Utilizing decision trees to predict student performance on the test "Saber Pro"

Santiago Gonzalez Universidad Eafit Colombia sgonzalez6@eafit.edu.co	Mariana Vasquez Escobar Universidad Eafit Colombia mvasqueze@eafit.edu.co	Miguel Correa Universidad Eafit Colombia macorream@eafit.edu.co	Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co

ABSTRACT

The objective of this report is to analyze and predict student performance during the test Saber Pro through analyzing different pieces of important data collected directly from the students. These pieces of data are things that could potentially affect a student's performance in a test, this can be from how many books they have read to what is the state of the place where they live.

This is important for giving students a slight insight into how well they could fare when taking the test, this also helps the teachers note which students need more help than others when facing the test.

There have been several different reports that have similar purposes to this one, all of which have helpful information on the matter at hand, these will be discussed further in chapter 2 of this report.

We used a CART algorithm, in which we got around 50-60% accuracy rate, the algorithm takes some time training the forest, as this is a process that is made with a total of 135000

Keywords

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

1. INTRODUCTION

This project has a primary motivation of making the results of the test Saber Pro better, thinking of the test as a show of the academic success of a higher education and making an analysis before an unfavorable situation for the test mentioned realized in 2019; the results should be projected not only on the test, but in the betterment of aptitude and proficiency of the Colombian professionals.

1.1. Problem

We try to identify individuals that are prone to obtaining an unfavorable score at the tests and help them find the necessary help in order to surpass patterns, factors and possible difficulties that could impede the academic success of a higher education.

1.2 Solution

In this work, we focused on decision trees because they provide great explainability, algo característico de los métodos White-box, que si bien son menos exactos que los black-box, funcionan de una forma más intuitiva y similar a la forma humana de razonar, resultando en modelos fáciles de interpretar [9], therefore We avoid black-box methods such as neural networks, support-vector machines and random forests because they lack explainability. [10]

For being able to identify the patterns and variables that influence the performance of each student, it was decided that the decision tree that will be used is the CART, which is not only easy to explain and interpret (Something that was mentioned previously) but it also has a good handling of numeric data and its structure tends to be non-variable[11].

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

These next entries were taken from the related problems tab in the GitHub, as this seemed like the best option for getting reports that relate almost directly to our problem.

2.1 Decision trees for predicting the academic success of the students.

This study used multiple types of algorithms for its testing, this report showed that some of the most accurate algorithms are REPTree and J4.8, they picked between the following algorithms for the study:

ID3, which is the most primitive of the ones listed, as is doesn't backtrack to check for possible errors or inconsistencies.

C4.5, which was created to surpass ID3 by surpassing the limitations the previous one had. This one measured the number of outcomes by using a gain ratio.

J4.8, described as the same algorithm for C4.5, but it uses the WEKA toolkit for the creation of the trees.

REPTree (Reduced Error Pruning Tree), this one is optimized for speed and uses a gain/variance reduction to analyze the information that is given to it. This algorithm creates a variety of different trees, before discarding all of them but the best one.

RandomTree, this one considers several random features that are given to it in the data, however, unlike the previous one, this one doesn't prune anything that is unnecessary.

RandomForest, it has the same functionality as the RandomTree, the main difference being that this one creates multiple trees and uses a random vector for independent sampling and distribution for all the trees that are created.

The data collected for this study was taken in the span of 3 years and took the following into account:

-Student data (age, gender, housing, etc...)

-High School and other completed programs.
-Average High School grades.
-Grades from the State exam and individual grades in a variety of subjects.
-The importance that was given to the enrolled faculty by the students.
The algorithms that yielded the best results where REPTree (79.35% accuracy) and J4.8 (73.76% accuracy), with J4.8 having more accuracy in calculating the high average students compared to REPTree.
Taken from: GitHub mauriciotoro/ST0245-EAFIT
https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/Decision%20trees%20for%20predicting%20the%20academic%20success%20of%0Astudents.pdf

2.2 Mining Students data using decision trees.

This study had the purpose to improve the quality of the higher education system by analyzing students' habits and performance when writing code (note that the data from this study was taken from students that entered a C++ course in Yarmouk university).
The methods used in this study where:
ID3, C4.5, Naïve Bayes
The data collected seemed to not be enough for the study, as the accuracy rate was low for all the trees that were created.
Taken from: GitHub mauriciotoro/ST0245-EAFIT
https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/MiningStudentDataUsingDecisionTrees.pdf

2.3 Predicting Students performance using ID3 and C4.5.

This study had the same purpose as the previous two: analyze student data to determine, in a way, their performance, this study had a different approach as to what they used in the creation of the decision trees. Although the used ID3 and C4.5, which were both mentioned in the previous reports, in this one they also mentioned the specific ways they used these:
HTML & CSS: Used to create content that is visible from the web, used to create semantic structures as: headers, paragraphs, lists, quotes, etc...
PHP & CodeIgniter: Scripting language directed towards the server, PHP is used for hypertext, just as HTML. CodeIgniter was used for this study.
MySQL: Data base generator. Used to store the data used in the study.
RapidMiner: Data mining tool used to gather the data for the decision trees.
After gathering and pruning the unnecessary data, the decision trees were created with both the training data and the real data, before creating a web interface for these trees to be seen.
Taken from: GitHub mauriciotoro/ST0245-EAFIT
https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/PREDICTING%20STUDENTS%E2%80%99%20PERFORMANCE%20USING%20ID3%20%26%20C4.5.pdf

2.4 Predicting Students final passing results using the CART algorithm.

This study had the main purpose of predicting the final grade for students that are about to finish their career, the data that was used for this was their performance in the previous semesters. As the title mentions, the algorithm used in this one was the CART algorithm, this algorithm is similar to ID3´s, the main difference being that ID3 uses concepts of entropy for its functionality, while CART uses a Gini index to search for impurities.
The conclusion compared the study made to a different study that had the same goal, it also mentioned that further similar research was on the horizon.
Taken from: GitHub mauriciotoro/ST0245-EAFIT
https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/Predicting%20students%E2%80%99%20final%20passing%20results%20(CART)%20algorithm.pdf

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 undersampling 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

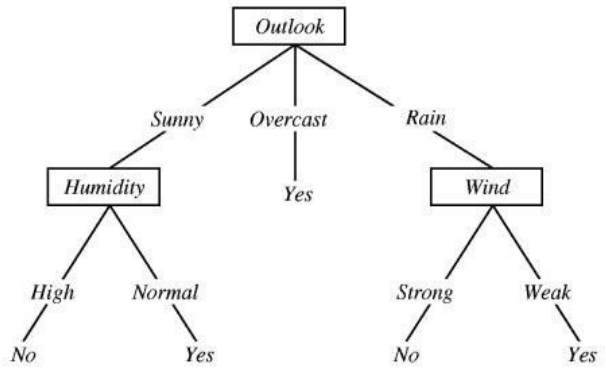
The following are decision tree algorithm alternatives.

3.2.1 ID3

It bases itself off the concept of entropy and information gain to better define the classification of a data set according to their attributes [3]:

- **Entropy**: Determines the purity of the classification according to how little entropy it has, this means that a tree with 0 entropy is perfectly classified.
- **Data gain:** Defined as the difference between the entropy of the original data set and the sum of all the entropy in each one of the data subsets.

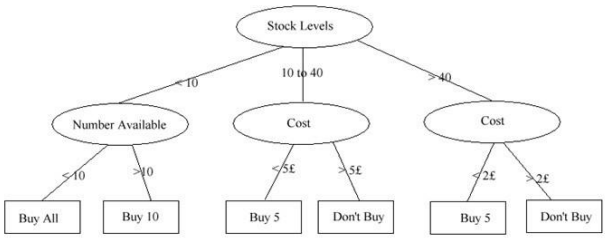
The attribute with the most gain is stablished as the *splitting attribute* [3] of a node, in a way that there is less entropy for the data that isn´t classified yet. This process is done in a recursive manner, dividing the main node in branches y smaller nodes that contain lesser amounts of data until all the elements are classified.



Taken from: GitHub mauriciotoro/ST0245-EAFIT https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/PREDICTING%20STUDENTS%E2%80%99%20PERFORMANCE%20USING%20ID3%20%26%20C4.5.pdf

3.2.2 C4.5

An extension of ID3, it was created to correct some of the possible mistakes ID3 could possess. By allowing the handling of numeric (discrete and other types) and categoric data it avoids over-classifying the data, correctly administers incomplete data and allows to create a hierarchy for the data. Similar to ID3, it divides the data set from the attribute that has the most information gain y proceeds to do the same process for the smaller pieces of data [5].



Ejemplo de algoritmo C4.5

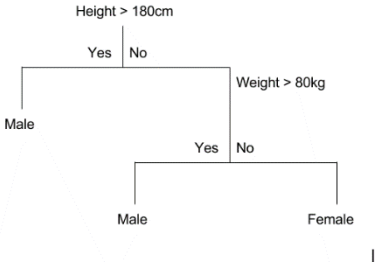
3.2.3 REPTree

The REPTree (Reduced Error Pruning Tree) algorithm is based on the C4.5, it makes different iterations of the trees and selects the best one out of the trees it generated. In a similar way as C4.5, it bases its accuracy rating based on impurity ratings and information gain [6].

3.2.4 CART

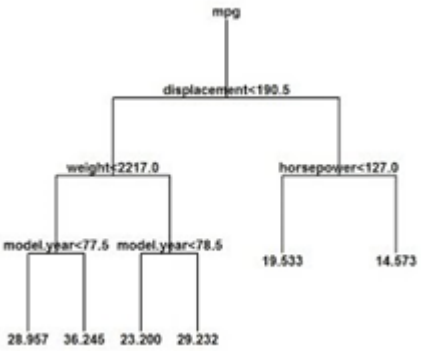
The CART (Classification and Regression Trees) algorithm produces simple binary trees and classifies them in classification and regression trees.[7]:

- **Classification trees**: These are used to determine the category an attribute belongs in.



Classification tree created with CART

- **Regression trees**: Used to predict the value of an attribute and its characteristics.



Regression tree created with CART

The same way as the previously mentioned algorithms, the CART utilizes measures of impurity such as Gini or entropy, it considers the attribute that splits the data set in the most homogenous way for each one of the nodes, and the entire process is made in a recursive manner.

3.2.5 RandomForest

It consists on the creation of a group of trees created via the same method, but providing them with a different data set (all variations of the original set). Similar to the CART algorithm, this one is useful for both the classification and the data regression When it uses some sort of classification, each tree throws as a result a class in which to sort the element to analyze, being the final result the most voted one by the forest. When it applies regression, it makes an average of the result set created by each tree [8].

4. ALGORITHM DESIGN AND IMPLEMENTATION

The following is an explanation on how the data structures and the implemented algorithm work. These will be available at: https://github.com/SantiagoGonzalezR/ST0245-002-/tree/master/proyecto [1].

4.1 Data Structure

The designated data structure that was used for this project was the binary tree known as CART (short for Classification and Regression Trees) The following image is an example of a short binary CART tree.

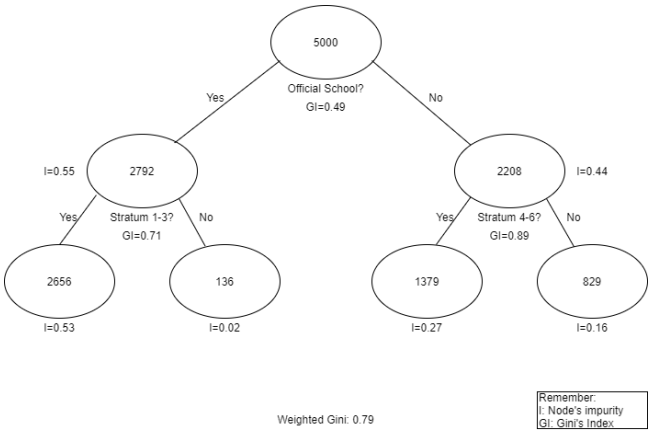


Figure 1: This is a small fragment of the tree that will be used to predict student success in the Saber Pro 11. This is not final, as the impurities are rather high, but the end product shall be much more accurate.

4.2 Algorithms

The algorithm can store the information of up to 125000 students (this could be changed in the future due to this not being user friendly in case of a bigger data set), the algorithm will then start organizing the values according to different criteria (as seen in figure 1), afterward it calculates the impurities of all the nodes that are created to check the accuracy of the binary tree (the lower the Gini impurity, the better).

4.2.1 Training the model

The algorithm will split the data by using a designated value that is given a specific value, say Stratum=3; School=Official; Salary=4SMLV; etc...

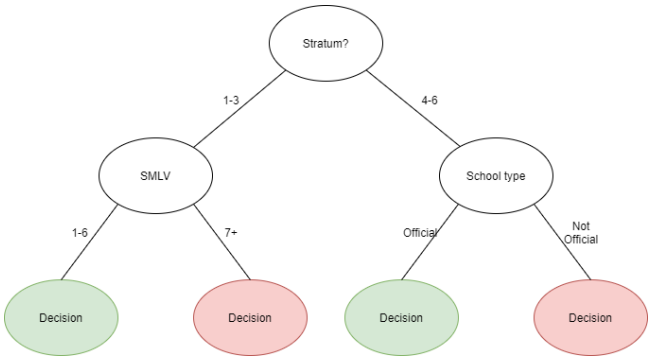


Figure 2: This is the training model for the CART tree that will be used in this project. In this example, we show a model to make decisions based on the selected information, the tree is not capable of making predictions as of the writing of this example, this is due to the algorithm not being fully implemented, the understanding of the reader is appreciated.

4.2.2 Testing algorithm

Each time a new set of nodes is created, the algorithm uses a new criterion for the creation of the next set of nodes, this is so the tree can be as specific and precise as it can be. The

decision that is made is based on how homogenous the new set of nodes will end up being.

4.3 Complexity analysis of the algorithms

This analysis was made after abstracting the complexity for each method and identifying the data structures that was used for each one. Big O notation rules we saw in class were used for the complexity in time, with these we arrived at an approximation of the complexity of the testing and the training trees. Complexity in time on the other hand required some analysis and research into how the data structures store information, this was so we could prioritize those that were present in the code and where more demanding memory-wise.

Algorithm	Time Complexity
Train the decision tree	$O(N^2+M)$
Test the decision tree	$O(N*M+4^{1-N})$

Table 2: Time Complexity of the training and testing algorithms. In the previous information table, N is the number of rows that the initial matrix had, which is equivalent to the number of students that are in the data set. Due to the differences in the methods of the code, M in the training is the amount of data that is stored in the nodes, and the number of nodes in the testing phase.

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

Table 3: Memory Complexity of the training and testing algorithms. *In this table N is the number of students and M for number of nodes in the training phase.*

4.4 Design criteria of the algorithm

We considered the CART algorithm would be the most optimal based on the way the recursion is made, as it shall be made apparent by the following paragraphs and as it was shown in the previous tables. For us, something that was remarkable about this decision tree was its execution time, although matrices do use a considerable amount of memory compared to other data structures, we opted for these because of their ease of comprehension and the reduced criteria that were left after pruning unnecessary data from the data sets, which made the amount of data optimal for storing is said matrices, resulting not only on decent time, but an easy-to-understand code.

5. RESULTS

5.1 Model evaluation

In this section, we present some metrics to evaluate the model. Accuracy is the ratio 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 3***
Accuracy	0.50	0.55	0.48
Precision	0.49	0.52	0.50
Recall	0.50	0.53	0.49

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 3***
Accuracy	0.5	0.55	0.48
Precision	0.5	0.50	0.52
Recall	0.5	0.52	0.51

Table 4. Model evaluation on the test datasets.

5.2 Execution times

	Dataset 1	***Dataset 2***	***...Dataset n***
Training time	66.2 s	100.4 s	192.4 s
Testing time	1.7 s	2.0s	20.8 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 (Metaspace)	8MB	11MB	17MB
Memory consumption (Metaspace)	228MB	247MB	300MB

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

6. DISCUSSION OF THE RESULTS

The results are not necessarily the best that they could be, but this doesn't mean it's a bad starting point, although, this is not a good model for determining who to give scholarships as it will almost always be a 50% chance for the student to either fail or succeed.

6.1 Future work

The algorithm could have its efficiency improved by a lot, as of right now it averages 50% accuracy and it takes around 3 and a half minutes to operate with a training dataset of 135000 and a test dataset of 45000, this isn't great, but it could always be worse.

ACKNOWLEDGEMENTS

We are grateful for Andres Echeverri's and Esteban Echeverri's assistance in developing the code, this saved a lot of time and extra effort that we couldn't afford.

REFERENCES

1. Decision trees for predicting the academic success of students. Retrieved August 12, 2020, from Mauricio Toro´s GitHub: https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/MiningStudentDataUsingDecisionTrees.pdf

2. Mining Student Data Using Decision Trees. Retrieved August 13, 2020, form Mauricio Toro´s GitHub: https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relacionados/MiningStudentDataUsingDecisionTrees.pdf
3. PREDICTING STUDENT PERFORMANCE USING ID3 AND C4.5. Retrieved August , 2020, from Mauricio Toro´s GitHub: https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relaciona-dos/PREDICTING%20STUDENTS%E2%80%99%20PERFORMANCE%20USING%20ID3%20%26%20C4.5.pdf
4. Predicting students´ final passing result (CART) algorithm. Retrieved August , 2020, from Mauricio Toro´s GitHub: https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/problemas-relaciona-dos/Predicting%20students%E2%80%99%20final%20passing%20results%20(CART)%20algorithm.pdf
5. Quora: What are the differences between ID3, C4.5 and CART? Retrieved August 15, 2020, from https://www.quora.com/What-are-the-differences-between-ID3-C4-5-and-CART
6. Sushilkumar Kalmeg. 2015. IJISET 2, 2 (Feb. 2015), 438-446. Analysis of WEKA Data Mining Algorithm REPTree, Simple Cart and RandomTree for Classification of Indian News.
7. Anukrati Mehta. 2019. Beginner's Guide to Classification and Regression. (January 2019). Retrieved August 15, 2020 from https://www.digitalvidya.com/blog/classification-and-regression-trees/
8. Augmented Startups. 2019. Random Forest – Fun and Easy Machine Learning. Video. Retrieved August 15, 2020 from https://www.youtube.com/watch?v=D_2LkhMJcfY
9. Grant Holtes. 2018. Decision Trees – Understanding Explainable AI. (March 2018). Retrieved October 1, 2020 from https://towardsdatascience.com/decision-trees-understanding-explainable-ai-620fc37e598d
10. Lars Hurstaelt. 2019. Black-box vs. white-box models. (March 2019). Retrieved October 1, 2020 from https://towardsdatascience.com/machine-learning-interpretability-techniques-662c723454f3
11. Charris L., Henriquez C., Hernández S., Jimeno L., Guillen O. and Moreno S. 2018. Comparative analysis of algorithms of decision trees in the processing of biological data. Universidad Simón Bolívar, Barranquilla, Colombia.