Electiva Bases de datos

Árboles de decisión

Jimmy Mateo Guerrero[1]

[1]Departamento de Sistemas
Facultad de Ingeniería

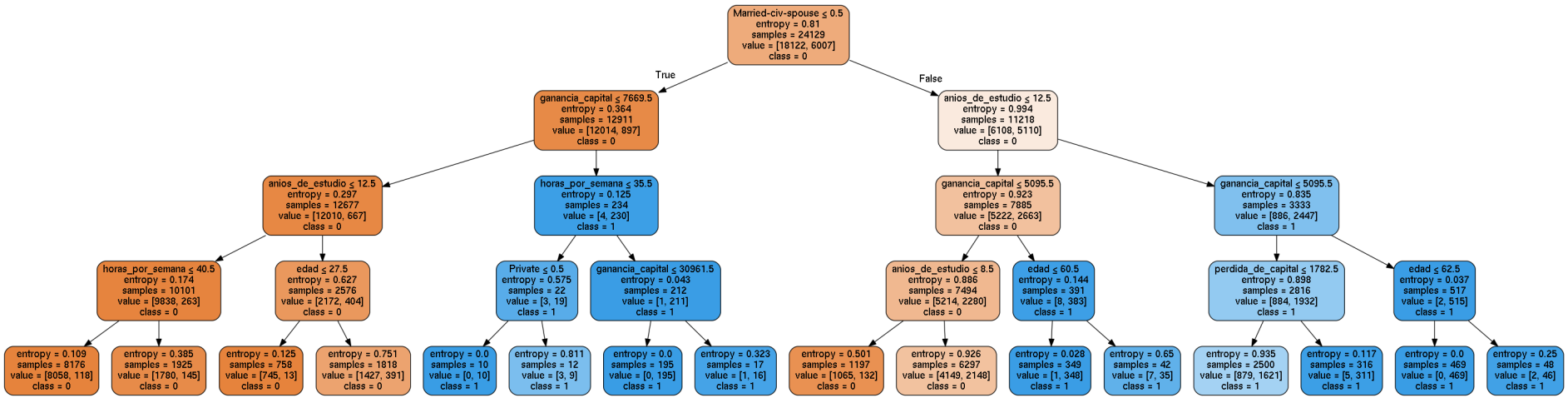
Árboles de decisión, 2019

Contenido

Árboles de decisión

- Probablemente el método más conocido para resolver problemas de clasificación.
- Uno de los primeros desarrollos en ML.
- Muy asociado a nuestra forma de proceder

Árboles de decisión



Árboles de decisión

- ID3 y sus sucesores han sido desarrollados por Ross Quinlan en la década de 1970.
- Posteriormente trabajó en Sydney Uni, Rand Corporation en California.
- Ahora dirige su propia compañía Rulequest.

Árboles de decisión

Características:

- Capaz de aprender conceptos disyuntivos.
- Tolerante al ruido en atributos y clase.
- Tolerante a valores faltantes en atributos.
- Estructura de datos recursiva definida por:
  - Nodos hoja que indican un rótulo de clase.
  - Nodos internos que contienen un test sobre el valor de un atributo. Para cada resultado del test hay una rama y un subárbol con la misma estructura que el árbol.
- Representación alternativa:
  conjunto de reglas: IF Condición THEN Clasificación

Árboles de decisión

El problema de encontrar un árbol de decisión que concuerde con el conjunto de entrenamiento tiene una solución trivial.

- Simplemente construimos un árbol de decisión que tenga una rama por cada ejemplo.
- El problema con este árbol trivial es que solo memoriza las observaciones.
- No extrae ningún patrón a partir de los ejemplos y no puede extrapolar a ejemplos que no vio.

Árboles de decisión

- Extraer un patrón significa poder describir un gran número de casos de un modo conciso. El lugar de sólo tratar de encontrar uno que concuerde con los ejemplos, deberíamos también tratar de encontrar uno conciso.
- Este es un ejemplo de un principio general del aprendizaje inductivo:
  - La hipótesis más probable es la más simple que sea consistente con todas las observaciones.

Árboles de decisión

- Navaja de Occam (Occam's Razor).
- Basado en una premisa muy simple:
  - En igualdad de condiciones la solución más sencilla es probablemente la correcta.
  - No ha de presumirse de la existencia de más cosas que las absolutamente necesarias.
  - No hay que desarrollar sistemas más complejos que lo necesario.

Árboles de decisión

- Desafortunadamente, encontrar el árbol de decisión óptimo (más chico) es un problema intratable, pero algunas heurísticas simples logran encontrar uno suficientemente pequeño.
- La idea básica del algoritmo es usar primero el atributo “más importante”, donde esto significa el que tiene un efecto más importante en la agrupación de los ejemplos en conjuntos que comparten la misma clasificación.

Árboles de decisión

¿Heurística?

- Como metodología científica, la heurística es aplicable a cualquier ciencia e incluye la utilización de: medios auxiliares, principios, reglas, estrategias o algoritmos que faciliten la búsqueda de vías de solución a problemas.

Induciendo árboles

Id3(Ejemplos, Clase, Atributos)

- Si todos los ejemplos son positivos retornar un nodo positivo.
- Si todos los ejemplos son negativos retornar un nodo negativo.
- Si Atributos está vacío retornar nodo de la clase más frecuente .
- En otro caso.
  - A = ELEGIR_MEJOR_ATRIBUTO (Atributos, Ejemplos)
  - Raíz = A
  - Para cada valor posible v de A
    - Agregar una nueva rama debajo de Raíz con test A = v?
    - Ejemplos_v = elementos de Ejemplos con A = v
    - Si Ejemplos_v está vacío devolver un nodo de la clase más frecuente en ejemplos.
    - Si no agregar debajo de esta rama el subárbol
      Id3 (Ejemplos_v, Clase, Atributos – A)

Outline

Criterio de selección

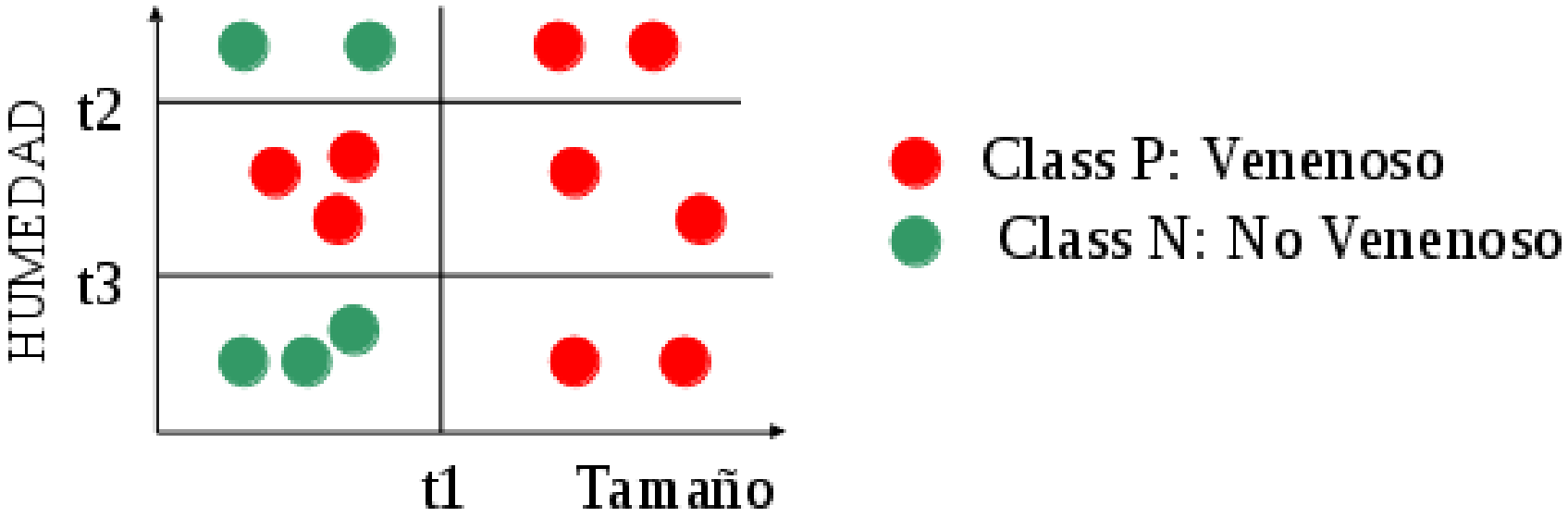
- Asumamos que el Criterio de Partición se limita a seleccionar un atributo.
- ¿Cómo elegir el atributo más "importante" para dividir?
- Muchas aproximaciones:
  - Information Gain.
  - Information ratio.
  - Gini.
  - . . . tantas como autores.

Criterio de selección

Existen varios aspectos prácticos a resolver a la hora de implementar árboles de decisión.

- Cuál es el objetivo del árbol?
- Cuál es un buen criterio de partición?
- Qué tan profundo debe ser el árbol?
- Qué pasa cuando hay valores faltantes?
- Cómo podemos afrontar la eficiencia computacional para grandes evidencias?

Criterio de selección



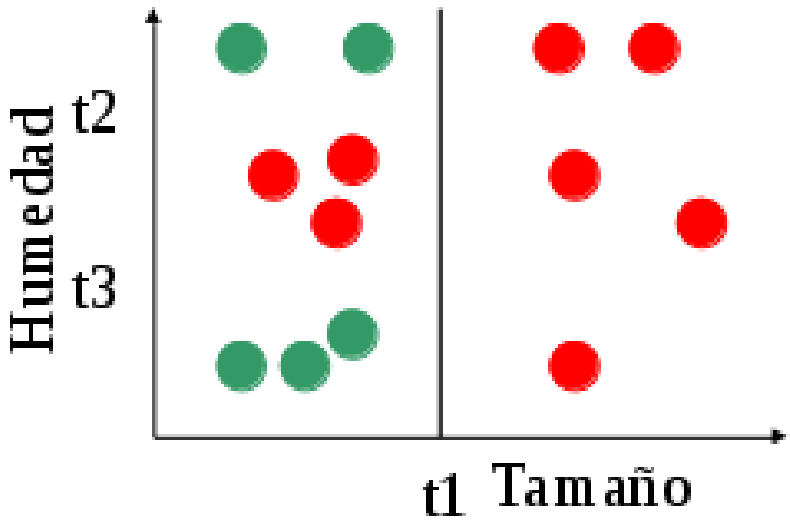
Atributos: tamaño y humedad

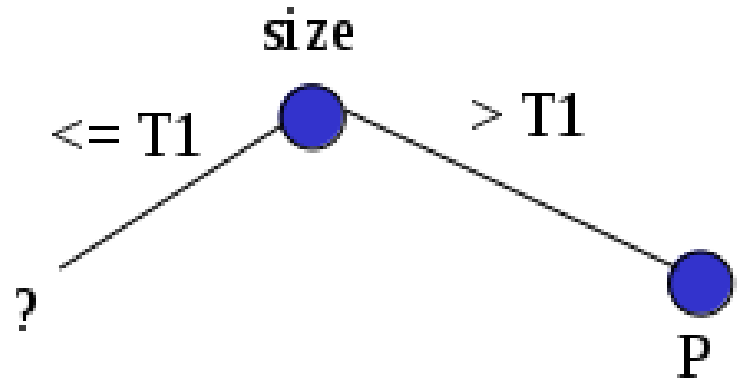
Tamaño tiene 2 valores: > t1 o <= t1

Huemdad tiene 3 valores: >t2, (>t3 y <=t2), <= t3

Criterio de selección

Supongamos que elegimos el tamaño como el mejor atributo:



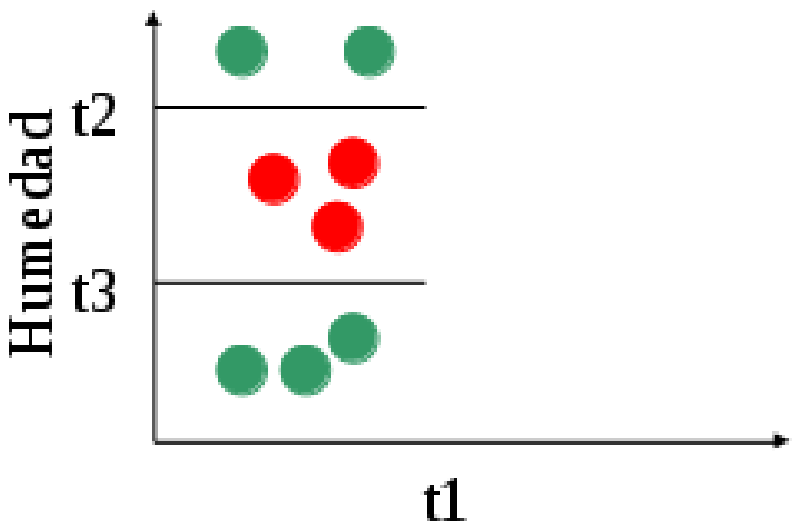


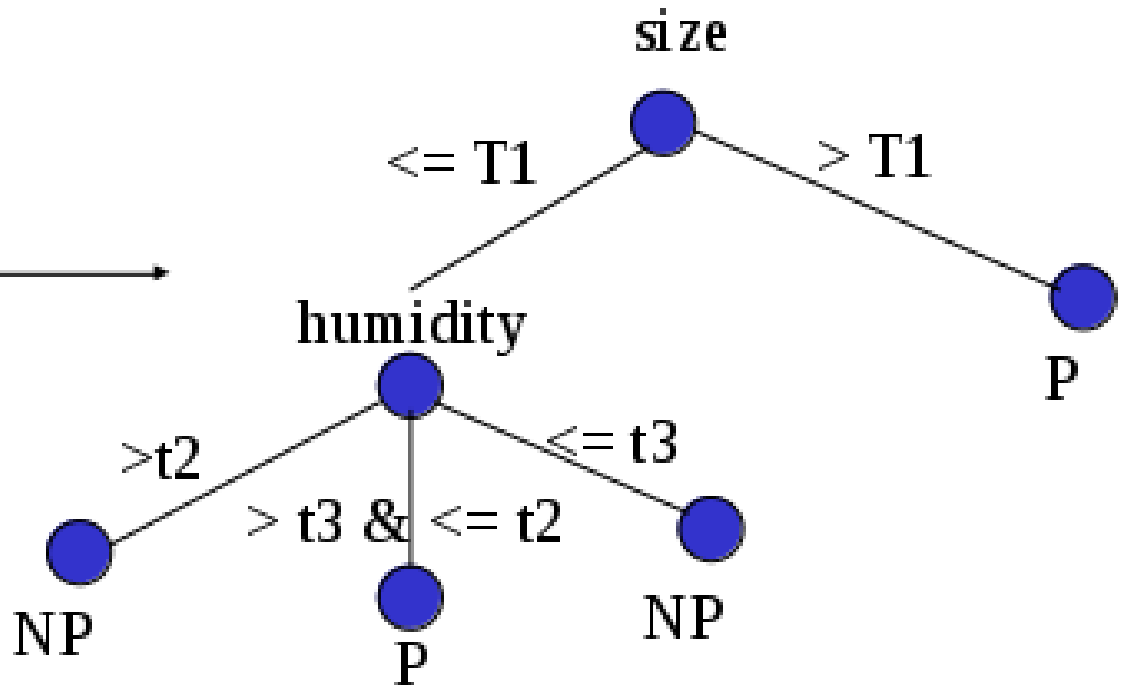
Class P: Venenoso

Class N: No Venenoso

Criterio de selección

Supongamos que elegimos la humedad como el siguiente mejor atributo:



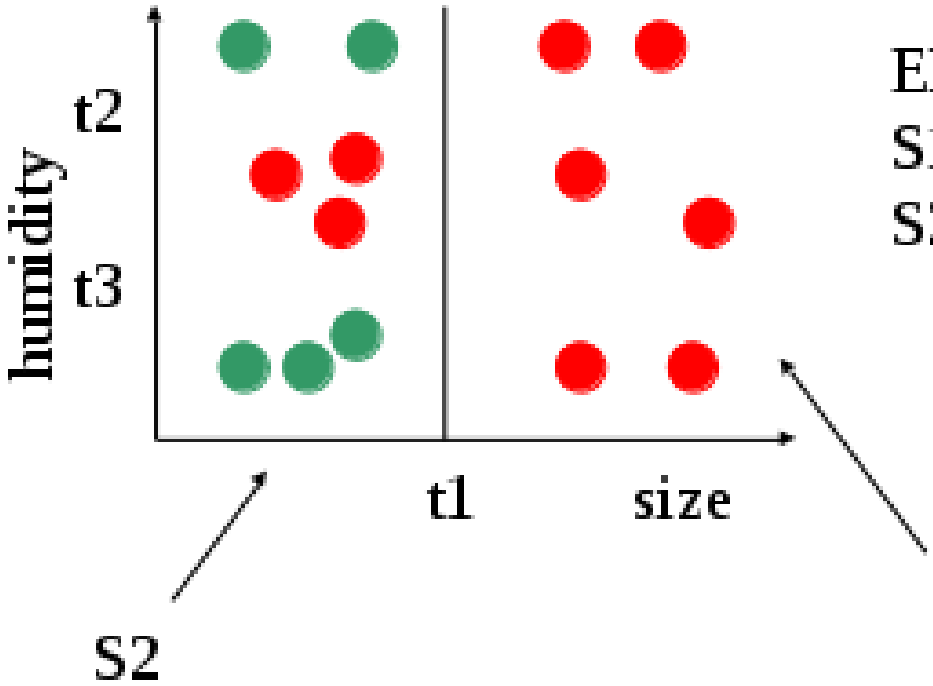


Criterio de selección

Para decidir cual es el mejor atributo para abrir el árbol:

- Usamos algunas definiciones de Teoría de la Información.
- La medida de incertidumbre o entropía asociada a una variable aleatorea X se define como:
  - $$H(X) = -\sum_{i=1} p_i log2(p_i)$$
  - Siendo log2 un redefinición de logaritmo en base 2 con log(0)=0.

Criterio de selección



El tamaño divide la muestra en dos.

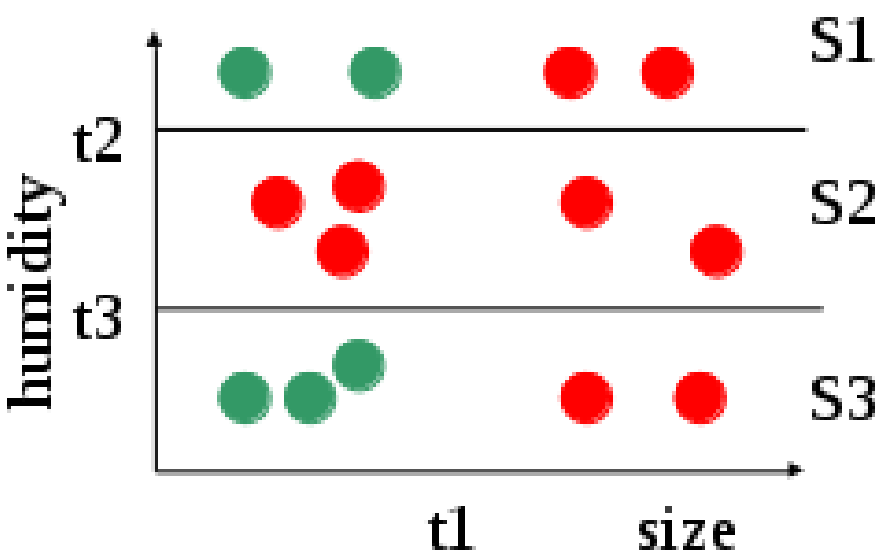
S1 = { 6P, 0NP} (size>t1)

S2 = { 3P, 5NP} (size<t1)

$H(S1) = -(6/6)\log2(6/6) + (0/6)\log2(0/6) = 0$

$H(S2) = -(3/8)\log2(3/8) + -(5/8)\log2(5/8) = 0.95$

Criterio de selección



La humedad divide la muestra en tres.

S1 = {2P, 2NP} humidity>t2

S2 = {5P, 0NP} t3<humidity<t2

S3 = {2P, 3NP} humidity<t2

$$H(S1) = -(2/4)\log2(2/4) + -(2/4)\log2(2/4) = 1$$

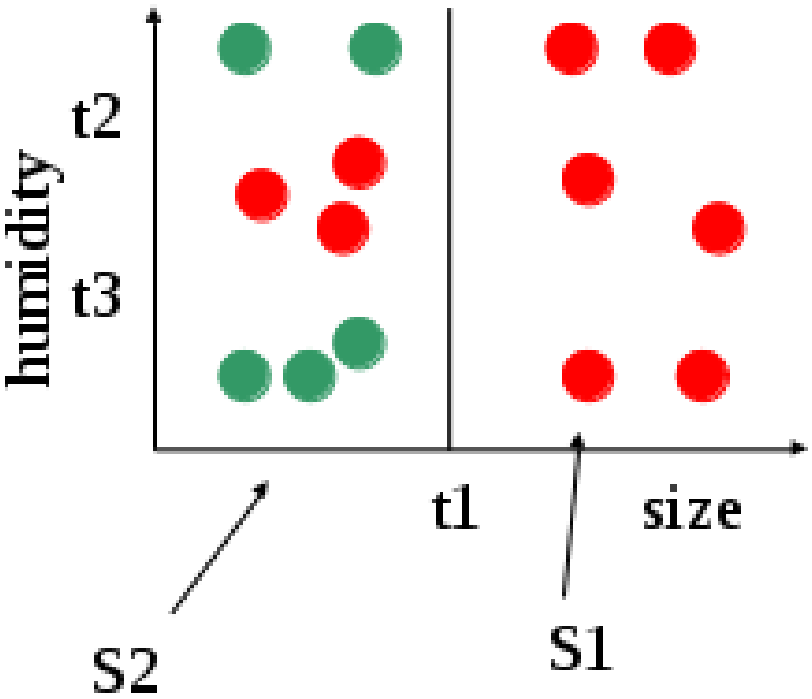
$$H(S2) = 0$$

$$H(S3) = -(2/5)\log2(2/5) + -(3/5)\log2(3/5) = 0.97$$

Criterio de selección

- Definimos la función de Ganancia de Información como:
  - $$IG(A) = H(S) - \sum_{v}(Sv/S)H(Sv)$$
  - Donde:
    - H(S) es la entropía del conjunto de todos los ejemplos
    - H(Sv) es la entropía del subconjunto de los ejemplos con valor v para el atributo A
  - Elegiremos al atributo que maximice la IG, por lo tanto debemos calcular IG para todos los atributos:

Criterio de selección



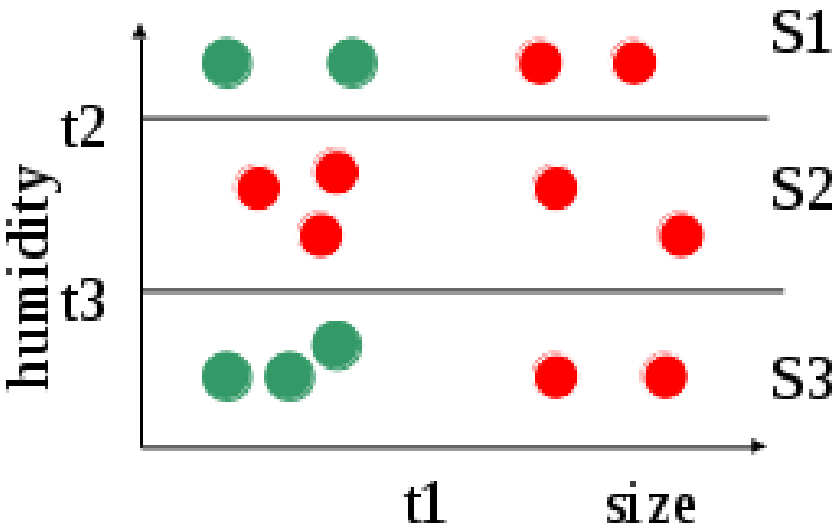
$$H(S1) = 0$$
$$H(S2) = 0.95$$

$$H(S) = -(9/14)\log2(9/14) + -(5/14)\log2(5/14) = 0.94$$

$$|S1|/|S| = 6/14$$
$$|S2|/|S| = 8/14$$

$$IG(size) = H(S) - (|S1|/|S|*H(S1) + |S2|/|S|*H(S2))$$
$$= 0.94 - (6/14 * 0 + 8/14 * 0.95) = 0.40$$

Criterio de selección



$H(S1) = 1$
$H(S2) = 0$
$H(S3) = 0.97$

$H(S) = 0.94$

$|S1|/|S| = 4/14$
$|S2|/|S| = 5/14$
$|S3|/|S| = 5/14$

$$IG(humidity) = H(S) - (|S1|/|S| * H(S1) + |S2|/|S| * H(S2) + |S3|/|S| * H(S3))$$
$$= 0.94 - (4/14 * 1 + 5/14 * 0 + 5/14 * 0.97)$$
$$= 0.31$$

Outline

Criterio de parada

Hasta cuando dividimos?

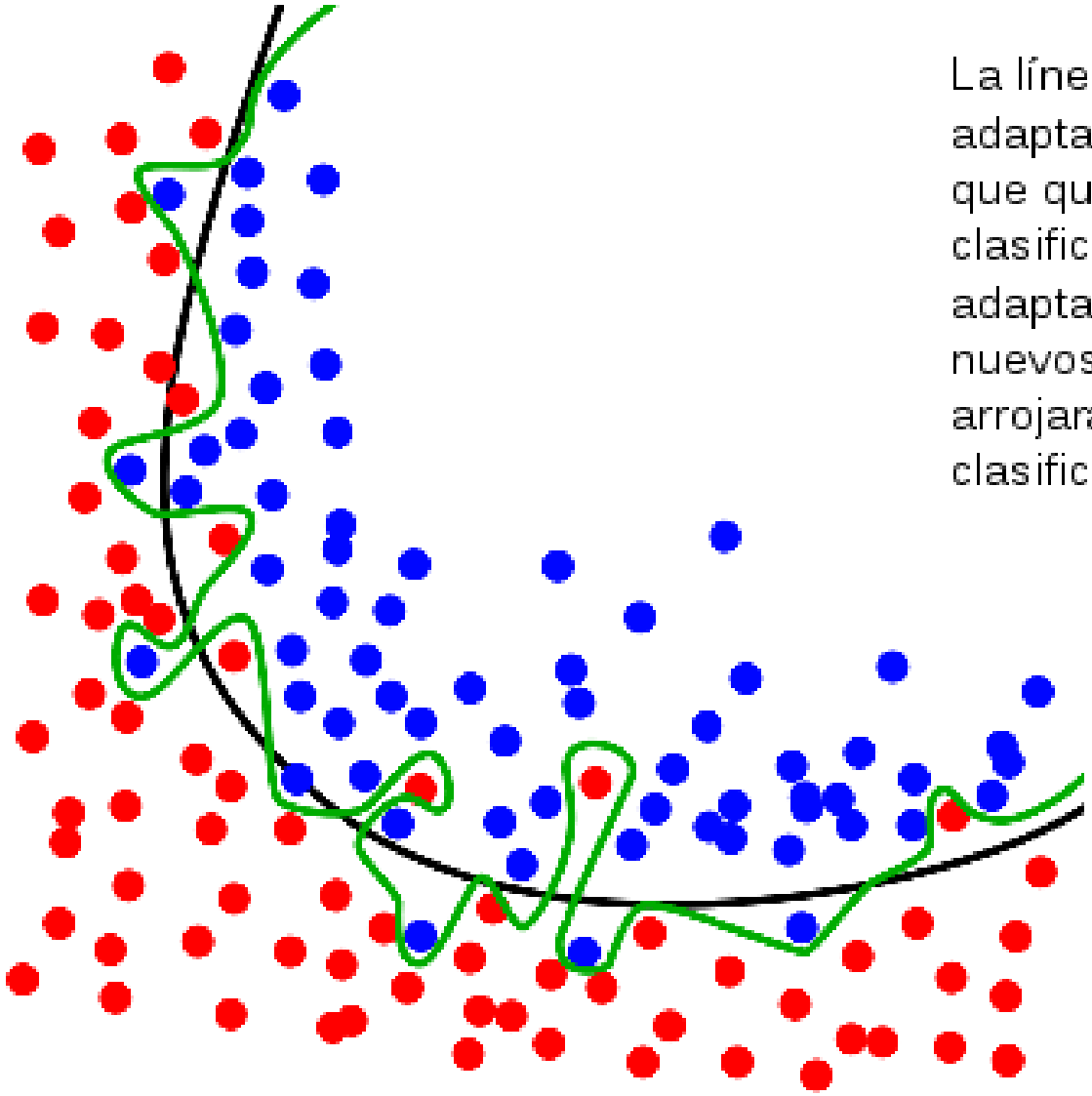
- Hasta que tenga clases puras en todos los nodos
- Hasta que no tenga más variables disponibles
- O hay algo mejor?

Criterio de parada

- La versión canónica dice que el árbol se sigue desarrollando hasta que se llega a una clase uniforme. Esto en la práctica no suele ser efectivo.
- Existen muchas opciones de criterio de parada:
  - No hay más atributos.
  - Clase uniforme.
  - Se supera un nivel de profundidad máximo.
  - El número de casos es menor a un mínimo.
  - El mejor valor del criterio de partición es menor de un umbral.
  - El número de casos de los nodos hijos es menor que un mínimo.

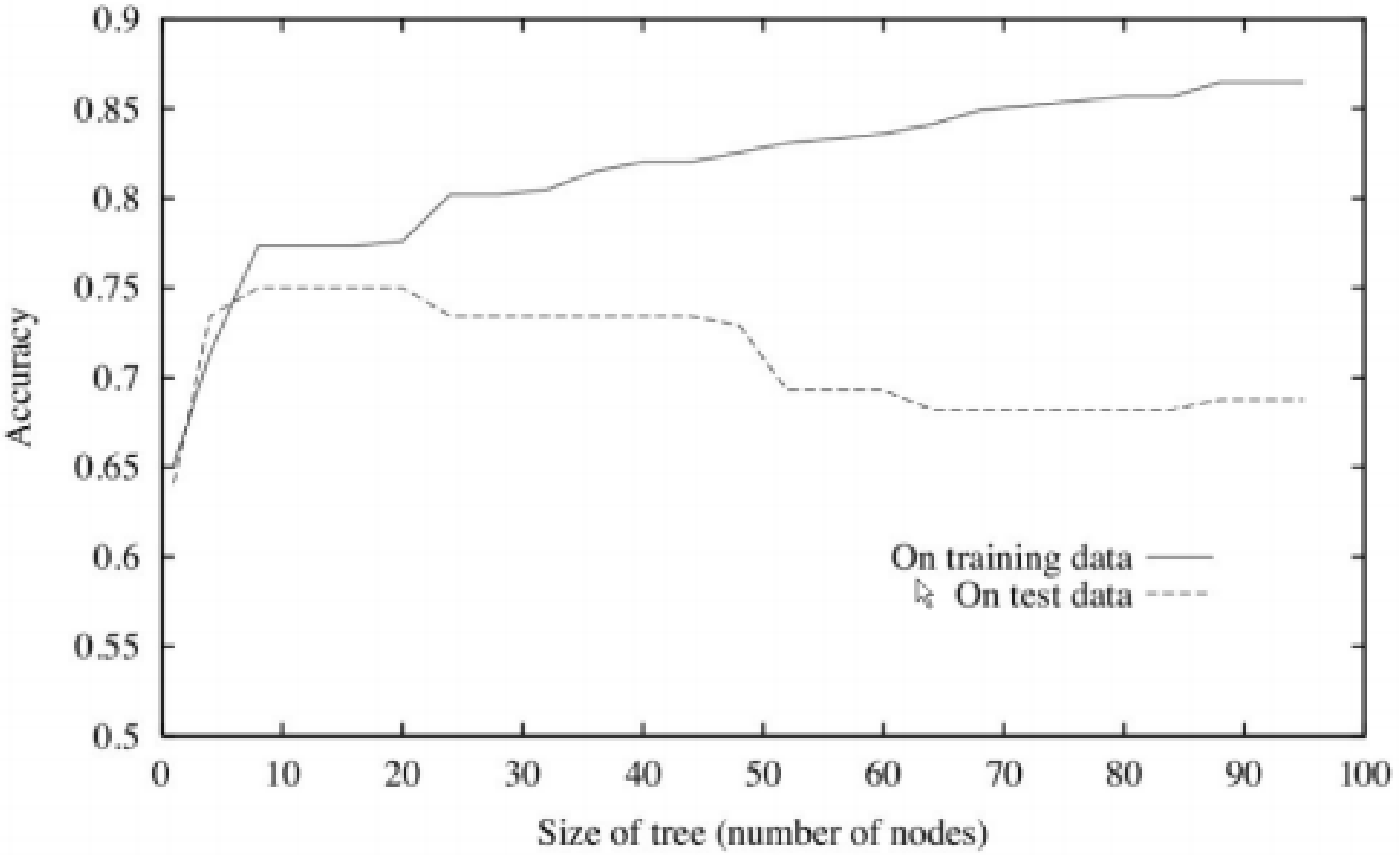
Outline

Sobreajuste

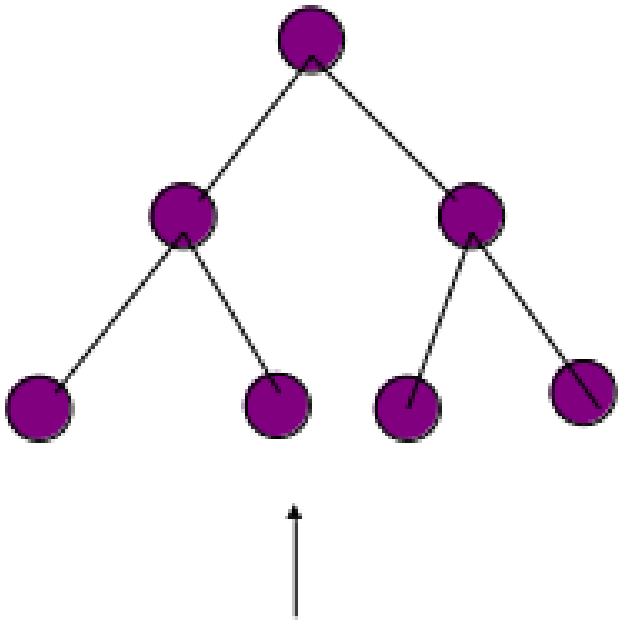


La línea verde como clasificador se adapta mejor a los datos con los que que se ha entrenado al clasificador, pero está demasiado adaptada a ellos, de forma que ante nuevos datos probablemente arrojará más errores que la clasificación usando la línea negra.

Sobreajuste

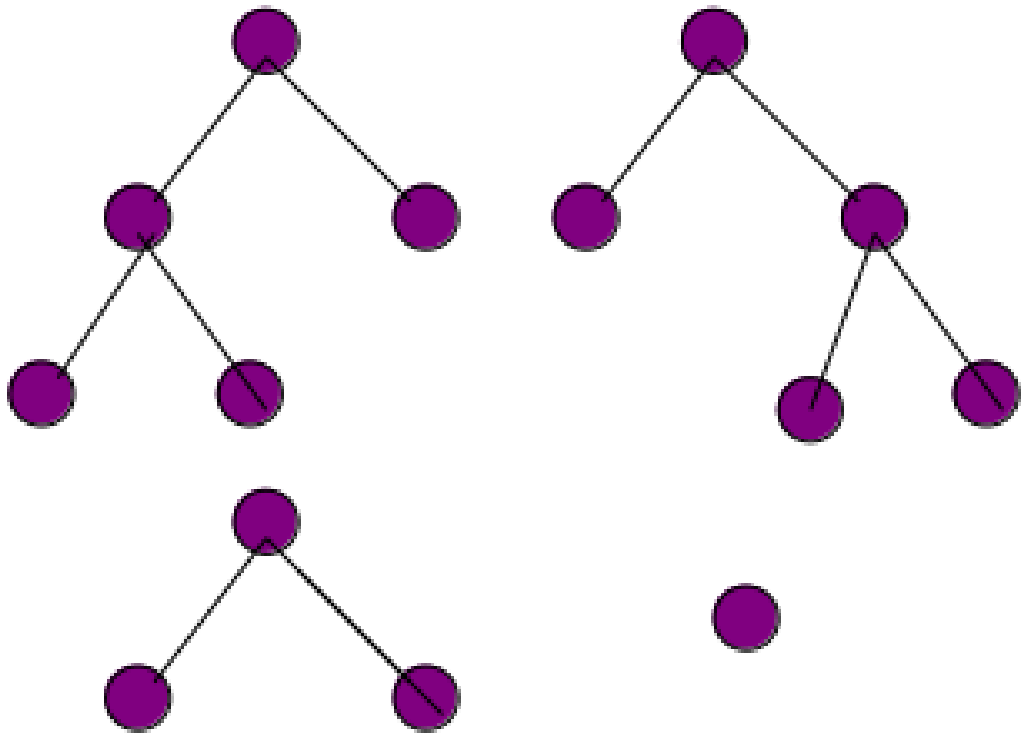


Sobreajuste



Árbol original

Posibles árboles después de la poda:



Bibliografía I

Aurélien Géron

Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.

Publisher: O'Reilly Media, Year: 2017

Jake VanderPlas.

Python Data Science Handbook: Essential Tools for Working with Data

O'Reilly Media, Year: 2016

Brett Lantz.

Machine learning with R

Packt Publishing, Year: 2013