

3. Árboles de decisión

Ejercicio 3.1.

- Hacer en papel y lápiz un árbol de decisión correspondiente a entrenar con los datos de la Tabla 1. Utilizar el criterio “Ganancia de Información” (reducción de entropía) para calcular el atributo que mejor separa cada decisión.
- Armar tres ejemplos posibles de instancias no vistas y usar el árbol para predecir la clase de salida.
- Calcular la importancia de cada atributo mediante la medida “disminución media de la impureza”.
- ¿Cómo cambiaría el árbol si restringiéramos su altura a 2 niveles? ¿Cuál sería el resultado de las predicciones? ¿Algunas de las instancias existentes en la tabla, serían clasificadas incorrectamente?

Cielo	Temperatura	Humedad	Viento	¿Salgo? (clase a predecir)
Sol	Calor	Alta	Débil	No
Sol	Calor	Alta	Fuerte	No
Nublado	Calor	Alta	Débil	Sí
Lluvia	Templado	Alta	Débil	Sí
Lluvia	Frío	Normal	Débil	Sí
Lluvia	Frío	Normal	Fuerte	No
Nublado	Frío	Normal	Fuerte	Sí
Sol	Templado	Alta	Débil	No
Sol	Frío	Normal	Débil	Sí
Lluvia	Templado	Normal	Débil	Sí
Sol	Templado	Normal	Fuerte	Sí
Nublado	Templado	Alta	Fuerte	Sí
Nublado	Calor	Normal	Débil	Sí
Lluvia	Templado	Alta	Fuerte	No

Tabla 1: Salgo a caminar

Ejercicio 3.2. ¿Quién es quién?

El quién es quién es un juego (buscar en Google) en el que hay que adivinar el nombre del personaje del rival. Para ello se hacen preguntas que son respondidas por sí o por no.

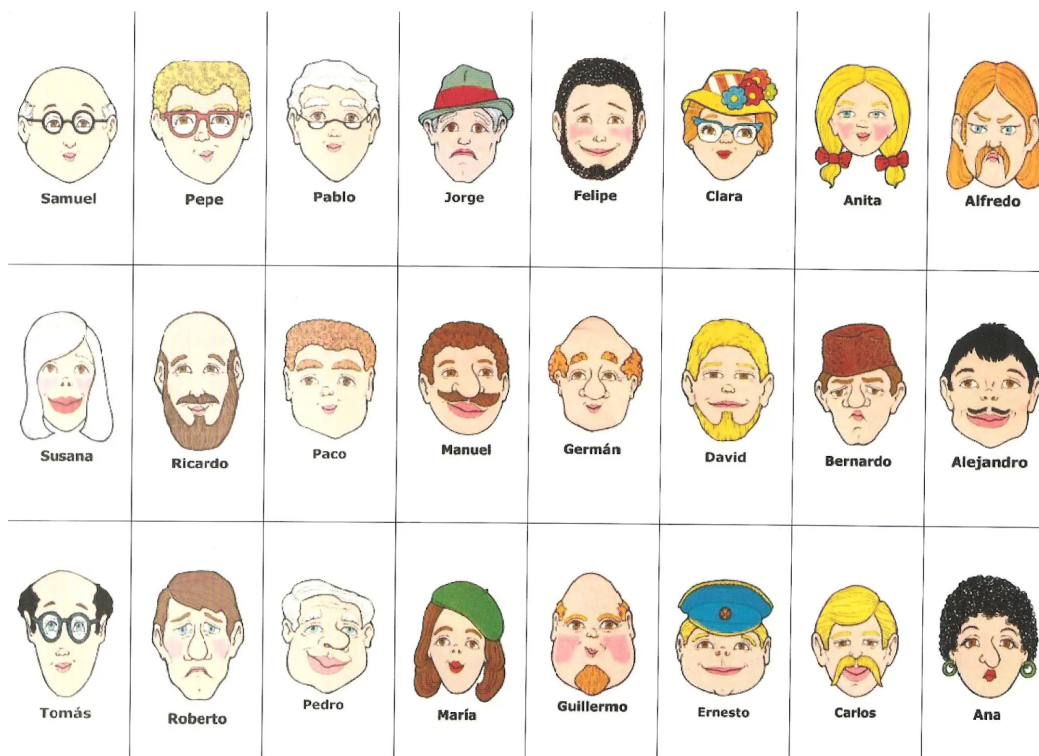


Figura 2: Juego de ¿quién es quién?

Tomemos la libertad de jugar únicamente a adivinar el sexo biológico (masculino/femenino).

- (a) Si preguntamos si el personaje es calvo... ¿Cuál es la ganancia gini al separar por este atributo?
- (b) Ordenar por valor de ganancia gini qué conviene como primera pregunta: i) ¿Tiene calvicie? ii) ¿Tiene cabello rubio? (De entre estas posibles categorías: morocho, negro, rubio, blanco, colorado) iii) ¿Tiene vello facial? (bigote y/o barba) iv) ¿Tiene sombrero?

Ejercicio 3.3. En la Figura 3 puede verse diversas regiones en el espacio de atributos para un problema de clasificación.

- (a) Determinar cuáles de ellas pueden haber sido generadas por árboles de decisión de altura máxima 5.
- (b) Para las que lo sean, mostrar un árbol que hubiese generado estas regiones (suponer ejes x_1 y x_2)

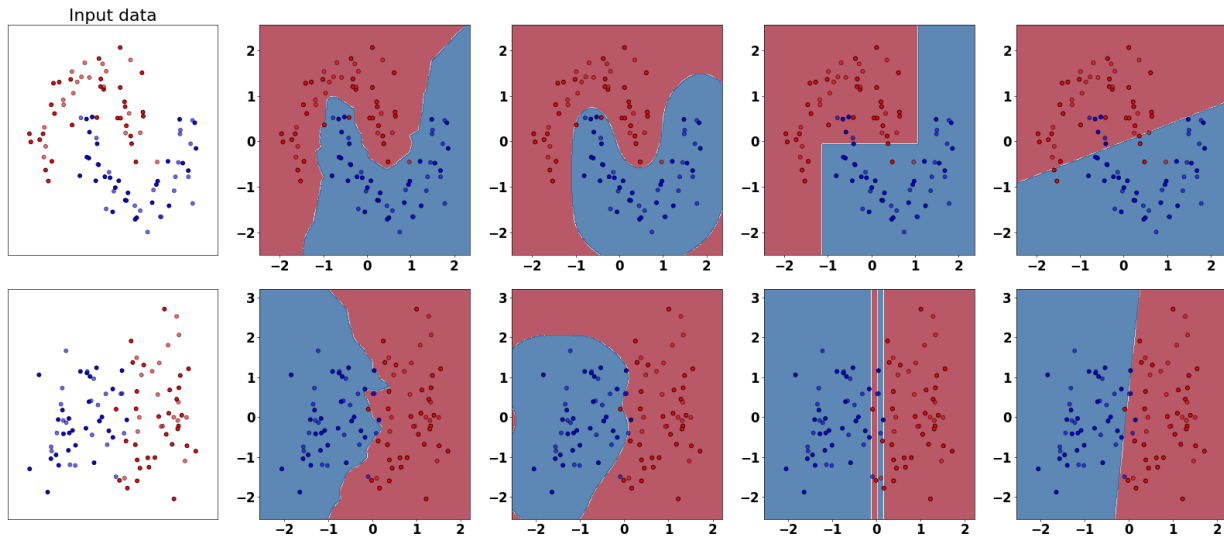


Figura 3: Cortes en el espacio de atributos

Ejercicio 3.4. En la Figura 4 se visualiza la altura real de 20 árboles de decisión distintos que fueron entrenados variando el parámetro max_depth (profundidad máxima) desde 1 hasta 20 inclusive. Observamos que a partir de max_depth igual a 12, la altura real del árbol es constante. Si suponemos que estos árboles crecen de manera balanceada:

- (a) Explicar una razón posible por la cual sucede este fenómeno.
- (b) Suponga un dataset con 1000 instancias ¿Hasta qué profundidad máxima tiene sentido probar?.
- (c) Además del número de instancias, ¿de qué otro factor puede depender la altura real del árbol?

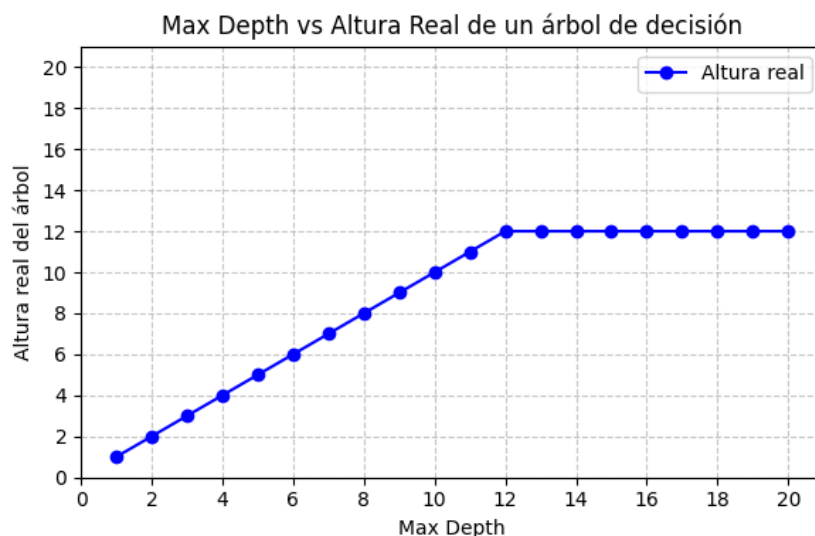


Figura 4: profundidad máxima vs. altura real: árboles decisión

Ejercicio 3.5. Determinar cuáles de las siguientes son afirmaciones verdaderas:

- (a) El objetivo de construir un árbol de decisión es crear el árbol de menor tamaño posible en el cual las hojas contengan valores de una sola clase.
- (b) Los algoritmos de construcción vistos (CART, ID3, etc) exploran todos los posibles árboles y se quedan con el que mejor separa a las instancias.
- (c) Un atributo puede aparecer sólo una vez en cada rama del árbol (llamamos rama a un camino directo desde una hoja hasta la raíz).
- (d) Un par (atributo, corte) puede aparecer sólo una vez en cada rama del árbol (llamamos rama a un camino directo desde una hoja hasta la raíz).
- (e) Un árbol de decisión, con criterios de corte suficientemente laxos, puede siempre conseguir 100 % de aciertos en los datos de entrenamiento.
- (f) Un árbol de decisión, con criterios de corte suficientemente laxos, puede siempre conseguir 100 % de aciertos en los datos de entrenamiento siempre y cuando no haya contradicciones entre las etiquetas de instancias observacionalmente iguales.

Ejercicio 3.6. En árboles de decisión, describa cómo computaría, para una instancia dada, el vector de probabilidades por clase (un vector que tenga en sus componentes, la probabilidad de que la instancia pertenezca a cada una de las distintas clases posibles).

Ejercicio 3.7. Sesgo inductivo

- (a) ¿Cuál el sesgo inductivo del algoritmo que construye el árbol de decisión?
- (b) ¿Qué sucede cuando dos atributos empatan en ganancia de información? ¿Esta decisión es parte del sesgo inductivo?

Ejercicio 3.8. Dado el algoritmo de construcción de árboles visto en clase para atributos continuos: Sea S una muestra de instancias con atributos A . Para construir un árbol de decisión ejecutamos:

(1) Mientras no se cumpla un criterio de detención:

- (I) Creamos `nodo_actual`, un nodo que representa una decisión de corte.
- (II) Elegimos el par $a \in A, c \in R$ entre los posibles pares $\langle \text{atributo}, \text{corte} \rangle$, que mejor divida a S para `nodo_actual` según $\Delta M(S, \langle a, c \rangle)$
(ΔM es una función que devuelve cuánto gana si dividido a S en dos utilizando $\langle a, c \rangle$ y según la medida M . Por ejemplo ΔM podría ser `GananciaDeInformación`).
- (III) Crear dos hijos de `nodo_actual`.
- (IV) Dividir las instancias de S en los nuevos nodos, según $\langle a, c \rangle$:

$$S_{\leq} \leftarrow \{x | x \in S \wedge x[a] \leq c\}$$

$$S_{>} \leftarrow \{x | x \in S \wedge x[a] > c\}$$

(v) Repetir para cada hijo.

- (2) El valor asignado a cada región resultante (a cada hoja) será el de la clase mayoritaria de las instancias que pertenezcan a esta región.

Se pide:

- (a) Escribir el pseudocódigo (puede ser similar a python) para el paso (b) (elegir el mejor par a, c entre los posibles pares). Es decir, definir `mejor_corte(S, A, ΔM)`, en donde S representa la muestra, A un conjunto de atributos y ΔM la función que computa la ganancia de una división. Puede suponer dadas funciones tales como $S[a]$ que devuelve la columna de valores para el atributo a . Es importante que esta función no evalúe dos veces cortes que devuelven exactamente las mismas regiones para un mismo atributo.
- (b) Introducir los cambios necesarios en el algoritmo general (y si fuera necesario, en la función que definieron para `mejor_corte`) que permita medir la importancia de cada atributo. La importancia deberá ser un valor numérico entre 0 y 1.

Ejercicio 3.9. Explicar cómo funciona el cálculo de Importancia de Permutación para un modelo genérico M . En el caso de árboles de decisión, ¿qué esperarían de la importancia de atributos altamente correlacionados?

Ejercicio 3.10. Resolver el notebook `notebook_3_arboles_de_decision_sklearn.ipynb`.

Ejercicio 3.11. Completar el notebook `notebook_4_implementacion_arbol.ipynb`. Este notebook contiene una implementación parcial de un algoritmo de creación de árboles de decisión.