

# Introduction to Hierarchical Clustering decision trees and random forests

Olivier J.J. MICHEL, Florent CHATELAIN

GIPSA-Lab, UMR 5216 CNRS



See [Dendogram\\_iris\\_data\\_example.ipynb](#)

# DECISION TREES

www Sources for figures & examples

<https://scikit-learn.org/stable/>

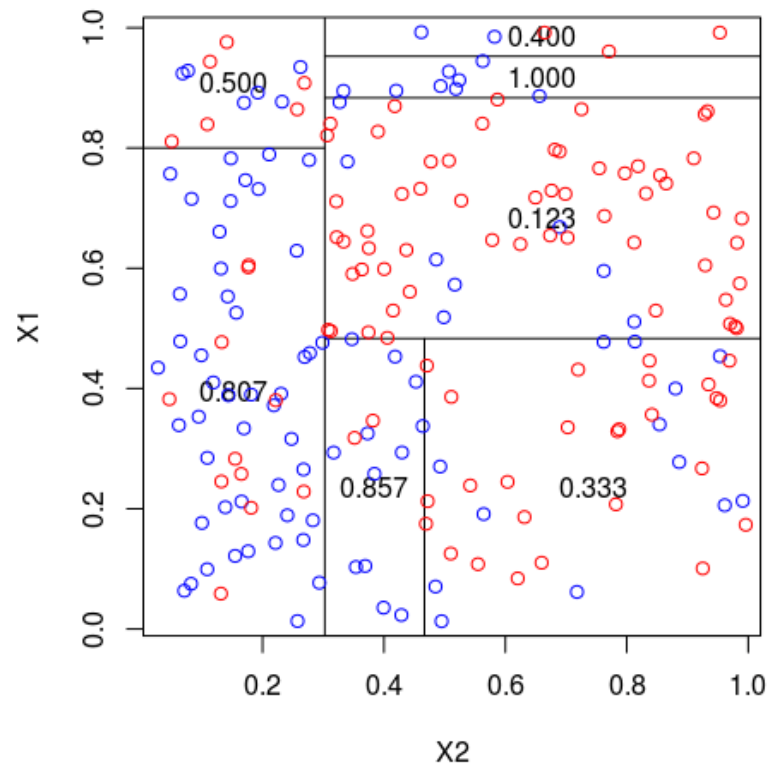
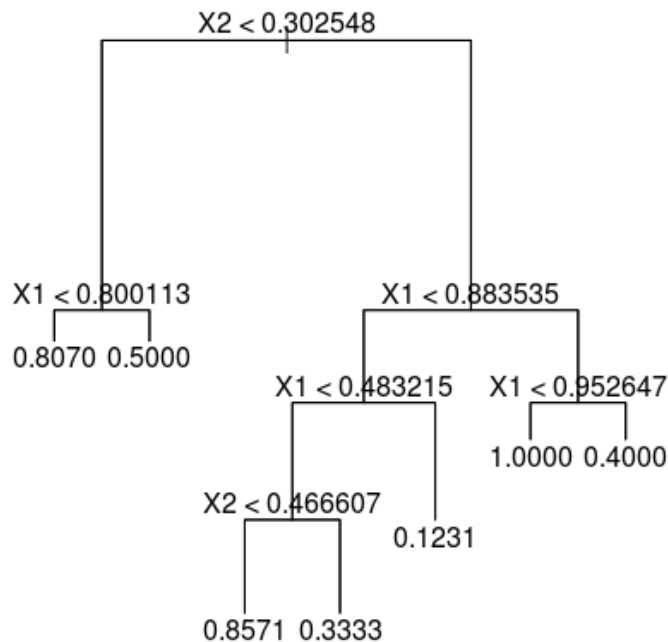
<https://dimensionless.in/introduction-to-random-forest/>

<https://gluon.mxnet.io/index.html>

<https://skymind.ai/wiki/>

**Decision tree** : deterministic data structure for modeling decision rules for a specific classification problem.

**Construction** : At each node, a single feature is selected to make separating decision. Splitting is stopped when leaf node has optimally less data points (wrt to a pre-defined criterion).



# DECISION TREES

**Goal** is to create a model that predicts the value of a target variable based on several input variables.

→ Each "interior" node of a tree corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

→ A tree can be "learned" by splitting the source set into subsets based on an attribute value test. The "recursion" is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions (greedy algo).

→ As any multiway split may be summerized as a series of binary splits, the focus will be made on binary splits.

→ Splits are obtained by choosing a test at each step that "best" splits the set of items : What does mean Choosing the "best" split  $S = S_r \cup S_l$  such that  $S_r \cap S_l = \emptyset$  for e test property  $T$  ? An obvious heuristic is to choose the query that decreases the impurity a much as possible.

## Choice of a metrics

## Gini Impurity

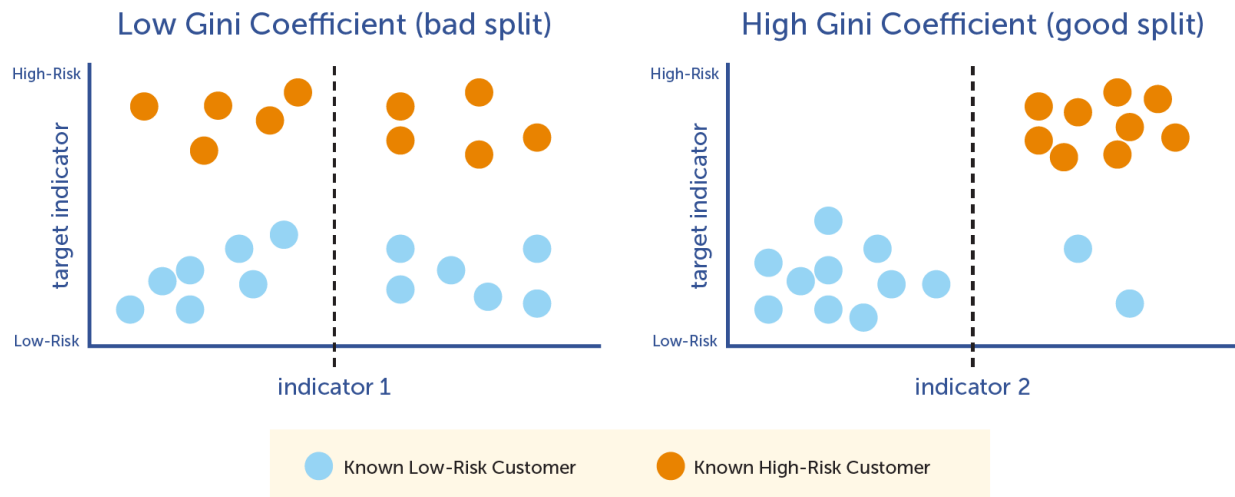
Gini Impurity of a set  $\mathbf{S}$  of cardinal  $N$  measures the probability that a randomly chosen element of the set would be incorrectly labeled if randomly labeled according to the label distribution in  $\mathbf{S}$ .

$$GI(S) = \sum_{k=1}^K P(y = k).P(y \neq k) = \sum_{k=1}^K P(y = k)[1 - P(y = k)] = 1 - \sum_{k=1}^K P^2(y = k)$$

then

$$\Delta GI(S) = GI(S) - P(S_l)GI(S_l) - (1 - P(S_l))GI(S_r)$$

where  $P(S_l) = \frac{N_l}{N}$  and  $N_l + N_r = N$



## Choice of a metrics

## Entropy (Information) Impurity

$$H(S) = - \sum_{k=1}^K P(y = k) \log_2 P(y = k)$$

the the information gain ( $IG$ ) associated to the split (or partition)  $(S_l, S_r)$  is

$$IG(S) = H(S) - H(S|(S_l, S_r)) = H(S) - P(S_l)H(S_l) - P(S_r)H(S_r)$$

where  $H(S_l)$  (resp  $H(S_r)$ ) are evaluated by using the empirical probabilities of the classes estimated from the subset  $S_l$  (resp  $S_r$ ) :

$$H(S_l) = - \sum_{k=1}^K P(y = k|X \in S_l) \log_2 P(y = k|X \in S_l)$$

## Choice of a metrics

Misclassification Impurity

$$MI(N) = 1 - \max_k P(y = k)$$

and hence the gain in MI

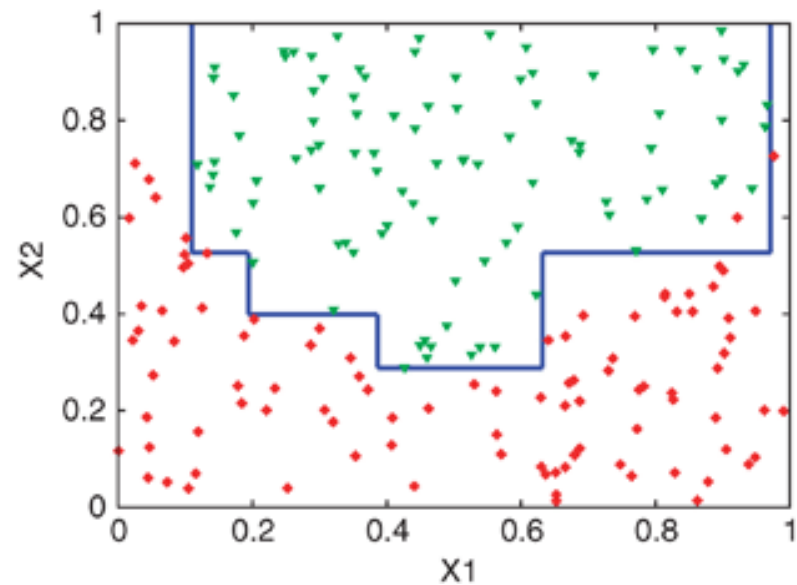
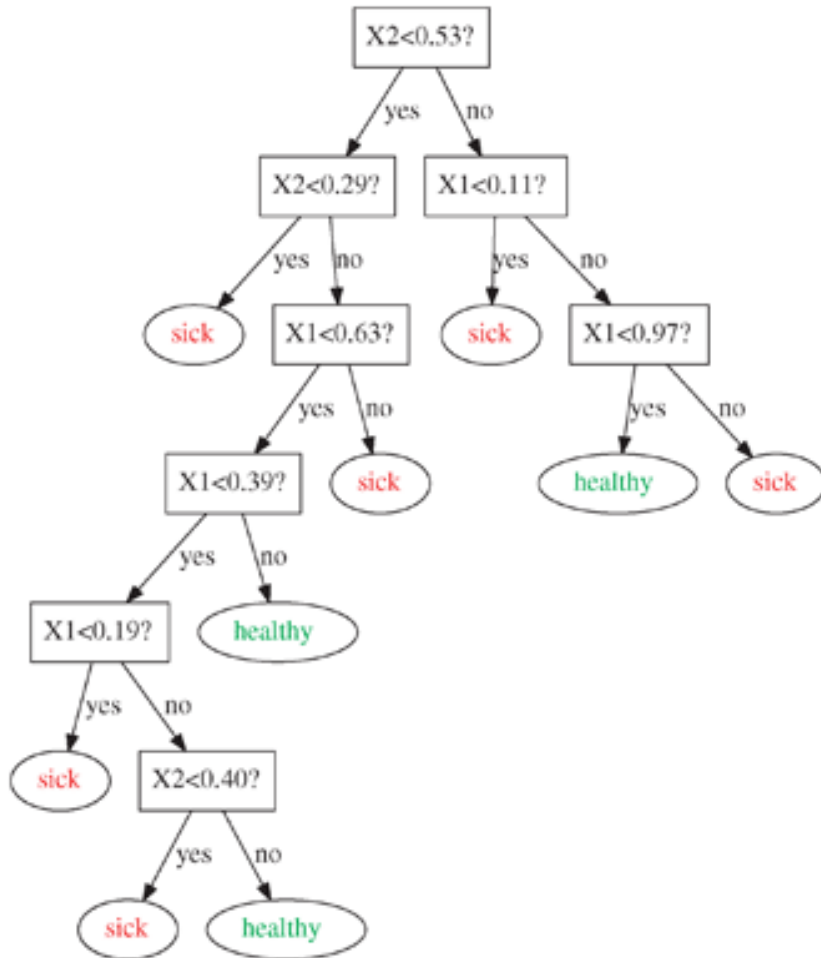
$$\begin{aligned}\Delta MI(N) &= MI(N) - P(S_l)MI(N_l) - P(S_r)MI(N_r) \\ &= MI(N) - 1 + P(S_l)\max_k P(y = k|X \in S_l) + P(S_r)\max_k P(y = k|X \in S_r)\end{aligned}$$

## Splitting strategy

Designers usually choose to split wrt a single attribute. In general for non numerical attributes, an exhaustive search over all possibilities is performed. For real values attribute, gradient method for identifying a separating hyperplane may be used; However, simple threshold on a single attribute is often preferred.

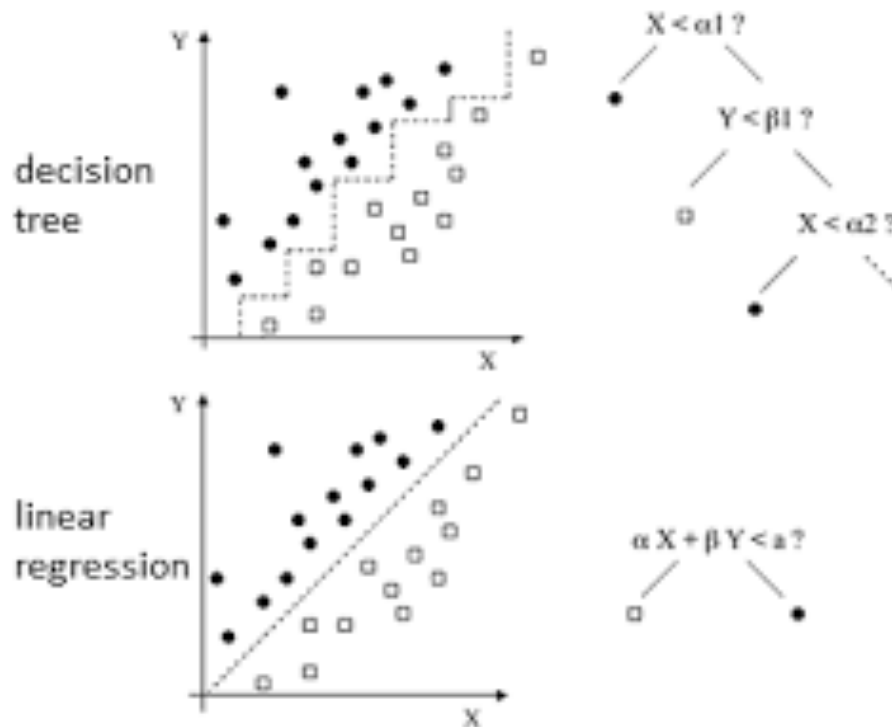
## Consequences :

Each split leads to a straight line classifying the dataset into two parts.  
Thus, the final decision boundary will consist of straight lines (or boxes).



## Consequences :

In comparison to regression, a decision tree can fit a stair case boundary to classify data.



See [TREES\\_iris\\_data\\_example.ipynb](#)



## Random forests

Random Forest consists in generating multiple small decision trees from random subsets of the data (hence the name “Random Forest”).

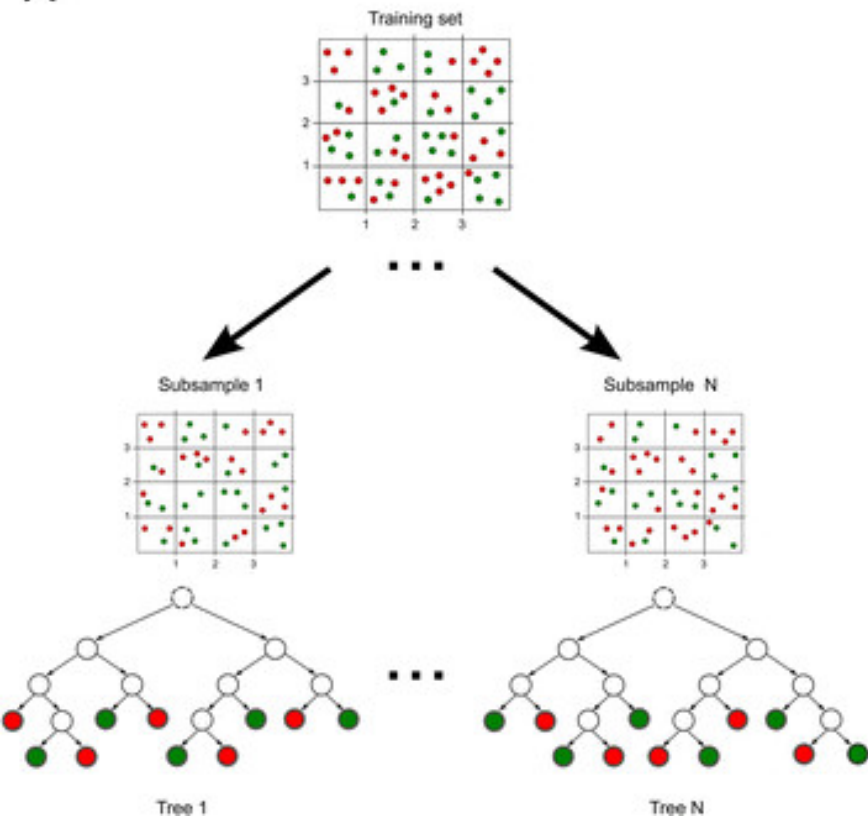
→ Each of the decision tree gives a biased classifier (as it only considers a subset of the data).

→ Each decision tree capture different trends in the data. This ensemble of trees is like a team of experts each with a little knowledge over the overall subject but thorough in their area of expertise. →

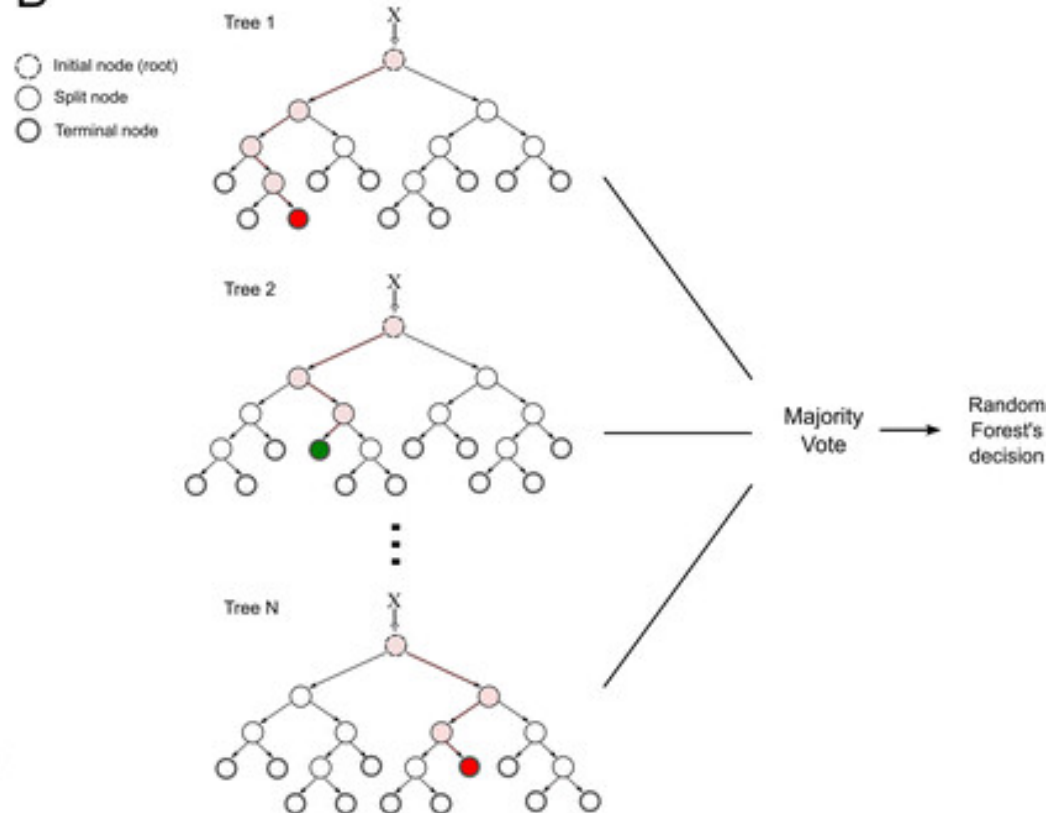
- In case of classification the majority vote is considered to classify a class.
- In case of Regression, we can use the avg. of all trees as our prediction.
- In addition to this, we can also weight some more decisive trees high relative to others by testing on the validation data.

# Random forest, illustrations :

A

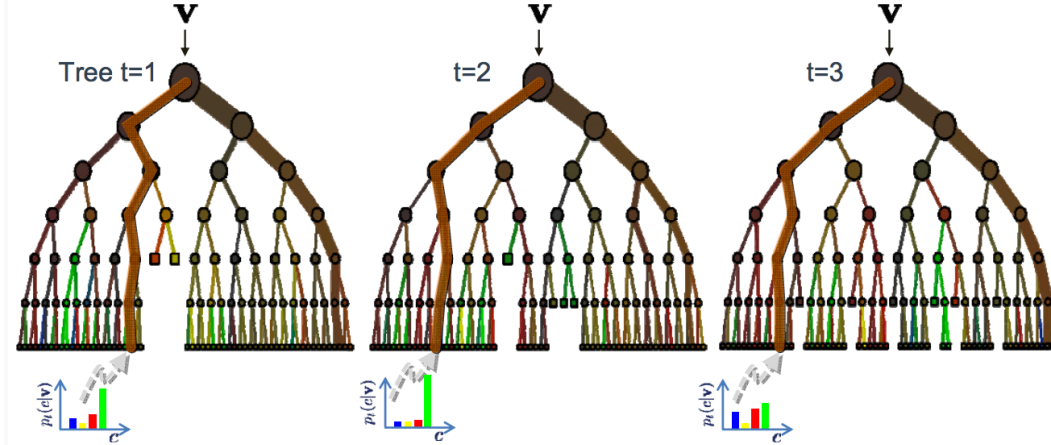


B



See [TREES\\_iris\\_data\\_example.ipynb](#)  
See [TREES\\_sonar\\_data\\_example.ipynb](#)

## Recap :



## Pros:

One of the most accurate decision models

Works well on large datasets.

Can be used to extract variable importance.

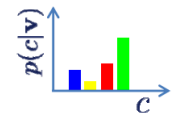
Do not require feature engineering (scaling and normalization)

Probabilities and set the threshold for classification.

Capable of generating complex decision boundaries

### The ensemble model

$$\text{Forest output probability } p(c|\mathbf{v}) = \frac{1}{T} \sum_t^T p_t(c|\mathbf{v})$$



## Cons:

Overfitting in case of noisy data.

Unlike decision trees, results may be quite difficult to interpret.

Hyperparameters need good tuning for high accuracy :

- Ntree : Nb of trees to grow in the forest
- N\_features : Nb of variables randomly sampled as candidates for each split for a given tree
- Replacement : Sampling done with or without replacement