Decision Trees

Daniel Nogueira
dnogueira@ipca.pt

Introduction

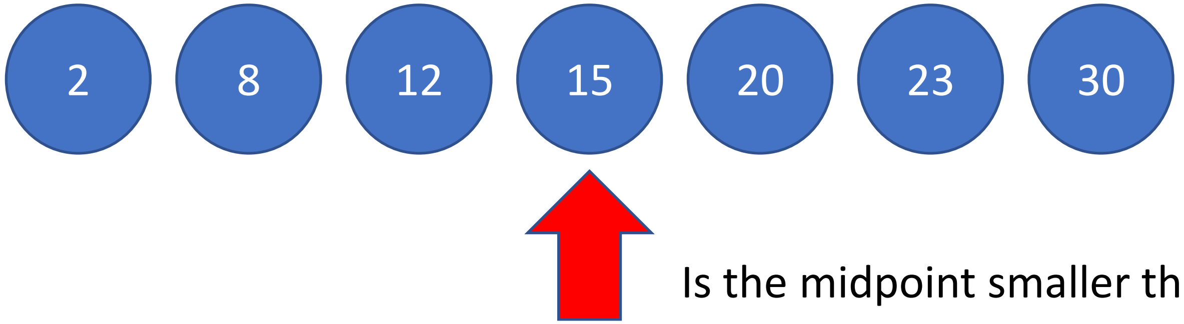
Binary Search

Look for # 30

Introduction

Binary Search

Look for # 30

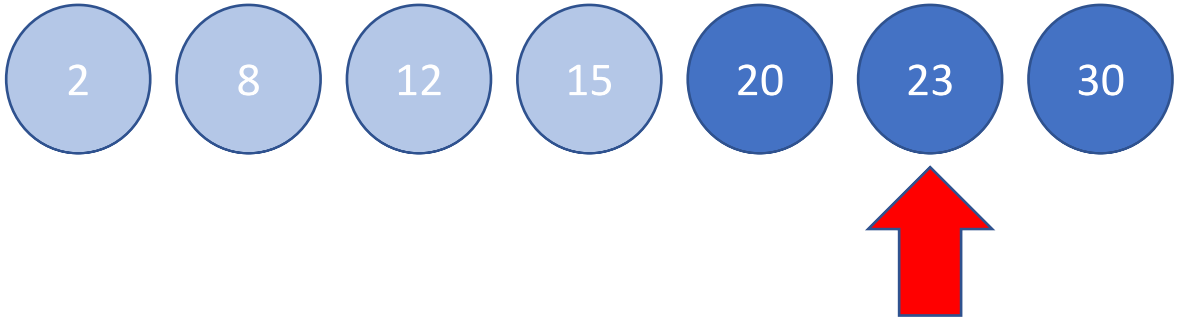


Is the midpoint smaller than the target?

Introduction

Binary Search

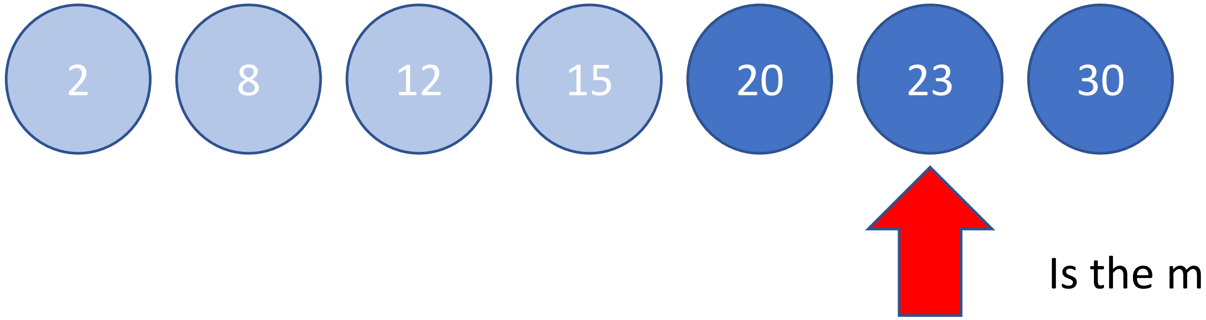
Look for # 30



Introduction

Binary Search

Look for # 30

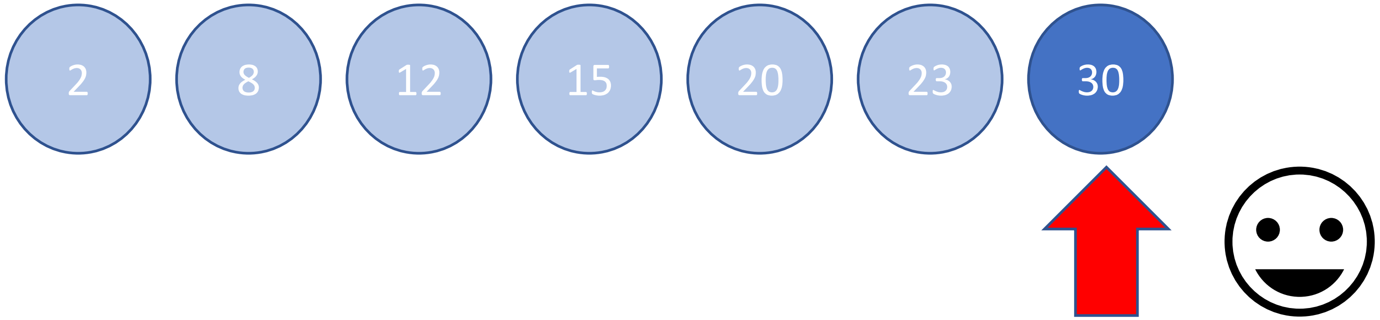


Is the midpoint smaller than the target?

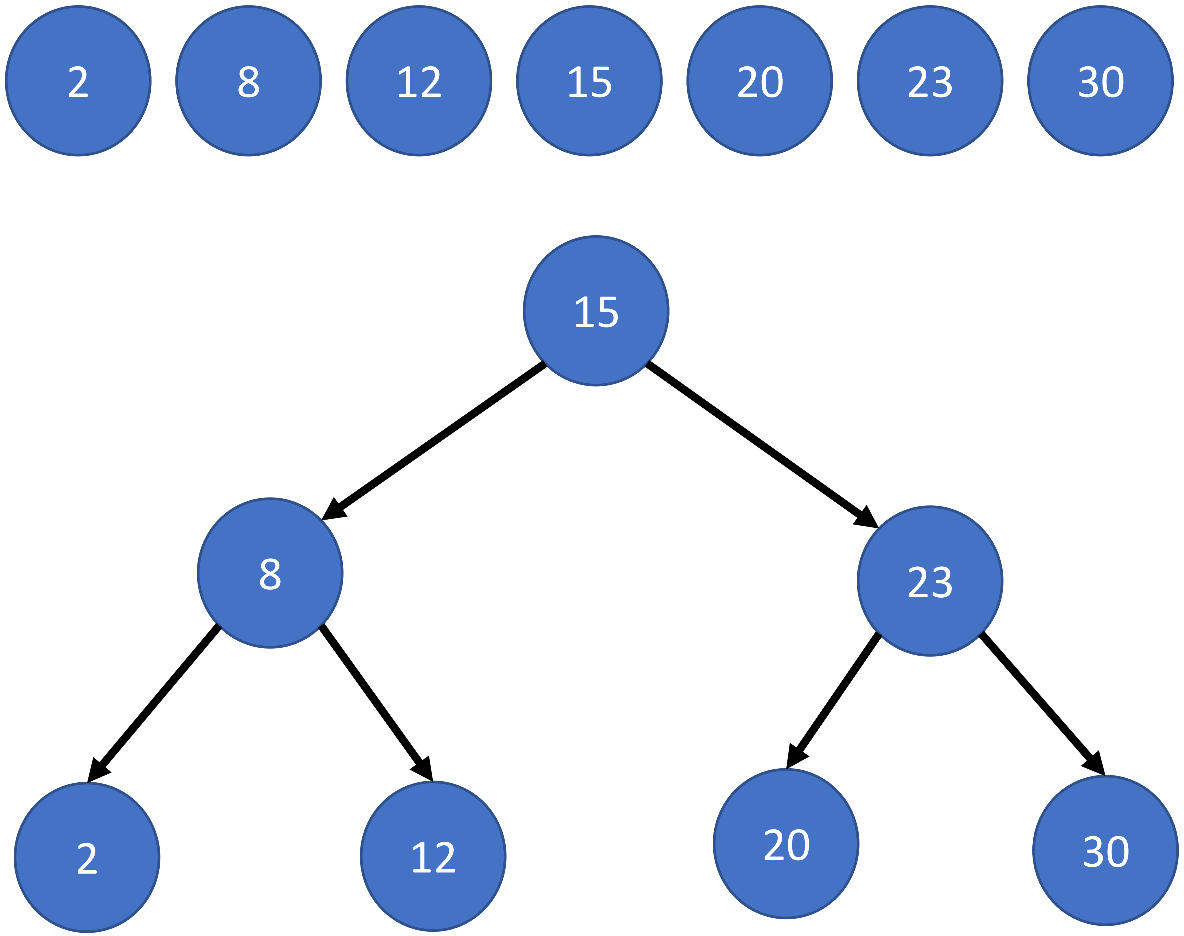
Introduction

Binary Search

Look for # 30



Introduction

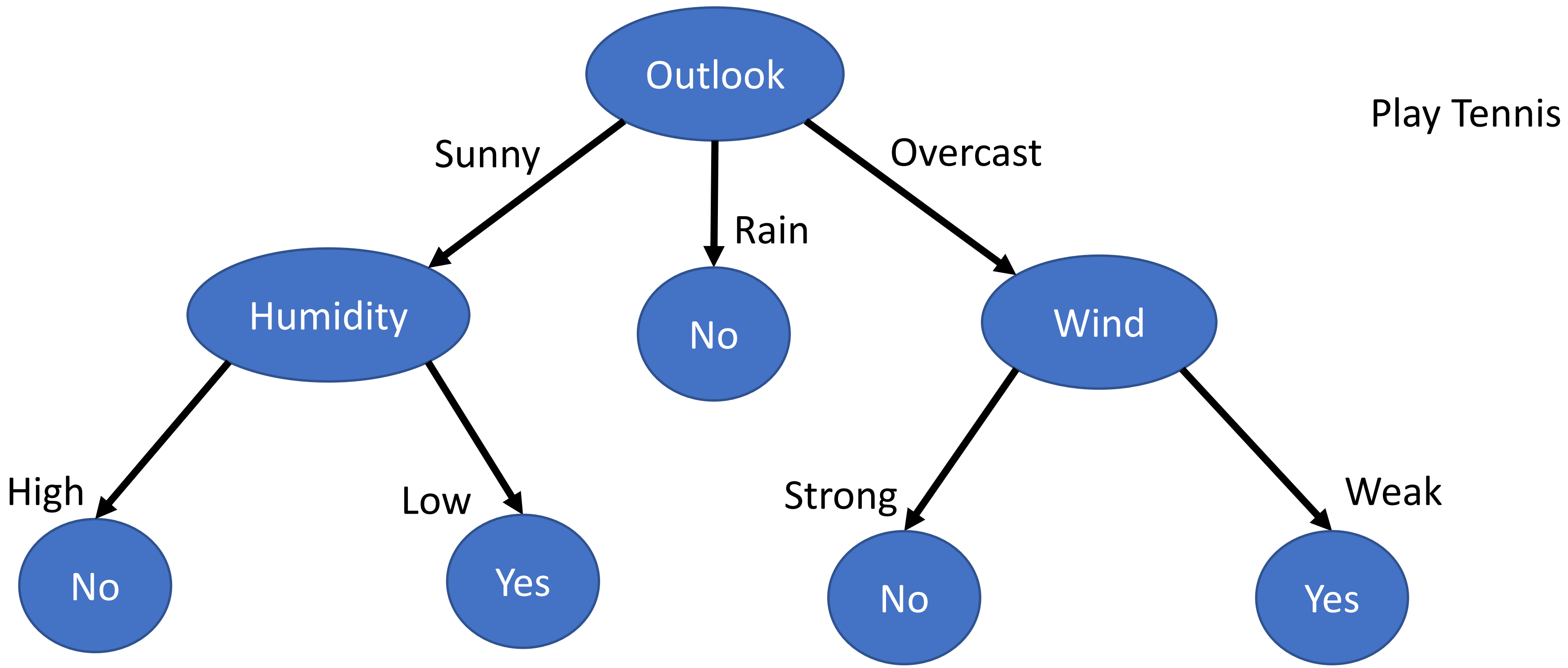


Introduction

Decision Tree → The decision tree progressively subdivides the data into smaller and more specific sets, in terms of their attributes, until they reach a size simplified enough to be labelled. The decision tree's main goal is to predict a class within a finite set of options.

Introduction

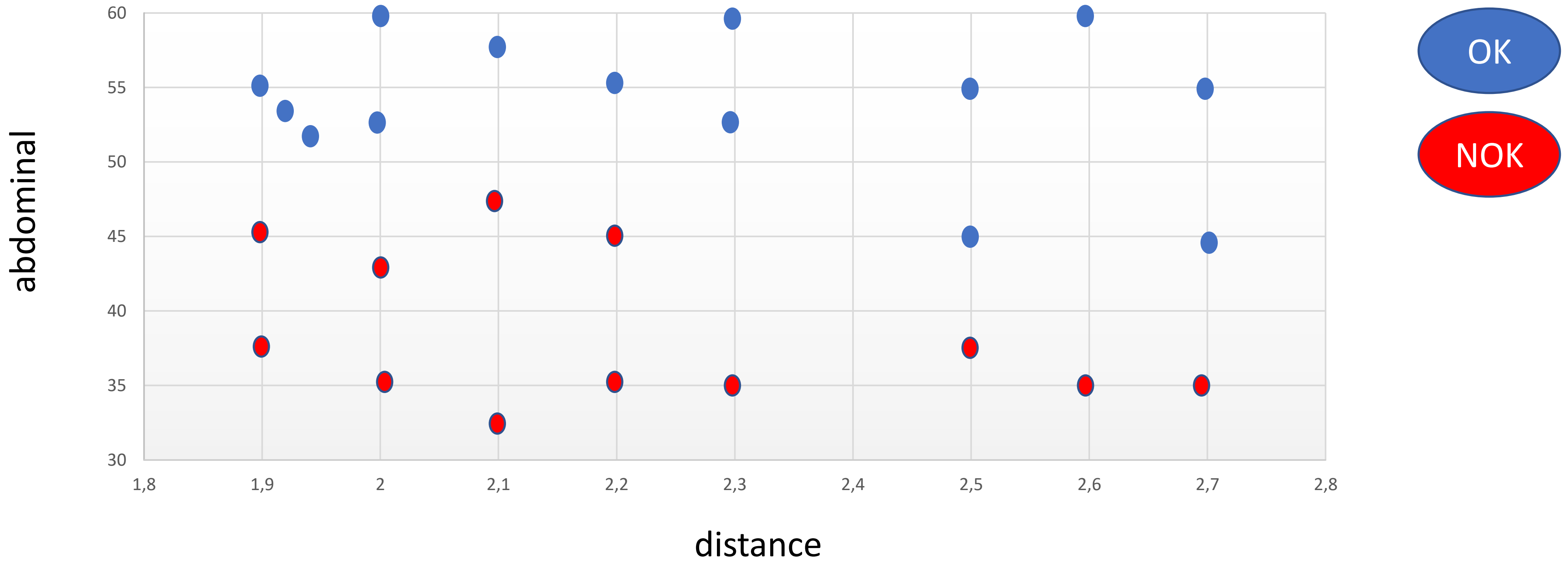
Decision Tree → The decision tree progressively subdivides the data into smaller and more specific sets, in terms of their attributes, until they reach a size simplified enough to be labelled. The decision tree's main goal is to predict a class within a finite set of options.



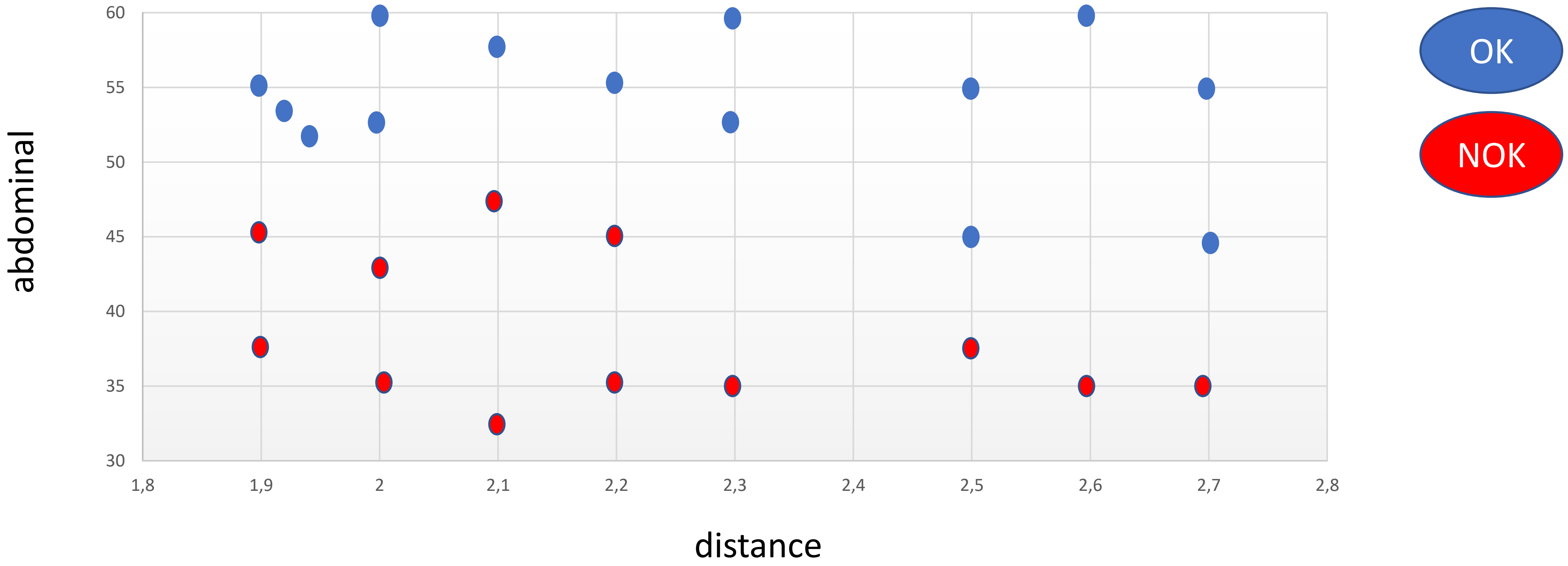
Introduction

Decision Tree

The decision tree progressively subdivides the data into smaller and more specific sets, in terms of their attributes, until they reach a size simplified enough to be labelled. The decision tree's main goal is to predict a class within a finite set of options.



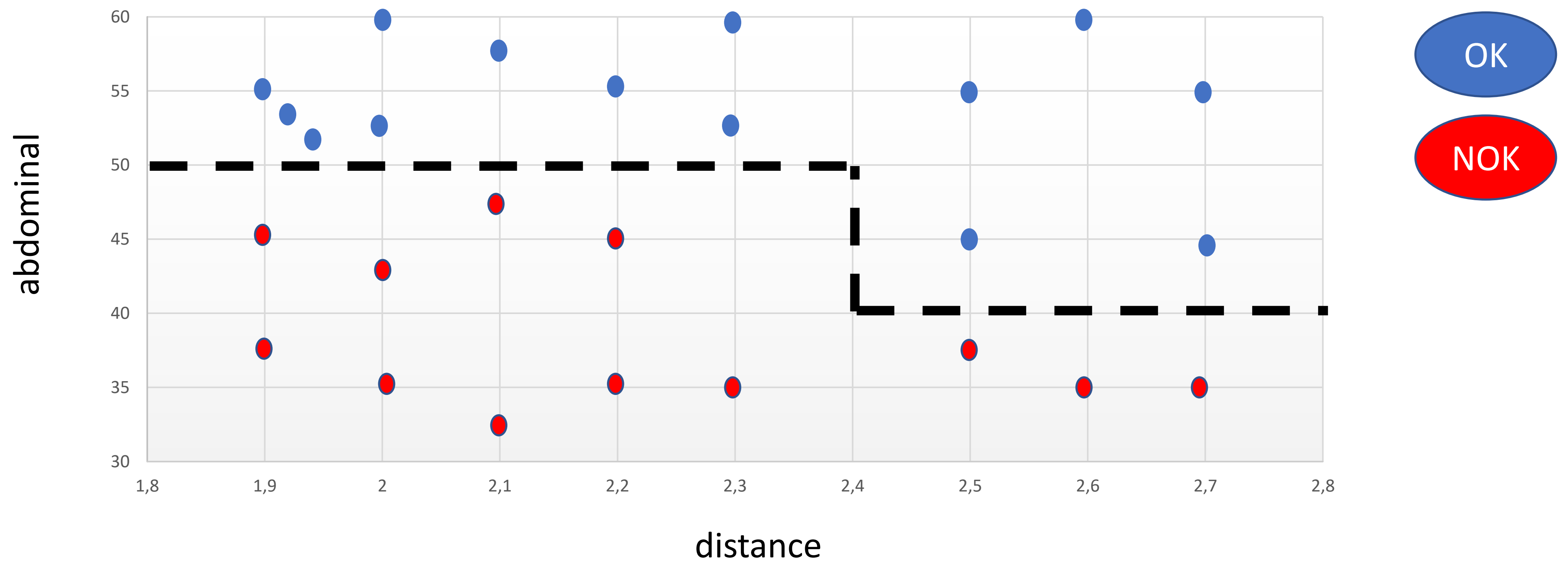
Introduction



Introduction

Decision Tree

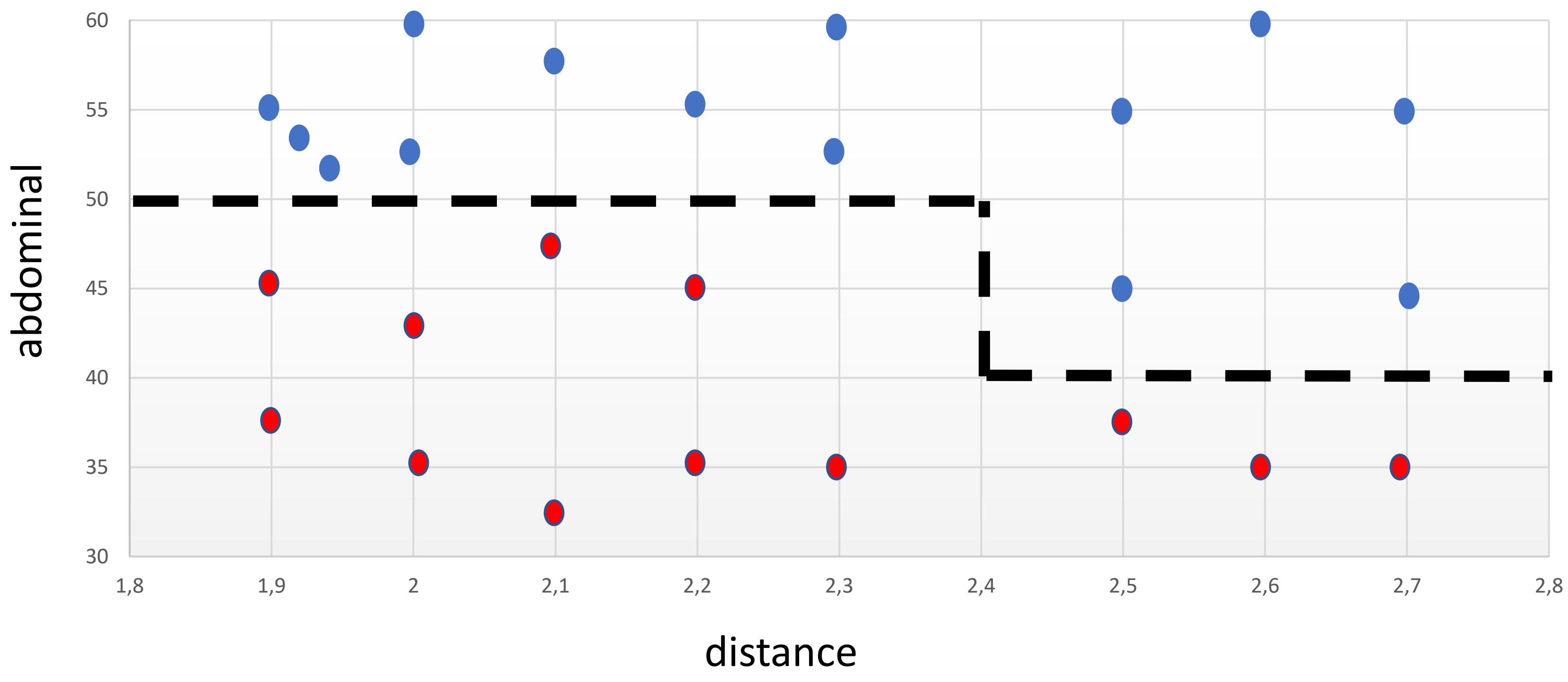
The decision tree progressively subdivides the data into smaller and more specific sets, in terms of their attributes, until they reach a size simplified enough to be labelled. The decision tree's main goal is to predict a class within a finite set of options.

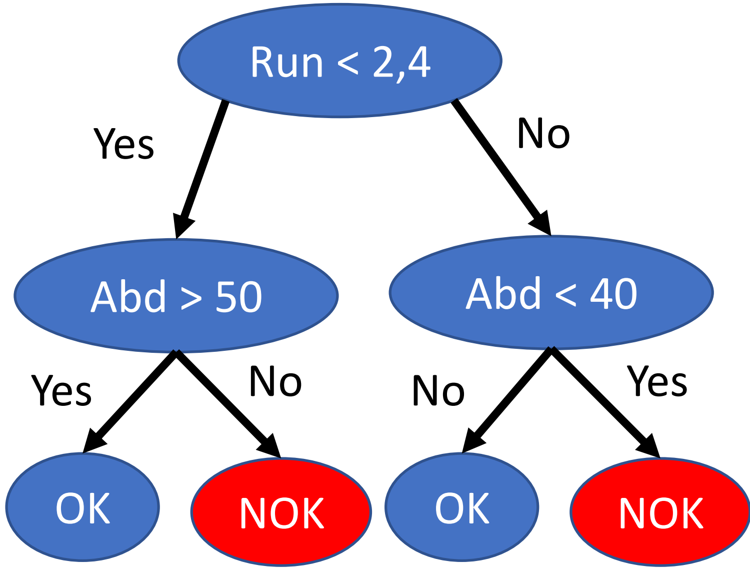


Introduction

Decision Tree

The decision tree progressively subdivides the data into smaller and more specific sets, in terms of their attributes, until they reach a size simplified enough to be labelled. The decision tree's main goal is to predict a class within a finite set of options.





Introduction

Decision Tree

Advantage

- Easy to understand: The visualization of a decision tree makes the problem easy to understand, even for people who don't have an analytical profile. It does not require any statistical knowledge to read and interpret. Its graphical representation is very intuitive and allows you to relate the hypotheses easily.

- Useful in data exploration: The decision tree is one of the fastest ways to identify the most significant variables and the relationship between two or more variables. With the help of decision trees, we can create new variables/characteristics that are better able to predict the target variable.

- Less need to clean data: Requires less data cleanup compared to other modelling techniques. Up to a certain level, it is not influenced by outliers or missing values.

- Not restricted by data types: Can handle numeric and categorical variables.

- Non-parametric method: The decision tree is considered a non-parametric method. This means that decision trees do not assume space distribution or classifier structure.

Introduction

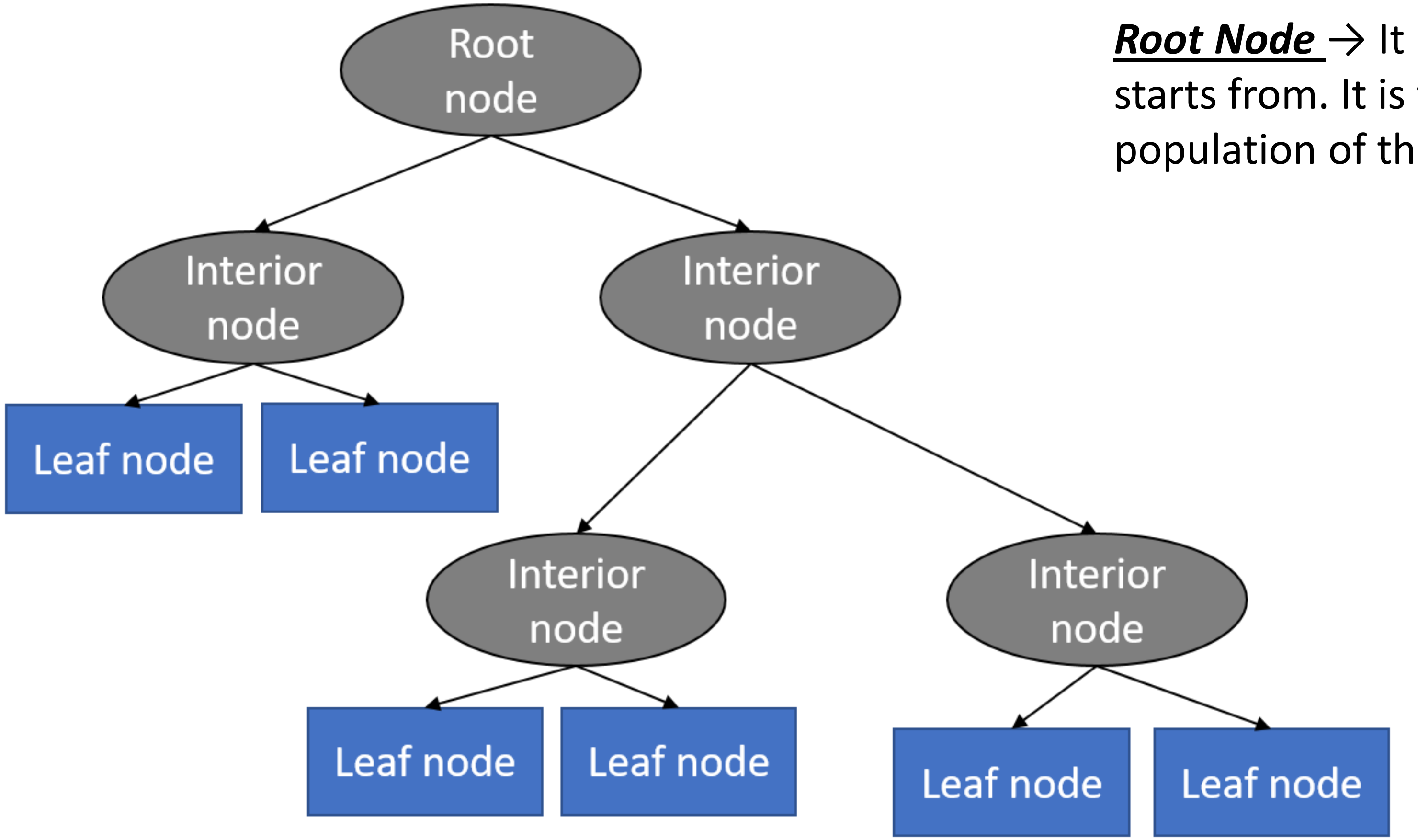
Decision Tree

Disadvantage

- Overfitting: Overfitting is one of the greatest difficulties for decision tree models. This problem is solved by defining constraints on the model and pruning parameters.

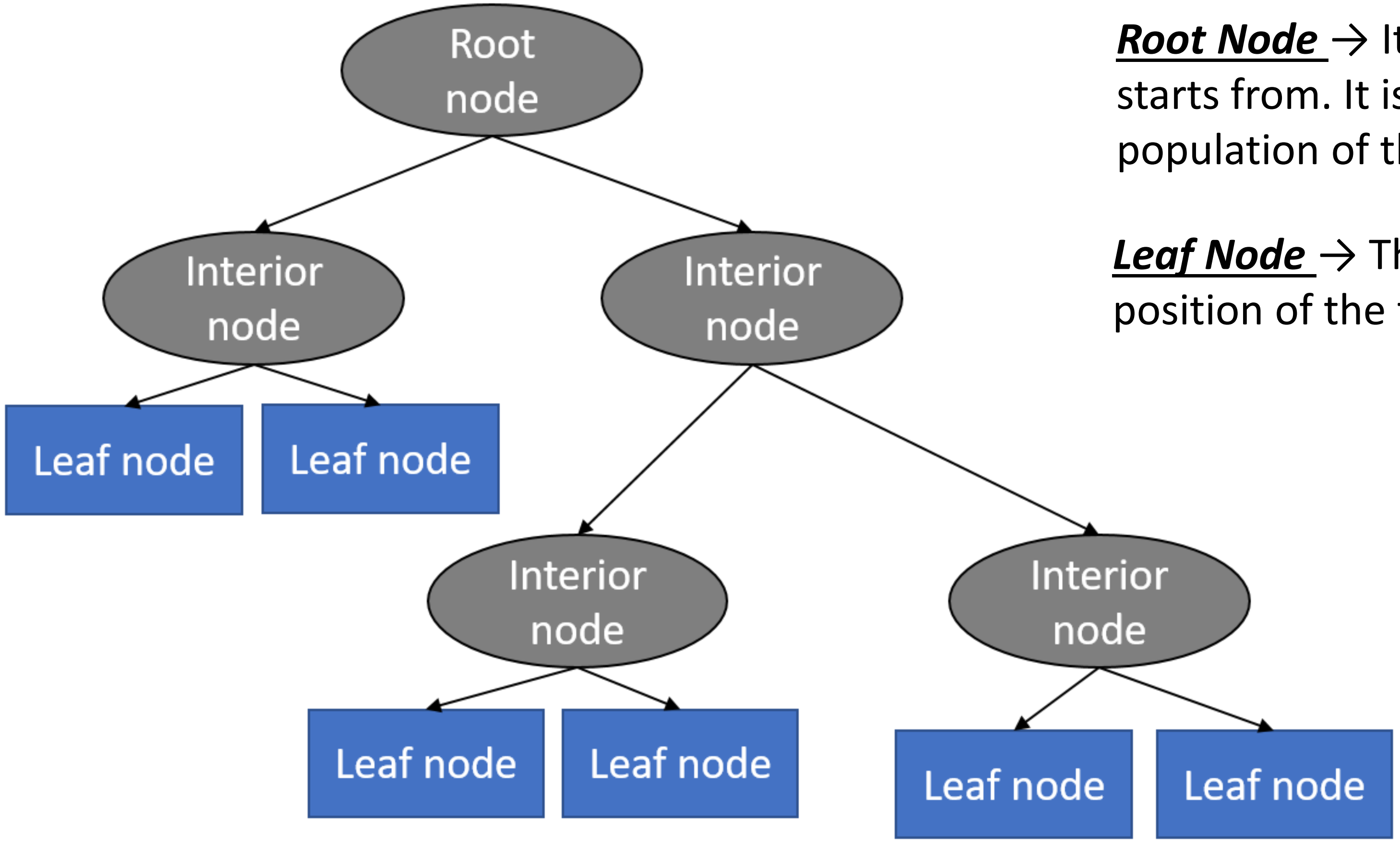
- Not suitable for continuous variables: When working with continuous numeric variables, the decision tree loses information when it categorizes variables into different categories.

Definitions



<u>Root Node</u> → It is the base node from where the entire tree starts from. It is the first node that represent the entire population of the tree.

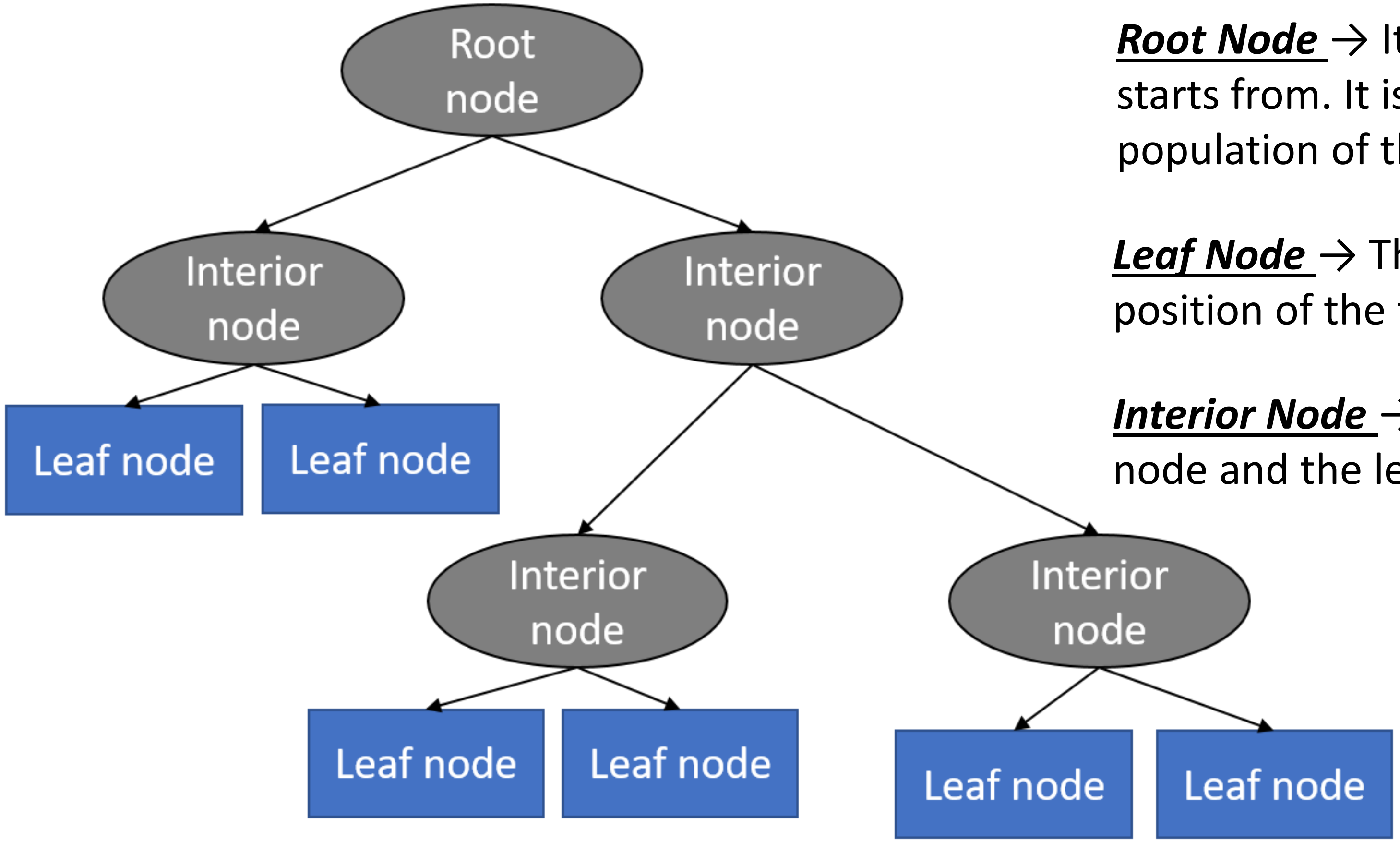
Definitions



Root Node → It is the base node from where the entire tree starts from. It is the first node that represent the entire population of the tree.

Leaf Node → The nodes which are present at the end or last position of the tree.

Definitions

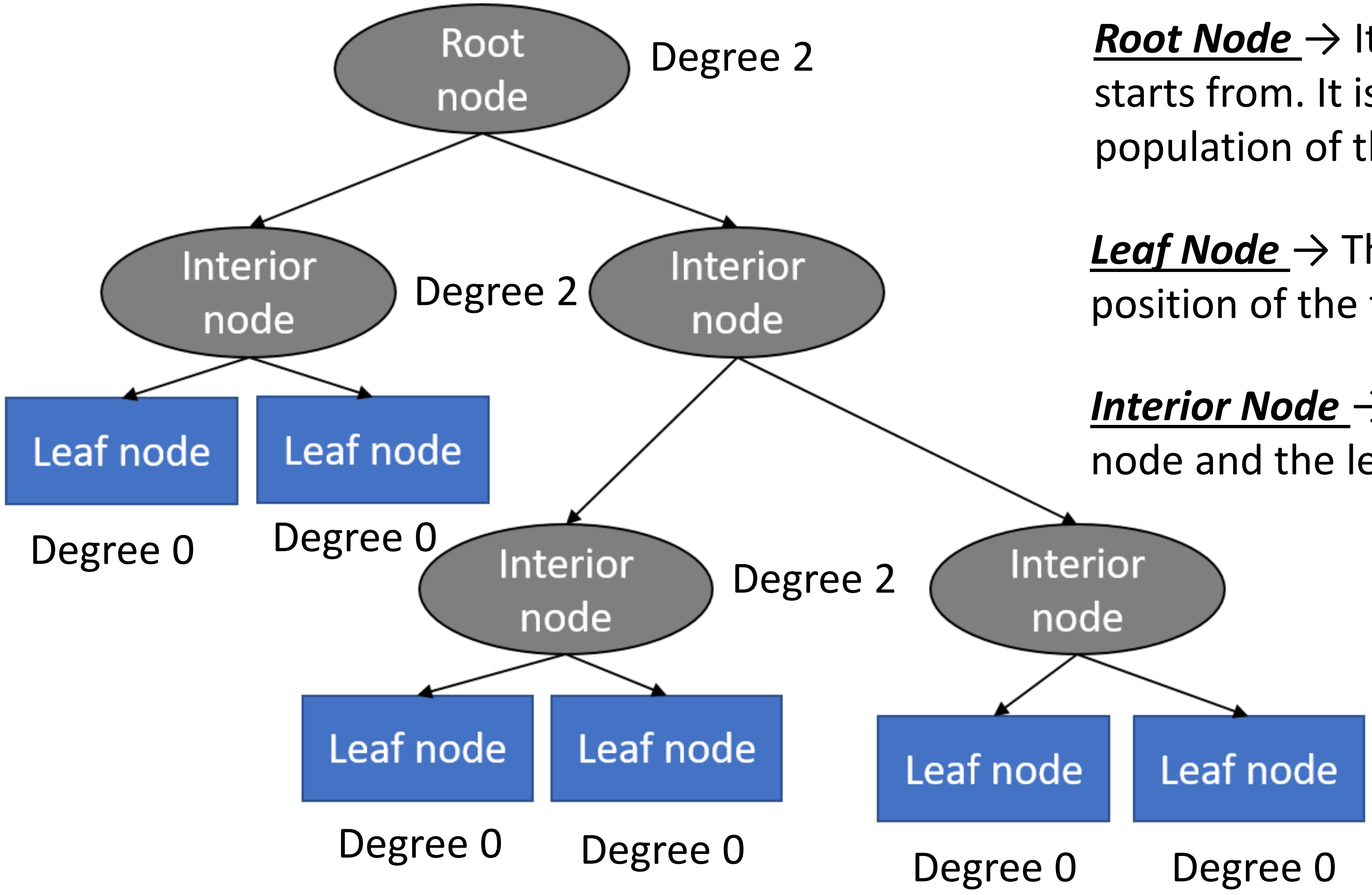


Root Node → It is the base node from where the entire tree starts from. It is the first node that represent the entire population of the tree.

Leaf Node → The nodes which are present at the end or last position of the tree.

Interior Node → The intermediate nodes between the root node and the leaf node.

Definitions

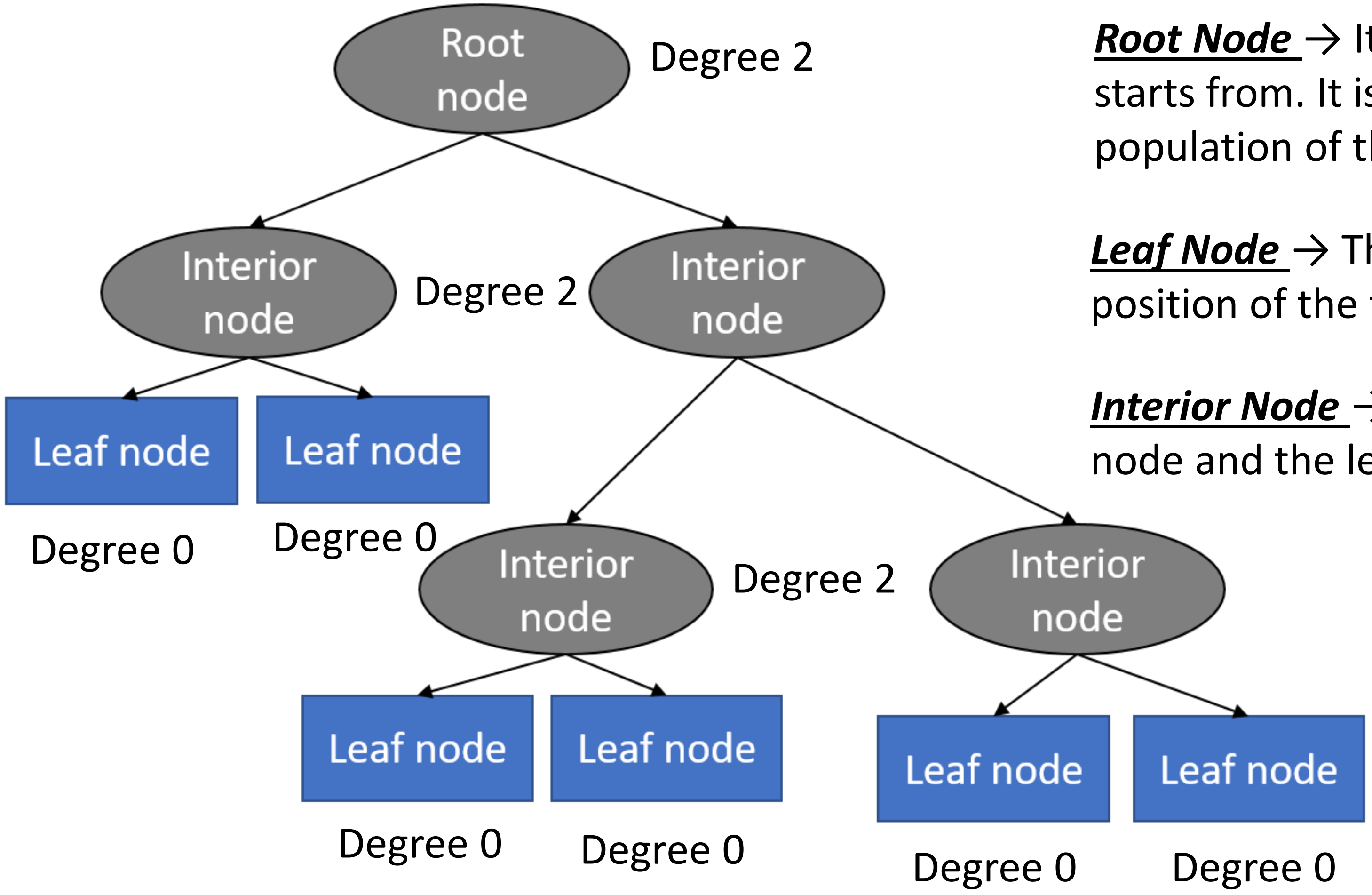


Root Node → It is the base node from where the entire tree starts from. It is the first node that represent the entire population of the tree.

Leaf Node → The nodes which are present at the end or last position of the tree.

Interior Node → The intermediate nodes between the root node and the leaf node.

Definitions



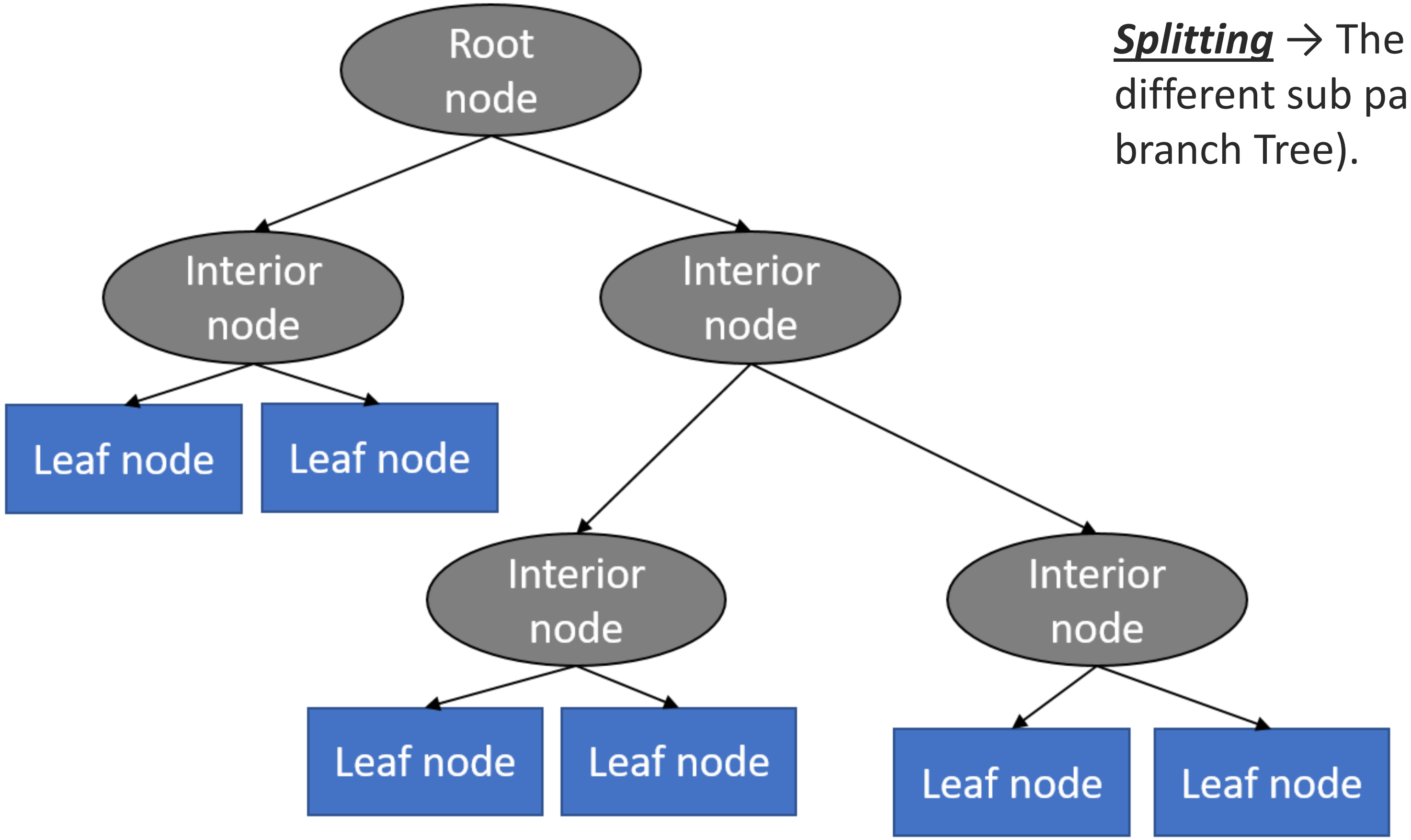
Root Node → It is the base node from where the entire tree starts from. It is the first node that represent the entire population of the tree.

Leaf Node → The nodes which are present at the end or last position of the tree.

Interior Node → The intermediate nodes between the root node and the leaf node.

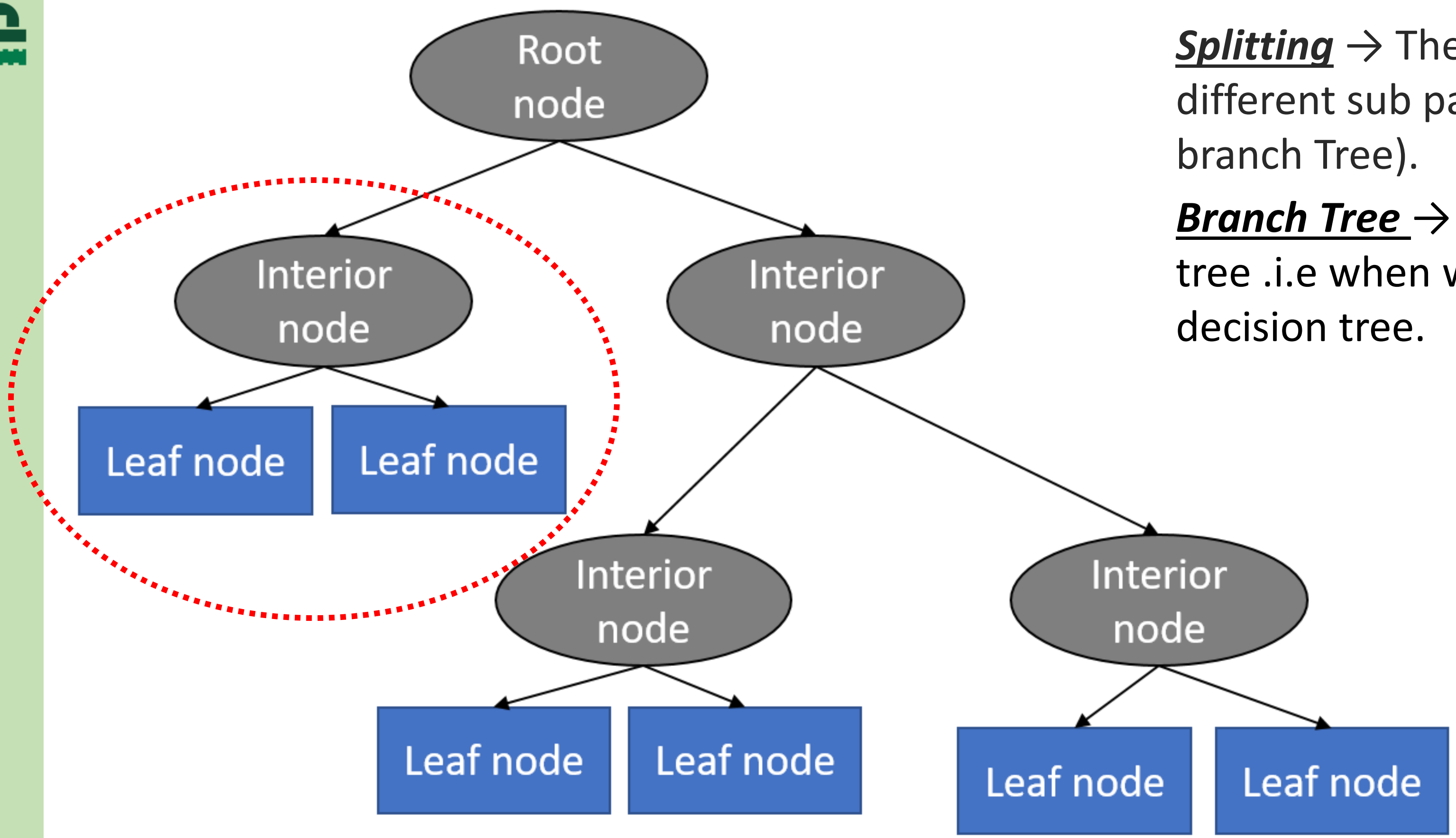
Binary Decision Tree → Node Degree < 3

Definitions



Splitting → The process of dividing the root node into several different sub parts i.e. interior and leaf nodes (subtrees or branch Tree).

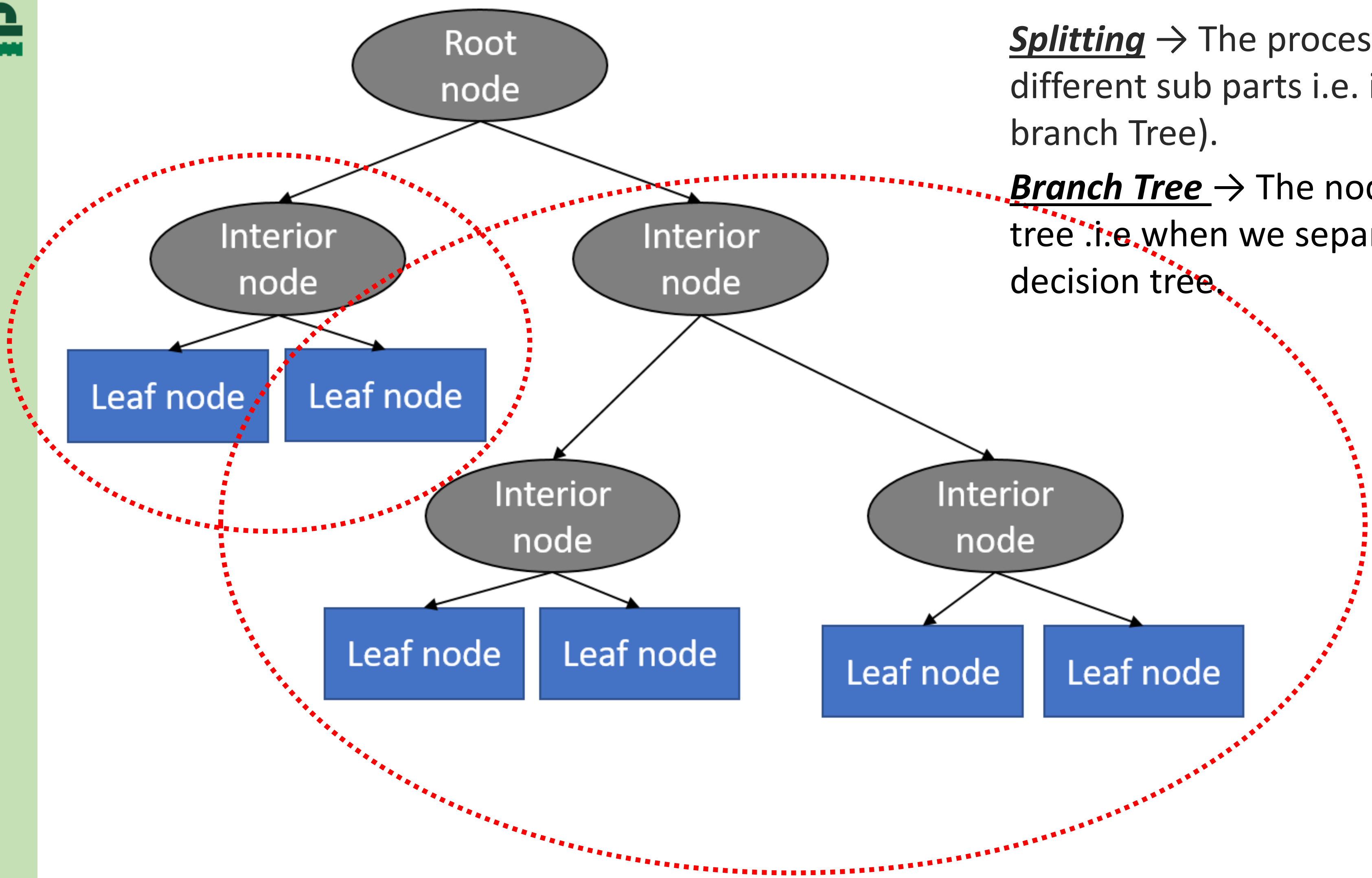
Definitions



Splitting → The process of dividing the root node into several different sub parts i.e. interior and leaf nodes (subtrees or branch Tree).

Branch Tree → The nodes that are generated when we split our tree .i.e when we separate out any particular section of our decision tree.

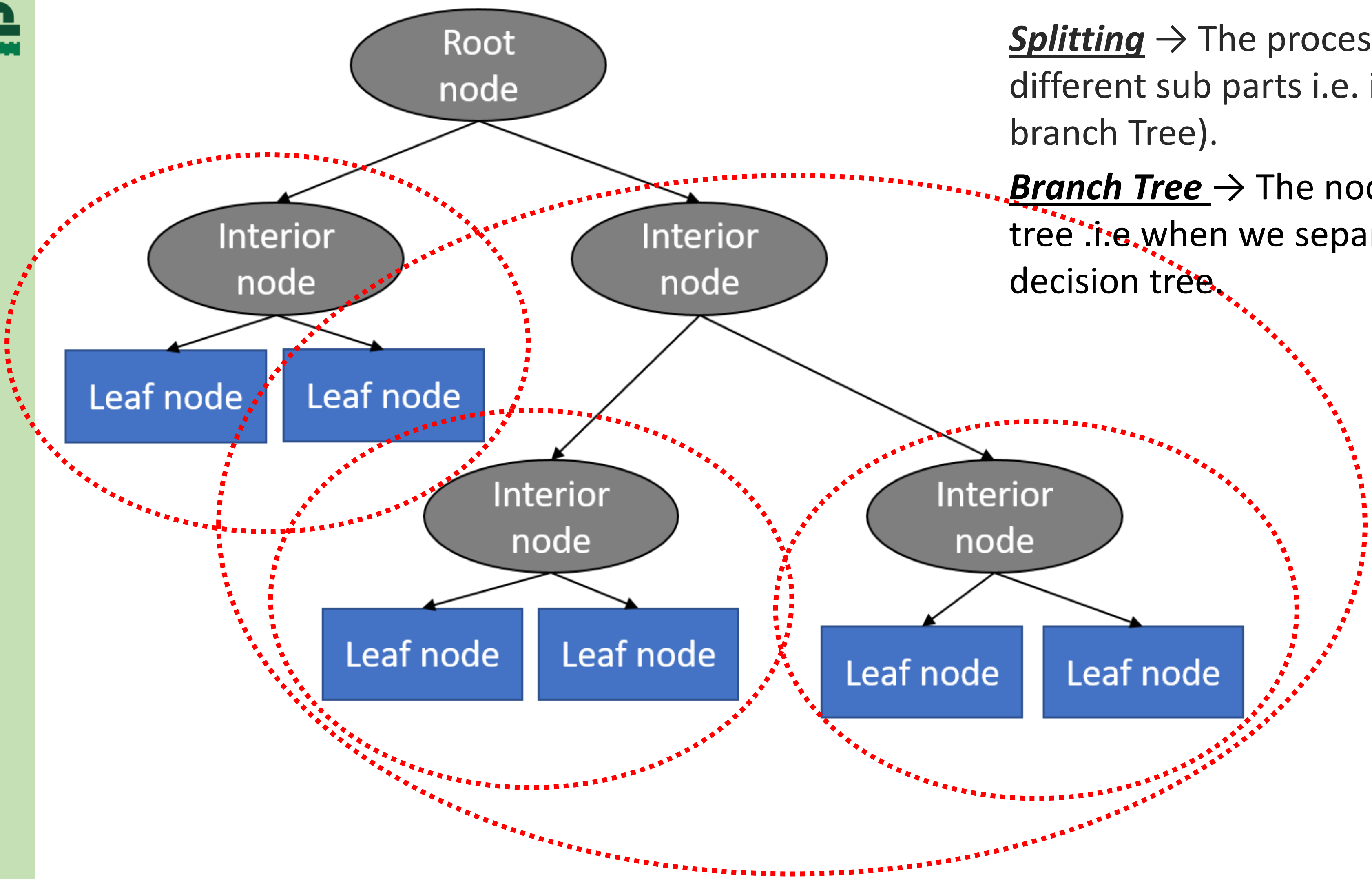
Definitions



Splitting → The process of dividing the root node into several different sub parts i.e. interior and leaf nodes (subtrees or branch Tree).

Branch Tree → The nodes that are generated when we split our tree .i.e when we separate out any particular section of our decision tree.

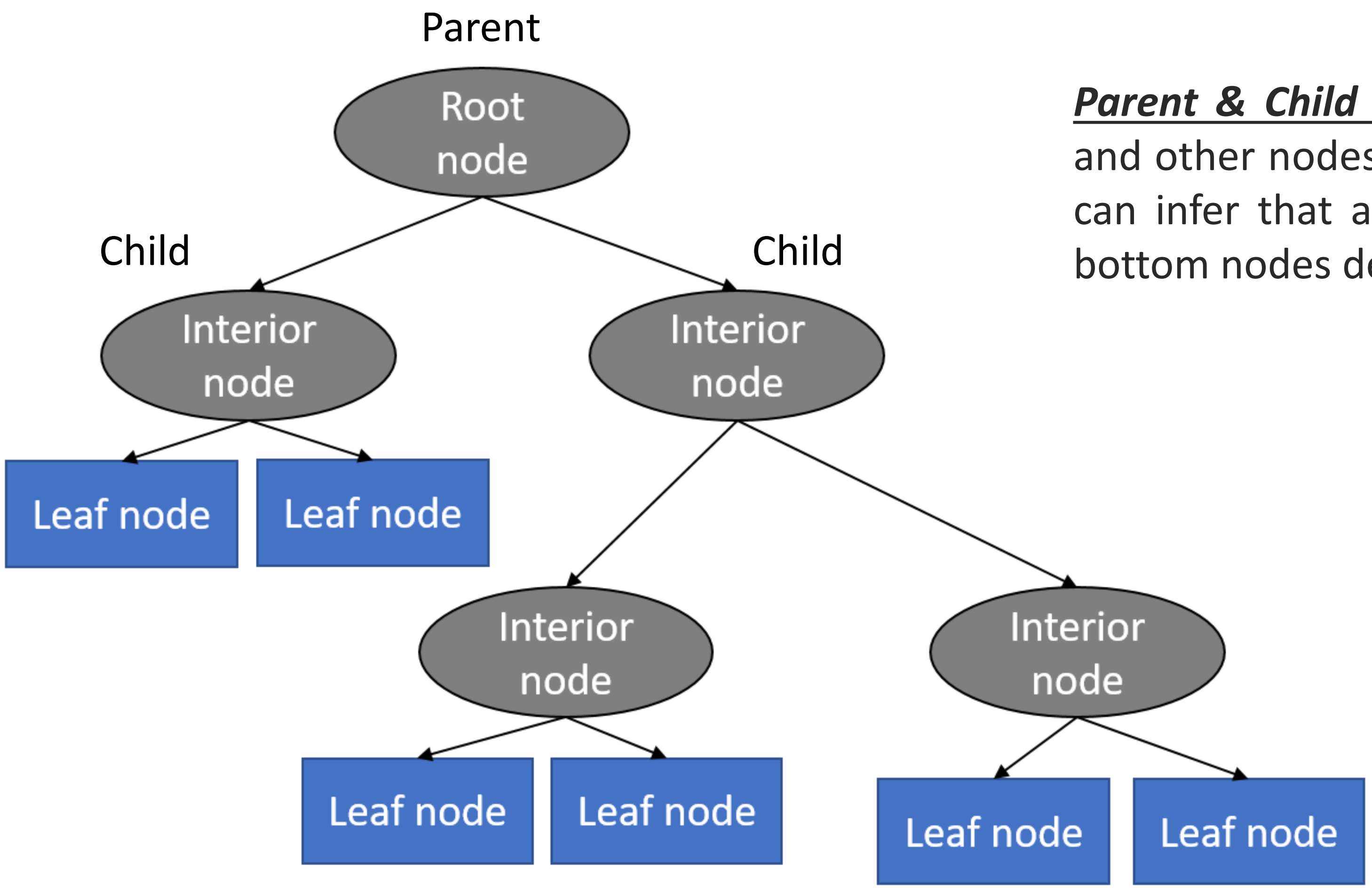
Definitions



Splitting → The process of dividing the root node into several different sub parts i.e. interior and leaf nodes (subtrees or branch Tree).

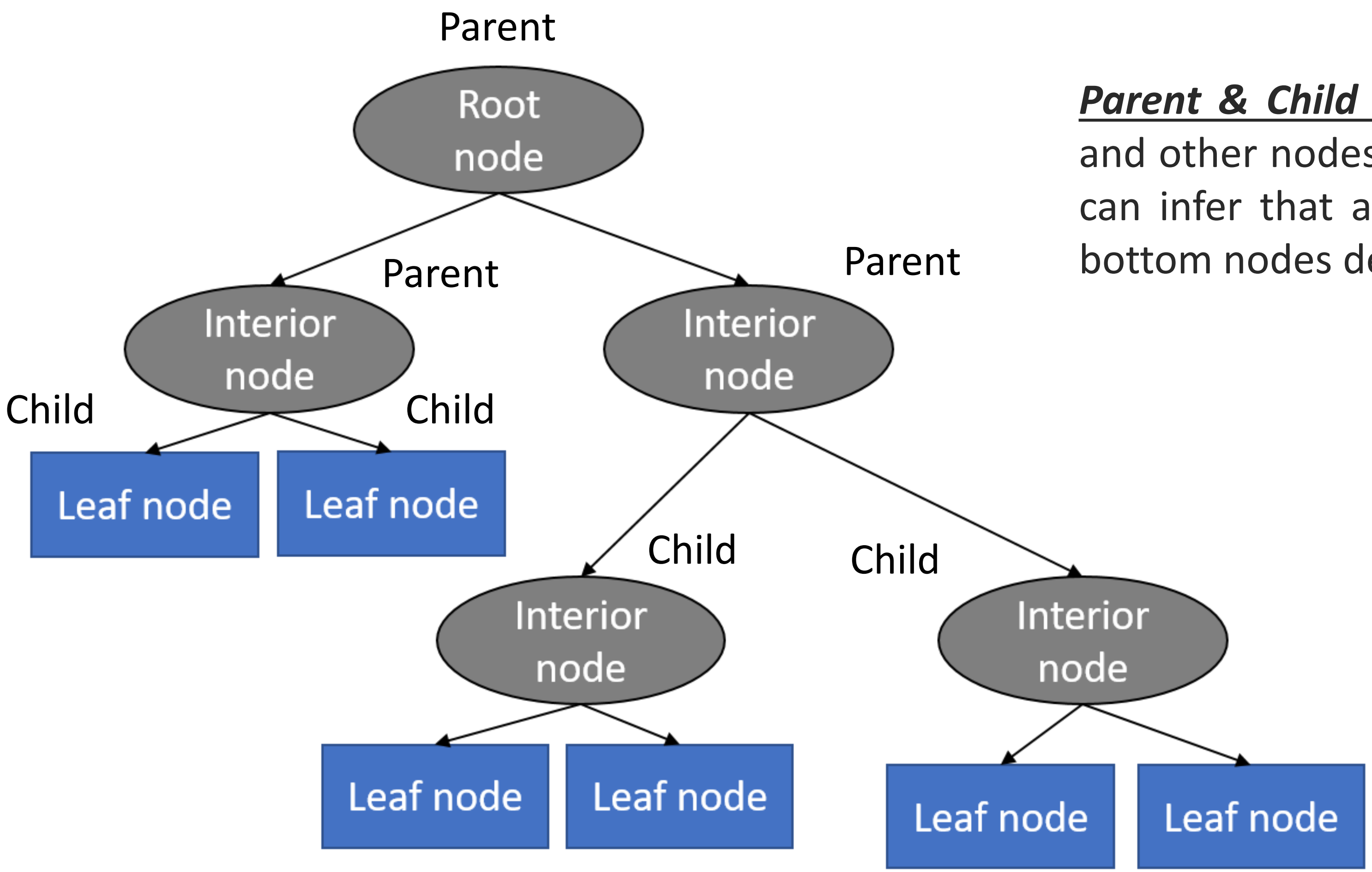
Branch Tree → The nodes that are generated when we split our tree .i.e when we separate out any particular section of our decision tree.

Definitions



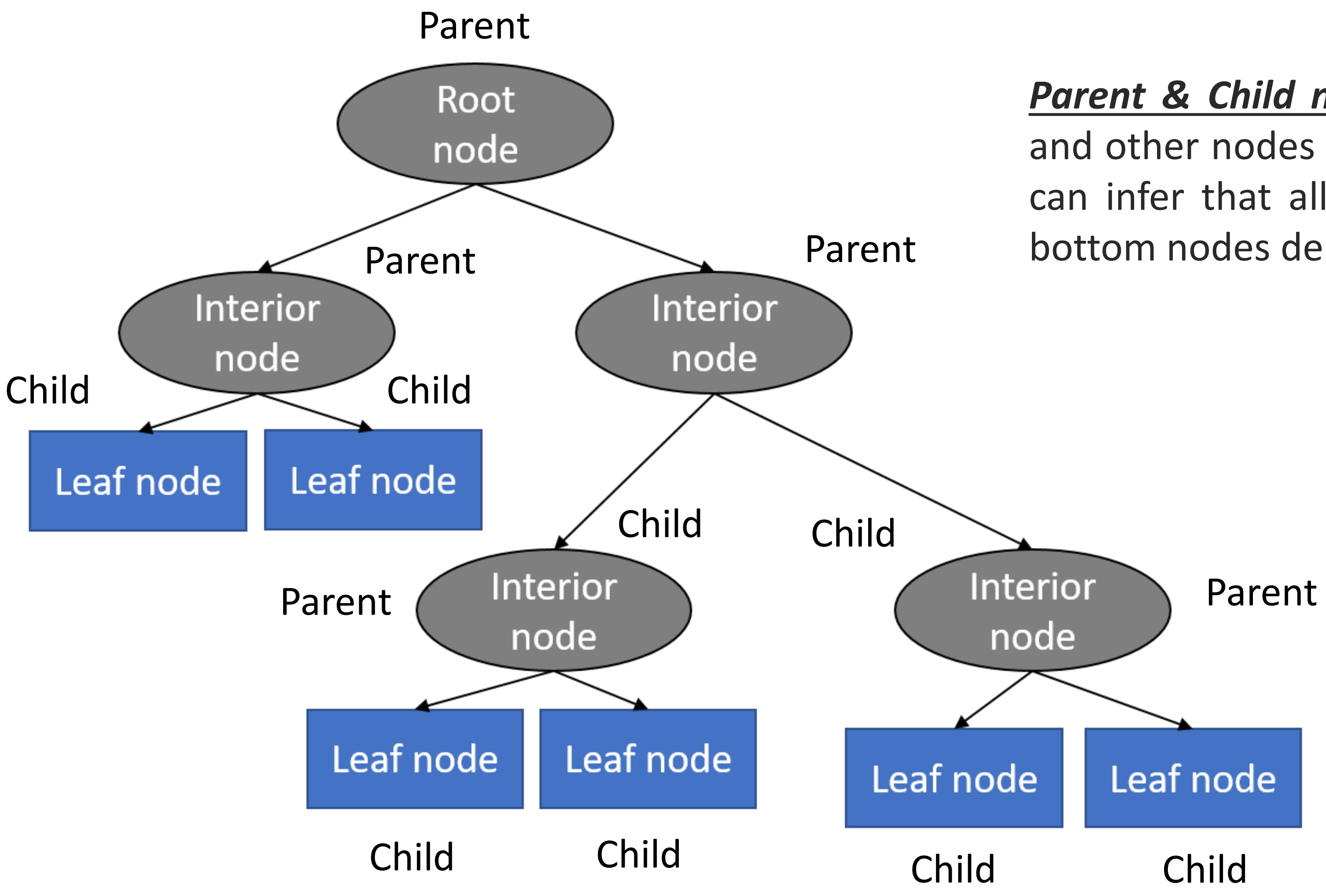
<u>Parent & Child node</u> → Root node is always the parent node and other nodes connected to it are all child node. Thereby, we can infer that all the top nodes are parent nodes and all the bottom nodes derived from the parent node are child nodes.

Definitions



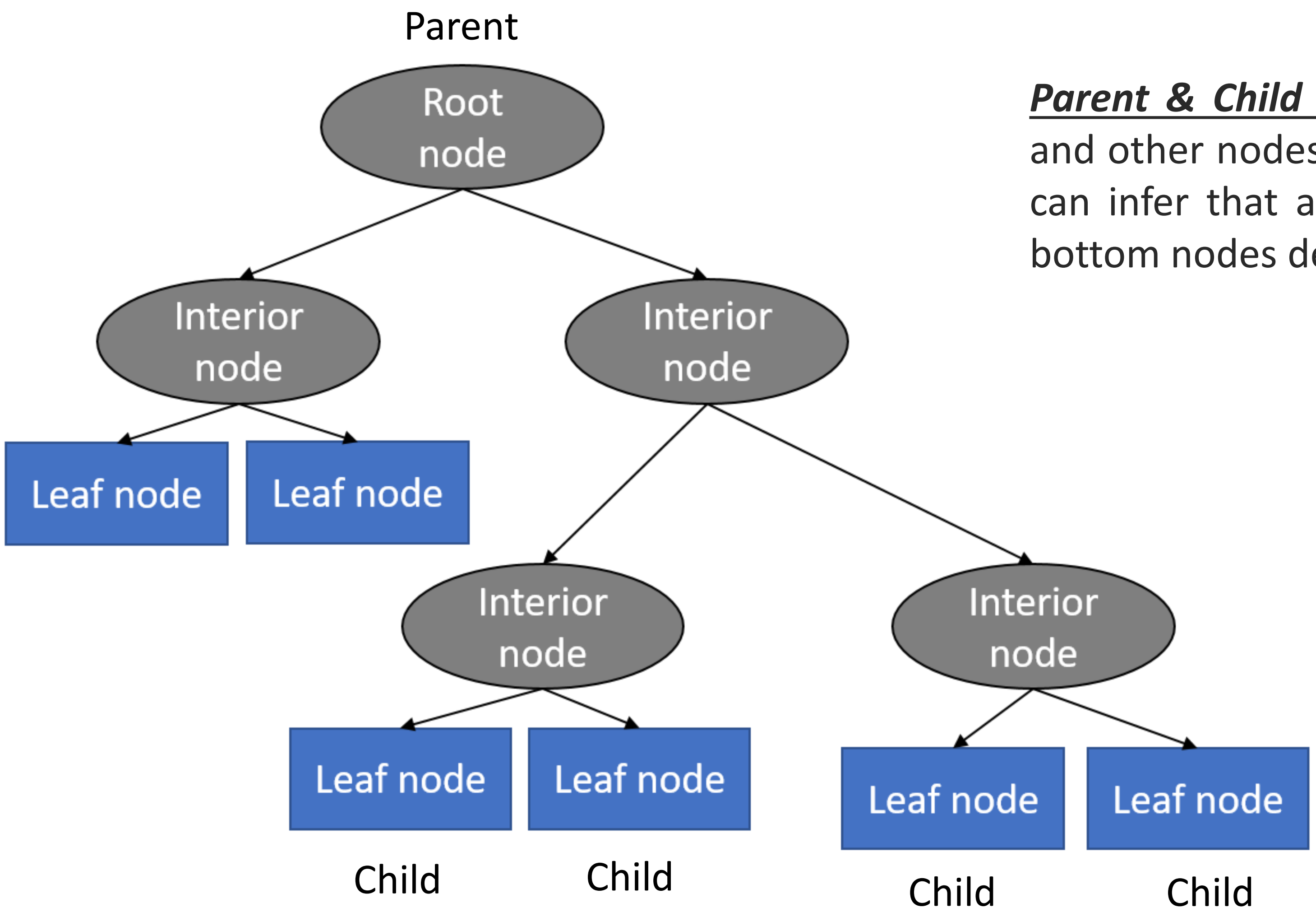
Parent & Child node → Root node is always the parent node and other nodes connected to it are all child node. Thereby, we can infer that all the top nodes are parent nodes and all the bottom nodes derived from the parent node are child nodes.

Definitions



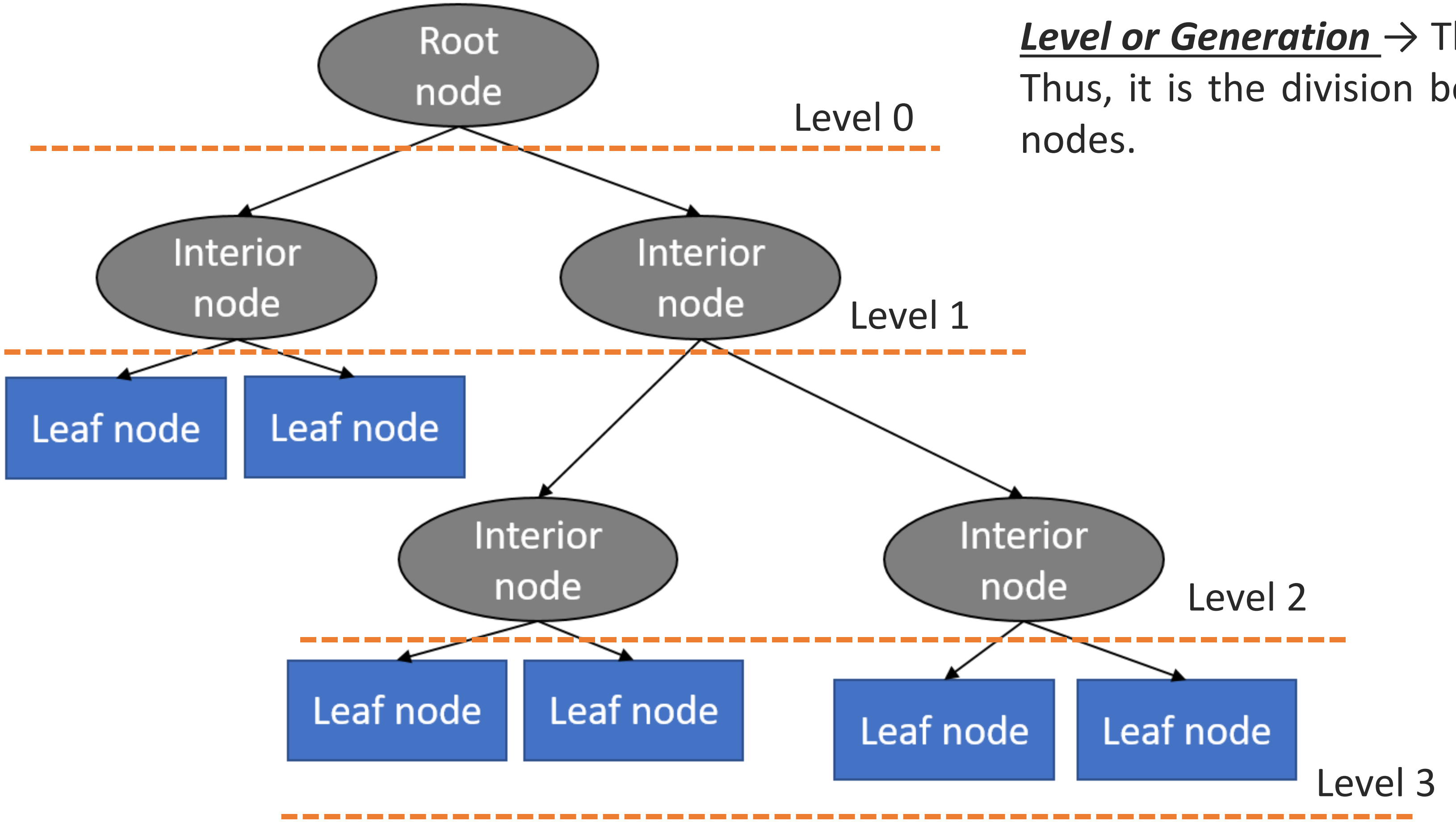
__Parent & Child node__ → Root node is always the parent node and other nodes connected to it are all child node. Thereby, we can infer that all the top nodes are parent nodes and all the bottom nodes derived from the parent node are child nodes.

Definitions



<u>Parent & Child node</u> → Root node is always the parent node and other nodes connected to it are all child node. Thereby, we can infer that all the top nodes are parent nodes and all the bottom nodes derived from the parent node are child nodes.

Definitions

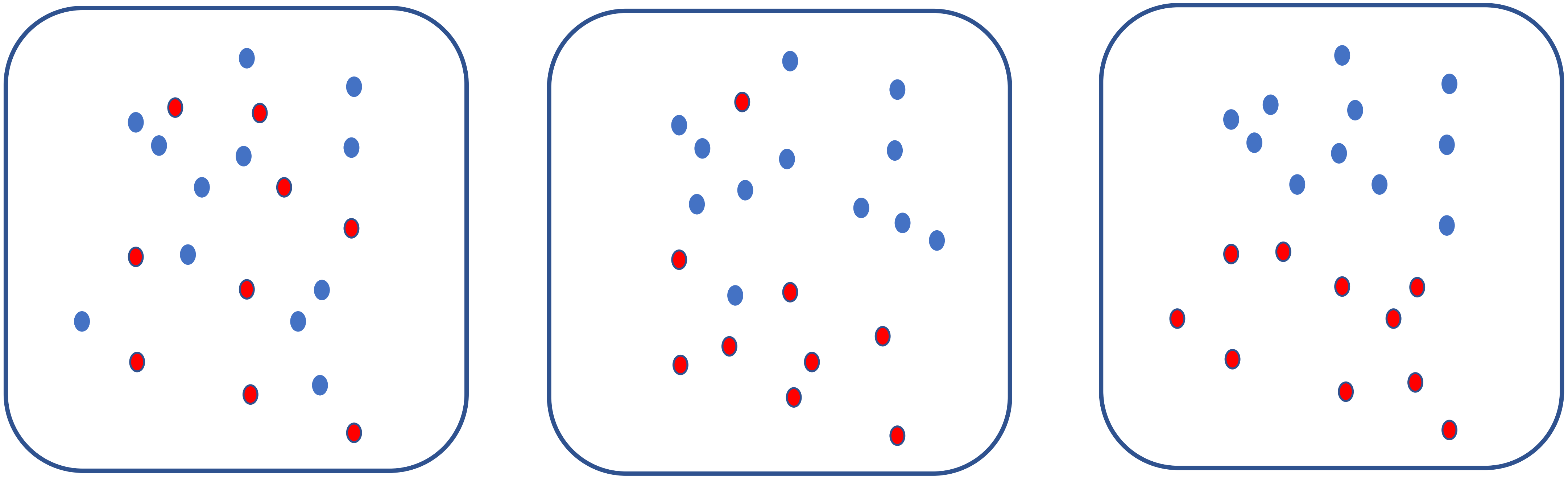


Level or Generation → They are the decision tree choice points. Thus, it is the division between the parent node and the child nodes.

Definitions

Entropy →

- It is an information theory metric that measures the impurity or uncertainty in a group of observations.
- It helps to decide the best attribute for start for start making decisions.
- It helps in telling the attribute with highest information gain.
- It is the presence of impurity (degree of randomness).



Very Impure   Less Impure   Minimum Impure

Definitions

Entropy →

- It is an information theory metric that measures the impurity or uncertainty in a group of observations.
- It helps to decide the best attribute for start for start making decisions.
- It helps in telling the attribute with highest information gain.
- It is the presence of impurity (degree of randomness).

Consider a dataset with N classes ➡

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

Pi is the probability of randomly selecting an example in class ***i***.

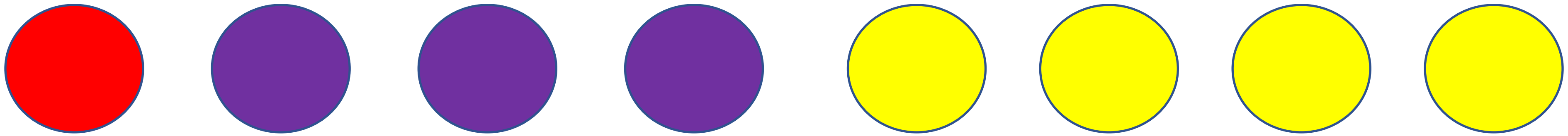
Definitions

Entropy

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

Pi is the probability of randomly selecting an example in class ***i***.

N = 3



Pr = 1/8

Pp = 3/8

Py = 4/8 or 1/2

$$E = -(\frac{1}{8}log_2(\frac{1}{8}) + \frac{3}{8}log_2(\frac{3}{8}) + \frac{4}{8}log_2(\frac{4}{8}))$$

0.42

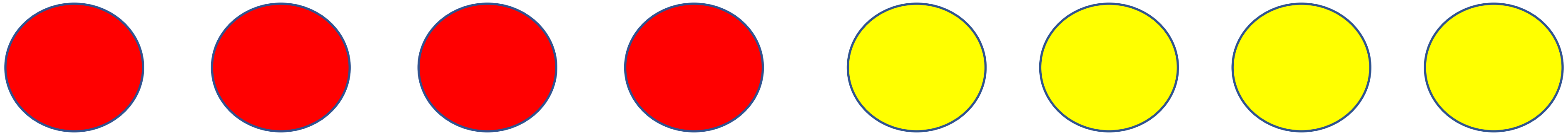
Definitions

Entropy

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

Pi is the probability of randomly selecting an example in class ***i***.

N = 2



Pr = 4/8 or 1/2

Py = 4/8 or 1/2

$$E = -((0.5log_2 0.5) + (0.5log_2 0.5))$$

1

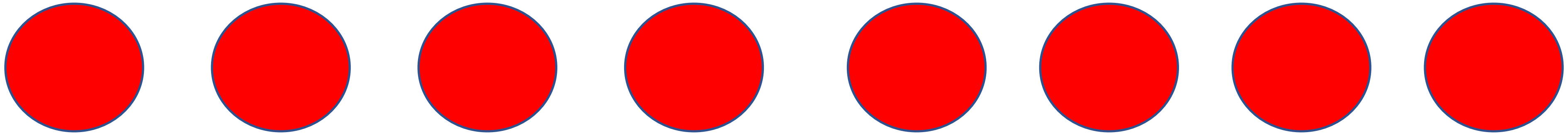
Definitions

Entropy

$$E = -\sum_{i=1}^{N} p_i log_2 p_i$$

Pi is the probability of randomly selecting an example in class ***i***.

N = 1



Pr = 8/8 or 1

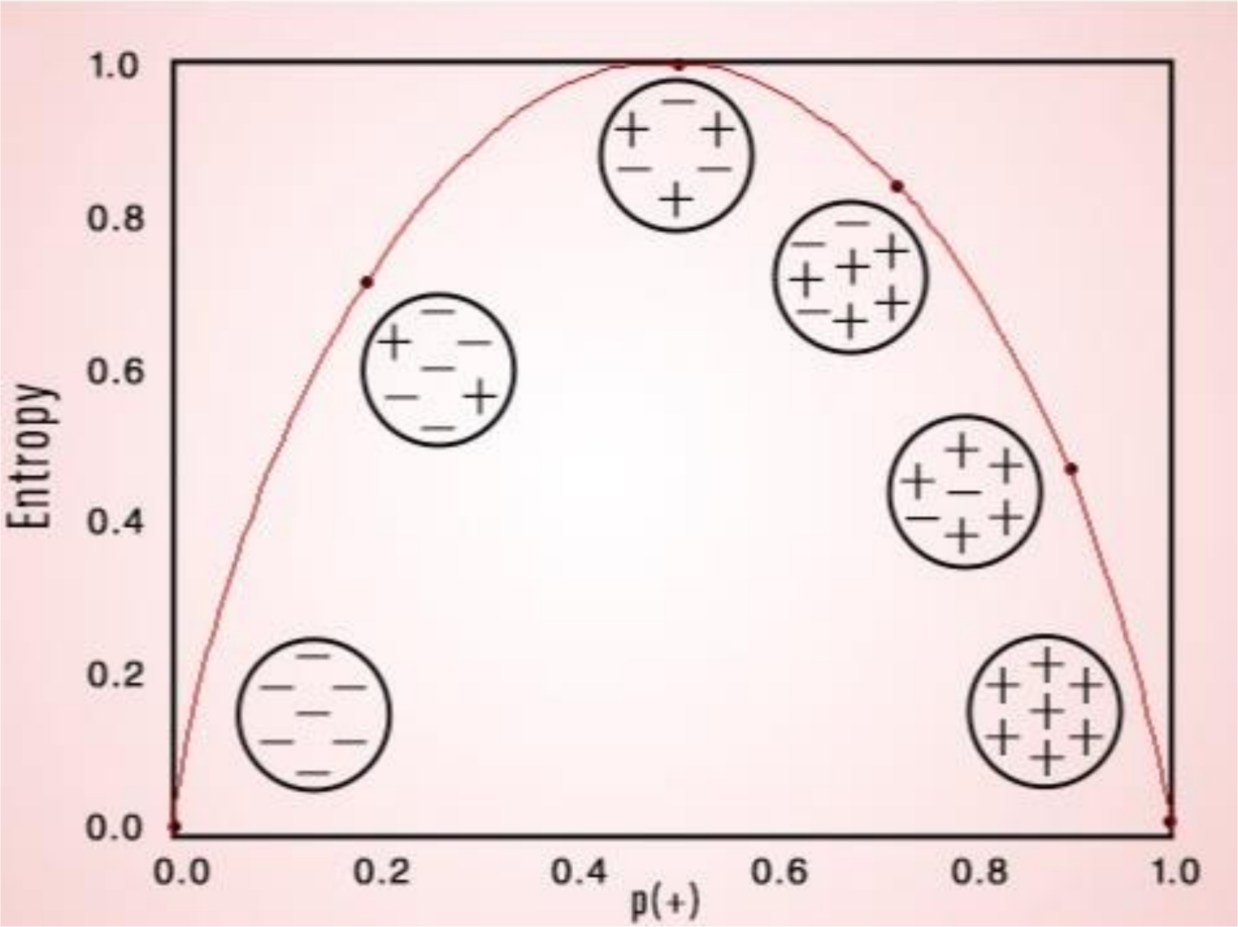
$$E = -(1log_2 1)$$

0

Definitions

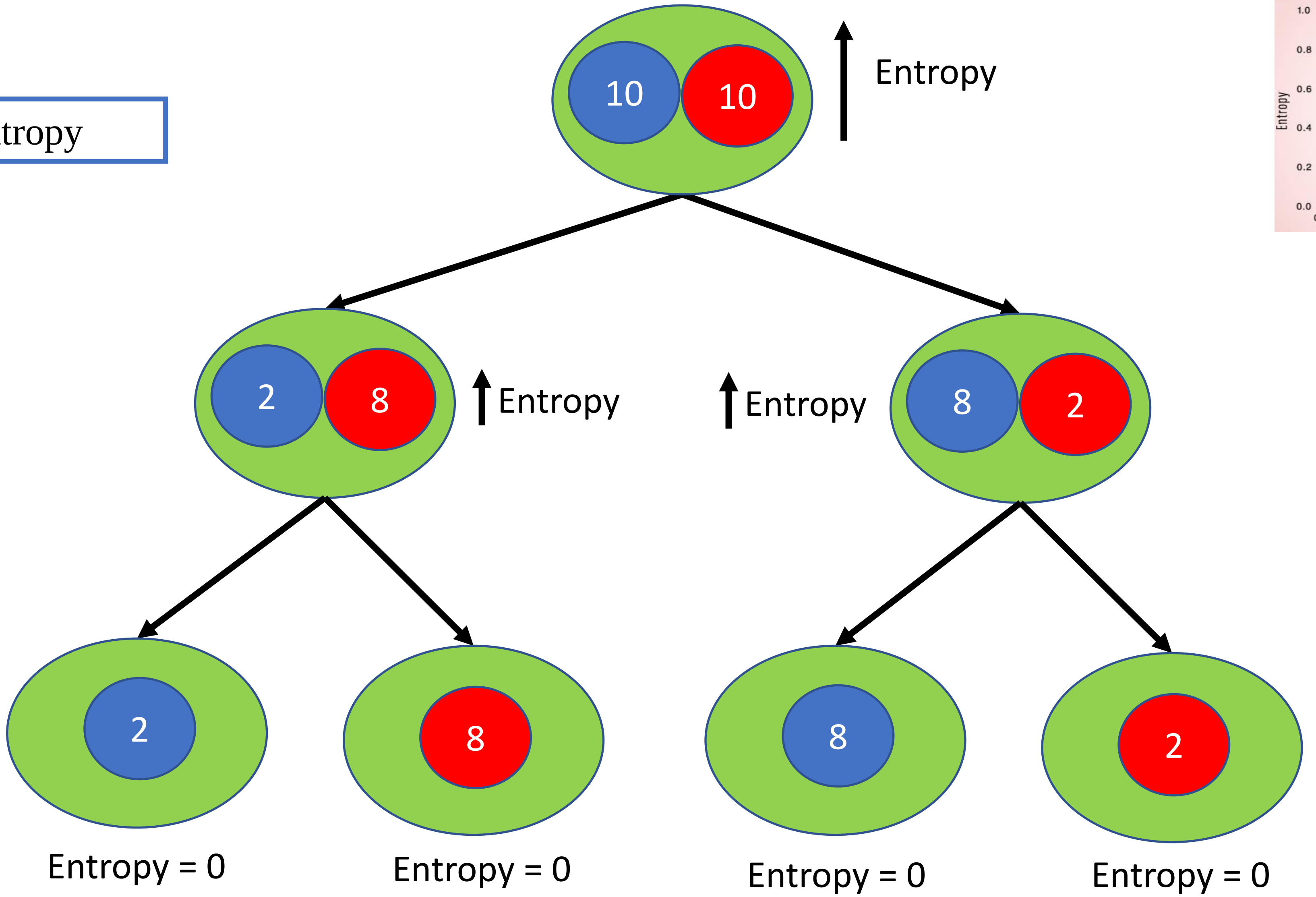
Entropy

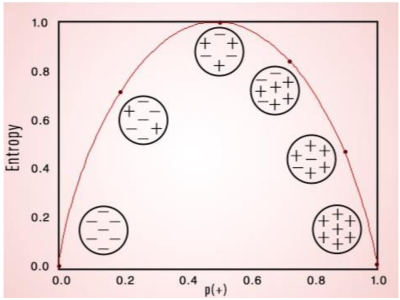
↑ Entropy ➡ ↑ Disorder



Definitions

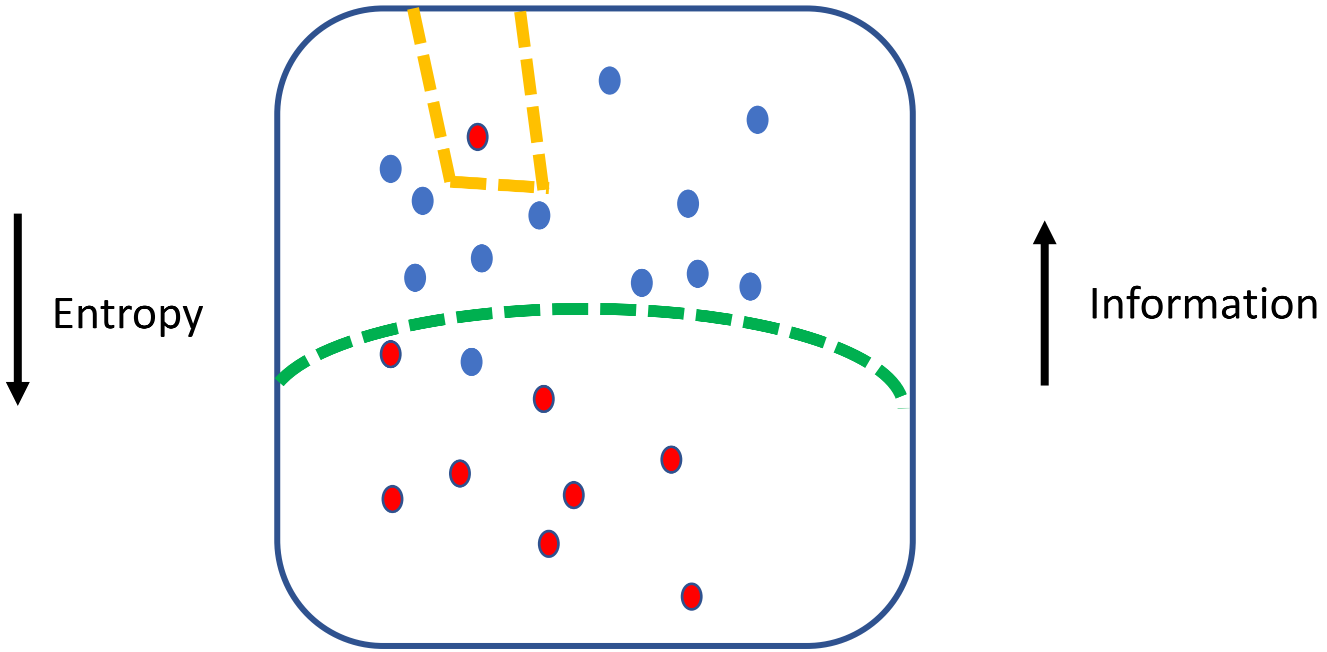
Entropy





Definitions

Entropy



Definitions

Information Gain →

- It define information gain as a measure of how much information a feature provides about a class.
- It is the decrease or reduction in entropy after a dataset is split on the basis of an attribute so that it helps to decide which attribute should be selected as the decision node.
- It helps to determine the order of attributes in the nodes of a decision tree.
- Constructing a Decision tree is all about finding the attribute that returns highest information gain.

$$Gain = E_{parent} - W * E_{children}$$

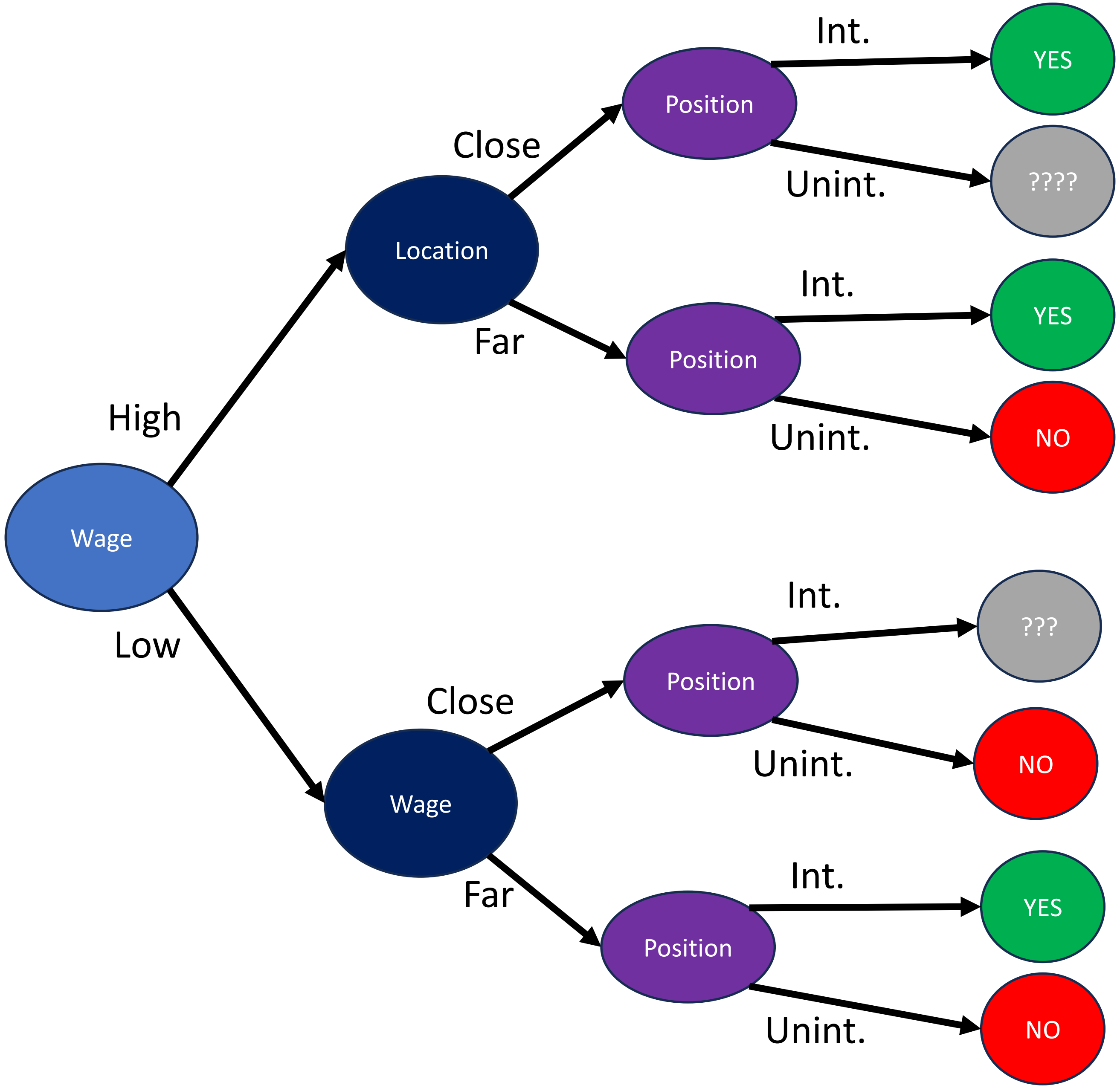
- $Gain$ represents information gain
- E_{parent} is the entropy of the parent node
- $E_{children}$ is the entropy of the child nodes
- W is the weight of the child nodes

Definitions

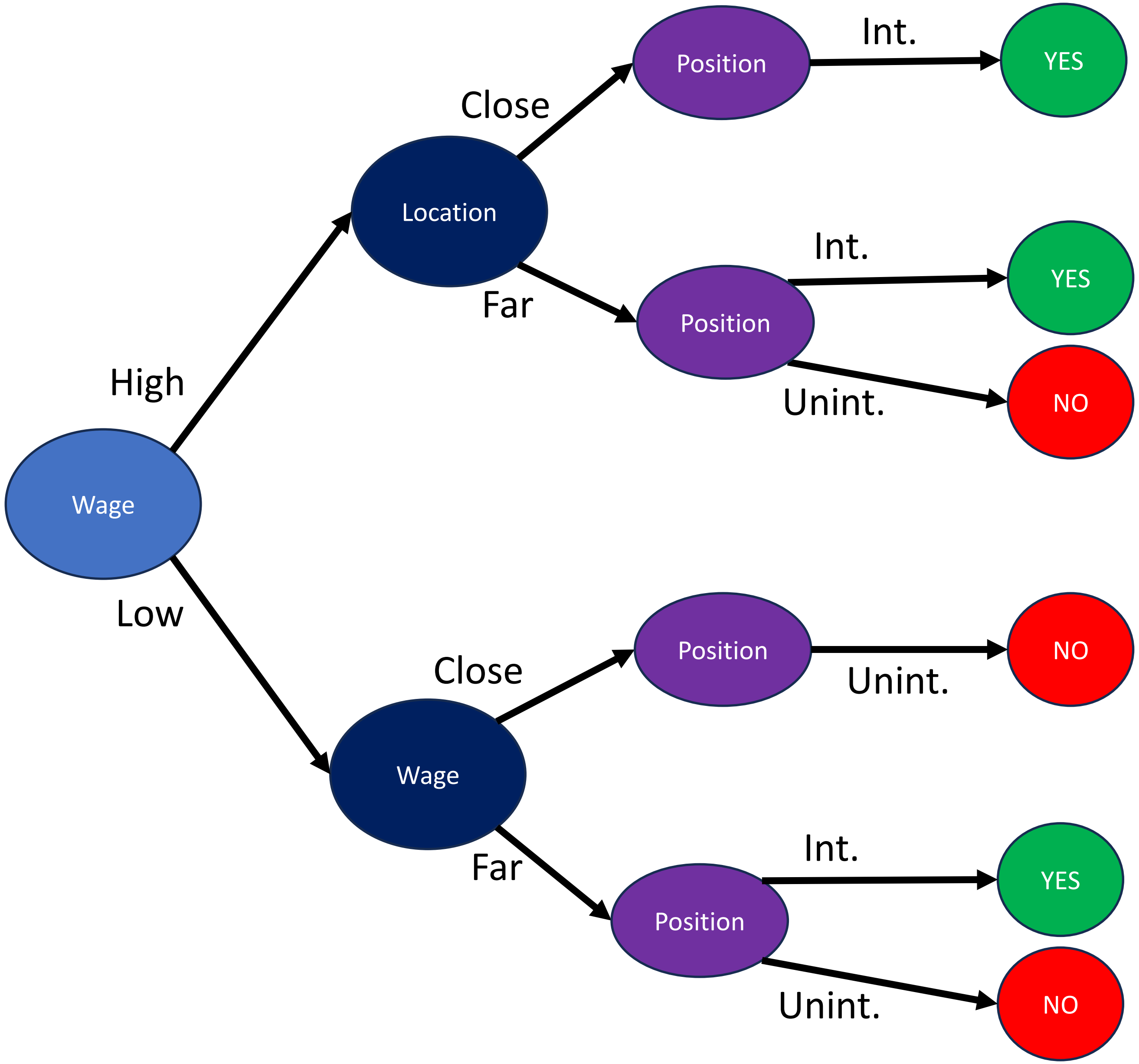
Information Gain

$$Gain = E_{parent} - W * E_{children}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO



Wage	Location	Position	Decision
High	Far	Int.	YES
Low	Close	Unint.	NO
Low	Far	Int.	YES
High	Far	Unint.	NO
High	Close	Int.	YES
Low	Far	Unint.	NO



Wage	Location	Position	Decision
High	Far	Int.	YES
Low	Close	Unint.	NO
Low	Far	Int.	YES
High	Far	Unint.	NO
High	Close	Int.	YES
Low	Far	Unint.	NO

Definitions

Information Gain

$$Gain = E_{parent} - W * E_{children}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

Interative Dichotomiser 3 – ID3

Dividir um conjunto de dados de forma a criar uma árvore que represente as decisões tomadas com base nas variáveis de entrada.

1. **Critério de divisão**: O ID3 escolhe o atributo que melhor separa os dados com base no conceito de **ganho de informação**. O algoritmo mede o quanto um atributo contribui para reduzir a incerteza (entropia) nos dados.

2. **Divisão recursiva**: Após selecionar o melhor atributo, o algoritmo divide os dados em subconjuntos com base nesse atributo e repete o processo para cada subconjunto.

3. **Condição de parada**: O algoritmo para de dividir os dados quando:
    - Todos os exemplos pertencem à mesma classe.
    - Não há mais atributos para dividir.

OBS: O ID3 é conhecido por ser eficiente, mas pode sofrer com o problema de **overfitting** em datasets pequenos ou ruidosos. O algoritmo foi a base para outras versões, como o **C4.5** e o **CART**.

Definitions

Information Gain

$$Gain = \boxed{E_{parent}} - W * E_{children}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

$P_y = 3/6 = 1/2$

$P_n = 3/6 = 1/2$

E_{parent} = -(1/2*log1/2+ 1/2*log1/2)

$\boxed{E_{parent} = 1}$

Definitions

Information Gain

$$Gain = E_{parent} - W * \boxed{E_{children}}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

Wage High = 3 cases (2 – YES + 1 – NO)
Wage Low = 3 cases (1 – YES + 2 – NO)

$Py_{Wage-High} = 2/3$

$Pn_{Wage-High} = 1/3$

$E_{Wage-High} =$ -(2/3*log2/3+ 1/3*log1/3)

$\boxed{E_{Wage-High} = 0.92}$

$Py_{Wage-Low} = 1/3$

$Pn_{Wage-Low} = 2/3$

$E_{Wage-Low} =$ -(1/3*log1/3+ 2/3*log2/3)

$\boxed{E_{Wage-Low} = 0.92}$

Definitions

Information Gain

$$Gain = E_{parent} - \boxed{W} * E_{children}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

$W_{Wage - High} = 3/6 = 1/2$

$W_{Wage - Low} = 3/6 = 1/2$

$E_{Wage - High} = 0.92$

$E_{Wage - Low} = 0.92$

$G_{ainWage} = 1 - (1/2*0.92 + 1/2*0.92)$

$$\boxed{Gain_{Wage} = 0.08}$$

Wage High = 3 cases (2 – YES + 1 – NO)
Wage Low = 3 cases (1 – YES + 2 – NO)

Definitions

Information Gain

$$Gain = E_{parent} - W * \boxed{E_{children}}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

Location Far = 4 cases (2 – YES + 2 – NO)
Location Close = 2 cases (1 – YES + 1 – NO)

$Py_{Loc\ -Far}$ = 2/4 = 1/2

$Pn_{Loc\ -Far}$ = 2/4 = 1/2

$E_{Loc\ -Far}$ = -(1/2*log1/2+ 1/2*log1/2)

$\boxed{E_{Loc\ -Far} = 1}$

$Py_{Loc\ -Close}$ = 1/2

$Pn_{loc\ -Close}$ = 1/2

$E_{Loc\ -Close}$ = -(1/2*log1/2+ 1/2*log1/2)

$\boxed{E_{Loc\ -Close} = 1}$

Definitions

Information Gain

$$Gain = E_{parent} - \boxed{W} * E_{children}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

$W_{Loc-Far}$ = 4/6 = 2/3

$W_{Loc-Close}$ = 2/6 = 1/3

$E_{Loc-Far} = 1$

$E_{Loc-Close} = 1$

$Gain_{Loc}$ = 1 – (2/3*1 + 1/3*1)

$\boxed{Gain_{Loc} = 0}$

Location Far = 4 cases (2 – YES + 2 – NO)

Location Close = 2 cases (1 – YES + 2 – NO)

Definitions

Information Gain

$$Gain = E_{parent} - W * E_{children}$$

Wage	Location	Position	Decision
High	Far	Interesting	YES
Low	Close	Uninteresting	NO
Low	Far	Interesting	YES
High	Far	Uninteresting	NO
High	Close	Interesting	YES
Low	Far	Uninteresting	NO

$W_{Pos-Int}$ = 3/6 = 1/2

$W_{Pos-Uint}$ = 3/6 = 1/2

$E_{Pos-Int} = 0$

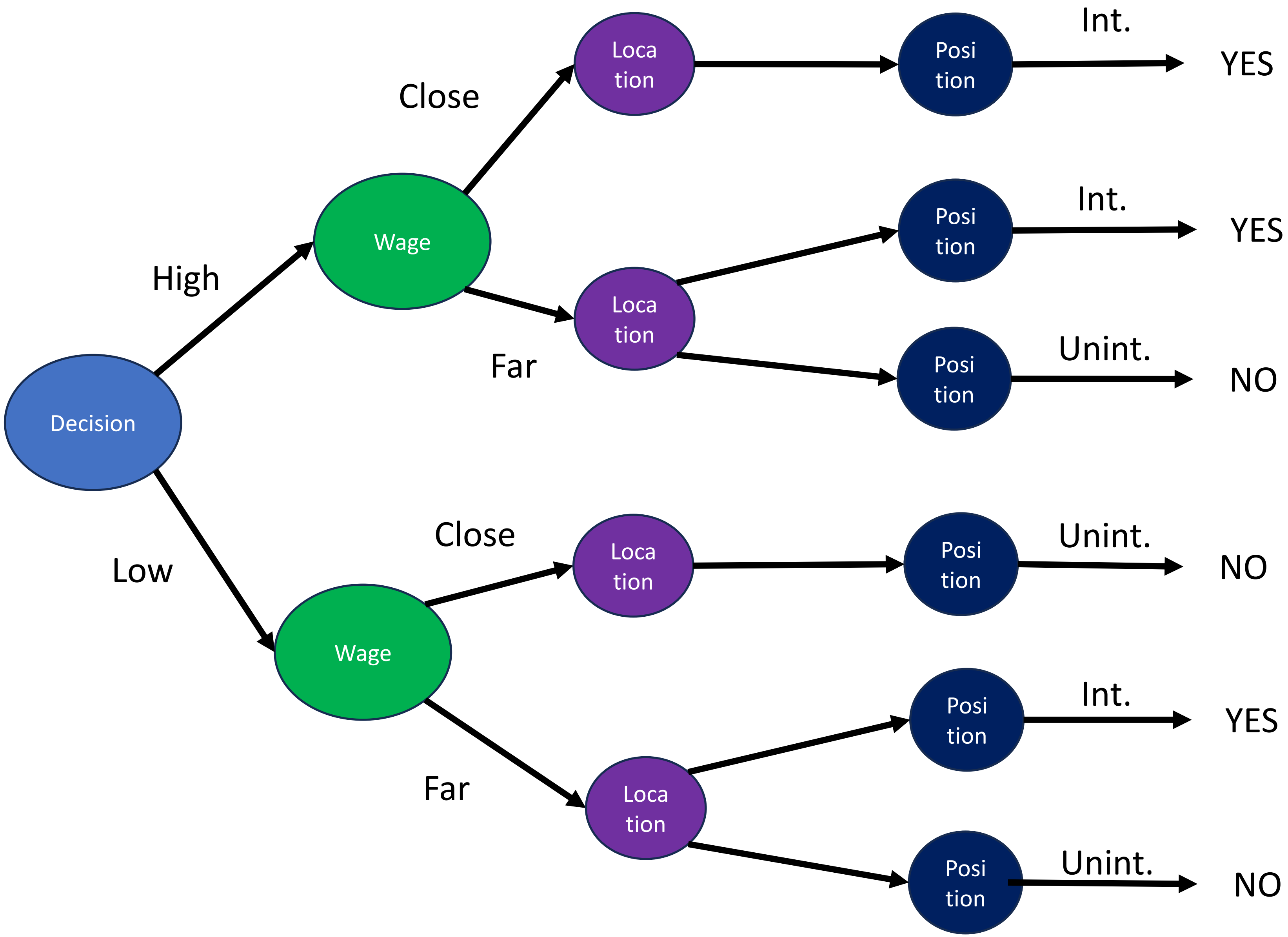
$E_{Pos-Uint} = 0$

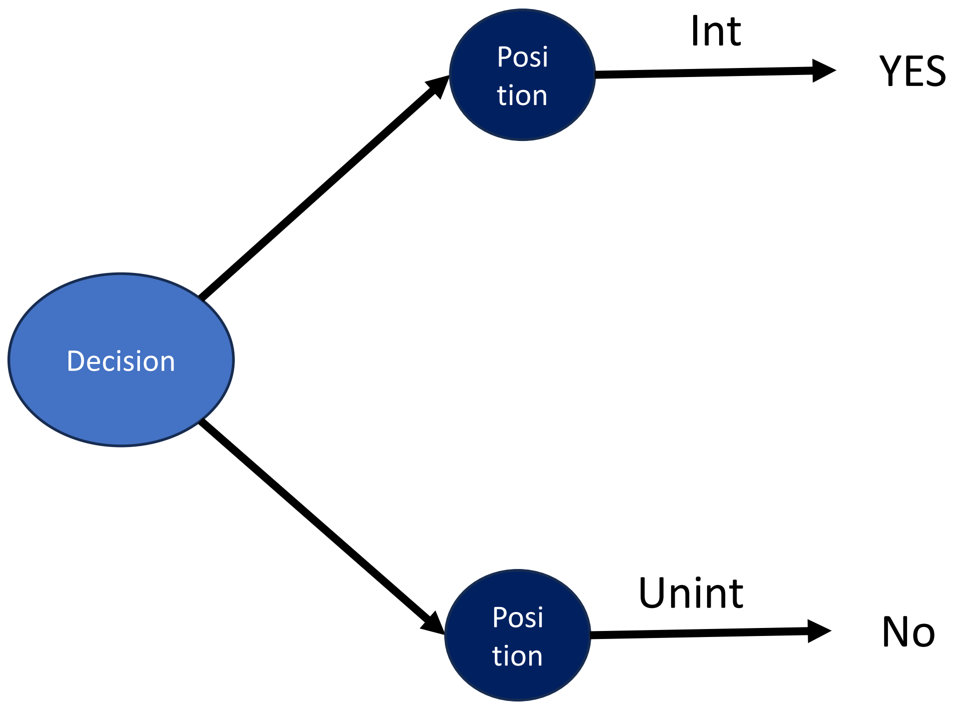
$Gain_{Pos}$ = 1 – (1/2*0 + 1/2*0)

$\boxed{Gain_{Pos} = 1}$

Position Int = 3 cases (3 – YES + 0 – NO)

Position Unint = 3 cases (0 – YES + 3 – NO)


Decision
High
Low
Wage
Wage
Close
Far
Close
Far
Loca tion
Loca tion
Loca tion
Loca tion
Posi tion
Posi tion
Posi tion
Posi tion
Posi tion
Posi tion
Int.
YES
Int.
YES
Unint.
NO
Unint.
NO
Int.
YES
Unint.
NO



Decision
Posi
tion
Int
YES
Posi
tion
Unint
No

Definitions

Gini Index →

- Gini impurity is a function that determines how well a decision tree was split.
- It helps to determine which splitter is best so that we can build a pure decision tree.
- Gini impurity ranges values from 0 to 0.5

Consider a dataset with N classes →

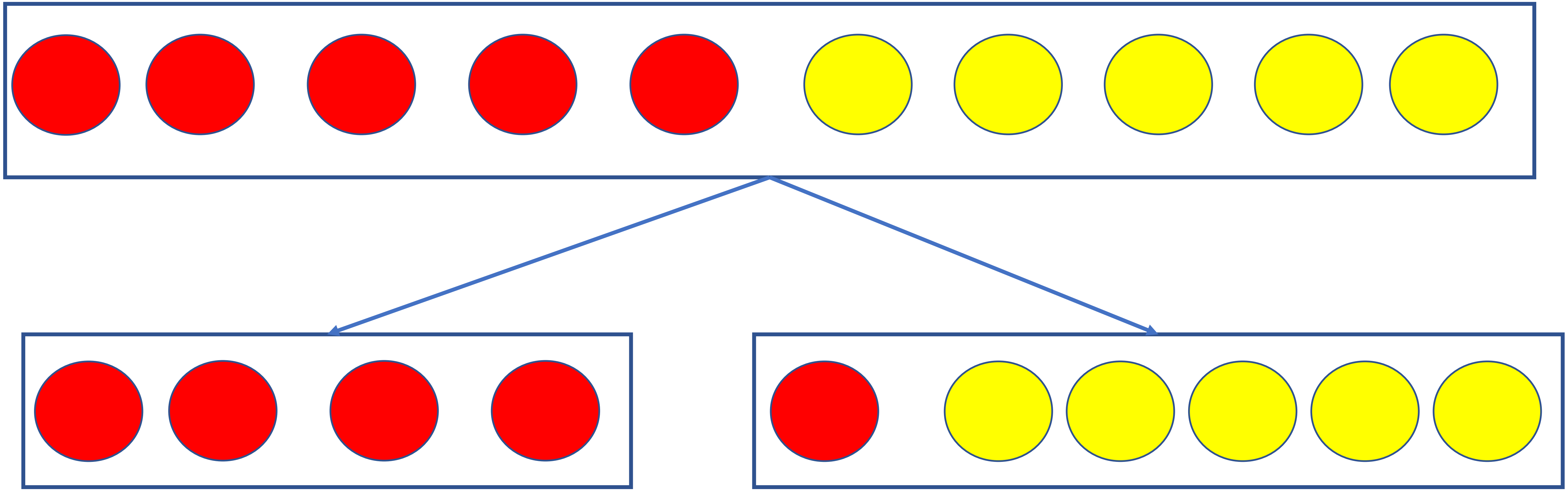
$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

Pi denotes the probability of an element being classified for a class ***i***.

Definitions

Gini Index

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$



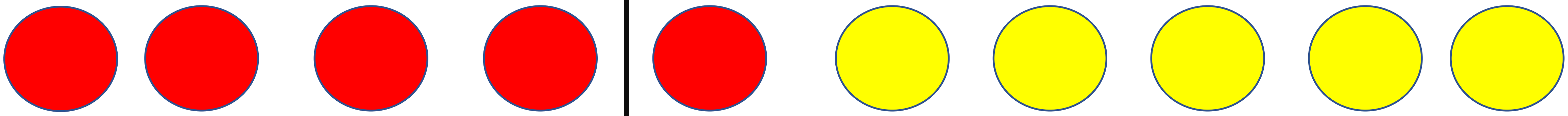
Definitions

Gini Index

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

N = 1

N = 2



Pr = 4/4 or 1

Pr = 1/6

Py = 5/6

$$Gleft = 1 - 1^2 = 0$$

$$Grigth = 1 - \left(\frac{1}{6}\right)^2 + \left(\frac{5}{6}\right)^2 = 0.278$$

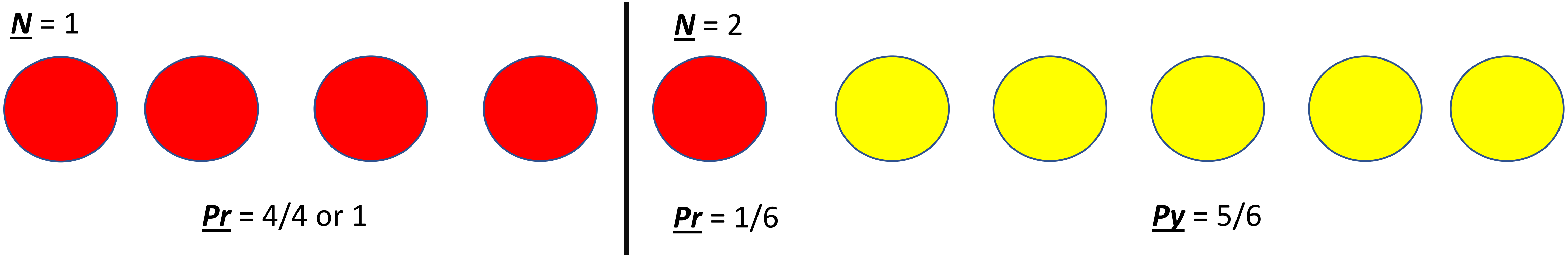
Definitions

Gini Index

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

N = 1

N = 2



Pr = 4/4 or 1

Pr = 1/6

Py = 5/6

$$Gimpurity = \left(\frac{4}{10}\right) * 0 + \left(\frac{6}{10}\right) * 0.278 = 0.167$$

$$\boxed{Gain = 0{,}5 - 0{,}167 = 0{,}333}$$

Definitions

Gini Index

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

N = 1

Pr = 5/5 or 1

N = 1

Py = 5/5 or 1

$$G_{left} = 1 - 1^2 = 0$$

$$G_{rigth} = 1 - 1^2 = 0$$

$$G_{impurity} = 1 * 0 + 1 * 0 = 0$$

$$\boxed{Gain = 0{,}5 - 0 = 0{,}5}$$

Gini Index x Information Gain

- The Gini Index facilitates the bigger distributions so easy to implement whereas the Information Gain favors lesser distributions having small count with multiple specific values.

- The method of the Gini Index is used by CART (Classification and Regression Tree) algorithms, in contrast to it, Information Gain is used in ID3, C4.5 algorithms.

- Gini index operates on the categorical target variables in terms of “success” or “failure” and performs only binary split, in opposite to that Information Gain computes the difference between entropy before and after the split and indicates the impurity in classes of elements.

Example

Dataset

sepal.length	***sepal.width***	***petal.length***	***petal.width***	***CLASS***
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
7	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
5.5	2.3	4	1.3	Versicolor
6.5	2.8	4.6	1.5	Versicolor
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3	5.9	2.1	Virginica
6.3	2.9	5.6	1.8	Virginica
6.5	3	5.8	2.2	Virginica

Example

Dataset

sepal.length	*sepal.width*	*petal.length*	*petal.width*	*CLASS*
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
7	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
5.5	2.3	4	1.3	Versicolor
6.5	2.8	4.6	1.5	Versicolor
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3	5.9	2.1	Virginica
6.3	2.9	5.6	1.8	Virginica
6.5	3	5.8	2.2	Virginica

Features

sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

Example

Dataset

sepal.length	*sepal.width*	*petal.length*	*petal.width*	*CLASS*
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5	3.6	1.4	0.2	Setosa
7	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
5.5	2.3	4	1.3	Versicolor
6.5	2.8	4.6	1.5	Versicolor
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3	5.9	2.1	Virginica
6.3	2.9	5.6	1.8	Virginica
6.5	3	5.8	2.2	Virginica

Features

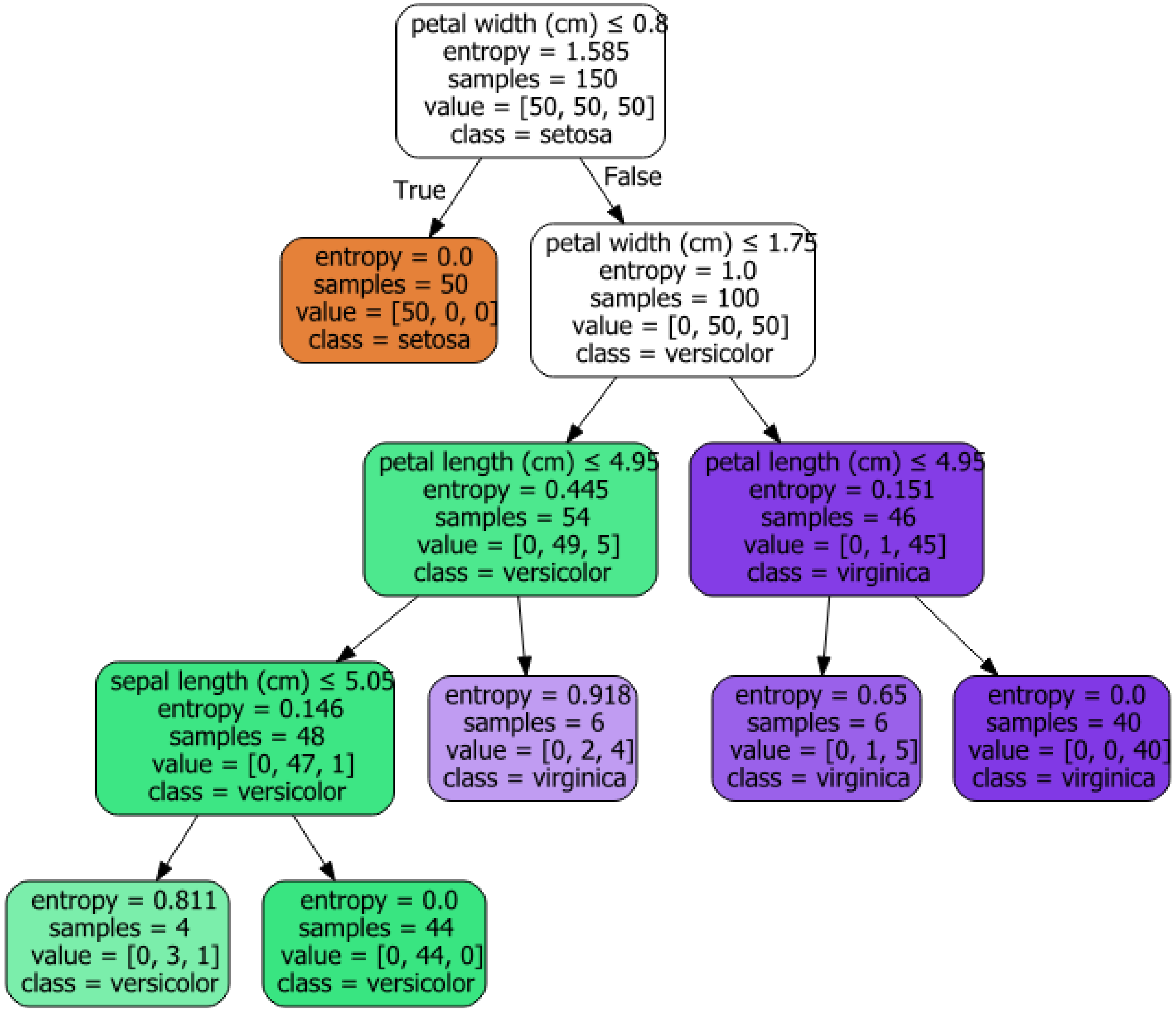
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

Classes

Setosa
Versicolor
Virginica

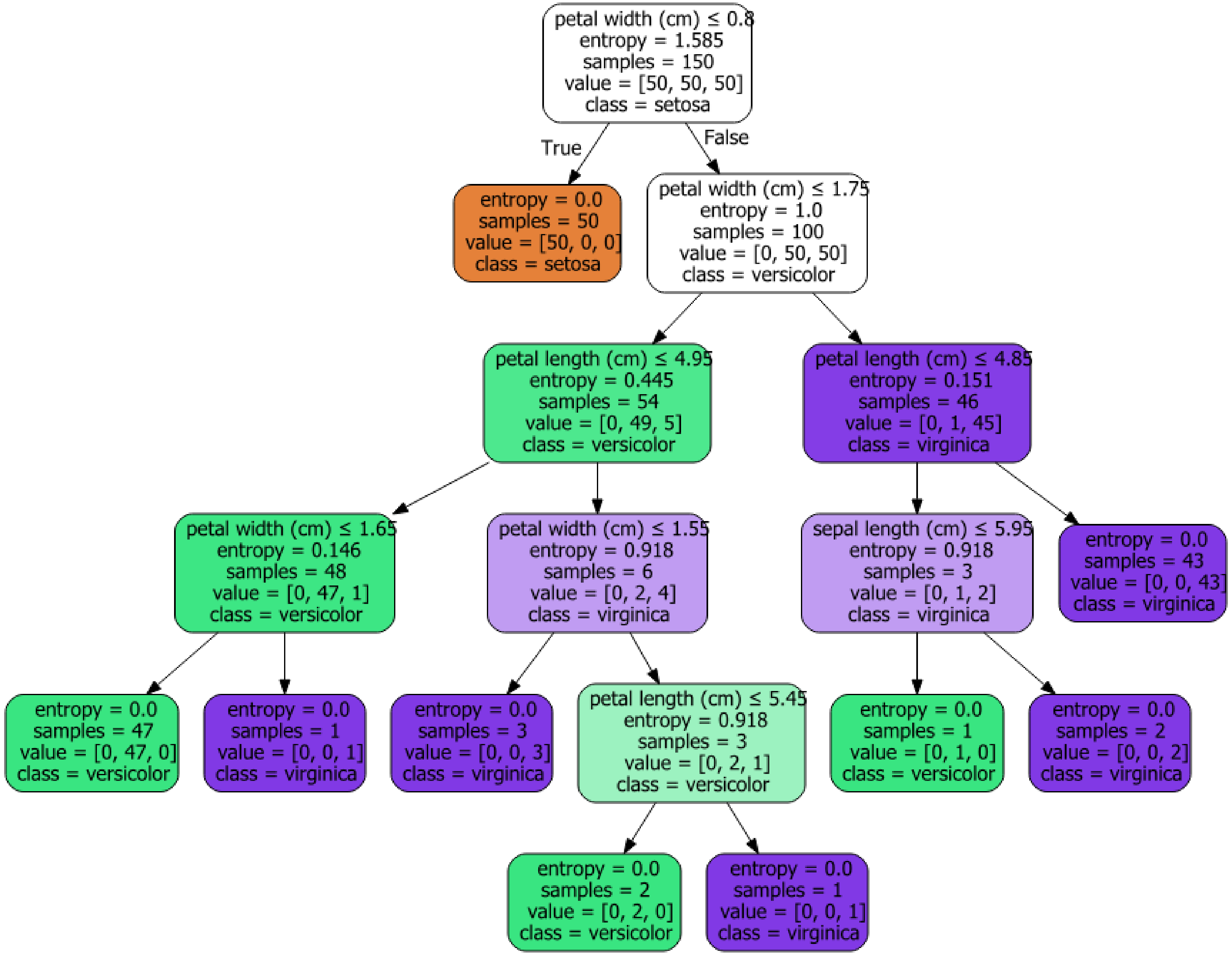
Example

Decision Tree



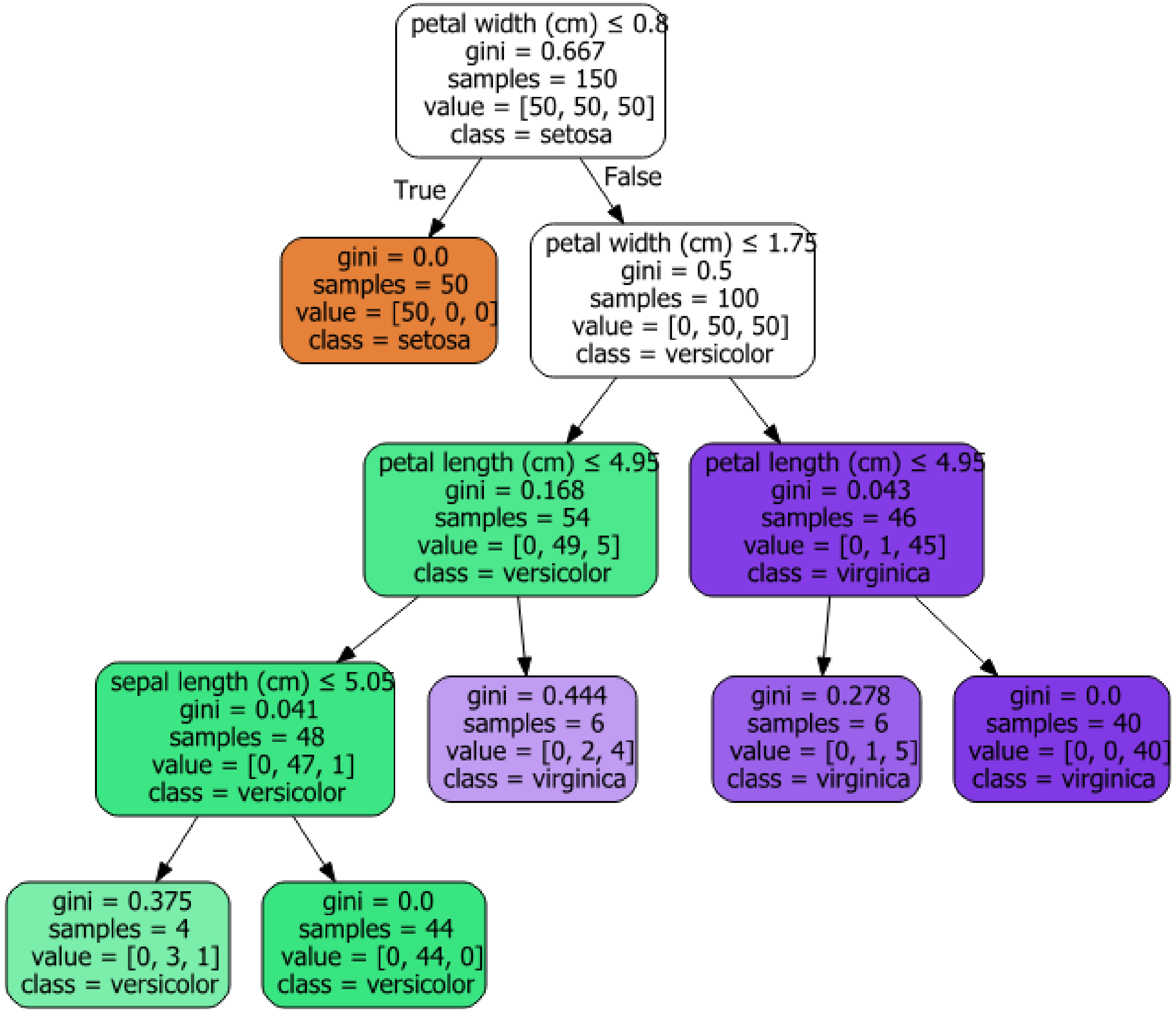
Example

Decision Tree



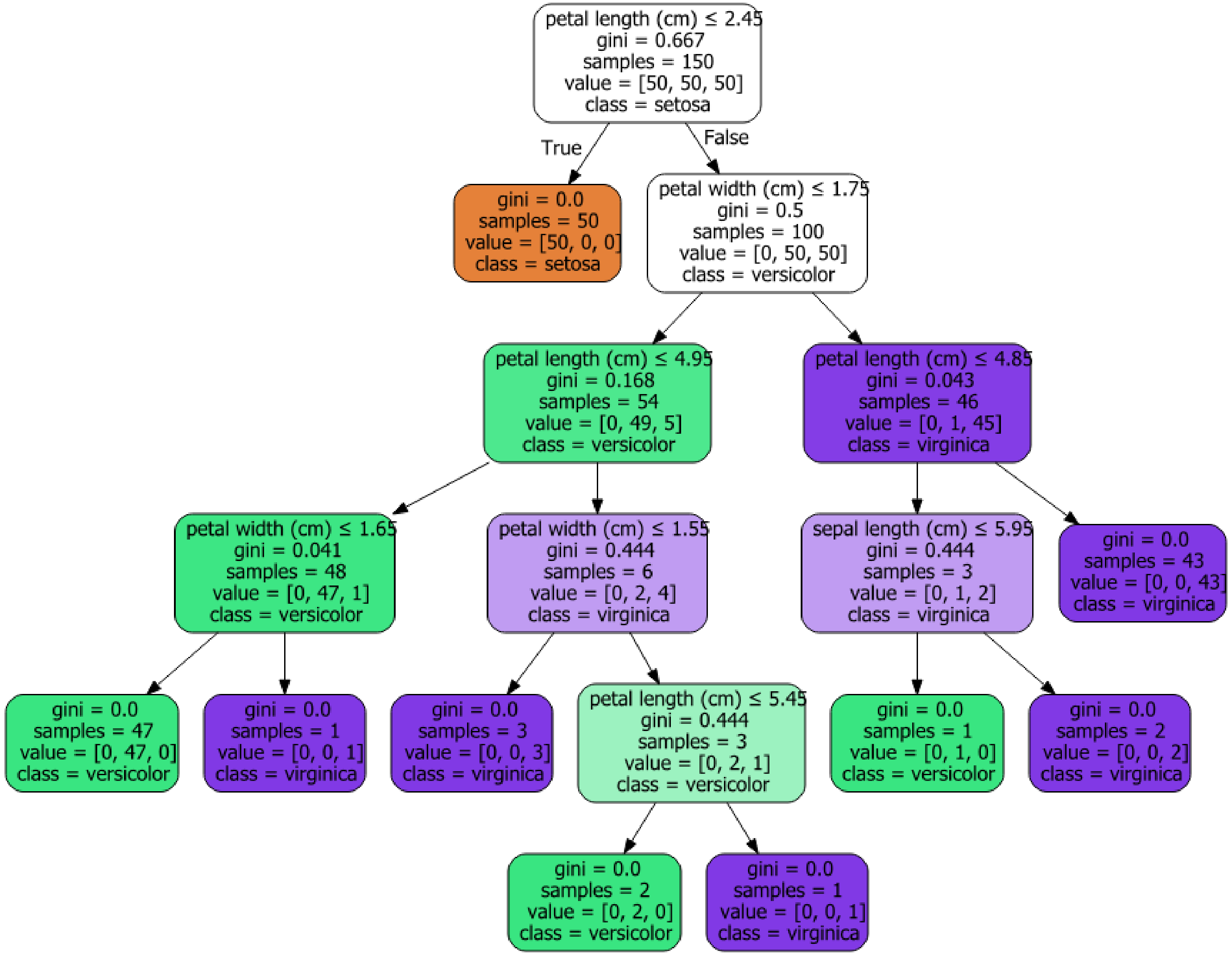
Example

Decision Tree



Example

Decision Tree



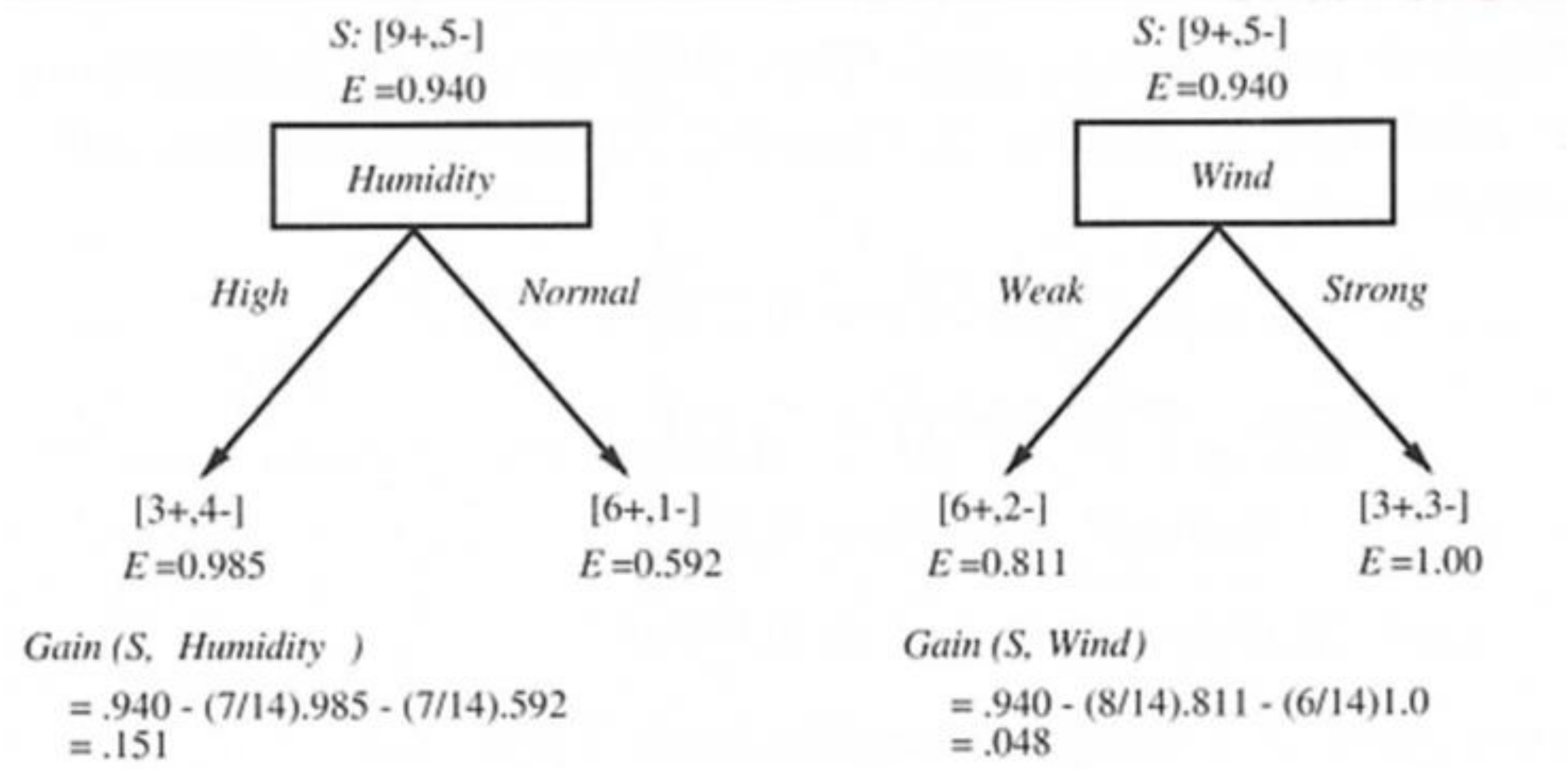
Example

ID3

	outlook	*temp*	*humidity*	*windy*	*play*
1	sunny	hot	high	False	no
2	sunny	hot	high	True	no
3	overcast	hot	high	False	yes
4	rainy	mild	high	False	yes
5	rainy	cool	normal	False	yes
6	rainy	cool	normal	True	no
7	overcast	cool	normal	True	yes
8	sunny	mild	high	False	no
9	sunny	cool	normal	False	yes
10	rainy	mild	normal	False	yes
11	sunny	mild	normal	True	yes
12	overcast	mild	high	True	yes
13	overcast	hot	normal	False	yes
14	rainy	mild	high	True	no

Example

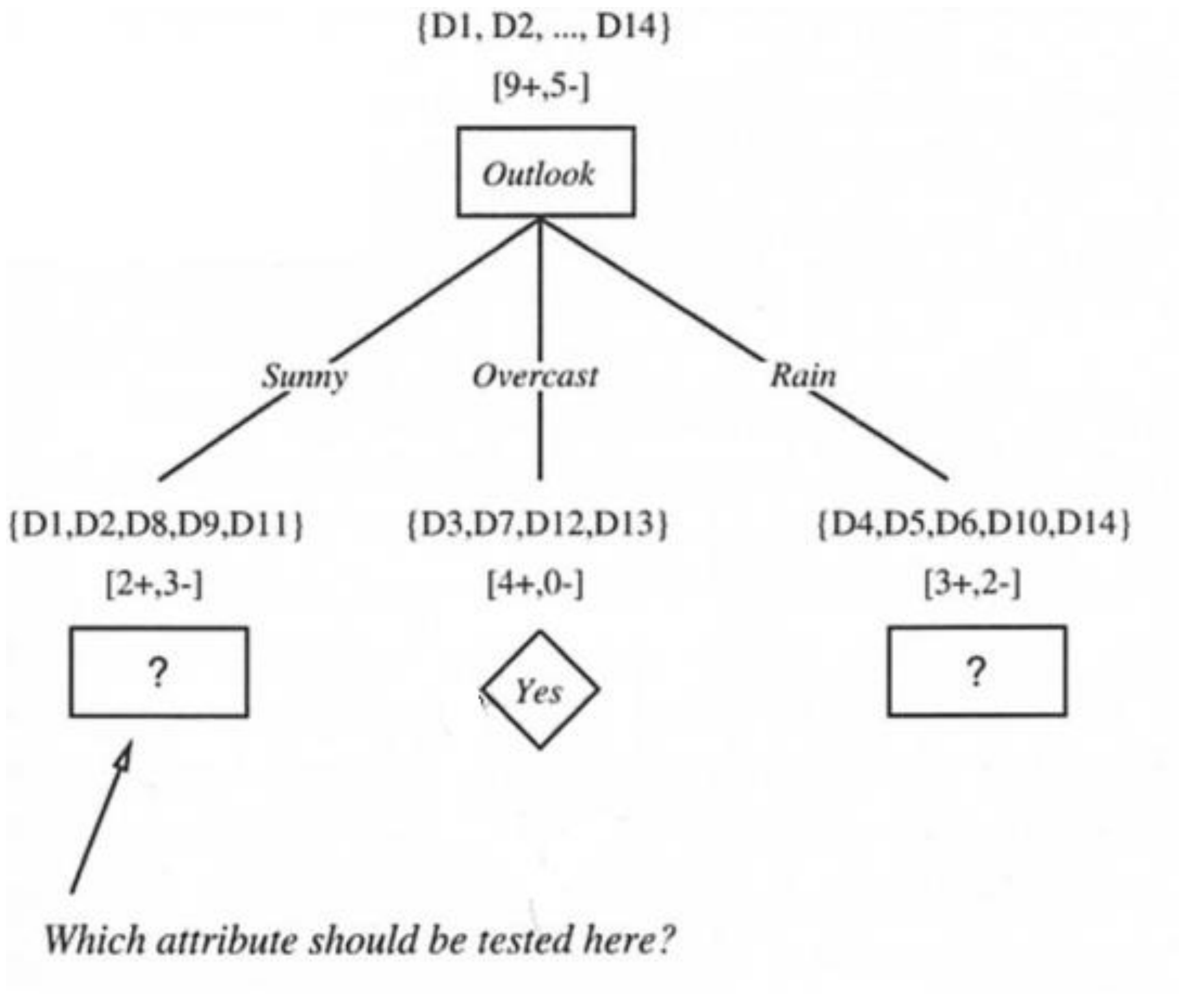
ID3


S: [9+,5-]
E =0.940
Humidity
High
Normal
[3+,4-]
E =0.985
[6+,1-]
E =0.592
Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151
S: [9+,5-]
E =0.940
Wind
Weak
Strong
[6+,2-]
E =0.811
[3+,3-]
E =1.00
Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

Example

ID3

– Gain (S, Outlook) = 0.246
– Gain (S, Humidity) = 0.151
– Gain (S, Wind) = 0.048
– Gain (S, Temperature) = 0.029



Example

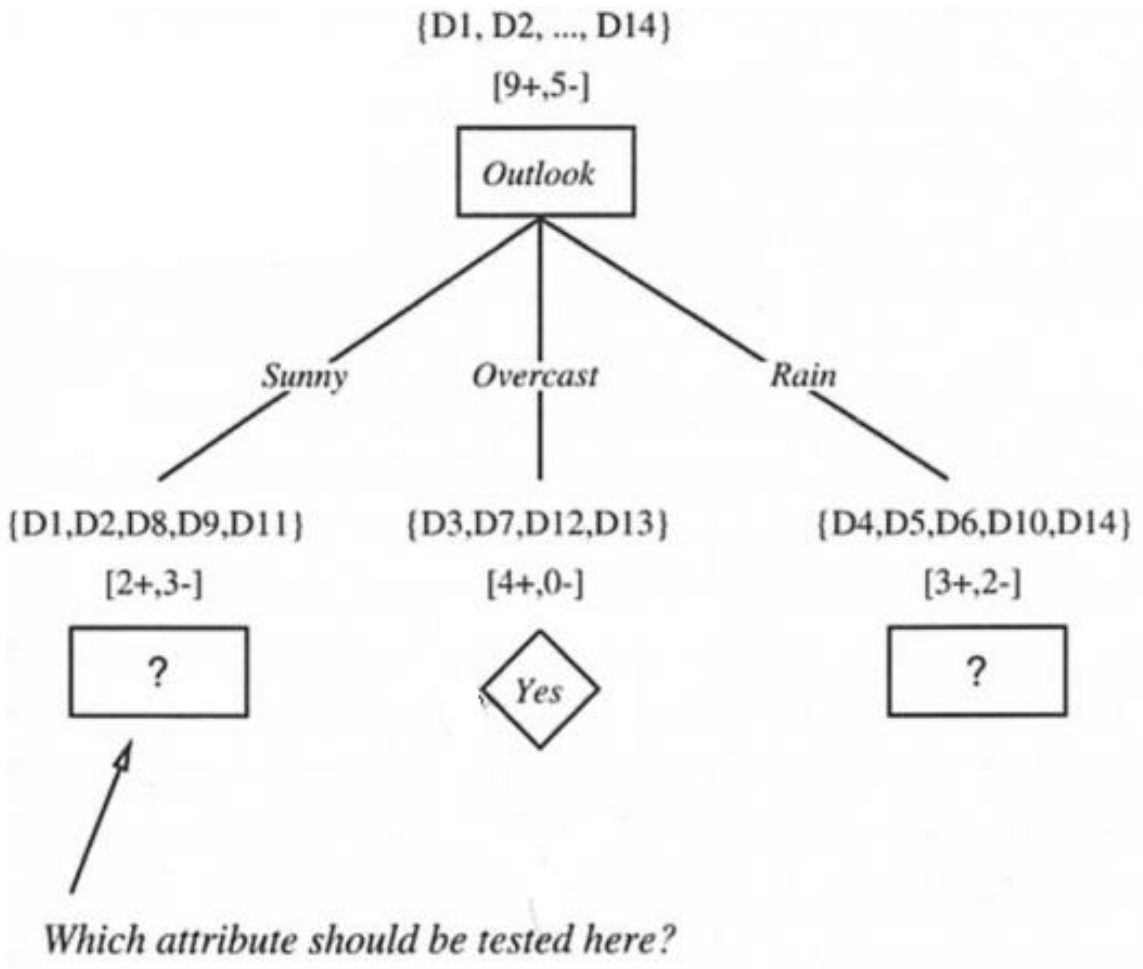
ID3

$S_{sunny} = \{D1, D2, D8, D9, D11\}$

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 + .970$

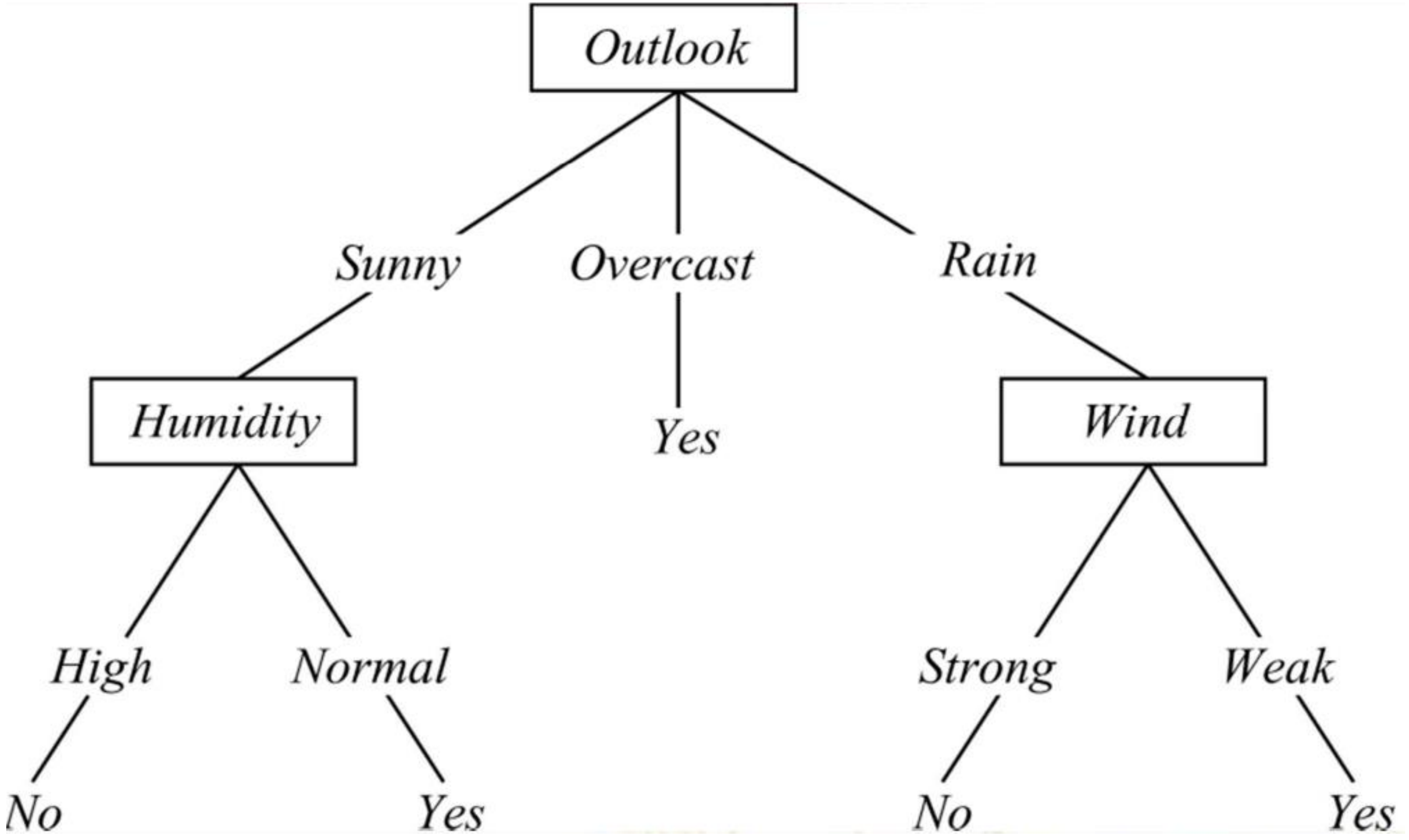
$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$



Example

ID3


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

Daniel Nogueira

dnogueira@ipca.pt