Predicting School Failure and Dropout by Using Data Mining Techniques

Article *in* Revista Iberoamericana de Tecnologias del Aprendizaje · February 2013

DOI: 10.1109/RITA.2013.2244695

3 authors:

Carlos Márquez

4 PUBLICATIONS **543** CITATIONS

SEE PROFILE

Cristóbal Romero

University of Cordoba (Spain)

130 PUBLICATIONS **11,236** CITATIONS

SEE PROFILE

Sebastian Ventura

University of Cordoba (Spain)

356 PUBLICATIONS **14,893** CITATIONS

SEE PROFILE

Appl Intell
DOI 10.1007/s10489-012-0374-8

Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data

Carlos Márquez-Vera · Alberto Cano · Cristóbal Romero · Sebastián Ventura

Abstract Predicting student failure at school has become a difficult challenge due to both the high number of factors that can affect the low performance of students and the imbalanced nature of these types of datasets. In this paper, a genetic programming algorithm and different data mining approaches are proposed for solving these problems using real data about 670 high school students from Zacatecas, Mexico. Firstly, we select the best attributes in order to resolve the problem of high dimensionality. Then, rebalancing of data and cost sensitive classification have been applied in order to resolve the problem of classifying imbalanced data. We also propose to use a genetic programming model versus different white box techniques in order to obtain both more comprehensible and accuracy classification rules. The outcomes of each approach are shown and compared in order to select the best to improve classification accuracy, specifically with regard to which students might fail.

Keywords Predicting student performance · Classification · Educational data mining · Student failure · Grammar-based genetic programming

C. Márquez-Vera
Autonomous University of Zacatecas, Zacatecas, México
e-mail: cmarquezvera@hotmail.com

A. Cano · C. Romero (✉) · S. Ventura
Department of Computer Science, University of Córdoba, Córdoba, Spain
e-mail: cromero@uco.es

A. Cano
e-mail: acano@uco.es

S. Ventura
e-mail: sventura@uco.es

Published online: 26 August 2012

1 Introduction

Detecting student failure at school is a major social problem and it has become very important for educational professionals to better understand why so many youths fail to complete their school studies. But, this is a difficult problem to resolve due to the large amount of risk factors or characteristics of the students that can influence school failure, such as demographics, cultural, social, family, or educational background, socioeconomic status, psychological profile, and academic progress [1]. In fact, this problem is also known as the "one thousand factors problem" [15]. In the past decades, a great deal of research has been done on identifying the main factors that affect the low performance of students such as failure and dropping out at different educational levels, such as primary, secondary, and higher [2]. One of the most influential theoretical explanations of this problem and its causes and cures is the path analysis model of Tinto [39]. This model suggest that the student's social and academic integration into the educational institution is the major determinant of completion and identifies some key influences on integration such as the student's family background, personal characteristics, previous schooling, prior academic performance, and interactions between student and the faculty. More recently, there has been a consensus that detection and prevention of student failure at school and early intervention make much more sense than remediation [33]. In this line, an effective way to detect student failure at school is the use of Data Mining (DM) techniques that have been successfully applied to other research areas such as ecommerce, where their use is now very popular. DM is part of the process of Knowledge Discovery and Data Mining (KDD) and is understood to be the non-trivial extraction of previously unknown and potentially useful, valid, and comprehensible information from a large volume of data [17]. DM is called

Educational Data Mining (EDM) when applied to an educational context [35] and is concerned with developing methods to explore unique types of data in educational settings and using these methods to better understand students and the settings in which they learn. In fact, there are good examples of how to apply EDM techniques to create models that predict dropping out and student failure specifically [20]. These works have shown promising results with respect to those sociological, economic, or educational characteristics that may be more relevant in the prediction of low academic performance. It is also important to notice that most of the research on the application of EDM to resolve the problems of student failure and drop-outs has been applied primarily to the specific case of higher education [19] and more specifically to online or distance education [22]. However, very little information about specific research on elementary and secondary education has been found, and what has been found uses only statistical methods, not DM techniques [29].

This study proposes to predict student failure at school in secondary education by using DM. In fact, we want to detect the factors that most influence student failure in young students by using classification techniques. Classification is one of the most studied tasks by DM and Machine Learning (ML) researchers and it consists of predicting the value of a (categorical) attribute (the class) based on the values of other attributes (the predicting attributes). In ML and DM fields, classification is usually approached as a supervised learning task. A search algorithm is used to induce a classifier from a set of correctly classified data instances, called the training set. Another set of correctly classified data instances known as the test set is used to measure the quality of the classifier obtained after the learning process. On the other hand, classification algorithms can be grouped in black and white box models. White box models can be used directly for decision making and provide an explanation for the classification which can be reviewed and agreed by an expert. In this paper, white box algorithms are used because the models obtained have the form of IF-THEN classification or prediction rules that show the reasons for classification and important dependence relations between data. This kind of rule consists of two parts. The rule antecedent (the IF part) contains a conjunction of m conditions on values of predictor attributes, whereas the rule consequent (the THEN part) contains a prediction about the value of a goal attribute or class. In our case, our final objective is to be able to identify the reasons that lead to school failure and to provide identification of students who show those characteristics (i.e. the factors that have the greatest influence on school failure) in order to offer them appropriate and personalized help in an attempt to reduce failure in school. In fact, in this study a new algorithm is proposed for predicting school failure as a classification task. Several experiments have been performed to compare our proposed algorithm with other classification algorithms. Different DM approaches have also been used to try to increase the accuracy of the classification model and to resolve the problems of high dimensionality and imbalanced data.

The paper is organized as follows: Section 2 shows some related works and background about detecting student failure using data mining. Section 3 presents our proposed method for predicting school failure. Section 4 describes the data used. Section 5 describes the data preprocessing step. Section 6 presents the classification models and the evolutionary algorithm proposed in this work. Section 7 shows the experiments carried out and the results obtained. Section 8 presents the rules discovered by the algorithms. In Sect. 9 there is a discussion of this work, and finally Sect. 10 summarizes the main conclusions and future research.

2 Background

The problem of predicting low performance, dropping out, and school failure among students is one of the oldest and most popular applications of EDM. There are a great number of works on how to apply EDM techniques to create models that predict dropping out and school failure [35]. For example, both statistical techniques such as correlation analysis and regression and DM techniques such as classification using decision trees, Bayesian networks, neural networks, K-nearest neighbor classifiers, and so on have been used, evaluated, and compared to predict students' academic success [14]. Next, some representative examples of works that use these techniques are described.

On the one hand, statistical techniques such as correlation analyses have been applied, for example, to predict web-student performance in online classes [42], to identify predictors of academic persistence in distance education [29], and to predict middle school students' probabilities of success at university [25]. Discriminant function analysis has been used to predict student academic success (attributes that are successful or not) [24]. Logistic regression models have been applied to determine the risk of a university student abandoning his or her degree [2] and to investigate whether leisure boredom predicts dropping out of middle school [43]. On the other hand, decision tree analysis, which is one of the most popular DM techniques, has been used for predicting the dropping-out feature of college students [30] and for identifying characteristics of students who drop out of high school [41]. More advanced DM algorithms such as neural networks and random forests together with decision trees have been applied to predict students' academic success, classifying them into categories of low-risk, medium-risk, and high-risk of failing [38]. Classification trees and random forests have been applied to identify factors associated with persistence in science and engineering majors [26]. Decision trees, neural networks, naive Bayes algorithms, and instance-based learning algorithms have been

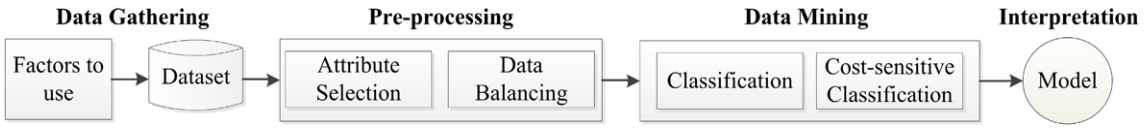

Fig. 1 Method proposed for the prediction of student failure

used to predict dropping-out among university students [21]. However, it is also important to notice that most of this research on EDM has been applied primarily to the specific case of higher education [19] and more specifically to online or distance education [22]. However, very little information has been found in specific research on elementary and secondary education, and what has been found uses only statistical methods [29, 43] or decision tree models [30, 41]. It is important to note that the factors that can affect low student performance may vary greatly depending on the student's educational level. In other words, certain factors which are key in primary or secondary education might not be so important in higher education and vice versa. It is therefore firstly necessary to fully and widely research all the possible factors. Having gathered the data, a feature selection process must then be used in order to identify and select the most important factors that have the greatest effect on student performance [23].

Finally, in most of the cases the data used to classify students at risk of failing or dropping out are imbalanced, which means that only a small proportion of students fail or leave and the vast majority pass or continue with their studies. Some methods proposed by the DM and ML communities to solve this problem have also been used in education, such as using cost-sensitive classification and resampling of the original datasets. For example, some case studies show that the cost-sensitive technique is a more effective solution to predict dropping-out of new university students [19], to improve the effectiveness of several classification techniques using university students datasets [7], and to predict the marks that university students will obtain in the final exam of a course [34]. Only our previous initial work [23] propose to use both cost-sensitive classification and rebalanced data for predicting school students' performance. However, in this paper we have extend our previous work and we have also proposed the use of a new evolutionary (grammar-based) algorithm to discover more accurate and comprehensible models for predicting student failure at school and we compare it versus traditional classification algorithms. Our work does in fact have two main advantages:

– Firstly, we propose a general method for predicting academic failure using data mining techniques. Predicting student performance is difficult since there are normally a large number of possible factors (high dimensional data) and most students pass the year (imbalanced data). We therefore propose that several data mining techniques be used in our methodology and as a case study we will show an example of applying our method over real data from high school students.
– Secondly, we propose a specific genetic programming algorithm for obtaining a shorter, more accurate classification model. Nowadays, there are many classification algorithms, and users can be at a loss when selecting the most appropriate to use with their data to predict a student's performance. We therefore propose a new algorithm and we compare it to other traditional classification algorithms using both performance (accuracy and geometric mean) and comprehensible (number and size of rules) evaluation measures.

3 Method

The method proposed in this paper for predicting the academic failure of students belongs to the process of Knowledge Discovery and Data Mining (see Fig. 1).

The main stages of the method are:

– *Data gathering*. This stage consists in gathering all available information on students. To do this, the set of factors that can affect the students' performance must be identified and collected from the different sources of data available. Finally, all the information should be integrated into a dataset.
– *Pre-processing*. At this stage the dataset is prepared to apply the data mining techniques. To do this, traditional preprocessing methods such as data cleaning, transformation of variables, and data partitioning have to be applied. Other techniques such as the selection of attributes and the re-balancing of data have also been applied in order to solve the problems of high dimensionality and imbalanced data that are typically presented in these datasets.
– *Data mining*. At this stage, DM algorithms are applied to predict student failure like a classification problem. To do this, a new programming classification algorithm based on genetic programming is proposed and compared with other classical classification algorithms based on classification rules and decision trees. In addition, a cost sensitive classification approach is also used in order to solve the imbalanced data problem.
– *Interpretation*. In this last stage, the obtained models are analyzed to detect student failure. To achieve this, the factors that appear (in the rules and decision trees) and how they are related are considered and interpreted.

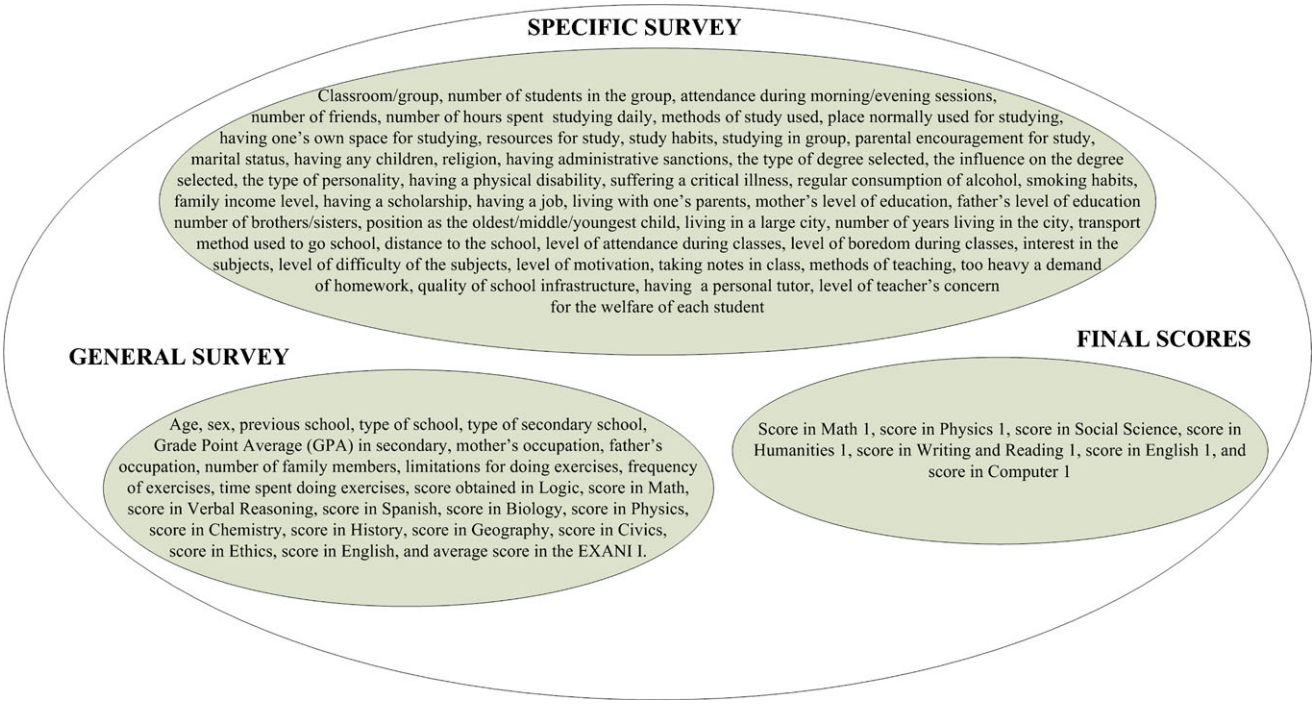

Fig. 2 Data sources and students' information

Table 1 Data sources and students' attributes used

Source	General survey	Specific survey	Scores
Type of information	Socioeconomic factors and previous marks	Personal, social, family and school factors	Current marks
Number of attributes	25	45	7

4 Data

In this paper, we have used information about high school students enrolled on Program II of the Academic Unit Preparation at the Autonomous University of Zacatecas (UAPAZ) for the 2009/10 academic year. In the Mexican educational system, high school corresponds to the preparatory years when the students are aged 15–18 years old. High school offers a three-year education program that provides the student with general knowledge to continue studying at university. We have only used information about first-year high school students, where most students are between the ages of 15 and 16, as this is the year with the highest failure rate. Students only proceed to the next year if they pass all the year subjects or if they only fail one. Each year consists of a group of subjects that are graded using a decimal system ranging from 0 to 10 in order to grade the students. In order to pass a subject, a student must score higher than 5.9.

All the information used in this study has been gathered from three different sources during the period from August to December 2010 (see Fig. 2). Firstly, a general survey is completed by students when they register in the National Evaluation Center (CENEVAL) for admission to many institutions of secondary and higher education. Then, a specific survey was designed and administered to all students in the middle of the course. Its purpose was to obtain personal and family information to identify some important factors that could affect school performance. Finally, the Department of School Services provides the scores obtained by students in all subjects of the course.

So, our dataset has 77 attributes, variables, or features for each student (see Table 1). This is a high dimensional dataset that put demands on the learning algorithm in terms of both efficiency and effectiveness [37]. The curse of dimensionality is a well-known phenomenon that occurs when the generation of a predictive model is misled by an overwhelming number of features to choose between, for example when deciding what feature to use in a node of a decision tree. And although some learning methods are less sensitive to this problem their computational cost may be prohibitively large. We will try to resolve this problem during our experiments.

Finally, the output variable/attribute or class to be predicted in our problem is the academic status or final student performance, which has two possible values: Pass (students who pass the course) or Fail (students who have to repeat the course). This attribute was provided by the Department of School Services of Academic Program II of UAPUAZ at the end of the course.

5 Pre-processing tasks

5.1 Integration, cleaning, and transformation

It must be pointed out that a very important task in this work was data pre-processing, due to the quality and reliability of available information, which directly affects the results obtained. In fact, some specific pre-processing tasks were applied to prepare all the previously described data so that the classification task could be carried out correctly. Firstly, all available data were integrated into a single dataset. During this process those students without 100 % complete information were eliminated. All students who did not answer our specific survey or the CENEVAL survey were excluded. Some modifications were also made to the values of some attributes. For example, words that contained the letter "Ñ" were replaced by "N". A new attribute of the age of each student in years was created using the day, month, and year of birth of each student. Furthermore, the continuous variables were transformed into discrete variables, which provide a much more comprehensible view of the data. For example, the numerical values of the scores obtained by students in each subject were changed to categorical values in the following way:

- Excellent: score between 9.5 and 10
- Very good: score between 8.5 and 9.4
- Good: score between 7.5 and 8.4
- Regular: score between 6.5 and 7.4
- Sufficient: score between 6.0 and 6.4
- Poor: between 4.0 and 5.9
- Very poor: less than 4.0
- Not presented

Then, all the information was integrated in a single dataset and it was saved in the .ARFF format of Weka [45]. Next, the whole dataset was divided randomly into 10 pairs of training and test data files (maintaining the original class distribution). This way, each classification algorithm can be evaluated using stratified tenfold cross-validation [18]. So, after preprocessing we have a dataset (split into 10 folds) with 77 attributes/variables of 670 students. However, our dataset has two typical problems that normally appear in these types of educational data. On the one hand, our dataset has high dimensionality; that is, the number of attributes or features becomes very large. Further, given a large number of attributes, some will usually not be meaningful for classification and it is likely that some attributes are correlated. On the other hand, the data are imbalanced (see Fig. 3); that is, the majority of students (610) passed and a minority (60) failed. The problem with imbalanced data [12] arises because learning algorithms tend to overlook less frequent classes (minority classes) and only pay attention to the most frequent ones (majority classes). As a result, the classifier obtained will not be able to correctly classify data instances corresponding to poorly represented classes.

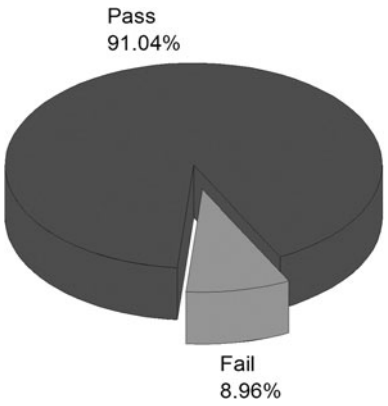

Fig. 3 Distribution of the final student performances/academic statuses

5.2 Selection of the best attributes

We decided to carry out a study of feature selection to try to identify which feature has the greatest effect on our output variable (academic status). The objective is to resolve the problem of high dimensional data by reducing the number of attributes without losing reliability in classification, using attribute selection algorithms that try to remove irrelevant attributes from data. In many practical situations there are far too many attributes for learning schemes to handle, and some of them can be irrelevant or redundant. There are a wide range of attribute selection algorithms that can be grouped in different ways [13]. Weka provides several feature selection algorithms from which we have selected the following ten [45]: CfsSubsetEval, ChiSquaredAttributeEval, ConsistencySubsetEval, FilteredAttributeEval, OneRAttributeEval, FilteredSubsetEval, GainRatioAttributeEval, InfoGainAttributeEval, ReliefFAttributeEval, SymmetricalUncertAttributeEval. The results obtained were ranked by these 10 algorithms to select the best attributes from our 77 available attributes (see Table 2). To find the ranking of the attributes, we counted the number of times each attribute was selected by one of the algorithms. Table 2 shows the frequency of each attribute. From this table only those with a frequency greater than two have been selected as the best attributes, that is, attributes that have been considered by more than two feature selection algorithms.

Table 2 Most influential attributes ranked by frequency of appearance

Attribute	Frequency
Scores in Humanities 1, and in English 1	10
Scores in Social Science 1, Math 1, Reading and Writing 1, Physics 1, and Computing 1	9
Level of motivation	5
Grade point average in secondary school	3
Age, number of brothers/sisters, classroom/group, smoking habits, and average score EXANI I	2
Studying in group, marital status, time spent doing exercises, and score in History	1

Finally, we selected only the attributes with a frequency greater than or equal to two (attributes selected by at least two algorithms). In this way, we can reduce the dimensionality of our dataset from the original 77 attributes to only the best 15 attributes.

5.3 Data rebalancing

The problem of imbalanced data classification occurs when the number of instances in one class is much smaller than the number of instances in another class or other classes [12]. Traditional classification algorithms have been developed to maximize the overall accuracy rate, which is independent of class distribution; this means that the majority of class classifiers are in the training stage, which leads to low sensitivity classification of minority class elements at the test stage. One way to solve this problem is to act during the pre-processing of data by carrying out a sampling or balancing of class distribution. There are several data balancing or rebalancing algorithms; one that is widely used and that is available in Weka as a supervised data filter is SMOTE (Synthetic Minority Over-sampling Technique) [40]. In the SMOTE algorithm, the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any or all of the k minority class nearest neighbors. Depending on the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. In our case, only the training files (with the best 15 attributes) have been rebalanced using the SMOTE algorithm, obtaining 50 % Pass students and 50 % Failed students and not rebalancing the test files.

Finally, after performing all the previous tasks of pre-processing, we obtained the next tenfold cross validation files:

- Ten training and test files with all attributes (77).
- Ten training and test files with only the best attributes (15).
- Ten training and test files with only the best attributes (15); the training files are rebalanced using SMOTE.

6 Classification models and genetic programming

6.1 Classification models

In DM and ML, a wide range of paradigms have been used to tackle classification: decision trees, inductive learning, instance-based learning, and, more recently, artificial neural networks and evolutionary algorithms. In this paper, decision tree, rule induction, and evolutionary models are used as they are white box classification techniques; that is, they provide an explanation for the classification result and can be used directly for decision making. A decision tree is a set of conditions organized in a hierarchical structure. An instance is classified by following the path of satisfied conditions from the root of the tree until a leaf is reached, which will correspond with a class label. Rule induction algorithms usually employ a specific-to-general approach, in which obtained rules are generalized (or specialized) until a satisfactory description of each class is obtained.

In this paper, 10 commonly used classical classification algorithms that are available in the well-known Weka DM software [45] have been used:

- Five rule induction algorithms: JRip [6], which is a propositional rule learner; NNge [36], which is a nearest-neighbor-like algorithm; OneR [16], which uses the minimum-error attribute for class prediction; Prism [5], which is an algorithm for inducing modular rules; and Ridor [32], which is an implementation of the RIpple-DOwn Rule learner.
- Five decision tree algorithms: J48 [31], which is an algorithm for generating a pruned or unpruned C4.5 decision tree; SimpleCart [3], which implements minimal cost-complexity pruning; ADTree [11], which is an alternating decision tree; RandomTree [45], which considers K randomly chosen attributes at each node of the tree; and REPTree [45], which is a Fast decision tree learner.

A decision tree can be directly transformed into a set of IF-THEN rules (which are obtained by rule induction algorithms), which are one of the most popular forms of knowledge representation due to their simplicity and comprehensibility. In this way a non-expert user of DM such as a teacher

Table 3 Confusion matrix

Pred.\Act.	Positive	Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

or instructor can directly use the output obtained by these algorithms to detect students with problems (classified as Fail) and to make decisions about how to help them and prevent their possible failure.

In order to evaluate the performance of classification algorithms, normally the confusion matrix is used. This matrix contains information about actual and predicted classifications (see Table 3).

Starting from the confusion matrix a wide range of classification performance measures can be obtained. In our case, the next four measures are used:

- Accuracy (Acc) is the overall accuracy rate or classification accuracy and is calculated as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- True Positive rate (TP rate), also called sensitivity (Se) or recall, is the proportion of actual positives which are predicted to be positive and is calculated as

$$TP_{rate} = \frac{TP}{TP + FN} \tag{2}$$

- True Negative rate (TN rate), or specificity (Sp), is the proportion of actual negatives which are predicted to be negative and is calculated as

$$TN_{rate} = \frac{TN}{TN + FP} \tag{3}$$

- Geometric Mean (GM) indicates the balance between classification performances in the majority and minority classes; that is, GM is a measure of the central tendency used with imbalanced datasets and is calculated as

$$GM = \sqrt{TP_{rate} \cdot TN_{rate}} \tag{4}$$

Finally, we have also developed a specific evolutionary algorithm called Interpretable Classification Rule Mining (ICRM), which is described in the following section.

6.2 Genetic programming for classification

Evolutionary algorithms are a paradigm based on the Darwin evolution process, where each individual codifies a solution and evolves into a better individual by means of genetic operators (mutation and crossover). Genetic Programming (GP) is an evolutionary algorithm based methodology

$\langle S \rangle \rightarrow \langle cmp \rangle \mid \langle cmp \rangle\ AND\ \langle S \rangle$
$\langle cmp \rangle \rightarrow \langle op\_cat \rangle \langle variable \rangle \langle value \rangle$
$\langle op\_cat \rangle \rightarrow = \mid \neq$
$\langle variable \rangle \rightarrow$ *Any valid attribute in dataset*
$\langle value \rangle \rightarrow$ *Any valid value*

Fig. 5 Grammar used to define the rules generated by ICRM

used to find computer programs that perform a user-defined task. It is a specialization of genetic algorithms where each individual is a computer program. Therefore, it is a machine learning technique used to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task. Genetic programming has been applied with success in various complex optimization, search and classification problems [8, 28].

The evolutionary algorithm proposed in our work is a variant of genetic programming known as grammar-based genetic programming (G3P) [44] in which a grammar is defined and the evolutionary process proceeds, guaranteeing that every individual generated is legal with respect to the grammar. In our case, we have developed a specific algorithm called Interpretable Classification Rule Mining (ICRM) that employs G3P to evolve rule-based classifiers and has been shown to perform well in other classification domains [4]. The algorithm presents three variants that allow us to conduct the process of searching for the kind of rules in which we are most interested, for example those that predict the reasons for student failure, which is a capability that, to the best of our knowledge, has no other existing algorithm in the literature.

Figure 4 shows the algorithm workflow, in which the algorithm iterates to find the best rules for the different classes using an individual = rule representation. This representation provides greater efficiency and addresses the cooperation–competition problem by dealing with the order of the rules within the rule base in the evolutionary process. We use a context-free grammar (see Fig. 5) to specify which relational operators are allowed to appear in the antecedents of the rules and which attribute must appear in the consequents (the class). The use of a grammar provides the expressiveness, flexibility, and ability to restrict the search space in the search for rules. The implementation of constraints using a grammar can be a very natural way to express the syntax of rules when individual representation is specified. The relational operators for nominal attributes are equal (=) and not equal ($\neq$). Therefore, the rules learned from this data comprise a combination of conditions on nominal attributes. These rules can be applied to a great number of learning problems.

The rules are constructed to find the conjunction of conditions on the relevant attributes that best discriminates a

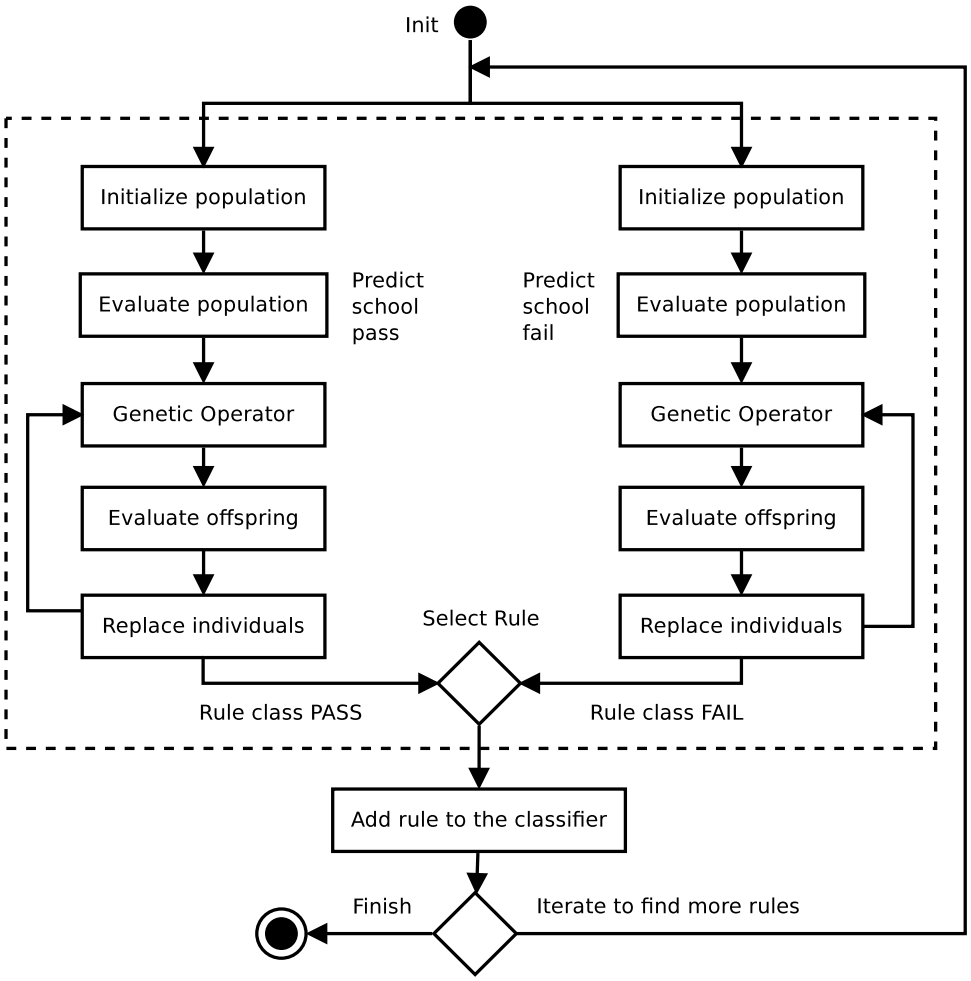

Fig. 4 ICRM algorithm workflow

class from the other classes. To do so, there is a genetic operator that employs the not-yet covered attributes of the rule to find the best condition on any other attribute that, when appended to the rule, improves its accuracy. The iterative process continues until the genetic operator does not find a new relevant attribute condition that improves the accuracy of the rule or until all the attributes have been covered.

The fitness function evaluates the quality of the represented solutions. We have used a combination of two measures that are commonplace in classification, namely the sensitivity (Se) and the specificity (Sp). These measures are also calculated by means of confusion matrix values (see Table 3). Then, the fitness value is calculated as the product of the sensitivity and the specificity to maximize the accuracy. This function performs well regardless of whether or not the data are imbalanced.

$$Fitness = Se \cdot Sp \tag{5}$$

Finally, there are several ways to build the rule base. In this paper we propose three different versions of our ICRM algorithm to obtain rule bases that are both accurate and useful to the user in the search for student failure information.

- The first approach (ICRM v1) states that there is one rule per class. The algorithm is responsible for learning accurate classification rules for both classes, but it finally decides which rule is set as the primary rule and therefore the other class is predicted by default. The most accurate rule is placed first.
- The second approach (ICRM v2) allows multiple rules for each class. The algorithm runs in a similar way to the first approach, but when the rules for both classes are learned, it appends the best rule to the classifier and removes the instances covered by the rule from the dataset. In the next iteration, the algorithm runs again but learns from the remaining instances. This process continues until all the training instances have been covered. The output classifier is comprised of multiple rules with different consequents.
- The third approach (ICRM v3) extends the second approach but focuses on the student failure problem. We are interested in obtaining accurate classification rules, but specifically it is even more interesting to find the relevant factors that lead a student to fail. Therefore, the rule base allows multiple rules, but they are fixed to predict the fail-

ure cases in which we are interested. Finally, if none of the failure rules cover an instance, the default class is set to predict the students who will pass the course. The number of rules required to predict failure cases is decided by the algorithm because of its ability to perform accurate classifications.

7 Experiments

We carried out several experiments to test and compare the three versions of our GGP algorithm with the 10 classical classification algorithms and to try to improve the accuracy of classification using different DM approaches for predicting student failure at school with high-dimensional and imbalanced data.

In the first experiment, all the classification algorithms were executed using tenfold cross-validation and all the available information, that is, the original data file with 77 attributes of 670 students. The results with the test files (an average of 10 executions) of classification algorithms are shown in Table 4. This table shows the rates or percentages of correct classifications for each of the two classes: Pass (TP rate) and Fail (TN rate), the overall Accuracy rate (Acc), and the Geometric Mean (GM). It also shows the number of rules (#Rules), the average number of conditions per rule (#Conditions per rule), and the average number of conditions of the classifier (#Conditions). The lower the number of rules and conditions, the higher the simplicity of the classifier and therefore the higher the comprehensibility of the knowledge discovered. It can be seen in Table 4 that the values generally obtained have high accuracy (greater than 85.2 %) and a high TP rate (greater than 84.4 %) but a lower TN rate (greater than 25.0 %) and lower GM (greater than 49.9 %). The best algorithm in terms of TP rate and Accuracy was ADTree (99.7 % and 97.6 % respectively), which is a high rate for predicting students who will pass. However, we are interested in the TN rate, students who failed, for which ICRM v3 obtains the best results and ICRM v1 obtains the best GM (93.3 % and 91.9 % respectively). OneR always obtains the simplest classifier (one rule to predict a class and another default class), but its prediction needs to be as general as possible to obtain the highest accuracy and therefore its TN rate is low (41.7 %).

However, the ICRM models obtain both a high TN rate and a low number of rules and conditions. Therefore, they are accurate but comprehensible classification models for predicting student failure with an appropriate trade-off regarding accuracy versus interpretability.

From this first experiment using all available attributes, we have seen that all the obtained models use only a few of the large number of attributes available (77).

In the second experiment, we executed all the classification algorithms using tenfold cross-validation and the reduced dataset (with only the best 15 attributes). Table 5 shows the results with the test files (the average of 10 executions) using only the best 15 attributes. When comparing the results obtained with the previous results obtained using all the attributes, that is, Table 4 versus Table 5, we can see that in general all the algorithms have improved in some measures (TN rate and GM). Furthermore, with regard to the other measures (TP rate and Accuracy) there are some algorithms that obtain a slightly worse or slightly better value, but they are very similar in general to the previous ones. In fact, the maximum values obtained are now better than the previous ones obtained using all attributes in two evaluation measures (TN rate and GM). The algorithm that obtains these maximum values is ICRM v1 (93.3 % TN rate and 92.5 % GM). It also obtains the minimum number of rules (2), along with OneR, but a much higher TN rate and a low number of conditions.

As we can see from Tables 4 and 5, TP rate values are normally higher than TN rate values. This is because our data are imbalanced.

In the third experiment, we again executed all the classification algorithms using tenfold cross-validation and the rebalanced training files (with only the best 15 attributes). The results obtained after re-executing the 10 classification algorithms using tenfold cross-validation are summarized in Table 6. If we analyse and compare this table with the previous one (Table 5), we can observe that just over half of the algorithms have increased the values obtained in all the evaluation measures, and some of them also obtain the new best maximum values in almost all measures except accuracy: Prism (99.8 % TP rate), ICRM v3 (98.7 % TN rate), ADTree (97.2 % Accuracy), and ICRM v2 (97.0 % GM). The best overall results are those obtained by ICRM models, since Prism and ADTree have a lower TN rate and GM than ICRM models. ICRM v3 obtains the highest TN rate ever and ICRM v2 the highest GM. This means that ICRM v3 is capable of predicting 98.7 % of student failures using only five rules and an average of 1.5 conditions per rule. ICRM v2 reduces the number of rules to two and achieves a TN rate of 98.0 %.

A different approach to solving the problem of imbalanced data classification is to apply cost-sensitive classification [9]. Optimizing the classification rate without taking into consideration the cost of errors can often lead to suboptimal results because high costs can result from the misclassification of a minority instance. In fact, in our particular problem, we are much more interested in the classification of Fail students (the minority class) than Pass students (the majority class). These costs can be incorporated into the algorithm and considered during classification. In the case of two classes, costs can be put into a 2×2 matrix in which

Table 4 Classification results using all attributes

Algorithm	TP rate	TN rate	Acc	GM	#Rules	#Conditions per rule	#Conditions
JRip	97.7	78.3	96.0	87.5	8.0	1.5	12.0
NNge	98.5	73.3	96.3	85.0	31.0	76.0	2356.0
OneR	98.9	41.7	93.7	64.2	2.0	0.5	1.0
Prism	99.5	25.0	93.1	49.9	76.0	1.4	110.0
Ridor	96.6	65.0	93.7	79.2	4.0	1.7	7.0
ADTree	99.7	76.7	97.6	87.4	21.0	1.7	36.0
J48	97.4	53.3	93.4	72.1	31.0	3.1	98.0
RandomTree	95.7	48.3	91.5	68	212.0	4.9	1041.0
REPTree	98.0	56.7	94.3	74.5	44.0	1.8	83.0
SimpleCart	97.7	65.0	94.8	79.7	5.0	12.8	64.0
ICRM v1	94.3	90.0	93.9	91.9	2.0	1.5	3.1
ICRM v2	97.5	75.0	95.5	85.0	7.6	1.9	14.7
ICRM v3	84.4	93.3	85.2	88.5	4.0	1.1	4.5

Table 5 Classification results using the best attributes

Algorithm	TP rate	TN rate	Acc	GM	#Rules	#Conditions per rule	#Conditions
JRip	97.0	81.7	95.7	89.0	5.7	1.5	8.7
NNge	98.0	76.7	96.1	86.7	22.2	14.0	310.8
OneR	98.9	41.7	93.7	64.2	2.0	0.8	1.6
Prism	99.2	44.2	94.7	66.2	55.6	1.7	93.8
Ridor	95.6	68.3	93.1	80.8	4.0	1.2	5.4
ADTree	99.2	78.3	97.3	88.1	21.0	3.0	63.0
J48	97.7	55.5	93.9	73.6	19.9	2.1	43.0
RandomTree	98.0	63.3	94.9	78.8	278.6	3.3	912.2
REPTree	97.9	60.0	94.5	76.6	30.0	1.9	68.4
SimpleCart	98.0	65.0	95.1	79.8	6.9	4.1	29.4
ICRM v1	92.0	93.3	92.1	92.5	2.0	2.4	4.9
ICRM v2	97.2	71.7	94.9	82.8	8.2	2.1	17.9
ICRM v3	75.9	85.0	76.7	79.0	4.0	0.9	3.8

diagonal elements represent the two types of correct classifications and the off-diagonal elements represent the two types of errors.

Table 7 shows the default cost matrix for two class cases whose values simply give the number of errors: misclassification costs are all 1. Weka allows any classification algorithm to be made cost sensitive by using the meta-classification algorithm CostSensitiveClassifier and setting its base classifier as the desired algorithm. In our problem, we have used the values of the matrix shown in Table 8 as the cost matrix, due to previous tests with different values of costs in which this matrix obtained the best results. This matrix indicates that performing the classification takes into consideration that it is four times more important to correctly classify Fail students than Pass students.

In the fourth experiment, we executed all the classification algorithms using tenfold cross-validation, considering different costs of classification and introducing a cost matrix (with only the best 15 attributes). Table 9 shows the results with test files obtained after applying tenfold cross-validation. On analyzing and comparing Table 9 with Table 6, some algorithms can be seen to obtain better values in some evaluation measures while other algorithms obtain worse values. So, there is no clear general improvement. One algorithm (JRip) does obtain the current best maximum values of the TN rate (93.3 %) and GM (94.6 %), which is very important in our problem. However, the best overall results were obtained for our ICRM proposal using data balancing, which achieved a TN rate of up to 98.7 % and a GM rate of up to 97.0 %; these are the best results from all the experiments. It is therefore the classification model

Table 6 Classification results using data balancing

Algorithm	TP rate	TN rate	Acc	GM	#Rules	#Conditions per rule	#Conditions
JRip	97.7	65.0	94.8	78.8	12.3	1.6	18.7
NNge	98.7	78.3	96.9	87.1	24.2	14.0	338.8
OneR	88.8	88.3	88.8	88.3	2.0	1.0	2.0
Prism	99.8	37.1	94.7	59.0	67.4	1.9	127.7
Ridor	97.9	70.0	95.4	81.4	9.3	1.9	17.7
ADTree	98.2	86.7	97.2	92.1	21.0	2.8	58.0
J48	96.7	75.0	94.8	84.8	45.4	2.8	137.3
RandomTree	96.1	68.3	93.6	79.6	284.1	3.6	1033.4
REPTree	96.5	75.0	94.6	84.6	66.9	2.2	153.9
SimpleCart	96.4	76.7	94.6	85.5	12.3	5.9	78.0
ICRM v1	93.8	98.0	95.9	95.9	2.0	1.5	3.1
ICRM v2	98.0	96.1	97.1	97.0	7.9	1.8	14.4
ICRM v3	86.7	98.7	92.7	92.5	5.0	1.5	7.6

Table 7 Default cost matrix

Pred.\Act.	Positive	Negative
Positive	0	1
Negative	1	0

Table 8 Used cost matrix

Pred.\Act.	Positive	Negative
Positive	0	1
Negative	4	0

which provides the most accurate and interesting results for predicting the TN cases (student failure).

Finally, we have used the average ranking of the previous DM approaches and measures (see Table 10) to compare them (Tables 4, 5, 6, and 9) and to obtain the best overall results. This average ranking has been calculated starting with four single ranking values between 1 and 4 that show the ranking of each algorithm and measure for each of the four approaches. So, the lower the average ranking, the better the results that are achieved by the algorithms using that approach.

As we can see from Table 10, the best TP rate ranking is obtained when using all the attributes, but the best TN rate, accuracy, and geometric mean ranking are obtained when using data balancing. On the other hand, fewer rules and conditions are obtained when using only the best attributes.

8 Discovered rules

In this section, some examples of different rules discovered by some of the algorithms are shown in order to compare their interpretability and usefulness for early identification of student with risk of failing and for making decisions about how to help this student. These rules show us the relevant factors and relationships that lead a student to pass or fail. All the values of these factors are available before the end of the year with the exception of the grades obtained in each subject. Although we use the final grades obtained by students in each subject to discover and evaluate the classification models, we must also use the partial grades obtained by the student in each subject at different times of the year (for example, in our case grades are available for three different dates) instead of the final grades to predict the performance of new students before the end of each year.

Figure 6 shows some examples of rules discovered by two different algorithms, one of each type (rule induction and decision trees). On the one hand, NNge rules are not very comprehensible because they are very long (they have a great number of conditions in the antecedent) and they use the OR operator (list of values between brackets and separated by commas). On the other hand, J48 rules are more comprehensible as they are shorter and do not use the OR operator.

Concerning the attributes that appear in the both previous models (see Table 11), it can be observed that the J48 algorithm only uses attributes about marks. It can be also observed that the NNge algorithm uses not only marks but also other attributes such as classroom/group enrolled, motivation to study, physical disability, smoking habits, age and the average score in the EXANI I. And in general, the score in Math is the most important attribute as it appears at the

Table 9 Classification results using cost-sensitive classification

Algorithm	TP rate	TN rate	Acc	GM	#Rules	#Conditions per rule	#Conditions
JRip	96.2	93.3	96.0	94.6	8.6	1.5	13.1
NNge	98.2	71.7	95.8	83.0	21.0	14.0	294.0
OneR	96.1	70.0	93.7	80.5	2.0	1.0	2.0
Prism	99.5	39.7	94.4	54.0	53.2	1.6	85.4
Ridor	96.9	58.3	93.4	74.0	6.0	1.7	9.9
ADTree	98.1	81.7	96.6	89.0	21.0	2.9	61.5
J48	95.7	80.0	94.3	87.1	41.9	2.8	121.3
RandomTree	96.6	68.3	94.0	80.4	251.3	3.4	863.1
REPTree	95.4	65.0	92.7	78.1	41.5	1.6	64.1
SimpleCart	97.2	90.5	96.6	93.6	10.2	5.1	56.5
ICRM v1	92.1	91.7	92.1	91.8	2.0	2.5	4.9
ICRM v2	94.4	86.7	93.7	90.3	3.0	3.0	9.1
ICRM v3	94.0	88.3	93.4	90.9	6.0	1.5	8.8

Table 10 Average ranking of classification results

Algorithm	TP rate	TN rate	Acc	GM	#Rules	#Conditions per rule	#Conditions
All attributes	1.92	3.38	2.46	3.46	1.85	2.23	2.38
Feature selection	2.62	2.69	2.46	2.69	1.62	1.92	2.00
Data balancing	2.23	1.54	2.08	1.54	3.00	2.85	3.15
Cost sensitive	3.00	2.15	2.54	2.23	2.00	2.31	2.23

```
NNGE classifier
------------------

  class FAIL IF : GROUP in {1B,1K,1u,1M,1o,1H} ^
      MOTIVATION_LEVEL in {Low,Regular,No} ^ PHYSICAL_DISABILITY in {Yes} ^
      TO_SMOKE in {Yes,ocasional} ^ AGE in {More_than_15} ^
      EXANI_I in {Poor,Very_Poor,Regular,Not_Presented} ^ MATE1 in {Poor,Not
      FIS1 in {Regular,Sufficient,Poor,Not_Presented} ^ CS1 in {Good,Suffici
      HUM1 in {Not_Presented} ^ TLR1 in {Sufficient,Regular,Poor ,Not_Preser
      ING1 in {Sufficient,Poor,Regular,Not_Presented} ^ COM1 in {Regular,Suf
  class FAIL IF : GROUP in {1B,1o,1C,1I,1L,1x,1g} ^
      MOTIVATION_LEVEL in {Regular,Low,No} ^ PHYSICAL_DISABILITY in {Yes} ^
      TO_SMOKE in {Yes,ocasional} ^ AGE in {15.0} ^
      EXANI_I in {Poor,Very_Poor} ^ MATE1 in {Poor,Not_Presented} ^
      FIS1 in {Poor} ^ CS1 in {Sufficient,Regular,Poor} ^ HUM1 in {Poor ,Goc
      TLR1 in {Sufficient,Poor} ^ ING1 in {Sufficient,Poor ,Regular,Not_Pres
  class FAIL IF : GROUP in {1u,1H} ^ MOTIVATION_LEVEL in {Low,Regular} ^
      PHYSICAL_DISABILITY in {Yes} ^ TO_SMOKE in {Yes} ^ AGE in {15.0} ^
      MATE1 in {Not_Presented,Poor} ^ FIS1 in {Not_Presented} ^ CS1 in {Not_
      HUM1 in {Not_Presented} ^ TLR1 in {Poor,Not_Presented} ^ ING1 in {Poor
  class PASS IF : GROUP in {1J,1I,1s,1q} ^
      MOTIVATION_LEVEL in {High} ^ PHYSICAL_DISABILITY in {No} ^
      TO_SMOKE in {No} ^ AGE in {15.0} ^
      EXANI_I in {Good,Very_Good} ^ MATE1 in {Good,Excellent,Very_Good} ^
      FIS1 in {Very_Good,Excellent,Good} ^ CS1 in {Good,Excellent,Very_Good
      HUM1 in {Good,Excellent,Very_Good} ^ TLR1 in {Very_Good,Excellent,Good
      ING1 in {Good,Very_Good,Excellent} ^ COM1 in {Good,Very_Good,Excellent
  class PASS IF : GROUP in {1J,1y,1a} ^ MOTIVATION_LEVEL in {High} ^ PHYSICA
      TO_SMOKE in {No} ^ AGE in {15.0} ^
      EXANI_I in {Very_Good,Excellent,Good} ^ MATE1 in {Very_Good,Excellent}
      CS1 in {Very_Good,Good} ^ HUM1 in {Very_Good,Excellent,Good} ^ TLR1 in
      ING1 in {Very_Good,Excellent,Good} ^ COM1 in {Very_Good,Excellent,Good
  class FAIL IF : GROUP in {1K,1u,1M,1H} ^ MOTIVATION_LEVEL in {Low,Regular}
```

```
J48 pruned tree
------------------

MATE1 = Sufficient
|   ING1 = Good: PASS
|   ING1 = Very_Good : PASS
|   ING1 = Sufficient: PASS
|   ING1 = Excellent: PASS
|   ING1 = Poor
|   |   HUM1 = Poor : PASS
|   |   HUM1 = Good: FAIL
|   |   HUM1 = Excellent: PASS
|   |   HUM1 = Very_Good : PASS
|   |   HUM1 = Regular: FAIL
|   |   HUM1 = Sufficient: PASS
|   |   HUM1 = Not_Presented: FAIL
|   ING1 = Regular: PASS
|   ING1 = Not_Presented: PASS
MATE1 = Good: PASS
MATE1 = Poor
|   FIS1 = Regular
|   |   HUM1 = Poor : PASS
|   |   HUM1 = Good: PASS
|   |   HUM1 = Excellent: PASS
|   |   HUM1 = Very_Good : PASS
|   |   HUM1 = Regular: PASS
|   |   HUM1 = Sufficient: PASS
|   |   HUM1 = Not_Presented: FAIL
|   FIS1 = Very_Good : PASS
|   FIS1 = Sufficient
|   |   CS1 = Good: PASS
|   |   CS1 = Sufficient
|   |   |   COM1 = Good: PASS
```

Fig. 6 Some rules discovered by the NNge and J48 algorithms using data balancing

```
IF (HUM1 != NP AND FIS1 != D AND MATE1 != NP AND ING1 != D AND CS1 != D ) THEN (CLASS = PASS)
ELSE (CLASS = FAIL)
```

Fig. 7 Some rules discovered by the ICRM v1 algorithm using best attributes

```
IF (MATE1 = Very_Poor  ) THEN (CLASS = FAIL)
ELSE IF (MATE1 = Not_Presented  ) THEN (CLASS = FAIL)
ELSE IF (MATE1 = Poor AND ING1 = Poor ) THEN (CLASS = FAIL)
ELSE IF (HUM1 = Not_Presented AND MOTIVATION_LEVEL = Low AND GROUP != 1J) THEN (CLASS = FAIL)
ELSE (CLASS = PASS)
```

Fig. 8 Some rules discovered by the ICRM v3 algorithm using data balancing

Table 11 Description of factors that appear in the discovered classification rules

Attribute	Description
MATE1	Score in Mathematics
ING1	Score in English
FIS1	Score in Physics
HUM1	Score in Humanities
CS1	Score in Computing
TLR1	Score in Reading and Writing
EXANI I	Average score in EXANI I
AGE	Age of the student in years
MOTIVATION LEVEL	Motivation to study
PHYSICAL DISABILITY	Physical Disability
TO SMOKE	Smoking habits
GROUP	Classroom/group enrolled

top of the decision trees and in almost all the rules; on the other hand, the grade point average in secondary school does not appear in any rule as might be expected.

Finally, Figs. 7 and 8 show some examples of rules discovered by our ICRM proposal. It produces a classifier that is like a decision tree. However, the accuracy and the low number of rules and conditions from ICRM contrast noticeably with the other models that have similar accuracy but consider a high number of rules and conditions; that is, the latter are less comprehensible models, which makes it difficult for teachers to detect and prevent student failure. In fact, these rules are the ones which predict student failure best, achieving a TN rate of up to 98.7 % using data balancing.

With regard to the attributes that appear in the models obtained, it can be seen that ICRM v1 only uses attributes about marks, but ICRM v3 also uses other attributes.

9 Discussion

In general, regarding the DM approaches used and the classification results obtained, the main conclusions are as follows.

- We have shown that classification algorithms can be used to successfully predict a student's academic performance and, in particular, to model the difference between Fail and Pass students.
- We have shown the utility of feature selection techniques when we have a great number of attributes. In our case, we have reduced the number of attributes used from the 77 initially available attributes to the best 15 attributes, obtaining fewer rules and conditions without losing classification performance.
- We have shown two different ways to address the problem of imbalanced data classification by rebalancing the data and considering different classification costs. In fact, rebalancing of the data has been able to improve the classification results obtained in TN rate, accuracy, and geometric mean.
- We have shown that our ICRM model has obtained the best overall TN ratio and GM when using data balancing. The model does not require cost-sensitive costs but uses appropriate data balancing. In that case, it is able to obtain the most accurate and comprehensible classification models by means of classification rule bases with a low number of rules and conditions. Specifically, it obtains the best predictions of student failure (98.7 %).

Regarding the specific knowledge extracted from the classification models obtained, the main conclusions are as follows.

- White box classification algorithms obtain models that can explain their predictions at a higher level of abstraction by IF-THEN rules. In our case, induction rule algorithms produce IF-THEN rules directly, and decision trees can be easily transformed into IF-THEN rules. IF-THEN rules are one of the most popular forms of knowledge representation, due to their simplicity and comprehensibility. These types of rules are easily understood and interpreted by non-expert DM users, such as instructors, and can be directly applied in decision-making processes.
- Concerning the specific factors or attributes related with student failure, there are some specific values that appear most frequently in the classification models obtained. For

example, the values of scores/grades that appear most frequently in the obtained classification rules are the values "Poor", "Very Poor", and "Not Presented" in the subjects of Physics, Humanities, Math, and English. Other factors frequently associated with failing are being over 15 years of age, having more than one sibling, attending an evening class/group, and having a low level of motivation to study.

In this study we have used the students' marks and we have not focused solely on social, economic, and cultural attributes for two main reasons. The first is that we obtained bad classification results when we did not consider the marks. Secondly, the grades obtained by students have been previously used in a great number of other similar studies [10, 27]. Finally, starting from models obtained by classification algorithms, an early warning system can be developed in order to alert teachers about students who are potentially at risk of failing so that they can provide help. For example, when students are detected as being potentially at risk of failing, they should be assigned a personal tutor in order to receive specific academic support, motivation, and guidance for preventing student failure.

10 Conclusions

As we have seen, predicting student failure at school can be a difficult task not only because it is a multifactor problem (in which there are a lot of personal, family, social, and economic factors that can be influential) but also because the available data are normally imbalanced (most of the students pass to the next course). To resolve these problems, we have shown the use of different DM algorithms and approaches for predicting student failure. We have carried out several experiments using real data from high school students in Mexico. We have applied different classification approaches for predicting the academic status or final student performance at the end of the course. We have proposed a genetic programming model to obtain accurate and comprehensible classification rules. Furthermore, we have shown that some approaches such as selecting the best attributes, cost-sensitive classification, and data balancing can also be very useful for improving accuracy. Finally, as the next step in our research, we aim to carry out more experiments using more data and also from different educational levels (primary, secondary, and higher) to test whether the same performance results are obtained with different DM approaches (feature selection, data balancing, and cost-sensitive classification) and our ICRM algorithm.

Acknowledgements This research has been supported by projects of the Regional Government of Andalucía and the Ministry of Science and Technology, P08-TIC-3720, TIN-2011-22408, FPU grant AP2010-0042 and FEDER funds.

References

1. Aloise-Young PA, Chavez EL (2002) Not all school dropouts are the same: Ethnic differences in the relation between reason for leaving school and adolescent substance use. Psychol Sch 39(5):539–547
2. Araque F, Roldan C, Salguero A (2009) Factors influencing university drop out rates. Comput Educ 53:563–574
3. Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Chapman & Hall, New York
4. Cano A, Zafra A, Ventura S (2011) An EP algorithm for learning highly interpretable classifiers. In: Proceedings of the 10th international conference on intelligent systems design and applications, ISDA'11, pp 325–330
5. Cendrowska J (1987) Prism: an algorithm for inducing modular rules. Int J Man-Mach Stud 27(4):349–370
6. Cohen WW (1995) Fast effective rule induction. In: Twelfth international conference on machine learning, pp 115–123
7. Dekker GW, Pechenizkiy M, Vleeshouwers JM (2009) Predicting students drop out: a case study. In: International conference on educational data mining
8. Diosan L, Rogozan A, Pecuchet J (2012) Improving classification performance of support vector machine by genetically optimising kernel shape and hyper-parameters. Appl Intell 36:280–294
9. Elkan C (2001) The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence, pp 1–6
10. Fourtin L, Marcotte D, Potvin P, Roger E, Joly J (2006) Typology of students at risk of dropping out of school: description by personal, family and school factors. Eur J Psychol Educ 21(4):363–383
11. Freund Y, Mason L (1999) The alternating decision tree algorithm. In: Proceedings of the 16th international conference on machine learning, pp 124–133
12. Gu Q, Cai Z, Zhu L, Huang B (2008) Data mining on imbalanced data sets. In: Proceedings of international conference on advanced computer theory and engineering, pp 1020–1024
13. Hall MA, Holmes G (2002) Benchmarking attribute selection techniques for data mining. Tech. rep, University of Waikato, Department of Computer Science, Hamilton, New Zealand
14. Hämäläinen W, Vinni M (2011) Classifiers for educational data mining. Chapman & Hall/CRC, London
15. Hernández MM (2002) Causas del Fracaso Escolar. In: XIII Congreso de la Sociedad Española de Medicina del Adolescente
16. Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Mach Learn 11:63–91
17. Klösgen W, Zytkow JM (2002) Handbook of data mining and knowledge discovery. Oxford University Press, London
18. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th international joint conference on artificial intelligence, pp 1137–1143
19. Kotsiantis S (2009) Educational data mining: a case study for predicting dropout-prone students. Int J Knowl Eng Soft Data Paradig 1(2):101–111
20. Kotsiantis S, Patriarcheas K, Xenos M (2010) A combinational incremental ensemble of classifiers as a technique for predicting students performance in distance education. Knowl-Based Syst 23(6):529–535
21. Kotsiantis SB, Pintelas PE (2005) Predicting students' marks in Hellenic Open University. In: IEEE international conference on advanced learning technologies, pp 664–668
22. Lykourentzou I, Giannoukos I, Nikolopoulos V, Mpardis G, Loumos V (2009) Dropout prediction in e-learning courses through the combination of machine learning techniques. Comput Educ 53:950–965
23. Márquez-Vera C, Romero C, Ventura S (2011) Predicting school failure using data mining. In: Educational data mining conference

24. Martínez D (2001) Predicting student outcomes using discriminant function analysis. In: Annual meeting of the research and planning group, pp 163–173
25. McDonald B (2004) Predicting student success. J Math Teach Learn 1:1–14
26. Mendez G, Buskirk TD, Lohr S, Haag S (2008) Factors associated with persistence in science and engineering majors: an exploratory study using classification trees and random forests. J Eng Educ 97:57–70
27. Moseley LG, Mead DM (2008) Predicting who will drop out of nursing courses: a machine learning exercise. Nurse Educ Today 28:469–475
28. Pan W (2012) The use of genetic programming for the construction of a financial management model in an enterprise. Appl Intell 36:271–279
29. Parker A (1999) A study of variables that predict dropout from distance education. Int J Educ Technol 1(2):1–11
30. Quadril MN, Kalyankar NV (2010) Drop out feature of student data for academic performance using decision tree techniques. J Comput Sci Technol 10:2–5
31. Quinlan JR (1983) C45. Programs for machine learning. Morgan Kaufman, San Mateo
32. Richards D (2009) Two decades of RDR research. Knowl Eng Rev 24(2):159–184
33. Slavin RE, Karweit NL, Wasik BA (1994) Preventing early school failure. Allyn & Bacon, Needham Heights
34. Romero C, Espejo PG, Zafra A, Romero JR, Ventura S (2012, in press) Web usage mining for predicting final marks of MOODLE students. Comput Appl Eng Educ, 1–12
35. Romero C, Ventura S (2010) Educational data mining: a review of the state of the art. IEEE Trans Syst Man Cybern 6:601–618
36. Roy S (2002) Nearest neighbor with generalization. Master's thesis, University of Canterbury, Christchurch, New Zealand
37. Deegalla S, Bostrom H (2006) Reducing high-dimensional data by principal component analysis vs random projection for nearest neighbor classification. In: International conference on machine learning and applications, pp 245–250
38. Superby JF, Vandamme JP, Meskens N (2006) Determination of factors influencing the achievement of the first-year university students using data mining methods. In: Educational data mining workshop, pp 1–8
39. Tinto V (1987) Leaving college: rethinking the causes and curse of students attrition. University of Chicago Press, Chicago
40. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
41. Veitch W (2004) Identifying characteristics of high school dropouts: data mining with a decision tree model. In: Annual meeting of the American educational Research Association, pp 1–11
42. Wang AY, Newlin MH (2002) Predictors of web-based performance: the role of self-efficacy and reasons for taking an on-line class. Comput Hum Behav J 18:151–163
43. Wegner L, Flisher AJ, Chikobvu P, Lombard C, King G (2008) Leisure boredom and high school dropout in Cape Town, South Africa. J Adolesc 31:421–431
44. Whigham PA (1996) Grammatical bias for evolutionary learning. PhD Dissertation, University of New South Wales
45. Witten IH, Eibe F, Hall MA (2011) Data mining, practical machine learning tools and techniques. Morgan Kaufman, San Mateo

Carlos Márquez-Vera is a professor in the Preparatory Unit Academic of the Autonomous University of Zacatecas, Mexico. He received the M.Sc. Degree in Physics Education from the University of Havana, Cuba in 1997. Currently is a Ph.D. student of the University of Córdoba, Spain and his research interests lie in Educational Data Mining.

Alberto Cano was born in Cordoba, Spain, in 1987. He is currently a Ph.D. student with a M.Sc. in Computer Science. His research is performed as a member of the Knowledge Discovery and Intelligent Systems Research Laboratory, and is focussed on general purpose GPU systems, parallel computing, soft computing, machine learning, data mining and its applications.

Cristóbal Romero received the B.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1996 and 2003, respectively. He is currently an Associate Professor in the Department of Computer Science and Numerical Analysis, University of Cordoba, Spain. He has authored or coauthored more than more than 50 international publications, 20 of them published in international journals. He is a member of the Knowledge Discovery and Intelligent Systems Research Laboratory and his research interests include applying data mining in e-learning systems. He is a member of the IEEE Computer Society, the International Educational Data Mining (EDM) Working Group, and the steering committee of the EDM Conferences.

Sebastián Ventura was born in Cordoba, Spain, in 1966. He received the B.Sc. and Ph.D. degrees from the University of Cordoba, in 1989 and 1996, respectively. He is currently Associate Professor in the Department of Computer Science and Numerical Analysis, the University of Cordoba, where he heads the Knowledge Discovery and Intelligent Systems Research Laboratory. He is the author or coauthor of more than 90 international publications, 35 of which have been published in international journals. He has also been engaged in twelve research projects (being the coordinator of three of them) supported by the Spanish and Andalusian governments and the European Union, concerning several aspects of the area of evolutionary computation, machine learning, and data mining and its applications. His current main research interests are in the fields of soft-computing, machine learning, data mining and its applications.