



Aprendizaje Automático

Árboles de decisión

Laura de la Fuente, Hernán Bocaccio

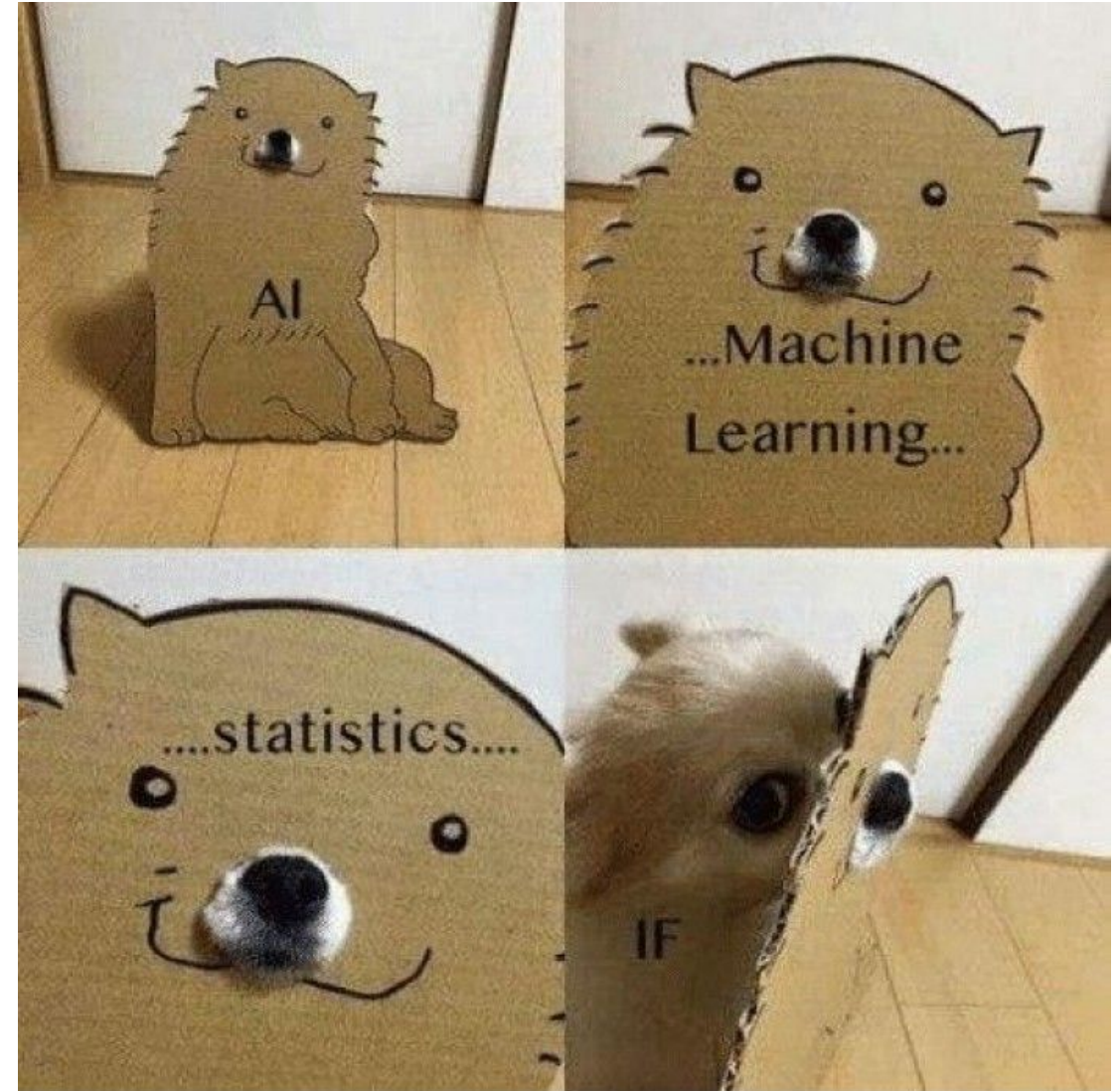
Ayudantes: Gastón Bujía, Diego Onna y Sofía Morena del Pozo

Dirección de e-mail de la materia:

datawillconfess@gmail.com

Itinerario de la clase

- Decisión (regla if-then)
- Árboles de decisión (clasificación)
- Construcción de un árbol
- Inducción top-down
- Criterios de impurezas
- Contornos de decisión
- Complejidad del árbol
- Pruning
- Árboles de decisión (regresión)

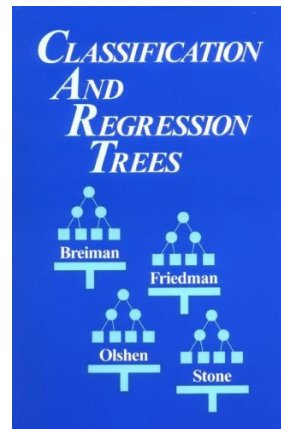


Árboles de decisión

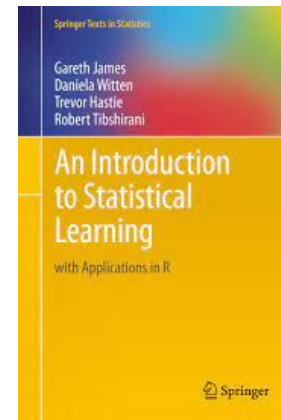
Aprendizaje supervisado: datos anotados

- Clasificación
- Regresión

CART: Classification and regression trees



Breiman et al 1984



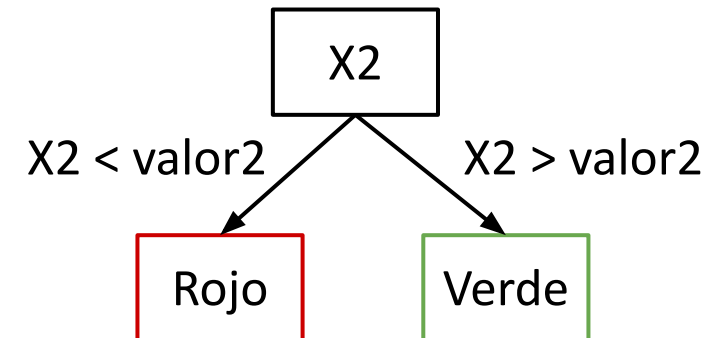
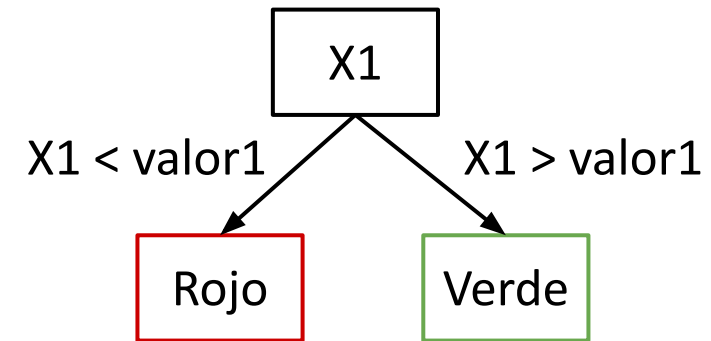
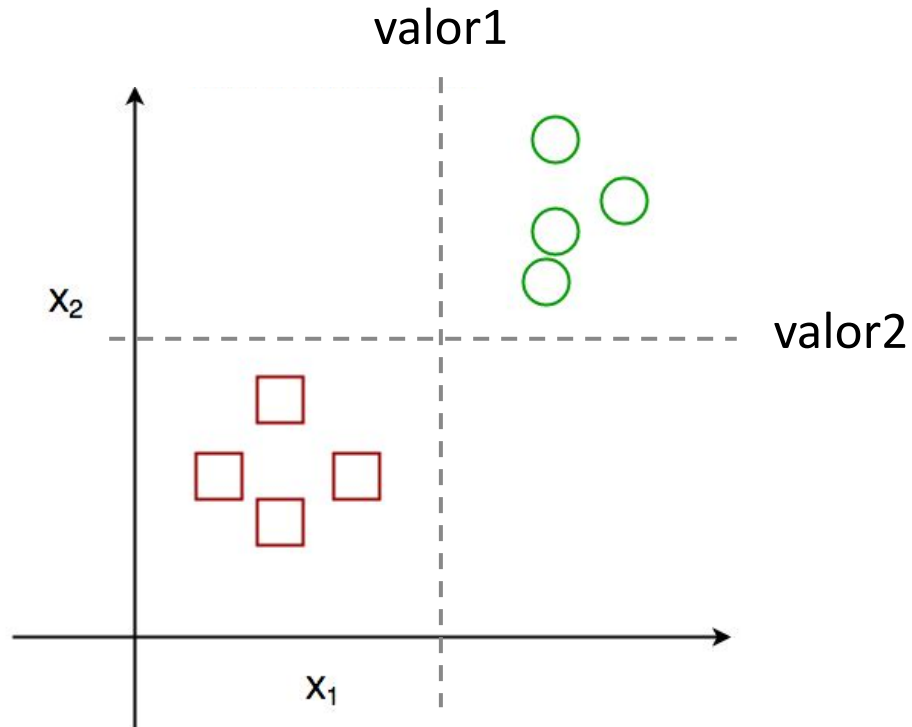
James et al 2017, Ch8



Mitchell 1997, Ch3

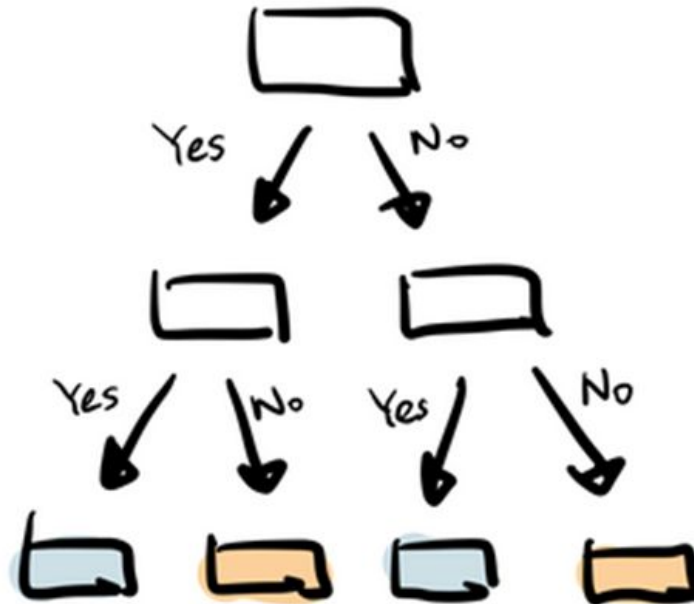
Árboles de decisión

Clasificación



Reglas if-then sobre valores de atributos.
Predicen el objetivo en función de esas reglas.

Árboles de decisión



Raíz: el nodo desde el cual inicia el árbol

Nodo: representa test sobre un atributo de la instancia

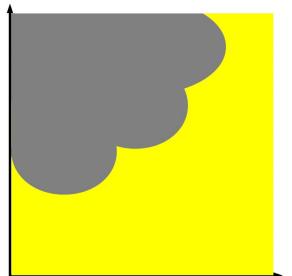
Rama desde un nodo: corresponde a un valor para ese atributo

Hojas: nodos que definen las clases de la decisión

Método de **inferencia inductiva** (busca aproximar una función objetivo).

El árbol representa **disyunción de conjunciones** sobre valores de atributos (y/o).

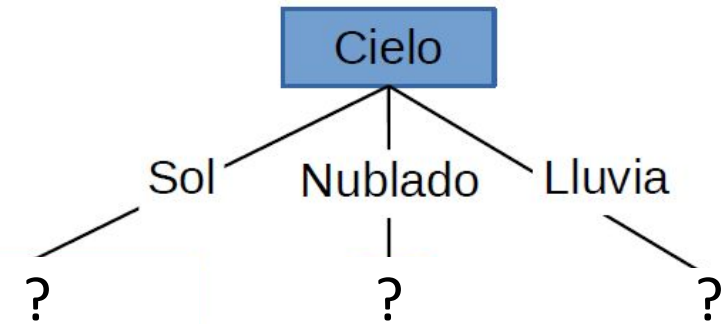
Aprende **reglas if-then** que **reducen localmente el error** con algún criterio.



Árboles de decisión

Cómo construyo un árbol?

Instancia	Atributos				Clase
	Cielo	Temperatura	Humedad	Viento	Va a correr?
1	sol	calor	alta	débil	No
2	sol	calor	alta	fuerte	No
3	nublado	calor	alta	débil	Sí
4	lluvia	templado	alta	débil	Sí
5	lluvia	frío	normal	débil	Sí
6	lluvia	frío	normal	fuerte	No
7	nublado	frío	normal	fuerte	Sí
8	sol	templado	alta	débil	No
9	sol	frío	normal	débil	Sí
10	lluvia	templado	normal	débil	Sí
11	sol	templado	normal	fuerte	Sí
12	nublado	templado	alta	fuerte	Sí
13	nublado	calor	normal	débil	Sí
14	lluvia	templado	alta	fuerte	No

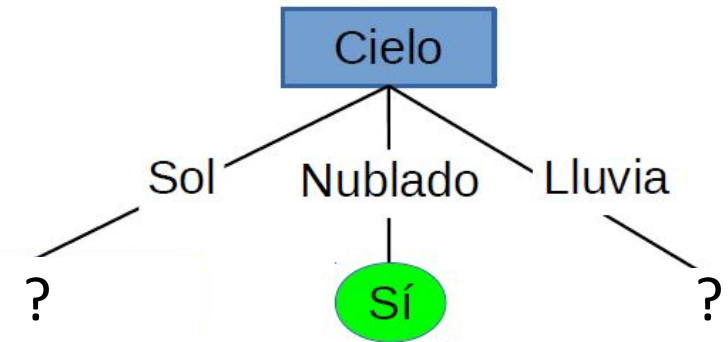


El atributo cielo, es un buen nodo para empezar el árbol, ya que cuando toma el valor “nublado” todas las instancias son de la clase “Sí”

Árboles de decisión

Cómo construyo un árbol?

Instancia	Atributos				Clase
	Cielo	Temperatura	Humedad	Viento	Va a correr?
1	sol	calor	alta	débil	No
2	sol	calor	alta	fuerte	No
3	nublado	calor	alta	débil	Sí
4	lluvia	templado	alta	débil	Sí
5	lluvia	frío	normal	débil	Sí
6	lluvia	frío	normal	fuerte	No
7	nublado	frío	normal	fuerte	Sí
8	sol	templado	alta	débil	No
9	sol	frío	normal	débil	Sí
10	lluvia	templado	normal	débil	Sí
11	sol	templado	normal	fuerte	Sí
12	nublado	templado	alta	fuerte	Sí
13	nublado	calor	normal	débil	Sí
14	lluvia	templado	alta	fuerte	No

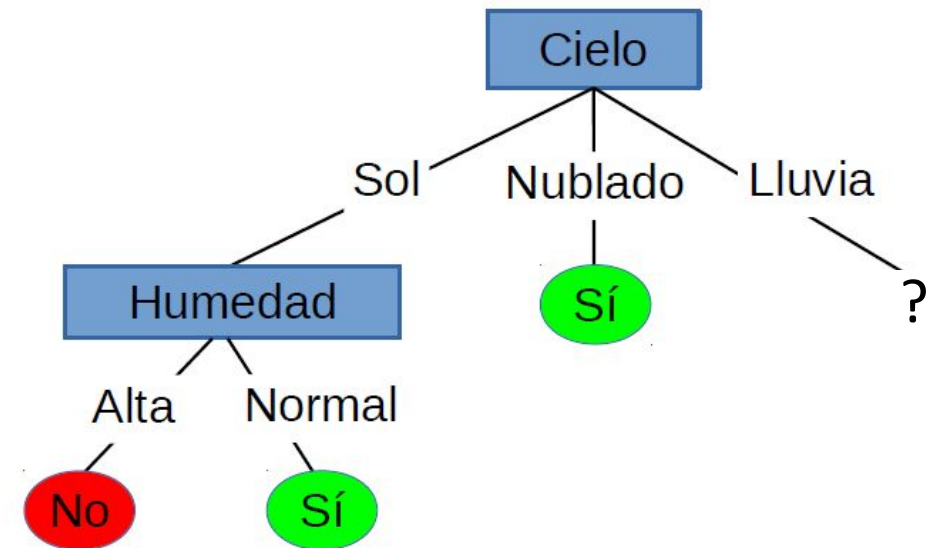


Cuando el valor que toma el atributo cielo es “sol”, algunas instancias son “Sí” y otras “No”. Tengo que buscar si existe algún atributo que me separe bien. **Spoiler:** humedad, que cuando es “alta” es “No” y si es “normal” es “Sí”

Árboles de decisión

Cómo construyo un árbol?

Instancia	Atributos				Clase
	Cielo	Temperatura	Humedad	Viento	Va a correr?
1	sol	calor	alta	débil	No
2	sol	calor	alta	fuerte	No
3	nublado	calor	alta	débil	Sí
4	lluvia	templado	alta	débil	Sí
5	lluvia	frío	normal	débil	Sí
6	lluvia	frío	normal	fuerte	No
7	nublado	frío	normal	fuerte	Sí
8	sol	templado	alta	débil	No
9	sol	frío	normal	débil	Sí
10	lluvia	templado	normal	débil	Sí
11	sol	templado	normal	fuerte	Sí
12	nublado	templado	alta	fuerte	Sí
13	nublado	calor	normal	débil	Sí
14	lluvia	templado	alta	fuerte	No

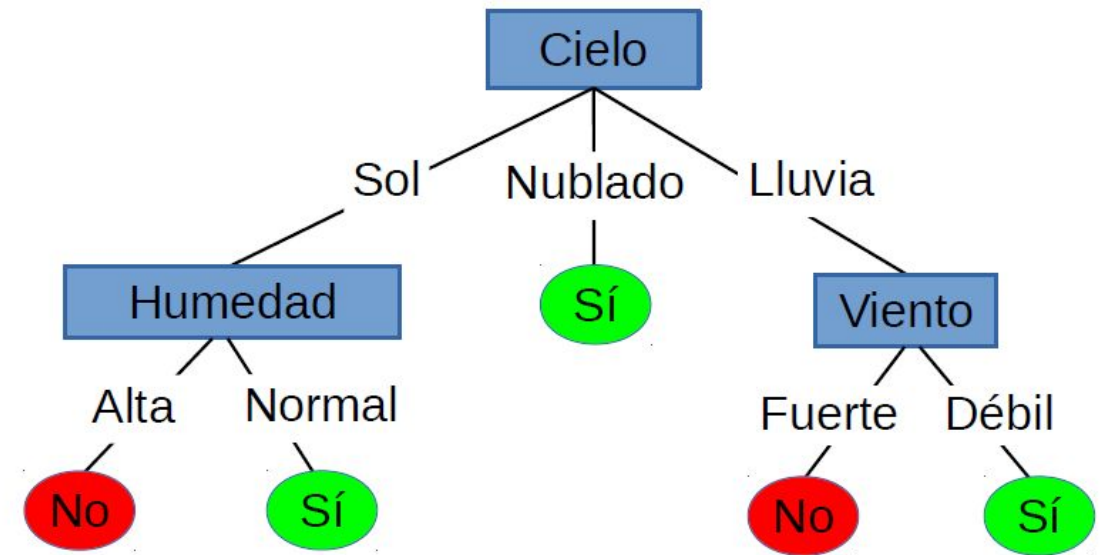


Busco otro atributo que me separe bien las instancias de cielo “lluvia”. **Spoiler:** es viento, que cuando es “fuerte” es “No” y cuando es “débil” es “Sí”

Árboles de decisión

Cómo construyo un árbol?

Instancia	Atributos				Clase
	Cielo	Temperatura	Humedad	Viento	Va a correr?
1	sol	calor	alta	débil	No
2	sol	calor	alta	fuerte	No
3	nublado	calor	alta	débil	Sí
4	lluvia	templado	alta	débil	Sí
5	lluvia	frío	normal	débil	Sí
6	lluvia	frío	normal	fuerte	No
7	nublado	frío	normal	fuerte	Sí
8	sol	templado	alta	débil	No
9	sol	frío	normal	débil	Sí
10	lluvia	templado	normal	débil	Sí
11	sol	templado	normal	fuerte	Sí
12	nublado	templado	alta	fuerte	Sí
13	nublado	calor	normal	débil	Sí
14	lluvia	templado	alta	fuerte	No



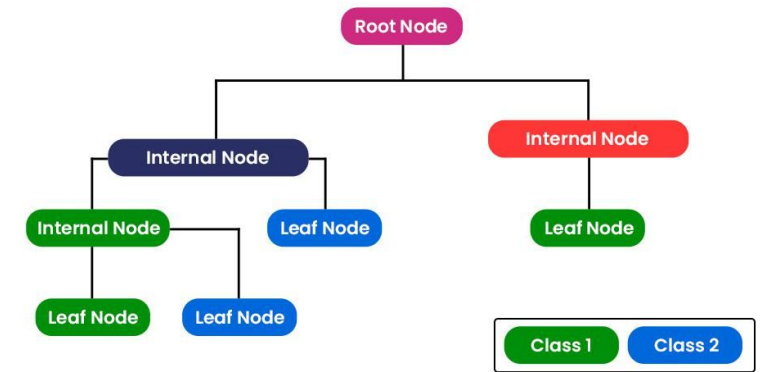
Cada **nodo** interno evalúa un atributo discreto X_i . Cada **rama** corresponde a un valor para ese atributo X_i . Cada **hoja** predice un valor de Y

Árboles de decisión

Cómo construyo un árbol?

Inducción Top-Down (CART; ID3; C4.5)

1. Encontrar X_i el “mejor” atributo para `nodo_actual`
2. Asignar X_i como atributo de decisión del `nodo_actual`
3. Para cada valor de X_i , crear un nuevo nodo hijo del `nodo_actual`
4. Clasificar (repartir) las instancias en los nuevos nodos, según el valor de X_i
5. Si las instancias están bien clasificadas: TERMINAR
Si no: Iterar sobre los nuevos nodos



Empezamos con 14 instancias: $[9\oplus, 5\ominus]$

¿Cuál es el
“mejor”
atributo?



Árboles de decisión

Cómo construyo un árbol?

En clasificación guiamos la construcción mediante **medidas de impureza**, la cual buscamos minimizar.

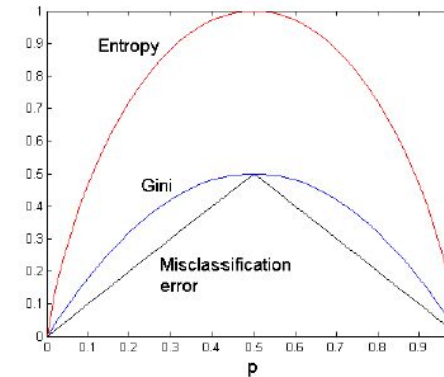


Gini gain

The *Gini index* is defined by

$$G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Frecuencia de etiquetar mal al azar



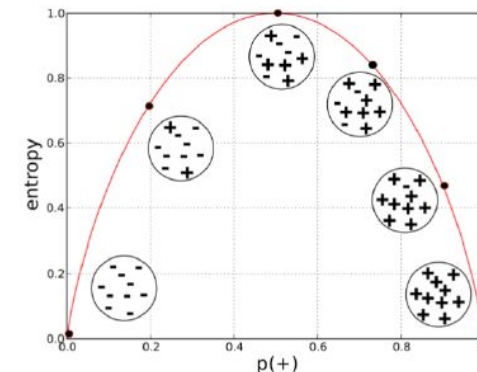
La ganancia es la reducción de impureza

Info gain

An alternative to the Gini index is *cross-entropy*, given by

$$D = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}.$$

Desorden de etiquetas en c/ estado



Quiero nodos puros (hojas)

Otra: Gain Ratio (corrige preferencia de Info Gain por atributos con demasiados valores)

Criterios de impureza

Medida de impureza: Gini gain (CART*)

- Algoritmo top-down sin retrocesos (empieza desde arriba)
- Algoritmo greedy (hace split por una decisión local, no global)
- Hace división binaria recursiva (sólo 2 posibles cortes)
- Usa la ganancia de Gini (descenso del Gini Index luego de una partición) para seleccionar el mejor atributo

Gini index:

$$I_G(p) = \sum_{i=1}^J p_i (1 - p_i) = 1 - \sum_{i=1}^J p_i^2 \quad p_i = \text{prop. de clase } i.$$

Se calcula el índice de Gini inicial, luego el que suma cada partición de cada atributo. Se selecciona la partición que más reduce (mayor ganancia)

- Aplica métodos para limitar la profundidad del árbol

* L. Breiman et al, "Classification and Regression Trees", Taylor & Francis, 1984

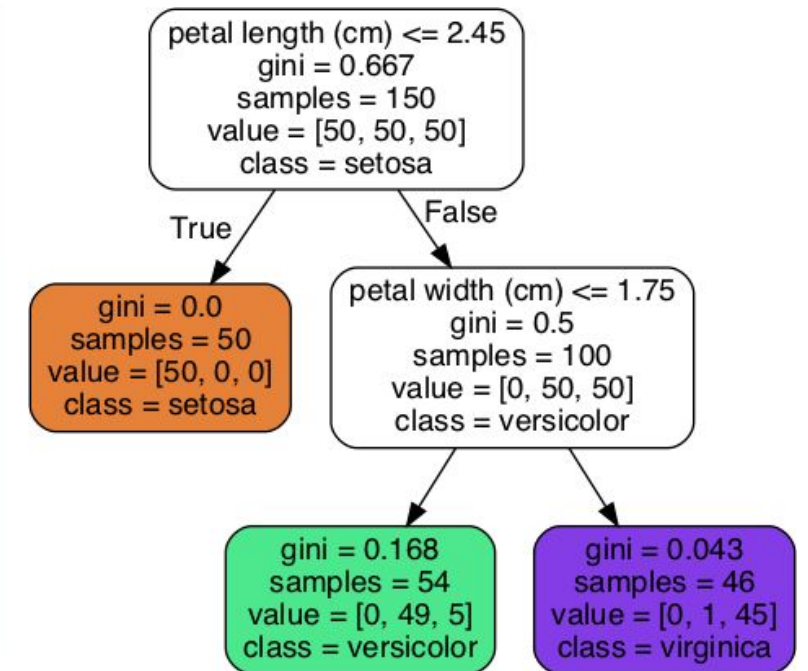
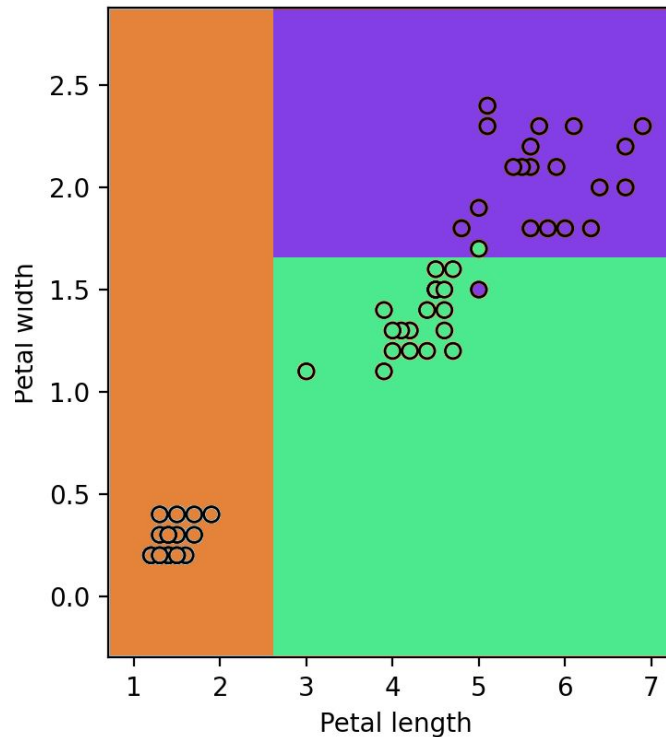
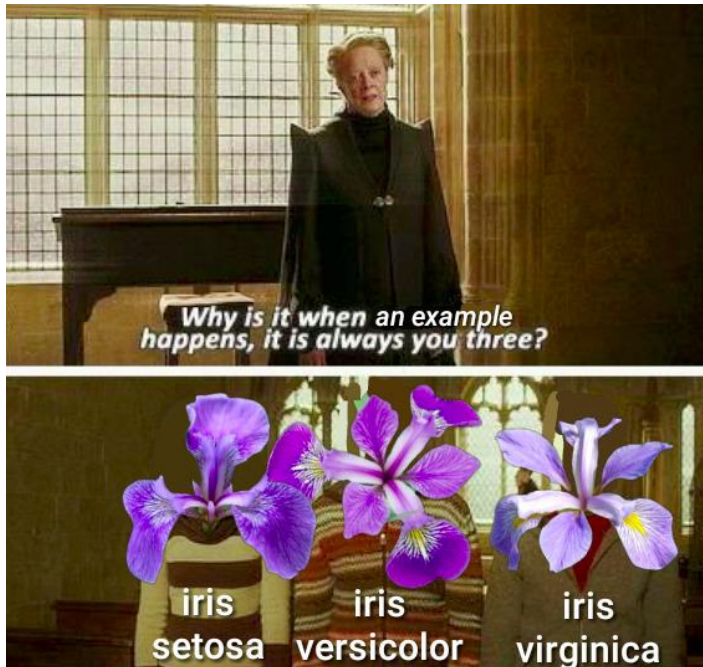
Crterios de impureza

Medida de impureza: Gini gain (CART)

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2$$

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

Ej: Iris Dataset



Géron 2019

Para un nodo, elijo el atributo que más reduce la impureza de Gini.

Para buscar el umbral a tomar, también puedo usar Gini Index sobre los valores del atributo ordenados.

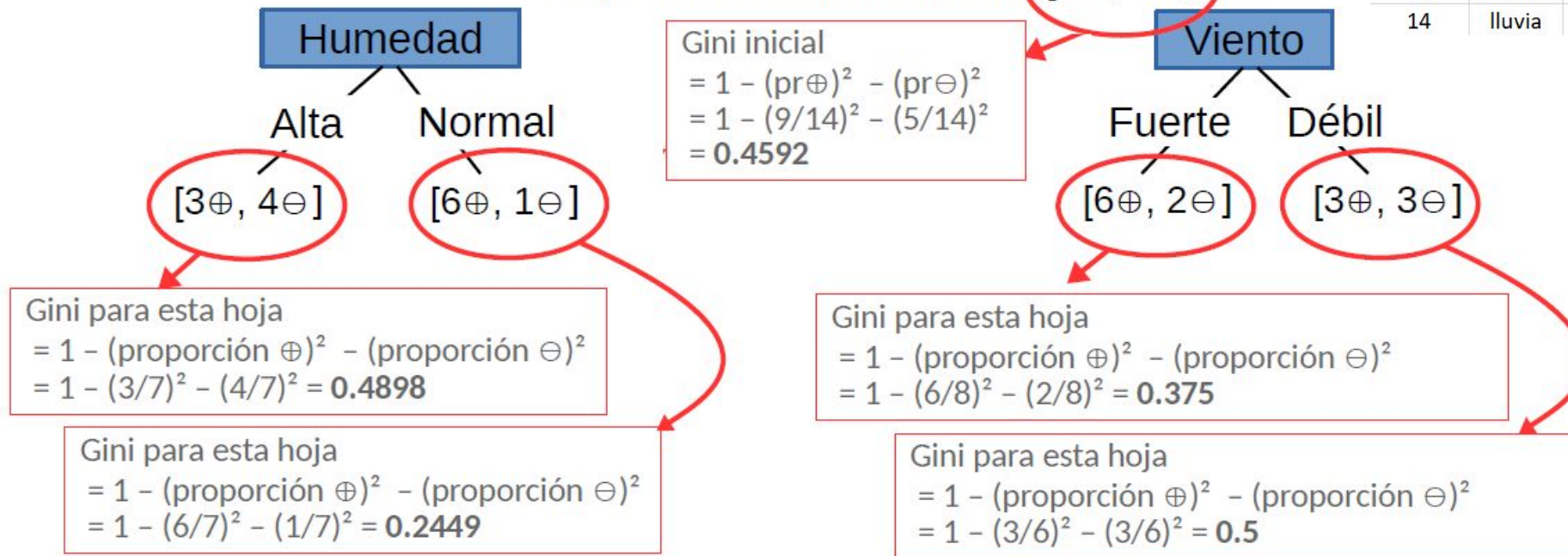
Criterios de impureza

Medida de impureza: Gini gain (CART)

$$G_i = 1 - \sum_{k=1}^n p_{i,k}^2 \quad J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

Instancia	Atributos				Clase
	Cielo	Temperatura	Humedad	Viento	Va a correr?
1	sol	calor	alta	débil	No
2	sol	calor	alta	fuerte	No
3	nublado	calor	alta	débil	Sí
4	lluvia	templado	alta	débil	Sí
5	lluvia	frío	normal	débil	Sí
6	lluvia	frío	normal	fuerte	No
7	nublado	frío	normal	fuerte	Sí
8	sol	templado	alta	débil	No
9	sol	frío	normal	débil	Sí
10	lluvia	templado	normal	débil	Sí
11	sol	templado	normal	fuerte	Sí
12	nublado	templado	alta	fuerte	Sí
13	nublado	calor	normal	débil	Sí
14	lluvia	templado	alta	fuerte	No

Empezamos con 14 instancias: $[9\oplus, 5\ominus]$



Gini de Humedad: Promedio ponderado del Gini de las hojas = $(7/14) 0.4898 + (7/14) 0.2449 = \mathbf{0.3674}$

Gini de Viento: Promedio ponderado del Gini de las hojas = $(8/14) 0.375 + (6/14) 0.5 = \mathbf{0.4286}$

El atributo que más reduce la impureza de Gini es humedad

Criterios de impureza

Medida de impureza: Info gain (ID3* o C4.5**)

- Algoritmo top-down sin retrocesos (empieza desde arriba)
- Algoritmo greedy (hace split por una decisión local, no global)
- Admite división no binaria
- Se usa la ganancia de información (descenso de la Entropía luego de una partición) para seleccionar el mejor atributo

Entropía:

$$Entropy = - \sum_{i=1}^n p_i \log_2(p_i) \quad p_i = \text{prop. de clase } i.$$

Se calcula la entropía inicial, luego la que suma cada partición de cada atributo.
Se selecciona la partición que más reduce (mayor ganancia)










- Aplica métodos para limitar la profundidad del árbol (diferentes a CART)

* J.J.R. Quinlan, “Induction of Decision Trees”, Machine Learning, 1(1):81-106, 1986.

** J.R. Quinlan, “Simplifying Decision Trees”, Intl. Journal of Human-Computer Studies, 51(2):497–510, 1999

Crterios de impureza

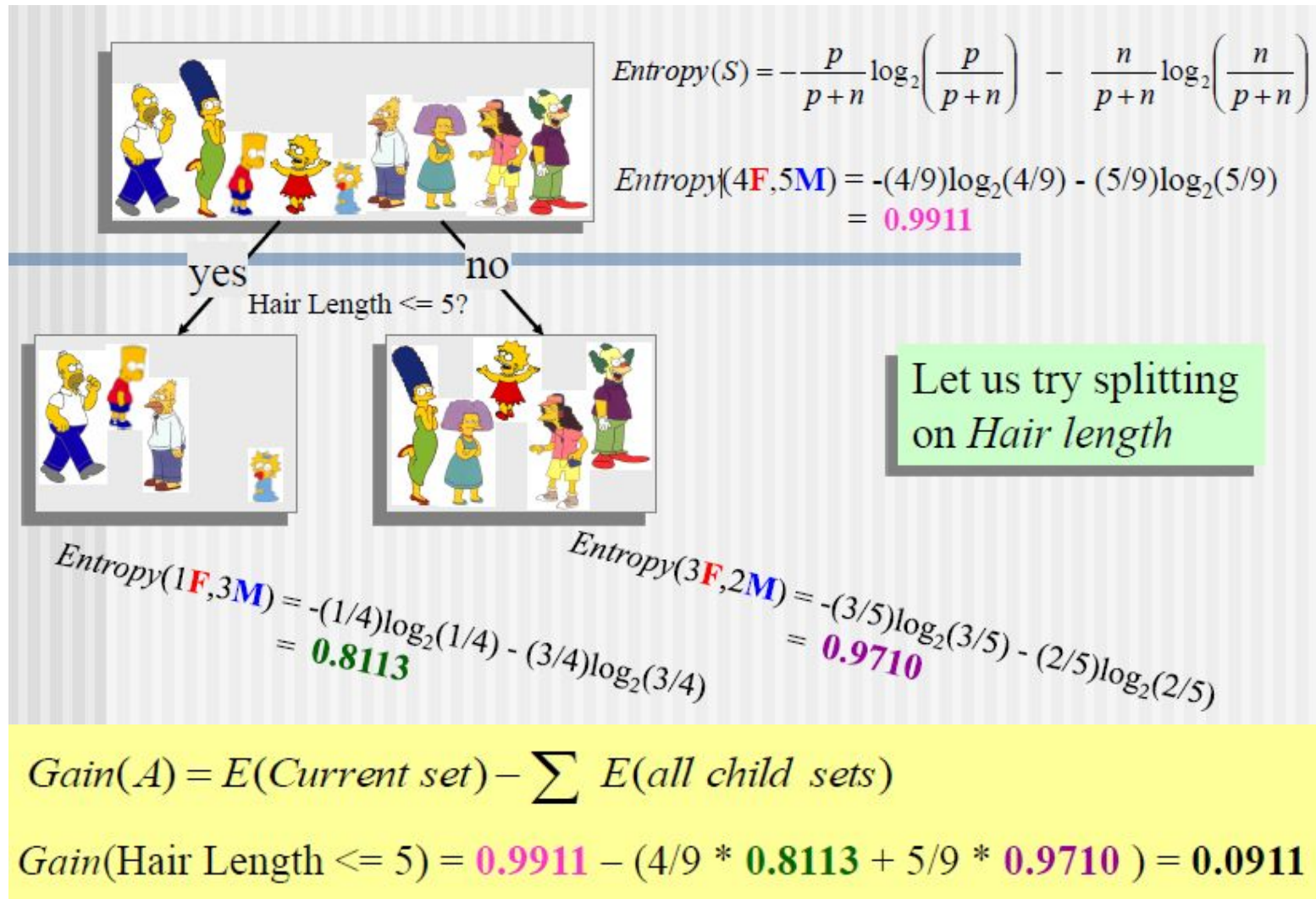
Medida de impureza: Info gain

Person	Hair Length	Weight	Age	Class
 Homer	0"	250	36	M
 Marge	10"	150	34	F
 Bart	2"	90	10	M
 Lisa	6"	78	8	F
 Maggie	4"	20	1	F
 Abe	1"	170	70	M
 Selma	8"	160	41	F
 Otto	10"	180	38	M
 Krusty	6"	200	45	M

	Comic	8"	290	38	?
---	-------	----	-----	----	----------

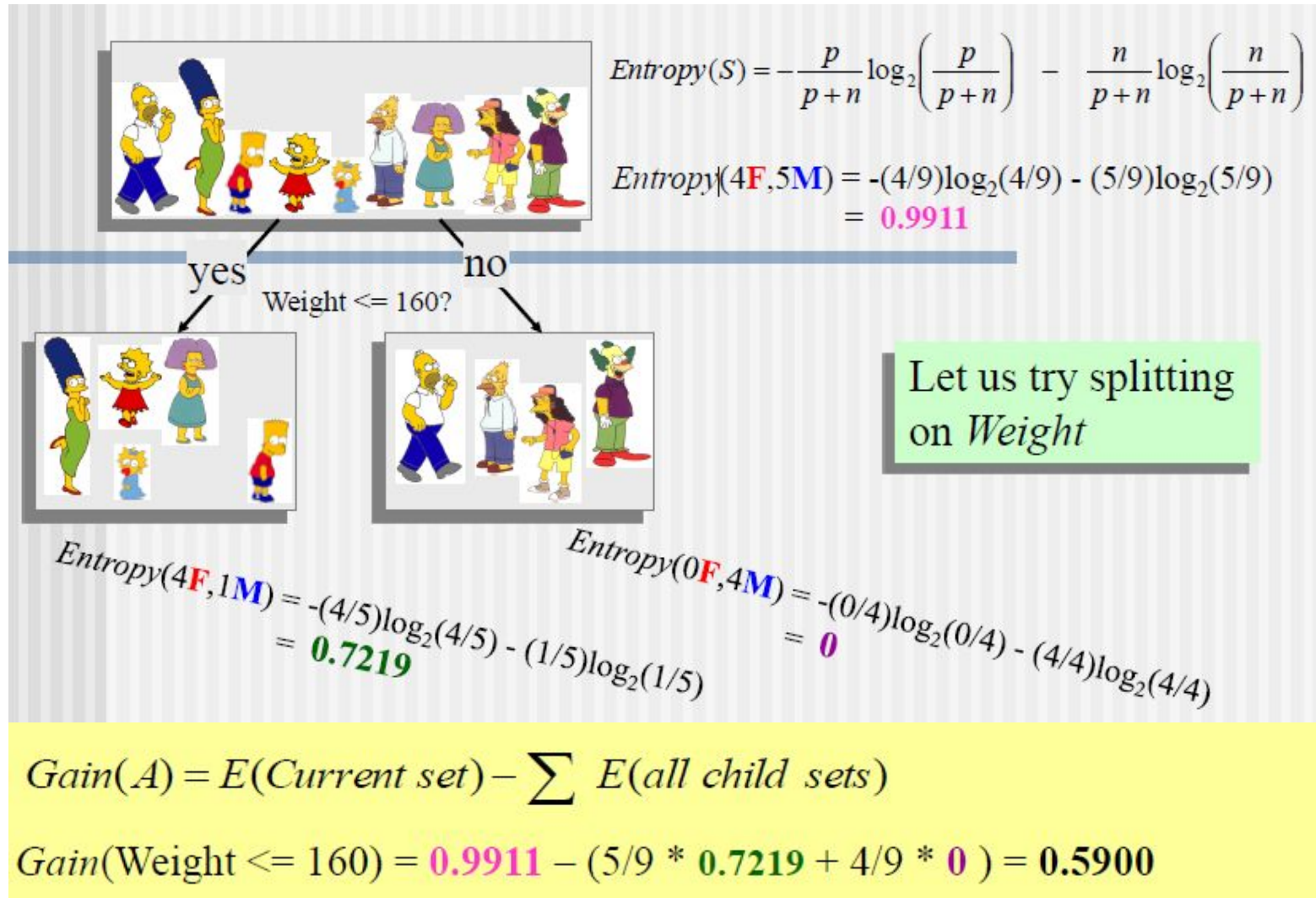
Crterios de impureza

Medida de impureza: Info gain



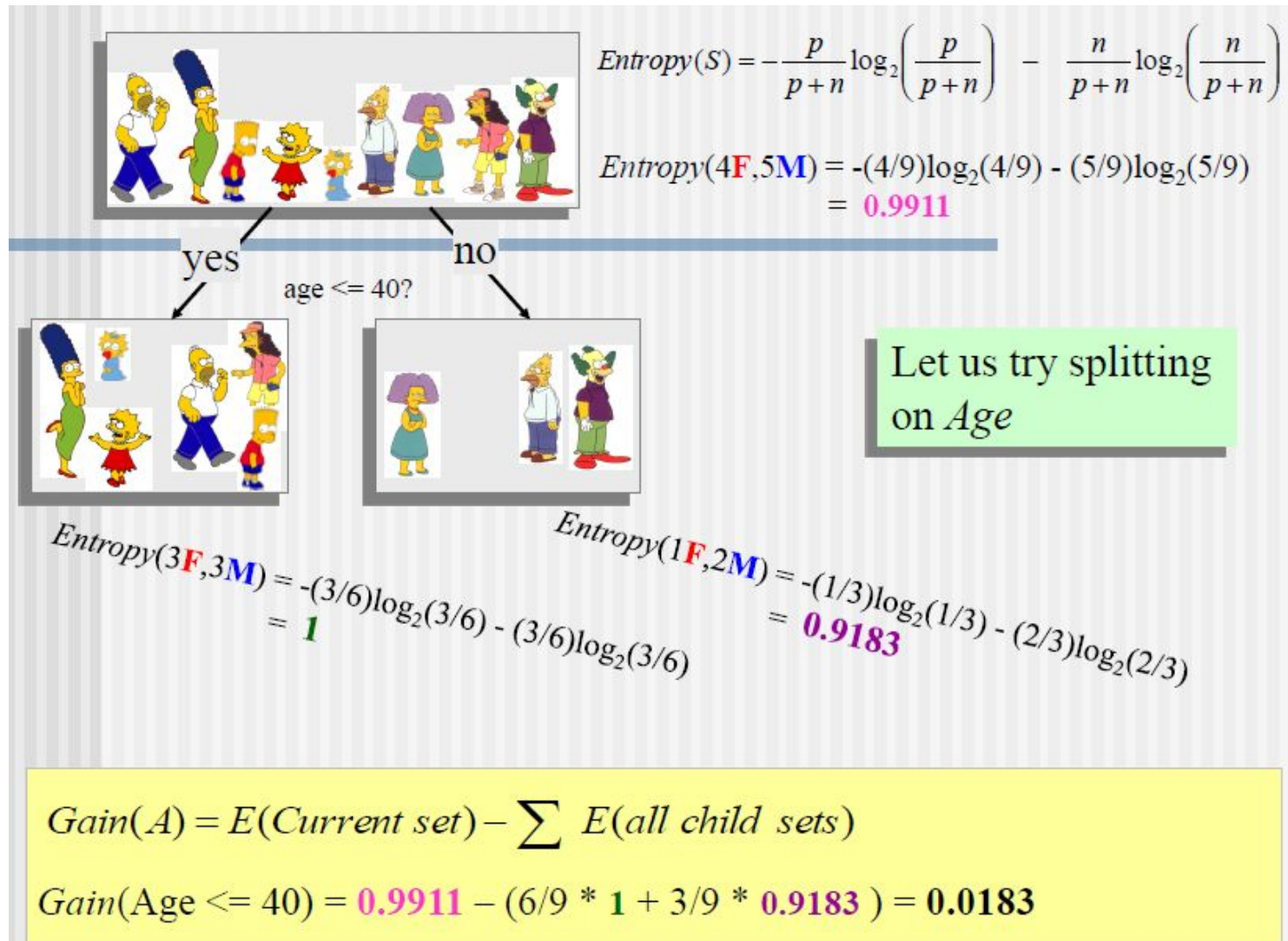
Crterios de impureza

Medida de impureza: Info gain



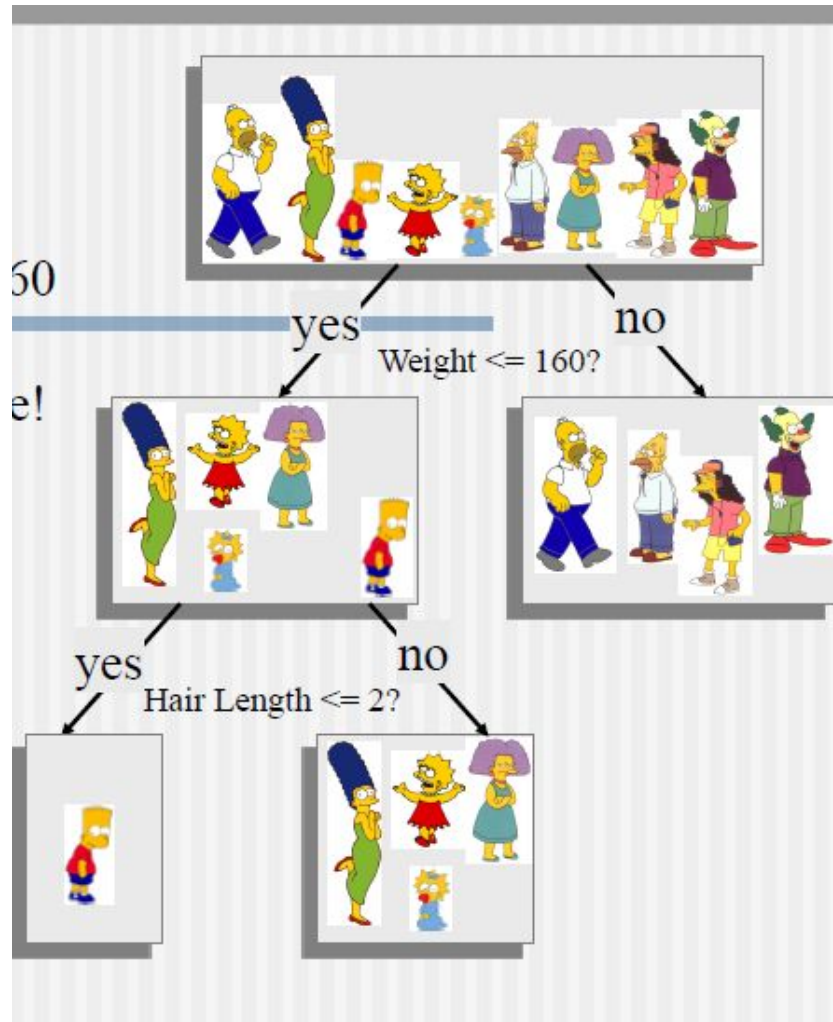
Crterios de impureza

Medida de impureza: Info gain



Criterios de impureza

Medida de impureza: Info gain



De los 3 atributos que teníamos, Weight era el mejor porque reduce más la impureza medida con entropía y entonces tiene mayor ganancia

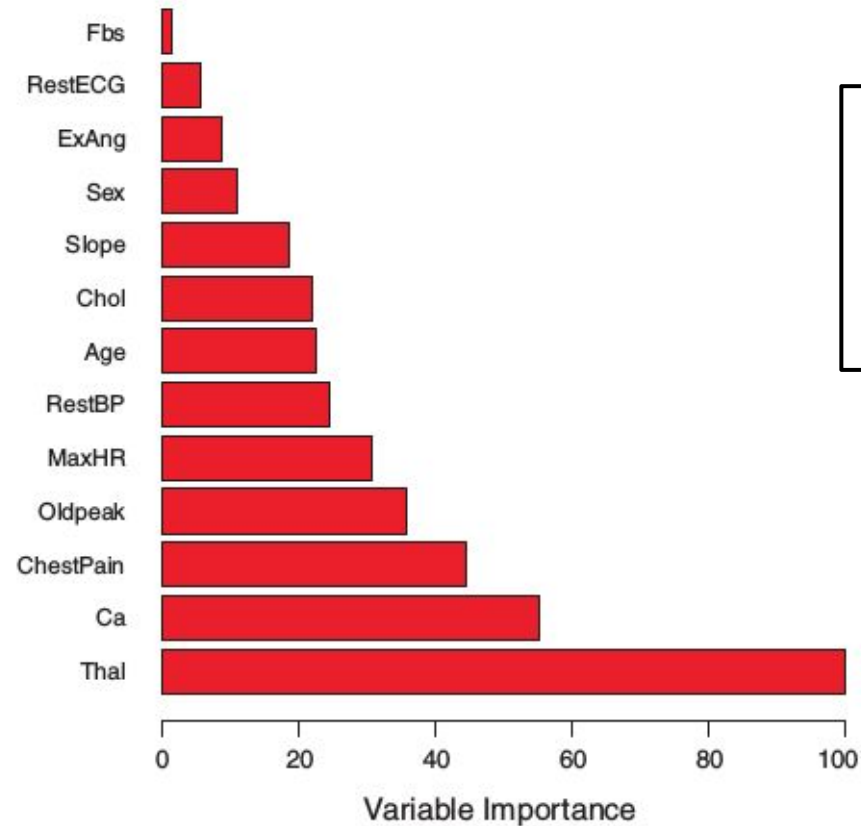
Pero mientras que las personas que pesan más de 160 están perfectamente clasificadas (como hombres), las personas que pesan menos de 160 no están perfectamente clasificadas...

Otra vuelta recursiva

Esta vez encontramos que podemos dividir el largo del cabello, ¡y listo!

Medidas de impureza

Se pueden usar para definir importancia de variables (atributos)

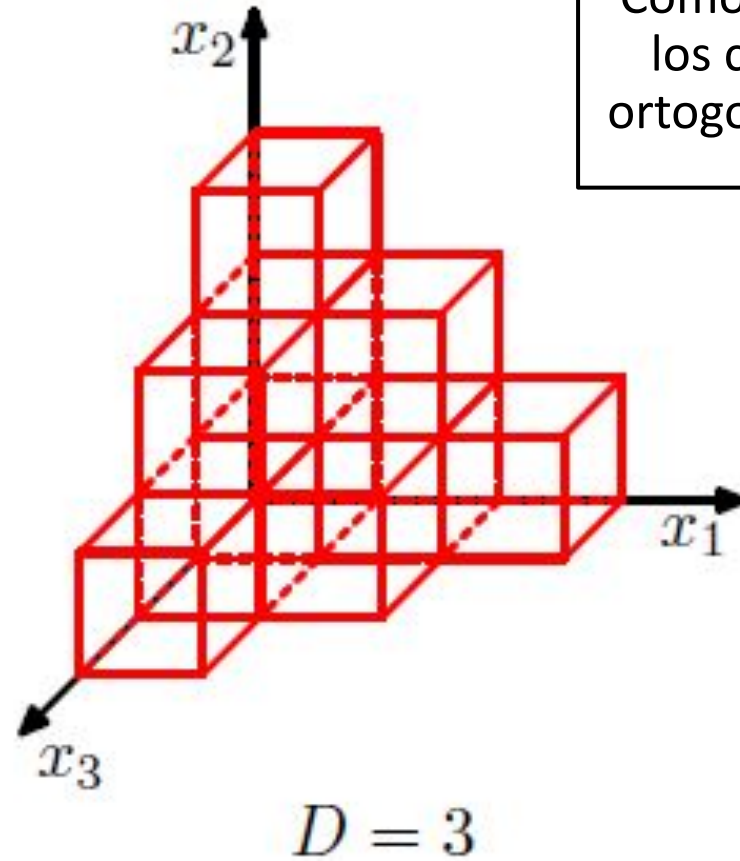
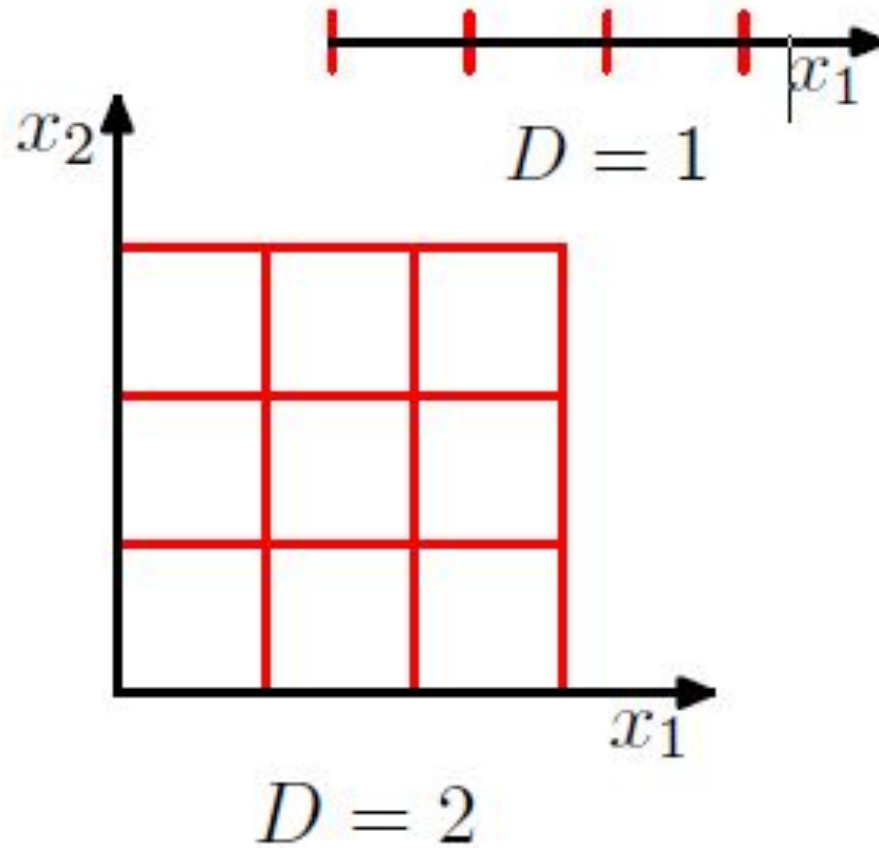


El atributo que más reduce la impureza de Gini, es el que considero como más importante

FIGURE 8.9. A variable importance plot for the **Heart** data. Variable importance is computed using the mean decrease in Gini index, and expressed relative to the maximum.

Contornos de decisión

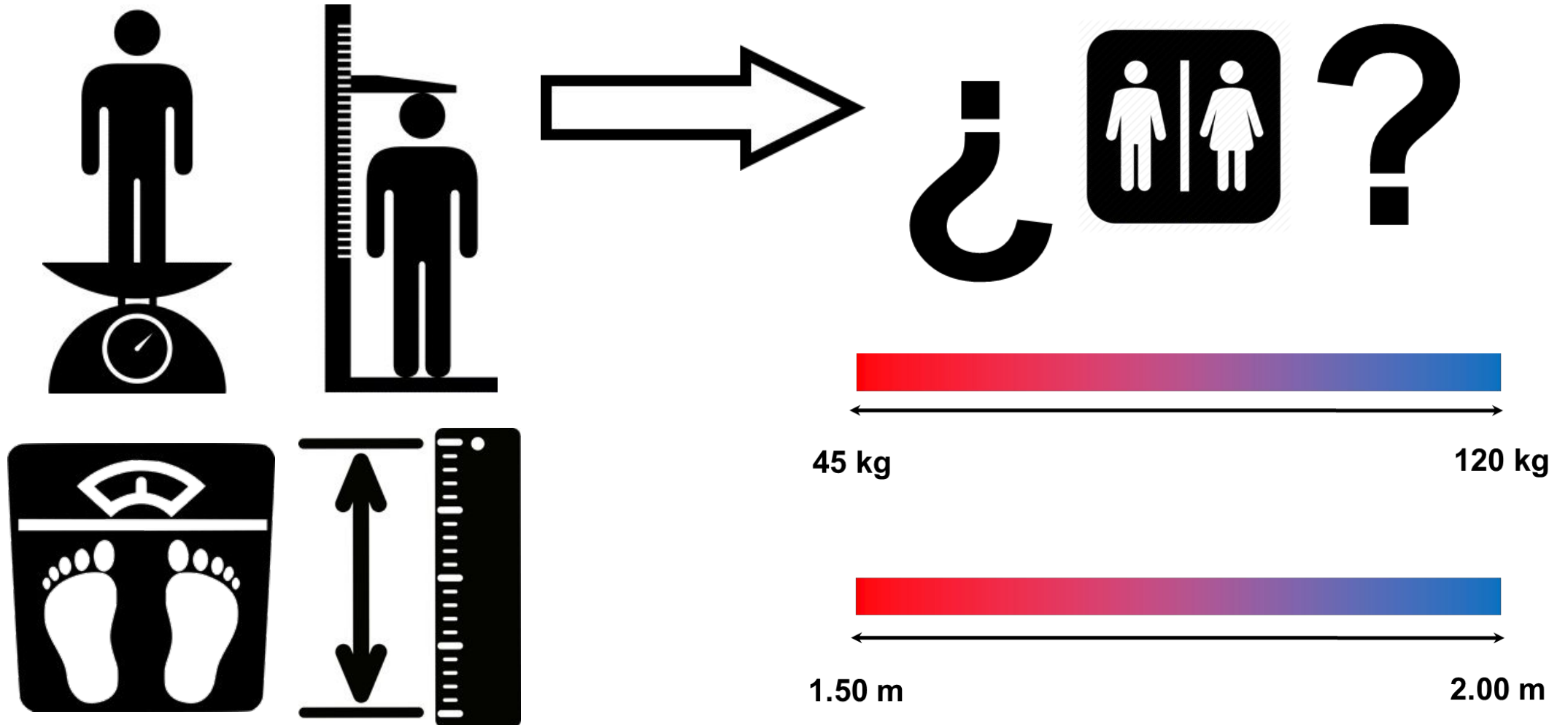
Las reglas if-then me definen contornos de decisión según la dimensión



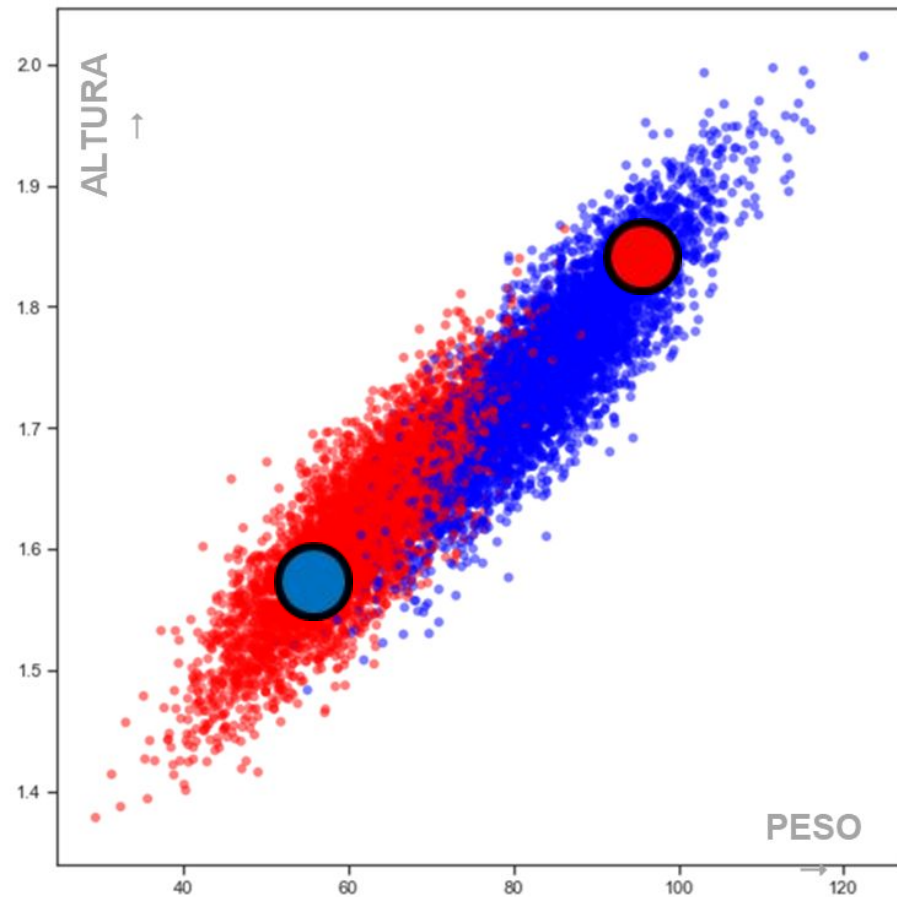
Como uso umbrales,
los contornos son
ortogonales a los ejes

Contornos de decisión

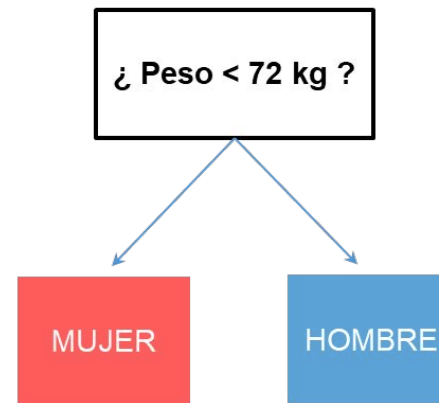
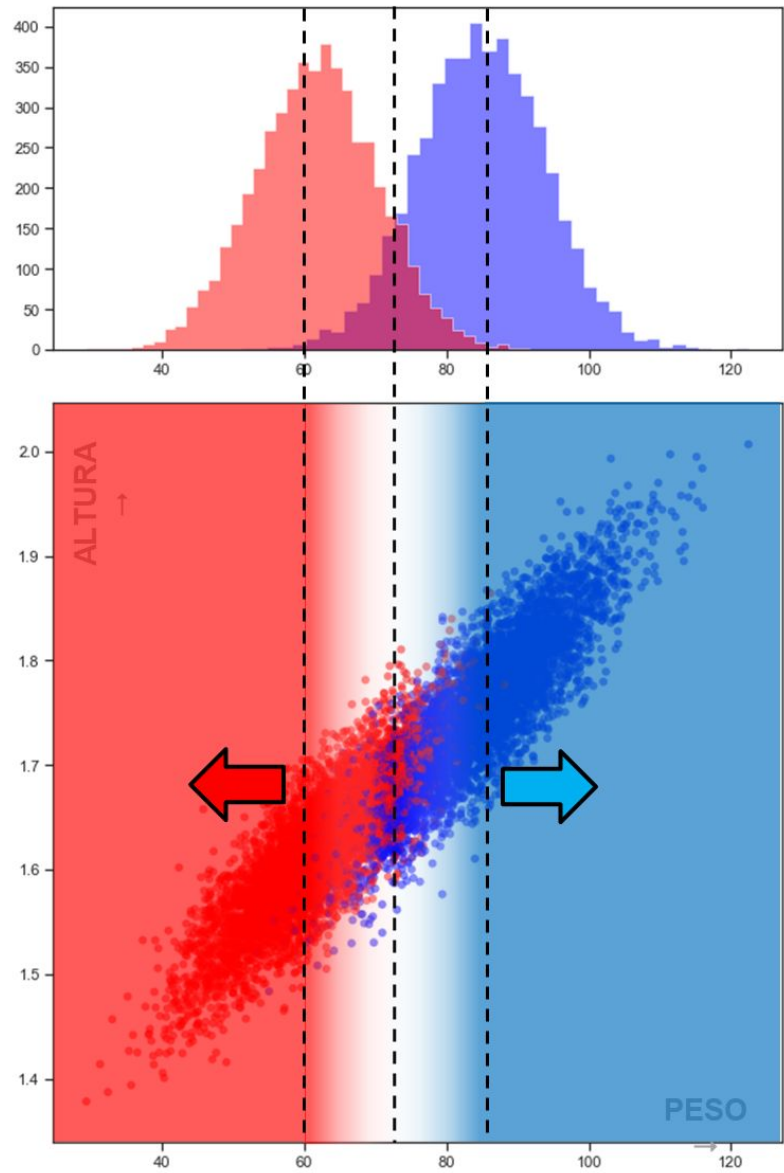
Veámoslo con un ejemplo de sexo biológico entre F y M según altura y peso



Contornos de decisión



Contornos de decisión



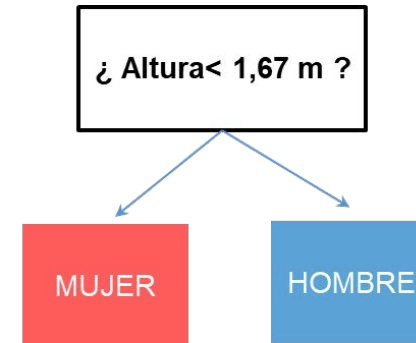
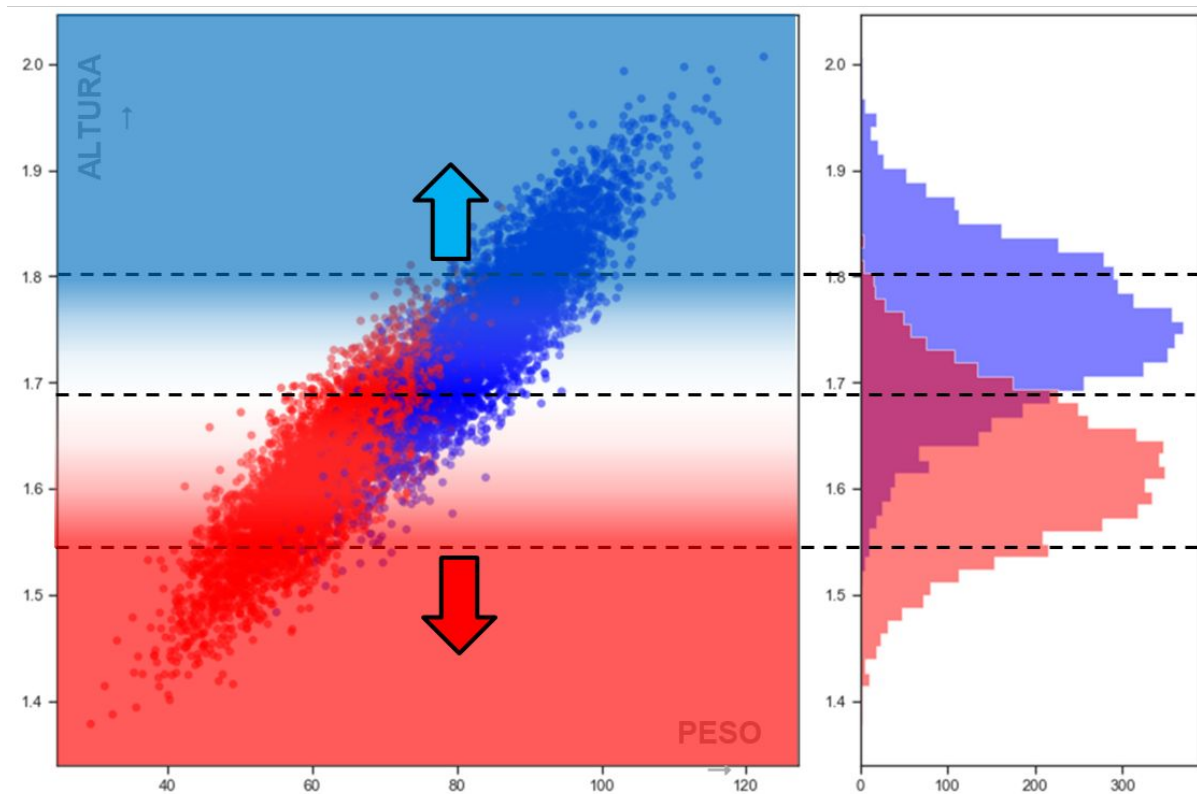
Puedo definir un umbral para el atributo peso

Aunque también puedo definir umbrales distintos para el atributo peso según la clase

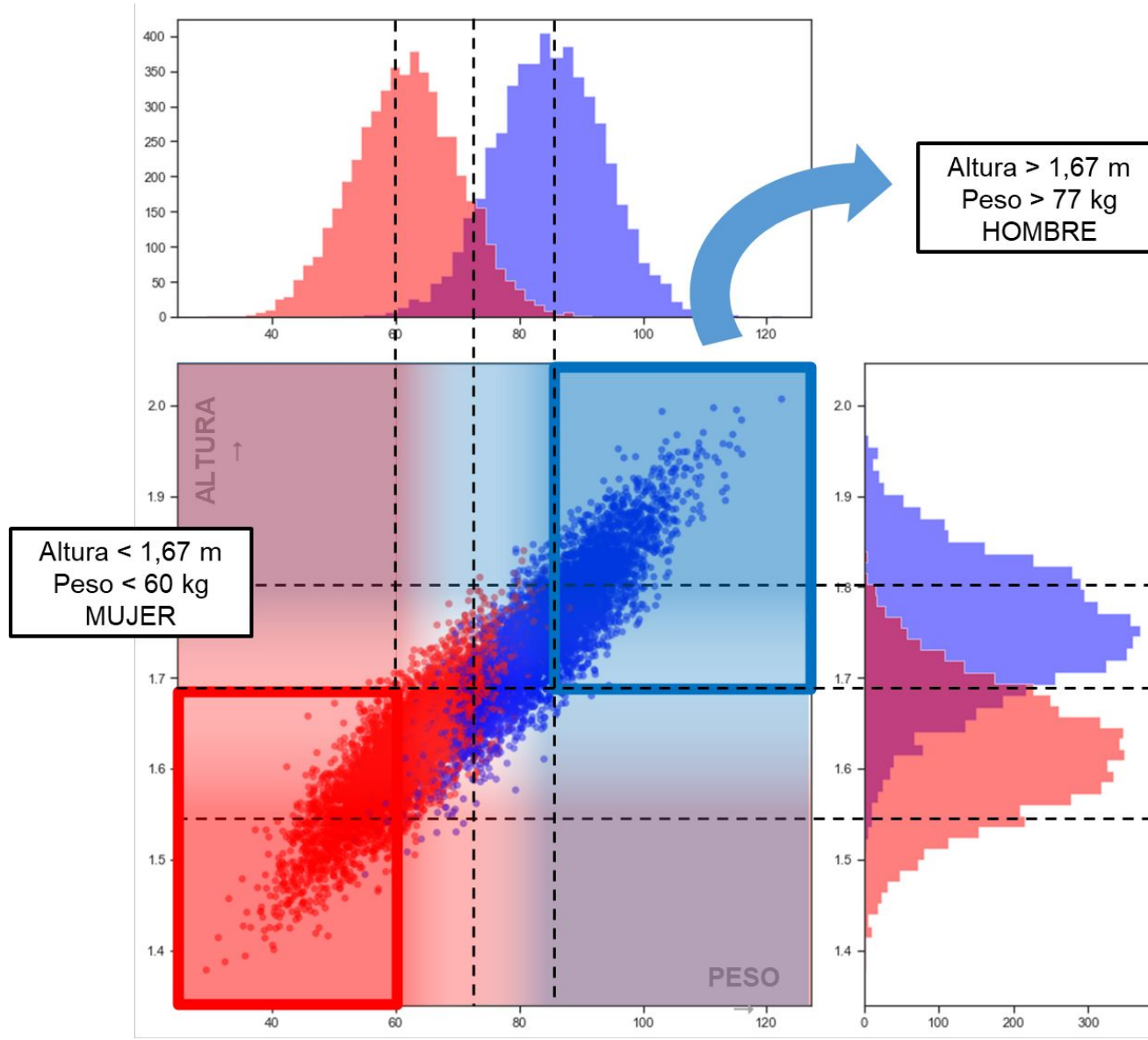
Contornos de decisión

Puedo definir un umbral
para el atributo altura

Aunque también puedo definir
umbrales distintos para el
atributo altura según la clase



Contornos de decisión

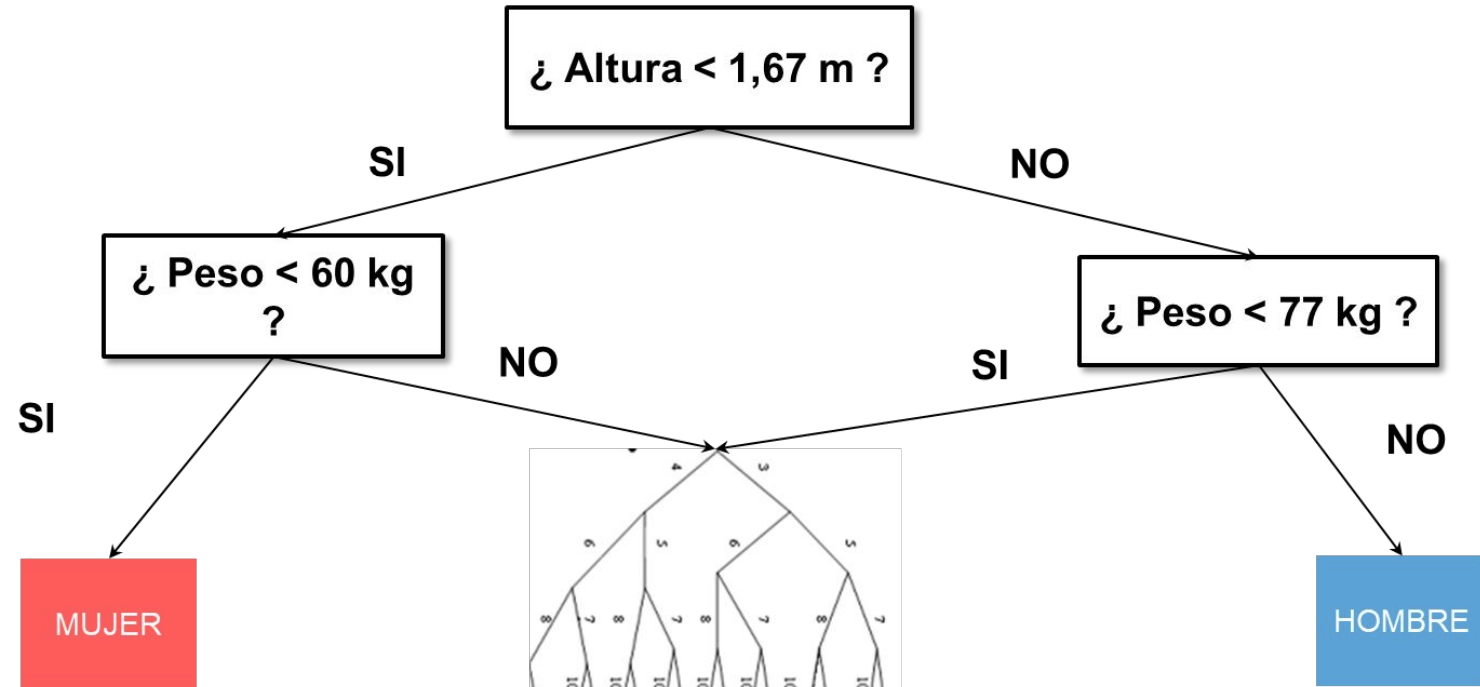


Puedo definir umbrales para ambos atributos, y según la clase

Se definen contornos de decisión rectangulares

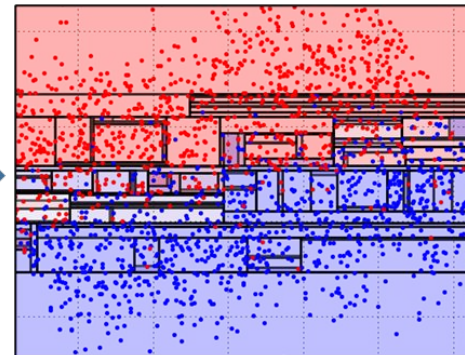
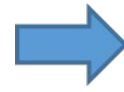
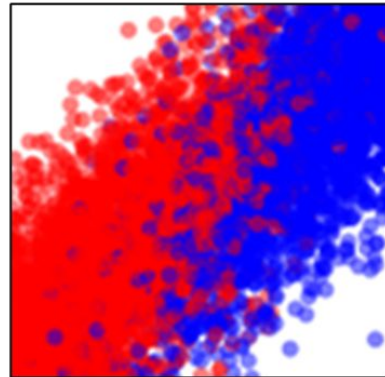
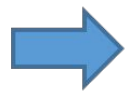
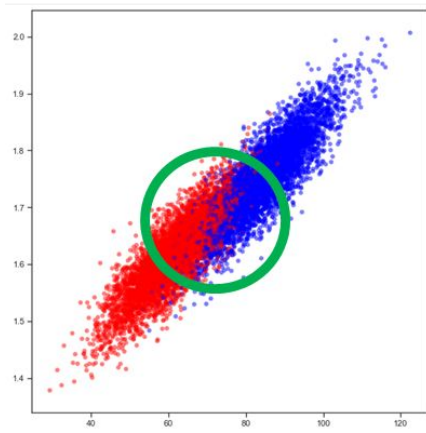
Qué pasa en las regiones intermedias?

Contornos de decisión

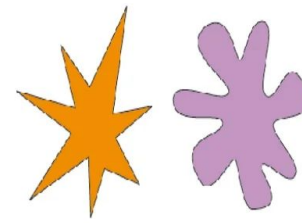


En la región intermedia
puedo construir ramas
del árbol que me
separen mejor los datos

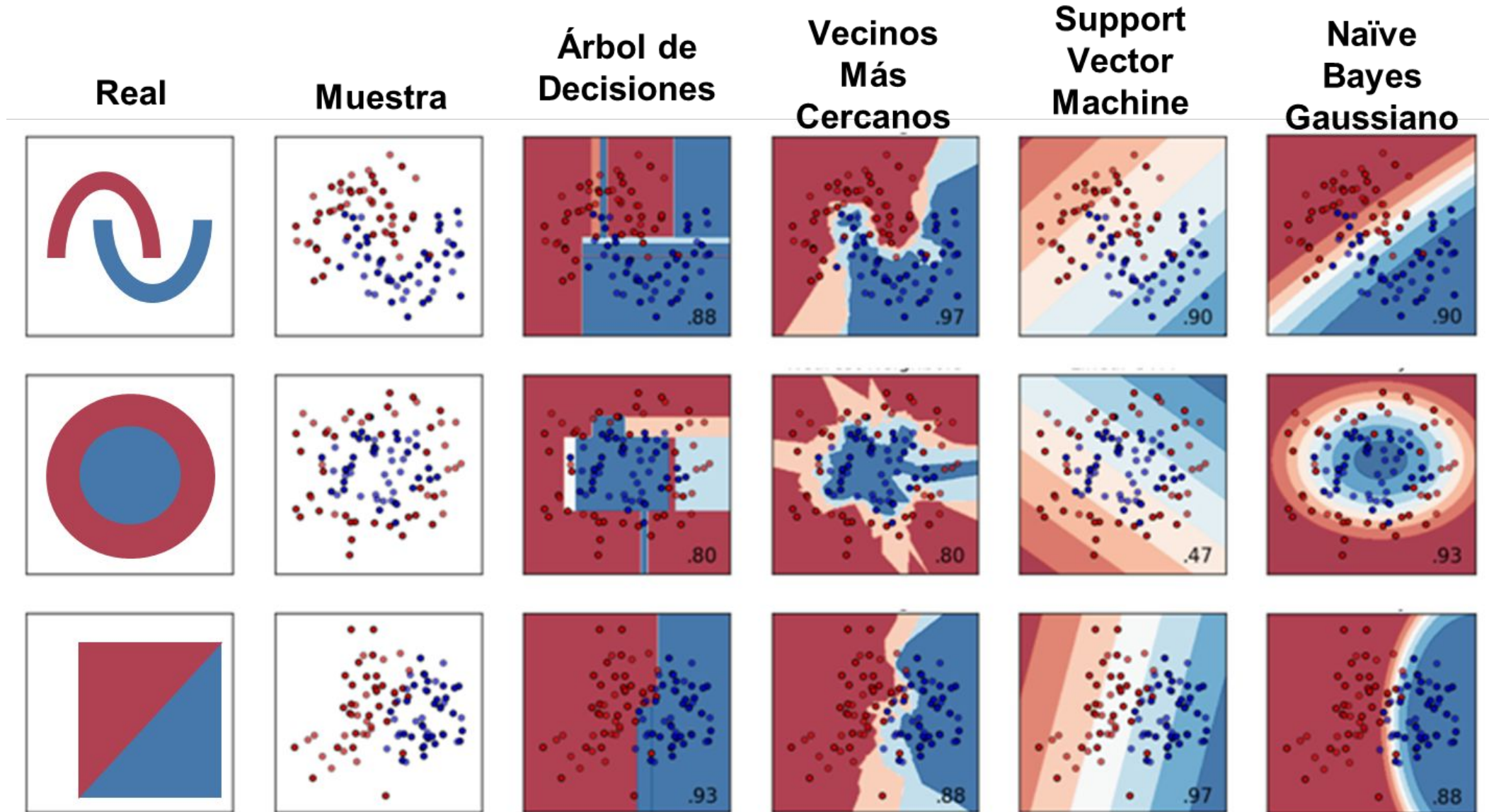
Pueden ser muchas
ramas y hacer del árbol
muy complejo



Contornos de decisión

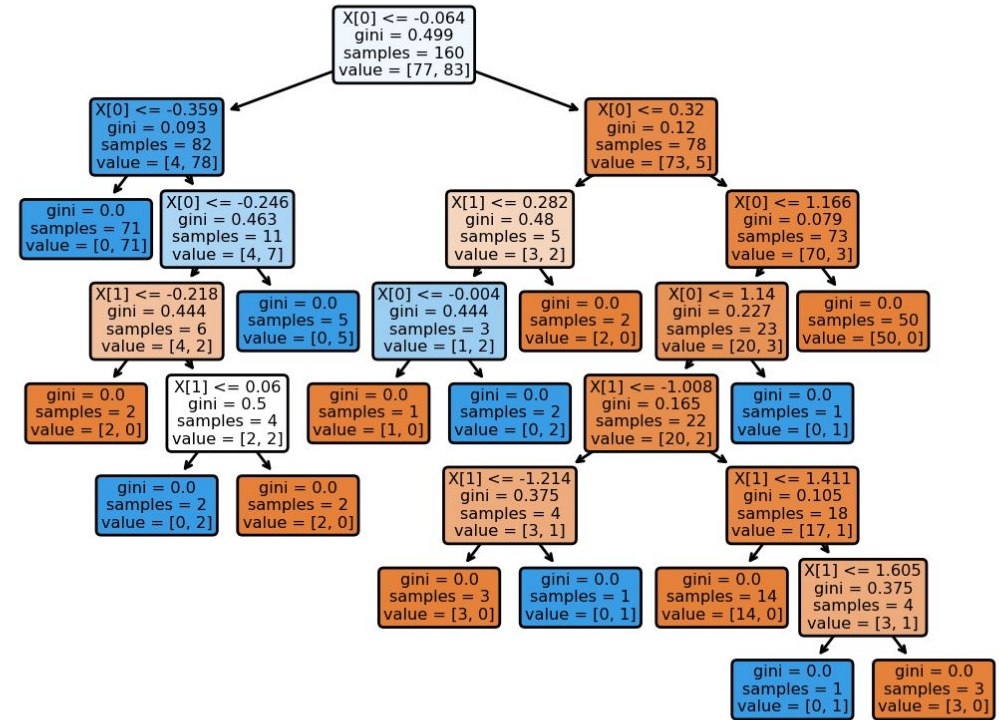
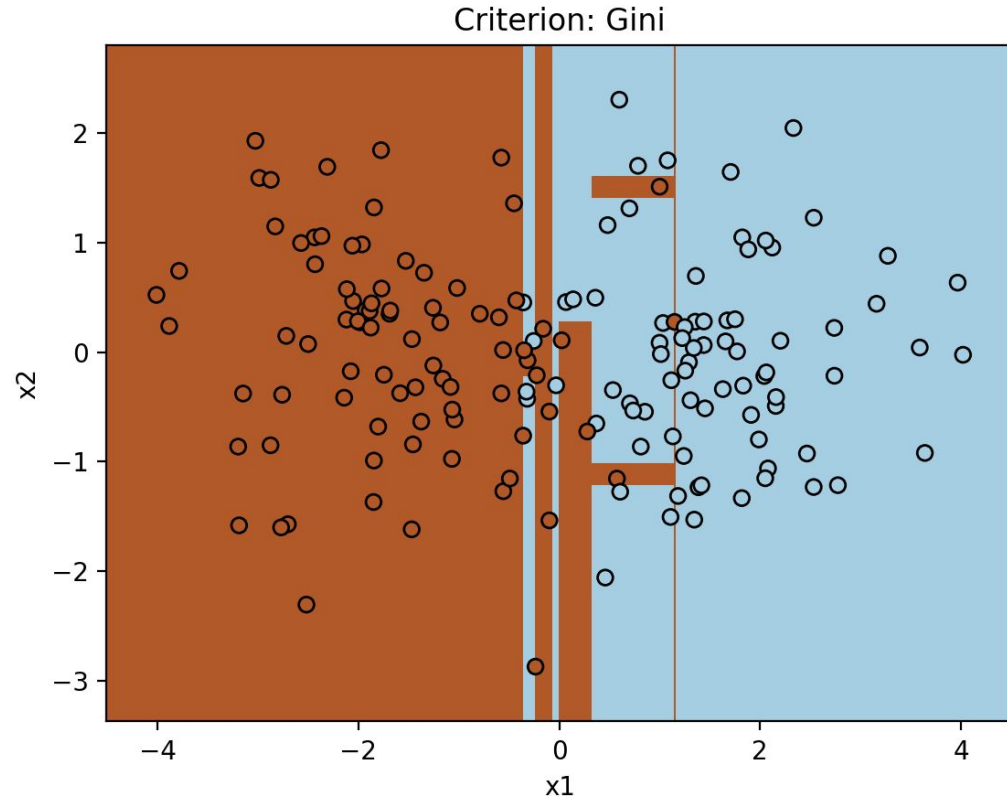


¿Quién es Buba y quién es Kiki?



Complejidad del árbol

Árboles muy complejos tienden a sobre-ajustar a los datos



Criterios de parada para evitar construir árboles muy complejos, o “poda” de árboles complejos para reducir el número de nodos a posteriori (pruning)

Complejidad del árbol

Hiperparámetros de árboles

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)
```

[\[source\]](#)

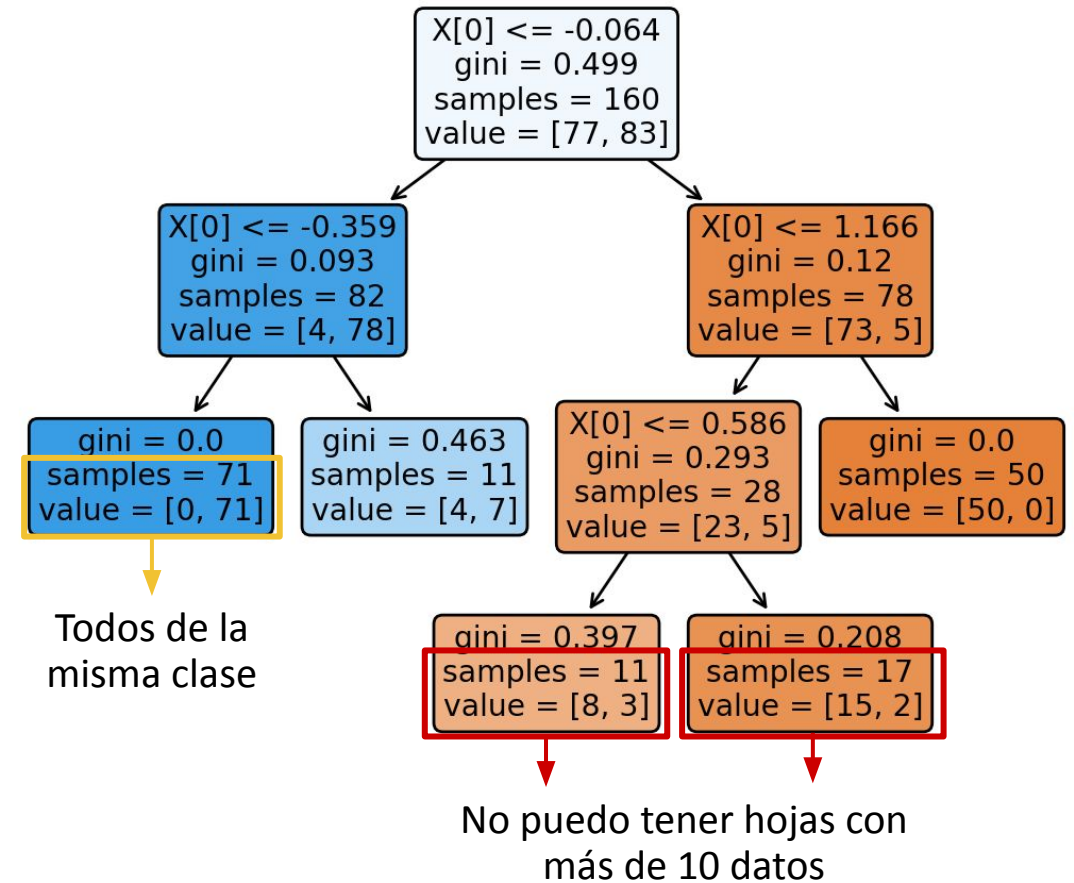
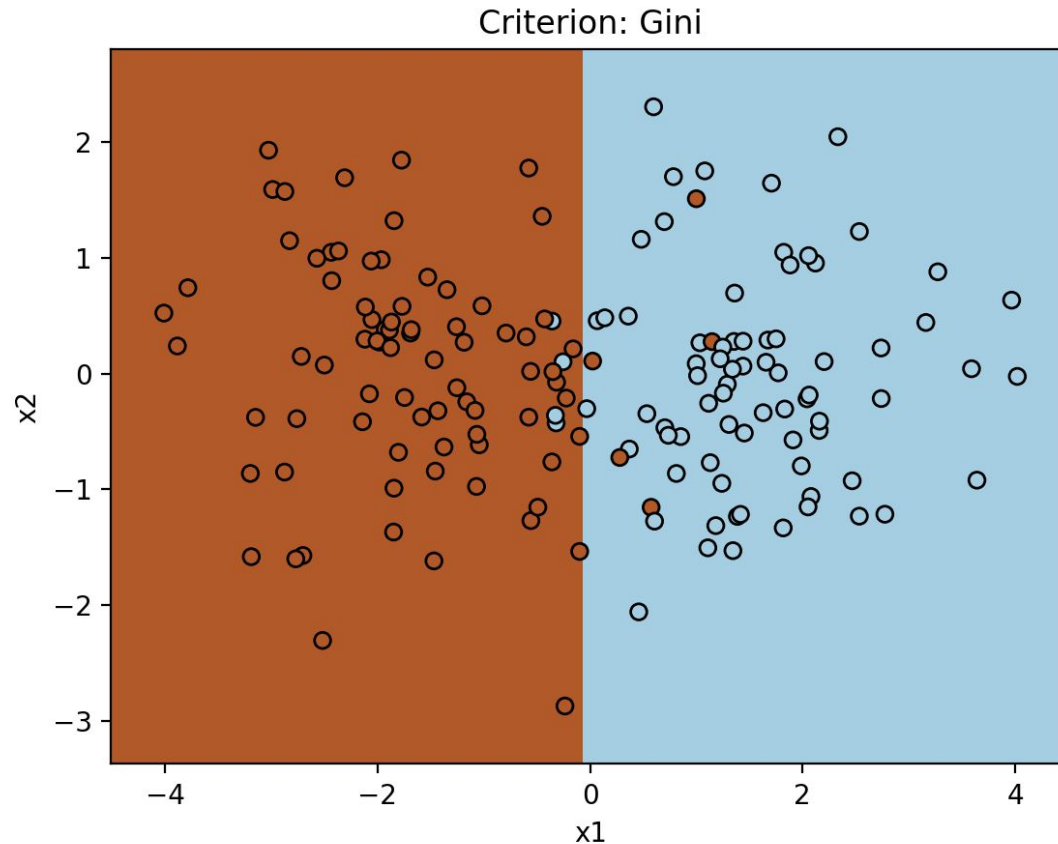
Criterios de parada

- Limitar a una profundidad máxima (`max_depth`)
- Restringir la decisión en un nodo interno sólo para situaciones en las que cada split tiene una cantidad mínima de instancias (`min_samples_split`)
- Definir un número mínimo de observaciones para aceptar una hoja (`min_samples_leaf`)
- ...

Cuando corto, el nodo truncado se transforma en hoja, de la clase mayoritaria

Complejidad del árbol

Efecto de `min_samples_leaf = 10`



Cuando las hojas quedarían con menos de 10 instancias, no las acepto. El nodo interno anterior se transforma en hoja y asigna la clase mayoritaria. Esto limita la profundidad del árbol.

Complejidad del árbol

Reducir complejidad después de construir el árbol

Poda (pruning):

Podar ramas cuando ello mejore la performance en datos separados.

Rule post-pruning:

Convertir árbol a reglas; sacar precondiciones de las reglas cuando ello mejore su performance sobre datos separados; reordenar las reglas según accuracy.

A Look at Pruning



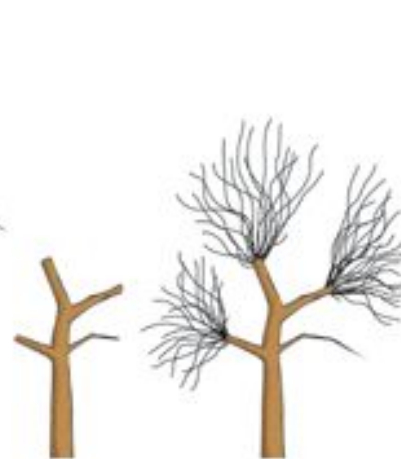
Before Pruning

GOOD



Well-Pruned, Open Head

NOT GOOD

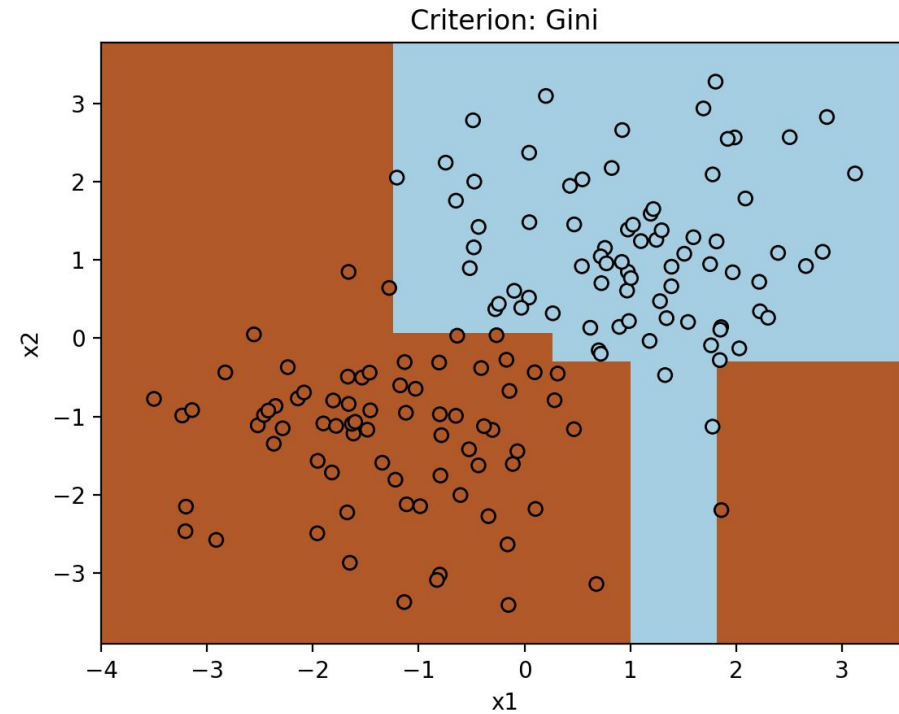
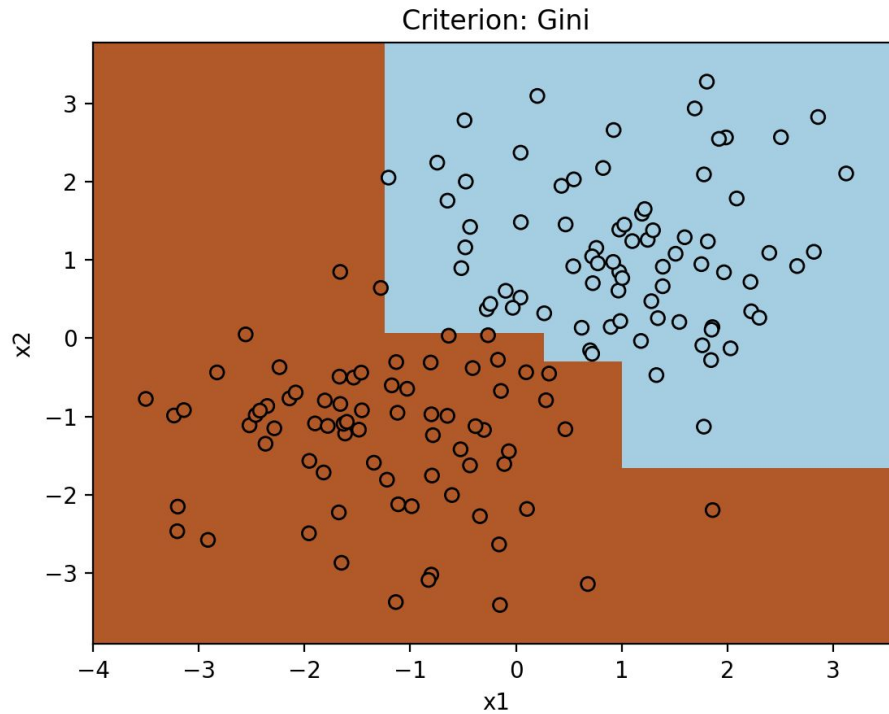


Topping produces clumps of uncontrolled growth



Sesgo inductivo

Inestabilidad de la estructura de los árboles (según datos y atributos)



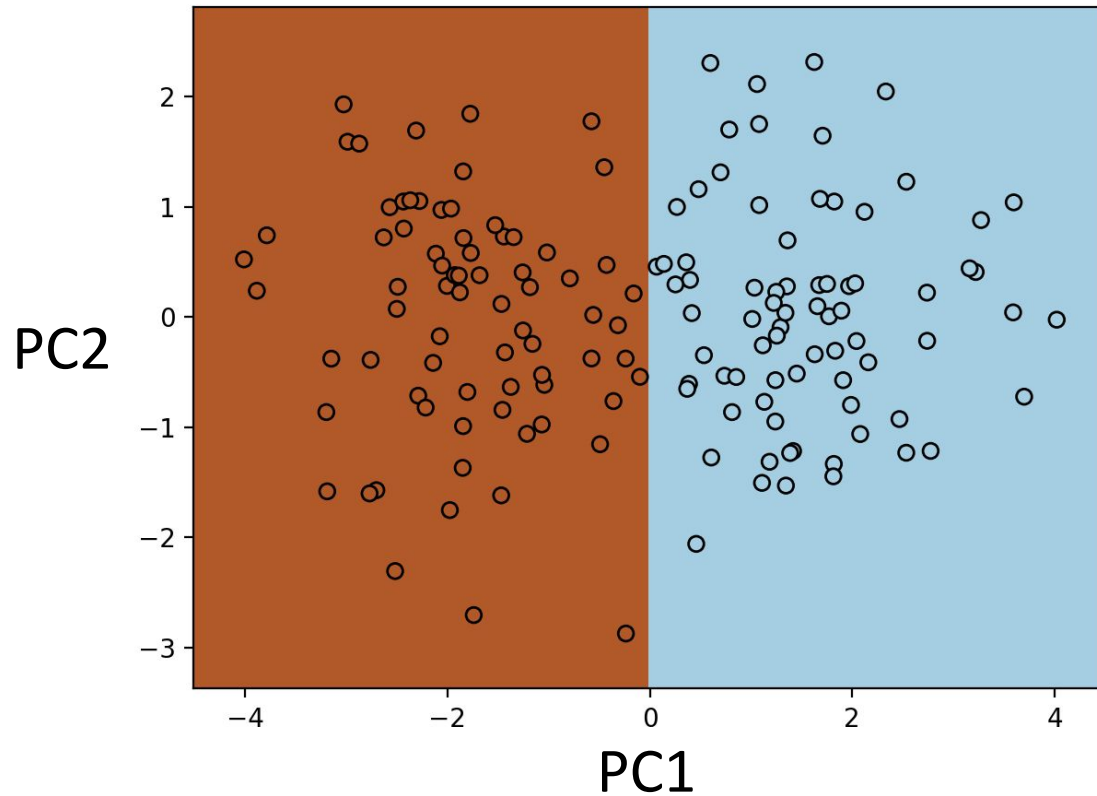
Cambios chicos en el entrenamiento pueden producir resultados muy distintos.

Sklearn usa un algoritmo estocástico. Con el mismo set, los resultados pueden cambiar.

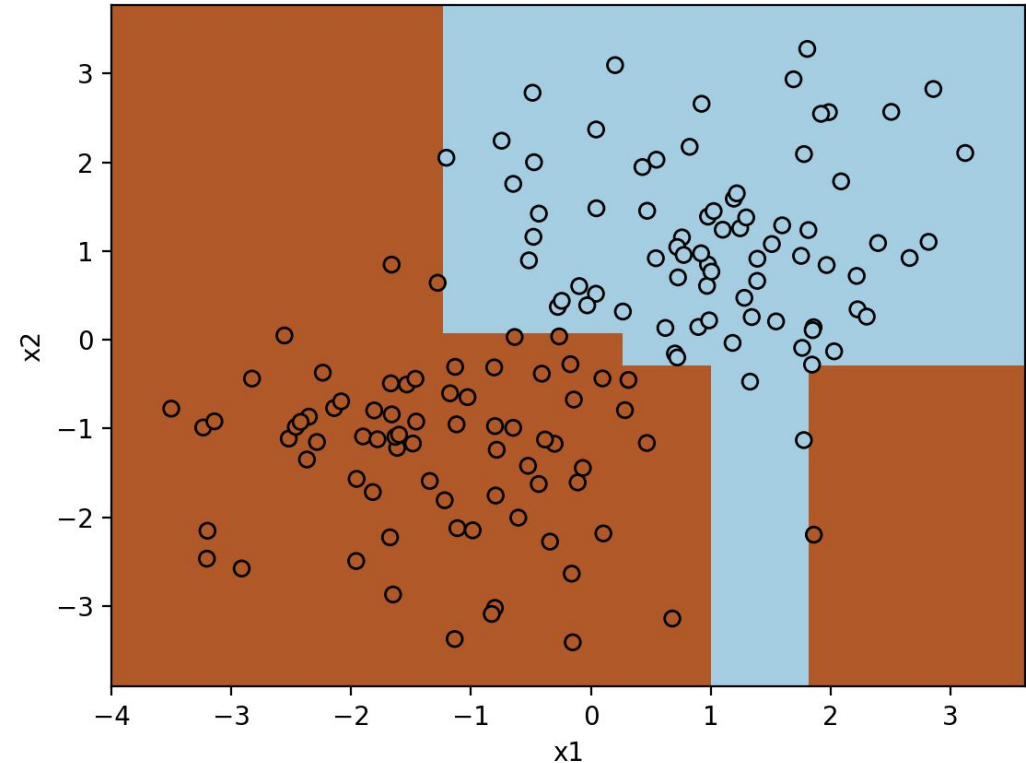
Para evitar esto se fija el `random_state`.

Sesgo inductivo

Ángulo de cortes



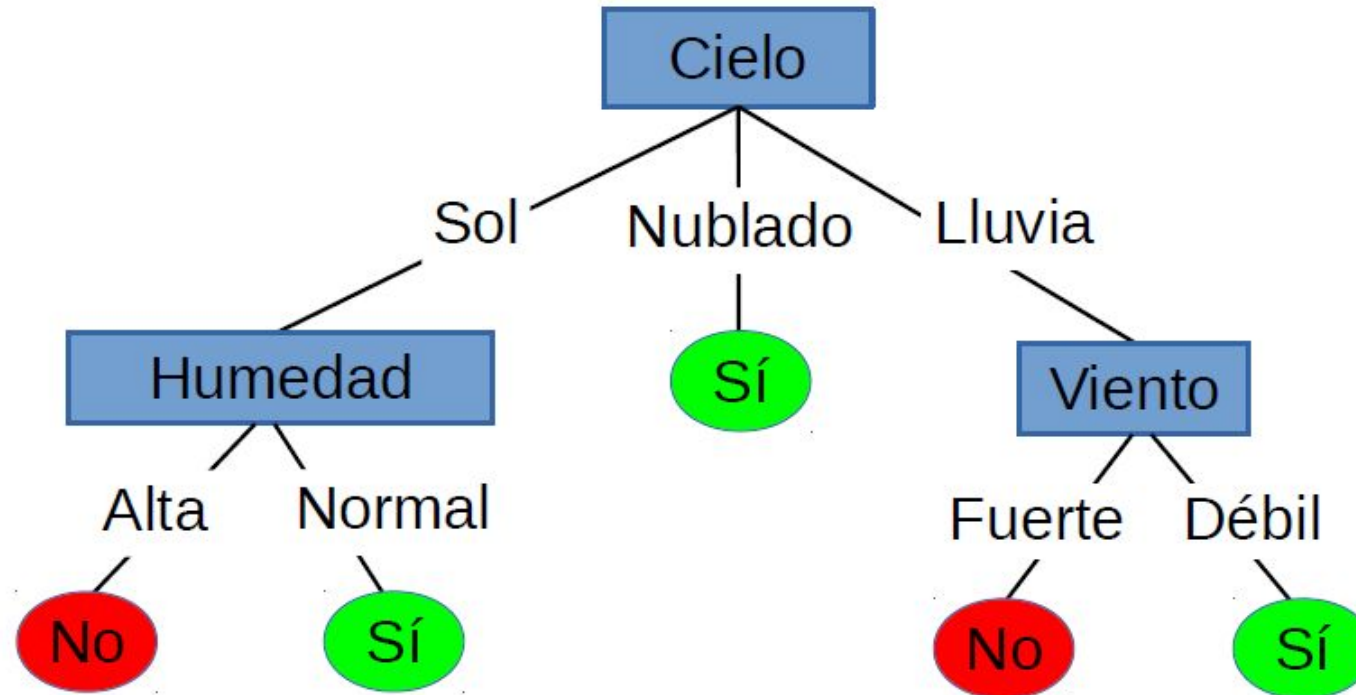
Mismos datos, rotados en espacio atributos



Al ser ortogonales es muy sensible a la dirección del espacio de las entradas.
Puedo transformar los datos para rotar y que facilite el problema (Ej: PCA)

Sesgo inductivo

Elijo como nodo raíz aquel atributo que resulta en mayor ganancia

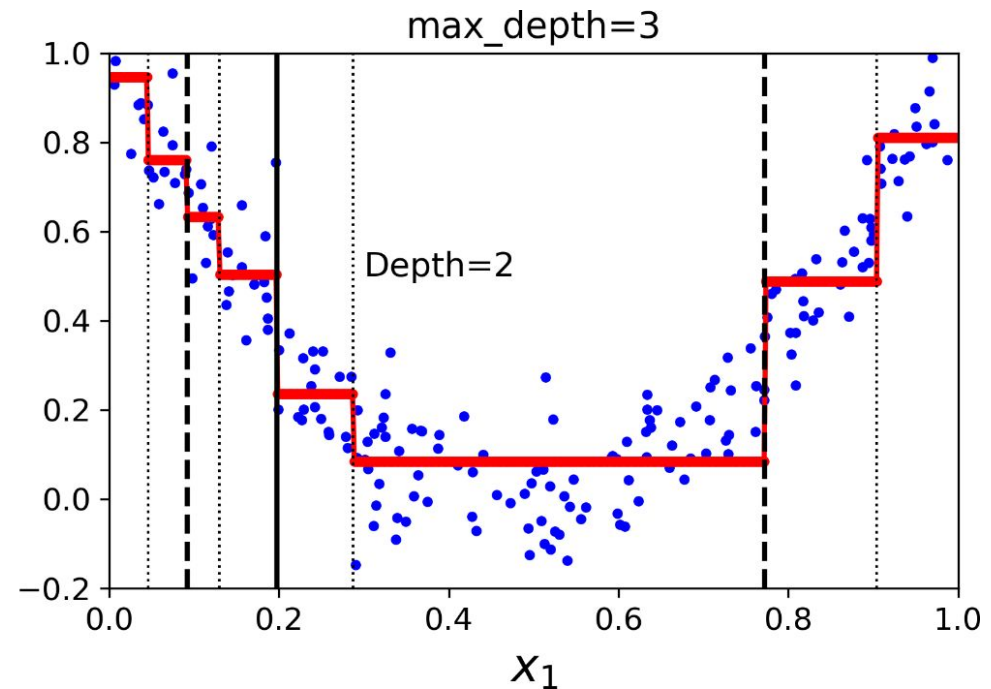
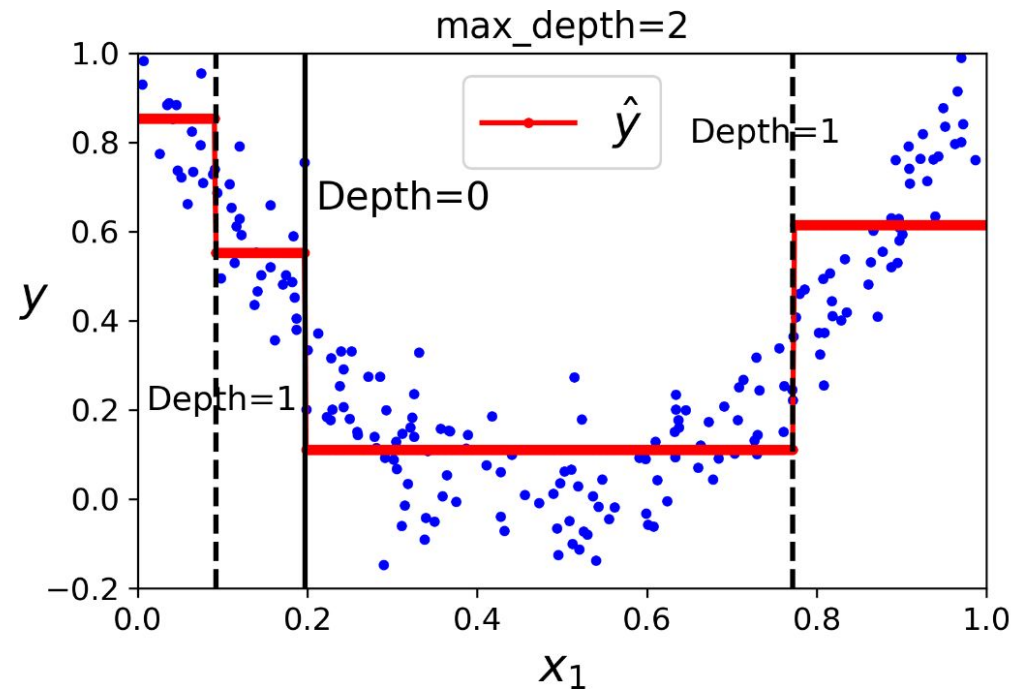


Los atributos más informativos quedan más cerca de la raíz

Árboles de decisión (regresión)

Construcción: En cada nodo, usar reducción de desvío estándar de Y en lugar de gini/info gain

Consulta: Al llegar a una hoja, devolver el promedio de Y sobre las instancias de la hoja



$$y_{\tau} = \frac{1}{N_{\tau}} \sum_{\mathbf{x}_n \in \mathcal{R}_{\tau}} t_n$$

$$Q_{\tau}(T) = \sum_{\mathbf{x}_n \in \mathcal{R}_{\tau}} \{t_n - y_{\tau}\}^2$$

sklearn.tree.DecisionTreeRegressor

Árboles de decisión

- + No paramétrico
- + Buen manejo de no linealidades
- + Facilidad para seleccionar atributos relevantes
- + Buen manejo de interacciones entre variables
- + Manejo elegante de atributos categóricos (según la implementación)
- + Interpretables si no son excesivamente profundos
- + Son rápidos para predicción ($O(\log_2(m))$)
- No tienen una gran capacidad predictiva
- Inestables ante cambios de los datos de entrenamiento (por ej. cambiar series)
- Pueden ser poco interpretables si tienen gran profundidad

++ Base de los Ensamblados
(Random Forest, XGBoost)

Clase que viene!



A los Colabs...!!

