International Review of Economics and Finance 61 (2019) 304–323

Contents lists available at ScienceDirect

International Review of Economics and Finance

journal homepage: www.elsevier.com/locate/iref

Predicting failure in the U.S. banking sector: An extreme gradient boosting approach

Pedro Carmona [a], Francisco Climent [b,*], Alexandre Momparler [c]

[a] *Department of Accounting, University of Valencia, Spain*
[b] *Department of Financial Economics, University of Valencia, Spain*
[c] *Department of Corporate Finance, University of Valencia, Spain*

ARTICLE INFO

Keywords:
Bank failure prediction
Bank failure prevention
Bank financial distress
Machine learning
Extreme gradient boosting
XGBoost

ABSTRACT

Banks play a central role in developed economies. Consequently, systemic banking crises destabilize financial markets and hamper global economic growth. In this study, extreme gradient boosting was used to predict bank failure in the U.S. banking sector. Key variables were identified to anticipate and prevent bank defaults. The data, which spanned the period 2001 to 2015, consisted of annual series of 30 financial ratios for 156 U.S. national commercial banks. Identifying leading indicators of bank failure is vital to help regulators and bank managers act swiftly before distressed financial institutions reach the point of no return. The findings indicate that lower values for retained earnings to average equity, pretax return on assets, and total risk-based capital ratio are associated with a higher risk of bank failure. In addition, an exceedingly high yield on earning assets increases the chance of bank financial distress.

1. Introduction

Banks are an essential part of the economy. They enable transactions and channel credit to households and businesses, two key functions for prosperous economic activity. Banks collect funds from private individuals and businesses, perform maturity transformation, and provide loans to households and businesses. They thereby enable the allocation of savings to investments. The banking industry is based on trust and reputation. If confidence in the banking system is lost, depositors and other short-term debt holders rapidly withdraw their funds. This loss of confidence can lead to bankruptcy because no bank has sufficient liquidity to cover all its short-term liabilities. Bank failures can jeopardize financial stability, especially when they lead to a loss of depositor confidence in other banks (European Commission, 2012). In this paper, we test the performance of a machine learning method, Extreme Gradient Boosting, which can improve the accuracy of bank failure prediction and help prevent bank financial distress.

The most recent financial crisis caused the largest number of U.S. bank failures after the savings and loan crisis of the late 1980s and early 1990s. Since the beginning of the financial crisis in 2008, 520 U.S. banks have failed (FDIC, 2016). The rise in bank failures was mostly due to the collapse of the U.S. housing market (Trussel & Johnson, 2012). Falling home prices destroyed the value of securities tied to home loans, which forced banks to write down assets on their balance sheets. The combined effect of falling home prices and losses in the stock and bond markets resulted in historic declines in household wealth.

Although certain features of the recent financial crisis make it unique, many of its underlying causes and outcomes are common to

[*] Corresponding author. Departamento de Economía Financiera y Actuarial, Facultad Economía, Campus dels Tarongers, Av. dels tarongers, S/N, 46022 Valencia, Spain..

E-mail addresses: pedro.carmona@uv.es (P. Carmona), f.jose.climent@uv.es (F. Climent), alexandre.momparler@uv.es (A. Momparler).

http://dx.doi.org/10.1016/j.iref.2018.03.008
Received 27 September 2017; Received in revised form 27 February 2018; Accepted 22 March 2018
Available online 26 March 2018

past crises (Dell'Ariccia, Igan & Laeven., 2012). It is therefore useful to develop models that are capable of foreseeing bank financial distress and that are not excessively tied to the economic and financial circumstances surrounding a specific crisis.

A number of early warning systems and leading indicator models have been developed to prevent bank failure. Yet the breadth and depth of the recent financial crisis indicates that these methods must improve if they are to serve as a useful tool for regulators and managers of financial institutions. The following paragraphs describe the early warning models that are currently used by U.S. regulators.

The Federal Deposit Insurance Corporation (FDIC) preserves and promotes public confidence in the U.S. financial system by insuring deposits in banks. The FDIC's current early warning model is the Statistical CAMELS Off-Site Rating (SCOR). Based on stepwise ordered logit analysis, SCOR uses quarterly financial data to identify banks that are expected to experience a downgrade at the next on-site examination. According to Collier, Forbush, Nuxoll, and O'Keefe (2003), the SCOR model plays an important role in the FDIC supervisory process, which involves offsite monitoring, allocating resources for examinations, and tracking industry trends. Tests of the accuracy of the FDIC model (Collier et al., 2003) indicate that approximately two-thirds of institutions that were actually downgraded were not identified by the model, and approximately two-thirds of institutions that the model did identify for downgrades were not downgraded. The model's accuracy relies on the accuracy of its financial data inputs.

The Federal Reserve promotes the safety and soundness of U.S. financial institutions, as well as their compliance with all applicable laws and regulations. Like SCOR, the Federal Reserve's System to Estimate Examination Ratings (SEER) model uses stepwise multinomial logit analysis to predict the probability that a bank will be assigned each of five possible ratings. The SEER rating is then calculated as the sum of the five rating levels multiplied by their corresponding probabilities. The variables that consistently remain statistically significant in each quarter are proxies for, capital adequacy, asset quality, earnings performance, and liquidity among others (Jagtiani, Kolari, Lemieux, & Shin, 2003). Macroeconomic variables such as unemployment, income per capita, and permits per capita are usually not statistically significant.

In a comprehensive study, Jones, Johnstone, and Wilson (2015) compared the predictive performance of classifiers ranging from conventional classifiers (such as logit/probit and linear discriminant analysis) to fully nonlinear classifiers, including neural networks, support vector machines, and recent statistical learning techniques such as generalized boosting, AdaBoost, and random forests. The newer classifiers outperformed all other classifiers. Exploring financial distress, Maciej Zięba, Tomczak, Tomczak (2016) compared methods based on statistical hypothesis testing, statistical modeling (e.g., generalized linear models), and recent artificial intelligence methods (e.g., neural networks, support vector machines, and decision tress). They examined the quality of various machine learning approaches that are designed to solve two-class problems. Extreme Gradient Boosting yielded the best results.

Much of the literature on bank failure prediction is based on conventional classifiers. We applied extreme gradient boosting (XGBoost), a state-of-the-art machine learning method, to predict bank failure. The results of our analysis suggest that this method is suited to the banking industry. A key assumption in machine learning is that the generation of data is a complex process (Krolikowski, 2018; Rey-Moreno & Medina-Molina, 2017). Machine learning seeks a response by observing inputs and responses and finding predominant patterns (Momparler, Carmona, & Climent, 2016). This process enhances the model's predictive capability and focuses on what is being predicted and how predictive accuracy should be measured.

Simplicity is a major benefit of single decision trees, but extreme gradient boosting yields a model with hundreds or even thousands of trees (Al Wakil, 2018; Capatina, Bleoju, Matos, & Vairinhos, 2017). This complexity poses a challenge when interpreting the final model. However, XGBoost does not have to be managed like a black box. In this paper, we show how XGBoost models can be summarized, evaluated, and interpreted similarly to conventional regression models. Although XGBoost models are complex, they can be summarized in ways that provide deep insight, and their predictive power is greater than most conventional methods.

XGBoost is an optimized distributed gradient boosting library. XGBoost was created by Tianqi Chen, a PhD Student working on a research project at the University of Washington (http://dmlc.cs.washington.edu/xgboost.html, accessed January 2018). After winning several Machine Learning competitions, XGBoost became well known in machine learning circles. According to Chen and Guestrin (2016), XGBoost is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. The GitHub repository contains all the information about this new algorithm (https://github.com/dmlc/xgboost, accessed January 2018).

This paper contributes to the literature on bank failure prediction in several ways. The extensive sampling period (2001–2015) spans the failures of all U.S. national banks that failed before, during, and after the most recent financial crisis. In addition, the application of a new method based on XGBoost represents an advance in the use of new techniques to predict U.S. bank failures. Finally, a balanced sample of banks was carefully selected to ensure that each failed bank was paired with a solvent bank of a similar size in terms of total assets.

The remainder of the paper is organized as follows: Section 2 presents a review of the literature on bank failures. Section 3 describes the data. Section 4 discusses the method. Section 5 presents the results of the analysis. Section 6 offers the main conclusions and managerial implications of the paper.

2. Literature review

In an early study on bank financial distress, Meyer and Pifer (1970) developed a model to analyze bank failure by matching each failed bank with a comparable solvent bank. They thus identified financial variables that can potentially indicate insolvency. They concluded that insolvency was indicated by factors such as fraud and other financial irregularities.

Boyd and Runkle (1993) linked indicators of bank failure to bank size rather than market-wide competition measures such as the Herfindahl index and concentration ratios. Arguably, the size variable is likely to correlate with market power, so the findings of these studies are at least suggestive of bank failure. They found that the probability of failure was essentially unrelated to bank size.

Lane, Looney, and Wansley (1986) compared techniques for developing failure prediction models. They used the Cox proportional hazards model to create a bank failure prediction model. This model constituted a semi-parametric technique based on survival analysis. The authors compared survival analysis with discriminant analysis, showing that survival analysis yields better results in two-year-before-failure models.

Kumar and Ravi (2007) classified failure prediction methods in two groups: statistical and intelligent techniques. Statistical techniques include logistic regression and factor analysis. Intelligent techniques include neural networks, nearest neighbor classifiers, operations research methods, and decision tree induction methods. They concluded that each standard algorithm had pros and cons.

Demyanyk and Hasan (2010) reviewed the empirical results presented in economics and operations research on bank failures. Their study outlined the methods used in previous studies, including an extensive review of intelligence techniques used in the operations literature to predict bank failure.

Cao, Wan, and Wang (2011) predicted financial distress for Chinese listed companies using an integrated model of rough set theory and support vector machines. The aim was to improve early warning methods and enhance prediction accuracy. The support vector machine was found to perform better than the model of rough set theory.

Olson, Denle, and Meng (2012) applied a variety of data mining tools to bankruptcy data to compare accuracy and number of rules. Decision trees were found to be more accurate than neural networks and support vector machines, albeit with an undesirably high number of rule nodes.

Ecer (2013) compared the ability of artificial neural networks and support vector machines to predict the failure of Turkish commercial banks. Neural networks were observed to have a slightly better predictive ability than support vector machines.

Eygi (2013) applied support vector machines to bank distress analysis using practical steps for 42 Turkish failed and nonfailed banks between 1997 and 2003. The results indicate that support vector machines are capable of extracting useful information from financial data and can be used as part of an early warning system.

Tian and Yu (2017) studied bankruptcy prediction in the international market using the Compustat Global database. They applied a variable selection method (adaptive least absolute shrinkage and selection operator, or LASSO) to select a parsimonious set of default predictor variables. For the Japanese market, three predictor variables (retained earnings/total asset, total debt/total assets, and current liabilities/sales) were selected by the adaptive LASSO method. For certain European countries, including the UK, Germany, and France, the equity ratio variable (equity/total liabilities) was consistently selected across different prediction horizons, whereas the other selected variables varied.

Ekinci and Erdal (2017) analyzed bank failure prediction for 37 commercial banks operating in Turkey between 1997 and 2001 using three common machine learning models. Logistic, J48, and voted perceptron were used as the base learners. Experimental results indicate that hybrid ensemble machine learning models outperform conventional base and ensemble models.

2.1. *The case of the EU*

Using bank-level and country-level data, Betz, Oprica, Peltonen, and Sarlin (2014) developed an early-warning model for predicting vulnerabilities that lead to distress in European banks. The key findings were that complementing bank-specific vulnerabilities with indicators for macro-financial imbalances and banking sector vulnerabilities improved model performance and yielded useful out-of-sample predictions of bank distress during the most recent financial crisis.

Other papers have used boosting for financial distress prediction. For instance, Alfaro, García, Gámez, and Elizondo (2008) compared the prediction accuracy of artificial neural networks and AdaBoost for a sample of European firms. They considered standard predicting variables such as financial ratios and qualitative variables such as firm size, activity, and legal structure. They concluded that AdaBoost decreases the generalization error by approximately 30% with respect to the neural network error. Comparing AdaBoost versus discriminant analysis, Cortés, Martínez, and Rubio (2008) presented AdaBoost as a classification technique that can be successfully used to forecast business failure. They applied AdaBoost to a sample of 1180 Spanish firms, finding that AdaBoost was more accurate than discriminant analysis. Momparler et al. (2016) applied the boosted classification tree methodology to predict failure in the EU banking sector between 2006 and 2012. They identified four key variables that can help anticipate and prevent bank financial distress.

Amendola, Restaino, and Sensini (2015) investigated the influence and effect of microeconomic indicators and firm-specific factors on different states of financial distress. The results for a sample of Italian firms for the period 2004 to 2009 support the hypothesis that the factors influencing firms' exit depend on exit routes. The findings highlight the need to distinguish between exit routes using a multiple-state approach.

Volkov, Benoit, and Van den Poel (2017) adopted a dynamic perspective of bankruptcy prediction modeling. They used financial ratios measured over multiple periods and introduced Markov-based variables for the discrimination model. The results of analysis of multiple samples of Belgian bankruptcy data indicate that using data collected from multiple periods outperforms snapshot data consisting of financial ratios measured at one time. In addition, the results imply that the inclusion of Markov for discrimination of model variables in nonensemble bankruptcy prediction models can improve classification performance.

2.2. *The case of the U.S*

Trussel and Johnson (2012) investigated the financial indicators that are associated with U.S. bank failures. They used logistic regression to weight the six financial indicators to create a composite measure of failure. They concluded that an increase in Tier 1 capital as a percentage of total assets and an increase in return on assets have the greatest influence on reducing the likelihood of failure.

Lu and Whidbee (2013) examined how charter type (national vs. state), holding company structure, and measures of bank fragility affected the likelihood of bank failure in the late 2000s financial crisis. The sample included all commercial banks in the 50 U.S. states and Washington. Lu and Whidbee (2013) estimated a series of logit regressions to identify the causes of failure and assess the role of bank-level characteristics while controlling for the economic and regulatory environment. The empirical results indicate that the failure of established institutions depended on whether the bank received bailout funds, had relatively low capital ratios, had relatively low liquidity, relied more heavily on brokered deposits, held a relatively large portfolio of real estate loans, had a relatively large proportion of nonperforming loans, and had less income diversity.

Serrano-Cinca, Fuertes-Callén, Gutiérrez-Nieto, and Cuellar-Fernández (2014) studied the bankruptcy of U.S. banks since 2009. They proposed several hypotheses on what causes failure, citing factors such as loan growth (some of them risky), specialization (focus on real estate), and a turnover-driven strategy that neglects margin. They presented and tested a structural equation model based on partial least squares path modeling (PLS-PM) and logistic regression. According to their results, five years before the crisis, failed banks had higher loan growth, higher focus on real estate loans, higher risk ratios, higher turnover, and lower margins than solvent banks did. In failed banks, the relationship between the percentage of real estate loans and risk was significant.

Serrano-Cinca and Gutierrez-Nieto (2013) used partial least squares discriminant analysis (PLS-DA) to predict the 2008 U.S. banking crisis. PLS regression was used to transform a set of correlated explanatory variables into a new set of uncorrelated variables, a suitable approach when there is multicollinearity. PLS-DA consists of PLS regression with a dichotomous dependent variable. This technique was compared to eight widely used algorithms in bankruptcy prediction. No algorithm was more accurate than any other. In terms of performance, each algorithm assigned a score to each bank and classified that bank as solvent or failed. PLS-DA results were similar to those yielded by linear discriminant analysis and support vector machines.

Lu and Whidbee (2016) examined the characteristics of 6236 U.S. commercial banks. These banks were the target of intervention in the form of bailouts or failure during the financial crisis. Considering banks for which intervention was necessary, Lu and Whidbee (2016) studied the characteristics that distinguished those that received bailout funds from those that were deemed failures. The empirical results indicate that many characteristics of bailed-out banks are similar to the characteristics of failed banks.

Cleary and Hebb (2016) used discriminant analysis to examine the failure of 132 U.S. banks between 2002 and 2009, 92% of the time successfully distinguishing between banks that failed and those that did not.

Chiaramonte, Liu, Poli, and Zhou (2016) evaluated how well Z-scores can predict bank failure. Using U.S. commercial bank data for 2004 to 2012, they found that, on average, Z-scores can predict 76% of bank failures. An additional set of bank-level and macro-level variables did not increase predictive power. They also found that the predictive power of Z-scores to predict bank defaults remains stable within the three-year forward window.

DeYoung and Torna (2013) tested whether income from nontraditional banking activities contributed to the failure of hundreds of U.S. commercial banks during the financial crisis. The authors' estimates from a multi-period logit model indicated that the probability of bank failure declined with pure fee-based nontraditional activities such as securities brokerage and insurance sales but increased with asset-based nontraditional activities such as venture capital, investment banking, and asset securitization.

Berger and Bouwman (2013) examined how capital affects bank performance (survival and market share) and how this effect varies across banking crises, market crises, and normal times in the U.S. over the past quarter century. They concluded that capital helps small banks increase their probability of survival and market share at all times (during banking crises, market crises, and normal times). Furthermore, capital enhances the performance of medium-sized and large banks primarily during banking crises.

3. Data

The cross-sectional data used in this study consisted of 30 annual financial ratio series (2001–2015) for 156 U.S. national commercial

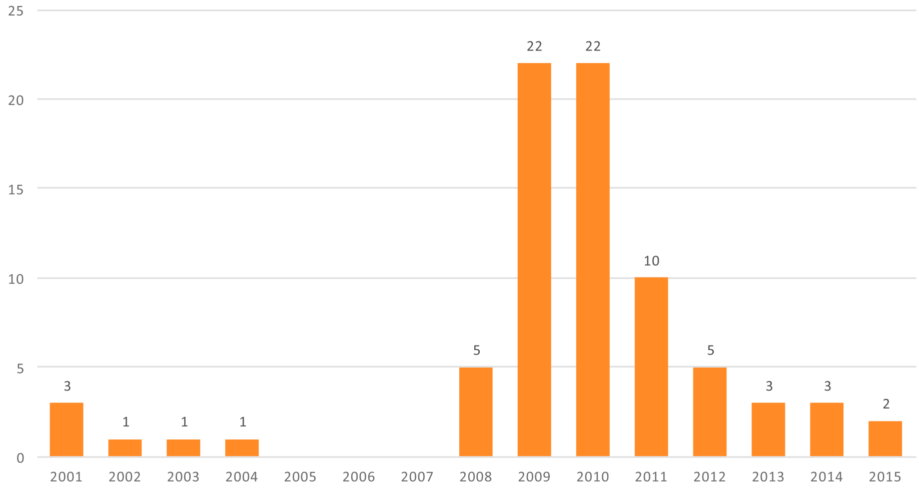


Fig. 1. Failed U.S. national banks 2001–2015.

banks: 78 failed banks (Fig. 1) matched with 78 nonfailed banks of a similar size in terms of total assets. The data source was the Federal Deposit Insurance Corporation database. We included all available financial ratios and total assets (Table 1).

4. Method

Our goal was to build a classification model to determine which variables should be monitored to anticipate bank failure. We collected accounting and financial data and applied a new machine learning algorithm to predict bank failures between 2001 and 2015. XGBoost is a recent development of gradient boosting machine tree-based models. Machine learning and data-driven approaches are becoming vital in many areas (Chen & Benesty, 2016), including spam classifiers, advertising systems, fraud detection, and anomalous event detection. Boosting is one of the approaches to building ensemble learning models. Kuhn and Johnson (2013) compiled a list of the best-known machine learning models: linear/logistic regression, k-nearest neighbors, support vector machines, tree-based models, decision trees, random forest, gradient boosting machines, and neural networks.

As reported by Chen and Benesty (2016), XGBoost is an efficient, scalable implementation of the gradient boosting framework originally proposed by Friedman (2001 and 2002). Momparler et al. (2016) applied gradient boosting methodology (GBM) to predict bank failure in the Eurozone. Although both XGBoost and GBM follow the principle of gradient boosting, XGBoost uses a more regularized model formalization to control overfitting, yielding better performance. XGBoost is similar to the GBM framework but is more efficient. It is widely used by data scientists to achieve state-of-the-art results in response to machine learning challenges (Chen & Guestrin, 2016). Evidence of its accuracy is that XGBoost is used in more than half of the winning solutions in machine learning challenges hosted at Kaggle (He, 2016). As stated on the Kaggle website (https://www.kaggle.com/about, accessed September 2016):

“Kaggle is the world's largest community of data scientists. They compete with each other to solve complex data science problems, and the top competitors are invited to work on the most interesting and sensitive business problems from some of the world's biggest companies through Masters competitions.”

Boosting is particularly useful to resolve classification problems because it minimizes overall errors by introducing supplementary models based on errors in previous iterations. Boosting methods ensemble several individual models to produce a combined model

Table 1
Explanatory variables.

1. Performance Ratios	
Key	Variable
P1	Yield on earning assets
P2	Cost of funding earning assets
P3	Net interest margin
P4	Noninterest income to assets
P5	Noninterest expense to assets
P6	Loan and lease loss provision to assets
P7	Net operating income to assets
P8	Return on assets (ROA)
P9	Pretax return on assets
P10	Return on equity (ROE)
P11	Retained earnings to average equity (YTD only)
P12	Net charge-offs to loans and leases
P13	Loan and lease loss provision to net charge-offs
P14	Earnings coverage of net loan charge-offs (X)
P15	Efficiency ratio
P16	Assets per employee (USD millions)
P17	Cash dividends to net income (YTD only)
2. Condition Ratios	
Key	Variable
C1	Earning assets to total assets
C2	Loss allowance to loans and leases
C3	Loss allowance to noncurrent loans and leases
C4	Noncurrent assets plus other real estate owned to assets
C5	Noncurrent loans to loans
C6	Net loans and leases to assets
C7	Net loans and leases to deposits
C8	Net loans and leases to core deposits
C9	Domestic deposits to total assets
C10	Equity capital to assets
C11	Core capital (leverage) ratio
C12	Tier 1 risk-based capital ratio
C13	Total risk-based capital ratio
3. Other	
Key	Variable
TA	Total assets

whose predictive power is better than the individual models when they are used alone (Chambers & Dinsmore, 2015). The tree ensemble model is a set of classification or regression trees. Because a single tree is too weak to be used in practice, the tree ensemble model, which sums the prediction of multiple trees, is applied instead. Kuhn and Johnson (2013) discussed boosting algorithms developed in the early 1990s when a number of weak classifiers were combined to produce an ensemble classifier with a superior generalized misclassification error rate. Friedman (2001) boosting machine is a method for improving model accuracy because trees depend on past fitted trees, have minimum depth, and contribute unequally to the final model. Friedman (2002) updated the boosting machine algorithm with a random sampling scheme. The new procedure was termed "stochastic gradient boosting." Stochasticity improves predictive performance, reducing the variance of the final model by using only a random subset of data to fit each new tree. In addition, boosting models allow for extreme results, nonlinearity, and missing data (when the algorithm encounters a missing value on a node then learns which path to take) and can handle different types of predictors such as categorical variables (James, Witten, Hastie, & Tibshirani, 2017).

Fitting multiple trees in boosting overcomes the biggest drawback of single-tree models: their relatively poor predictive performance (Elith, Leathwick, & Hastie, 2008). The boosting approach fits a series of small decision trees in a sequential process, with each tree built after the previous one. Each individual decision tree is adjusted to the residuals from the model and is added into the fitted function to update the residuals. Trees can be rather small, with just a few terminal nodes, which improves the model (James et al., 2017). The final model is a linear combination of hundreds or even thousands of trees. It can be thought of as a regression model where each term is a tree (Elith et al., 2008).

XGBoost is also referred to as a regularized gradient boosting technique because it enables formal control of the variable weights. Standard GBM implementation has no regularization like XGBoost. Therefore, XGBoost also helps reduce overfitting. Regularization attempts to push the weights for many variables to zero and thus performs variable selection, which plays a key role in high-dimensional problems.

XGBoost requires multiple parameters to be determined through learning from data. Controlling the best combination of parameters is necessary to optimize and improve the model. Tuning parameters usually regulate the model's complexity and are a key element for prediction. The most important parameters are discussed in the following paragraphs (Chen & Benesty, 2016).

Number of rounds or maximum number of iterations is the optimal number of rounds or trees required in an XGBoost model. It can be determined using cross-validation methods.

Maximum depth or size of a tree is the number of splits in each tree. Maximum depth controls the complexity of the boosted structure. It is used to control overfitting because higher depth allows the model to learn relationships that are highly specific to a particular sample. XGBoost makes splits up to the maximum depth specified and then starts pruning the tree backwards and removes splits beyond which there is no positive gain.

Learning rate or shrinkage is a technique first introduced by Friedman (2002). It is generally a small positive number (ranging from 0 to 1) that determines how quickly the algorithm adapts or the contribution of each tree to the growing model. The learning rate is used to prevent overfitting by making the boosting process more conservative. Doing so reduces the influence of each individual tree and leaves space for future trees to improve the model. A low value means that the model is more robust to overfitting. The intuition behind this technique is that it is better to improve a model by taking many small steps than few large steps (Natekin & Knoll, 2013).

Gamma or minimum loss reduction required to make a further partition on a leaf node of the tree. A node is split only when the resulting split gives a positive reduction in the loss function. Larger values of gamma correspond to more conservative algorithms.

Column and observation sample is a subsample ratio of variables and observations when constructing each tree. The column and observation sample denotes the fraction of variables and observations to be randomly sampled for each tree. The value ranges from 0 to 1. It prevents overfitting and speeds up computations of the algorithm.

The minimum child weight or minimum sum of instance weight needed in a child defines the minimum sum of weights of all observations required in a tree child. If the tree partition step results in a leaf node whose sum of instance weight is less than the value assigned to this parameter, then the building process will give up further partitioning. Larger minimum child weights correspond to more conservative algorithms because a large value prevents the model from learning relations that might be highly specific to the particular sample selected for a tree. Excessively high values, however, can lead to underfitting. A small value is recommended when there is an imbalanced class problem. Leaf nodes can then have smaller size groups.

Regularization or penalty term on weights. A precise choice of this parameter reduces overfitting. When regularization is equal to 0, the penalty has no effect. As the term increases, however, the impact of the shrinkage penalty grows. The regularization term controls the complexity of the model, helping avoid overfitting.

Regularization fundamentally differentiates XGBoost from other boosting models such as GBM. XGBoost regularizes the weights of variables or, equivalently, shrinks the weights toward zero. Regularization advantage is rooted in the bias-variance trade-off.[1] As regularization increases, the weights of variables decreases, leading to a substantial reduction in the variance of the predictions at the expense of an increase in bias (James et al., 2017). It has the effect of shrinking the weight of the variables toward zero. Penalty regularization reduces overfitting without producing a loss in predictive power. The regularization will tend to select a model employing simple and predictive functions. When the regularization parameter is set to zero, the objective reverts to the traditional gradient tree boosting or GBM (Chen & Benesty, 2016).

The optimal combination of XGBoost's parameters can be determined using cross-validation methods. This technique enables computation of the prediction error for an independent test sample. In k-fold cross-validation, the samples are randomly partitioned into k sets (called folds) of roughly equal size. An XGBoost model is fitted using all samples, except the first subset. Then, the prediction error

[1] Detailed explanations of the bias-variance trade-off can be found at http://scott.fortmann-roe.com/docs/BiasVariance.html(Scott Fortmann-Roe, 2012).

Table 2
Summary of descriptive statistics (failed vs. nonfailed banks).

Variable	Mean		SD		Median		Min		Max	
	NF	F	NF	F	NF	F	NF	F	NF	F
P1	0.05	0.06	0.01	0.02	0.06	0.06	0.01	0.03	0.09	0.12
P2	0.02	0.03	0.01	0.01	0.02	0.03	0.00	0.00	0.04	0.06
P3	0.04	0.03	0.01	0.01	0.04	0.03	0.00	0.01	0.06	0.09
P4	0.02	0.00	0.04	0.01	0.01	0.00	−0.01	−0.07	0.24	0.04
P5	0.04	0.05	0.03	0.03	0.03	0.04	0.01	0.01	0.23	0.24
P7	0.01	−0.04	0.01	0.04	0.01	−0.04	−0.07	−0.20	0.04	0.02
P9	0.01	−0.05	0.01	0.04	0.01	−0.05	−0.07	−0.22	0.03	0.03
P11	0.03	−0.73	0.10	0.79	0.03	−0.70	−0.48	−3.53	0.22	3.50
P12	0.01	0.03	0.01	0.03	0.00	0.03	0.00	−0.01	0.11	0.14
P13	1.40	4.50	15.38	23.01	1.21	1.31	−71.86	−5.78	77.67	200.00
P14	51.77	17.46	167.01	147.27	7.16	−0.38	−1.77	−32.31	1285.86	1265.33
P15	0.89	53.96	1.61	464.55	0.67	1.30	0.31	−3.54	14.77	4104.12
P16	89.89	4.62	761.09	2.53	3.25	4.34	0.70	1.12	6725.50	15.96
P17	0.51	0.02	0.50	0.25	0.48	0.00	0.00	−0.68	3.18	1.73
C1	0.91	0.86	0.05	0.07	0.92	0.87	0.68	0.61	0.99	0.97
C2	0.02	0.04	0.01	0.02	0.01	0.04	0.00	0.01	0.11	0.07
C3	8.69	0.78	56.93	1.52	1.22	0.42	0.21	0.11	497.50	10.00
C5	0.02	0.13	0.04	0.10	0.01	0.12	0.00	0.00	0.33	0.50
C6	0.57	0.65	0.17	0.13	0.58	0.65	0.14	0.34	0.99	0.90
C8	0.88	0.99	0.51	0.37	0.81	0.96	0.23	0.35	3.97	2.18
C9	0.81	0.88	0.12	0.08	0.84	0.89	0.25	0.61	0.92	0.99
C10	0.12	0.05	0.07	0.04	0.10	0.04	0.06	−0.05	0.58	0.15
C13	0.23	0.07	0.39	0.05	0.16	0.07	0.10	−0.17	3.50	0.25
logTA	12.63	12.37	1.54	1.49	12.48	12.30	9.73	8.97	17.11	15.94

NOTES: F: failed banks; NF: nonfailed banks that did not received government financial aid; variables are defined in Appendix 1.

of the fitted model is calculated using the first holdout fold. The same operation is repeated for each fold, and the model's performance is calculated by averaging the errors across the test sets. Cross-validation provides an estimate of the test error for each model. In our study, for parameter tuning purposes, we used 10-fold cross-validation.

In summary, we built a classification model to predict a qualitative response variable consisting of banks that failed between 2001 and 2015. The model was based on the XGBoost algorithm explained above. The model computes the probability that a bank failure occurs. In successive rounds, the algorithm seeks to fit a model that maximizes its performance for the best combination of model parameters, learning from the relationship between the response and its predictors. As one of the most widely used methods, cross-validation is applied for model selection and for choosing tuning parameter values.

All models were fitted in *R* (R Core Team, 2016) version 3.3.0 using the *XGBoost* package[2] version 0.4–3 (Chen & Benesty, 2016) and the *caret* package version 6.0–68 (Kuhn, 2016).

5. Results

Before conducting the analysis, we checked for the presence of multicollinearity problems. Pearson's correlation coefficient revealed that some ratios were highly correlated and had to be removed. These ratios were *P6, P8, P10, C4, C7, C11,* and *C12.* The subsequent analysis omitted these ratios.

The data in Table 2 highlight differences between the mean values for failed and nonfailed banks. For failed banks, the value of ratio P13 (loan and lease loss provision to net charge-offs) was notably higher, which may signal higher future risk. In addition, the value of ratio P14 (earnings coverage of net loan charge-offs) was considerably higher for nonfailed banks, suggesting that nonfailed banks can better handle loan charge-offs. Nevertheless, the standard deviation values for P14 were high for failed and nonfailed banks, indicating the great variability of ratio P14, as confirmed by the wide spread between the minimum and maximum values.

Lower values for ratio P15 (efficiency) indicate greater efficiency, hence the higher value for failed banks. The values for ratio P16 (assets per employee) imply that nonfailed banks were capable of managing a higher volume of assets per employee, which is related to the concept of efficiency in HR management. The spread between minimum and maximum values was nonetheless wide.

Finally, as this study includes a balanced sample in terms of total assets, the ratio logTA does not show meaningful differences between failed and nonfailed banks.

5.1. XGBoost parameters

This section explains how we tuned the XGBoost model parameters to fit the best model and identify structure in the data. The data should enable us to determine the best settings for the model's parameters that yield the most accurate prediction of bank failure. A good

[2] The R package XGBoost won the 2016 John M. Chambers Statistical Software Award.

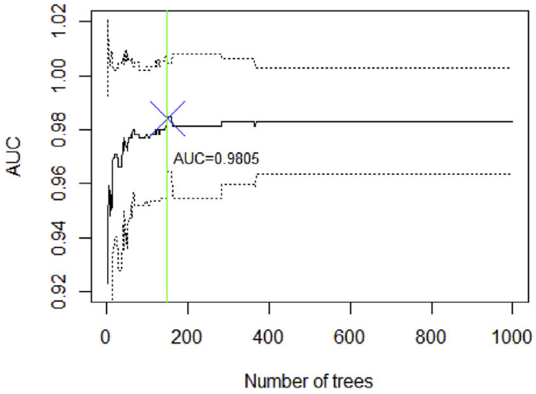


Fig. 2. Parameter tuning: Model 1.

combination of parameters is necessary to ensure the model can be generalized and avoid overfitting. Controlling these parameters is particularly important for XGBoost because its sequential model fitting allows trees to be added continuously, which can lead to overfitting. At each step, we calculated the area under the curve (AUC) as the optimization objective metric. AUC is a performance measure used for machine learning algorithms such as tree-based classification methods. The maximum value of AUC is 1. An AUC value close to 1 indicates that the model provides a good fit to the data. For binary classification problems, AUC is a better performance measure than accuracy.

We used 75% of the observations to train the XGBoost model. The remaining 25% was used as test data to validate the performance of the best-fitted model (the holdout sample). Both training and test samples consisted of paired banks. Failed banks were matched with nonfailed banks of a similar size in terms of total assets. We started training the model by choosing a high learning rate of 0.1, a maximum tree depth of 5, a gamma of 0, a subsample ratio of variables and instances of 0.8, a minimum child weight of 1, and a regularization value of 0. A small value of 1 for the minimum sum of instance weight needed in a child is recommended (Jain, 2016). All parameters were initial estimates and were later tuned. Using cross-validation, our goal was to identify the combination of key parameters described in the method section to achieve maximum AUC,[3] which was our chosen performance metric. This process yielded Model 1 (Fig. 2), which had an optimal number of 149 trees. In Fig. 2, the solid black curve is the AUC mean (loss function) for the excluded or holdout folds of the cross-validation for each number of possible trees. The dotted curves represent the interval of one standard error for the changes in predictive AUC. The blue cross shows the maximum AUC mean (0.9805). The green line is the optimal number of trees at which the AUC mean attains its maximum value. Above the optimal number of trees (149), the predictive AUC barely decreases, which might result from a problem of overfitting. It is crucial to tune the size of decision trees in XGBoost. Shallow trees are expected to perform poorly because they capture few details of the problem and are generally referred to as weak learners. Deeper trees generally capture too many details of the problem and overfit the training dataset, limiting the ability to make good predictions on new data (Brownlee, 2016).

The size of trees in XGBoost is particularly important because, for most problems, adding more trees beyond a limit does not improve the performance of the model. The reason is that the boosted tree model is constructed sequentially, where each new tree attempts to correct for the errors made by the sequence of previous trees. The model quickly reaches a point of diminishing returns (Brownlee, 2016). After tuning the optimal number of trees, we used cross-validation to tune the maximum depth of a tree, gamma, column and observation sample, minimum child weight, and regularization. To do so, the XGBoost algorithm was fed with different combinations of these parameters, and the 10-fold cross-validation determined the optimal configuration of these parameters using AUC increase as the success measure. This process yielded Model 2 (Fig. 3). According to the blue cross and green line in Fig. 3, the optimal fitted model had a holdout AUC of 0.9817, which was higher than the AUC for Model 1.

The last tuning step is to obtain a better model (Model 3) by modifying the learning rate based on the optimal parameter configuration in Model 2. A smaller learning parameter is generally preferable because it decreases the contribution of each tree, so the final model should better predict the response. After trying different values for the learning rate, the best learning rate was lower than the previous learning rate (0.05). The learning rate was again calculated with 10-fold cross-validation. Therefore, for this classification model and the selected banking data, reducing the learning rate did improve the XGBoost model's performance. Fig. 4 illustrates the new optimal parameter configuration, which yielded Model 3 (final model). According to the blue cross and green line in Fig. 4, the optimal fitted model had a holdout AUC of 0.9825, which was higher than in Model 2.

For Model 3, Fig. 5 shows the training and test curves of AUC as a performance measure. The test curve remains stable. A decrease as the number of trees or iterations increased might have denoted a problem of overfitting.

[3] To be precise, we calculated the AUC minus the standard deviation and chose the iteration with the largest value.

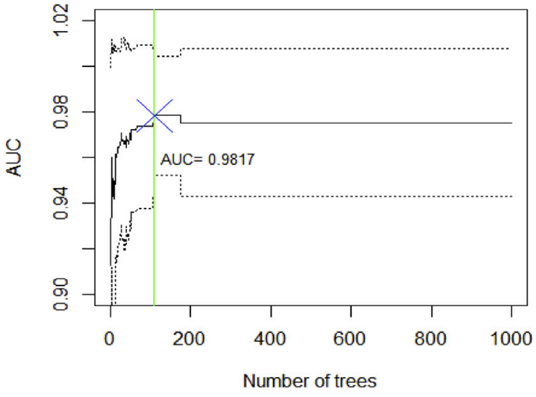


Fig. 3. Parameter tuning: Model 2.

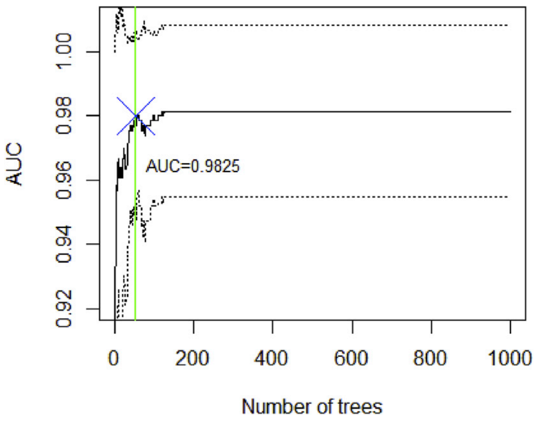


Fig. 4. Parameter tuning: Model 3 (final model).

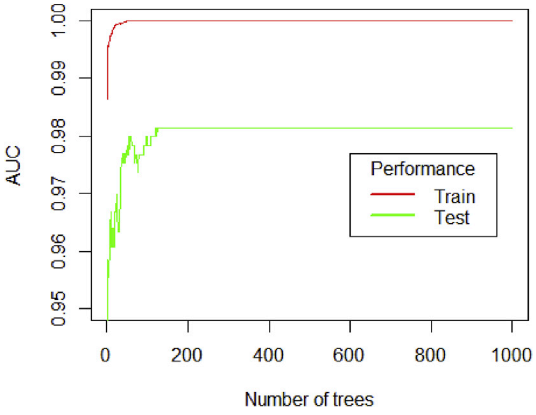


Fig. 5. Model 3: Training and test curves of AUC.

5.2. Variable importance

In Model 3, the most important variables with a high relative influence on the response variable were *P11*, *P9*, *C13,* and *P1* (Fig. 6). The plot in Fig. 6 shows the gain contribution of each variable to the model. For the boosted tree model, each gain of each feature of each tree is considered. The average per feature is then calculated to give an overview of the entire model. High percentages denote important features to predict the response variable (Chen & Benesty, 2016).

It can be difficult to understand the functional relationships between predictors and the outcome when using prediction methods like

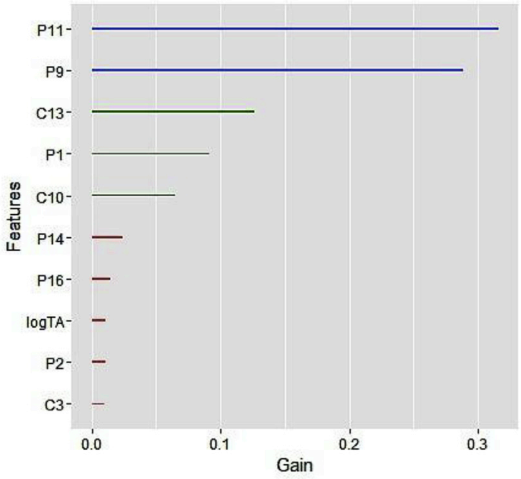


Fig. 6. Most important variables or features.

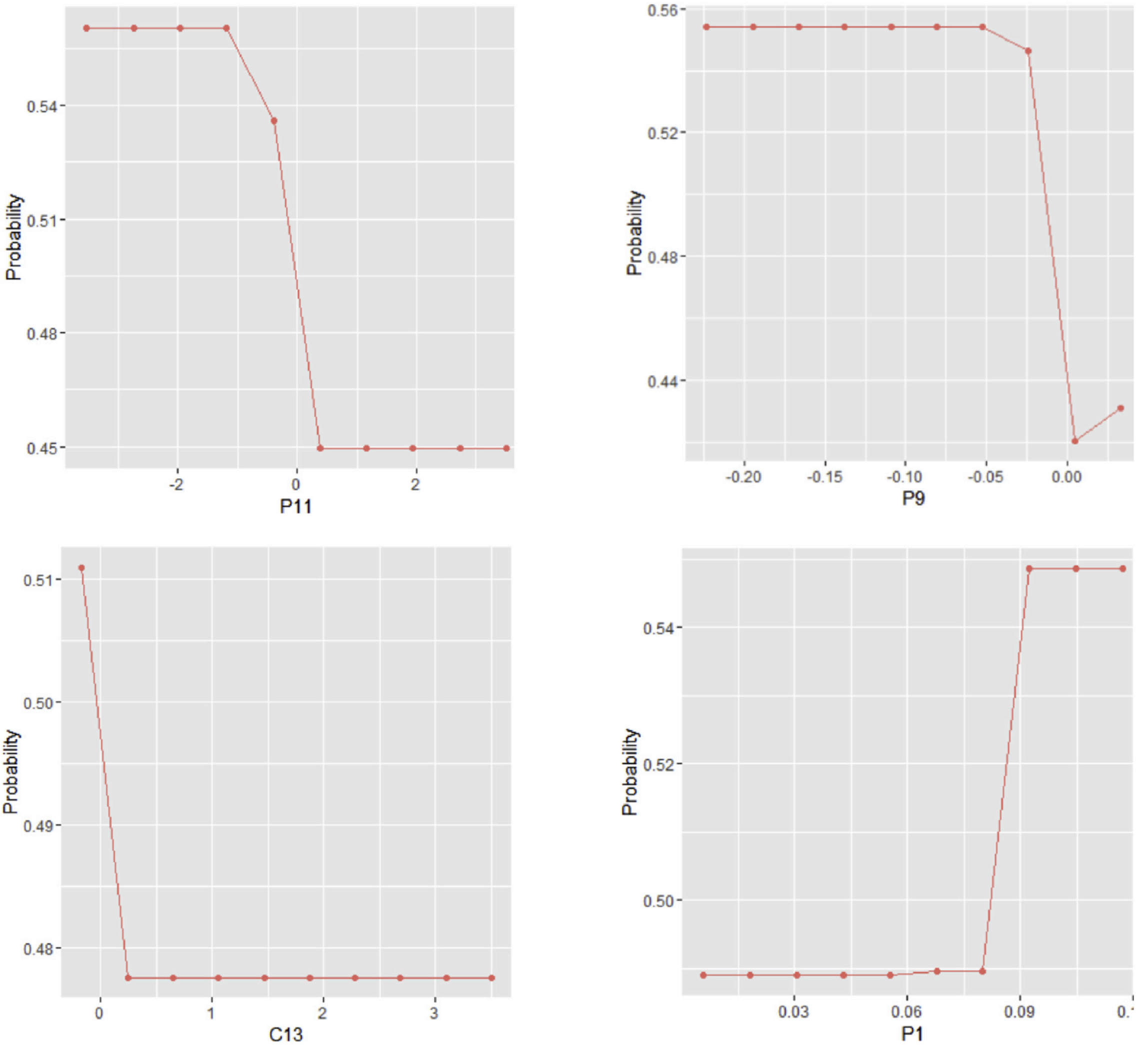
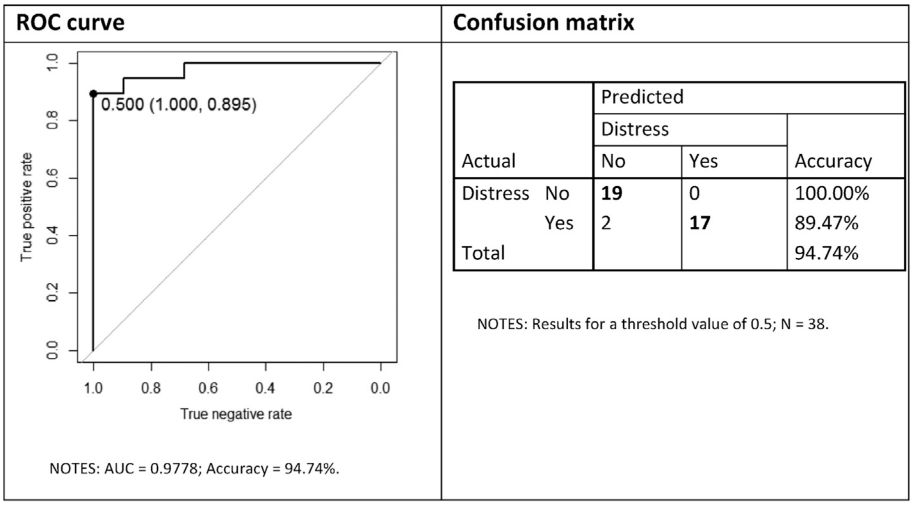


Fig. 7. Partial dependence plots for the four most influential variables.



NOTES: AUC = 0.9778; Accuracy = 94.74%.

		Predicted		
		Distress		
Actual		No	Yes	Accuracy
Distress	No	**19**	0	100.00%
	Yes	2	**17**	89.47%
Total				94.74%

NOTES: Results for a threshold value of 0.5; N = 38.

Fig. 8. Extreme gradient boosting performance for test dataset.

XGBoost. One way to investigate these relationships is with partial dependence plots. Fig. 7 shows the partial dependence plots for the four most important variables. These plots illustrate the marginal effect of a given variable on bank failure (response variable). Although these plots are not perfect representations of the relationship between predictors and the response variable, they illustrate general trends and provide a useful tool for interpretation (Natekin & Knoll, 2013).

Bank failure is best explained by four influential variables: A negative relationship was observed between bank distress and *P11*, *P9*, and *C13*, and a positive relationship was observed between bank distress and P1. The findings indicate that higher values for retained earnings to average equity, pretax return on assets, and total risk-based capital ratio are associated with a lower risk of bank failure. Conversely, an exceedingly high yield on earning assets increases the chance of bank financial distress. Therefore, failed banks' higher return on earning assets may signal an unjustified asset expansion that is based on unduly risky investments.

5.3. Model validation and model performance

Model validation is the process of assessing a model's performance. Many modern classification and regression models are highly adjustable, and they are capable of modeling complex relationships. Nonetheless, they can very easily overemphasize patterns that are not generalizable (Kuhn & Johnson, 2013). It is unadvisable to assess the predictive accuracy of a model using the same observations that were used to estimate the model. Therefore, to assess the models' predictive performance an independent dataset should be used. We randomly divided the balanced sampled of paired banks into a dataset comprising 75% of the observations and a dataset comprising 25% of the observations. The larger dataset was used to train and fit the XGBoost model, as previously described. After tuning the parameters, we identified the most important variables or features (Figs. 6 and 7). The remaining 25% of observations constituted the test or validation set. These observations were used to ensure the model was generalizable. Predictive performance should be estimated

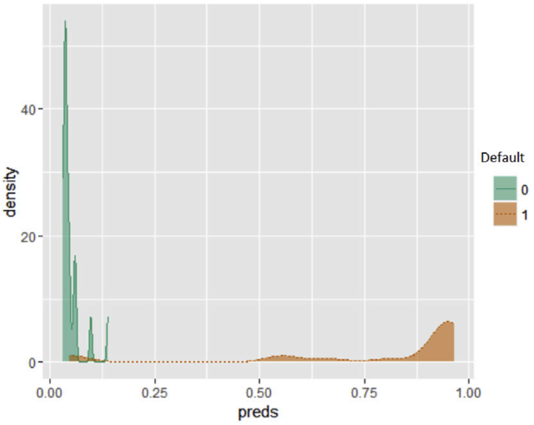


Fig. 9. Density plot of bank failure probability prediction for test dataset.

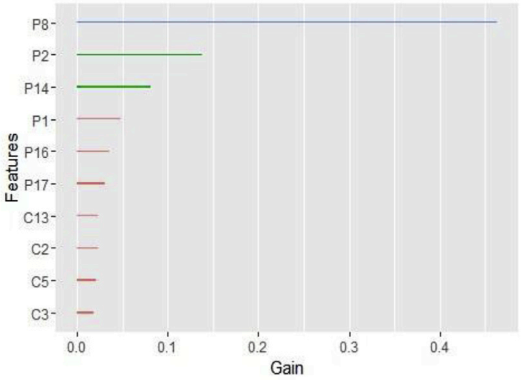
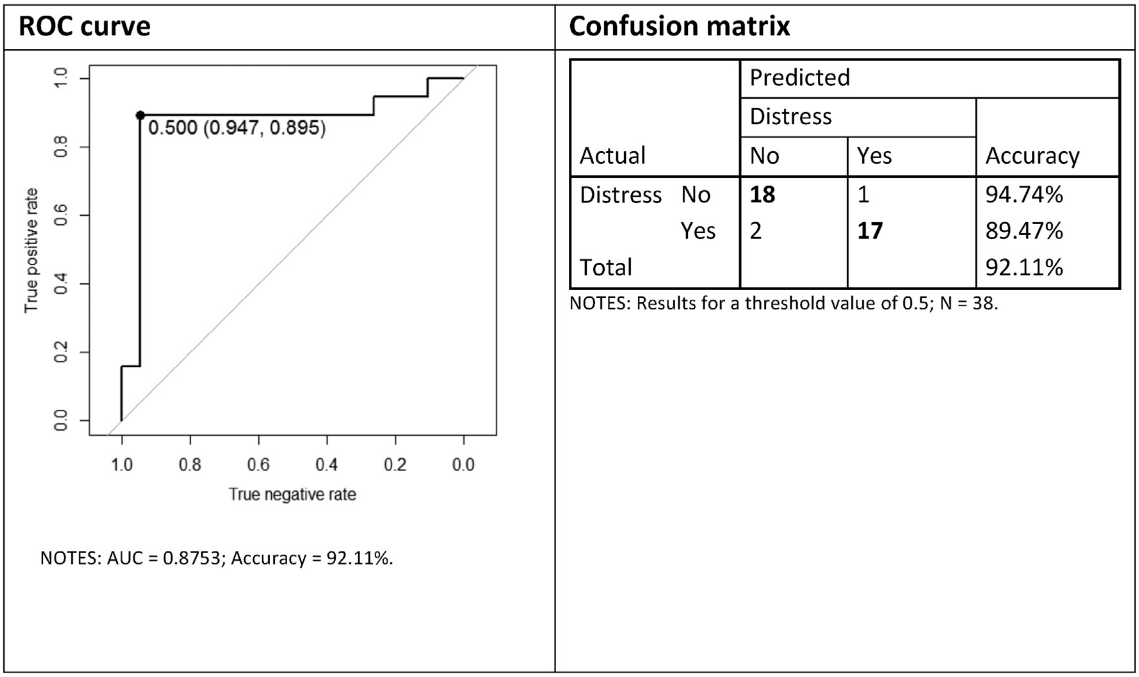


Fig. 10. Most important variables or features two years before failure.



NOTES: AUC = 0.8753; Accuracy = 92.11%.

		Predicted		
		Distress		
Actual		No	Yes	Accuracy
Distress	No	**18**	1	94.74%
	Yes	2	**17**	89.47%
Total				92.11%

NOTES: Results for a threshold value of 0.5; N = 38.

Fig. 11. Extreme gradient boosting performance for test data two years before failure.

using a validation or test dataset (independent data) to calculate the test accuracy rate. Validation or test data are used to predict the model's response to new observations. Given a dataset, a particular statistical learning method is appropriate if it yields a high accuracy test rate (Hastie, Tibshirani, & Friedman, 2009).

To measure the overall performance of our XGBoost model, Fig. 8 shows the receiver operating characteristic (ROC) curve for the XGBoost classifier applied to the test dataset. The ROC offers a useful visual tool to evaluate a classifier. It simultaneously displays the true positive rate (sensitivity of a classifier or proportion of correctly classified positive observations) and the true negative rate (specificity of a classifier or proportion of correctly classified negative observations) for all possible thresholds.[4] ROC curves do not depend on class probabilities because they summarize performance over all possible thresholds. Doing so facilitates their interpretation and comparison across different datasets and models. An ideal ROC curve hugs the top left corner, so a larger area under the ROC curve (AUC) denotes a better classifier.

In our study, the AUC statistic was used for quantitative assessment of the model. For the test data, the AUC was 0.978 (Fig. 8). This high AUC value indicates that the classifier performed well for the test data, correctly predicting almost every bank failure. Fig. 8 also displays the number of misclassifications for a 0.5 threshold value. The right panel shows the confusion matrix or table of prediction

[4] For binary scoring classifiers, a *threshold* (or *cutoff*) value controls how predicted posterior probabilities are converted into class labels.

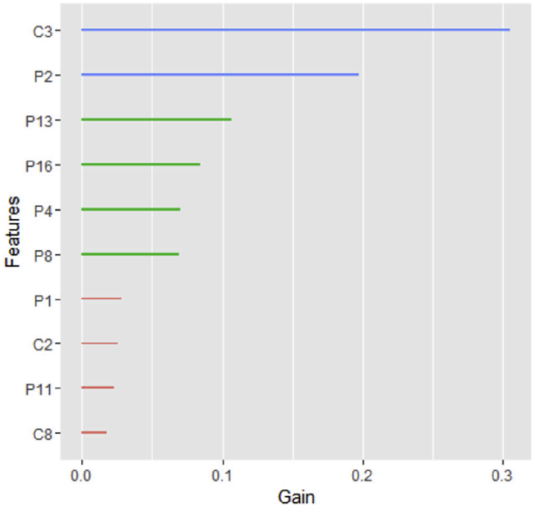


Fig. 12. Most important variables or features three years before failure.

results for this threshold. This table layout compares the actual outcomes against the predicted ones, enabling visualization of the performance of the XGBoost technique used to fit the model. The probability obtained by the XGBoost model was transformed into a binary prediction: 1 if it exceeded the threshold of 0.5, and 0 otherwise. The global percentage of errors or cases incorrectly classified for the test data was 5.26% (100%–94.74%). The model had a sensitivity rate of 89.47% (true positive rate) and a specificity rate of 100% (true negative rate). These results confirm that model accuracy was very good for the test dataset.

Likewise, the density plot of the predicted probabilities of the XGBoost model for bank distress shows that these values were higher for cases of bank distress and lower for cases of nondistress. Thus, the two probability distributions are almost perfectly separated (Fig. 9). The 0.5 probability threshold between the two groups would enable us to draw a vertical line that almost completely separated the predicted probability of bank failure. Therefore, the estimated model yielded a good prediction for the test data.

5.4. XGBoost comparison of two and three years before bank failure

In this section, the results discussed in the previous section are compared with other XGBoost models fitted in the two years before failure and in the three years before failure. The ROC curve (AUC) and global accuracy were used to assess these two new models for the test data. Fig. 10 displays the most important variables identified two years before failure. Fig. 11 shows an AUC value of 0.875 and a global accuracy of 92.11%, both obtained from the test data.

Likewise, for the XGBoost model fitted three years before failure on test data, Figs. 12 and 13 display an AUC statistic of 0.795 and a global accuracy of 78.95%. Therefore, according to these two performance measures, the model performance worsened over time. The best predictive performance was attained one year before failure.

Different combinations for the most influential variables emerged for all three models (one, two, and three years before failure). However, ratio P1 (yield on earning assets) was the fourth most important variable for both the one-year- and two-year-before-failure models. The ratio P2 (cost of funding earning assets) was the second most important variable for both the two-year- and three-year-before-failure models.

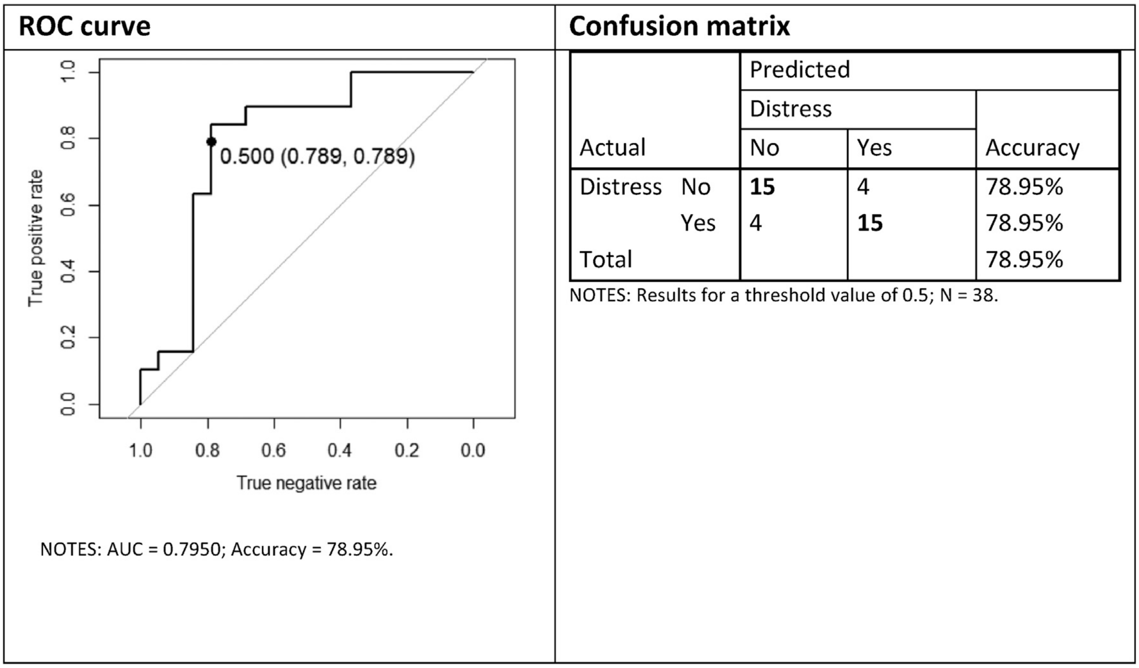
Finally, we compared the XGBoost performance for the three years under study. Fig. 14 shows 10-fold cross-validation of AUC (ROC) performance distributions for the three models and for 1000 iterations or rounds. The AUC decreased as the data for the fitted XGBoost model moved away from the date of the bank default (one, two, or three years). Indeed, p-values for the differences between the three models were 0.00, reflecting statistically significant differences in the AUC performance metric. These differences imply that the two-years-before-failure and the three-years-before-failure models performed worse than the one-year-before-failure model.

5.5. A parsimonious XGBoost model for the prediction of bank failure

As discussed previously, the XGBoost algorithm yielded good models for predicting bank failure. Performance metrics for the test data such as accuracy and AUC were high. These results indicate that models that are fitted to the training set do not overfit the banking data. We avoided overfitting by tuning and selecting the optimal XGBoost model parameter values through cross-validation. We now apply the principle of Occam's razor.[5]

According to Occam's razor, or the law of parsimony, for any two models that explain the occurrence of a certain event, the simpler

[5] Occam's razor is a problem-solving principle attributed to William of Ockham (c. 1287–1347), an English Franciscan friar, scholastic philosopher, and theologian.



NOTES: AUC = 0.7950; Accuracy = 78.95%.

		Predicted		
		Distress		
Actual		No	Yes	Accuracy
Distress	No	**15**	4	78.95%
	Yes	4	**15**	78.95%
Total				78.95%

NOTES: Results for a threshold value of 0.5; N = 38.

Fig. 13. Extreme gradient boosting performance for test data three years before failure.

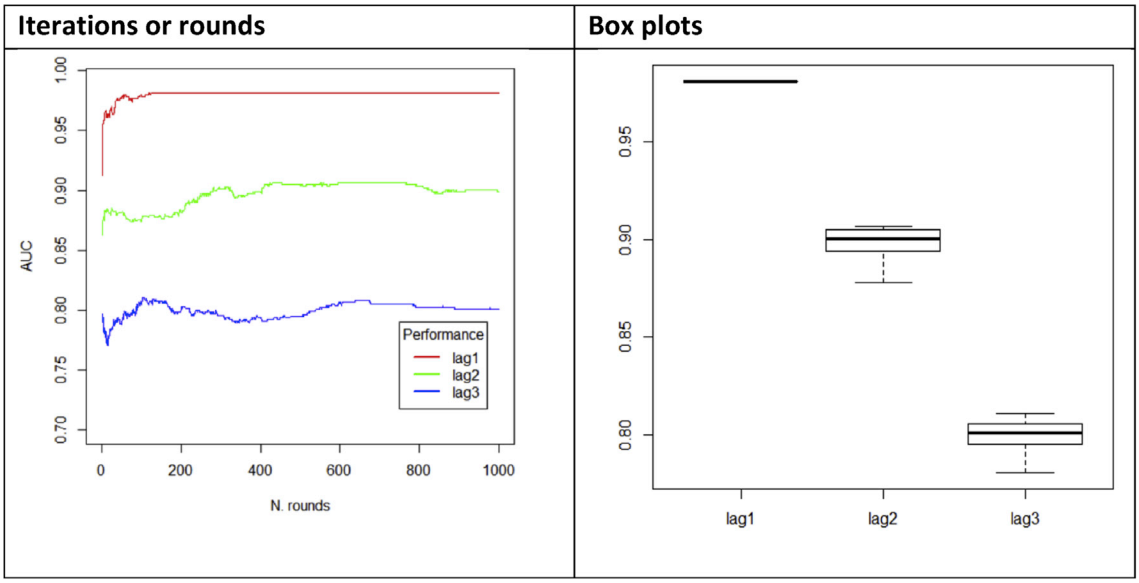
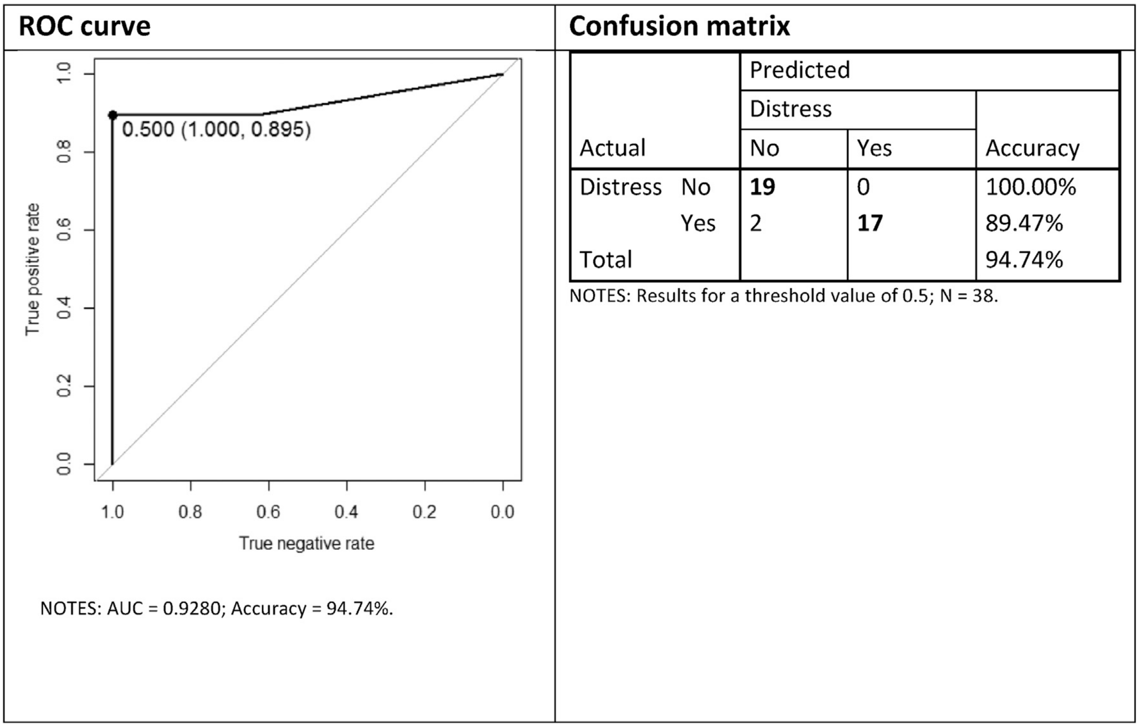


Fig. 14. 10-fold cross-validation AUC (ROC) performance distributions.

one is preferable. To test this law of parsimony, a new XGBoost model was fitted considering only the four most important variables: *P11*, *P9*, *C13*, and *P1*. Fig. 15 shows that the parsimonious model, fitted to the test data, had a high predictive global accuracy (94.74%) and a high AUC value (0.928). Therefore, a parsimonious approach that considers only the most important variables yields an excellent final model. With only four variables or ratios, it is possible to predict bank failure highly accurately using the XGBoost algorithm, after tuning and selecting the best model parameter configuration.

5.6. Extreme gradient boosting model interpretability

In this section, we show how to make an XGBoost model as transparent as a white box or a single decision tree. This state-of-the-art



NOTES: AUC = 0.9280; Accuracy = 94.74%.

Confusion matrix

		Predicted		
		Distress		
Actual		No	Yes	Accuracy
Distress	No	**19**	0	100.00%
	Yes	2	**17**	89.47%
Total				94.74%

NOTES: Results for a threshold value of 0.5; N = 38.

Fig. 15. Extreme gradient boosting performance for the test dataset using only the four most important variables.

procedure explains how each particular observation is classified. XGBoost interpretability was carried out for the parsimonious model.

The XGBoost models fitted in the previous sections yield a likelihood of bank failure. However, because many trees contribute to the prediction, it is difficult to judge the influence of each individual variable. The *R XGBoost Explainer* package (Foster, 2017a) allows predictions from an XGBoost model to be split into the impact of each feature, making the model as transparent as a linear regression or decision tree.

For each bank, the prediction of the likelihood of bank failure is broken down into the impact of each individual feature. Specifically, the prediction log odds are broken down. The probability is just the logistic function applied to the log-odds (Foster, 2017b).

XGBoost bank failure prediction is explained in Fig. 16 for two different cases: The first case (left panel) shows a predicted failed bank for an actual failed bank, and the second case (right panel) shows a predicted nonfailed bank for an actual nonfailed bank. For example, the 94.79% prediction is broken down into the influence of each individual feature. Specifically, it breaks down the log odds of the bank failure prediction, which in this case are 2.903. Step-by-step, the log odds change as follows:

- 0.03: Baseline (Intercept)
+ 1.17: P9 (Pretax return on assets) [prediction is now 1.14]
+ 1.07: C13 (Total risk-based capital ratio) [prediction is now 2.21]
+ 0.68: P11 (Retained earnings to average equity) [prediction is now 2.89]
+ 0.01: P1 (Yield on earning assets) [prediction is now 2.9]

This kind of breakdown can be applied to every bank in the test dataset to fully explain the XGBoost predictions. The key to interpreting the breakdown is understanding how the log odds contribution of each feature is calculated. It involves adding up the contributions of each feature for every tree in the ensemble in exactly the same way as for a single decision tree. Thus, each prediction is expressed as the sum of feature impacts. These impacts are not static coefficients as in a logistic regression. The impact of a feature depends on the specific path that the observation took through the ensemble of trees (Foster, 2017b).

5.7. Comparison with other methodologies

Although thoroughly comparing different machine learning methods is beyond the scope of this study, this section compares the findings discussed above with two other approaches to bank failure prediction: a conventional method (logistic regression) and a modern machine learning approach (the random forest algorithm). The results show that XGBoost performs best.

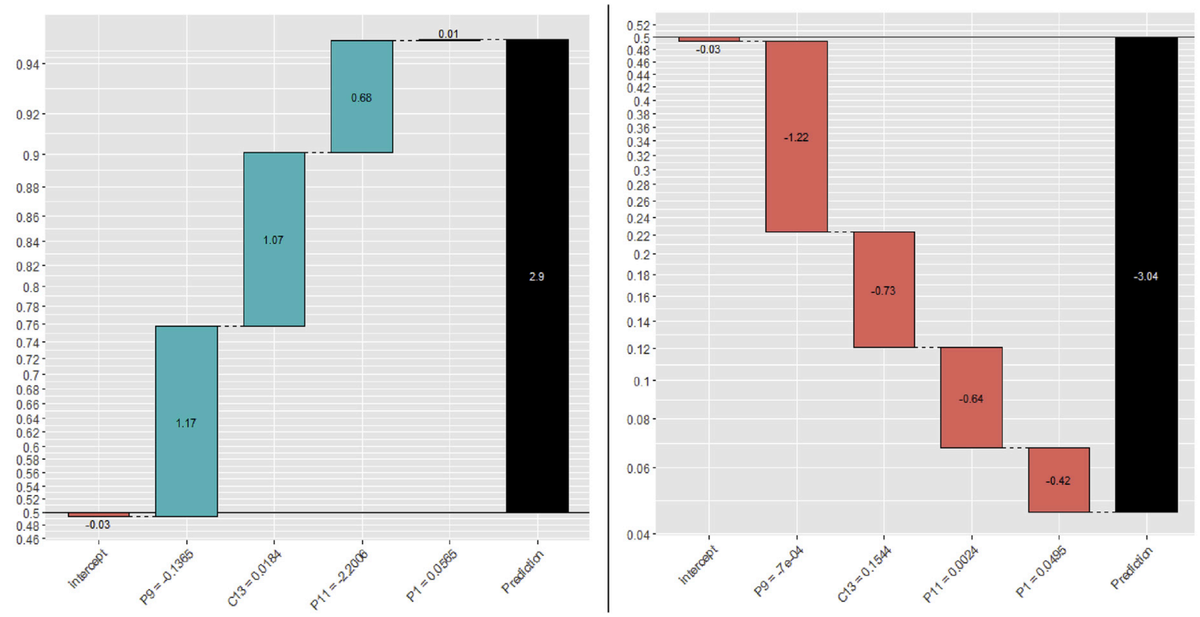
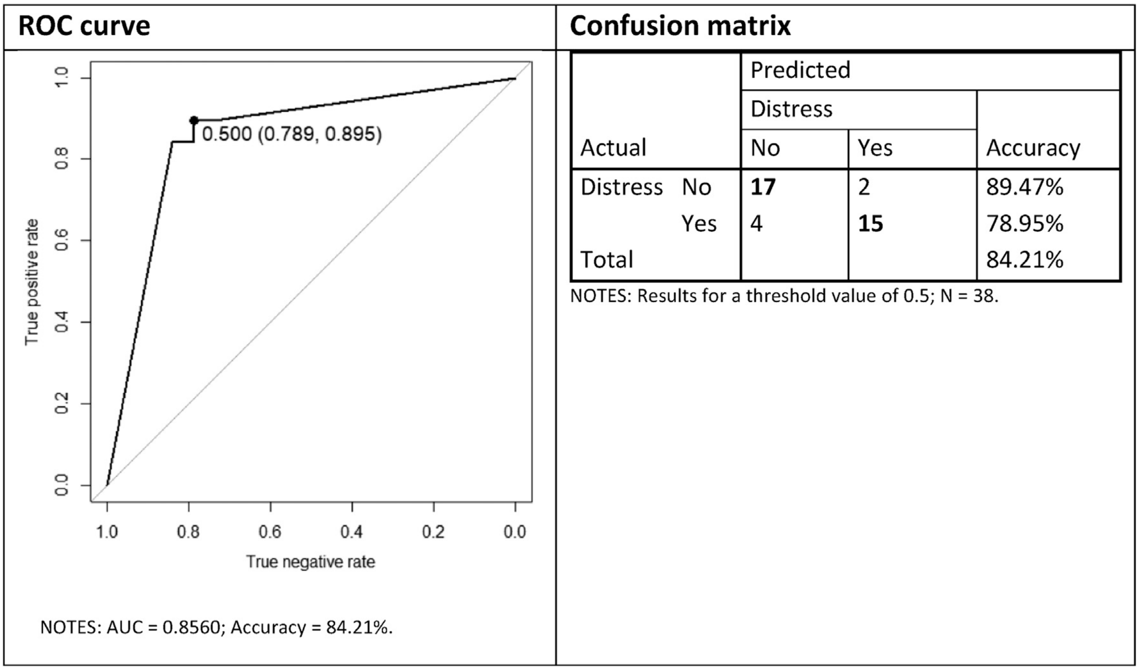


Fig. 16. Extreme gradient boosting prediction: Explanation of bank failure for two cases.


ROC curve

NOTES: AUC = 0.8560; Accuracy = 84.21%.

Confusion matrix

Actual		Predicted Distress No	 Yes	 Accuracy
Distress	No	**17**	2	89.47%
	Yes	4	**15**	78.95%
Total				84.21%

NOTES: Results for a threshold value of 0.5; N = 38.

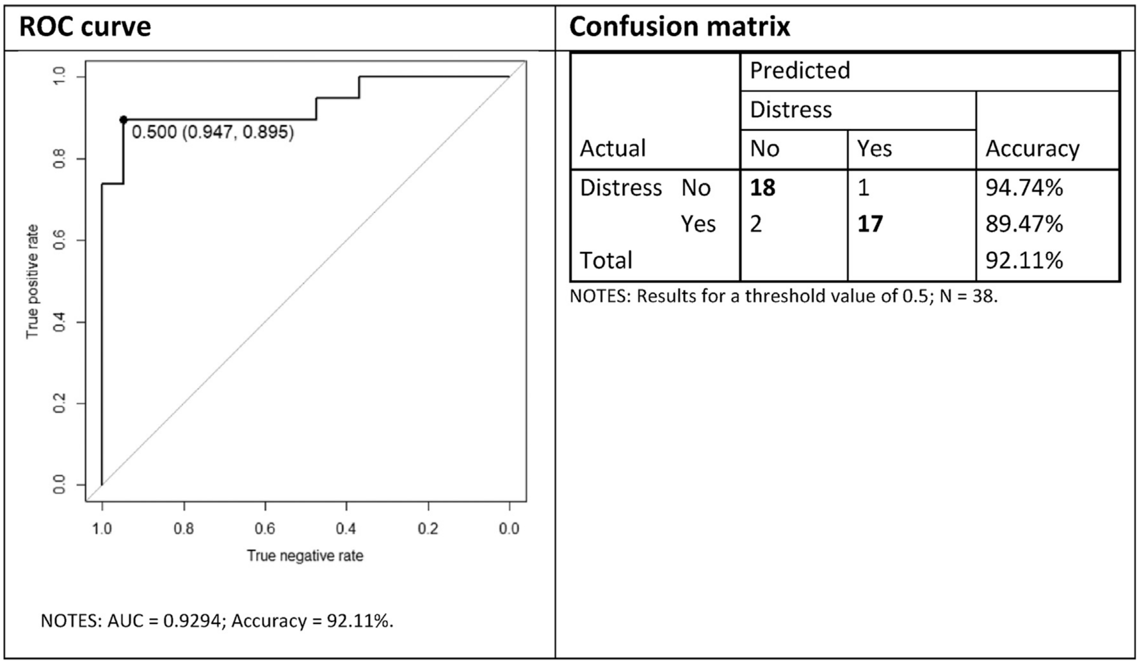
Fig. 17. Logistic regression performance for the test dataset.

5.7.1. Logistic regression

Logistic regression is a conventional approach that is widely applied to bank failure prediction. A logit design implies a linear model. Its key advantage over nonlinear methods is its interpretability. A summary of the resulting model is reported in Fig. 17, which displays the overall performance of the logistic regression for the test data. The AUC was 0.86 (left panel). For a threshold of 0.5, the confusion matrix of the logistic regression prediction of bank failure (right panel) illustrates overall accuracy of 84.21%.

5.7.2. Random forest

Random forest is one of the most common supervised classification algorithms. It is based on decision tree models. Random forest



ROC curve

NOTES: AUC = 0.9294; Accuracy = 92.11%.

Confusion matrix

		Predicted		
		Distress		
Actual		No	Yes	Accuracy
Distress	No	**18**	1	94.74%
	Yes	2	**17**	89.47%
Total				92.11%

NOTES: Results for a threshold value of 0.5; N = 38.

Fig. 18. Random forest performance for the test dataset.

tries to build multiple tree models with different samples and different variables, creating a forest with a certain number of trees. For instance, it might require a random sample of 30 bank observations and 5 randomly chosen ratios or predictors to build a tree model. It repeats the process a designated number of times and then makes a final failure prediction for each instance as the mean of each individual tree prediction. Thus, final predictions are a function of each tree model prediction (Shrivastava, 2014). Fig. 18 shows a summary of the resulting random forest model. It exhibits an AUC performance of 0.93 for the test data (left panel). For a threshold of 0.5, the confusion matrix of the random forest prediction of bank failure (right panel) shows overall accuracy of 92.11%.

As reflected by the analysis, both classification models (i.e., logistic regression and random forest) yield worse performance metrics than the XGBoost algorithm. Table 3 summarizes the results for the test dataset.

6. Conclusions

Our goal was to predict bank failure in the U.S. banking sector. To this end, we conducted empirical analysis using Extreme Gradient Boosting. This method was developed from other boosting methods such as AdaBoost and boosted classification trees, and it has been applied in recent business failure forecasting studies. We tested the suitability of the XGBoost algorithm and its capability to improve the accuracy of bank failure prediction. The sampling strategy of case-control matching helped eliminate any bias between failed and nonfailed banks. This study focused on predictive power rather than explanatory modeling. The ultimate goal was to develop a model that avoids overfitting and makes generalizable predictions. XGBoost has greater predictive power than both Logistic Regression and Random Forest methods.

The four most influential variables identified by the general one-year-before-failure XGBoost model were used to train a parsimonious model. The 95% total predictive accuracy of the parsimonious model validated the relevance of the four variables and highlighted the need to examine these variables from a practitioner's perspective.

Retained earnings is a major source of financing that is worthy of managers' attention. Although keeping shareholders satisfied through dividend payments is important, it is even more important to ensure financial soundness and funding growth through reinvestment of net income. Under certain circumstances, managers should adjust dividend policies or even temporarily suspend dividend payments until market and business conditions improve.

Pretax return on assets offers a direct performance measure that managers must track closely and compare with direct competitors and the industry average. Underperformance is a first sign of potential financial distress, and it requires rapid response from managers. Bank managers should be able to identify the underlying reasons for long-term underperformance so that they can address these issues.

Total risk-based capital offers a cushion against asset malfunction, so high levels of bank capitalization help prevent bank failure, which occurs when a bank is unable to pay depositors and creditors. When technology or real estate bubbles burst and bad loans are written off, the value of assets falls. Only banks with high capital have the capital required to remain solvent. Unfortunately, however, because of the high costs of holding capital, bank managers tend to hold less bank capital than is required by regulatory authorities.

Managers may view above-industry-average yields on earning assets as appealing. Yet yields that far exceed the industry-average

Table 3
Summary of model performance for bank distress (test dataset).

Model	Performance	
	AUC	Accuracy
XGBoost	0.98	94.74%
Logistic regression	0.84	84.21%
Random forest	0.93	92.11%

NOTE: Models fitted on training data and performance measured on test data.

often come from highly risky investments. Managers should avoid high-risk investments that improve the bottom line in one year only. Instead, managers should focus on investments that are likely to offer steady, sustainable income streams that support long-term financial soundness. The counter-intuitive nature of this variable may deserve further research through a qualitative analysis of bank financial statements.

In summary, the high predictive power of the model developed in this paper should encourage bank managers to track the four scorecard variables that were identified in the general one-year-before-failure complete model and that were validated in the parsimonious model. Managers can avoid financial distress by taking timely action rather than waiting for government intervention. Regulators may also find the model useful to identify and warn potentially distressed banks.

This study offers some interesting research prospects. Expanding the sample to include state banks could improve the model's robustness, even though the average size of U.S. state banks is small. In addition, taking a balanced sample of European failed and nonfailed banks for the same period and identifying the most relevant leading indicators of bank failure would help identify differences and similarities between the U.S. and Europe banking sectors.

Acknowledgments

The authors express their gratitude to the Ministerio de Economía y Competitividad for providing financial support through project I + D: ECO2013-40816-P. Francisco Climent express their gratitude to the Ministerio de Economía y Competitividad for providing financial support through project I+D: ECO2013-40816-P.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.iref.2018.03.008.

Appendix 1. Definition of explanatory variables

1. Performance ratios	Variable	Definition
Key		
P1	Yield on earning assets	Total interest income (annualized) as a percent of average income derived from bank services and sources other than interest bearing assets (annualized) as a percent of average total assets. Earning assets: the average of all loans and other investments that earn interest or dividends
P2	Cost of funding earning assets	Annualized total interest expense on deposits and other borrowed money as a percent of average earning assets on a consolidated basis
P3	Net interest margin	Total interest income less total interest expense (annualized) as a percent of average earning assets
P4	Noninterest income to assets	Income derived from bank services and sources other than interest bearing assets (annualized) as a percent of average total assets
P5	Noninterest expense to assets	Salaries and employee benefits, expenses of premises and fixed assets, and other noninterest expenses (annualized) as a percent of average total assets
P6	Loan and lease loss provision to assets	The annualized provision for loans and lease losses as a percent of average total assets on a consolidated basis
P7	Net operating income to assets	Net operating income (annualized) as a percent of average total assets
P8	Return on assets (ROA)	Net income after taxes and extraordinary items (annualized) as a percent of average total assets
P9	Pretax return on assets	Annualized pretax net income as a percent of average total assets (includes extraordinary items and other adjustments, net of taxes)
P10	Return on equity (ROE)	Annualized net income as a percent of average total equity on a consolidated basis
P11	Retained earnings to average equity (YTD only)	Net income (year-to-date, annualized), less cash dividends declared (year-to-date, annualized), as a percent of average total equity capital

(*continued on next page*)

(*continued*)

1. Performance ratios	Variable	Definition
Key		
P12	Net charge-offs to loans and leases	Gross loan and lease financing receivable charge-offs, less gross recoveries, (annualized) as a percent of average total loans and lease financing receivables. Average total loans and lease financing receivables: the average of total loans and lease financing receivables, net of unearned income
P13	Loan and lease loss provision to net charge-offs	Provision for possible credit and allocated transfer risk as a percent of net charge-offs
P14	Earnings coverage of net loan charge-offs (X)	Income before income taxes and extraordinary items and other adjustments, plus provisions for loan and lease losses and allocated transfer risk reserve, plus gains (losses) on securities not held in trading accounts (annualized) divided by net loan and lease charge-offs (annualized)
P15	Efficiency ratio	Noninterest expense less amortization of intangible assets as a percent of net interest income plus noninterest income. This ratio measures the proportion of net operating revenues that are absorbed by overhead expenses, so that a lower value indicates greater efficiency.
P16	Assets per employee (USD millions)	Total assets in millions of dollars as a percent of the number of full-time equivalent employees
P17	Cash dividends to net income (YTD only)	Total of all cash dividends declared (year-to-date, annualized) as a percent of net income (year-to-date, annualized)

2. Condition Ratios	Variable	Definition
Key		
C1	Earning assets to total assets	Interest earning assets as a percent of total assets
C2	Loss allowance to loans and leases	Allowance for loan and lease losses as a percent of total loan and lease financing receivables, excluding unearned income
C3	Loss allowance to noncurrent loans and leases	Allowance for loan and lease losses as a percent of noncurrent loans and leases
C4	Noncurrent assets plus other real estate owned to assets	Noncurrent assets as a percent of total assets. Noncurrent assets are defined as assets that are past due 90 days or more plus assets placed in nonaccrual status plus other real estate owned (excluding direct and indirect investments in real estate)
C5	Noncurrent loans to loans	Total noncurrent loans and leases, Loans and leases 90 days or more past due plus loans in nonaccrual status, as a percent of loans and leases (including unearned income)
C6	Net loans and leases to assets	Loan and lease financing receivables, net of unearned income, allowances, and reserves, as a percent of total assets
C7	Net loans and leases to deposits	Loans and lease financing receivables net of unearned income, allowances and reserves as a percent of total deposits
C8	Net loans and leases to core deposits	Loan and lease financing receivables, net of allowances and reserves, as a percent of core deposits
C9	Domestic deposits to total assets	Total domestic office deposits as a percent of total assets
C10	Equity capital to assets	Total equity capital as a percent of total assets
C11	Core capital (leverage) ratio	Tier 1 (core) capital as a percent of average total assets minus ineligible intangibles
C12	Tier 1 risk-based capital ratio	Tier 1 (core) capital as a percent of risk-weighted assets as defined by the appropriate federal regulator for prompt corrective action during that time period
C13	Total risk-based capital ratio	Total risk based capital as a percent of risk-weighted assets as defined by the appropriate federal regulator for prompt corrective action during that time period

3. Other	Variable	Definition
Key		
TA	Total assets	The sum of all assets owned by the institution including cash, loans, securities, bank premises and other assets

References

Al Wakil, A. (2018). When gambling is not Winning: Exploring optimality of VIX trading under the expected utility theory. *Journal of Business Accounting and Finance Perspectives, 1*(1). https://proview.thomsonreuters.com/title.html?redirect=true&titleKey=aranz%2Fmonografias%2F193456932%2Fv1.4&titleStage=F&titleAcct=i0adc419000000147a1c91f053c81274e#sl=e&eid=4afad5d4d7866f860e9f5ef710d336ee&eat=a-196305847&pg=4&psl=&nvgS=false (Accessed October 2017).

Alfaro, E., García, N., Gámez, M., & Elizondo, D. (2008). Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks. *Decision Support Systems, 45*, 110–122.

Amendola, A., Restaino, M., & Sensini, L. (2015). An analysis of the determinants of financial distress in Italy: A competing risks approach. *International Review of Economics & Finance, 37*, 33–41.

Berger, A. N., & Bouwman, C. H. S. (2013). How does capital affect bank performance during financial crises? *Journal of Financial Economics, 109*, 146–176.

Betz, F., Oprica, S., Peltonen, T. A., & Sarlin, P. (2014). Predicting distress in European banks. *Journal of Banking & Finance, 45*, 225–241.

Boyd, J. H., & Runkle, D. E. (1993). Size and performance of banking firms: Testing the predictions of theory. *Journal of Monetary Economics, 31*(1), 47–67.

Brownlee, J. (2016). *How to tune the number and size of decision trees with XGBoost in Python*. http://machinelearningmastery.com/tune-number-size-decision-trees-XGBoost-python/ (accessed September 2016).

Cao, Y., Wan, G., & Wang, F. (2011). Predicting financial distress of Chinese listed companies using rough set theory and support vector machine. *Asia Pacific Journal of Operational Research, 28*(1), 95–109.

Capatina, A., Bleoju, G., Matos, F., & Vairinhos, V. (2017). Leveraging intellectual capital through Lewin's Force Field Analysis: The case of software development companies. *Journal of Innovation and Knowledge, 2*(3), 125–133.

Chambers, M., & Dinsmore, T. W. (2015). *Advanced analytics Methodologies: Driving business value with analytics*. New Jersey, USA: Pearson Education.

Chen, T., & Benesty, M. (2016). *XGBoost: eXtreme gradient boosting. R package version 0.4-3*. https://cran.r-project.org/web/packages/XGBoost/vignettes/XGBoost.pdf.

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. arXiv:1603.02754v3 [cs.LG].

Chiaramonte, L., Liu, H., Poli, F., & Zhou, M. (2016). How accurately can z-score predict bank failure? *Financial Markets, Institutions & Instruments, 25*(5), 333–360.

Cleary, S., & Hebb, G. (2016). An efficient and functional model for predicting bank distress: In and out of sample evidence. *Journal of Banking & Finance, 64*, 101–111.

Collier, C., Forbush, S., Nuxoll, D., & O'Keefe, J. (2003). The SCOR system of off-site monitoring. *FDIC Banking Review, 15*(3), 17–32.

Cortés, E. A., Martínez, M. G., & Rubio, N. G. (2008). Linear discriminant analysis versus adaboost for failure forecasting. *Revista Española de Financiación y Contabilidad, 37*(137), 13–32.

Dell'Ariccia, G., Igan, D., & Laeven, L. (2012). Credit booms and lending standards: Evidence from the subprime mortgage market. *Journal of Money, Credit, and Banking, 44*(2–3), 367–384. March–April 2012.

Demyanyk, Y., & Hasan, I. (2010). Financial crises and bank failures: A review of prediction methods. *Omega, 38*, 315–324.

DeYoung, R., & Torna, G. (2013). Nontraditional banking activities and bank failures during the financial crisis. *Journal of Financial Intermediation, 22*(3), 397–421.

Ecer, F. (2013). Comparing the bank failure prediction performance of neural networks and support vector machines: The Turkish case. *Economic Research, 26*(3), 81–98.

Ekinci, A., & Erdal, H. I. (2017). Forecasting bank failure: Base learners, ensembles and hybrid ensembles. *Computational Economics, 49*(4), 677–686.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecography, 77*, 802–813.

European Commission. (2012). *Comissión staff working document, impact assessment, SWD(2012) 167 final, brussels*, 6.6.2012.

Eygi, B. (2013). Prediction of bankruptcy using support vector machines: An application to bank bankruptcy. *Journal of Statistical Computation and Simulation, 83*(8), 1543–1555.

FDIC. (2016). *bank failures in brief*. www.fdic.gov/bank/historical/bank/index.html (Accessed December, 2016).

Foster, D. (2017a). *xgboostExplainer: XGBoost Model Explainer. R package version 0.1*. https://github.com/AppliedDataSciencePartners/xgboostExplainer.

Foster, D. (2017b). *NEW R package that makes XGBoost interpretable*. https://medium.com/applied-data-science/new-r-package-the-xgboost-explainer-51dd7d1aa211 (Accessed January, 2017).

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367–378.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. New York, USA: Springer.

He, T. (2016). *An Introduction to XGBoost R package*. http://dmlc.ml/rstats/2016/03/10/XGBoost.html (accessed July 2016).

Jagtiani, J., Kolari, J., Lemieux, C., & Shin, H. (2003). Early warning models for bank supervision: Simpler could be better. *Economic Perspectives Federal Reserve Bank of Chicago, 3Q*(2003), 49–60.

Jain, A. (2016). *Complete guide to parameter tuning in XGBoost*. http://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-XGBoost-with-codes-python/ (Accessed January 2017).

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning* (7th ed.). New York: Springer.

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.

Kaggle. (2016). https://www.kaggle.com/about (Accessed September 2016).

Krolikowski, M. W. (2018). Economic shocks, competition and merger activity. *Journal of Business Accounting and Finance Perspectives, 1*(1). https://proview.thomsonreuters.com/title.html?redirect=true&titleKey=aranz%2Fmonografias%2F193456932%2Fv1.4&titleStage=F&titleAcct=i0adc419000000147a1c91f053c81274e#sl=e&eid=4afad5d4d7866f860e9f5ef710d336ee&eat=a-196305847&pg=4&psl=&nvgS=false (Accessed September 2017).

Kuhn, M. (2016). *Caret: Classification and regression training. R package version 6.0-68*. http://CRAN.R-project.org/package=caret.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. New York, NY: Springer.

Kumar, P. R., & Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques–A review. *European Journal of Operational Research, 180*, 1–28.

Lane, W. R., Looney, S. W., & Wansley, J. W. (1986). An application of the Cox proportional hazards model to bank failure. *Journal of Banking & Finance, 10*, 511–531.

Lu, W., & Whidbee, D. A. (2013). Bank structure and failure during the financial crisis. *Journal of Financial Economic Policy, 5*(3), 281–299.

Lu, W., & Whidbee, D. A. (2016). US bank failure and bailout during the financial crisis Examining the determinants of regulatory intervention decisions. *Journal of Financial Economic Policy, 8*(3), 316–347.

Maciej 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.

Meyer, P. A., & Pifer, H. W. (1970). Prediction of bank failures. *The Journal of Finance, 25*(4), 853–868.

Momparler, A., Carmona, P., & Climent, F. (2016). Banking failure prediction: A boosting classification tree approach. *Spanish Journal of Finance and Accounting/Revista Española de Financiación y Contabilidad, 45*(1), 63–91.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics, 7*(21).

Olson, D., Denle, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems, 52*, 464–473.

R Core Team. (2016). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. URL http://www.R-project.org/.

Rey-Moreno, M., & Medina-Molina, C. (2017). Inhibitors of e-Government adoption: Determinants of habit and adoption intentions. *Journal of Innovation and Knowledge, 2*(3), 172–180.

Scott Fortmann-Roe. (2012). *Understanding the bias-variance tradeoff*. http://scott.fortmann-roe.com/docs/BiasVariance.html (Accessed September 2017).

Serrano-Cinca, C., Fuertes-Callén, Y., Gutiérrez-Nieto, B., & Cuellar-Fernández, B. (2014). Path modelling to bankruptcy: Causes and symptoms of the banking crisis. *Applied Economics, 46*(31), 3798–3811.

Serrano-Cinca, C., & Gutiérrez-Nieto, B. (2013). Partial least square discriminant analysis for bankruptcy prediction. *Decision Support Systems, 54*, 1245–1255.

Shrivastava, T. (2014). *Introduction to random forest – simplified*. https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/ (Accessed December 2017).

Tian, S., & Yu, Y. (2017). Financial ratios and bankruptcy predictions: An international evidence. *International Review of Economics & Finance, 51*, 510–526.

Trussel, J. J., & Johnson, L. (2012). A parsimonious and predictive model of the recent bank failures. *Academy of Banking Studies Journal, 11*(1), 15–30.

Volkov, A., Benoit, D. F., & Van den Poel, D. (2017). Incorporating sequential information in bankruptcy prediction with predictors based on Markov for discrimination. *Decision Support Systems, 98*, 59–68.