DECISION TREES TO PREDICT COLOMBIAN STUDENT'S SUCCESS ON SABER PRO BASED ON SABER 11 RESULTS

Juan Jose Escobar Bernal
Universidad EAFIT
Colombia
jjescobarb@eafit.edu.co

Simón Lopera Botero
Universidad EAFIT
Colombia
sloperab1@eafit.edu.co

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

ABSTRACT

The purpose of this report is to find a connection between the scores of Colombian bachelor students in Pruebas Saber 11 and Pruebas Saber Pro; as well as, socio-economic details regarding their personal and familiar lifestyle. Through this relationship, it will be possible to predict the results on the Pruebas Saber Pro resulting in scores above or below average.

Resulting predictions from the use of the decision tree can help solve the deficiency in education found in Colombia. It can help find a correlation between socio-economic conditions and results in Pruebas Saber Pro; allowing a hint and suggestions towards the most vulnerable stratum to destine more funds directly.

1.INTRODUCTION:

Education and technology are two areas that have been left behind in comparison with other countries in the region. Using decision trees to predict student results in the Pruebas Saber Pro, allows improvement in Colombian education. It helps distribute resources in the education area of society.

The information from each student that will be used includes the specific results on every area of the Pruebas Saber 11, information about their lifestyle, where they live, as well as parent occupations. The data includes more than 30 different factors that help the prediction on their success be as accurate as possible.

2.PROBLEM:

The problem consists in predicting if a student taking the Pruebas Saber Pro will have a score below or above the average using information from their lifestyle and their results on the Pruebas Saber 11 with the implementation of a decision tree.

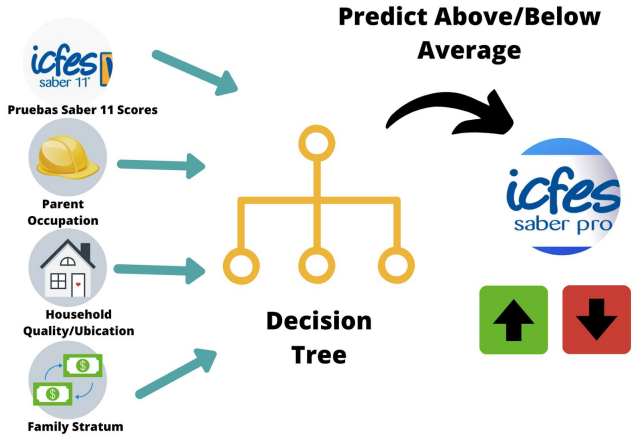

Figure 1. Visual representation of the problem [6][3]

3.POSSIBLE SOLUTIONS:

3.1. ID3

The iterative dichotomiser is a type of decision tree created by Ross Quinlan. It is considered to be one of the simplest but yet very efficient type of decision tree. It works by taking all of the attributes and sorting them by highest information gain. It tests the attribute with the highest information gain and then continues down a bunch of branches in a certain order until it gets to a final result. The algorithm works mostly with categorical attributes and is not the fastest since it has to check one attribute at a time, some others can check multiple attributes at the same time. [5]

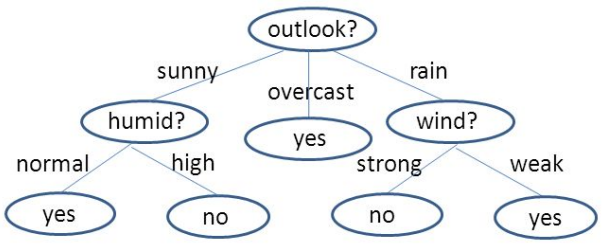

Figure 2: ID3 decision type tree [2]

3.2. C4.5

C4.5 is an algorithm that works as an extension to ID3 developed and created by Ross Quinlan. It is used for classification and works more effectively than ID3 because it works with continuous and discrete variables. It forms numerous reduced trees that are easier to understand and divides the problem multiple times. C4.5 is practical to use making its understanding easier while being able to use categorical and continuous values;however, it is not recommended for small sets.[5]

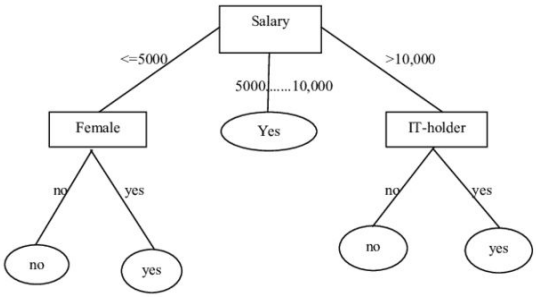

Figure 3: C4.5 decision type tree [1]

3.3. Random Forest

Random Forest is a type of decision tree created by Leo Breiman, consisting of numerous simple trees that return a response given certain predictor values declared at the start. It works for both classification and regression values and problems. This type of tree is one of the most accurate learning algorithms that gives an estimate of the most valuable variables classified. As a disadvantage, it is not easily interpreted by humans.[5]

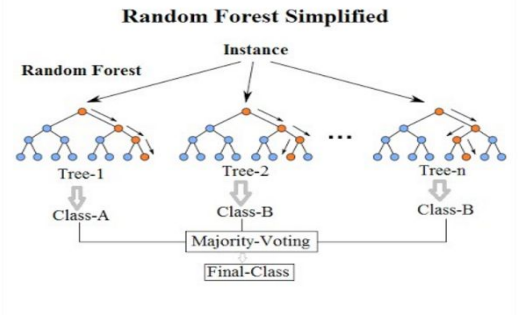

Figure 4: Random Forest decision type tree

```
To generate c classifiers:
    for i = 1 to c do
        Randomly sample the training data D with replacement to produce Di
        Create a root node,Ni containing Di
        Call BuildTree(Ni)
    end for

    BuildTree(N):
if N contains instances of only one class then
        return
    else
    Randomly select x% of the possible splitting features in N
    Select the feature F with the highest information gain to split on
    Create f child nodes of N,N1,NF, where F has f possible values(F1,Ff)
    for i = 1 to f do
        Set the content of Ni to Di,where Di is all intansces in N that match
        Fi
        Call BuildTree(Ni)
    end for
end if
```

Figure 5: Random Forest PseudoCode[7]

3.4. CART

CART, also known as Classification and Regression trees, is a type of decision tree that includes both classification and regression trees. One particular attribute about CART trees is that it only does binary separation, meaning every decision can only take two possible routes.[5]

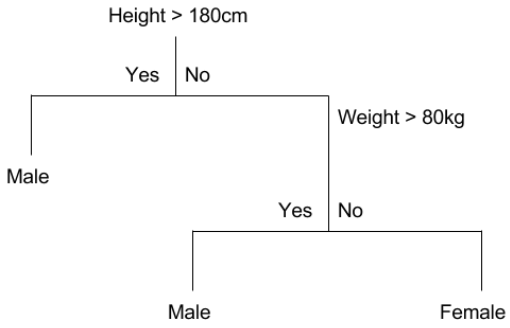

Figure 6: CART decision type tree [4]

DATA STRUCTURE DESIGN

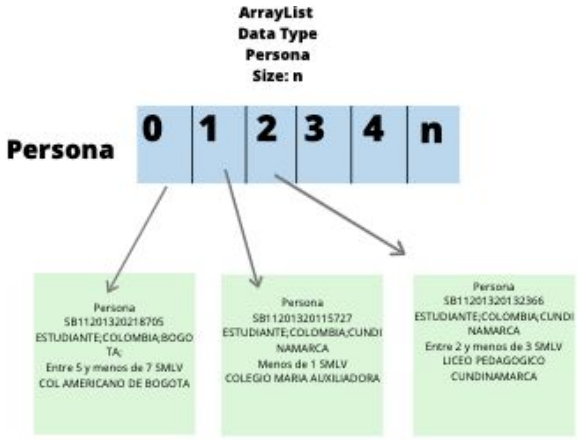

Figure 7. Visual representation of data structure. Each Persona contains an ID,Status,City,Salary,School and all the other ICFES questions.[6]

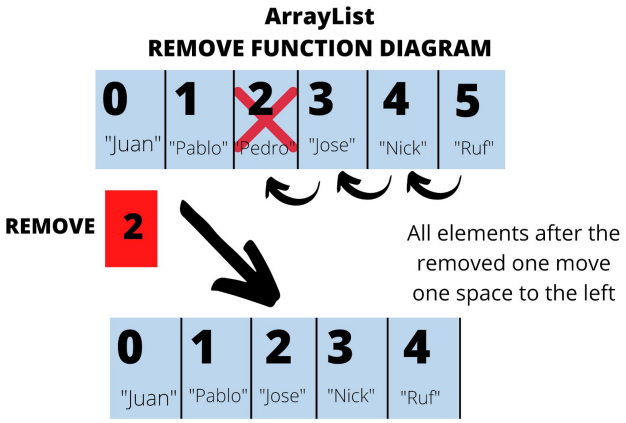

Figure 8. Visual Representation of Remove Function on ArrayList[6]

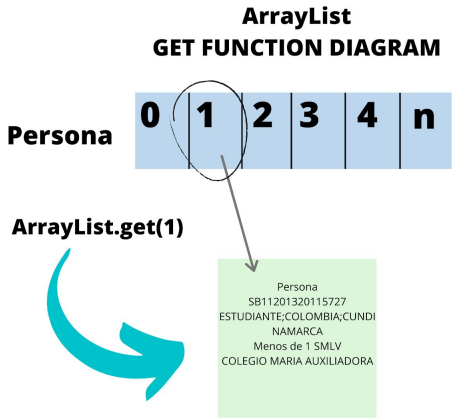

Figure 9. Visual Representation of Get Function on ArrayList[6]

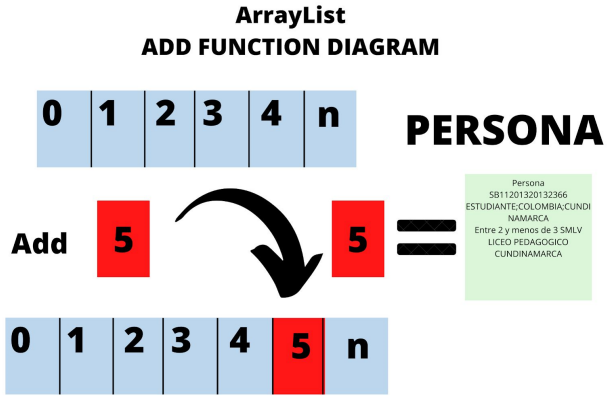

Figure 10. Visual Representation of Add Function on ArrayList[6]

DATA STRUCTURE COMPLEXITY

PERSONA ARRAYLIST FUNCTION	O COMPLEXITY
ADD	O(1)
GET	O(1)
REMOVE	O(n)

DATA STRUCTURE DESIGN CRITERIA

In order to decide which data structure to use; complexity,memory usage and execution time were three factors taken into account in order to choose a structure suitable for the desired solution. Since ArrayList has a complexity of O(1) in both add and get, these being the most used functions(remove is rarely used in this case), it became an optimal structure that would run on a short executable time. Regarding memory usage, ArrayList is not exactly the most optimal structure since according to StackOverflow “Whenever ArrayList reaches its current limit a new much bigger backing array is created and all of the old items are copied in. This can lead to massive arrays only half-filled.To avoid this,... measure the count beforehand and create the array with full capacity.” This allowed the algorithm to reduce the amount of memory used since the amount of items is constant and no “half-filled” spaces will be left. Taking both of these benefits into account, ArrayList was a practical and useful structure that has fast running times and low complexity while keeping memory usage low.

REFERENCES

[1] Anon. 2011. Decision Trees – C4.5. (November 2011). Retrieved February 9, 2020 from https://octaviansima.wordpress.com/2011/03/25/decision-trees-c4-5/

[2] Anon.Retrieved February 8, 2020 from https://www.philippe-fournier-viger.com/spmf/ID3.php

[3] Anon. Inscripciones abiertas para pruebas Saber 11 calendario A, Pre Saber y validación del bachillerato académico. Retrieved February 6, 2020 from https://www.valledelcauca.gov.co/publicaciones/62908/inscripciones-abiertas-para-pruebas-saber-11-calendario-a-pre-saber-y-validacion-del-bachillerato-academico/

[4] Jason Brownlee. 2019. Classification And Regression Trees for Machine Learning. (August 2019). Retrieved February 7, 2020 from https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/..

[5] Bhumika Gupta, Aditya Rawat, Akshay Jain, Arpit Arora and Naresh Dhami. Analysis of Various Decision Tree Algorithms for Classification in Data Mining. *International Journal of Computer Applications* 163(8):15-19, April 2017.

[6] Anon. Canva - Collaborate & Create Amazing Graphic Design for Free. Retrieved February 9, 2020 from https://www.canva.com/

[7] Luluah Alhusain and Alaaeldin M. Hafez. 2017. Cluster ensemble based on Random Forests for genetic data. (December 2017). Retrieved February 10, 2020 from https://link.springer.com/article/10.1186/s13040-017-0156-2

[8] AdarshAdarsh 6111 silver badge66 bronze badges, OldCurmudgeonOldCurmudgeon 57.9k1414 gold badges9898 silver badges184184 bronze badges, Zbynek Vyskovsky - kvr000Zbynek Vyskovsky - kvr000 15.6k22 gold badges2424 silver badges3737 bronze badges, and ControlAltDelControlAltDel 25.1k55 gold badges3838 silver badges6565 bronze badges. 1965. High Memory consumption for ArrayList object. (October 1965). Retrieved April 3, 2020 from https://stackoverflow.com/questions/34338695/high-memory-consumption-for-arraylist-object