



# Curso DM

## Clasificación

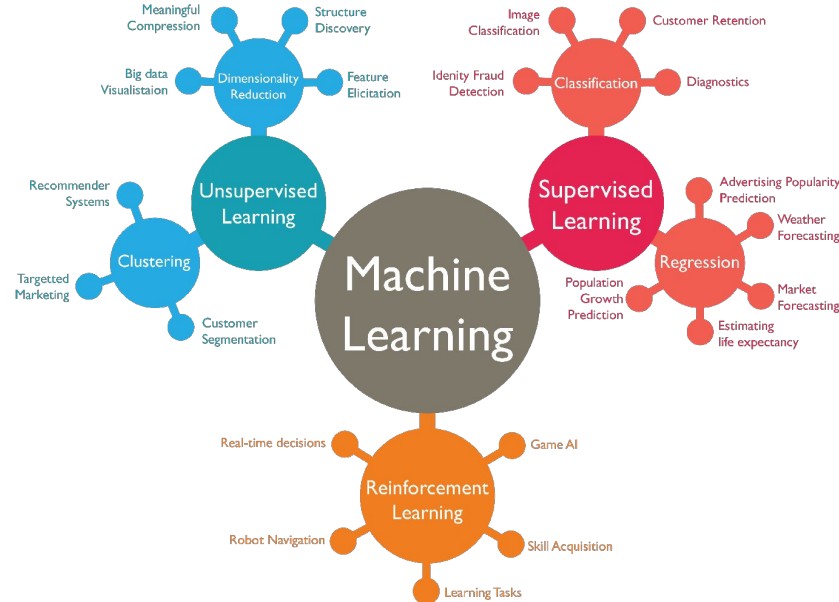
Algoritmos (Árboles, KNN, Naive Bayes)

Primavera 2023

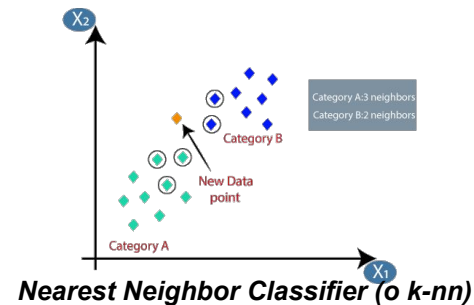
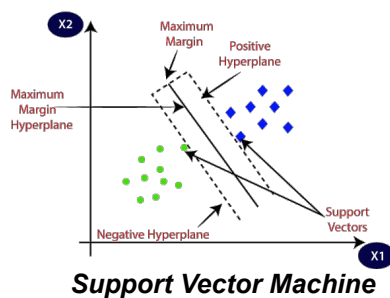
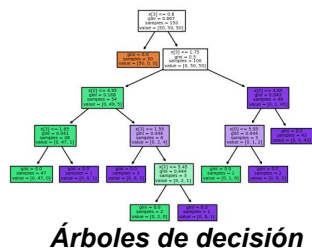
Basado en las slides de Bárbara Poblete

# Algoritmos de aprendizaje de máquinas

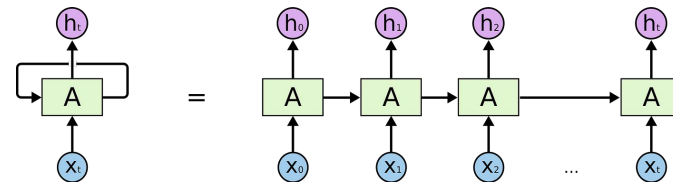
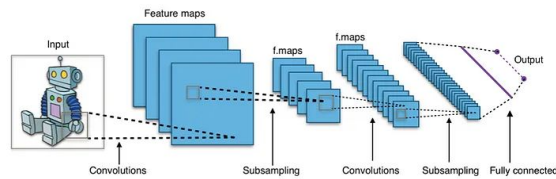
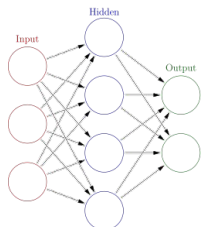
Algoritmos que pueden aprender de forma automática a través de datos. Esperamos que la máquina desarrolle su propia representación.



Algoritmos tradicionales. Ej:



Algoritmos basados en redes neuronales. Ej:

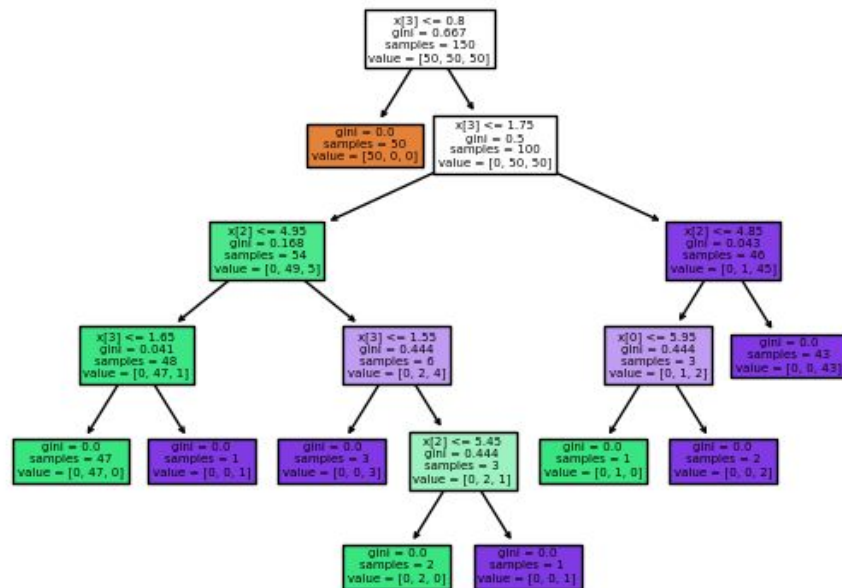


# Árbol de Decisión

Decision tree trained on all the iris features

El árbol tiene tres tipos de nodos:

1. Un **nodo raíz** que no tiene arcos entrantes y tiene arcos salientes.
2. **Nodos internos**, cada uno de los cuales tiene exactamente un arco entrante y dos o más arcos salientes.
3. **Nodos hoja o terminales**, cada uno de los cuales tiene exactamente un arco entrante.



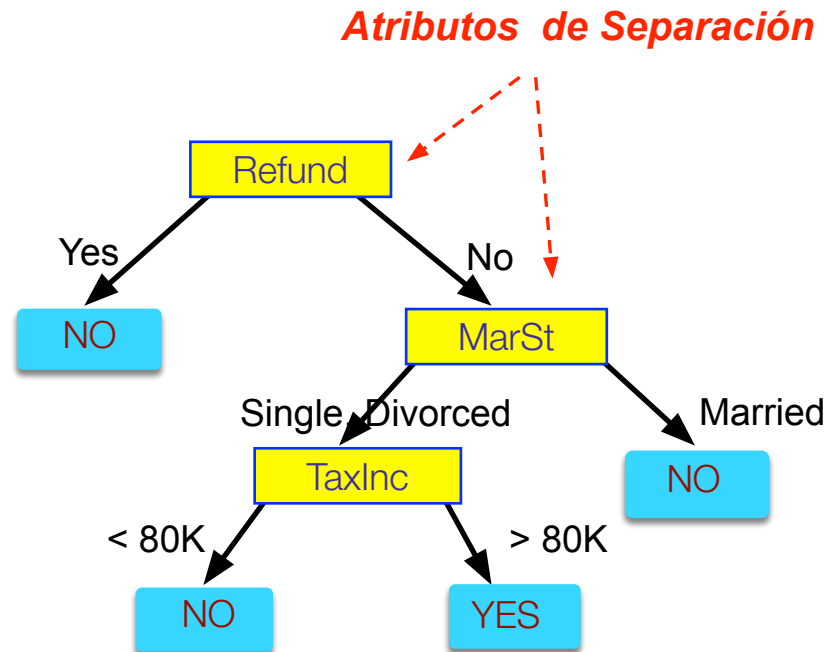
# Árbol de Decisión

- A cada nodo de hoja se le asigna una etiqueta de clase.
- Los nodos no terminales, que incluyen la raíz y otros nodos internos, contienen tests sobre los atributos para separar los ejemplos que tienen valores diferentes para esos atributos.
- El árbol de decisión fragmenta el dataset de manera recursiva hasta asignar los ejemplos a una clase.

# Árbol de Decisión

<i>Tid</i>	<i>Refund</i>	<i>Marital Status</i>	<i>Taxable Income</i>	<i>Cheat</i>
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Datos de Entrenamiento

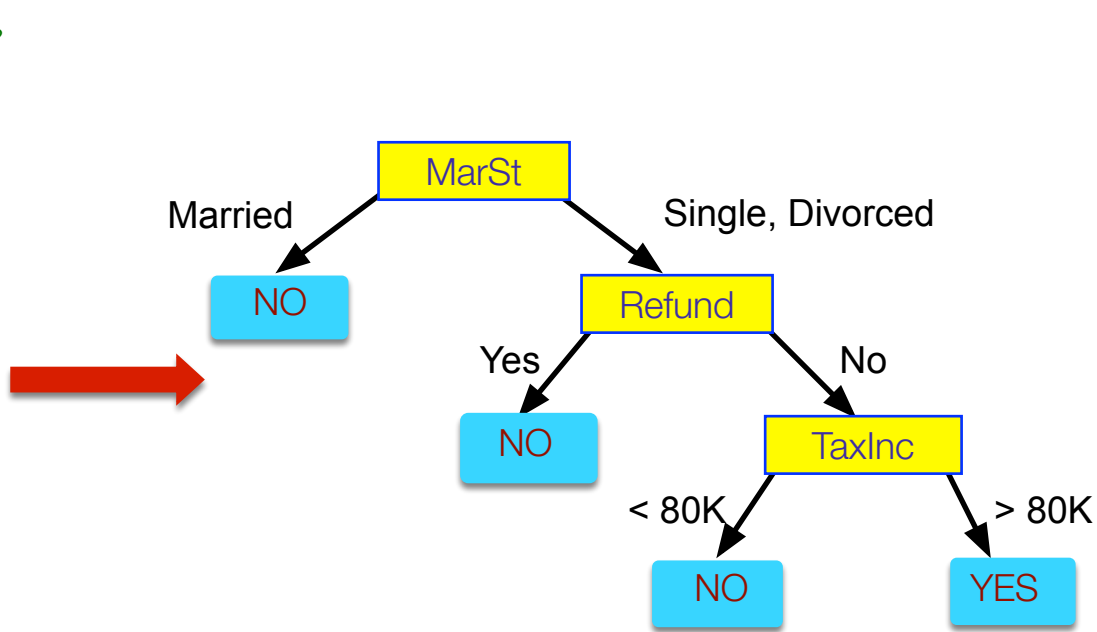


Modelo: Árbol de Decisión

# Árbol de Decisión

<i>Tid</i>	<i>Refund</i>	<i>Marital Status</i>	<i>Taxable Income</i>	<i>Cheat</i>
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Datos de Entrenamiento



¡Puede existir más de un árbol que se ajuste a los datos!

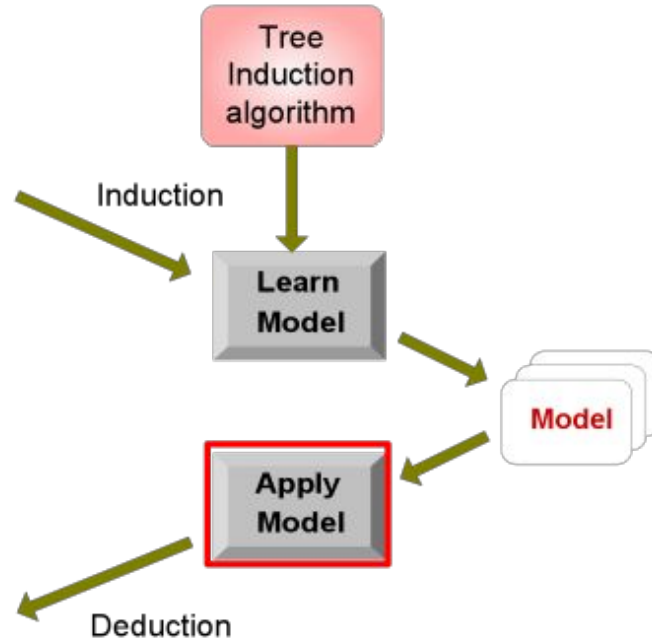
# Clasificando con un árbol de decisión

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

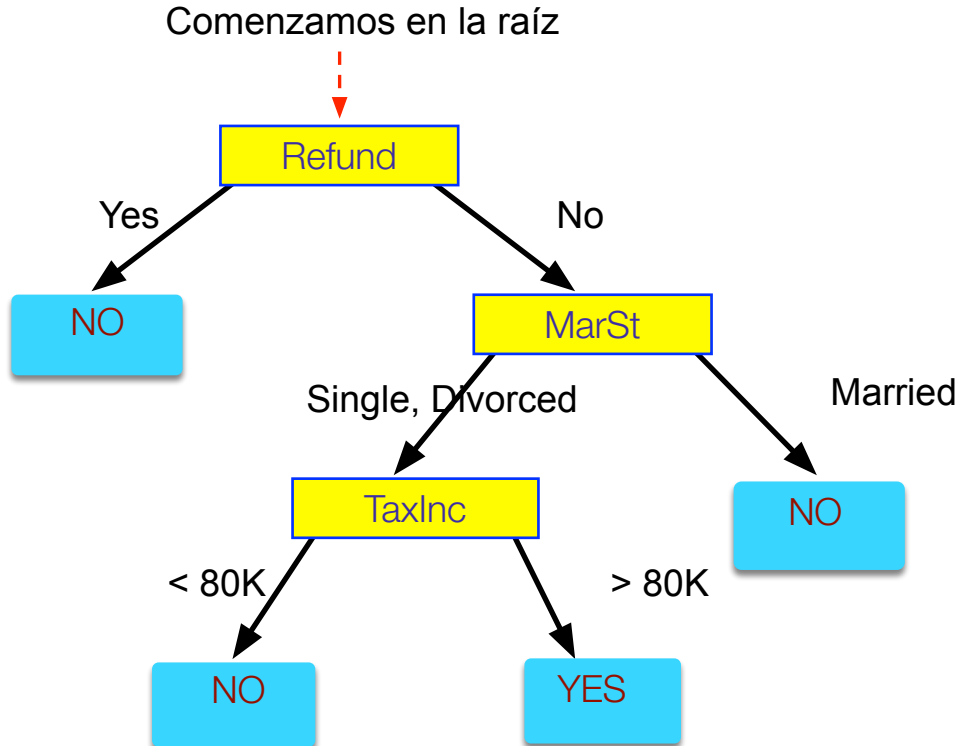




# Aplicamos el modelo

## Dato de Evaluación

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Fáciles de interpretar  
y visualizar

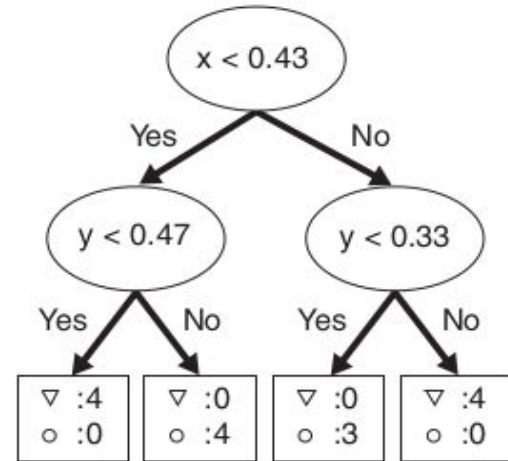
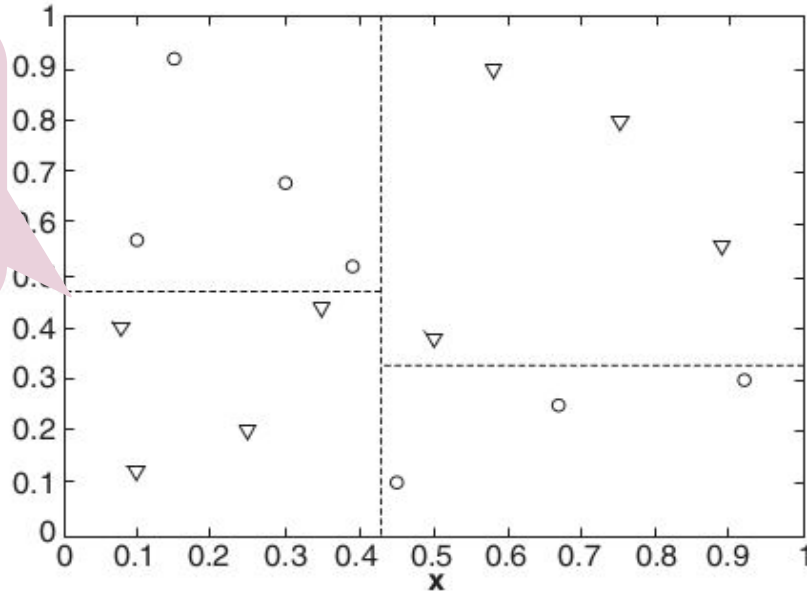
# Construyendo un Árbol de Decisión

## **Estrategia: Top down (greedy) - Divide y vencerás recursiva**

- Primero: seleccionar un atributo para el nodo raíz y crear rama para cada valor posible del atributo .
- Luego: dividir las instancias del dataset en subconjuntos, uno para cada rama que se extiende desde el nodo.
- Por último: repetir de forma recursiva para cada rama, utilizando sólo las instancias que llegan a ésta.
- Detenerse cuando todas las instancias del nodo sean de la misma clase.

# Un árbol de decisión hace cortes perpendiculares a los ejes

Encontrar cortes que crean regiones puras (con ejemplos de una misma clase).  
-> Necesitamos medir la pureza



**Figure 3.20.** Example of a decision tree and its decision boundaries for a two-dimensional data set.

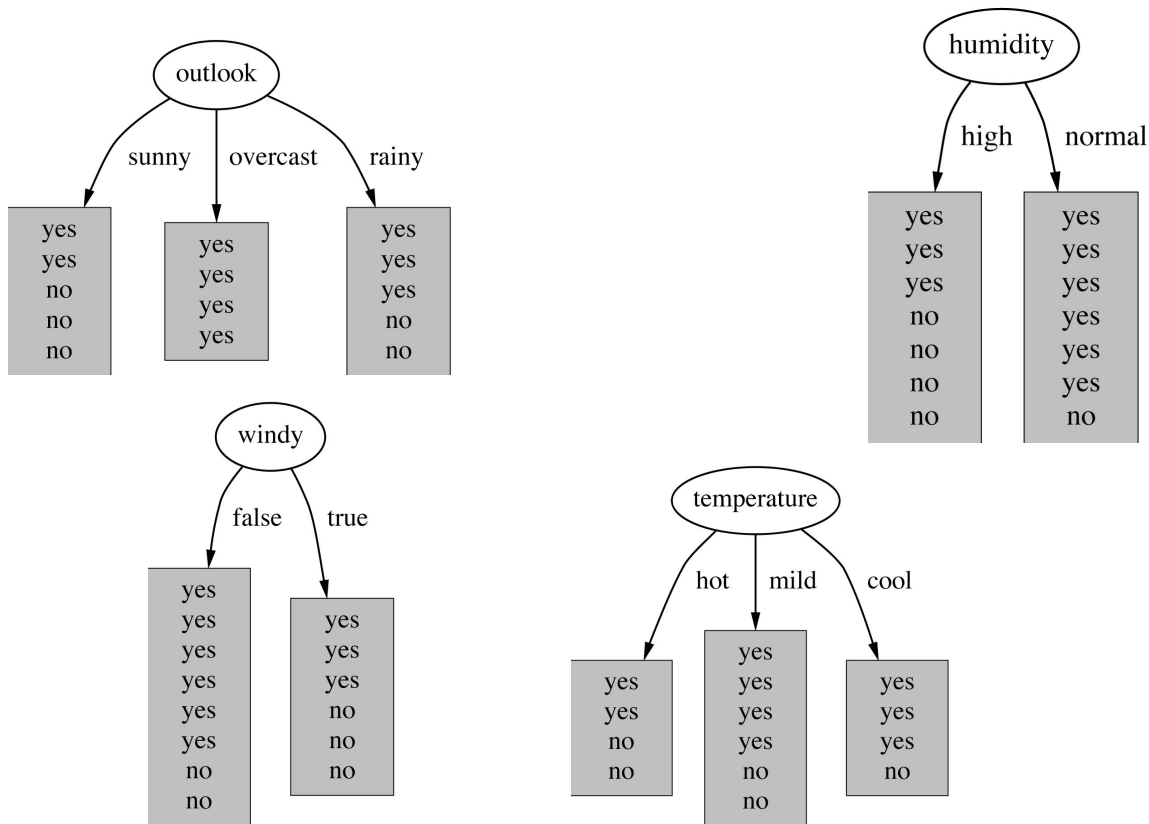
# El dataset Weather

Condiciones para salir a jugar tenis

**Table 4.6 The weather data with identification codes.**

ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

# ¿Cómo escoger atributos?



# Criterio para escoger el mejor atributo

## ¿Qué atributo escojo?

- La idea es crear el árbol más pequeño posible.
- Heurística: escoge el atributo que produce nodos lo más “puros” posible.


El criterio más popular de pureza: **information gain**

- Information gain crece cuando crece la pureza promedio de los subconjuntos.

**Estrategia:** escoger el atributo que maximiza el valor de information gain.

# Criterio para escoger el mejor atributo

**Information gain:** Información antes del split – información después del split

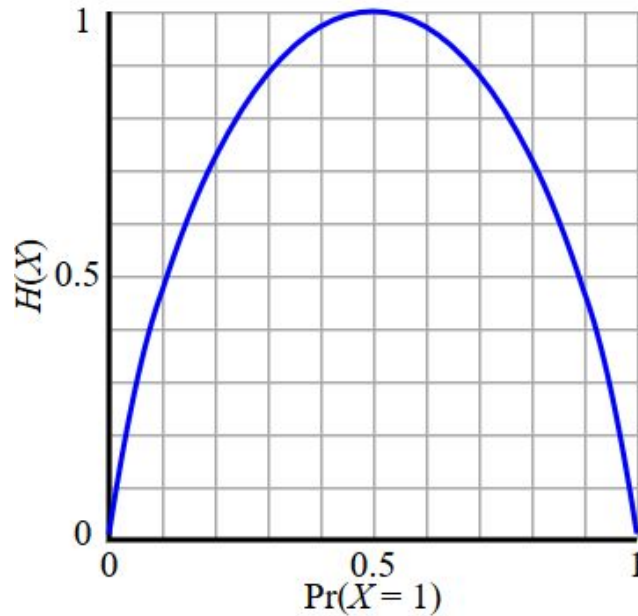
$$Gain(S, D) = \boxed{H(S)} - \sum_{V \in D} \frac{|V|}{|S|} H(V)$$


**Entropía:** información promedio requerida para codificar un evento dado una distribución de probabilidad (viene de la teoría de información de Claude Shannon). Nos entrega la información esperada en bits.

$$entropy(p_1, p_2, \dots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \dots - p_n \log_2 p_n$$

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i)$$

# Entropía para dos Clases con distintas Proporciones

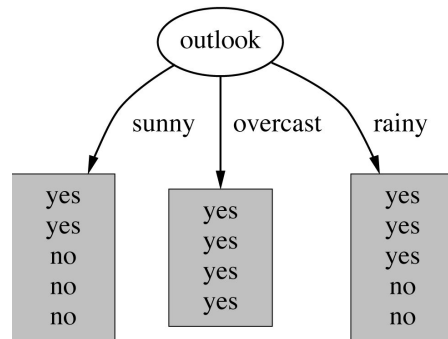


La entropía toma su máximo valor cuando  $p=0.5$  (máxima incerteza).



# Computando la Información

## Ejemplo: atributo outlook



### Outlook = Sunny:

$$\text{info}([2,3]) = \text{entropy}(2/5, 3/5) = -2/5 \log(2/5) - 3/5 \log(3/5) = 0.971 \text{ bits}$$

### Outlook = Overcast:

$$\text{info}([4,0]) = \text{entropy}(1,0) = -1 \log(1) - 0 \log(0) = 0 \text{ bits}$$

*(Nota: esto normalmente queda indefinido)*

**Entropía mínima -> región pura**

### Outlook = Rainy:

$$\text{info}([3,2]) = \text{entropy}(3/5, 2/5) = -3/5 \log(3/5) - 2/5 \log(2/5) = 0.971 \text{ bits}$$

### Información esperada para el atributo

$$\text{info}([2,3], [4,0], [3,2]) = (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 = 0.693 \text{ bits}$$

# Criterio para escoger el mejor atributo

**Information gain:** Información antes del split – información después del split

$$Gain(S, D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V)$$

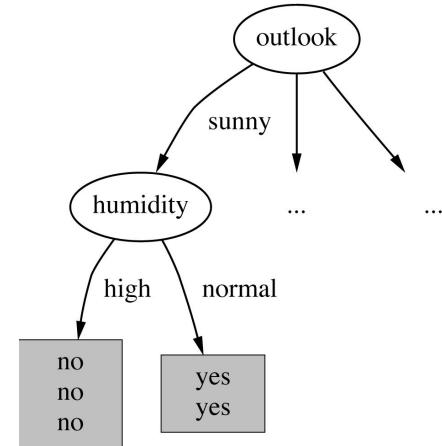
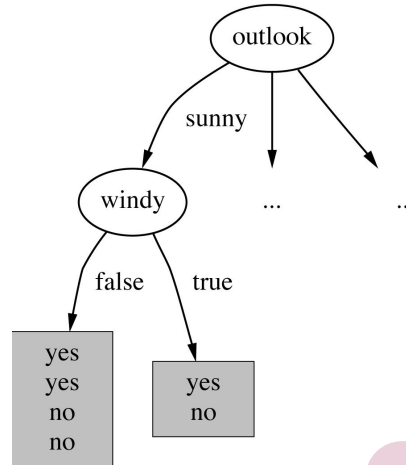
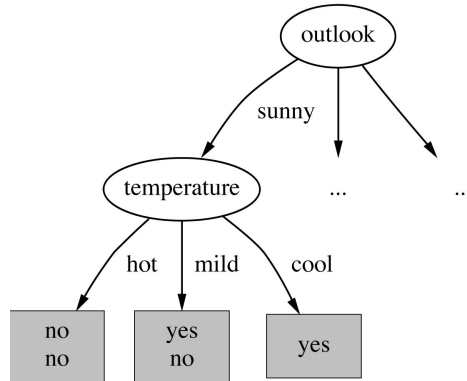
$$\begin{aligned} \text{gain(Outlook)} &= \text{info}([9,5]) - \text{info}([2,3],[4,0],[3,2]) \\ &= 0.940 - 0.693 \\ &= 0.247 \text{ bits} \end{aligned}$$

Information gain para los atributos de los datos de weather:

$$\begin{aligned} \text{gain(Outlook)} &= 0.247 \text{ bits} \\ \text{gain(Temperature)} &= 0.029 \text{ bits} \\ \text{gain(Humidity)} &= 0.152 \text{ bits} \\ \text{gain(Windy)} &= 0.048 \text{ bits} \end{aligned}$$

Outlook es el mejor atributo. Temperature y Windy tienen poca ganancia de información.

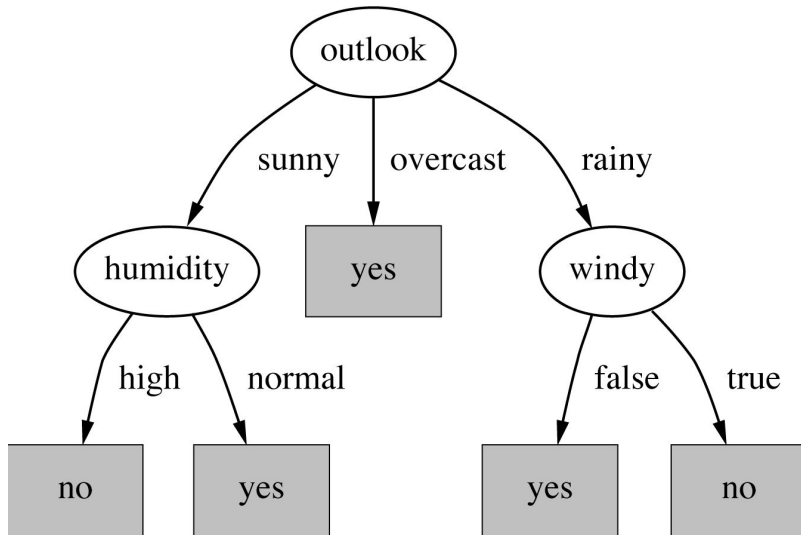
# Seguimos particionando



gain(Temperature) = 0.571 bits  
gain(Humidity) = 0.971 bits  
gain(Windy) = 0.020 bits

Humidity es el mejor atributo porque llegó a regiones (más) puras.

# Árbol de Decisión Resultante



**Nota:** no todas las hojas tienen que ser puras; a veces instancias idénticas tienen clases diferentes.

→ El splitting termina cuando los datos no se pueden seguir particionando.

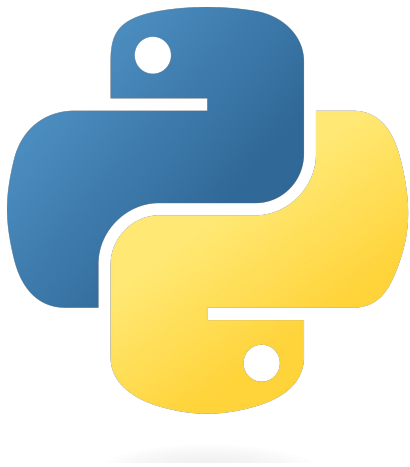
Se puede exigir un mínimo número de instancias en la hoja para evitar sobreajuste.

Puede predecir probabilidades usando las frecuencias relativas de las clases en la hoja.

# Comentarios

- Information gain tiende a favorecer atributos de muchas categorías por su capacidad de fragmentar el dataset en muchas bifurcaciones. Una solución es usar una métrica llamada **Gain ratio**.
- Gain ratio toma en cuenta el número y el tamaño de las ramas (respecto a la cantidad de ejemplos que alcanzan) al elegir un atributo.
- Los atributos numéricos son discretizados, escogiendo la partición que maximice information gain (o gain ratio).
- Existen otras métricas para medir pureza distintas a entropía como el índice de **Gini**  $= 1 - \Pr$  (Sacar dos ejemplos de la misma clase).
- Para evitar sobre-ajuste los árboles pueden ser podados (se eliminan ramas que alcanzan muy pocos ejemplos).
- La gran ventaja de los árboles es la **interpretabilidad**.

# Parte práctica



Veremos:

- Carga de datos
- Partición de datos en train y test
- Entrenamiento y evaluación del modelo (árbol de decisión)
  - Holdout
  - Cross-validation
- Visualización de matriz de confusión
- Seleccionar hiperparámetros
- Recomendaciones de lectura (baselines, codificación de atributos categóricos, balanceo de datos)

<https://colab.research.google.com/drive/1ERkAyYTIYa7BWTd20RbiI52CtgCz1YPQ?usp=sharing>



**dcc**

CIENCIAS DE LA COMPUTACIÓN  
UNIVERSIDAD DE CHILE

[www.dcc.uchile.cl](http://www.dcc.uchile.cl)