Aprendizaje Automático
Segundo Cuatrimestre de 2018

Árboles de Decisión

DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Aproximación de Funciones

Ejemplo:

- Los sábados a la mañana, un vecino a veces sale a caminar y a veces o no.
- Desconocemos su criterio para salir a caminar o no (función objetivo desconocida), pero sospechamos que depende del estado del tiempo:
  - Cielo: {Sol, Nublado, Lluvia}
  - Temperatura: {Calor, Templado, Frío}
  - Humedad: {Alta, Normal}
  - Viento: {Fuerte, Débil}
- Queremos aprender una función *Caminar* que aproxime al criterio del vecino:

$$Caminar : \text{Cielo} \times \text{Temperatura} \times \text{Humedad} \times \text{Viento} \rightarrow \{\text{Sí}, \text{No}\}$$

- Empezamos por juntar datos: registramos el comportamiento del vecino durante unas semanas...

atributos

clase

instancias

Cielo	Temperatura	Humedad	Viento	¿Camina?
Sol	Calor	Alta	Débil	No
Sol	Calor	Alta	Fuerte	No
Nublado	Calor	Alta	Débil	Sí
Lluvia	Templado	Alta	Débil	Sí
Lluvia	Frío	Normal	Débil	Sí
Lluvia	Frío	Normal	Fuerte	No
Nublado	Frío	Normal	Fuerte	Sí
Sol	Templado	Alta	Débil	No
Sol	Frío	Normal	Débil	Sí
Lluvia	Templado	Normal	Débil	Sí
Sol	Templado	Normal	Fuerte	Sí
Nublado	Templado	Alta	Fuerte	Sí
Nublado	Calor	Normal	Débil	Sí
Lluvia	Templado	Alta	Fuerte	No

Aproximación de Funciones

Marco del problema:

- Conjunto de instancias X. Cada instancia $x \in X$ tiene atributos.

  En nuestro ejemplo, X son los días, con atributos Cielo, Temp, Humedad, Viento.

- Función objetivo desconocida $f: X \rightarrow Y$

  f: Cielo x Temp x Humedad x Viento x ... $\rightarrow$ {Sí, No}

  f puede depender de otras cosas!

- Espacio de hipótesis $H = \{ h \mid h : X \rightarrow Y \}$

  Depende del algoritmo de aprendizaje, y está limitado por los atributos que tenemos de X.

Entrada del algoritmo de aprendizaje:

- Datos de entrenamiento $\{<x^{(i)}, y^{(i)}>\}$.

Salida del algoritmo de aprendizaje:

- Hipótesis (o modelo) $h \in H$ que aproxima a la función f.

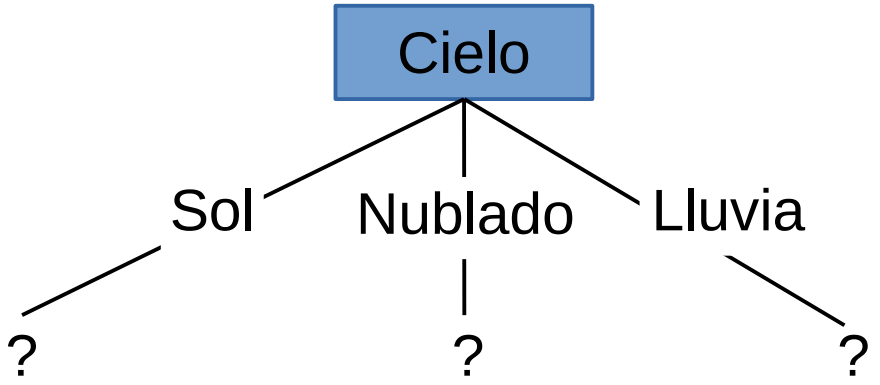
atributos

clase

instancias

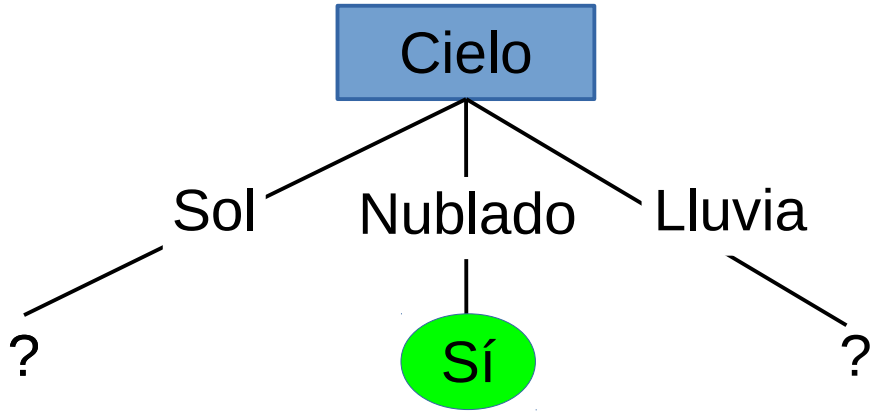
Cielo	Temperatura	Humedad	Viento	¿Camina?
Sol	Calor	Alta	Débil	No
Sol	Calor	Alta	Fuerte	No
Nublado	Calor	Alta	Débil	Sí
Lluvia	Templado	Alta	Débil	Sí
Lluvia	Frío	Normal	Débil	Sí
Lluvia	Frío	Normal	Fuerte	No
Nublado	Frío	Normal	Fuerte	Sí
Sol	Templado	Alta	Débil	No
Sol	Frío	Normal	Débil	Sí
Lluvia	Templado	Normal	Débil	Sí
Sol	Templado	Normal	Fuerte	Sí
Nublado	Templado	Alta	Fuerte	Sí
Nublado	Calor	Normal	Débil	Sí
Lluvia	Templado	Alta	Fuerte	No

Construcción de un Árbol de Decisión



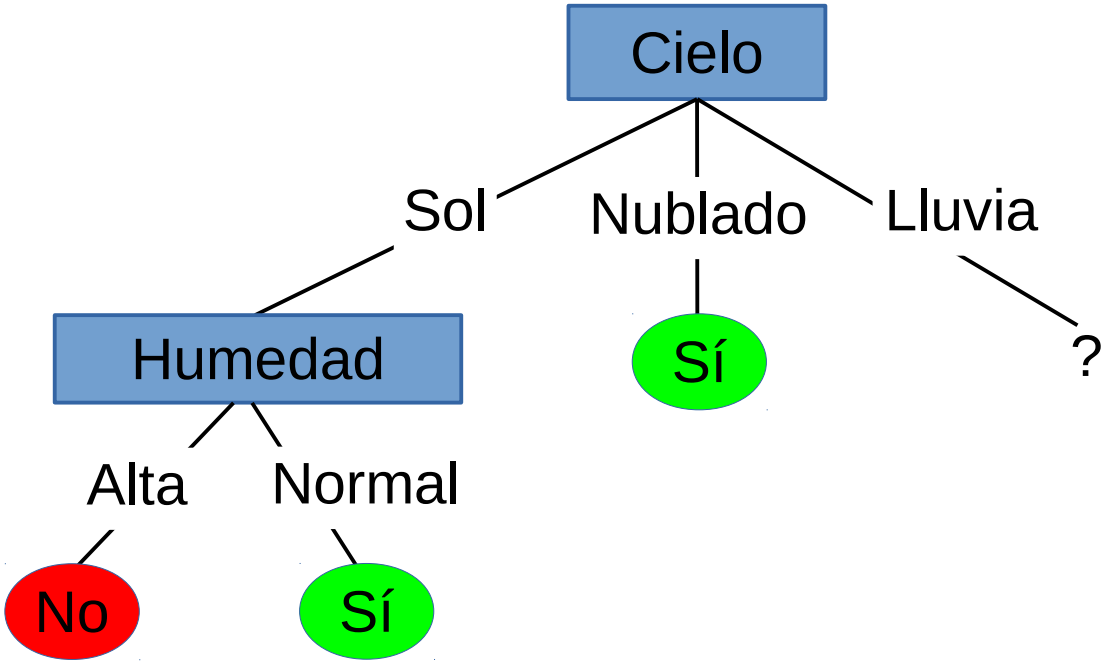
El atributo **Cielo** parece ser bueno para comenzar el árbol…

Construcción de un Árbol de Decisión



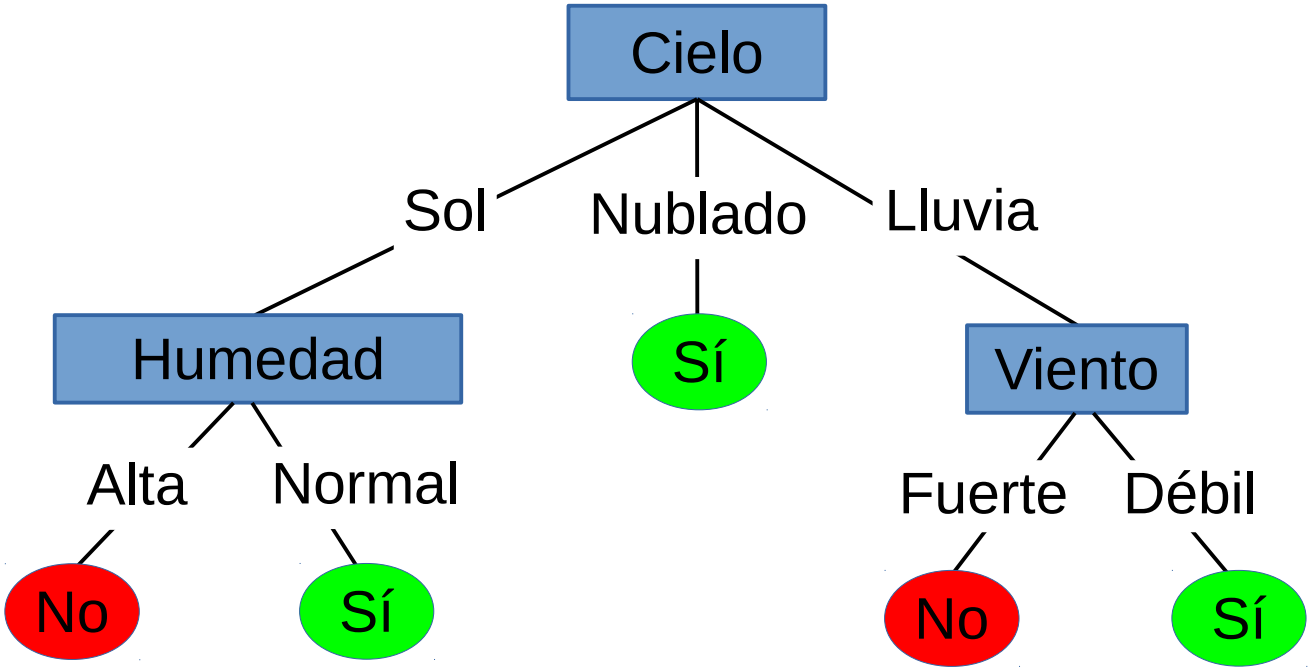
Las instancias con **Cielo**==Nublado son todas positivas.

Construcción de un Árbol de Decisión



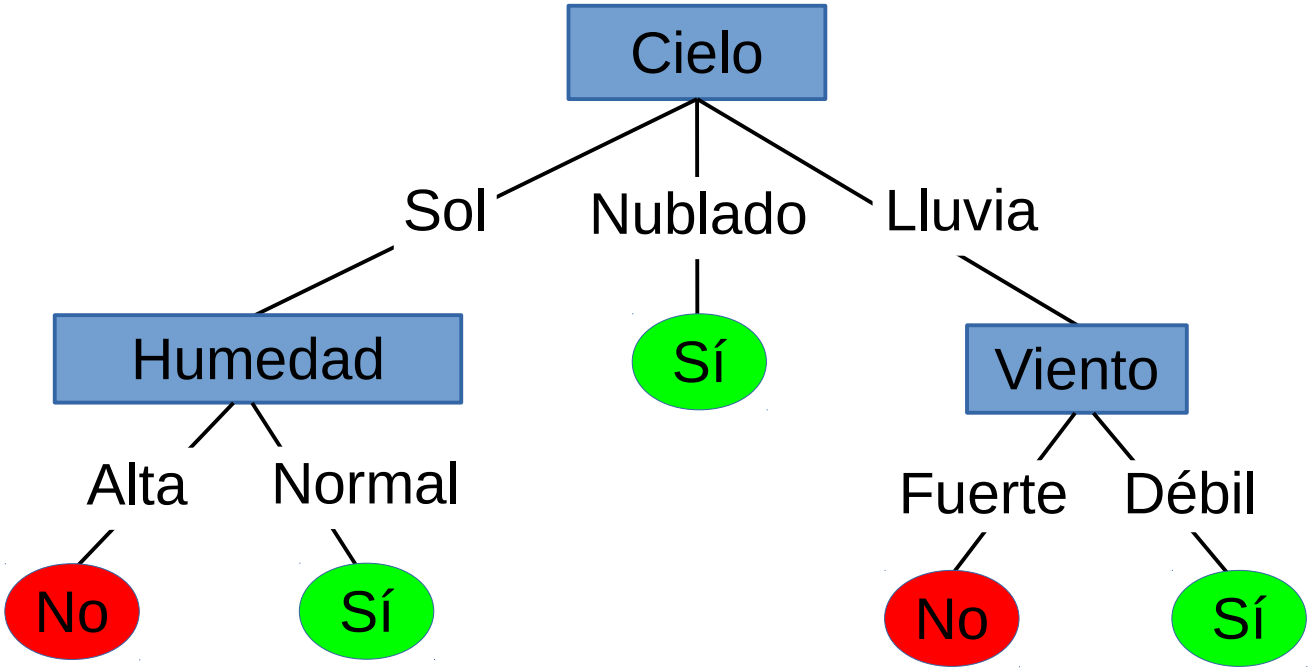
Para las instancias con **Cielo**==Sol continuamos
con el atributo **Humedad**, que separa perfectamente.

Construcción de un Árbol de Decisión



Para las instancias con **Cielo**==Lluvia,
el atributo **Viento** separa perfectamente.

Árboles de Decisión

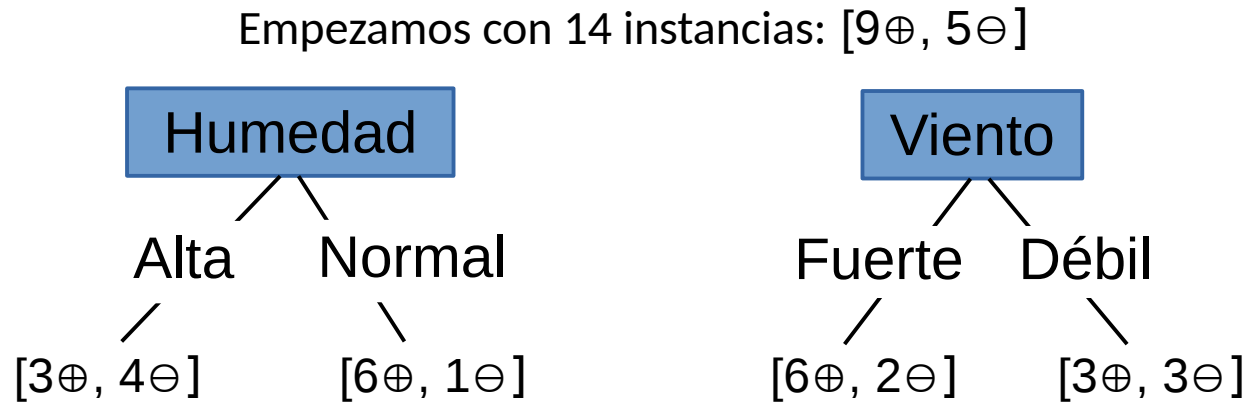


- $h : \langle X_1, \ldots, X_p \rangle \rightarrow Y$
- Cada nodo interno evalúa un atributo discreto X_i
- Cada rama corresponde a un valor para X_i
- Cada hoja predice un valor de Y

Inducción *Top-Down* de Árboles de Decisión
(ID3[(a)] y C4.5[(b)], Quinlan)

1) *A* ← el "mejor" atributo para *nodo_actual*.
2) Asignar *A* como atributo de decisión del *nodo_actual*.
3) Para cada valor de *A*, crear un nuevo hijo del *nodo_actual*.
4) Clasificar (*repartir*) las instancias en los nuevos nodos, según el valor de *A*.
5) Si las instancias están clasificadas "suficientemente bien": FIN.
   Si no: iterar sobre los nuevos nodos.

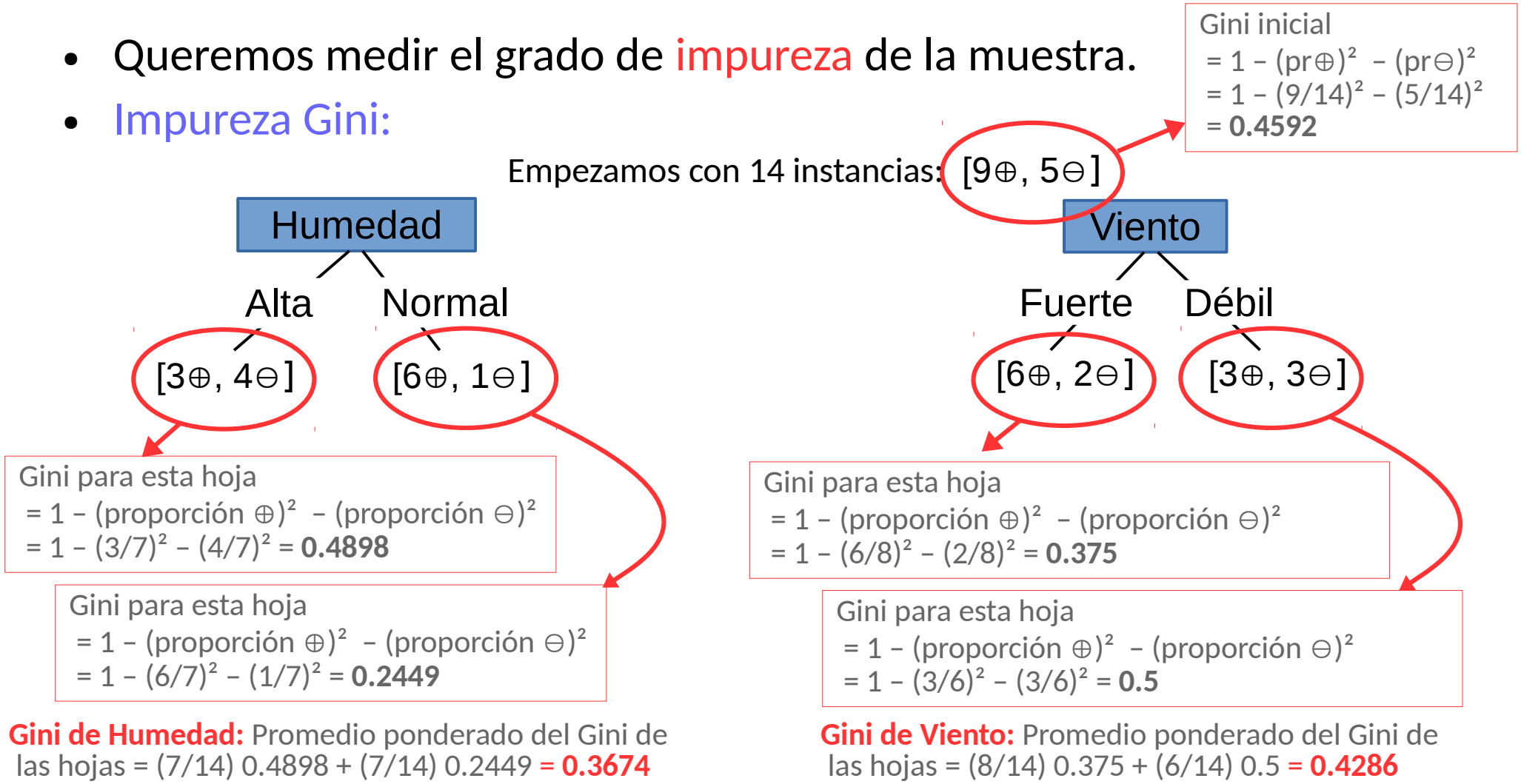
¿Cuál es el "mejor" atributo?



(a) J.R. Quinlan, "Induction of Decision Trees", Machine Learning, 1(1):81-106, 1986.
(b) J.R. Quinlan, "Simplifying Decision Trees", Intl. Journal of Human-Computer Studies, 51(2):497–510, 1999.

¿Cuál es el mejor atributo?
Opción 1: Impureza Gini

- Queremos medir el grado de impureza de la muestra.
- Impureza Gini:



- Elegimos el atributo con mayor reducción de impureza (Gini Gain):

  Humedad: 0.4592 – 0.3674 = **0.0918** Viento: 0.4592 – 0.4286 = **0.0306**

¿Cuál es el mejor atributo?
Opción 1: Impureza Gini

- Queremos medir el grado de impureza de la muestra.
- Impureza Gini:
  - Impureza de una muestra *S*:

$$Gini(S) = 1 - \sum_{c \in Clases} \left(\frac{|S_c|}{|S|} \right)^2$$

S_c es el conjunto de instancias que pertenecen a la clase c.

  - Reducción de impureza de una muestra *S* con respecto a un atributo *A*:

$$GiniGain(S, A) = Gini(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Gini(S_v)$$

Valores(*A*) es el conjunto de valores posibles del atributo *A*.

$S_v = \{s \in S \mid A(s) = v\}$

- Elegimos el atributo con mayor reducción de impureza (Gini Gain).

¿Cuál es el mejor atributo?
Opción 2: Ganancia de Información

- Entropía de una muestra *S* con respecto a la variable objetivo:

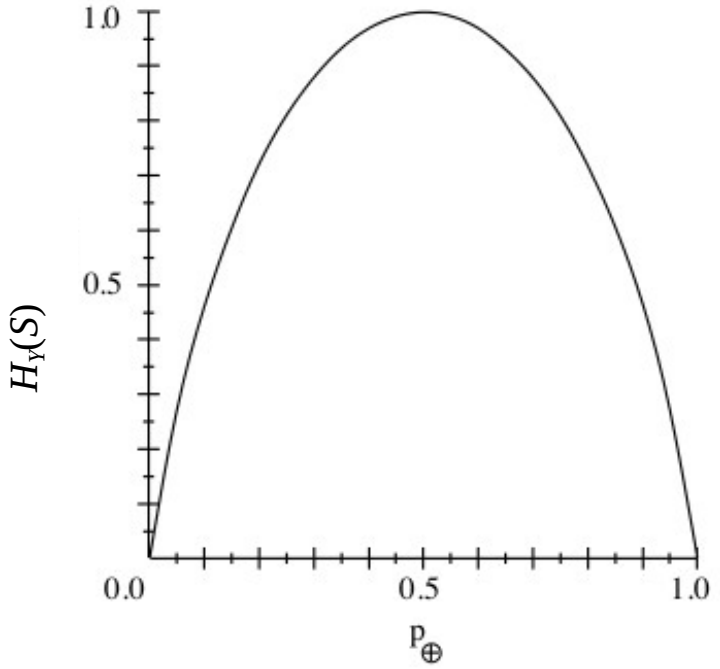
$$H(S) = \sum_{c \in Clases} -p_c \log_2 p_c$$

p_c : proporción de instancias en *S* pertenecientes a la clase *c*

- La entropía es otra forma de medir el grado de impureza de *S*.

- Ejemplo: *c*=2

  $H(S) = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$



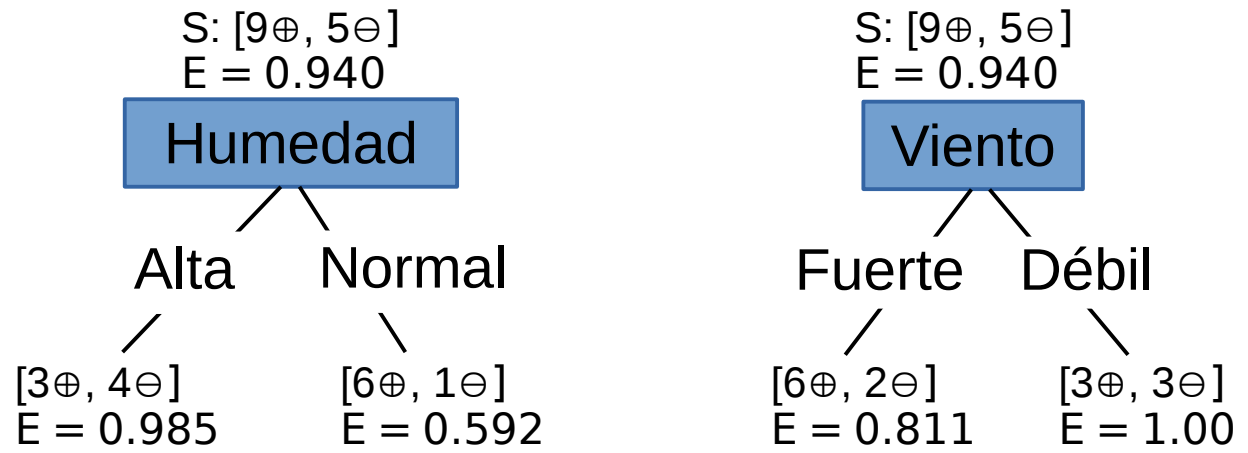
¿Cuál es el mejor atributo?
Opción 2: Ganancia de Información

- Es la reducción de entropía de la muestra *S* (respecto de la variable objetivo *Y*), después de clasificar las instancias según *A*.

$$InfoGain(S, A) = H(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} H(S_v)$$

Valores(*A*) : conjunto de valores posibles del atributo *A*.

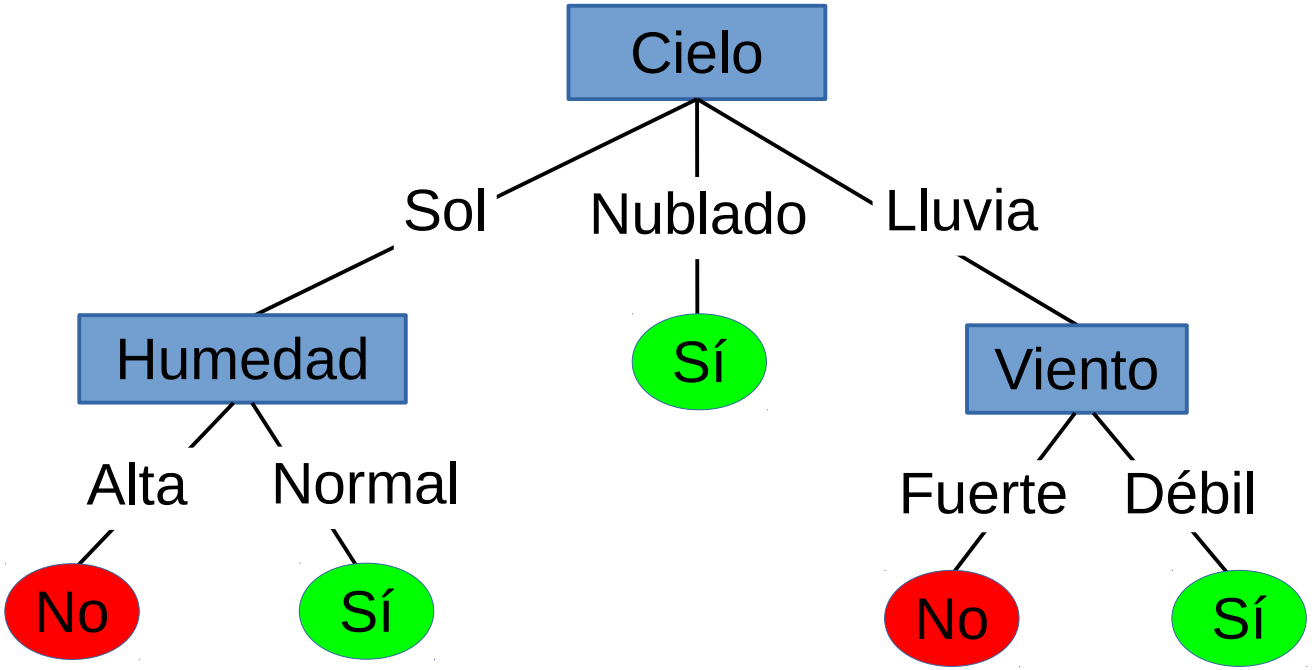
$S_v = \{s \in S | A(s) = v\}$



InfoGain(*S*, Humedad) = .940 – (7/14) .985 – (7/14) .592 = 0.151
InfoGain(*S*, Viento) = .940 – (8/14) .811 – (6/14) 1.00 = 0.048

- Otra métrica: *Gain Ratio* (corrige preferencia de *InfoGain* por atributos con demasiados valores)

Inducción *Top-Down* de Árboles de Decisión
(ID3, C4.5, CART, etc.)



- Entonces, en cada nodo elegimos el atributo que más reduce la impureza de las submuestras de sus hijos.

Inducción *Top-Down* de Árboles de Decisión
(ID3, C4.5, CART, etc.)

- Complejidad temporal (*n*: #instancias, *p*: #atributos)
  - Construcción: $O(n\ p^2)$ peor caso[(a)], $O(n\ p)$ promedio[(b)]
  - Consulta: $O(p)$
- Espacio de hipótesis:
  - Fórmulas lógicas de valores discretos. En principio puede construirse cualquier árbol.
- Sesgo inductivo:
  - Construcción de árboles cada vez más complejos.
  - Hill-climbing sin backtracking (converge a un máximo local).
  - Atributos más informativos → cerca de la raíz.

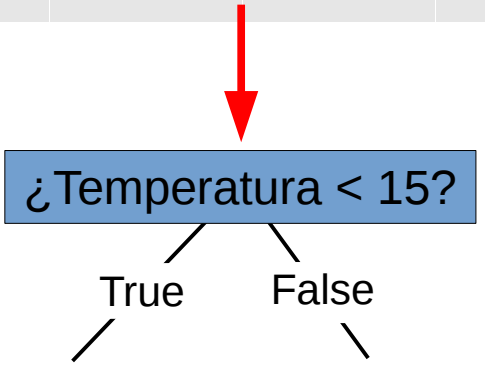
(a) P. E. Utgoff. "Incremental induction of decision trees". Machine Learning, 4(2):161–186, 1989

(b) J. W. Shavlik, R. J. Mooney, and G. Towell. "Symbolic and neural learning algorithm: An experimental comparison". Machine Learning, 6:111–143, 1991.

Atributos Numéricos

- ¿Qué pasa si tenemos un atributo numérico *A*?
- Buscamos un umbral *c*, para discriminar según *A*<*c*.
- ¿Cómo elegir *c*?
  1) Ordenar las instancias según A.
  2) Buscar la forma de partir la lista que maximice la reducción de impureza.

Temperatura:	10	12	18	21	28	31
¿Camina?:	No	No	Sí	Sí	Sí	No



Para pensar...

- ¿Cuán robustos son los Árboles de Decisión ante...

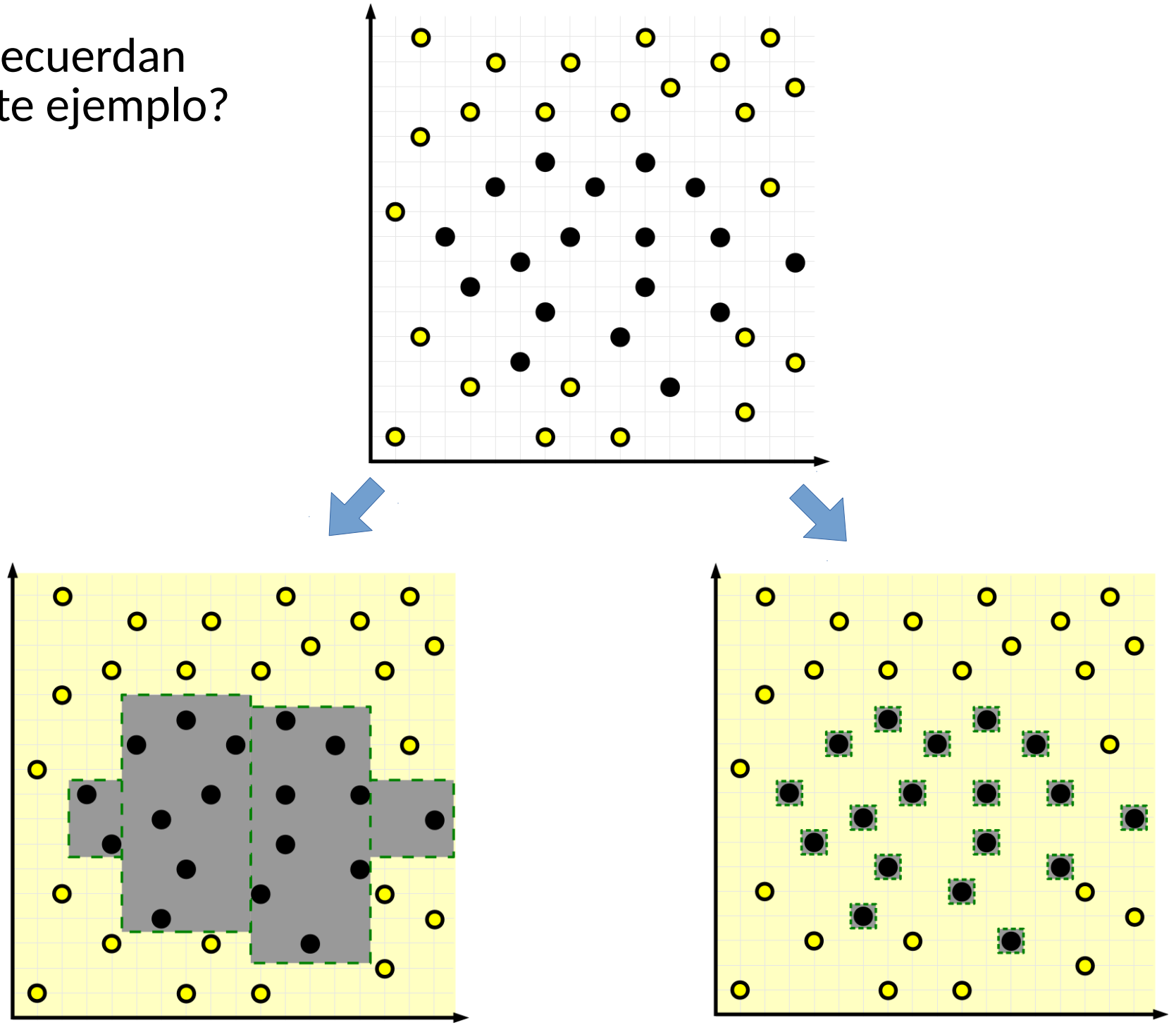
  ...atributos faltantes?

  – Instancias de entrenamiento con valores indefinidos en algunos atributos
  – Ej: datos clínicos de un paciente incompletos

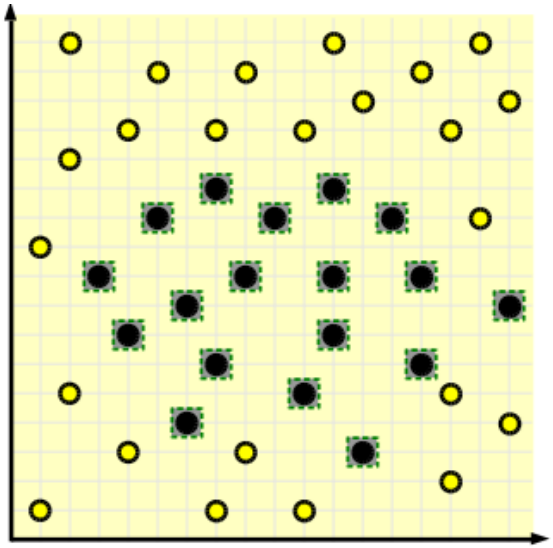
  ...datos ruidosos?

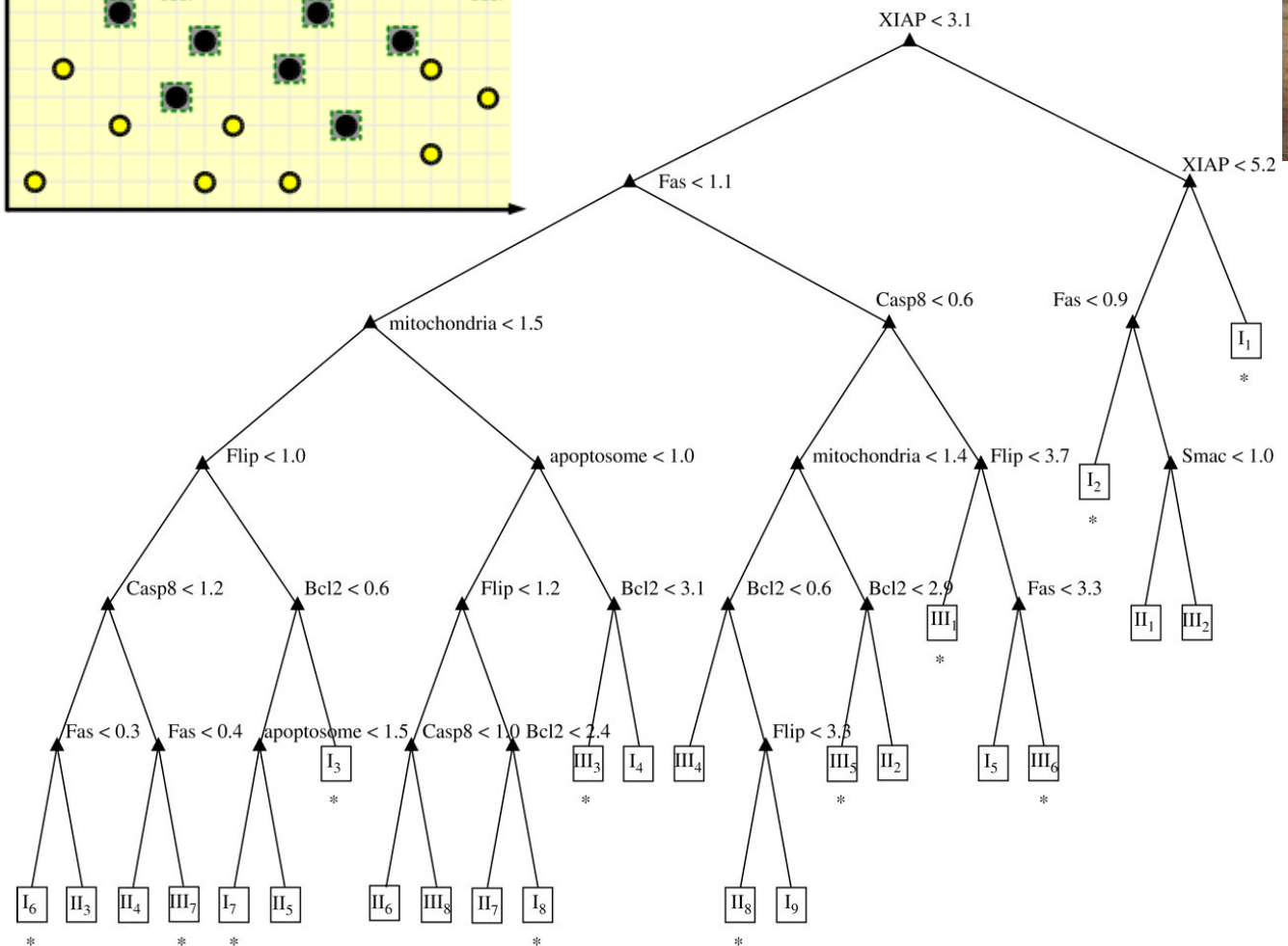
  – Instancias de entrenamiento mal clasificadas
  – Ej: errores cometidos al ingresar datos manualmente

¿Recuerdan
este ejemplo?



Sobreajuste (*Overfitting*)





En árboles de decisión, el sobreajuste se produce cuando el árbol se hace "demasiado" profundo.

En un caso extremo, el camino de la raíz a una hoja sería una descripción perfecta de una única instancia (recordar a *Funes el memorioso*).

Sobreajuste *(Overfitting)*

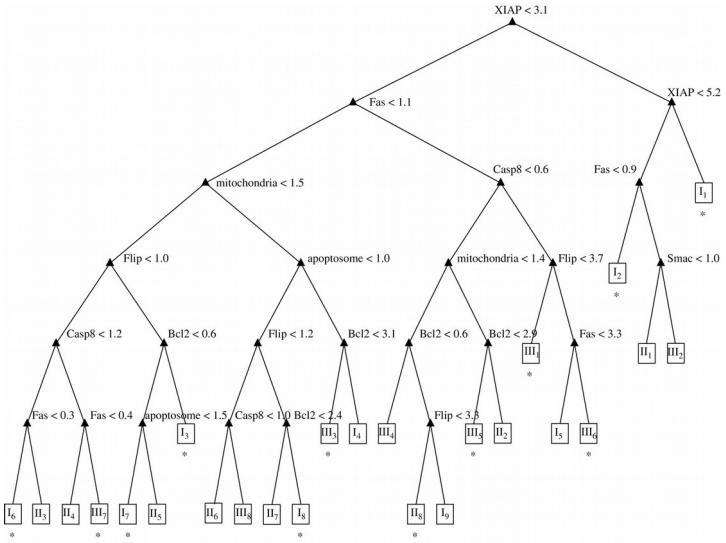
- Considerar el error de un modelo M sobre:
  - D (instancias de entrenamiento): $\text{error}_D(M)$
  - X (todas las instancias posibles): $\text{error}_X(M)$
- Definición: Un modelo M_1 sobreajusta a los datos de entrenamiento si existe otro modelo M_2 tal que

$$\text{error}_D(M_1) < \text{error}_D(M_2)$$

$$\text{error}_X(M_1) > \text{error}_X(M_2)$$

- O sea: M_1 es mejor sobre D, pero M_2 generaliza mejor.

Sobreajuste en Árboles



- Soluciones:
  - Criterio de parada
    - No construir más allá de cierta profundidad.
  - Pruning (poda)
    - Construir el árbol entero; podar las ramas cuando ello mejore la performance *sobre datos separados*.
  - Rule post-pruning
    - Construir el árbol entero; convertir árbol a reglas; sacar precondiciones de las reglas cuando ello mejore su performance *sobre datos separados*; reordenar las reglas según accuracy.

Resumen

- Árboles de decisión: construcción y consulta.
- Métricas para evaluar atributos: impureza Gini.
- Espacio de hipótesis. Sesgo inductivo. Complejidad temporal.
- Atributos discretos y numéricos.
- Robustez ante datos faltantes y ruidosos.
- Sobreajuste *(overfitting)*
  - Sobreajuste en árboles: criterios de parada; poda.