

# Fouille de Données

# Data Mining

## **Classification - Partie 1**

# Plan du cours

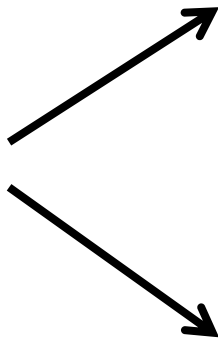
1. Contexte et buts
2. Organisation
3. Phases de classification
4. Les arbres de décision
5. Algorithme ID3

# Classification

SAVOIR - **PREDIRE** - DECIDER



Données



Algorithmes



Connaissances

## Classification : Dataset Labelisé



attributs / features

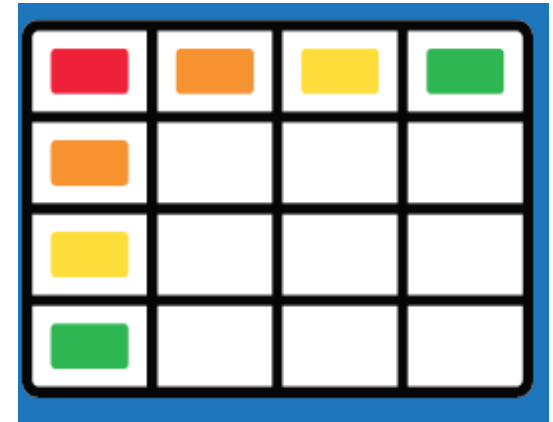
classe/label

	$f_1$	$f_2$	...	$f_K$	$y$
$x_1$	$x_{1,1}$	$x_{1,2}$	...	$x_{1,K}$	$y_1$
$x_2$	$x_{2,1}$	$x_{2,2}$	...	$x_{2,K}$	$y_2$
...	...	...	...	...	...
$x_r$	$x_{r,1}$	$x_{r,2}$	...	$x_{r,K}$	$y_r$

exemples

## Classification : Dataset Labelisé

- Chaque exemple de l'ensemble d'exemples  **$S$**  est représenté par  **$m$**  attributs et sa classe  **$y \in Y$** .
- Classe, ou label, ou étiquette.
- Dans la classification, la classe prend sa valeur parmi un ensemble fini.
- **Classe** = attribut qualitatif/catégoriel.
- $|Y| = 1$  : Classification mono-classe
- $|Y| = 2$  : Classification binaire
- $|Y| > 2$  : Classification multi-classe
- On considère que chaque donnée appartient à une et une seule classe.



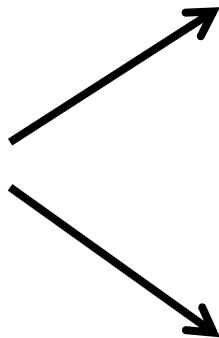
Red	Orange	Yellow	Green
Orange			
Yellow			
Green			

## Classification : Algorithmes

**SAVOIR - PREDIRE - DECIDER**



Données



#114983146



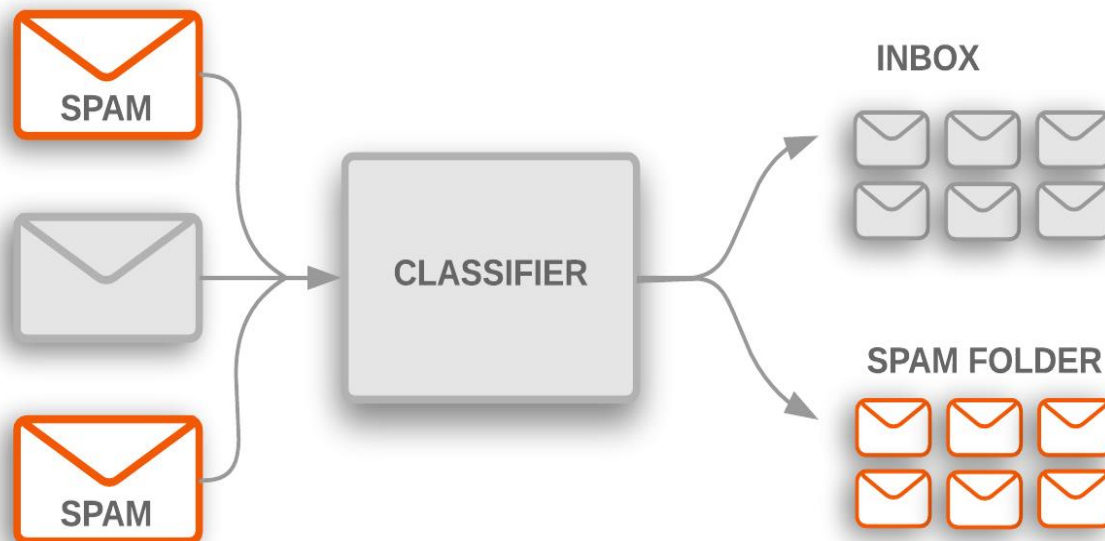
Connaissances

Decision Trees, Naive Bayes,  
KNN, RNN, SVM, etc.

## Contexte et Buts



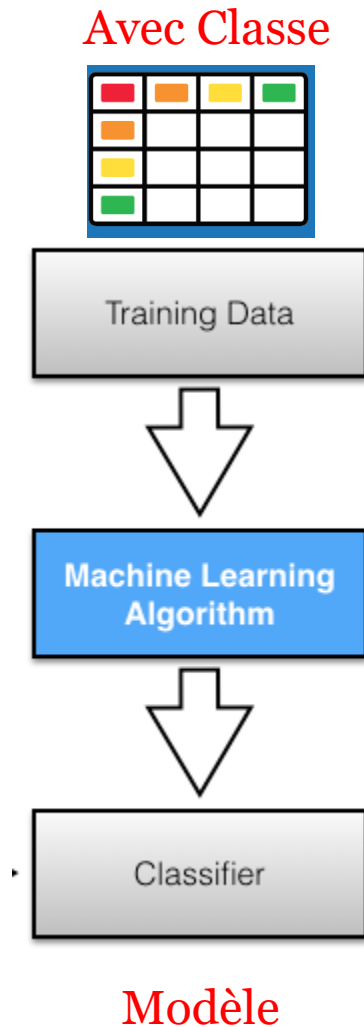
- **Classification : Apprentissage Supervisé.**
- Permet d'apprendre des **modèles (classifier)** de décision pour **prédire**/classifier le comportement des exemples futurs.
- Ex: Une tumeur est-elle bénigne ou maligne ?
- Ex: Une transaction carte de crédit est-elle frauduleuse ou non ?
- Ex: Catégorie d'une actualité/news : Sport, Politique, Musique, etc.



## Contexte et Buts

### Deux étapes :

1. Apprentissage (**entraînement**)

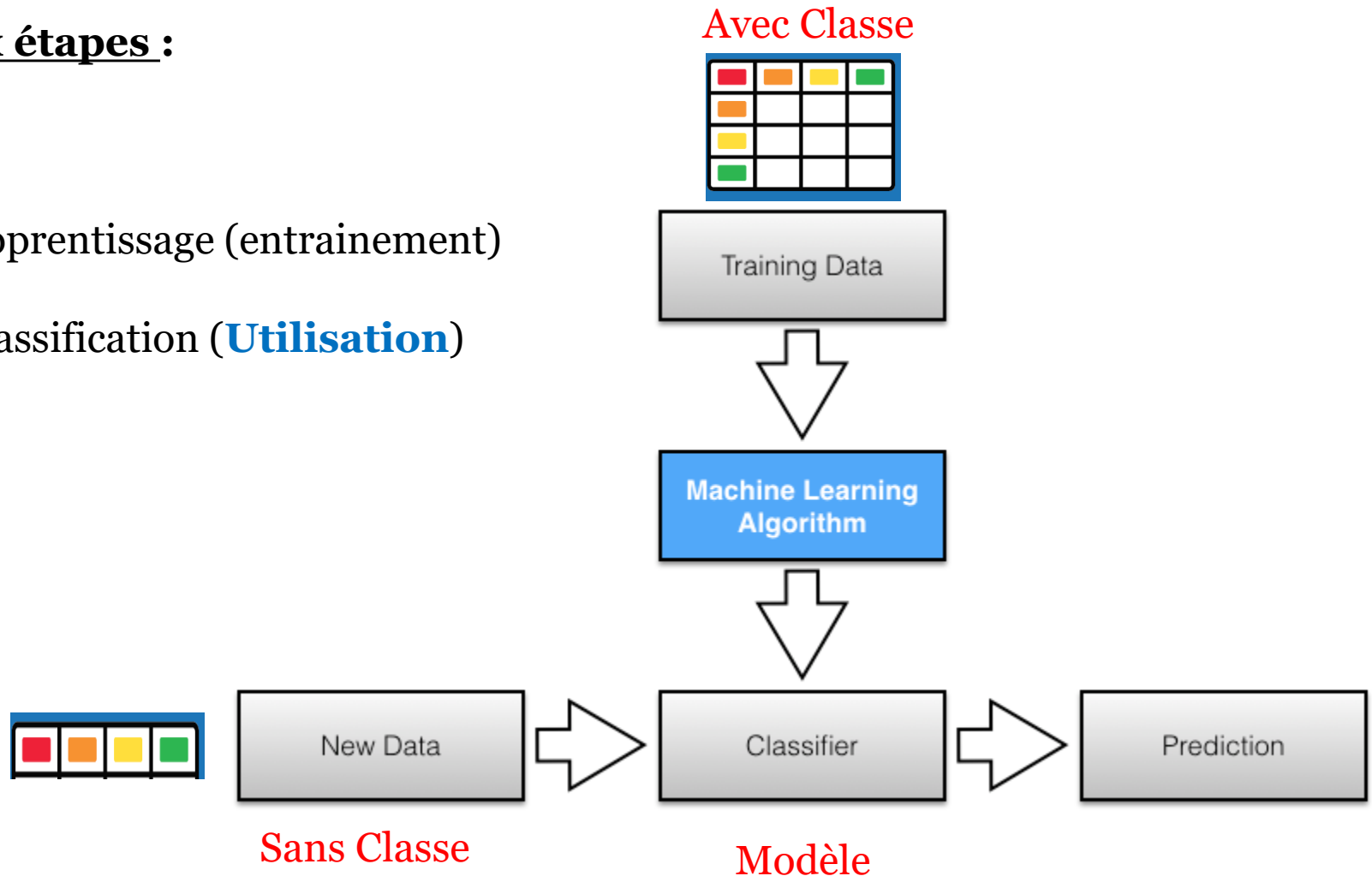




## Contexte et Buts

### Deux étapes :

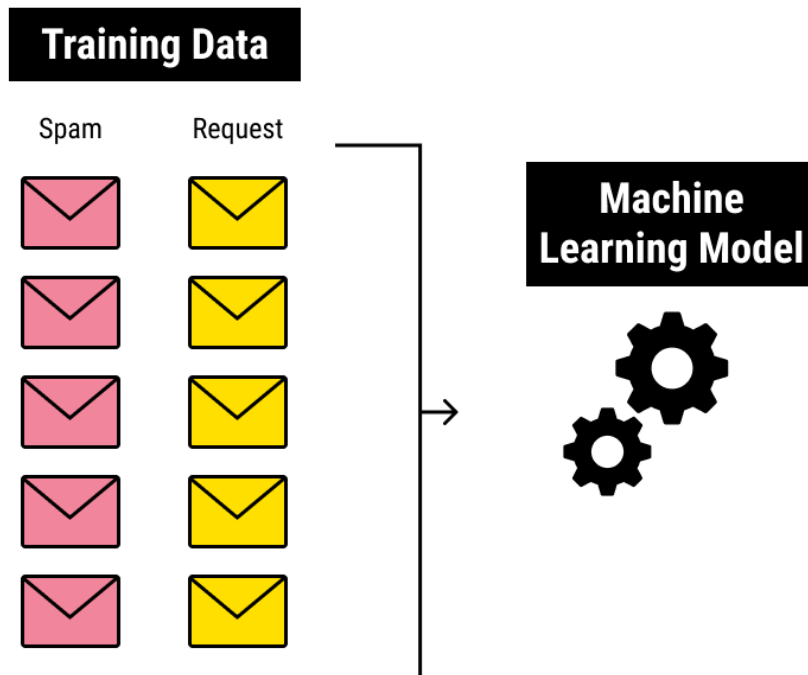
1. Apprentissage (entrainement)
2. Classification (**Utilisation**)



# Contexte et Buts

## Deux étapes : **Exemple**

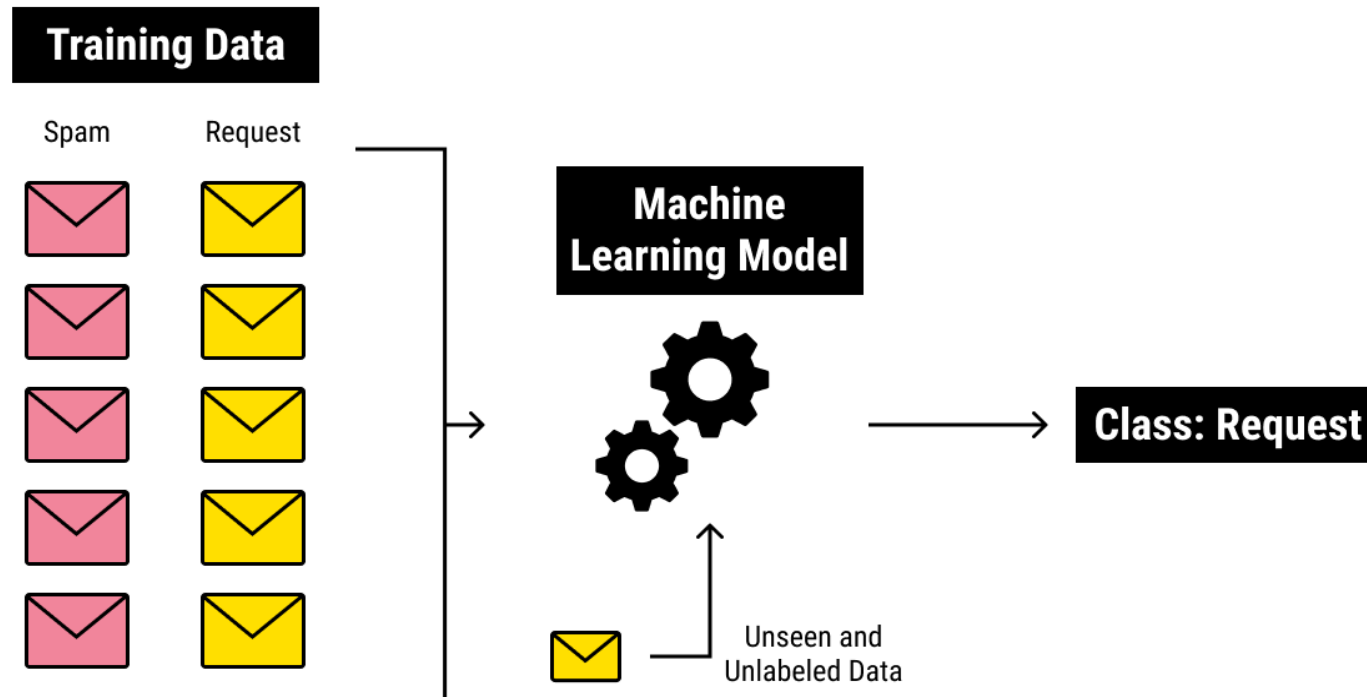
### 1. Apprentissage (entraînement)



## Contexte et Buts

### Deux étapes : **Exemple**

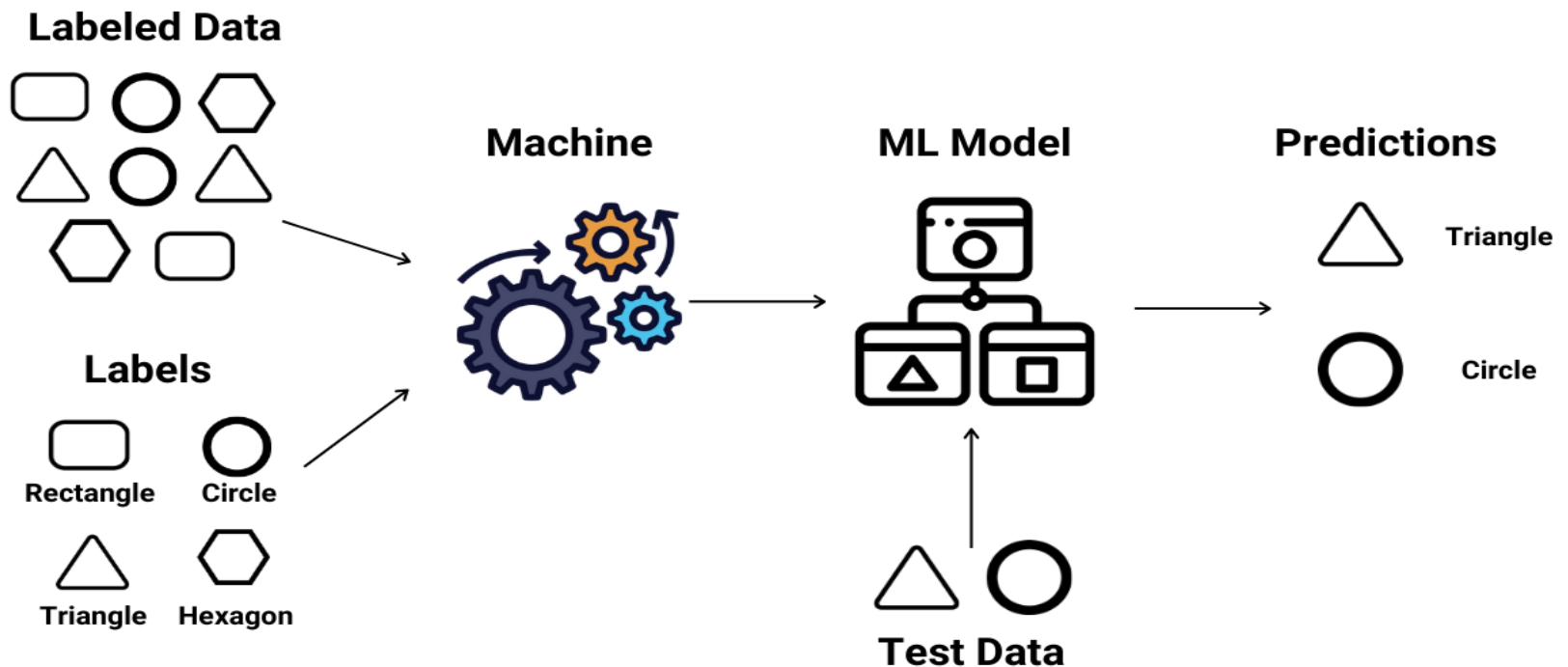
1. Apprentissage (entraînement)
2. Classification (Utilisation)



# Contexte et Buts

## Deux étapes : **Exemple 2**

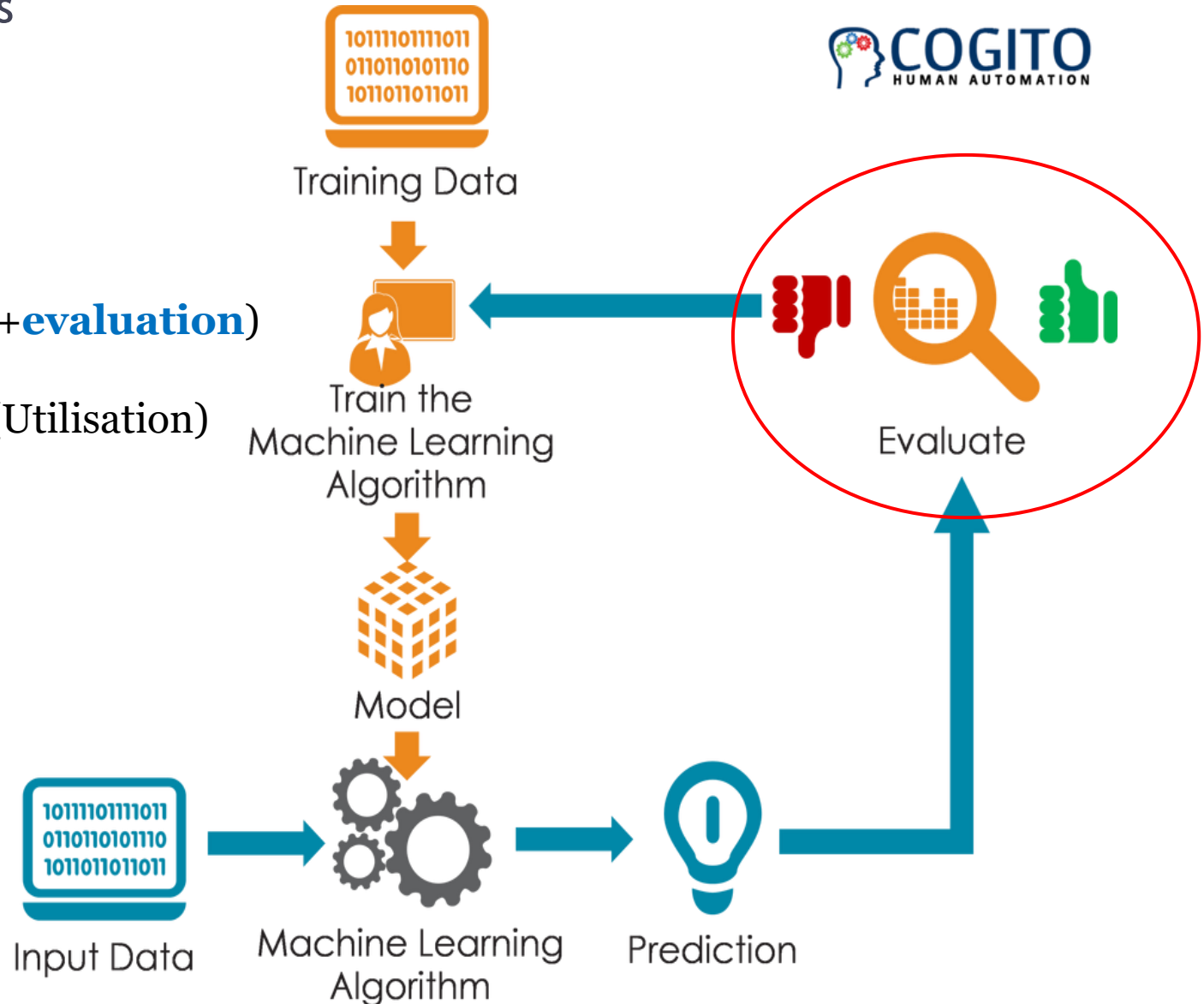
1. Apprentissage (entrainement)
2. Classification (Utilisation)



## Contexte et Buts

### Deux étapes:

1. Apprentissage  
(entraînement+**evaluation**)
2. Classification (Utilisation)



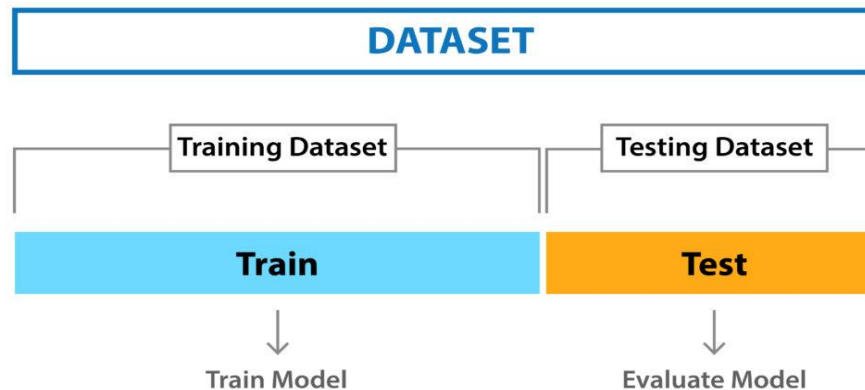
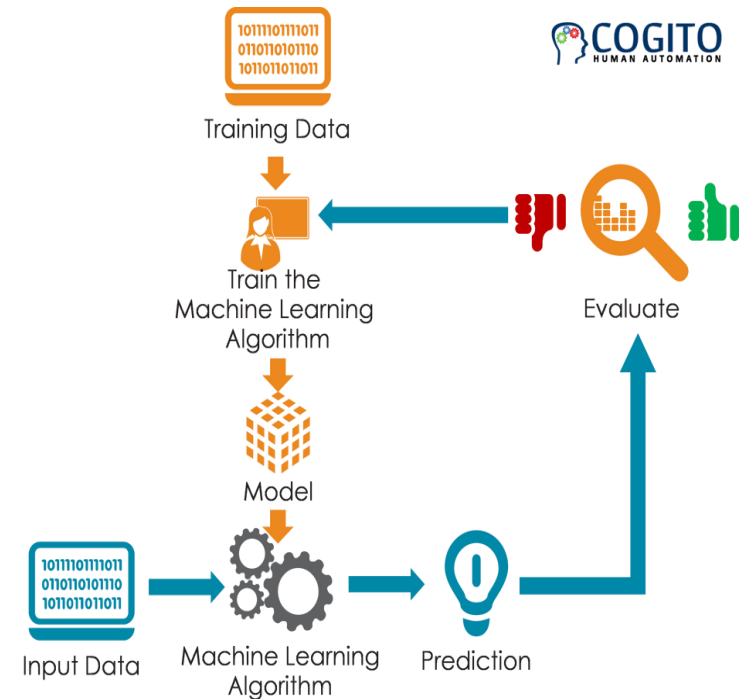
# Contexte et Buts

## Deux étapes:

1. Apprentissage  
(entraînement + **evaluation**)
1. Classification (Utilisation)

## Deux bases d'exemples:

1. Training Set
2. Test Set



## Contexte et Buts

	attributs / features				classe
	$f_1$	$f_2$	...	$f_K$	$y$
$x_1$	$x_{1,1}$	$x_{1,2}$	...	$x_{1,K}$	$y_1$
$x_2$	$x_{2,1}$	$x_{2,2}$	...	$x_{2,K}$	$y_2$
...	...	...	...	...	...
$x_r$	$x_{r,1}$	$x_{r,2}$	...	$x_{r,K}$	$y_r$

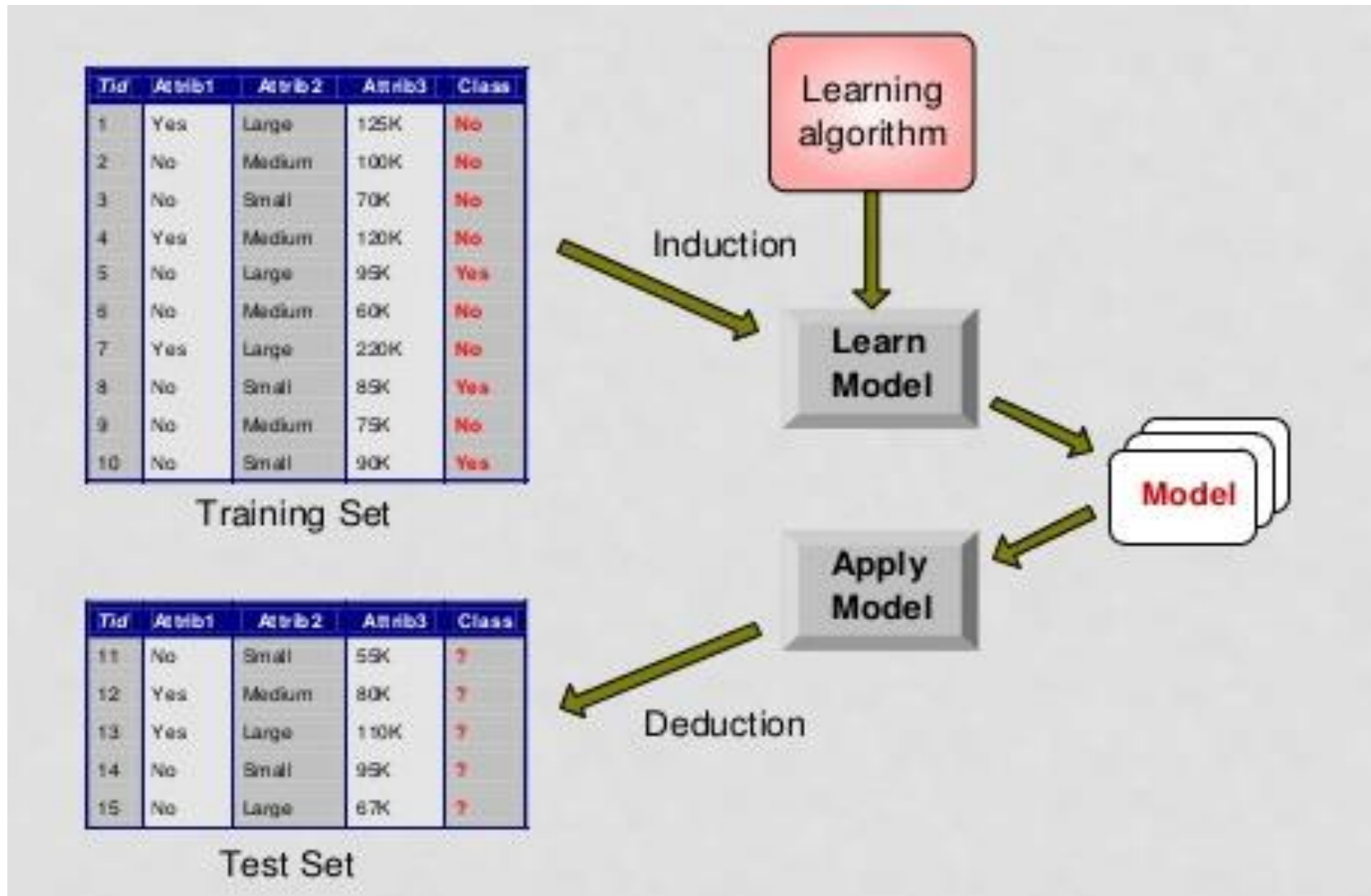
Training set

## Contexte et Buts

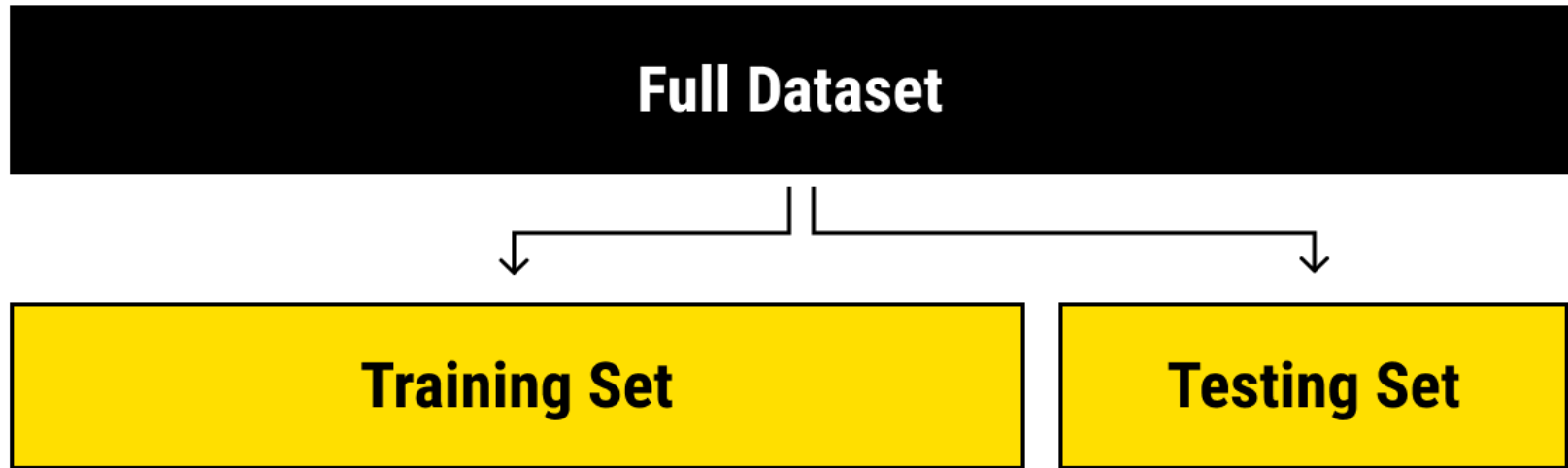
	attributs / features				prédire la classe
	$f_1$	$f_2$	...	$f_K$	
$X_1$	$X_{1,1}$	$X_{1,2}$	...	$X_{1,K}$	Test Set
$X_2$	$X_{2,1}$	$X_{2,2}$	...	$X_{2,K}$	
...	...	...	...	...	
$X_r$	$X_{r,1}$	$X_{r,2}$	...	$X_{r,K}$	



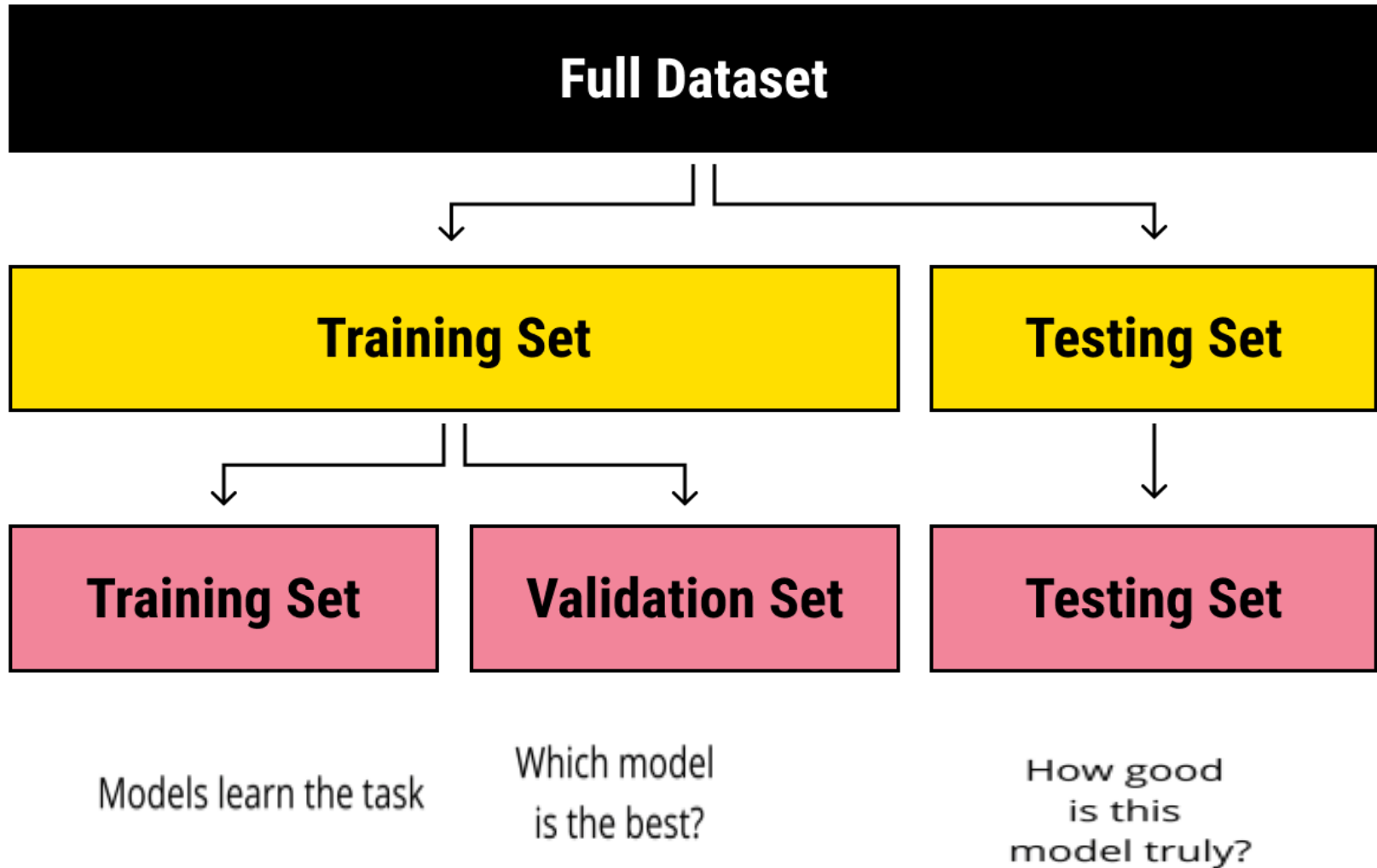
## Contexte et Buts



## Contexte et Buts : Evaluation et Validation



## Contexte et Buts : Evaluation et Validation



## Contexte et Buts

### Pourquoi deux bases ?

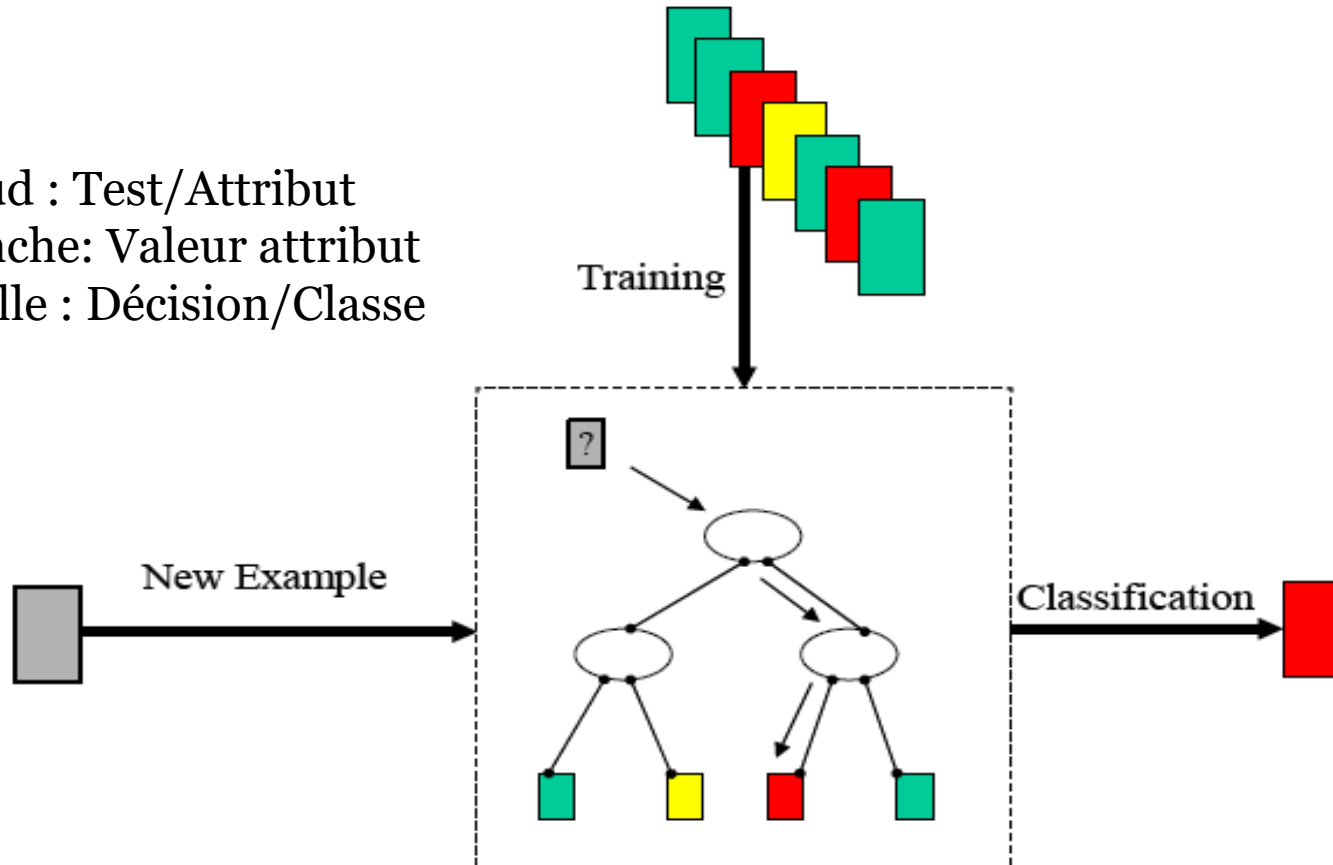
- Les données d'entraînement peuvent contenir des données bruitées ou erronées.
- Des données qui ne représentent pas le cas général tirant le modèle vers leurs caractéristiques.
- Problème de Sur-apprentissage - **Overfitting**.
- => Utilisation de la base de test.
- La base de test est un ensemble d'exemples ayant les mêmes caractéristiques que ceux de la base d'entraînement et qui sont écartés au départ de l'entraînement pour effectuer les tests.

## Les arbres de décision

- Méthode très efficace d'apprentissage et de classification supervisés.
- Partitionner un ensemble de données en des groupes les plus **homogènes** possible du point de vue de la classe à prédire.

### Arbre

- ✓ Nœud : Test/Attribut
- ✓ Branche: Valeur attribut
- ✓ Feuille : Décision/Classe



# Les arbres de décision

## Algorithmes de construction d'arbres de décision

- Diviser pour régner – **Divide and Conquer**.
- **TDIDT**: Top-Down Induction of Decision Trees.
- L'arbre est construit récursivement de haut en bas selon le principe « diviser pour régner ».
  - Diviser le problème en sous-problèmes.
  - Résoudre chaque sous-problème.
- Au début, tous les exemples sont dans la racine.
- Ensuite, les exemples sont partitionnés récursivement selon les attributs sélectionnés.

# Les arbres de décision

## Exemple

<i>Day</i>	<i>Temperature</i>	<i>Outlook</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play Golf?</i>
07-05	hot	sunny	high	weak	no
07-06	hot	sunny	high	strong	no
07-07	hot	overcast	high	weak	yes
07-09	cool	rain	normal	weak	yes
07-10	cool	overcast	normal	strong	yes
07-12	mild	sunny	high	weak	no
07-14	cool	sunny	normal	weak	yes
07-15	mild	rain	normal	weak	yes
07-20	mild	sunny	normal	strong	yes
07-21	mild	overcast	high	strong	yes
07-22	hot	overcast	normal	weak	yes
07-23	mild	rain	high	strong	no
07-26	cool	rain	normal	strong	no
07-30	mild	rain	high	weak	yes

today	cool	sunny	normal	weak	?
tomorrow	mild	sunny	normal	weak	?

# Les arbres de décision

## Exemple

Training examples: **9 yes / 5 no**



Outlook	Humid	Wind
Sunny	High	Weak
Sunny	High	Strong
Sunny	High	Weak
Sunny	Normal	Weak
Sunny	Normal	Strong

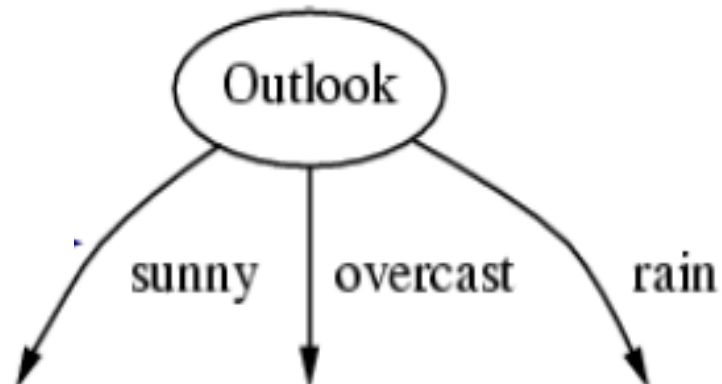
**2 yes / 3 no**  
split further

Outlook	Humidity	Wind	Play
Sunny	High	Weak	No
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Rain	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Strong	No
Overcast	Normal	Strong	Yes
Sunny	High	Weak	No
Sunny	Normal	Weak	Yes
Rain	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Overcast	High	Strong	Yes
Overcast	Normal	Weak	Yes
Rain	High	Strong	No



# Les arbres de décision

## Exemple



Outlook	Humid	Wind
Overcast	High	Weak
Overcast	Normal	Strong
Overcast	High	Strong
Overcast	Normal	Weak

**4 yes / 0 no**  
**pure subset**

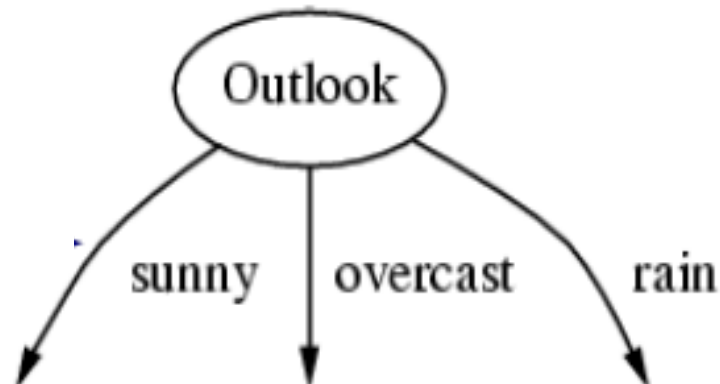
Training examples: **9 yes / 5 no**

Outlook	Humidity	Wind	Play
Sunny	High	Weak	No
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Rain	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Strong	No
Overcast	Normal	Strong	Yes
Sunny	High	Weak	No
Sunny	Normal	Weak	Yes
Rain	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Overcast	High	Strong	Yes
Overcast	Normal	Weak	Yes
Rain	High	Strong	No

# Les arbres de décision

## Exemple

Training examples: **9 yes / 5 no**



Outlook	Humid	Wind
Rain	High	Weak
Rain	Normal	Weak
Rain	Normal	Strong
Rain	Normal	Weak
Rain	High	Strong

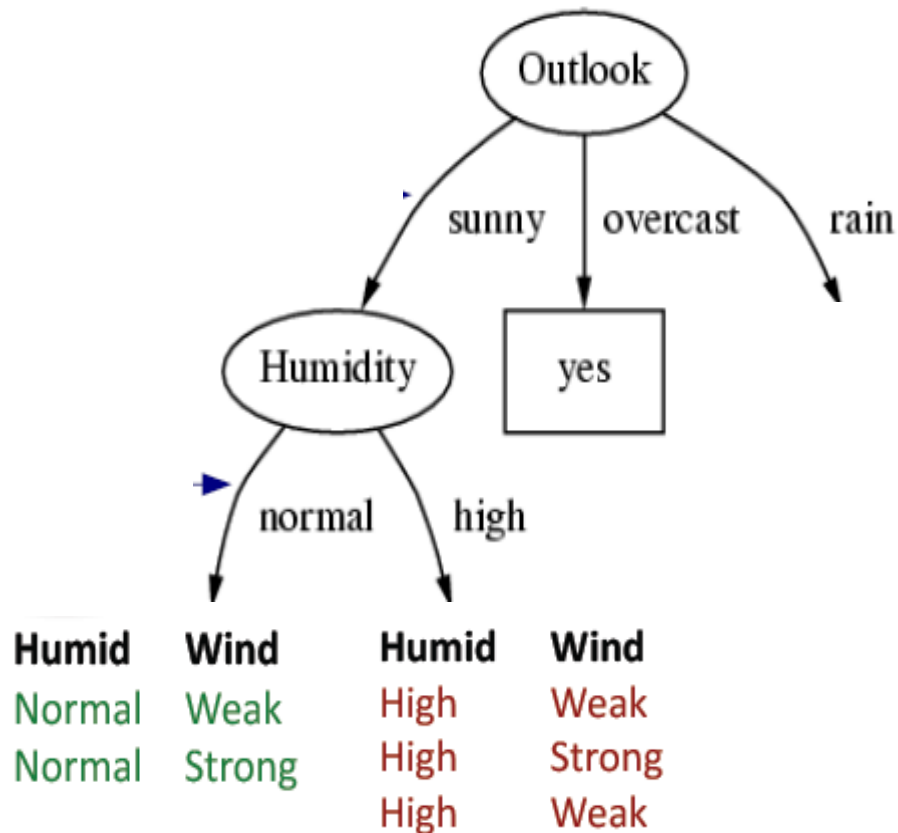
**3 yes / 2 no**  
split further

Outlook	Humidity	Wind	Play
Sunny	High	Weak	No
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Rain	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Strong	No
Overcast	Normal	Strong	Yes
Sunny	High	Weak	No
Sunny	Normal	Weak	Yes
Rain	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Overcast	High	Strong	Yes
Overcast	Normal	Weak	Yes
Rain	High	Strong	No

# Les arbres de décision

## Exemple

Training examples: **9 yes / 5 no**

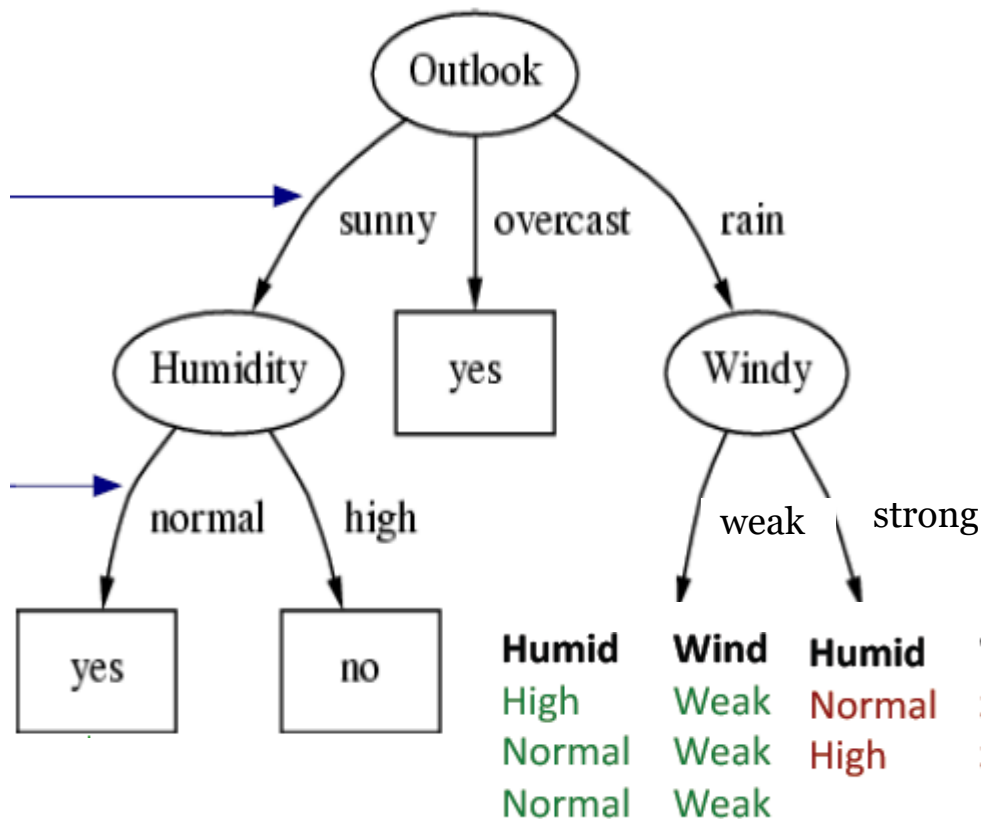


Outlook	Humidity	Wind	Play
Sunny	High	Weak	No
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Rain	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Strong	No
Overcast	Normal	Strong	Yes
Sunny	High	Weak	No
Sunny	Normal	Weak	Yes
Rain	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Overcast	High	Strong	Yes
Overcast	Normal	Weak	Yes
Rain	High	Strong	No

# Les arbres de décision

## Exemple

Training examples: **9 yes / 5 no**



### Outlook

Sunny  
Sunny  
Overcast  
Rain  
Rain  
Rain  
Overcast  
Sunny  
Sunny  
Rain  
Sunny  
Overcast  
Overcast  
Rain

### Humidity

High  
High  
High  
High  
Normal  
Normal  
Normal  
High  
Normal  
Normal  
Normal  
High  
Normal  
High

### Wind

Weak  
Strong  
Weak  
Weak  
Weak  
Strong  
Strong  
Weak  
Weak  
Weak  
Strong  
Strong  
Weak  
Strong

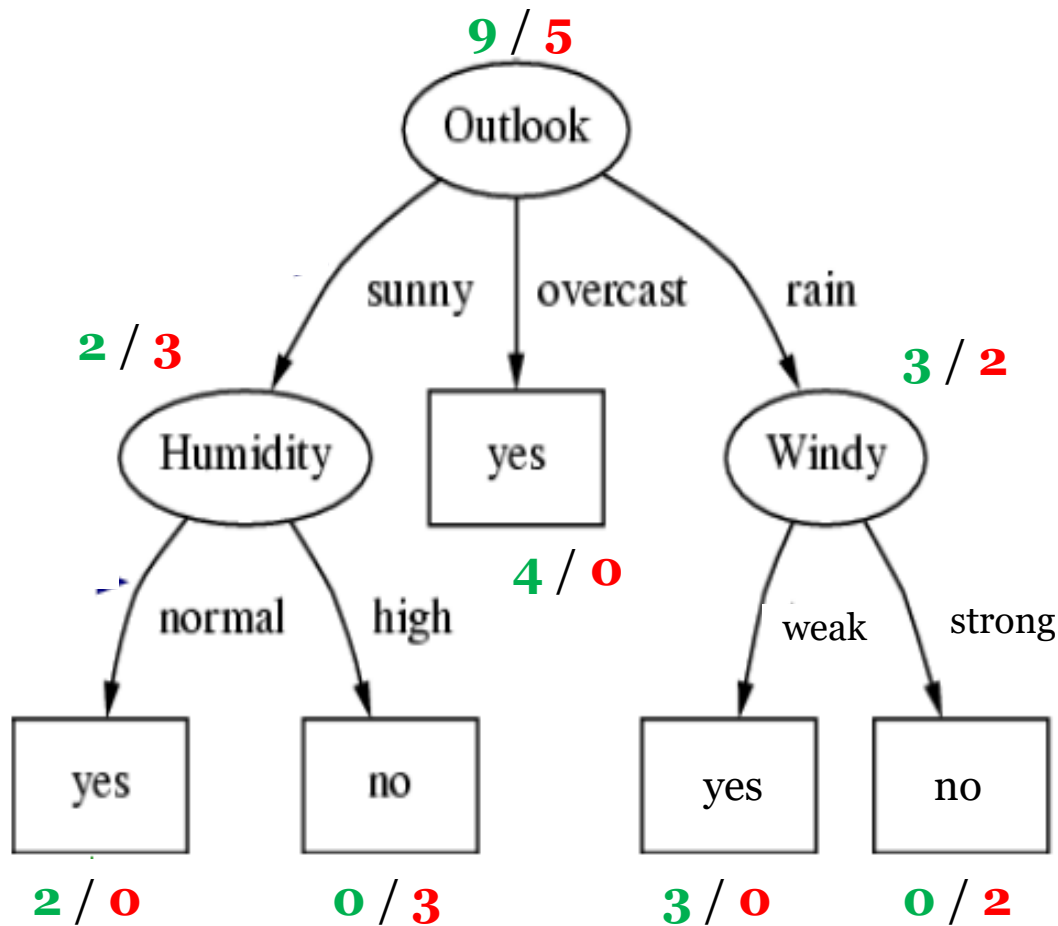
### Play

No  
No  
Yes  
Yes  
Yes  
No  
Yes  
No  
Yes  
Yes  
Yes  
Yes  
Yes  
No

# Les arbres de décision

## Exemple

Training examples: **9 yes / 5 no**



### Outlook

Sunny  
Sunny  
Overcast  
Rain  
Rain  
Rain  
Overcast  
Sunny  
Sunny  
Rain  
Sunny  
Overcast  
Overcast  
Rain

### Humidity

High  
High  
High  
High  
Normal  
Normal  
Normal  
High  
Normal  
Normal  
Normal  
High  
Normal  
High

### Wind

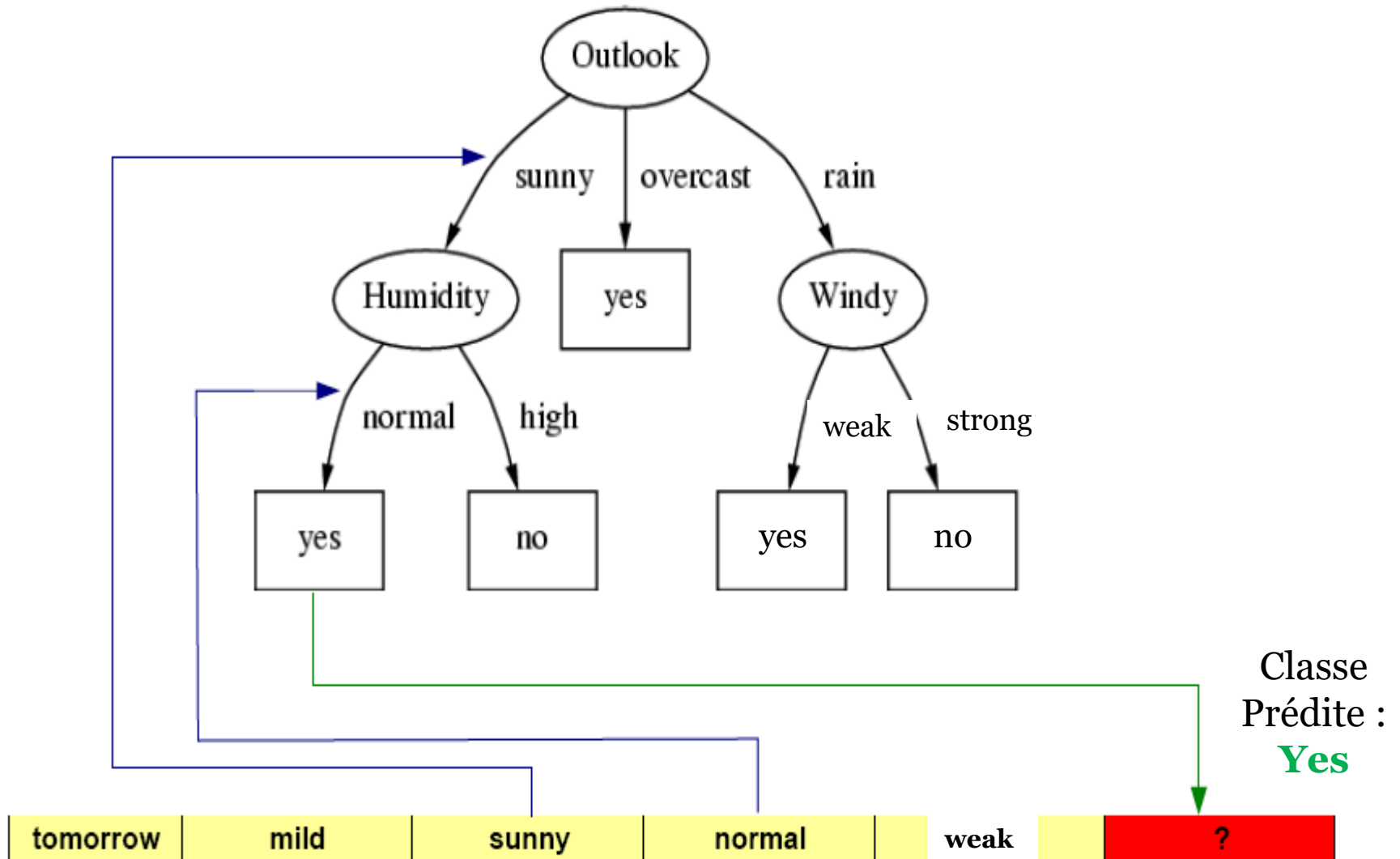
Weak  
Strong  
Weak  
Weak  
Weak  
Strong  
Strong  
Weak  
Weak  
Weak  
Strong  
Strong  
Weak  
Strong

### Play

No  
No  
Yes  
Yes  
Yes  
No  
Yes  
No  
Yes  
Yes  
Yes  
Yes  
Yes  
No

# Les arbres de décision

## Exemple



# Les arbres de décision

## Algorithmes de construction d'arbres de décision

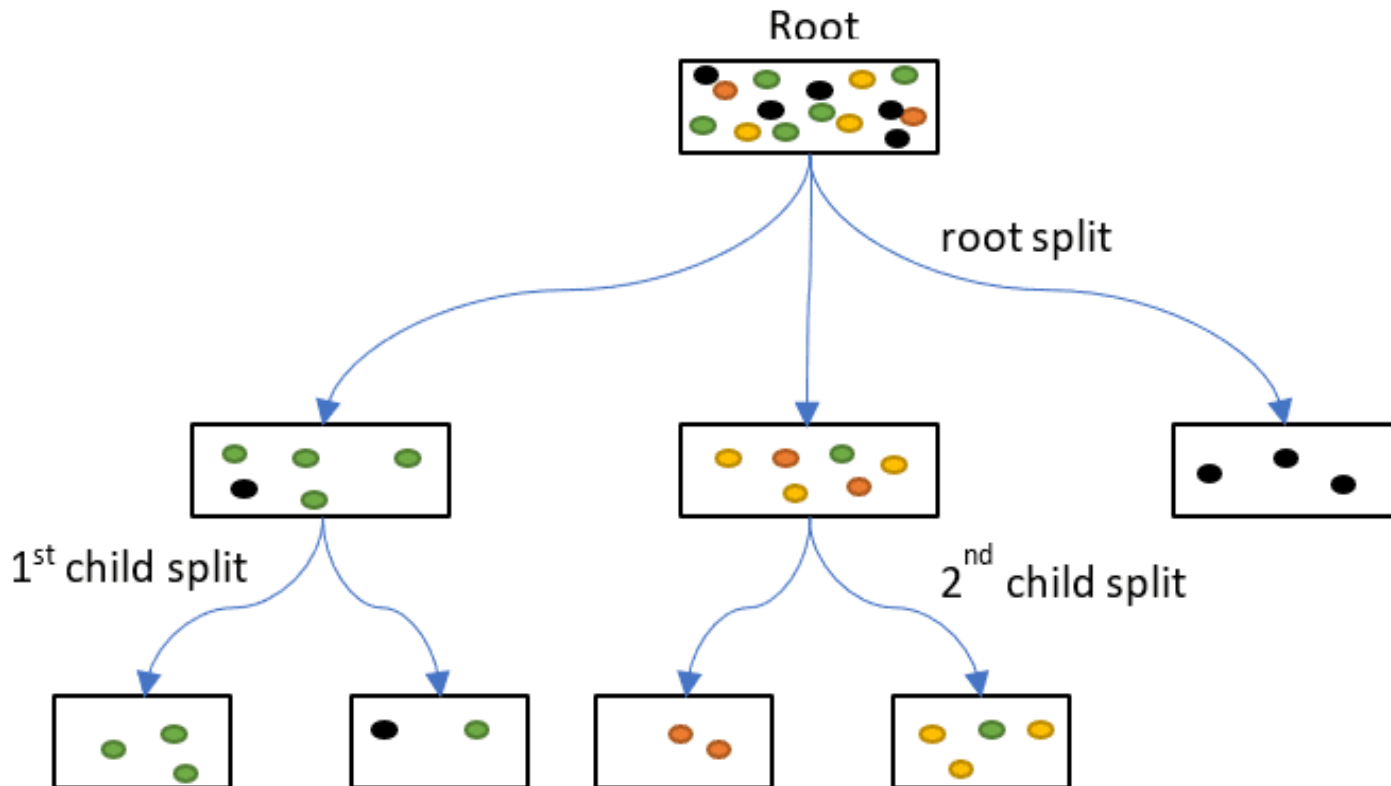
Etapes générales :

1. Sélectionner un attribut pour le nœud racine.
  - Créez une branche pour chaque valeur possible de l'attribut.
2. Diviser les exemples en sous-ensembles.
  - Un pour chaque branche s'étendant à partir du nœud.
3. Répéter récursivement pour chaque branche, en utilisant uniquement les exemples qui atteignent la branche.
4. Arrêter la récursivité pour une branche si tous ses exemples ont la même classe.

# Les arbres de décision

## Algorithmes de construction d'arbres de décision

Etapes générales :





# Les arbres de décision

## Algorithmes de construction d'arbres de décision

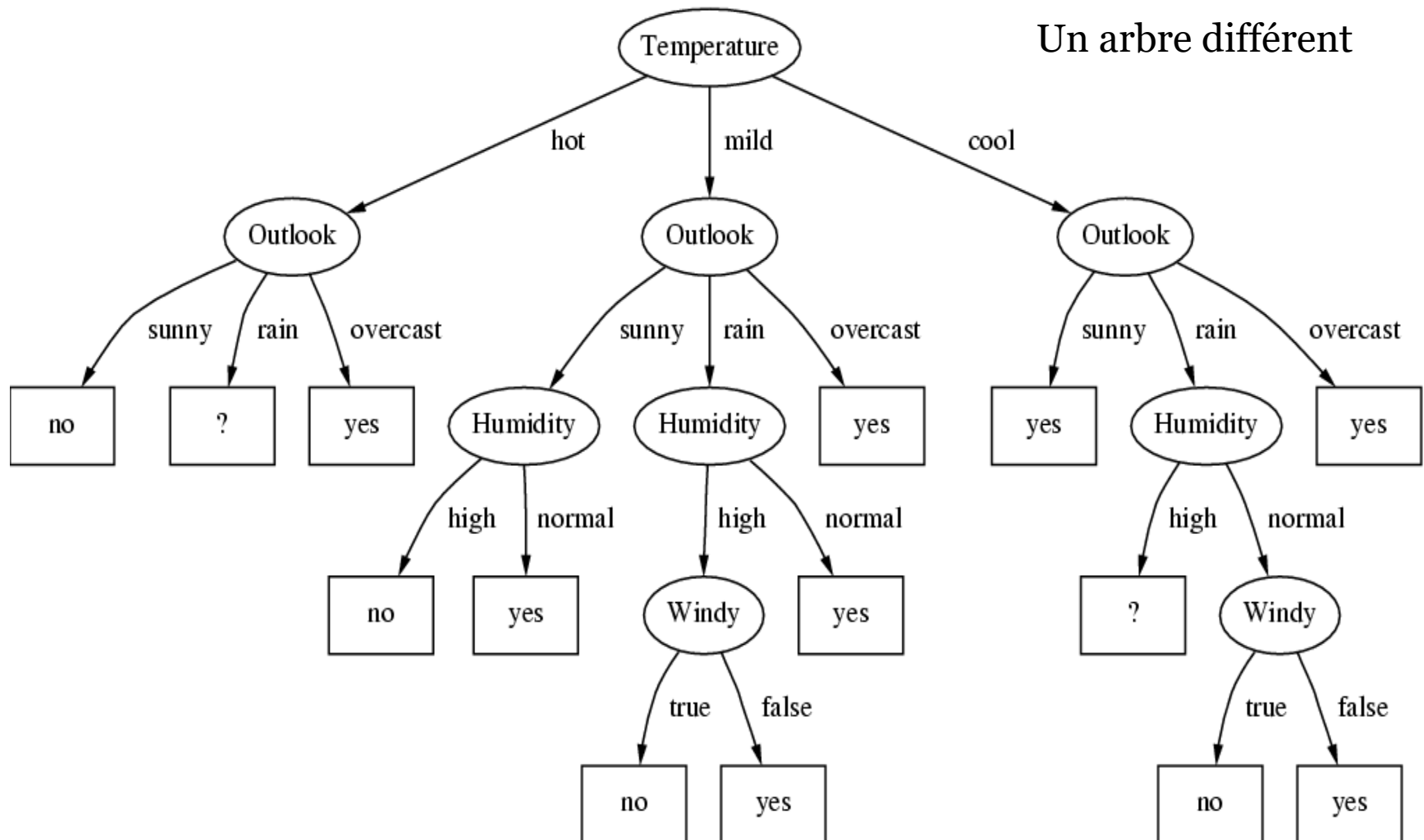
Etapes générales : **Le meilleur, le plus pur, comment ?**



1. **Sélectionner un attribut** pour le nœud racine.
  - Créez une branche pour chaque valeur possible de l'attribut.
2. Diviser les exemples en sous-ensembles.
  - Un pour chaque branche s'étendant à partir du nœud.
3. Répéter récursivement pour chaque branche, en utilisant uniquement les exemples qui atteignent la branche.
4. Arrêter la récursivité pour une branche si tous ses exemples ont la même classe.

# Les arbres de décision

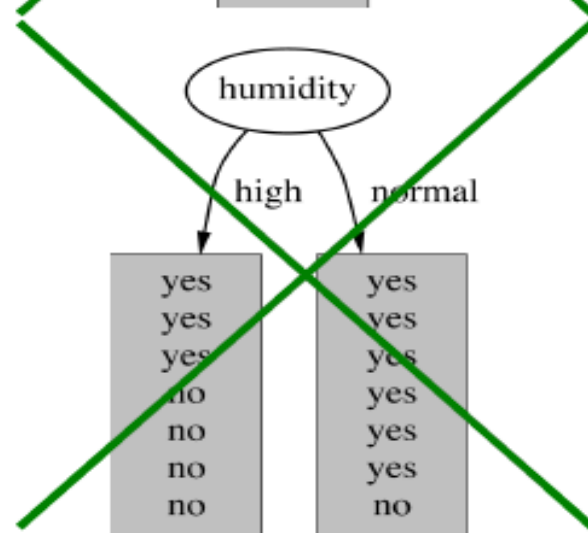
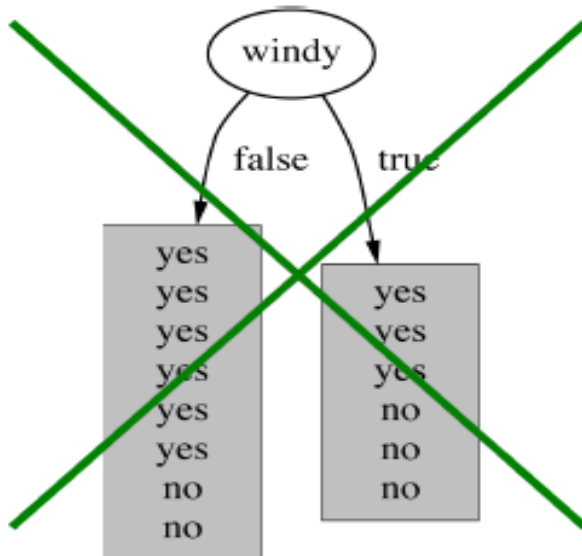
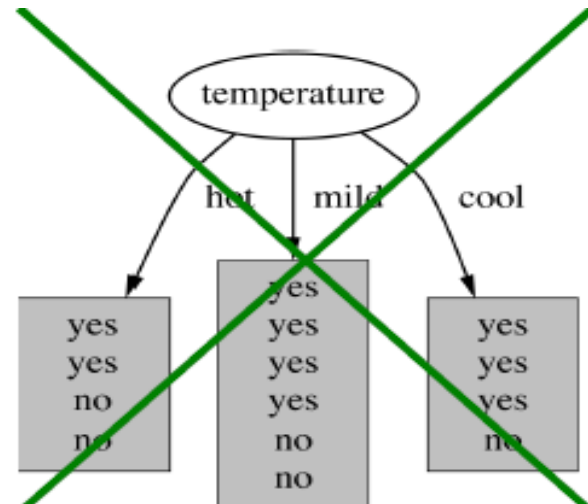
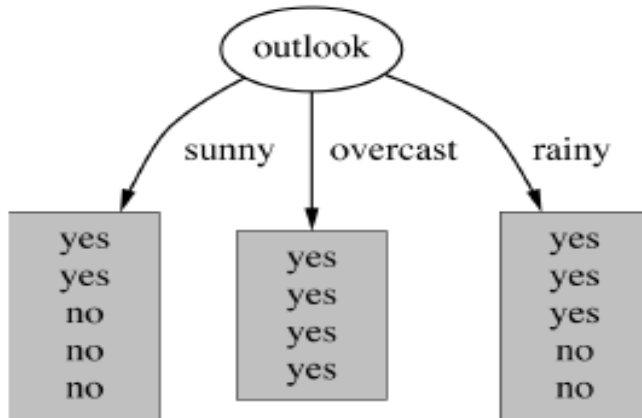
## Algorithmes de construction d'arbres de décision



# Les arbres de décision

## Algorithmes de construction d'arbres de décision

Quel attribut choisir  
comme racine ?



# Les arbres de décision

## Algorithmes de construction d'arbres de décision

Plusieurs problèmes :

Quel attribut choisir  
comme racine ?

- Comment choisir l'attribut qui sépare le mieux l'ensemble d'exemples? On parle souvent de la **variable de segmentation (Split)**.
- Comment choisir les critères de séparation d'un ensemble selon l'attribut choisi, et comment ces critères varient selon que l'attribut soit numérique ou symbolique ?
- Quel est le nombre optimal du nombre de critères qui minimise la taille de l'arbre et maximise la précision ?
- Quels sont les critères d'arrêt de ce partitionnement, sachant que souvent l'arbre est d'une taille gigantesque ?

# Les arbres de décision

## Algorithmes de construction d'arbres de décision

- Comment **mesurer la pureté d'un attribut** ?
- Différentes mesures :
  - Entropie
  - Gain d'information
  - GainRatio
  - Indice de Gini
- Dépend de l'algorithme choisi.

# Les arbres de décision

## Algorithme ID3

- ID3 construit l'arbre récursivement.
- Utilise le **gain d'information** pour mesurer la pureté d'un attribut.
- Son calcul se fait à base de l'**entropie** de Shannon.
- L'algorithme suppose que tous les attributs sont **catégoriels**;
- Si des attributs sont numériques, ils doivent être discrétisés.

## Les arbres de décision

### Algorithme ID3: Pseudo-Code

- Créer nœud  $N$
- **Si** tous les exemples de  $D$  sont de la même classe  $C$  **alors**  
Retourner  $N$  comme une feuille étiquetée par  $C$  ;
- **Si** la liste des attributs est vide **alors**  
Retourner  $N$  Comme une feuille étiquetée de la classe de la majorité dans  $D$  ;
- Sélectionner l'attribut  $A$  du meilleur **Gain** dans  $D$  ;
- Etiqueter  $N$  par l'attribut sélectionné ;
- Liste d'attributs  $\leftarrow$  Liste d'attributs -  $A$  ;
- **Pour** chaque valeur  $V_i$  de  $A$  **Faire**
  - Soit  $D_i$  l'ensemble d'exemples de  $D$  ayant la valeur de  $A = V_i$  ;
  - Attacher à  $N$  le sous arbre généré par l'ensemble  $D_i$  et la liste d'attributs
- **FinPour** ;
- **Fin** ;

# Les arbres de décision

## Choix d'attribut: Entropie

- Supposons qu'il y a deux classes : **Yes** (+) et **No** (-).
- Soit l'ensemble d'exemples **S** contenant  $p$  exemples de la classe **Yes** et  $n$  exemples de la classe **No**.
- L'entropie est la quantité d'information nécessaire pour décider qu'un exemple dans **S** appartienne à **Yes** ou **No**.
- Elle est définie par :

$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

Où,  $p_+$  est la proportion des exemples **Yes**,

et  $p_-$  est la proportion des exemples **No**

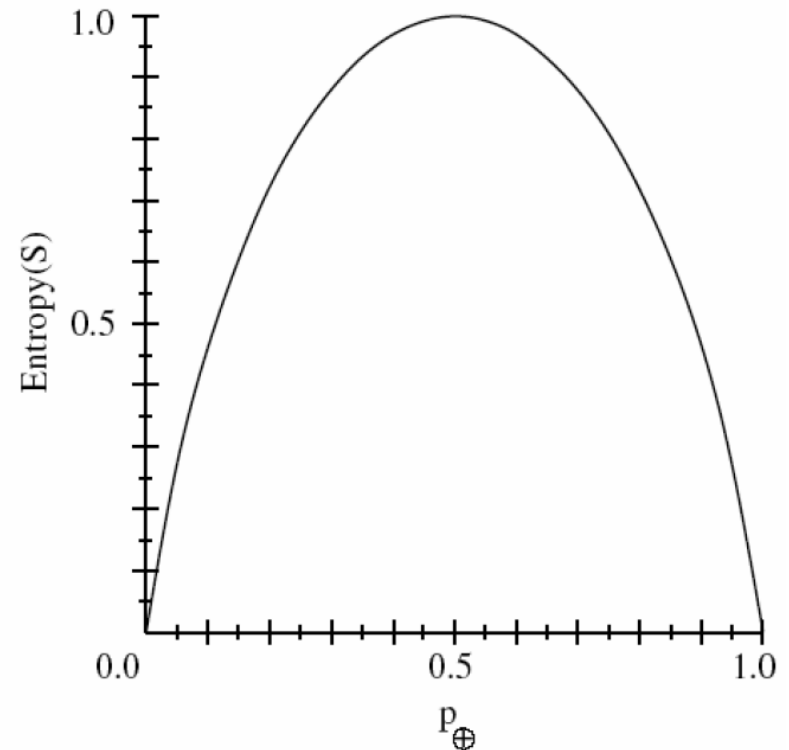


# Les arbres de décision

## Choix d'attribut: Entropie

$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

- Se mesure en **bits**.
- Si tous exemples sont soit tous **Yes**, soit tous **No**, l'entropie est nulle.
- Si  $p_+ = p_- = 0.5$  alors l'entropie est égale à 1.



Exemple : **9+** **5-**

$$E(S) = E([9+, 5-]) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.940$$

# Les arbres de décision

## Choix d'attribut: Entropie

$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

Training examples: **9 yes / 5 no**

- Se mesure en **bits**.
- Si tous exemples sont soit tous **Yes**, soit tous **No**, l'entropie est nulle.
- Si  $p_+ = p_- = 0.5$  alors l'entropie est égale à 1.

Exemple : **9+** **5-**

Outlook	Humidity	Wind	Play
Sunny	High	Weak	No
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Rain	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Strong	No
Overcast	Normal	Strong	Yes
Sunny	High	Weak	No
Sunny	Normal	Weak	Yes
Rain	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Overcast	High	Strong	Yes
Overcast	Normal	Weak	Yes
Rain	High	Strong	No

$$E(S) = E([9+,5-]) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.940$$

## Les arbres de décision

**Choix d'attribut:** Entropie

$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

**Cas de plus de 2 classes :**

$$E(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \dots - p_n \log_2 p_n = -\sum_{i=1}^n p_i \log_2 p_i$$

**Entropie Maximale** =  $\log_2$  (nombre de classes)

# Les arbres de décision

## Choix d'attribut: Gain d'information

- L'entropie ne calcule que la qualité d'un seul (sous) ensemble d'exemples.
  - Correspond à une valeur unique.
- Comment calculer la qualité de l'ensemble du Split ?
  - Correspond à un attribut entier.
- => Gain d'information pour un attribut A :  **$G(S, A)$**

$$Gain(S, A) = E(S) - \sum_{v \in \text{valeurs}(A)} \frac{|S_v|}{|S|} E(S_v)$$

- L'attribut qui maximise cette différence est sélectionné.

# Les arbres de décision

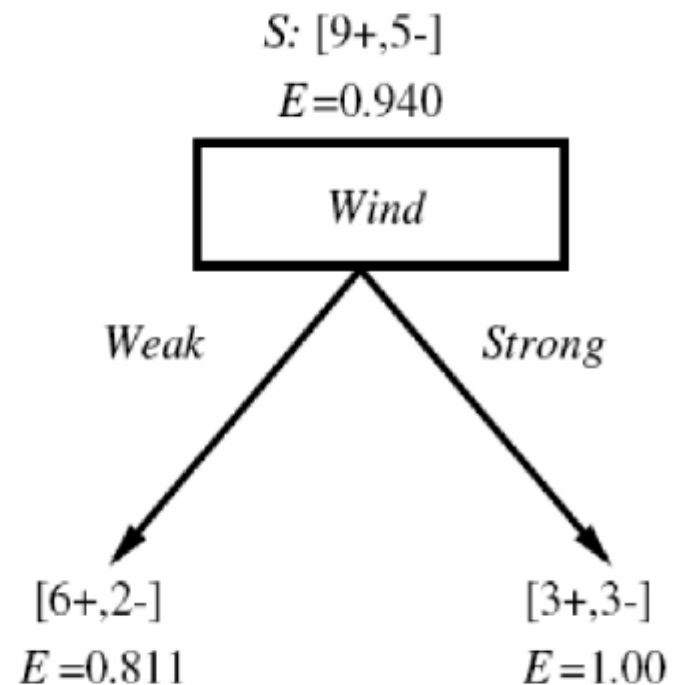
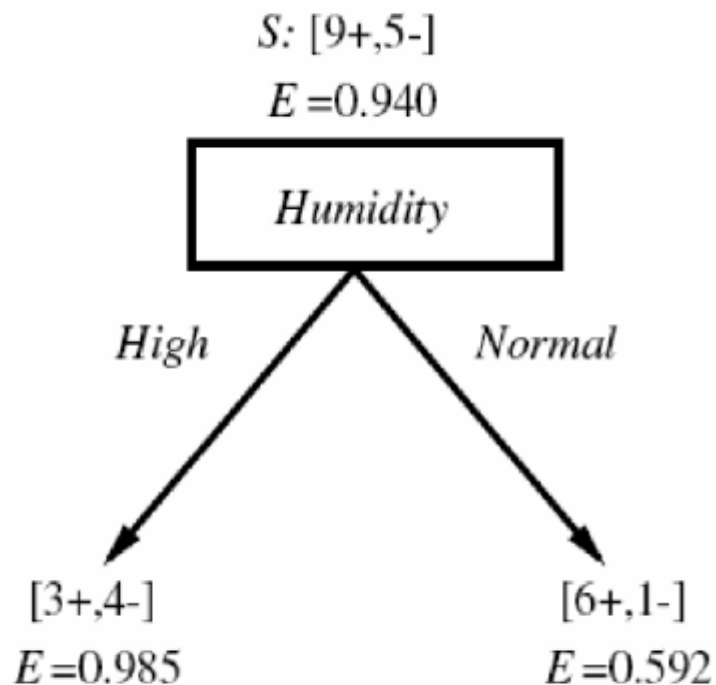
## Exemple

**Classe**

<i>Temperature</i>	<i>Outlook</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play Golf?</i>
hot	sunny	high	weak	no
hot	sunny	high	strong	no
hot	overcast	high	weak	yes
cool	rain	normal	weak	yes
cool	overcast	normal	strong	yes
mild	sunny	high	weak	no
cool	sunny	normal	weak	yes
mild	rain	normal	weak	yes
mild	sunny	normal	strong	yes
mild	overcast	high	strong	yes
hot	overcast	normal	weak	yes
mild	rain	high	strong	no
cool	rain	normal	strong	no
mild	rain	high	weak	yes

## Les arbres de décision

### Choix d'attribut: Gain d'information



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14).985 - (7/14).592 \\ &= .151 \end{aligned}$$

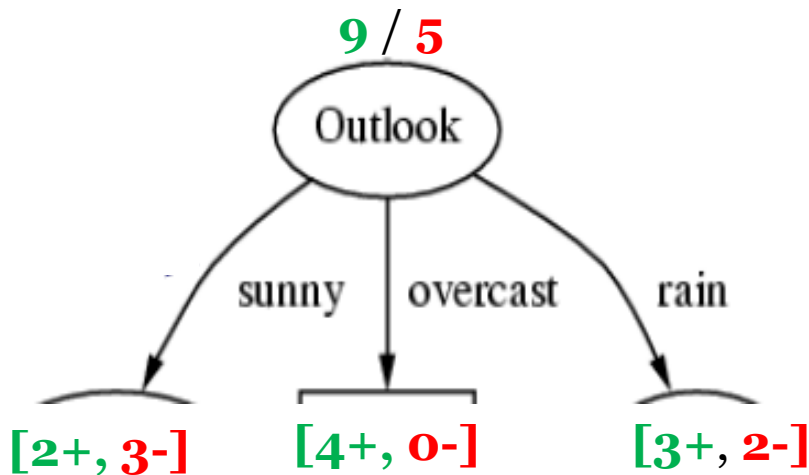
$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$

$$\text{Gain}(S, \text{Outlook}) = 0.246$$

$$\text{Gain}(S, \text{Temperature}) = 0.029$$

## Les arbres de décision

### Choix d'attribut: Gain d'information



$$E(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$$

$$Gain(S, A) = E(S) - \sum_{v \in \text{valeurs}(A)} \frac{|S_v|}{|S|} E(S_v)$$

*Gain (S, Humidity )*

$$\begin{aligned} &= .940 - (7/14).985 - (7/14).592 \\ &= .151 \end{aligned}$$

$$Gain(S, Outlook) = 0.246$$

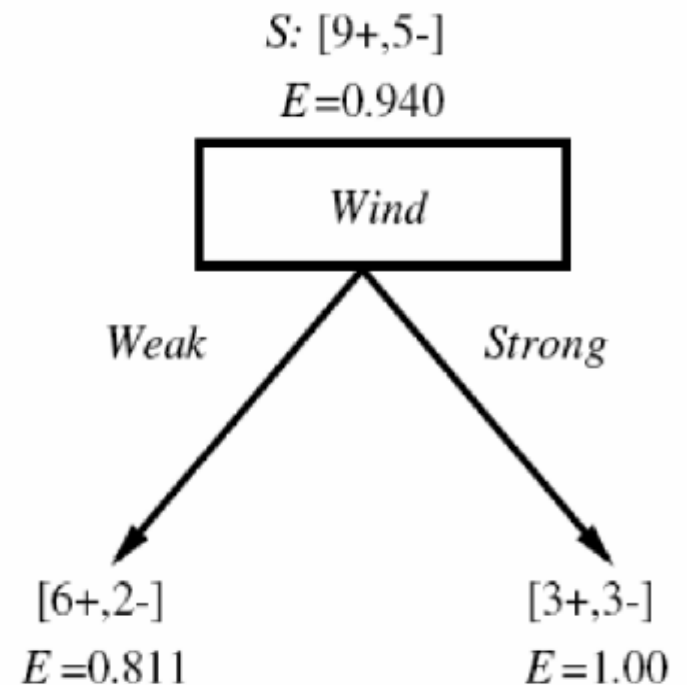
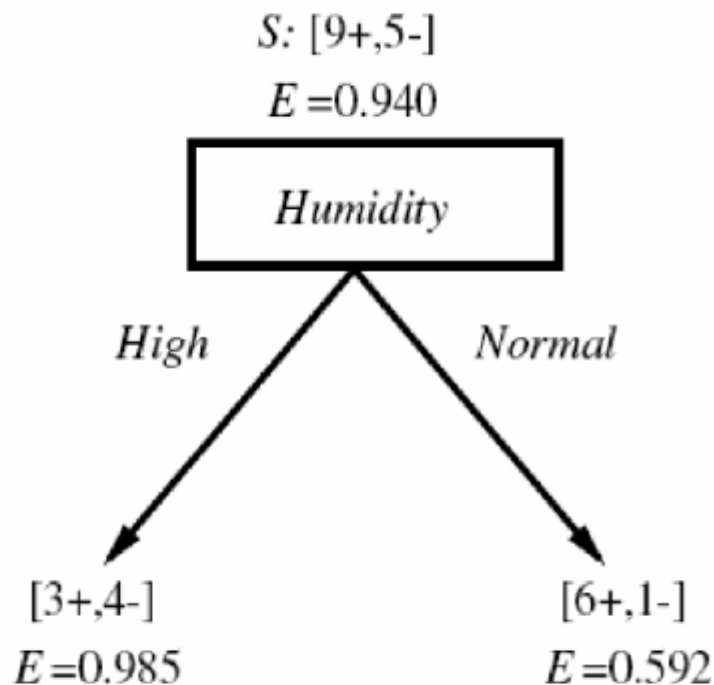
*Gain (S, Wind)*

$$\begin{aligned} &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$

$$Gain(S, Temperature) = 0.029$$

## Les arbres de décision

### Choix d'attribut: Gain d'information



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14).985 - (7/14).592 \\ &= .151 \end{aligned}$$

$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$

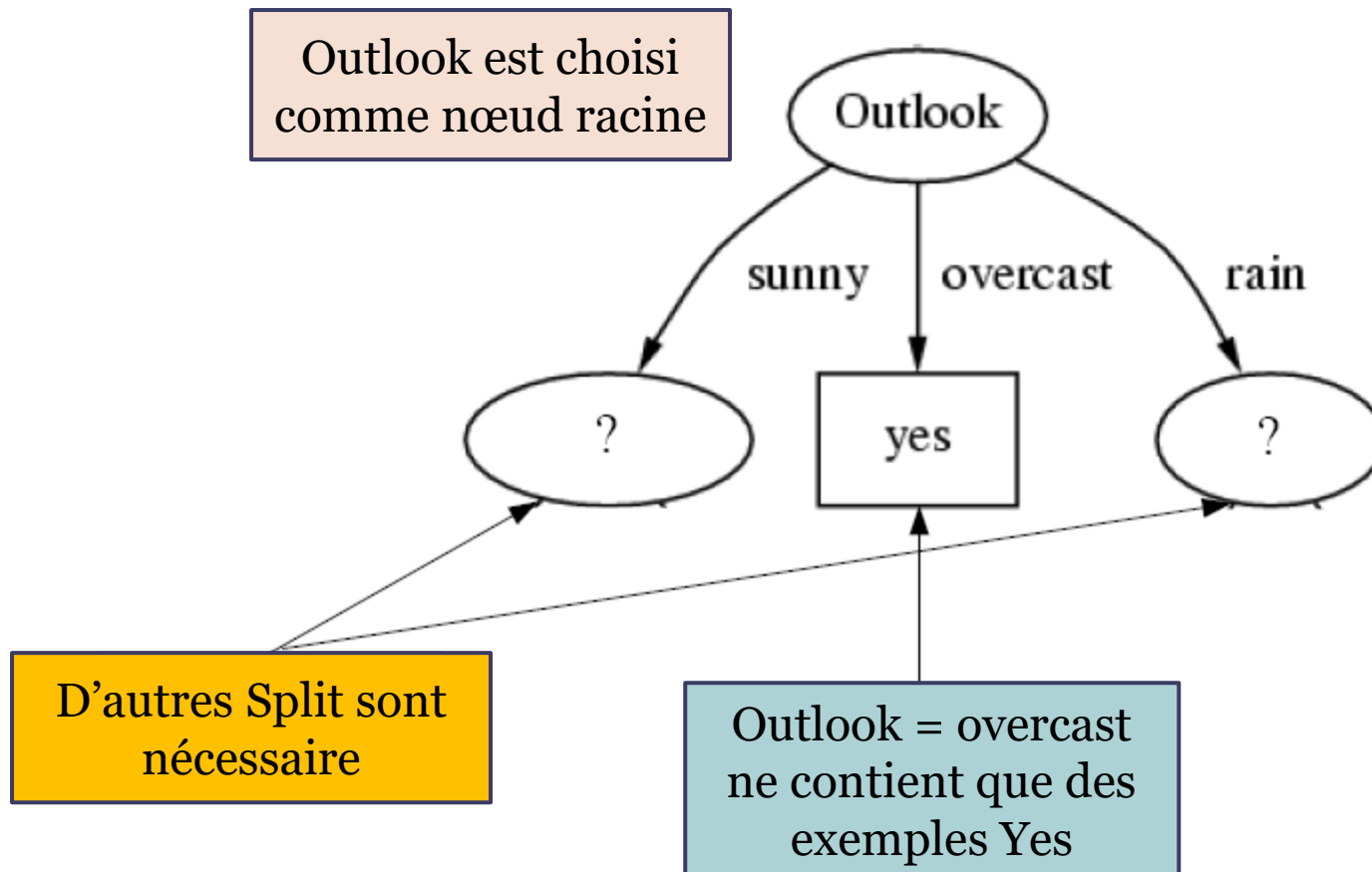
$$\text{Gain}(S, \text{Outlook}) = 0.246$$

$$\text{Gain}(S, \text{Temperature}) = 0.029$$



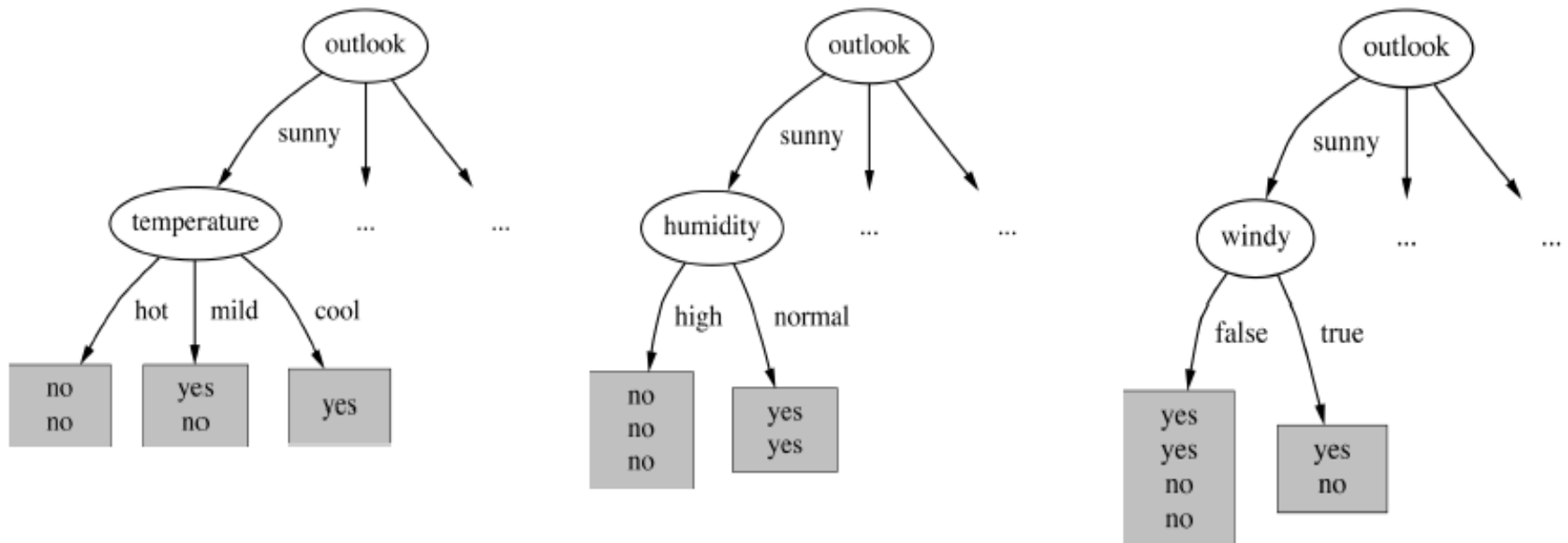
# Les arbres de décision

**Choix d'attribut:** Gain d'information



## Les arbres de décision

### Choix d'attribut: Gain d'information



$\text{Gain}(\text{Temperature})$

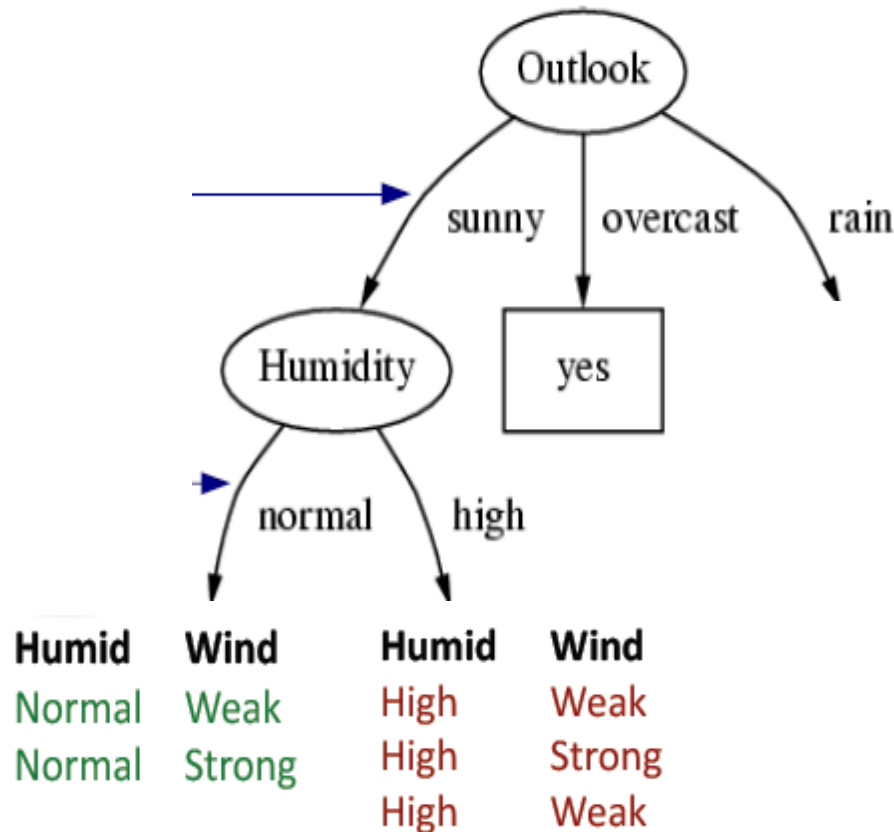
$\text{Gain}(\text{Humidity})$

$\text{Gain}(\text{Windy})$

$$S \Rightarrow S_{\text{sunny}}$$

## Les arbres de décision

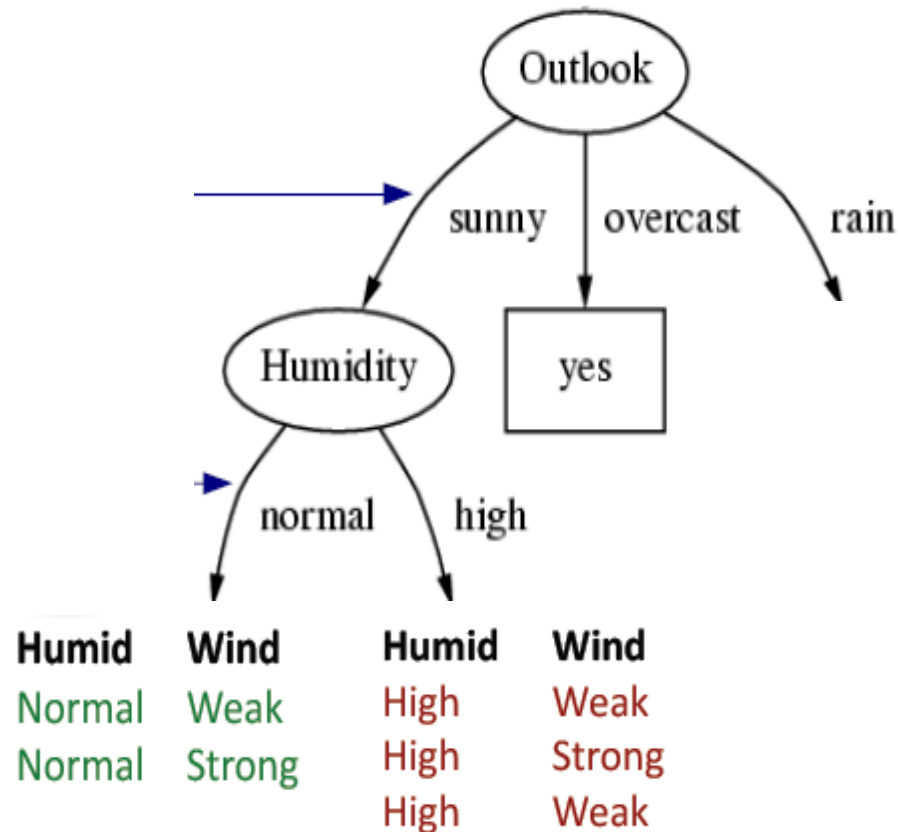
**Choix d'attribut:** Gain d'information



$$Gain(S_{Sunny}, Humidity) = E(S_{Sunny}) - \sum_{v \in \text{valeurs}(Humidity)} \frac{|S_{Sunny-v}|}{|S|} E(S_{Sunny-v})$$

## Les arbres de décision

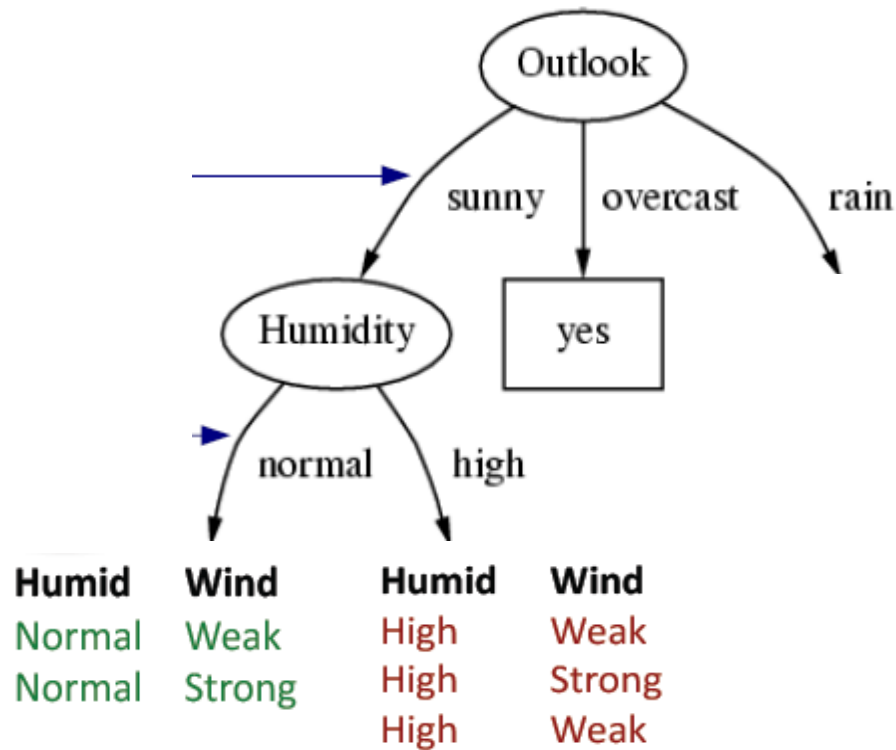
**Choix d'attribut:** Gain d'information



$$E(S_{Sunny}) = E([2+, 3-]) = -\frac{2}{5} \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \log_2\left(\frac{3}{5}\right) = 0.971$$

## Les arbres de décision

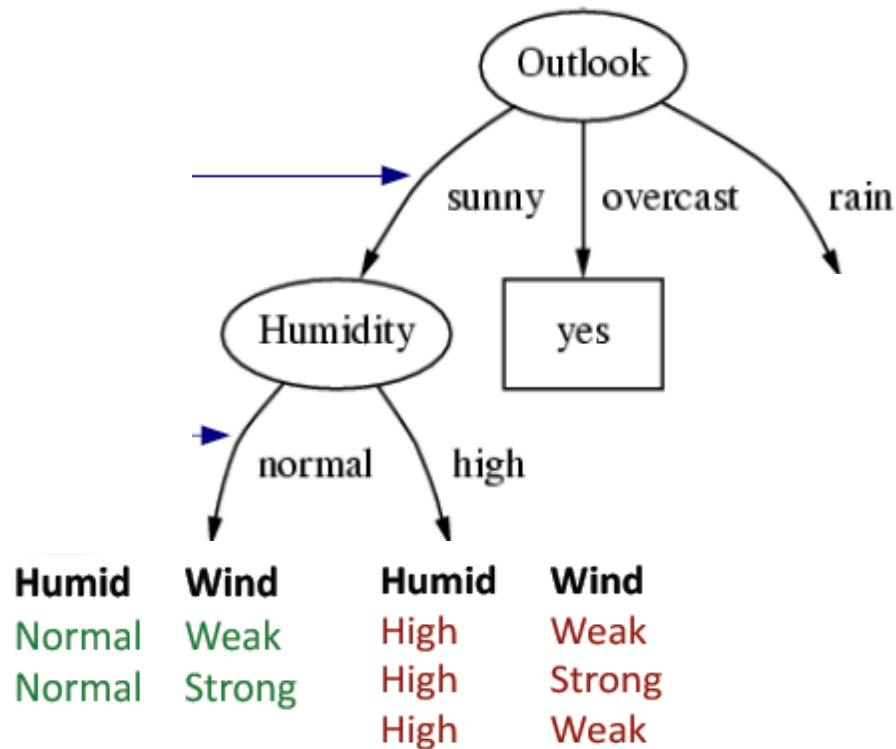
**Choix d'attribut:** Gain d'information



$$\sum_{v \in \text{valeurs}(\text{Humidity})} \frac{|S_{\text{Sunny}-v}|}{|S|} E(S_{\text{Sunny}-v})$$

## Les arbres de décision

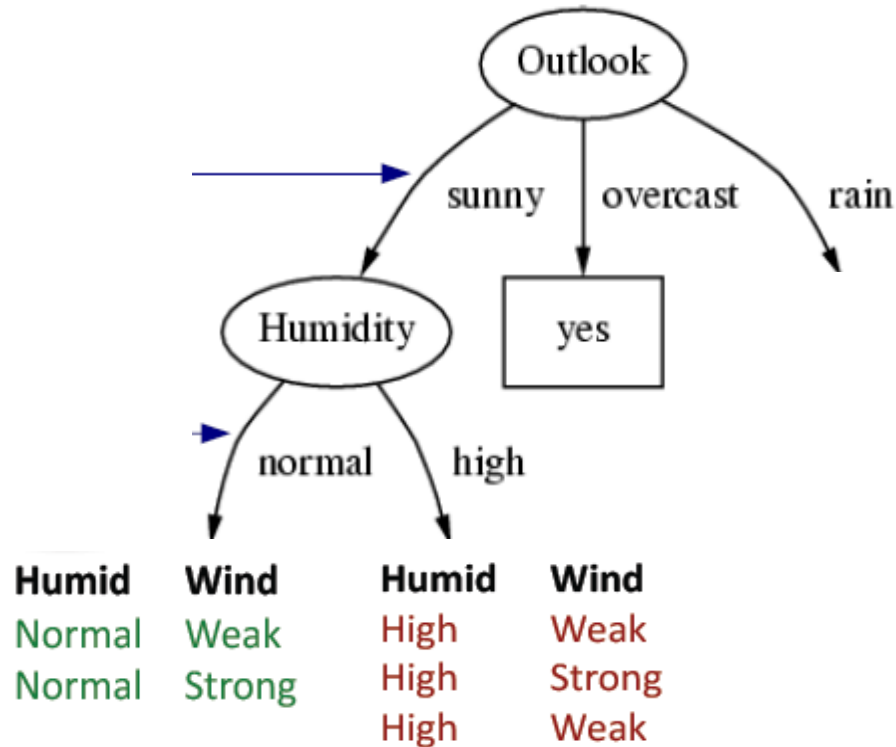
**Choix d'attribut:** Gain d'information



$$= \frac{|S_{Sunny-High}|}{|S_{Sunny}|} E(S_{Sunny-High}) + \frac{|S_{Sunny-Normal}|}{|S_{Sunny}|} E(S_{Sunny-Normal})$$

## Les arbres de décision

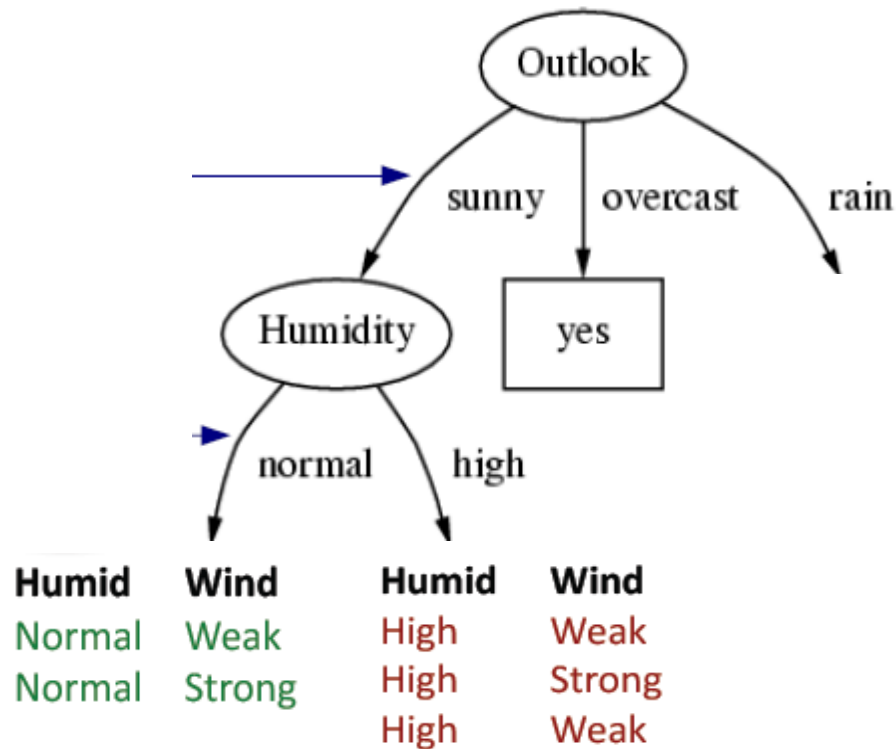
**Choix d'attribut:** Gain d'information



$$= \frac{3}{5} E([3+, 0-]) + \frac{2}{5} E([0+, 2-])$$

## Les arbres de décision

**Choix d'attribut:** Gain d'information

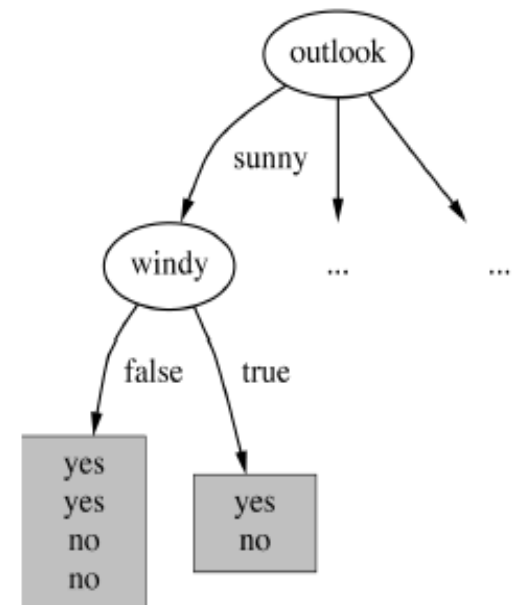
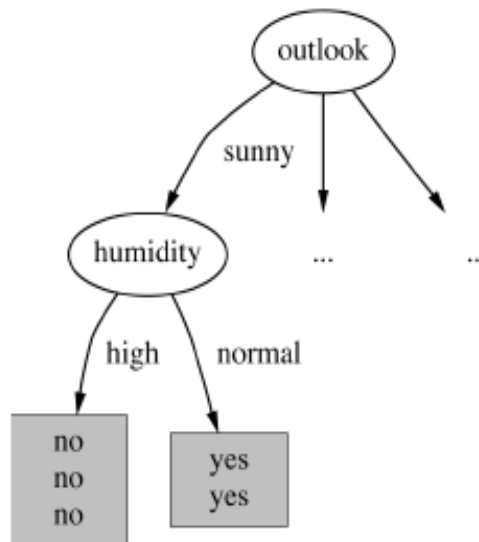
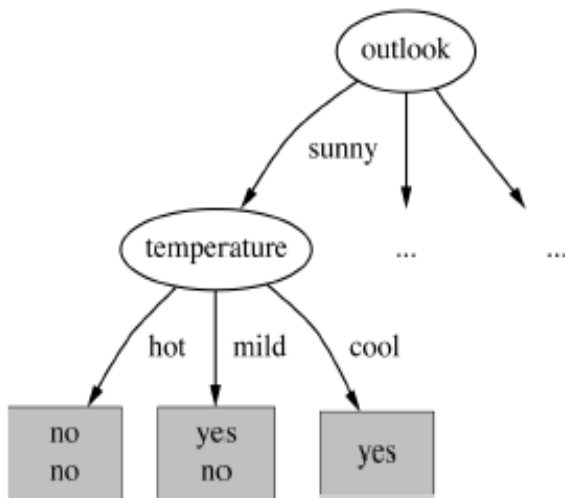


$$= 0.971 - \left(\frac{3}{5} * 0\right) + \frac{2}{5} * 0 = 0.971$$



## Les arbres de décision

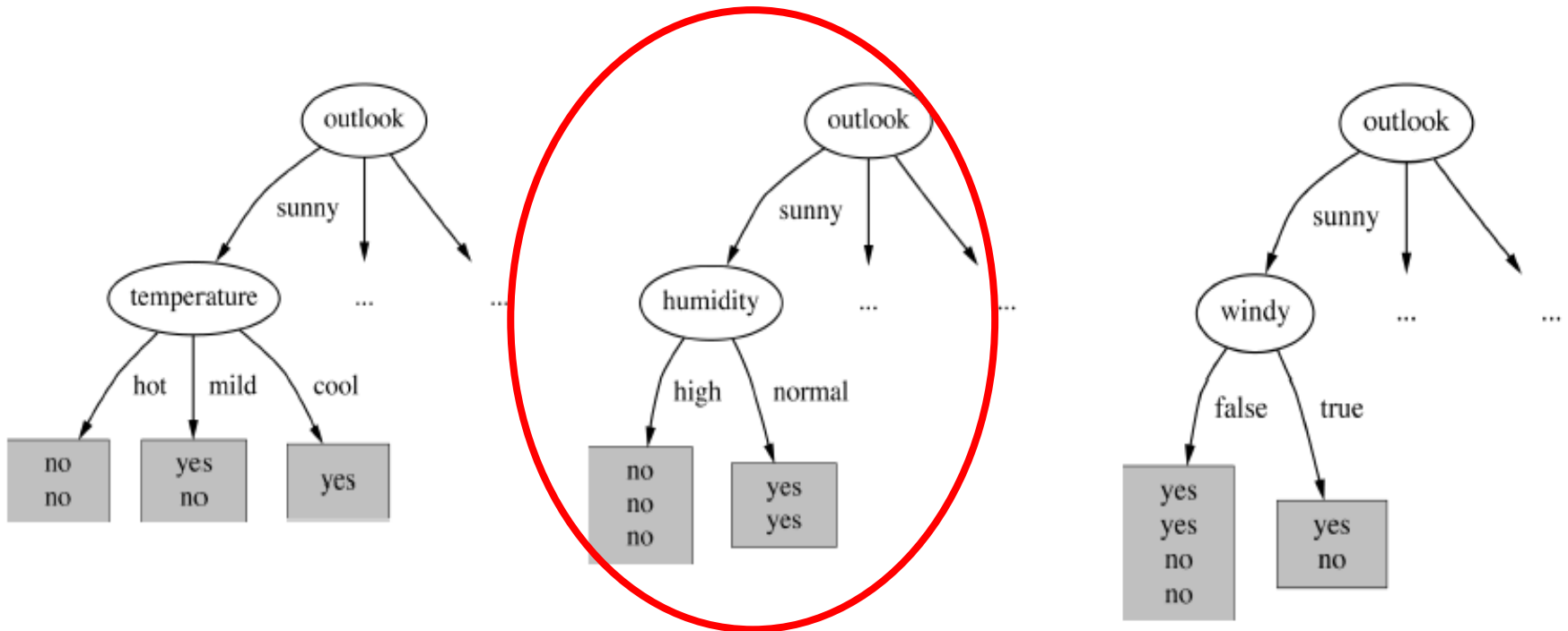
### Choix d'attribut: Gain d'information



$$\left. \begin{array}{ll} \text{Gain}(\text{Temperature}) & = 0.571 \text{ bits} \\ \text{Gain}(\text{Humidity}) & = 0.971 \text{ bits} \\ \text{Gain}(\text{Windy}) & = 0.020 \text{ bits} \end{array} \right\}$$

## Les arbres de décision

### Choix d'attribut: Gain d'information



$\text{Gain}(\text{Temperature}) = 0.571 \text{ bits}$

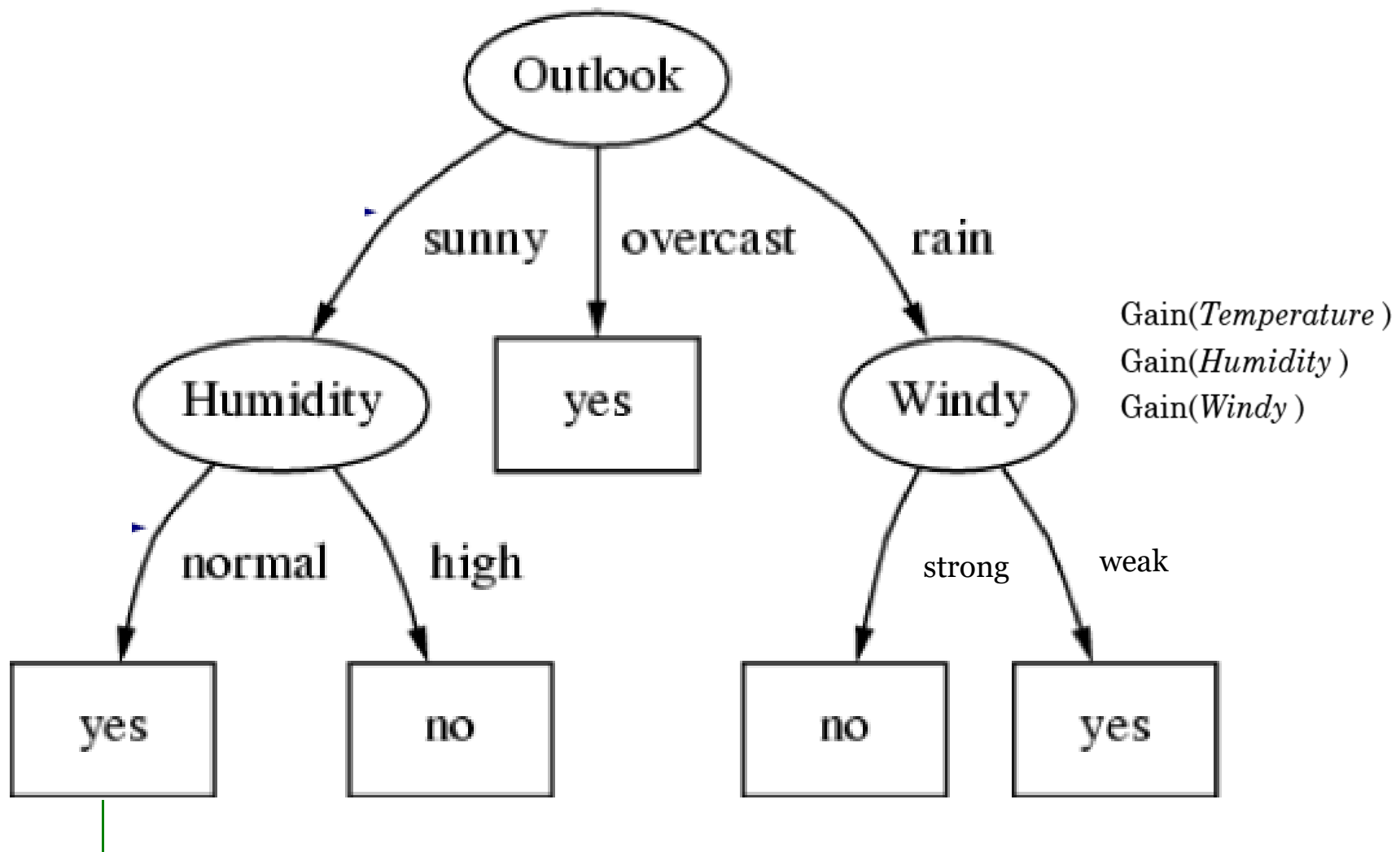
$\text{Gain}(\text{Humidity}) = 0.971 \text{ bits}$

$\text{Gain}(\text{Windy}) = 0.020 \text{ bits}$

Humidity est choisi

# Les arbres de décision

## Exemple



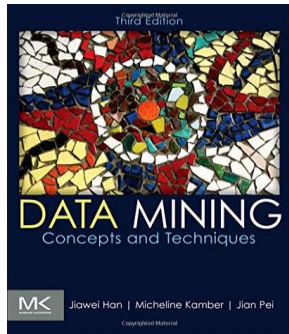
# Les arbres de décision

## D'autres algorithmes

- Algorithme C4.5 (J48)
  - Amélioration de ID3
  - Prends en compte les attributs numériques.
  - Utilise le GainRatio pour le Split.
- Algorithme CART
  - CART : Classification And Regression Trees.
  - Utilise l'indice de Gini pour le Split/Segmentation.
- Forêts aléatoires (Random Forests)
  - Plus efficaces mais difficilement interprétables.
  - Construction des arbres se base sur le Bootstrap (le Bagging).

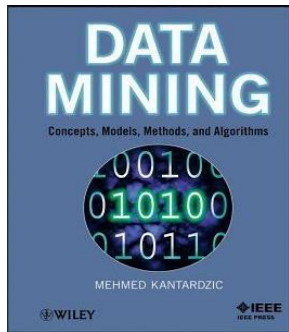


# Ressources



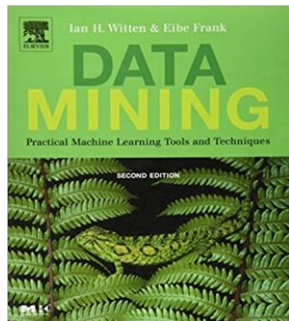
## **Data Mining : concepts and techniques, 3rd Edition**

- ✓ Auteur : Jiawei Han, Micheline Kamber, Jian Pei
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2011 - 744 pages - ISBN 9780123814807



## **Data Mining : concepts, models, methods, and algorithms**

- ✓ Auteur : Mehmed Kantardzi
- ✓ Éditeur : John Wiley & Sons
- ✓ Edition : Aout 2011 – 552 pages - ISBN : 9781118029121



## **Data Mining: Practical Machine Learning Tools and Techniques**

- ✓ Auteur : Ian H. Witten & Eibe Frank
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2005 - 664 pages - ISBN : 0-12-088407-0

# Ressources

Cours – Abdelhamid DJEFFAL – Fouille de données avancée

✓ [www.abdelhamid-djeffal.net](http://www.abdelhamid-djeffal.net)

WekaMOOC – Ian Witten – Data Mining with Weka

✓ <https://www.youtube.com/user/WekaMOOC/featured>

Cours - Laboratoire ERIC Lyon - DATA MINING et DATA SCIENCE

✓ [https://eric.univ-lyon2.fr/~ricco/cours/supports\\_data\\_mining.html](https://eric.univ-lyon2.fr/~ricco/cours/supports_data_mining.html)

Gregory Piatetsky-Shapiro - KDNuggets

✓ <http://www.kdnuggets.com/>