

INTELIGENCIA ARTIFICIAL

Tema 6

MATERIAL COMPLEMENTARIO

**EL PROBLEMA DE LA
CLASIFICACIÓN**

ÁRBOLES DE DECISIÓN

El problema de la clasificación

- 1. Definición del problema de clasificación**
2. Etapas en el proceso de clasificación
3. Criterios para evaluar un clasificador
4. Algunos ejemplos de clasificadores sencillos
5. Evaluación de clasificadores

1. Definición del problema de clasificación. Ejemplo

Ejemplo: El problema de clasificación de la flor de *Iris*

Problema simple muy conocido: *clasificación de lirios*.

Tres clases de lirios: *setosa*, *versicolor* y *virginica*.

Cuatro atributos: *longitud* y *anchura* de *pétalo* y *sépalo*, respectivamente.

150 ejemplos, 50 de cada clase.

Disponible en

<http://www.ics.uci.edu/~mlearn/MLRepository.html>

	Sepal length				Sepal width				Petal length				Petal width				Class
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}		
x_1	5.9	3.0	4.2	1.5													Iris-versicolor
x_2	6.9	3.1	4.9	1.5													Iris-versicolor
x_3	6.6	2.9	4.6	1.3													Iris-versicolor
x_4	4.6	3.2	1.4	0.2													Iris-setosa
x_5	6.0	2.2	4.0	1.0													Iris-versicolor
x_6	4.7	3.2	1.3	0.2													Iris-setosa
x_7	6.5	3.0	5.8	2.2													Iris-virginica
x_8	5.8	2.7	5.1	1.9													Iris-virginica
:	:	:	:	:												:	
x_{149}	7.7	3.8	6.7	2.2													Iris-virginica
x_{150}	5.1	3.4	1.5	0.2													Iris-setosa



setosa



versicolor

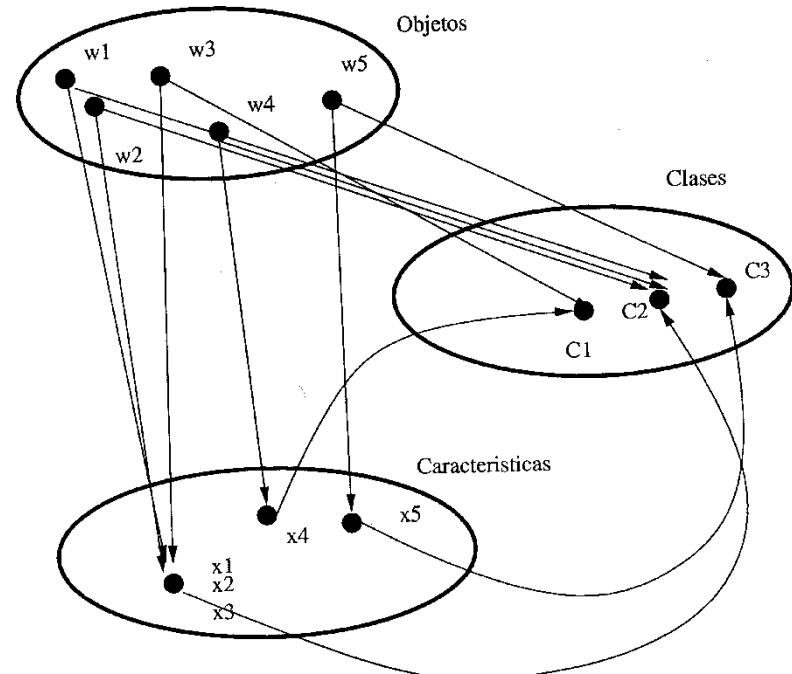


virginica

1. Definición del problema de clasificación

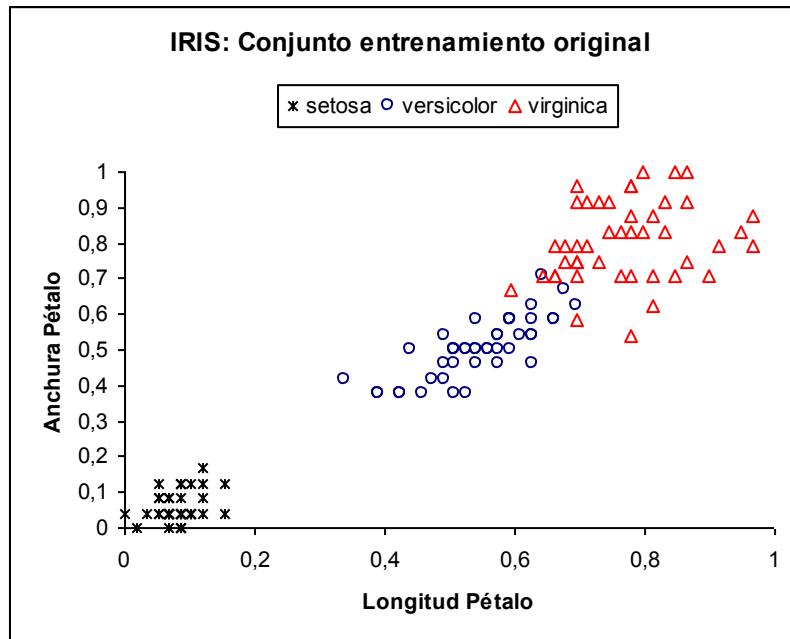
- Es un tipo de *aprendizaje supervisado*: Se conoce la clase verdadera de cada uno de los ejemplos que se utilizan para construir el clasificador
- El *problema de clasificación* consiste en predecir una determinada clase (categórica) para un objeto
- La *tarea de clasificación*: Dados un conjunto de ejemplos ya clasificados, construir un modelo o clasificador que permita clasificar nuevos casos

El problema fundamental de la clasificación está directamente relacionado con la separabilidad de las clases.



1. Definición del problema de clasificación. Ejemplo

Ejemplos de conjuntos seleccionados sobre *Iris*:

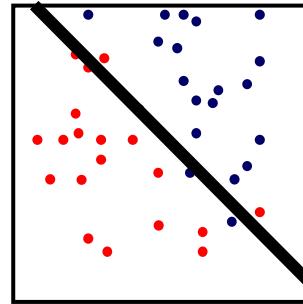
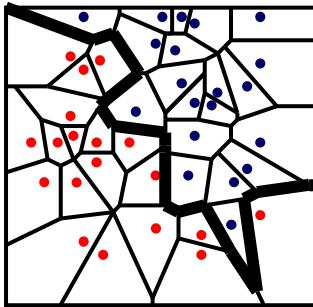
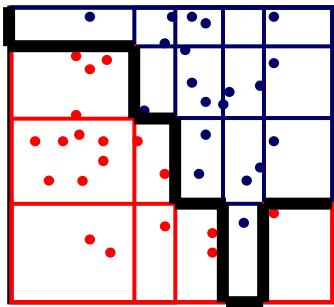


El problema fundamental de la clasificación está directamente relacionado con la separabilidad de las clases.

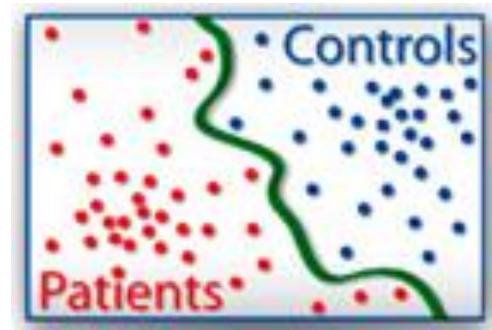
En este problema la dificultad está en la separación entre las clases versicolor y virgínica

1. Definición del problema de clasificación

- Un *clasificador* puede ser un conjunto de reglas, un árbol de decisión, una red neuronal, etc.



- Aplicaciones típicas:
 - Aprobación de créditos, marketing directo, detección de fraudes, diagnóstico médico...



1. Definición del problema de clasificación

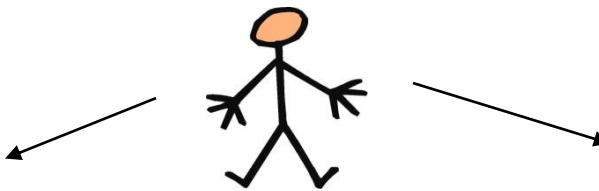
- Un objeto se describe a través de un conjunto de características (variables o atributos)

$$X \rightarrow \{X_1, X_2, \dots, X_n\}$$

$$e_i \rightarrow \{x_1, x_2, \dots, x_n, c_k\}$$

- Edad
- Astigmatismo
- Ratio de lagrimeo
- Miopía

CLASIFICACIÓN: Tipo de lentillas



-Ingresos
-Deudas
-Propiedades
...
-CLASIFICACIÓN:
Conceder el crédito

- **Objetivo de la tarea de clasificación:** Clasificar el objetivo dentro de una de las categorías de la clase $C = \{c_1, \dots, c_k\}$

$$f: X_1 \times X_2 \times \dots \times X_n \rightarrow C$$

- Las características o variables elegidas dependen del problema de clasificación

Kaggle: The Home of Data Science



Digit Recognizer

Classify handwritten digits using the famous MNIST data

3 months to go · [Getting Started](#)

1,116 teams
6,153 kernels
Knowledge



Titanic: Machine Learning from Disaster

Predict survival on the Titanic using Excel, Python, R & Random Forests

3 months to go · [Getting Started](#)

5,095 teams
9,571 kernels
Knowledge

0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9

MNIST data

Kaggle: The Home of Data Science

Santander Customer Satisfaction

5 1 2 3 teams

5 6 9 6 players

9 3 5 5 8 entries



Banco
Santander

- Objetivo: crear un modelo que **prediga** qué clientes no están satisfechos
- Premio **de 60.000€** entre las 3 mejores soluciones
- Competición activa desde el **2 de marzo hasta el 2 de mayo, 2016**
- Conjunto de datos: 76020 instancias y 371 variables
- Datos públicos 50%

Kaggle: The Home of Data Science

otto group

\$10,000 • 1,987 teams

Otto Group Product Classification Challenge

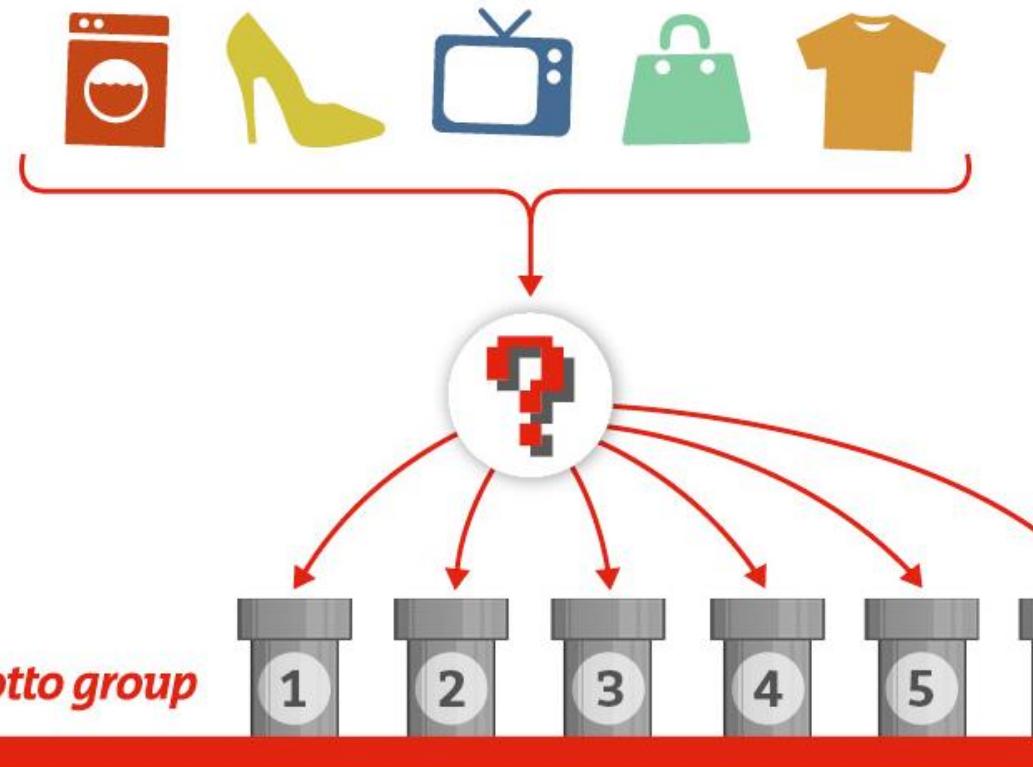
Tue 17 Mar 2015

Enter/Merge by

Mon 18 May 2015 (39 days to go)

A KAGGLE competition with a Multiclass and imbalanced problem

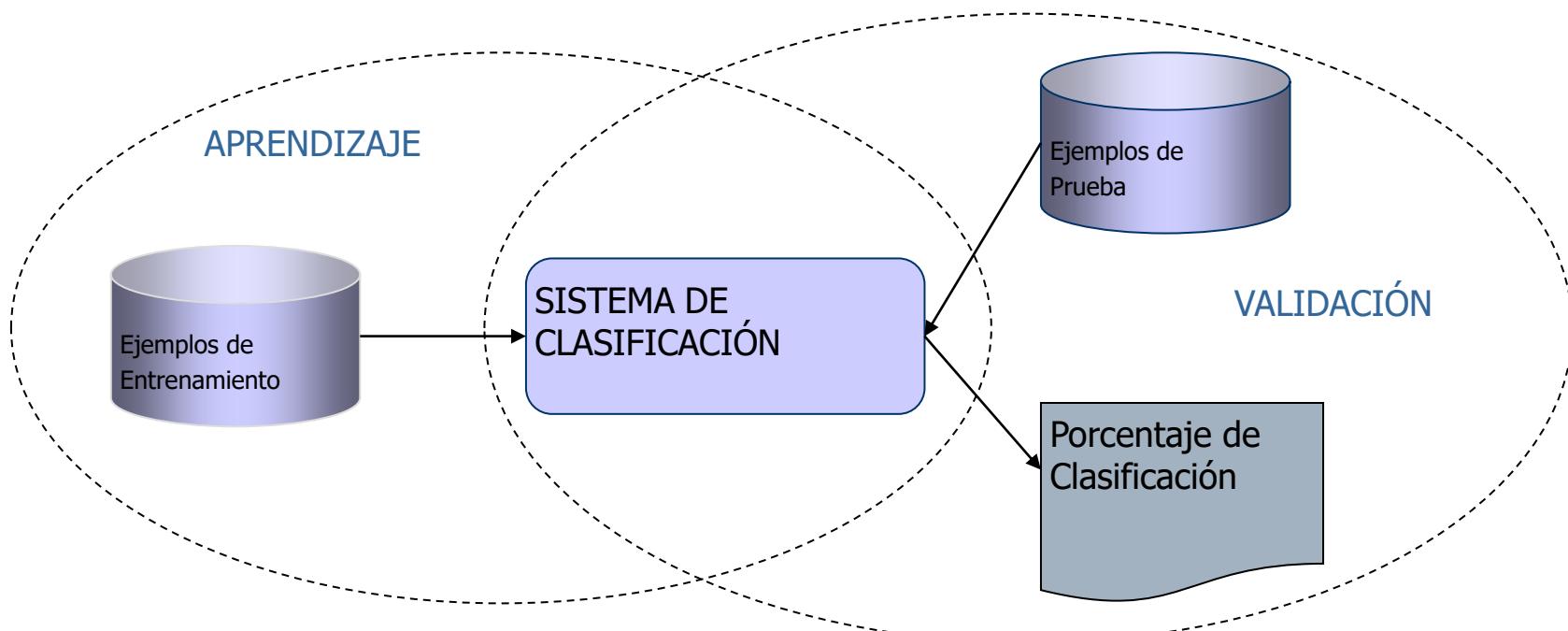
For this competition, we have provided a dataset with 93 features for more than 200,000 products. The objective is to build a predictive model which is able to distinguish between our main product categories. The winning models will be open sourced.



El problema de la clasificación

1. Definición del problema de clasificación
- 2. Etapas en el proceso de clasificación**
3. Criterios para evaluar un clasificador
4. Algunos ejemplos de clasificadores sencillos
5. Evaluación de clasificadores

2. Etapas en el proceso de clasificación



2. Etapas en el proceso de clasificación

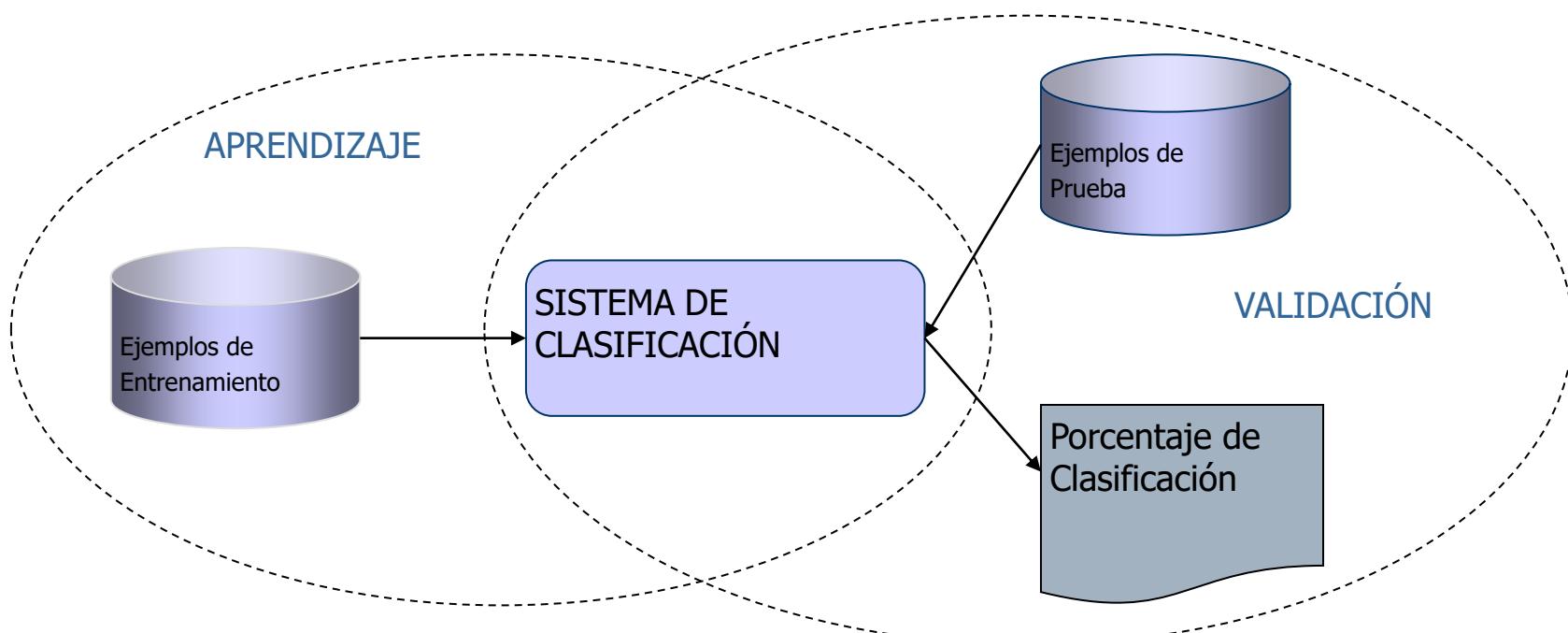
1. Construcción del modelo que describe el conjunto de clases (predeterminadas): Fase de Aprendizaje

- Cada ejemplo (tupla) se sabe que pertenece a una clase (etiqueta del atributo clase)
- Se utiliza un conjunto de ejemplos para la construcción del modelo: **conjunto de entrenamiento** (*training set*)
- El modelo obtenido se representa como un conjunto de reglas de clasificación, árboles de decisión, fórmula matemática, ...

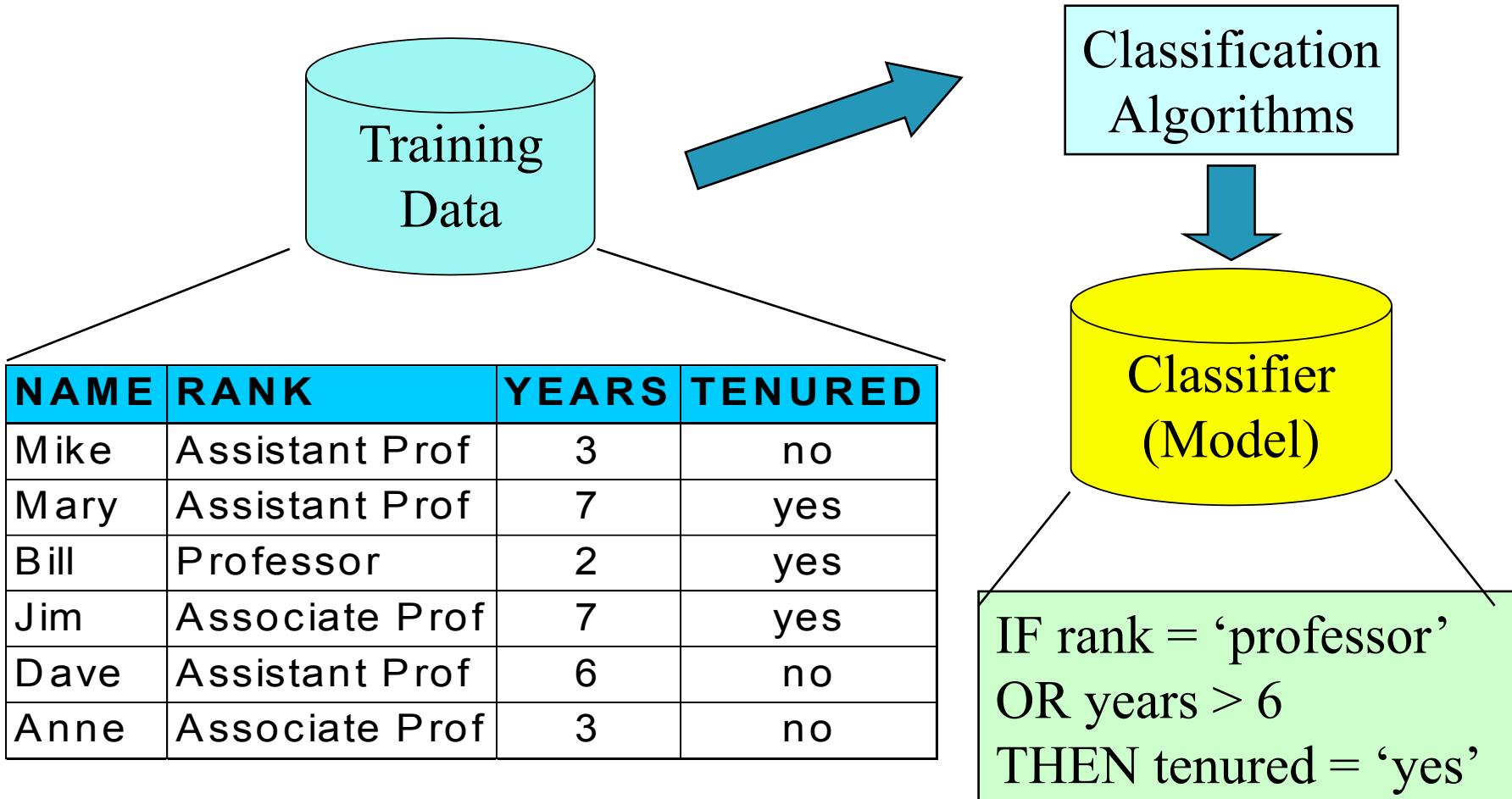
2. Utilización del modelo: Validación

- Estimación de la precisión del modelo
 - Se utiliza un conjunto de ejemplos distintos de los utilizados para la construcción del modelo: **conjunto de prueba** (*test set*)
 - Si el conjunto de test no fuese independiente del de entrenamiento ocurría un proceso de sobreajuste (*overfitting*)
 - Para cada ejemplo de test se compara la clase determinada por el modelo con la clase real (conocida)
 - El ratio de precisión es el porcentaje de ejemplos de test que el modelo clasifica correctamente

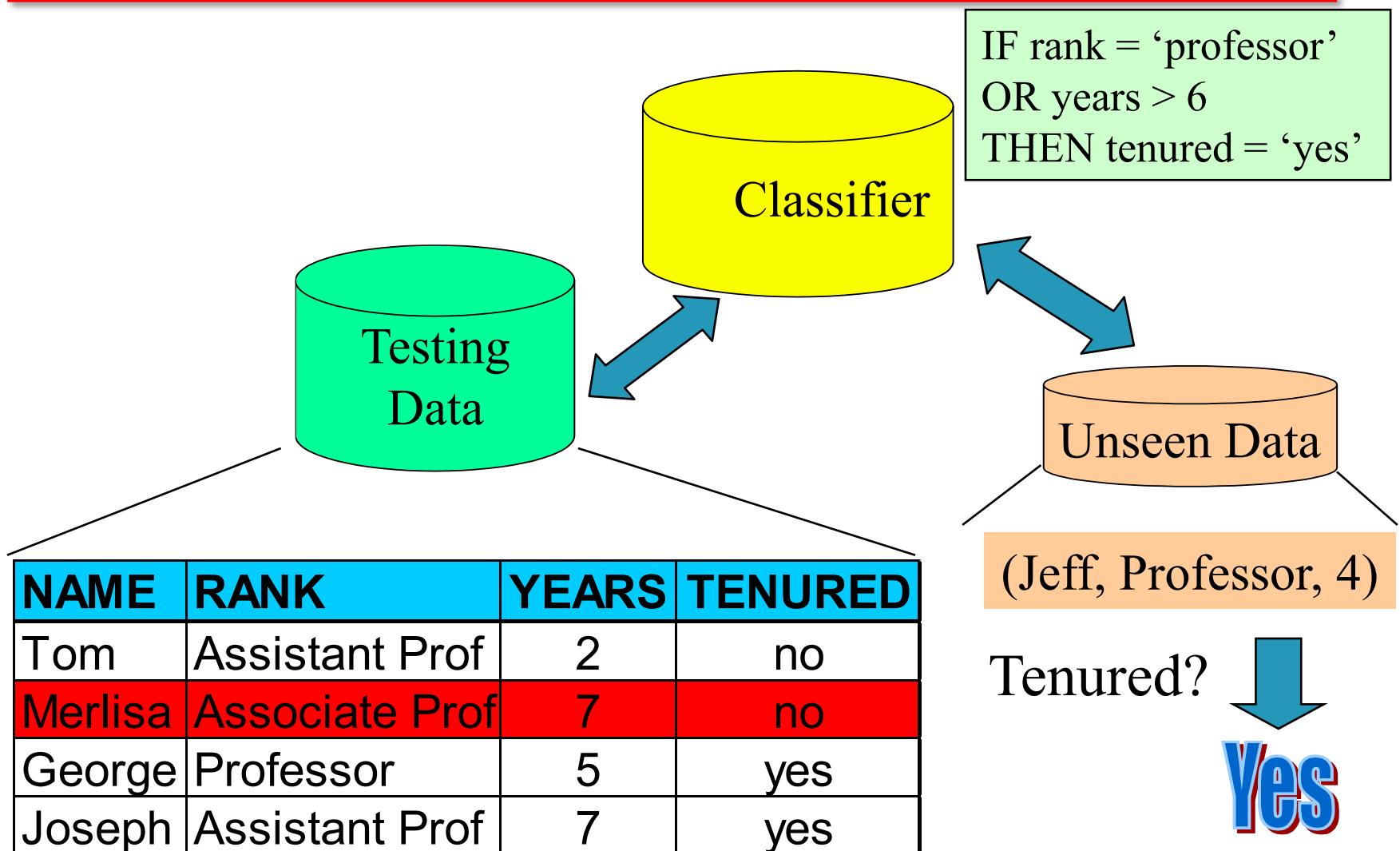
2. Etapas en el proceso de clasificación



2. Etapas en el proceso de clasificación



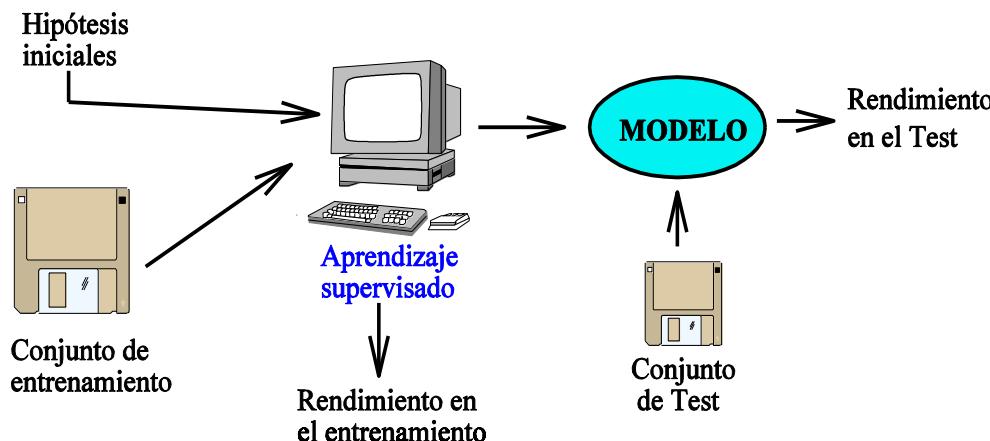
2. Etapas en el proceso de clasificación



El problema de la clasificación

1. Definición del problema de clasificación
2. Etapas en el proceso de clasificación
- 3. Criterios para evaluar un clasificador**
4. Algunos ejemplos de clasificadores sencillos
5. Evaluación de clasificadores

3. Criterios para evaluar un clasificador



- Rendimiento (matriz de confusión):

		Clasificación como	
		Si	No
Clase	SI	Verdadero positivo (VP)	Falso negativo (FN)
	real NO	Falso Positivo (FP)	Verdadero Negativo (VN)

$$N = VP + VN + FP + FN$$

- Tasa de acierto

$$s = \frac{VP + VN}{N}$$

- Tasa de error

$$\varepsilon = 1 - s$$

3. Criterios para evaluar un clasificador

- **Precisión o exactitud** (s, ϵ)
 - En ocasiones se debe considerar el costo de la clasificación incorrecta
- **Velocidad**
 - Tiempo necesario para la construcción del modelo
 - Tiempo necesario para usar el modelo
- **Robustez**: capacidad para tratar con valores desconocidos
- **Escalabilidad**: Aumento del tiempo necesario (en construcción y evaluación) con el tamaño de la BD
- **Interpretabilidad**: comprensibilidad del modelo obtenido
- **Complejidad del modelo**: Tamaño del árbol de clasificación, número de reglas, antecedentes en las reglas,...

3. Criterios para evaluar un clasificador

- Matriz de confusión

Dado un problema de clasificación con m clases, una matriz de confusión es una matriz $m \times m$ en la que una entrada $c_{i,j}$ indica el número de ejemplos que se han asignado a la clase c_j , cuando la clase correcta es c_i

Ejemplo: Para la BD Height, si suponemos que output1 es la clasificación correcta y output2 es la que hace el clasificador, la matriz de confusión es

3. Criterios para evaluar un clasificador

Name	Gender	Height	Output1	Output2
Kristina	F	1.6m	Short	Medium
Jim	M	2m	Tall	Medium
Maggie	F	1.9m	Medium	Tall
Martha	F	1.88m	Medium	Tall
Stephanie	F	1.7m	Short	Medium
Bob	M	1.85m	Medium	Medium
Kathy	F	1.6m	Short	Medium
Dave	M	1.7m	Short	Medium
Worth	M	2.2m	Tall	Tall
Steven	M	2.1m	Tall	Tall
Debbie	F	1.8m	Medium	Medium
Todd	M	1.95m	Medium	Medium
Kim	F	1.9m	Medium	Tall
Amy	F	1.8m	Medium	Medium
Wynette	F	1.75m	Medium	Medium

	Asignación		
	Short	Medium	Tall
Short	0	4	0
Medium	0	5	3
Tall	0	1	2

Matriz de confusión

El problema de la clasificación

1. Definición del problema de clasificación
2. Etapas en el proceso de clasificación
3. Criterios para evaluar un clasificador
- 4. Algunos ejemplos de clasificadores sencillos**
5. Evaluación de clasificadores

4. Algunos ejemplos de clasificadores sencillos

Edad	Miopía	Astigmatismo	Lagrimeo	Lentes
Joven	Miope	No	Reducido	Ninguna
Joven	Miope	No	Normal	Blandas
Joven	Miope	Si	Reducido	Ninguna
Joven	Miope	Si	Normal	Duras
Joven	Hipermétrope	No	Reducido	Ninguna
Joven	Hipermétrope	No	Normal	Blandas
Joven	Hipermétrope	Si	Reducido	Ninguna
Joven	Hipermétrope	Si	Normal	Duras
Pre-presbiopico	Miope	No	Reducido	Ninguna
Pre-presbiopico	Miope	No	Normal	Blandas
Pre-presbiopico	Miope	Si	Reducido	Ninguna
Pre-presbiopico	Miope	Si	Normal	Duras
Pre-presbiopico	Hipermétrope	No	Reducido	Ninguna
Pre-presbiopico	Hipermétrope	No	Normal	Blandas
Pre-presbiopico	Hipermétrope	Si	Reducido	Ninguna
Pre-presbiopico	Hipermétrope	Si	Normal	Ninguna
Presbiopico	Miope	No	Reducido	Ninguna
Presbiopico	Miope	No	Normal	Ninguna
Presbiopico	Miope	Si	Reducido	Ninguna
Presbiopico	Miope	Si	Normal	Duras
Presbiopico	Hipermétrope	No	Reducido	Ninguna
Presbiopico	Hipermétrope	No	Normal	Blandas
Presbiopico	Hipermétrope	Si	Reducido	Ninguna
Presbiopico	Hipermétrope	Si	Normal	Ninguna

4. Algunos ejemplos de clasificadores sencillos

Clasificador ZeroR

- Es el clasificador más sencillo
- Todas las instancias se clasifican como pertenecientes a la clase mayoritaria
- Se usa como caso base para realizar comparaciones (cualquier algoritmo debería al menos igualar su rendimiento)
- Ejemplo: Con la BD Lentes de contacto
 - Lentes=ninguna 15/24
 - Lentes=blandas 5/24
 - Lentes=duras 4/24

Por tanto la regla sería: lentes=ninguna

$$s = 0.625$$

$$\varepsilon = 0.375$$

4. Algunos ejemplos de clasificadores sencillos

Clasificador OneR

- Objetivo: crear un clasificador formado por reglas con una única variable en el antecedente
- Se generan todas las reglas del tipo
Si variable=valor entonces Clase = categoría
- También se utiliza como algoritmo para realizar comparaciones

For each attribute,

For each value of the attribute, make a rule as follows:

count how often each class appears

find the most frequent class

make the rule assign that class to this attribute-value

Calculate the error rate of the rules

Choose the rules with the smallest error rate

Nota: Los valores perdidos se utilizan como un valor especial de las variables

4. Algunos ejemplos de clasificadores sencillos

Ejemplo: Con la BD lentes de contacto

Regla			ninguna	blandas	Duras
Si edad=joven	entonces	Lentes=ninguna	4	2	2
Si edad=pre-presbiopico	entonces	Lentes=ninguna	5	2	1
Si edad=presbiopico	entonces	Lentes=ninguna	6	1	1
Si miopia=miope	entonces	Lentes=ninguna	7	2	3
Si miopía=hipermétrope	entonces	Lentes=ninguna	8	3	1
Si astigmatismo=no	entonces	Lentes=ninguna	7	5	0
Si astigmatismo=sí	entonces	Lentes=ninguna	8	0	4
Si lagrimeo=reducido	entonces	Lentes=ninguna	12	0	0
Si lagrimeo=normal	entonces	Lentes=blandas	3	5	4

4. Algunos ejemplos de clasificadores sencillos

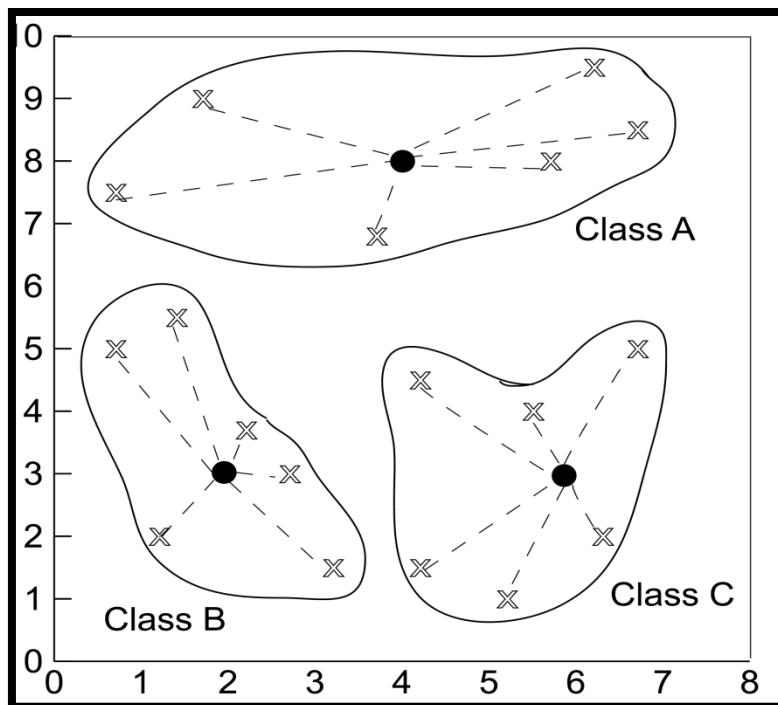
Clasificador Del Vecino Más Cercano (k-NN)

- k-NN (*k vecinos más cercanos*) es uno de los clasificadores más utilizados por su simplicidad.
- El proceso de aprendizaje de este clasificador consiste en almacenar una tabla con los ejemplos disponibles, junto a la clase asociada a cada uno de ellos.
- Ante un nuevo ejemplo a clasificar, se calcula su distancia (usaremos la Euclídea) con respecto a los *n* ejemplos existentes en la tabla, y se consideran los *k* más cercanos.
- El nuevo ejemplo se clasifica según la clase mayoritaria de los *k* ejemplos más cercanos.
- El caso más sencillo es cuando $k = 1$ (*1-NN*).

4. Algunos ejemplos de clasificadores sencillos

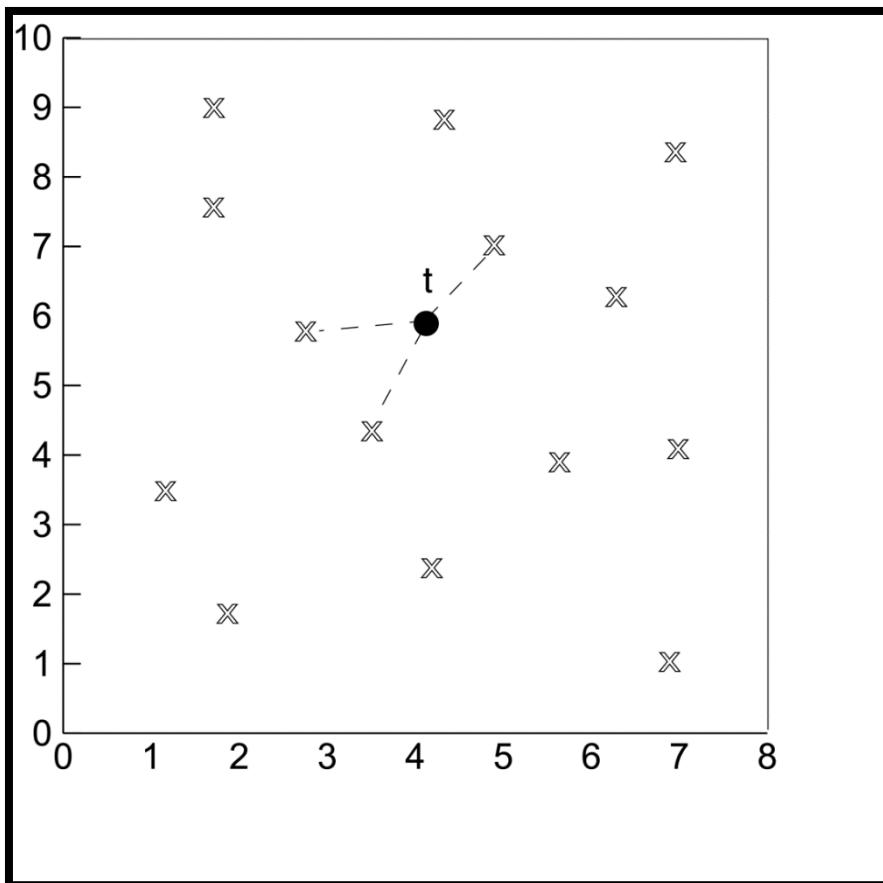
Clasificador Del Vecino Más Cercano (k-NN): Basado en distancias

Basado en Distancias



4. Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN): Ejemplo para k=3



4. Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN)

Dado el siguiente conjunto con 4 instancias, 3 atributos y 2 clases:

x_1 : 0.4 0.8 0.2 positiva

x_2 : 0.2 0.7 0.9 positiva

x_3 : 0.9 0.8 0.9 negativa

x_4 : 0.8 0.1 0.0 negativa

Calculamos la distancia del ejemplo con todos los del conjunto:

$$d(x_1, y_1) = \sqrt{(0.4 - 0.7)^2 + (0.8 - 0.2)^2 + (0.2 - 0.1)^2} = 0.678$$

$$d(x_2, y_1) = \sqrt{(0.2 - 0.7)^2 + (0.7 - 0.2)^2 + (0.9 - 0.1)^2} = 1.068$$

$$d(x_3, y_1) = \sqrt{(0.9 - 0.7)^2 + (0.8 - 0.2)^2 + (0.9 - 0.1)^2} = 1.020$$

$$d(x_4, y_1) = \sqrt{(0.8 - 0.7)^2 + (0.1 - 0.2)^2 + (0.0 - 0.1)^2} = 0.173$$

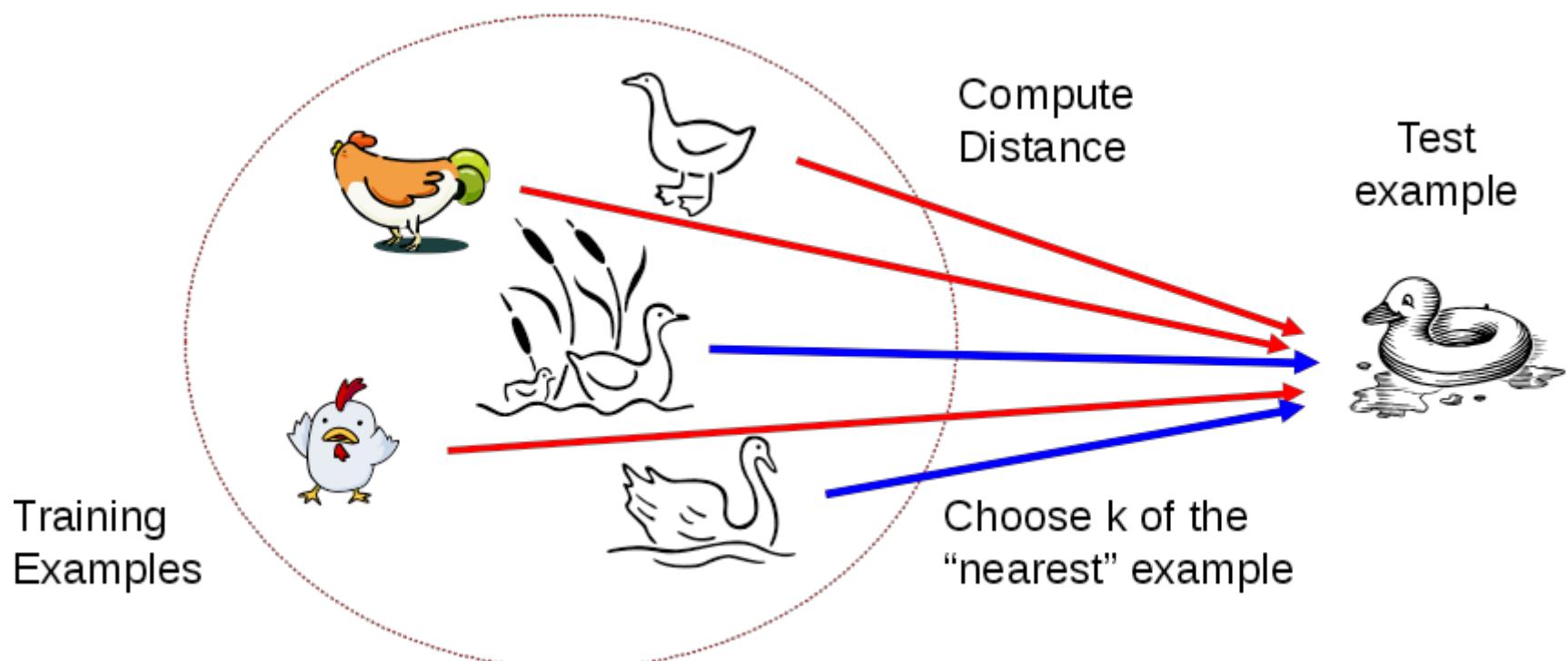
Por tanto, el ejemplo se clasificará con respecto a la clase *negativa*.

IMPORTANTE: Los atributos deben estar normalizados [0,1] para no priorizarlos sobre otros.

4. Algunos ejemplos de clasificadores sencillos

Clasificador del Vecino Más Cercano (k-NN)

- If it walks like a duck, quacks like a duck,
then it's probably a duck



4. Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN) Problema en las fronteras

Ejemplo: Diseño de un Clasificador para *Iris*

- Problema simple muy conocido: *clasificación de lirios*.
- Tres clases de lirios: *setosa*, *versicolor* y *virginica*.
- Cuatro atributos: *longitud* y *anchura* de *pétalo* y *sépalo*, respectivamente.
- 150 ejemplos, 50 de cada clase.
- Disponible en <http://www.ics.uci.edu/~mlearn/MLRepository.html>



4. Algunos ejemplos de clasificadores sencillos

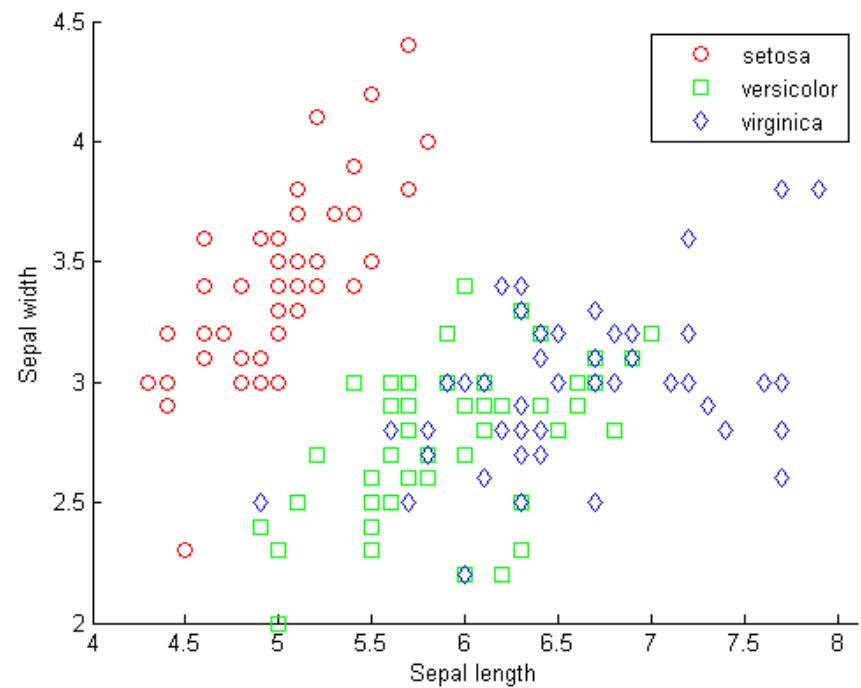
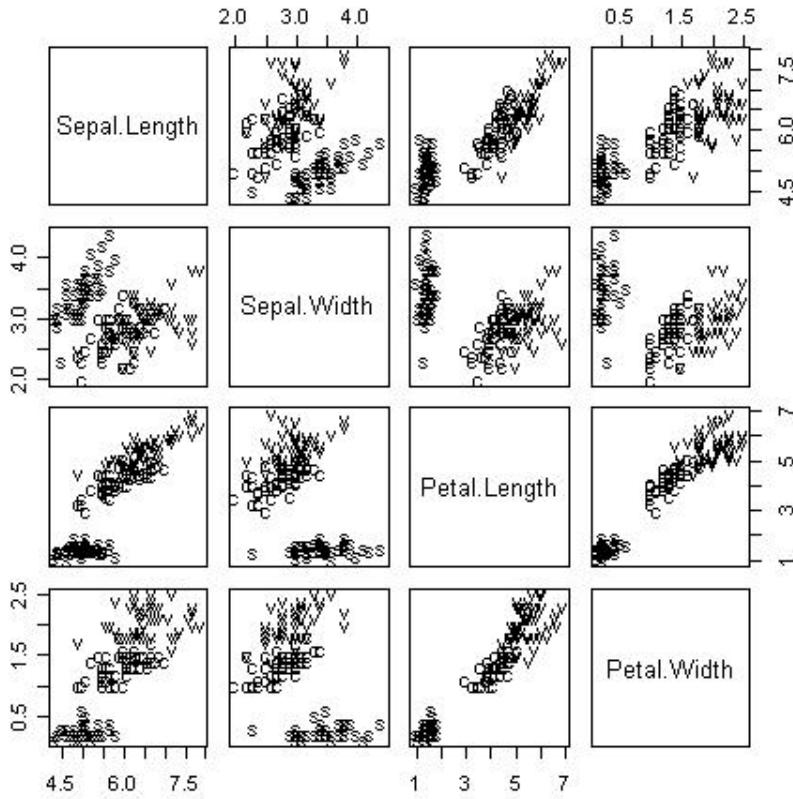
Clasificador Del Vecino Más Cercano (k-NN): Ejemplo IRIS.
Problema en las fronteras

	Sepal length	Sepal width	Petal length	Petal width	Class
	X_1	X_2	X_3	X_4	X_5
\mathbf{x}_1	5.9	3.0	4.2	1.5	Iris-versicolor
\mathbf{x}_2	6.9	3.1	4.9	1.5	Iris-versicolor
\mathbf{x}_3	6.6	2.9	4.6	1.3	Iris-versicolor
\mathbf{x}_4	4.6	3.2	1.4	0.2	Iris-setosa
\mathbf{x}_5	6.0	2.2	4.0	1.0	Iris-versicolor
\mathbf{x}_6	4.7	3.2	1.3	0.2	Iris-setosa
\mathbf{x}_7	6.5	3.0	5.8	2.2	Iris-virginica
\mathbf{x}_8	5.8	2.7	5.1	1.9	Iris-virginica
:	:	:	:	:	:
\mathbf{x}_{149}	7.7	3.8	6.7	2.2	Iris-virginica
\mathbf{x}_{150}	5.1	3.4	1.5	0.2	Iris-setosa

4. Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN). Problema en las fronteras

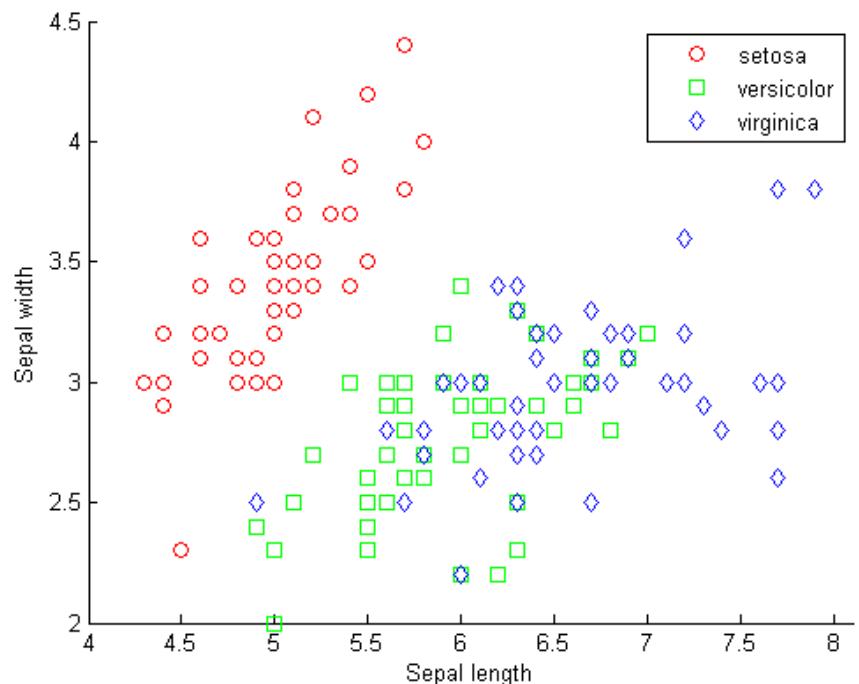
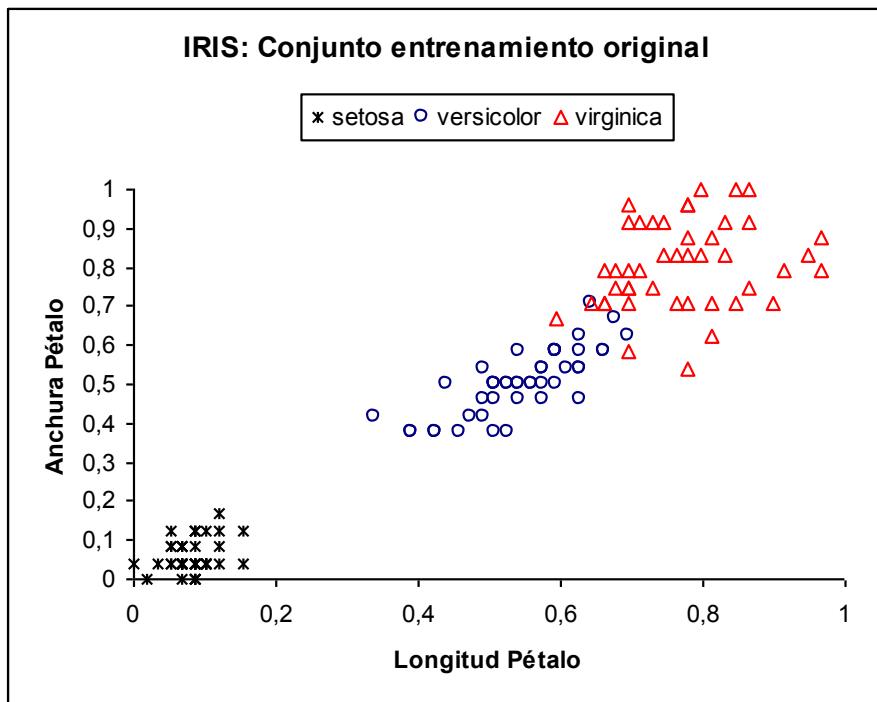
setosa, versicolor (C) y virginica



4. Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN). Problema en las fronteras

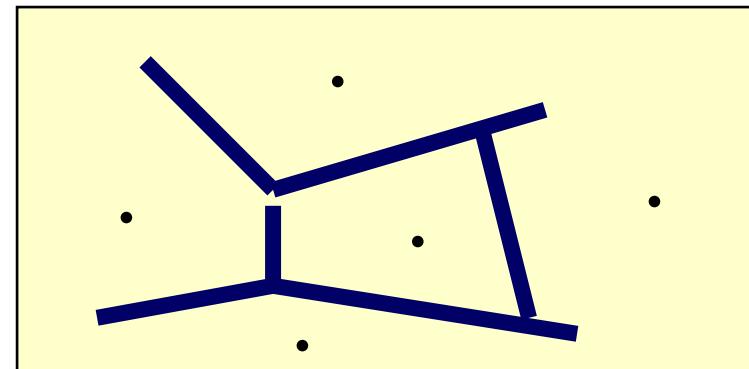
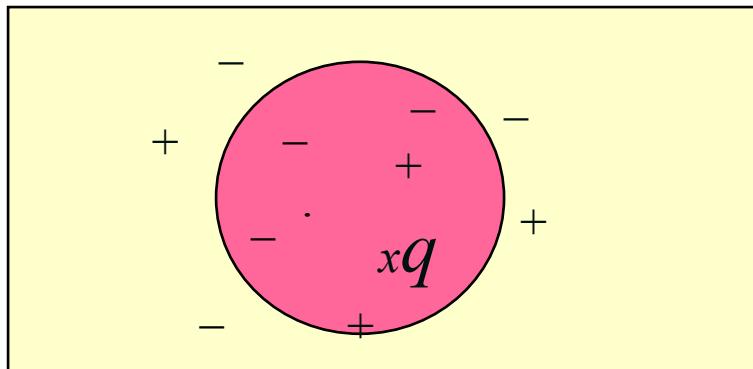
setosa, versicolor (C) y virginica



4. Algunos ejemplos de clasificadores sencillos

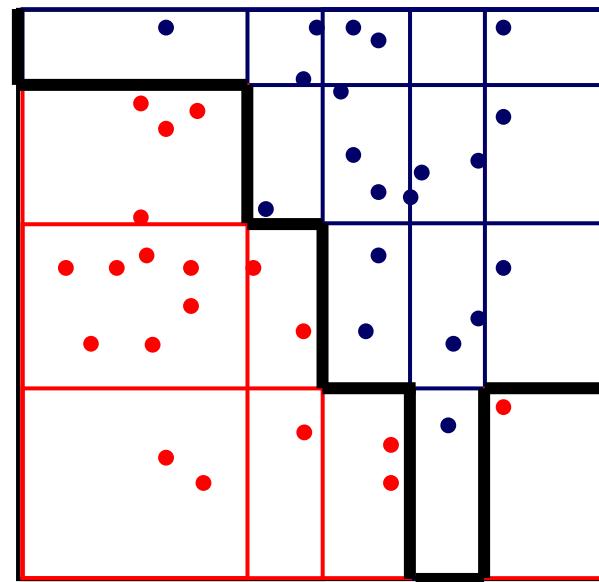
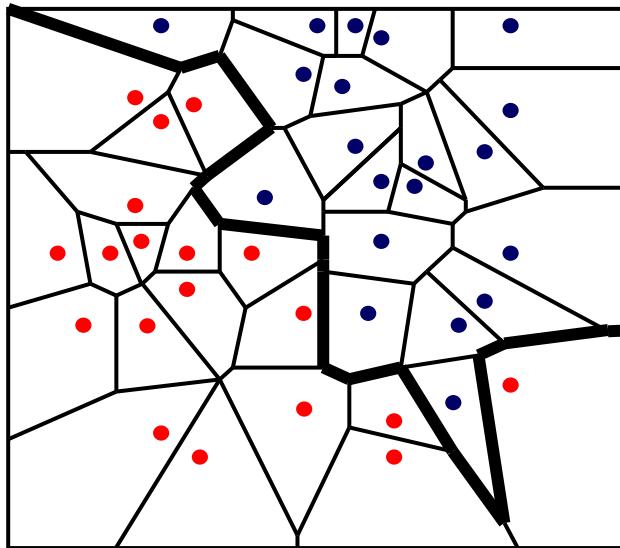
Clasificador Del Vecino Más Cercano (k-NN)

- k -NN devuelve la clase más repetida de entre todos los k ejemplos de entrenamiento cercanos a x_q .
- Diagrama de Voronoi: superficie de decisión inducida por 1-NN para un conjunto dado de ejemplos de entrenamiento.



4. Algunos ejemplos de clasificadores sencillos

Clasificador Del Vecino Más Cercano (k-NN): Basado en distancias (muy diferente a los basados en particiones)



El problema de la clasificación

1. Definición del problema de clasificación
2. Etapas en el proceso de clasificación
3. Criterios para evaluar un clasificador
4. Algunos ejemplos de clasificadores sencillos
- 5. Evaluación de clasificadores**

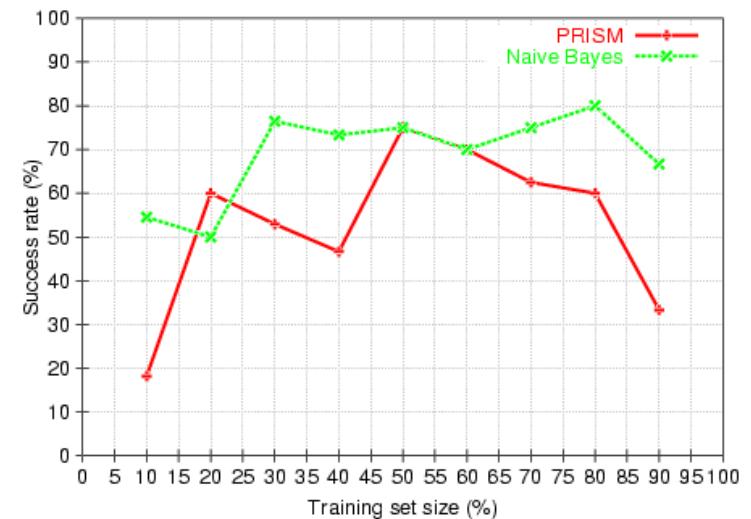
5. Evaluación de clasificadores

- El objetivo de los métodos de validación es realizar una estimación honesta de la bondad del clasificador construido
- Utilizar como bondad la tase de acierto sobre el conjunto de entrenamiento no es realista
 - El porcentaje obtenido suele ser demasiado optimista debido a que el modelo estará sobreajustado a los datos utilizados durante el proceso de aprendizaje
- Existen distintas técnicas de validación de clasificadores, entre ellas
 - *Hold-out*
 - Validación cruzada
 - *Leave-one-out*
 - *Bootstraping*

5. Evaluación de clasificadores

Hold-out

- Consiste en dividir la BD en dos conjuntos independientes: entrenamiento (CE) y test (CT)
- El tamaño del CE normalmente es mayor que el del CT ($2/3, 1/3, 4/5, 1/5, \dots$)
- Los elementos del CE suelen obtenerse mediante muestreo sin reemplazamiento de la BD inicial. El CT está formado por los elementos no incluidos en el CE
- Suele utilizarse en BBDD de tamaño grande



$$\text{Test set (\%)} + \text{Training set (\%)} = 100\%$$

5. Evaluación de clasificadores

Validación cruzada

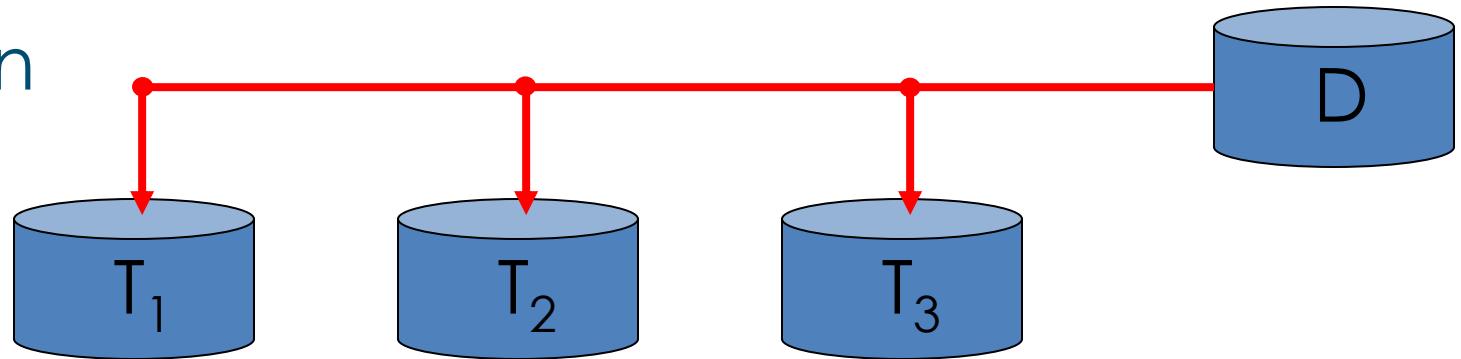
- Consiste en:
 1. Dividir la BD en k subconjuntos (*folds*), $\{S_1, \dots, S_k\}$ de igual tamaño
 2. Aprender k clasificadores utilizando en cada uno de ellos un CE distinto. Validar con el CT correspondiente

$$CE = S_1 \cup \dots \cup S_{i-1} \cup S_{i+1} \cup \dots \cup S_k$$

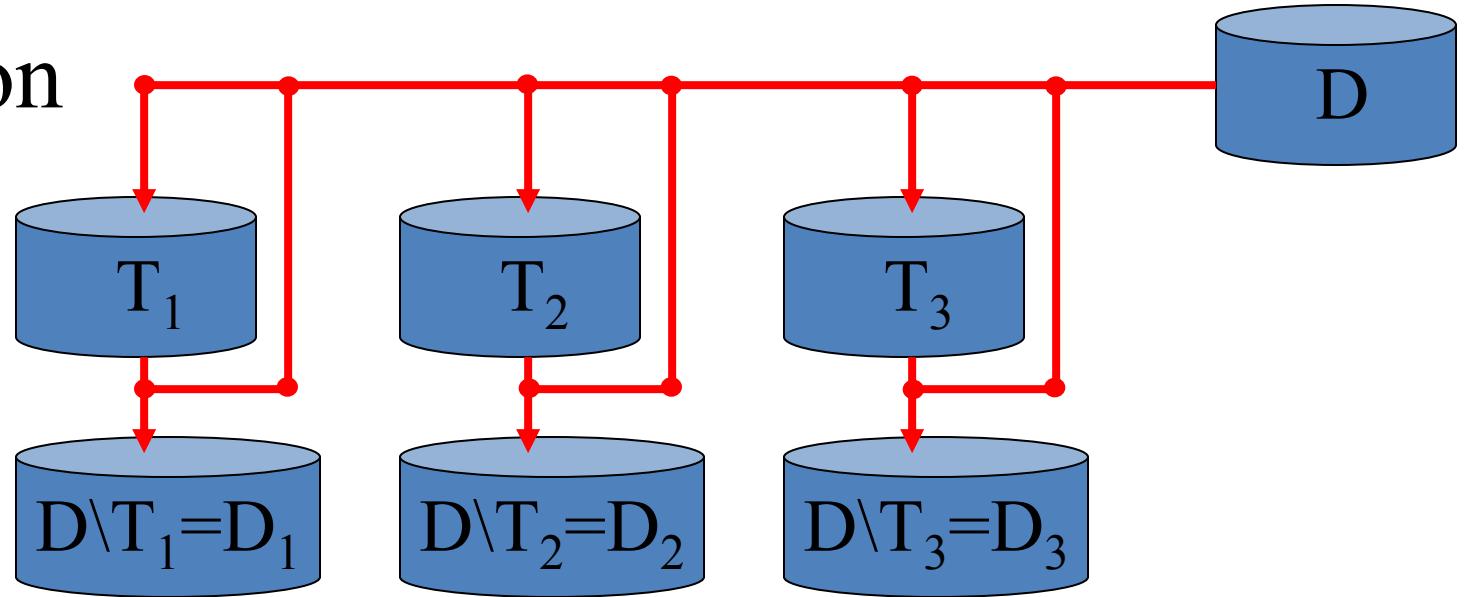
$$CT = S_i$$

- 3. Devolver como tasa de acierto (error) el promedio obtenido en las k iteraciones
- Validación cruzada estratificada: Los subconjuntos se estratifican en función de la variable clase
- Valores típicos de $k=5,10$
- Suele utilizarse en BBDD de tamaño moderado

- Partition

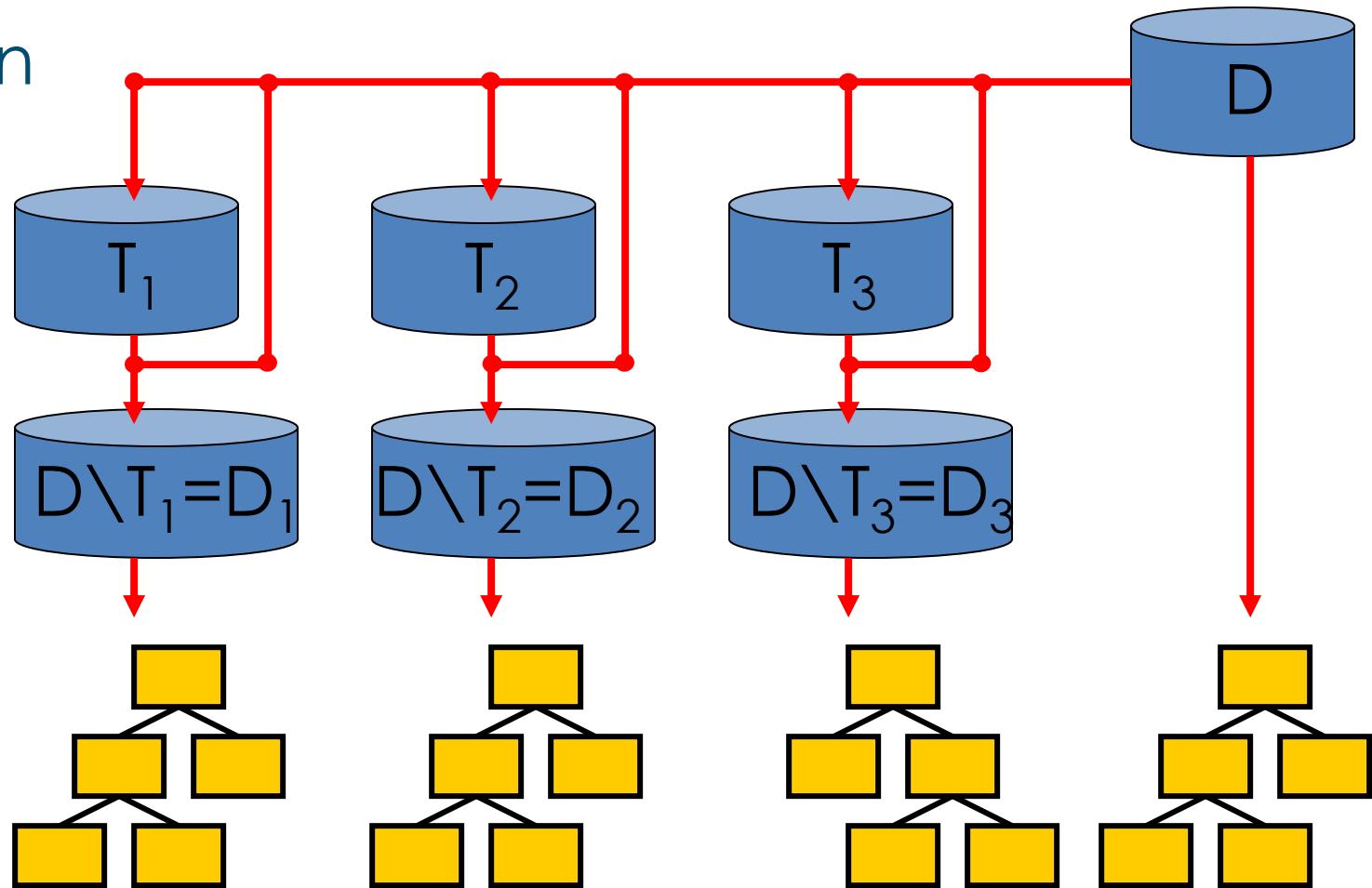


- Partition

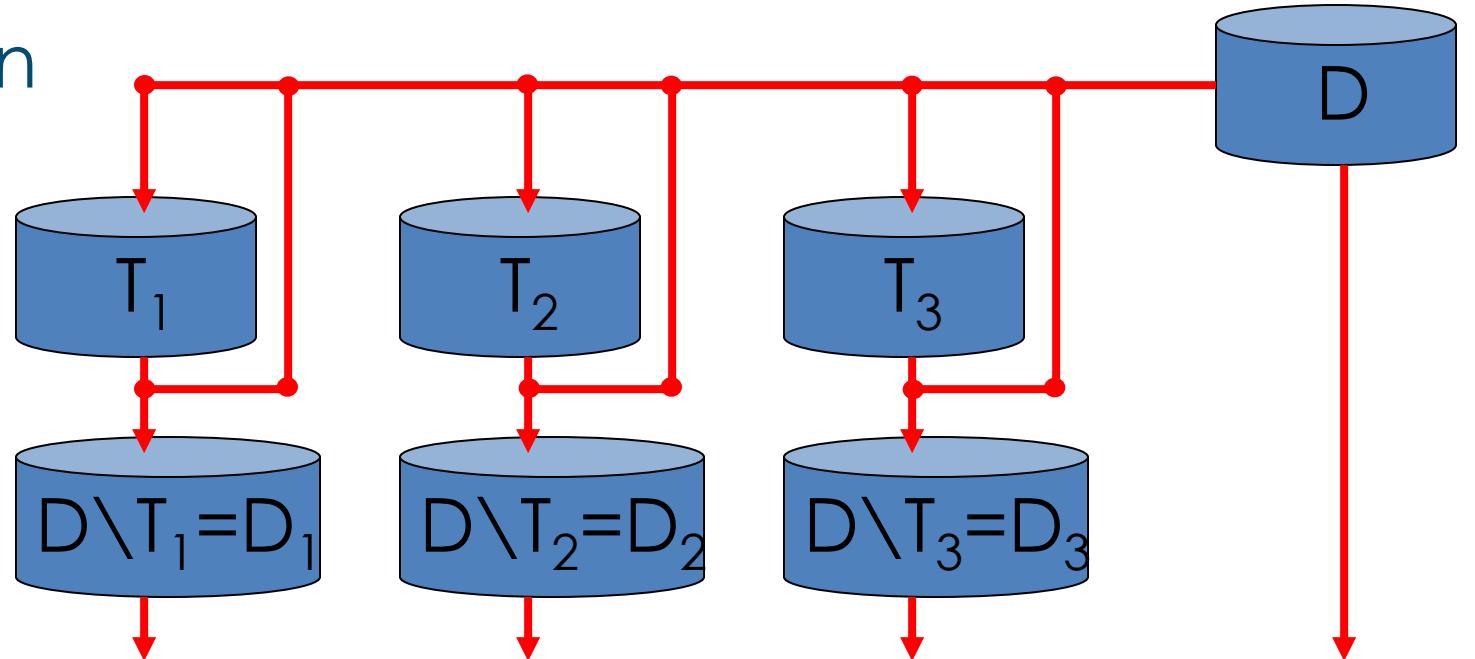


- Train

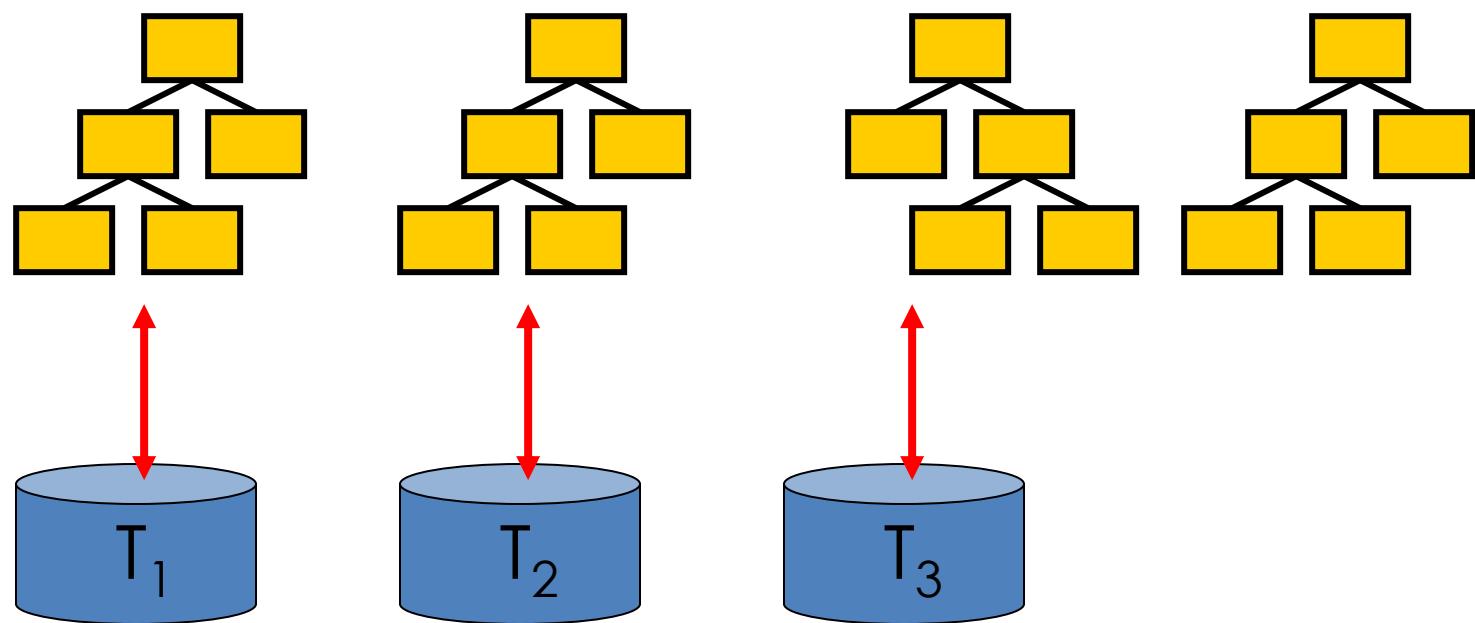
- Partition



- Partition



- Train



- Test

5. Evaluación de clasificadores

Leaving-one-out

- Es un caso especial de validación cruzada en el que k es igual al número de registros
 - Tiene la ventaja de que el proceso es determinista y de que en todo momento se utiliza el máximo posible de datos para la inducción del clasificador
 - Se utiliza en BBDD muy pequeñas, debido a su alto costo computacional

5. Evaluación de clasificadores

Bootstrap

- Está basado en el proceso de muestreo con reemplazo
- A partir de una BD con n registros se obtiene un CE con n casos
- Como CT se utilizan los registros de la BD no seleccionados para el CE

¿Cuántos casos habrá en CT? ¿qué porcentaje respecto a n ?

- La probabilidad de que se elija un registro es $1/n$. La probabilidad de que no se elija es $1-1/n$
- Se hacen n extracciones, por tanto la probabilidad de que un ejemplo no sea elegido es

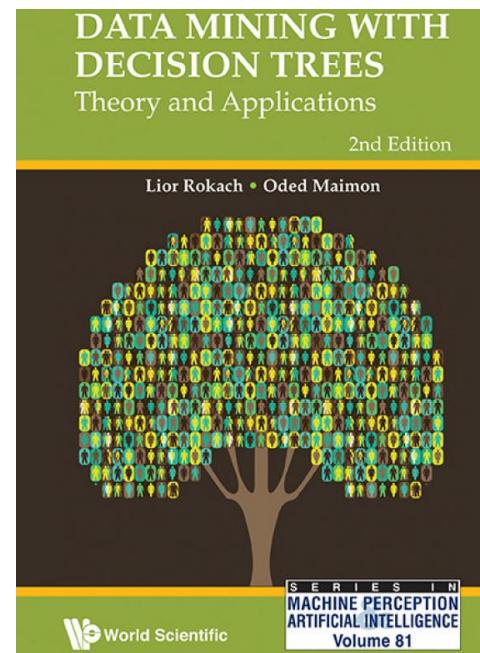
$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

- El CE tendrá aproximadamente el 63.2% de los registros de la BD y el CT el 36.8 %
- Esta técnica se conoce como 0.632 bootstrap
- El error sobre el CT suele ser bastante pesimista por lo que se corrige

$$error = 0.632 \cdot error_{CT} + 0.368 \cdot error_{CE}$$

Clasificación

1. El problema de clasificación
2. **Clasificación con árboles y reglas**
3. Clasificación con otras técnicas
4. Multiclasificadores



Clasificación con árboles y reglas

1. Árboles de decisión

1.1. Definición de árboles de decisión

1.2. Construcción de árboles de decisión

1.3. Criterios de selección de variables

1.4. Particionamiento del espacio con un árbol de decisión

1.5. Ventajas e inconvenientes del uso de árboles de decisión en clasificación

1.6. Algunos algoritmos de minería de datos basados en árboles de decisión

2. Clasificadores basados en reglas

1. Definición de árboles de decisión

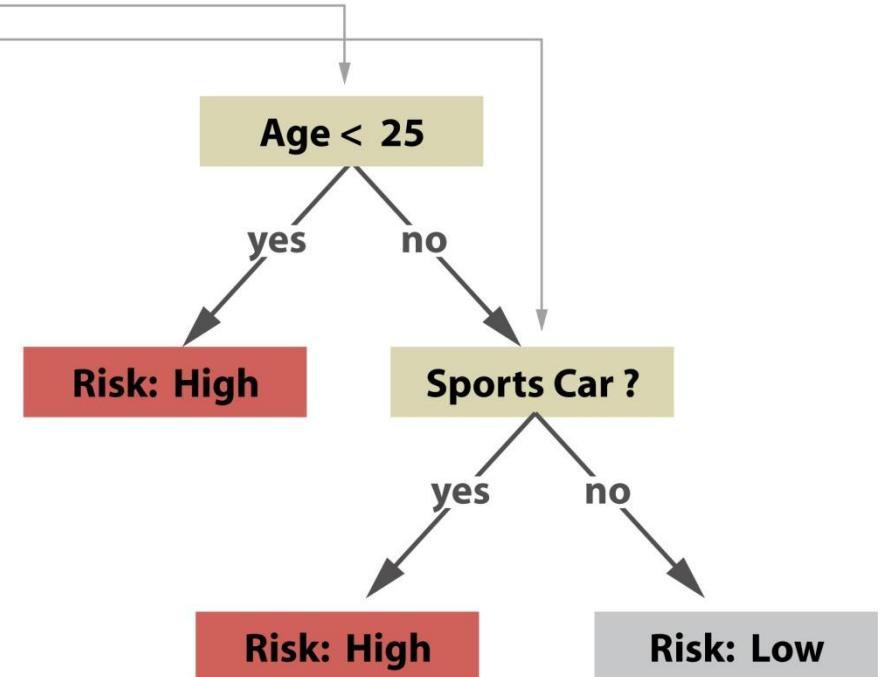
- Un árbol de decisión es un clasificador que en función de un conjunto de atributos permite determinar a que clase pertenece el caso objeto de estudio
- La estructura de un árbol de decisión es:
 - Cada **hoja** es una categoría (clase) de la variable objeto de la clasificación
 - Cada **nodo** es un nodo de decisión que especifica una prueba simple a realizar
 - Los descendientes de cada nodo son los posibles resultados de la prueba del nodo

1. Definición de árboles de decisión



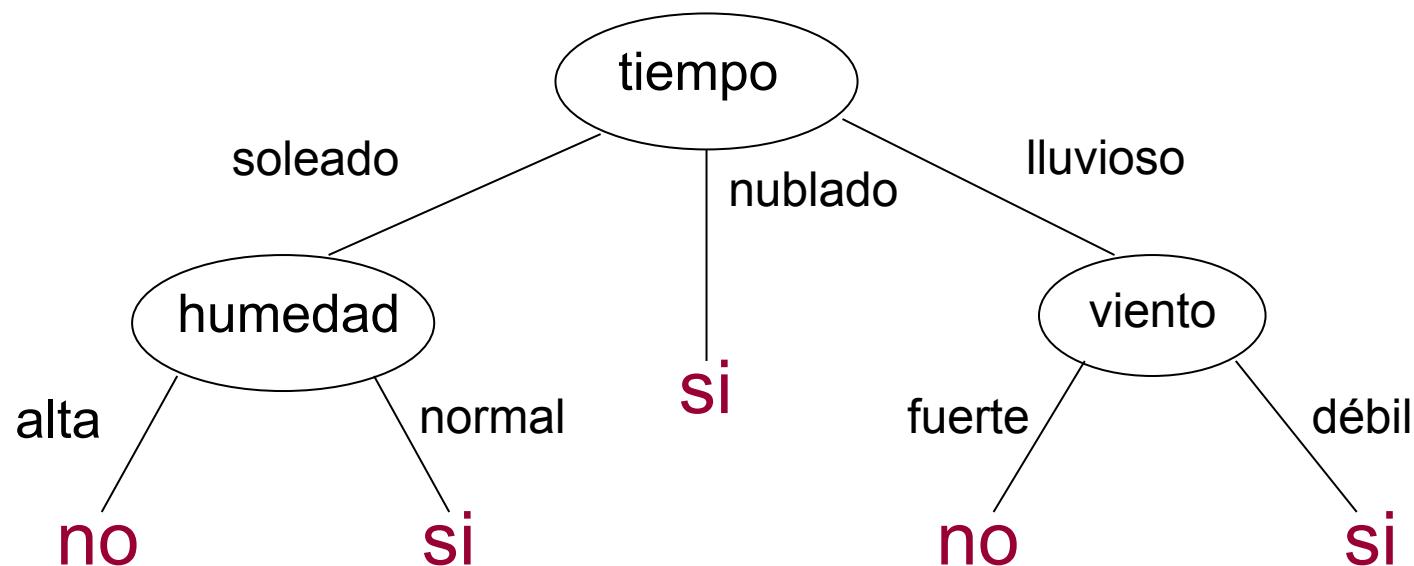
Insurance Risk Assessment

Age	Car Type	Risk
23	family	High
17	sports	High
43	sports	High
68	family	Low
32	truck	Low
20	family	High



1. Definición de árboles de decisión

- **Ejemplo:** Decidir si se puede jugar al tenis dependiendo del tiempo que hace, de la humedad y del viento. Un posible árbol de decisión es



Para clasificar se comienza por la raíz del árbol y se van haciendo las pruebas necesarias hasta llegar a una hoja (clase). P.e. (tiempo=s, viento=d, temperatura=72, humedad=n) → si

1. Definición de árboles de decisión

- Otra opción es generar el conjunto de reglas asociado al árbol y utilizar un sistema basado en reglas (SBR). Del árbol anterior:
 - 1) Si (tiempo soleado) y (humedad alta) entonces (jugar no)
 - 2) Si (tiempo soleado) y (humedad normal) entonces (jugar si)
 - 3) Si (tiempo nublado) entonces (jugar si)
 - 4) Si (tiempo lluvioso) y (viento fuerte) entonces (jugar no)
 - 5) Si (tiempo lluvioso) y (viento débil) entonces (jugar si)

Clasificación con árboles y reglas

1. Árboles de decisión
 - 1.1. Definición de árboles de decisión
 - 1.2. Construcción de árboles de decisión**
 - 1.3. Criterios de selección de variables
 - 1.4. Particionamiento del espacio con un árbol de decisión
 - 1.5. Ventajas e inconvenientes del uso de árboles de decisión en clasificación
 - 1.6. Algunos algoritmos de minería de datos basados en árboles de decisión
2. Clasificadores basados en reglas

2. Construcción de árboles de decisión

- El proceso de generación de un árbol de decisión consiste en dos fases
 - Construcción del árbol
 - Al principio todos los ejemplos de entrenamiento están en el nodo raíz
 - Se van dividiendo recursivamente los ejemplos en base a los atributos seleccionados
 - Poda del árbol
 - Identificar y quitar ramas que describen ruido o datos anómalos
- Uso del árbol de decisión: Clasificar un ejemplo desconocido
 - Comprobar los valores de los atributos del ejemplo contra el árbol de decisión

2. Construcción de árboles de decisión

- **Algoritmo básico (algoritmo voraz)**
 - Se construye el árbol mediante la técnica divide y vencerás aplicada de forma recursiva
 - Al principio todos los ejemplos de entrenamiento están en el nodo raíz
 - Los atributos son categóricos (si son continuos, se discretizan previamente)
 - Los ejemplos se dividen recursivamente basándose en atributos seleccionados
 - Los atributos de test se seleccionan en base a una medida heurística o estadística (por ejemplo, la ganancia de información)
- **Condiciones para terminar el particionamiento**
 - Todos los ejemplos para un nodo dado pertenecen a la misma clase
 - No quedan más atributos para seguir particionando. En este caso se utiliza el voto de la mayoría para clasificar en el nodo hoja
 - NO quedan ejemplos

2. Construcción de árboles de decisión

- Entrada: Sea T el conjunto de ejemplos, $A=\{A_1, \dots, A_n\}$ el conjunto de atributos y $C=\{C_1, \dots, C_k\}$ el conjunto de valores que puede tomar la clase

ConstruirArbol (T, C, A)

1. Crear un nodo RAIZ para el árbol
2. Si todos los ejemplos en T pertenecen a la misma clase C_i , devolver el nodo RAIZ con etiqueta C_i
3. Si $A=\emptyset$ devolver el nodo RAIZ con etiqueta C_i donde C_i es la clase mayoritaria en T
4. $a \leftarrow$ el atributo de A que mejor clasifica T
5. Etiquetar RAIZ con a
6. Para cada valor v_i de a hacer
 1. Añadir una nueva rama debajo de RAIZ con el test $a=v_i$
 2. Sea T_i el subconjunto de T en el que $a = v_i$
 3. ConstruirArbol ($T_i, C, A-a$)
7. Devolver RAIZ

2. Construcción de árboles de decisión

Problema de asignación de crédito

crédito	ingresos	propietario	Gastos-mensuales
N	Bajos	N	Altos
N	Bajos	S	Altos
N	Medios	S	Altos
N	Medios	N	Altos
N	Altos	N	Altos
S	Altos	S	Altos
N	Bajos	N	Bajos
N	Medios	N	Bajos
N	Altos	N	Bajos
S	Medios	S	Bajos

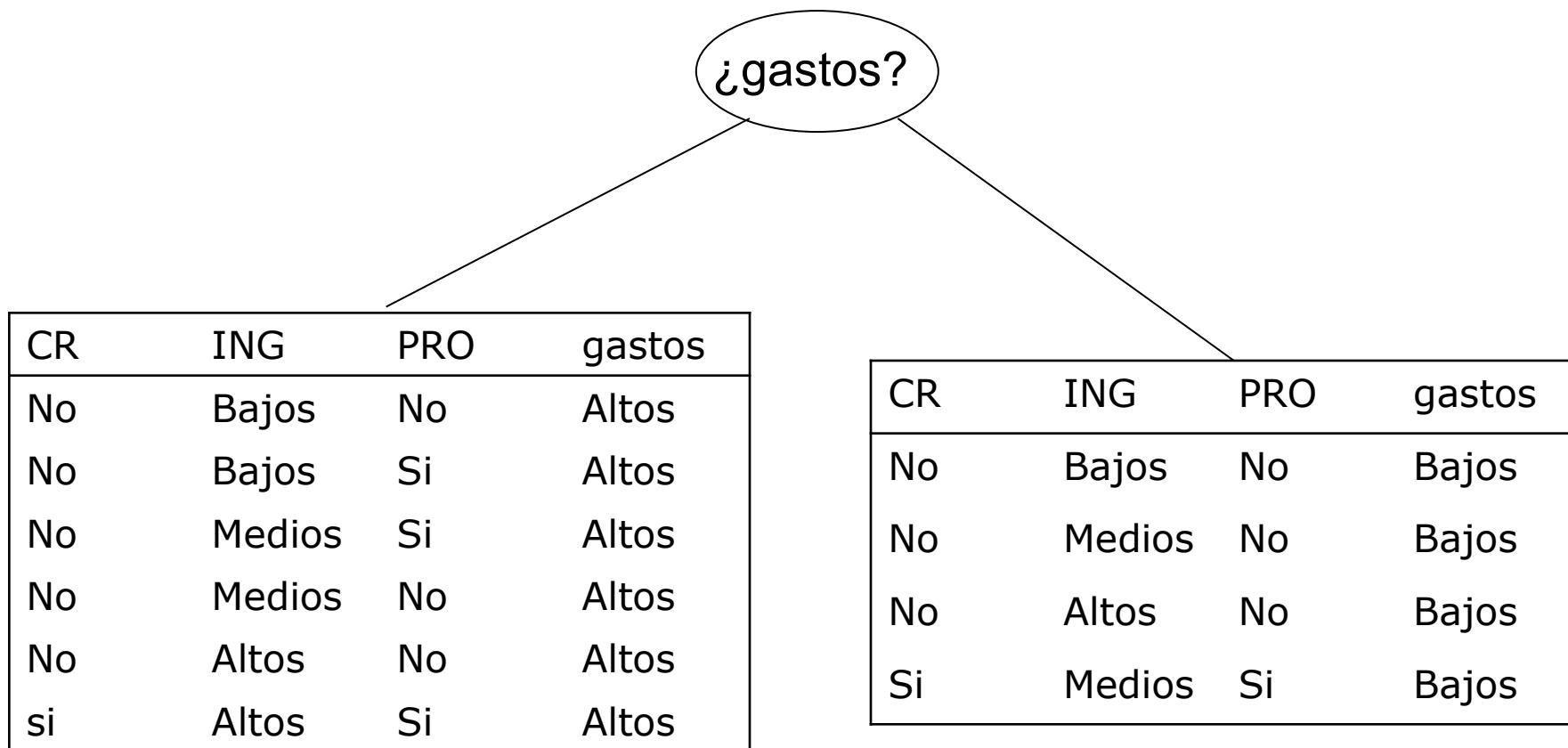
2. Construcción de árboles de decisión

1. Se llama al algoritmo sobre el nodo raiz

crédito	ingresos	propietario	Gastos-mensuales
N	Bajos	N	Altos
N	Bajos	S	Altos
N	Medios	S	Altos
N	Medios	N	Altos
N	Altos	N	Altos
S	Altos	S	Altos
N	Bajos	N	Bajos
N	Medios	N	Bajos
N	Altos	N	Bajos
S	Medios	S	Bajos

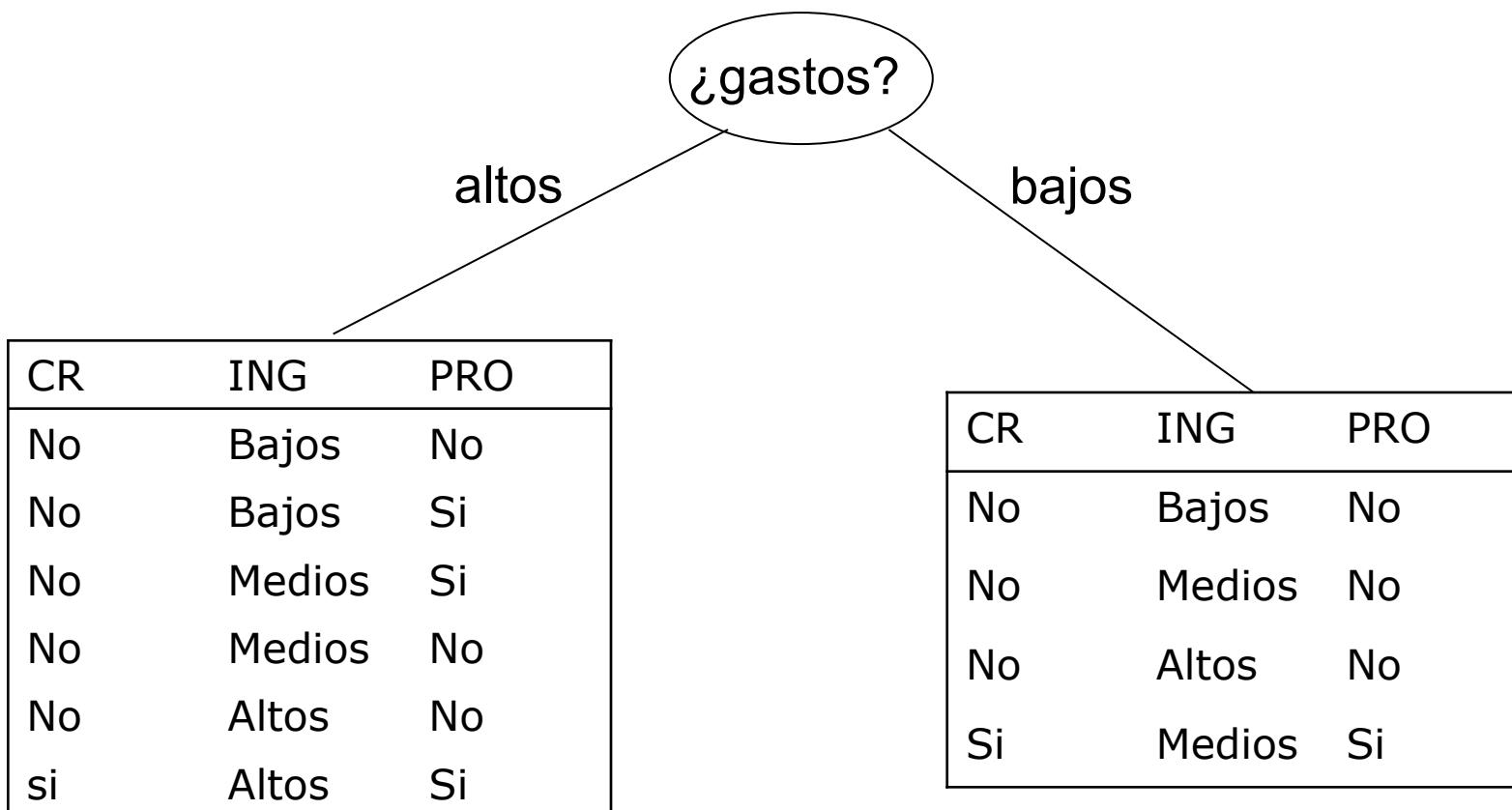
2. Construcción de árboles de decisión

2. Seleccionamos **gastos** como atributo que mejor clasifica



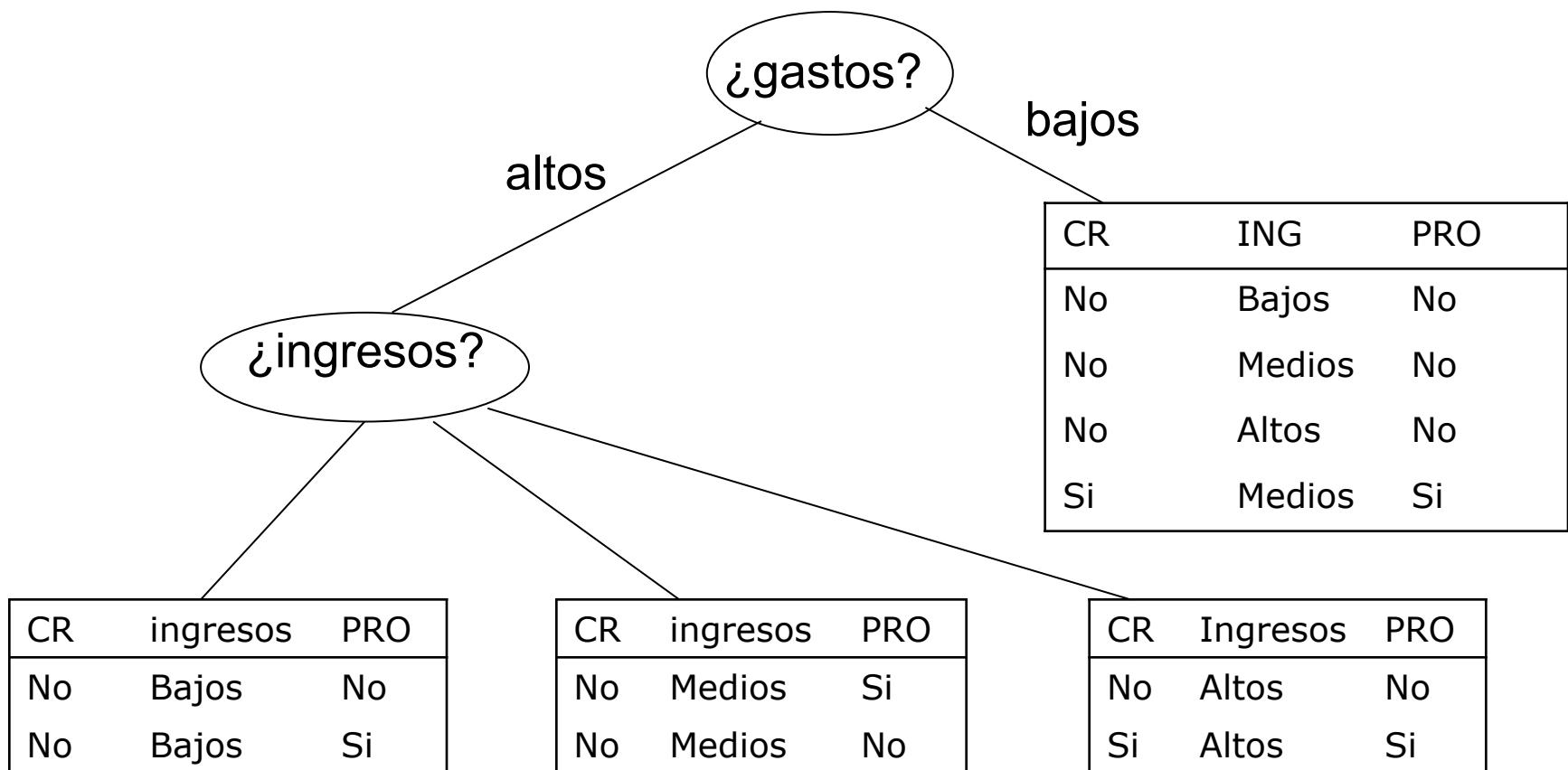
2. Construcción de árboles de decisión

3. Preparamos los nodos para las llamadas recursivas



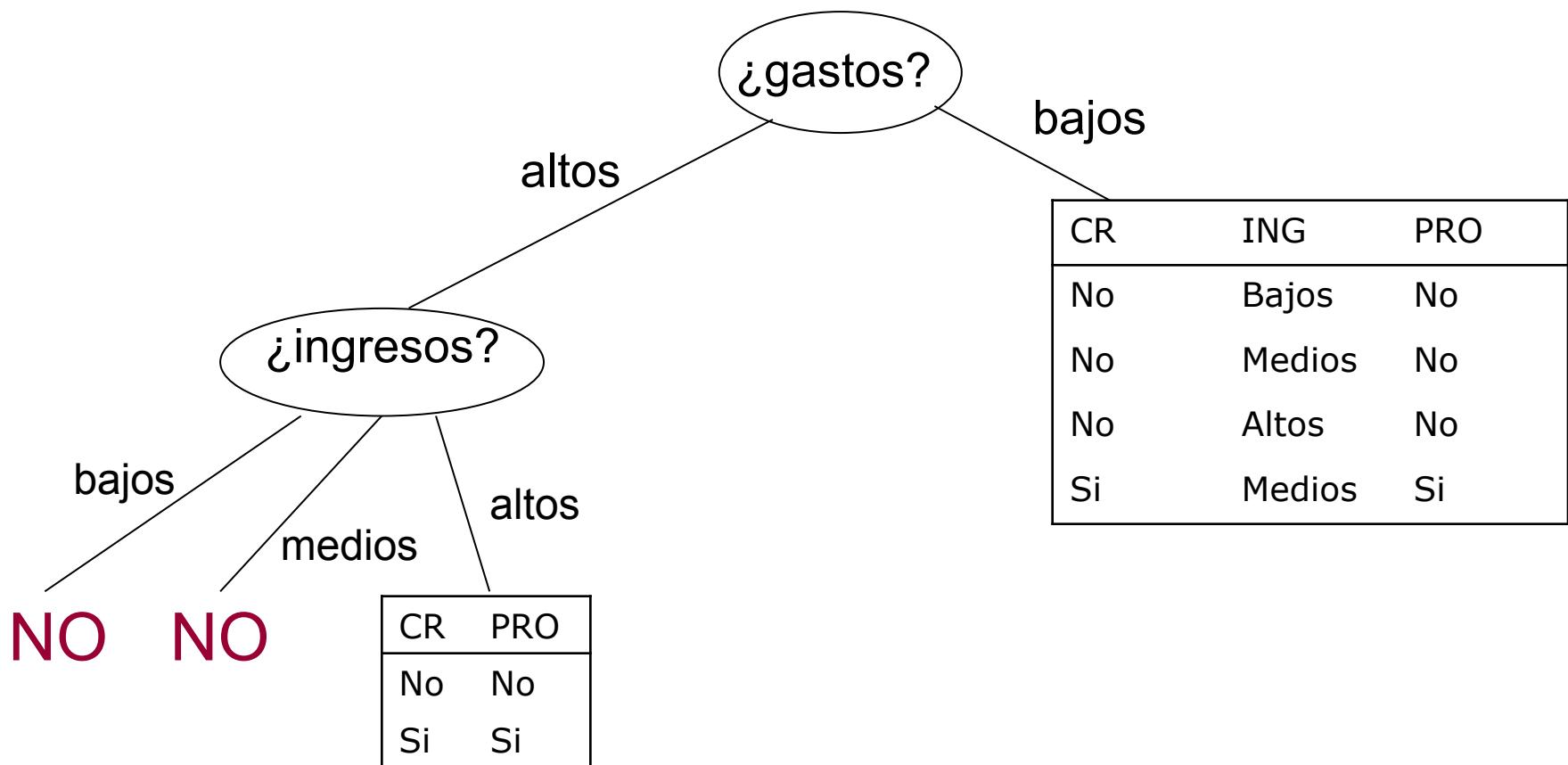
2. Construcción de árboles de decisión

4. Seleccionamos **ingresos** como atributo en gastos = altos



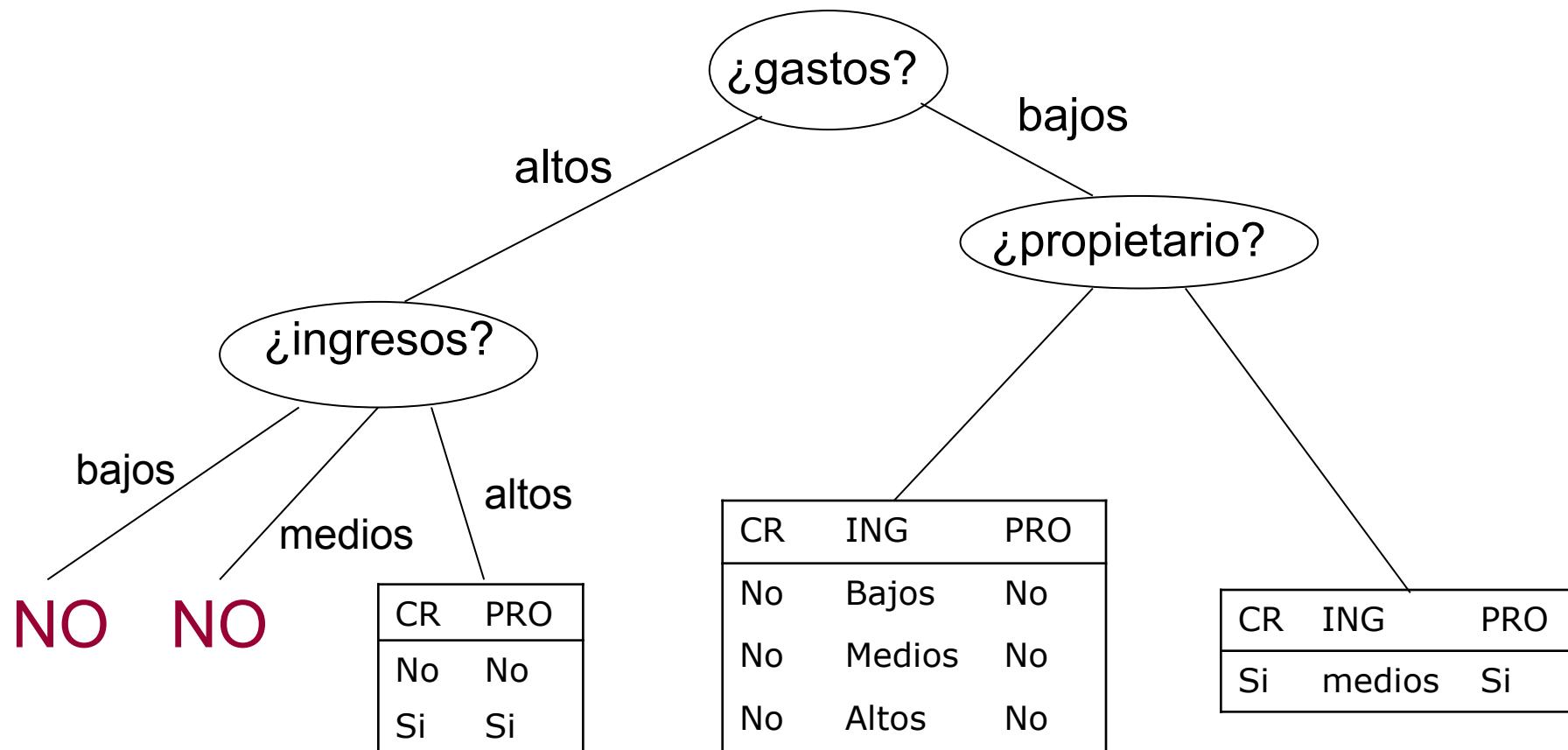
2. Construcción de árboles de decisión

5. Creamos nodos hoja



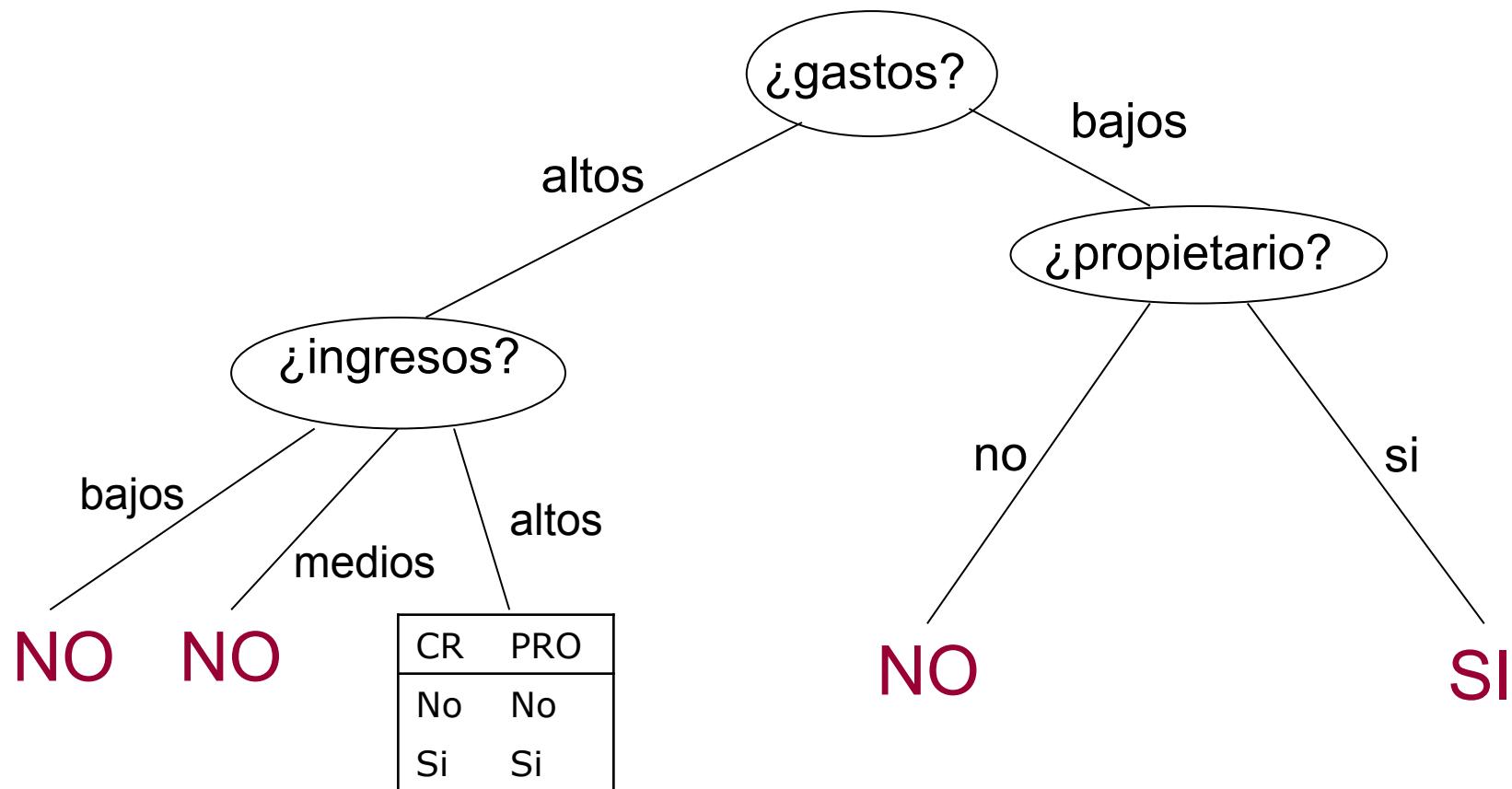
2. Construcción de árboles de decisión

6. Seleccionamos **propietario** como atributo en gastos = bajos



2. Construcción de árboles de decisión

7. Creamos nodos hoja



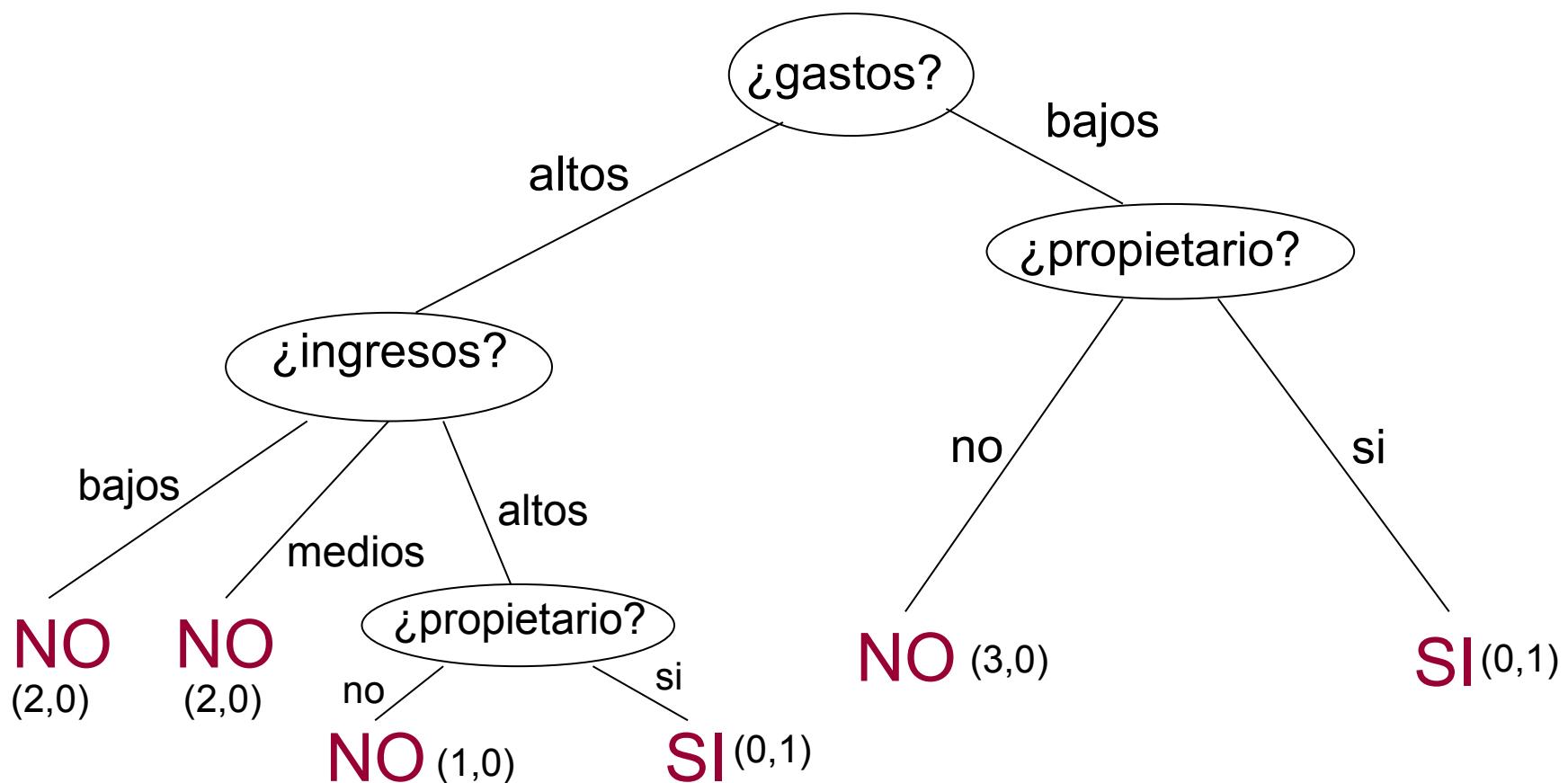
2. Construcción de árboles de decisión

8. Seleccionamos **propietario** como atributo en gastos=altos, ingresos=altos



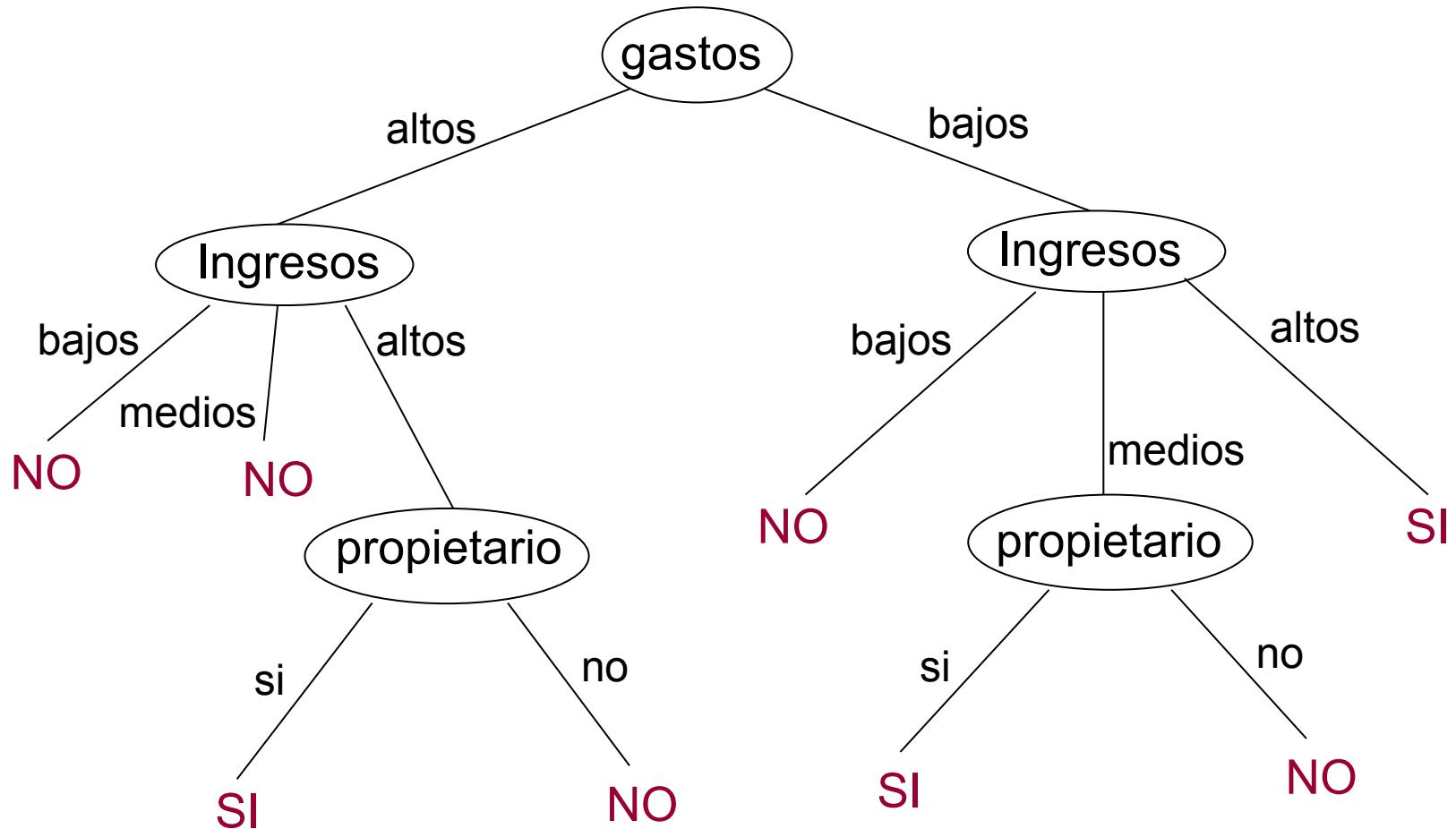
2. Construcción de árboles de decisión

9. Árbol de decisión (#no, #si)



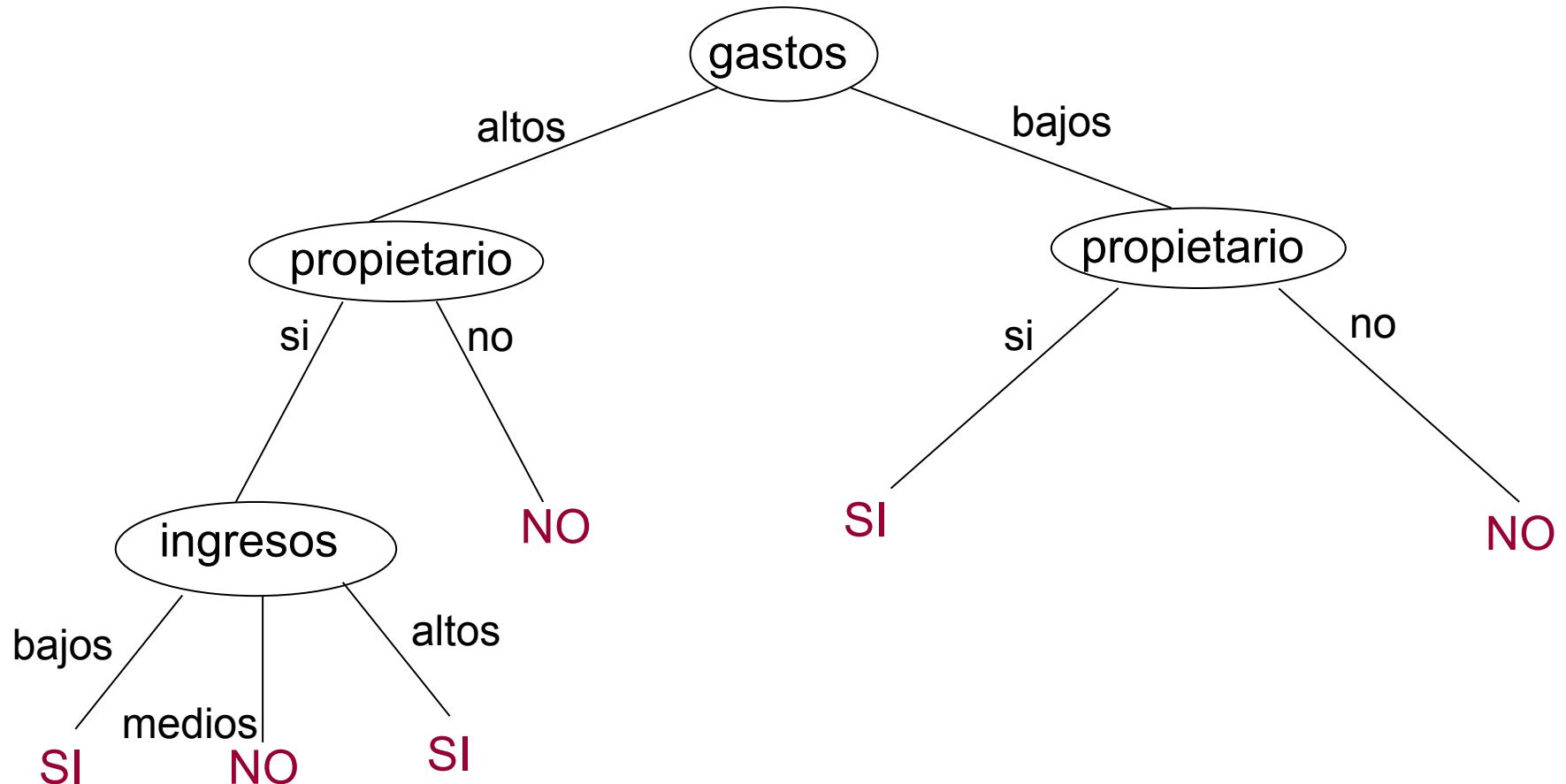
2. Construcción de árboles de decisión

Otro árbol de decisión para crédito (árbol 2)



2. Construcción de árboles de decisión

Otro árbol de decisión para crédito (árbol 3)



2. Construcción de árboles de decisión

- 1er. árbol: 6 reglas (2.33 premisas por regla)
 - 2do. árbol: 8 reglas (2.5 premisas por regla)
 - 3er. árbol: 6 reglas (2.5 premisas por regla)
-
- Dependiendo del orden en el que se van tomando los atributos obtenemos clasificadores de distinta complejidad
 - Lo ideal sería tomar en todo momento el atributo que mejor clasifica

¿Cómo decidir qué atributo es el mejor?

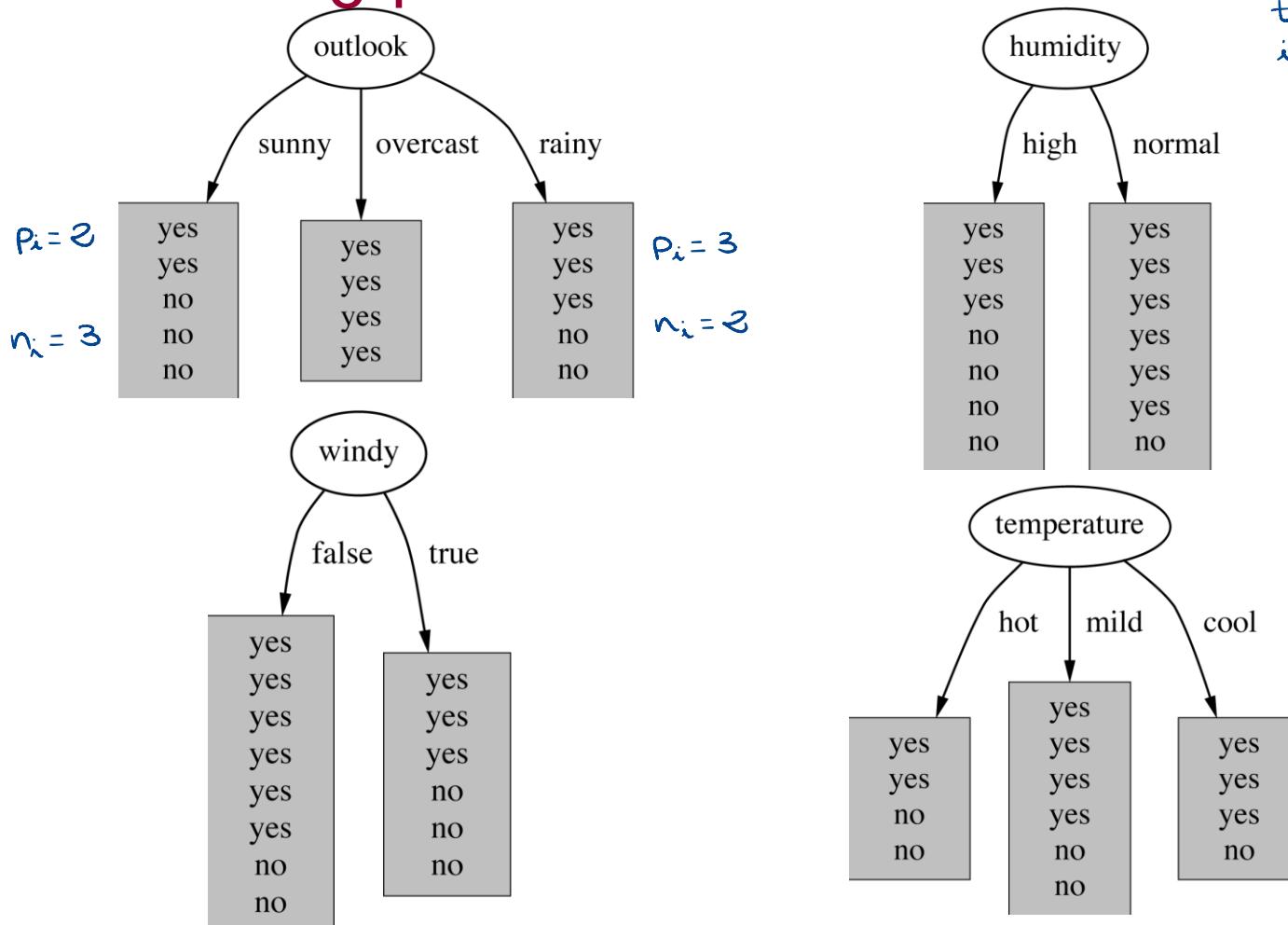
2. Construcción de árboles de decisión

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

2. Construcción de árboles de decisión

¿qué atributo seleccionamos?

Coger el que tenga menos incertidumbre



Clasificación con árboles y reglas

1. Árboles de decisión
 - 1.1. Definición de árboles de decisión
 - 1.2. Construcción de árboles de decisión
 - 1.3. Criterios de selección de variables**
 - 1.4. Particionamiento del espacio con un árbol de decisión
 - 1.5. Ventajas e inconvenientes del uso de árboles de decisión en clasificación
 - 1.6. Algunos algoritmos de minería de datos basados en árboles de decisión
2. Clasificadores basados en reglas

3. Criterios de selección de variables

Existen distintos criterios para seleccionar el atributo X^* de test en cada momento. Algunos de los más conocidos son:

- *InfoGain*: Ganancia de información (ID3) (H es la entropía de la variable)

$$X^* = \max_X (H(C) - H(C|X))$$

$$H(X) = - \sum_i p(x_i) \log_2 p(x_i)$$

- *GainRatio*: Ganancia de información modificada (C4.5)

$$X^* = \max_X \frac{H(C) - H(C|X)}{H(X)}$$

- GINI(CART) $X^* = \max_X (G(C) - G(C|X))$ con $G = 1 - \sum_{i=1}^n p_i^2$

P.e.: $G(\text{gastos}) \approx 0.008$; $G(\text{crédito}) = 0.32$; $G(\text{credito}|\text{gastos}) \approx 0.312$

3. Criterios de selección de variables

Ganancia de información

- Objetivo: Seleccionar el atributo con la mayor ganancia de información
- Consideremos un problema con dos clases: P y N
 - Sea S el conjunto de ejemplos que contiene p elementos de la clase P y n elementos de la clase N
 - La cantidad de información necesaria para decidir si un ejemplo arbitrario dentro de S pertenece a P o a N viene definido por

$$H(C)=H(p,n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

Mayor incertidumbre: (0.5, 0.5)

La entropía mide la aleatoriedad o sorpresa o incertidumbre al predecir una clase

3. Criterios de selección de variables

- Consideremos que utilizando un atributo A , el conjunto S se puede dividir en v conjuntos $\{S_1, S_2, \dots, S_v\}$
 - Si S_i contiene p_i ejemplos de P y n_i ejemplos de N , la entropía o la información que se espera sea necesaria para clasificar los objetos en todos los subárboles S_i es

$$H(C/A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} H(p_i, n_i)$$

- La información que se podría ganar con una rama que considere A es *Te quedas con la variable que te da la mayor ganancia*

$$Gain(A) = H(p, n) - H(C/A)$$

Es la reducción en la entropía al considerar el atributo A

3. Criterios de selección de variables

Conjunto de
entrenamiento

age	income	student	credit_rating
<=30	high	no	fair
<=30	high	no	excellent
31...40	high	no	fair
>40	medium	no	fair
>40	low	yes	fair
>40	low	yes	excellent
31...40	low	yes	excellent
<=30	medium	no	fair
<=30	low	yes	fair
>40	medium	yes	fair
<=30	medium	yes	excellent
31...40	medium	no	excellent
31...40	high	yes	fair
>40	medium	no	excellent

3. Criterios de selección de variables

- Class P: $\text{buys_computer} = \text{"yes"}$

$$H(\text{Clase} / \text{age}) = \frac{5}{14} H(2,3) + \frac{4}{14} H(4,0)$$

- Class N: $\text{buys_computer} = \text{"no"}$

$$+ \frac{5}{14} H(3,2) = 0.69$$

- Cálculo de la entropía para age :

$$\text{Gain}(\text{age}) = H(p,n) - H(\text{Clase} / \text{age})$$

De forma similar

age	p_i	n_i	$I(p_i, n_i)$
≤ 30	2	3	0.971
30...40	4	0	0
>40	3	2	0.971

$$\text{Gain}(\text{income}) = 0.029$$

$$\text{Gain}(\text{student}) = 0.151$$

$$\text{Gain}(\text{credit_rating}) = 0.048$$

3. Criterios de selección de variables

GINI

Ese otro criterio de selección

- Si un conjunto de datos T tiene ejemplos pertenecientes a n clases, el índice gini, se define como

$$gini(T) = 1 - \sum_{j=1}^n p_j^2$$

donde p_j es la frecuencia relativa de la clase j in T

- Si se divide un conjunto de datos T en dos subconjuntos T_1 y T_2 de tamaños N_1 y N_2 respectivamente, el índice *gini* de los datos separados conteniendo ejemplos de las n clases se define como

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

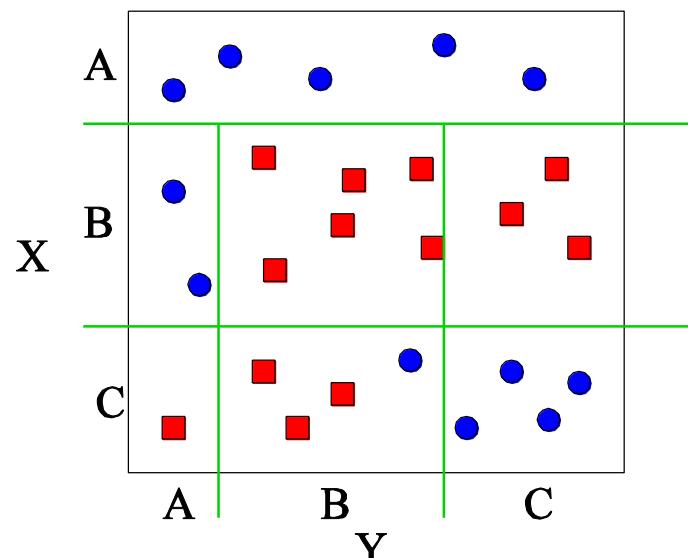
- Se elige para dividir el nodo, el atributo que proporciona el índice $gini_{split}(T)$ más pequeño (es necesario enumerar todos los posibles puntos de división para cada atributo)

Clasificación con árboles y reglas

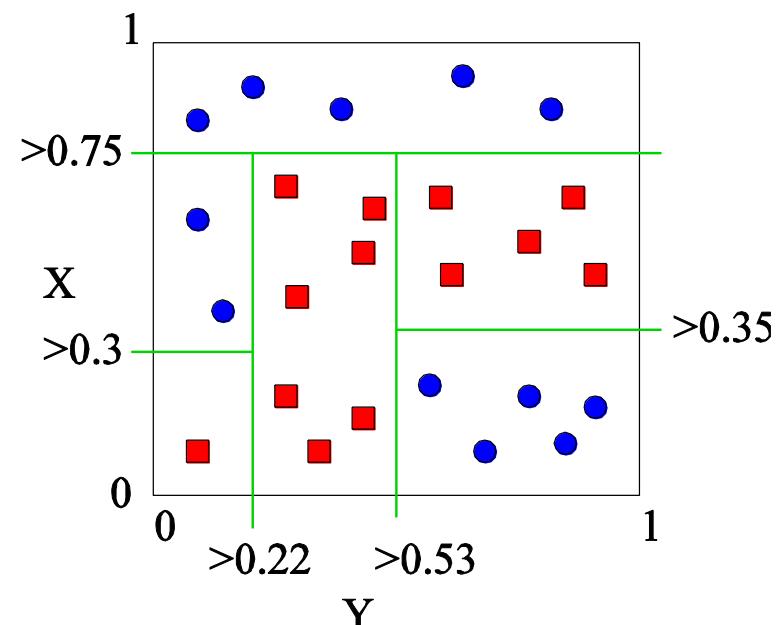
1. Árboles de decisión
 - 1.1. Definición de árboles de decisión
 - 1.2. Construcción de árboles de decisión
 - 1.3. Criterios de selección de variables
 - 1.4. Particionamiento del espacio con un árbol de decisión**
 - 1.5. Ventajas e inconvenientes del uso de árboles de decisión en clasificación
 - 1.6. Algunos algoritmos de minería de datos basados en árboles de decisión
2. Clasificadores basados en reglas

4. Particionamiento del espacio de un árbol de decisión

Los árboles partitionan el espacio de soluciones de forma exhaustiva

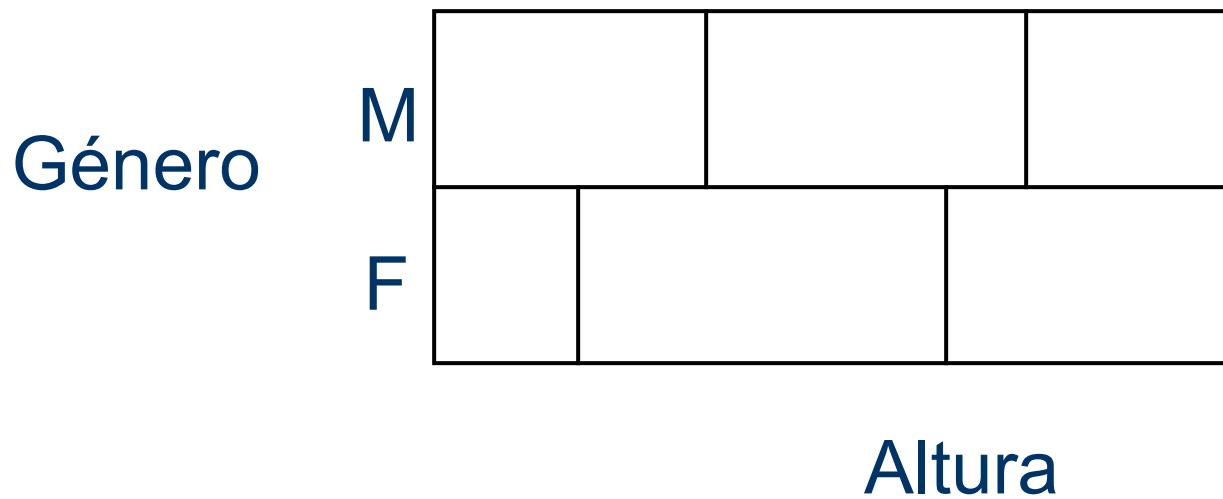


Variables X e Y discretas



Variables X e Y continuas

4. Partitionamiento del espacio de un árbol de decisión



Clasificación con árboles y reglas

1. Árboles de decisión
 - 1.1. Definición de árboles de decisión
 - 1.2. Construcción de árboles de decisión
 - 1.3. Criterios de selección de variables
 - 1.4. Particionamiento del espacio con un árbol de decisión
 - 1.5. Ventajas e inconvenientes del uso de árboles de decisión en clasificación**
 - 1.6. Algunos algoritmos de minería de datos basados en árboles de decisión
2. Clasificadores basados en reglas

5. Ventajas e inconvenientes del uso de arboles de decisión en clasificación

Ventajas:

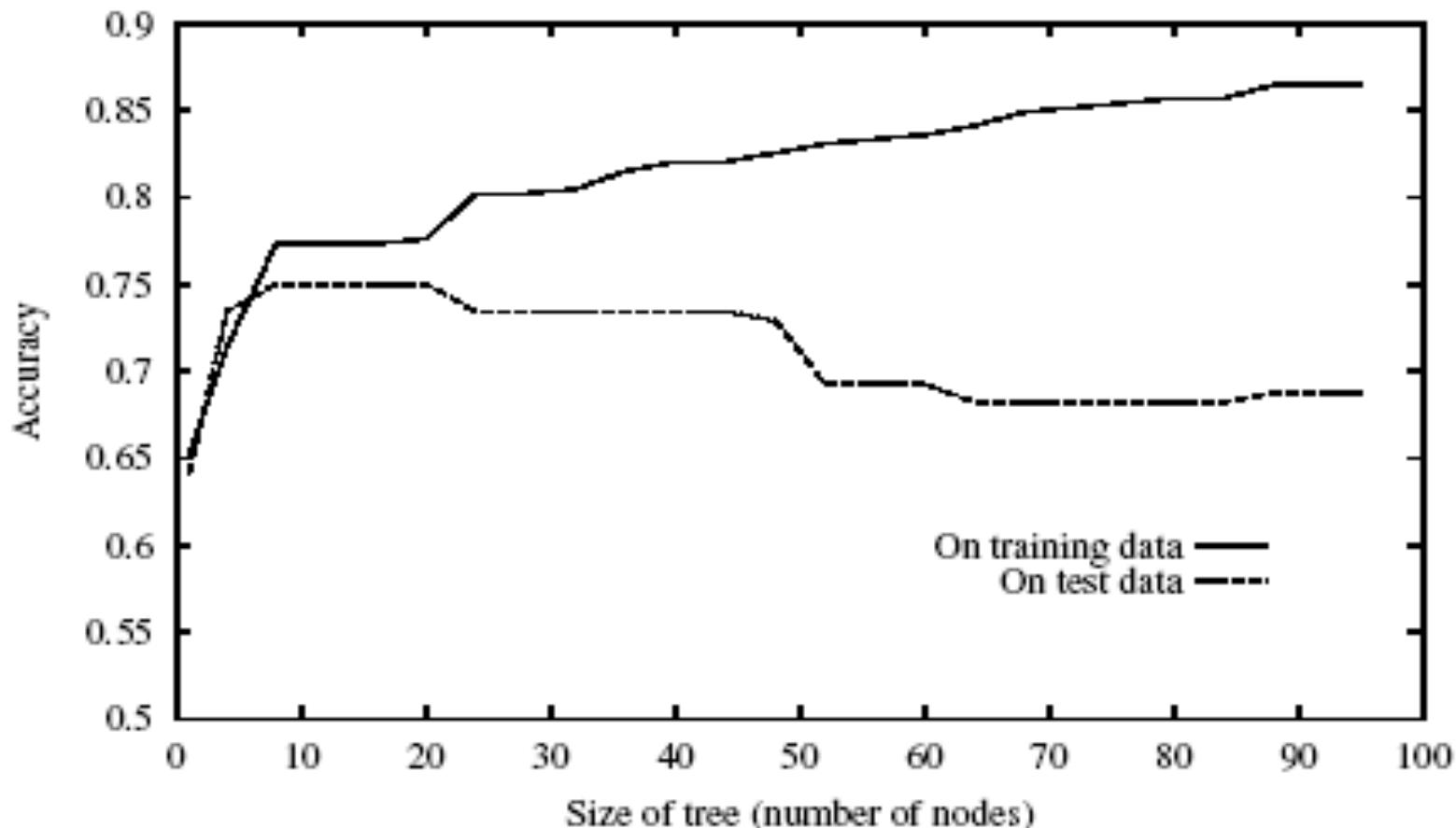
- Los árboles de decisión son fáciles de utilizar y eficientes
- Las reglas que generan son fáciles de interpretar
- Escalan mejor que otros tipos de técnicas
- Tratan bien los datos con ruido

Inconvenientes:

- No manejan de forma fácil los atributos continuos
- Tratan de dividir el dominio de los atributos en regiones rectangulares y no todos los problemas son de ese tipo
- Tienen dificultad para trabajar con valores perdidos
- Pueden tener problemas de sobreaprendizaje
- No detectan correlaciones entre atributos
- No permiten procesar imágenes

→ Para esto se usa deep learning

5. Ventajas e inconvenientes del uso de arboles de decisión en clasificación



Clasificación con árboles y reglas

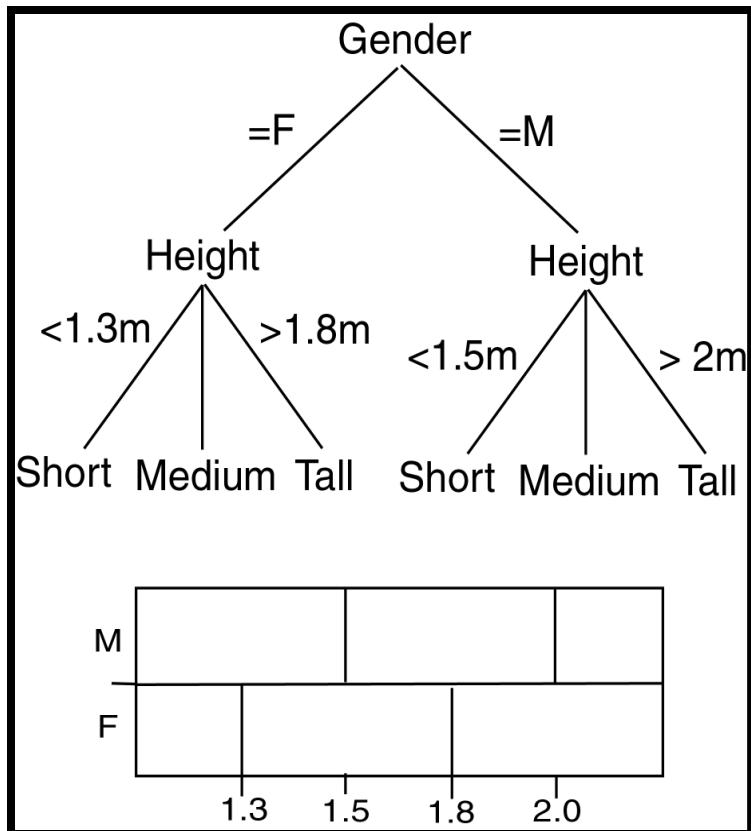
1. Árboles de decisión
 - 1.1. Definición de árboles de decisión
 - 1.2. Construcción de árboles de decisión
 - 1.3. Criterios de selección de variables
 - 1.4. Particionamiento del espacio con un árbol de decisión
 - 1.5. Ventajas e inconvenientes del uso de árboles de decisión en clasificación
- 1.6. Algunos algoritmos de minería de datos basados en árboles de decisión**
2. Clasificadores basados en reglas

6. Algunos algoritmos de clasificación basados en árboles de decisión

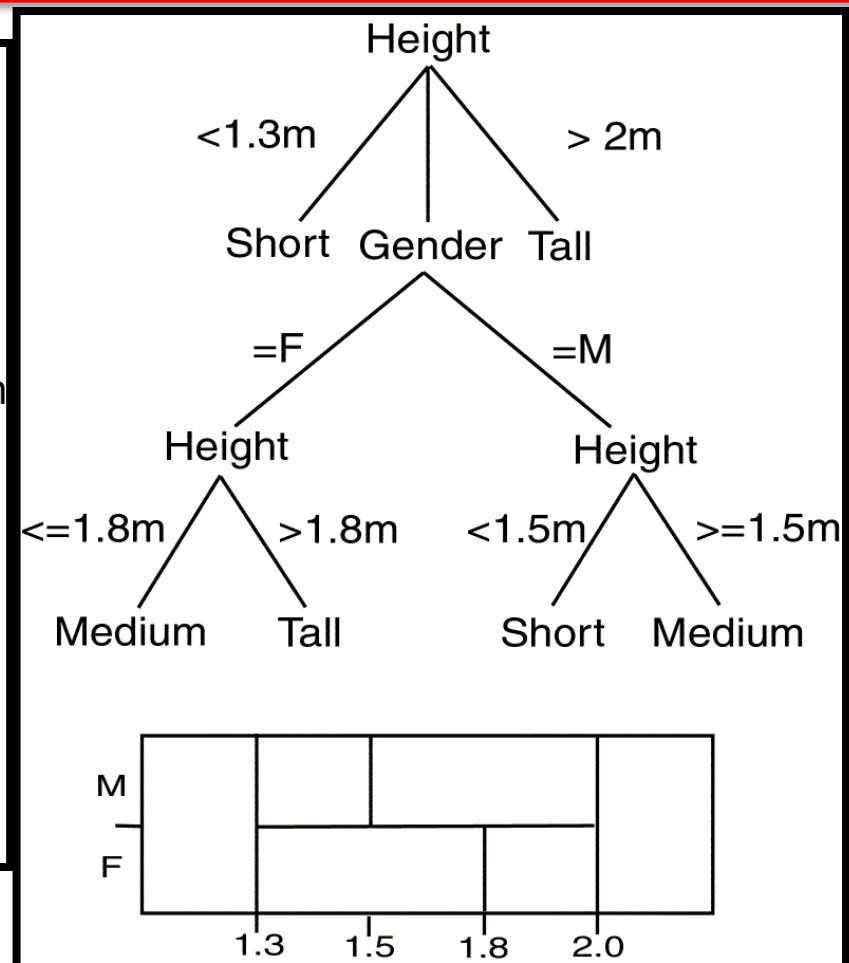
Aspectos importantes en cualquier algoritmo de MD basado en árboles de decisión:

- **Elección de los atributos test:** La forma en que se eligen las variables test tiene gran influencia en el rendimiento del árbol de decisión
 - En ocasiones la elección no sólo implica la revisión de los datos de entrenamiento sino también el uso de la información de expertos
- **Ordenación de los atributos test.** El orden en que se eligen los atributos test es importante para evitar comparaciones innecesarias
- **Divisiones:** Para algunos atributos y dominios las divisiones son obvias mientras que para otros es difícil de determinar
- **Estructura del árbol:** Para mejorar el rendimiento al aplicar el árbol de decisión es preferible un árbol balanceado
- **Criterio de parada.** En muchos problemas puede ser necesario parar antes para prevenir árboles muy grandes. Es un equilibrio entre precisión y rendimiento
- **Ejemplos de entrenamiento.** La estructura del árbol de decisión depende de los ejemplos de entrenamiento
- **Poda:** En ocasiones es necesario realizar un proceso de poda para mejorar el rendimiento del árbol durante la clasificación

6. Algunos algoritmos de clasificación basados en árboles de decisión



Balanceado



Profundidad

6. Algunos algoritmos de clasificación basados en árboles de decisión

ID3

J.R. Quinlan. Induction of Decision Trees. Machine Learning, vol. 1, pp 81-106, 1986

- Crea el árbol utilizando conceptos de teoría de información
- Intenta reducir el número de comparaciones
- ID3 elige el atributo test con máxima ganancia de información
 - Basada en la entropía que se utiliza como una medida de la cantidad de incertidumbre o sorpresa en un conjunto de datos

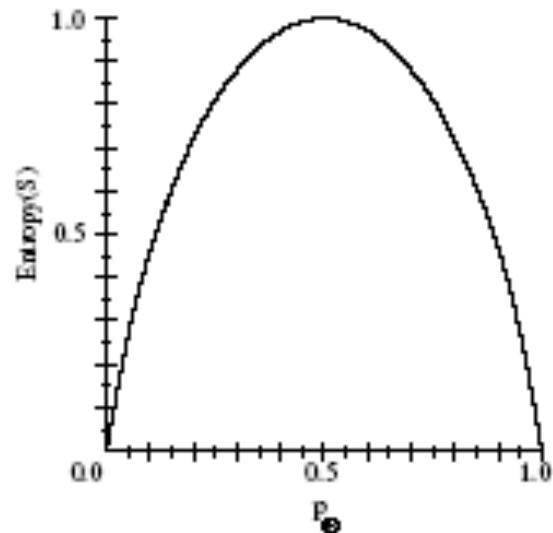
6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

Definición de **entropía**. Dados:

- Un problema con dos clases (positiva y negativa)
- S , un conjunto de ejemplos

La entropía mide la incertidumbre en S

$$\text{Entropía}(S) = H(S) = -p_+ \cdot \log_2 p_+ - p_- \cdot \log_2 p_- = \\ p_+ \cdot \log_2(1/p_+) + p_- \cdot \log_2(1/p_-)$$



- Con k clases:

$$\text{Entropía}(S) = \sum_{i=1}^k p_i \cdot \log_2(1/p_i)$$

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

- ID3 elige el atributo con mayor ganancia de información
 - Ganancia de información: diferencia entre la cantidad de información que se necesita para hacer una clasificación antes de hacer la división y después
 - Se calcula determinando la diferencias entre la entropía del conjunto de datos de partida y la suma ponderada de las entropías una vez dividido el conjunto de ejemplos

$$Gain(S, A) = Entropia(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

Esquema algoritmo ID3

1. Seleccionar el atributo A que maximice la ganancia $G(S,A)$
 2. Crear un nodo para ese atributo con tantos sucesores como valores tenga
 3. Introducir los ejemplos en los sucesores según el valor que tenga el atributo A
 4. Por cada sucesor:
 - Si sólo hay ejemplos de una clase, C_k
Entonces etiquetarlo con C_k
 - Si no, llamar a ID3 con un conjunto de ejemplos formado por los ejemplos de ese nodo, eliminando la columna del atributo A
-
- Termina cuando todos los datos del nodo son de la misma clase y la entropía es cero

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

Ejemplo:

Name	Gender	Height	Output1
Kristina	F	1.6m	Short
Jim	M	2m	Tall
Maggie	F	1.9m	Medium
Martha	F	1.88m	Medium
Stephanie	F	1.7m	Short
Bob	M	1.85m	Medium
Kathy	F	1.6m	Short
Dave	M	1.7m	Short
Worth	M	2.2m	Tall
Steven	M	2.1m	Tall
Debbie	F	1.8m	Medium
Todd	M	1.95m	Medium
Kim	F	1.9m	Medium
Amy	F	1.8m	Medium
Wynette	F	1.75m	Medium

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

1. Inicialmente, la entropía del conjunto es

$$H(S) = \text{Entropia}(S) = 4/15 \log(15/4) + 8/15 \log(15/8) + 3/15 \log(15/3) = 0.4384$$

2. Elegir el atributo test con máxima ganancia de información

Para genero,

$$H(S, \text{gender}=F) = 3/9 \log(9/3) + 6/9 \log(9/6) = 0.2764$$

$$H(S, \text{gender}=M) = 1/6 \log(6/1) + 2/6 \log(6/2) + 3/6 \log(6/3) = 0.4392$$

Name	Gender	Height	Output1
Kristina	F	1.6m	Short
Maggie	F	1.9m	Medium
Martha	F	1.88m	Medium
Stephanie	F	1.7m	Short
Kathy	F	1.6m	Short
Debbie	F	1.8m	Medium
Kim	F	1.9m	Medium
Amy	F	1.8m	Medium
Wynette	F	1.75m	Medium

Name	Gender	Height	Output1
Jim	M	2m	Tall
Bob	M	1.85m	Medium
Dave	M	1.7m	Short
Worth	M	2.2m	Tall
Steven	M	2.1m	Tall
Todd	M	1.95m	Medium

$$\text{Gain}(S, \text{gender}) = 0.4384 - 9/15 \cdot 0.2764 - 6/15 \cdot 0.4392 = 0.09688$$

Para altura, es necesario dividir en rangos: 1:(0,1.6], 2:(1.6,1.7], 3:(1.7,1.8], 4:(1.8,1.9], 5:(1.9,2.0], 6:(2.0,∞)

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

Name	Gender	Height	Output1
Kristina	F	1	Short
Jim	M	5	Tall
Maggie	F	4	Medium
Martha	F	4	Medium
Stephanie	F	2	Short
Bob	M	4	Medium
Kathy	F	1	Short
Dave	M	2	Short
Worth	M	6	Tall
Steven	M	6	Tall
Debbie	F	3	Medium
Todd	M	5	Medium
Kim	F	4	Medium
Amy	F	3	Medium
Wynette	F	3	Medium

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

$$H(S, \text{height}=1) = 2/2 \log(2/2) = 0$$

$$H(S, \text{height}=2) = 2/2 \log(2/2) = 0$$

$$H(S, \text{height}=3) = 3/3 \log(3/3) = 0$$

$$H(S, \text{height}=4) = 4/4 \log(4/4) = 0$$

$$H(S, \text{height}=5) = 1/2 \log 2 + 1/2 \log 2 = 0.301$$

$$H(S, \text{height}=6) = 2/2 \log (2/2) = 0$$

$$\text{Gain}(S, \text{height}) = 0.4384 - 2/15 \cdot 0.301 = 0.3983$$

Es mayor, por lo que se elige el atributo height

Name	Gender	Height	Output1
Kristina	F	1	Short
Kathy	F	1	Short

Name	Gender	Height	Output1
Stephanie	F	2	Short
Dave	M	2	Short

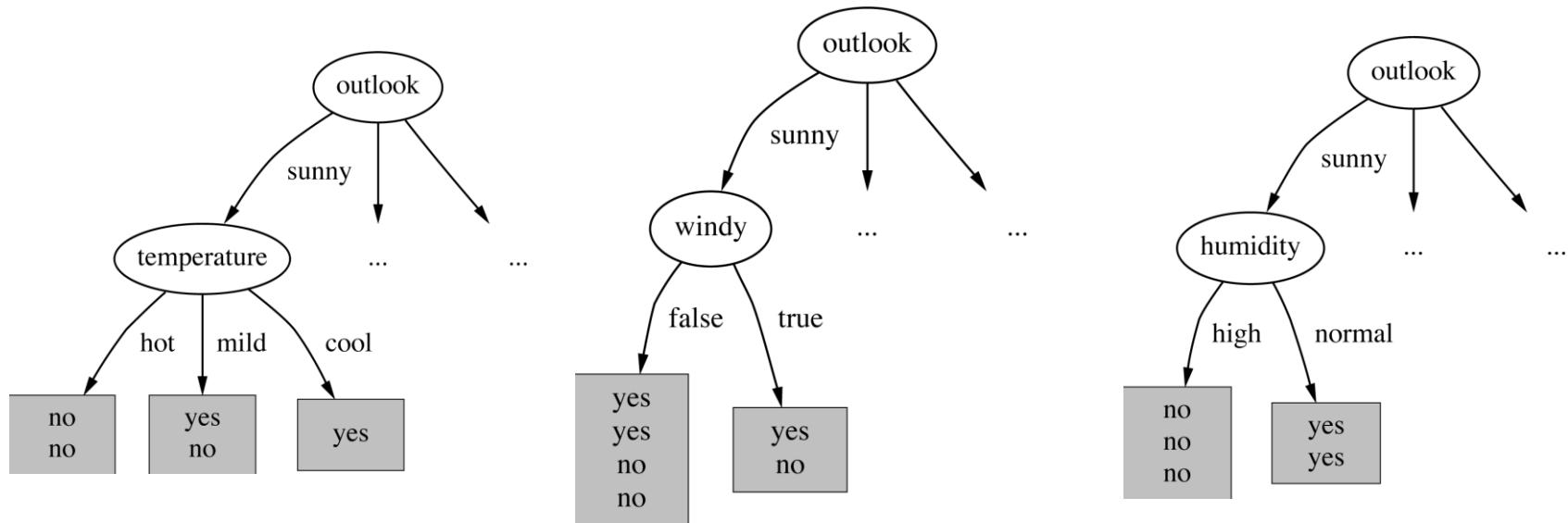
Name	Gender	Height	Output1
Debbie	F	3	Medium
Amy	F	3	Medium
Wynette	F	3	Medium

Name	Gender	Height	Output1
Maggie	F	4	Medium
Martha	F	4	Medium
Bob	M	4	Medium
Kim	F	4	Medium

Name	Gender	Height	Output1
Jim	M	5	Tall
Todd	M	5	Medium

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

Ejemplo 2: Jugar al tenis

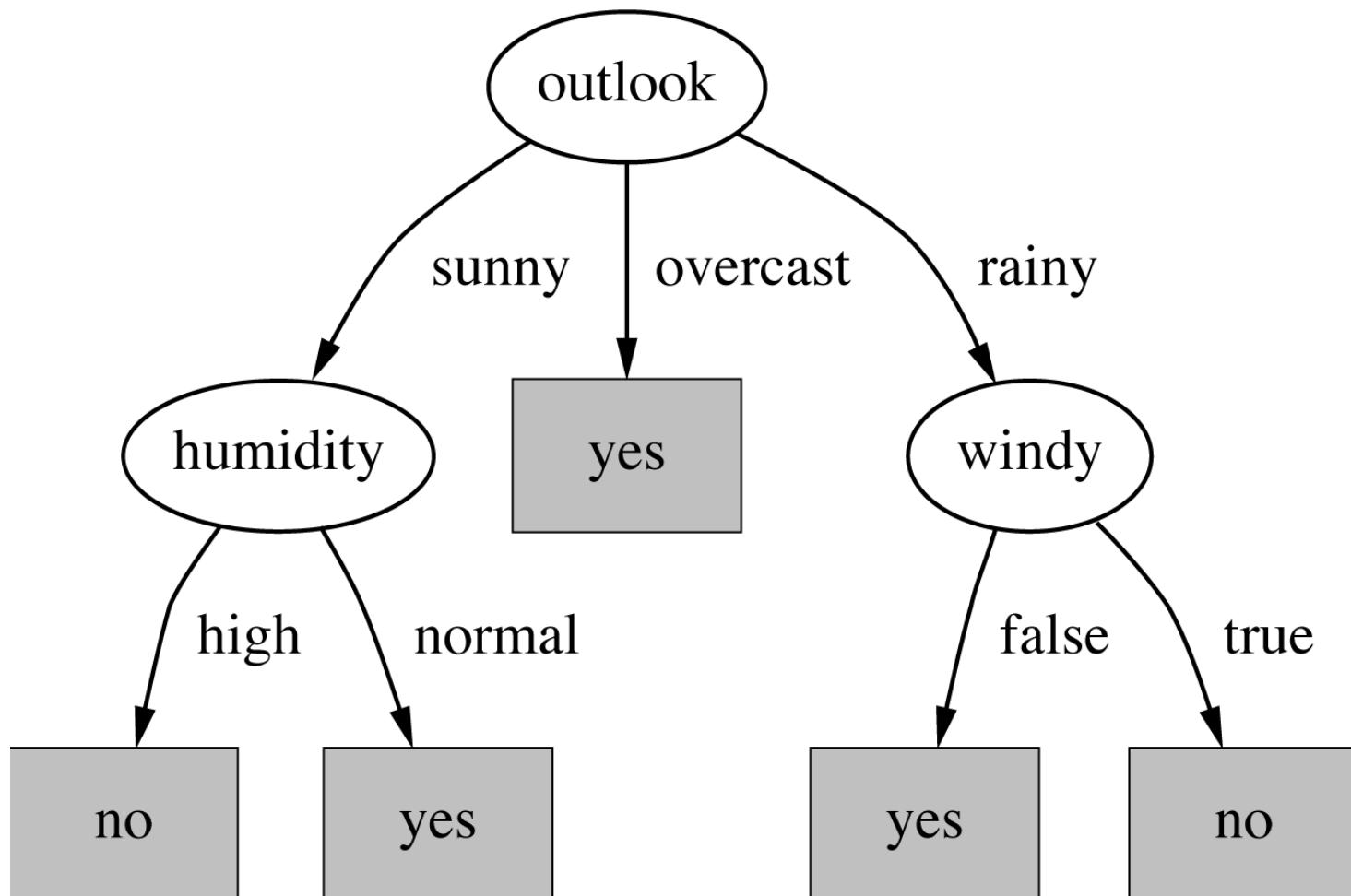


$$\text{gain}(" \text{Humidity}") = 0.971$$

$$\text{gain}(" \text{Temperatur e}") = 0.571$$

$$\text{gain}(" \text{Windy}") = 0.020$$

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3



6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

- El espacio de hipótesis es completo, la función objetivo está incluida en él
- Selecciona una única hipótesis
 - No es capaz de determinar todos los árboles compatibles con los ejemplos de entrenamiento
 - No puede proponer ejemplos que reduzcan el espacio de búsqueda
- No hay vueltas atrás
 - Encuentra un óptimo local que puede no ser el óptimo global (hill-climbing)
- En cada paso utiliza información estadística de todos los ejemplos en lugar de considerar los ejemplos uno a uno
 - Permite ruido en los datos de entrenamiento
- Por la ganancia de información, tiene tendencia a elegir atributos con muchos valores

6. Algunos algoritmos de clasificación basados en árboles de decisión: ID3

De todos los árboles compatibles con los ejemplos de entrenamiento ¿Cuál elige ID3 ?

- Se prefieren árboles cortos frente a largos, con los atributos que producen una mayor ganancia de información cerca de la raíz

Refinamiento de ID3

- Cuándo parar en la construcción del árbol
- Atributos continuos
- Otras medidas de selección de atributos
- Atributos con coste diferente
- Ejemplos incompletos

6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

C4.5

J.R. Quinlan. C4.5: Programs for Machine Learning. San Francisco: Morgan Kaufmann, 1993

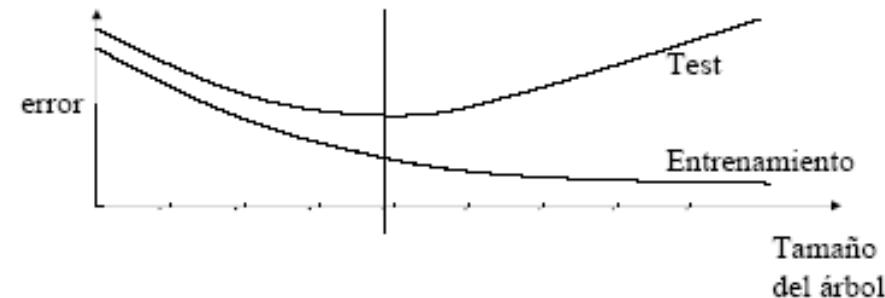
Mejora a ID3 en los siguientes aspectos:

- **Datos perdidos:**
 - Cuando se construye el árbol, los datos perdidos se ignoran (el árbol de decisión se construye mirando sólo los registros que tienen valor para ese atributo)
 - Para clasificar un ejemplo con valor perdido, éste se predice en base a lo que se sabe sobre los valores del atributo para otros registros
- **Datos continuos:** Se divide en rangos en base a los valores encontrados en el conjunto de entrenamiento
- Propone soluciones para el sobreaprendizaje. Posibilidades
 - pre-poda: se decide cuándo dejar de subdividir el árbol
 - post-poda: se construye el árbol y después se poda

6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

Pre-poda:

- no se divide un nodo si se tiene poca confianza en él (no es significativa la diferencia de clases), o
- se valida con un conjunto de test independiente y se para cuando la curva del conjunto de test empieza a subir

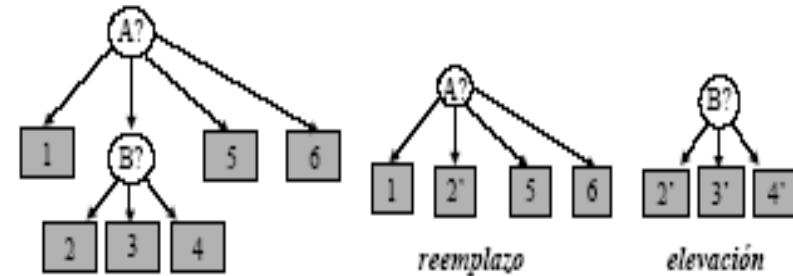


Hay dos estrategias de **post- poda** en C4.5

- Reemplazamiento de subárboles:

Se reemplaza un subárbol por una hoja si al hacerlo el error es similar al original

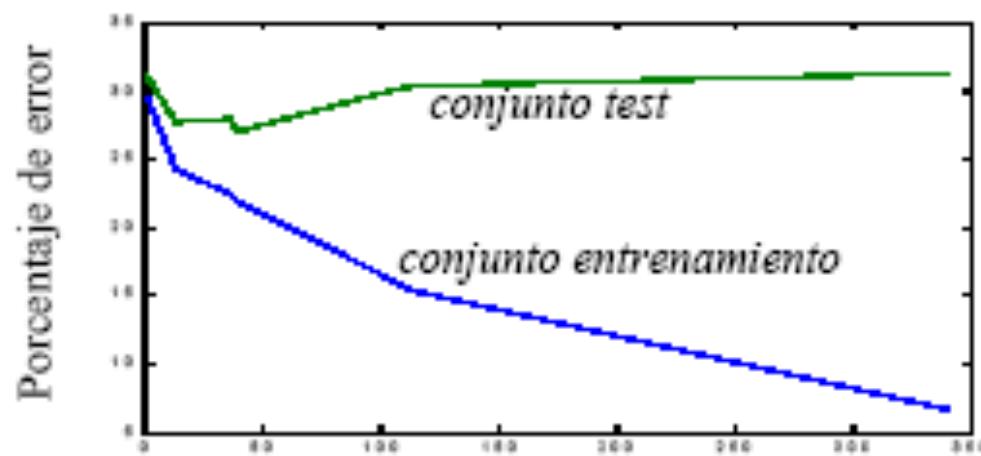
- Elevación de subárbol: Reemplaza un subárbol por su subárbol más utilizado (un subárbol se mueve de su localización a un nodo superior en el árbol)



6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

Efecto del tamaño del árbol

Tamaño del árbol	Conjunto de entrenamiento		Conjunto de test	
	instancias incorrectas	porcentaje de error	instancias incorrectas	porcentaje de error
1	207	29.57 %	93	31%
13	170	24.29 %	83	27.67 %
36	157	22.43%	84	28%
39	154	22%	81	27%
95	119	17%	89	29.67%
113	108	15.43%	91	30.3%
340	47	6.71%	94	31.3%



6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

- Reglas: C4.5 permite clasificación mediante el árbol o a través de las reglas que se generan a partir de él
 - Además, incluye técnicas para simplificar las reglas
- Selección de atributos:
 - ID3 favorece atributos con muchas divisiones
 - En C4.5 se utiliza como criterio de selección el ratio de ganancia de información que tiene en cuenta la cardinalidad de cada división

$$GainRatio(S, A) = \frac{Gain(S, A)}{H\left(\frac{|S_1|}{|S|}, \dots, \frac{|S_v|}{|S|}\right)}$$

$v \in Valores\ de\ A$

6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

En el ejemplo anterior:

Para evaluar *gender*

$$H(9/15, 6/15) = \frac{9}{15} \log \left(\frac{15}{9}\right) + \frac{6}{15} \log \left(\frac{15}{6}\right) = 0.292$$

$$\text{GainRatio}(S, \text{gender}) = \frac{0.09688}{0.292} = 0.332$$

Para evaluar *height*

$$H(2/13, 2/13, 3/13, 4/13, 2/13) = \\ 2/13 \log \left(\frac{13}{2}\right) + 2/13 \log \left(\frac{13}{2}\right) + \\ 3/13 \log \left(\frac{13}{3}\right) + 4/13 \log \left(\frac{13}{4}\right) + \\ 2/13 \log \left(\frac{13}{3}\right)$$

Name	Gender	Height	Output1
Kristina	F	1.6m	Short
Maggie	F	1.9m	Medium
Martha	F	1.88m	Medium
Stephanie	F	1.7m	Short
Kathy	F	1.6m	Short
Debbie	F	1.8m	Medium
Kim	F	1.9m	Medium
Amy	F	1.8m	Medium
Wynette	F	1.75m	Medium

Name	Gender	Height	Output1
Jim	M	2m	Tall
Bob	M	1.85m	Medium
Dave	M	1.7m	Short
Worth	M	2.2m	Tall
Steven	M	2.1m	Tall
Todd	M	1.95m	Medium

Name	Gender	Height	Output1
Kristina	F	1	Short
Kathy	F	1	Short

Name	Gender	Height	Output1
Stephanie	F	2	Short
Dave	M	2	Short

Name	Gender	Height	Output1
Debbie	F	3	Medium
Amy	F	3	Medium
Wynette	F	3	Medium

Name	Gender	Height	Output1
Maggie	F	4	Medium
Martha	F	4	Medium
Bob	M	4	Medium
Kim	F	4	Medium

Name	Gender	Height	Output1
Jim	M	5	Tall
Todd	M	5	Medium

6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

- Ejemplo de poda por número de ejemplos. Árbol original

```
tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
| astigmatism = no
| | age = young: soft (2.0)
| | age = pre-presbyopic: soft (2.0)
| | age = presbyopic
| | | spectacle-prescrip = myope: none (1.0)
| | | spectacle-prescrip = hypermetrope: soft (1.0)
| astigmatism = yes
| | spectacle-prescrip = myope: hard (3.0)
| | spectacle-prescrip = hypermetrope
| | | age = young: hard (1.0)
| | | age = pre-presbyopic: none (1.0)
| | | age = presbyopic: none (1.0)
```

6. Algunos algoritmos de clasificación basados en árboles de decisión: C4.5

2 ejemplos por hoja

tear-prod-rate = reduced: none (12.0)

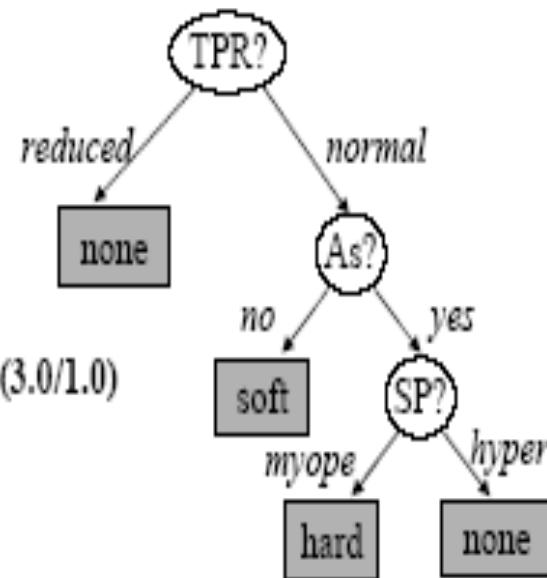
tear-prod-rate = normal

| astigmatism = no: soft (6.0/1.0)

| astigmatism = yes

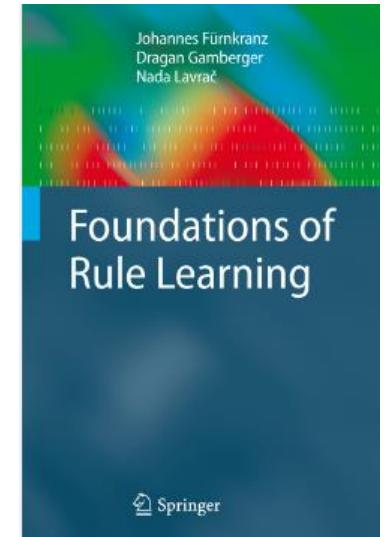
|| spectacle-prescrip = myope: hard (3.0)

|| spectacle-prescrip = hypermetrope: none (3.0/1.0)



Clasificación con árboles y reglas

1. Árboles de decisión
2. **Clasificadores basados en reglas**
 - 2.1. Algoritmos de minería de datos basados en reglas
 - 2.2. Generación de reglas mediante árboles de decisión
 - 2.3. Generación de reglas mediante cobertura



1. Algoritmos de minería de datos basados en reglas

- Una regla de clasificación está formada por
 - un antecedente, que contiene un predicado que se evaluará como verdadero o falso con respecto a cada ejemplo de la base de ejemplos
 - Un consecuente, que contiene una etiqueta de clase
- Se puede utilizar un árbol de decisión para generar reglas pero no son equivalentes:
 - El árbol tiene implícito un orden en el que se van comprobando los atributos. Con las reglas, en general, no hay orden
 - El árbol se crea mirando todas las clases. Cuando se genera una regla sólo se examina una clase
- Líneas principales en los algoritmos de minería de datos para la obtención de reglas:
 - Generación de reglas mediante árboles de decisión
 - Generación de reglas mediante cobertura

Clasificación con árboles y reglas

1. Árboles de decisión
2. **Clasificadores basados en reglas**
 - 2.1. Algoritmos de minería de datos basados en reglas
 - 2.2. **Generación de reglas mediante árboles de decisión**
 - 2.3. Generación de reglas mediante cobertura

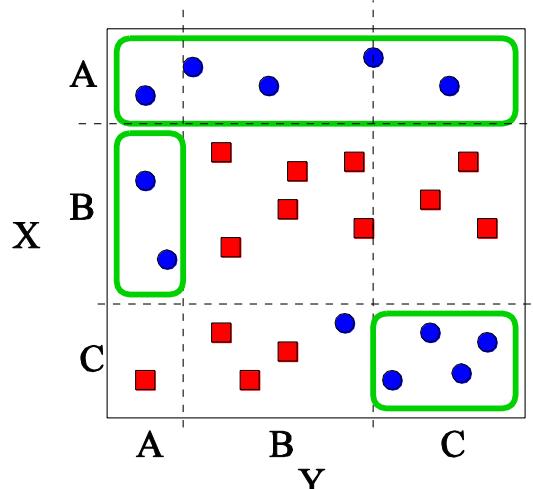
2. Generación de reglas mediante árboles de decisión

- Podemos extraer las reglas de un árbol de decisión y usarlas como un SBR (p.e. para el primer árbol de crédito):
 - SI gastos=altos Y ingresos=bajos ENTONCES NO
 - SI gastos=altos Y ingresos=medios ENTONCES NO
 - Si gastos=altos Y ingresos=altos Y propietario=no ENTONCES NO
 - SI gastos=altos Y ingresos=altos Y propietario=sí ENTONCES SI
 - SI gastos=bajos Y propietario=no ENTONCES NO
 - SI gastos=bajos Y propietario=sí ENTONCES SI
- El algoritmo para obtener el SBR, para cada camino de la raíz del árbol a un nodo hoja, genera una regla y la añade al conjunto final (SBR)

2. Generación de reglas mediante árboles de decisión

Listas de decisión

- Las reglas están ordenadas
 - SI gastos=altos Y ingresos=altos y propietario=sí ENTONCES SI
 - SI gastos=bajos y propietario=sí ENTONCES SI EN OTRO CASO NO
- En estos clasificadores lo que se intenta es buscar las reglas que cubran el mayor número de ejemplos positivos (o el menor de ejemplos negativos)
- No se realiza un particionamiento exhaustivo del espacio de soluciones:



SI $X=A$ ENTONCES círculo
SI $X=B$ e $Y=A$ ENTONCES círculo
Si $X=C$ e $Y=C$ ENTONCES círculo
EN OTRO CASO cuadrado

Clasificación con árboles y reglas

1. Árboles de decisión
 2. **Clasificadores basados en reglas**
 - 2.1. Algoritmos de minería de datos basados en reglas
 - 2.2. Generación de reglas mediante árboles de decisión
- 2.2. Generación de reglas mediante cobertura**

3. Generación de reglas mediante cobertura

- Los algoritmos de cobertura intentan generar reglas que cubran exactamente una clase
- Habitualmente generan la mejor regla posible optimizando la probabilidad de clasificación deseada
- A diferencia de los algoritmos de generación de árboles (seleccionan el mejor atributo), suelen elegir el mejor par (atributo, valor)

Ejemplo: Si se quiere generar una regla para clasificar personas altas, se busca una regla

Si ???? Entonces clase = alto

El objetivo para algoritmos de cobertura es reemplazar ??? por el predicado que obtenga la mejor probabilidad de que la clase sea alto

- Ejemplo básico de algoritmo de cobertura: 1R (genera un conjunto de reglas equivalente a un árbol de decisión de primer nivel). Elige el mejor atributo para realizar la clasificación basándose en datos de entrenamiento
- Otro ejemplo PRISM: genera reglas para cada clase mirando en el conjunto de entrenamiento y añadiendo reglas que describan todos los ejemplos de dicha clase

3. Generación de reglas mediante cobertura

Esquema algoritmo PRISM

$R \leftarrow \emptyset$

Para cada valor c de clase hacer

$E \leftarrow BD$

Mientras E contenga registros con clase c hacer

Regla \leftarrow CrearRegla (antecedentes(\emptyset),consecuente(c))

$RI \leftarrow \emptyset$

Hasta que Regla sea perfecta* hacer

Para todo atributo X no incluido en la Regla hacer

Para todo valor v de X hacer

$RI \leftarrow RI \cup AñadirAntecedente(X=v)$

Regla \leftarrow SeleccionarMejorRegla (RI)**

$EC \leftarrow$ registros cubiertos por Regla

$R \leftarrow R \cup \{\text{Regla}\}$

$E \leftarrow E - EC$

* O no haya más atributos para utilizar

** Selecciona en base a la proporción de ejemplos positivos de dicha regla

3. Ejemplo: algoritmo basado en cobertura

- Aplicación sobre la BD crédito:

Crédito	Ingresos	Propietario	Gastos-mensuales
NO	Bajos	No	Altos
NO	Altos	No	Altos
NO	Bajos	No	Bajos
NO	Medios	No	Bajos
NO	Altos	No	Bajos
NO	Medios	No	Altos
NO	Bajos	Si	Altos
NO	Medios	Si	Altos
SI	Medios	Si	Bajos
SI	Altos	Si	Bajos

- Para seleccionar la mejor regla utilizaremos la exactitud s o el tanto por ciento de acierto

3. Ejemplo: algoritmo basado en cobertura

- Comenzamos con la clase crédito=si regla: $\Phi \rightarrow \text{crédito} = \text{SI}$
- En la primera iteración crearíamos las siguientes reglas en RI

antecedente	consecuente	s
Ingresos=altos	Si	1/3
Ingresos=medios	Si	1/4
Ingresos=bajos	Si	0/3
Propietario=si	Si	2/4
Propietario=no	Si	0/6
Gastos=altos	Si	0/5
Gastos=bajos	Si	2/5

Crédito	Ingresos	Propietario	Gastos-mensuales
NO	Bajos	No	Altos
NO	Altos	No	Altos
NO	Bajos	No	Bajos
NO	Medios	No	Bajos
NO	Altos	No	Bajos
NO	Medios	No	Altos
NO	Bajos	Si	Altos
NO	Medios	Si	Altos
SI	Medios	Si	Bajos
SI	Altos	Si	Bajos

- Seleccionamos la regla propietario=si \rightarrow crédito=SI
Como NO es perfecta seguimos refinando

3. Ejemplo: algoritmo basado en cobertura

- Nuestra regla base ahora es: (regla: propietario=si → crédito=SI)
- Se añaden las siguientes reglas a RI:

antecedente	consecuente	s
propietario=si Y ingresos=altos	Si	1/1
propietario=si Y ingresos=medios	Si	1/2
propietario=si Y ingresos=bajos	Si	0/1
propietario=si Y gastos=altos	Si	0/2
propietario=si Y gastos=bajos	Si	2/2

Crédito	Ingresos	Propietario	Gastos-mensuales
---------	----------	-------------	------------------

NO	Bajos	No	Altos
NO	Altos	No	Altos
NO	Bajos	No	Bajos
NO	Medios	No	Bajos
NO	Altos	No	Bajos
NO	Medios	No	Altos
NO	Bajos	Si	Altos
NO	Medios	Si	Altos
SI	Medios	Si	Bajos
SI	Altos	Si	Bajos

- Seleccionamos la regla
propietario=si Y ingresos=altos
→ crédito=SI
Como es perfecta añadimos la regla a R
- Aún quedan casos con
clase SI sin tratar

Crédito	Ingresos	Propietario	Gastos-mensuales
NO	Bajos	No	Altos
NO	Altos	No	Altos
NO	Bajos	No	Bajos
NO	Medios	No	Bajos
NO	Altos	No	Bajos
NO	Medios	No	Altos
NO	Bajos	Si	Altos
NO	Medios	Si	Altos
SI	Medios	Si	Bajos

3. Ejemplo: algoritmo basado en cobertura

- Procedemos para crédito=SI (regla: $\Phi \rightarrow \text{crédito}=si$)
- En la primera iteración crearíamos las siguientes reglas en RI

antecedente	consecuente	s
Ingresos=altos	Si	0/2
Ingresos=medios	Si	1/4
Ingresos=bajos	Si	0/3
Propietario=si	Si	1/3
Propietario=no	Si	0/6
Gastos=altos	Si	0/5
Gastos=bajos	Si	1/4

Crédito	Ingresos	Propietario	Gastos-mensuales
NO	Bajos	No	Altos
NO	Altos	No	Altos
NO	Bajos	No	Bajos
NO	Medios	No	Bajos
NO	Altos	No	Bajos
NO	Medios	No	Altos
NO	Bajos	Si	Altos
NO	Medios	Si	Altos
SI	Medios	Si	Bajos

- Seleccionamos la regla propietario=si \rightarrow crédito=si
Como NO es perfecta seguimos refinando

3. Ejemplo: algoritmo basado en cobertura

- Regla base: (proprietario=si → crédito=SI)
- Se añaden las siguientes reglas a RI:

Antecedente	consecuente	s
Proprietario=si Y ingresos=altos	Si	0/0
Proprietario=si Y ingresos=medios	Si	1/2
Proprietario=si Y ingresos=bajos	Si	0/1
Proprietario=si Y gastos=altos	Si	0/2
Proprietario=Si Y gastos=bajos	si	1/1

Crédito	Ingresos	Propietario	Gastos-mensuales
NO	Bajos	No	Altos
NO	Altos	No	Altos
NO	Bajos	No	Bajos
NO	Medios	No	Bajos
NO	Altos	No	Bajos
NO	Medios	No	Altos
NO	Bajos	Si	Altos
NO	Medios	Si	Altos
SI	Medios	Si	Bajos

- Seleccionamos la regla propietario=si Y gastos=bajos → crédito = SI. Como es perfecta añadimos la regla a R
- Ya no quedan casos con clase Si, por tanto, pasariamos a la siguiente categoría

3. Ejemplo: algoritmo basado en cobertura

- Si seguimos iterando obtendremos el siguiente conjunto de reglas:

Si propietario = si & ingresos=altos	Crédito=SI
Si propietario = SI & gastos=bajos	Crédito=SI
Si propietario = NO	Crédito=NO
Si ingresos = bajos	Crédito=NO
Si ingresos = medios & gastos = altos	Crédito=NO

- Podríamos aplicar el algoritmo para todas las categorías de la variable clase menos una (la más probable a priori) y dejar ésta como regla por defecto:

Si propietario = si & ingresos=altos	Crédito=SI
Si propietario = SI & gastos=bajos	Crédito=SI
En otro caso	Crédito=NO

3. Ejemplo: algoritmo basado en cobertura

- Genera reglas para todas las categorías (puede modificarse para incluir regla por defecto)
- El conjunto óptimo de reglas puede no obtenerse, debido a que actúa de forma voraz incluyendo una condición en cada paso
- A no ser que haya inconsistencias en la BD consigue una clasificación perfecta → Sobreajusta a los datos

Por ejemplo: si tenemos

- Regla 1: con $s=1/1$, y
- Regla 2: con $s=19/20$

El algoritmo elegiría r1 cuando r2 es mucho más general

Alternativa: utilizar otros criterios en lugar de la exactitud(s) como puede ser la medida F

$$F = \frac{2 \cdot VP}{2 \cdot VP + FP + FN}$$

- ¿Qué ocurre con los atributos numéricos? Condiciones del tipo $\{<, \leq, \geq, >\}$
- Existen algoritmos basados en cobertura/recubrimiento mucho más avanzados que el descrito