Machine Learning I

Tema 2. Aprendizaje Supervisado
2.1 Árboles de Decisión

Profesor: José Luis Sancho Gómez

Curso 2023-2024

¿Qué es?

Los árboles de decisión son máquinas de aprendizaje supervisado que sirven para clasificar o aproximar.

Supongamos el siguiente problema:

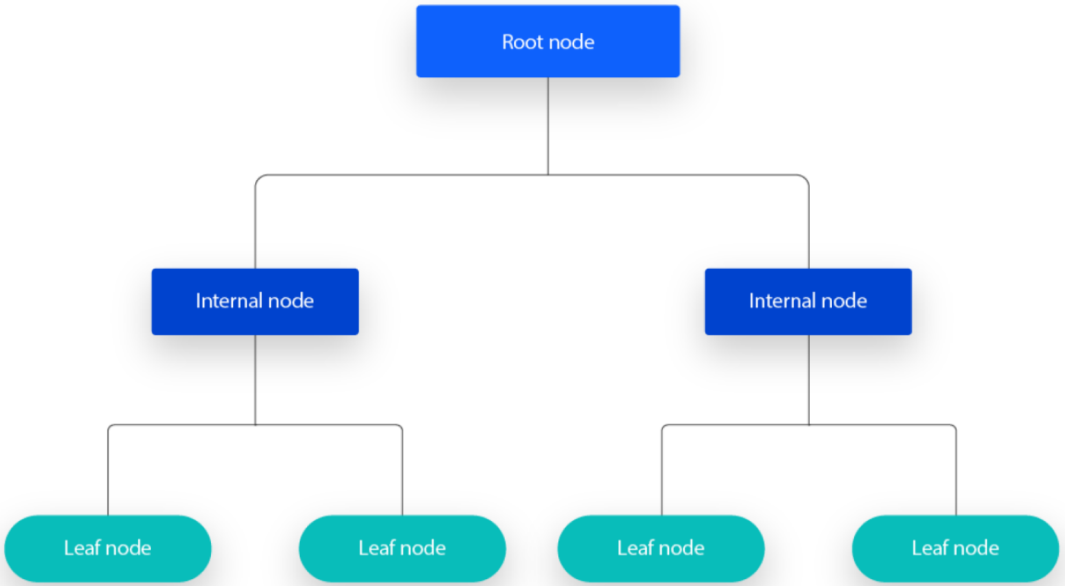
Paciente	Presión Aterial	Urea en sangre	Gota	Hipotiroidismo	Administrar Tratamiento
1	Alta	Alta	Sí	No	No
2	Alta	Alta	Sí	Sí	No
3	Normal	Alta	Sí	No	Sí
4	Baja	Normal	Sí	No	Sí
5	Baja	Baja	No	No	Sí
6	Baja	Baja	No	Sí	No
7	Normal	Baja	No	Sí	Sí
8	Alta	Normal	Sí	No	No
9	Alta	Baja	No	No	Sí
10	Baja	Normal	No	No	Sí
11	Alta	Normal	No	Sí	Sí
12	Normal	Normal	Sí	Sí	Sí
13	Normal	Alta	No	No	Sí
14	Baja	Normal	Sí	Sí	No

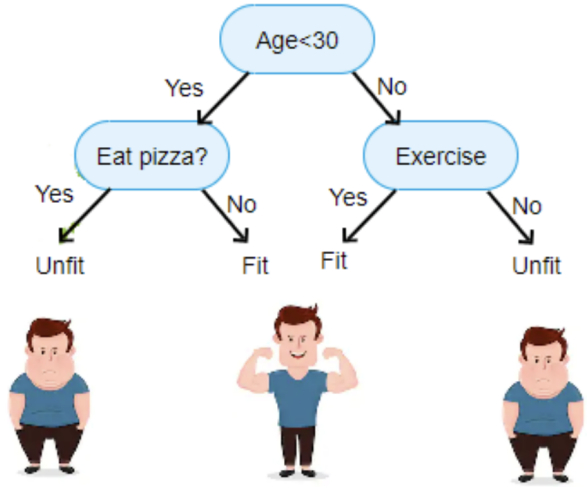
- Planteamiento del problema: ¿Cuál es la **mejor secuencia de preguntas** para saber la clase a la que pertenece un objeto descrito por sus atributos?
- Evidentemente, la "mejor secuencia" es aquella que con el **menor número de preguntas**, devuelve una respuesta suficientemente buena.
- ¿Qué es mejor preguntar primero si tiene gota o cómo tiene la presión arterial?

Arquitectura

Un árbol de decisión es una estructura jerárquica que consta de un nodo raíz, ramas, nodos internos y nodos hoja.

- Comienza con un **nodo raíz** sin ramas entrantes. Las ramas salientes del nodo raíz alimentan los nodos internos.
- Los **nodos internos** evalúan características disponibles para formar subconjuntos homogéneos, indicados por nodos hoja o nodos terminales.
- Los **nodos hoja** representan todos los resultados posibles dentro del conjunto de datos.





Ventajas y desventajas

Pros

- Fáciles de entender e interpretar.
- Sirven también para establecer reglas.
- No lineales.
- Menos pre-procesado de los datos: son robustos ante presencia de datos erróneos (outliers), valores faltantes o tipo de datos.
- Es un método no paramétrico (i.e., no hay suposición acerca del espacio de distribución y la estructura del clasificador).

Cons

- Sobreajuste. Los árboles más pequeños son más fáciles de interpretar, pero los más grandes pueden resultar en sobreajuste.
- Pérdida de información al categorizar variables continuas.
- Precisión: otros métodos (por ejemplo, SVM) a menudo tienen tasas de error 30 % más bajas que los árboles básicos (ID.3 y CART).
- Inestabilidad: un pequeño cambio en los datos puede modificar ampliamente la estructura del árbol (distintos conjuntos, distintos árboles). Varianza alevada.

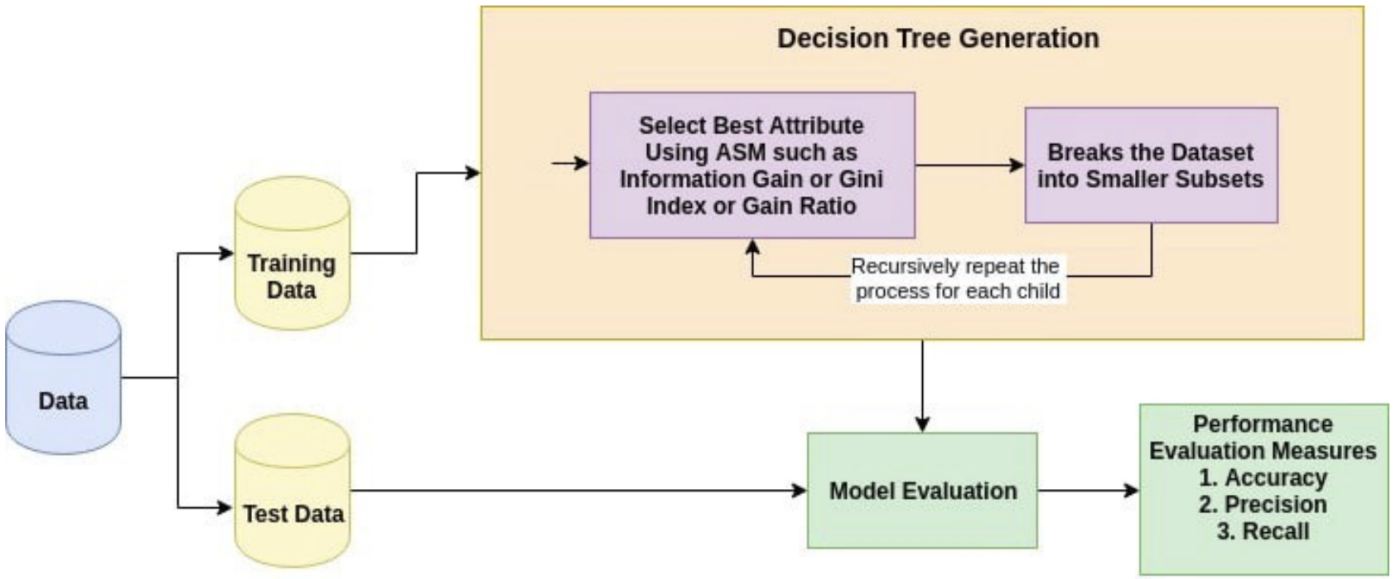
Árboles de decisión

Definición alternativa: recursividad

Un árbol de decisión es una estructura recursiva formada por nodos, en el que existe:

- Un nodo raíz
- El nodo raíz tiene uno o más subnodos
- Cada uno de los subnodos puede ser, a su vez, raiz de un árbol

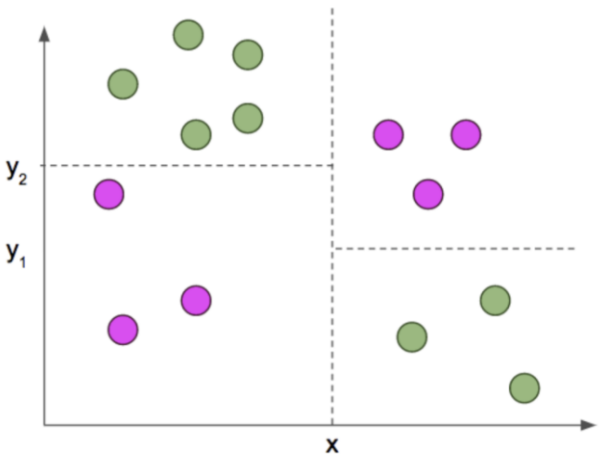
Está característica recursiva hace que muchos de los algoritmos para crearlos se comporten también de manera recursiva.

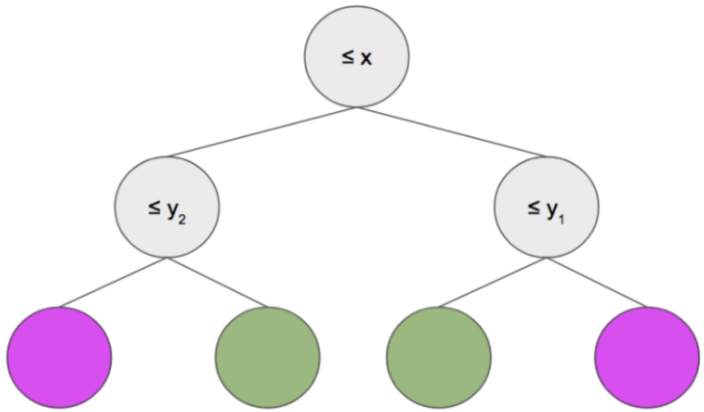


Clasificación vs. Regresión

Clasificación

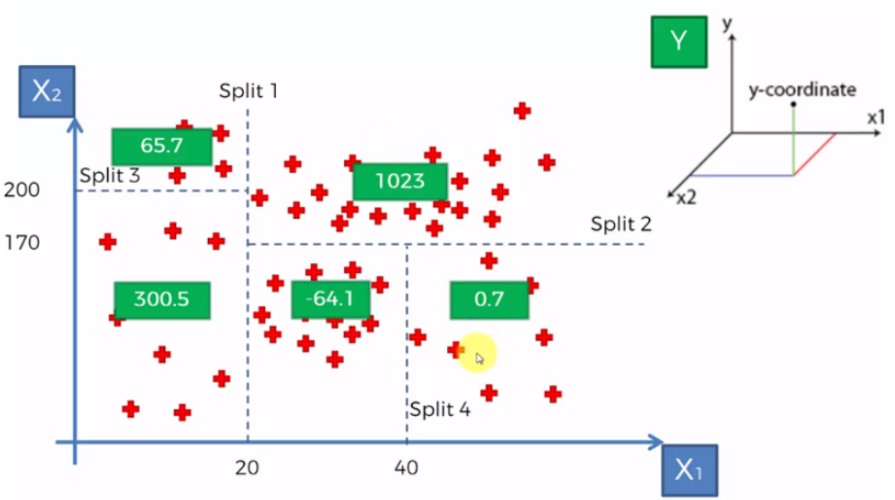
- Variable dependiente es categórica.
- Los valores de los nodos hoja son la **moda** de las obervaciones de la región.

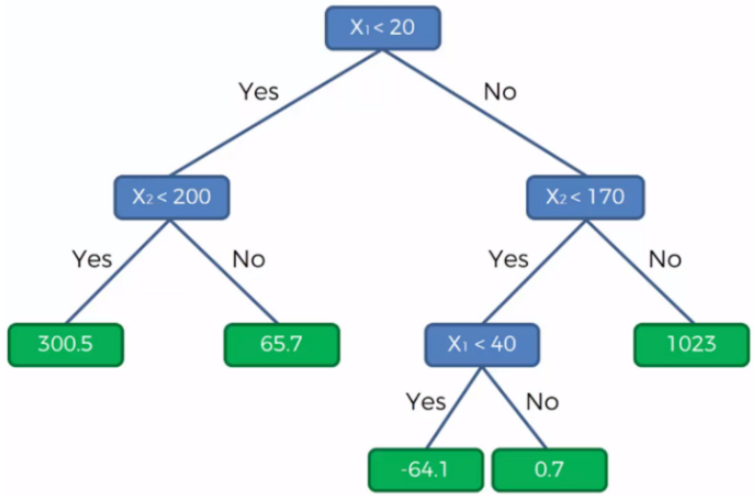




Regresión

- Variable dependiente es continua.
- Los valores de los nodos hoja son la **media** de las obervaciones de la región.





Particiones

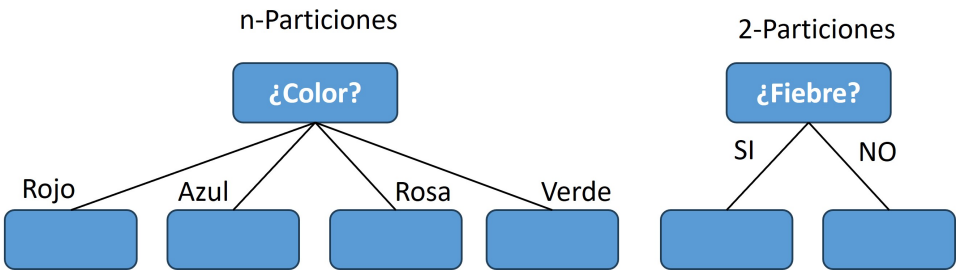
- Cada nodo define una **partición** del conjunto de entrenamiento en función de los datos que representa.
- Las particiones producen subconjuntos que son **exhaustivos y excluyentes**.
- Cuestiones clave:
  - **Tipos de particiones**: cuantos más, más posibilidad de encontrar patrones y, por tanto, los árboles más precisos y expresivos.
  - **Número de particiones**: A más particiones mayor complejidad: equilibrio entre complejidad y precisión.
  - Selección del **mejor atributo** en cada paso
  - Selección del **mejor valor** de umbral de los valores.
  - ...

Construcción: Particiones posibles I

- Los algoritmos más populares sólo proponen un tipo de partición para valores nominales y otro para valores numéricos:
  - **Particiones nominales**. En el caso que tengamos un atributo x_i que tenga como posibles valores $\{v_1, v_2, \cdots, v_n\}$, sólo es posible la partición

$$(x_1 = v_1, x_2 = v_2, \cdots, x_n = v_n)$$

que da lugar a árboles con nodos con más de dos nodos hijos.

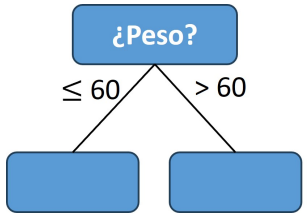


En el caso de árboles binarios se tienen que evaluar n particiones (una por cada posible valor), definidas por $(x_i = v_i, x_i \neq v_i)$.

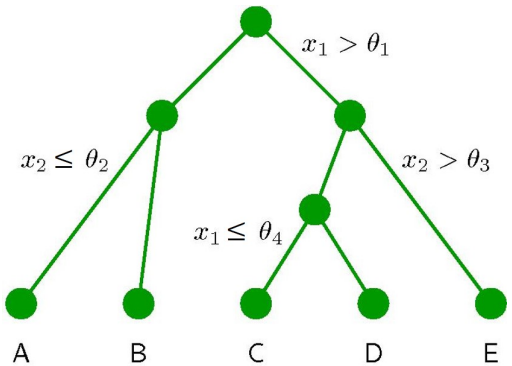
Construcción: Particiones posibles II

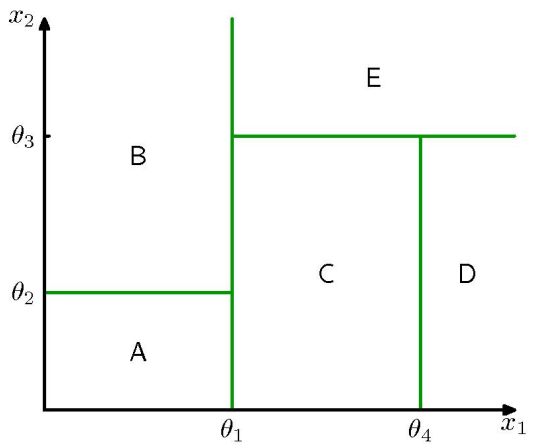
- **Particiones numéricas:** si el atributo x_i es numérico y continuo, se intenta definir particiones que separe las instancias en intervalos de la forma

$$(x_i \leq a, x_i > a)$$



eligiendo diferentes valores de a tenemos diferentes particiones. La expresividad resultante se conoce como *expresividad cuadrangular* y que no relacionan atributos (sólo un atributo cada vez).





ID3: Algoritmo básico de aprendizaje

ID3

El algoritmo básico de aprendizaje es el **ID3 (Iterative Dichotomiser 3)**, J. Ross Quinlan, investigador australiano que propuso el método en 1983.

- El método ID3 trata de encontrar una partición que asegure la **máxima capacidad predictiva y la máxima homogeneidad** de las clases.
- Medida de homegeneidad: la **entropía**.
- Repetición de "**cortes en dos**" hasta que se cumpla una determinada condición.

ID3: Entropía

- Para determinar el mejor atributo, el ID3 utiliza la **entropía**.

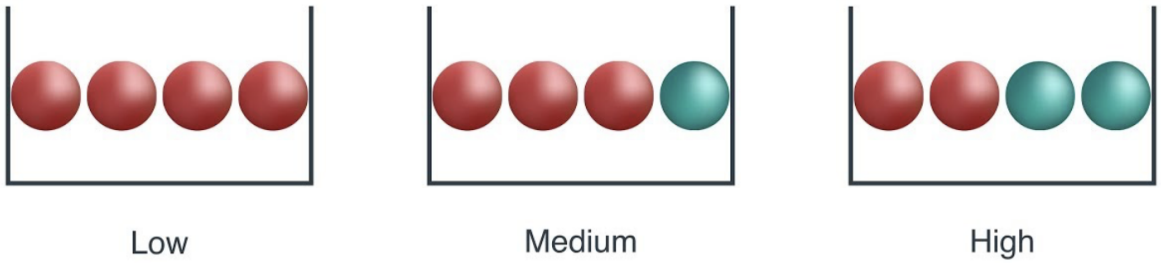
Sea S un conjunto de entrenamiento. Sea $p_\oplus$ la proporción de instancias positivas en S y $p_\ominus$ la proporción de instancias negativas en S. La **entropía de S** es:

$$H(S) = p_\oplus \log_2 \frac{1}{p_\oplus} + p_\ominus \log_2 \frac{1}{p_\ominus} = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

(Relación de la entropía con los conceptos de desorden, equiprobabilidad y homogeneidad)

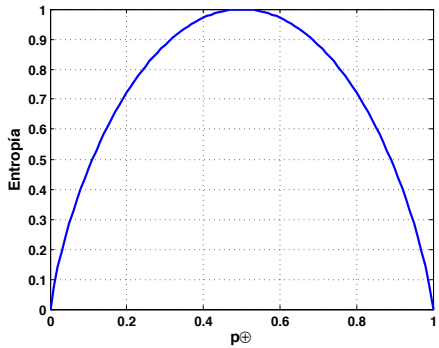
ID3: Entropía

La entropía nos mide la homogeneidad de los datos (relación inversa)



Para el caso binario:

- Entropía igual a 1 $\rightarrow$ mínima homogeneidad (equiprobabilidad: $p_{\ominus} = p_{\oplus}$).
- Entropía igual a 0 $\rightarrow$ máxima homogeneidad (todas las instancias de una clase).



A la hora de construir un árbol es preferible crear nodos con nodos hoja homogéneos, es decir, de **baja entropía**.

ID3: Ganancia de Información

- Sea un conjunto de datos $\mathcal{X}$ con entropía $H(\mathcal{X})$.
- Si elegimos un atributo A para crear un nodo del árbol, la entropía esperada es:

$$H(\mathcal{X}, A) = \sum_{v \in \text{valores}(A)} \frac{|\mathcal{X}_v|}{|\mathcal{X}|} H(\mathcal{X}_v)$$

  siendo $\mathcal{X}_v$ el subconjunto de $\mathcal{X}$ con todos las instancias con $A = v$.
- Por lo tanto, la reducción esperada de la entropía, o lo que es lo mismo la **Ganancia de Información**, al elegir el atributo A como nodo de decisión del árbol es

$$Ganancia(\mathcal{X}, A) = H(\mathcal{X}) - H(\mathcal{X}, A)$$

- Por tanto, se elige el atributo que produzca hojas homogéneas, es decir, la **máxima** ganancia de información.

ID3: Ejemplo

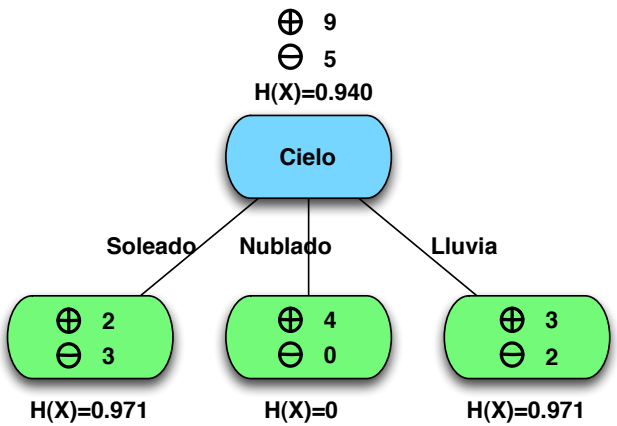
Atributos nominales (no numéricos)

Día	Cielo	Temperatura	Humedad	Viento	Jugar
D1	Soleado	Calor	Alta	Flojo	No
D2	Soleado	Calor	Alta	Fuerte	No
D3	Nublado	Calor	Alta	Flojo	Si
D4	Lluvia	Templado	Alta	Flojo	Si
D5	Lluvia	Frio	Normal	Flojo	Si
D6	Lluvia	Frio	Normal	Fuerte	No
D7	Nublado	Frio	Normal	Fuerte	Si
D8	Soleado	Templado	Alta	Flojo	No
D9	Soleado	Frio	Normal	Flojo	Si
D10	Lluvia	Templado	Normal	Flojo	Si
D11	Soleado	Templado	Normal	Fuerte	Si
D12	Nublado	Templado	Alta	Fuerte	Si
D13	Nublado	Calor	Normal	Flojo	Si
D14	Lluvia	Templado	Alta	Fuerte	No

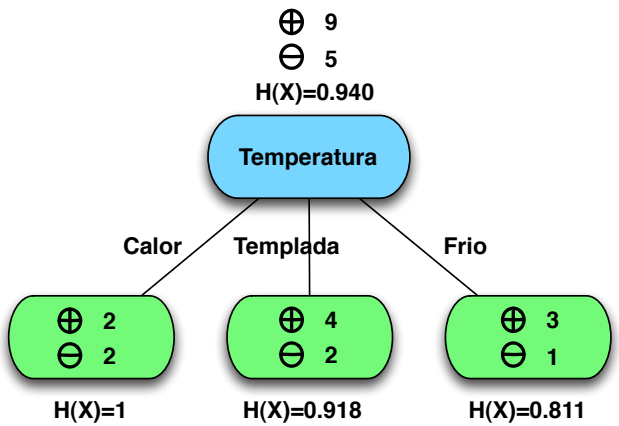
- En el ejemplo del tenis tenemos 9 objetos clasificados como $\oplus$ y 5 como $\ominus$, con lo que

$$H([9\oplus, 5\ominus]) = -0.642 \cdot \log_2 0.642 - 0.358 \cdot \log_2 0.358 = 0.94$$

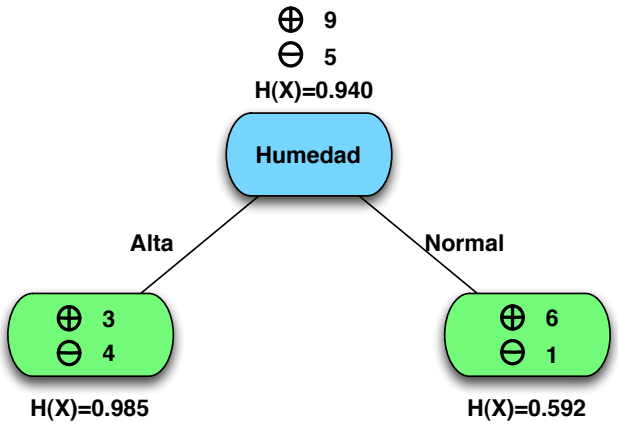
ID3: Ejemplo



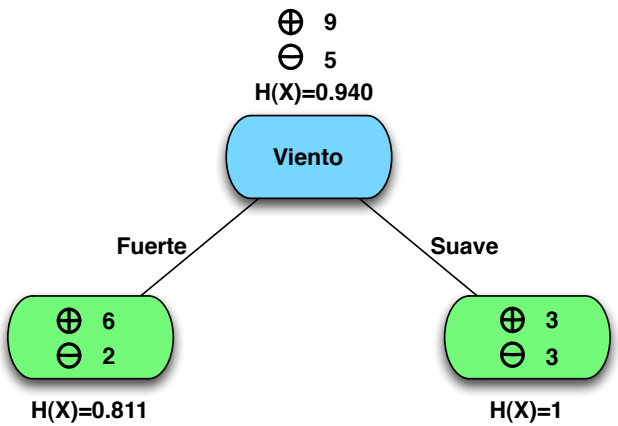
Ganancia(X,Cielo)=
0.94-(5/14) * 0.971- (4/14) * 0 -(5/14) * 0.971 = 0.246



Ganancia(X,Temperatura)=
0.94-(4/14) * 1 - (6/14) * 0.918 -(4/14) * 0.811 = 0.02

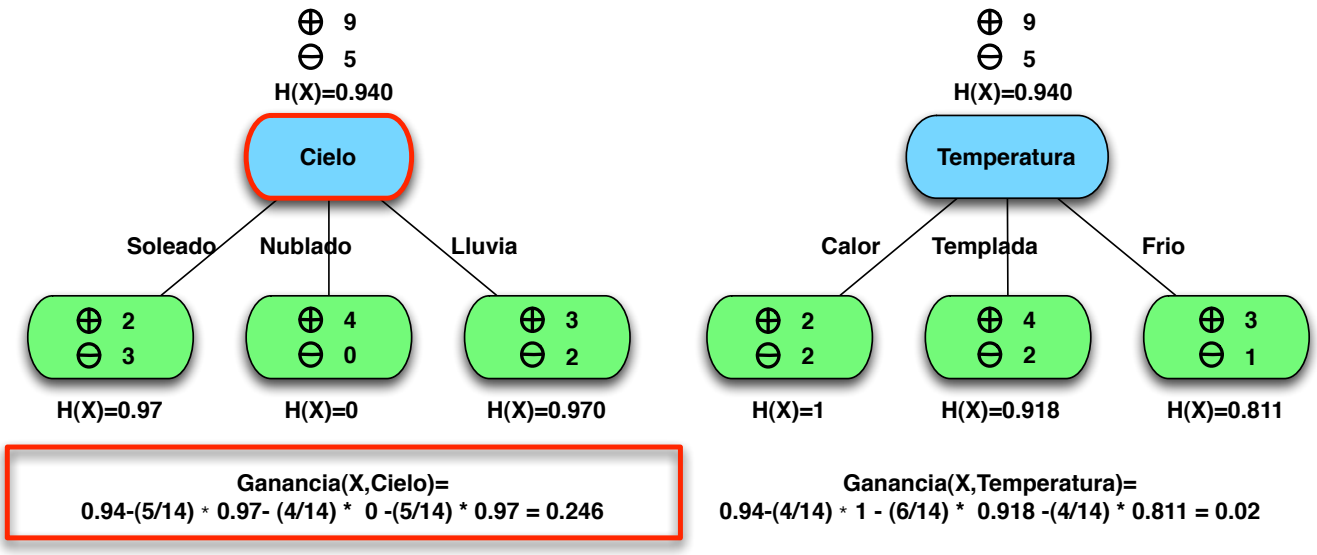


Ganancia(X,Humedad)=
0.94-(7/14) * 0.985 - (7/14) * 0.592 = 0.151



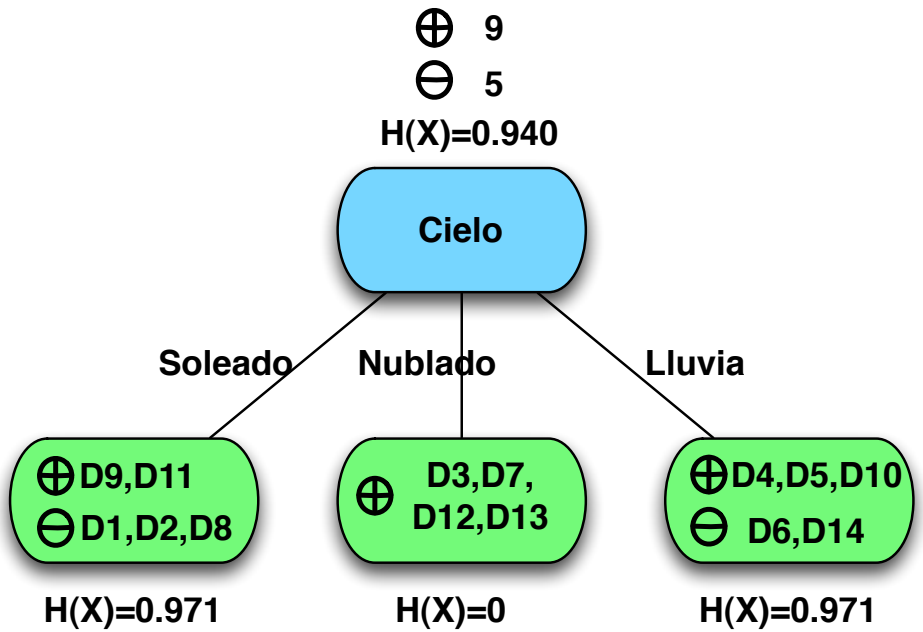
Ganancia(X,Viento)=
0.94-(8/14) * 0.811 - (6/14) * 1 = 0.048

ID3: Ejemplo


⊕ 9
⊖ 5
H(X)=0.940
Cielo
Soleado
Nublado
Lluvia
⊕ 2
⊖ 3
H(X)=0.97
⊕ 4
⊖ 0
H(X)=0
⊕ 3
⊖ 2
H(X)=0.970
Ganancia(X,Cielo)=
0.94-(5/14) * 0.97- (4/14) * 0 -(5/14) * 0.97 = 0.246
⊕ 9
⊖ 5
H(X)=0.940
Temperatura
Calor
Templada
Frio
⊕ 2
⊖ 2
H(X)=1
⊕ 4
⊖ 2
H(X)=0.918
⊕ 3
⊖ 1
H(X)=0.811
Ganancia(X,Temperatura)=
0.94-(4/14) * 1 - (6/14) * 0.918 -(4/14) * 0.811 = 0.02
⊕ 9
⊖ 5
H(X)=0.940
Humedad
Alta
Normal
⊕ 3
⊖ 4
H(X)=0.985
⊕ 6
⊖ 1
H(X)=0.592
Ganancia(X,Humedad)=
0.94-(7/14) * 0.985 - (7/14) * 0.592 = 0.151
⊕ 9
⊖ 5
H(X)=0.940
Viento
Fuerte
Suave
⊕ 6
⊖ 2
H(X)=0.811
⊕ 3
⊖ 3
H(X)=1
Ganancia(X,Viento)=
0.94-(8/14) * 0.811 - (6/14) * 1 = 0.048

ID3: Ejemplo

- Por lo tanto, el atributo que ofrece una mayor ganancia de información es el atributo **Cielo**.
- Utilizando **Cielo** como nodo raíz el árbol inicial quedaría:



ID3: Ejemplo

$$Ganancia(\mathcal{X}, A) = H(\mathcal{X}) - \sum_{v \in \text{valores}(A)} \frac{|\mathcal{X}_v|}{|\mathcal{X}|} H(\mathcal{X}_v)$$

Día	Cielo	Temperatura	Humedad	Viento	Jugar
D1	Soleado	Calor	Alta	Flojo	No
D2	Soleado	Calor	Alta	Fuerte	No
D3	Nublado	Calor	Alta	Flojo	Si
D4	Lluvia	Templado	Alta	Flojo	Si
D5	Lluvia	Frio	Normal	Flojo	Si
D6	Lluvia	Frio	Normal	Fuerte	No
D7	Nublado	Frio	Normal	Fuerte	Si
D8	Soleado	Templado	Alta	Flojo	No
D9	Soleado	Frio	Normal	Flojo	Si
D10	Lluvia	Templado	Normal	Flojo	Si
D11	Soleado	Templado	Normal	Fuerte	Si
D12	Nublado	Templado	Alta	Fuerte	Si
D13	Nublado	Calor	Normal	Flojo	Si
D14	Lluvia	Templado	Alta	Fuerte	No

Ahora habría que repetir el proceso con los nodos correspondientes a los valores **soleado** y **lluvia** (el nodo **nublado** sólo contiene una clase).

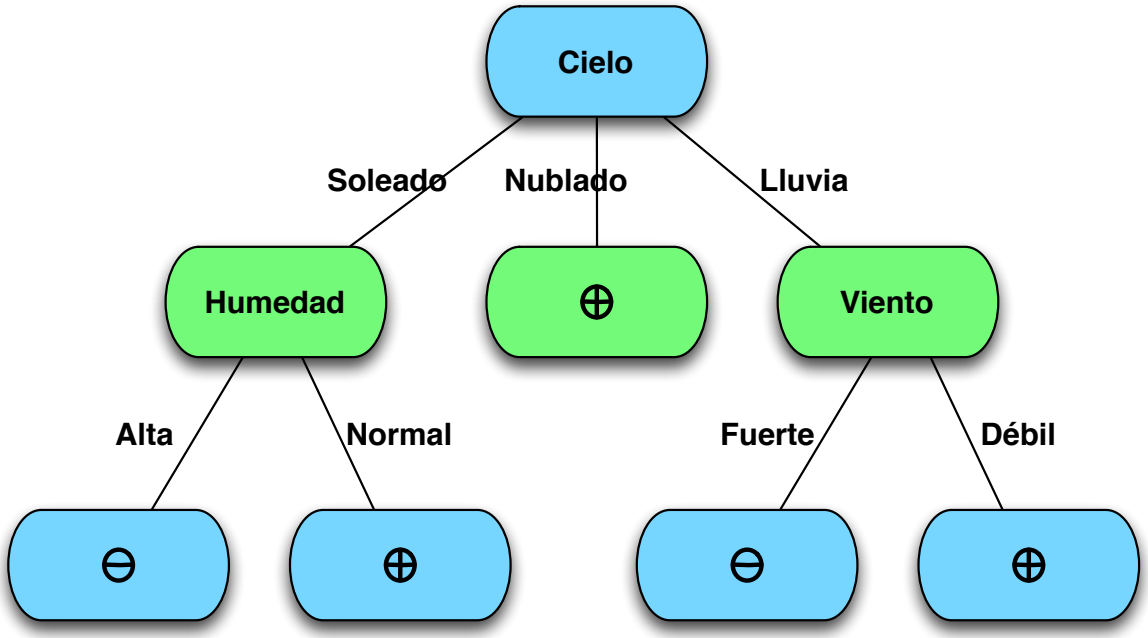
- Para el nodo $Cielo = Soleado$:
  - $\mathcal{X}_{soleado} = \{D_1, D_2, D_8, D_9, D_{11}\}$ con $H(\mathcal{X}_{soleado}) = 0.971$.
  - $Ganancia(\mathcal{X}_{soleado}, Temperatura) = 0.971 - \frac{2}{5} \cdot 0 - \frac{2}{5} \cdot 1 - \frac{1}{5} \cdot 0 = 0.570$
  - $Ganancia(\mathcal{X}_{soleado}, Humedad) = 0.971 - \frac{3}{5} \cdot 0 - \frac{2}{5} \cdot 0 = 0.971$
  - $Ganancia(\mathcal{X}_{soleado}, viento) = 0.971 - \frac{2}{5} \cdot 1 - \frac{3}{5} \cdot 0.918 = 0.019$

ID3: Ejemplo

- Para el nodo $Cielo = Lluvia$:
  - $\mathcal{X}_{lluvia} = \{D_4, D_5, D_6, D_{10}, D_{14}\}$ con $H(\mathcal{X}_{lluvia}) = 0.971$.
  - $Ganancia(\mathcal{X}_{lluvia}, Temperatura) = 0.971 - \frac{3}{5} \cdot 0.918 - \frac{2}{5} \cdot 1 = 0.820$
  - $Ganancia(\mathcal{X}_{lluvia}, Humedad) = 0.971 - \frac{2}{5} \cdot 1 - \frac{3}{5} \cdot 0.918 = 0.820$
  - $Ganancia(\mathcal{X}_{lluvia}, Viento) = 0.971 - \frac{3}{5} \cdot 0 - \frac{2}{5} \cdot 0 = 0.971$

ID3: Ejemplo

Por lo tanto, el árbol resultante sería



Día	Cielo	Temperatura	Humedad	Viento	Jugar
D1	Soleado	Calor	Alta	Flojo	No
D2	Soleado	Calor	Alta	Fuerte	No
D3	Nublado	Calor	Alta	Flojo	Si
D4	Lluvia	Templado	Alta	Flojo	Si
D5	Lluvia	Frio	Normal	Flojo	Si
D6	Lluvia	Frio	Normal	Fuerte	No
D7	Nublado	Frio	Normal	Fuerte	Si
D8	Soleado	Templado	Alta	Flojo	No
D9	Soleado	Frio	Normal	Flojo	Si
D10	Lluvia	Templado	Normal	Flojo	Si
D11	Soleado	Templado	Normal	Fuerte	Si
D12	Nublado	Templado	Alta	Fuerte	Si
D13	Nublado	Calor	Normal	Flojo	Si
D14	Lluvia	Templado	Alta	Fuerte	No

Todos los nodos hoja tienen una entropía nula (solo instancias de una clase).

ID3: Error global

Error global

Es la probabilidad de error, es decir, suma ponderada de los errores de todas las hojas del árbol.

$$E = \sum_{i=1}^{n_h} w_i e_i$$

donde

- n_h es el número de hojas del árbol.
- w_i es el peso o probabilidad de la hoja i, es decir, la probabilidad de que una instancia sea clasificada por la partición representada por la rama que acaba en la hoja i.
- e_i es el error correspondiente a la rama que acaba en la hoja i (número de instancias erróneas que caen en la hoja i entre el número de instancias que caen en la hoja i.)

ID3: Algoritmo

- El algoritmo básico de aprendizaje es el ID3 (Iterative Dichotomiser 3).

Algoritmo arbol $\leftarrow$ aprenderArbol($datos$)

si todos los ejemplos en $datos$ tienen la misma etiqueta **entonces**
  devolver un nodo hoja con dicha etiqueta.
sino
  Sea A el atributo que clasifica mejor a los objetos en $datos$.
  para todo posible valor v de A **hacer**
    $data(v) \leftarrow$ todos los objetos con $A = v$.
    Añadir nueva rama $\leftarrow$ aprenderArbol($data(v)$).
  fin para
  devolver árbol.
fin si

ID3: Algoritmo

ID3(Instancias, Etiquetas, Atributos)

Entrada: `Instancias`: el conjunto de datos.
Entrada: `Etiquetas`: el conjunto de posibles clases.
Entrada: `Atributos`: el conjunto de atributos en el conjunto `Instancias`.

si todos las instancias son positivos **entonces**
  devolver el nodo raíz con etiqueta $+$.
sino, **si** todos las instancias son negativos **entonces**
  devolver el nodo raíz con etiqueta -.
sino, **si** `Atributos`$=\emptyset$ **entonces**
  devolver el nodo raíz con el valor de `Etiquetas` más probable en `Instancias`.
fin si
Sea A el atributo que clasifica mejor a las instancias en $datos$.
Crear un árbol con un nodo etiquetado con A.
para todo posible valor v_i del atributo A **hacer**
  añadir un arco bajo la raíz con la comprobación $A = v_i$.
  sea `Instancias` $_{v_i}$ el subconjunto de `Instancias` con $A = v_i$.
  si `Instancias`$_{v_i} = \emptyset$ **entonces**
    añadir un nodo hoja al arco añadido con el valor de `Etiquetas` más probable en `Ejemplos`.
  sino
    añadir al nuevo arco el subárbol generado por **ID3(**`Instancias`$_{v_i}$**,** `Etiquetas`**,**`Atributos-`$\{A\}$**)**.
  fin si
fin para
devolver nodo raíz.

Espacio de hipótesis y sobre-ajuste

El **espacio de hipótesis** H (no confundir con la entropía) en árboles de decisión abarca todas las posibles combinaciones de atributos y valores que pueden formar árboles de decisión, y nuestra tarea es encontrar la hipótesis más adecuada para clasificar correctamente las instancias de entrada.

En general, se prefieren hipótesis cortas para evitar el sobre-ajuste (**overfitting**).

Sobre-ajuste

Dado un espacio de hipótesis H, se dice que una hipótesis particular $h \in H$ sobreajusta los datos de entrenamiento si existe una hipótesis alternativa $h' \in H$, tal que h presenta un error menor que h' sobre los ejemplos de entrenamiento, pero h' presenta un error menor que h sobre el conjunto total de observaciones.

Proceso de poda

¿Cómo podemos evitar el sobre-aprendizaje o sobre-ajuste?

Poda: Eliminar condiciones de las ramas del árbol encontrar modelos más pequeños. Existen dos tipos de poda:

- **Prepoda:** El proceso se realiza durante la construcción del árbol, estableciendo un criterio de parada
  - El número de instancias por nodo.
  - El error esperado.
  - MDL (Minimun Description Lenght).
- **Postpoda:** El proceso se realiza después de la construcción del árbol.
  - Consiste en ir el eliminado nodos de abajo a arriba mientras se vaya cumpliendo un criterio determinado.

Se puden combinar ambas aproximaciones.

Otras medidas

Medidas alternativas: En algunos casos se suele utilizar otras medidas como

- El $Ratio$ de la ganancia de información, para evitar el hecho de que se favorezca la selección de los atributos con más valores.
- MSE para regresión
- Índice Gini empleado por el algoritmo CART
- DKM, basados en AUC, MDL,...

CART

- **CART** (**C**lassification **A**nd **R**egression **T**rees). Similar al ID3 pero:
  - Permite que la variable que define la clase sea continua y no construye un conjunto de reglas.
  - Utiliza el índice de Gini en vez de la ganancia de información para seleccionar el mejor atributo (la mejor partición).
  - Utiliza también el esquema de partición recursiva utilizando una estrategia voraz.
  - También permite resolver problemas de regresión.

CART

- Se elige la partición que produce el menor valor de la función de coste.
  - Para regresión: RSME.
  - Para clasificación: GINI

$$Gini(p) = \sum_{i=1}^{n} p_i(1 - p_i)$$

con p_i las proporciones de instancias de la clase i en la partición

- Prepoda: Utiliza como criterio de parada el número mínimo de instancias asignadas al nodo.
- Postpoda: utiliza un criterio que controla la importancia relativa del error frente la complejidad (tamaño del árbol).

C4.5, C50

- **C4.5**: Permite, a diferencia que ID3, que las características puedan ser continuas, definiendo de forma dinámica un atributo discreto particionando los atributos continuos en un conjunto discreto de intervalos.
  - C4.5 transforma los árboles obtenidos en un conjunto de reglas del tipo *if-then*. La precisión de cada regla es evaluada de forma independiente para determinar el orden en el que deben ser aplicadas.
  - Un proceso de poda elimina antecedentes de las reglas si con esto se mejora la precisión de la misma.
  - C5.0, utiliza menos memoria y obtiene un conjunto de reglas menor y más preciso.
  - J48 es la implementación en código abierto de C4.5

Random Forests

- Random Forest es una técnica que construye un gran número de árboles de decisión no correlacionados.
- Se basa en la técnica de Agregación de Bootstrap (Bagging), tecnica de agregación de clasificadores o regresores que:
  - Aumenta la precisión y estabilidad reduciendo los efectos del ruido.
  - Reduce la varianza en las predicciones
  - Ayuda a evitar el sobre ajuste.
- La predicción del modelo se elige analizando las predicciones de cada uno de los árboles considerados en el modelo.

Random Forests: Algoritmo

Algoritmo RF $\leftarrow$ RandomForest($datos$)

1: **para** $i \leftarrow 1$ to $n\_arboles$ **hacer**
2: Extraer una muestra de tamaño $size(data)$ de $datos$ por bootstrapping
3: Construir un árbol T_i repitiendo recursivamente para cada nodo hoja
4: 1. Seleccionar aleatoriamente m atributos.
5: 2. Seleccionar la mejor partición de las inducidas por los m atributos.
6: 3. Dividir el nodo en dos nodos hijos.
7: 4. Si el tamaño del nodo alcanza n_{min} no continuar dividiendo.
8: **fin para**
9: **devolver** El conjunto de árboles $\{T_i\}_1^{n\_arboles}$.

Random Forests: Algoritmo

- Si tenemos p atributos las recomendaciones para el valor de m son:
  - Clasificación: $\lfloor \sqrt{m} \rfloor$
  - Regresión: $\lfloor p/3 \rfloor$
  - Siendo el valor mínimo 1.
- Una vez se han construido los árboles para hacer predicciones:
  - Clasificación: La clase más votada por los árboles.
  - Regresión: La media de todas las predicciones.

Random Forests: OBB e importancia de las variables.

- **OBB (out of the bag) error estimate**: para cada muestra se predice el error utilizando sólo los árboles en los que no ha sido utilizada (no ha sido elegida en el bootstrapping)
  - Los valores son parecidos a los que se obtienen mediante una validación cruzada de N-pliegues.
- **Importancia de los atributos**:
  - En cada partición del árbol se registra la mejora del criterio de división.
  - Este valor se considera la importancia del atributo
  - Para cada variable se agregan los valores generados en cada árbol.

Conclusiones

- La utilización de árboles de decisión es muy popular en muchas disciplinas.
- Experimentalmente han demostrado una buena capacidad de clasificación.
- La clave reside en la función a optimizar a la hora de elegir el atributo para disgregar el árbol.
- Los ensambles nos permiten mejorar los resultados combinando información.

Bibliografía I

- José Hernández Orallo, M.José Ramírez Quintana, Cèsar Ferri Ramírez, 2004. Introducción a la Minería de Datos. Pearson education. Editorial Pearson, 2004. ISBN: 84 205 4091 9
- Jiawei Han, Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman Publishers, 2003.
- Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006
- L. A. Breslow and D. W. Aha. *Simplifying decision trees: a survey*. Knowledge Engineering Review, 12(1):1–40, 1997.

Bibliografía II

- Sheerama K. Murthy. *Automatic construction of decision trees from data: A multi-disciplinary survey*. Data Mining and Knowledge Discovery, 1997.
- Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press 2004.
- José T. Palma y Roque Marín (eds.). *Inteligencia Artificial: Métodos, Técnicas y Aplicaciones*. McGraw-Hill, 2008