Risk Management (2024) 26:1
https://doi.org/10.1057/s41283-023-00132-2

ORIGINAL ARTICLE

Machine learning techniques for default prediction: an application to small Italian companies

Flavio Bazzana[1] · Marco Bee[2] · Ahmed Almustfa Hussin Adam Khatir[3]

Accepted: 18 September 2023 / Published online: 15 November 2023

Abstract

Default prediction is the primary goal of credit risk management. This problem has long been tackled using well-established statistical classification models. Still, nowadays, the availability of large datasets and cheap software implementations makes it possible to employ machine learning techniques. This paper uses a large sample of small Italian companies to compare the performance of various machine learning classifiers and a more traditional logistic regression approach. In particular, we perform feature selection, use the algorithms for default prediction, evaluate their accuracy, and find a more suitable threshold as a function of sensitivity and specificity. Our outcomes suggest that machine learning is slightly better than logistic regression. However, the relatively small performance gain is insufficient to conclude that classical statistical classifiers should be abandoned, as they are characterized by more straightforward interpretation and implementation.

Keywords Default risk · Classification · Feature selection · Imbalanced classes

✉ Marco Bee
marco.bee@unitn.it

Flavio Bazzana
flavio.bazzana@unitn.it

Ahmed Almustfa Hussin Adam Khatir
AhmedKhatir22@gmail.com

[1] Department of Economics and Management, University of Trento, Trento, Italy

[2] Department of Economics and Management, University of Trento, via Inama, 5, 38122 Trento, Italy

[3] Tomasi Auto, Mantova, Italy

Introduction

In recent years, attention to machine learning (ML) has increased dramatically in many fields, and credit risk management makes no exception. Indeed, credit risk measurement mainly deals with classification, which is one of the most important goals of ML techniques. On the other hand, extensive literature dating back to the seminal works by Altman (1968) and Merton (1974) uses statistical methods for bankruptcy prediction. In particular, discriminant analysis and logistic regression are well known and widely employed: see, e.g., Duffie and Singleton (2003) or Bolder (2019) for details.

Most approaches to credit risk are based on supervised learning models characterized by a binary target variable y, the so-called default indicator, and d possible predictors, usually given by financial ratios and customer-related variables. The size of the data ranges from relatively small in the large-corporate case, where the observations are large companies, to very large in the retail credit framework. Accordingly, the methods of analysis and the independent variables employed are quite different: in the former case, financial information contained in balance sheets is of paramount importance, whereas in the latter setup, only basic personal data are available.

In both the credit risk and the ML literature, many papers about the impact of ML methods have been published in the last few years: for a recent overview, see, e.g., van Liebergen (2017), Leo et al. (2019), Shi et al. (2022), and the references therein. On the practitioners' side, the major rating agencies, such as Moody's and Standard & Poor's, have also considered ML techniques: see Bacham and Zhao (2017) and Vidovic and Yue (2020). Furthermore, a survey by the Bank of England (Bank of England, Financial Conduct Authority 2019) concludes that about two-thirds of the respondents use machine learning techniques to some extent.

This paper exploits a large dataset of Italian small companies with various aims. We implement the main ML methods and a benchmark statistical approach, namely logistic regression (LR), to rank the ML techniques' performance and compare ML to LR. It has generally been found (see, e.g., Shi et al. 2022) that ML outperforms statistical techniques in terms of error rate. However, sensitivity and specificity are often more important than overall classification accuracy, especially in the credit risk setup, where classes are typically imbalanced. Hence, we try to take care of this issue by finding the classification cutoff as a function of sensitivity and specificity. Moreover, before using the classification algorithms, we perform a predictor-selection exercise based on three approaches: Random Forest Recursive Feature Elimination, Chi-Squared Feature Selection, and L_1-based Feature Selection.

To sum up, the paper contributes to the credit risk management literature by comparing the performance of ML classifiers using a large database of Italian small companies. The analysis considers all the steps required by typical credit risks datasets, such as predictor selection, consideration of class imbalance, and comparison to state-of-the-art statistical techniques. Even though the most recent literature suggests that ML methods are more accurate than classical statistical approaches, to perform a fair comparison, one should consider that ML methods are less interpretable, computationally heavier and often require the user to carry out some non-trivial preliminary fine-tuning of input parameters. Thus, we aim to study ML and LR's relative usefulness according to all these remarks. Such an analysis is crucial, given the possible use of the techniques in the banking sector, where interpretability and ease of implementation are crucial.

The rest of the paper is organized as follows. In "Methodology" section, we review the main machine learning techniques used in classification setups, particularly on credit risk applications. In "Empirical analysis" section, we perform feature selection and apply the methods to an extensive database of Italian small companies: the analysis is based on the ML methods introduced in the previous section and on LR; classification theory is implemented both using the standard 50% classification threshold, and a smaller threshold found according to sensitivity and specificity. Finally, in "Conclusion" section, we discuss our findings and outline some issues open to further research. Some outcomes are reported in an online supplement.

Methodology

Related work

Since the literature about applications of ML approaches in credit risk measurement has grown markedly in the last few years, it would be impossible to give a complete account here. Hence, in this section, we limit ourselves to discussing some recent contributions that analyze the accuracy of ML and statistical classifiers.

Barboza et al. (2017) conducts a comparative assessment of the bankruptcy prediction performance of support vector machines, bagging, boosting, random forests, and neural networks with respect to some statistical models (discriminant analysis, logistic regression). The paper uses data on North American firms from 1985 to 2013, integrating information from the Salomon Center database and Compustat and analyzing more than 10,000 firm-year observations. To improve the prediction accuracy of the models, six financial indicators in addition to the original Altman's Z-score are employed; see Carton and Hofer (2006) for details. The takeaway is that ML algorithms are approximately 10% more efficient than traditional models.

Similarly, Le and Viviani (2018) carries out a comparative analysis of the prediction accuracy of statistical approaches (discriminant analysis and logistic regression) relative to three ML methods (neural networks, support vector machines, and K-nearest neighbors). The data are collected for five years for US banks so that the dataset contains 3000 observations (1438 defaulted and 1562 active banks) with 31 financial ratios to be used as predictors. The main findings are that neural networks and K-nearest neighbors significantly outperform statistical models, whereas support vector machines are not better.

Moscatelli et al. (2019) try to shed some light on the forecasting performance of random forests and gradient-boosted trees with respect to statistical classifiers (discriminant analysis, logistic regression, and penalized logistic regression). The dataset contains about 300,000 observations of financial ratios and credit behavior indicators for Italian non-financial firms from 2011 to 2017. The authors conclude that

ML models provide a more accurate forecasting performance regarding discriminatory power and precision. However, when the dataset size is insufficient to robustly estimate the relationships between the predictors and the target variable, the performances of ML and statistical models are not significantly different.

Dumitrescu et al. (2022) propose a high-performance and interpretable credit-scoring method called penalized logistic tree regression (PLTR), using Monte Carlo simulation to improve the performance of logistic regression using the information provided by decision trees. For this purpose, decision trees are built with the original predictive variables, and rules extracted from various short-depth decision trees are used as predictors in a penalized logistic regression model. Applying the algorithm to several credit-scoring datasets suggests that PLTR has a better out-of-sample performance with respect to traditional linear and non-linear logistic regression, support vector machines, and neural networks and is competitive relative to the random forest.

Finally, over the previous decade, Shi et al. (2022) review many research papers using statistical, ML, and deep learning approaches in the credit risk setup. The authors also consider further issues, such as data imbalance, dataset inconsistency, model transparency, and inappropriate use of deep learning methods. The outcomes suggest that most deep learning models prefer classic ML and statistical classifiers and that the ensemble method outperforms single models.

Machine learning techniques

In this section, we describe the machine learning techniques used in the following; the reader interested in a more detailed statistical analysis is referred to James et al. (2021). Classification algorithms are supervised learning models estimated from training data whose class membership is known. The classifiers try to learn the relationship between the features and the indicator of class membership, where features (predictors) are individual measurable properties of the observed process. The performance of the trained model is then assessed on new data, the so-called test set (James et al. 2021).

Let y_i and $\boldsymbol{x}_i = (x_{i1}, \dots, x_{id})^T$, $i = 1, \dots, n$ be the training observations, where d is the number of features available. In the present setup, $y \in \{0, 1\}$ is a binary variable: 0 and 1 correspond to non-default and default, respectively. Overall, all the methods aim at modeling the relation between $\boldsymbol{x}$ and y. The assumptions about the function $f : \boldsymbol{x} \to y$ and the techniques employed for its estimation differ widely across classifiers. Ultimately, f will be used to compute the estimated default probability $P(y = 1|\boldsymbol{x})$: with balanced classes, if $P(y = 1|\boldsymbol{x}) > 0.5$, the new observation is classified as Default, otherwise the estimated class is Active (see, e.g., James et al. 2021):

$$\hat{y}_i = \begin{cases} 1 & \text{if } P(y_i = 1|\boldsymbol{x}) > 0.5; \\ 0 & \text{if } P(y_i = 1|\boldsymbol{x}) \leq 0.5. \end{cases} \tag{1}$$

Tree-based methods

A decision tree (DT) is determined by a series of decisions represented as a tree structure (Breiman et al. 1984). The intermediate nodes are based on a single feature, and

Algorithm 1 Random Forest

```
Given training observations $(y_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$, the number of features $d_e$
selected for the ensembles, and the number of trees $B$ in the ensemble, the
following steps are performed:

  1. Use bagging (Sect. 8.2.1) to create $B$ training samples of size $n$;
  2. Choose the predictors randomly from the $d_e$ features selected in advance,
     and grow the trees without pruning;
  3. Use the training samples obtained in Step 1 to train B decision trees.
     Make a final classification decision via the majority vote mechanism.
```

the terminal nodes correspond to the final classification. Each split is determined by node purity, measured by either the Gini index or the cross-entropy: the higher the purity, the stronger the predictive power of the predictor used in the corresponding node. The probability that an observation $(y, \boldsymbol{x})$ in the jth class ends up into the leaf C_s is estimated by

$$p_{sj} = \frac{\#\{(y, \boldsymbol{x}) \in C_s \, : \, y = S_j\}}{n_{C_s}},$$

where n_{C_s} is the number of training observations in C_s, and $S_j \in \{0, 1\}$.

Unfortunately, DTs are non-robust and suffer from high variance. Even though this problem can be mitigated (by pruning, for example), they are usually worse than other classifiers in terms of prediction accuracy. A significant improvement is provided by a straightforward generalization called random forest, which we now detail.

Random forests are ensembles of decision trees (Breiman 2001): B trees are built on bootstrapped samples obtained from the observed sample, and each tree uses a subset of randomly chosen features. Since each tree yields a predicted class, the RF prediction uses the majority vote criterion (James et al. 2021, p. 341) so that the predicted category is the most commonly occurring predicted class in the B bootstrapped trees.

With respect to decision trees, random forests have a marginally heavier computational burden but dramatically cut the variance, so that they have largely replaced decision trees in most practical applications.

Artificial neural networks

The peculiarity of an artificial neural network (NN) is the approximation of the non-linear function f linking the features $x_1, \dots, x_d$ to the dependent variable y. A single-layer NN with K hidden units has the form $f(X) = \beta_0 + \sum_{i=1}^{K} \beta_k h_k(X)$, where $A_k = h_k(X) = g(w_{k0} + \sum_{j=1}^{d} w_{kj} x_j)$, $k = 1, \dots, K$, are the so-called activations, or hidden units, and g is the prespecified activation function. Each activation is a different transformation of the original features. The activations map the hidden layer into the output layer:

$$f(X) = \beta_0 + \sum_{i=1}^{K} \beta_k A_k. \tag{2}$$

The most common activation function is the ReLU (Rectified linear unit, James et al. 2021, p. 405):

$$g(z) = \begin{cases} 0 & z < 0, \\ z & \text{otherwise} \end{cases}$$

All the parameters in (2) are estimated by minimizing a squared-error loss. The nonlinearity of the activation function g is essential: if it were linear, (2) would just reduce to a linear model. Nowadays, most NNs used in practice are based on either one hidden layer with many hidden units or on more than one hidden layer (multilayer neural networks).

Neural networks are a complex approach with many parameters, requiring a large sample size to be adequately trained. Hence, they have become more successful in recent years, mainly because of the availability of large datasets that guarantee an efficient learning process.

K-nearest neighbor

The K-Nearest Neighbor (KNN) algorithm is a simple non-parametric algorithm: it classifies a new observation with predictors $\boldsymbol{x}^*$ in the class of the majority of the K-nearest neighbors of $\boldsymbol{x}^*$ in the training dataset. The metric determining the nearest neighbors is usually the Euclidean distance. After identifying the neighborhood $\mathcal{N}_0$, the posterior probability of class j, $j = 1, \dots, C$, where C is the number of classes, is estimated as

$$P(Y = j | X = x^*) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathbb{1}(y_i = j),\ j = 1, \dots, C,$$

and $\boldsymbol{x}^*$ is assigned to the class with the largest posterior probability.

The algorithm's flexibility can be tuned by varying the "size" of the neighborhood (i.e., the parameter K). Since the method becomes more flexible as K decreases, too small values of K may overfit the training data.

Accuracy measures

The models are ranked according to the established evaluation measures in credit scoring. In particular, we use the area under the ROC curve (AUC; James et al. 2021, Sect. 4.4.2) and the average accuracy (equal to $1 - \text{er}$, where er is the error rate). To better interpret the analysis, we also compute the sensitivity (true positive rate, equal to $1 -$ Type II error) and the specificity (true negative rate, equivalent to $1 -$ Type I error) (see James et al. 2021, p. 152).

Average accuracy, sensitivity, and specificity are given by

$$\text{Av. Acc.} = \frac{\text{TP} + \text{TN}}{n}, \quad \text{Sens.} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{Spec.} = \frac{\text{TN}}{\text{TN} + \text{FP}},$$

where the quantities TP, TN, FP, and FN are shown in the so-called confusion matrix below.

		Predicted	
		Positive	Negative
Actual	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

The relevance of sensitivity and specificity in a default prediction model is related to the fact that the misclassification of a company can occur in two ways. If the predicted class of a defaulting client is non-default, the main cost for the bank is the loss of interest and capital. On the other hand, when the model classifies a non-defaulting customer as default, the bank faces the opportunity cost of not lending to a non-defaulting client, which is a lost business opportunity. The cost of the former (i.e., a false negative) is typically higher for a bank.

In the literature, it is well known (see, e.g., Dopuch et al. 1987; Koh 1992; Nanda and Pendharkar 2001) that incorporating sensitivity and specificity into the models can lead to more accurate predictions, especially when the two types of error imply different costs. Hence, for decision-making purposes, if a bank can estimate the cost of Type I and Type II errors, sensitivity and specificity measures are often more important than accuracy.

Threshold adjustment

In the presence of balanced classes, standard classification theory proves that it is best to predict Default according to (1), i.e., when $P(y = 1|\boldsymbol{x}) > 0.5$. However, in credit risk management applications, where the classes are imbalanced because of few "positive" events (defaults), this way of proceeding yields a low sensitivity (Kuhn and Johnson 2013). Hence, it is essential to seek a threshold different from 0.5, obtained by maximizing some function of sensitivity and specificity, since they cannot be maximized simultaneously. It would also be possible to use over- and under-sampling techniques: this approach is more common in other fields, but see Hussin Adam Khatir and Bee (2022) for an application to credit risk.

Here, we use the Geometric Mean (G-Mean) of sensitivity and specificity as a metric that evaluates the balance between the classification performances in the majority and minority classes. A low G-Mean value often denotes a poor performance in the classification of positive cases (Akosa 2017), which is an issue that should be avoided in a credit risk setup. Hence, the rationale of the G-mean approach is to find the value of the threshold $\gamma \in [0, 1]$ that balances sensitivity and specificity or, in other words, avoid overfitting the negative class and underfitting the positive class. In a binary classification setup, the threshold $\hat{\gamma}$ is formally defined as follows (Wald et al. 2013):

$$\hat{\gamma} = \arg\max_{\gamma \in [0,1]} \sqrt{\text{Sensitivity} \cdot \text{Specificity}}.$$

Accordingly, the classification rule (1) becomes

$$\hat{y}_i = \begin{cases} 1 & \text{if } P(y_i = 1|\boldsymbol{x}) > \hat{\gamma}; \\ 0 & \text{if } P(y_i = 1|\boldsymbol{x}) \leq \hat{\gamma}. \end{cases}$$

Feature-selection techniques

Feature selection (or *variable elimination*) is the process of determining relevant features for prediction purposes. It is important from various points of view, such as interpreting data, reducing the computational burden, avoiding the curse of dimensionality, and improving prediction accuracy (Chandrashekar and Sahin 2014). Here, we review some methods employed in "Empirical analysis" section to select the most important predictors. Note that, even though two of the methods presented in the following three sections exploit some specific classifier (random forest recursive feature elimination is based on RFs and the L_1-based approach uses support vector machines), all of these approaches are "self-contained." Hence, after selecting the features using any of the approaches described in "Random forest recursive feature elimination," "Chi-squared feature selection," "Feature selection with L_1 support vector machines" sections, it is possible to perform the classification task via any other algorithm.

Algorithm 2 RFRE

```
1. Train the RF classifier with all $d$ features.
2. Measure the performance and rank the predictors by importance.
3. For each subset size $m$ ($m = 1, \dots, d$), repeat the following steps:
    (a) Train the classifier with only the $m$ most important predictors.
    (b) Compute the performance and rank the $m$ predictors using the
        Gini index's mean reduction.
4. Use the classifier with the optimal number of predictors $m^*$, corresponding
   to the highest performance at Step 3(b) above.
```

Random forest recursive feature elimination

Random Forest Recursive Feature Elimination (RFRE; Zhou et al. 2014; Gregorutti et al. 2016) is based on the use of the RF classifier of ”Tree-based methods” section. It ranks the features by iteratively measuring the classifier performance and eliminating predictors accordingly. RFRE starts by training the classifier with all d features and calculating the importance of each feature via the information gain method or the mean reduction in the Gini index (James et al. 2021, p. 336; Ustebay et al. 2018). Subsequently, subsets of predictors of progressively smaller sizes $m = d, d-1, \dots, 1$ are obtained by iterative elimination. Within each subset, the model is retrained, and its accuracy is recomputed. Hence, RFRE is a feature-selection method that exploits the mean reduction in the Gini index in a random forest framework, as outlined in Algorithm 2 (see Ustebay et al. 2018 for details).

Chi-squared feature selection

In this approach, we employ the well-known chi-squared test of independence to assess the null hypothesis of independence between the category label and each feature (Alshaer et al. 2021). Since continuous features must be discretized, denote with m_s the number of classes of the sth predictor. The d tests are given by

$$\chi_s^2 = \sum_{i=1}^{m_s} \sum_{j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad s = 1, \dots, d, \tag{3}$$

where O and E are the observed and expected frequency, respectively, and k is the number of classes of the target variable.

Large values of χ_s^2 correspond to solid evidence against the null hypothesis of independence of y and x_s; since the distribution of (3) under the null is $\chi^2_{(k-1)(m_s-1)}$, the critical values are given by the quantiles of this distribution. We discard from the model the predictors x_s for which independence from y cannot be rejected.

Feature selection with L_1 support vector machines

The feature-selection method presented in this section is based on support vector machines (SVM—James et al. 2021, Chap. 9). SVMs map the data into a high-dimensional space and find an optimal separating hyperplane, obtained using *kernel* functions. As pointed out by Zhu et al. (2003), the most common version is based on the L_2 norm, but L_1 norm-based SVM may have some advantages. Expanding on this remark, Brankl et al. (2002) propose a feature-selection technique that exploits L_1-based SVM.

When the kernels are linear, i.e., $K(\boldsymbol{x}, \boldsymbol{v}) = \boldsymbol{x}^T\boldsymbol{v}$, the L_1-based SVM prediction of a new observation $\boldsymbol{x}$ can formally be written as $\mathrm{pred}(\boldsymbol{x}) = \mathrm{sign}(w_0 + \sum_{i=1}^n w_i K(\boldsymbol{x}, \boldsymbol{x}_i))$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, $i = 1, \dots, n$ are the predictors in the training set, and $\boldsymbol{w}$ is a vector of weights that can be computed explicitly as follows (Brankl et al. 2002; Zhu et al. 2003):

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \left[1 - y_i \left(w_0 + \sum_{j=1}^{n} w_j K(\boldsymbol{x}, \boldsymbol{x}_i) \right) \right]_+$$

with the constraint $||\boldsymbol{w}||_1 \le s$, where $[x]_+ \overset{\text{def}}{=} \max\{x, 0\}$.

Given a new observation $(y^*, \boldsymbol{x}^*)$, the SVM algorithm checks whether the linear combination $w_1 x_1^* + \cdots + w_d x_d^*$ is larger or smaller than $-\beta_0$, and classifies $(y^*, \boldsymbol{x}^*)$ accordingly (Brankl et al. 2002). The j-th feature is more likely to be important if the absolute value of its weight w_j is large, since it will likely drive

$w_1 x_1^* + \cdots + w_d x_d^*$ well above (or below) the threshold. Hence, this feature-selection method retains the features whose absolute weights $|w_j|$ are large. This type of feature weighting is rather intuitive: a predictor with a small $|w_j|$ has a smaller impact on the predictions and can be ignored; see Sindhwani et al. (2001) for a theoretical justification.

Empirical analysis

Dataset description

The data source for this study is the AIDA (Analisi Informatizzata Delle Aziende Italiane, `https://aida.bvdinfo.com`) database, which contains accounting records and financial ratios of all Italian companies required to file their accounts. The total number of observations is about one million. We consider various financial indicators for 2014–2018, focusing on small companies: according to the definition given by the European Commission (`https://single-market-economy.ec.europa.eu/smes/sme-definition_en`), a company is small if its total annual revenue is no larger than 10 million Euro.

For each company, the candidate predictors in the dataset can be grouped into four categories:

- Financial characteristics (balance sheet and income statement);
- Dimensionality, measured by the total assets discretized in three classes: Small (companies with total assets from the minimum to the first quartile), Medium (total assets between the first quartile and the mean), and Big (total assets between the mean and the maximum);
- Geographical area, identified by the region (North, Center and South);
- Type, which can be Limited Liability (*Società a Responsabilità Limitata* - S.R.L.) or Public Limited Companies by share (*Società Per Azioni* - S.P.A.).

In addition, each company's status (Default or Active) is described by a binary target variable. According to Ciampi and Gordini (2013), default is defined as the beginning of a legal proceeding for debt, such as bankruptcy or liquidation. Based on the information reported on AIDA's website, the default flag corresponds to one of liquidation, bankruptcy, voluntary liquidation, or compulsory administrative liquidation. Table 1 details the features; note that "Solvency" in AIDA is measured by the "Solvency ratio," given by "Shareholder's funds divided by Total assets."

The full dataset contains 35,081 Italian small companies operating in Italy in different sectors. However, the financial ratios of small companies may be contaminated by missing values and outliers, also because many of these companies are

Table 1 Description of the features in the dataset

No	Variable name	Description
1	Region	Geographical location of the company
2	Liquidity ratio	(Total current assets − Total inventories)/Payable due within 12 months
3	Current liability/Total asset	Payables due within 12 months/(Payables due within 12 months + Payables due beyond 12 months)
4	Leverage	Total assets/Shareholder's funds
5	Interest/turnover	(Total financial charges/(Revenues from sales and services + Other revenues)*100
6	Solvency	(Shareholder's funds/Total assets)*100
7	EBITDA/Sales	(Operating margin + Depreciation, amortization and writedowns)/(Revenues from sales and services + Other revenues)*100
8	Company size	Company's total amount of asset
9	ROA	(Operating margin/Total assets)*100
10	Company type	Type of company (S.R.L. or S.P.A.)
11	Net working capital	Total current assets − Payables due within 12 months

characterized by weak financial stability. Thus, we must clean the data to build a more stable and accurate default prediction model. Descriptive statistics are displayed in Table 2. Note that Company size and Net working capital are missing because they have become categorical after discretization.

As for missing data, every observation $\boldsymbol{x}$ containing one or more missing values is dropped from the dataset. As concerns outliers, we employ the classical box-plot approach based on the interquartile range (IQR) to detect and remove outliers: all data points that lie below $Q_1 - 1.5 \cdot IQR$ or above $Q_3 + 1.5 \cdot IQR$ (where Q_i, $i = 1, 3$, is the i-th quartile) are considered to be outliers and discarded. Various cut-offs have been tried in the box-plot method. For example, instead of $Q_1 - 1.5 \cdot IQR$, $Q_3 + 1.5 \cdot IQR$, we have replaced Q_1 and Q_3 with the 10th and 90th percentile, or with the 15th and 85th percentile, but the outcomes obtained with Q_1 and Q_3 seem preferable, since they give higher accuracy without significantly increasing the number of observations discarded.

Another possibility would be to cap outliers, but we prefer to avoid this solution since it can introduce bias into the analysis and make the data distribution more skewed, especially if the number of outliers is large.

After performing these two steps, the sample size reduces to 17,973 companies. Among them, 2660 defaulted, and the remaining 15,313 did not. Hence, the percentage of defaulted companies is 14.8%, so we face a class-imbalance problem, albeit less strong than in similar credit risk applications based on different datasets: for example, the default rate of Italian non-financial firms in the period 2011–2017 was always smaller than 5% (Moscatelli et al. 2019, Table 1).

Selecting the features

The selection of the financial ratios employed in the analysis is based on two stages. In the first step, we choose candidate features according to the results of previous investigations in the bankruptcy prediction literature (Beaver et al. 1967; Altman 1968; Blum 1974; Altman and Sabato 2005; Altman et al. 2010) and their known ability to measure the firm performance in terms of liquidity, profitability, and solvency. According to these criteria, nine financial variables are considered. Moreover,

Table 2 Descriptive statistics

	Liq. ratio	CL/TA	Lever.	Interest/Turn.	Solv.	EBITDA/Sales	ROA
Mean	1.65	0.83	5.01	0.41	29.07	5.97	6.03
Std	0.65	0.18	3.48	0.45	17.39	5.17	6.08
Min	0.01	0.25	− 6.73	0.00	− 39.86	− 11.25	− 12.04
25%	1.19	0.70	2.45	0.06	15.10	2.47	2.03
50%	1.50	0.88	3.88	0.24	25.60	4.88	4.84
75%	2.00	1.00	6.56	0.60	40.54	8.65	9.49
Max	3.66	1.00	16.63	2.00	92.48	23.78	22.98

CL current liability, *TA* total assets, *lever* leverage, *turn* turnover, *Solv.* solvency

Table 3 Predictors selected by the three algorithms

No	Feature	Type	RFFE	Chi-squared	L_1-based
1	Region	Categorical	×	✓	×
2	Liquidity ratio	Numerical	✓	✓	×
3	Current liab/T.asset	Numerical	✓	✓	✓
4	Leverage	Numerical	✓	✓	×
5	Interest/turnover	Numerical	✓	✓	✓
6	Solvency	Numerical	✓	✓	×
7	EBITDA/sales	Numerical	✓	×	✓
8	ROA	Numerical	✓	×	✓
9	Net W Cap	Categorical	✓	✓	✓
10	Company size	Categorical	✓	✓	✓
11	Company type	Categorical	×	✓	×

as said in "Dataset description" section, we include the geographical area and the type as predictive variables.

In the second step, we use, in turn each of the three feature-selection algorithms introduced in "Feature selection techniques" section to find the variables with the largest discriminatory power. Table 3 lists the features yielded by the RFRE, Chi-squared and L_1-based support vector machine feature-selection method.

Figure 1 shows the mean decrease in the Gini index (James et al. 2021), the key quantity for variable selection in the RFRE approach.

Figure 2 displays the p values of the Chi-squared test, and the penultimate column of Table 3 highlights the selected features. Finally, the last column of the table lists the predictors included by the L_1-based SVM selection technique.

Comparing the sets of features selected via the first two methods shows that both techniques find seven predictors. The L_1-based support vector machine criterion selects six variables: all of them are also selected by RFRE, and four by the chi-squared method. Four variables (Current liability/total assets, interest/turnover, Company size, and Net working capital) are always selected. Hence, the relevant predictors' choice seems relatively robust with respect to the feature-selection algorithms.

Implementation details

We employ all combinations of the five classifiers introduced in "Machine learning techniques" section and the three feature-selection techniques presented in "Feature selection techniques" section. However, since the accuracies obtained with the three feature-selection techniques are very similar, here we only show the outcomes based on the RFRE feature-selection approach, which is slightly better than Chi-squared and L_1.

To train the models and compare the performance of the classifiers, each of the four datasets (aggregate, North, Center, South) is randomly split into a training and a

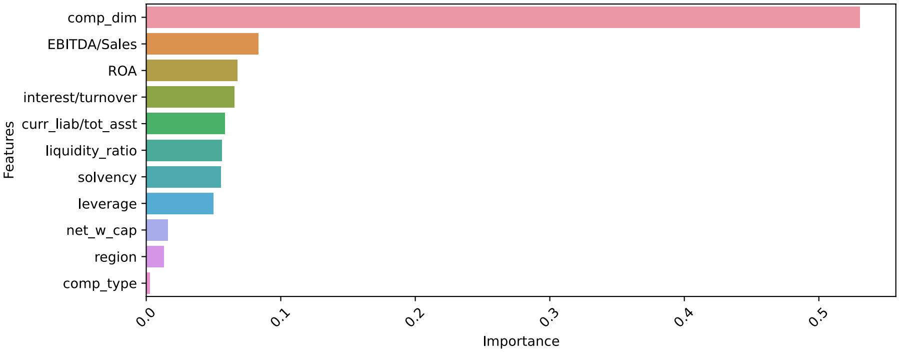

Fig. 1 Mean decrease in Gini index in Random Forest Recursive Feature Elimination

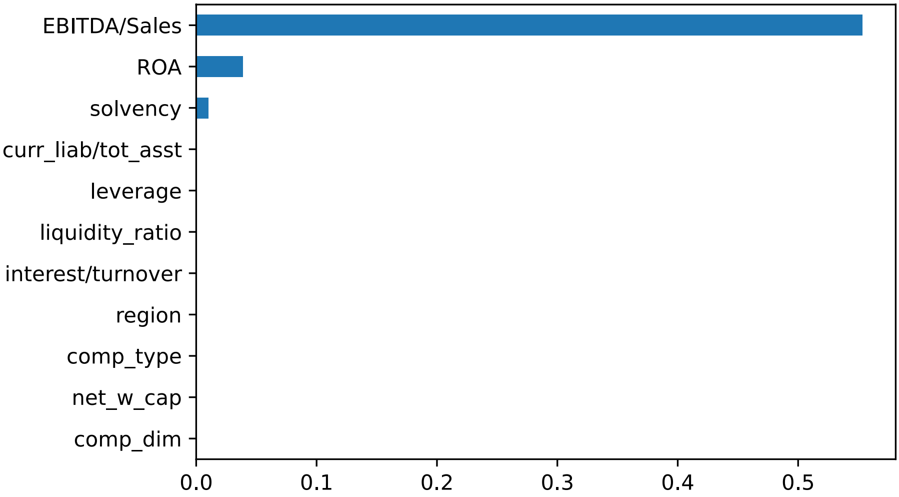

Fig. 2 *p* Values of the predictors in Chi-squared Feature Selection

test set containing 75% and 25% of the observations, respectively. The classifiers are estimated in the former, whereas the latter is used to test prediction accuracy. The implementation is based on the following packages of the `Python` programming language: `Pandas` for data manipulation, `Matplotlib` and `Seaborn` for data visualization, and `Scikit-learn` for data preprocessing and model fitting.

The numerical values of the main inputs of the classifiers have been found via k-fold cross-validation (CV) with $k = 5$. For RF, NN, and LR, the cross-validated values are equal to the default ones used by the `Python` commands; for DT and KNN, some of them are different. In any case, for all algorithms, the numerical values are reported below.

In DT, node-splitting is performed according to node purity, measured by the Gini index. The minimum number of samples required to split an internal node and to be at a leaf node equals 74 and 17, respectively. These values are found by CV and are different from the default ones.

For random forests, at each node, the mean decrease in the Gini index measures the quality of a split. The number of trees in the forest is 100, and the number of predictors employed as split candidates is equal to $\sqrt{d}$.

The neural network uses one hidden layer with 100 hidden units, and the activation function is ReLU. The optimization is performed using the `adam` algorithm (Kingma and Ba 2014), which works well on relatively large datasets. To avoid overfitting, we employ L_2 regularization with $\alpha = 0.0001$, which is the default value, double-checked via CV.

In regularized logistic regression, the norm of the penalty is L_2 and the log-likelihood is maximized using the L-BFGS algorithm. In the subsequent analysis, all the parameters are significantly different from zero at the 5% level.

The KNN algorithm employs a number of neighbors $K = 13$, different from the default value $K = 5$. This choice results from CV, according to the procedure illustrated in James et al. (2021, Sect. 5.1.5).

Classification results

This section shows the outcomes obtained using the classifiers presented in "Methodology" section combined with the RFRE feature-selection technique.

The ROC curves corresponding to all classifiers are shown in Figs. 3, 4, 5, and 6 for aggregate data, Northern Italy, Central Italy, and Southern Italy, respectively. The big picture represented by these graphs suggests that, in each dataset, there is little difference between the methods. Similarly, the geographical area does not significantly affect classification accuracy. More insight is provided by the accuracy measures displayed in Tables 4, 5, 6, and 7.

When considering the entire dataset, Table 4 shows that RF and NN achieve the best performance in terms of accuracy and AUC. However, the remaining classifiers produce similar results, with accuracies larger than 95%. As for sensitivity and specificity, the latter is always higher than the former, probably because of the class imbalance: since defaulters are fewer than non-defaulters, the algorithms tend to favor the majority class. It is worth noting that, in terms of sensitivity and specificity, LR performs extremely well.

The results of this section allow us to notice that the Random Forest model has the highest AUC, except in Northern and Central Italy, where NN is preferable. In terms of AUC, the worst classifier is a decision tree. Second, focusing on logistic regression, its accuracy is only marginally (0.12% to 0.83%) lower than the best classifier's accuracy, always with the second or third largest accuracy.

Overall, there are no significant differences between aggregate and regional level results. The reason is likely to be that we are assessing the probability of default, which measures the likelihood of a firm failing to meet its financial obligations and is primarily determined by factors such as the firm's financial health, the industry conditions, and the market dynamics. The regionality of a firm, referring to its

Table 4 Accuracy measures for aggregate data

Model	Accuracy	Sensitivity	Specificity	AUC
RF	0.958	0.772	0.992	0.972
DT	0.952	0.765	0.986	0.951
KNN	0.952	0.729	0.993	0.957
NN	0.960	0.804	0.989	0.970
LR	0.956	0.744	0.994	0.956

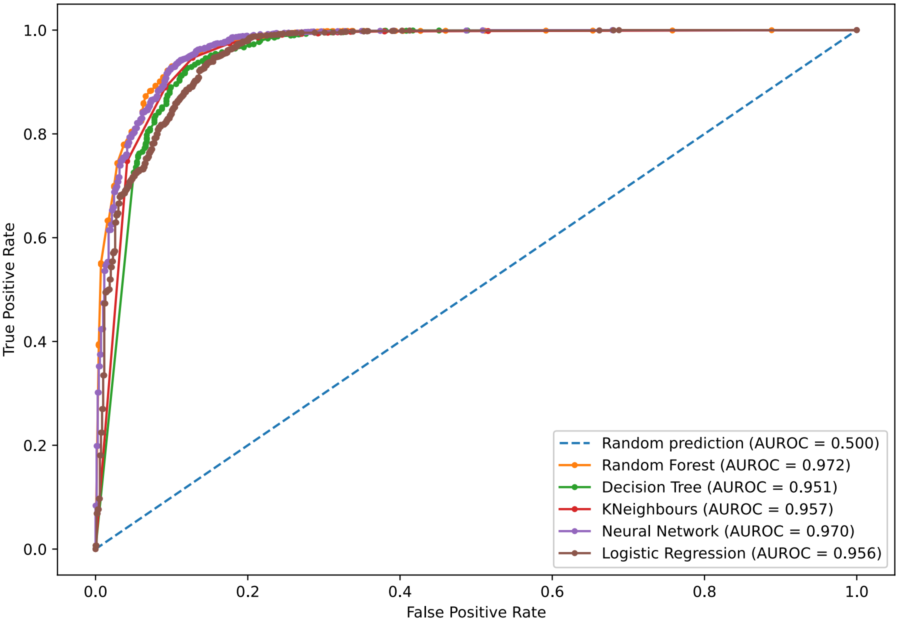

Fig. 3 Aggregate data: ROC curves for all classifiers

geographic location or markets, does probably not directly impact this probability. Factors like firm's creditworthiness, market conditions, diversification strategies, and risk management practices are more influential in assessing credit risk. It is worth noting that, while the regionality of a firm may have little effect on the probability of default, regional factors can indirectly influence a firm's financial health and credit risk. For example, a severe economic downturn or political instability in a specific region may affect the firm's overall performance and increase its credit risk. However, these effects are typically incorporated into the broader assessment of creditworthiness and are not solely determined by regionality.

Even though the values of the AUC are quite large, there are at least two reasons why we do not think they are caused by overfitting. First, the AUC and the accuracy evaluation measures (accuracy, sensitivity, and specificity) are computed in the

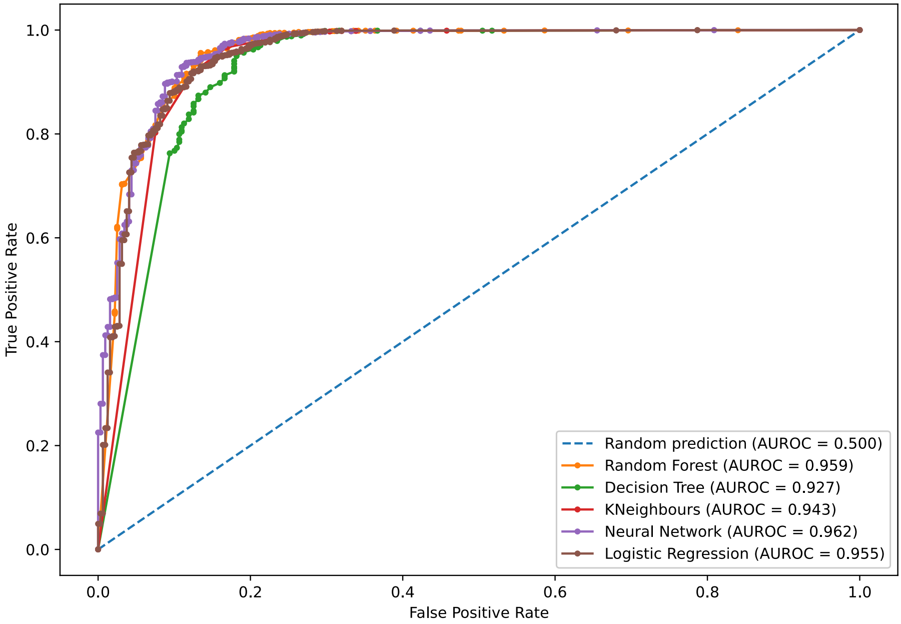

Fig. 4 Northern Italy: ROC curves for all classifiers

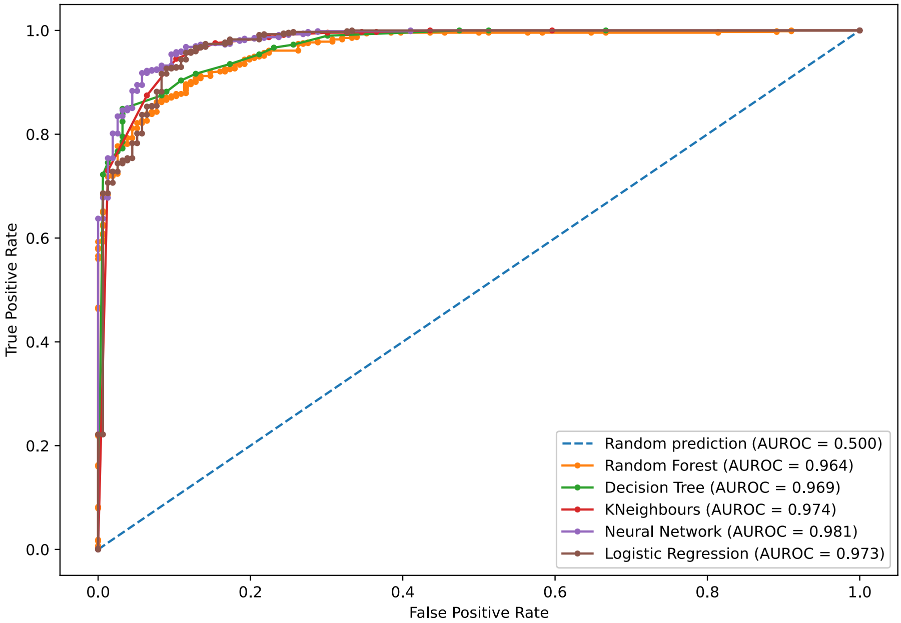

Fig. 5 Central Italy: ROC curves for all classifiers

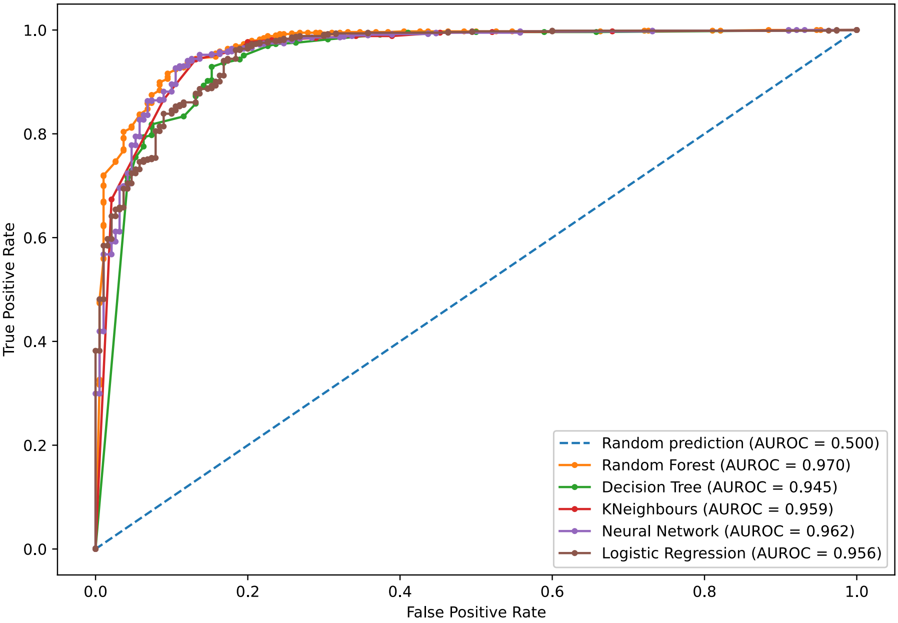

Fig. 6 Southern Italy: ROC curves for all classifiers

Table 5 Accuracy measures for companies in Northern Italy

Models	Accuracy	Sensitivity	Specificity	AUC
RF	0.966	0.743	0.995	0.959
DT	0.960	0.746	0.988	0.927
KNN	0.965	0.718	0.996	0.943
NN	0.966	0.771	0.991	0.962
LR	0.964	0.715	0.996	0.955

Table 6 Accuracy measures for companies in Central Italy

Models	Accuracy	Sensitivity	Specificity	AUC
RF	0.928	0.628	0.996	0.964
DT	0.931	0.769	0.967	0.969
KNN	0.948	0.756	0.991	0.974
NN	0.951	0.814	0.981	0.981
LR	0.952	0.788	0.988	0.973

test set. Second, all the methods are implemented through regularization and/or the numerical values of their inputs are double-checked via cross-validation.

Table 7 Accuracy measures for companies in Southern Italy

Models	Accuracy	Sensitivity	Specificity	AUC
RF	0.942	0.795	0.978	0.970
DT	0.931	0.774	0.969	0.945
KNN	0.934	0.732	0.983	0.959
NN	0.934	0.795	0.968	0.962
LR	0.937	0.758	0.981	0.956

Table 8 G-mean approach: accuracy measures for aggregate data

Models	Accuracy	Sensitivity	Specificity	AUC	Threshold
RF	0.930	0.937	0.891	0.972	0.18
DT	0.927	0.944	0.837	0.960	0.25
KNN	0.875	0.866	0.921	0.945	0.18
NN	0.945	0.958	0.871	0.970	0.23
LR	0.932	0.942	0.842	0.956	0.21

Table 9 G-mean approach: accuracy measures for companies in Northern Italy

Models	Accuracy	Sensitivity	Specificity	AUC	Threshold
RF	0.947	0.961	0.847	0.958	0.20
DT	0.924	0.932	0.868	0.948	0.14
KNN	0.893	0.896	0.871	0.922	0.18
NN	0.956	0.969	0.859	0.959	0.16
LR	0.934	0.946	0.847	0.952	0.15

Using the adjusted threshold

In "Classification results" section, we have classified as Default the companies with estimated posterior probability of default larger than 50%. However, since the percentage of defaulted companies in our data is 14.8%, we now focus on the approach introduced in "Threshold adjustment" section, which allows us to consider the class-imbalance issue.

The results obtained with the adjusted cutoff $\hat{\gamma}$ given by the G-mean method of "Threshold adjustment" section are shown in Tables 8, 9, 10, and 11. Only the accuracy measures corresponding to the predictors selected by RFRE are displayed here; the remaining results are reported in the online appendix.

As can be seen from the tables, the adjusted threshold is considerably smaller than 0.5, as expected, because the Default class is underrepresented. Whereas the numerical results are similar across all datasets, i.e., aggregate and regional data, there are some significant differences with respect to the outcomes in "Classification results" section.

Table 10 G-mean approach: accuracy measures for companies in Central Italy

Models	Accuracy	Sensitivity	Specificity	AUC	Threshold
RF	0.910	0.904	0.935	0.971	0.23
DT	0.927	0.948	0.832	0.960	0.32
KNN	0.833	0.809	0.942	0.952	0.18
NN	0.923	0.926	0.910	0.973	0.22
LR	0.927	0.936	0.884	0.971	0.23

Table 11 G-mean approach: accuracy measures for companies in Southern Italy

Models	Accuracy	Sensitivity	Specificity	AUC	Threshold
RF	0.913	0.921	0.876	0.953	0.24
DT	0.860	0.857	0.870	0.925	0.23
KNN	0.839	0.826	0.892	0.917	0.18
NN	0.919	0.925	0.892	0.962	0.19
LR	0.910	0.923	0.854	0.951	0.19

1. The threshold is mostly between 15% and 25%, which is close to the percentage of defaulting companies in the dataset, equal to 14.8%.
2. The average decrease in accuracy is small (approximately 1–3%) for most methods; only KNN's accuracy decreases more significantly;
3. Sensitivity increases considerably at the price of a small decrease in specificity. This means that the use of a smaller threshold reduces the false negatives. For a bank, this is probably the most important outcome.

Conclusion

In this paper, we shed some light on the performance of the most widespread ML classifiers compared to the classical logistic regression approach. The empirical results convey the following main messages.

Among the ML methods, neural networks and random forests are preferable. However, logistic regression has a very close classification accuracy. More explicitly, our findings suggest to rank the methods as follows: neural networks and random forests are first, with a similar performance, logistic regression follows at a short distance, and the remaining methods are one step behind.

These remarks remain true also when the analysis is based on the adjusted threshold found by maximizing the geometric mean of sensitivity and specificity. Such a threshold is significantly smaller than 50%, and, as pointed out in the previous section, the sensitivity associated with the adjusted threshold is considerably larger. This is an important result, as fewer defaulting customers are erroneously classified as non-defaulting when using the adjusted threshold. Accordingly, the take-home message is that, due to class imbalance, one should classify observations using a threshold chosen to increase sensitivity.

The drivers of the overperformance of neural networks and random forests are hard to identify. Generally speaking, if the most flexible methods such as NN and RF are better, the reason is that the relationships between the target variable (i.e., the default indicator) and the predictors are "highly nonlinear." However, investigating such relationships, even with a moderate number of predictors, is quite difficult. Moreover, since the fitted models are also difficult to interpret, they do not shed much light on the relative importance of the predictors. This is a common problem with advanced ML techniques: the results are often very good, but it is difficult to understand thoroughly why ML approaches perform better.

From the point of view of a bank that needs to decide about the practical implementation of these methods, the rather modest gain of ML-based classifiers with respect to LR suggests the following conclusions.

- In principle, it is worth implementing both LR and the most efficient ML classifiers, i.e., random forests and neural networks; running the models side by side would also make it possible to double-check the robustness of the algorithms.
- LR has a lighter computational burden and a more straightforward implementation, as it does not require to pre-set numerical values of input parameters (such as, for example, the number of neighbors K in KNN, or the minimum number of observations at each node in DT). In addition, interpreting the results is easier, in particular as concerns the relationship of each predictor to the target variable.
- If regulators must scrutinize default prediction models, traditional statistical approaches such as LR may be less problematic since they have been employed and tested in the industry for a long time and are less dependent on the discretionary interventions of the users of the models.
- Finally, LR is more interpretable and allows the final user to assess quantitatively both the relative importance of the explanatory variables and the impact of each predictor on the target variable, via the logit transformation (see, e.g., James et al. 2021, Sect. 4.3.4).

Since ML methods tend to perform better and better when the sample size increases, the previous remarks may become even more meaningful when the databases are smaller than in the present work.

It may be interesting to perform further analyses based on these methods: in particular, it would be important to study whether the results are stable across different sectors.

Another issue that would be worth exploring is the development of a model evaluation procedure that takes into account the number of predictors used by a model: for example, the L_1-based feature-selection technique yields only six predictors, i.e., a more parsimonious model with respect to the ones based on the RFRE or Chi-squared feature-selection techniques. Finding a way of incorporating this parsimony into some model evaluation procedure would be quite useful.

Acknowledgements We would like to thank the associate editor and an anonymous reviewer for valuable comments that have considerably improved a preliminary version of the paper.

Funding Open access funding provided by Università degli Studi di Trento within the CRUI-CARE Agreement.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Akosa, J. 2017. Predictive accuracy: A misleading performance measure for highly imbalanced data. In *Proceedings of the SAS global forum*, Paper 942.

Alshaer, H., M.A. Otair, L. Abualigah, M. Alshinwan, and A. Khasawneh. 2021. Feature selection method using improved Chi Square on Arabic text classifiers: Analysis and application. *Multimedia Tools and Applications* 80 (7): 10373–10390.

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., and G. Sabato. 2005. Effects of the new Basel capital accord on bank capital requirements for SMEs. *Journal of Financial Services Research* 28 (1): 15–42.

Altman, E.I., G. Sabato, and N. Wilson. 2010. The value of non-financial information in SME risk management. *Journal of Credit Risk* 6: 95–127.

Bacham, D., and J. Zhao. 2017. *Machine learning: Challenges, lessons, and opportunities in credit risk modeling*. IX: Moody's analytic perspectives.

Bank of England, Financial Conduct Authority. 2019. Machine learning in UK financial services.

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

Beaver, W., S. Wallenstein, R. Houde, and A. Rogers. 1967. A clinical comparison of the analgesie effects of methadone and morphine administered intramuscularly, and of orally and parenterally administered methadone. *Clinical Pharmacology & Therapeutics* 8 (3): 415–426.

Blum, M. 1974. Failing company discriminant analysis. *Journal of Accounting Research* 12: 1–25.

Bolder, D. 2019. *Credit-risk modelling: Theoretical foundations, diagnostic tools, practical examples, and numerical recipes in Python*. New York: Springer.

Brankl, J., M. Grobelnikl, N. Milić-Frayling, and D. Mladenić. 2002. Feature selection using support vector machines. In *Data mining III*, ed. A. Zanasi, C. Brebbia, N. Ebecken, and P. Melli. Southampton: WIT Press.

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

Breiman, L., J. Friedman, C. Stone, and R. Olshen. 1984. *Classification and regression trees*. Boca Raton: Chapman and Hall.

Carton, R.B., and C.W. Hofer. 2006. *Measuring organizational performance: Metrics for entrepreneurship and strategic management research*. Northampton: Edward Elgar Publishing.

Chandrashekar, G., and F. Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40 (1): 16–28.

Ciampi, F., and N. Gordini. 2013. Small enterprise default prediction modeling through artificial neural networks: An empirical analysis of Italian small enterprises. *Journal of Small Business Management* 51 (1): 23–45.

Dopuch, N., R.W. Holthausen, and R.W. Leftwich. 1987. Predicting audit qualifications with financial and market variables. *Accounting Review* 62: 431–454.

Duffie, D., and K.J. Singleton. 2003. *Credit risk: Pricing, measurement, and management*. Princeton: Princeton University Press.

Dumitrescu, E., S. Hué, C. Hurlin, and S. Tokpavi. 2022. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research* 297 (3): 1178–1192.

Gregorutti, B., B. Michel, and P. Saint-Pierre. 2016. Correlation and variable importance in random forests. *Statistics and Computing* 27 (3): 659–678.

Hussin Adam Khatir, A.A., and M. Bee. 2022. Machine learning models and data-balancing techniques for credit scoring: What is the best combination? *Risks* 10 (9): 169.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2021. *An introduction to statistical learning*, 2nd ed. New York: Springer.

Kingma, D., and J. Ba. 2014. Adam: A method for stochastic optimization. In *3rd international conference on learning representations*.

Koh, H.C. 1992. The sensitivity of optimal cutoff points to misclassification costs of type I and type II errors in the going-concern prediction context. *Journal of Business Finance & Accounting* 19 (2): 187–197.

Kuhn, M., and K. Johnson. 2013. *Applied predictive modeling*. New York: Springer.

Le, H.H., and J.-L. Viviani. 2018. Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios. *Research in International Business and Finance* 44: 16–25.

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

Merton, R. 1974. On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance* 29 (2): 449–470.

Moscatelli, M., S. Narizzano, F. Parlapiano, and G. Viggiano. 2019. Corporate default forecasting with machine learning. Temi di discussione, 1256.

Nanda, S., and P. Pendharkar. 2001. Linear models for minimizing misclassification costs in bankruptcy prediction. *Intelligent Systems in Accounting, Finance & Management* 10 (3): 155–168.

Shi, S., R. Tse, W. Luo, S. D'Addona, and G. Pau. 2022. Machine learning-driven credit risk: A systemic review. *Neural Computing and Applications* 34: 14327–14339.

Sindhwani, V., P. Bhattacharya, and S. Rakshit. 2001. Information theoretic feature crediting in multiclass support vector machines. In *Proceedings of the 2001 SIAM international conference on data mining*, 1–18. SIAM.

Ustebay, S., Z. Turgut, and M. Aydin. 2018. Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In *2018 International congress on big data, deep learning and fighting cyber terrorism*, 71–76. IEEE.

van Liebergen, B. 2017. Machine learning: A revolution in risk management and compliance? *Journal of Financial Transformation* 45: 60–67.

Vidovic, L., and L. Yue. 2020. Machine learning and credit risk modelling. Technical report, Standard & Poor's.

Wald, R., T. Khoshgoftaar, and A. Napolitano. 2013. The importance of performance metrics within wrapper feature selection. In *2013 IEEE 14th international conference on information reuse & integration*, 105–111. IEEE.

Zhou, Q., H. Zhou, Q. Zhou, F. Yang, and L. Luo. 2014. Structure damage detection based on random forest recursive feature elimination. *Mechanical Systems and Signal Processing* 46 (1): 82–90.

Zhu, J., S. Rosset, R. Tibshirani, and T. Hastie. 2003. 1-Norm support vector machines. In *Advances in neural information processing systems*, vol. 16, ed. S. Thrun, L. Saul, and B. Schölkopf. Cambridge: MIT Press.