

Ejemplo

Problema: decidir si se debe de esperar por una mesa en un restaurante utilizando los siguientes atributos:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range (\$, \$\$, \$\$\$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

Ejemplo

- Los ejemplos se describen mediante los valores de los atributos (Booleanos, discretos, continuos)

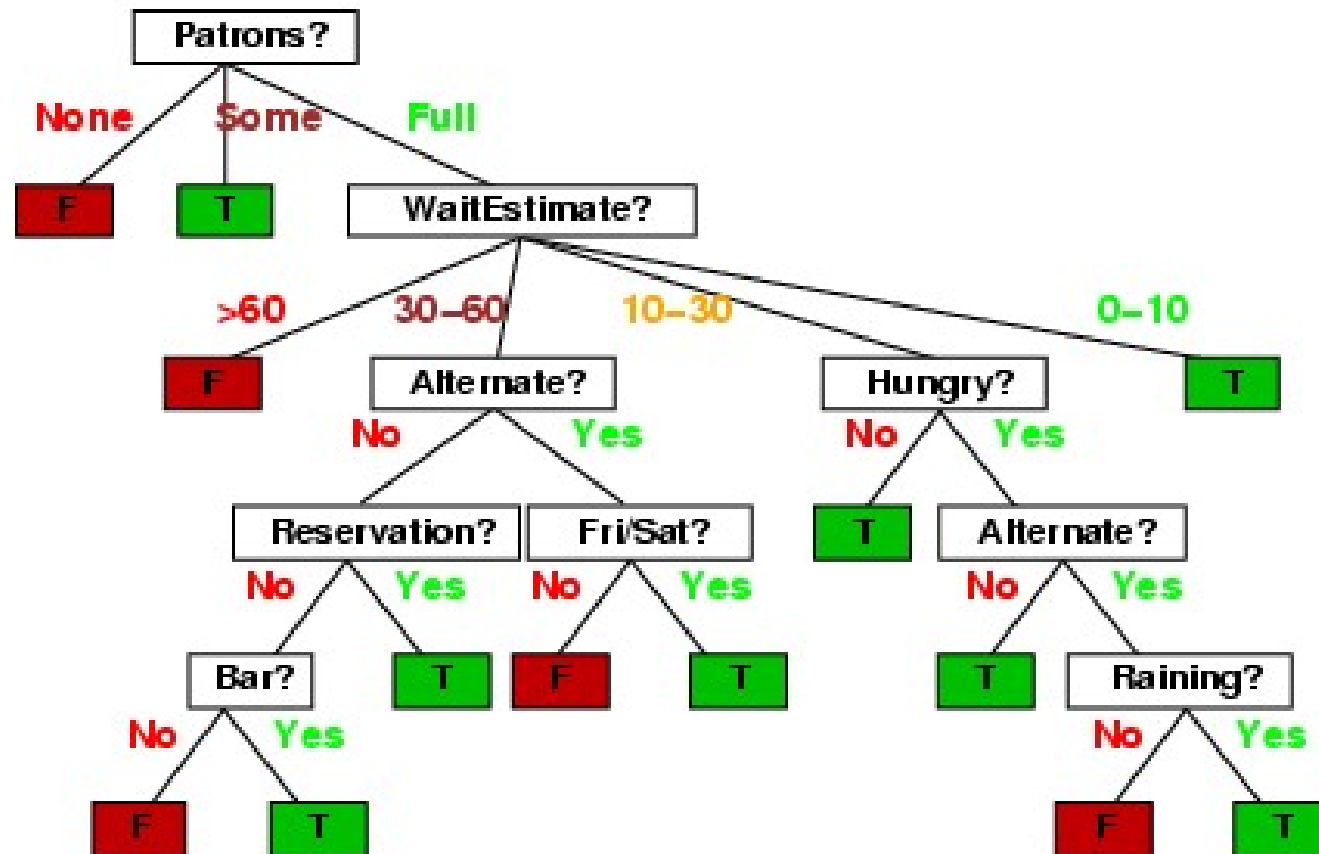
Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>Wait</i>
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

- La clasificación de los ejemplos es positiva (T) o negativa (F).

Ejemplos positivos y negativos

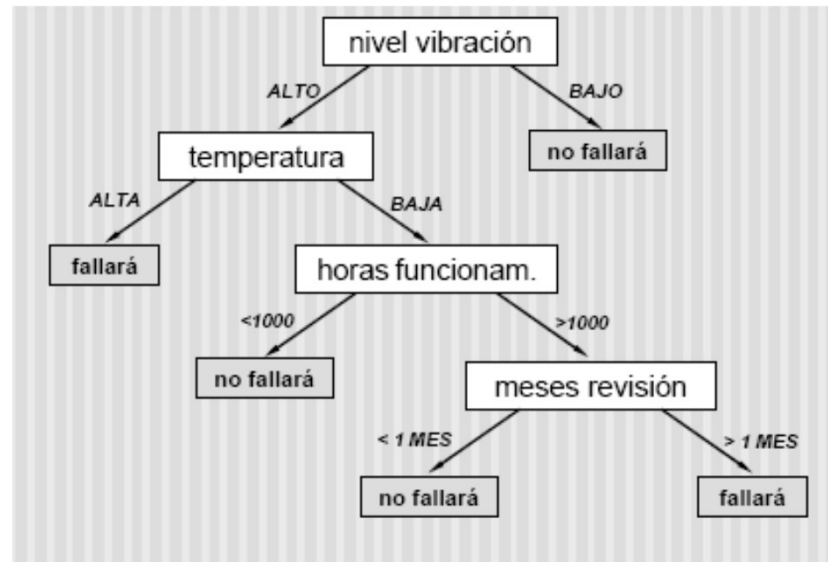
- Los ejemplos positivos son aquellos en los que la meta *esperar* es verdadera (X_1, X_3, \dots).
- Los ejemplos negativos son aquellos en los que es falsa (X_2, X_5, \dots).
- El conjunto de ejemplos completo se denomina **conjunto de entrenamiento**.

Una posible representación para la hipótesis



Ejemplo

Modelado de la probabilidad de fallo de una máquina



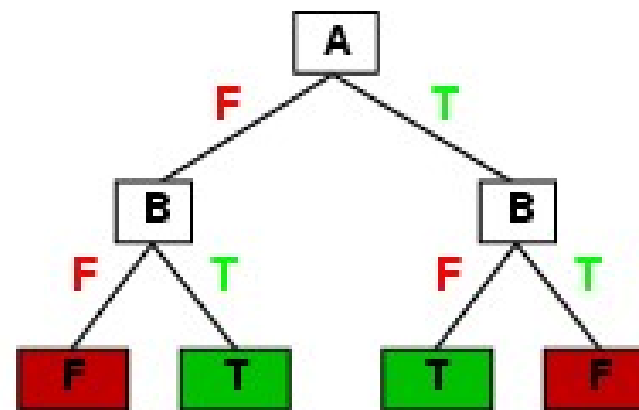
Ejemplo

Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará

Expresividad de los árboles de decisión

- Los árboles de decisión pueden expresar cualquier función a partir de los atributos de entrada.
- Por ejemplo, para funciones Booleanas, cada fila de la tabla de verdad se traslada a un camino del árbol:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



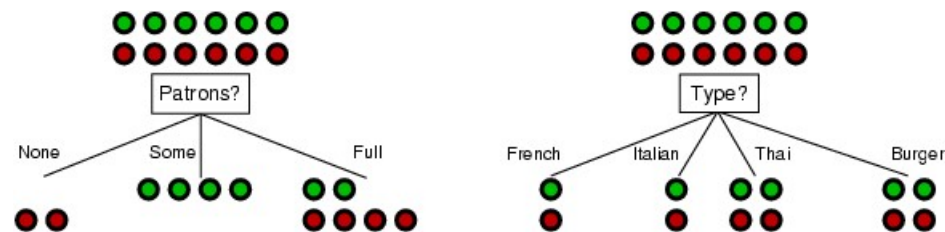
- De forma trivial, hay un árbol de decisión consistente para cualquier conjunto de entrenamiento con un camino asociado a cada ejemplo, pero seguramente no será bueno para generalizar nuevos ejemplos.
- Es preferible encontrar árboles de decisión más compactos.

Inducción de árboles de decisión

- Múltiples formas de inferir el árbol:
 - Trivial: se crea una ruta del árbol por cada instancia de entrenamiento.
 - Árboles excesivamente grandes.
 - No funcionan bien con instancias nuevas.
 - Optimo: el árbol más pequeño posible compatible con todas las instancias (navaja de Ockham).
 - Inviabile computacionalmente.
 - Pseudo-optimo (heurístico): selección del atributo en cada nivel del árbol en función de la calidad de la división que produce.
 - Los principales programas de generación de árboles utilizan procedimientos similares (ID3, C4.5, CART, etc).

Elección de atributos

Idea: un buen atributo debería dividir el conjunto de ejemplos en subconjuntos que sean o “todos positivos” o “todos negativos”.



Patrons=Clientes es una buena elección

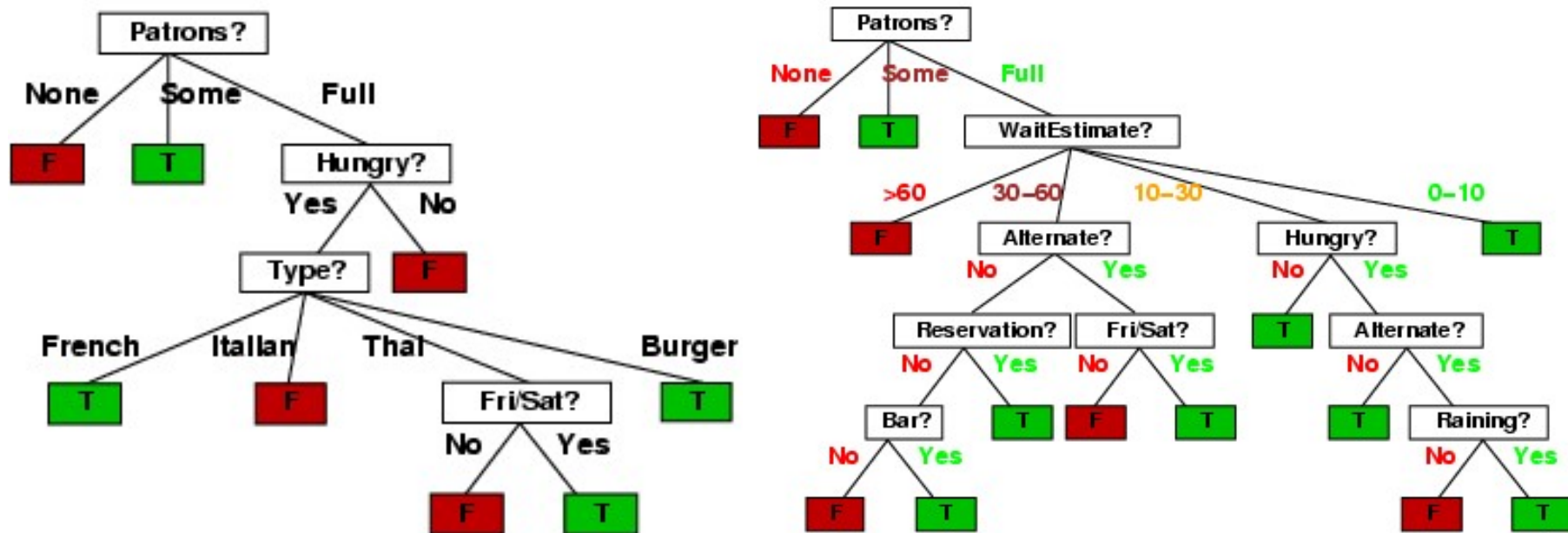
DTL

```
function DTL(examples, attributes, default) returns a decision tree
  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 MODE(examples)
  else
    best  $\leftarrow$  CHOOSE-ATTRIBUTE(attributes, examples)
    tree  $\leftarrow$  a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi  $\leftarrow$  {elements of examples with best =  $v_i$ }
      subtree  $\leftarrow$  DTL(examplesi, attributes - best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
    return tree
```

1. No quedan ejemplos: valor por defecto calculado a partir de la mayoría en el nodo padre.
2. Todos los ejemplos son positivos o negativos.
3. No quedan atributos: voto de la mayoría de los ejemplos que quedan.
4. Quedan ejemplos positivos y negativos.

Árbol de decisión obtenido

Árbol de decisión obtenido utilizando los 12 ejemplos



Es más simple que el árbol “verdadero”.

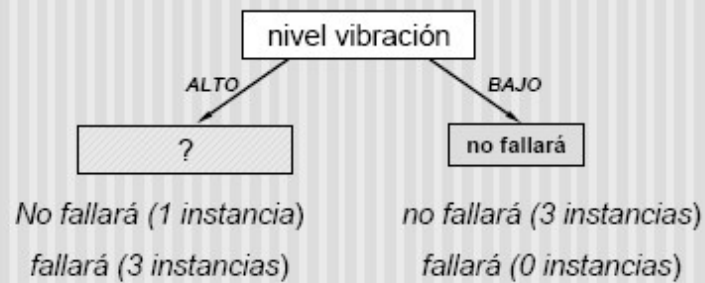
Ejemplo detallado

- ¿Qué atributo elegir para el primer nodo?

ATRIBUTO	VALORES	CLASE	
		<i>fallará</i>	<i>no fallará</i>
Temperatura	Alto	2	2
	Bajo	1	2
Nivel de vibraciones	Alto	3	1
	Bajo	0	3
Horas defuncionamiento	< 1000	2	3
	>1000	1	1
Meses desde revisión	> 1 mes	2	3
	< 1 mes	1	1

Ejemplo detallado

- Árbol construido hasta el momento:



- ¿Qué atributo se debe usar en el siguiente nivel del árbol (rama izquierda)?

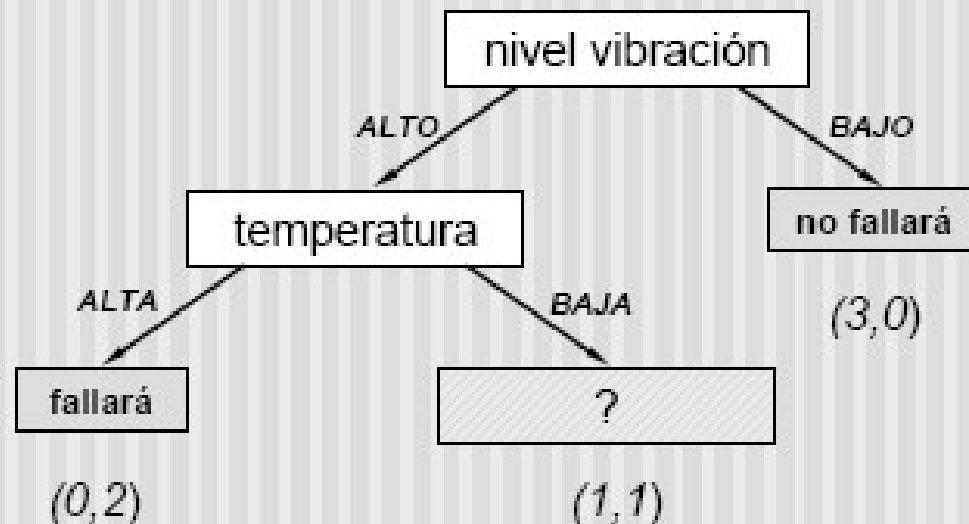
Ejemplo detallado

Sólo aquellos ejemplos de entrenamiento que llegan al nodo se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		<i>fallará</i>	<i>No fallará</i>
Temperatura	Alta	2	0
	BAja	1	1
Horas de funcionamiento	< 1000	2	1
	>1000	1	0
Meses desde revisión	> 1 mes	2	1
	< 1 mes	1	0

Ejemplo detallado

- Árbol construido hasta el momento:



- ¿Qué atributo se debe usar en el siguiente nivel del árbol (rama derecha)?

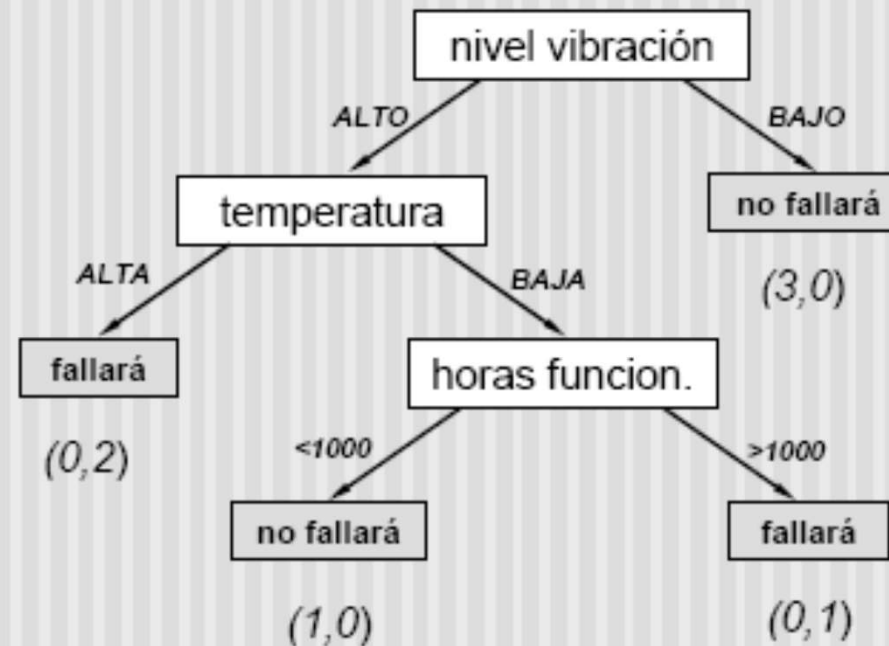
Ejemplo detallado

De nuevo, sólo aquellos ejemplos de entrenamiento que llegan al nodo se utilizan para elegir el nuevo atributo:

ATRIBUTO	VALORES	CLASE	
		<i>fails</i>	<i>works</i>
Horas de funcionamiento	< 1000	0	1
	>1000	1	0
Meses desde revisión	> 1 mes	1	1
	< 1 mes	0	0

Ejemplo detallado

■ Árbol obtenido finalmente:



... muy similar al árbol original, utilizando sólo 7 ejemplos de entrenamiento!