# Machine Learning model to predict success in the Saber Pro exam according to the Saber 11 exam

Julián Mazo
Universidad EAFIT
Medellín, Colombia
jamazoz@eafit.edu.co

Cristian Zapata
Universidad EAFIT
Medellín, Colombia
cczapatag@eafit.edu.co

Harold González
Universidad EAFIT
Medellín, Colombia
hsgonzaleo@eafit.edu.co

Gloria Sepúlveda
Universidad EAFIT
Medellín, Colombia
gsepulv1@eafit.edu.co

## ABSTRACT

The objective of this report is to perform an analysis of the results of the ICFES tests, seeking to determine what are the factors that influence the success of a student in the future. In this way, alternatives can be proposed to improve the quality of life of citizens, focusing on those characteristics that need to be improved.

To perform the analysis, we created a program that builds a classification tree using data sets with the results of the Saber 11 exam, so that using the results of previous Saber Pro tests, we can review which characteristics of the students helped them to be successful in the future.

## Keywords

Data Structure; Binary Tree; CART algorithm; Complexity Analysis; Data Analysis; ID3 algorithm; Machine Learning; Predictions

## 1. INTRODUCTION

The outlook for Colombia's education system has not been good in recent years. According to the World Bank, the dropout rate for students is 42%, placing the country as the second in Latin America with the highest dropout rate. Therefore, it is necessary to take action on the matter so that it is necessary to find what affects the quality of education. For this, several studies on education are carried out in the country, one of them being the influence of the Saber 11 tests on the success of students for their future. This will be the main subject of study for the article.

## 2. PROBLEM

Throughout the paper, the characteristics that can affect the success of a student will be sought, according to the data collected by the ICFES. In this way, we will try to make an algorithm that can predict from the results of the Saber 11 test if the student will be successful in the Saber Pro test.

An analysis will be made of the different alternatives that can be used to create this algorithm, including the different types of classification trees that belong to supervised learning algorithms. We will work on a program that is as efficient as possible, using some of the investigated algorithms.

Finally, the certainty of the program will be evaluated and if it could serve as a model to make a decision regarding the country's educational system.

## 3. RELATED WORK

Next, some of the alternatives that can be used to carry out the program will be briefly described to predict the probability of success that a student will have.

### 3.1 CART Algorithm

The main algorithm that we are going to use to do the data analysis is the one called CART (Classification And Regression Trees) which uses the "Gini impurity" method (not Gini coefficient) to be able to learn based on decision trees.

- The classification trees predict categories of objects.
- The regression trees predict values continuous.

The CART algorithm generates binary decision trees (Which means that every node is divided in two branches), in each iteration the variable is selected predictive and the breakpoint that best reduce 'impurity'; it uses the Gini index to calculate the measure of impurity [4]:

$$G(A_i) = \sum_{j=i}^{M_i} p(A_{ij}) \cdot G(\frac{C}{A_{ij}})$$

Being $G(\frac{C}{A_{ij}})$ equals to:

$$G(\frac{C}{A_{ij}}) = -\sum_{j=i}^{M_i} p(\frac{C_k}{A_{ij}}) \cdot (1 - p(\frac{C_k}{A_{ij}}))$$

- $A_{ij}$ is the attribute use to branch the tree.

© 2020 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

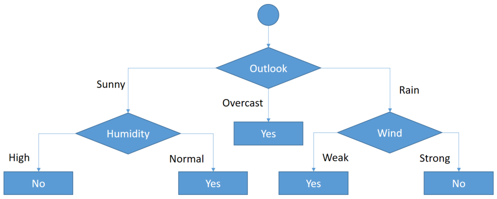


**Figure 1: [4] Final form of a decision tree built by CART algorithm**

- $j$ is the number of classes.
- $M_i$ are the different values that the attribute $A_i$ has.
- $p(A_{ij})$ constitutes the probability that $A$ takes its $j$-th value.
- $p(C_k/A_{ij})$ represents the probability that an example is of class $C_k$ when its attribute $A_i$ takes its $j$-th value.

## 3.2 CHAID Algorithm

Another choice that could be implemented instead of CART algorithm is CHAID tree decision algorithm. This one is the oldest decision tree algorithm in the history [5]. It uses the chi-square metric to find significance of a feature (A higher value means a higher significance). It works for classification problems, so it would be useful for this project.
The formula of chi-square testing is:

$$Chi = \sqrt{\frac{(actual - expected)^2}{expected}}$$

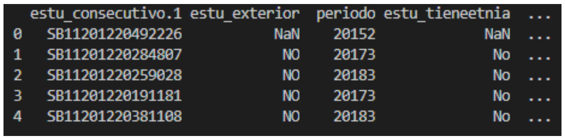
Where the actual value means times that a decision was taken, and the expected value means times that a decision is expected to be took. The formula of expected value for any statement is:

$$expected = \frac{decisions}{classes}$$

The root of the decision tree will be the statement that have the most chi-square value. Then, the data will be sorted and rearranged accord the specifications of the root, and the branches will be the statements with the most chi-square value of the new arrange of data. When a branch just arrives to a definitive decision (example, in the case that a statement just have a yes decision or a not decision), we stop making more branches for that branch (becoming a leaf).
When we done arriving to leaves, the tree will be finished and ready for make the data classification.

## 3.3 C4.5 Algorithm

Algorithm C4.5 is our third option, C4.5 is an extension of ID3, it is an algorithm applied to generate decision trees, it can be employed for classification, an interesting factor to take into account.C4.5 build trees using the concept of information entropy. At each node in the tree, C4.5 chooses an attribute of the data that most efficiently divides the set of samples into subsets enriched in one class or another. Its criterion is the normalized one for information gain (entropy difference) that results in the selection of an attribute to divide the data [3].



```
  estu_consecutivo.1 estu_exterior  periodo estu_tieneetnia  ...
0   SB11201220492226           NaN    20152             NaN  ...
1   SB11201220284807            NO    20173              No  ...
2   SB11201220259028            NO    20183              No  ...
3   SB11201220191181            NO    20173              No  ...
4   SB11201220381108            NO    20183              No  ...
```

**Figure 2: A DataFrame made with a dataset**

The attribute with the highest normalized information gain is preferred as the decision parameter. The C4.5 algorithm recursively divides into smaller sub lists. This algorithm has a few base cases which are:

- All the samples in the list belong to the same class. When this happens, you just create a leaf node for the decision tree by saying to choose that class.
- Neither feature provides any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class
- Previously unseen class instance found. Again, C4.5 creates a decision node higher up the tree with the expected value.

# 4. MATRICES

For the implementation of the algorithm, the first thing we must do is read the data sets in order to create the classification tree. To do this, we will use the pandas library that can read a .csv file to convert it into an object of type DataFrame. However, handling pandas DataFrames is not very easy and it can be quite tedious to work with them when creating trees, so once the data is read, the DataFrame will be converted to a type list using the tolist() method of pandas and it will contain lists representing the rows of the DataFrame. Besides, a list with the labels of the DataFrame will be created to facilitate the creation of the tree.

## 4.1 Complexity Analysis

The advantage of using matrices is that accessing any of its elements only has a complexity of $O(1)$, so it makes the work too easy when creating the classification tree. Due to the amount of data that each set has, the creation of the matrices will have a complexity of $O(n \cdot m)$ where n is the number of rows in the matrix and m is the number of columns in the matrix. This is because adding each item costs an elementary operation, so a total of operations equal to the amount of data in the set is required.

## 4.2 Time Analysis

The Table 1 contains the execution times of the dataset reading algorithm for five different datasets.

Table 1: Execution time of the dataset reading algorithm

| Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Dataset 5 |
|---|---|---|---|---|
| 15,000 | 45,000 | 75,000 | 105,000 | 135,000 |
| 0.524 s | 3.083 s | 4.351 s | 3.674 s | 5.309 s |

Table 2: Execution time of the tree creating and data classifying algorithms

| Recursion depth | 3 | 5 | 7 | 9 | 11 |
|---|---|---|---|---|---|
| Creation | 224 s | 373 s | 509 s | 643 s | 782 s |
| Classify | 0.84 s | 1.12 s | 1.61 s | 1.74 s | 1.79 s |

We can say that the algorithm takes little time when it uses a large amount of data, so we conclude that the reading of the data is optimal.

# 5. TREE IMPLEMENTATION

To make predictions of student success, we will use classification trees, using the CART algorithm [1]. To implement the trees, we create a series of methods that, using a set of data, will look for the variables that give us the most information, building the tree so that when classifying the data to evaluate it, you have good certainty.

We decided to use the CART algorithm as it is one of the simplest to understand and implement, and for this type of problem, it is usually very effective when making predictions.

## 5.1 Complexity Analysis

The creation and training of the tree requires methods with complexity from $O(1)$ to $O(n^2)$, but by itself its complexity is $O(2^n)$ because it makes use of recursion to create two subtrees per each node, which require the same methods [2]. Of course, we speak in the case that to get to the sheets it is required to divide each piece of data in the data set separately, which is unlikely to happen. Hopefully it will take time to create each tree.

To carry out the tree evaluations, only a complexity of $O(n)$ is necessary, since each piece of data goes through the tree a certain number of times, so it is capable of making the predictions really fast.

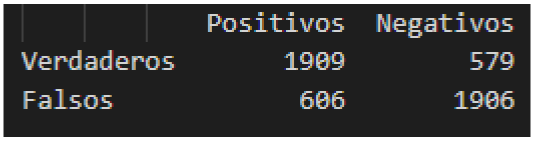
## 5.2 Time Analysis

The Table 2 contains the execution times of the tree creation algorithm using the dataset 5 and the classification of the testing data for five different recursion levels.

We check that our conclusions about the complexity of the algorithm are correct.

## 5.3 Certainty Analysis

To evaluate the certainty of our algorithm, we implement a method that creates an error matrix to determine what percentage of hits it has and how this certainty can be improved. Table 3 shows the certainty of the algorithm for three different recursion levels using datasets 1, 3 and 5.



```
           Positivos  Negativos
Verdaderos      1909        579
Falsos           606       1906
```

Figure 3: Error matrix of a tree built with the dataset 1

Table 3: Certainty of some trees created by the tree creation algorithm

| Dataset | Depth 3 | Depth 7 | Depth 11 |
|---|---|---|---|
| 1 | 76% | 78% | 76% |
| 3 | 75% | 78.6% | 78.4% |
| 5 | 75.31% | 78.54% | 75.31% |

We can see that the algorithm tends to have a certainty of 78% with a recursion level of 7. Thus, we can say that the algorithm has a fairly good certainty, in addition to that in the other cases it has a certainty greater than 75%

# 6. CONCLUSIONS

Having finished the analysis of the algorithm that we created, we concluded that it can be useful to predict how well a student will do in the future based on their results in the Saber 11 test. Although, the current situation of the educational system is not the best Carrying out this type of study is good to start working on solutions that allow us to advance towards progress.

We have noticed that the created program is efficient as an alternative to the initial problem, in addition to having a certainty of 78% on average, we can conclude that the CART tree was one of the best bets we could have made for the development of the project.

Most of the trees we build make their predictions based on the scores of each individual test, so we can say that it is an influential factor in the success of the tests. Also, where each student resides can make a big difference as well.

Finally, we can say that it is still a simple model of reality and that it will not always be accurate, but due to its high degree of certainty, it can be used for various purposes such as giving a greater focus to areas with a low probability of success. or provide good opportunities to young people with good results.

# 7. ACKNOWLEDGMENTS

We give credit to EAFIT University for offering the course on Data Structure and Algorithms where we learned what is necessary to carry out the project. We also thank Luisa Toro Villegas, who helped us with support material in which we guided ourselves for the creation of the algorithm.

# 8. REFERENCES

[1] J. Gordon. Let's write a decision tree classifier from scratch - machine learning recipes 8, 2017.

[2] R. C. T. Lee, S. S. Tseng, R. C. Chang, and Y. T. Tsai. *Introducción al diseño y análisis de algoritmos.* McGraw-Hill Education, 2007.

[3] S. I. Serengil. A step by step c4.5 decision tree example, 2018.

[4] S. I. Serengil. A step by step cart decision tree example, 2018.

[5] S. I. Serengil. A step by step chaid decision tree example, 2018.