# Demystifying decision forests

To build and apply a predictive model, a popular technique is called \*decision forest\*. It popularity is due to its versatility and minimality of data preparation requirements.

This article explains the technique without requiring any technical background. By the end of this article, you will get these takeaways

* Explanation of the technique
* Interpretation of the results
* Analogies with the human problem-solving approach

To demonstrate this, we use as an example a mock use-case based on simulated data. Please be aware that the assumptions behind the example are not necessarily related to reality.

A bank gives loans to its customer. The goal is to understand what customers are trustworthy enough to receive a loan. The bank can use some information related to the customer such as

* \*\*Credit score\*\*: score measuring the “reliability” of the customer, assessed by a third party
* \*\*House age\*\*: age of the house they’re purchasing
* \*\*Years of employment\*\*
* \*\*Credit card debt\*\*: Debt on the credit card
* \*\*House year\*\*: Year in which they purchased the house

The bank collected historical data about its previous customers. Besides, it traced what customers defaulted via a \*\*default\*\* column. For showing purposes, half of the previous customer defaulted.

| Credit score | House age | Years of employment | Credit card debt | House year

|---|---|---|---|---|

708|21|5|4190|2002|0|

649|20|3|2151|2003|0|

725|23|3|3603|2005|0|

785|24|4|5534|2009|0|

649|14|6|5243|2002|0|

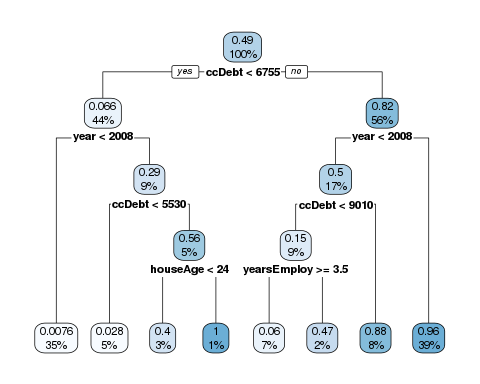
734|21|0|3441|2008|0|

Via a statistical metric, we can identify the most default-related information, e.g. the credit card debt. Setting a threshold, we can divide the bank customers into two groups.

Then, for each of the two groups, we repeat the process of finding the most relevant feature and we split the customers into two sub-groups.

A popular algorithm, called decision tree, works exactly in this way. Ultimately, the tree divides the customers into eights groups for which we compute the number of customers and default percentage.

The following chart shows the tree.



At each split of the tree, the first number is a score representing the portion of customer failing. Since 49% of the customer had a default, the initial score is 0.49. Also, the colour of the boxes gets darker as the score gets higher. The second number represents the percentage of customers belonging to the group. At the beginning, it’s 100%. Then, at each level, it shows the proportion of customers into each split.

Not surprisingly, the most relevant indicator is the debt on the credit card. 6.6% of the customers having a debt below 6755 experienced a failure, opposed to 82% of the customers having a debt equal to or above 6755.

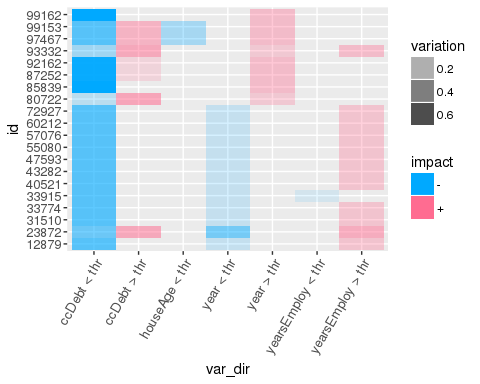
The tree visualization shows in a simple way how the model works. Having a new customer, the bank can estimate its propensity of not paying off the debt. How does it work? Not having any information about a customer, its probability of failing is the percentage of failures in the past. In this case, it’s 49%. Then, knowing the credit card debt, we go down to the second level of the tree and the score can either increase or decrease, depending on the impact of credit card debt. The impact of knowing the credit card debt is:

* if below ccDebt < 6755, the score becomes 6.6%, so it gets decreased by 42.4%
* if above ccDebt <= 6755, the score becomes 82%, so it gets increased by 33%

Summarising, this is the impact:

* ccDebt < threshold: -0.424
* ccDebt <= threshold: +0.33

In the same way, we can measure the impact of being above or below the threshold of other features. If the same feature is being used more than once, we can simply sum up the effects. Depending on the customer, the path on the tree is different, so the features being used and their impact changes. To keep track of that, we can sum up the effects separately for each customer and visualise them through a **decision-making heatmap**. Each row corresponds to a customer and each column correspond to being above or below a threshold for a given features (e.g., ccDebt < any threshold at some point in the tree). The colour is blue if the score decreases and red if it increases, and the less transparent it is, the higher the impact.



Measuring the impact of each feature, the bank can explain the decision-making process to assign each score, given more transparency and clarity to its customers.

As shown above, a decision tree operates in a way that is like the human problem solving. Coming back to the original topic, we want to demystify decision forests. As the name suggests, a decision forest is a combination of decision trees.

A hundred people can access some information of the bank, but each person is aware of a part of the customers and of a part of their features. If we ask each of them to build a decision tree, we’ll have 100 different trees. Applying the trees on a new customer, each of them will assign a different score. To come up with a unique answer, we can compute the average score. That’s exactly how a decision forest works: we build decision trees on a random subset of the information, then we combine them. This methodology, called “ensembling”, usually improves the performance of the solution.

Since there might be several trees in the forest, it’ll be time-consuming to have a look at each tree separately. However, it’s still possible to measure the impact of each feature across all the trees. Combining them, we can build a decision-making heatmap taking account of everything. Therefore, the bank can explain the decision-making process of the entire forest.

The methodology shows by the article gives transparency and clarity about the way random forests operate. A similar approach is applicable in other contexts and it allows to give more clarity about how machine learning operates.