TITLE (A SHORT DESCRIPTION OF THE PROJECT, BEWEEN 8 AND 12 WORDS)

Name of first author University (name in Spanish) Country E-mail at Eafit	Name of second author University (name in Spanish) Country E-mail at Eafit	Miguel Correa Universidad Eafit Colombia macorream@eafit.edu.co	Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co

For each version of this report: 1. Detele all text in red. 2. Adjust spaces among words and paragraphs. 3. Change the color of all the texts to black.

Red text = Comments

Black text = Miguel and Mauricio's contribution

Green text = To complete for the 1st deliverable

Blue text = To complete for the 2nd deliverable

Violet text = To complete for the 3rd deliverable

ABSTRACT

To write an abstract, you should answer the following questions in a paragraph: What is the problem? Why is the problem important? Which are the related problems? Which is the algorithm you proposed?, What results did you achieve? , What are the conclusions of this work? Abstract should have **at most 200 words**. (*In this semester, you should summarize here execution times, memory consumption, accurracy, precision and sensibility*)

Keywords

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

1. INTRODUCTION

Explain the motivation, in the real world, that leads to the problem. Include some history of this problem. *(In this semester, motivation is why we need to predict test scores or academic success in bachelor degrees in Latin America)*

1.1. Problem

In a few words, explain the problem, the impact that has in society and why it is important to solve the problem. *(In this semester, the problem is to predict academic success)*

1.2 Solution

In this work, we focused on decision trees because they provide great explainability (*A citation for this argument is missing!*). We avoid black-box methods such as neural networks, support-vector machines and random forests because they lack explainability. (*Another citation for this argument is missing!*)

Explain, briefly, your solution to the problem *(In this semester, the solution is an implementation of a decision-tree algorithm to predict academic success. Which algorithm did you choose? Why?)*

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

Explain four (4) articles related to the problem described in Section 1.1. You may find the related problems in scientific journals. Consider Google Scholar for your search. *(In this semester, related work is research on decision trees to predict student-test scores or academic success)*

3.1 Write a title for the first related problem

You should mention the problem they solved, the algorithm they used, the accuracy they achieved, and the citation.

3.2 Write a title for the second related problem

You should mention the problem they solved, the algorithm they used, the accuracy they achieved, and the citation.

3.3 Write a title for the third related problem

You should mention the problem they solved, the algorithm they used, the accuracy they achieved, and the citation.

3.4 Wite a title for the fourth related problem

You should mention the problem they solved, the algorithm they used, the accuracy they achieved, and the citation.

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

In what follows, we present different algorithms to solve to automatically build a binary decision tree. *(In this semester, examples of such algorithms are ID3, C4.5 and CART).*

3.2.1 Name of the first algorithm

Please explain the algorithm, its complexity and include a vectorized Figure.

3.2.2 Name of the second algorithm

Please explain the algorithm, its complexity and include a vectorized Figure.

3.2.3 Name of the third algorithm

Please explain the algorithm, its complexity and include a vectorized Figure.

3.2.4 Name of the fourth algorithm

Please explain the algorithm, its complexity and include a vectorized Figure.

4. ALGORITHM DESIGN AND IMPLEMENTATION

In what follows, we explain the data structure and the algorithms used in this work. The implementation of the data structure and algorithm is available at Github[1].

4.1 Data Structure

Explain the data structure used to make the prediction and make a figure explaining it. Do not use figures from the Internet. *(In this semester, the data structure is a binary decision tree)*

Figure 1: A binary decision tree to predict Saber Pro based on the results of Saber 11. Violet nodes represent those with a high probability of success, green medium probability and red a low probability of success.

4.2 Algorithms

Explain the design of the algorithm to solve the problem and make a figure. Do not use figures from the Internet, make your own. *(In this semester, one algorithm must be an algorithm to train a decision-tree algorithm such as ID3, C4.5, CART and the second algorithm must be an algorithm to classify new data using such a tree).*

4.2.1 Training the model

Explain, briefly, how did you train the model: This is equivalent to explain how does your algorithm build automatically a binary decision tree.

[1]http://www.github.com/ ????????? /proyecto/

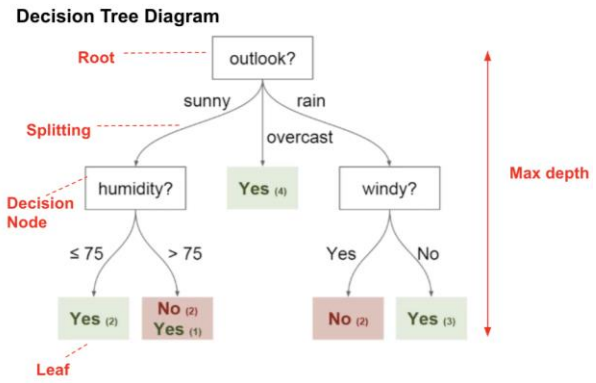

Figure 2: Training a binary decision tree using *(In this semester, one could be CART, ID3, C4.5… please choose)*. In this example, we show a model to predict whether or not to play Golf, according to weather.

4.2.2 Testing algorithm

Explain, briefly, how did you test the model: This is equivalent to explain how does your algorithm classifies new data after the tree is built.

4.3 Complexity analysis of the algorithms

Explain in your own words the analysis for the worst case using O notation. How did you calculate such complexities.

Algorithm	Time Complexity
Train the decision tree	$O(N^2*M^2)$
Test the decision tree	$O(N^3*M*2^N)$

Table 2: Time Complexity of the training and testing algorithms. *(Please explain what do N and M mean in this problem.)*

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

Table 3: Memory Complexity of the training and testing algorithms. *(Please explain what do N and M mean in this problem.)*

4.4 Design criteria of the algorithm

Explain why the algorithm was designed that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory consumption. Examples of non-objective criteria are: "I was sick", "it was the first data structure that I found on the Internet", "I did it on the last day before deadline", etc. Remember: This is 40% of the project grading.

5. RESULTS

5.1 Model evaluation

In this section, we present some metrics to evaluate the model. Accuracy is the ratio of number 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 n*
Accuracy	0.7	0.75	0.9
Precision	0.7	0.75	0.9
Recall	0.7	0.75	0.9

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 n*
Accuracy	0.5	0.55	0.7
Precision	0.5	0.55	0.7
Recall	0.5	0.55	0.8

Table 4. Model evaluation on the test datasets.

5.2 Execution times

Compute execution time for each dataset in github. Measure execution time 100 times for each dataset and report average execution time for each dataset.

	Dataset 1	*Dataset 2*	*...Dataset n*
Training time	10.2 s	20.4 s	5.1 s
Testing time	1.1 s	1.3 s	3.3 s

Table 5: Execution time of the *(Please write the name of the algorithm, C4.5, ID3)* 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	10 MB	20 MB	5 MB

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

To measure memory consumption, you should use a profiler. An very good one for Java is VisualVM, developed by Oracle, http://docs.oracle.com/javase/7/docs/technotes/guides/visualvm/profiler.html For Python, use C Profiler.

6. DISCUSSION OF THE RESULTS

Explain the results obtained. Is precision, accuracy and sensibility appropriate for this problem? Is the model over-fitting? Is memory consumption and time consumption appropriate? *(In this semester, according to the results, can this be applied to give scholarships or to help students with low probability of success? For which one is better?)*

6.1 Future work

Answer, what would you like to improve in the future? How would you like to improve your algorithm and its implementation? What about using random forest?

ACKNOWLEDGEMENTS

Identify the kind of acknowledgment you want to write: for a person or for an institution. Consider the following guidelines: 1. Name of teacher is not mentioned because he is an author. 2. You should not mention websites of authors of articles that you have not contacted. 3. You should mention students, teachers from other courses that helped you.

As an example: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved the manuscript.

REFERENCES

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

As an example, consider this two references:

1.Adobe Acrobat Reader 7, Be sure that the references sections text is Ragged Right, Not Justified. http://www.adobe.com/products/acrobat/.

2. Fischer, G. and Nakakoji, K. Amplifying designers' creativity with domainoriented design environments. in Dartnall, T. ed. Artificial Intelligence and Creativity: An Interdisciplinary Approach, Kluwer Academic Publishers, Dordrecht, 1994, 343-364.