Classification et prédiction

Classification vs. Prédiction

- Classification:
  - Classifier les données (construire un modèle) en se basant sur un ensemble où l'on connaît déjà l'association données-classes (*training set: ensemble d'apprentissage*)
- Prédiction:
  - Modéliser des valeurs connues pour prédire des valeurs inconnues

Classification: Définition

- Étant donné une collection d'enregistrements (ensemble d'apprentissage)
  - Chaque enregistrement est caractérisé par un tuple (x, y), où x est l'ensemble des attributs et y l'étiquette de la classe.
    - *x : attribut, prédicteur, variable indépendante, entrée*
    - *y : classe, réponse, variable dépendante, résultat*
- Tâche :
  - Entrainer un modèle qui associe chaque ensemble d'attributs x à l'une des étiquettes de classe prédéfinies y.

Exemples de tâches de classification

Tâche	Ensemble d'attributs, x	Étiquette de la classe, y
Catégorisation des messages électroniques	Caractéristiques extraites de l'en-tête et du contenu des messages électroniques	spam ou non-spam
Identifier les cellules tumorales	Caractéristiques extraites de radiographies ou d'IRM	cellules malignes ou bénignes
Cataloguer les galaxies	Caractéristiques extraites des images de télescopes	Galaxies elliptiques, spirales ou de forme irrégulière

Approche générale pour l'élaboration d'un modèle de classification

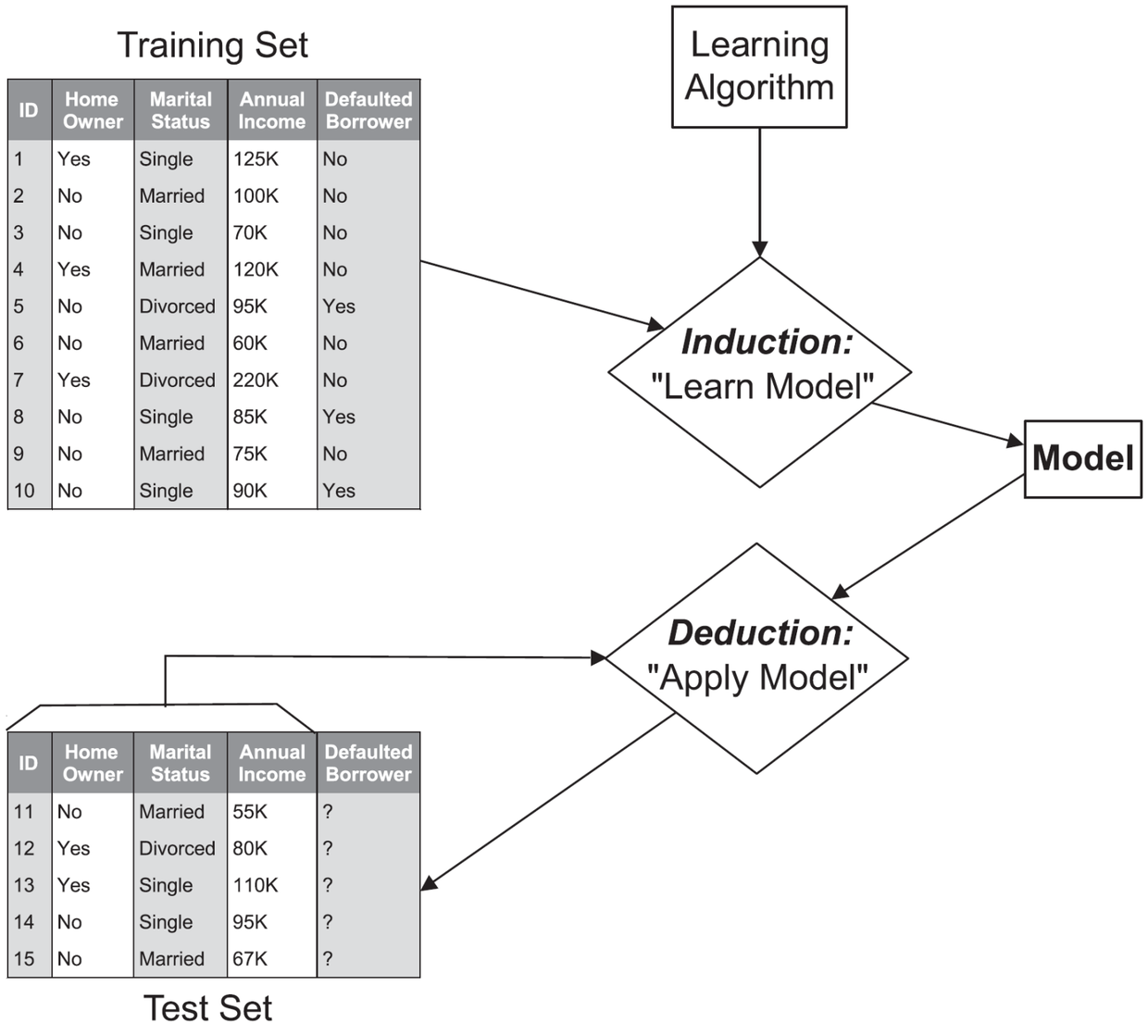
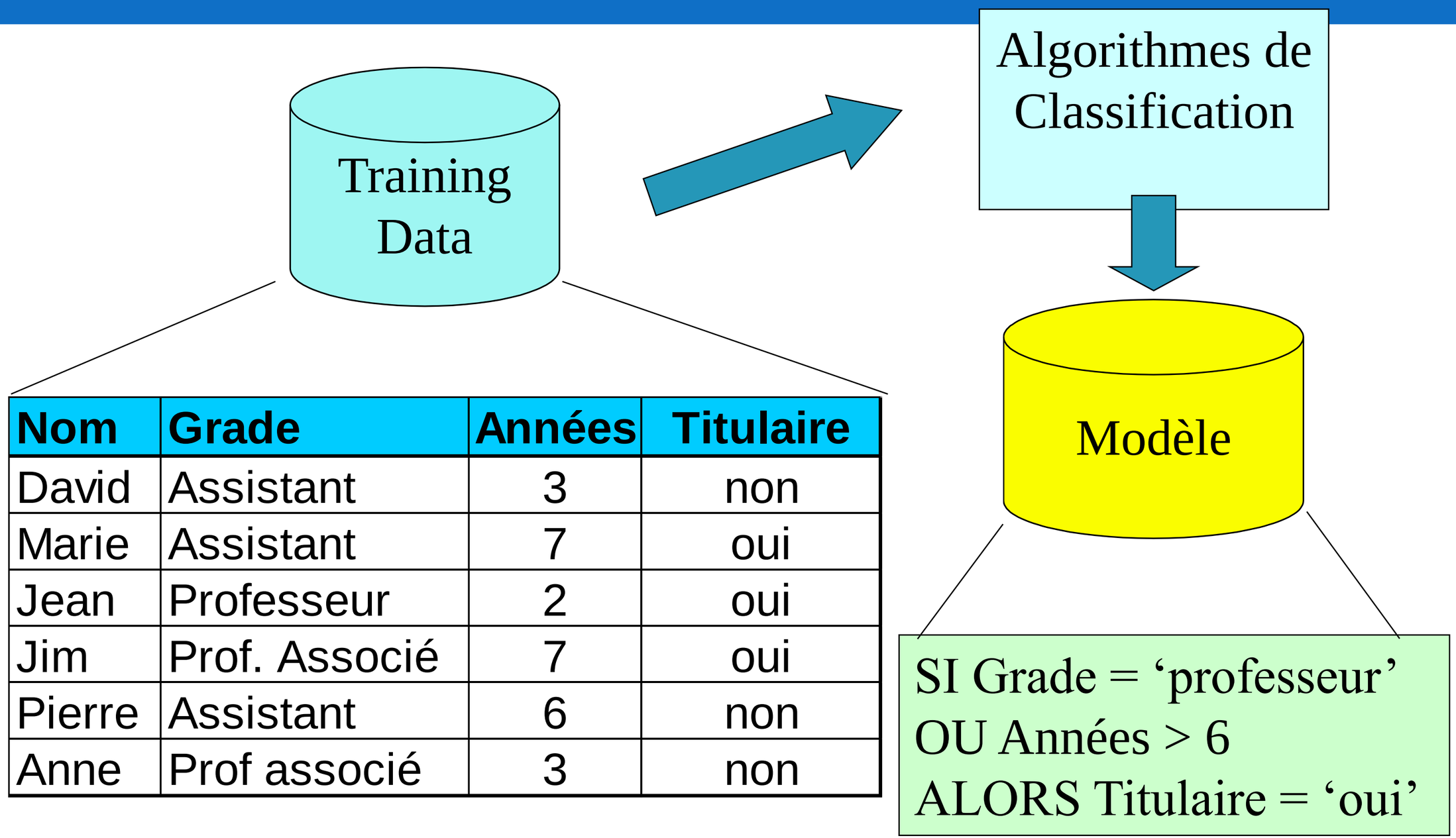


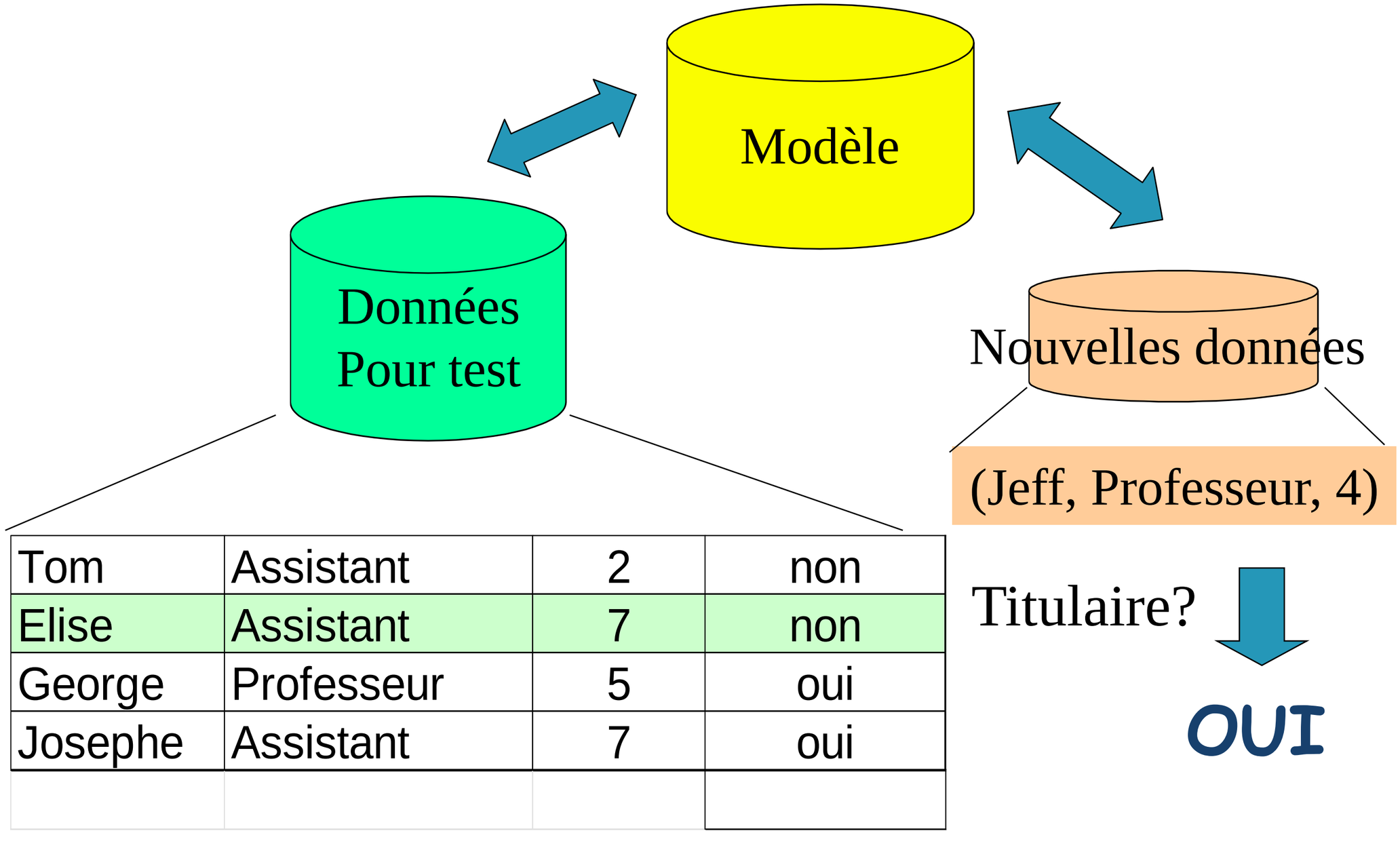
Figure 3.3. General framework for building a classification model.

Processus de Classification (1): Construction du modèle



Nom	Grade	Années	Titulaire
David	Assistant	3	non
Marie	Assistant	7	oui
Jean	Professeur	2	oui
Jim	Prof. Associé	7	oui
Pierre	Assistant	6	non
Anne	Prof associé	3	non

Processus de Classification (2): Prédiction



Tom	Assistant	2	non
Elise	Assistant	7	non
George	Professeur	5	oui
Josephe	Assistant	7	oui

Classification Techniques

- Classificateurs de base
  - Méthodes basées sur les arbres de décision
  - Méthodes fondées sur des règles
  - Plus proche Voisin (Nearest-neighbor)
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
  - Neural Networks, Deep Neural Nets

- Ensemble Classifiers (methodes d'agrégation)
  - Boosting, Bagging, Forêts aléatoires

Apprentissage Supervisé vs non supervisé

- Apprentissage Supervisé (classification)
  - Supervision: les données d'apprentissage (observations) sont accompagnés par les labels indiquant leurs classes
  - Les nouvelles données sont classifiées en se basant sur le training set
- Apprentissage non supervisé (regroupement)
  - Le label de classe des éléments observés (training set) n'est pas connu
  - Le but est de déceler l'existence de classes ou groupes dans les données

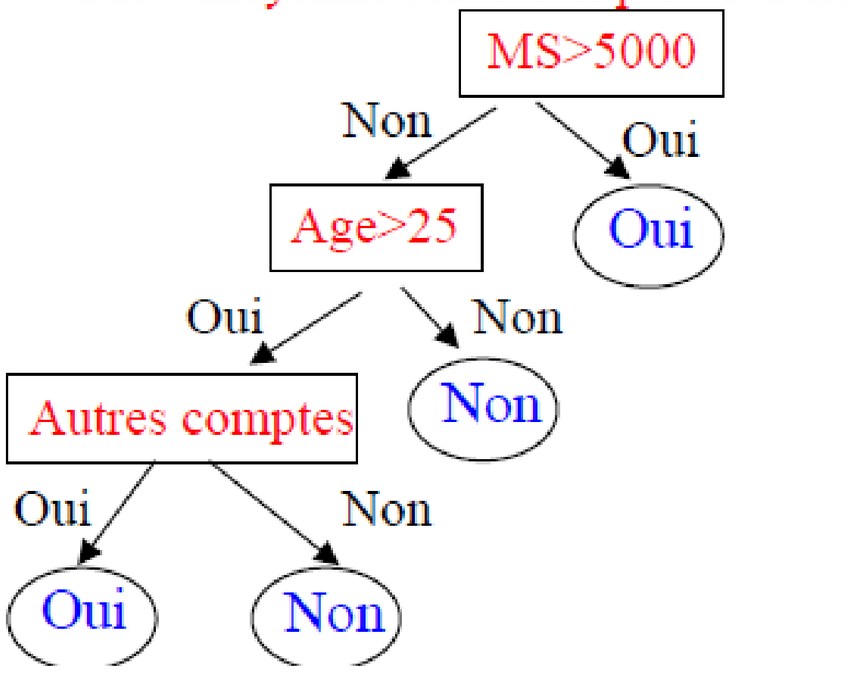
Classification avec arbres de décision

- Arbre de Décision
  - Les nœuds internes correspondent à des tests
  - Un arc correspond au résultat d'un test
  - Les nœuds feuilles représentent des classes
- La génération se fait en 2 phases
  - Construction de l'arbre
    - Au début tous les tuples se trouvent sur la racine
    - Partitionner les tuples récursivement en se basant à chaque fois sur un attribut sélectionné
  - Simplification de l'arbre
    - Identifier et supprimer les branches qui correspondent à des exceptions
- Utilisation:
  - Tester les attributs du tuple par rapport à l'arbre pour trouver la branche et qu'il satisfait donc sa classe

- Génération d'arbres de décision à partir des données
- Arbre = Représentation graphique d'une procédure de classification

Accord d'un prêt bancaire

MS : moyenne solde compte courant



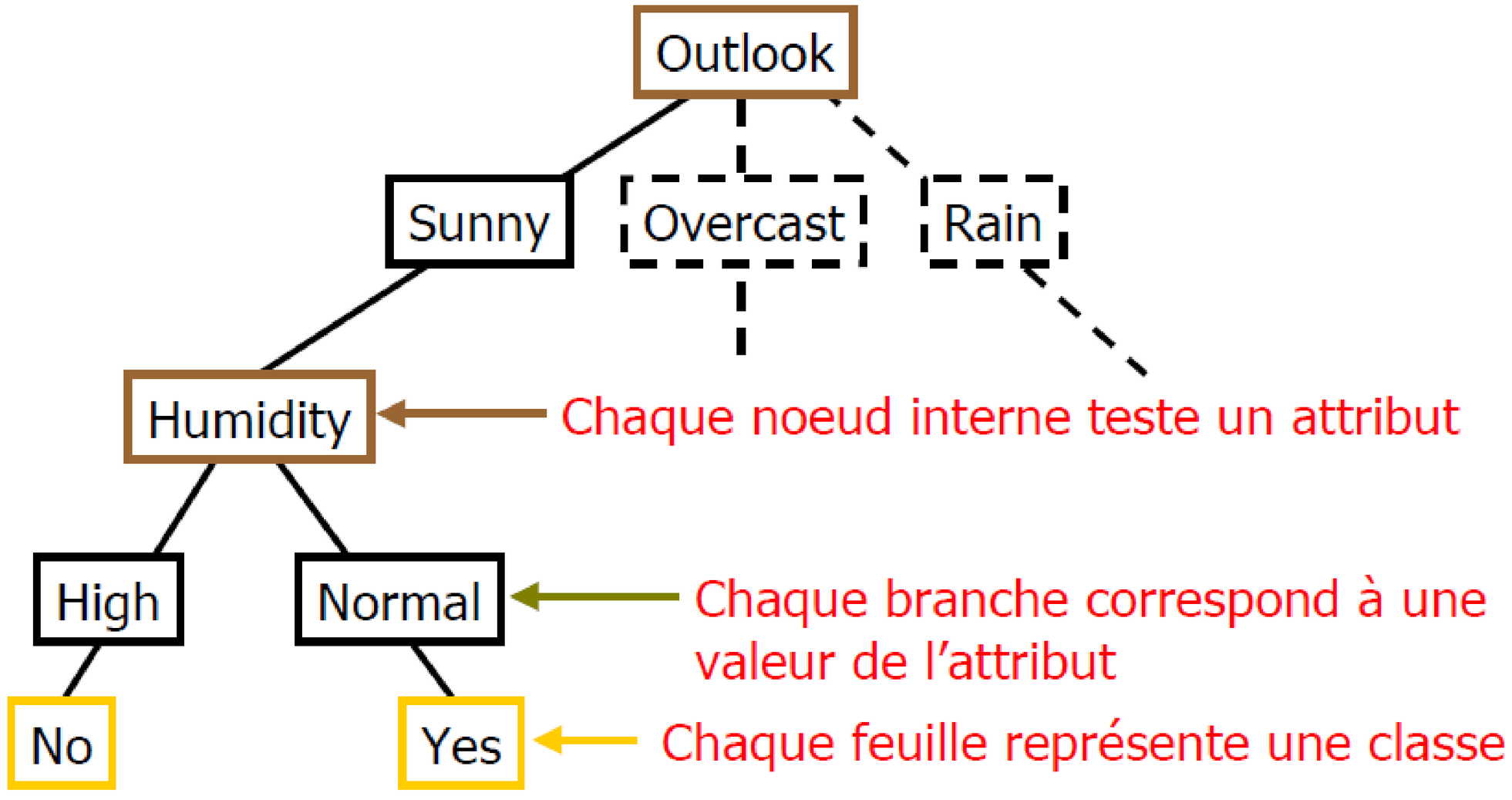
Un arbre de décision est un arbre où :

- **Noeud interne** = un attribut
- **Branche d'un noeud** = un test sur un attribut
- **Feuilles** = classe donnée

Ensemble d'apprentissage

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

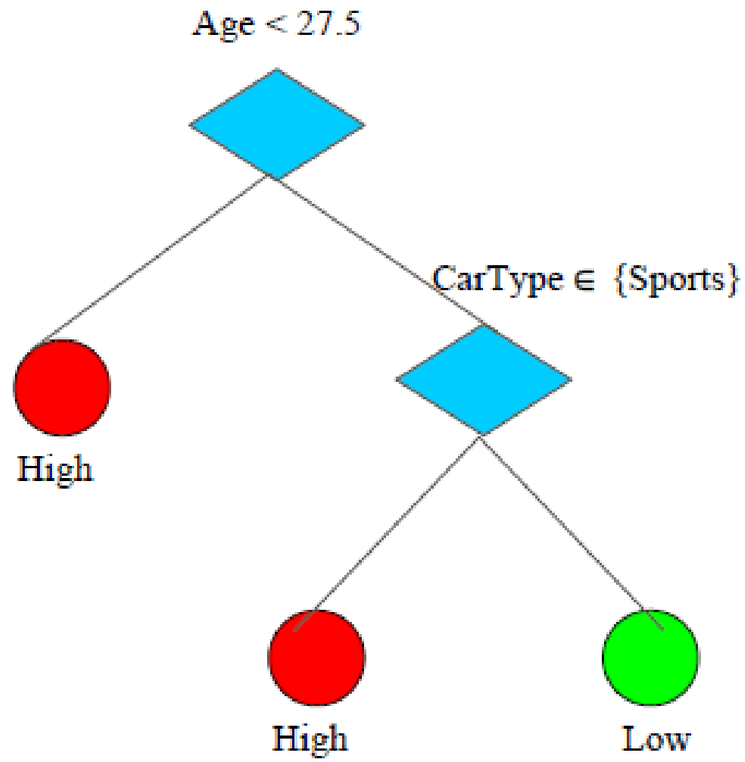
Jouer au tennis ?


Outlook
Sunny
Overcast
Rain
Humidity
Chaque noeud interne teste un attribut
High
Normal
Chaque branche correspond à une valeur de l'attribut
No
Yes
Chaque feuille représente une classe

Risque - Assurances

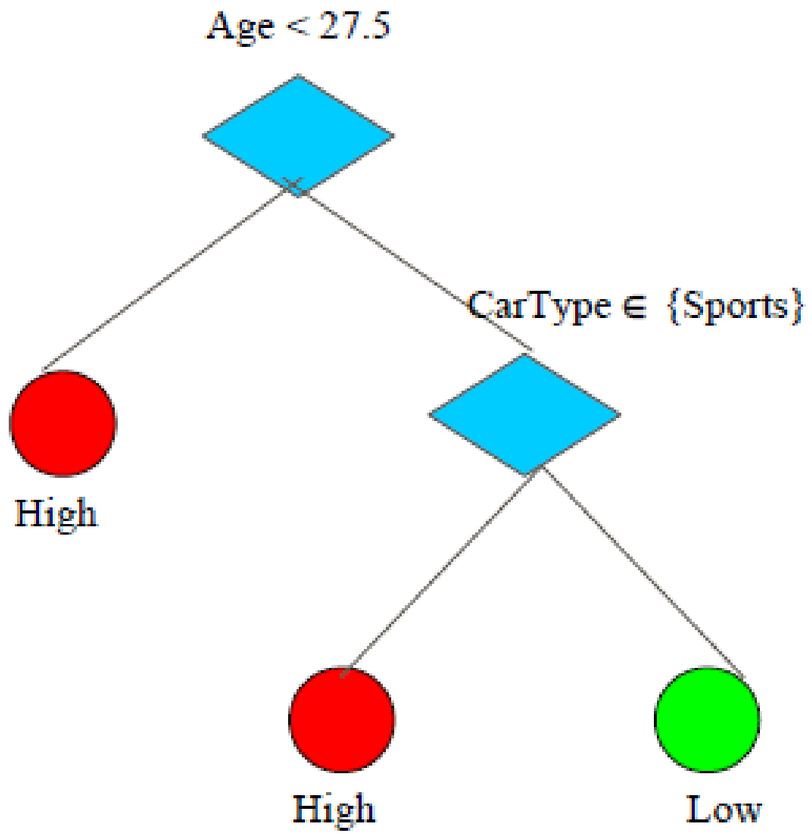
Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Numérique Enumératif



Age=40, CarType=Family ⟹ Class=Low

Des arbres de décision aux règles



1) Age < 27.5 ⇒ High

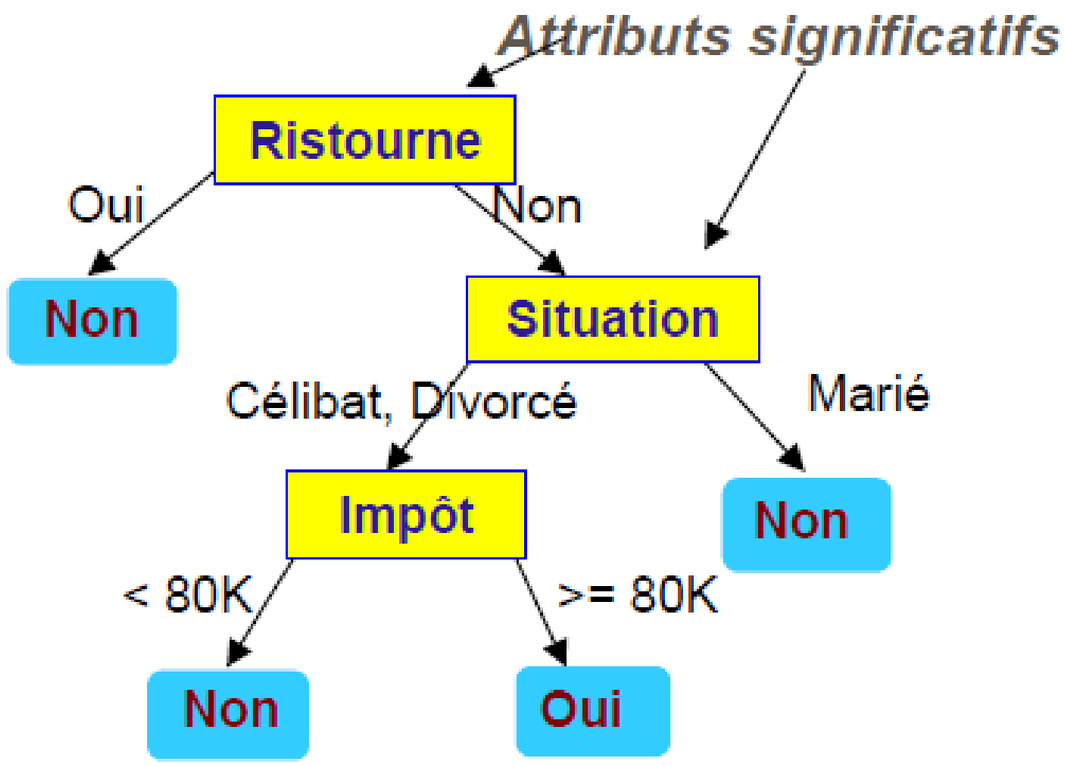
2) Age >= 27.5 and
CarType = Sports ⇒ High

3) Age >= 27.5 and
CarType ≠ Sports ⇒ High

Exemple: Détection de fraudes fiscales

énumératif énumératif numérique classe

Id	Ristourne	Situation famille	Impôt revenu	Fraude
1	Oui	Célibat.	125K	Non
2	Non	Marié	100K	Non
3	Non	Célibat.	70K	Non
4	Oui	Marié	120K	Non
5	Non	Divorcé	95K	Oui
6	Non	Marié	60K	Non
7	Oui	Divorcé	220K	Non
8	Non	Célibat.	85K	Oui
9	Non	Marié	75K	Non
10	Non	Célibat.	90K	Oui



- **L'attribut significatif à un noeud est déterminé en se basant sur l'indice**
- **Pour classer une instance : descendre dans l'arbre selon les réponses aux différents tests. Ex = (Ristourne=Non, Situation=Divorcé, Impôt=100K) ➔ Oui**

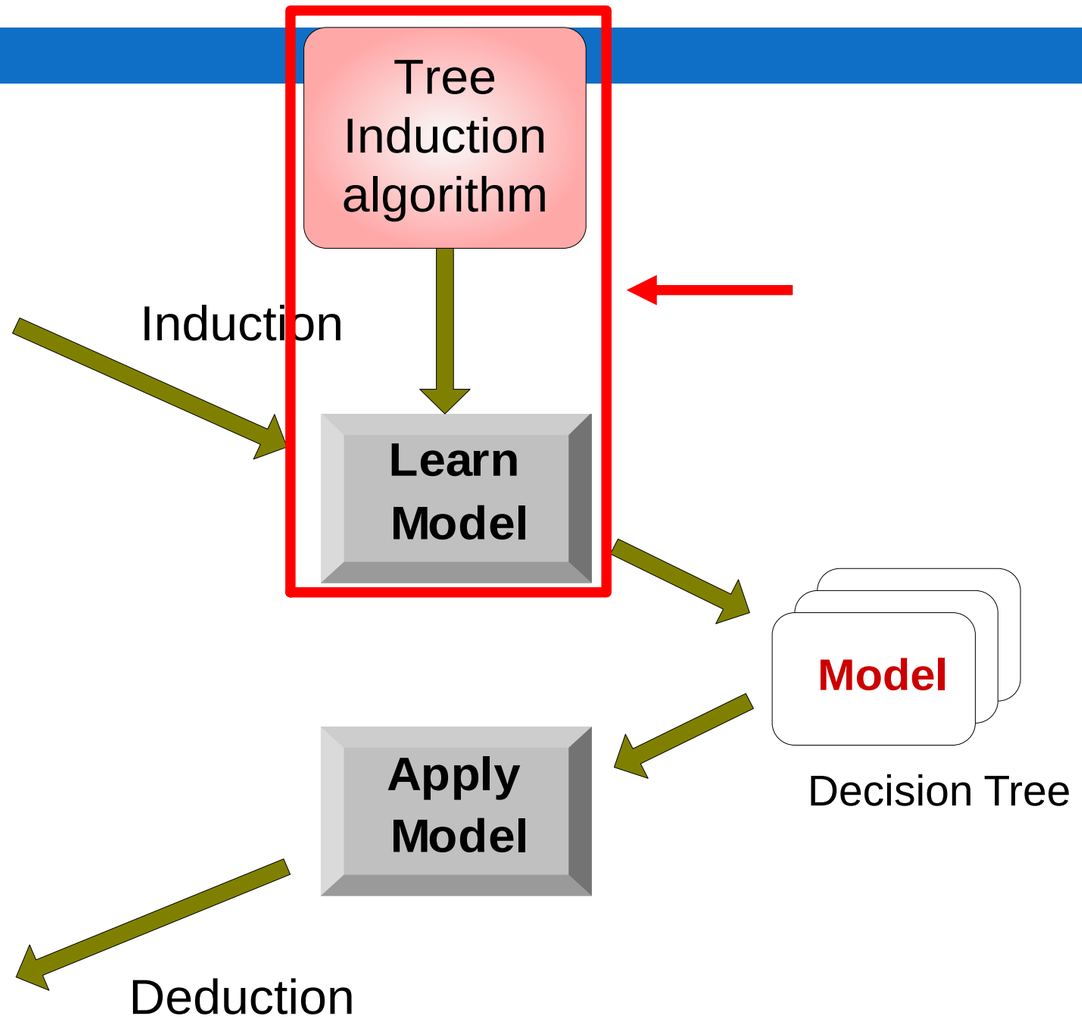
Tâche de classification par arbre de décision

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

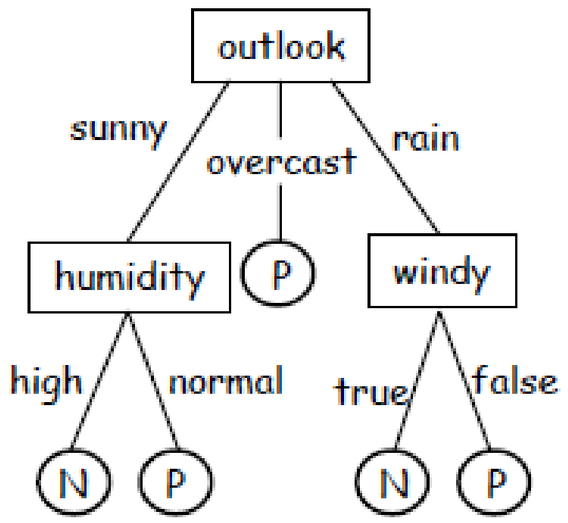
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Des arbres de décision aux règles

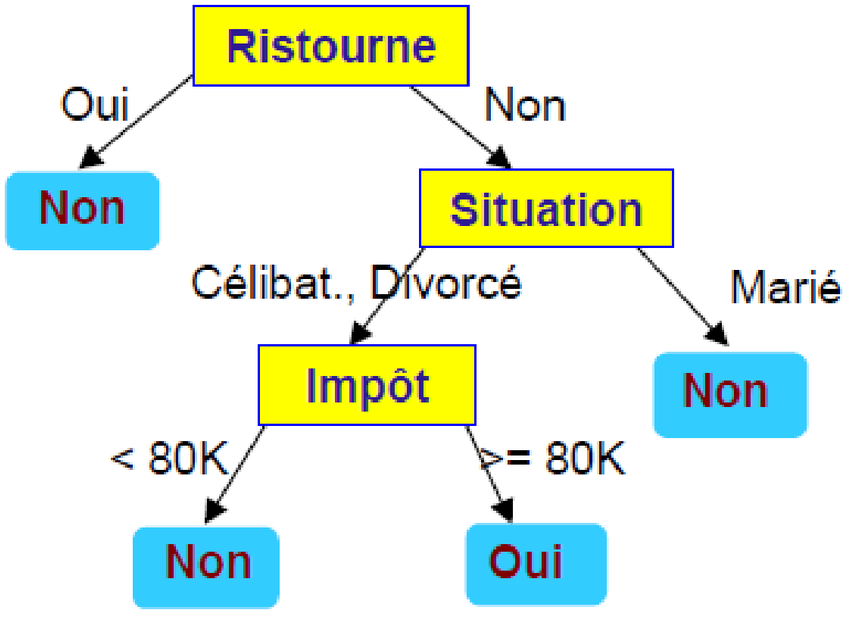


Si outlook=sunny
Et humidity=normal
Alors play tennis

- une **règle** est générée pour chaque **chemin** de l'arbre (de la racine à une feuille)
- Les paires attribut-valeur d'un chemin forment une conjonction
- Le noeud terminal représente la classe prédite
- Les règles sont généralement plus faciles à comprendre que les arbres

Des arbres de décision aux règles

Arbre de décision = Système de règles exhaustives et mutuellement exclusives

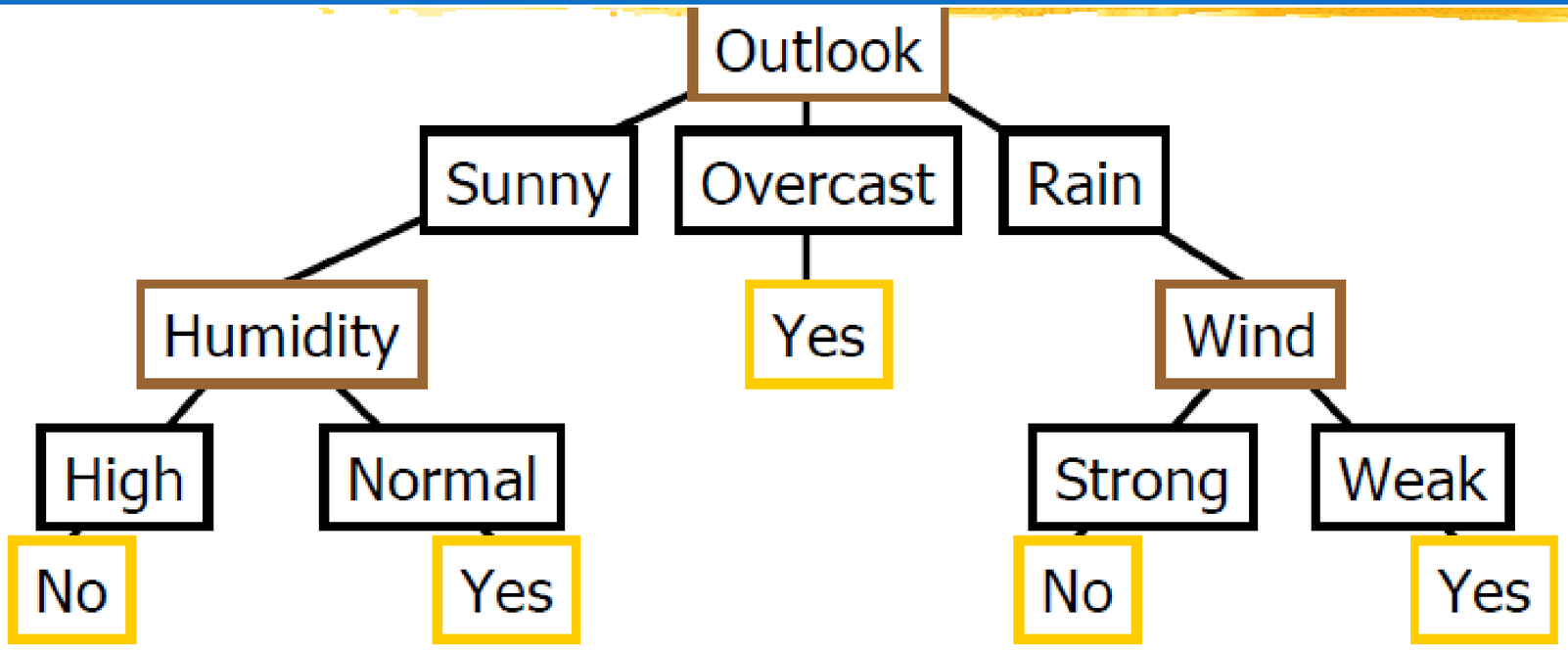


1) Ristourne = Oui ⇒ Non

2) Ristourne = Non et
Situation in {Célibat., Divorcé}
et Impôt < 80K ⇒ Non

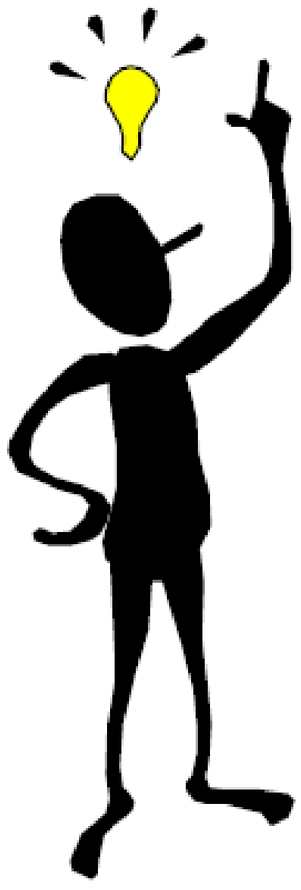
3) Ristourne = Non et
Situation in {Célibat., Divorcé}
et Impôt >= 80K ⇒ Oui

4) Ristourne = Non et
Situation in {Marié} ⇒ Non



R_1: If (Outlook=Sunny) $\wedge$ (Humidity=High) Then PlayTennis=No
R_2: If (Outlook=Sunny) $\wedge$ (Humidity=Normal) Then PlayTennis=Ye
R_3: If (Outlook=Overcast) Then PlayTennis=Yes
R_4: If (Outlook=Rain) $\wedge$ (Wind=Strong) Then PlayTennis=No
R_5: If (Outlook=Rain) $\wedge$ (Wind=Weak) Then PlayTennis=Yes

Génération de l'arbre de décision



Deux phases dans la génération de l'arbre :

- **Construction de l'arbre**
  - Arbre peut atteindre une taille élevée
- **Elaguer l'arbre (Pruning)**
  - Identifier et supprimer les branches qui représentent du "bruit" → Améliorer le taux d'erreur

Algorithme

- Construction de l'arbre
  - Au départ, toutes les instances d'apprentissage sont à la racine de l'arbre
  - Sélectionner un attribut et choisir un test de séparation (split) sur l'attribut, qui sépare le "mieux" les instances.

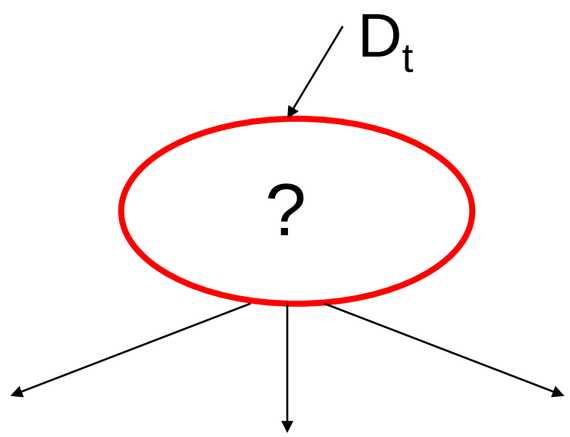
    La sélection des attributs est basée sur une heuristique ou une mesure statistique.
  - Partitionner les instances entre les noeuds fils suivant la satisfaction des tests logiques

- Traiter chaque noeud fils de façon récursive
- Répéter jusqu'à ce que tous les noeuds soient des terminaux. Un noeud courant est terminal si :
  - Il n'y a plus d'attributs disponibles
  - Le noeud est "pur", i.e. toutes les instances appartiennent à une seule classe,
  - Le noeud est "presque pur", i.e. la majorité des instances appartiennent à une seule classe (Ex : 95%)
  - Nombre minimun d'instances par branche (Ex : algorithme C5 évite la croissance de l'arbre, k=2 par défaut)
- Etiqueter le noeud terminal par la classe majoritaire

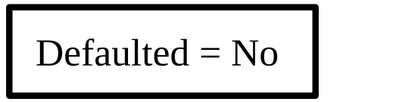
Structure générale de l'algorithme de Hunt (l'un des premiers)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

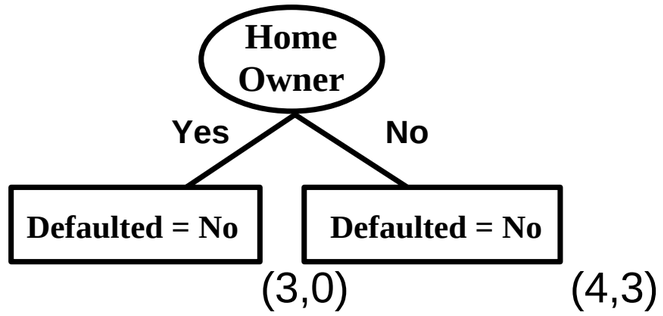
- Soit D_t l'ensemble des enregistrements d'apprentissage qui atteignent un nœud t
- Procédure générale :
  - Si D_t contient des enregistrements appartenant à la même classe y_t, alors t est un nœud feuille étiqueté y_t.
  - Si D_t contient des enregistrements appartenant à plus d'une classe, utilisez un test d'attributs pour diviser les données en sous-ensembles plus petits. Appliquer récursivement la procédure à chaque sous-ensemble.



Algorithme de Hunt (l'un des premiers)

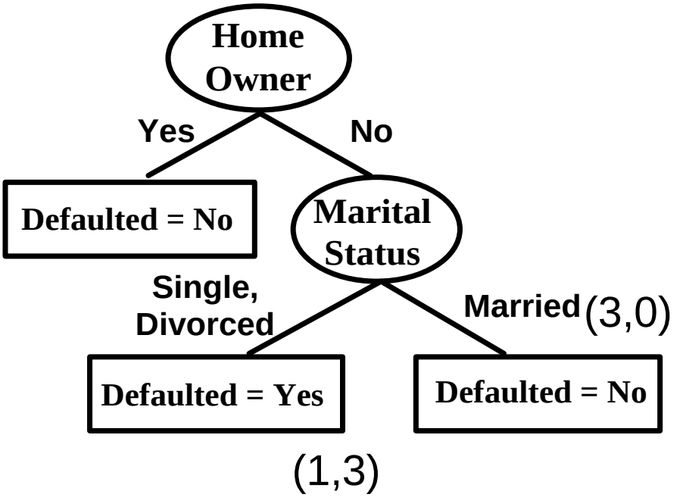


(a)

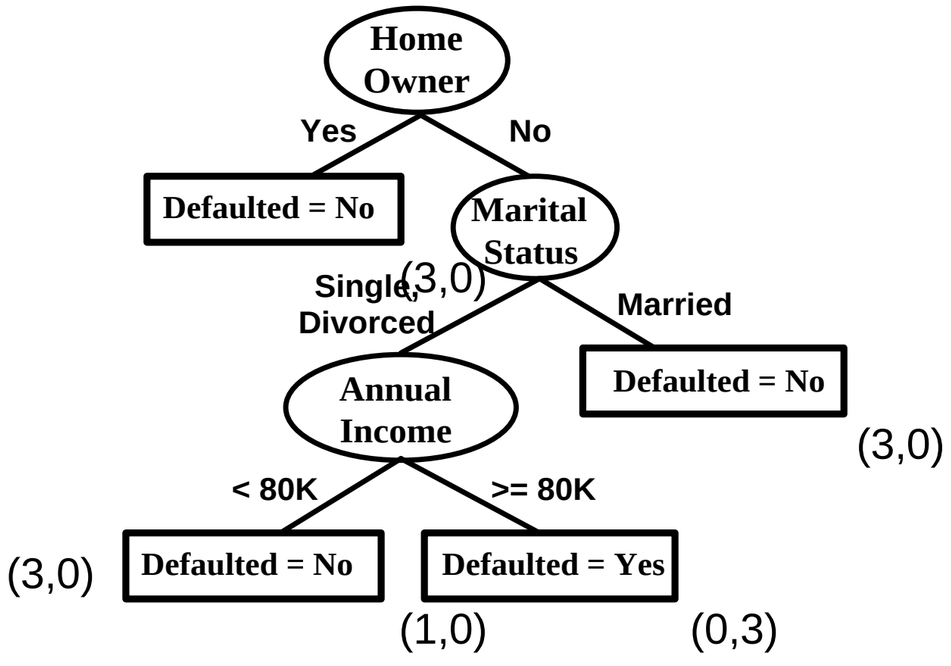


(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



(c)



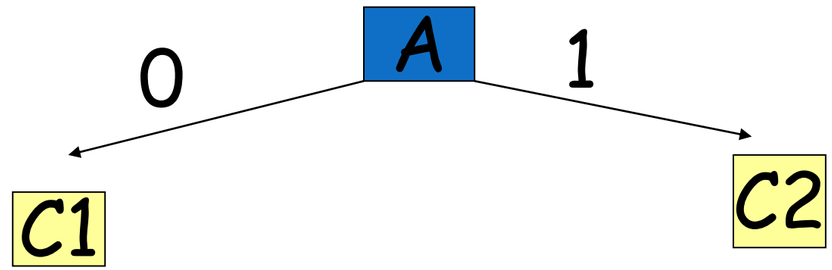
(d)

Choix de l'attribut de partionnement (1)

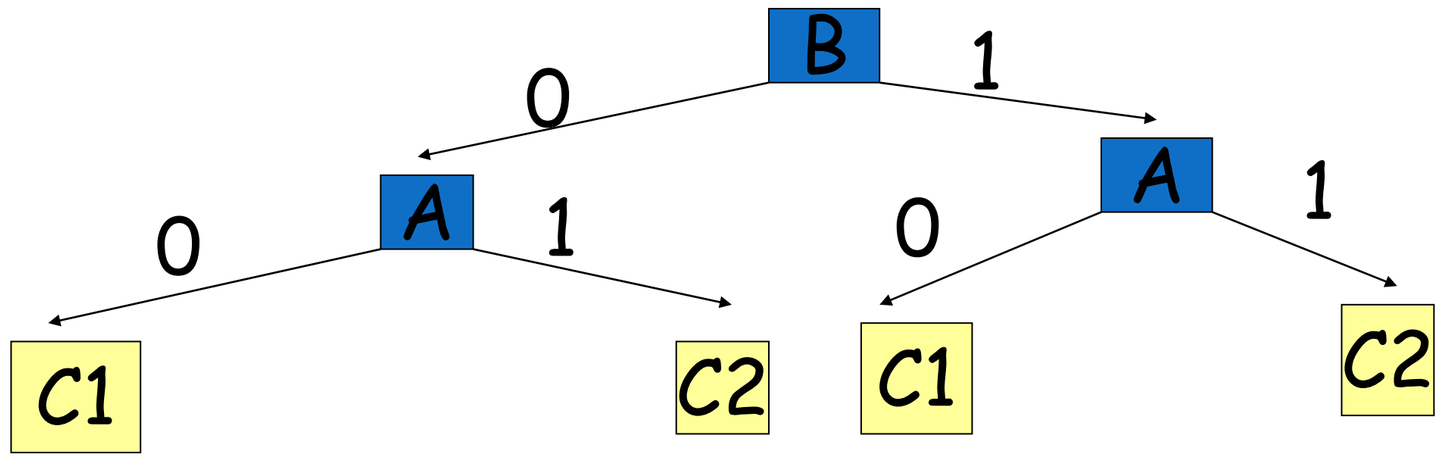
- Soit le training set suivant

A	B	Classe
0	1	C1
0	0	C1
1	1	C2
1	0	C2

Si c'est A qui est choisi en premier



Si c'est B qui est choisi en premier



- **Algorithme de base**
  - Construction récursive d'un arbre de manière "diviser-pour-régner" descendante
  - Attributs considérés énumératifs

- **Plusieurs variantes** : ID3, C4.5, CART, CHAID
  - Différence principale : mesure de sélection d'un attribut – critère de branchement (split)

Mesures de sélection d'attributs

- Gain d'Information (ID3, C4.5)
- Indice Gini (CART)
- Table de contingence statistique χ^2 (CHAID)
- G-statistic

Questions relatives à la conception de l'induction par arbre de décision

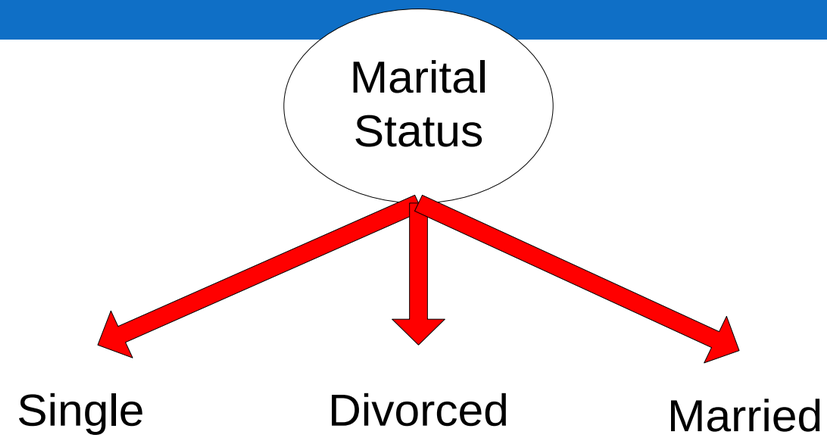
- Comment les enregistrements d'entrainement doivent-ils être répartis ?
  - Méthode pour exprimer la condition de test x
    - en fonction des types d'attributs
  - Mesure permettant d'évaluer la qualité d'une condition de test
- Comment la procédure de découpage doit-elle s'arrêter ?
  - Arrêter le découpage si tous les enregistrements appartiennent à la même classe ou ont des valeurs d'attributs identiques
  - Arrêt anticipé

Méthodes d'expression des conditions de test

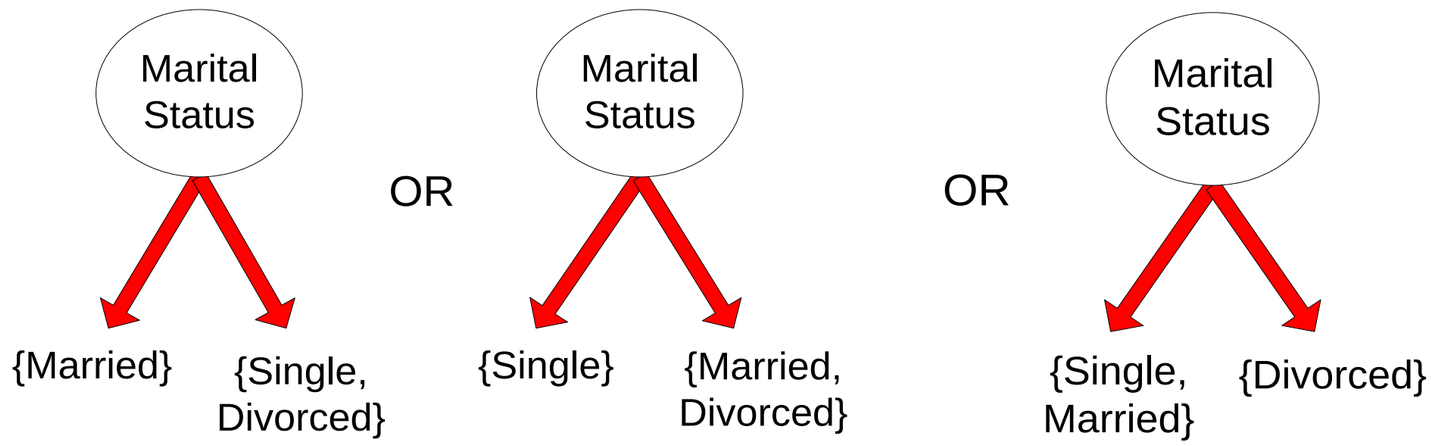
- Dépend des types d'attributs
  - Binaire
  - Nominal
  - Ordinal
  - Continue

Conditions de test pour les attributs nominaux

- Découpage multidirectionnelle :
  - Utiliser autant de partitions que de valeurs distinctes.

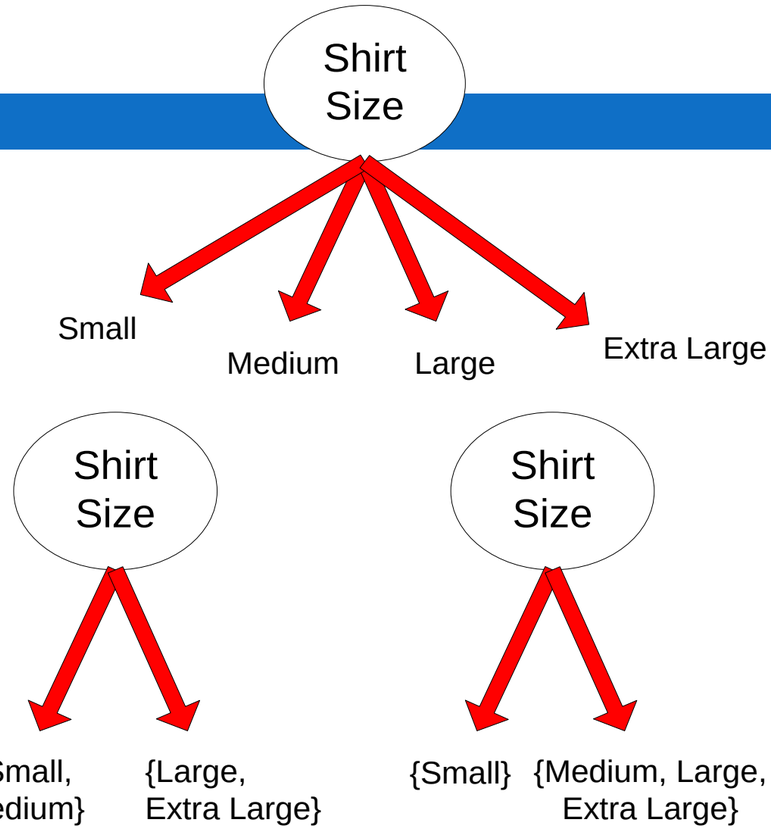


- Découpage binaire :
  - Divise les valeurs en deux sous-ensembles

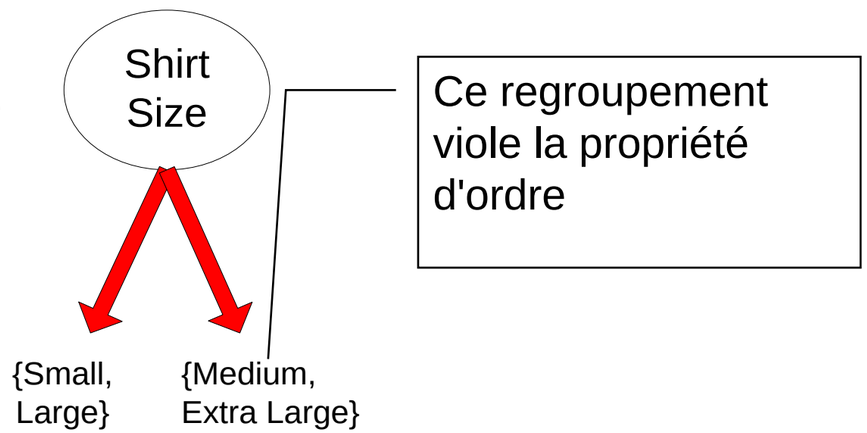


Condition de test pour les attributs ordinaux

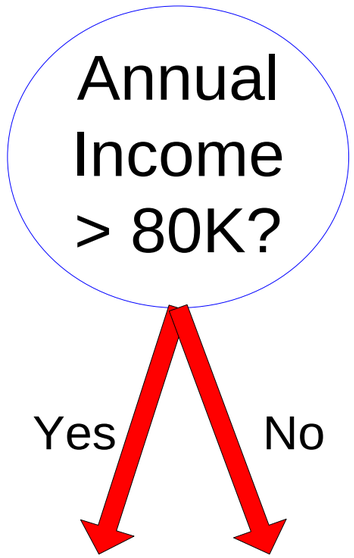
- Découpage multidirectionnelle :
  - Utiliser autant de partitions que de valeurs distinctes



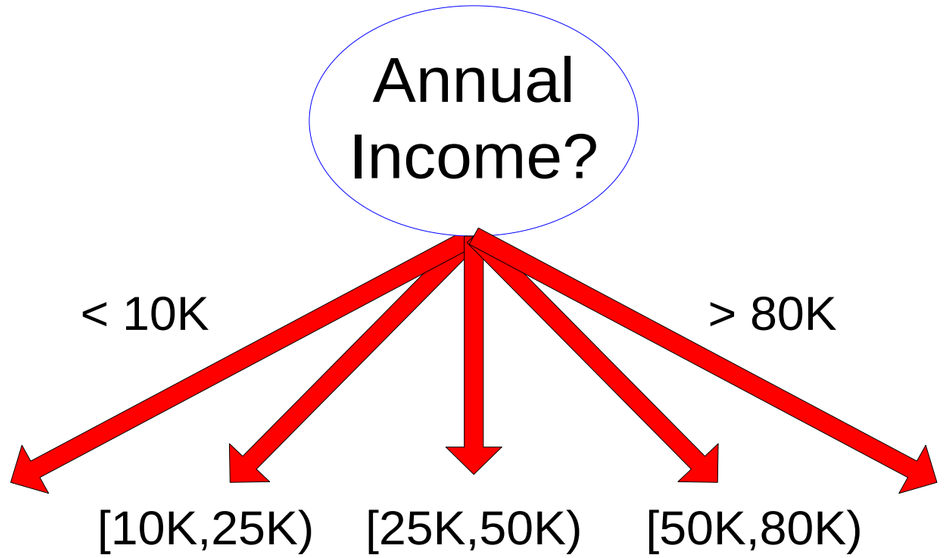
- Découpage binaire :
  - Divise les valeurs en deux sous-ensembles
  - Préserver la propriété d'ordre parmi les valeurs d'attributs



Condition de test pour les attributs continus



(i) Binary split



(ii) Multi-way split

Découpage basé sur des attributs continus

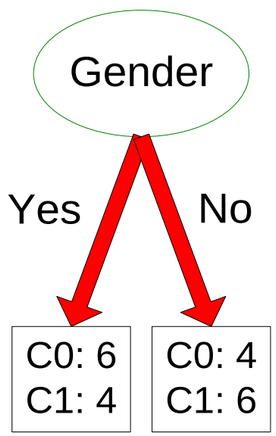
- Différents modes de traitement
  - Discrétisation pour former un attribut catégorial ordinal

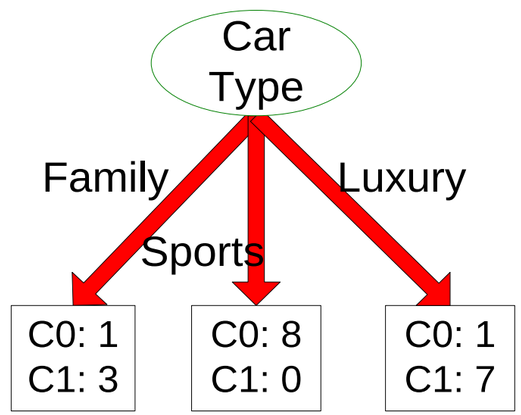
    Les fourchettes peuvent être trouvées par regroupement d'intervalles égaux, regroupement de fréquences égales (percentiles) ou regroupement en grappes.
    - Statique - discrétisation une fois au début
    - Dynamique - se répète à chaque nœud
  - Décision Binaire: $(A < v)$ or $(A \geq v)$
    - considère tous les découpages possibles et trouve le meilleur découpage
    - peut être plus gourmande en ressources informatiques

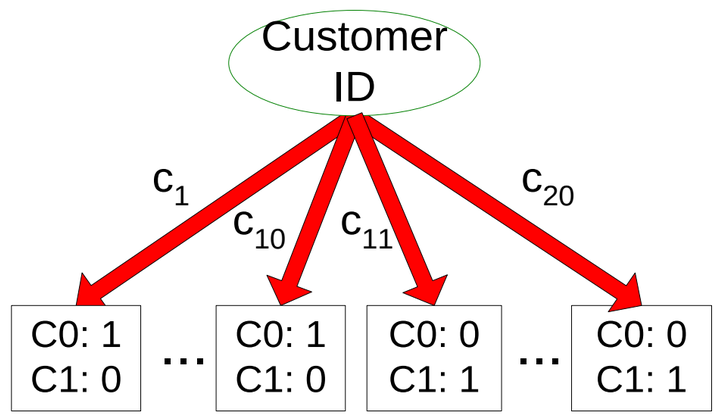
Comment déterminer le meilleur découpage

Avant le découpage: 10 enregistrements de la classe « 0 », 10 enregistrements de la classe « 1 »

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1







Quelle est la meilleure condition de test ?

Comment déterminer le meilleur découpage

- Approche gourmande :
  - Les nœuds dont la distribution des classes est plus pure sont privilégiés
- Besoin d'une mesure de l'impureté d'un nœud :

C0: 5
C1: 5

Haut degré d'impureté

C0: 9
C1: 1

Faible degré d'impureté

Mesures de l'impureté des nœuds

- Indice de Gini

$$\text{Indice de } Gini = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

- Entropie

$$Entropie = -\sum_{i=0}^{c-1} p_i(t) log_2 p_i(t)$$

- Erreur de classification

$$erreur\ de\ Classification = 1 - \max[p_i(t)]$$

Où $\boldsymbol{p_i(t)}$ est la fréquence de la classe $\boldsymbol{i}$ au nœud t, et $\boldsymbol{c}$ est le nombre total de classes.

Trouver le meilleur découpage

1. Calculer la mesure d'impureté (P) avant le découpage
2. Calculer la mesure d'impureté (M) après découpage
   - Calculer la mesure d'impureté de chaque nœud enfant
   - M est l'impureté pondérée des nœuds enfants
3. Choisissez la condition de test d'attribut qui produit le gain le plus élevé

   Gain = P - M

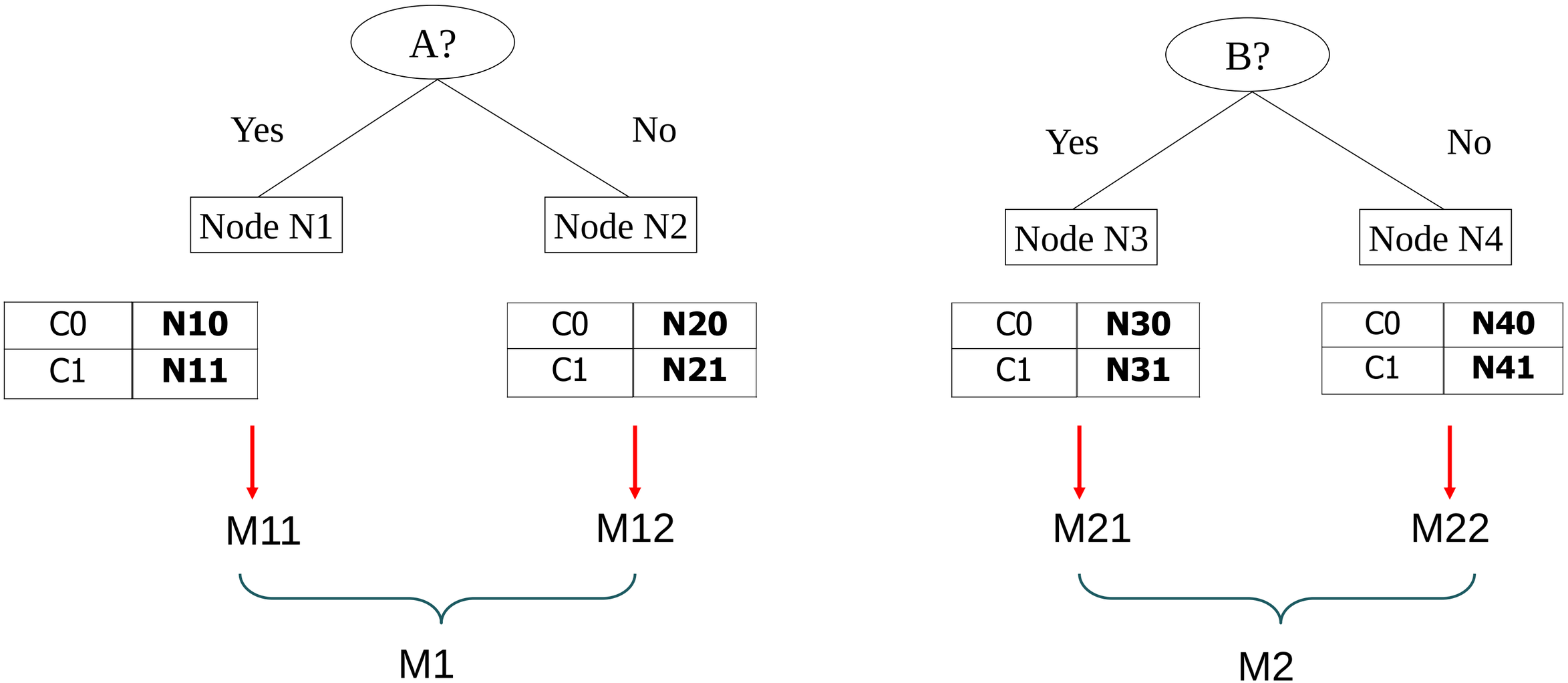
   ou, de manière équivalente, la mesure d'impureté la plus faible (M) après découpage

Trouver le meilleur découpage

Avant le découpage :

C0	**N00**
C1	**N01**

→ P



Gain = P – M1 vs P – M2

Mesure de l'impureté : GINI

- Indice de Gini pour un nœud donné t

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Où $\boldsymbol{p_i(t)}$ est la fréquence de la classe $\boldsymbol{i}$ au nœud t, et $\boldsymbol{c}$ est le nombre total de classes.

- Maximum de 1-1/c lorsque les enregistrements sont également répartis entre toutes les classes, ce qui implique la situation la moins avantageuse pour la classification.

- Minimum de 0 lorsque tous les enregistrements appartiennent à une seule classe, ce qui implique la situation la plus avantageuse pour la classification.

- L'indice de Gini est utilisé dans les algorithmes d'arbres de décision tels que CART, SLIQ, SPRINT.

Mesure de l'impureté : GINI

- Indice de Gini pour un nœud donné t :

$$Indice\ de\ Gini\ = 1\ - \sum_{i=0}^{c-1} p_i(t)^2$$

- Pour un problème à 2 classes (p, 1 – p):
  - GINI = $1 - p^2 - (1 - p)^2 = 2p\ (1-p)$

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Calcul de l'indice de Gini pour un seul nœud

$$Indice\ de\ Gini = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)^2 – (5/6)^2 = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)^2 – (4/6)^2 = 0.444

Calcul de l'indice de Gini pour une collection de nœuds

- Lorsqu'un nœud p est divisé en k partitions (enfants)

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

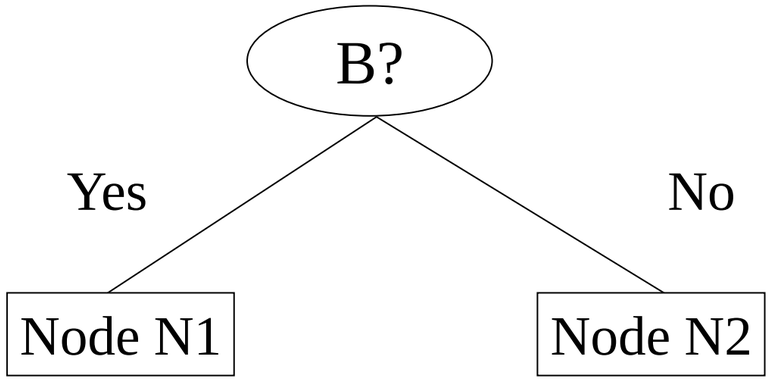
où,

n_i = nombre d'enregistrements chez l'enfant i,

n = nombre d'enregistrements au nœud parent p.

Attributs binaires : Calcul de l'indice GINI

- Se divise en deux partitions (nœuds enfants)
- Effet des partitions pondérées :
  - Des partitions plus grandes et plus pures sont recherchées



	Parent
C1	**7**
C2	**5**
Gini = 0.486	

Gini(N1)
= 1 – $(5/6)^2$ – $(1/6)^2$
= 0.278

Gini(N2)
= 1 – $(2/6)^2$ – $(4/6)^2$
= 0.444

	N1	N2
C1	**5**	**2**
C2	**1**	**4**
Gini=0.361		

Gini pondéré de N1 N2

= 6/12 * 0.278 +
  6/12 * 0.444
= 0.361

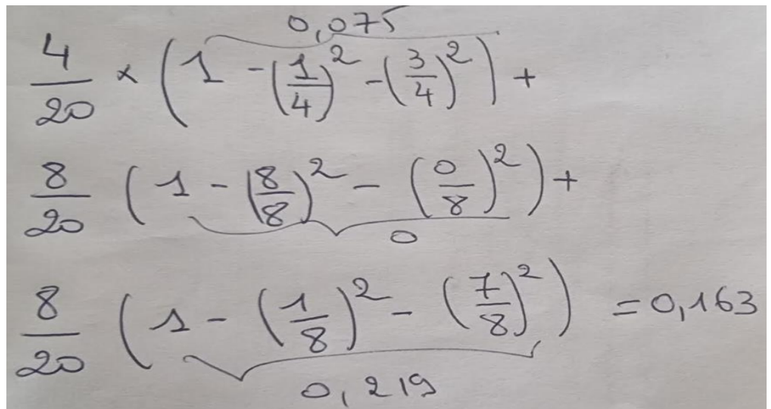
Gain = 0.486 – 0.361 = 0.125

Attributs catégoriels : Calcul de l'indice de Gini

- Pour chaque valeur distincte, rassemblez les comptes pour chaque classe de l'ensemble de données.
- Utiliser la matrice de comptage pour prendre des décisions

Découpage multiple

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		



Découpage à deux voies (trouver le meilleur découpage des valeurs)

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

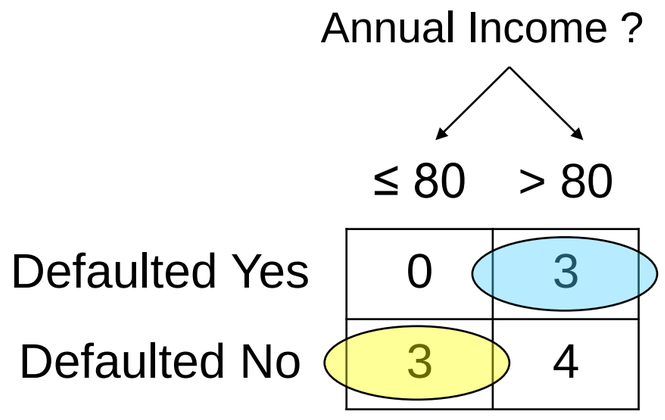
	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

Lequel est le meilleur ?

Attributs continus : Calcul de l'indice de Gini

- Utiliser des décisions binaires basées sur une valeur
- Plusieurs choix pour la valeur du découpage
  - Nombre de valeurs de découpage possibles = Nombre de valeurs distinctes
- Chaque valeur de découpage est associée à une matrice de comptage
  - Nombre de classes dans chacune des partitions, $A \leq v$ et $A > v$
- Méthode simple pour choisir le meilleur v
  - Pour chaque v, parcourir la base de données pour obtenir la matrice de comptage et calculer l'indice de Gini.
  - Inefficacité sur le plan informatique ! Plusieurs répétitions

ID	Home Owner	Marital Status	Annual Income	Defaulted
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Attributs continus : Calcul de l'indice de Gini...

- Pour un calcul efficace : pour chaque attribut,
  - Trier l'attribut par valeurs
  - Balayer linéairement ces valeurs, en mettant à chaque fois à jour la matrice de comptage et en calculant l'indice de Gini.
  - Choisir la position de partage qui a l'indice de Gini le plus faible

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Annual Income										
Valeurs triées →	60	70	75	85	90	95	100	120	125	220

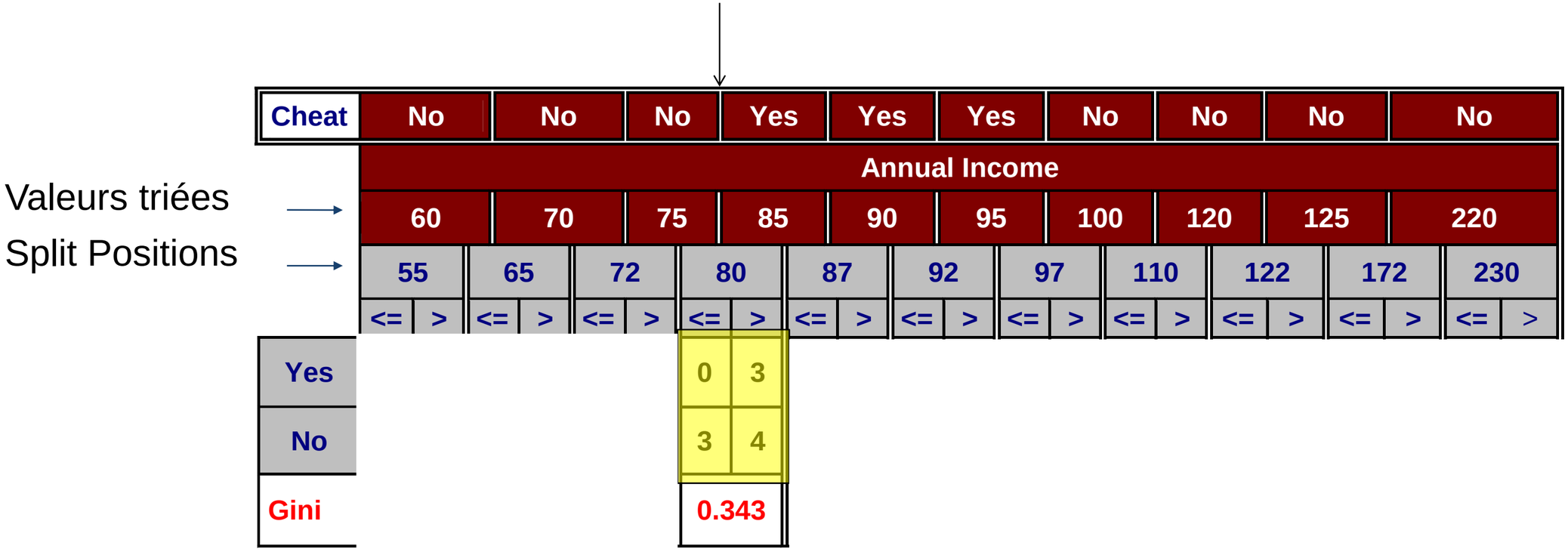
Attributs continus : Calcul de l'indice de Gini...

- Pour un calcul efficace : pour chaque attribut,
  - **Trier** l'attribut par valeurs
  - Balayer linéairement ces valeurs, en mettant à chaque fois à jour la matrice de comptage et en calculant l'indice de Gini.
  - Choisir la position de partage qui a l'indice de Gini le plus faible

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Annual Income										
Valeurs triées →	60	70	75	85	90	95	100	120	125	220
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >

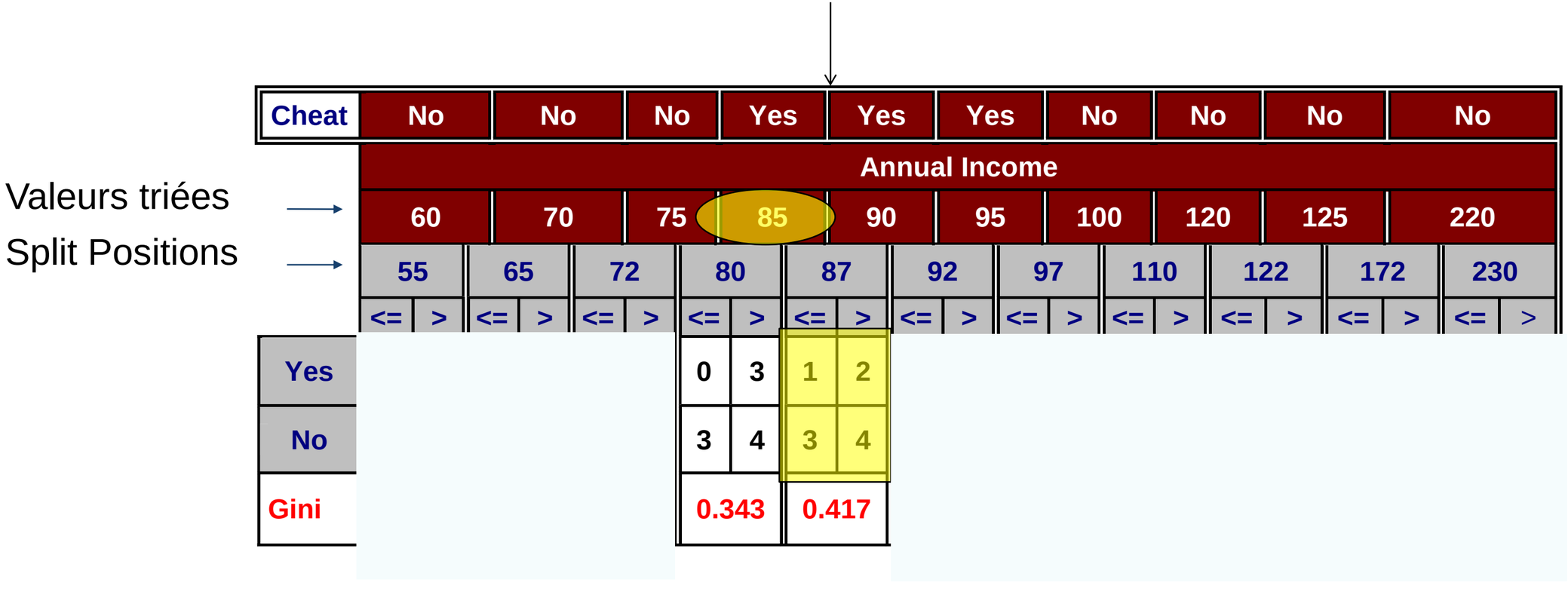
Continuous Attributes: Computing Gini Index...

- Pour un calcul efficace : pour chaque attribut,
  - Trier l'attribut par valeurs
  - Balayer linéairement ces valeurs, en mettant à chaque fois à jour la matrice de comptage et en calculant l'indice de Gini.
  - Choisir la position de partage qui a l'indice de Gini le plus faible



Continuous Attributes: Computing Gini Index...

- Pour un calcul efficace : pour chaque attribut,
  - Trier l'attribut par valeurs
  - Balayer linéairement ces valeurs, en mettant à chaque fois à jour la matrice de comptage et en calculant l'indice de Gini.
  - Choisir la position de partage qui a l'indice de Gini le plus faible



Continuous Attributes: Computing Gini Index...

- Pour un calcul efficace : pour chaque attribut,
  - Trier l'attribut par valeurs
  - Balayer linéairement ces valeurs, en mettant à chaque fois à jour la matrice de comptage et en calculant l'indice de Gini.
  - Choisir la position de partage qui a l'indice de Gini le plus faible

Sorted Values →

Split Positions →

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Annual Income										
	60	70	75	85	90	95	100	120	125	220
	55	65	72	80	87	92	97	110	122	172	230

	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

Mesure de l'impureté : Entropie

- Entropie à un nœud donné t

$$Entropie = -\sum_{i=0}^{c-1} p_i(t) log_2 p_i(t)$$

Où $\boldsymbol{p_i(t)}$ est la fréquence de la classe $\boldsymbol{i}$ au nœud t, et $\boldsymbol{c}$ est le nombre total de classes.

  - Maximum du $\log_2 c$ lorsque les enregistrements sont également répartis entre toutes les classes, ce qui implique la situation la moins avantageuse pour la classification.

  - Minimum de 0 lorsque tous les enregistrements appartiennent à la même classe, ce qui implique la situation la plus avantageuse pour la classification.

- Les calculs basés sur l'entropie sont assez similaires aux calculs de l'indice de GINI.

Calcul de l'entropie d'un seul nœud

$$Entropie = -\sum_{i=0}^{c-1} p_i(t) log_2 p_i(t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropie = – 1 log 1 = – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropie = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropie = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Calcul du gain d'information après découpage

- Gain d'Information :

$$Gain_{split} = Entropie(p) - \sum_{i=1}^{k} \frac{n_i}{n} Entropie(i)$$

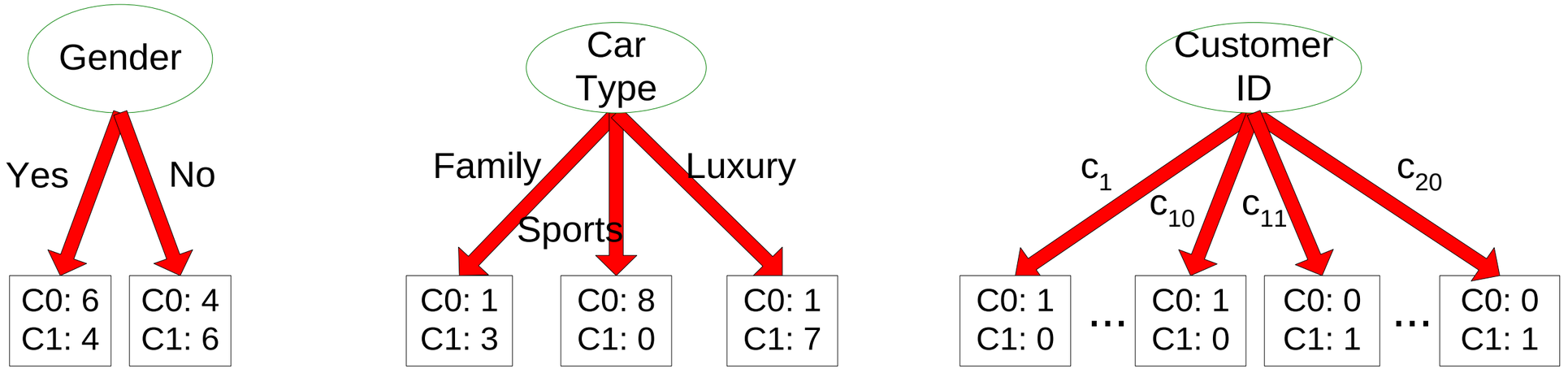
Le nœud parent p est divisé en k partitions (enfants)

n_i est le nombre d'enregistrements dans le nœud enfant i

- Choisir la division qui permet d'obtenir la plus grande réduction (maximiser le GAIN)
- Utilisé dans les algorithmes d'arbre de décision **ID3 et C4.5**
- Le gain d'information est l'information mutuelle entre la variable de classe et la variable de découpage.

Problème avec un grand nombre de partitions

- Les mesures d'impureté des nœuds tendent à préférer les découpages qui aboutissent à un grand nombre de partitions, chacune étant petite mais pure.



  - L'identifiant du client présente le gain d'information le plus élevé, car l'entropie de tous les enfants est nulle.

Gain Ratio

- Gain Ratio:

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info} \qquad Split\ Info = -\sum_{i=1}^{k} \frac{n_i}{n} log_2 \frac{n_i}{n}$$

  Le nœud parent p est divisé en k partitions (enfants)

  n_i est le nombre d'enregistrements dans le nœud enfant i

  - Ajuste le gain d'information en fonction de l'entropie du partitionnement ($Split\ Info$).
    - Le partitionnement à forte entropie (grand nombre de petites partitions) est pénalisé !
  - Utilisé dans l'algorithme C4.5
  - Conçu pour surmonter l'inconvénient du gain d'information

Gain Ratio

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info} \qquad Split\ Info = -\sum_{i=1}^{k} \frac{n_i}{n} log_2 \frac{n_i}{n}$$

Le nœud parent p est divisé en k partitions (enfants)

n_i est le nombre d'enregistrements dans le nœud enfant i

	CarType		
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52

	CarType	
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

SplitINFO = 0.72

	CarType	
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

SplitINFO = 0.97

((0,2×log(0,2)÷0,3)+(0,4
×log(0,4)÷0,3)+(0,4×log(0,4)
÷0,3))×(−1)

1,5271533593

4/20=0,2
8/20=0,4

Mesure de l'impureté : Erreur de classification

- Erreur de classification à un nœud t

$$Error(t) = 1 - \max_i [p_i(t)]$$

  - Maximum de 1-1/c lorsque les enregistrements sont également répartis entre toutes les classes, ce qui implique la situation la moins intéressante.
  - Minimum de 0 lorsque tous les enregistrements appartiennent à la même classe, ce qui implique la situation la plus intéressante.

Erreur de calcul d'un seul nœud

$$Error(t) = 1 - \max_i [p_i(t)]$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

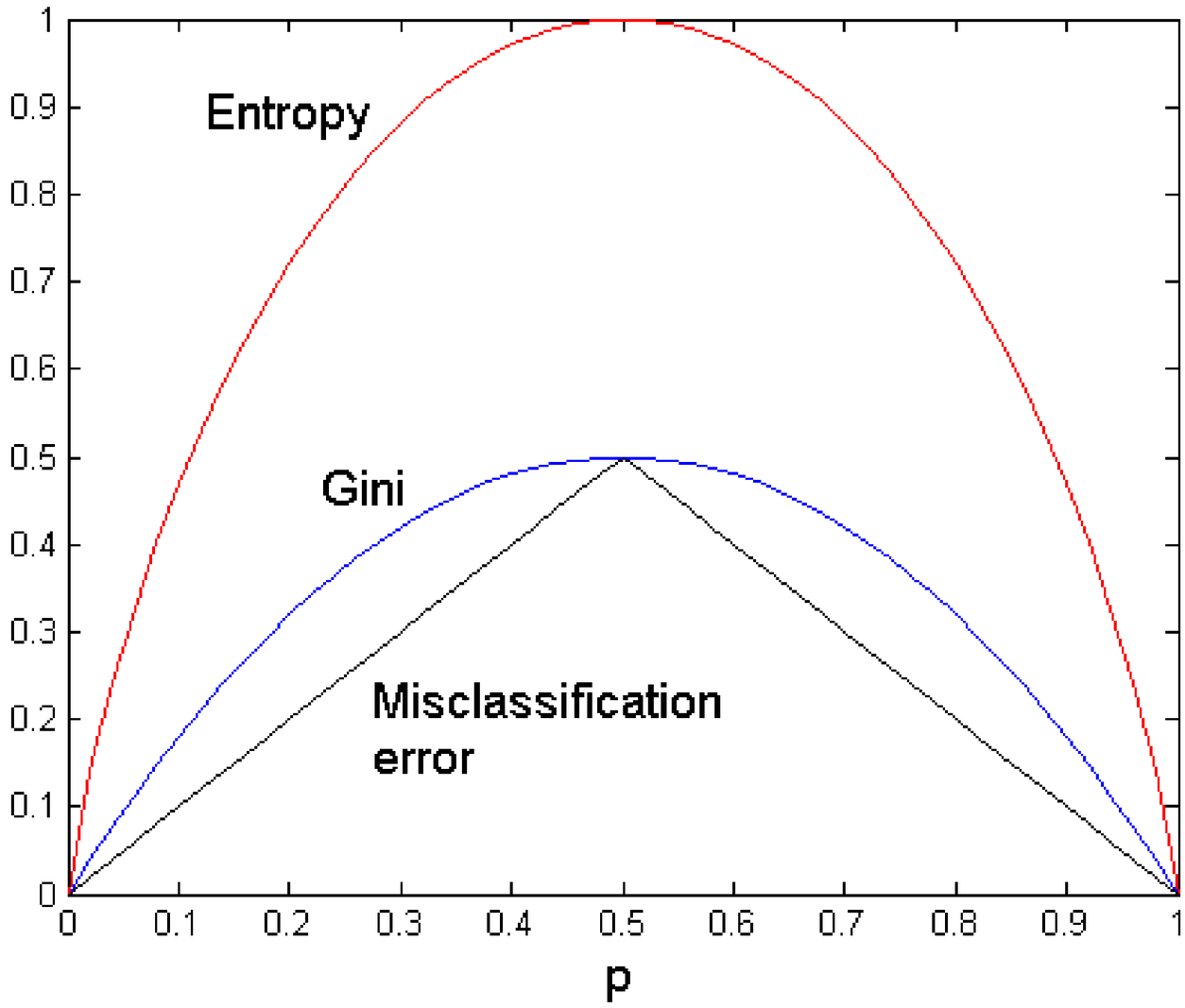
C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Comparaison des mesures d'impureté

Pour un problème à 2 classes :



Algorithmes

- Les deux algorithmes les plus connus et les plus utilisés (l'un ou l'autre ou les deux sont présents dans les environnements de fouille de données) sont CART (Classification And Regression Trees [BFOS84]) et C5 (version la plus récente après ID3 et C4.5 [Qui93]).

- [BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Technical report, Wadsworth International, Monterey, CA, 1984.
- [Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

Avantages des AD

- Compréhensible pour tout utilisateur (lisibilité du résultat - règles - arbre)
- Justification de la classification d'une instance (racine → feuille)
- Tout type de données
- Robuste au bruit et aux valeurs manquantes
- Attributs apparaissent dans l'ordre de pertinence → tâche de pré-traitement (sélection d'attributs)
- Classification rapide (parcours d'un chemin dans un arbre)
- Outils disponibles dans la plupart des environnements de data mining

Inconvénients

- Sensibles au nombre de classes : performances se dégradent

- Evolutivité dans le temps : si les données évoluent dans le temps, il est nécessaire de relance la phase d'apprentissage