

Raisonnement (1)

Une règle



Deduction



Un fait

Raisonnement certain

Plusieurs faits *



Induction

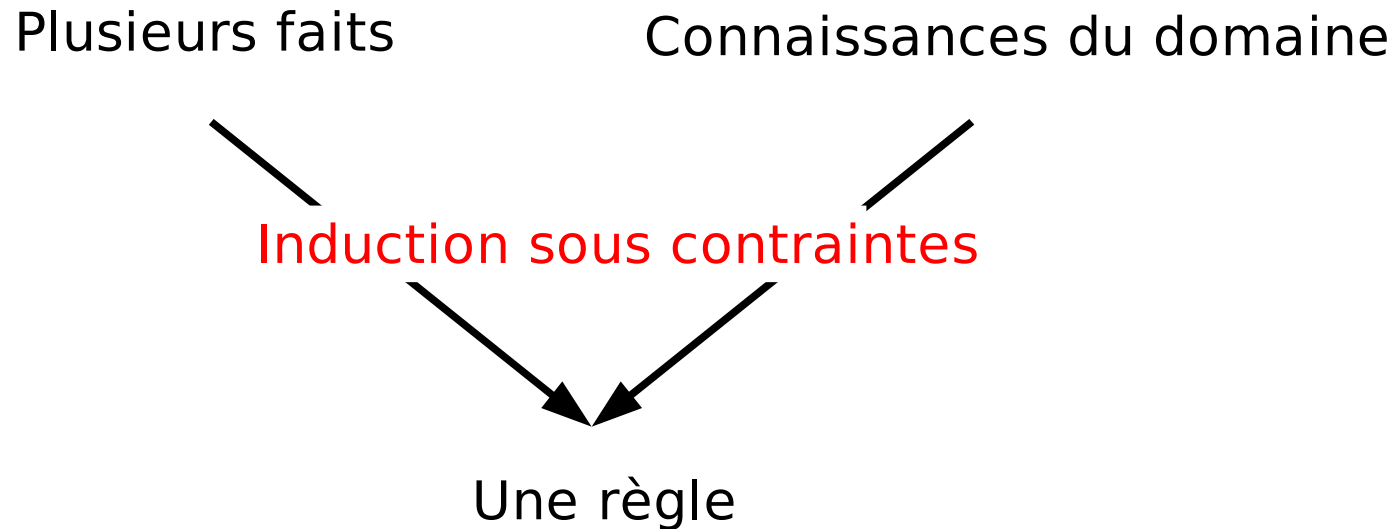


Une règle

Raisonnement incertain

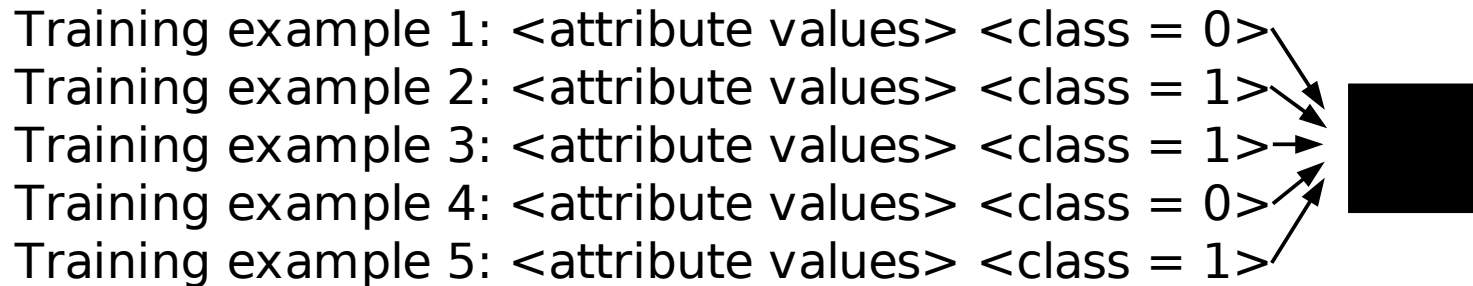
* La plupart du temps. Pensez au nombre requis pour la signifiante statistique, mais pensez également à ce que vous concluez en remarquant qu'un magasin est fermé un lundi.

Raisonnement (2)

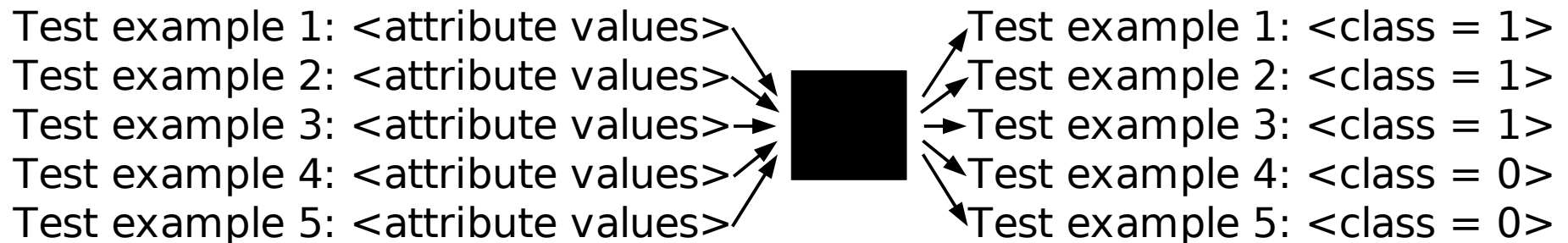


Le data mining sert à **comprendre** et/ou **modéliser** le monde décrit par la base de données.
La recherche en data mining est un domaine de la recherche en Intelligence Artificielle.

Construire un classifieur



Tester ou utiliser a classifieur



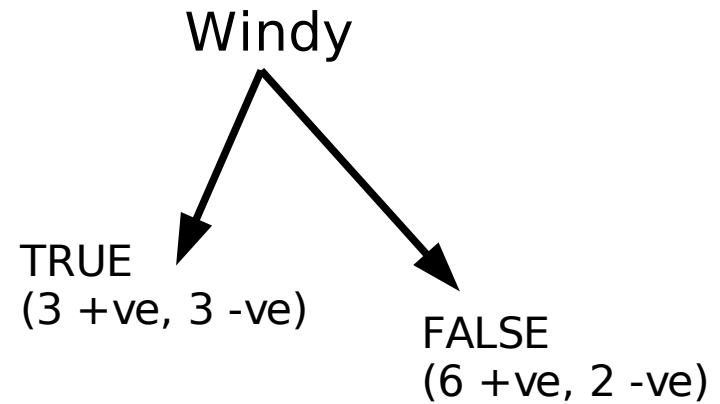
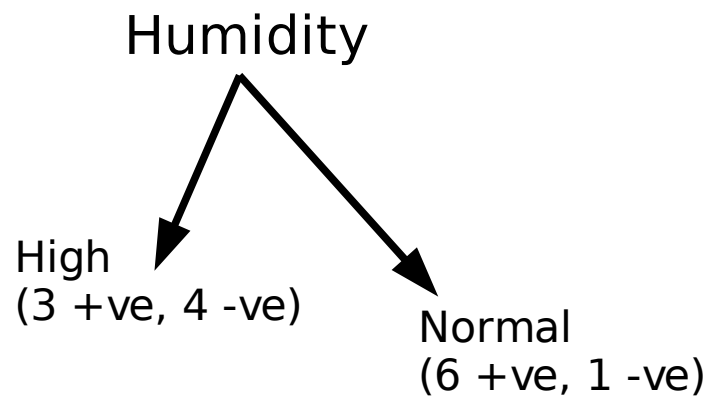
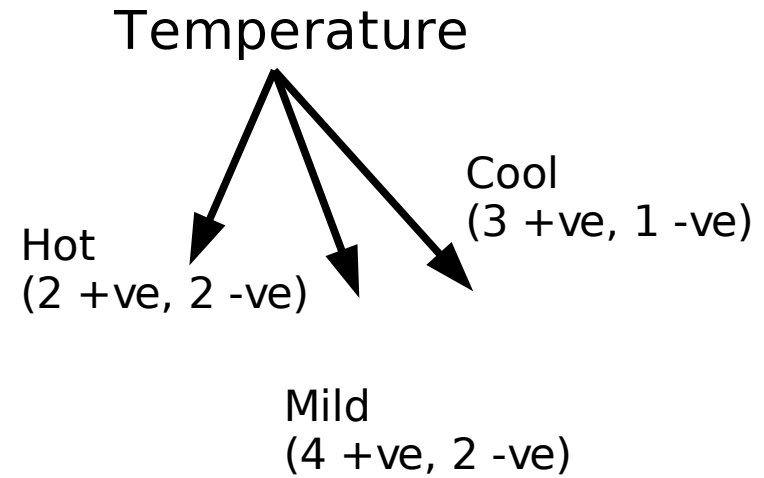
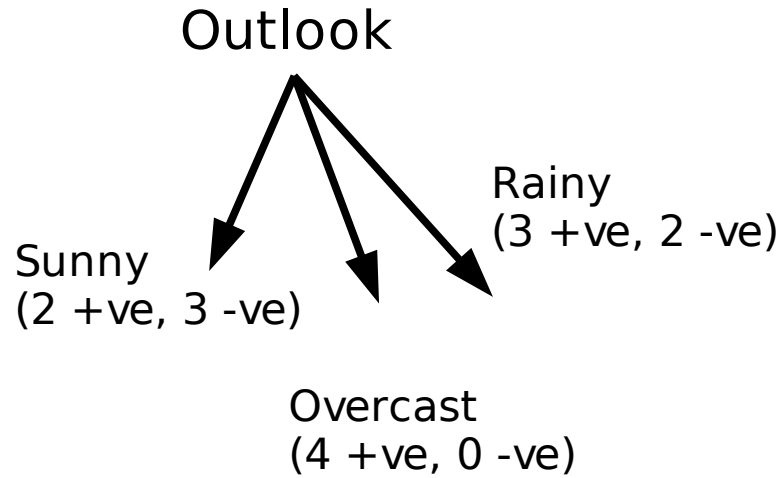
Testez **TOUJOURS** votre classifieur sur des données inconnues

The "Play Tennis" database (1)

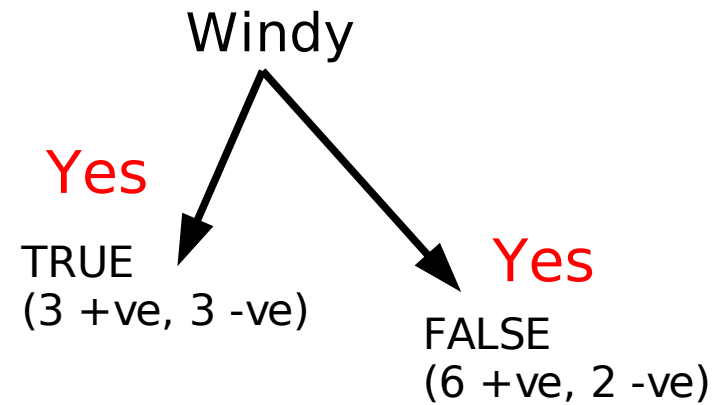
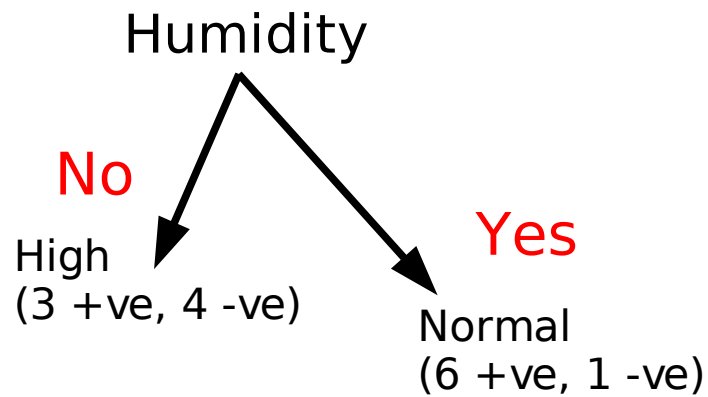
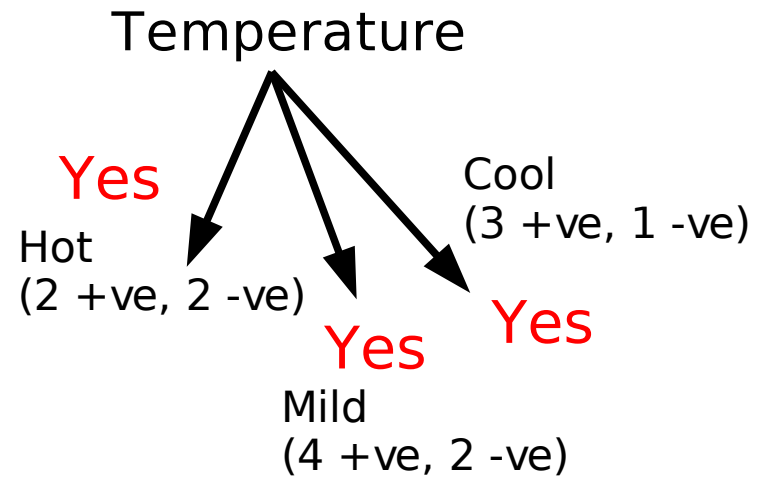
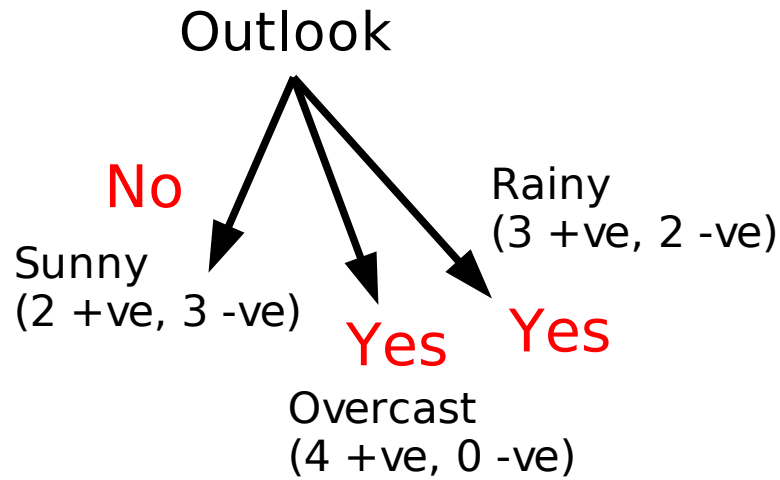
	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Class
	Outlook	Temperature	Humidity	Windy	Play
Example 1	sunny	hot	high	FALSE	no
Example 2	sunny	hot	high	TRUE	no
Example 3	overcast	hot	high	FALSE	yes
Example 4	rainy	mild	high	FALSE	yes
Example 5	rainy	cool	normal	FALSE	yes
Example 6	rainy	cool	normal	TRUE	no
Example 7	overcast	cool	normal	TRUE	yes
Example 8	sunny	mild	high	FALSE	no
Example 9	sunny	cool	normal	FALSE	yes
Example 10	rainy	mild	normal	FALSE	yes
Example 11	sunny	mild	normal	TRUE	yes
Example 12	overcast	mild	high	TRUE	yes
Example 13	overcast	hot	normal	FALSE	yes
Example 14	rainy	mild	high	TRUE	no

O = Overcast, T = hot, H = high, W = TRUE. Play Tennis?

The "Play Tennis" database (2)



The "Play Tennis" database (3)



Ceci sont des **souches de décision**

The "Play Tennis" database (4)

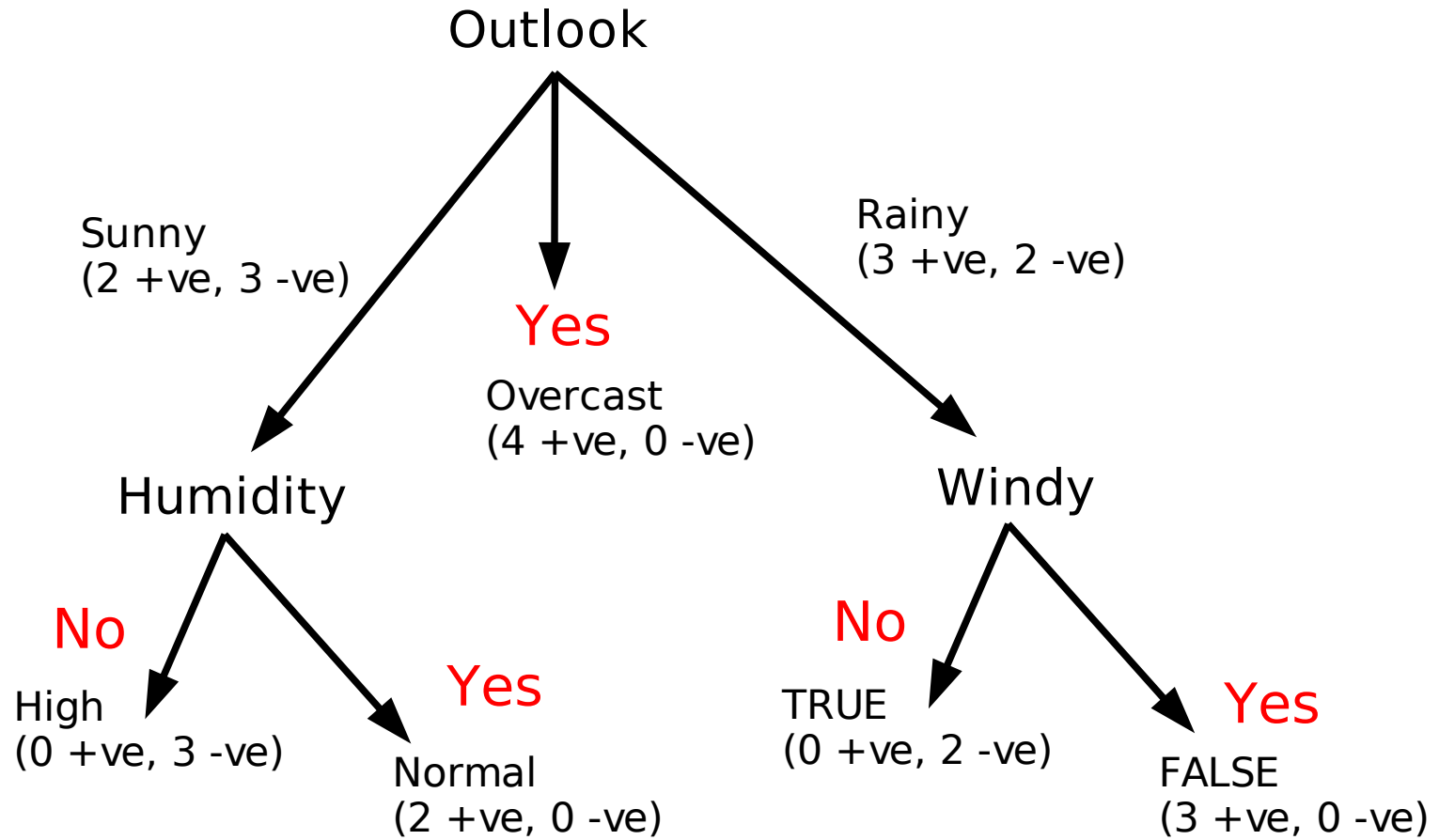
Outlook = Sunny

Temperature	Humidity	Windy	Play
cool	normal	FALSE	yes
hot	high	TRUE	no
mild	normal	TRUE	yes
mild	high	FALSE	no
hot	high	FALSE	no

Outlook = Rainy

Temperature	Humidity	Windy	Play
cool	normal	FALSE	yes
cool	normal	TRUE	no
mild	normal	FALSE	yes
mild	high	TRUE	no
mild	high	FALSE	yes

The "Play Tennis" database (5)

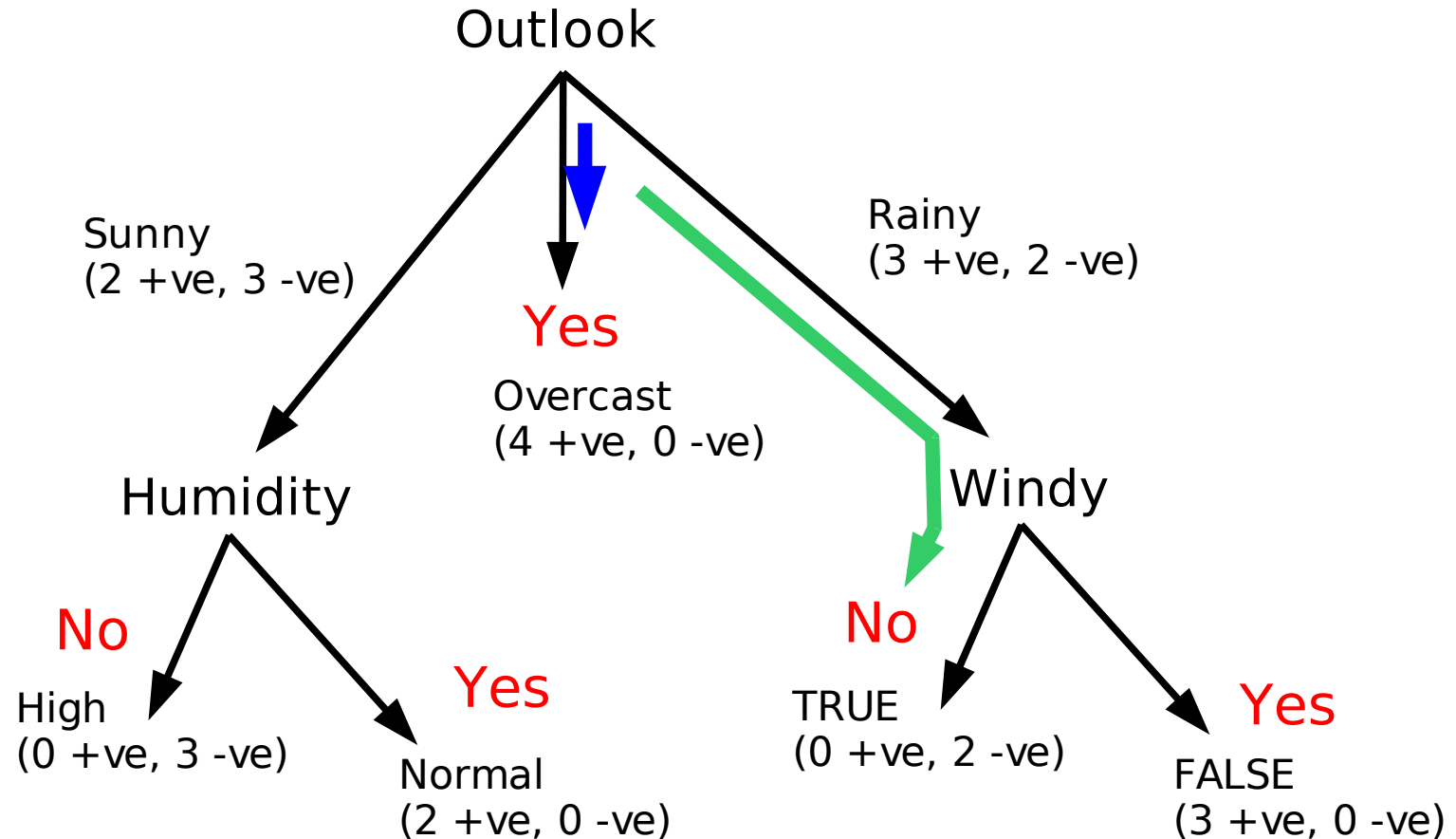


Ceci est un **arbre de décision**

The "Play Tennis" database (6)

O = Overcast, T = hot, H = high, W = TRUE. Play = Yes

O = Rainy, T = hot, H = high, W = TRUE. Play = No



Utiliser un classifieur pour faire des predictions est la **classification**

Conclusion

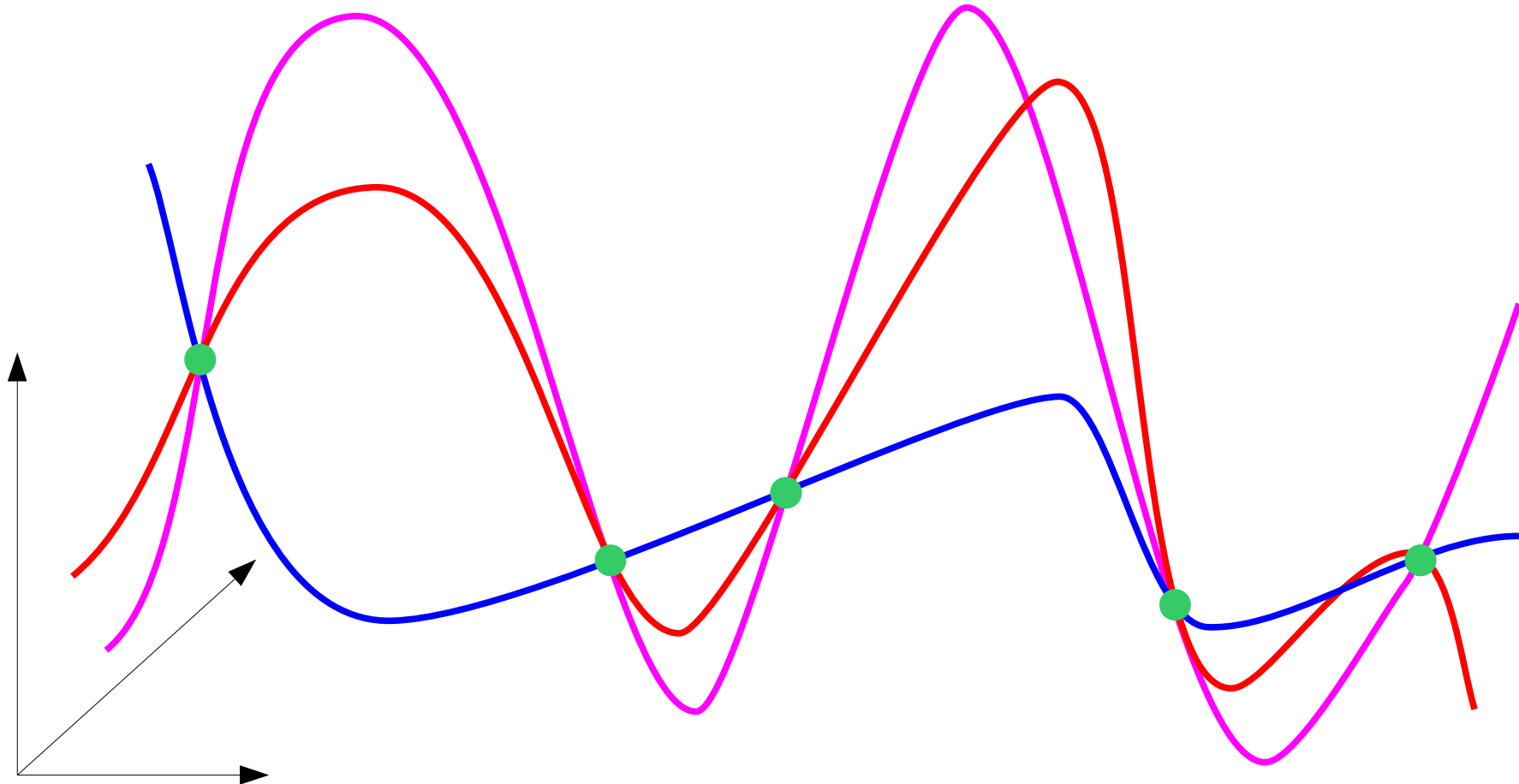
Tout arbre consistant avec les données est correct !

Il y a plus d'une façon de le faire

“There is no such thing as a free lunch”: The No Free Lunch Theorem (NFL) states that no single algorithm outperforms random search (or systematic linear search) when amortised over all functions. (Wolpert and Macready, No Free Lunch Theorems, 1995 and Davis H. Wolpert, The Supervised Learning No-Free-Lunch Theorems, 2001).

Les Théorèmes “No Free Lunch” (1)

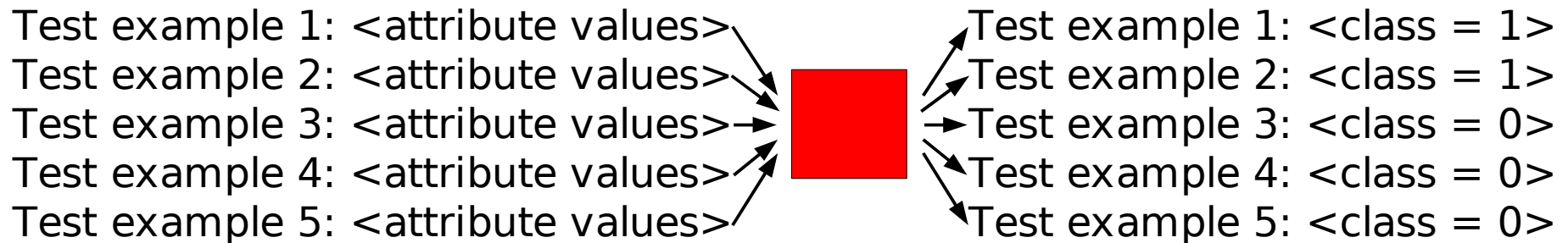
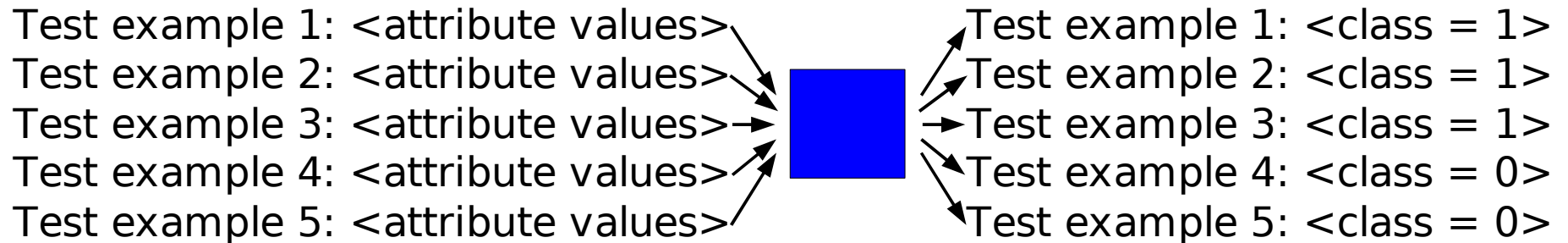
Intuition pour des données continues (numériques)



fonction de classification (un exemple inconnu) = la class de cet exemple

Les Théorèmes “No Free Lunch” (2)

Intuition pour des données discrètes (nominales)



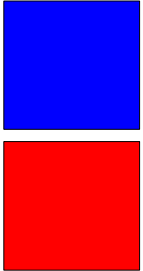
Test example 1: <attribute values> <class = 1>

Test example 2: <attribute values> <class = 1>

Test example 3: <attribute values> <class = 1>

Test example 4: <attribute values> <class = 0>

Test example 5: <attribute values> <class = 1>

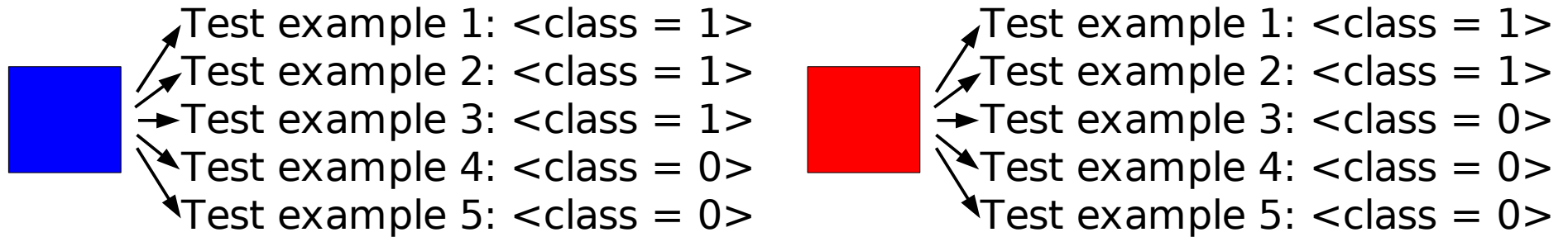


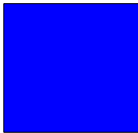
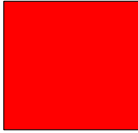
Accuracy = 80%

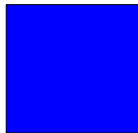
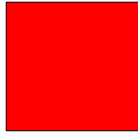
Accuracy = 60%

Les Théorèmes “No Free Lunch” (3)

Intuition pour des données discrètes (nominales)



Test example 1: <attribute values> <class = 1>		Accuracy = 80%
Test example 2: <attribute values> <class = 1>		
Test example 3: <attribute values> <class = 1>		
Test example 4: <attribute values> <class = 0>		Accuracy = 60%
Test example 5: <attribute values> <class = 1>		

Test example 1: <attribute values> <class = 0>		Accuracy = 20%
Test example 2: <attribute values> <class = 0>		
Test example 3: <attribute values> <class = 0>		
Test example 4: <attribute values> <class = 1>		Accuracy = 40%
Test example 5: <attribute values> <class = 0>		

Les Théorèmes “No Free Lunch” (4)

Les théorèmes “no free lunch” démontrent que la moyenne de l'exactitude de n'importe quel algorithme de classification, sur toutes les bases de données possibles (divisées entre des ensembles d'apprentissage et des ensembles de test) est exactement 50%.

La présupposition de cet algorithme est que toutes les bases de données possibles sont équiprobables. Ceci est une prémisse valable dans le monde mathématique mais pas dans le monde réel. Quand les bases de données décrivent des phénomènes du monde réel, elles ne sont pas toutes équiprobables.

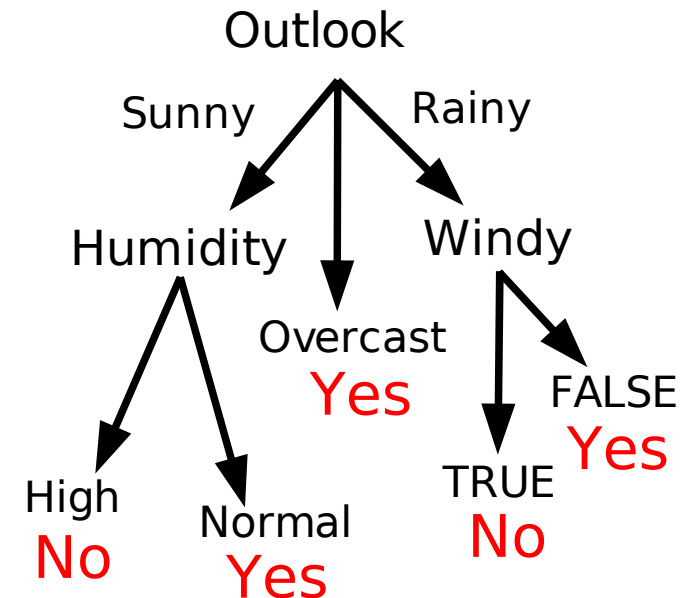
Par exemple, dans une base de données où les attributs sont des descriptifs d'animaux, si la valeur de l'attribut “A des cornes” est “Oui” pour un exemple donné, il est très improbable que la classe de cet exemple soit égale à “Chat”.

Ainsi que Rudolf Clausius énonce la seconde loi de la thermodynamique: “L'entropie de l'univers tend vers le maximum”.

The "Play Tennis" database (7)

```
IF (Outlook = overcast) THEN (Play = yes)
IF ((Outlook = sunny) AND (Humidity = high)) THEN (Play = no)
IF ((Outlook = sunny) AND (Humidity = normal)) THEN (Play = yes)
IF ((Outlook = rainy) AND (Windy = true)) THEN (Play = no)
IF ((Outlook = rainy) AND (Windy = false)) THEN (Play = yes)
```

```
IF (Outlook = overcast) THEN (Play = yes)
ELSE IF (Outlook = sunny) {
    IF (Humidity = high) THEN (Play = no)
    ELSE (Play = yes)
}
ELSE {
    IF (Windy = true) THEN (Play = no)
    ELSE (Play = yes)
}
```



Ceci sont des **règles de décision**, construites par une approche "**top-down**", aussi appelée "de haut en bas" ou "diviser pour régner"

The "Play Tennis" database (8)

Une approche **bottom-up** (de bas en haut) à l'induction de règles de décision

Commencer avec un exemple (le premier)

numéro 1: (Sunny, Hot, High, False, No)

Regarder ce qui peut être généralisé sans perdre la classe.

(* , Hot, High, False, No)? Contre-exemple le numéro 3: (Overcast, Hot, High, False, Yes)

(Sunny, * , High, False, No)? Ceci ajoute le numéro 8: (Sunny, Mild, High, False, No)

(Sunny, * , * , False, No)? Contre-exemple le numéro 9: (Sunny, Cool, Normal, False, Yes)

(Sunny, * , High, * , No)? Ceci ajoute le numéro 2: (Sunny, Hot, High, True, No)

Générer la règle:

IF ((Outlook = sunny) AND (Humidity = high)) THEN (Play = no)

Enlever les exemples couverts par la règle (ici les numéros 1, 2 and 8) de la base de données.

Recommencer.

The "Play Tennis" database (9)

	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Class
	Outlook	Temperature	Humidity	Windy	Play
Example 1	sunny	hot	high	FALSE	no
Example 2	sunny	hot	high	TRUE	no
Example 3	overcast	hot	high	FALSE	yes
Example 4	rainy	mild	high	FALSE	yes
Example 5	rainy	cool	normal	FALSE	yes
Example 6	rainy	cool	normal	TRUE	no
Example 7	overcast	cool	normal	TRUE	yes
Example 8	sunny	mild	high	FALSE	no
Example 9	sunny	cool	normal	FALSE	yes
Example 10	rainy	mild	normal	FALSE	yes
Example 11	sunny	mild	normal	TRUE	yes
Example 12	overcast	mild	high	TRUE	yes
Example 13	overcast	hot	normal	FALSE	yes
Example 14	rainy	mild	high	TRUE	no

The "Play Tennis" database (10)

Une approche **bottom-up** à l'induction de règles de décision

Recommencer avec un exemple (le premier de ceux qui restent)

numéro 3: (Overcast, Hot, High, False, Yes)

Regarder ce qui peut être généralisé sans perdre la classe.

(* , Hot, High, False, Yes)? Aucun exemple dans ce cas, continuer

(* , * , High, False, Yes)? Ceci ajoute le numéro 4: (Rainy, Mild, High, False, Yes)

(* , * , * , False, Yes)? Ceci ajoute les exemples numéro 5, 9, 10, 13

Générer la règle:

IF (Windy = false) THEN (Play = yes)

Enlever les exemples couverts par la règle (now numbers 3, 4, 5, 9, 10 and 13) de la base de données.

Recommencer.

The "Play Tennis" database (11)

	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Class
	Outlook	Temperature	Humidity	Windy	Play
Example 1	sunny	hot	high	FALSE	no
Example 2	sunny	hot	high	TRUE	no
Example 3	overcast	hot	high	FALSE	yes
Example 4	rainy	mild	high	FALSE	yes
Example 5	rainy	cool	normal	FALSE	yes
Example 6	rainy	cool	normal	TRUE	no
Example 7	overcast	cool	normal	TRUE	yes
Example 8	sunny	mild	high	FALSE	no
Example 9	sunny	cool	normal	FALSE	yes
Example 10	rainy	mild	normal	FALSE	yes
Example 11	sunny	mild	normal	TRUE	yes
Example 12	overcast	mild	high	TRUE	yes
Example 13	overcast	hot	normal	FALSE	yes
Example 14	rainy	mild	high	TRUE	no

The "Play Tennis" database (12)

Une approche **bottom-up** à l'induction de règles de décision

Recommencer avec un exemple (le premier de ceux qui restent)

numéro 6: (Rainy, Cool, Normal, True, No)

Regarder ce qui peut être généralisé sans perdre la classe.

(* , Cool, Normal, True, No)? Contre-exemple le numéro 7: (Overcast, Cool, Normal, True, Yes)

(Rainy, * , Normal, True, No)? Aucun exemple dans ce cas, continuer

(Rainy, * , * , True, No)? Ceci ajoute le numéro 14: (Rainy, Mild, High, True, No)

(Rainy, * , * , * , No)? Aucun exemple dans ce cas, la règle est acceptée

Générer la règle:

IF (Outlook = rainy) THEN (Play = no)

Enlever les exemples couverts par la règle (now numbers 6 and 14) de la base de données.

Recommencer.

The "Play Tennis" database (13)

	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Class
	Outlook	Temperature	Humidity	Windy	Play
Example 1	sunny	hot	high	FALSE	no
Example 2	sunny	hot	high	TRUE	no
Example 3	overcast	hot	high	FALSE	yes
Example 4	rainy	mild	high	FALSE	yes
Example 5	rainy	cool	normal	FALSE	yes
Example 6	rainy	cool	normal	TRUE	no
Example 7	overcast	cool	normal	TRUE	yes
Example 8	sunny	mild	high	FALSE	no
Example 9	sunny	cool	normal	FALSE	yes
Example 10	rainy	mild	normal	FALSE	yes
Example 11	sunny	mild	normal	TRUE	yes
Example 12	overcast	mild	high	TRUE	yes
Example 13	overcast	hot	normal	FALSE	yes
Example 14	rainy	mild	high	TRUE	no

The "Play Tennis" database (14)

Une approche **bottom-up** à l'induction de règles de décision

Remarquer que tous les exemples restants appartiennent à la même classe: stop.

Générer la règle:

(Play = yes)

Etablir une liste **DANS L'ORDRE** et insérer les clauses **ELSE**

IF ((Outlook = sunny) AND (Humidity = high)) THEN (Play = no)

ELSE IF (Windy = false) THEN (Play = yes)

ELSE IF (Outlook = rainy) THEN (Play = no)

ELSE (Play = yes)

Fini.

Induction “bottom-up” de règles de décision

Avec n attributs il y a 2^n généralisations possibles

Avec 3 attributs ceux-ci sont: $\langle \text{att1val}, \text{att2val}, \text{att3val} \rangle$ $\langle *, \text{att2val}, \text{att3val} \rangle$ $\langle \text{att1val}, *, \text{att3val} \rangle$ $\langle \text{att1val}, \text{att2val}, * \rangle$ $\langle *, *, \text{att3val} \rangle$ $\langle *, \text{att2val}, * \rangle$ $\langle \text{att1val}, *, * \rangle$ $\langle *, *, * \rangle$

Coverage = nombre d'exemples couverts par la règle / nombre total d'exemples

Algorithme

Jusqu'à ce que tous les exemples appartiennent à la même classe {

Prendre un exemple pas encore couvert

Tester toutes les généralisations possibles

Trouver la généralisation qui a le meilleur coverage, sans recouvrir d'exemples d'une autre classe

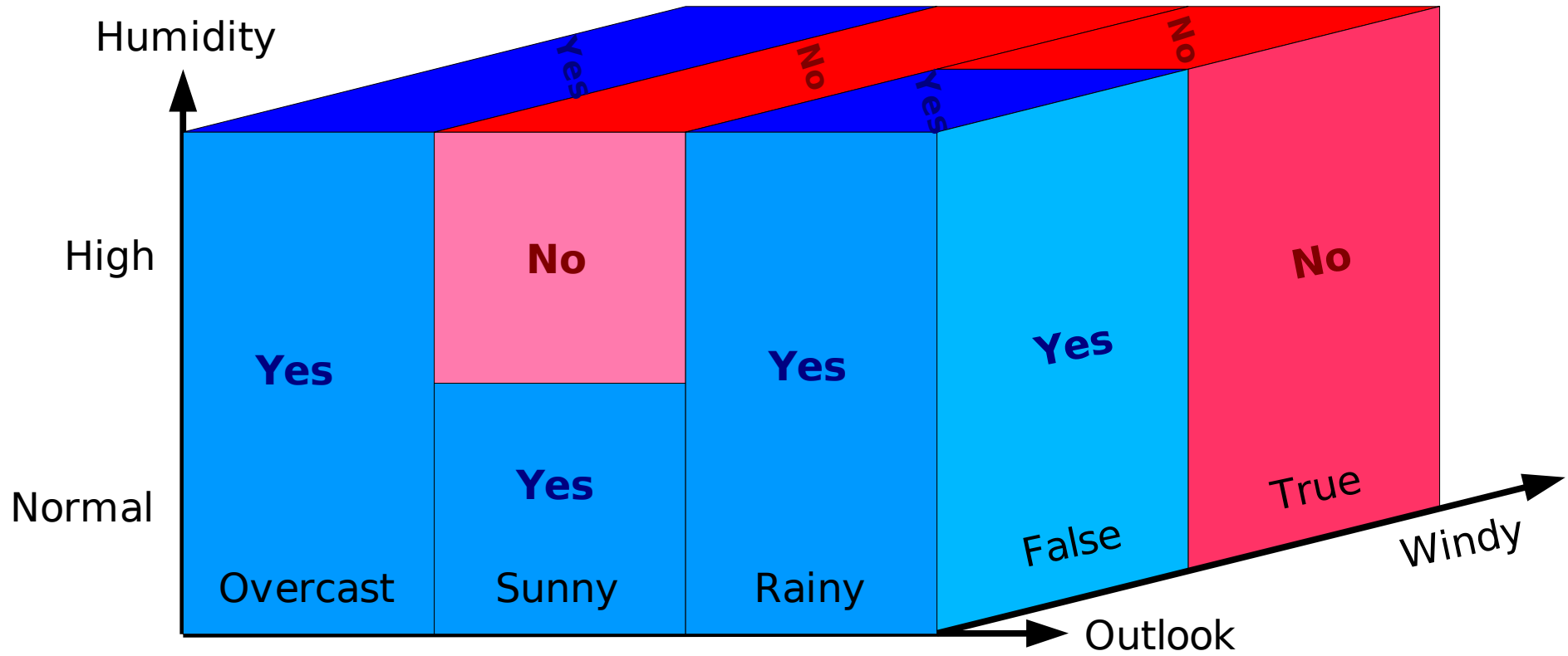
Générer la règle correspondant à cette généralisation et l'ajouter à la fin de la liste

Enlever les exemples couverts par cette règle

}

Quand tous les exemples qui restent appartiennent à la même classe, générer la dernière règle, la règle par défaut.

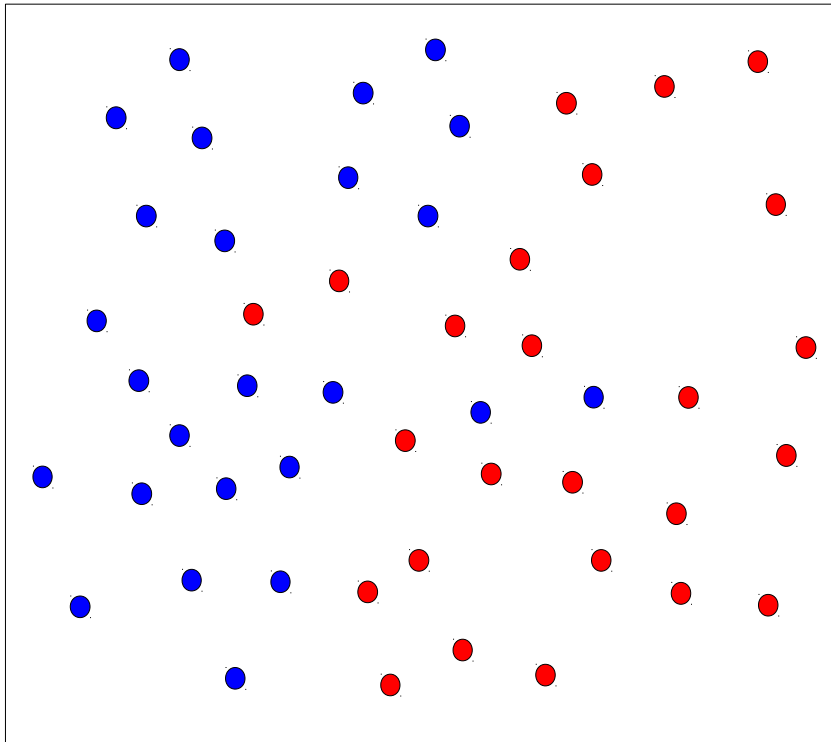
The "Play Tennis" database (15)



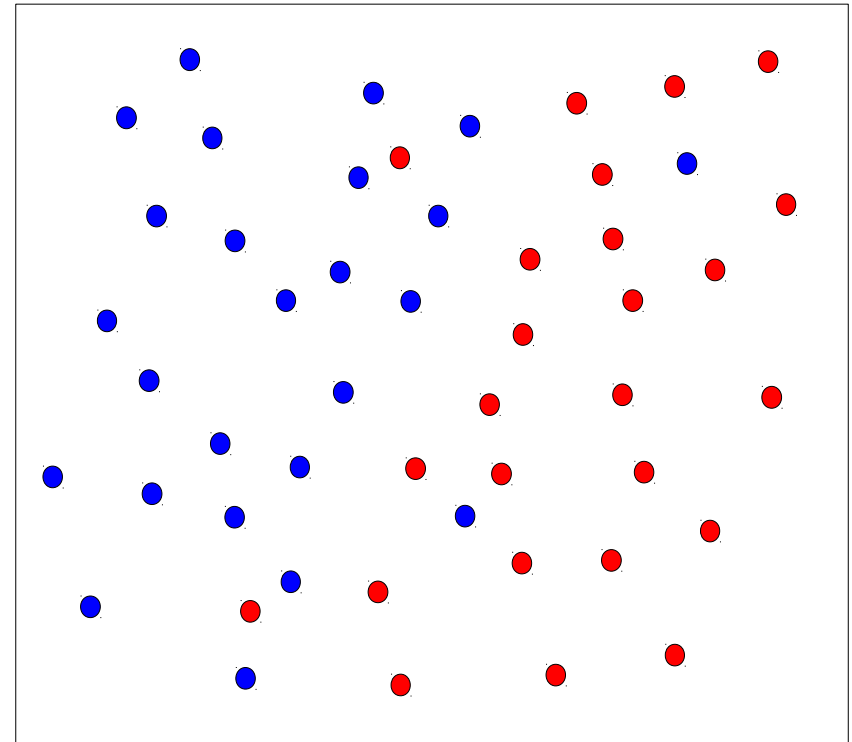
Les arbres de décision et les règles d'ordre zéro découpent l'espace de recherche perpendiculairement aux axes.

Overfitting (1)

Overfitting se traduit par “surapprentissage”

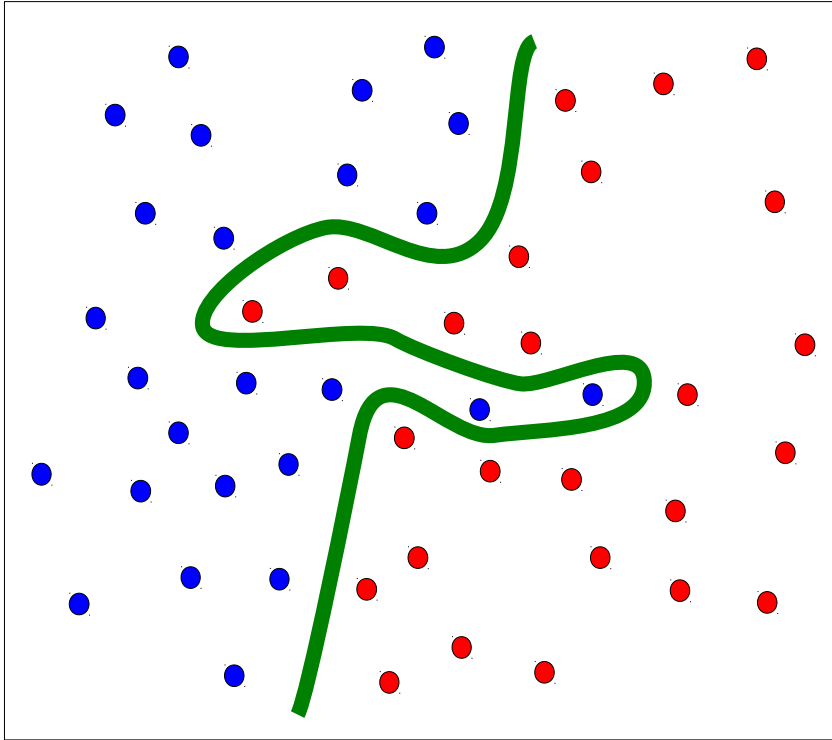


Training set

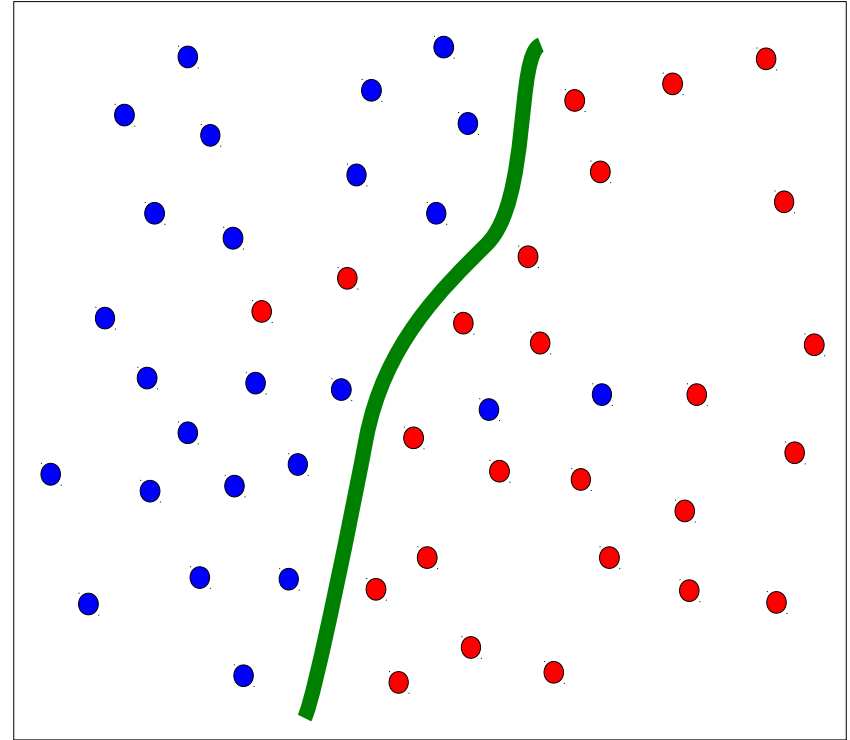


Test set

Overfitting (2)

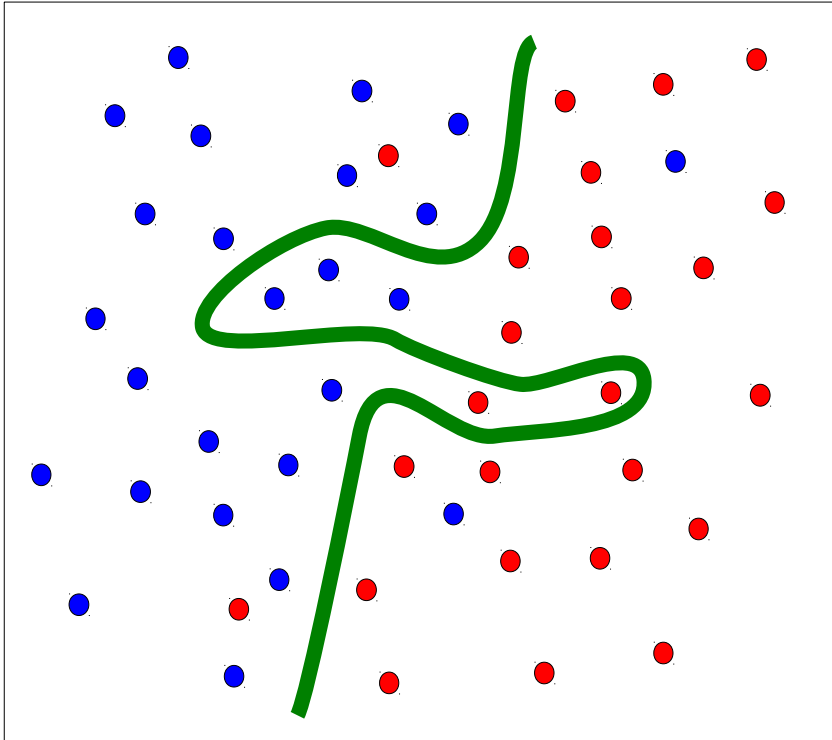


Accuracy over the training set
100 %

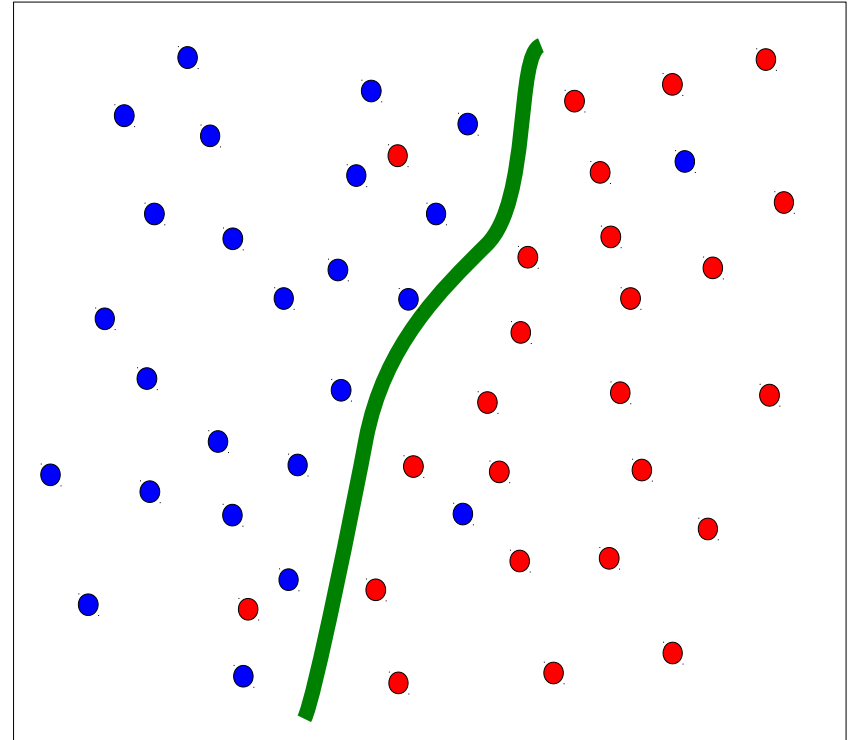


Accuracy over the training set
92 %

Overfitting (3)



Accuracy over the test set
82 %

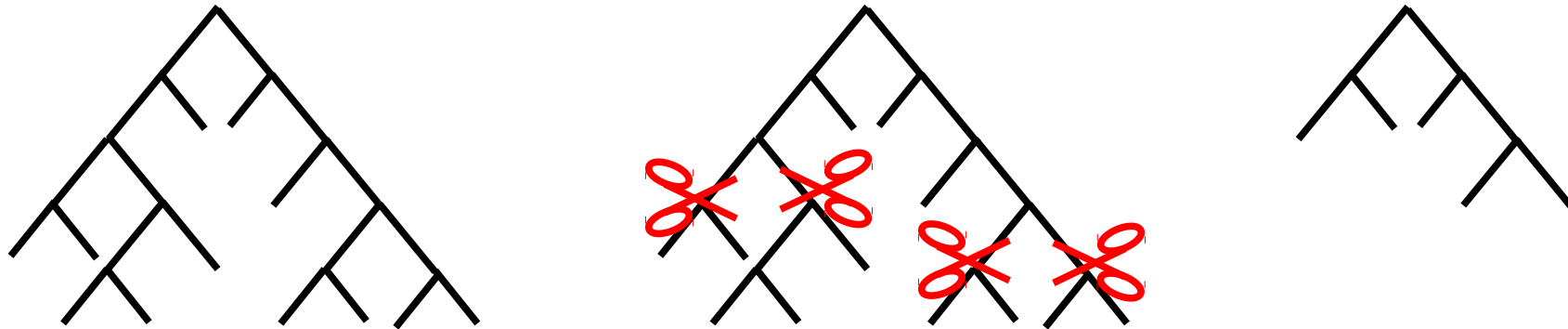


Accuracy over the test set
92 %

Overfitting (4)

Solution pour les arbres de décision: **pruning** (élaguage)

Par exemple, on peut décider que si une feuille contient moins de n exemples elle doit être enlevée. Les exemples qu'elle contenait sont ajoutés au noeud immédiatement supérieur qui est étiqueté en fonction de la classe localement majoritaire.

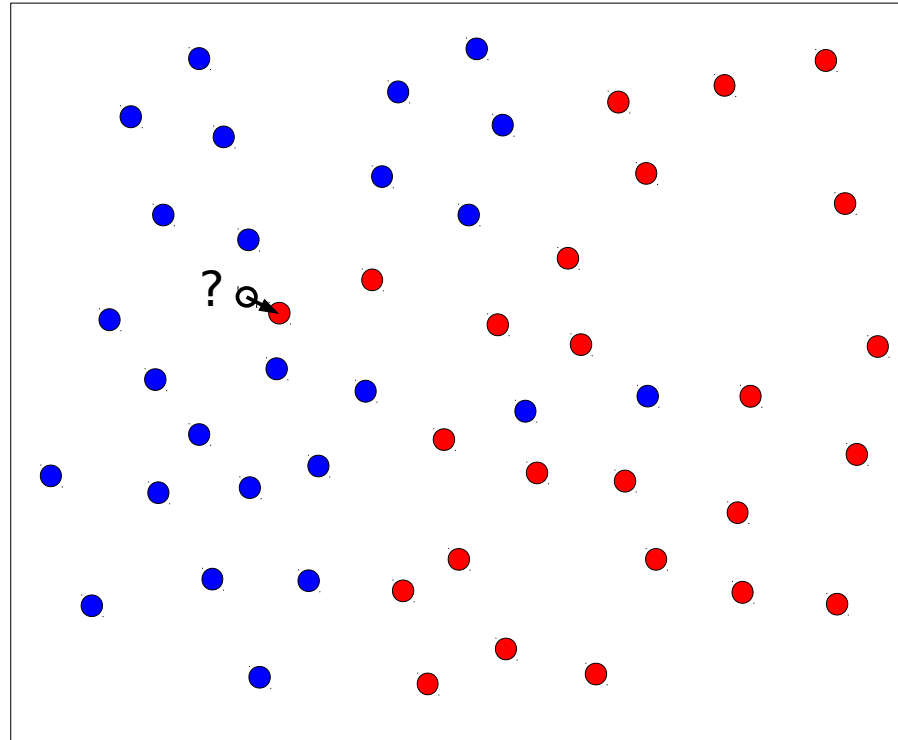


Solution pour les règles de décision: la **généralisation forcée**

Par exemple, si une règle testée couvre n exemples de la classe majoritaire et m exemples d'une autre classe, la règle peut être induite si $m/n < \text{ratio d'erreur toléré}$.

Il existe d'autres critères mathématiques pour régler l'élaguage. La sélection du critère qui donne les meilleurs résultats pour une base de donnée se fait souvent par trial-and-error (essais).

k Nearest Neighbours: kNN (1)

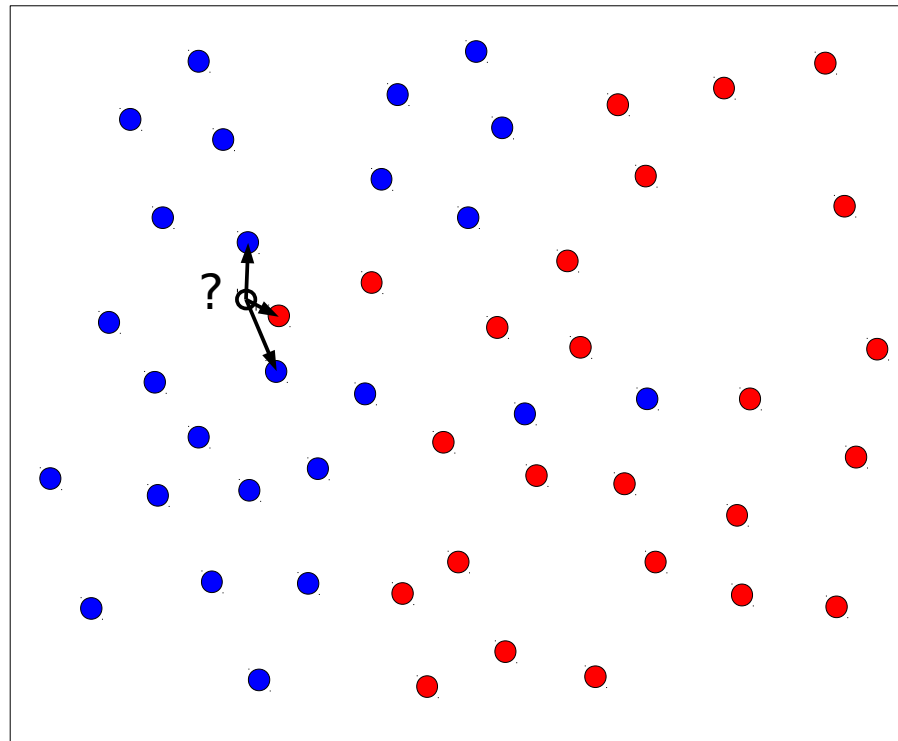


kNN se traduit par “les k plus proches voisins”.

Idée: les exemples qui se ressemblent ont tendance à appartenir à la même classe.

Règle de la majorité: la valeur choisie pour k est généralement impaire lorsque le nombre de valeurs possible de la classe est 2.

k Nearest Neighbours: kNN (2)

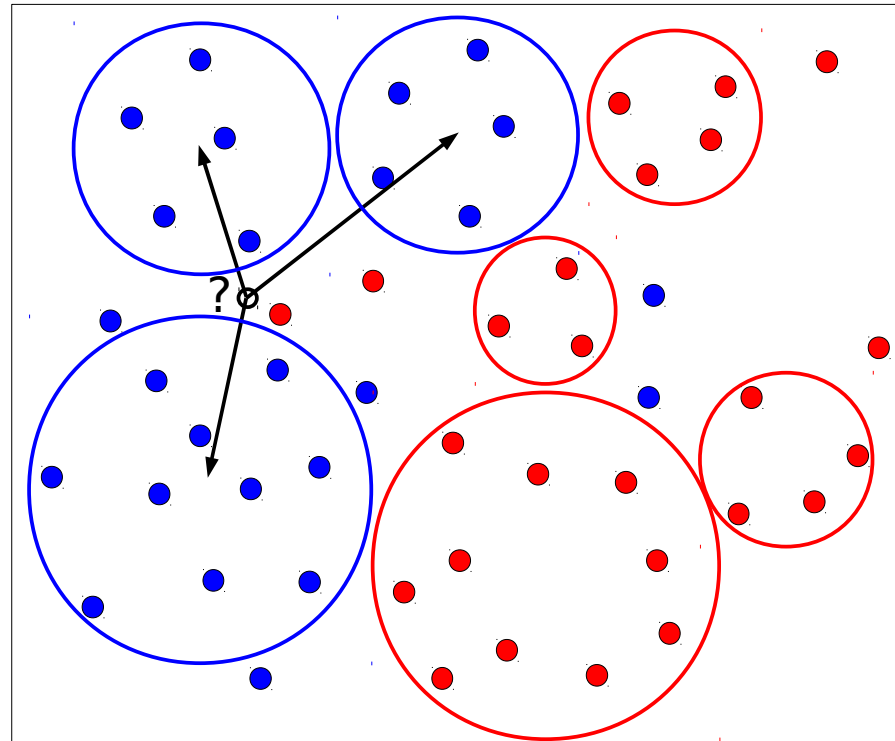


k ne doit pas être trop grand, un grand k tend à produire des erreurs sauf si les classes sont bien séparées.

Définir une mesure de distance adéquate entre les exemples peut être difficile.

Le classifieur ne produit pas un modèle intelligible par les humains. Il classifie mais n'explique pas.

k Nearest Neighbours: kNN (3) and clustering



Le classifieur utilise tous les exemples pour classifier chaque nouvel exemple. Le temps de calcul dépend donc du nombre d'exemples.

Si on a une mesure de distance appropriée, on peut trouver des zones denses d'exemples appartenant à la même classe. Le centre de ces zones est alors utilisé pour classifier les nouveaux exemples, ce qui réduit le temps de calcul.

Naive Bayes

Un algorithme de classification basé sur le théorème de Bayes' theorem qui assume “naïvement” que les exemples sont indépendants.

Bayes' theorem: $P(\text{Class} \mid \text{Attributes}) = P(\text{Class}) * P(\text{Attributes} \mid \text{Class}) / P(\text{Attributes})$

$P(\text{Attributes})$ est constant pour un ensemble d'exemples donné et peut donc être ignoré.

Indépendance supposée: $P(\text{Class} \mid \text{Attributes}) = P(\text{Class}) * P(\text{Att1} \mid \text{Class}) * \dots * P(\text{Attn} \mid \text{Class})$

The “Play Tennis” database (16)

$$\begin{aligned} P((\text{Class} = \text{no}) \mid (\text{O} = \text{Rainy}, \text{T} = \text{hot}, \text{H} = \text{high}, \text{W} = \text{TRUE})) \\ = (5 / 14) * (2 / 5) * (2 / 5) * (4 / 5) * (3 / 5) = 240 / 8750 = 0.0274 \end{aligned}$$

$$\begin{aligned} P((\text{Class} = \text{yes}) \mid (\text{O} = \text{Rainy}, \text{T} = \text{hot}, \text{H} = \text{high}, \text{W} = \text{TRUE})) \\ = (9 / 14) * (3 / 9) * (2 / 9) * (3 / 9) * (3 / 9) = 486 / 91854 = 0.0053 \end{aligned}$$

La probabilité que cet exemple appartienne à la classe “no” est plus élevée, c'est cette valeur de classe qui est attribuée à l'exemple. Comme le précédant, cet algorithme classifie mais n'explique pas.