Decision Support Systems 52 (2012) 464–473

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

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

Comparative analysis of data mining methods for bankruptcy prediction

David L. Olson [a,*], Dursun Delen [b], Yanyan Meng [a]

[a] University of Nebraska, Management, 3300 Sheridan Court, Lincoln, NE 68506, United States
[b] State University, Management Science and Information Systems, United States

ARTICLE INFO

Article history:
Received 11 March 2011
Received in revised form 3 August 2011
Accepted 4 October 2011
Available online 13 October 2011

Keywords:
Bankruptcy prediction
Data mining
Neural networks
Decision trees
Support vector machines
Transparency
Transportability

ABSTRACT

A great deal of research has been devoted to prediction of bankruptcy, to include application of data mining. Neural networks, support vector machines, and other algorithms often fit data well, but because of lack of comprehensibility, they are considered black box technologies. Conversely, decision trees are more comprehensible by human users. However, sometimes far too many rules result in another form of incomprehensibility. The number of rules obtained from decision tree algorithms can be controlled to some degree through setting different minimum support levels. This study applies a variety of data mining tools to bankruptcy data, with the purpose of comparing accuracy and number of rules. For this data, decision trees were found to be relatively more accurate compared to neural networks and support vector machines, but there were more rule nodes than desired. Adjustment of minimum support yielded more tractable rule sets.

1. Introduction

Bankruptcy prediction has been a focus of study in business analytics because of the importance of accurate and timely strategic business decisions. Even though the accuracy of the prediction model is a very important criterion, understandability and transportability of the model are also important. The accurate prediction of bankruptcy has been a critical issue to shareholders, creditors, policy makers, and business managers.

There is a wealth of research that has been applied to this field [6,9,30,33,38,42], both in finance and in other fields [38]. Among the thousands of refereed journal articles, many recent studies have applied neural networks (NNs) [1,3,18,19,23,24,27,33,34,43,44,46,48]. Another popular approach is decision trees (DTs) [10,37,42,50]. Support vector machines (SVMs) have been proposed for smaller datasets with highly nonlinear relationships [12,15,21,35,40].

The vast majority of studies in this domain have focused on NNs, and how good they are compared to their statistical counterpart (i.e., logistic regression) at fitting data (fidelity [22]). However, neural network models are black boxes [4,51], lacking transparency (seeing what the model is doing, or comprehensibility) and transportability (being able to easily deploy the model into a decision support system for new cases). We argue that decision trees (DTs) can be as accurate, and provide transparency and transportability that NNs are often criticized for.

The paper is organized as follows. Section 2 reviews previous research on bankruptcy prediction based on data mining methods. Section 3 describes data mining methodologies. Section 4 discusses the data collected and Section 5 presents data analysis and prediction model building methods as well as the results obtained from different data mining techniques. Section 6 gives our conclusions.

2. Data mining model transparency

Model transparency relates to human ability to understand what the model consists of, leading ideally to the ability to apply it to new observations (which we might term transportability). If a model is transparent, it can be transported. Some models have consistently proven to be strong in their ability to fit data, such as neural network models, but to have low transparency or transportability. Neural networks by their nature involve highly complex sets of node connections and weights that can be obtained from software, but at a high cost in terms of transparency and transportability because there are so many nodes and weights. Conversely, logistic regression (or logit regression) have a form that can be understood and transported quite easily. Beta weights can be used to multiply times observation measures, yielding a score that can be used to classify new observations with relative ease. Support vector machines share the characteristics of transparency and transportability with neural network models. Decision tree models are highly transparent, yielding IF–THEN rules that are easier to comprehend and apply than even regression models.

Thus the issue of transparency primarily applies to neural network models. Önsei et al. [25] used neural network models to generate weights of 178 criteria which were then used in a model to classify country

[*] Corresponding author. Tel.: +1 402 472 4521; fax: +1 402 472 5855.
E-mail address: dolson3@unl.edu (D.L. Olson).

0167-9236/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2011.10.007

competitiveness. It has been recognized in the engineering field that neural network models need greater transparency [20]. There have been a number of applications [17,27,37] proposing a neurofuzzy framework to take advantage of neural network learning ability and rule-based transparency. Risser et al. [31] used neural networks to fit data, and jack-knife, bootstrap, and their own validation samples to obtain transparent models for evaluation of driver's license suspension. Yuan et al. [47] proposed a fuzzy neural network controller in the electronics field as a means to combine semantic transparency of rule-based fuzzy systems with the ability of neural networks to fit data. Chan et al. [8] used a similar approach to support vector regression models.

3. Data mining methodology

In a comparative analysis of multiple prediction models, it is a common practice to split the complete data set into training and testing sub sets, and compare and contrast the prediction models based on their accuracy on the test data set. In splitting the data into training and testing dataset one can choose to make a single split (e.g., half of the data for training and other half of the data for testing) or multiple splits, which is commonly referred to as *k*-fold cross validation. The idea behind *k*-fold cross validation is to minimize the bias associated with the random sampling of the training and holdout data samples. Specifically, in *k*-fold cross validation the complete data set is randomly split into *k* mutually exclusive subsets of approximately equal size. Each prediction model is trained and tested *k* times using exactly the same *k* data sets (i.e., folds). Each time, the model is trained on all but one folds and tested on the remaining single fold. The cross validation estimate of the overall accuracy of a model is calculated by averaging the *k* individual accuracy measures as shown in the following equation

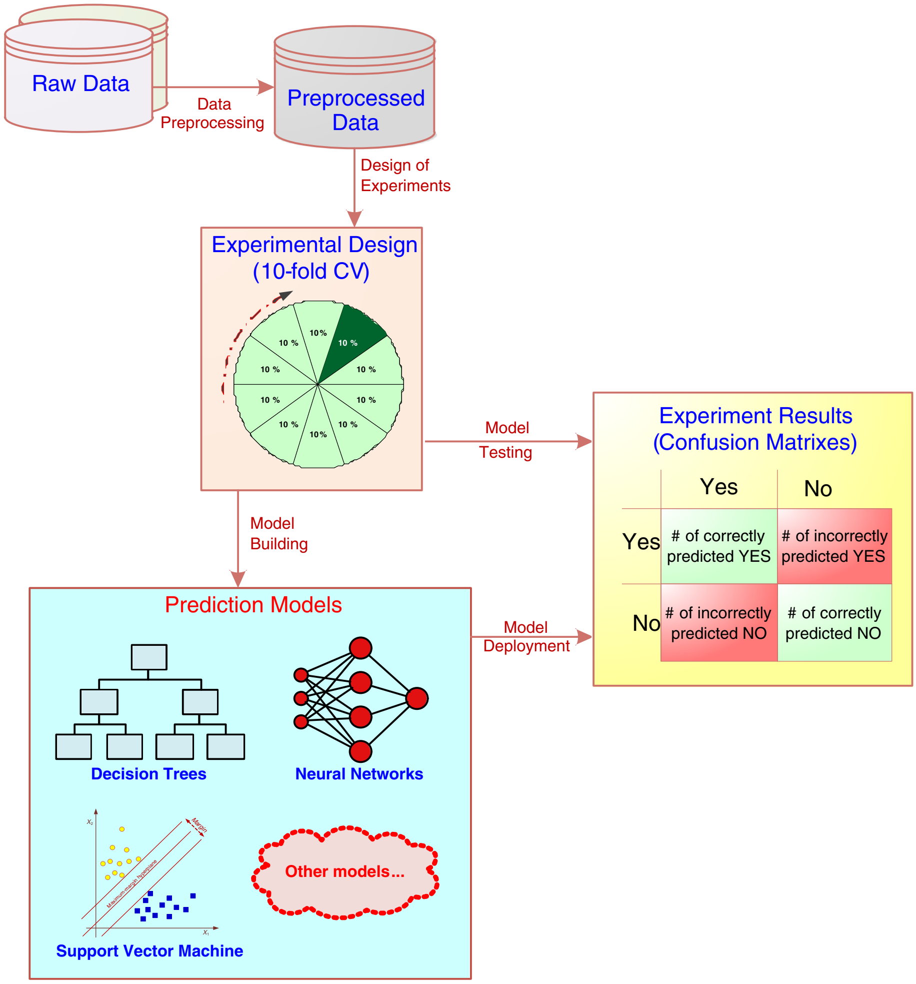
$$OA = \frac{1}{k}\sum_{i=1}^{k} A_i$$

where *OA* stands for overall cross validation accuracy, *k* is the number of folds used, and *A* is the accuracy measure of each folds.

Since the cross-validation accuracy would depend on the random assignment of the individual cases into *k* distinct folds, a common practice is to stratify the folds themselves. In stratified *k*-fold cross validation, the folds are created in a way that they contain approximately the same proportion of predictor labels as the original dataset. Empirical studies showed that stratified cross validation tend to generate comparison results with lower bias and lower variance when compared to regular cross-validation [16]. In this study, to estimate the performance of predictors a stratified 10-fold cross validation approach is used. Empirical studies showed that 10 seem to be an “optimal” number of folds (that balances the time it takes to complete the test and the bias and variance associated with the validation process) [7,16]. The methodology followed in the study is depicted in Fig. 1.

3.1. Prediction methods

In this study, several popular classification methods (e.g., artificial neural networks, decision trees, support vector machines and logistic



	Yes	No
Yes	# of correctly predicted YES	# of incorrectly predicted YES
No	# of incorrectly predicted NO	# of correctly predicted NO

Fig. 1. A graphical depiction of the methodology followed in this study.