5.2 Árboles de decisión

Inteligencia Artificial

3er curso INF

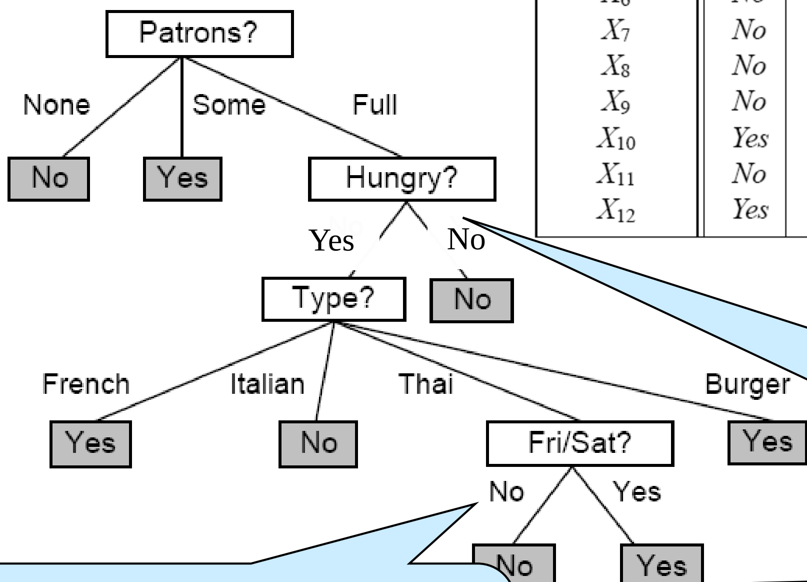
Escuela Politécnica Superior

Árbol de decisión

Datos de entrenamiento

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

Árbol aprendido:



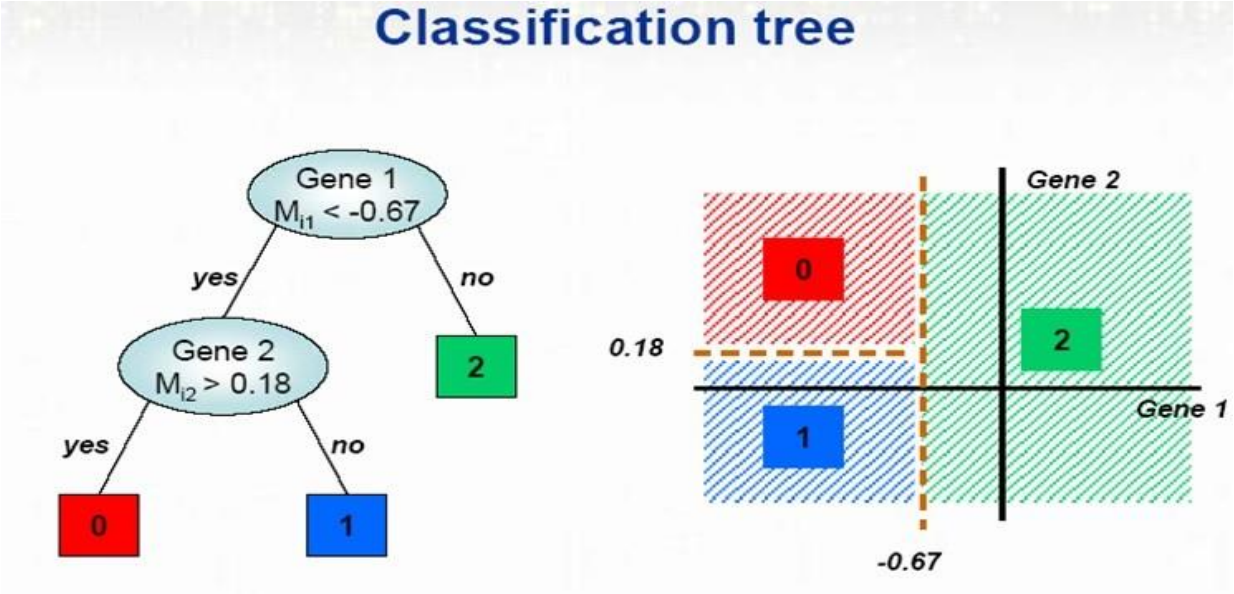
Instancias para las cuales *Patrons = full* están asociados a este nodo

En nodos internos los ejemplos están separados de acuerdo a una prueba en un atributo

Instancias que alcanzan este nodo de la hoja se clasifican como WillWait = Sí

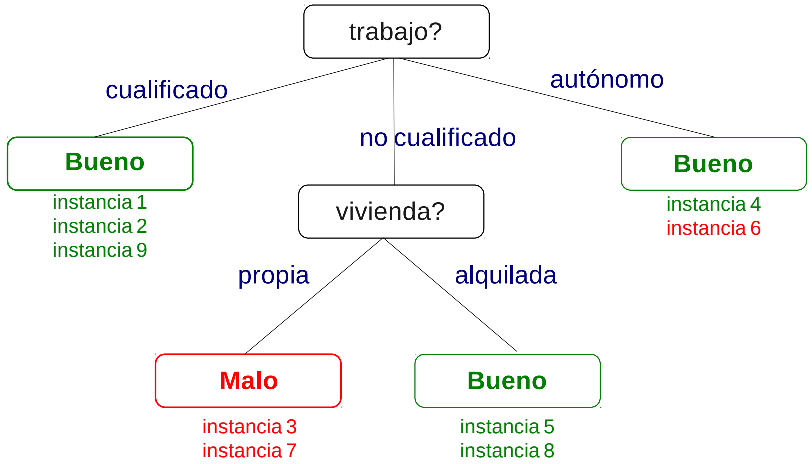
Árbol de decision: Classification and Regression Tree (CART)

- Particiona el espacio de la muestra en rectángulos y luego predice un modelo simple en cada uno de ellos
- Los arboles binarios van discriminando el espacio en dos submuestras (nodos) a partir de una anterior



Ejemplo

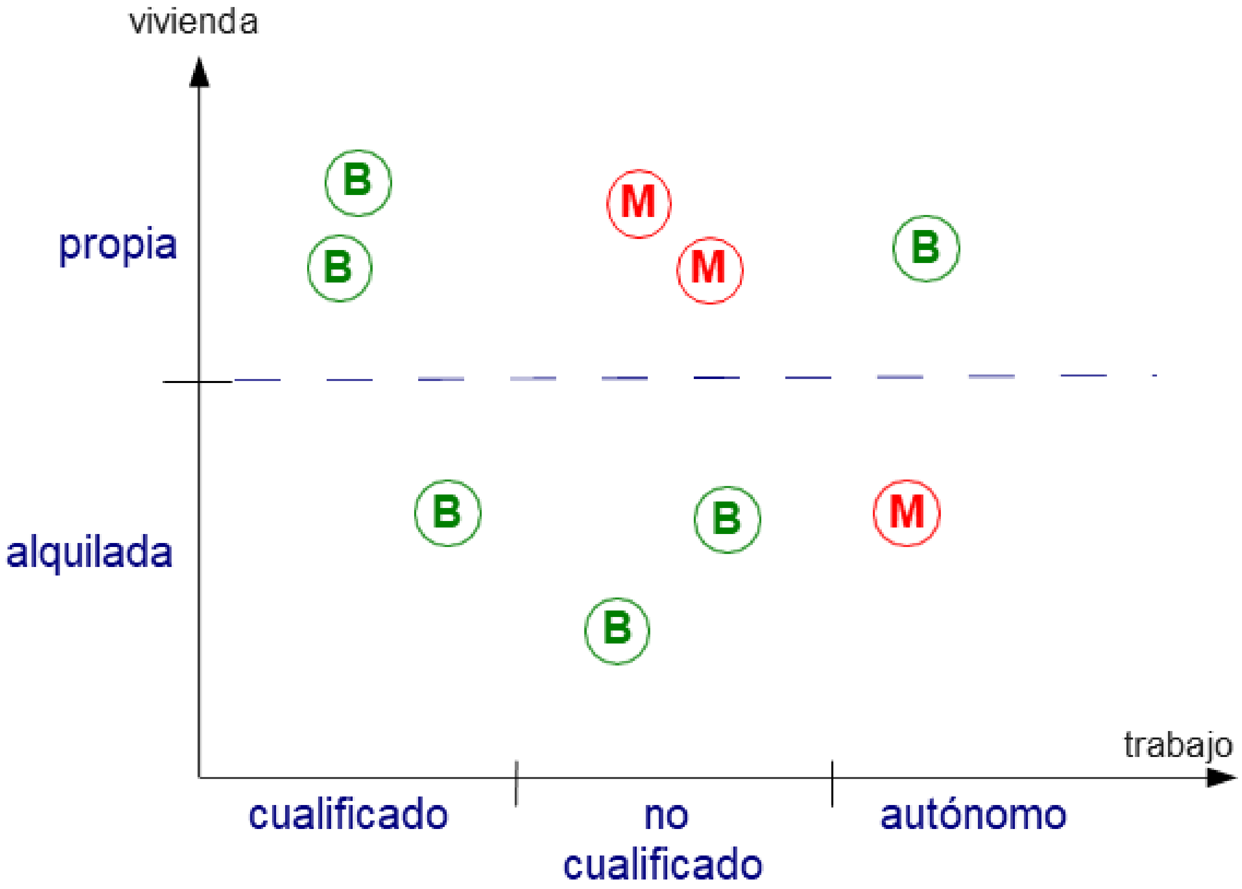
	edad	estado civil	ahorros	formación	trabajo	vivienda	cantidad	clase
1	35	soltero	7,000	básica	cua	propia	50K	bueno
2	23	casado	2,000	f.p	cua	alquilada	70K	bueno
3	30	casado	1,000	básica	No cua	propia	60K	malo
4	26	soltero	15,000	univ.	aut	propia	120K	bueno
5	50	divorciado	3,500	univ.	No cua	alquilada	40K	bueno
6	43	soltero	N.S.	básica	aut	N.S	30K	malo
7	31	divorciado	28,000	máster o +	No cua	propia	90K	malo
8	33	casado	N.S.	univ.	No cua	alquilada	30K	bueno
9	40	soltero	11,000	máster o +	cua	propia	100K	bueno



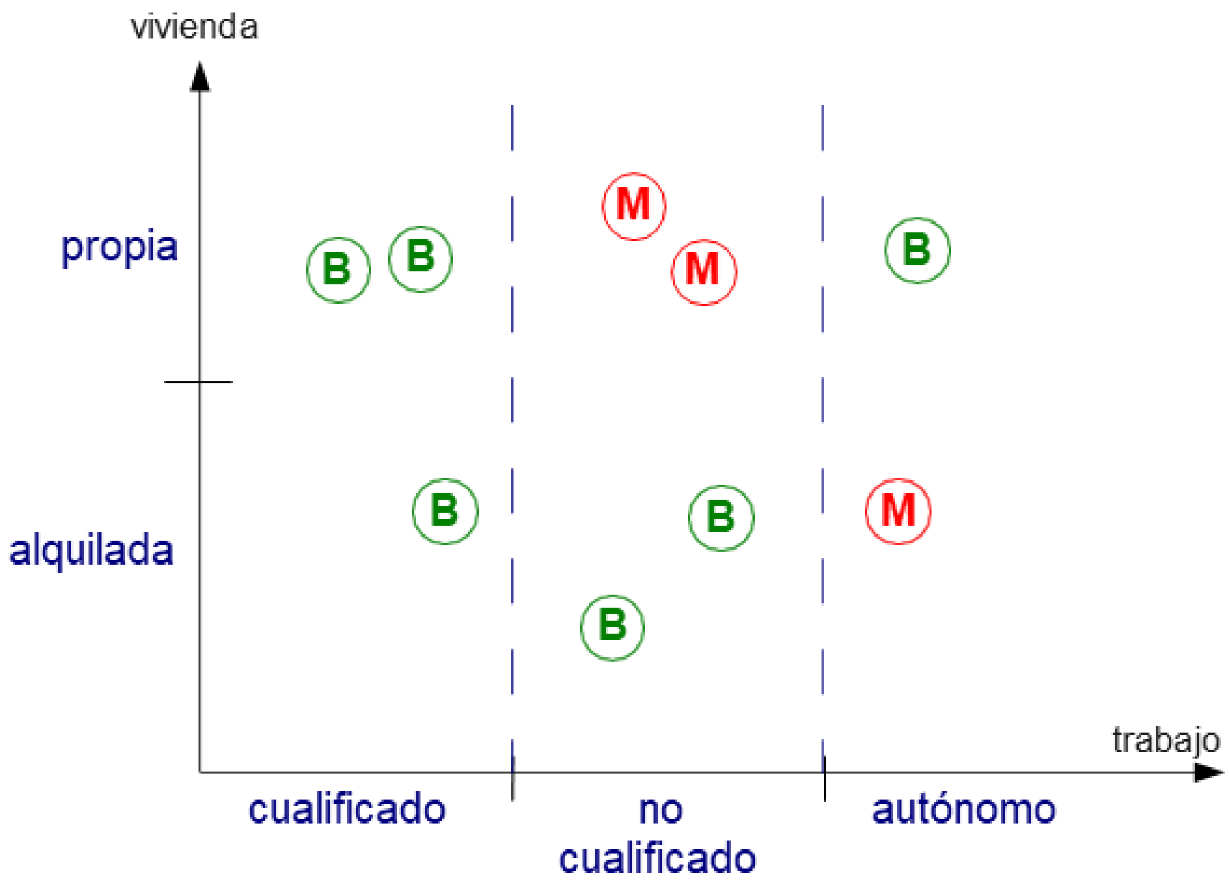
Conceptos

- Diagrama que representa condiciones sucesivas sobre los atributos para clasificar una instancia
- Tipos de nodos
  - Nodos internos:
    - Preguntas condicionales sobre los atributos
    - Cada respuesta sigue a un arco o flecha
    - Separación completa de los ejemplos entre las posibles respuestas
  - Nodos hoja
    - Clase → predicción
    - confianza de predicción
    - ejemplos de entrenamiento que cumplieron las condiciones hasta el nodo hoja
- Objetivos del modelado
  - Construir el árbol más sencillo que mejor separe las instancias por clase
  - el modelo final debe generalizar para clasificar bien futuras instancias

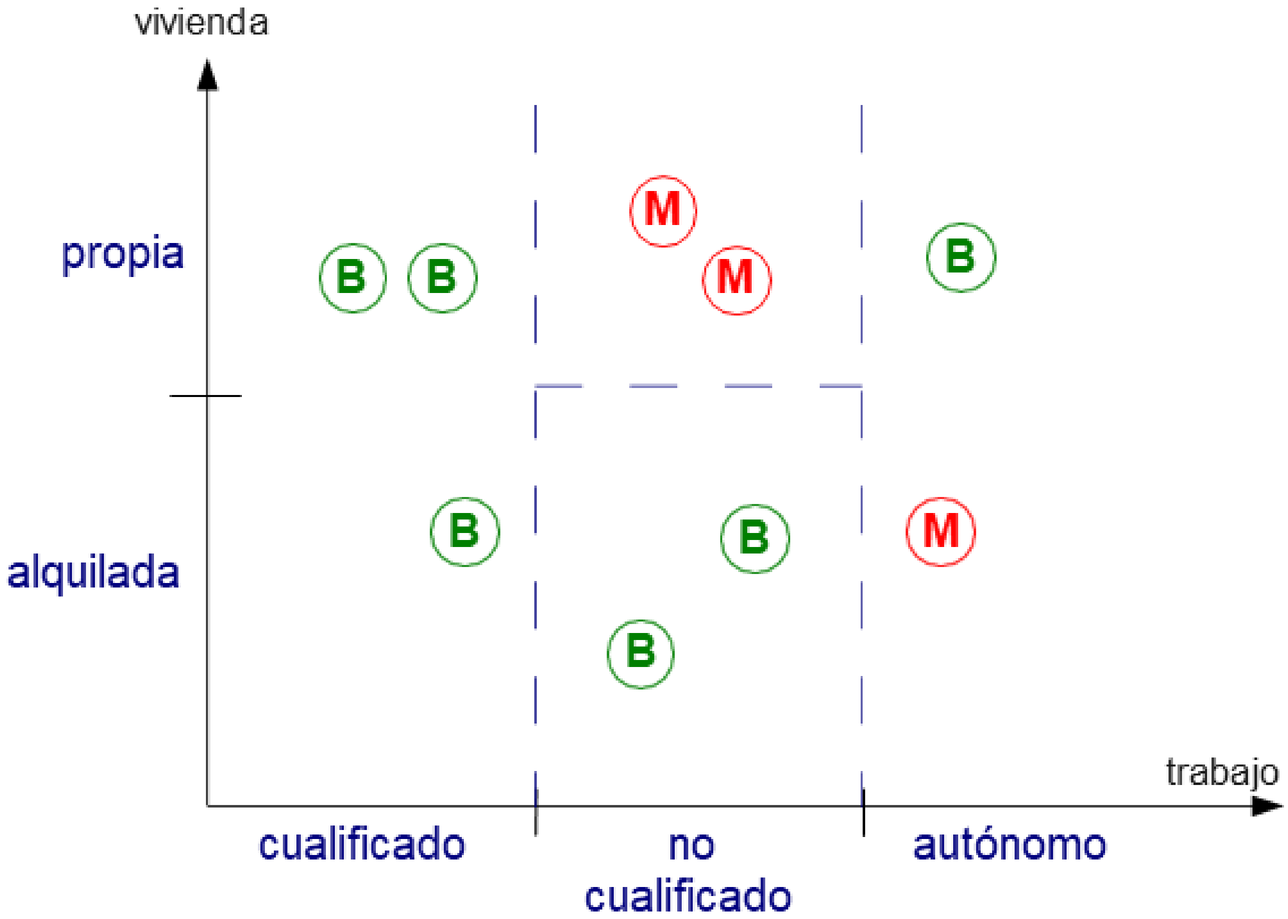
Estrategia Árbol de Decision (ineficiente)



Estrategia Árbol de Decision



Estrategia Árbol de Decision



¿Cómo funciona?

- Construcción inductiva: Se añade en cada paso un atributo con todas sus posibles respuestas
- Se selecciona el atributo que mejor separe (ordene) los ejemplos de acuerdo a las clases
- Criterios de separación
  - Entropía
  - Impureza de Gini
  - Ganancia de información
  - Ratio de ganancia de información
  - Precisión
- El proceso se detiene cuando añadir un nuevo atributo no mejora el criterio de separación
- Relevancia de atributos: La construcción de arboles de decisión realiza una selección de atributos implícita

Es decir:

- Un árbol de decisión es un **cuestionario jerárquico** que **divide** los datos de acuerdo con una **secuencia de tests en** sus **atributos**.
- **Cada ejemplo**, cuando es procesado por el árbol, sigue una **ruta** única desde el **nodo raíz** hasta la **hoja** correspondiente de acuerdo con los resultados de las **tests de los atributos** realizados en cada uno de los **nodos internos intermedios**.
- La **clase asociada a un nodo** corresponde a la **etiqueta mayoritaria de las instancias de entrenamiento** asignadas a ese nodo.
- Los **tests** en los nodos internos se determinan **maximizando una cantidad** (por ejemplo, la ganancia de información) que favorece una **separación más clara de las clases** en los hijos de dichos nodos.
- La **predicción de etiqueta** de clase para un ejemplo tiene lugar en el **nodo hoja** correspondiente.

Algoritmo de aprendizaje

```
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
   inputs: examples, set of examples
           attributes, set of attributes
           default, default value for the goal predicate

   if examples is empty then return default
   else if all examples have the same classification then return the classification
   else if attributes is empty then return MAJORITY-VALUE(examples)
   else
       best ← CHOOSE-ATTRIBUTE(attributes, examples)
       tree ← a new decision tree with root test best
       for each value v_i of best do
           examples_i ← {elements of examples with best = v_i}
           subtree ← DECISION-TREE-LEARNING(examples_i, attributes − best,
                                            MAJORITY-VALUE(examples))
           add a branch to tree with label v_i and subtree subtree
       end
       return tree
```

Versión simplificada de ID3 de Quinlan (1986)

El **mejor atributo** es el que proporciona la cantidad de información más grande sobre la etiqueta de la clase.

Recursión

ID3

- El árbol se construye de arriba a abajo, trabajando por niveles
- En cada iteración del algoritmo se pretende:
  - Obtener el atributo en base al cual ramificar el nodo problema
  - Seleccionar el que mejor discrimine entre el conjunto de ejemplos
    - Heurística para obtener árboles pequeños (en profundidad)
    - El atributo más discriminante será aquél que conduzca a un estado con menor entropía o menor desorden (mayor información)
- La entropía (Shannon, 1948) mide la **ausencia de homogeneidad** de un conjunto de ejemplos con respecto a su clase
  - Es una medida estándar del desorden (0 es homogeneidad total)
- Ganancia de información es la diferencia entre
  - la entropía del conjunto original y la de los subconjuntos obtenidos

ID3

- Para cada atributo se calcula la disminución de entropía causada por su utilización
  - Disminución de entropía$_A$(X)= E(X) – E_A(X)
  - Disminución de entropía$_B$(X)= E(X) – E_B(X)
  - Disminución de entropía$_C$(X)= E(X) – E_C(X) ...

- En cada nodo, se selecciona aquel atributo que mayor disminución de entropía provoca

- Esta medida, tiene tendencia a favorecer la elección de
  - atributos con muchos valores posibles,
  - lo que redunda en una peor generalización

Información de medición: entropía binaria

“v.a” significa “variable aleatoria”

“c.p.” significa “con probabilidad”

Considera la v.a binaria $X = \begin{cases} x_1 & c.p.\ p \\ x_0 & c.p.\ q = 1 - p \end{cases}$

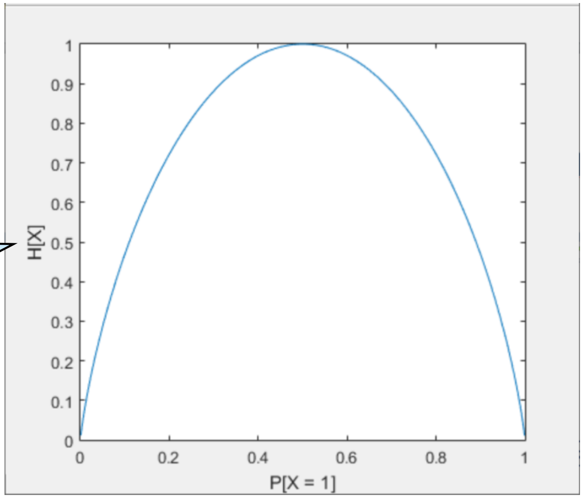
Entropía De v.a. X

$$H(X) = H_b(p) = -p\ \log_2 p - q \log_2 q \text{ [en bits]}$$

$p = P[X = x_1]$
$q = P[X = x_0]$

$0 \leq p, q \leq 1$
$p + q = 1$

$0 \leq H(X) \leq 1$
La información no Puede ser negativa



Una unidad común de medir información

Otra unidadmenos común de medir información

También: $H(X) = -p \log p - q \log q$ **[en nats]**

Logaritmo natural

Ejemplo.Codificación de mensajes: tasa de entropía

$x_1 =$ 'a'
$x_0 =$ 'b'

- **Contenidos de información de mensajes con {'a','b'} aleatorios:**
  - **Mensaje 1: "aaaaaaaaaaaaaaaaaaaa"**

$$\hat{p} = 1; \hat{q} = 0$$

Estima las probabilidades de las frecuencias de los símbolos en el mensaje

$H_b(p)$ es mínimo para p $= 1, q = 0$
p $= 0, q = 1$

$$H(X) = 0\ bits$$

Enviar mensaje: "20 a's"
Cuando la longitud del mensaje real es muy grande, la cantidad promedio de información por símbolo que necesita para ser transmitido se acerca a 0 bits

  - **Mensaje 2: "aaababaaabaaabaaaaab"**

$$\hat{p} = \frac{3}{4}; \hat{q} = \frac{1}{4}$$

$$H(X) = 0.81\ bits$$

  - **Mensaje 3: "abbababbaababbaaabba"**

$H_b(p)$ es máximo para p $= q = \frac{1}{2}$

$$\hat{p} = \frac{1}{2}; \hat{q} = \frac{1}{2}$$

$$H(X) = 1\ bit$$

Todos los símbolos deben transmitirse: En promedio 1 bit por símbolo

Entropía para una v.a. discreta

- **Considera la v.a. discreta**

$\mathcal{X}$ es el espacio en el que la variable aleatoria toma valores

$$X \in \mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$$

La variable aleatoria X puede tomar $|\mathcal{X}|$ valores diferentes

Máximo: $H(X) = \log_2|\mathcal{X}|$
para $P[X = x_i] = \frac{1}{|\mathcal{X}|},\ i = 1,2, \dots, |\mathcal{X}|$

$$H(X) = -\sum_{i=1}^{|\mathcal{X}|} P[X = x_i] \log_2 P[X = x_i] \qquad \text{[en bits]}$$

Si el emisor desea transmitir valores (muestreados independientemente) de la v.a. X a un receptor, la entropía mide la cantidad promedio de bits por símbolo del mensaje de longitud mínima

Ejemplo. Codificación de mensajes: tasa de entropía

$x_1 = $'a'
$x_2 = $'b'
$x_3 = $'c'
$x_4 = $'d'

- **Información en los mensajes con {'a','b','c','d'}:**
  - **Mensaje 1:** $p('a') = p('b') = p('c') = p('d') = 1/2$

$H(X) = 2\ bits$

Símbolo	'a'	'b'	'c'	'd'
Codificación	00	01	10	11

Código de longitud fija

  - **Mensaje 2:** $p('a') = \frac{1}{2}; \quad p('b') = p('c') = \frac{1}{4}; \quad p('d') = 0$

$H(X) = 1.5\ bits$

Símbolo	'a'	'b'	'c'	'd'
Codificación	0	10	11	-

Longitud de la variable del código prefijo

- Promedio de # bits por símbolo de mensaje codificado:

$$p('a') \times 1\ bit + p('b') \times 2\ bit + p('c') \times 2\ bits + p('d') \times 0\ bits = 1.5\ bits$$

Una alternativa: la impureza de Gini

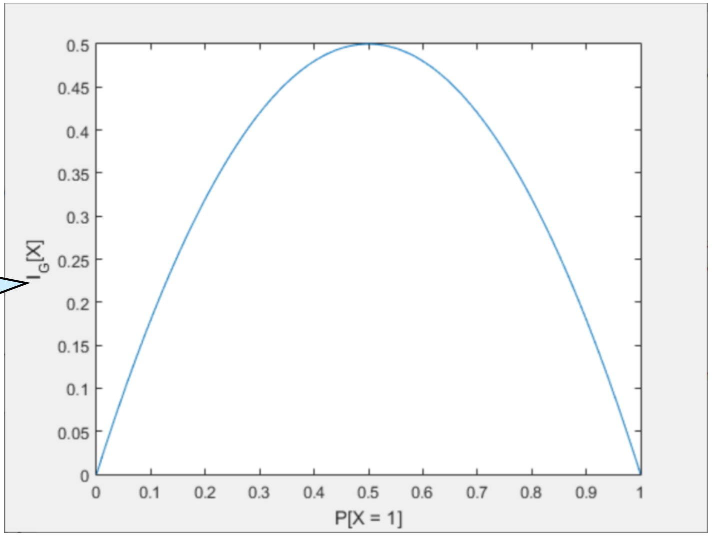
- **Considera la v.a. discreta** $X \in \mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$

$$I_G(X) = 1 - \sum_{i=1}^{|\mathcal{X}|} (P[X = x_i])^2$$

Se usa para determinar divisiones en árboles de decisión CART (Breiman et al. 1984)

- **Impureza de Gini para una v.a. binaria** $I_G(X) = 1 - p^2 - q^2$

Forma similar a la entropía binaria



Entropía condicional

- **Considera las v.'s discretas:** $X \in \mathcal{X} = \{x_1, x_2, \dots, x_{|\mathcal{X}|}\}$

  $Y \in \mathcal{Y} = \{y_1, y_2, \dots, y_{|\mathcal{Y}|}\}$

- Entropía de v.a. Y condicionada en $X = x_i$

$$H(Y \mid X = x_i) = \sum_{j=1}^{|\mathcal{Y}|} P[Y = y_j | X = x_i] \log_2 P[Y = y_j | X = x_i]$$

- **Entropía condicional :**

Promedio sobre los posibles valores de X

$$H(Y \mid X) = -\sum_{i=1}^{|\mathcal{Y}|} P[X = x_i] H(Y \mid X = x_i)$$

$H(Y|X) \leq H(Y)$

$H(Y|X) = H(Y)$
si X e Y
Son independientes

Si un emisor quiere transmitir valores de Y, la entropía condicional
mide el número promedio de bits por símbolo del
mensaje mínimo, suponiendo que el receptor conozca el valor de X

Ganancia de información

$H(Y) \geq H(Y|X)$
Entonces, $IG(Y \mid X) \geq 0$
$IG(Y \mid X) = 0$ si X e Y son independientes

$$IG(Y \mid X) = H(Y) - H(Y|X)$$ **[en bits]**

Mide el número promedio de bits por símbolo del mensaje mínimo para transmitir valores de la variable aleatoria Y que se guarda suponiendo que el receptor conoce el valor de X

- Selecciona el **mejor atributo** como el que **maximiza la ganancia de información** de la clase dada por ese atributo.

Split en la raíz del árbol

¿Cómo determino la primera pregunta en el árbol de decisión?

- Variable de clase: $WillWait \in \{yes, no\}$

  Numero de instancias: $N = 12$ $(N_{yes} = 6; N_{no} = 6)$

$$H(WillWait) = H_b\left(\frac{6}{12}\right) = 1 \text{ bit}$$

- El mejor atributo para hacer una división en la raíz del árbol es el que maximiza la ganancia de información.
  - $IG(WillWait|Type) = 0$ bits
  - $IG(WillWait|Patrons) = 0.64$ bits
  - …

No obtiene información

Mayor valor de la ganancia de información. Mejor atributo: *Patrons*

Comprueba los otros valores!

GI de *WillWait* de*Patrons*?

- Atributo: $Patrons \in \{none, some, full\}$
  - $N_{none} = 2 \;\; (N_{yes,none} = 0; N_{no,none} = 2)$

    $$H(WillWait|none) = H_b\left(\frac{0}{2}\right) = 0 \text{ bits}$$
  - $N_{some} = 4 \; (N_{yes,some} = 4; N_{no,some} = 0)$

    $$H(WillWait|some) = H_b\left(\frac{4}{4}\right) = 0 \text{ bits}$$
  - $N_{full} = 6 \;\; (N_{yes,full} = 2; N_{no,full} = 4)$

    $$H(WillWait|full) = H_b\left(\frac{2}{6}\right) = 0.92 \text{ bits}$$

$$H(WillWait|Patrons) = \frac{2}{12}0 + \frac{2}{12}0 + \frac{6}{12}0.92 = 0.46 \text{ bits}$$

$$IG(WillWait|Patrons) = 1 - 0.46 = 0.64 \text{ bits}$$

Recursión: Split en el nodo *Patrons* = *full*

¿Cómo determino
la siguiente pregunta en
el árbol de decisión?

- Instancias de entrenamiento en el nodo

Patrons = *full*:

$$\{X_2, X_4, X_5, X_9, X_{10}, X_{12}\}$$

$$N = 6\ (N_{yes} = 2; N_{no} = 4) \Rightarrow$$

$$H(WillWait) = H_b\left(\frac{2}{6}\right) = 0.92 \text{ bits}$$

Comprueba esto!

- El mejor atributo para hacer una división en este nodo es $Hungry$

$$H(WillWait|Hungry) = \frac{4}{6}H_b\left(\frac{2}{4}\right) + \frac{2}{6}H_b\left(\frac{0}{2}\right) = 0.67 \text{ bits}$$

$$IG(WillWait|Hungry) = 0.92 - 0.67 = 0.25 \text{ bits}$$

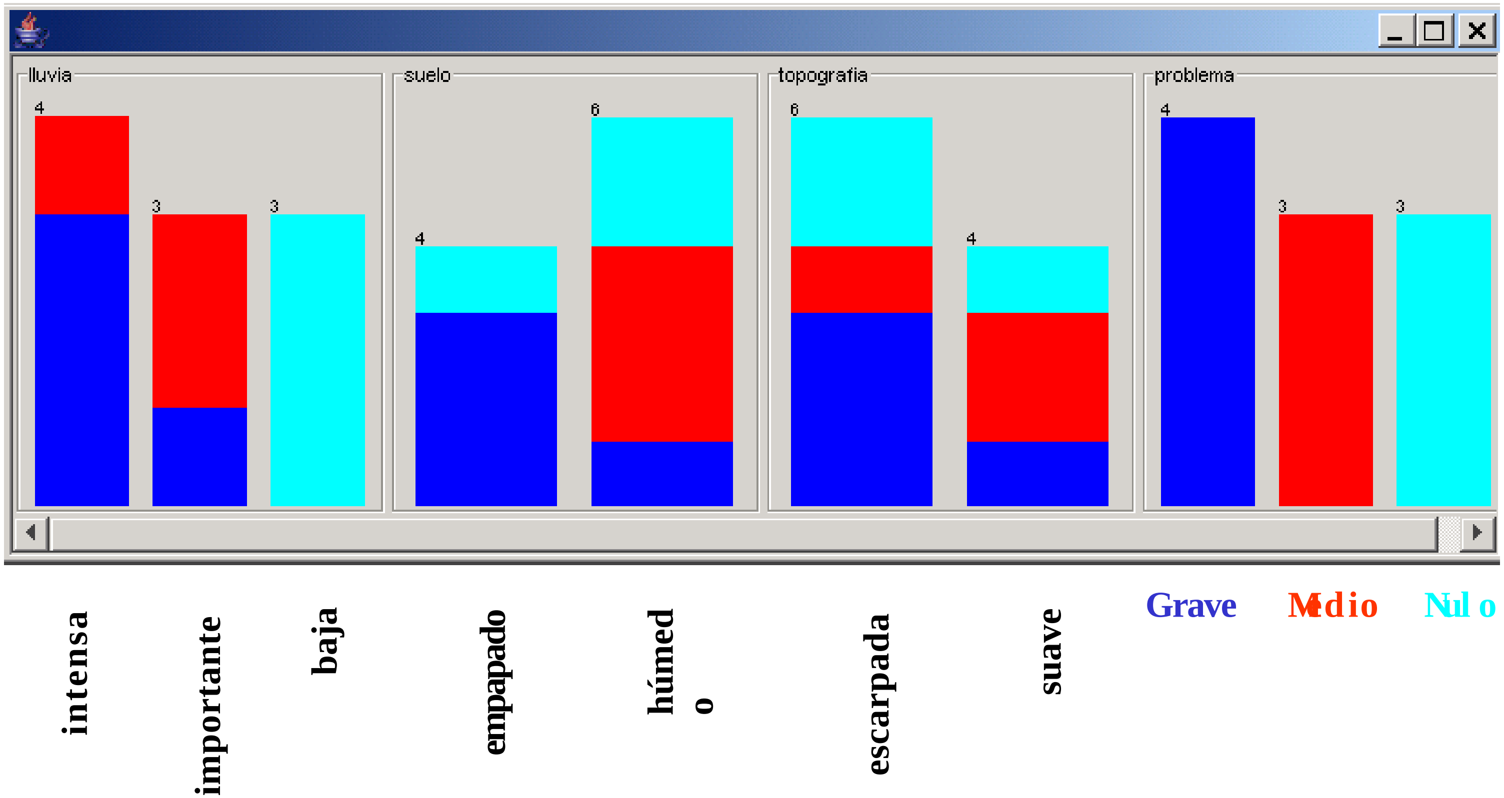
¿Cuándo se deja de dividir un nodo?

- Los ejemplos de entrenamiento asignados a ese nodo pertenecen a la misma clase. [El nodo hoja asigna esa etiqueta de clase]
- El nodo no tiene ejemplos asociados. [El nodo hoja asigna la etiqueta de clase predeterminada]
- No quedan más atributos para dividir los datos. [El nodo hoja asigna la etiqueta de clase mayoritaria en ese nodo]
- Poda previa (con límite = el tamaño del árbol para evitar **sobreajuste**)
  - El número de ejemplos de entrenamiento asociados al nodo está por debajo de un umbral.
  - La Ganancia de Información está por debajo de un umbral.

  [El nodo hoja asigna la etiqueta de clase mayoritaria en ese nodo]

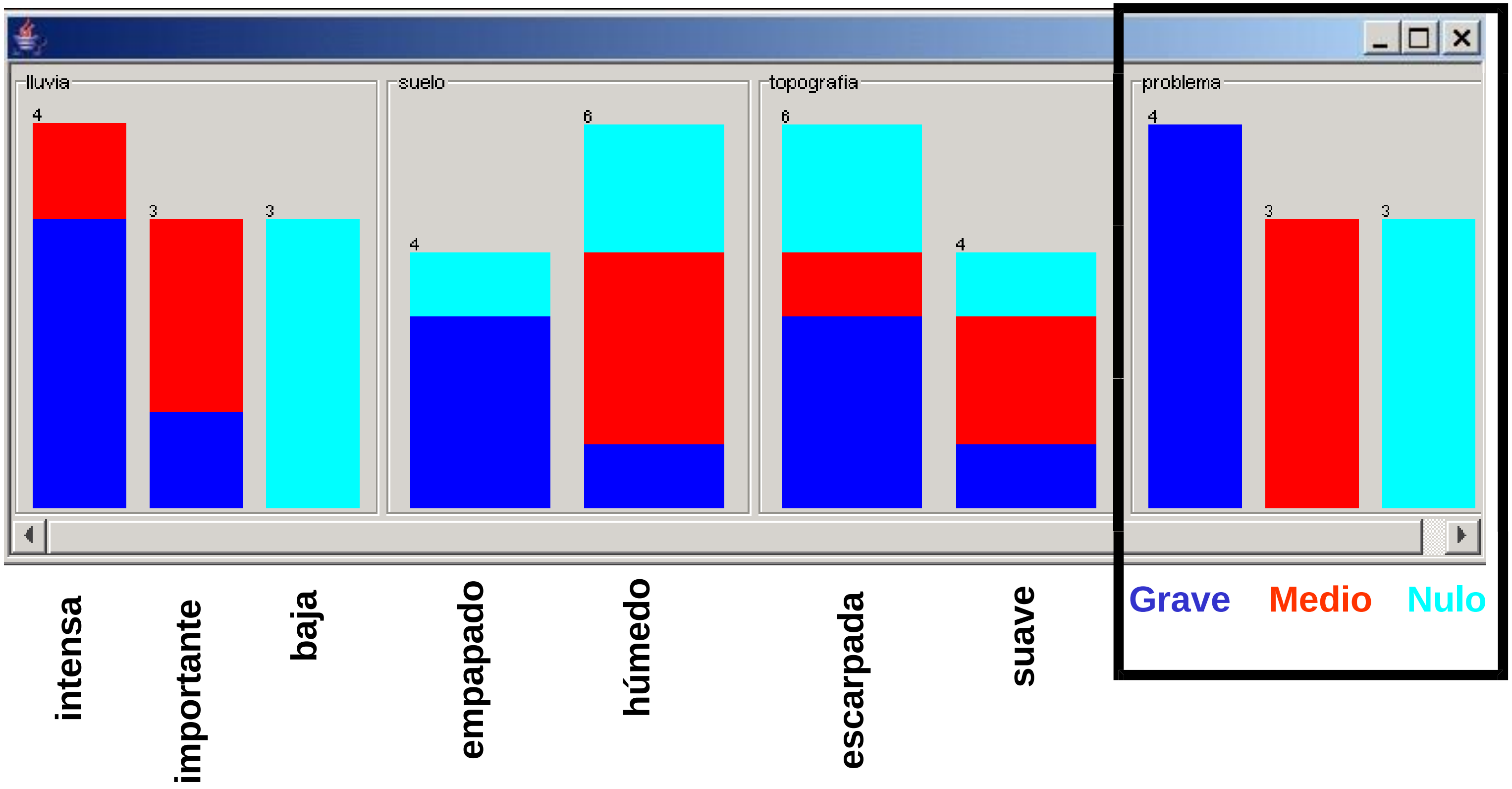
Ej. Umbral= I_G de una división aleatoria

ID3: ejemplo


lluvia
4
3
3
suelo
4
6
topografia
6
4
problema
4
3
3
intensa
importante
baja
empapado
húmedo
escarpada
suave
Grave
Medio
Nulo

ID3: ejemplo

- Entropía inicial en la raíz del árbol: (del problema global)
  - P(grave) = 0,4   P(medio) = 0,3   P(nulo) = 0,3
  - E(raíz) = -0,4 log20,4 – 0,3 log20,3 – 0,3 log20,3 =



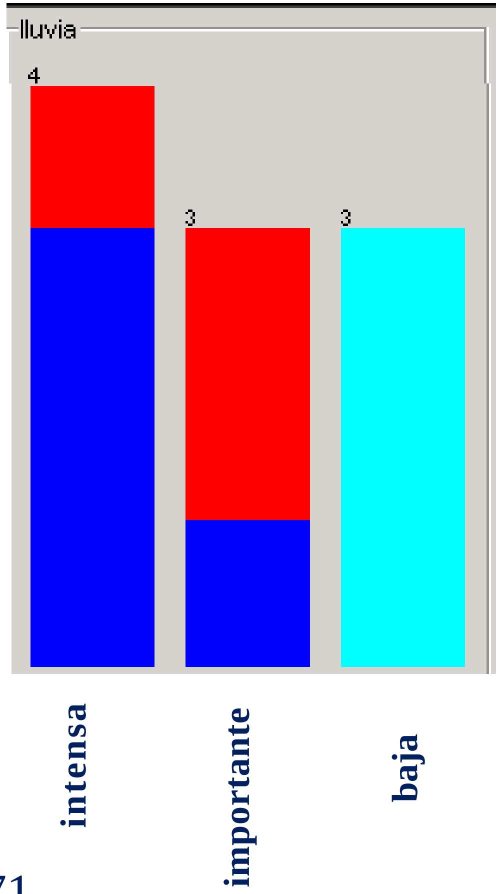
ID3: ejemplo

- Entropía final clasificando según lluvia (A):

  A_1: lluvia intensa,

  A_2: lluvia importante,

  A_3: lluvia baja



$E(A_1) = -0{,}75 \log_2 0{,}75 – 0{,}25 \log_2 0{,}25 = 0{,}811$

$E(A_2) = 0{,}918$

$E(A_3) = 0$

entropí a

$E_A(\text{raíz}) = 0{,}4 * 0{,}811 + 0{,}3 * 0{,}918 = 0{,}6$

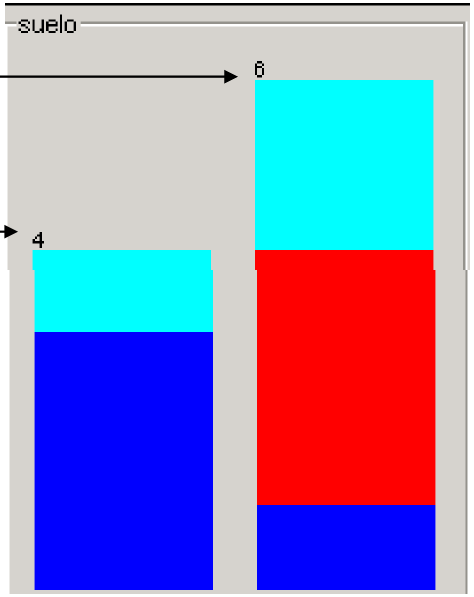
Probabilidad de "intensa"

Disminución de entropía$_A$(raíz) = 1,571 – 0,60 = 0,971

ID3: ejemplo

- Entropía final clasificando según suelo(B):

  E_B(raíz) = 0,4*0,811 + 0,6*1,459 = 1,20



$-0{,}75 \log_2 0{,}75 - 0{,}25 \log_2 0{,}25$

$-1/6 \log_2 1/6 - 3/6 \log_2 1/2 - 2/6 \log_2 1/3$

Disminución de entropía$_B$(raíz) = 1,571 – 1,20 = 0,371

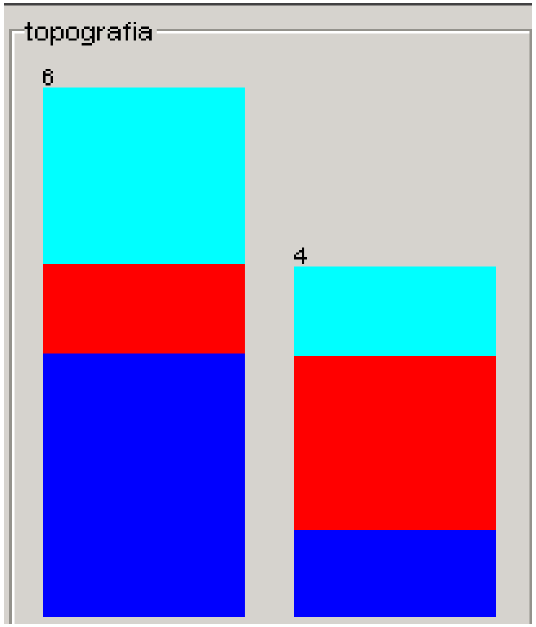
ID3: ejemplo

- Entropía final clasificando según topografía (C):

$-1/4 \log_2 1/4 \;-\; 2/4 \log_2 2/4 \;-\; 1/4 \log_2 1/4$

$E_C(\text{raíz}) = 0{,}6 * 1{,}459 + 0{,}4 * 1{,}50 = 1{,}475$

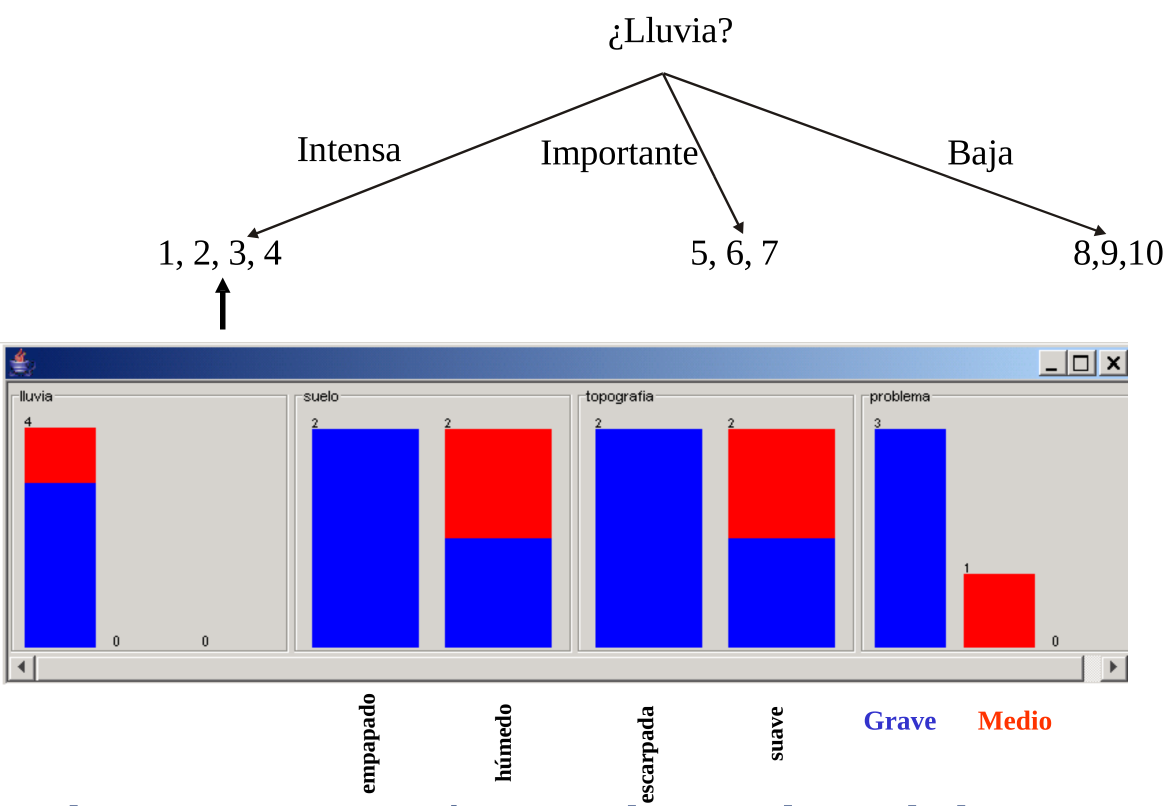
$-1/6 \log_2 1/6 - 3/6 \log_2 1/2 - 2/6 \log_2 1/3$



Disminución de entropía$_C$(raíz) = 1,571 – 1,475 = 0,096

- La mayor disminución de entropía se consigue con el atributo A y por ello éste es el seleccionado para el primer nivel del árbol

ID3: ejemplo



- En la siguiente iteración se vuelve a aplicar el algoritmo sobre cada uno de los tres nuevos nodos, considerando en cada uno el subconjunto de ejemplos obtenido y habiendo eliminado el atributo lluvia del conjunto de atributos

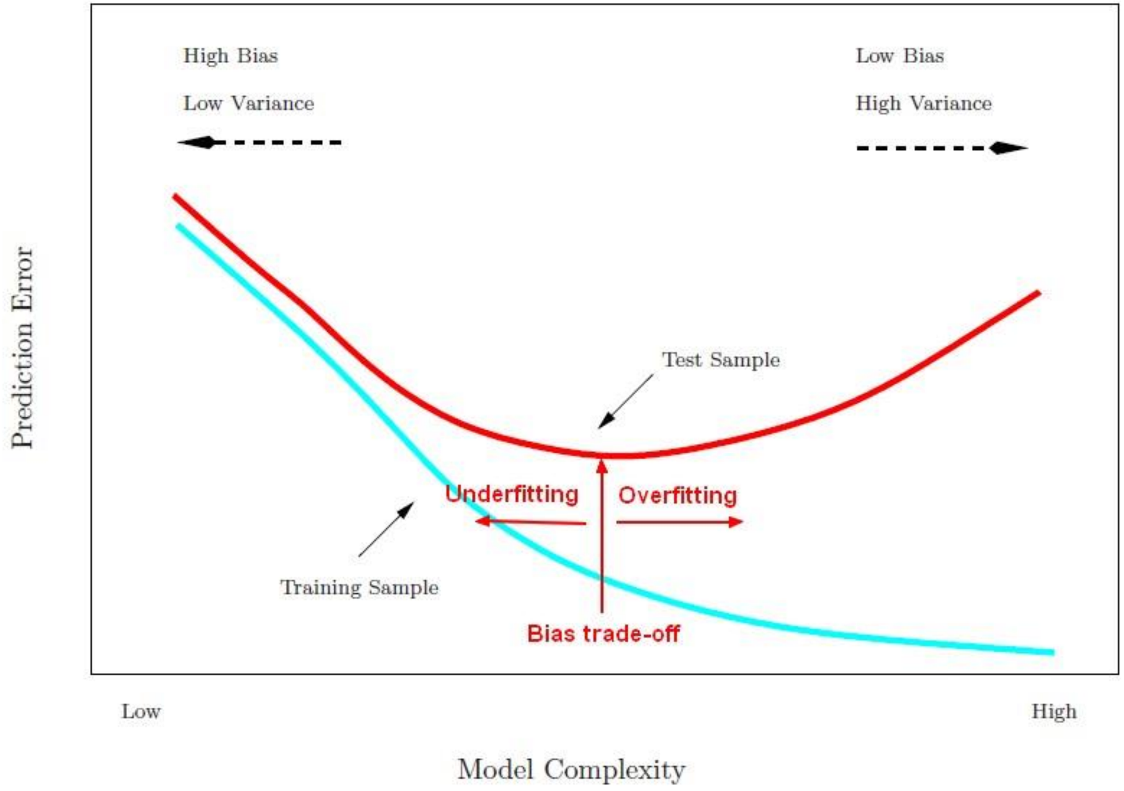
Underfitting / overfitting

- **Infraajuste (Underfitting)**

  El tipo de predictor considerado tiene **baja capacidad expresiva**. En consecuencia, no puede capturar las dependencias entre los atributos y la variable a predecir. El error del predictor es demasiado alto.

- **Sobreajuste (Overfitting)**
  - Los modelos se refinan tanto que describen bien las instancias de entrenamiento pero obtienen un error alto en ejemplos externos. El tipo de predictor considerado es **demasiado flexible** y aprende patrones espurios que no son relevantes para la predicción. La estimación es demasiado optimista y subestima el error real.
  - Causas:
    - pocos ejemplos
    - las instancias de entrenamiento no son una muestra representativa
    - ruido en las instancias de entrenamiento
    - error en los datos

Underfitting / overfitting



Source: https://gerardnico.com/ under
license CC Attribution-Noncommercial-Share Alike 4.0 International

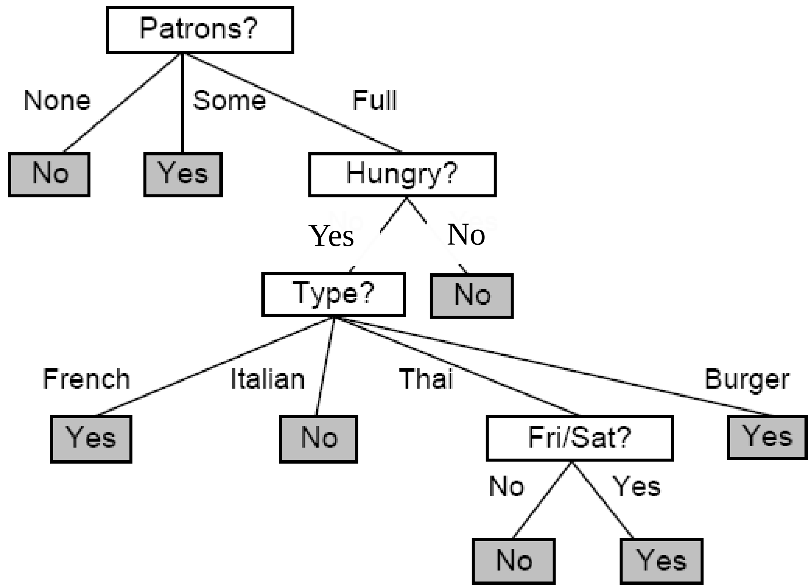
Poda para evitar el sobreajuste en arboles de decisión

- Sesgo hacia árboles más pequeños (menos complejos).
  - Poda previa
  - Post-poda: Genera el árbol hasta un tamaño grande y luego poda los sub-árboles que no brinden ganancias significativas en la precisión predictiva.
    - Considera un nodo interno.
    - Si convertir ese nodo en una hoja no conduce a una disminución significativa en la precisión predictiva del árbol de decisión podado, elimine el subárbol que tiene ese nodo como raíz.
    - Para este proceso, la precisión se puede estimar en un conjunto de validación separado (poda de error reducida) o por V.C. (ej., como en CART)
    - Continúe podando hasta un deterioro significativo de la precisión.

La poda posterior generalmente se prefiere. Esta es una estrategia común en el aprendizaje automático: considera primero un modelo potencialmente complejo y luego penaliza la complejidad

Interpretabilidad: extracción de reglas

Todos los árboles de decisión pueden traducirse a reglas de decisión



Sistema de reglas:

El **grupo se queda si**

bien el restaurante tiene **algún patrón**

o (el restaurante está **lleno y** el **grupo está hambriento y**

(el tipo de comida es **Francesa o** (**Thai y** es **Viernes/Sábado**) **o Burger**)

Sino, el **grupo se va**.

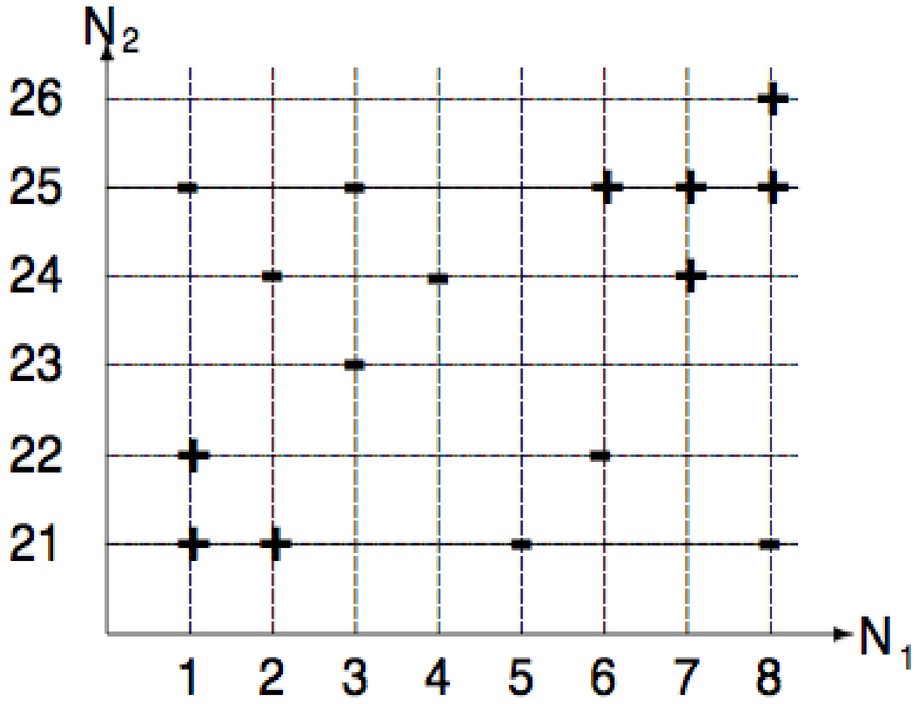
El modelo es interpretable!

C4.5 árbol de decisón

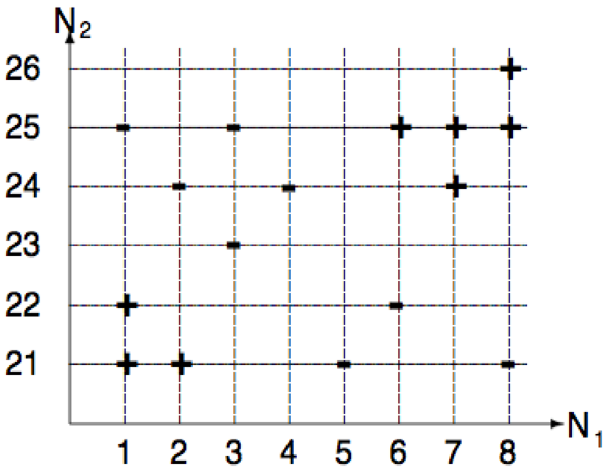
- Evolución de ID3 por Quinlan (1992)
- Incluye
  - Pruebas basadas en atributos numéricos.
  - Decisiones confusas
  - Poda posterior
  - Normalización de la ganancia de información para atributos multivalor.
  - Manejo de valores perdidos
  - Regla extracción y poda
- Soluciona un pequeño problema de ID3: tiene una cierta tendencia a favorecer la elección de atributos con muchos valores posibles, lo que redunda en una peor generalización de las observaciones
  - Usa el **Ratio de Ganancia** (de información) para cada atributo

C4.5: Manejo de atributos numéricos.

- Atributos: N_1, N_2
- Clase: "+", "-" $H(Class) = H_b\left(\frac{8}{16}\right) = 1$ bit

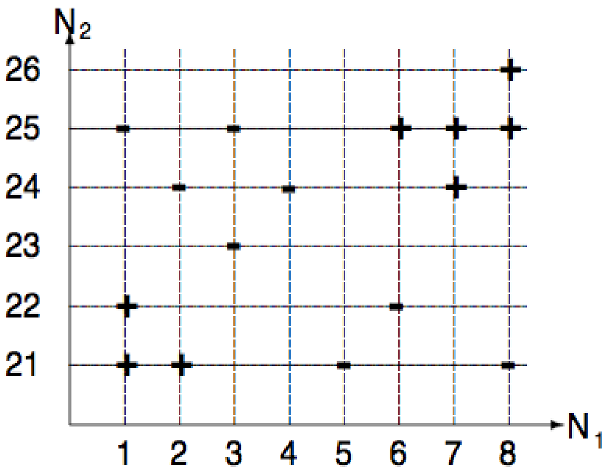


Tests en N_1



Pregunta	Rama "No"	Rama "Sí"	Entropía clase en Rama "No"	Entropía clase en Rama "Sí"	H(clase | Pre-gunta)	IG
N_1>1	2+, 1-	6+, 7-	H(2/3, 1/3) = 0.918 bits	H(6/13, 7/13) = 0.996 bits	3/16*0.918+ 13/16*0.996 = 0.981 bits	1-0.981= 0.019 bits
N_1>2	3+, 2-	5+, 6-	H(3/5, 2/5) = 0.971 bits	H(5/11, 6/11) = 0.994 bits	5/16*0.971+ 11/16*0.994 = 0.987 bits	1-0.987= 0.013 bits
N_1>3	3+, 4-	5+, 4-	H(3/7, 4/7) = 0.985 bits	H(5/9, 4/9) = 0.991 bits	7/16*0.985+ 9/16*0.991 = 0.988 bits	1-0.988= 0.012 bits
N_1>4	3+, 5-	5+, 3-	H(3/8, 5/8) = 0.954 bits	H(5/8, 3/8) = 0.954 bits	8/16*0.954+ 8/16*0.954 = 0.954 bits	1-0.954= 0.046 bits
N_1>5	3+, 6-	5+, 2-	H(3/9, 6/9) = 0.918 bits	H(5/7, 2/7) = 0.863 bits	9/16*0.918+ 7/16*0.863 = 0.894 bits	1-0.894= 0.106 bits
N_1>6	4+, 7-	4+, 1-	H(4/11, 7/11)= 0.946 bits	H(4/5, 1/5) = 0.722 bits	11/16*0.946+ 5/16*0.722 = 0.876 bits	1-0.876= 0.124 bits
N_1>7	6+, 7-	2+, 1-	H(6/13, 7/13)= 0.996 bits	H(2/3, 1/3) = 0.918 bits	13/16*0.996+ 3/16*0.918 = 0.981 bits	1-0.981= 0.019 bits
N_1>8	8+, 8-	0+, 0-	1 bit	--	16/16*1+ 0/16*-- = 0 bits	1-1 = 0 bits

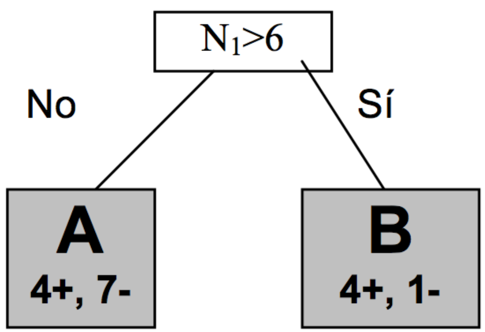
Tests en N_2

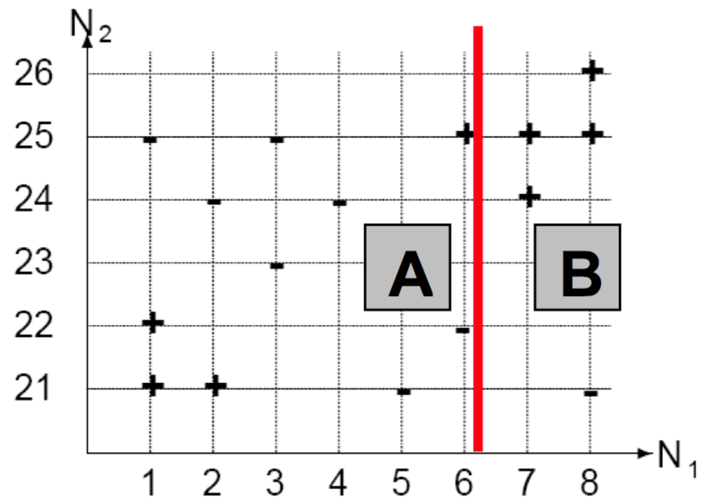


Pregunta	Rama "No"	Rama "Sí"	Entropía clase en Rama "No"	Entropía clase en Rama "Sí"	H(clase | Pre-gunta)	IG
N_2>21	2+, 2-	6+, 6-	H(2/4, 2/4) = 1 bits	H(6/12, 6/12) = 1 bits	4/16*1+ 12/16*1 = 1 bits	1-1= 0 bits
N_2>22	3+, 3-	5+, 5-	H(3/6, 3/6) = 1 bits	H(5/10, 5/10) = 1 bits	6/16*1+ 10/16*1 = 1 bits	1-1= 0 bits
N_2>23	3+, 4-	5+, 4-	H(3/7, 4/7) = 0.985 bits	H(5/9, 4/9) = 0.991 bits	7/16*0.985+ 9/16*0.991 = 0.988 bits	1-0.988= 0.012 bits
N_2>24	4+, 6-	4+, 2-	H(4/10, 6/10)= 0.971 bits	H(4/6, 2/6) = 0.918 bits	10/16*0.971+ 6/16*0.918 = 0.951 bits	1-0.951= 0.049 bits
N_2>25	7+, 8-	1+, 0-	H(7/15, 8/15)= 0.997 bits	H(1/1, 0/1) = 0 bits	15/16*0.997+ 1/16*0 = 0.935 bits	1-0.894= 0.[illegible]5 bits

Más pequeños que la GI del test ($N_1 > 6?$)

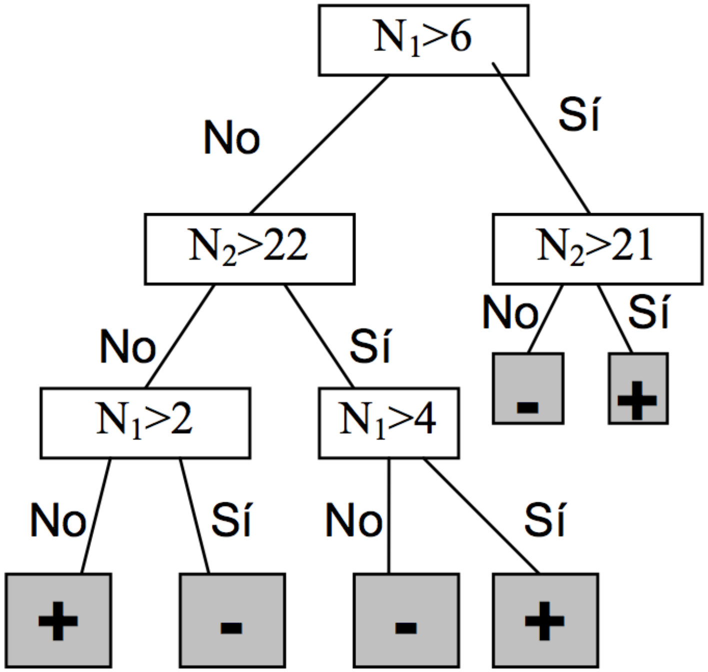
Test en el nodo ráiz

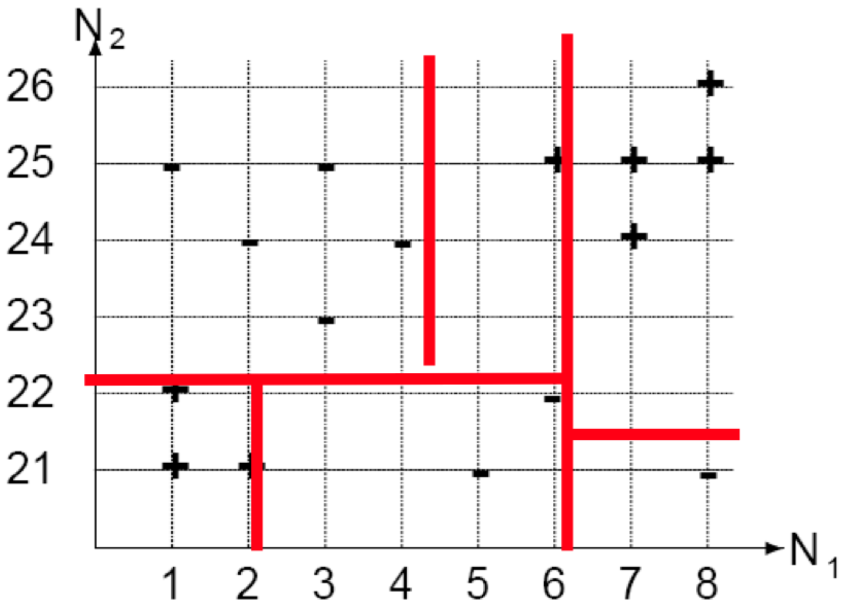




- El espacio de atributo original se ha dividido en 2 subespacios disjuntos (A y B)
- Usando una estrategia de "**divide y vencerás**", y recursivamente particiona A y B por separado.

C4.5 árbol de decision final





Árboles de decisión: pros & cons

- **Ventajas**
  - Implementación simple.
  - Resultados interpretables.
  - Entrenamiento rápido y predicción.
- **Inconvenientes**
  - Predicciones no muy precisas. Sin embargo, pueden usarse como aprendices básicos para un conjunto.

https://scikit-learn.org/stable/modules/tree.html

Bosques de decision (decision forests): Conjuntos de Árboles de decisión

- **Aleatorización**
  - **Embolsado (Bagging)**
  - **Random forest**
- **Aleatorización + optimización**
  - **Refuerzo (Boosting)**
  - **Gradient boosting**
  - **Xgboost (Extreme Gradient Boosting)**

  [https://xgboost.readthedocs.io/en/latest/]

Los conjuntos de embolsado y refuerzo también se pueden componer de otros tipos de aprendizaje básicos, como las redes neuronales

Random forest, gradient boosting, Y xgboost tienen un excelente rendimiento listo- para-usar en problemas no estructurados

[https://scikit-learn.org/stable/modules/ensemble.html]