DETECTION OF THE COFFEE LEAF RUST USING DECISION TREES

Andrés Grimaldos
Universidad Eafit
Colombia
agrimaldoe@eafit.edu.co

Alejandro Salazar
Universidad Eafit
Colombia
asalazara1@eafit.edu.co

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

ABSTRACT

Throughout the history, coffee has become an important part of some countries' economy, creating employment and improving the first sector of the economy. Countries as Brazil, Vietnam and Colombia have now an exportation system based on coffee production and marketing. Because of this, the reduction on its cultivation can involve not only a clear decrease on its production rate, but a destabilization of the economy of thousands of people, who depends on it.

That's why concerned about the consequences that a plague as the coffee leaf rust can have in multiple topics, we have proposed to find a way of predicting and controlling this fungus based on decision trees considering the physicochemist data.

We tried to solve the problem through a decision tree built by a CART algorithm which has a 72.15% of accuracy and a data structure based on Python dictionary where the data is stored.

Keywords:

Data structures; coffee leaf; big O complexity; decision trees; algorithms; execution time; storage; CART algorithm

ACM keywords:

Information systems → Data management systems → Data structures → Data layout → Data compression

Theory of computation → Design and analysis of algorithms → Data structures design and analysis → Pattern matching

Theory of computation → Design and analysis of algorithms → Algorithm design techniques → Dynamic programming

Theory of computation → Theory and algorithms for application domains → Database theory → Data structures and algorithms for data management

1. INTRODUCTION

Nowadays, the principal kind of coffee used in exportations is the Coffee arabica, however, there is a fungus called Hemileia vastatrix, that affects this plant producing a disease known as coffee leaf rust which it first appearance was in XIX century and arrived in 1970 in Brazil and Colombia in 1980s, until becoming in a serious problem currently, because facts like climate change, have allowed this complaint spreads out.

Although Coffee arabica has troubles, some solutions were found out related with the management of the plant, like changing the way to irrigate (as dripping) the plant or controlling the humidity levels on the areas, because factors as warmness, humidity and the accumulation of water can be dangerous for the plant and can produce this fungus.

In addition, wireless sensors have had an important place, because they surveillance constantly coffee plants, sending information that can be used to predict the state and decide if coffee leaf rust is occurring, therefore, it opens the options to create algorithms in order to get a solution.

2. PROBLEM

The objective of this project is to design a time efficient algorithm which allow us to predict, in certain time, if the *Coffea arabica* has coffee leaf rust in a certain growing zone.

In this work, we have information about some important physicochemist variables such as soil pH, soil temperature, soil moisture, illuminance, environment temperature and humidity and a label of an expert who qualified whether the cultivation has or no coffee leaf rust.

So, the first input of the program would be the physicochemist variables and after that, receive a second input with the information (except the label), and finally the algorithm, considering some aspects, will say if the cultivation has or no coffee leaf rust.

With this algorithm, the study of this disease will improve and allows to coffee cultivators be careful, and finally the economy of coffee countries will be better.

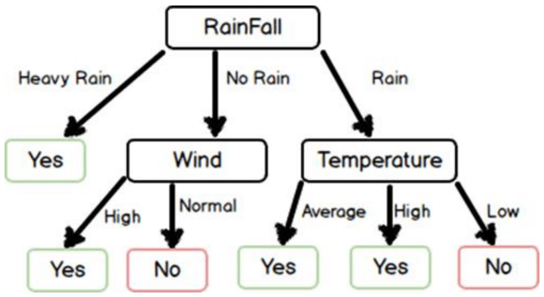
3. RELATED WORK

3.1 ID3 Algorithm:

This algorithm is commonly used in machine learning which, from a dataset, creates a decision tree; to do that, it is necessary to know the entropy and the maximum gain information where first is related with the uncertainly or probability of an event and the second with the difference between the entropy and one of its options. [1]

The decision tree is composed by root nodes, child or options nodes and leaf nodes, also has attributes. Besides, is recursively so it is based on iterations.

The algorithm start in a root node which has the dataset, after that the algorithm search for the best attribute according to the gain information we are looking for, then the set will be split by this decision until get a leaf node which give us the class or classification of that information through a binary answer like yes or no.



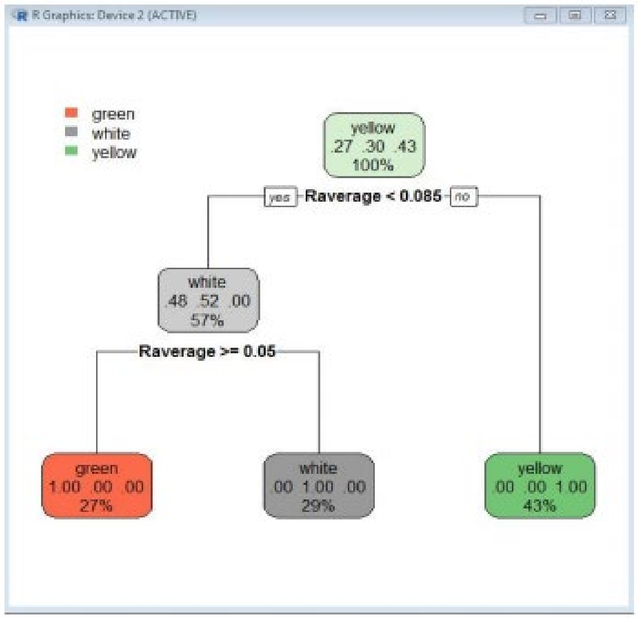
3.2 C4.5 Algorithm

This sequence of paths to do a decision tree is like ID3 algorithm but allow you to extend the strategy to more applications, for example, you can use this algorithm both for discrete and continues variables. As ID3 algorithm, C4.5 create a decision tree based on a training information, so it classifies the information of a certain data.

One of the advantages of this method is a process called pruning which is useful for huge data and improves efficiency ignoring information could be in a bigger generalization., avoiding overfitted (an ID3 algorithm problem).

Has the same structure of ID3 algorithm trees, with nodes, leaves and branches, where the course is also according to the entropy and information gain, starting to calculating the best attribute to split the training information, decreases the entropy until get a decision, because while the iteration divides the data, in each one is will be easier to choose the class of the information, due to information gain is bigger and bigger when entropy is smaller; that is, for each attribute calculate the potential data by a test to get the best attribute to branch on until end the tree. [2]

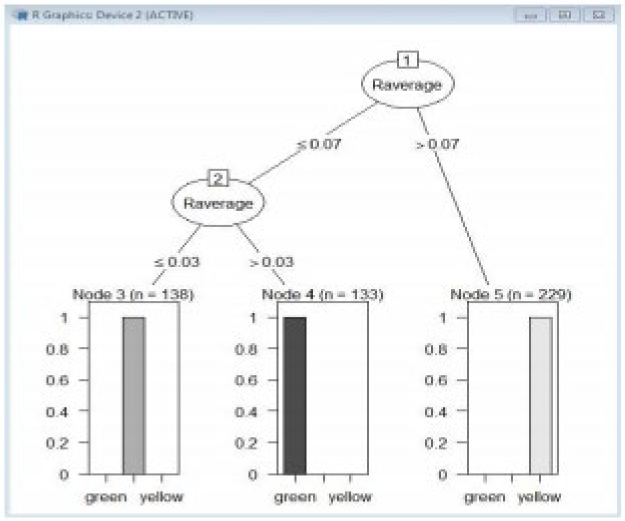
Regarding to the project this kind of algorithms to generate a classifiers trees based on a dataset and aspects which influence in the decision, would be good to implement in order to find out if from a physicochemist information of Coffee arabica, this plant has the coffee leaf rust.



3.3 C5.0 Algorithm

This Algorithm is considered by a lot of people the "upgrade" of the C4.5 Algorithm. It is the industry standard for the construction of decision because of its ease to solve most of the problems out of the box and the facilities it has for users to applicate it. [3]

Its structure is closely alike to C4.5's and uses the same information for building the tree (As it is entropy and information gain) but has new implementation for improving processing speed, memory and efficiency. [4]

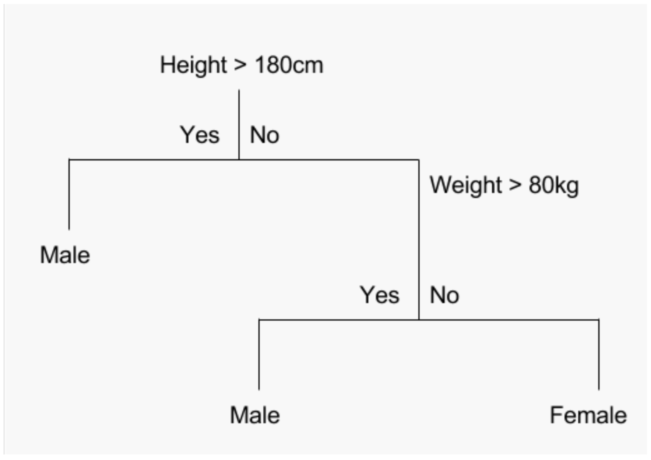


3.4 CART Algorithm

The CART model is a decision tree model which is not only used for classification but for regression too. Its structure is very similar to other decision trees' but its difference with the rest is in the way the tree is built.

The CART algorithm is based, instead of Entropy and information gain values, in the Gini index function, which simplifies the way to define the pureness of the nodes and leaves.

It is also easy to use because of the ease how it can give a good interpretation of the data, only needing it to be a good representation of the problem. [5]



4. DATA STRUCTURE (FIRST IMPLEMENTATION)

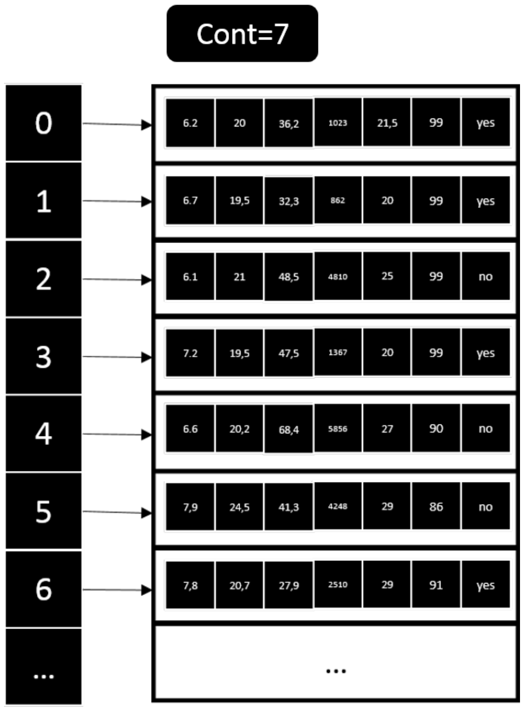

Cont=7
0
1
2
3
4
5
6
...
6.2 20 36,2 1023 21,5 99 yes
6.7 19,5 32,3 862 20 99 yes
6.1 21 48,5 4810 25 99 no
7.2 19,5 47,5 1367 20 99 yes
6.6 20,2 68,4 5856 27 90 no
7,9 24,5 41,3 4248 29 86 no
7,8 20,7 27,9 2510 29 91 yes
...

Figure 1: Hash Table of objects form the class node, which contains every physicochemical characteristic of the data set and the label, which says if the plant has or not leaf rust

4.1 Operations

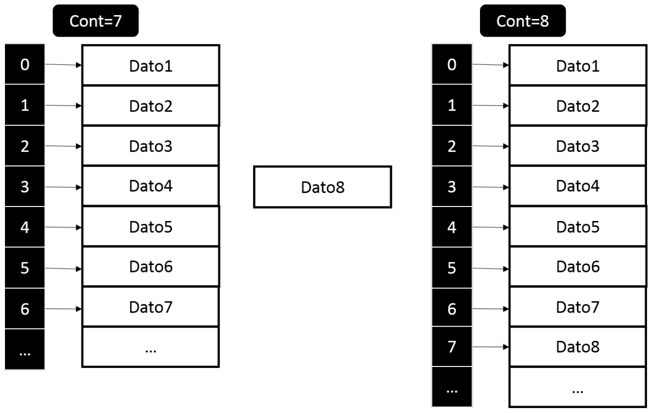


Figure 2: Representation of the adding function

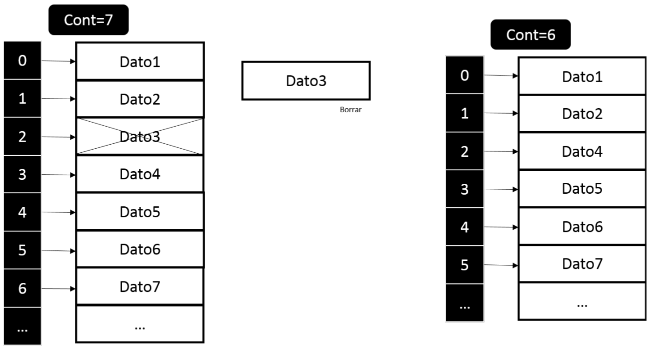


Figure 3: Representation of the deleting function

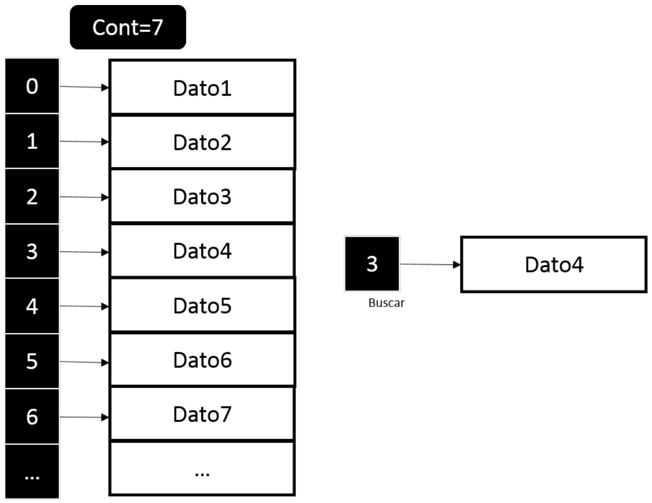


Figure 4: Representation of the searching function

4.2 Why this data structure?

We decided to work with this data structure because it let us add, search and delete components from the structure with a complexity of O(1) and because it can help us to make a good use of a decision tree for predicting the presence of the coffee leaf rust on the plant.

4.3 Complexity

Method	Complexity
Reading a Data Set	O(n)
Adding	O(1)
Searching	O(1)
Deleting	O(n)

Table 1: Complexity of methods

4.4 Execution time

Method	Best time	Worst Time	Average Time
Adding	0.000244140625 ms	0.0126953125 ms	0.000739746 ms
Searching	0.000244140625 ms	0.0009765625 ms	0.0004443359375 ms
Deleting	0.072509765625 ms	0.126953125 ms	0.0004443359375 ms

Table 2: Execution time of methods

4.5 Data structure storage

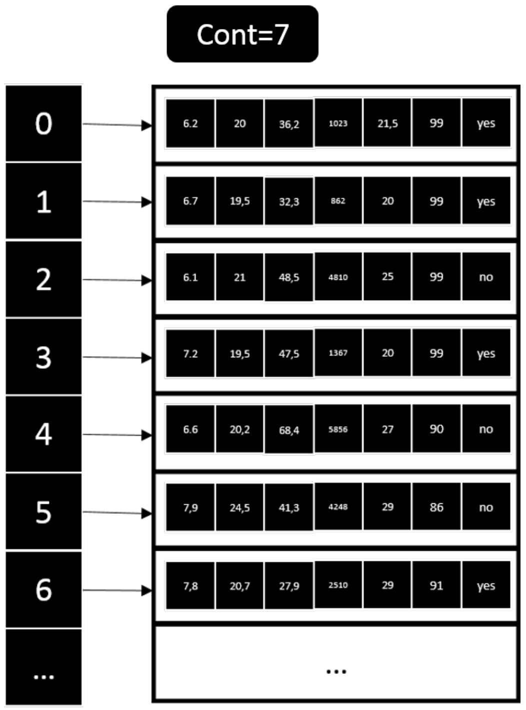
	Dataset1 (data_set.csv)	Dataset2 (data_set_test.csv)	Dataset3 (data_set_train.csv)
Data Structure	0.3MB	0.261 MB	0.273 MB

Table 3: Data structure storage

5. DATA STRUCTURE (LAST IMPLEMENTATION)

Our final implementation was the same as the first one (so the figures are the same) but adding the CART algorithm to build a decision tree (at the end of the document you can see the decision tree).

Regarding complexity, it is the same as the first complexity, adding the methods for the building of the CART tree.



Cont=7
0
1
2
3
4
5
6
...
6.2 | 20 | 36,2 | 1023 | 21,5 | 99 | yes
6.7 | 19,5 | 32,3 | 862 | 20 | 99 | yes
6.1 | 21 | 48,5 | 4810 | 25 | 99 | no
7.2 | 19,5 | 47,5 | 1367 | 20 | 99 | yes
6.6 | 20,2 | 68,4 | 5856 | 27 | 90 | no
7,9 | 24,5 | 41,3 | 4248 | 29 | 86 | no
7,8 | 20,7 | 27,9 | 2510 | 29 | 91 | yes
...

Figure 5: Hash Table of objects form the class node, which contains every physicochemical characteristic of the data set and the label, which says if the plant has or not leaf rust

5.1 Operations

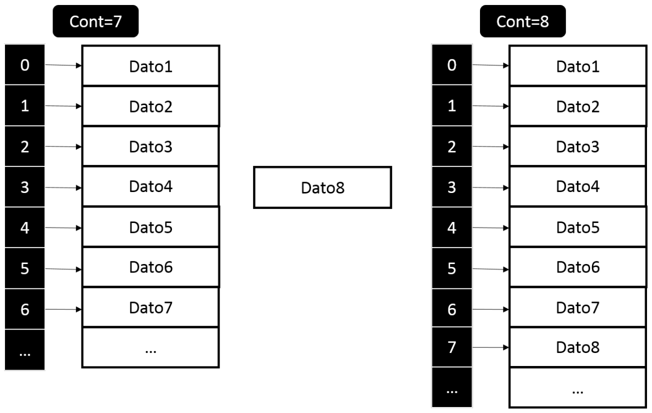


Figure 6: Representation of the adding function

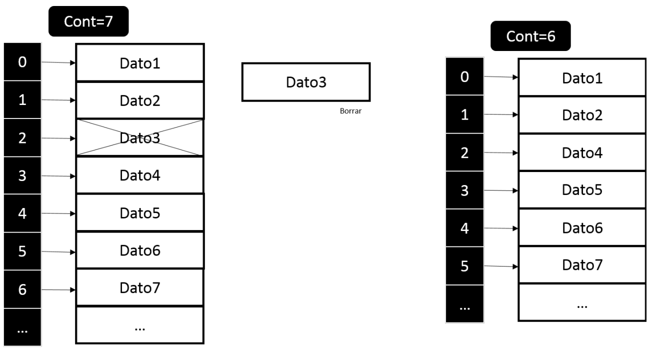


Figure 7: Representation of the deleting function

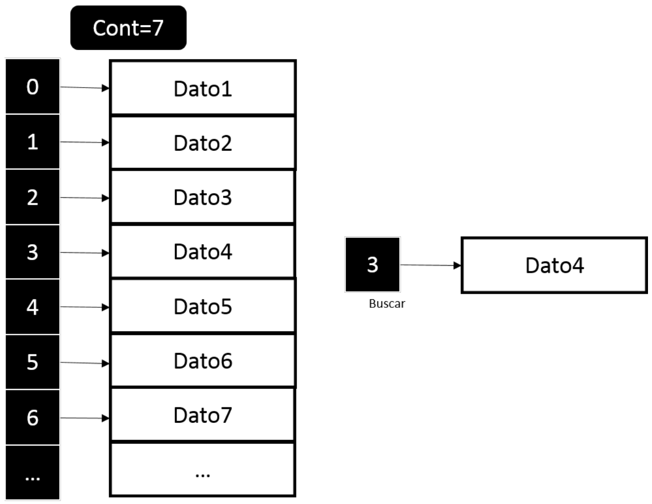


Figure 8: Representation of the searching function

5.2 Why this data structure?

We decided to work with this data structure because it let us add, search and delete components from the structure with a complexity of O(1) and because it can help us to make a good use of a decision tree for predicting the presence of the coffee leaf rust on the plant.

5.3 Complexity

Method	Complexity
Reading a Data Set	O(n*m)
Adding	O(1)
Searching	O(1)
Deleting	O(n)
Building Tree	$O(2^m * n)$
Find Best Split	O(m*n)
Deciding	O(m)

Table 4: Complexity of methods

With

n: number of Instances

m: number of Columns

5.4 Execution time

Method	Best time	Worst Time	Average Time
Adding	0.0000019073486328 ms	0.000042676925659 ms	0.00001263494 ms
Searching	0.001220703125 ms	0.00634765625 ms	0.00231917 ms
Deleting	0.021728515625 ms	0.021728515625 ms	0.0110349 ms
CART algorithm	12.88599 s	14.18105 s	13.0192 s

Table 5: Final implementation execution time.

5.5 Memory usage

According to sys library of pyhton, here is the memory usage of the data structure:

	Dataset1 (data_set_balanced.csv)
Data Structure storage	0.284019 MB

Table 6: Final implementation memory usage.

6. CONCLUSIONS

To sum up, we can see two important details, first, the coffee leaf rust is a problem nowadays which affects coffee productors countries in their economies because of the damage it makes, thus, it is important to find out a way to predict and reduce this issue. Second, there are many algorithms which allows create decision trees and this way to solve problems is very efficient and organized; therefore, they are useful. Then, this solution based on a CART algorithm is a good option to get close to a final predict solution.

About our results, they were good, because we got an accuracy of 72.15% with the implementation of decision tree. Moreover, we designed a data structure with a decent big O complexity, execution times and memory usage; and these are important details to consider now to choose a data structure. On the other hand, our first and final implementation differs only in the second one has the CART algorithm. Also, we have problems to how to choose the way to store the data in order to waste low memory and the complexity were acceptable, to get that we decided a data structure based on Python dictionary. Implementation of the CART algorithm also was hard, but after read, see codes and understand the idea, we constructed de decision tree for our problem.

6.1 Future works

Finally, this is not a final solution of the coffee leaf rust problem, although the accuracy of the decision tree were good, it could be better and, also is important to see the results and analyze them in the reality and research in a physicochemical way why that factors influences in that issue.

ACKNOWLEDGEMENTS

We thank for assistance with showing where to look for information and ideas to build a decision tree to Paulina Pérez Garcés, student of math engineering in EAFIT.

DECISION TREE BY CART ALGORITHM

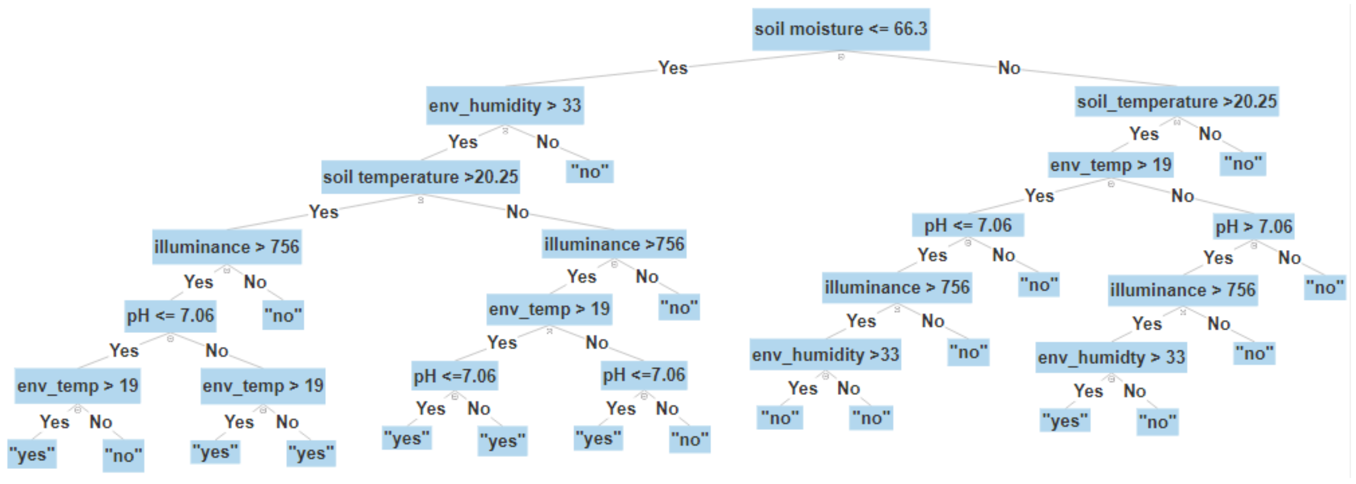


Figure 9: Decision tree by CART algorithm

REFERENCES

1. Sani, P., Rai, S., Jain, A.K. *International Journal of Compter, 1*(1). 31-33. From Banasthali University: http://airccse.org/journal/ijcax/papers/1114ijcax04.pdf
2. Korting, T.S. C4.5 algorithm and Multivariate Decision Trees. Researchgate.net. From National Institute for Space Research: https://www.researchgate.net/profile/Thales_Koerting/publication/267945462_C45_algorithm_and_Multivariate_Decision_Trees/links/5475b99b0cf29afed612b236.pdf
3. Yobero, C. Determining Creditworthiness for Loan Applications Using C5.0 Decision Trees: https://rpubs.com/cyobero/C50
4. Revathy, R., Lawrance, R. *International Journal of Innovative Research in Computer and Communication Engineering 5(1).* 50-58 From: http://www.ijircce.com/upload/2017/ncirest/6_Comparative%20analysis%20of%20C45%20and%20C5.pdf
5. Brownlee, J. Classification And Regression Trees for Machine Learning: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/