uc3m | Universidad Carlos III de Madrid

OPENCOURSEWARE
APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS DE DATOS
GRADO EN ESTADÍSTICA Y EMPRESA
Ricardo Aler

MODELOS: ÁRBOLES DE DECISIÓN Y REGLAS

Datos de entrada

¿Es un buen día para jugar al tenis?

Cielo	Temperatura	Humedad	Viento	Tenis
sol	85	85	no	no
sol	80	90	si	no
nubes	83	86	no	si
lluvia	70	96	no	si
lluvia	68	80	no	si
lluvia	65	70	si	no
nubes	64	65	si	si
sol	72	95	no	no
sol	69	70	no	si
lluvia	75	80	no	si
sol	75	70	si	si
nubes	72	90	si	si
nubes	81	75	no	si
lluvia	71	91	si	no

MODELOS: ÁRBOLES DE DECISIÓN

Esquema general

Atributos, características, variables de entrada, variables independientes, predictores

Etiqueta, clase, variable de salida, variable dependiente, respuesta

Instancias, ejemplos, datos, patrones

Cielo	Temp	Hum	Viento	Tenis
sol	85	85	no	no
sol	80	90	si	no
nubes	83	86	no	si
lluvia	70	96	no	si
lluvia	68	80	no	si
lluvia	65	70	si	no
nubes	64	65	si	si
sol	72	95	no	no
sol	69	70	no	si
lluvia	75	80	no	si
sol	75	70	si	si
nubes	72	90	si	si
nubes	81	75	no	si
lluvia	71	91	si	no

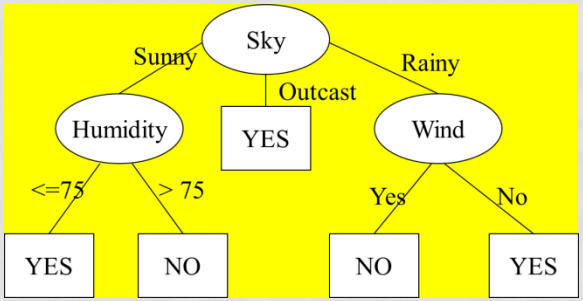
Datos Entrenamiento

Datos test

Cielo	Temp.	Humedad	Viento	Tenis
Sol	60	65	No	?????

Algoritmo

DECISION TREE



Modelo (Clasificador)

Clase = Si

Predicción

MODELOS: OTROS

Atributos, características, variables de entrada, variables independientes, predictores

Etiqueta, clase, variable de salida, variable dependiente

Instancias, ejemplos, datos, patrones

Cielo	Temp	Hum	Viento	Tenis
sol	85	85	no	no
sol	80	90	si	no
nubes	83	86	no	si
lluvia	70	96	no	si
lluvia	68	80	no	si
lluvia	65	70	si	no
nubes	64	65	si	si
sol	72	95	no	no
sol	69	70	no	si
lluvia	75	80	no	si
sol	75	70	si	si
nubes	72	90	si	si
nubes	81	75	no	si
lluvia	71	91	si	no

Datos Entrenamiento

Datos test

Cielo	Temp.	Humedad	Viento	Tenis
Sol	60	65	No	?????

Algoritmo

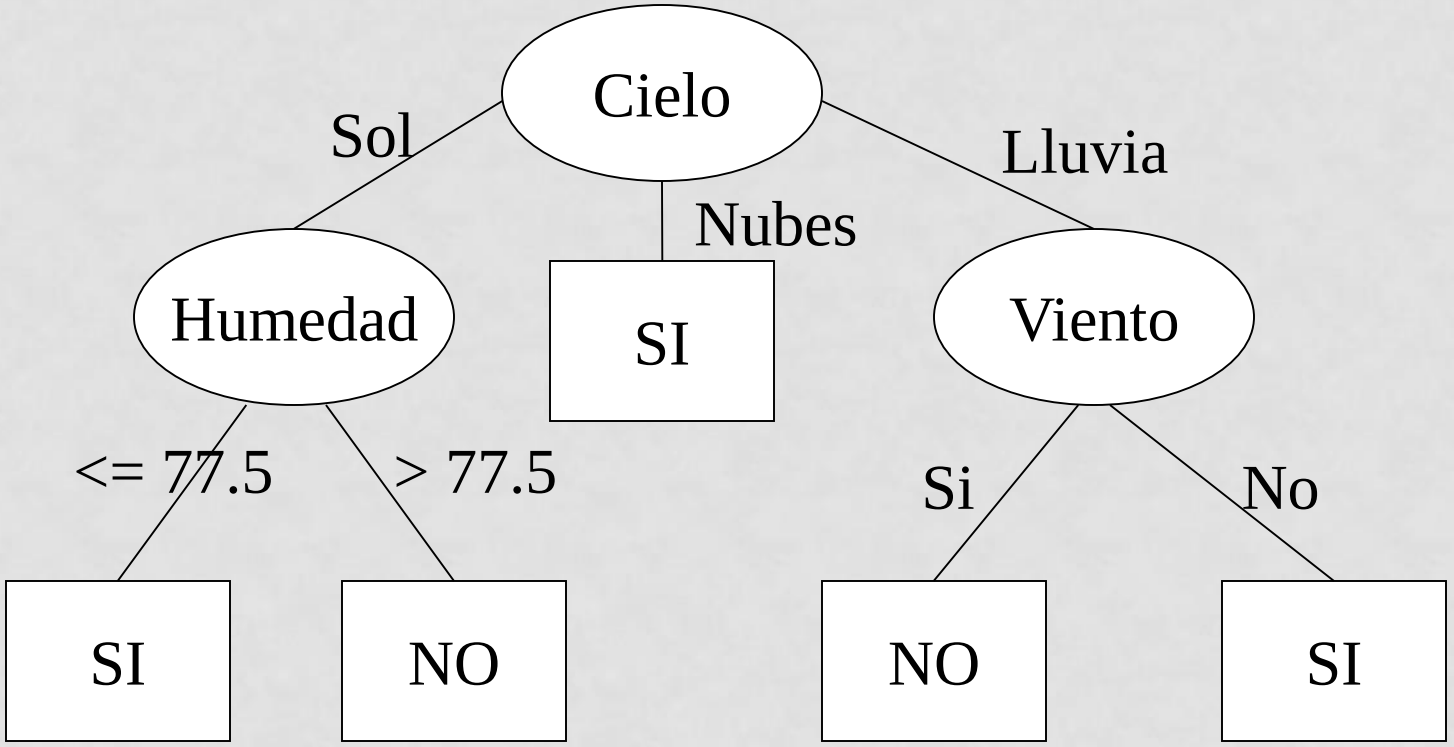
- Nearest neighbor (KNN)
- Ensembles (bagging, boosting, stacking, …)
- Funciones: redes de neuronas, deep learning, SVMs, …
- Naive bayes, redes bayesianas

Clase = Si

Modelo (Clasificador)

Predicción

Árboles de decisión



Algoritmos de construcción de árboles de decisión

- El más básico es el ID3: construye árboles de decisión de manera recursiva, de la raíz hacia las hojas, seleccionando en cada momento el mejor nodo para poner en el árbol
- El C4.5 (o J48), trata con valores continuos y utiliza criterios estadísticos para impedir que el árbol se sobreadapte (que "crezca demasiado", que se aprenda los datos en lugar de generalizar)

Algoritmo ID3 simplificado

1. Detener la construcción del árbol si:
   1. Todos los ejemplos pertenecen a la misma clase
   2. Si no quedan ejemplos o atributos
2. Si no, elegir el mejor atributo para poner en ese nodo (el que minimice la entropía media)
3. Crear de manera recursiva tantos subárboles como posibles valores tenga el atributo seleccionado

Algoritmo ID3 detallado

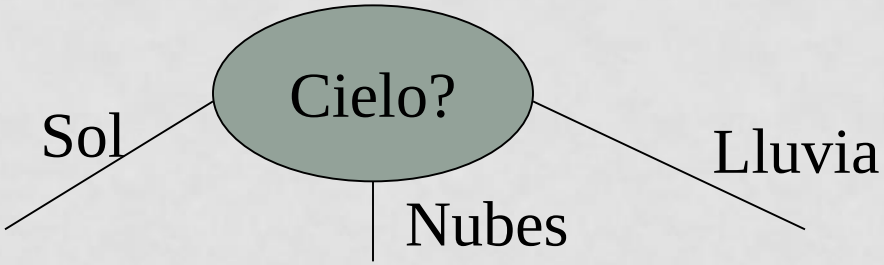
- **ID3(Ejemplos, Atributo-objetivo, Atributos)**

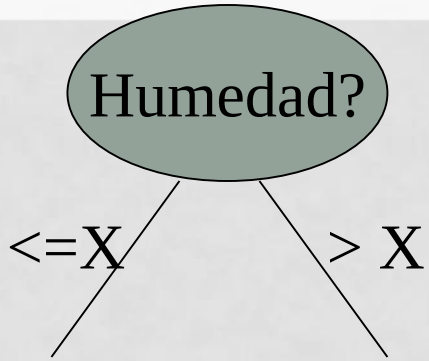
1. Si todos los Ejemplos son positivos, devolver un nodo etiquetado con +
2. Si todos los Ejemplos son negativos, devolver un nodo etiquetado con -
3. Si Atributos está vacío, devolver un nodo etiquetado con el valor más frecuente de Atributo-objetivo en Ejemplos.
4. En otro caso:
   4.1. Sea A el atributo de Atributos que MEJOR clasifica Ejemplos
   4.2. Crear Árbol, con un nodo etiquetado con A.
   4.3. Para cada posible valor v de A, hacer:
   * Añadir un arco a Árbol, etiquetado con v.
   * Sea Ejemplos(v) el subconjunto de Ejemplos con valor del atributo A igual a v.
   * Si Ejemplos(v) es vacío:
   - Entonces colocar debajo del arco anterior un nodo etiquetado con el valor más frecuente de Atributo-objetivo en Ejemplos.
   - Si no, colocar debajo del arco anterior el subárbol ID3(Ejemplos(v), Atributo-objetivo, Atributos-{A}).
   4.4 Devolver Árbol

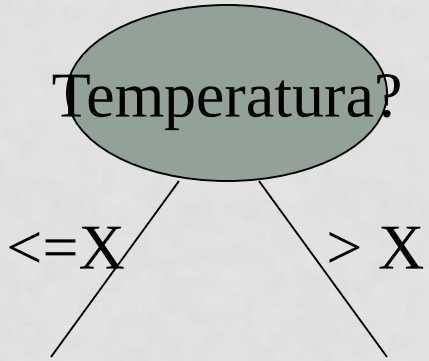
Algoritmo C4.5 simplificado

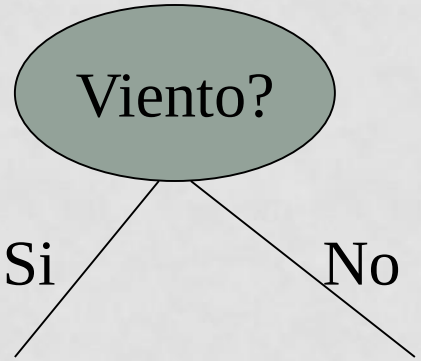
1. Detener la construcción del árbol si:
   1. Todos los ejemplos pertenecen a la misma clase
   2. Si no quedan ejemplos o atributos
   3. Si no se espera que se produzcan mejoras continuando la subdivisión
2. Si no, elegir el mejor atributo para poner en ese nodo (el que minimice la entropía media)
3. Crear de manera recursiva tantos subárboles como posibles valores tenga el atributo seleccionado

¿Qué nodo es el mejor para poner en la raíz del árbol?



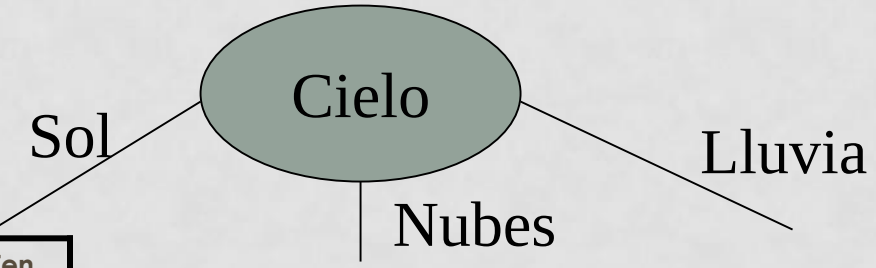






Supongamos que usamos Cielo

Cielo nos genera tres particiones de los datos, tantas como valores posibles tiene



C	T	H	V	Ten
Sol	85	85	No	No
Sol	80	90	Si	No
Sol	72	95	No	No
Sol	69	70	No	Si
Sol	75	70	Si	Si

C	T	H	V	Ten
Nubes	83	86	No	Si
Nubes	64	65	Si	Si
Nubes	72	90	Si	Si
Nubes	81	75	No	Si

C	T	H	V	Ten
Lluvia	70	96	No	Si
Lluvia	68	80	No	Si
Lluvia	75	80	No	Si
Lluvia	65	70	Si	No
Lluvia	71	91	Si	No

“3 No, 2 Si”

Tendencia al “no”

“0 No, 4 Si”

Partición perfecta

“2 No, 3 Si”

Tendencia al “si”

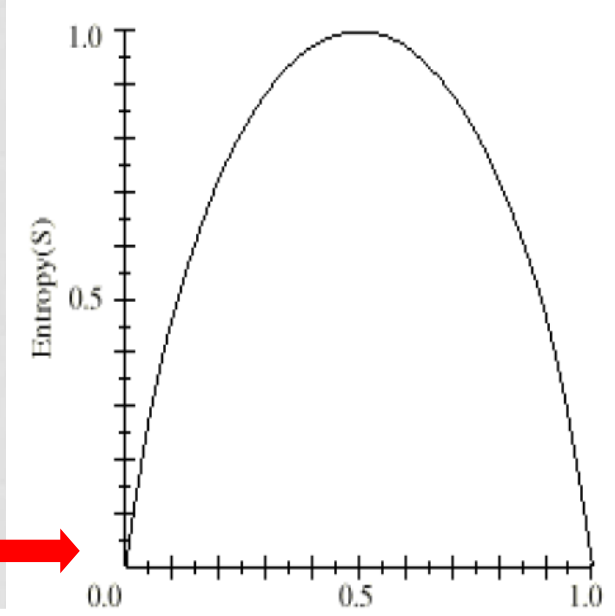
¿Cómo medimos lo bueno que es Cielo como atributo para clasificar?

- Usaremos una medida que obtenga el mejor valor cuando el atributo me obtenga particiones lo mas homogéneas posible, en media
  - Homogénea: "0 No, todo Si"; o bien "todo No, 0 Si"
  - Indecisión: "50% No, 50% Si"
- Una medida que me dice lo lejana que está una partición de la perfección es la entropía
- A mayor entropía, peor es la partición

$$H(P) = -\sum_{Ci} p_{Ci} \log_2(p_{Ci})$$

$$H(P) = -(p_{si} \log_2(p_{si}) + p_{no} \log_2(p_{no}))$$

$$p_{no} = (1 - p_{si})$$

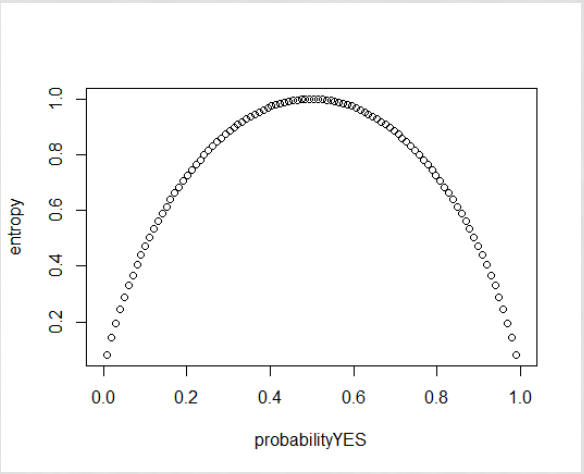


Proporción de "Si"es

OTRAS MEDIDAS DE LA CALIDAD DE LAS PARTICIONES

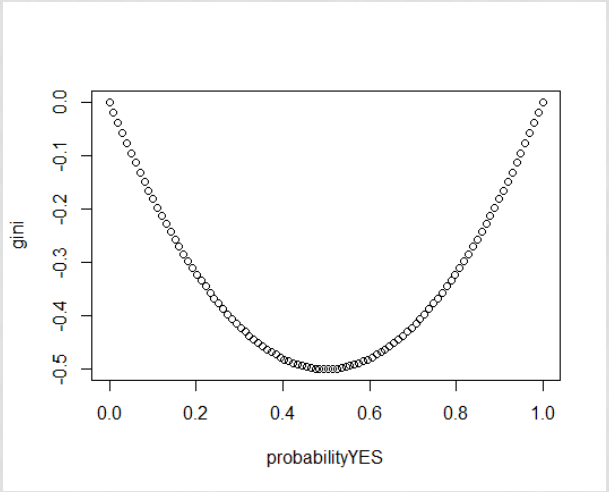
Entropía

$$H(P) = -\sum_{Ci} p_{Ci} \log_2(p_{Ci})$$



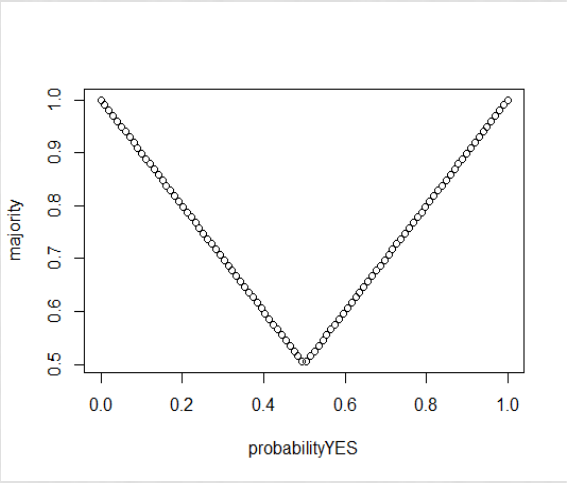
Gini

$$Gini(P) = -\sum_{Ci} p_{Ci}(1 - p_{Ci})$$



Majority

$$M(P) = \max(p_{si}, p_{no})$$



Entropía media de Cielo

- Cielo genera tres particiones cuya entropía es:
  1. "3 No, 2 Si": $H=-((3/5)*\log_2(3/5)+(2/5)*\log_2(2/5) = 0.97$
  2. "0 No, 4 Si": $H=-((0/4)*\log_2(0/4)+1*\log_2(1))=0$
  3. "3 Si, 2 No": $H=-((2/5)*\log_2(2/5)+(3/5)*\log_2(3/5))= 0.97$

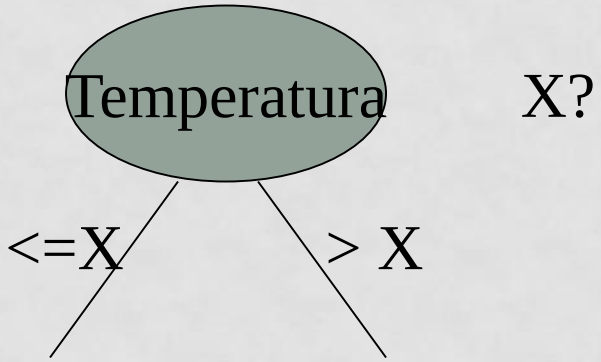
La entropía media ponderada de Cielo será:

- $HP=(5/14)*0.97+(4/14)*0+(5/14)*0.97 =$ **0.69**
- Nota: hay 14 datos en total

¿Y si el atributo es contínuo?

Hay que partir por el valor X, donde sea mas conveniente, minimizando la entropía

Nota: solo hemos probado algunas de los posibles puntos de corte (thresholds), siendo el mejor X=84 con entropia media ponderada = 0.83



64 – Si, 65-No, 68 – Si, 69 – Si, 70 – Si, 71 – No, 72 – SiNo, 75 – SiSi, 80 – No, 81 – Si, 83 – Si, 85 - No

HP = 0.83

4 No, 9 Si

X=84

1 No, 0 Si

64 – Si, 65-No, 68 – Si, 69 – Si, 70 – Si, 71 – No, 72 – SiNo, 75 – SiSi, 80 – No, 81 – Si, 83 – Si, 85 - No

HP = 0.93

2 No, 4 Si

X=71.5

3 No, 5 Si

64 – Si, 65-No, 68 – Si, 69 – Si, 70 – Si, 71 – No, 72 – SiNo, 75 – SiSi, 80 – No, 81 – Si, 83 – Si, 85 - No

HP = 0.89

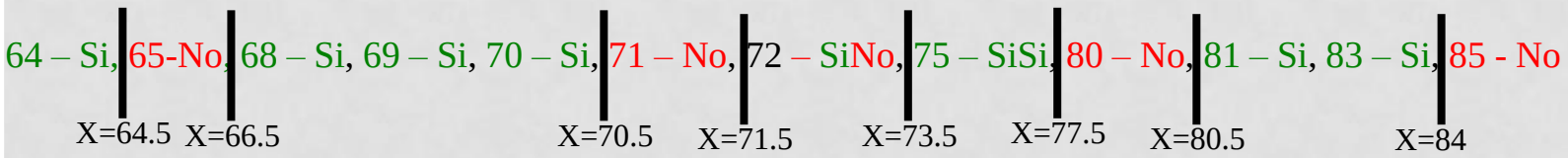
1 No, 4 Si

X=70.5

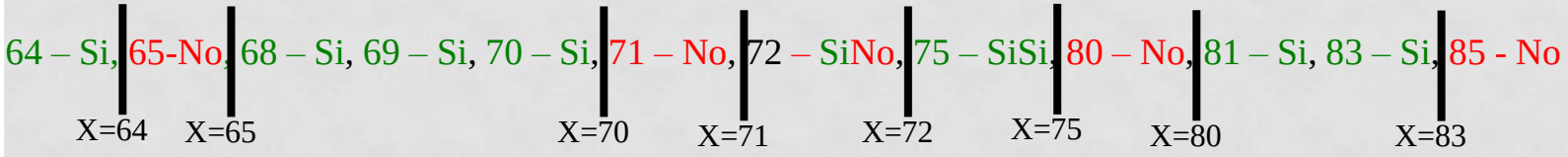
4 No, 5 Si

Posibles puntos de corte (thresholds)

Los posibles puntos de corte son las transiciones de Si a No, o de No a Si.

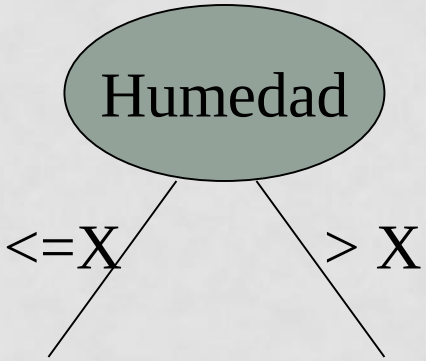


- Cómo se calcula el punto de corte depende de detalles específicos de la implementación del algoritmo que construye el árbol. Una buena manera de hacerlo es usando la media. Ej: 64.5 = (64+65)/2.
- Otras implementaciones usan el valor máximo de la partición de la izquierda. En ese caso, los puntos de corte hubieran sido los siguientes



- Nótese que se haga de una manera u otra, las entropías calculadas con los datos de entrenamiento son las mismas, puesto que las dos maneras de calcular los puntos de corte dan lugar a las mismas particiones (con los datos de entrenamiento)

Caso de humedad



X?

65-Si, 70-NoSiSi, 75–Si, 80-SiSi, | 85-No, 86-Si, 90-NoSi, 91-No, 95-No, 96-Si,

X=82.5

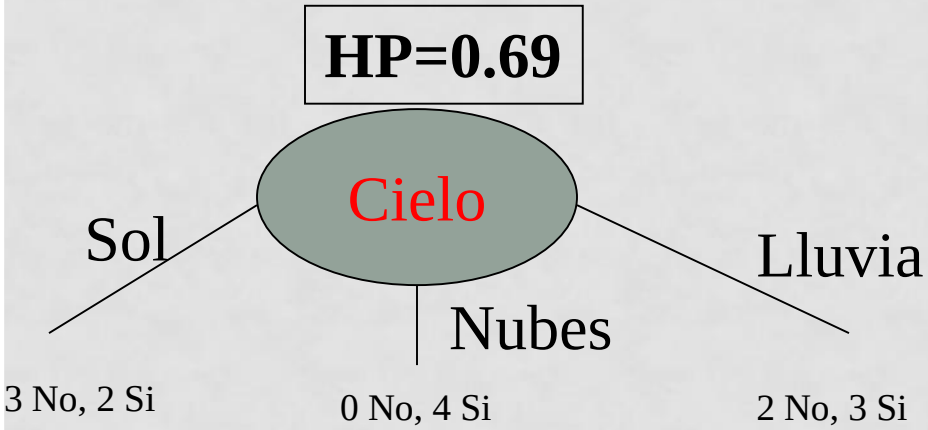
1 No, 6 Si

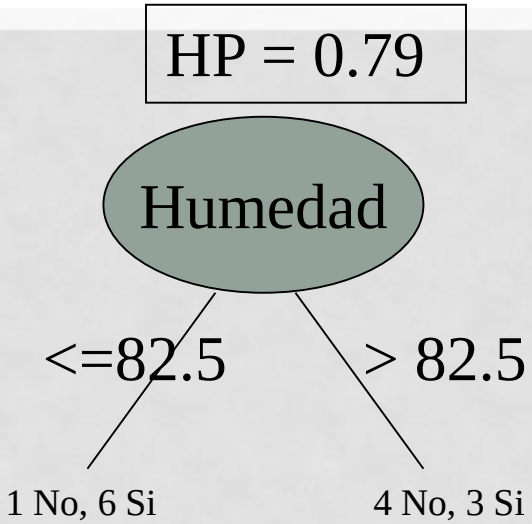
4 No, 3 Si

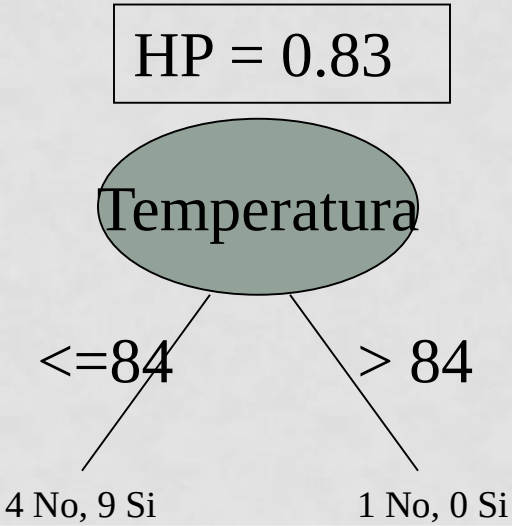
HP = 0.79

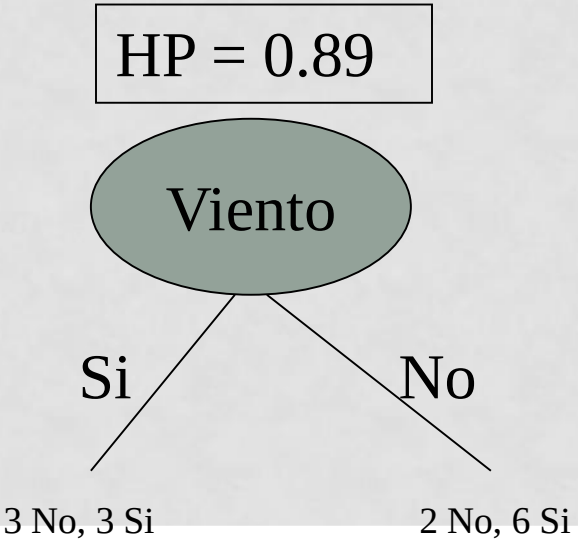
Nota: hay otras posibilidades para puntos de corte, pero esta es la mejor (la entropía ponderada media es mínima)

¿Qué nodo es el mejor para poner en la raíz?



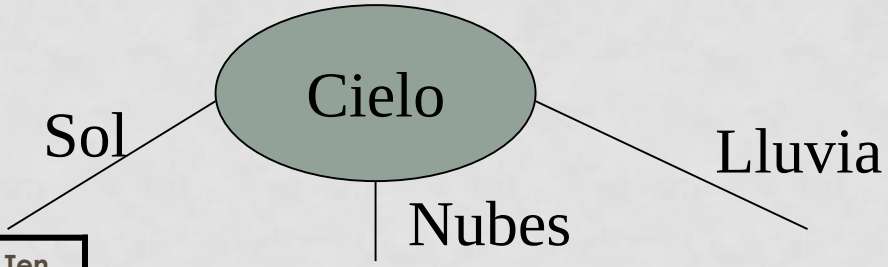






Construcción recursiva del árbol

Ahora que ya tenemos el nodo raíz, el proceso continua recursivamente: hay que construir tres subárboles con los datos que se muestran en cada rama



C	T	H	V	Ten
Sol	85	85	No	No
Sol	80	90	Si	No
Sol	72	95	No	No
Sol	69	70	No	Si
Sol	75	70	Si	Si

C	T	H	V	Ten
Nubes	83	86	No	Si
Nubes	64	65	Si	Si
Nubes	72	90	Si	Si
Nubes	81	75	No	Si

C	T	H	V	Ten
Lluvia	70	96	No	Si
Lluvia	68	80	No	Si
Lluvia	75	80	No	Si
Lluvia	65	70	Si	No
Lluvia	71	91	Si	No

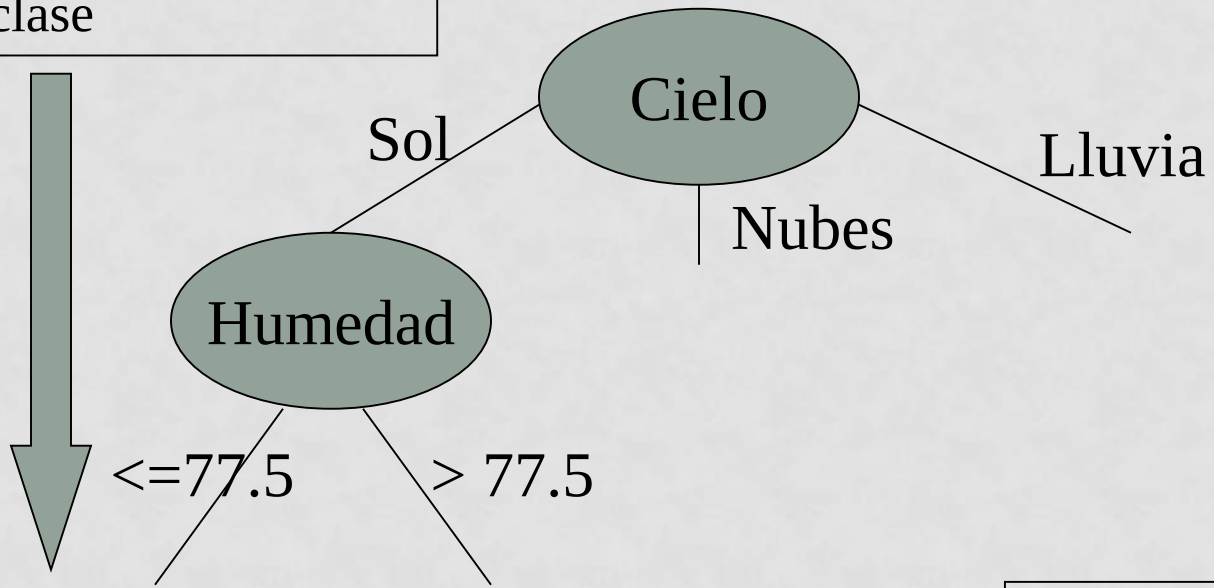
"3 No, 2 Si"

"0 No, 4 Si"

"2 No, 3 Si"

Construcción recursiva del árbol

Ya no es necesario seguir subdividiendo porque todos los datos son de la misma clase

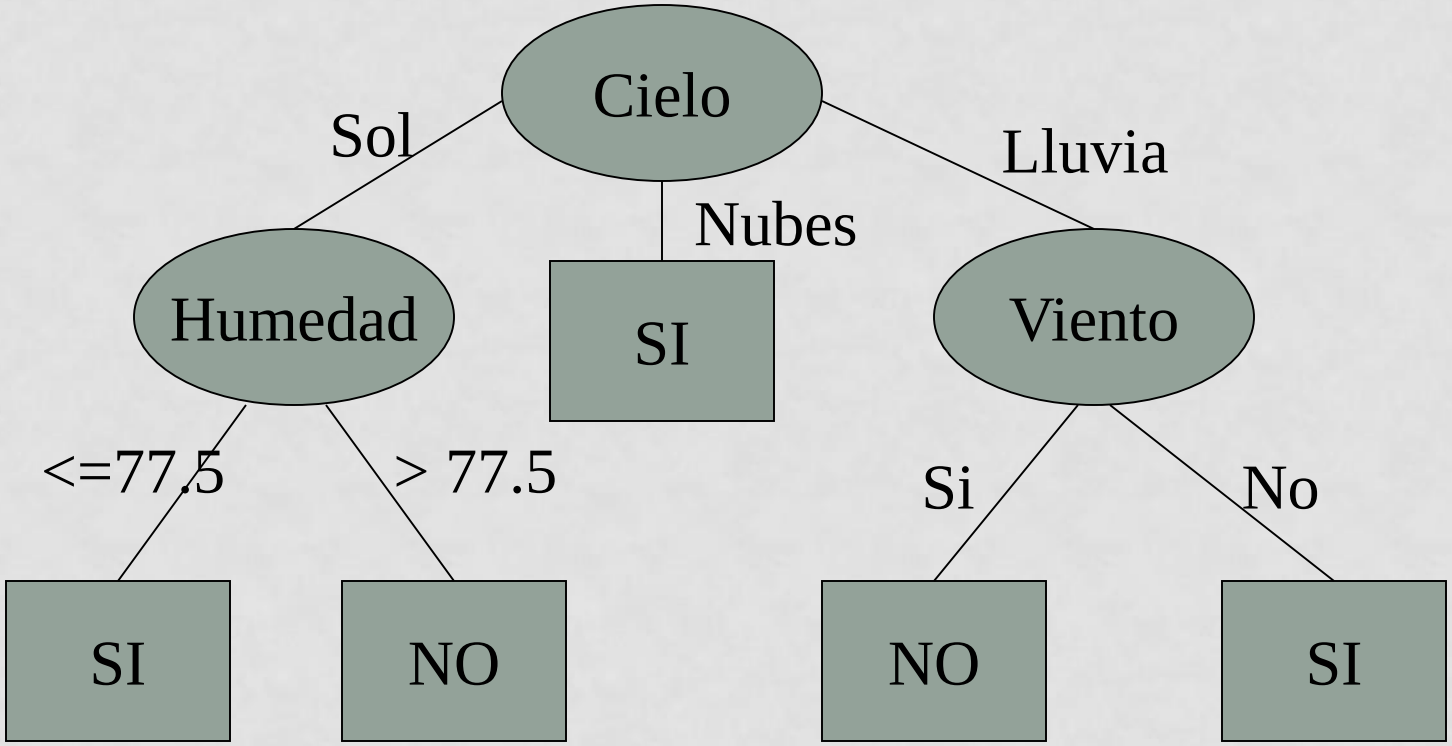


T	H	V	Ten
69	70	No	Si
75	70	Si	Si

T	H	V	Ten
85	85	No	No
80	90	Si	No
72	95	No	No

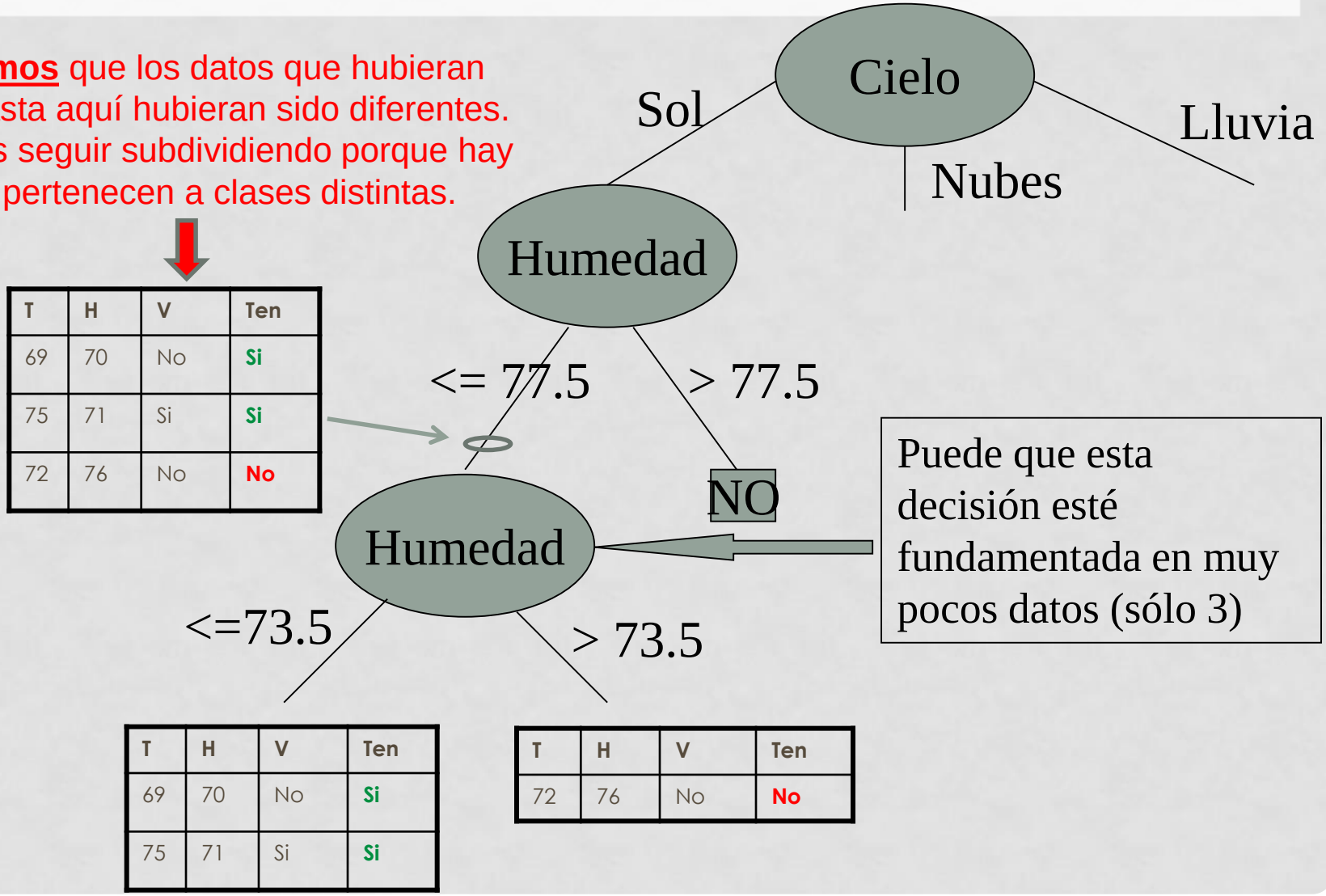
Ya no es necesario seguir subdividiendo porque todos los datos son de la misma clase

Árbol definitivo



¿Por qué no seguir subdividiendo?

Supongamos que los datos que hubieran llegado hasta aquí hubieran sido diferentes. Podríamos seguir subdividiendo porque hay datos que pertenecen a clases distintas.



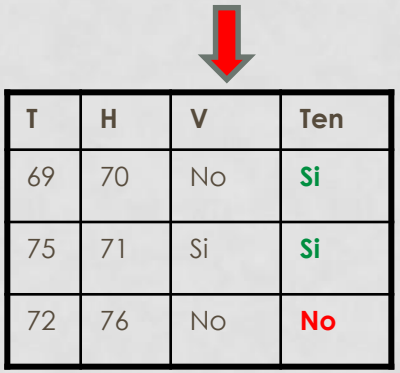
T	H	V	Ten
69	70	No	Si
75	71	Si	Si
72	76	No	No

T	H	V	Ten
69	70	No	Si
75	71	Si	Si

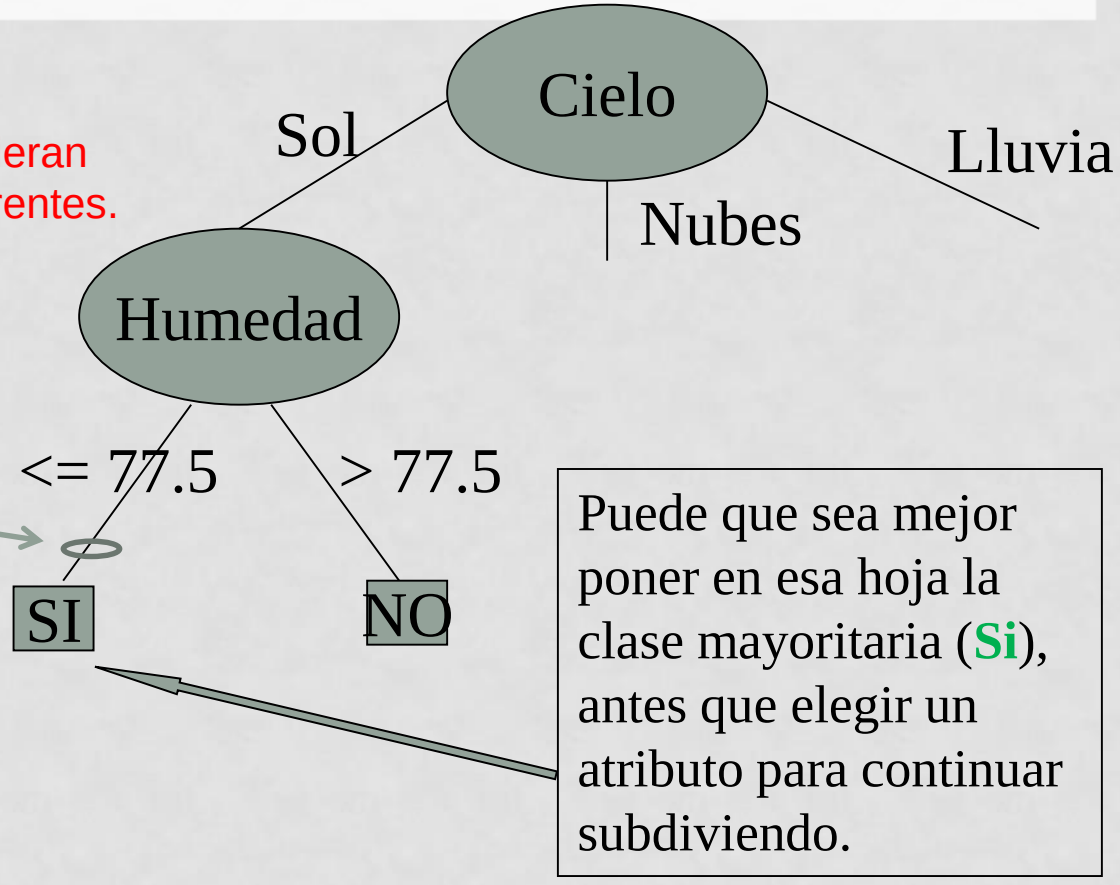
T	H	V	Ten
72	76	No	No

¿Por qué no seguir subdividiendo?

Supongamos que los datos que hubieran llegado hasta aquí hubieran sido diferentes.



T	H	V	Ten
69	70	No	Si
75	71	Si	Si
72	76	No	No

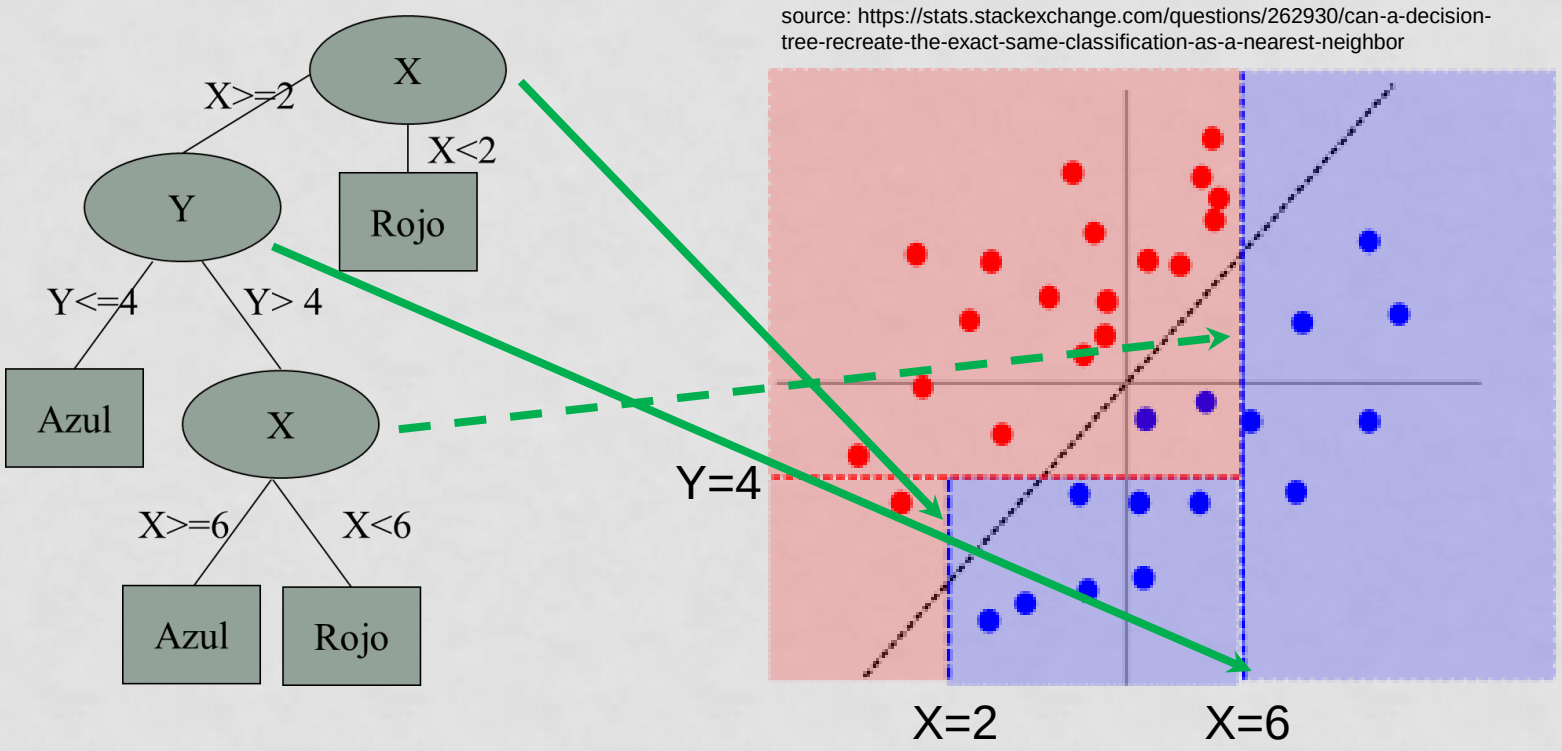


Puede que sea mejor poner en esa hoja la clase mayoritaria (**Si**), antes que elegir un atributo para continuar subdiviendo.

El algoritmo usa un criterio estadístico para determinar si la muestra es demasiado pequeña, y si es mejor continuar subdividiendo o poner una hoja con la clase mayoritaria.

C4.5 (J48) Tipo de clasificador

- Es no lineal. Las fronteras de separación entre las clases son rectas paralelas a los ejes
- No muy bueno para fronteras oblicuas (pero esto se puede solucionar con ensembles). También existen "oblique trees"



source: https://stats.stackexchange.com/questions/262930/can-a-decision-tree-recreate-the-exact-same-classification-as-a-nearest-neighbor

Reglas (creadas a partir del árbol de decisión)

Se obtiene una regla por cada camino de la raíz a las hojas

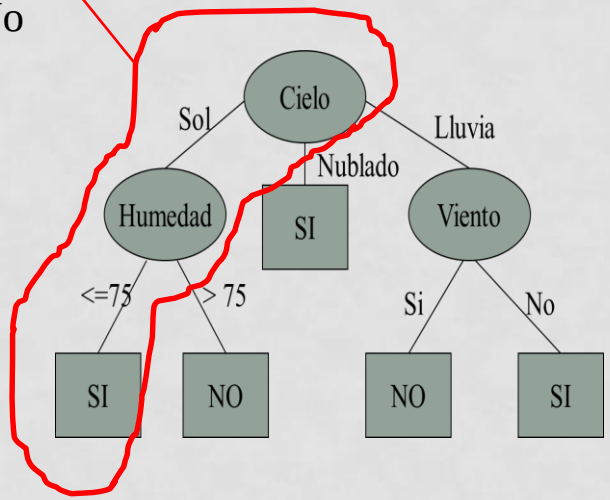
IF Cielo = Sol **AND** Humedad <= 75 **THEN** Tenis = Si

ELSE IF Cielo = Sol AND Humedad > 75 **THEN** Tennis = No

ELSE IF Cielo = Nublado **THEN** Tenis = Si

ELSE IF Cielo = Lluvia AND Viento = Si **THEN** Tenis = Si

ELSE Tenis = No

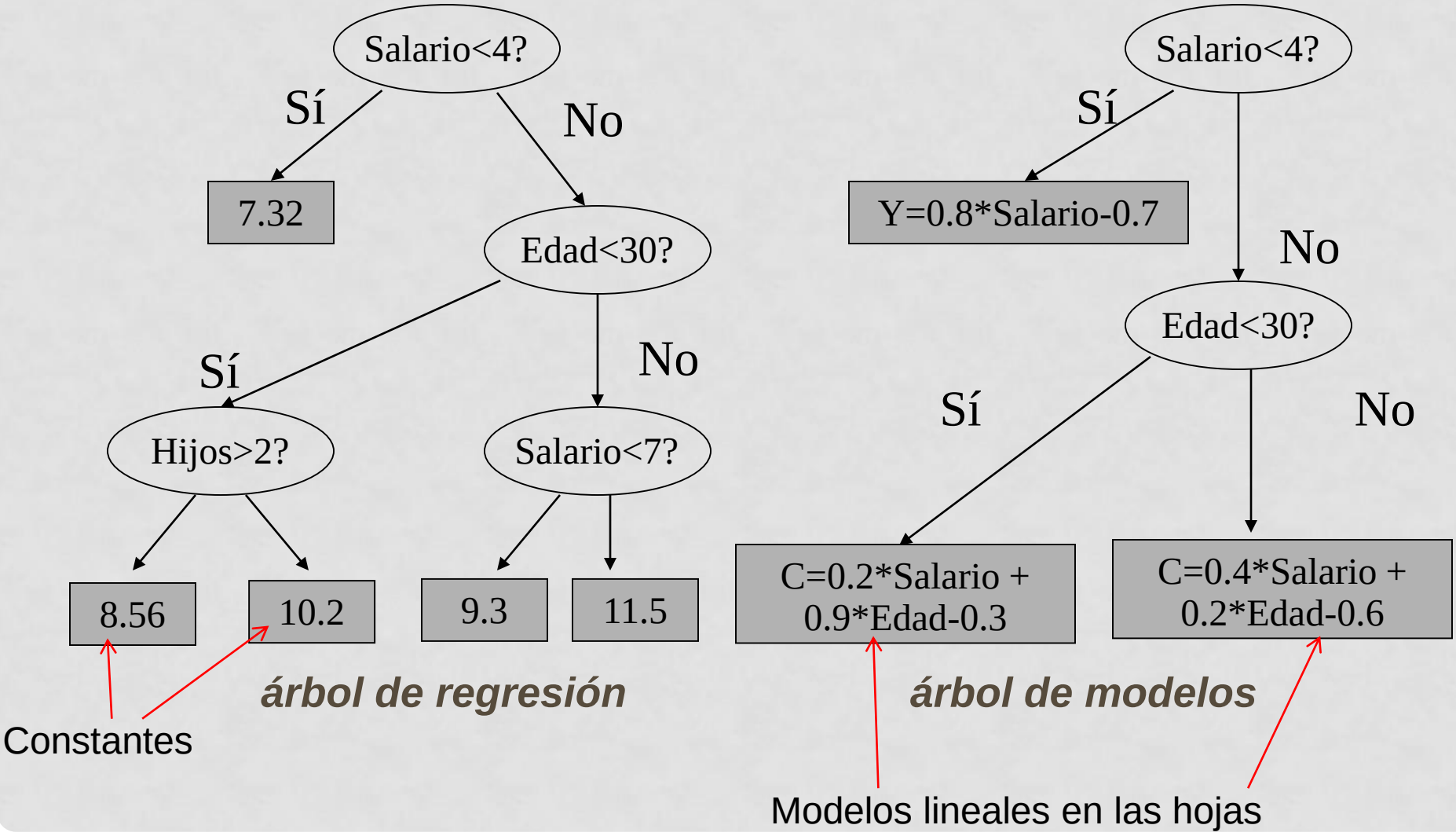


También hay algoritmos que construyen directamente reglas (PART)

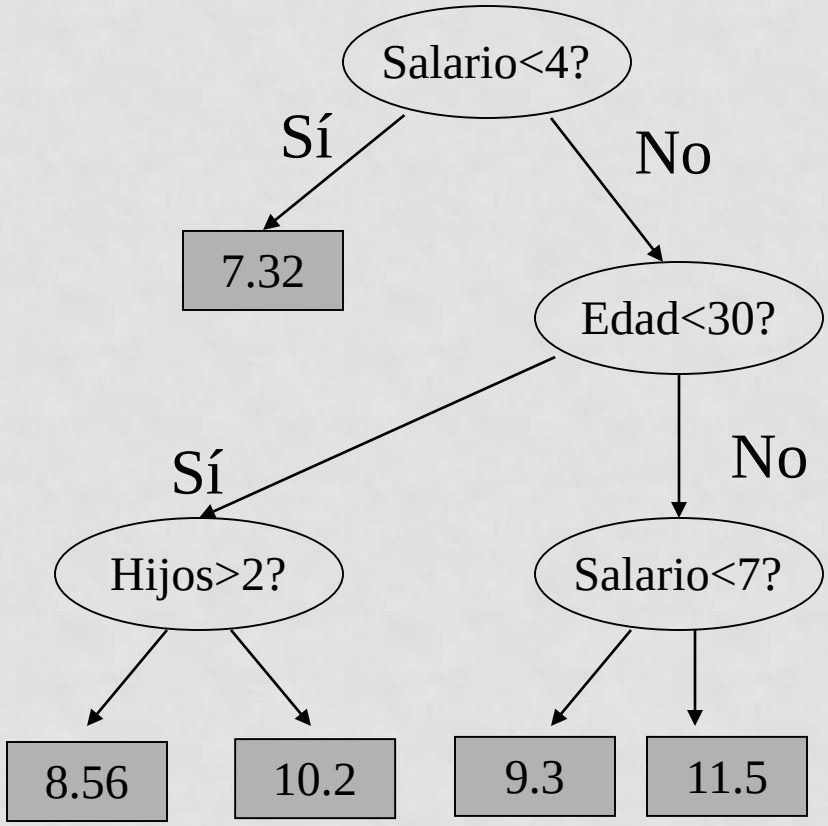
ÁRBOLES PARA PREDICCIÓN NUMÉRICA

- ¿Qué hacer si la etiqueta es contínua?
  - Respuesta: reducción de varianza en lugar de reducción de entropía
- Dos tipos:
  - Árboles de modelos
  - Árboles de regresión

ÁRBOLES PARA PREDICCIÓN NUMÉRICA



ÁRBOLES PARA PREDICCIÓN NUMÉRICA

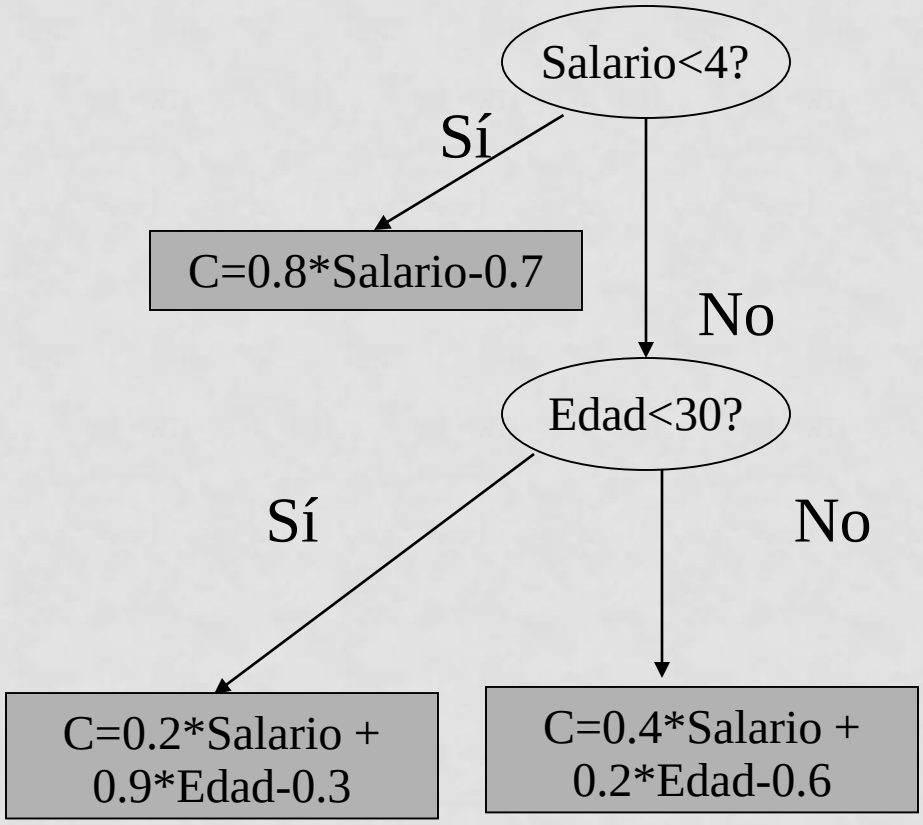


árbol de regresión

En las hojas, pondremos la media de las salidas de los datos que hayan llegado a cada hoja.

ÁRBOLES PARA PREDICCIÓN NUMÉRICA

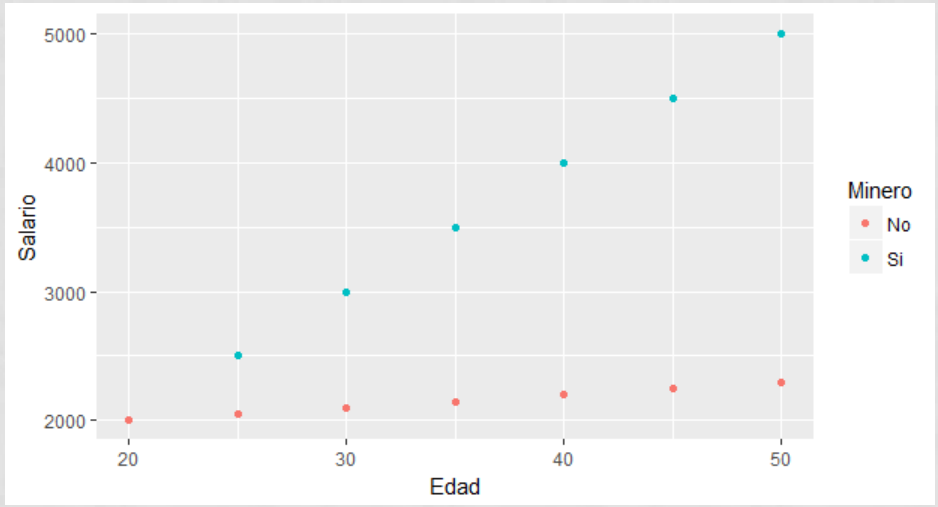
En cada hoja, construiremos un modelo de regresión lineal con los datos que hayan llegado a esa hoja.



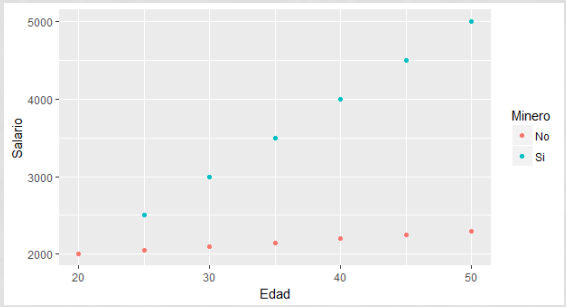
árbol de modelos

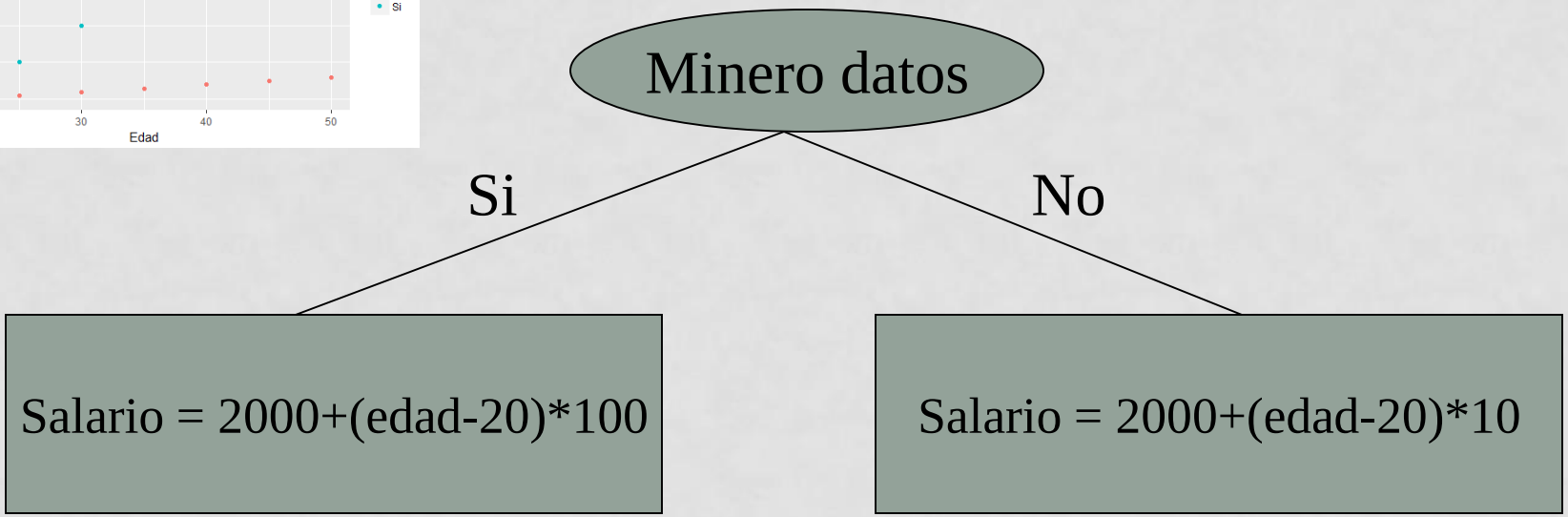
ÁRBOLES DE MODELOS. EJEMPLO

Minero datos	Edad	Salario
Si	20	2000
Si	25	2500
Si	30	3000
Si	35	3500
Si	40	4000
Si	45	4500
Si	50	5000
No	20	2000
No	25	2050
No	30	2100
No	35	2150
No	40	2200
No	45	2250
No	50	2300

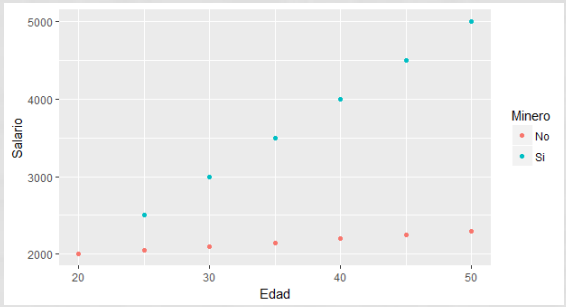


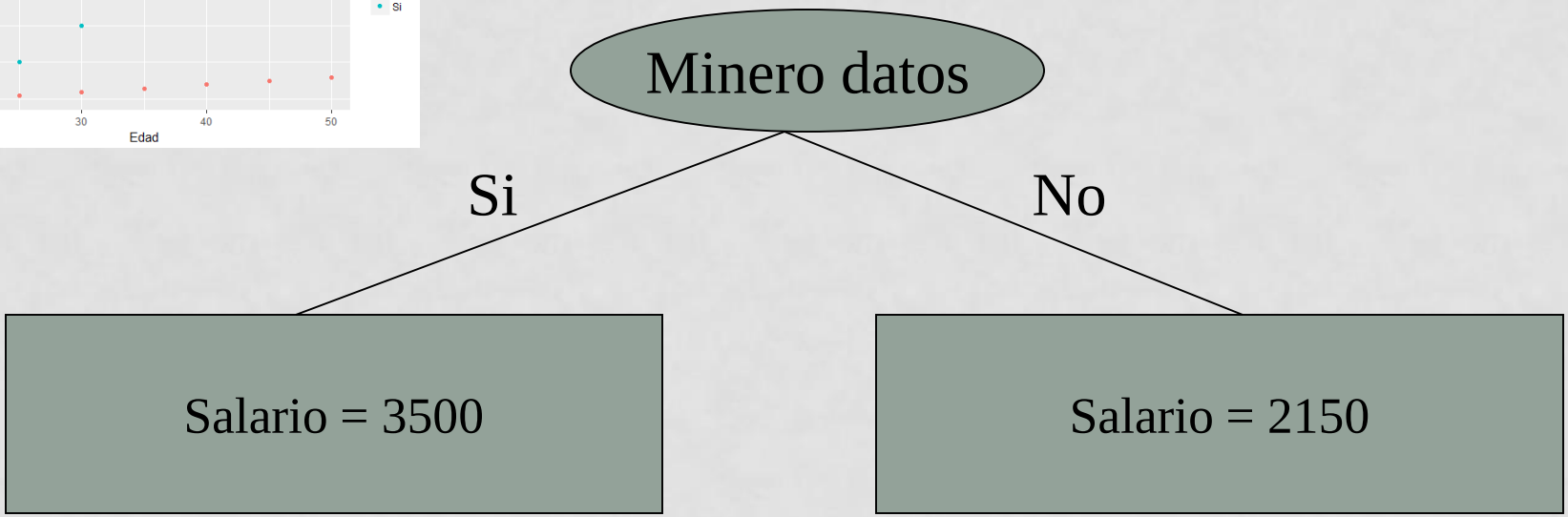
ÁRBOLES DE MODELOS. EJEMPLO





ÁRBOLES DE REGRESIÓN. EJEMPLO





En las hojas, salario medio para minero de datos (3500 euros) y salario medio para no mineros de datos (2150 euros)

ÁRBOLES PARA PREDICCIÓN NUMÉRICA

- Ambos se construyen de manera similar pero:
  - Para árboles de regresión se calcula la media en las hojas
  - Para árboles de modelos se construyen modelos lineales (M5' (Quinlan, 93))

- Ambos se construyen de manera similar, minimizando la **varianza o desviación estándar** (en vez de la entropía)

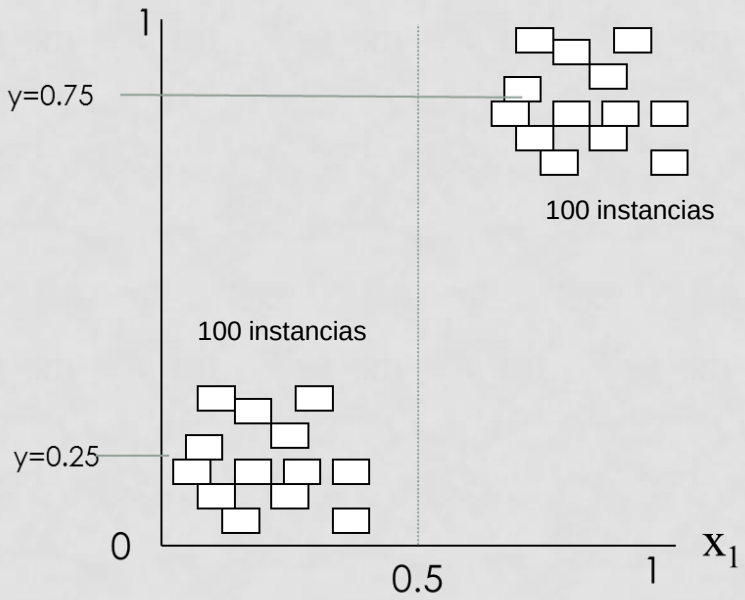
¿CUÁL ES EL MEJOR NODO PARA PONER EN LA RAÍZ?

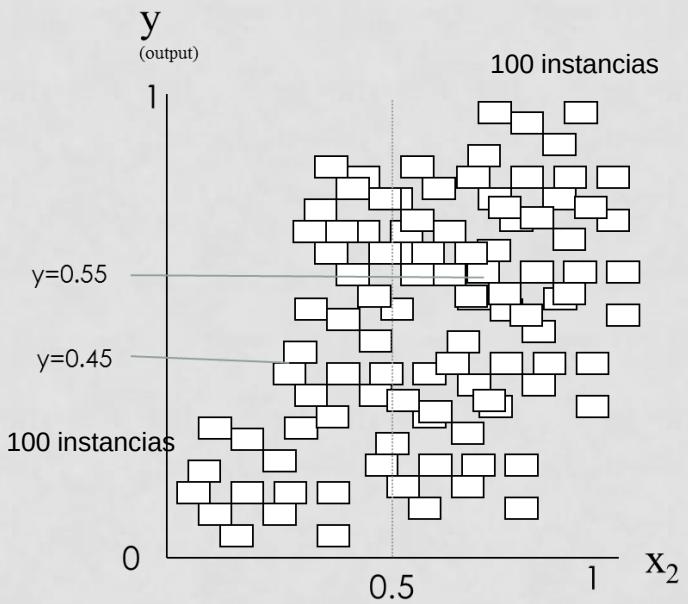
¿Qué atributo es mejor?

¿x_1 o x_2?

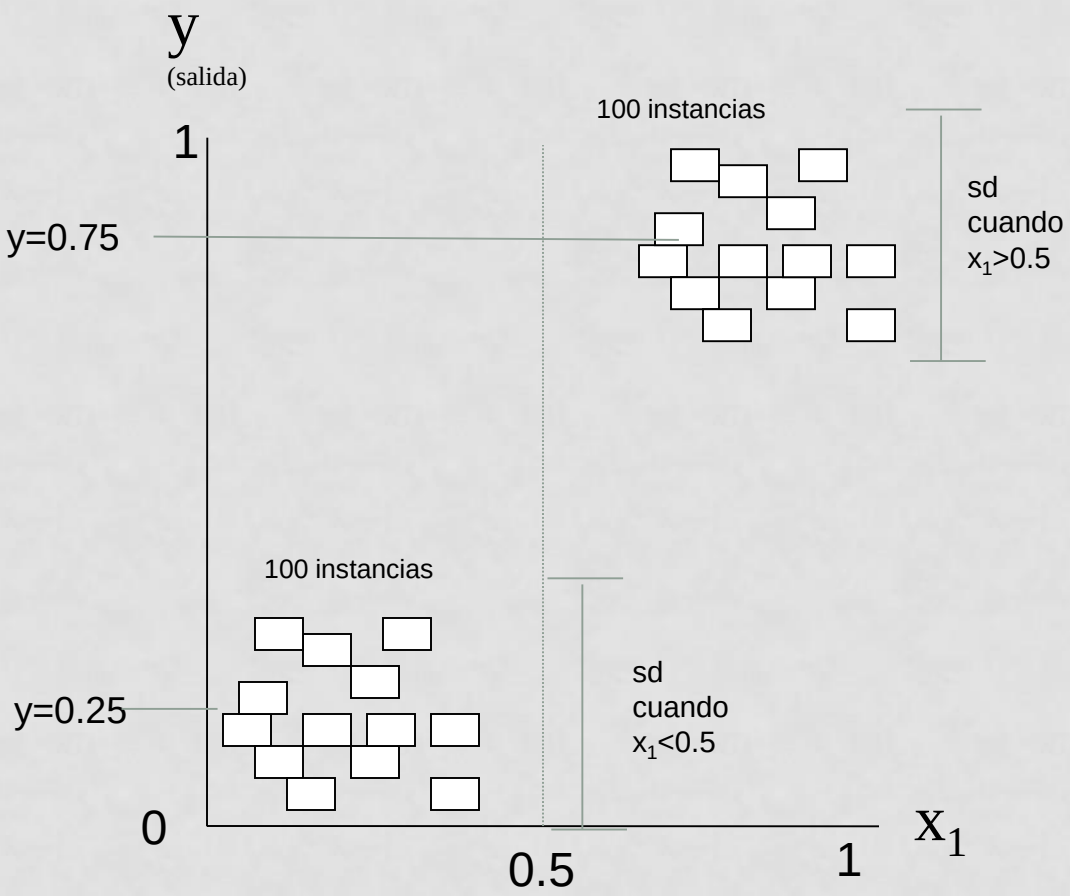
Elegiremos el atributo cuya *sd* media **tras la partición** es pequeña:

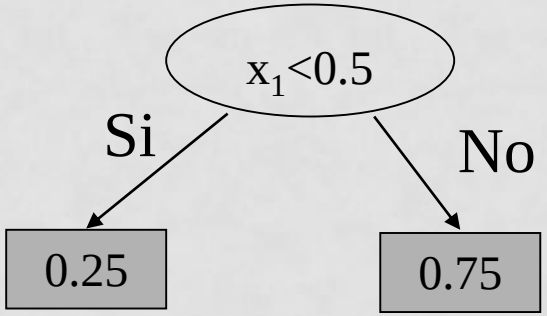
100/200 * *sd*(x_i<0.5) + 100/200 * *sd*(x_i>0.5)





¿CUÁL ES EL MEJOR NODO PARA PONER EN LA RAÍZ?



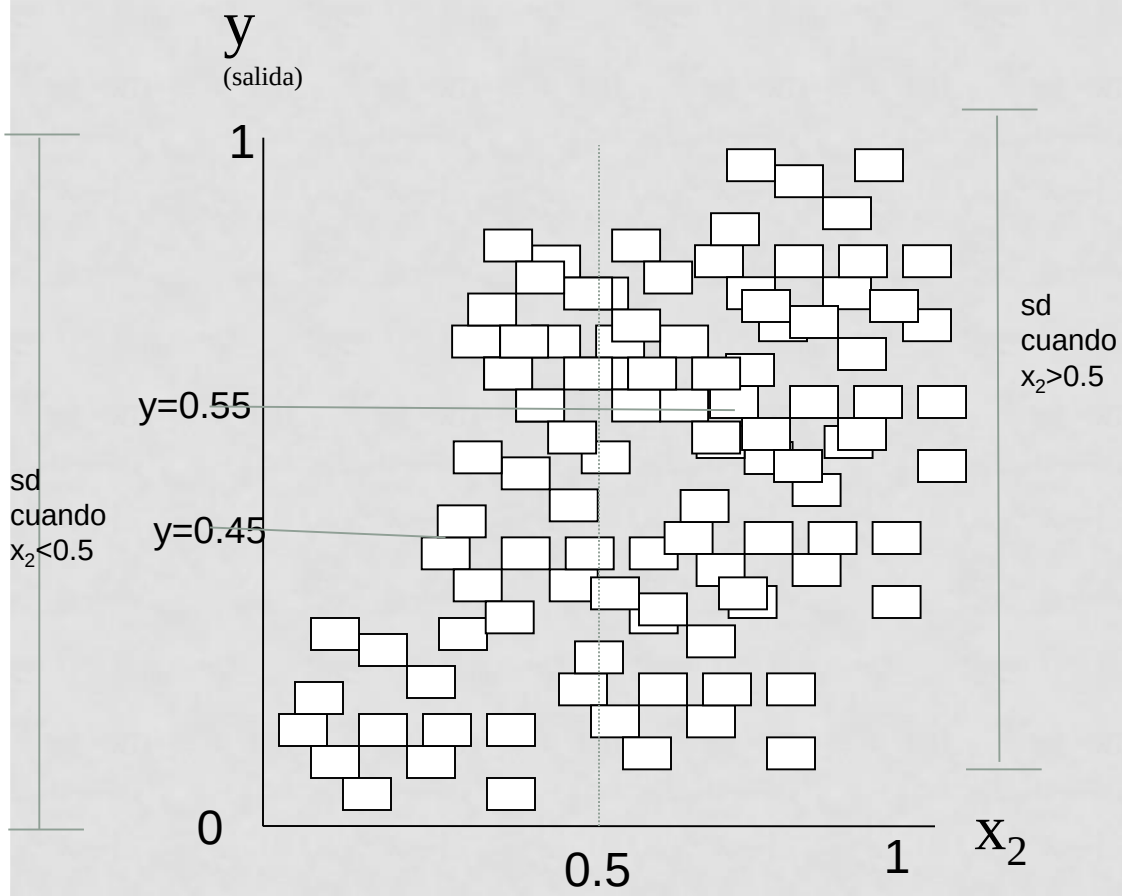


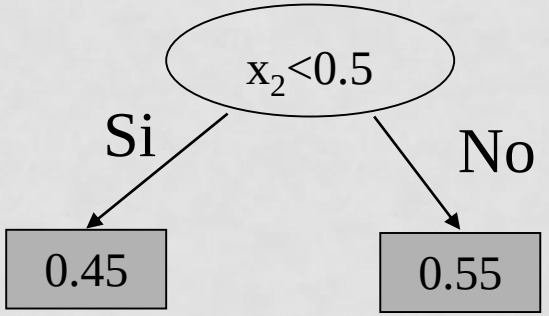
Se puede ver que x_1 es bastante predictivo

Las instancias tras la partición están concentradas

½*sd(x_1<0.5) + ½* sd(x_1>0.5) es pequeña

¿CUÁL ES EL MEJOR NODO PARA PONER EN LA RAÍZ?





x_2 no es demasiado predictivo

Las instancias tras la partición están muy esparcidas

½* sd(x_2<0.5) + ½* sd(x_2>0.5) es muy grande

¿CUÁNDO PARAR?

- El árbol se construye recursivamente, hasta que:
  - Hay pocos ejemplos en el nodo (2)
  - sd < 5% de la sd original

ALGORITMOS PARA ÁRBOLES PARA PREDICCIÓN NUMÉRICA

- Árboles de regresión: CART (Quinlan, 93). Disponible en R.
- Árboles de modelos: M5P (Quinlan, 93), Cubist (disponible en R, basado en reglas)
  - Wang, Y., & Witten, I. H. (1996). Induction of model trees for predicting continuous classes.