

# Técnicas (automatizadas) para la adquisición de conocimientos

Juan Luis Castro Peña

# Tareas a automatizar

Conceptualización del dominio → obtención de conceptos y factores relevantes

Repertory Grid (rejilla de repertorio)

Obtención de reglas -> obtención de reglas candidatas a partir de ejemplos

Aprendizaje de árboles de decisión

Aprendizaje de reglas

# Rejilla de repertorio

- También conocida como “emparrillado”, repertory grid, ....
- Desarrollada por Kelly 1955 → para ayudar a enfermos mentales
- Aplicada en muchos ámbitos:
  - Asesoramiento
  - Estudios demográficos
  - Dinámica de grupos
  - Adquisición de conocimiento

# Idea de la rejilla de repertorio

- cada persona tiene su propia visión del mundo: asocia y distingue elementos de acuerdo a sus propios criterios (constructor, traducido como constructores o construcciones según los autores)
  - Constructor= característica bipolar que se aplica de forma gradual a los elementos
- Cuando a una persona se le pide que de razones esos parar ir distinguiendo o agrupando elementos, va a indicar los criterios y su forma de usarlos.

# Proceso de rejilla de repertorio

- Dialogo inicial con el experto
- Sesión de valoración
- Análisis de resultados

# Construcctores

- Una construcción es una característica bipolar en la cual cada elemento tiene cierto grado (o escala).

Ejemplos: Pesado-Ligero, Valiente-Cobarde, o Largo-Corto.

- Estas características se pueden representar en una escala lineal con valores ponderados (relativos a la característica).

Ejemplo, si la curvatura fuera una posible característica, entonces una posible construcción sería: Recta-Curva, las esferas y los círculos tendrían ratios mayores mientras que las líneas y cubos ratios menores

# Valoración de objetos según un constructor

- Como es el E el que crea las construcciones es trabajo suyo comprender qué hace que una construcción sea válida y cómo se usa. Los ratios pueden expresarse por nombres en lugar de por números:

muy pesado	pesado	medio	ligero	muy ligero
1	2	3	4	5
simpático				poco amistoso

- La escala no debe variar en una misma construcción aunque podrá variar de una construcción a otra.
- Un elemento se clasifica (pondera) según cada construcción usando un ratio subjetivo dado por el E; esto permite clasificar o ponderar a los elementos a la vez que compararlos.

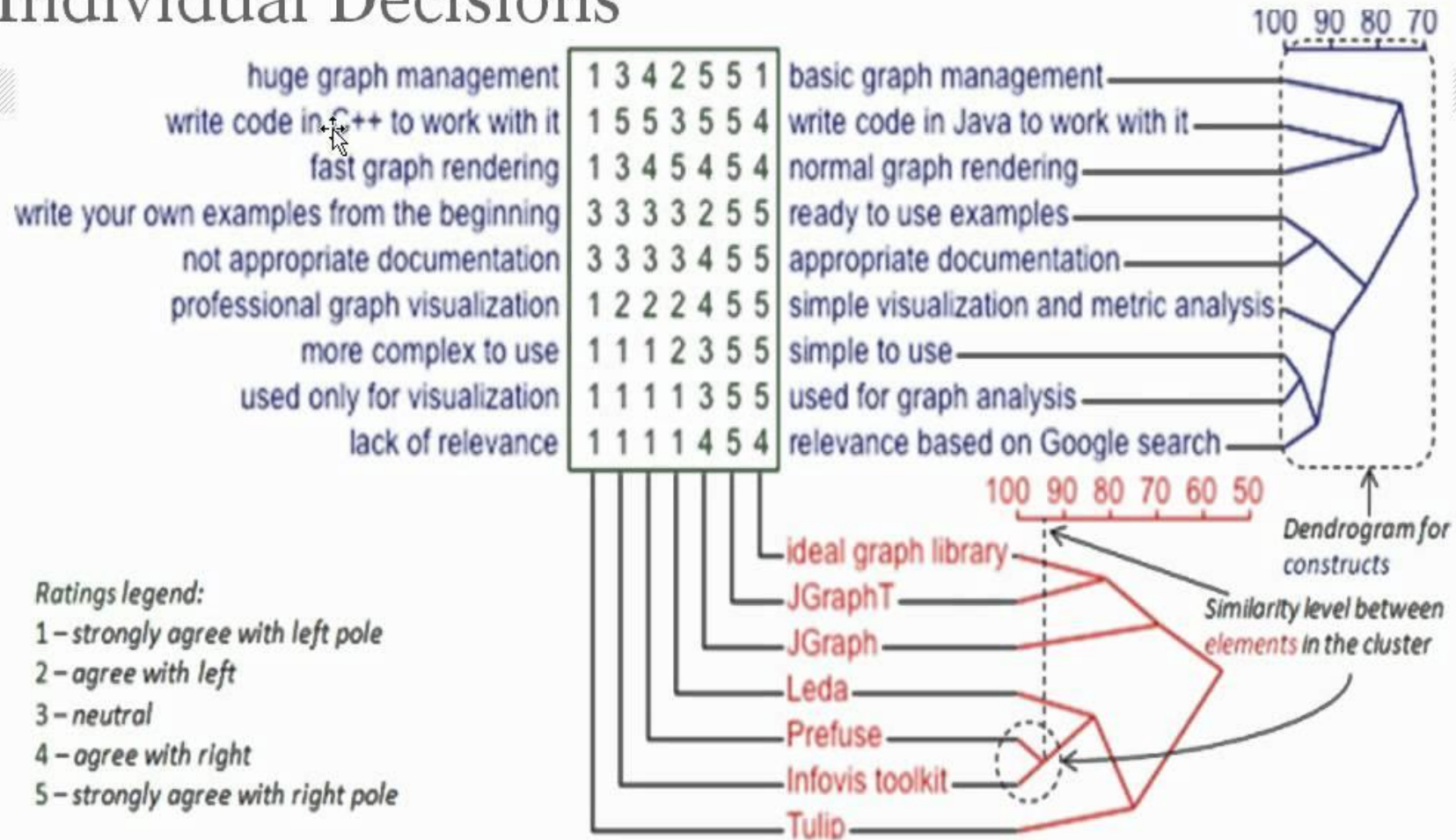
# Utilidad de la rejilla de repertorio en la IC

El grid es un sistema de correspondencia entre elementos y construcciones. Esta técnica es útil para los expertos o ingenieros del conocimiento por dos razones:

- Hace que el Experto piense sobre el problema y por tanto ayuda a clarificar consecuencias en su mente.
- Los grids se pueden analizar para encontrar modelos, clases y conceptos o asociaciones a investigar con mayor profundidad.



# Individual Decisions



# Obtención del Grid: preparación

- Elegir el problema:
  - Clasificación de objetos → Taxonomía
  - Evaluación de candidatos o alternativas,
  - Obtención de características que determinan una o varias clases de objetos
- Recoger una lista de elementos iniciales

# Obtención del grid: relleno

- Implica repetidas comparaciones de elementos. Se suelen elegir un grupo de tres elementos para empezar a describir similitudes o diferencias.
- Los grupos de tres pueden elegirse aleatoria o sistemáticamente.

- **Por ejemplo:**

El IC considera los elementos Marta, Juan, y José, y busca los más similares.

E. Marta y Juan son los más similares y diferentes de José.

IC. ¿Qué hace a Marta y Juan similares?

E. Ambos son maduros.

IC. ¿Qué es lo opuesto a maduro?, lo cual describe a José.

E. Él es inmaduro.

IC. Los polos para tu construcción son maduro e inmaduro. Ahora clasificamos cada uno de los elemento sobre una escala de 1-5 (1 maduro, 5 inmaduro).

IC: Vamos a realizar un estudio para intentar encontrar sus criterios de selección. Vamos a realizar un estudio de los candidatos en este proceso de selección. Por favor, proporcione sus elementos -candidatos- que piense que son importantes:

E: Ellos son:

- |              |            |
|--------------|------------|
| 1) Francisco | 4) José    |
| 2) Elena     | 5) Isabel  |
| 3) Manolo    | 6) Yolanda |

IC: Entre Francisco, Elena y Manolo selecciona dos que sean similares y diferentes al tercero.

E: Francisco y Manolo son similares- Elena es diferente.

IC: ¿Qué hace a Francisco y Manolo similares?

E: Ambos son responsables

IC: Responsable es un polo de nuestra construcción. El otro debe ser Irresponsable.

IC: Vamos a clasificar a los elementos según esta construcción, donde 1 significa irresponsable y 3 responsable

E: Francisco 3, Elena 1, Manolo 3, José 1, Isabel 1, Yolanda 1.

IC: Tu próximo grupo es José, Isabel, y Yolanda.

E: Isabel tiene mejores clasificaciones que José y Yolanda. Los polos serán clasificaciones pobres (1) y buenas (3). Los elementos son:

Francisco 1, Elena 2, Manolo 1, José 2, Isabel 3, y Yolanda 2

IC: El próximo grupo es Francisco, Manolo y Isabel.

E: Francisco y Isabel tienen confianza en sí mismo; Manolo no. Los polos son inseguro (1), y seguro de sí mismo (3).

Francisco 3, Elena 1, Manolo 1, José 2, Isabel 3, y Yolanda 1

IC: Ahora el grupo es Elena, José y Yolanda.

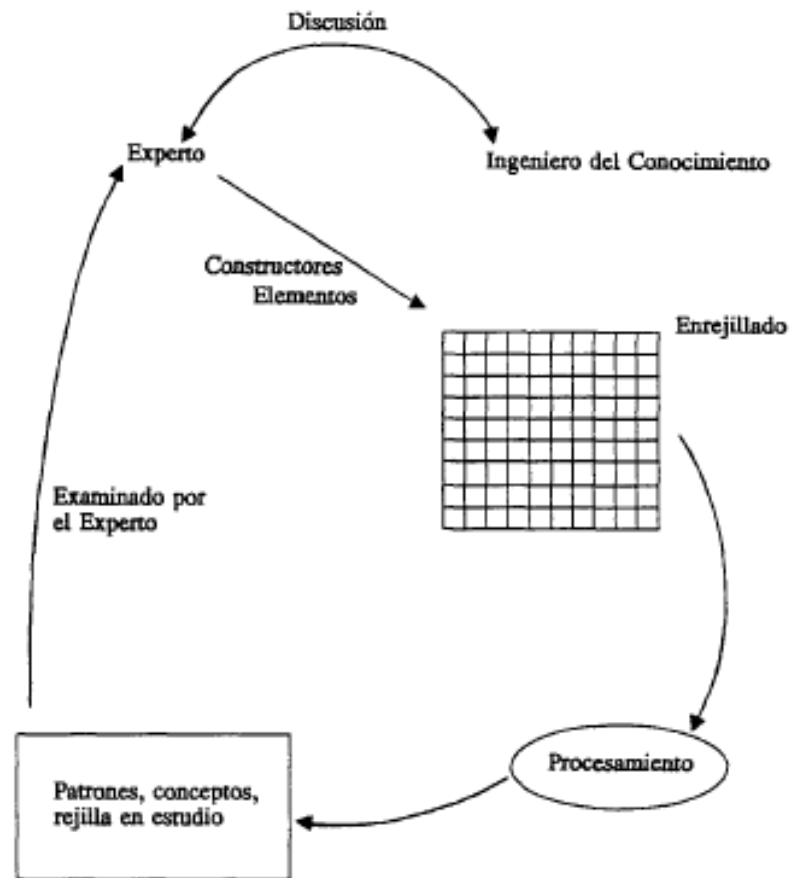
E: Elena y Yolanda no hacen preguntas en clase, José si lo hace. Mis polos serán hacen preguntas (1) y callados (3).

Francisco 1, Elena 3, Manolo 2, José 3, Isabel 1, y Yolanda 3. Pienso que voy a cambiar el ratio de José a 2.

IC: ¿Quieres seleccionar tu propio grupo?

	<i>C1. Responsable/Inresponsable</i>	<i>C2. Clasificaciones Pobres/Buenas</i>	<i>C3. Inseguro/Seguro ..</i>	<i>C4. Preguntan/No preguntan</i>	<i>C5. Atentos/Distruidos</i>	<i>C6. Trabajan/No trabajan</i>	<i>C7. Integrados/No Integrados</i>	<i>C8. Organizadores/No organizados</i>
E1. Francisco	3	1	3	1	1	1	1	3
E2. Elena	1	2	1	3	3	3	3	1
E3. Manolo	3	1	1	2	3	2	3	1
E4. José	1	2	2	2	1	1	2	2
E5. Isabel	1	3	3	1	2	1	2	3
E6. Yolanda	1	2	1	3	2	3	3	2
E7. David	1	2	3	2	3	1	1	1

# Análisis del grid



Con el objeto de ejecutar la técnica de clustering necesitamos un criterio que mida la distancia entre pares de elementos, en este ejemplo usaremos la suma del valor absoluto de las diferencias entre los ratios de los elementos (distancia mínima)<sup>2</sup>. Por ejemplo, comparando Francisco con Elena:

Francisco	3	1	3	1	1	1	1	3
Elena	1	2	1	3	3	3	3	1

La suma de diferencias es:  $2+1+2+2+2+2+2+2=15$

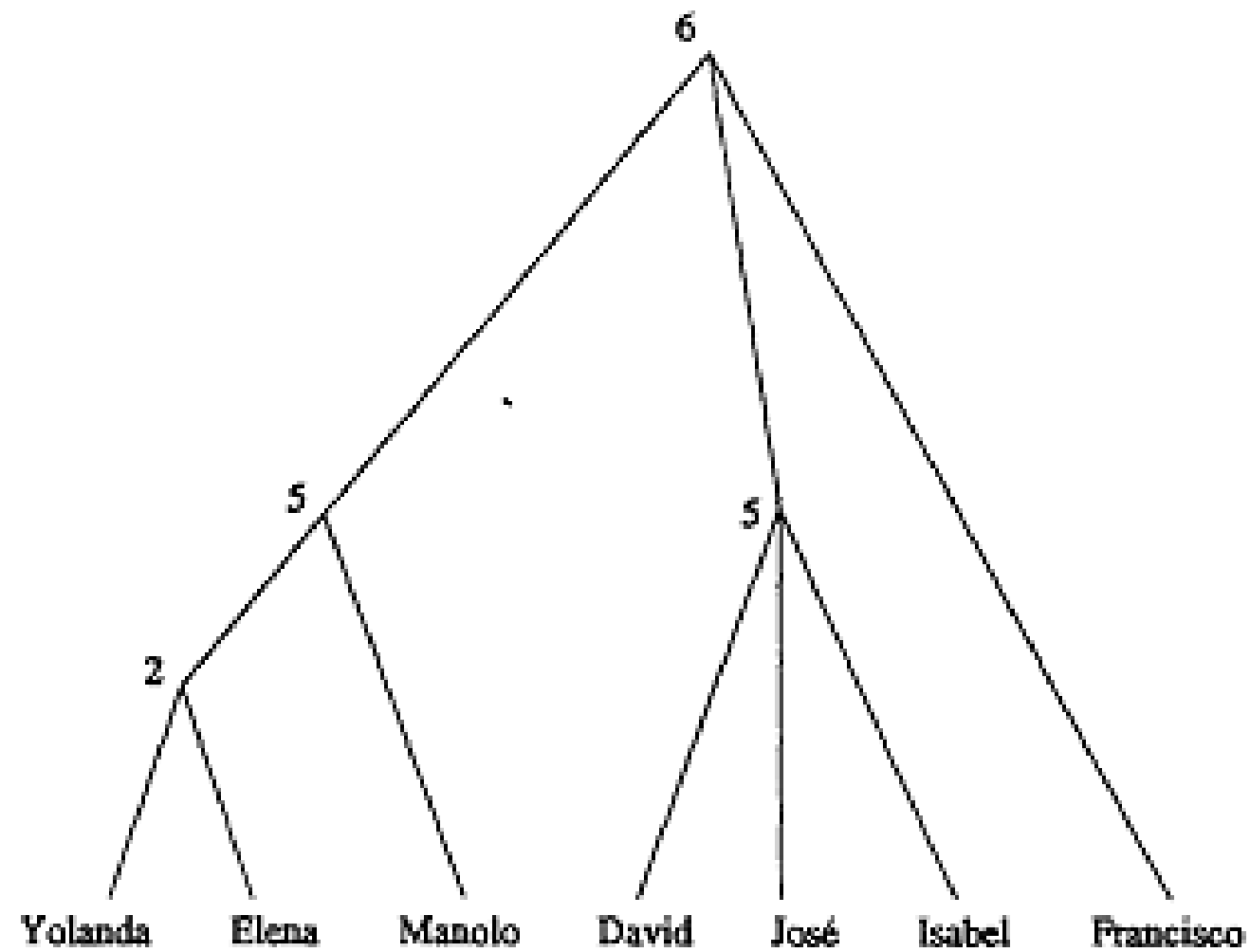


	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>E6</b>	<b>E7</b>
<b>E1</b>		15	10	7	6	13	8
<b>E2</b>			5	10	11	2	7
<b>E3</b>				9	12	7	8
<b>E4</b>					5	6	5
<b>E5</b>						9	6
<b>E6</b>							9
<b>E7</b>							

A partir de esta tabla, se construye un árbol siguiendo el siguiente procedimiento:

1. Se señalan en la parrilla, los elementos de distancia mínima. En este caso el E2 y el E6.
2. Esos elementos se reemplazan en la matriz por el conjunto que forman. (Grupo [E2, E6]).
3. La distancia de este conjunto con un elemento de la matriz se calcula como el mínimo de las distancias entre ese elemento y cada uno de los elementos del conjunto. Por ejemplo, la distancia entre el elemento E3 y el grupo [E2, E6] sería el mínimo de 5 y 7. La matriz resultante se presenta en la tabla A.3.
4. Se repite el proceso con la matriz obtenida hasta que sólo queden dos categorías en la tabla.

	<b>E1</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>	<b>E7</b>	<b>[E2-E6]</b>
<b>E1</b>		10	7	6	8	13
<b>E3</b>			9	12	8	5
<b>E4</b>				5	5	6
<b>E5</b>					6	9
<b>E7</b>						7
<b>[E2-E6]</b>						



# Análisis del cluster: construcciones

El procedimiento es el mismo que para los elementos, aunque la distancia entre dos construcciones requiere observarla en ambas direcciones. Por ejemplo si comparamos dos construcciones relacionadas con automóviles como barato/caro y deportivo/utilitario, no esperaremos una alta similaridad (correlación) entre ellas. Sin embargo, si invertimos una de ellas podría esperarse similaridad. Por ejemplo, barato/caro y utilitario/deportivo podrían ser similares.

Una construcción X' se deriva de la construcción X sustituyendo el ratio opuesto.

En el ejemplo que estamos estudiando sería el 1 por el 3:

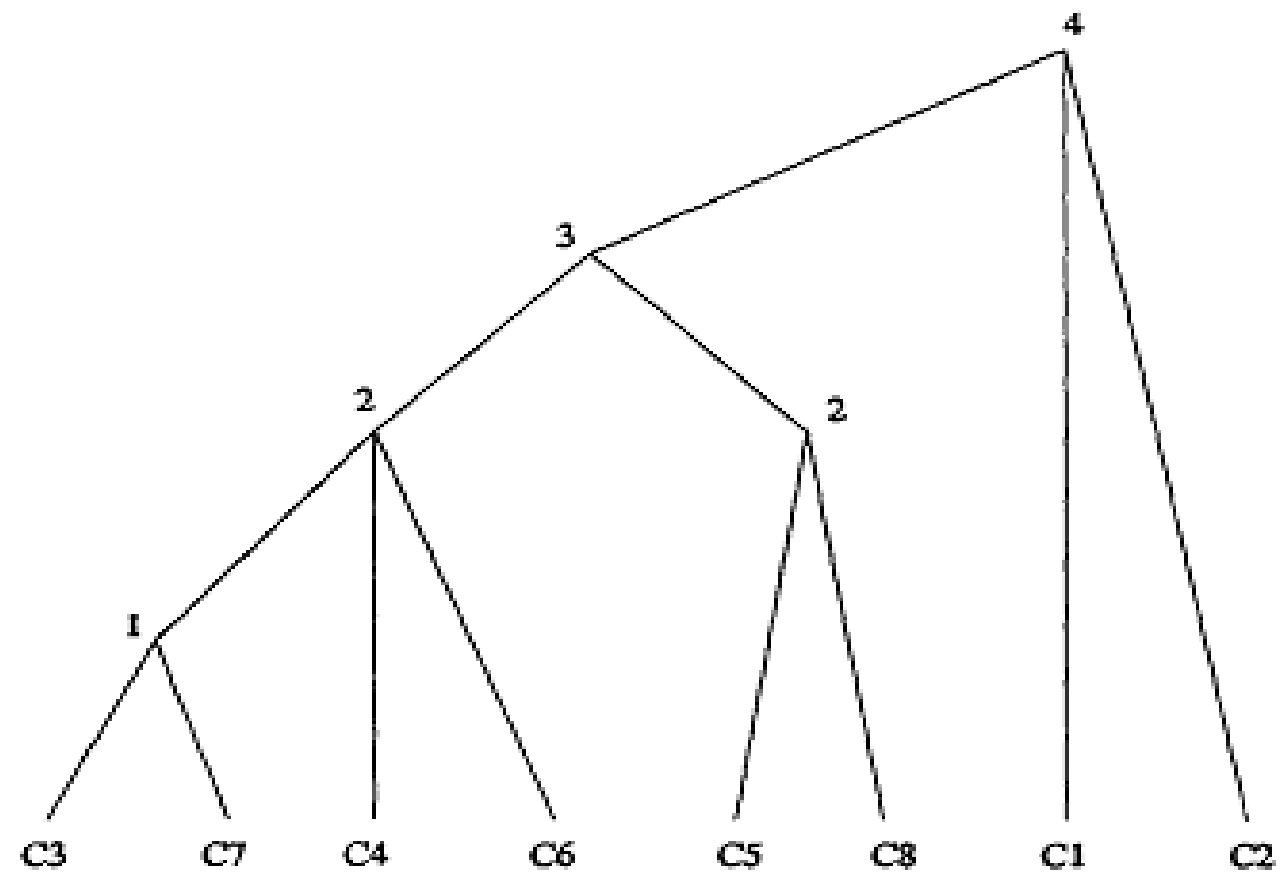
	E1	E2	E3	E4	E5	E6	E7
Atentos/Distraídos	3	1	1	3	2	2	1

	C1	C2	C3	C4	C5	C6	C7	C8
C1		10	7	9	8	7	9	6
C2	4		5	5	6	7	6	4
C3	7	7		10	9	12	11	3
C4	5	5	2		5	2	3	9
C5	6	6	5	9		5	4	10
C6	7	7	2	10	9		3	9
C7	6	6	1	9	8	11		7
C8	8	6	9	3	2	5	4	

Superior derecha en orden directo, inferior  
izquierda en orden inverso

	C1	C2	C3	C4	C5	C6	C7	C8
C1		4	7	5	6	7	6	6
C2			5	5	6	7	6	4
C3				2	5	2	1	3
C4					5	2	3	3
C5						5	4	2
C6							3	5
C7								4
C8								

Quedándose con la distancia mínima entre el  
orden directo e inverso





# Análisis de resultados

Cuando se analizan las categorizaciones realizadas, el E puede corroborar, refutar o matizar los agrupamientos realizados, pudiendo producirse los casos siguientes:

1. Dos elementos aparecen ligados cuando no deberían estarlo. En este caso se pueden reconsiderar los valores atribuidos a los objetos. Si los valores parecen correctos, el IC debe solicitar del E un motivo de diferenciación de los elementos. Dicha característica debe añadirse al estudio y generar dos nuevos árboles de agrupamientos.
2. Dos elementos aparecen como disjuntos cuando deberían estar ligados. El proceso es similar al caso anterior. Se añade una característica que ligue ambos elementos.
3. Dos características aparecen ligadas cuando no deberían estarlo. Si los valores atribuidos a los objetos para esas dos características son correctas, se le pide al E que cite un elemento que contradiga la relación. Si existe dicho elemento se añade al grid inicial y se vuelve a repetir todo el proceso.

# Árboles de decisión

- Un árbol de decisión toma como entrada un objeto o una situación descrita a través de un conjunto de atributos y devuelve una “decisión”, el valor previsto de la salida dada la entrada.
- Atributos: discreto o continuos.
- Salida:
  - Discreta: clasificación.
  - Continua: regresión.

# Árboles de decisión

## Ejemplo

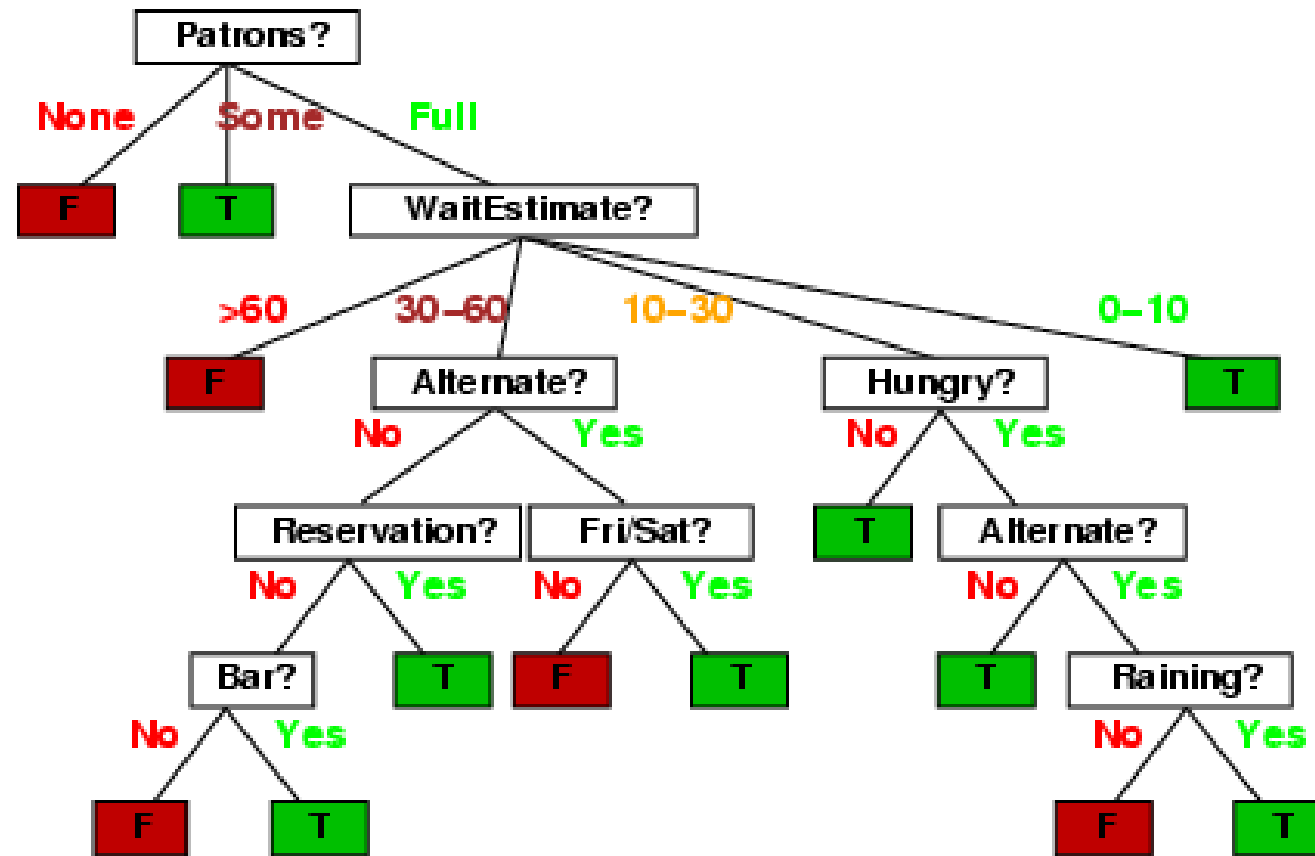
Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>Wait</i>
$X_1$	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
$X_2$	T	F	F	T	Full	\$	F	F	Thai	30–60	F
$X_3$	F	T	F	F	Some	\$	F	F	Burger	0–10	T
$X_4$	T	F	T	T	Full	\$	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

# Árboles de decisión

## Ejemplos positivos y negativos

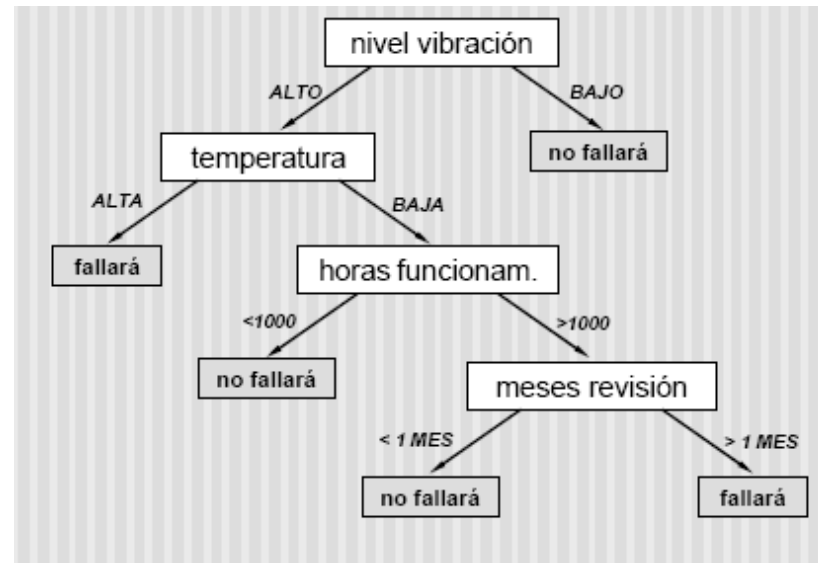
- Los ejemplos positivos son aquellos en los que la meta esperar es verdadera ( $X_1, X_3, \dots$ ).
- Los ejemplos negativos son aquellos en los que es falsa ( $X_2, X_5, \dots$ ).
- El conjunto de ejemplos completo se denomina **conjunto de entrenamiento**.

# Una posible representación para la hipótesis



# Ejemplo

Modelado de la probabilidad de fallo de una máquina



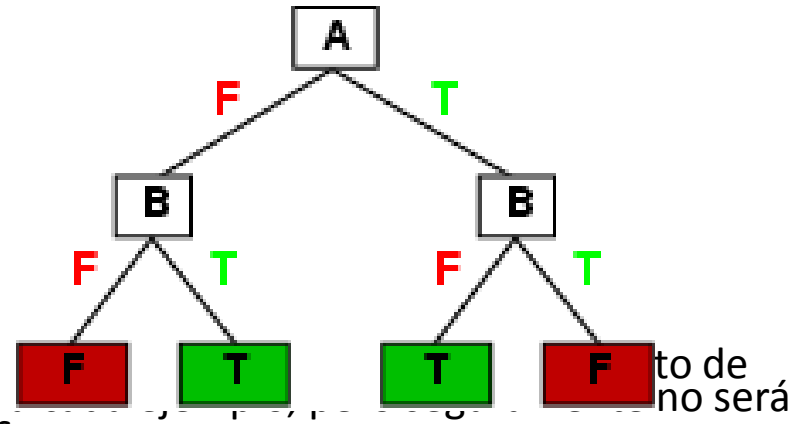
# Ejemplo

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 entrenam  
bueno para generalizar nuevos ejemplos.
- Es preferible encontrar árboles de decisión más compactos.

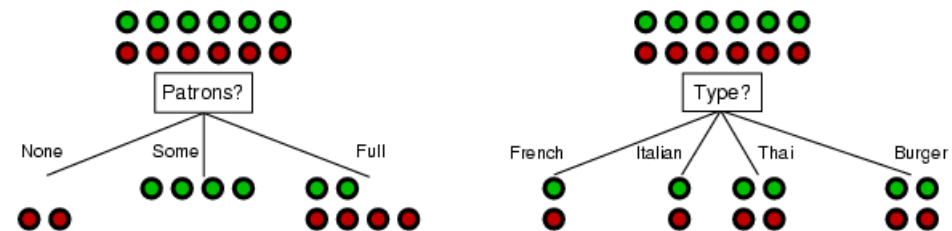


# Inducción de árboles de decisión

- 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

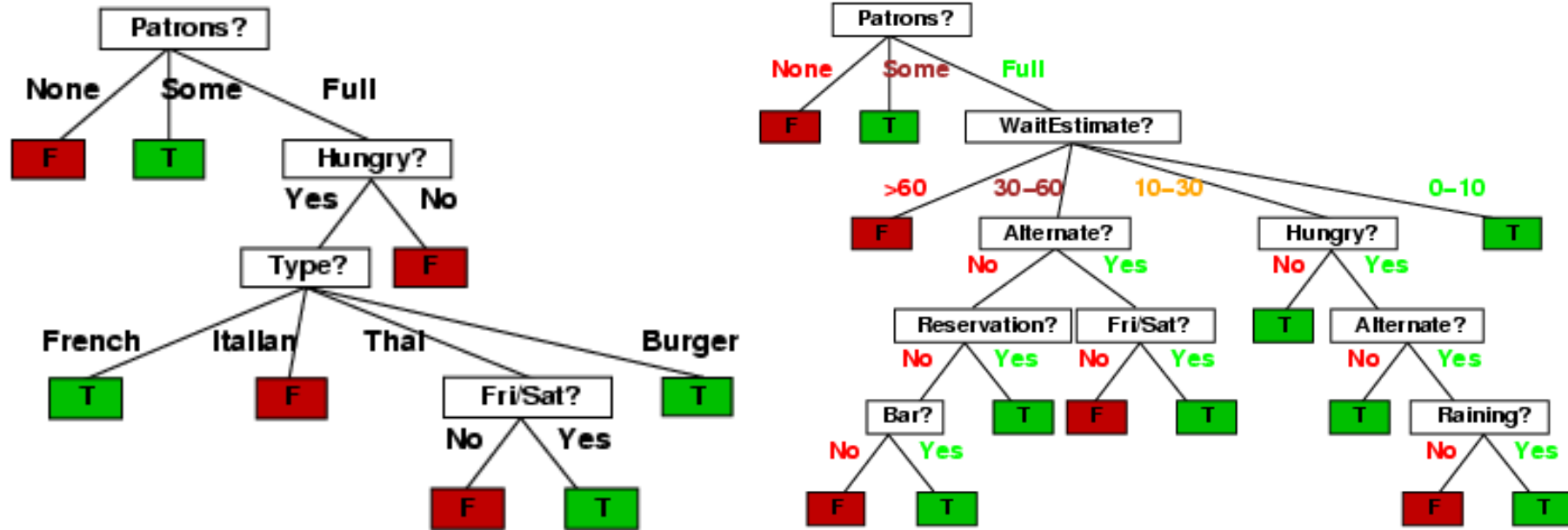
# DTL

```
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)?

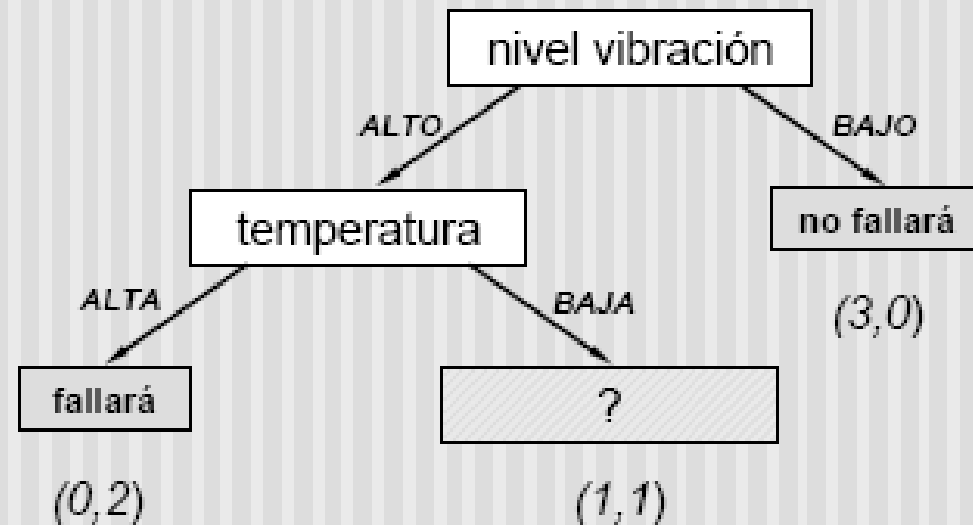
# Ejemplo detallado

**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)?



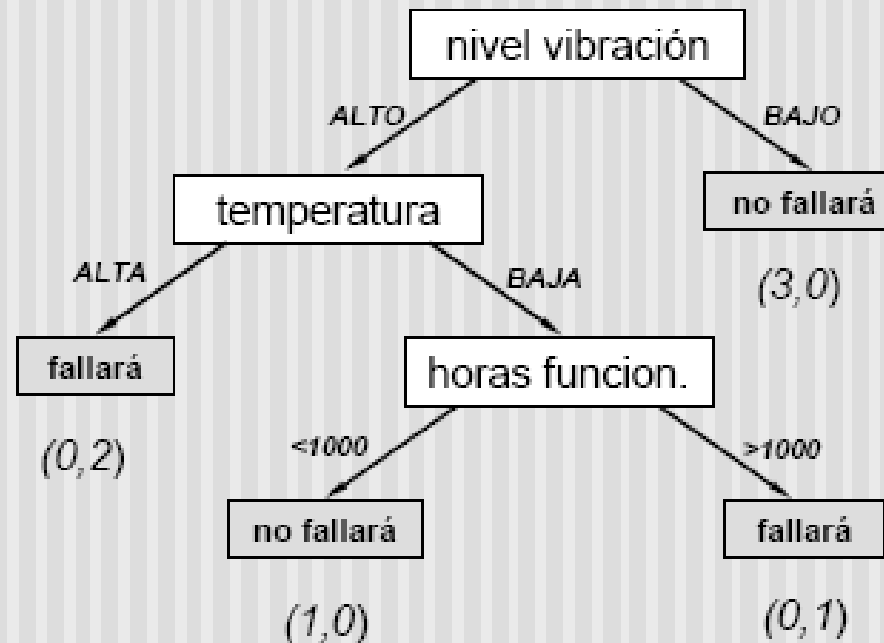
# Ejemplo detallado

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!

# Elección de los atributos de test

- Un atributo perfecto divide los ejemplos en conjuntos que contienen solo ejemplos positivos o negativos.
- Definir una medida de atributo “bastante adecuado” o “inadecuado”.

$$I(P(v_1), \dots, P(v_n)) = \sum_{i=1}^n -P(v_i) \log_2 P(v_i)$$

- Para un conjunto de entrenamiento que contenga  $p$  ejemplos positivos y  $n$  ejemplos negativos

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

# Elección de los atributos de test

- Intuición: Mide la ausencia de “homogeneidad” de la clasificación.
- Teoría de la Información: cantidad media de información (en bits) necesaria para codificar la clasificación de un ejemplo.
- Ejemplos:
  - $I([9+, 5-]) = - 9/14 \times \log_2 9/14 - 5/14 \times \log_2 5/14 = 0,94$
  - $I([k+, k-]) = 1$  (ausencia total de homogeneidad)
  - $I([p+, 0]) = I([0, n-]) = 0$  (homogeneidad total)

# Ganancia de información

- Entropía esperada después de usar un atributo A en el árbol:

$$\text{resto} (A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Ganancia de información esperada después de usar un atributo:

$$\text{Ganancia} (A) = I\left(\frac{p}{p + n}, \frac{n}{p + n}\right) - \text{resto} (A)$$

- Se elige el atributo con mayor valor de G (ID3).

# Ejemplo

Para el conjunto de entrenamiento,  $p = n = 6$ ,  $I(6/12, 6/12) = 1$  bit

Considerando los atributos Clientes y Tipo:

$$Ganancia \text{ (Clientes)} = 1 - \left[ \frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I\left(\frac{2}{6}, \frac{4}{6}\right) \right] = .0541 \text{ bits}$$

$$Ganancia \text{ (Tipo)} = 1 - \left[ \frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right) \right] = 0 \text{ bits}$$

# Otras medidas

El criterio de ganancia tiene un fuerte sesgo a favor de test con muchas salidas:

Ratio de Ganancia:

$$RGanancia(A) = \frac{Ganancia(A)}{dINFO(A)}$$

con

$$dINFO(A) = - \sum_{i=1}^v \frac{p_i + n_i}{p + n} \log_2 \left( \frac{p_i + n_i}{p + n} \right)$$

# Heurística Gini

- Otro criterio es la heurística Gini que es similar a la ganancia de información pero reemplazando la entropía por:

$$g(A) = 1 - \sum_{i=1}^v p_i^2$$

- La calidad de las particiones quedan entonces:

$$Gini(A) = g(A) - \sum_{i=1}^v p_i \cdot g(A_i)$$



# Atributos de entrada continuos

- Se ordenan los valores del atributo, y se especifica la clase a la que pertenecen

30	36	44	60	72	78
no	no	sí	sí	sí	no

- Hay dos puntos de corte, ~~(36-44)~~ y ~~(72-78)~~, en los que se pueden calcular los valores medios: 40 y 75
- Se crean los siguientes nodos de decisión

$\text{peso} < 40$ ,  $40 < \text{peso}$ ,  $\text{peso} < 75$ ,  $75 < \text{peso}$

# Atributos de salida continuos

- Árbol de regresión: en cada hoja se dispone de una función lineal de algún subconjunto de atributos numéricos.
- El algoritmo de aprendizaje debe decidir cuándo dejar de dividir para comenzar a aplicar regresión lineal utilizando los atributos restantes (o un subconjunto).

# WEKA



- Algoritmos clásicos como CART, ID3, C4.5 o CN2.
- También existen sistemas de inducción de reglas o árboles de inducción integrados en paquetes de minería de datos o aplicaciones estadísticas.
- WEKA es un conjunto de librerías JAVA para la extracción del conocimiento desde bases de datos.
- Fue desarrollado por la Universidad de Waikato bajo licencia GPL.
- WEKA contiene una gran cantidad de métodos y un entorno de experimentación para comparar el rendimiento de los algoritmos utilizando técnicas estadísticas.
- <http://www.cs.waikato.ac.nz/ml/weka/>