# A TRANSPARENT, SIMPLE AI TOOL FOR CONSTRUCTING EFFICIENT AND ROBUST FAST AND FRUGAL TREES FOR CLASSIFICATION UNDER RISK

Laura Martignon[1], Joachim Engel[1], and Tim Erickson[2]
[1]Department of Mathematics and Computer Science, Ludwigsburg University of Education
[2]Eeps Media, California, USA
martignon@ph-ludwigsburg.de

*Artificial Intelligence (AI) has produced extremely efficient and effective classification and decision "machines" that learn from given data sets and generalize well to unknown data. These are mostly celebrated tools produced by methodologies of machine learning. There is a drawback, though, namely their lack of transparency in construction. Agents often ignore the construction steps and use them as black-box algorithms. We exhibit simple and transparent steps for creating robust and yet simple heuristics for classification based on the AI tool ARBOR. We also claim that these transparent classifiers compete well against powerful machines, especially when training sets are small.*

## BACKGROUND

Humans have had to make decisions under risk and uncertainty through all eras of their history. Nevertheless, the concept of risk is only about 800 years old and emerged as insurance became popular. It developed in parallel with the first official insurance contracts between shipping companies transporting goods and wares, and the craftsmen in Renaissance Italy who produced them (Scheller, 2017). The mathematical tools for risk analysis were developed during the late Renaissance and the Enlightenment, and the first analytical treatment of risks and so-called life contingencies was written by the British mathematician Augustus de Morgan (1838). Risk became an important area of application of probability theory and mathematical analysis, especially used in finance and economics. However, basic elements of this analysis have acquired relevance for communicating about problems in many basic domains of life such as health and the environment. As an example of the urgency for higher risk literacy among citizens, we cite the COVID-19 pandemic and the difficult discussions in the media about the benefits of tests and vaccines (see also Kaiser & Martignon, 2022). We claim that basic risk literacy should become an educational topic in school. This claim, and the suggestions on how this literacy may become part of the curriculum even in primary school, are reported at length by Martignon and Hoffrage (2019). Here we summarize the four basic components or levels of risk literacy as proposed by Martignon and Hoffrage. These are illustrated in Figure 1.

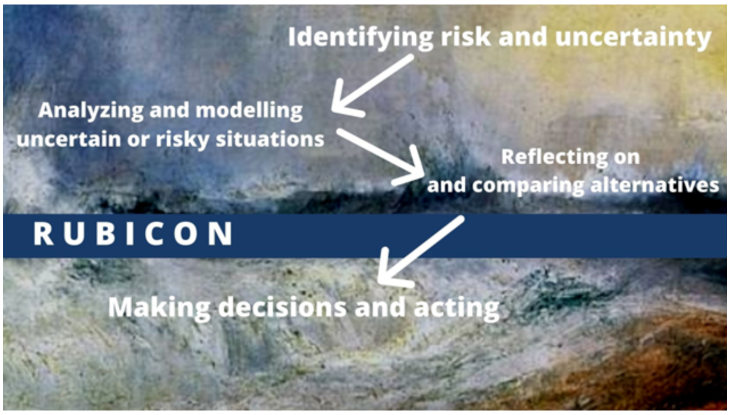

Figure 1. The four levels of risk literacy

Analyzing and modelling risks is possible today owing to the tools of probability and statistics for estimating the reliability of features of risky situations. These tools, developed during the Enlightenment, are used for estimating sensitivities and predictive values of features or tests, and symptoms. Yet, as is well known, Bayes' theorem for one feature only is seldom enough for a classification that leads to a decision in a situation of high risk. Thus, more complex evidential reasoning under uncertainty becomes necessary. The pandemic has highlighted this necessity of

In S. A. Peters, L. Zapata-Cardona, F. Bonafini, & A. Fan (Eds.), *Bridging the Gap: Empowering & Educating Today's Learners in Statistics. Proceedings of the 11th International Conference on Teaching Statistics (ICOTS11 2022), Rosario, Argentina.* International Association for Statistical Education. iase-web.org ©2022 ISI/IASE

evidential reasoning under uncertainty. During the past two years, doctors and medical researchers around the world have had to analyze, combine, and model features/cues of tests of different kinds. Doctors and hospital assistants have had to assess the predictive value or diagnostic power of a combination of such features and decide accordingly, whether a patient with COVID-19 symptoms should be assigned to a hospital bed. Figure 2 on the left displays a fast and frugal decision tree that crystallized from statements from interviews with 17 doctors/hospital assistants in the city of Tübingen (conducted by the first author).

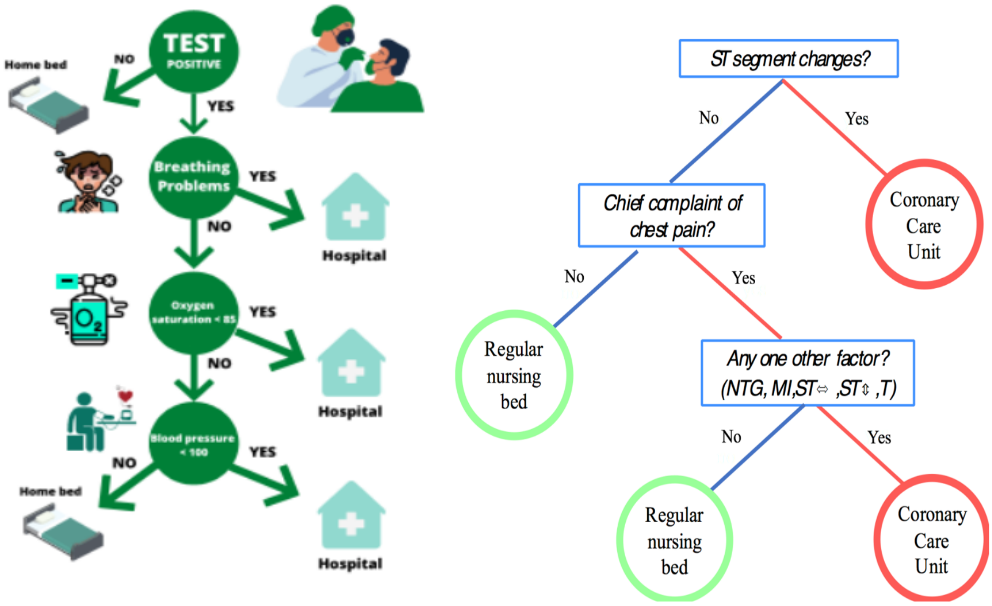

Figure 2. On the left, the reader sees a fast and frugal tree used by doctors in Tübingen to assign a patient to a hospital or home bed during the first phase of COVID-19. The tree depicted on the right is a fast-and-frugal tree that helps emergency room doctors decide whether to send a patient to a regular nursing bed or to the coronary care unit (Green & Mehr, 1997).

What matters in these cases is making a quick decision based on a quick diagnosis by means of a few cues. The predictive value for an oxygen saturation of less than 90 (or, in some cases, less than 85) for high COVID-19 risk has been assessed collectively based on shared knowledge and shared expertise. Low blood pressure is also considered predictive of imminent high risk. Here three features are combined for judgment.

Assessing the reliability of one single cue is based on a straightforward application of Bayes' theorem. If a physician has to establish whether a patient suffers from a disease, D, based on just one piece of evidence, which could be, for instance, a symptom or a test result, she will make use of three elements:

1. The prior probability, or *base rate*, of the disease. If D+ denotes the actual presence of the disease, the base rate or prevalence of the disease in the population can be denoted by P(D+).
2. The *sensitivity*, or true positive rate of the evidence. If the piece of evidence is a test T, with a positive result denoted by T+, then the sensitivity, or probability of a positive test if the patient has the disease, is denoted by $P(T+|D+)$.
3. The *specificity*, or true negative rate of the evidence. If a negative result of the test is denoted by T- and absence of the disease is denoted by D-, then the specificity or probability of a negative result if the patient does not have the disease is denoted by $P(T-|D-)$. Assume also, that the doctor knows the sensitivity of the test, $P(T+|D+)$, and its specificity, $P(T-|D-)$.

Assuming that the doctor knows the base rate, sensitivity, and specificity, the probability that the patient has the disease can be found by applying Bayes' theorem:

$$P(D+|T+) = \frac{P(D+)P(T+|D+)}{P(D+)P(T+|D+) + P(D-)P(T+|D-)}$$

We will not go into the topic of the difficulties people, sometimes even doctors, have with this formula and the progress in fostering intuitions about this form of Bayesian reasoning by means of good representations. The paper by Kaiser and Martignon (2022), presented at ICOTS 11 in Rosario, offers a review of different representational approaches used successfully with students. The concern here, as mentioned above, is that just one feature, be it a test or a symptom or some other characteristic, is seldom enough for making crucial decisions in the health domain. The problem is then one of extending the approach to many cues. Bayesian networks, neural networks, Classification and Regression Trees (CART), random forests, and regression models are important methods for combining information from many cues. These models are all complex in construction, if not in execution. Their construction steps often include optimizing strategies of some kind, or the use of selection criteria, which require subtle calculations. They may be highly accurate when fitting known data and robust in generalization, but they often lack transparency, be it in construction or in execution.

## FAST AND FRUGAL TREES

A fast-and-frugal tree is a tree-structured representation for making a classification on the basis of several cues. One begins, as in any classification tree, at the root and traces the tree until a leaf is reached that indicates the classification. A fast-and-frugal tree has a single exit at every level before the last one, where it has two. (See Figure 2 for examples.) The cues are ordered by means of simple ranking criteria. Fast-and-frugal trees are implemented step by step with a limited memory load and can be set up and executed by the unaided mind, requiring, at most, paper and pencil.

A fast-and-frugal tree does not in general classify optimally. It rather "satisfices," producing good enough solutions with reasonable cognitive effort. For generalizing, that is learning structure from training sets and extrapolating to test sets, fast-and-frugal trees have proven surprisingly robust. The predictive accuracy and the robustness of the fast-and-frugal tree has been amply demonstrated (e.g., Martignon & Laskey 2019; Woike et al., 2015). Fast-and-frugal trees have a simple characterization:

> Theorem: Given binary cues with values of 0 and 1, a fast and frugal tree based on these cues is characterized by the existence of a unique cue profile of 0's and 1's that operates as a splitting profile of the classification procedure. Any item with a profile lexicographically lower than the splitting profile will be classified in one category, while the remaining items will be placed in the other category. (For details, see Martignon et al., 2008.)

## TREE CONSTRUCTION USING ARBOR

ARBOR is an educational Artificial Intelligence (AI) tool for constructing trees based on basic measures of the features involved. More precisely, the only computations required for constructing these trees are those for obtaining predictive values and sensitivities out of the nodes in the natural frequency trees. These are paper and pencil calculations, yet they have to be performed without errors and quickly.

Basically, ARBOR invites the user to play around with features, placing them sequentially, one after the other. At each step ARBOR emits precisely those measures that are relevant for the user's choice of an adequate tree. ARBOR is found easily in the third authors' large webpage composed of a series of different plugins for the Common Online Data Analysis Platform (CODAP) at https://codap.xyz. CODAP is a flexible, free, open-source educational data analysis platform. By opening the webpage and scrolling down to the section for "plugins and data with trees," the reader has access to more than one plugin for tree construction based on ARBOR. A direct link is https://codap.xyz/plugins/arbor/.

Here we present instructions and some explanations about using the CODAP ARBOR plugin to create classification trees using the dataset about heart attacks found in Green and Mehr (1997); this data set and the corresponding tree construction is also found under "plugins and data with trees." For brevity, we do not give instructions on how to use CODAP; if the user is new to CODAP, and wants to know more, she can find help at https://codap.concord.org or by using the help button in the upper right of the CODAP window.

When the user clicks the link at MI tree, the browser will load CODAP and a saved document. He will see a rectangular area labeled diagnostic tree; a table of data labeled MIGreenMehr, and another currently empty data table called Classification Tree Records. There is also a “minimized” graph that the user can experiment with, but we will not refer to it in this explanation.

The MIGreenMehr data table has information for about 89 patients, each with four attributes.

- MI: did the patient have a heart attack?
- Pain: did the patient complain of chest pain?
- STelev: is the electrocardiogram (ECG) ST segment elevated?
- oneOf: did the patient have any of a set of four other symptoms?

The idea is to use these data as a training sample to create a classification tree to predict, given pain, STelev, and oneOf, who will have a heart attack (myocardial infarction, MI). That is, the user will assign MI as the outcome variable, and the other three will be predictors.

The simplest possible tree is easily set up. It has a single node, which is to be labeled as “yes” to indicate that we predict that all of the patients will have an MI. Below the tree, the user can see how well the tree performs: there are 15 “true positives” (TP) and 74 “false positives” (FP). What does this mean? This means that all 15 of the heart-attack victims were identified as at risk for heart attack. But this also means that all 74 of the other patients were also identified as at risk. In a diagnostic setting, the user will have deployed costly resources on patients that might not need it.

The typical measures of these two competing outcomes are:

- *Sensitivity* (sens): the proportion of “positive” cases that we identify as positive. For this simple tree, the sensitivity is 1.00, the best possible value.
- *Positive predictive value* (ppv): the fraction of all positive diagnoses that are in fact positive. For our tree, the ppv is 15 out of 89, or about 0.17.

The user can now begin by taking each of the features or predictors individually. One possibility is to start with the predictor pain. Below are the instructions for the user.

- Drag the blue pain label from below the tree and drop it on the single node. If nothing happens, try again! The tree will update to produce something like the left-hand illustration in Figure 3.
- The gray circles are our diagnoses for each value of pain. Click on them to change them; identify pain = “yes” with a positive diagnosis and “no” with a negative diagnosis, which only makes sense. The tree should now look like the right-hand illustration in Figure 3.
- Below the tree, click the button labeled Emit Data. A row of data will appear in the Classification Tree Records table.

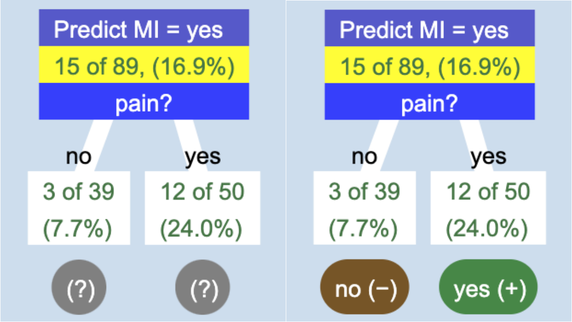

Figure 3. The tree for one feature, namely “Pain?”

The user should then take a moment to understand the tree display. The “yes” box, for example, tells us that 50 patients experienced chest pain, and 12 out of those 50 (or 24%) got heart attacks. We identified them as being at risk (the green yes (+) label), a “positive” diagnosis. In the other branch, out of the 39 people we diagnosed as not at risk, three got heart attacks anyway.

How well does this tree perform? According to our two measures, that 24% is, by definition, the positive predictive value, ppv, an improvement over the “null” tree from before. But the sensitivity (sens) is now 12 out of 15, or 0.80, a step down from our earlier, perfect score.

In the Classification Tree Records table, CODAP has calculated values for ppv and sens. The table also records other metrics for the tree, such as the base rate (base), the number of true positives (TP), and the depth of the tree. “Hover” over any column heading to see its definition.

We can do the same for the other two predictors, STelev and oneOf. Here are the instructions.

- Just drag the corresponding label from the area below the tree and drop it in the “root” node. The new variable will replace pain.
- Specify, for each “leaf” node—the gray circles—which diagnosis corresponds to that new node. (Of course, a “yes” value for the symptom should give a “yes” value for the diagnosis.)
- Click Emit Data again to record the performance of the tree in the second table.
- Do that for each of the two additional variables.
- Now it is time to “grow the tree.” The user is not limited to a single cue; after adding one variable, he can drag a second variable to any node in the tree to expand it. But which variable should we start with?

In this scheme for making a tree, we want to choose the “best” variable for our first branching. The ppv measure, i.e., can be used to determine the best one. In that case, STelev is the best, with a ppv of 0.39. Pain is next, followed by oneOf. The instructions now can be:

- Restore the tree to the state where there is one branching, using STelev.
- Now drop pain onto the “no” branch of the tree, and assign the leaves. We now have a tree like the left-hand illustration in Figure 4.
- Next, drop oneOf onto the node corresponding to STelev = no and pain = yes. When you assign diagnoses to the leaves, you get the tree on the right in Figure 4.
- Finally, click Emit Data to record the performance of this tree. You’ll see that the tree’s ppv is 0.30 and its sensitivity is 1.00.

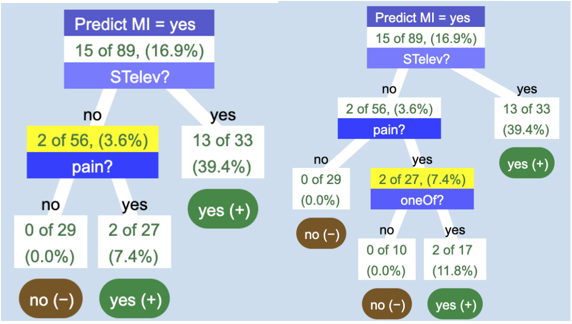

Figure 4. The left tree depicts two cues, while the right tree was grown by means of a third cue

This procedure completes our tree. The process a doctor might use now to proceed could look like this:

- The first question is whether the patient has an elevated ST segment on the ECG. If so, we send the patient to the cardiac care unit.
- If not, we ask if the patient has chest pain as a main symptom. If they have NO chest pain, the patient does not get special treatment; after all, none of the patients in that category—normal ST segment, no chest pain—got heart attacks.
- However, if that patient does have chest pain, we ask whether they have one of the other four symptoms. If they do, they go to cardiac care. If not, they get no special treatment.

The overall performance of the tree is quite good, with a positive predictive value of 0.30, that is, 30% of our cardiac care patients might get heart attacks. In addition, our sensitivity is perfect: none of the patients we are treating normally got the MI. In effect, we traded some ppv in order to “catch” the two heart attacks among patients with normal ST segments.

Observe that if one clicks on a row in the Classification Tree Records table, the tree will transform into the tree corresponding to the row. Therefore, one does not have to remember which row is which or reconstruct a complex tree. If one has emitted data, it is automatically saved.

Suppose one wants to use specificity as a construction criterion. Is this possible? The answer is affirmative: one can generate a new column in the second table and give it a formula. One can thus create any measure one considers relevant.

The short set of instructions is specific to these data, but the tree tool is quite flexible. There is a help tab in the plugin, with many more instructions. Fast and frugal trees can be constructed, for instance, by choosing at each step the cue with highest overall predictive value, or by alternating between positive and negative predictive value (Martignon et al., 2008). By design the system presents the simplest possible tree set up. It has a single node, and it has been labeled "yes," indicating the prediction that all of the patients will have an MI. Below the tree, it is possible to see how well the tree performs: there are 15 "true positives" (TP) and 74 "false positives" (FP). This is a fundamental feature of the construction.

First pilot studies with students at a recent Summer Institute organized by the university of Milano in Como, Italy, have revealed that the instructions can be easily followed and implemented. More studies are planned.

## CONCLUSIONS

Standardized classification methods have gained importance in the biomedical diagnostic field. Lötsch, Kringel, and Ultsch (2022) have amply described their use for the support of medical diagnosis. One problematic resulting effect, on which these authors comment extensively, has been that the traditional doctor–patient relationship based on trust built through the years is giving place to a less transparent interaction, in which the doctor communicates diagnostic results obtained by such AI algorithms. Fast-and-Frugal trees, which we have illustrated here, are listed also by the authors as a transparent, simple alternative.

They are fast and frugal but, as has also been shown by means of comparison studies (Woike et al., 2017), they are robust in generalization and do not have to trade accuracy for simplicity.

The intention of this paper is educational and concerned with risk literacy: when more than one feature is necessary for judgments and decisions under risk, the simple trees described here become handy and easily grasped.

## REFERENCES

De Morgan, A. (1838). *An essay on probabilities, and their application to life contingencies and insurance offices.* Longmans.

Green, L., & Mehr, D. R. (1997). What alters physicians‘ decisions to admit to the coronary care unit? *The Journal of Family Practice, 45*(3), 219–226.

Kaiser, A., & Martignon, L. (2022). Boosting students' shared knowledge on basics of statistics and probabilities for understanding critical facts about COVID-19. In S. A. Peters, L. Zapata-Cardona, F. Bonafini, & A. Fan (Eds.), *Bridging the gap: Empowering and educating today*'s *learners in statistics. Proceedings of the Eleventh International Conference on Teaching Statistics (ICOTS11).* ISI/IASE.

Lötsch, J., Kringel, D. & Ultsch, A. (2022). Explainable artificial intelligence (XAI) in biomedicine: Making AI decisions trustworthy for physicians and patients. *Biomedinformatics, 2*(1), 1–17. https://doi.org/10.3390/biomedinformatics2010001

Martignon, L., & Hoffrage, U. (2019). *Wer wagt, gewinnt? Wie sie die risikokompetenz von kindern und jugendlichen fördern können* [Who dares wins? How you can promote the risk competence of children and young people]. Hogrefe.

Martignon, L., Katsikopoulos, K., & Woike, J. (2008). Categorization with limited resources: A family of simple heuristics. *Journal of Mathematical Psychology, 52*(6), 352–361. https://psycnet.apa.org/doi/10.1016/j.jmp.2008.04.003

Martignon, L., & Laskey, K. (2019). Statistical literacy for classification under risk: An educational perspective. *AStA Wirtschafts- und Sozialstatistisches Archiv*, *13,* 269–278. https://doi.org/10.1007/s11943-019-00259-3

Scheller, B. (2017). *Fortuna maris—and how to handle her. Strategies to cope with the dangers of the sea in Renaissance Italy* [video file]. https://www.fau.tv/clip/id/8890

Woike, J. K., Hoffrage, U., & Martignon, L. (2017). Integrating and testing natural frequencies, naïve Bayes, and fast-and-frugal trees. *Decision, 4*(4), 234–260. https://doi.org/10.1037/dec0000086