DATA STRUCTURE FOR THE PREDICTION OF STUDENT RESULTS IN THE SABER PRO TEST

Miguel Valencia
Eafit University
Colombia
coberndorm@eafit.edu.co

Camillo Oberndorfer
Eafit University
Colombia
coberndorm@eafit.edu.co

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

ABSTRACT:

The objective of this paper is to analyze and to find a solution to the prediction of the Saber Pro marks which University Students across Colombia will achieve. Through the evaluation of different factors registered in the Saber 11 test. This hereby paper was motivated due to the necessity of seeing which factors will influence a student's professional training, thus education can be improved. Since this isn't the first-time student's data has been analyzed with such purposes, some of those other papers will be analyzed to provide a more accurate prediction.

Key Words: *Students Data, Saber Pro Test, Data Mining, Decision Tree.*

ACM CLASSIFICATION Keywords

Theory of computation → Analysis of Algorithms and Problem Complexity → Complexity Theory and Logic

Computing Methodologies → Artificial Intelligence → Problem Solving, Control Methods and Search

1. INTRODUCTION:

The national education and the academic quality of now-a-days professionals is a primary subject in almost every government, and in order to achieve a better educational system for the appearance of well-prepared professionals. It is necessary to figure out the multiple variables that affect the overall students' performance. That's why around the world other projects like this have taken place but most have been affected by the country's culture which can cause some alterations on the result of the predictions.

Taking into account the current Colombian environment and educational goals, we find ourselves in need of updating and creating an algorithm that predicts the professional success among Colombian population to be able to propose new educational goals as to improve the results according to the results of this.

2. PROBLEM:

The problem we face on this project is the creation of an algorithm capable of predicting accurate results on the Saber Pro test of the Colombian population according to some very concrete variables. Solving this problem would open a lot of opportunities of improvement on the Colombian educative system and could even be an example for other future similar projects based on machine learning and its real-world applications.

3. RELATED WORK

3.1 Random Forest

Proposed and first executed by Tin Kam Ho in 1995, a random forest is the implementation of many decision trees solving the same problem with the same variables, then an average of all the results is made, producing generally extremely accurate results. Using this system for our problem would mean that our solution will depend on how many decision trees work to get an answer.

3.2 Perceptron

Invented in 1958 at the Cornell Aeronautical Laboratory by Frank Rosenblatt, a perceptron is a type of supervised leaning, it is generally called a "neuron" which consists of four main parts, the inputs, the weights, the bias and the output. It works similarly to a human neuron. It is trained by generations each slightly modified, or "evolved" from the past one.

- First it is given some inputs.
- The inputs are then multiplied by their respective weights (a value given randomly and improved through generations, this way the importance of certain variables of decision are determined).
- The sum of the bias and the product of the weights and the inputs is calculated

$$\sum_{i=1}^{m} Wi * Xi + b$$

- Generally, then an activation function (threshold) is used to determine a binary output.

3.3 Linear Regression:

First proposed to predict behavior by Francis Galton in the 1920s, linear regression is the relation between two given continuous variables, it consists from one or more independent variables and a dependent variable (output). According to the results on the type of relation and how related the variables are, a prediction can be mad. Applying this method to solve our problem would mean establishing the relation between each of the variables considering and the overall result.

3.4 Linear Discriminant Analysis

Formulated in 1936 by Ronald A. Fisher, linear discriminant analysis is a very common technique for dimensionality reduction problems as a pre-processing step for machine learning and pattern classification applications. This method maximizes the ratio of between class variance to the within-class variance in any data set thereby guaranteeing maximal separability

4. Matrix

Est.Exterior	Periodo	Etnia	Curso de Preparacion	Curso docentes IDC	Curso docente IES
Si/ /No	2017-1...	Si/ /No	Si/ /No	No tomo curso...	No tomo curso...
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Organized with all the variables to take in count and data of each student including empty spaces due to unanswered questions.
This is a dynamic vector of dynamic vectors.

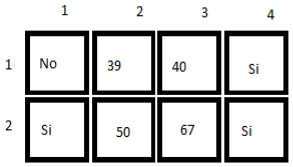
4.1 Complexity analysis

Method	Complexity
Read Data	O(n*m)
Search Student	O(n)
Search Student Data	O(n+m)
Add Student	O(n)

4.2 Design criteria of the data structure

We chose a list on python because this lets us organize the data in a way that we can access it in a O(1) complexity and also add new information at the end of the list in O(1), this will be very helpful as we will be able to minimize the execution time a lot at the moment of using the data for the construction of the decision tree

4.3 Operations of the data structure



Adding a line to the matrix after reading enough data from the archive, organizes the data read from the csv text to an array and then adds that array at the end (new line) of the already existing matrix. This helps to efficiently organize the data in an efficient and accessible Matrix.

4.4 Execution time

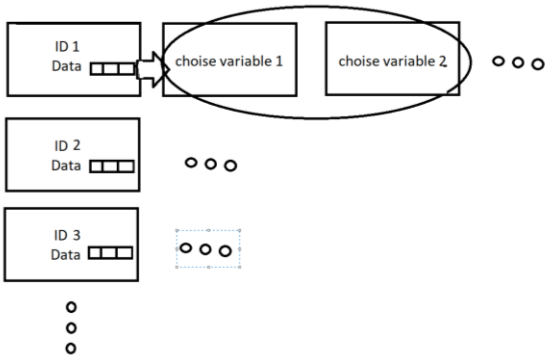
	Data set 0 15000	Data set 1 45000	Data set 2 75000	Data set 3 105000	Data set 4 135000
Read Data	0.275s	0.791s	1.378s	1.99s	2.61s
Search Student	0.004s	0.016s	0.027s	0.032s	0.041s
Search Student Data	0.005s	0.015s	0.025s	0.031s	0.046s

4.5 Memory used

	Data set 0 15000	Data set 1 45000	Data set 2 75000	Data set 3 105000	Data set 4 135000
Read Data	42.39 Mb	127.09 Mb	211.88 Mb	296.65 Mb	381.43 Mb

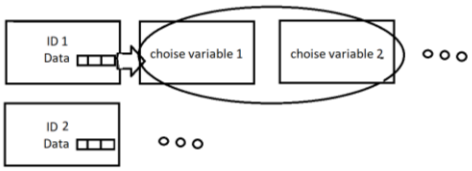
5. Title of the last data structure designed

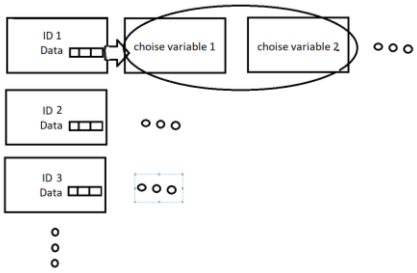
Dictionary with all the information of the students organized in dynamic arrays for each variable (column)



5.1 Operations of the data structure

-Add information of a new student





-Get all students and their information

-The data structure is used to get the information to get the gini impurity of every variable later

5.2 Design criteria of the data structure

We decided to change to a dictionary as the most used method is the access to a certain information of a student when calculating the Gini indexes and this is the most done operation in the program and dictionaries have a O(1) complexity in the access operation, also the dictionary made it easier to track a student since the use of their id helped differentiate each student instead of using an array which would have made it harder to keep track of the students, since you're just saving their positions in the array.

5.3 Complexity analysis

Method	**Complexity**
Read Data	O(n*m)
Search Student	O(1)
Search Student Data	O(m)
Add Student	O(n)
Calculate gini	O(n*k)
Create tree	O(2^n*(n*m)^2*k)
Test data	O(2^n*n*m)

N = number of students

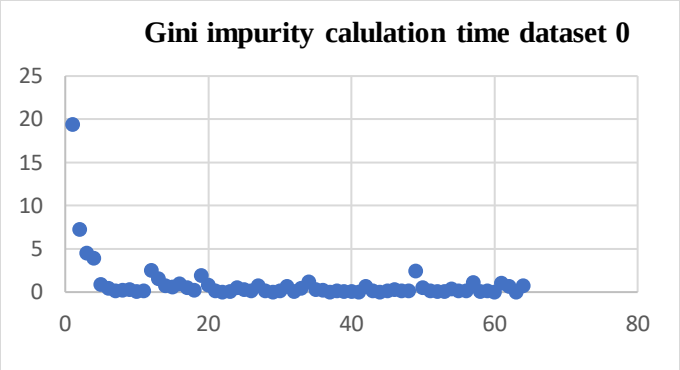
M = number of variables

K = possible dividers for each gini calculation

5.4 Execution times

	Data set 0 15000	**Data set 1 45000**	**Data set 2 75000**	**Data set 3 105000**	**Data set 4 135000**
Gini Calculation	**1s Max:16s**	**3s Max:59s**	**5.48s Max:97s**	**8.11s Max:140s**	**11.5s Max:180s**
Tree construction	**64s**	**212s**	**360s**	**532s**	**750s**
Read Data	**0.275s**	**0.791s**	**1.378s**	**1.99s**	**2.61s**

Gini impurity calculation times tend to decrease exponentially in their calculations because each time it is calculated, it is calculated with roughly half the amount of data it with which it was calculated before.



5.5 Memory usage

	Data set 0 15000	Data set 1 45000	Data set 2 75000	Data set 3 105000	Data set 4 135000
Gini calculation	62.38 Mb	153.71 Mb	275.43 Mb	357.54 Mb	464.33 Mb
Tree maker	90.92 Mb	203.52 Mb	336.17 Mb	441.05 Mb	593.32 Mb
Read Data	42.39 Mb	127.09 Mb	211.88 Mb	296.65 Mb	381.43 Mb

5.6 Result analysis

In each of the cases the accuracy was around 76% in the lower case 74.7% and the higher case 78.4%, we think this could be because we are making trees of maximum 5 levels counting leaf nodes and also can be because of the way we managed the unknown data in the code.

6 Conclusions

After the realization of this project we conclude that the creation of a decision tree has a very high complexity and this causes some trouble at the time efficiency of the algorithm as we had to choose between time efficiency and accuracy of the program, as we decided to use a small height for the tree we had some good times but not the best accuracy, anyways we have a program in which changing the height of the tree is a very easy process and depending on the specific desire of who uses it can change between one and the other very easily.

We also concluded from the analysis of the results there are some certain variables that always have more importance in the student's success as these variables appear on every tree made on every dataset on the same height.

Finally we recognize the hard work of those who dedicate their lives and jobs to keep improving the methods to develop and use decision trees, random forests and many other AI techniques.

References

Perceptron. n.d. *DeepAI*. https://deepai.org/machine-learning-glossary-and-terms/perceptron, Retrieved February 9, 2020.

Swaminathan, S. 2018. Linear Regression — Detailed View. *Medium*. https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86, February 9, 2020.

AIP Conference Proceedings 1982, 020005 (2018); https://doi.org/10.1063/1.5045411 Retrieved: February 9, 2020

Yiu, T. 2019. Understanding Random Forest. *Medium*. https://towardsdatascience.com/understanding-random-forest-58381e0602d2, February 9, 2020.

Tharwat, A. 2017. IOS Press Content Library. *IOSpress*. http://content.iospress.com/articles/ai-communications/aic729, February 10, 2020.