



APRENDIZAJE AUTOMÁTICO: TERCER CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

Clasificación: Árboles de Decisión

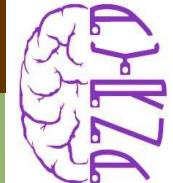
César Hervás-Martínez
Grupo de Investigación AYRNA

Departamento de Informática y Análisis
Numérico
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es

2019-2020

1

1



1 Introducción

2 Algoritmo básico: ID3

3 Mejoras a ID3

4 C4.5

5 Tutorial de Weka

6 Conclusiones



INTRODUCCIÓN

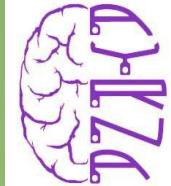


Árbol de decisión

- Muy utilizado y popular
- Aproxima funciones que toman valores discretos.
- La función aprendida se representa como un árbol
- Robusto a datos con ruido
- Aprende expresiones disyuntivas: los árboles aprendidos se pueden representar como reglas if-then (intuitivas)
- Numerosas aplicaciones: diagnósticos médicos, causa de fallo en equipos, evaluación de riesgos de créditos en la concesión de prestamos...



INTRODUCCIÓN



Representación como árboles

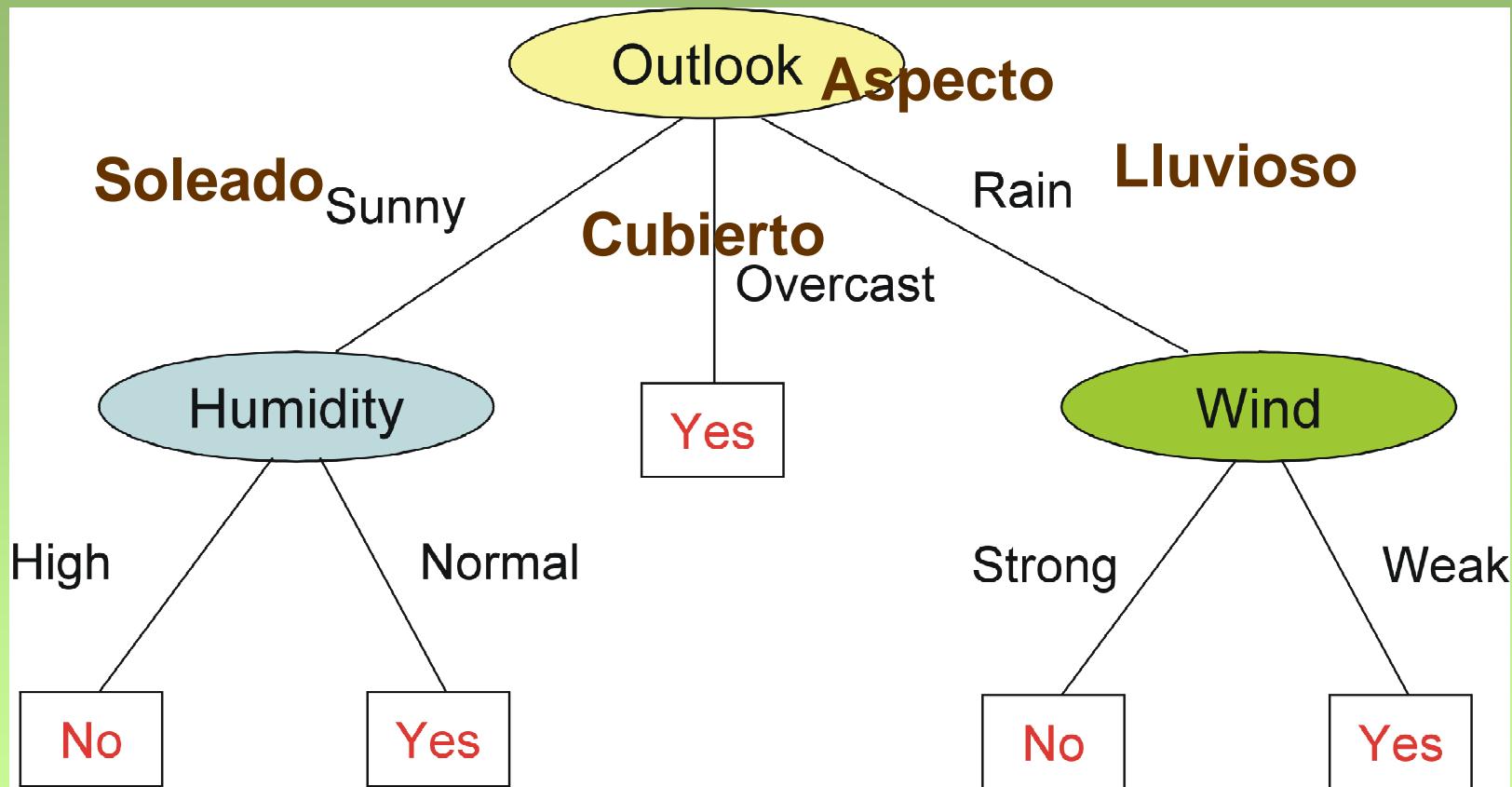
- Cada nodo (no terminal) especifica un test de algún atributo de la instancia.
- Cada rama corresponde a un posible valor del atributo.
- Cada nodo terminal indica la clase en la que se clasifica.
- Las instancias no vistas se clasifican recorriendo el árbol: Pasándoles el modelo test en cada nodo, por orden desde el nodo raíz hasta algún nodo hoja, que da su clasificación.



INTRODUCCIÓN: Ejemplo: Jugar al Tenis



Clasificar las mañanas de sábado en si son o no adecuadas, para el director, para jugar al tenis



Instancia o patrón:

Outlook= Sunny, Humidity= High, Wind= Strong

entra por el camino izdo. y se predice **PlayTennis= No**



INTRODUCCIÓN



Ejemplo: Play Tennis

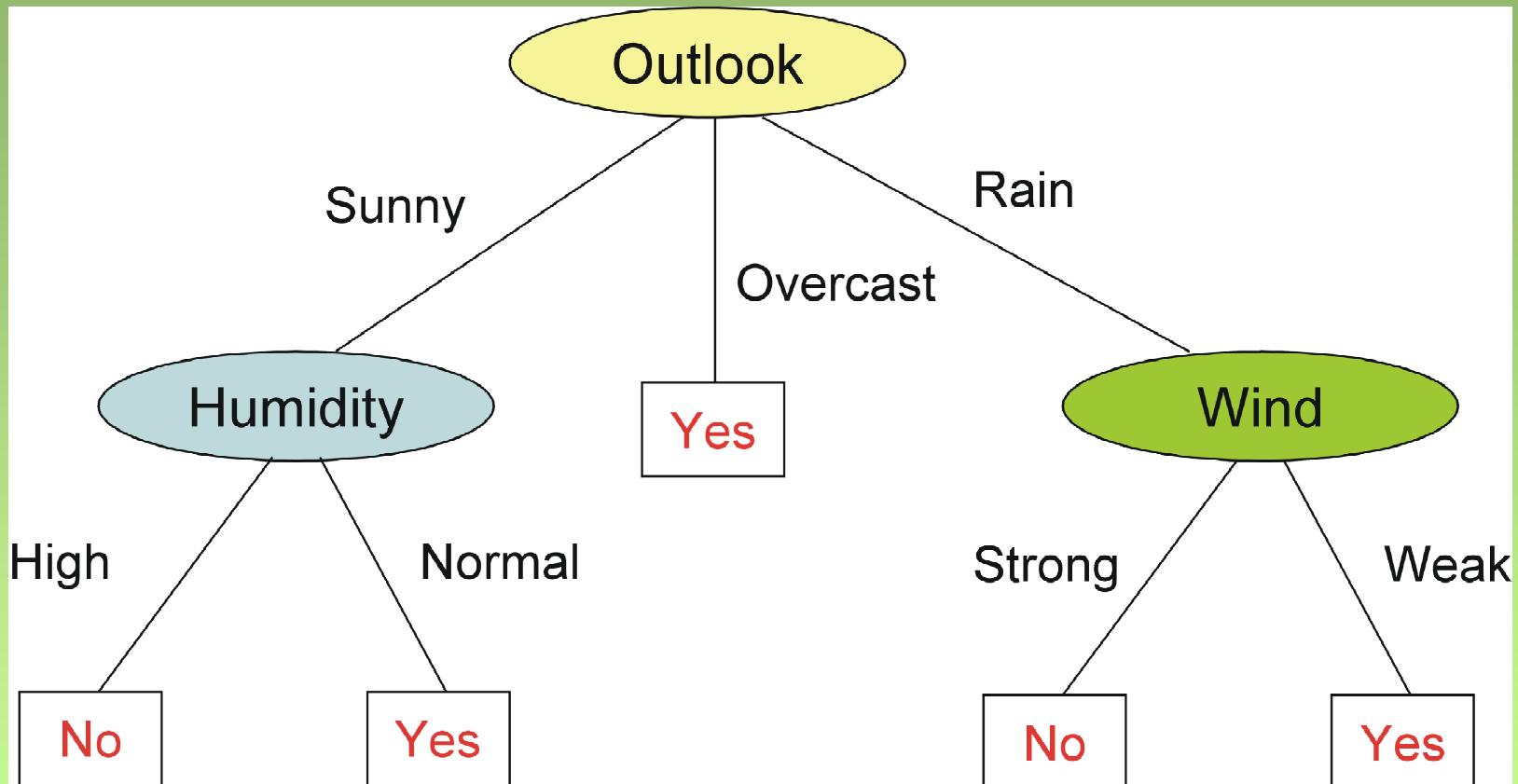
- El árbol representa una disyunción de conjunciones de restricciones sobre los valores de los atributos de las instancias.
- Un camino = una conjunción de test
- Todo el árbol = disyunción de estas conjunciones



INTRODUCCIÓN



Ejemplo: Play Tennis

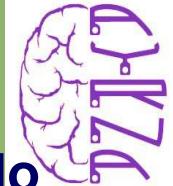


Este árbol para jugar a tenis es:

$$\begin{aligned} & (\text{Outlook}=\text{Sunny} \wedge \text{Humidity}=\text{Normal}) \vee (\text{Outlook}=\text{Overcast}) \\ & \vee (\text{Outlook}=\text{Rain} \wedge \text{Wind}=\text{Weak}) \end{aligned}$$



INTRODUCCIÓN: Generación de reglas



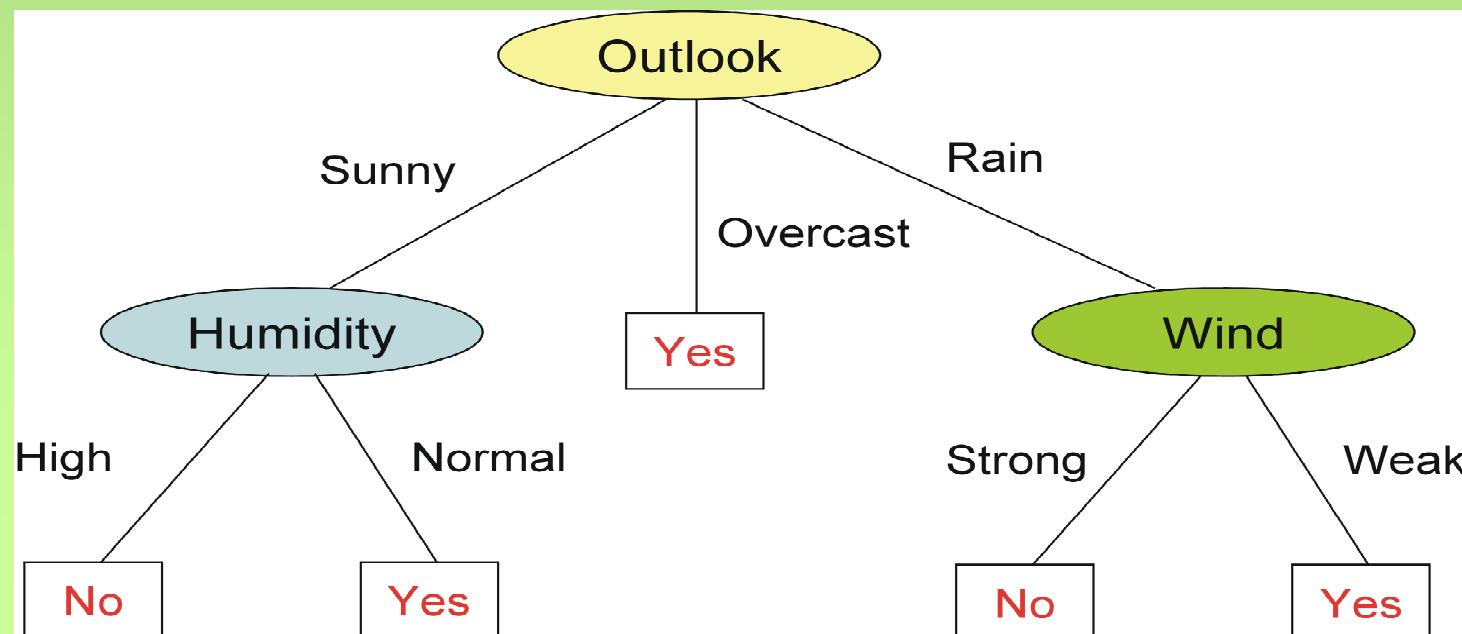
R1: IF (Outlook=Sunny) AND (Humidity=High) Then PlayTennis=No

R2: IF (Outlook=Sunny) AND (Humidity=Normal) then PlayTennis=Yes

R3: IF (Outlook=Overcast) then PlayTennis=Yes

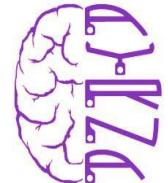
R4: IF (Outlook=Rain) AND (Wind=Strong) then PlayTennis=No

R5: IF (Outlook=Rain) AND (Wind=Weak) then PlayTennis=Yes





INTRODUCCIÓN



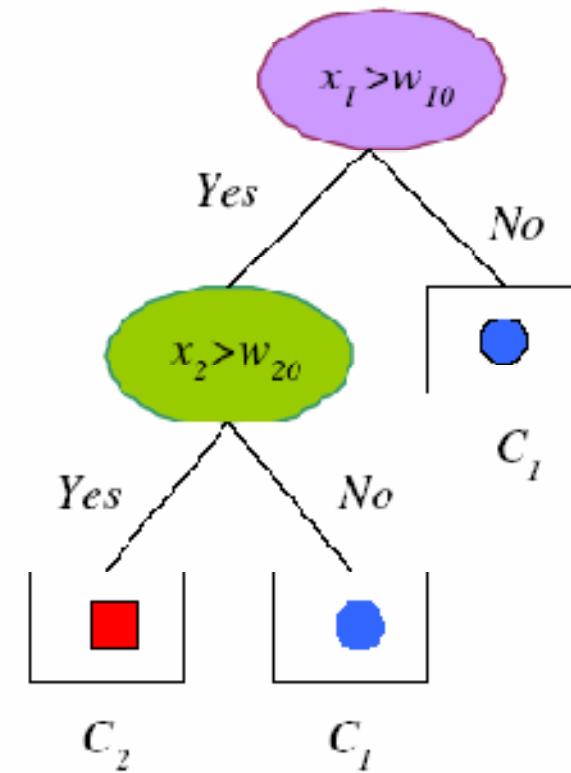
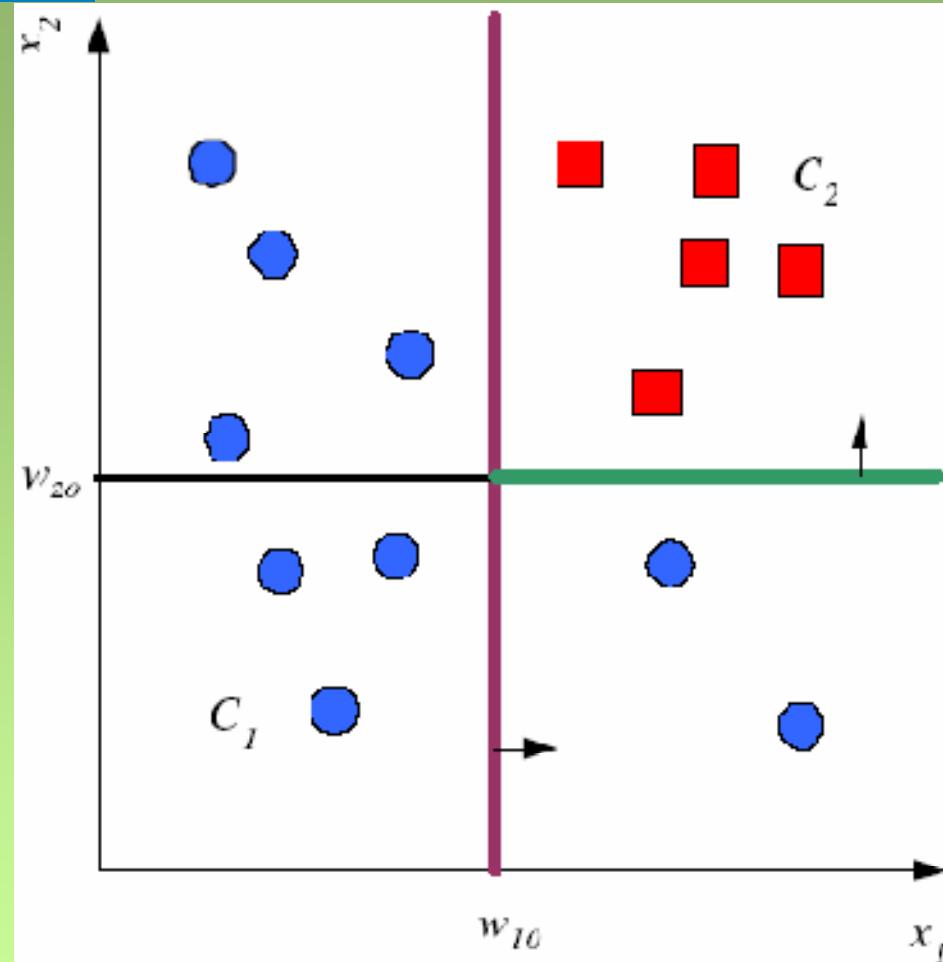
Tipos de problemas apropiados para esta metodología

- Instancias representadas como pares atributo-valor (atributos discretos o reales)
- La función objetivo toma valores discretos (clasificación)...
- ...hay extensiones para valores continuos: árboles de Regresión.
- Robustez a errores en los datos (de entrenamiento), tanto en la variable de respuesta como en los atributos.
- Puede haber datos perdidos (**missing**) para algunas instancias en los valores de los atributos.
- Dominios complejos donde no existe una clara separación lineal



INTRODUCCIÓN

EJEMPLO



Se divide el espacio en regiones etiquetadas con una sola clase y son hiperrectángulos (ojo!)



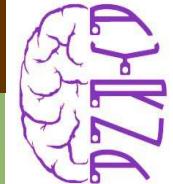
INTRODUCCIÓN



Tipos de árboles

Árboles de clasificación: valores de respuesta discretos **CLS, ID3, C4.5, ID4, ID5, C4.8, C5.0**

Árboles de regresión: valores de respuesta continuos **CART, M5, M5'**



1 Introducción

2 Algoritmo básico: ID3

3 Mejoras a ID3

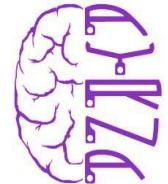
4 C4.5

5 Tutorial de Weka

6 Conclusiones



ID3= Iterative dicotomiser [Quinlan, 1986]



- Basado en el algoritmo CLS (Concept Learning Systems) [Hunt et al., 1966], que usaba solo atributos binarios.
- Estrategia de búsqueda voraz (greedy) por el espacio de posibles árboles de clasificación.
- Construir el árbol de arriba a abajo, preguntando:
¿Que atributo seleccionar como nodo raíz?
- Evaluar cada atributo para determinar lo bien que clasifica los ejemplos el solo.



ID3=Iterative dicotomiser [Quinlan, 1986]



- Seleccionar el mejor como nodo, se abre el árbol para cada posible valor del atributo y los ejemplos se clasifican y colocan en los nodos apropiados
- Repetir todo el proceso usando los ejemplos asociados con el nodo en el que estemos (siempre hacia delante, buscando entre los atributos no usados en este camino)
- Parar cuando el árbol clasifica correctamente los ejemplos o cuando se han usado todos los atributos
- Etiquetar el nodo hoja con la clase de los ejemplos



ID3=Iterative dicotomiser [Quinlan, 1986]



Aprendizaje de árboles de decisión. ID3(Quinlan, 83)

CLS (Hunt, Marin, y Stone, 66) fue el precursor de ID3

- Utilizaba solo atributos binarios
- Tenía heurísticas para decidir que atributo escoger
- Conjunto de técnicas que han tenido mucho éxito comercial.
- Genera árboles de decisión a partir de ejemplos de partida.
- Intenta encontrar el árbol mas sencillo que separa mejor los ejemplos.
- Utiliza la entropía para elegir.



ID3=Iterative dicotomiser [Quinlan, 1986]

ALGORITMO BÁSICO ID3



- **Paso clave: como seleccionar el atributo a testar en cada nodo del árbol?**
- Nos gustaría seleccionar el mas útil para clasificar ejemplos; el que los separe mejor
- **ID3 escoge la Cantidad de Información Mutua, I, como medida de la valía de cada atributo (maximizarla)**
 $I(C, X_i) = H(C) - H(C|X_i)$ (**ganancia de información**)

$$H(C) = -\sum_{c=1}^n p(c) \log_2 p(c)$$

$$H(C | X) = -\sum_{c=1}^n p(x | c) \log_2 p(c | x) = -\sum_c \sum_x p(x, c) \log_2 p(c | x)$$

- **Reducción esperada en entropía (incertidumbre), causada al dividir los ejemplos de acuerdo a este atributo**



EJEMPLO

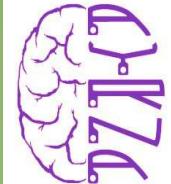


Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



Wind?

EJEMPLO ID3: PLAY TENNIS



$$I(X, Y) = H(X) - H(X | Y) = -\sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

$$H(C) = -\sum_c^n p(c) \log_2 p(c) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0,940$$

$$H(C | X) = -\sum_c \sum_x p(x, c) \log_2 p(c | x) = -P(\text{Strong}, \text{Yes}) \log_2 P(\text{Yes} | \text{Strong}) - P(\text{Strong}, \text{No}) \log_2 P(\text{No} | \text{Strong}) - P(\text{Weak}, \text{Yes}) \log_2 P(\text{Yes} | \text{Weak}) - P(\text{Weak}, \text{No}) \log_2 P(\text{No} | \text{Weak})$$

$$H(C | X) = -\frac{3}{14} \log_2 \frac{3}{6} - \frac{3}{14} \log_2 \frac{3}{6} - \frac{6}{14} \log_2 \frac{6}{8} - \frac{2}{14} \log_2 \frac{2}{8} = 0,892$$

$$I(C, \text{Wind}) = 0,940 - 0,892 = 0,048$$

Análogamente,

$$I(C, \text{Humidity}) = 0,151$$

$$I(C, \text{Outlook}) = 0,246 \leftarrow \text{Escogeremos Outlook como nodo raíz}$$

$$I(C, \text{Temperature}) = 0,029$$

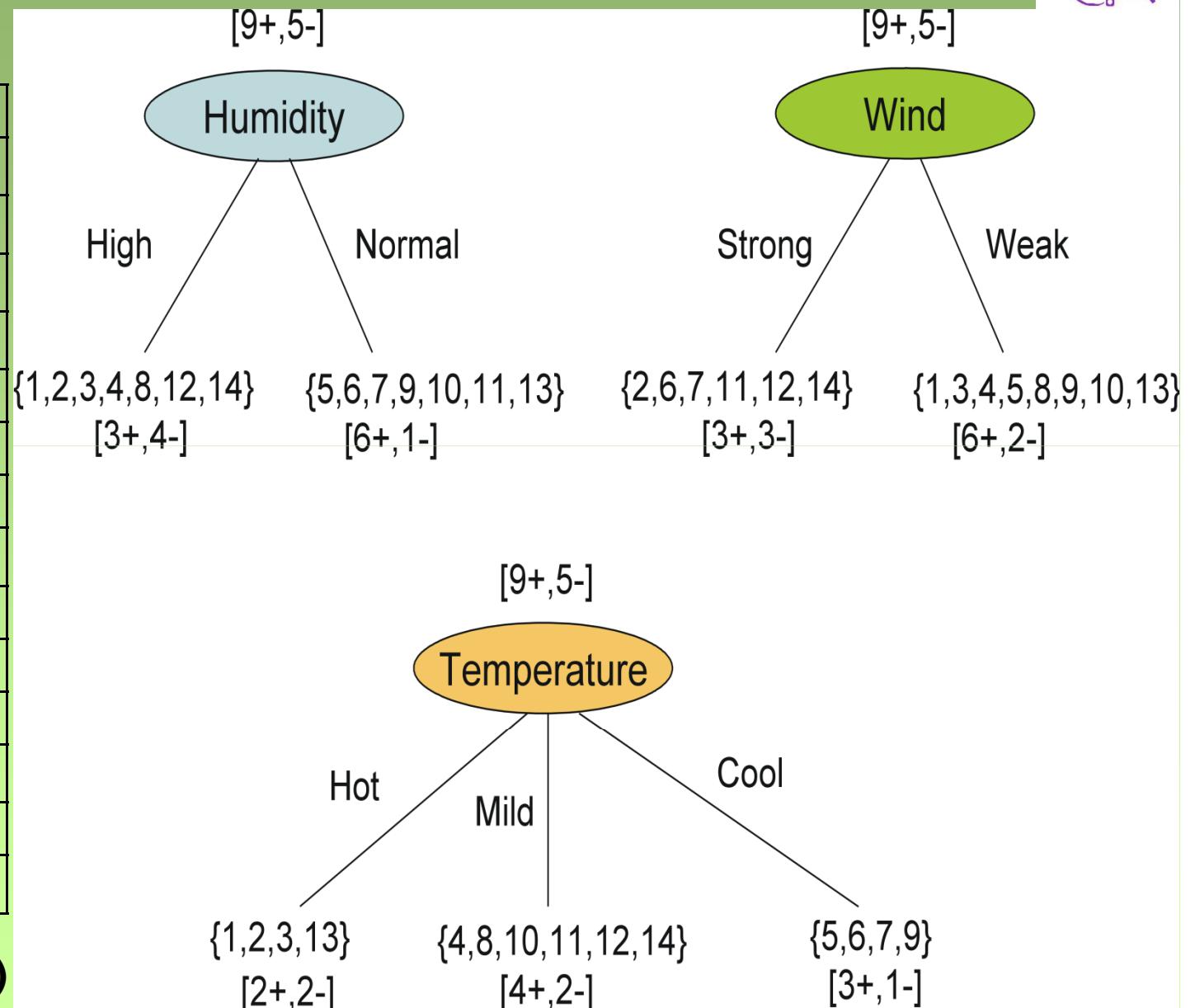


EJEMPLO ID3: PLAY TENNIS



Day	Humidity	Play
1	High	No
2	High	No
3	High	Yes
4	High	Yes
5	Normal	Yes
6	Normal	No
7	Normal	Yes
8	High	No
9	Normal	Yes
10	Normal	Yes
11	Normal	Yes
12	High	Yes
13	Normal	Yes
14	High	No

High 3 Yes(+), 4 No(-)

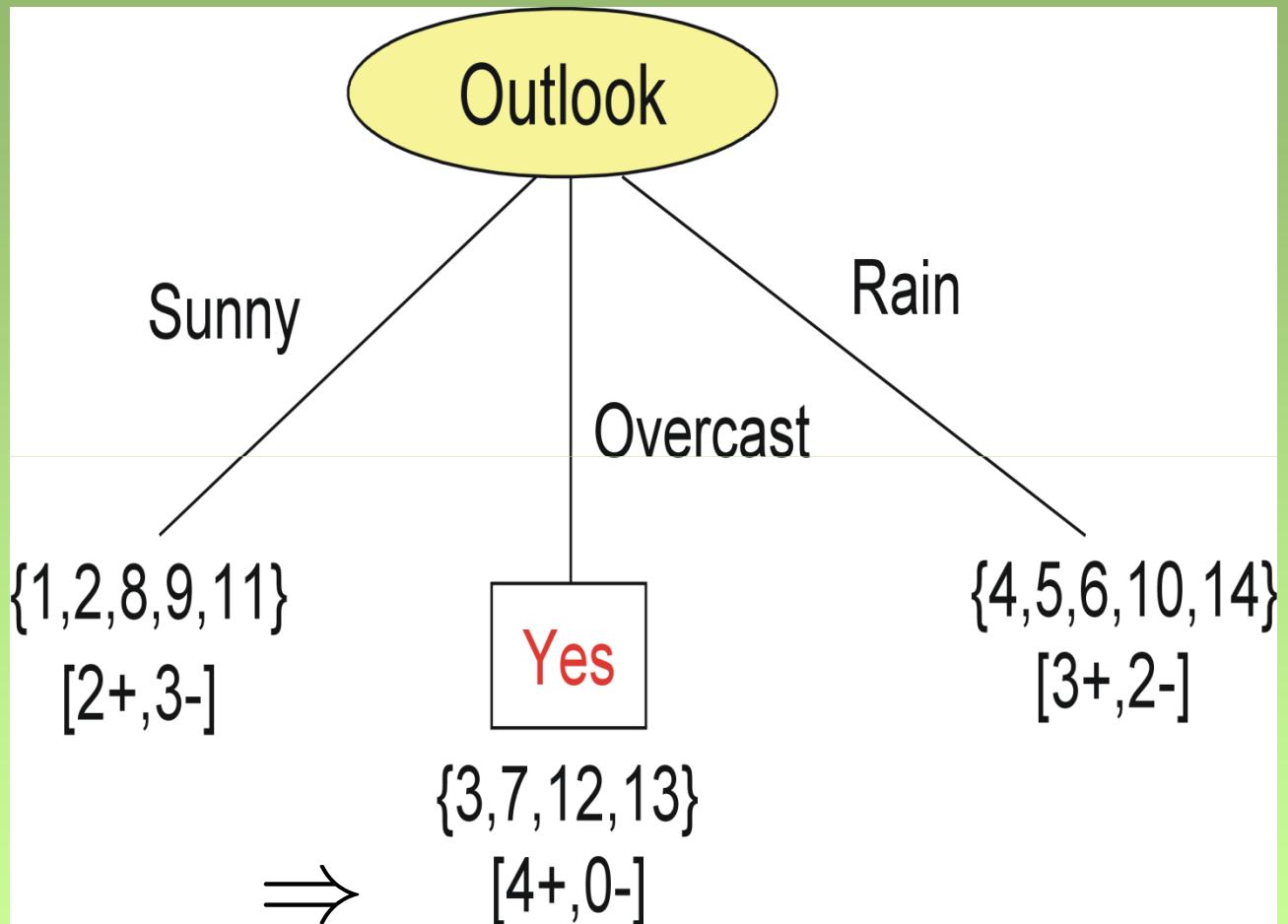




EJEMPLO ID3: PLAY TENNIS



Day	Outlook	Play
1	Sunny	No
2	Sunny	No
3	Overcast	Yes
4	Rain	Yes
5	Rain	Yes
6	Rain	No
7	Overcast	Yes
8	Sunny	No
9	Sunny	Yes
10	Rain	Yes
11	Sunny	Yes
12	Overcast	Yes
13	Overcast	Yes
14	Rain	No



Sunny 2+,3- Todos los ejemplos con Outlook=Overcast son positivos

Se convierte en nodo hoja

El resto tienen entropía no cero y el árbol debe seguir



EJEMPLO ID3: PLAY TENNIS



Day	Outlook	Play
1	Sunny	No
2	Sunny	No
3	Overcast	Yes
4	Rain	Yes
5	Rain	Yes
6	Rain	No
7	Overcast	Yes
8	Sunny	No
9	Sunny	Yes
10	Rain	Yes
11	Sunny	Yes
12	Overcast	Yes
13	Overcast	Yes
14	Rain	No

Se repite el proceso para cada nodo descendiente no terminal, usando sólo los ejemplos asociados con el nodo.

Cualquier atributo aparece como mucho una vez en cada camino

P.e., por la rama Sunny (con 5 instancias), buscamos el atributo siguiente:

$$I(C_{\text{sunny}}, \text{Temperature}) = 0,97 - 0,40 = 0,57$$

$$I(C_{\text{sunny}}, \text{Humidity}) = 0,94$$

$$I(C_{\text{sunny}}, \text{Wind}) = 0,02$$

$$H(C_{\text{sunny}}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0,97$$

$$H(C_{\text{sunny}} \mid \text{Tem}) = -\sum_{c=1}^n p(x \mid c) \log_2 p(c \mid x) = -\sum_c \sum_x p(x, c) \log_2 p(c \mid x)$$



EJEMPLO ID3: PLAY TENNIS



Outlook	Temperature	Play Tennis
Sunny	Hot	No
Sunny	Hot	No
Sunny	Mild	No
Sunny	Cool	Yes
Sunny	Mild	Yes

$$I(C_{\text{sunny}}, \text{Temperature}) = 0,97 - 0,4 = 0,57$$

$$H(C_{\text{sunny}}) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0,97$$

$$\begin{aligned} H(C_{\text{sunny}} \mid \text{Tem}) &= -\sum_c \sum_x p(x, c) \log_2 p(c \mid x) = \\ &= -P(\text{Hot}, \text{No}) \log_2 P(\text{No} \mid \text{Hot}) - P(\text{Mild}, \text{No}) \log_2 P(\text{No} \mid \text{Mild}) \\ &\quad - P(\text{Mild}, \text{Yes}) \log_2 P(\text{Yes} \mid \text{Mild}) - P(\text{Cool}, \text{Yes}) \log_2 P(\text{Yes} \mid \text{Cool}) = \\ &= -(2/5) * \log_2 1 - (1/5) * \log_2 (1/2) - (1/5) * \log_2 (1/2) - (1/5) * \log_2 1 = \\ &= 0 + (1/5) + (1/5) + 0 = 2/5 = 0,40 \end{aligned}$$



EJEMPLO ID3: PLAY TENNIS

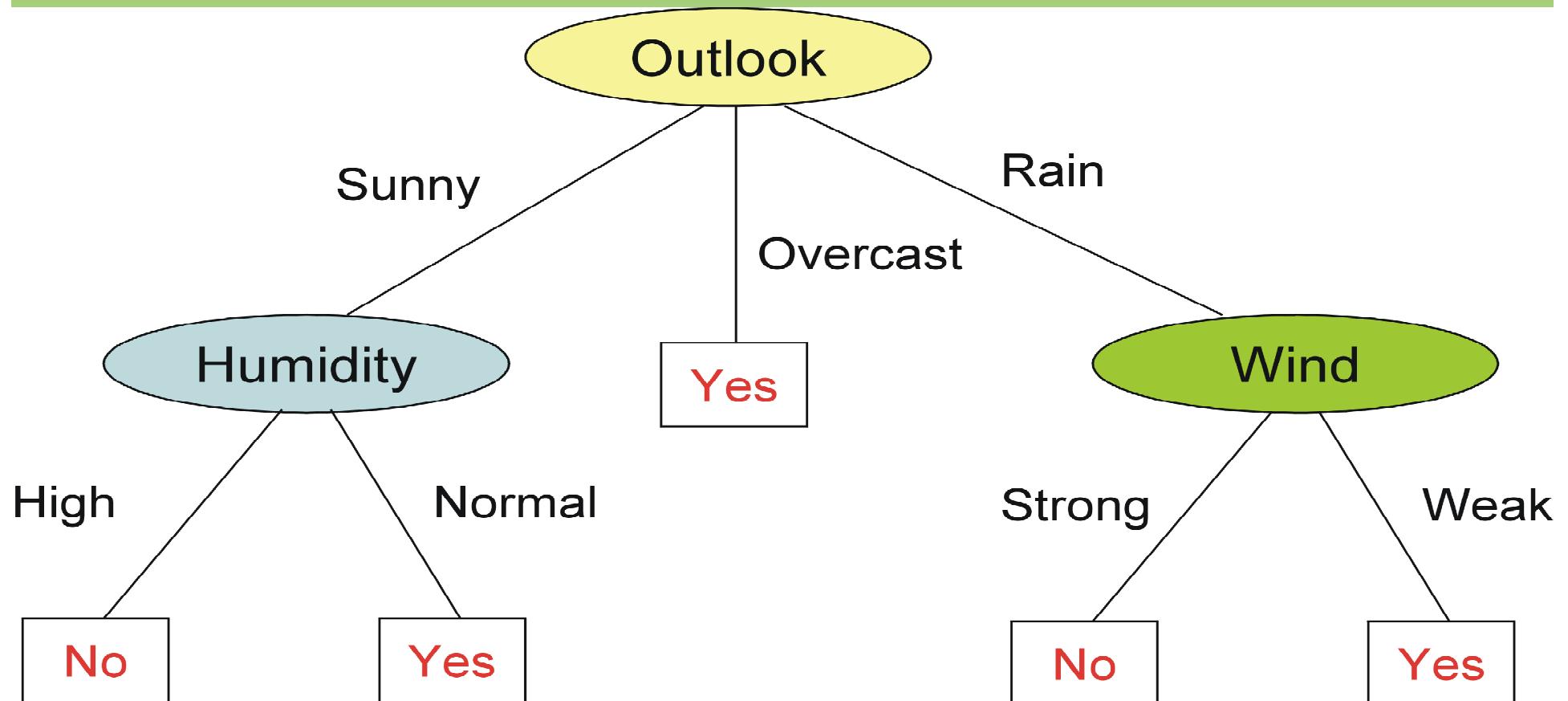


$$I(C_{\text{sunny}}, \text{Temperature}) = 0,97 - 0,4 = 0,57$$

$$I(C_{\text{sunny}}, \text{Humidity}) = 0,94$$

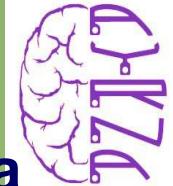
$$I(C_{\text{sunny}}, \text{Wind}) = 0,02$$

...Árbol final es el visto antes





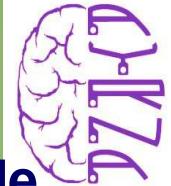
ALGORITMO ID3



- Seleccionar el atributo A_i que maximice la ganancia $G(A_i)$
- Crear un nodo para ese atributo con tantos sucesores como valores tenga.
- Introducir los ejemplos en los sucesores según el valor que tenga el atributo A_i
- Por cada sucesor,
 - Si solo hay ejemplos de una clase C_k
Entonces etiquetarlo con C_k
 - Si no, llamar al ID3 con una tabla formada por los ejemplos de ese nodo, eliminando la columna del atributo A_i .



HEURISTICA



Seleccionar el atributo que mejor separe (ordene) los ejemplos de acuerdo a las clases.

La Entropía es una medida de cómo está ordenado el universo

La Teoría de la Información (basada en la entropía) calcula el número de bits (información, preguntas sobre atributos) que hace falta suministrar para conocer la clase a la que pertenece un ejemplo.

Entropía de la clasificación de una colección de datos que pertenecen a una de entre dos categorías (clasificación binaria):

$$H(C) = -\sum_{i=1}^2 p(C_i) \log_2 p(C_i)$$

donde $p(C_1)$ es la probabilidad de ejemplos positivos sobre el total, y $p(C_2)$ es la probabilidad de ejemplos negativos.



ENTROPIA



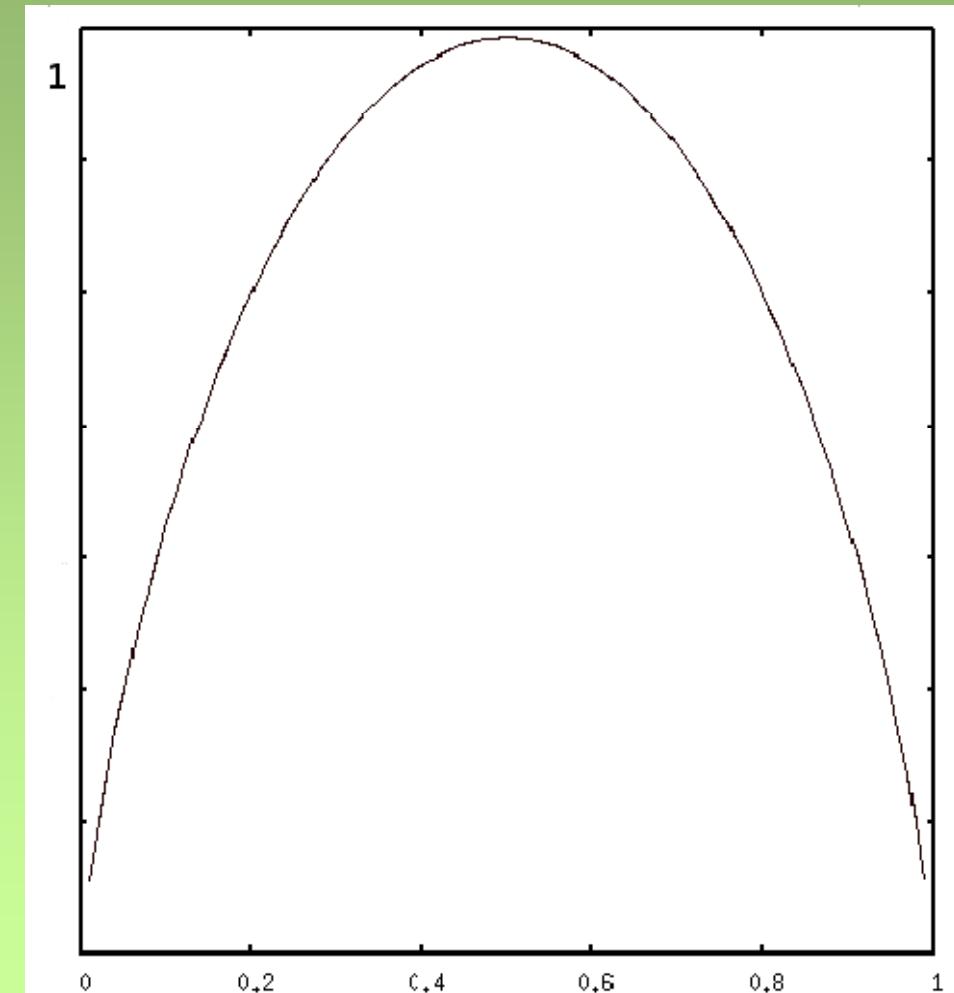
Si $p(C_1) = p(C_2) = 0,5$, Entonces la Entropía es máxima

La Entropía tiende a 0 cuanto mas se diferencian las probabilidades “a priori“ de las dos clases

Con múltiples clases:

$$H(C) = -\sum_{i=1}^K p(C_i) \log_2 p(C_i)$$

Objetivo: Minimizar la entropía





GANANCIA DE INFORMACION DE UN ATRIBUTO



Esperanza de reducción de entropía cuando se divide el conjunto de datos original según el atributo dado

$$A = \arg \max_{a \in A} G(a) = \arg \max_{a \in A} [H - H(a)] = \arg \min_{a \in A} H(a)$$

Entropía del atributo A_i :

$$H(A_i) = \sum_{j=1}^{nv(A_i)} \frac{n_{ij}}{n} H_{ij}$$

Entropía de la partición j del atributo A_i :

$$H_{ij} = -\sum_{k=1}^{nc} \frac{n_{ijk}}{n_{ij}} \log_2 \frac{n_{ijk}}{n_{ij}}$$



EJEMPLO 2



EJEMPLO DE ARBOL DE DECISIÓN

Ejemplo	Sitio de acceso A_1	1a cantidad gastada A_2	Vivienda (zona) A_3	Última compra A_4	Clase
1	1	0	2	Libro	Bueno
2	1	0	1	Disco	Malo
3	1	2	0	Libro	Bueno
4	0	2	1	Libro	Bueno
5	1	1	1	Libro	Malo
6	2	2	1	Libro	Malo



ID3 SOLUCIÓN DEL EJEMPLO 2



$$\begin{aligned}H(A_1) &= \sum_{j=1}^{nv(A_1)} \frac{n_{1j}}{n} H_{1j} = \sum_{j=1}^3 \frac{n_{1j}}{6} H_{1j} = \\&\frac{n_{10}}{6} H_{10} + \frac{n_{11}}{6} H_{11} + \frac{n_{12}}{6} H_{12} = \\&= \frac{1}{6} H_{10} + \frac{4}{6} H_{11} + \frac{1}{6} H_{12} = 0 + \frac{4}{6} + 0 \\&= 4 / 6 = 0.66\end{aligned}$$

A1	Clase
1	Bueno
1	Malo
1	Bueno
0	Bueno
1	Malo
2	Malo

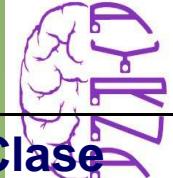
$$H_{10} = -\sum_{k=1}^2 \frac{n_{10k}}{n_{10}} \log_2 \frac{n_{10k}}{n_{10}} = -\frac{1}{1} \log_2 \frac{1}{1} - \frac{0}{1} \log_2 \frac{0}{1} = 0$$

$$H_{11} = -\sum_{k=1}^2 \frac{n_{11k}}{n_{11}} \log_2 \frac{n_{11k}}{n_{11}} = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1$$

$$H_{12} = -\sum_{k=1}^2 \frac{n_{12k}}{n_{12}} \log_2 \frac{n_{12k}}{n_{12}} = -\frac{0}{1} \log_2 \frac{0}{1} - \frac{1}{1} \log_2 \frac{1}{1} = 0$$



ID3 SOLUCIÓN DEL EJEMPLO 2



Ejemplo	Sitio de acceso A1	1a cantidad gastada A2	Vivienda (zona) A3	Última compra A4	Clase
1	1	0	2	Libro	Bueno
2	1	0	1	Disco	Malo
3	1	2	0	Libro	Bueno
4	0	2	1	Libro	Bueno
5	1	1	1	Libro	Malo
6	2	2	1	Libro	Malo

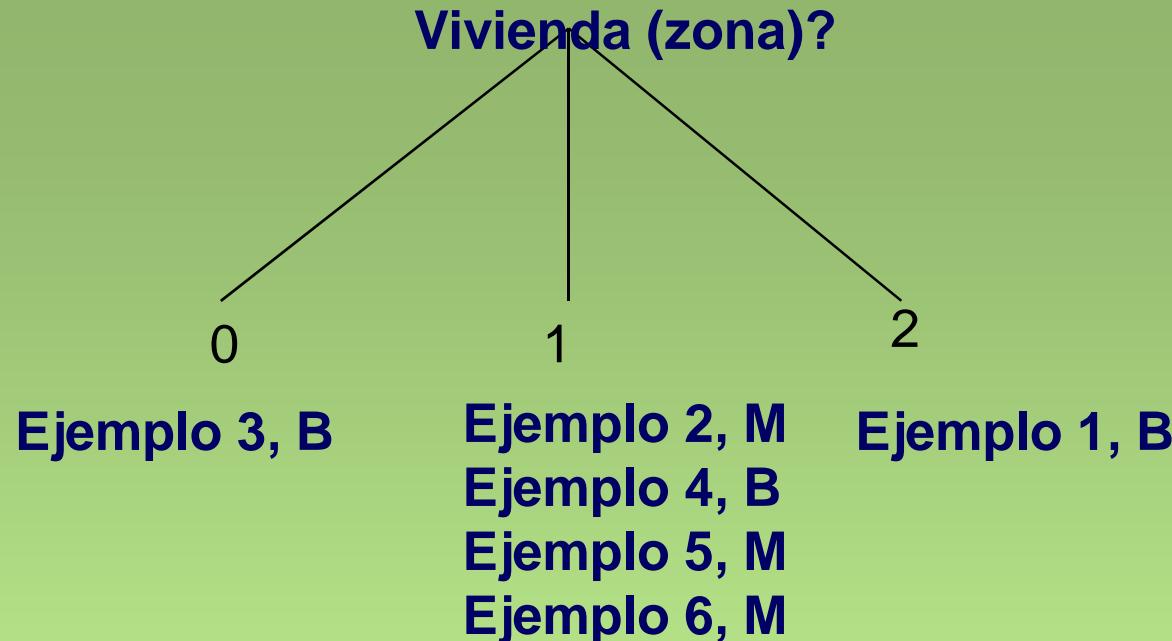
$$H(A_2) = \frac{2}{6} H_{20} + \frac{1}{6} H_{21} + \frac{3}{6} H_{22} = \frac{2}{6} 1 + \frac{1}{6} 0 + \frac{3}{6} \left(-\frac{2}{3} \log_2 \frac{2}{3} - \frac{1}{3} \log_2 \frac{1}{3} \right) = 0,79$$

$$H(A_3) = \frac{1}{6} H_{30} + \frac{4}{6} H_{31} + \frac{1}{6} H_{32} = \frac{1}{6} 1 + \frac{4}{6} \left(-\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4} \right) + \frac{1}{6} 0 = 0,54$$

$$H(A_4) = \frac{1}{6} H_{4Disco} + \frac{5}{6} H_{4Libro} = \frac{1}{6} 0 + \frac{5}{6} \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) = 0,81$$



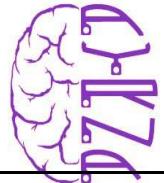
ID3 SOLUCIÓN DEL EJEMPLO 2



Ejemplo	Sitio de acceso A1	1a cantidad gastada A2	Vivienda (zona) A3	Última compra A4	Clase
1	1	0	2	Libro	Bueno
2	1	0	1	Disco	Malo
3	1	2	0	Libro	Bueno
4	0	2	1	Libro	Bueno
5	1	1	1	Libro	Malo
6	2	2	1	Libro	Malo



EJEMPLO 2 (Continuación)



Ejemplo	Sitio de Acceso A1	1 ^a Cantidad Gastada A2	Ultima Compra A4	Clase
2	1	0	Disco	Malo
4	0	2	Libro	Bueno
5	1	1	Libro	Malo
6	2	2	Libro	Malo

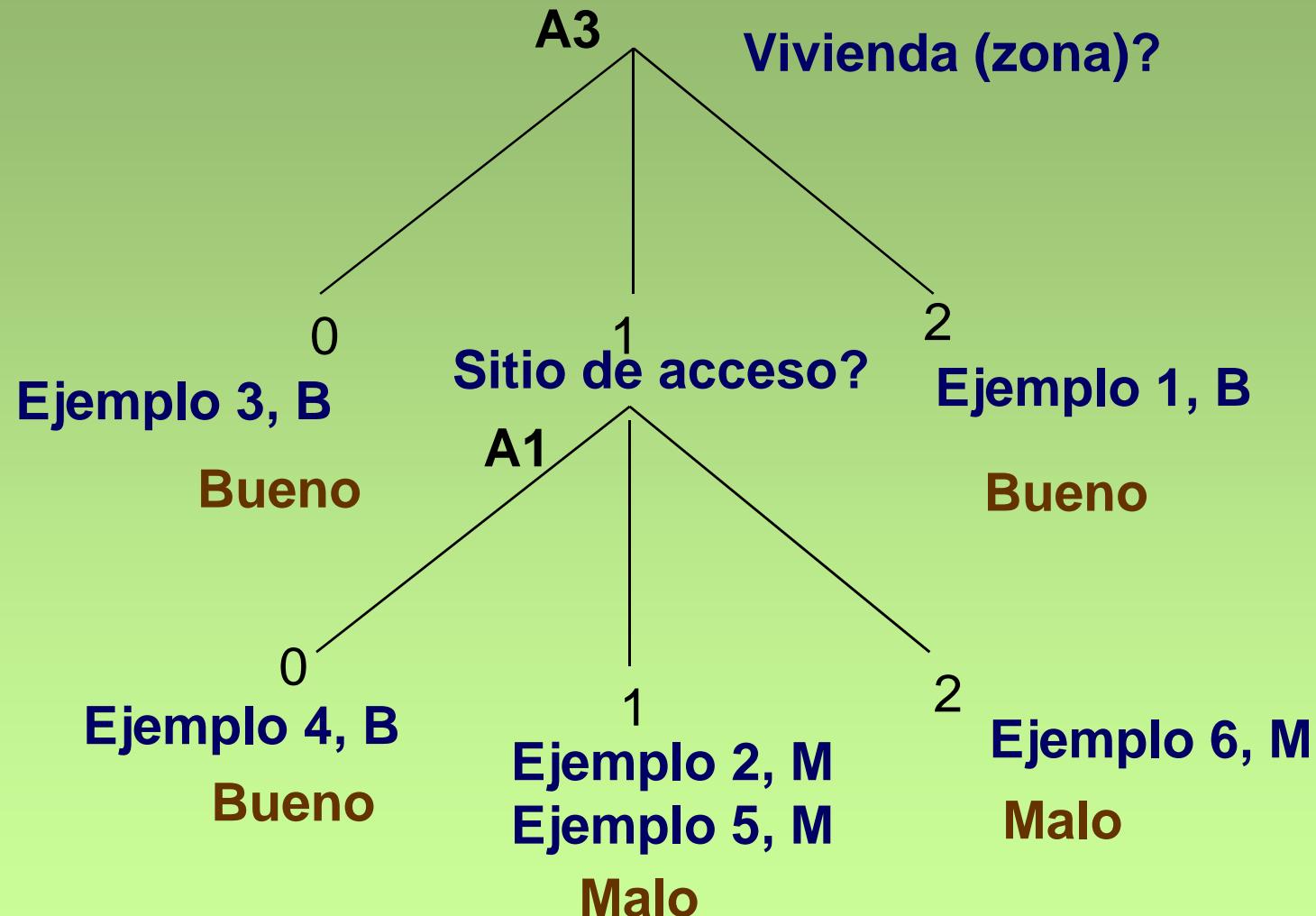
$$H(A_1) = \frac{1}{4}H_{10} + \frac{2}{4}H_{11} + \frac{1}{4}H_{12} = \frac{1}{4}0 + \frac{2}{4}0 + \frac{1}{4}0 = 0$$

$$H(A_2) = \frac{1}{4}H_{20} + \frac{1}{4}H_{21} + \frac{2}{4}H_{22} = \frac{1}{4}0 + \frac{1}{4}0 + \frac{2}{4}1 = 0,5$$

$$H(A_4) = \frac{1}{4}H_{4Disco} + \frac{3}{4}H_{4Libro} = \frac{1}{4}0 + \frac{3}{4}\left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right) = 0,23$$



ID3 SOLUCIÓN DEL EJEMPLO 3





Traducción a reglas del ejemplo 2



Cualquier árbol de decisión se puede convertir a reglas

Regla: estructura del tipo Si-Entonces

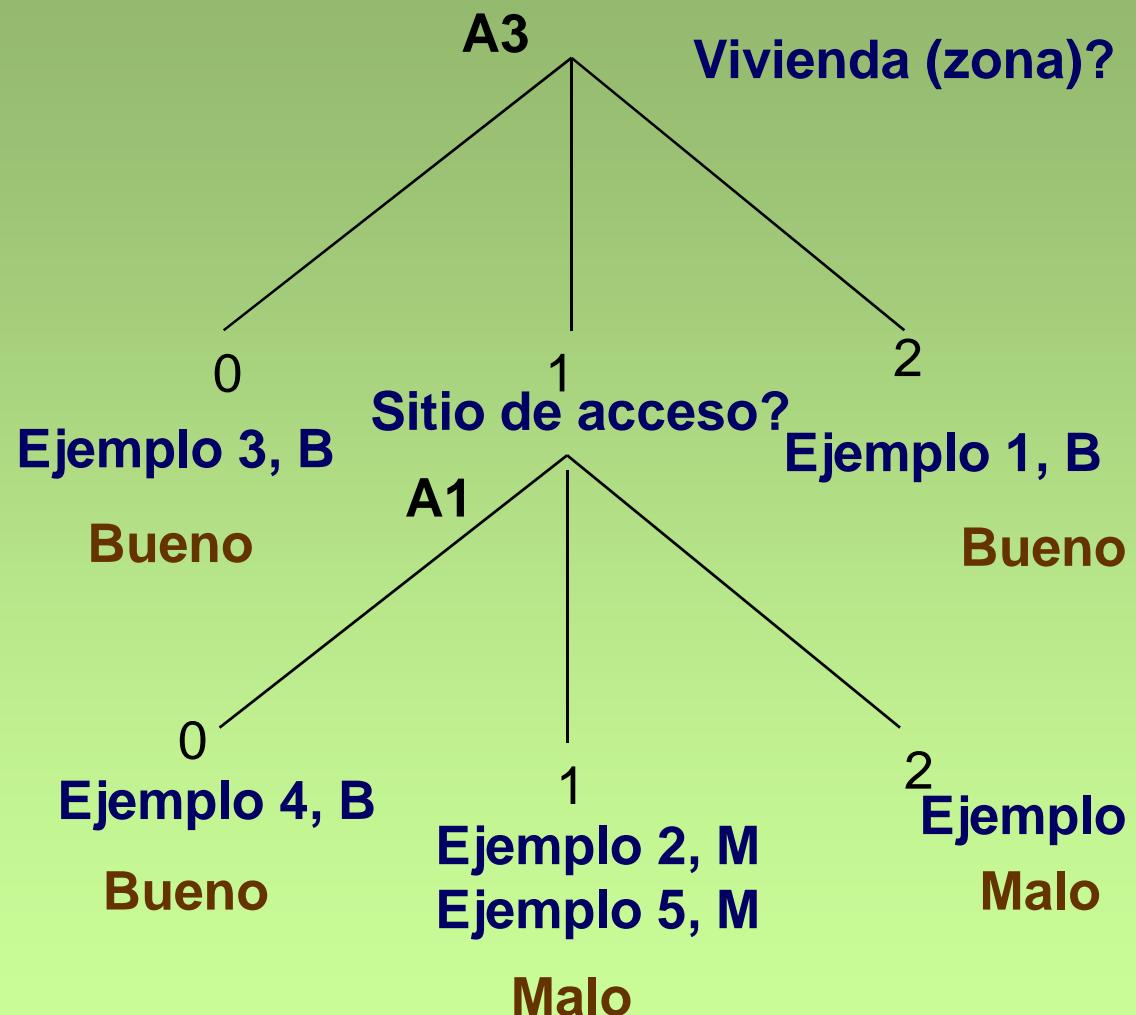
Ejemplo

**SI Vivienda (zona)=1 Y Sitio de acceso=0
ENTONCES Bueno**

Algoritmo: por cada rama del árbol, las preguntas y sus valores estarán en la parte izquierda de las reglas y la etiqueta del nodo hoja correspondiente será la parte derecha (clasificación)



Traducción a reglas del ejemplo 2



Si Vivienda (zona)=0
Entonces Bueno

Si Vivienda (zona)=1 y
Sitio de acceso=0
Entonces Bueno

Si Vivienda (zona)=1 y
Sitio de acceso=1
Entonces Malo

Si Vivienda (zona)=1 y
Sitio de acceso=2
Entonces Malo

Si Vivienda (zona)=2
Entonces Bueno



ID3 COMO UN PROBLEMA DE BUSQUEDA



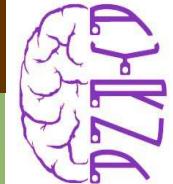
Conjunto de estados: cada estado es un árbol de decisión

Conjunto de operadores: el único operador es “introducir en un nodo la pregunta del atributo correspondiente”

Estado inicial: árbol de decisión vacío

Meta: árbol de decisión que separa los ejemplos de entrenamiento dependiendo de su clase

Heurística: elegir aquel atributo que minimice la entropía



1 Introducción

2 Algoritmo básico: ID3

3 Mejoras a ID3

4 C4.5

5 Tutorial de Weka

6 Conclusiones



ALGORITMO ID3

Observaciones generales I



■ Al ser voraz, puede que conduzca a una solución optima local en vez de global.

Existe una extensión de ID3 para añadir una forma de backtracking (post-poda).

■ Por usar propiedades estadísticas de todos los ejemplos (en la Ganancia de Información), la búsqueda es menos sensible a errores en los datos.



ALGORITMO ID3



Observaciones generales II

Una extensión de ID3 para manejar datos con ruido, es modificar su criterio de parada: crear hoja sin necesidad de esperar a que todas las etiquetas sean iguales (etiquetar con la mayoría)

- La complejidad crece linealmente con el número de instancias de entrenamiento y exponencialmente con el numero de atributos
- Matemáticamente se demuestra que favorece la elección de variables con mayor número de valores



Cuanto hacer crecer el árbol



Problema de sobreajuste

Problemas por hacer crecer el árbol hasta que clasifique correctamente todos los ejemplos de entrenamiento:

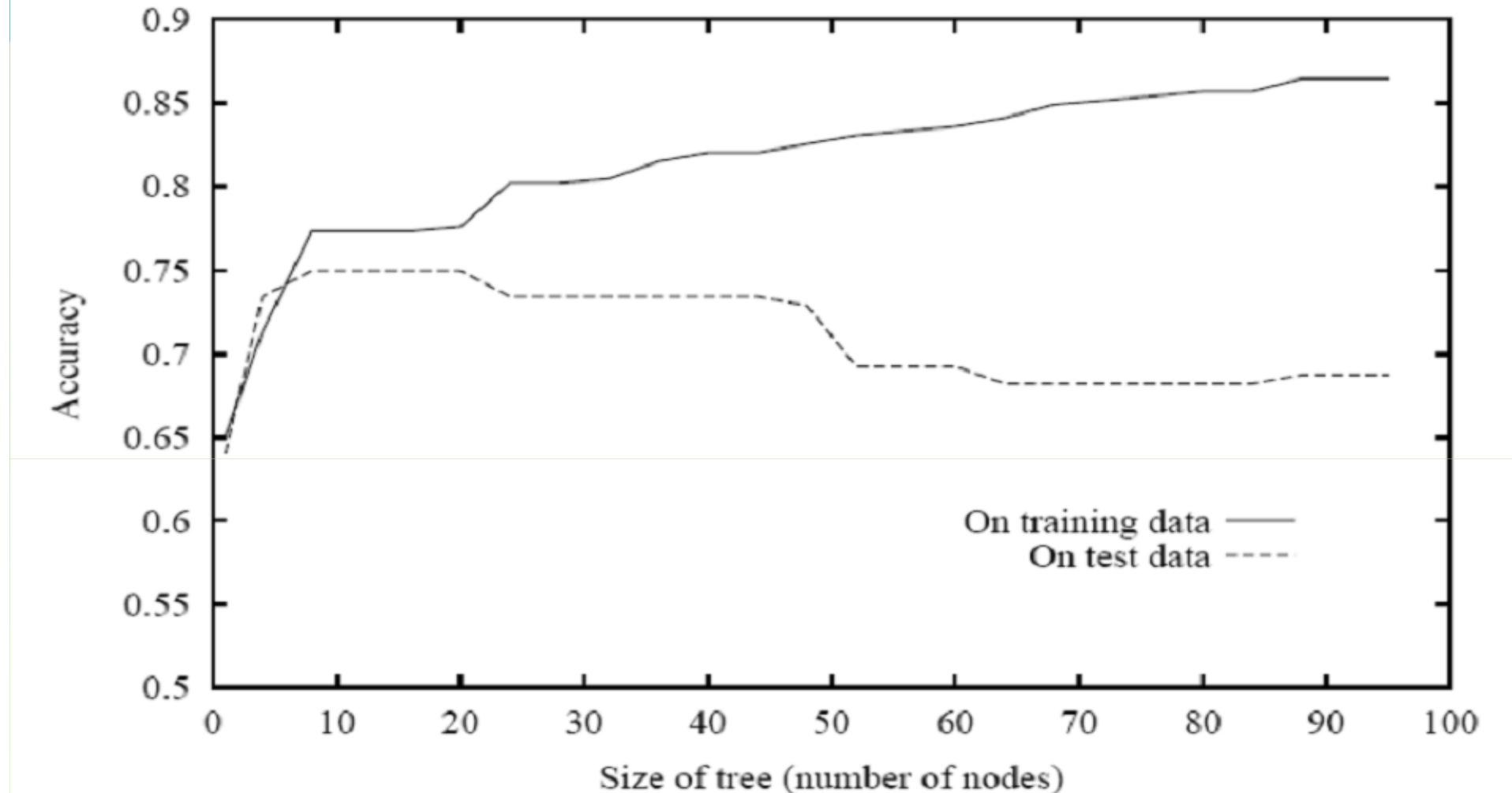
Si hay ruido en los ejemplos, !podemos aprender el ruido!

Si hay pocos ejemplos asociados a los nodos hoja, no son representativos de la verdadera función

ID3 produce árboles que **sobreajustan los ejemplos de entrenamiento (y no funciona adecuadamente con nuevos ejemplos)**. El modelo no es capaz de generalizar



Cuanto hacer crecer el árbol





SOBREENTRENAMIENTO



Soluciones al sobreentrenamiento

Hay dos grupos de técnicas, que tratan de simplificar el árbol:

Pre-poda: parar de aumentar el árbol antes de que alcance el punto en el que clasifica perfectamente los ejemplos de entrenamiento

⇒ Difícil estimar cuándo

Post-poda: permitir que sobreajuste los datos, y después podarlo reemplazando subárboles por una hoja

⇒ La mejor en la práctica, aunque tiene mayor coste computacional.



SOBREENTRENAMIENTO



Pre-podas

Aplicar un test estadístico para estimar si expandiendo un nodo particular es probable producir una mejora mas allá del conjunto de entrenamiento (e.g. el test 2 descrito en Quinlan, 1986)

Post-podas

Comenzando desde abajo, examinar los subárboles de los nodos no terminales.

Podar un nodo significa eliminar su subárbol correspondiente con raíz en ese nodo, convertir el nodo en nodo hoja y asignarle la clasificación mas común de los ejemplos de entrenamiento asociados a ese nodo.



SOBREENTRENAMIENTO



- **Post-podas**
- **Se poda solo si el árbol podado resultante mejora o iguala el rendimiento del árbol original sobre el conjunto de testeo.**
- **Podar iterativamente, escogiendo siempre el nodo a podar que mejore más la precisión en el conjunto de testeo.**
- **... hasta que ya no convenga (disminuya la precisión).**



OTRAS MEDIDAS DE SELECCION DE ATRIBUTOS



Favorecer la elección de atributos con muchos valores

Ejemplo extremo: “Fecha: Día” en el problema de Play Tennis sale elegido como raíz (predice perfectamente los ejemplos de entrenamiento y tendríamos un árbol muy ancho y de profundidad 1, con un nodo hoja por cada ejemplo)

Su ganancia de información es la mayor (al tener tantas ramas le obliga a separar los ejemplos en subconjuntos muy pequeños)

...Pero sería muy malo en ejemplos no vistos



EJEMPLO



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



OTRAS MEDIDAS DE SELECCION DE ATRIBUTOS

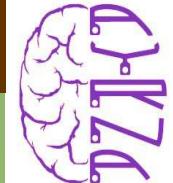


Favorecer la elección de atributos con muchos valores

Por ello hay otras medidas en vez de la ganancia de información:

⇒ p.e. ratio de ganancia $I(C;X_i)/H(X_i)$, que penaliza los atributos con muchos valores y uniformemente distribuidos.

Algunos estudios experimentales hablan de no demasiada influencia de estas medidas sobre el rendimiento, y sí de las post-podas.



1 Introducción

2 Algoritmo básico: ID3

3 Mejoras a ID3

4 C4.5

5 Tutorial de Weka

6 Conclusiones



ALGORITMO C4.5 [QUINLAN, 1993]



Escoge atributos usando el porcentaje de ganancia)
para (maximizarla)

$$I(C; X_i) / H(X_i)$$

Incorporación de post-poda de reglas: generar las
reglas (una por camino) y eliminar precondiciones
(antecedentes) siempre que mejore o iguale el error



ALGORITMO C4.5 [QUINLAN, 1993]



Algoritmo

Convertir el árbol en un conjunto de reglas R

Error = error de clasificación con R

Para cada regla r de R:

Nuevo-error = error al eliminar antecedente j de r

Si Nuevo-error \leq Error,

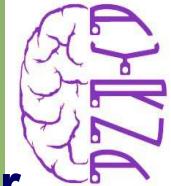
Entonces Nuevo-error = Error y eliminar de r este
antecedente

Si no hay más antecedentes en r, borrar r de R

Ordenar las reglas por su error estimado (de menos a más) y
considerar esta secuencia cuando se clasifiquen instancias



ALGORITMO C4.5



- **Poda pesimista:** C4.5 suele estimar el error e por resustitución, pero corrigiéndolo hacia una posición pesimista: mediante la aproximación $e + 1,96 * \hat{\sigma}$, siendo $\hat{\sigma}$ una estimación de la desviación típica.
- **¿Por qué usar reglas en vez del árbol?**
 - Podemos podar contextos (caminos), en vez de subárboles
 - Fáciles de entender
- **Última versión:** C4.8, que es el C4.5 revisión 8, última versión publica de esta familia de algoritmos antes de la implementación comercial C5.0



implementado en WEKA como J48



Bibliografía



Textos

- Alpaydin, E (2004) **Introduction to Machine Learning**, MIT Press
- Duda, R., Hart, P.E., Stork, D.G. (2001) **Pattern Classification**, Wiley
- Hernández-Orallo, J., Ramírez, M.J., Ferri, C. (2004) **Introducción a la Minería de Datos**, Pearson Educación
- Mitchell, T. (1997) **Machine Learning**, McGraw-Hill
- Webb, A. (2002) **Statistical Pattern Recognition** Wiley
- Witten, I., Frank, E. (2005) **Data Mining**, Morgan Kaufmann, 2a ed.

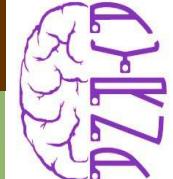


Bibliografía



Artículos

- Quinlan, J.R. (1986) Induction of trees, *Machine Learning*, 1, 81-106. [ID3]
- Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984) *Classification and Regression Trees*, Wadsworth. [CART]
- Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann. [C4.5]
- Quinlan, J.R. (1992) Learning with continuous classes, Proc. of the 5th Australian Joint Conference on AI, 343-348. [M5]
- Wang, Y., Witten, I. (1997) Induction of model trees for predicting continuous classes, Proc. of the Poster Papers of the ECML, 128-137 [M5']
- Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I. (1998) Using model trees for classification, *Machine Learning*, 32, 63-76
- Friedman, J.H. (1991) Multivariate adaptive regression splines, *Annals of Statistics*, 19, 1-141 [MARS]



1 Introducción

2 Algoritmo básico: ID3

3 Mejoras a ID3

4 C4.5

5 Tutorial de Weka

6 Conclusiones



APRENDIZAJE AUTOMÁTICO: TERCER CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

Tutorial de Weka Clasificación utilizando árboles de decisión

César Hervás-Martínez
Grupo de Investigación AYRNA

Departamento de Informática y Análisis
Numérico
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es



EJEMPLO BBDD IRIS

Primera meta de WEKA: Clasificar muestras de iris



Árbol de
decisión
en WEKA

?

Meta:

Crear un árbol de
decisión en WEKA
para clasificar flores
de lírios de especie
desconocida



Iris setosa



Iris versicolor



Iris virginica



CONCEPTO DE ÁRBOL DE DECISIÓN

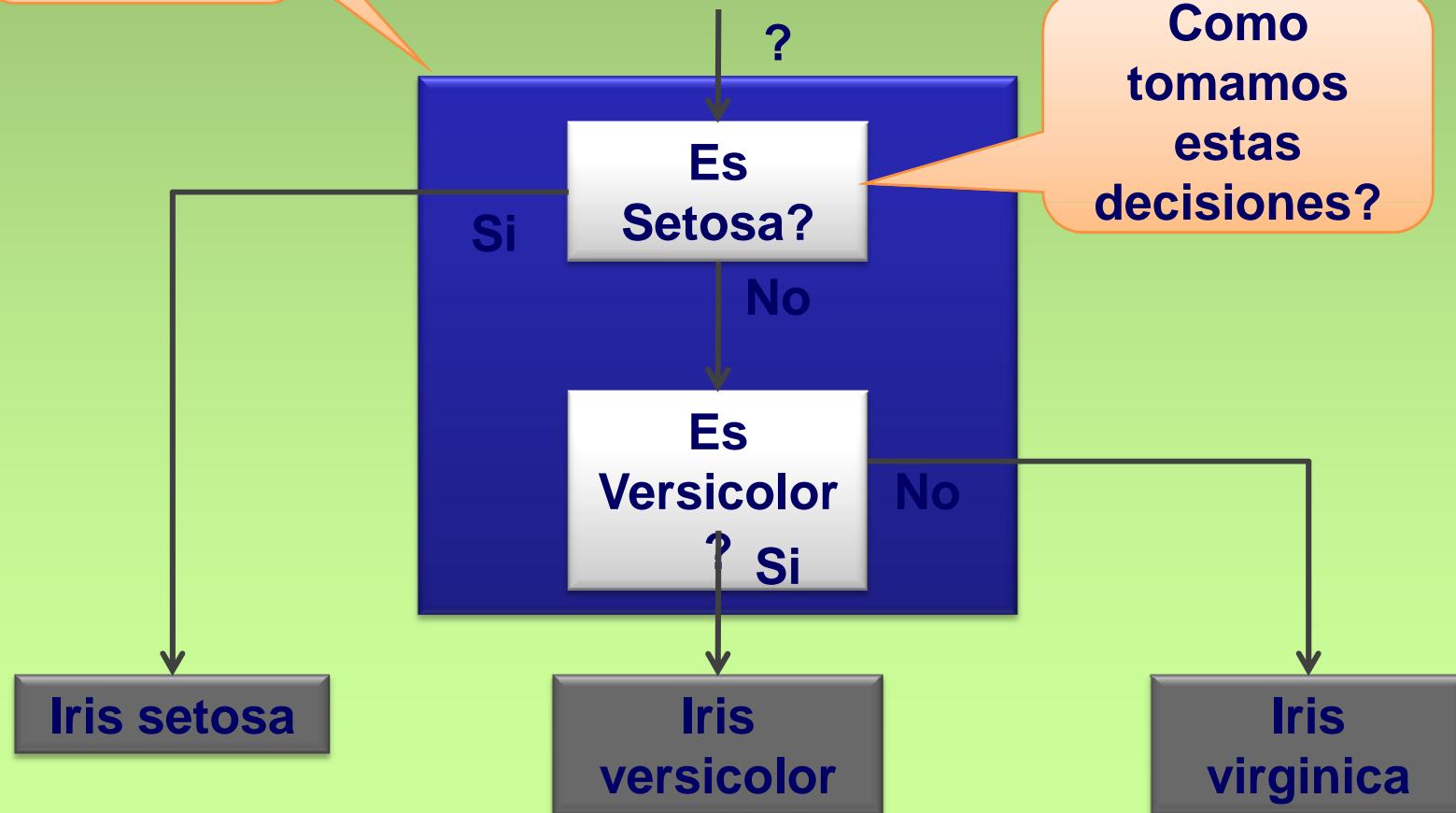


Reglas del
árbol de
decisión

Medidas (características):

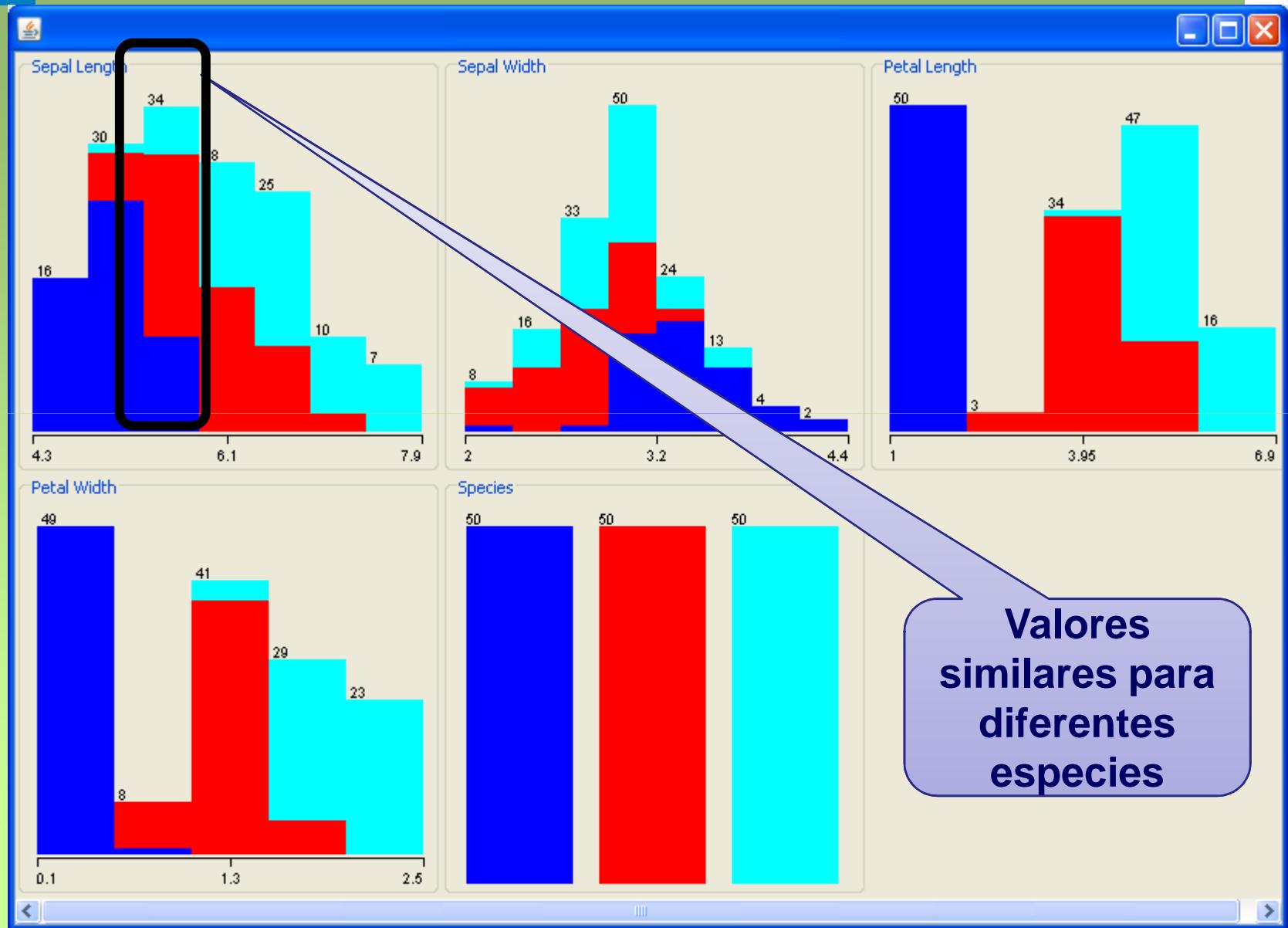
- Longitud del Sepalo
- Ancho del Sepalo
- Longitud del Petalo
- Ancho del Petalo

Como
tomamos
estas
decisiones?





INCERTIDUMBRE EN EL ÁRBOL DE DECISIÓN





INCERTIDUMBRE EN EL ÁRBOL DE DECISIÓN



Instance	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
35	4.9	3.1	1.5	0.2	setosa
38	4.9	3.6	1.4	0.1	setosa
58	4.9	2.4	3.3	1	versicolor
107	4.9	2.5	4.5	1.7	virginica
5	5	3.6	1.4	0.2	setosa

¿Que especie de iris es aquella que tiene una longitud de sepalo de 4.9 cm?



CONCEPTO DE ÁRBOL DE DECISIÓN



Como
obtenemos estas
probabilidades

Medidas (características):
•Longitud del Sepalo
•Ancho del Sepalo
•Longitud del Petalo
•Ancho del Petalo

?

Probablemente Si

Es
Setosa?

Probablemente No

Es
Versicolor

Probablemente No

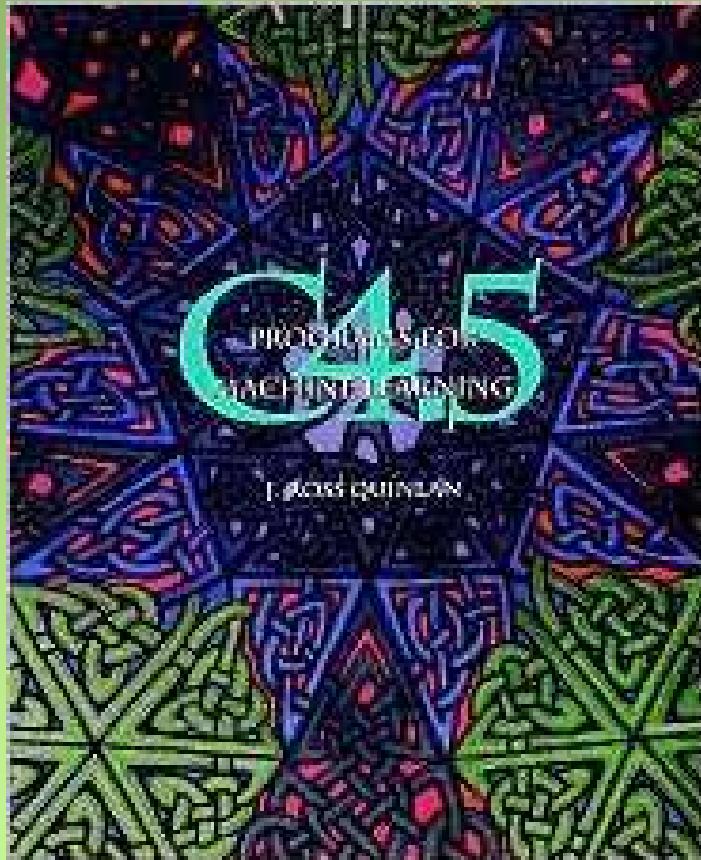
Probablemente Si

Iris
versicolor

Iris
virginica



WEKA J4.8 APRENDIZAJE DE ÁRBOLES DE DECISIÓN



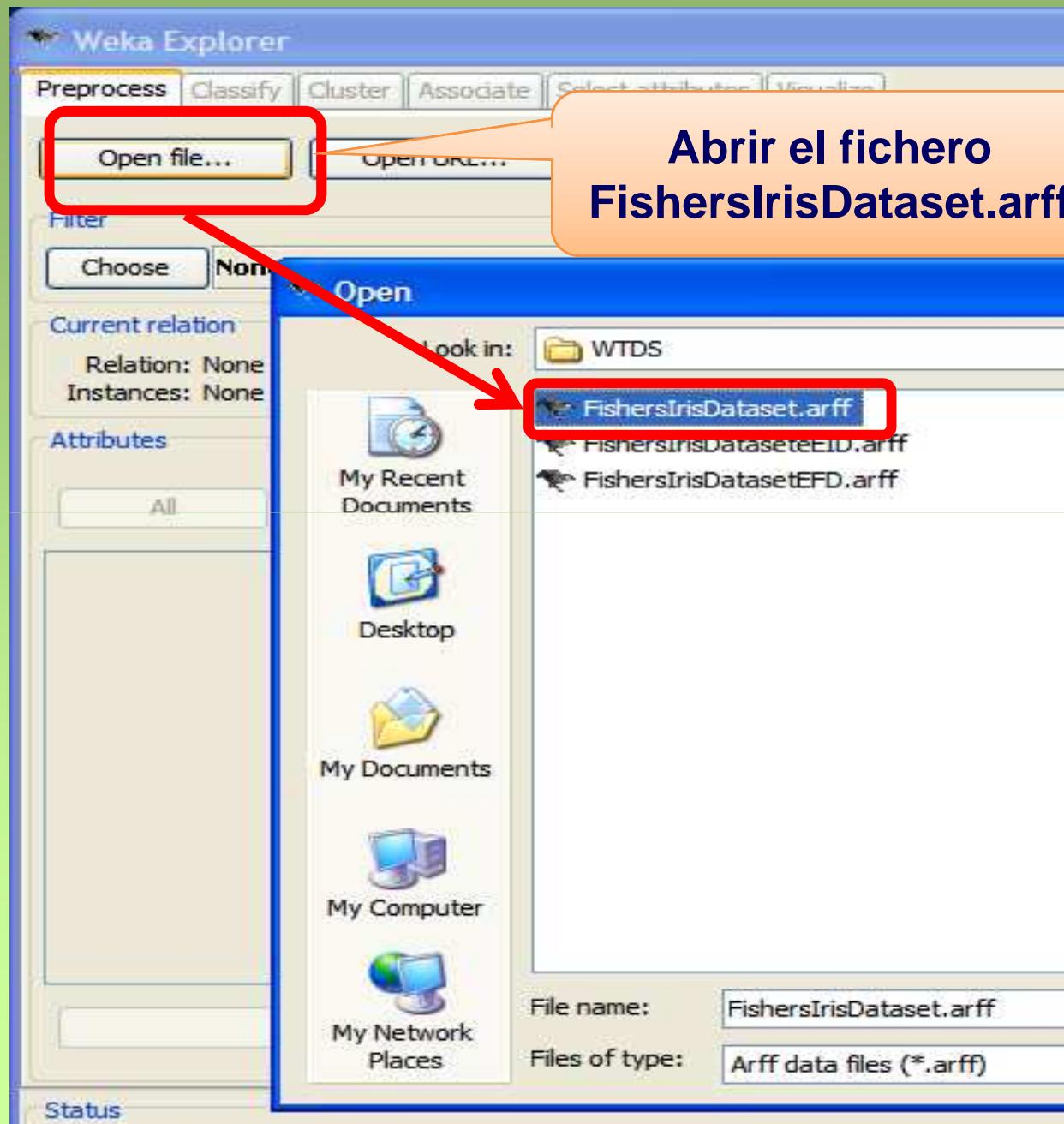
La herramienta
utilizada por el
tutorial

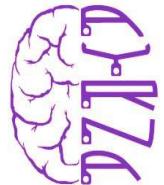
Weka J4.8
Aprendizaje de
Árboles de Decisión

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



WEKA J4.8





Elegir el algoritmo de Árboles de Decisión

The screenshot shows the Weka Explorer interface. On the left, there is a large orange callout box containing the following text:

weka
→ classifiers
→ trees
→ J48

The main window has several tabs at the top: Preprocess, Classify, Cluster, Associate, Select attributes, and Visualize. The 'Classify' tab is selected. Below it, there is a 'Choose' button with a red box around it, and a dropdown menu currently set to 'zeroR'. A red arrow points from the bottom of this box down to the 'J48' option in the list of classifiers.

The right side of the window shows a tree view of available classifiers under the 'weka' folder. The 'trees' folder is expanded, showing various classifier options. The 'J48' classifier is highlighted with a red box. Other classifiers listed include ADTree, BFTree, DecisionStump, FT, IBD, LADTree, LMT, M5P, NBTree, RandomForest, RandomTree, REPTree, SimpleCart, and UserClassifier.



Pulse el botón izquierdo del ratón sobre J48 para abrir el Editor Genérico de Objetos

The 'weka.gui.GenericObjectEditor' dialog is open, showing configuration options for the J48 classifier. The 'About' section states: "Class for generating a pruned or unpruned C4." The dialog lists several parameters:

- binarySplits: False
- confidenceFactor: 0.25
- debug: False
- minNumObj: 2
- numFolds: 3
- reducedErrorPruning: False
- saveInstanceData: True (highlighted with a red box)
- seed: 1
- subtreeRaising: True
- unpruned: False
- useLaplace: False

At the bottom are buttons: Open..., Save..., OK (highlighted with a red box), and Cancel.

WEKA J4.8

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

Use training set
 Supplied test set
 Cross-validation Folds 10
 Percentage split % 66

(Nom) Species

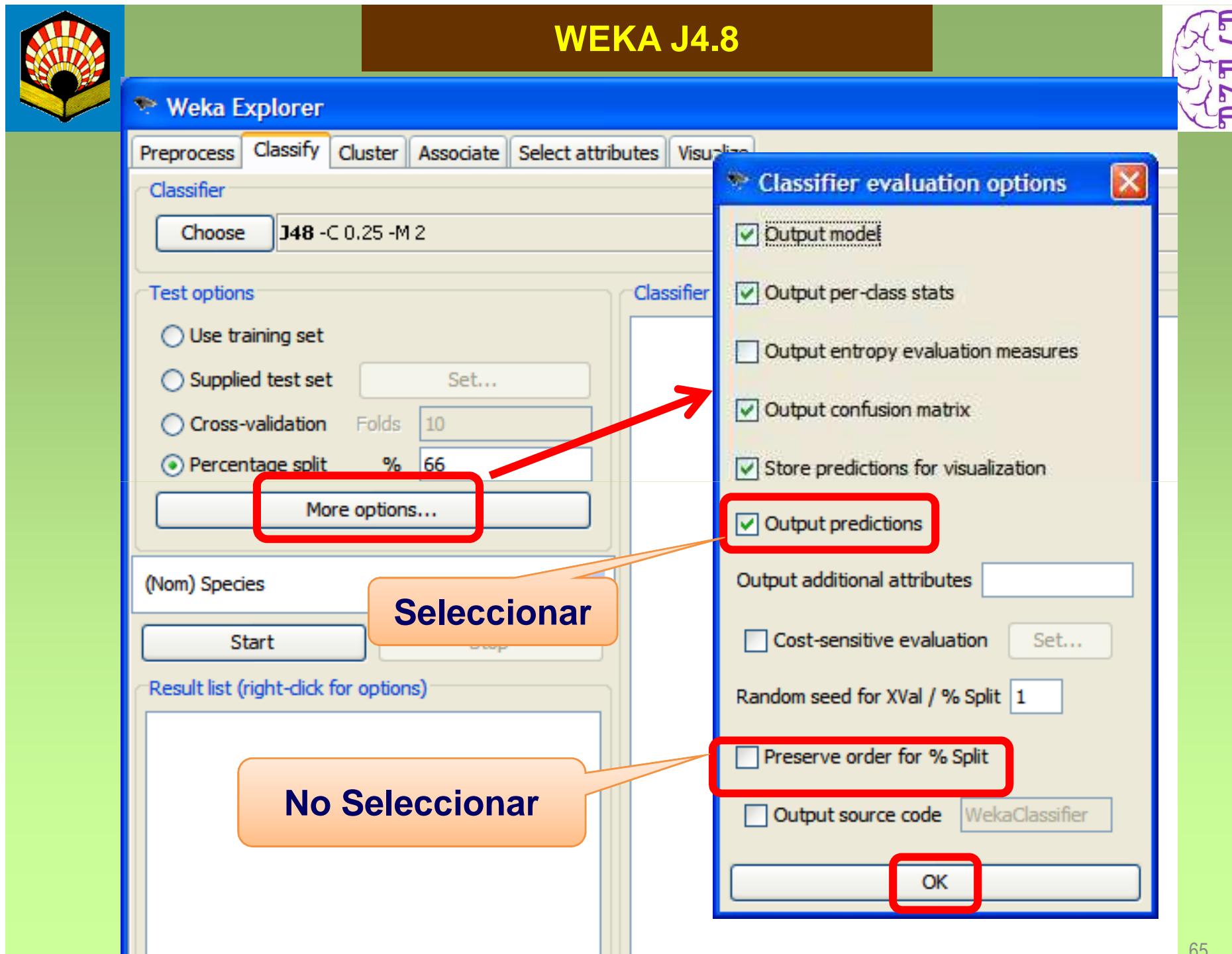
Result list (right-click for options)

Seleccionar

No Seleccionar

Classifier evaluation options

Output model
 Output per-class stats
 Output entropy evaluation measures
 Output confusion matrix
 Store predictions for visualization
 Output predictions
 Output additional attributes
 Cost-sensitive evaluation
Random seed for XVal / % Split 1
 Preserve order for % Split
 Output source code WekaClassifier





WEKA J4.8 OPCIÓN DE PORCENTAJE DE PATRONES PARA ENTRENAMIENTO (holdout)



Weka Explorer

Preprocess Classify Cluster Associate Selection

Classifier

Choose J48 -C 0.25 -M 2

Test options

Use training set

Supplied test set Set...

Cross-validation Folds 10

Percentage split % 66

More options...

(Nom) Species

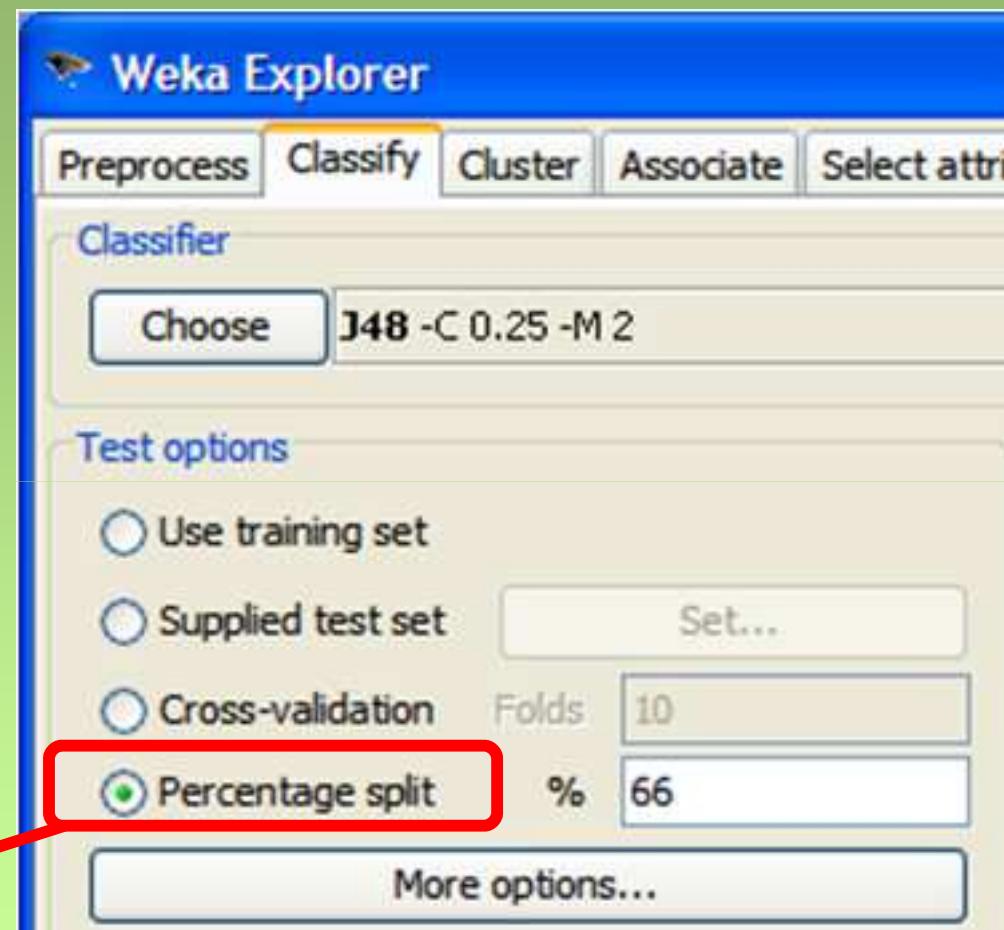
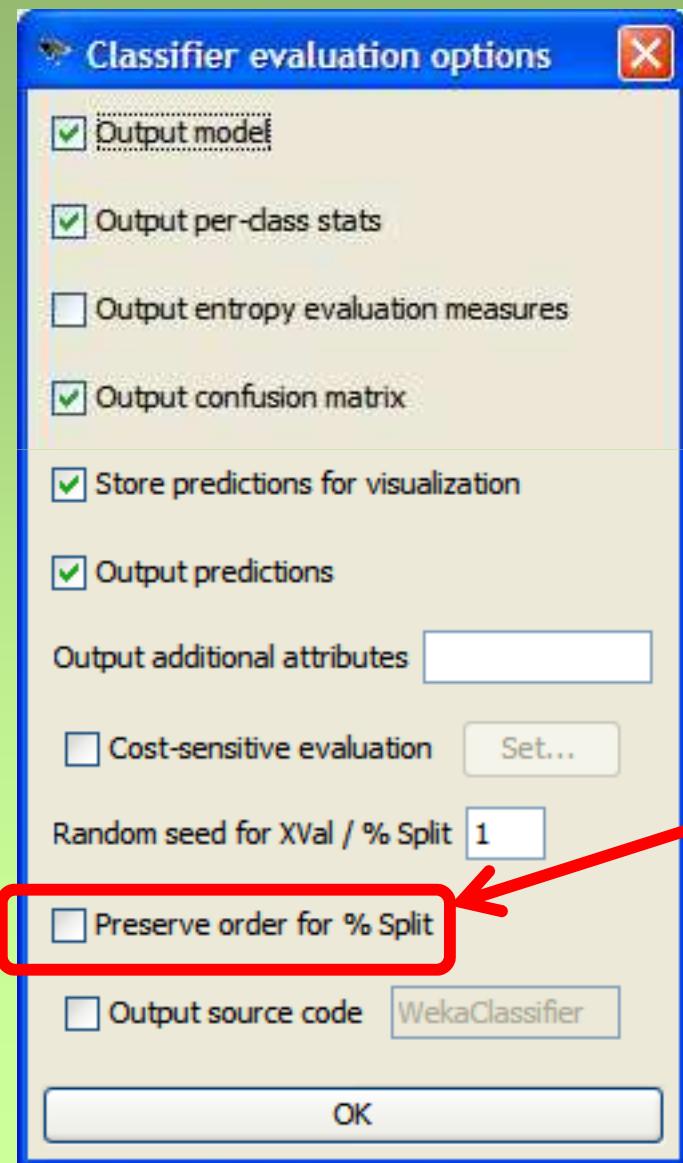
Start Stop

Result list (right-click for options)

Utilizaremos aproximadamente 2/3 de cada una de las clases para formar el conjunto de entrenamiento del árbol y los restantes para el conjunto de test utilizado para testar el árbol de decisión resultante



PORCENTAJE DE PATRONES PARA ENTRENAMIENTO (holdout) manteniendo el orden de los patrones en la BBDD





WEKA J4.8: Inicio y Resultados



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

Use training set

Supplied test set Set...

Cross-validation Folds 10

Percentage split % 66

More options...

(Nom) Species

Start Stop

Result list (right-click for options)

16:23:27 - trees.J48
16:25:11 - trees 148

Status OK Log x 0

Classifier output

Presentación de resultados

==== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: FishersIrisDataset-weka.filters.unsuper

Instances: 150

Attributes: 5

Sepal Length

Sepal Width

Petal Length

Petal Width

Species

Test mode: split 66.0% train, remainder test

Pulsar aquí para construir el árbol



Weka Exp

Preprocess

Classif

Classifier

Choose

J48 -C 0.25 -M 2

WEKA J4.8: Visualización del Árbol



Pulsar botón
derecho

Classifier output

== Run information ==

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: FishersIrisDataset-weka.filters.unsuperv
Instances: 150
Attributes: 5
Sepal Length
Sepal Width
Petal Length
Petal Width
Species
Test mode: split 66.0% train, remainder test

Start Stop

Result list (right-click for options)

16:32:36 - trees.J48

Status

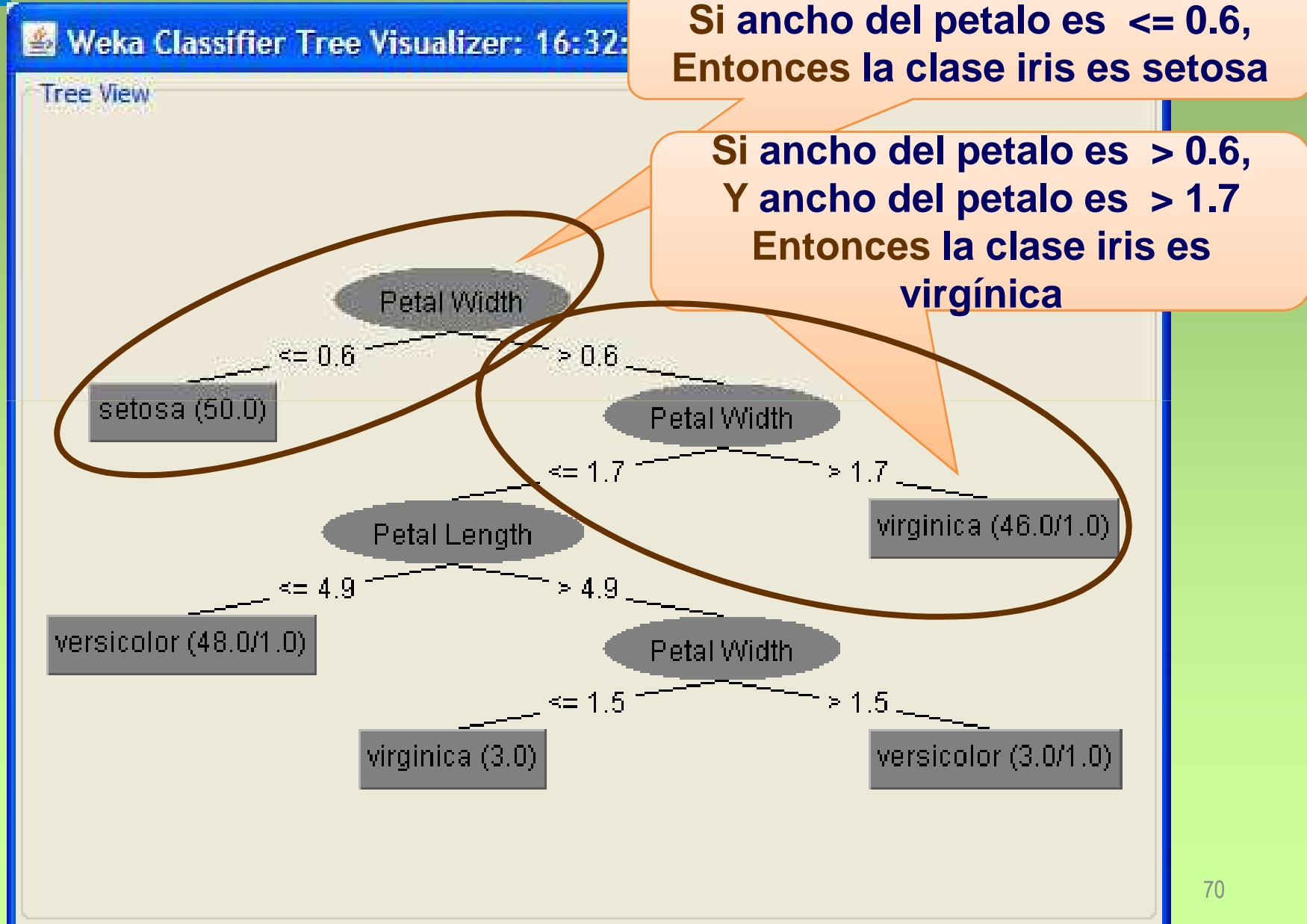
OK

- View in main window
- View in separate window
- Save result buffer
- Delete result buffer
- Load model
- Save model
- Re-evaluate model on current test set
- Visualize classifier errors
- Visualize tree**
- Visualize margin curve
- Visualize threshold curve
- Visualize cost curve

Pulsar botón
izquierdo

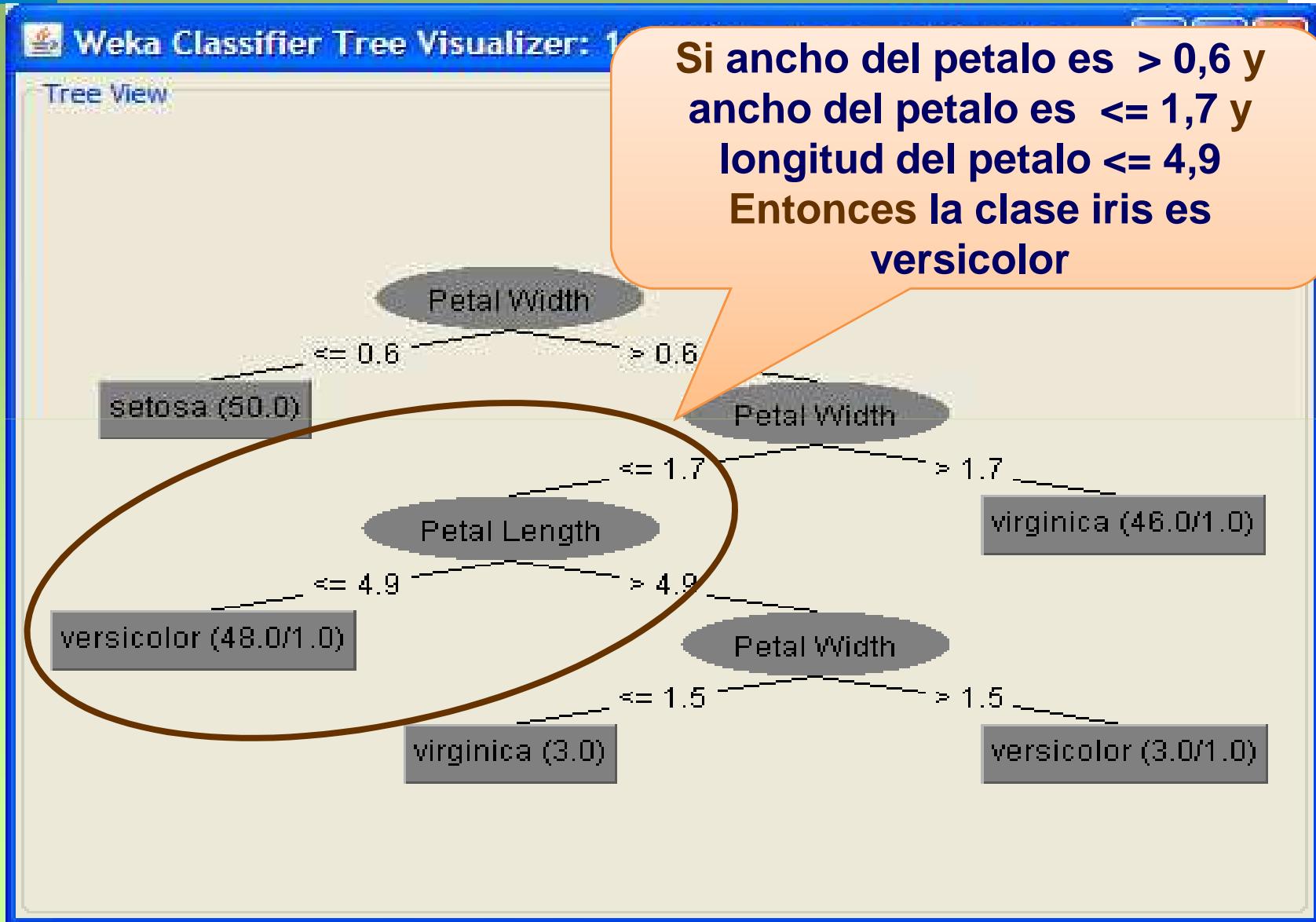


WEKA J4.8: INTERPRETAR EL ÁRBOL DE DECISIÓN



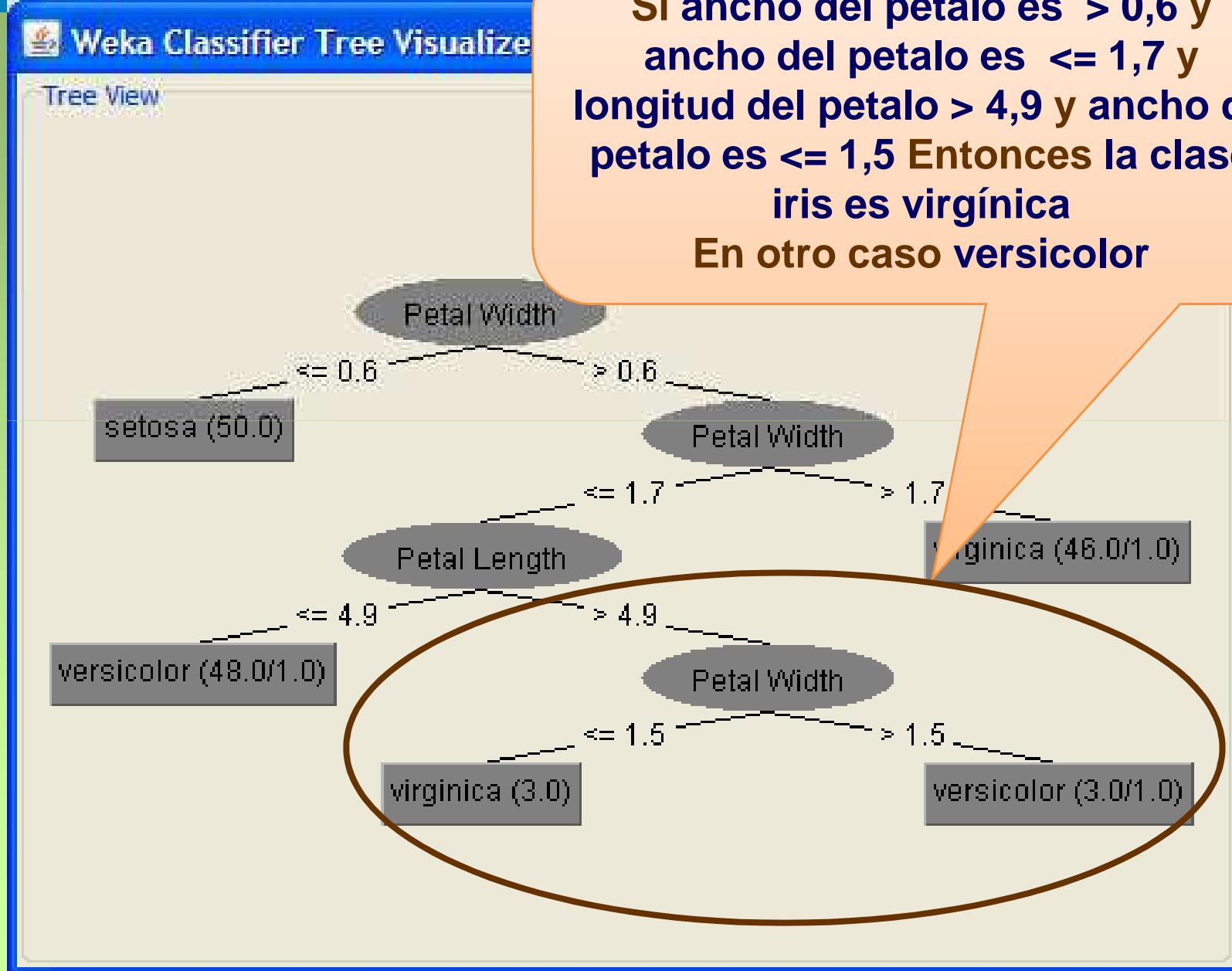


WEKA J4.8: INTERPRETAR EL ÁRBOL DE DECISIÓN





WEKA J4.8: INTERPRETAR EL ÁRBOL DE DECISIÓN



INTERPRETAR EL ÁRBOL DE DECISIÓN

Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

J48 -C 0.25 -M 2

Test options

Use training set

Supplied test set Set...

Cross-validation Folds 10

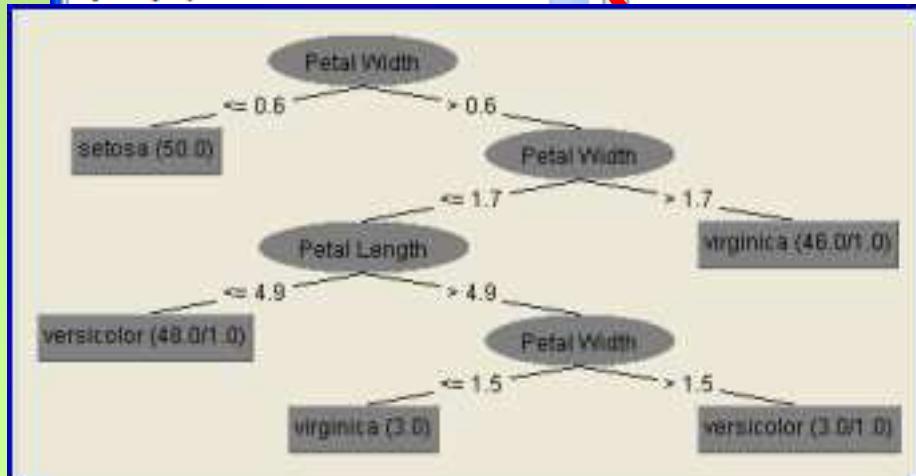
Percentage split /%

More options...

(Nom) Species

Classifier output

```
Petal Width <= 0.6: setosa (50.0)
Petal Width > 0.6
|   Petal Width <= 1.7
|   |   Petal Length <= 4.9: versicolor (48.0/1.0)
|   |   Petal Length > 4.9
|   |   |   Petal Width <= 1.5: virginica (3.0)
|   |   |   Petal Width > 1.5: versicolor (3.0/1.0)
|   |   Petal Width > 1.7: virginica (46.0/1.0)
```



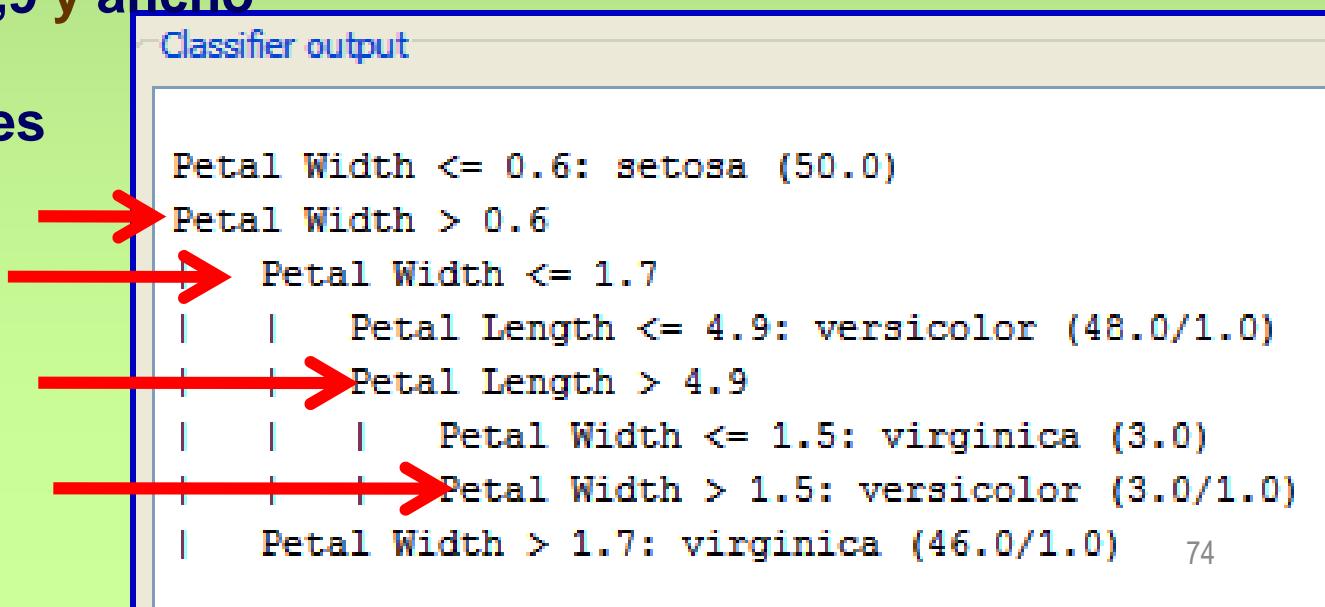
Árbol de decisión
como reglas.

INTERPRETAR EL ÁRBOL DE DECISIÓN

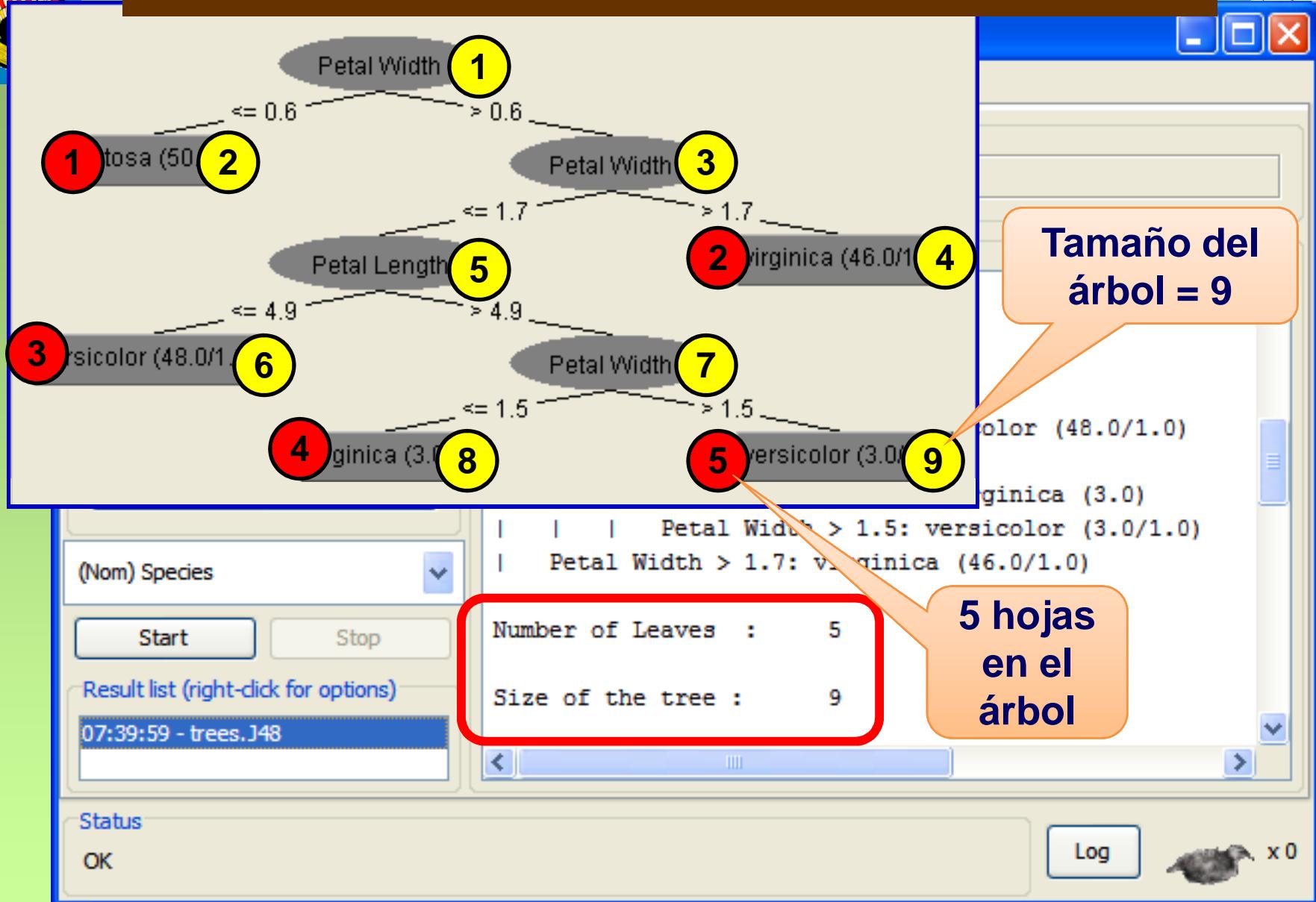


Lirio de especie desconocida:
Ancho de pétalo = 1.6 cm
Longitud de pétalo = 5.0 cm

Como ancho del petalo es > 0,6
y ancho del petalo es <= 1,7 y
longitud del petalo > 4,9 y ancho
de petalo es > 1,5
Entonces la clase iris es
versicolor



INTERPRETAR EL ÁRBOL DE DECISIÓN

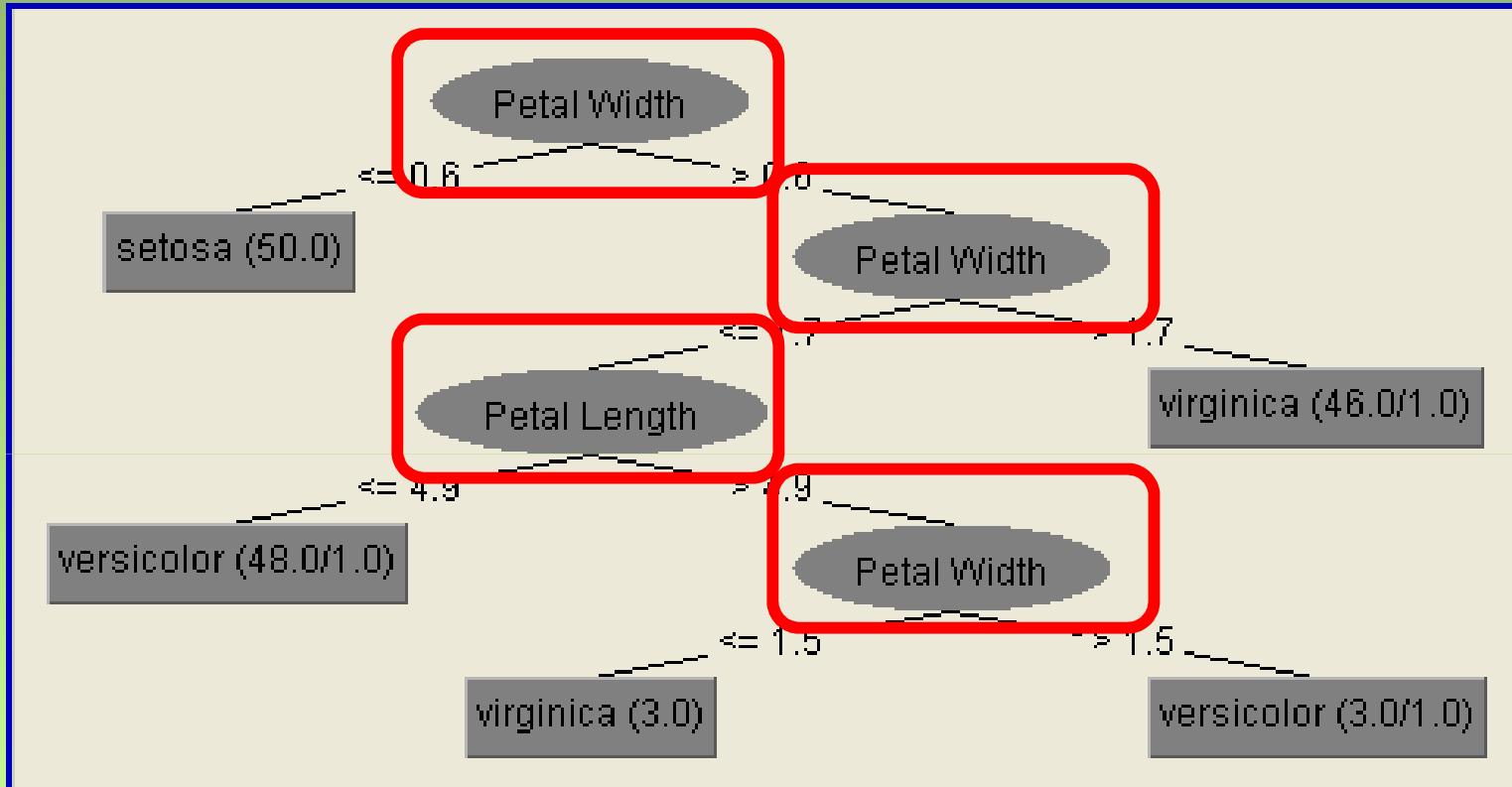




INTERPRETAR EL ÁRBOL DE DECISIÓN



Sólo necesitamos medir la longitud y el ancho del pétalo!

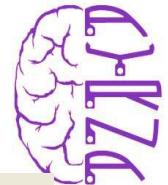


La meta es:

- Clasificar correctamente el mayor número posible de patrones del conjunto de test, esto es el CCR o precisión.
- Mientras minimizamos el tamaño del arból y el número de hojas.



INTERPRETAR LA SALIDA DEL ALGORITMO



Classifier output

== Evaluation on test split ==

== Summary ==

Correctly Classified Instances	49	96.0784 %
Incorrectly Classified Instances	2	3.9216 %
Kappa statistic	0.9408	
Mean absolute error	0.0396	
Root mean squared error	0.1579	
Relative absolute error	8.8979 %	
Root relative squared error	33.4091 %	
Total Number of Instances	51	

El árbol clasifica bien 49 de las 51 muestras del conjunto de test.

51 muestras del conjunto de test (el 34%)



INTERPRETAR LA SALIDA DEL ALGORITMO



MATRIZ DE CONFUSIÓN

Classifier output

==== Confusion Matrix ===

a	b	c	<-- classified as
15	0	0	a = setosa
0	19	0	b = versicolor
0	2	15	c = virginica

Los 15 patrones de la clase setosa los clasifica correctamente

Los 19 patrones de la clase versicolor los clasifica correctamente

2 patrones de la clase virginica del conjunto de test los clasifica incorrectamente



INTERPRETAR LA SALIDA DEL ALGORITMO



Classifier output

== Predictions on test split ==

inst#	actual	predicted	error	probability distribution
1	2:versicol	2:versicol	0	*0.968 0.032
2	3:virginic	3:virginic	0	0.032 *0.968
3	2:versicol	2:versicol	0	*0.968 0.032
4	1:seto	1:seto	*1	0 0
5	3:virgi	3:virgi	0	0.032 *0.968
6	2:versi	2:versi	0	*0.968 0.032
7	1:seto	1:seto	*1	0 0
8	1:setosa	1:setosa	*1	0 0
9	2:versicol	2:versicol	0	*0.968 0.032
10	3:virginic	3:virginic	0	0.032 *0.968
11	2:versicol	2:versicol	0	*0.968 0.032
12	3:virginic	3:virginic	0	0.032 *0.968
13	2:versicol	2:versicol	*1	0 0
14	2:versicol	2:versicol	0	*0.968 0.032
15	2:versicol	2:versicol	0	*0.968 0.032
16	3:virginic	2:versicol	+	*0.968 0.032

Clase predicha
para este patrón

patrón
incorrectamente
clasificado.



INTERPRETAR LA SALIDA DEL ALGORITMO



Pulsando botón derecho, sale....

Pulsar botón izquierdo

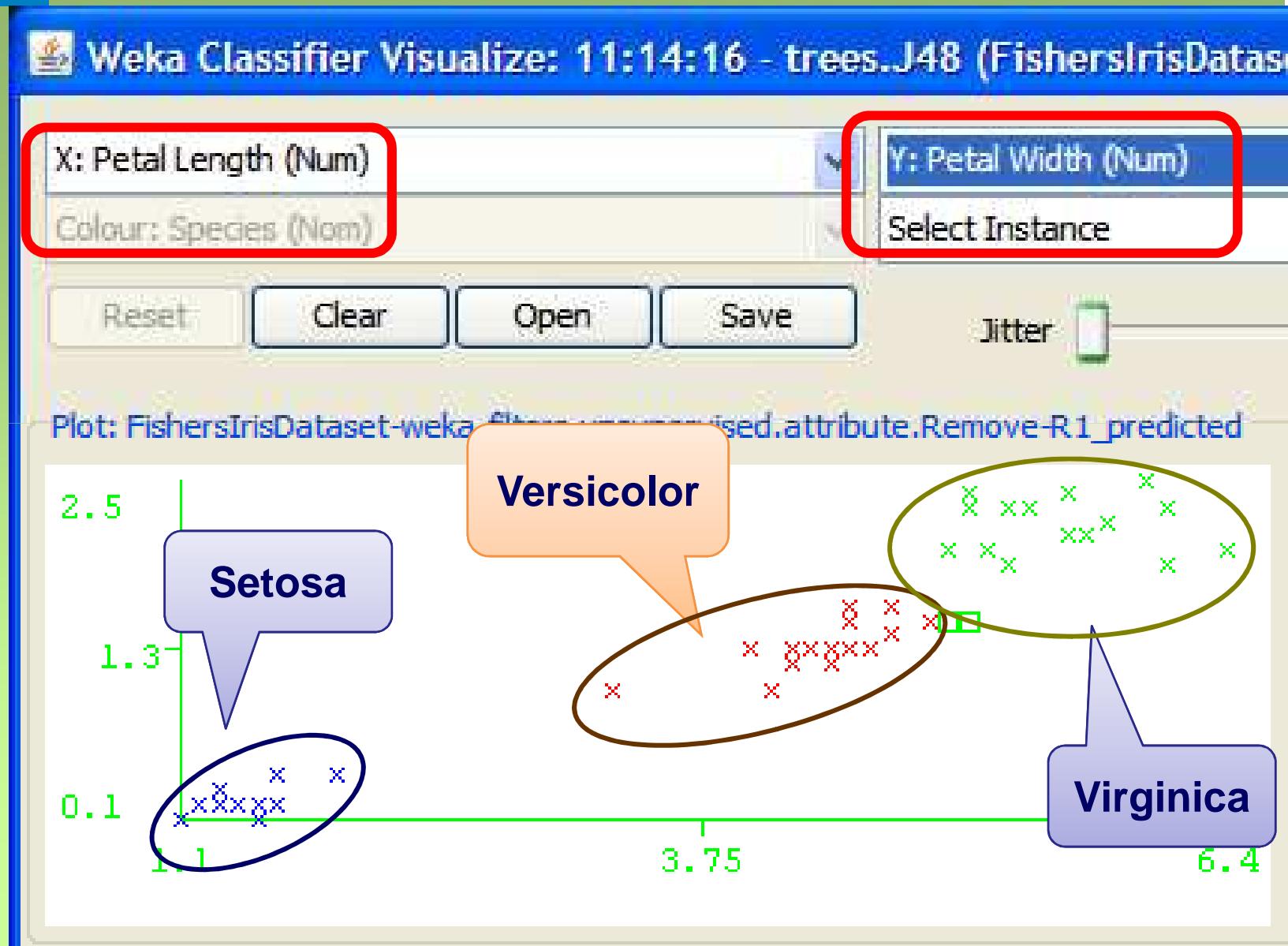
The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'J48 -C 0.25 -M 2' is chosen. Under 'Test options', 'Percentage split' is selected with 66% chosen. The 'Classifier output' pane displays various error metrics and detailed accuracy. A context menu is open over a list item '11:14:16 - trees.J48'. The menu items are:

- View in main window
- View in separate window
- Save result buffer
- Delete result buffer
- Load model
- Save model
- Re-evaluate model on current test set
- Visualize classifier errors** (highlighted with a red box and arrow)
- Visualize ROC
- Visualize margin curve
- Visualize threshold curve
- Visualize cost curve

The status bar at the bottom left shows 'Status OK'.



INTERPRETAR LA SALIDA DEL ALGORITMO





INTERPRETAR LA SALIDA DEL ALGORITMO



2.5

Ancho de petalo versus largo de petalo

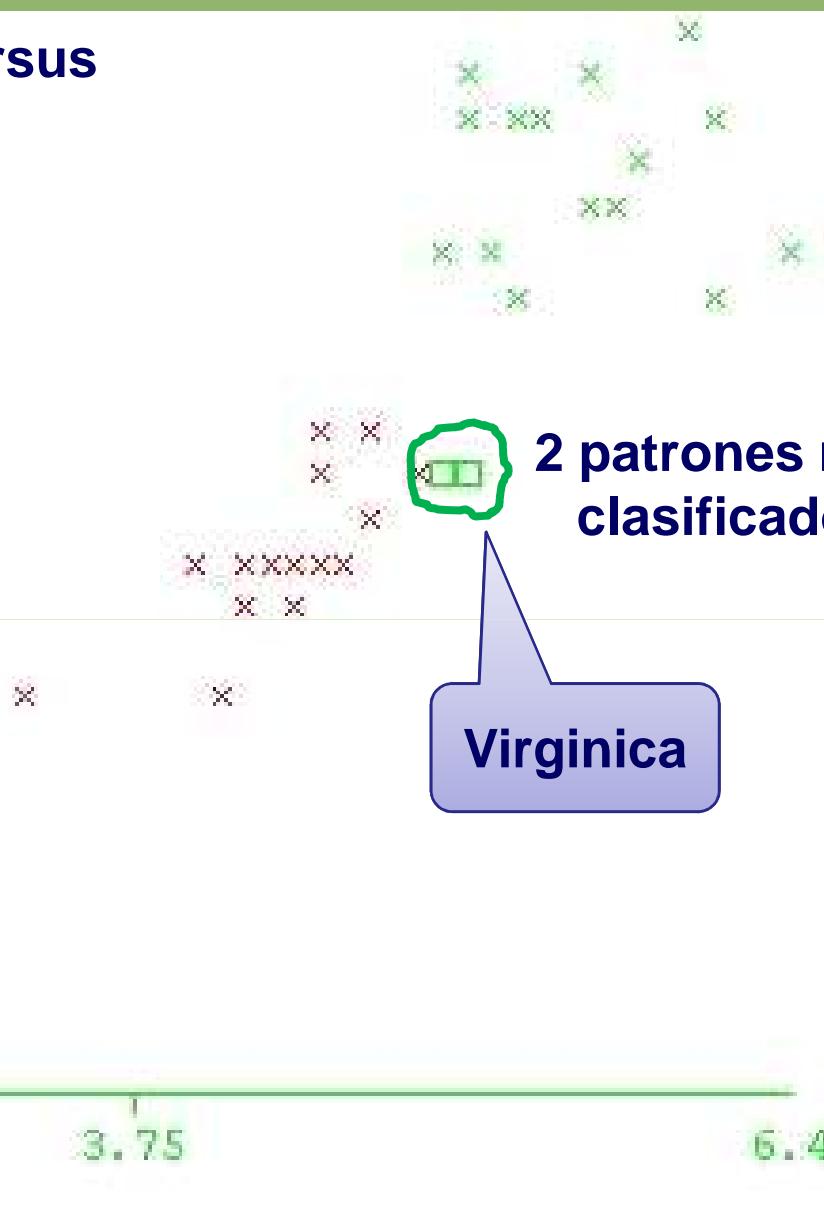
1.3

0.1

1.1

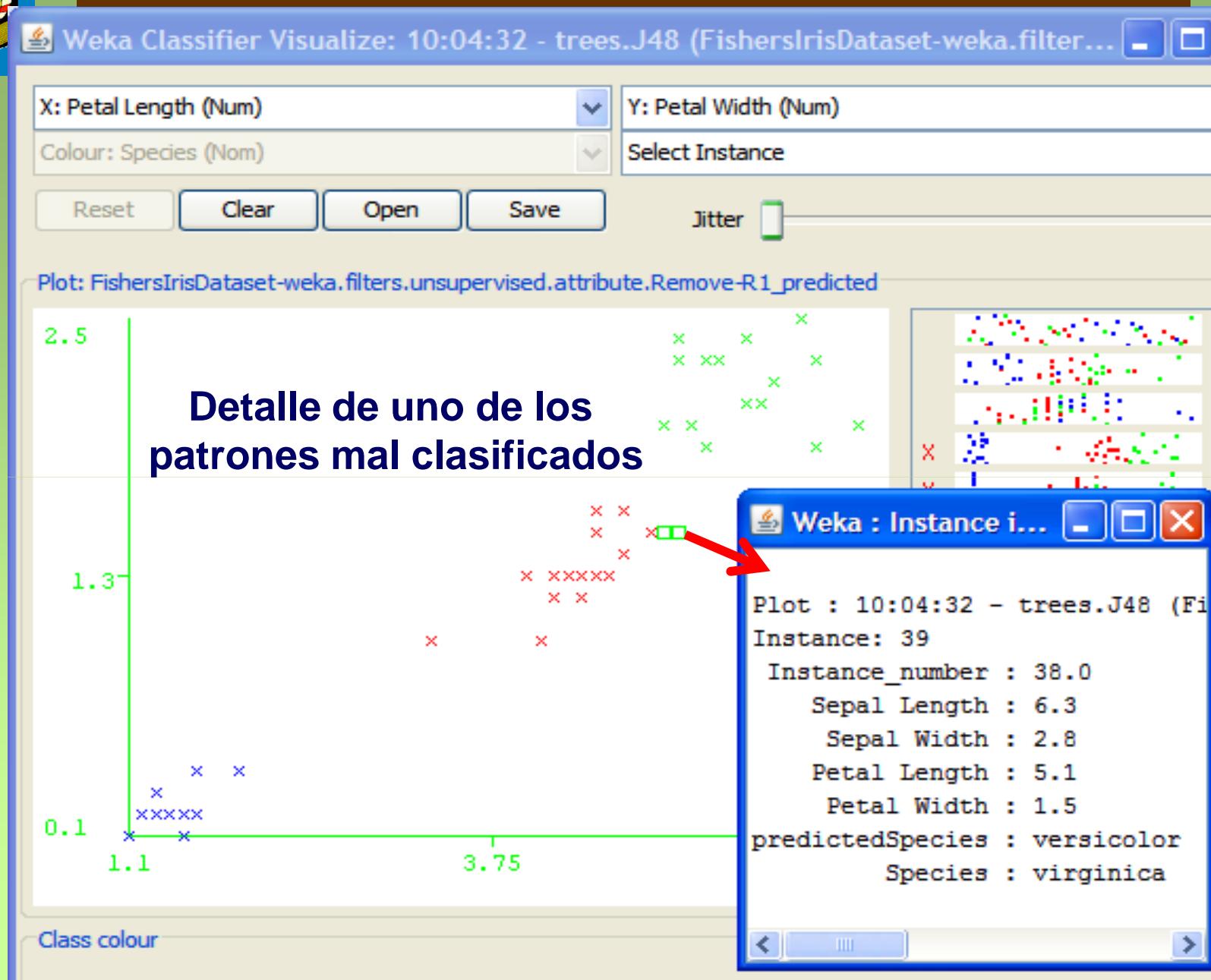
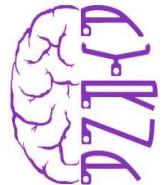
3.75

6.4



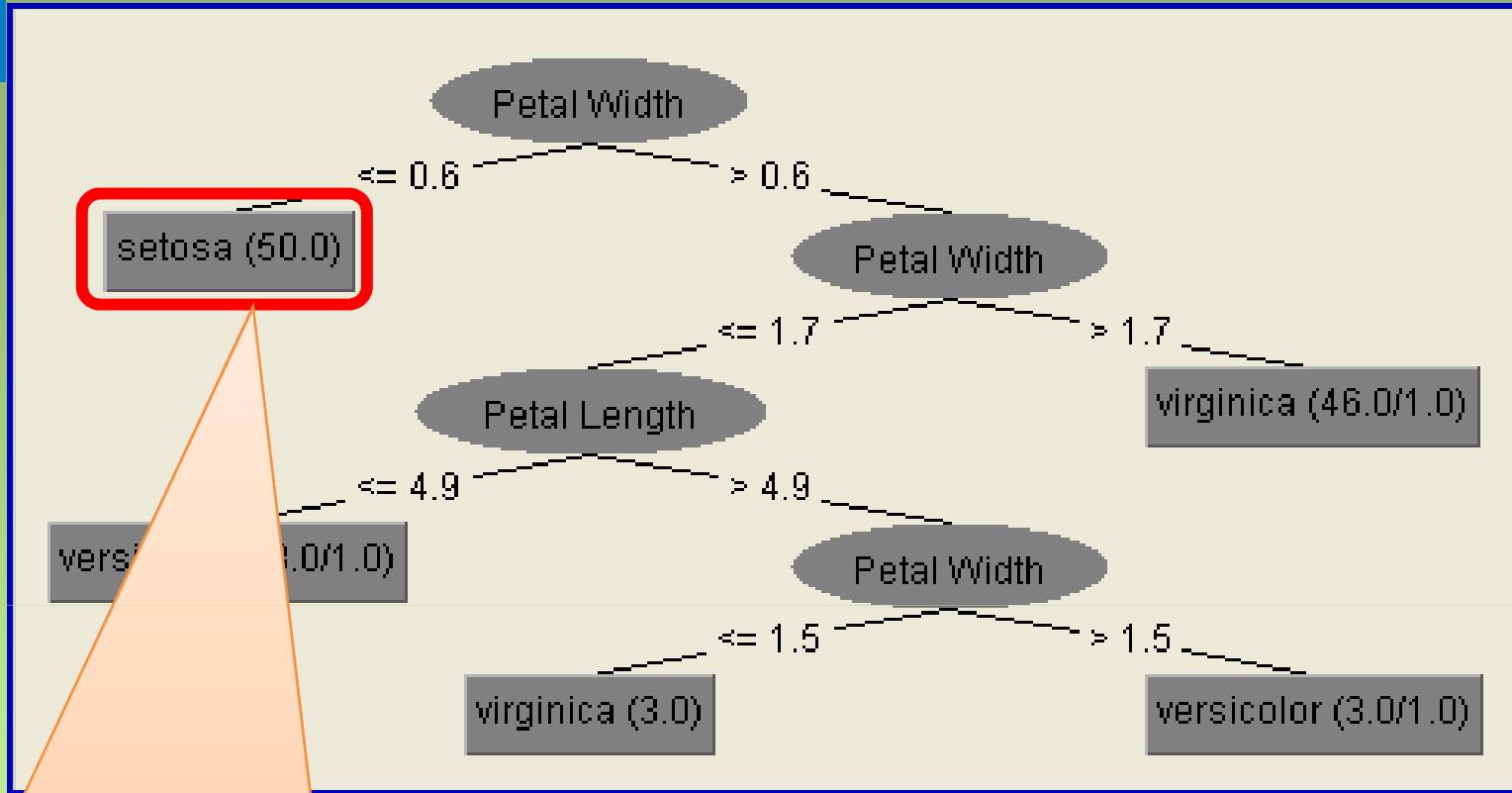


INTERPRETAR LA SALIDA DEL ALGORITMO





INTERPRETAR LA SALIDA DEL ALGORITMO



50 de las 150 muestras alcanzan esta hoja en el árbol.

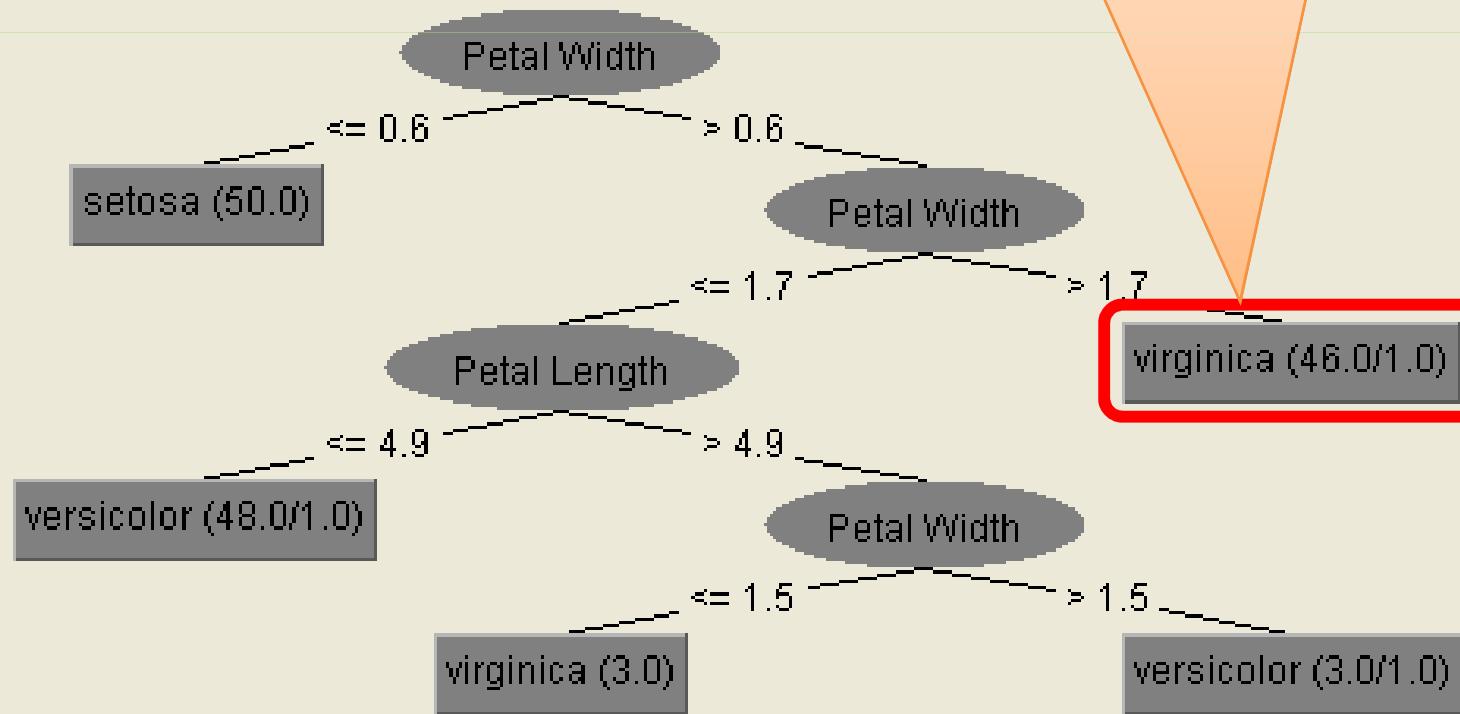
Puesto que existen 50 muestras de setosa en los datos originales, todos los patrones de setosa estarán correctamente clasificados.



--- Confusion Matrix ---

a	b	c	<-- classified as
15	0	0	a = setosa
0	19	0	b = versicolor
0	2	15	c = virginica

46 patrones llegaron a esta hoja de virginica. 45 eran virginica, pero 1 no

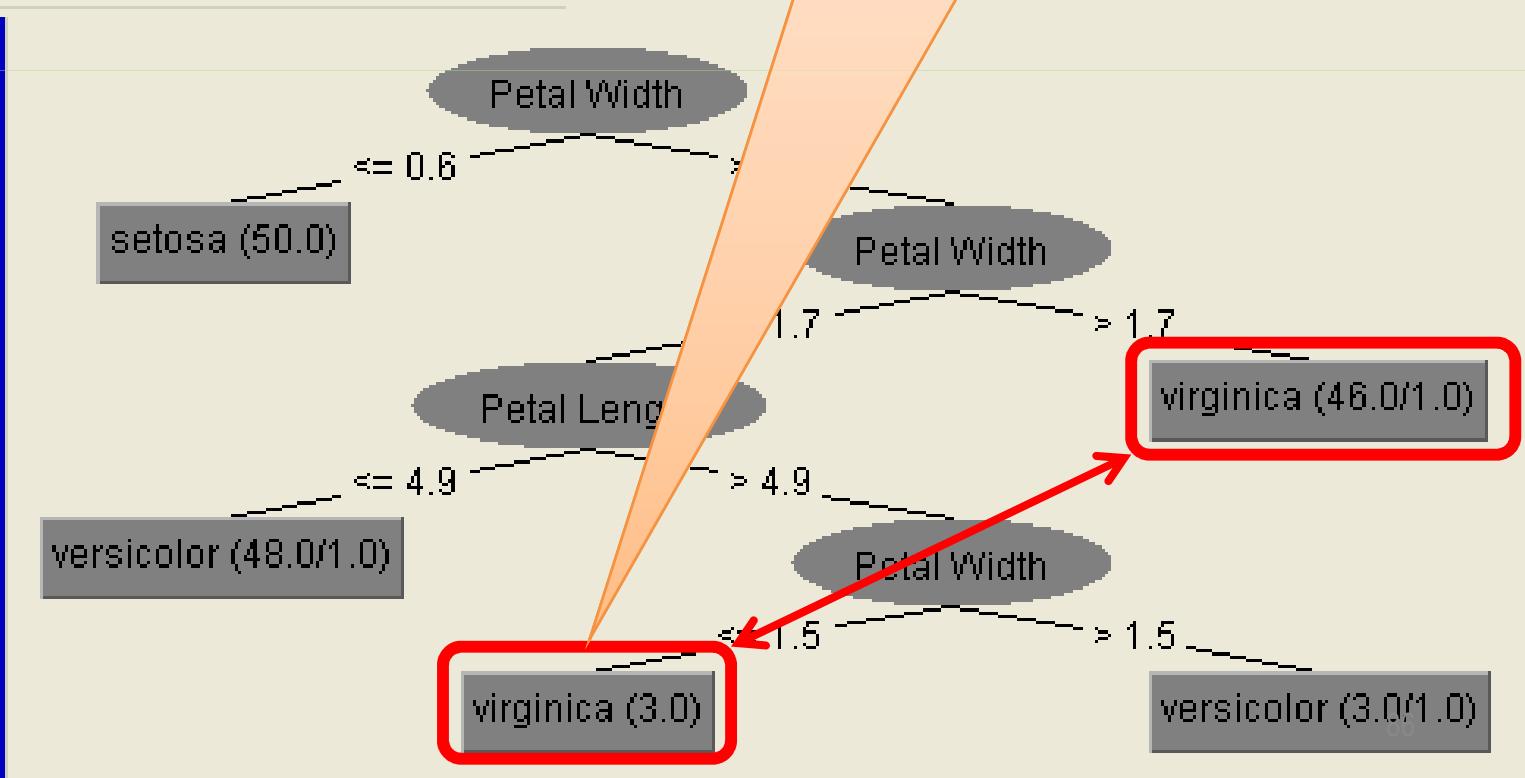




--- Confusion Matrix ---

a	b	c		<-- classified as
15	0	0		a = setosa
0	19	0		b = versicolor
0	2	15		c = virginica

3 patrones llegaron a esta hoja de virgínica. Todos son virgínica.

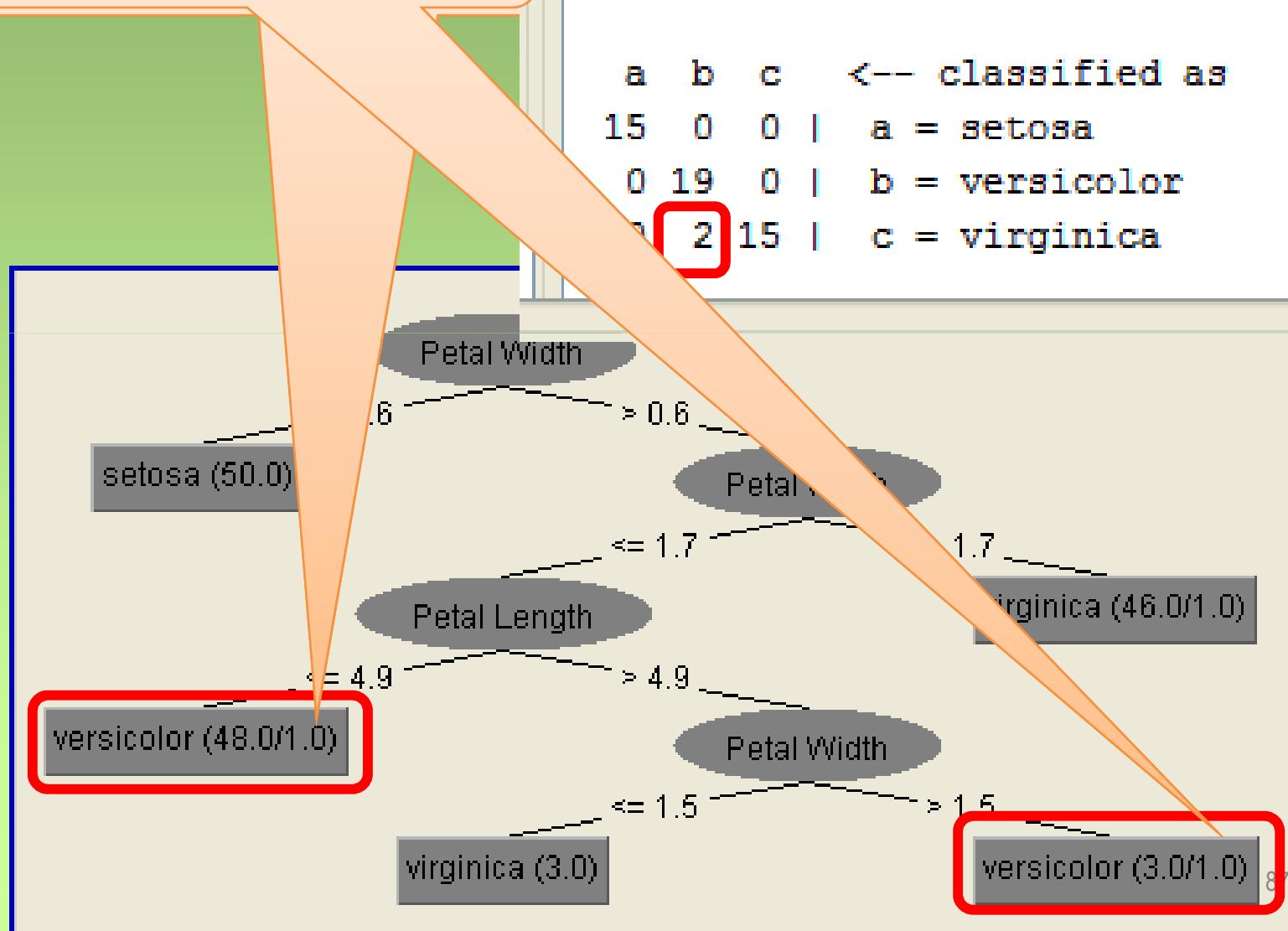




INTERPRETAR LA SALIDA DEL ALGORITMO



2 patrones de virginica del conjunto de test mal clasificados como versicolor

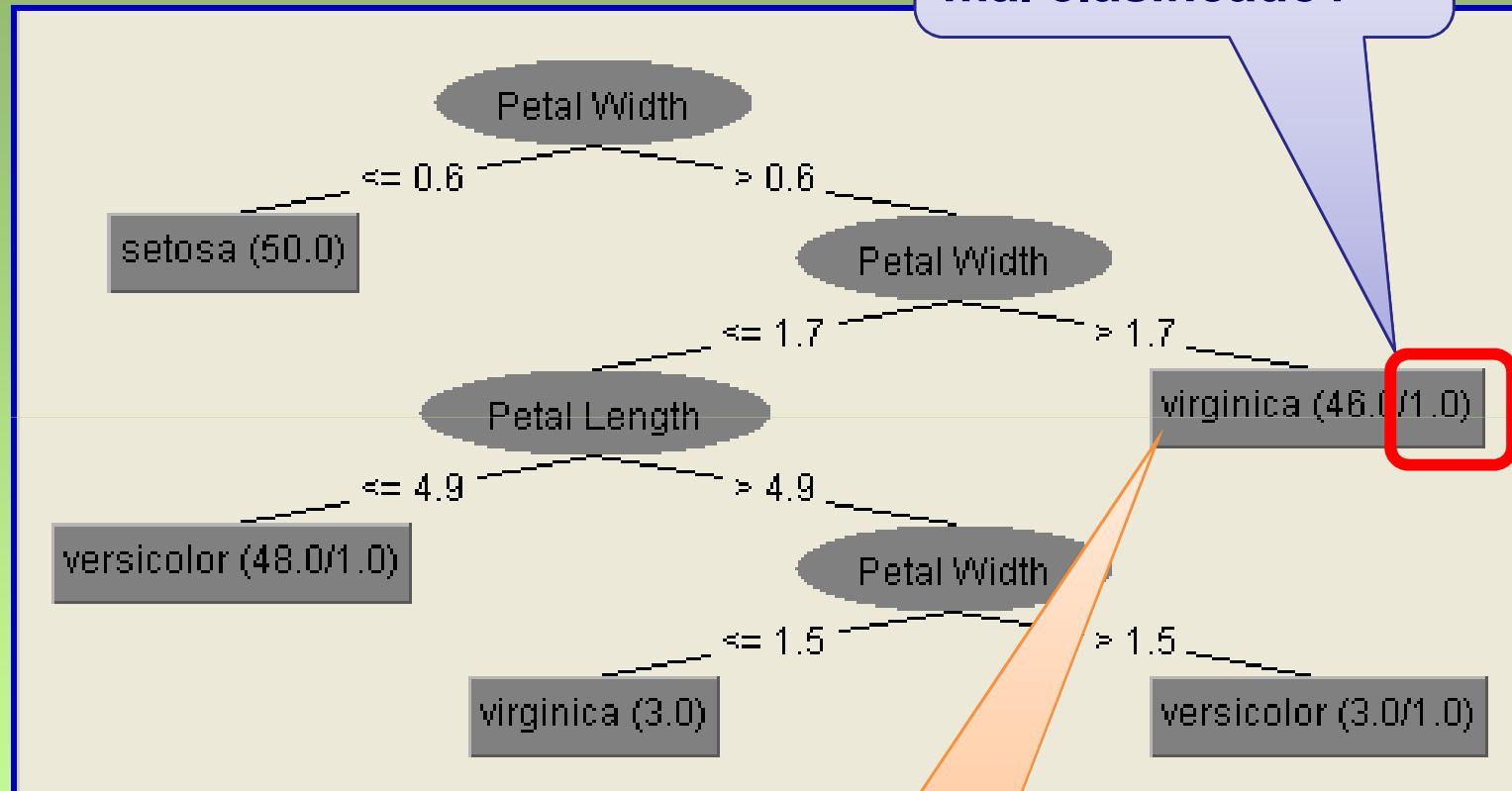




INTERPRETAR LA SALIDA DEL ALGORITMO



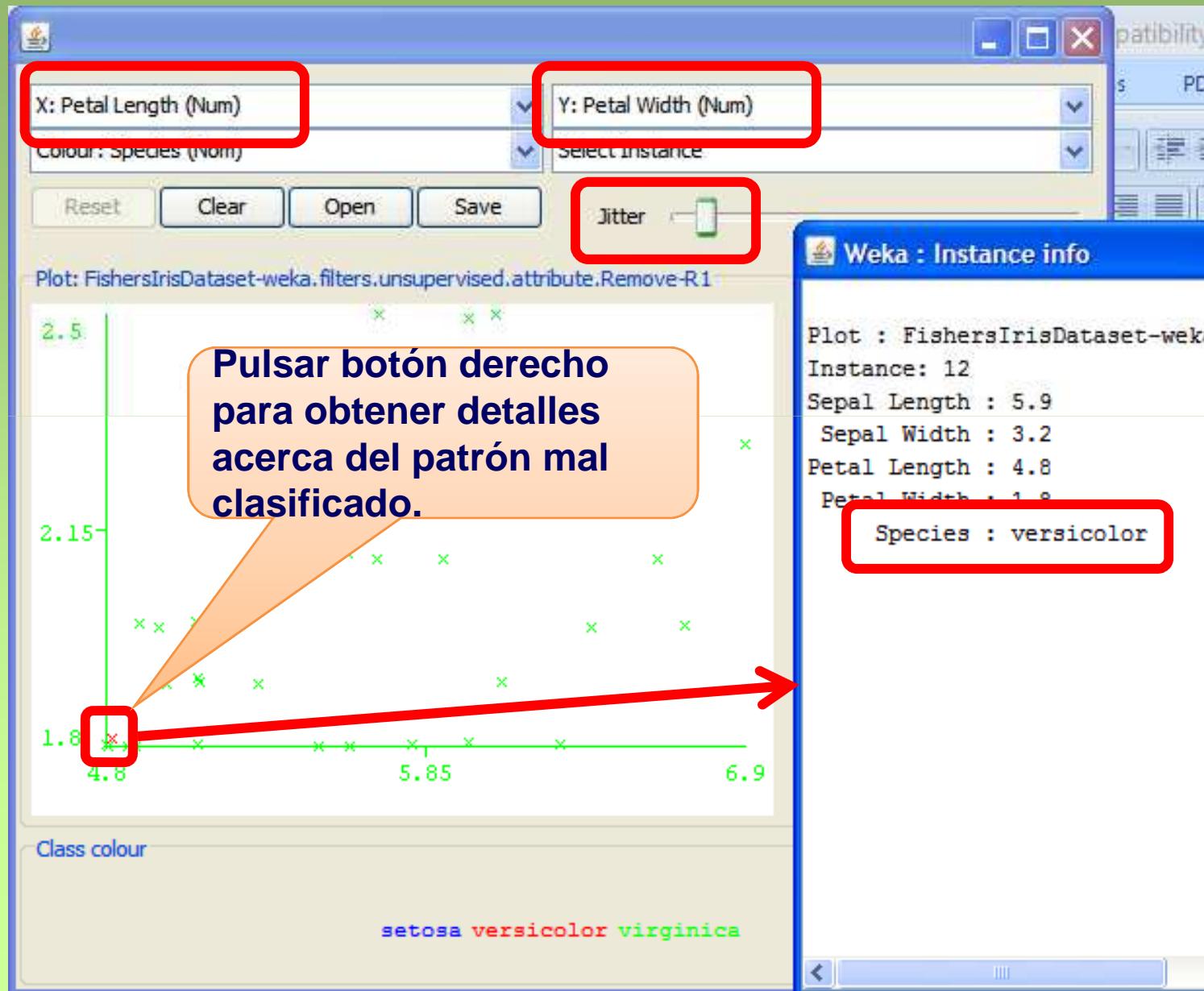
Que es este patrón
mal clasificado?



Pulsar botón
derecho
sobre la hoja.



INTERPRETAR LA SALIDA DEL ALGORITMO





INTERPRETAR LA SALIDA DEL ALGORITMO



Resumen de la salida del árbol de decisión para los datos de entrenamiento y test.

		Predicted class		
		setosa	versicolor	virginica
Actual class	setosa	50	0	0
	versicolor	0	49	1
	virginica	0	2	48

Como de bueno es nuestro árbol comparado con otros clasificadores?

Porque nuestro árbol obtiene este dato mal?



Por qué nuestro árbol clasifica algunas muestras de forma incorrecta?



1. Errores en la medida de los atributos,

p. e., medidas inapropiadas de pétalos y sépalos.

2. Errores en la identificación de las clases de las etiquetas,

p.e. Algunas muestras de setosa clasificadas como versicolor, etc.

3. Muestras de tipo “Outlier”,

p.e., algunas muestras de flores raquílicas por la sequía.

4. Conjunto atípico de muestras,

p.e., muestras en el conjunto de entrenamiento que siendo amantes del sol son recogidas en una parte umbría.

5. Utilizar un algoritmo de clasificación inapropiado.

El algoritmo de aprendizaje mediante árboles de decisión no funciona bien para los lirios!



Estadísticos utilizados para evaluar y comparar el rendimiento del clasificador.



Classifier output

==== Evaluation on test split ====

==== Summary ====

Correctly Classified Instances	49	96.0784 %
Incorrectly Classified Instances	2	3.9216 %
Kappa statistic	0.9408	
Mean absolute error	0.0396	
Root mean squared error	0.1579	
Relative absolute error	8.8979 %	
Root relative squared error	33.4091 %	
Total Number of Instances	51	

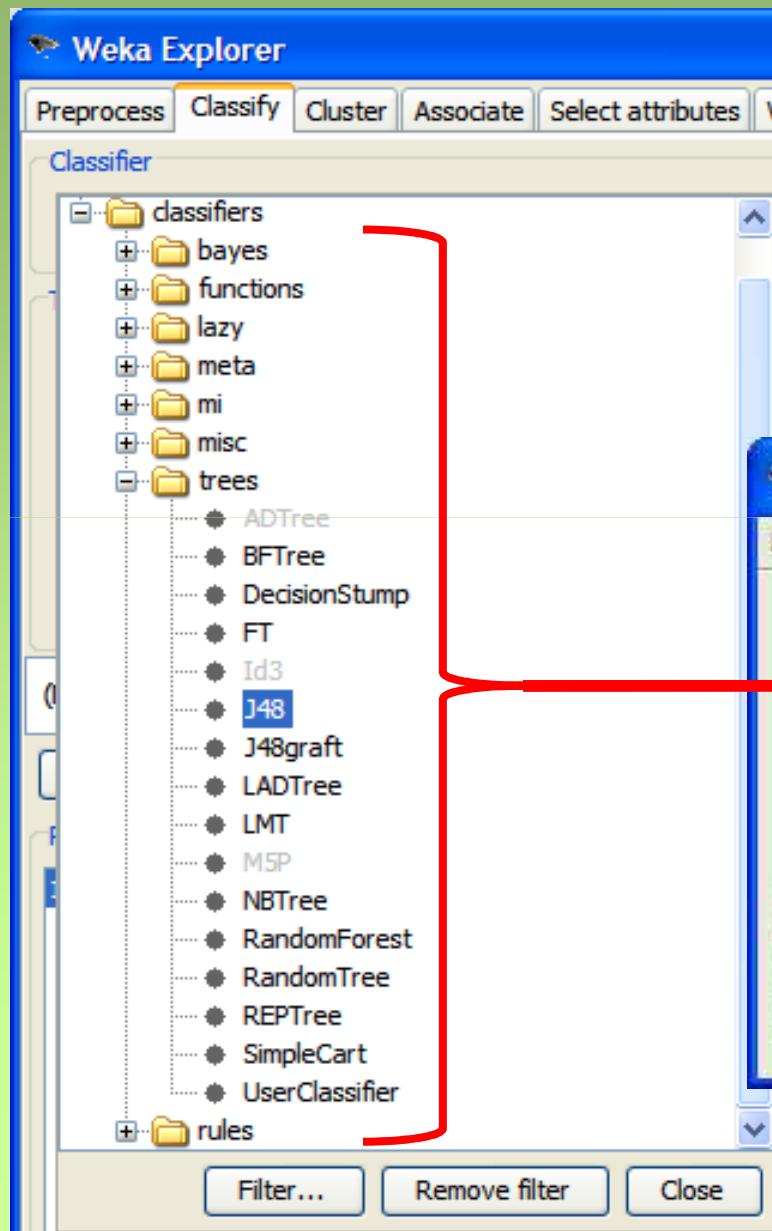
Classifier output

==== Detailed Accuracy By Class ====

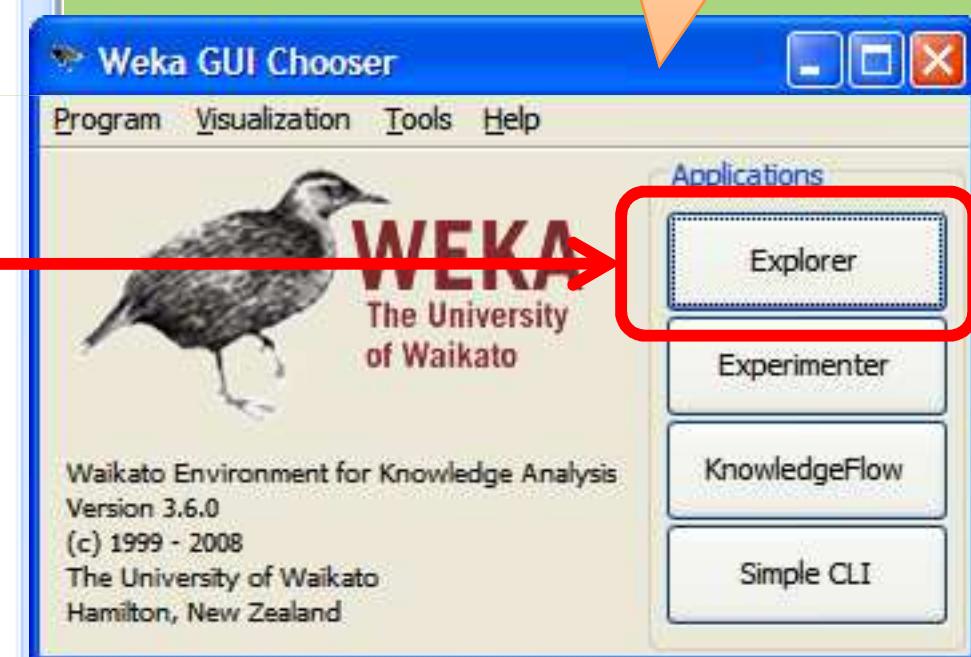
	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
setosa	1	0	1	1	1	1	setosa
versicolor	1	0.063	0.905	1	0.95	0.969	versicolor
virginica	0.882	0	1	0.882	0.938	0.967	virginica
Weighted Avg.	0.961	0.023	0.965	0.961	0.961	0.977	



EXPLORER DE WEKA

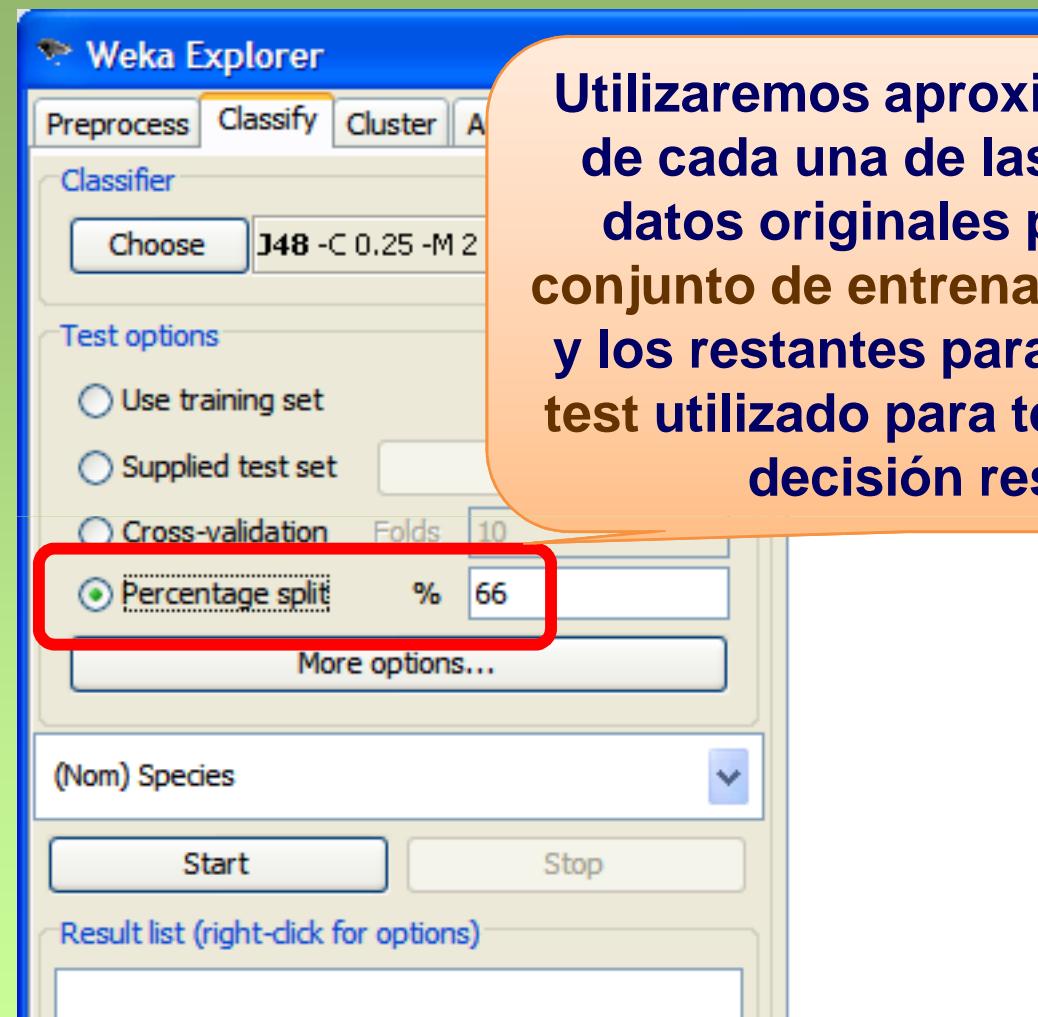


El Explorer de Weka nos permite comparar algoritmos de minería de datos.





OPCIÓN DE PORCENTAJE DE PATRONES PARA ENTRENAMIENTO (holdout)



Utilizaremos aproximadamente 2/3 de cada una de las clases de los datos originales para formar el conjunto de entrenamiento del árbol y los restantes para el conjunto de test utilizado para testar el árbol de decisión resultante



ARFF-Viewer - C:\Docume

File Edit View

FishersIrisDataset.arff

Relation: FishersIrisDataset-weka.filters.unsupervised.attribute.Relation

No.	Sepal Length Numeric	Sepal Width Numeric	Petal Length Numeric	Petal Width Numeric	Species
48	4.6	3.2	1.4	0.2	setosa
49	5.3	3.7	1.5	0.2	setosa
50	5.0	3.3	1.4	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1		1.5	versicolor

Si longitud de sépalo = 5,0
y ancho de sépalo = 3,3
y longitud de pétalo = 1,4
y ancho de pétalo = 0,2
entonces lirio= setosa

Si longitud de sépalo = 7,0
y ancho de sépalo = 3,2
y longitud de pétalo = 4,7
y ancho de pétalo = 1,4
entonces lirio= versicolor



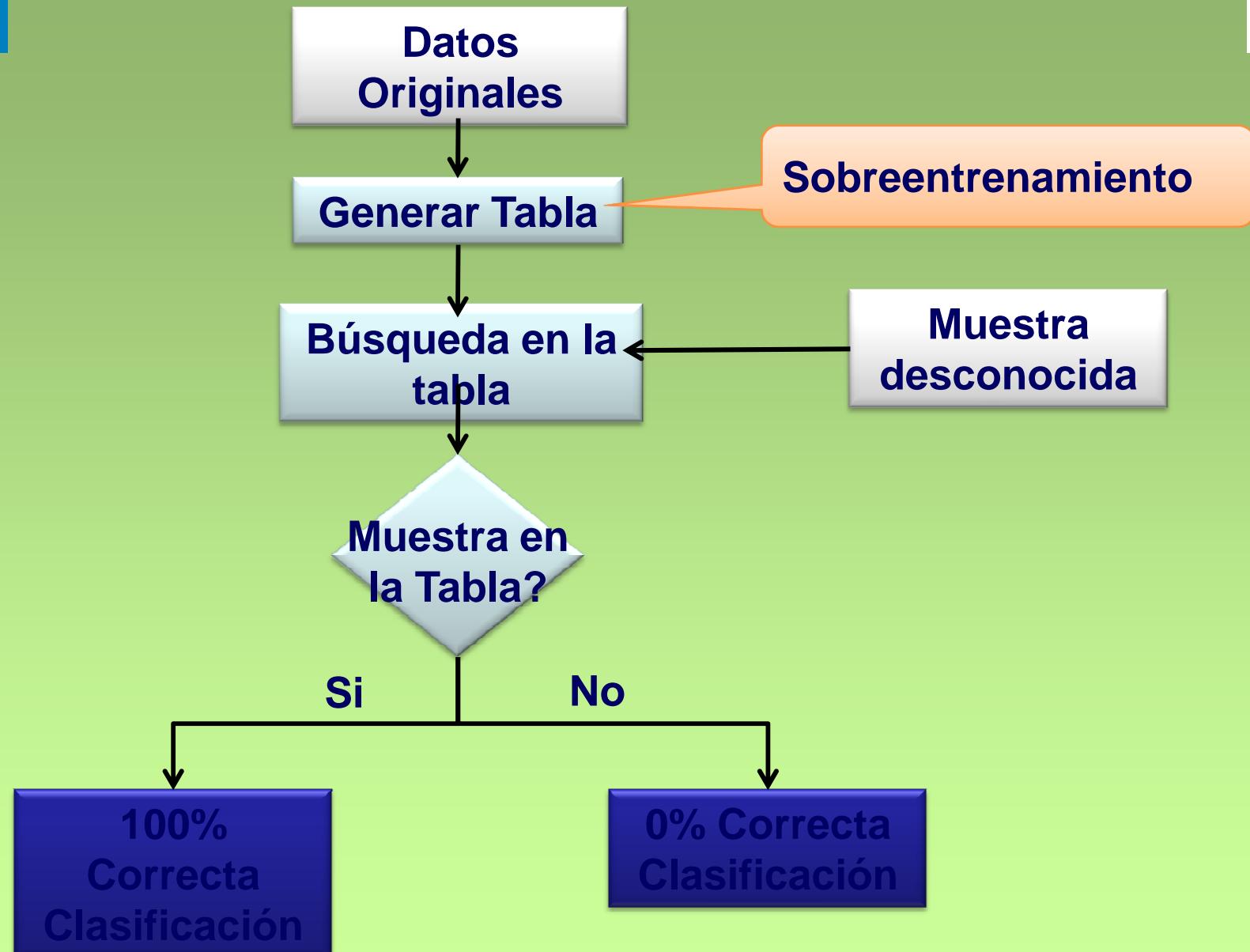
CLASIFICACIÓN



Inst.	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
16	5.7	4.4	1.5	0.4	setosa
93	5.8	2.6	4.0	1.2	versicolor
83	5.8	2.7	3.9	1.2	versicolor
68	5.8	2.7	4.1	1.0	versicolor
102	5.8	2.7	5.1	1.9	virginica
143	5.8	2.7	5.1	1.9	virginica
115	5.8	2.8	5.1	2.4	virginica
15	5.8	4.0	1.2	0.2	setosa
62	5.9	3.0	4.2	1.5	versicolor

Como podemos
clasificar un patrón con
estas características

sepal length = 7.0
sepal width = 3.2
petal length = 4.7
petal width = 1.4





1 Introducción

2 Algoritmo básico: ID3

3 Mejoras a ID3

4 C4.5

5 Tutorial de Weka

6 Conclusiones



METODOLOGÍA DE CLASIFICACIÓN





Resumen



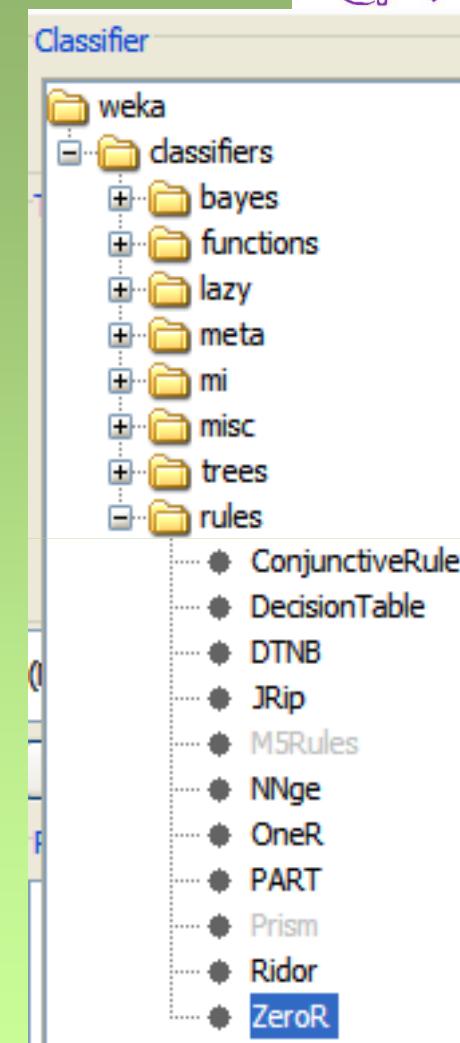
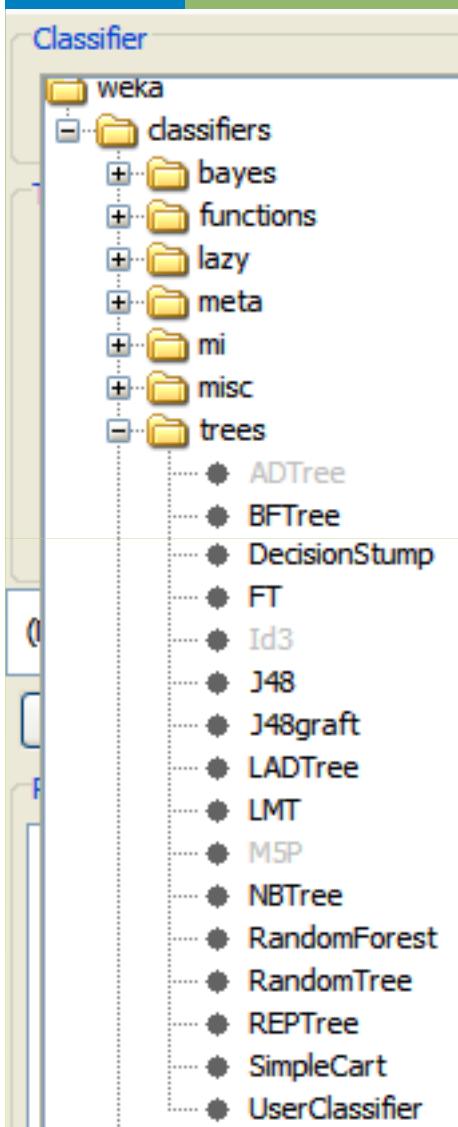
- En Weka están disponibles una amplia variedad de herramientas de minería de datos

Cada vez es más y más fácil de encontrar modelos para describir los datos.

Cada vez es más y más fácil obtener modelos de sobre-ajuste!

Cada vez es menos importante como de bien un modelo describe los datos de entrenamiento.

Lo que importa es lo bien que describe los datos de test.





Árboles de regresión y árboles de modelos Variable de clase continua (predicción numérica)



Generalizar estas ideas para obtener funciones de regresión continuas

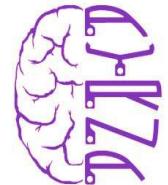
Árbol de regresión: en las hojas el valor representa el valor medio de la respuesta de las instancias que llegan a esa hoja

Es lo que utiliza el sistema CART [Breiman et al., 1984]. Aproxima la función objetivo mediante una función constante a trozos. Sirve para cualquier tipo de atributos

Árbol de modelos de regresión: en las hojas hay un modelo de regresión lineal para predecir la respuesta de las instancias que llegan a esa hoja

Algoritmos M5 [Quinlan, 1992] y M5' [Wang y Witten, 1997] Todo tipo de atributos, especialmente continuos.

Los árboles de regresión son un caso particular de árboles de modelos de regresión



Documención de Weka :

Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)

Ian H. Witten, Eibe Frank

Morgan
Kaufmann
June 2005 →
525 pages
Paper
ISBN 0-12-
088407-0



Eibe Frank and Ian Witten



[Click here to order from Amazon.com](#)



APRENDIZAJE AUTOMÁTICO: TERCER CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

Clasificación: Árboles de Decisión

GRACIAS POR SU ATENCIÓN

César Hervás-Martínez
Grupo de Investigación AYRNA

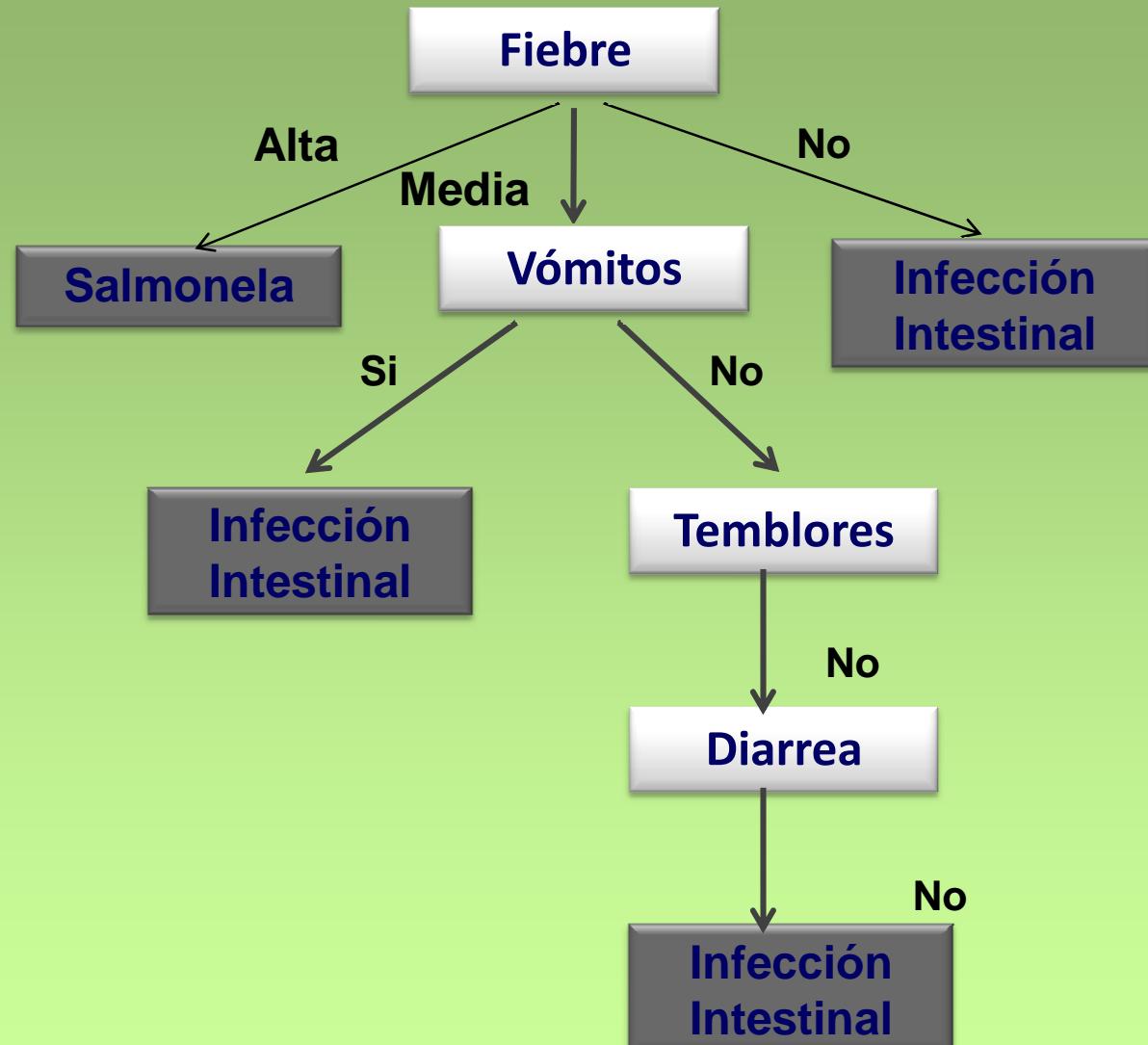
Departamento de Informática y Análisis
Numérico
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es

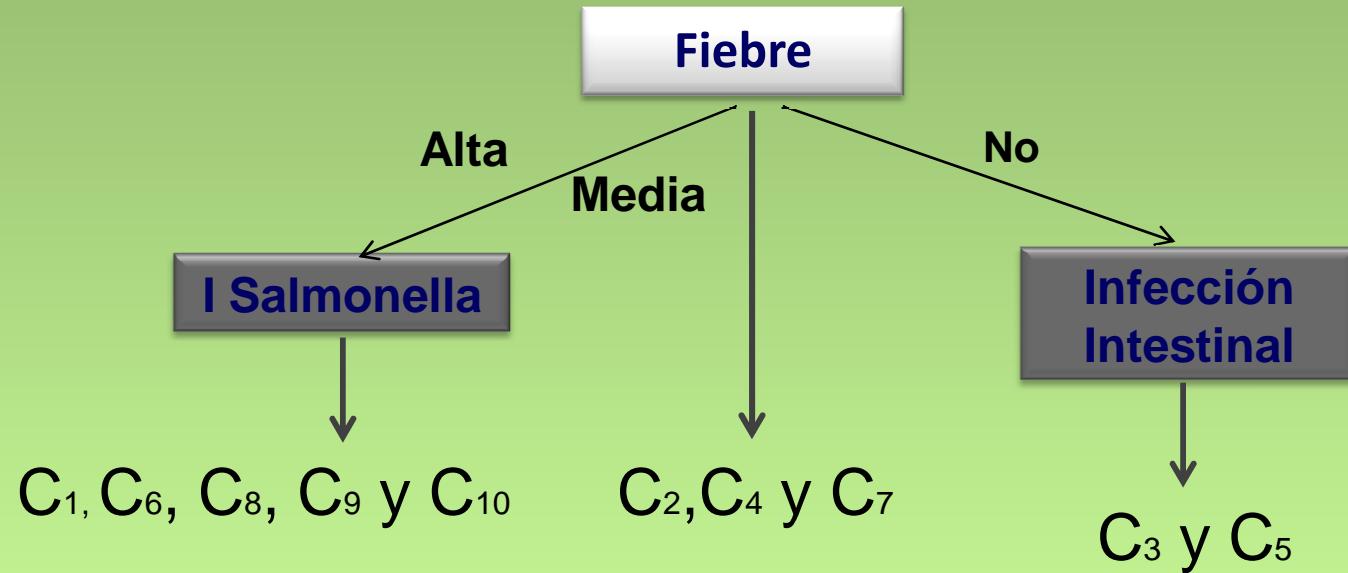
2019-2020
103

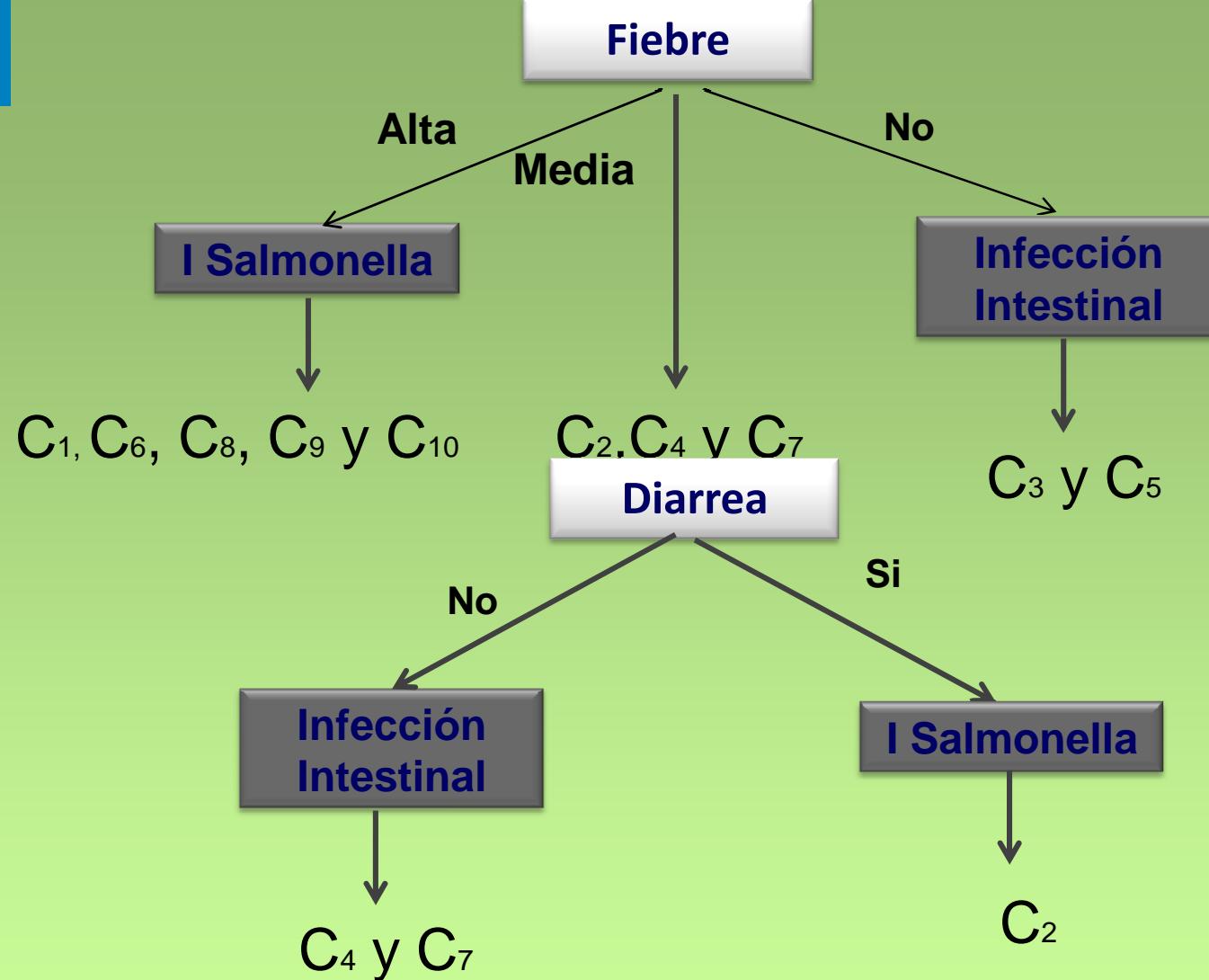
103

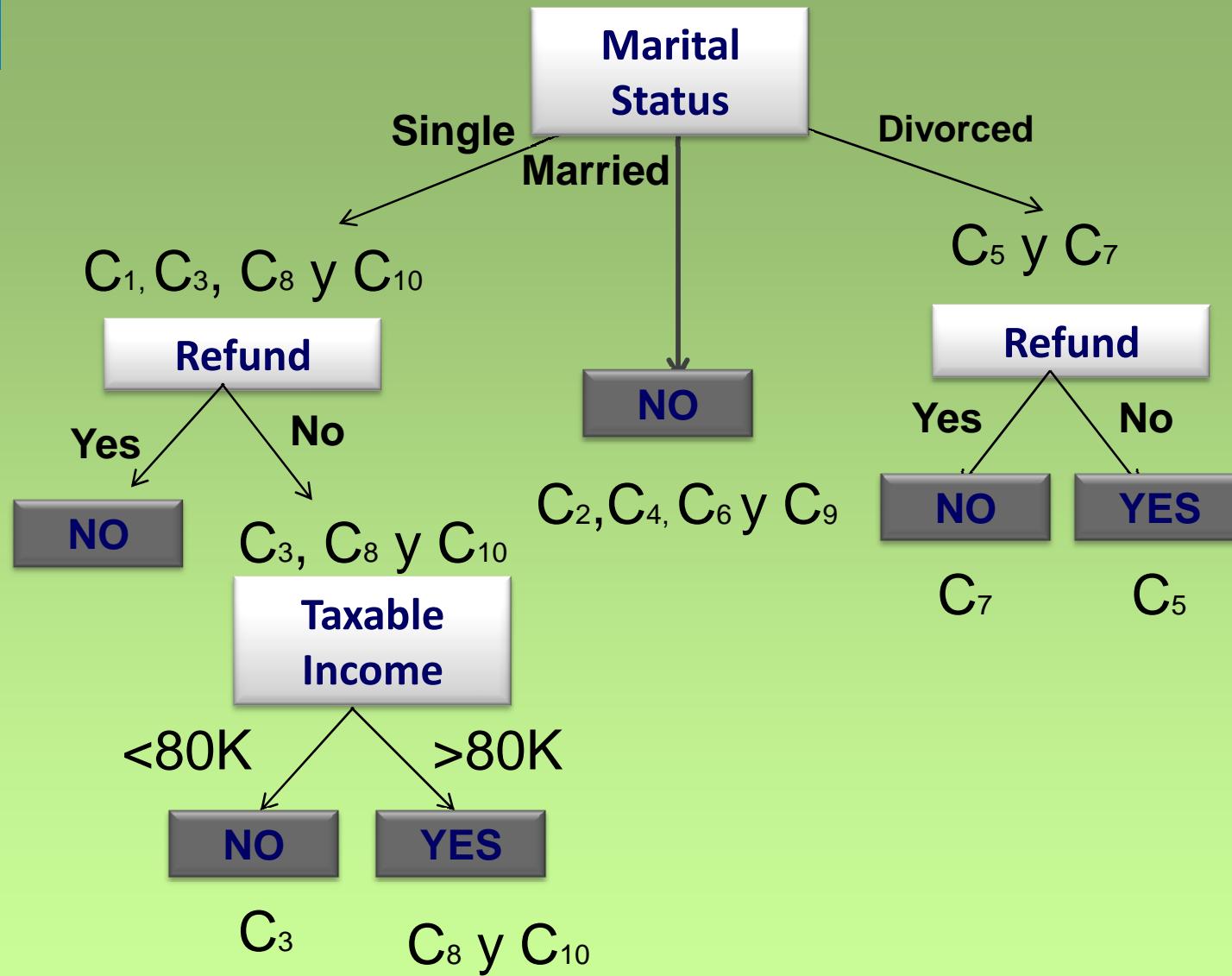


ÁRBOL DE DECISIÓN PROBABILISTICO











ÁRBOL DE DECISIÓN PROBABILISTICO

