Chapitre 3

Arbres de décision

Zied Elouedi
2018/2019

Plan

- Composants
- Construction
  - Procédure de construction
  - Paramètres (Mesure de sélection d'attributs, Stratégie de partitionnement, critères d'arrêt)
- Algorithmes incrémentaux
- Classification
- Élagage
  - Pré-élagage
  - Post-élagage
- Mesures de la qualité de l'arbre
- Attributs à valeurs continues
- Attributs à valeurs manquantes
- Variantes d'arbres de décision
- Bagging et boosting
- Conclusion et perspectives

Introduction

Arbre de décision est une technique de classification en apprentissage supervisé

Utilisation dans le domaine de **l'intelligence artificielle**

☺ 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.

Applications

- Gestion de crédits
- Diagnostic médical
- Analyse du marché
- Contrôle de production
  - .
  - .
  - .

Composants

Composants

Représente tout l'ensemble d'apprentissage

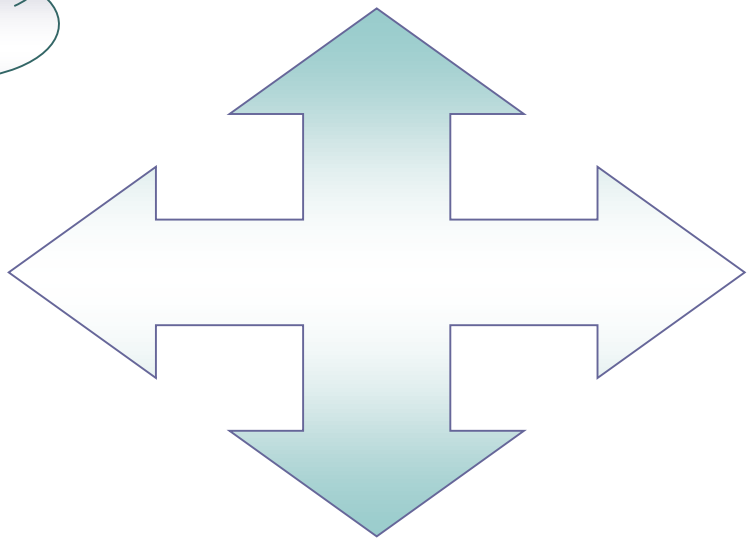
Racine

- Test sur les attributs

Représente des sous-ensembles d'apprentissage

Nœuds de décision

- Tests sur les attributs



Nœuds feuilles (Nœuds terminaux)

- Classes

Branches

- Valeurs de l'attribut

Ensemble d'apprentissage

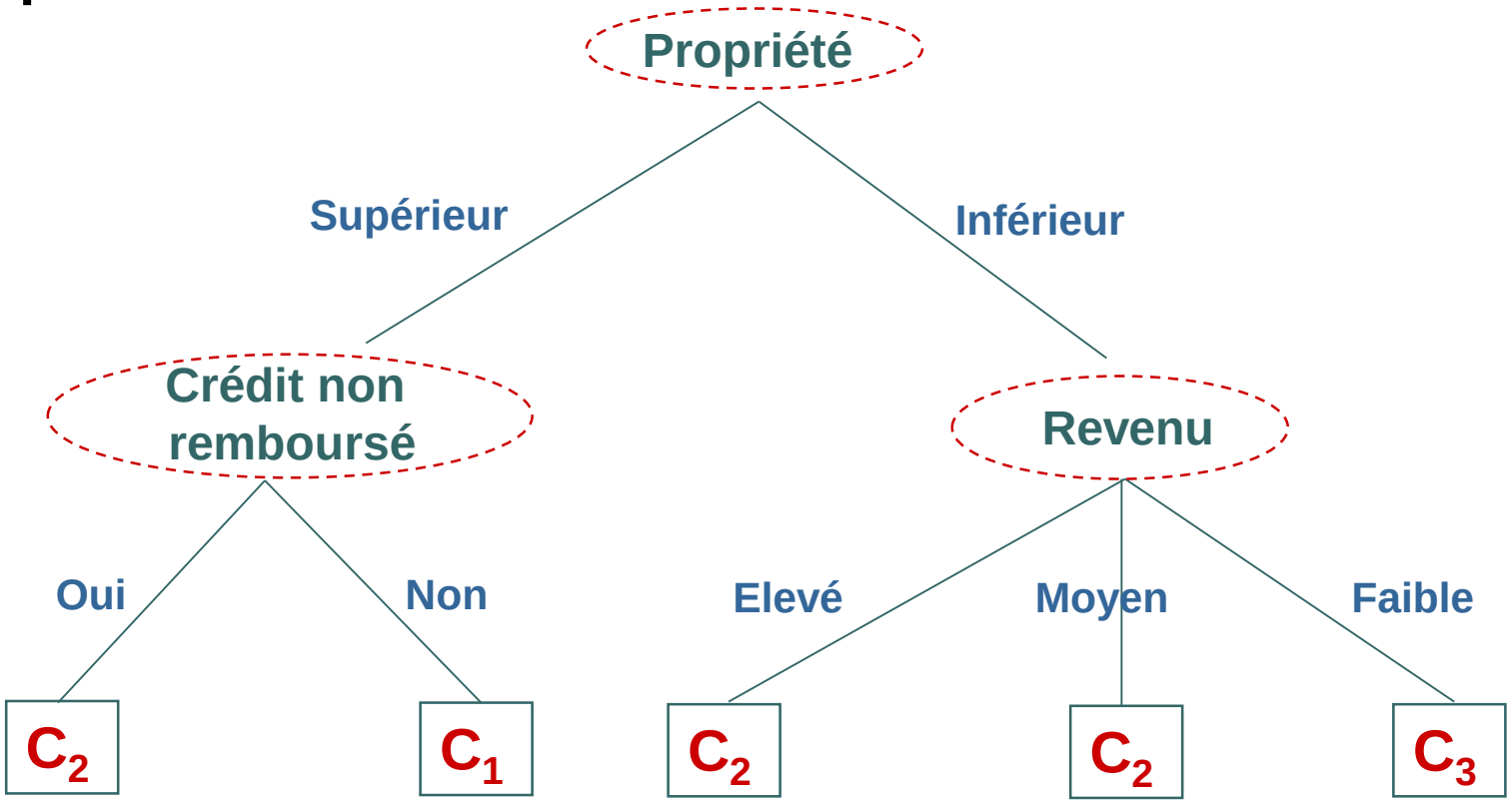
Attributs

Valeurs des attributs

Revenu	Propriété	Crédit non remboursé	Classes
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.

Arbre de décision



Construction

Construction d'un arbre de décision

Problème

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

Objectif

- Être efficace en généralisation

⇒ Être capable de classer correctement un nouvel objet (exemple).

Un algorithme horrible!!

- Générer tous les arbres de décision possibles.
- Tester combien chaque arbre décrit l'ensemble d'apprentissage.
- Choisir le meilleur arbre de décision.

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.

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)

Procédure de construction (1)

Processus récursif

- L'arbre commence à un nœud représentant 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 attributs qui séparent le mieux les objets en classes homogènes.
  - La récursion s'arrête quand au moins l'un des critères d'arrêt est vérifié.

Procédure de construction (2)

- Recherche à chaque niveau, l'attribut le plus discriminant.

- Partition (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 T_1, T_2, …T_k;
  - Pour i = 1 à k faire Partition(T_i);

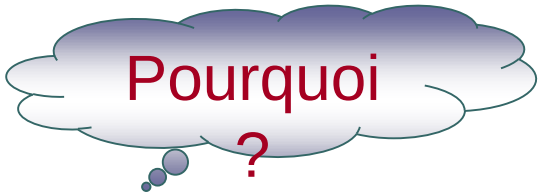
Paramètres

Mesure de sélection d'attributs

Stratégie de partitionnement

Critères d'arrêt

Comment choisir l'attribut ?

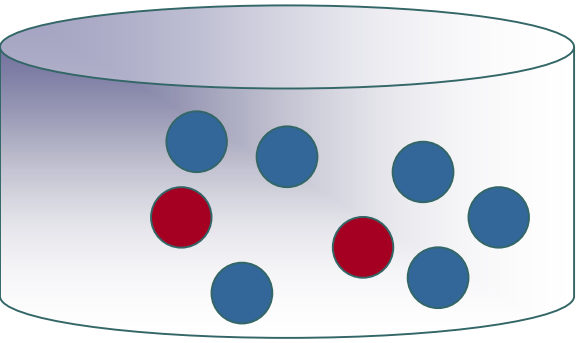


- Personne ne le sait !!
- Plusieurs mesures ont été proposées.
  - Gain d'information
  - Indice de Gini (Gini(D) = 1- $\sum_j (p_j)^2$)
  - Ratio de gain
  - .
  - .
  - .

Mesure de l'information

- L'entropie de Shannon exprime la quantité d'information.

⇨ Le nombre de bits nécessaires pour coder l'information.



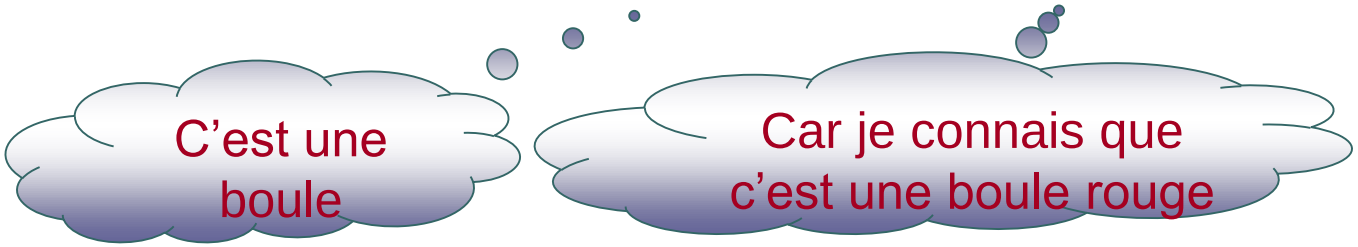
La probabilité de tirer une boule bleue est $\frac{6}{6+2} = \frac{3}{4}$

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

Apport de l'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.
- 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:

$$\log_2(N) \text{ bits} - \log_2(Nr) \text{ bits}$$

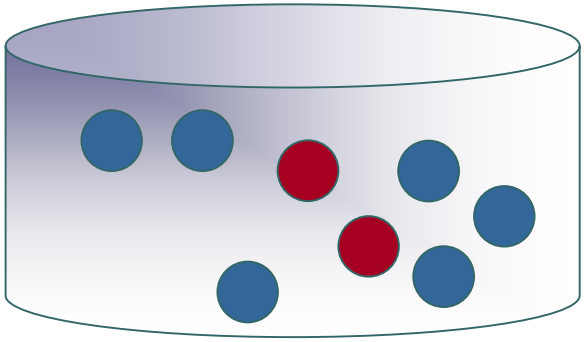


- Si je tire une boule au hasard et qu'on me donne sa couleur, l'information acquise sera:

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

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

Exemple



? ?

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

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

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

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

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

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 maximal quand $\forall$ i/ $p_i = 1/n$

Classes équiprobables

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 similiaire 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.

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, ça peut générer un arbre de décision d'une profondeur de 1 (ou faible) qui ne sera pas très bon pour les instances futures.

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.

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

Proportion d'information génerée par T et utile pour la classification

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

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

Quand s'arrêter ?

Si tous les objets appartiennent à la même classe.

S'il n'y a plus d'attributs à tester.

Feuille vide

Il n'y a pas d'objets avec la valeur d'attribut.

Tous les ratios de gain sont ≤ 0

Absence d'apport informationnel des attributs.

Info

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

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

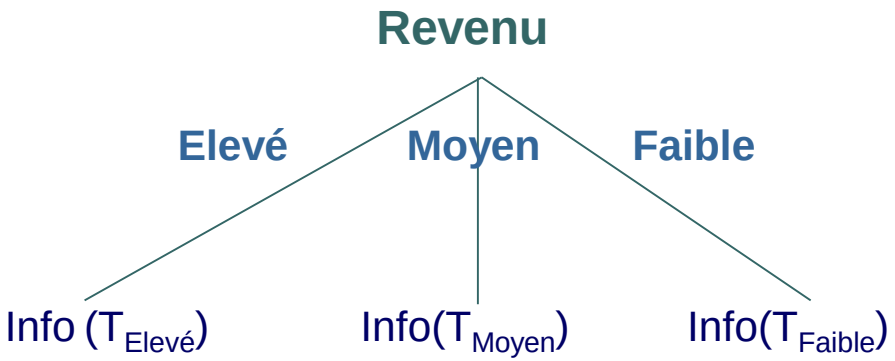
$$Info(T) = - 3/10 \log_2 3/10 - 5/10 \log_2 5/10 - 2/10 \log_2 2/10 = 1.485$$

$Info_{Revenu}(T)$

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}



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

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

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

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

Gain ratio(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

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

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

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

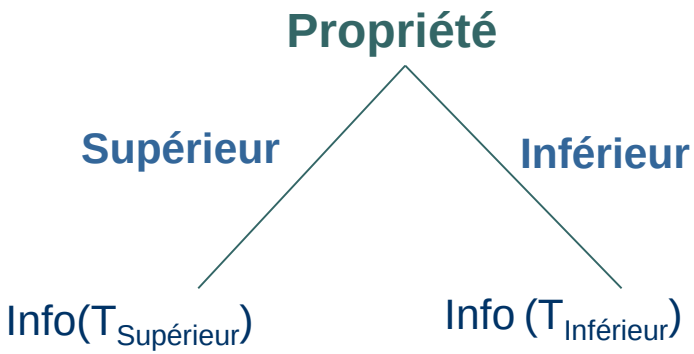
$$Gain\ Ratio(T, Revenu) = \frac{0.761}{1.522} = 0.5$$

$Info_{Propriété}(T)$

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}



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

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

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

Gain ratio (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

$$Gain(T, Propriété) = Info(T) - Info_{Propriété}(T) = 0.514$$

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

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

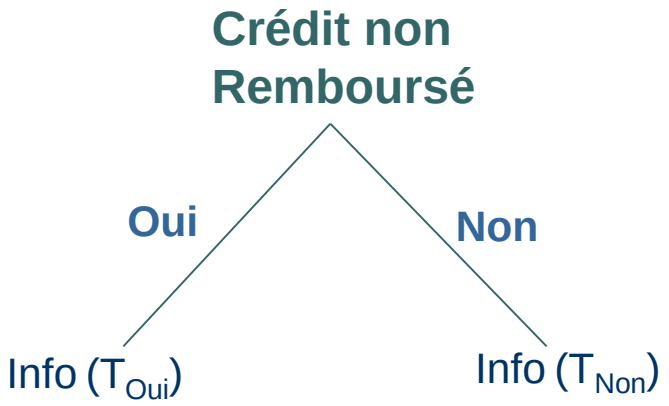
$$Gain\ Ratio(T, Propriété) = \frac{0.514}{1} = 0.514$$

$Info_{Crédit\ non\ remboursé}(T)$

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

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

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



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

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

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

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

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

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

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

Arbre de décision: Niveau 1

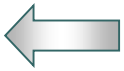
Gain Ratio(T, Revenu) = 0.5

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

Inférieur

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

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

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

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

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

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

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

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

$Gain(S, Revenu) = 0.02$

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

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

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) = $-\ 2/5\ \log_2\ 2/5 - 3/5\ \log_2\ 3/5 = 0.971$

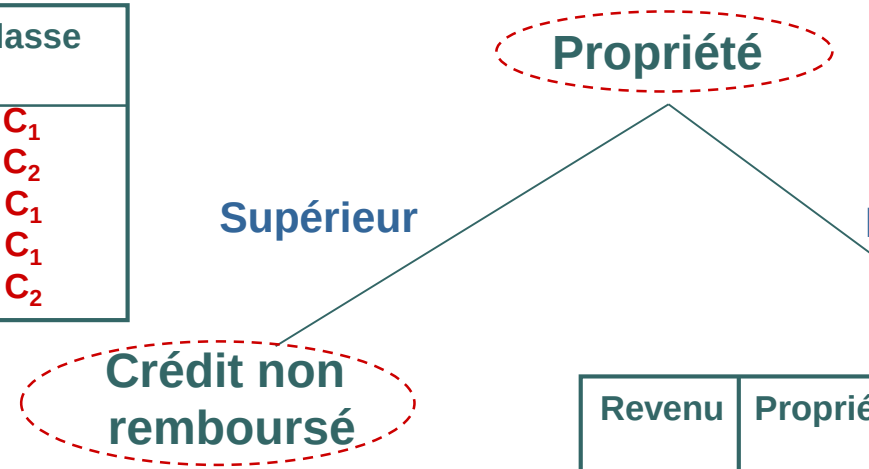
Gain Ratio(S, Crédit non remboursé) = 1

Arbre de décision: Niveau 2 (1)

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

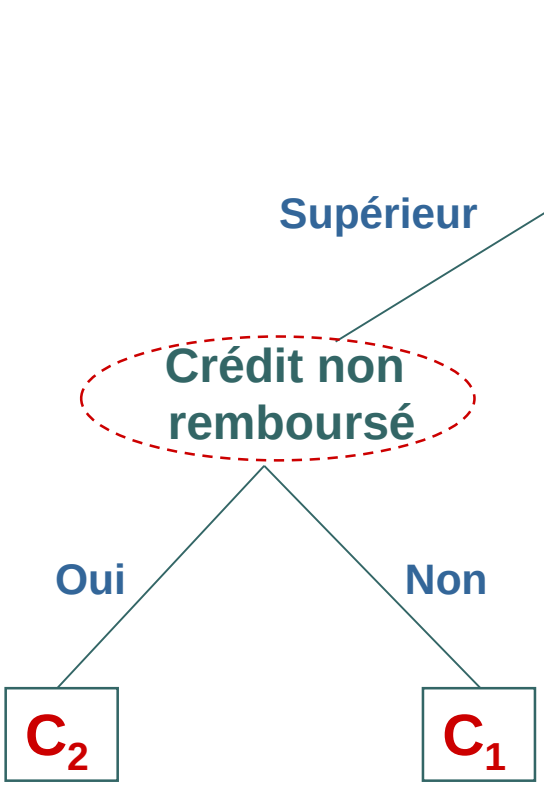


Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

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

Arbre de décision: Niveau 2 (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

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

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

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

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

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

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

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

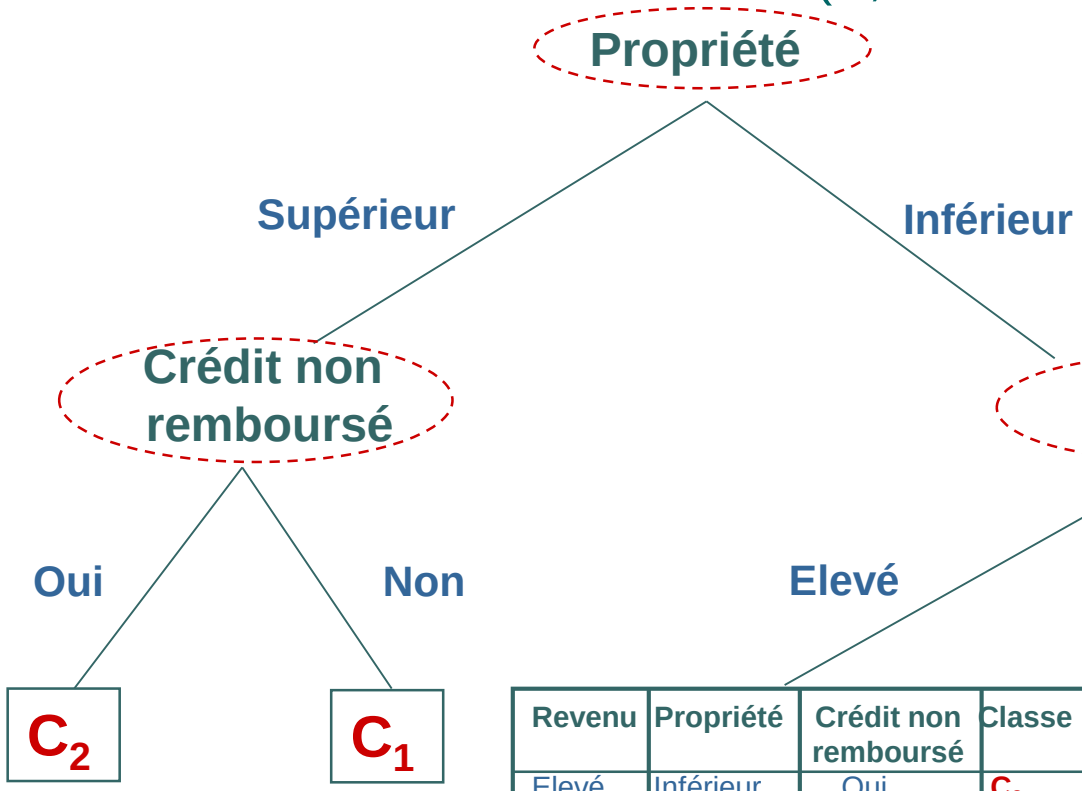
Gain(I, Crédit non remboursé) = 0.02

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

Arbre de décision: Niveau 2 (3)

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



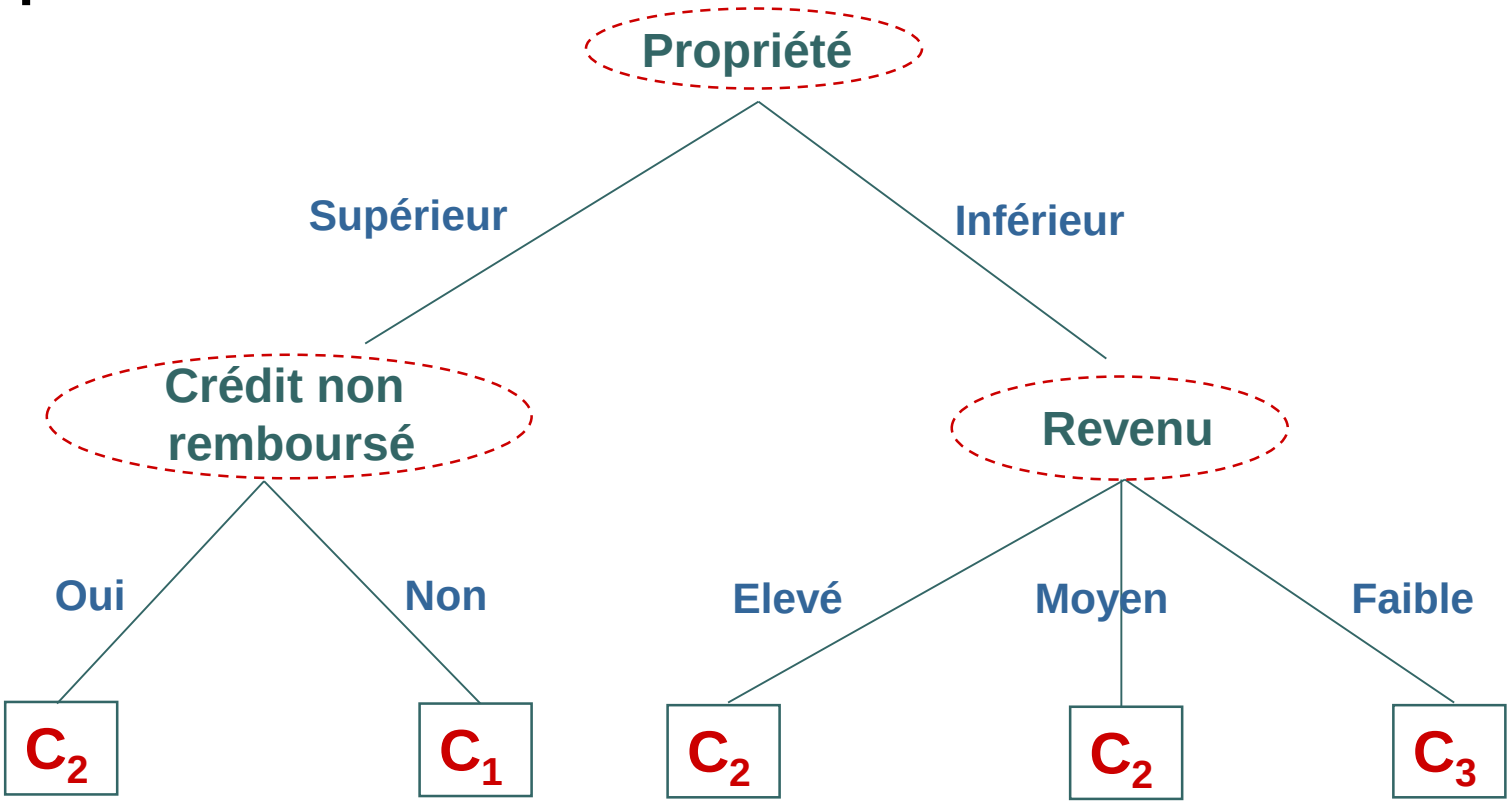
Revenu	Propriété	Crédit non remboursé	Classe
Elevé	Inférieur	Oui	C_2
Moyen	Inférieur	Non	C_2
Moyen	Inférieur	Oui	C_2
Faible	Inférieur	Non	C_3
Faible	Inférieur	Oui	C_3

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

Arbre de décision final



Travail à faire

1) Construire l'arbre de décision correspondant à l'ensemble d'apprentissage suivant :

Age	Concurrence	Type	Profit
Agé	Non	Software	Baisse
Moyen	Oui	Software	Baisse
Moyen	Non	Hardware	Hausse
Agé	Non	Hardware	Baisse
Récent	Non	Hardware	Hausse
Récent	Non	Software	Hausse
Moyen	Non	Software	Hausse
Récent	Oui	Software	Hausse
Moyen	Oui	Hardware	Baisse
Agé	Oui	Software	Baisse

2) Vérifier que Weka (Logiciel à télécharger d'internet) donne le même résultat pour le même ensemble d'apprentissage.

Solution

1) Niveau 1

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

Age

Info($T_{Agé}$) = - 3/3 $\log_2$ 3/3 = 0

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

Info($T_{Récent}$) = - 3/3 $\log_2$ 3/3 = 0

$Info_{Age}(T)$= 3/10 Info ($T_{Agé}$) + 4/10 Info (T_{Moyen}) +3/10 Info ($T_{Récent}$)=0,4

Gain(Age) = 1-0,4 = 0,6

Split Info(T, Age) = - 3/10 $\log_2$ 3/10 - 4/10 $\log_2$ 4/10 - 3/10 $\log_2$ 3/10 = 1.571

Gain Ratio(Age) = 0.382

Concurrence

Info(T_{Oui}) = 1/4 $\log_2$ 1/4 - 3/4 $\log_2$ 3/4 = 0,811

Info(T_{Non}) = - 4/6 $\log_2$ 4/6 - 2/6 $\log_2$ 2/6 = 0,918

$Info_{Concurrence}(T)$= 4/10 Info (T_{Oui}) + 6/10 Info (T_{Non}) = 0,875

Gain(Concurrence) = 1-0,4 = 0,125

Split Info(T, Concurrence) = - 4/10 $\log_2$ 4/10 - 6/10 $\log_2$ 6/10 = 0.971

Gain Ratio(concurrence) = 0.129

Solution

Type

$Info(T_{Software}) = - 3/6 \log_2 3/6 - 3/6 \log_2 3/6 = 1$

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

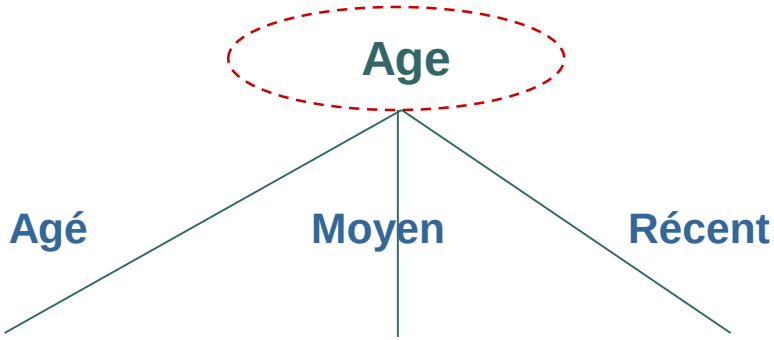
$Info_{Type}(T) = 6/10\ Info\ (T_{Software}) + 4/10\ Info\ (T_{Hardware}) = 1$

$Gain(Age) = 1-1 = 0$

$Split\ Info(T, Type) = - 6/10 \log_2 6/10 - 4/10 \log_2 4/10 = 0,971$

$Gain\ Ratio(Type) = 0$

Donc Age meilleur attribut



Solution

Niveau 2

Pour valeur = agé et valeur = récent

ce sont des feuilles

Pour valeur = Moyen

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

Concurrence

$Info(S_{Oui}) = 2/2 \log_2 2/2 = 0$

$Info(S_{Non}) = - 2/2 \log_2 2/2 = 0$

$Info_{Concurrence}(S) = 2/4\ Info\ (S_{Oui}) + 2/4\ Info\ (S_{Non}) = 0$

Gain(Concurrence) = 1-0 = 1

Split Info(S, Concurrence) = $- 2/4 \log_2 2/4 - 2/4 \log_2 2/4 = 1$

Gain Ratio(concurrence) = 1

Type

$Info(S_{Software}) = - 1/2 \log_2 1/2 - 1/2 \log_2 1/2 = 1$

$Info(S_{Hardware}) = - 1/2 \log_2 1/2 - 1/2 \log_2 1/2 = 1$

$Info_{Type}(S) = 2/4\ Info\ (S_{Software}) + 2/4\ Info\ (S_{Hardware}) = 1$

Gain(Type) = 1-1 = 0

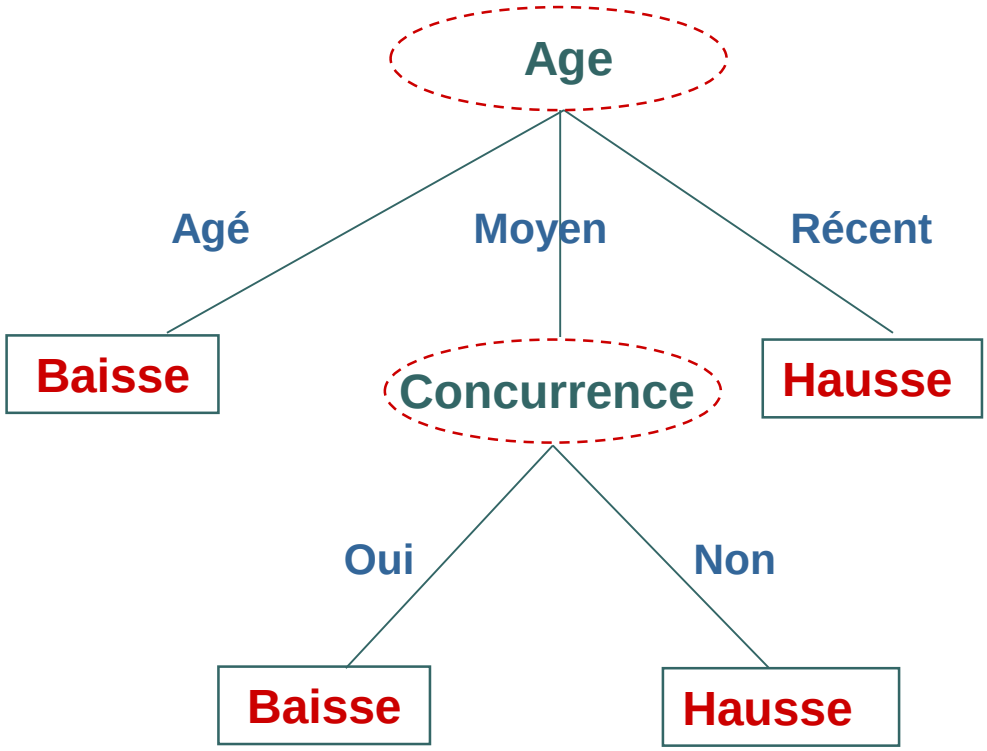
Split Info(T, Type) = $- 2/4 \log_2 2/4 - 2/4 \log_2 2/4 = 1$

Gain Ratio(type) = 0

Donc concurrence meilleur attribut

Solution

Pour valeur = Oui et Non
ce sont des feuilles



2) On a les mêmes résultats qu'avec Weka.

Algorithmes incrémentaux

- Comment traiter des objets qui arrivent continûment (dans l'ensemble d'apprentissage)?

  ⇨ L'ensemble d'apprentissage n'est pas complet.

  - ID4 (Schlimmer et Fisher, 1986)
  - ID5 (Utgoff, 1988)
  - ID5R (Utgoff, 1989)

L'ordre ne doit pas faire varier le résultat

Classification

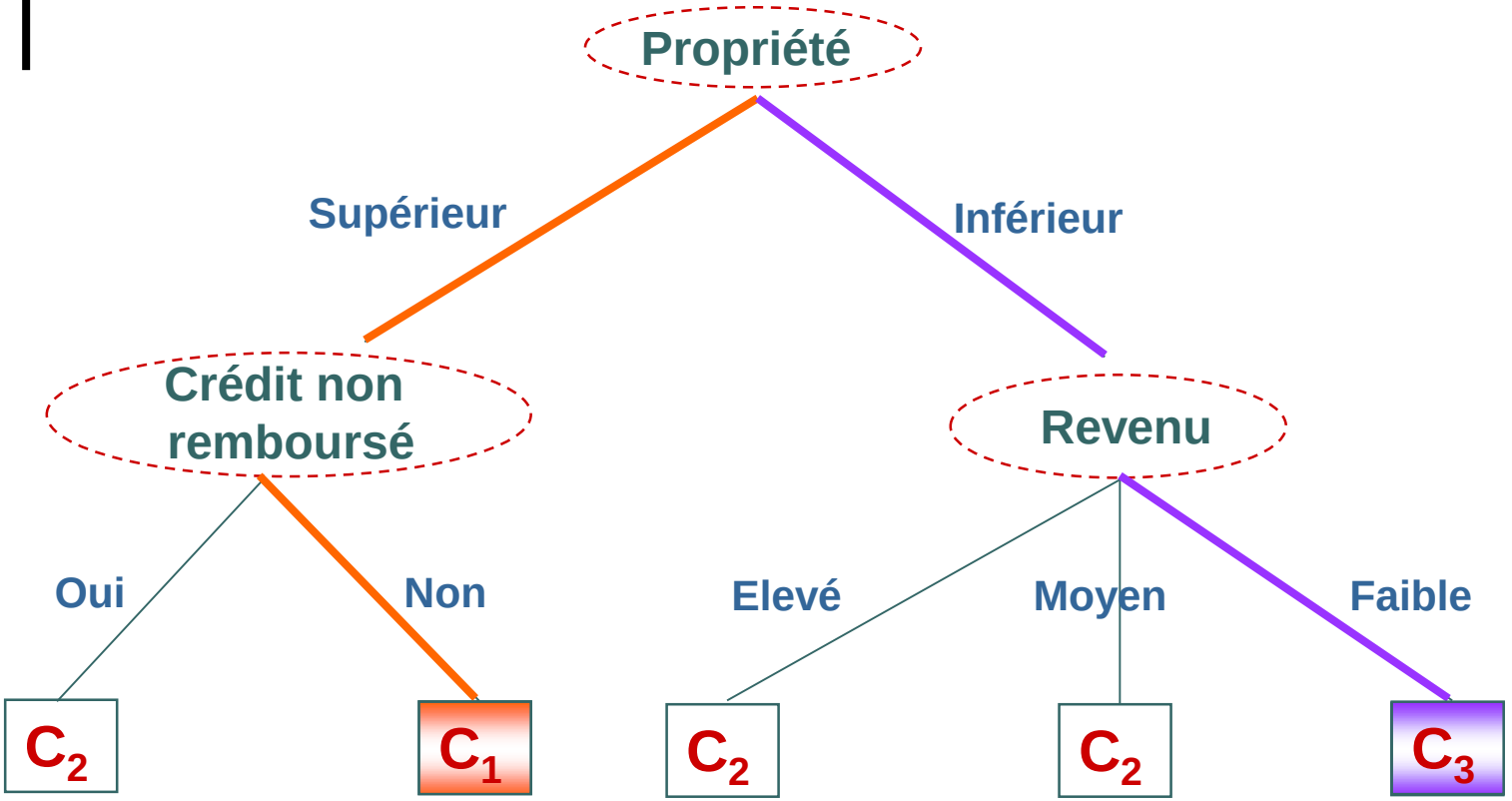
Classification (1)

- 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)



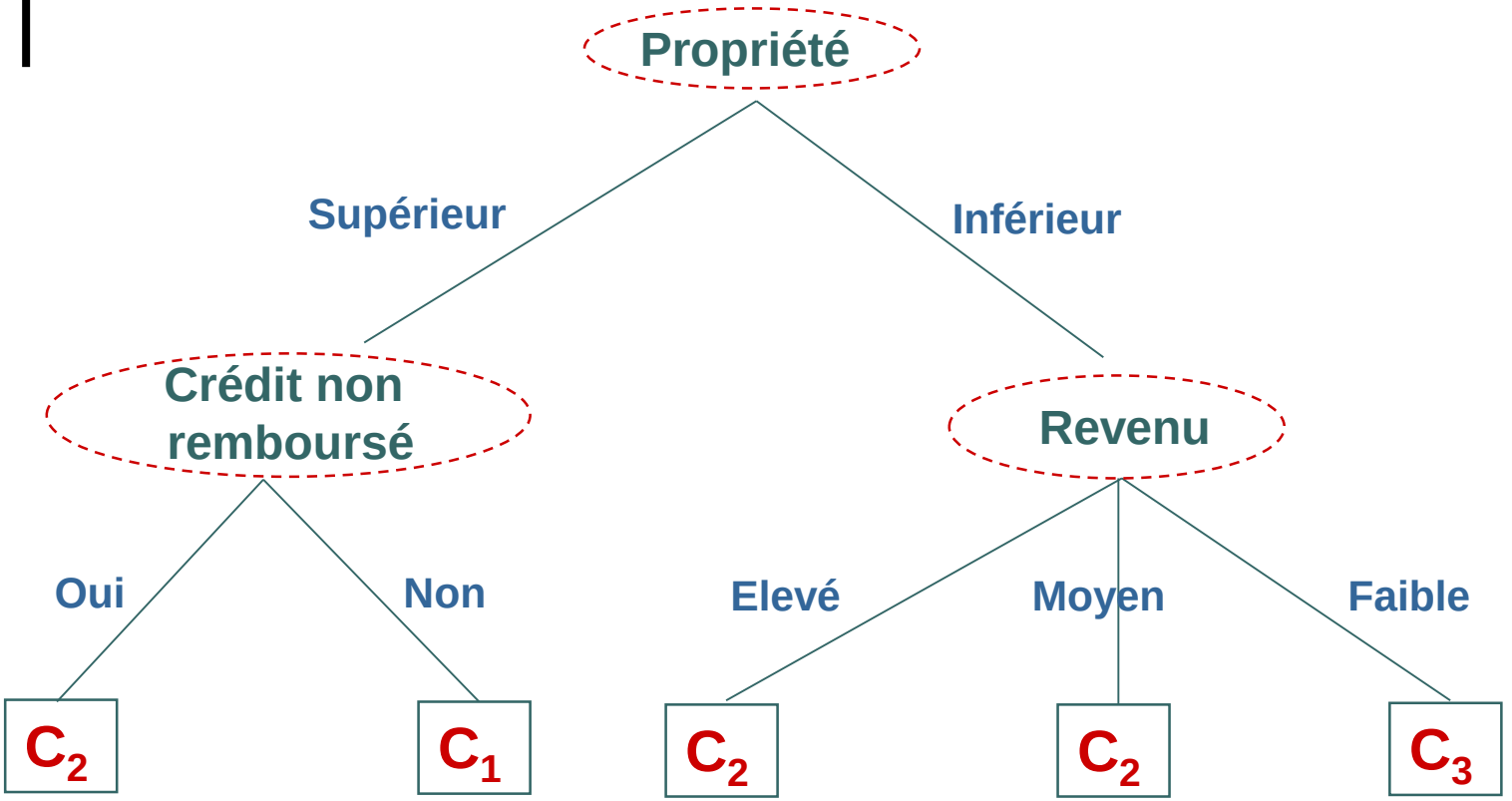
À classer ?

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

Convertir l’arbre en règles (1)

- 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.

Convertir l'arbre en règles (2)



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

Elagage

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 d'un arbre

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.

Comment élaguer ?

Pré-élagage (pre-pruning)

- Arrêter le développement d'un noeud.
- Ne pas partitionner si le résultat va s'affaiblir

⇒ Créer une feuille si la classe est majoritairement représentée (seuil).

S'arrêter avant d'engendrer un nœud 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.
- Pour chaque noeud de décision, voir si ça sera meilleur de le remplacer par:
  - Une feuille.
  - Un de ses fils (le plus fréquent).

Générer l'arbre entier, puis élaguer

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)

Mesures de qualité de l’arbre

- PCC: Pourcentage de Classifcation Correcte.
- Complexité
  - Taille de l’arbre
  - Nombre de feuilles
- Temps
- Trouver un arbre de décision minimal consistant avec l’ensemble d’apprentissage est un problème NP-complet.

Attributs à valeurs continues

Problème des attributs continus

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

⇒ Découper en sous-ensembles ordonnés.

  - Division en segments $[a_0,a_1[$, $[a_1,a_2[$, …., $[a_{n-1},a_n]$.
  - Utiliser moyenne, médiane, …

- Investiguer différents cas et retenir le meilleur.

Attributs à valeurs continues

- 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	40	48	60	72	80	90
JouerTennis	Non	Non	Oui	Oui	Oui	Non

  - On met les valeurs en ordre croissant et on regarde les endroits ou 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 (ou ratio gain).

Travail à faire

Soient les valeurs de l'attribut Température:

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.

Solution (1)

Il faut tout d'abord ordonner selon la valeur de l'attribut (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 2 coupures binaires possibles avec changement de classe, il faut voir laquelle apporte le meilleur ratio de gain ?

Les coupures sont 12 et 25 (où il y a changement de classe).

Solution (2)

Info (T) = $2/6 \log_2 2/6 - 4/6 \log_2 4/6 = 0{,}918$

Info(T, 12) = 2/6 Info(T, <12) + 4/6 Info(T, >12)
Info(T, <12) = 0
Info(T, >12)= $-2/4\log_2 2/4 - 2/4\log_2 2/4 = 1$
Info(T, 12) = 0,666
Gain (T, 12) = 0,918 – 0,666 = 0, 252
Split Info(T, 12) = $-2/6 \log_2 2/6 - 4/6 \log_2 4/6 = 0{,}918$
Ratio Gain (T, 12) = 0,252/0,918 = 0,274

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
Gain (T, 25) = 0,918 – 0,666 = 0, 252
Split Info(T, 25) = $-4/6 \log_2 4/6 - 2/6 \log_2 2/6 = 0{,}918$
Ratio Gain (T, 25) = 0,252/0,918 = 0,274

Donc ici on peut choisir l'une des coupures 12 ou 25.

Attributs à valeurs manquantes

Valeurs manquantes d'attributs

- Attribuer la valeur la plus fréquente parmi les exemples du noeud
- Attribuer la valeur la plus fréquente dans l'ensemble d'apprentissage.
- Attribuer une probabilité pour chaque valeur de l'attribut.

Variantes d'arbres de décision

Arbres de décision obliques

- Nouveaux attributs : combinaisons linéaires d'anciens attributs.

cette variante permet de lever la contrainte « parallèle aux axes » lors du partitionnement dans l'espace de représentation.

- Généralement, l'arbre produit est plus concis. En revanche la lecture des règles de décision est un peu plus compliquée.

Arbres de décision avec options

- Les arbres de décision avec options représentent une généralisation des arbres de décsion standards. En plus des noeuds de decision et des feuilles, ils contiennent des noeuds d'option.

- Un nœud d'option est un noeud qui permet d'avoir plusieurs tests c.à.d contient plus qu'un attribut.

Arbres de décision paresseux
- Lazy decision trees -

- En théorie, on veut sélectionner le meilleur arbre de décision pour chaque instance à tester, c.à.d choisir le meilleur arbre parmi les arbres de décision possibles.

- En pratique, seul le chemin de l'instance test est à construire.

Bagging et boosting

Bagging

Bagging = Bootsrap 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.

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 quand les classifieurs présentent de mauvais résultats.

- Les classifieurs peuvent être de types différents.

Conclusion et perspectives (1)

- 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).

Conclusion et perspectives (2)

- Arbres de décision et attributs continus (Fayyad, Irani, 1992; Quinlan, 1993).
- Arbres de décision et élagage (Mingers, 1989).
- Arbres de décision obliques (Mruthy et al., 1994).
- Arbres de décision avec options (Kohavi et Kunz, 1997).
- Arbres de décision Paresseux (Friedman et et al., 1996).
- Arbres de décision et Bagging (Quinlan, 1997).
- Arbres de décision et Boosting (Quinlan, 1997).

.

.

.

Conclusion et perspectives (3)

- Arbres de décision et théories de l'incertain.
  - Arbres de décision probabilistes (Quinlan, 1990).
  - Arbres de décision flous (Y. Yuan, M. J. Shaw, 1995; Janikow, 1998).
  - Arbres de décision crédibilistes (Elouedi, Mellouli, Smets, 2000. Denoeux, M. Skarstein-Bjanger, 2000, S. Trabelsi, Elouedi, Mellouli, 2007, S. Trabelsi Elouedi, El Aroaui, A. Trabelsi, Elouedi Lefèvre 2016).
  - Arbres de décision qualitatifs avec options (Jenhani, Elouedi, Ben Amor, Mellouli, 2005).
  - Arbres de décision possibilistes (Ben Amor, BenFerhat, Elouedi, 2004, Jenhani et al. 2007, Boutaib et Elouedi 2018).

TD

Exercice 1

1) Construire l'arbre de décision correspondant à l'ensemble d'apprentissage suivant :

Age	Concurrence	Type	Profit
Agé	Non	Software	Baisse
Moyen	Oui	Software	Baisse
Moyen	Non	Hardware	Hausse
Agé	Non	Hardware	Baisse
Récent	Non	Hardware	Hausse
Récent	Non	Software	Hausse
Moyen	Non	Software	Hausse
Récent	Oui	Software	Hausse
Moyen	Oui	Hardware	Baisse
Agé	Oui	Software	Baisse

2) Vérifier que Weka (Logiciel à télécharger d'internet) donne le même résultat pour le même ensemble d'apprentissage.

Exercice 2

Choisir 3 bases dans weka

1) Construire et tester les arbres de décision (sans élagage et avec élagage) correspondant à ces 3 bases.

2) Pour chaque base, comparer les résultats avec élagage et sans élagage.

Exercice 3

1) Expliquer le traitement des attributs à valeurs continues par l'algorithme C4.5

2) Soient les valeurs de l'attribut Température:

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.

Articles à exposer

1) Ensemble-based classifiers

2) Pruning belief decision tree methods in averaging and conjunctive approaches