COFFEE LEAF RUST SOLVING: WIRELESS SENSOR NETWORK, DATA STRUCTURES AND ALGORITHMS APPLICATIONS

Pablo Buitrago
Universidad EAFIT
Colombia
pbuitragoj@eafit.edu.co

Miguel Ángel Correa Manrique
Universidad EAFIT
Colombia
macorream@eafit.edu.co

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

ABSTRACT

Humanity presents diverses hazards linked to the agriculture production, sometimes leaded to the distribution or production process, when the last one is a little more difficult to embrace because of the fact that is a natural growth process handed by people or machines.

An example of what was mentioned is the coffee leaf rust (CLR) disease presented in cultivation caturra coffee, a disease that is devastating susceptible coffee plantations, farms have to manage how to solve this problem such as putting hybrids that have strong genetic resistance to rust.

Another way to prevent this is finding a way to detect these diseases by technology prompt resources that could detect them before the mitigation of all plant plantations, increasing the plant survival rate and of course the production amount on farms leaded to resources generated to feed humanity.

KEYWORDS

Decision tree, Coffe rust, Agriculture, Machine Learning, Data Structures

ACM CLASSIFICATION KEYWORDS

CCS → Theory of computation → Theory and algorithms for application domains → Machine learning theory → Structured prediction

1. INTRODUCTION

Nowadays humanity faces difficulties against different ways in how to get resources to feed themselves, particularly with those related with agriculture ambits.

Studies have found that the one majority global warming mitigation comes from meat production and distribution, what leads to think about alternatives such as seeding that could reduce this carbon footprint.

Encountering contrivances that help to do seeding will contribute to the progress of the alternatives spoken such as the ways that could diminish diseases while the seeds grow.

The project has been focused in the CLR disease detection and resolving by the develop of a system artefact that could reduce this disease on caturra coffee, identifying the relevant data needed.

2. PROBLEM

EAFIT University has detected adversities in the growth process of caturra coffee, topic related with the CLR disease, evidencing that having an artificial artefact that could recognize diseases will increase the survival rate on a plantation, meaning it is needed to develop this system and artefact recognition solution to the caturra coffee cultivation disease by qualifying each of them, decision which is derived by the data and information collected from each plant, such as temperature, humidity, illumination, ph, etc... in the plantation.

3. RELATED WORK

3.1 Random Forest

It stands random forest because of the fact that it is used a lot of decision trees.

Random forest algorithm works as a large collection of decorrelated decision trees and it used them to make a classification.

$$S = \begin{bmatrix} f_{A1} & f_{B1} & f_{C1} & C_1 \\ \vdots & & \vdots & \\ f_{AN} & f_{BN} & f_{CN} & C_N \end{bmatrix}$$

Figure 1 : Table random forest representation [7]

The matrix S is a matrix of training samples that were submitted to the algorithm to create a classification model, in S f is defined as feature, each arrow is defined as a sample and C stands for all the other features, meaning that it is already obtained a training class.

The aim is to create a random forest to classify the sample set. In order to achieve this is necessary to create random subsets with random values, where for each subset a decision tree will be created and depending on their evaluation it is going to classified the sample.

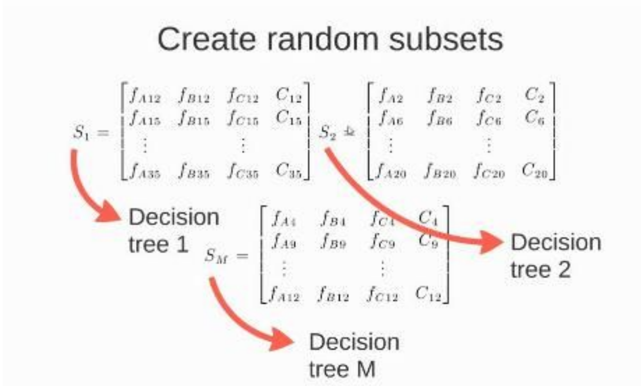
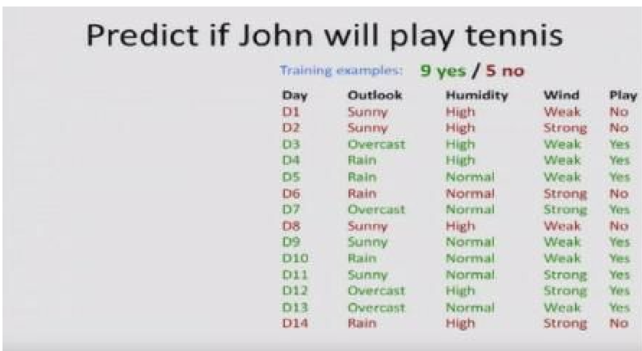


Figure 2 : Random forest's subsets[7]

3.2 ID3



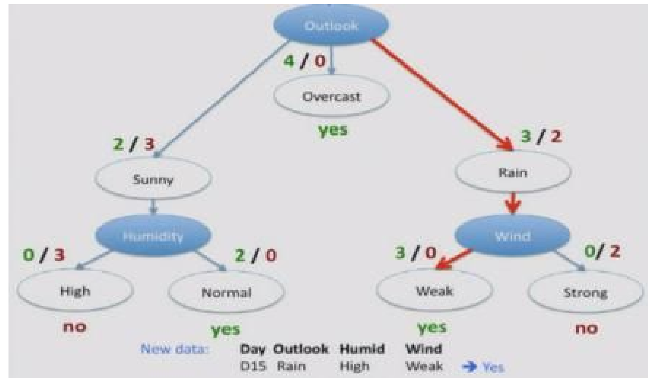
Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

Figure 3 : Data representation for ID3 [6]

It is a recursive algorithm which is going to be divided into "nodes", every point in the algorithm were the data is split because of the fact that one have a lot of training examples with different values on each attribute.

For some attribute A the algorithm will evaluate for the root node if it is already in a pure or not subset, defining them as when while the attribute is given for some classification the attribute is embrace by just one classification or if it is mixed with others respectively.

Being the first case the algorithm stops, it already knows what is the classification for the sample, but if not it will continue with the next child node, evaluating the next attribute given on the sample, repeating the process of evaluation with the pure or not subsets for each child node.



It takes a training set of data and it is going to be sorted into pure subsets based on some attribute values.

Figure 4: Decision Tree Algorithm Representation [6]

3.3 CART

Classification And Regression Trees, or commonly known as CART , is a decision tree algorithm that can be used for classification or regression predictive modeling problems.

It is a binary tree where each root node represents a single input variable and a split point on that variable (assuming that is numeric); The leaf nodes contain an output variable which is used to make a prediction.

```
If Height > 180 cm Then Male
If Height <= 180 cm AND Weight > 80 kg Then Male
If Height <= 180 cm AND Weight <= 80 kg Then Female
Make Predictions With CART Models
```

Figure 5 : Binary representation of a CART model example [?]

Given a new input, the tree is traversed by evaluating the specific input started at the root node of the tree.

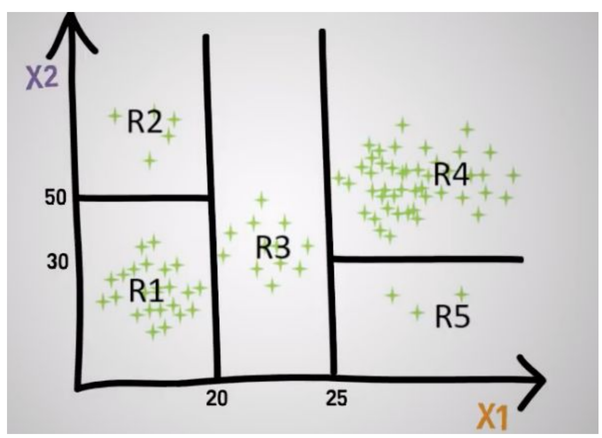


Figure 6 : Cart algorithm representation [8]

A learned binary tree is actually a partitioning of the input space. You can think of each input variable as a dimension on a p-dimensional space. The decision tree split this up into rectangles (quadrants) where all the data will be accommodated once is filtered by the algorithm . It is usually used with greedy splitting (the best split is always selected). The Gini index is used for classification, providing an indication of how "pure" the leaf nodes are.

$$G = \text{sum}(pk * (1 - pk))$$

Where G is the Gini index over all classes, pk are the proportion of training instances with class k in the rectangle of interest. A node with perfect class purity will have G=0, and a node with the worst purity will have G=0.5

The most common stopping procedure is to use a minimum count on the number of training instances assigned to each leaf node. If the count is less than some minimum then the split is not accepted and the node is taken as a final leaf node.

3.4 CHAID

Chi-squared automatic interaction detection or commonly known as CHAID, CHAID will "build" non-binary trees (i.e., trees where more than two branches can attach to a single root or node), based on a relatively simple algorithm that is particularly well suited for the analysis of larger datasets, which for classification problems (when the dependent variable is categorical in nature) relies on the Chi-square test to determine the best next split at each step.

The algorithm proceeds as follows:

Preparing predictors. The first step is to create categorical predictors by dividing the distributions, with an approximately equal number of observation into a number of categories.

Mergin categories. Then, the step is to cycle through the predictors to determine for each predictor, the pair of categories that is at least significantly different with respect to the dependent variable. For classification problems with categorical dependent variable it will compute the Pearson Chi-square test. If the test for a given pair of predictor categories is not statistically significant as defined by an alpha to merge value, it will merge the respective predictor categories and repeat this step (find the next pair of categories, which now may include previously merged categories);

Selecting the split variable. The next step is to choose the split variable with the smallest adjusted p-value (the predictor variable that will yield the most significant split). If the smallest adjusted p-value is greater than some alpha to split value, then no splits will be performed and the respectictive node is a terminal one (This process will continue until no further splits can be performed with the given alpha values).

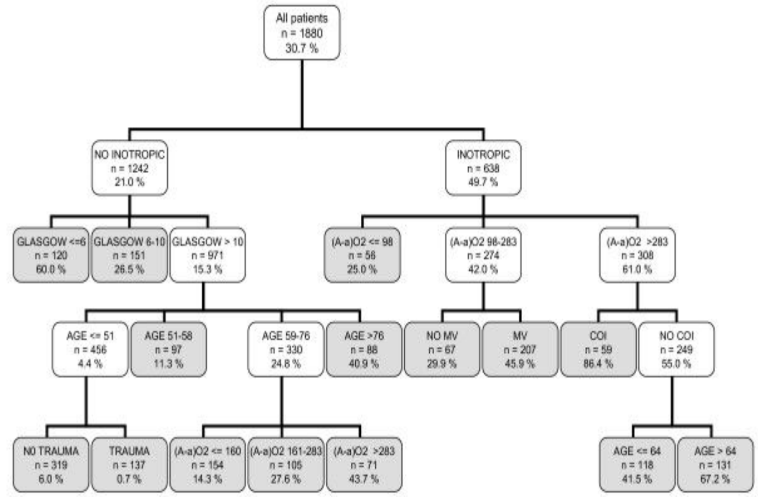


Figure 7 : CHAID Tree classification example [11]

4 .0

TABLE

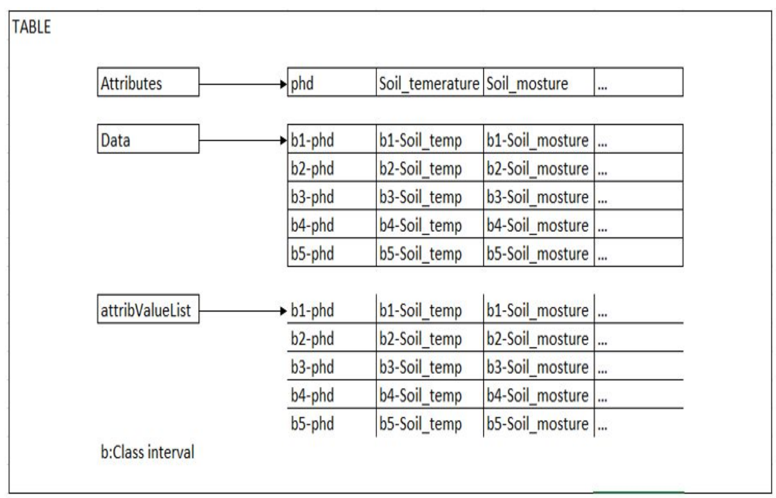


Figure 8: On "Data" is possible to find repeated elements, it represents the data set classification (4.2 for more information).

MyLittleTree

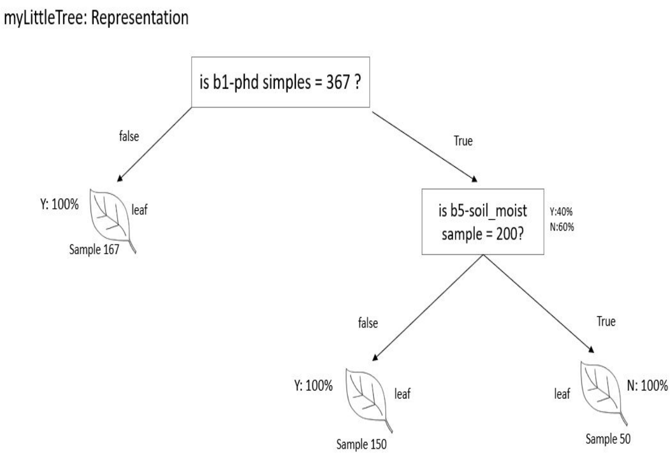


Figure 9 : myLittleTree representation

4.1 TABLE OPERATIONS

attribValues Method

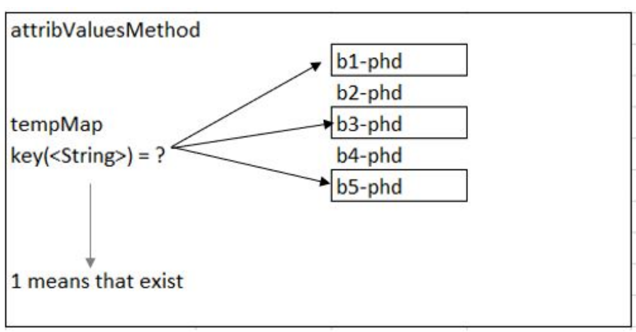


Figure 10: the attribValueMethod contains only existent class intervals. It gives the categories on each column of the data set (already discretized).

discretizeData Method

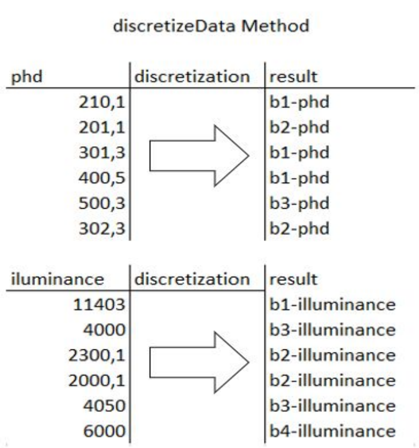


Figure 11 : Demonstrates how the data is discretized graphically.

4.2 DESIGN OF THE DATA STRUCTURE

Our named “Table” data structure is going to behave as its name tells.

Composed by:

Attributes: As string vectors.

Data: As vectors of string vectors.

Attribute List: As vectors of string vectors.

Since it was necessary to store all the values of the data set to perform the solution, but also because it gives us the possibility to access the data in constant time (operation that it is going to be performing a lot through the program); don’t have to do deletion or insertion on different indexes, ArrayList or vector, was selected because of this reason.

Attributes stores the amount of attributes, that indicates the number of columns of our table and Attribute List stores the categorization that could receive data on a specific column.

One thing to clarify is that data could have been a vector of another data structure that contains the information of a particular information row, but for the convenient data manipulation it is better to visualize it as matrices or vectors of vectors, one doesn’t know how much amount of data is going to be in.

4.3 COMPLEXITY ANALYSIS

Method	Complexity
AttribValues()	O(n*s)
discretizeData()	O(n)

n : Amount of rows of the table

s : Amount of attributes

Table 1 : Complexity of the methods presented on the data structure.

4.4 EXECUTION TIME

	Data Set with Len 300	Data Set with Len 373	Data Set with Len 457	Data Set with Len 673
Creation	0.000628721s	0.000742888s	0.000890782s	0.00133364s

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

4.5 MEMORY USED

	Data Set with Len 300	Data Set with Len 373	Data Set with Len 457	Data Set with Len 673
Memory Usage	0.0672 MB	0.08332 MB	0.10214 MB	0.15052 MB

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

Note that the memory usage is an approximation given by the amount of data received multiplicated by the bytes of the string data type in C++ language times the number of rows presented on the data set.

4.6 RESULT ANALYSIS

	Best TIme	Worse TIme	Average Time	Best Memory	Worse Memory	Average Memory
Creation	0.000628721s	0.00133364s	0.000895s	0.0672 MB	0.15025 MB	0.1 MB

Table 4: Analysis of the results

Considering the data sets given, these are the results, notice that more data means more computation deriving in a lot of processor, ram and storage consuming.

REFERENCES

[1.Abrams, A. How Eating Less Meat Could Help Protect the Planet From Climate Change. *Time*, 2019. https://time.com/5648082/un-climate-report-less-meat/ .

[2.Augmented Startups. *Decision Tree (CART) - Machine Learning Fun and Easy*. 2017. https://www.youtube.com/watch?v=DCZ3tsQIoGU

[3.Brownlee, J. Classification And Regression Trees for Machine Learning. *Machine Learning Mastery*, 2019. https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/.

[4.Choose Bin Sizes for Histograms in Easy Steps + Sturge's Rule - Statistics How To. *Statistics How To*, 2018.https://www.statisticshowto.datasciencecentral.com/choose-bin-sizes-statistics/.

[5.EAFIT, U. Eafitenses aprovechan la inteligencia artificial para diagnosticar la roya del cafeto. *Eafit.edu.co*, 2019. http://www.eafit.edu.co/noticias/agenciadenoticias/2019/eafitenses-aprovechan-inteligencia-artificial-para-diagnosticar-roya-cafe.

[6.Google Developers. *Let's Write a Decision Tree Classifier from Scratch - Machine Learning Recipes #8*. 2017.

https://www.youtube.com/watch?v=LDRbO9a6XPU

[7.Gupta, B., Rawat, A., Jain, A., Arora, A. and Dhami, N. Analysis of Various Decision Tree Algorithms for Classification in Data Mining. International Journal of Computer Applications, 163 (8), 15-18 https://pdfs.semanticscholar.org/fd39/e1fa85e5b3fd2b0d000230f6f8bc9dc694ae.pdf.

[8.Körting, T. *How Random Forest algorithm works*. 2014. https://www.youtube.com/watch?v=loNcrMjYh64

[9.Raj, R. *Data Mining & Business Intelligence | Tutorial #19 | Data Discretization*. 2018.https://www.youtube.com/watch?v=TEVWfjTNIXk

[10.Sadawi, N. *Transforming Numerical to Categorical: Entropy-based Binning*. 2014.

https://www.youtube.com/watch?v=gmiINKkYcF8&t=533s

[11.Statsoft. Popular Decision Tree: CHAID Analysis, Automatic Interaction Detection. *Statsoft.com*, 2019. http://www.statsoft.com/Textbook/CHAID-Analysis.

TEAM WORK HISTORY :

GITHUB:

* commit c46ebd8

| Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>

| Date: 7 weeks ago

|

| Add files via upload

|

| arreglado espaciado y quité texto en negro enunciado

|

* commit c494ae5

| Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>

| Date: 7 weeks ago

|

| Delete ED1-Guia de Informe - English Vr 1.0(1).pdf

|

* commit 853180a

| Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>

| Date: 7 weeks ago

|

| Add files via upload

|

| version con email de pablo

|

* commit 8d86983

| Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>

| Date: 7 weeks ago

|

| Delete ED1-Guia de Informe - English Vr 1.0.pdf

|

| faltaba el email de pablo

|

* commit 073935d

| Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>

| Date: 7 weeks ago

|

| Delete ED1 entrega proyecto 1.doc

|

* commit 2d693e3

| Author: miguelmque <41609302+MiguelMque@users.noreply.github.com>

| Date: 7 weeks ago

|

| Add files via upload

|

| version pdf

|

* commit 9e9ce7d

| Author: pablo4buitrago <52968548+pablo4buitrago@users.noreply.github.com>

| Date: 7 weeks ago

|

| Add files via upload

DOCUMENT:

TODAY

September 29, 10:40 PM
Current version
- Miguel Mque

September 29, 10:40 PM
- Miguel Mque

September 29, 10:27 PM
- Miguel Mque

September 29, 10:26 PM
- Pablo Buitrago

September 29, 10:21 PM
- Pablo Buitrago

September 29, 10:21 PM
- Pablo Buitrago

September 29, 10:13 PM
- Pablo Buitrago

September 29, 10:11 PM
- Miguel Mque

September 29, 10:09 PM
- Pablo Buitrago

September 29, 10:06 PM
- Pablo Buitrago
- Miguel Mque

September 29, 10:04 PM
- Pablo Buitrago

September 29, 10:03 PM
- Pablo Buitrago

September 29, 10:02 PM
- Pablo Buitrago

September 29, 10:02 PM
- Pablo Buitrago

September 29, 10:02 PM
- Pablo Buitrago

September 29, 10:01 PM
- Pablo Buitrago

September 29, 10:00 PM
- Pablo Buitrago

September 29, 9:58 PM
- Pablo Buitrago
- Miguel Mque

September 29, 9:56 PM
- Miguel Mque

September 29, 9:55 PM
- Pablo Buitrago

September 29, 9:53 PM
- Miguel Mque
- Pablo Buitrago

September 29, 9:52 PM
- Miguel Mque

September 29, 9:50 PM

September 29, 9:32 PM
Pablo Buitrago

September 29, 9:31 PM
Pablo Buitrago

September 29, 9:31 PM
Pablo Buitrago

September 29, 9:30 PM
Pablo Buitrago

September 29, 9:30 PM
Pablo Buitrago

September 29, 9:29 PM
Pablo Buitrago
Miguel Mque

September 29, 9:27 PM
Miguel Mque

September 29, 9:25 PM
Miguel Mque

September 29, 9:23 PM
Miguel Mque

September 29, 9:23 PM
Miguel Mque

September 29, 8:54 PM
Miguel Mque

September 29, 8:54 PM
Miguel Mque

September 29, 8:53 PM
Miguel Mque

September 29, 8:53 PM
Miguel Mque

September 29, 8:51 PM
Miguel Mque

September 29, 8:50 PM
Miguel Mque

September 29, 8:40 PM
All anonymous users

September 29, 8:39 PM
All anonymous users

September 29, 8:37 PM
All anonymous users

September 29, 8:37 PM
All anonymous users

September 29, 8:36 PM
All anonymous users

September 29, 8:31 PM
All anonymous users

September 29, 8:30 PM
All anonymous users

September 29, 8:30 PM
All anonymous users

All anonymous users

August 11, 7:06 PM
All anonymous users

August 11, 7:05 PM
All anonymous users

August 11, 7:03 PM
All anonymous users

August 11, 7:00 PM
All anonymous users

August 11, 6:59 PM
All anonymous users

August 11, 6:58 PM
All anonymous users

August 11, 6:46 PM
Miguel Mque

August 11, 6:41 PM
All anonymous users

August 11, 6:40 PM
All anonymous users

August 11, 6:40 PM
All anonymous users

August 11, 6:40 PM
All anonymous users

August 11, 6:39 PM
All anonymous users

August 11, 6:32 PM
All anonymous users

August 11, 6:32 PM
All anonymous users

August 11, 6:31 PM
All anonymous users

August 11, 6:30 PM
All anonymous users

August 11, 6:30 PM
All anonymous users

August 11, 6:28 PM
All anonymous users

August 11, 6:27 PM
All anonymous users

August 11, 6:27 PM
All anonymous users

August 11, 6:25 PM
All anonymous users

August 11, 11:47 AM
Miguel Mque

August 10, 3:16 PM
Miguel Mque

August 10, 2:42 PM
Miguel Mque