ALGORITHM TO PREDICT ACADEMIC SUCCESS IN HIGHER EDUCATION

Manuela Moreno Cordoba
Universidad Eafit
Colombia
mmorenoc2@eafit.edu.co

Yhilmar Andres Chaverra Castaño
Universidad Eafit
Colombia
yachaverrc@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

ABSTRACT

The purpose of this project is to analyze and solve the problem of predicting the success of a higher education student in Colombia. One of the key factors to execute this project is to be able to find and use an algorithm based on decision trees, which allows me to collect data and that in turn provides the tendency that a student can have for success through a certain amount of data.

This problem is of the utmost importance since the solution allows us to know which Colombian population could obtain success in higher education, making a better analysis of the country's growth and progress knowing this data. There are many cases like this one, which will be explained as this document is read

Keywords:
prediction
academic success
Data structures
algorithms
decision trees

ACM CLASSIFICATION Keywords

- Information systems > Storage and retrieval of information> Search and retrieval of information > Grouping
- Consultation Formulation > Recovery models > Search process > Selection process

1. INTRODUCTION

Education is one of the factors that most influences the progress of people and societies around the world, the quality of education can determine important areas in a society such as economic growth, access to better employment opportunities, improve the condition of life, the advancement of science, technology, and innovation.

Education can influence the success of a person in an influential way and Colombia, the digital era will revolutionize the area of education, so it is important to consider the Colombian population that may or may not succeed in education higher. However, there are very few tests and solutions on how to predict the success of a student in the academic field, and since success can be defined in multiple ways, we will define academic success as a student's tendency to obtain a Total score, above the average of your cohort, in the Saber Protests. The Saber Protests are the standardized tests that the Colombian government performs at the end of a university degree.

2. PROBLEM

The problem is that we must design an algorithm, based on decision trees and ICFES data, to predict whether a student will have a total score, on the Saber Protests, above average or not. The variables required for the algorithm must also be considered since it must have the flexibility to interpret data even if it does not respond to all the necessary data.

3. RELATED WORK

To understand the solution, 4 algorithms are provided here that can help to solve the problem

3.1 ID3:

ID3 (Iterative Dichoiser 3) was developed in 1986 by Ross Quinlan. It builds a decision tree for the given data in a top-down fashion, starting from a set of objects and a specification of properties Resources and Information. each node of the tree, one property is tested based on maximizing information gain and minimizing entropy, and the results are used to split the object set. This process is recursively done until the set-in a given sub-tree is homogeneous (i.e. it contains objects belonging to the same category). The ID3 algorithm uses a greedy search. It selects a test using the information gain criterion, and then never explores the possibility of alternate choices You should mention the first algorithmic problem and its solution.

For this algorithm to work, it requires:

- Data structure (Tree)
- Searching algorithms (greedy algorithm, heuristic search, hill-climbing, alpha-beta pruning)
- Logic (OR, AND rules)
- Probability (Dependent and Independent)
- Info

3.2 C4.5:

C4.5 is based on ID3, therefore, the main structure of both methods is the same.

C4.5 builds a decision tree using the "divide and conquer" algorithm and evaluate the information in each case using the criteria of Entropy, Profit or profit ratio.

C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. This accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it.

3.3 C5:

This node uses the C5.0 algorithm to build either a decision tree or a rule set. A C5.0 model works by splitting the sample based on the field that provides the maximum information gain. Each subsample defined by the first split is then split again, usually based on a different field, and the process repeats until the subsamples cannot be split any further. Finally, the lowest-level splits are reexamined, and those that do not contribute significantly to the value of the model are removed or pruned.

C5.0 can produce two kinds of models. A decision tree is a straightforward description of the splits found by the algorithm. Each terminal (or "leaf") node describes a subset of the training data, and each case in the training data belongs to exactly one terminal node in the tree. In other words, exactly one prediction is possible for any data record presented to a decision tree.

3.4 CART:

The CART algorithm is structured as a sequence of questions, the answers to which determine the next question if any should be. The result of these questions is a tree-like structure where the ends are terminal nodes at which point there are no more questions. A simple example of a decision tree is as follows

The main elements of CART (and any decision tree algorithm) are:

Rules for splitting data at a node based on the value of one variable;

Stopping rules for deciding when a branch is terminal and can be split no more; and

Finally, a prediction for the target variable in each terminal node.

4. Array of arrays:

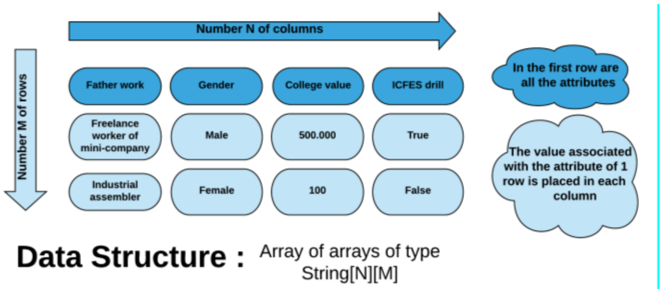

Number N of columns
Number M of rows
Father work
Gender
College value
ICFES drill
Freelance worker of mini-company
Male
500.000
True
Industrial assembler
Female
100
False
In the first row are all the attributes
The value associated with the attribute of 1 row is placed in each column
Data Structure :
Array of arrays of type String[N][M]

Figure 1: Array of arrays of type String with a number N of columns and number M of rows. In the first row are all the attributes and the value associated with the attribute of 1 row is placed in each column.

4.1 Operations of the data structure

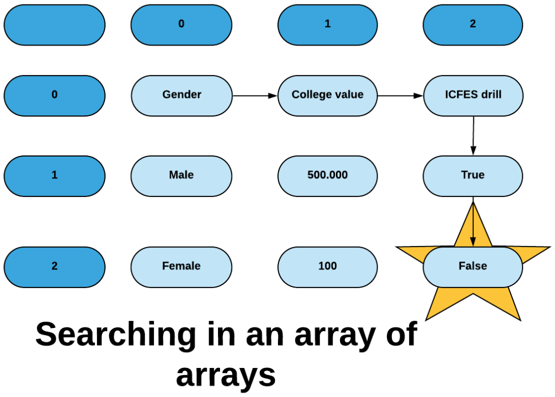


Figure 2: Search for data in an array of arrays

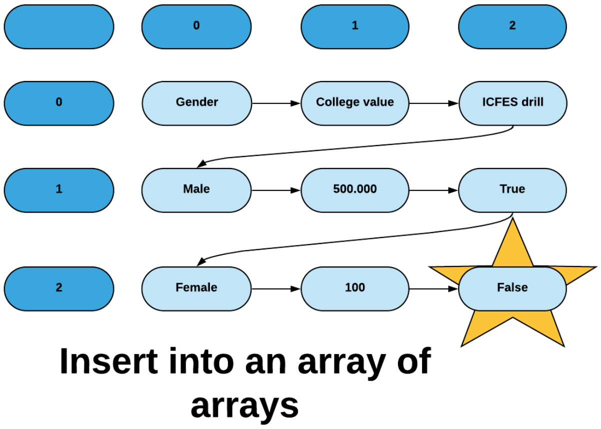
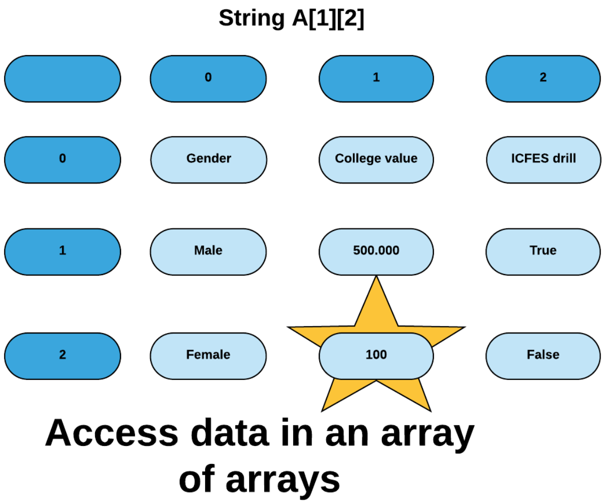


Figure 3: Insert data into an array of arrays



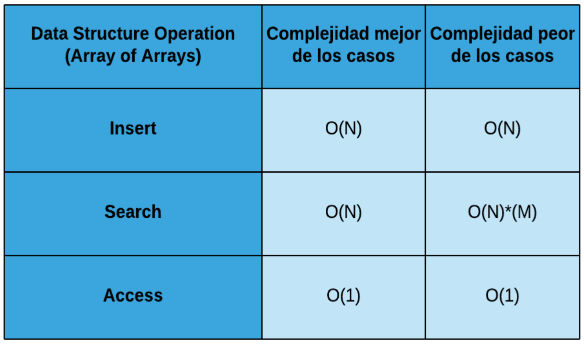
	0	1	2
0	Gender	College value	ICFES drill
1	Male	500.000	True
2	Female	100	False

Figure 4: Access data in an array of arrays

4.2 Design criteria of the data structure

When deciding what data structure to use for the solution, the first thing we could analyze is that we needed to save the data that is read from the CSV file, our first idea was to save the data in an ArrayList, where we had a person class, however after calculating the complexity of the algorithm, we see that adding an ArrayList is of complexity o (n) and if we add N people the algorithm would be O (N) * (M), making the code less efficient. Instead if we use an array of arrays where the first row is the attributes and The value associated with the attribute of the first row is placed in each column, it would give us that the complexity of the algorithm is O (n) so it makes the code much more efficient

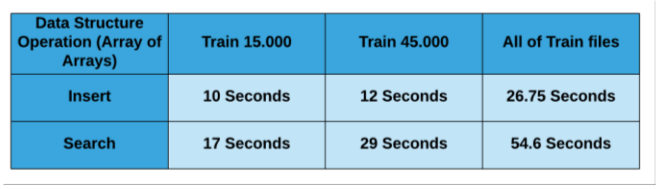
4.3 Complexity analysis



Data Structure Operation (Array of Arrays)	Complejidad mejor de los casos	Complejidad peor de los casos
Insert	O(N)	O(N)
Search	O(N)	O(N)*(M)
Access	O(1)	O(1)

Table 1: Table to report complexity analysis

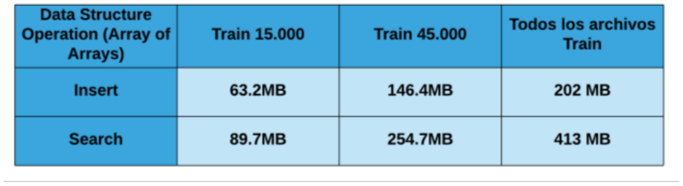
4.4 Execution time



Data Structure Operation (Array of Arrays)	Train 15.000	Train 45.000	All of Train files
Insert	10 Seconds	12 Seconds	26.75 Seconds
Search	17 Seconds	29 Seconds	54.6 Seconds

Table 2: Execution time of the operations of the data structure for each data set.

4.5 Memory used



Data Structure Operation (Array of Arrays)	Train 15.000	Train 45.000	Todos los archivos Train
Insert	63.2MB	146.4MB	202 MB
Search	89.7MB	254.7MB	413 MB

REFERENCES

Reference sourced using ACM reference format. Read ACM guidelines in http://bit.ly/2pZnE5g

1. What is the C4.5 algorithm and how does it work? - https://towardsdatascience.com/what-is-the-c4-5-algorithm-and-how-does-it-work-2b971a9e7db0

2. C5.0 Node - https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/c50node_general.htm

3. Introduction to Classification & Regression Trees (CART) - https://www.datasciencecentral.com/profiles/blogs/introduction-to-classification-regression-trees-cart

4. Common Data Structure Operations - https://www.bigocheatsheet.com/