Curso sobre Modelamiento Estadístico y Sistemas Recomendadores

Clase Presencial 3: Técnicas de Clasificación

(Aprendizaje Supervisado)

3.1.- Introducción

Introducción

- Las técnicas de clasificación son formas de análisis de datos que pretenden generar modelos que describen un atributo discreto de interés.

- Estos modelos reciben el nombre de *clasificadores* y "predicen" clases discretas no ordenables.

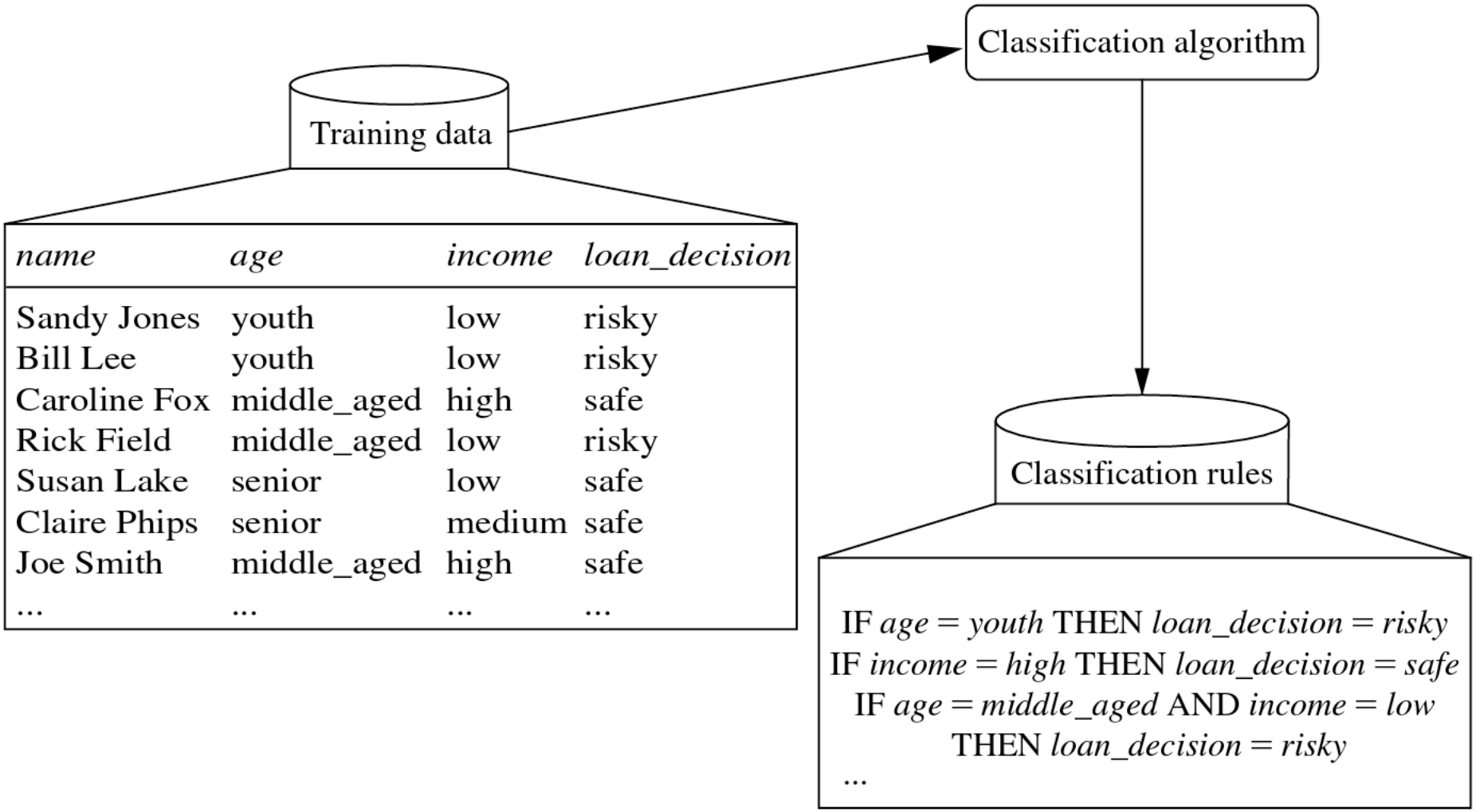
- Ejemplos de aplicaciones:
  - categorización de textos (ej., spam).
  - detección de fraudes (ej. firmas).
  - visión de máquinas (ej., detección de rostros).
  - segmentación de mercado (ej., tipos de clientes que responden a una promoción).
  - bioinformática (ej., proteinas de acuerdo a su función).

Introducción

El proceso de clasificación consta de dos pasos:

1) Paso de aprendizaje: se construye el modelo basado en datos recopilados previamente (trainning data).

Introducción



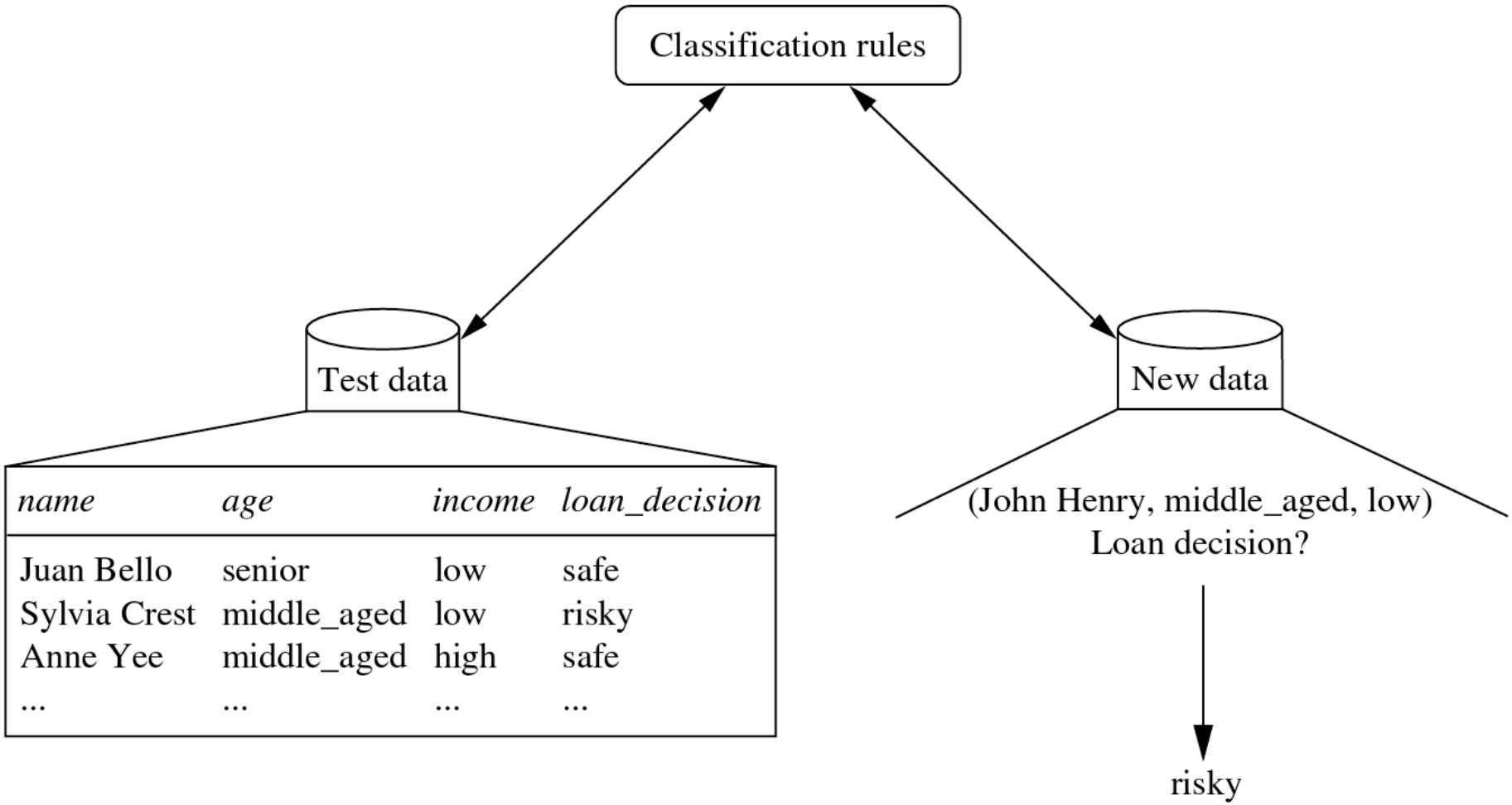
name	*age*	*income*	*loan_decision*
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

Introducción

El proceso de clasificación consta de dos pasos:

2) Paso de validación y clasificación: se determina la precisión del clasificador, y si es aceptable, se usa para predecir las clases de un conjunto de datos.

Introducción



name	*age*	*income*	*loan_decision*
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

3.2.- Árboles de Decisión

Definición

Definición

Un árbol de decisión es un diagrama de flujo, donde:

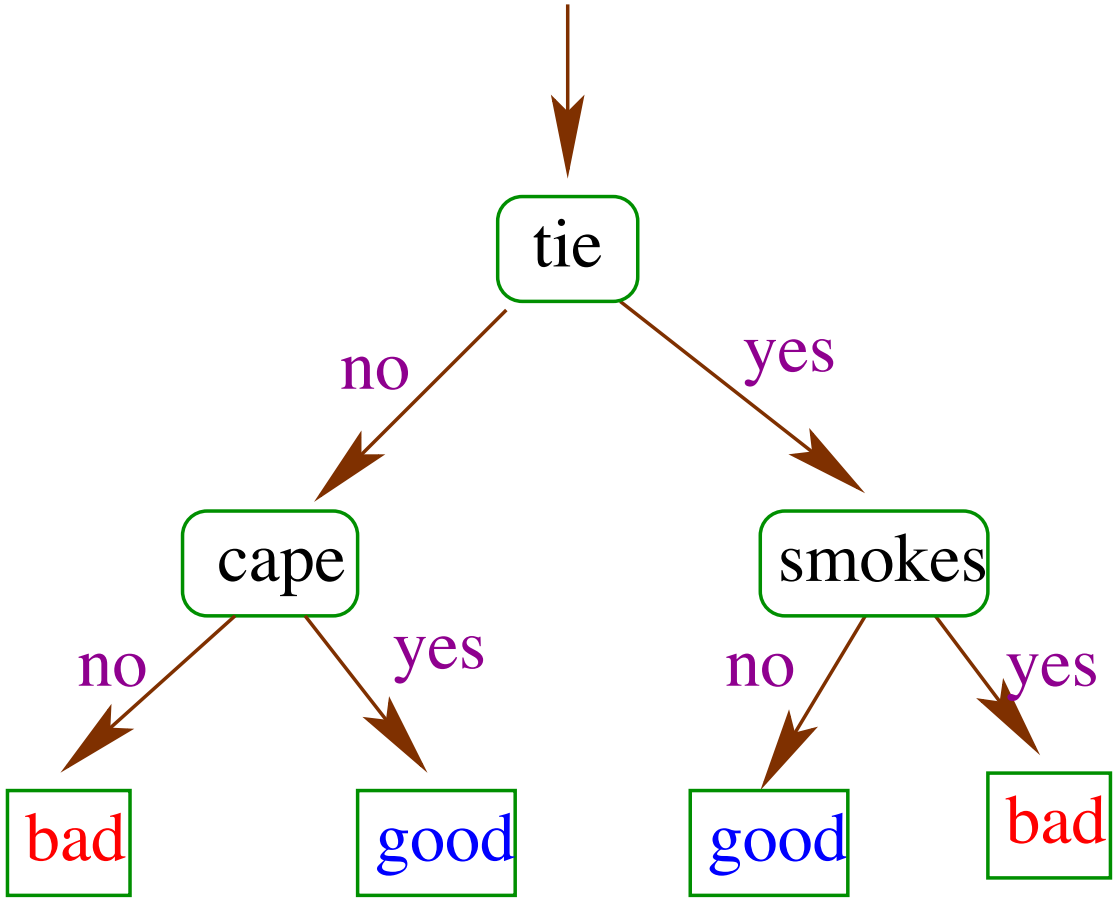
- *cada nodo interno representa una "pregunta" sobre un atributo,*
- *cada rama representa una respuesta a esa pregunta y*
- *cada hoja o nodo terminal representa la clase de clasificación.*

Ejemplo

Identificación de personas de acuerdo a su apariencia.

	sex	mask	cape	tie	ears	smokes	class
			training data				
batman	male	yes	yes	no	yes	no	Good
robin	male	yes	yes	no	no	no	Good
alfred	male	no	no	yes	no	no	Good
penguin	male	no	no	yes	no	yes	Bad
catwoman	female	yes	no	no	yes	no	Bad
joker	male	no	no	no	no	no	Bad
			test data				
batgirl	female	yes	yes	no	yes	no	??
riddler	male	yes	no	no	no	no	??

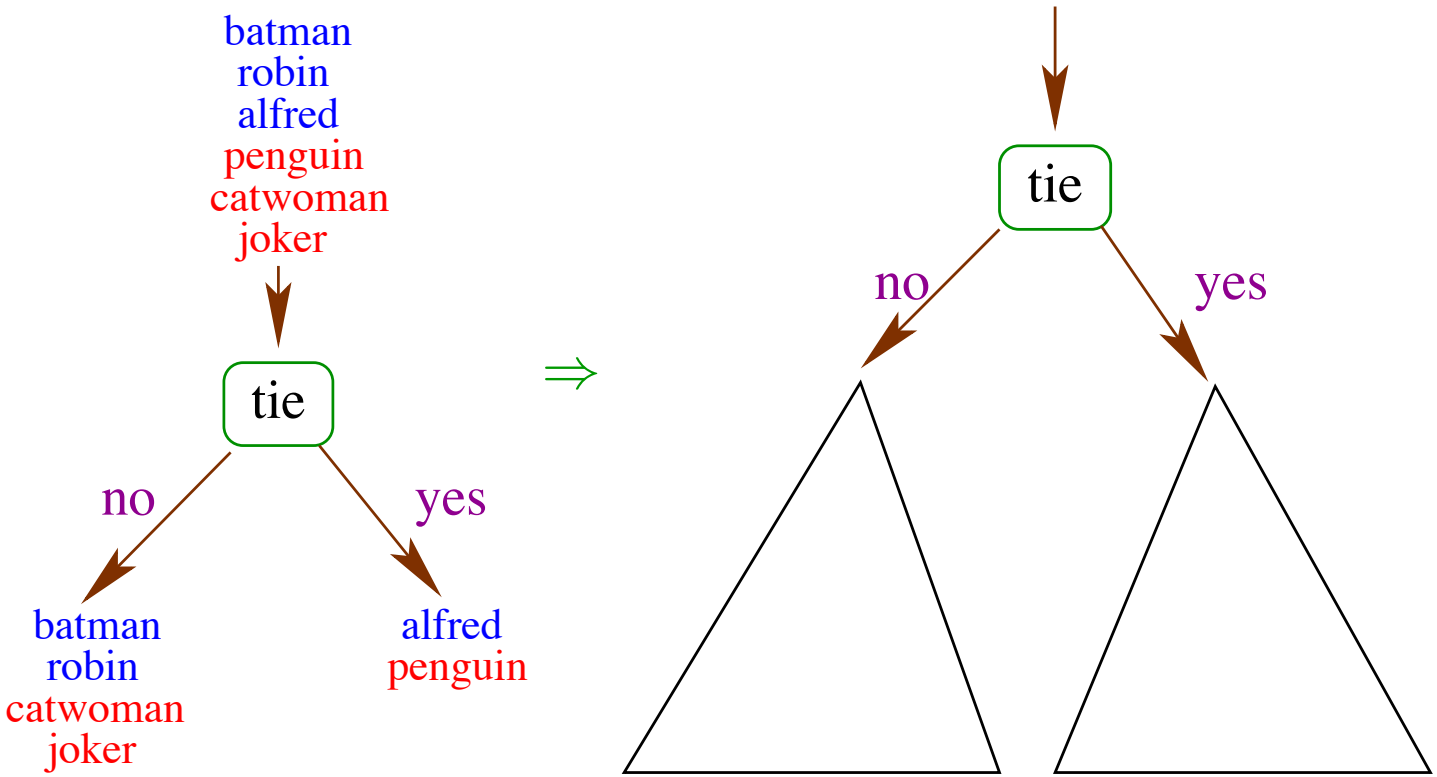
Ejemplo



Algoritmo Básico

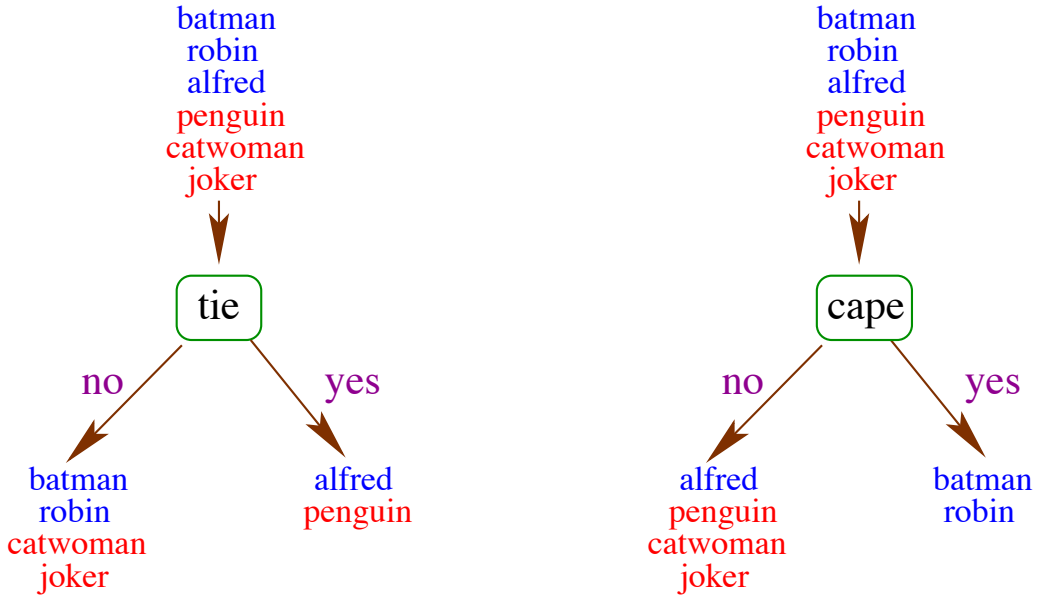
- ¿Cómo se construyen los árboles de decisión?
  - Elección de la regla en base a la cuál particionar los datos.
  - Dividir los datos en subconjuntos disjuntos utilizando la regla de partición.
  - Repetir en forma recursiva para cada subconjunto.
  - Parar cuando las hojas son (casi) “puras”.

Algoritmo Básico



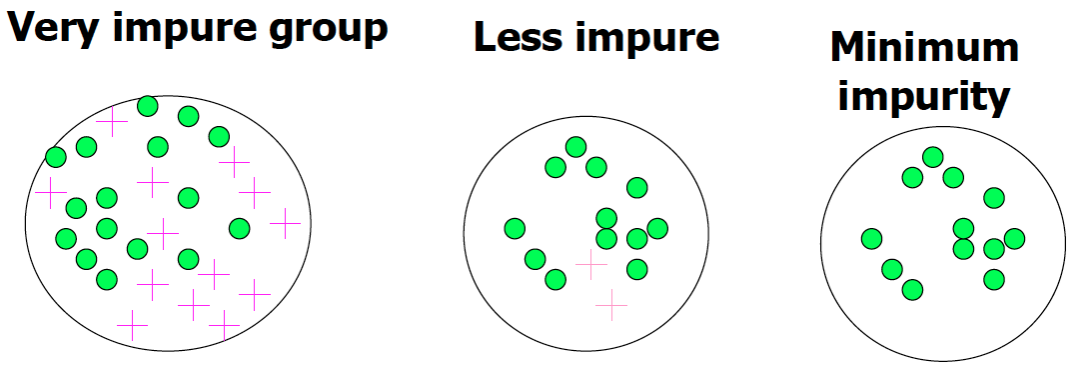
Algoritmo Básico

- ¿Cómo se escoge la mejor regla de partición?



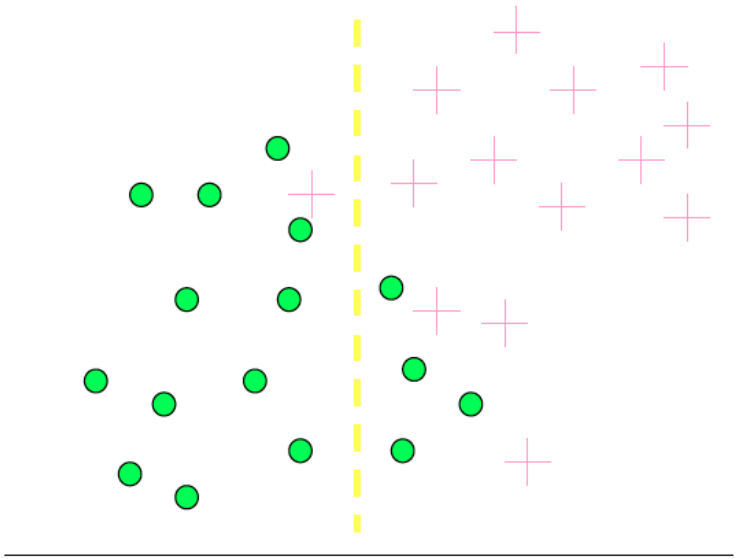
Algoritmo Básico

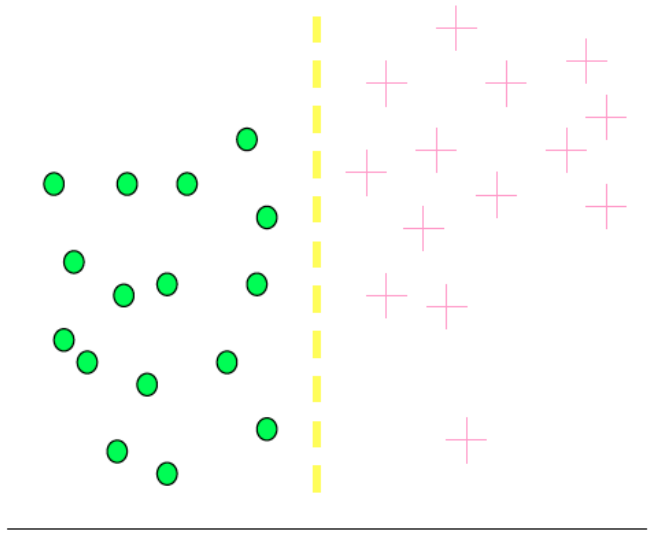
- Una opción es escoger la regla que permita el mayor aumento en la “pureza” de los grupos.
- Entonces necesitamos una medida que cuantifique el nivel de “impureza” en un grupo:



Algoritmo Básico

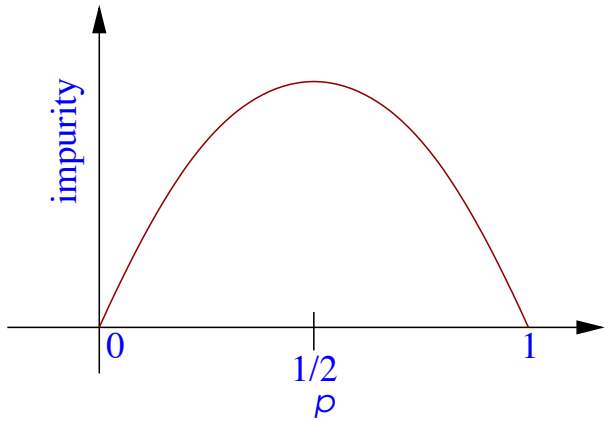
¿Qué atributo es más informativo?





Algoritmo Básico

- Nos gustaría que la función que cuantifique el nivel de impureza se vea así (p = fracción de casos positivos):



- Medidas de impureza utilizadas usualmente:
  - Entropia: $-p\log(p) - (1-p)\log(1-p)$.
  - Índice de Gini: $p(1-p)$.

Entropía

La información esperada necesaria para clasificar una unidad experimental 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 unidad arbitraria de D pertenezca a la clase C_i y es estimada por:

$$\frac{|C_{i,D}|}{|D|}$$

$Info(D)$ también se conoce como entropía.

Entropía

Para saber cuánta más información necesitaríamos, luego de la partición, para llegar a una clasificación exacta, disponemos de:

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

$Info_A(D)$ es la información esperada requerida para clasificar una unidad desde D basado en la partición sugerida por A. A menor valor de información esperada requerido, mayor es la pureza de las particiones.

Entropía

Finalmente, la ganancia de información se define como la diferencia entre el requerimiento de información original (basado en la proporción de clases) y el nuevo requerimiento (obtenido luego de particionar en A). Esto es:

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

Árboles de Decisión: Entropía

- *Gain*(*A*) dice cuánto se ganaría al ramificar por *A*.
- Es la reducción en información esperada causada por conocer el valor de *A*.
- El atributo *A* con mayor ganancia de información, es elegido como el atributo de división del nodo *N*.

Atributos continuos

Debemos determinar el mejor punto de corte para A en un umbral.

1. Se deben ordenar crecientemente los valores de A.

2. Típicamente, el punto medio entre cada par de valores adyacentes es un posible punto de corte. Por lo tanto, dados v valores de A, debemos evaluar $v - 1$ cortes.

3. Para cada posibles punto de corte para A, se evalúa $Info_A(D)$, donde el número de elementos de la partición es 2.

4. El punto con el requerimiento mínimo de información esperada para A se selecciona como el punto de corte para A.

5. D_1 es el conjunto de tuplas que satisfacen $A \leq$ *punto corte*, y D_2 es el conjunto de tuplas que satisfacen $A >$ *punto corte*.

Árboles de Decisión: Razón de ganancia

- La ganancia de información favorece atributos con muchas categorías.
- Una alternativa es usar una extensión de la ganancia de información, conocida como razón de ganancia.

Árboles de Decisión: Razón de ganancia

- "Normalización" a la ganancia de información, usando la información de división.

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

- Potencial información generada al separar el conjunto de datos D, en una partición con v grupos, correspondientes a las v posibilidades del atributo A.

- Para cada valor o categoría de A, considera el número de tuplas que tienen esa categoría con respecto al total de tuplas en D.

Razón de ganancia

La razón de ganancia está definida por:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

Se selecciona aquel atributo con mayor razón de ganancia.

Índice Gini

Mide la impureza de D mediante:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

donde p_i es la probabilidad de que una tupla en D pertenezca a la clase C_i ($i = 1, \ldots, m$), la que es estimada por:

$$\frac{|C_{i,D}|}{|D|}.$$

Índice Gini

- Considera una división binaria para cada atributo.
- Sea A variable nominal con valores $\{a_1, a_2, \ldots, a_v\}$
- Examina todos los posibles subconjuntos, S_A, que pueden formarse usando los valores de A.
- ¿$A \in S_A$?
- Hay 2^v posibles subconjuntos.
- Ejemplo: `ingreso` con 3 posibilidades (bajo, medio, alto).

Índice Gini

- El conjunto completo y el vacío no son consideradas divisiones.
- $2^v - 2$ posibles formas de particionar los datos en 2, basadas en una división binaria de A.
- Se calcula una suma ponderada de la impureza de cada partición resultante.
- Si D se particiona en dos grupos de acuerdo al atributo A, resultando D_1 y D_2, el índice Gini de D dada esa partición es:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

Índice Gini

- Se considera cada una de las posibles divisiones binarias para cada atributo.
- Para una variable nominal, el subconjunto que entrega el menor índice Gini es seleccionado.
- Para variables continuas, se debe considerar cada punto de división posible. La estrategia es similar a la descrita en el caso de la ganancia de información, considerando los puntos medios de todos los valores adyacentes ordenados de manera creciente.

Índice Gini

La reducción en impureza obtenida por una división de una variable nominal o continua A es:

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

Se selecciona el atributo que maximice la reducción de impureza (menor índice Gini).

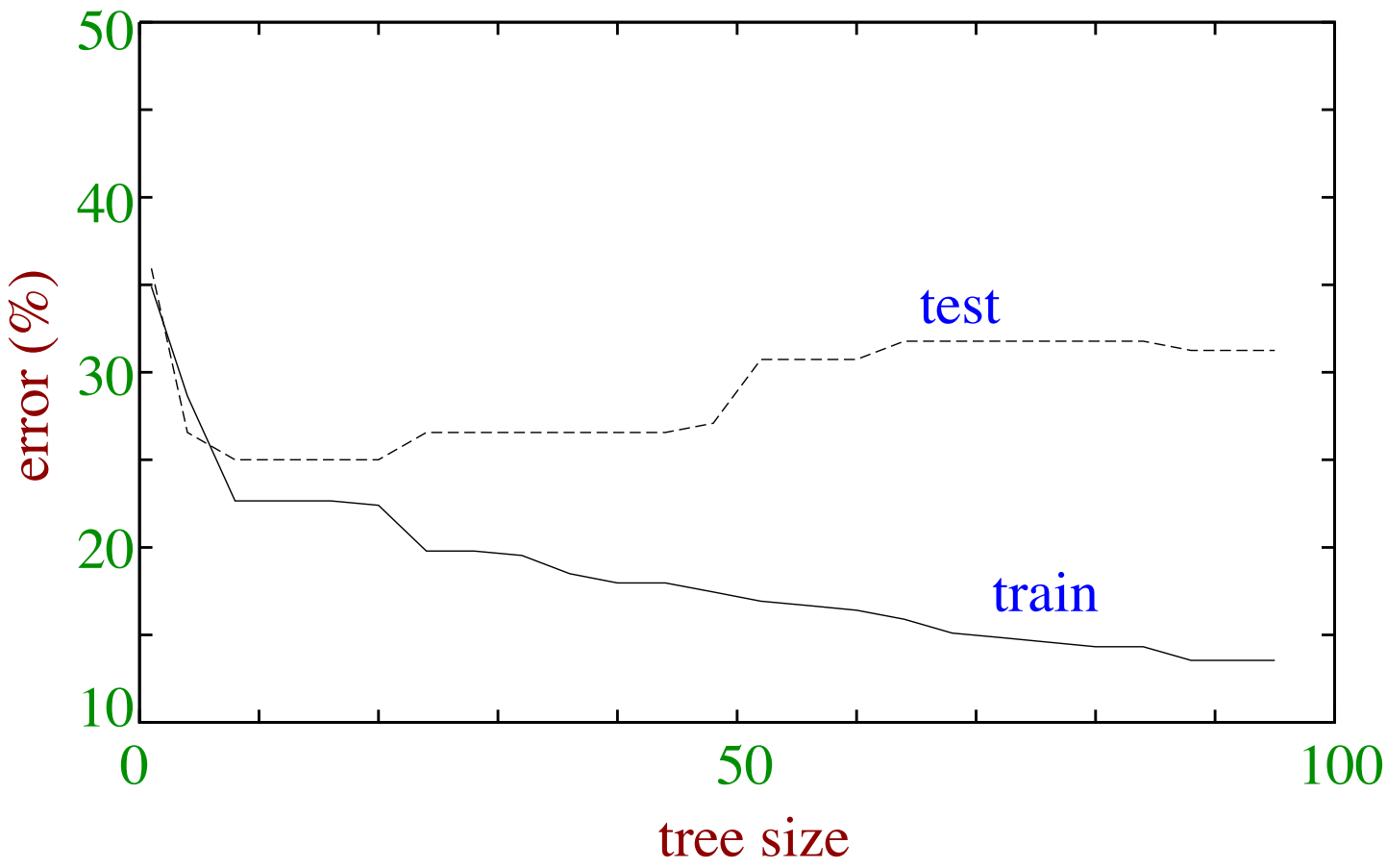
El criterio de división está dado por el atributo elegido y el subconjunto o punto de separación.

Tipos de Errores y Poda

- **Error de entrenamiento:** proporción de los datos de entrenamiento incorrectamente clasificados.
- **Error de prueba:** proporción de los datos de prueba clasificados de forma incorrecta.

Tipos de Errores y Poda

- Tamaño del árbol vs tipo de error:



Tipos de Errores y Poda

- Los árboles deben se suficientemente grandes para ajustar los datos de entrenamiento.
- Sin embargo, árboles muy “complejos” pueden “sobreajustar” los datos de entrenamiento (capturando ruidos en los datos o patrones espurios) y clasificar mal registros futuros.

Tipos de Errores y Poda

- Los métodos de poda, pretenden solucionar el problema de sobreajuste.
- Un árbol podado tiende a ser más pequeño y menos complejo, por lo que es más comprensible.
- Existen 2 métodos de poda: pre-poda (controlan crecimiento del árbol antes de ajuste perfectamente los datos) y post-poda (permite ajuste perfecto y luego se poda).
- En la práctica se usan más las técnicas de post-poda, porque es difícil estimar con precisión cuando se debe detener el crecimiento del árbol.

Costo de complejidad de un árbol

Usualmente utilizado como criterio de post-poda.

- Penaliza la calidad de la clasificación de un árbol por su complejidad.
- Corresponde a una función de la **tasa de mala clasificación** y el **número de hojas** del árbol.
- Una vez entrenado un árbol, se utiliza el costo de complejidad para evaluar cada uno de sus sub-árboles.

Costo de complejidad de un árbol

Costo de un árbol

Dado un árbol T cualquiera, y un **parámetro de complejidad** $\alpha > 0$ dado, el **costo** de dicho árbol, $R_\alpha(T)$, se define como:

$$R_\alpha(T) = R(T) + \alpha|T|,$$

donde $R(T)$ corresponde a la tasa de mala clasificación de T y $|T|$ a su número de hojas.

Costo de complejidad de un árbol

Árbol asociado a un parámetro α

Dado un árbol T, y un parámetro $\alpha > 0$ fijo, se define T_α como el sub-árbol de T que minimiza:

$$R_\alpha(T_i) = R(T_i) + \alpha|T_i|,$$

donde los T_i son todos los sub-árboles de T.

Elección del parámetro α

Se demuestran los siguientes dos resultados:

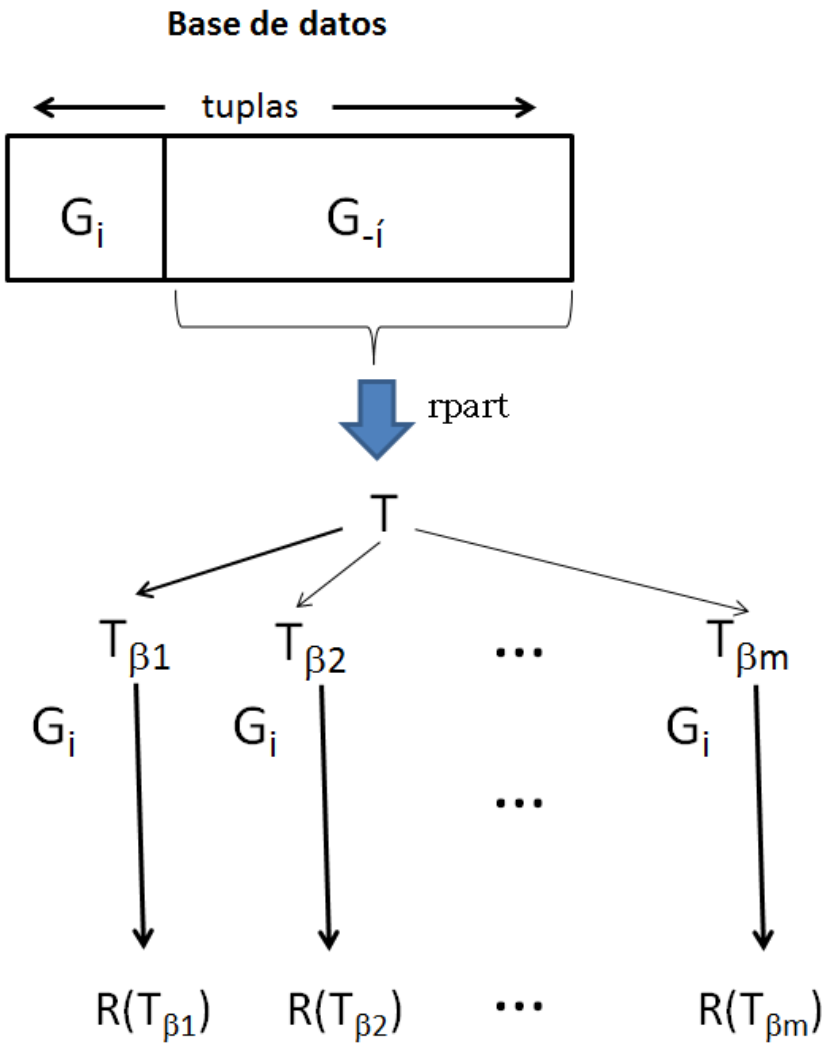
i. Si T_1 y T_2 son sub-árboles de T tales que $R_\alpha(T_1) = R_\alpha(T_2)$ entonces $T_1 = T_2$ o uno de ellos corresponde a un sub-árbol del otro.

ii. Si $\alpha > \beta$, entonces $T_\alpha = T_\beta$ o T_α es sub-árbol de T_β.

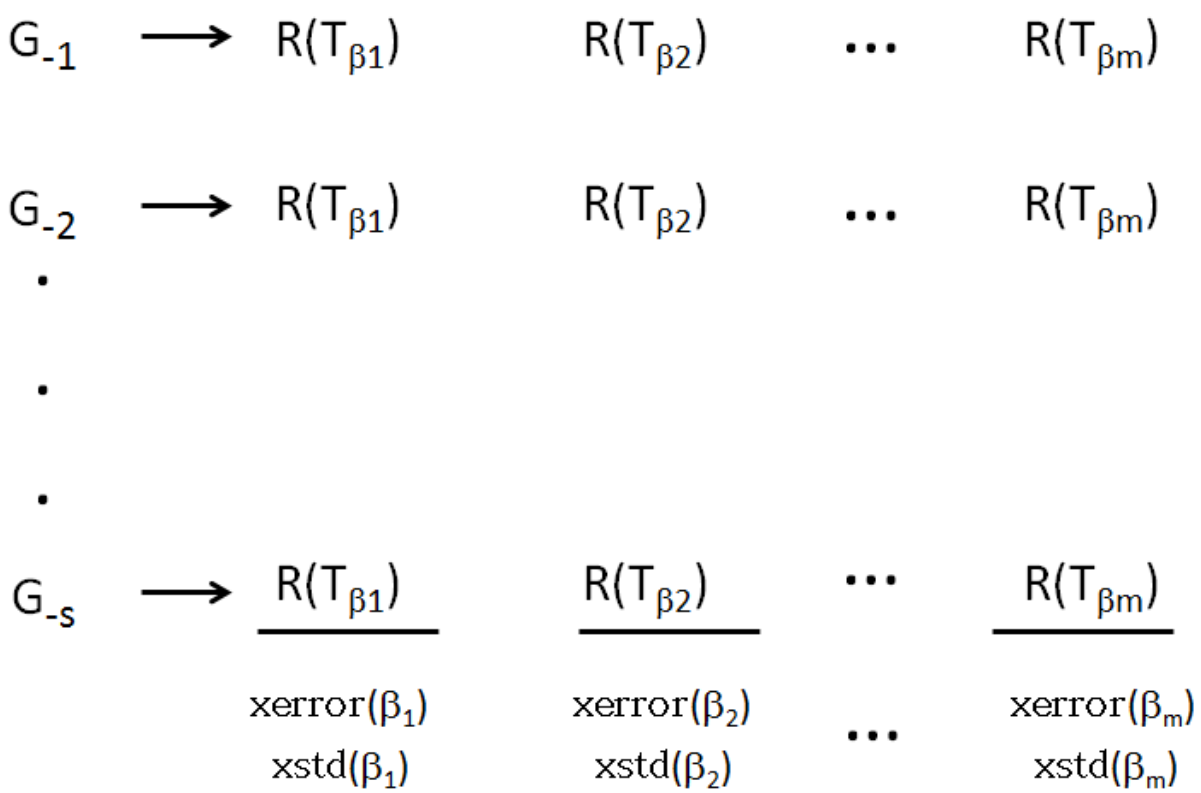
Se define T_α como el árbol más pequeño entre aquellos que minimizan $R_\alpha(T)$.

Estimación de $R(T)$

- Se divide el conjunto de tuplas en s grupos, $G_1, \ldots, G_s$.
- Para cada uno de ellos, G_i, se entrena un árbol de decisión, T, utilizando los datos restantes, G_{-i}, a través de la función `rpart`.
- Para cada parámetro de complejidad candidato, $\beta_j, j = 1, \ldots, m$, se determina T_{β_j}.
- Para cada T_{β_j} se obtiene su costo asociado $R(T_{\beta_j})$, utilizando el grupo de datos que no participó en el entrenamiento del árbol, G_i.



Tipos de Errores y Poda



Tipos de Errores y Poda

Ejemplo salida de R

	CP	nsplit	rel error	xerror	xstd
1	0.1055556	0	1.00000	1.09444	0.0095501
2	0.0888889	2	0.79444	1.01667	0.0219110
3	0.0777778	3	0.70556	0.90556	0.0305075
4	0.0666667	5	0.55556	0.75000	0.0367990
5	0.0555556	8	0.36111	0.56111	0.0392817
6	0.0166667	9	0.30556	0.36111	0.0367990
7	0.0111111	11	0.27222	0.37778	0.0372181
8	0.0083333	12	0.26111	0.36111	0.0367990
9	0.0055556	16	0.22778	0.35556	0.0366498
10	0.0027778	27	0.16667	0.34444	0.0363369
11	0.0013889	31	0.15556	0.36667	0.0369434
12	0.0000000	35	0.15000	0.36667	0.0369434

Tipos de Errores y Poda