Saber PRO success prediction using decision tree algorithm

Simón Álvarez Ospina	David Madrid Restrepo	Miguel Correa	Mauricio Toro
Universidad Eafit	Universidad Eafit	Universidad Eafit	Universidad Eafit
Colombia	Colombia	Colombia	Colombia
salvarezo1@eafit.edu.co	dmadridr@eafit.edu.co	macorream@eafit.edu.co	mtorobe@eafit.edu.co

ABSTRACT

This research aims to predict students' performance in statal test 'Saber PRO' through their past results and the different variables which could affect their academic performance. This problem opens up opportunities to develop in different areas, like education, tools that can contribute to improve the results of students in their college career.

Our project is related with some questions such as finding the most influential variables in students results and the methods that can be applied to improve their global score in statal tests.

Keywords

Decision trees, machine learning, academic success, standardized student scores, test-score prediction.

1. INTRODUCTION

Due to technology expansion and the expected results that some algorithms produced for the resolution of nonlinear problems [1], the application of different methods of prediction is sought by other knowledge areas where systems have increased their importance. One of the essential challenges that Latin America has gone through is the quality of the education and, despite the different efforts in others knowledge areas, sufficiently noticeable measures have not been taken to improve it. Thus, this project intends to contribute with tests and new interpretations of decision trees' results for the development of a quality higher education and the training of competent professionals.

1.1. Problem

Predicting students' academic success is a matter where researchers have been developing different solutions, which deliver varied but no determinant results. Finding an algorithm that yields an appropriate solution can help to understand which variables have the greatest impact in student's academic performance.

1.2 Solution

In this work, we focused on decision trees because they provide great explainability. We avoid black-box methods such as neural networks, support-vector machines and random forests because they lack explainability.

For this problem, we used CART decision trees that learn from the 'Saber 11' information of a student how to predict if it would be successful or not in the 'Saber Pro'.

1.3 Article structure

In what follows, in Section 2, we present related work to the problem. Later, in Section 3 we present the datasets and methods used in this research. In Section 4, we present the algorithm design. After, in Section 5, we present the results. Finally, in Section 6, we discuss the results and we propose some future work directions.

2. RELATED WORK

2.1 Student's academic success in their first year through decision trees.

Josip Mesarić and Dario Šebalj in their project were looking for solutions to predict student's success rate in their first year through the study of variables that could affect their performance, dividing their results in two groups. For this, they used different decision trees algorithms like ID3 and J4.8, besides random forests and REPTree. In the end, they achieved a 79.35% precision with the REPTree algorithm. [1]

2.2 Data mining and decision trees to find the most influential variables in higher education.

Qasem et. al. made use of different data mining techniques to find the most influential variables in the performance of students on higher education. Essentially, they used a classification method in algorithms like ID3, C4.5 and Naïve Bayes to achieve their goal. The precision they got with the 3 methods did not reach the 40% threshold. [2]

2.3 Prediction of students' academic performance through classification algorithms ID3 and C4.5.

Kalpesh Adhatrao et. al. studied the problem of how to manage the performance of the students with low efficiency through the prediction of their future results with data mining techniques. They used the algorithms ID3 and C4.5, the second one used with the purpose of complement ID3 deficiencies. They obtained successful results, achieving a 72.275% of accuracy. [3]

2.4 Prediction of students' final results through the algorithm of Classification And Regression Trees (CART).

Julianti Kasih, Mewati Ayub and Sani Susanto in their research were looking for a more practical way to predict the results of Indonesian students, which they classified in three different levels. Previously, they used discriminant analysis, where they concluded that was unpractical for the kind of problem they were studying. In this new paper, they were based in the CART method to explain its efficiency in prediction. Their research did not work with known results, since its purpose was to explain the differences between the methods that they used to achieve a simpler functional algorithm. [4]

3. MATERIALS AND METHODS

In this section, we explain how the data was collected and processed and, after, different solution alternatives considered to choose a decision-tree algorithm.

3.1 Data Collection and Processing

We collected data from the *Colombian Institute for the Promotion of Higher Education* (ICFES), which is available online at ftp.icfes.gov.co. Such data includes anonymized Saber 11 and Saber Pro results. Saber 11 scores of all Colombian high schools graduated from 2008 to 2014 and Saber Pro scores of all Colombian bachelor-degree graduates from 2012 to 2018 were obtained. There were 864,000 records for Saber 11 and records 430,000 for Saber Pro. Both Saber 11 and Saber Pro, included, not only the scores but also socio-economic data from the students, gathered by ICFES, before the test.

In the next step, both datasets were merged using the unique identifier assigned to each student. Therefore, a new dataset that included students that made both standardized tests was created. The size of this new dataset is 212,010 students. After, the binary predictor variable was defined as follows: Does the student score in Saber Pro is higher than the national average of the period?

It was found out that the datasets were not balanced. There were 95,741 students above average and 101,332 students below average. We performed under sampling to balance the dataset to a 50%-50% ratio. After undersampling, the final dataset had 191,412 students.

Finally, to analyze the efficiency and learning rates of our implementation, we randomly created subsets of the main dataset, as shown in Table 1. The dataset was divided into 70% for training and 30% for testing. Datasets are available at https://github.com/mauriciotoro/ST0245-Eafit/tree/master/proyecto/datasets .

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
Train	15,000	45,000	75,000	105,000	135,000
Test	5,000	15,000	25,000	35,000	45,000

Table 1. Number of students in each dataset used for training and testing.

3.2 Decision-tree algorithm alternatives

Now, we are going to show different algorithms used to automatically build a binary decision tree.

3.2.1 Iterative Dichotomiser 3 (ID3)

Iterative Dochotomiser 3 (ID3), created by Ross Quinlan, is the most used algorithm to generate decision trees. It consists of a measure called entropy, which is the responsible of finding the amount of uncertain information. This makes of ID3 a greedy heuristic. This method has complications when try to work with information of continuous intervals and, by its composition, it doesn't guarantee an optimal solution. Its complexity, in the worst scenario, is from O(h), where h is the height of the tree. [5]

The following picture represents how the algorithm works.

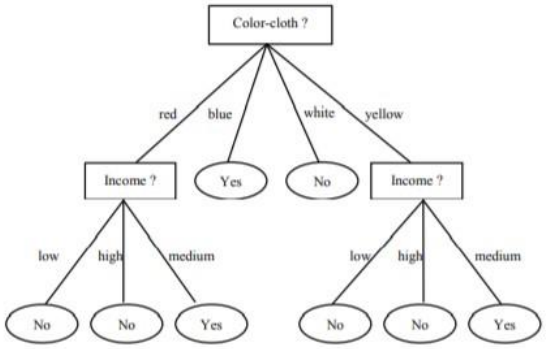


Figure 1: How ID3 algorithm works. [5]

C4.5 is an algorithm also created by Quinlan, that searches to cover the ID3 inefficiencies. C4.5 avoid some steps that ID3 uses and focus in other options to get a better base condition. Another of C4.5 advantages is that it allows to work with continuous variables, which means that it extends its area of application. Because it uses only one recursively-call, its complexity must be O(h).

3.2.3 Classification and regression trees (CART)

It is a term to study both analysis of classification and regression trees. The classification trees are in charge of deliver a result of the same type of variable that is in the

sample, while the regression trees are in charge of work with variables that have real values, like the students' grades or the work time spent in an office.

3.2.4 Random Forests

Random Forests is an algorithm used to create multiple decision trees and get the one with the best results through a selection of a random subset of variables in the sample. Although it seems counterintuitive think that this method is efficient, in practice random forests helps to find and rank the most important variables in a regressive or a classification problem, which we will focus in this project. The complexity of the random forests can become to be O(k*nlog(n)), where n is the number of registers and k is the number of variables.

4. ALGORITHM DESIGN AND IMPLEMENTATION

In what follows, we explain the data structure and the algorithms used in this work. The implementation of the data structure and algorithm is available at GitHub[1].

4.1 Data Structure

The data structure used in this algorithm is a binary decision tree, which is used to predict if someone will be over average in 'Saber pro' using their score in 'Saber 11'.

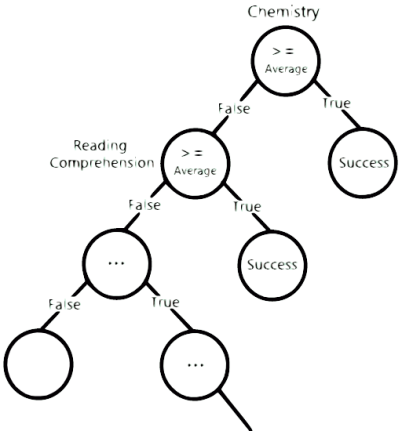
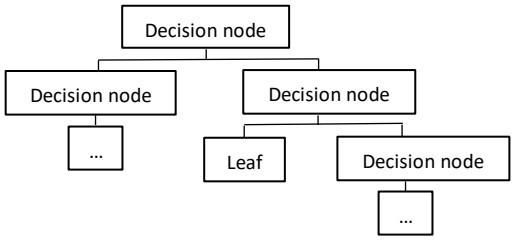


Figure 1: A binary decision tree to predict the performance in Saber Pro based on the results of Saber 11. Most right nodes represent those with a high probability of success, mid nodes medium probability and most left a low probability of success.

4.2 Algorithms

A CART decision tree will be implemented in this algorithm. Here is an example of the idea behind the implementation of this tree.



4.2.1 Training the model

The algorithm will take some data and will see the impact in the predictions through the Gini impurity, and with that will generate the condition for the binary decision tree.

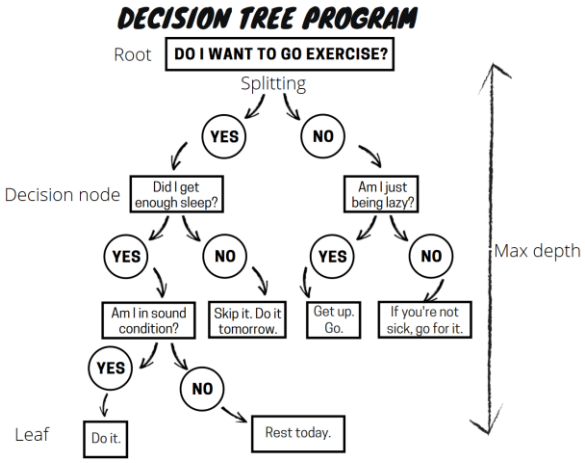


Figure 2: Training a binary decision tree using *CART*. In this example, we show a model to predict if someone will go out and exercise or not.

4.2.2 Testing algorithm

We use an existing tree built with the dataset students information to classify new students that based in their results in 'Saber 11' and knowing if it's above the average in 'Saber PRO' will help to know the algorithms reliability.

4.3 Complexity analysis of the algorithms

For the worst case, we suppose that the algorithm suggests the same condition in the decision nodes at the same level and that there won't be a leaf until the algorithm reaches its maximum height. Thus, the time and memory complexity is as we present in the table 2 and 3 respectively.

[1] http://www.github.com/dmadridr/blob/master/

Algorithm	Time Complexity
Train the decision tree	$O(N^2*M)$
Test the decision tree	O(N*M)

Table 2: Time Complexity of the training and testing algorithms. *(Where N is the height of the tree and M is the number of students).*

Algorithm	Memory Complexity
Train the decision tree	O(M*2^N)
Test the decision tree	O(M)

Table 3: Memory Complexity of the training and testing algorithms. *(Where N is the height of the tree and M is the number of students).*

4.4 Design criteria of the algorithm

The data structure used in this algorithm is the ideal for many reasons, for example, it is so fast to get the results by discarding one of the sides of the tree. Another example is that it is easier to implement the binary decision tree for this application.

5. RESULTS

5.1 Model evaluation

In this section, we present some metrics to evaluate the model. Accuracy is the ratio of number of correct predictions to the total number of input samples. Precision. is the ratio of successful students identified correctly by the model to successful students identified by the model. Finally, Recall is the ratio of successful students identified correctly by the model to successful students in the dataset.

5.1.1 Evaluation on training datasets

In what follows, we present the evaluation metrics for the training datasets in Table 3.

	Dataset 1	***Dataset 2***	***...Dataset n***
Accuracy	0.76	0.76	0.81
Precision	0.76	0.76	0.81
Recall	0.76	0.77	0.82

Table 3. Model evaluation on the training datasets.

5.1.2 Evaluation on test datasets

In what follows, we present the evaluation metrics for the test datasets in Table 4.

	Dataset 1	***Dataset 2***	***...Dataset n***
Accuracy	0.67	0.68	0.74
Precision	0.67	0.68	0.74
Recall	0.67	0.68	0.79

Table 4. Model evaluation on the test datasets.

5.2 Execution times

In the next time, we present the execution time of datasets with 45000, 75000 and 135000 students that were used to train different decision trees and how much time it costed to classify new data into it.

Dataset size	***45000***	***75000***	***135000***
Training time	339.52 s	570.04 s	1034.26 s
Testing time	35.74 s	59.36 s	99.12 s

Table 5: Execution time of the *CART* algorithm for different datasets.

5.3 Memory consumption

We present memory consumption of the binary decision tree, for different datasets, in Table 6.

	Dataset 1	***Dataset 2***	***...Dataset n***
Memory consumption	7891 MB	7907 MB	7788 MB

Table 6: Memory consumption of the binary decision tree for different datasets.

6. DISCUSSION OF THE RESULTS

The results obtained were as expected, since the precision of the algorithm is high enough to take it into account when allocating financial resources for scholarships. It is also useful for underperforming students to reinforce weak areas.

6.1 Future work

For a future upgrade is required to migrate the project to a faster and more memory-friendly programming language. But would be handy if other than that, the decision tree algorithm would be rethought to work as a random forest, because it can handle with thousands of variables that would be useful for larger csv files.

REFERENCES

1. Mesarić, J., Šebalj, D. Decision trees for predicting the academic success of studens, *Croatian Operational Research Review, (7).* Retrieved 2016, from University of Josip Juraj Strossmayer in Osijek.

2. Al-Radaideh, Q., Al-Shawakfa, E., and Al-Najjar, M. Mining Student Data Using Decision Trees. In *The 2006 International Arab Conference on Information Technology, (Jordan, 2006).*

3. Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R. and Honrao, V. Predicting Stundets' Performance Using ID3 and C4.5 Classification Algorithms. *International Journal of Data Mining & Knowledge Management Process 3 ().* Retrieved September, 2013, from Fr. C.R.I.T, Navi Mumbai, Maharashtra, India.

4. Kasih, J., Ayub, M. and Susanto, S. Predicting student's final passing results using the Classification and Regression Trees (CART) algorithm. *World Transactions on Engineering and Technology Education, 11 (1).* Retrieved on 2013, from Maranatha Christian University and Parahyangan Catholic University.

5. De-lin, L., Jin, C. and Fen-xiang, M. An Improved ID3 Decision Tree Algorithm. *Proceedings of 2009 4th International Conference on Computer Science & Education.*