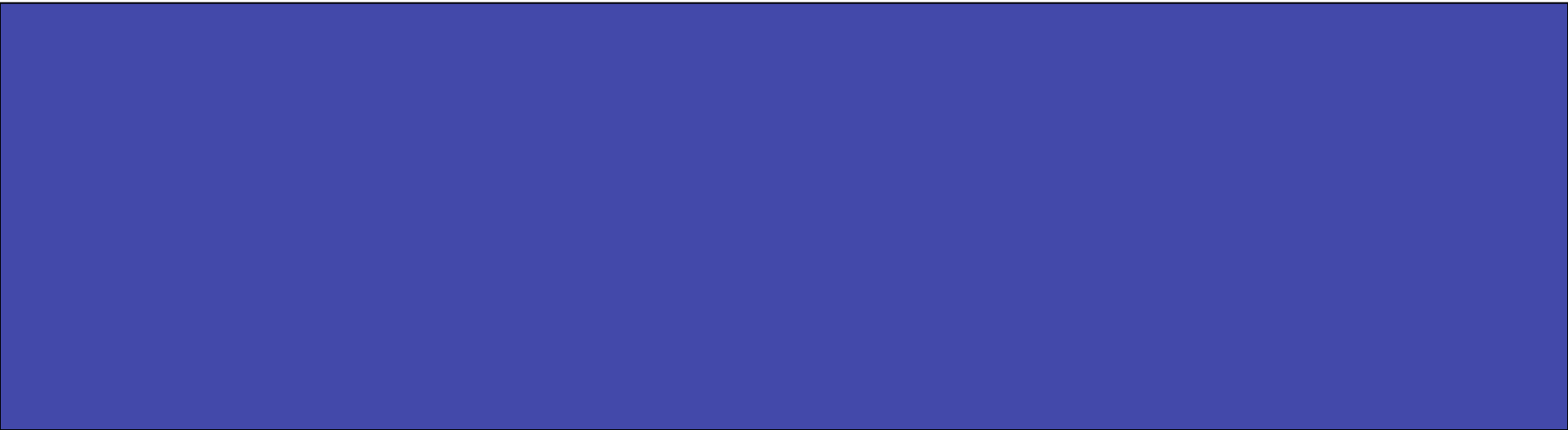


# Classification



# Classification

- Elle permet de **prédire** si un élément est membre d'un groupe ou d'une catégorie donnée.
- **Classes**
  - Identification de groupes avec des profils particuliers
  - Possibilité de décider de l'appartenance d'une entité à une classe
- Caractéristiques
  - **Apprentissage supervisé** : classes connues à l'avance
  - Pb : qualité de la classification (taux d'erreur)
    - Ex : établir un diagnostic (si erreur !!!)

# Classification - Applications

Comprendre les critères prépondérants pour l'achat d'un produit ou d'un service

Isoler les critères explicatifs d'un comportement d'achat

Analyse de risque: détecter les facteurs prédisant un comportement de non paiement

Détecter les causes de réclamation

# Processus à deux étapes



## Etape 1 :

Construction du modèle à partir de l'ensemble d'apprentissage (training set)

## Etape 2 :

Utilisation du modèle : tester la précision du modèle et l'utiliser dans la classification de nouvelles données

# Construction du modèle



Chaque **instance** est supposée appartenir à une classe prédéfinie

La classe d'une instance est déterminée par l'attribut "**classe**"

L'ensemble des instances d'apprentissage est utilisé dans la construction du modèle

Le **modèle** est représenté par des règles de classification, arbres de décision, formules mathématiques, ...

# Utilisation du modèle

Classification de nouvelles instances ou instances inconnues



Estimer le taux d'erreur du modèle

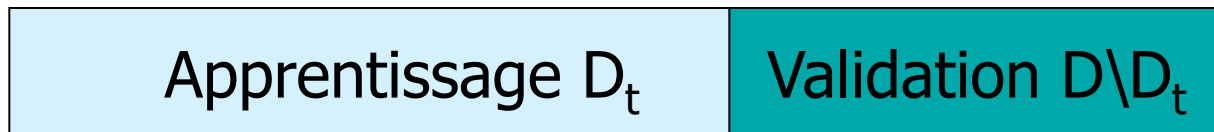
- la classe connue d'une instance test est comparée avec le résultat du modèle
- Taux d'erreur = pourcentage de tests incorrectement classés par le modèle

# Validation de la Classification (accuracy)

Estimation des taux d'erreurs :

Partitionnement : apprentissage et test (ensemble de données important)

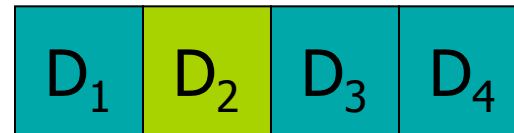
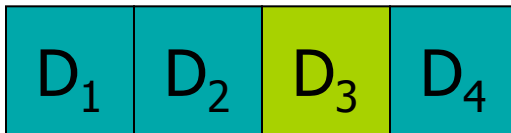
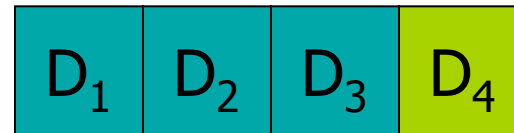
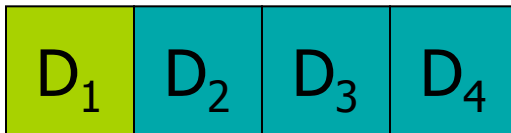
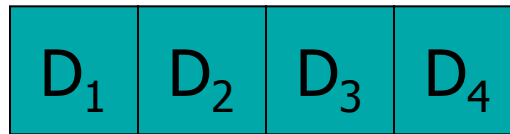
- Utiliser 2 ensembles indépendants, e.g., ensemble d'apprentissage (2/3), ensemble test (1/3)



# Validation de la Classification (accuracy)

## Validation croisée (ensemble de données modéré)

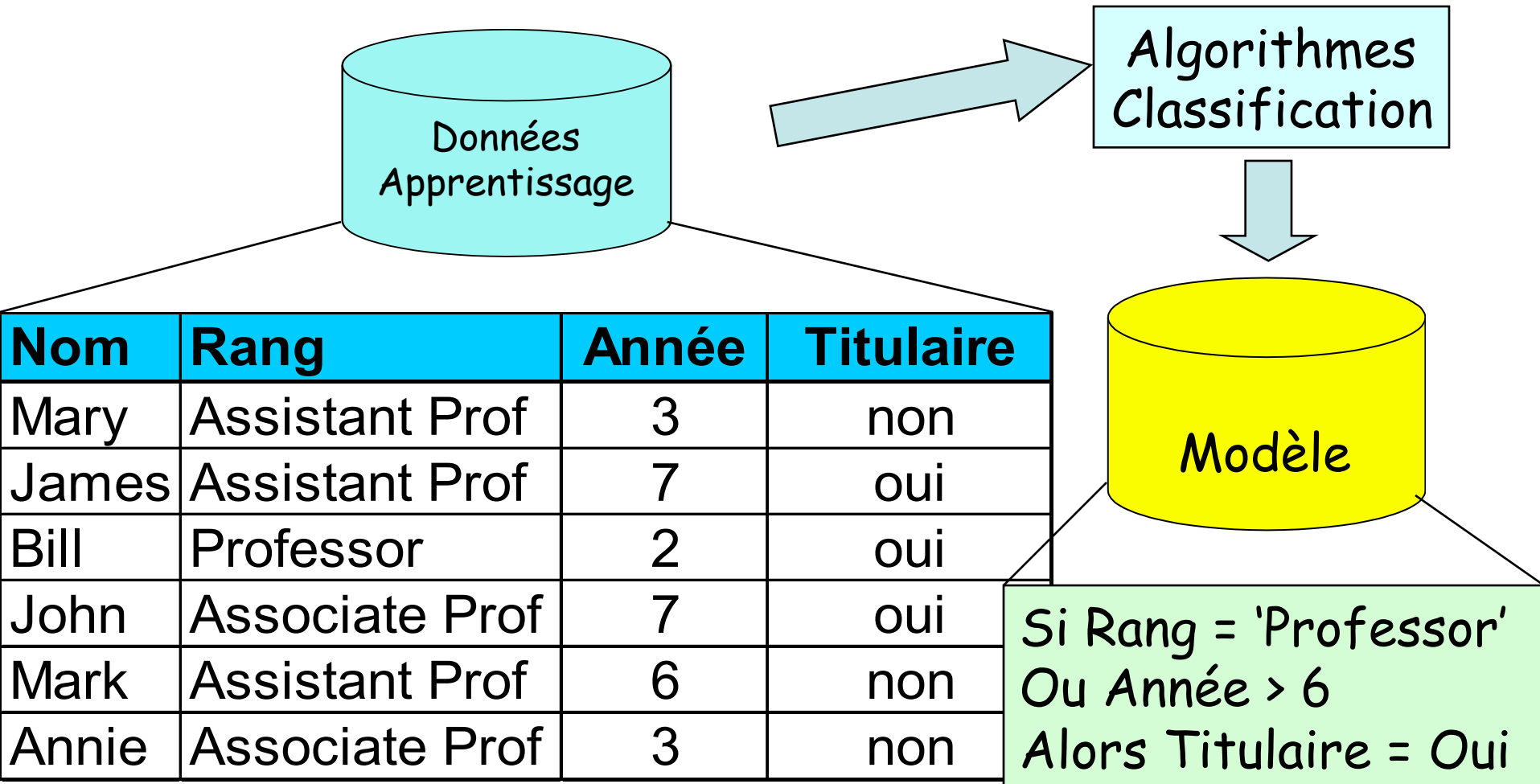
- Diviser les données en  $k$  sous-ensembles
- Utiliser  $k-1$  sous-ensembles comme données d'apprentissage et un sous-ensemble comme données test



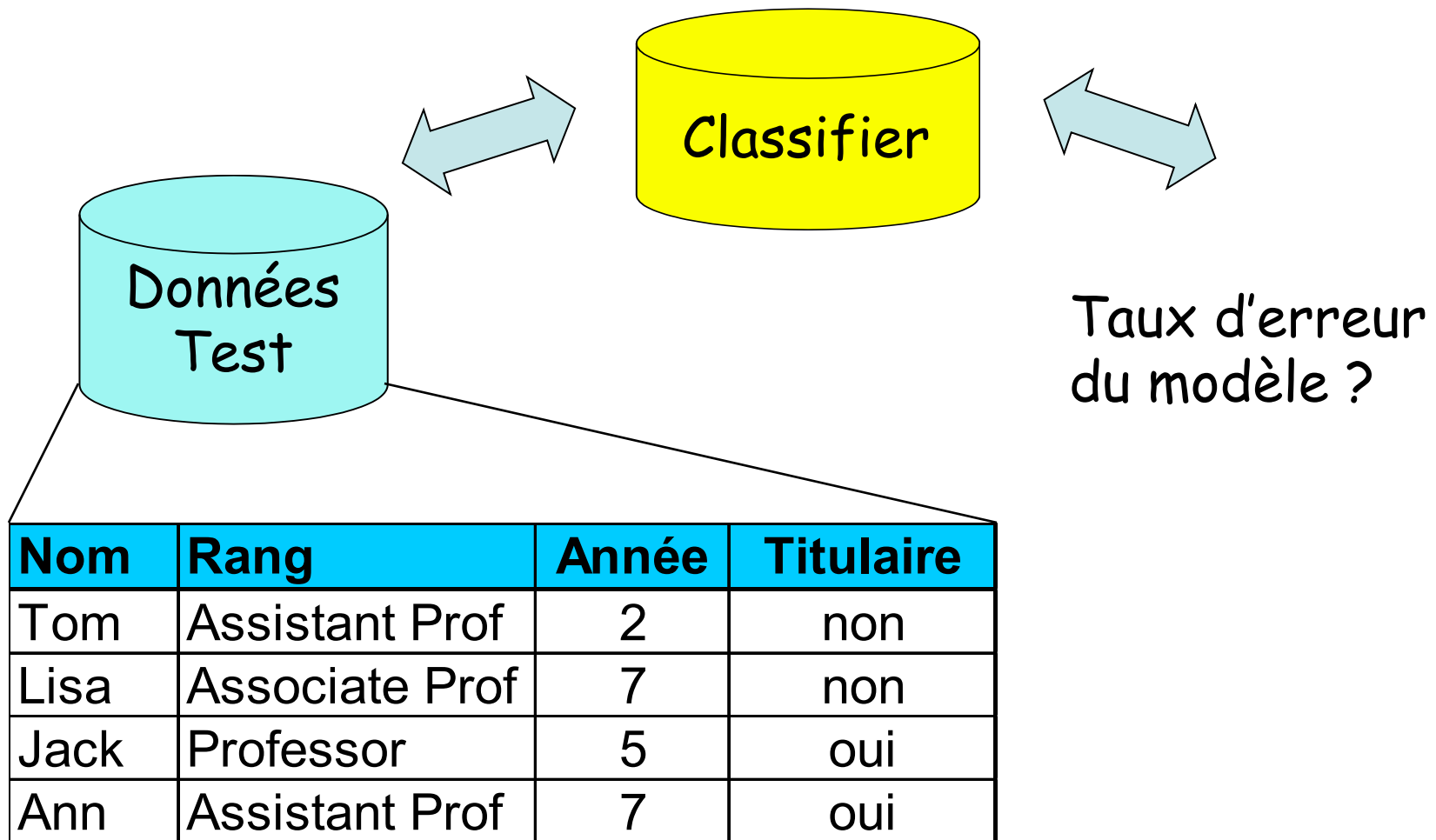
Bootstrapping :  $n$  instances test aléatoires (ensemble de données réduit)



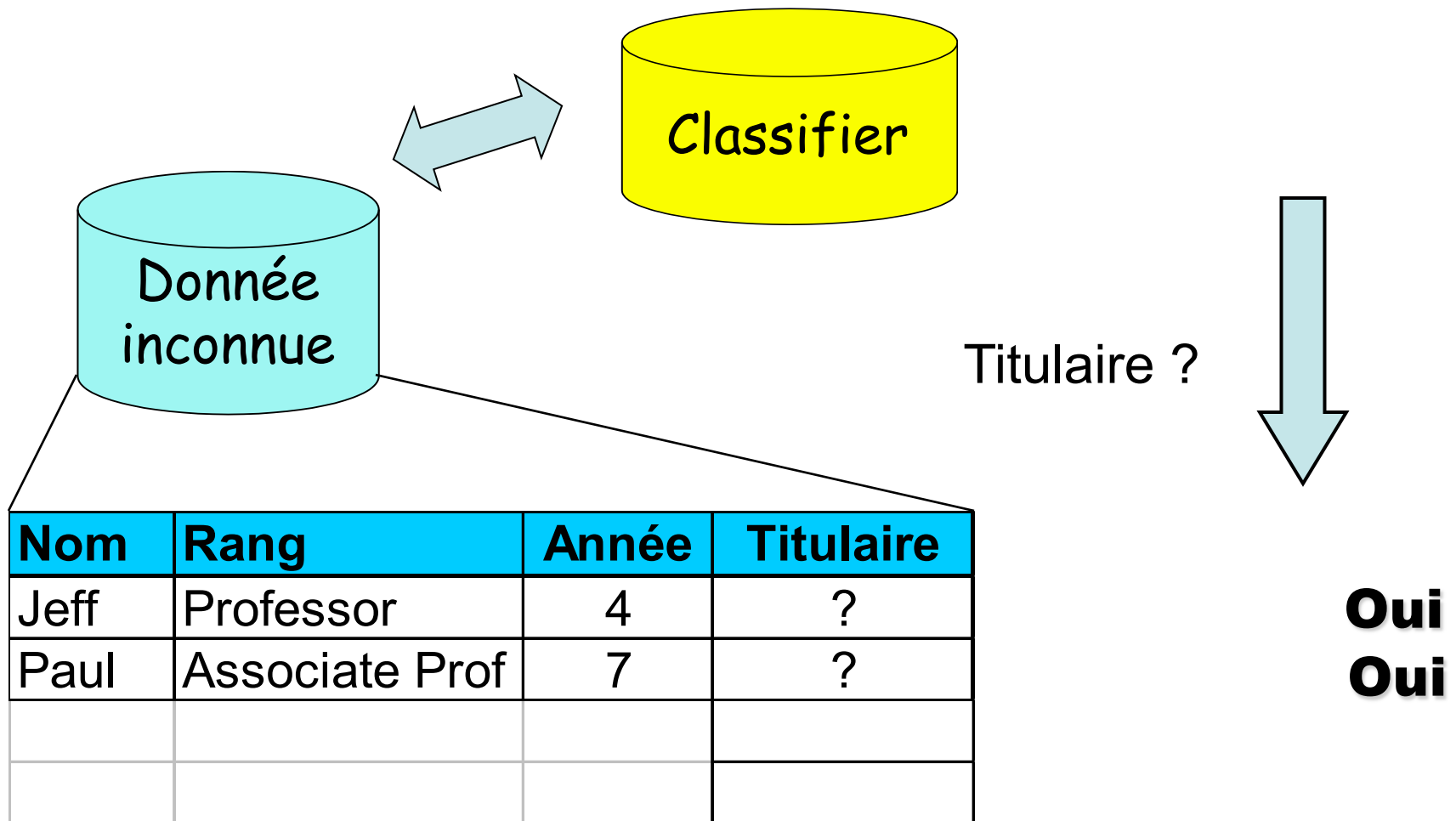
# Exemple : Construction du modèle



# Exemple : Utilisation du modèle



# Exemple : Utilisation du modèle



# Evaluation des méthodes de classification

Taux d'erreur (Accuracy)

Temps d'exécution (construction, utilisation)

Robustesse (bruit, données manquantes,...)

Extensibilité

Interprétabilité

Simplicité

# Méthodes de Classification

- Méthode K-NN (plus proche voisin)
- Arbres de décision
- Réseaux de neurones
- Classification bayésienne
- **Caractéristiques**
  - Apprentissage supervisé (classes connues)

# KNN

# Méthode des plus proches voisins

Méthode dédiée à la classification (k-NN : nearest Neighbors).

**Méthode de raisonnement** à partir de cas : prendre des décisions en recherchant un ou des cas similaires déjà résolus.

**Pas d'étape d'apprentissage** : construction d'un modèle à partir d'un échantillon d'apprentissage (réseaux de neurones, arbres de décision, ...).

Modèle = échantillon d'apprentissage + fonction de distance + fonction de choix de la classe en fonction des classes des voisins les plus proches.

# Algorithme kNN (K-nearest neighbors)

Objectif : affecter une classe à une nouvelle instance

donnée : un échantillon de  $m$  enregistrements classés  $(x, c(x))$

entrée : un enregistrement  $y$

- 1. Déterminer les  $k$  plus proches enregistrements de  $y$
- 2. combiner les classes de ces  $k$  exemples en une classe  $c$

sortie : la classe de  $y$  est  $c(y)=c$



# Algorithme kNN : sélection de la classe

Basé sur l'apprentissage par analogie

Basée sur une notion de distance et Similarité

**Solution simple** : rechercher le cas le plus proche et prendre la même décision (Méthode 1-NN).

**Combinaison des k classes** :

- Heuristique :  $k = \text{nombre d'attributs} + 1$
- Vote majoritaire : prendre la classe majoritaire.
- Vote majoritaire pondéré : chaque classe est pondérée. Le poids de  $c(x_i)$  est inversement proportionnel à la distance  $d(y, x_i)$ .

**Confiance** : Définir une confiance dans la classe attribuée = rapport entre les votes gagnants et le total des votes.

# Algorithme k-NN

Class (X) {

// Training collection  $T = \{X_1, X_2, \dots, X_n\}$

// Predefined classes  $C = \{C_1, C_2, \dots, C_m\}$

// Compute similarities

For  $i=1..N$  similar[i] = Max - distance(X,  $X_i$ );

SortDescending(similar[]);

kNN=Select k nearest neighbors with highest similarity;

// Calculer les scores des classes

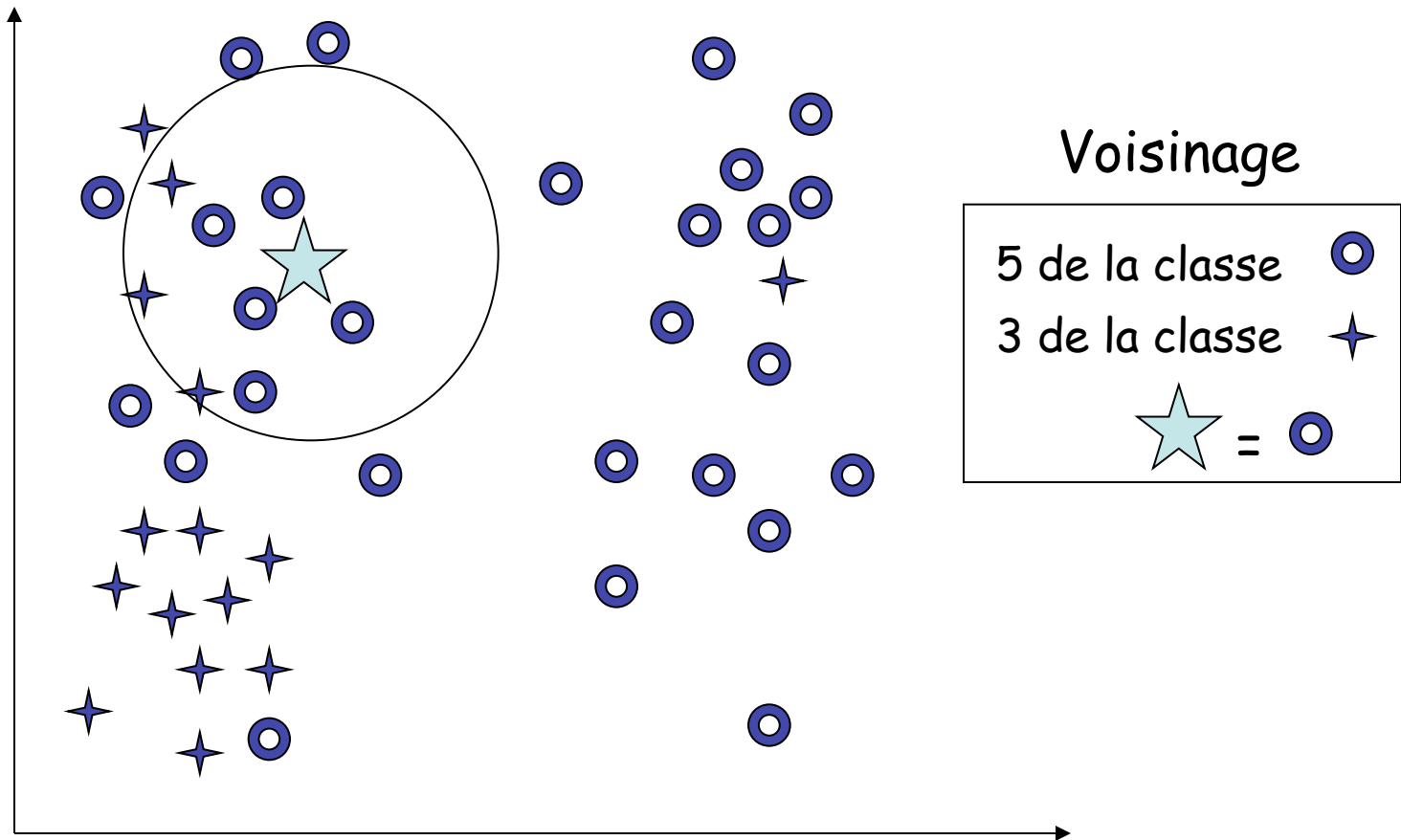
score[ $C_j$ ] = f( $C_j$ , kNN) ;

Class(X) = Class  $C_j$  with highest score;







}

# Exemple

8 plus proches voisins









# Retour sur KNN : Exemple (1)

Customer	Age	Income	No. credit cards	Loyal
John 	35	35K	3	No
Rachel 	22	50K	2	Yes
Hannah 	63	200K	1	No
Tom 	59	170K	1	No
Nellie 	25	40K	4	Yes
David 	37	50K	2	?

# Retour sur KNN : Exemple (2)

$K = 3$

Customer	Age	Income	No. credit cards	Loyal	Distance from David
John 	35	35K	3	No	$\text{sqrt} [(35-37)^2 + (35-50)^2 + (3-2)^2] = 15.16$
Rachel 	22	50K	2	Yes	$\text{sqrt} [(22-37)^2 + (50-50)^2 + (2-2)^2] = 15$
Hannah 	63	200K	1	No	$\text{sqrt} [(63-37)^2 + (200-50)^2 + (1-2)^2] = 152.23$
Tom 	59	170K	1	No	$\text{sqrt} [(59-37)^2 + (170-50)^2 + (1-2)^2] = 122$
Nellie 	25	40K	4	Yes	$\text{sqrt} [(25-37)^2 + (40-50)^2 + (4-2)^2] = 15.74$
David 	37	50K	2	Yes	

# Forces et faiblesses

## Les attributs ont le même poids

- centrer et réduire pour éviter les biais
- certains peuvent être moins classant que d'autres

## Apprentissage paresseux

- rien n'est préparé avant le classement
- tous les calculs sont fait lors du classement
- nécessité de technique d'indexation pour large BD

## Calcul du score d'une classe

- peut changer les résultats; variantes possibles

# ARBRES DE DÉCISION

# 3. Arbres de décision

## Définition

- Arbre permettant de classer des enregistrements par division hiérarchiques en sous-classes
  - un nœud représente une classe de plus en plus fine depuis la racine
  - un arc représente un prédicat de partitionnement de la classe source
- Un attribut sert d'étiquette de classe (attribut cible à prédire), les autres permettant de partitionner

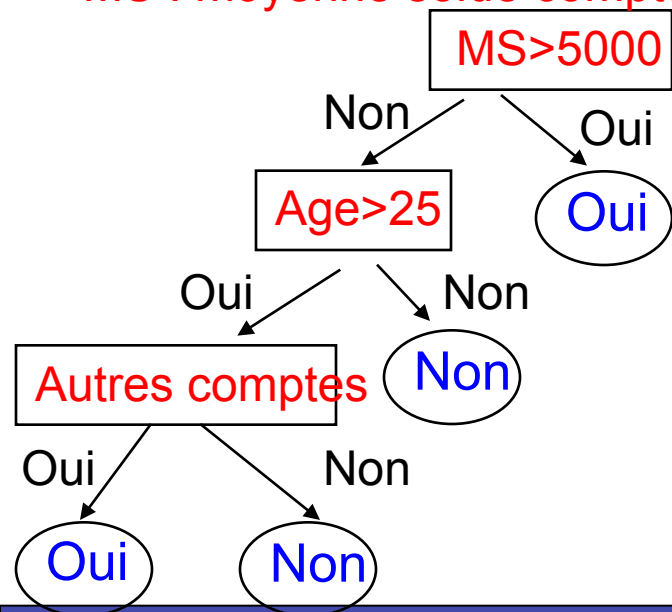


# Arbres de décision

- 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

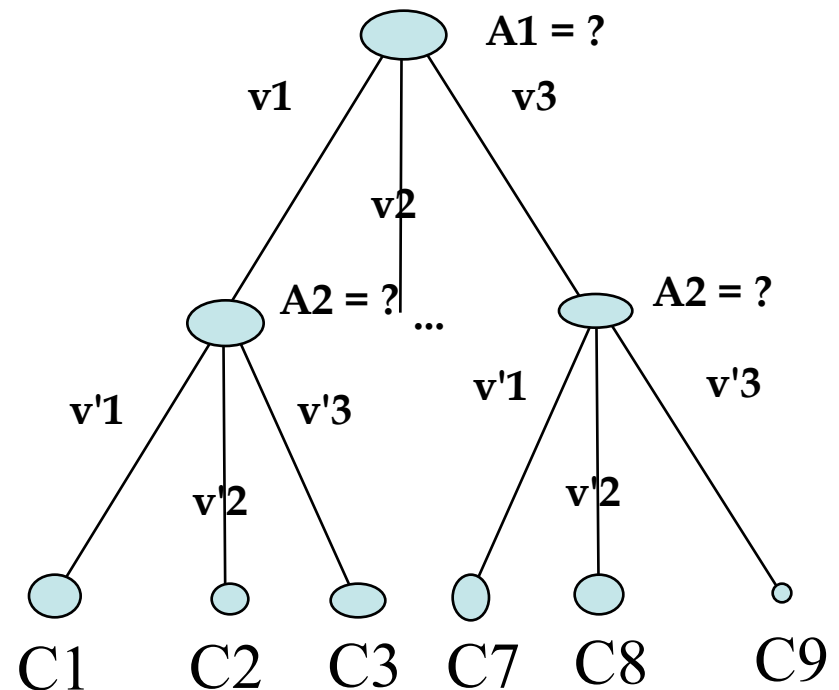
# Génération de l'arbre

## Objectif:

- obtenir des classes homogènes
- couvrir au mieux les données

Comment choisir les attributs ( $A_i$ ) ?

Comment isoler les valeurs discriminantes ( $v_j$ ) ?

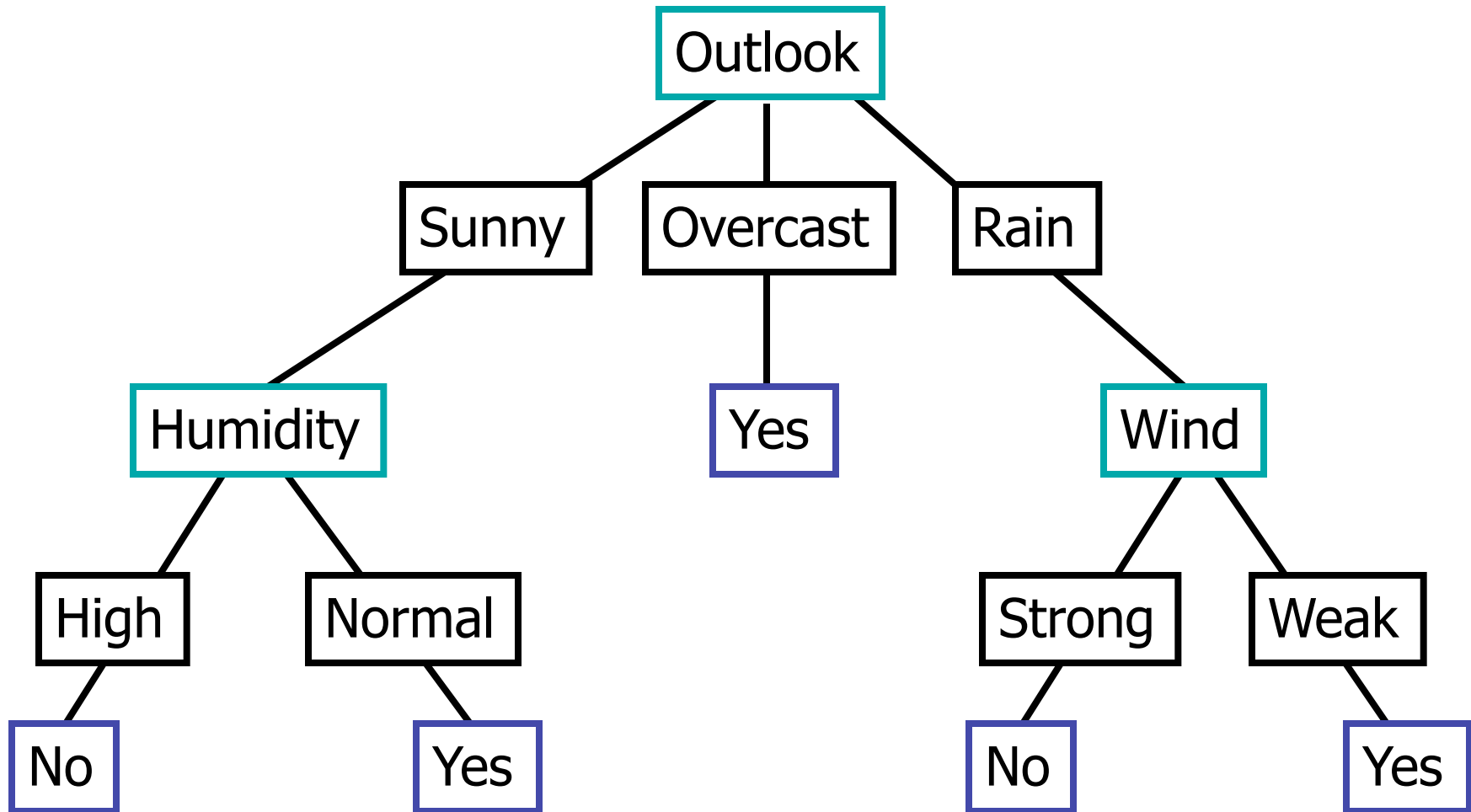


# Arbre de décision - Exemple

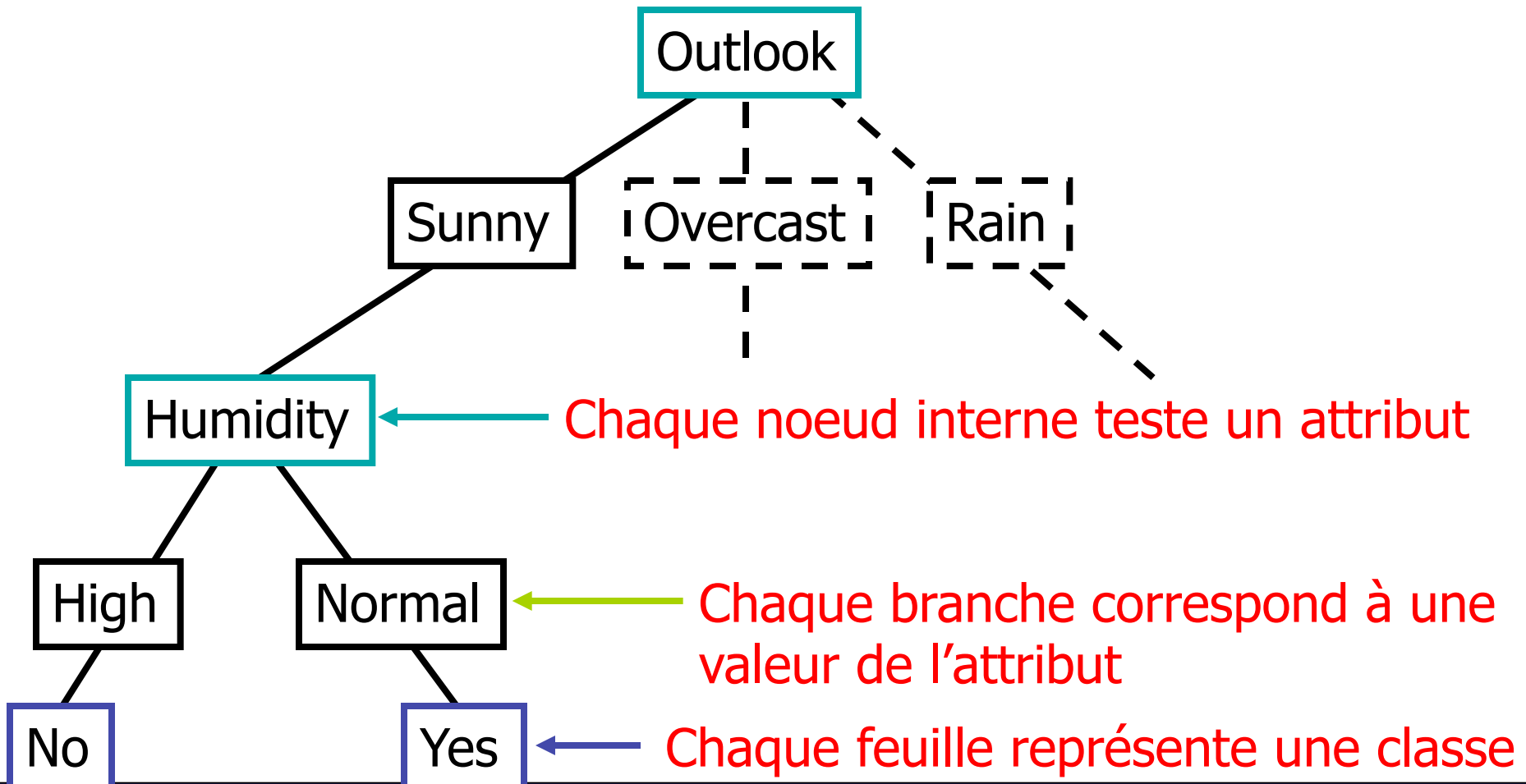
**Ensemble  
d'apprentissage**

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

# Arbre de décision - Exemple



# Exemple – Jouer au tennis ?



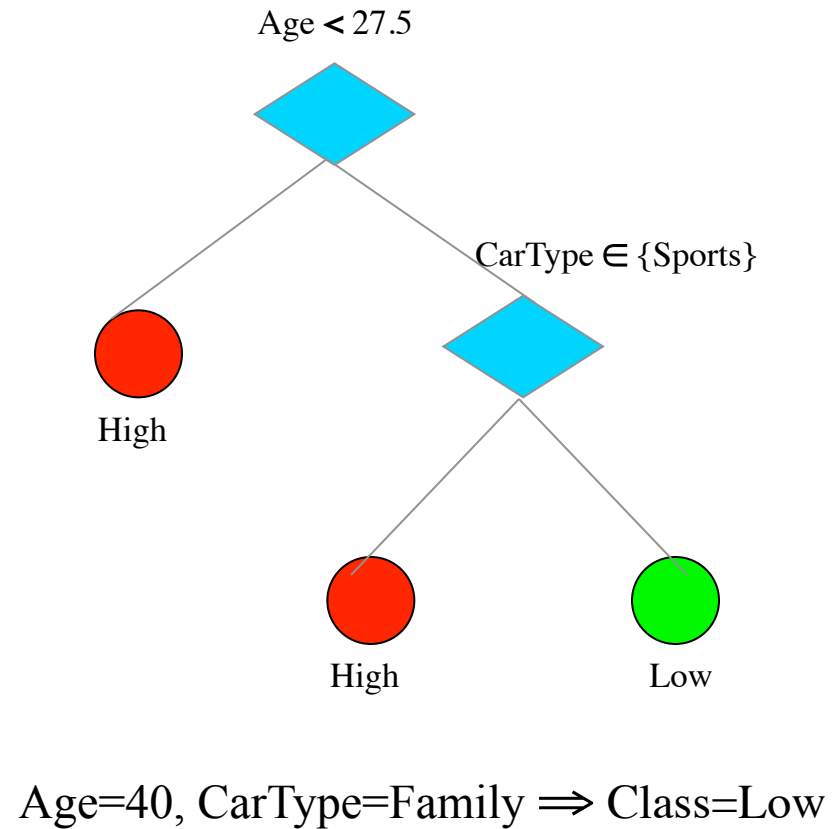
# Arbres de décision – Exemple

Risque - Assurances

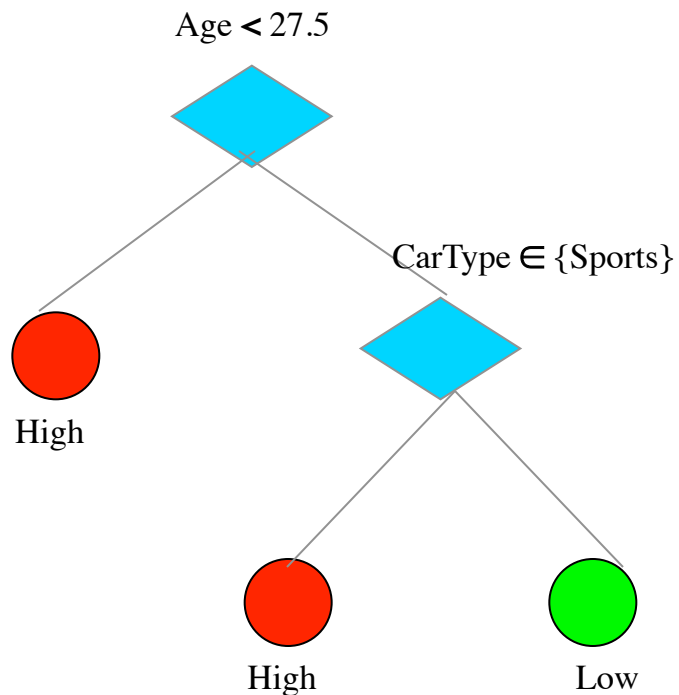
Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

Numérique

Enumératif



# Des arbres de décision aux règles



1)  $\text{Age} < 27.5 \Rightarrow \text{High}$

2)  $\text{Age} \geq 27.5$  and  
 $\text{CarType} = \text{Sports} \Rightarrow \text{High}$

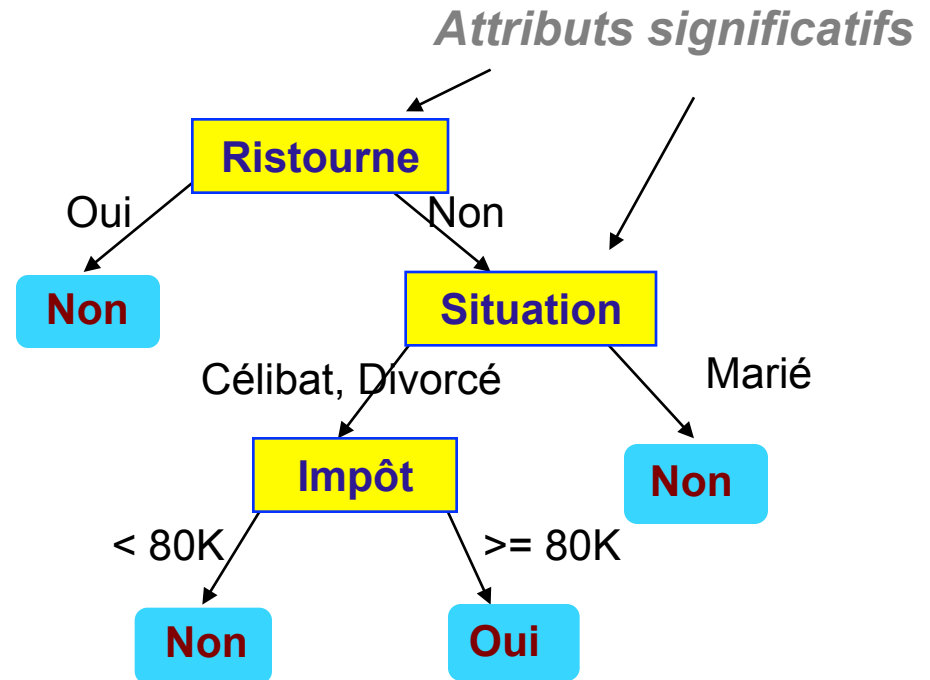
3)  $\text{Age} \geq 27.5$  and  
 $\text{CarType} \neq \text{Sports} \Rightarrow \text{Low}$

# Arbres de décision – Exemple

## Détection de fraudes fiscales

énumératif      énumératif      numérique      classe

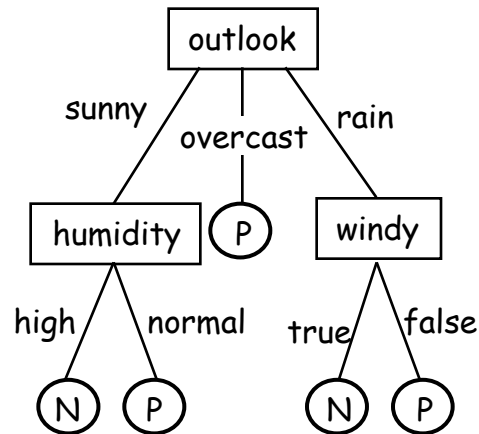
Id	Ristourne	Situation famille	Impôt revenu	Fraude
1	Oui	Célibat.	125K	Non
2	Non	Marié	100K	Non
3	Non	Célibat.	70K	Non
4	Oui	Marié	120K	Non
5	Non	Divorcé	95K	Oui
6	Non	Marié	60K	Non
7	Oui	Divorcé	220K	Non
8	Non	Célibat.	85K	Oui
9	Non	Marié	75K	Non
10	Non	Célibat.	90K	Oui



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



# De l'arbre de décision aux règles de classification



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

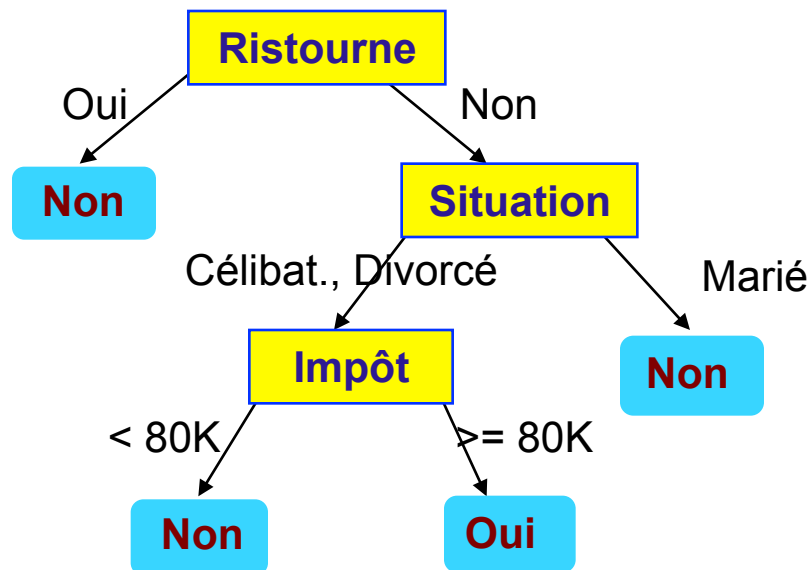
Le nœud terminal représente la classe prédite

Les règles sont généralement plus faciles à comprendre que les arbres

**Si** outlook=sunny  
**Et** humidity=normal  
**Alors** play tennis

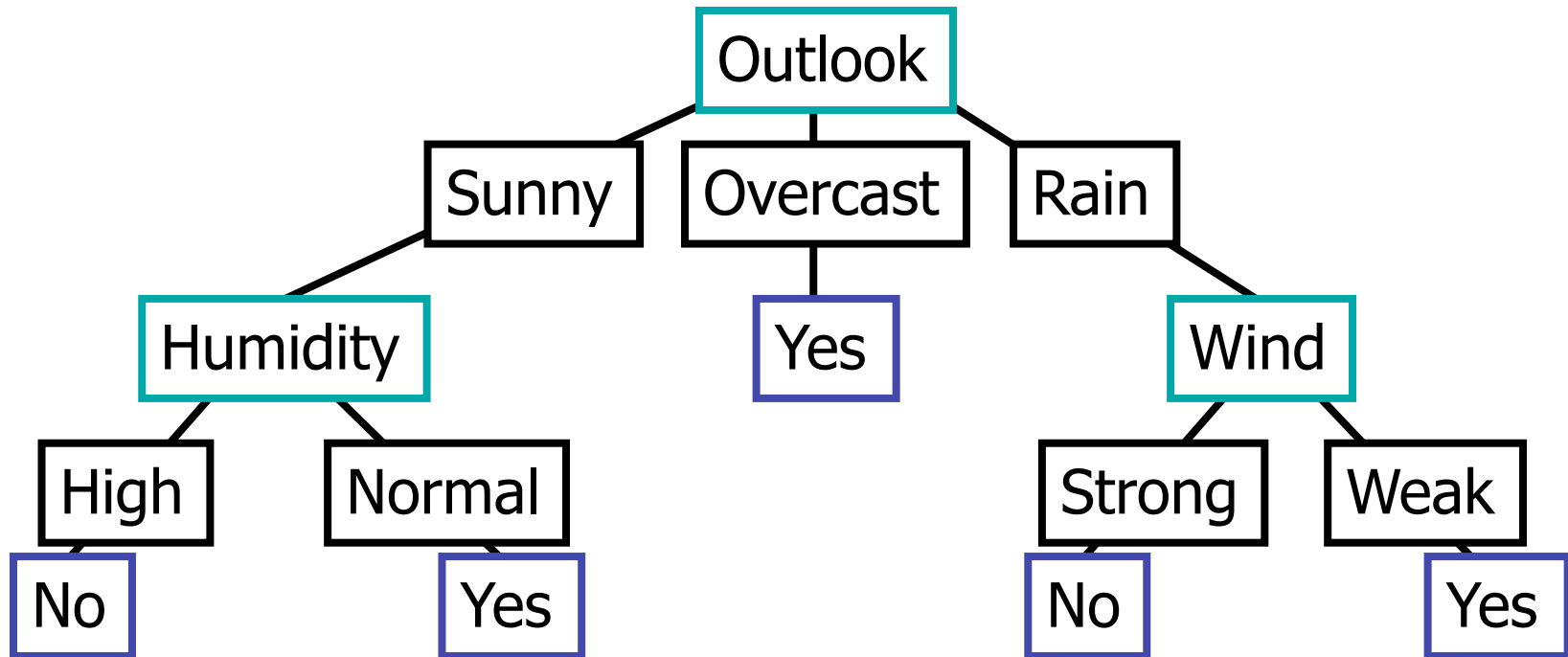
# Des arbres de décision aux règles

**Arbre de décision** = Système de règles exhaustives et mutuellement exclusives



- 1) Ristourne = Oui  $\Rightarrow$  Non
- 2) Ristourne = Non et Situation in {Célibat., Divorcé} et Impôt < 80K  $\Rightarrow$  Non
- 3) Ristourne = Non et Situation in {Célibat., Divorcé} et Impôt  $\geq$  80K  $\Rightarrow$  Oui
- 4) Ristourne = Non et Situation in {Marié}  $\Rightarrow$  Non

# Des arbres de décision aux règles



- $R_1$ : If (Outlook=Sunny)  $\wedge$  (Humidity=High) Then PlayTennis=No  
 $R_2$ : If (Outlook=Sunny)  $\wedge$  (Humidity=Normal) Then PlayTennis=Yes  
 $R_3$ : 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 :

## 1. Construction de l'arbre

- Arbre peut atteindre une taille élevée

## 2. Elaguer l'arbre (Pruning)

- Identifier et supprimer les branches qui représentent du “bruit” → Améliorer le taux d'erreur

# Procédure de construction (1)

recherche à chaque niveau de l'attribut le plus discriminant

## Partition (nœud P)

- si (tous les éléments de P sont dans la même classe) alors retour;
- pour chaque attribut A faire
  - évaluer la qualité du partitionnement sur A;
- utiliser le meilleur partitionnement pour diviser P en  $P_1, P_2, \dots, P_n$
- pour  $i = 1$  à  $n$  faire Partition( $P_i$ );

# Procédure de Construction (2)

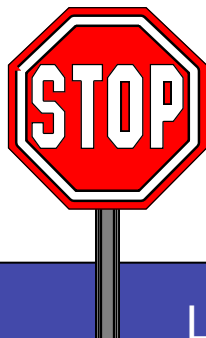
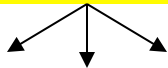
## Processus récursif

- L'arbre commence à un nœud représentant toutes les données
- Si les objets sont de la même classe, alors le nœud devient une feuille étiqueté par le nom de la classe.
- Sinon, sélectionner les attributs qui séparent le mieux les objets en classes homogènes => Fonction de qualité



Class

Atr=?



La récursion s'arrête quand:

- Les objets sont assignés à une classe homogène
- Il n'y a plus d'attributs pour diviser,
- Il n'y a pas d'objet avec la valeur d'attribut

# Choix de l'attribut de division

## Différentes mesures introduites

- il s'agit d'ordonner le désordre
- des indicateurs basés sur la théorie de l'information

## Choix des meilleurs attributs et valeurs

- les meilleurs tests

## Possibilité de retour arrière

- élaguer les arbres résultants (classes inutiles)
- revoir certains partitionnements (zoom, réduire)

# Mesure de qualité

La mesure est appelé fonction de qualité

- Goodness Function en anglais

Varie selon l'algorithme :

- Gain d'information (ID3/C4.5)
  - Suppose des attributs nominaux (discrets)
  - Peut-être étendu à des attributs continus
- Gini Index
  - Suppose des attributs continus
  - Suppose plusieurs valeurs de division pour chaque attribut
  - Peut-être étendu pour des attributs nominaux



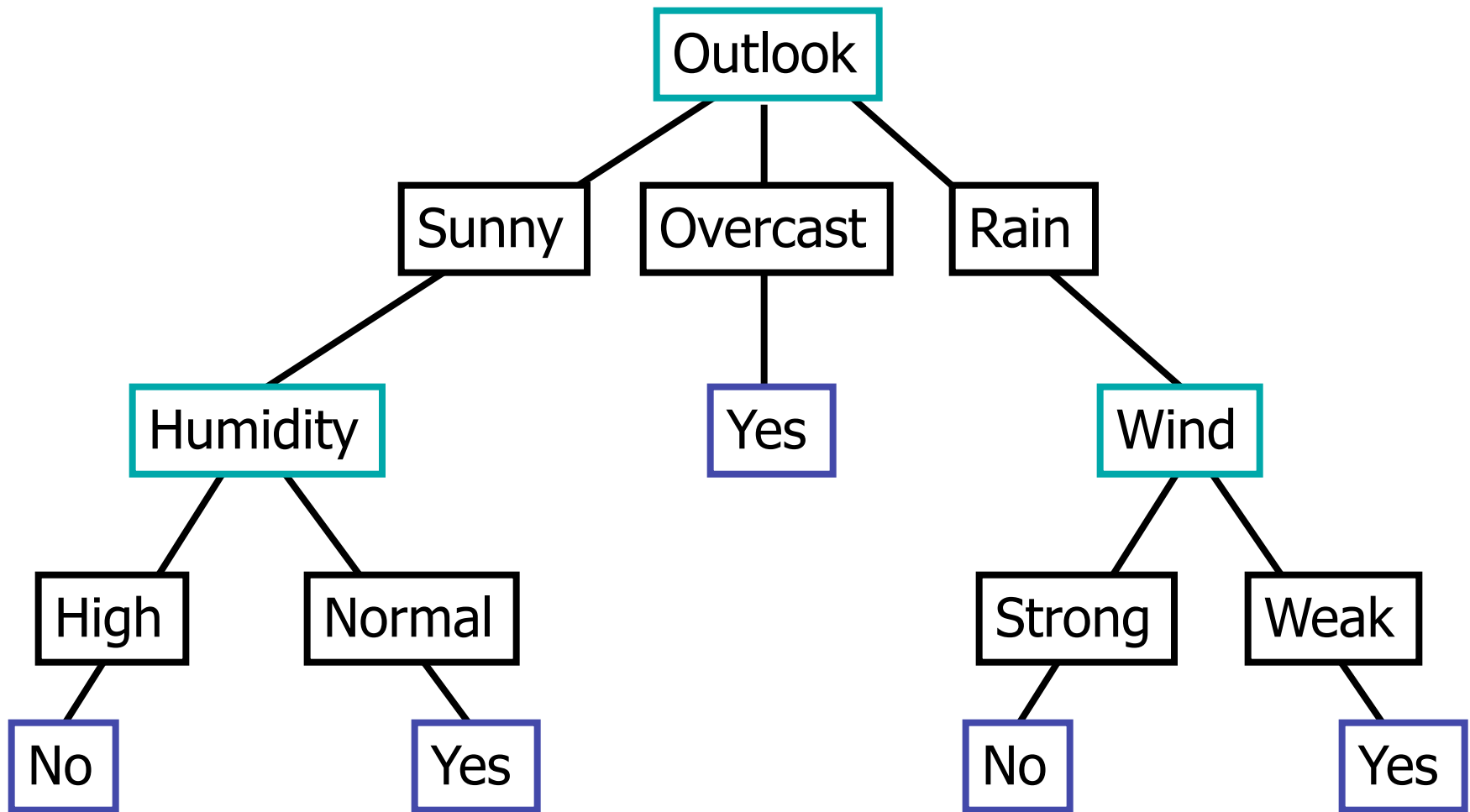
# Exemple : Jouer au tennis ?

**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

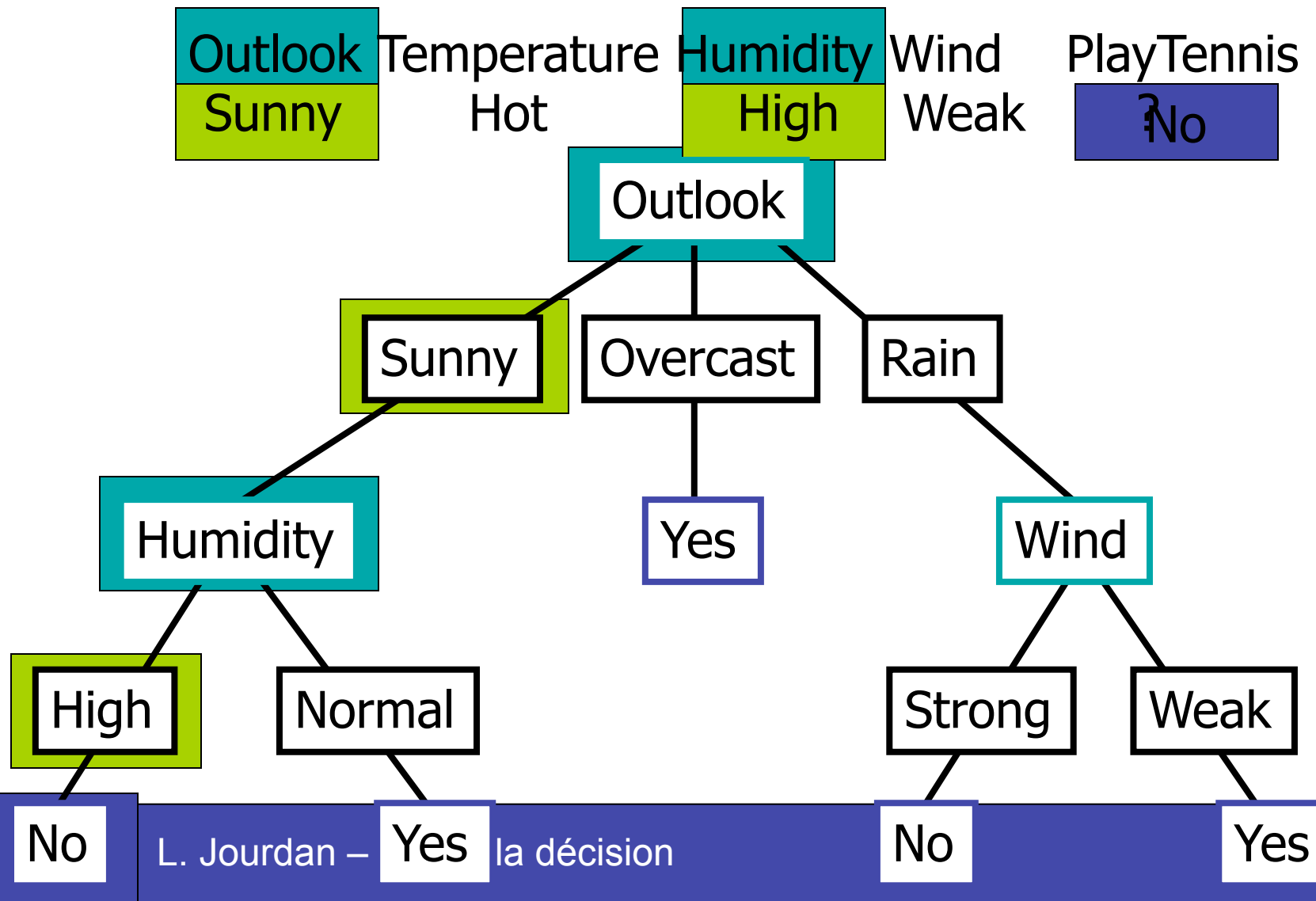
# Arbre de décision obtenu avec ID3 (Quinlan 86)

42



# Arbre de décision obtenu avec ID3 (Quinlan 86)

43



# Algorithmes pour les arbres de décision

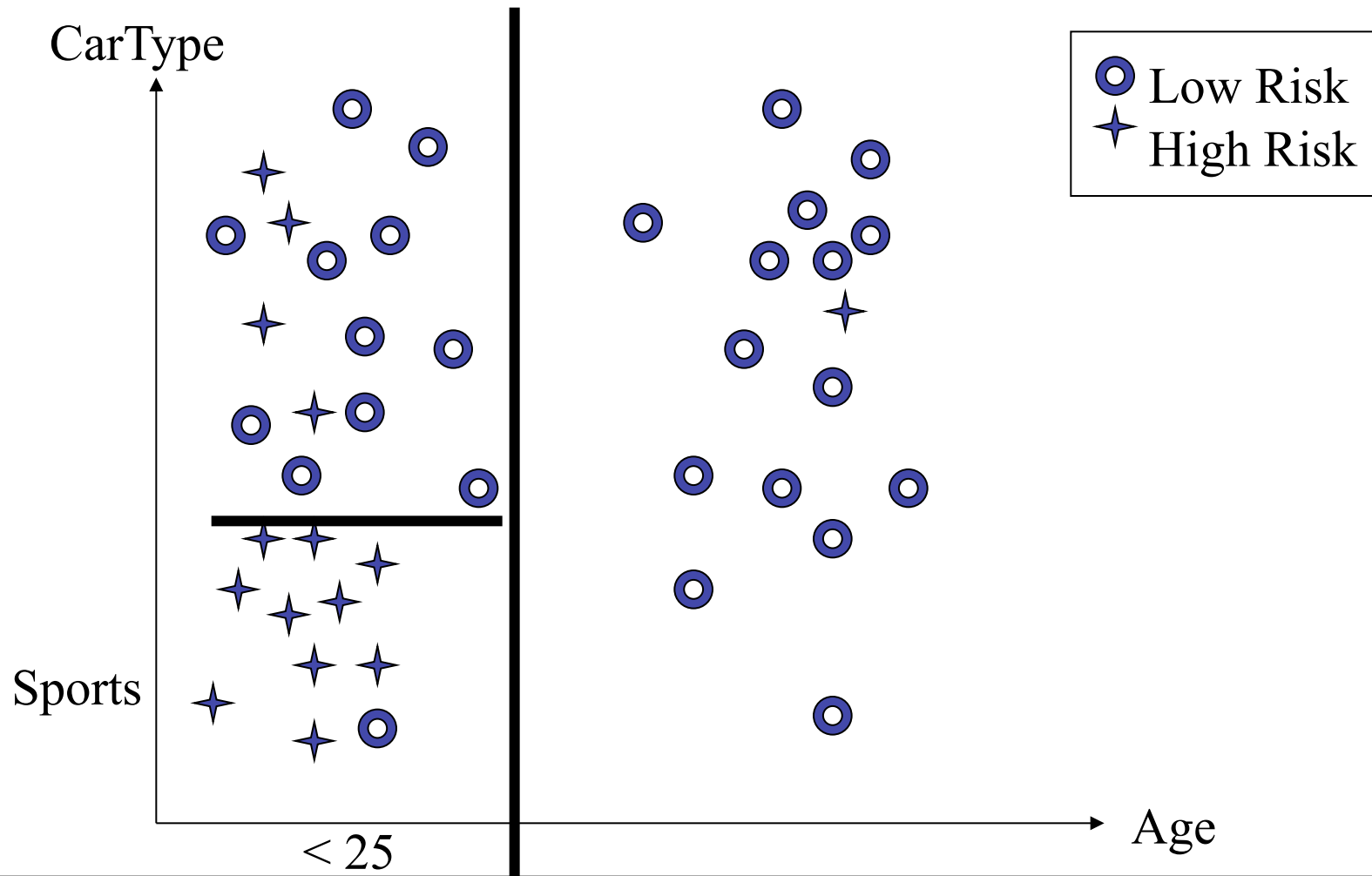
## Algorithme de base

- Construction récursive d'un arbre de manière “diviser-pour-régner” descendante
- Attributs considérés énumératifs
- Glouton (piégé par les optima locaux)

## Plusieurs variantes : ID3, C4.5, CART, CHAID

- Différence principale : mesure de sélection d'un attribut – critère de branchement (split)
- Ex : CART : 2 partitions par nœuds

# Bonne sélection et branchement ?



# Gain d'information

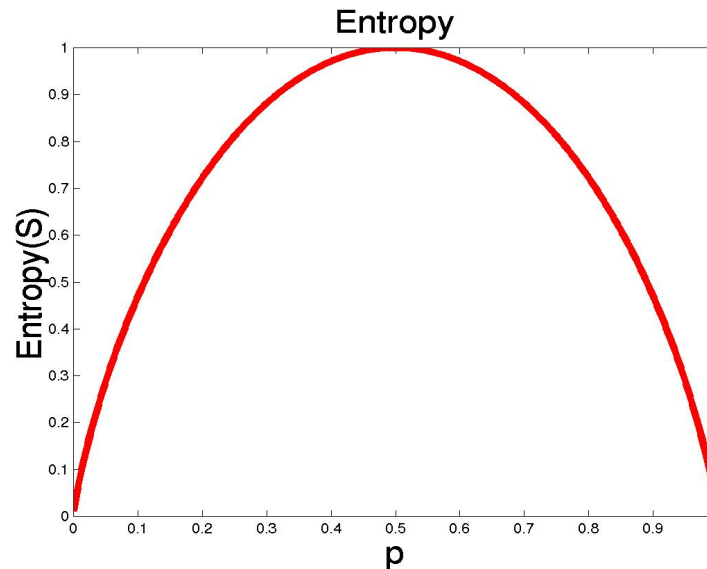
Sélectionner l'attribut avec le plus grand gain d'information

Soient P et N deux classes et S un ensemble d'instances avec p éléments de P et n éléments de N

L'information nécessaire pour déterminer si une instance prise au hasard fait partie de P ou N est (entropie) :

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

# Entropie



Ex : var. booléenne  $X=1$   
Avec probabilité  $p$

S est l'ensemble d'apprentissage

$p_+$  est la proportion d'exemples positifs (P)

$p_-$  est la proportion d'exemples négatifs (N)

Entropie mesure l'impureté de S

- $\text{Entropie}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$

# Gain d'information

Soient les ensembles  $\{S_1, S_2, \dots, S_v\}$  formant une partition de l'ensemble  $S$ , en utilisant l'attribut  $A$

Toute partition  $S_i$  contient  $p_i$  instances de  $P$  et  $n_i$  instances de  $N$

L'entropie, ou l'information nécessaire pour classer les instances dans les sous-arbres  $S_i$  est (entropie conditionnelle classe/attribut  $A$ ):

$$E(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

Le gain d'information par rapport au branchement sur  $A$  est

$$Gain(A) = I(p, n) - E(A)$$

Choisir l'attribut qui maximise le gain  $\rightarrow$  besoin d'information minimal (recherche "greedy" – gloutonne)



# Exemple: Partitions de boules (1)

## Partition selon A1 (densité)

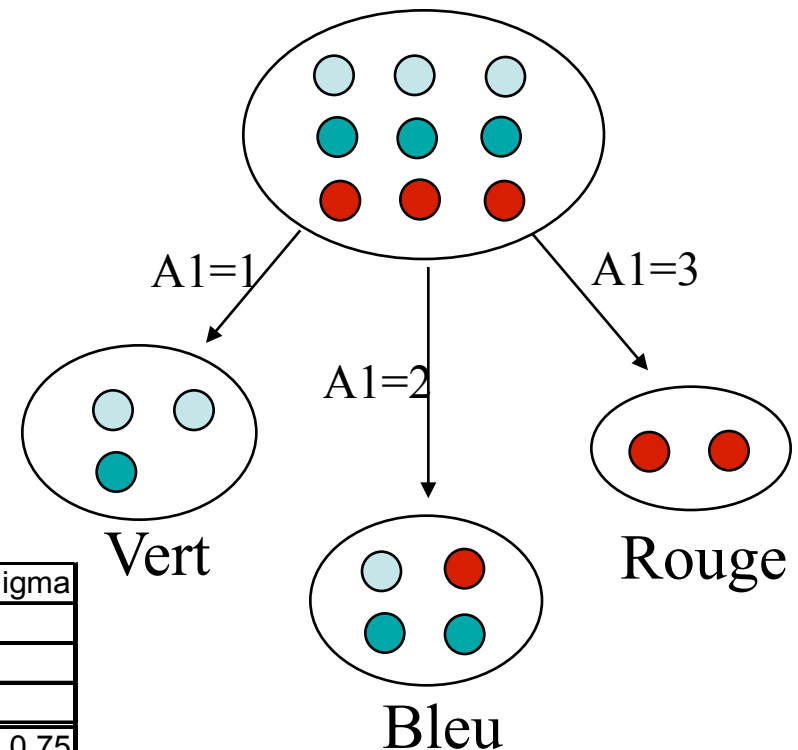
### Indice d'impureté :

- $i(N) = \sum_i^k \sum_j^k \{ p_i * p_j \}$  avec  $i \neq j$
- $P_i$  est la proportion d'individus de la classe  $i$  dans  $N$ .

### Entropie d'un segment $s$ :

- $E(N) = - \sum_i p_i \log_2(p_i)$

Proportion	C1	C2	C3	Sigma
Vert	0,67	0,25	0,00	
Bleu	0,33	0,50	0,00	
Rouge	0,00	0,25	1,00	
Entropie	0,92	1,00	0,00	0,75
N2 log2(N2)	-0,39	-0,50	0,00	
N3 log2(N3)	-0,53	-0,50	0,00	
N4 log2(N4)	0,00	0,00	0,00	
Impureté	0,44444444	0,625	0	0,43

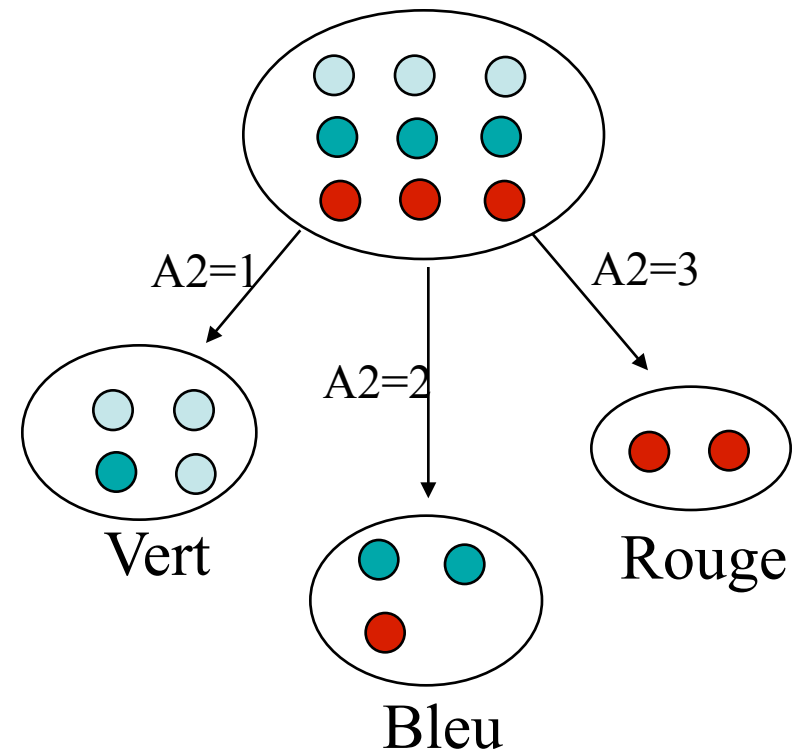


# Exemple: Partitions de boules (2)

## Partition selon A2

- Position et 4 au plus par partition

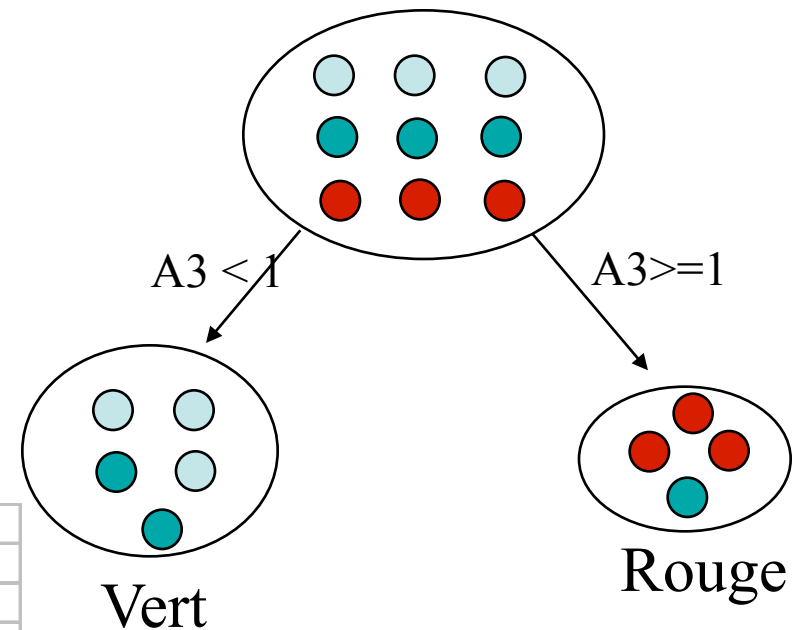
Proportion	C1	C2	C3	Sigma
Vert	0,75	0,00	0,00	
Bleu	0,25	0,67	0,00	
Rouge	0,00	0,33	1,00	
Entropie	0,81	0,39	0,00	0,49
N2 log2(N2)	-0,31	0,00	0,00	
N3 log2(N3)	-0,50	-0,39	0,00	
N4 log2(N4)	0,00	0,00	0,00	
Impureté	0,375	0,44444444	0	0,31



# Exemple: Partitions de boules (3)

## Partition selon A3

- Poids



Proportion	C1	C2	Sigma
Vert	0,60	0,00	
Bleu	0,40	0,25	
Rouge	0,00	0,75	
Entropie	0,97	0,50	0,76
N2 log2(N2)	-0,44	0,00	
N3 log2(N3)	-0,53	-0,50	
N4 log2(N4)	0,00	0,00	
Impureté	0,48	0,375	0,43

# Gain d'information - Exemple

Hypothèses :

Classe P : jouer\_tennis = “oui”

Classe N : jouer\_tennis = “non”

Information nécessaire pour  
classer un exemple donné est :

$$I(p, n) = I(9, 5) = 0.940$$

# Gain d'information - Exemple

Calculer l'entropie pour  
l'attribut outlook :

outlook	$p_i$	$n_i$	$I(p_i, n_i)$
sunny	2	3	0,971
overcast	4	0	0
rain	3	2	0,971

On a 
$$E(outlook) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

Alors 
$$Gain(outlook) = I(9,5) - E(outlook) = 0.246$$

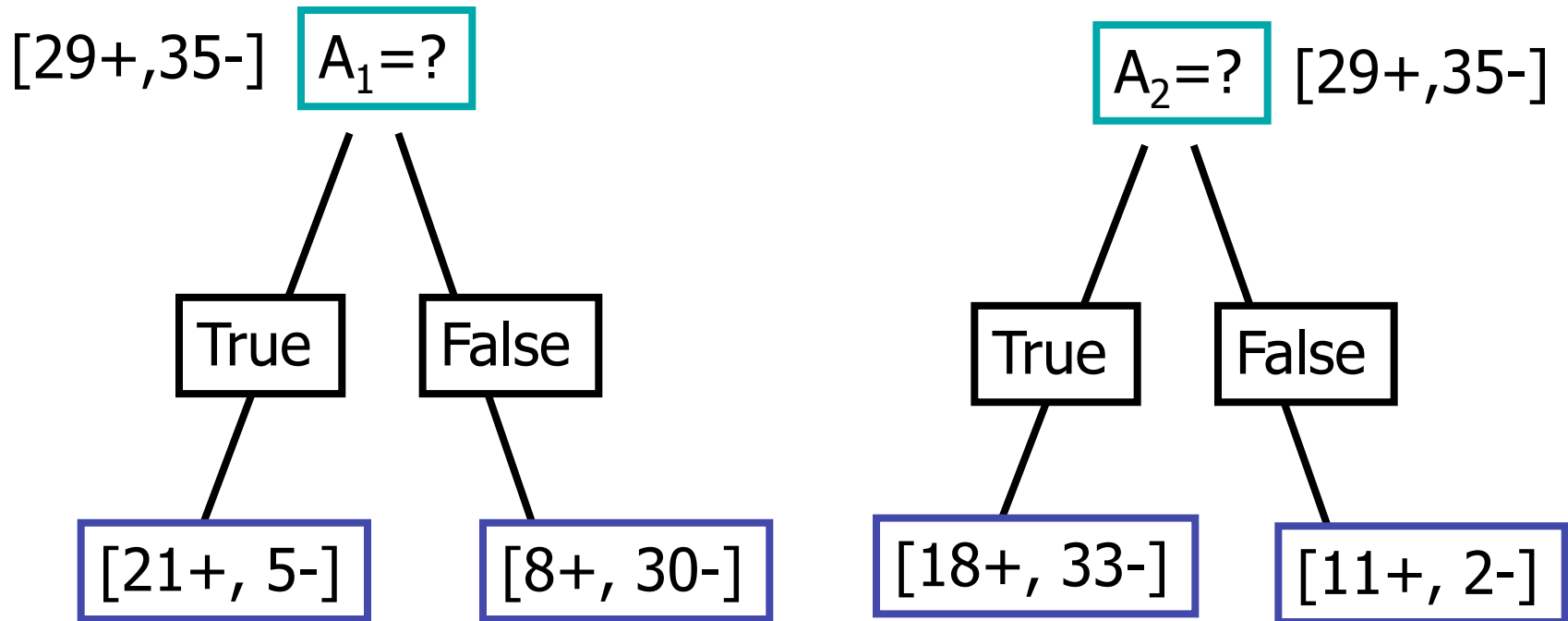
De manière similaire

$$Gain(temperature) = 0.029$$

$$Gain(humidity) = 0.151$$

$$Gain(windy) = 0.048$$

# Quel Attribut est "meilleur" ?

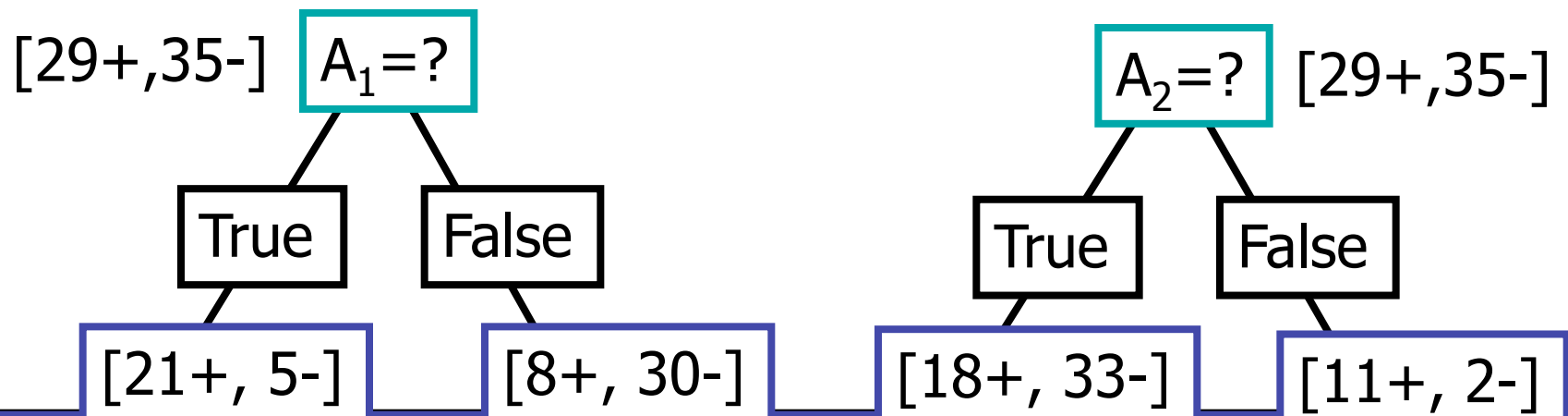


# Gain d'information - Exemple

Gain(S,A) : réduction attendue de l'entropie dûe au branchement de S sur l'attribut A

$$\text{Gain}(S,A) = \text{Entropie}(S) - \sum_{v \in \text{values}(A)} |S_v|/|S| \text{Entropie}(S_v)$$

$$\begin{aligned} \text{Entropie}([29+, 35-]) &= -29/64 \log_2 29/64 - 35/64 \log_2 35/64 \\ &= 0.99 \end{aligned}$$

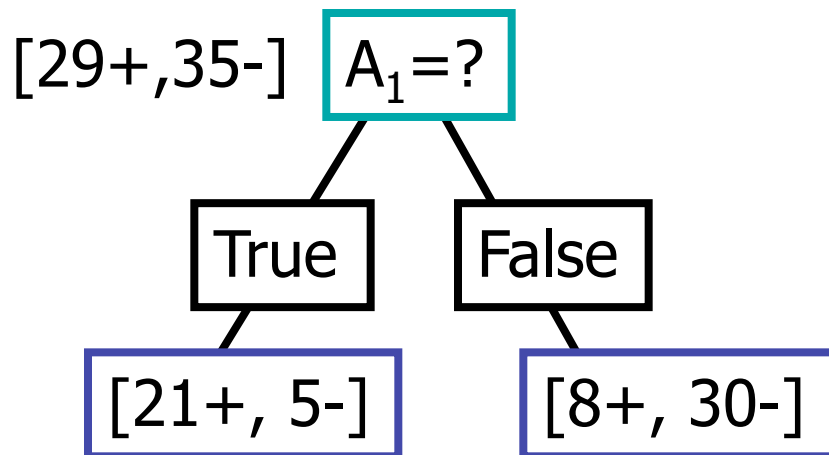


# Gain d'information - Exemple

$$\text{Entropie}([21+, 5-]) = 0.71$$

$$\text{Entropie}([8+, 30-]) = 0.74$$

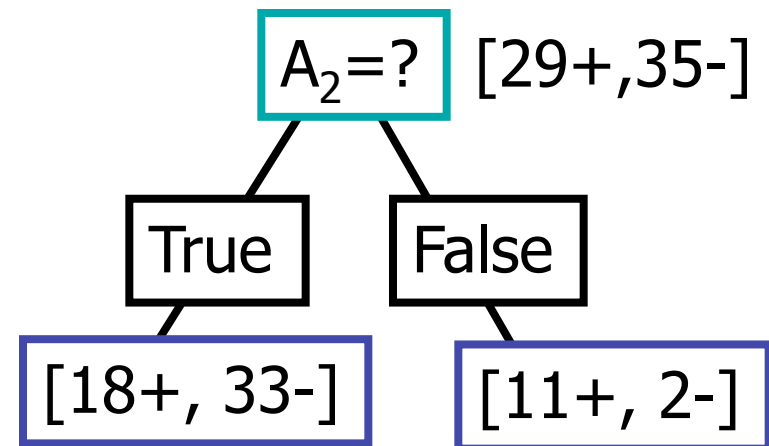
$$\begin{aligned} \text{Gain}(S, A_1) &= \text{Entropie}(S) \\ &\quad - 26/64 * \text{Entropie}([21+, 5-]) \\ &\quad - 38/64 * \text{Entropie}([8+, 30-]) \\ &= 0.27 \end{aligned}$$



$$\text{Entropie}([18+, 33-]) = 0.94$$

$$\text{Entropie}([11+, 2-]) = 0.62$$

$$\begin{aligned} \text{Gain}(S, A_2) &= \text{Entropie}(S) \\ &\quad - 51/64 * \text{Entropie}([18+, 33-]) \\ &\quad - 13/64 * \text{Entropie}([11+, 2-]) \\ &= 0.12 \end{aligned}$$

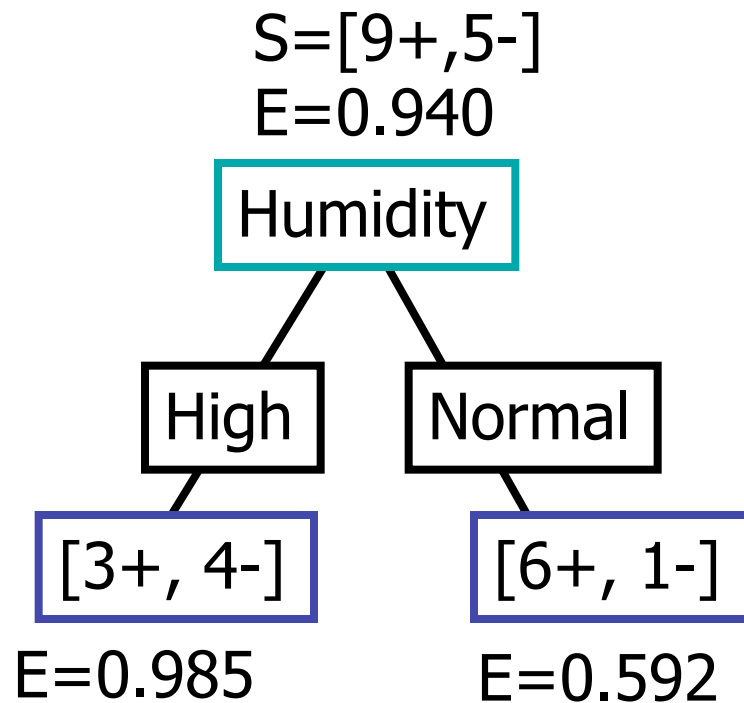




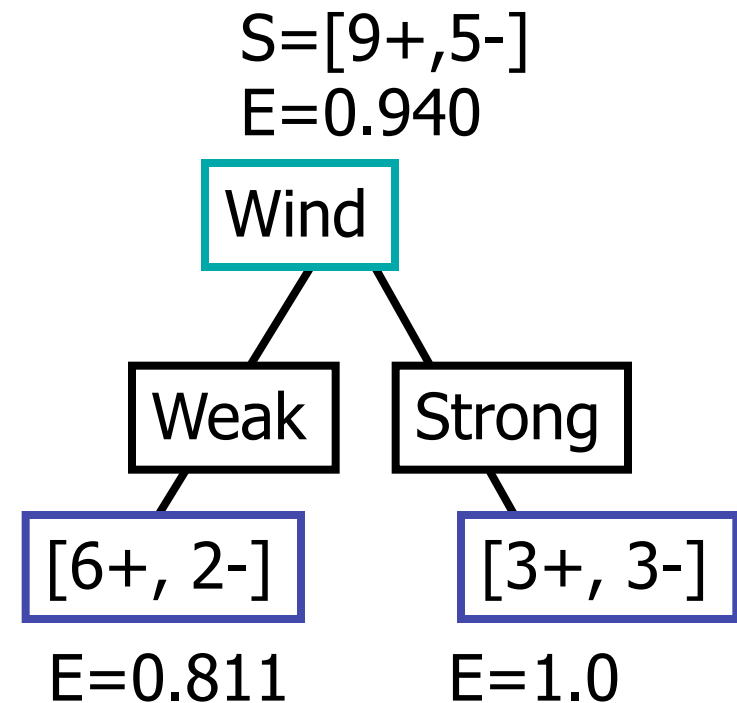
# Exemple d'apprentissage

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

# Sélection de l'attribut suivant

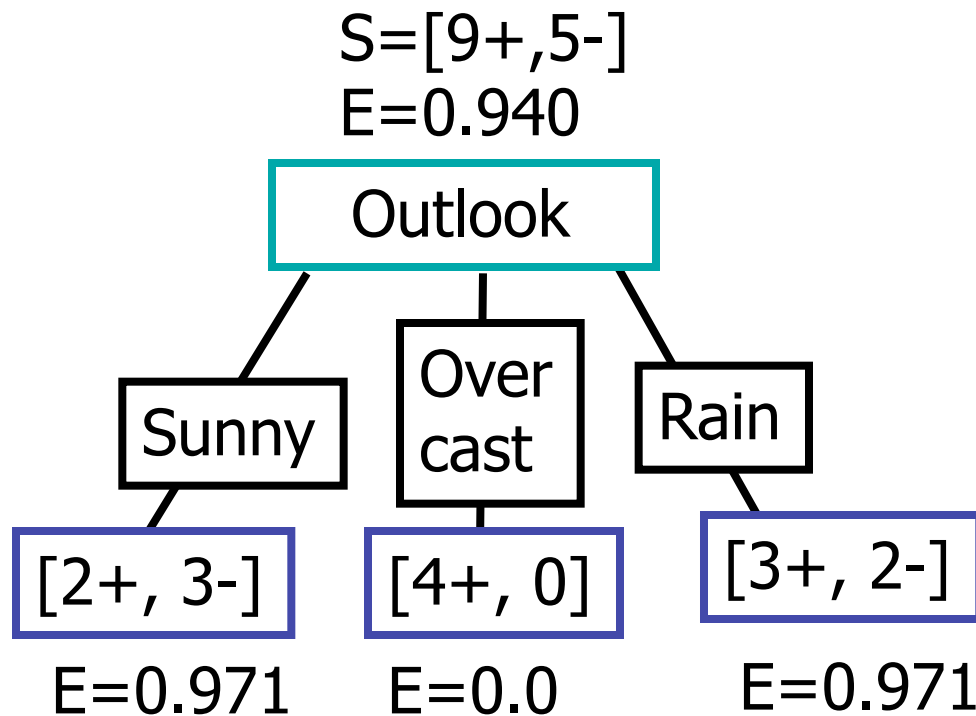


$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= 0.940 - (7/14) * 0.985 \\ &\quad - (7/14) * 0.592 \\ &= 0.151 \end{aligned}$$



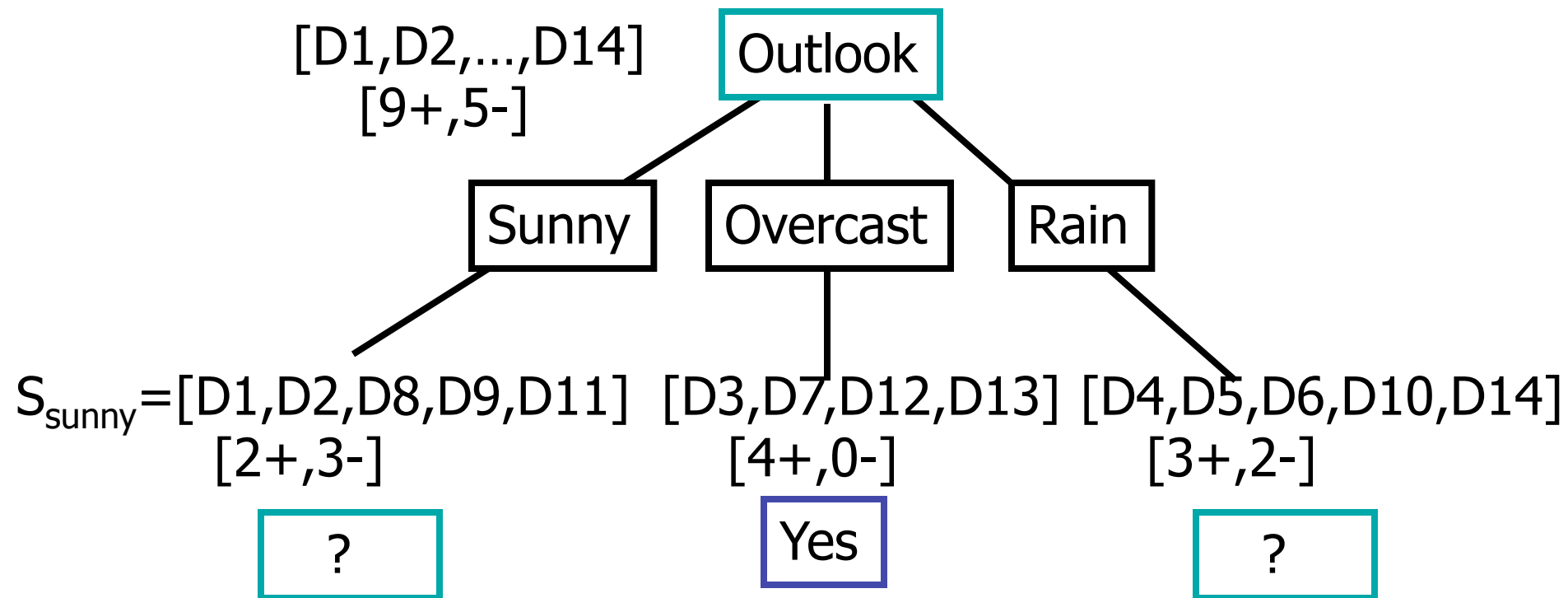
$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= 0.940 - (8/14) * 0.811 \\ &\quad - (6/14) * 1.0 \\ &= 0.048 \end{aligned}$$

# Sélection de l'attribut suivant



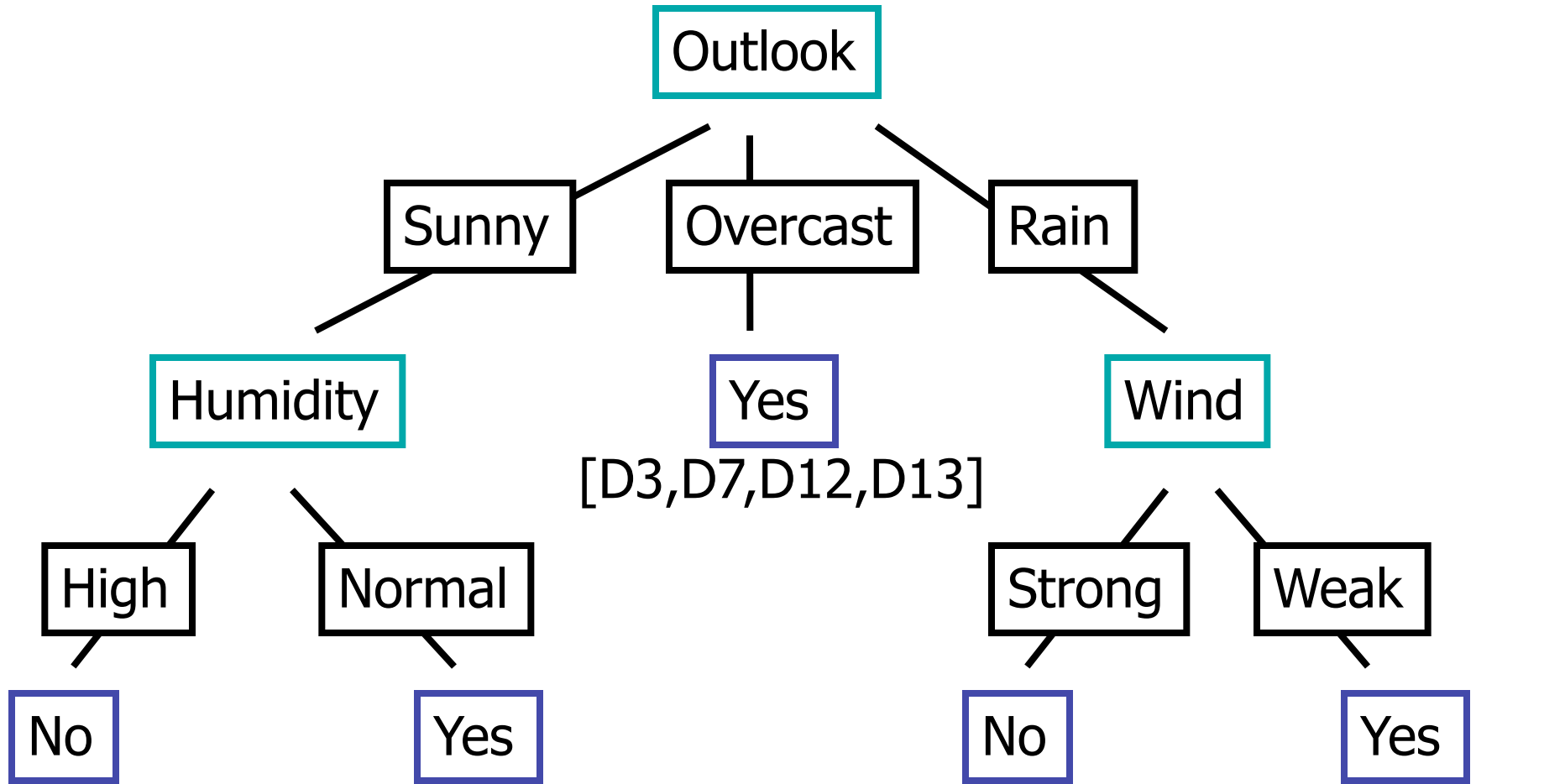
$$\begin{aligned}
 \text{Gain}(S, \text{Outlook}) &= 0.940 - (5/14) * 0.971 \\
 &\quad - (4/14) * 0.0 - (5/14) * 0.0971 \\
 &= 0.247
 \end{aligned}$$

# Algorithme ID3



$$\begin{aligned} \text{Gain}(S_{\text{sunny}}, \text{Humidity}) &= 0.970 - (3/5)0.0 - 2/5(0.0) = 0.970 \\ \text{Gain}(S_{\text{sunny}}, \text{Temp.}) &= 0.970 - (2/5)0.0 - 2/5(1.0) - (1/5)0.0 = 0.570 \\ \text{Gain}(S_{\text{sunny}}, \text{Wind}) &= 0.970 - (2/5)1.0 - 3/5(0.918) = 0.019 \end{aligned}$$

# Algorithme ID3



# Problème des attributs continus

Certains attributs sont continus

- exemple : salaire

découper en sous-ensembles ordonnés  
(e.g., déciles)

- division en segments  $[a_0, a_1[$ ,  $[a_1, a_2[$ , ...,  $[a_{n-1}, a_n]$

utiliser moyenne, médiane, ... pour représenter  
minimiser la variance, une mesure de dispersion ...

investiguer différents cas et retenir le meilleur

- exemple : 2, 4, 8, etc. par découpe d'intervalles en 2 successivement

# Indice Gini

Utiliser l'indice Gini pour un partitionnement pur

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

$$Gini(S_1, S_2) = \frac{n_1}{n} Gini(S_1) + \frac{n_2}{n} Gini(S_2)$$

$p_i$  est la fréquence relative de la classe  $c$  dans  $S$

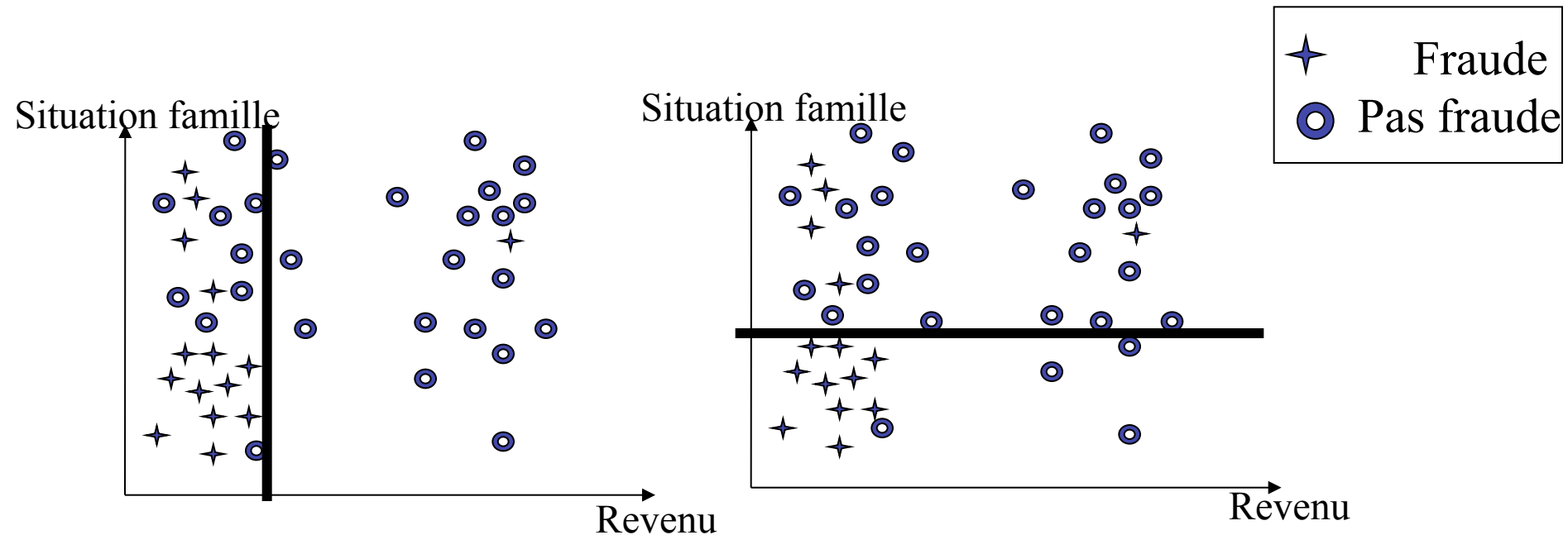
Si  $S$  est pur (classe unique),  $Gini(S) = 0$

$Gini(S_1, S_2)$  = Gini pour une partition de  $S$  en deux sous-ensembles  $S_1$  et  $S_2$  selon un test donné.

Trouver le branchement (split-point) qui **minimise** l'indice Gini

Nécessite seulement les distributions de classes

# Indice Gini - Exemple



Calcul de Gini nécessite une **Matrice de dénombrement**

	Non	Oui
<80K	14	9
>80K	1	18

$$\text{Gini(split)} = \mathbf{0.31}$$

	Non	Oui
M	5	23
F	10	4

$$\text{Gini(split)} = \mathbf{0.34}$$



# Attributs énumératifs – indice GINI

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	<b>0.400</b>	

Partage en plusieurs classes

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	<b>0.393</b>		

- Pour chaque valeur distincte, calculer le nombre d'instances de chaque classe
- Utiliser la **matrice de dénombrement** pour la prise de décision

Partage en deux “classes”  
(trouver la meilleure partition de valeurs)

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	<b>0.419</b>	

# Attributs numériques – indice GINI

**calcul efficace** : pour chaque attribut,

- Trier les instances selon la valeur de l'attribut
- Entre chaque valeur de cette liste : un test possible (split)
- Evaluation de Gini pour chacun des test
- Choisir le split qui minimise l'indice gini

Fraude		No		No		No		Yes		Yes		Yes		No		No		No		No			
es Split	→	Revenu imposable																					
		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		<u>0.300</u>		0.343		0.375		0.400		0.420	

# Algorithme CART

## Indice de Gini

$$I = 1 - \sum_i^n f_i^2$$

- N = nombre de classes à prédire
- $F_i$  = fréquence de la classe  $i$  dans le nœud

Plus l'indice de Gini est bas, plus le nœud est pure

# Algorithme CART

## Problèmes des arbres trop étoffés

- Complexité de l'arbre, trop de règles
- Trop spécifique aux données d'apprentissage
  - Règles non reproductibles (« surapprentissage »)
- Trop peu d'individus dans les feuilles (aucune signification réelle)
  - minimum conseillé : 20-30 individus

Solution → Élagage

# Algorithme CART

## Processus d'élagage de CART

- Création de l'arbre maximum
  - Toutes les feuilles des extrémités sont pures
- Élagages successifs de l'arbre
- Retient l'arbre élagué pour lequel le taux d'erreur estimé mesuré sur un échantillon test est le plus bas possible

# Avantages

## Résultats explicites

- Arbre
- Règles de décisions simples
- Modèle facilement programmable pour affecter de nouveaux individus

## Peu de perturbation des individus extrêmes

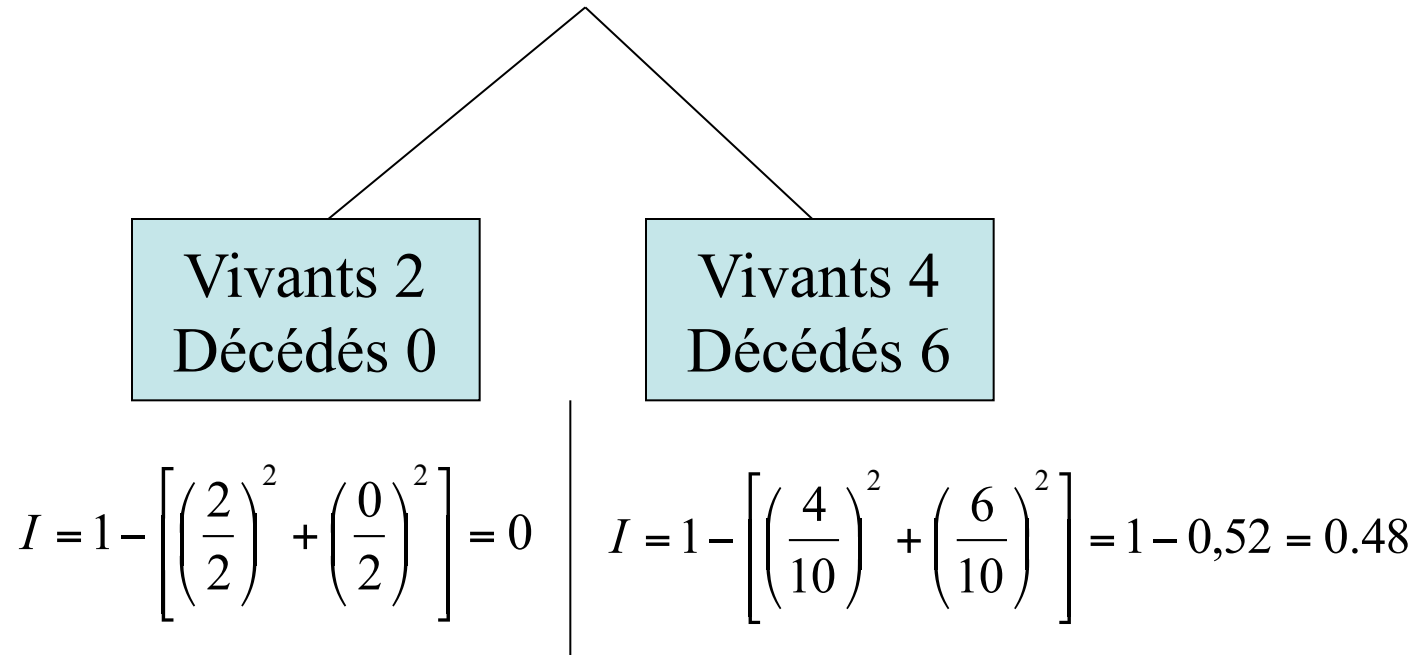
- Isolés dans des petites feuilles

## Peu sensible au bruit des variables non discriminantes

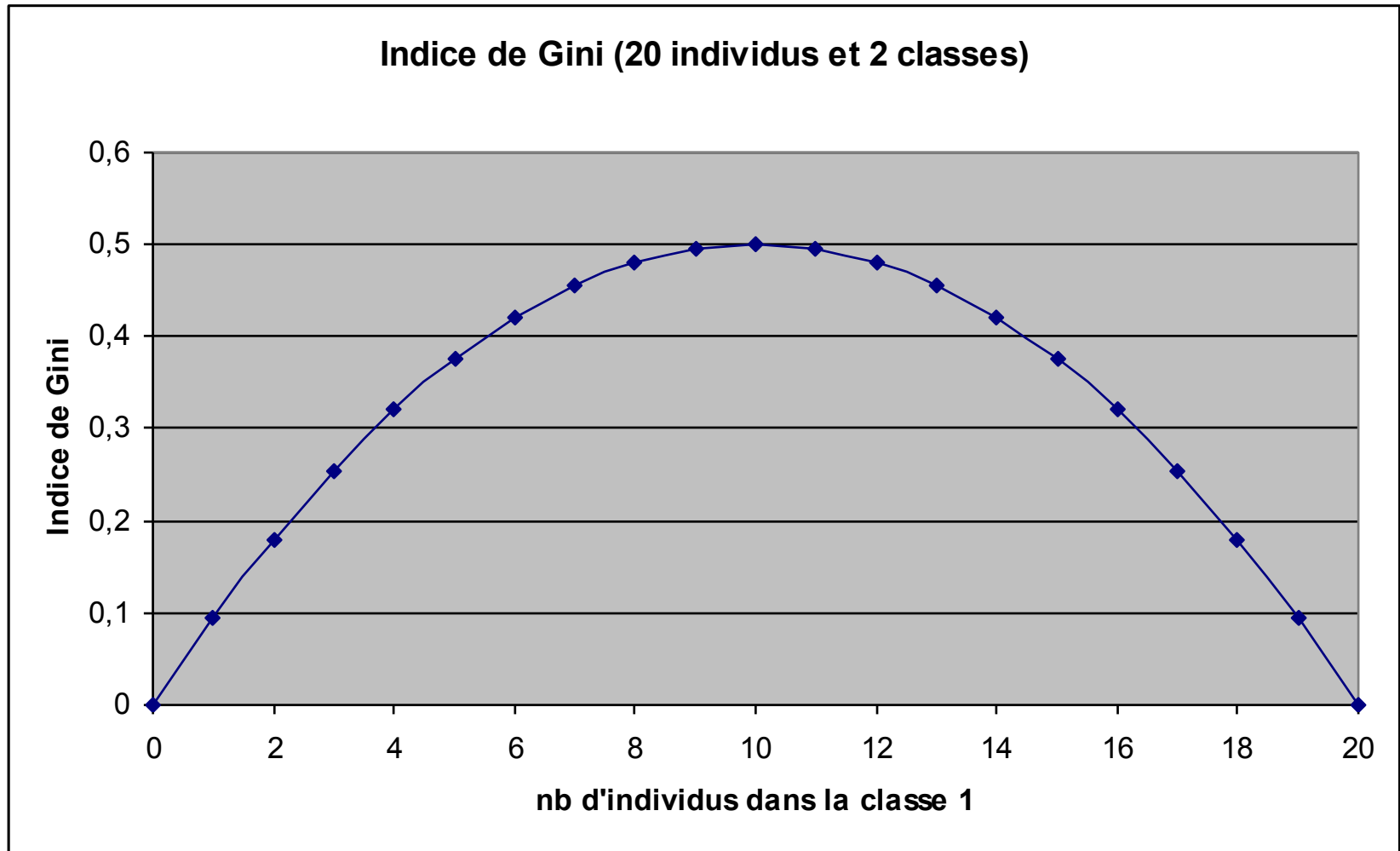
- Non introduites dans le modèle

# Algorithme CART

Exemple :



# Algorithme CART





# Algorithme CART

Ainsi,

- En séparant 1 nœud en 2 nœuds fils on cherche la plus grande hausse de la pureté
- La variable la plus discriminante doit maximiser :

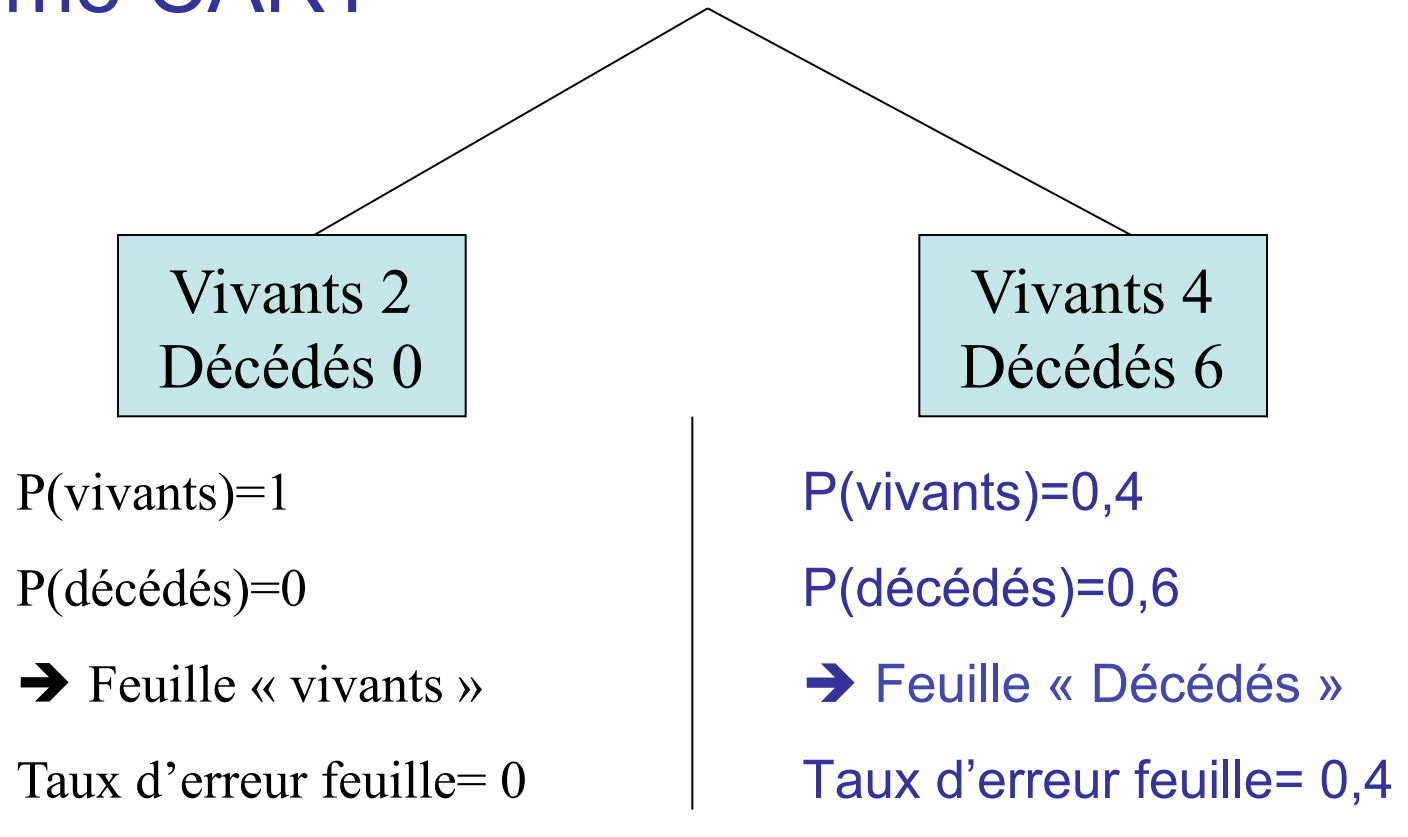
$$IG(\text{avant sep.}) - [IG(\text{fils1}) + IG(\text{fils2})]$$

# Algorithme CART

## Répartition des individus dans les nœuds

- Quand l'arbre est construit : critères de division connus
- On affecte chaque individu selon les règles obtenues → remplissage des feuilles
  - Pour chaque feuille : plusieurs classes  $C$ 
    - $P_c$  = Proportion d'individus de la feuille appartenant à la classe  $c$
  - On affecte à la feuille la classe pour laquelle  $P_c$  est la plus grande

# Algorithme CART



Taux d'erreur global de l'arbre = somme pondérée des taux d'erreur des feuilles

Pondération = proba qu'un individu soit dans la feuille (= taille de la feuille)

# Algorithme CART

## Problèmes des arbres trop étoffés

- Complexité de l'arbre, trop de règles
- Trop spécifique aux données d'apprentissage
  - Règles non reproductibles (« surapprentissage »)
- Trop peu d'individus dans les feuilles (aucune signification réelle)
  - minimum conseillé : 20-30 individus

Solution → Élagage

# Méthodes à base d'arbres de décision

CART (BFO'80 - Classification and regression trees, variables numériques, Gini, Elagage ascendant)

C5 (Quinlan'93 - dernière version ID3 et C4.5, attributs d'arité quelconque, entropie et gain d'information)

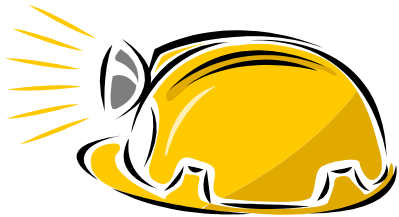
SLIQ (EDBT'96 — Mehta et al. IBM)

SPRINT (VLDB'96—J. Shafer et al. IBM)

PUBLIC (VLDB'98 — Rastogi & Shim)

RainForest (VLDB'98 — Gehrke, Ramakrishnan & Ganti)

CHAID (Chi-square Automation Interaction Detection – variables discrètes)



# Arbres de décision - Avantages

- 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

# Arbres de décision - Inconvénients

**Sensibles au nombre de classes** : performances se dégradent

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

Construction du modèle plus ou moins coûteuse

# RÉSEAUX BAYESIENS



# Classification bayésienne : Pourquoi ? (1)

## Apprentissage probabiliste :

- calcul explicite de probabilités sur des hypothèses
- Approche pratique pour certains types de problèmes d'apprentissage

## Incrémental :

- Chaque instance d'apprentissage peut de façon incrémentale augmenter/diminuer la probabilité qu'une hypothèse est correcte
- Des connaissances a priori peuvent être combinées avec les données observées.

# Classification bayésienne : Pourquoi ? (2)

## Prédiction Probabiliste :

- Prédit des hypothèses multiples, pondérées par leurs probabilités.

## Référence en terme d'évaluation :

- Même si les méthodes bayésiennes sont coûteuses en temps d'exécution, elles peuvent fournir des solutions optimales à partir desquelles les autres méthodes peuvent être évaluées.

# Classification bayésienne

Le problème de classification peut être formulé en utilisant les probabilités a-posteriori :

- $P(C|X)$  = probabilité que le tuple (instance)
- $X = \langle x_1, \dots, x_k \rangle$  est dans la classe  $C$

Par exemple

- $P(\text{classe} = N \mid \text{outlook} = \text{sunny}, \text{windy} = \text{true}, \dots)$

Idée : affecter à une instance  $X$  la classe  $C$  telle que  $P(C|X)$  est maximale

# Estimation des probabilités a-posteriori

Théorème de Bayes :

- $P(C|X) = P(X|C) \cdot P(C) / P(X)$

$P(X)$  est une constante pour toutes les classes

$P(C)$  = fréquence relative des instances de la classe  
 $C$

$C$  tel que  $P(C|X)$  est maximal =

$C$  tel que  $P(X|C) \cdot P(C)$  est maximal

Problème : calculer  $P(X|C)$  est non faisable !

# Classification bayésienne naïve

Hypothèse Naïve : indépendance des attributs

$$P(x_1, \dots, x_k | C) = P(x_1 | C) \cdot \dots \cdot P(x_k | C)$$

$P(x_i | C)$  est estimée comme la fréquence relative des instances possédant la valeur  $x_i$  ( $i$ -ème attribut) dans la classe  $C$

Non coûteux à calculer dans les deux cas

# Exemple de problème

Faut-il effectuer un contrôle fiscal ?

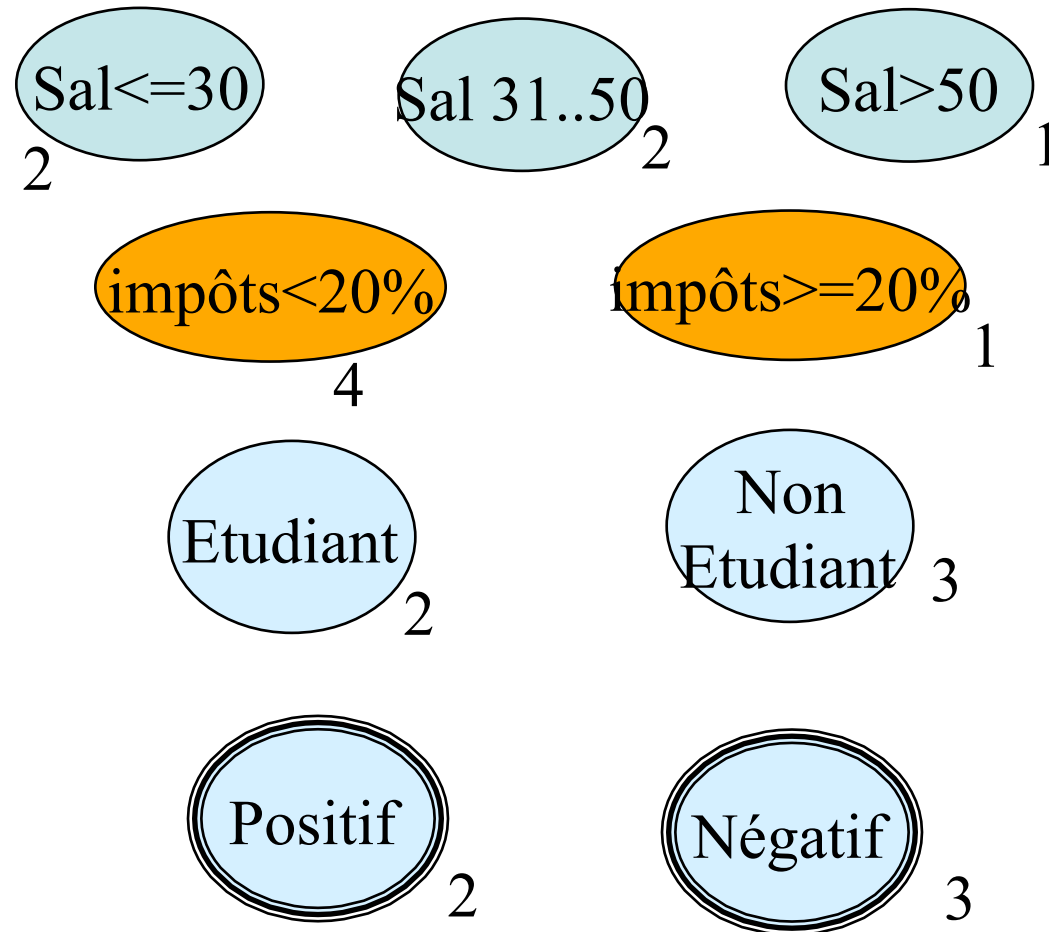
- Échantillon de contrôlés

Salaire	Impôts	Etudiant	Contrôle
20	0	oui	négatif
30	0	non	positif
40	5	oui	positif
40	10	non	négatif
60	10	non	positif

- Faut-il contrôler un nouvel arrivant ?

35	2	oui	???
----	---	-----	-----

# Les classes nominales



# Calcul de Probabilités

Il s'agit de choisir  $C_i$  maximisant  $P(C_i/X)$  :

- $P(\text{Positif}/X) = P(X/\text{Positif})P(\text{Positif})/P(X)$
- $P(\text{Négatif}/X) = P(X/\text{Négatif})P(\text{Négatif})/P(X)$
- $P(X)$  est supposé constant

Donc, choisir le plus grand de  $\{P(X/\text{Positif})P(\text{Positif}), P(X/\text{Négatif})P(\text{Négatif})\}$

- $P(X/\text{Positif}) = \prod_k P(X_k/\text{Positif}) = P(\text{sal}30..50/\text{Positif}) * P(\text{impots} < 20\%/\text{Positif}) * P(\text{Etudiant}/\text{Positif}) = 2/3 * 1 * 1/3 = 2/9$ ;  
 $P(\text{Positif}) = 3/5 \rightarrow \text{Produit} = 0.13$
- $P(X/\text{Négatif}) = \prod_k P(X_k/\text{Négatif}) = P(\text{sal}30..50/\text{Négatif}) * P(\text{impots} < 20\%/\text{Négatif}) * P(\text{Etudiant}/\text{Négatif}) = 1/2 * 1/2 * 1/2 = 1/8$ ;  
 $P(\text{Négatif}) = 2/5 \rightarrow \text{Produit} = 0.05$

On effectuera donc un contrôle !



# Classification bayésienne – Exemple (1)

Estimation de  $P(x_i|C)$

$$P(p) = 9/14$$

$$P(n) = 5/14$$

Outlook	
$P(\text{sunny}   p) = 2/9$	$P(\text{sunny}   n) = 3/5$
$P(\text{overcast}   p) = 4/9$	$P(\text{overcast}   n) = 0$
$P(\text{rain}   p) = 3/9$	$P(\text{rain}   n) = 2/5$
Temperature	
$P(\text{hot}   p) = 2/9$	$P(\text{hot}   n) = 2/5$
$P(\text{mild}   p) = 4/9$	$P(\text{mild}   n) = 2/5$
$P(\text{cool}   p) = 3/9$	$P(\text{cool}   n) = 1/5$

Humidity	
$P(\text{high}   p) = 3/9$	$P(\text{high}   n) = 4/5$
$P(\text{normal}   p) = 6/9$	$P(\text{normal}   n) = 1/5$
Windy	
$P(\text{true}   p) = 3/9$	$P(\text{true}   n) = 3/5$
$P(\text{false}   p) = 6/9$	$P(\text{false}   n) = 2/5$

# Classification bayésienne – Exemple (1)

Classification de X :

- Une instance inconnue  $X = \langle \text{rain, hot, high, false} \rangle$
- $P(X|p) \cdot P(p) =$   
 $P(\text{rain}|p) \cdot P(\text{hot}|p) \cdot P(\text{high}|p) \cdot P(\text{false}|p) \cdot P(p) = 3/9 \cdot 2/9 \cdot 3/9 \cdot 6/9 \cdot 9/14 = 0.010582$
- $P(X|n) \cdot P(n) =$   
 $P(\text{rain}|n) \cdot P(\text{hot}|n) \cdot P(\text{high}|n) \cdot P(\text{false}|n) \cdot P(n) = 2/5 \cdot 2/5 \cdot 4/5 \cdot 2/5 \cdot 5/14 = 0.018286$
- Instance X est classifiée dans la classe n (ne pas jouer)

# Réseau Bayésien

Nœuds = Variables aléatoires

## Structure

- Graphe direct acyclique de dépendance
- $X \rightarrow Y$  signifie que  $X$  est un parent de  $Y$
- $X \rightarrow \rightarrow Y$  signifie que  $X$  est un descendant de  $Y$
- Les variables non liées sont indépendantes

## Classes à déterminer

- Nœuds singuliers du réseau

## Probabilités connues

- à priori et conditionnelles (arcs)

# Autre exemple

## Classification de pannes d'ordinateurs

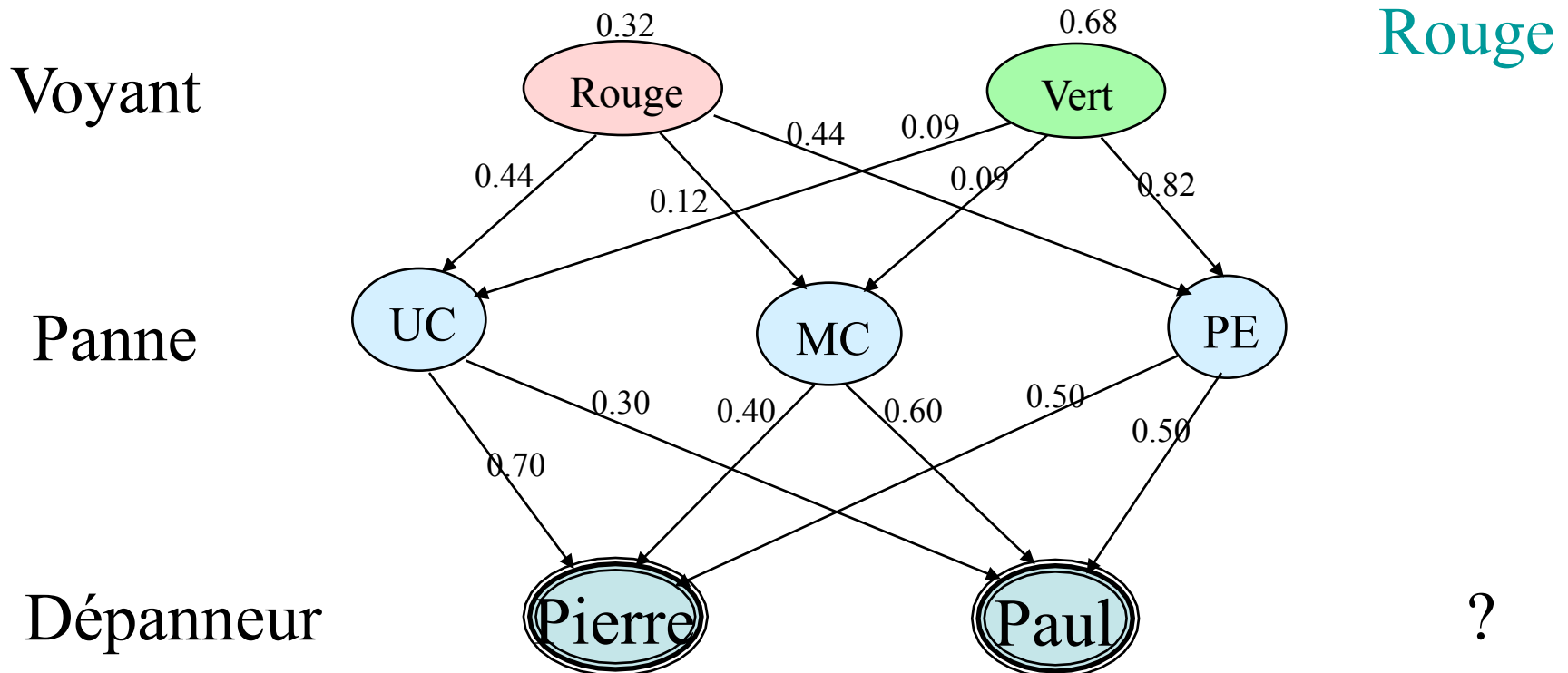
- Couleur de voyant (Rouge, Vert)
- Équipement défaillant (UC,MC,PE)

Envoie d'un dépanneur selon la classe

Calcul de probabilités sur le training set

P(Couleur/Panne)	Rouge	Vert	P(Panne)
UC	0,70	0,30	0,20
MC	0,40	0,60	0,10
PE	0,20	0,80	0,70
P(Couleur)	0,32	0,68	1,00

# Exemple de réseau



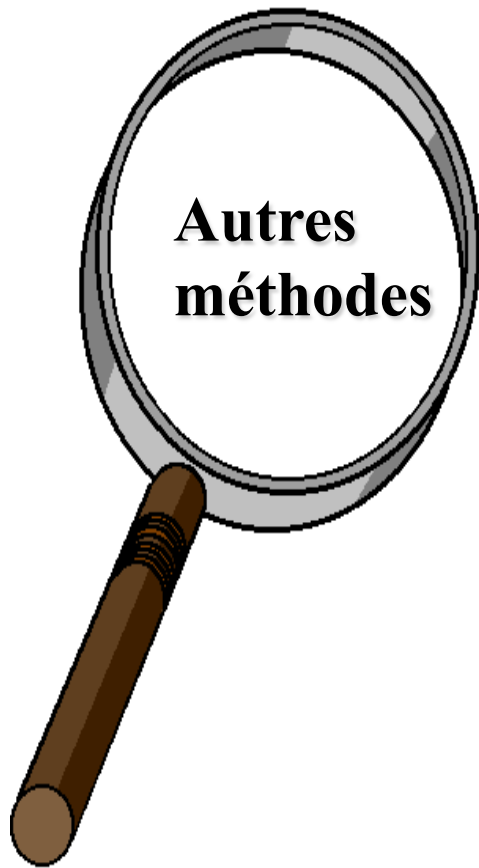
# Classification bayésienne – l'hypothèse d'indépendance

- ... fait que le calcul est possible
- ... trouve un modèle de classification optimal si hypothèse satisfaite
- ... mais est rarement satisfaite en pratique, étant donné que les attributs (variables) sont souvent corrélés.

Pour éliminer cette limitation :

- Réseaux bayésiens, qui combinent le raisonnement bayésien et la relation causale entre attributs
- Arbres de décision, qui traitent un attribut à la fois, considérant les attributs les plus importants en premier

# Autres méthodes de classification



Réseaux bayésiens

Algorithmes génétiques

Case-based reasoning

Ensembles flous

Rough set

Analyse discriminante (Discriminant linéaire de Fisher, Algorithme Closest Class Mean - CCM-)

Chaînes de Markov cachées

# Classification - Résumé

La **classification** est un problème largement étudié

La **classification**, avec ses nombreuses extensions, est probablement la technique la plus répandue

- **Modèles**

- Arbres de décision
- Règles d'induction
- Modèles de régression
- Réseaux de neurones

↑ Facile à comprendre

↓ Difficile à comprendre



# Classification - Résumé

**L'extensibilité** reste une issue importante pour les applications

**Directions de recherche** : classification de données non relationnels, e.x., texte, spatiales et données multimédia

# Classification - Références

- J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufman, 1993.
- J. R. Quinlan. Induction of decision trees. Machine Learning, 1:81-106, 1986.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984.
- S. M. Weiss and C. A. Kulikowski. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Morgan Kaufman, 1991.
- D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning internal representation by error propagation. In D. E. Rumelhart and J. L. McClelland (eds.) Parallel Distributed Processing. The MIT Press, 1986