

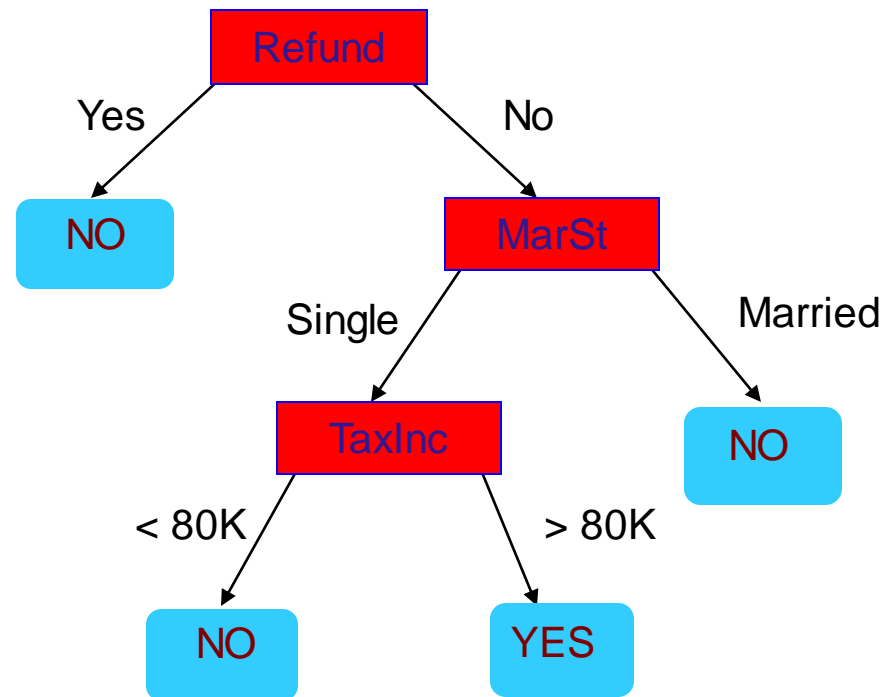
ÁRBOLES DE DECISIÓN

Definición de árboles de decisión para clasificación

- Un árbol de decisión es un clasificador que en función de un conjunto de atributos permite determinar a que clase pertenece el caso objeto de estudio
- La **estructura** de un árbol de decisión es:
 - ▣ Cada **hoja** es una **categoría (clase)** de la atributo objeto de la clasificación
 - ▣ Cada **nodo** es un nodo de decisión que **especifica una prueba/comprobación/test simple a realizar** (atributo)
 - ▣ Los **descendientes de cada nodo son los posibles resultados** (valores) de la prueba del nodo

Ejemplo de árbol de decisión

- Problema: en base a 3 variables de entrada (Refund, MarSt y TaxInc) predecir el valor de otra con dos clases (Cheat: yes o no)
- Árbol de decisión

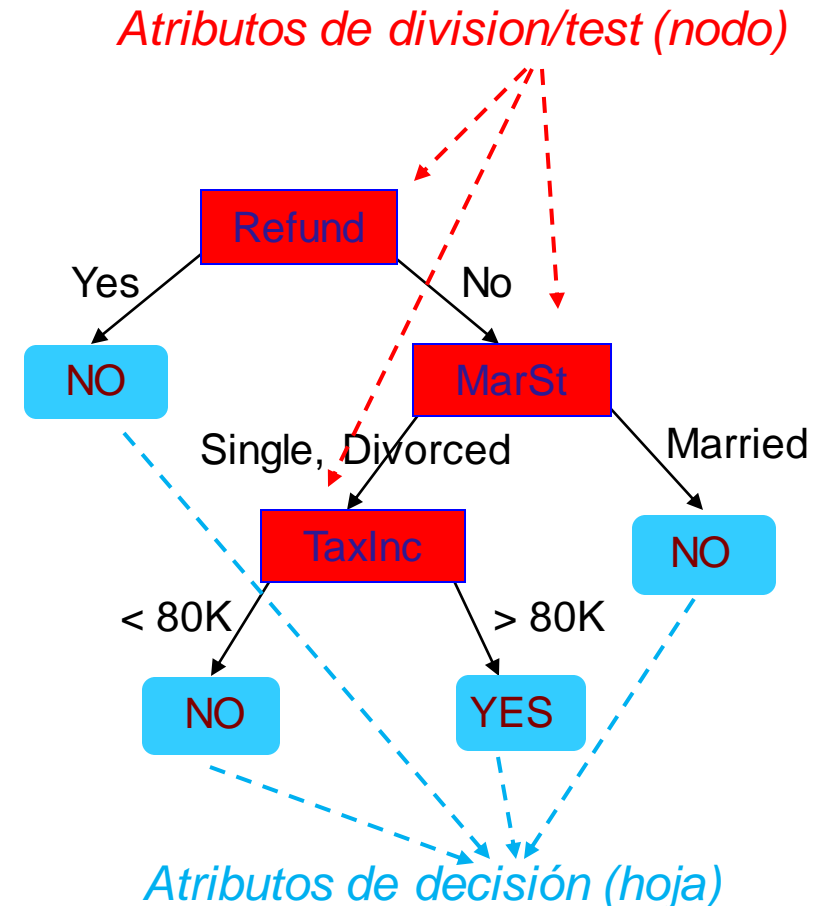


Ejemplo de árbol de decisión

discreto discreto continuo clase

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

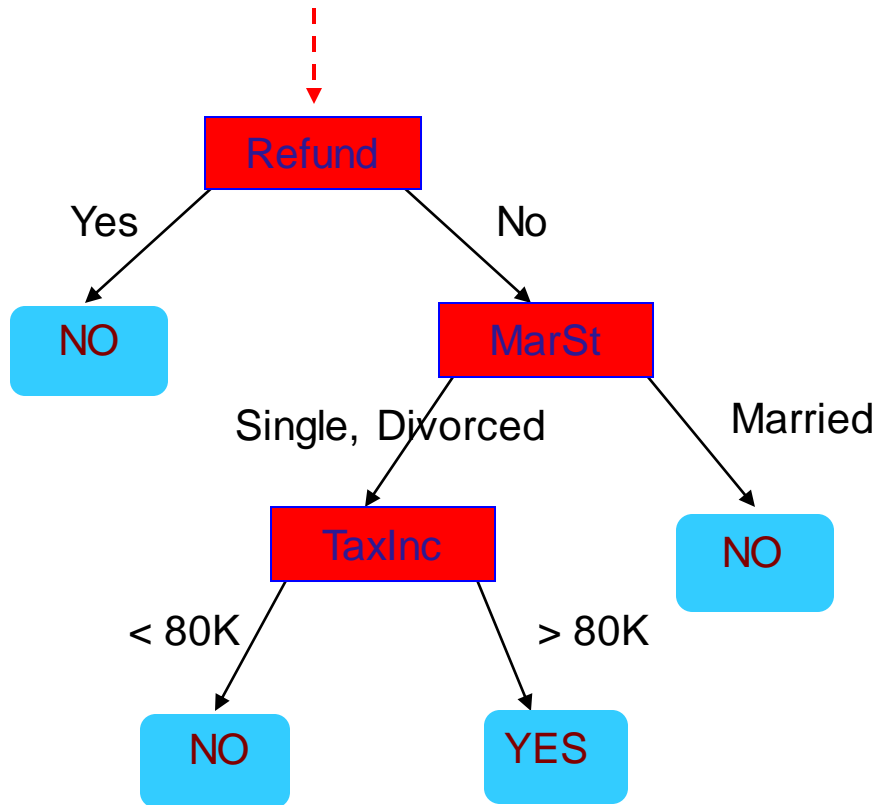
Datos de entrenamiento



Modelo: Árbol de decisión

¿Cómo aplicar el modelo?

Empezar del nodo raíz



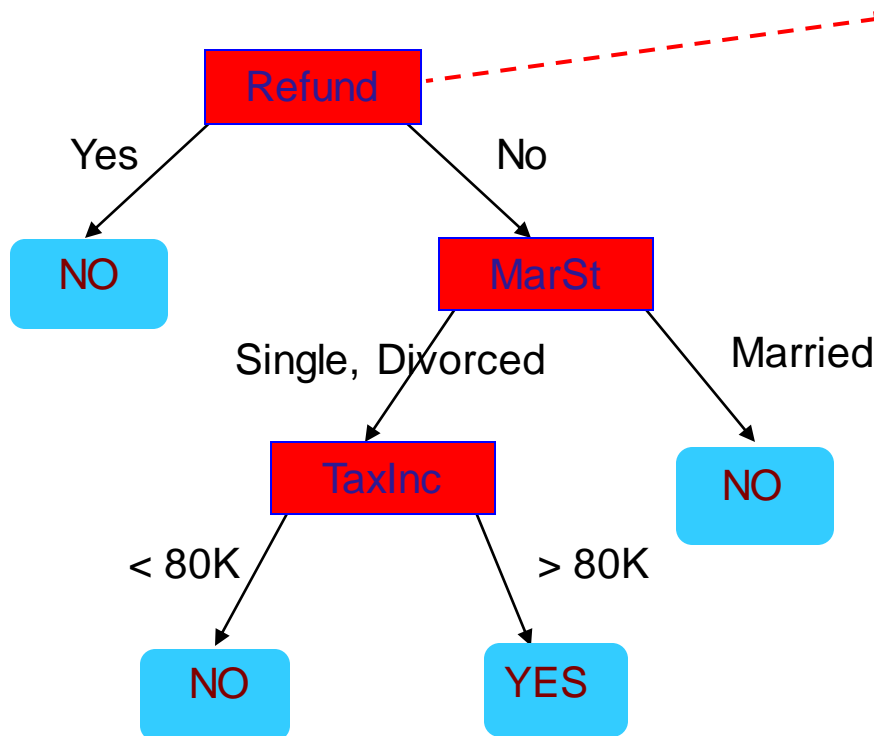
Ejemplo a clasificar

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

¿Cómo aplicar el modelo?

Ejemplo a clasificar

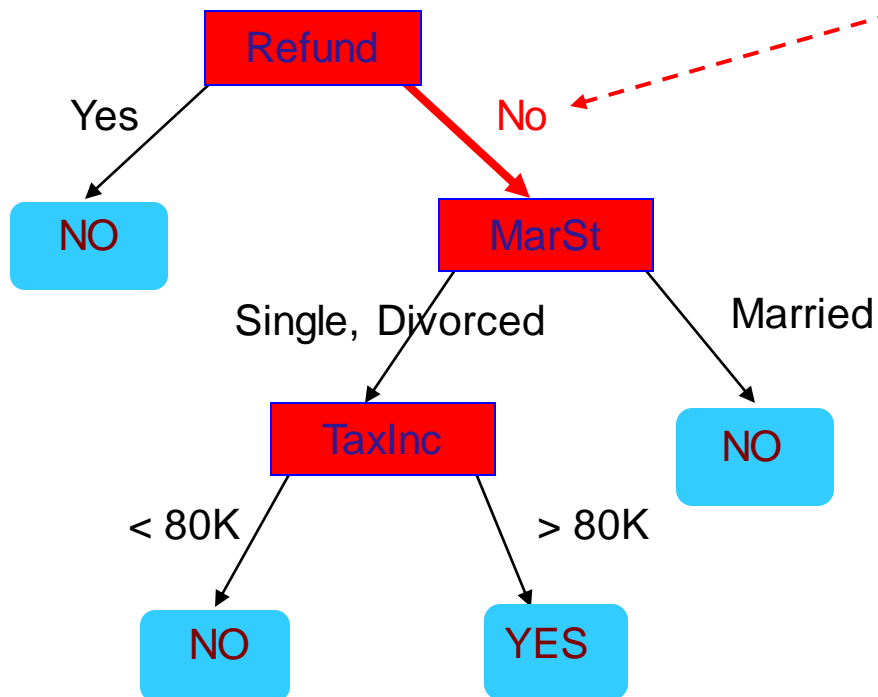
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



¿Cómo aplicar el modelo?

Ejemplo a clasificar

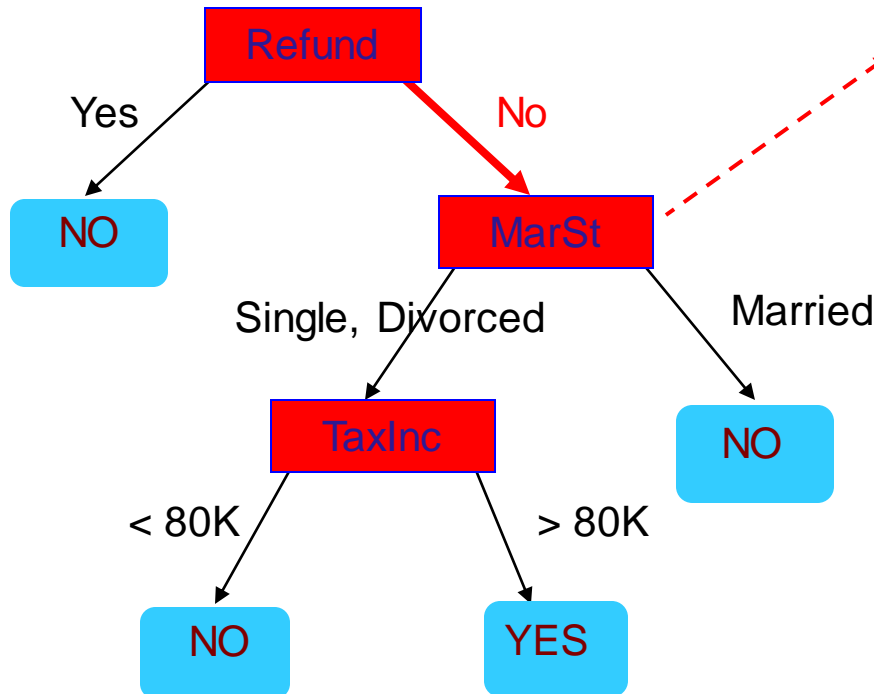
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



¿Cómo aplicar el modelo?

Ejemplo a clasificar

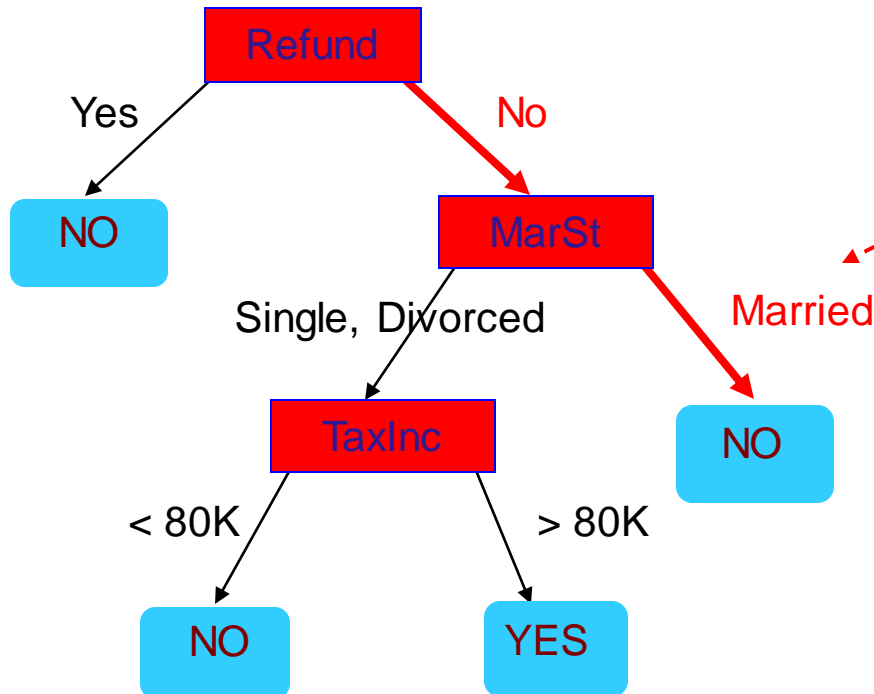
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



¿Cómo aplicar el modelo?

Ejemplo a clasificar

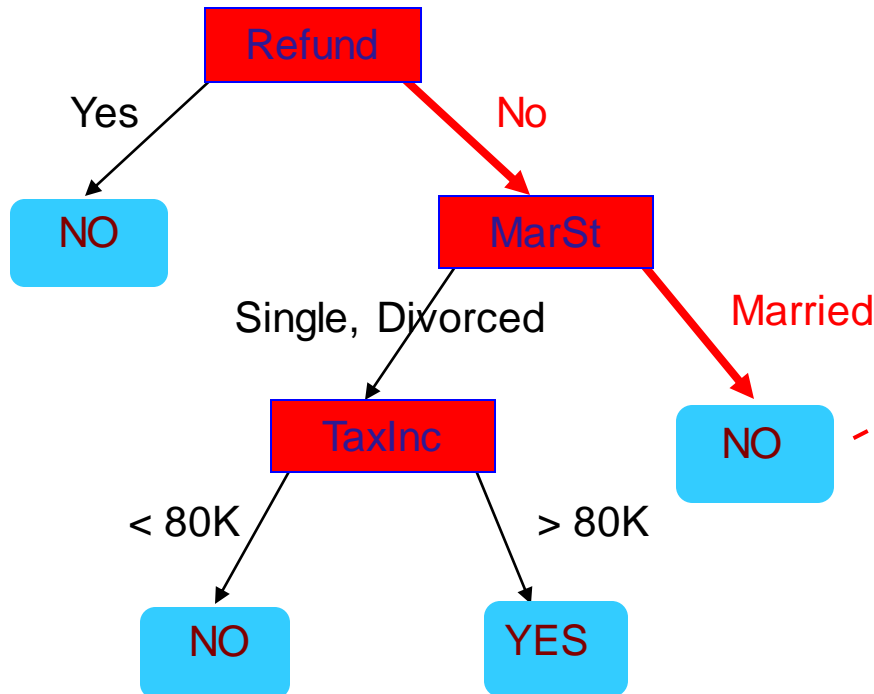
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



¿Cómo aplicar el modelo?

Ejemplo a clasificar

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Asignar Cheat
como "No"

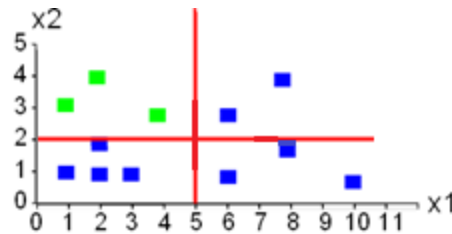
Clasificar el ejemplo
en la clase "No"

Aprendizaje del árbol de decisión

- **Algoritmo básico de aprendizaje** (algoritmo voraz)
 - ▣ Los atributos son categóricos
 - Si son continuos, se discretizan previamente
 - ▣ Se construye el árbol mediante la técnica **divide y vencerás aplicada de forma recursiva**
 - Al principio todos los ejemplos de entrenamiento están en el nodo raíz
 - Los **atributos de test (nodo interno)** se seleccionan en base a una medida heurística o estadística
 - Por ejemplo, la ganancia de información
 - Los **ejemplos se dividen recursivamente basándose en los atributos seleccionados**

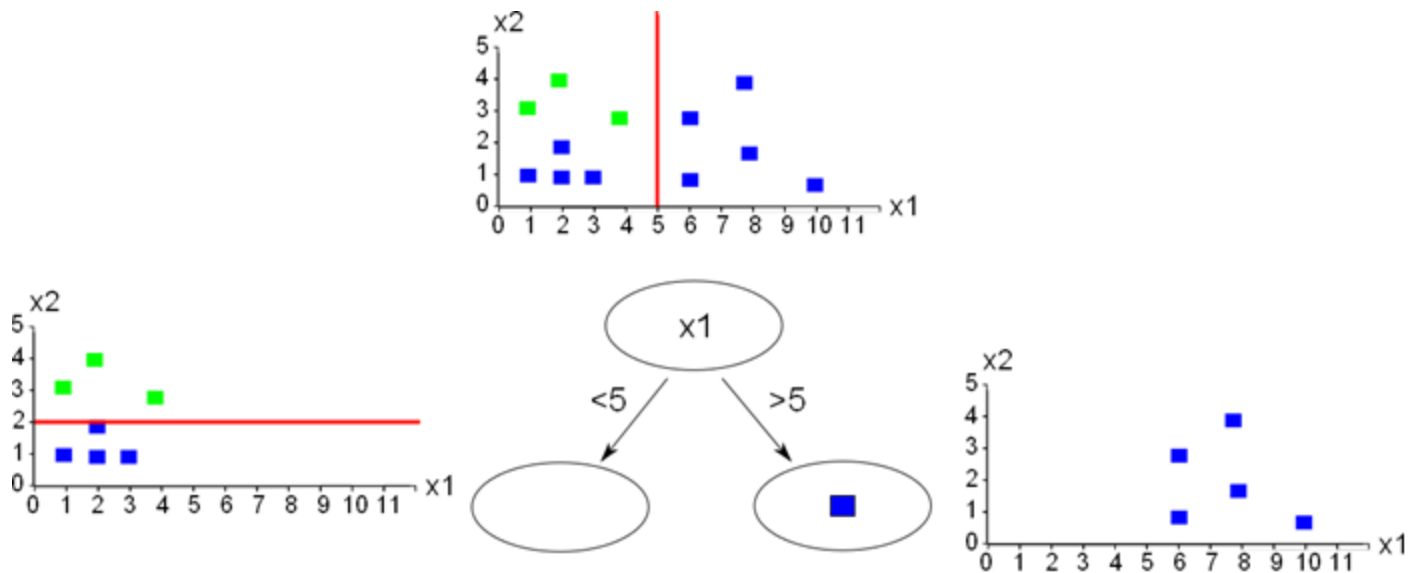
Aprendizaje del árbol de decisión

- Particionado recursivo del espacio de ejemplos



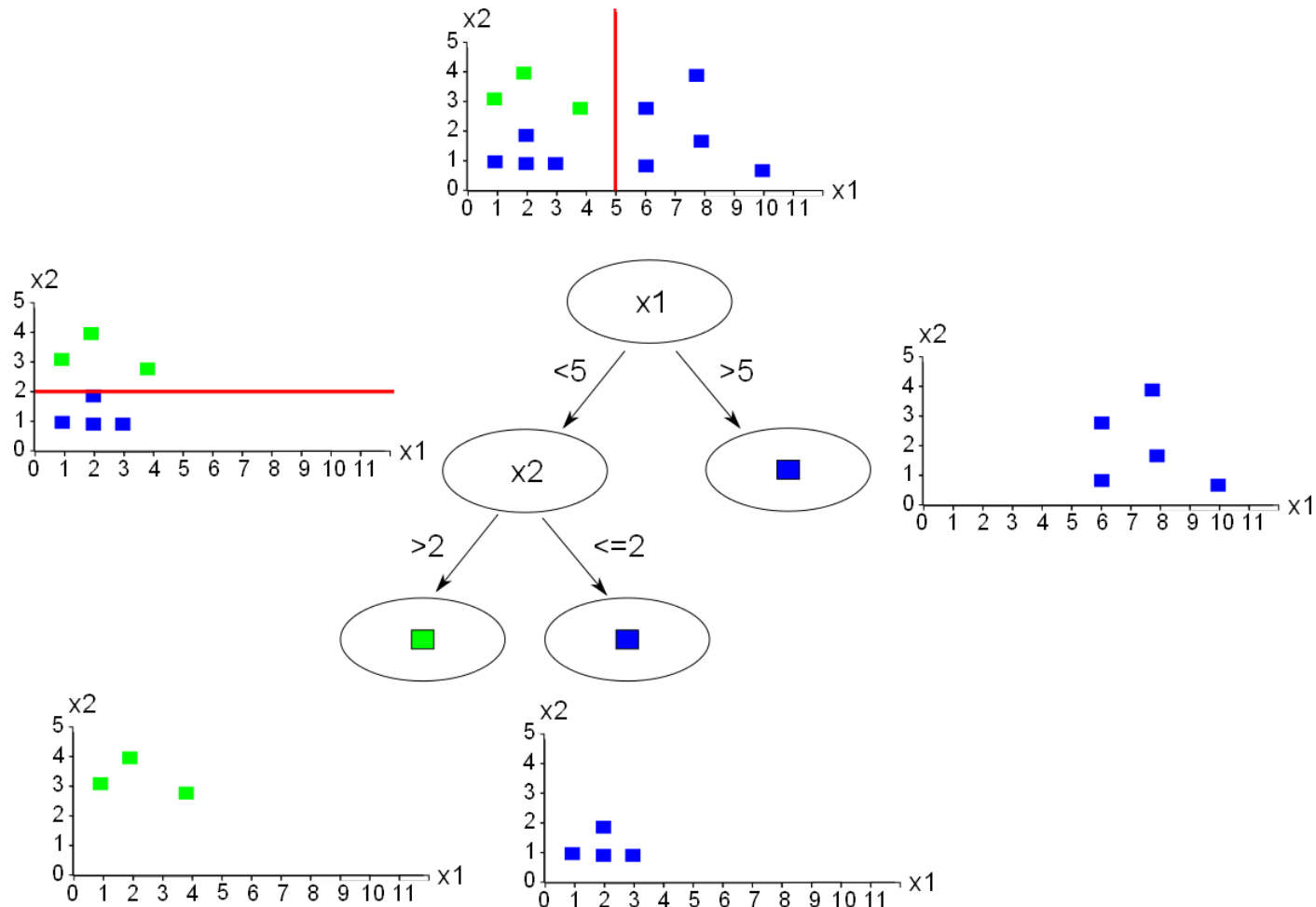
Aprendizaje del árbol de decisión

□ Particionado recursivo del espacio de ejemplos



Aprendizaje del árbol de decisión

□ Particionado recursivo del espacio de ejemplos



Aprendizaje del árbol de decisión

- Condiciones para terminar el particionamiento
 - ▣ Todos los ejemplos para un nodo dado pertenecen a la misma clase
 - Se crea una hoja etiquetada con la clase
 - ▣ No quedan más atributos para seguir particionando
 - Se crea una hoja etiquetada con la clase mayoritaria del nodo
 - ▣ No quedan ejemplos
 - No se crea hoja

Ejemplo aprendizaje árbol decisión

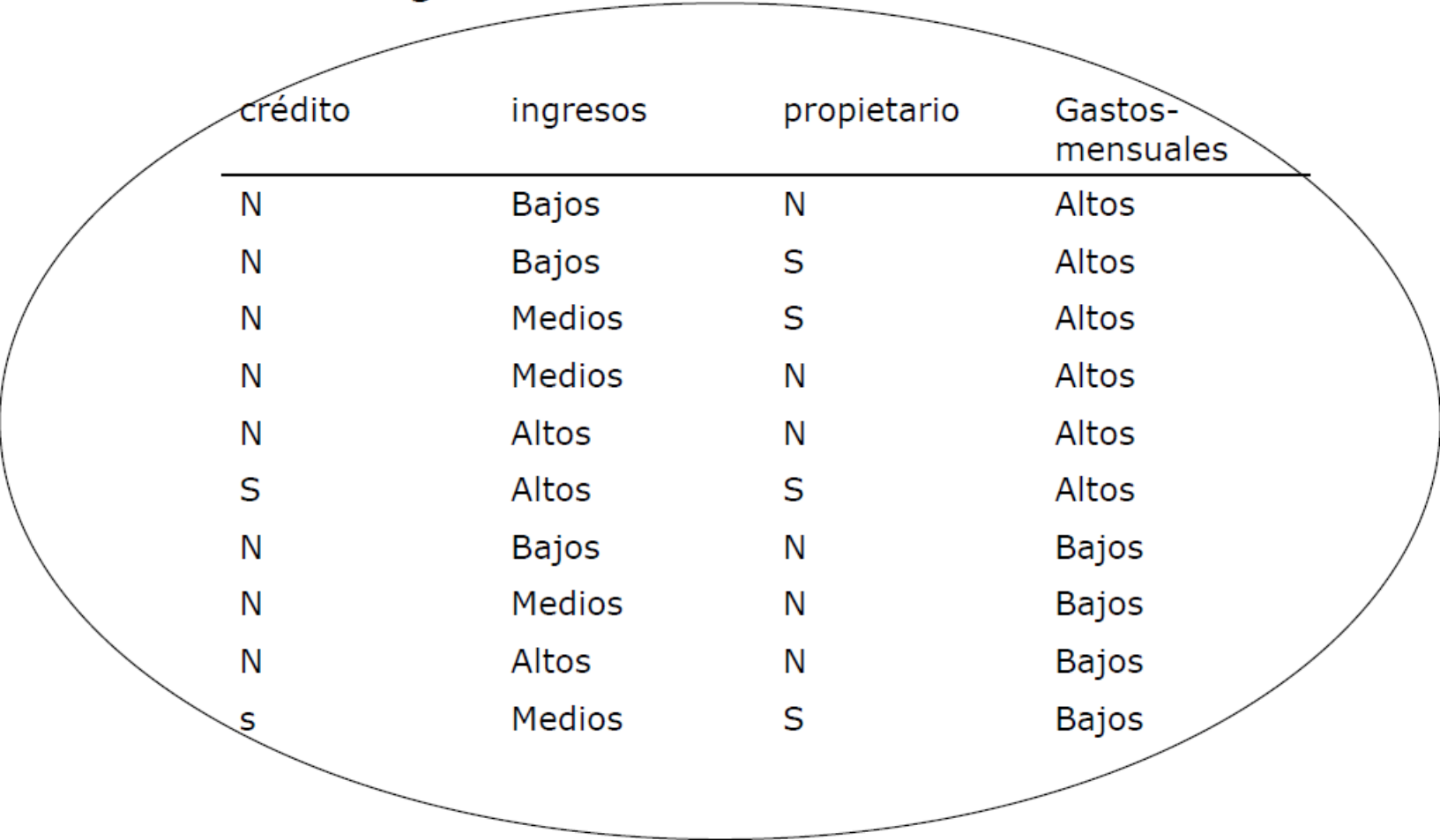
- Problema de 3 variables de entrada, 2 clases (Variable crédito: N/S) y 10 ejemplos

Problema de asignación de crédito

crédito	ingresos	propietario	Gastos-mensuales
N	Bajos	N	Altos
N	Bajos	S	Altos
N	Medios	S	Altos
N	Medios	N	Altos
N	Altos	N	Altos
S	Altos	S	Altos
N	Bajos	N	Bajos
N	Medios	N	Bajos
N	Altos	N	Bajos
s	Medios	S	Bajos

Ejemplo aprendizaje árbol decisión

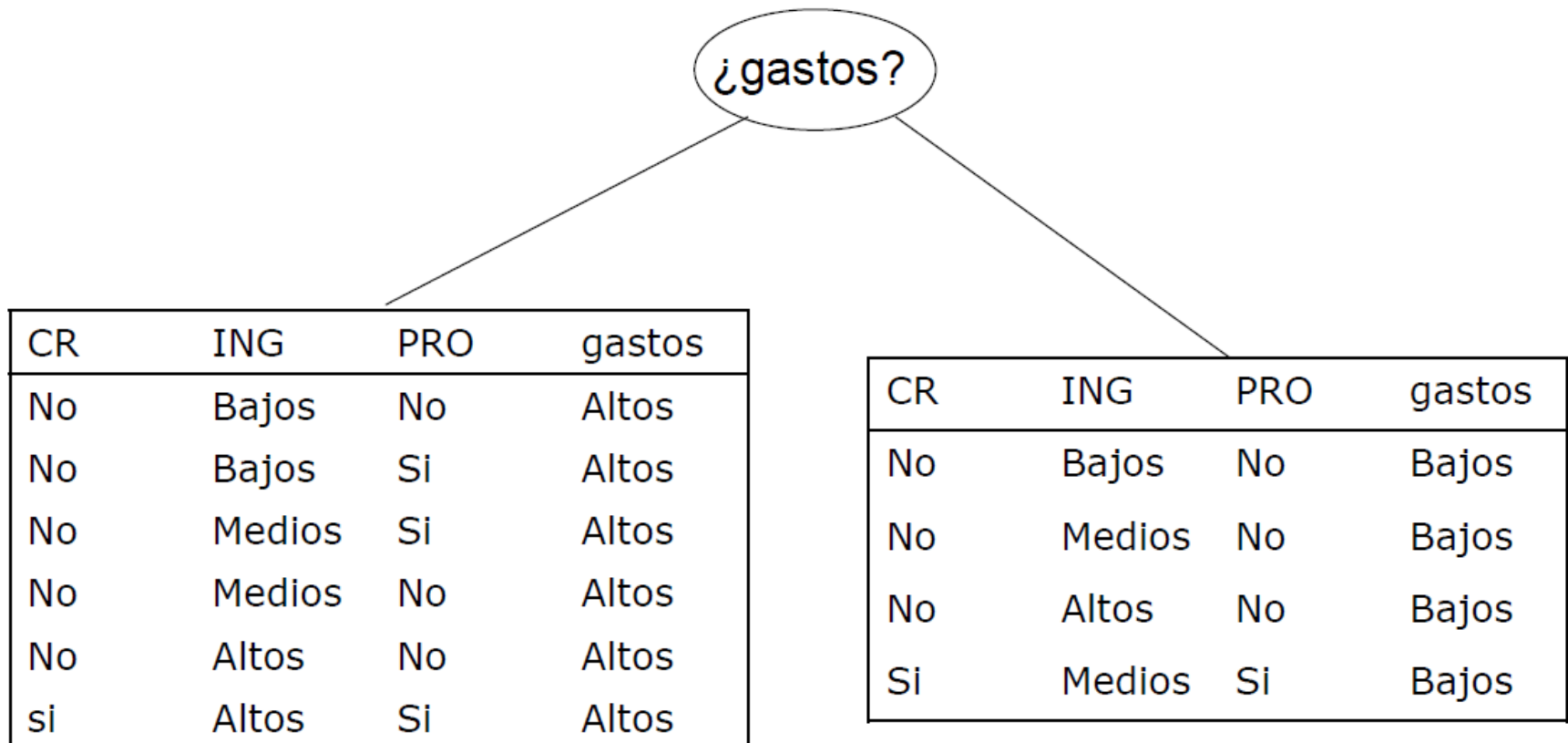
1. Se llama al algoritmo sobre el nodo raíz



crédito	ingresos	propietario	Gastos-mensuales
N	Bajos	N	Altos
N	Bajos	S	Altos
N	Medios	S	Altos
N	Medios	N	Altos
N	Altos	N	Altos
S	Altos	S	Altos
N	Bajos	N	Bajos
N	Medios	N	Bajos
N	Altos	N	Bajos
S	Medios	S	Bajos

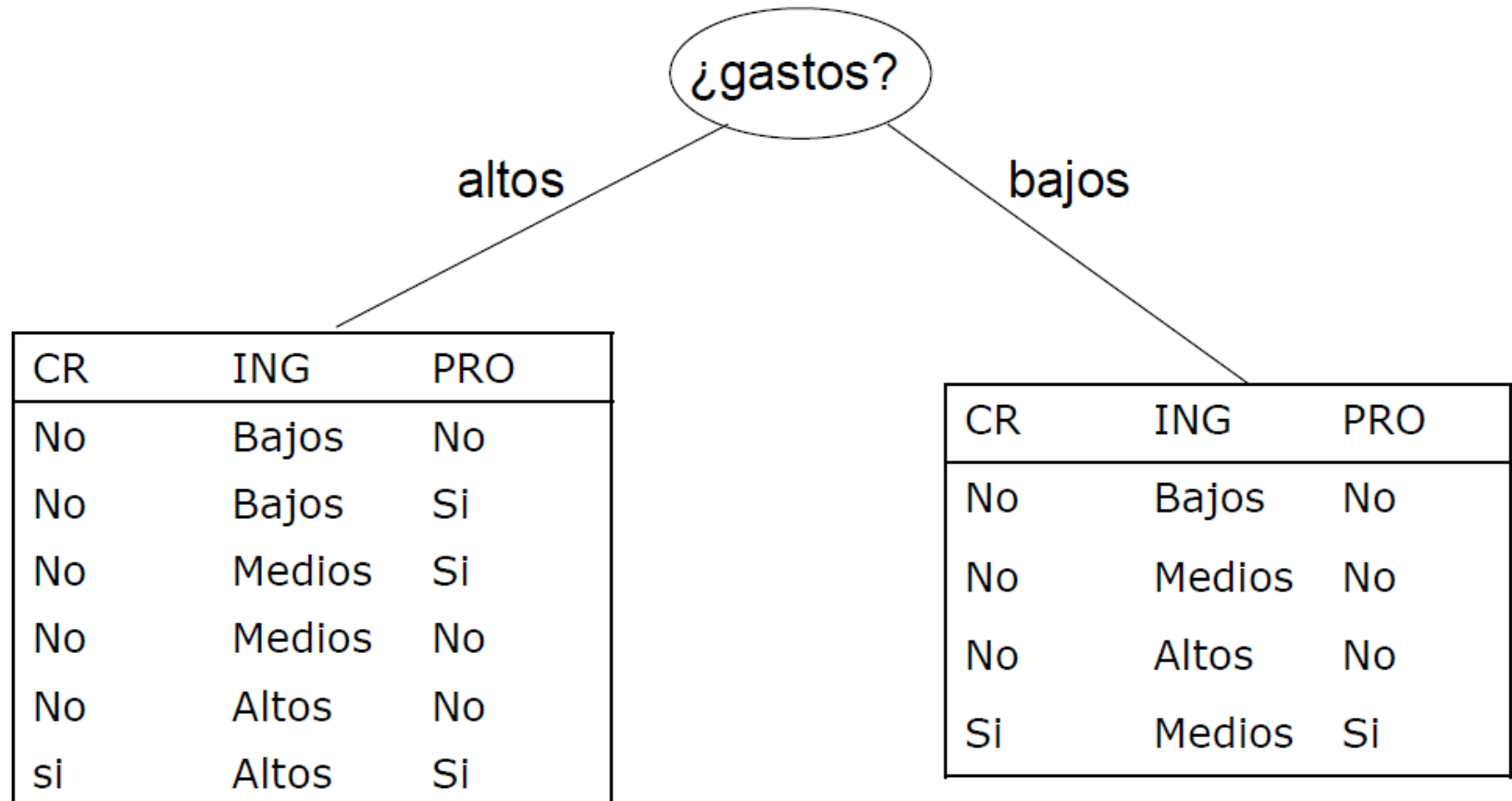
Ejemplo aprendizaje árbol decisión

2. Seleccionamos **gastos** como test



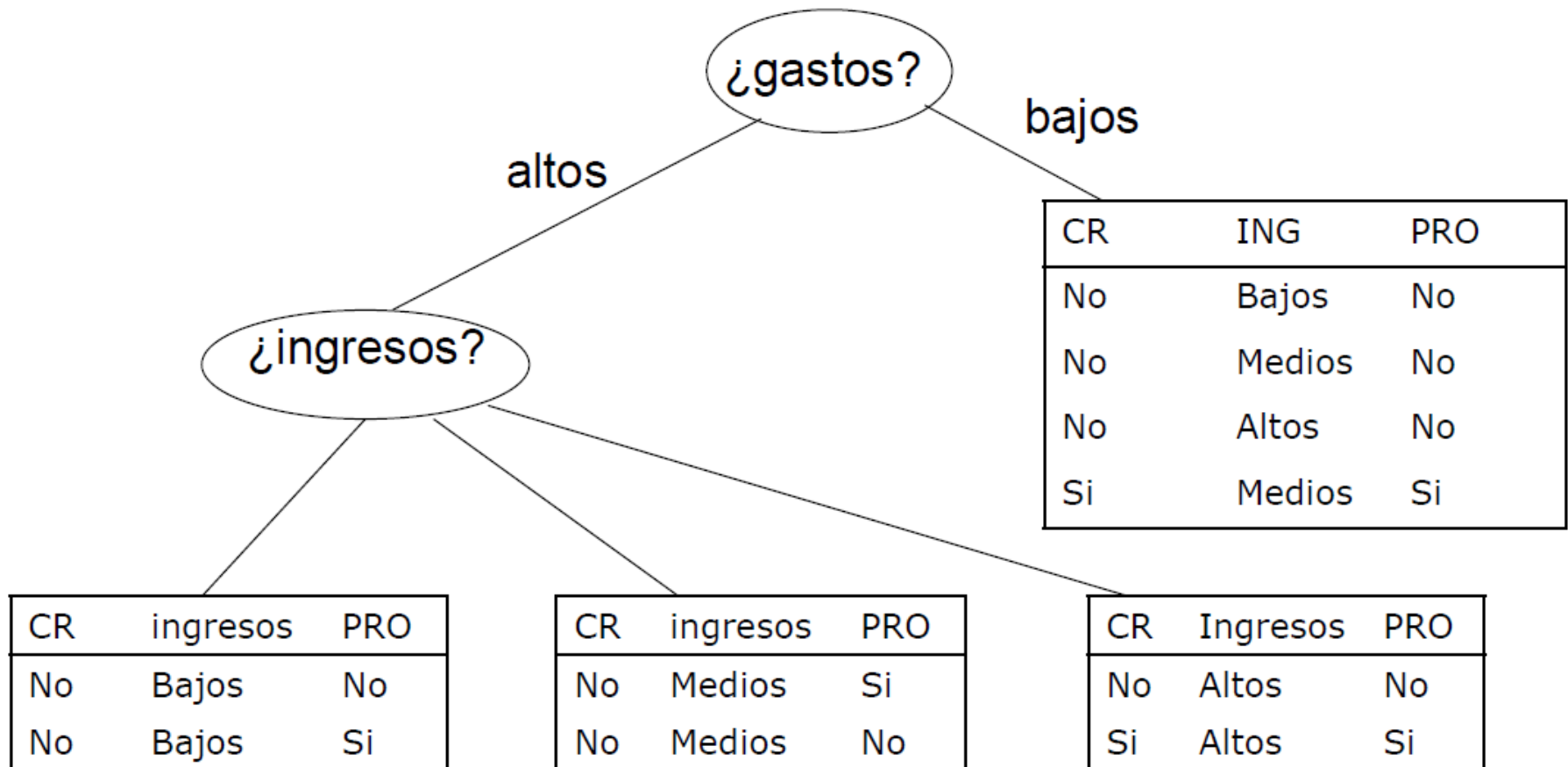
Ejemplo aprendizaje árbol decisión

3. Preparamos los nodos para las llamadas recursivas



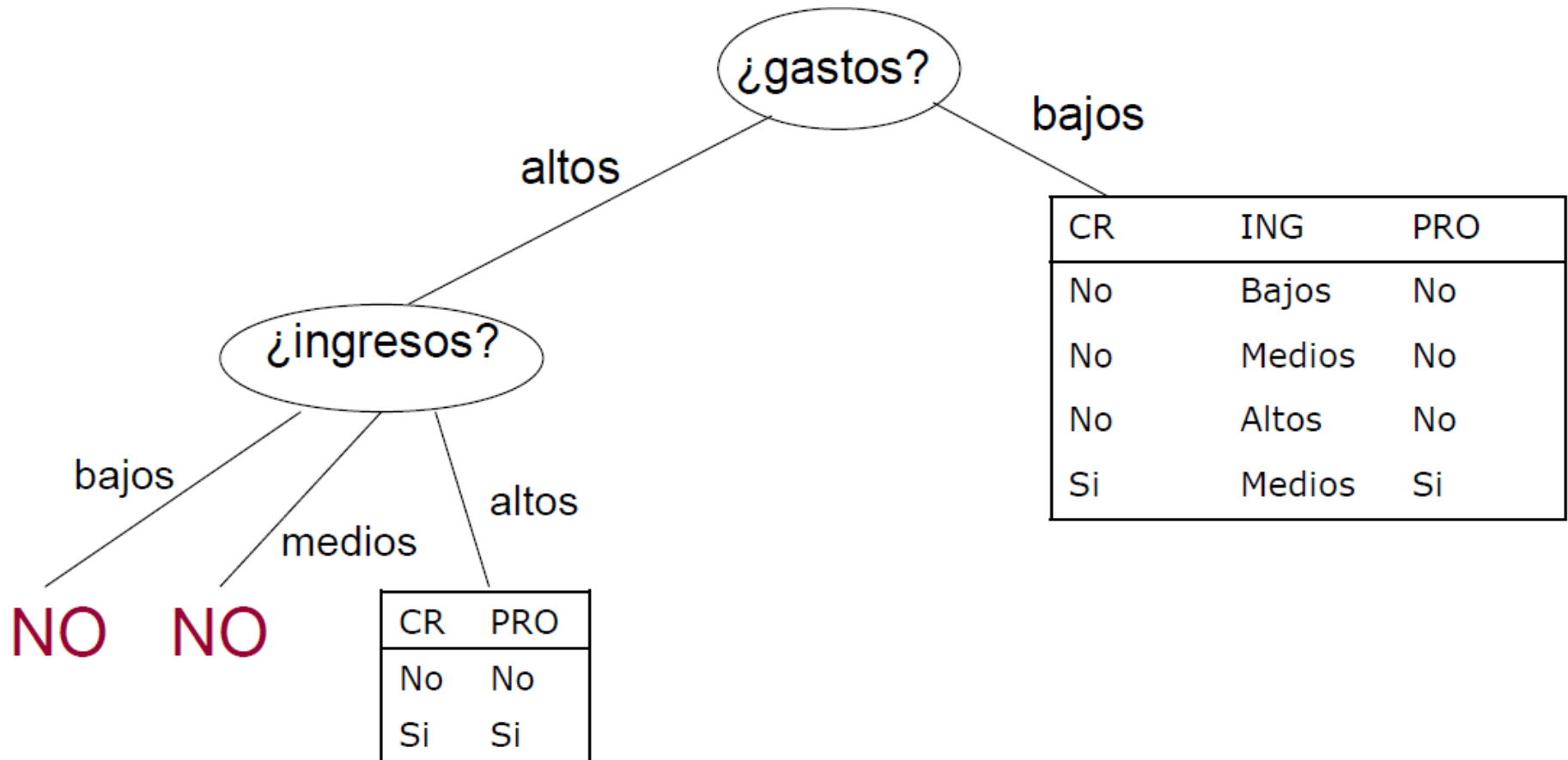
Ejemplo aprendizaje árbol decisión

4. Seleccionamos **ingresos** como test en gastos = altos



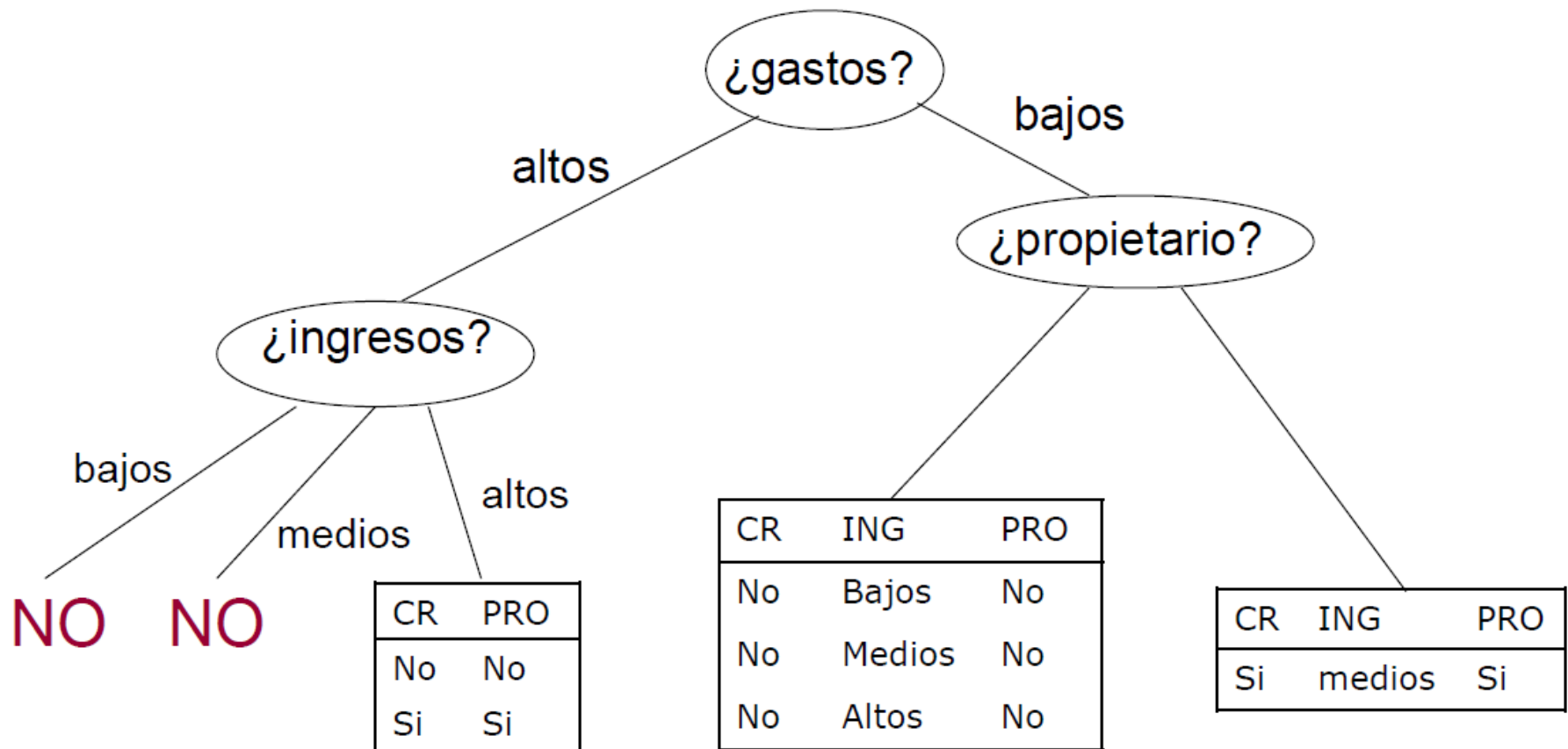
Ejemplo aprendizaje árbol decisión

5. Creamos nodos hoja



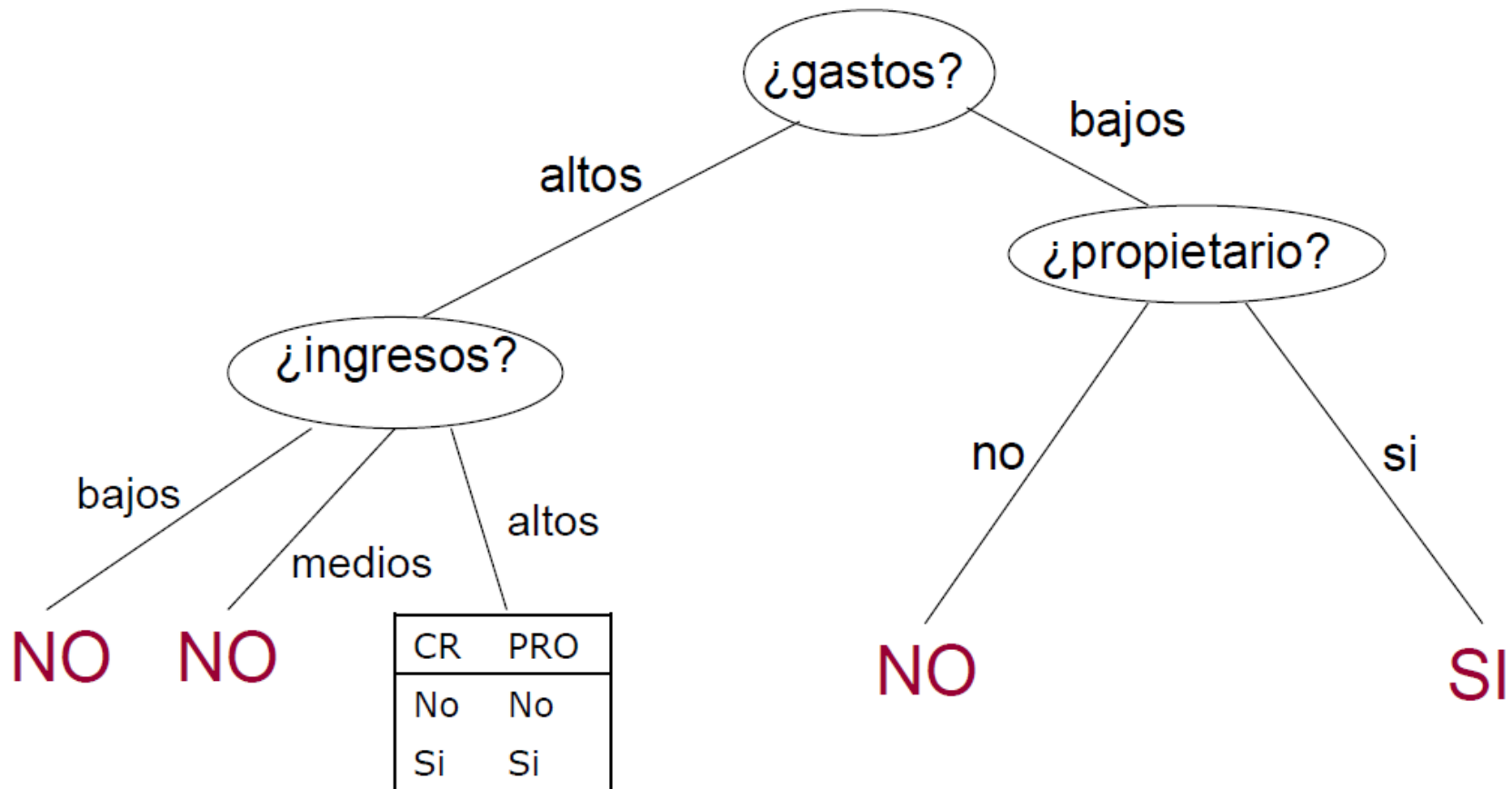
Ejemplo aprendizaje árbol decisión

6. Seleccionamos **propietario** como test en gastos = bajos



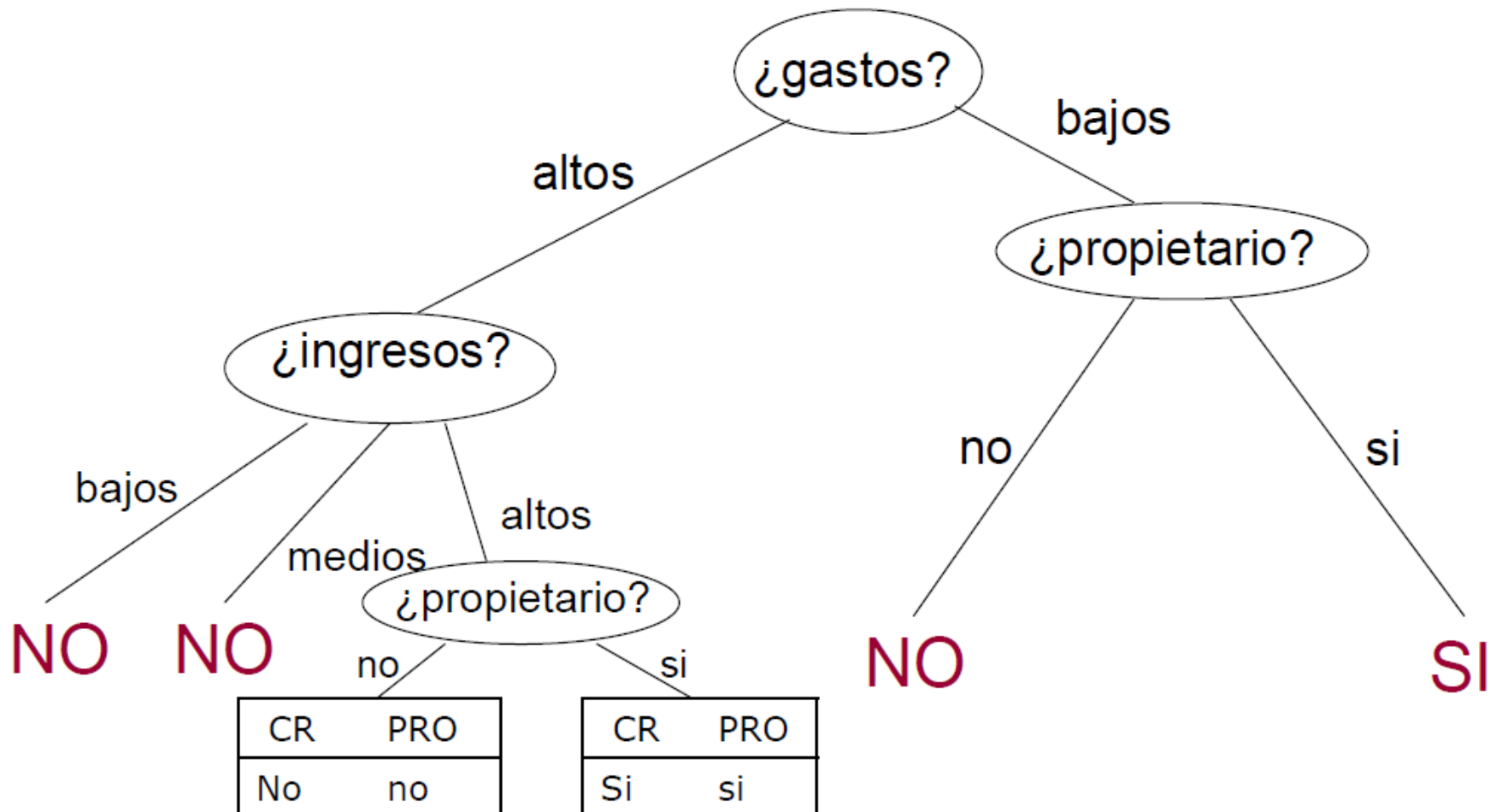
Ejemplo aprendizaje árbol decisión

7. Creamos nodos hoja



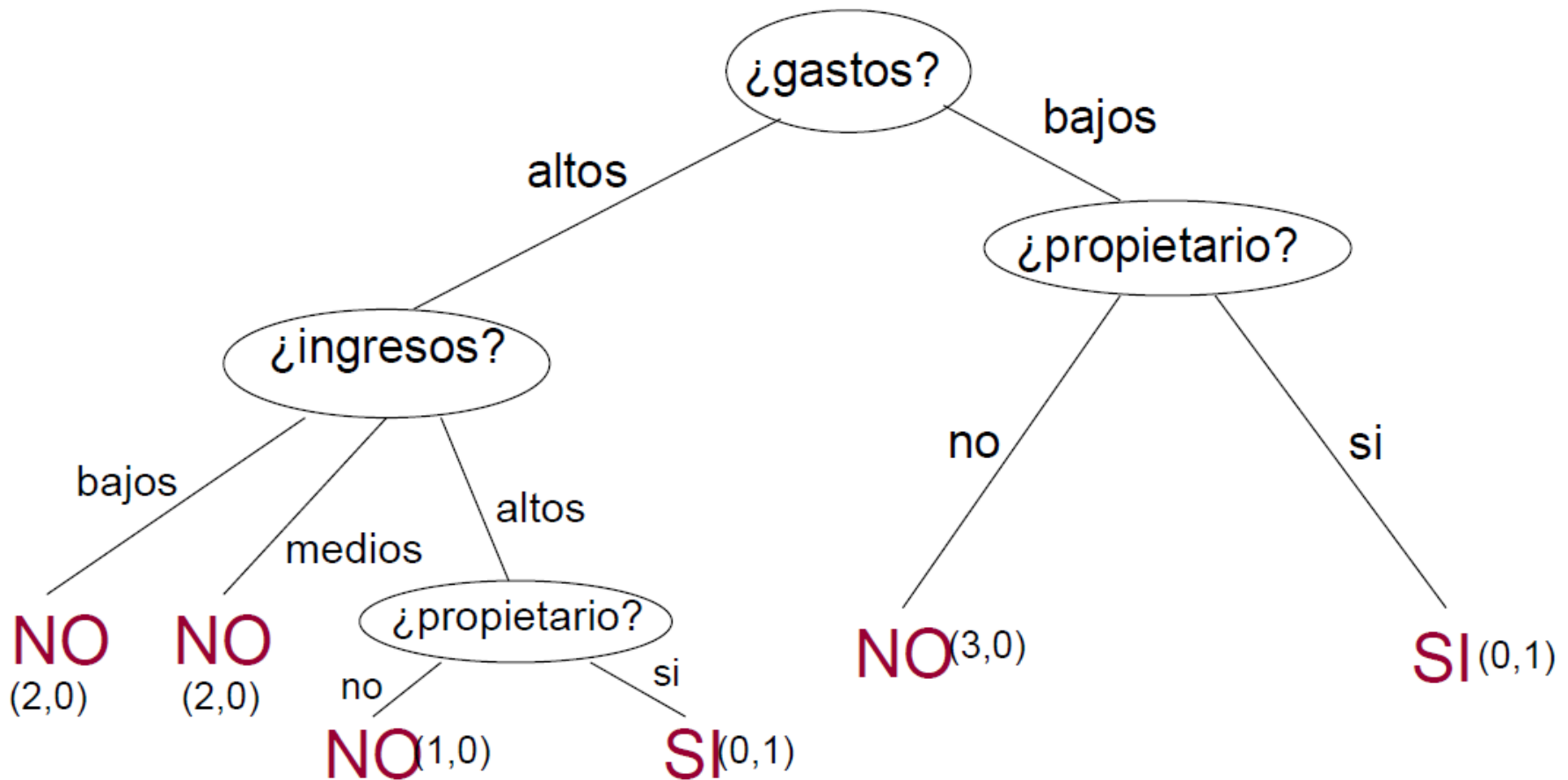
Ejemplo aprendizaje árbol decisión

8. Seleccionamos **propietario** como test en gastos=altos, ingresos=altos



Ejemplo aprendizaje árbol decisión

9. Árbol de decisión (#no, #si)

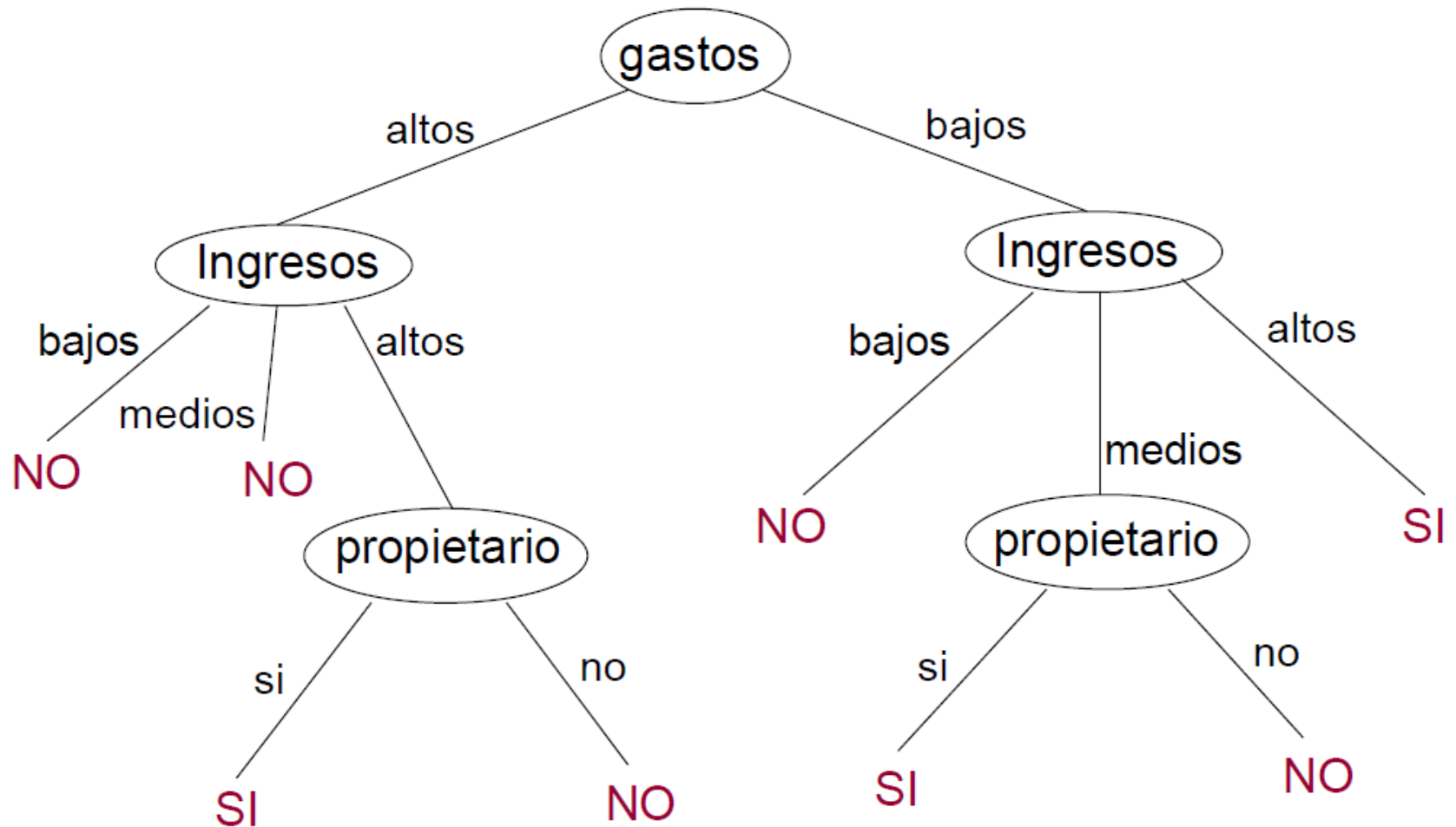


Aprendizaje del árbol de decisión

- Sea T el conjunto de ejemplos, C el conjunto de clases y A el conjunto de atributos. El algoritmo básico para generar un árbol es el siguiente:
- ConstruirArbol (T, C, A)
 1. Crear un nodo RAIZ para el árbol
 2. Si todos los ejemplos en T pertenecen a la misma clase C_i
Devolver el nodo RAIZ con etiqueta C_i
 3. Si $A = \emptyset$
Devolver el nodo RAIZ con etiqueta C_i donde C_i es la clase mayoritaria en T
 4. $a \leftarrow$ atributo de A que mejor clasifica T
 5. Etiquetar RAIZ con a
 6. Para cada valor v_i de a hacer
 1. Añadir una nueva rama debajo de RAIZ con el test $a = v_i$
 2. Sea T_i el subconjunto de T en el que $a = v_i$
 3. Si $T_i \neq \emptyset \rightarrow$ ConstruirArbol ($T_i, C, A - a$)
 7. Devolver RAIZ

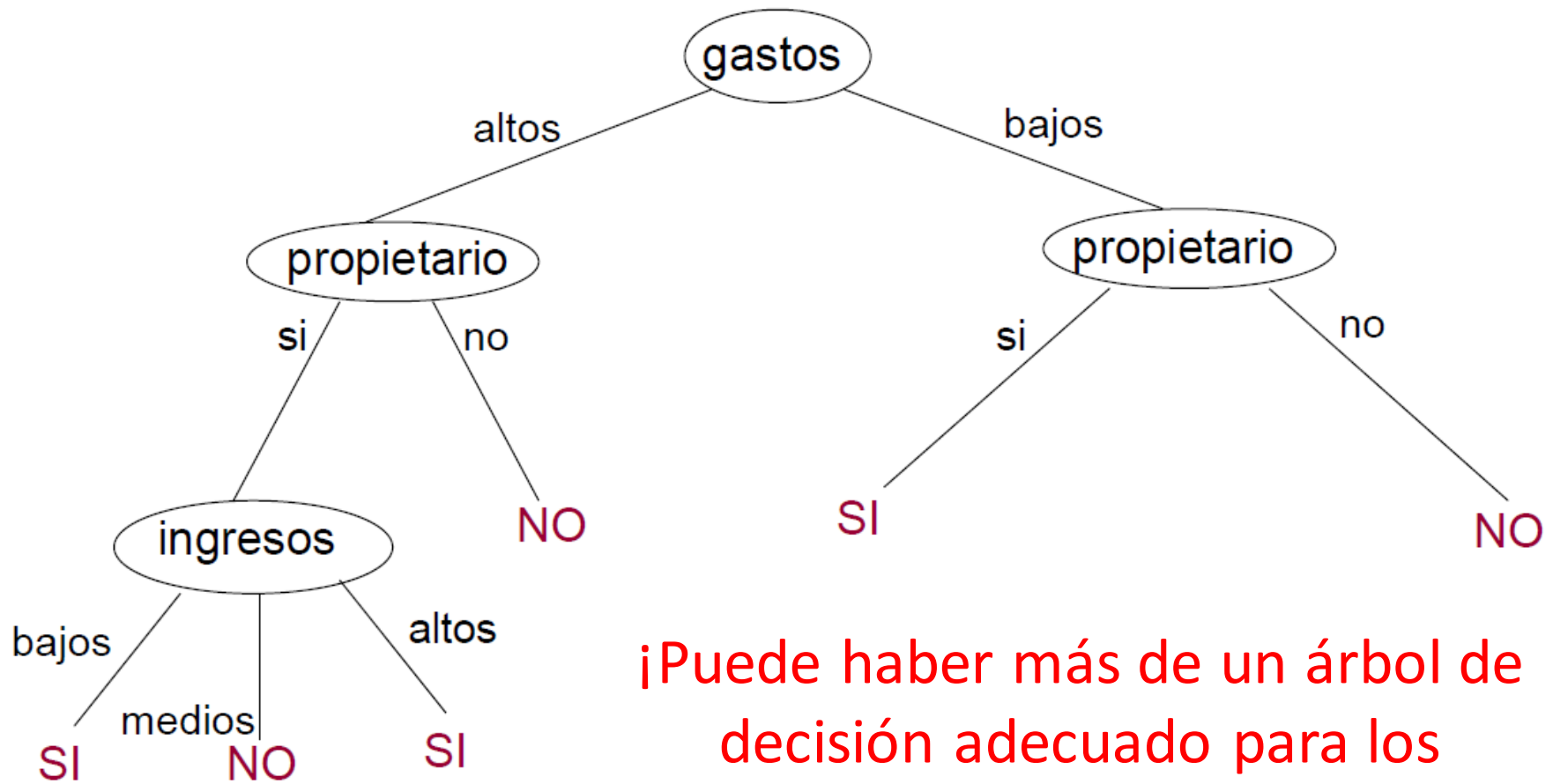
Ejemplo aprendizaje árbol decisión

Otro árbol de decisión para crédito (árbol 2)



Ejemplo aprendizaje árbol decisión

Otro árbol de decisión para crédito (árbol 3)



¡Puede haber más de un árbol de decisión adecuado para los mismos datos!

Ejemplo aprendizaje árbol decisión

1er. árbol: 6 reglas (2.33 premisas por regla)

2do. árbol: 8 reglas (2.5 premisas por regla)

3er. árbol: 6 reglas (2.5 premisas por regla)

- Dependiendo del orden en el que se van tomando los atributos obtenemos clasificadores de distinta complejidad (interpretabilidad)
- Lo ideal sería tomar en todo momento el atributo que mejor clasifica

¿Cómo decidir qué atributo es el mejor?

¿Cómo decidir qué atributo es el mejor?

- Preferencia por los nodos que tengan una distribución de las clases **homogénea**
 - ▣ Necesita una **medida de la impureza del nodo**

C0: 5

C1: 5

No homogéneo,
Alto grado de impureza

C0: 10

C1: 0

Homogéneo,
Bajo grado de impureza

Medidas de la impureza del nodo

- Índice Gini

- ▣ Árbol de decision: CART

$$GINI(t) = 1 - \sum_{j=1}^c p(j|t)^2$$

- Ratio de ganancia de información (entropía)

- ▣ Árboles de decision: ID3, C4.5 (J48 en Weka)

$$E(t) = \sum_{j=1}^c -p(j|t) \log_2(p(j|t))$$

Tratan de seleccionar los atributos que más reducen la
duda (heterogeneidad)

Ejemplos del cálculo de la entropía

$$E(t) = \sum_{j=1}^c -p(j|t) \log_2(p(j|t))$$

C1	0
C2	6

$$P(C1 | t) = 0/6 = 0 \quad P(C2 | t) = 6/6 = 1$$

$$\text{Entropía} = -0 \cdot \log_2(0) + -1 \cdot \log_2(1) = 0 + 0 = 0$$

C1	1
C2	5

$$P(C1 | t) = 1/6 \quad P(C2 | t) = 5/6$$

$$\text{Entropía} = -1/6 \cdot \log_2(1/6) + -5/6 \cdot \log_2(5/6) = 0.43 + 0.22 = 0.65$$

C1	2
C2	4

$$P(C1 | t) = 2/6 \quad P(C2 | t) = 4/6$$

$$\text{Entropía} = -2/6 \cdot \log_2(2/6) + -4/6 \cdot \log_2(4/6) = 0.53 + 0.39 = 0.92$$

C1	3
C2	3

$$P(C1 | t) = 3/6 \quad P(C2 | t) = 3/6$$

$$\text{Entropía} = -3/6 \cdot \log_2(3/6) + -3/6 \cdot \log_2(3/6) = 0.5 + 0.5 = 1$$

División basada en Information Gain

- La ganancia de información es la cantidad de información que se gana con respecto al atributo a predecir (clases, C) si dividimos por el atributo A

$$Gain(A) = E(C) - E(A)$$

- Dado un atributo A con k valores $\{A_1, A_2, \dots, A_k\}$
- Si el atributo A es seleccionado como nodo
 - Los ejemplos que hayan llegado al nodo, T , se particionarán de acuerdo a los k valores del atributo A
 - $\{A_1, A_2, \dots, A_k\} \rightarrow \{T_1, T_2, \dots, T_k\}$
 - En el nodo i -ésimo asignaremos los ejemplos del nodo T que tengan el valor i -ésimo en el atributo A
 - La entropía media del atributo A en sus sub-espacios (nodos hijos generados) se calcula aplicando:

$$E(A) = \sum_{i=1}^k \frac{n_i}{n} * E(T_i)$$

- n_i es el número de ejemplos del nodo hijo T_i
- n es el número de ejemplos del nodo T (el dividido)

Deberemos elegir el atributo que maximice la ganancia de información

Ejemplos del cálculo del índice GINI

$$GINI(t) = 1 - \sum_{j=1}^c p(j|t)^2$$

C1	0
C2	6

$$P(C1 | t) = 0/6 = 0 \quad P(C2 | t) = 6/6 = 1$$

$$Gini = 1 - P(C1 | t)^2 - P(C2 | t)^2 = 1 - 0 - 1 = 0$$

C1	1
C2	5

$$P(C1 | t) = 1/6 \quad P(C2 | t) = 5/6$$

$$Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C1	2
C2	4

$$P(C1 | t) = 2/6 \quad P(C2 | t) = 4/6$$

$$Gini = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

C1	3
C2	3

$$P(C1 | t) = 3/6 \quad P(C2 | t) = 3/6$$

$$Gini = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

Comentarios generales

□ C4.5 y CART

- Modelo interpretable
- Frontera de decisión no lineal
- Determinan las variables más importantes para resolver el problema
- Permiten abordar tanto problemas de clasificación binarios como multi-clase
- Admiten variables de entrada tanto numéricas como discretas
 - En las numéricas determinan el mejor valor del umbral de división en cada nodo
- Permiten manejar ejemplos con valores perdidos
- Control de sobre-aprendizaje
 - Pre poda: parar de ramificar si hay pocos ejemplos (nodo o descendientes)
 - Técnicas de poda (bottom-up calculando el error)
- Considerados como dos de las 10 mejores técnicas de minería de datos

□ CART

- Permite abordar problemas de regresión
 - Reducen la varianza (error) al ramificar
- Realiza ramificación binaria para atributos numéricos y discretos

□ C4.5

- Realiza ramificación binaria para atributos numéricos
- Para atributos discretos crea tantas ramas como valores