

UP2 : Apprentissage Statistique et Analyse de données

Arbre de décision et Forêt aléatoire

ANIS S. HOAYEK

Octobre 2025

- ① Apprentissage statistique pour l'aide à la décision.
- ② Méthodes de Classification : Principes généraux
- ③ Arbres de décision.
- ④ Validité : Sensibilité, Spécificité, ROC, AUC, LIFT, etc.
- ⑤ Forêts aléatoires.
- ⑥ Lien avec Bagging et Boosting.
- ⑦ Forêts aléatoires non supervisées (Isolation Forest).
- ⑧ Applications classiques TD (à la main) + TP (sous R et/ou Python).
- ⑨ Évaluation : Examen écrit (13 Novembre 2025) + TP noté (compte rendu par groupes selon la liste prédéfinie).

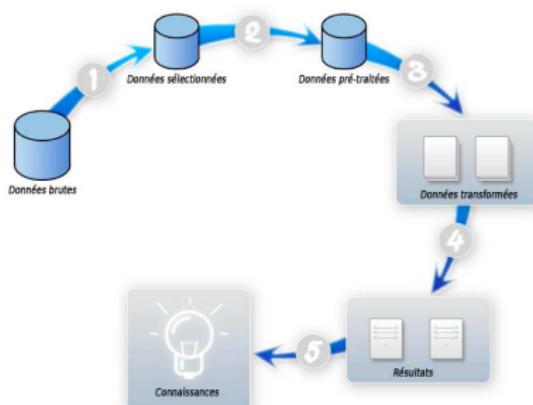
Apprentissage statistique pour l'aide à la décision

- ① La statistique est incontournable dans les sciences expérimentales : données dans le but spécifique de confirmer ou d'infirmer des hypothèses scientifiques.
- ② Données existantes avant même qu'on se pose des questions spécifiques.
- ③ Apprendre de ces données les réponses à des questions \Rightarrow La statistique permet de faire cet apprentissage \Rightarrow prendre des décisions.
- ④ Méthodes de la statistique dans un contexte expérimental :
 - ① Planification du recueil des données (Échantillonnage, Plan d'expérience, Data management)
 - ② Description et présentation des données (Statistique descriptive)
 - ③ Induction de règles générales à partir de ce qui a été observé (Modélisation et Inférence statistique)

Définition.

Le data-mining, est un processus d'extraction des connaissances qui comprend plusieurs étapes :

Pré-traitement des données \Rightarrow Fouille des données pré-traitées \Rightarrow Apprentissage à partir de ce qui a été observé



Domaines d'application

Entreprises stockent dans des entrepôts de données (data warehouse) des millions d'octets de données relatives à leurs activités à des fins de :

- ① Gestion des stocks, des services, des ressources humaines, des clients, etc.
- ② Contrôle de qualité : suivi en ligne des paramètres de production, traçabilité, etc.
- ③ Secteur bancaire : Identification du risque de crédit d'un client en fonction de sa probabilité de défaut de paiement.
- ④ Médecine : Identification des patients à risque et les tendances de la maladie.
- ⑤ Marketing : Identification du taux de désabonnement des clients.

Remarque.

Les données n'ont pas été recueillies à ces fins !

Étape 1 : Nettoyage et mise en forme des données

- ① S'assurer de la fiabilité des données,
- ② Éliminer les sujets atypiques ou non pertinents pour l'analyse
- ③ Obtenir un "feeling" du jeu de données :
 - ① Analyses descriptives unidimensionnelles adaptées à la nature des variables.
 - ② Analyses statistiques bivariées adaptées à la nature des couples de variables pour faire apparaître les liens entre elles.
 - ③ Analyses statistiques multivariées.

BUT : Repérer les variables ayant des distributions statistiques "bizarres". Le cas échéant, décider s'il convient d'éliminer ou de transformer les données.

Étape 2 : Analyse exploratoire des données

- ① Réduire la dimension du vecteur de variables
- ② Éliminer les variables redondantes

Via des techniques statistiques qui ne sont pas l'objet de ce cours : ACP, AFC, AFD, etc.

Étape 3 : Classification/Régression

- ① Une variable d'intérêt (ou cible) Y qualitative/discrète ou quantitative, définissant des groupes de la population
- ② Une ou plusieurs variables explicatives ou prédictives X_1, \dots, X_p .
- ③ Sur un individu, on va observer uniquement la valeur (x_1, \dots, x_p) de (X_1, \dots, X_p) .
- ④ On voudra lui assigner sa valeur y de Y (son groupe ou sa valeur) en minimisant les risques d'erreur.
- ⑤ Pour établir le classifieur qui fera ce travail, on dispose de données du type $(x_1, \dots, x_p, y) \implies$ On veut à partir de ces données établir la meilleure règle de classification/régression possible pour classer par la suite les éventuels prospects.

Étape 3 bis : Clustering

- ➊ On dispose des valeurs observées (x_1, \dots, x_p) sur n individus des variables (X_1, \dots, X_p) .
- ➋ Regrouper, en K groupes les plus dissemblables entre eux possibles, les individus de façon à ce qu'à l'intérieur de chaque groupe, les données soient le plus homogène possible.
 - ➌ K connu : Clustering supervisée
 - ➍ K inconnu : Clustering non supervisée
- ➎ Analyser les particularités de chacun des K groupes

Remarque.

Le clustering revient à "créer" une variable qualitative/discrète indiquant les groupes auxquels peuvent appartenir les individus.

Étape 3 bis : Modélisation

- ① La modélisation a pour objectif d'étudier les liens plus fins entre :
 - ① Une (ou plusieurs) variables d'intérêt (ou cible) Y
 - ② Une ou plusieurs variables explicatives (ou prédictives) X_1, \dots, X_p
- ② Elle se fait notamment via les méthodes de :
 - ① Régression linéaire simple, multiple, non-linéaire : Y et X_1, \dots, X_p quantitatives/continues
 - ② ANOVA, ANCOVA : Y quantitative/continue, au moins une des X_1, \dots, X_p qualitative/discrète
 - ③ Régression logistique : Y qualitative/discrète et X_1, \dots, X_p quantitatives/continues
 - ④ Régression log-linéaire : Y et X_1, \dots, X_p qualitatives/discrètes
 - ⑤ Autres méthodes : séries chronologiques etc...

Remarque.

Dans ce cours, on se concentre sur les méthodes d'apprentissage que sont la classification, et la régression.

- ① Extraction des données de l'entrepôt, via éventuellement un sondage,
- ② Nettoyage : analyse graphique, validation des codages etc..
- ③ Exploration : réduction de la dimension, élimination de variables redondantes,
- ④ Analyse (classification, régression, clustering, modélisation)
- ⑤ Exploitation du modèle et diffusion des résultats.

Pièges du data-mining

- ➊ Attaque du problème sans but précis.
- ➋ Data-snooping : "If you torture the data long enough, it will confess to anything" (Ronald Coase)
- ➌ Incompréhension des algorithmes utilisés par les logiciels.
- ➍ Confiance aveugle dans les sorties des logiciels informatiques de plus en plus "tout intégré" et de plus en plus "boîte noire".
- ➎ Absence d'interrogation et d'esprit critique dans le choix des "valeurs par défaut" fait par les logiciels.

Méthodes de Classification : Principes généraux

Problème : Déterminer à quel groupe $\mathcal{G}_1, \dots, \mathcal{G}_K$ appartient un individu :

- L'appartenance à un groupe est déterminée par la valeur y d'une v.a. $Y \in \{1, 2, \dots, K\}$.
- Si $y = k$, alors l'individu $\in \mathcal{G}_k$.

Exemple : Un client demande un prêt à une banque. La banque veut savoir si le client est en mesure de rembourser son prêt ($y = 1$) ou non ($y = 2$). Évidemment, l'observation de y est impossible, sauf à attendre l'issue du prêt.

Idée : Mesurer sur l'individu un certain nombre de variables

$X = (X_1, \dots, X_p)$, (e.g. : ancienneté, compte courant, compte sur livret, assurance-vie, etc...) dites discriminantes, classifiantes ou de

"procuration", et déterminer à partir de la valeur observée

$x = (x_1, \dots, x_p)$ de X si l'individu fait (fera) parti de \mathcal{G}_1 ($y = 1$),
 \mathcal{G}_2 ($y = 2$), ..., \mathcal{G}_K ($y = K$).

Postulat de base : X est un vecteur aléatoire. Il a donc une loi de probabilité (notée \mathcal{L}_k) qui diffère selon la valeur de Y (c-à-d : si X provient de \mathcal{G}_k , $X \sim \mathcal{L}_k$)

Définition.

Un classifieur est une fonction $d(\cdot) : \mathbb{R}^p \longrightarrow \{1, 2, \dots, K\}$ qui, à chaque $x = (x_1, \dots, x_p) \in \mathbb{R}^p$, associe un et un seul $k \in \{1, 2, \dots, K\}$.

La classification se fait comme suit : Si un individu a x pour valeur de X , on l'assigne à \mathcal{G}_K si $d(x) = k$.

Comment déterminer $d(\cdot)$? On aura besoin :

- ➊ D'une méthode pour construire $d(\cdot)$
- ➋ D'information sur le contexte du problème (\mathcal{L}_k , probabilités a priori d'appartenance aux groupes)
et/ou
- ➌ D'un échantillon \mathcal{E} composé de valeurs observées
 $(x_i, y_i), i = 1, \dots, n$.

Difficulté : \exists une infinité de façons de choisir $d(\cdot)$. Quels critères utiliser pour ce choix ?

La construction d'un classifieur dépend de l'information dont on dispose.

- ① Si on connaît parfaitement la loi \mathcal{L}_k de X (de densité $f_k(\cdot)$) quand $y = k$, on peut utiliser cette loi pour construire $d(\cdot)$ (cas d'école car il n'utilise pas \mathcal{E})
- ② Si on connaît imparfaitement cette densité (i.e. à des paramètres près), on peut utiliser \mathcal{E} pour estimer ces paramètres, puis fonctionner comme en 1) (classifieur du Maximum de Vraisemblance avec paramètres estimés)
- ③ Si on n'a aucune information sur les différentes lois \mathcal{L}_k , on doit utiliser \mathcal{E} pour construire un classifieur (e.g. arbre de décision). C'est le cas le plus réaliste.

À ces informations peut s'ajouter :

- Une information a priori sur les "chances" qu'un individu $\in \mathcal{G}_k$. Cette information prend la forme de probabilités a priori π_k , $k = 1, \dots, K$ avec $0 < \pi_k < 1$ et $\pi_1 + \dots + \pi_K = 1$
- Une information sur les coûts d'une mauvaise classification $C(k', k)$ (par. ex coût de consentir un prêt à un client mauvais payeur)

Remarque.

Nous allons nous concentrer sur le cas 3) qui est celui rencontré en pratique.

Évaluation d'un Classifieur : Probabilités d'assignation correcte et incorrecte

- L'individu sera assigné à \mathcal{G}_k selon la valeur observée x de X .
- Cette assignation est correcte si l'individu provient bien de \mathcal{G}_k . Sinon, l'assignation est incorrecte.
- Comme cette assignation se fait à partir de la valeur observée x de X , qui est aléatoire, il existe donc des probabilités de bonne et mauvaise assignation.

Définition.

La probabilité d'assigner un individu à $\mathcal{G}_{k'}$ alors qu'il est issu du groupe \mathcal{G}_k est définie par

$$\rho_{k'k} = \mathbb{P}[X \mapsto \mathcal{G}_{k'} \mid X \in \mathcal{G}_k \text{ ou } X \in \mathcal{L}_k] = \int_{\mathcal{G}_{k'}} f_k(x) dx,$$

où $f_k(\cdot)$ est la densité de X (donc de \mathcal{L}_k) quand l'individu est issu de \mathcal{G}_k (note : premier indice = là où il est envoyé ; deuxième indice, d'où il provient)

Évaluation d'un Classifieur : Probabilités d'assignation correcte et incorrecte

- Si les densités $f_k(\cdot)$ des lois \mathcal{L}_k sont connues, on peut en principe calculer les ρ_{kk} (mais ce n'est pas nécessairement facile).
- ρ_{kk} ($k = 1, \dots, K$) sont les probabilités d'assignation correcte d'un individu provenant de \mathcal{G}_k .
- $1 - \rho_{kk}$ ($k = 1, \dots, K$) est la probabilité d'assigner de façon incorrecte un individu provenant de \mathcal{G}_k .
- Ces quantités peuvent être regroupées dans une matrice dite matrice de confusion dont la somme de chaque colonne = 1.

Mat. Confusion		vérité			
		\mathcal{G}_1	\mathcal{G}_2	...	\mathcal{G}_K
prédition	\mathcal{G}_1	ρ_{11}	ρ_{12}	...	ρ_{1K}
	\mathcal{G}_2	ρ_{21}	ρ_{22}	...	ρ_{2K}
	:	:	:	:	:
	\mathcal{G}_K	ρ_{K1}	ρ_{K2}	...	ρ_{KK}
total		1	1	...	1

Comparaison de deux classificateurs : Admissibilité et Probabilité globale d'assignation incorrecte

Définition.

Soit d un classifieur avec matrice de confusion $\{\rho_{k'k}, k', k = 1, \dots, K\}$ et \tilde{d} un second classifieur avec matrice de confusion $\{\tilde{\rho}_{k'k}, k', k = 1, \dots, K\}$. On dit que d est aussi bon que \tilde{d} si $\rho_{kk} \geq \tilde{\rho}_{kk} \forall k$. On dit que d est meilleur que \tilde{d} si $\rho_{kk} > \tilde{\rho}_{kk}$ pour au moins un k . Si d est un classifieur pour lequel il n'existe pas de meilleur classifieur, alors d est dit admissible.

Comparaison de deux classificateurs : Admissibilité et Probabilité globale d'assignation incorrecte

- Dans la définition précédente, les classificateurs sont jugés en fonction des éléments diagonaux de la matrice de confusion. Ce n'est pas la seule possibilité.
- Supposons que l'on dispose de l'information supplémentaire suivante :
 - On sait qu'a priori l'individu a une probabilité π_k ($\in]0, 1[$) d'appartenir à \mathcal{G}_k .
 - La probabilité globale (ou a posteriori) d'assignation correcte d'un classifieur d avec matrice de confusion $\{\rho_{kk}\}$ est définie par :

$$\begin{aligned}\mathbb{P}[\text{Individu} \longmapsto \text{à son groupe}] &= \sum_{k=1}^K \mathbb{P}[\text{Individu} \longmapsto \mathcal{G}_k \mid \text{Individu} \in \mathcal{G}_k] \mathbb{P}[\text{Individu} \in \mathcal{G}_k], \\ &= \sum_{k=1}^K \rho_{kk} \pi_k.\end{aligned}$$

Comparaison de deux classificateurs : Admissibilité et Probabilité globale d'assignation incorrecte

La probabilité globale d'assignation incorrecte de d est ainsi définie par :

$$\rho(d) = 1 - \sum_{k=1}^K \rho_{kk} \pi_k = \sum_{k=1}^K \sum_{k' \neq k} \rho_{k'k} \pi_k$$

Définition.

Supposons données les probabilités a priori π_k ($\in]0, 1[$) d'appartenir à \mathcal{G}_k . Soit d un classifieur avec matrice de confusion $\{\rho_{k'k}, k', k = 1, \dots, K\}$ et probabilité globale d'assignation incorrecte $\rho(d)$. Soit \tilde{d} un second classifieur avec matrice de confusion $\{\tilde{\rho}_{k'k}, k', k = 1, \dots, K\}$ et probabilité globale d'assignation incorrecte $\rho(\tilde{d})$. On dit que d est aussi bon que \tilde{d} si $\rho(d) = \rho(\tilde{d})$. On dit que d est meilleur que \tilde{d} si $\rho(d) < \rho(\tilde{d})$.

Note : En pratique, quand $\rho(d) < 0.2$, on considère que le classifieur est "bon".

Estimation des probabilités de bonne et mauvaise assignation

De façon générale, avec K groupes on peut estimer les $\rho_{k'k}$ par

$$p_{k'k} = \frac{n_{k'k}}{n_k}$$

où $n_{k'k}$ = nombre d'observations issues de \mathcal{G}_k et assignées à $\mathcal{G}_{k'}$ et n_k = nombre d'observations de \mathcal{E} provenant de \mathcal{G}_k . L'estimateur ainsi obtenu est appelé l'estimateur par resubstitution. Le tableau des $n_{k'k}$ est appelé la matrice d'incidence par resubstitution.

Mat. Incidence		vérité				total
		\mathcal{G}_1	\mathcal{G}_2	...	\mathcal{G}_K	
prédition	\mathcal{G}_1	n_{11}	n_{12}	...	n_{1K}	n_{1+}
	\mathcal{G}_2	n_{21}	n_{22}	...	n_{2K}	n_{2+}
	:	:	:	..	:	:
	\mathcal{G}_K	n_{K1}	n_{K2}	...	n_{KK}	n_{K+}
total		n_1	n_2	...	n_K	n

Estimation des probabilités de bonne et mauvaise assignation

Le tableau des $p_{k'k}$ est la matrice de confusion estimée par resubstitution :

Mat. Conf. Est.		vérité			
		\mathcal{G}_1	\mathcal{G}_2	...	\mathcal{G}_K
prédition	\mathcal{G}_1	p_{11}	p_{12}	...	p_{1K}
	\mathcal{G}_2	p_{21}	p_{22}	...	p_{2K}
	:	:	:	..	:
	\mathcal{G}_K	p_{K1}	p_{K2}	...	p_{KK}
total		1	1	...	1

Ces estimateurs sont en général trop optimistes car **les mêmes données servent à la fois à construire le classifieur et à estimer ces probabilités.**

Estimation des probabilités de bonne et mauvaise assignation

Il existe plusieurs façons de contourner ce problème.

Nous en évoquons une ici rapidement. Elle consiste à partitionner

l'échantillon \mathcal{E} en deux sous-échantillons : \mathcal{E}_{app} et \mathcal{E}_{est}

\mathcal{E}_{app} = échantillon d'apprentissage (avec n_{app} données) sur lequel on construit le classifieur d_{app} .

\mathcal{E}_{est} = échantillon d'estimation (avec n_{est} données) qu'on "passe" dans le classifieur pour obtenir les estimateurs :

- $p_{k'k}^{\text{est}}$, l'estimateur de $\rho_{k'k}$ pour le classifieur d
- $R^{\text{est}}(d)$, l'estimateur de $\rho(d) = 1 - \sum_{k=1}^K \rho_{kk}\pi_k$ (si les π_k sont disponibles).

Estimation des probabilités de bonne et mauvaise assignation

- Ces dernières quantités estiment en fait les $\rho_{k'k}$ de d_{app} et $\rho(d_{\text{app}})$ et non pas $\rho_{k'k}$ de d et $\rho(d)$
- Si n est grand, n_{app} et n_{est} le seront aussi. Ainsi $d \simeq d_{\text{app}} \implies p_{k'k}^{\text{est}} \approx \rho_{k'k}$ de d et $R^{\text{est}}(d) \approx \rho(d)$.
- Ainsi, cette méthode est bien adaptée au cas où n est grand. On appelle cette méthode la validation externe ou encore "**out-of sample validation**". Elle dépend cependant du choix (aléatoire) de \mathcal{E}_{est} . Si n est grand, les $p_{k'k}^{\text{est}}$ et $R^{\text{est}}(d)$ ne devraient pas trop varier d'un \mathcal{E}_{est} à l'autre.

Estimation des probabilités de bonne et mauvaise assignation

- Si n est petit, une variante est la validation croisée ou interne (leave one out cross validation)
- L'idée consiste à prendre $\mathcal{E}_{\text{est}} = \{x_1\}$ et $\mathcal{E}_{\text{app}} = \mathcal{E} \setminus \{x_1\}$. On calcule le classifieur sur \mathcal{E}_{app} ; celui-ci ne va pas différer de beaucoup de celui obtenu de \mathcal{E} .
- On regarde ensuite si l'individu de \mathcal{E}_{est} est bien classé.
- On répète successivement cette procédure avec chacune des données. Au final, on estime les $\rho_{k'k}$ par le nombre de fois qu'une observation de \mathcal{G}_k est classée en $\mathcal{G}_{k'}$.
- Des variantes utilisent des \mathcal{E}_{est} de taille > 1 (**V-fold cross validation**).
- Le prix est un temps de calcul plus long.

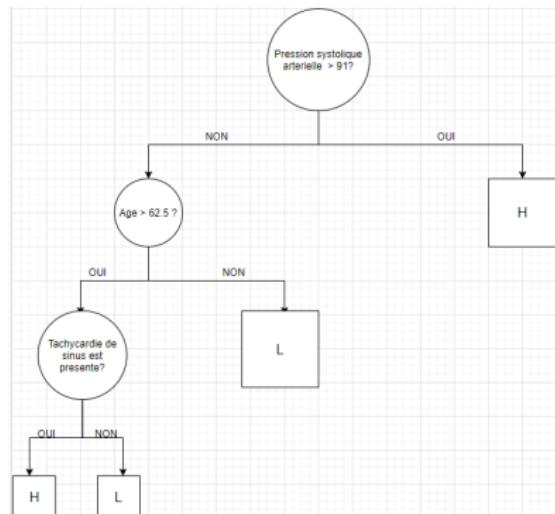
Arbres de Décision

Fait.

Référence "classique" : Breiman, L. Friedman, J. H. Olshen, R.A., Stone, C. (1984) : *Classification and regression trees*. Wadsworth International Group. Belmont, California.

- ① Algorithme de classification/régression supervisé.
- ② Méthode statistique non-paramétrique.
- ③ Permet de classer un ensemble d'individus décrits par des variables qualitatives et quantitatives.
- ④ Produit des classes les plus homogènes possibles.
- ⑤ Classifications compréhensibles pour l'utilisateur (dans les méthodes classiques (hiérarchique, k-means,...) l'information est perdue dans les classes).

- 1 Dans un hôpital, pour chaque nouveau patient avec une crise cardiaque, on mesure 15 variables pendant les premières 18 heures. Parmi les variables : la pression artérielle, l'âge et 13 autres caractéristiques résumant les différents symptômes.
- 2 L'objectif de l'étude est d'identifier les patients à haut risque (ceux qui ne survivront pas au moins 30 jours).



- Représentation :
 - La racine : χ .
 - Un nœud : sous ensemble de χ (représenté par un cercle).
 - Nœuds terminaux : sous-ensembles qui ne sont plus divisés (représentés par des boites).
 - Chaque nœud terminal est marqué par une classe cible qui est une des valeurs y d'un attribut cible Y .
- Construction d'un arbre de décision :
 - Un arbre de classification se construit par segmentations récursives de l'échantillon $\mathcal{E} = \{(x_i, y_i), i = 1, \dots, n\}$.
 - On sélectionne la variable qui sépare "le mieux" les données.
 - Le processus se répète pour chaque sous-groupe.
 - On s'arrête quand les sous-groupes atteignent la taille minimale, ou quand il n'y plus d'amélioration.

Ainsi, la construction d'un arbre de décision nécessite :

- Sélectionner les coupes.
- Décider de déclarer un nœud terminal (convertir le nœud en feuille) ou continuer de le scinder à nouveau. (L'idée principale est de choisir chaque split de façon à ce que les deux nœuds descendants soient chacun plus "**purs**" que le nœud parent).
- Affecter une classe à chaque nœud terminal.

Notations

- n : taille de l'échantillon (nombre des observations).
- K : nombre de classes de la variable cible Y .
- $N(t)$: nombre d'observations dans le nœud t .
- $N_k(t)$: nombre d'observations de la classe $k \in \{1, 2, \dots, K\}$ dans le nœud t .
- $p(k|t)$: proportion d'observations dans le nœud t appartenant à la classe $k \in \{1, 2, \dots, K\}$

$$p(k|t) = \frac{N_k(t)}{N(t)}.$$

- $p(t)$: vecteur de proportions correspondant au nœud t

$$p(t) = [p(1|t), p(2|t), \dots, p(K|t)].$$

- $Y(t)$: classe attribuée au nœud t

$$Y(t) = \arg \max_{k=1, \dots, K} p(k|t).$$

Définition.

Une mesure d'impureté d'un nœud t dans un arbre de décision ayant une variable cible Y de K classes est une fonction ayant la forme :

$$Imp(t) = \phi(p(t)),$$

où ϕ est une fonction non-négative de $p(t)$ qui satisfait les conditions suivantes :

- ① ϕ atteint son maximum unique en $p(t) = [\frac{1}{K}, \frac{1}{K}, \dots, \frac{1}{K}]$.
- ② ϕ atteint le minimum en $[1, 0, \dots, 0], [0, 1, \dots, 0], \dots, [0, 0, \dots, 1]$.
- ③ ϕ est une fonction symétrique de $p(1|t), p(2|t), \dots, p(K|t)$.

Remarque.

$Imp(t)$ est maximale quand toutes les classes sont mélangées avec des "parts égales" (Distribution uniforme/équiprobable) et est minimale quand le nœud ne contient qu'une seule classe (certitude totale).

Définitions préliminaires

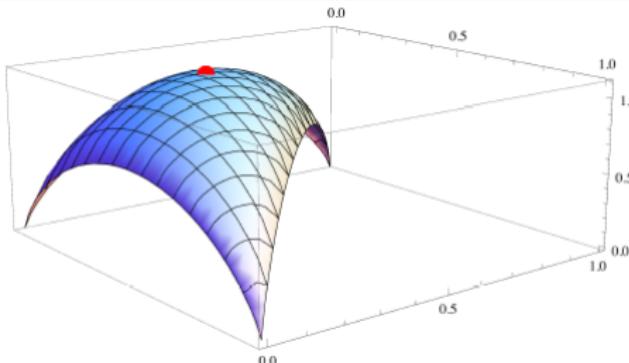
Remarque.

Comme $p(1|t) + p(2|t) + \dots + p(K|t) = 1$, on a

$$\begin{aligned}Imp(t) &= \phi(p(t)), \\&= \phi(p(1|t), p(2|t), \dots, p(K-1|t)), \\&\quad 1 - p(1|t) - p(2|t) - \dots - p(K-1|t))\end{aligned}$$

Ce qui montre que $Imp(t)$ est en fait une fonction de $K - 1$ variables.

Quand $K = 3$, on peut représenter une telle fonction et un exemple est : (le ● indique le maximum de la fonction, atteint en $(1/3, 1/3)$) :



Exemples des mesures d'impureté :

- ➊ Entropie :

$$Imp(t) = \mathcal{H}(t) = - \sum_{k=1}^K p(k|t) \log_2 p(k|t),$$

avec : $0 \log_2 0 = 0$.

- ➋ Indice de Gini :

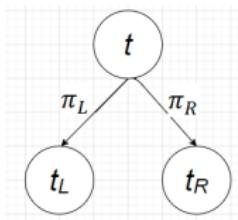
$$Imp(t) = \mathcal{G}(t) = 1 - \sum_{k=1}^K p^2(k|t).$$

Remarque.

Un nœud est pure s'il contient des données d'une seule classe. Dans ce cas $\mathcal{H}(t) = \mathcal{G}(t) = 0$.

Couper ou ne pas couper ?

On considère un nœud t avec deux nœuds fils t_L et t_R . On note cette opération de coupe par \mathcal{S} .



avec :

- π_L = proportion d'observations de t qui vont vers t_L .
- π_R = proportion d'observations de t qui vont vers t_R .

Couper ou ne pas couper ?

Qualité de la coupe \mathcal{S} est définie par la variation de la mesure d'impureté :

$$\varPhi(\mathcal{S}, t) = \Delta Imp(t) = Imp(t) - \pi_L Imp(t_L) - \pi_R Imp(t_R).$$

L'idée est de choisir une coupe \mathcal{S} qui maximise $\varPhi(\mathcal{S}, t)$. On a :

$$\varPhi(\mathcal{S}, t) \in [0, Imp(t)],$$

- Si $\varPhi(\mathcal{S}, t) = 0$, le split n'a pas diminué l'impureté.
- Si $\varPhi(\mathcal{S}, t) = Imp(t)$, alors $Imp(t_L) = Imp(t_R) = 0$ et les 2 nœuds descendants sont purs : le split a parfaitement séparé les groupes de t .

Définition 1.

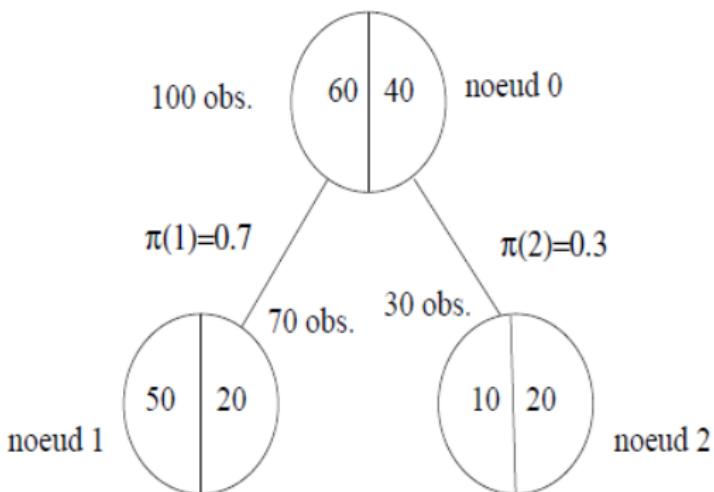
L'impureté globale d'un arbre de décision T est définie par :

$$Imp(T) = \sum_{t \in \tilde{T}} \pi(t) Imp(t),$$

où, $\tilde{T} =$ l'ensemble des nœuds terminaux

et $\pi(t) =$ la proportion de la population globale en nœud t .

Exemple Numérique



Soit $X = [X_1, X_2, \dots, X_p]$ le vecteur de variables explicatives qui se présentent dans un contexte donné.

Les coupes des nœuds d'un arbre de décision doivent vérifier les conditions suivantes :

- Chaque coupe ne dépend que d'une seule variable.
- Pour les X_i quantitatives, le critère de la coupe est de la forme :

Est-ce que $X_i \leq c$, avec $c \in \mathbb{R}$.

- Si X_i est catégorielle à valeurs dans $B = \{b_1, b_2, \dots, b_m\}$, le critère de la coupe est de la forme :

Est-ce que $X_i \in A$, avec $A \subset B$.

- A chaque nœud on considère les variables X_i une par une : 1) On trouve la meilleure coupe de chaque X_i , 2) On choisit la meilleure variable en terme de qualité de la coupe.

Arrêter les coupes si :

- La variation de la mesure d'impureté d'un nœud est inférieure à un certain seuil. (i.e. Quand $\Phi(\mathcal{S}^*, t) < \text{seuil}$ (on n'a quasiment rien gagné en splittant le nœud t)).
- Profondeur de l'arbre est supérieure à une valeur prédéfinie.
- Nombre des observations est inférieur à une valeur prédéfinie.
- On rappelle que la construction de l'arbre se termine lorsque tous les nœuds terminaux ont été attribués à une classe.

Règles d'arrêt et d'attribution

On attribue la classe :

$$Y(t) = \arg \max_{k=1,\dots,K} p(k|t),$$

à un nœud terminal.

Supposons que $Y(t) = k$

- Le nombre d'individus appartenant au groupe k dans la feuille t assignée à $\mathcal{G}_{k'} (k' \neq k)$ sera noté $n_{k'k}(t)$.
- Le nombre total d'individus appartenant au groupe k assignée à $\mathcal{G}_{k'} (k' \neq k)$ par l'arbre sera noté $n_{k'k}$.

Règles d'arrêt et d'attribution

- La démarche de construction précédente fournit l'arbre maximal T_{max} .
- T_{max} peut conduire à un modèle de prévision très instable car fortement dépendant des échantillons qui ont permis la construction de l'arbre.
- C'est une situation de sur-ajustement.
- Solution : procédure d'élagage de l'arbre au lieu de règles d'arrêt.

Définition 2.

Taux d'erreurs de classification du nœud t :

$$R(t) = \sum_{k=1, k \neq Y(t)}^K p(k|t),$$

avec

$$Y(t) = \arg \max_{k=1, \dots, K} p(k|t) = \text{classe attribuée au noeud } t .$$

Définition 3.

Soit $\tilde{T} = \{t_1, t_2, \dots, t_m\}$ les nœuds terminaux d'un arbre T . Le taux d'erreurs de classification de T est :

$$R(T) = \sum_{i=1}^m \frac{N(t_i)}{n} R(t_i).$$

De plus, on définit $\text{taille}(T) = \text{Card}(\tilde{T})$ et $\alpha > 0$ un paramètre de complexité (CP) qui nous aide à imposer une certaine pénalité pour les grands arbres. Ainsi, le taux d'erreurs pénalisé de T est donné par :

$$R_\alpha(T) = R(T) + \alpha \text{taille}(T).$$

Remarque.

En général, $R(T)$ est un mauvais estimateur de $\rho(T)$ car trop optimiste. Pour le voir, imaginons qu'on laisse pousser l'arbre jusqu'à ce que chaque feuille ne contienne d'une seule observation. Alors $n_{kk} = 1$ pour toutes les feuilles, de sorte que $R(T) = 0$.



Il existe plusieurs façons de contourner ce problème. Nous en évoquons une ici rapidement. Elle consiste à partitionner l'échantillon \mathcal{E} en deux : \mathcal{E}_{app} et \mathcal{E}_{est} .

- \mathcal{E}_{app} = échantillon d'apprentissage (avec n_{app} données) sur lequel on construit l'arbre T_{app} .
- \mathcal{E}_{est} = échantillon d'estimation (avec n_{est} données) qu'on passe dans l'arbre pour obtenir les estimateurs $p_{k'k}^{est}$ est et $R^{est}(T)$.

Noter que ces dernières quantités estiment les $\rho_{k'k}$ de T_{app} et $\rho(T_{app})$ et non pas $\rho_{k'k}$ de T et $\rho(T)$.

Cependant, si n est grand, n_{app} et n_{est} le seront aussi. Ainsi $T \simeq T_{app} \implies p_{k'k}^{est} \approx \rho_{k'k}$ de T et $R^{est}(T) \approx \rho(T)$.

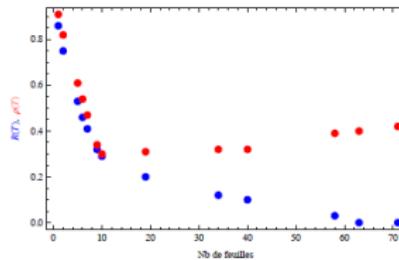
A tout le moins, le problème évoqué plus haut ($R(T) = 0$) disparaît. On appelle cette méthode la validation externe ou encore out-of sample.

Dans la suite on va appeler T_{max} l'arbre maximal et T_0 l'arbre minimal formée juste de la racine :

- Entre T_0 et T_{max} , existe-t-il des arbres T qui donnent d'aussi bons résultats, voire même de meilleurs (en terme de $\rho(T)$ = probabilité globale d'assignation incorrecte) que T_{max} tout en étant plus simples (moins de niveau) ?
- Si oui, l'utilisation de cet arbre s'imposerait tant pour sa taille que son $\rho(T)$.
- On veut donc déterminer si, entre T_0 et T_{max} , il existe un arbre T tel que $\rho(T) \leq \rho(T_{max})$.

Élagage et coût de complexité

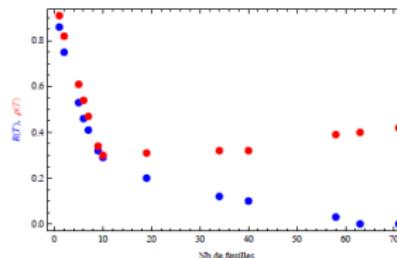
L'exemple (artificiel) suivant montre la situation typique dans le cas d'un ensemble d'arbres se situant "entre" T_0 et T_{max} de nombre différent de feuilles :



- Il y a bien un arbre qui minimise les valeurs de $\rho(T)$ et cet arbre, appelons le T_{opt} , est beaucoup plus petit que T_{max} .
- Quand T est "proche" de T_{max} , $R(T) \simeq 0$ et on a faussement l'impression que l'arbre est excellent en terme de probabilité globale d'assignation incorrecte (i.e. excellent pouvoir de classification).
- Quand T est "proche" de T_0 , $R(T)$ est proche de $\rho(T)$.

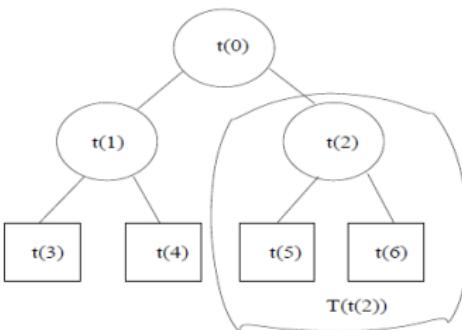
Élagage et coût de complexité

L'exemple (artificiel) suivant montre la situation typique dans le cas d'un ensemble d'arbres se situant "entre" T_0 et T_{max} de nombre différent de feuilles :



- Si on connaissait les $\rho(T)$, on pourrait déterminer ce T_{opt} . Mais mettre la main sur les $\rho(T)$ est une cause perdue.
- En solution, construire un estimateur "honnête" (c-à-d sans biais) de $\rho(T)$, notons-le $R^{hon}(T)$,
- Les points $R^{hon}(T)$ forment un graphique semblable à celui des $\rho(T)$,
- On pourrait utiliser ces $R^{hon}(T)$ pour déterminer un estimateur \widehat{T}_{opt} de T_{opt} .

Élagage et coût de complexité



- On note par $T(t)$ le sous-arbre avec la racine en t .
- Élaguer d'un arbre T sa branche $T(t)$ consiste à remplacer la branche $T(t)$ par un noeud terminal à t . L'arbre élagué (qui peut-être noté $T - T(t)$) est appelé un sous-arbre de T .
- Il est naturel d'imposer que la suite d'arbres qu'on cherche à créer soit une suite de sous-arbres les uns des autres. Plus précisément, on se restreint au cas où $T_0 \ll T_1 \ll T_2 \ll \dots \ll T_{max}$.
- À la fin du processus, on pourra repasser dans la suite de sous-arbres pour en déterminer le T_{opt} (ou plutôt \hat{T}_{opt})

Remarque.

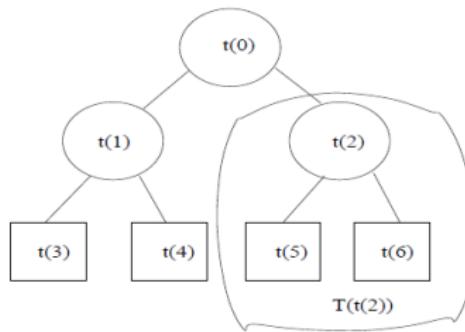
Soit $T_k \ll T_{k+1}$ alors $R(T_{k+1}) \leq R(T_k)$.

⇒ L'utilisation de $R(T)$ pour déterminer T_{opt} ne peut fonctionner : plus on splitte, mieux on croit, à tort, bien faire. Il faut donc "corriger" la dérive de $R(T)$ vers 0. Une façon classique est de lui ajouter un terme pénalisant la complexité (ç-à-d. le nombre de feuilles/nœuds terminaux de l'arbre).

Si on dispose d'un estimateur "honnête" (e.g. sans biais), disons $R^{hon}(T)$ de $\rho(T)$, alors il est raisonnable d'estimer le meilleur arbre \widehat{T}_{opt} par

$$\widehat{T}_{opt} = \operatorname{argmin}_{T_j} R^{hon}(T_j).$$

Élagage et coût de complexité



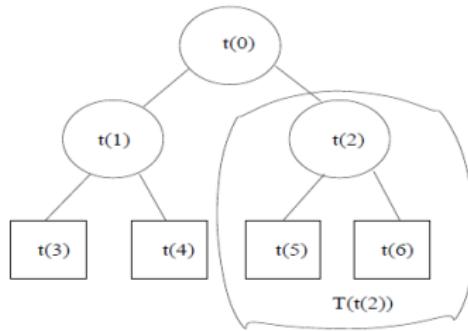
- L'erreur du sous-arbre $T(t_2)$ et l'erreur du nœud t_2 sont respectivement :

$$\begin{aligned} R_\alpha(T(t_2)) &= R(T(t_2)) + \alpha \text{taille}(T(t_2)), \\ &= \frac{1}{n'} [N(t_5)R(t_5) + N(t_6)R(t_6)] + 2\alpha, \end{aligned}$$

avec, n' = taille de l'échantillon dans le sous-arbre de racine t_2 .

$$\begin{aligned} R_\alpha(t_2) &= R(t_2) + \alpha, \\ &= \sum_{k=1, k \neq Y(t_2)}^K p(k|t_2) + \alpha. \end{aligned}$$

Élagage et coût de complexité



L'élagage en vaut la peine si :

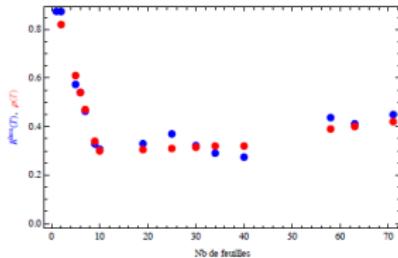
$$R_\alpha(t_2) \leq R_\alpha(T(t_2)) \Leftrightarrow g(t_2, T) \equiv \frac{R(t_2) - R(T(t_2))}{\text{taille}(T(t_2)) - 1} \leq \alpha.$$

La fonction $g(t, T)$ peut être calculée pour chaque nœud interne de l'arbre.

- Algorithme de coupe du maillon faible :
 - Commencer avec l'arbre complet T .
 - Pour chaque nœud non-terminal $t \in T$ calculer $g(t, T)$, et trouver $t_1 = \arg \min_{t \in T} g(t, T)$. On pose $\alpha_1 = g(t_1, T)$.
 - Définir un nouvel arbre T_1 en enlevant la branche partant de t_1 .
 - Trouver le maillon faible de T_1 et procéder comme pour T .
- Le résultat : une suite décroissante d'arbres et l'arbre final sera déterminé en fonction de son pouvoir de prédiction. (Out-of sample estimation/Cross Validation)

Élagage et coût de complexité

Typiquement, le graphe de $R^{hon}(T_j)$ en fonction de la taille de T_j prend la forme suivante :



La position du minimum est instable car $R^{hon}(T)$ est une v.a. et pourrait varier beaucoup selon le choix des sous-échantillons (de la validation croisée). Il importe donc de trouver une façon de stabiliser le choix de \widehat{T}_{opt} .

D'où la règle d'un écart-type qui :

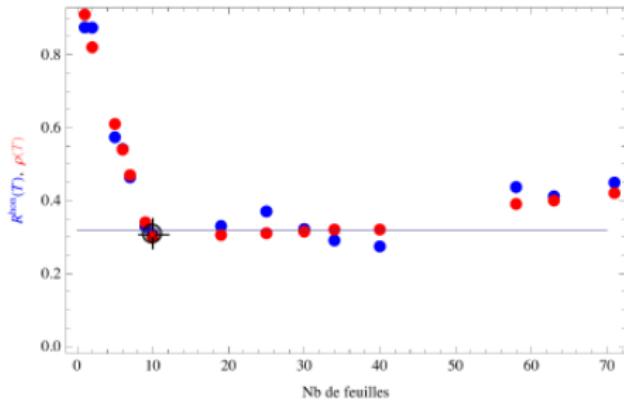
- ➊ Cherche à réduire l'instabilité notée plus haut,
- ➋ Cherche à sélectionner l'arbre le plus simple dont le taux de mauvaise classification est comparable à celui de \widehat{T}_{opt} .
- ➌ Ici le terme comparable réfère à des quantités aléatoires pour lesquelles on ne peut pas dire qu'elles sont statistiquement différentes.
- ➍ Cette règle mène à choisir comme meilleur arbre le plus petit arbre \widehat{T}_{opt}^* (en nombre de feuilles) tel que

$$R^{hon}\left(\widehat{T}_{opt}^*\right) \leq R^{hon}\left(\widehat{T}_{opt}\right) + \sqrt{\frac{R^{hon}\left(\widehat{T}_{opt}\right)\left(1 - R^{hon}\left(\widehat{T}_{opt}\right)\right)}{n_{est}}}$$

où n_{est} est la taille de l'échantillon d'estimation.

Élagage et coût de complexité

Ainsi en appliquant cette approche à notre cas l'arbre \widehat{T}_{opt}^* est celui à 10 feuilles.



Application 1 :

Données “weather”

Application 1

Le tableau est composé de 14 observations, il s'agit d'expliquer le comportement des individus par rapport à un jeu {jouer, ne pas jouer} à partir des prévisions météorologiques :

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
1	soleil	Chaud	Élevée	non	non
2	soleil	Chaud	Élevée	oui	non
3	couvert	Chaud	Élevée	non	oui
4	pluie	Tiède	Élevée	non	oui
5	pluie	Fraîche	Normale	non	oui
6	pluie	Fraîche	Normale	oui	non
7	couver	Fraîche	Normale	oui	oui
8	soleil	Tiède	Élevée	non	non
9	soleil	Fraîche	Normale	non	oui
10	pluie	Tiède	Normale	non	oui
11	soleil	Tiède	Normale	oui	oui
12	couver	Tiède	Élevée	oui	oui
13	couver	Chaud	Normale	non	oui
14	pluie	Tiède	Élevée	oui	non

Application 1

- On commence par calculer les gains d'information pour chaque attribut :

Variable	Variation d'impureté
Ensoleillement	
Température	
Humidité	
Vent	

- Donc, la racine de l'arbre de décision est la variable présentant la variation d'impureté la plus élevée.
- On refait le calcul de l'étape précédente pour chacune des différentes valeurs de la variable sélectionnée comme racine de l'arbre.

Application 1 bis

Cas d'un attribut numérique :

Numéro	soleil/nuage	Température	Humidité	Vent	Jouer
1	soleil	27.5	85	non	non
2	soleil	25	90	oui	non
3	couvert	26.5	86	non	oui
4	pluie	20	96	non	oui
5	pluie	19	80	non	oui
6	pluie	17.5	70	oui	non
7	couvert	17	65	oui	oui
8	soleil	21	95	non	non
9	soleil	16.5	70	non	oui
10	pluie	22.5	80	non	oui
11	soleil	22.5	70	oui	oui
12	couvert	21	90	oui	oui
13	couvert	25.5	75	non	oui
14	pluie	20.5	91	oui	non

Application 1 bis

Afin de préciser le seuil pour lequel on peut couper une variable numérique :

- Trier la variable numérique dans l'ordre croissant :

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
9	soleil	16.5	70	non	oui
7	couvert	17	65	oui	oui
6	pluie	17.5	70	oui	non
5	pluie	19	80	non	oui
4	pluie	20	96	non	oui
14	pluie	20.5	91	oui	non
8	soleil	21	95	non	non
12	couvert	21	90	oui	oui
10	pluie	22.5	80	non	oui
11	soleil	22.5	70	oui	oui
2	soleil	25	90	oui	non
13	couvert	25.5	75	non	oui
3	couvert	26.5	86	non	oui
1	soleil	27.5	85	non	non

- Ne pas séparer deux observations successives ayant la même classe.
- Si on coupe entre deux valeurs v et w ($v < w$) le seuil s est fixe à v ou aussi $s = \frac{v+w}{2}$.
- Choisir s de telle manière que le gain d'information soit maximal.

Application 1 bis

Pour la variable “Température” les valeurs possibles de s sont :

17.25; 18.25; 20.25; 21; 23.75; 25.25; 27

- Pour $s = 17.25$ le gain de l'information est :

$$\begin{aligned}\Phi_{\text{Temperature}}(s = 17.25, t_0) &= \Delta \text{Imp}(t_0) = \text{Imp}(t_0) - \pi_L \text{Imp}(t_L) - \pi_R \text{Imp}(t_R), \\ &= \left[-\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) \right] - \frac{2}{14} \times 0 \\ &\quad - \frac{12}{14} \times \left[-\frac{7}{12} \log_2 \left(\frac{7}{12} \right) - \frac{5}{12} \log_2 \left(\frac{5}{12} \right) \right], \\ &= 0.1.\end{aligned}$$

- De la même manière, en fonction du seuil, le gain d'information est alors :

Seuil = s	$\Phi_{\text{Temperature}}(s, t_0)$
17.25	0.1
18.25	...
20.25	...
21	...
23.75	...
25.25	...
27	...

Remarque.

Nous avons montré comment choisir le seuil pour un attribut numérique donné.

⇒ On applique cette méthode pour chaque attribut numérique et on détermine pour chacun un seuil produisant un gain d'information maximal.

⇒ Le gain d'information associé à chacun des attributs numériques est celui pour lequel le seuil entraîne un maximum.

⇒ L'attribut choisi pour effectuer la coupe est celui, parmi les numériques et les catégoriels, qui produit un gain d'information maximal.

Application 2 :

Cancer de prostate en stage C

Application 2

Données : On considère 7 variables sur 146 patients au stade C du cancer de prostate :

- pgtime : dernier suivi (années).
- pgstat : le statut au dernier suivi (1 = progression, 0 = pas de progression).
- age : l'age du patient.
- eet : thérapie endocrinienne précoce (1=no, 2=yes).
- grade : grade de la tumeur (1-4) - Farrow system.
- gleason : grade de la tumeur - Gleason system.
- ploidy : l'état de la tumeur diploid/tetraploid/aneuploid.

Paramètres par défaut :

- $\text{minsplit} = 20$: nombre minimal d'observations dans un nœud pour lequel la coupe est calculée.
- $\text{minbucket} = \text{minsplit}/3$: nombre minimal d'observation dans un nœud terminal.
- $\text{cp} = 0.01$: paramètre de complexité.

Sous R avec la fonction rpart on obtient :

- Le tableau de CP est imprimé du plus petit arbre (0 coupes) au plus grand (5 coupes pour les données de cancer).
- rel error : l'erreur relative $R(T)/R(T_0)$.
- xerror : l'erreur calculée par validation croisée.
- xstd : l'écart-type de l'erreur calculée par validation croisée.