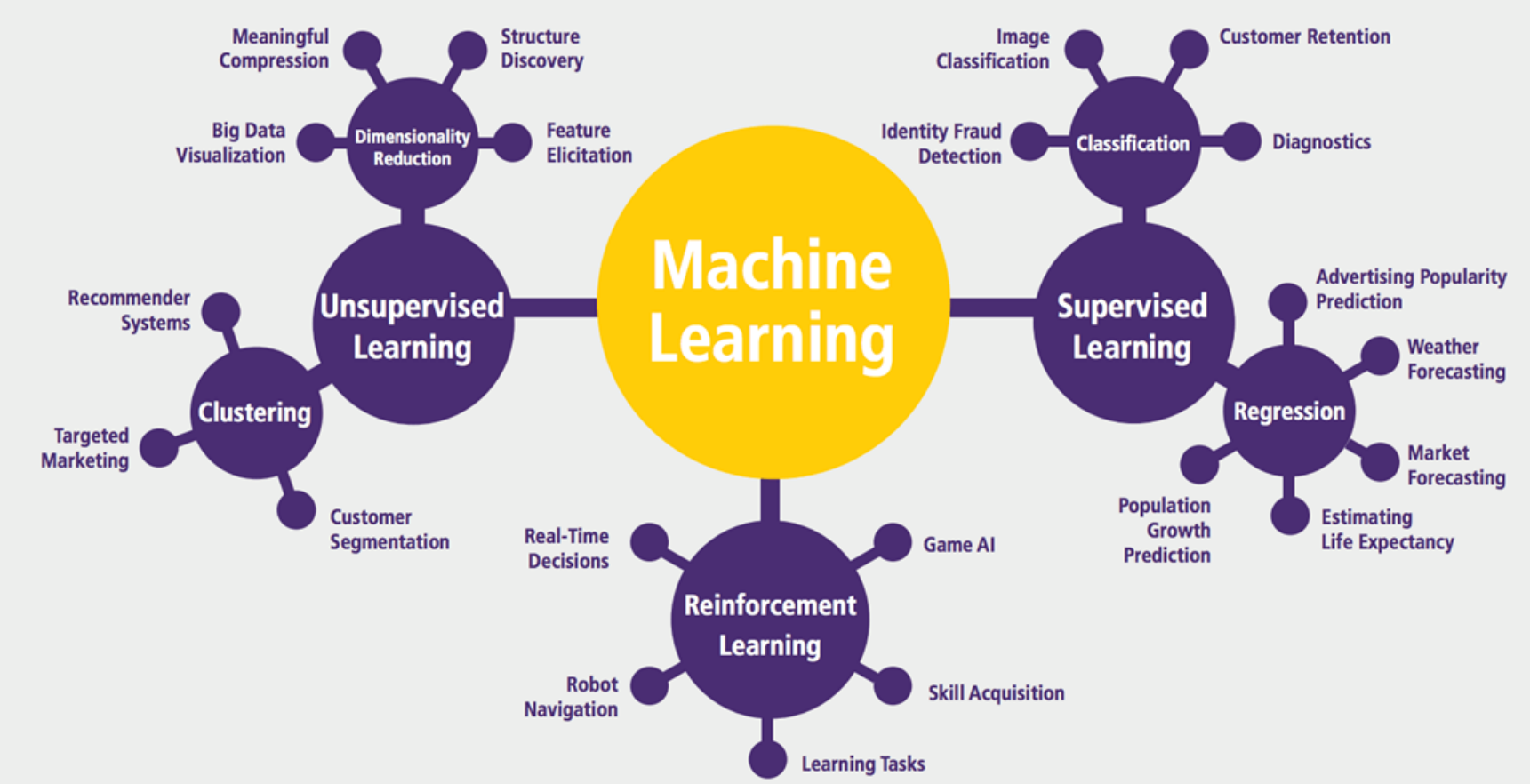
Aprendizaje Automatizado

Árboles de Clasificación o Decisión

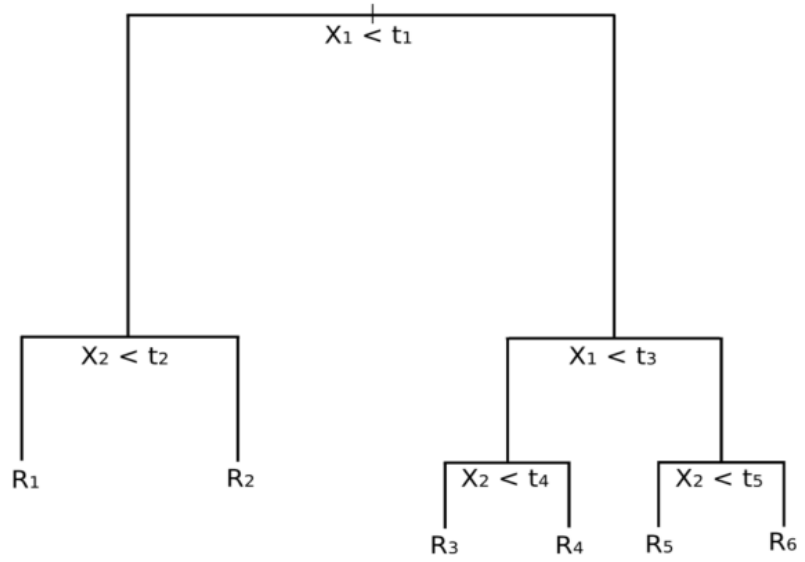
Aprendizaje Automatizado



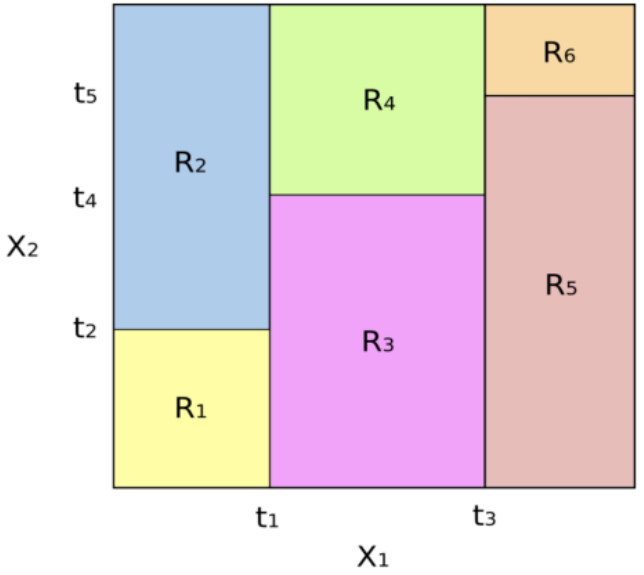
ÁRBOLES DE DECISIÓN

Árboles de Decisión

- Cómo dividir una región en bloques…



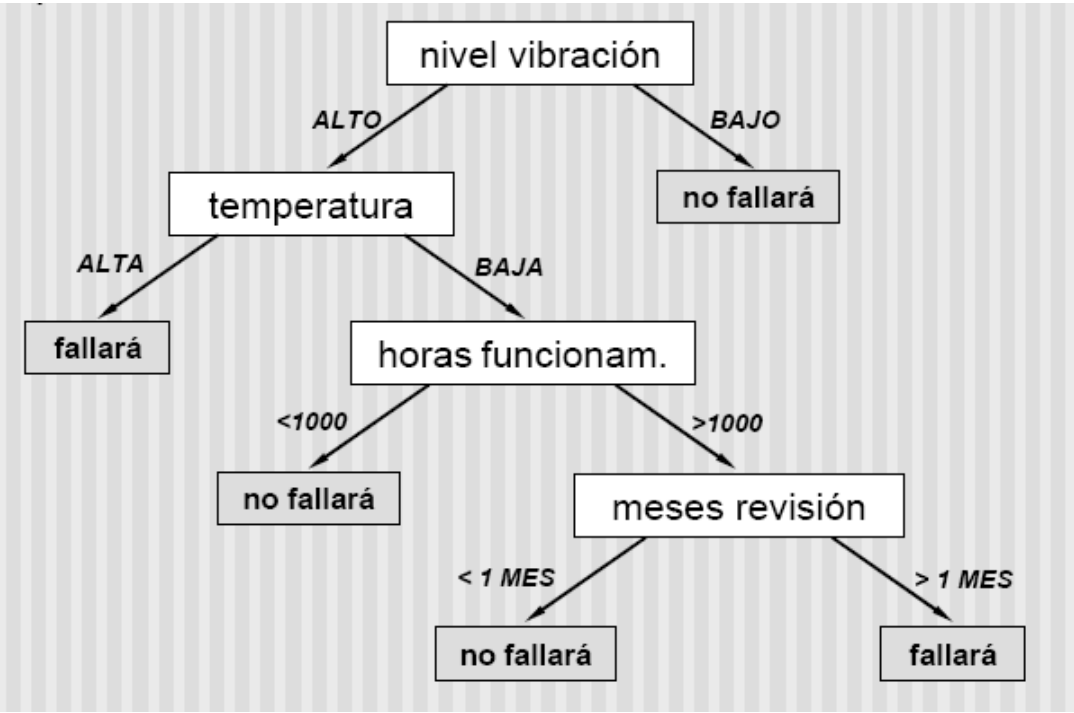
A Decision Tree with six separate regions



The resulting partition of the subset of $\mathbb{R}^2$ into six regional "blocks"

Árboles de Decisión

- Ejemplo: modelado de la posible falla de una máquina.



Árboles de Decisión

- Compuestos de nodos y ramas.
- Representan reglas lógicas (if - then).
- Nodos internos = atributos (atributo-valor).
- Nodos hoja = clases.
- Nodo raíz = nodo superior del árbol.

➢ Objetivo en AA: Obtener un árbol de decisión (resultado) a partir de un conjunto de instancias o ejemplos.

➢ Bias: árbol mínimo

Árboles de Decisión

- Ejemplo de un conjunto de entrenamiento.

Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará

Aprendizaje de Arboles de Decisión (DTL)

- Método para aproximar funciones de clasificación (variable objetivo toma valores discretos)
- Uno de los métodos de aprendizaje inductivos más conocidos
- Actualmente se utilizan ensambles de árboles de decisión (conjuntos de árboles-bosques)

Aprendizaje de Arboles de Decisión (DTL)

- Estudiaremos algoritmos para la creación de árboles a partir de datos.
  - Objetivo de obtener los árboles más pequeños
- Selección de atributos comenzando en el nodo raíz.
  - Utilizando la ganancia de información
- Proceso recursivo.

Árboles de Decisión

- Ejemplo de un conjunto de entrenamiento.

Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará

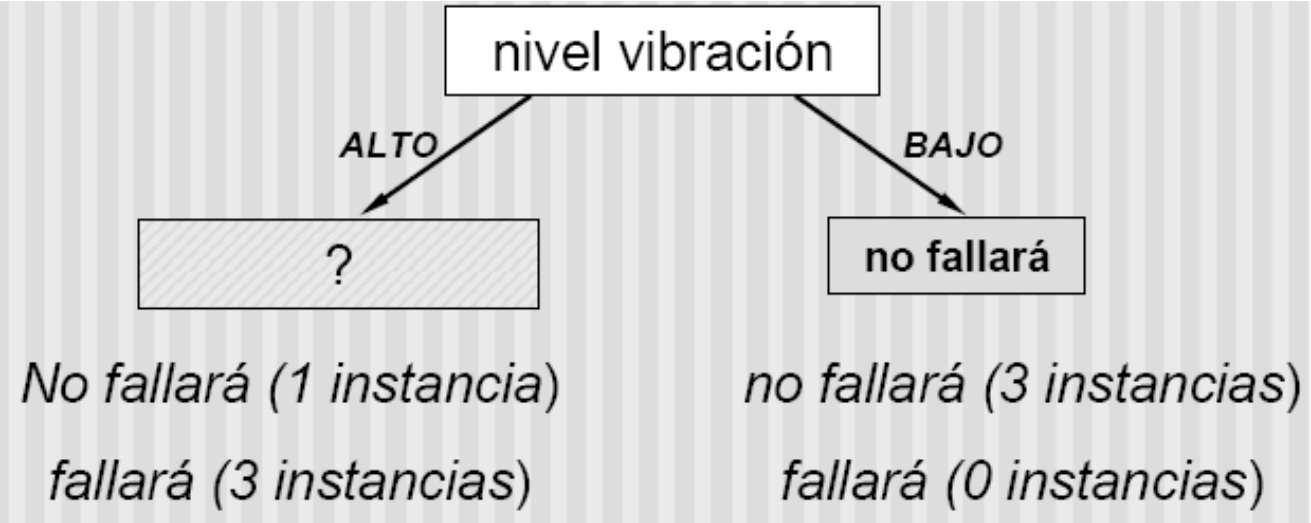
Árboles de Decisión

Crearemos un árbol a partir de los ejemplos de entrenamiento anteriores. ¿Qué atributo elegir para el primer nodo?

ATRIBUTO	VALORES	CLASE	
		fallará	*no fallará*
Temperatura	Alto	2	2
	Bajo	1	2
Nivel de vibraciones	Alto	3	1
	Bajo	0	3
Horas defuncionamiento	< 1000	2	3
	>1000	1	1
Meses desde revisión	> 1 mes	2	3
	< 1 mes	1	1

Árboles de Decisión

- Árbol construido hasta el momento:



- Qué atributo usamos en el siguiente nivel del árbol (rama izquierda)?

Aprendizaje de Arboles de Decisión (DTL)

- Entrada: conjunto de entrenamiento, objetos caracterizables mediante propiedades (pares atributos-valor),
  - La función objetivo toma valores discretos
  - Los datos de entrenamiento pueden contener errores
- Salida: un árbol de clasificación (nodos hojas en una clase)
  - En árboles de decisión: una decisión (sí o no).
- Conjunto de reglas equivalentes.

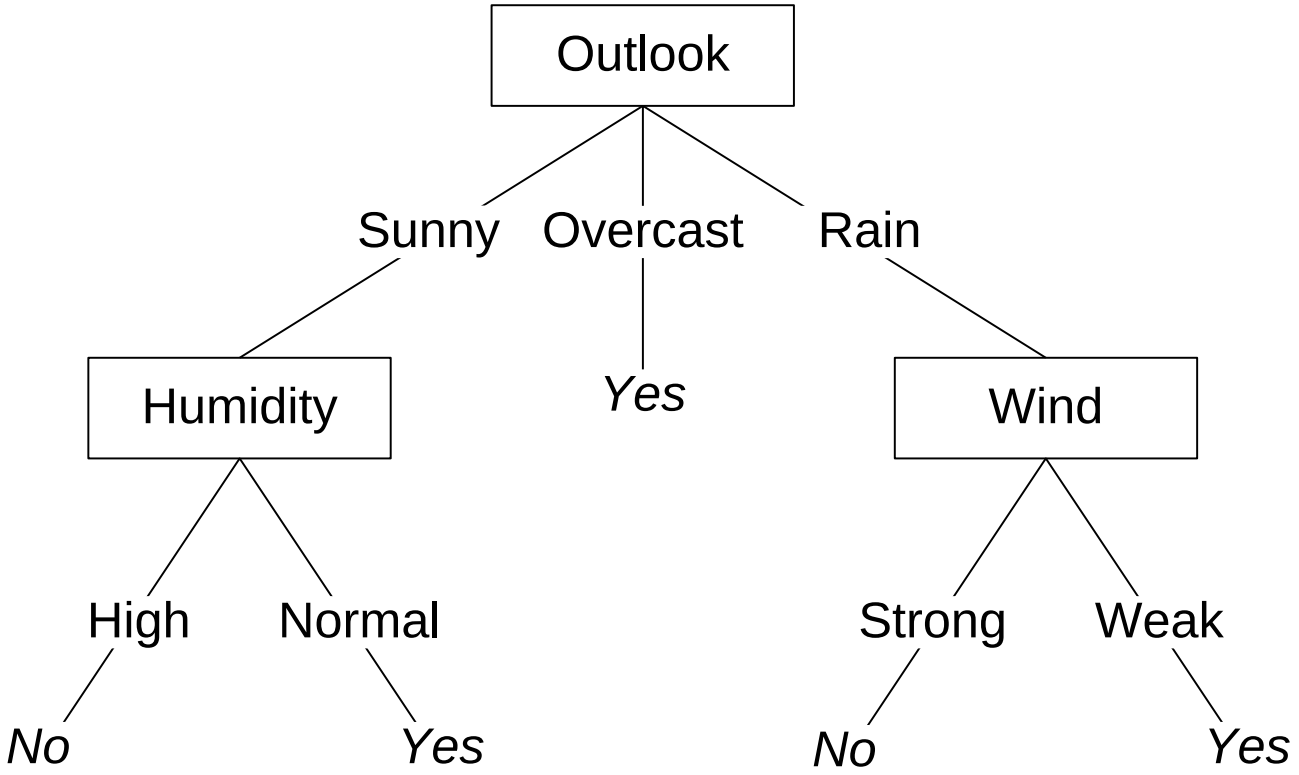
Árboles de Clasificación

- Se clasifican las instancias desde la raíz hacia las hojas, las cuales proveen la clasificación.
- Cada nodo especifica el test de algún atributo.
- Ejemplo: Si
  - (Temp=ALTA, NivelVib=ALTO, Horas=800, Meses rev=2meses) ….**Fallará?**
  - (Outlook = Sunny, Humedity = High, Temperature = Hot, Wind = Strong)….Juego al tenis?

Play Tennis

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Play Tennis


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Play Tennis

- Reglas: Disyunción de conjunciones:

IF
(Outlook = Sunny **And** Humidity = Normal)
Or (Outlook = Overcast)
Or (Outlook = Rain **And** Wind = Weak)

THEN
Yes (Play Tennis)

Problemas Apropiados

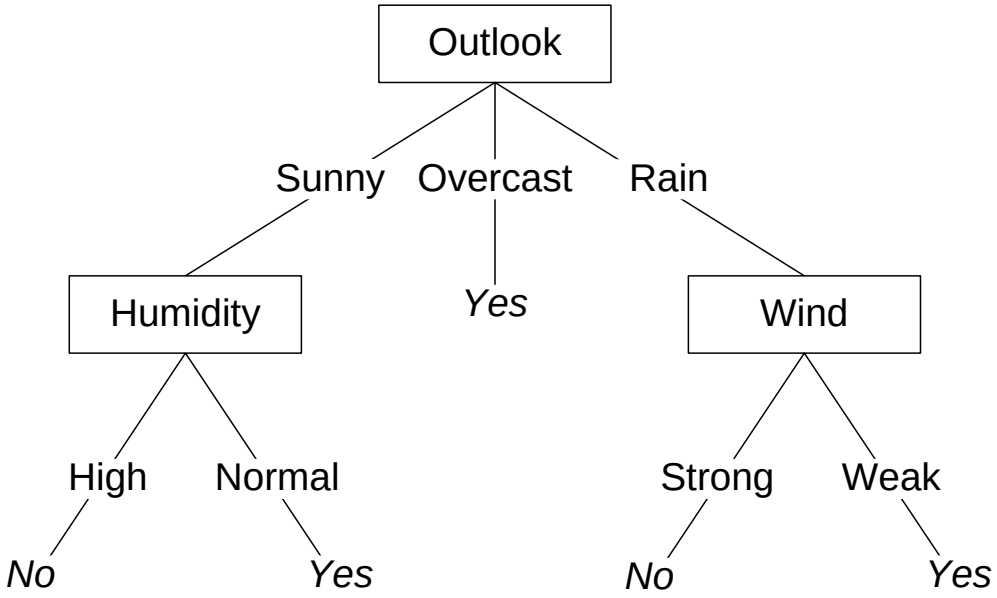
- Las instancias pueden ser representadas por pares (atributo, valor).
- La función objetivo tiene valores discretos (o pueden ser discretizados).
- Nos interesa tener un conjunto de reglas lógicas de clasificación.
- Posiblemente existen errores en los datos de entrenamiento (robustos al ruido).
- Posiblemente falta información en algunos de los datos de entrenamiento.

Algoritmo básico para obtener un árbol de decisión (DTL)

- Encontrar **el árbol mas chico** es un problema NP complete (Quinlan 1986), por lo cual estamos forzados a usar algún algoritmo local de búsqueda para encontrar soluciones razonables.
- La elección de nodos más altos se basa en **ganancia de información**.
  - También se usa el método de Gini….

Algoritmo básico para obtener un árbol de decisión (DTL)

- Start, progress, stop



Algoritmo básico para obtener un árbol de decisión

- Un árbol puede ser "aprendido" mediante el fraccionamiento del conjunto inicial en subconjuntos basados en una prueba de valor de atributo.
- Este proceso se repite en cada subconjunto derivado de una manera **recursiva** llamada particionamiento recursivo.
- La recursividad termina cuando el subconjunto en un nodo tiene todo el mismo valor de la variable objetivo, o cuando la partición ya no agrega valor a las predicciones (condición de parada).

Algoritmo básico para obtener un árbol de decisión

- Búsqueda local, en profundidad (de arriba hacia abajo), a través del espacio de posibles árboles de decisión (ID3: Iterative Dichotomiser 3 y C4.5).
- Raíz: el atributo que mejor clasifica los datos

  Cuál atributo es el mejor clasificador?

  $\Rightarrow$ respuesta basada en la **ganancia de información.**

Algoritmo básico para obtener un árbol de decisión

- Hay **ganancia de información** cuando la división envía instancias con clases distintas a los distintos nodos.
- El atributo que permite obtener mayor ganancia de información es el seleccionado para dividir el nodo.

Algoritmo ID3

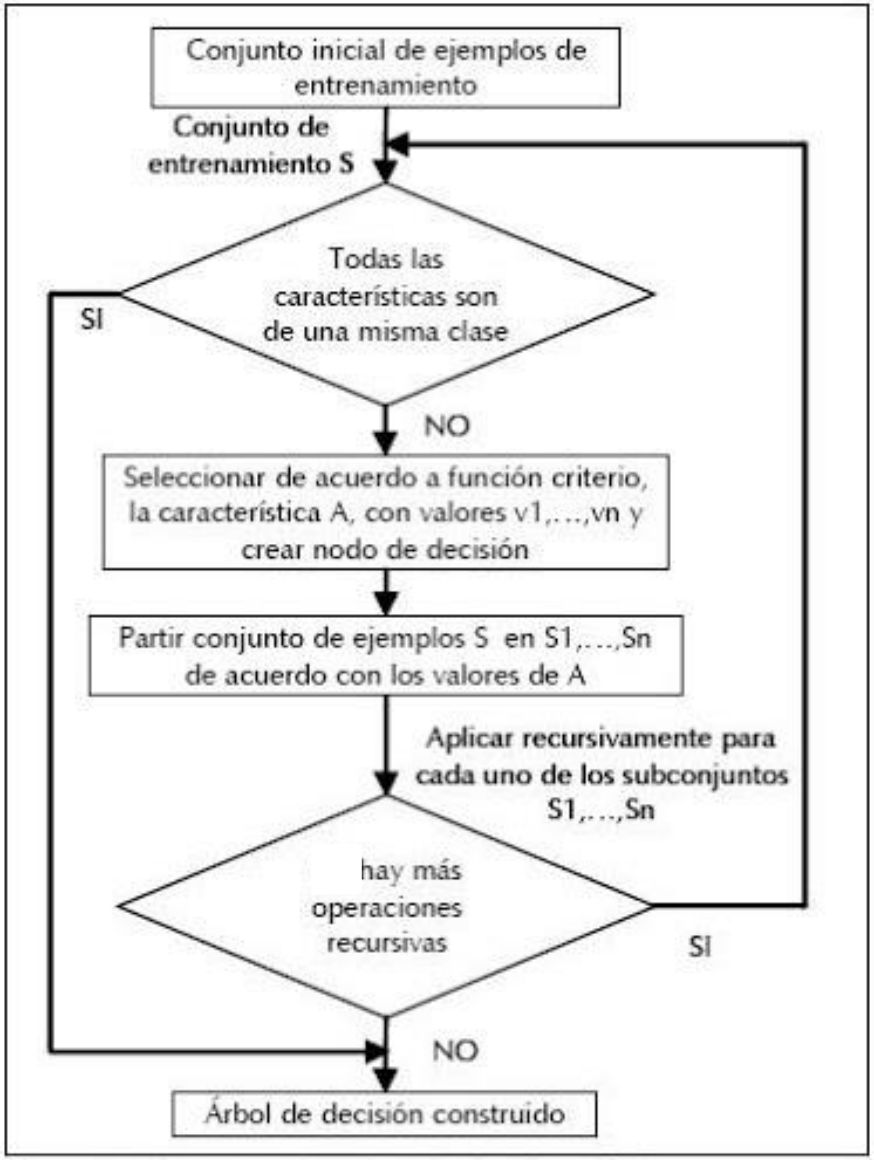
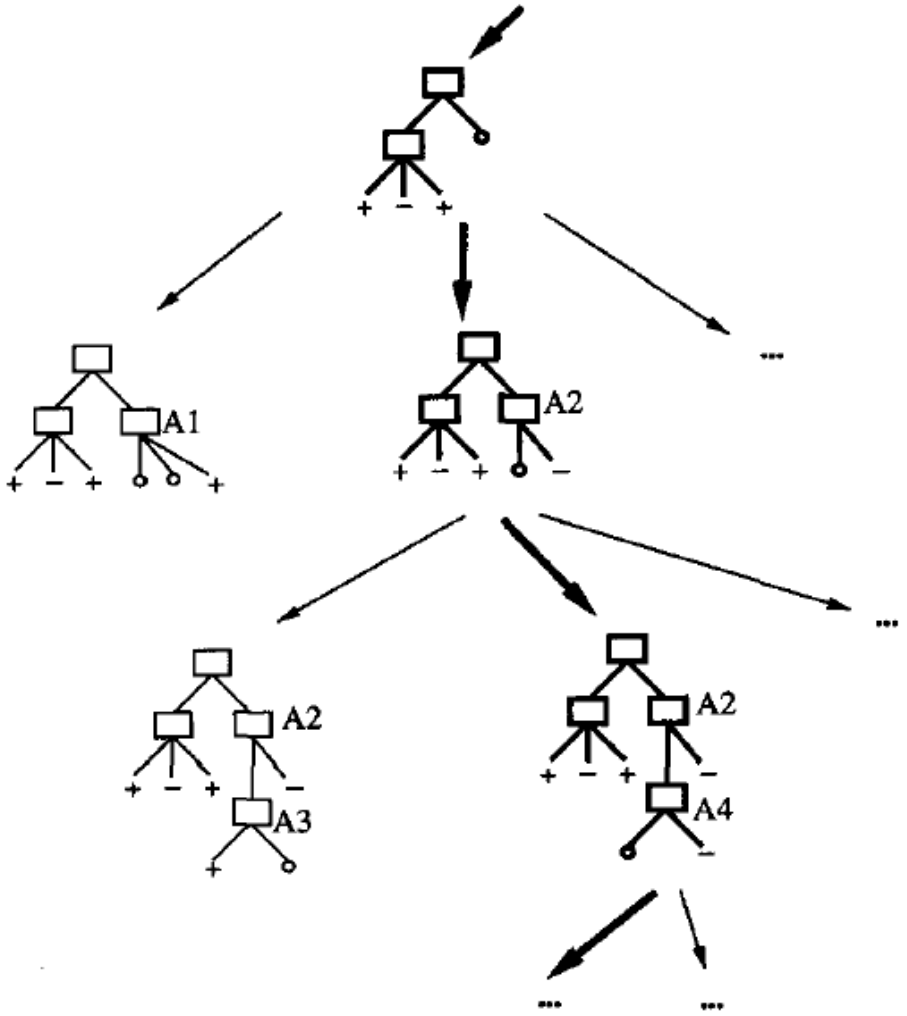


Figura 1. Diagrama de flujo del algoritmo ID3

Espacio de búsqueda de hipótesis en el algoritmo de DTL

- El espacio de hipótesis (modelo) para ID3 es el conjunto de todos los árboles posibles.
- ID3 realiza una búsqueda hill-climbing de lo simple a complejo, no hace backtracking
- Comienza con el árbol vacío y utiliza para la evaluación la ganancia de información



Algoritmo: ID3 (Interactive Dichotomizer Version 3)

- Ganancia de información

Entropía

Es la medida de la incertidumbre que hay en un sistema. Es decir, ante una determinada situación, la Probabilidad de que ocurra cada uno de los posibles resultados.

Algoritmo: ID3

- Entropía

$$Entropía(S) \equiv - p\oplus \log_2 p\oplus - p\ominus \log_2 p\ominus$$

S: conjunto de datos actual.
$p\oplus$ = proporción de ejemplos positivos.
$p\ominus$ = proporción de ejemplos negativos.

Por ejemplo, en el conjunto de datos Play Tennis
$p\oplus = 9/14$, $p\ominus = 5/14$ y $E(S) = 0.94$

En general: $Entropía(S) = - \sum_{i=1,c} pi \log_2 pi$

PlayTennis
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No

Algoritmo: ID3

- Por ejemplo:

Si S1 es el subconjunto de S en el cual Humedity = High

Entonces:

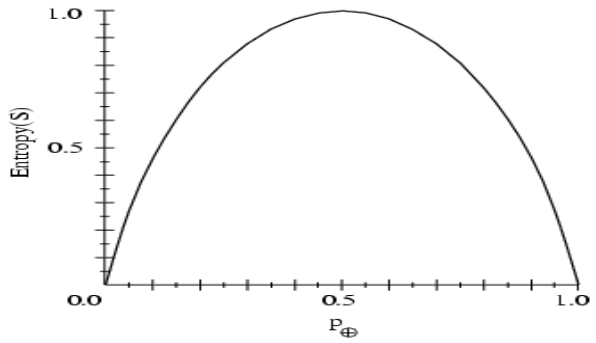
- $p\oplus = 3/7$
- $p\ominus = 4/7$

- Entropía(S1) =
  $-3/7 \log_2 3/7 - 4/7 \log_2 4/7 = 0.985$

Humidity	*Wind*	*PlayTennis*
High	Weak	No
High	Strong	No
High	Weak	Yes
High	Weak	Yes
Normal	Weak	Yes
Normal	Strong	No
Normal	Strong	Yes
High	Weak	No
Normal	Weak	Yes
Normal	Weak	Yes
Normal	Strong	Yes
High	Strong	Yes
Normal	Weak	Yes
High	Strong	No

Entropía y proporción de positivos

Entropy



- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$Entropy(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Ganancia de información

- Mide la reducción esperada de entropía sabiendo el valor del atributo A

$$Gain(S,A) \equiv Entropía(S) - \sum_{v \in Valores(A)} (|Sv|/|S|)Entropía(Sv)$$

Valores(A): Conjunto de posibles valores del atributo A
Sv: Subconjunto de S en el cual el atributo A tiene el valor *v*

Ej: Gain(S, Humedad) = 0.940 - (7/14) 0.985 - (7/14) 0.592

Ent (Sha) Ent (Shn)

proporción de humedad alta (ha)

proporción de humedad normal (hn)

Otras medidas para decidir...

- Ganancia de información

$$Gain(S,A) \equiv Entropía(S) - \sum_{v \in Valores(A)} (|Sv|/|S|)Entropía(Sv)$$

- Impureza de Gini (métodos estadísticos)

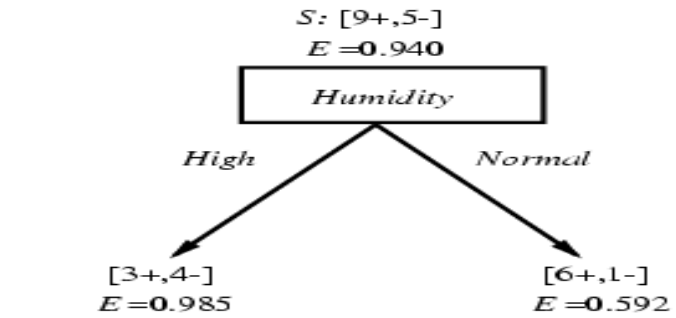
  se puede calcular sumando la probabilidad de cada elemento siendo elegido multiplicado por la probabilidad de un error en la categorización de ese elemento. Alcanza su mínimo (cero) cuando todos los casos del nodo corresponden a una sola categoría de destino.

- Reducción de la varianza

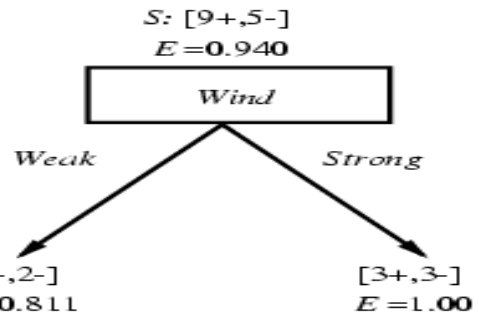
Play Tennis

Selecting the Next Attribute

Which attribute is the best classifier?



Gain (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151



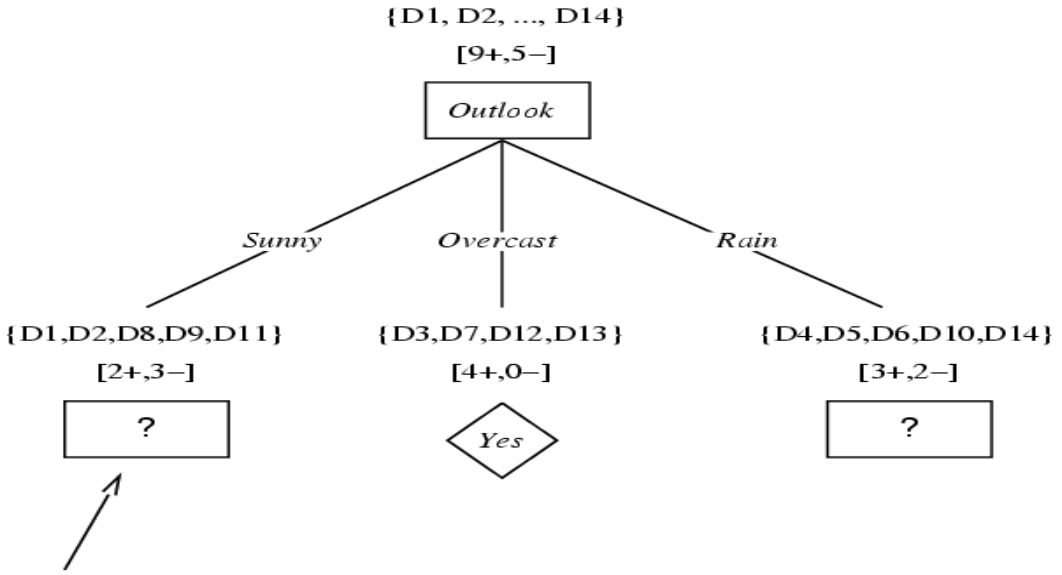
Gain (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048

Play Tennis

- Gain(S,Outlook) = 0.246
- Gain(S,Humidity) = 0.151
- Gain(S,Wind) = 0.048
- Gain(S,Temperature) = 0.029

$\Rightarrow$ Outlook es el atributo del nodo raíz.

Play Tennis



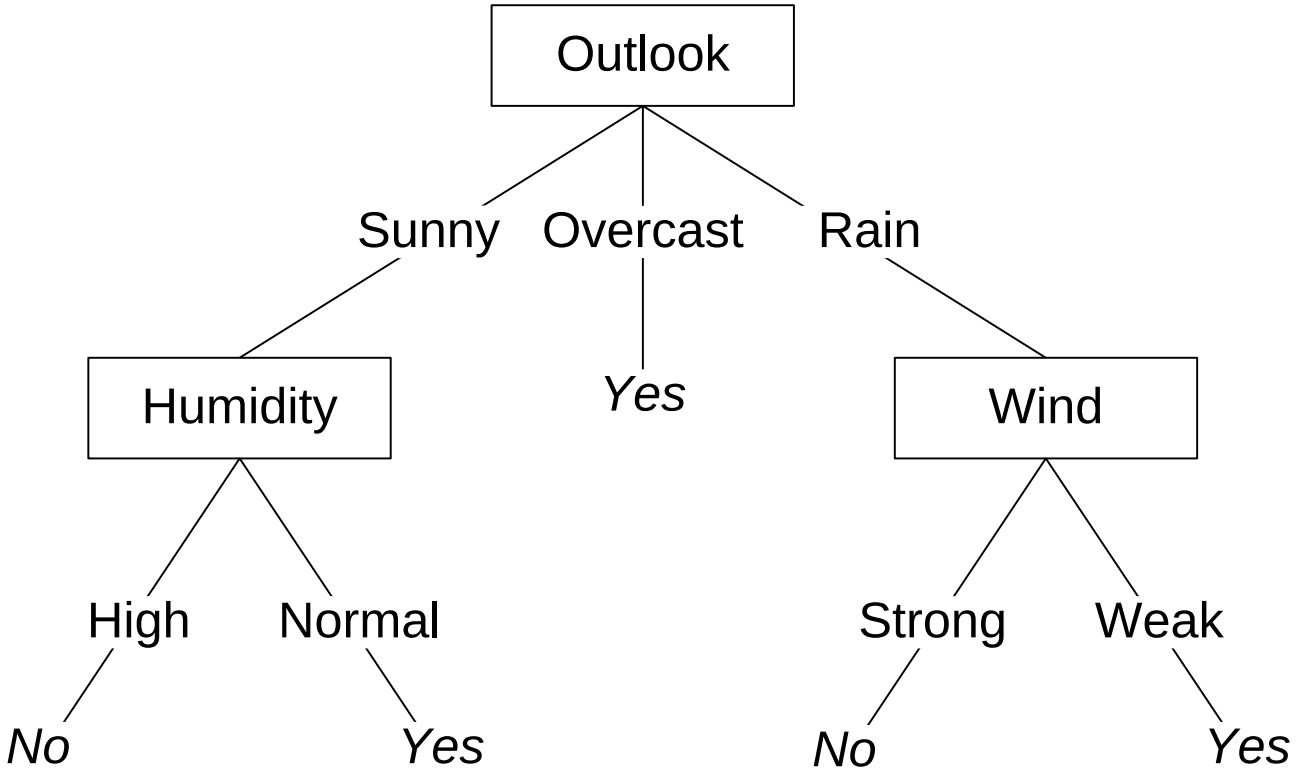
S_{sunny} = {D1,D2,D8,D9,D11}

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Play Tennis


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Algoritmo básico para obtener un árbol de decisión

- El algoritmo ID3 se aplica a atributos discretos.
  - En cada nodo queda seleccionado un atributo y un valor (ej. temperatura = alta).
- El algoritmo C4.5 además se puede aplicar a atributos continuos (se discretizan).
  - En cada nodo queda seleccionado un atributo y un umbral para realizar la división (ej. temperatura > 26).
  - Requiere un mínimo de ganancia en cada nodo

Algoritmo básico para obtener un árbol de decisión

- ID3 nunca produce árboles demasiado grandes.
- C4.5 sí, pues puede repetir atributos (temp < 26, temp > 24, temp < 25, etc).
- Un árbol demasiado grande puede producir sobreajuste (*overfitting*).
- Es necesario podar los árboles (*pruning*).

Overfitting (Sobreentrenamiento)

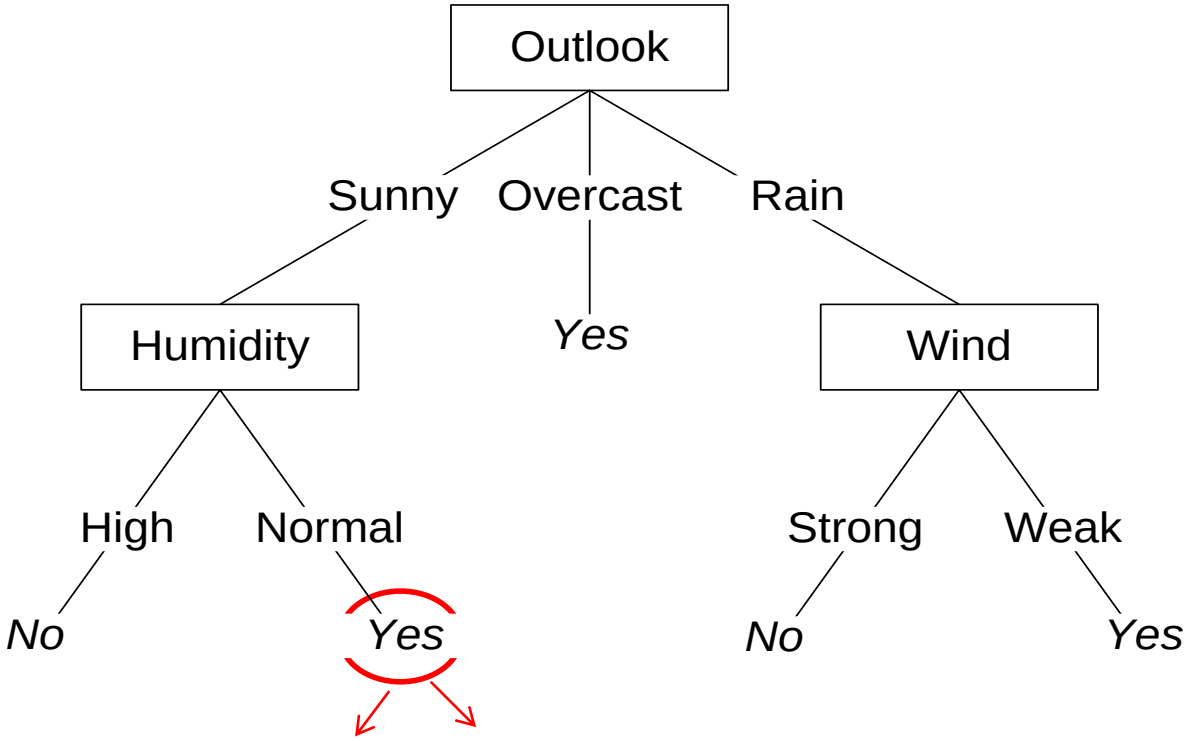
Por qué overfitting?

Un modelo puede ser más complejo
de lo que la función objetivo (generalización)
debe ser, cuando también **trata de satisfacer datos ruidosos** (lectura ruidosa, muestra chica)

Por ej: instancia etiquetada incorrectamente como Negativo: (Sunny; Hot; Normal; Strong; PlayTennis = No)

Overfitting

D15: (Sunny; Hot; Normal; Strong; PlayTennis = No)

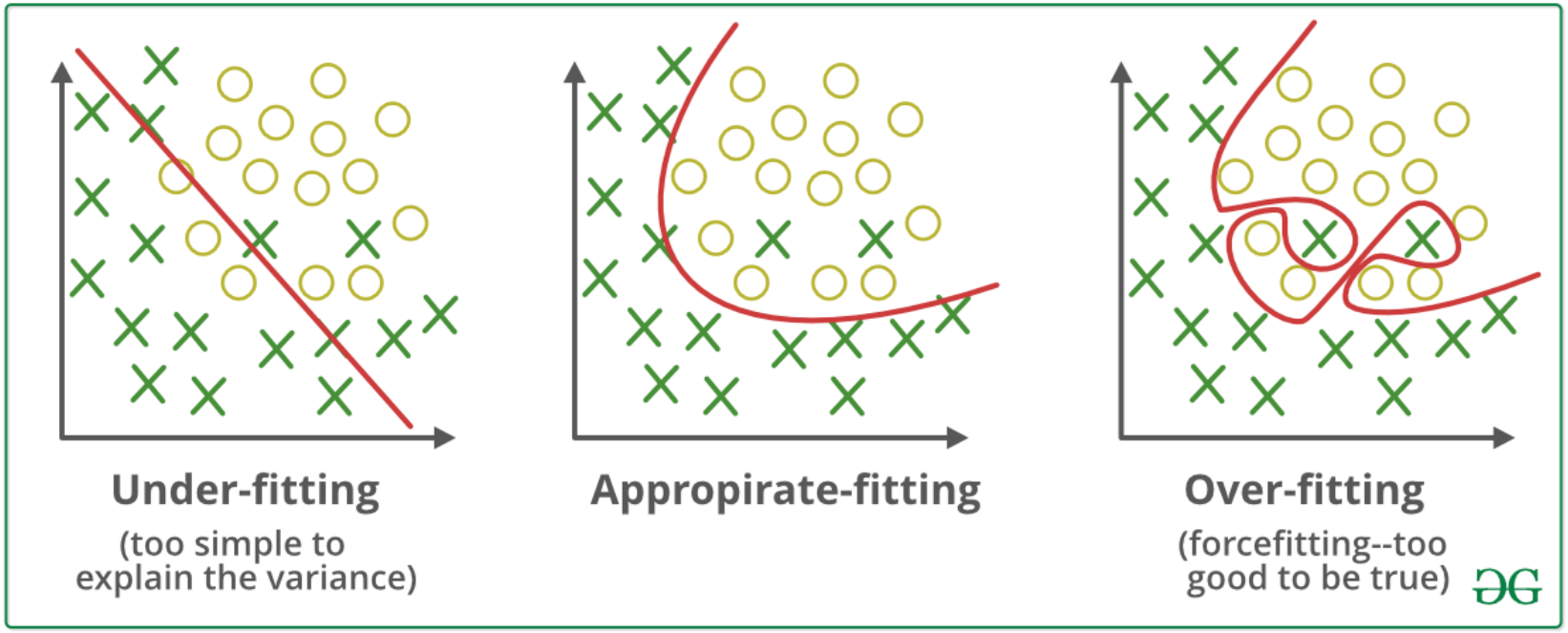


Sobreentrenamiento

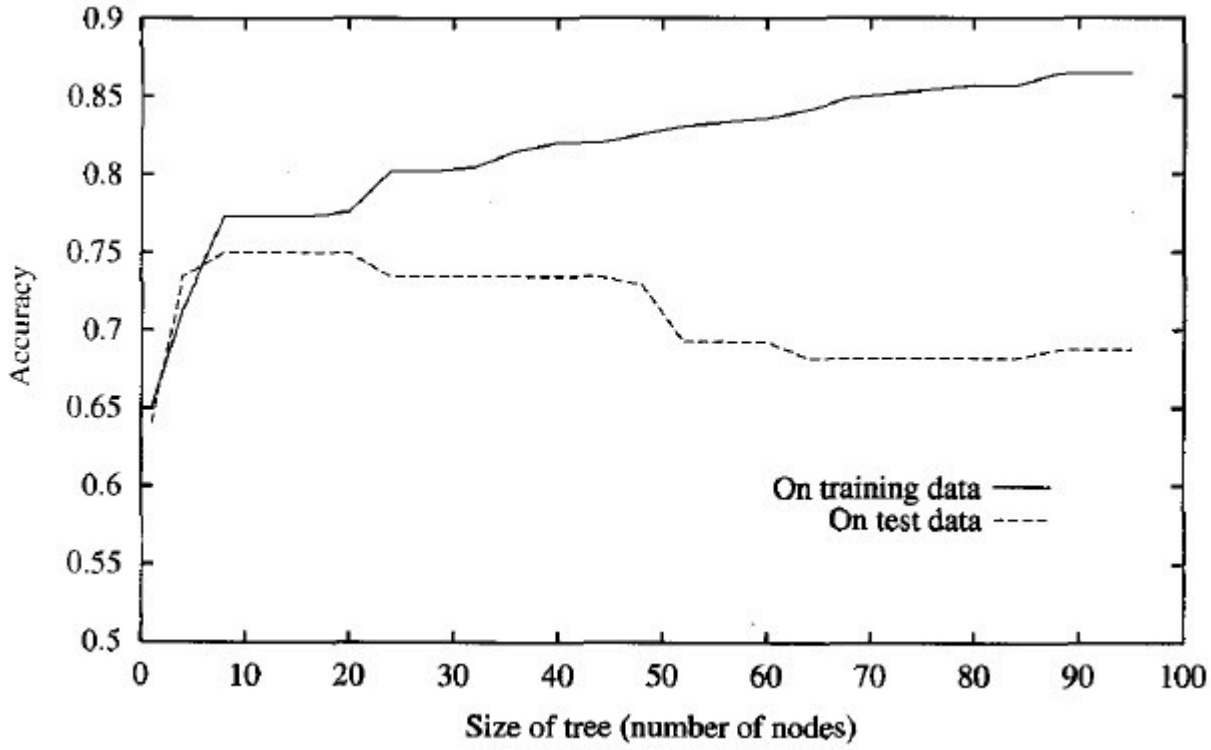
Definición de overfitting

En una hipótesis (modelo) se dice que existe sobreentrenamiento si existe alguna otra hipótesis que tiene **mayor** error sobre los datos de entrenamiento pero **menor** error sobre todos los datos.

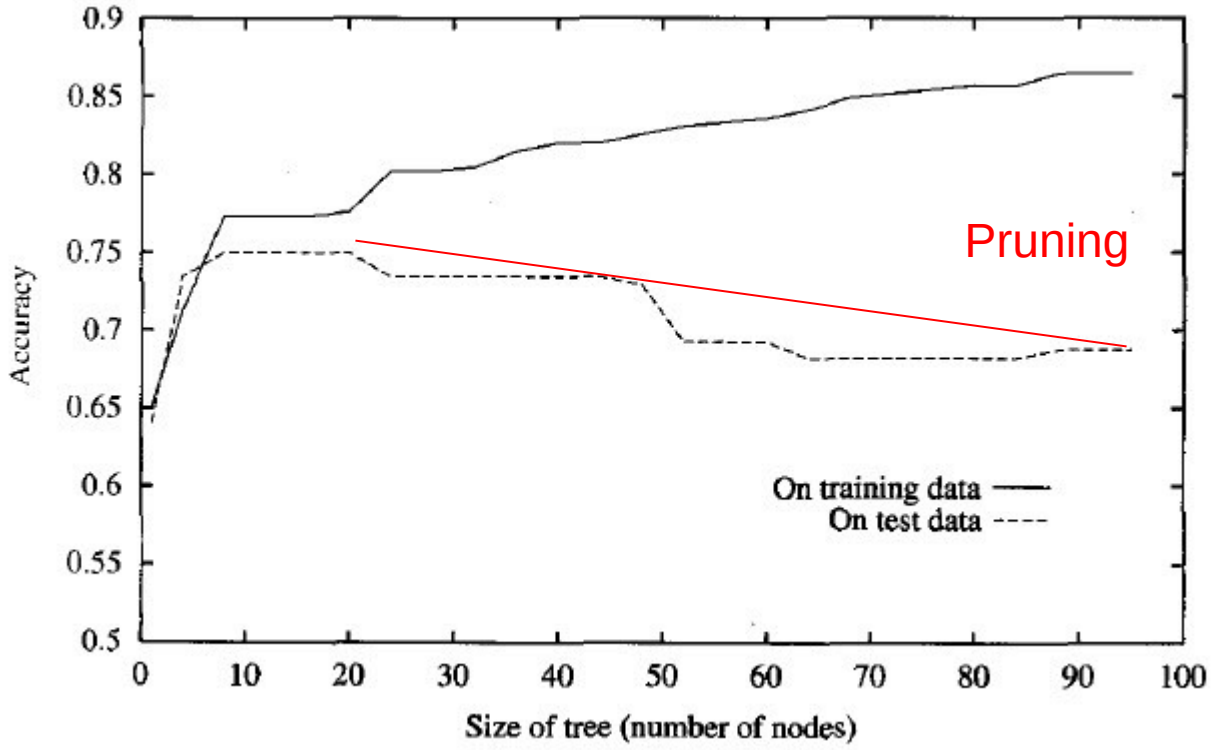
Sobreentrenamiento



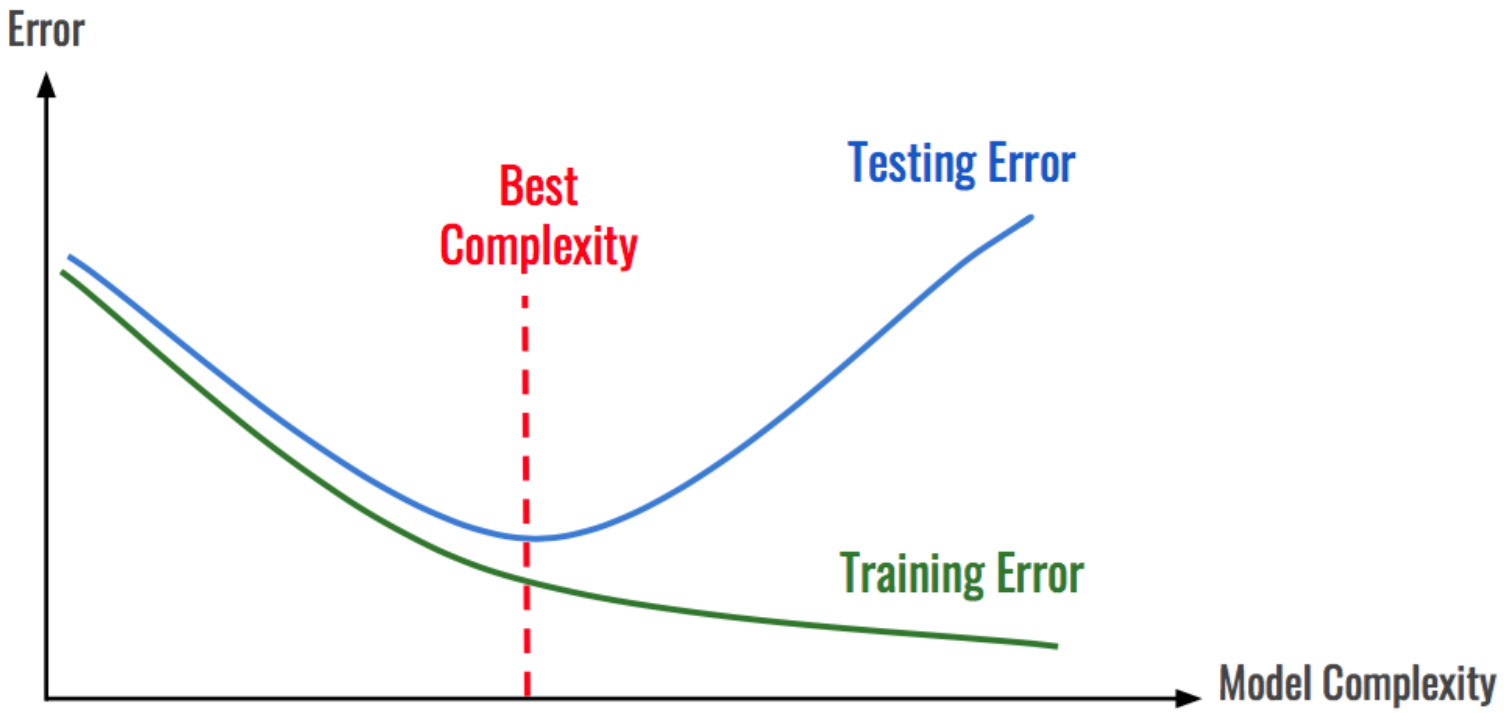
Sobreentrenamiento



Sobreentrenamiento



Sobreentrenamiento



Sobreentrenamiento

- Se debe evitar el sobreentrenamiento
  - Parar el crecimiento del árbol.
  - Post-procesamiento del árbol (poda)

Cómo?

  - Usar un conjunto de ejemplos de validación
  - Usar estadísticas

Bibliografía

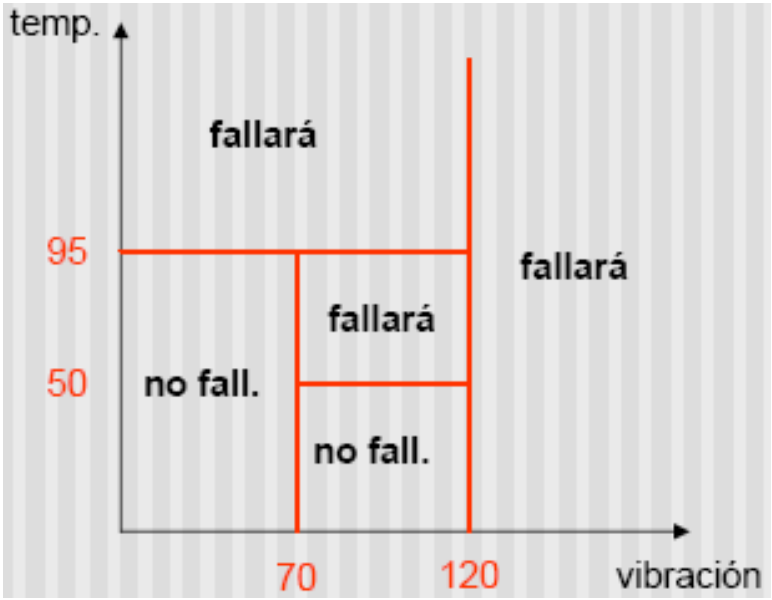
- Machine Learning - Tom Mitchell – McGrawHill
- Apuntes ML – Pablo Granitto

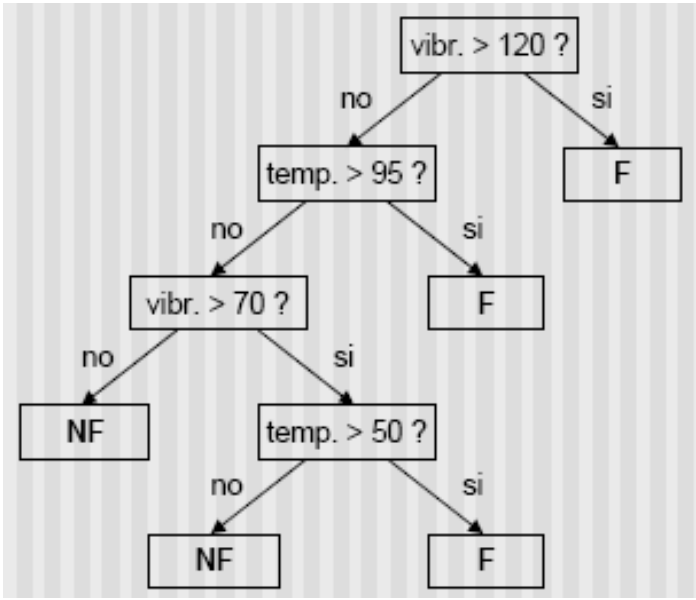
https://sites.google.com/site/aprendizajeautomatizadounr/Inicio/apuntes

- Curso de doctorado "Aprendizaje Automatizado y Data Mining" Universidad Miguel Hernández http://isa.umh.es/asignaturas/aprendizaje/index.html

Árboles de Decisión - Resumen (I)

- Capacidad de representación:
  - No muy elevada, las superficies de decisión son siempre perpendiculares a los ejes:





Árboles de Decisión - Resumen (II)

- Legibilidad: muy alta. Uno de los mejores modelos en este sentido.
- Tiempo de cómputo on-line: muy rápido. Clasificar un nuevo ejemplo es recorrer el árbol hasta alcanzar un nodo hoja.
- Tiempo de cómputo off-line: rápido. Los algoritmos son simples.
- Robustez ante instancias de entrenamiento ruidosas: robusto.
- Sobreentrenamiento o sobreajuste: Se controla a través de una poda.