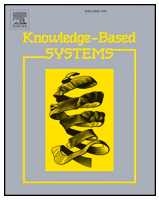
Knowledge-Based Systems 289 (2024) 111529

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Variable selection in the prediction of business failure using genetic programming

Ángel Beade [a], Manuel Rodríguez [b], José Santos [c,*]

[a] Business Department, University of A Coruña, Campus de Elviña, s/n 15071, A Coruña, Spain
[b] Business Department, University of A Coruña, Campus de Elviña, s/n 15071, A Coruña, Cátedra AECA-Abanca, UIE, Spain
[c] Department of Computer Science and Information Technologies, CITIC (Centre for Information and Communications Technology Research), University of A Coruña, Campus de Elviña, s/n 15071, A Coruña, Spain

ARTICLE INFO

Keywords:
Business failure
Dimensionality reduction
Feature selection
Evolutionary computation
Genetic programming

ABSTRACT

This study focuses on dimensionality reduction by variable selection in business failure prediction models. A new method of dimensionality reduction by variable selection using Genetic Programming is proposed, which takes into account the relative frequency of occurrence of the explanatory variables in the evolved solutions, as well as the statistical relevance of that frequency. For a better evaluation of the proposed method and its comparison with other well-tested and widely used variable selection methods, the prediction of business failure in three temporal horizons (1, 5 and 9 years prior to failure) is considered. Additionally, a comparison of the sets of variables selected with different feature selection methods is performed, also considering different classifiers in the comparison, among which Genetic Programming is included as a classifier. The results indicate that the proposed method (using Genetic Programming as a variable selection method) is superior to the most tested and widely used methods analyzed, and this superiority increases if Genetic Programming is also used as a classification method.

1. Introduction

In addition to the selection of the classification method, in business failure prediction (Bankruptcy Prediction - BP) one of the basic challenges is the selection of the explanatory variables [1]. Studies have shown that BP models can be more effective if procedures of data preprocessing are performed (among which the selection of explanatory variables occupies a relevant place) [2]. However, there is no consensus when it comes to focusing on this variable selection. Financial ratios have traditionally been the most commonly used type of explanatory variable in BP work. As Barnes [3] points out, financial ratios have typically been selected on the basis of their popularity, in addition to the fact that each researcher can add a new one (ratio) in his or her particular work. Laitinen [4] also comments that "They are (the financial ratios) usually selected from data simply statistically without any rigorous hypotheses on the behavior of the firm before failure" and, in the aforementioned study, Laitinen [4] proposes as an objective "to select the financial ratios on the basis of a theoretical model".

This choice is still the usual one, as indicated by du Jardin [5] when he summarizes the criteria used in the selection of explanatory variables to be included in BP models. As the author points out, 40% of the analyzed works use "Popularity in the literature or predictive ability assessed in previous studies" as a criterion [5]. Thus, there are currently a multitude of studies that use as explanatory variables those pre-selected in previous studies [6,7], while other authors apply different techniques for the selection of variables [1,8,9]. In this line, Alaka et al. [10] provide a general analysis of the different methods used in variable selection, considering 49 BP studies in the period between 2010 and 2015, which highlights the aforementioned lack of consensus.

There has been an exponential increase in the information available from companies (restricted and/or public use), at the same time that classification methods have increased their potential to deal with a larger number of variables and also with different types of variables. However, these available data are often overloaded with a multitude of features (i.e. input variables or independent variables) which, in addition to the increase in cost and time to obtain solutions, may result in overfitting of the BP model. More data does not necessarily mean better results, as concluded by Chandrashekar and Sahin [11].

Consequently, two different ways can be summarized when selecting the most relevant variables in BP:

* Corresponding author.
E-mail address: jose.santos@udc.es (J. Santos).

https://doi.org/10.1016/j.knosys.2024.111529
Received 10 October 2023; Received in revised form 2 February 2024; Accepted 14 February 2024
Available online 15 February 2024

- Use variables pre-selected in previous studies. This generally involves using reduced sets of variables.
- Selecting variables for BP through a process of dimensionality reduction of initial sets with a large number of available variables.

In this study, the second alternative was considered by choosing a large initial sample of commonly used variables (mostly financial ratios), additionally extended with other variables that may be useful in insolvency prediction. Then, dimensionality reduction with automatic variable selection is addressed, as this initial set is then subjected to different Feature Selection (FS) methods. That is, instead of a selection process based on the presence of variables in previous studies, the variable selection process can automatically detect the most relevant variables from a large initial set, with the possibility of improving the performance of the prediction models.

Within the BP framework, there are some works that analyze different combinations of selection techniques of the explanatory variables and classification methods [12] or the impact of variable selection methods on the results obtained with different classification methods [13]. In our study, tree-based Genetic Programming (GP) [14] is proposed as a dimensionality reduction method to detect the most relevant explanatory variables for BP models. This will be based on the hypothesis that relevant explanatory variables remain in the population of solutions and throughout the evolutionary process, while irrelevant variables disappear over generations and due to selection pressure. Our novel approach will consider the statistical relevance of the appearance of variables in the evolutionary process, in order to ensure a correct choice of the significant variables.

The set of variables obtained with GP is evaluated – together with those obtained with other dimensionality reduction methods – by means of different classification methods in the field of Machine Learning (ML) (including GP, which is used in this work as a method to reduce dimensionality and as a classification method). The interest of GP in the BP field is because GP-based insolvency prediction models have the important property of direct interpretability. Solutions based on GP trees are usually more interpretable (obviously, depending on the complexity of the tree) than other solutions (classifiers used for BP) that can be considered black boxes. In Brabazon et al. [15] words, GP can provide solutions that are human-readable. And our study, in addition, takes advantage of and analyzes GP as a variable selection technique in the field of BP.

Therefore, the main objective of our study is to analyze our proposed use of GP as a dimensionality reduction method. Our proposal will consider the frequency of selection of input variables in the GP evolutionary process and this selection will start from a wide set of variables in the BP field. This selection will be performed in the context of predicting failure at three different temporal horizons: at 1, 5 and 9 years prior to failure, thus taking into account prediction scenarios that can be considered in the short, medium and long term. These horizons also provide several different scenarios in which to perform a detailed evaluation and comparison with other FS approaches, also taking into account different classification methods in the comparison.

The rest of the paper is structured as follows: Section 2 presents the general approach to the dimensionality reduction problem and synthesizes the approach to this problem in the BP field. Section 3 explains the proposed feature selection approach with GP. Section 4 explains the methods used in the study, detailing the aspects involved in the design of the BP models: data used, the input variables and how the training and test sets are defined, as well as the dimensionality reduction methods and classifiers used in the comparisons. Section 5 first details the setup of all the methods considered and then details and analyzes the results obtained, highlighting those of our study proposal. Section 6 provides a general discussion of the comparisons performed and the results obtained. Finally, Section 7 includes the main conclusions that can be obtained from the study, as well as some possible avenues for future research.

2. Dimensionality reduction

Dimensionality reduction is the process of obtaining a set of main (or explanatory) variables of a classification/prediction model from an initial set of input variables. The most common applications of dimensionality reduction are found in classification, clustering and regression tasks [16].

Following Tang et al. [17], the expected advantages of reducing the dimensionality of the set of input variables could be summarized as follows: reduced computational complexity, reduced storage requirements, improved ML model performance, as well as models that generalize better.

Generally speaking, there are two ways to approach dimensionality reduction:

- Feature selection
  It consists of eliminating some variables if it is considered that they are not providing relevant information about the dataset. It is easy to implement, but – as a disadvantage – information about the eliminated variables could be lost.
- Variable extraction
  This is the formation of new variables from old ones. Two categories can be considered, depending on whether a linear or a nonlinear dimensionality reduction transformation is applied.

In this work, we will address dimensionality reduction through feature selection. The reason is to use and identify the most relevant variables from a large initial set of explanatory variables that correspond to traditional features used in insolvency prediction models (such as financial ratios). This improves the interpretability (and even acceptance) of the prediction models by an end user of the model and compared to the use of explanatory variables that would correspond to linear/non-linear dimensionality reduction methods in the second option of variable extraction.

There are multiple ways to classify different feature selection methods [16,18]. The most common is the classification that is made according to the relationship with the learning method. According to this criterion, feature selection methods are classified into: filter methods, wrapper methods and embedded methods.

The basic characteristics of each of them are, very succinctly, as follows [16,18]:

- *Filter* methods: These are independent of any learning method. They select features based on a performance measure independent of the learning method. The idea is to filter or detect those features that best discriminate the examples of different classes, taking into account their intra-class and inter-class variation. Once the best feature subset has been found, the learning method is applied. They are not computationally expensive and tend to generalize well.
- *Wrapper* methods: These evaluate the subsets of features by means of their performance in a modeling algorithm, which acts as a "black box" evaluator. These are more computationally expensive than filtering methods and tend to obtain subsets with better modeling results.
- *Embedded methods*: these perform feature selection as the learning method is applied, as they are embedded in the learning method (either as a normal functionality or as an extended functionality).

In the case of BP, the relevance of the issue of explanatory variable selection and the inadequacy of the method of selection by pre-selected variables in other studies, make the reduction of the dimensionality of the initial set of input variables currently an unresolved challenge [19]. Some reviews [10,20] provide an overview of the most commonly used methods.

The following section details our proposal, which uses GP as a feature selection method, reasoning its embedded nature. This use of GP

for variable selection is novel in the field of BP.

3. Proposed feature selection method with genetic programming

3.1. Brief comments on genetic programming

In Evolutionary Computation (EC) methods, the main feature of Genetic Programming (GP) [14,21] is that it evolves "programs", which are typically represented as decision trees. The evolutionary process evolves and optimizes these genetic population programs over generations. The standard genetic operators in GP for dealing with decision trees can be found in Poli et al. [21]. Being able to obtain programs or decision trees in an automatic way, without knowing a priori the structure of the optimal solution, is what gives GP great versatility with respect to other evolutionary methods [22].

In BP the objective is to obtain models with high predictive power and, in this goal, GP has several advantages: i) No prior assumptions are made by GP regarding the explanatory variables to be used in a prediction model. ii) As previously mentioned, GP provides a direct interpretability of the evolved program/tree. iii) The possibility to adjust the complexity of the optimized program/tree, for example by adjusting the number of tree nodes, the depth of the tree or the functions that can be used on the tree nodes to find an optimal solution.

Specifying GP use in BP, on one hand, there has been a limited amount of previous work on the use of GP as a classification method in the field of BP [23–30]. These works focused almost exclusively on comparing the results obtained by GP versus those obtained by other techniques and making mostly short-term predictions (1 year prior to failure). Moreover, as Brabazon et al. [15] point out, knowledge of GP among finance professionals, or even among finance academics, is rather scarce, despite the fact that BP is an area where GP can give good results [21].

On the other hand, and as already indicated, it is proposed to use GP as a method for dimensionality reduction by means of feature selection. In this context, the surveys by Dokeroglu et al. [31] and Xue et al. [32] analyze different possibilities of using EC algorithms for feature selection. Using EC for FS has the advantage of the global search of the evolutionary algorithms, in this case in the feature space (contrary to many FS methods based on local search procedures, such as some of those selected below in Section 4.2 for the comparison of results). Several FS methods with EC have been proposed [30–34] and, in most of these EC-based methods, a fixed-length encoding is used to represent subsets of selected features. The works [33,34] propose a length-adaptive genetic algorithm to overcome this problem, especially in high-dimensional feature spaces. It should be noted that, in the case of GP as FS, since the FS process is inherent to the GP evolutionary process, compared to other EC-based solutions for FS, it does not require explicit genotypic coding for FS.

Those surveys on EC-based methods for FS ([31][32]) also discuss different possibilities of using GP in feature selection, including the GP possibility of construction of new (high-level) features that can increase the performance of the classifier. In this case, GP as a dimensionality reduction method has been practically not considered in the BP environment, although there is a small number of works in other areas that corroborate the effectiveness of GP as a feature selection method [30, 35–39]. In what follows, our proposal for GP as a feature selection is detailed, which is then applied to BP.

3.2. Proposed approach

3.2.1. Basis of the proposed approach

An interesting feature in the case of GP is that it intrinsically selects explanatory variables that are relevant for classification using decision trees. This is due to the selective pressure in the evolutionary process, which leads the decision trees, generation after generation, to use the most relevant variables (individually or in conjunction with other variables) for classification, while progressively discarding those with less classification capacity. This intrinsic selection is an important feature of GP compared to other methods in EC. From this point of view, dimensionality reduction by feature selection would not be necessary in GP as it is inherent to it (contrary to other automatic feature selection methods in ML).

Also taking into account the stochastic process that governs GP, we propose an approach based on the hypothesis that variables that are relevant remain in the population of solutions and across the GP generations, as opposed to irrelevant variables that will gradually disappear due to selection pressure across the GP evolutionary process. In our proposal, the relative frequencies of appearance of each input variable in the nodes of all the trees of the genetic population (and in all generations of each GP run) are considered. With aggregation, the relative frequencies of the input variables are calculated over a set of runs of the stochastic GP process (or a subset of GP runs). This measure (relative frequency) may not be strictly precise, but the relevant input variables are expected to remain progressively in the evolved solutions.

In the stochastic process that governs GP, branches of the tree (parts of the solution or program) that do not alter (or worsen) the performance of the model may arise spontaneously. This growth of the program without (significant) improvement of the fitness of the program is called "bloat". However, it should be noted that the use of independent GP runs to consider the relevance of variables decreases the potential problem of bloat that may appear in particular GP runs. An irrelevant variable (due to the bloat problem) may appear in a particular GP run, but the same variable is unlikely to appear repeatedly in different independent GP runs. This is why several GP runs must be used to consider (statistically) the relevance of variables, as detailed below.

Note that our approach can be considered an embedded approach, taking into account the classification given above in Section 2, since the selection process is embedded and occurs across the GP evolutionary process, as also indicated in Xue et al. [32] when the authors state that "Among current EC techniques, only genetic programming (GP) and learning classifier systems are able to perform embedded feature selection". Our approach is in line with that used by Neshatian and Zhang [37], although these authors only considered the frequency of appearance of variables in the best GP individuals to select the top-ranked features. On the contrary, in our case we will take into account the statistical relevance of the occurrence of variables in the evolutionary process (as detailed in the following subsection), thus ensuring a more correct choice of the significant variables according to their presence in the evolved trees.

The objective of applying the proposed feature selection is to evaluate whether, by using only a limited number of variables (selected in the aforementioned way), there is an improvement in the performance of the prediction models with respect to the results obtained with the total number of variables. The use of GP for the selection of features in the aforementioned way provides a ranking of variables as the final product. In addition, this GP-based feature selection method is context-sensitive. In contrast, most of the different feature selection methods that provide variable rankings are context-insensitive. In a problem, a feature may show no relevance in the absence of other relevant features. However, such a feature may be relevant in the presence of other features [5,37]. Therefore, the context must be taken into account, which GP achieves, as a variable is selected in a tree either because of its individual classification ability or in the presence of other selected variables in the tree. Consequently, the variable selection by GP is considered context-sensitive [37], which is also a differential characteristic with respect to other ranking methods (such as the case of univariate filters).

3.2.2. Steps of the proposed feature selection method

We refer to our proposed method as GPFS (GP Feature Selection). For the selection of features with GP (as a function of their relative frequency), for a given model (prediction X years prior to failure), the

following steps will be followed:

Step 1. Taking into account all the input variables, a large experiment (1000 independent runs of GP) is performed. That is, GP evolves trees that can use all these input variables, trees that perform the prediction/classification of companies. Performing an experiment with many runs reduces the uncertainty in the results.
Step 2. A subset of GP runs is chosen that generate what are considered to be the best solutions, meaning those with the highest Area Under the ROC (Receiver Operating Characteristic) Curve (AUC) in their classification. In the experiments, the subset corresponds to 5% of the total 1000 independent GP runs.
Step 3. For each input variable, its relative frequency in the runs of GP that provide these best solutions is calculated. If the selection of variables were done randomly (without the GP algorithm intrinsically selecting variables), the relative frequency distribution of a given variable could be approximated by a normal distribution and its values typified by a standard normal - N(0,1) – according to the Moivre-Laplace theorem –. Testing the fit between the actual relative frequency distribution of a variable (obtained with the subset of GP runs that provide the best solutions) and the expected distribution N (0,1) allows to reject – or not – the hypothesis of randomness in the algorithm (Kolmogorov-Smirnov test). The aggregation of the solutions of the independent GP runs, selected for the calculation of the frequencies of occurrence of the variables, ensures that a correct frequency distribution of the variables is obtained to check their relevance by means of the statistical test. Consequently, a ranking of the variables will be available, based on their p-value. Fig. 1 shows the flowchart of the GPFS method, which summarizes its feature selection process. Finally, it is decided which subset of input variables to select. For example, variables with a p-value$<$0.05 can be chosen to select variables with statistically relevant results, rather than due to chance.

It should be noted that the disadvantage of the proposed GP-based approach is the high computational cost with respect to other FS methods (such as those used later in the comparison in Section 5), since different GP runs are necessary to (statistically) guarantee the correct detection of the relevant variables.

4. Methods

4.1. Design of the prediction models

4.1.1. Sample of companies

This paper focuses on BP models for medium-sized Spanish firms. A set of 11,158 firms (1067 classified as failures and 10,091 as non-failures) is available, which is a larger dataset than those used in most BP work. The concept of failure used is the declaration of insolvency (which implies the suspension of payments by the debtor), since this is the most common state for failure definition in the BP field.

Several sectors that use specific valuation or accounting criteria, which could alter the financial ratio interpretation and consequently distort the results, have been excluded, sectors such as building construction, financial services (e.g., pension funds and insurance), general government activities and compulsory social security. Furthermore, only limited liability companies, public limited companies and co-operatives are considered in the study.

In order to assess, in detail, the proposals of the study, the prediction horizons of the business failure prediction models carried out are: 1 year, 5 years and 9 years prior to failure (named: Model 1, Model 5 and Model 9, respectively). The observations (data of failed or non-failed companies in a financial year) from 2005 to 2007 (both included) are used to obtain the different temporal BP models.

The sources of information used are the following:

- Accounting information of the companies: SABI database (www.informa.es/en/business-risk/sabi) - Iberian Balance sheet Analysis System.
- Legal information on the state of failure: register of public insolvency (www.publicidadconcursal.es).

4.1.2. Initial set of input variables

The explanatory variables considered in the BP models are mostly

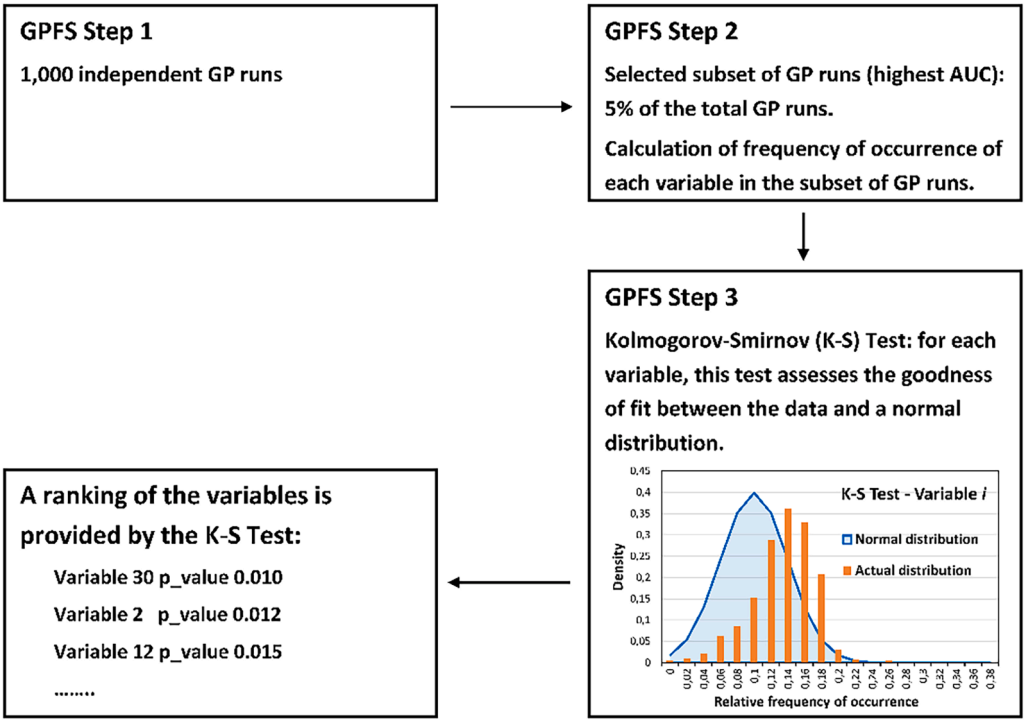


Fig. 1. Steps of the GPFS process.

financial ratios, defined from the companies' annual accounts. Financial ratios have become the most widely used type of explanatory variable for predicting business failure. Du Jardin [5] indicates that 93% of the 190 studies analyzed in his work use financial ratios as explanatory variables (53% of the studies use financial ratios exclusively).

The choice of the explanatory variables to be used was made taking into account: i) their relevance in the literature and also their presence in the BP models considered in previous work and ii) the inclusion in the set of explanatory variables of other variables that refer to aspects that are not widely used (for example, magnitude variations or ratio variations) or that are very infrequent or novel in BP models (for example, those related to fraud and productivity and those related to the decomposition degree of the balance sheet). These additional variables are also obtained exclusively from the companies' annual accounts.

In the process of selecting the explanatory variables mentioned above, different relevant previous works have been considered: Altman and Sabato [40], Altman et al. [41], Bellovary et al. [42], Beneish [43], du Jardin [44], Tian and Yu [45] and Yardeni et al. [46]. The initial and total set of explanatory variables corresponds to the following categories:

Changes in ratios	2 ratios
Contribution	2 ratios
Degree of decomposition	3 variables
Efficiency	11 ratios
Financial structure	14 ratios
Fraud	11 variables
Growth	1 ratio
Interest expenses	5 ratios
Liquidity and solvency	18 ratios
Productivity	4 ratios
Profitability	12 ratios
Size	4 variables
Turnover	7 ratios
Variations in magnitudes	3 ratios

It can be seen that the range of explanatory variables expands considerably beyond the traditional financial ratios. This approach seeks to include variables that reflect changes over time and to cover a broad spectrum of aspects considered relevant in the prediction of business failure. Thus, in this context, a total of 97 input variables are used, as described above.

However, before using these input variables, several steps are carried out to ensure the integrity and quality of the data. Firstly, comprehensive checks are made on the total figures and the balance sheet and income statement breakdowns for each of the companies in each of the years. This makes it possible to eliminate observations that do not allow the precise calculation of the explanatory variables. In addition, a limitation of extreme values is made. Those variables that have values below the 2.5% percentile or above the 97.5% percentile in a specific accounting period are replaced by the value corresponding to the reference percentile. This is done in order to eliminate very extreme values and to avoid learning difficulties.

Finally, standardization of the data is carried out. In this last step, with the bounded values of the input variables, they are transformed according to the logistic distribution. This transformation is performed using the mean and standard deviation of each variable in the total data period. The purpose of this step is to homogenize the ranges of variation of the previously bounded variables. In summary, the process of transforming the input data involves ensuring the integrity of the data, limiting outliers and standardizing the variables. In this way, the aim is to use a robust and homogeneous set of explanatory variables to analyze and understand business failure.

4.1.3. Training and test sets

When setting up the training set, it should be noted that in the problem of business failure prediction the populations are totally imbalanced. There is a majority class (observations of non-failed companies) that far outweighs the minority class (observations of failed companies). In this situation, it is feasible to consider a training set with imbalanced classes. However, the largest challenge faced by the lack of representation of the minority class instances is that their data could be overlooked due to their low number. Therefore, even if the overall classification model achieves high accuracy, the results for the minority class may be poor. When the minority class is particularly relevant (as in the case of BP), this risk is not acceptable and attention needs to be paid to the minority class. This is why, according to Alaka et al. [10], it is concluded that 80% of studies on business failure use training sets with majority class/minority class percentages between 50%−50% and 60%−40%. In this case, the 50%−50% ratio has been chosen, so the training set will have the same number of observations corresponding to failed companies as to non-failed companies. This use of a balanced training set allows to obtain a trained classifier that does not focus its learning on a particular class (the majority class), as could happen with a very imbalanced training.

It is common, in BP studies, to select observations of failed and non-failed companies for the training sets on the basis of size, age, sector, etc. In any case, Palepu [47] points out that, not selecting a sample randomly presents at least two important drawbacks: i) overestimating the predictive capacity of the model and ii) making generalization to the rest of the population difficult. On the other hand, More [48] warns that the responses with other subsampling techniques, when defining the association between the observations of both classes in the training set, are highly dependent on the classification problem. Based on the above, we have chosen to select the totality of these observations randomly (both those corresponding to failed companies and those corresponding to non-failed companies).

The number of observations corresponding to failed and non-failed companies used in the training and test sets of the prediction models (period 2005–2007) was as follows:

- Model 1: 82 failures and 22,330 non-failures.
  - Model 1 - training: 82 observations (41 failures + 41 non-failures).
  - Model 1 - test: 22,330 observations (41 failures + 22,289 non-failures).
- Model 5: 282 failures and 22,330 non-failures.
  - Model 5 - training: 282 observations (141 failures + 141 non-failures).
  - Model 5 - test: 22,330 observations (141 failures + 22,189 non-failures).
- Model 9: 122 failures and 22,330 non-failures.
  - Model 9 - training: 122 observations (61 failures + 61 non-failures).
  - Model 9 - test: 22,330 observations (61 failures + 22,269 non-failures).

The evaluation and comparison of the different methods used is performed on the results obtained in the test set.

4.2. Dimensionality reduction methods

Starting from the detailed initial set of input variables, one of the following dimensionality reduction alternatives is applied:

- Different methods of dimensionality reduction by feature selection.
- The proposed method of dimensionality reduction based on GP.

The feature selection methods chosen are some of the most popular ones, looking for a representative variety of them. In all cases, the starting point for obtaining the selected subset of explanatory variables is the initial set of financial variables of the company (applying the logistic transformation). Comparison between algorithms of feature selection can only be done using a single data set, as each underlying algorithm will behave differently for different data [11]. Therefore, all

methods are compared with the same training/test data at the three prediction temporal horizons.

The feature selection methods used are as follows (including some basic comments):

- *Filter*
  - Information Gain (InfoGain) [49]:
    - Univariate (evaluates each characteristic one by one).
    - A ranking is made by evaluating each attribute by means of the information gain with respect to the class.
  - RelieF [50,51]:
    - Univariate.
    - It is a method based on the distance of features. It performs a ranking of the evaluations of each feature, obtained by repeatedly sampling an instance and considering a score value of the given feature taking into account the nearest instance of the same class and the nearest instance of a different class. The score of any feature decreases when it differs from the same feature in close instances of the same class more than in close instances of the other class. In the opposite case, the score increases.
  - Chi-squared
    - Univariate.
    - Performs a ranking of the evaluations of the behavior of each feature, obtained by means of the Chi-square statistic with respect to the class.
  - Correlation
    - Univariate.
    - Performs a ranking of the evaluations of each feature, obtained by measuring the Pearson correlation coefficient between the feature and the class.
  - Support Vector Machine (SVM) [52]
    - Univariate.
    - It can be categorized as a filter method since each feature is evaluated using a classification method – SVM – as the filtering method. Features are ordered by the square of the weight assigned by the SVM in the decision function. These weights are a function of only a small subset of the training examples, the "support vectors".
  - Correlation-based Feature Selection (CFS) [53]:
    - Multivariate (evaluates a subset of features).
    - Subsets of features uncorrelated with each other, but highly correlated with the class are searched for.
    - Search strategy: *LinearForwardSelection*, which is an extension of *BestFirst*. *BestFirst* is an attribute subset search algorithm that uses a heuristic search by greedy hill climbing and adding backtracking capability. *LinearForwardSelection* takes into account a restricted number of k features (selected by an initial sorting as the most important ones).
  - Consistency-based [54,55]:
    - Multivariate.
    - The value of a subset of features is evaluated by the level of consistency in the class values when training instances are projected onto the aforementioned subset of features. In the words of Dash and Liu [54], the consistency measure does not try to maximize the separability of the class, but tries to maintain the discriminative power of the data defined by the original features. The aim would be to find the smallest subset of features that distinguishes the classes the same as the original set of features.
    - Search strategy: *LinearForwardSelection*.
- *Wrapper*
  - Wrapper J48 [56]:
    - It uses J48 as an auxiliary classification algorithm, a free java implementation of the C4.5 algorithm, which uses the concept of information entropy for the selection of variables that provide the best classification in the class under study. The C4.5 algorithm is used to generate a decision tree.
    - Search strategy: *LinearForwardSelection*.
  - Wrapper Naive Bayes [57]:
    - It uses Naive Bayes as an auxiliary classification algorithm, which is a probabilistic algorithm based on Bayes' theorem.
    - Search strategy: *LinearForwardSelection*.

The final result of applying the aforementioned dimensionality reduction methods is a subset of the initial set of input variables (a different one for each of the methods and for each of the temporal horizons over which the comparison will be made: 1, 5 and 9 years prior to failure) that will constitute the set of explanatory variables on which different classification methods will then be applied.

4.3. Classification methods

The different selected sets of explanatory variables (one for each feature selection method and for each of the temporal horizons) are evaluated with each of the chosen classification methods. These classification methods have also been chosen for their popularity and in order to offer a representative variety of them. The selected classification methods (except GP) are divided into two categories: single methods and ensemble methods.

The selected classification methods are the following, again including some basic comments on them:

- *Single methods* (only one classifier)
  - J48: generates a type (C4.5) of decision trees (pruned or unpruned) that have certain advantages over other types of decision trees (including mitigating overfitting) [58].
  - Multilayer Perceptron (MLP): This is a classical artificial neural network topology consisting of several layers, which allows it to address problems that are not linearly separable. It uses back-propagation for learning.
  - JRip: Implements a propositional rule learner called Repeated Incremental Pruning to Produce Error Reduction (RIPPER). This RIPPER algorithm is a classification method based on rules that are extracted from the training set, which was proposed by Cohen [59] as an optimized version of IREP (Incremental Reduced Error Pruning).
  - QDA: Quadratic Discriminant Analysis is a generalization of LDA (Linear Discriminant Analysis) that uses a quadratic decision surface to separate two or more classes.
  - Logistic: performs the classification using a multinomial logistic regression model with a ridge estimator.
  - Naive Bayes: it is a probabilistic method based on Bayes' Theorem, with some simplifications about the independence of the explanatory variables.
  - LibSVM: Support Vector Machine implementation for classification.
  - KStar: It is a classifier in which the class of an observation is determined by the class of other observations similar to it, according to some previously defined similarity function. It uses a distance function based on entropy [60].
- *Ensemble methods*
  - Bagging: Also known as "bootstrapping and aggregating". It is an ensemble algorithm that, starting from a training set, selects random samples from that set (with replacement) and fits to each of them a weak learning model, understanding as such those models with behaviors slightly superior to that of a random model. It then combines these models to make a single prediction [61].
  - Decorate: It builds ensembles of classifiers by creating artificial training sets. It is considered to be more accurate than Bagging and Random Forest [62].

- Random Forest: Constructs a forest of random trees and then combines them. Each tree depends on a "random vector" generated independently of the random vectors of the rest of the trees, but with the same distribution for each of them and which governs the growth of each tree in the set [63]. It is a modification of Bagging. Briefly, the random vector contains the key to randomly choose the elements of the training set and the variables of each of them to be used in the development of the tree.
- Vote8: The idea of voting methods is to combine different machine learning models and predict the class label by means of the majority vote or the average of the predicted probabilities. In our case, we decided to combine the eight different single classifiers above mentioned and the average of the predicted probabilities as a rule to predict the class [64].
- Random Subspace: This method (also called "feature bagging" or "attribute bagging") constructs a decision tree-based classifier. The classifier consists of multiple trees. Each tree is trained on all examples, but only considers a random subset of the attributes instead of the entire feature set. The size of these subsets is the parameter of the method, and the result is the average or voting of the individual results of the models. In this way, the method attempts to reduce the correlation between estimators [65].
- AdaBoost M1: Boosting always works with the full input set (unlike bagging) and modifies the weights of the outputs obtained from weak classifiers to create different models. In each iteration the weights of the misclassified items are increased in order that in the next iteration these items will be more important and more likely to be classified well. We use the AdaBoost M1 algorithm [66].

Some ensemble methods (such as boosting) intrinsically implement feature selection (as does GP) since it uses an iterative approach to build a sequence of models, where each model focuses on correcting the errors of the previous model, assigning different weights to the models (and consequently to their input variables), resulting in an intrinsic selection of variables. Other ensemble methods such as bagging and random subspace use random subsets of variables for learning, so it is debatable whether this is intrinsic variable selection at all. Finally, the voting ensemble methods do not perform intrinsic variable selection. Anyway, it is documented that the performance of ensemble classifiers improves with feature selection [67,68]. On the other hand, ensemble methods can be profitably used for feature selection [69,70].

- *Genetic programming*

As previously mentioned, GP obtains optimized programs/trees through an evolutionary process over several generations [14,21]. In order to compare the different dimensionality reduction methods using GP only as a classification method, an experiment (1000 independent GP runs) with a similar profile except for the set of input variables is carried out for each of the 10 options considered: the 9 external ones (InfoGain, RelieF, …) and one corresponding to the dimensionality reduction by relative frequency in the case of GP. This is done independently for each of the three prediction temporal horizons.

It should be noted that these 1000 runs of GP, which are used with GP as a classifier, are independent and are not related to the 1000 GP runs used with GP as a feature selector. That is, the initial selection of variables with GP (as a feature selector) is an independent and prior process to the possible later use of GP as a classifier.

5. Results

5.1. Setup of feature selection methods and classifiers used

To obtain the feature subsets according to the aforementioned methods (in Section 4.2) the Weka software and the configurations it provides by default were used. A complete description of that software can be found in the online appendix of the book by Frank et al. [71,72]. The FS approaches used (with Weka's default configuration employed) in the comparison do not have stochastic components. The number of features selected when the method provides as output a ranking of the features has been set to 20 (20.61% of the initial 97). To set the number of features to be selected in the methods that provide a ranking, the Pareto principle (80% of the effects come from 20% of the causes) has been used. Other criteria could be used, such as setting a threshold of importance, a cumulative contribution, etc., but the criterion applied allows for a homogeneous approach for all the methods concerned.

Weka software was also used to implement the classifiers (except GP) that have been used to compare the selected sets of features. All parameters associated with such classifiers, specified in Section 4.3, have been set as those that Weka software sets by default.

Regarding GP, HeuristicLab (HL) software [73] (https://dev.heuristiclab.com/trac.fcgi/) was used to implement the BP models in GP. HeuristicLab was selected because of its detailed user interface, and especially because it strongly abstracts the process of heuristic optimization [73].

The solution representation used is the traditional tree-based representation in GP with the HL environment. Note that the solutions of the GP population (classifiers of insolvency) can use as inputs the ample set of explanatory variables previously detailed in Section 4.1.2.

Table 1 lists the most relevant GP parameters (considering HL nomenclature), showing also their options or values. The parameters are the same when GP is used both as a classifier and as a feature selector. The values of some parameters are kept as the usual values set in HL (such as "Model Creator" and "Solution Creator"), while the others (such as "Maximum Length" and "Maximum Depth" in the evolved trees, "Mutation Probability", "Maximum Generations", "Population Size" and "Tournament window size") were experimentally adjusted with the objective of obtaining BP models with high performance in the classification (considering the 1000 independent GP runs, used when GP is used both as a feature selector and as a classifier, Sections 3.2 and 4.3). Likewise, the function set was chosen experimentally, since with the basic arithmetic functions (+, -, *, /) high-performance GP solutions were obtained in the classification.

Each of the methods of feature selection by dimensionality reduction (categorized by filter or wrapper and including GP by relative frequency) was evaluated using the classification methods divided into the following groups: single methods, ensemble methods and GP. The results will focus on the comparison between the use of selected features with respect to the use of all features in the classifiers, as well as on the comparison of the performance of the feature selection methods with the

Table 1
GP parameters.

GP Parameter	Value/option
Evaluator	Mean squared error
Solution Creator	Probabilistic Tree Creator
Tree Grammar	Arithmetic Functions (+, -, * and /).
Maximum Depth	10 (maximum tree depth)
Maximum Length	100 (maximum number of tree nodes)
Maximum Generations	100
Mutation Probability	15%
Population Size	1500
Selector	Tournament - Window size 8 (used in crossover and mutation)
Elites	1 (the best solution is maintained in the population over generations.)
Crossover	At the crossing point, it is used a Subtree Swapping Crossover
Mutation	Multi Symbolic Expression Tree Manipulator (it allows different mutation types)
Model Creator	Accuracy Maximizing Thresholds (the optimized solution is returned as the one that uses a classification threshold that maximizes accuracy and considering the training set)

different classifiers.

For reference, when using GP as FS with the HL environment, each independent GP run with the considered setup requires an average computational time of 76 s (using Model 5, which has the largest training size, Section 4.1.3), on a platform with an Intel i7–7700 processor and 24 GB of RAM. As noted above in Section 3.2.2, this implies a large computational time when running 1000 independent GPs. However, the computational time of the independent runs can be reduced since HL allows executing in parallel the independent GP runs on the available processor threads.

Finally, the performance measure considered in the following results is the Area Under the ROC Curve (AUC). The AUC measures the two-dimensional area under the full ROC curve, providing an aggregate measure of performance at all possible classification thresholds. It is one of the most common measures in the comparative analysis of business failure prediction. Therefore, this measure – AUC – is the one that will be used to evaluate the different methods at different temporal horizons. To obtain the corresponding AUC, the Weka software and its default settings have been used for each of the methods, except in the case of those corresponding to GP, which have been performed with HeuristicLab.

In this AUC measure and for the comparison between classifiers, the possible stochasticity in the training of classifiers must be considered. Stochastic classification methods could be defined as those that incorporate some element of randomness in their learning process (this may be due to several factors, such as random selection of training data, random selection of model parameters or the use of stochastic optimization techniques). Among the selected classification methods there are some that are non-stochastic or deterministic (e.g., QDA), some have stochasticity in the learning optimization technique (e.g., MLP) and some that, depending on the parameters selected (which depend partly on the software employed) can be used as stochastic or deterministic (e.g., LibSVM). As indicated above, for each of the selected classification methods, the default parameter configuration of the Weka software has been used. This leads to the following classification methods being considered stochastic in this work: MLP, JRip, Bagging, Decorate, Random Forest, Random Subspace and AdaBoostM1. On the contrary, J48, QDA, Logistic, Naive Bayes, LibSVM, KStar and Vote8, are considered deterministic.

The incorporation of stochastic classification methods implies the need to establish a criterion to evaluate their performance. In this paper we have sampled the AUC obtained in each temporal horizon (Models 1, 5 and 9) and for each of the tuples defined by the classification method (J48, MLP, …) and the variable selection method (CFS, Consistency, …). Each classification method with stochasticity was trained with 30 independent runs and using the features provided by each FS method (or all features as will be discussed below), calculating the corresponding AUC for each independent training. For each of these AUC samples, a confidence interval for the average AUC has been obtained with a confidence level set at 99%. The upper limit of this confidence interval for the average AUC is the reference to be used for the comparison of the performances in the case of any of the methods indicated as stochastic. This upper limit of the confidence interval is a homogeneous approximation of the maximum average performance that could be expected from the stochastic model in each of the cases.

Moreover, based on the upper limit of the confidence interval above, it is possible to calculate an estimated maximum value. This would be done by calculating the population standard deviation (based on the sample standard deviation) and calculating the result of adding two standard deviations to the upper limit. This yields a maximum value with a confidence greater than 99.9%.[1]

5.2. Results with the different classification methods (except GP)

The following tables (Table 2, Table 3 and Table 4) show, for each of the prediction temporal horizons (it should be remembered that Models 1, 5 and 9 refer to BP models with a prediction horizon of 1, 5 and 9 years before failure), the AUC obtained by each of the feature selection methods with each of the classification methods (with the exception of GP which is analyzed separately). As explained above, for a classification method with stochasticity, the AUC shown in the tables corresponds to the upper limit of its confidence interval for the average AUC (99% confidence level) after several independent training runs, while for a classification method without stochasticity it corresponds to its AUC after training the model.

The results corresponding to GP as a feature selector are identified in all tables as GPFS. Also included as a reference in the tables is the AUC obtained when using the initial set of unreduced input variables with each of the classification methods ("TotalVar"). The maximums per row (of each classification method) are highlighted in bold and the maximums per column (per feature selector method) are highlighted in gray fill.

As shown in the tables, the AUCs decrease as the prediction temporal horizon increases. It should be taken into account that, at present, studies with temporal horizons longer than 5 years are scarce and that the problem of deterioration of the forecasting power of BP models is still a minority field of research. In this particular field of deterioration of predictive power, the works of Matenda et al. [19], Zambrano Farias et al. [74] and, especially, Altman et al. [6,41] provide a detailed overview of the research progress and results obtained by various methods. However, addressing this deterioration is not the focus of this study.

Based on these results, the effectiveness and performance (as defined below) of the different methods are analyzed.

5.2.1. Effectiveness of feature selection methods

We define a feature selection method as "effective" if it improves classification results over the use of all features. However, it is observed that dimensionality reduction by feature selection does not always improve the results obtained when the classifier uses the totality of variables. That is, $AUC_{t,x,a} < AUC_{t,0,a}$ (t is the prediction horizon, x is the feature selection method, 0 refers to TotalVar and a is the classification method).

Calculating the number of possible scenarios (3 temporal horizons, 10 feature selection methods and 14 classification methods, not including GP as a classifier) gives 420 scenarios, of which, in 176, feature selection is ineffective (i.e., $AUC_{t,x,a} < AUC_{t,0,a}$). In more detail, the most effective feature selection methods are as follows:

[1] It must be taken into account that the probability that the average is less than or equal to the calculated upper limit is 99.5% – since the confidence level is 99% –. Similarly, since the probability that a value is in the interval defined by the average and two standard deviations is 95.44%, the probability that the value is less than the average plus two standard deviations is 97.72%. With these figures, the probability that a value is higher than the calculated upper limit of the average AUC plus two standard deviations – estimated maximum value – is (1-0.9950)*(1-0.9772)=0.000114 and the probability that it is lower than the aforementioned reference will be (1-0.000114)=0.999886

Table 2
AUC by feature selector (columns) and classifier (rows) - Model 1. "TotalVar" refers to the use of all input variables. Values in bold: best AUC per row (of each classification method, including TotalVar). Values in grey fill: best AUC per column (of each feature selector method). For classification methods labeled with a final "*" (with stochasticity in training), the value corresponds to the upper limit of its confidence interval of the average AUC.

	Model 1													
	CFS	Consistency	Chi-squared	Correlation	SVM	InfoGain	RelieF	Wrapper Naive Bayes	Wrapper J48	GPFS	TotalVar	Average filter selection methods	Average wrapper selection methods	Average selection methods without GPFS
J48	0.77	0.75	**0.82**	0.79	0.78	**0.82**	0.79	0.81	0.81	0.80	0.80	0.79	0.81	0.79
MLP*	0.91	0.90	0.89	0.89	0.90	0.89	**0.92**	0.87	0.84	0.90	0.90	0.90	0.85	0.89
JRip*	0.79	**0.81**	0.80	0.80	0.79	0.79	0.81	0.79	0.81	0.78	0.79	0.80	0.80	0.80
QDA	0.85	**0.86**	0.83	0.81	0.82	0.83	0.85	0.83	0.84	0.74	0.83	0.84	0.84	0.84
Logistic	0.84	**0.89**	0.79	0.83	0.84	0.80	0.82	0.87	0.80	0.85	0.77	0.83	0.83	0.83
Naive Bayes	**0.91**	0.86	0.89	0.89	0.91	0.89	0.90	0.89	0.85	0.89	0.87	0.89	0.87	0.89
LibSVM	0.86	0.82	0.82	0.84	**0.86**	0.82	0.83	0.78	0.78	0.84	0.83	0.83	0.78	0.82
KStar	0.90	0.90	0.89	0.89	**0.91**	0.89	0.90	0.89	0.87	0.85	0.78	0.90	0.88	0.89
Bagging*	**0.90**	0.89	0.89	0.90	0.89	0.89	0.90	0.88	0.86	0.89	0.89	0.89	0.87	0.89
Decorate*	0.92	0.91	0.91	0.91	0.91	0.91	**0.92**	0.89	0.86	0.92	0.92	0.91	0.88	0.91
Random Forest*	0.92	0.91	0.91	0.92	0.92	0.91	**0.92**	0.90	0.86	0.92	0.92	0.91	0.88	0.91
Vote8	0.92	0.91	0.89	0.91	**0.92**	0.90	0.92	0.90	0.85	0.92	0.92	0.91	0.87	0.90
Random Subspace*	0.89	**0.90**	0.89	0.89	0.89	0.89	0.90	0.88	0.85	0.89	0.89	0.89	0.87	0.89
AdaBoostM1*	0.89	0.89	0.89	**0.90**	0.88	0.89	0.90	0.89	0.85	0.88	0.89	0.89	0.87	0.89

Table 3
AUC by feature selector (columns) and classifier (rows) - Model 5. "TotalVar" refers to the use of all input variables. Values in bold: best AUC per row (of each classification method, including TotalVar). Values in grey fill: best AUC per column (of each feature selector method). For classification methods labeled with a final "*" (with stochasticity in training), the value corresponds to the upper limit of its confidence interval of the average AUC.

	Model 5													
	CFS	Consistency	Chi-squared	Correlation	SVM	InfoGain	RelieF	Wrapper Naive Bayes	Wrapper J48	GPFS	TotalVar	Average filter selection methods	Average wrapper selection methods	Average selection methods without GPFS
J48	0.67	0.61	**0.73**	0.68	0.66	**0.73**	0.68	0.71	0.63	0.70	0.65	0.68	0.67	0.68
MLP*	0.75	0.76	0.70	0.71	0.75	0.71	0.74	0.74	0.75	0.75	**0.76**	0.73	0.75	0.73
JRip*	0.70	0.69	0.69	0.68	0.67	0.69	**0.70**	0.70	0.69	0.70	0.70	0.69	0.70	0.69
QDA	**0.78**	0.77	0.75	0.76	0.74	0.75	0.78	0.74	0.74	0.74	0.68	0.76	0.74	0.76
Logistic	0.78	0.79	0.75	0.76	0.77	0.75	0.77	0.72	0.72	**0.79**	0.69	0.77	0.72	0.76
Naive Bayes	**0.78**	0.78	0.73	0.74	0.74	0.73	0.77	0.71	0.73	0.76	0.76	0.75	0.72	0.75
LibSVM	0.73	**0.74**	0.67	0.71	0.69	0.67	0.72	0.68	0.68	0.73	0.72	0.70	0.68	0.70
KStar	0.74	**0.75**	0.69	0.67	0.73	0.69	0.73	0.71	0.71	0.74	0.73	0.71	0.71	0.71
Bagging*	0.79	0.78	0.77	0.76	0.76	0.77	0.79	0.76	0.77	**0.79**	0.79	0.77	0.76	0.77
Decorate*	0.79	0.78	0.77	0.76	0.77	0.77	0.80	0.76	0.76	**0.80**	0.79	0.78	0.76	0.77
Random Forest*	0.81	0.80	0.77	0.78	0.79	0.77	0.81	0.77	0.77	**0.82**	0.81	0.79	0.77	0.79
Vote8	**0.81**	0.79	0.77	0.76	0.80	0.77	0.79	0.76	0.76	0.80	0.80	0.78	0.76	0.78
Random Subspace*	0.79	0.78	0.76	0.76	0.76	0.76	**0.79**	0.76	0.76	0.79	0.78	0.77	0.76	0.77
AdaBoostM1*	0.77	0.77	0.75	0.76	0.74	0.75	0.76	0.75	0.75	**0.77**	0.76	0.76	0.75	0.76

Table 4
AUC by feature selector (columns) and classifier (rows) - Model 9. "TotalVar" refers to the use of all input variables. Values in bold: best AUC per row (of each classification method, including TotalVar). Values in grey fill: best AUC per column (of each feature selector method). For classification methods labeled with a final "*" (with stochasticity in training), the value corresponds to the upper limit of its confidence interval of the average AUC.

	Model 9													
	CFS	Consistency	Chi-squared	Correlation	SVM	InfoGain	ReliefF	Wrapper Naive Bayes	Wrapper J48	GPFS	TotalVar	Average filter selection methods	Average wrapper selection methods	Average selection methods without GPFS
J48	0.61	0.61	0.56	0.61	0.56	0.56	0.61	0.58	0.59	0.59	**0.61**	0.59	0.59	0.59
MLP*	**0.64**	0.62	0.58	0.63	0.58	0.57	0.59	0.64	0.64	0.62	0.58	0.60	0.64	0.61
JRip*	0.60	0.60	0.60	0.60	0.60	0.60	0.60	**0.60**	0.60	0.60	0.60	0.60	0.60	0.60
QDA	**0.65**	0.64	0.58	0.59	0.54	0.58	0.60	0.63	0.62	0.60	0.56	0.60	0.62	0.60
Logistic	0.66	0.65	0.61	0.61	0.58	0.61	0.63	0.64	0.63	**0.68**	0.58	0.62	0.63	0.62
Naive Bayes	0.65	0.64	0.64	0.63	0.61	0.64	**0.66**	0.64	0.62	0.65	0.63	0.64	0.63	0.64
LibSVM	0.62	0.61	0.61	0.60	0.60	0.61	0.62	**0.63**	0.61	0.60	0.57	0.61	0.62	0.61
KStar	**0.64**	0.63	0.60	0.61	0.59	0.60	0.62	0.57	0.63	0.57	0.58	0.61	0.60	0.61
Bagging*	0.65	0.65	0.64	0.65	0.64	0.64	0.65	0.63	0.64	**0.65**	0.64	0.65	0.64	0.64
Decorate*	**0.66**	0.65	0.65	0.65	0.63	0.65	0.65	0.64	0.64	0.65	0.65	0.65	0.64	0.65
Random Forest*	**0.66**	0.65	0.65	0.65	0.64	0.65	0.65	0.61	0.61	0.66	0.65	0.65	0.61	0.64
Vote8	**0.67**	0.66	0.63	0.65	0.62	0.64	0.65	0.63	0.64	0.64	0.64	0.64	0.64	0.64
Random Subspace*	0.65	0.65	0.65	0.65	0.65	0.65	**0.65**	0.65	0.64	0.65	0.65	0.65	0.64	0.65
AdaBoostM1*	0.65	0.65	0.64	0.65	0.64	0.64	**0.65**	0.63	0.63	0.65	0.63	0.65	0.63	0.64

- CFS: ineffective in 6 out of 42 scenarios (3 prediction temporal horizons, 14 classification methods)[2] .
- RelieF: ineffective in 7 scenarios.
- GPFS: ineffective in 8 scenarios.

At the other extreme are:

- Wrapper J48: ineffective in 27 scenarios out of 42.
- Wrapper Naive Bayes: ineffective in 26 scenarios
- SVM and InfoGain: ineffective in 24 scenarios.

Moreover, it is also observed that not all the classification methods analyzed react equally to feature selection. The 176 scenarios, in which feature selection methods are ineffective, are not evenly distributed among the different classification methods (excluding GP). There are classification methods in which most of the subsets provided by the different feature selection methods (explanatory variables of the model) present higher AUC than those obtained with the same classifier and the totality of input variables. As an example, the Logistic classification method presents 0 inefficient scenarios (out of a total of 30, given by 3 temporal horizons and 10 feature selection methods) and QDA 4. At the opposite extreme are Decorate (21 ineffective scenarios), MLP and Vote8 (each 18 ineffective scenarios) and RandomForest and J48 (17 ineffective scenarios). This suggests that attention should be focused on the 2-tuple defined by the feature selection method and the classification method, and not only on the first term of the tuple. It should be remembered that the above refers exclusively to the fact that $AUC_{t,x,a} < AUC_{t,0,a}$, which says absolutely nothing about the value achieved by $AUC_{t,x,a}$, or that $AUC_{t,x,a} > AUC_{t,x,b}$ (where a and b are different classification methods). At this point, we analyze efficiency and not comparative performance between different classifiers (which is analyzed in the following subsection).

Table 5 summarizes the average behavior (ineffectiveness percentage) of the different tuples (feature selection method; classification method) in the three temporal horizons and at the aggregate level of feature selector/classifier types. The column "Average selection methods" shows the average of each classifier considering all of the feature selection methods. In terms of ineffectiveness, it is observed that filter-type feature selection methods perform better than wrapper-type methods and both perform worse with ensemble-type classifiers than with single classifiers. That is, for ensemble classifiers, the selection provided by many feature selection methods is not effective in increasing their classification performance. On the contrary, GPFS has the best average effectiveness with both single and ensemble classifiers. Thus, the selection provided by GPFS (totally different from analyzing the intra and interclass variation of a variable as in filter methods, or based on how well it works with a particular classifier, as in wrapper methods) certainly does provide a subset of features that increases

Table 5
Percentage of ineffectiveness at the aggregate level by feature selector type (columns) and classifier type (rows), combining the results of Models 1, 5 and 9.

	Filter	Wrapper	GPFS	Average selection methods
Average single methods	32.7%	41.7%	20.8%	33.3%
Average ensemble methods	47.6%	91.7%	16.7%	53.3%

[2] It should be noted when analyzing the data in Tables 2, 3 and 4 that they have been calculated to 3 decimal places, although they are only shown to 2 decimal places.

effectiveness when used by many different classifiers.

5.2.2. Performance of feature selection methods

Now, for a given classifier, the AUCs obtained with the different feature selection methods are compared to obtain a measure of the performance of each selection approach. First, given a temporal horizon t and a classification method a, the AUC obtained by GPFS (GP-based selection proposal) is compared with the averages of the AUC obtained by the other feature selection methods (CFS, Consistency, etc.) grouped by type, which can provide that performance measure of the GPFS selection method. These averages are shown in three columns in Tables 2–4 at each prediction temporal horizon. The result of the comparison of these values are:

- The AUC obtained with GPFS and the average AUC obtained by the other 7 filter-type selection methods show that in 29 out of 42 scenarios (3 prediction temporal horizons, 14 classification methods) the AUC obtained by GPFS is higher than the average of the selection methods of filter type.
- The AUC obtained with GPFS and the average AUC obtained by the other 2 wrapper-type selection methods show that in 33 out of 42 scenarios the AUC obtained by GPFS is higher than the average of the wrapper methods.
- In summary, the AUC obtained by GPFS is higher than the average obtained by all feature selection methods in 31 of the 42 scenarios.

After this first consideration, the following values are analyzed: i) the number of times (out of 42 scenarios: 3 prediction temporal horizons and 14 classification methods) in which each of the different selection methods obtains the highest AUC (with each of the classifiers) and ii) the number of times in which each of the selection methods is among the top three with the highest AUC.

In terms of the number of times each of the selection methods obtained the highest AUC in the 42 scenarios, the positions are as follows:

- CFS obtained the highest AUC in 11 scenarios (6 with single classifiers and 5 with ensemble classifiers).
- Relief obtained the highest AUC in 9 scenarios (4 with single classifiers and 5 with ensemble classifiers).
- GPFS obtained the highest AUC in 7 scenarios (2 with single classifiers and 5 with ensemble classifiers).
- Consistency obtained the highest AUC in 7 scenarios (6 with single classifiers and 1 with ensemble classifiers).

When evaluating the number of times each method is among the three with the highest AUC, the positions are as follows:

- CFS: among the top 3 with the highest AUC in 35 (out of 42 scenarios).
- Relief: among the top 3 with the highest AUC in 26 scenarios.
- Consistency: among the top 3 with the highest AUC in 20 scenarios.
- GPFS: among the top 3 with the highest AUC in 18 scenarios.

This shows the high effectiveness and performance of GP as a feature selector. More comments on this are discussed below (Section 6).

In this point, two aspects of the performance of the ensemble classifiers should also be highlighted:

- Tables 2–4 show that, given a feature selector (CFS, Consistency, etc.), the ensemble classifiers perform better than the single ones in 26 out of 30 (10 feature selector methods and 3 temporal horizons). This fact can be generalized when using the total set of variables (See Tables 2–4, values in grey fill). These results ratify the ones of Dietterich [75].
- Ensemble classifiers (even if they perform intrinsic feature selection) can improve their performance when a feature selection method is used [67,68]. Without exceptions in the 18 scenarios considered (6 ensemble classifiers and 3 temporal horizons), the best performance of each ensemble classifier is obtained in conjunction with a feature selection method. That is, in all cases, ensemble classifiers obtain their maximum AUC with a subset of selected features and not with the totality of the variables, although, when using the subsets provided by most feature selection methods, their performance decreases with respect to the use of all the variables (as noted above in the comments to Table 5). That is, ensemble classifiers are very sensitive to the prior feature selection applied.

5.3. Results with genetic programming as a classification method

As indicated (Section 4.3), to perform the comparison of the different methods of dimensionality reduction, having GP as the classification method, an experiment (1000 GP runs) with the same profile is performed for each of the prediction temporal horizons, except for the set of input variables given by each of the 10 feature selection methods. The stochastic nature of GP leads to experiments with a high number of runs and implies that a very good solution can be obtained by chance. Hence, it is important not only to take into account the best solution (the highest AUC in this case), but also to verify that good solutions can be obtained recurrently. Therefore, the average AUC of the 5 best solutions obtained in the experiment is evaluated together with the best solution.

The results obtained with GP as the classification method and each of the feature selection methods in each prediction temporal horizon are shown in Table 6. Each column of the table specifies in bold with which set of selected variables the best AUC value was obtained. It is observed that the proposed feature selection (GPFS), based on GP, offers – with the best solution – the best result in 1 prediction temporal horizon (Model 9) while RelieF offers the best result in Model 1 and Model 5. In the case of assessing the average of the 5 best solutions, it is observed that the proposed feature selection (GPFS) offers the best result in 2 prediction temporal horizons (Models 1 and 9) while RelieF offers the best result in Model 5.

It is also observed that, with GP as the classification method, there are still ineffective feature selection methods. Of the 30 tuples (prediction temporal horizon, feature selection method) used with GP as the classification method, 18 (60%) are ineffective, in that better results (AUC of the best solution) are obtained with the total variables than with the subset proposed by the feature selection method (Table 6). That is, $AUC_{t,x,GP} < AUC_{t,0,GP}$ (t is the prediction horizon, x is the feature selection method and 0 is TotalVar, and GP is the classification method). However, note that the selection of GPFS is effective since with the selected features, GP (as a classifier) performs better with respect to the use of all features and at all three temporal horizons. Table 7 summarizes the results by type of feature selection method.

Table 6
Classification with genetic programming – AUC of the best solution and average AUC of the 5 best solutions. Values in bold: selected feature set that provides the best AUC with GP as a classifier (best value per column).

	GP - best solution			GP - average 5 best solutions		
	Model 1	Model 5	Model 9	Model 1	Model 5	Model 9
CFS	0.94	0.81	0.68	0.94	0.80	0.67
Consistency	0.93	0.80	0.68	0.93	0.79	0.68
Chi-squared	0.92	0.79	0.69	0.92	0.79	0.68
Correlation	0.94	0.80	0.68	0.94	0.79	0.68
SVM	0.94	0.79	0.68	0.93	0.79	0.67
InfoGain	0.94	0.80	0.67	0.94	0.79	0.67
RelieF	**0.95**	**0.81**	0.69	0.94	**0.81**	0.69
Wrapper Naive Bayes	0.93	0.77	0.68	0.93	0.77	0.68
Wrapper J48	0.91	0.78	0.69	0.91	0.78	0.69
GPFS	0.94	0.81	**0.70**	**0.94**	0.81	**0.69**
TotalVar	0.94	0.80	0.69	0.94	0.80	0.68

Table 7
Percentage of ineffectiveness at the aggregate level by feature selector type (columns) and using GP as a classifier - best solution (row), combining the results of Models 1, 5 and 9.

	Filter	Wrapper	GPFS	Average selection method
GP as a classifier - best solution	61.9%	83.3%	0.0%	60.0%

Now, we consider the comparison between the different classification methods versus GP as a classifier. For each prediction temporal horizon (1, 5 and 9 years prior to failure) and each feature selection method (CFS, Consistency, etc.), we compare the results of the different classification methods (J48, MLP, etc.) with those corresponding to the application of GP (AUC of the best solution). Therefore, in this case, there are 420 possible scenarios.

Comparing the values in Table 6 with the corresponding values in Tables 2–4, it is observed that in 415 scenarios (99%) the AUC of the best GP solutions is higher than the AUC of the other classifiers used. If instead of the upper limit of the average confidence interval for the classification methods with training stochasticity (shown in Tables 2–4), the estimated maximum AUC values (defined as the average that corresponds to the upper limit of the confidence interval + 2 population standard deviations, Section 5.1) were considered, in 92% of the cases (387 scenarios out of 420), $AUC_{t,x,a} < AUC_{t,x,GP}$. Also considering the estimated maximum AUC, if instead of using the AUC of the best solution as a measure, the average AUC of the 5 best solutions is used, there would be 373 scenarios (89%) in which the AUC of the average 5 GP solutions is higher than the estimated maximum AUC of the other classifiers used. In other words, it is remarkable the fact that using GP as a classifier consistently improves the results provided by the other classification methods.

Finally, Fig. 2 includes an example of an evolved GP classifier. It corresponds to the best solution (best AUC) of Model 1. The evolved tree is shown in the hierarchical format provided by HeuristisLab. In Fig. 2, the input variables are denoted in HL as log_rXX, where "XX" refers to the specific input variable (out of the 97 indicated in Section 4.1.2) and "log" corresponds to the logistic distribution normalization used on the variables (Section 4.1.2). These explanatory variables used in this solution are:

r20	Shareholder funds / Total assets
r40	Operating income / Total assets
r43	Income from ordinary activities / Total assets
r63	Financial expenses / Total sales
r69	Financial expenses / Total debt
r82	Variation of purchases/ Variation of trade payables, being Variation (X) = X(t)/X(t-1), t refers to fiscal year
r86	DSR(t) /DSR(t-1) DSR = Days Sales Receivables
r89	Log (Total assets)
r91	(Short-term debt (t) - Short-term debt (t-1)) / Total assets

As can be seen, most of the variables used in the model refer to the short-term, either because they include income statement accounts or because they are time variations (t vs. t-1). Only one variable (r20) refers to balance sheet structure and another (r89) to size. This is perfectly consistent with the temporal horizon of the prediction model and with the behavior shown by the rest of the solutions with this temporal horizon (see Section 5.4 below).

In addition, HeuristicLab allows a subsequent manual pruning of this tree (not shown here, which may slightly change the AUC), allowing to have a simpler classification tree for easier interpretability.

5.4. Analysis of selected features with the dimensionality reduction methods

Each of the proposed dimensionality reduction methods (CFS, Consistency, etc.) selects a subset of features (among the initial variables) that become the explanatory variables of the model created by each classification method at each temporal horizon. When analyzing these subsets, it is necessary to remember the following limitations:

- As stated by Viegas et al. [38], each feature selection approach may select different features, as they use different criteria in their selection process. Moreover, "These different sets may contain good discriminative features as well as not so relevant ones. Furthermore, good features selected by one method will not necessarily be selected by a different one" [38].
- Only the appearance of a given variable in the different subsets of variables that determine the feature selection methods will be considered, but not the relevance of the variables in the final solution of the given models. A variable may appear in a model, but the relative relevance to the classification may be higher or lower.
- No ranking of the selected variables is used since there are variable selection methods that provide such ranking information (e.g.,

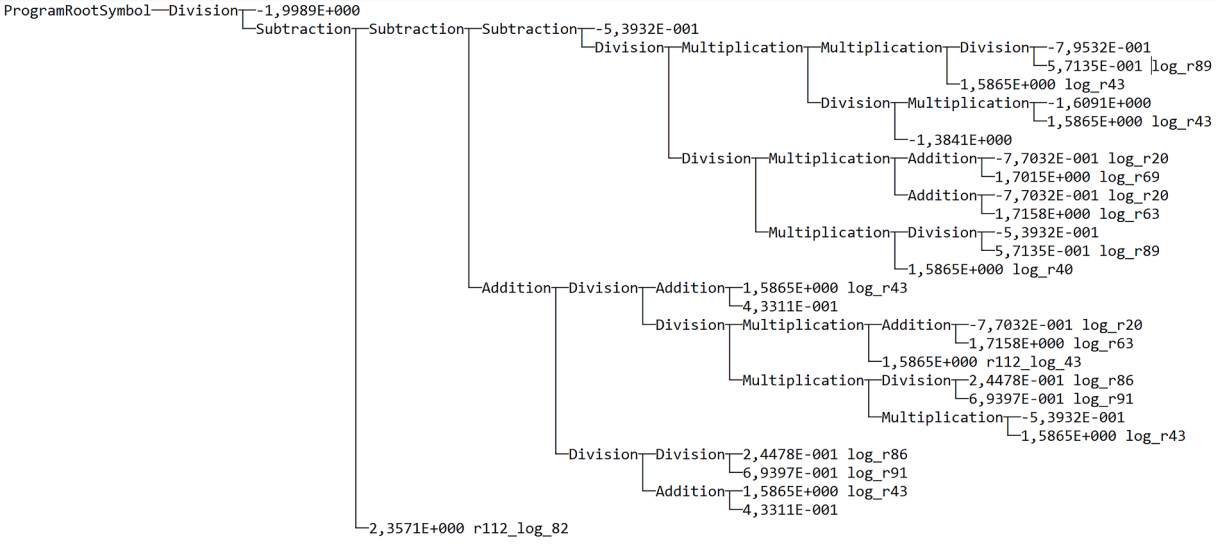


Fig. 2. Best evolved GP solution of Model 1.

Table 8
Number of variables selected per model.

	N. of distinct variables in total methods	N. of distinct variables in total methods without GPFS	N. distinct variables in GPFS	N. distinct variables contributed by GPFS	Sum of selected variables in total methods
Model 1	70	45	46	25	177
Model 5	54	49	31	5	183
Model 9	49	46	16	3	136

InfoGain, RelieF and GPFS), while others do not (e.g., CFS and Consistency).
- There are subjective parameters that condition the selected variables (e.g.: set p-value <= 0.05 when using GP as a variable selection method).

Despite these limitations, it may be relevant to analyze some of the results. First, an analysis of the number of variables selected by the different FS methods can be confusing, since 6 of the 10 methods provide a ranking of the variables. In five of these methods, 20 variables are selected from the total of 97 (Section 5.1). In addition, in the case of GPFS (which also provides that ranking), the number of variables depends on the threshold (p-value) used. Because of this, the analysis that takes into account the number of distinct variables selected by the different FS methods is more informative. In this regard, Table 8 contains detailed information on the distinct selected variables. The meaning of the different columns in Table 8 is as follows:

- "N. of distinct variables in total methods" indicates the number of variables that have been selected at least 1 time in any of the methods. It is computed only once for each variable.
- "N. of distinct variables in total methods without GPFS" is the same concept as above, but does not include GPFS among the variable selection methods.
- "N. distinct variables in GPFS" indicates the number of variables selected by GPFS.
- "N. distinct variables contributed by GPFS" indicates the variables selected by GPFS and which were not selected in any other method.
- "Sum of selected variables in total methods" indicates the sum of the number of variables selected by the different feature selection methods. The value indicated in this column is greater than the number of initial variables considered (97), since the same variable could have been selected by different methods, computing several times in this final value of the column.

It can be seen that the number of different variables used by the total number of feature selection methods is quite high (recall that there are a total of 97 initial variables). This is a sign of the lack of consensus among the different variable selection methods as to which variables are relevant within the total number of variables. It is also observed that the total number of different variables selected decreases as the prediction temporal horizon increases. That is, the potential predictors of BP are smaller as the temporal horizon is longer.

It is observed that the different selection methods (without GPFS) remain at a fairly constant number of different variables selected over the different temporal horizons (between 45 and 49). The same can be seen considering the average size of the selected subsets (measured as "Sum of selected variables in total methods"/"Number of selection methods"), although falling in the 9-year temporal horizon. However, focusing on the behavior of GPFS, this GP-based selection method shifts from using 46 variables (Model 1) to using 16 (Model 9). In addition, it selects 25 different variables than the other methods in Model 1 (although this number decreases to 5 and 3 in Models 5 and 9, respectively). This means that GP clearly focuses on fewer features for possible BP models over the longest temporal horizons.

Table 9
Relative frequency of the groups of variables selected in each model. Values in bold: the 5 highest relative frequencies for each model.

	Model 1	Model 5	Model 9
LIQUIDITY AND SOLVENCY	**13.56%**	**20.22%**	**21.32%**
FINANCIAL STRUCTURE	9.04%	**19.67%**	**22.79%**
PROFITABILITY	**22.60%**	**14.75%**	**11.76%**
EFFICIENCY	**10.73%**	2.73%	1.47%
TURNOVER	2.26%	**9.84%**	**10.29%**
VARIATIONS IN MAGNITUDES	2.82%	2.73%	0.74%
CONTRIBUTION	3.95%	4.37%	5.88%
INTEREST EXPENSE	**16.38%**	**13.11%**	**18.38%**
SIZE	5.08%	8.20%	1.47%
GROWTH	0.56%	0.00%	0.00%
CHANGES IN RATIOS	1.13%	0.55%	0.74%
DEGREE OF DECOMPOSITION	0.00%	0.00%	0.74%
PRODUCTIVITY	2.26%	1.64%	0.00%
FRAUD	**9.60%**	2.19%	4.41%

It is also possible to analyze the frequency of selection of each variable (the number of times each variable is selected by one of the variable selection methods) by grouping the selected variables into the categories indicated in Section 4.1.2. Table 9 provides the relative frequency (in percentages) of the variables selected by the totality of dimensionality reduction methods, grouped by each of the categories. The 5 highest relative frequencies for each model are indicated in bold. An analysis of the groups of ratios to which the five highest frequencies per model correspond shows that, beyond the short-term (Models 5 and 9), the relevant groups (based on the relative frequency of their appearance in any of the models) are the same (in no order of precedence): Liquidity and solvency, Financial structure, Profitability, Turnover and Interest expense. Of these, 3 are repeated as important in the very short-term (Model 1): Liquidity and solvency, Profitability and Interest expense. In addition, two new groups appear (in Model 1 with respect to Models 5 and 9): Efficiency and Fraud (replacing Financial structure and Turnover).

Therefore, the relevant groups in Model 1 focus exclusively on the short-term: some based on various aspects of the income statement, such as Profitability, Efficiency and Interest expense, another on Liquidity and Solvency and, finally, the Fraud group, which is based on year-on-year variations of certain items. On the other hand, the relevant groups in Models 5 and 9 include some that contain a strong incidence of variables with longer-term significance (Financial structure and Turnover). Consequently, selection methods do tend to focus on the appropriate variables for each specific temporal horizon.

6. Overall discussion of the results

The environment in which the different dimensionality reduction methods (by feature selection) were compared is characterized by the fact that the BP problem generally presents two relevant circumstances:

- An initial set of available input variables, with a high number of highly correlated variables and – probably – with redundant variables (e.g.: different transformations of a single original variable).
- The evaluation of the models is performed on a highly imbalanced test set, with a negative class (observations of non-failed firms) with a much larger number of examples than the positive class (observations of failed firms).

In no particular order of priority, there are some relevant aspects to pay attention to:

- The percentage of scenarios in which feature selection is ineffective is noteworthy, 43.1% (194 out of 450 scenarios, considering now 15 classification methods). This is especially important in some of the selection methods analyzed (Wrapper Naive Bayes, SVM, InfoGain and Wrapper J48, where the above percentage exceeds 50%, or Consistency, Chi-squared and Correlation where the percentage exceeds 40% but not 50%). This fact shows that feature selection is not fully effective, since it does not always achieve the expected effect after dimensionality reduction (at least if the expected effect is AUC improvement). This possibility is not new: as indicated by Liang et al. [13] in their work focused on financial distress prediction, depending on the techniques chosen, feature selection does not improve prediction performance in all cases.
- The reduction of the dimension of the initial set of input variables by the proposed feature selection method (GPFS) is a more efficient feature selection method than most of the methods analyzed (only 8 inefficient scenarios out of 45 analyzed, Sections 5.2.1 and 5.3). This makes it a reliable feature selection method, since the results obtained by the different prediction models when using its output (explanatory variables) will be – in most cases – better than the results of the prediction models obtained with the totality of input variables as explanatory variables.
  The above, together with the high performance (as defined in Section 5.2.2) obtained by the GP reduction method (GPFS) compared to other feature selection methods, make GPFS an excellent feature selection method.
- According to the results obtained, it is observed that – in the case of the BP addressed in this work – the feature selection methods by filtering clearly outperform the wrapper methods analyzed, which occupy the last positions in both categories (efficiency and performance). This poor performance of the analyzed wrapper methods occurs both when using single classifiers, ensemble classifiers or GP, even when using as a classifier the same method they have used for feature selection (Wrapper Naive Bayes and Wrapper J48). It should be noted that filtering methods tend to generalize better [18] than wrapper methods – which present better performance when modeling [16] – and the evaluation is being performed on the test set.
- GPFS presents its best performance in conjunction with the GP as a classifier, where its superiority over the other methods used is evident, since it presents the highest AUC in 2 of the prediction temporal horizons (Section 5.3).
- Considering the results of the comparison, in terms of performance, between the classifiers, in most cases GP has provided better classification results with respect to the other classification methods (as pointed out in Section 5.3). Therefore, it can be concluded that GP as a classification method is an excellent method that improves the results of practically any 2-tuple (temporal horizon, selection method).

On the other hand, with specific regard to the variables chosen by the different feature selection methods, there are a number of data that provide interesting indications. Briefly, they are as follows:

- The different feature selection methods do not converge on a subset of selected variables. The percentage of the number of distinct variables used by the methods out of the total number of input variables is sufficiently high (especially in Model 1 where the methods select 70 distinct variables out of a total of 97 input variables) to affirm the lack of consensus (Table 8 in Section 5.4).
- Diversity (understood as the number of different variables selected) decreases as the prediction horizon increases (Table 8). Long-term BP indicators (distinct variables selected by the models 5 and 9 years prior to failure) are less than in the short-term (1 year prior to failure). In other words, the predictors, the signals warning of failure, decrease as the prediction horizon does, which would make prediction more difficult and is in line with the deterioration of performance as the prediction temporal horizon increases.
- The number of different variables used by GPFS in each of the prediction models and the number of different variables added by GPFS over those provided by the other feature selection methods (Table 8) can be interpreted as a sign of adaptability (greater adaptation to the temporal horizon) compared to other models. This results in a lower probability, with GPFS, of losing relevant variables in the process of reducing the initial set of variables, as corroborated by the results shown in Section 5.
- Despite the wide variety of variables selected, there does seem to be a consensus among the different models (Models 1, 5 and 9) on the basic aspects that warn of BP. As detailed in Section 5.4, it has been shown that the selection methods do tend to focus on the appropriate variables for each specific temporal horizon (groups of variables focused on the short-term in Model 1 and variables with longer-term significance in Models 5 and 9). This implies that the selection of variables is indeed consistently focused on those with relevance for the temporal horizon considered.
- These aspects discussed can be analyzed in more detail if the relevance of each of the explanatory variables (and, therefore, of the groups) is also taken into account. However, this analysis would belong to the field of Explainable Artificial Intelligence (XAI) and is beyond the scope of this study.

7. Conclusions and future work

In this study, a new GP-based feature selection method within the BP domain was proposed. A comparative analysis of the proposed method with different feature selection methods has been performed, also considering different classifiers in the analysis. For this comparison, the AUC has been used and it has been carried out based on two aspects: efficiency and performance.

The two main contributions in our study are:

i) The proposed new GPFS method is based on the relative frequency of the presence of variables in the evolved trees/programs in the evolutionary process. For a correct detection of the relevant variables, the statistical significance of this frequency is taken into account. GPFS performs a global search in the feature space, the FS process is inherent to the GP evolutionary process, is context sensitive and provides a ranking of the most relevant variables.

ii) The results indicate that the proposed method (using Genetic Programming as a variable selection method) is superior to the most tested and widely used methods analyzed. Superiority was tested in terms of efficiency (i.e., the FS method improves classification results over using all features), and in terms of performance (i.e., for a given classifier, the classification performances – AUC – obtained with the different feature selection methods are compared). Furthermore, the superiority increases if Genetic Programming is also used as a classification method.

On the basis of this study, future work could be carried out in three directions:

- GPFS parameter selection. In this work, we have chosen those variables with p-value<0.05 (referring to the relative frequency of the different variables in the GP runs that lead to obtaining the best solutions). Note that, in addition to testing different p-values, another selection criterion could be used, e.g., the mean and/or median (to qualify the possible relevance of the different variables).
- One of the risks of feature selection is to lose information on relevant variables. Variable selection is a non-monotonic problem, in the sense indicated by du Jardin [5]: “Variable selection remains difficult because it is often non-monotonic. Indeed, the best subset of p variables rarely includes the best subset of q variables, where $q<p$”.

In the proposal presented in this work (GPFS), dimensionality reduction has been performed according to the statistical significance of the relative frequency of occurrence (in the evolved trees) of each variable of the input set (p-value<0.05) and the results show that it is a good approximation.

The second methodology proposed could be a stepwise filtering process, which would attempt a progressive reduction of the dimension of the set of input variables by eliminating, in each of the stages, those variables with clear signs of irrelevance (e.g.: p-value>0.333). In other words, the central idea would be to gradually reduce the number of input variables while minimizing the risk of eliminating relevant variables. A p-value<=0.333 means that the variables whose probability of being significant is >= 66.67% will be selected in each iteration or filtering stage, which is a very lax criterion, as opposed to the usual 95–99%.

- Another possible aspect to explore would be to replace statistical significance with some more precise measure of the real impact of the variable on the classification. This line of action is "a priori" more complicated since classifiers will often supply non-linear models in which the calculation of a single value of impact per variable will be complex.

As a final conclusion and as the main contribution of our work, it is worth noting that the proposal analyzed in this study (using Genetic Programming as a method of feature selection), becomes – in terms of the combination of efficiency and performance – a feature selection method superior to the rest of the methods analyzed, highlighting that the proposed variable selection method provides a performance that stands out over the performance of the other variable selection methods when GP is also used as a classification method.

CRediT authorship contribution statement

Ángel Beade: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Manuel Rodríguez:** Writing – review & editing, Validation, Supervision, Investigation, Formal analysis, Conceptualization. **José Santos:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This study was funded by the European Union (European Regional Development Fund - Galicia 2014-2020 Program) and Xunta de Galicia, with grants GPC ED431B 2022/33 and CITIC (ED431G 2019/01), and by the Spanish Ministry of Science and Innovation (project PID2020-116201GB-I00).

References

[1] A. Volkov, D.F. Benoit, D. Van den Poel, Incorporating sequential information in bankruptcy prediction with predictors based on Markov for discrimination, Decis. Support. Syst. 98 (2017) 59–68, https://doi.org/10.1016/j.dss.2017.04.008.
[2] C.-F. Tsai, K.-L. Sue, Y.-H. Hu, A. Chiu, Combining feature selection, instance selection, and ensemble classification techniques for improved financial distress prediction, J. Bus. Res. 130 (2021) 200–209, https://doi.org/10.1016/j.jbusres.2021.03.018.
[3] P. Barnes, The analysis and use of financial ratios, J. Bus. Finance Account. 14 (1987) 449.
[4] E.K. Laitinen, Financial ratios and different failure processes, J. Bus. Finance Account. 18 (1991) 649–673, https://doi.org/10.1111/j.1468-5957.1991.tb00231.x.
[5] P. du Jardin, Bankruptcy prediction models: how to choose the most relevant variables?, (2009) 39–46. https://mpra.ub.uni-muenchen.de/44380/(accessed July 3, 2020).
[6] E.I. Altman, M. Iwanicz-Drozdowska, E.K. Laitinen, A. Suvas, A race for long horizon bankruptcy prediction, Appl. Econ. 52 (2020) 4092–4111, https://doi.org/10.1080/00036846.2020.1730762.
[7] P. du Jardin, Dynamic self-organizing feature map-based models applied to bankruptcy prediction, (2021). https://doi.org/10.1016/j.dss.2021.113576.
[8] T. Hosaka, Bankruptcy prediction using imaged financial ratios and convolutional neural networks, Expert Syst. Appl. 117 (2019) 287–299, https://doi.org/10.1016/j.eswa.2018.09.039.
[9] M.A. Muslim, Y. Dasril, Company bankruptcy prediction framework based on the most influential features using XGBoost and stacking ensemble learning, IJECE 11 (2021) 5549, https://doi.org/10.11591/ijece.v11i6.pp5549-5557.
[10] H.A. Alaka, L.O. Oyedele, H.A. Owolabi, V. Kumar, S.O. Ajayi, O.O. Akinade, M. Bilal, Systematic review of bankruptcy prediction models: towards a framework for tool selection, Expert. Syst. Appl. 94 (2018) 164–184, https://doi.org/10.1016/j.eswa.2017.10.040.
[11] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electrical Eng. 40 (2014) 16–28, https://doi.org/10.1016/j.compeleceng.2013.11.024.
[12] W.-C. Lin, Y.-H. Lu, C.-F. Tsai, Feature selection in single and ensemble learning-based bankruptcy prediction models, Expert. Syst. 36 (2019) e12335, https://doi.org/10.1111/exsy.12335.
[13] D. Liang, C.-F. Tsai, H.-T. Wu, The effect of feature selection on financial distress prediction, Knowl. Based. Syst. 73 (2015) 289–297, https://doi.org/10.1016/j.knosys.2014.10.010.
[14] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, The MIT Press, Cambridge, 1992 [etc.].
[15] A. Brabazon, M. Kampouridis, M. O'Neill, Applications of genetic programming to finance and economics: past, present, future, Genet. Program. Evol. Mach. 21 (2020) 33–53, https://doi.org/10.1007/s10710-019-09359-z.
[16] A. Jović, K. Brkić, N. Bogunović, A review of feature selection methods with applications. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015, pp. 1200–1205, https://doi.org/10.1109/MIPRO.2015.7160458.
[17] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: a review. Data Classification: Algorithms and Applications, 2014, p. 37.
[18] B. Remeseiro, V. Bolon-Canedo, A review of feature selection methods in medical applications, Comput. Biol. Med. 112 (2019) 103375, https://doi.org/10.1016/j.compbiomed.2019.103375.
[19] F.R. Matenda, M. Sibanda, E. Chikodza, V. Gumbo, Bankruptcy prediction for private firms in developing economies: a scoping review and guidance for future research, Manag Rev Q (2021), https://doi.org/10.1007/s11301-021-00216-x.
[20] L. Papíková, M. Papík, Effects of classification, feature selection, and resampling methods on bankruptcy prediction of small and medium-sized enterprises, Intell. Syst. Account., Finance Manag. (2022), https://doi.org/10.1002/isaf.1521 n/a.
[21] R. Poli, W.B. (William B.) Langdon, N.F. McPhee, J.R. Koza, A field guide to genetic programming, [S.I.] : [Lulu Press], lulu.com, 2008. http://archive.org/details/AFieldGuideToGeneticProgramming (accessed March 8, 2020).
[22] A. Petrowski, S. Ben-Hamida, Evolutionary Algorithms, John Wiley & Sons, 2017.
[23] E. Alfaro-Cid, K. Sharman, A. Esparcia-Alcazar, A genetic programming approach for bankruptcy prediction using a highly unbalanced database, in: M. Giacobini (Ed.), Applications of Evolutionary Computing, Proceedings, 2007, pp. 169–178.
[24] M. Divsalar, H. Roodsaz, F. Vahdatinia, G. Norouzzadeh, A.H. Behrooz, A robust data-mining approach to bankruptcy prediction, J. Forecast. 31 (2012) 504–523, https://doi.org/10.1002/for.1232.
[25] H. Etemadi, A.A. Anvary Rostamy, H.F. Dehkordi, A genetic programming model for bankruptcy prediction: empirical evidence from Iran, Expert. Syst. Appl.. 36 (2009) 3199–3207. https://doi.org/10.1016/j.eswa.2008.01.012.
[26] A.L. Garcia-Almanza, B. Alexandrova-Kabadjova, S. Martinez-Jaramillo, Understanding Bank Failure: A Close Examination of Rules Created by Genetic Programming, ieee Computer Soc, Los Alamitos, 2010, https://doi.org/10.1109/CERMA.2010.14.
[27] T. Lensberg, A. Eilifsen, T.E. McKee, Bankruptcy theory development and classification via genetic programming, Eur. J. Operat. Res. 169 (2006) 677–697, https://doi.org/10.1016/j.ejor.2004.06.013.
[28] T.E. McKee, T. Lensberg, Genetic programming and rough sets: a hybrid approach to bankruptcy classification, Eur. J. Operat. Res. 138 (2002) 436–451, https://doi.org/10.1016/S0377-2217(01)00130-8.
[29] S. Salcedo-Sanz, J.L. Fernandez-Villacanas, M.J. Segovia-Vargas, C. Bousono-Calzon, Genetic programming for the prediction of insolvency in non-life insurance companies, Comput. Oper. Res. 32 (2005) 749–765, https://doi.org/10.1016/j.cor.2003.08.015.
[30] Á. Beade, M. Rodríguez, J. Santos, Evolutionary feature selection approaches for insolvency business prediction with genetic programming, Nat. Comput. (2023), https://doi.org/10.1007/s11047-023-09951-4.

[31] T. Dokeroglu, A. Deniz, H.E. Kiziloz, A comprehensive survey on recent metaheuristics for feature selection, Neurocomputing. 494 (2022) 269–296, https://doi.org/10.1016/j.neucom.2022.04.083.
[32] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (2016) 606–626, https://doi.org/10.1109/TEVC.2015.2504420.
[33] Y. Gong, J. Zhou, Q. Wu, M. Zhou, J. Wen, A length-adaptive non-dominated sorting genetic algorithm for bi-objective high-dimensional feature selection, IEEE/CAA J. Automatica Sinica 10 (2023) 1834–1844.
[34] J. Zhou, Q. Wu, M. Zhou, J. Wen, Y. Al-Turki, A. Abusorrah, LAGAM: a length-adaptive genetic algorithm with Markov blanket for high-dimensional feature selection in classification, IEEe Trans. Cybern. 53 (2023) 6858–6869, https://doi.org/10.1109/TCYB.2022.3163577.
[35] J. Ma, X. Gao, A filter-based feature construction and feature selection approach for classification using genetic programming, Knowledge-Based Syst. 196 (2020) 105806, https://doi.org/10.1016/j.knosys.2020.105806.
[36] D.P. Muni, N.R. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design, IEEE Trans. Syst. Man Cybern. Part B-Cybern. 36 (2006) 106–117, https://doi.org/10.1109/TSMCB.2005.854499.
[37] K. Neshatian, M. Zhang, Using genetic programming for context-sensitive feature scoring in classification problems, Conn. Sci. 23 (2011) 183–207, https://doi.org/10.1080/09540091.2011.630065.
[38] F. Viegas, L. Rocha, M. Gonçalves, F. Mourão, G. Sá, T. Salles, G. Andrade, I. Sandin, A genetic programming approach for feature selection in highly dimensional skewed data, Neurocomputing. 273 (2018) 554–569, https://doi.org/10.1016/j.neucom.2017.08.050.
[39] J. Yu, J. Yu, A.A. Almal, S.M. Dhanasekaran, D. Ghosh, W.P. Worzel, A. M. Chinnaiyan, Feature selection and molecular classification of cancer using genetic programming, Neoplasia 9 (2007) 292, https://doi.org/10.1593/neo.07121. -U16.
[40] E.I. Altman, G. Sabato, Modelling credit risk for SMEs: evidence from the U.S. market, Abacus 43 (2007) 332–357, https://doi.org/10.1111/j.1467-6281.2007.00234.x.
[41] E.I. Altman, M. Iwanicz-Drozdowska, E. Laitinen, A. Suvas, Financial and non-financial variables as long-horizon predictors of bankruptcy, SSRN Electron. J. (2015), https://doi.org/10.2139/ssrn.2669668.
[42] J.L. Bellovary, D.E. Giacomino, M.D. Akers, A review of bankruptcy prediction studies: 1930 to present, J. Financ. Educ. 33 (2007) 1–42.
[43] M.D. Beneish, The detection of earnings manipulation, Financ. Anal. J. 55 (1999) 24, https://doi.org/10.2469/faj.v55.n5.2296.
[44] P. du Jardin, Predicting bankruptcy using neural networks and other classification methods: the influence of variable selection techniques on model accuracy, Neurocomputing. 73 (2010) 2047–2060, https://doi.org/10.1016/j.neucom.2009.11.034.
[45] S. Tian, Y. Yu, Financial ratios and bankruptcy predictions: an international evidence, Int. Rev. Econ. Finance 51 (2017) 510–526, https://doi.org/10.1016/j.iref.2017.07.025.
[46] E. Yardeni, J. Abbot, M. Quintana, S&P 500 Financial ratios, Yardeni Research, Inc, 2019, p. 15.
[47] K. Palepu, Predicting takeover targets - A methodological and empirical-analysis, J. Account. Econ. 8 (1986) 3–35, https://doi.org/10.1016/0165-4101(86)90008-X.
[48] A. More, Survey of resampling techniques for improving classification performance in unbalanced datasets, arXiv:1608.06048 [Cs, Stat]. (2016). http://arxiv.org/abs/1608.06048 (accessed August 5, 2021).
[49] M.A. Hall, L.A. Smith, Practical Feature Subset Selection For Machine Learning, Springer, 1998. https://researchcommons.waikato.ac.nz/handle/10289/1512 (accessed May 23, 2022).
[50] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, California, AAAI Press, 1992, pp. 129–134.
[51] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: F. Bergadano, L. De Raedt (Eds.), Machine Learning: ECML-94, Springer, Berlin, Heidelberg, 1994, pp. 171–182, https://doi.org/10.1007/3-540-57868-4_57.
[52] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422, https://doi.org/10.1023/A:1012487302797.
[53] M. Hall, Correlation-based feature selection for machine learning, Department Comput. Sci. 19 (2000).
[54] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (2003) 155–176, https://doi.org/10.1016/S0004-3702(03)00079-1.
[55] H. Liu, R. Setiono, A probabilistic approach to feature selection - a filter solution, (1996). https://www.semanticscholar.org/paper/A-Probabilistic-Approach-to-Feature-Selection-A-Liu-Setiono/7285ee82aa0cde847fafb8b1109dd19dbdc04e35 (accessed May 23, 2022).
[56] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[57] G.H. John, P. Langley, Estimating continuous distributions in bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.
[58] J.R. Quinlan, C4.5: Programs for Machine Learning, Elsevier, 2014.
[59] W.W. Cohen, Fast effective rule induction, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 115–123.
[60] J.G. Cleary, L.E. Trigg, An instance-based learner using an entropic distance measure, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 108–114.
[61] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140, https://doi.org/10.1007/BF00058655.
[62] P. Melville, R.J. Mooney, Constructing Diverse Classifier Ensembles Using Artificial Training Examples, Ijcai, 2003, pp. 505–510.
[63] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[64] L.I. Kuncheva, Combining Pattern classifiers: Methods and Algorithms, 1st ed., Wiley, 2004 https://doi.org/10.1002/0471660264.
[65] Tin Kam Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Machine Intell. 20 (1998) 832–844, https://doi.org/10.1109/34.709601.
[66] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1996, pp. 148–156.
[67] F. Matloob, T.M. Ghazal, N. Taleb, S. Aftab, M. Ahmad, M.A. Khan, S. Abbas, T. R. Soomro, Software defect prediction using ensemble learning: a systematic literature review, IEEe Access. 9 (2021) 98754–98771, https://doi.org/10.1109/ACCESS.2021.3095559.
[68] M. Torabi, N.I. Udzir, M.T. Abdullah, R. Yaakob, A review on feature selection and ensemble techniques for intrusion detection system, Int. J. Adv. Comput. Sci. Appl. 12 (2021).
[69] V. Bolón-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: a review and future trends, Inf. Fusion 52 (2019) 1–12, https://doi.org/10.1016/j.inffus.2018.11.008.
[70] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, M.K. Rasel, A review of ensemble learning based feature selection, IETE Tech. Rev. 31 (2014) 190–198, https://doi.org/10.1080/02564602.2014.906859.
[71] E. Frank, M.A. Hall, I.H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed., Morgan Kaufmann, Burlington, MA, 2016.
[72] E. Frank, M.A. Hall, I.H. Witten, The Weka workbench, (2016). https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.
[73] S. Wagner, G. Kronberger, A. Beham, M. Kommenda, A. Scheibenpflug, E. Pitzer, S. Vonolfen, M. Kofler, S. Winkler, V. Dorfer, M. Affenzeller, Architecture and design of the HeuristicLab optimization environment, in: R. Klempous, J. Nikodem, W. Jacak, Z. Chaczko (Eds.), Advanced Methods and Applications in Computational Intelligence, Springer, 2014, pp. 197–261. http://link.springer.com/chapter/10.1007/978-3-319-01436-4_10.
[74] F. Zambrano Farias, M.del C. Valls Martínez, P.A. Martín-Cervantes, Explanatory factors of business failure: literature review and global trends, Sustainability. 13 (2021) 10154, https://doi.org/10.3390/su131810154.
[75] T.G. Dietterich, Ensemble methods in machine learning. Multiple Classifier Systems, Springer, 2000, pp. 1–15, https://doi.org/10.1007/3-540-45014-9_1.