Tema 8: Árboles de decisión

Rosa María Maza Quiroga – rosammq@uma.es

Departamento de Lenguajes y Ciencias de la Computación

Universidad de Málaga

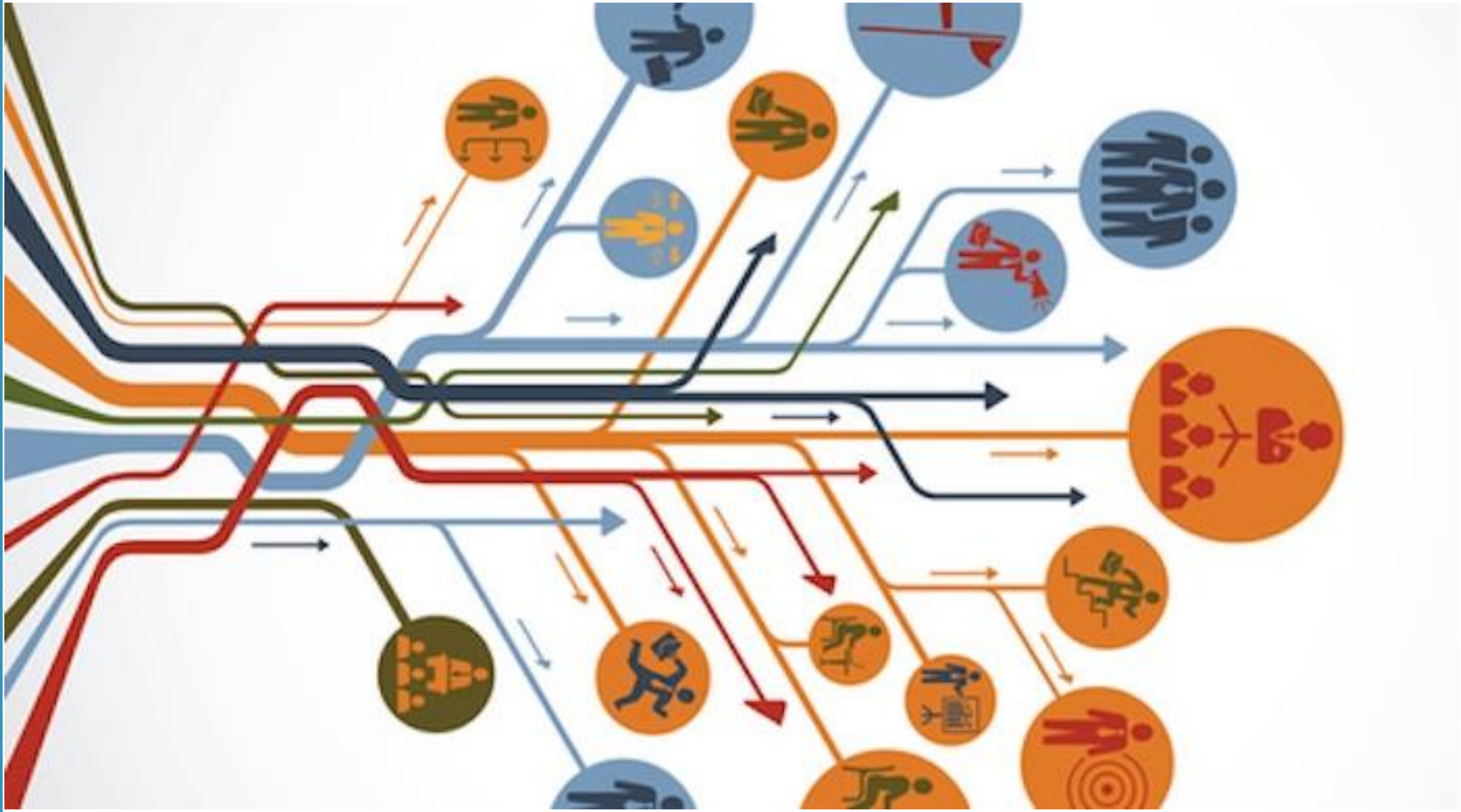
Resumen

1. Fundamentos
2. Algoritmo ID3
3. Medidas para clasificación

Fundamentos

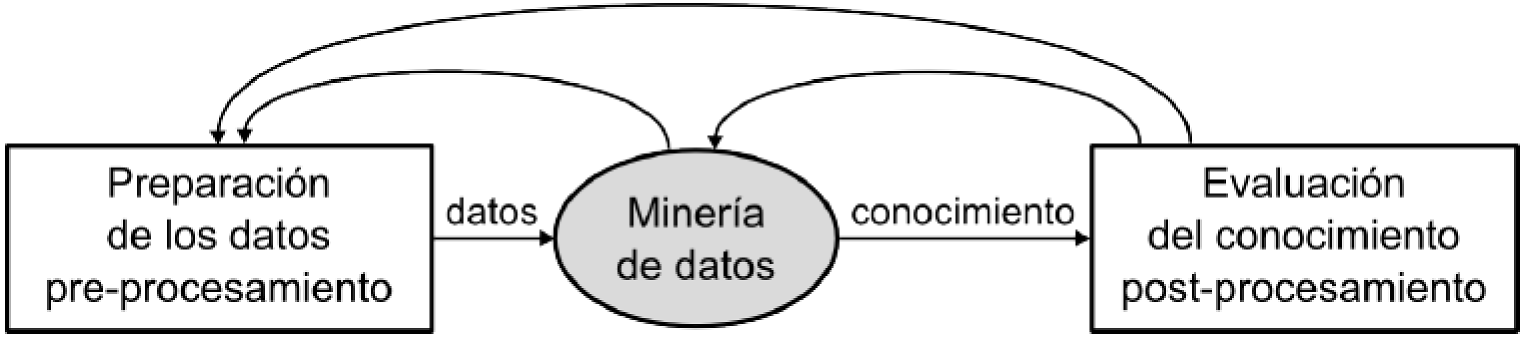
KDD

TDIDT



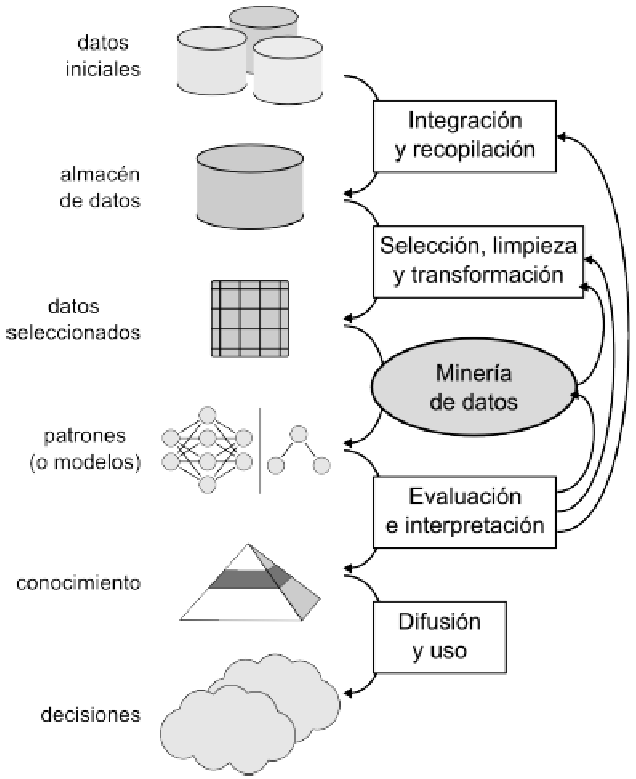
Exrtacción de conocimiento KDD (Knowledge Discovery in Databases)

- Prepara, sondea y explora los datos.
- 5 etapas:
  - Selección: limpieza e integración de diferentes fuentes.
  - Preprocesamiento: estandarización, normalización…
  - Transformación: selección de características, transformación (ej. discretización de variables…)…
  - **Minería de datos**: descubrir información y conocimiento oculto en los datos. Información desconocida y potencialmente útil.
  - Interpretación: resultados.

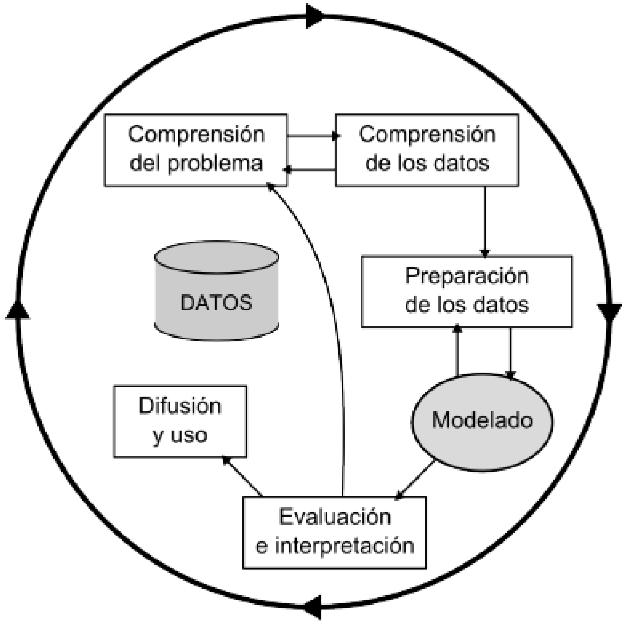


Extracción de conocimiento
KDD (Knowledge Discovery in Databases)

Académico (investigación)



Industrial



Minería de datos

- Métodos, algoritmos y técnicas por las cuales, a partir de un conjunto de experiencias estructuradas de un problema, obtenemos cierto conocimiento sobre dicho problema.

- ÁREAS:
  - **Aprendizaje Automático** (**Machine Learning**):
    - **Aprendizaje inductivo**: no hay símil biológico, conocimiento accesible.
    - Aprendizaje neural: sí hay símil biológico, conocimiento no accesible.
    - Otros:
      - Ej. Similitud, votación.
  - Técnicas estadísticas.

https://topbigdata.es/una-suave-introduccion-a-la-teoria-del-aprendizaje-computacional/

Tipos de Aprendizaje Inductivo

- **No supervisado**: no tenemos las clases de las experiencias.
- **Supervisado**: sí tenemos las clases de las experiencias.
  - **Con conocimiento adicional**: son conocidos aspectos de la estructura del problema.

  Ej. Jerarquía de atributos.
  - **Sin conocimiento adicional**: no se conoce la estructura del problema.
    - **Con modelo**: se dispone de un modelo teórico sobre el concepto a aprender.
      - **Exacto**: exigimos que el concepto aprendido sea idéntico al concepto objetivo
      - **Aproximado**: no exigimos que el concepto aprendido sea idéntico al concepto objetivo, sólo 'parecido'.
    - **Sin modelo**: no se dispone de un modelo teórico sobre el problema a tratar.
      - **Reglas**
      - **Árboles de decisión**
        - **Algoritmos TDIDT** (Top Down Induction Decision Tree, Inducción Descendente de Árboles de Decisión)

Algoritmo TDIDT
Definición

- Es un algoritmo de Aprendizaje Automático inductivo supervisado, sin conocimiento adicional y sin modelo.
- Se utiliza en Minería de Datos, dentro del proceso de KDD (extracción de conocimiento de las bases de datos).

Árbol de decisión
Ejemplo - Enunciado

- Atributos:
  - Edad: menos de 30 años, entre 30 y 60, más de 60.
  - Trastorno refractivo del ojo: miopía, hipermetropía.
  - Presencia de astigmatismo: sí, no.
  - Ritmo de producción de lágrimas: reducida, normal.
- Clase:
  - Lente de contacto recomendada: rígida, blanda, ninguna.
- Experiencias:

	Edad	Trast.	Astig.	Lágr	Clase
e_1	< 30	miop.	sí	norm.	bland.
e_2	< 30	miop.	no	norm.	rígida
$e_{\dots}$	...	...	...	...	...
e_{24}	> 60	miop.	no	reduc.	ning.

Árbol de decisión
Definición del problema

Un conjunto *E* de experiencias (o Ejemplos), cada una de ellas definida por:

- Un conjunto finito de atributos: *A, B, C*, ... con valores discretos (o continuos discretizados) y finitos.
- *p* clases, a priori, en las que clasificar cada experiencia.

Así, para una experiencia concreta *j* perteneciente a *E* tendremos:

j: Am , Bn , Cs , ... / *k* donde *Am , Bn , Cs* son los valores de los atributos *A, B, C* en la experiencia *j*, y *k* es la clase a la que pertenece la experiencia j ($k \in \{1,..., p\}$)

Árbol de decisión
Definición I

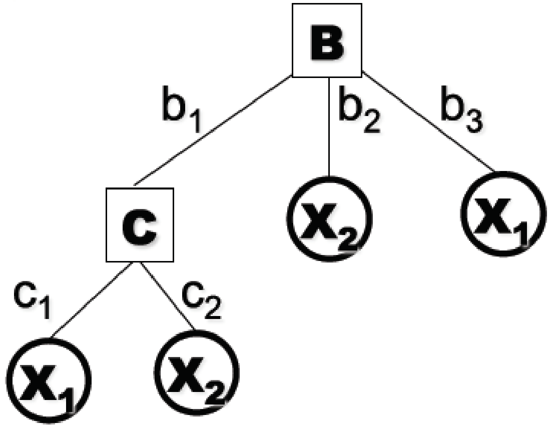
- Un árbol de decisión representa una función que toma un vector de atributos *x*, y devuelve una decisión *y* (el valor de salida, también llamado objetivo).
  - El árbol será establecido de un conjunto de ejemplos {(*x1, y1*)…, (*xn, yn*)} de tamaño N.
  - Tanto las entradas como las salidas pueden ser discretas o continuas.
  - Este tema: nos centraremos en el caso discreto.

Árbol de decisión
Definición II

- Grafo dirigido acíclico etiquetado, donde todos los nodos salvo la raíz tienen un solo padre y pueden tener cero, uno o más hijos.

- Componentes:
  - Nodos (atributos): A, B y C.
  - Arcos (valores de los atributos): a1, a2, b1, b2, c1 y c3.
  - Hojas (clases): x1 y x2.

$$A=\{a_1,a_2\}$$
$$B=\{b_1,b_2,b_3\}$$
$$C=\{c_1,c_2\}$$
$$X=\{x_1,x_2\}$$



https://www.youtube.com/watch?v=gNyroz4Iuso&list=RDCMUC8KCb358oioQMcJ5pUfs8UQ&start_radio=1&rv=gNyroz4Iuso&t=235

https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/

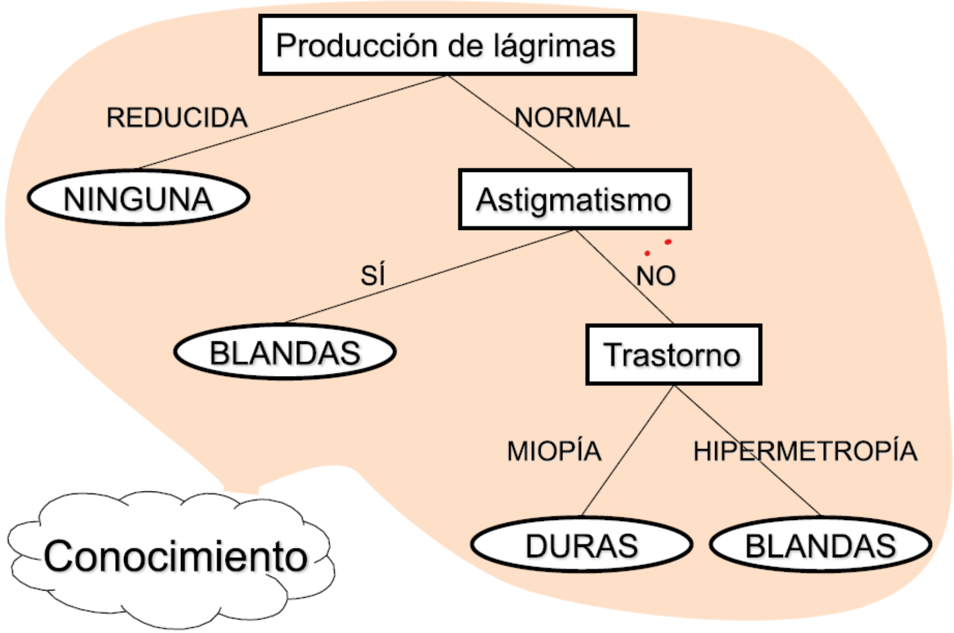
Árbol de decisión
Definición III

- Un árbol de decisión establece su decisión realizando una secuencia:
  - Cada nodo interno evalúa el valor de uno de los atributos de entrada *Ai*.
  - Las ramas del nodo son etiquetadas con los valores posible del atributo $v \in Valores(A)$.
  - Cada nodo hoja especifica un valor y será devuelto por la función.
  - Cada camino que lleva a un nodo hoja puede ser representado por una sentencia (regla) de lógica proposicional. Es decir, **una regla por cada nodo hoja.**

Árbol de decisión
Ejemplo - Solución

- Atributos:
  - Edad: menos de 30 años, entre 30 y 60, más de 60.
  - Trastorno refractivo del ojo, miopía, hipermetropía.
  - Presencia de astigmatismo: sí, no.
  - Ritmo de producción de lágrimas: reducida, normal.
- Clase:
  - Lente de contacto recomendada: rígida, blanda, ninguna.
- Experiencias:

	Edad	Trast.	Astig.	Lágr	Clase
e_1	< 30	miop.	sí	norm.	bland.
e_2	< 30	miop.	no	norm.	rígida
$e_{...}$	...	...	...	...	...
e_{24}	> 60	miop.	no	reduc.	ning.



Representa el conocimiento que ha aprendido el algoritmo de árboles de decisión que se ha Aplicado, a partir de las 24 experiencias.
Se infieren 4 reglas.

Árbol de decisión
Definición

- La **predicción** no tiene por qué ser absoluta:
  - Ejemplo anterior: clase blanda, rígida o ninguna.
  - Ejemplos en general: se muestra un vector de predicción, que indica la probabilidad de pertenencia a cada una de las clases. Ejemplo anterior: v(0.7, 0.1, 0.2): Nota: Las probabilidades siempre suman 1.
- La **importancia** de los atributos aumenta conforme **más cerca están de la raíz**.
- La **complejidad** de los árboles y su **calidad** puede **mejorarse** podándolos (**pruning**):
  - Los atributos más bajos tienen menos importancia.
  - Evita sobreajuste (*overfitting*): evita que el modelo se fije en lo particular.
- Posibles valores sin rama: ciertas combinaciones de atributos no se pueden dar.
  - Ej. Si llueve -suelo mojado

Árbol de decisión
Estructura

Construir nodo inicial: tiene todas las experiencias asociadas a él.

Para cada nodo no analizado hacer:

Si **condición_hoja** entonces: (simple: ej. 80% clase/compleja: poda con control predictivo)

Marcar como hoja.

Etiquetarlo con la clase mayoritaria.

Si no, hacer:

Marcar como no hoja.

Elegir mejor atributo según **Medida** (cuál es el atributo siguiente a escoger).

Etiquetarlo con el atributo elegido.

Expandir nodo: etiquetando cada rama con los diferentes valores del atributo elegido.

Elección del mejor atributo según Medida

- Objetivo: escoger el atributo que menos desordene = más ordene.
- **Entropía**:

La más usada. Es la misma entropía que la entropía de la Teoría de la información de Shannon. La entropía favorece a los atributos con un gran número de valores. Ej. Edad vs sexo.

https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_informaci%C3%B3n

https://es.wikipedia.org/wiki/Entrop%C3%ADa_(informaci%C3%B3n)#:~:text=Shannon%20ofrece%20una%20definici%C3%B3n%20de,debe%20cambiar%20poco%20la%20entrop%C3%ADa.

Para corregir este sesgo se usa otra medida: La Razón de Ganancia

- **Razón de Ganancia**: ganancia = entropía/coeficiente para normalizar.
- En investigación se desarrollan nuevas medidas:
  - LCC: desarrollar nuevas medidas: beta, gamma.

Medida I. Entropía

- Entropía H(*Ejemplos*) es una medida de incertidumbre del conjunto de ejemplos.
  - Se seleccionan los valores bajos de entropía.
  - H(*Ejemplos*)=0 sii todos los ejemplos pertenecen a la misma clase. Ej. Clasificación perfecta.

$$H(Ejemplos) = -\sum_{v\, \in Atributo\ obtejivo} \frac{|Ejemplos\ (v)|}{|Ejemplos|} \log_2 \frac{|Ejemplos\ (v)|}{|Ejemplos|}$$

Medida II. Ganancia de información

- Ganancia de información (Information Gain) IG(*Ejemplos*, *A*) es la diferencia en entropía de *Ejemplos* cuando es dividido por un *Atributo A*. Es decir, cuánto reducimos la incertidumbre en el conjunto de *Ejemplo* después de dividirlo por *A*.
  - Mide qué tan bien un atributo separa los ejemplos de acuerdo a la clase.
  - Elegimos los atributos que tienen mayor ganancia de información.
  - Denotamos con *T* el conjunto de todos los subconjuntos obtenidos mediante la división de *Ejemplos* por *A*.

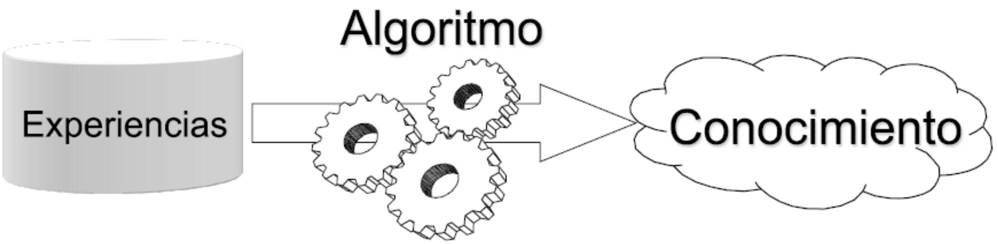
$$G(Ejemplos, A) = \mathrm{H}(Ejemplos) - \sum_{t\,\in\,Humidity} \frac{|t|}{|Ejemplos|} H(Humidity)$$

Terminología en Minería de Datos: Objetivos: clasificar != predecir

CLASIFICAR: COMPRENDER EL PROBLEMA

Queremos obtener un conjunto de reglas que justifique a qué clase pertenece cada experiencia en función de los valores de algunos de sus atributos.

Un sistema de clasificación de experiencias preclasificadas que es más conciso que un listado completo de dichas experiencias. y que aporta además un conocimiento sobre la importancia de los atributos en el problema abordado.

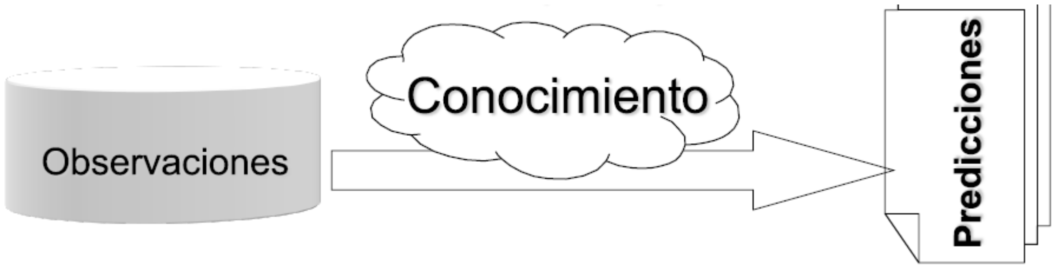


training

PREDECIR FUTURAS OBSERVACIONES

Una observación futura es una experiencia de la que desconocemos su clase.

Predecir consiste en usar el árbol de decisión (modelo obtenido) para asignar clases a observaciones dadas.



Árbol de decisión
Ejemplo – Objetivos

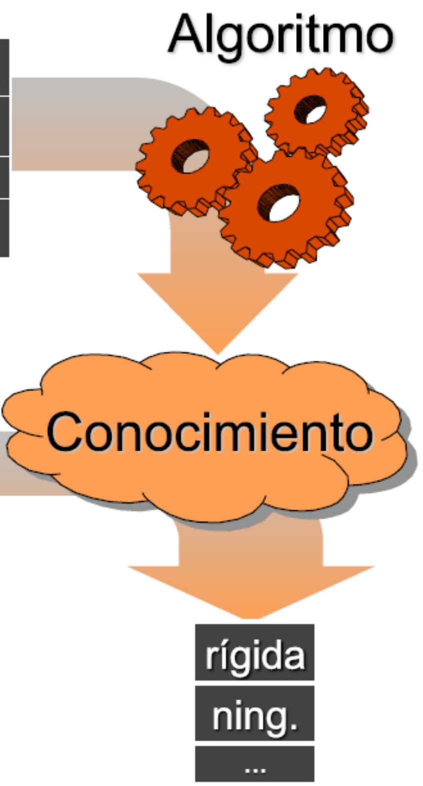
En Minería de Dato se persigue, entre otras cosas, hacer la predicción de la variable de clase sobre observaciones (casos de los que se desconoce la variable de clase).

Experiencias

Edad	Trast.	Astig.	Lágr	Clase
< 30	miop.	sí	norm.	bland.
< 30	miop.	no	norm.	rígida
...	...	...	...	...
> 60	miop.	no	reduc.	ning.

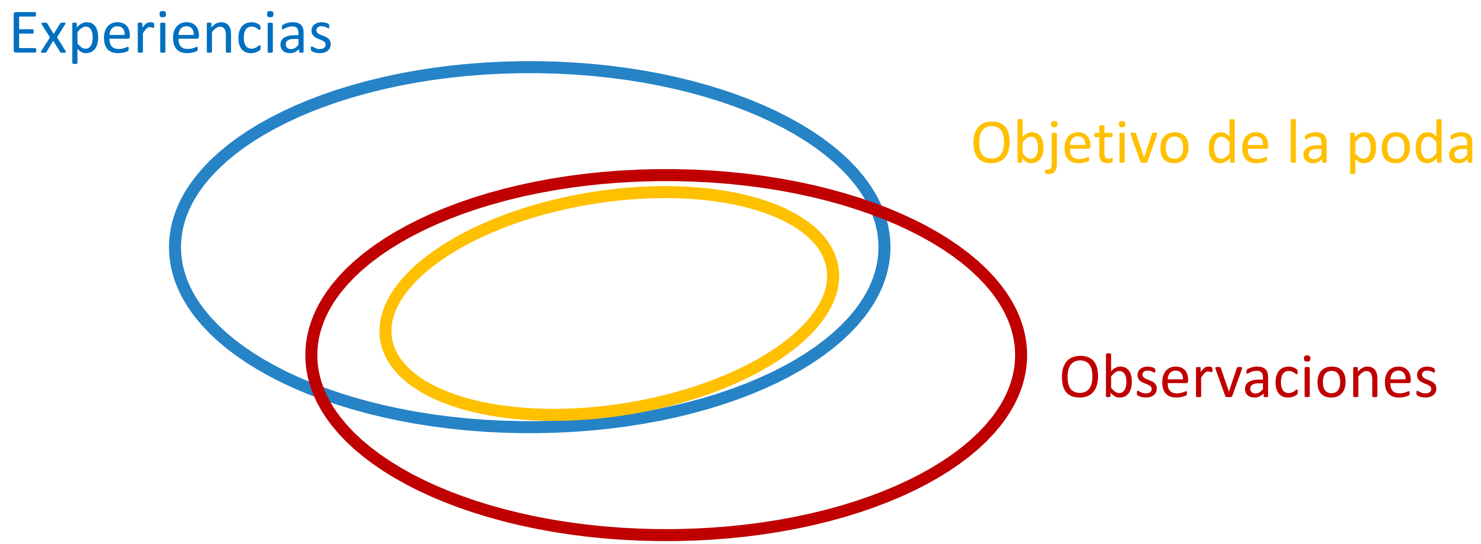
Observaciones

Edad	Trast.	Astig.	Lágr	
30-60	hiper.	no	norm.	?
> 60	miop.	si	reduc.	?
...	...	...	...	...



Técnicas: Poda

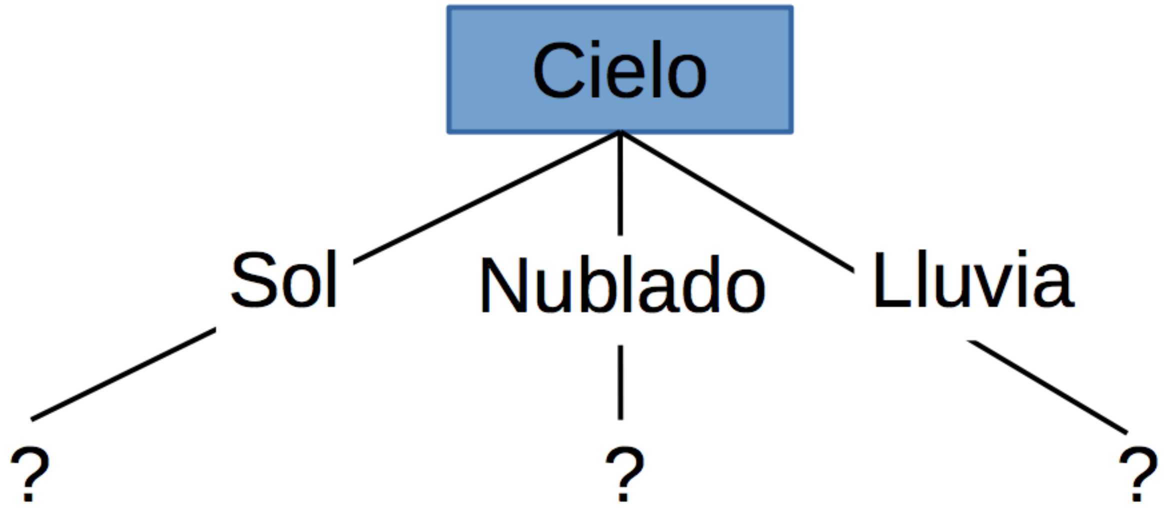
- Tipos de poda:
  - Pre-poda: no se genera el árbol completo. Condición hoja indica la altura de la poda.
  - Post-poda: se genera el árbol completo y se poda a continuación. Más coste computacional.



Técnicas: Binarización

- **¡¡Da mejores resultados!!**
- Llamadas variables Dummy: https://en.wikipedia.org/wiki/Dummy_variable_(statistics)
- Ej: Sin binarizar:
  - Atributos:
    - Edad: menos de 30 años, entre 30 y 60, más de 60.
    - Trastorno refractivo del ojo, miopía, hipermetropía.
    - etc.
- Ej. Binarizado
  - Atributos:
    - Menos de 30 años: sí, no.
    - Entre 30 y 60: sí, no.
    - etc.
    - Cuando binarizamos a variable de clase tenemos que hacer un árbol por cada variable de clase binarizada.

Algoritmo ID3



Ejemplo – Enunciado - Datos

Dada estos datos, generar un árbol de decisión ID3:

Experiencias

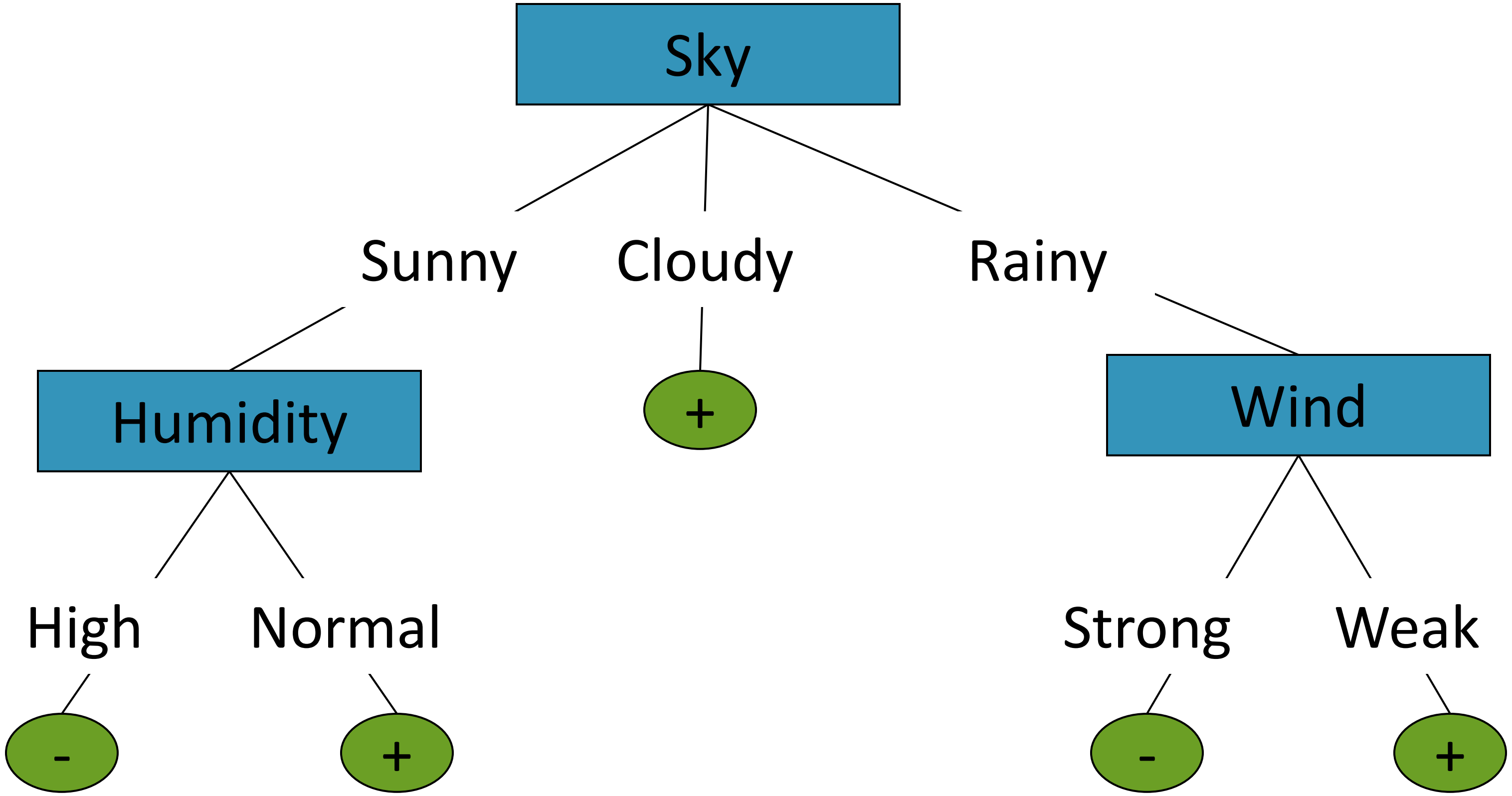
Atributos: Sky, Temperature,

Humidity, Wind

Clase objetivo: Play Tennis

Example	Sky	Temperature	Humidity	Wind	PlayTennis
$\mathbf{x}_1$	Sunny	High	High	Weak	-
$\mathbf{x}_2$	Sunny	High	High	Strong	-
$\mathbf{x}_3$	Cloudy	High	High	Weak	+
$\mathbf{x}_4$	Rainy	Warm	High	Weak	+
$\mathbf{x}_5$	Rainy	Low	Normal	Weak	+
$\mathbf{x}_6$	Rainy	Low	Normal	Strong	-
$\mathbf{x}_7$	Cloudy	Low	Normal	Strong	+
$\mathbf{x}_8$	Sunny	Warm	High	Weak	-
$\mathbf{x}_9$	Sunny	Low	Normal	Weak	+
$\mathbf{x}_{10}$	Rainy	Warm	Normal	Weak	+
$\mathbf{x}_{11}$	Sunny	Warm	Normal	Strong	+
$\mathbf{x}_{12}$	Cloudy	Warm	High	Strong	+
$\mathbf{x}_{13}$	Cloudy	High	Normal	Weak	+
$\mathbf{x}_{14}$	Rainy	Warm	High	Strong	-

Ejemplo - Solución - Árbol de decisión



Descripción algoritmo ID3

- [Quinlan 1983][Quinlan 1986]
- Árbol de decisión basado en 'divide y vencerás'.
- Tiene 3 argumentos de entrada: *Ejemplos*, *Atributo objetivo* y *Atributos*. Devuelve árbol de decisión.
- Si todos los *Ejemplos* tienen el mismo valor objetivo, devolverá un único nodo etiquetado con ese valor.
- Si *Atributos* está vacío, devolverá un único nodo etiquetado con el valor objetivo más frecuente en *Ejemplos*.
- De lo contrario, devolverá el atributo *A*, el que mejor clasifique *Ejemplos*.
- Para cada $v \in Values(A)$:
    - Añadir una nueva rama debajo de la raíz con la etiqueta *v.*
    - Sea *Ejemplos(v)* el conjunto de *Ejemplos con A = v.*
    - Si *Ejemplos(v)* está vacío, entonces añade debajo de la rama un nodo hoja etiquetado con el valor obejtivo más frecuente en *Ejemplos.*
    - En caso contrario, añada debajo la rama del nuevo subárbol ID3*(Ejemplos(v), Atributo Objetivo, Atributos – {A}).*

Ejecución de ID3 en el conjunto de datos

- Analizamos la incertidumbre que hay en el conjunto de datos total:

Entríopa inicial: H({*x1*, ... , *x14*}) = 0.94

Esta entropía será utilizada para escoger el atributo del nodo inicial.

Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_1	Sunny	High	High	Weak	-
x_2	Sunny	High	High	Strong	-
x_3	Cloudy	High	High	Weak	+
x_4	Rainy	Warm	High	Weak	+
x_5	Rainy	Low	Normal	Weak	+
x_6	Rainy	Low	Normal	Strong	-
x_7	Cloudy	Low	Normal	Strong	+
x_8	Sunny	Warm	High	Weak	-
x_9	Sunny	Low	Normal	Weak	+
x_{10}	Rainy	Warm	Normal	Weak	+
x_{11}	Sunny	Warm	Normal	Strong	+
x_{12}	Cloudy	Warm	High	Strong	+
x_{13}	Cloudy	High	Normal	Weak	+
x_{14}	Rainy	Warm	High	Strong	-

Desarrollo matemático llevado a cabo:

$$\text{H}(Ejemplos) = -\sum_{v\,\in Atributo\ obtejivo} \frac{|Ejemplos\ (v)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (v)|}{|Ejemplos|} =$$

$$= -\left(\frac{|Ejemplos\ (+)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (+)|}{|Ejemplos|} + \frac{|Ejemplos\ (-)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (-)|}{|Ejemplos|}\right) =$$

$$= -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) = 0{,}94$$

Escogemos un atributo para el nodo raíz: *Humidity*

Siendo $Ejemplos$ = {x1, x2, …, x14}

$$\mathrm{G}(Ejemplos, Humidity) = \mathrm{H}(Ejemplos) - \sum_{t \in Humidity} \frac{|t|}{|Ejemplos|} \mathrm{H}(Humidity) =$$

$$= 0{,}94 - \left(\frac{7}{14} * \left(-\left(\frac{3}{7}\log_2\frac{3}{7} + \frac{4}{7}\log_2\frac{4}{7}\right)\right) + \frac{7}{14} * \left(-\left(\frac{6}{7}\log_2\frac{6}{7} + \frac{1}{7}\log_2\frac{1}{7}\right)\right) = 0{,}151$$

Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_1	Sunny	High	High	Weak	-
x_2	Sunny	High	High	Strong	-
x_3	Cloudy	High	High	Weak	+
x_4	Rainy	Warm	High	Weak	+
x_5	Rainy	Low	Normal	Weak	+
x_6	Rainy	Low	Normal	Strong	-
x_7	Cloudy	Low	Normal	Strong	+
x_8	Sunny	Warm	High	Weak	-
x_9	Sunny	Low	Normal	Weak	+
x_{10}	Rainy	Warm	Normal	Weak	+
x_{11}	Sunny	Warm	Normal	Strong	+
x_{12}	Cloudy	Warm	High	Strong	+
x_{13}	Cloudy	High	Normal	Weak	+
x_{14}	Rainy	Warm	High	Strong	-

Escogemos un atributo para el nodo raíz: *Wind*

Siendo $Ejemplos$ = {x1, x2, …, x14}

$$G(Ejemplos, Wind) = H(Ejemplos) - \sum_{t \in Wind} \frac{|t|}{|Ejemplos|} H(Wind) = 0{,}048$$

Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_1	Sunny	High	High	Weak	-
x_2	Sunny	High	High	Strong	-
x_3	Cloudy	High	High	Weak	+
x_4	Rainy	Warm	High	Weak	+
x_5	Rainy	Low	Normal	Weak	+
x_6	Rainy	Low	Normal	Strong	-
x_7	Cloudy	Low	Normal	Strong	+
x_8	Sunny	Warm	High	Weak	-
x_9	Sunny	Low	Normal	Weak	+
x_{10}	Rainy	Warm	Normal	Weak	+
x_{11}	Sunny	Warm	Normal	Strong	+
x_{12}	Cloudy	Warm	High	Strong	+
x_{13}	Cloudy	High	Normal	Weak	+
x_{14}	Rainy	Warm	High	Strong	-

Escogemos un nodo para la raíz: *Sky*

Siendo $Ejemplos$ = {x1, x2, …, x14}

G($Ejemplos$, *Sky*) = H($Ejemplos$) $- \sum_{t \in Sky} \frac{|t|}{|Ejemplos|}$ H(*Sky*) = 0,246

Nótese que H(*Cloudy*) = 0

Ya que todos los ejemplos pertenecen a la misma clase, a la clase +.

Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_1	Sunny	High	High	Weak	-
x_2	Sunny	High	High	Strong	-
x_3	Cloudy	High	High	Weak	+
x_4	Rainy	Warm	High	Weak	+
x_5	Rainy	Low	Normal	Weak	+
x_6	Rainy	Low	Normal	Strong	-
x_7	Cloudy	Low	Normal	Strong	+
x_8	Sunny	Warm	High	Weak	-
x_9	Sunny	Low	Normal	Weak	+
x_{10}	Rainy	Warm	Normal	Weak	+
x_{11}	Sunny	Warm	Normal	Strong	+
x_{12}	Cloudy	Warm	High	Strong	+
x_{13}	Cloudy	High	Normal	Weak	+
x_{14}	Rainy	Warm	High	Strong	-

Escogemos un nodo para la raíz: *Temperature*

Siendo $Ejemplos$ = {x1, x2, …, x14}

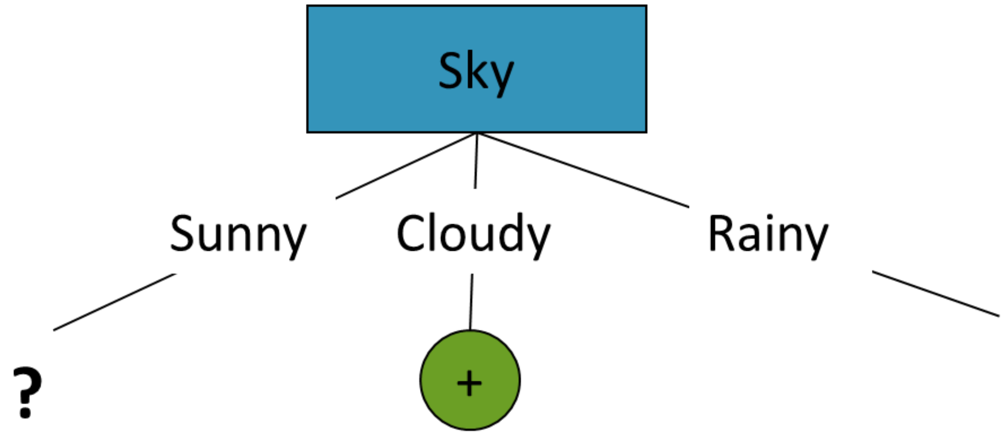
$$\mathrm{G}(Ejemplos, Temperature) = \mathrm{H}(Ejemplos) - \sum_{t\,\in Temperature} \frac{|t|}{|Ejemplos|}\mathrm{H}(Temperature) = 0{,}029$$

Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_1	Sunny	High	High	Weak	-
x_2	Sunny	High	High	Strong	-
x_3	Cloudy	High	High	Weak	+
x_4	Rainy	Warm	High	Weak	+
x_5	Rainy	Low	Normal	Weak	+
x_6	Rainy	Low	Normal	Strong	-
x_7	Cloudy	Low	Normal	Strong	+
x_8	Sunny	Warm	High	Weak	-
x_9	Sunny	Low	Normal	Weak	+
x_{10}	Rainy	Warm	Normal	Weak	+
x_{11}	Sunny	Warm	Normal	Strong	+
x_{12}	Cloudy	Warm	High	Strong	+
x_{13}	Cloudy	High	Normal	Weak	+
x_{14}	Rainy	Warm	High	Strong	-

- ¿Qué atributo proporciona la mejor ganancia? (qué atributo ordena más los ejemplos)

✓ *Sky*

- Por tanto, se divide el conjunto de ejemplos {x1, x2, …, x14} según el atributo *Sky*.
- Para el caso de *Sky=Cloudy* el atributo objetivo es siempre (+), por tanto se crea un nodo hoja en el árbol.



- Para el nodo *Sky* =*Sunny* la entropía es:

Siendo Sky_{Sunny} = {x1, x2, x8, x9, x11}

$$H(Sky_{Sunny}) = -\left(\frac{|Ejemplos\ (+)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (+)|}{|Ejemplos|} + \frac{|Ejemplos\ (-)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (-)|}{|Ejemplos|}\right) =$$

$$-\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0{,}971$$

Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_1	Sunny	High	High	Weak	-
x_2	Sunny	High	High	Strong	-
x_8	Sunny	Warm	High	Weak	-
x_9	Sunny	Low	Normal	Weak	+
x_{11}	Sunny	Warm	Normal	Strong	+

- Probamos los diferentes atributos para Sky=Sunny:
  - G(Sky_{Sunny}, *Temperature*) = 0,571
  - G(Sky_{Sunny}, *Humidity*) = 0,971
  - G(Sky_{Sunny}, *Wind*) = 0,020

- Escogemos el atributo *Humidity* por tener la mayor ganancia.

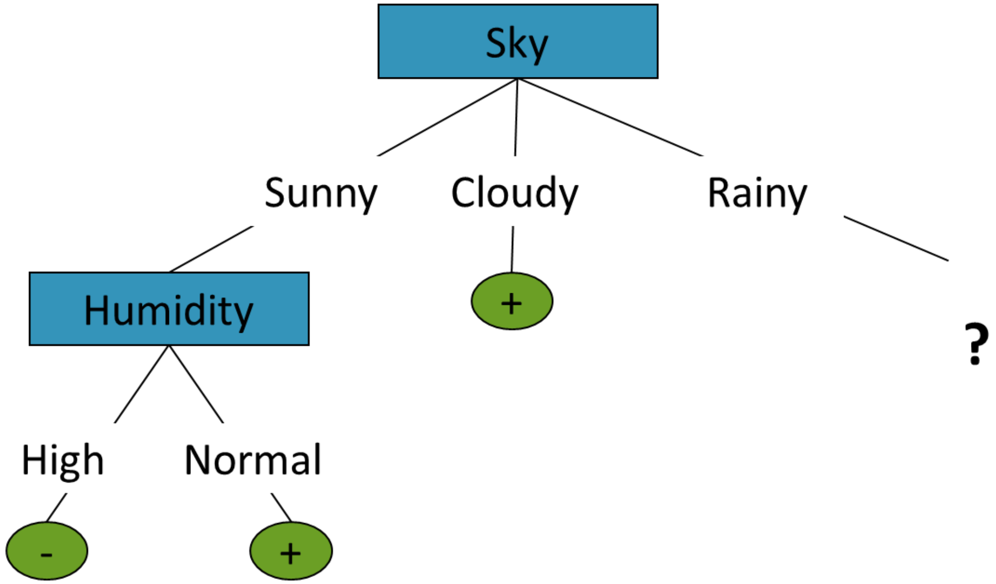
- ¿Qué atributo proporciona la mejor ganancia? (qué atributo ordena más los ejemplos)

✓ *Humidity*

- Por tanto, se divide el conjunto de ejemplos Sky_{Sunny} = {x1, x2, x8, x9, x11} según el atributo *Humidity*.

- Para el caso de *Humidity* = *Hight* el atributo objetivo es siempre (-), por tanto se crea un nodo hoja en el árbol.

- Para el caso de *Humidity* = *Normal* el atributo objetivo es siempre (+), por tanto se crea un nodo hoja en el árbol.



- Para el nodo Sky =Rainy la entropía es:

Siendo Sky_{Rainy} = {x4, x5, x6, x10, x14}

$$H(Sky_{Rainy}) = -\left(\frac{|Ejemplos\ (+)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (+)|}{|Ejemplos|} + \frac{|Ejemplos\ (-)|}{|Ejemplos|}\log_2\frac{|Ejemplos\ (-)|}{|Ejemplos|}\right) =$$

$$-\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0{,}971$$

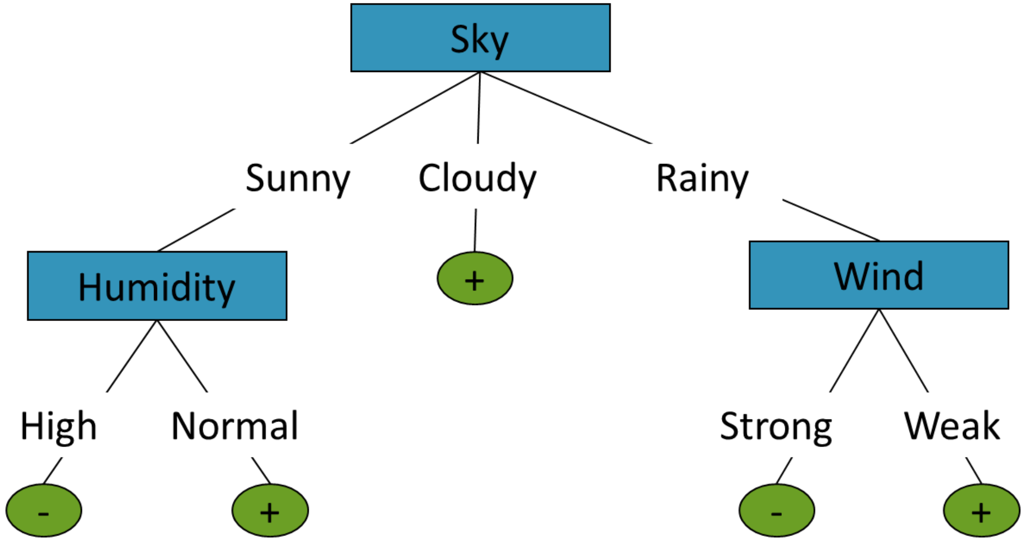
Example	Sky	Temperature	Humidity	Wind	PlayTennis
x_4	Rainy	Warm	High	Weak	+
x_5	Rainy	Low	Normal	Weak	+
x_6	Rainy	Low	Normal	Strong	-
x_{10}	Rainy	Warm	Normal	Weak	+
x_{14}	Rainy	Warm	High	Strong	-

- Probamos los diferentes atributos para Sky=Sunny:
  - G(Sky_{Rainy}, *Temperature*) = 0,020
  - G(Sky_{Rainy}, *Humidity*) = 0,020
  - G(Sky_{Rainy}, *Wind*) = 0,971
- Escogemos el atributo *Wind* por tener la mayor ganancia.

- ¿ Qué atributo proporciona la mejor ganancia? (qué atributo ordena más los ejemplos)

✓ *Wind*

- Por tanto, se divide el conjunto de ejemplos Sky_{Rainy} = {x4, x5, x6, x10, x14} según el atributo *Wind*.
- Para el caso de *Wind* = *Strong* el atributo objetivo es siempre (-), por tanto se crea un nodo hoja en el árbol.
- Para el caso de *Wind* = *Weak* el atributo objetivo es siempre (+), por tanto se crea un nodo hoja en el árbol.



Descripción algoritmo C4.5

- [Quinlan 1993], Evolución del ID3.
- Permite trabajar con atributos numéricos:
  - Atributos continuos.
  - Separa los valores en dos ramas a partir de un umbral.
- Árboles menos frondosos:
  - Cada hoja no cubre una clase, sino una distribución de clases.
- Utiliza estrategia de Primero en profundidad (Deep-first).
- Antes de cada partición de datos, el algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información o en la mayor proporción de ganancia de información.
  - Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo.
  - Para cada atributo continuo se realiza una prueba binaria sobre cada uno de los valores que toma el atributo en los datos.
- La implementación en **Weka** de este árbol de decisión de aprendizaje es el algoritmo **J4.8**.

Otros algoritmos

CART

[Breiman et al. 1984]

Binario: árbol profundo

J48

[Quinlan 1993], Evolución del ID3

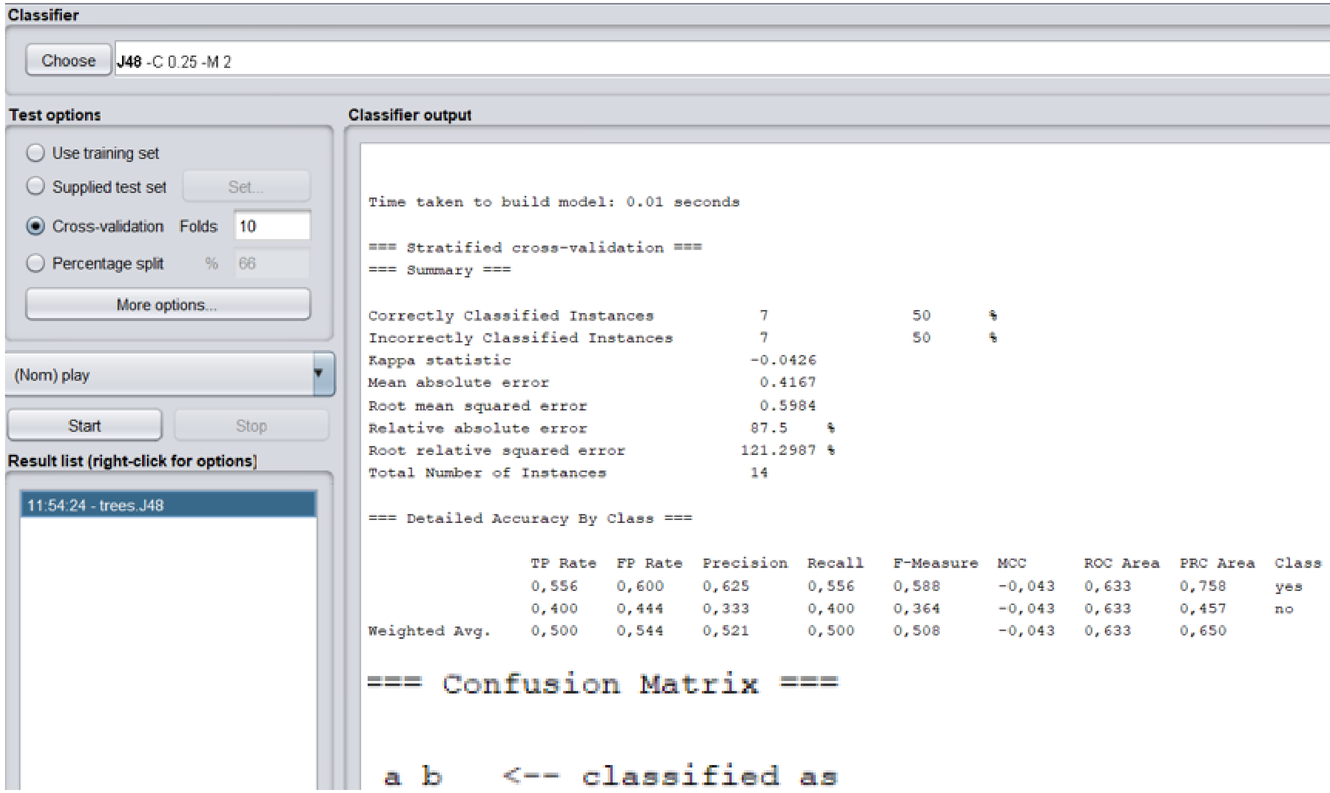
N-ario: árbol menos profundo

Medidas clasificación

Ejemplo en WEKA

- Datos weather.nominal.arff
- Algoritmo J48
- Evaluación de los datos con Validación cruzada K=10
- Prestamos atención a la matriz de confusión:
  - 5 de la clase *a* bien clasificados.
  - 4 de la clase *a* clasificados como *b*.
  - 2 de la clase *b* bien clasificados.
  - 3 de la clase *b* clasificados como *a*.

		Clase actual	
		a	b
Clase predicha	a	5	3
	b	4	2



En Weka se muestra así:

```
 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

Matriz de confusión

- La matriz de confusión tiene tamaño C x C. Cada elemento de la matriz (i, j) indica el número de ejemplos de la clase j que han sido clasificados como i.
- El rendimiento de la clase j puede ser evaluado estudiando la columna j de la matriz.

		Clase Actual		
Clase Predicha		1	…	C
	1	n_{11}	…	n_{1C}
	…	…		…
	C	n_{C1}	…	n_{CC}

Medidas de clasificación generales

- **Accuracy** (precisión): es el número de ejemplos correctamente clasificados por el modelo dividido por el número de ejemplos totales.
- **Rand index**: es el número de pares de ejemplos que han sido correctamente clasificados en la misma clase, más el número de pares de ejemplos que han sido correctamente clasificados en las diferentes clases, dividido por el número de pares de ejemplos.
  - Ambas medidas se encuentran en el intervalo [0, 1], y cuanto mayor es el valor mejor (1 significa la clasificación perfecta).

Medidas de clasificación binarias

- Para un problema de clasificación binaria (variable de clase con 2 categorías). Una clase se devine como **clase positiva** y la otra clase se define como **clase negativa**, así se define:
- Verdadero positivo (VP) = n11
- Verdadero negativo (VN) = n22
- Falso positivo (FP, también llamado Error tipo I) = n12
- Falso negativo (FN, también llamado Error tipo II) = n21
- Así, el accuracy es:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

	Clase actual			
Clase predicha		1	…	C
	1	n_{11}	…	n_{1C}
	…	…		…
	C	n_{C1}	…	n_{CC}

Medidas de clasificación binarias

- **Precision**: valores predichos positivos, cuanto mayor mejor.
- **Fallout** (ratio falso positivo): cuanto más bajo mejor.
- **Recall** (sensibilidad o ratio de verdadero positivo):

cuanto más alto mejor.

- **F-measure**: cuanto más alto mejor

$$Precision = \frac{TP}{TP + FP}$$

$$Fallout = \frac{FP}{FP + TN}$$

$$Recall = \frac{TP}{TP + FN}$$

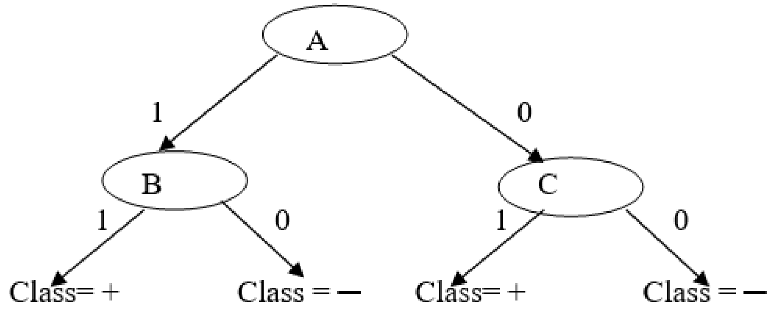
$$F - measure = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

Ejercicio - enunciado

- Calcula el accuracy, fallout, recall y F-mearuse del árbol de decisión dado con éstos datos de test.

Exercise 7:

Consider the following decision tree:



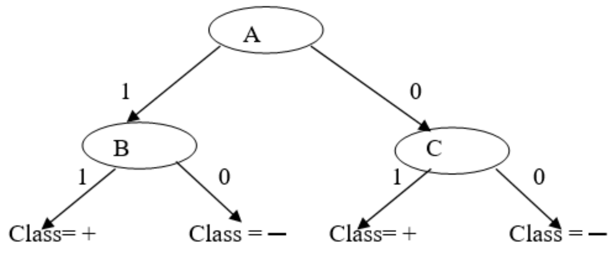
Consider the following test data:

A	B	C	Class
1	1	1	+
1	1	0	+
1	0	1	+
1	0	0	+
0	1	1	+
0	1	0	–
0	0	1	–
0	0	0	–

Ejercicio - solución

- Calcula el accuracy, fallout, recall y F-mearuse del árbol de decisión dado con éstos datos de test.

Exercise 7:



A	**B**	**C**	**Predicted Class**		**True Class**
1	1	1	+	tp	+
1	1	0	+	tp	+
1	0	1	—	fn	+
1	0	0	—	fn	+
0	1	1	+	tp	+
0	1	0	—	tn	—
0	0	1	+	fp	—
0	0	0	—	tn	—

$tp = 3$
$tn = 2$
$fp = 1$
$fn = 2$

So

accuracy = $(3+2)/8 = 0.625$
precision = $3/(3+1) = 0.750$
fallout = $1/(1+2) = 0.333$
recall = $3/(3+2) = 0.600$
f-measure = $2 \times 0.750 \times 0.600/(0.750 + 0.600) = 0.666$