

Université de Bordj Bou Arreridj
Faculté Mathématiques Informatique
Département d'informatique

1^{ère} Master Informatique décisionnelle

Module : Fouille et Extraction de de données (F.E.D)

Chapitre 04

Arbre de décision

1

1

Dr. Boutouhami-Nouioua
boutouhaminouioua@gmail.com

Arbre de décision

- Ensemble de règles de classification basant leur décision sur des tests associés aux attributs, organisés de manière arborescente.
- **Gros avantages des arbres de décision**
 - Classification très rapides.
 - Décisions aisément interprétables,

Arbre de décision

Principe

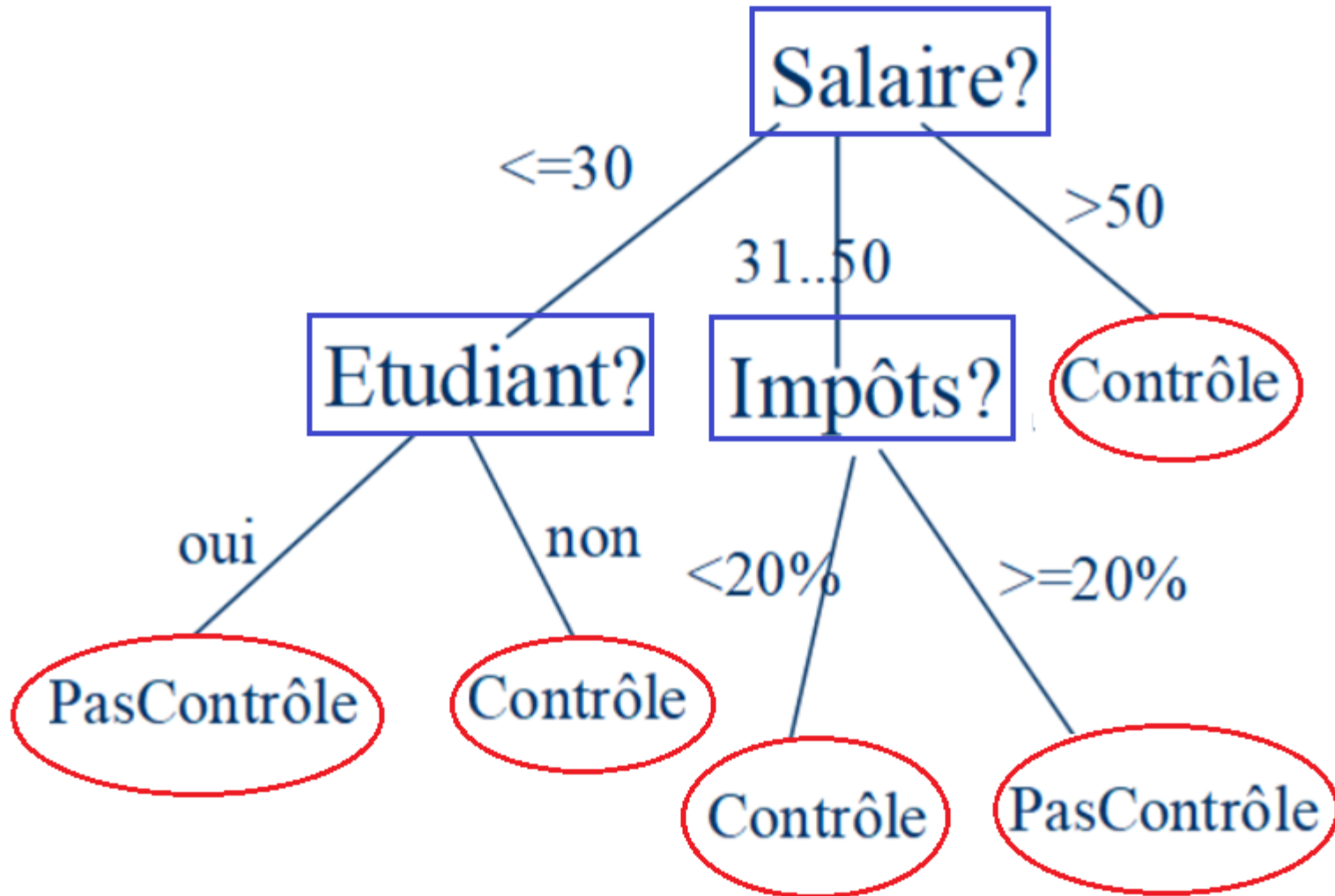
Prédire la valeur d'un attribut(variable cible) à partir d'un ensemble de valeurs d'attributs (variables prédictives).

Un arbre est composé :

- **de nœuds** : classes d'individus de plus en plus fines depuis la racine.
- **d'arcs (arcs)** : prédicats de partitionnement de la classe source.

Exemple 1

Faut-il envoyer un contrôleur fiscal?

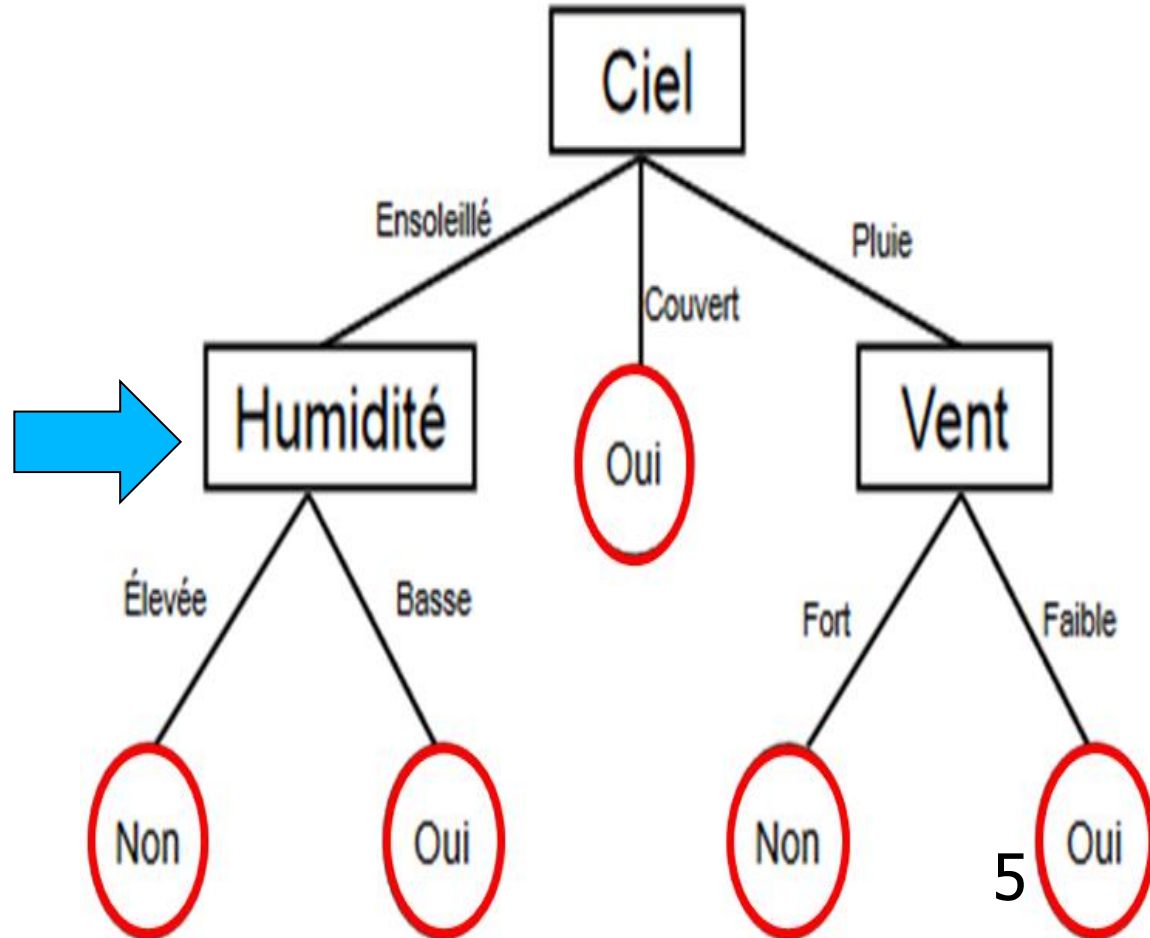


Exemple 1

Est-ce que les conditions sont favorable pour jouer au tennis?

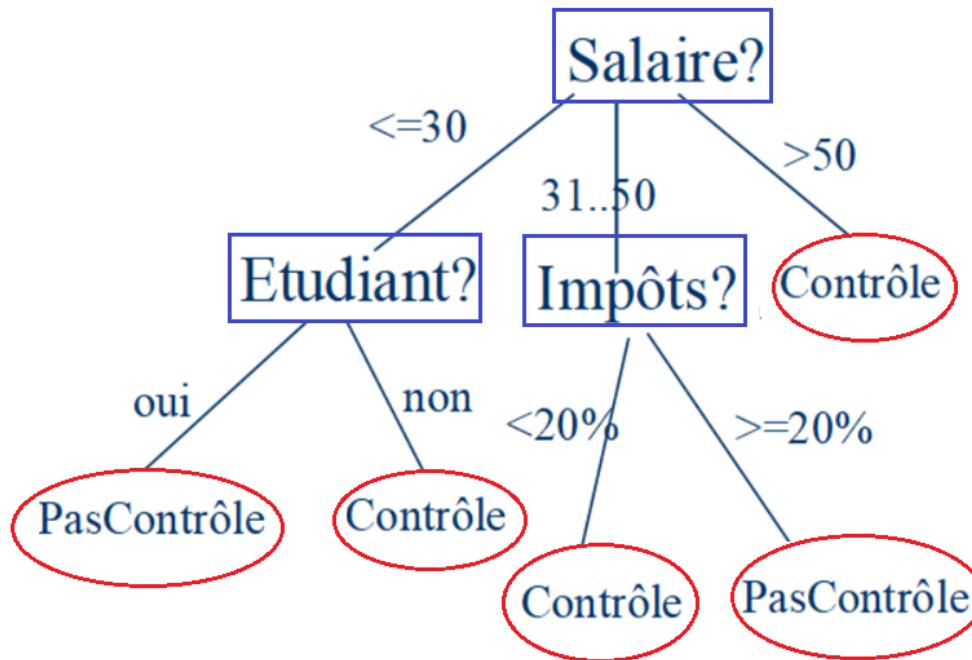
Echantillon d'apprentissage

Ciel	Humidité	Vent	Jouer
Ensoleillé	Elevée	Faible	NON
Ensoleillé	Elevée	Fort	NON
Couvert	Basse	Faible	OUI
Pluie	Basse	Faible	OUI
Pluie	Elevée	Fort	NON
Ensoleillé	Basse	Faible	NON
Couvert	Basse	Fort	OUI
Couvert	Basse	Faible	OUI
Pluie	Basse	Faible	NON



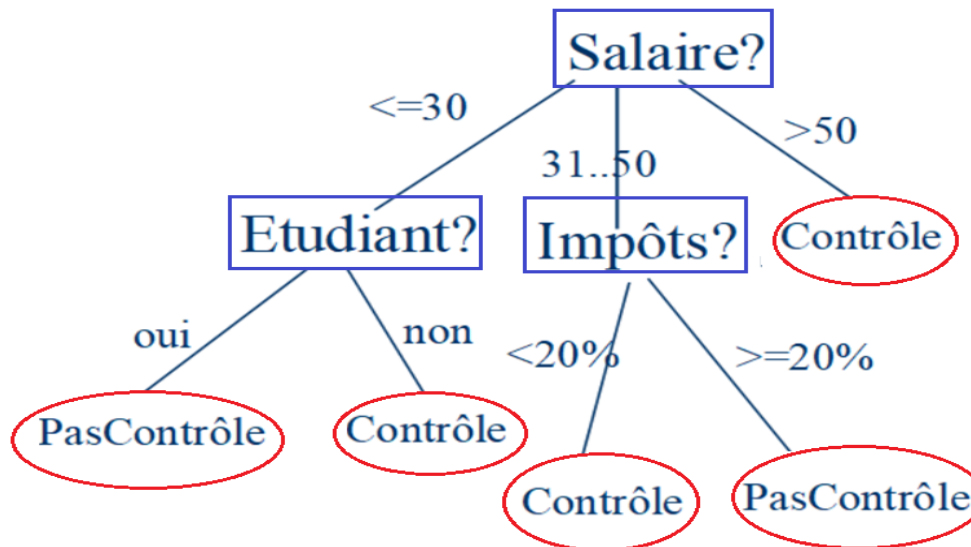
Un peu de vocabulaire

- Un arbre est constitué de **nœuds** connectés entre eux par des **arcs**.
- Un **arc** entre deux nœuds est **orientée** : l'un des nœuds de la connexion est dit «**nœud-parent**», et l'autre «**nœud-enfant**».



Un peu de vocabulaire

- Chaque nœud est connecté à un et un seul nœud parent, sauf le **nœud racine (racine)** qui n'a pas de parent.
- Chaque nœud peut être connecté à 0 ou n nœuds enfants.
- Un nœud qui *n'a pas* de nœuds enfants est appelé «**nœud feuille**» ou «**feuille**».
- Un arbre de décision est constitué principalement de **nœuds de décision**.

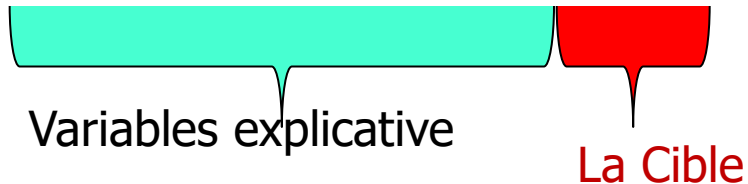


Un peu de vocabulaire

Un arbre de décision travaille sur une variable cible avec plusieurs variables prédictives.

- **La variable cible** est **qualitative** (comme pour toutes les classifications supervisées). On aura donc une classification en **c** classes (généralement 2 classes).
- **Les prédicteurs** peuvent être des variables **qualitative** ou **quantitative**.

Ciel	Humidité	Vent	Jouer
Ensoleillé	37,5	Faible	NON
Ensoleillé	40	Fort	NON
Couvert	32	Faible	OUI

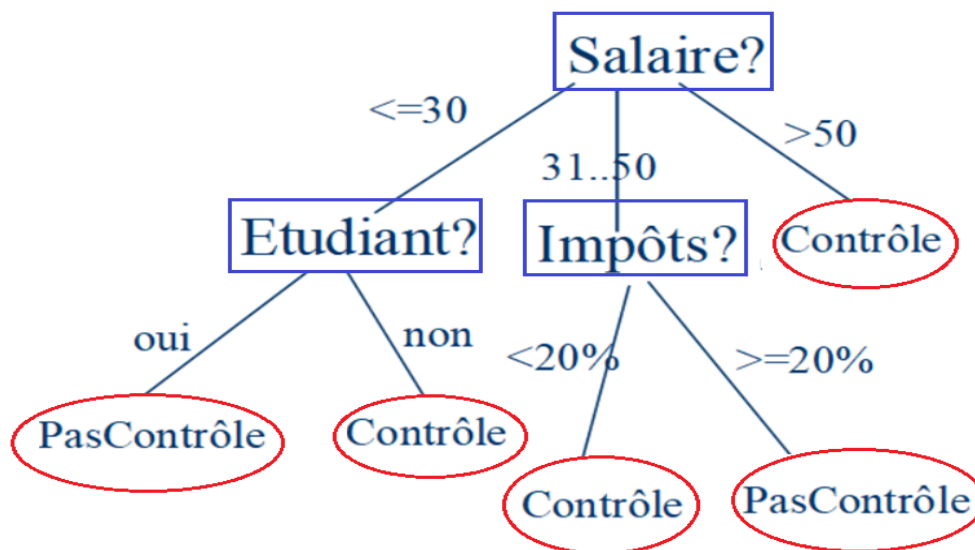


Variables explicative

La Cible

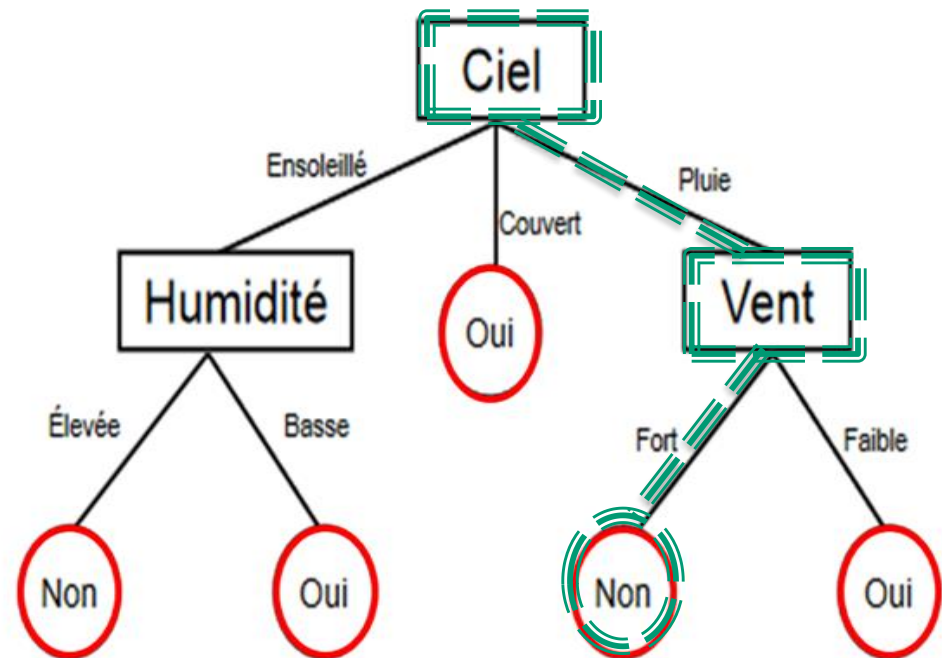
Un peu de vocabulaire

- Chaque nœud non-feuille (**nœud de décision**) correspond à une variable prédictive.
- Chaque nœud feuille (feuille) correspond à la variable cible.
- Chaque arc correspond à une valeur pour la variable prédictive du nœud parent (ou un ensemble de valeurs).



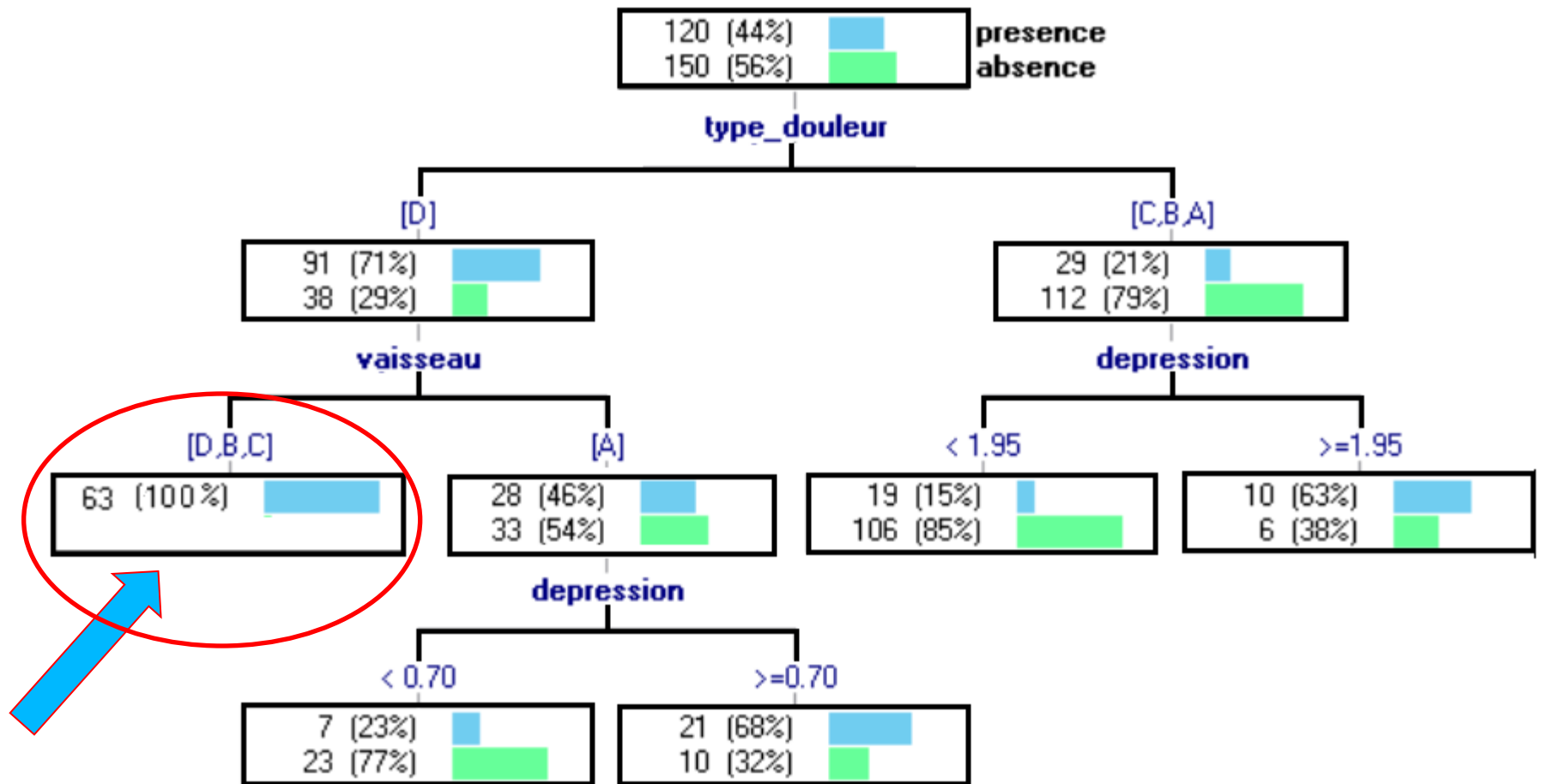
Un peu de vocabulaire

- Un **chemin** est un parcours du **nœud racine** jusqu'à un **nœud feuille**.
- Un **chemin** précise la ou les valeurs prévues pour les enregistrements de la variable cible pour ce chemin particulier.



Un peu de vocabulaire

Un nœud feuille est *pur* s'il contient des données d'une seule classe.



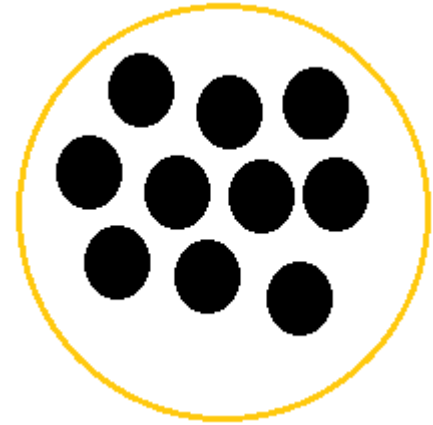
Un peu de vocabulaire



Non-homogène,
Impureté forte



Homogène,
Impureté faible



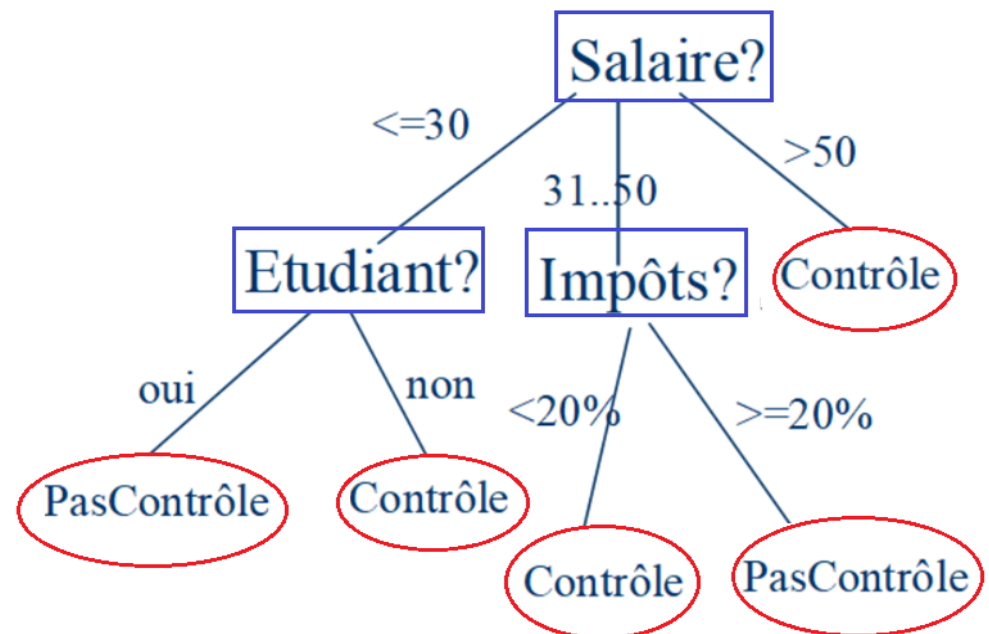
Pure

Arbre de décision : exemple

Décider s'il faut envoyer un contrôleur fiscal ou pas en fonction du fait d'être étudiants, du salaire et des impôts

Arbre de décision :

- 2 classes pour la variable cible **CONTROLLER** : contrôle, pas-contrôle
- 3 variables : Etudiant, Salaire, Impôts



Algorithme d'apprentissage générique

Entrée

- n individus
- p variables continues ou discrètes
- Une variables supplémentaires contenant la classe de chaque individus (c classes)
 - c valeurs possibles de la variable cible

Sortie

- L'arbre de décision T

Algorithme d'apprentissage générique

Notations

- $N(p)$ = nombre d'individus associés au **nœud p** i.e, à la position p .
- $N(k/p)$ =nombre d'individus appartenant à la classe k en sachant qu'ils sont associés au **nœud p** (à la position p).
- $P(k/p)=N(k/p)/N(p)$: proportion des individus appartenant à la classe k parmi ceux du **nœud p** (de la position p)

Algorithme d'apprentissage générique

Entrée : échantillon S

Début

Initialiser l'arbre courant à l'arbre vide ; la racine est le noeud courant

répéter

Décider si le noeud courant est terminal

Si le noeud est terminal **alors** Lui affecter une classe

sinon *Sélectionner* un test et créer autant de
nouveaux noeuds fils qu'il y a de réponses
possibles au test

FinSi

Passer au noeud suivant non exploré s'il en existe Jusqu'à obtenir un arbre
de décision

Jusqu'à obtenir un arbre de décision

Fin

Algorithme d'apprentissage générique

Nœud Terminal (feuille)

- Lorsque tous (ou presque) les exemples associés à ce nœud sont dans la même classe,
- Lorsqu'il n'y a plus d'attributs non utilisés dans la arc correspondante.

Quelle classe à un nœud terminal (feuille)

- La classe majoritaire.
- La classe la plus représentée, si égalité.

Construction de l'arbre de décision : Exemple

But Construire un arbre de décision qui classe et détermine les caractéristiques des clients qui consultent leurs comptes sur internet

Variables

- ***M*** : moyenne des montants sur le compte
- ***A*** : âge du client
- ***R*** : lieu de résidence du client
- ***E*** : le client à des études supérieures ?
- ***I*** : le client consulte ses comptes sur internet ?

Construction de l'arbre de décision : Exemple

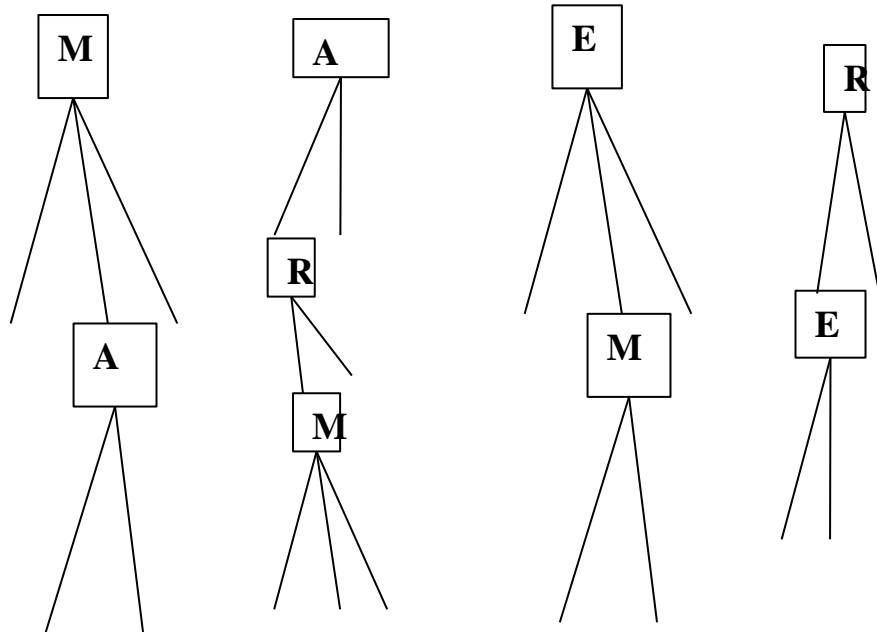
Exemple

Construire un arbre de décision qui classe et détermine les caractéristiques des clients qui consultent leurs comptes sur internet

Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

Construction de l'arbre de décision : Exemple

Echantillon d'apprentissage

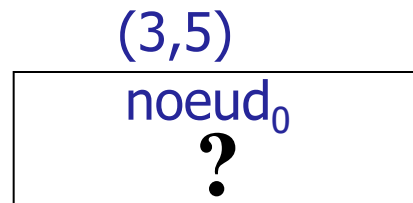


Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

Construction de l'arbre de décision : Exemple

- La construction est descendante
- Au début tous les individus sont regroupés
- Est-ce que *le nœud initial* (3, 5) [3 oui, 5 non] c'est un nœud terminal ou est-ce qu'on peut construire un test sur une variable qui permettra de mieux discriminer (deviser, partitionner) les individus ?
- Quatre constructions possibles, suivant les variables Montant (M), Age (A), Résidence (R) et Etudes (E)

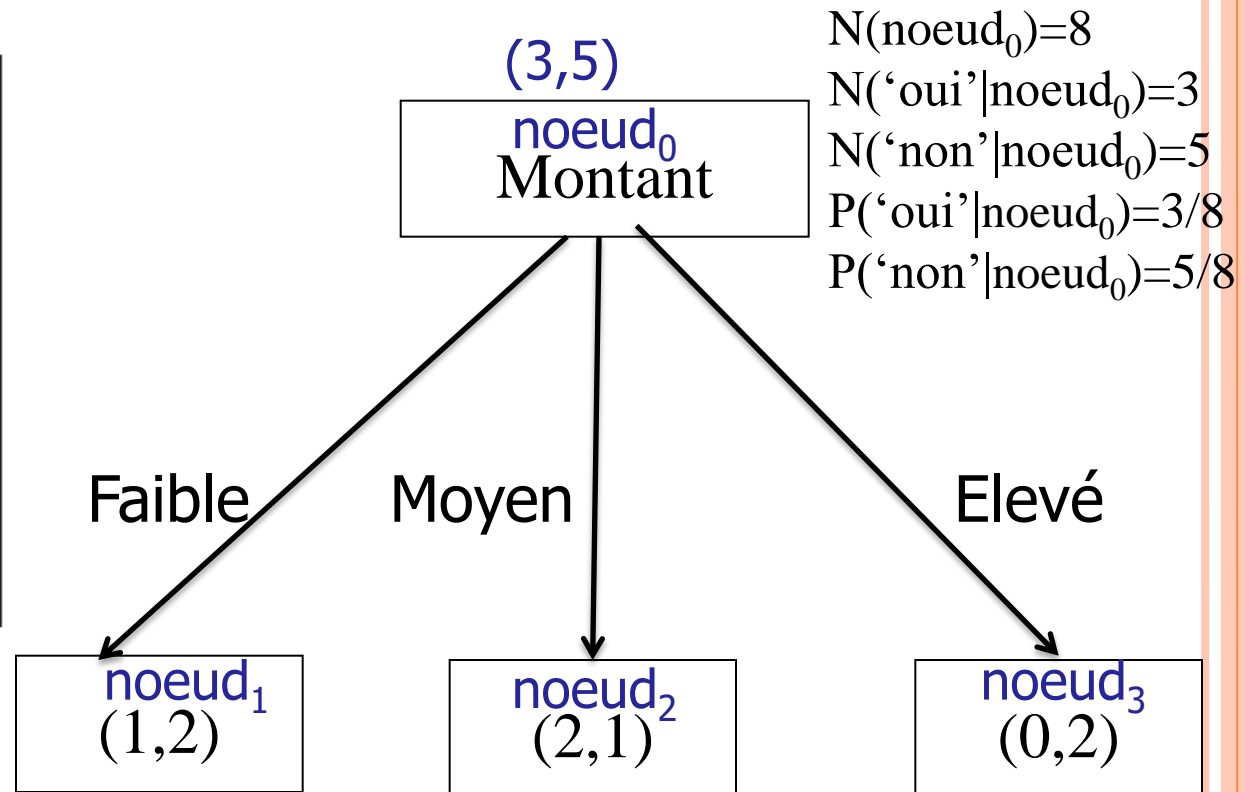
Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non



Construction de l'arbre de décision : Exemple

1. Construction selon la variable Montant (M)

Client	M	I
1	moyen	oui
2	élevé	non
3	faible	non
4	faible	oui
5	moyen	oui
6	élevé	non
7	moyen	non
8	faible	non



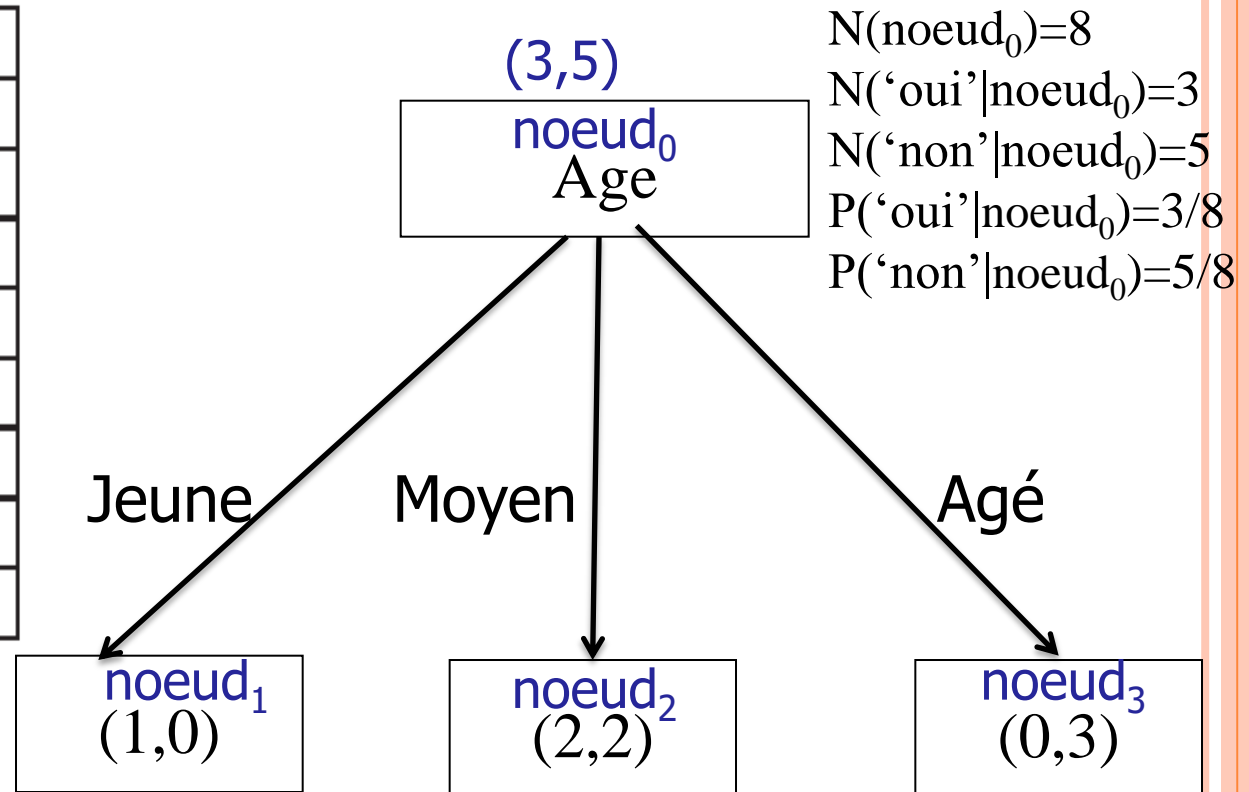
$N(\text{noeud}_0)=8$
 $N(\text{'oui'}|\text{noeud}_0)=3$
 $N(\text{'non'}|\text{noeud}_0)=5$
 $P(\text{'oui'}|\text{noeud}_0)=3/8$
 $P(\text{'non'}|\text{noeud}_0)=5/8$

$N(\text{noeud}_1)=3$
 $N(\text{'oui'}|\text{noeud}_1)=1$
 $N(\text{'non'}|\text{noeud}_1)=2$
 $P(\text{'oui'}|\text{noeud}_1)=1/3$
 $P(\text{'non'}|\text{noeud}_1)=2/3$

Construction de l'arbre de décision : Exemple

2. Construction selon la variable Age (A)

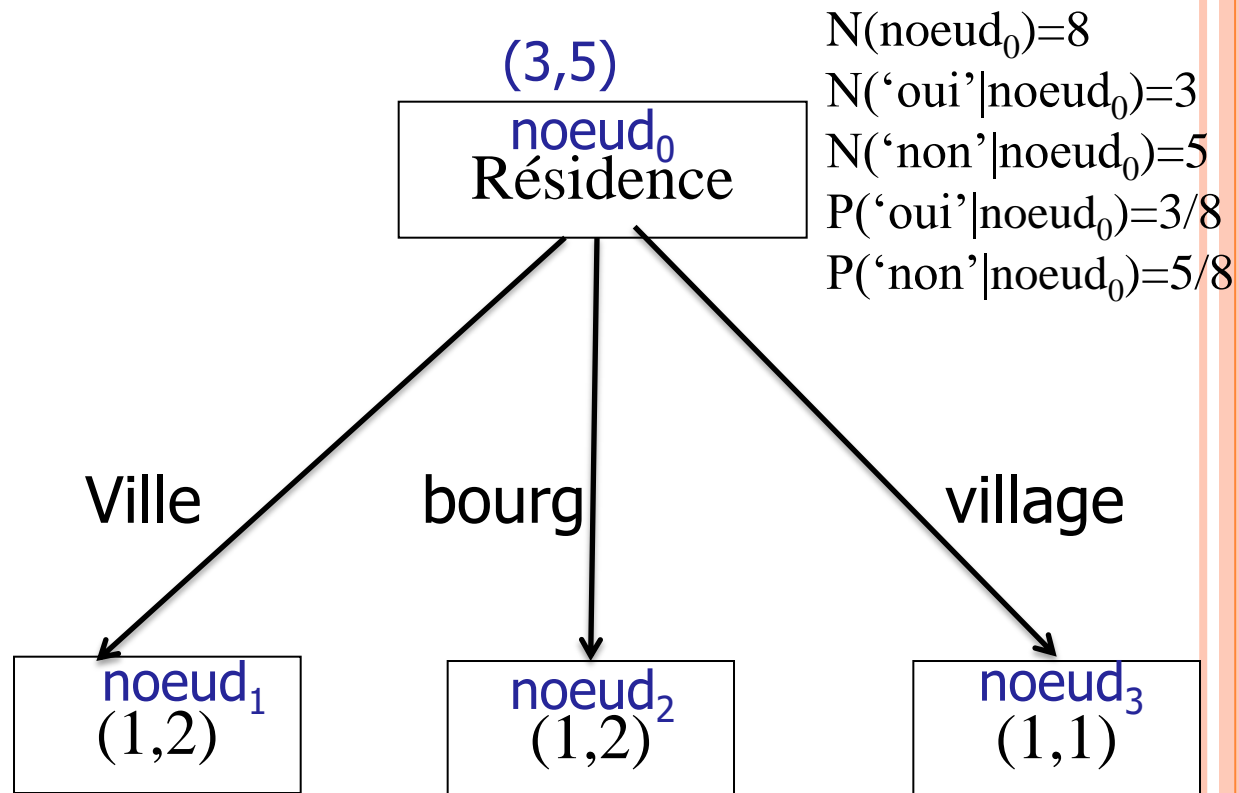
Client	A	I
1	moyen	oui
2	moyen	non
3	âgé	non
4	moyen	oui
5	jeune	oui
6	âgé	non
7	âgé	non
8	moyen	non



Construction de l'arbre de décision : Exemple

3. Construction selon la variable Résidence (R)

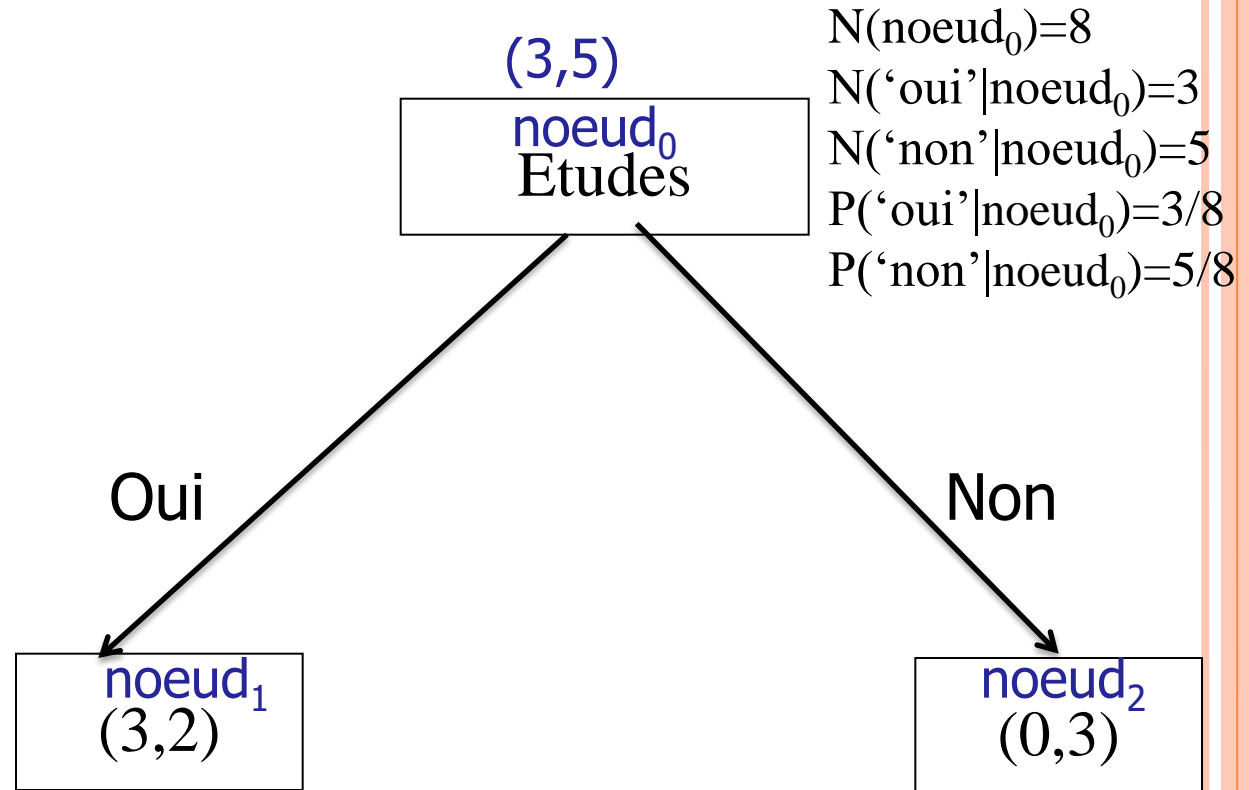
Client	R	I
1	village	oui
2	bourg	non
3	bourg	non
4	bourg	oui
5	ville	oui
6	ville	non
7	ville	non
8	village	non



Construction de l'arbre de décision : Exemple

4. Construction selon la variable Etudes (E)

Client	E	I
1	oui	oui
2	non	non
3	non	non
4	oui	oui
5	oui	oui
6	oui	non
7	oui	non
8	non	non



Construction de l'arbre de décision : Exemple

Quel test (*attribut*) choisir ?

Variable test	Composition noeuds
Montant (M)	(1,2),(2,1),(0,2)
Age (A)	(1,0),(2,2),(0,3)
Résidence (R)	(1,2),(1,2),(1,1)
Etudes (E)	(3,2),(0,3)

Un test intéressant s'il permet une bonne discrimination

- Sur R, aucune discrimination sur aucune branche) On ne gagne rien avec ce test !
- Sur A, deux nœuds sur trois sont “*purs*” !

Comment tout écrire mathématiquement ?

- On a besoin de comparer les différents choix possibles.

Construction de l'arbre de décision

Le but d'un algorithme de construction d'un arbre de décision est de créer un ensemble de **nœuds feuilles qui soient les plus pures possible** avec des **arcs les plus courtes et les moins nombreuses possibles.**

- ❖ Utiliser en premier les attributs les plus importants c'est -à -dire ceux qui découpent de manière la plus nette l'échantillon d'apprentissage au regard de la variable cible.

Construction de l'arbre de décision

Puisqu'on cherche à construire un arbre de décision le plus petit possible rendant compte au mieux des données.

- **La sélection d'un test à associer à un nœud est délicate.**
- Il nous faut un test qui partage les exemples en sous-ensembles homogènes par rapport à une classe donnée.
- Le test parfait **diviserait** les données en sous-ensembles pures, appartenant tous à la même classe.



Construction de l'arbre de décision

Problèmes fondamentaux pour construire l'arbre

- 1) Choix de l'attribut discriminant.
- 2) Affectation d'un label à une feuille.
- 3) Arrêt de la procédure de segmentation (i.e. profondeur de l'arbre). Si un arbre est trop profond, il est trop complexe et trop adapté à l'ensemble d'apprentissage, i.e. pas assez généraliste.
- 4) Choix des bornes de discrétisation (i.e. comment découper les valeurs d'un attribut continu).

Mesures de sélection d'attributs :

Différentes mesures introduites(basées sur la théorie de l'information)

Ces mesures permettent de mesurer le degré de **désordre** dans les différentes classes (pureté d'un noeud)

Propriétés des mesures (fonctions) (degré de mélange ou désordre) :

- Le **minimum** est atteint lorsque tous les nœuds sont purs : tous les exemples ou individus sont dans une même classe.
- Le **maximum** est atteint lorsque les individus sont équirépartis entre les classes.

Exemples de fonctions de mesure :

- Indice de Gini
- Entropie

Mesures de sélection d'attributs :

❖ **L'entropie de Shanon (Algorithme : ID3/C4.5)**

- Suppose des attributs nominaux (discrets)
- Peut-être étendu à des attributs continus

❖ L'entropie d'un nœud p (attribut p) est :

$$\textit{Entropie}(p) = - \sum_{k=1}^C P(k|p) \log(k|p)$$

C le nombre de cardinalité de la variable prédictive (les classes)

$P(k|p) = N(k|p)/N(p)$: proportion des individus appartenant à la classe k parmi ceux du **nœud p** (de la position p)

La variable explicative est souvent binaire (deux valeurs, deux classes). Le log utilisée est log à base 2.

Mesures de sélection d'attributs :

❖ Le critère de Gini (Algorithme : CART)

- Suppose des attributs continus
- Suppose plusieurs valeurs de division pour chaque attribut
- Peut-être étendu pour des attributs nominaux

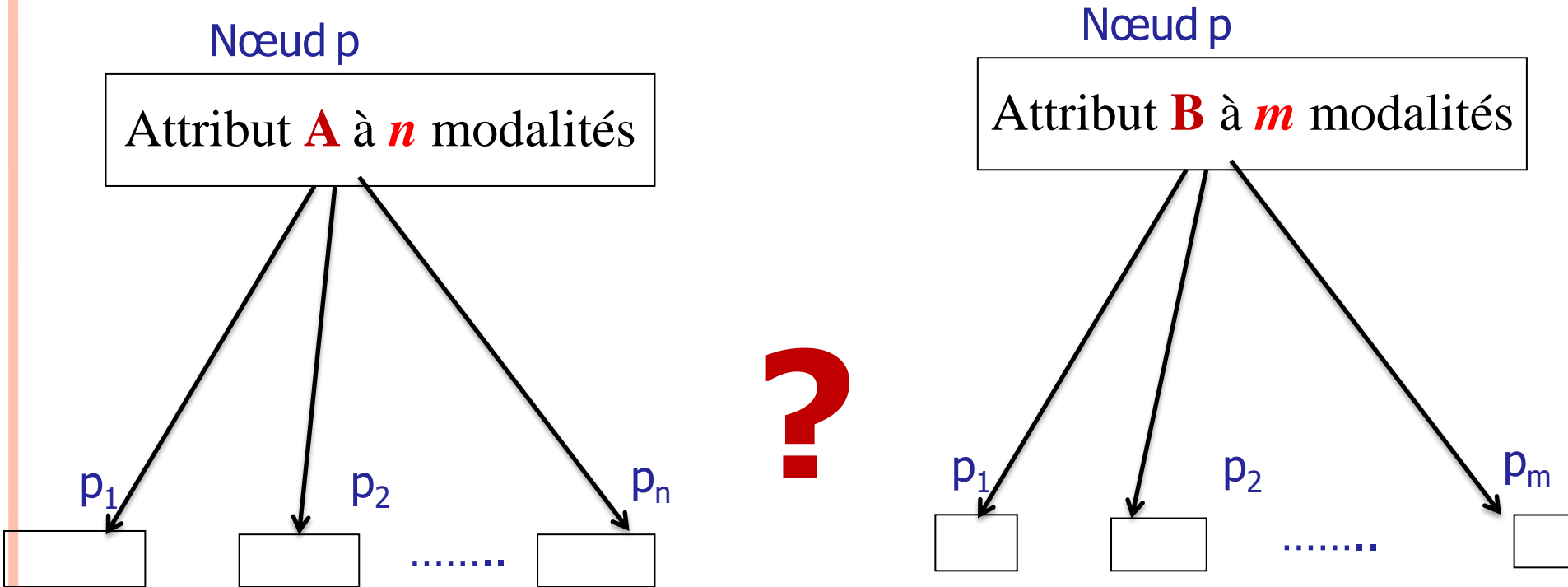
❖ L'entropie d'un nœud p (attribut p) est :

$$Gini(p) = 1 - \sum_{k=1}^C (P(k|p))^2$$

$P(k|p) = N(k|p)/N(p)$: proportion des individus appartenant à la classe k parmi ceux du **nœud p** (de la position p)

Mesures de sélection d'attributs :

Gain d'information

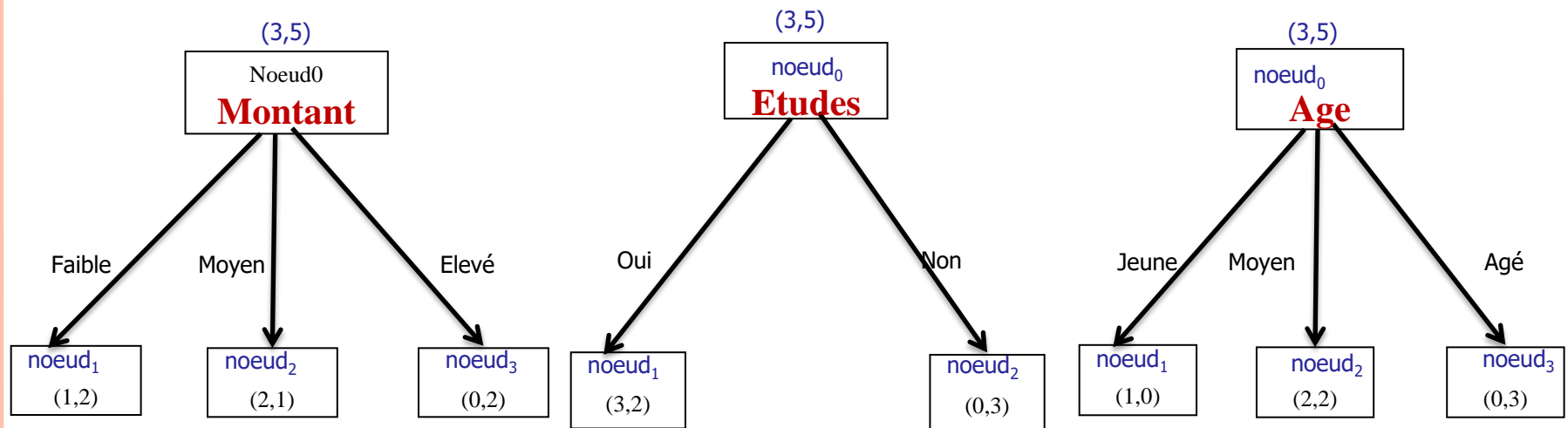


Quel est l'attribut qui permet la meilleure scission A ou B?

Mesures de sélection d'attributs :

Gain d'information

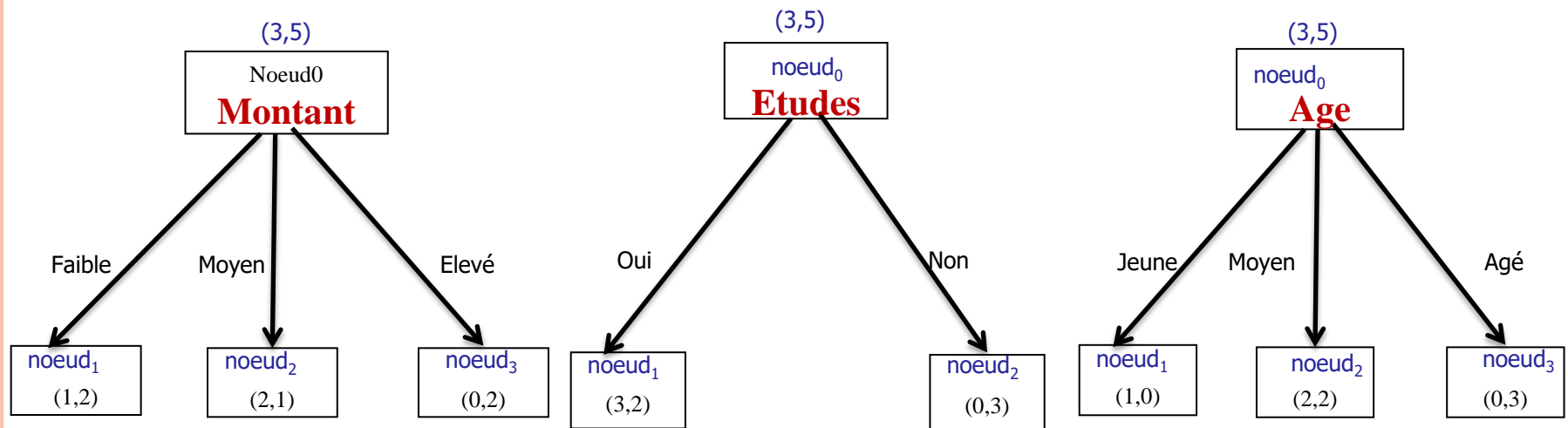
Quel est l'attribut qui permet la meilleure scission du nœud0?



Mesures de sélection d'attributs :

Gain d'information

L'attribut qui Maximise le gain d'information

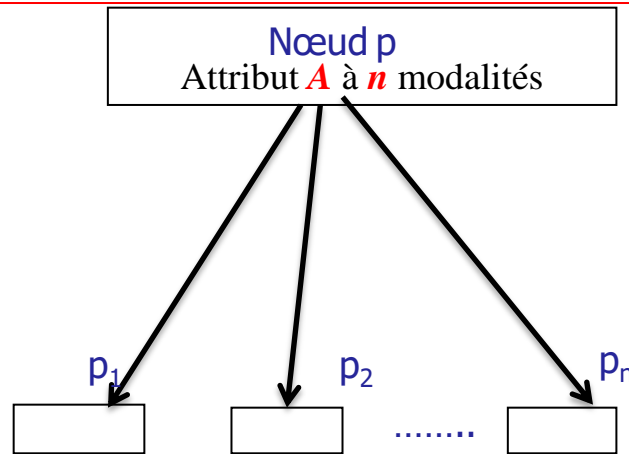


Mesures de sélection d'attributs :

Gain d'information

La fonction de gain:

Le gain de la scission d'un nœud **p** par rapport à un Attribut **A** (ayant **n** modalités):



Le gain de la scission d'un nœud **p** par rapport à un Attribut **A** (ayant **n** modalités)

$$\text{impureté nœud } p(\text{avant séparation}) - [P_{\text{fils1}} * \text{impureté-neud}(\text{fils1}) \\ + \dots + P_{\text{filsn}} * \text{impureté_noeudn}(\text{filsn})]$$

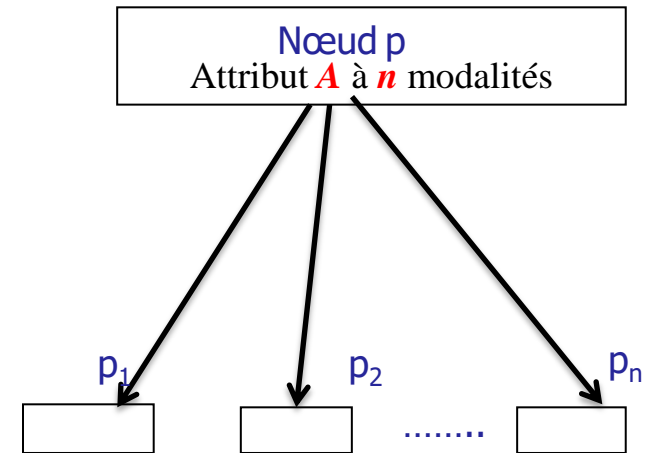
Mesures de sélection d'attributs :

Gain d'information

La fonction de gain:

Le gain de la scission d'un nœud **p** par rapport à un Attribut **A** (ayant **n** modalités):

$$Gain(p, A) = i(p) - \sum_{j=1}^n P_j \cdot i(p_j)$$



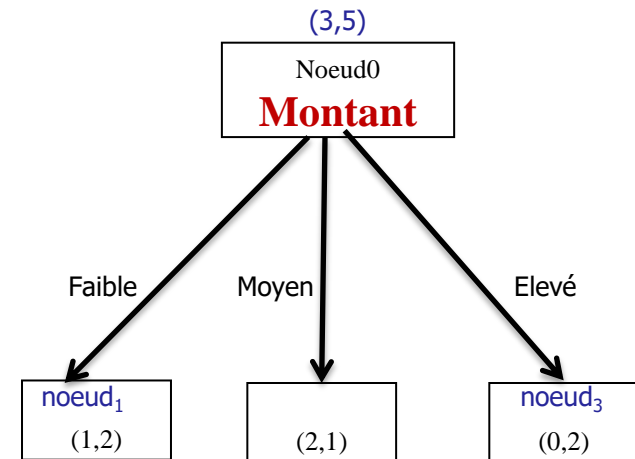
- $i(p)$: fonction d'impureté (entropie, gini)
- n le nombre de modalités de la variable A (attribut de test)
- p_j : la proportion des individus du nœud (p) qui vont en position p_j (nœud p_j) selon la valeur de l'attribut A.

Mesures de sélection d'attributs :

Gain d'information

La fonction de gain:

Le gain de la scission d'un nœud **p** par rapport à un Attribut **A** (ayant **n** modalités):



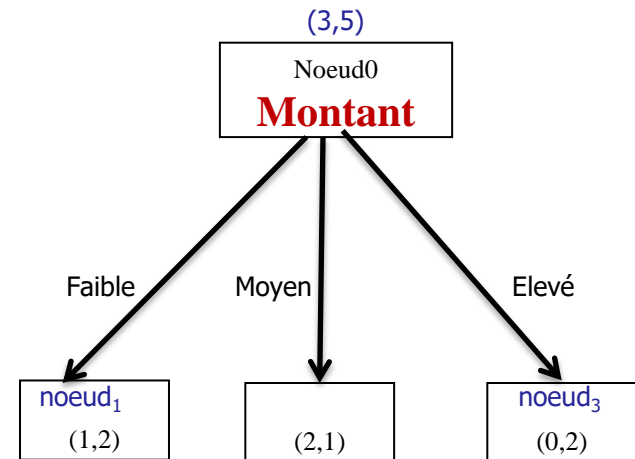
$$\begin{aligned} \text{Gain}(\text{noeud0}, \text{Montant}) = & i(\text{noeud0}) - (\\ & + \text{proportion}(\text{montant} = \text{faible})i(\text{noeud1}) \\ & + \text{proportion}(\text{montant} = \text{moyen})i(\text{noeud2}) \\ & + \text{proportion}(\text{montant} = \text{Elevé})i(\text{noeud3})) \end{aligned}$$

Mesures de sélection d'attributs :

Gain d'information

La fonction de gain:

Le gain de la scission d'un nœud **p** par rapport à un Attribut **A** (ayant **n** modalités):



$$\begin{aligned} \text{Gain}(\text{noeud0}, \text{Montant}) = & i(\text{noeud0}) - \left(\right. \\ & \frac{3}{8} * i(\text{noeud1}) \\ & + \frac{3}{8} * i(\text{noeud2}) \\ & \left. + \frac{2}{8} i(\text{noeud3}) \right) \end{aligned}$$

Construction de l'arbre de décision : Exemple

Gain d'information

- On cherche le *test* (attribut) qui maximise le gain

$$Gain(p, A) = i(p) - \sum_{j=1}^n P_j \cdot i(p_j)$$

Le nœud initial (3, 5)



Nous allons choisir la fonction **Entropie**

$$Entropie(p) = - \sum_{k=1}^c P(k|p) \log(k|p)$$

Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

$$Entropie(noeud_0) = -(3/8) \log(3/8) - (5/8) \log(5/8) = 0,95$$

Construction de l'arbre de décision : Exemple

Gain d'information

Tester sur la variable Montant (M)

$$Gain(p, t) = i(p) - \sum_{j=1}^n P_j \cdot i(p_j)$$

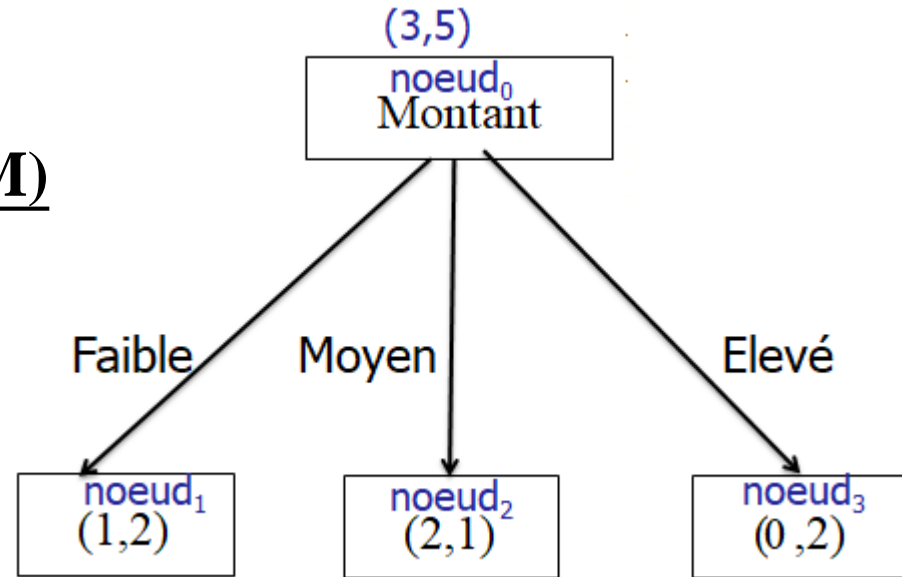
Gain(noeud₀, Montant) =

$$\begin{aligned} &= \text{Entropie}(\text{noeud}_0) - (P_{\text{faible}} \cdot \text{Entropie}(\text{noeud}_1) + P_{\text{moyen}} \cdot \text{Entropie}(\text{noeud}_2) + P_{\text{élevé}} \cdot \text{Entropie}(\text{noeud}_3)) \\ &= \text{Entropie}(\text{noeud}_0) - (3/8 \cdot \text{Entropie}(\text{noeud}_1) + 3/8 \cdot \text{Entropie}(\text{noeud}_2) + 2/8 \cdot \text{Entropie}(\text{noeud}_3)) \end{aligned}$$

- Entropie(noeud₁) = - (1/3 log(1/3) + 2/3 log(2/3)) = 0,92
- Entropie(noeud₂) = - (2/3 log(2/3) + 1/3 log(1/3)) = 0,92
- Entropie(noeud₃) = - (2/2 log(2/2)) = 0 **Un nœud pur**

Gain(noeud₀, Montant) =

$$\begin{aligned} &= \text{Entropie}(\text{noeud}_0) - (3/8 \cdot \text{Entropie}(\text{noeud}_1) + 3/8 \cdot \text{Entropie}(\text{noeud}_2) + 2/8 \cdot \text{Entropie}(\text{noeud}_3)) \\ &= \text{Entropie}(\text{noeud}_0) - (3/8 \cdot 0,92 + 3/8 \cdot 0,92 + 2/8 \cdot 0) \\ &= \text{Entropie}(\text{noeud}_0) - 0,69 = 0,95 - 0,69 = \mathbf{0,26} \end{aligned}$$

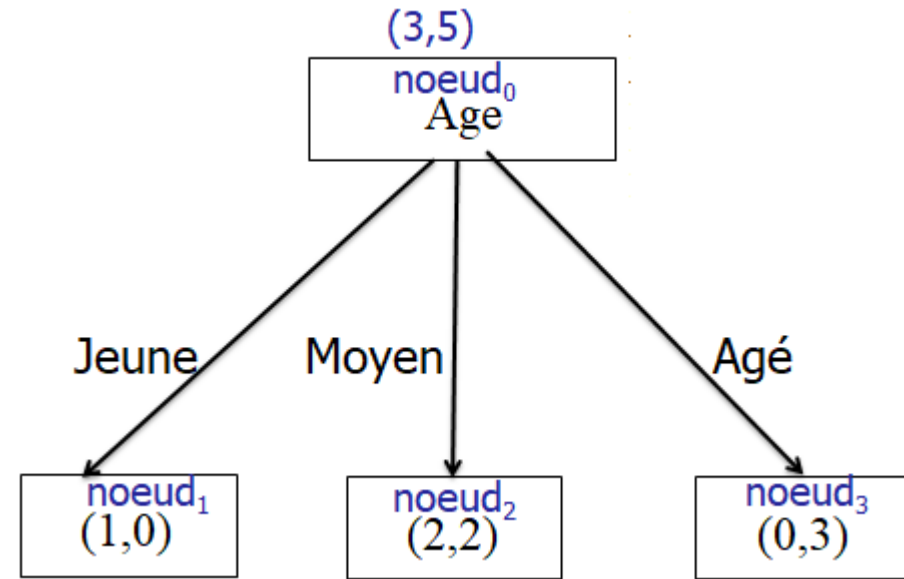


Construction de l'arbre de décision : Exemple

Gain d'information

Tester sur la variable Age (A)

$$Gain(p, t) = i(p) - \sum_{j=1}^n P_j \cdot i(p_j)$$



Gain(noeud₀,Age)=

$$\begin{aligned} &= \text{Entropie}(\text{noeud}_0) - (P_{\text{jeune}} \cdot \text{Entropie}(\text{noeud}_1) + P_{\text{moyen}} \cdot \text{Entropie}(\text{noeud}_2) + P_{\text{agé}} \cdot \text{Entropie}(\text{noeud}_3)) \\ &= \text{Entropie}(\text{noeud}_0) - (1/8 \cdot \text{Entropie}(\text{noeud}_1) + 4/8 \cdot \text{Entropie}(\text{noeud}_2) + 3/8 \cdot \text{Entropie}(\text{noeud}_3)) \end{aligned}$$

- Entropie(noeud₁) = - (1/1 log(1/1)) = 0 **Un nœud pur**
- Entropie(noeud₂) = - (2/4 log(2/4) + 2/4 log(2/2)) = 1
- Entropie(noeud₃) = - (2/2 log(2/2)) = 0 **Un nœud pur**

Gain(noeud₀,Age)=

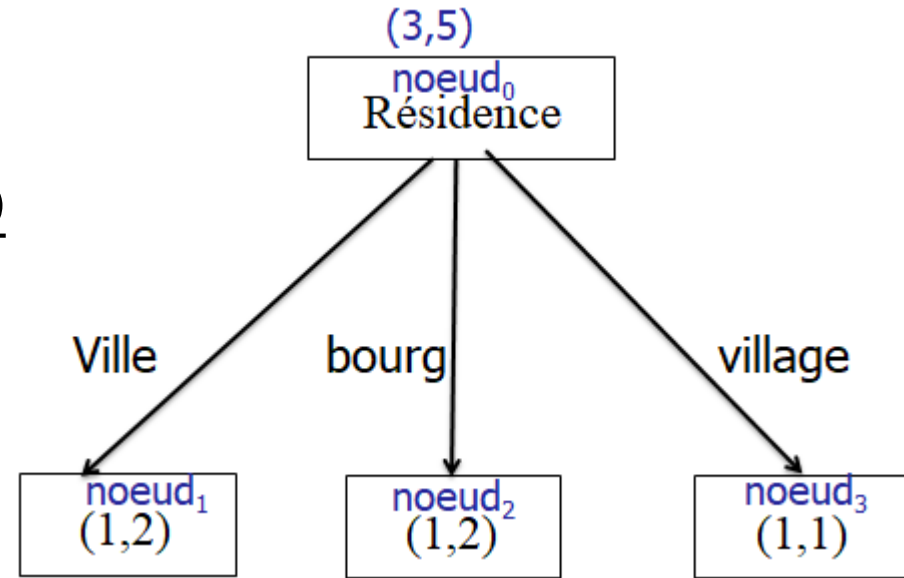
$$\begin{aligned} &= \text{Entropie}(\text{noeud}_0) - (1/8 \cdot \text{Entropie}(\text{noeud}_1) + 4/8 \cdot \text{Entropie}(\text{noeud}_2) + 3/8 \cdot \text{Entropie}(\text{noeud}_3)) \\ &= \text{Entropie}(\text{noeud}_0) - (1/8 \cdot 0 + (4/8) \cdot 1 + 3/8 \cdot 0) \\ &= \text{Entropie}(\text{noeud}_0) - 0,5 = 0,95 - 0,5 = \mathbf{0,45} \end{aligned}$$

Construction de l'arbre de décision : Exemple

Gain d'information

Tester sur la variable Résidence (R)

$$\text{Gain}(p, t) = i(p) - \sum_{j=1}^n P_j \cdot i(p_j)$$



Gain(noeud₀, Résidence)=

$$\begin{aligned} &= \text{Entropie}(\text{noeud}_0) - (P_{\text{ville}} \cdot \text{Entropie}(\text{noeud}_1) + P_{\text{bourg}} \cdot \text{Entropie}(\text{noeud}_2) + P_{\text{village}} \cdot \text{Entropie}(\text{noeud}_3)) \\ &= \text{Entropie}(\text{noeud}_0) - (3/8 \cdot \text{Entropie}(\text{noeud}_1) + 3/8 \cdot \text{Entropie}(\text{noeud}_2) + 2/8 \cdot \text{Entropie}(\text{noeud}_3)) \end{aligned}$$

- Entropie(noeud₁) = - (1/3 log(1/3) + 2/3 log(2/3)) = 0,92
- Entropie(noeud₂) = - (1/3 log(1/3) + 2/3 log(2/3)) = 0,92
- Entropie(noeud₃) = - (1/2 log(1/2) + 1/2 log(1/2)) = 1

Gain(noeud₀, Résidence)=

$$\begin{aligned} &= \text{Entropie}(\text{noeud}_0) - (3/8 \cdot \text{Entropie}(\text{noeud}_1) + 3/8 \cdot \text{Entropie}(\text{noeud}_2) + 2/8 \cdot \text{Entropie}(\text{noeud}_3)) \\ &= \text{Entropie}(\text{noeud}_0) - (3/8 \cdot 0,92 + 3/8 \cdot 0,92 + 2/8 \cdot 1) \\ &= \text{Entropie}(\text{noeud}_0) - 0,94 = 0,95 - 0,94 = \mathbf{0,01} \end{aligned}$$

Construction de l'arbre de décision : Exemple

Gain d'information

Tester sur la variable Etudes (E)

$$Gain(p, t) = i(p) - \sum_{j=1}^n P_j \cdot i(p_j)$$

Gain(noeud₀, Etudes)=

= Entropie(noeud₀) - (P_{oui} · Entropie(noeud₁) + P_{non} · Entropie(noeud₂))

= Entropie(noeud₀) - (5/8 · Entropie(noeud₁) + 3/8 · Entropie(noeud₂))

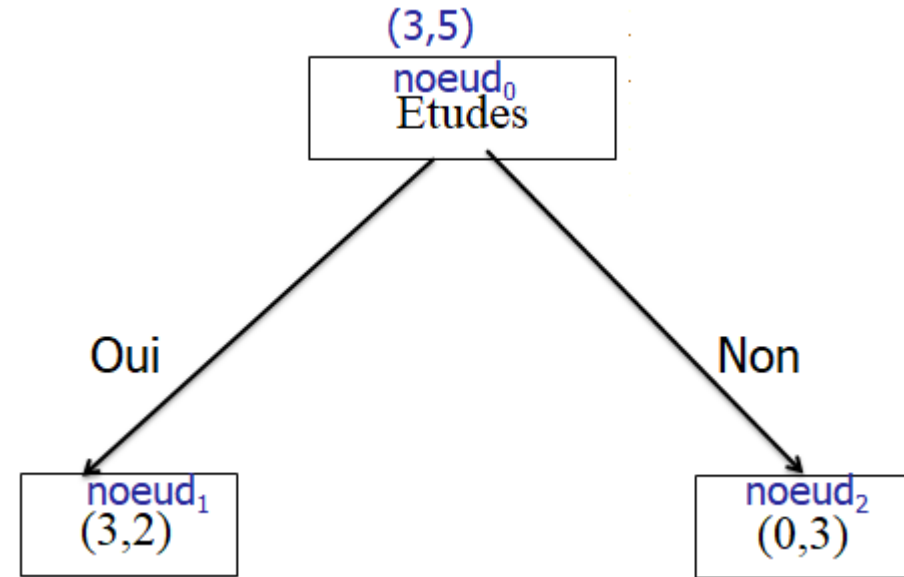
- Entropie(noeud₁) = - (3/5 log(3/5) + 2/5 log(2/5)) = 0,97
- Entropie(noeud₂) = - (3/3 log(3/3)) = 0 **un nœud pur**

Gain(noeud₀, Etudes)=

= Entropie(noeud₀) - (5/8 · Entropie(noeud₁) + 3/8 · Entropie(noeud₂))

= Entropie(noeud₀) - (5/8 · 0,97 + 3/8 · 0)

= Entropie(noeud₀) - 0,60 = 0,95 - 0,60 = **0,35**



Construction de l'arbre de décision : Exemple

- On cherche le *test* (attribut) qui maximise le gain

On considère le noeud 0 : (3,5)

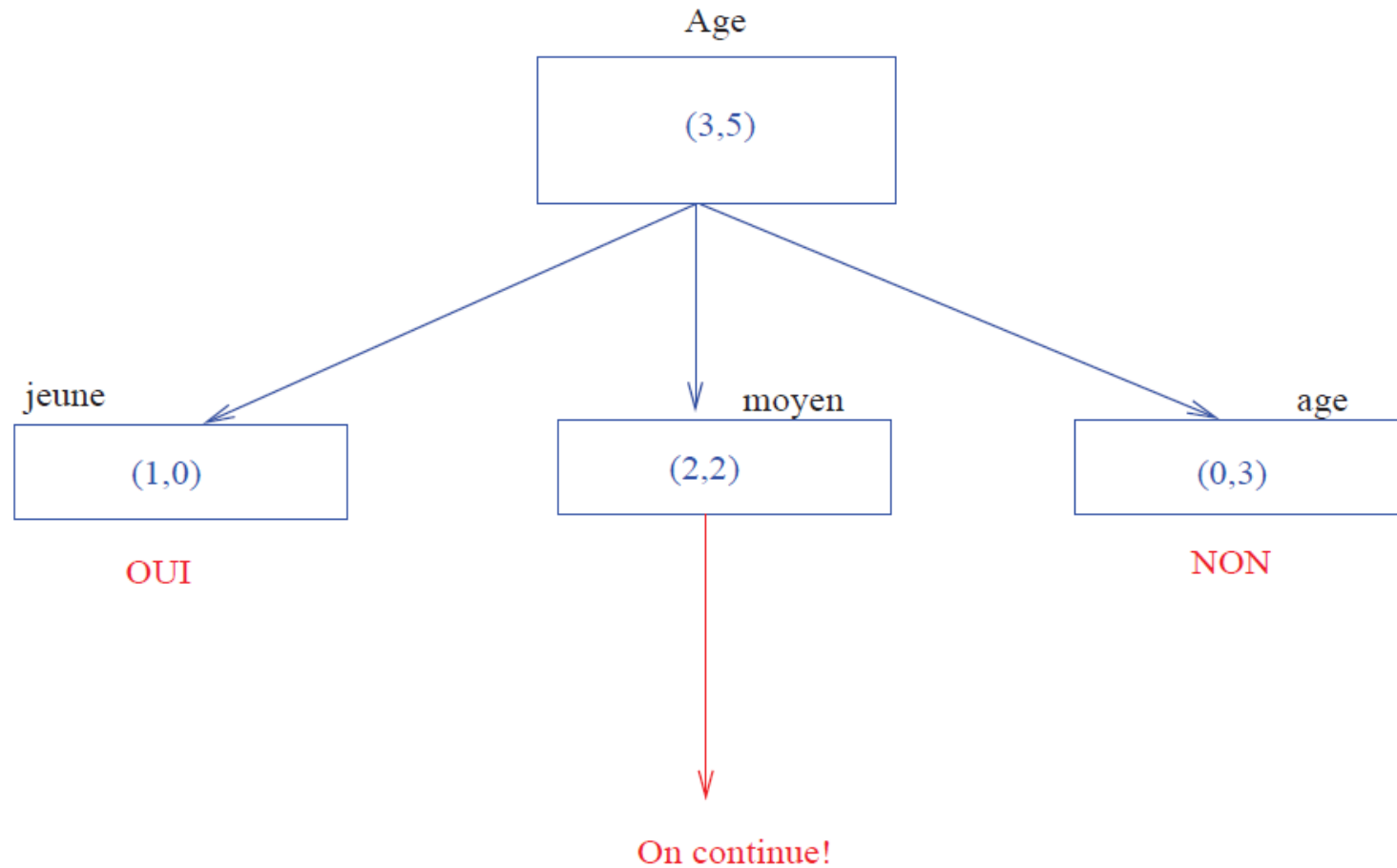
Variable test	Gain
Montant (M)	$\text{Gain}(0, M) = \text{Entropie}(0) - 0,69 = 0,95 - 0,69 = 0,26$
Age (A)	$\text{Gain}(0, A) = \text{Entropie}(0) - \mathbf{0,5} = \mathbf{0,95 - 0,5} = \mathbf{0,45}$
Résidence (R)	$\text{Gain}(0, R) = \text{Entropie}(0) - 0,94 = 0,95 - 0,94 = 0,01$
Etudes (E)	$\text{Gain}(0, R) = \text{Entropie}(0) - 0,60 = 0,95 - 0,60 = 0,35$

Le noeud initial (3, 5)

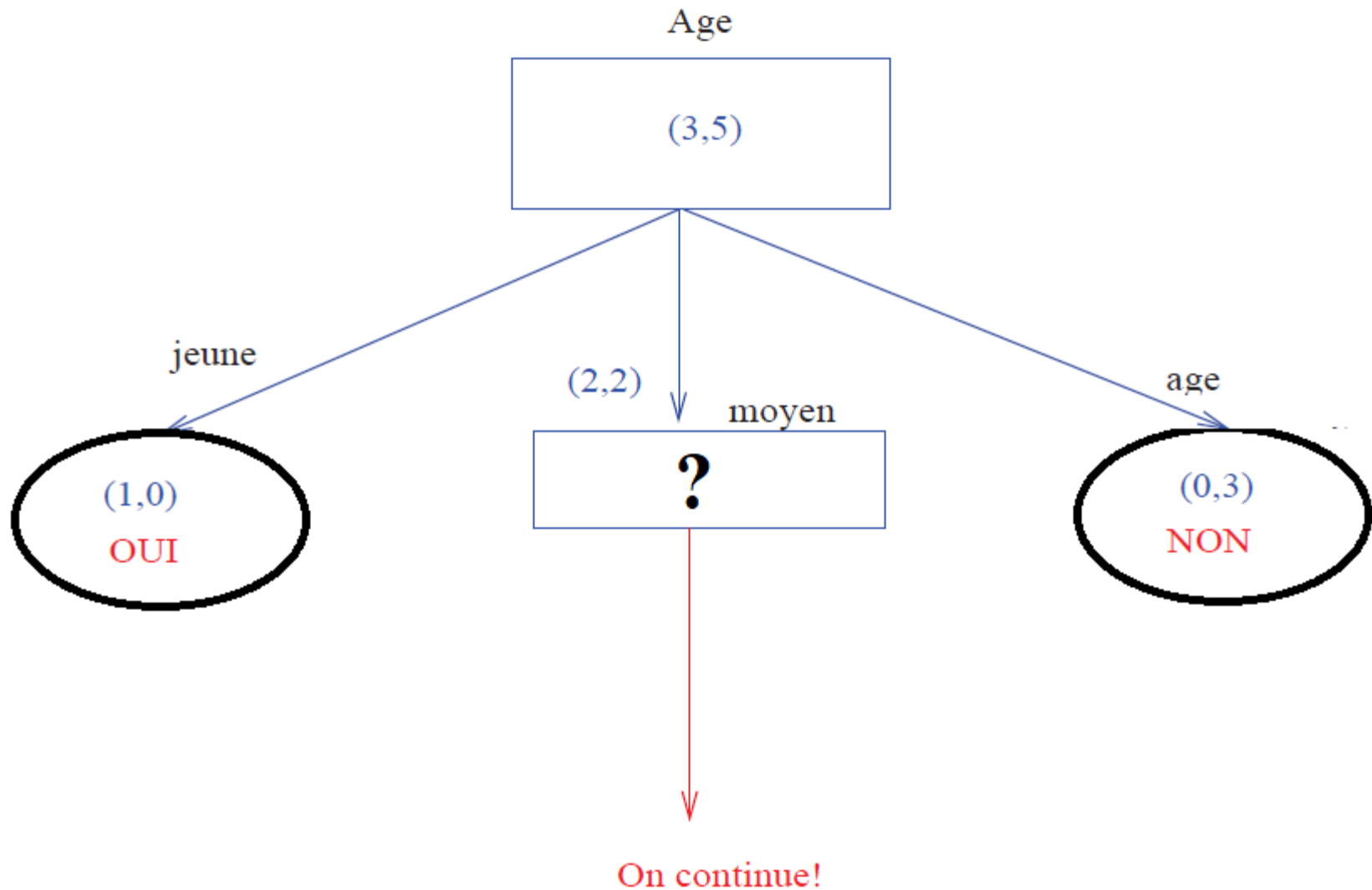
? = A

L'attribut **A** est choisit

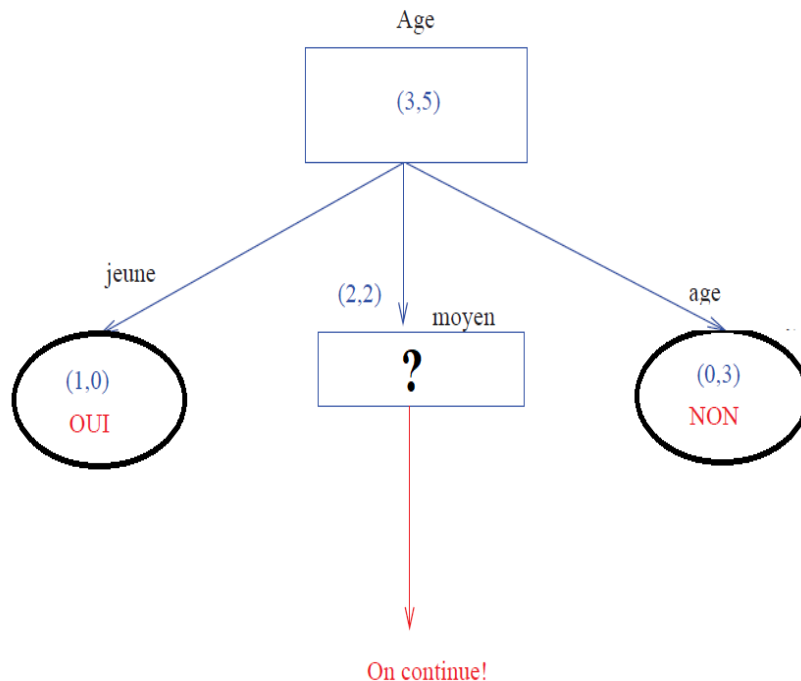
Construction de l'arbre de décision : Exemple



Construction de l'arbre de décision : Exemple



Construction de l'arbre de décision : Exemple

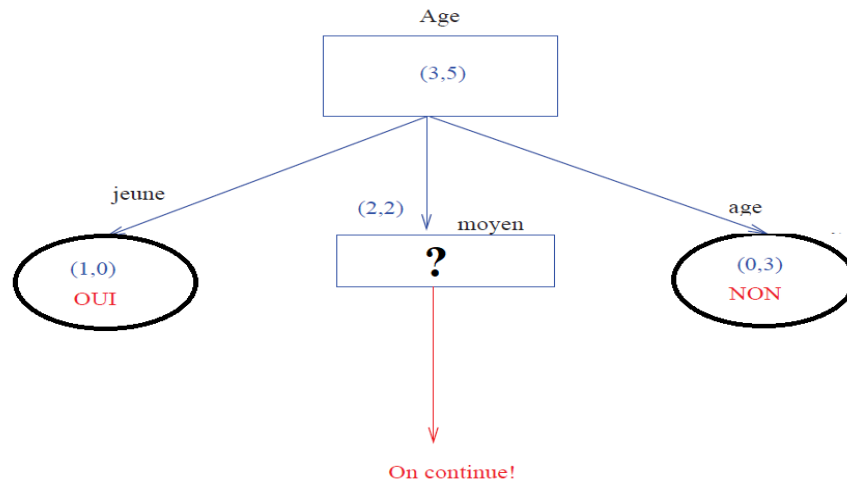


Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

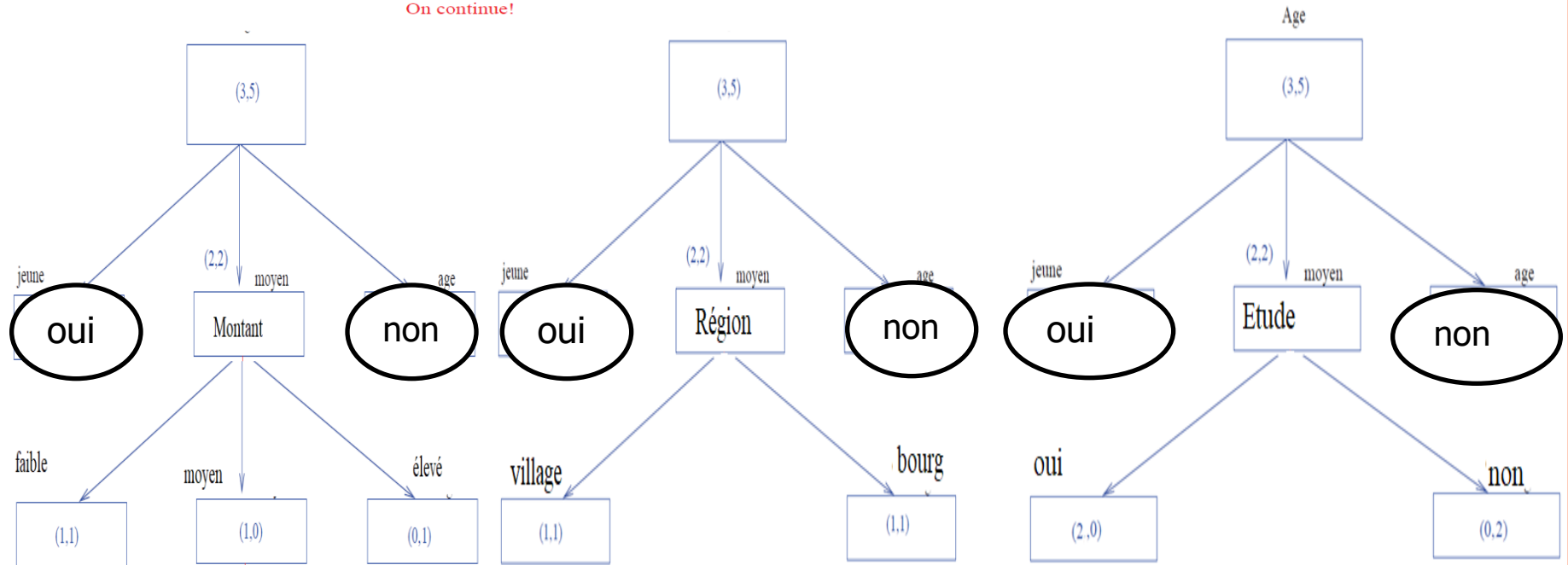


Client	M	R	E	I
1	moyen	village	oui	oui
2	élevé	bourg	non	non
4	faible	bourg	oui	oui
8	faible	village	non	non

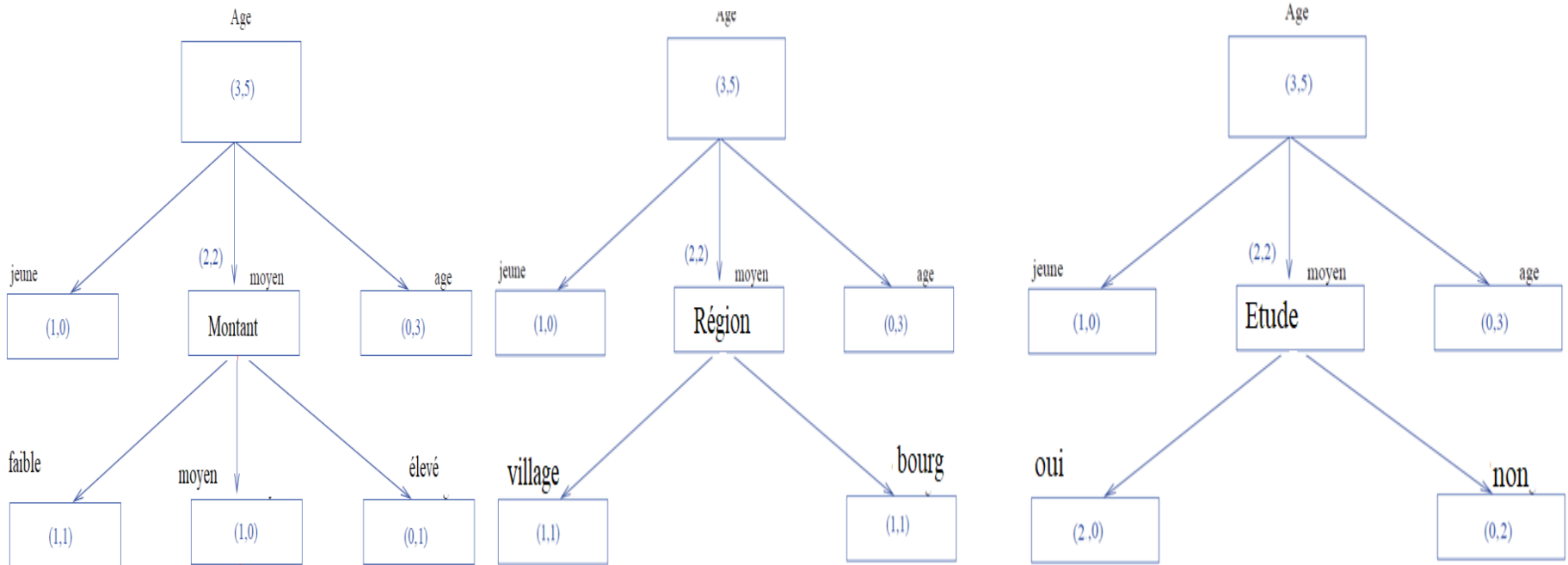
Construction de l'arbre de décision : Exemple



Client	M	R	E	I
1	moyen	village	oui	oui
2	élevé	bourg	non	non
4	faible	bourg	oui	oui
8	faible	village	non	non



Construction de l'arbre de décision : Exemple

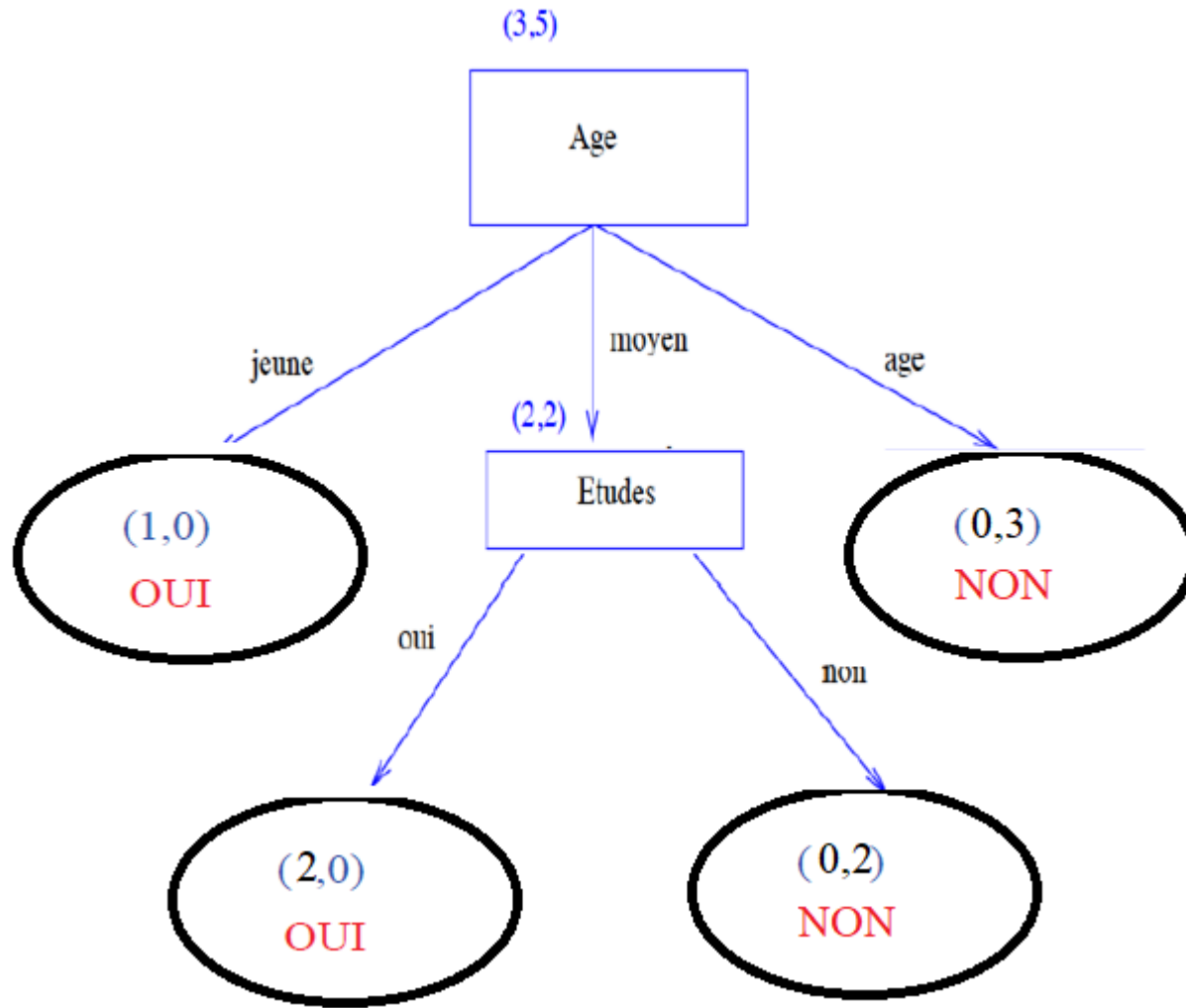


Quel test choisir?

Variable test	Composition noeuds
Montant (M)	(1,1),(1,0),(0,1)
Résidence (R)	(1,1),(1,1)
Etudes (E)	(2,0),(0,2)

Calculer le gain pour chaque test ?

Construction de l'arbre de décision : Exemple



Client	M	A	R	E	I
1	moyen	moyen	village	oui	oui
2	élevé	moyen	bourg	non	non
3	faible	âgé	bourg	non	non
4	faible	moyen	bourg	oui	oui
5	moyen	jeune	ville	oui	oui
6	élevé	âgé	ville	oui	non
7	moyen	âgé	ville	oui	non
8	faible	moyen	village	non	non

Problèmes liés au calcul du gain

Gain d'information et **proportion de gain (ratio-gain)**

Le **gain d'information** présente un désavantage. Il a tendance à préférer les attributs qui ont beaucoup de valeurs possibles, et qui partagent les données en nombreux petits sous-ensembles pures

Pour corriger ce problème, on utilise une autre mesure le rapport de gain (gain ratio)

La proportion de gain (gain d'information proportionnel) est une mesure alternative qui divise le gain d'information par une mesure proportionnelle à la taille de la partition générée par les valeurs d'un attribut (si il y a beaucoup de valeurs le dénominateur sera plus large)

Choisir l'attribut avec le plus grand rapport de gain



La proportion de gain (gain-ratio)

$$\text{GainRatio}(\text{noeud}, \text{Attribut}) = \frac{\text{Gain}(\text{noeud}, \text{Attribut})}{\text{SplitInfo}(\text{noeud}, \text{Attribut})}$$

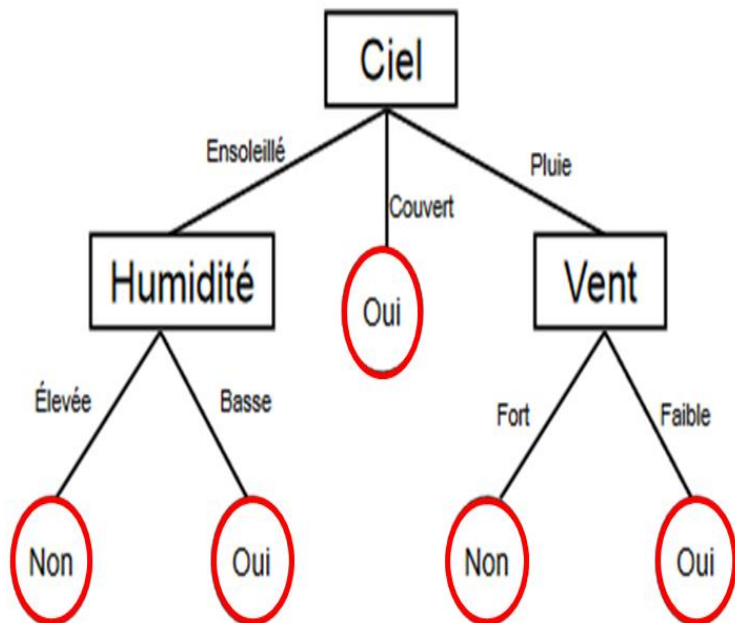
$$\text{SplitInfo}(\text{noeud}, \text{Attribut } A) = \sum_{i=1}^k -\frac{|S_i|}{|S|} \times \log_2 \frac{|S_i|}{|S|}$$

avec :

- S : ensemble d'exemples. S_i : nombre d'exemples valant A_i pour l'attribut A .
- k : nombre de valeurs de l'attribut A

Arbre de décision : exemple

Est-ce que les conditions sont favorable pour jouer au tennis?



Arbre de décision

- Variable cible **Jouer** → 2 classes (**Oui**, **NON**)
- 3 variables explicatives :
 - **Ciel** avec valeurs : Ensoleillé, Couvert, Pluie
 - **Humidité** avec valeurs : Elevée, Basse
 - **Vent** avec valeurs : Faible, Fort

Classification d'une nouvelle donnée(instance)

Décider si le match va avoir lieu (jouer=Oui) ou bien (Jouer=NON)

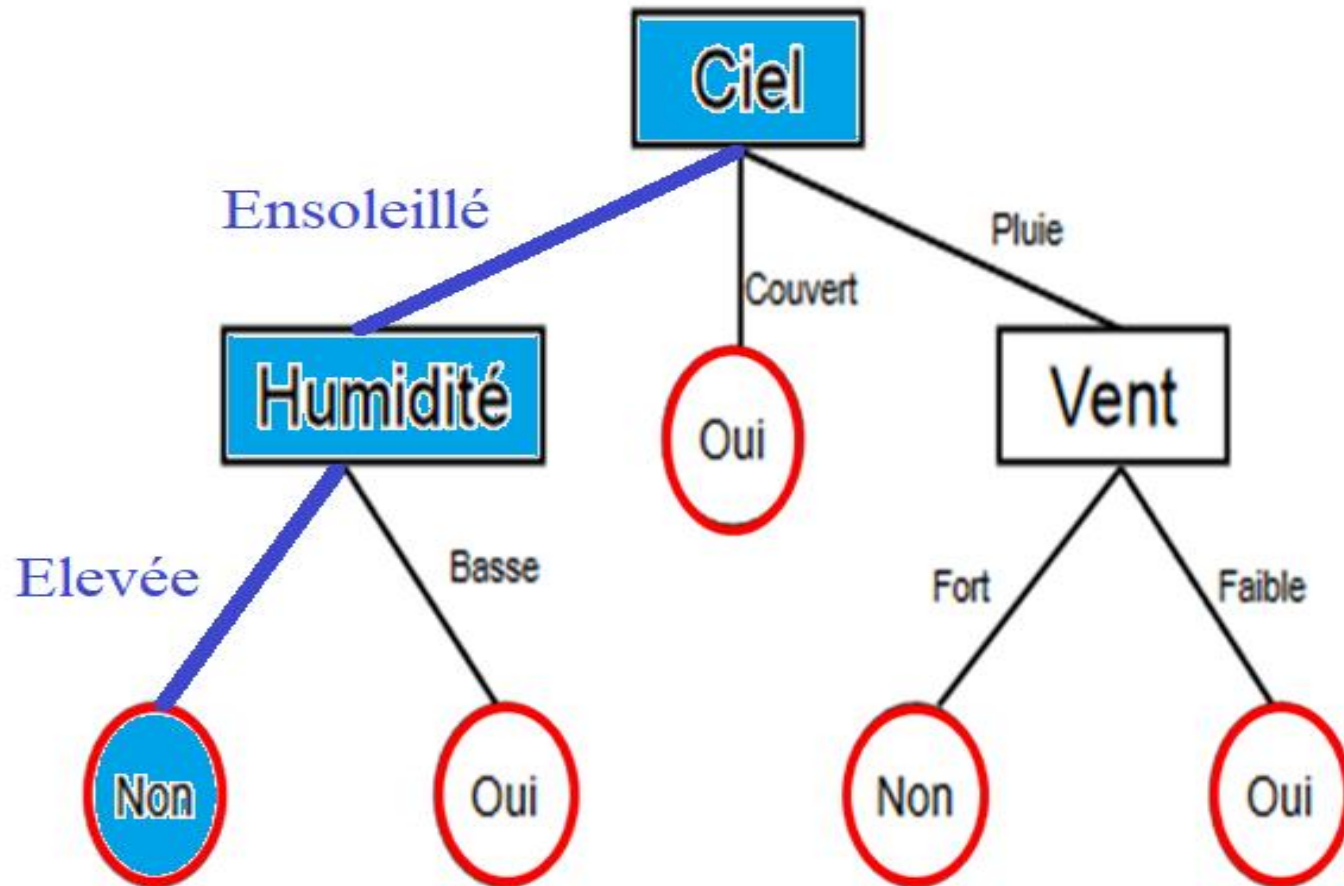
En fonction du : Ciel, Humidité et le Vent

Exemple : Ciel = ensoleillé, Humidité= Elevée, Vent= Fort

Arbre de décision : exemple

Est-ce que les conditions sont favorable pour jouer au tennis?

Ciel = ensoleillé, **Humidité**= Elevée, **Vent**= Fort



Extraction des règles

Représenter sous forme de règles IF-THEN

Une règle est créée pour chaque chemin de la racine vers la feuille

Chaque paire de valeurs d'attributs forme une conjonction

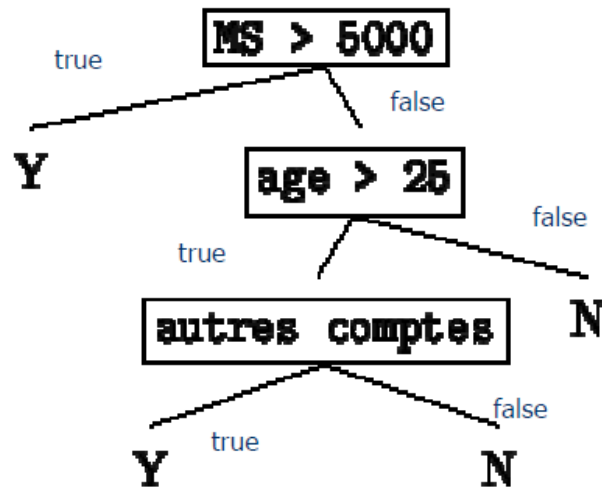
Les feuilles représentent les classes prédites

Les règles sont faciles à comprendre pour les humains



Extraction des règles

- Attribution d'un prêt suivant la moyenne des soldes courants (MS), l'âge et la possession d'autres comptes



If MS > 5000 **then** Pret = Yes

If MS ≤ 5000 **and** age ≤ 25 **then** Pret = No

If MS ≤ 5000 **and** age > 25 **and** autres_comptes = Yes **then** Pret = Yes

If MS ≤ 5000 **and** age > 25 **and** autres_comptes = No **then** Pret = No

Représentation d'une expression par un arbre de décision

Toute formule de la logique propositionnelle peut être représentée par un arbre de décision

La logique propositionnelle est construite à partir de:

- Variables propositionnelles
- D'opérateurs logiques: and, or, not, (implication),(équivalence)



Règles \Rightarrow Arbre de décision

- Exemple:

if X and Y then A

if X and W and V then B

if Y and V then A

Peuvent être représentées par un arbre de décision.

De plus, Les règles peuvent être combinées en:

if Y and (X or V) then A

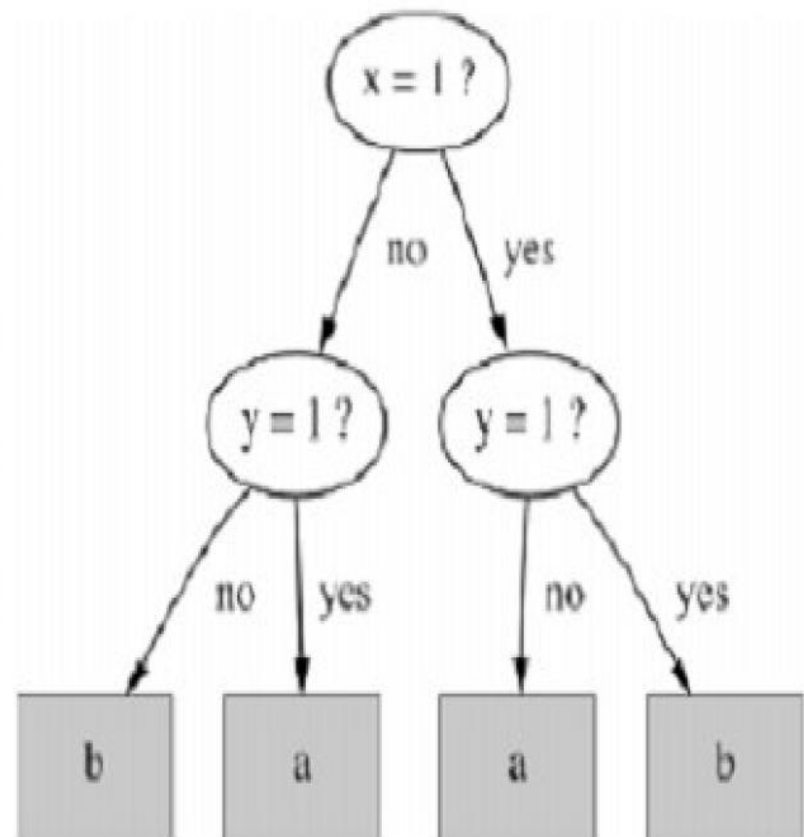
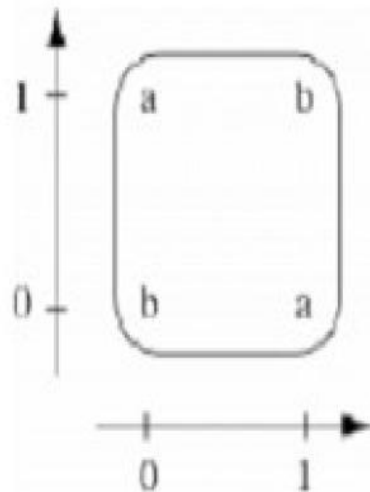
if X and W and V then B

Et on obtient un autre arbre de décision de ces 2 règles

Règles \Rightarrow Arbre de décision

Le ou exclusif (XOR)

```
If x = 1 and y = 0  
  then class = a  
If x = 0 and y = 1  
  then class = a  
If x = 0 and y = 0  
  then class = b  
If x = 1 and y = 1  
  then class = b
```



Construction de l'arbre de décision

L'idéal est de trouver un critère permettant d'arrêter la croissance de l'arbre au bon moment.

Le risque d'arrêter trop tôt la croissance de l'arbre est plus important que de l'arrêter trop tard.

Les méthodes utilisées procèdent souvent en deux phases.

La première phase correspond à l'algorithme de génération d'arbre de décision

La seconde phase, **Élagage** de l'arbre obtenu pour essayer de faire diminuer l'erreur réelle (élaguer un arbre consiste à en supprimer certains sous-arbres).

Les méthodes se distinguent les unes des autres par les choix des opérateurs, mais aussi par les méthodes d'élagage utilisées



Élagage d'un arbre de décision

Contrôler la complexité du nombre des branches et des feuilles pour réaliser un arbre de décision.

Minimiser la taille de l'arbre.

Trouver le nombre optimale de nœuds.

Minimiser le taux d'erreur réel de classification par l'arbre de décision.

Éviter le problème du sur-apprentissage (taux d'erreur d'apprentissage est très faible, taux d'erreur de validation est très élevé)

Deux techniques d'élagage

Pré-élagage

Post-élagage



Pré-élagage

Pré-élagage : l'élagage est réalisé pendant de la construction d'arbre de décision.

Arrêter de diviser un nœud quand la pureté des points qui domine est non parfaite mais suffisante.

Arrêter quand il y a une classe majoritaire dans le nœud.

Utiliser un seuil pour détecter une classe dominantes.



Post élagage

Post-élagage : se déroule après la construction d'arbre de décision.

Finir la construction de l'arbre.

Simplifier l'arbre en remontant des feuilles vers la racine pour trouver ou élaguer.

Utiliser des critères de qualité qui mesure un compromis : l'erreur obtenue et la complexité de l'arbre.

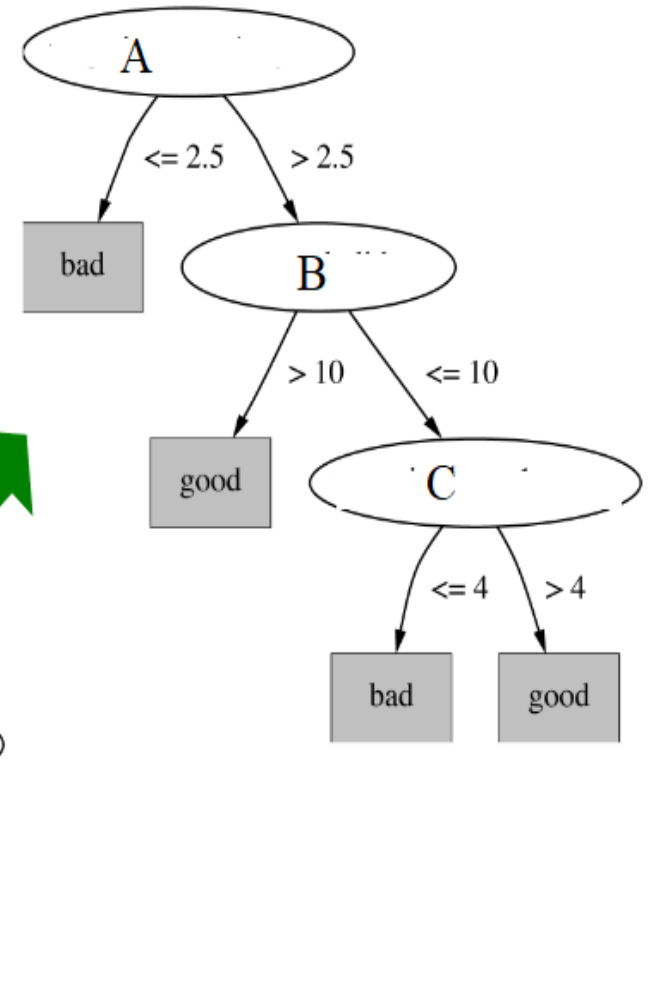
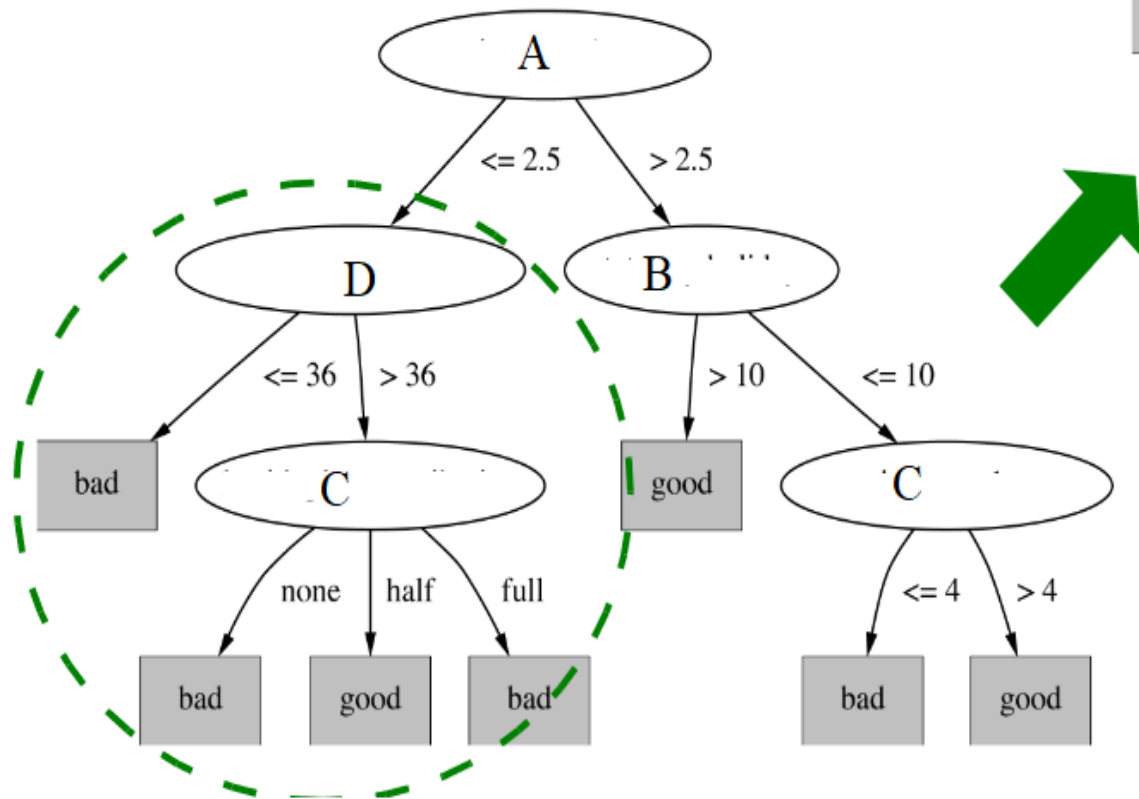
Utiliser un ensemble de validation pour mesurer l'erreur à chaque nœuds.



Principe du post-élagage

- Construire l'arbre complet.
- Pour chaque nœud interne, regarder s'il ne serait pas meilleur de le remplacer :
 - Par une feuille.
 - Par un de ses fils (son fils le plus fréquent).

Élagage d'un arbre de décision



Post-élagage de l'algorithme CART

- Consiste à laisser se terminer l'algorithme de construction, puis à construire une série d'arbres plus simples par regroupement des nœuds et enfin, de choisir le meilleur d'entre eux avec un ensemble de validation.
- L'ensemble d'individus est coupé en deux parties : l'une pour construire T_{max} (ensemble d'apprentissage) et proposer des élagages, l'autre (ensemble de validation) pour choisir le meilleur parmi les élagages proposés.
- Le problème est donc de remplacer certains des noeuds de T_{max} par des feuilles. La classe affectée à une feuille f ainsi créée sera ensuite choisie comme celle qui est majoritaire dans les feuilles de T_{max} dominées par le noeud que l'on vient de remplacer par f .



Post-élagage de l'algorithme CART

- L'algorithme optimal consisterait à calculer le taux d'erreur de l'ensemble de validation sur tous les arbres qu'il est possible d'obtenir par élagage de T_{max} .
- On utilise donc des solutions sous-optimales, dont la plus classique consiste à construire une séquence d'arbres par élagages successifs :
$$S = \{T_{max}, T_1, \dots, T_k, \dots, T_m\},$$
 où T_m est constitué d'une seule feuille comprenant les m individus d'apprentissage.
- Pour passer de T_k à T_{k+1} , il faut transformer en feuilles un ou plusieurs nœuds dans T_k .



Post-élagage de l'algorithme CART

La technique généralement utilisée est de choisir les noeuds qui minimisent sur l'ensemble des noeuds de T_k le critère suivant :

$$\omega(T_k, d) = \frac{MC(d, k) - MCT(d, k)}{n(k) \cdot (nt(d, k) - 1)}$$

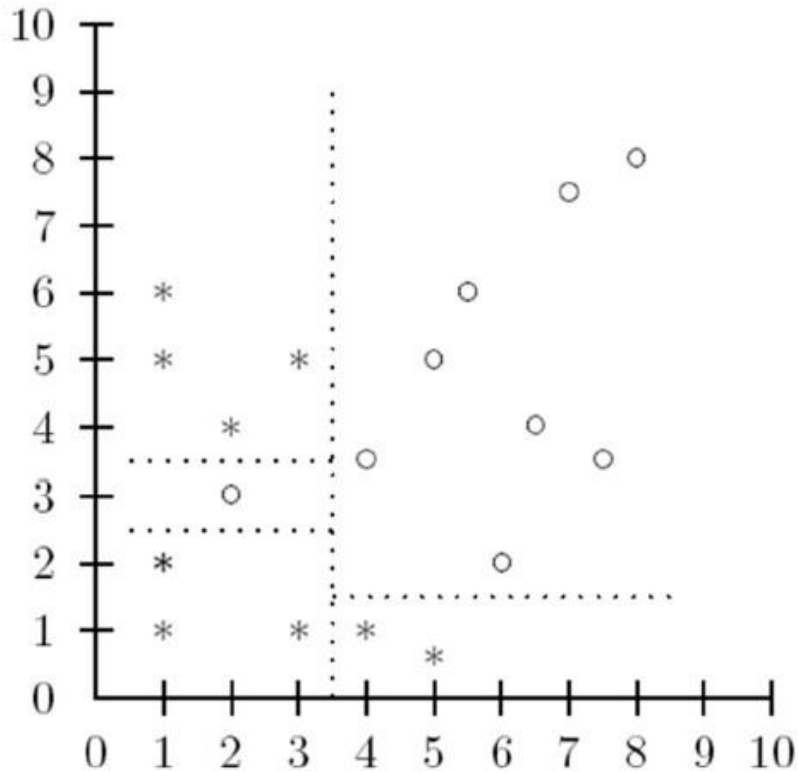
- $MC(d, k)$ est le nombre d'individus de l'ensemble d'apprentissage mal classés par le noeud d de T_k quand on fait l'hypothèse qu'il a été transformé en feuille.
- $MCT(d, k)$ le nombre d'individus de l'ensemble d'apprentissage mal classés par les feuilles de T_k situées sous le noeud d .
- $n(k)$ le nombre de feuilles de T_k .
- $nt(d, k)$ le nombre de feuilles du sous-arbre de T_k situé sous le noeud d .

Post-élagage de l'algorithme CART

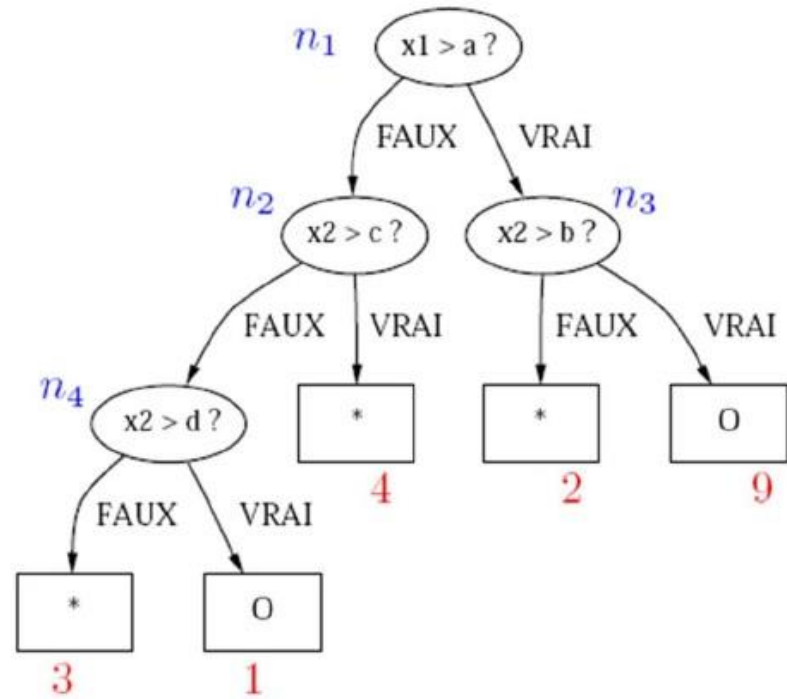
- Finalement, la suite $S = \{T_{max}, T_1, \dots, T_k, \dots, T_m\}$ que l'on construit ainsi possède un élément T_{k0} pour lequel le nombre d'erreurs commises sur l'ensemble de validation est minimal.
- C'est cet arbre-là qui sera finalement retenu par la procédure d'élagage.



Post-élagage de l'algorithme CART



L'ensemble d'apprentissage
pour l'arbre T_{max} .



L'arbre de décision construit sur les
individus d'apprentissage.

Post-élagage de l'algorithme CART

- En appelant n_1 le noeud racine de T_{max} , n_2 et n_3 ses fils Gauche et Droit et n_4 son dernier noeud intérieur (le fils gauche de n_2) :

$$\omega(T_{max}, n_1) = \frac{MC(d, k) - MCT(d, k)}{n(k) \cdot (nt(d, k) - 1)} = \frac{9 - 0}{5 \cdot (5 - 1)} = 9/20$$

$$\omega(T_{max}, n_2) = \frac{1 - 0}{5 \cdot (3 - 1)} = 1/10$$

$$\omega(T_{max}, n_3) = \frac{2 - 0}{5 \cdot (2 - 1)} = 2/5$$

$$\omega(T_{max}, n_4) = \frac{1 - 0}{5 \cdot (2 - 1)} = 1/5$$

- Par conséquent, l'arbre T_1 sera le résultat de l'élagage de T_{max} au noeud n_2 .



Post-élagage de l'algorithme CART

- En travaillant désormais sur T_1 , on trouve les valeurs :

$$\omega(T_1, n_1) = \frac{9 - 1}{3 \cdot (3 - 1)} = 4/3$$

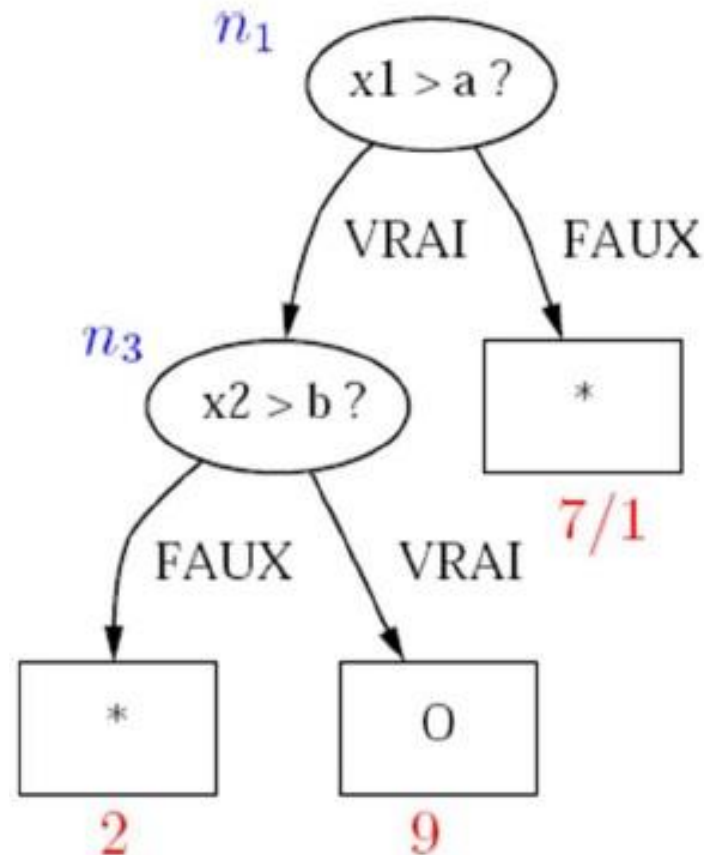
$$\omega(T_1, n_3) = \frac{2 - 0}{3 \cdot (2 - 1)} = 2/3$$

- L'arbre T_2 choisi résultera de l'élagage de n_3 dans T_1 ; il aura donc pour seul nœud la racine de T_{max} , avec une feuille pour chaque classe.

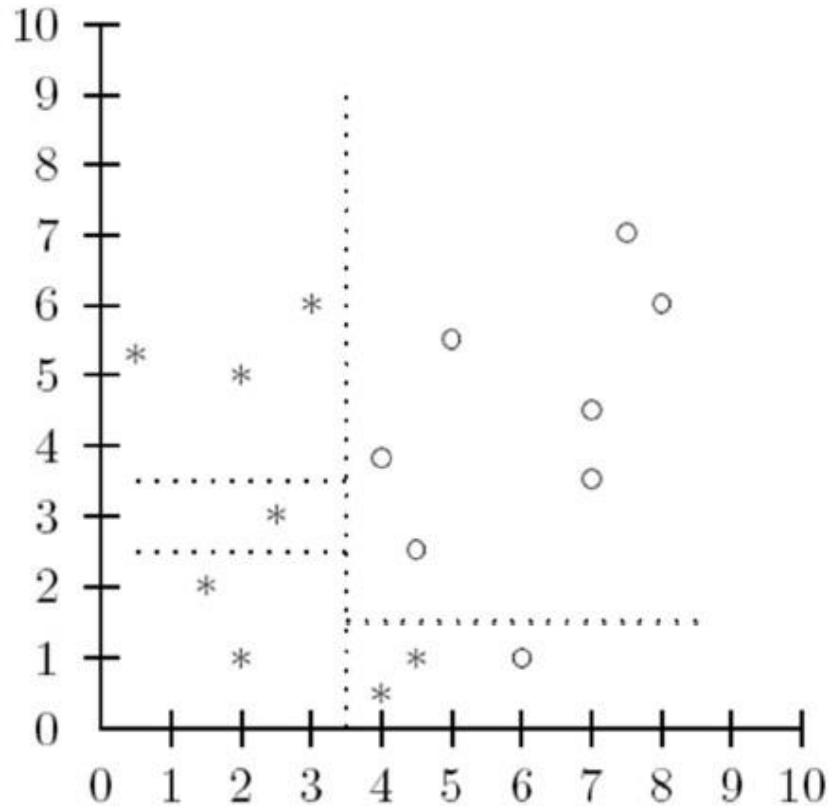


Post-élagage de l'algorithme CART

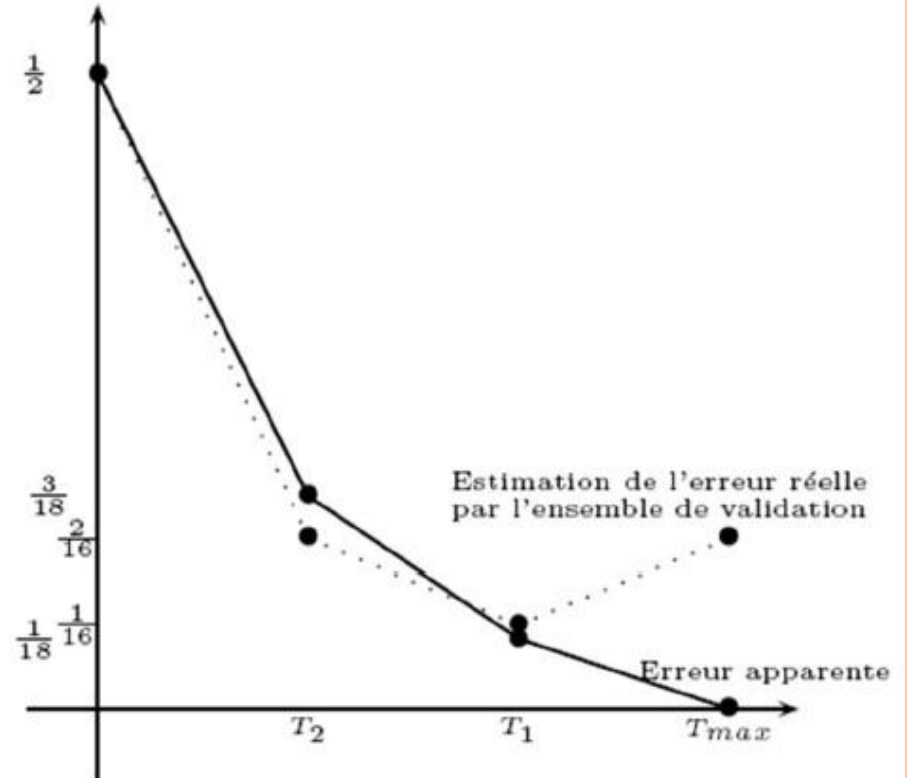
- On suppose disposer d'un ensemble de validation. On teste ce dernier sur les arbres T_{max} , T_1 et T_2 que l'on vient de calculer.



Post-élagage de l'algorithme CART



L'ensemble de validation sur
l'arbre T_{max}.



BIBLIOGRAPHIE

- Jamal Atif : « Apprentissage Articiel et fouille de donnees »: « Arbres de decision », Universite Paris Dauphine D'apres Celine Hudelot (ECP), d'apres Tan, Steinbach, Kumar , M2R ISI Universite Paris-Dauphine
- Cécile Capponi : Arbres de décision, cours M2 MASS. Université Aix-Marseille