










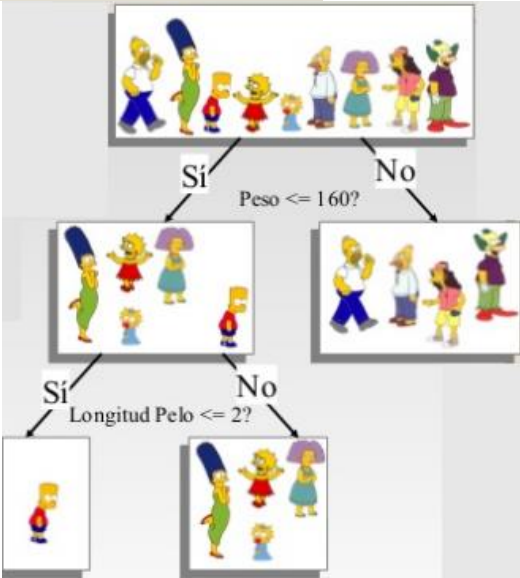
6. Decision trees

A **decision tree** model is a supervised learning technique that can be applied to solve regression and classification problems. The input values can be discrete or continuous. The name of this technique is because it can be represented as a tree structure with nodes that represent conditional instructions for categorizing the data.

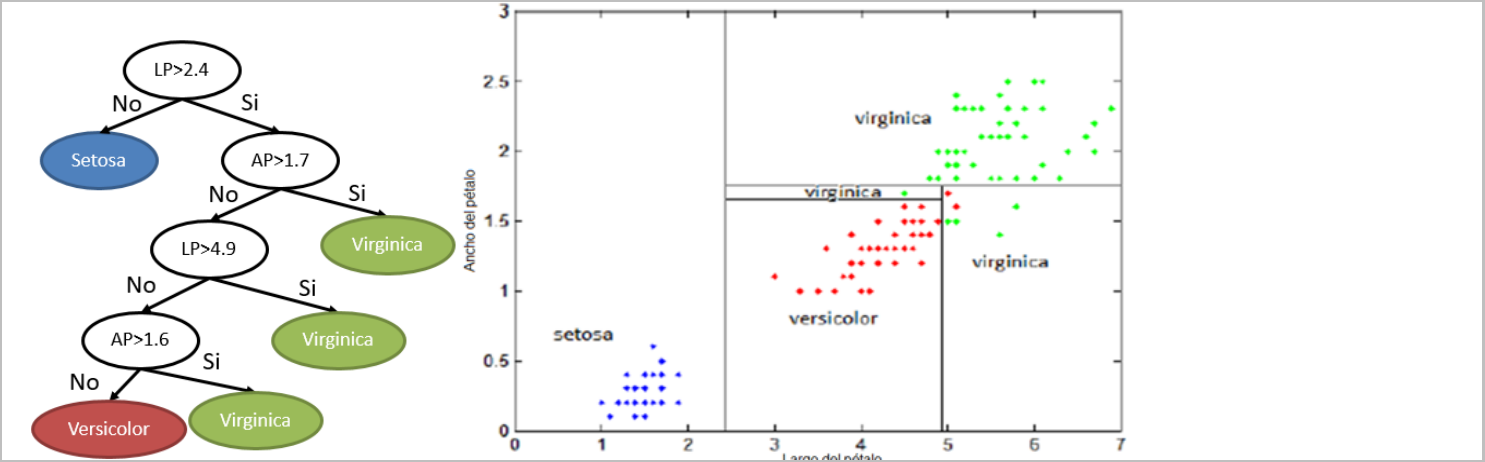
Example: Determine if a character is a woman or man

Personaje	Longitud Pelo	Peso	Edad	Género
 Homer	0"	250	36	H
 Marge	10"	150	34	M
 Bart	2"	90	10	H
 Lisa	6"	78	8	M
 Maggie	4"	20	1	M
 Abe	1"	170	70	H
 Selma	8"	160	41	M
 Otto	10"	180	38	H
 Krusty	6"	200	45	H
 Comic	8"	290	38	?

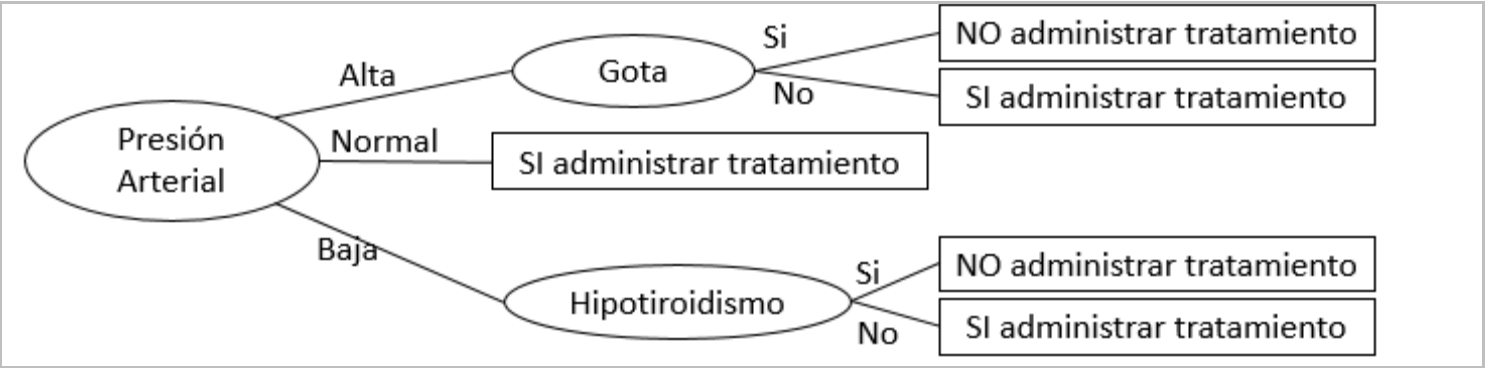
If weight <= 160
 If hair length <= 2
 man
 Else if (hair length >2)
 woman
Else if (weight > 160)
 man



Example: Decision tree for the classification problem of Fisher-iris:



Example: Decision tree for determining if a medication must be administrared or not



The elements of a decision tree are:

- **Nodes:** They represent a logical conditional that only depends on one feature. It can be seen as the if..else sentence. There may be several nodes with the same feature.
- **Branches:** They grow in the n nodes and correspond to the values that the logical conditional can have. Each node can have two o more branches.
- **Leaves:** They are the final nodes and correspond to the final conclusions.
 - If it is a classification decision tree: a leaf node represents a class
 - If it is a regression decision tree: a leaf node represents a value

6.1 ID3 algorithm

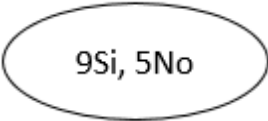
ID3 is an algorithm that constructs classification decision trees based on discrete inputs. It is recursive and generates the nodes from top to bottom recursively. Each time that a node is generated, it evaluates the division gain and calculates the feature that must be used in the node.

$$Division\ gain = Current\ entropy - \sum \frac{\#branch\ data}{\#node\ data} Branch\ entropy\ after\ division$$

Example:

Paciente	Presión Aterial	Urea en sangre	Gota	Hipotiroidismo	Administrar Tratamiento
1	Alta	Alta	Si	No	No
2	Alta	Alta	Si	Si	No
3	Normal	Alta	Si	No	Si
4	Baja	Normal	Si	No	Si
5	Baja	Baja	No	No	Si
6	Baja	Baja	No	Si	No
7	Normal	Baja	No	Si	Si
8	Alta	Normal	Si	No	No
9	Alta	Baja	No	No	Si
10	Baja	Normal	No	No	Si
11	Alta	Normal	No	Si	Si
12	Normal	Normal	Si	Si	Si
13	Normal	Alta	No	No	Si
14	Baja	Normal	Si	Si	No

Initial node



$$Entropy(9Si, 5No) = - \left[\frac{9}{14} * \log_2 \left(\frac{9}{14} \right) + \frac{5}{14} * \log_2 \left(\frac{5}{14} \right) \right] = 0.94$$

Dividing by: **presión arterial**

$$Gain(presion) = E(9S,5N) - \left[\frac{5}{14} E_{alta}(2S,3N) + \frac{4}{14} E_{normal}(4S,0N) + \frac{5}{14} E_{baja}(3S,2N) \right]$$

$$E(2,3) = - \left[\frac{2}{5} * \log_2 \left(\frac{2}{5} \right) + \frac{3}{5} * \log_2 \left(\frac{3}{5} \right) \right] = 0.954$$

$$E(4,0) = - \left[\frac{4}{4} * \log_2 \left(\frac{4}{4} \right) + \frac{0}{4} * \log_2 \left(\frac{0}{4} \right) \right] = 0$$

$$Gain(presion) = 0.94 - \left[\frac{5}{14} * 0.954 + \frac{4}{14} * 0 + \frac{5}{14} * 0.954 \right] = 0.258$$

Dividing by: **urea en sangre**

$$Gain(urea\ en\ sangre) = E(9S,5N) - \left[\frac{4}{14} E(2S,2N) + \frac{6}{14} E(4S,2N) + \frac{4}{14} E(3S,1N) \right]$$

$$= .94 - \left[\frac{4}{14} * 1 + \frac{6}{14} * 0.918 + \frac{4}{14} * 0.811 \right] = 0.029$$

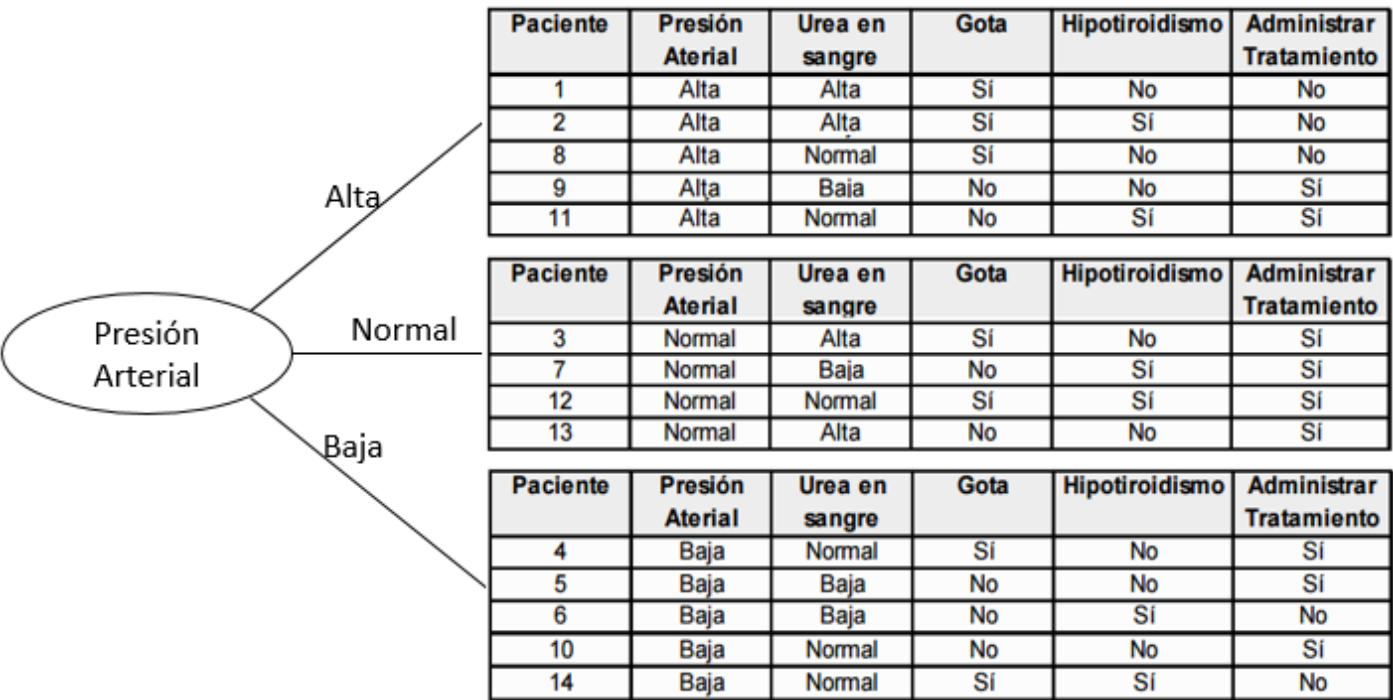
Dividing by: **gota**

$Gain(gota) = E(9S,5N) - \left[\frac{7}{14}E(3S,4N) + \frac{7}{14}E(6S,1N) \right] = .94 - \left[\frac{7}{14} * 0.98 + \frac{7}{14} * 0.59 \right] = 0.155$

Dividing by: **hipotiroidismo**

$Gain(hipotiroidismo) = E(9S,5N) - \left[\frac{6}{14}E(3S,3N) + \frac{8}{14}E(6S,3N) \right] = .94 - \left[\frac{6}{14} * 1 + \frac{8}{14} * 0.81 \right] = 0.05$

The first division must be **Presión Arterial** because it has the highest gain value.



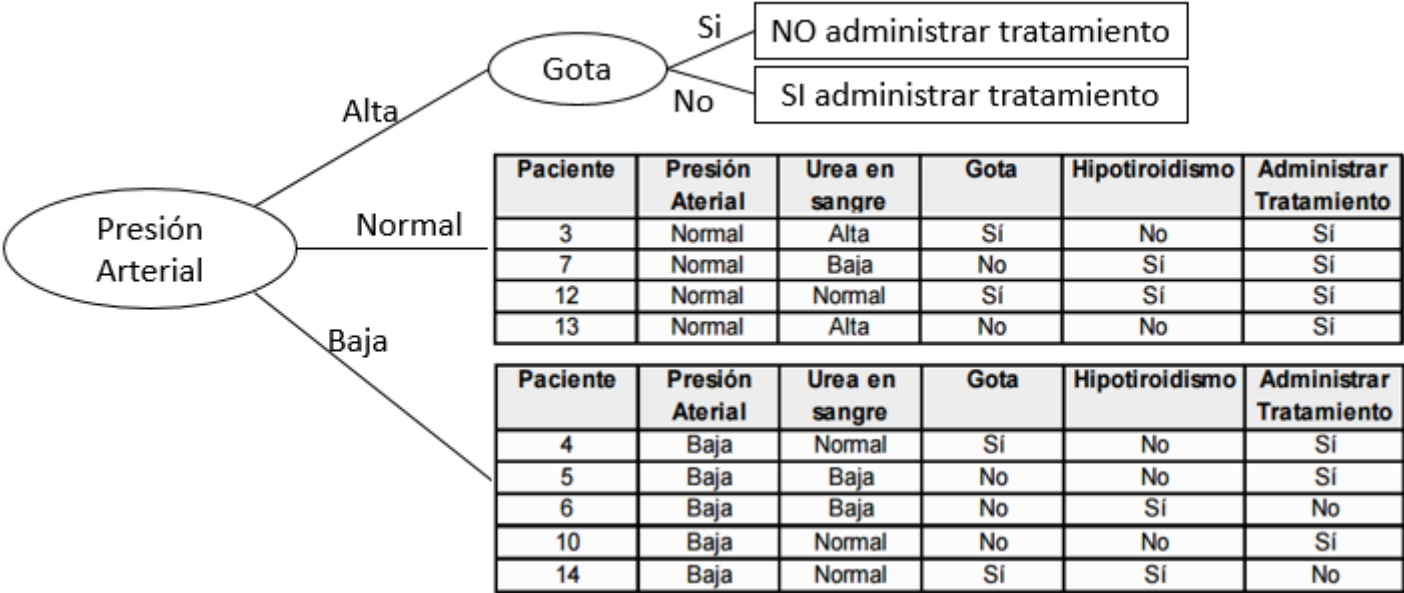
Then, each node needs to be analyzed and divided (if it is the case) recursively:

Presión Arterial Alta -----

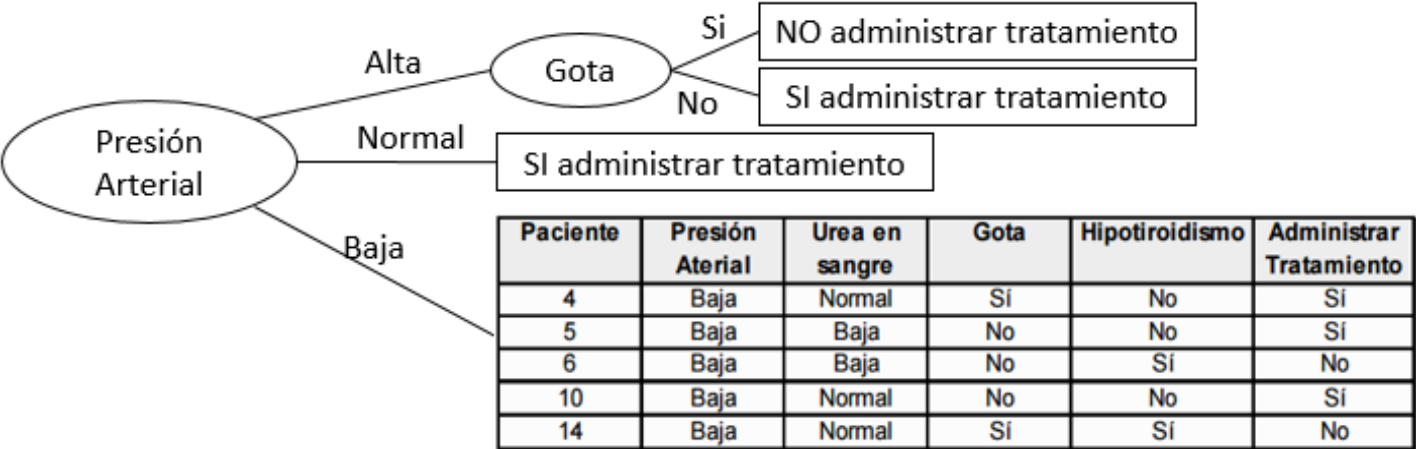
$Gain(urea\ en\ sangre) = E(2S,3N) - \left[\frac{2}{5}E(0S,2N) + \frac{2}{5}E(1S,1N) + \frac{1}{5}E(1S,0N) \right] = .97 - \left[\frac{2}{5} * 0 + \frac{2}{5} * 1 + \frac{1}{5} * 0 \right] = 0.57$

$Gain(gota) = E(2S,3N) - \left[\frac{3}{5}E(0S,3N) + \frac{2}{5}E(2S,0N) \right] = .97 - \left[\frac{3}{5} * 0 + \frac{2}{5} * 0 \right] = 0.97$

$Gain(hipotiroidismo) = E(2S,3N) - \left[\frac{2}{5}E(1S,1N) + \frac{3}{5}E(1S,2N) \right] = .97 - \left[\frac{2}{5} * 1 + \frac{3}{5} * 0.92 \right] = 0.018$

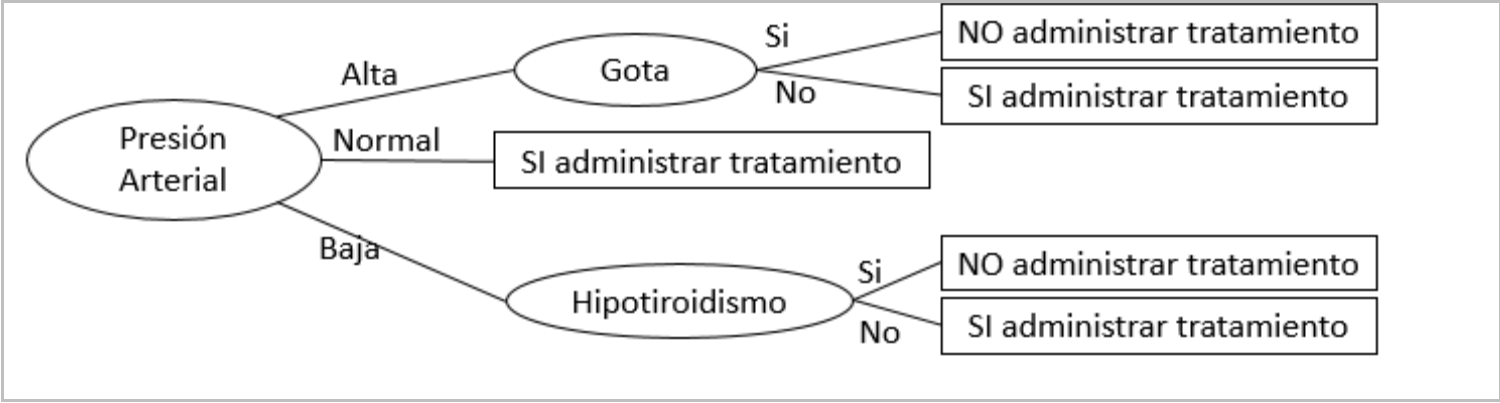


Presión Arterial Media -----
The node “Presión Arterial media” is a leaf node because it has an entropy value of 0 (all the values are yes).



Presión Arterial Baja -----

$$Gain(urea\ en\ sangre) = E(3S,2N) - \left[\frac{3}{5}E(2S,1N) + \frac{2}{5}E(1S,1N) \right] = .97 - \left[\frac{3}{5} * 0.92 + \frac{2}{5} * 1 \right] = 0.018$$
$$Gain(gota) = E(3S,2N) - \left[\frac{2}{5}E(1S,1N) + \frac{3}{5}E(2S,1N) \right] = .97 - \left[\frac{2}{5} * 1 + \frac{3}{5} * 0.92 \right] = 0.018$$
$$Gain(hipotiroidismo) = E(3S,2N) - \left[\frac{2}{5}E(0S,2N) + \frac{3}{5}E(3S,0N) \right] = .97 - \left[\frac{2}{5} * 0 + \frac{3}{5} * 0 \right] = 0.97$$



Activity: Use the ID3 algorithm to construct the decision tree to determine the note based on the following dataset.

Alumno	ATRIBUTO			Nota
	Punt. y asist.	Participación	Aprovechamiento	
1	No asiste	Media	Excelente	Exento
2	Asiste	Alta	Bueno	Exento
3	No asiste	Media	Bueno	Final
4	No asiste	Baja	Bueno	Final
5	Asiste	Alta	Regular	Final
6	Asiste	Baja	Deficiente	Extraordinario
7	No asiste	Media	Regular	Extraordinario

6.2 CART algorithm (Classification and Regression Trees)

CART is an algorithm for constructing decision trees for classification and regression with discrete and continuous data. CART constructs binary trees. It measures the impurity in each node, and the objective is to reduce the impurity in the leaf nodes.

Impurity measures	
Clasification	
Entropy	$H(f) = - \sum_{i=1}^m f_i \log_2 f_i$
Gini impurity	$I_G(f) = \sum_{i=1}^m f_i(1 - f_i) = \sum_{i=1}^m (f_i - f_i^2) = \sum_{i=1}^m f_i - \sum_{i=1}^m f_i^2 = 1 - \sum_{i=1}^m f_i^2$ <p>Where i goes for all the class labels, this is $i = \{1,2,\dots,m\}$, and f_i represents the percentage of data labeled as class i. Its range goes from 0 to ~1, where 0 is totally pure and ~1 is very impure.</p>
Regression	
Variance	$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2$

The nodes contain the following instructions:

- If the feature is numeric, the conditional follows the next structure:
If [feature_value] <= [value] then <NODE1>, else <NODE2>
- If the feature is categorical, the conditional follows the next structure:
If [feature_value] is in {value1,value2,...,valuen} then <NODE1>, else <NODE2>

Algorithm for the construction of a decision tree:

Input:
<ul style="list-style-type: none">• X: training data, where $X \in \mathbb{R}^{M \times N}$, M is the samples number and N the features number.• Y: output of the training data, where $Y \in \mathbb{R}^M$.
Output: a decision tree.
Create the first node with all the training data.
Divide the first node using the function NodeDivision().

NodeDivision() (recursive function to divide all the nodes)

<ul style="list-style-type: none">• Calculate the node impurity• Test all the possible ways to divide the node and choose the feature and the value with the highest gain
$Division\ gain = Current\ impurity - \sum \frac{\#branch\ data}{\#node\ data} Branch\ impurity\ after\ division$
<ul style="list-style-type: none">• If the gain is less than 0: DO NOT DIVIDE THE NODE• If the gain is higher than 0: the node is divided and the function NodeDivision is called for each sub node

Feature importance:

Feature importance is proportional to the impurity reduction of all nodes related to that feature. The impurity reduction IR_j in each node j representing a rule can be calculated with:

$$IR_j = w_j I_j - (w_{left} I_{left} + w_{right} I_{right})$$

where *left* and *right* represent the children's nodes of node j , I represents the impurity of each node, and the weights w are the samples' proportion in nodes, and they are calculated as the number of samples in the node divided by the total number of samples. Once the impurity reduction in all nodes is known, the importance of the feature k , FI_k , is calculated as following:

$$FI_k = \frac{\sum_{j \in N_k} IR_j}{\sum_{j \in N} IR_j}$$

6.3 Advantages and disadvantages of decision trees

Advantages:

- Easy to interpret and understand
- Require few or null data pre-processing
- Can handle numerical and categorical data (sklearn version only can be executed with numerical data)
- It is a white box model
- It is possible to validate the model using statistical tests
- Work well with big data

Disadvantages:

- They have a big problem with overfitting
- They are greedy algorithms

6.4 Ensemble

Ensemble: It creates several simple learning models with the same or different training datasets. The final decision is calculated based on a votation scheme. Ensembles obtain better predictive performance than all their models by themselves. They reduce the overfitting because the models are very simple.

Random Forest is a supervised learning model formed by several decision trees. All the trees are different among them because they are trained with a random subset of samples and features.