Mastère Professionnel : Business Intelligence

Apprentissage automatique

Arbres de décision

Apprentissage supervisé

Hind Elouedi

Semestre 2 - 2021

Objectifs

- Définir les concepts de base relatifs aux arbres de décision.
- Comprendre l'algorithme de construction d'un arbre de décision.
- Utiliser les arbres de décision dans la classification.
- Comprendre la notion d'élagage.
- Définir les notions de Bagging et Boosting.

Points abordés

- Introduction
- Composants
- Construction d’un arbre
- Classification
- Elagage
- Attributs à valeurs continues
- Bagging et boosting
- Conclusion

Introduction

Un peu d'histoire…

- Statistiques:
  - Morgan et Sonquist (1963): Arbres de régression dans un processus de prédiction et d'explication (AID – Automatic Interaction Detection).
  - Morgan et Messenger (1973): THAID (Theta AID), utilisation des arbres pour les classements et les discriminations.
  - Kass (1980): CHAID (CHi-squared Automatic Interaction Detector), prédiction ou détection d'interactions entre variables.
  - Breiman et al. (1984): CART (Classification And Regression Tree).

Introduction (2)

Un peu d'histoire…

- Apprentissage automatique:
  - Quinlan (1979): ID3 (Induction of Decision Tree), les travaux de Quinlan sont rattachés à ceux de Hunt (1962).
  - Quinlan (1993): Ensemble de travaux couronnés par beaucoup d'articles et la mise en place de la méthode C4.5, la référence incontournable sur les arbres de décision.

Introduction (3)

- Arbre de décision:
  - Technique de classification en apprentissage supervisé.
  - Technique utilisée en intelligence artificielle.
- Avantages:
  - Traitement des problèmes complexes.
  - Expression simple de la connaissance.
  - Facilité dans la compréhension et l'interprétation des résultats.
  - Participation des experts dans l'élaboration des règles.

Introduction (4)

- Domaines d’application:
  - Gestion de crédits.
  - Diagnostic médical.
  - Analyse du marché.
  - Détection d’intrusion.
  - Contrôle de production.

Composants

Représente tout l'ensemble d'apprentissage

Racine

- Test sur les attributs

Représente des sous-ensembles d'apprentissage

Nœuds de décision

- Tests sur les attributs

Nœuds feuilles

- Classes

Branches

- Valeurs de l'attribut

Composants (2)

- Ensemble d'apprentissage

Attributs

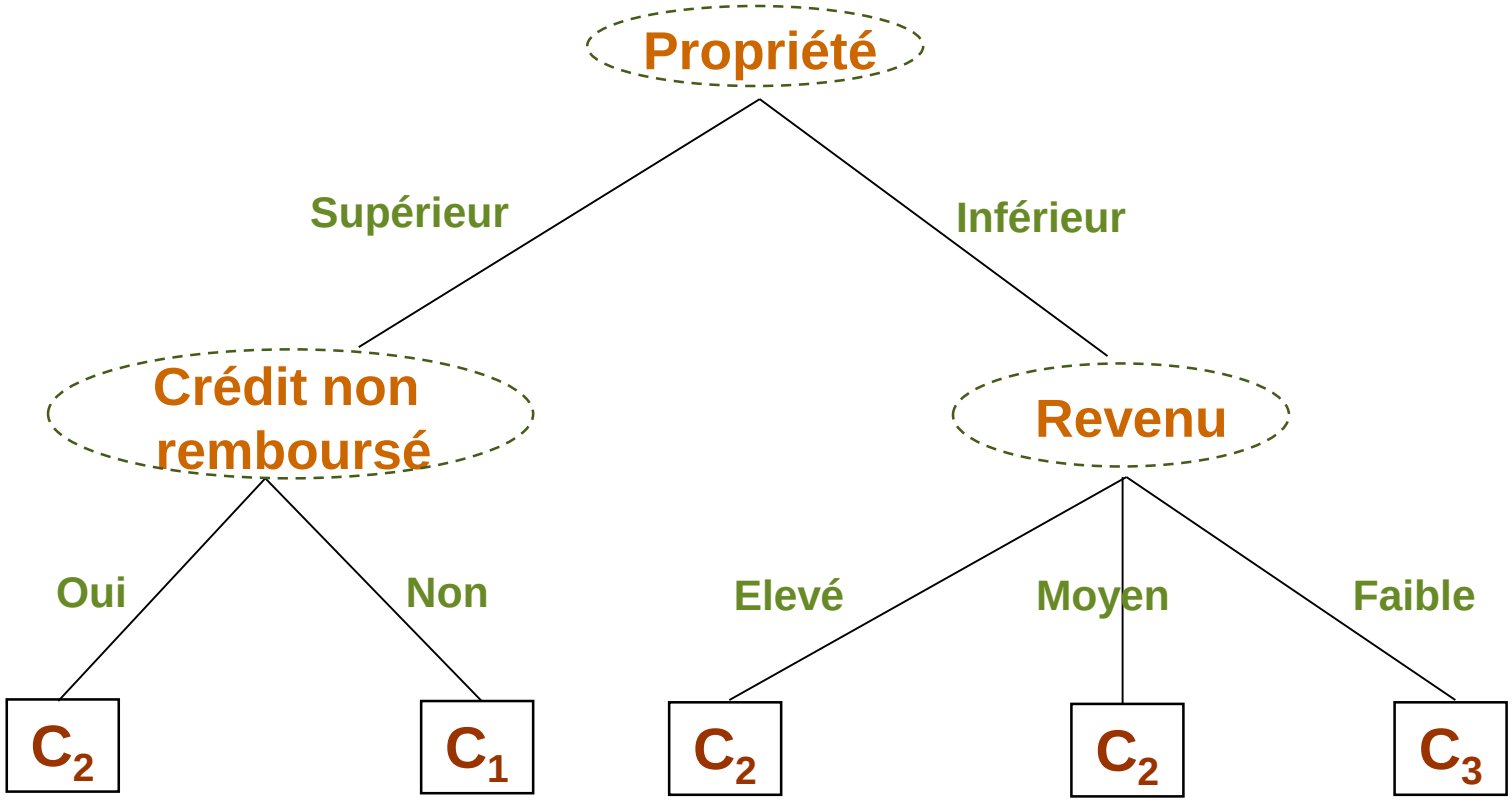
Valeurs des attributs

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

C_1: Attribuer tout le crédit - **C_2**: Attribuer une partie crédit - **C_3**: Ne pas attribuer le crédit.

Composants (3)

Arbre de décision



Construction

- Problématique
- Procédure de construction
- Paramètres
- Application

Construction (2)

- Problématique
  - Construire un arbre de décision à partir d'un ensemble d'apprentissage.

Généralisation: Être capable de classer correctement un nouvel objet.

Algorithme complexe et horrible !!!

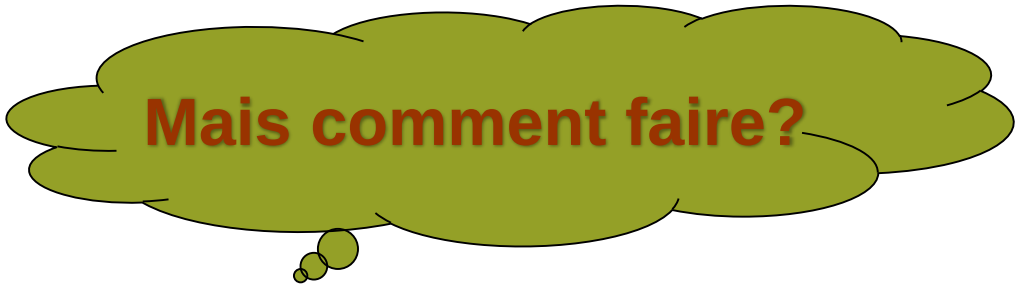
- Générer tous les arbres de décision possibles
- Tester la justesse de chaque arbre généré
- Choisir le meilleur arbre de décision

Construction (3)

Processus très coûteux

Proposer un meilleur algorithme

- Choisir le **meilleur** attribut.
- **Partitionner** l'ensemble d'apprentissage.
- **Répéter** jusqu'à ce que chaque élément de l'ensemble d'apprentissage soit correctement classé.



Construction (4)

- Algorithmes: Top Down Induction of Decision Trees (TDIDT)

  ⇒ Diviser pour régner (Induction descendante)

  - ID3 (Quinlan, 1979)
  - CART (Breiman et al., 1984)
  - ASSISTANT (Bratko, 1984)
  - C4.5 (Quinlan, 1993)

  .
  .
  .

Construction (5)

Procédure de construction

- Processus récursif:
  - L'arbre commence à un nœud qu'on appelle **racine** et qui représente toutes les données.
    - Si les objets sont de la même classe, alors le nœud devient une **feuille** libellée par le nom de la classe.
    - Sinon, **sélectionner** les nœuds qui séparent **le mieux** les objets en classes **homogènes**.
    - Le traitement récursif s'arrête quand au moins l'un des critères d'arrêt est vérifié.

Construction (6)

Procédure de construction (2)

- Recherche à chaque niveau de l'attribut le plus **discriminant**.
- Partition des données (T):
  - Si tous les éléments de T sont dans la même classe alors retour;
  - Pour chaque attribut A, évaluer **la qualité du partitionnement** sur A;
  - Utiliser **le meilleur partitionnement** pour **diviser** T en T1, T2, …Tk;
  - Pour i = 1 à k faire **Partition(T_i)**;

Construction (7)

Paramètres

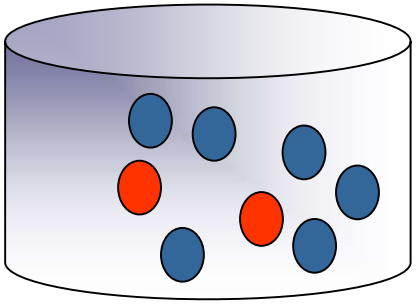
- Mesure de sélection des attributs.
- Stratégie de partitionnement.
- Critères d’arrêt.

Paramètres (2)

- Choix de l'attribut: Plusieurs mesures:
  - Gain d'information.
  - Indice de Gini.
  - Ratio de gain.
- Mesure de l'information:
  - L'**entropie** de **Shannon** exprime la quantité d'information: Le nombre de bits nécessaires pour coder l'information.

Paramètres (3)

- **Exemple**



La probabilité de tirer une boule bleue est: $\frac{6}{6+2} = \frac{3}{4}$

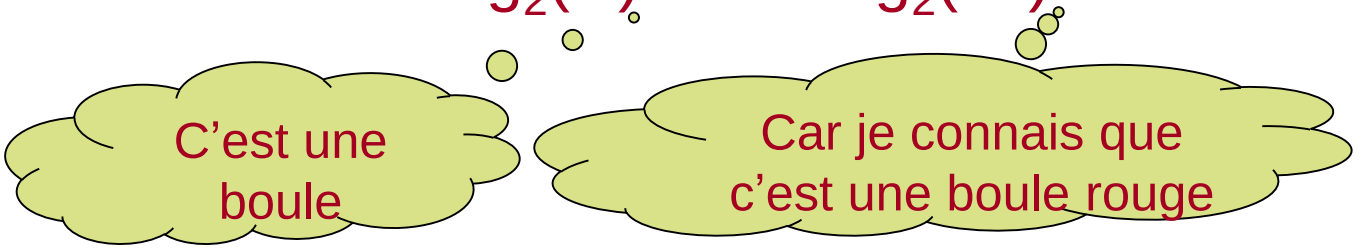
La probabilité de tirer une boule rouge est: $\frac{2}{6+2} = \frac{1}{4}$

- **Apport d'information**
  - Nombre de bits nécessaires pour distinguer chaque boule parmi N:
    - P bits permettent de coder 2^P informations.
    - $\log_2(N)$ bits permettent de coder N informations.

Paramètres (4)

- Si je tire une boule (parmi N boules) et que je ne connais que sa couleur (par exemple elle est rouge), l'information acquise sera:

$\log_2(N)$ bits - $\log_2(Nr)$ bits



- Si je tire une boule au hasard et qu'on me donne sa couleur, l'information acquise sera:

$$\text{Prob(Bleue)}\,(\log_2(N) - \log_2(Nb)) + \text{Prob(Rouge)}(\log_2(N) - \log_2(Nr))$$

$$\frac{3}{4}(\log_2 8 - \log_2 6) \quad + \quad \frac{1}{4}(\log_2 8 - \log_2 2)$$

Paramètres (5)

$$\text{Prob(Bleue)}\,(\log_2(N) - \log_2(Nb)) + \text{Prob(Rouge)}(\log_2(N) - \log_2(Nr))$$

$$\frac{Nb}{N}\left(\log_2 \frac{N}{Nb}\right) + \frac{Nr}{N}\left(\log_2 \frac{N}{Nr}\right) \Leftrightarrow -\frac{Nb}{N}\left(\log_2 \frac{Nb}{N}\right) - \frac{Nr}{N}\left(\log_2 \frac{Nr}{N}\right)$$

$$\Leftrightarrow -\text{Prob(Bleue)}\log_2(\text{Prob(Bleue)}) - \text{Prob(Rouge)}\log_2(\text{Prob(Rouge)})$$

$$-\frac{3}{4}\left(\log_2 \frac{3}{4}\right) - \frac{1}{4}\left(\log_2 \frac{1}{4}\right)$$

C’est la quantité d’information apportée par la couleur.

Paramètres (6)

- **Mesure de l'information**
  - Si on a n classes (C_1, C_2,.., C_n) de probabilités respectives p_1, p_2,.., p_n, la quantité d'information relative à la connaissance de la classe est définie par **l'entropie d'information**:

$$I = \sum_{i=1..n} -p_i \log_2 p_i$$

  - I = 0 quand $\exists$ i/ $p_i = 1$ (une seule classe).
  - I est maximale quand $\forall$ i/ $p_i = 1/n$ (classes équiprobables).

Paramètres (7)

- **Gain d'information (ID3)**
  - freq(T, C_j): Nombre d'objets de T appartenant à la classe C_j.
  - L'information relative à T est définie:

Quantité moyenne d'information nécessaire pour identifier la classe d'un objet de T

$$Info(T) = -\sum_{j=1}^{n} \frac{freq(T, C_j)}{|T|} \log_2 \frac{freq(T, C_j)}{|T|}$$

  - Une mesure similaire de T après partition selon l'attribut A (contenant n valeurs) est:

$$Info_A(T) = \sum_{i \in D_A} \frac{|T_i|}{|T|} Info(T_i)$$

D_A =Domaine de valeurs de l'attribut A.

  - Le gain d'information mesure le gain obtenu suite au partitionnement selon l'attribut A: $Gain(T, A) = Info(T) - Info_A(T)$

⇒ On sélectionne l'attribut offrant le plus de gain.

Paramètres (8)

- **Attributs multivalués**
  - Le Critère de gain d'information présente une limite: il favorise les attributs ayant plusieurs valeurs.
  - Lorsqu'un attribut a plusieurs valeurs possibles, son gain peut être très élevé, car il classifie parfaitement les objets.
  - Par contre, cela peut générer un arbre de décision d'une profondeur de 1 (ou faible) qui ne sera pas très bon pour les instances futures.

Paramètres (9)

- **Ratio de gain (C4.5)**
  - Une mesure de l'information contenue dans l'attribut A (mesure de dispersion) est définie:

$$\text{Split Info}(T, A) = -\sum_{i \in D_A} \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}$$

  - Le ratio de gain mesure le gain calibré par Split Info.

$$\text{Gain Ratio}(T, A) = \frac{\text{Gain}(T, A)}{\text{Split Info}(T, A)}$$

Quantité d'information générée par T et utile pour la classification

On sélectionne l'attribut offrant le ratio de gain le plus élevé.

Paramètres (10)

- **Stratégie de partitionnement**
  - Pour chaque valeur de l'attribut, on va associer une branche dans l'arbre.
  - Problème avec les attributs continus.

Découper en sous-ensembles ordonnés

Paramètres (11)

- **Critères d'arrêt**
  - Si tous les objets appartiennent à la même classe.
  - S'il n'y a plus d'attributs à tester.
  - S'il n'y a pas d'objets avec la valeur d'attribut.
  - Absence d'apport informationnel des attributs. (tous les ratios de gain ≤ **0**)

Construction (8): Application

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$$Info(T) = - \sum_{j=1}^{3} \frac{freq(T, C_j)}{|T|} \log_2 \frac{freq(T, C_j)}{|T|}$$

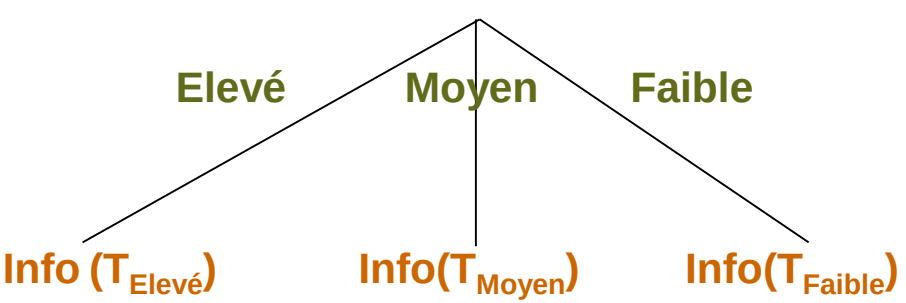
$Info(T) = - 3/10 \log_2 3/10 - 5/10 \log_2 5/10 - 2/10 \log_2 2/10 = 1.485$

Application (2): $Info_{Revenu}(T)$

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$$Info_{Revenu}(T) = \sum_{i \in D_{Revenu}} \frac{|T_i|}{|T|} Info(T_i)$$

D_{Revenu} ={Elevé, Moyen, Faible}



$Info(T_{Elevé}) = - 2/4 \log_2 2/4 - 2/4 \log_2 2/4 = 1$

$Info(T_{Moyen}) = - 1/4 \log_2 1/4 - 3/4 \log_2 3/4 = 0.812$

$Info(T_{Faible}) = - 2/2 \log_2 2/2 = 0$

$Info_{Revenu}(T) = 4/10\ Info(T_{Elevé}) + 4/10\ Info(T_{Moyen}) + 2/10\ Info(T_{Faible}) = 0.725$

Application (3): Gain ration (T, Revenu)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$$Gain(T, Revenu) = Info(T) - Info_{Revenu}(T) = 0.761$$

$$Split\ Info(T, Revenu) = -\sum_{i \in D_{Revenu}} \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}$$

$$Split\ Info(T, Revenu) = -\ 4/10 \log_2 4/10 - 4/10 \log_2 4/10 - 2/10 \log_2 2/10 = 1.522$$

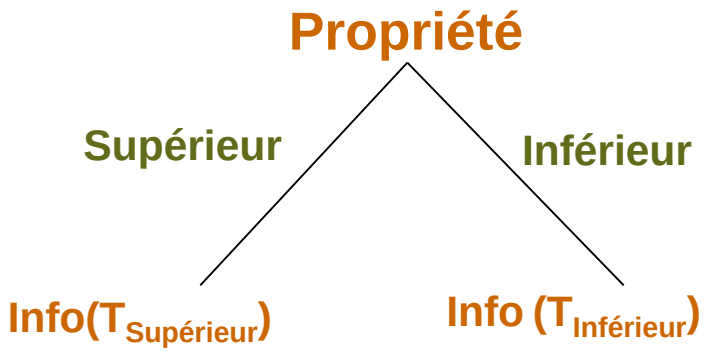
$$\textbf{Gain Ratio(T, Revenu)} = \frac{\textbf{0.761}}{\textbf{1.522}} = \textbf{0.5}$$

Application (4): $Info_{Propriété}(T)$

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$$Info_{Propriété}(T) = \sum_{i \in D_{Propriété}} \frac{|T_i|}{|T|} Info(T_i)$$

$D_{Propriété}$ ={Supérieur, Inférieur}



$$Info(T_{Supérieur}) = - 3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971$$

$$Info(T_{Inférieur}) = - 3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971$$

$Info_{Prorpiété}(T) = 5/10\ Info(T_{Supérieur}) + 5/10\ Info(T_{Inférieur}) = 0.971$

Application (5): Gain ration (T, Propriété)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

Gain(T, Propriété) = Info(T) – $\text{Info}_{\text{Propriété}}$(T) = 0.514

$$\text{Split Info(T, Propriété)} = -\sum_{i \in D_{\text{Propriété}}} \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}$$

Split Info(T, Propriété) = - 5/10 $\log_2$ 5/10 - 5/10 $\log_2$ 5/10 = 1

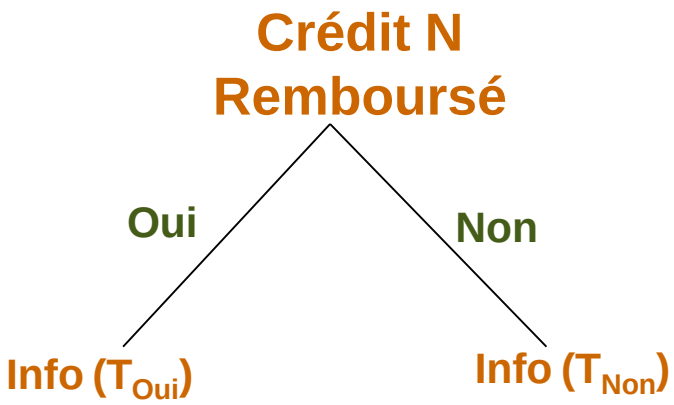
Gain Ratio(T, Propriété) = $\frac{0.514}{1}$ = 0.514

Application (6): $Info_{Crédit\ N\ remboursé}(T)$

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$$Info_{Crédit\ N\ remboursé}(T) = \sum_{i \in D_{Crédit\ N\ remboursé}} \frac{|T_i|}{|T|} Info(T_i)$$

$D_{Crédit\ non\ remboursté} = \{Oui, Non\}$



$Info(T_{Oui}) = - 4/5 \log_2 4/5 - 1/5 \log_2 1/5 = 0.722$

$Info(T_{Non}) = - 3/5 \log_2 3/5 - 1/5 \log_2 1/5 - 1/5 \log_2 1/5 = 1.371$

$Info_{Crédit\ N\ remboursé}(T) = 5/10\ Info(T_{Oui}) + 5/10\ Info(T_{Non}) = 1.046$

Application (6): Gain ration (T, Crédit N remboursé)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

Gain(T, Crédit N remboursé) =
$\text{Info}(T) - \text{Info}_{\text{Crédit non remboursé}}(T) = 0.439$

$$\text{Split Info}(T, \text{Crédit non remboursé}) = -\sum_{i \in D_{\substack{\text{Crédit N} \\ \text{remboursé}}}} \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}$$

$$\text{Split Info}(T, \text{Crédit non remboursé}) = -\ 5/10 \log_2 5/10 - 5/10 \log_2 5/10 = 1$$

$$\textbf{Gain Ratio}(T, \text{Crédit non remboursé}) = \frac{0.439}{1} = 0.439$$

Application (7): Arbre de décision (Niveau 1)

Gain Ratio(T, Revenu) = 0.5

Gain Ratio(T, Propriété) = 0.514

Gain Ratio(T, Crédit non remboursé) = 0.439

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Elevé	Inférieur	Oui	C_2
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

Racine ⟵ **Propriété**

Supérieur

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2

Inférieur

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

Application (8): Propriété = Supérieur (1)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2

$$Info(T_{Supérieur}) = Info(S) = - 3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971$$

Application (9): Propriété = Supérieur (2)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2

$Info_{Revenu}(S_{Elevé}) = -\ 2/3 \log_2 2/3 - 1/3 \log_2 1/3 = 0.918$

$Info_{Revenu}(S_{Moyen}) = -\ 1/2 \log_2 1/2 - 1/2 \log_2 1/2 = 1$

$Info_{Revenu}(S_{Faible}) = 0$

$Info_{Rvenu}(S) = ((3/5) * 0.918) + ((2/5) * 1) + (0*0) = 0.951$

$Gain(S, Revenu) = 0.02$

$Split\ Info(S, Revenu) = -\ 3/5 \log_2 3/5 - 2/5 \log_2 2/5 - 0 = 0.971$

$Gain\ Ratio(S, Revenu) = 0.02$

Application (10): Propriété = Supérieur (3)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2

$Info_{Crédit\ non\ remboursé}(S_{Oui}) = -\ 2/2\ \log_2\ 2/2 = 0$

$Info_{Crédit\ non\ remboursé}(S_{Non}) = -\ 3/3\ \log_2\ 3/3 = 0$

$Info_{Crédit\ non\ remboursé}(S) = ((3/5) * 0) + ((2/5) * 0) = 0$

Gain(S, Crédit non remboursé) = 0.971

Split Info(S, Crédit non remboursé) = $-\ 2/5\ \log_2\ 2/5 - 3/5\ \log_2\ 3/5 = 0.971$

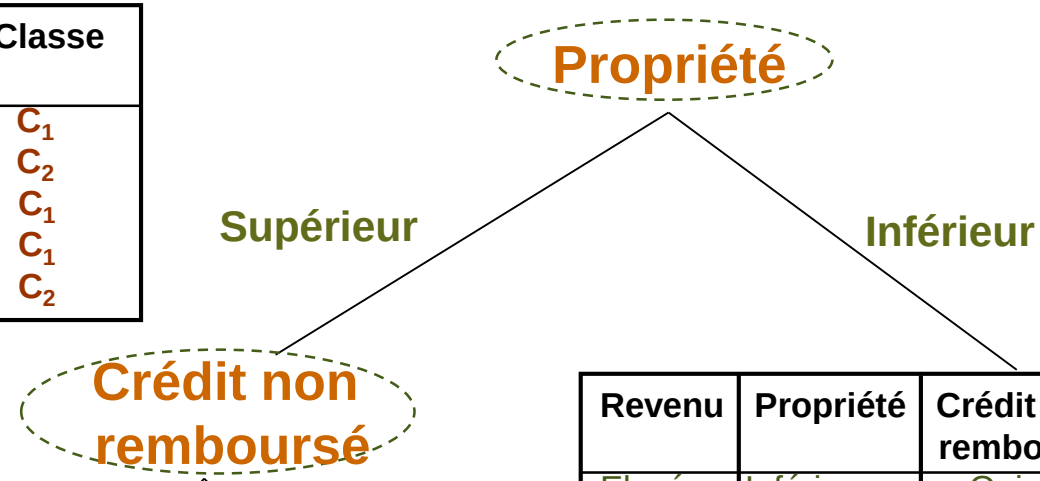
Gain Ratio(S, Crédit non remboursé) = 1

Application (11): Arbre de décision (Niveau 2)

Gain Ratio(S, Revenu) = 0.02

Gain Ratio(S, Crédit non remboursé) = 1

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Oui	C_2
Elevé	Supérieur	Non	C_1
Moyen	Supérieur	Non	C_1
Moyen	Supérieur	Oui	C_2

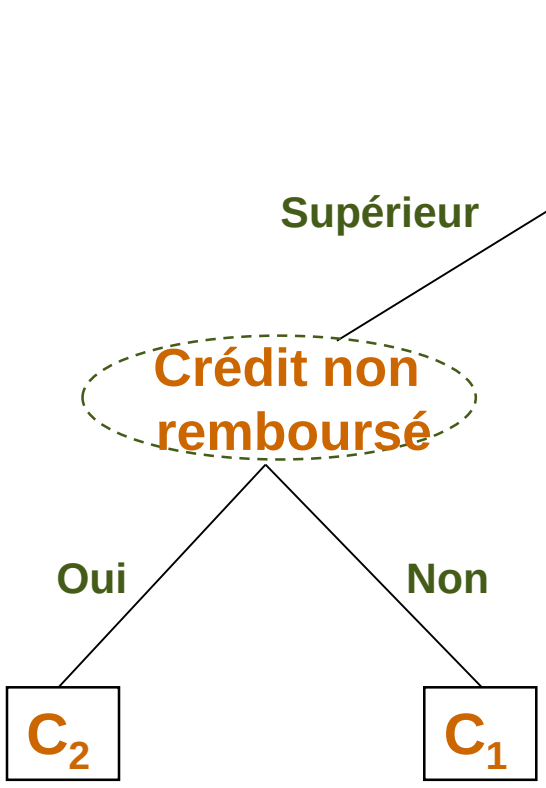


Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Oui	C_2
Moyen	Supérieur	Oui	C_2

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Supérieur	Non	C_1
Elevé	Supérieur	Non	C_1
Moyen	Supérieur	Non	C_1

Application (12): Arbre de décision (Niveau 2)



Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

Application (13): Propriété = Inférieur (1)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$$Info(T_{Inférieur}) = Info(I) = - 3/5 \log_2 3/5 - 2/5 \log_2 2/5 = 0.971$$

Application (14): Propriété = Inférieur (2)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$Info_{Revenu}(I_{Elevé}) = -\ 1/1 \log_2 1/1 = 0$

$Info_{Revenu}(I_{Moyen}) = -\ 2/2 \log_2 2/2 = 0$

$Info_{Revenu}(I_{Faible}) = -\ 2/2 \log_2 2/2 = 0$

$Info_{Revenu}(I) = ((1/5) * 0) + ((2/5) * 0) + ((2/5) * 0) = 0$

$Gain(I, Revenu) = 0.971$

$Split\ Info(I, Revenu) = -\ 1/5 \log_2 1/5 - 2/5 \log_2 2/5 - 2/5 \log_2 2/5 = 1.522$

$Gain\ Ratio(I, Revenu) = 0.638$

Application (15): Propriété = Inférieur (3)

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

$Info_{Crédit\ non\ remboursé}(I_{Oui}) = -\ 2/3\ \log_2\ 2/3 - 1/3\ \log_2\ 1/3 = 0.918$

$Info_{Crédit\ non\ remboursé}(I_{Non}) = -\ 1/2\ \log_2\ 1/2 - 1/2\ \log_2\ 1/2 = 1$

$Info_{Crédit\ non\ remboursé}(I) = ((3/5) * 0.918) + ((2/5) * 1) = 0.951$

Gain(I, Crédit non remboursé) = 0.02

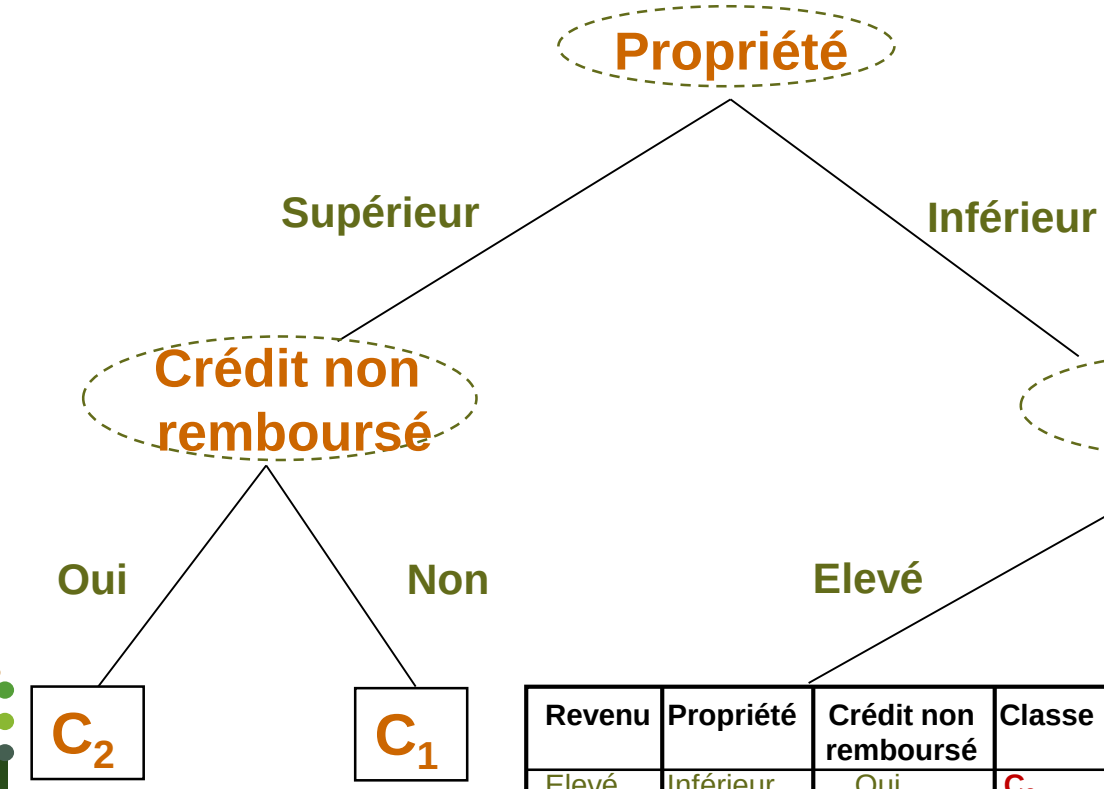
Split Info(I, Crédit non remboursé) = $-\ 3/5\ \log_2\ 3/5 - 2/5\ \log_2\ 2/5 = 0.971$

Gain Ratio(I, Crédit non remboursé) = 0.02

Application (16): Arbre de décision (Niveau 2)

Gain Ratio(S, Revenu) = 0.638
Gain Ratio(S, Crédit non remboursé) = 0.02

Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

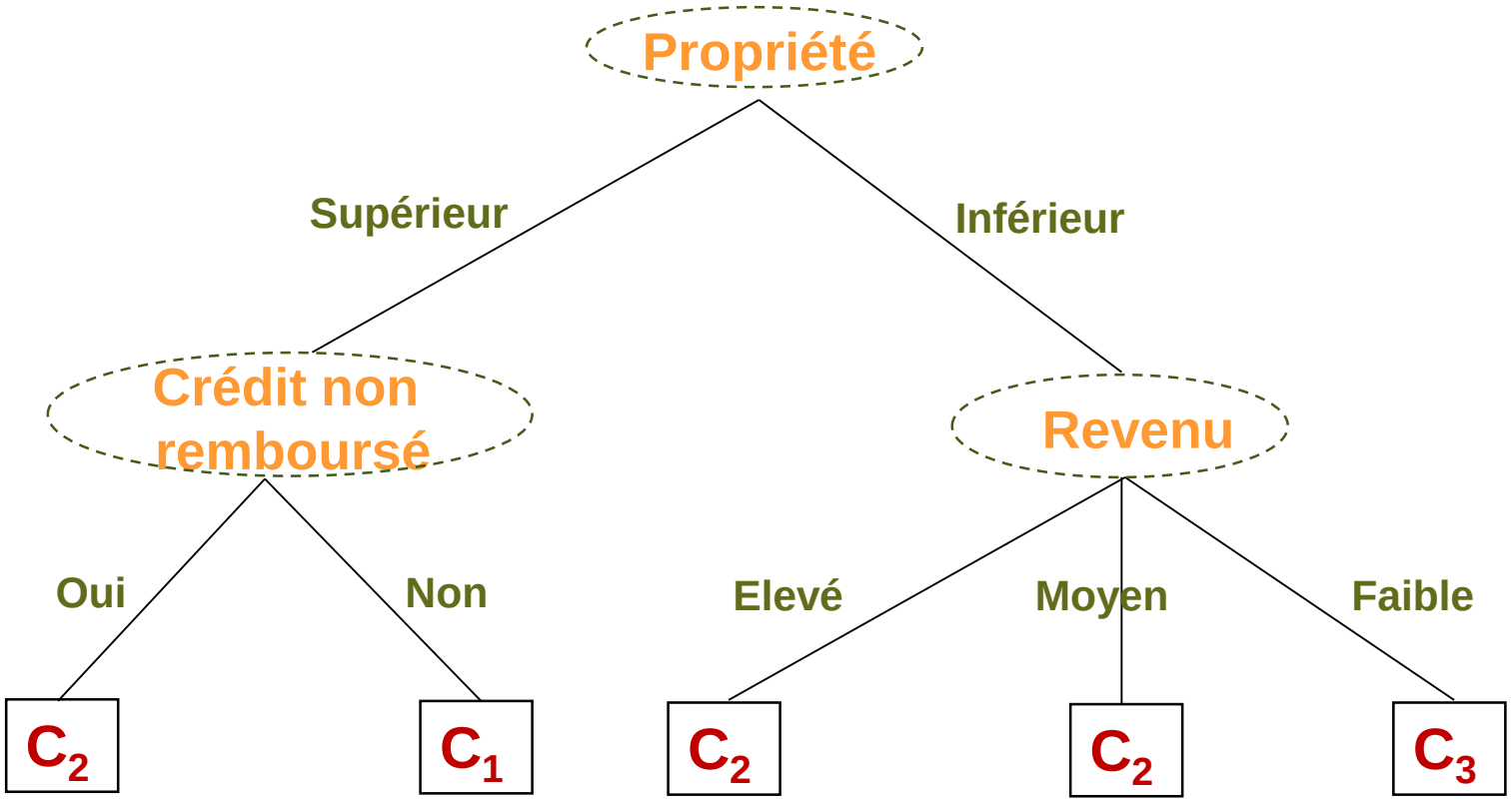


Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2

Revenu	Propriété	Crédit non remboursé	Classe
Faible	Inférieur	non	C_3
Faible	Inférieur	Oui	C_3

Revenu	Propriété	Crédit non remboursé	Classe
Moyen	Inférieur	non	C_2
Moyen	Inférieur	Oui	C_2

Application (17): Arbre de décision final



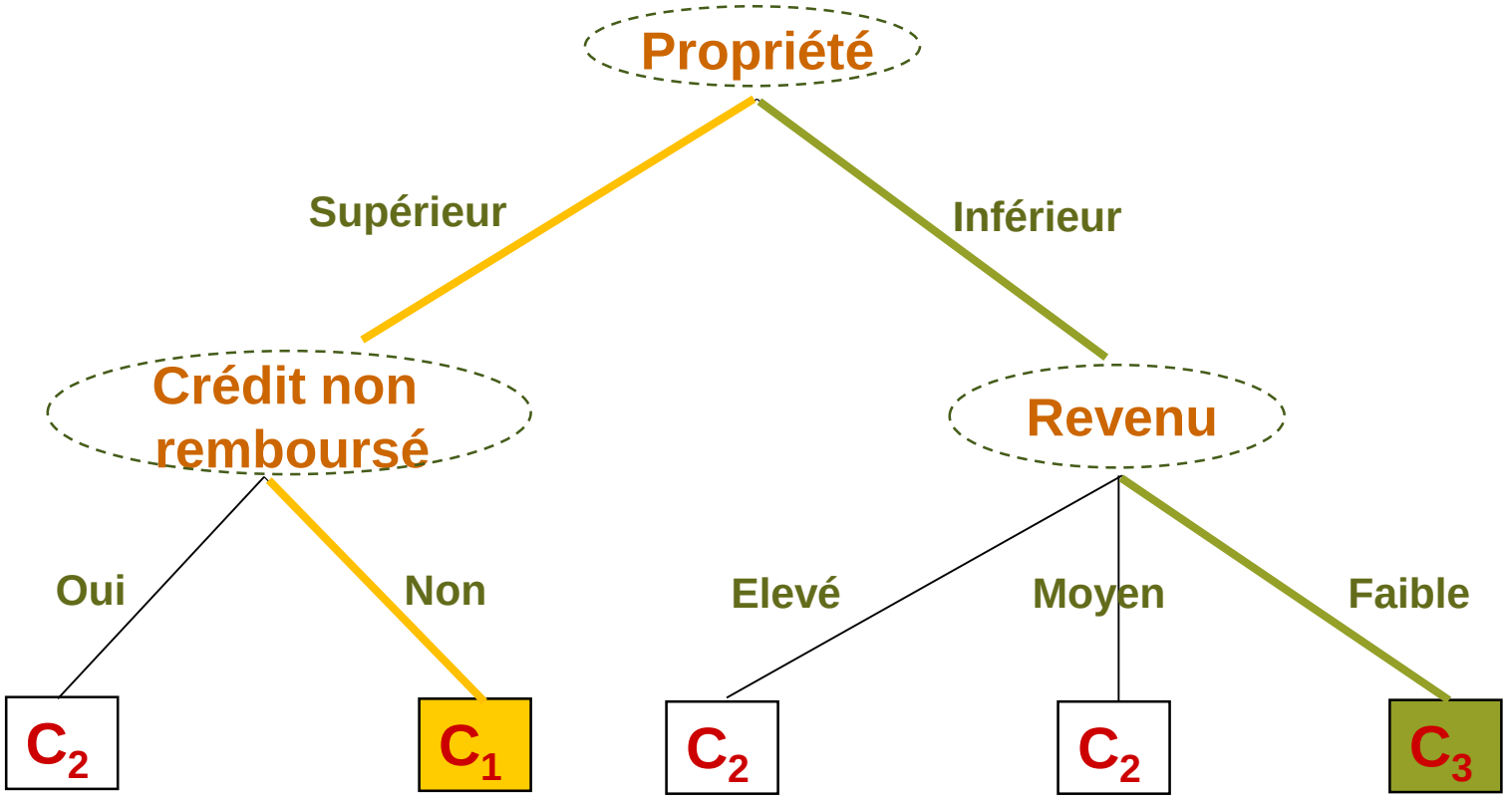
Classification

- Classification basée sur une séquence de questions portant sur un attribut.
- La question est représentée par un nœud.
- On prend la branche qui correspond à la réponse jusqu'à la question suivante.
- La feuille désigne la classe correspondant à l'objet à classer.

⇒ Organiser les questions/réponses sous la forme d'un arbre

Trouver le chemin relatif à l'objet à classer menant de la racine à l'une des feuilles de l'arbre

Classification (2)



À classer ?

Revenu	Propriété	Crédit non remboursé	Classe
Moyen	Supérieur	Non	?
Faible	Inférieur	Oui	?

Classification (3)

Convertir l’arbre en règles

- Représenter la connaissance sous la forme de Si….alors.
- Une règle est créée pour chaque chemin de la racine jusqu’à la feuille.
- Les feuilles contiennent la classe à prédire.
- Les règles sont plus faciles à comprendre et à interpréter.

Classification (4)

Convertir l'arbre en règles (2)

Si (Propriété = Supérieur) **ET** (Crédit non remboursé = Oui)
alors C_2
Si (Propriété = Supérieur) **ET** (Crédit non remboursé = Non)
alors C_1
Si (Propriété = Inférieur) **ET** (Revenu = Elevé)
alors C_2
Si (Propriété = Inférieur) **ET** (Revenu = Moyen)
alors C_2
Si (Propriété = Inférieur) **ET** (Revenu = Faible)
alors C_3

Élagage

Pourquoi élaguer?

- Problème de sur-apprentissage (Overfitting)
  - Améliorer un modèle en le rendant meilleur sur l'ensemble d'apprentissage mais il sera de plus en plus compliqué.
    - Plusieurs branches.
    - Arbre illisible.
    - Faible résultat de classification.

Il faut élaguer !!

    - Réduire la taille de l'arbre.
    - Améliorer la performance.

Rendre l'arbre plus compréhensible.

Mesurer la performance sur un ensemble **différent** de l'ensemble d'apprentissage.

Élagage (2)

- Objectif: Minimiser la longueur de l'arbre
  - Cette méthode coupe des parties de l'arbre en choisissant un noeud et en enlevant tout son sous-arbre.

➡ Ce noeud devient une feuille et on lui attribue la valeur de classification qui revient le plus souvent.

  - Des noeuds sont enlevés seulement si l'arbre résultant n'est pas pire que l'arbre initial sur les exemples de validation.
  - On continue tant que l'arbre résultant offre de meilleurs résultats sur les exemples de validation.

➡ Réduire l'arbre en enlevant des branches qui auraient été ajoutées par une erreur dans les objets d'apprentissage.

Élagage (3)

- Pré-élagage (pre-pruning)
  - Arrêter le développement d'un nœud.
  - Ne pas partitionner si le résultat va s'affaiblir.

  ⇒ Créer une feuille si la classe est majoritairement représentée (s'arrêter avant d'engendrer un noeud inutile).
- Post-élagage (post-pruning)
  - Élaguer après la construction de l'arbre en entier, en remplaçant les sous-arbres satisfaisant le critère d'élgage par un noeud (générer l'arbre entier puis élaguer).
  - Élaguer après la construction de l'arbre en entier, en remplaçant les sous-arbres satisfaisant le critère d'élgage par un noeud:
    - Une feuille.
    - Un de ses fils (le plus fréquent).

Élagage (4)

- Méthodes d'élagage
  - **MCCP:** Minimal Cost Complexity Pruning (Breiman, 1984).
  - **MEP:** Minimum Error Pruning (Niblett et Bratko, 1986).
  - **CVP**: Critical Value Pruning (Mingers, 1987).
  - **PEP**: Pessimistic Error Pruning (Quinlan, 1987).
  - **REP**: Reduced Error Pruning (Quinlan, 1987, 1993).
  - **EBP**: Error Based Pruning (Quinlan, 1993).

Élagage (5)

Mesure de qualité de l’arbre

- PCC: Pourcentage de Classifcation Correcte.
- Complexité:
  - Taille de l’arbre.
  - Nombre de feuilles.
- Temps.

Attributs à valeurs continues

- Problème:
  - Seuils au lieu d'une infinité de valeurs.
  - Certains attributs sont continus.

⇒ Découper en sous-ensembles ordonnés

  - Division en segments [a0,a1[, [a1,a2[, …., [an-1,an].
  - Utiliser moyenne, médiane,…
  - Tester plusieurs cas et retenir le meilleur.

Attributs à valeurs continues (2)

- On utilise un point de coupe pour obtenir une discrétisation des variables continues.
  - Ex.: la variable Température est continue et on a les 6 exemples suivants.

Température	0	8	18	26	30	38
JouerTennis	Non	Non	Oui	Oui	Oui	Non

  - On met les valeurs en ordre croissant et on regarde les endroits où la classe change de valeur. À ces endroits, on choisit la médiane comme valeur de coupe.
  - On compare toutes les valeurs de coupe et on choisit celle qui apporte le plus grand gain d'information.

Attributs à valeurs continues (3)

- Exemple:

Objet	Température	Jouer
O_1	15	Oui
O_2	20	Oui
O_3	5	Non
O_4	30	Non
O_5	9	Non
O_6	35	Non

- Appliquer la procédure de traitement des attributs continus sur cet exemple

Attributs à valeurs continues (4)

- Solution : Il faut d'abord ordonner selon la valeur de l'attribut température (ordre croissant):

Objet	Température	Jouer
O_3	5	Non
O_5	9	Non
O_1	15	Oui
O_2	20	Oui
O_4	30	Non
O_6	35	Non

  - Il y a deux coupures binaires possibles avec changement de classe, il faut voir laquelle apporte le meilleur gain?
  - Puisque la valeur de info est toujours la même, on se contente de trouver quelle coupure présente une valeur d'$info_{Température}$ minimale.
  - Les coupures sont 12 et 25 (où il y a changement de classe).

Attributs à valeurs continues (5)

- Info(T, 12) = 2/6 Info(T, <12) + 4/6 Info(T, >12)
- Info(T, <12) = 0
- Info(T, >12)= -2/4log 2/4 -2/4log 2/4 =1
- Info(T, 12) = 0.666

- Info(T, 25) = 4/6 Info(T, <25) + 2/6 Info(T, >25)
- Info(T, <25) = -2/4 log 2/4 -2/4 log 2/4 =1
- Info(T, >25) = 0
- Info(T, 25) = 0.666

- On peut aussi utiliser le ratio de gain donc calculer le split info.
- Donc ici on peut choisir l'une des coupures 12 ou 25

Bagging et Boosting

- Bagging: Bootstrap aggregation
  - Echantillon Bootstrap = Un sous-ensemble d'apprentissage
  - Génération de k échantillons à partir de l'ensemble d'apprentissage.
  - Pour chaque échantillon, construire l'arbre de décision correspondant.
  - La décision finale pour la classe d'un nouvel objet est obtenue par vote majoritaire.

⇒ Le bagging améliore la précision d'un classifieur instable

Bagging et Boosting (2)

- Boosting:
  - C’est une approche collaboratrice contrairement au bagging (compétitive).
  - Les sous classifieurs sont introduits un à la fois et travaillent sur des sous-ensembles différents.
  - Chaque nouveau sous-classifieur s’occupe des objets mal classés.

    ⇒ L’intérêt d’appliquer le boosting est quand les classifieurs présentent de mauvais résultats.
  - Les classifieurs peuvent être de types différents.

Conclusion

- Applicables à des variables quantitatives et qualitatives.
- Intelligibilité de la procédure de décision (traduction sous forme de règles).
- Rapidité de décision.
- Très utilisés en data mining (recherche d'informations dans de grandes bases de données hétérogènes).