Aprendizaje Supervisado: Árboles de Decisión.

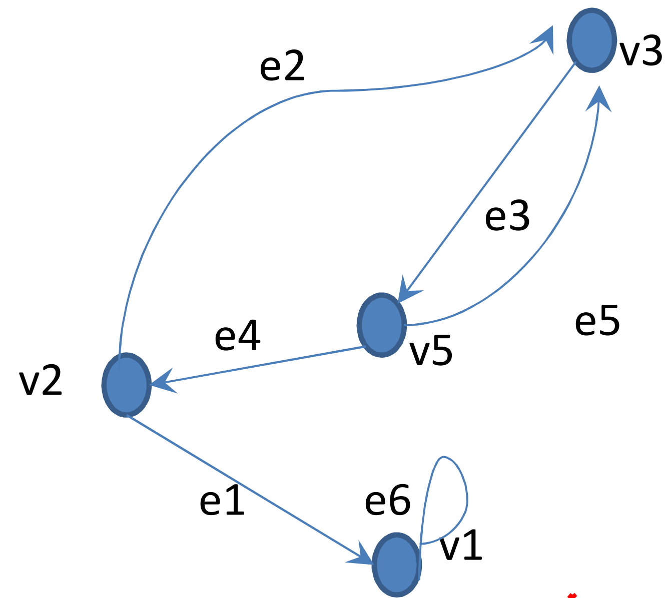
- Introducción
  - Antecedentes
- Árboles de Decisión (DT)
  - Algoritmo
  - Entropía para construir DTs
- Ejemplo

Preliminares de DT: Teoría de Grafos

Grafo:

$G = (V, E)$ *V vértices, E aristas*

Si las aristas están ordenadas, el grafo es <u>dirigido</u>.



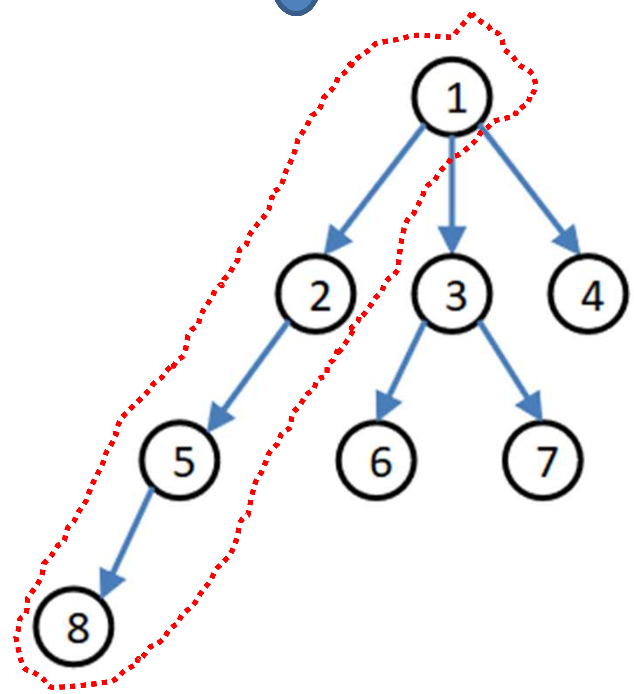
<u>Camino en el grafo:</u>

Secuencia de pares ordenados $(v_i, v_j), \ldots, (v_k, v_n)$ donde v_i es el nodo/vértice inicial, v_n el nodo final correspondiente arista.

<u>Grafo acíclico dirigido</u>

Grafo dirigido donde las aristas no forman ningún ciclo. Satisface 3 propiedades:

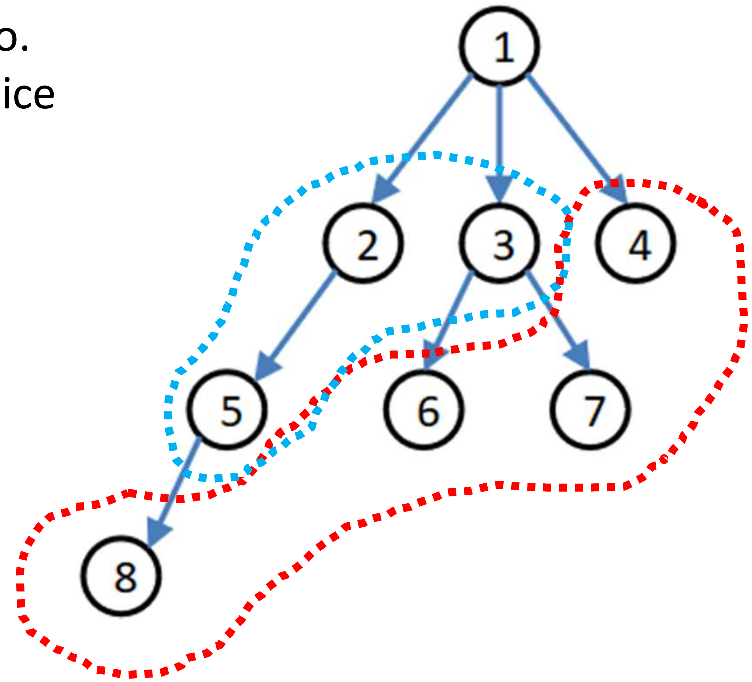
- Tiene un solo Nodo raíz
    - Sin aristas entrantes.
- Todos los nodos excepto el raíz, tiene una sola arista entrante
- Hay exactamente un solo camino entre la RAÍZ y cada nodo.



Preliminares de DT: Teoría de Grafos

Padres e Hijos:

Si (v, w) es una arista formada entre los nodos v y w, se dice que el primero es el padre del segundo. Si existe un camino entre v y w tal que $(v \neq w)$ se dice que v es el ancestro propio de w, así como w es el descendiente propio de v.



Nodo Hoja:

Cuando un nodo v_i no tiene descendientes.

Nodo Interno:

Cualquier nodo que no es raíz ni hoja.

Profundidad, Anchura y Nivel:

$P(v_j) = length(v_r, v_j)$ donde v_r es el nodo raíz.

$A = \max_{v_k \in G}(length(v_r, v_k))$ donde v_r es el nodo raíz.

$L(v_j) = A - P(v_j)$.

Preliminares de DT:
Teoría de Grafos

Árbol Ordenado:
Un árbol cuyos nodos hoja están ordenados de izq. a der.

Árbol Binario Ordenado:
- Solo dos hijos (hijo izq. o der.)
- Ningún nodo tiene mas de un hijo izq./der.

T

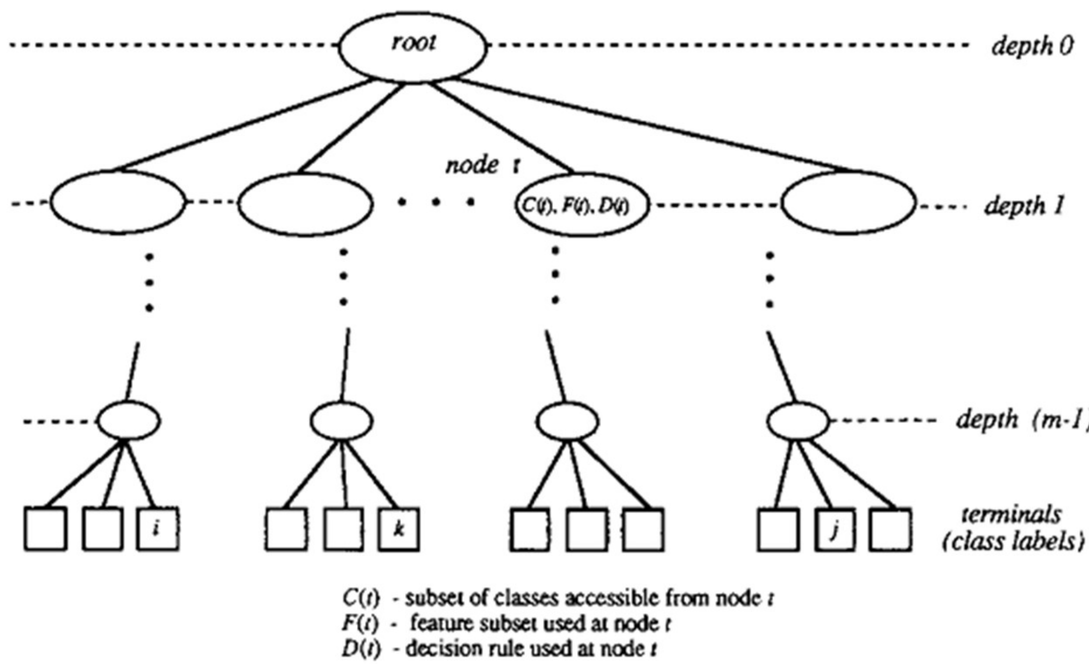
Profundidad promedio:

$$\mu^{p} = \frac{1}{N} \sum_{i \in Hojas} Capas(v_r, v_i)$$

Anchura promedio:

$$\mu^{A} = \frac{1}{N} \sum_{i \in L} |v_I|$$

$$v_I = \{v_k, \ldots, v_l\} \in L_i$$



Problema de Clasificación Binario

Dado un conjunto de datos en la forma

$$(\bar{x}, y) \in X x Y$$

$$x \in \Re^N$$

$$y \in \{\pm 1\}$$

Se busca una función f (.), tal que

$$f(\bar{x}_{new}) \to \{y = \pm 1\}$$

Clasificadores

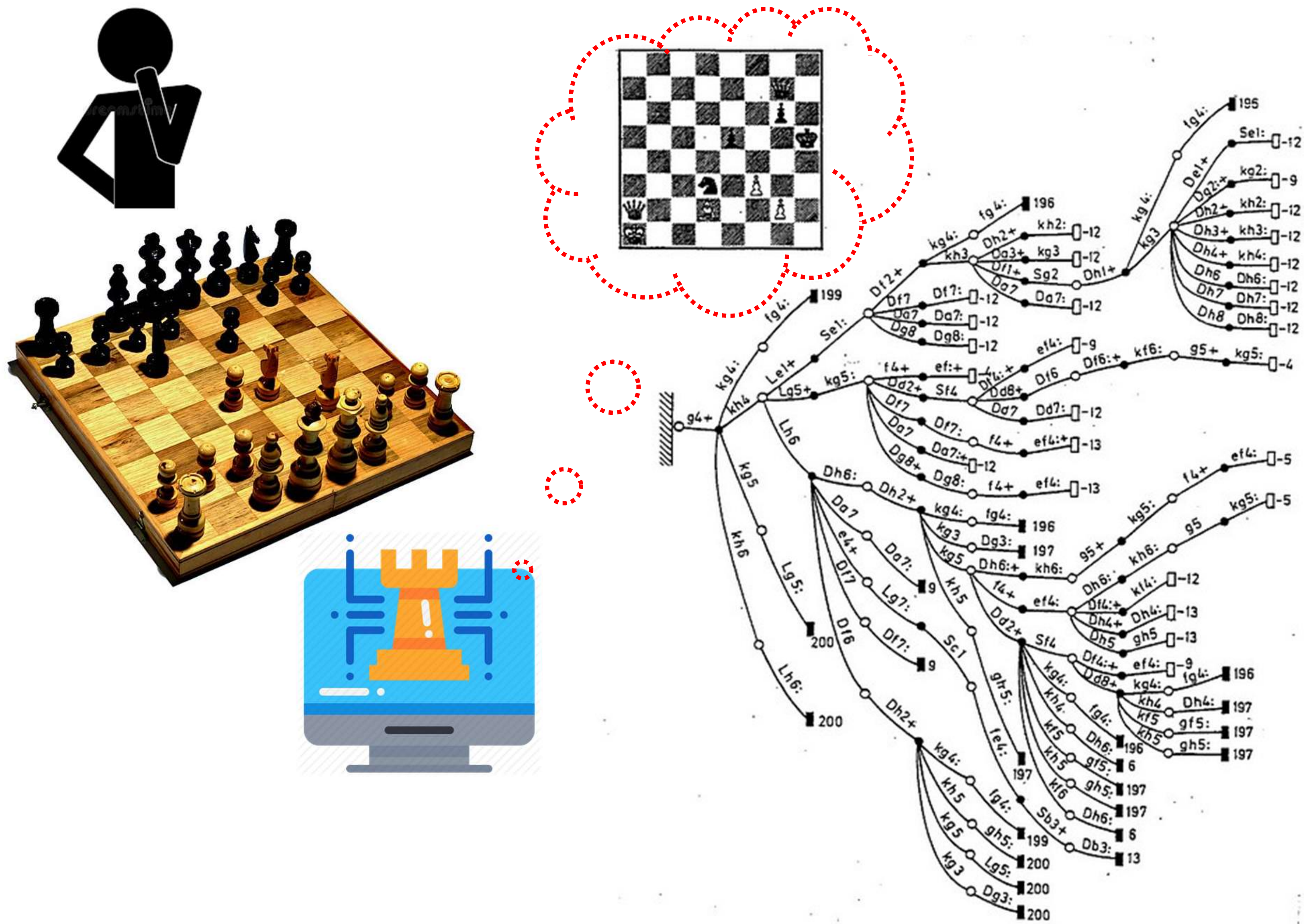
K-NN

- Sin MODELO
  - Sin supuestos
- Aproximación Local
- Emplea:
  - BD de Referencia
  - Sin Entrenamiento
  - $D = (\boldsymbol{x}_i, \boldsymbol{x}_{new})$
  - K: número de vecinos
- NO INTERPRETABLE

DT

Regresión Logística

- MODELO LINEAL
- $\log\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \varepsilon$
- Aproximación Global
- Emplea:
- BD de Referencia
- Con entrenamiento
- $\boldsymbol{\beta}$
- Máxima Verosimilitud (en inglés Maximum Likelihood)
- INTERPRETABLE



Quinlan J.R. (1983) **Learning Efficient Classification Procedures and Their Application to Chess End Games**. In: Michalski R.S., Carbonell J.G., Mitchell T.M. (eds) Machine Learning. Symbolic Computation. Springer, Berlin, Heidelberg

ID3

Métodos Basados en Árboles

Dado

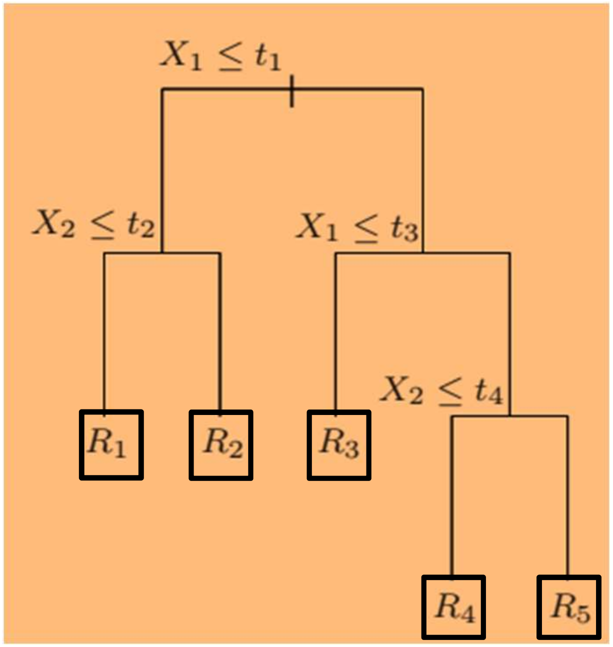
$$\left(\overline{x}, y\right) \in X x Y$$

$$\overline{x} \in \Re^N$$

$$y \in \Re$$

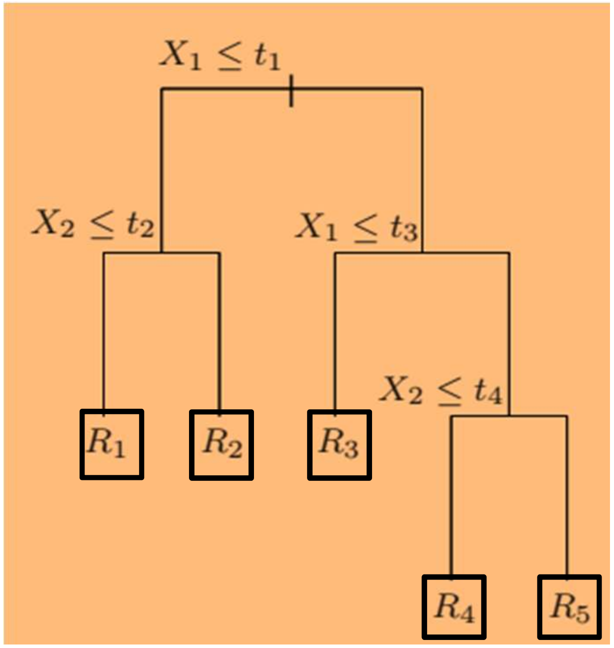
Se busca

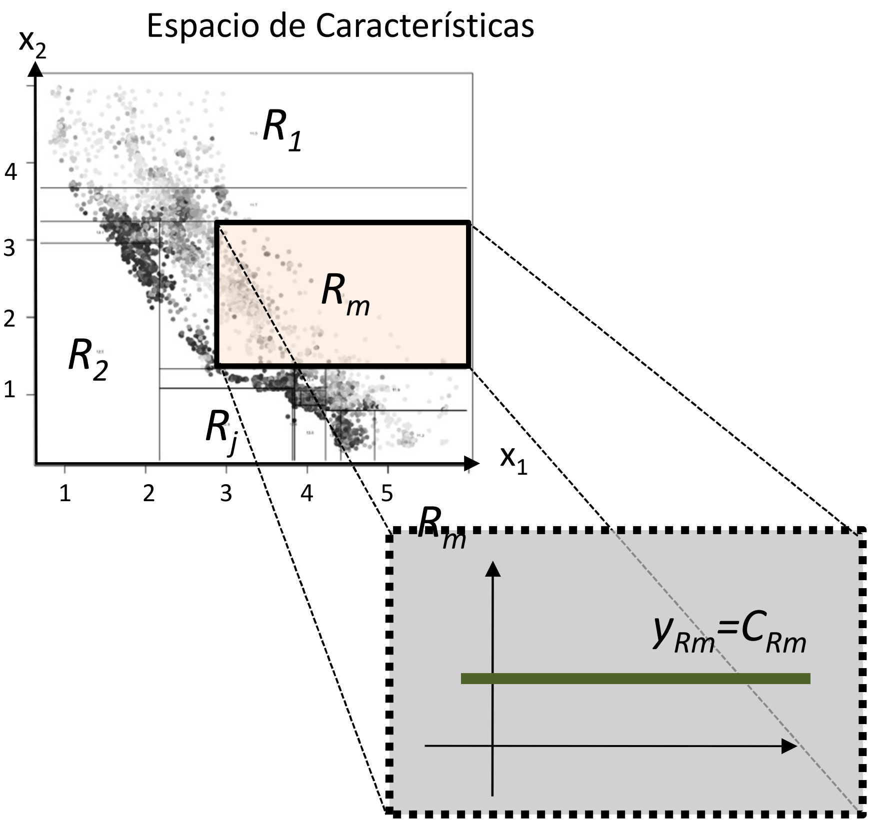
$$f : X \rightarrow Y$$



- Clasificador en forma de un **<u>árbol</u>**:
  - **Nodo Decisión**: especifica una prueba sobre una sola variable
  - **Nodo Hoja**: Índica la clase
  - **Arco**: División de una variable
  - **Camino**: Regla de asociación

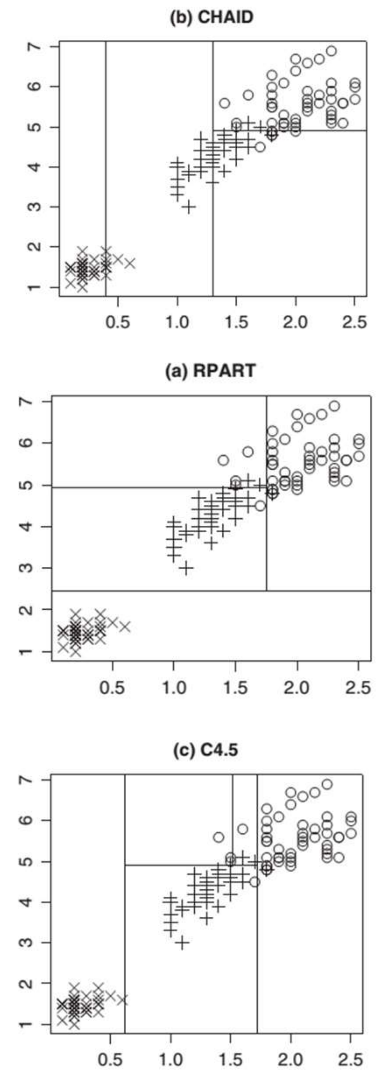
Un DT clasifica un $\boldsymbol{x}_{new}$ comenzando desde el nodo **raíz** y moviéndose por los nodos hasta llegar a un nodo **hoja.**





Algoritmos para construir DT

Algoritmo	Tipo de Problema C (clasificación) R (regresión)	Variables	
		Nominales	Numéricas
ID3 (Quinlan, 1979)	C	√	
CHAID (Kass, 1980)	C	√	
CART (Breiman et al, 1984)	C/R	√	√
C4.5 (Quinlan, 1993)	C	√	√
M5’ (Witten & Frank,2000)	R	√	√



W. Loh. Fifty Years of Classification and Regression Trees. International Statistical Review, 82(3):329{348, 2014

Classification And Regression Trees (CART) alias Recursive Partitions (RPart)

- Es árbol binario construido iterativamente, su algoritmo es el siguiente:

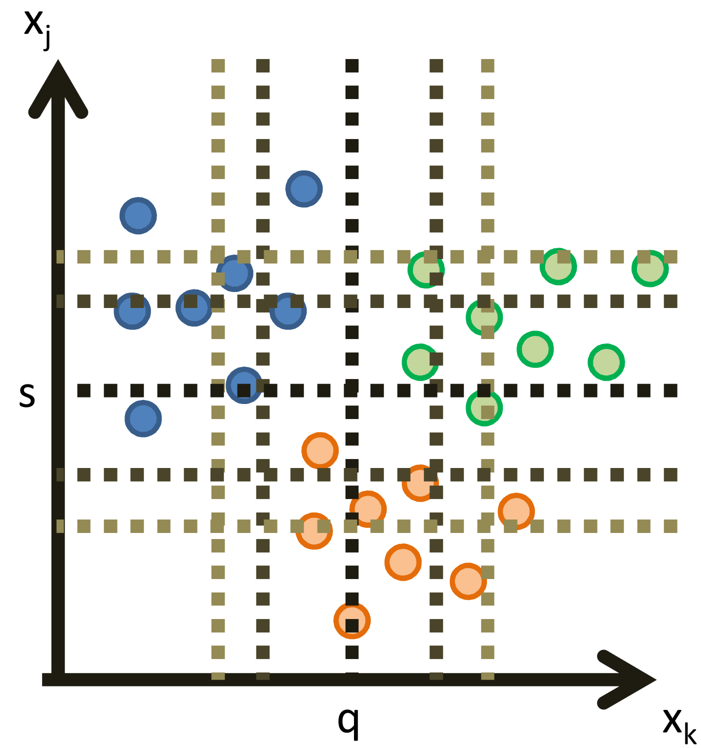
1. Encontrar para cada variable la mejor **división**
   - Para una variable con k valores, existen k-1 particiones

   a) Encuentre la división que maximice el **criterio de división.**

   $s_i^j \in S$, la división en el valor i de la variable j que satisface $\theta\left(s_i^j\right) \geq \theta\left(s_k^j\right), \forall k \neq i.$

2. Encontrar la mejor división $s^* = s_i^j \mid \theta\left(s_i^j\right) \geq \theta\left(s_k^l\right), \forall l \neq j.$

3. Dividir el nodo usando s^*, y repetir los pasos 1 y 2 hasta satisfacer un criterio de paro**.



** No detallaremos ni el criterio de paro, ni el podado de un árbol.

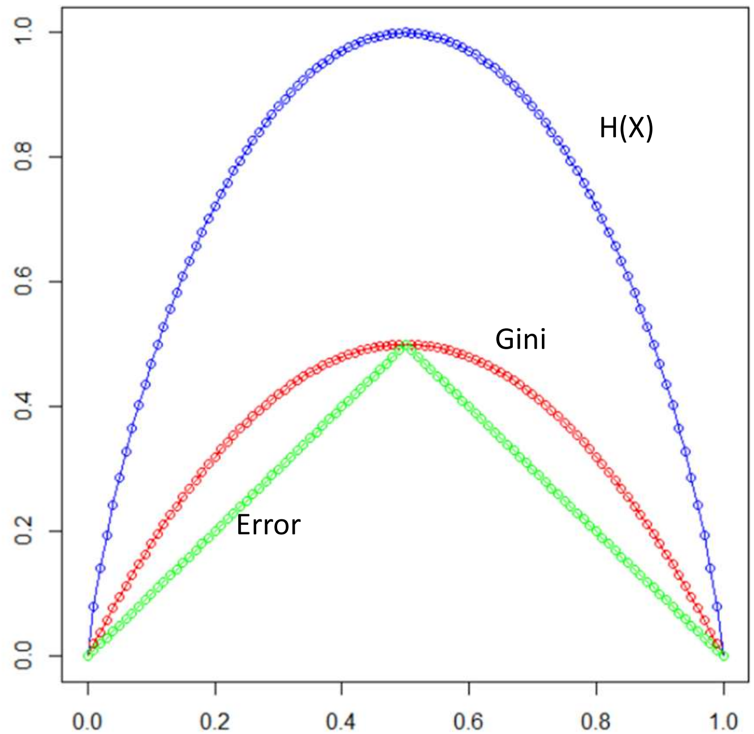
Pero… ¿Cómo construimos un DT? Medidas de Pureza

$$H(X) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

$$Gini(X) = 1 - \sum_{i}^{c} p_i^2$$

$$Error(X) = 1 - \max_{i \in c} p_i$$

$$I(X) \neq H(X)^{**}$$



P(X=1)

Usando medidas de pureza se crean los **criterios de división**

- **Ganancia de Información** (IG):

$$IG(T, x_i) = H_{Padre}(X) - H_{hijo}(X|x_i)$$

1. Calcular $H_{padre}(X)$
2. Para todas las variables y sus valores calculamos

$$H(x_i, s_k) = -\sum_{j=1}^{c} p_j \log_2 p_j$$

$\boldsymbol{p_j}$ es la probabilidad de la clase m
c es el número de clases

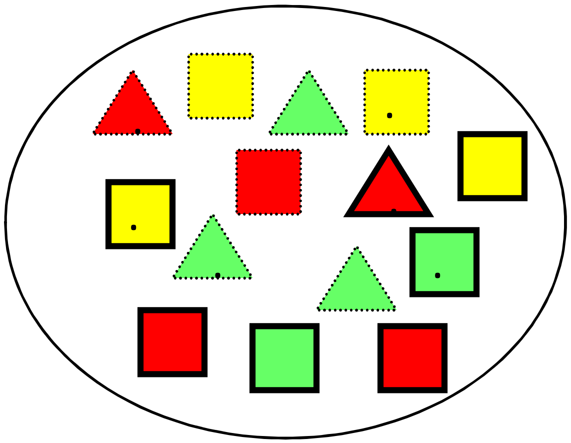
3. La entropía de cada variable hijo x_i se calcula como

$$H_{hijo}(X|x_i) = -\sum_{j=1}^{k} p_j \, H(x_i, s_k)$$

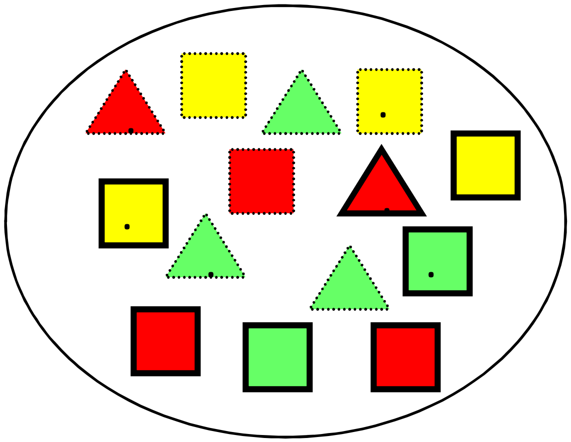
4. Seleccionar la que minimice $IG(T, x_i)$

Entrenando un DT:
Triángulos y Cuadrados

#	Attribute			Shape
	Color	Outline	Dot	
1	green	dashed	no	triange
2	green	dashed	yes	triange
3	yellow	dashed	no	square
4	red	dashed	no	square
5	red	solid	no	square
6	red	solid	yes	triange
7	green	solid	no	square
8	green	dashed	no	triange
9	yellow	solid	yes	square
10	red	solid	no	square
11	green	solid	yes	square
12	yellow	dashed	yes	square
13	yellow	solid	no	square
14	red	dashed	yes	triange



Calculamos la entropía de todo el conjunto



- 5 triángulos
- 9 cuadrados
- Probabilidades de clase:

$$p(\square) = \frac{9}{14}$$

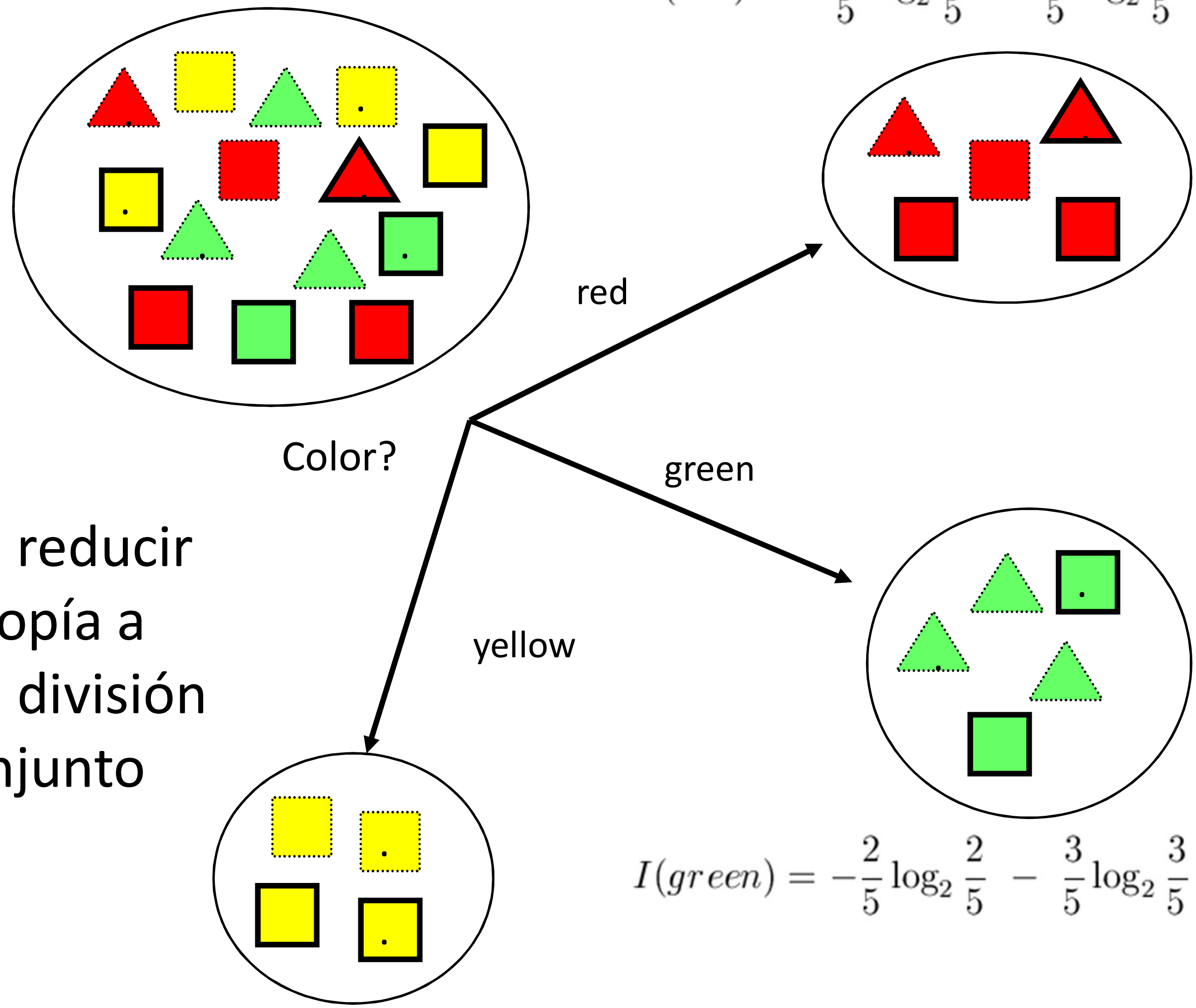
$$p(\triangle) = \frac{5}{14}$$

- Entropía:

$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ bits}$$

$$I = 0.940$$

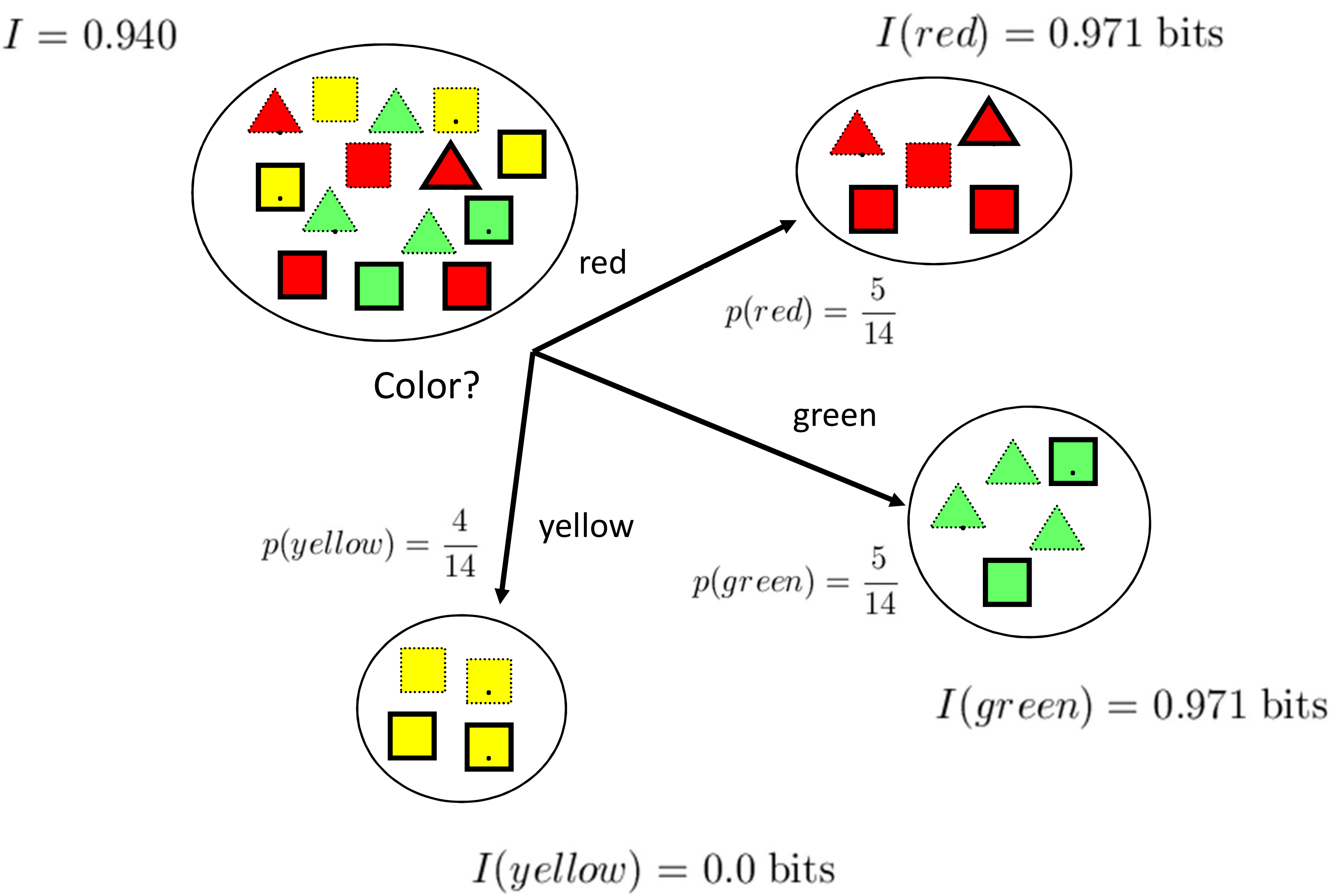
$$I(red) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$$



Se busca reducir la entropía a partir del división del conjunto

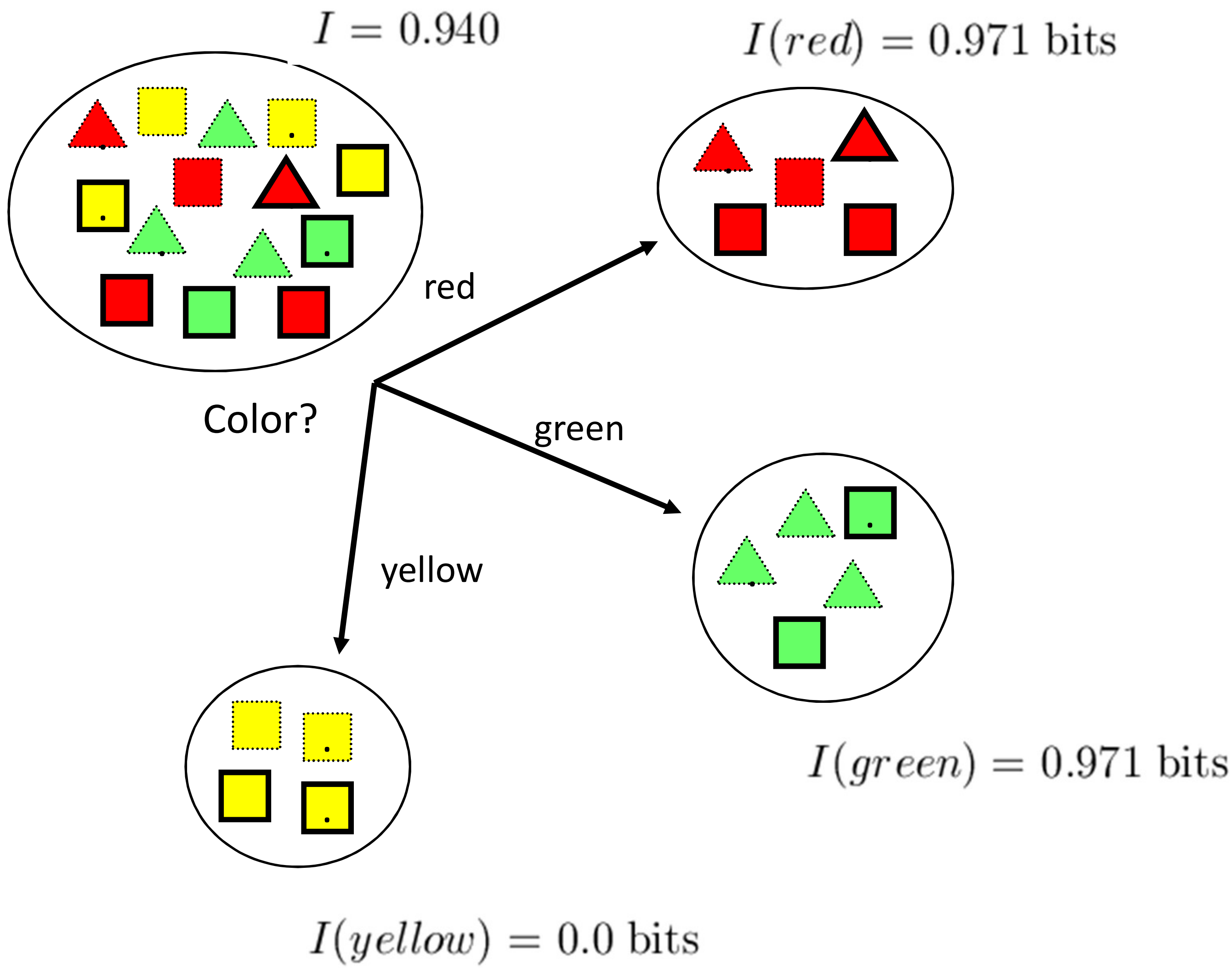
$$I(green) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \text{ bits}$$

$$I(yellow) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0.0 \text{ bits}$$



$$I_{res}(\text{Color}) = \sum p(v)I(v) = \frac{5}{14}0.971 + \frac{5}{14}0.971 + \frac{4}{14}0.0 = 0.694\ bits$$

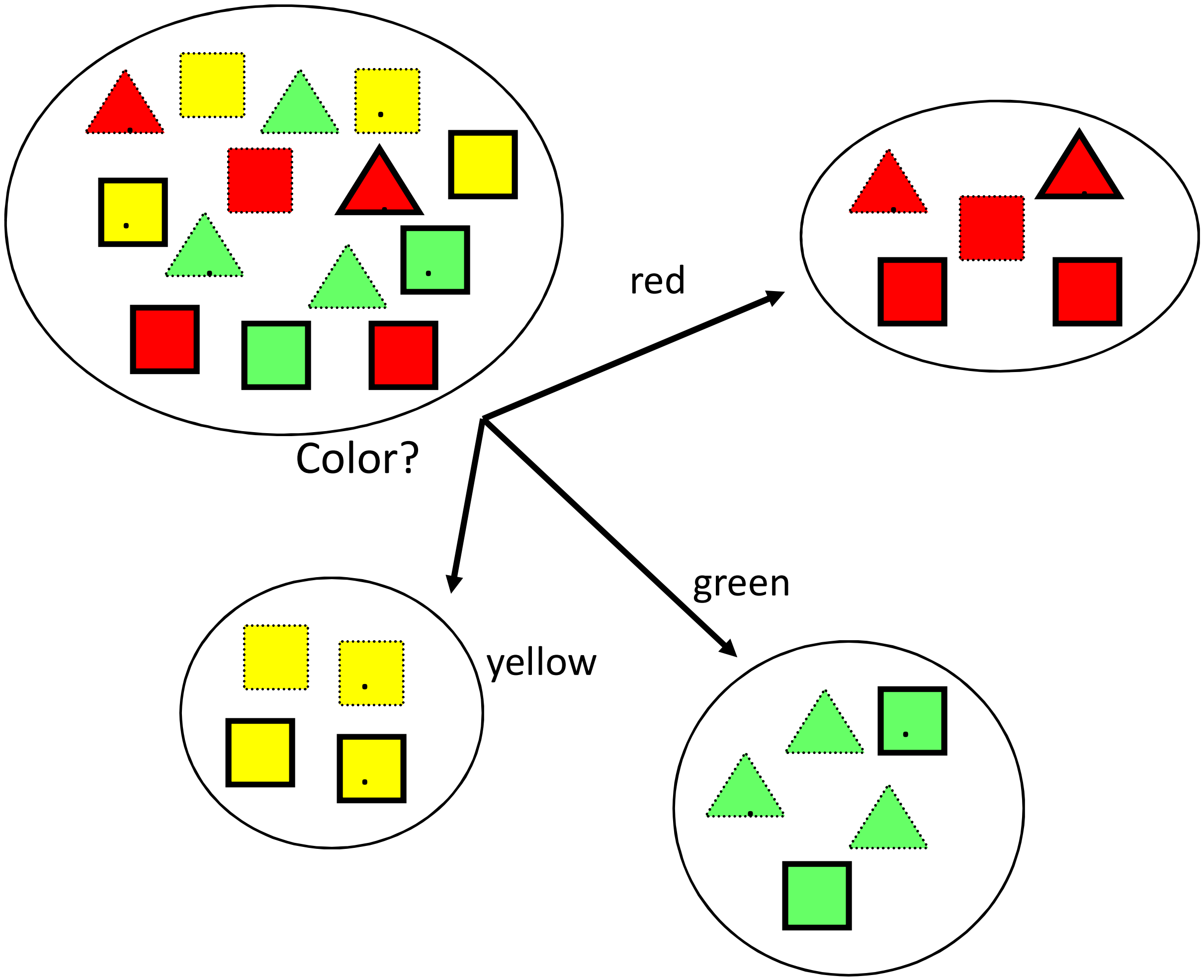
Ganancia de información



$$\boxed{Gain(\text{Color}) = I - I_{res}(\text{Color}) = 0.940 - 0.694 = 0.246\ bits}$$

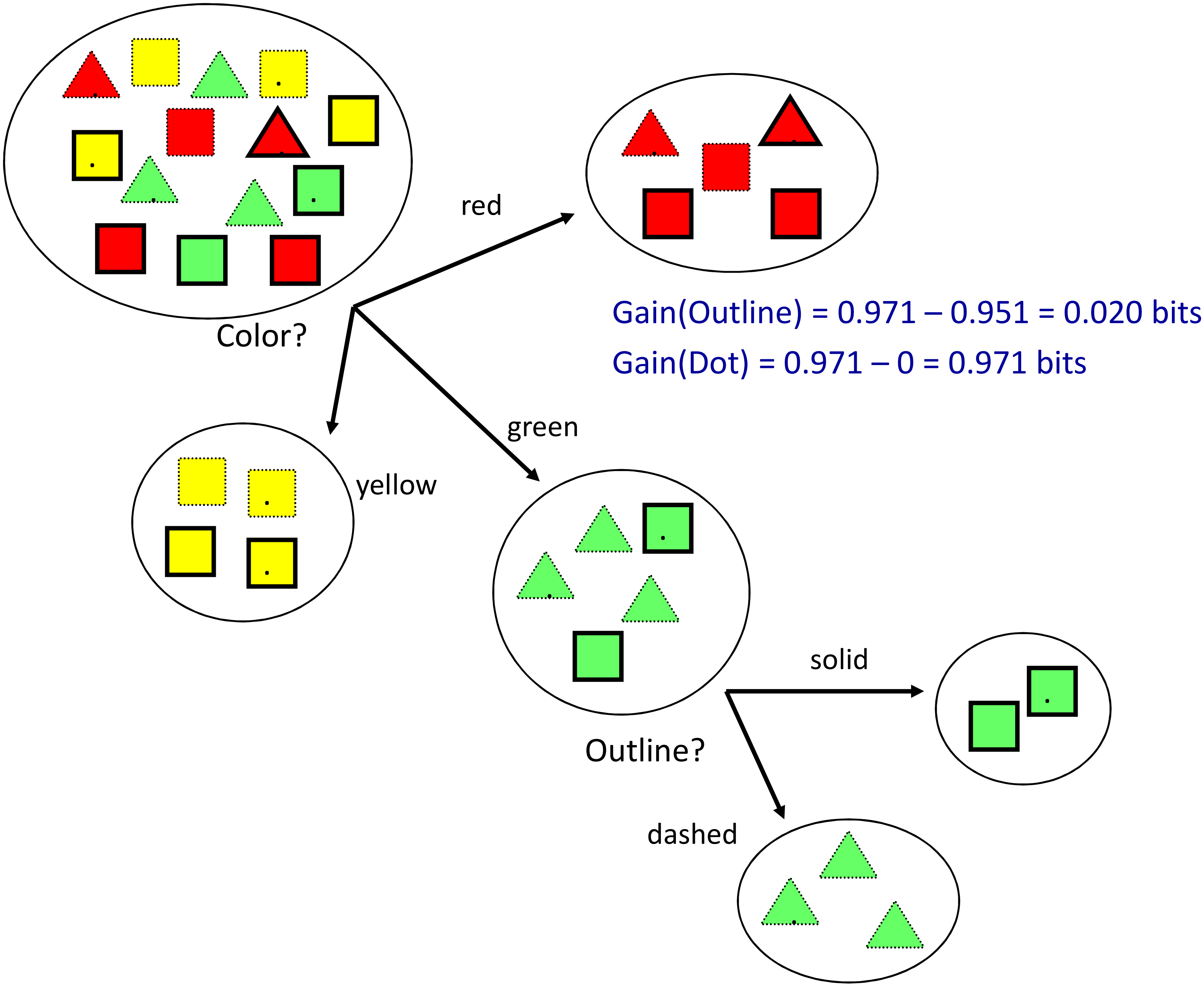
Ganancia de información de los atributos

- Atributos:
  - Gain(Color) = 0.246
  - Gain(Outline) = 0.151
  - Gain(Dot) = 0.048
- Heuristíca: El atributo con la más alta ganancia es seleccionado para particionar el conjunto.

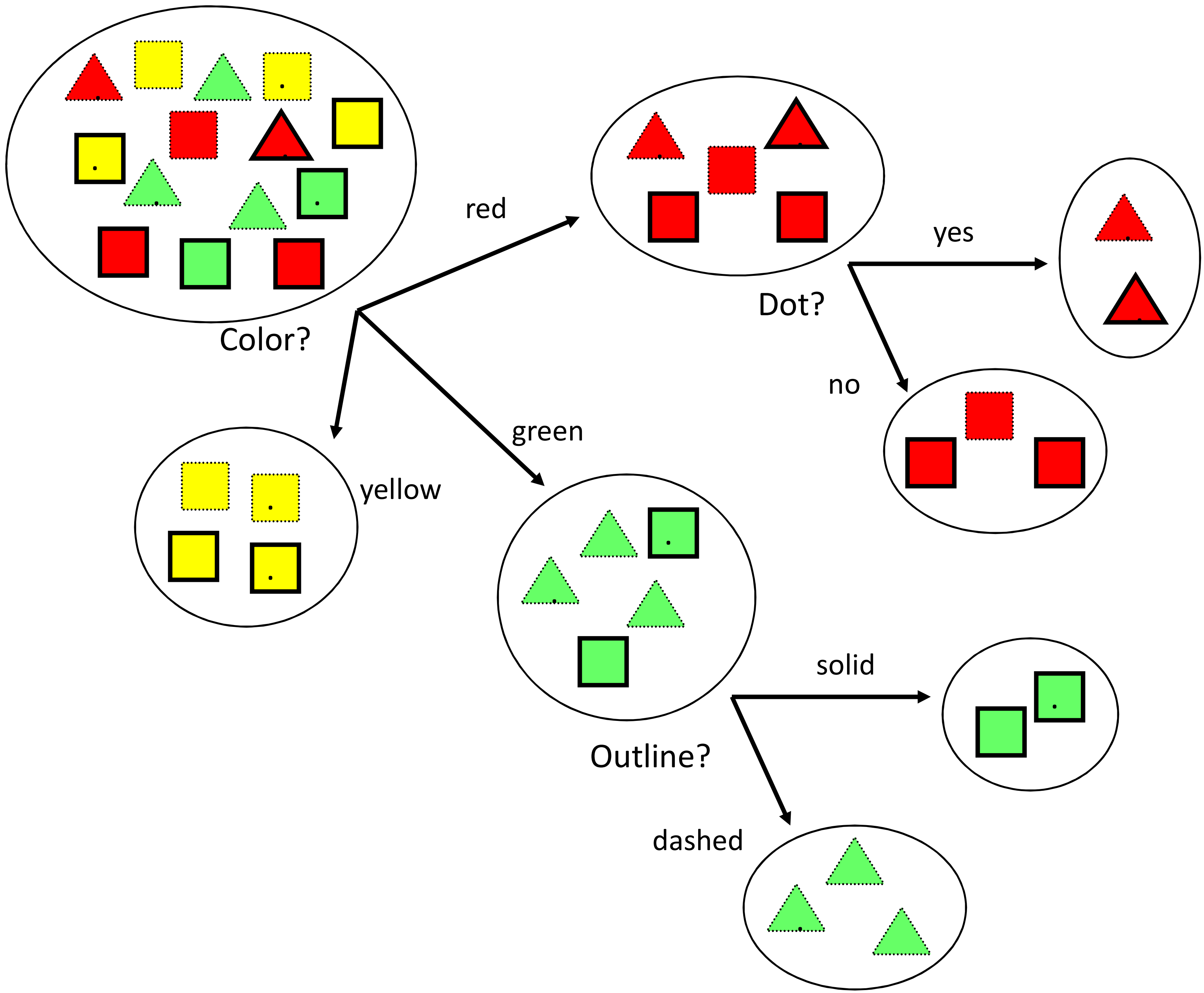


Gain(Outline) = 0.971 – 0 = 0.971 bits

Gain(Dot) = 0.971 – 0.951 = 0.020 bits

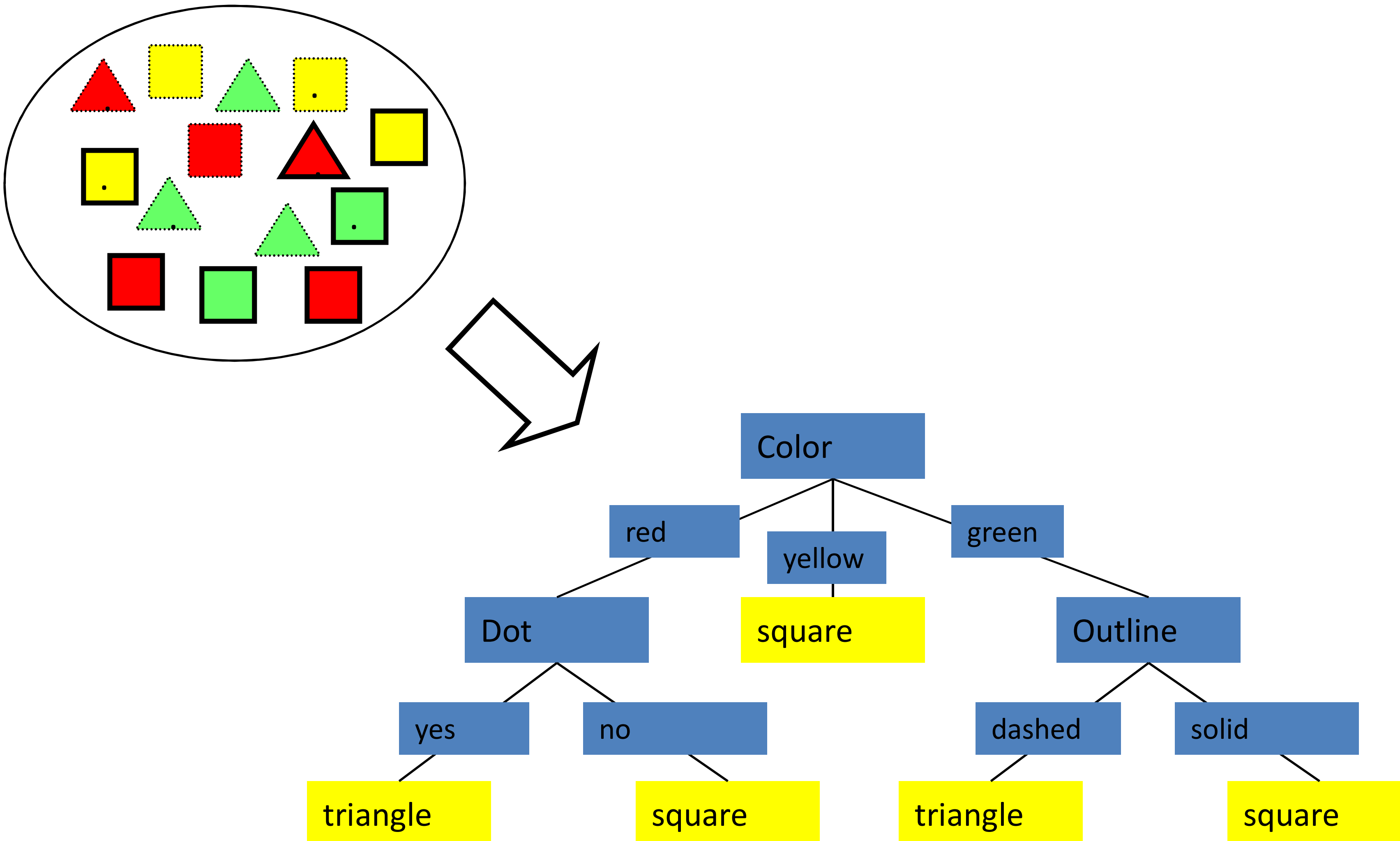


red
Gain(Outline) = 0.971 – 0.951 = 0.020 bits
Gain(Dot) = 0.971 – 0 = 0.971 bits
Color?
green
yellow
solid
Outline?
dashed



Color?
red
yellow
green
Dot?
yes
no
Outline?
solid
dashed

Árbol de decisión resultante



Discusión

- No requiere Modelo
- Aproximación Local
- INTERPRETABLE

- ***Pros***:
  - Selección e Importancia de Variables.
  - Manejo de valores faltantes.
  - Transformación de Datos.
  - Representación del conocimiento en forma de reglas

- ***Cons***:
  - Tienden a sobre-generalizar

¿PREGUNTAS?