & Maddulety, K. (2019). Machine learning in banking risk management: A literature review. *Risks*, *7*(1), 29 (cit. on p. 2).

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An upyear of research. *European Journal of Operational Research*, *247*(1), 124–136 (cit. on pp. 5, 37, 38, 52).

Liberati, C., Camillo, F., & Saporta, G. (2017). Advances in credit scoring: Combining performance and interpretation in kernel discriminant analysis. *Advances in Data Analysis and Classification*, *11*(1), 121–138 (cit. on p. 31).

Lin, S. M., Ansell, J., & Andreeva, G. (2012). Predicting default of a small business using different definitions of financial distress. *Journal of the Operational Research Society*, *63*(4), 539–548 (cit. on pp. 31, 33).

Lipton, Z. C. (2018). The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, *16*(3), 31–57 (cit. on p. 30).

Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 4765–4774 (cit. on pp. 14, 31, 36).

Mai, F., Tian, S., Lee, C., & Ma, L. (2019). Deep learning models for bankruptcy prediction using textual disclosures. *European journal of operational research*, *274*(2), 743–758 (cit. on p. 30).

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133 (cit. on pp. 51, 52).

Michala, D., Grammatikos, T., & Filipe, S. F. (2013). *Forecasting distress in European SME portfolios* (EIF Working Paper No. 2013/17). European Investment Fund (EIF). Luxembourg. (Cit. on pp. 33, 39, 42).

Minsky, M., & Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry*. The MIT Press. (Cit. on p. 8).

Mitchell, M. (2021). Why ai is harder than we think. *arXiv preprint arXiv:2104.12871* (cit. on p. 1).

Modina, M., & Pietrovito, F. (2014). A default prediction model for Italian SMEs: The relevance of the capital structure. *Applied Financial Economics*, *24*(23), 1537–1554 (cit. on pp. 31, 39).

Molnar, C., Bischl, B., & Casalicchio, G. (2018). Iml: An r package for interpretable machine learning. *JOSS*, *3*(26), 786 (cit. on p. 12).

Molnar, C., Casalicchio, G., & Bischl, B. (2020). Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges. In I. Koprinska, M. Kamp, A. Appice, C. Loglisci, L. Antonie, A. Zimmermann, R. Guidotti, Ö. Özgöbek, R. P. Ribeiro, R. Gavaldà, J. Gama, L. Adilova, Y. Krishnamurthy, P. M. Ferreira, D. Malerba, I. Medeiros, M. Ceci, G. Manco, E. Masciari, … J. A. Gulla (Eds.), *ECML PKDD 2020 Workshops* (pp. 417–431). Springer International Publishing. (Cit. on pp. 9, 54).

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15 (cit. on p. 30).

Montgomery, D. C. (2017). *Design and analysis of experiments*. John wiley & sons. (Cit. on p. 58).

Moscatelli, M., Parlapiano, F., Narizzano, S., & Viggiano, G. (2020). Corporate default forecasting with machine learning. *Expert Systems with Applications*, *161*, 113567 (cit. on p. 45).

Muralidhar, N., Islam, M. R., Marwah, M., Karpatne, A., & Ramakrishnan, N. (2018). Incorporating Prior Domain Knowledge into Deep Neural Networks. *2018 IEEE International Conference on Big Data (Big Data)*, 36–45 (cit. on p. 50).

Nauck, D., & Kruse, R. (1999). Obtaining interpretable fuzzy classification rules from medical data. *Artificial intelligence in medicine*, *16*(2), 149–169 (cit. on p. 8).

Nosratabadi, S., Mosavi, A., Duan, P., Ghamisi, P., Filip, F., Band, S. S., Reuter, U., Gama, J., & Gandomi, A. H. (2020). Data Science in Economics: Comprehensive Review of Advanced Machine Learning and Deep Learning Methods. *Mathematics*, *8*(10), 1799 (cit. on p. 50).

Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications : A survey. *Applied Soft Computing*, *93*, 106384 (cit. on p. 50).

Ozgur, O., Karagol, E. T., & Ozbugday, F. C. (2021). Machine learning approach to drivers of bank lending: Evidence from an emerging economy. *Financial Innovation*, *7*(1), 1–29 (cit. on p. 36).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. (Cit. on p. 50).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830 (cit. on p. 21).

Petri, G., Musslick, S., Dey, B., Özcimder, K., Turner, D., Ahmed, N. K., Willke, T. L., & Cohen, J. D. (2021). Topological limits to the parallel processing capability of network architectures. *Nature Physics*, *17*(5), 646–651 (cit. on p. 2).

Petropoulos, A., Siakoulis, V., Stavroulakis, E., & Klamargias, A. (2019). A robust machine learning approach for credit risk analysis of large loan-level datasets using deep learning and extreme gradient boosting. In *IFC Bulletins chapters*. Bank for International Settlements. (Cit. on pp. 34, 42).

Platt, H. D., & Platt, M. B. (2002). Predicting corporate financial distress: Reflections on choice-based sample bias. *Journal of economics and finance*, *26*(2), 184–199 (cit. on p. 60).

Psillaki, M., Tsolas, I. E., & Margaritis, D. (2010). Evaluation of credit risk based on firm performance. *European Journal of Operational Research*, *201*(3), 873–881 (cit. on p. 39).

Python Core Team. (2019). *Python: A dynamic, open source programming language*. Python Software Foundation. (Cit. on p. 7).

R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. (Cit. on pp. 7, 57).

Rao, Q., & Frtunikj, J. (2018). Deep learning for self-driving cars: Chances and challenges. *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, 35–38 (cit. on pp. 5, 49).

Repetto, M. (2022). Multicriteria interpretability driven deep learning. *Annals of Operations Research* (cit. on pp. 17, 47).

Repetto, M., & La Torre, D. (2022). Making it simple? training deep learning models toward simplicity. *2022 International Conference on Decision Aid Sciences and Applications (DASA)* (cit. on p. 24).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016a). " Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD Iinternational Conference on Knowledge Discovery and Data Mining*, 1135–1144 (cit. on pp. 8, 31, 36).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016b). Model-agnostic interpretability of machine learning. (Cit. on pp. 15, 31).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1) (cit. on pp. 15, 36).

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408 (cit. on p. 51).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215 (cit. on pp. 6, 50).

Saito, K., & Nakano, R. (1988). Medical diagnostic expert system based on pdp model. *ICNN*, 255–262 (cit. on p. 8).

Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., & Santos, J. (2018). Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches. *IEEE Computational Intelligence Magazine*, *13*(4), 59–76 (cit. on p. 38).

Setiono, R. (1996). Extracting rules from pruned networks for breast cancer diagnosis. *Artificial Intelligence in Medicine*, *8*(1), 37–51 (cit. on p. 8).

Setiono, R., & Liu, H. (1995). Understanding neural networks via rule extraction. *IJCAI*, *1*, 480–485 (cit. on p. 8).

Shapley, L. S. (1953). 17. A Value for n-Person Games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the Theory of Games (AM-28), Volume II* (pp. 307–318). Princeton University Press. (Cit. on pp. 12, 35).

Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C., & Cohen, S. N. (1975). Computer-based consultations in clinical therapeutics: Explanation and rule acquisition capabilities of the mycin system. *Computers and Biomedical Research*, *8*(4), 303–320 (cit. on p. 8).

Sigrist, F., & Hirnschall, C. (2019). Grabit: Gradient tree-boosted Tobit models for default prediction. *Journal of Banking & Finance*, *102*, 177–192 (cit. on p. 31).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. (Cit. on p. 55).

Stevenson, M., Mues, C., & Bravo, C. (2021). The value of text for small business default prediction: A Deep Learning approach. *European Journal of Operational Research*, *295*(2), 758–771 (cit. on p. 31).

Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, *11*, 1–18 (cit. on p. 8).

Štrumbelj, E., & Kononenko, I. (2013). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665 (cit. on pp. 12, 14).

Succurro, M., Mannarino, L., et al. (2014). The impact of financial structure on firms' probability of bankruptcy: A comparison across western europe convergence regions. *Regional and Sectoral Economic Studies*, *14*(1), 81–94 (cit. on p. 33).

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic Attribution for Deep Networks. (Cit. on p. 35).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. (Cit. on p. 6).

Tamburri, D. A. (2020). Sustainable mlops: Trends and challenges. *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (cit. on p. 9).

Tomsett, R., Harborne, D., Chakraborty, S., Gurram, P., & Preece, A. (2020). Sanity checks for saliency metrics. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 6021–6029 (cit. on p. 24).

Veganzones, D., & Séverin, E. (2018). An investigation of bankruptcy prediction in imbalanced datasets. *Decision Support Systems*, *112*, 111–124 (cit. on pp. 37, 38).

Venables, W. N., & Ripley, B. D. (1999). *Modern Applied Statistics with S-PLUS* (3rd ed.). Springer-Verlag. (Cit. on p. 52).

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., … Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354 (cit. on p. 50).

von Kurnatowski, M., Schmid, J., Link, P., Zache, R., Morand, L., Kraft, T., Schmidt, I., & Stoll, A. (2021). Compensating data shortages in manufacturing with monotonicity knowledge. (Cit. on pp. 50, 61).

von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., & Schuecker, J. (2021). Informed Machine Learning – A Taxonomy and Survey of Integrating Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, 1–1 (cit. on pp. 50, 53).

Wei, G.-W. (2019). Protein structure prediction beyond alphafold. *Nature Machine Intelligence*, *1*(8), 336–337 (cit. on p. 1).

Wellers, D., Elliott, T., & Noga, M. (2014). Ways machine learning is improving companies' work processes. *Harvard Business Review*, *1*(1), 2–6 (cit. on p. 1).

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, *27*(11-12), 1131–1152 (cit. on p. 35).

West, D., Dellana, S., & Qian, J. (2005). Neural network ensemble strategies for financial decision applications. *Computers & Operations Research*, *32*(10), 2543–2559 (cit. on p. 35).

Westerlund, M. (2019). The emergence of deepfake technology: A review. *Technology Innovation Management Review*, *9*(11), 39–52 (cit. on p. 6).

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686 (cit. on p. 10).

Xu, Q.-S., & Liang, Y.-Z. (2001). Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*(1), 1–11 (cit. on p. 37).

Yang, M., Nazir, S., Xu, Q., & Ali, S. (2020). Deep learning algorithms and multicriteria decision-making used in big data: A systematic literature review. *Complexity*, *2020*, 1–18 (cit. on p. 53).

Yıldırım, M., Okay, F. Y., & Özdemir, S. (2021). Big data analytics for default prediction using graph theory. *Expert Systems with Applications*, *176*, 114840 (cit. on p. 31).

Yu, Q., Miche, Y., Séverin, E., & Lendasse, A. (2014). Bankruptcy prediction using extreme learning machine and financial expertise. *Neurocomputing*, *128*, 296–302 (cit. on pp. 2, 3).

Zhang, D., Cao, D., & Chen, H. (2019). Deep learning decoding of mental state in non-invasive brain computer interface. *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, 1–5 (cit. on p. 50).

Zhang, L., Hu, H., & Zhang, D. (2015). A credit risk assessment model based on SVM for small and medium enterprises in supply chain finance. *Financial Innovation*, *1*(1), 1–21 (cit. on p. 30).

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157 (cit. on p. 50).

Zhong, Y. D., Dey, B., & Chakraborty, A. (2021). Extending lagrangian and hamiltonian neural networks with differentiable contact models. *Advances in Neural Information Processing Systems*, *34*, 21910–21922 (cit. on p. 2).

Zikeba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications* (cit. on pp. 56, 58).

Appendix

Interpretability in Machine Learning

The current appendix contains additional materials regarding the following datasets:

- Boston;
- Titanic;
- Capital-Bikeshare.

Boston dataset

The dataset contains information from the U.S. Census Service concerning housing in the Boston area. The dataset's size is small, with only 506 cases.

It consists of 14 attributes per observation. Specifically, the features are:

- CRIM: per capita crime rate by the town;
- ZN: proportion of residential land zoned for lots over 25.000 square feet;
- INDUS: proportion of non-retail business acres per town;
- CHAS: Charles River dummy variable taking the values of 1 if tract bounds river or 0 otherwise;
- NOX: nitric oxides concentration, parts per 10 million;
- RM: average number of rooms per dwelling;
- AGE: proportion of owner-occupied units built before 1940;
- DIS: weighted distances to five Boston employment centers;
- RAD: index of accessibility to radial highways;

- TAX: full-value property-tax rate per \$10,000;
- PTRATIO: pupil-teacher ratio by the town;
- B: a numerical base on the proportion of blacks Bk, such that $Bk = 1000(Bk - 0.63)^2$;
- LSTAT: percentage of the lower status of the population;
- MEDV: median value of owner-occupied homes in $1000's$.

Titanic dataset

The titanic dataset describes the survival status of individual passengers on the Titanic. The dataset does not contain the crew's information, but it includes the actual ages of half of the passengers. The principal source for data about Titanic passengers is the Encyclopedia Titanica. One of the sources is Eaton & Haas "Titanic: Triumph and Tragedy, Patrick Stephens Ltd," which includes a passenger list created by many researchers and edited by Michael A. Findlay.

There are 15 attributes in each case of the dataset. More in detail, the features are:

- Pclass: refers to passenger class (1st, 2nd, 3rd) and is a proxy for socio-economic class;
- survival: a dichotomous variable signaling survival, zero in the case of non-survival one otherwise;
- name: the name of the passenger;
- sex: the sex of the passenger, male or female;
- age: age in years, with some missing values;
- sibsp: the number of siblings or spouses aboard the Titanic;
- parch: the number of parents or children aboard the Titanic;
- ticket: the ticket number;
- fare: the fare paid in British Pounds;
- cabin: the cabin number, if available;
- embarked: the port in which the passenger embarked (C = Cherbourg, Q = Queenstown, S = Southampton);

- boat: the lifeboat number, if available;
- body: the body number, if available;
- home.dest: the passenger's home destination.

Capital-Bikeshare dataset

The dataset contains daily ridership data of the Capital Bikeshare system in Washington, D.C., for the years 2011 and 2012.

There are 731 observations in total, where the data on each day is recorded as a single observation with 14 variables, namely:

- dteday: timestamp of the observation with the format: yyyy-mm-dd;
- season: seasonal feature with the following encoding:
  - 1: winter;
  - 2: summer;
  - 3: fall;
  - 4: spring.
- year: year of the observation, encoded with 0 for 2011 and 1 for 2012;
- mnth: month feature with the following encoding:
  - 1: January;
  - 2: February;
  - 3: March;
  - 4: April;
  - 5: May;
  - 6: June;
  - 7: July;
  - 8: August;
  - 9: September;
  - 10: October;
  - 11: November;
  - 12: December.

- holiday: whether or not the day is a public holiday with value zero in case of not a holiday one otherwise;
- weekday: day of the week with the following encoding:
  - 0: Sunday;
  - 1: Monday;
  - 2: Tuesday;
  - 3: Wednesday;
  - 4: Thursday;
  - 5: Friday;
  - 6: Saturday.
- workingday: whether or not the day is a working day, in other words, the day is neither a weekend day nor a public holiday;
- weathersit: weather condition with the following encoding:
  - 1: clear/partly cloudy;
  - 2: cloudy/misty;
  - 3: light rain/light snow.
- temp: normalized temperature in Celsius, calculated as follows:

  $$temp = \frac{t - tmin}{tmax - tmin}$$

  with tmin=-8 and tmax=+39;
- atemp: normalized feeling temperature in Celsius, calculated as follows:

  $$atemp = \frac{t - tmin}{tmax - tmin}$$

  where tmin=-16 and tmax=+50;
- hum: normalized humidity measured as a percentage divided by 100;
- windspeed: wind speed measured in miles per hour divided by 67;
- casual: daily ridership count of non-registered users;
- registered: daily ridership count of registered users;
- cnt: total daily ridership count including registered and non-registered users.

Lost in a black-box? Interpretable Machine Learning for assessing Italian SMEs default

The current appendix contains the following additional materials:

Item 1 Pearson correlation coefficients of the financial features used during the modeling process;

Item 2 Classifier development and evaluation flowchart;

Item 3 Scatterplots of the hyperparameter optimization routines of both eXtreme Gradient Boosting and Feedforward Neural Network models with respect to the Logistic Regression model;

Item 4 Models performances under different sampling schemes;

Item 5 Models performances under different train/test ratios, considering features one and two years before the target variable;

Item 6 Notes on the Italian institutional framework in the context of firms' bankruptcy.

Item 1

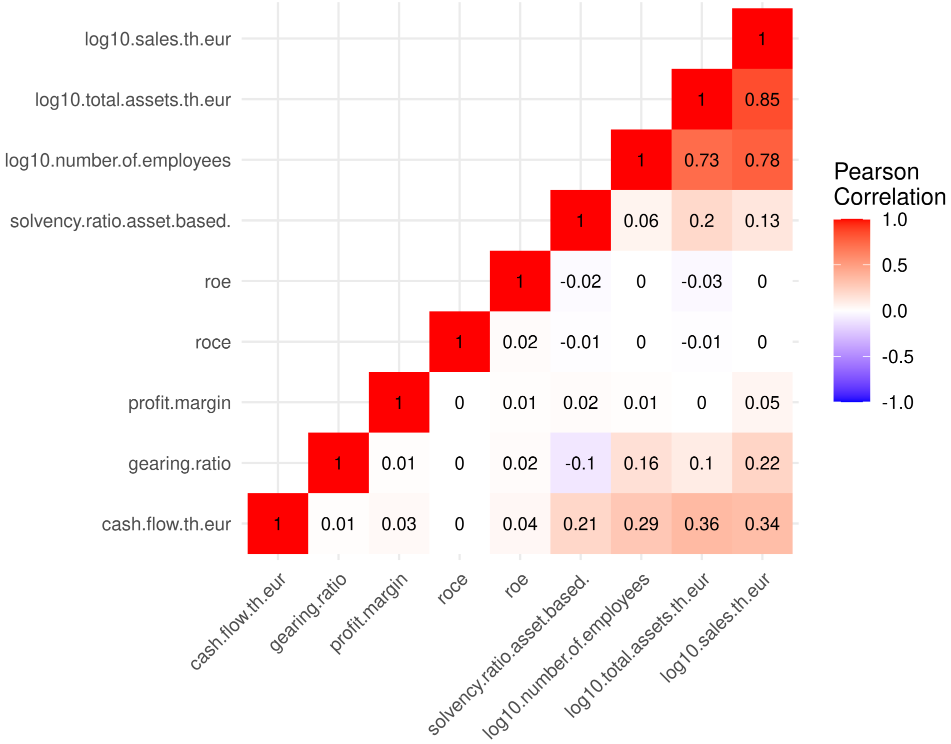


Figure 1: Pearson correlation coefficients of the relevant financial features. Higher positive correlations are in red, whereas negative ones are in blue.

Item 2

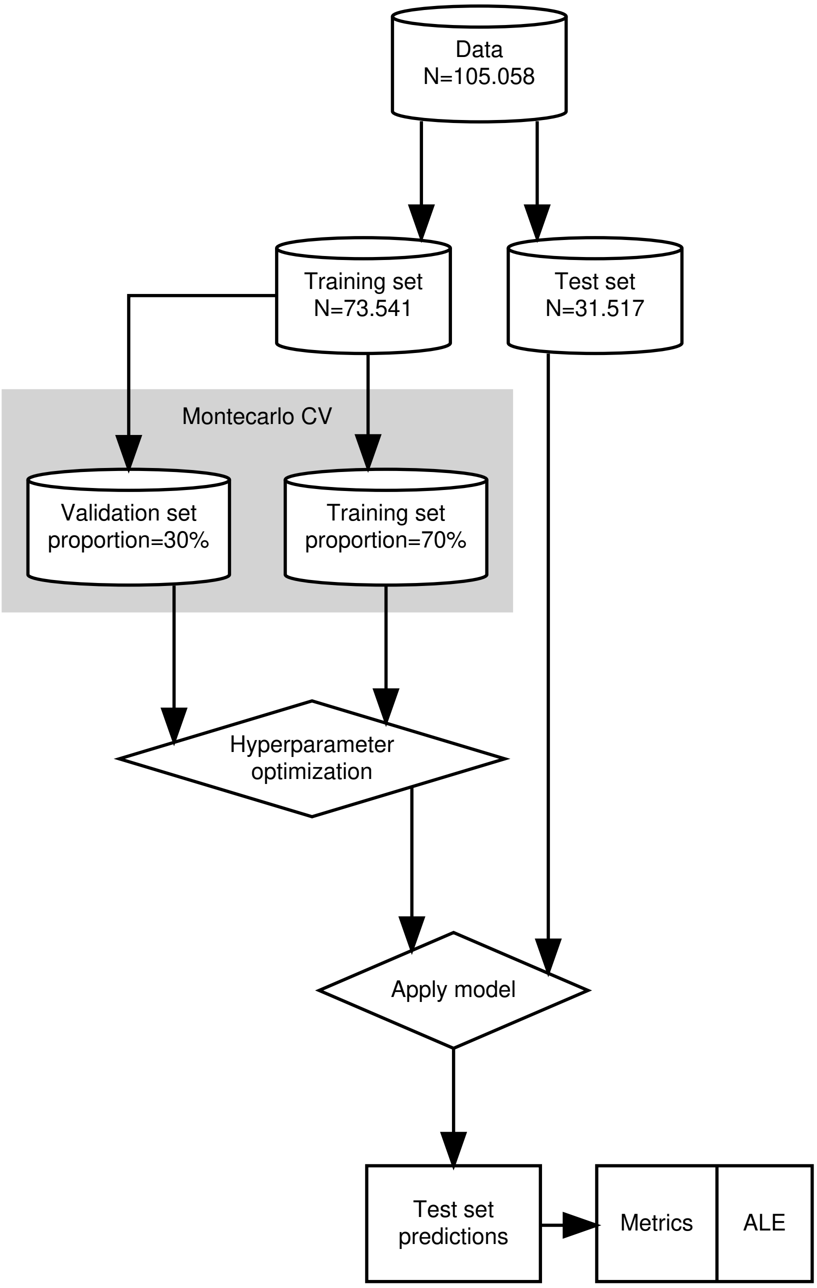
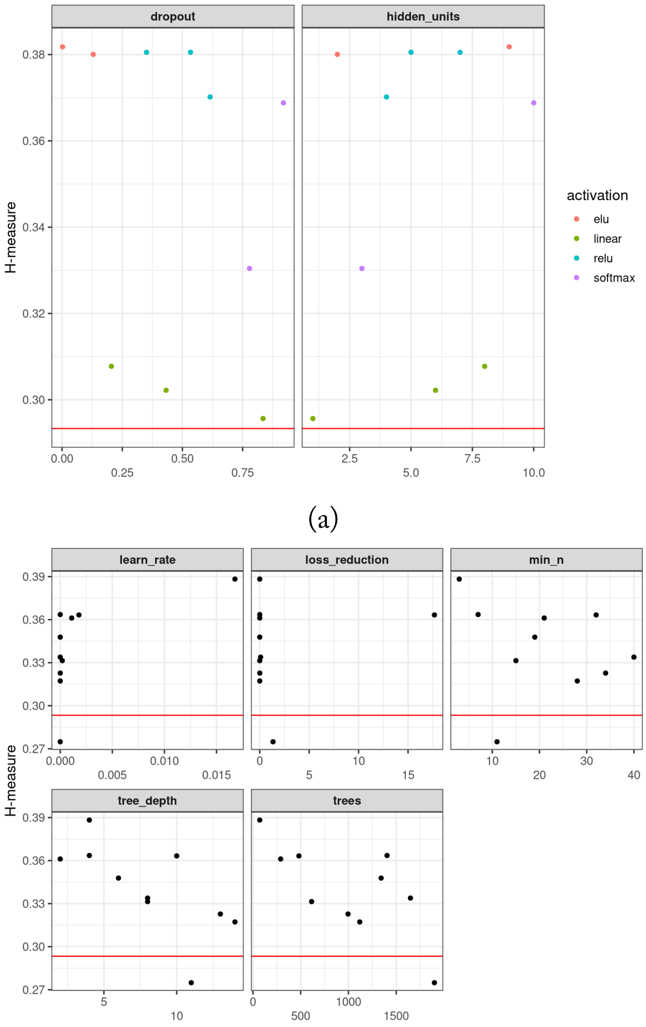


Figure 2: Classifier development and evaluation process.

Item 3



(a)

(b)

Figure 3: Performances of the Feedforward Neural Network (a) and eXtreme Gradient Boosting (b) models under different hyperparameters. The performances, measured as mean H-measures under Montecarlo Crossvalidations and compared to the baseline model, the Logistic Regression (red line).

Item 4

Table 1: Classification results according to different resampling schemes and for the whole dataset.

Model	Sampling scheme	Sensitivity	Specificity	H-measure	AUC	Brier Score	KS Statistics
FANN	whole dataset	0.000	1.000	0.385	0.830	0.081	0.469
FANN	Undersampling	0.694	0.829	0.391	0.827	0.187	0.501
FANN	SMOTE	0.625	0.872	0.373	0.837	0.167	0.523
FANN	RobROSE	0.770	0.692	0.309	0.793	0.112	0.362
XGBoost	whole dataset	0.039	0.999	0.406	0.803	0.019	0.551
XGBoost	Undersampling	0.821	0.719	0.383	0.843	0.146	0.552
XGBoost	SMOTE	0.559	0.930	0.418	0.852	0.086	0.548
XGBoost	RobROSE	0.613	0.842	0.335	0.771	0.024	0.521
BGEVA	whole dataset	0.002	1.000	0.287	0.799	0.021	0.437
BGEVA	Undersampling	0.752	0.727	0.331	0.819	0.178	0.481
BGEVA	SMOTE	0.657	0.810	0.309	0.813	0.157	0.463
BGEVA	RobROSE	0.809	0.634	0.298	0.807	0.191	0.451
LR	whole dataset	0.010	0.999	0.281	0.796	0.022	0.418
LR	Undersampling	0.745	0.736	0.303	0.809	0.151	0.483
LR	SMOTE	0.662	0.808	0.306	0.811	0.158	0.461
LR	RobROSE	0.824	0.638	0.310	0.814	0.179	0.452
Probit	whole dataset	0.003	1.000	0.280	0.795	0.021	0.43
Probit	Undersampling	0.738	0.737	0.299	0.809	0.190	0.448
Probit	SMOTE	0.627	0.809	0.282	0.799	0.420	0.331
Probit	RobROSE	0.120	0.987	0.074	0.554	0.192	0.459

Item 5

Table 2: Models' performances on the test set from a 60-40 split based on features one year previous the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.368	0.838	0.186	0.529
XGBoost	0.264	0.79	0.188	0.424
BGEVA	0.276	0.79	0.187	0.438
LR	0.263	0.78	0.189	0.422
Probit	0.39	0.853	0.163	0.55

Table 3: Models' performances on the test set from a 70-30 split based on features one year previous the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.391	0.827	0.187	0.501
XGBoost	0.383	0.843	0.146	0.552
BGEVA	0.331	0.819	0.178	0.481
LR	0.303	0.809	0.151	0.483
Probit	0.299	0.809	0.19	0.448

Table 4: Models' performances on the test set from an 80-20 split based on features one year previous the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.444	0.867	0.19	0.486
XGBoost	0.345	0.798	0.14	0.541
BGEVA	0.343	0.82	0.154	0.446
LR	0.322	0.815	0.155	0.431
Probit	0.323	0.813	0.157	0.437

Table 5: Models' performances on the test set from a 90-10 split based on features one year previous the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.44	0.872	0.171	0.55
XGBoost	0.364	0.827	0.136	0.564
BGEVA	0.333	0.82	0.146	0.533
LR	0.321	0.814	0.176	0.505
Probit	0.321	0.815	0.186	0.448

Table 6: Models' performances on the test set from a 60-40 split based on features two years previous to the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.175	0.746	0.209	0.38
XGBoost	0.189	0.754	0.185	0.375
BGEVA	0.164	0.729	0.181	0.35
LR	0.157	0.728	0.19	0.342
Probit	0.157	0.729	0.207	0.343

Table 7: Models' performances on the test set from a 70-30 split based on features two years previous to the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.103	0.62	0.222	0.241
XGBoost	0.211	0.765	0.161	0.395
BGEVA	0.149	0.716	0.205	0.304
LR	0.147	0.713	0.205	0.306
Probit	0.156	0.713	0.205	0.29

Table 8: Models' performances on the test set from an 80-20 split based on features two years previous to the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.103	0.62	0.222	0.241
XGBoost	0.211	0.765	0.161	0.395
BGEVA	0.149	0.716	0.205	0.304
LR	0.147	0.713	0.205	0.306
Probit	0.156	0.713	0.205	0.29

Table 9: Models' performances on the test set from a 90-10 split based on features two years previous to the target variable.

Model	H-measure	AUC	Brier Score	KS Statistics
FANN	0.204	0.754	0.215	0.396
XGBoost	0.244	0.785	0.203	0.446
BGEVA	0.198	0.731	0.211	0.365
LR	0.192	0.733	0.208	0.351
Probit	0.192	0.736	0.212	0.357

Item 6

In Italy, the bankruptcy regime is generally considered to be pro-debtor, meaning that it is designed to protect the interests of the debtor and to allow them to restructure their debt and continue operating their business.

Under Italian law, there are several options available to individuals and businesses who are facing financial difficulties. One option is the so-called ”concordato preventivo,” which is a procedure that allows the debtor to negotiate a repayment plan with their creditors. If the plan is approved by the court, it becomes binding on all creditors and allows the debtor to avoid bankruptcy.

Another option is the ”fallimento,” which is the Italian equivalent of bankruptcy. Under this procedure, the debtor's assets are liquidated and the proceeds are used to pay off their debts. The fallimento procedure is generally considered to be a last resort, as it can have significant consequences for the debtor and their business.

It is important to note that the Italian bankruptcy regime is not static and has undergone several reforms in recent years. For example, in 2015, the Italian government introduced a new law that made it easier for small and medium-sized enterprises to access the concordato preventivo procedure, in an effort to promote entrepreneurship and support the Italian economy.

Overall, the Italian bankruptcy regime is relevant in that it can influence the outcome of a bankruptcy case and the options available to the debtor. It is important for debtors and creditors to be familiar with the legal framework and the various procedures that are available in order to navigate the bankruptcy process effectively.

The legal framework in a particular country can be an important factor in a bankruptcy model, as it can influence the options available to the debtor and the consequences of bankruptcy. For example, in a pro-debtor regime like Italy, debtors may have more options available to them to restructure their debt and avoid bankruptcy, which could reduce the probability of bankruptcy as predicted by the model. In contrast, in a pro-creditor regime, the options available to the debtor may be more limited and the consequences of bankruptcy may be more severe, which could increase the probability of bankruptcy as predicted by the model.

It is important to note that bankruptcy models are just one tool that can be used to analyze the financial health of a company or individual and are not meant to be used in isolation. They should be used in conjunction with other tools and analysis to provide a more comprehensive understanding of the financial situation.

Multicriteria interpretability driven Deep Learning

The current appendix contains the following additional materials:

Item 1 Dataset financial indicators with corresponding descriptions and effect on bankruptcy probability (Table 10);

Item 2 Pearson correlation coefficients of the dataset's financial features: one, two, and three years before the bankruptcy (respectively, Figure 4, Figure 5 and 6);

Item 3 Accumulated Local Effects of the Multilayer Perceptron models with and without knowledge injections under two and three years before bankruptcy data (respectively, Figure 7, Figure 8);

Item 4 Accumulated Local Effects of the Multilayer Perceptron models with and without knowledge injections under one year before bankruptcy data, using variables with less than $|0.1|$ Pearson correlation (Figure 9);

Item 5 Accumulated Local Effects of the Multilayer Perceptron models with and without knowledge injections under one year before bankruptcy data, using a downsample 1:1 scheme (Figure 10);

Item 6 Notes on the Polish institutional framework in the context of firms' bankruptcy.

Item 1

Table 10: Dataset financial indicators with the corresponding description.

ID	Description	Effect	ID	Description	Effect
Attr 1	net profit / total assets	↘	Attr 33	operating expenses / short-term liabilities	↗
Attr 2	total liabilities / total assets	↗	Attr 34	operating expenses / total liabilities	↗
Attr 3	working capital / total assets	↘	Attr 35	profit on sales / total assets	↘
Attr 4	current assets / short-term liabilities	↘	Attr 36	total sales / total assets	↘
Attr 5	{[](cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation){]} * 365	↗	Attr 37	(current assets - inventories) / long-term liabilities	↘
Attr 6	retained earnings / total assets	↘	Attr 38	constant capital / total assets	↗
Attr 7	EBIT / total assets	↘	Attr 39	profit on sales / sales	↘
Attr 8	book value of equity / total liabilities	↘	Attr 40	(current assets - inventory - receivables) / short-term liabilities	↗
Attr 9	sales / total assets	↘	Attr 41	total liabilities / ((profit on operating activities + depreciation) * (12/365))	↗
Attr 10	equity / total assets	↗	Attr 42	profit on operating activities / sales	↘
Attr 11	(gross profit + extraordinary items + financial expenses) / total assets	↗	Attr 43	rotation receivables + inventory turnover in days	↗
Attr 12	gross profit / short-term liabilities	↘	Attr 44	(receivables * 365) / sales	↗
Attr 13	(gross profit + depreciation) / sales	↘	Attr 45	net profit / inventory	↘
Attr 14	(gross profit + interest) / total assets	↘	Attr 46	(current assets - inventory) / short-term liabilities	↗
Attr 15	(total liabilities * 365) / (gross profit + depreciation)	↗	Attr 47	(inventory * 365) / cost of products sold	↗
Attr 16	(gross profit + depreciation) / total liabilities	↘	Attr 48	EBITDA (profit on operating activities - depreciation) / total assets	↘
Attr 17	total assets / total liabilities	↘	Attr 49	EBITDA (profit on operating activities - depreciation) / sales	↘
Attr 18	gross profit / total assets	↘	Attr 50	current assets / total liabilities	↘
Attr 19	gross profit / sales	↘	Attr 51	short-term liabilities / total assets	↗
Attr 20	(inventory * 365) / sales	↗	Attr 52	(short-term liabilities * 365) / cost of products sold)	↗
Attr 21	sales (n) / sales (n-1)	↘	Attr 53	equity / fixed assets	↗
Attr 22	profit on operating activities / total assets	↘	Attr 54	constant capital / fixed assets	↗
Attr 23	net profit / sales	↘	Attr 55	working capital	↘
Attr 24	gross profit (in 3 years) / total assets	↘	Attr 56	(sales - cost of products sold) / sales	↘
Attr 25	(equity - share capital) / total assets	↗	Attr 57	(current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)	↗
Attr 26	(net profit + depreciation) / total liabilities	↘	Attr 58	total costs /total sales	↗
Attr 27	profit on operating activities / financial expenses	↘	Attr 59	long-term liabilities / equity	↗
Attr 28	working capital / fixed assets	↘	Attr 60	sales / inventory	↘
Attr 29	logarithm of total assets	↘	Attr 61	sales / receivables	↘
Attr 30	(total liabilities - cash) / sales	↗	Attr 62	(short-term liabilities *365) / sales	↗
Attr 31	(gross profit + interest) / sales	↘	Attr 63	sales / short-term liabilities	↘
Attr 32	(current liabilities * 365) / cost of products sold	↗	Attr 64	sales / fixed assets	

Item 2

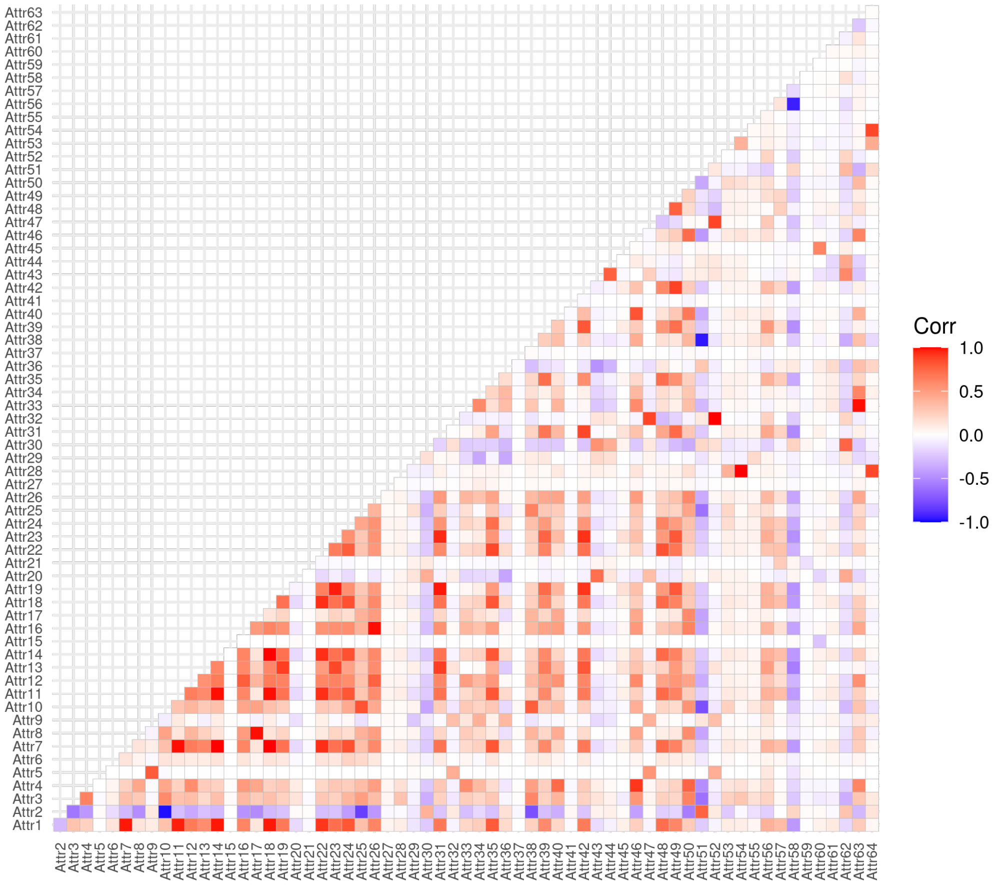


Figure 4: One year before bankruptcy, Pearson correlation coefficients of the dataset's financial features. Higher positive correlations are colored in red, whereas negative ones are in blue.

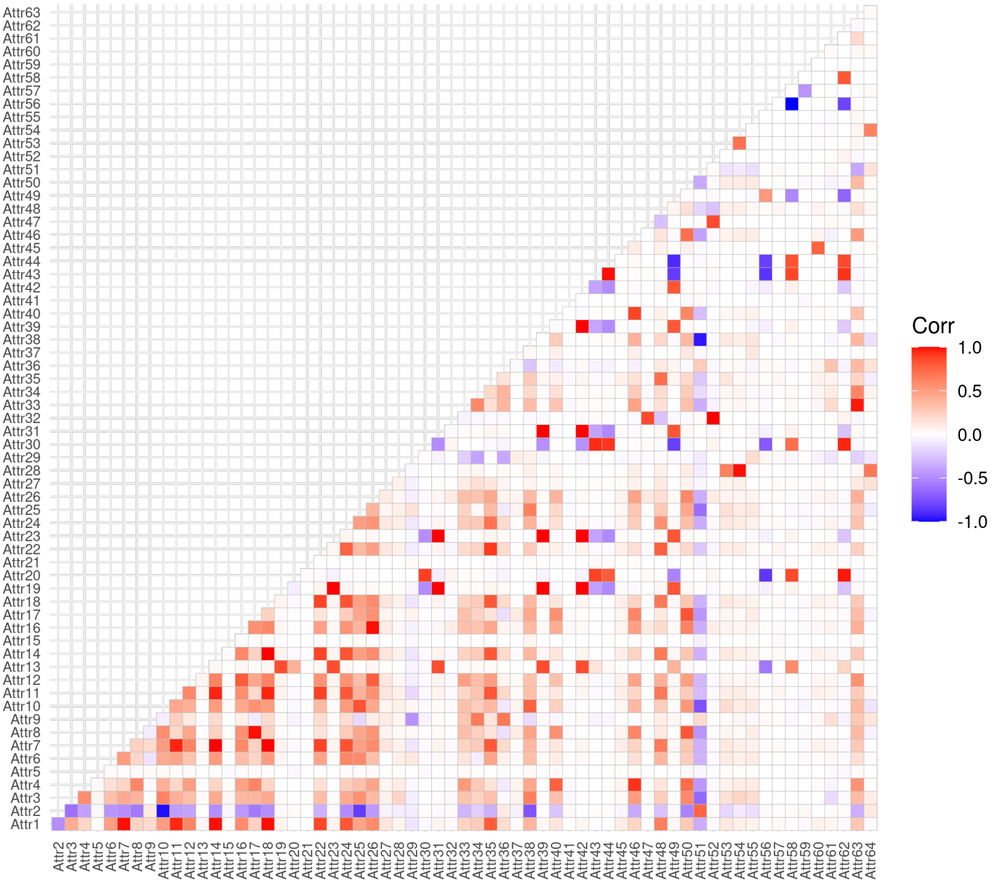


Figure 5: Two years before the bankruptcy, Pearson correlation coefficients of the dataset's financial features. Higher positive correlations are colored in red, whereas negative ones are in blue.

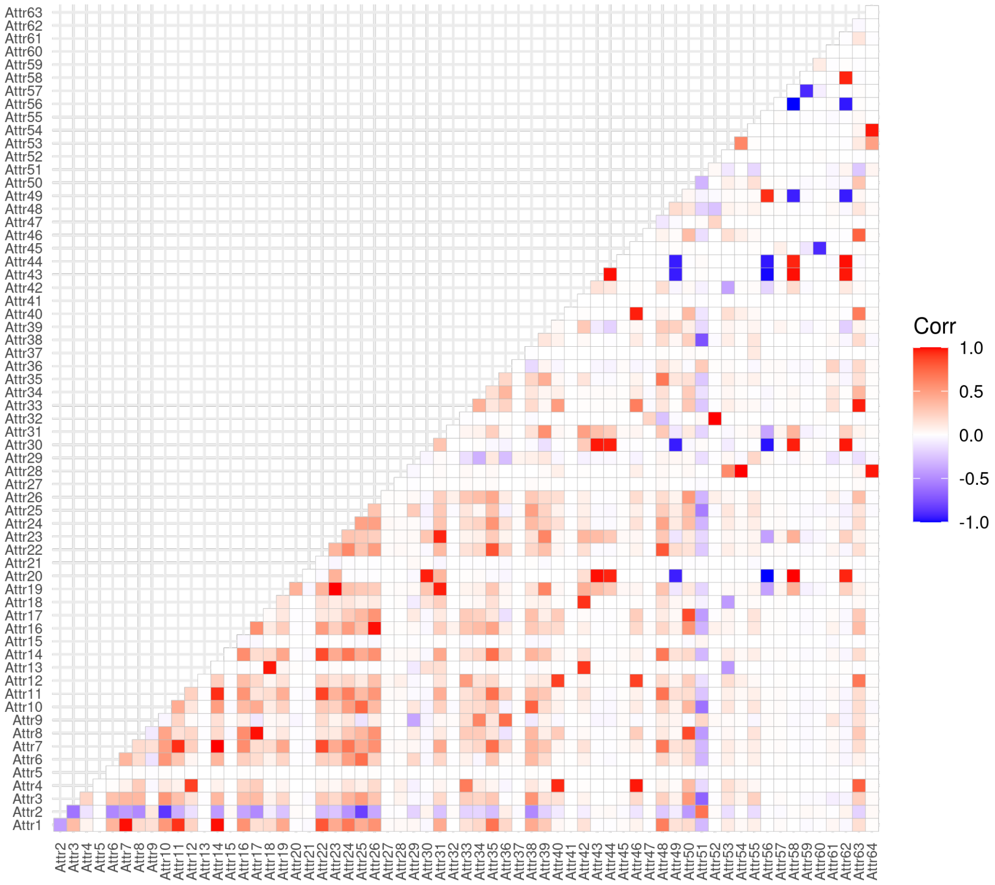
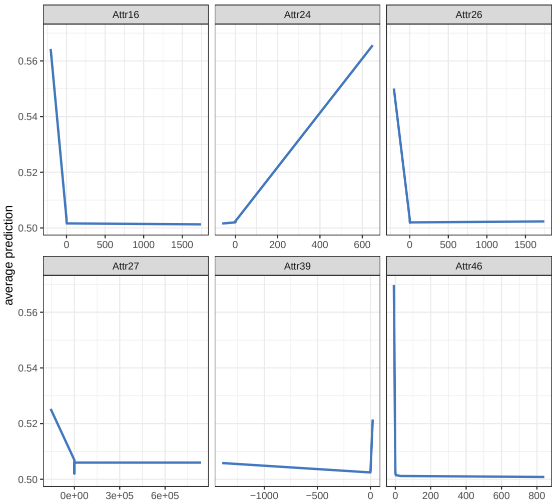
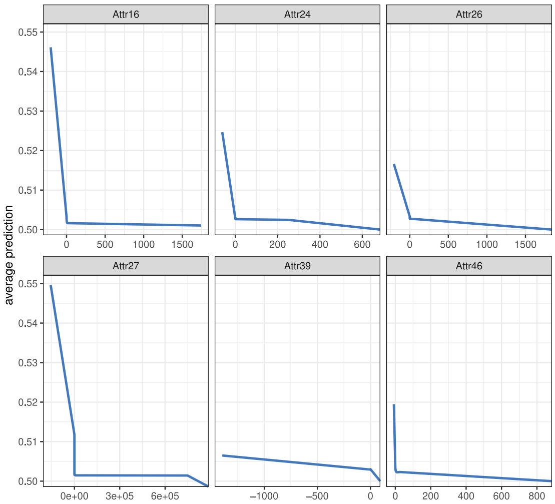


Figure 6: Three years before the bankruptcy, Pearson correlation coefficients of the dataset's financial features. Higher positive correlations are colored in red, whereas negative ones are in blue.

Item 3

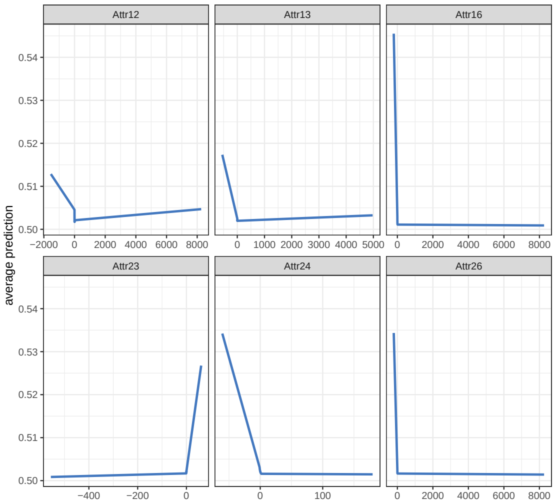


(a) Accumulated Local Effects plot of the Multilayer Perceptron architecture, without regularization and knowledge injection (i.e. $\lambda_1 = 1, \lambda_2 = 0.0, \lambda_3 = 0.0$).

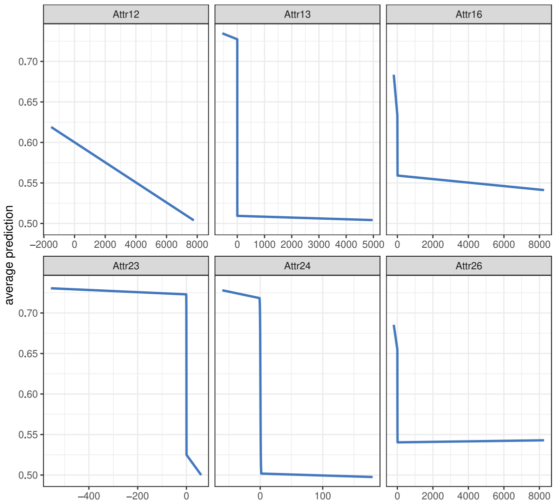


(b) Accumulated Local Effects plot of the Multilayer Perceptron architecture, with regularisation and knowledge injection optimally selected from the hyperparameter optimization procedure (i.e. $\lambda_1 = 0.4, \lambda_2 = 0.2, \lambda_3 = 0.4$).

Figure 7: Two years before the bankruptcy, Accumulated Local Effects plot of the Multilayer Perceptron architecture, with and without regularization and knowledge injection.



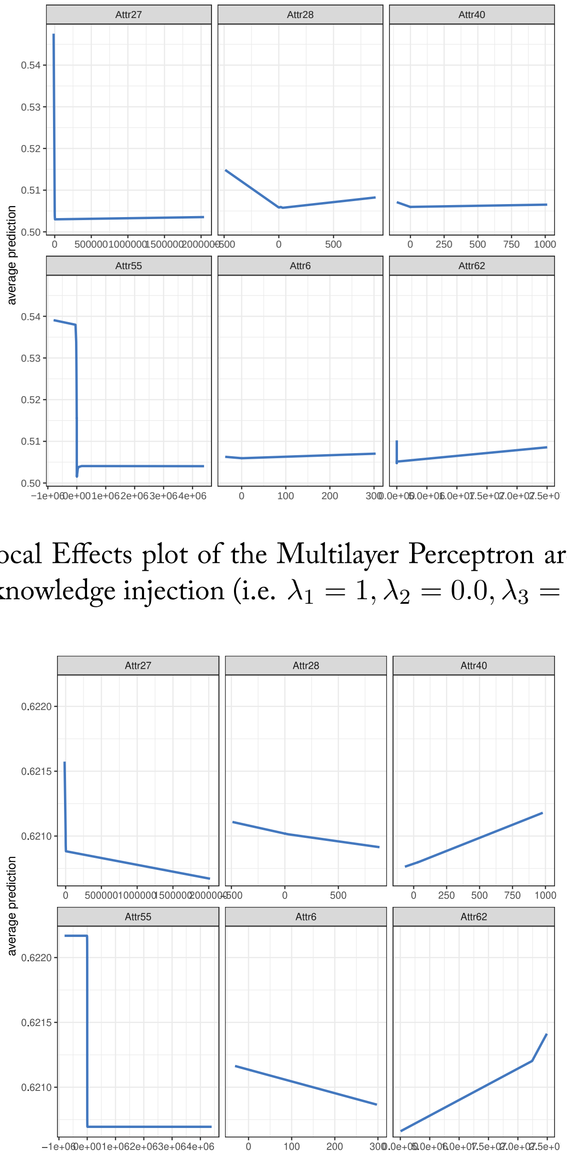
(a) Accumulated Local Effects plot of the Multilayer Perceptron architecture, without regularization and knowledge injection (i.e. $\lambda_1 = 1, \lambda_2 = 0.0, \lambda_3 = 0.0$).



(b) Accumulated Local Effects plot of the Multilayer Perceptron architecture, with regularisation and knowledge injection optimally selected from the hyperparameter optimization procedure (i.e. $\lambda_1 = 0.5, \lambda_2 = 0.0, \lambda_3 = 0.5$).

Figure 8: Three years before the bankruptcy, Accumulated Local Effects plot of the Multilayer Perceptron architecture, with and without regularization and knowledge injection.

Item 4

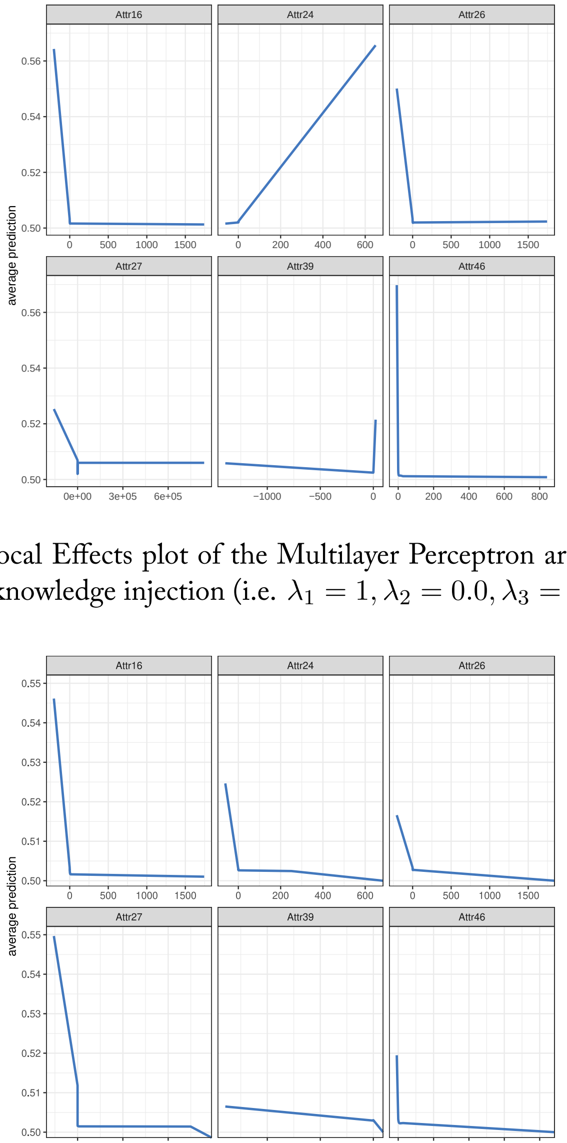


(a) Accumulated Local Effects plot of the Multilayer Perceptron architecture, without regularization and knowledge injection (i.e. $\lambda_1 = 1, \lambda_2 = 0.0, \lambda_3 = 0.0$).

(b) Accumulated Local Effects plot of the Multilayer Perceptron architecture, with regularisation and knowledge injection optimally selected from the hyperparameter optimization procedure (i.e. $\lambda_1 = 0.3, \lambda_2 = 0.1, \lambda_3 = 0.6$).

Figure 9: One year before bankruptcy, Accumulated Local Effects plots of the Multilayer Perceptron architecture, with and without regularization and knowledge injection using variables with less than $|0.1|$ Pearson correlation.

Item 5



(a) Accumulated Local Effects plot of the Multilayer Perceptron architecture, without regularization and knowledge injection (i.e. $\lambda_1 = 1, \lambda_2 = 0.0, \lambda_3 = 0.0$).

(b) Accumulated Local Effects plot of the Multilayer Perceptron architecture, with regularisation and knowledge injection optimally selected from the hyperparameter optimization procedure (i.e. $\lambda_1 = 0.5, \lambda_2 = 0.0, \lambda_3 = 0.5$).

Figure 10: One year before bankruptcy, Accumulated Local Effects plots of the Multilayer Perceptron architecture, with and without regularization and knowledge injection, using a downsample 1:1 scheme.

Item 6

In Poland, the bankruptcy regime is generally considered to be pro-creditor, as the main focus of the system is on the repayment of debts to creditors. This is reflected in the provisions of the Polish Bankruptcy and Rehabilitation Law, which sets out the procedures for bankruptcy and debt restructuring in Poland.

Under the Polish bankruptcy regime, the primary goal of bankruptcy proceedings is to liquidate the debtor's assets and distribute the proceeds to creditors in accordance with the priority of their claims. In general, secured creditors (such as banks with a security interest in the debtor's assets) are given priority over unsecured creditors in the distribution of assets.

The Polish bankruptcy regime also provides for the possibility of debt restructuring, through which the debtor and creditors can negotiate a plan to restructure the debtor's debts and potentially avoid bankruptcy. However, the success of debt restructuring proceedings depends on the willingness of the debtor and creditors to negotiate and reach an agreement, and on the debtor's ability to make the required payments under the restructuring plan.

In summary, the Polish bankruptcy regime is generally considered to be pro-creditor, as it prioritizes the repayment of debts to creditors and provides for the liquidation of the debtor's assets in the event of bankruptcy. However, the possibility of debt restructuring can provide an opportunity for debtors to avoid bankruptcy and negotiate a plan to repay their debts.