USING MACHINE LEARNING AS A FORM OF SUCCESS PREDICCCION

Gregorio Pérez Bernal
Universidad EAFIT
Colombia
gperezb1@eafit.edu.co

Luisa Toro Villegas
Universidad EAFIT
Colombia
ltorov@eafit.edu.co

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

ABSTRACT

To write the abstract, you should answer the following questions in a paragraph: What is the problem? Why is the problem important? Which are the related problems?

The primary objective of this report is to determine what influences the success rates of students who have studied in Colombia, analyzing the Saber 11, some socioeconomical aspects and comparing the Saber Pro results with the national average. The problem this faces is to find what influences success, but it also provides an insight in the countries education dynamics and predicts one's opportunities to be prosperous. The opposite situation to the one presented in this paper could be the desertion levels, in the sense that by detecting what makes someone outstanding, these factors can say what makes one unsuccessful.

Keywords

•Applied computing~Operations research~Decision analysis•Information systems~Information systems applications~Decision support systems•Theory of computation~Design and analysis of algorithms•Mathematics of computing~Discrete mathematics~Graph theory~Trees•Computing methodologies~Symbolic and algebraic manipulation~Symbolic and algebraic algorithms•Applied computing~Education•Computing methodologies~Artificial intelligence•Computing methodologies~Machine learning~Machine learning algorithms

1. INTRODUCTION

The Colombian education system is known to be a flawed one, having one of the highest superior education desertion rates in Latin America with a shocking 42% in 2018, according to el Banco Mundial (Casas Mogollón, 2018). One of the priorities the country ought to have is to determine the reasons why this is occurring and how to stop it, or at least to drastically reduce it. The present work will use the rising technology of machine learning, which is an artificial intelligence (AI) and as the name says it, it focuses on how a computer can learn from "experience", which in the context refers to data that had been previously classified and it uses information gain to learn the most of each factor.

Retrieved from: https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1
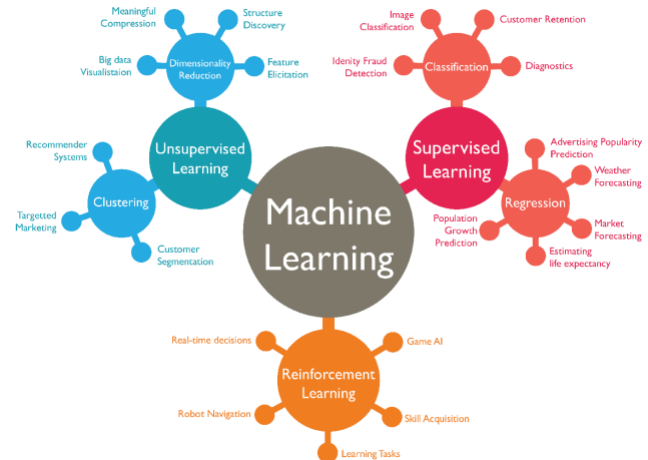
2. PROBLEM

During this paper, the authors will attempt to predict the level of success of a given sample of people who have gone through the Colombian education system by exploring how certain aspects, such as their socio-economic situation, their highest education degree, their access to technology, and their Saber 11 results, which is the exam that high school graduates take in Colombia in their senior year. The success will be measured by whether said person obtained a higher than average score in the Saber Pro, which is the exam that every university student must take in order to graduate, exam or not.

The following project is of vast importance to Colombian society now a days due to the fact that in order to keep growing as a country, it needs to focus on the educating the youth, which represents the work force of our future, and Colombia has had a distinct history of only educating properly a selected few: the wealthy. This is not sustainable in the long run, therefore, it's primordial to determine and detect what is failing in the education system and correct it immediately.

In order to approach this problem, the authors will use an algorithm called decision trees, that is a type of machine learning, not unlike logistic regression, artificial neural networks and random forests.

3. RELATED WORK

The following section will briefly explain some types of decision trees and explore their advantages and disadvantages. All these algorithms have to decide how to act on certain aspects like which attributes to use for splitting, in which order to split them, number of splits, pruning (which data is important and which is not), and the stopping criteria. (Bhatt, Mehta, & D'mello, 2016)
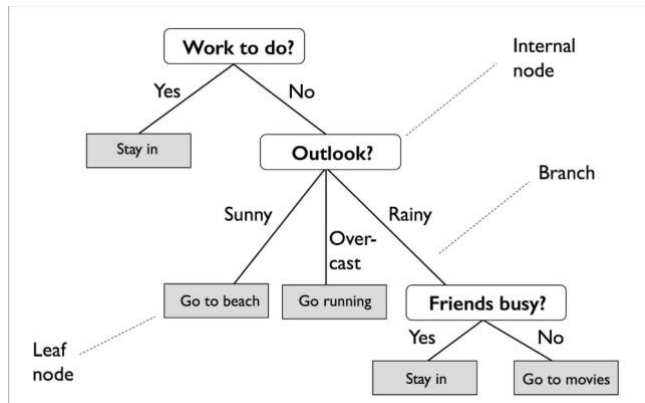
Moreover, for each type, a real-life example of how the algorithms have been used shall be given.

Before commencing, a few terms shall be introduced.
- Data mining:
- Entropy: in technology, Oxford defines entropy as "a way of measuring the lack of order that exists in a system." (Saha, 2018) In decision trees, entropy is a negative aspect, it decreases the certainty of the answer the algorithms provide us.
- Greedy splitting: "This is a numerical procedure where all the values are lined up and different split points are tried and tested using a cost function. The split with the best cost (lowest cost because we minimize cost) is selected" (Brownlee, 2016)
- Information gain: it "measures how much "information" a feature gives us about the class." (Sujan, 2018)
- Pruning: "Pruning refers to the removal of those branches in our decision tree which we feel do not contribute significantly to our decision process." (Saha, 2018)

3.1 ID 3 algorithm

The ID 3 (Iterative Dichotomiser 3) algorithm was first introduced by Ross Quinlan in 1975, which creates a decision tree from a given dataset. It offers the advantage that it is able to construct a small tree in a considerably short time. This tree is relatively quick due to the fact that while the data is being tested, the algorithm prunes it, reducing considerably the number of tests that are needed.

However, during the decision-making process, it only takes one attribute into account, which makes it hard to take into account problems of larger complexity. As well as that, implementing the ID3 algorithm with continuous variables may not work or may occupy a considerable amount of storage. This being said, the algorithm only works properly with discreet variables, which reduces the level of difficulty that this algorithm is able to solve.

The 1D3 algorithm was used in a study in 2015 in Mumbai University to predict campus placement, which refers to the hiring of college students before graduation, and this is evaluated based on their GPA, experience, programming skills and evaluation scores. The university used the study to help the performance of students in the future. (Bhatt, Mehta, & D'mello, 2016)

3.2 C4.5 algorithm

The C4.5 algorithm is a type of decision tree that was developed by Ross Quinlan, a pioneer in data mining. C4.5 is the successor of the ID 3 algorithm, and it brought two main improvements to the complexity of the problems that decision trees are able to solve. First, it includes the possibility of implementing discreet variables, which shaped the way in which problems were solved, for numeric values became available to be used. Moreover, this tree allows the existence of missing data to be implemented, meaning that whenever the value of something is unknown, the algorithm still works, guiding itself through the other bits of information presented.

On the other hand, C4.5 resolved the problem that the tree allows the existence of missing data to be implemented, however it backfired. Whilst this issue was cleared, as there are unknown variables that while the tree is being built, it constructs empty branches. Nodes with zero values or close to zero values have been encountered in multiple occasions. These values neither contribute to generate rules nor help to construct any class for classification task, thus, it only contributes by wasting valuable memory.

The C4.5 algorithm was used in a study about pregnancy risks and complications performed in 2016 by a group of professors which were published on the International Conference on Emerging Trends in Engineering, Science and Technology. The goal of the study was to standardize the information inquired during medical visits of women during pregnancy, giving priority to those who provide more information gain to the decision trees. It was shown by the algorithm that the most important factors were state, blood pressure, blood glucose level, weight, trimester and month. Then, they tried the algorithm on both standardized data and unstandardized data and showed that the prediction was more accurate for the first. (Lakshmi, Indumathi, & Ravi, 2016)

3.3 C 5.0 algorithm

The C5.0 algorithm was developed by Ross Quinlan, which was created to improve upon C4.5. When applied, it produces either a binary tree, that in which each node splits into two possible options, or multibranched tree, in which each node can split into any number of options. To determine into which factors it splits, it uses a method of measuring entropy which is information gain, and the goal is to maximize it, that way it takes better informed decisions. When presented with missing values or data, it roughly calculates its value basing itself upon the data of other variables.

This algorithm can be used to both create decision trees and rule-based models. Rule based models can use the structure of the tree to create mutually exclusive rules, it "assumes that the world can be understood as systems with interrelated parts that perform predictable behaviors." (Whitney, 2017)

An application to the C 5.0 algorithm is for mapping the locations of possible groundwater. The investigation was done in Iran and it is of special importance due to the fact that said country is extremely dry and water is needed in almost every aspect of daily life. A dataset of already known spring locations was inputted, and some of their variables like "altitude, slope aspect, slope angle, plan curvature, profile curvature, topographic wetness index (TWI), slope length, distance from rivers and faults, rivers and faults density, land use, and lithology" (Golkarian, Naghibi, Kalantar, & Pradhan, 2018) were used.

3.4 CART (Classification and Regression Trees)

The CART algorithm was first described by Leo Breiman. It is characterized by the fact that it constructs binary trees, namely each internal node has exactly two outgoing edges.

CART enables the usage of both numerical and categorical variables, which places an advantage towards ID3. As well as that, it uses greedy splitting to select the input variables that will be used. Something very beneficial about CART is the fact that it is able to determine which variables are the most useful, and immediately prioritize them, and the most useless and it will automatically delete them.

The problem with CART comes with the fact that the tree that is generated is highly unstable. A slight modification of the learning sample may change considerably the way in which the results are obtained. Besides, CART splits only by one variable, which reduces the number of problems that this tree can manage to solve.

A study conducted in Humboldt University in Berlin found the many Financial Applications that the CART decision tree can have. It was done by simulating a market and results show how a CART-based business application can perform on stock market as well as utilizing CART as a forecasting system. The conclusion reached was that simulated financial performance was above the average level when CART was applied to the Simulation. This means that by implementing the CART algorithm, one can actually perform better in the stock market and eventually becoming wealthier.

(Andriyashin, 2005)

4. An array of arrays (Matrix)

In order to solve the problem, the data structure that was implemented is an array of arrays (shown in Figure 1), in which each element of the bigger array corresponds to a person, and each element of the person's array corresponds to a characteristic of them, for instance, one's ICFES exam score, or city of residency. It is important to point out that the first array of the large array corresponds to the headers of each column.
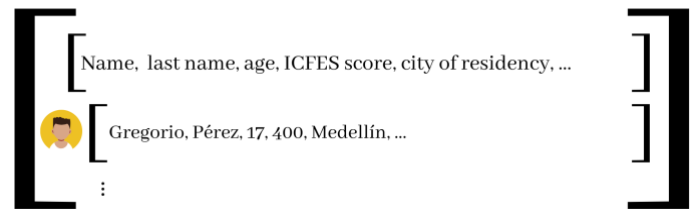

Figure 1: Matrix of persons. Each person is an individual array that contains a name, last name, age, ICFES score, city of residency and many other factors.

4.1 Operations of the data structure

For the data structure to solve the problem efficiently, the code will need certain operations to process the data given. The first and most crucial is the append operation, which will add the data read into the matrix (shown in Figure 2). The second is divided into two, access a specific element of a given person (Figure 3) and access a person in general (Figure 4). And the third operation is search (Figure 5), will when a certain factor is needed (e.g. name) it will search through the array [0], which is the array that will contain the column names and it will return the column number.
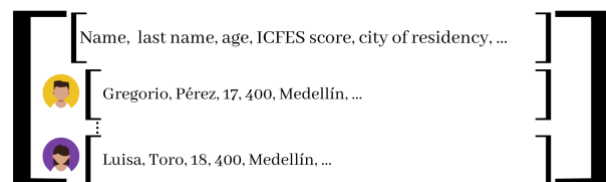

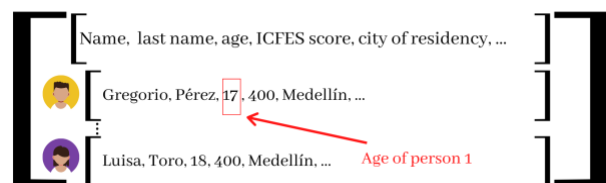**Figure 2:** Append operation of the matrix.
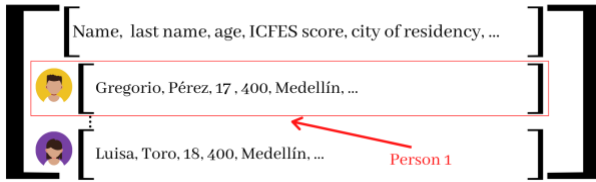

Figure 3: Access element operation of the matrix.

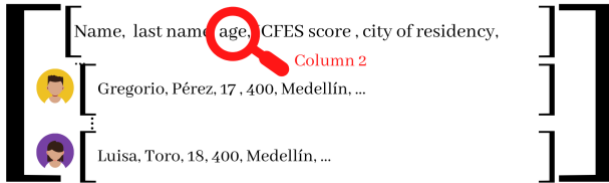**Figure 4:** Access person operation of the matrix.

Figure 5: Search for criteria operation in the array [0] of the matrix.

4.2 Design criteria of the data structure

In order to have an efficient algorithm, the main data structure used will be one in the form of a matrix. This was chosen because of two fundamental reasons. First, the complexity of accessing any element of it is $O(1)$, which is really beneficial. Another benefit is that accessing to information is extremely easy, due to the fact that everything is saved on a fixed position.

4.3 Complexity analysis

The main reason why this data structure was selected is the fact that the complexity to access a single person or a characteristic of a person was found to have a complexity of $O(1)$. The insertion method was found to be $O(m*n)$, n being the number of people and m being the number of characteristics per person. This complexity is not the best one but considering the fact that the algorithm only needs to insert one time, it is not that bad. As well as that, the method for deletion will not be taken into account, for the algorithm will examine which variables are useless and will ignore it instead of deleting them.

In order to search for an specific characteristic of a person, the idea of using a HashMap, whose key contains the position in which each attribute of a person is located. That means the "Search method would only have a complexity of $O(n)$, instead of $O(n*m)$.

In Table 1, these complexities will be displayed according to each operation.

Operations	Complexity of an array of arrays
Access	O(1)
Append	O(n*m)
Search	O(n)

Table 1: Table to report complexity analysis (m refers to the number of people, n refers to the number of characteristics per person).

4.4 Execution time

In this section, the times taken to execute each operation according to each dataset are measured and are shown in Table 2.

Operations	Data set 1 (5000) (s)	Data set 2 (15000) (s)	Data set 3 (45000) (s)	Data set 4 (75000) (s)
Append (create)	0.301	0.241	0.874	1.398
Access element	2.025e-05	9.418e-06	9.586e-06	7.956e-06
Access person	5.795e-06	1.990e-06	5.037e-06	3.967e-06
Search	1.704e-05	8.855e-06	1.352e-05	8.147e-06

Operations	Data set 5 (105000) (s)	Data set 6 (135000) (s)	Average (s)
Append (create)	1.866	2.817	1.2495
Access element	6.054e-06	1.077e-05	1.067e-05
Access person	1.608e-06	3.180e-06	3.596e-06
Search	1.408e-05	5.723e-06	1.123e-05

Table 2: Execution time of the operations of the data structure for each data set (worst case scenarios).

4.5 Memory used

In this section, the memory consumption of the data structure implemented will be measured and shown according to which data set was used (it also specifies the amount of lines in each data set). The data obtained is shown in Table 3.

	Data set 1 (5000)	Data set 2 (15000)	Data set 3 (45000)
Memory consumption	61 MB	68 MB	83 MB

	Data set 4 (55000)	Data set 5 (105000)	Data set 6 (135000)
Memory consumption	122 MB	232 MB	269 MB

Table 3: Memory used for each operation of the data structure and for each data set data sets.

4.6 Result analysis

The results obtained in the previous section are optimal. The only considerable time-consuming operation is the append operation, also called create operation, which is the one that inputs the data into the program, and taking into account the amount of data inputted, it's still good timing.

In comparison to other data structures, if queues, linked lists or stacks were used, in order to access to information, the complexity would be O(n), which presents a big disadvantage over arrays

In the next table, the times measured will be compared with the times that other data structures would take.

	Access	Search	Insert	Delete
Array	O(1)	O(n)	O(n)	O(n)
Linked List ()	O(n)	O(n)	O(1)	O(1)
Stack	O(n)	O(n)	O(1)	O(n)
Queue	O(n)	O(n)	O(n)	O(1)

Table 4: Analysis of the results

REFERENCES

Andriyashin, A. 2005. *Financial Applications of Classification and Regression Trees*. Humboldt University, Berlin.

Bhatt, H., Metha, S. and D'mello, L. 2016. *Use of ID3 Decision Tree Algorithm for Placement Prediction*. International Journal of Computer Science and Information Technologies, Mumbai.

Brownlee, J. 2020. Classification And Regression Trees for Machine Learning. *Machine Learning Mastery*. https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/.

Casas Mogollón, P. 2018. El problema no es solo plata: 42 % de los universitarios deserta | ELESPECTADOR.COM. *ELESPECTADOR.COM*. https://www.elespectador.com/noticias/educacion/el-problema-no-es-solo-plata-42-de-los-universitarios-deserta-articulo-827739.

GmbH, R. 2020. ID3 - RapidMiner Documentation. Docs.rapidminer.com. https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/id3.html.

Golkarian, A., Naghibi, S., Kalantar, B. and Pradhan, B. 2018. *Groundwater potential mapping using C5.0, random forest, and multivariate adaptive regression spline models in GIS*. Springer International Publishing.

IBM Knowledge Center. 2020. *Ibm.com*. https://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/c50node_general.htm.

Khadka, R. 2017. Machine Learning Types #2. *Medium*. https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1.

Kuhn, M. and Quilan, R. C5.0 Classification Models. Topepo.github.io. https://topepo.github.io/C5.0/articles/C5.0.html.

Lakshimi, B., Indumathi, T. and Ravi, N. 2016. *A study on C4.5 Decision Tree Classification Algorithm for Risk Predictions during Pregnancy*. International Conference on Emerging Trends in Engineering, Science and Technology, Karnataka, India.

Li, L. 2019. Classification and Regression Analysis with Decision Trees. *Medium*. https://towardsdatascience.com/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054.

Nguyen, A. Comparative Study of C5.0 and CART algorithms.

Rizvi, A. 2010. ID3 Algorithm. Sacramento, United States of America.

Saha, S. 2018. What is the C4.5 algorithm and how does it work?. *Medium*. https://towardsdatascience.com/what-is-the-c4-5-algorithm-and-how-does-it-work-2b971a9e7db0.

Sujan, N. 2018. What is Entropy and why Information gain matter in Decision Trees?. *Medium*. https://medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01.

Tree algorithms: ID3, C4.5, C5.0 and CART. 2019. *Medium*. https://medium.com/datadriveninvestor/tree-algorithms-id3-c4-5-c5-0-and-cart-413387342164.

Whitney, V. 2017. Rule Based Modeling. *Medium*. https://medium.com/data-mining-the-city/rule-based-modeling-203b8af9fbc7