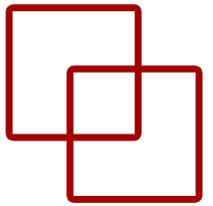
Árboles de decisión

Manuel Castillo-Cara, Luis Sarro
www.manuelcastillo.eu
Department of Artificial Intelligence
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia (UNED)

Índice



- CART
- Algoritmo base
- Construir un árbol
- Ganancia de información
- Árboles de regresión
- Selección de modelos

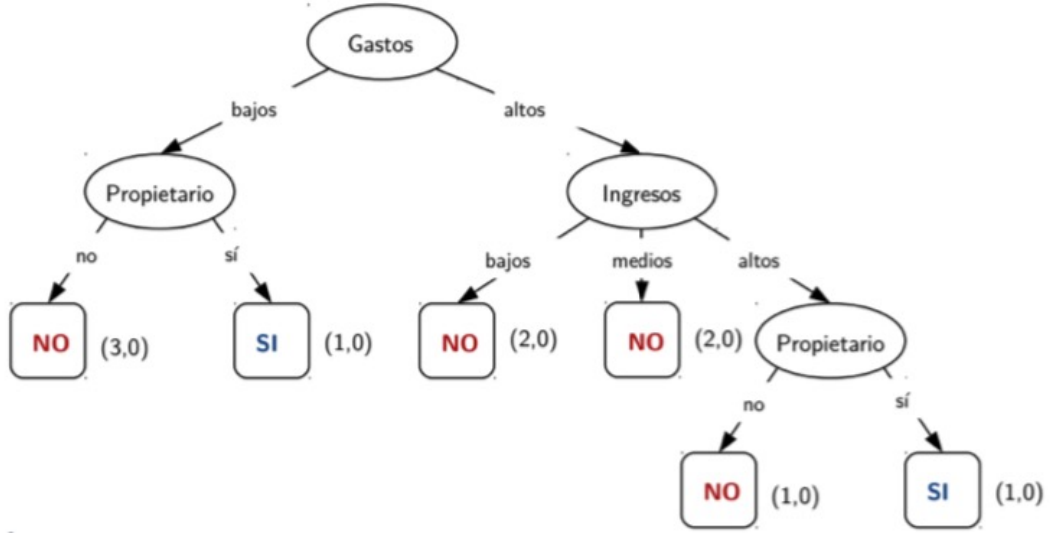
CART

Definición

- Un **árbol de decisión** representa la función de hipótesis mediante grafo dirigido, $f(X) \rightarrow C$, tal que:
  - Cada **nodo** representa una variable de **entrada**, $X_i \in X$.
  - Cada **rama** que pende de un nodo representa uno de los posibles valores que puede tomar la variable correspondiente X_i.
  - Las **hojas** corresponden con valores de la variable de las **clases**.
  - Un ejemplo se clasifica recorriendo el árbol desde la raíz y eligiendo en cada momento la rama que satisface la condición para el valor del atributo correspondiente. La clase elegida sería la asignada a la hoja a que se llega.

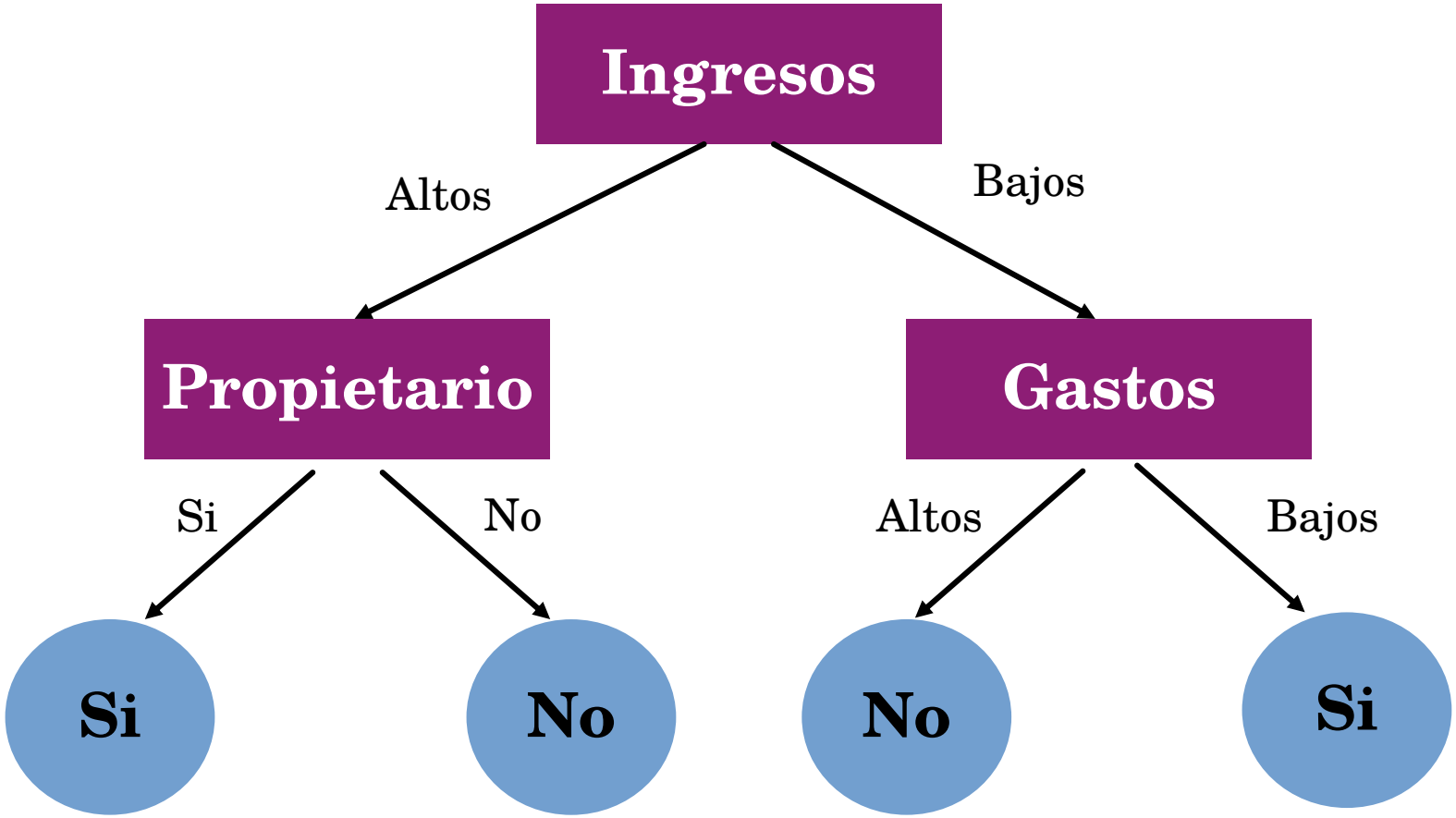
Ejemplo:

ingresos	propietario	gastos	crédito
bajos	no	altos	NO
bajos	si	altos	NO
medios	si	altos	NO
medios	no	altos	NO
altos	no	altos	NO
altos	si	altos	SI
bajos	no	bajos	NO
medios	no	bajos	NO
altos	no	bajos	NO
medios	si	bajos	SI



Ejemplo

- Ingresos=Altos, Propietario=No —> Crédito = ?
- Ingresos=Bajos, Gastos=Altos —> Crédito = ?



Algoritmo base

Algoritmo ID3

Algoritmo recursivo

Para parar la recursión

Volvemos a llamar al árbol con los atributos menos el que acabo de expandir, la clase y la proyección realizada

Algoritmo 1 ID3 (Algoritmo de Hunt utilizando la ganancia de información)

```
1: A: Lista de atributos
2: Y: Variable de clase
3: D: Partición de la base de datos correspondiente a la rama (o raíz)
4: procedimiento CONSTRUIRÁRBOL(A,Y,D)
5:       # Si toda la base de datos tiene la misma clase, crea una hoja
6:     si y^(i) = c_k   ∀(x^(i), y^(i)) ∈ D entonces
7:         devuelve CREARHOJA(c_k)
8:     fin si
9:       # Si no quedan atributos, se devuelve la clase mayoritaria
10:    si A = ∅ entonces
11:        c_k ←CLASEMAYORITARIA(D)
12:        devuelve CREARHOJA(c_k)
13:    fin si
14:     # Si no es una hoja, se crea un nodo normal.
15:    X_max ← MaxGain(A,D) # Selecciona el atributo con mayor ganancia de información.
16:    Nodo ← CREARNODO(X_max) # Crea un nodo con la variable seleccionada
17:    para x_l posible valor de X_max hacer
18:        AÑADIRRAMA(nodo,x_l)
19:        D_l ← {(x^(i), y^(i)) ∈ D|X^(i)_max = x_l}
20:        CONSTRUIRÁRBOL(A − X_max, Y, D_l)
21:    fin para
22:    devuelve Nodo
23: fin procedimiento
```

Construcción del árbol

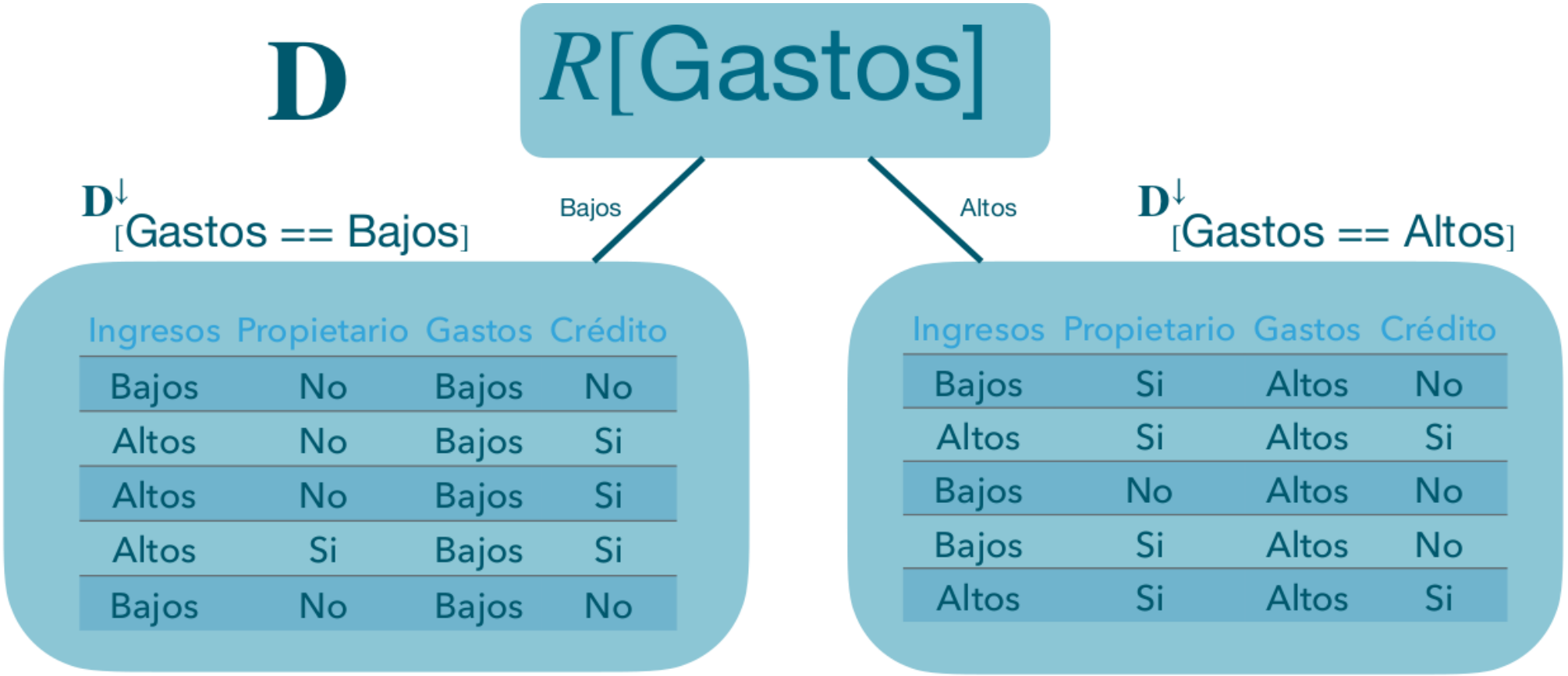
Variable “Gastos”

D R

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Construcción del árbol

Variable “Gastos”

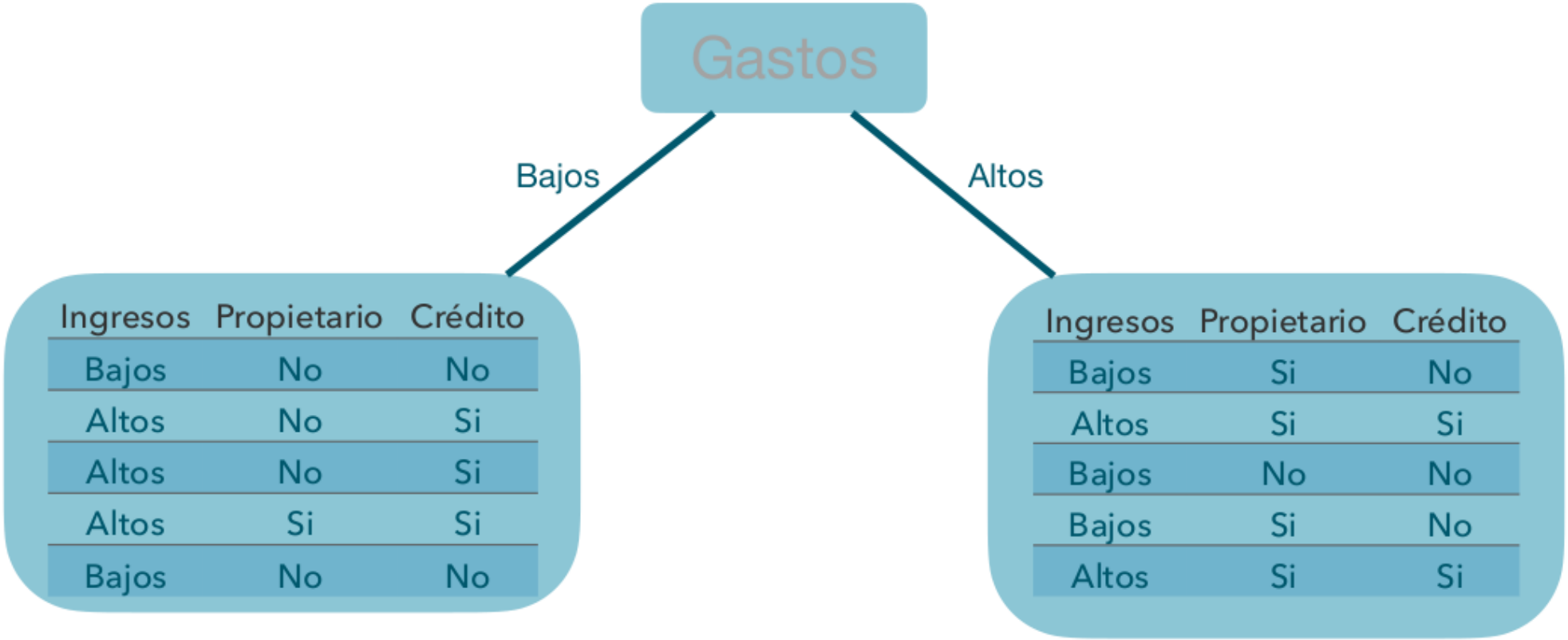


Ingresos	Propietario	Gastos	Crédito
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si

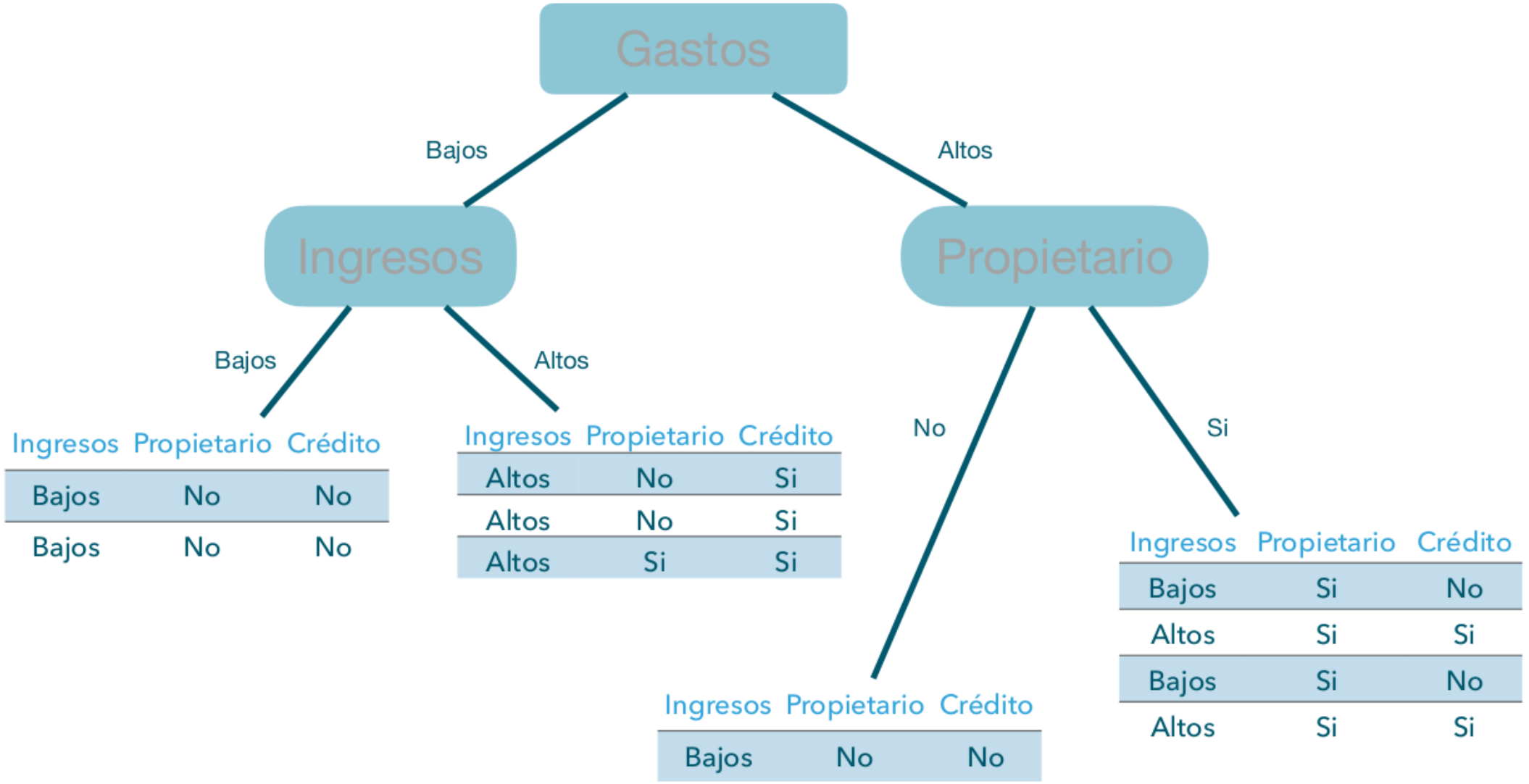
Construcción del árbol

Variable “Gastos”



Construcción del árbol

Variable “Gastos”



Ingresos	Propietario	Crédito
Bajos	No	No
Bajos	No	No

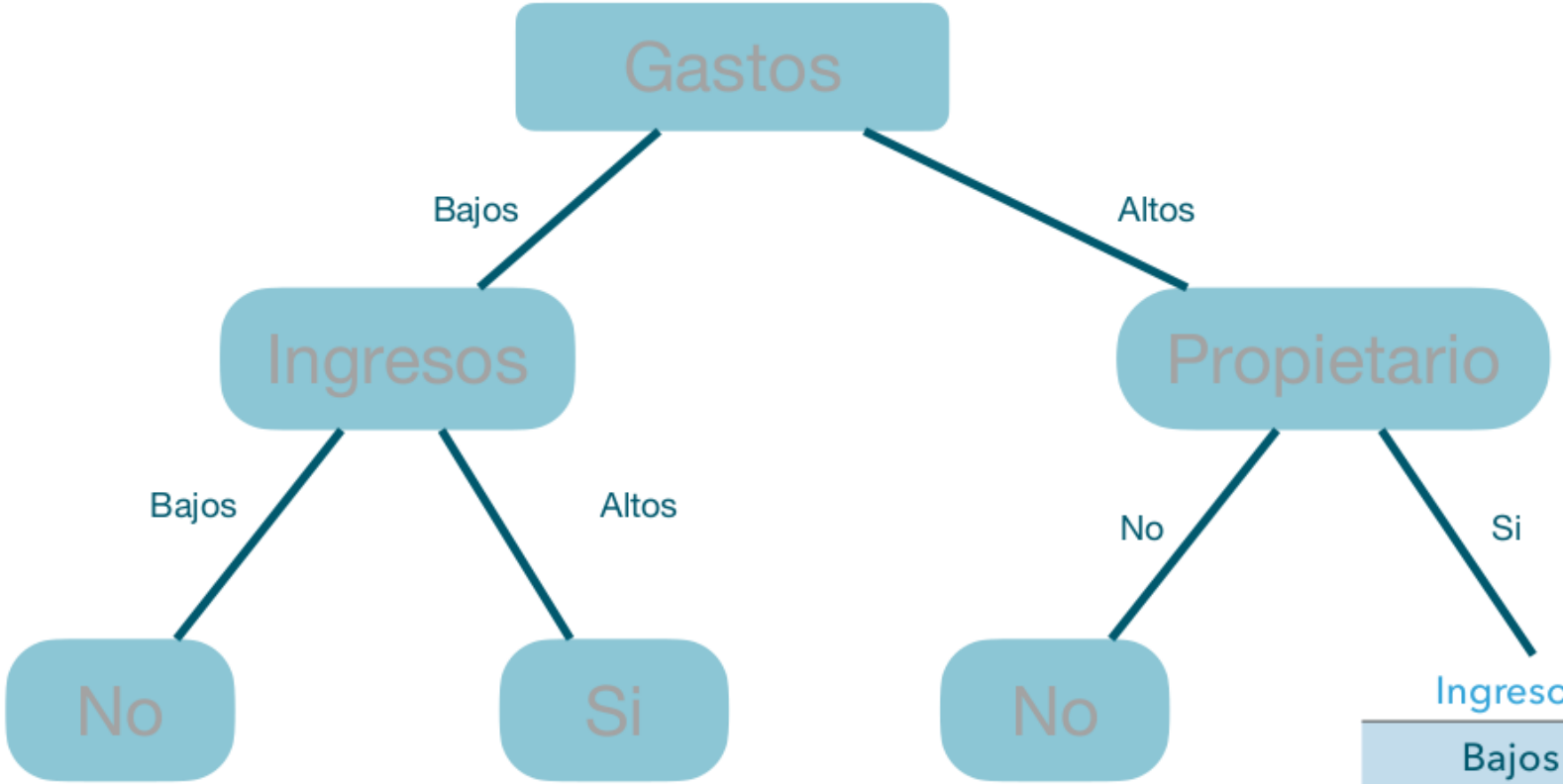
Ingresos	Propietario	Crédito
Altos	No	Si
Altos	No	Si
Altos	Si	Si

Ingresos	Propietario	Crédito
Bajos	No	No

Ingresos	Propietario	Crédito
Bajos	Si	No
Altos	Si	Si
Bajos	Si	No
Altos	Si	Si

Construcción del árbol

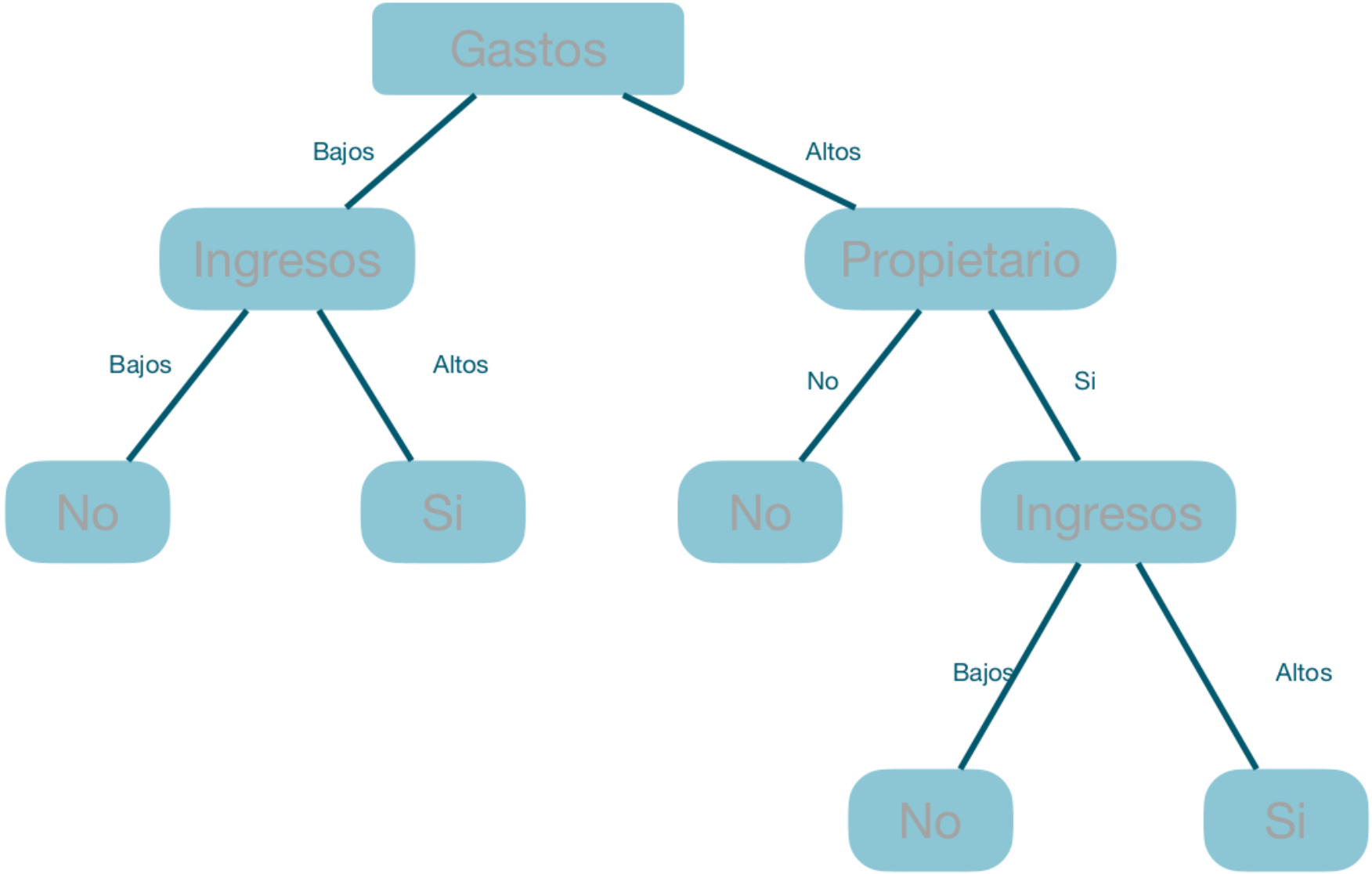
Variable “Gastos”



Ingresos	Crédito
Bajos	No
Altos	Si
Bajos	No
Altos	Si

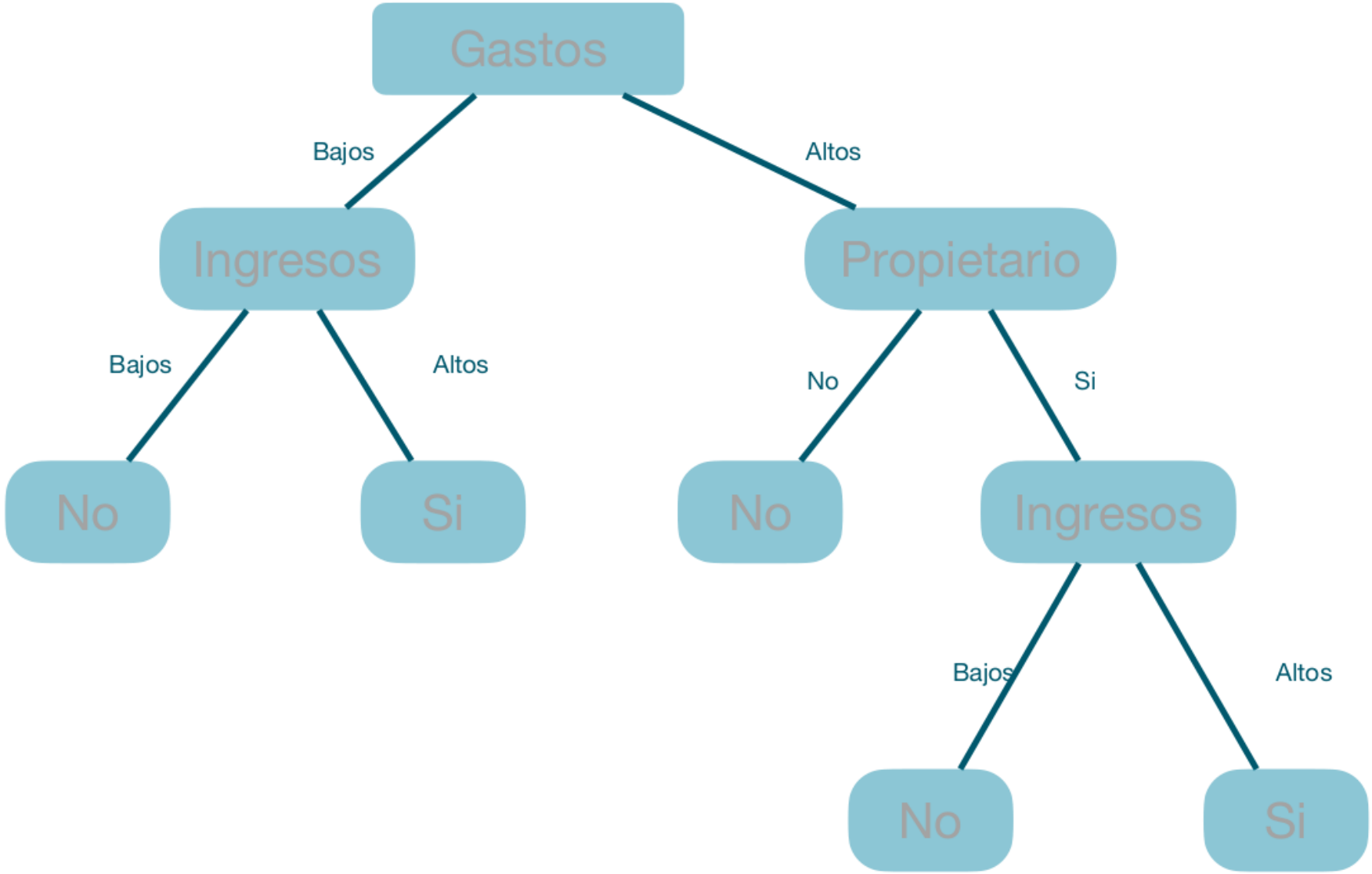
Construcción del árbol

Variable “Gastos”

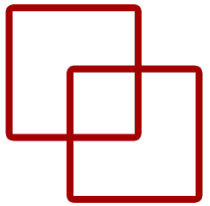


Construcción del árbol

Variable “Gastos”

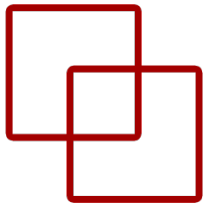


¿Cuál es el mejor árbol?



- Para elegir el mejor árbol posible debemos especificar un criterio.

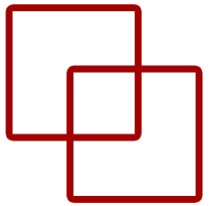
¿Cuál es el mejor árbol?



- Para elegir el mejor árbol posible debemos especificar un criterio.

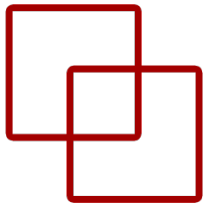
- Queremos el mejor árbol que se adapte a los datos y que sea menos complejo.

¿Cuál es el mejor árbol?



- Para elegir el mejor árbol posible debemos especificar un criterio.
- Queremos el mejor árbol que se adapte a los datos y que sea menos complejo.
- ¿Adaptación del árbol a los datos?

¿Cuál es el mejor árbol?



- Para elegir el mejor árbol posible debemos especificar un criterio.
- Queremos el mejor árbol que se adapte a los datos y que sea menos complejo.
- ¿Adaptación del árbol a los datos?
- ¿Complejidad de un árbol?

Construcción del árbol

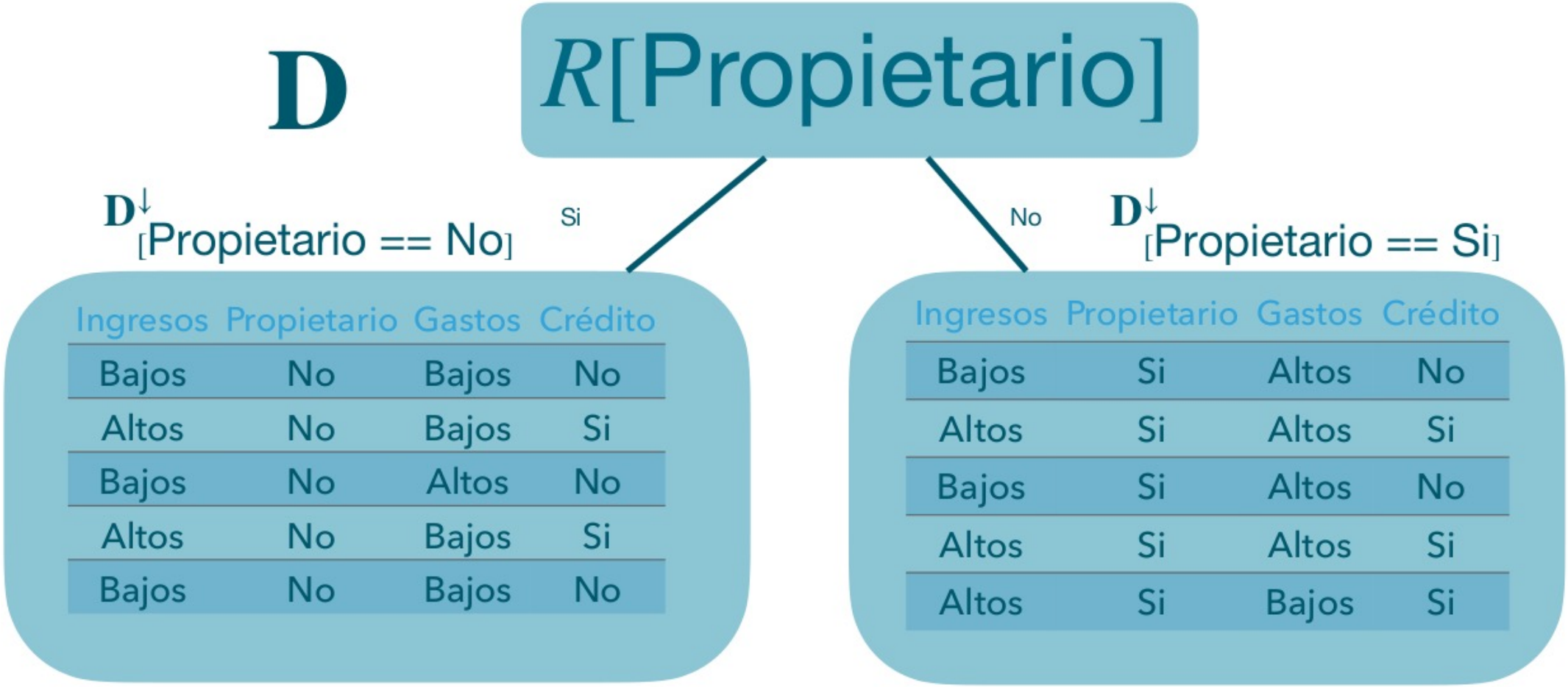
Variable “Propietario”

D *R*

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Construcción del árbol

Variable “Propietario”

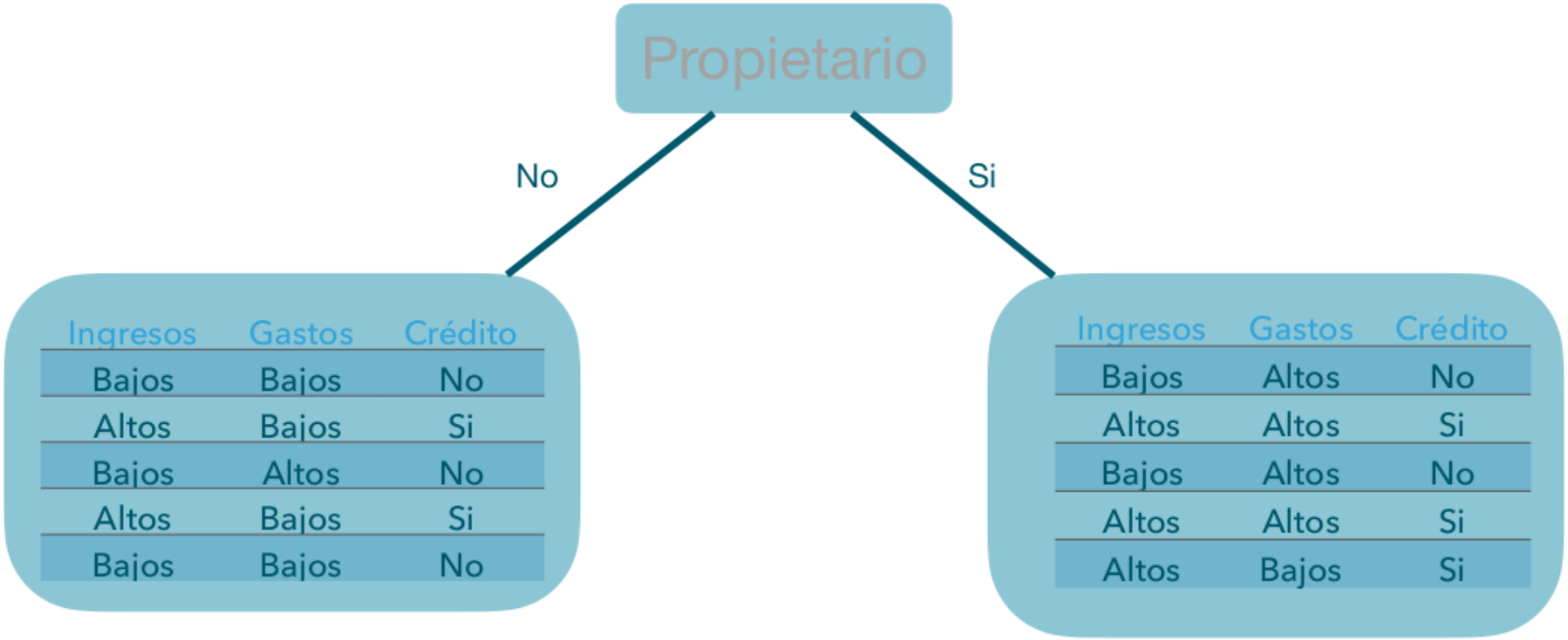


Ingresos	Propietario	Gastos	Crédito
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Bajos	No	Altos	No
Altos	No	Bajos	Si
Bajos	No	Bajos	No

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	Si	Bajos	Si

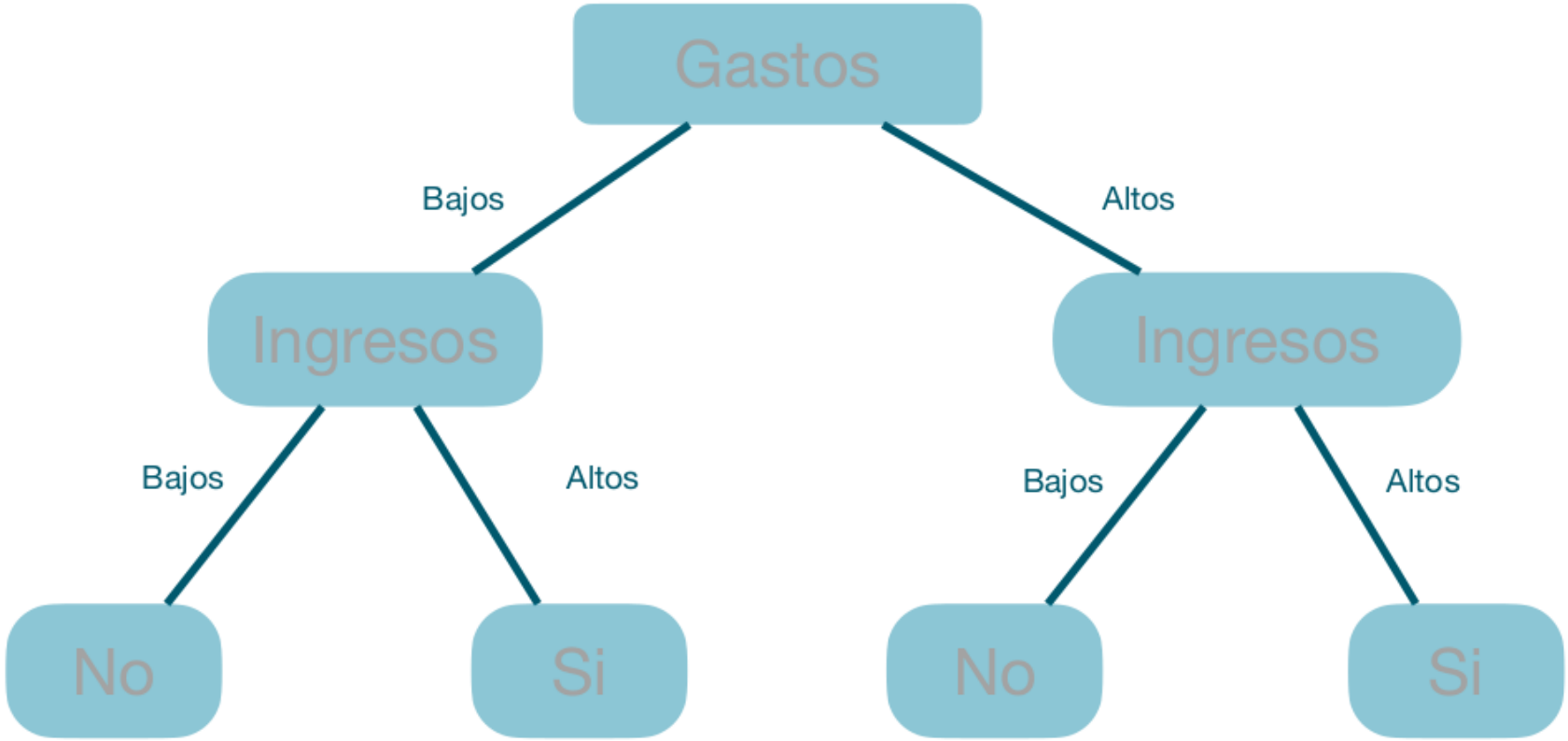
Construcción del árbol

Variable “Propietario”



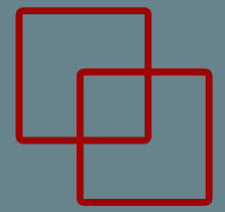
Construcción del árbol

Variable “Gastos” (desde otra perspectiva)



Construcción del árbol

Variable “Gastos” (desde otra perspectiva)

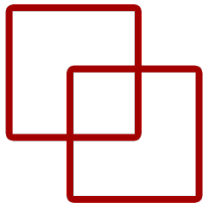


UNED

ETS de Ingeniería Informática

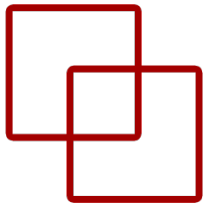
Construir un árbol

Introducción



- Se construyen particionando el espacio de entrada de manera recursiva.
  - En cada paso se elige la variable que produce la **partición óptima**.
  - La partición será la rama que se representa en un árbol.

Introducción

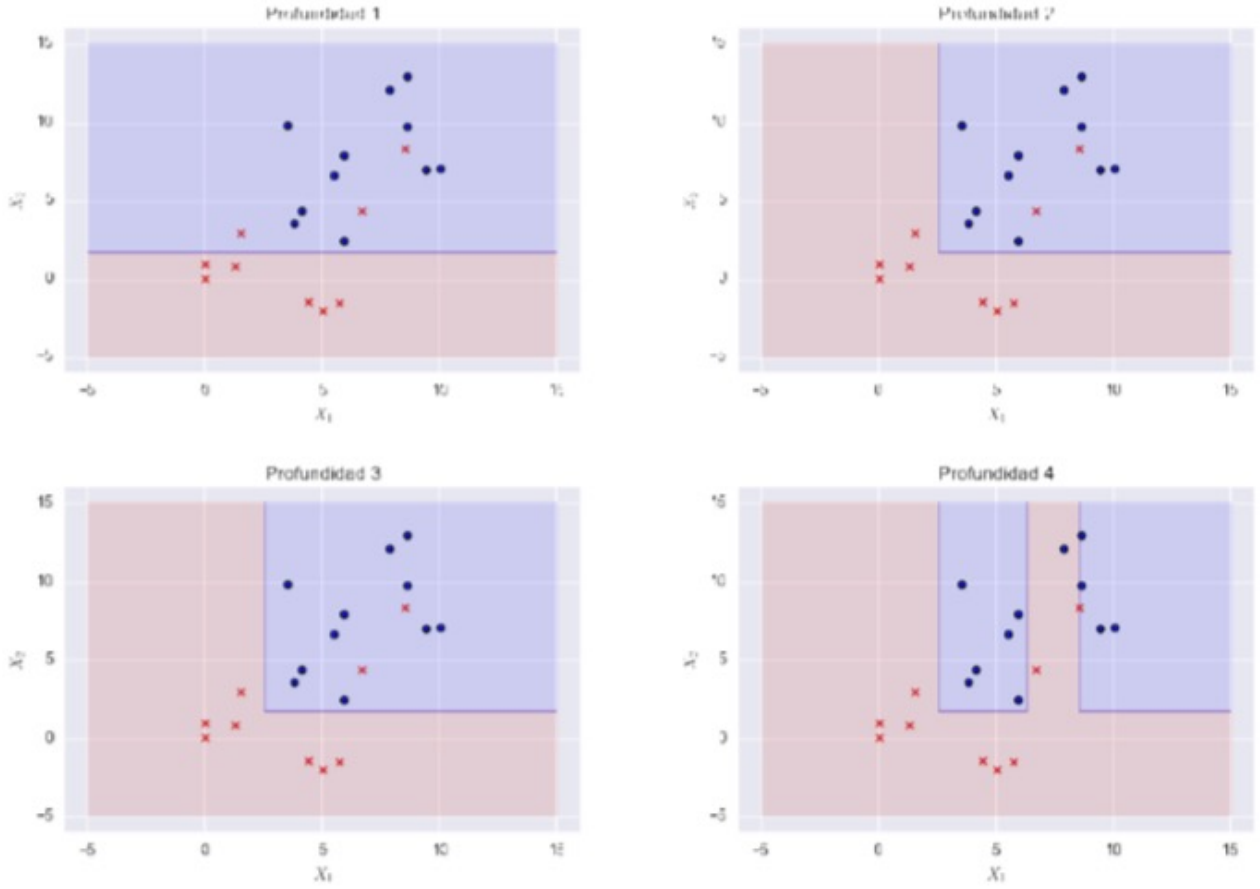


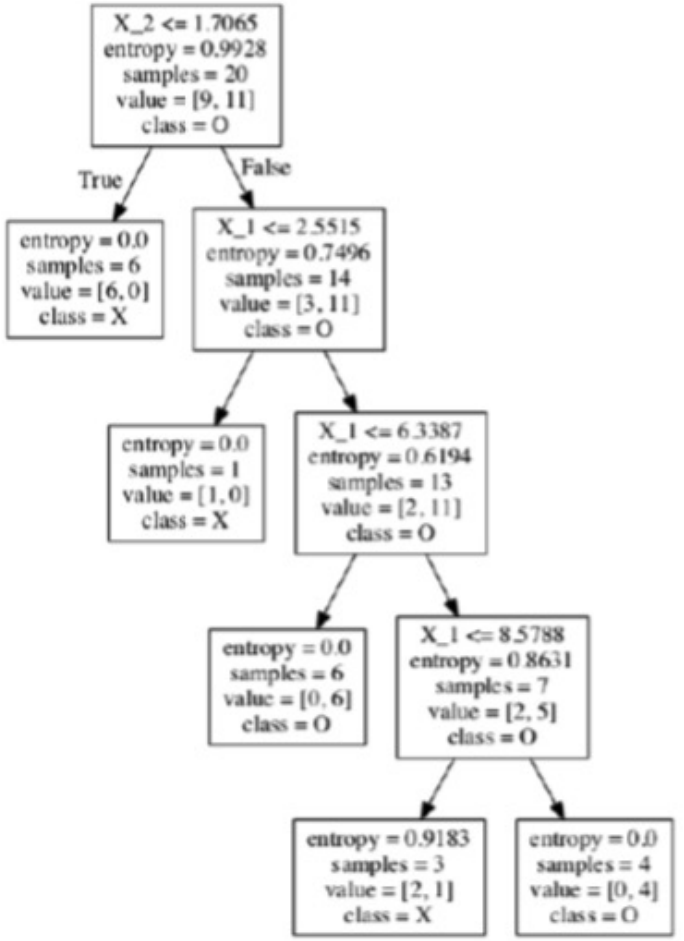
- Se construyen particionando el espacio de entrada de manera recursiva.
  - En cada paso se elige la variable que produce la **partición óptima**.
  - La partición será la rama que se representa en un árbol.

- En caso de entrada se procesa **recorriendo** el árbol, eligiendo en cada nodo el correspondiente al valor de la variable, y asignando el valor correspondiente a la hoja alcanzada.

Regresión

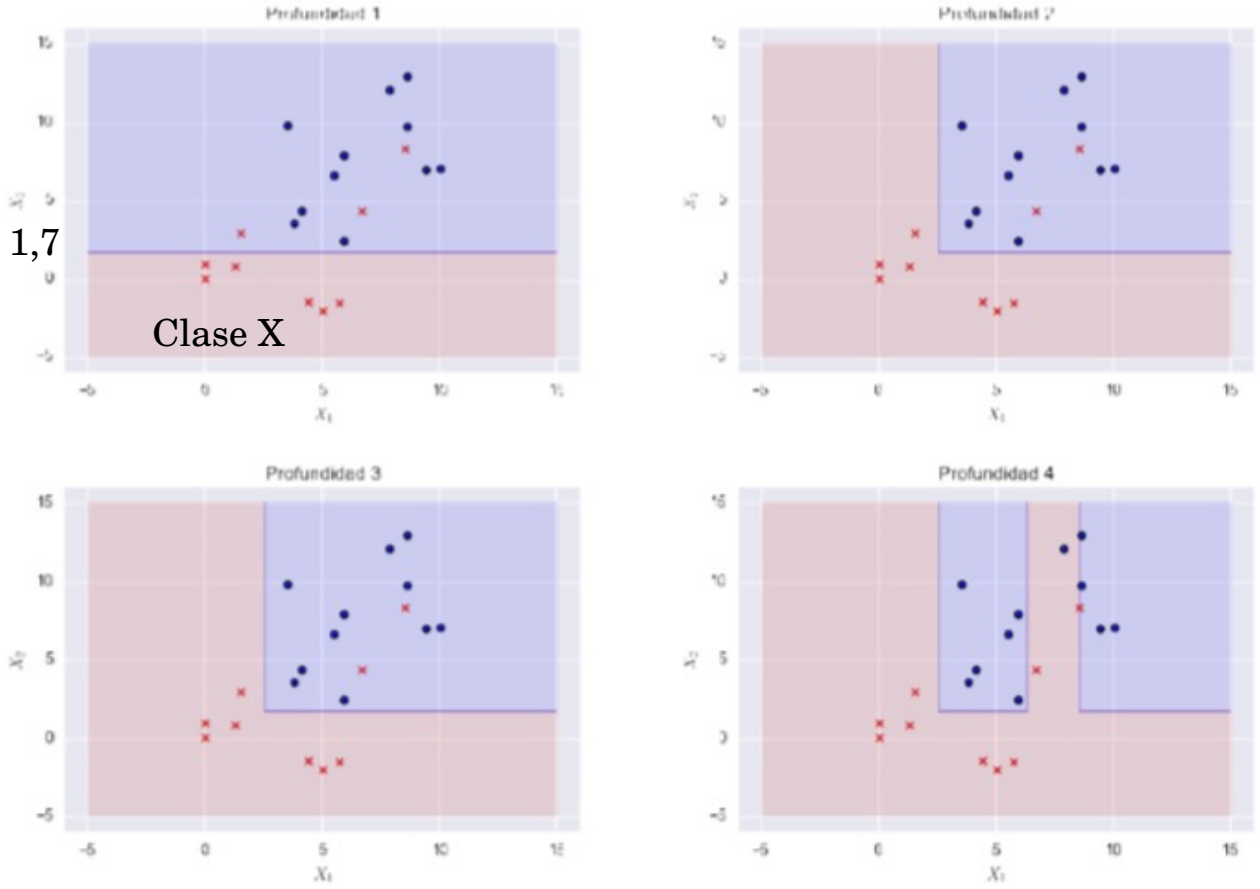
Ejemplo con profundidad máxima 4

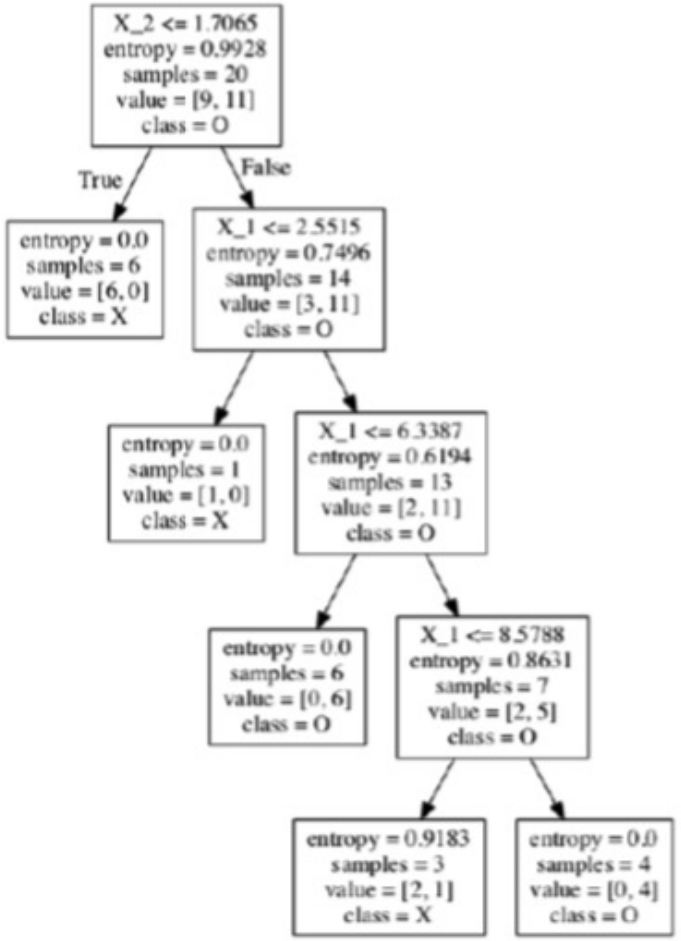




Regresión

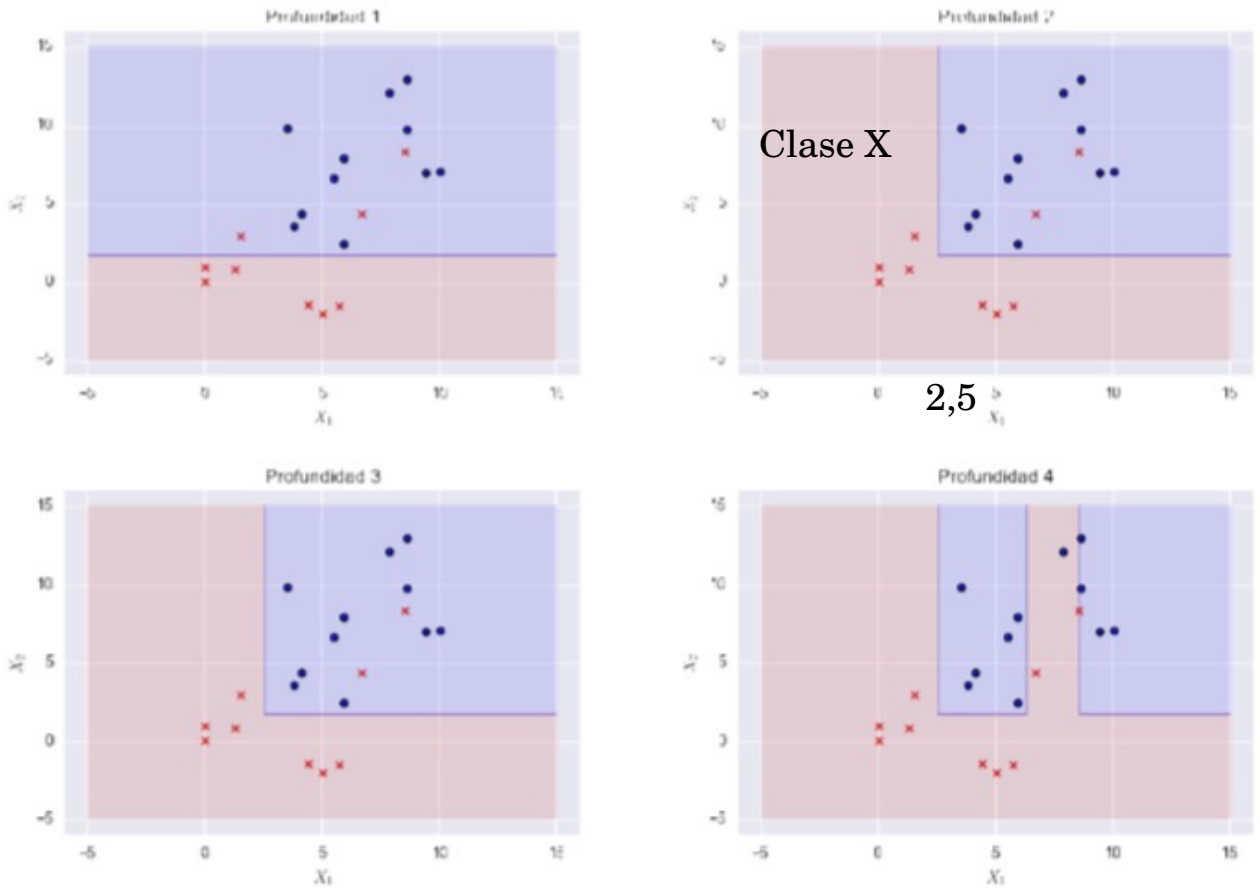
Ejemplo con profundidad máxima 4

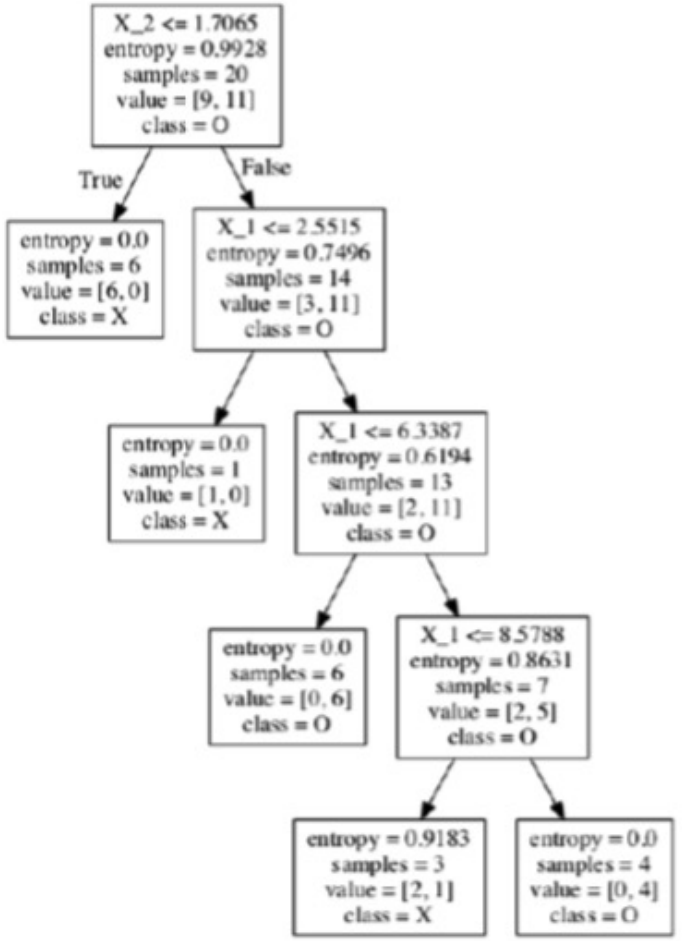




Regresión

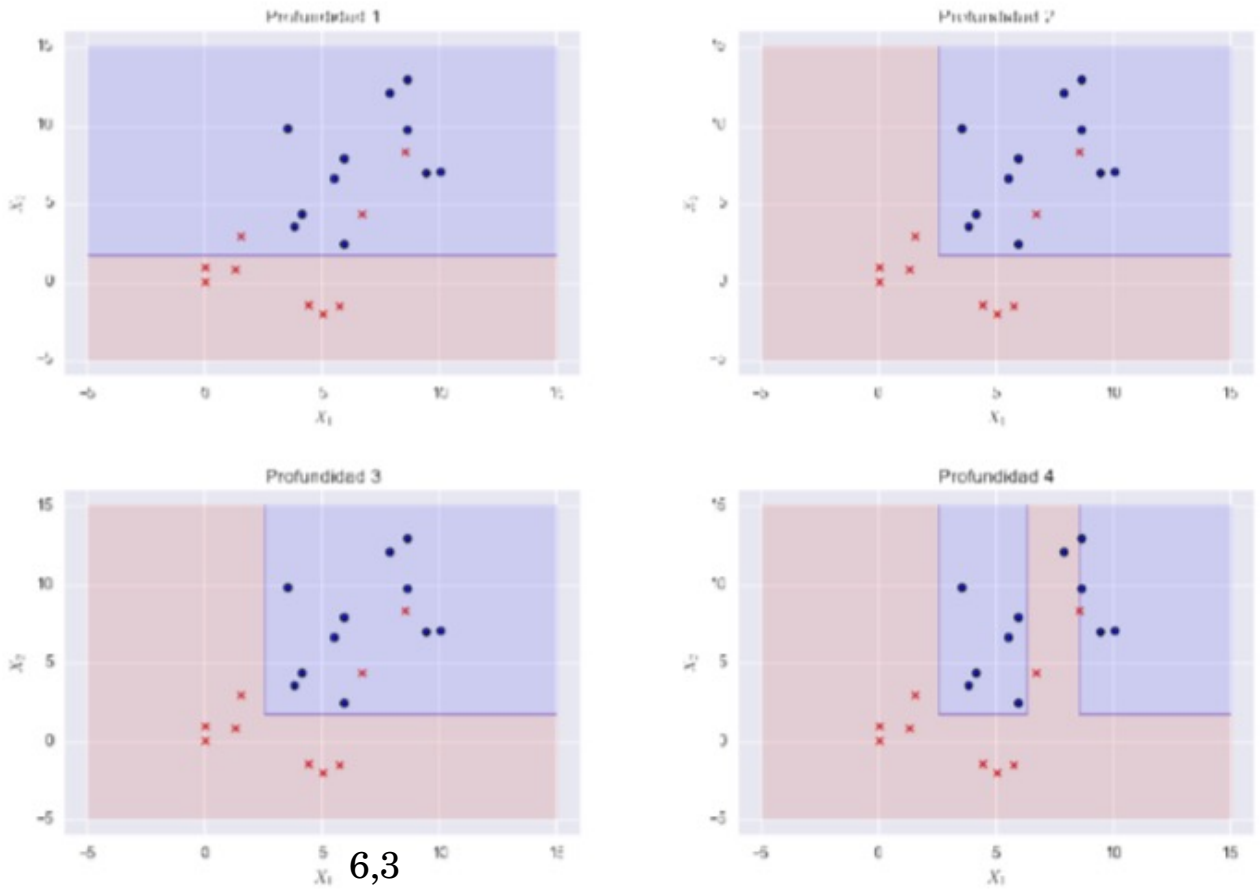
Ejemplo con profundidad máxima 4



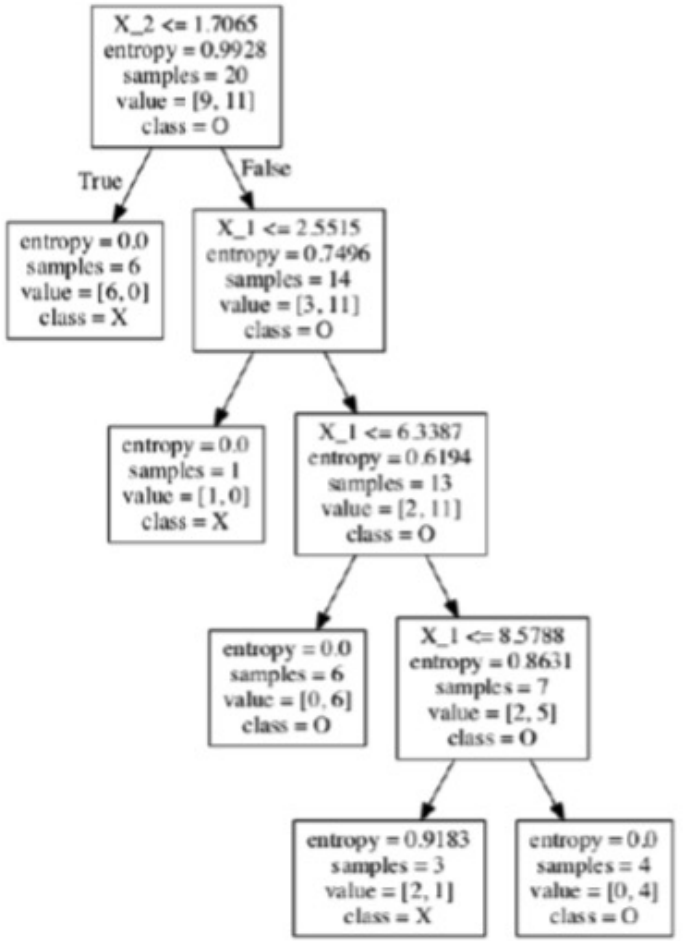


Regresión

Ejemplo con profundidad máxima 4



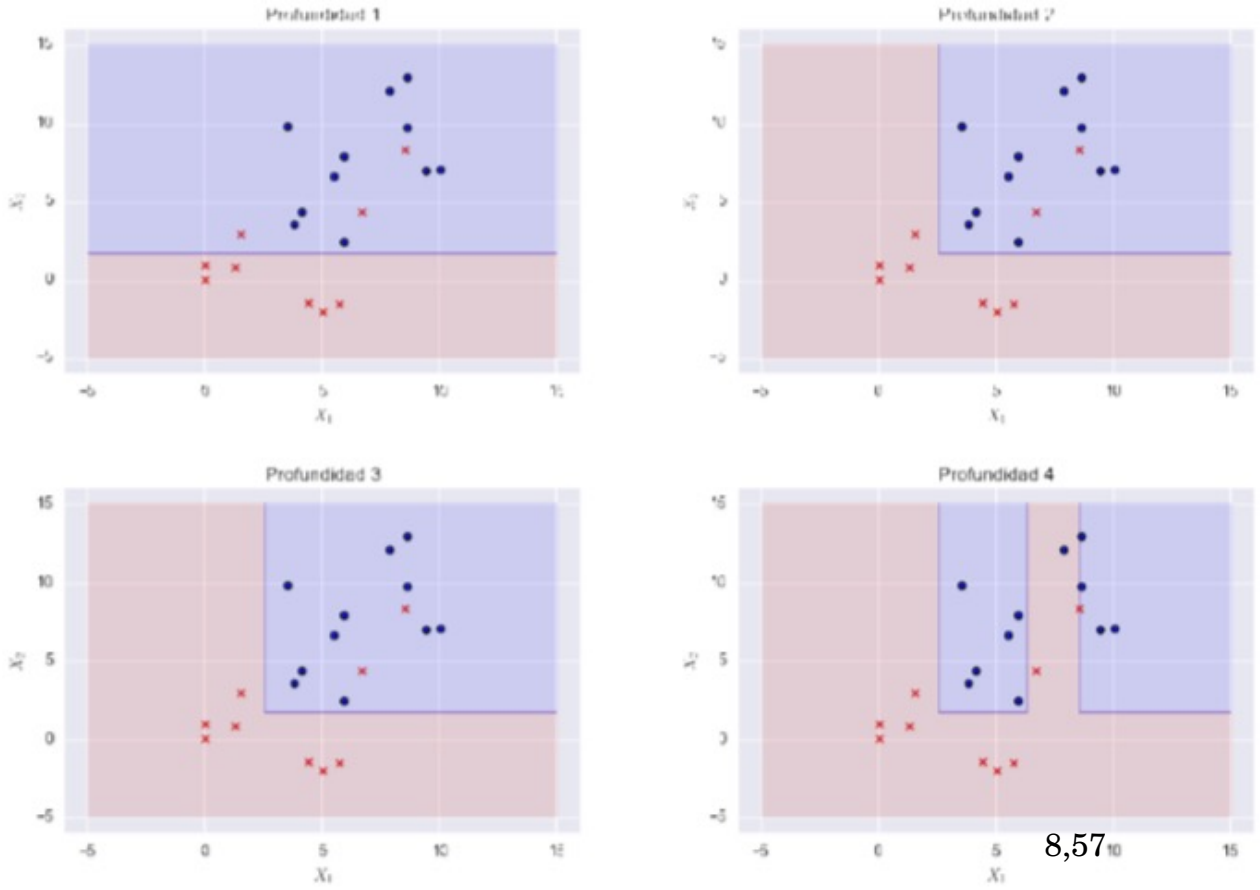
6,3

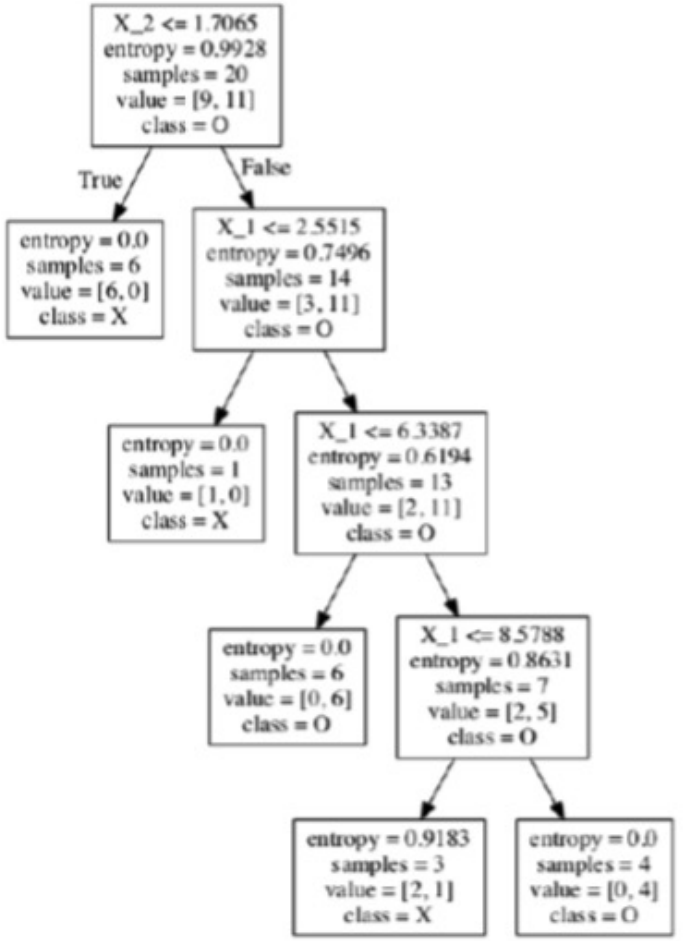


Clase O

Regresión

Ejemplo con profundidad máxima 4

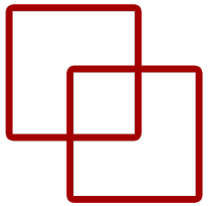




Clase X

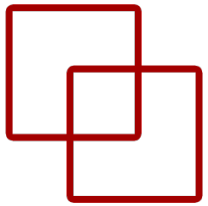
8,57

Algoritmo basado en divide y vencerás



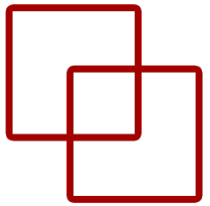
- La construcción se realiza, generalmente, de forma voraz, siguiendo dicho esquema de particionamiento.

Construcción de un árbol
Algoritmo basado en divide y vencerás



- La construcción se realiza, generalmente, de forma voraz, siguiendo dicho esquema de particionamiento.
- **TDIDT:** *Top-Down Induction of Decision Trees*
  - Algoritmo voraz que construye un árbol de decisión aplicando recursivamente y de forma *top-down* un procedimiento de divide y vencerás.

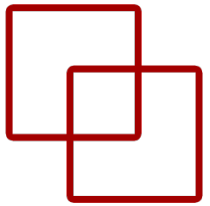
Construcción de un árbol
Algoritmo basado en divide y vencerás



- La construcción se realiza, generalmente, de forma voraz, siguiendo dicho esquema de particionamiento.
- **TDIDT:** *Top-Down Induction of Decision Trees*
  - Algoritmo voraz que construye un árbol de decisión aplicando recursivamente y de forma *top-down* un procedimiento de divide y vencerás.

– Un árbol de decisión se construye recursivamente desde la raíz.

Construcción de un árbol
Algoritmo basado en divide y vencerás



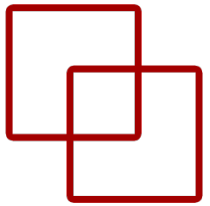
- La construcción se realiza, generalmente, de forma voraz, siguiendo dicho esquema de particionamiento.
- **TDIDT:** *Top-Down Induction of Decision Trees*
  - Algoritmo voraz que construye un árbol de decisión aplicando recursivamente y de forma *top-down* un procedimiento de divide y vencerás.

– Un árbol de decisión se construye recursivamente desde la raíz.

– El procedimiento recibe un dataset para crear un nodo.

- Si para dichos datos la variable **clase tiene muy poca variabilidad o son todos iguales**, el proceso se detienen **creando una variable hoja**.
- **En otro caso**, usando la información contenida en los datos, **se selecciona una variable como criterio de decisión**.

Algoritmo basado en divide y vencerás

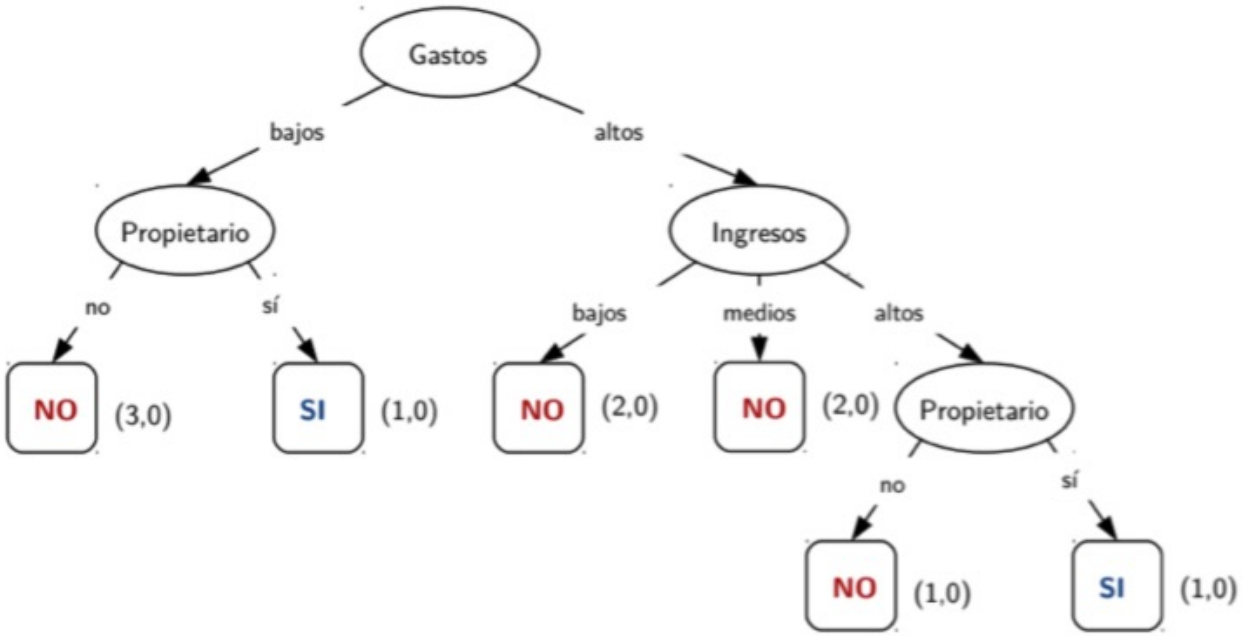


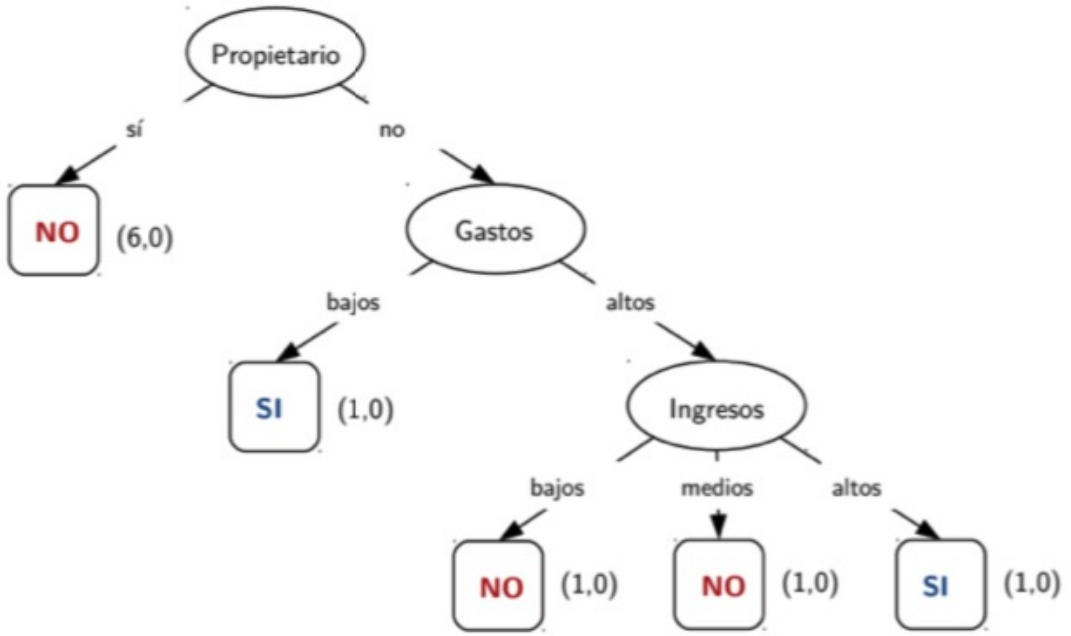
- La construcción se realiza, generalmente, de forma voraz, siguiendo dicho esquema de particionamiento.
- **TDIDT:** *Top-Down Induction of Decision Trees*
  - Algoritmo voraz que construye un árbol de decisión aplicando recursivamente y de forma *top-down* un procedimiento de divide y vencerás.

- Un árbol de decisión se construye recursivamente desde la raíz.
- El procedimiento recibe un dataset para crear un nodo.
  - Si para dichos datos la variable **clase tiene muy poca variabilidad o son todos iguales**, el proceso se detienen **creando una variable hoja**.
  - **En otro caso**, usando la información contenida en los datos, **se selecciona una variable como criterio de decisión**.
- Usando la variable seleccionada, el dataset se particiona en una serie de subconjuntos y se invoca el procedimiento recursivamente para cada uno de ellos.

Construcción de un árbol
Selección de modelos

¿Qué árbol es mejor?

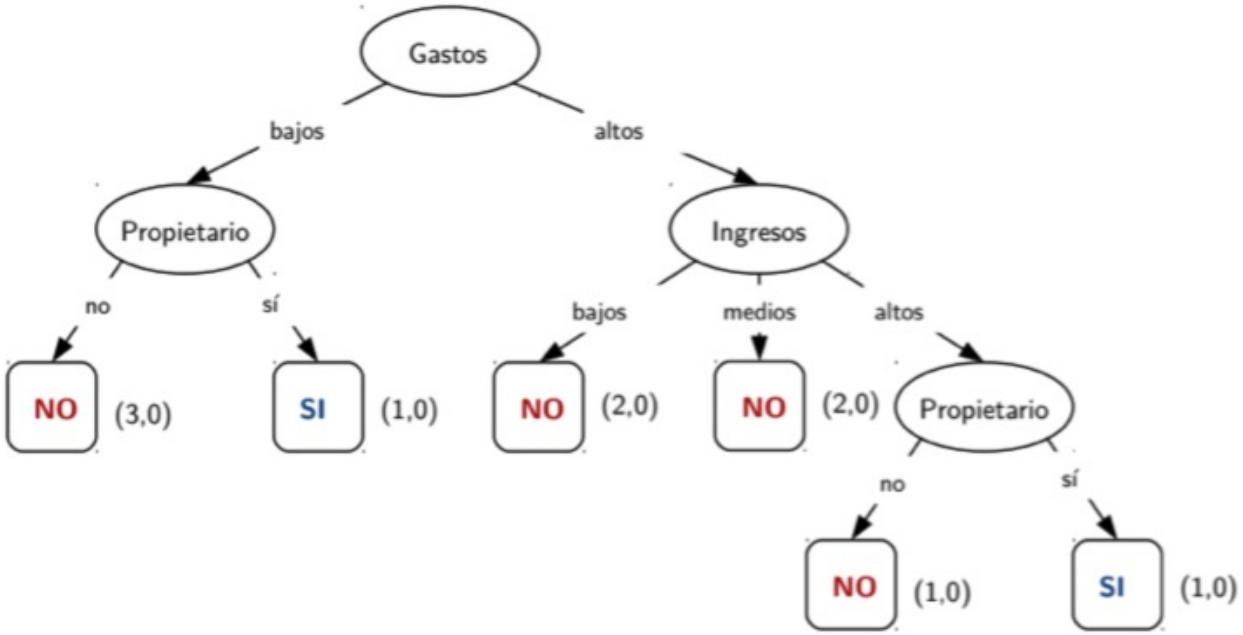


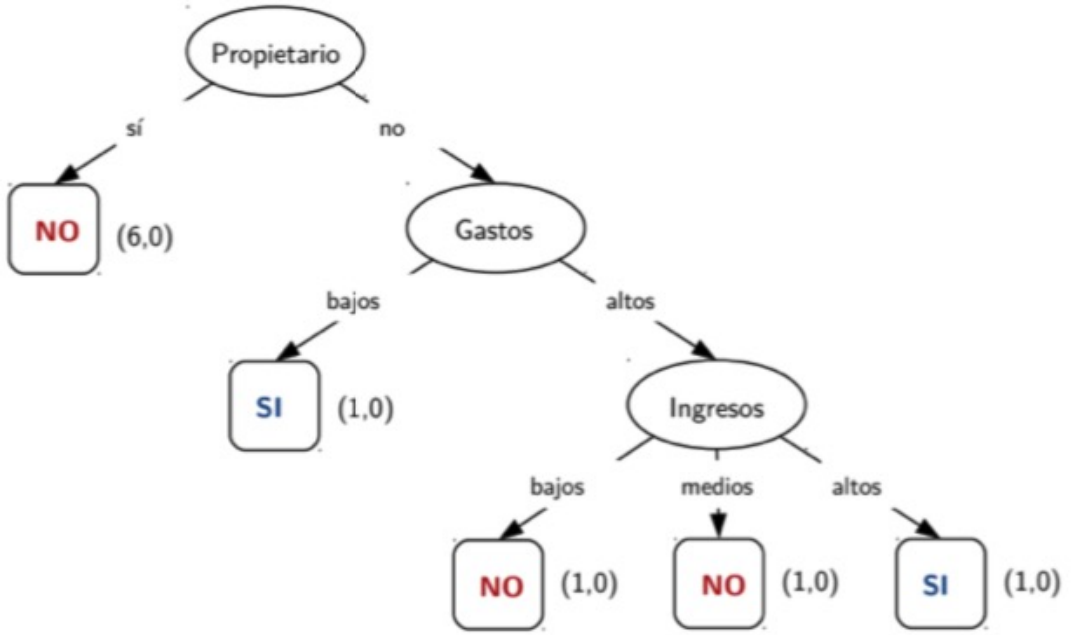


Construcción de un árbol
Selección de modelos

¿Qué árbol es mejor?

Si los dos tienen el mismo error entonces evalurá:
1) El que tenga menor número de hojas (6 vs. 5)
2) El tamaño medio del camino a las hojas sea menor





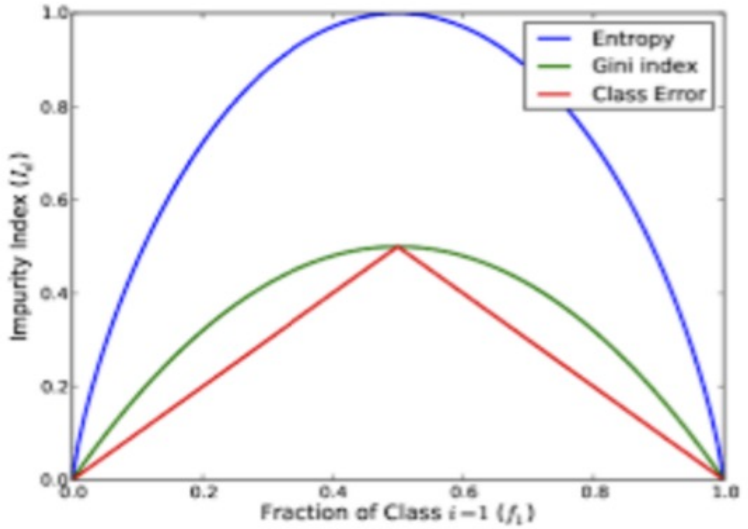
Ganancia de información

Criterios de selección

El objetivo es seleccionar una variable que conduzca a una partición **más pura/menos ruidosa** con respecto a la variable objetivo, es decir, que todos los objetos (casos) que estén en esa partición sea de la misma clase.

- **Entropía** (ID3, C4.5).

$$H(X) = -\sum_{i=1}^{L} P_i \cdot \log_2(P_i)$$



- **Indice Gini** (CART).

$$Gini(X) = 1 - \sum_{i=1}^{L} P_i^2$$

- **Error de clasificación**.

$$Err(X) = 1 - max(P_1, \ldots, P_L)$$

Criterios de selección

Por la incertidumbre que hay en la distribución de probabilidad

- **Entropía** (ID3, C4.5).

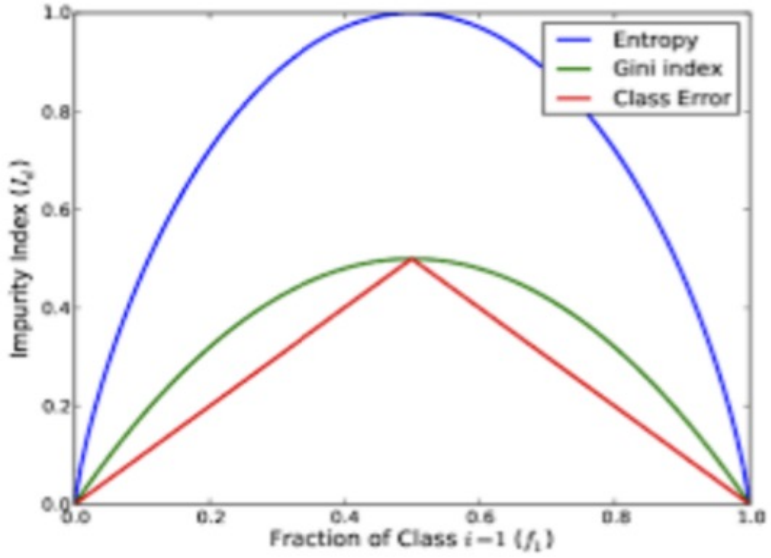
$$H(X) = -\sum_{i=1}^{L} P_i \cdot \log_2(P_i)$$

- **Indice Gini** (CART).

$$Gini(X) = 1 - \sum_{i=1}^{L} P_i^2$$

- **Error de clasificación**.

$$Err(X) = 1 - max(P_1, \ldots, P_L)$$

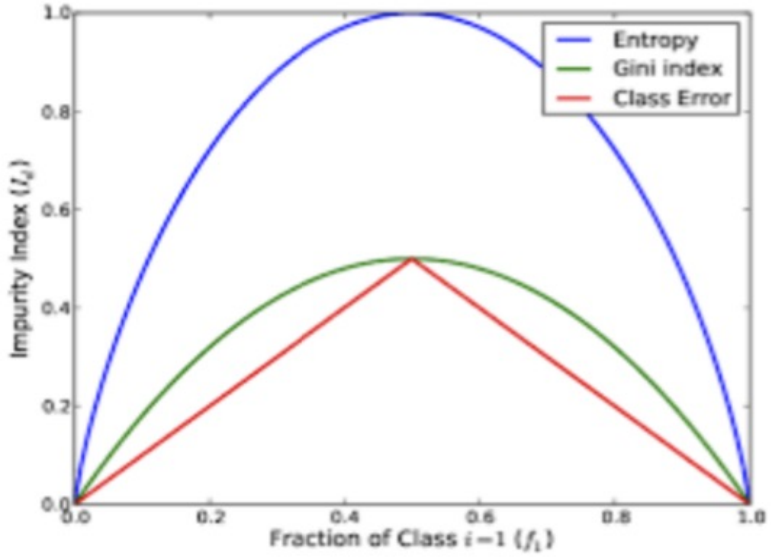


Criterios de selección

Parecido a Entropía
Más extremas tendrán menos ruido más cerca a 0,5 - 0,5 más ruido

- **Entropía** (ID3, C4.5).

$$H(X) = -\sum_{i=1}^{L} P_i \cdot \log_2(P_i)$$



- **Indice Gini** (CART).

$$Gini(X) = 1 - \sum_{i=1}^{L} P_i^2$$

- **Error de clasificación**.

$$Err(X) = 1 - max(P_1, \ldots, P_L)$$

Criterios de selección

Elegimos al que genere un menor error en la tasa de acierto

- **Entropía** (ID3, C4.5).

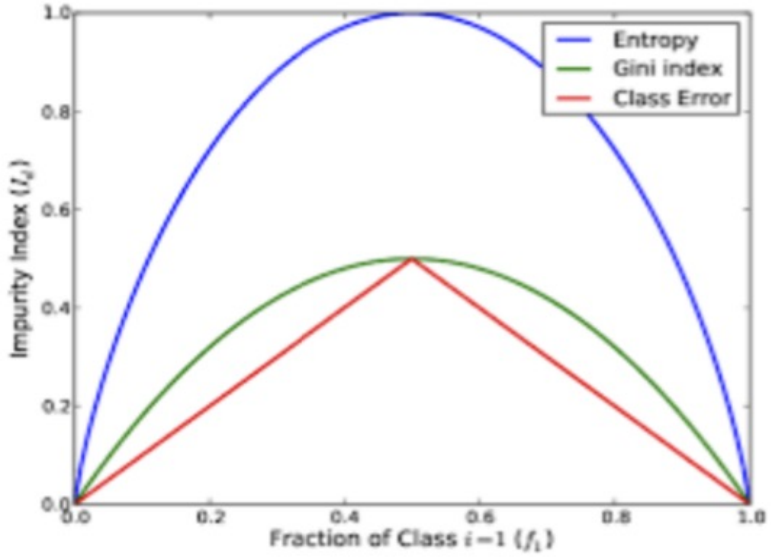
$$H(X) = -\sum_{i=1}^{L} P_i \cdot \log_2(P_i)$$

- **Indice Gini** (CART).

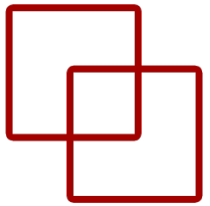
$$Gini(X) = 1 - \sum_{i=1}^{L} P_i^2$$

- **Error de clasificación**.

$$Err(X) = 1 - max(P_1, \ldots, P_L)$$



Ganancia de información



- Para cada variable candidata, calculamos la ganancia con respecto al criterio elegido. Seleccionamos la que maximiza la ganancia.

Ganancia de información

- Para cada variable candidata, calculamos la ganancia con respecto al criterio elegido. Seleccionamos la que maximiza la ganancia.
  - $\boldsymbol{D}$ se divide en $\boldsymbol{D^1_X}$,..., $\boldsymbol{D^r_X}$ usando la variable X. (Los datos los dividimos en *r* proyecciones, es decir, si tenemos 3 posibles valores dividimos los datos en 3 posibles ramas)

Ganancia de información

- Para cada variable candidata, calculamos la ganancia con respecto al criterio elegido. Seleccionamos la que maximiza la ganancia.
  - $\boldsymbol{D}$ se divide en $\boldsymbol{D^1_X}$,..., $\boldsymbol{D^r_X}$ usando la variable X. (Los datos los dividimos en *r* proyecciones, es decir, si tenemos 3 posibles valores dividimos los datos en 3 posibles ramas)
  - Calculamos el criterio correspondiente (H, Gini, Err) para la variable clase en el conjunto inicial $\boldsymbol{C(D)}$.

Ganancia de información

- Para cada variable candidata, calculamos la ganancia con respecto al criterio elegido. Seleccionamos la que maximiza la ganancia.
  - $\boldsymbol{D}$ se divide en $\boldsymbol{D^1_X}$,..., $\boldsymbol{D^r_X}$ usando la variable X. (Los datos los dividimos en *r* proyecciones, es decir, si tenemos 3 posibles valores dividimos los datos en 3 posibles ramas)
  - Calculamos el criterio correspondiente (H, Gini, Err) para la variable clase en el conjunto inicial $\boldsymbol{C(D)}$.
  - Calculamos para todos los subconjuntos resultantes de la partición $\boldsymbol{C(D^1_X)}$,..., $\boldsymbol{C(D^r_X)}$

Ganancia de información

- Para cada variable candidata, calculamos la ganancia con respecto al criterio elegido. Seleccionamos la que maximiza la ganancia.
  - $\boldsymbol{D}$ se divide en $\boldsymbol{D^1_X}$,..., $\boldsymbol{D^r_X}$ usando la variable X. (Los datos los dividimos en *r* proyecciones, es decir, si tenemos 3 posibles valores dividimos los datos en 3 posibles ramas)
  - Calculamos el criterio correspondiente (H, Gini, Err) para la variable clase en el conjunto inicial $\boldsymbol{C(D)}$.
  - Calculamos para todos los subconjuntos resultantes de la partición $\boldsymbol{C(D^1_X)}$,..., $\boldsymbol{C(D^r_X)}$
  - Calculamos para la partición promediando:

$$C(\mathbf{D}^1_X, \ldots, \mathbf{D}^r_X) = \sum_{i=1}^{r} \frac{|\mathbf{D}^i_X|}{|\mathbf{D}|} \cdot C(\mathbf{D}^i_X)$$

Ganancia de información

- Para cada variable candidata, calculamos la ganancia con respecto al criterio elegido. Seleccionamos la que maximiza la ganancia.
  - $\boldsymbol{D}$ se divide en $\boldsymbol{D^1_X}$,..., $\boldsymbol{D^r_X}$ usando la variable X. (Los datos los dividimos en *r* proyecciones, es decir, si tenemos 3 posibles valores dividimos los datos en 3 posibles ramas)
  - Calculamos el criterio correspondiente (H, Gini, Err) para la variable clase en el conjunto inicial $\boldsymbol{C(D)}$.
  - Calculamos para todos los subconjuntos resultantes de la partición $\boldsymbol{C(D^1_X)}$,..., $\boldsymbol{C(D^r_X)}$
  - Calculamos para la partición promediando:

$$C(\mathbf{D}^1_X, \ldots, \mathbf{D}^r_X) = \sum_{i=1}^{r} \frac{|\mathbf{D}^i_X|}{|\mathbf{D}|} \cdot C(\mathbf{D}^i_X)$$

  - Y obtenemos la ganancia (de los datos sin proyectar (C(D)) menos los datos proyectados (C(D^1_X,...,D^r_X)):

$$gain(X) = C(\mathbf{D}) - C(\mathbf{D}^1_X, \ldots, \mathbf{D}^r_X)$$

Ganancia de información
Resumen

- Para una variable X_i dividimos los datos D con respecto a sus valores de salida:

  $D\downarrow_{[Xi == vk]}$

- Calculamos la medida con respecto a cada partición $M(D\downarrow_{[Xi == vk]})$
- Calculamos el promedio de la partición:

$$M(\mathbf{D}^{\downarrow}) = \sum_{k=1}^{r} \frac{|\mathbf{D}^{\downarrow}_{[\mathbf{X_i}==\mathbf{v_k}]}|}{|\mathbf{D}|} M(\mathbf{D}^{\downarrow}_{[X_i==v_k]})$$

- Calculamos la ganancia con respecto a su nodo padre:

  $Gain(Xi) = M(D) - M(D\downarrow)$

División del árbol

Ganancia por entropía

D

R

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Entropía por Target → 5 No – 5 Si

$H(C) = 5/5 ; 5/5 – (5/5 \log(5/5) – 5/5 \log(5/5) =$

$= 1$

División del árbol

Ganancia por entropía

D *R*

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Entropía por Propietario

$H(C \mid Prop = Si) = (3Si\ ;\ 2No) =$

$= (3/5\ ;\ 2/5) =$

$= 1 - (3/5 \log(3/5) + 2/5 \log(2/5) = 0{,}7$

$H(C \mid Prop = No) = (3No\ ;\ 2Si) =$

$= (3/5\ ;\ 2/5) =$

$= 1 - (3/5 \log(3/5) + 2/5 \log(2/5) = 0{,}7$

División del árbol
Ganancia por entropía

D

R

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Entropía por Propietario (T → 5No / 5Si)
H(Prop) = 0,7 * 5/10 + 0,7 * 5/10 = 0,67
G (C | Prop) = 1 – 0,67 = 0,33

División del árbol
Ganancia por entropía

D

R

Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Entropía por Ingresos (T → 5No / 5Si)
G(C | Ingresos) = 1 – 0,1 = 0,9

División del árbol
Ganancia por entropía

D

R

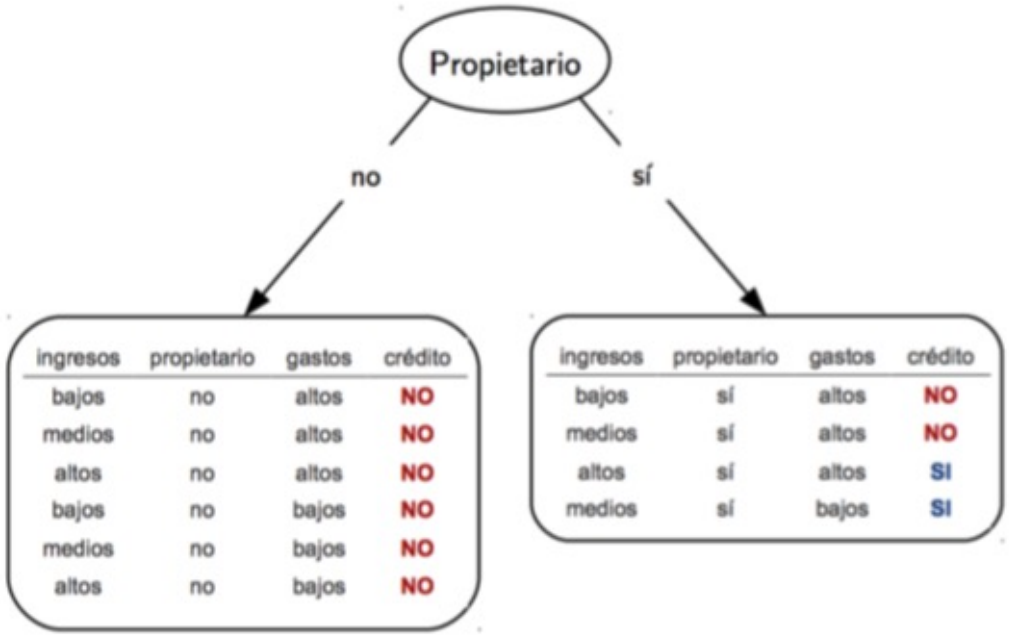
Ingresos	Propietario	Gastos	Crédito
Bajos	Si	Altos	No
Bajos	No	Bajos	No
Altos	No	Bajos	Si
Altos	Si	Altos	Si
Bajos	No	Altos	No
Bajos	Si	Altos	No
Altos	Si	Altos	Si
Altos	No	Bajos	Si
Altos	Si	Bajos	Si
Bajos	No	Bajos	No

Entropía por Ingresos (T → 5No / 5Si)
G(C | Ingresos) = 1 – 0,1 = 0,9

Se divide por ingresos ya que tiene mayor ganancia

Construcción: Ejemplo

ingresos	propietario	gastos	crédito
bajos	no	altos	NO
bajos	si	altos	NO
medios	si	altos	NO
medios	no	altos	NO
altos	no	altos	NO
altos	si	altos	SI
bajos	no	bajos	NO
medios	no	bajos	NO
altos	no	bajos	NO
medios	si	bajos	SI



$$H(crédito) = 0.72$$

$$\textbf{Gain(propietario)} = \textbf{0.32}$$

$$Gain(ingresos) = 0.12$$

$$Gain(gastos) = 0.01$$

$$(\text{propietario} = \text{no}) \Longrightarrow (\text{crédito} = \textbf{NO})$$

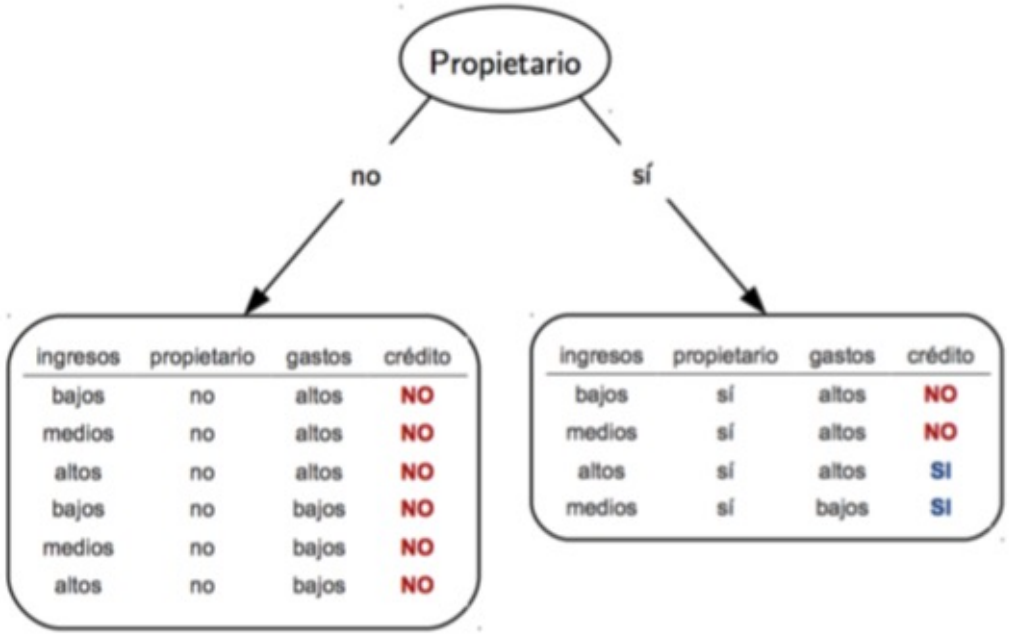
$$(\text{propietario} = \text{sí}) \Longrightarrow (\text{crédito} = \textbf{?})$$

Se puede expandir la rama recursivamente

Construcción: Ejemplo

ingresos	propietario	gastos	crédito
bajos	no	altos	NO
bajos	si	altos	NO
medios	si	altos	NO
medios	no	altos	NO
altos	no	altos	NO
altos	si	altos	SI
bajos	no	bajos	NO
medios	no	bajos	NO
altos	no	bajos	NO
medios	si	bajos	SI

Como tenemos más Atributos y propietario tiene en sí las dos clases posibles tenemos que seguir expandiendo



Elegimos propietario porque es el que tiene mayor ganancia de información

$$H(crédito) = 0.72$$

$$\mathbf{Gain(propietario) = 0.32}$$

$$Gain(ingresos) = 0.12$$

$$Gain(gastos) = 0.01$$

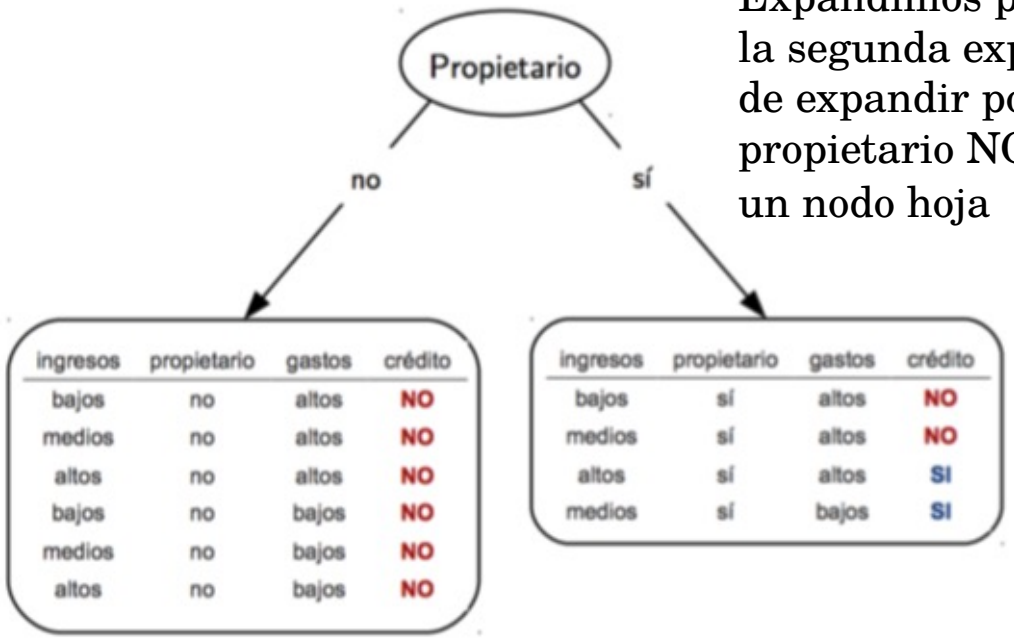
$$(\text{propietario} = \text{no}) \Longrightarrow (\text{crédito} = \mathbf{NO})$$

$$(\text{propietario} = \text{sí}) \Longrightarrow (\text{crédito} = ?)$$

Se puede expandir la rama recursivamente

Construcción: Ejemplo

ingresos	propietario	gastos	crédito
bajos	no	altos	NO
bajos	si	altos	NO
medios	si	altos	NO
medios	no	altos	NO
altos	no	altos	NO
altos	si	altos	SI
bajos	no	bajos	NO
medios	no	bajos	NO
altos	no	bajos	NO
medios	si	bajos	SI



Expandimos propietario y en la segunda expansión pararía de expandir por la rama propietario NO, es decir sería un nodo hoja

$$H(\textit{crédito}) = 0.72$$

$$\textbf{Gain(propietario)} = \textbf{0.32}$$

$$\textit{Gain}(\textit{ingresos}) = 0.12$$

$$\textit{Gain}(\textit{gastos}) = 0.01$$

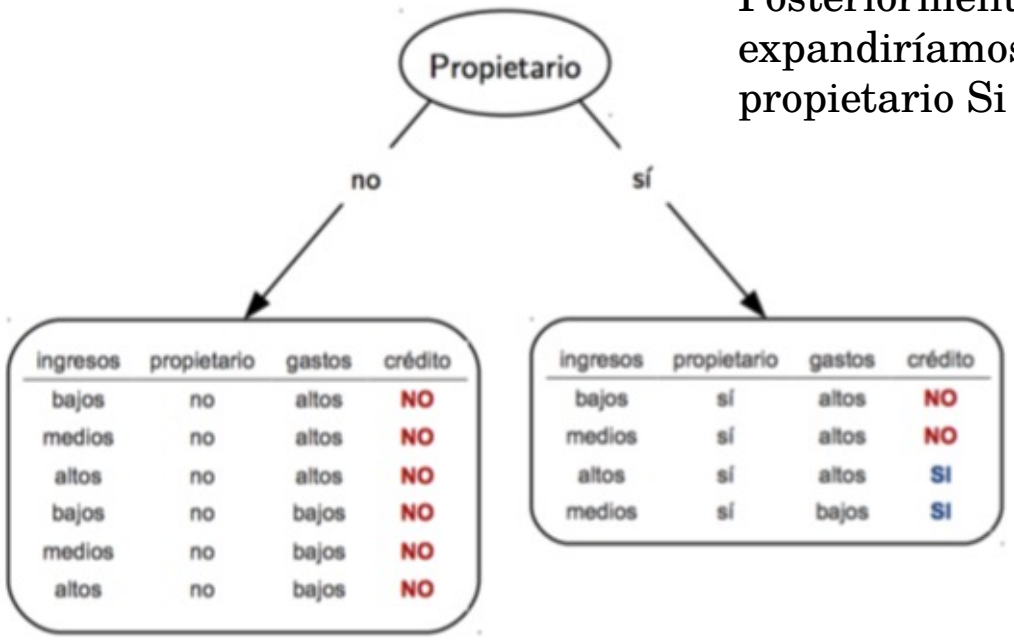
$$(\text{propietario} = \text{no}) \Longrightarrow (\text{crédito} = \textbf{NO})$$

$$(\text{propietario} = \text{sí}) \Longrightarrow (\text{crédito} = \textbf{?})$$

Se puede expandir la rama recursivamente

Construcción: Ejemplo

ingresos	propietario	gastos	crédito
bajos	no	altos	NO
bajos	si	altos	NO
medios	si	altos	NO
medios	no	altos	NO
altos	no	altos	NO
altos	si	altos	SI
bajos	no	bajos	NO
medios	no	bajos	NO
altos	no	bajos	NO
medios	si	bajos	SI



Posteriormente expandiríamos el nodo propietario Si

$$H(crédito) = 0.72$$

$$\textbf{Gain(propietario)} = \textbf{0.32}$$

$$Gain(ingresos) = 0.12$$

$$Gain(gastos) = 0.01$$

$$(\text{propietario} = \text{no}) \Longrightarrow (\text{crédito} = \textbf{NO})$$

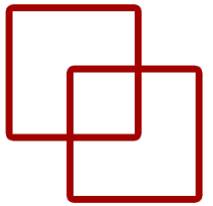
$$(\text{propietario} = \text{sí}) \Longrightarrow (\text{crédito} = \textbf{?})$$

Se puede expandir la rama recursivamente

Variables continuas

- Las variables discretas solo aparecen una sola vez en el camino desde la raíz.

Variables continuas



- Las variables discretas solo aparecen una sola vez en el camino desde la raíz.
- Las variables continuas se deben ir **particionando en intervalos a diferentes profundidades** del árbol.

Variables continuas

- Las variables discretas solo aparecen una sola vez en el camino desde la raíz.
- Las variables continuas se deben ir **particionando en intervalos a diferentes profundidades** del árbol.
- Se ordena la característica de **menor a mayor y para cada umbral** se calcula la ganancia de información.

Variables continuas

- Las variables discretas solo aparecen una sola vez en el camino desde la raíz.
- Las variables continuas se deben ir **particionando en intervalos a diferentes profundidades** del árbol.
- Se ordena la característica de **menor a mayor y para cada umbral** se calcula la ganancia de información.
- Nos quedamos con el umbral que nos dé mayor ganancia de información

Variables continuas

- Las variables discretas solo aparecen una sola vez en el camino desde la raíz.
- Las variables continuas se deben ir **particionando en intervalos a diferentes profundidades** del árbol.
- Se ordena la característica de **menor a mayor y para cada umbral** se calcula la ganancia de información.
- Nos quedamos con el umbral que nos dé mayor ganancia de información
- Ejemplo:

Teoría	Prácticas	Aprobado
2.5	regulares	no
3	malas	no
4	regulares	no
5	malas	no
5	buenas	si
6	regulares	si
7.5	buenas	si
7.5	malas	no
9	buenas	si
9.5	regulares	si

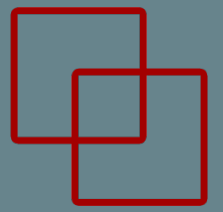
$umbral = 3$

$\Longrightarrow$

Teoría	Prácticas	Aprobado
bajo	regulares	no
bajo	malas	no
alto	regulares	no
alto	malas	no
alto	buenas	si
alto	regulares	si
alto	buenas	si
alto	malas	no
alto	buenas	si
alto	regulares	si

La entropía condicionada al atributo teoría, utilizando *umbral = 3*

$$H(Aprobado|teoria\,[umbral = 3]) = \frac{2}{10} \cdot H(2,0) + \frac{8}{10} \cdot H(3,5) = 0.76$$



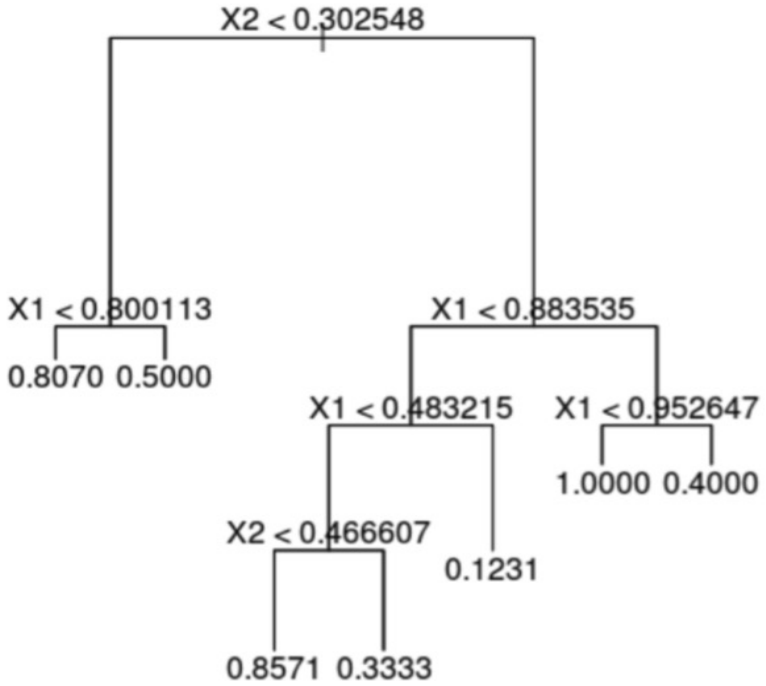
UNED

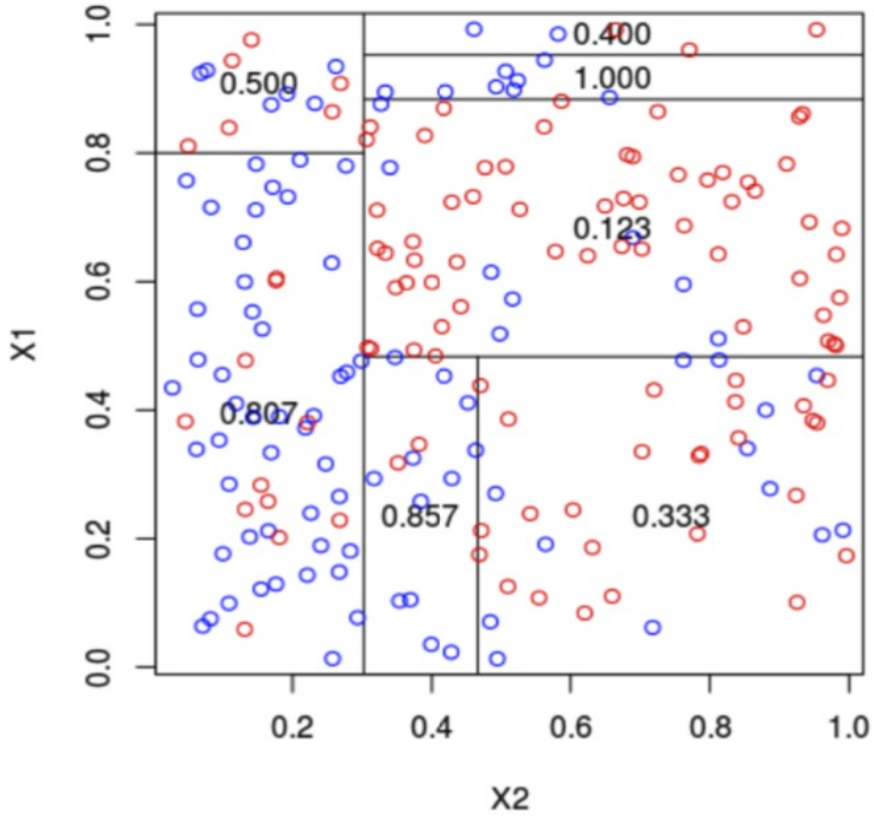
ETS de Ingeniería Informática

Árboles de regresión

Árboles de Regresión

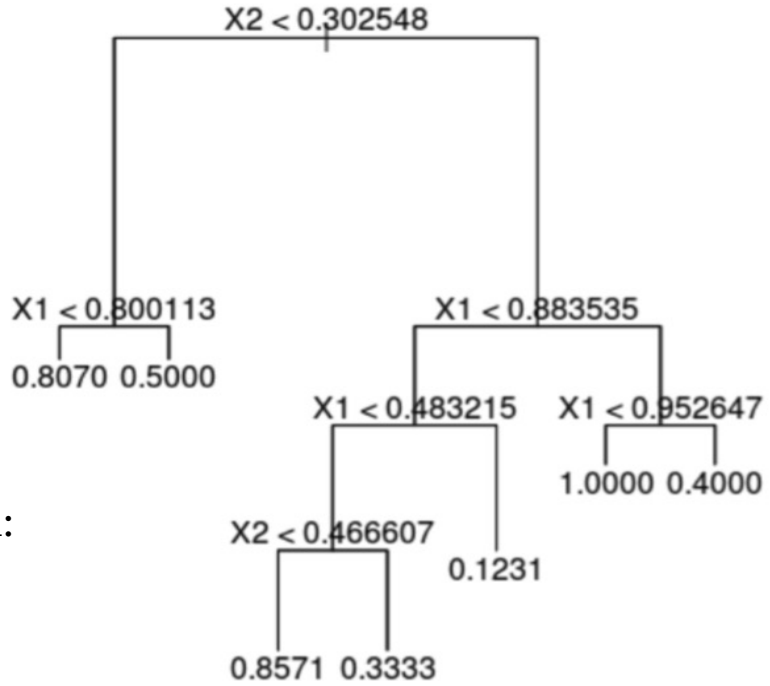
- La estructura es idéntica a un árbol de decisión, pero ahora las hojas contienen un valor numérico, **normalmente** la media para la variable objetivo en todos los registros que llegan al hipercubo correspondiente.

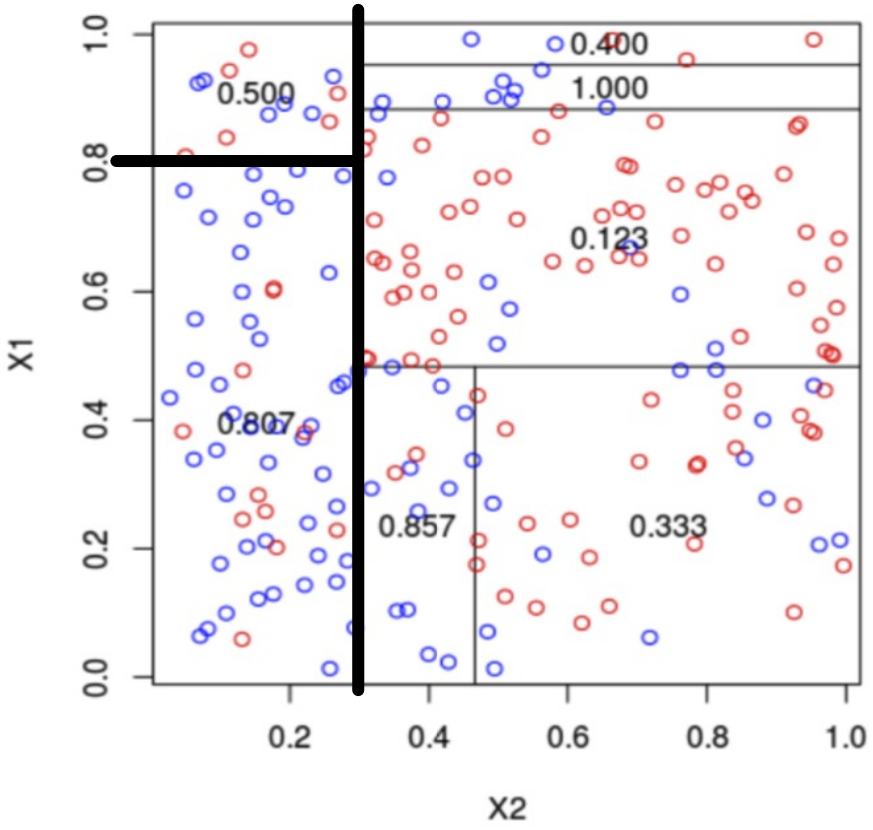




Árboles de Regresión

- La estructura es idéntica a un árbol de decisión, pero ahora las hojas contienen un valor numérico, **normalmente** la media para la variable objetivo en todos los registros que llegan al hipercubo correspondiente.

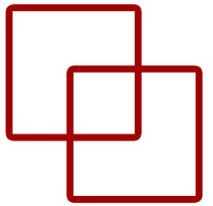




Las hojas las puede calcular con:
1) Medias de las muestras
2) Utilizando modelos de LiR

Árboles de Regresión

Aprendizaje



- **Criterio de ramificación**: Elegir el punto de corte tal que minimize los errores en las regiones inducidas por el corte.

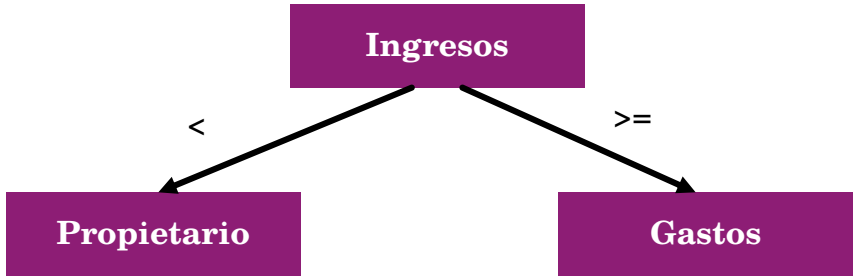
$$\sum_{i:X_i<t} (y^{(i)} - \bar{y}_{R_{<t}})^2 + \sum_{i:X_i\geq t} (y^{(i)} - \bar{y}_{R_{\geq t}})^2$$

- Otras opciones es minimizar la varianza de los datos de las regiones inducidas.

Árboles de Regresión

Aprendizaje

- **Criterio de ramificación**: Elegir el punto de corte tal que minimize los errores en las regiones inducidas por el corte.



- Otras opciones es minimizar la varianza de los datos de las regiones inducidas.

$$\sum_{i:X_i<t} (y^{(i)} - \bar{y}_{R_{<t}})^2 + \sum_{i:X_i\geq t} (y^{(i)} - \bar{y}_{R_{\geq t}})^2$$

Selección de modelos

- Dividiremos el espacio de soluciones en un conjunto de regiones $\{R_1, R_2, \ldots, R_m\}$, distintas y sin solapamiento t.q. para todo ejemplo que caiga en una región devolveremos el mismo valor, la media de los ejemplos del conjunto de training que caen en dicha región: $\bar{y}_{R_j}$
- Objetivo: encontrar el conjunto de regiones $\{R_1, R_2, \ldots, R_m\}$ que minimice el Error Cuadrático Medio (ECM):

$$\sum_{j=1}^{m} \sum_{i \in R_j} (y^{(i)} - \bar{y}_{R_j})^2$$

- **Diferencia** respecto a clasificación: **criterio para elegir la variable** y el umbral en cada nodo.
  - Dado un nodo concreto a ramificar, se elegirá la variable X_j y el umbral t en su dominio, t.q. se minimice la siguiente expresión (RSS):

$$\sum_{i:x_j^{(i)} < t} (y^{(i)} - \bar{y}_{R_{<t}})^2 + \sum_{i:x_j^{(i)} \geq t} (y^{(i)} - \bar{y}_{R_{\geq t}})^2$$

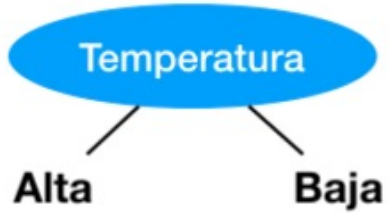
  - Este es el criterio en CART (Breiman). Computacionalmente el proceso es similar al de tratar atributos numéricos en C4.5
  - Otra opción es minimizar la varianza o la desviación estándar resultante.

Selección de modelos

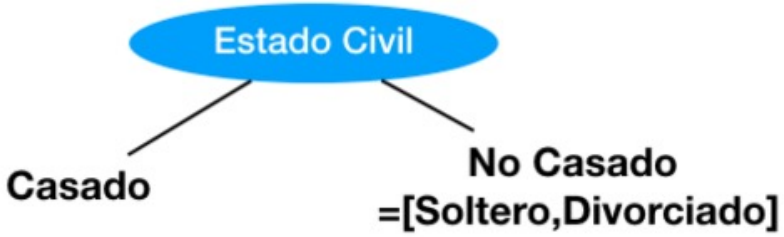
Ramificación de atributos

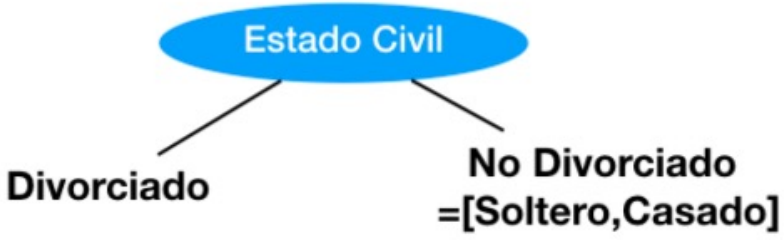
Atributos nominales

- **Atributo binario**: Se generan dos ramas.



- **Atributo nominal**: Se pueden expresar directamente mediante una rama por cada valor o binarizando el atributo

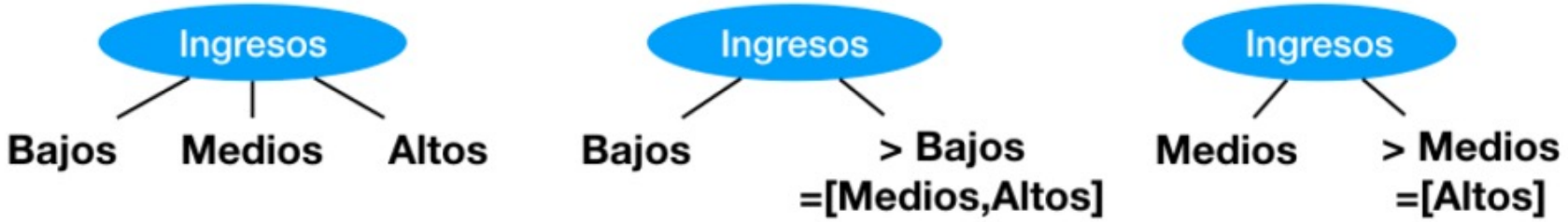




Ramificación de atributos

Atributos ordinarios

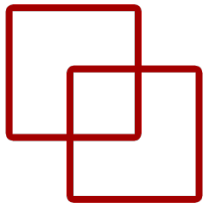
- **Atributo Ordinal**: Se pueden expresar directamente mediante una rama por cada valor o binarizando el atributo, pero siguiendo el orden.



- **Atributo continuo**: Se pueden expresar binarizando el atributo mediante un umbral, o discretizando mediante un rango de umbrales

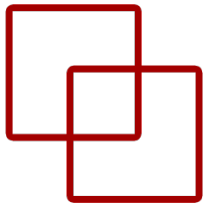
Decisión ramificación

Corrección del sesgo



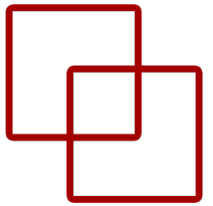
- El algoritmo propuesto ramifica las variables no continuas en tantas ramas como valores tiene.

Decisión ramificación
Corrección del sesgo



- El algoritmo propuesto ramifica las variables no continuas en tantas ramas como valores tiene.
- Tanto en Entropia como Gini existe un sesgo hacia las variables con más valores.
  - Solución C4.5. Ganancia de Información “normalizada”.
    - Divide la Ganancia del atributo obtenida por la Entropia del atributo que se quiere dividir → $G = G(X_i) / H(X_i)$
    - Por tanto, penaliza aquellos atributos que tienen más salidas.
  - Solución CART. Se crea un árbol binario.
    - Siempre se particiona por variable binarias (ya sea continua o nominal)

Evitar el sobreajuste



- ID3 →Se expande hasta el final; C4.5 hace poda recursiva
- **Poda** del árbol.

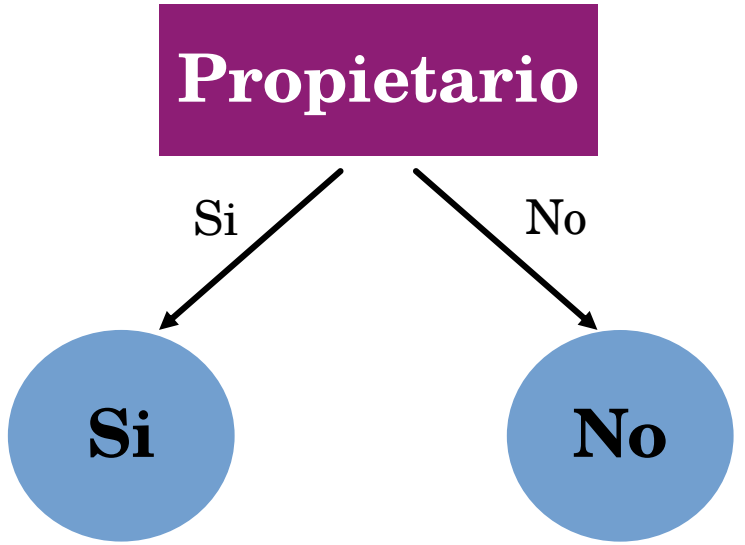
$$E(N) = \frac{N - n + k - 1}{N + k}$$

- Evitar ramificar hasta el final:
  - Exigiendo un **mínimo** número de ejemplos para seguir ramificando.
  - Imponiendo un **límite en la profundidad** del árbol.

Evitar el sobreajuste

Ejemplo de Poda

$$E(N) = \frac{N - n + k - 1}{N + k}$$



Han caído muestras:
(5Si, 2No)

Han caído muestras:
(6No, 2Sí)

E(Si) = (6 - 5 + 2 - 2) / (6 + 2)

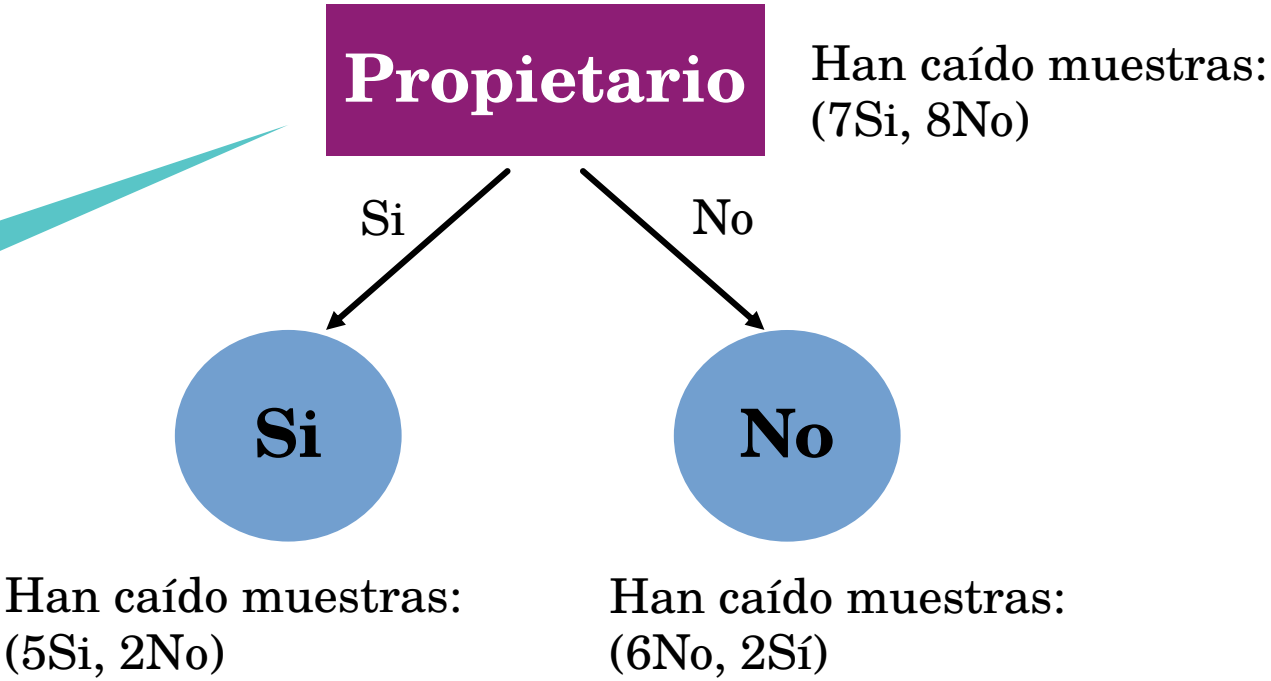
E(No) = (5 - 6 + 2 - 2) / (6 + 2)

Evitar el sobreajuste

Ejemplo de Poda

$$E(N) = \frac{N - n + k - 1}{N + k}$$

Evaluamos que sea Nodo hoja como hay más No evaluamos con E(No)



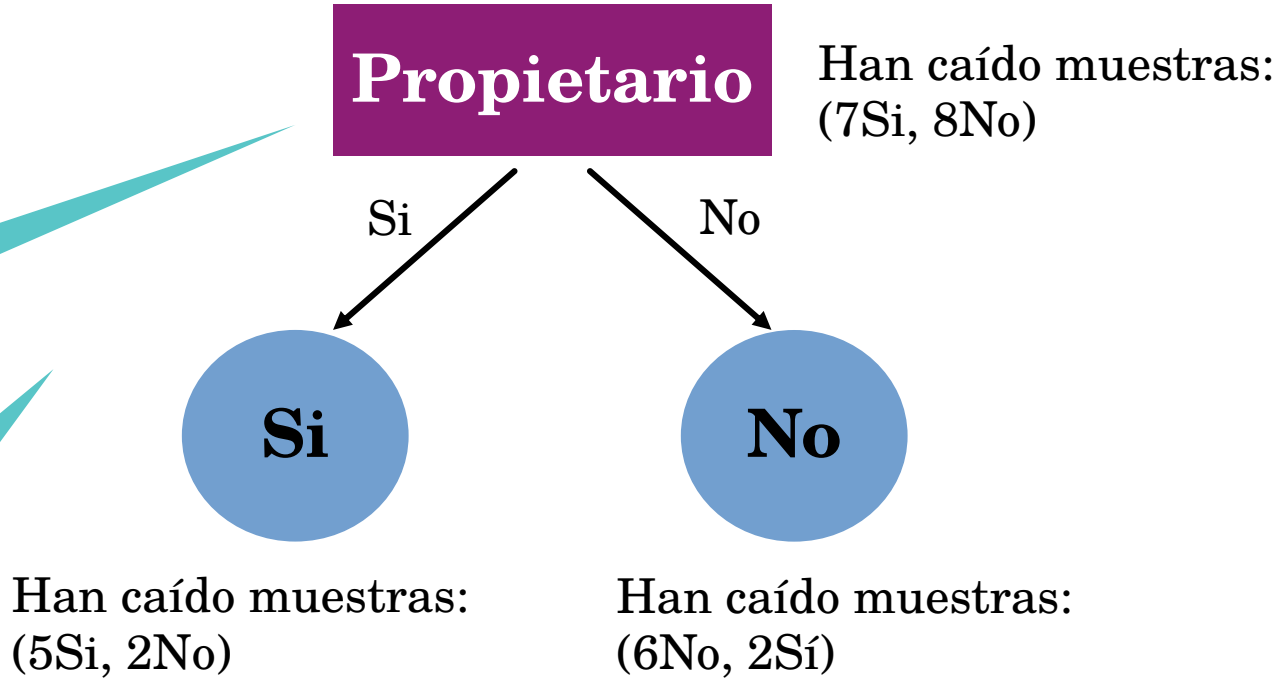
Evitar el sobreajuste

Ejemplo de Poda

$$E(N) = \frac{N - n + k - 1}{N + k}$$

Evaluamos que sea Nodo hoja como hay más No evaluamos con E(No)

Calculamos con No aquí y Si el error es menor se poda



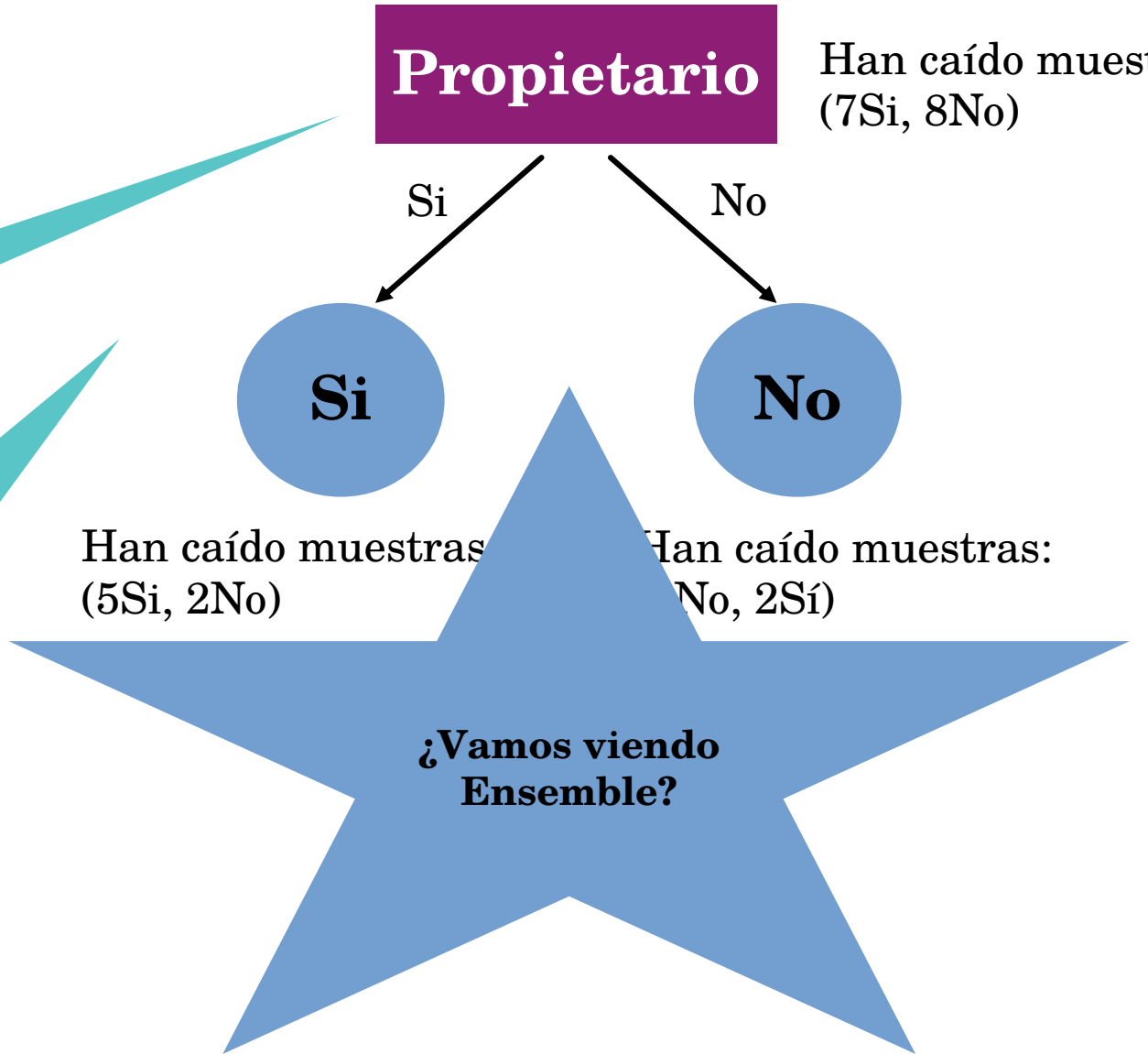
Evitar el sobreajuste

Ejemplo de Poda

$$E(N) = \frac{N - n + k - 1}{N + k}$$

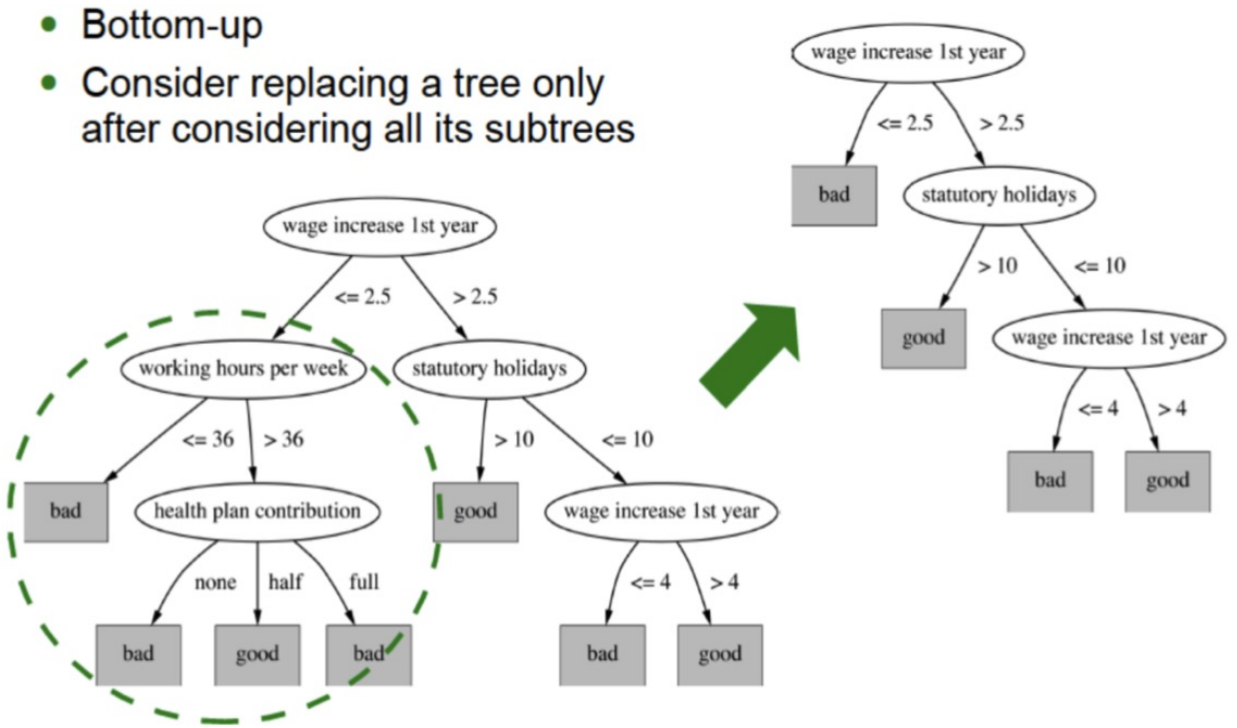
Evaluamos que sea Nodo hoja como hay más No evaluamos con E(No)

Calculamos con No aquí y Si el error es menor se poda



Sobreajuste
Poda en *c4.5*

- En *c4.5* hay dos operaciones de poda para luego sustituir subárboles por hojas:
  - **Subtree-replacement**: un subárbol se sustituye por una hoja por la clase mayoritaria calculando el error estimado.
  - **Subtree-raising**: un subárbol sustituye a otro.



En este caso se escoge la mayoritaria.
Ojo: Esto ocurre cuando el error estimado después de la poda es menor que sin podar

¡Gracias!

Manuel Castillo-Cara, Luis Sarro
www.manuelcastillo.eu
Department of Artificial Intelligence
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia (UNED)