Arbres de Décision

Arthur Mussard

December 1, 2023

Contents

1 Arbres

Les arbres de décision sont des modèles puissants et polyvalents utilisés dans le domaine de l'apprentissage automatique. Cette section examine en détail les différentes facettes des arbres de décision, allant de leur structure de base à leurs applications spécifiques en régression et classification.

1.1 Terminologie

Un arbre de décision est une représentation arborescente d'un processus de décision.

1.2 Terminologie

La structure d'un arbre de décision comprend une racine, des nœuds internes représentant des caractéristiques, des branches indiquant des décisions, et des feuilles contenant les résultats ou les prédictions.

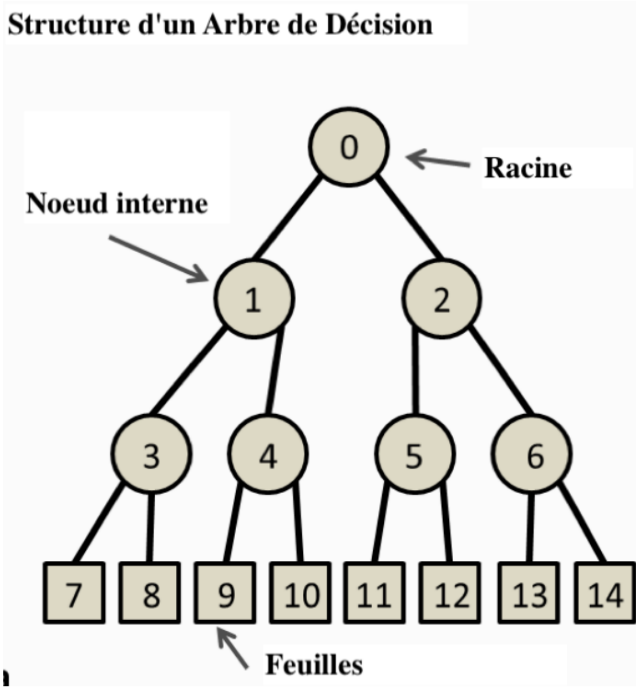


Figure 1: Structure d'un arbre de décision.

1.3 Arbre de décision binaire

Les arbres de décision binaires sont des arbres dans lesquels chaque nœud a 2 ou 0 fils. Cela signifie que chaque nœud de l'arbre prend une décision binaire basée sur une caractéristique spécifique. Un nœud peut se diviser en deux branches,

représentant deux résultats possibles, ou rester une feuille sans branches pour indiquer une prédiction ou un résultat final.

1.4 Arbre de décision binaire sur $\mathbb{R}^2$

Voici un exemple d'arbre de décision binaire dans l'espace $\mathbb{R}^2$. Dans cet exemple, chaque nœud de l'arbre est associé à une variable de test $x(j)$ et à un seuil t:

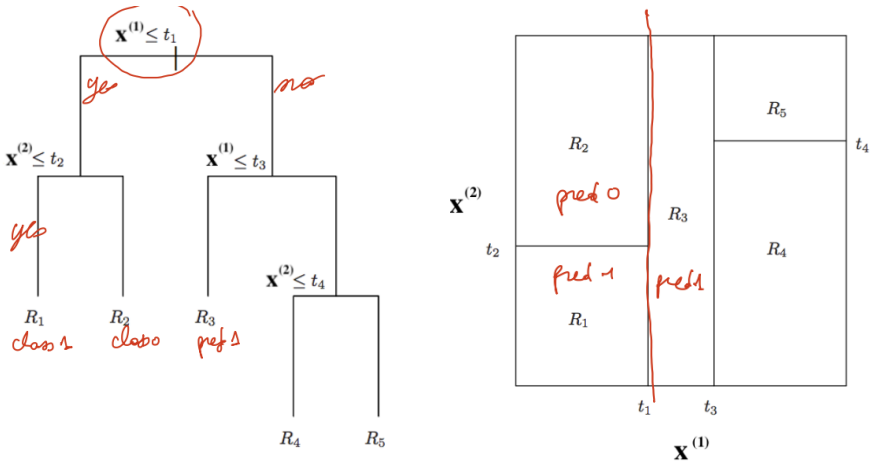


Figure 2: Exemple d'arbre de décision binaire sur $\mathbb{R}^2$.

L'arbre divise l'espace $\mathbb{R}^2$ en régions distinctes, chaque région étant associée à une classe de sortie.

1.5 Types d'arbres de décision

Dans ce cours, nous allons nous intéresser qu'à un seul type d'arbre :

- **Arbres Binaires (vs Arbres avec Plus de 2 Fils) :** Les arbres binaires ont chaque nœud avec soit 2 fils, soit aucun, simplifiant la structure mais il existe des arbres avec plus de 2 fils qui peuvent offrir une plus grande complexité en permettant plusieurs branches à chaque nœud.
- **Décisions sur une Seule Variable :** Chaque nœud prend une décision basée sur une seule variable, simplifiant le processus de décision.

- **Décisions sur les Variables Continues :** Les décisions pour les variables continues sont généralement de la forme $x(j) \geq t$.
- **Décisions pour les Variables Discrètes :** Pour les variables discrètes, les décisions partitionnent les valeurs en deux groupes distincts.

Il existe cependant d'autres types d'arbres que nous n'allons pas détailler dans ce cours:

- **Arbres de Décision Obliques :** Ces arbres permettent des décisions inclinées, ce qui peut être utile pour des relations complexes entre variables.
- **Binary Space Partition Trees (BSP Trees) :** Ces arbres sont utilisés pour partitionner l'espace, souvent utilisés dans la modélisation de géométrie.
- **Sphere Trees (Arbres Sphériques) :** Ces arbres sont basés sur des sphères pour définir des régions de décision, adaptés à certains types de données.

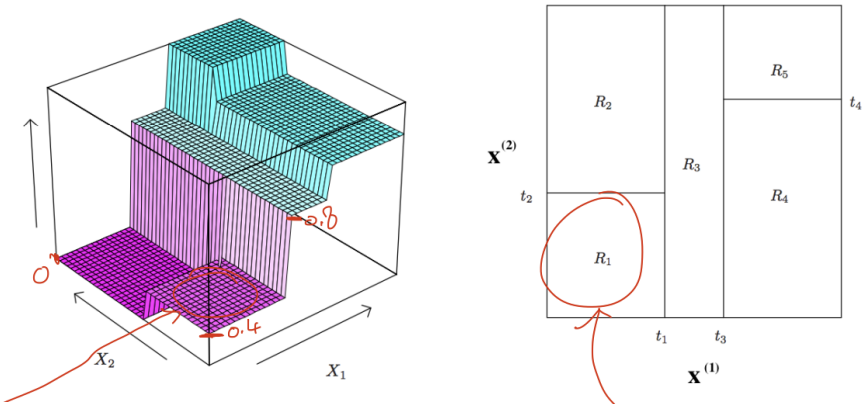
2 Arbres de Régression

Dans le contexte de la régression, les arbres de décision sont utilisés pour modéliser des relations complexes entre les caractéristiques et une variable de sortie continue. Chaque feuille de l'arbre contient une valeur numérique, représentant la prédiction pour une observation donnée.

2.1 Arbre de régression sur $\mathbb{R}^2$

Considérons un arbre binaire sur $X = \mathbb{R}^2$. Un exemple concret peut être illustré avec $x = (x(1), x(2))$.

En utilisant un arbre de régression sur $\mathbb{R}^2$, nous construisons un modèle pour représenter une fonction dans un espace bidimensionnel. Cela se fait en décomposant l'espace en segments délimités par les décisions prises par l'arbre. Chaque segment, défini par un nœud de l'arbre, représente une région spécifique de l'espace où une fonction constante est utilisée pour modéliser les données. Ainsi, l'arbre de régression divise l'espace $\mathbb{R}^2$ en régions distinctes, permettant une modélisation adaptative de la fonction sur différentes parties de cet espace.



2.2 Apprendre un Arbre de Régression

L'apprentissage d'un arbre de régression se déroule en plusieurs étapes. Tout d'abord, l'espace X est partitionné en régions distinctes $R_1, R_2, \ldots, R_M$, où $X = R_1 \cup R_2 \cup \ldots \cup R_M$ et $R_i \cap R_j = \emptyset$ pour $i \neq j$. Ensuite, pour chaque région R_m, le nombre d'observations N_m est calculé comme $N_m = |\{i : x_i \in R_m\}|$, avec x_i représentant les observations. Dans chaque région, une fonction constante est modélisée. Avec la partition $\{R_1, \ldots, R_M\}$, la prédiction finale dans un arbre de régression est formulée comme :

$$f(x) = \sum_{m=1}^{M} c_m \cdot \mathbb{1}(x \in R_m)$$

Lorsque la partition est donnée, le choix des coefficients $c_1, \ldots, c_M$ se fait en optimisant la fonction objectif du modèle. Une approche courante est de minimiser la somme des carrés des erreurs entre les valeurs prédites $f(x_i)$ et les valeurs cibles réelles y_i. Cela peut être formulé comme un problème d'optimisation où les coefficients sont ajustés pour minimiser la perte globale :

$$c_1, \ldots, c_M = argmin \sum_{i=1}^{n} (f(x_i) - y_i)^2$$

$$c_1, \ldots, c_M = argmin \sum_{i=1}^{n} (\sum_{m=1}^{M} c_m \cdot \mathbb{1}(x \in R_m) - y_i)^2)$$

$$c_1, \ldots, c_M = argmin \sum_{m=1}^{M} \sum_{i \in R_m} (c_m - y_i)^2$$

$$\forall m, c_m = argmin \sum_{i \in R_m} (c_m - y_i)^2 = average(y_i | i \in R_m)$$

Pour la fonction de perte quadratique $L(\hat{y}, y) = (\hat{y} - y)^2$, l'application de l'ERM dans un nœud spécifique R_m d'un arbre de régression implique le calcul du prédicteur optimal $\hat{c}_m$ pour minimiser la perte quadratique moyenne.

La valeur optimale de $\hat{c}_m$ est obtenue en calculant la moyenne des valeurs cibles y_i dans la région R_m, formulée comme :

$$\hat{c}_m = \frac{1}{|\{i : x_i \in R_m\}|} \sum_{\{i : x_i \in R_m\}} y_i$$

Cela peut également être exprimé comme l'argument minimum de la somme des pertes quadratiques pour chaque observation dans la région R_m :

$$\hat{c}_m = argmin_{\hat{y}} \sum_{\{i : x_i \in R_m\}} (\hat{y} - y_i)^2$$

La perte associée à ce nœud est la somme des pertes quadratiques pour chaque observation dans la région R_m, exprimée comme :

$$\sum_{\{i : x_i \in R_m\}} (\hat{y} - y_i)^2 = \sum_{\{i : x_i \in R_m\}} (\hat{c}_m - y_i)^2 = N_m \cdot Var(\hat{y})$$

Ici, N_m est le nombre d'observations dans la région R_m et $Var(\hat{y})$ est la variance des valeurs prédites dans la région R_m.

2.3 Noeud Racine et Variables Réelles

Considérons l'arbre de régression suivant :

- Soit $x = (x(1), \ldots, x(d)) \in \mathbb{R}^d$. (d variables)
- Variable de test $x(j)$.
- Seuil $s \in \mathbb{R}$.
- Partition basée sur $x(j)$ et s:
  - $R_1(j, s) = \{x \mid x(j) \leq s\}$
  - $R_2(j, s) = \{x \mid x(j) > s\}$

Lorsque les variables sont continues, au niveau du nœud racine, pour chaque variable $x(j)$ et seuil s, les prédictions $\hat{c}_1(j, s)$ et $\hat{c}_2(j, s)$ sont calculées comme

les moyennes des valeurs cibles (y_i) dans les régions $R_1(j,s)$ et $R_2(j,s)$, respectivement.

$$\hat{c}_1(j,s) = average(y_i \mid x_i \in R_1(j,s))$$

$$\hat{c}_2(j,s) = average(y_i \mid x_i \in R_2(j,s))$$

L'objectif est de trouver les valeurs de j et s qui minimisent la fonction de perte quadratique $L(j,s)$, définie comme la somme des variances pondérées des valeurs cibles dans les deux régions :

$$L(j,s) = \sum_{i:x_i \in R_1(j,s)} (y_i - \hat{c}_1(j,s))^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{c}_2(j,s))^2$$

Cela équivaut à minimiser la somme des variances pondérées $N_1 \cdot Var(\hat{y}_1)$ et $N_2 \cdot Var(\hat{y}_2)$ dans les deux régions respectives.

La recherche des valeurs optimales de j et s est une étape cruciale dans la construction de l'arbre de décision et est généralement effectuée par des méthodes d'optimisation.

2.4 Trouver le Seuil

Pour trouver le seuil lorsqu'on a choisi la variable de test $x(j)$, on suppose que les valeurs $x(j)_1, \ldots, x(j)_N$ sont triées en ordre croissant. Traditionnellement, le choix du seuil se fait entre deux valeurs consécutives $s_j \in \left[\frac{x(j)_i + x(j)_{i+1}}{2}\right]$ pour $i = 1, \ldots, N-1$.

Ainsi, on teste $N-1$ seuils différents entre chaque paire de valeurs consécutives. Ce processus permet de déterminer la meilleure division de l'espace des variables continues.

La complexité pour trouver le noeud et le seuil dans ce contexte est $O(dN^2)$, où d est le nombre de variables et N est le nombre d'observations. Cette complexité s'explique par la nécessité de tester tous les seuils possibles pour chaque variable, ce qui peut devenir coûteux avec un grand nombre de variables et d'observations. On peut améliorer la recherche de seuil pour obtenir une complexité de $O(dNlog(N))$.

2.5 Continuer l'Apprentissage de l'Arbre Récursivement

Une fois les régions R_1 et R_2 déterminées, l'apprentissage de l'arbre se poursuit de manière récursive. On choisit le meilleur split (j,s) dans R_1 et le meilleur split (j',s') dans R_2. Ce processus continue jusqu'à ce que les critères d'arrêt soient atteints.

Les critères d'arrêt incluent :

- Atteindre une profondeur maximale de l'arbre.
- Avoir un nombre minimum d'observations dans chaque feuille.
- Atteindre une pprécision suffisante dans les feuilles, basée sur la variance des valeurs cibles.

Ces critères assurent que l'arbre ne devient pas trop complexe et permettent de prévenir le surajustement aux données d'entraînement.

2.6 Contrôler la Complexité de l'Arbre

Pour prévenir le surapprentissage (si l'arbre est trop grand, chaque exemple x_i a sa propre partition) ou l'underfitting (si l'arbre est trop petit), on peut limiter sa profondeur, ne poursuivre les divisions que sur les nœuds contenant un nombre minimum d'exemples, ou opter pour des techniques d'élagage à postériori, comme dans CART (Breiman et al., 1984), consistant à construire un arbre très grand jusqu'à ce que chaque région ait un nombre minimum de points, puis à élaguer l'arbre de manière gloutonne du bas vers le haut, tout en évaluant la performance sur un ensemble de test, poursuivant l'élagage tant que la performance estimée ne décroît pas.

3 Arbres de Décision pour la Classification

Lorsqu'il s'agit de la classification, les arbres de décision sont utilisés pour assigner des observations à des classes spécifiques.

3.1 Arbres pour la Classification (avec perte 0/1)

Considérons $Y = \{1, \ldots, K\}$. En suivant un raisonnement similaire à celui précédent, supposons que nous ayons déjà les régions R_i.

Le nœud m représente la région R_m avec N_m exemples. La proportion d'exemples de classe $k \in Y$ dans R_m est donnée par:

$$\hat{\eta}_{m,k} = \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} 1(y_i = k)$$

.

Si nous prédisons la classe k au nœud m, alors le taux d'erreur sur les exemples d'apprentissage de R_m sera $1 - \hat{\eta}_{m,k}$. Ainsi, pour minimiser le taux d'erreur (perte 0/1), la classe prédite pour le nœud m sera $\hat{y}(m) = argmax_k \hat{\eta}_{m,k}$.

3.2 Exemple

Considérons un exemple d'arbre de classification avec les variables : `Student`, `Credit Rating`, et la classe `Buy PDA`.

Student	**Credit Rating**	**Class: Buy PDA**
No	Fair	No
No	Excellent	No
No	Fair	Yes
No	Fair	Yes
Yes	Fair	Yes
Yes	Excellent	No
Yes	Excellent	Yes
No	Excellent	No

1. Test selon la variable `Student`:

- Sous-région 1 (`Student = Yes`):

$$\hat{\eta}_{1,No} = \frac{1}{3}, \quad \hat{\eta}_{1,Yes} = \frac{2}{3}, N_1 = 3, \hat{c}_1 = Yes$$

- Sous-région 2 (`Student = No`):

$$\hat{\eta}_{2,No} = \frac{3}{5}, \quad \hat{\eta}_{2,Yes} = \frac{2}{5}, N_2 = 5, \hat{c}_2 = No$$

-

$$Erreur = \frac{N_1}{N} \times \frac{1}{3} + \frac{N_1}{N} \times \frac{2}{5} = \frac{3}{N} = \frac{3}{8}$$

2. Test selon la variable `Credit Rating`: - Sous-région 1 (`Credit Rating = Fair`):

$$\hat{\eta}_{1,No} = \frac{1}{4}, \quad \hat{\eta}_{1,Yes} = \frac{3}{4}, N_1 = 4, \hat{c}_1 = Yes$$

- Sous-région 2 (`Credit Rating = No`):

$$\hat{\eta}_{2,No} = \frac{3}{4}, \quad \hat{\eta}_{2,Yes} = \frac{1}{4}, N_2 = 4, \hat{c}_2 = No$$

-

$$Erreur = \frac{N_1}{N} \times \frac{1}{4} + \frac{N_1}{N} \times \frac{1}{4} = \frac{2}{N} = \frac{1}{4}$$

Ainsi la classification est meilleur selon la variable `Credit Rating` car l'erreur est la plus faible.

3.3 Choix des Prédictions en Classification

En régression, avec la partition $\{R_1, \ldots, R_M\}$, la prédiction finale est $f(x) = \sum_{m=1}^{M} c_m \mathbb{I}(x \in R_m)$. Pour choisir $c_1, \ldots, c_M$ avec la fonction de perte quadratique $(\hat{y}, y) = (\hat{y} - y)^2$, l'ERM propose

$$\hat{c}_m = argmin_{\hat{y}} \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} (\hat{y} - y_i)^2 = average(y_i | x_i \in R_m)$$

.

Pour une perte à probabilité de classe, si $Y = \{0, 1\}$, le nœud m représente la région R_m avec N_m exemples. La proportion d'exemples de classe k dans R_m est $\hat{\eta}_m = \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} 1(y_i = k)$. L'ERM propose la prédiction

$$\hat{c}_m = argmin_{\hat{y}} \frac{1}{N_m} \sum_{\{i: x_i \in R_m\}} (\hat{y} - y_i)^2$$

. Pour une perte PC-loss propre, telle que la cross-entropie, $\hat{c}_m = \hat{\eta}_m$, et la valeur de la perte en R_m sera

$$\sum_{\{i: x_i \in R_m\}} \mathcal{L}(\hat{\eta}_m, y_i)$$

.

Si la fonction de perte est la cross-entropy $\mathcal{L}(\hat{\eta}, y) = y \log(\hat{\eta}) - (1-y) \log(1 - \hat{\eta})$, la valeur de la perte en R_m sera

$$-N_m(\hat{\eta}_m \log(\hat{\eta}_m) + (1 - \hat{\eta}_m) \log(1 - \hat{\eta}_m))$$

, correspondant à l'Entropie de Shannon.

Si l'objectif est de prédire une classe plutôt qu'une probabilité, la prédiction sera $\hat{y}(R_m) = 1$ si $\hat{\eta}_m > \frac{1}{2}$, sinon $\hat{y}(R_m) = 0$. Remarque : pour toute perte PC, la valeur de cette perte dans la région R_m est l'entropie généralisée $\mathcal{H}(\hat{\eta}_m)$.

3.4 Mesures d'Impureté des Nœuds

Considérons une classification binaire où p représente la fréquence relative de la classe 1. Trois mesures d'impureté pour les nœuds en fonction de p sont couramment utilisées:

- L'indice de Gini évalue l'impureté en considérant la probabilité qu'une paire d'éléments choisis aléatoirement appartienne à des classes différentes. Plus l'indice de Gini est élevé, plus le nœud est impur.

- L'entropie, basée sur la théorie de l'information, mesure l'incertitude associée à la distribution des classes. Plus l'entropie est élevée, plus le nœud est impur.
- L'erreur de classification est une mesure simple qui évalue la probabilité d'une classification incorrecte. Plus cette probabilité est élevée, plus le nœud est impur.

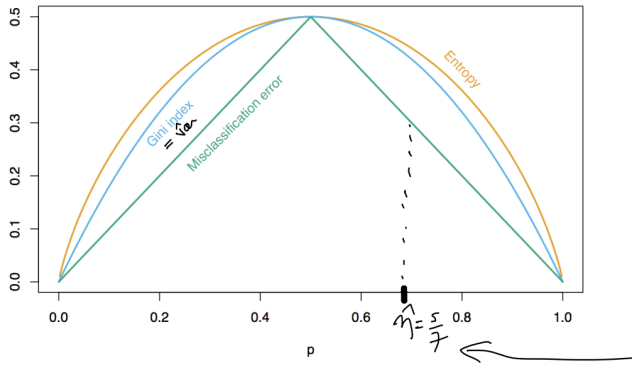


Figure 3: Mesures d'impureté

Considérons un nœud terminal m représentant la région R_m avec N_m observations. Trois mesures d'impureté de nœud pour le nœud terminal m sont souvent utilisées.

La **Erreur de Classification (H0/1)** cherche à minimiser la probabilité de mauvaise classification dans le nœud m. Elle est définie comme

$$H0/1(\hat{\eta}) = \min_k (1 - \hat{\eta}_{m,k})$$

.

L'**Indice de Gini (HGini)**, mesurant l'impureté en considérant la probabilité que deux éléments choisis aléatoirement appartiennent à des classes différentes, est donné par

$$HGini(\hat{\eta}) = \sum_{k=1}^{K} \hat{\eta}_{m,k}(1 - \hat{\eta}_{m,k})$$

.

L'**Entropie ou déviance (HCE)**, équivalent à l'utilisation du gain d'information, est définie comme

$$HCE(\hat{\eta}) = -\sum_{k=1}^{K} \hat{\eta}_{m,k} \log(\hat{\eta}_{m,k})$$

.

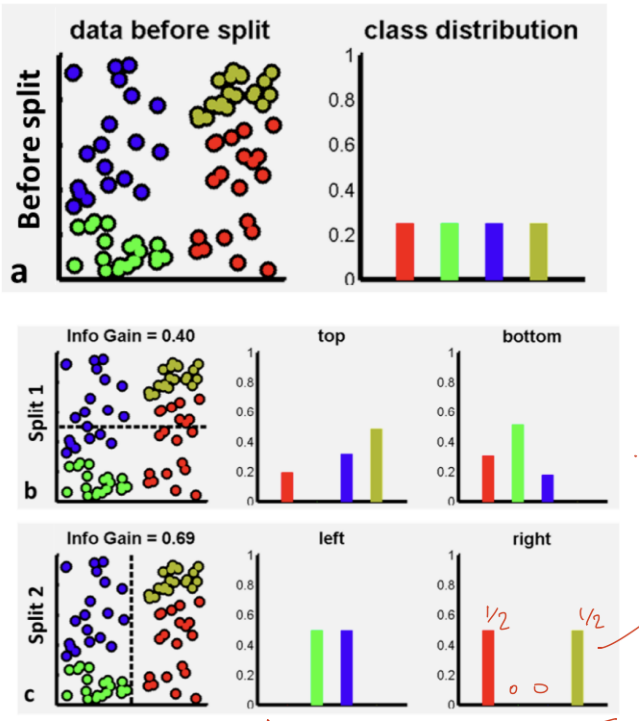


Figure 4: Pré Split et Splitting

Lors du splitting des nœuds, le processus précis se déroule comme suit.

Considérons R_1 et R_2 comme les régions correspondant à un potentiel split de nœud. Supposons que nous ayons N_1 points dans R_1 et N_2 points dans R_2. Soient $H(R_1)$ et $H(R_2)$ les mesures d'impureté généralisée (les mesures d'impureté du nœud).

Ensuite, nous cherchons le split qui minimise la moyenne pondérée des impuretés des nœuds :

$$EntropyMoy = \frac{N_1}{N} \cdot H(R_1) + \frac{N_2}{N} \cdot H(R_2)$$

.

Pour construire l'arbre, Gini et Entropy semblent être plus efficaces. Ils favorisent la création de nœuds plus purs, plutôt que de simplement minimiser le taux de mauvaise classification. Il est important de noter qu'une bonne division peut ne pas du tout changer le taux de mauvaise classification.

Considérons un problème à deux classes avec 4 observations dans chaque classe. Lors de deux splits possibles, le premier ayant des régions (3,1) et (1,3), et le second avec des régions (2,4) et (2,0), le taux de mauvaise classification reste le même pour les deux splits. Cependant, les critères Gini et Entropy privilégient le deuxième split, qui crée une région pure.

Ainsi, pour construire l'arbre, les mesures Gini et Entropy sont privilégiées en raison de leur propension à favoriser des nœuds plus purs plutôt que simplement minimiser le taux de mauvaise classification.