TD Arbres de Décision

Supplément Théorique Approfondi

Master 2 Banque Finance Assurance
Paris Dauphine

1. Critères de Division : Gini et Entropie

1.1 Indice de Gini

Formule : $Gini(t) = 1 - \Sigma p_i^2$

D'où ça vient ?
Le Gini mesure la **probabilité de mal classer** une observation tirée au hasard si on lui attribue aléatoirement une étiquette selon la distribution du nœud.

Les p_i :
• p_i = proportion de la classe i dans le nœud t
• $p_i = n_i / n_t$ avec n_i = nombre d'observations de classe i, n_t = nombre total d'observations dans le nœud

Exemple concret :

Situation	Calcul
Nœud avec 100 clients	70 sans défaut, 30 avec défaut
p₀ (sans défaut)	70/100 = 0.7
p₁ (avec défaut)	30/100 = 0.3
Gini	$1 - (0.7^2 + 0.3^2) = 1 - (0.49 + 0.09) = 0.42$

Interprétation :
• Gini = 0 → nœud pur (toutes les observations de la même classe)
• Gini = 0.5 → impureté maximale (50/50 pour 2 classes)
→ **Plus le Gini est bas, plus le nœud est homogène (bon) !**

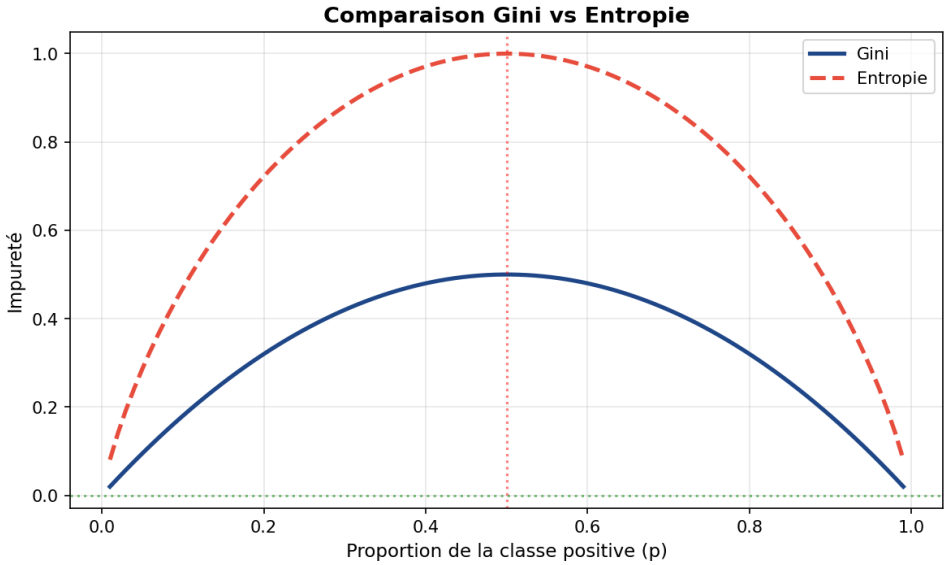
1.2 Entropie (critère de Shannon)

Formule : $Entropy(t) = - \Sigma p_i \log_2(p_i)$

D'où ça vient ?
L'entropie vient de la **théorie de l'information** (Shannon, 1948). Elle mesure le **désordre** ou l'**incertitude** d'une distribution. Plus l'entropie est élevée, plus il y a d'incertitude.

Exemple (même nœud que précédemment) :
100 clients : 70 sans défaut, 30 avec défaut
• Entropy = $-(0.7 \times \log_2(0.7) + 0.3 \times \log_2(0.3))$
• Entropy = $-(0.7 \times (-0.515) + 0.3 \times (-1.737))$
• Entropy = $-(-0.360 - 0.521)$ = **0.881**

Note : Par convention, $0 \log_2(0) = 0$ (limite mathématique)



Observations :

• Les deux courbes sont similaires (forme en cloche)
• Maximum à p=0.5 (classes équilibrées)
• Minimum à p=0 ou p=1 (nœud pur)
• Gini est légèrement plus "plat" au sommet
→ **En pratique : choisissez Gini pour la rapidité !**

1.3 Gain d'Information

Formule :
Gain = Impureté(parent) - Σ (n_{enfant} / n_{parent}) × Impureté(enfant)

L'algorithme teste **toutes les divisions possibles** et choisit celle qui maximise le gain.

Exemple de calcul complet :

Étape	Calcul	Résultat
Nœud parent	100 clients (70 OK, 30 défaut)	Gini = 0.42
Division testée	ratio_endettement ≤ 0.5	
Nœud gauche	60 clients (50 OK, 10 défaut)	Gini = 0.278
Nœud droite	40 clients (20 OK, 20 défaut)	Gini = 0.5
Gain	0.42 - (0.6×0.278 + 0.4×0.5)	0.053

Important : L'algorithme teste TOUTES les variables et TOUS les seuils possibles, puis garde la division qui a le gain maximal !

2. Overfitting et Hyperparamètres

2.1 Pre-Pruning (Élagage Précoce)

Principe : Arrêter la croissance de l'arbre **avant** qu'il ne devienne trop complexe.

Paramètre	Exemple	Impact Interprétabilité
max_depth=3	Max 8 feuilles (2^3)	■■■ Excellent
min_samples_leaf=50	50+ clients par règle	■■■ Robuste
max_leaf_nodes=15	Max 15 règles finales	■■■ Documentable

2.2 Post-Pruning (Élagage A Posteriori)

Principe : Construire un arbre complet, puis **élaguer** les branches peu utiles.

Paramètre ccp_alpha :
$R_\alpha(T) = R(T) + \alpha\,|T|$
où R(T) = erreur, |T| = nombre de feuilles, α = pénalité

Alpha	Nb feuilles	Interprétabilité	Usage
0.0	100+	■ Impossible	Jamais
0.001	30-50	■■ Difficile	Exploration
0.01	10-20	■ Bonne	Production
0.05	5-8	■ Excellente	Présentation

Exemple visuel d'impact :

Avec alpha=0 (70 feuilles) :
Règle 47: SI endettement > 45.32% ET revenu ≤ 23847€ ET ancienneté > 3.2 ans ET nombre_credits = 2 ET age > 34 ET ... (10+ conditions)
→ ■ **Inintelligible**

Avec alpha=0.01 (12 feuilles) :
Règle 1: SI endettement > 60% ALORS défaut probable
Règle 2: SI endettement ≤ 60% ET revenu < 25k ALORS risque moyen
Règle 3: SI endettement ≤ 60% ET revenu ≥ 25k ALORS faible risque
→ ■ **Clair et actionnable**

3. Frontières de Décision Orthogonales

Les arbres de décision créent des **divisions parallèles aux axes** uniquement. Ils ne peuvent pas créer de frontières diagonales ou courbes directement.

Illustration :
Si la vraie frontière optimale est une diagonale (ex: "défaut SI revenu < 1000 × ratio_endettement"), l'arbre devra faire **beaucoup de divisions en escalier** pour l'approximer.

Conséquence :
• Arbre plus profond nécessaire
• Risque accru de surapprentissage
• Plus difficile à interpréter

Solution :
Random Forests et Gradient Boosting combinent plusieurs arbres pour compenser cette limitation.