International Journal of Combinatorial Optimization Problems and Informatics, 13(2), May-Aug 2022, 88–97. ISSN: 2007-1558

www.editada.org

Endogenous Prediction of Bankruptcy using a Support Vector Machinevector

Jorge Zazueta Gutierrez[1]*, *Andrea Chavez Heredia*[2], *Jorge Zazueta Hernández*[2]
[1] School of Economics, Universidad Autónoma de San Luis Potosí
[2] Department of Mathematics, Universidad de Guanajuato
*Correspondence: jorge.zazueta@uaslp.mx

Abstract. We build a global bankruptcy prediction model using a support vector machine trained only on firms' endogenous information in the form of financial ratios. The model is tested not only on entirely random unseen data but on samples taken from specific global regions and industries to test for prediction bias, achieving satisfactory prediction performance in all cases. While support vector machines are not easily interpretable, we explore variable importance and find it consistent with economic intuition.

Keywords: Bankruptcy, Support Vector Machines, Global Model

Article Info

Received: August 10, 2021
Accepted: January 3, 2022

1 Introduction

Several attempts have been made to create bankruptcy prediction models with a good measure of success. Accuracy ranges from 73% to 99% [1]. However, most of these models were based on data from a specific country or region. As Alaminos [2] reports, world financial crises leading to increasing bankruptcies in several countries have brought attention to the study of bankruptcy in an increasingly international context. Naturally, this beckons for models that are more global, or less country specific.

Following Alaminos' analysis of the State of the Art, an increasingly international financial and economic cultures has come to imply an increasingly homogenous financial behavior [1][2]. Although the natural response to these observations would be the development of explicitly global models of bankruptcy, much of the literature and study has ultimately focused on companies within a single country [3][4][5][6]. Working towards a more truly global approach, Alaminos [7] proposed a global model using logistic regression with data from three global regions (America, Europe, and Asia), reporting a test accuracy of 84.86% for his global model and 90.11% after adding dummy variables to account for regions. This approach is sensible given that the available feature space might not carry complete information, so the region variable acts as a control. However, by doing so, the model becomes, again, geography dependent.

An ideal global model should be independent of geography and rely solely on the company's endogenous information. As discussed, the nature of globalization would make it reasonable to expect a somewhat homogeneous global environment and financial practices, at least for publicly traded companies, making feasible and valuable an endogenous model. A characteristic of the models in the literature is the diversity in the predictive variables used, calling for a theoretical foundation of bankruptcy as an economic phenomenon. Endogenous model development informs theory by focusing on the firm's behavior rather than its environment.

In this paper, we build a prediction model that uses only internal data in the form of financial ratios and test its performance in specific regions and industries, aiming for high out-of-sample prediction accuracy. We are interested not only in observed accuracy on a general random test set but also in the same region or industry's subsamples. To do this, we rely on the same dataset used by [7], containing information of 468 companies in three continents.

2 Support Vector Machines

SVMs are highly flexible and robust classifiers. Although generally regarded as hard to interpret, variable importance can be measured by running ROC curve analysis on each predictor [8]. SVMs were first introduced in the computer science field by Vapnik [9] during the 1990s at AT&T Bell Laboratories. Generally speaking, the SVM can be thought of as a linear classifier in a high-dimensional feature space that is non-linearly related to input space and can thus be applied to linear and non-linear classification and regression problems.

The underlying idea behind the algorithm is that, given a training data set where every point belongs to one of two categories, we want to create a hyperplane that separates the data set, maximizing the width of the gap between both categories (See Figure 1). This is: in a 2-dimensional space, we want to be able to draw a line that clearly separates the data into two different sections. The hyperplane equation can classify new examples by assuming the same distribution as the original training, input, and data sets. When we can find a hyperplane that completely classifies the points in a data set, we say that the data is linearly separable. However, in many practical problems, finding such a hyperplane might not be feasible or possible. To deal with non-linearly separable sets (See Figure 2), we introduce the *kernel trick*, which implicitly maps the data into a feature space of higher dimension.

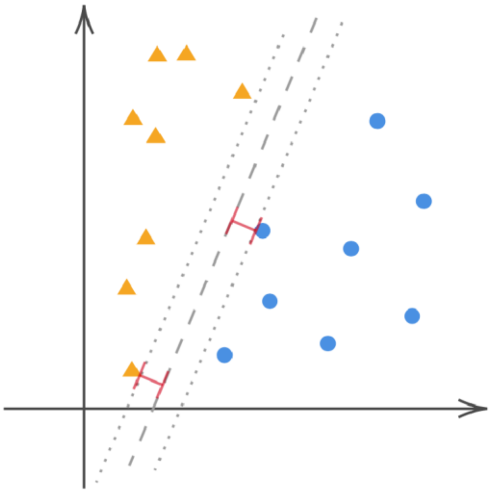

Fig. 1. A 1-hyperplane classifying two types of data in a 2-dimensional space

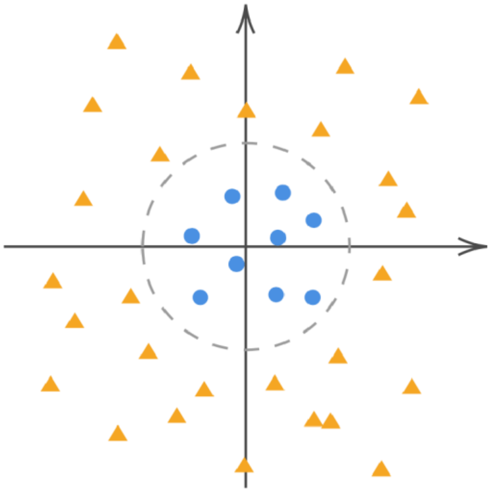

Fig. 2. A data set that a hyperplane cannot classify

2.1 Maximal Margin Classifier and Support Vector Classifiers

Given the data set $D=\{(x_i,y_i)\mid x_i\in\square^{p} \text{ and } y_i\in\{1,-1\}\}_{i=1}^{n}$,$\mathrm{D} = \{(\mathrm{x_i},\mathrm{y_i})|\mathrm{x_i} \in \mathrm{R^p andy_i} \in \{1,\text{-}1\}\}_{\mathrm{i=1}}^{\mathrm{n}}$, our objective is to know if we can separate its points with a $(p-1)$-dimensional hyperplane. This is what we call a *linear classifier*. Among the multiple hyperplanes that may classify the data, a natural choice is the separating hyperplane that is farthest from the training observations; this natural choice is known as the *maximal margin hyperplane* or optimal separating hyperplane. We can then classify a test observation based on which side of the maximal margin hyperplane it lies. This is known as a *maximal margin classifier*.

In practice, linearly separable sets are rare. S*upport vector classifiers* (SVCs) generalize the maximal margin classifier by allowing some points to be incorrectly classified, on the wrong side of the margin (See Figure 3). This is achieved by introducing slack variables in the underlying optimization problem. For this reason, SVCs are sometimes referred to as *soft margin classifiers*. By relaxing perfect separation, we obtain a more robust model that typically performs better at classifying out-of-sample points. Note that the hyperplane depends only on those data points that lie on the margin, called *support vectors*, making the classifier robust to new data that lies outside the margin.

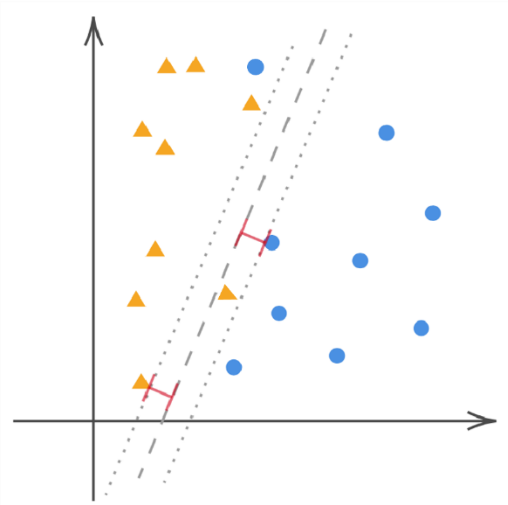

Fig. 3. A Support Vector Classifier allows for some points to cross the margin

2.2 Dealing with Non-Linearity

When a linear classification boundary is appropriate, SVC is a natural approach. However, it is common in practice to come across instances in which a non-linear boundary is necessary, as illustrated in Figure 2. The *support vector machine (SVM)* is an extension of the support vector classifier that results from mapping the input data into a high dimensional feature space via a non-linear transformation in order to perform the linear algorithm in the enlarged space, resulting in a non-linear boundary when projected back onto the original space.

Without delving into the details of the calculation, finding the maximal margin hyperplane is a quadratic programming problem that depends only on inner products between points in the feature space, as does the final decision function:

$$f(\mathbf{x}) = sign\left(\sum_i \alpha_i \left(\mathbf{x}\cdot\mathbf{x}_i + b\right)\right) \tag{1}$$

As we are working on the enlarged feature space, we substitute the inner product in Equation 2 by the function $K(\mathbf{x},\mathbf{y}) = \varphi(\mathbf{x})\cdot\varphi(\mathbf{y})$. Where φ is a (typically) non-linear function that maps the input space into the enlarged feature space. Due to the high dimensionality of the feature space, evaluating all the inner products can be computationally challenging. However, there are simple *kernels* that can be evaluated efficiently by performing operations directly on the input space.

In this paper, we will utilize a kernel commonly used by practitioners, the radial basis function (RBF) kernel:

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\sigma\left\|\mathbf{x}-\mathbf{y}\right\|^2\right) \tag{2}$$

Our specific decision function takes

$$f(\mathbf{x}) = sign\left(\sum_i \alpha_i \left(\exp\left(-\sigma\left\|\mathbf{x}-\mathbf{x}_i\right\|^2\right) + b\right)\right) \tag{3}$$

The parameters α_i and b are computed via the quadratic programming problem, and σ is a hyperparameter that needs to be tuned by the practitioner, along with a regularization cost hyperparameter *C* that is part of the optimization problem. We can see that when the Euclidean distance between a new observation and a training point becomes large, its corresponding component becomes very small, implying that the radial kernel has local behavior in the sense that only nearby training observations affect the class label. A detailed discussion of the algorithm can be found in [10][11] and [12].

2.3 Software and Tools

All our calculations are performed in R [13], with a heavy reliance on the *caret* package [14]. Tables were handled by flextable [15] and Graphs were created using *ggplot2* [16] and Mathcha [17].

3 Model Development

3.1 The Data

Table 1. Company distribution by country and region

Country	Asia	Europe	America	Total	Bankrupt
Japan	112			112	56
Korea	8			8	4
Singapore	2			2	1
Taiwan	2			2	1
Austria		6		6	3
Bermuda		2		2	1
Denmark		20		20	10
France		32		32	16
Germany		22		22	11
Ireland		2		2	1
Italy		8		8	4
Luxembourg		2		2	1
Netherlands		8		8	4
Norway		10		10	5
Poland		8		8	4
Portugal		2		2	1
Spain		2		2	1
Sweden		22		22	11
Switzerland		2		2	1
United Kingdom		26		26	13
United States of America			154	154	77
Canada			16	16	8
Total	124	174	170	468	234

In his paper, Alaminos [7] shared a global dataset including data from 468 publicly traded companies in 22 countries across Asia, Europe, and America from the period 1990-2013. Features include country, financial ratios (See Table 1), and industry code. Two hundred thirty-four of the companies in the sample legally declared bankruptcy. The rest of the sample was selected randomly from active companies following a match criterion by country, industry, and year. Financial ratios for each company correspond to the year prior to bankruptcy. The distribution of companies in the sample is summarized in Table 1. We will rely on this data to build and test our model. The overall methodology is depicted in Figure 4.

3.2 Data Preparation

Our dataset has a total of 73 missing values, distributed by feature as of Table 2. Note that the variables are categorical. On the other hand, we assume that the missing data are Missing at Random (MAR), which means that the probability that a value is missing depends only on the observed values and can be predicted using the latter. Subsequently, since the proportion of missing values is insignificant and we assume they are MAR, we impute the missing values using Multivariate Imputation by Chain Equations as implemented by the MICE package in R [18]. For numerical data, MICE uses predictive mean matching to impute each incomplete variable with a separate model.

Table 2. Number of missing data by feature

Ratio	Missing
Earnings/Total Assets	9
Current Assets/Current Liabilities	9
Working Capital/Total Assets	7
Retained Earnings/Total Assets	10
EBIT/Total Assets	6
Sales/Total Assets	2
(Current Assets + Cash Flow)/Current Liabilities	8
Total Debt/Total Assets	7
Current Assets/Total Assets	3
Earnings/Net Worth	12

From our imputed data, we generate a test set by randomly selecting 25% of the data and training a Support Vector Machine with a radial basis kernel with the remaining 75%. The training set is stripped from region and industry information to develop a model based solely on the ten financial ratios available and subsets of the test data by region and industry to assess model performance on specialized subsamples (i.e., only Asian countries or Consumer Discretionary focused companies).

3.3 Model Fitting

We recall, from [12] and [19], that the *k-fold cross validation* predictive method involves splitting the dataset into k-subsets. For each subset is held out while the model is trained on all other subsets. This process is completed until accuracy is determined for each instance in the dataset, and overall accuracy estimate is provided. In the case of the repeated *k-fold cross validation*, the process of splitting the data into k-folds can be repeated a number of times, therefore the name of “repeated”. Thus, the final model accuracy is taken as the mean from the number of repeats. For example, five repeats of 7-fold cross validation would give 35 total resamples that are averaged, but this is not the same as 35-fold cv.

To fit our SVM we use repeated 10-fold cross validation with ROC's AUC as a selection metric. For hyperparameter tuning we resort to the empirical observation that the optimal values of the parameter σ in RBF kernel lie between the .10 and .90 quantiles of the $\left\|\mathbf{x}-\mathbf{x}_i\right\|^2$ statistics [14]. The *sigest* function from *kernlab* uses a sample to estimate the quantiles and returns a vector with the .10, .50 and .90 quantiles. We run a grid search with these three values and cost, C, ranging from 2 to 200 in steps of 0.5. with scaled and centered data. Our results are shown in Figure 5. With the best model's hyperparameters being $C = 9.5$ and $\sigma = 0.1497$**.**

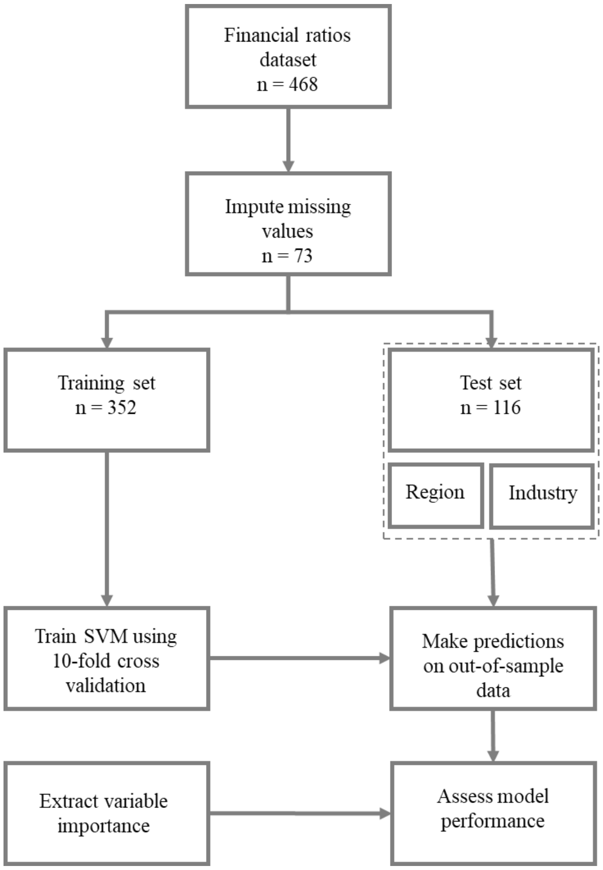
Financial ratios dataset n = 468
Impute missing values n = 73
Training set n = 352
Test set n = 116
Region
Industry
Train SVM using 10-fold cross validation
Make predictions on out-of-sample data
Extract variable importance
Assess model performance

Fig. 4. Model development methodology

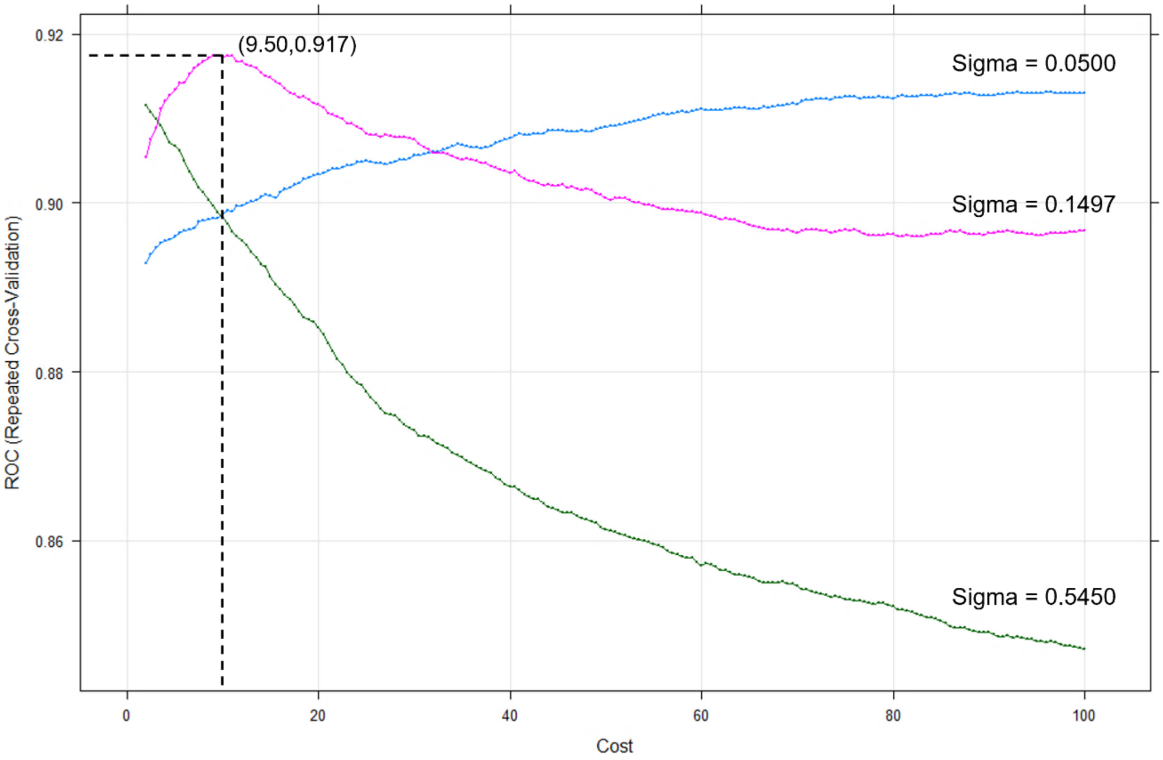

Fig. 5. Cost hyperparameter tuning

4 Results

4.1 Performance Metrics

The final model has an average cross validated accuracy of 0.842 and a percentual average confusion matrix shown in Table 3.

Table 3. Final Model Cross Validated Confusion Matrix

	Reference	
	Yes	No
Prediction	87.87%	12.13%
	22.82%	98.56%

Overall prediction performance on the test set is summarized in Table 4. Where the first four reported performance indicators are defined in terms of the confusion matrix elements as follows:

$$Accuracy = \frac{number\text{ of true positives + numer of true negatives}}{number\text{ of positvies + number of negatives}} \tag{4}$$

$$Sensitivity = \frac{\text{number of true positives}}{number\text{ of true positives + number of false negatives}} \tag{5}$$

$$Specificity = \frac{\text{number of true negatives}}{\text{number of true negatives + number of false positives}} \tag{6}$$

$$Kappa = \frac{\text{Accuracy - Expected Accuracy}}{1 - Expected\text{ Accuracy}} \tag{7}$$

And AUC is the area under the Receiver Operator Characteristic curve. Performance on the overall testing set is quite satisfactory, with accuracy of 0.88 and an AUC of 0.93.

Table 4. Overall, out-of-sample model performance

n	Accuracy	Sensitivity	Specificity	Kappa	AUC
116	0.888	0.879	0.897	0.776	0.934

The model also performs well on specific region, with similar performance (see Table 5).

From the training sample, we focused on industry subsets with more than ten elements, recovering Industrials, Consumer Discretionary and Information Technology. Model performance remains stable with similar scores (see Table 6).

Table 5. Model performance metrics by region

Region	n	Accuracy	Sensitivity	Specificity	Kappa	AUC
Asia	34	0.853	0.875	0.833	0.706	0.924
America	41	0.878	0.857	0.900	0.756	0.902
Europe	41	0.927	0.905	0.950	0.854	0.957

Table 6. Model performance by industry

Industry	n	Accuracy	Sensitivity	Specificity	Kappa	AUC
Industrials	28	0.893	0.933	0.846	0.784	0.974
Consumer Discretionary	40	0.900	0.857	0.947	0.800	0.937
Information Technology	26	0.885	0.933	0.818	0.761	0.927

4.2 Interpretability

Feature importance was calculated with the caret package and is depicted in Figure 6. It is interesting to note that, according to the model, the most relevant variables belong to the profitability and liquidity categories, followed closely by debt. It is also noteworthy that efficiency, as represented by sales/(total assets) does not play a significant role in bankruptcy prediction. This structure is reasonable from an economic and financial standpoint, suggesting that a purposeful exploration of larger feature sets might shed some light on a theoretical formulation of the bankruptcy phenomena.

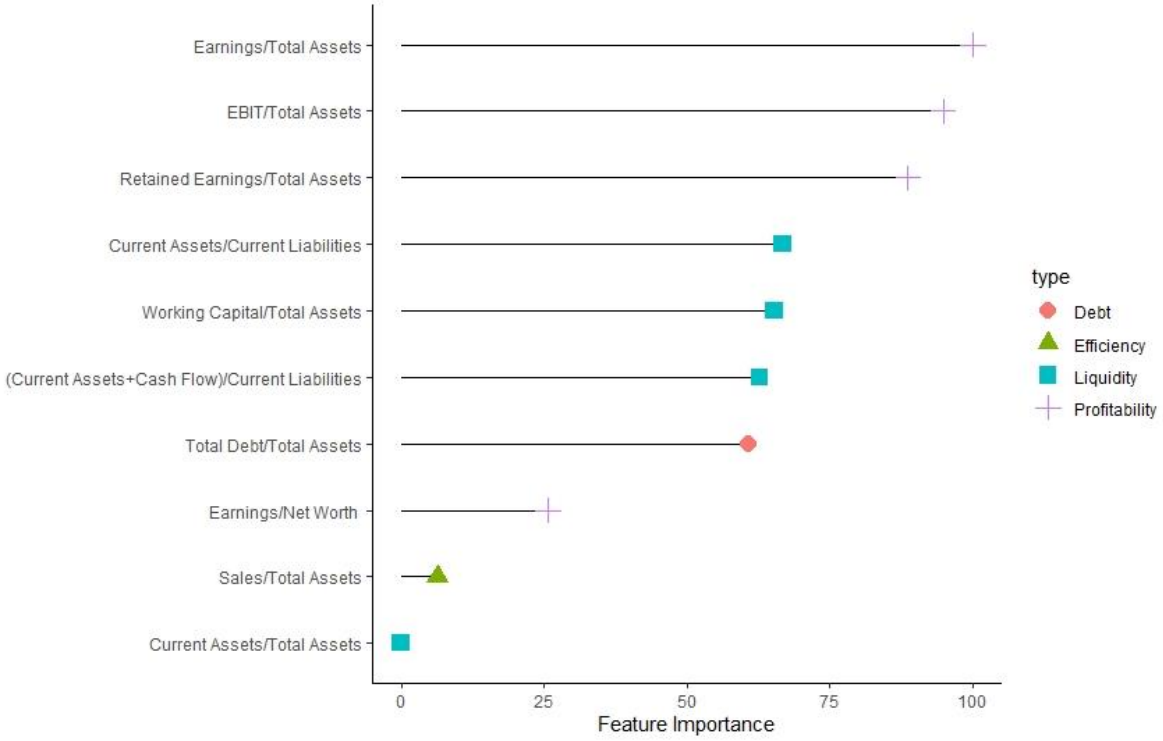

Fig. 6. Feature importance by ratio type

5 Study Limitations and Future Research

As any numerical exploration, our results are limited by the characteristics of the sample. A serious limitation is the availability of features. Replicating this exercise on a feature-rich sample will help us better understand the relationship between endogenous company information and bankruptcy. Our modelling effort was limited—by design—to one period in advance prediction. Developing earlier warnings is a topic of interest in the literature that might be improved by incorporating time series-based models as well as a theoretical foundation.

It is perhaps also important to note again, for clarity's sake, that calculations were largely carried out using the *caret* package for *R.* Technical aspects such as the specific AUC formula(s) used, the justification for certain imputation methods, the method used for feature importance, and similar details are simply consequences of the specific package implementation. It can and should be useful to explore these same results with different specifications, be it within the same *caret* package or otherwise.

Appendix

Ratio	Type
Earnings/Total Assets	Profitability
Current Assets/Current Liabilities	Liquidity
Working Capital/Total Assets	Liquidity
Retained Earnings/Total Assets	Profitability
EBIT/Total Assets	Profitability
Sales/Total Assets	Efficiency
(Current Assets + Cash Flow)/Current Liabilities	Liquidity
Total Debt/Total Assets	Leverage
Current Assets/Total Assets	Liquidity
Earnings/Net Worth	Profitability

References

1. P. Arestis and S. Basu, "Financial globalization and regulation," *Research in International Business and Finance*, vol. 18, pp. 129–140, 2004. URL: https://ideas.repec.org/a/eee/riibaf/v18y2004i2p129-140.html
2. F. Moshirian, "Globalization and financial market integration," *Journal of Multinational Finance Management,* vol. 13, pp. 289–302, 2003. https://doi.org/10.1016/s1042-444x(03)00012-4
3. M. Odom and R. Sharda, "A neural network model for bankruptcy prediction," In Proceedings of the IEEE International Conference on Neural Networks, vol. 2, pp. 163–168, 1990. https://doi.org/10.1016/S0957-4174(97)00011-0
4. E. Latinen and T. Latinen, "Bankrupptcy prediction application of the Taylor's expansion in logistic regression," *International Review of Financial Analysis*, vol. 27, pp. 327–349, 2000. https://doi.org/10.1016/S1057-5219(00)00039-9
5. M. Tinoco and N. Wilson, "Financial distress and bankruptcy and bankruptcy prediction amonng listed companies using accounting, market and macroeconomic variables," *International Review of Financial Analysis*, vol. 30, pp. 394–419, 2013. https://doi.org/10.1016/j.irfa.2013.02.013
6. L. Cultera and X. Bre'dart, "Bankruptcy prediction: the case of Belgian SMEs," *Review of Accounting and Finance*, vol. 15, pp. 1–25, 2015. https://doi.org/10.1108/RAF-06-2014-0059
7. D. Alaminos, A. Del Castillo, and M. A. Fernández. "A Global Model for Bankruptcy Prediction," *PLoS ONE*, 2016. https://doi.org/10.1371/journal.pone.0208476
8. G. Rivera, R. Florencia, V. García, A. Ruiz, J.P. Sánchez-Solís, "News classification for identifying traffic incident points in a Spanish-speaking country: A real-world case study of class imbalance learning", *Applied Sciences*, vol. 10, no. 18, pp. 6253 (2020). doi.org/10.3390/app10186253
9. V. Vapnik, The Nature of Statistical Learning Theory, New York: Springer, 1996. https://doi.org/10.1007/978-1-4757-3264-1
10. G. James, D. Witten, T. Hastie, and R. Tibshirani (2013). "An Introduction to Statistical Learning", vol. 12, New York, Heidelberg, Dordrecht, London: Springer, 2017. https://doi.org/10.1007/978-1-4614-7138-7
11. A. Kowalczyk, "Support Vector Machines Succinctly", Morrisville, NC: Syncfusion, 2017. URL: https://www.syncfusion.com/succinctly-free-ebooks/support-vector-machines-succinctly
12. T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning", New York: Springer-Verlag, 2009. https://doi.org/10.1007/b94608

13. R. C. Team, R: A Language and Environment for Statistical Computing, Viena: R Foundation for Statistical Computing, 2019. URL: https://www.r-project.org/
14. M. Kuhn, "Building Predictive Models in R Using the caret Package," *Journal of Statistical Software*, vol. 28, no 1, pp. 1–26, 2008. https://doi.org/10.18637/jss.v028.i05
15. D. Gohel, C. Jager, Q. Fazilleau, M. Nazarov, T. Robert, M. Barrowman, A. Yasumoto, P. Julian, "Flextable: Functions for Tabular Reporting," 6 April 2021. [Online]. Available: https://cran.r-project.org/web/packages/flextable/
16. H. Wickham, "ggplot2: Elegant Graphics for Data Analysis", New York: Springer-Verlag, 2016. https://doi.org/10.1007/978-0-387-98141-3
17. B. D. Nha, 6 April 2021. [Online]. Available: https://www.mathcha.io/
18. S. van Buuren and K. Groothuis-Oudshoorn, "MICE: Multivariate Imputation by Chained Equations in R," *Journal of Statistical Software*, 2011. https://doi.org/10.18637/jss.v045.i03
19. I. H. Witten and F. E. Hall, "Data mining: practical machine learning tools and techniques", Morgan Kaufmann, 2016. https://doi.org/10.1016/C2009-0-19715-5