INSTITUT DE STATISTIQUES APPLIQUÉES
ET DE SCIENCE DES DONNÉES

AIMS
African Institute for
Mathematical Sciences

Decision trees

- Generalities,
- Definitions,
- Examples and implementation in Scikit-Learn

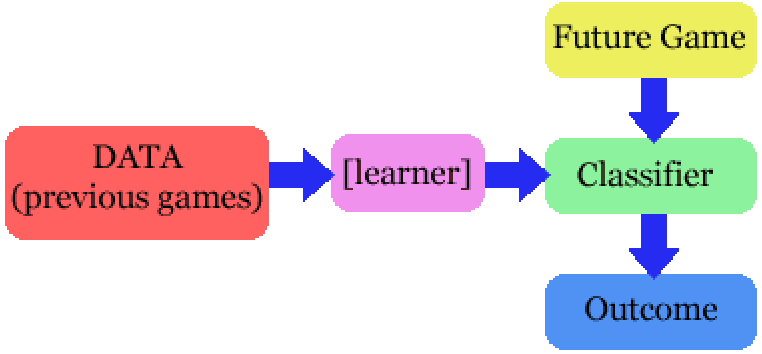
Souleyman N'Doye,
Data Scientist

Outline

- Introduction
- Example
- Principles
  - Entropy
  - Information gain
- Evaluations
- Demo

The problem

- Given a set of training cases/objects and their attribute values, try to determine the target attribute value of new examples.
  - Classification
  - Prediction



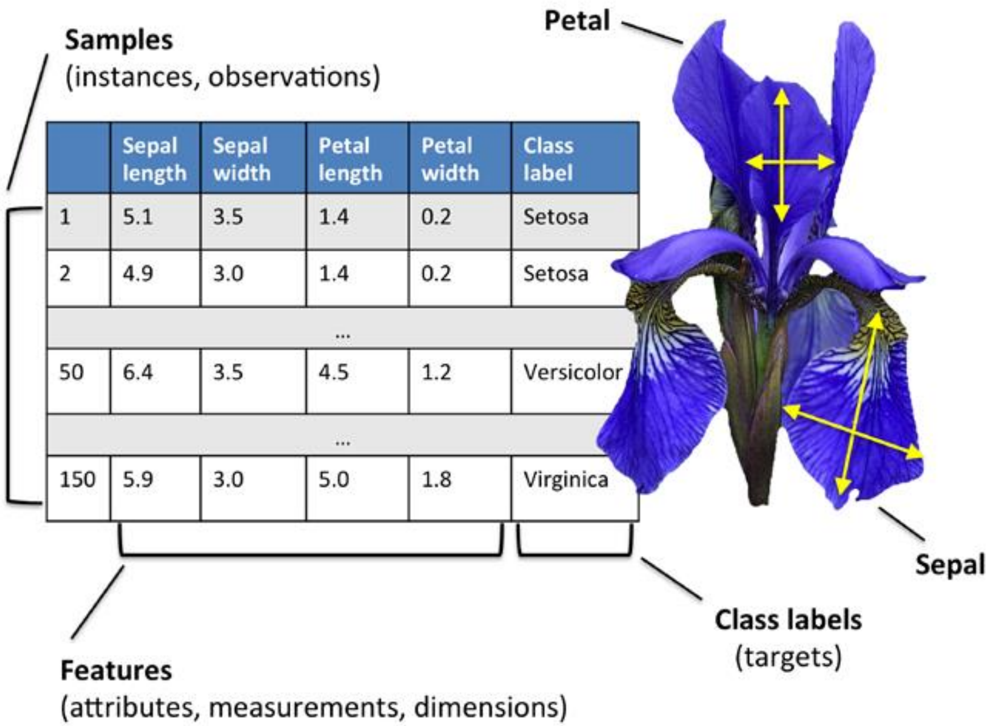
Why decision Trees

- Decision trees are powerful and popular tools for classification and prediction.
- Decision trees represent *rules*, which can be understood by humans and used in knowledge system such as database.

Keys requirements

- **Attribute-value description:** object or case must be expressible in terms of a fixed collection of properties or attributes (e.g., hot, mild, cold).
- **Predefined classes (target values):** the target function has **discrete output values** (bollean or multiclass)
- **Sufficient data:** enough training cases should be provided to learn the model.

A classical example: Fisher Iris

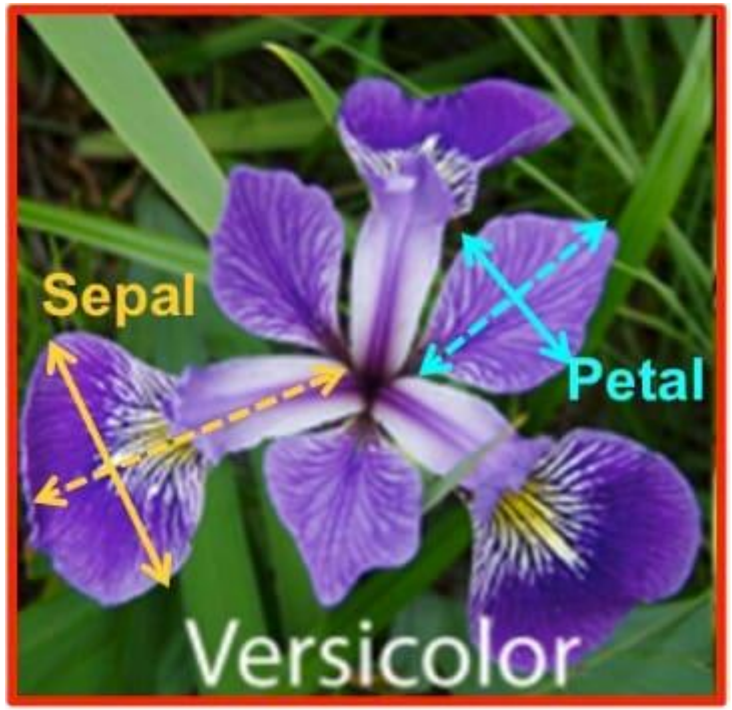


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

4 attributes

1 target of multiple groups

150 observations

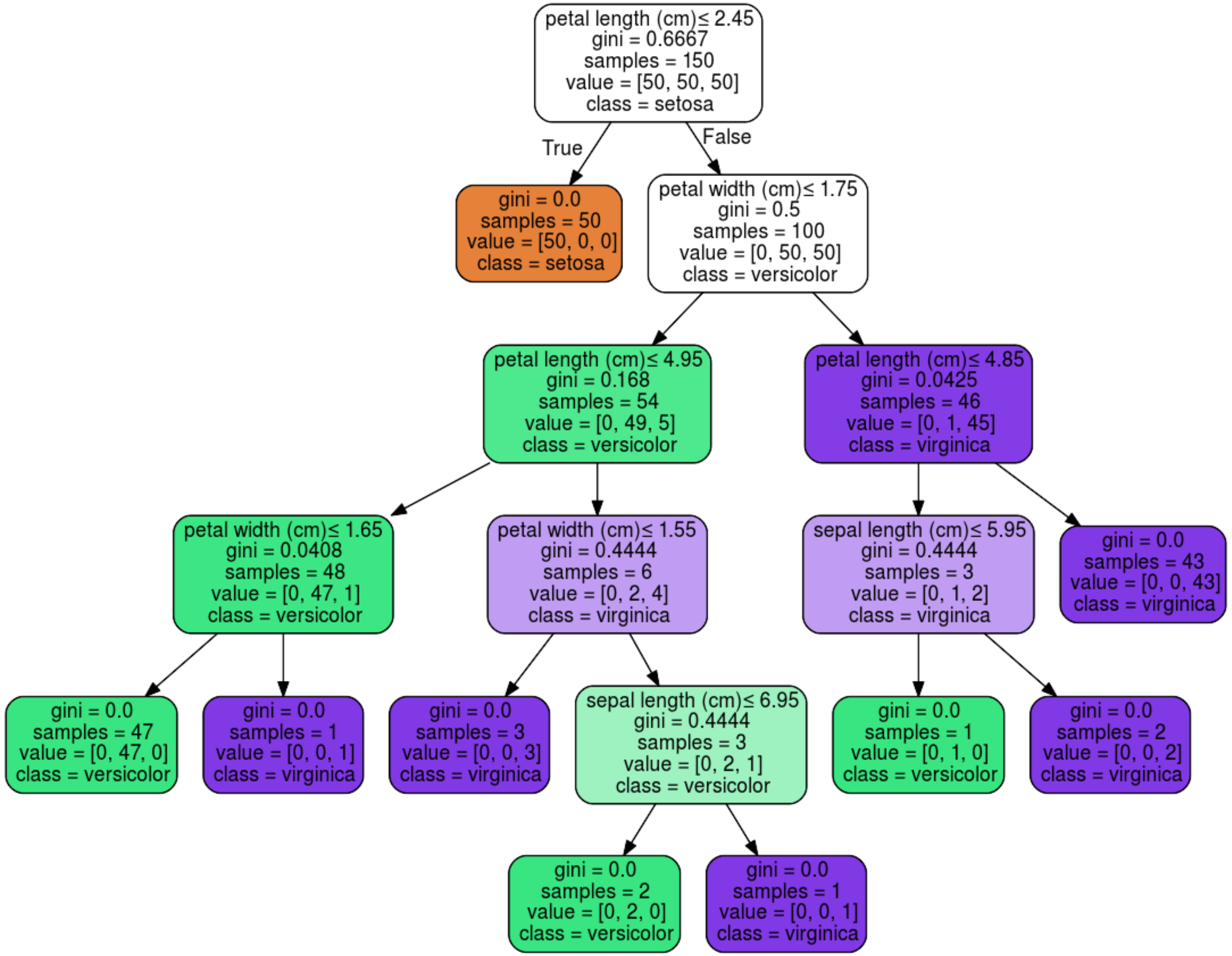


Definition

- Decision tree is a classifier in the form of a tree structure
  - Decision node: specifies a test on a single attribute
  - Leaf node: indicates the value of the target attribute
  - Arc/edge: split of one attribute
  - Path: a disjunction of test to make the final decision

- Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

Illustration



Random split

- The tree can grow huge
- These trees are hard to understand.
- Larger trees are typically less accurate than smaller trees

Principled criterion

- Selection of an attribute to test at each node - choosing the most useful attribute for classifying examples.
- Information gain
  - measures how well a given attribute separates the training examples according to their target classification
  - This measure is used to select among the candidate attributes at each step while growing the tree

Entropy

- A measure of homogeneity of the set of examples.
- Given a set S of positive and negative examples of some target concept (a 2-class problem), the entropy of set S relative to this binary classification is:

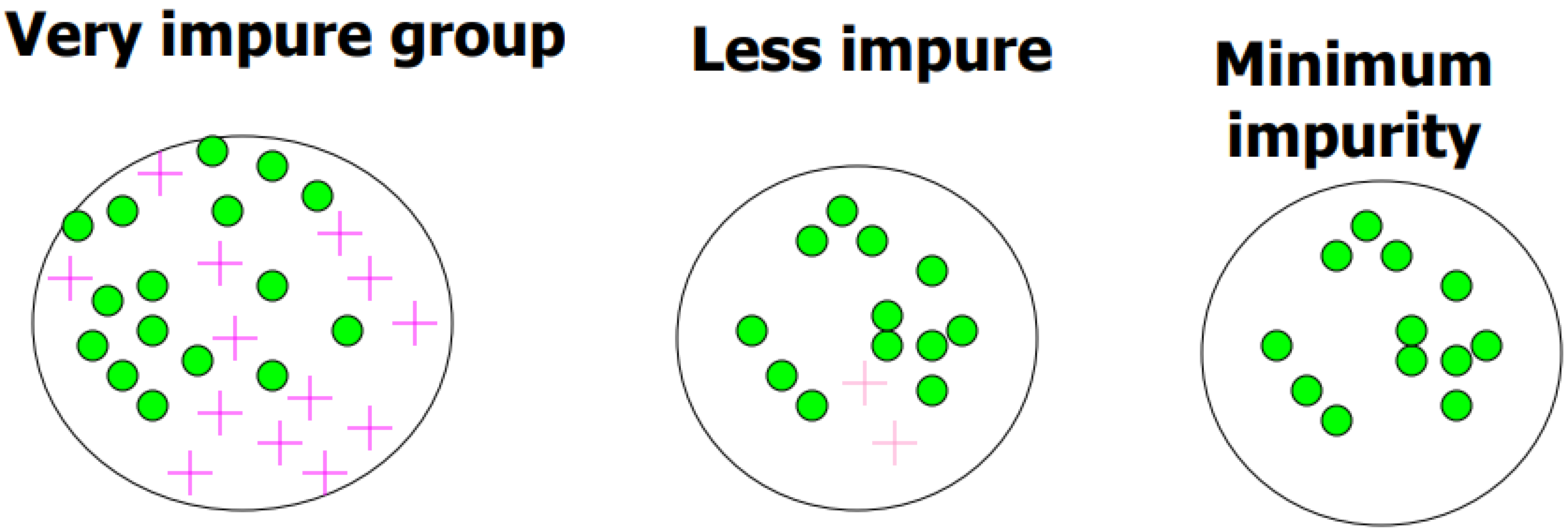
$$E(S) = - p(P)\log2\ p(P) – p(N)\log2\ p(N)$$

- Suppose S has 25 examples, 15 positive and 10 negatives [15+, 10-]. Then the entropy of S relative to this classification is:

$$E(S)=-(15/25)\ \log2(15/25)\ -\ (10/25)\ \log2\ (10/25)$$

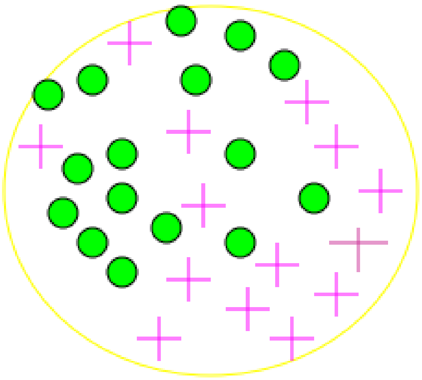
Entropy

- Measures the level of impurity in a group of examples/ Entropy (informal)



Entropy

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



  p_i is the probability of class i
  Compute it as the proportion of class i in the set.

  16/30 are green circles; 14/30 are pink crosses
  $\log_2(16/30) = -.9$; $\log_2(14/30) = -1.1$
  Entropy = -(16/30)(-.9) –(14/30)(-1.1) = .99

- Entropy comes from information theory. The higher the entropy the more the information content.

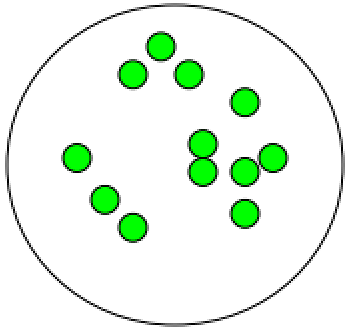
  What does that mean for learning from examples?

Entropy : 2 classes cases

- What is the entropy of a group in which all examples belong to the same class?
  - entropy = - 1 $\log_2 1$ = 0

not a good training set for learning

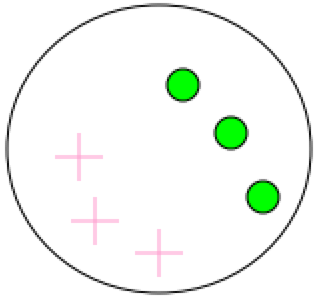
Minimum impurity



- What is the entropy of a group with 50% in either class?
  - entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

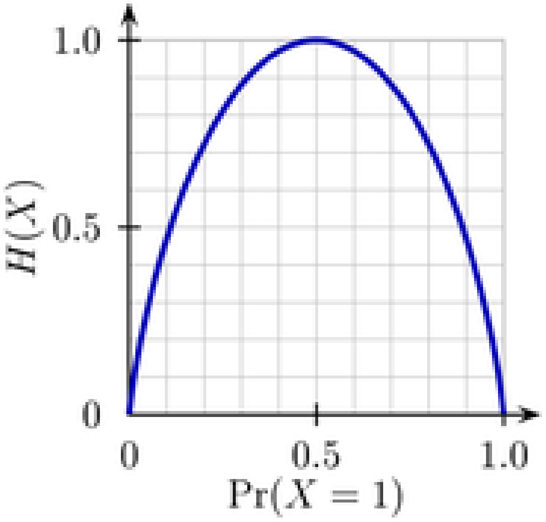
good training set for learning

Maximum impurity



Some intuitions

- The entropy is 0 if the outcome is ``certain''.
- The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).



Entropy of a 2-class problem with regard to the portion of one of the two groups

Information Gain

- Information gain measures the expected reduction in entropy, or uncertainty.
  - Values(A) is the set of all possible values for attribute A, and Sv the subset of S for which attribute A has value v Sv = {s in S | A(s) = v}.
  - the first term in the equation for *Gain* is just the entropy of the original collection *S*
  - the second term is the expected value of the entropy after S is partitioned using attribute A

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Information Gain

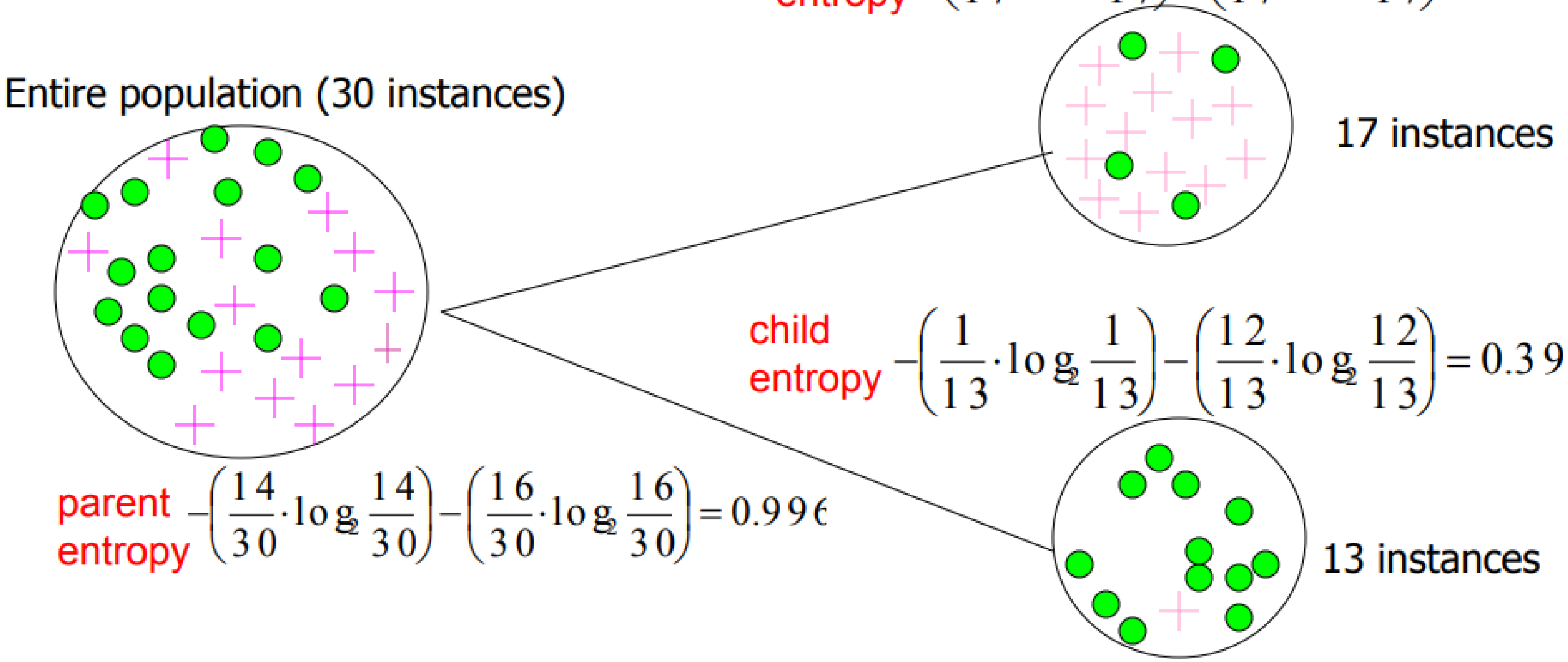
- It is simply the expected reduction in entropy caused by partitioning the examples according to this attribute.
- It is the number of bits saved when encoding the target value of an arbitrary member of *S*, by knowing the value of attribute *A.*
- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.
- Information gain tells us how important a given attribute of the feature vectors is.
- We will use it to decide the ordering of attributes in the nodes of a decision tree.

Information Gain

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.78$

child entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.39$



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

(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38 for this split

Exemple 1

Training Set: 3 features and 2 classes

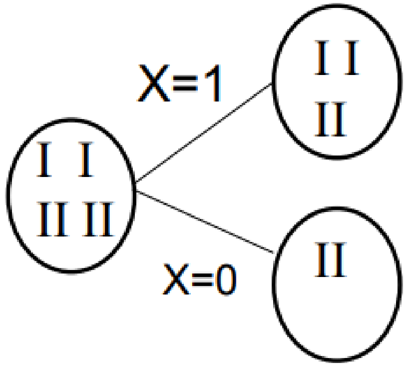
X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

How would you distinguish class I from class II?

Exemple 1

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Split on attribute X



If X is the best attribute,
this node would be further split.

$E_{child1} = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3)$
$= .5284 + .39$
$= .9184$

$E_{child2} = 0$

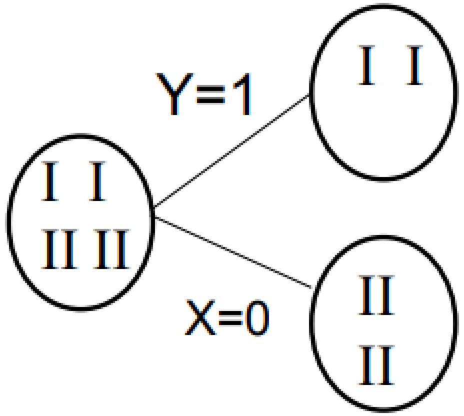
$E_{parent} = 1$

$\text{GAIN} = 1 – (3/4)(.9184) – (1/4)(0) = .3112$

Exemple 1

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Split on attribute Y



$E_{child1} = 0$

$E_{child2} = 0$

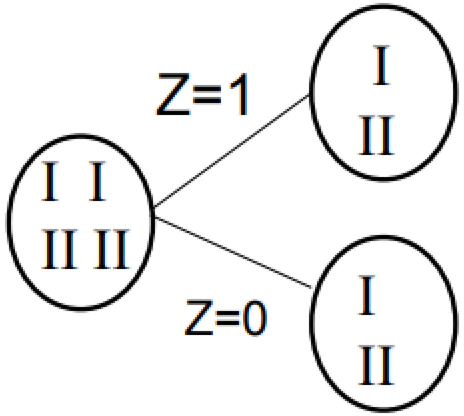
$E_{parent} = 1$

GAIN = 1 –(1/2) 0 – (1/2)0 = 1; BEST ONE

Exemple 1

X	Y	Z	C
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II

Split on attribute Z



$E_{child1} = 1$

$E_{child2} = 1$

$E_{parent} = 1$

GAIN = 1 – (1/2)(1) – (1/2)(1) = 0   ie. NO GAIN; WORST

Strength

- can generate understandable rules
- perform classification without much computation
- can handle continuous and categorical variables
- provide a clear indication of which fields are most important for prediction or classification

Weakness

- Perform poorly with many class and small data.
- Computationally expensive to train.

Implementions : Sckit-Learn

- `sklearn.tree`**.DecisionTreeClassifier**