IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Machine Learning Approaches : Decision Trees

1. *Decision Trees Concept*

2. *Decision Trees Induction Algorithms*

2.a. *ID3 Algorithm*

2.b. *Classification And Regression Tree, CART, Algorithm*

3. *Overfitting problem*

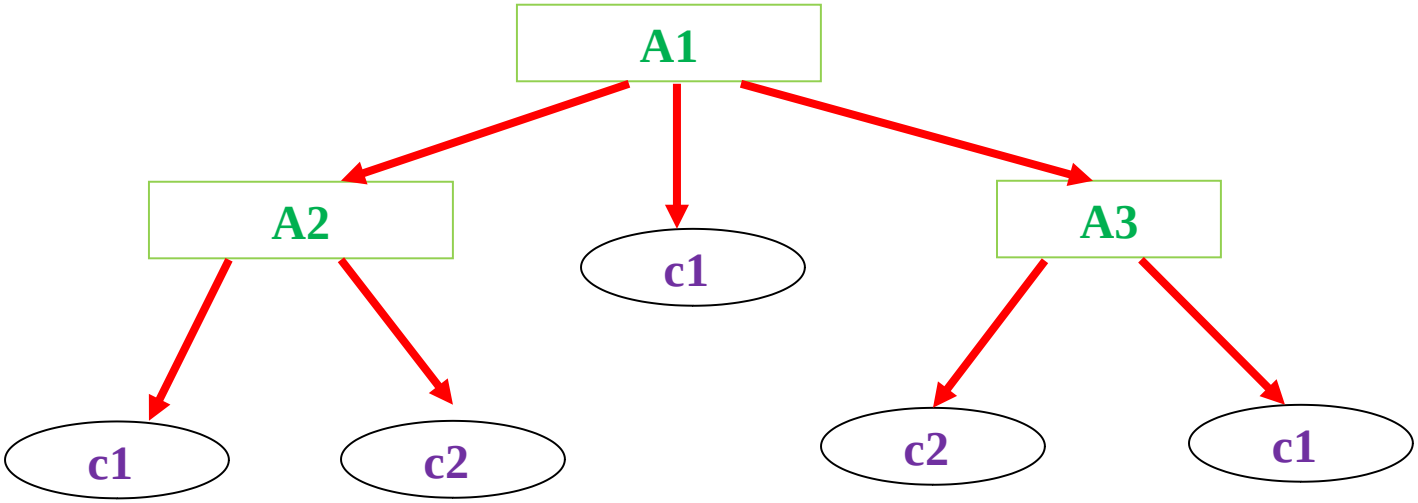
4. *Regression trees for prediction*

1. *Decision Trees Concept*

1. *Decision Trees Concept*

DECISION TREE: Piecewise constant tree-like algorithm obtained with recursive partitioning of the input data space (i.e., $\Re^K$) into axis-parallel regions, labeled with class value, where:

Each *interior node* tests an attribute
Each *branch* corresponds to an attribute value
Each *leaf node* is labelled with a class

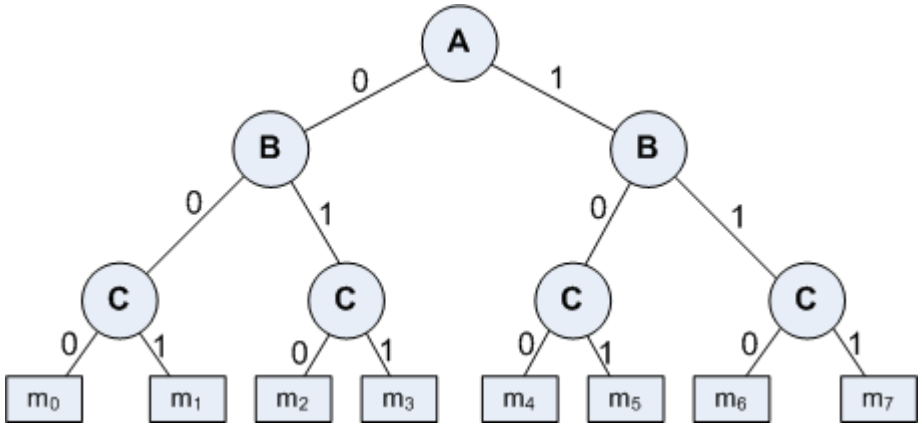


1. *Decision Trees Concept*

***A Decision Tree** is an important data structure known to solve many computational problems*

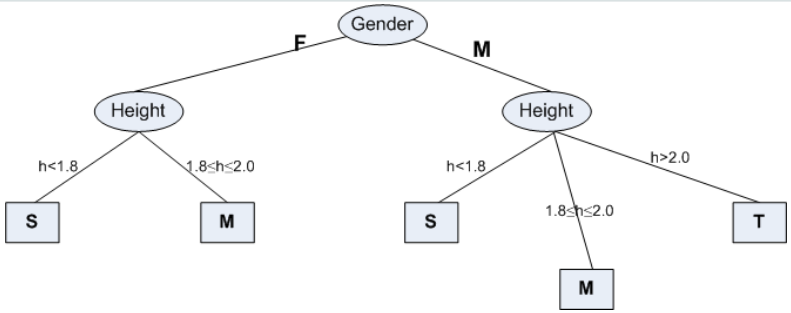
Example : Binary Decision Tree

A	B	C	f
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7



1. *Decision Trees Concept*

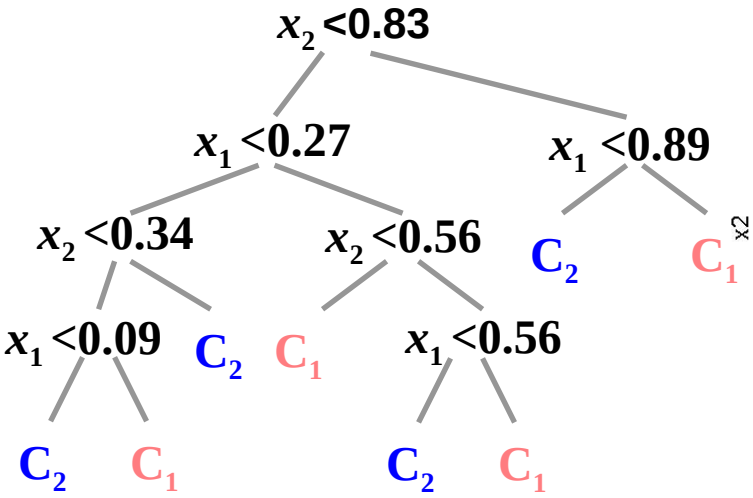
Some Characteristics

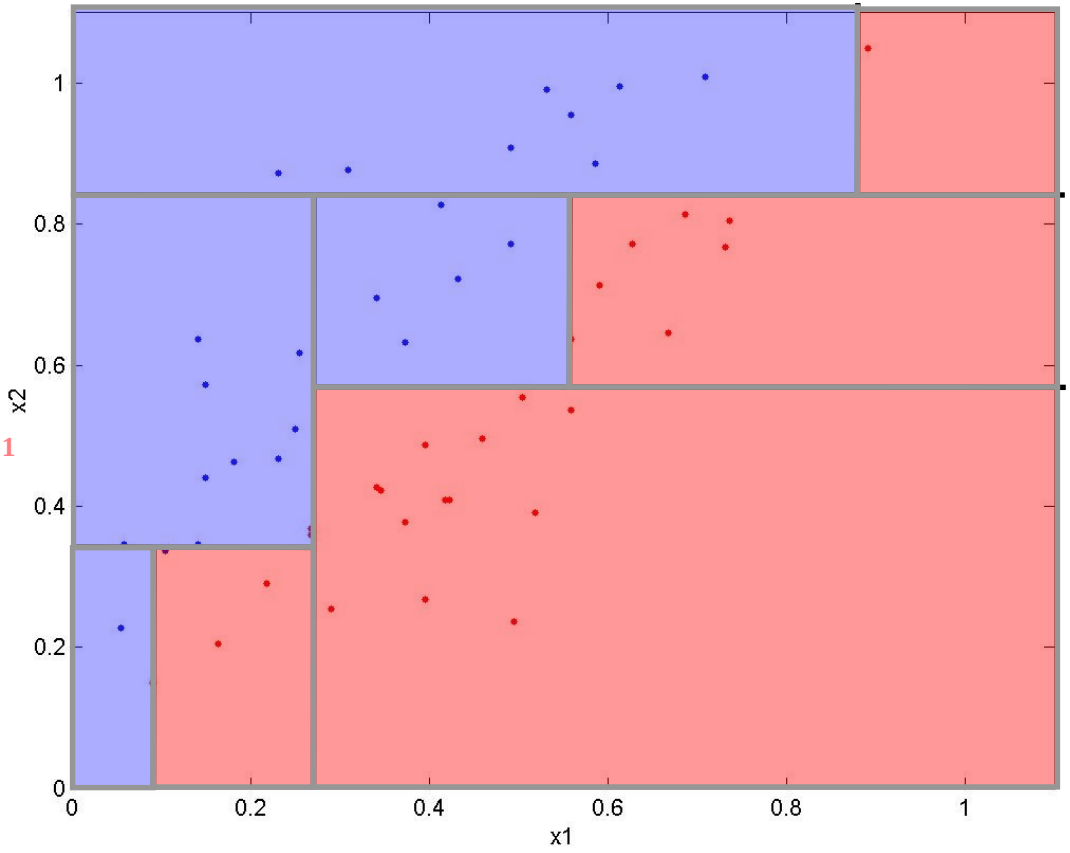


- **Decision tree may be *n*-ary, $n \geq 2$**
- **Decision tree is not unique, as different ordering of internal nodes can give different decision tree**
- **Decision tree helps us to classify data**
  - **Internal nodes are some attribute**
  - **Edges are the values of attributes**
  - **External nodes are the outcome of classification**

1. *Decision Trees Concept*

- Splits are parallel to the axes
- At every step of the "binary" partitioning, data in the current node is split "at best" (i.e., to have the greatest decrease, in term of heterogeneity in the two child nodes





IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2. *Decision Trees Induction Algorithms*

2.a. *ID3 Algorithm*

2.b. *Classification And Regression Tree, CART, Algorithm*

Decision tree induction **(i.e., construction) :**

- A top-down, recursive and divide-and-conquer approach

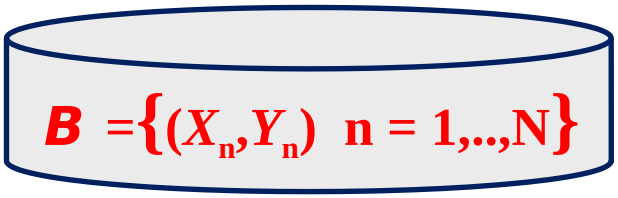
- The procedure is to choose an attribute and split it into from a larger training set into smaller training sets.

- Different algorithms have been proposed to take a good control over

1. **Choosing the best attribute to be splitted, and**
2. **Splitting criteria**

Learning decision trees from training data

Goal : Build a decision tree to classify examples of a concept using supervised learning from a training set



Properties we want the decision tree to have ?

1. *It should be consistent with the learning training set*
   - Trivial algorithm: construct a decision tree that has one path to a leaf for each example
   - Problem: it does not capture useful information from the database

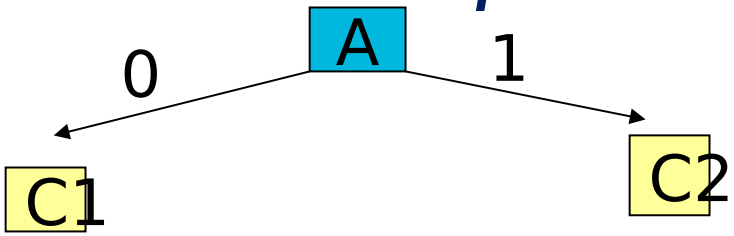
2. *It should be as simple as possible*
   - Generate all trees and pick the simplest one that is consistent with the learning sample.
   - Problem: intractable, there are too many trees

Learning decision trees from training data

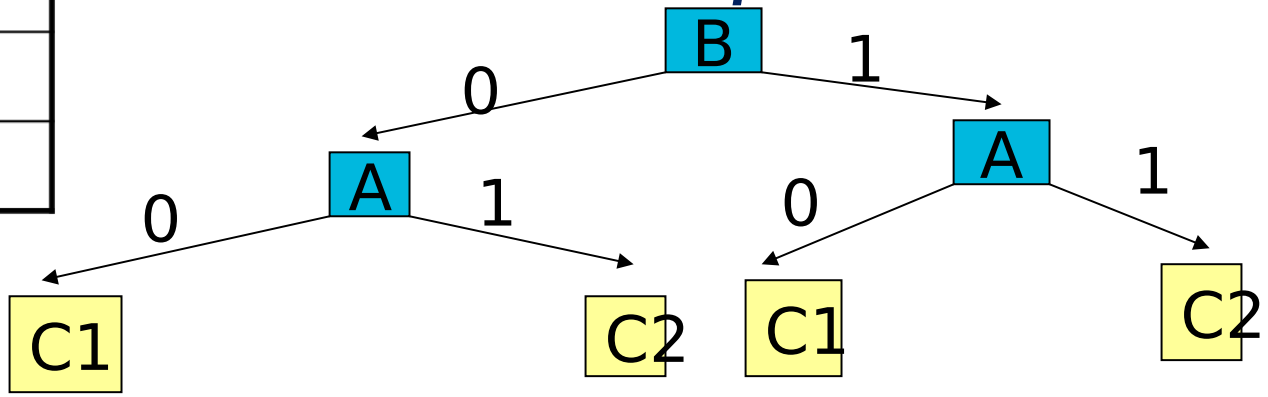
Example

A	B	Class
0	1	C1
0	0	C1
1	1	C2
1	0	C2

Attribute A is chosen first:



Attribute B is chosen first:

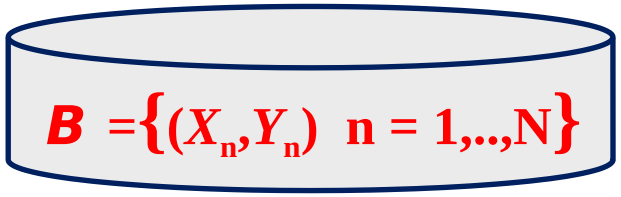


2. Decision Trees Induction Algorithms

A Tree learning algorithm allows to choose the tree structure and to determine the predictions at leaf nodes

Predictions: to minimize the misclassification error, associate the majority class among the learning sample cases reaching this node

Top-Down induction learning of decision trees



- *Choose « best » attribute*
- *Split the learning sample*

IF: all sample data from the training set have the same class

Create a leaf with that class

ELSE:

Proceed recursively until each sample data is correctly classified

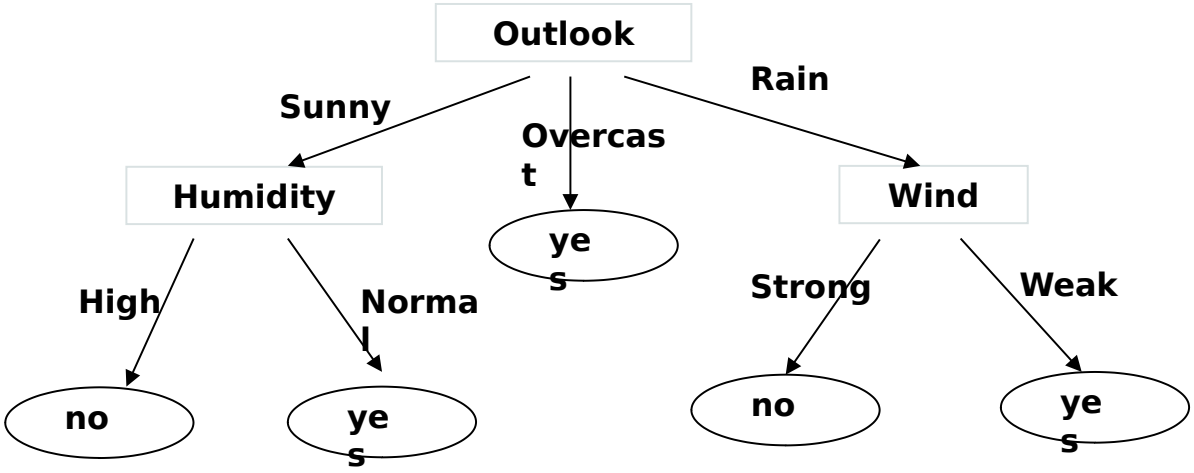
Which attribute is best ?

We should maximize the class separation at each step, i.e., make successors as pure as possible

⇒

This will favour short paths in the trees

Tree decision model : Use of decision rules



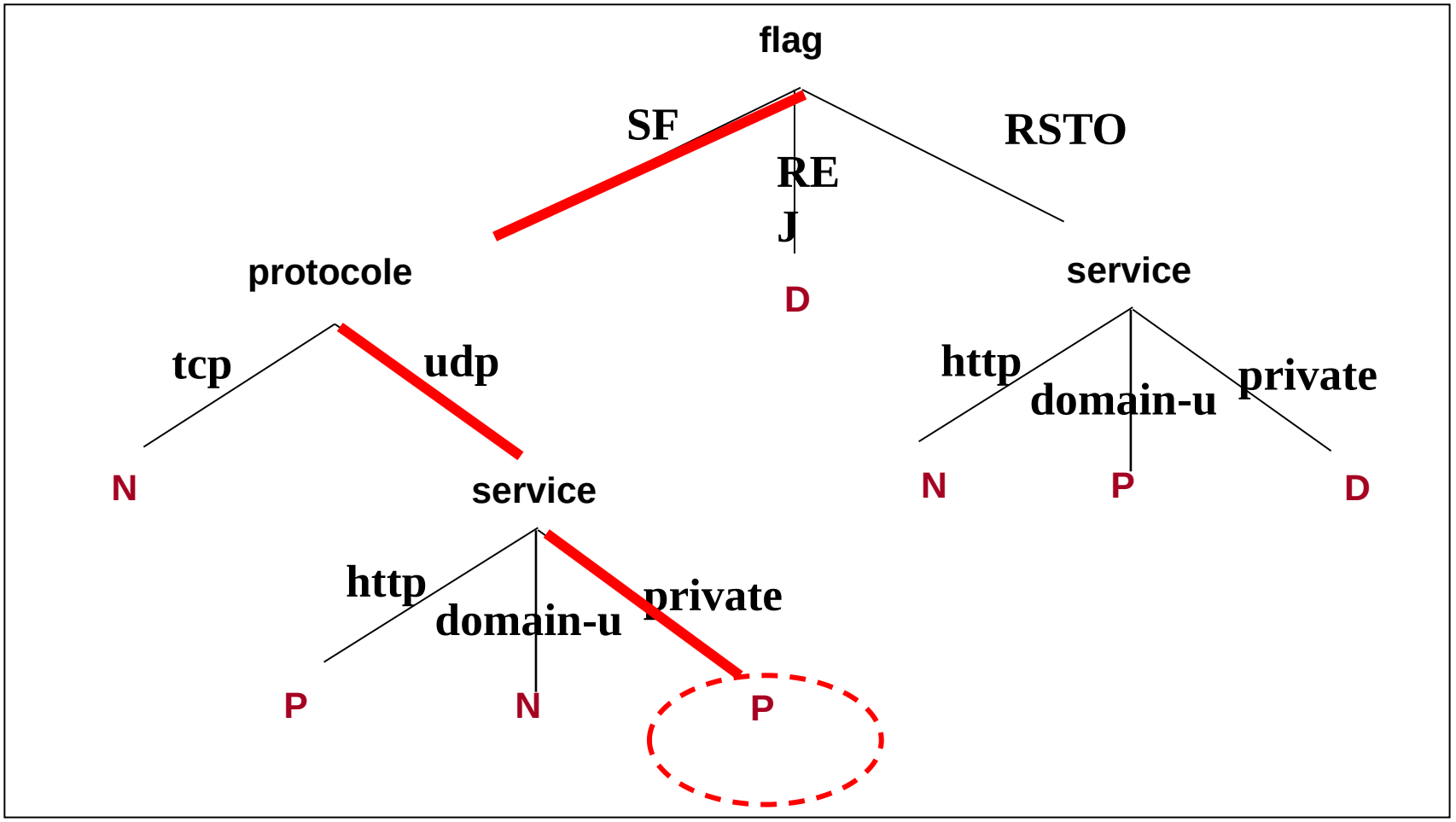
A rule set is finally derived from a decision tree where a rule is generated for each path in the decision tree from the root to a leaf. The left-hand side of a rule is easily built from the label of the nodes and the labels of the arcs

2. *Decision Trees Induction Algorithms*

Protocole	Service	Flag	Classe
tcp	http	SF	Normal
tcp	http	RSTO	Normal
tcp	http	REJ	Probing
tcp	time	SF	Probing
tcp	time	SO	DOS
tcp	auth	SF	Normal
tcp	auth	SO	DOS
tcp	private	SF	Normal
tcp	private	SF	Normal
tcp	private	REJ	Probing
tcp	private	RSTO	DOS
tcp	private	SO	DOS
udp	domain_u	SF	Normal
udp	private	SF	DOS
tcp	http	RSTO	Normal
tcp	private	RSTO	DOS
tcp	http	SF	Normal

Connection Nature

2. *Decision Trees Induction Algorithms*



New connexion

udp	Private	SF	C=?

2. *Decision Trees Induction Algorithms*

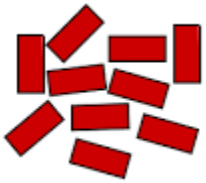
2.a. *ID3 Algorithm*

2.b. *Classification And Regression Tree, CART, Algorithm*

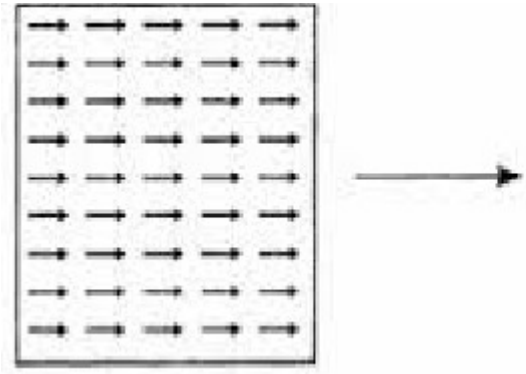
2.a. *ID3 Algorithm*

ID3 algorithm is based on the use of entropy in order to measure how informative a node is

More **ordered**
less **entropy**



Less ordered
higher entropy



More organized or
ordered (less **probable**)

Less organized or
disordered (**more** probable)

In a classification job, entropy is an important concept, which is considered as an information-theoretic measure of the "uncertainty" contained in a training data (due to the presence of more than one classes)

2.a. *ID3 Algorithm*

ID3 algorithm is based on the use of Shannon's entropy in order to measure how informative a node is

Splitting (on any attribute) has the property that average entropy of the resulting training subsets is less than or equal to that of the previous training set

ID3 algorithm defines a measurement of a splitting called *Information Gain* to determine the goodness of a split:

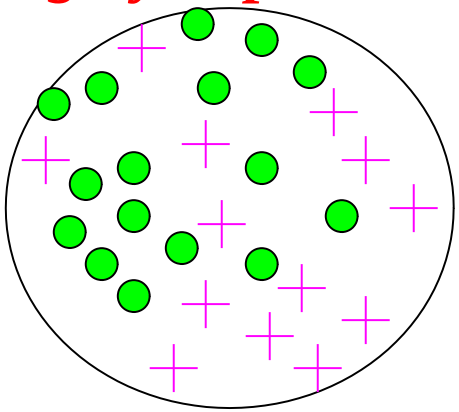
The attribute with the largest value of information gain is chosen as the splitting attribute and

it partitions into a number of smaller training sets based on the distinct values of attribute under split.

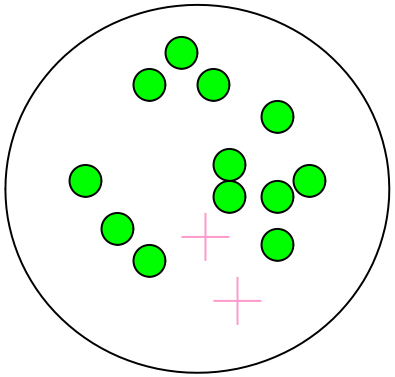
2.a. *ID3 Algorithm*

Measure of impurity !

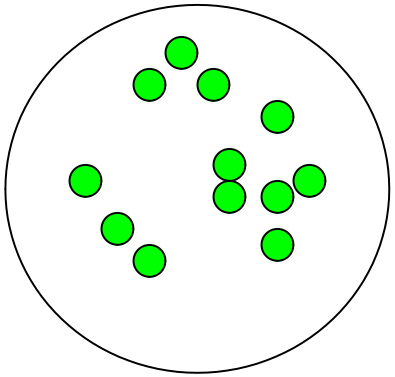
Highly impure set



Impure set

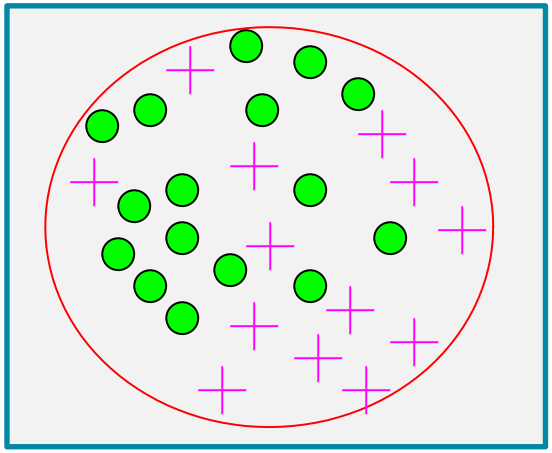


Minimal impure set

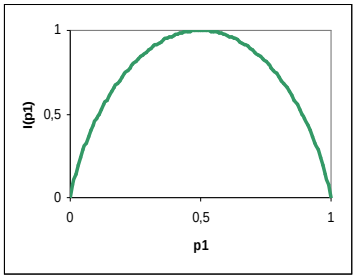


2.a. *ID3 Algorithm*

$$Entropy = \sum_i - p_i \log_2 p_i$$

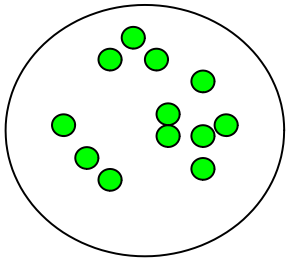


p_i (Class « i » probability) : Proportion of samples labeled from class « i » in the considered set



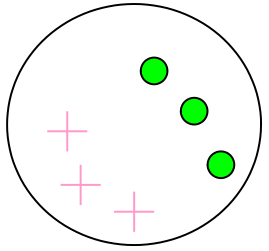
Two classes :

Highest purity



Entropy = $- 1 \log_2 1 = 0$

Lowest purity



Entropy = $-0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$

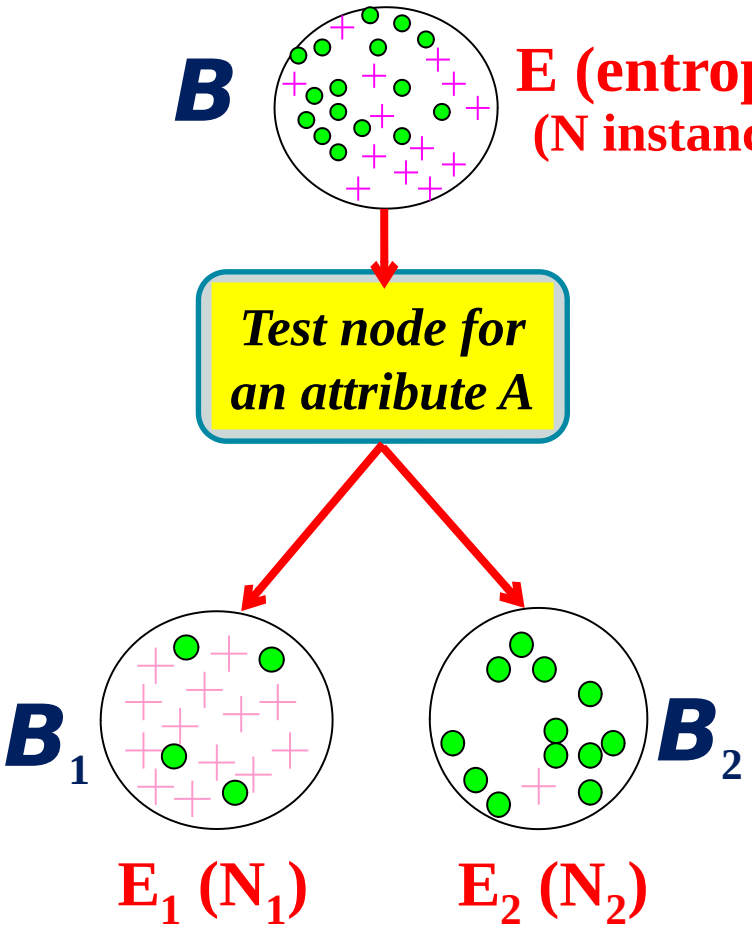
0 ⟶ **1**

2.a. *ID3 Algorithm*

- ***Entropy takes it's minimum value (zero) if and only if all the instances have the same class (i.e., the training set with only one non-empty class, for which the probability 1)***

- ***Entropy takes its maximum value when the instances are equally distributed among K possible classes. In this case, the maximum value of the entropy is***

Information Gain splitting criterion



Information Gain
(due to attribute *A*)
=
$$E - \{(N_1/N)\, E_1 + (N_2/N)\, E_2\}$$

This represents the difference between :

Information needed to identify an element of *B*

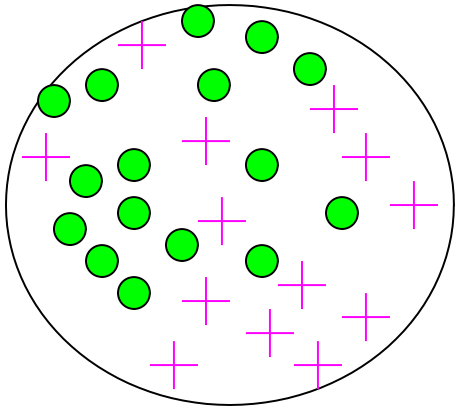
and

Information needed to identify an element of *B* after the value of attribute *A* has been obtained

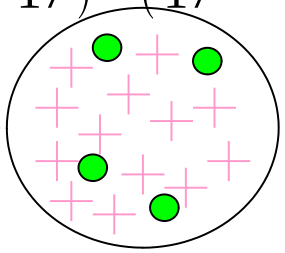
2.a. *ID3 Algorithm*

Example

N= 30 instances)

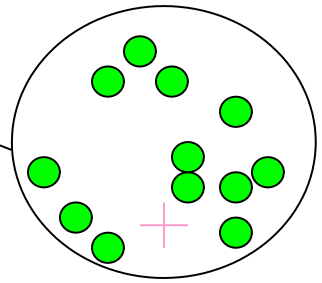


$$E = -\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right) - \left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right) = 0.996$$



$$-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right) - \left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right) = 0.787$$

$N_1 = 17$



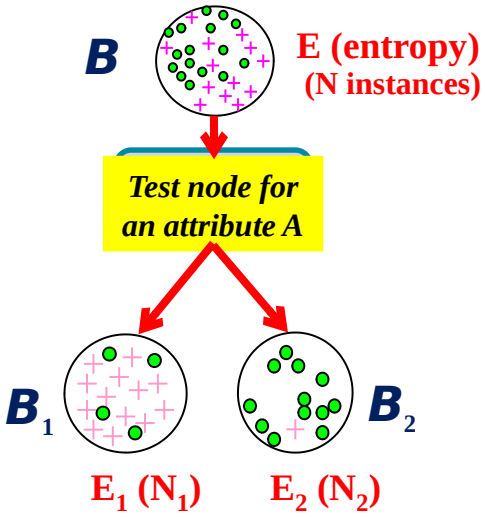
$$-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right) - \left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right) = 0.391$$

$N_3 = 13$

Mean weighted entropy $= \left(\frac{17}{30}\cdot 0.787\right) + \left(\frac{13}{30}\cdot 0.391\right) = 0.615$

Information Gain = 0.996 - 0.615 = 0.38

Information Gain splitting criterion



Information Gain is used to rank attributes and to build decision trees where at each node is located the attribute with the greatest information gain among the attributes not yet considered in the path from the root

2.a. *ID3 Algorithm*

Example

Attributs	Pif	Temp	Humid	Vent
Valeurs possibles	soleil,couvert,pluie	chaud,bon,frais	normale,haute	vrai,faux

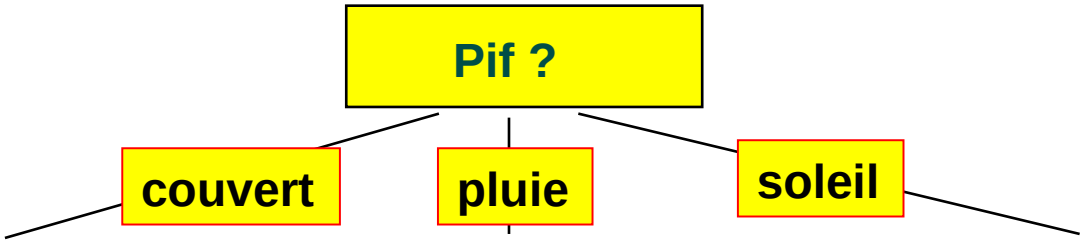
N°	Pif	Temp	Humid	Vent	Golf
1	soleil	chaud	haute	faux	NePasJouer
2	soleil	chaud	haute	vrai	NePasJouer
3	couvert	chaud	haute	faux	Jouer
4	pluie	bon	haute	faux	Jouer
5	pluie	frais	normale	faux	Jouer
6	pluie	frais	normale	vrai	NePasJouer
7	couvert	frais	normale	vrai	Jouer
8	soleil	bon	haute	faux	NePasJouer
9	soleil	frais	normale	faux	Jouer
10	pluie	bon	normale	faux	Jouer
11	soleil	bon	normale	vrai	Jouer
12	couvert	bon	haute	vrai	Jouer
13	couvert	chaud	normale	faux	Jouer
14	pluie	bon	haute	vrai	NePasJouer

Entropie de l'ensemble initial d'exemples

$$E = - 9/14 \log_2(9/14) - 5/14 \log_2(5/14)$$

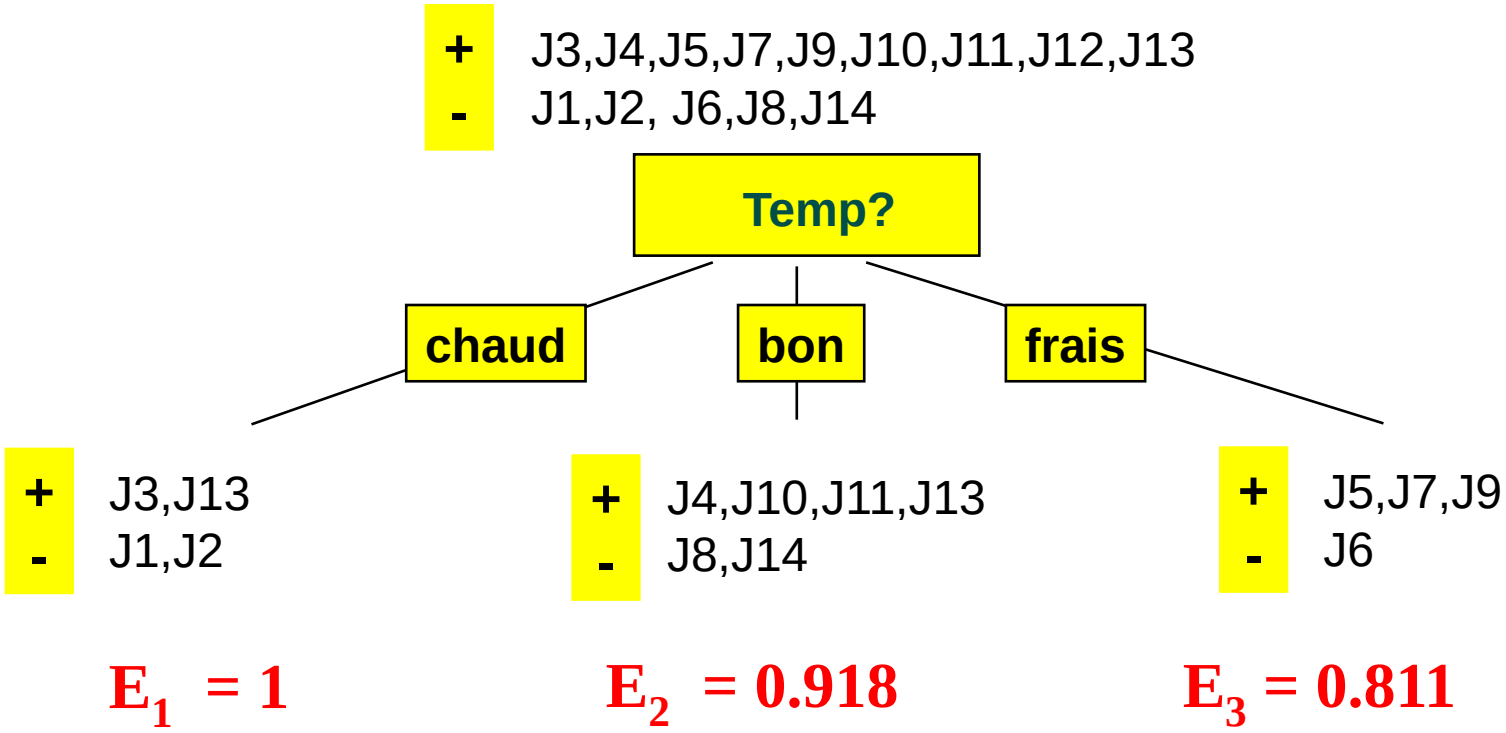
Entropie des sous-arbres associés au test sur Pif ?

\+ J3,J4,J5,J7,J9,J10,J11,J12,J13
\- J1,J2, J6,J8,J14



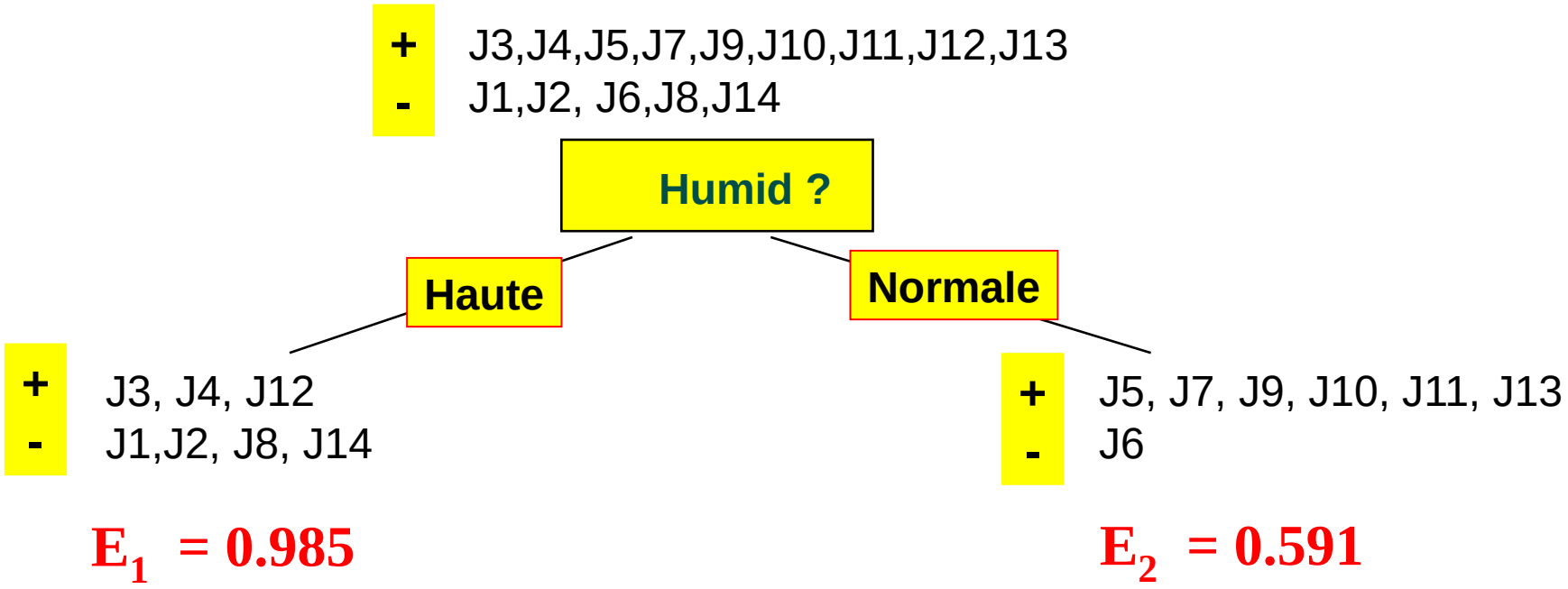
couvert	pluie	soleil
+ J3,J13,J7,J12	+ J4,J5,J10	+ J9,J11
-	- J6,J14	- J1,J8,J2
$E_1 = 0$	$E_2 = 0.971$	$E_3 = 0.971$

Entropie des sous-arbres associés au test sur TEMPS?



$E_1 = 1$ $E_2 = 0.918$ $E_3 = 0.811$

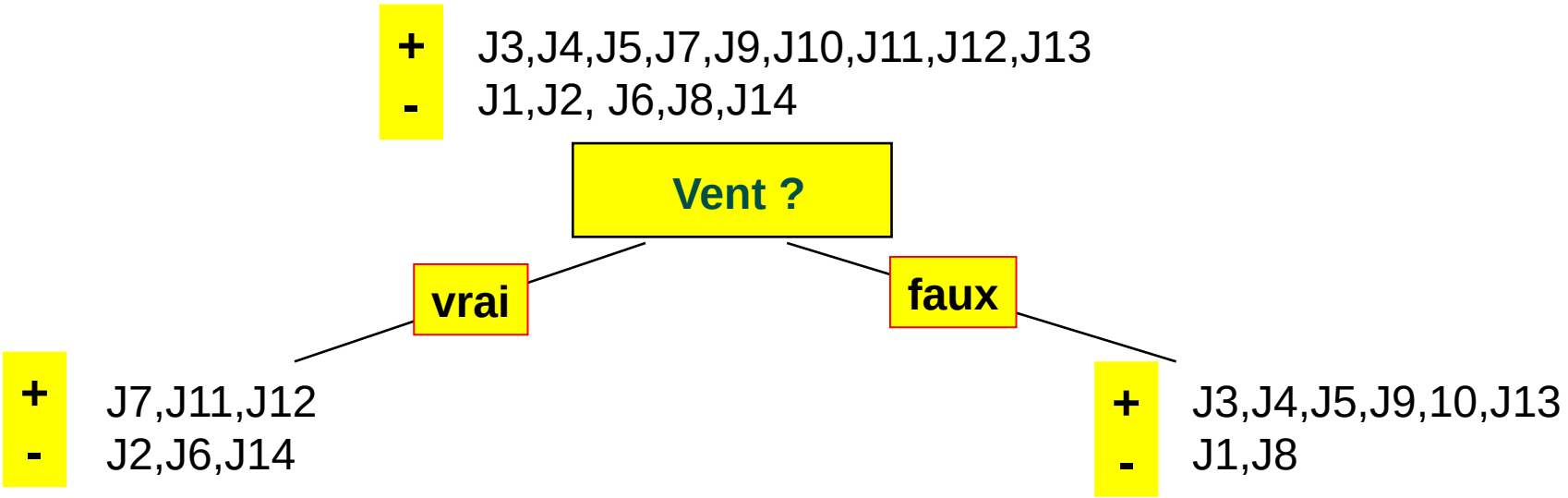
Entropie des sous-arbres associés au test sur Humid?



$E_1 = 0.985$

$E_2 = 0.591$

Entropie des sous-arbres associés au test sur Vent?



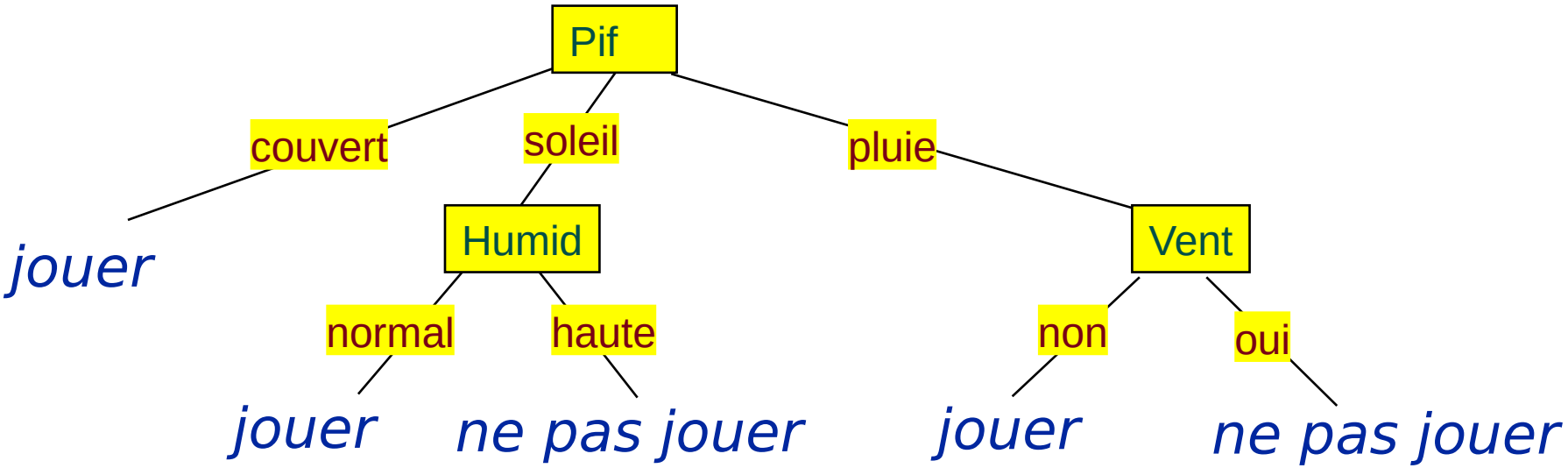
$E_1 = 1$

$E_2 = 0.811$

Gain(Temp) = 0.029 bits
Gain(Pif) = 0.246 bits
Gain(Humid) = 0.151 bits
Gain(Vent) = 0.048 bits

Choix de l'attribut Pif pour le premier test

Application récursive ID3: Arbre final obtenu :



2. *Decision Trees Induction Algorithms*

2.a. *ID3 Algorithm*

2.b. *Classification And Regression Tree, CART, Algorithm*

2.b. *Classification And Regression Trees, CART*

CART is a technique that generates a binary decision tree

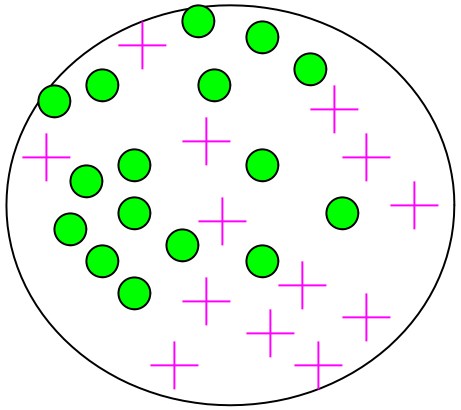
ID3 uses Information gain as a measure to select the best attribute to be splitted, whereas *CART does the same but using another measurement called Gini index*

GINI INDEX

If a data set T contains examples from n classes, gini index, gini(T) is defined as:

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class j in T



$n = 2,\ p_1 = 14 / 30, p_1 = 16 / 30$

Note : ***gini(T)*** **measures the "impurity" of data set *T*.**

The smallest value of gini(T) is zero which it takes when all the classifications are same (if the classes in T are skewed)

It takes its largest value , when the classes are evenly distributed between the tuples, that is the frequency of each class is

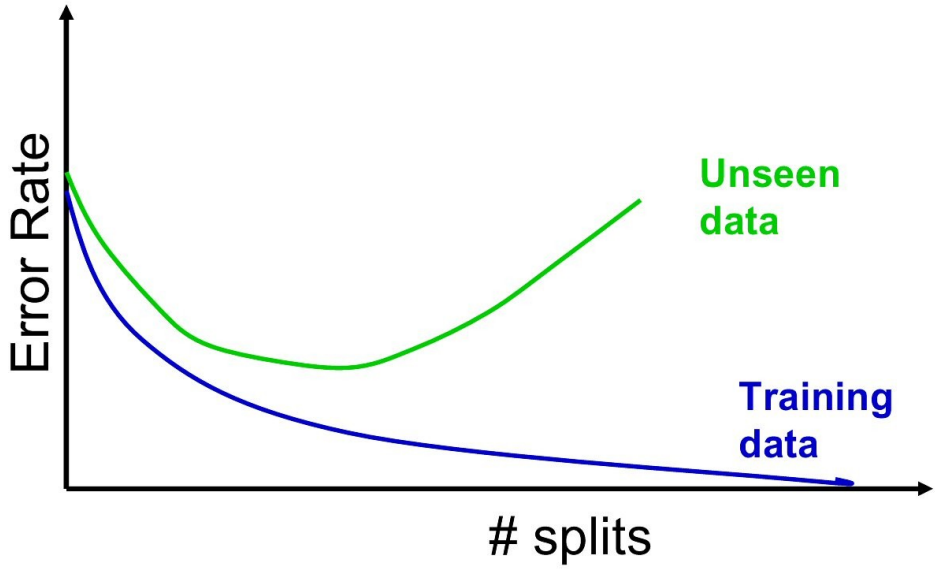
3. *Overfitting Problem*

STOPPING TREE GROWTH

Natural end of process is 100% purity in each leaf

Overfitting leads to low predictive accuracy of new data

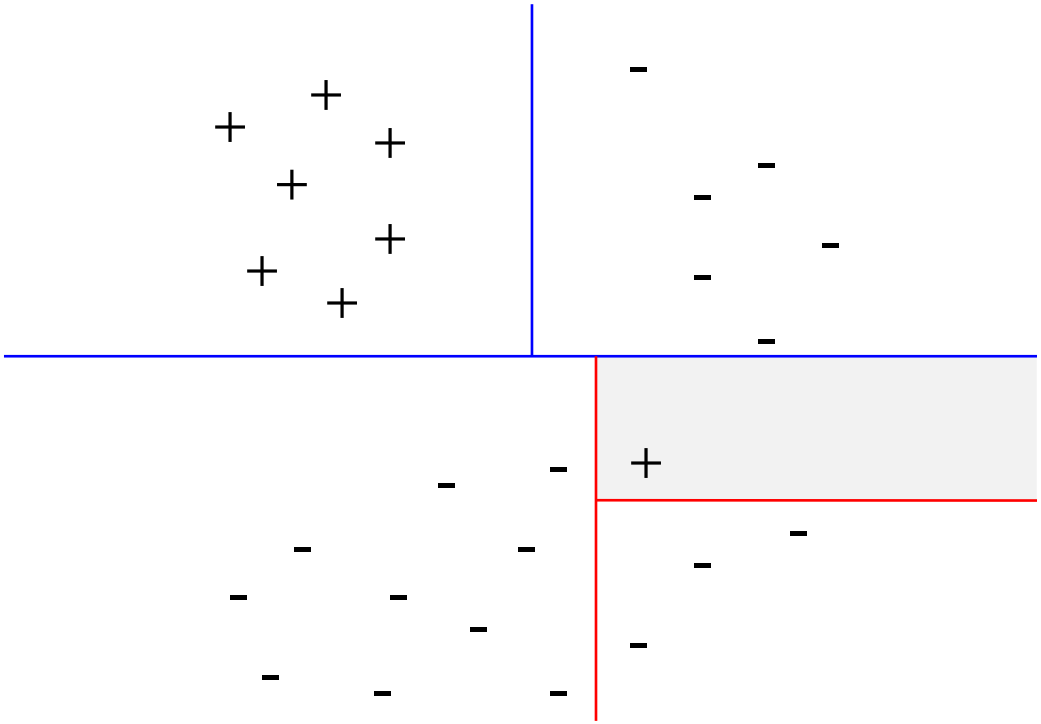
Past a certain point, the error rate for the validation data starts to increase



3. Overfitting Problem

REASONS OF OVERFITTING

Overfits the data will end up fitting noise in the data



Area with probably noise samples

We may not have enough data in some part of the learning sample to make a good decision

3. Overfitting Problem

HOW CAN WE AVOID OVERFITTING ?

Pre-pruning: *stop growing the tree earlier, before it reaches the point where it perfectly classifies the learning sample.*

Stop splitting a node if :

- *The number of objects is too small*
- *The impurity is low enough*
- *The best test is not statistically significant (according to some statistical test)*

Post-pruning: *allow the tree to overfit and then post-prune the tree*

Compute a sequence of trees $\{T_1, T_2, \ldots\}$ where

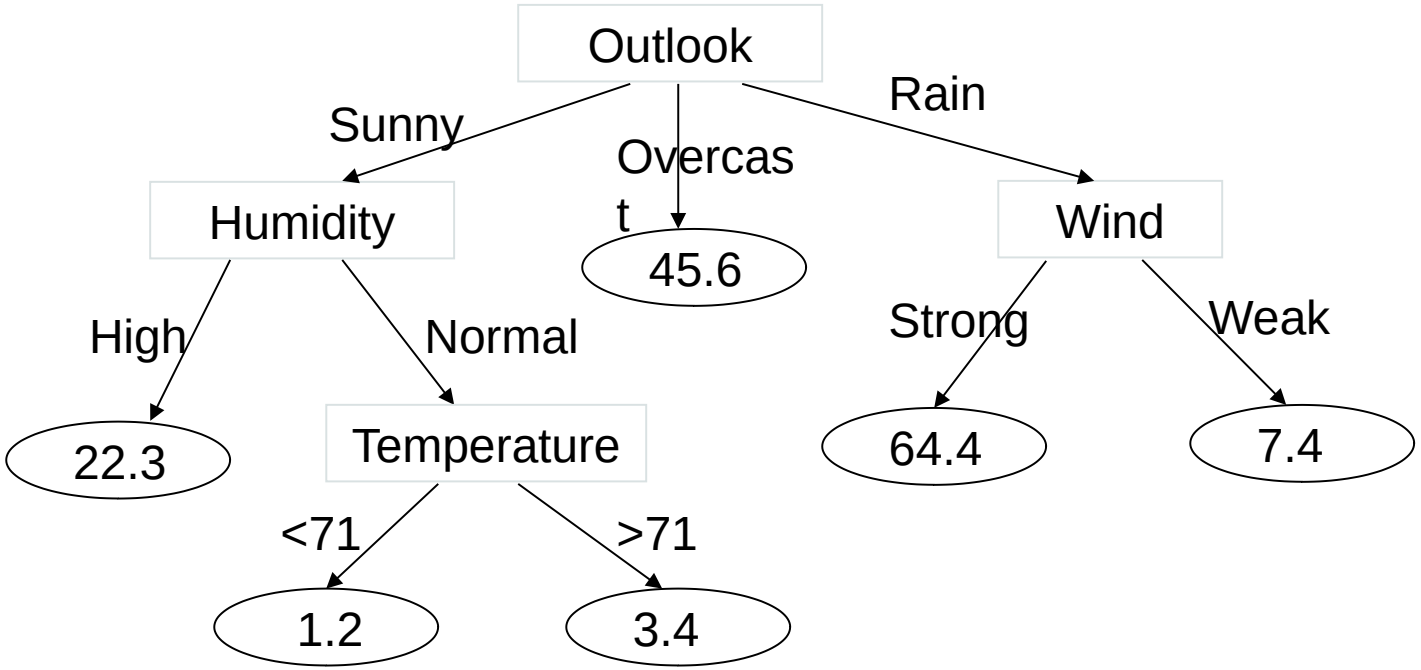
- *T_1 is the complete tree*
- *T_i is obtained by removing some test nodes from T_{i-1}*

Select the tree T_i^* from the sequence that minimizes the error on a validation sample set

4. *Regression trees for prediction*

4. Regression trees for prediction

A regression Tree : ***Has exactly the same model as a decision tree, but with a number (called the prediction value) in each leaf instead of a class***



4. Regression trees for prediction

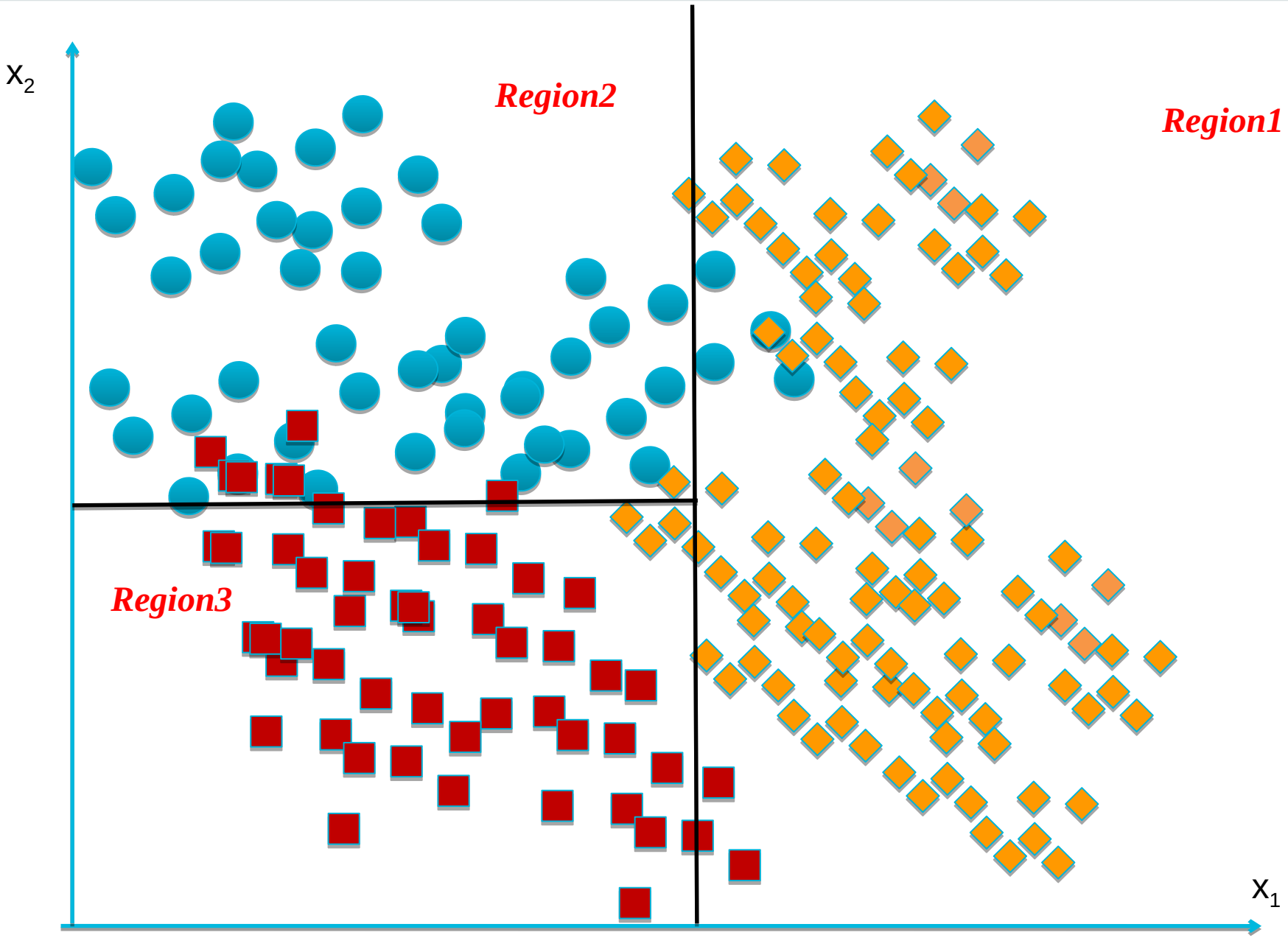
The regression tree targets to segment the "predictor space" into sub-regions and to learn from the training set the value to predict

Building a regression tree

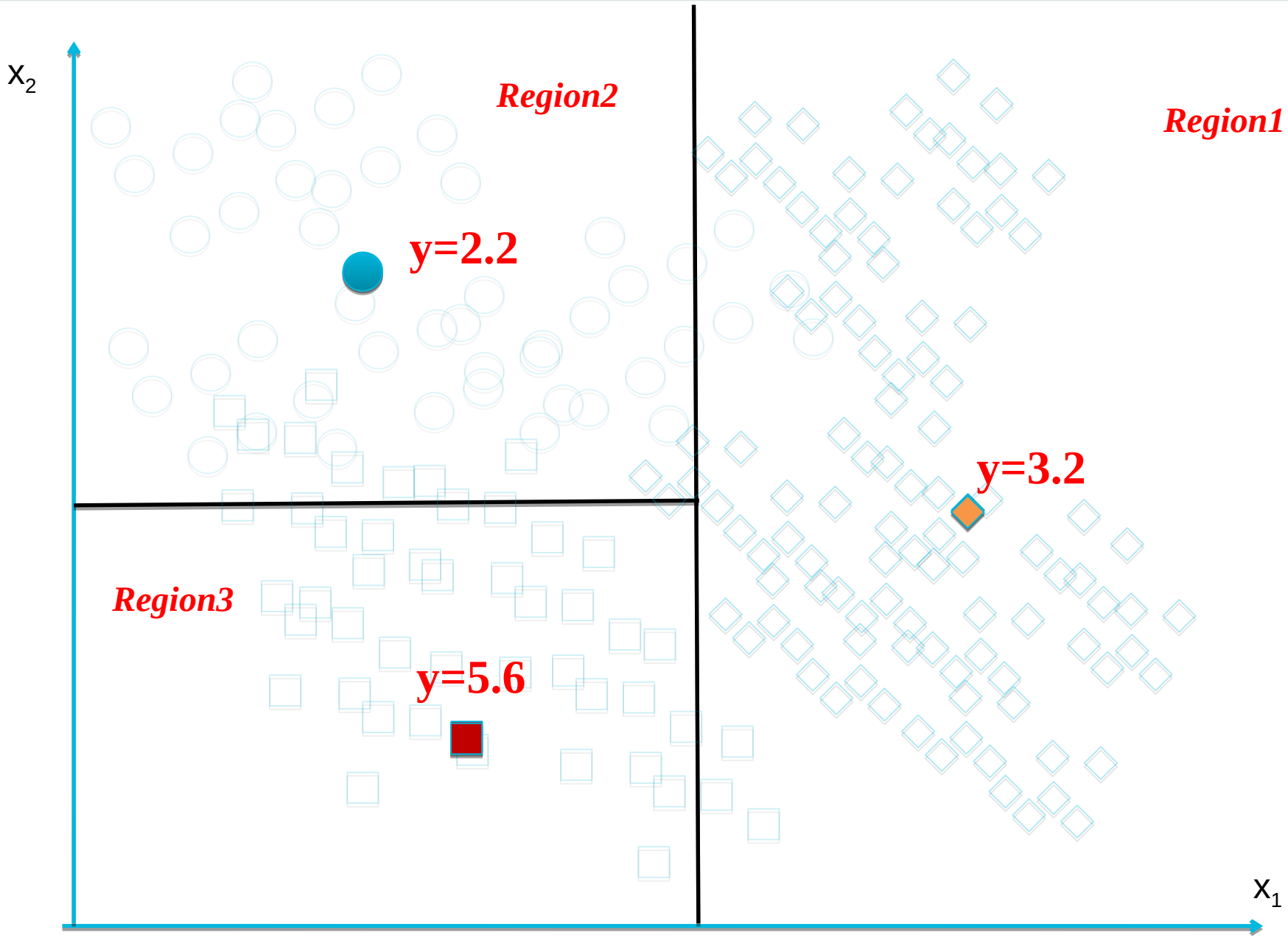
1. *Building a decision tree (using a label information, categorical attribute) followed by*

The use, as prediction values) of the mean or mode or median of the variable (to predicted) of the training examples in sub-regions

4. Regression trees for prediction



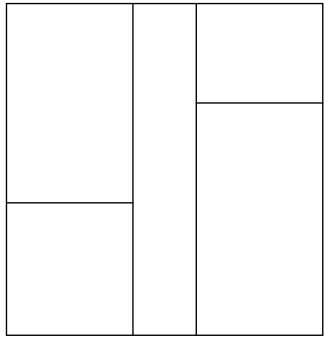
4. Regression trees for prediction



Building a regression tree

2. *Direct building the regression tree*

(without the step of constructing a decision tree)



Find boxes $R_1, \ldots, R_M$ that minimize :

$$\sum_{m=1,..,M} \sum_{\text{Sample(i)} \in R_m} \{ [x_i - x_i(R_M)]^2$$

$x_i(R_m)$: *mean response value of all training observations in the* R_m *region*