Analysis of academic success according to the test "Saber pro" using decision trees

Juan David Echeverri Universidad Eafit Colombia jdecheverv@eafit.edu.co	Octavio Vásquez Universidad Eafit Colombia ovasquezz@eafit.edu.co	Miguel Correa Universidad Eafit Colombia macorream@eafit.edu.co	Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co

For each version of this report: 1. Delete all text in red. 2. Adjust spaces among words and paragraphs. 3. Change the color of all the texts to black.

Red text = Comments

Black text = Miguel and Mauricio's contribution

Green text = To complete for the 1st deliverable

Blue text = To complete for the 2nd deliverable

Violet text = To complete for the 3rd deliverable

ABSTRACT

Due to the high's quantity of desertion in universities, several studies have been made to predict the causes of it, but not too much people have concerned about the studies related with the probability of success of the students. Calculating the probability of success of the students may be quite useful for several reasons. For example, it will allow the creation of better circumstances for the students, increasing the success ratio and the results of them. Also, it could improve the education system, allowing it to focus on the student´s problems and needs. In other hand, its necessary to say that the information acquired through this project may help to solve related problems such academic desertion, academic failure, and welfare of the students.

Which is the algorithm you proposed? What results did you achieve? What are the conclusions of this work? Abstract should have **at most 200 words**. (*In this semester, you should summarize here execution times, memory consumption, accurracy, precision and sensibility*)

Keywords

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

1. INTRODUCTION

Throughout the years, the test Saber Pro in Colombia was the way of check the abilities of the students after finishing their careers. The purpose of this test is to check the abilities of the students and the effectiveness of the different universities around country, and it is also a prerequisite for the student's graduation, allowing the government to make sure that they have nice professionals for the future. Getting an effective way to predict the results in this test is very important to generate a math graph with spoused data, and with these prognostics the government could make an idea about the conditions that allows the students to have a good performance in the test

1.1. Problem

One of the problems in this topic is how the variables affect the students and which changes on them provide more benefits in their performance. We focus in many variables like how old of student, parents' income, the professional career to course, the results in the test Saber 11, the gender, the socioeconomical stratum, how many hours they use to explore information in internet and other variables.

1.2 Solution

Due to the fact that black box methods (such as neural networks, support-vector machines and random forests) don't go much into detail and directly tests the functionality of the software, we decided to avoid them. Instead, in this work, we focused on decision trees because they provide better explainability. Specifically, we decided to use the c4.5 algorithm, because it allows us to divide the dataset and select the half that contains the higher gain of information. Also, it allows us to make a complex type code, where, according to Bruno López Takeyas, "the values are assigned to a number of variables of group with its respective possible result for each of them"[9], this due that the scores that an student can get in the "Saber Pro" test are related with different aspects of his life, like the quantity of books that his family possess, the school where he studied the high school, whether or not his school was bilingual, where he is from (either country or state), or others. Separating properly our dataset with these variables will allow us to predict in a more efficient way the probability of success of a student, making the decision tree a quite powerful tool to solve the problematic.

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 How Predicting the Academic Success of Students of the ESPAM MFL: A Preliminary Decision Trees Based Study

The topic of this article is how at own project, they use the decision tree's for predict the performance of the students in ESPAM, considering many aspects like at what does a student can do to improve their ratings. For the method used here they pre-processed the data to duplicate the instances and delete the loss data, representing each variable with numeric data, and they assign three categories, "Acceptable", "Good", "Excellent". Finally, they apply this structure in all variables (1086). The accuracy in this investigation have a precision of 55% and 49%. [1]

2.2 Performance Prediction of Engineering Students using Decision Trees

Through the project described in the article, the authors managed to create a model capable of predict the students' performance in the first year of engineering exam. They used the algorithm J48, managing to reach 60.46% and 69.94% of accuracy (they did two models), showing precisely those who were more likely to fail, allowing the university to counsel them so they could improve their results. [2]

2.3 A CHAID based performance prediction model in educational data mining

In this article, the authors developed a prediction model to be able to analyze the relation between the variables and the performance of students at higher secondary school. The features such as medium of instruction, marks obtained in secondary education, location of school, living area and type of secondary education were the most important ones. The algorithm used was CHAID, reaching 44.69% of prediction accuracy. [3]

2.4 Early prediction of student success: Mining student enrollment data

In this investigation Z. J. Kovacic presents an investigation of case of enlistment information mining, which can be used to foreshadow the success of students. The calculations were carried out by the CHAID and CART algorithms on the substitute recruitment information for the ordinary data frames for New Zealand Polytechnic students, thus achieving two trees that classify effective and ineffective substitutes. the precision obtained with CHAD and CART were 59.4% and 60.5% respectively. [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 under sampling, 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

3.2.1 ID3

ID3 is an algorithm invented by Ross Quinlan. It uses a greedy approach to build a decision tree from top to down, selecting the best feature at the moment to create a node. ID3 is only used for classification problems with nominal features. Its complexity its $O(m \cdot n^2)$ where m is the size of the training data and n is the number of attributes.

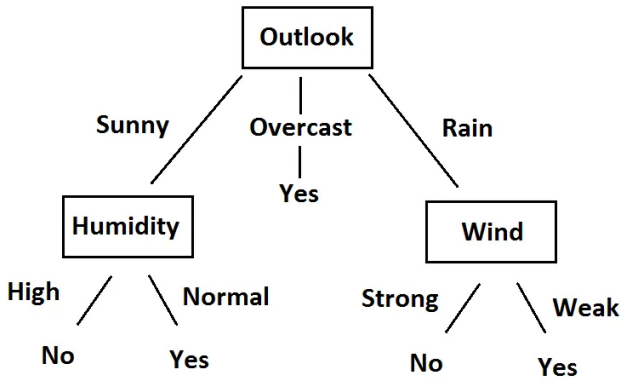


Figure 1 [5]

3.2.2 C4.5

C4.5 is one of the successors of ID3. C4.5 made several improvements in regard to its predecessor, C4.5 uses Gain ratio as an attribute selection measure. Also, C4.5 can handle both discrete and continuous attribute. Its complexity its $O(m \cdot n^2)$ where m is the size of the training data and n is the number of attributes.

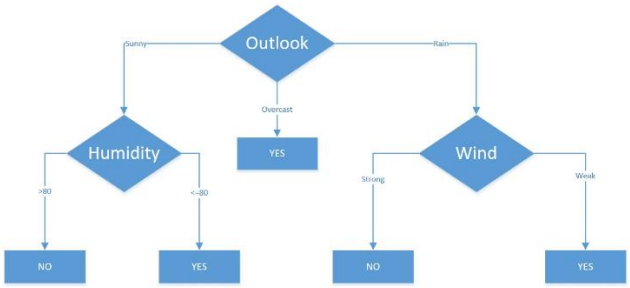


Figure 2 [6]

3.2.3 CART

The CART algorithm is the acronym for Classification and Regression Trees, it was designed by Breiman et al. In this algorithm binary decision trees are generated, so that each node is divided into exactly two branches, true and false. The complexity is O(v·nlogn) the first loop is the v, the second loop is n and the third loop are identical to the second.

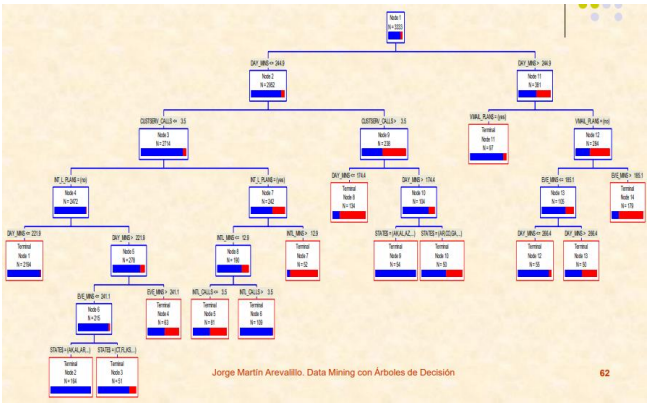


Figure 3 [7]

3.2.4 CHAID

The CHAID algorithm analyzes the values for each variable to predict and through the Chi-square, which was created in 1980 by G. V. Kass and later adapted by Magidson, J. in 1994 and allows us to work with a categorical dependent variable.

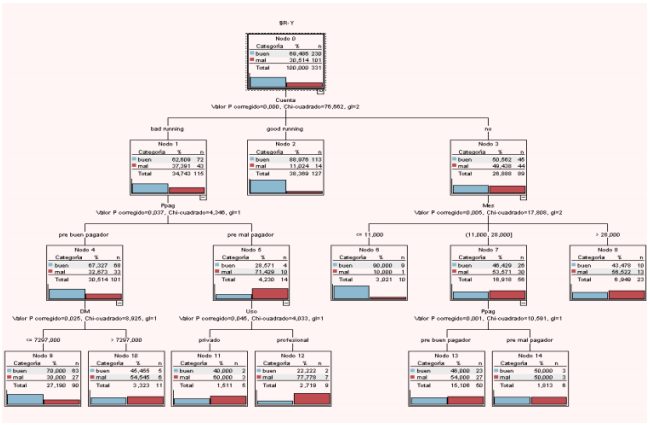
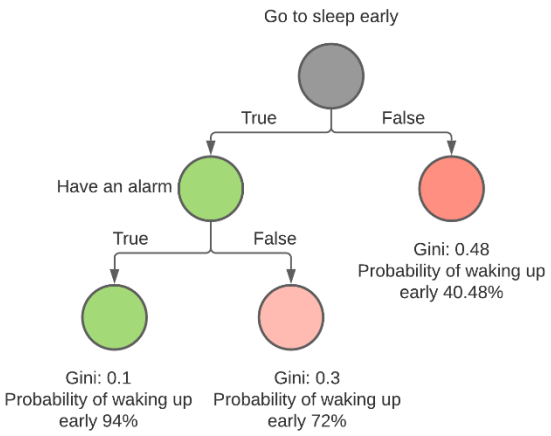


Figure 4 [8]

4. ALGORITHM DESIGN AND IMPLEMENTATION

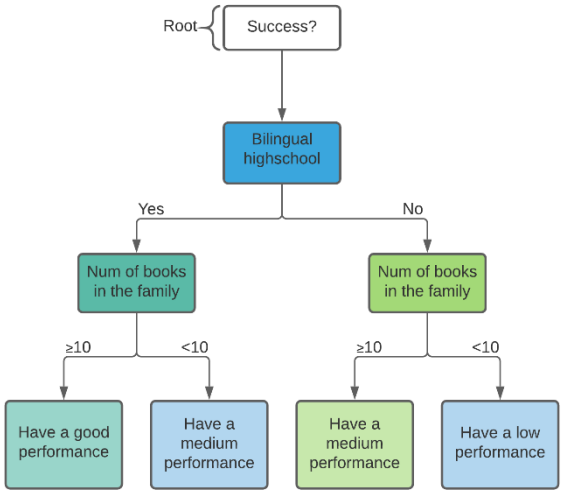
4.1 Data Structure

The data structure that we used to predict the success of the students in the test “Saber pro” was a binary decision tree (see figure 5). It stores the data in nodes, which each node could point to another 2 nodes. The branches on the tree depend of certain conditions that are preselected and it allows the dataset to be filtered over and over. Selecting these conditions properly will filtrate the dataset better, allowing us to know under which conditions the people are more likely to success.



4.2 Algorithms

We used the algorithm c4.5 to solve the problematic, it is designed to filter the datasets according to the conditions that the student matches. It is the responsible to build up the decision tree according to the given parameters, and then we are able to classify new data by calculating with our algorithm the Gini impurity. This calculation allows us to see how efficient works each filter and makes easier to classify new data using the decision tree and therefore giving a more accurate result.



4.2.1 Training the model

The conditions used to train the model, and therefore the conditions that the binary decision tree is built around are: if the student took a preparation course, took an extern support course, made a simulacrum type “icfes”, the number of books of the family, the department of residence, whether or not the student has internet, if he works currently, if his school is bilingual and the average of his results in the test ”Saber 11”.

4.2.2 Testing algorithm

After the decision tree were built up, the other algorithm that we made proceeds to calculate the Gini impurity of the nodes. This calculation basically says how impure is each node. For example, if we tag with 1 the successful students and with 0 the others each student of a given dataset, the node is purer when the tags are more separated. That means that Gini impurity shows how mixed are the tags and how well is working the filter (condition used to build the tree) to separate the data. With that information, classify new information should not be a problem, because we already know which conditions are more likely to filter a successful student, and by that, making the algorithm able to predict with great accuracy the probability of success of students.

4.3 Complexity analysis of the algorithms

Explain in your own words the analysis for the worst-case using O notation. How did you calculate such complexities?

Algorithm	Time Complexity
Train the decision tree	$O(N^2*M^2)$
Test the decision tree	$O(N^3*M*2^N)$

Table 2: Time Complexity of the training and testing algorithms. *(Please explain what do N and M mean in this problem.)*

Algorithm	Memory Complexity
Train the decision tree	$O(N*M*2^N)$
Test the decision tree	O(1)

Table 3: Memory Complexity of the training and testing algorithms. *(Please explain what do N and M mean in this problem.)*

4.4 Design criteria of the algorithm

Explain why the algorithm was designed that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory consumption. Examples of non-objective criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, etc. Remember: This is 40% of the project grading.

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	0.7	0.75	0.9
Precision	0.7	0.75	0.9
Recall	0.7	0.75	0.9

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	0.5	0.55	0.7
Precision	0.5	0.55	0.7
Recall	0.5	0.55	0.8

Table 4. Model evaluation on the test datasets.

5.2 Execution times

Compute execution time for each dataset in GitHub. Measure execution time 100 times for each dataset and report average execution time for each dataset.

	Dataset 1	***Dataset 2***	***...Dataset n***
Training time	10.2 s	20.4 s	5.1 s
Testing time	1.1 s	1.3 s	3.3 s

Table 5: Execution time of the *(Please write the name of the algorithm, C4.5, ID3)* 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	10 MB	20 MB	5 MB

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

To measure memory consumption, you should use a profiler. A very good one for Java is VisualVM, developed by Oracle, http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html For Python, use C Profiler.

6. DISCUSSION OF THE RESULTS

Explain the results obtained. Is precision, accuracy and sensibility appropriate for this problem? Is the model over-fitting? Is memory consumption and time consumption appropriate? *(In this semester, according to the results, can this be applied to give scholarships or to help students with low probability of success? For which one is better?)*

6.1 Future work

Answer, what would you like to improve in the future? How would you like to improve your algorithm and its implementation? What about using random forest?

ACKNOWLEDGEMENTS

Identify the kind of acknowledgment you want to write: for a person or for an institution. Consider the following guidelines: 1. Name of teacher is not mentioned because he is an author. 2. You should not mention websites of authors of articles that you have not contacted. 3. You should mention students, teachers from other courses that helped you.

As an example: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved the manuscript.

REFERENCES

[1] J. M. Carrillo and J. Parraga-Alava. How Predicting the Academic Success of Students of the ESPAM MFL: A Preliminary Decision Trees Based Study. IEEE Third Ecuador Technical Chapters Meeting (ETCM). ESPAMMFL, Cuenca, 2018, pp. 1-6, https://bit.ly/2E8O4cS

[2] R. R. Kabra, & R. S. Bichkar. Performance Prediction of Engineering Students using Decision Trees. International Journal of Computer Applications. Education Foundation's College of Engineering and Management, Ahmednagar, 2011, 0975 – 8887. https://bit.ly/2FlAAeu

[3] M. Ramaswami and R. Bhaskaran. A CHAID based performance prediction model in educational data mining. IJCSI International Journal of Computer Science Issues. Madurai Kamaraj University, Madurai, 2010, 1694-0784. https://bit.ly/3iHlCOg

[4] Z. J. Kovacic. Early prediction of student success: Mining student enrollment data, Proceedings of Informing Science & IT Education Conference (InSITE), Open Polytechnic, Wellington, 2010. https://bit.ly/31WLwqF

[5] Sefik Ilkin Serengil. (2017). A Step by Step ID3 Decision Tree Example. Retrieved from https://bit.ly/314359c

[6] Sefik Ilkin Serengil. (2018). A Step by Step C4.5 Decision Tree Example. Retrieved from https://bit.ly/31XhvXN

[7] J. M. Arevalillo. (2013). Churn. Segmentación CHAID. Retrieved from https://bit.ly/3awGVj2

[8] J. M. Arevalillo. (2013). Churn. Segmentación CART. Retrieved from https://bit.ly/3awGVj2

[9] B. López Takeyas (2005), Inteligencia Artificial. https://bit.ly/30rJor5.