# Introduction to machine learning
# Non Parametric Regression/Classification
# Decision Tree/ Bagging/ Random Forest/ Extra Trees

Mathilde Mougeot

ensIIE & ENS Paris-Saclay, France

2025

# Decision Trees & Ensemble methods

**1** Decision tree

**2** Regression Tree
  Model selection

**3** Ensemble methods
  Bagging
  Random Forest
  Extra Trees
  Variable Importance Measure for Ensemble Methods

**4** Classification Tree

**5** Applications

**6** Pruning and Model selection - original approach

# Outline

**1** Decision tree

**2** Regression Tree

**3** Ensemble methods

**4** Classification Tree

**5** Applications

**6** Pruning and Model selection - original approach

# Regression & Classification Trees

# Regression Tree

Boston Housing Data. The original data are $n = 506$ observations on $p = 14$ variables,

| | |
|---|---|
| medv | median value, being the target variable |
| crim | per capita crime rate by town |
| zn | proportion of residential land zoned for lots over 25,000 sq.ft |
| indus | proportion of non-retail business acres per town |
| chas | Charles River dummy variable ($= 1$ if tract bounds river; 0 otherwise) |
| nox | nitric oxides concentration (parts per 10 million) |
| rm | average number of rooms per dwelling |
| age | proportion of owner-occupied units built prior to 1940 |
| dis | weighted distances to five Boston employment centres |
| rad | index of accessibility to radial highways |
| tax | full-value property-tax rate per USD 10,000 |
| ptratio | pupil-teacher ratio by town |
| b | $1000(B - 0.63)^2$ where B is the proportion of blacks by town |
| lstat | percentage of lower status of the population |
| medv | median value of owner-occupied homes in USD 1000's |

# Boston Housing Data

**Raw data :**

| n | crim | zn | indus | chas | nox | rm | age | dis | rad | tax | ptratio | b | lstat | medv |
|---|------|----|-------|------|-----|----|-----|-----|-----|-----|---------|---|-------|------|
| 1 | 0.006 | 18 | 2.3 | 0 | 0.53 | 6.57 | 65.2 | 4.09 | 1 | 296 | 15.3 | 396.9 | 4.9 | 24.0 |
| 2 | 0.027 | 0 | 7.0 | 0 | 0.46 | 6.42 | 78.9 | 4.96 | 2 | 242 | 17.8 | 396.9 | 9.1 | 21.6 |
| 3 | 0.027 | 0 | 7.0 | 0 | 0.46 | 7.18 | 61.1 | 4.96 | 2 | 242 | 17.8 | 392.8 | 4.0 | 34.7 |
| 4 | 0.032 | 0 | 2.1 | 0 | 0.45 | 6.99 | 45.8 | 6.06 | 3 | 222 | 18.7 | 394.6 | 2.9 | 33.4 |
| 5 | 0.069 | 0 | 2.1 | 0 | 0.45 | 7.14 | 54.2 | 6.06 | 3 | 222 | 18.7 | 396.9 | 5.3 | 36.2 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

# Regression Tree. Boston Housing Data

Application : Prediction of the prize of an house in Boston given several co-variables.
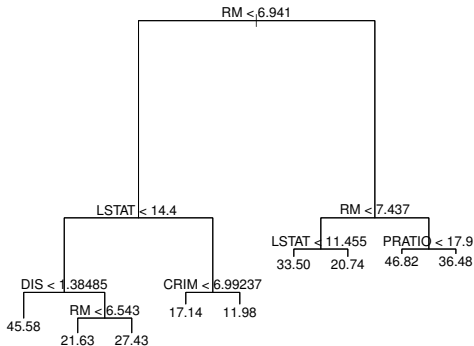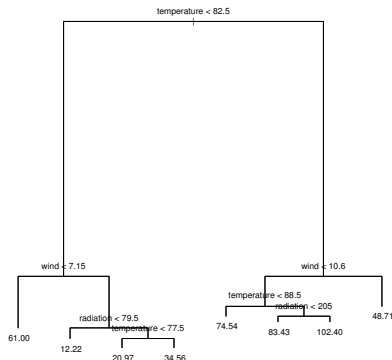
## Regression Tree.

Application : Prediction of the ozone rate given several co-variables

Target variable (Y) : ozone rate.

Co-Variables (X) : Temperature, Wind, radiation rate

# Classification Tree

Application : Coronary Heart Disease (variable CHD $\{0, 1\}$)
Co-Variables :

| | |
|---|---|
| chd | response, coronary heart disease |
| sbp | systolic blood pressure |
| tobacco | cumulative tobacco (kg) |
| ldl | low density lipoprotein cholesterol |
| adiposity | |
| famhist | family history of heart disease (Present, Absent) |
| typea | type-A behavior |
| obesity | |
| alcohol | current alcohol consumption |
| age | age at onset |

# Classification tree.

Application : Coronary Heart Disease (variable CHD $\{0, 1\}$)

## The data :

| obs | sbp | tobacco | ldl | adiposity | famhist | typea | obesity | alcohol | age | chd |
|-----|-----|---------|------|-----------|---------|-------|---------|---------|-----|-----|
| 1   | 160 | 12.00   | 5.73 | 23.11     | Present | 49    | 25.30   | 97.20   | 52  | 1   |
| 2   | 144 | 0.01    | 4.41 | 28.61     | Absent  | 55    | 28.87   | 2.06    | 63  | 1   |
| 3   | 118 | 0.08    | 3.48 | 32.28     | Present | 52    | 29.14   | 3.81    | 46  | 0   |
| 4   | 170 | 7.50    | 6.41 | 38.03     | Present | 51    | 31.99   | 24.26   | 58  | 1   |
| 5   | 134 | 13.60   | 3.50 | 27.78     | Present | 60    | 25.99   | 57.34   | 49  | 1   |
| 6   | 132 | 6.20    | 6.47 | 36.21     | Present | 62    | 30.77   | 14.14   | 45  | 0   |
| 7   | 142 | 4.05    | 3.38 | 16.20     | Absent  | 59    | 20.81   | 2.62    | 38  | 0   |
| 8   | 114 | 4.08    | 4.59 | 14.60     | Present | 62    | 23.11   | 6.72    | 58  | 1   |
| 9   | 114 | 0.00    | 3.83 | 19.40     | Present | 49    | 24.86   | 2.49    | 29  | 0   |
| 10  | 132 | 0.00    | 5.80 | 30.96     | Present | 69    | 30.11   | 0.00    | 53  | 1   |

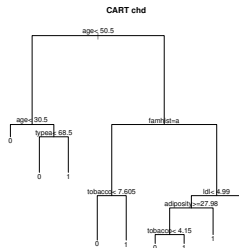## Classification tree.

Application : Coronary Heart Disease (variable CHD $\{0, 1\}$)

Decision tree :



**CART chd**
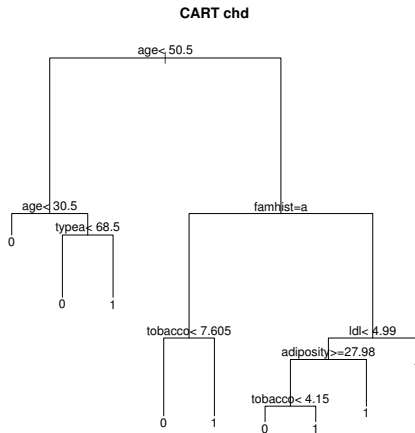
# Decision tree
## Some vocabulary



- Root : first node of the tree

- leaf : terminal node

- Rule between the root and a leaf
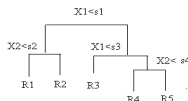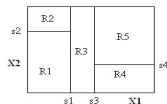
- Regions : spaces

# Decision Trees

Leo Breiman, Friedman, Olshen 1984

Decision tree is a method which splits the input space in a set of rectangularly domains, in which a **constant model** is adjusted.

$\rightarrow$ The global regression function is given by

$$f(x) = \sum_{m=1}^{M} c_m 1_{x \in \mathcal{R}_m}$$

$c_m$ is a modality for each region $\mathcal{R}_m$

# CART

**Classification And Regression Tree, Arbres de Décision**

- Y : target variable
  - Qualitative : Classification Tres
  - Quantitative : Regression Tree
  - The same approach let to study regression or classification problems

- $X^j$ : covariables, $1 \leq j \leq p$ qualitatives ou quantitatives

  CART belongs to the Non Parametric method family.

    No assumption is made on the data distribution

  $\rightarrow$ The method builds a binary tree

# Outline

# Regression trees

- The optimization problem is NP-hard.
- A greedy algorithm is used ('greedy' : algorithme glouton) :

Recursive construction :

- Recursive splits are computed regarding a set of given data (the data of the Training set),
  using

- Nodes
    - The Decision criteria at each node involves only one variable
    - and one threshold (for the selected variable)

# Definition of the regions

Regarding the type (quantitative or qualitative) of each explanatory variable :

$\rightarrow$ $X$ Quantitative, Continuous, Ordinal

| $\mathcal{R}_1$ (Left) | $\mathcal{R}_2$ (Right) |
|:---:|:---:|
| $X < S$ | $X \geq S$ |

$\rightarrow$ $X$ Qualitative. Example for 4 modalities $X \in \{g_1, g_2, g_3, g_4\}$

| $\mathcal{R}_1$ (Left) | $\mathcal{R}_2$ (Right) |
|:---:|:---:|
| $g_1$ | $g_2, g_3, g_4$ |
| $g_2$ | $g_1, g_3, g_4$ |
| $g_3$ | $g_1, g_2, g_4$ |
| $g_4$ | $g_1, g_2, g_3$ |
| $g_1, g_2$ | $g_3, g_4$ |
| $g_1, g_3$ | $g_2, g_4$ |
| $g_1, g_4$ | $g_2, g_3$ |

# Regression Trees. Construction (1/3)

The Target variable $Y$ is a quantitative variable
The $X_{j_0}$ variable is selected if it minimizes the *Deviance* (for Regression) function :

- $(j_0, s_0) = ArgMin_{\{j,s\}} \mathcal{D}(j, s)$

- with
  $\mathcal{D}(j, s) = min_{c1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + min_{c2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2$

- $\hat{c}_1 = mean(y_i / x_i \in R_1(j, s))$ et $\hat{c}_2 = mean(y_i / x_i \in R_2(j, s))$

- $\mathcal{D}(j, s) = \mathcal{D}_{\mathcal{R}_1} + \mathcal{D}_{\mathcal{R}_2}$
  $\mathcal{D}(j, s) = \mathcal{D}_{Left} + \mathcal{D}_{Right}$

# Regression Trees. Construction (2/3)

**Rule for Stopping the construction of the tree :**
A node is a terminal if the observations data (in this node) are "homogeneous".
No more admissible partitions if

- The number of observations (nodes or leaves) is below a given threshold (algorithmic default parameters)

- The deviance ratio (before and after the split) is lower than a given threshold : $\mathcal{D} - (\mathcal{D}_{Left} + \mathcal{D}_{Right}) <$ Threshold (algorithmic default parameter)

- in R for the rpart() function : MinCut, MinSize, MinDev parameters.

# Regression Trees. Prediction (3/3)

- The regression function is given by

$$f(x) = \sum_{m=1}^{M} c_m 1_{x \in \mathcal{R}_m}$$

- The predicted regression model is given by

$$\hat{f}(x) = \sum_{m=1}^{M} \hat{c}_m 1_{x \in \mathcal{R}_m}$$

- Notations :
  - $\hat{c}_m$ is the average of the Train observation data in region $\mathcal{R}_m$
  - Prediction for one observation : $x \in \mathcal{R}_m$, $\hat{y} = \hat{c}_m$.

# Over fitting is what you want to avoid ! !

- Question : what is the optimal size of a tree ?
  - Deep enough for a well approximation of the data
  - Not too deep to avoid over fitting
  - The minimization of the Deviance itself is not enough !

- Theory : to able to balance the bias and variance
  - short trees $\rightarrow$ important bias /low variance
  - Deep trees $\rightarrow$ low bias / large variance

- Idea : to be able to prune a original deep tree by cutting some branches using a penalized criteria

$$\text{erreur}(\text{Tree}) + \lambda \cdot |\text{Tree}|$$

## Tree drawbacks

- Their instability !

- Very different trees may be computed for different samples (of the same distribution)

- The greedy optimization (very efficient) provides a "local" solution

- For improving the robustness : Aggregation

# Outline

**1** Decision tree

**2** Regression Tree

**3** Ensemble methods

**4** Classification Tree

**5** Applications

**6** Pruning and Model selection - original approach

# Decision trees.

# Ensemble methods

# Ensemble methods

- Bagging : Random strategies on the observation data.
  Bagging pour Bootstrap Aggegating (Breiman, 1996)

- Random Forest : Random strategies on both observation data and
  variables (Breiman, 2001)

- Boosting : strategies on the $c_m$ aggregation parameters
  (ADABOOST : Schapire 1990, Freund & Schapire 1995)

We illustrate these three strategies (Bagging, Random Forest and
Boosting) on regression trees.

# Decision trees

# Bagging

# Family of random observation sets. Bagging

- $Y$ target variable, quantitative or qualitative

- $X = (X^1, \ldots, X^p)$ explanatory variables ($\mathbb{R}^p$)

- $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ $n$ sample with a distribution law $\mathcal{F}$.

- $\Phi(x)$ a model function of $x = (X^1, \ldots, X^p)$

- $B$ Bootstrap independant samples $\{\mathcal{S}_b\}_{b=1,B}$
    - $\rightarrow \hat{\Phi}_{\mathcal{S}_b}(.)$

- Aggregation of the different prediction models
    - $Y$ quantitative : $\hat{\Phi}_B(.) = \frac{1}{B} \sum_{b=1}^{B} \hat{\Phi}_{\mathcal{S}_b}(.)$ (average)
    - $Y$ qualitative : $\hat{\Phi}_B(.) = argmax_j \, card\{b|\hat{\Phi}_{\mathcal{S}_b}\}$ (vote)

- $\Phi(.) = \mathbb{E}_{\mathcal{F}}(\hat{\Phi}_{\mathcal{S}})$ estimator with no bias

Averaging independant prediction in order to decrease the variance
$B$ independant Bootstrap samples, $B$ bootstrap replications.

# Bagging. Breiman 1996

Goal : Averaging "independent" predictions in order to reduce the variance using independant classifiers ($B$ bootstrap replications).

$$\mathsf{Var}(\bar{Z}) = \frac{\mathsf{Var}(Z)}{\#\{B\}}$$

Method :

- $Y$ target variable, qualitative or quantitative

- $X = (X^1, \ldots, X^p)$ Multi dimensional co variables ($\mathbb{R}^p$)

- $\Phi(X)$ a given model function of the covariable $X = (X^1, \ldots, X^p)$

- $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ $n$ sample of $\mathcal{F}$ distribution law.

Bagging provides a family of random models :

- $B$ independant samples are generated with replacement : $\{\mathcal{S}_b\}_{b=1,B}$
    - $Y$ qualitative : $\hat{\Phi}_B(.) = \arg\max_j \mathsf{card}\{b | \hat{\Phi}_{\mathcal{S}_b}\}$ (vote)
    - $Y$ quantitative : $\hat{\Phi}_B(.) = \frac{1}{B} \sum_{b=1}^{B} \hat{\Phi}_{\mathcal{S}_b}(.)$ average

- $\Phi(.) = \mathbb{E}_{\mathcal{F}}(\hat{\Phi}_{\mathcal{S}})$ estimator with low/ no biais

# Bagging. Illustration
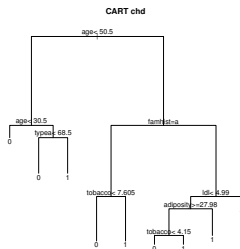


Random observations          Random observations          ...
$1^{st}$ sampling                 $2^{nd}$ sampling                ...
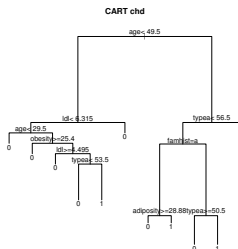
$\rightarrow$ A tree is built for each Bootstrap sample

# Bagging. Benefits and drawbacks

- Complete tree for every bootstrap sample. Faible biais (+)

- Reduced variance for aggregated models. (+)

- Sorage of all calibrated models (-)

- Computation time (-)

- Black-box model (-)

- choice of the M value(-)
  -numbers of models (number of trees)

## Statistical benefits

- No bias amplification

- With low correlation contributors

$$\mathrm{cov}(g_t(x), g_t'(x)) \simeq 0$$

We observe a variance reduction of the order $1/T$

$$\mathbb{V}(\widetilde{g}_T(x)) \simeq \frac{1}{T}\mathbb{V}(g_1(x))$$

# OOB. Out Of Bag Error

$\mathcal{D}_n = \{(x_i, y_i) \ y_i \in \{1, .., k..., K\} \ 1 \le i \le n\}$ is the original data set.

For each bootstrap sample, (for each $b$ tree) :

- The tree is built using the data of $\mathcal{D}_n^b$, a $n$ Bootstrap sample ($n$ observations drawn with replacement).

- The remaining observations of $\mathcal{D}_n$ which are not in the train data set are used to evaluate the generalization power :
  $\bar{\mathcal{D}}_n^b = \mathcal{D}_n - \mathcal{D}_n^b$ (observations of $\mathcal{D}_n$ which are not in $\mathcal{D}_n^b$)

For each observation $x_i$, the OOB performance (or error) is defined

- $\hat{\phi}^{OOB}(x_i)$ computed if $x_i \in \bar{\mathcal{D}}_n^b, b = 1...B$, with $n_B = \sum_{b=1}^{B} 1_{x_i \in \bar{\mathcal{D}}_n^b}$

- $E^{OOB}(x_i) = y_i - \frac{1}{n_B} \sum_{x_i \in \bar{\mathcal{D}}_n^b} \hat{\phi}_{OOB}(x_i)$

- $E^{OOB} = \frac{1}{n} \sum_{i=1}^{n} E^{OOB}(x_i)$

# Decision Trees
# Random Forest

# Random Forest. Breiman 2001.

## Similar Goal as Bagging :

Averaging "independent" predictions in order to reduce the variance using independant classifiers ($B$ bootstrap replications). $\mathsf{Var}(\bar{Z}) = \frac{\mathsf{Var}(Z)}{\#\{B\}}$

## Method :

- The model is based on a bagging procedure

- At each node, the final variable is selected in a sub-set of variables chosen at random.

- Tuning parameters : R sofware by default : classification : p/3 ; regression $\sqrt{p}$.

$\rightarrow$ Random choice of the variables to provide more independant classifiers

# Feature subsampling

**Feature subsampling step :** the particularity of random forests is that only a random subset of p features is tried each time when looking for a split. This reduces the correlation between the trees.

Recommendations from the authors ($p$ number of variables) :

- Classification : default value for $p_{\text{tried}}$ is $\lfloor \sqrt{p} \rfloor$ and minimum node size is 1.

- Regression : default value for $p_{\text{tried}}$ is $\lfloor p/3 \rfloor$ and minimum node size is 5. In practice, best values depend on the problem : tuning parameters.

# Some remarks on the RF method

- Usually, trees are not pruned in random forests. Indeed, the gain of pruning does not seem crucial, and hence, this choice allows to save a tuning parameter.

- When the number of variables is large, but the fraction of relevant variables small, random forests do not perform very well for small p, since, at each split, the chance that the relevant variables will be selected is small.

# A variant of random forests, ExtraTrees

**ExtraTrees :** Extremely randomized trees.

Here, trees are built on the initial sample $\mathcal{D}_n$, **no bootstrap**.

1. At each node, a random subset of features is drawn as for random forests.

2. A few number of thresholds is then selected uniformly at random in the feature range.

3. Finally, the best split among these thresholds and features is selected using some impurity criterion (Gini, Entropy).

Remark :

- Due to this random choice, the algorithm is faster.

- Random thresholds lead to more diversified trees, and thus, are some form of regularization.

- Since trees are combined and averaged, not so bad to select the thresholds at random...

$\rightarrow$ Python : sklearn.ensemble.ExtraTreesClassifier()

# RF Variable importance

Regarding RF interpretability, once again, the resulting estimator is not a tree. Nevertheless, random forests come along with variable importance plots, yielding very useful information.

Several criteria may be found to compute variable importance :

1. At each split in each tree, the improvement in the split-criterion (ex Gini) is attributed to the splitting variable, and then, for each variable, the values are accumulated over all the trees in the forest.

2. A different variable importance measure, based on the OOB samples. When the b-th tree is grown, the prediction accuracy is compared with the prediction accuracy when the values for the j-th variable are randomly permuted in the OOB samples. The decrease in accuracy due to this permuting is averaged over all trees, and is used as a measure of the importance of variable $j$ -th in the random forest.

# Variable Importance Measure for Random Forest

The **V**ariable **I**mportance **M**easure (VIM) quantifies the mean decrease in Accuracy for each covariable thanks to the following protocol.

Considering a RF with $B$ bootstrap sample trees.

- For each $b$ tree, the performance error, $E_{OOB(b)}$ is computed ($1 \leq b \leq B$)
  using the Out Of Bootstrap sample $OOB(b)$.
    1. For each variable $X_k$, $1 \leq k \leq p$, the observations of the OOB sample are **uniformly shuffled for the $b^{th}$ tree** .
    2. The VIM is then computed for the $X_k$variable and the $b$ tree :
    $VIM(b, k) = E_{OOB(b)}^k - E_{OOB(b)}$ (several definition/ possible renormalisation)

- The Mean Decrease in Accuracy for each variable $X_k$ is then measured for the RF thanks to all trees :
  $VIM(k) = \frac{1}{B} \sum_{b=1}^{b=B} VIM(k, b)$

# Variable Importance Measure (VIM) for Random Forest

- The VIM is computed independently for each variable : two correlated variables have a similar VIM value
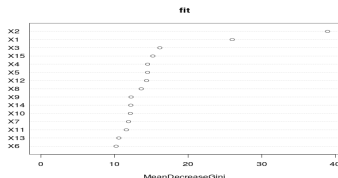


- Several error indexes can be introduced as Gini, entropy... indexes.

- The VIM indicator may be normalized as initially proposed by Leo Breiman :
  $VIM(k) = \frac{1}{B} \sum_{b=1}^{b=B} VIM(k, b)/E_{OOB(b)}$
  $VIM(k) = \frac{1}{B} \sum_{b=1}^{b=B} VIM(k, b)/S_{OOB}$ ($S_{OOB}$ : std of OOB for $B$ trees)

# Variable Importance

The variable Importance is computed by the average decrease in impurity provided by each variable calculated by the Gini index. The decrease for each node is cumulative, then an average over all trees is performed.
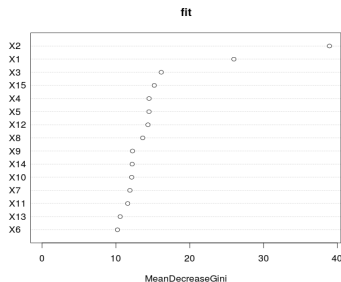


Graph computed with the VarImpPlot() function in R.

# Outline

# CART-

# Classification tree

# Classification Trees

Classification tree is a method which splits the input space in a set of rectangularly domains, in which a **constant model** is adjusted.

$\rightarrow$ The global classification function is given by

$$f(x) = \sum_{m=1}^{M} c_m 1_{x \in \mathcal{R}_m}$$

$c_m$ is a constant modality for each region $\mathcal{R}_m$

Question : How to compute $c_m$, $1 \leq m \leq M$, for the $M$ regions ?
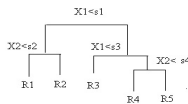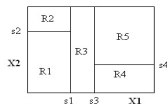
# Classification Trees

**Classification trees** are used for a qualitative target variable and are associated in this case with a classification criteria.

- Using the Training set.
  For node $m$ corresponding to region $\mathcal{R}_m$ with $N_m$ observations

  - In node $m$, the frequency for modality $k$ is estimated :
    for $k \in \{1, ..., K\}$

    $$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in \mathcal{R}_m} I(y_i = k; (x_i, y_i) \in \text{TrainDataSet})$$

  - In region (node) m, an **new** observation is affected to class $k_0$ if

    $$k_0 = \arg\max_{k \in 1..K} \{\hat{p}_{mk}\}$$

    which represents the most represented class in node $m$.

# Classification Trees

- The classification function is given by

$$f(x) = \sum_{m=1}^{M} c_m 1_{x \in \mathcal{R}_m}$$

- Notations : $c_m$ corresponds to the main modality on the training set for the region $\mathcal{R}_m$

- The estimated classification function is given by

$$\hat{f}(x) = \sum_{m=1}^{M} \hat{c}_m 1_{x \in \mathcal{R}_m}$$

- For an observation $x \in \mathcal{R}_m$, $\hat{y} = \hat{c}_m$. $\hat{c}_m$ estimated on Train DataSet.

# Classification Tree. Construction of the Tree

In the classification setting, ther are several ways to measure the quality of a split. **Node Impurity measures (Left ou Right) :**

- Missclassification :
  $$\mathcal{D}_{\mathcal{R}_m} = \frac{1}{N_m} \sum_{i \in \mathcal{R}_m} I(y \neq k(m)) = 1 - \hat{p}_m$$

- Gini index (most used) :
  $$\mathcal{D}_{\mathcal{R}_m} = \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$
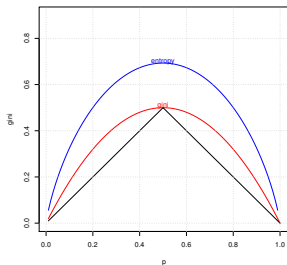  $$= 2p(1 - p)$$

- Entropy :
  $$\mathcal{D}_{\mathcal{R}_m} = - \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$
  $$= -p \log_2 p + (1 - p) \log_2(1 - p)$$

with $p_m = \frac{1}{N_m} \sum_{i \in \mathcal{R}_m} I(y = k(m))$.
* For 2 classes and for each region

# Decision trees

**Comparaisons of node impurity**



A function of impurity is by default chosen (R : gini)

# Classification trees. Node creation

**For a impurity measure**

- k : node number

- $\mathcal{R}_1$ et $\mathcal{R}_2$ regions of the two leaves

- The algorithm computes the optimal partition for which the value of $\mathcal{D}_{\mathcal{R}_1} + \mathcal{D}_{\mathcal{R}_2}$ is minimal

- i.e. at each step $k$ , the split of "one upper region" in "two lower regions" (corresponding to the recursive construction of the tree) maximizes the difference of node impurity measure (deviance) :

$$\Delta \mathcal{D}_{\mathcal{R} \to \mathcal{R}_1 + \mathcal{R}_2} = \mathcal{D}_{\mathcal{R}} - (\frac{N_{\mathcal{R}_1}}{N_{\mathcal{R}}} \mathcal{D}_{\mathcal{R}_1} + \frac{N_{\mathcal{R}_2}}{N_{\mathcal{R}}} \mathcal{D}_{\mathcal{R}_2})$$
$$\{X^j, 1 \leq j \leq p\}$$

# Classification tree. Node creation. Illustration

**Impurity measure : Gini index**

|        |           | $Y = 0$   | $Y = 1$   |           |
|--------|-----------|-----------|-----------|-----------|
| Left   | $X < S$   | $n_{11}$  | $n_{12}$  | $n_{1+}$  |
| Right  | $X \geq S$ | $n_{21}$  | $n_{22}$  | $n_{2+}$  |
| Top    | global    | $n_{+1}$  | $n_{+2}$  | $n_{++}$  |

Gini indices :

| Top $G$ | $2 * \frac{n_{+1}}{n_{++}}(1 - \frac{n_{+1}}{n_{++}})$ |
|---|---|

| Left $G_L$ | $2 * \frac{n_{11}}{n_{1+}} * (1 - \frac{n_{11}}{n_{1+}})$ |
|---|---|

| Right $G_R$ | $2 * \frac{n_{21}}{n_{2+}} * (1 - \frac{n_{21}}{n_{2+}})$ |
|---|---|

# Classification tree. Node creation. Illustration

**Impurity measure. Entropy**

|        |             | $Y = 0$  | $Y = 1$  |         |
|--------|-------------|----------|----------|---------|
| Left   | $X < S$     | $n_{11}$ | $n_{12}$ | $n_{1+}$ |
| Right  | $X \geq S$  | $n_{21}$ | $n_{22}$ | $n_{2+}$ |
| Top    | (global)    | $n_{+1}$ | $n_{+2}$ | $n_{++}$ |

Entropy

top $H$     $\frac{n_{+1}}{n_{++}} \log \frac{n_{+1}}{n_{++}} + \frac{n_{+2}}{n_{++}} \log \frac{n_{+2}}{n_{++}}$

Left $H_L$     $\frac{n_{11}}{n_{1+}} \log \frac{n_{11}}{n_{1+}} + \frac{n_{12}}{n_{1+}} \log \frac{n_{12}}{n_{1+}}$

Right $H_R$     $\frac{n_{21}}{n_{2+}} \log \frac{n_{21}}{n_{2+}} + \frac{n_{22}}{n_{2+}} \log \frac{n_{22}}{n_{2+}}$

# Decision tree. Cardiac Heart Disease application (chd)

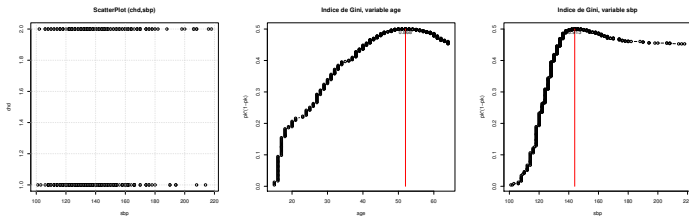The target variable is chd (binary variable).
$\rightarrow$ **Computation of the decision threshold for the quantitative co-variable age and for the Gini index**
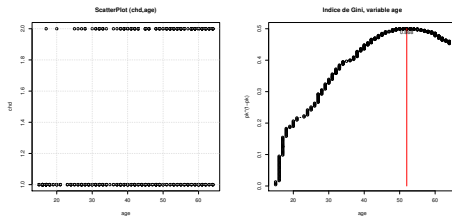


$\mathcal{D}(\text{age})=0.868$

# Decision tree. Cardiac Heart Disease application (chd)

**Decision threshold computation and variable selection between two covariables (age, spb) based on the Gini Index**



The age variable is selected : $\mathcal{D}(\text{age})=0.868 < \mathcal{D}(\text{spb})=0.915$

# Decision Tree, Cardiac Heart Disease application (chd)

**Decision threshold computation and variable selection between two covariables (age, ldl) based on the Gini Index**
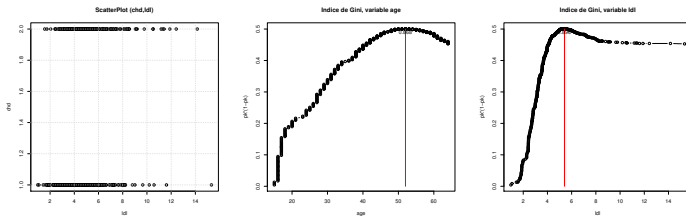


The age variable is selected : $\mathcal{D}(\text{age})=0.868 < \mathcal{D}(\text{ldl})=0.896$

# Classification Trees

**Stopping the recursive split process**

A node is terminal if :

- the region is homogeneous (only one label)

- The is no authorized partitions regarding the algorithmic rule of decreasing the variance criteria ($\Delta\mathcal{D}$).

- The number of observations in the region NCut (or in the sub regions Nsize) is lower than a given threshold then No authorized split. (algorithm parameters).

<div align="center">Software Tuning parameters : (NCut, Nsize, $\Delta\mathcal{D}$)</div>

# Classification trees. Estimation. Prediction

From Estimation to Prediction :

- $Y$ quantitative : average of observations of the training data set
- $Y$ qualitative. Each leave is affected to one given class $C_k$ of $Y$ regarding a conditional approach regarding the training data set

    - **The more frequently class represented in the node (training data set)**

    - or

    - the "more probable" class if some apriori exists (training data set)

    - The "cheapest" class if there exists some coast indications.

# Outline

# Illustration. SPAM classifier

## Problem : to be able to automatically classify a regular email from a spam email

## SPAM
From : Felix Damians, From .Abidjan Cote D.Ivoire

Hello Dear, Pardon me for not having the pleasure of knowing your mindset before making you this offer and it is utterly confidential and genuine by virtue of its nature. I want someone like you to help me out after i had pray ,then believes that you are a good person and that i can stay with you for the rest of my life , am 24 years old, My late father is a wealthy and successful business man before he died , My mum died when i was a baby, am the only child in my family. Honestly speaking , i am ready to give you 15percent of this total money for your assistance and with extra 5percent for your expenses on phone call, please reply me now if really serious to help me out so that i can tell you more about my intention and forward to you some of the legal papers after knowing you more better. Yours Felix Damians.

# Illustration. SPAM classifier

## A Text Mining is first performed to compute **features**

### SPAM

From : Felix Damians, From .Abidjan Cote D.Ivoire

Hello Dear, Pardon me for not having the pleasure of knowing your mindset before making you this offer and it is utterly confidential and genuine by virtue of its nature. I want someone like you to **help** me out after i had pray ,then believes that you are a good person and that i can stay with you for the rest of my life , am 24 years old, My late father is a wealthy and successful business man before he died , My mum died when i was a baby, am the only child in my family. And he told me that he used my name to deposit ( us dollars 12.5million ) in the bank and he seriously advise me to transfer this total **money** to oversea account for my investment, where i will start my new life and finish my education , Because of this reason, i am soliciting your assistance for the claim and transfer to your bank account for the business. Honestly speaking , i am ready to give you 15percent of this total money for your assistance and with extra 5percent for your expenses on phone call, please reply me now if really serious to **help** me out so that i can tell you more about my intention and forward to you some of the legal papers after knowing you more better. All about the money are legal and i have all the legal documents and papers of the money money with me which the bank issued to my late father the day of the deposit, Because of the war in our country now the bank manager here has promised to me that they will use their branch corrospodant bank in Europ or Asia.to transfer the money into any account i provided to them that is why i contacted you to **help** me out in receiveing of the money money into an account over there in your country before i join you over to has rest of mind. Thanks and remain bless with your family as i wait for your

# Illustration. SPAM classifier

## SPAM data base

- 48 continuous real [0, 100] attributes of type word-freq-WORD = percentage of words in the e-mail that match WORD, i.e. 100 * (number of times the WORD appears in the e-mail) / total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

- 6 continuous real [0, 100] attributes of type char-freq-CHAR = percentage of characters in the e-mail that match CHAR, i.e. 100 * (number of CHAR occurences) / total characters in e-mail

- 1 continuous real [1, ...] attribute of type capital-run-length-average = average length of uninterrupted sequences of capital letters

- 1 continuous integer [1, ...] attribute of type capital-run-length-longest = length of longest uninterrupted sequence of capital letters

- 1 continuous integer [1, ...] attribute of type capital-run-length-total = sum of length of uninterrupted sequences of capital letters = total number of capital letters in the e-mail

- 1 nominal 0,1 class attribute of type spam = denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.
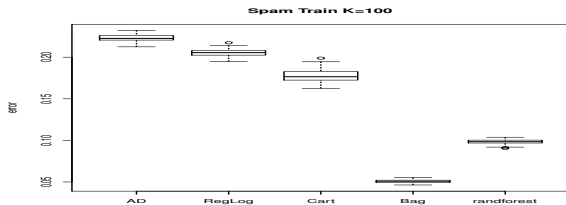
## Historical data base

- $p = 57$ features computed from the initial texts ( $p = 56$)

- $n = 4601$ Emails with

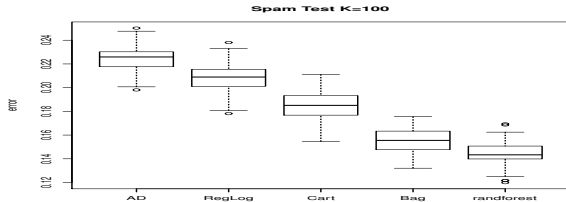- $Y \in \{0, 1\}$ a binary indicator

- $n >> p$

# Predictive performance comparison for Several Classifiers
Based on K-fold.

# Outline

**1** Decision tree

**2** Regression Tree

**3** Ensemble methods

**4** Classification Tree

**5** Applications

**6** Pruning and Model selection - original approach

# Classification trees. Model selection. Pruning

In order to avoid (minimize) overfitting, the length of the tree is penalized.

One the maximal tree is built (one tree), the pruning algorithm proposes several trees by pruning. A comparison between all these the trees helps to select the tree which minimize the following complexity criteria.

The tree with the lowest error is finally selected.

Complexity criteria :

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|$$

avec

- $|T|$ : terminal node number
- $N_m = \sharp\{x_i \in \mathcal{R}_m\}$
- $\hat{c}_m = \frac{1}{N_m} \sum_{x_1 \in \mathcal{R}_m} y_i$
- $\hat{Q}_m(T) = \frac{1}{N_m} \sum_{x_i \in \mathcal{R}_m} 1_{y_i \neq k_m}$
- $\alpha$ is selected by cross-validation

# Classification trees & Predictive Power

- A deep decision tree may fit perfectly the training dataset (low biais/ strong variance).
- $\rightarrow$ Pruning : increasing the Predictive Power for keeping the interpretability with one given decision tree.
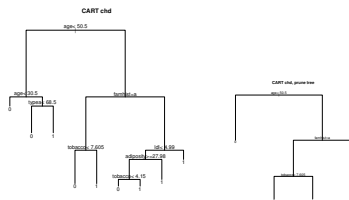


Complexity criteria : $C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T|$
$\alpha$ selected by cross-validation