

Decision Tree Learning

José David Rojas

Definición

- ▶ Por lo tanto, los DT son un modelo no paramétrico.
- ▶ Para crear el DT, se **intenta** optimizar el promedio de la máxima verosimilitud:

$$\frac{1}{N} \sum_{i=1}^N (y_i \ln f_{ID3}(\mathbf{x}_i) + (1 - y_i) \ln (1 - f_{ID3}(\mathbf{x}_i)))$$

donde f_{ID3} es un DT y $f_{ID3}(\mathbf{x}) \stackrel{\text{def}}{=} Pr(y = 1|\mathbf{x})$

Construcción

- ▶ Para construir el árbol, en cada nodo de decisión, se intenta minimizar la entropía de la información.
- ▶ La entropía de un conjunto \mathcal{S} viene dada por:

$$H(\mathcal{S}) \stackrel{\text{def}}{=} -f_{ID3}^{\mathcal{S}} \log_2(f_{ID3}^{\mathcal{S}}) - (1 - f_{ID3}^{\mathcal{S}}) \log_2(1 - f_{ID3}^{\mathcal{S}})$$

- ▶ Y si un grupo se divide en dos, la entropía es la suma ponderada de cada subconjunto:

$$H(\mathcal{S}_-, \mathcal{S}_+) \stackrel{\text{def}}{=} \frac{|\mathcal{S}_-|}{|\mathcal{S}|} H(\mathcal{S}_-) + \frac{|\mathcal{S}_+|}{|\mathcal{S}|} H(\mathcal{S}_+)$$

Ejemplo

Consideremos los siguientes datos:

Atributos

- ▶ Edad: viejo (v), media-vida(m), nuevo (nv)
- ▶ Competencia: no(n), sí(s)
- ▶ Tipo: software (swr), hardware (hwr)

Edad	Competencia	Tipo	Ganancia
v	s	swr	baja
v	n	swr	baja
v	n	hwr	baja
m	s	swr	baja
m	s	hwr	baja
m	n	hwr	sube
m	n	swr	sube
nv	s	swr	sube
nv	n	hwr	sube
nv	n	swr	sube

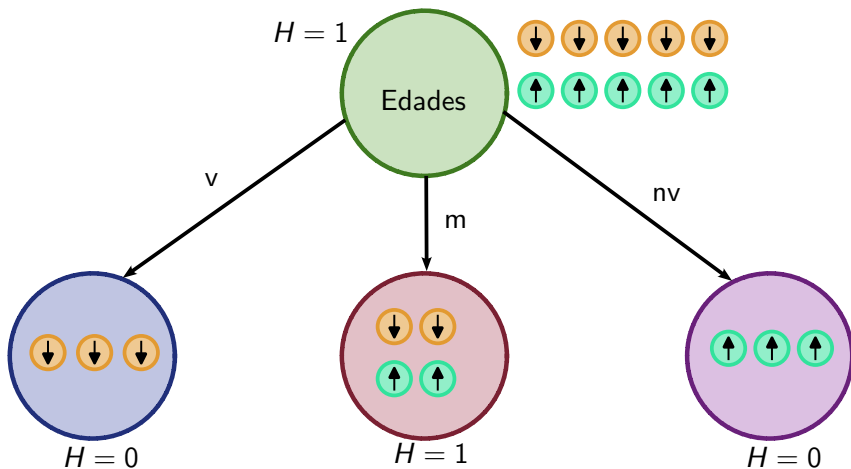
Cálculo de las entropías

Primero se tiene que probar todos los features para ver cuál tiene mayor ganancia de información (reduce la entropía) Entropía total:

$$\begin{aligned} H(S) &= \text{Entropía de los casos baja} + \text{Entropía de los casos sube} \\ &= -\frac{5}{10} * \log_2\left(\frac{5}{10}\right) - \frac{5}{10} * \log_2\left(\frac{5}{10}\right) \\ &= 1 \end{aligned}$$

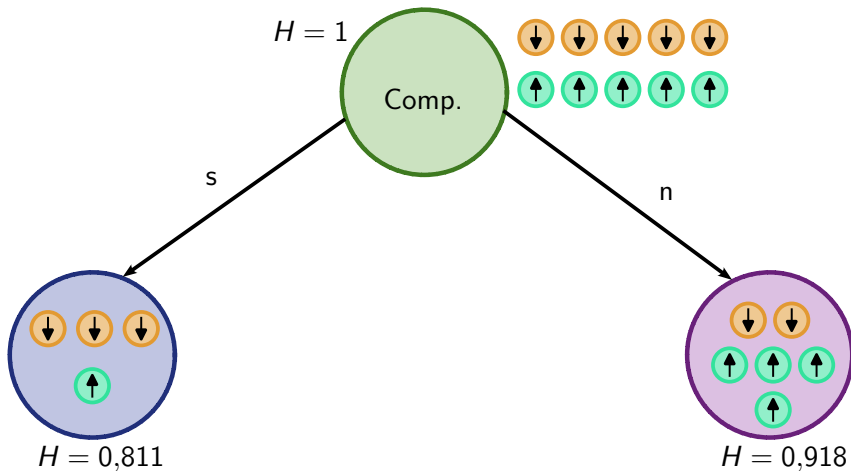
Máxima entropía

Decidiendo con las edades



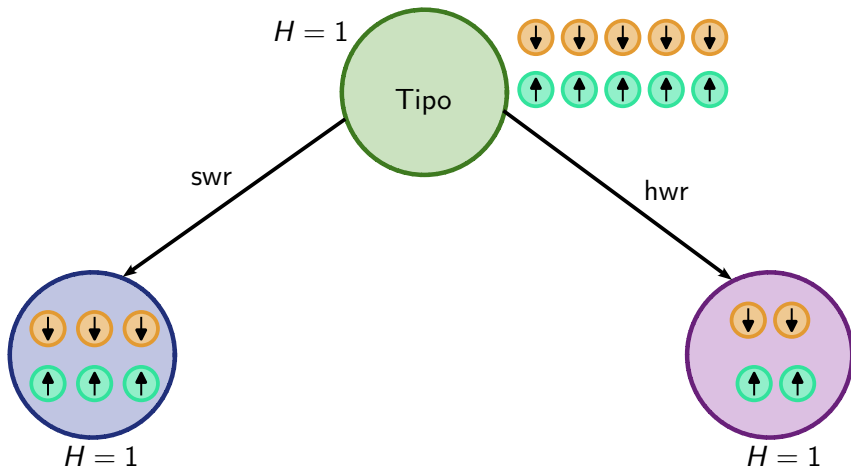
$$H = \frac{3}{10} \cdot 0 + \frac{4}{10} \cdot 1 + \frac{3}{10} \cdot 0 = 0,4$$

Decidiendo con la competencia



$$H = \frac{4}{10} \cdot 0,811 + \frac{6}{10} \cdot 0,918 = 0,8752$$

Decidiendo con el tipo

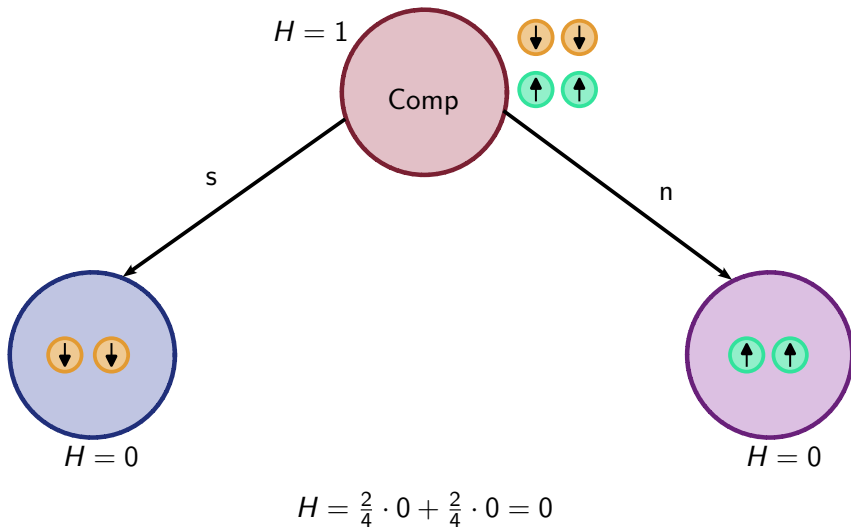


$$H = \frac{6}{10} \cdot 1 + \frac{4}{10} \cdot 1 = 1$$

Primer nivel

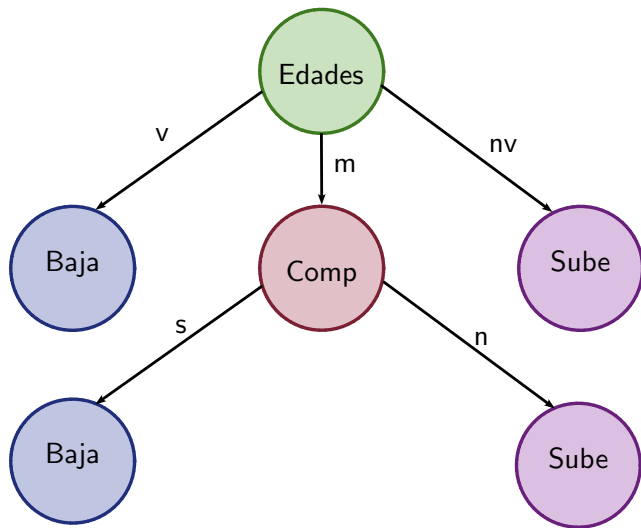
- ▶ Concluimos que lo que nos da la máxima ganancia de información es primero decidir por edades, eso nos deja dos nodos hoja y un nodo rama que debemos volver a separar

Segundo nivel por competencia

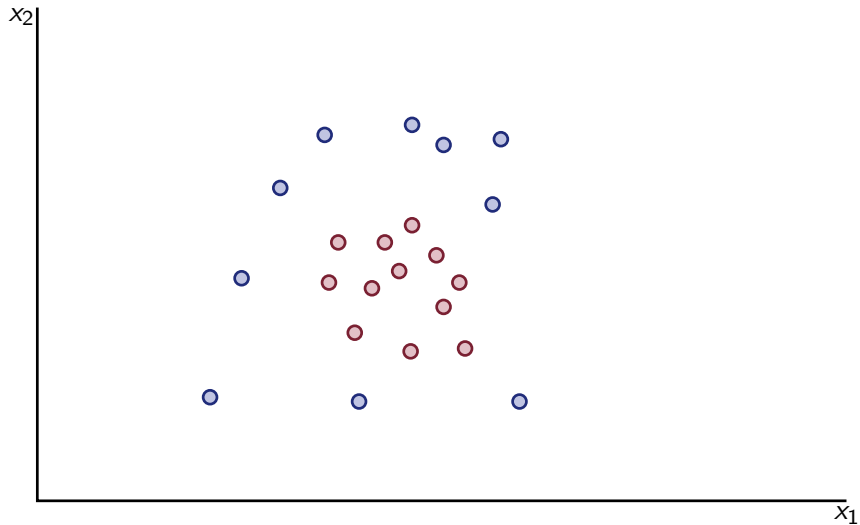


Final

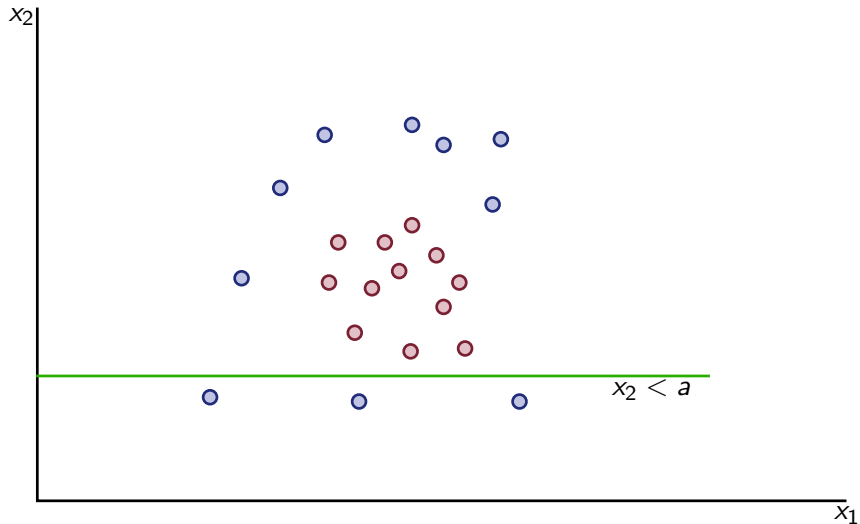
- Con esto ya se clasificaron todos los datos, puesto que terminamos solo con nodos hojas



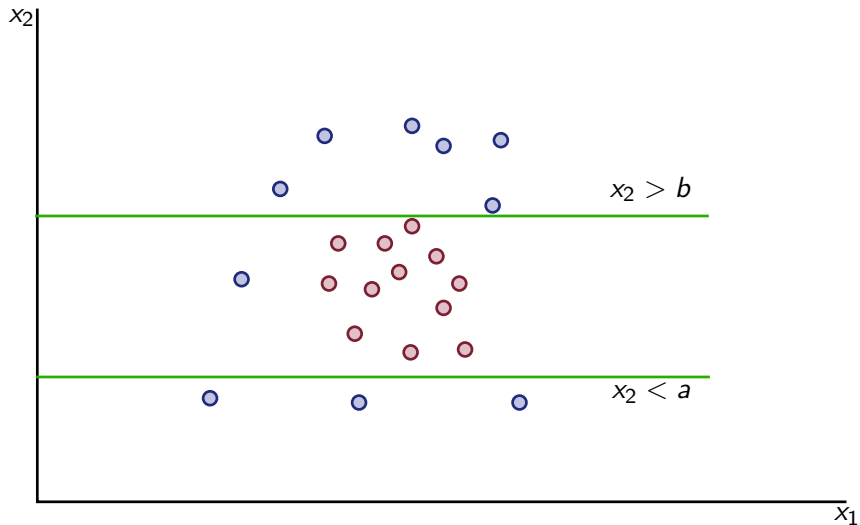
Con valores numéricos



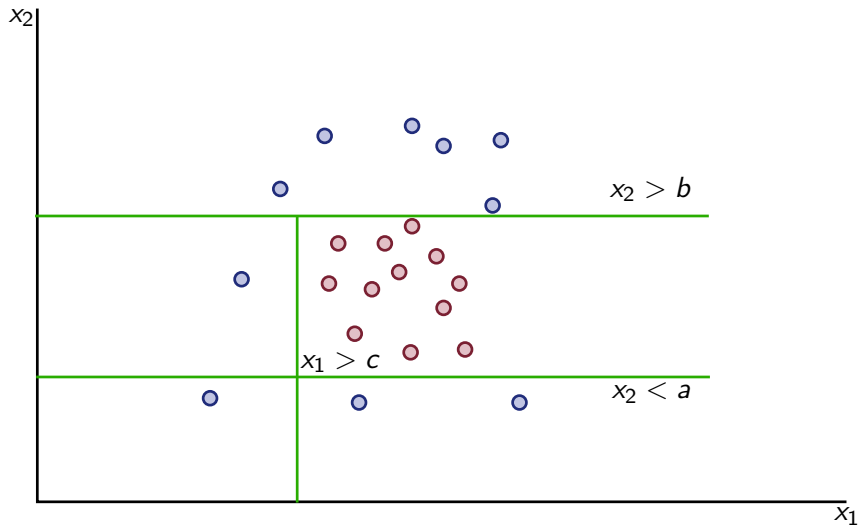
Con valores numéricos



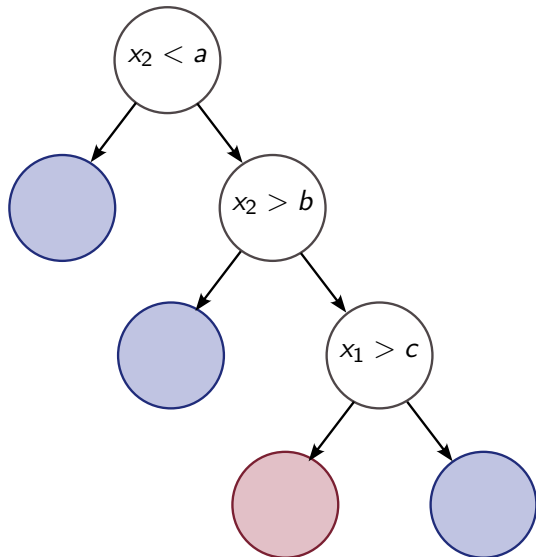
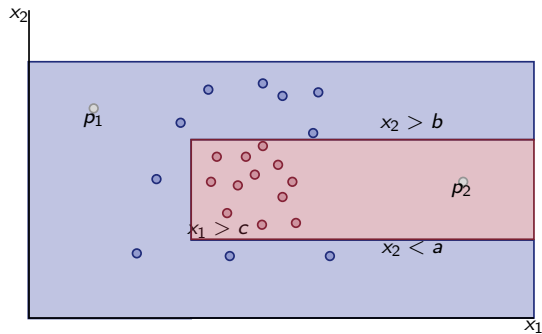
Con valores numéricos



Con valores numéricos



Resultado final con valores numéricos



Usando Scikit Learn

Estos son los comandos básicos para utilizar un árbol con Scikit learn

```
from sklearn import tree
X = # Lista con los features (lista de listas)
Y = # Lista con los labels
# Se define la variable que tendrá el árbol
clf = tree.DecisionTreeClassifier()
# Se calcula el árbol
clf = clf.fit(X, Y)
# Se utiliza el árbol para predecir el label de un dato nuevo
clf.predict_proba(X0)
# Se dibuja el árbol
tree.plot_tree(clf)
```

Decision Tree Learning

José David Rojas