Computational Economics (2022) 59:263–295
https://doi.org/10.1007/s10614-020-10078-2

Predicting Firm-Level Bankruptcy in the Spanish Economy Using Extreme Gradient Boosting

Matthew Smith[1,2] · **Francisco Alvarez**[3]

Accepted: 23 November 2020 / Published online: 6 January 2021

Abstract

We apply a machine learning (ML) algorithm in order to predict bankruptcy rates among companies within the Spanish economy from 1992 to 2016. The model identifies some relevant variables when predicting bankruptcy: such as the ratio total liabilities to total assets or current liability to financial expenses along with size factors such as the log of sales. Additionally, the model allows us to analyse firms individually: the marginal contribution of a given variable to the firm's prediction depends on all its other observed characteristics. This can be particularly useful in analysing case by case lending decisions within financial institutions. An exercise on the cost of extending the forecasting horizon up to 4 years ahead is also provided, as financial institutions are naturally interested in the early detection of bankruptcy. We also compare XGBoost to a number of ML models, such as a Logistic Model, Support Vector Machine, Neural Network, Random Forest and LightGBM.

Keywords Extreme gradient boosting · Machine learning · Bankruptcy prediction · Non-linear modelling

JEL Classification G17 (Financial forecasting and simulation) · G33 (Bankruptcy) · C53(Forecasting and prediction methods)

We thank S. Sacchetto and two anonymous referees for very significant and insightful comments. Any remaining errors are our own. Alvarez is supported by Grant ECO2017-86245-P from Ministerio de Ciencia Innovación y Universidades. The research in this study was carried out using an Amazon Web Services (AWS) EC2 instance in which Smith received an AWS Research Grant.

✉ Matthew Smith
msmith01@ucm.es

Francisco Alvarez
fralvare@ccee.ucm.es

[1] Universidad Complutense Madrid, Madrid, Spain

[2] Barcelona Supercomputing Center, Barcelona, Spain

[3] Department of Economic Analysis, ICAE, Universidad Complutense Madrid, Madrid, Spain

1 Introduction

Recent developments in regulatory requirements from central banks and governments have caused credit risk management to take a leading role amongst practitioners. The Basel agreement requires financial institutions to limit the amount of risk-weighted assets that a bank can hold, this affects the type and number of loans that a bank can issue in relation to its capital. Therefore, it has become more apparent to the banking industry that new and improved, more innovative ways should be adopted in order to identify counter-party default risk amongst its corporate clients early on. On a more economic scale the ability to adequately shift financial resources from an ailing firm to more positive recipients will help clear up inefficiencies within the financial lending sectors. Forecasting the failure of an organisation is an important economic and financial challenge, failure to adequately manage financially distressed firms within a timely manner in an economy can have profound negative economic consequences. Additionally the insolvency procedure can stretch across many years. Hernandez-Tinoco and Wilson (2013) found that UK firms have an average time gap of 1.17 years from the events which caused the firm to go bankrupt and the date in which bankruptcy was filed. Theodossiou (1993) found that firms in the US fail to provide financial accounts 2 years prior to bankruptcy. This suggests that firms feel the struggle of financial distress before filing for bankruptcy and it is therefore critical for financial institutions to identify these firms early on.

This paper addresses the issue of new and innovative bankruptcy prediction models by applying a Machine Learning algorithm in order to better classify and distinguish bankrupt firms from non-bankrupt firms. We apply our model over 4 years of financial accounts, aiming to identify financially distressed firms early on. We aim to capture the rarity of financially troubled firms in an economy by using an imbalanced dataset and finally, we aim to capture the imperfectness of financial data through the inclusion of extreme and missing values as information. Moreover we show that Machine Learning models need not give just a simple black-box prediction but can be interpretable through a number of ways as in traditional regression models. We finally compare our results with a series of other Machine Learning models, notably a Support Vector Machine (SVM), Neural Network, Logistic Regression, Random Forest and Light Gradient Boosted Machine (LightGBM). The higher predictive accuracy and interpretability are just two ways why the model proposed in this paper is considered one of the most prominent Machine Learning models currently in use and therefore would be well posed for the problem of bankruptcy prediction.

The problem of bankruptcy prediction is well suited for classical binary Machine Learning classification problems. A company can either be in a state of bankruptcy or not. The companies with the status of bankruptcy are described as the positive class since we wish to positively identify these firms. The negative class are the non-bankrupt firms, which are included such that the model can learn the differences between the two classes. When a model correctly predicts that a company is in a state of bankruptcy, then it is called a True Positive (TP). Conversely when a model

correctly predicts that a company is in a state of non-bankruptcy, then it is called a True Negative (TN). The case when a model predicts that a company is bankrupt but the actual status is non-bankrupt is called a False Positive (FP or Type I error). Finally, the case when the model predicts that a company is non-bankrupt but the actual status is bankrupt is called a False Negative (FN or Type II error). The values TP, FP, FN, TN conform, by rows, a 2×2 matrix denoted as the confusion matrix.

The first challenge corresponds to analysing the cost of extending the forecasting horizon. Predicting bankruptcy early is an important factor in any lending policy. We make predictions for firms that went bankrupt using data 1, 2, 3 and 4 years prior to bankruptcy, in all cases using the whole dataset (including firms that remained active for the whole period). A number of statistics based on a confusion matrix have been computed, all of them with a common—and expected—message: extending the forecasting horizon impairs future predictions. The relevant question, of course, is to quantify this impairment through relevant statistics. A representative statistic of the performance of any classification algorithm is the Area Under the Curve (AUC) or the Area Under the Precision-Recall curve (AUPRC). The AUC takes a value of 1 for perfect classification, that is, 0 Type I and II errors, while it takes a value of 0.5 for pure random guessing.[1] The AUPRC takes value 1 for a perfect model and 0 for a poor model and is more informative than the AUC for imbalanced data sets. Our model achieves AUC values ranging from 0.84 to 0.74 and AUPRC values ranging from 0.66 to 0.42 for 1–4 year prior prediction respectively. It is worth noting that these values can be understood as *optimal* in some sense. Roughly, the goal of Extreme Gradient Boosting (XGBoost) is to push the limit of computational resources for boosted tree algorithms. XGBoost minimizes numerically a loss function which contains a number of parameters which need to be optimised. For this we carried out a grid search on a parameter space in order to minimise prediction errors on an in-sample test set using 10 fold cross validation, which is briefly discussed in more detail later. This implies that we located the parameter values which maximised the in-sample test AUC, AUPRC and minimise some other loss functions. We report our final analysis and statistics on a held-out test set.

Next, we analyse which variables have a higher impact on the likelihood of bankruptcy. As mentioned, the XGBoost model minimizes a loss function. More precisely, the algorithm selects a loss minimizing collection of trees, denoted by ϕ. Each tree in ϕ is a step function that assigns a score to each firm depending on the firm's characteristics. The overall score of the firm is just the sum of these scores across all trees in ϕ. There is a standard monotonic function that maps overall scores into a probability of bankruptcy and, finally, the firm is predicted to be bankrupt whenever that probability overtakes a certain threshold, which we define. This scheme provides a natural environment to identify the most important variables in predicting bankruptcy. On average, across firms, the variables with the highest marginal contribution to the overall score are the ratio Total Liabilities to Total Assets (TL.TA), the logarithm of Total Assets (logTA), the logarithm of Sales (logSALES), Current Liabilities to Financial Expense (CL.FinExp) and Earnings

[1] It takes value 0 for a *perfect misclassification*, which can trivially be turned into a perfect classification.

Before Interest and Tax (EBIT.FinExp). Our analysis indicates that there are slight variations depending on the forecasting horizon: when we take 1 year prior predictions the most relevant characteristic is TL.TA, while for longer horizons it turns out to be logTA or logSALES. Perhaps more importantly, the across-firm variability is essential for any lending policy. The algorithm allows us to compute marginal impacts of each variable within each firm. A case study is provided for each quadrant of the confusion matrix. Generally, we illustrate the contribution of each variables changes from one firm to another in a way that is highly non-linear but roughly well captured by ϕ.

Our final step is to compare XGBoost to other Machine Learning models. The key difference between the models is not non-linear vs linear—although this is an important characteristic of the models—but on how to deal with complexity. The loss function to be minimized under the XGBoost algorithm has two terms: the prediction errors and the overall complexity of ϕ, respectively.[2] The parameters which help control the complexity are decided by the practitioner, through domain knowledge and cross validation, otherwise default parameter values are given. Once the parameters are chosen, the algorithm starts from a very simple tree structure and recursively adds trees as to minimise an objective function up to a maximum number of trees, again usually determined at the cross validation stage.[3] To summarize, for a parametrized loss function, the XGBoost model automatically evaluates whether each increment of complexity pays off in terms of error prediction improvement. This automatism is absent in logistic models and other Machine Learning models.

The rest of the paper is organized as follows. Section 2 positions this paper within the existing literature, in Sect. 3 we present the data, Sect. 4 discusses the methodology, Sect. 5 presents the main results, interpretation and case studies, Sect. 6 compares the performance to other Machine Learning models. Finally, Sect. 7 concludes the paper.

2 Previous Literature

The discussion of financial ratios as indicators of bankruptcy stretches as far back as the 1930s, see Fitzpatrick (1932) and Winakor and Smith (1935). Statistical modelling was first introduced through Beaver (1966), who analysed 30 univariate financial ratios one by one in order to classify firms as bankrupt or not. Altman (1968) expanded upon this approach by using multivariate discriminate analysis, constructing a Z-score, a measure of the likelihood of bankruptcy in the US manufacturing sector ($Z > 2.675$: healthy firm, $Z < 2.675$: unhealthy firm). Multivariate conditional probability models were then applied. West (1985) used factor scores and applied a multivariate logistic model as an early warning bankruptcy

[2] The measure of complexity is detailed later in the paper.

[3] In our case we ran our model on a maximum number of 1500 tress and told the algorithm to stop learning once it had failed to learn after 500 trees. That is, the AUC failed to improve for 500 consecutive rounds or trees. The model stopped learning after 88 tress and therefore our model recursively adds trees up to tree number 88.

prediction system. The factors they used closely resemble the components of the CAMEL (*Capital Adequacy, Asset Quality, Earnings & Liquidity*) rating system used by bank examiners at the time. Martin (1977), Ohlson (1980), applied financial ratios to a multivariate logistic model, Zmijewski (1984) to a Probit model. Logistic models are still amongst the most popularly used models since no assumptions are made about the distribution of predictor variables however, Begley et al. (1996) show that traditional models used in Altman (1968) and Ohlson (1980)—which were estimated using data from the 1940s through the 1970s—obtain higher measurement errors when estimated on data from the 1980s, thus traditional models do not generalise well on new, unseen data. Frydman et al. (1985) found that recursive partitioning outperformed discriminant analysis and that additional information can be derived from the assessment of both models.

The next step in the advancement of predicting financial default comes in the form of a subset of artificial intelligence. Throughout the early 1990s artificial neural networks became a popular method, Odom and Sharda (1990), Bell et al. (1990), Hansen and Messier (1991), Tam and Kiang (1992) and through to the mid 1990s, Altman et al. (1994), Wilson and Sharda (1994), Etheridge and Sriram (1996), Pompe and Feelders (1997). The objective in all studies was to capture firm-level counter-party insolvency using balance sheet and income statement data. Recursive partitioning models such as decision trees, random forests and gradient boosting are becoming increasingly common in classification and regression problems. Decision trees are simple and interpretable. Random Forests, see Breiman (2001) are similar but with the added advantage of using multiple decision trees in order to make a classification. Creamer and Freund (2004) were one of the first to apply random forests to the problem of bankruptcy prediction. Barboza et al. (2017) show a series of bagging, boosting and random forest classifiers outperform traditional statistical models. Zhao et al. (2017) uses a Kernel Extreme Learning Machine (KELM) to discriminate bankrupt companies with non-bankrupt companies. They find that KELM performs better than Support Vector Machines, Extreme Learning Machines, Random Forest, Particle Swarm Optimisation Enhanced Fuzzy K-Nearest Neighbour and a Logistic model in terms of overall accuracy, Type I error, Type II error and AUC.

Zhou and Lai (2017) applied AdaBoost to corporate bankruptcy prediction with missing values and found that it performs better than other benchmark models. Adaptive Boosting (AdaBoost) Freund et al. (1996) is similar to XGBoost in that both models build weak learners, However, XGBoost uses a regularised model formalisation to control over-fitting and improves on the recursive tree-based partitioning method of gradient boosting Friedman (2001), Friedman (2002). The model builds sequentially a series of shallow trees, in which each additional tree corrects the residual error from all previous trees. XGBoost is more efficient since it applies a sparsity-aware split finding method when training on sparse data. Zięba et al. (2016) applied an extreme gradient boosting approach to predict bankruptcy within the Polish market. They used 700 bankrupt companies and 10,000 non-bankrupt companies to train and test their model. However, they only report on the mean and standard deviation of an AUC evaluation metric. Carmona et al. (2018) applied the same method to predict bankruptcy amongst U.S. national commercial

banks. They used a balanced model of 78 failed banks and 78 non-failed banks, reporting on the Sensitivity, Specificity, Accuracy and AUC of their model. This study differentiates itself from the previous two studies by using a larger sample of firms with extreme and missing values, we document a number of new evaluation metrics better suited for imbalanced data classes such as AUPRC and MCC. We go deeper into the interpretability of the model and analyse four case studies. Finally, we compare the model to a series of other Machine Learning models. The model proposed in this paper has been more widely applied to credit scoring models as opposed to bankruptcy default, therefore, we contribute through using new models applied to a different kind of dataset. Xia et al. (2017) applied XGBoost to credit risk default and found that it performs as well if not better than the many other Machine Learning models they applied.

While there is extensive literature surrounding the prediction of bankruptcy, much of it has focused on an equally balanced sample of bankrupt and non-bankrupt firms, unrepresentative of bankruptcy filings in the real economy. There are far more non-failed firms operating within an economy than there are failed, therefore, we feel it is important to capture this class imbalance within our study. Other scientific fields are utilising Machine Learning models on such imbalanced data, it is frequently used in medicine when classifying patients with a rare disease from a large population. Computer sciences use imbalanced data when classifying images of a specific type from a population of many images, i.e. classifying correctly stop signs from all other images taken in automated driving. The model proposed in this paper has been recognised by CERN as the leading method in identifying the extremely rare Higgs Boson particle from the Large Hadron Collider.[4]

Olmeda and Fernández (1997) analysed Spanish banking data from 1977 to 1985 the Spanish banking sector suffered its worst—at the time—crisis in its history, affecting 52% of the 110 banks operating at the beginning of the period. They used a data set consisting of 9 balance sheet ratios, split into a training set of 15 failed and 19 non-failed banks and a testing set of 14 failed and 18 non-failed banks. They found that Neural Networks provided the best results. De Andrés et al. (2005) applied two parametric models, Linear Discriminant Analysis (LDA) and a Logistic model along with two non-parametric models, Neural Network and Additive Fuzzy Rule-based Systems to the case of classifying Spanish commercial and industrial company profitability groups, based on a set of financial ratios. They show that non-parametric models performed better than parametric models when classifying companies into two distinct profitability groups. Callejón et al. (2013) used a Neural Network to predict the failure of European industrial companies using information for 2 years prior to bankruptcy. Moreover, Spain accounted for approximately 20% of their data set. Fernández-Gámez et al. (2016) applied a Neural Network using financial and non-financial variables on a sample of a 108 Spanish hotels which went bankrupt between 2005 and 2012, they studied the periods 1, 2 and 3 year prior to insolvency.

[4] See Chen and He (2015).

3 Data

Firm level data was collected from "Sistema de Análisis de Balances Ibéricos" (SABI), which contains balance sheet, income statement and cash flow information that firms are due to report annually. The sample period stretches from 1992 to 2016, firms with total assets of less than 25,000 EUR were removed along with some additional criteria to define a relatively homogeneous population of firms.[5] The data consists of 6057 bankrupt firms along with 58,000 non-bankrupt firms,[6] for each firm the last 4 years of available financial accounts were collected. Bankruptcy responds to a situation of definitive insolvency as the asset is less than the liability although it could also be the case of being a provisional insolvency. The status of bankruptcy is assigned when the process is declared officially. We define the bankruptcy state as 1 and the non-bankruptcy state as 0 throughout the paper.

A number of ratios which are standard in the literature on financial default were constructed. The analysis will reveal that the most relevant variables in predicting bankruptcy are the ratios Total Liabilities to Total Assets, Current Liabilities to Financial Expense, Earnings Before Interest and Tax to Financial Expense, denoted as TL.TA, CL.FinExp and EBIT.FinExp respectively, along with the variables the logarithm of Total Assets and the logarithm of Sales, denoted logTA and logSALES, respectively. A description of the ratios and variables under consideration is presented in Appendix A.

Before we describe the data, a few remarks on the Spanish economy throughout the sample period are in order. Clearly, the most salient feature of that period was the crisis that affected most of western economies by 2008, which also hit the Spanish economy. The impact in Spain was probably more severe than in other European countries in terms of GDP, unemployment and government debt. Yearly GDP growth was steadily around 5% during 1992 to 2007, whereas it fell to 1.6% between 2008 and 2016, with some recovery starting in 2015. The unemployment rate, which had been around 11% during the pre-crisis period, went above 20% in 2011, the highest in the Euro zone, up until 2016 it remained roughly at that level. Moreover, government debt, as percentage of GDP, increased dramatically, from a yearly average of 60% in the pre-crisis period to 90% in 2012 and it further increased in the following years.[7]

Figure 1 plots the breakdown of bankrupt and non-bankrupt companies by industry and by regions, in the left and right plot, respectively. By industry, the highest percentage of bankruptcy rates occurs in the construction sector, which in turn, was the third most important sector by number of firms. The construction

[5] The data was filtered by firms with Spanish national legal form in order to eliminate multinational firms with subsidiaries in Spain. Only firms with the entity type as Corporate was selected, eliminating financial companies and banks which have different balance sheets and initial capital requirements than ordinary firms. Data was removed for consolidated accounts where unconsolidated accounts exist, in order to remove the issue of double accounting. Firms were removed with no recent financial data along with Non-profit/Public authorities/State and Government-owned companies, since the decision for filing for bankruptcy is different to that of private firms.

[6] A ratio of 0.104 of bankrupt firms to non-bankrupt.

[7] Source: OECD data.

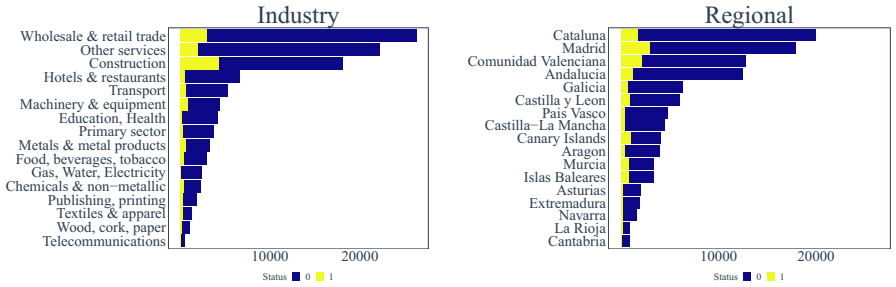

Fig. 1 Sample by sectors and regions

sector in Spain was one of the more severely affected sectors during the financial crisis. By regions, the top four, most populous regions, Madrid, Catalonia, Valencia and Andalusia all report a large number of bankruptcies over the study period, however Castilla y Leon, the Canary Islands, Murcia and the Balearic Islands all report relatively high numbers of bankruptcy rates relative to their size, which are significantly smaller compared to the top four regions in Spain. Table 5, in Appendix A, presents some basic statistics for the variables under consideration, separately for bankruptcy and non-bankruptcy firms.

4 Methodology

We propose a recently developed Machine Learning algorithm, *Extreme Gradient Boosting*, or XGBoost, developed by Chen and Guestrin (2016). Extreme Gradient Boosting can be thought of as a regularised gradient boosting model. Gradient boosting uses an ensemble learning method, which essentially combines the predictive power of several weaker models—also called trees or classifiers—in order to obtain a superior predictive model. These individual models are called base learners or weak learners and may only be slightly better than random guessing. The combination of these weak learners will yield better predictive performance than any individual base learner on its own. The effect is that the combination of weak learners can quickly fit and then potentially over-fit the training data, to correct for this, regularisation is added to the learning objective.

In order to describe XGBoost more precisely, some notation is required. We consider a set of firms, $\mathcal{I} := \{1, \ldots, I\}$, with $i \in \mathcal{I}$ being a generic firm. Each firm i has some characteristics $\mathbf{x}_i$ which imperfectly determine its bankruptcy state, $y_i \in \{0, 1\}$. We denote $y_i = 1$ if firm i is bankrupt and $y_i = 0$ otherwise. The aim is to predict y_i from $\mathbf{x}_i$ for every $i \in \mathcal{I}$. A tree[8] is as depicted in Fig. 2. It is nothing but a step function that maps a vector of characteristics into an score. Let f_k denote a specific tree, such that $f_k(\mathbf{x}_i)$ is the score, or weight, that the tree f_k assigns to any firm with characteristics $\mathbf{x}_i$. If we have a set of K trees, $\phi = \{f_1, \ldots, f_K\}$, the overall

[8] Mathematically, it is a *directed tree* in graph theory.

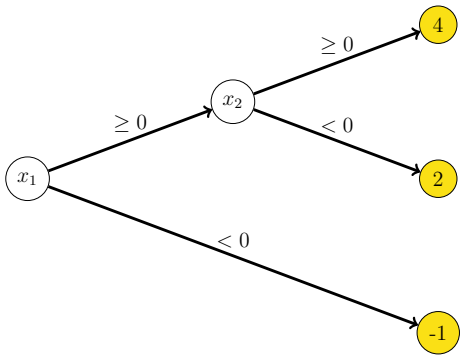

Fig. 2 Example of a tree. This tree assigns scores depending on characteristics x_1 and x_2. The scores are located in the *leaves*, or terminal nodes, in yellow. For instance, a firm with $x_1 \geq 0$ and $x_2 \geq 0$ is classified into the upper leaf, and thus it is given a score of 4. Using the notation in the text and taking $\mathbf{x} = (x_1, x_2)$, we have, for instance, $f(2,1) = 4$

score for any such firm is $\hat{y}_i := \sum_{k=1}^{K} f_k(\mathbf{x}_i)$, which constitutes the prediction for y_i based on $\mathbf{x}_i$ using ϕ.[9]

The question, of course, is how to select each element in ϕ from some function space $\mathcal{F}$ of step functions. The XGBoost algorithm selects trees from $\mathcal{F}$ as to minimize a loss function. The optimisation procedure is done recursively, such that at every iteration a new tree enters ϕ. The loss function is defined as:

$$L(\phi) = \sum_{i \in \mathcal{I}} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{1}$$

where l is a logistic loss function, and Ω is a regularisation term which penalizes the complexity of each tree in ϕ. XGBoost measures the complexity of each $f_k \in \phi$ as follows:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda ||\omega_k||^2, \tag{2}$$

where $\gamma \in (0, \infty)$ and $\lambda \in (0, 1]$ are parameters, T_k is the number of terminal nodes, or leaves, while ω_k is a vector of scores, one at each leaf, in f_k. For instance, if f_k is the tree in Fig. 2, we have $T_k = 3$ and $\omega_k = (4, 2, -1)$, thus $||\omega_k||^2 = 21$. Both γ and λ are *regularisation parameters*, while the first and second terms in (2) are L_1 and L_2 regularisation terms, respectively. Two additional comments on including complexity in the loss function are in order. First, a tree with enough number of leaves can reduce to zero the prediction errors, i.e: $y_i = \hat{y}_i$ for all $i \in \mathcal{I}$. Trivially, it suffices to use one leaf for each firm. However, by doing so we loose the capacity to identify the essential features within the characteristic space that determine bankruptcy. Thus, we do not want *too many* leaves, and the first term in (2) accounts for that. Second, the L_2 normalisation, which is the second term in (2), prevents the tree from having too high a score on any particular leaf. Equivalently, it prevents any leaf in any tree to be *too important* within ϕ. Moreover, if such a large score is allowed, it will be ingrained over multiple trees and the significance of this score will be

[9] There is a slight abuse of notation here. The scores are scalars. For our problem, it is necessary that some transformation is needed to turn the overall score into a probability of bankruptcy. Specifically, we compute that probability as $\hat{y}_i = (1 + exp(-\sum_k f_k(\mathbf{x}_i)))^{-1}$.

incorporated across the whole set of trees. Finally, a common feature to all boosted methods is that trees are recursively added to ϕ analogously to gradient descent algorithms in classical optimization. The novelty of XGBoost is that it includes the second term in (2). In what follows, the set of trees ϕ is called *a model*.

We firstly split the data into a training set and a hold-out test set, 75% and 25% respectively, then we apply 10-fold cross-validation on the training set in order to find optimal parameters. In k-fold cross-validation, the sample is partitioned into k equally sized sub-samples, each of which is called a *fold*. The model (ϕ) is trained on $k-1$ folds. Then the resulting ϕ is tested using data in the remaining fold. There are k rounds, such that each fold becomes part of the training sample $k-1$ times and the testing sample just once. The purpose of k-fold cross validation is to analyse the performance of different models on multiple in-sample test data sets, this helps avoid over-fitting. The model which performs well within k-fold cross validation should be able to be generalised onto the held-out test data in which the model is finally evaluated.

We have used the implementation of XGBoost in R[10]. The algorithm that minimizes the loss function uses some hyper-parameters which we document below. We carried out a grid search during the cross validation phase in order to locate parameter values which produce the best average performances on the in-sample test data, in terms of the AUC, AUPRC and some loss functions. The optimal parameter values are presented in Table 1, in which the notation is as in the documentation of the XGBoost package. In the table there are a number of hyper-parameters, other hyper-parameters can be optimised but were omitted. `Max Tree Depth` is the maximum depth a tree is allowed before it is forced to make a decision, which can be understood as the maximum number of nodes from the initial node to any leaf node. In theory, an uncontrolled tree can grow to the size of the number of instances in the data such that each terminal node represents a given firm, however this would most likely not generalise well onto new unseen data. To control for this we can set the maximum tree depth or terminate a tree once a node has a minimum number of observations. `Eta` controls the marginal contribution of each tree within ϕ, higher values of `Eta` would very quickly reduce the loss but also very quickly plateau after relatively few trees, smaller values allow us to obtain a lower loss, but at the cost of doing so over many more trees. `Gamma` appears in (2) and basically specifies the minimum loss reduction required to make an additional split. `Lambda` also appears in (2) and is an L2 regularisation parameter on the scores. `Sub Sample` and `Col Sample` are the percentage of randomly selected observations and columns when growing each tree.

[10] See R Core Team (2013) for the programming language and Chen et al. (2018) for the package.

Table 1 Main hyper-parameters space

Hyper-parameter	Optimal value	Other values
`Max Tree Depth`	5	(3, 5, 8)
`Eta`	0.1	(0.05, 0.1, 1)
`Gamma`	0.5	(0, 0.5, 1, 1.5)
`Lambda`	1	(1)
`Sub Sample`	1	(0.75, 1)
`Col Sample`	0.75	(0.75, 1)

5 Results

This section contains the results. They are organized around three questions, analysed in separate subsections: what is the cost of enlarging the bankruptcy forecasting horizon? How do different variables contribute to the prediction of bankruptcy? How can XGBoost be used to analyse individual case studies?

5.1 Forecasting Horizon

Let us consider a firm that switches from an active state to a bankrupt state in, say, year 2013. We can try to predict that switch with 2012 data or, being more ambitious, we can try to predict it with 2011, 2010 or even 2009 data. We label these forecasting horizons as *1–4 year prior* predictions, respectively. For firms that remain active for the whole sample period, we predict the firm's state at the last year available in the dataset using the same forecasting horizons. Of course, increasing the horizon should have a cost in terms of the quality of the prediction, measured by the number of prediction errors. The aim of this subsection is to report these costs.

A brief description on how prediction errors are usually reported in Machine Learning models is in order. A model is a classifier that assigns—or predicts—bankruptcy probabilities to each firm depending on the firm's characteristics. We have denoted $\hat{y}_i$ to the predicted probability of bankruptcy for firm i, whereas the firm's actual state is either bankrupt or active, denoted as $y_i = 1$ and $y_i = 0$, respectively.[11] In order to assign these predictions to actual states, we define a probability threshold y^*, such that we say the model predicts bankruptcy for firm i or, equivalently, the model classifies firm i as bankrupt, whenever $\hat{y}_i \geq y^*$ holds. For a given y^*, the possible outcomes are usually structured in a *confusion matrix*, which as we define has predicted states by rows and actual states by columns.

In Fig. 3 we present the confusion matrix for all 4 years of data. We set $y^* = 0.5$ as the cut-off probability threshold.[12]. The model made a significant number of correct classifications across all 4 years, correctly classifying firms as bankrupt and non-bankrupt, however it did make a number of misclassifications. The model made

[11] The standard notation is to denote 1 to the state that we positively want to identify.

[12] We set $y^* = 0.5$ since we apply a positive weighting scale to control the balance of positive and negative weights which is useful for imbalanced class distributions. Positive weight scale = *sum*(negative instances)/*sum*(positive instances)

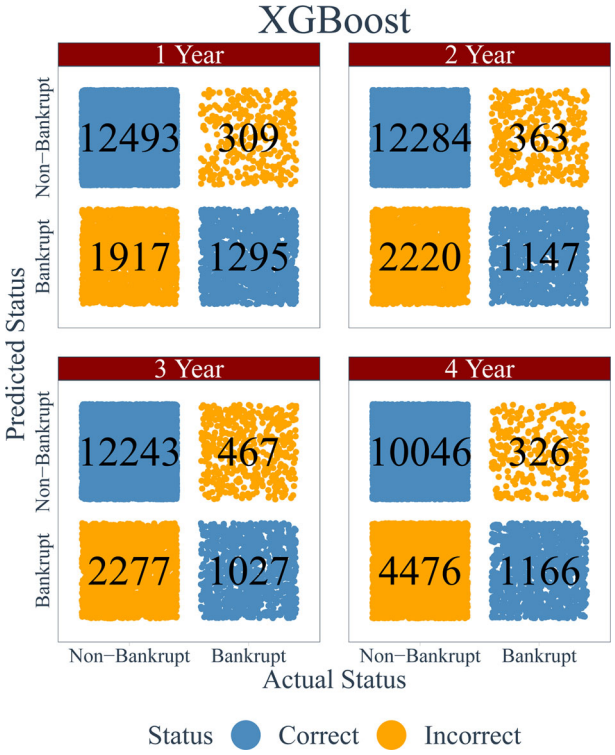

Fig. 3 XGBoost: confusion matrix graphic

Table 2 XGBoost: confusion matrix analysis

Statistics: XGBoost				
Metric	1 Year	2 Year	3 Year	4 Year
Accuracy	0.86	0.84	0.83	0.70
Sensitivity	0.81	0.76	0.69	0.78
Specificity	0.87	0.85	0.84	0.69
Precision	0.40	0.34	0.31	0.21
F1	0.54	0.47	0.43	0.33
MCC	0.51	0.43	0.38	0.29
AUC	0.84	0.80	0.77	0.74
AUPRC	0.66	0.55	0.50	0.42

a relatively large number of Type I errors (1917 for 1 year prior) and Type II errors (309 for 1 year prior) over the sample period. Table 2 reports a number of standard summary statistics for classification problems for imbalanced data Bekkar et al. (2013). Roughly, Accuracy tells us overall how often the model is correct, but it is

biased when—as in our case—a state is much more frequent than the other.[13] This motivates the use of state specific statistics: Sensitivity (also called Recall), Specificity and Precision. Sensitivity tells us when a firm is bankrupt, how often did the model correctly predict bankrupt. Specificity is similar, but for the active state. Precision tells us the proportion of correctly predicted bankrupt firms over the total number of bankrupt predictions or how many of the predicted bankrupt firms are actually bankrupt. F1 combines Precision and Recall using the Harmonic mean, thus its highest occurs whenever Precision is equal to Recall. The Matthew's Correlation Coefficient (MCC), is usually presented with a more complex formula.[14] To ease its interpretation, Table 3 shows its connection with the usual chi-squared statistic in a 2×2 contingency table. Its range is $[-1, 1]$, with -1 indicating a perfectly opposite classification and $+1$ indicating a perfectly correct classification while the center, 0, indicates perfect randomness. The statistics presented thus far are constructed under the threshold $y^* = 0.5$. The choice of this threshold depends on the relative weights between Type I and Type II errors: by increasing the threshold we become more strict when predicting bankruptcy, thus we reduce Type I errors, but at the cost of increasing Type II errors. The last two rows in Table 2 are linked to Figs. 4 and 5 presented next.

The ROC curves (Receiver Operating Characteristic), in Fig. 4, emphasizes the trade off between Type I and Type II errors by plotting Sensitivity (vertical) vs. the complementary Specificity (horizontal), also called false positive rate, defined $FPR = 1 - Specificity$. Each curve represents a different forecasting horizon.[15] In order to evaluate the overall performance of the model in a single number, taking into account each threshold decision, the Area Under the Curve or AUC is calculated which is reported in the last but one row in Table 2, with values of 1 indicating perfect classification.

The ROC curve is a popular measure when evaluating binary classification models, however caution must be made when the data structure is of an imbalanced nature. CROC (Concentrated ROC) and CC (Cost Curves) have been suggested as alternatives to ROC curves, however ROC curves are much more widely used. An alternative is the Precision-Recall curve (PRC) in Fig. 5 and is considered a more informative measure than the ROC, CROC and CC plots for binary classification with an uneven class distribution since the PRC is the only plot which changes in relation with the ratio of positives and negatives, see Saito and Rehmsmeier (2015).

[13] Consider a sample with 99 active firms and just one bankrupt firm. If we predict 100 active firms, we have a high accuracy, though we completely fail to predict our target state.

[14] That formula being:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where *True Positive*, *True Negative*, *False Positive* (Type I error) and *False Negative* (Type II error) are denoted as TP, TN, FP, FN.

[15] Each point within a curve corresponds to a value of a threshold $y^* \in [0, 1]$, the extreme points (0,0) and (1,1) are for $y^* = 1$ and $y^* = 0$, respectively. The performance of the model is better the closer the ROC curve is to the left and upper contour of the figure, while a purely random classifier would sit on the diagonal line.

Table 3 Notation: total is a summation of all four quadrants, $H_{x,y}$ is the harmonic mean of x, y, χ^2 is the chi-square statistic in a 2×2 contingency table

Accuracy	$\frac{TN+TP}{Total}$	Sensitivity, recall	$\frac{TP}{TP+FN}$	Specificity	$\frac{TN}{TN+FP}$
Precision	$\frac{TP}{TP+FP}$	F1	$H_{Precision,Recall}$	MCC	$\left(\frac{\chi^2}{Total}\right)^{1/2}$

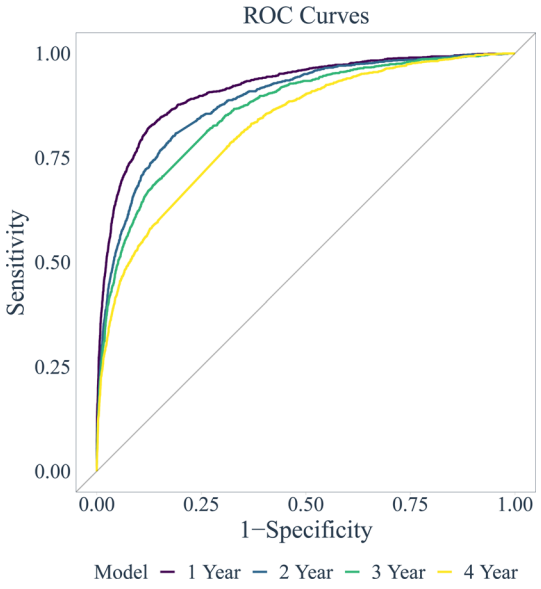

Fig. 4 Receiver operating characteristic curves

The Precision-Recall curves plots for every threshold $y^* \in [0, 1]$ the ratio between the Precision and Recall and in order to evaluate the model across all thresholds in one number the AUPRC (Area Under the Precision Recall Curve) is calculated which corresponds to the last row in Table 2. The AUPRC takes on values between 0 and 1 with values closer to 1 indicating perfect accuracy, which would be indicated by a line running horizontally along the top of Fig. 5.

Figure 6 plots the distribution of predicted probabilities for each of the years. The threshold was set to $y^* = 0.5$ so all firms above this threshold were assigned a 1 for bankrupt and all firms below were assigned a 0 for non-bankrupt. The colours indicate the true status of the firm in the test data, blue being non-bankrupt firms and red being bankrupt firms. The blue shading, above the $y^* = 0.5$ threshold corresponds to misclassified non-bankrupt firms (false positives— Type I error) and the red shading, below the $y^* = 0.5$ threshold corresponds to the misclassified bankrupt firms (false negatives—Type II error). It is evident that the density of the XGBoost model predicted probabilities is more compact at the upper and lower tails of the distribution the closer we are to the bankruptcy event, the models then become wider with spikes appearing around the $y^* = 0.5$.

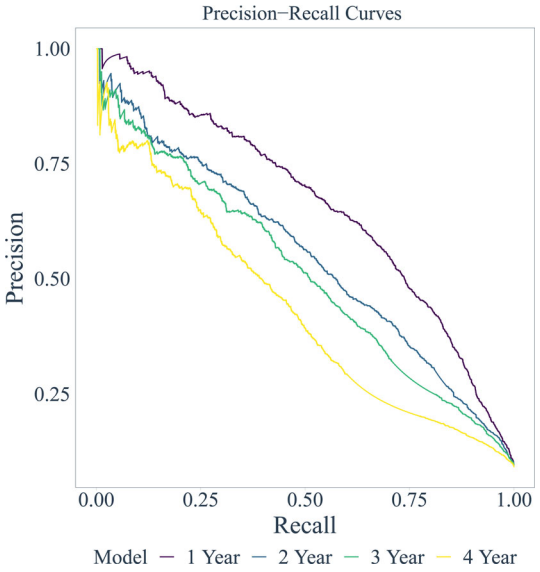

Fig. 5 Precision-recall curves

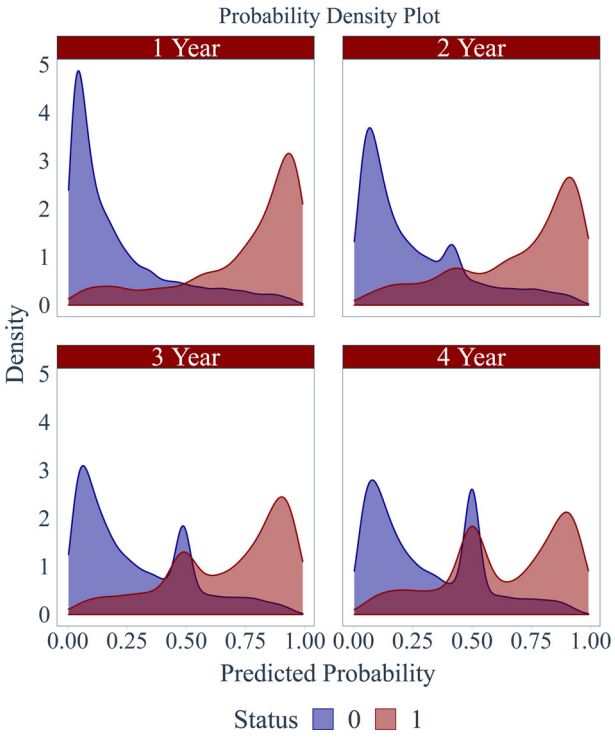

Fig. 6 Probability density plots

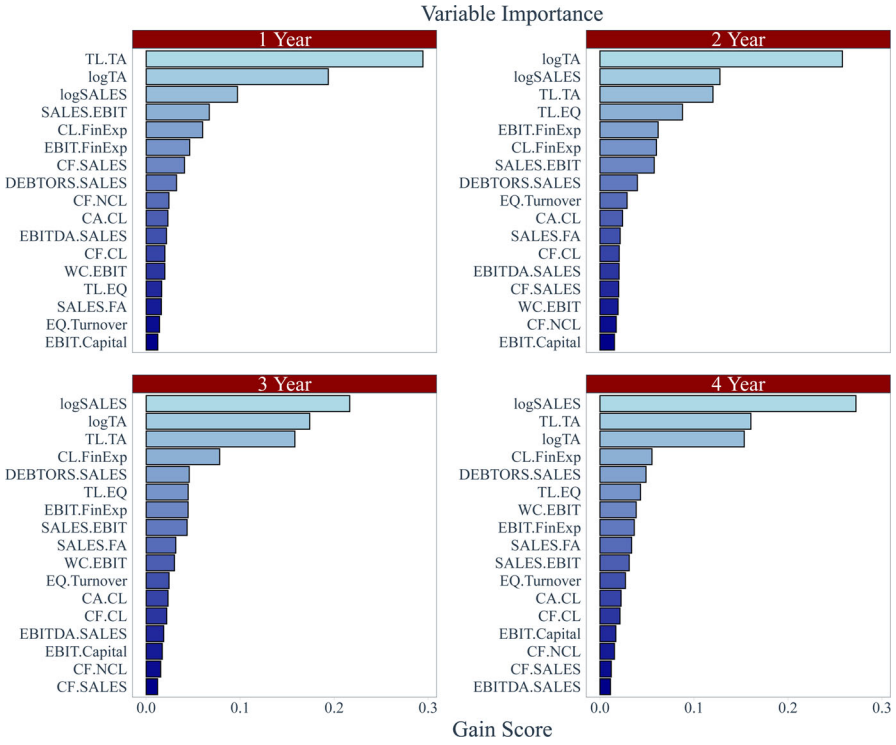

Fig. 7 Variable importance plots

5.2 Variable Importance

After decision tree based models are constructed it is possible to obtain variable importance scores. These scores indicate how useful each variable was when constructing each of the boosted decision trees in the model. It takes the contribution each variable made at each tree in the model and measures the relative contribution over all trees. Higher values indicate that the variable contributed more to the overall *Gain* over all trees. The variable importance plot takes an average of all the *Gains* each variable contributed at each tree. Figure 7 plots the variable importance plot for each of the models.

The most important variables found in the model for all 4 years of data were TL.TA, logTA, logSALES, CL.FinExp and EBIT.FinExp. That is, these variables contributed the most to the overall model prediction relative to other variables. The model found that a leverage ratio (TL.TA), size factors (logTA) and (logSALES), a debt to interest expense ratio (CL.FinExp) and an interest coverage ratio (EBIT.FinExp) were among the determining factors when making a decision on bankruptcy.

It is important to remark that Fig. 7 reports the *aggregated and unconditional* impact of each variable on the prediction of bankruptcy. There are two basic differences in measuring this impact with Machine Learning models with respect to

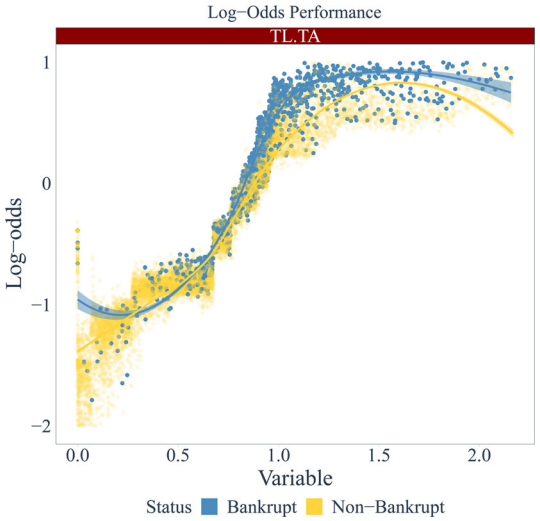

Fig. 8 How log-odds scores are affected by changing variable values

measures provided by estimates in linear models. First, in our model, the effect of each variable might be non-monotonic. Second, in our model the marginal contribution to the score of a given variable will generally differ across firms, as this marginal contribution depends on other characteristics of the firms under consideration. Figure 8 illustrates how the log odds scores changes with respect to Total Liability to Total Assets (TL.TA). The non-monotonicity of the ratio Total Liability to Total Assets (TL.TA) shows that as the ratio approaches 0.75 and above, the log-odds score goes from being negative to positive. That is, low values of TL.TA had a negative impact on the prediction of bankruptcy, i.e. most of these companies assets are financed through equity. Higher values of the same ratio suggest most of the companies assets are financed through debt and are therefore given positive log-odds scores, thus positively contributing to the prediction of bankruptcy. The *blue* points indicate a bankrupt firm and *yellow* points a non-bankrupt firm. Many of the *blue* dots appear at the upper end of the TL.TA ratio.[16]

Figure 12, in Appendix B, shows the analogous plots for all variables and ratios under consideration. Some additional comments on other ratios are in order. The current ratio (CA.CL) shows a lot of dispersion when companies have a CA.CL ratio around 0 with the variable contributing negative log-odds scores (~ -0.8) and positive log-odds scores ($\sim +0.5$). Moreover, when the ratio begins to grow greater than 1.0 the log-odds scores begin to fall and thus firms with these ratios have a negative contribution to the prediction of bankruptcy. This is intuitive since ratios greater than 1.0 is a desirable situation to be in since it means the Current Assets > Current Liabilities and therefore firms can more readably pay off its short-term debt obligations with its short-term assets. The EBITDA.SALES is an

[16] Note: The figure is dominated by *yellow* dots due to the greater number of non-bankrupt firms in the data. The data for this plot were filtered in order to remove the top and bottom 10% of extreme outliers which distorts the graphics axes and renders the plots unreadable, thus the reader should be mindful that there are points which lie outside of the plot region.

operating profitability and cash flow ratio indicating the percentage of earnings remaining once operating expenses have been accounted for. Values equal to 1 highlight that a company has no interest, no taxes, no depreciation and no amortisation. Thus, the ratio shows the percentage of revenue left once the company has paid its operating expenses and therefore values closer to 1 suggest a healthier firm. The model assigns negative log-odds scores as the ratio tends to 1 and positive log-odds scores when the ratio tends to 0. Moreover the model finds that ratios $> \sim +0.25$ have a negative impact when predicting bankruptcy and ratios $< \sim +0.20$ have a positive impact when predicting bankruptcy. The model is able to learn a number of interesting characteristics which are used by analysts and have been known in the literature for a number of decades.

5.3 Case Studies

A decision tree is interpretable but not very good at prediction, Extreme Gradient Boosting (an ensemble of decision trees) is very good at prediction, however trying to interpret all of the individual decision trees in a model is simply not feasible. The following subsection allows us to interpret the XGBoost model. What sets this model (along with other tree models) apart, from other traditional black-box Machine Learning models is that it is possible to see how each variable contributes to the overall prediction for each observation or firm in the model. There are four possible cases, each representing a different position in the confusion matrix. In the main text we present a True Positive (TP) and a False Positive (FP) case, while the TN and FN cases are left to Fig. 13 in Appendix C.

True positive (TP). Figure 9 shows the breakdown of how a positive case (bankruptcy) was correctly predicted. Given a particular variable, shown in the x-axis, a log-odds score is calculated (displayed inside each box), the sum of the log-odds scores are added up in order to obtain a final log-odds result (displayed in the final black box) and then a logistic function is applied to the result in order to obtain a predicted probability (shown on the y-axis). The horizontal line demonstrates a $y^* = 0.5$ probability cut-off threshold previously defined. Firms above this line are classified as bankrupt and firms below this line are classified as non-bankrupt. Notice, that the final log-odds prediction score is 4.58, which is assigned a predicted probability of bankruptcy $(1 + exp(-4.58))^{-1} = 0.990$.

False positive (FP) Figure 10 shows a firm that was incorrectly predicted as bankrupt. The model incorrectly predicted that the firm would be bankrupt with a final log-odds score of 0.02 and a subsequent bankruptcy probability of $(1 + exp(-0.02))^{-1} = 0.505$, sitting just above the cut-off threshold $y^* = 0.5$. It is possible that a more informed decision can be made by financial institutions through the use of these plots as opposed to other black-box Machine Learning models which would essentially make a decision simply based on this firm lying on the wrong side of a cut-off threshold.

Table 6, in Appendix C, contains the contributions of the five main variables to the score of the firms for the four case studies. The table summarises some points that have been raised in the previous discussion. First, the relative weight of

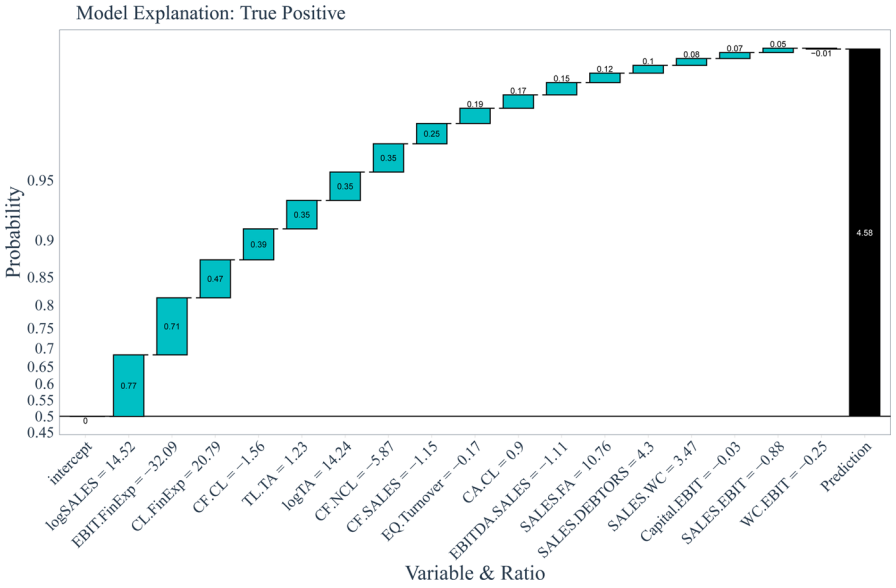

Fig. 9 True Positive: firm correctly predicted to be bankrupt case

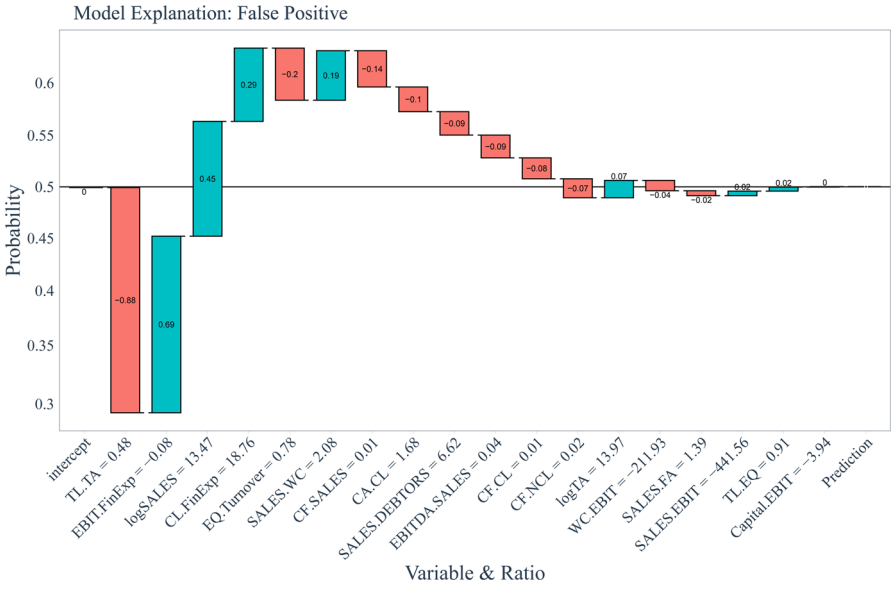

Fig. 10 False Positive: firm incorrectly predicted to be bankrupt case

different variables changes across firms. For instance, the TL.TA makes the 5th highest contribution to the TP case but it makes the highest in the FP case. Second, the TP and TN cases are very clear. Recall that a contribution to the score increases

the bankrupt probability whenever it is positive. In the TP case the table shows that all the contributions are positive. Analogously, the TN case is very clear with its main contributions being negative. In contrast, the FP and FN case have both positive and negative signs in its main contributions, and in both cases the overall probability sits very close to the cut-off threshold. Moreover, unlike in linear models, the marginal contribution of each variable is not independent of the other variables. In other words, within a row, there is no linear relation between values of the variable and its contribution to the score. The relation is not monotonic, as for instance the EBIT.FinExp row shows.

Overall these plots allow for a deeper understanding into how and why the model made a given prediction. The four cases can be linked to the log odds performance figures. Consider the TN case (in Table 6 in Appendix C) and the TL.TA variable, with a value of 0.02. It gives the second highest marginal contribution to the prediction of bankruptcy for this firm (negatively contributing to the prediction of bankruptcy by assigning a log-odds score of -1.08). This variable would sit as a yellow point in the lower left corner of the TL.TA log-odds plot in Fig. 8 where points in this space negatively contribute to bankruptcy prediction. Contrast that with the TP case with a TL.TA ratio of 1.23 in which the model assigns a log-odds score of 0.35. This would sit as a blue point in the upper right segment of the TL.TA plot in Fig. 8. Moreover, the other ratios can be interpreted and analysed in a similar manner where we are able to analyse the relationship between the log-odds scores that the model assigns and the values for each variable, Fig. 12 shows the log-odds performance for all variables in the model.

6 Comparison to Other Machine Learning models

This section compares XGBoost to other Machine Learning models. We removed all missing values, extreme values and centred and scaled the data.[17]

We compare XGBoost with: two different Neural Networks using the Keras API with TensorFlow back-end, see Chollet et al. (2015) and Abadi et al. (2015), a Support Vector Machine with a linear kernel and a radial kernel, see Cortes and Vapnik (1995) and Boser et al. (1992), a Logistic Regression, a Random Forest, see Breiman (2001), and a Light Gradient Boosting Model (LightGBM), see Ke et al. (2017).

Table 4 presents the metrics based on the confusion matrix for each of the models for 1 year prior predictions. The overall accuracy for the linear models Logistic Regression and Support Vector Machine are markedly higher than the non-linear models XGBoost, LightGBM, Random Forest and Neural Networks since the linear models make significantly less bankrupt predictions than the non-linear models and thus predicting that all companies are non-bankrupt will yield higher accuracy

[17] Scaling by $\frac{x - mean(x)}{sd(x)}$ since Neural Networks perform better when the data is centred and scaled, K-Means forms better clusters when the data is centred and scaled. Logistic models perform better with the exclusion of extreme values and all models with the exception of LightGBM and XGBoost cannot handle missing data points, thus we remove the observations with missing values.

Table 4 Comparison to other machine learning methods for 1 year prior predictions

Metric	XGBoost	Logistic	Light GBM	Shallow NN	Deep NN	R. Forest	SVM (R)	SVM (L)
Accuracy	0.87	0.91	0.89	0.91	0.91	0.92	0.90	0.91
Sensitivity	0.73	0.25	0.61	0.32	0.30	0.37	0.16	0.18
Specificity	0.88	0.99	0.92	0.98	0.98	0.99	0.99	0.99
Precision	0.42	0.66	0.47	0.60	0.62	0.79	0.66	0.71
F1	0.54	0.36	0.53	0.42	0.41	0.51	0.26	0.29
MCC	0.49	0.37	0.47	0.39	0.39	0.51	0.30	0.33
AUC	0.81	0.81	0.76	0.65	0.64	0.68	0.58	0.59

NN neural network, *SVM* support vector machine with (R) radial, (L) linear kernel

results when the data class is imbalanced, this can be seen in comparing the linear models Specificity scores to that of the non-linear models Specificity. The linear models scores significantly lower for the Sensitivity results than the non-linear models.

The XGBoost model makes significantly more bankrupt predictions than any of the other models, the LightGBM model also make a significant amount of bankrupt predictions. A cost to this is that both models make more Type 1 errors or commits more False Positives. However, all of the other models—excluding XGBoost and LightGBM—make significantly more Type 2 errors or commits more False Negatives, that is, these models predict that a company is non-bankrupt when the companies actual status was bankrupt. These models can be seen as far more costly to lending institutions than boosted models.[18]

After removing missing values and correcting for outliers in order to compare the different Machine Learning models, the XGBoost model here compares relatively similar or only marginally better to the original model which included extreme values, missing values and outliers.

We take our analysis a step further in order to understand why XGBoost and LightGBM models made significantly more bankrupt predictions than all other models. We filtered the data down to a random sample of 100 observations and plot the decisions boundary for each model. Figure 11 plots the boundaries for different models regarding two variables TL.TA (vertical axis) and logSALES (horizontal axis). The hollow circles, common to all plots, represent the 100 observations the different models are trained on. The blue and red area (in fact it is a densely

[18] We also note that a well trained Neural Network model on sufficient enough data would also be able to make more bankruptcy predictions at least on par with the two gradient boosted models presented in this paper.

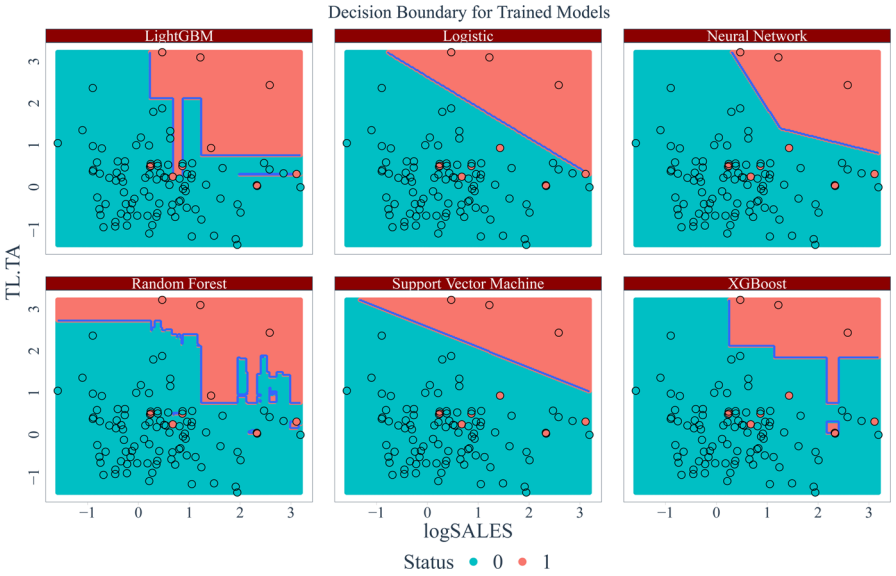

Fig. 11 Decision boundary

populated set of dots) represent synthetically created non-bankruptcy and bankruptcy states, respectively.[19]

Figure 11 complements the previous analysis by showing how different models accommodate the relationship among variables. The linear models make a crude linear classification decision boundary, ultimately misclassifying many observations. The Shallow Neural Network model is able to adapt slightly more.[20] The Random Forest (a bagging) model over fits the data and makes erratic decision boundaries. The LightGBM and XGBoost models uses boosting and adds a regularisation parameter when fitting each tree. These two models are able to make more robust and generalisable decision boundaries over linear models.

Finally, the Table 7, in Appendix D, shows the estimates of the logistic regression model. This table illustrates how Machine Learning methods can build upon classical econometrics. Consider, for instance, the estimated coefficient of the

[19] We synthetically created new data points by creating a sequence—with increments of 0.01—from the min(TL.TA) to the max(TL.TA) and also from the min(logSALES) to the max(logSALES) of the 100 randomly sampled observations. The model was trained on the 100 observations and tested on 40,000 synthetically created observations constrained by the bounds of the min/max of the TL.TA and logSALES variables. The decision boundary line was drawn at the frontier of the predictions.

[20] Trained with a Rectified Linear Unit (ReLU) and a Sigmoid activation function.

TL.TA ratio in the table, which is 0.976. According to Fig. 7, that is unconditionally the most relevant variable for 1 year prior predictions. Its marginal effect, that is, summing along all other variables, is in fact non-monotonic, as Fig. 8 shows. Furthermore, *augmenting the zoom* down to a firm-level view, the case studies presented in the previous section show that its impact changes depending on other firms characteristics, in our examples it goes from 1.23 (TP case) to 0.48 (FP), see Figs. 9 and 10, respectively. Finally, its relationship with other variables, as logSALES, shown in Fig. 11, follows highly non-linear boundaries.

Two additional comments on the comparison of XGBoost to other ML models are worth mentioning. First, XGBoost allows for individual firm-level interpretations. The United States Equal Credit Opportunity Act (ECOA)[21] prohibits creditors from discriminating against any credit applicants and mandates that credit lenders give reasoning for any credit rejection. Moreover, models such as the one proposed in this paper are useful for identifying such reasons for loan application rejections, whether that be for personal credit or corporate credit denial, banks are able to identify reasons why an application should or should not be given credit in a way other models are unable to do so. Second, XGBoost allows for a systematic or optimized treatment of complexity. It allows the practitioner to set the price of complexity in terms of accuracy within a loss function, then the algorithm chooses sequentially the loss-minimizing elements in a function space given these prices. Thus, putting things together, it is about how rather than what to predict.

We have performed some additional robustness analysis on the comparison across methods. We have computed McNemar and Chochran tests. In addition, since some models might be sensitive to a previous variable selection, we suggest replace underlined text by "we have applied different regression models for variable selection (Logistic, Stepwise Logistic, LASSO and RIDGE), then we have run again the models presented in this section after removing variables that the regression models had marked as irrelevant. Finally, we have also done some optimization on hyper-parameter values for each model. Our results show qualitatively similar conclusions to the presented here. In a nutshell, in terms of predictive capacity, XGBoost does at least as good as any other method, while it delivers very interpretable output for economic applications and it requires of little previous treatment of the data.[22]

[21] https://www.justice.gov/crt/equal-credit-opportunity-act-3

[22] We specially thank a referee for specific and very helpful suggestions on making the comparison across ML methods more robust. Some supplementary material with details on this analysis will be made available by the corresponding author upon request.

7 Conclusion

This paper applies a state of the art Machine Learning algorithm, specifically XGBoost, in order to classify firms as either being bankrupt or non-bankrupt. We use annual financial statements of 58,000 Spanish companies from 1992 to 2016 collected from "Sistema de Análisis de Balances Ibéricos", around 6000 of which went bankrupt at some point during the study period.

We find that the ratio Total Liabilities to Total Assets (TL.TA), Current Liabilities to Financial Expense (CL.FinExp), Earnings Before Interest and Tax to Financial Expense (EBIT.FinExp), the logarithm of Total Assets (logTA) and the logarithm of Sales (logSALES) were consistently ranked amongst the most important variables for all 4 years of financial accounts when determining the state of bankruptcy. That is, a leverage ratio, a debt to interest expense ratio and an interest coverage ratio along with two size factors were seen to contribute more than other variables when classifying firms as bankrupt or not. An interesting component of tree based models is that we can track the marginal contribution of each variable for each individual firm. We note that these marginal contributions can be different across firms and thus we present case studies for illustration.

We also quantify the cost of extending the forecasting horizon, that is, we aim to predict bankruptcy up to 4 years before the event itself. Our analysis yields a slight drop in performance the further out from the event we go, which is to be expected, however the model was still able to correctly classify bankrupt and non-bankrupt firms and remain consistent throughout the time horizon. On this regard, it should be noted that the sample period under consideration includes some years of a deep economic recession within the Spanish economy, XGBoost may be able to capture these non-linearities better than traditional models.

Finally, we remove all missing values and compare XGBoost to other Machine Learning models, such as Support Vector Machines, a Logistic model, Neural Networks, Random Forest and LightGBM. Generally, XGBoost significantly outperforms the Logistic model over a wide range of performance criteria. XGBoost and other Machine Learning models deliver roughly a similar capacity to systematically capture dependencies that vary across firms, particularly in populations—as in our dataset—with imbalanced response variable.

Appendix

A. Variable Description and Summary Statistics

Table 5 and the ratios description (variable, definition and ratio).

Table 5 One year summary statistics

Variable	Mean		SD		Median		Kurtosis		Skewness		Missing	
	0	1	0	1	0	1	0	1	0	1	0	1
CA.CL	92	27	4141	1395	1.40	0.97	21, 388	5813	133	76	1034	53
CF.CL	11	3	564	78	0.17	0.12	12, 553	3886	103	59	1981	170
CF.NCL	34	22	3089	1079	0.30	0.26	34, 066	4791	183	69	22, 380	1170
CF.SALES	1402	22	318, 015	1017	0.07	0.09	51, 606	3841	228	61	6392	736
CL.FinExp	3464	3050	200, 765	82, 437	52	23	33, 931	3103	179	53	14, 937	522
DEBTORS.SALES	1.90	5	66	100	0.17	0.27	7917	2232	83	46	9347	787
EBIT.Capital	9	18	92	138	1.10	1.80	11, 109	2177	91	44	1393	96
EBIT.FinExp	612	253	13, 370	4469	8	3	4645	2367	62	44	14, 681	511
EBITDA.SALES	3	20	239	917	0.08	0.11	48, 307	3902	217	61	6382	730
EQ.Turnover	559	269	110, 990	15, 484	0.36	0.23	53, 100	5058	231	71	4601	524
logSALES	13	14	1.80	1.80	13	14	5	5	−0.03	−0.68	6364	683
logTA	13	15	1.60	1.70	13	15	4	5	0.72	−0.28	12	0
SALES.EBIT	155	76	8179	1068	16	10	31, 708	3429	166	55	6389	686
SALES.FA	79	96	2710	2089	4	4	19, 967	2313	132	47	8450	798
TL.EQ	25	48	1074	799	1.80	5	36, 892	2762	178	50	18	0
TL.TA	0.77	8	1.20	250	0.67	0.96	2575	2413	38	48	18	0
WC.EBIT	92	127	4612	1963	4	5	38, 495	1603	185	36	2292	112

One year means the last year of available data for a firm, either before it turned into bankrupcty state, 1, or it did not, 0

Variable	Definition and Ratio
CA.CL	The current ratio to determine a companies ability to pay its short-term debt obligations (i.e. how a company's cash or soon to be cash items can be used to pay its short-term obligations within a year). $\text{CA.CL} = \frac{\text{Current Assets}}{\text{Current Liabilities}}$
CF.CL	An operating cash flow liquidity ratio measuring how current liabilities are being covered by the cash flows generated from operations. Ratios above 1 show that a company has generated more cash than what is required to pay it's current liabilities in the same period. Low ratios might indicate that a company needs more capital. $\text{CF.CL} = \frac{\text{Cash Flow}}{\text{Current Liabilities}}$
CF.NCL	A cash flow liquidity ratio with a longer term horizon, measuring the companies ability to pay long-term debts with the cash generated from operations in the current period. $\text{CF.NCL} = \frac{\text{Cash Flow}}{\text{Non-Current Liabilities}}$
CF.SALES	A operating ratio showing a company's ability to turn its sales into cash. Low ratios may suggest a change in the terms of sales or inefficient management of accounts receivables. $\text{CF.SALES} = \frac{\text{Cash Flow}}{\text{Sales}}$
CL.FinExp	A debt to interest payments ratio measuring the rate of interest a company is paying on its short term debt obligations. $\text{CL.FinExp} = \frac{\text{Current Liabilities}}{\text{Financial Expenses}}$
DEBTORS.SALES	A liquidity ratio measuring how much a company's sales occur on credit. A high ratio can be a negative indicator to debt providers, since it suggests that the company operates with high credit sales and therefore compromise the company's ability to pay back its interest payment obligations, since the money is tied up in credit and not cash. $\text{DEBTORS.SALES} = \frac{\text{Debtors}}{\text{Sales}}$
EBIT.Capital	Return on Capital Employed (ROCE) states the amount of capital & equity the company has used to generate its profits. $\text{EBIT.Capital} = \frac{\text{EBIT}}{\text{Capital Employed}}$
EBIT.FinExp	An interest coverage ratio which measures a company's ability to pay back interest on its outstanding debt. High ratios indicate a company can more easily pay back its interest. $\text{EBIT.FinExp} = \frac{\text{EBIT}}{\text{Financial Expenses}}$
EBITDA.SALES	EBITDA margin ratio. A profitability ratio showing the amount in which a company expects to receive after operating costs have been paid. Higher values indicate that efficient processes have kept expenses low which in turn keeps earnings high. $\text{EBITDA.SALES} = \frac{\text{EBITDA}}{\text{Sales}}$

Variable	Definition and Ratio
EQ.Turnover	A ratio to determine whether a company is creating enough turnover to justify continued operations for its shareholders. $\text{EQ.Turnover} = \frac{\text{Shareholders Equity}}{\text{Turnover}}$
logSALES	A proxy variable to measure firm size. Adjusted for 2016 inflation levels using the Spanish CPI index $\text{logSALES} = log(\text{Sales})$
logTA	A proxy variable to measure firm size. Adjusted for 2016 inflation levels using the Spanish CPI index $\text{logTA} = log(\text{Total Assets})$
SALES.EBIT	A profitability ratio indicating the percentage of a company's earnings remaining after operating expenses and before interest and tax expenses have been considered. $\text{SALES.EBIT} = \frac{\text{Sales}}{\text{EBIT}}$
SALES.FA	Fixed Asset Turnover ratio which measures a firms operating performance and efficiency. It is a measure of a company's ability to generate sales from its fixed asset investments such as, property, plant and equipment. Higher ratios indicate that a company has used its fixed asset investments to generate sales more effectively. $\text{SALES.FA} = \frac{\text{Sales}}{\text{Fixed Assets}}$
TL.EQ	A gearing ratio measuring how a company finances its operations through debt or through the shareholders own funds and reflects the ability of a businesses needing to cover outstanding debt obligations through its shareholders. $\text{TL.EQ} = \frac{\text{Total Liabilities}}{\text{Shareholder's Equity}}$
TL.TA	A leverage ratio measuring a companies ability to use its assets to pay off its liabilities. The higher the ratio the higher the degree of leverage. $\text{TL.TA} = \frac{\text{Total Liabilities}}{\text{Total Assets}}$
WC.EBIT	Working Capital (Current Assets - Current Liabilities) A short-term liquidity to earnings ratio. $\text{WC.EBIT} = \frac{\text{Working Capital}}{\text{EBIT}}$

B. Variable importance

See Figure 12.

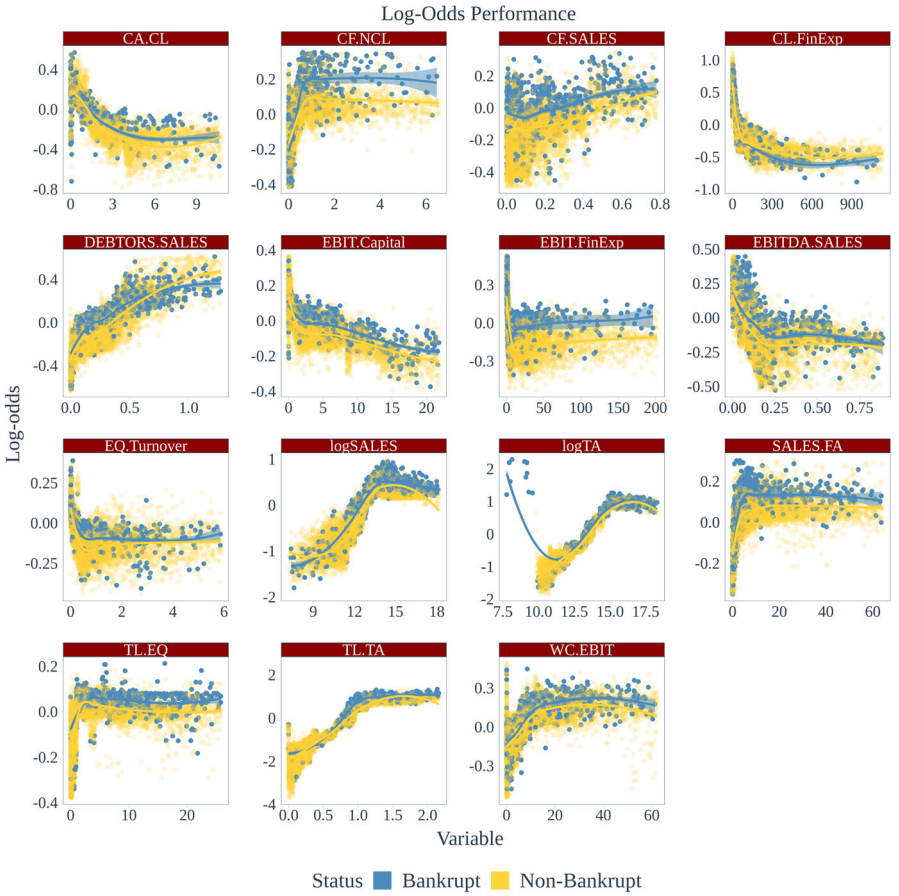

Fig. 12 How log-odds scores are affected by changing variable values

C. Case studies

The left plot of Fig. 13 shows a True Negative. The firm under consideration had significantly different financial ratios than the negative cases and thus the model assigns negative log-odds scores to each of its variables, indicating that these ratios negatively contribute to the prediction of bankruptcy. The probability of bankruptcy for this case is; $(1 + exp(-(-5.70)))^{-1} = 0.003$. Perhaps a more costly scenario would be the False Negative case, which corresponds to the right plot in Fig. 13 where the model predicts that a firm is a non-bankrupt firm and it turns out to be a bankrupt firm. This firm had a negative log-odds score of -0.09 and a subsequent probability of bankruptcy of $(1 + exp(-(-0.09)))^{-1} = 0.480$. Given the fact that it sits a little below the threshold of y^* this firm was incorrectly classified as non-bankrupt (see Table 6 for feature values of case studies in Figs. 9, 10 and 13).

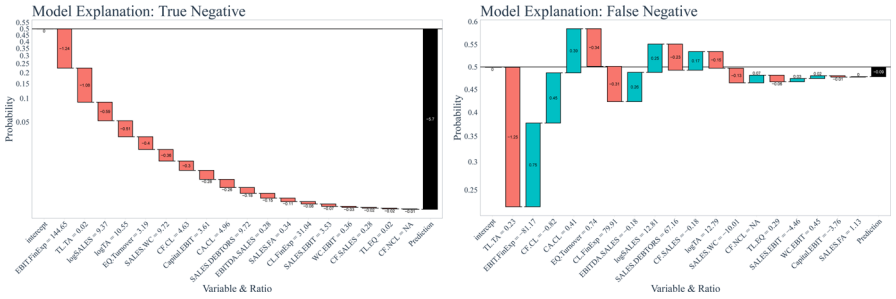

Fig. 13 Two case studies: a True Negative (left) and a False Negative (right)

Table 6 Summary of contributions of TL.TA, logTA, logSALES, CL.FinExp, EBIT.FinExp for the different case studies. Cases are in columns while variables are in rows. For instance, the cell corresponding to logSALES and the TP case shows, for that firm, that variable makes the 1st (highest in absolute value) contribution to the score with 0.77, and the actual value of that variable for that firm is 14.52. A contribution to the score increases the bankruptcy probability whenever it is positive. Predicted probabilities are in the bottom line. The cut-off threshold value is 0.5

Figure	9	13 (left)	10	13 (right)
Predicted actual	TP	TN	FP (Err Type I)	FN (Err Type II)
	Bankrupt	Active	Bankrupt	Active
	Bankrupt	Active	Active	Bankrupt
TL.TA	5th (0.35)	2nd (−1.08)	1st (−0.88)	1st (−1.25)
	1.23	0.02	0.48	0.23
logTA	6th (0.35)	4th (−0.51)	13th (0.07)	11th (−0.15)
	14.24	10.55	13.97	12.79
logSALES	1st (0.77)	3rd (−0.59)	3rd (0.45)	8th (0.25)
	14.52	9.37	13.47	12.81
CL.FinExp	3rd (0.47)	13th (−0.08)	4th (0.29)	6th (−0.31)
	20.79	31.04	18.76	79.91
EBIT.FinExp	2nd (0.71)	1st (−1.24)	2nd (0.69)	2nd (0.75)
	−32.09	144.65	−0.08	−81.17
Prob	0.990	0.003	0.505	0.480

D. Additional Information on Other ML Methods

See Table 7.

Table 7 Logistic regression results

	Dependent variable			
	Binary status of bankruptcy			
	1 Year	2 Year	3 Year	4 Year
TL.TA	0.976***	0.606***	0.463***	0.428***
	(0.059)	(0.060)	(0.060)	(0.064)
CA.CL	−0.253***	−0.053	−0.073	−0.016
	(0.087)	(0.070)	(0.065)	(0.065)
TL.EQ	0.257***	0.208***	0.209***	0.188***
	(0.037)	(0.035)	(0.035)	(0.037)
EBIT.Capital	−0.096***	−0.046	−0.053*	−0.014
	(0.035)	(0.032)	(0.032)	(0.032)
WC.EBIT	0.385***	0.331***	0.401***	0.387***
	(0.055)	(0.049)	(0.046)	(0.049)
EBIT.FinExp	0.512***	0.445***	0.285**	−0.304**
	(0.123)	(0.117)	(0.121)	(0.145)
SALES.EBIT	−0.631***	−0.434***	−0.435***	−0.512***
	(0.072)	(0.059)	(0.057)	(0.059)
CL.FinExp	−1.870***	−1.791***	−1.237***	−0.413***
	(0.201)	(0.191)	(0.158)	(0.123)
EQ.Turnover	−0.250**	−0.244***	−0.140*	−0.198**
	(0.098)	(0.084)	(0.080)	(0.090)
CF.NCL	0.106**	−0.094*	−0.059	0.014
	(0.047)	(0.051)	(0.043)	(0.042)
logTA	0.269*	0.111	−0.101	−0.460***
	(0.143)	(0.137)	(0.140)	(0.153)
logSALES	0.992***	1.079***	1.287***	1.680***
	(0.157)	(0.146)	(0.148)	(0.158)
CF.CL	−0.668***	−0.915***	−0.649***	−0.473***
	(0.097)	(0.102)	(0.091)	(0.087)
SALES.FA	−0.016	−0.029	0.051	−0.012
	(0.056)	(0.047)	(0.043)	(0.045)
CF.SALES	0.817***	0.595***	0.537***	−0.175
	(0.101)	(0.097)	(0.099)	(0.114)
EBITDA.SALES	−0.386***	−0.219**	−0.188*	0.459***
	(0.103)	(0.094)	(0.098)	(0.106)
DEBTORS.SALES	0.248***	0.346***	0.312***	0.233***
	(0.040)	(0.037)	(0.036)	(0.038)
Constant	−4.044***	−3.610***	−3.235***	−3.148***
	(0.092)	(0.083)	(0.071)	(0.066)
Observations	11,794	11,355	10,920	10,483

Table 7 continued

	Dependent variable			
	Binary status of bankruptcy			
	1 Year	2 Year	3 Year	4 Year
Log likelihood	−2743.099	−3250.657	−3353.721	−3300.723
Akaike Inf. Crit.	5522.198	6537.314	6743.442	6637.446

$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

References

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., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X., (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

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

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.

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

Beaver, W. H. (1966). Financial ratios as predictors of failure. *Journal of Accounting Research*, *4*, 71–111.

Begley, J., Ming, J., & Watts, S. (1996). Bankruptcy classification errors in the 1980s: An empirical analysis of Altman's and Ohlson's models. *Review of Accounting Studies*, *1*(4), 267–284.

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications* , *3*(10), 27–38.

Bell, T. B., Ribar, G. S., & Verchio, J., (1990). Neural nets versus logistic regression: a comparison of each model's ability to predict commercial bank failures. In *Auditing symposium on auditing problems* (pp. 29–53).

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Callejón, A., Casado, A. M., Fernández, M. A., & Peláez, J. I. (2013). A system of insolvency prediction for industrial companies using a financial alternative model with neural networks. *International Journal of Computational Intelligence Systems*, *6*(1), 29–37.

Carmona, P., Climent, F., & Momparler, A. (2018). Predicting failure in the us banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, *61*, 304–323.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794). ACM.

Chen, T., & He, T. (2015). Higgs boson discovery with boosted trees. In: *NIPS 2014 workshop on high-energy physics and machine learning* (pp. 69–80).

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., & Li, Y. (2018). xgboost: Extreme gradient boosting. R package version 0.71.2. https://CRAN.R-project.org/package=xgboost.

Chollet, F., et al. (2015). Keras. https://keras.io.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.
Creamer, G., Freund, Y. (2004). Predicting performance and quantifying corporate governance risk for Latin American adrs and banks.***Creamer, G., & Freund, Y. (2004). Predicting performance and quantifying corporate governance risk for Latin American ADRs and banks. In *Proceedings of the second IASTED international conference on financial engineering and applications* (pp. 91–101). Cambridge, MA, 8–10 November 2004.
De Andrés, J., Landajo, M., & Lorca, P. (2005). Forecasting business profitability by using classification techniques: A comparative analysis based on a Spanish case. *European Journal of Operational Research*, *167*(2), 518–542.
Etheridge, H., & Sriram, R. (1996). A neural network approach to financial distress analysis. *Advances in Accounting Information Systems*, *4*, 201–222.
Fernández-Gámez, M. Á., Cisneros-Ruiz, A. J., & Callejón-Gil, Á. (2016). Applying a probabilistic neural network to hotel bankruptcy prediction. *Tourism & Management Studies*, *12*(1), 40–52.
Fitzpatrick, P. (1932). A comparison of ratios of successful industrial enterprises with those of failed companies, certified public accountants. *Certified Public Accountant*, *6*, 727–731.
Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *ICML* (Vol. 96, pp. 148–156). Citeseer.
Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*, 1189–1232.
Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.
Frydman, H., Altman, E. I., & Kao, D.-L. (1985). Introducing recursive partitioning for financial classification: The case of financial distress. *The Journal of Finance*, *40*(1), 269–291.
Hansen, J. V., & Messier, W. F, Jr. (1991). Artificial neural networks: Foundations and application to a decision problem. *Expert Systems with Applications*, *3*(1), 135–141.
Hernandez-Tinoco, M., & Wilson, N. (2013). Financial distress and bankruptcy prediction among listed companies using accounting, market and macroeconomic variables. *International Review of Financial Analysis*, *30*, 394–419.
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146–3154).
Martin, D. (1977). Early warning of bank failure: A logit regression approach. *Journal of Banking & Finance*, *1*(3), 249–276.
Odom, M. D., & Sharda, R. (1990). A neural network model for bankruptcy prediction. In *1990 IJCNN international joint conference on neural networks, 1990* (pp. 163–168). IEEE.
Ohlson, J. A. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, *18*, 109–131.
Olmeda, I., & Fernández, E. (1997). Hybrid classifiers for financial multicriteria decision making: The case of bankruptcy prediction. *Computational Economics*, *10*(4), 317–335.
Pompe, P., & Feelders, A. (1997). Using machine learning, neural networks, and statistics to predict corporate bankruptcy. *Computer-Aided Civil and Infrastructure Engineering*, *12*(4), 267–276.
R Core Team. (2013). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.
Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3), e0118432.
Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of neural networks: The case of bank failure predictions. *Management Science*, *38*(7), 926–947.
Theodossiou, P. T. (1993). Predicting shifts in the mean of a multivariate time series process: An application in predicting business failures. *Journal of the American Statistical Association*, *88*(422), 441–449.
West, R. C. (1985). A factor-analytic approach to bank condition. *Journal of Banking & Finance*, *9*(2), 253–266.
Wilson, R. L., & Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems*, *11*(5), 545–557.
Winakor, A., & Smith, R. (1935). Changes in the financial structure of unsuccessful industrial corporations. *Bulletin*, *51*, 44.
Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, *78*, 225–241.

Zhao, D., Huang, C., Wei, Y., Yu, F., Wang, M., & Chen, H. (2017). An effective computational model for bankruptcy prediction using kernel extreme learning machine approach. *Computational Economics*, *49*(2), 325–341.

Zhou, L., & Lai, K. K. (2017). Adaboost models for corporate bankruptcy prediction with missing data. *Computational Economics*, *50*(1), 69–94.

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, *58*, 93–101.

Zmijewski, M. E. (1984). Methodological issues related to the estimation of financial distress prediction models. *Journal of Accounting Research*, *22*, 59–82.