Universidad Nacional Autónoma de México
Facultad de Ciencias
Almacenes y Minería de Datos

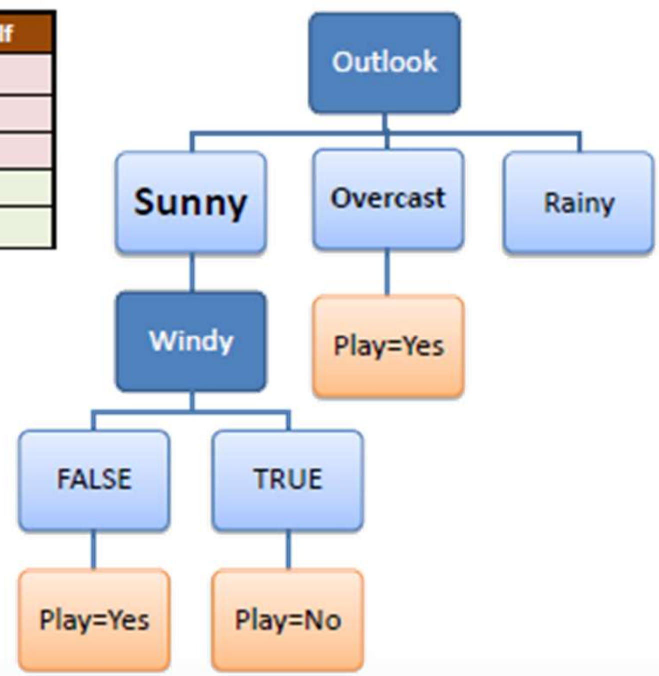
Árboles de decisión: ID3 y C4.5

Gerardo Avilés Rosas
gar@ciencias.unam.mx

Introducción

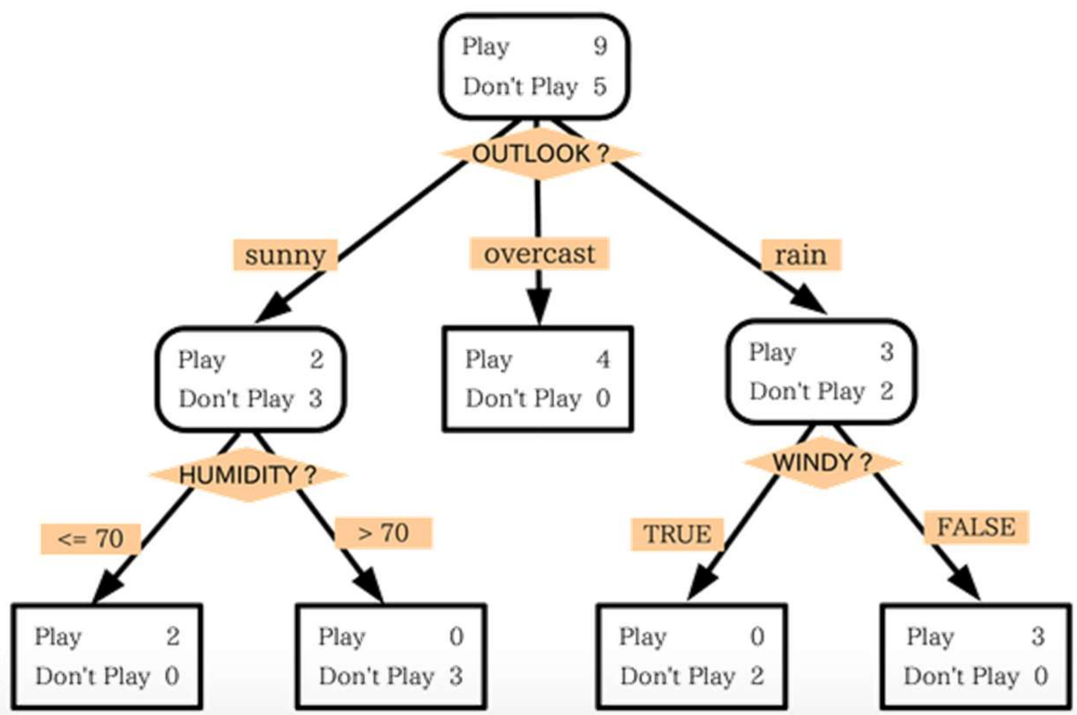
- Los **árboles de decisión** son una de las opciones **más populares** para aprender sobre características basadas en ejemplos. Han sido objeto de varias modificaciones para hacer frente a consideraciones lingüísticas, requisitos de memoria y de eficiencia.
- Se trata de un esquema de clasificación que genera un **árbol** y **un conjunto de reglas**, que representa el modelo de diferentes clases, a partir de un conjunto de datos dado.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Inducción de Árboles de Decisión

- Esta técnica permite que los **árboles de decisión aprendan** de un conjunto de **tuplas de entrenamiento** con **etiquetas de clase**.
- Es una especie de **diagrama de flujo** que tiene la estructura de un árbol, donde cada ***nodo interno*** denota una **prueba** sobre un atributo y ***cada rama*** representa el **resultado** de una prueba y cada ***nodo hoja*** almacena una **etiqueta de clase**.

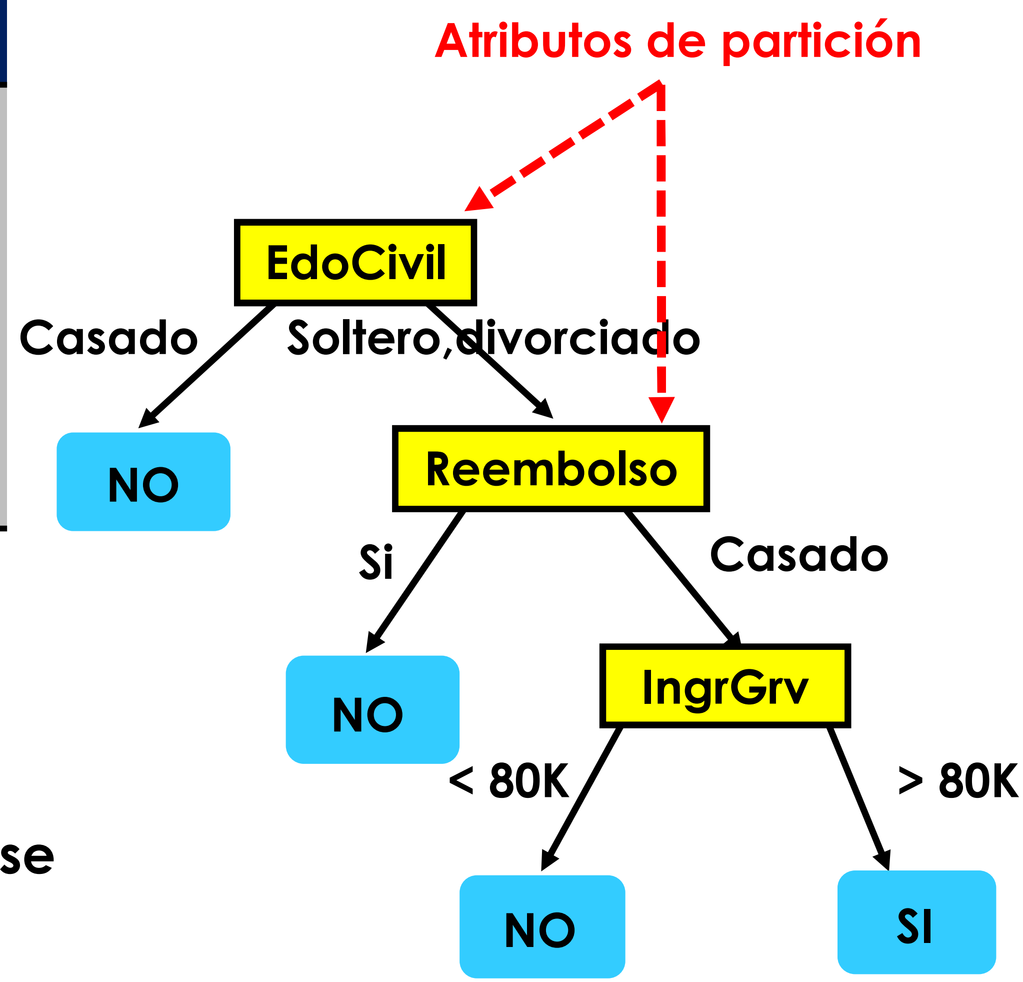


…Inducción de Árboles de Decisión

categórica categórica continua clase

id	Reembolso	Estado civil	Ingreso gravable	¿Engaña?
1	Si	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Si	Casado	120K	No
5	No	Divorciado	95K	Si
6	No	Casado	60K	No
7	Si	Divorciado	220K	No
8	No	Soltero	85K	Si
9	No	Casado	75K	No
10	No	Soltero	90K	Si

Tuplas de entrenamiento



Modelo: Árbol de decisión

¿Podría haber más de un árbol que se ajuste a los mismos datos?

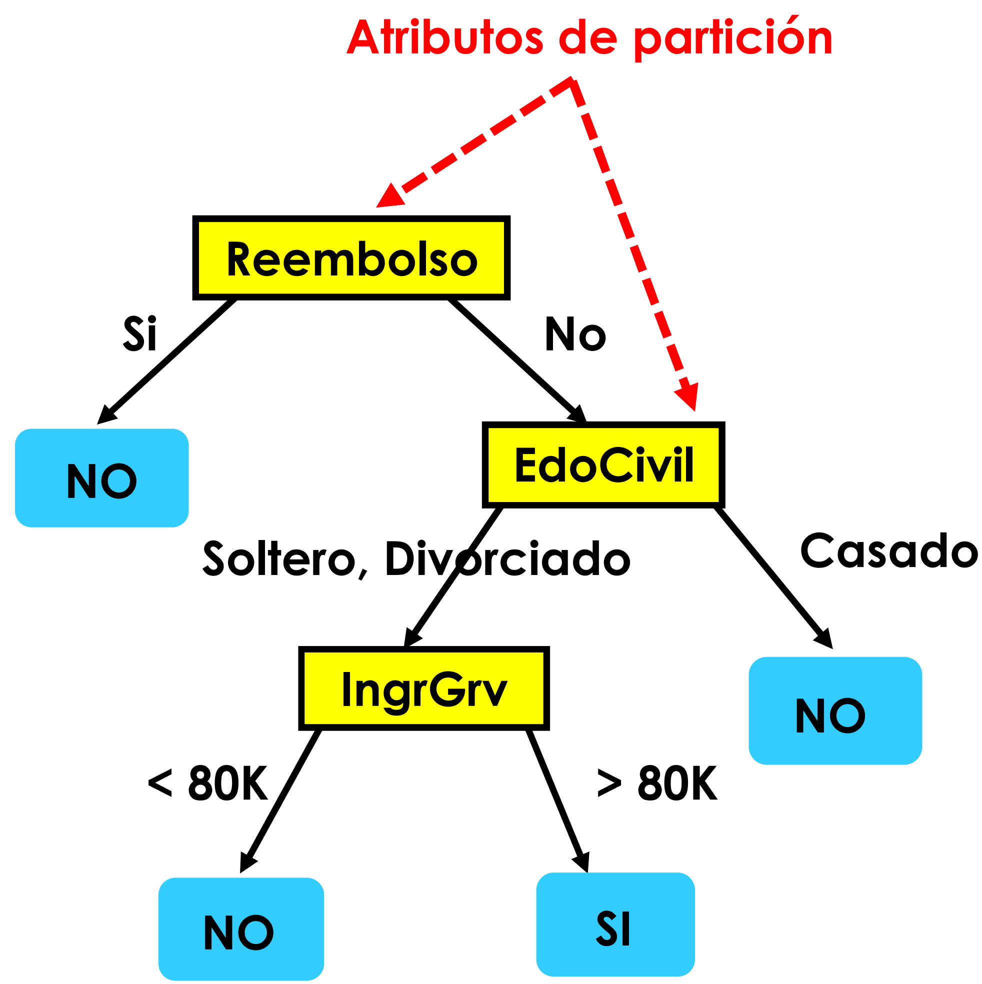
…Inducción de Árboles de Decisión

categórica categórica continua clase

id	Reembolso	Estado civil	Ingreso gravable	¿Engaña?
1	Si	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Si	Casado	120K	No
5	No	Divorciado	95K	Si
6	No	Casado	60K	No
7	Si	Divorciado	220K	No
8	No	Soltero	85K	Si
9	No	Casado	75K	No
10	No	Soltero	90K	Si

Tuplas de entrenamiento

Buscar el "mejor árbol"



Modelo: Árbol de decisión

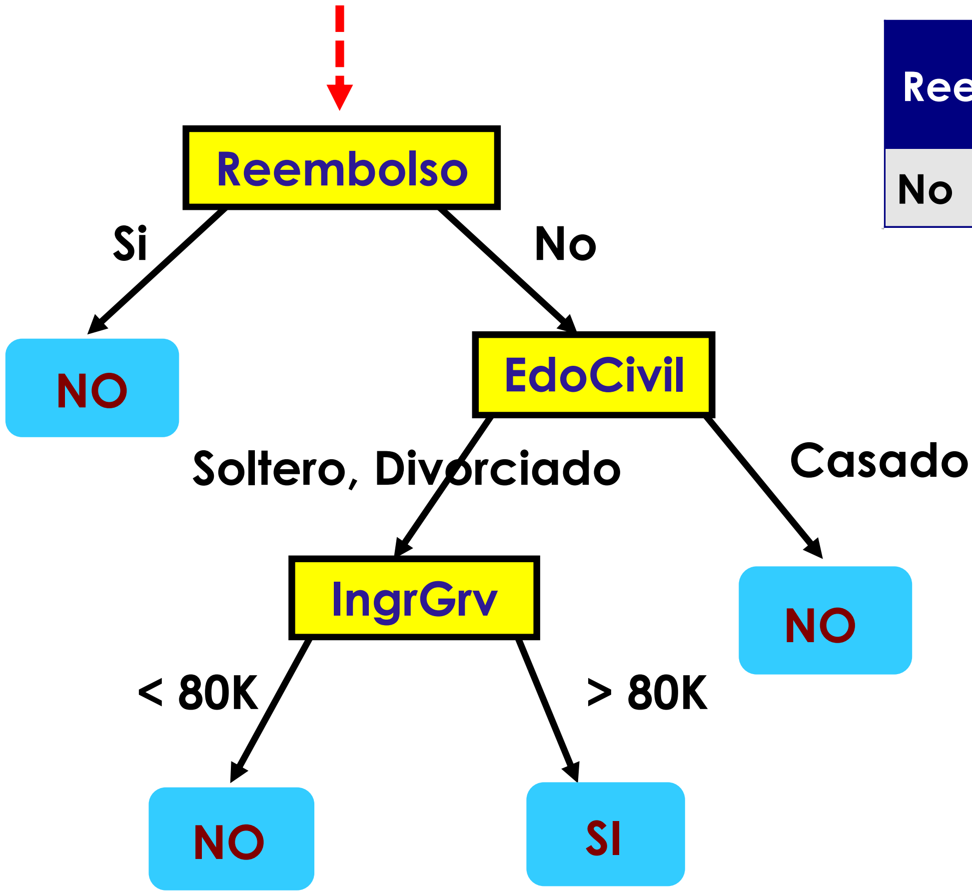
…Inducción de Árboles de Decisión

La forma en que se hace la clasificación es la siguiente:

- *Dada una* ***tupla X*** *(para la cual no se conoce la etiqueta de clase asociada), los valores de cada atributo de la tupla se prueban contra el* ***árbol de decisión****.*
- *Un camino se traza desde la* ***raíz*** *al* ***nodo hoja*** *(el cual conoce la etiqueta de clase predicha).*
- *Los árboles de decisión pueden ser fácilmente convertidos a* ***reglas de clasificación****.*

…Inducción de Árboles de Decisión

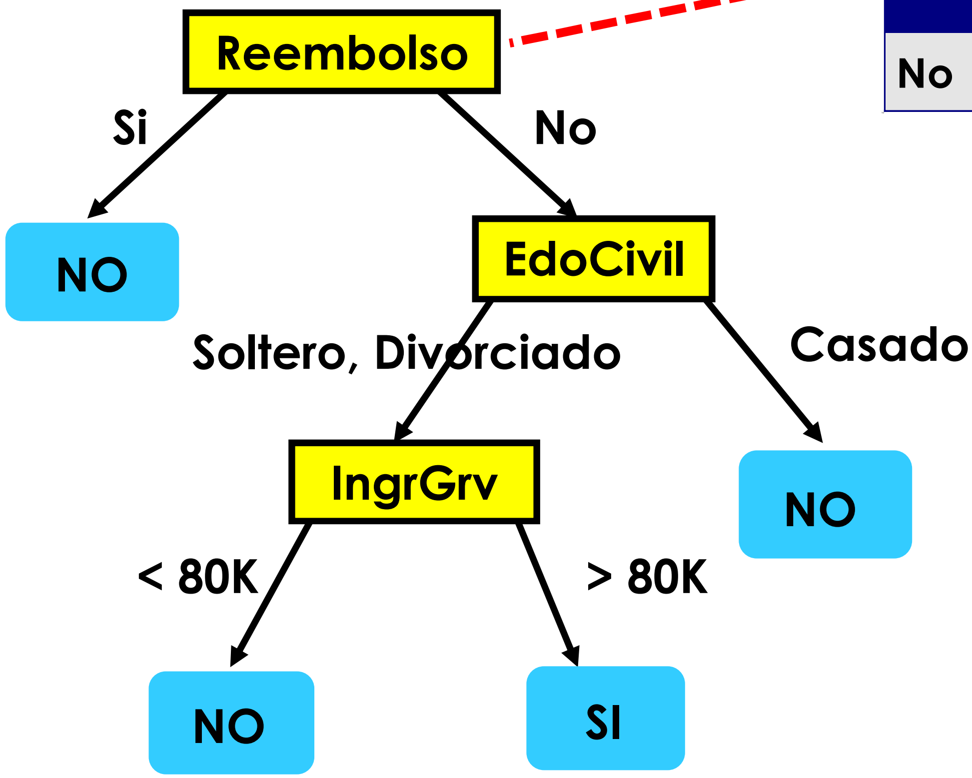
Comienza con la raíz del árbol



Datos de prueba

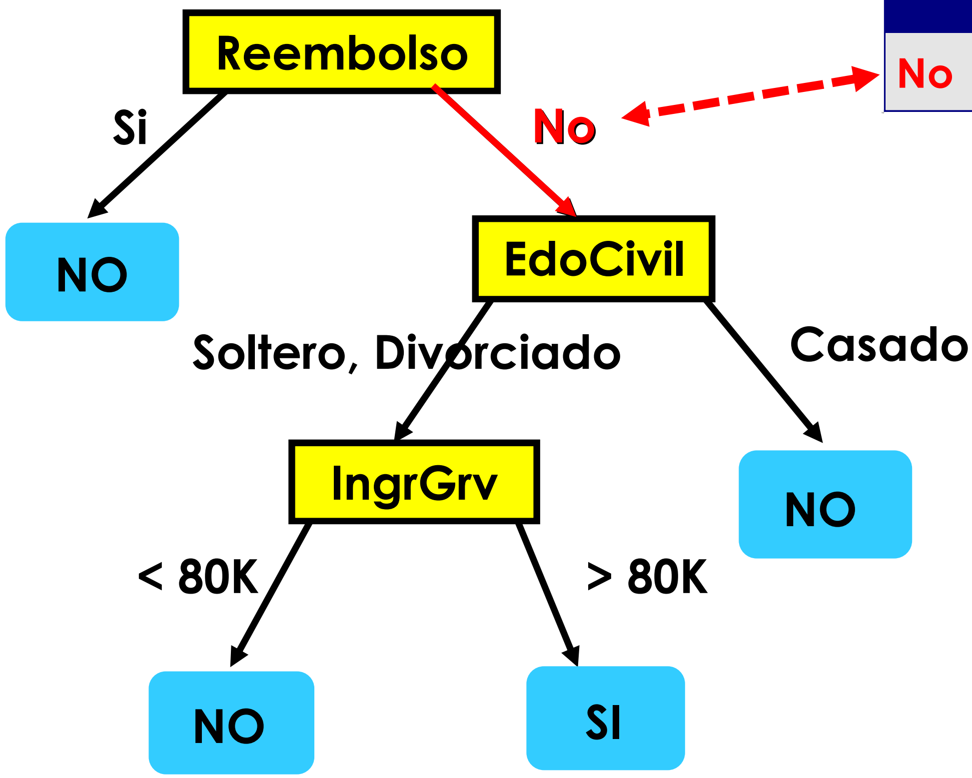
Reembolso	Estado Civil	Ingreso gravable	¿Engaña?
No	Casado	80K	?

...Inducción de Árboles de Decisión



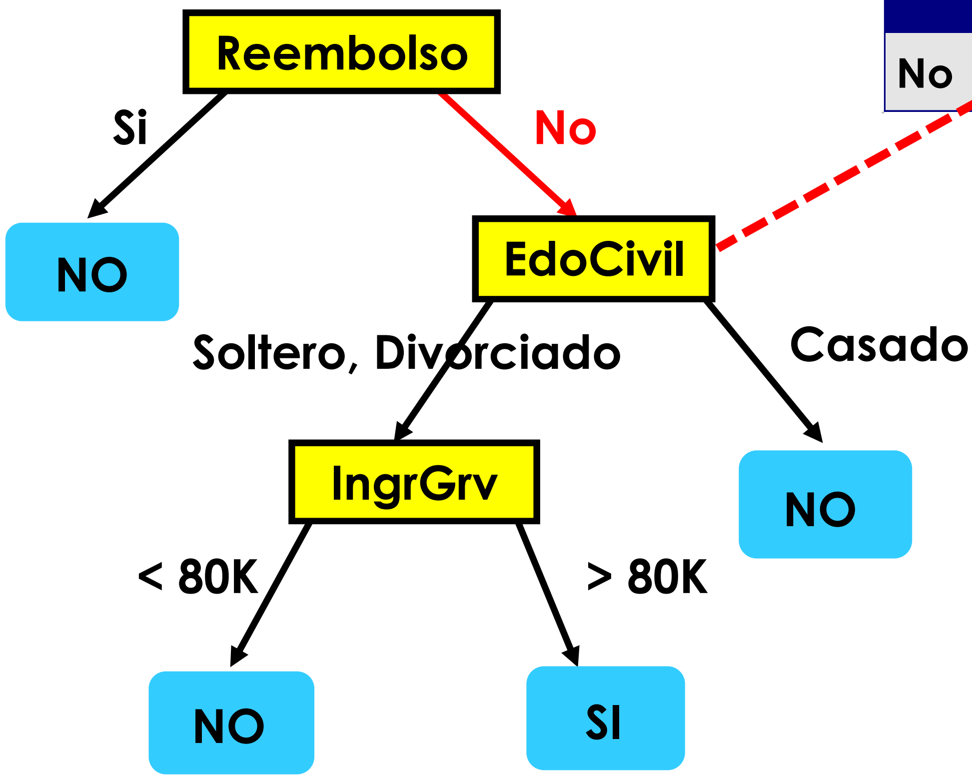
Datos de prueba

Reembolso	Estado Civil	Ingreso gravable	¿Engaña?
No	Casado	80K	?



Datos de prueba

Reembolso	Estado Civil	Ingreso gravable	¿Engaña?
No	Casado	80K	?

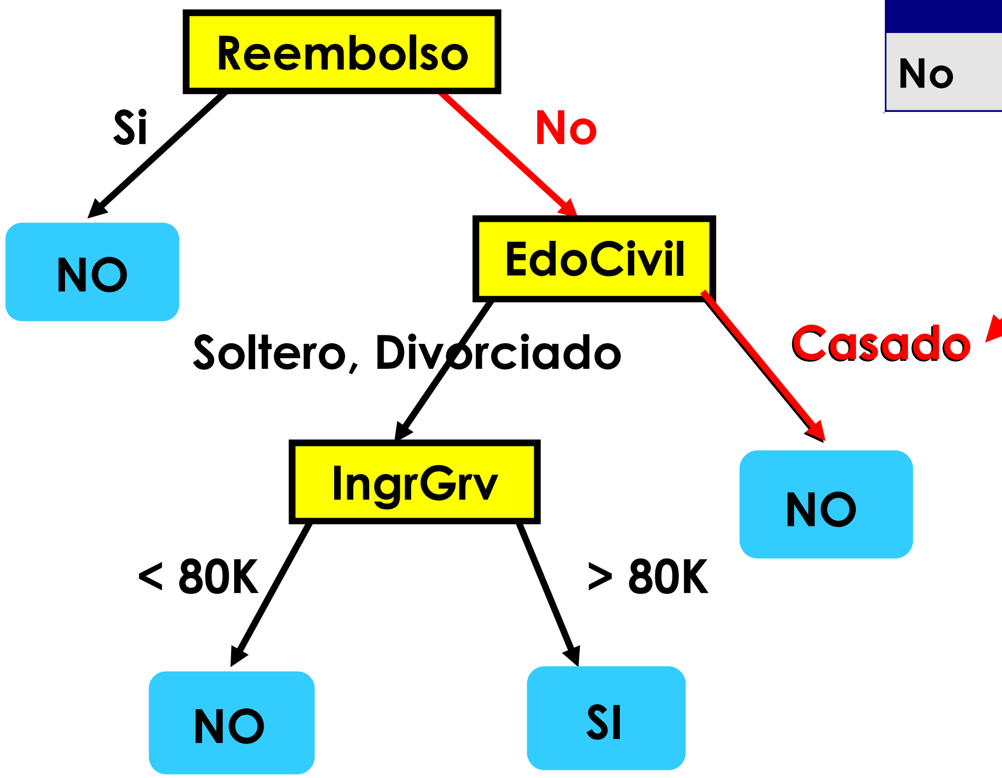


Datos de prueba

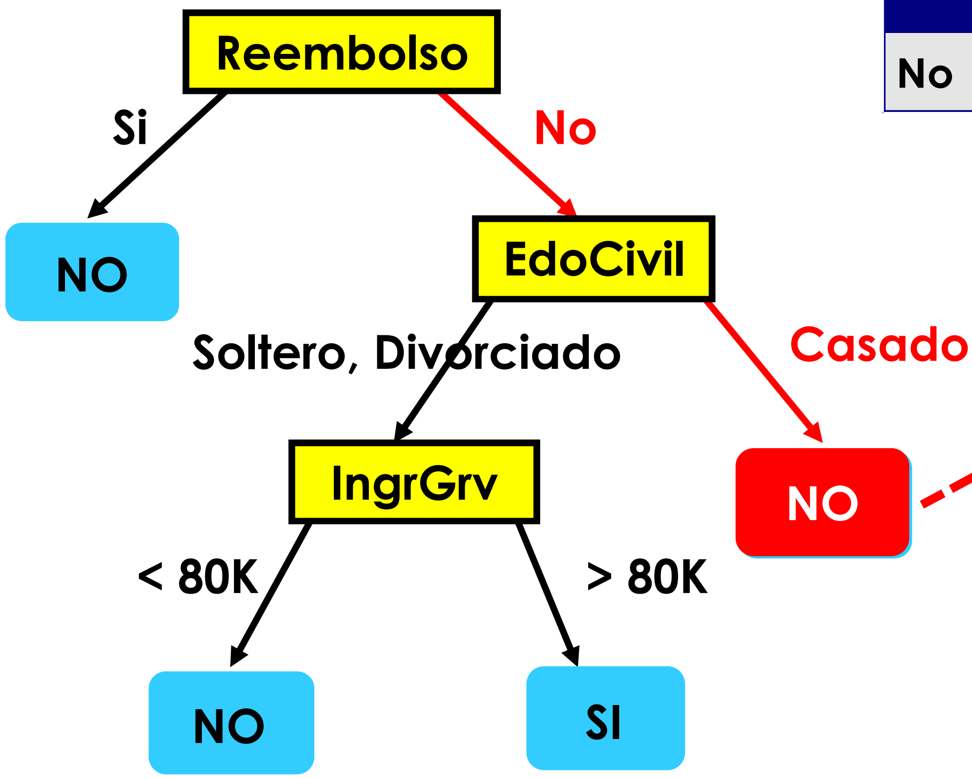
Reembolso	Estado Civil	Ingreso gravable	¿Engaña?
No	Casado	80K	?

Datos de prueba

Reembolso	Estado Civil	Ingreso gravable	¿Engaña?
No	Casado	80K	?



...Inducción de Árboles de Decisión



Datos de prueba

Reembolso	Estado Civil	Ingreso gravable	¿Engaña?
No	Casado	80K	NO

Asigna **¿Engaña?** como **“No”**

- Los **árboles de decisión** son muy populares ya que para su construcción no se requiere ningún conocimiento de dominio o establecimiento de parámetros, por lo que son recomendados para hacer un **descubrimiento de conocimiento**.
- Debido a su forma de construcción son **fáciles de asimilar**.
- Los pasos de aprendizaje y clasificación son simples y rápidos, además de que en general tienen buena **exactitud**.
- Su éxito depende de los datos sobre los que se aplique.
- Son utilizados en varias áreas de aplicación: **medicina, manufactura y producción, análisis financiero, astronomía, biología molecular, etc**.

¿Cómo construir un árbol de decisión a partir de un conjunto de entrenamiento?

Algoritmo de Hunt

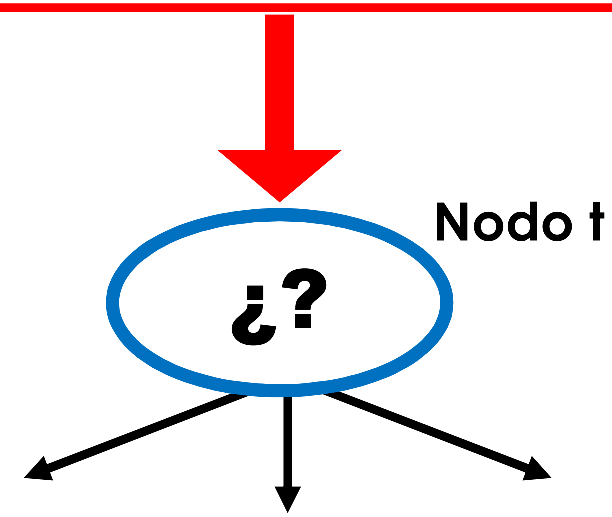
Dado un conjunto D_t (*tuplas de entrenamiento*) que llega a un nodo t, el procedimiento es el siguiente:

- Si D_t contiene registros que pertenecen a una misma clase y_t, entonces t es un nodo hoja etiquetado con y_t .
- Si D_t es un conjunto vacío, entonces t es una nodo hoja etiquetado con la clase default y_d.
- Si D_t contiene registros que pertenecen a más de una clase, utilizar un atributo de prueba para **dividir** los datos en subconjuntos más pequeños.
- Aplicar el procedimiento de forma recursiva a cada subconjunto.

¿Cuál atributo debiera probarse en cada división?

D_t

id	Reemb.	Estado civil	Ingreso gravable	Engaña?
1	Si	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Si	Casado	120K	No
5	No	Divorciado	95K	Si
6	No	Casado	60K	No
7	Si	Divorciado	220K	No
8	No	Soltero	85K	Si
9	No	Casado	75K	No
10	No	Soltero	90K	Si



Generación de un árbol de decisión

Algoritmo:

- **Objetivo:**

  Generar un árbol de decisión a partir de un conjunto de tuplas de entrenamiento de una partición D.

- **Entrada:**
  - ❑ **Una partición de datos D**, la cual es un conjunto de tuplas de entrenamiento y sus etiquetas de clase asociadas;
  - ❑ **Una *lista_de_atributos***, la cual es el conjunto de atributos de partición candidatos;

- **Método_selección_atributos:**

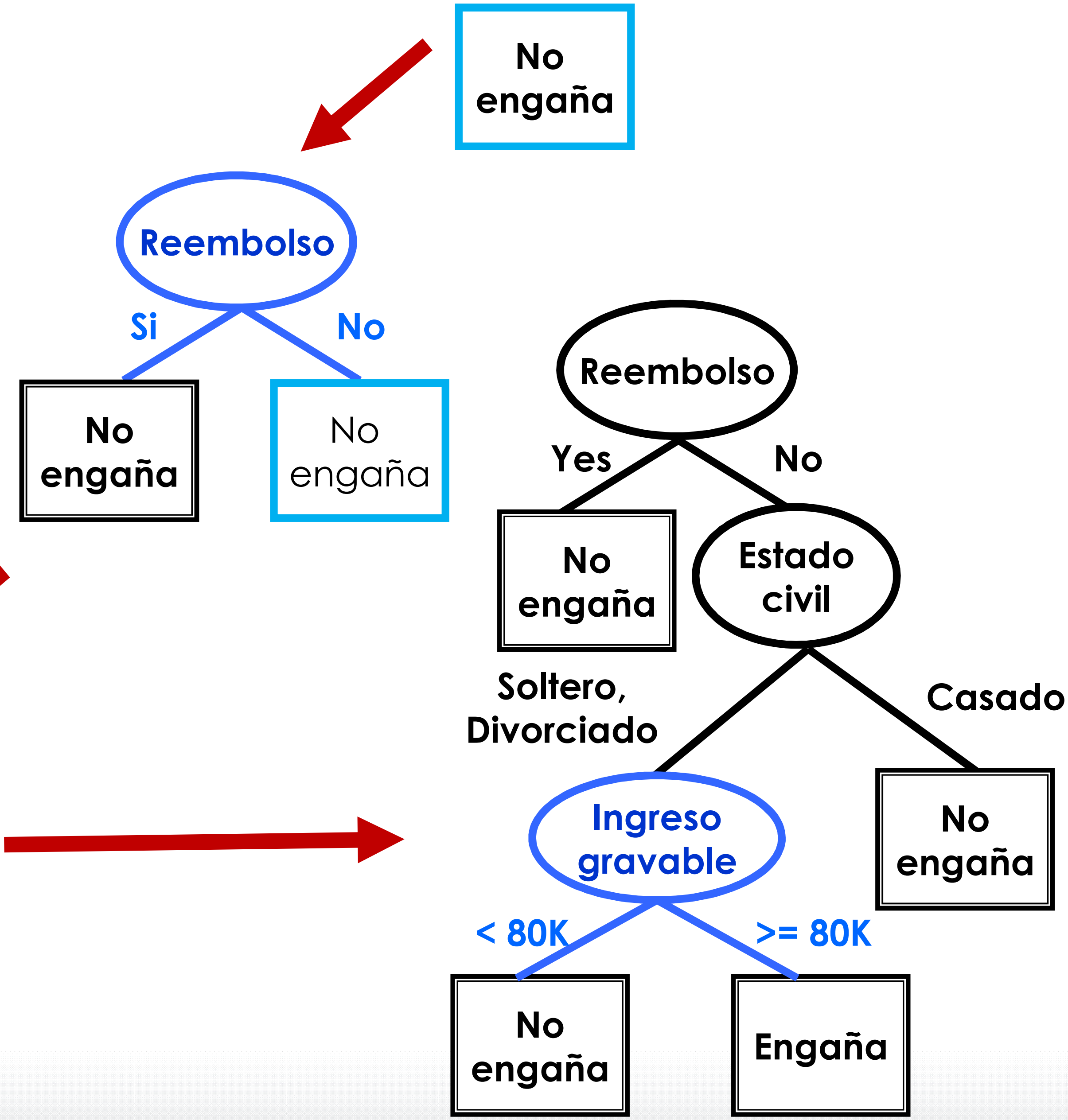
  Un procedimiento para determinar el **criterio de partición** que **mejor divida** las tuplas en **clases individuales**. Este criterio consiste de un **atributo de partición** y posiblemente de un punto de partición o un subconjunto de partición.

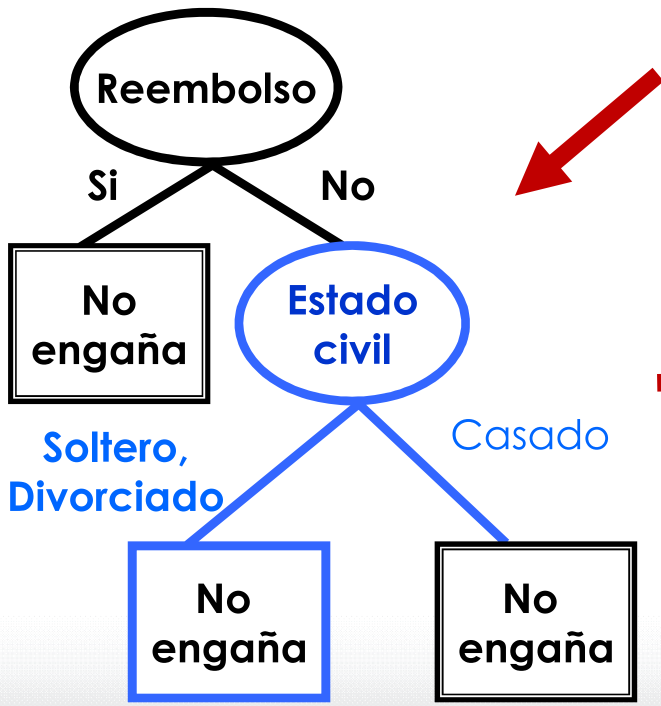
- **Salida:**

  Un árbol de decisión.

…Generación de un árbol de decisión

id	Reemb	Estado civil	Ingreso gravable	Engaña
1	Si	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Si	Casado	120K	No
5	No	Divorciado	95K	Si
6	No	Casado	60K	No
7	Si	Divorciado	220K	No
8	No	Soltero	85K	Si
9	No	Casado	75K	No
10	No	Soltero	90K	Si





Aspectos a considerar

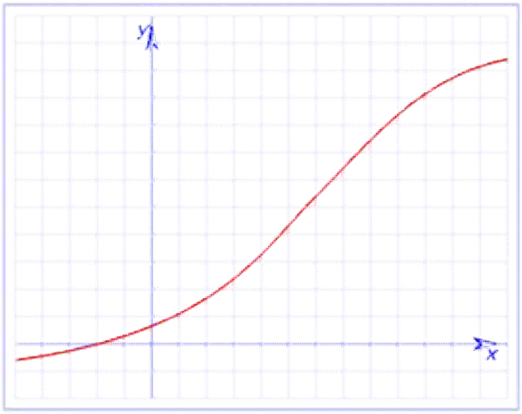
- Utiliza una estrategia **greedy**:

  Dividir los registros en función de una prueba de atributo que optimiza cierto criterio.

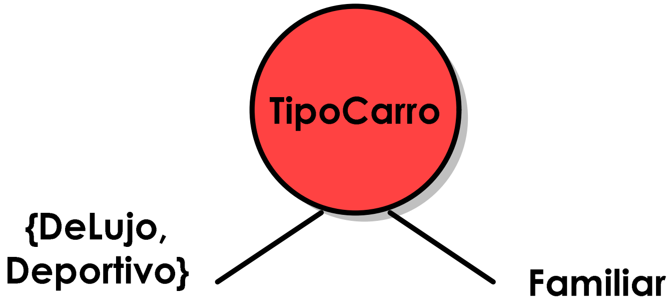
- Cuestiones:
  - ❑ Determinar cómo particionar los registros:
    - → **¿Cómo especificar la condición de prueba?**
    - → **¿Cómo determinar la mejor partición?**
  - ❑ Determinar cuándo detener el particionado
  - ❑ ¿Se debe utilizar una partición binaria o múltiple?

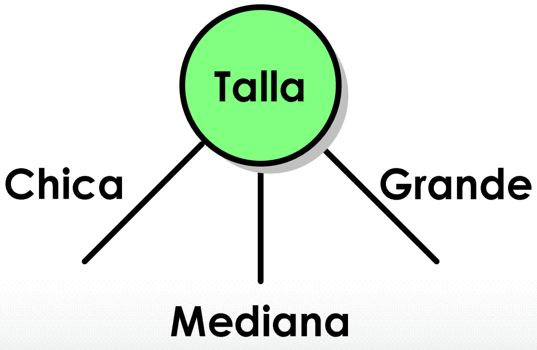
Condición de prueba

- Depende del tipo de atributo
  - ***Nominal***
  - ***Ordinal***
  - ***Continuo***

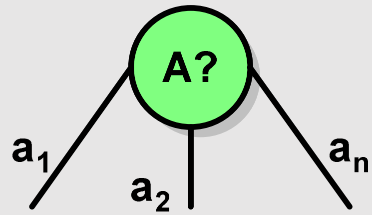
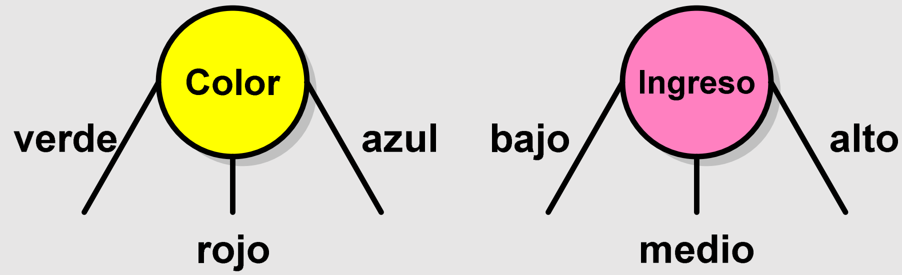
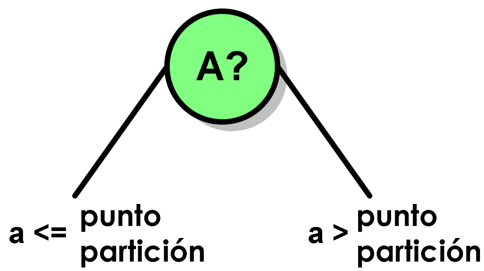
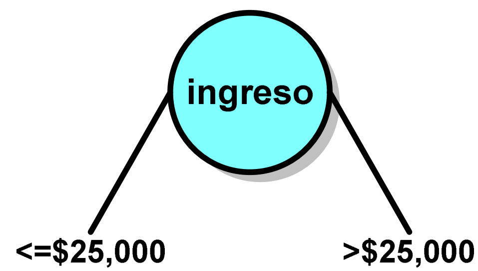
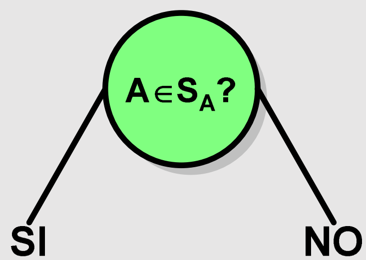
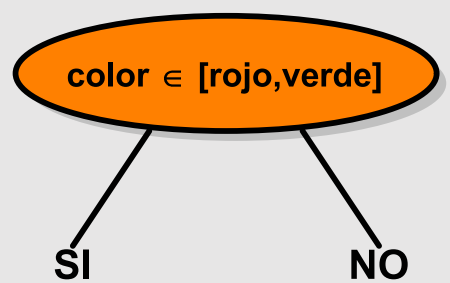


- Depende del número de formas de dividir:



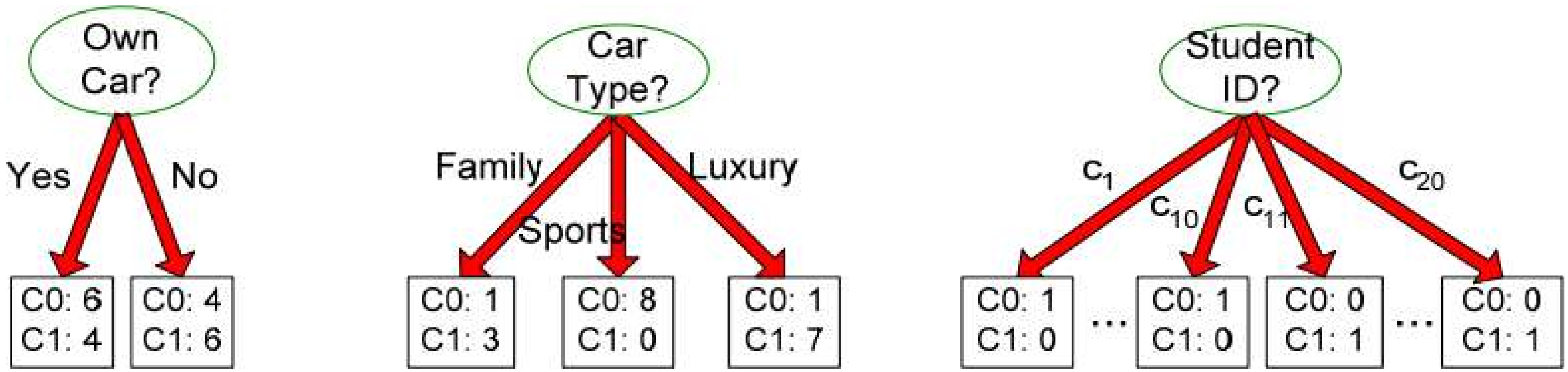


Particiones

Tipo de partición		Ejemplo
Valor discreto		
Valor continuo		
Valor discreto para árbol binario		

¿Cuál es la mejor partición?

- Se tiene un conjunto de **20 tuplas**, 10 de ellas etiquetadas con la **clase 0** y 10 etiquetados con la **clase 1**.



- **¿Cuál condición de prueba es la mejor?**
  - ❑ Se van a preferir nodos con distribución de clases homogéneos.
  - ❑ Necesitamos por ende, una medida de la impureza del nodo.

Non-homogeneous,

High degree of impurity

Homogeneous,

Low degree of impurity

Medidas de selección de atributos

- Se trata de un conjunto de **heurísticas** para determinar el **criterio de partición** que **mejor** divida un conjunto de datos **D** (*contiene etiquetas de clase y tuplas de entrenamiento*) en **clases individuales**.
- Si se desea dividir **D** en particiones **más pequeñas** de acuerdo a los resultados del **criterio de partición**, idealmente cada partición debería ser **pura** (*las tuplas que pertenecen a una partición determinada son de la misma clase*).
- Estas medidas también son conocidas como **reglas de partición** (*determinan cómo las tuplas en un nodo dado se deben dividir*)
- Estas medidas proporcionan un **ranking** por cada atributo descrito en las tuplas de entrenamiento que se proporcionan.
- El atributo que tiene la **mejor puntuación** para la medida es el que se elige como atributo de partición para las tuplas dadas.

ID3: Ganancia de información

- A finales de los **70s** y principios de los **80s J. Ross Quinlan** (*investigador en máquinas de aprendizaje*) desarrolló el algoritmo ID3 (***Iterative Dicotomiser***).
- Utiliza un enfoque **greedy** apoyándose en el enfoque ***top-down***.

Machine Learning 1: 81–106, 1986

Induction of Decision Trees

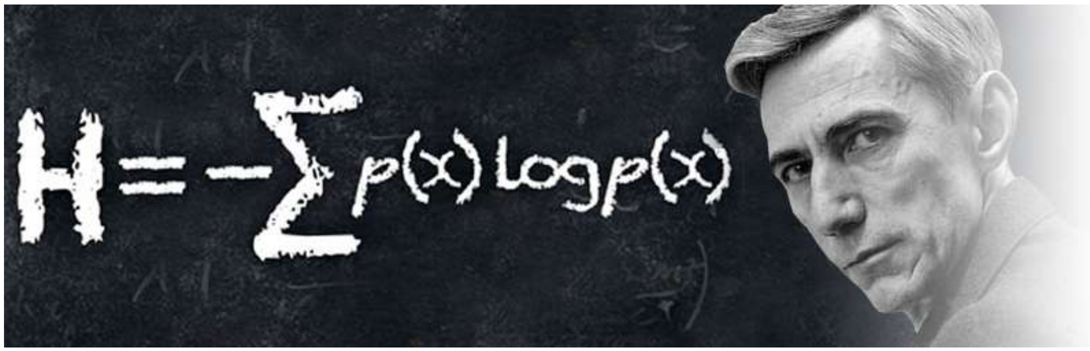
J.R. QUINLAN (munnari!nswitgould.oz!quinlan@scismo.css.gov)
Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney 2007, Australia

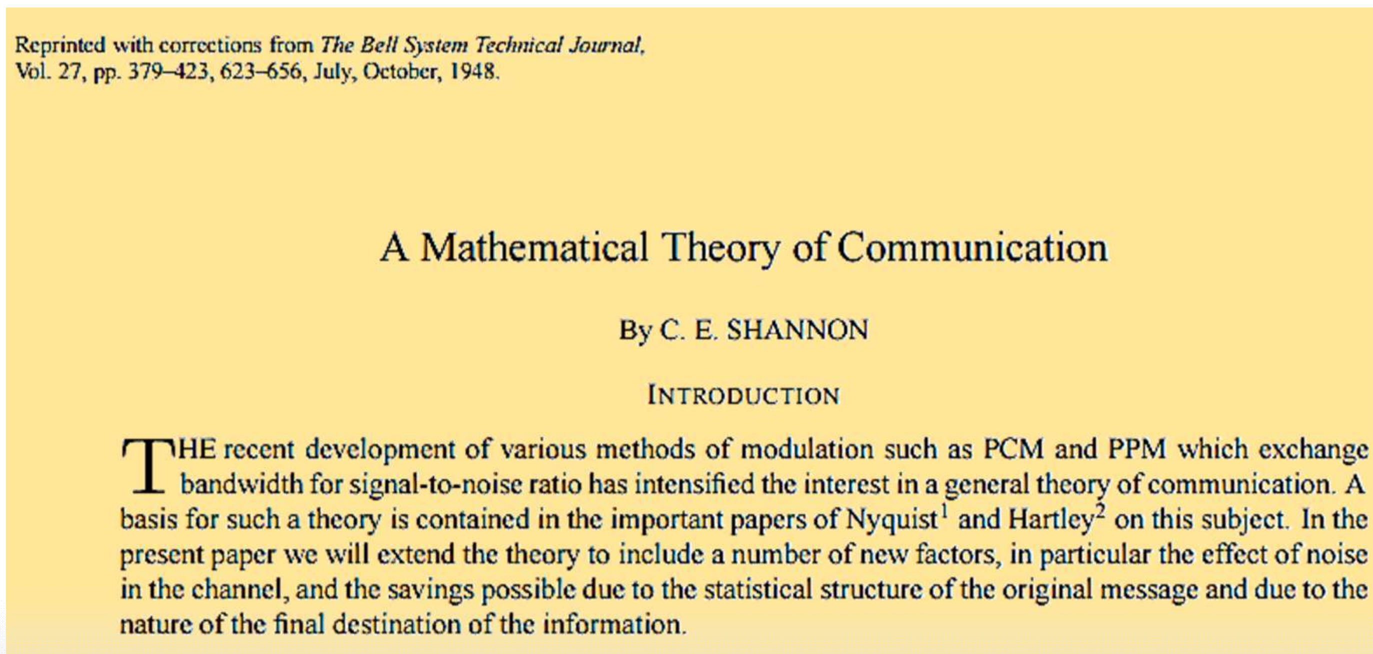
(Received August 1, 1985)

Key words: classification, induction, decision trees, information theory, knowledge acquisition, expert systems

…ID3: Ganancia de información

- Este algoritmo se basa en los estudios de **Claude Shannon** (*pionero de la Teoría de la Información*), que estudiaba el **valor o contenido** de la información de los mensajes:





Reprinted with corrections from *The Bell System Technical Journal*,
Vol. 27, pp. 379–423, 623–656, July, October, 1948.

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

…ID3: Ganancia de información

- Estableció el concepto de **Entropía de la información,** la cual *mide* la incertidumbre de una fuente de información.
- La entropía se puede considerar como la ***cantidad de información promedio*** que contienen los símbolos usados:
  - Los símbolos con ***menor probabilidad*** son los que aportan mayor información; por ejemplo, si se considera como sistema de símbolos a las palabras en un texto, palabras frecuentes como **"*que*", "*el*", "*a*"** aportan poca información, mientras que palabras menos frecuentes como **"*corren*", "*niño*", "*perro*"** aportan más información.
  - Si de un texto dado borramos un **"*que*"**, seguramente no afectará a la comprensión y se sobreentenderá, lo cual no ocurriría si borramos la palabra **"*niño*"** del mismo texto original.

…ID3: Ganancia de información

Entropía (informática)

cantidad de información (promedio)

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

distribución de los símbolos

probabilidad de observar un símbolo particular

1948

…ID3: Ganancia de información

- Dado un nodo **N** que representa a las tuplas de la partición **D**, el atributo que tenga la **mayor ganancia de información** se elige como el atributo de partición para el nodo **N**.
- Este atributo **reduce al mínimo** la **información necesaria** para clasificar las tuplas en las particiones resultantes y **refleja menos aleatoriedad** o "impureza" en estas particiones.
- Este enfoque **minimiza** el número esperado de ensayos necesarios para **clasificar una tupla dada** y **garantiza encontrar un árbol de forma simple** (*pero no necesariamente el más simple*).

…ID3: Ganancia de información

La **información esperada**, necesaria para clasificar una tupla en D está dada por:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Donde:

- **p_i** es la probabilidad de que una tupla arbitraria en **D** pertenezca a una clase **C_i**, se estima a partir de **$|C_{i,D}|/|D|$**
- **$|C_{i,D}|$** número de tuplas de la clase **C_i** en la partición **D**
- **$|D|$** es el número de tuplas en la partición **D**.
- Se utiliza **$\log_2$** debido a que la información se codifica en **bits**.
- **Info(D)** es la cantidad promedio de información necesaria para identificar la etiqueta de clase de una tupla en **D.**

A **Info(D)** también se le conoce como **Entropía**.

…ID3: Ganancia de información

- Ahora supongamos que queremos dividir la tuplas en **D** en algunos atributos de **A**, que tiene **n** valores distintos, $\{a_1, a_2, \ldots, a_n\}$.
- Si **A** tiene valores discretos, estos valores corresponden directamente a **n resultados** de una prueba en **A**, entonces, el atributo **A** puede utilizarse para dividir **D** en **n particiones** o subconjuntos, $\{D_1, D_2, \ldots, D_n\}$, donde D_j contiene aquellas tuplas en **D** que tiene el resultado a_j de **A**:

*Idealmente, deseamos que estas particiones produzcan clasificaciones exactas de tuplas (**particiones puras**).*

- Sin embargo, es mucho más probable que se obtengan particiones impuras.

¿***Cuánta información adicional*** *necesitamos (después de la partición)*

*para obtener una **partición exacta**?*

…ID3: Ganancia de información

$$Info_A(D) = \sum_{j=1}^{n} \frac{|D_j|}{|D|} \times Info(D_j)$$

Donde:

- El término **$|D_j|/|D|$** actúa como el peso de la partición de **orden j**.
- **$Info_A(D)$** es la información esperada necesaria para clasificar una tupla de D **basada en la partición hecha por A**.
- **Cuanto menor** sea la información esperada requerida, **mayor es la pureza** de las particiones.

…ID3: Ganancia de información

- Finalmente, la **ganancia de información** se define como la **diferencia** entre el **requerimiento de información original** (es decir, sobre la base de sólo la proporción de clases) y el **nuevo requerimiento** (es decir, obtenida después de la partición en A):

$$Gain(A) = Info(D) - Info_A(D)$$

- La ganancia nos dice qué tanto ganaríamos si partimos un nodo **N** en el atributo **A.**
- El atributo **A** con la **mayor ganancia de información** se elige como el atributo de partición en el nodo **N**.

ID3: Ejemplo

Supongamos que se tiene el siguiente conjunto de entrenamiento D, con tuplas que tienen etiquetas de clase:

ID	edad	ingreso	estudiante	calificacion_credito	comprar_computadora
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

ID3: Ejemplo

Realizando los conteos correspondientes:

comprar_computadora		
SI	9	14
NO	5	

Edad			
youth	SI	2	5
	NO	3	
middle_age	SI	4	4
	NO	0	
senior	SI	3	5
	NO	2	

Ingreso			
low	SI	3	4
	NO	1	
medium	SI	4	6
	NO	2	
high	SI	2	4
	NO	2	

Estudiante			
si	SI	6	7
	NO	1	
no	SI	3	7
	NO	4	

calificación_credito			
fair	SI	6	8
	NO	2	
excellent	SI	3	6
	NO	3	

ID3: Ejemplo

Para determinar el criterio de partición, necesitamos calcular la **ganancia de información** de cada atributo:

- Lo primero que debemos hacer es calcular la información necesaria esperada para clasificar una tupla en la partición D:

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$$

- Ahora, vamos a calcular los requerimientos de información esperados para cada atributo. Comencemos con edad:

$$Info_A(D) = \sum_{j=1}^{n} \frac{|D_j|}{|D|} \times Info(D_j)$$

Total de jóvenes que SI comprarían

Total de jóvenes que NO comprarían

Total de jóvenes

$$Info_{edad}(D) = \frac{5}{14} \times \left(\underbrace{-\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}}_{\textbf{Total de jóvenes}}\log_2\left(\frac{3}{5}\right) \right) + \cdots$$

Total de tuplas

Entonces, ***Info**$_{edad}$**(D)*** quedaría de la siguiente forma:

$$Info_{edad}(D) = \frac{5}{14}\times\left(-\frac{2}{5}\log_2\left(\frac{2}{5}\right)-\frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)+$$
$$\frac{4}{14}\times\left(-\frac{4}{4}\log_2\left(\frac{4}{4}\right)-\frac{0}{4}\log_2\left(\frac{0}{4}\right)\right)+$$
$$\frac{5}{14}\times\left(-\frac{3}{5}\log_2\left(\frac{3}{5}\right)-\frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)$$

$$Info_{edad}\left(D\right) = 0.3468 + 0 + 0.3468 = 0.6936 \text{ bits}$$

$$Info_{estudiante}(D) = \frac{7}{14}\times\left(-\frac{3}{7}\log_2\left(\frac{3}{7}\right)-\frac{4}{7}\log_2\left(\frac{4}{7}\right)\right)+$$
$$\frac{7}{14}\times\left(-\frac{6}{7}\log_2\left(\frac{6}{7}\right)-\frac{1}{7}\log_2\left(\frac{1}{7}\right)\right) = 0.789 \text{ bits}$$

$$Info_{ingreso}(D) = \frac{4}{14}\times\left(-\frac{2}{4}\log_2\left(\frac{2}{4}\right)-\frac{2}{4}\log_2\left(\frac{2}{4}\right)\right)+ \frac{6}{14}\times\left(-\frac{4}{6}\log_2\left(\frac{4}{6}\right)-\frac{2}{6}\log_2\left(\frac{2}{6}\right)\right)+ \frac{4}{14}\times\left(-\frac{3}{4}\log_2\left(\frac{3}{4}\right)-\frac{1}{4}\log_2\left(\frac{1}{4}\right)\right)=0.911\text{ bits}$$

$$Info_{calif\_crédito}(D) = \frac{8}{14}\times\left(-\frac{6}{8}\log_2\left(\frac{6}{8}\right)-\frac{2}{8}\log_2\left(\frac{2}{8}\right)\right)+ \frac{6}{14}\times\left(-\frac{3}{6}\log_2\left(\frac{3}{6}\right)-\frac{3}{6}\log_2\left(\frac{3}{6}\right)\right)=0.892\text{ bits}$$

$$\underline{Ganancia(edad) = 0.940 - 0.694 = 0.246}$$

$$Ganancia(\text{ingreso}) = 0.940 - 0.911 = 0.029$$

$$Ganancia(\text{estudiante}) = 0.940 - 0.789 = 0.151$$

$$Ganancia(\text{calif\_credito}) = 0.940 - 0.892 = 0.048$$

Edad

youth

ingreso	estudiante	Calificación crédito	Comprar computadora
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

Midle_aged

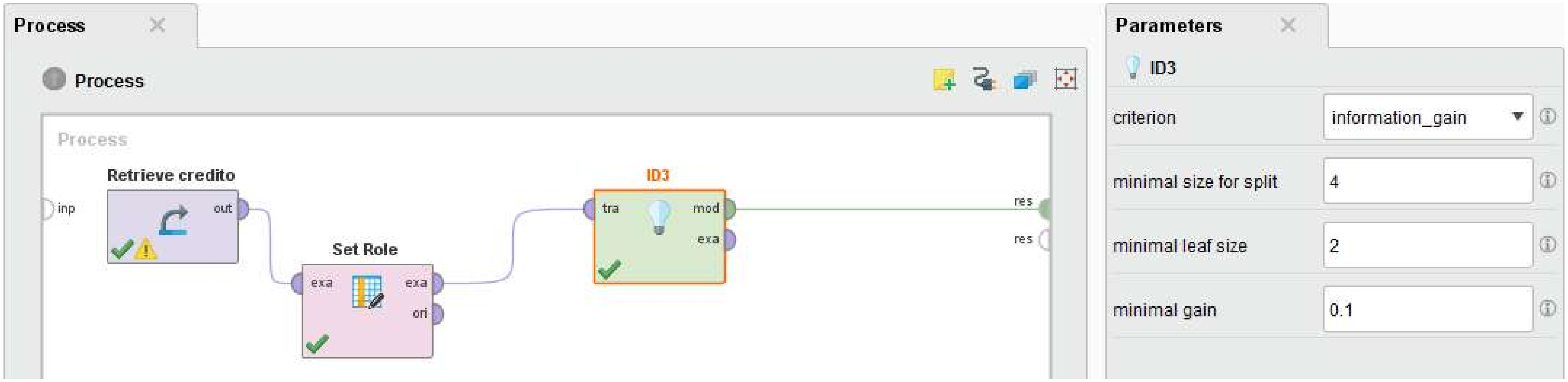
ingreso	estudiante	Calificación crédito	Comprar computadora
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Partición pura

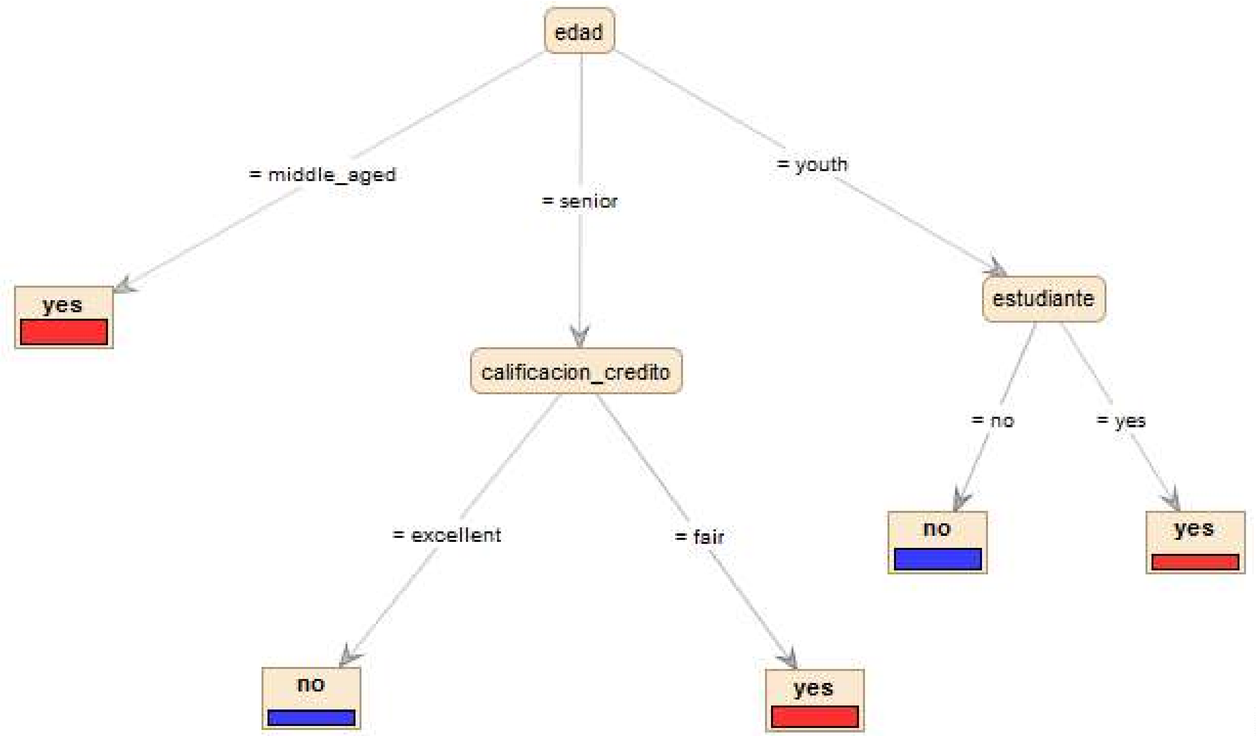
senior

ingreso	estudiante	Calificación crédito	Comprar computadora
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

…ID3: Ejemplo



Process
Process
Process
Retrieve credito
inp
out
Set Role
exa
exa
ori
ID3
tra
mod
exa
res
res
Parameters
ID3
criterion
information_gain
minimal size for split
4
minimal leaf size
2
minimal gain
0.1



edad
= middle_aged
= senior
= youth
yes
calificacion_credito
estudiante
= excellent
= fair
= no
= yes
no
yes
no
yes

ID3: Atributos continuos

- Se debe seleccionar el mejor punto de partición para **A**. La partición se hace en un conjunto discreto de intervalos, por ejemplo: **A < c** y **A ≥ c**.
- **¿Cómo seleccionar c?** Nos gustaría el valor que produzca la mayor ganancia de información. Se sigue la siguiente estrategia:
  - Se **ordenan** todos los valores de forma **creciente.**
  - Típicamente, se selecciona el **punto intermedio** que se encuentra entre cada par de valores adyacentes y cada uno se considera como posible punto de división:

$$\frac{a_i + a_{i+1}}{2}$$

  - Para cada posible punto de división se necesita evaluar $\mathbf{Info_A(D)}$, donde el numero de particiones es **2**.
  - El punto con los **menor requerimiento de información** se selecciona para hacer la partición.

- Por ejemplo, pensemos que tenemos los siguientes datos:

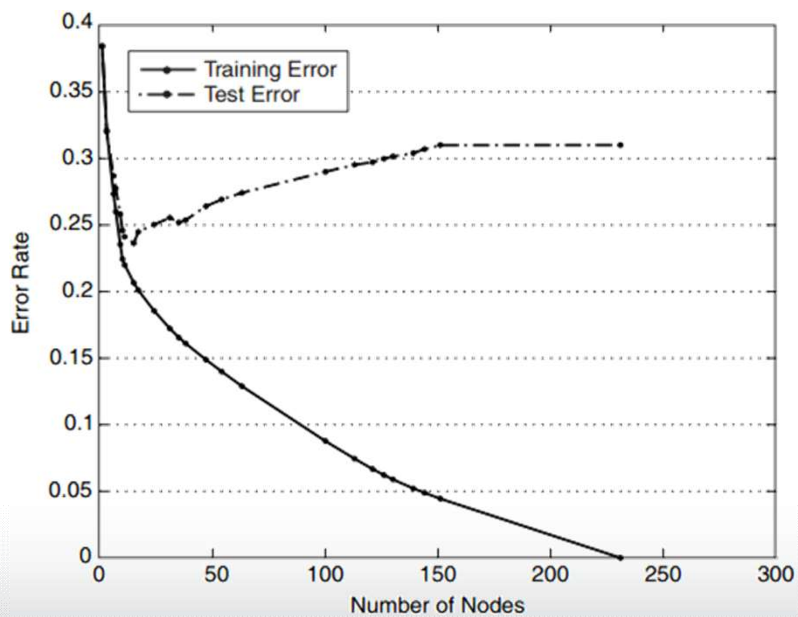
Temperatura	**40**	**48**	**60**	**72**	**80**	**90**
Jugar Tenis	**No**	**No**	**Si**	**Si**	**Si**	**No**

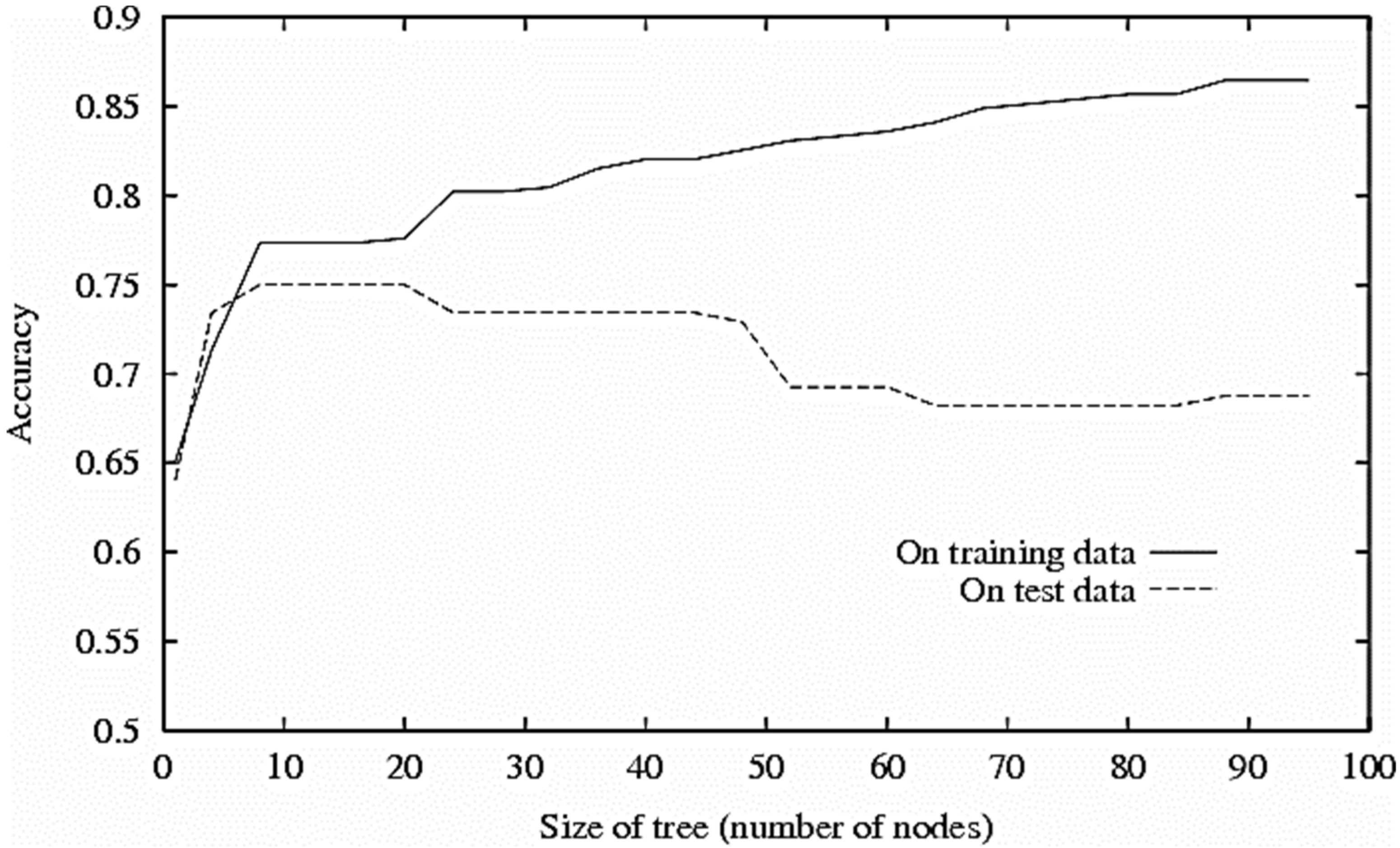
- Candidatos para particionar:

$$C_1 = \frac{40+48}{2} = 44 \qquad C_2 = \frac{48+60}{2} = 54 \qquad C_3 = \frac{60+72}{2} = 66$$

$$C_4 = \frac{72+80}{2} = 76 \qquad C_5 = \frac{80+90}{2} = 85$$

- Su complejidad crece **linealmente** con el número de tuplas de entrenamiento y **exponencialmente** con el número de atributos.
- Favorece la elección de variables con **mayor número** de valores.
- **Problema de sobreajuste:** al hacer crecer el árbol hasta que clasifique correctamente todas la tuplas de entrenamiento:
  - Si hay ruido en las tuplas, el árbol **aprende del ruido.**
  - Si hay pocas tuplas en los nodos hoja, **no son representativos.**
  - **No son capaces de generalizar.**

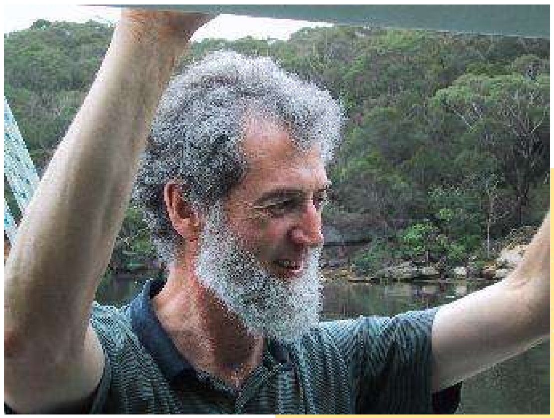




¿Cómo evitarlo?

- Detener el crecimiento cuando la partición no sea estadísticamente significativa.
- Obtener el árbol completo y hacer una **post-poda**.

- **J. Ross Quinlan** propuso en **1993** al sucesor del algoritmo **ID3**, al cual llamó algoritmo **C4.5** y se convirtió en un ***benchmark*** para los nuevos algoritmos de aprendizaje supervisado.



Machine Learning, 16, 235–240 (1994)
© 1994 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Book Review: *C4.5: Programs for Machine Learning* **by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993.**

STEVEN L. SALZBERG salzberg@cs.jhu.edu

Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218

…Árbol C4.5

- Al igual que **ID3** adopta un enfoque ***greedy***.
- Genera un árbol de decisión a partir de los datos de entrenamiento mediante **particiones recursivas**.
- Utiliza la estrategia **profundidad-primero** (*depth-first*), considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona aquella que resulte con **mayor ganancia de información**.
- Trabaja con valores **discretos y continuos**, separando los posibles resultados en **dos ramas**.
- Se trata de árboles **menos frondosos** ya que cada hoja cubre **una distribución de clases**.
- La última versión libre fue el algoritmo **C4.8** antes de que se publicara su versión comercial: **C5**

C4.5: Información de partición

- La **ganancia de información** es una medida que está **sesgada** hacia pruebas que tienen muchos resultados: *prefiere seleccionar atributos que tengan un gran numero de valores*:
  - Por ejemplo, pensemos en un atributo que funciona como **identificador único** (p.e. un ID), si hiciéramos un partición sobre éste, se encontrarían un **gran número de particiones** (tantas como valores se tengan en el atributo), cada una conteniendo **solo una tupla**.
  - En este caso resultan **particiones puras**, donde la información necesaria para clasificar un conjunto de datos **D** basados en esta partición sería **cero**.
  - La información obtenida mediante la división de este atributo es **máxima**.
  - **Es evidente que una partición de este tipo es inútil para la clasificación.**

…C4.5: Información de partición

- El algoritmo **C4.5** utiliza una heurística llamada **tasa de ganancia** (*gain ratio*), la cual intenta superar este sesgo.
- Esta medida aplica una especie de **normalización** a la ganancia de información usando un valor llamado **información de partición**:

$$SplitInfo_A(D) = -\sum_{i=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- Dicha medida representa la **información potencial** que se generaría si se dividiera el conjunto de entrenamiento en ***v particiones*** que corresponden a ***v resultados*** de una prueba sobre un atributo **A**.
- Se diferencia de **ganancia de información**, ya que ésta mide la información con respecto a la clasificación que se adquiere basada en la **misma partición**.

C4.5: Tasa de ganancia

- La tasa de ganancia se define entonces:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

- De esta forma, el atributo con la **máxima tasa de ganancia** es seleccionado como el atributo de partición.
- Es importante hacer notar que si ***SplitInfo*** se **aproxima a cero**, la tasa se vuelve **inestable**, sin embargo el cálculo tiene una restricción ya que la ganancia de información deberá ser muy grande, al menos tan grande como el promedio de ganancia sobre todas las pruebas examinadas.

C4.5: Ejemplo

Regresando al ejemplo que se analizó para el árbol **ID3**:

ID	edad	ingreso	estudiante	calificacion_credito	comprar_computadora
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Vamos por ejemplo a calcular la tasa de ganancia para el atributo **ingreso**:

- Una prueba sobre este atributo dividiría los datos en **tres particiones** (***low, medium y high***), las cuales contienen **4, 6 y 4** tuplas respectivamente, por lo tanto:

$$SplitInfo_{ingreso}(D) = -\frac{4}{14}\times\log_2\left(\frac{4}{14}\right) - \frac{6}{14}\times\log_2\left(\frac{6}{14}\right) - \frac{4}{14}\times\log_2\left(\frac{4}{14}\right)$$

$$= 1.557$$

$$GainRatio(ingreso) = \frac{0.029}{1.557} = 0.0186$$

Para los otros tres atributos quedaría de la siguiente forma:

$$SplitInfo_{edad}(D) = -\frac{5}{14}\times\log_2\left(\frac{5}{14}\right) - \frac{4}{14}\times\log_2\left(\frac{4}{14}\right) - \frac{5}{14}\times\log_2\left(\frac{5}{14}\right)$$

$$= 1.577$$

$$GainRatio(edad) = 0.246/1.577 = 0.156$$

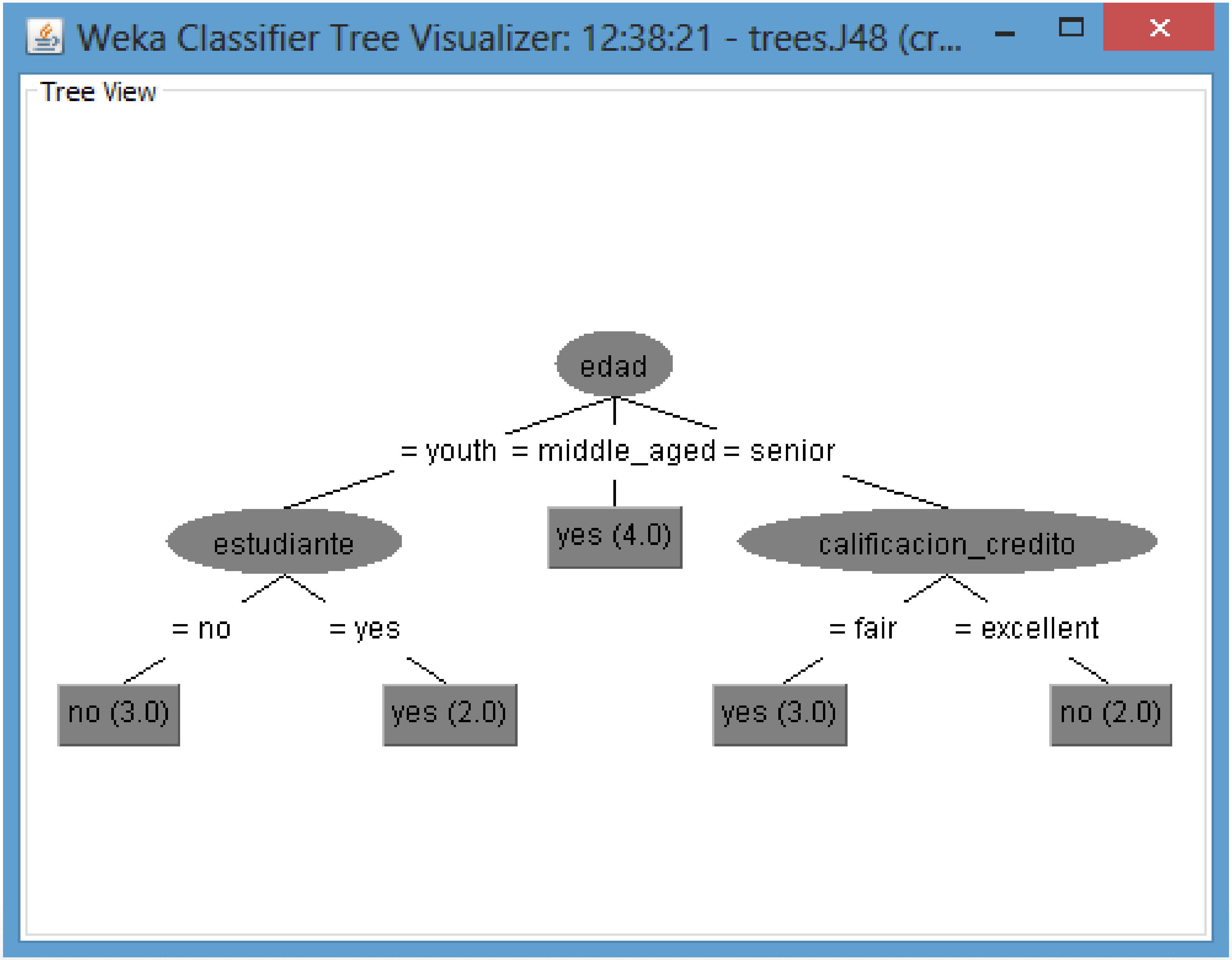
$$SplitInfo_{estudiante}(D) = -\frac{7}{14}\times\log_2\left(\frac{7}{14}\right) - \frac{7}{14}\times\log_2\left(\frac{7}{14}\right) = 1.0$$

$$GainRatio(estudiante) = 0.151/1.0 = 0.151$$

$$SplitInfo_{calif\_cred}(D) = -\frac{8}{14}\times\log_2\left(\frac{8}{14}\right) - \frac{6}{14}\times\log_2\left(\frac{6}{14}\right) = 0.985$$

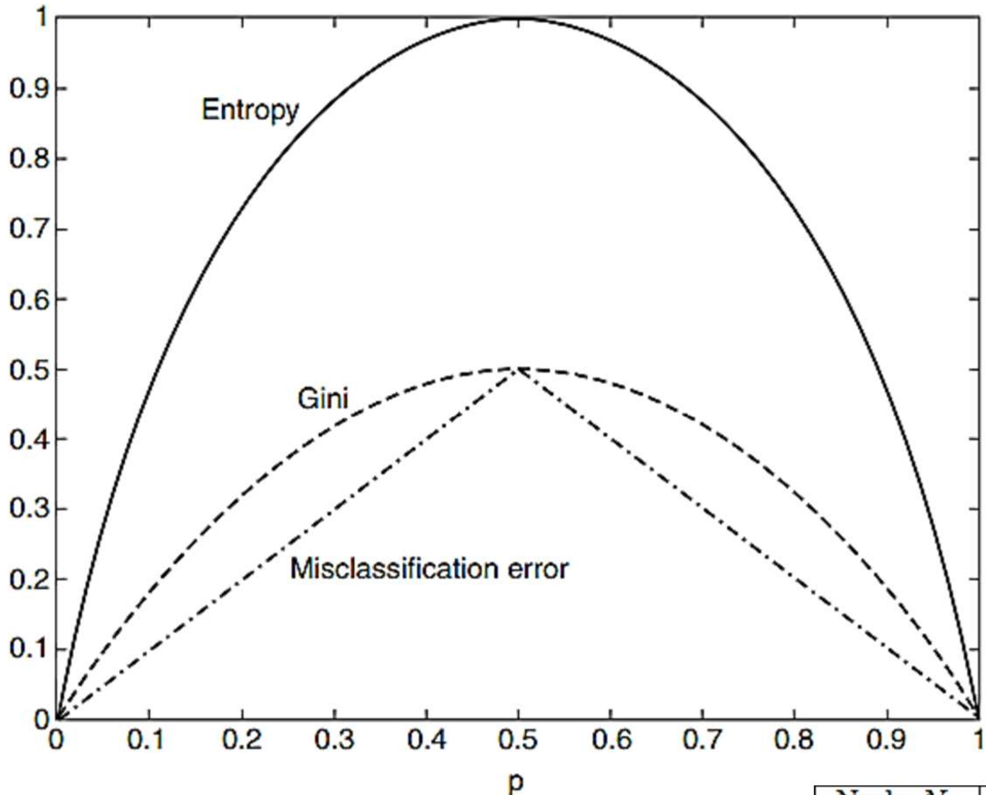
$$GainRatio(calif\_cred) = 0.048/0.985 = 0.048$$

...C4.5: Ejemplo



Weka Classifier Tree Visualizer: 12:38:21 - trees.J48 (cr...
Tree View
edad
= youth
= middle_aged
= senior
estudiante
yes (4.0)
calificacion_credito
= no
= yes
= fair
= excellent
no (3.0)
yes (2.0)
yes (3.0)
no (2.0)

Comparación criterios



Node N_1	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$
$$\text{Entropy} = -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$$
$$\text{Error} = 1 - \max[0/6, 6/6] = 0$$

Node N_2	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$
$$\text{Entropy} = -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$$
$$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$$

Node N_3	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$
$$\text{Entropy} = -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$$
$$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$$