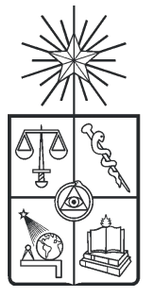


dcc
CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Clasificación I

Árboles y KNN

Felipe Bravo
(Basado en una versión previa de Bárbara Poblete)

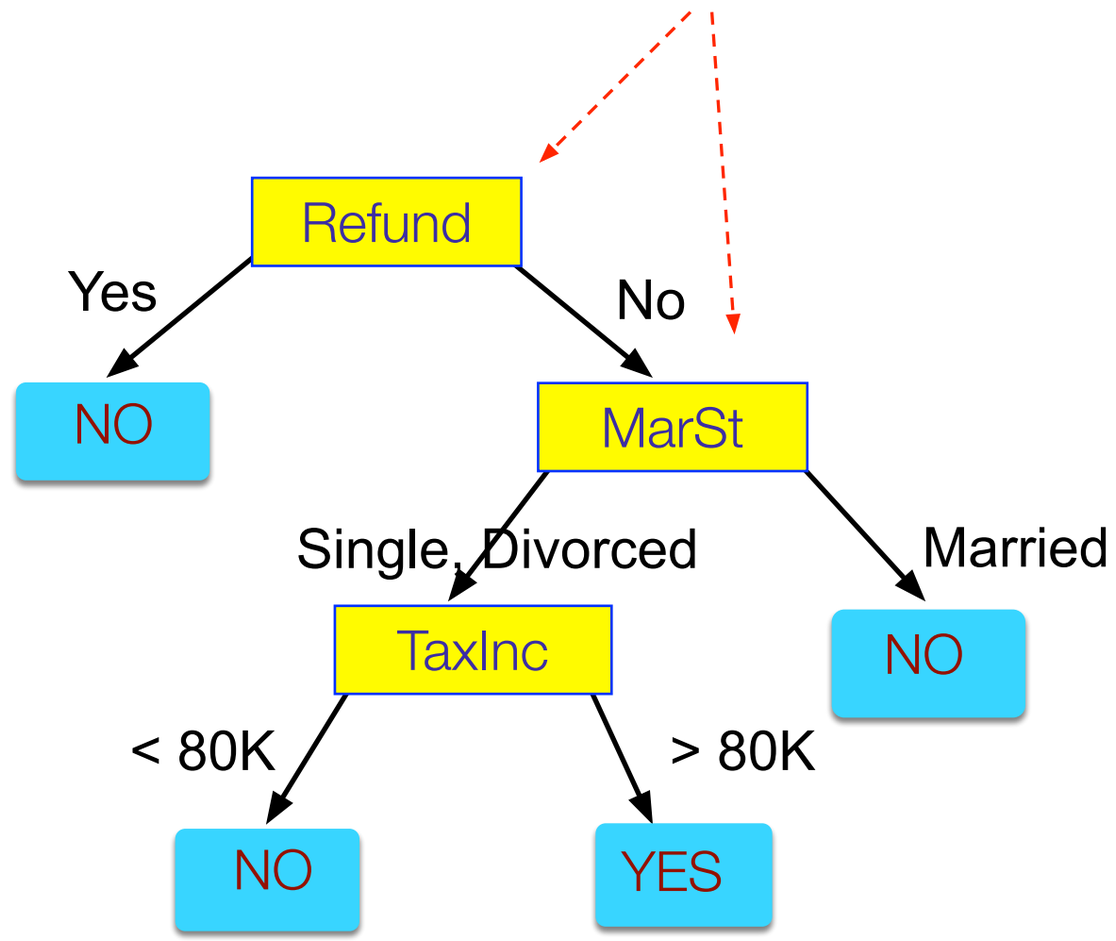
Árbol de Decisión

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
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

Atributos de Separación



Modelo: Árbol de Decisión

Árbol de Decisión

El árbol tiene tres tipos de nodos:

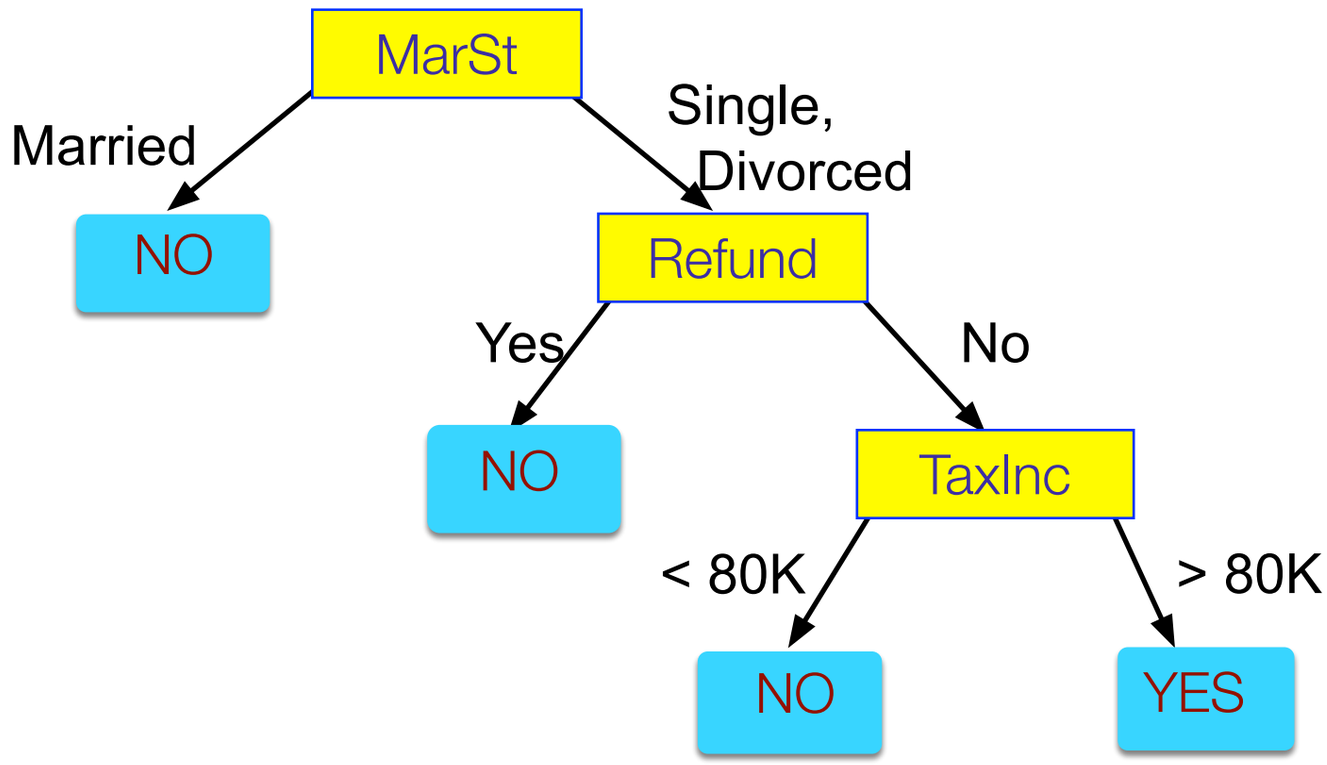
1. Un nodo raíz que no tiene arcos entrantes y tiene arcos salientes.
2. Nodos internos, cada uno de los cuales tiene exactamente un arco entrante y dos o más arcos salientes.
3. Nodos hoja o terminales, cada uno de los cuales tiene exactamente un arco entrante.

- 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.

Otro Ejemplo

Tid	Refund	Marital Status	Taxable Income	Cheat
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

categorical categorical continuous class

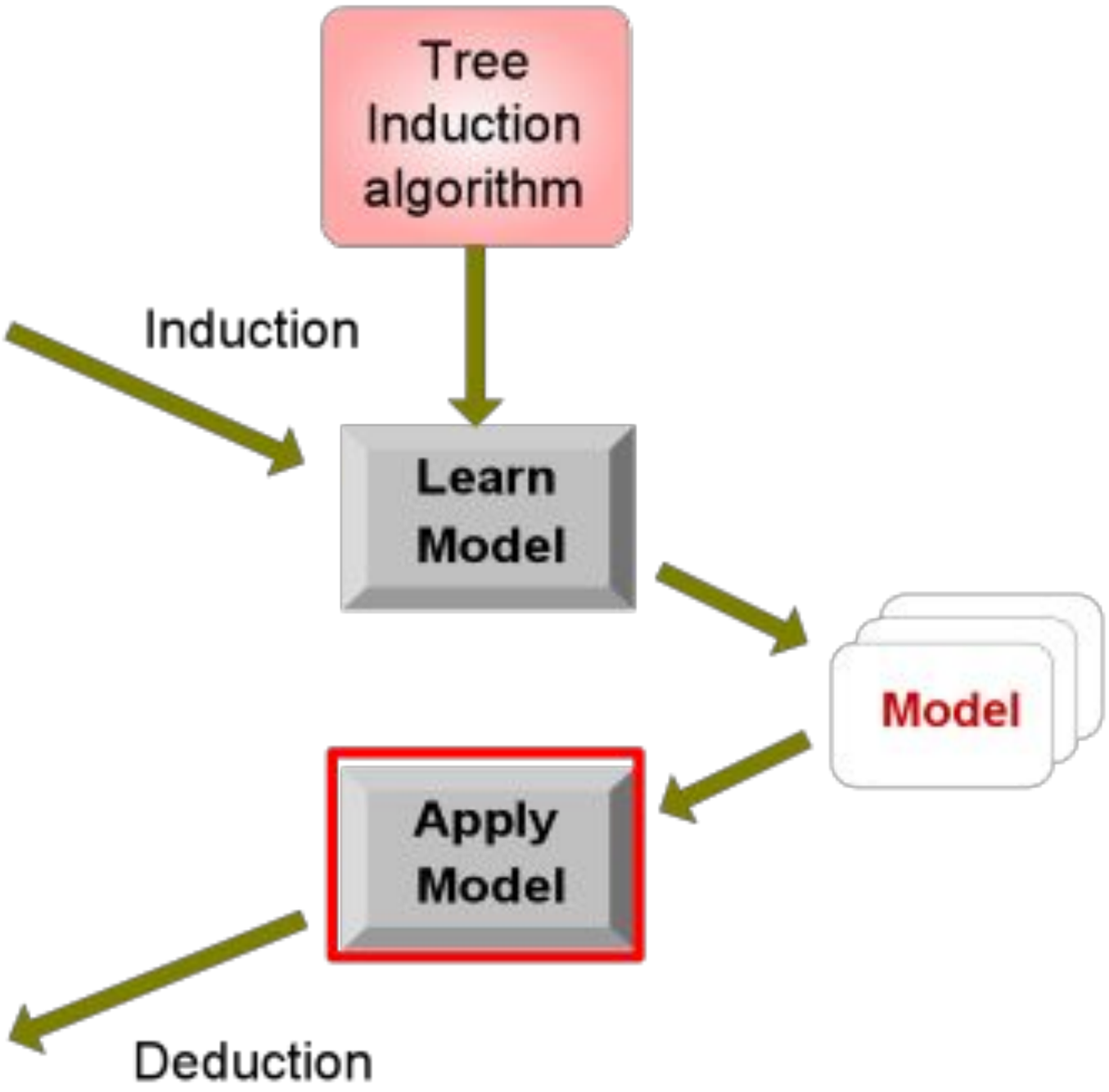


¡Puede existir más de un árbol que se ajuste a los datos!

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

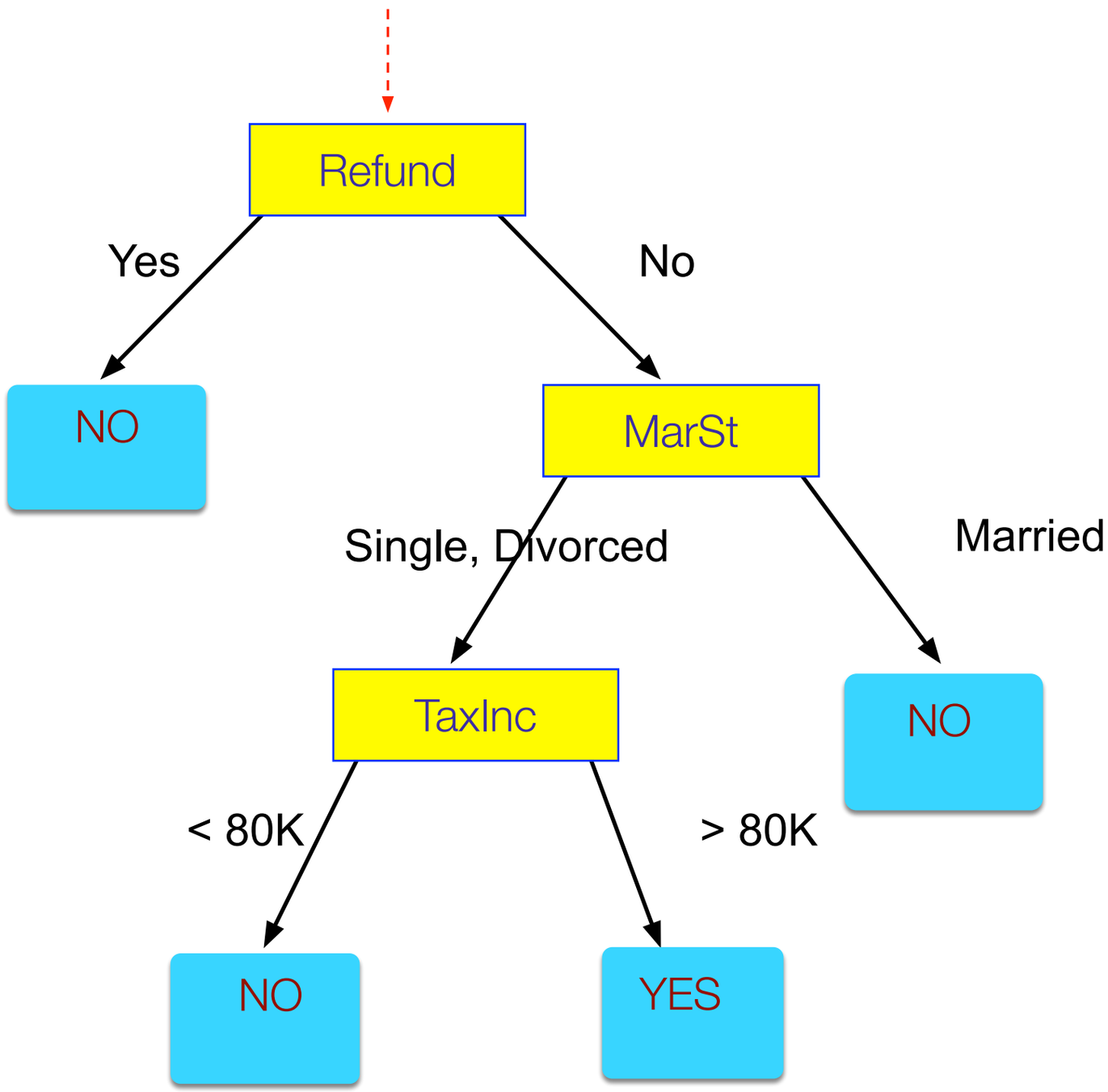


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Aplicamos el modelo

Comenzamos en la raíz



Dato de Evaluación

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Clasificando con un árbol de decisión

- Muchos algoritmos
  - CART
  - ID3, C4.5 **(J48 en Weka)**
  - SLIQ, SPRINT

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

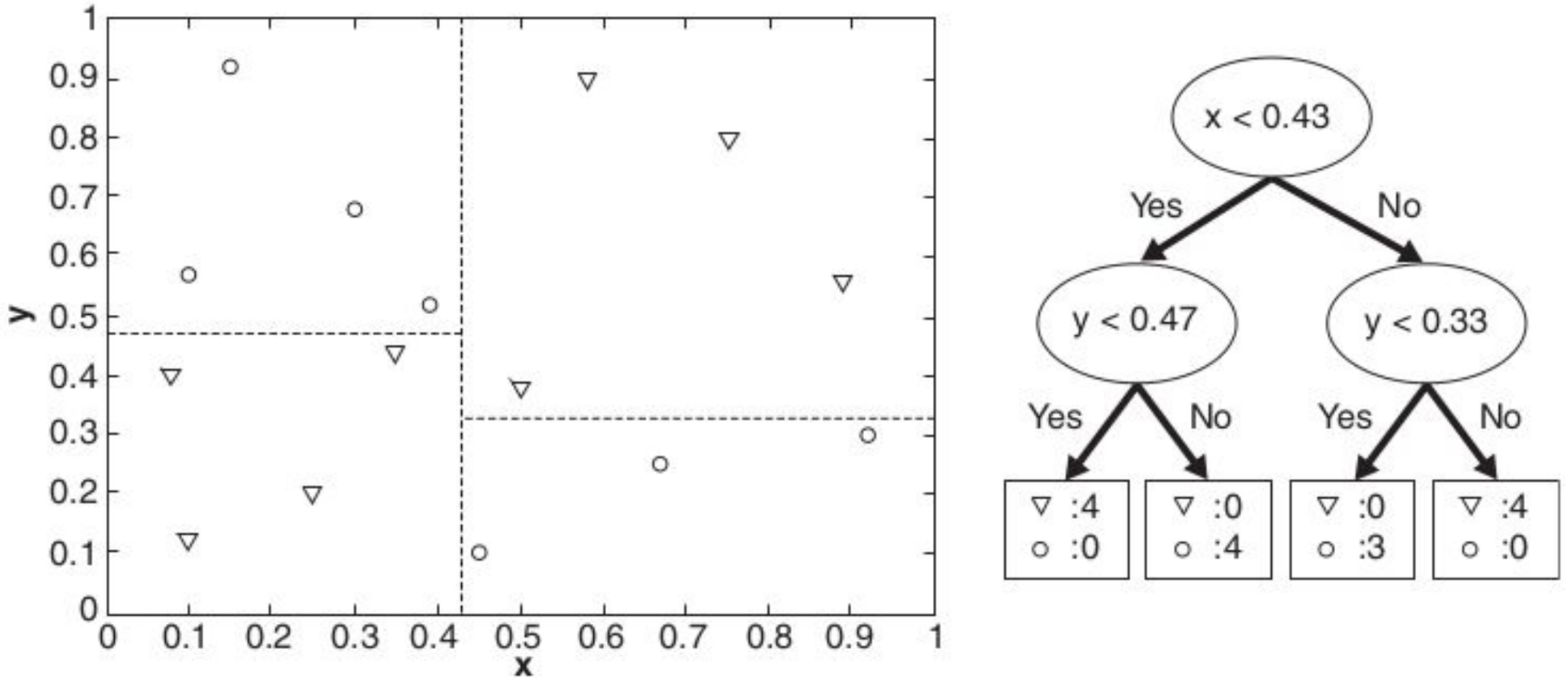


Figure 3.20. Example of a decision tree and its decision boundaries for a two-dimensional data set.

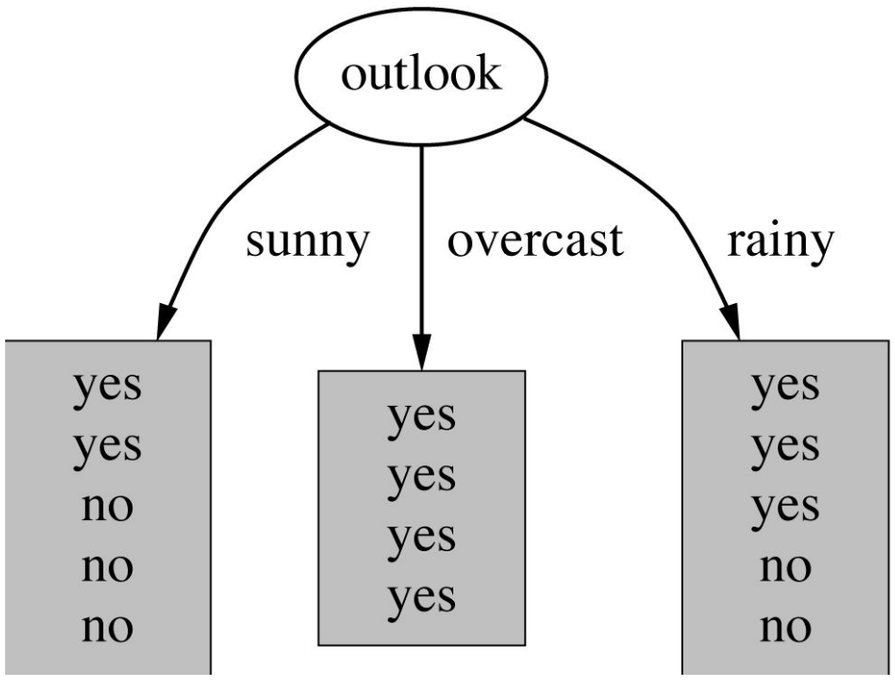
El dataset Weather

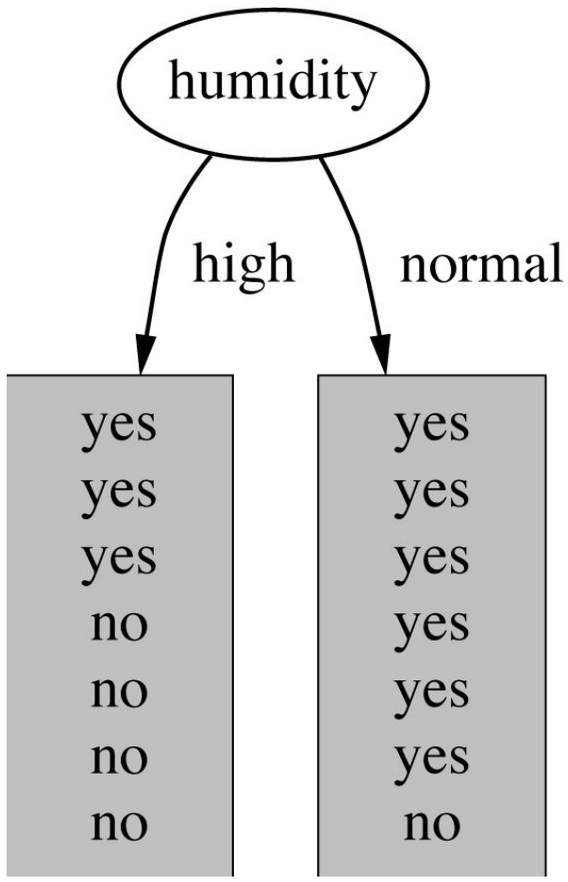
Condiciones para salir a jugar tenis:

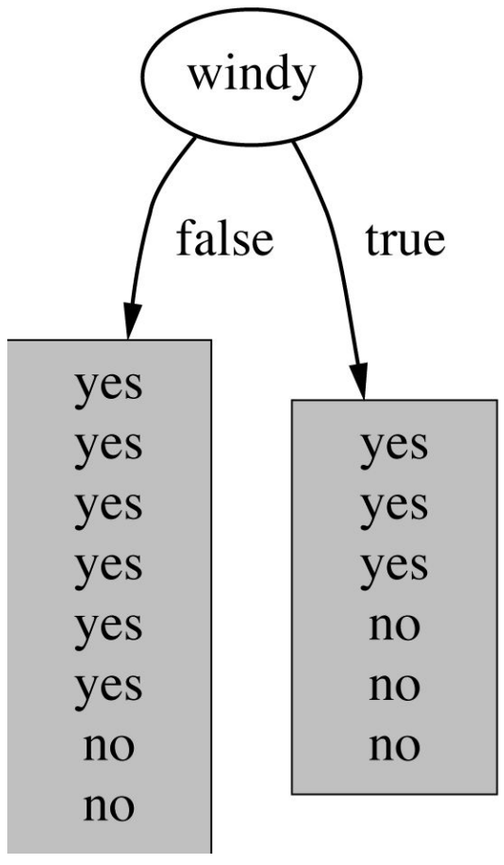
Table 4.6 The weather data with identification codes.

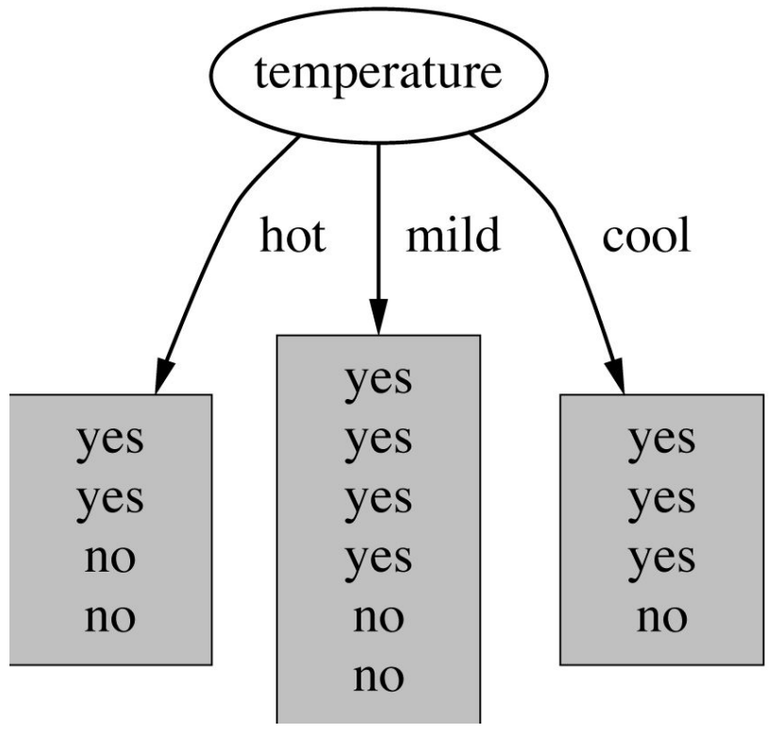
ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes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

Computando la Información

La información se puede medir en **bits**

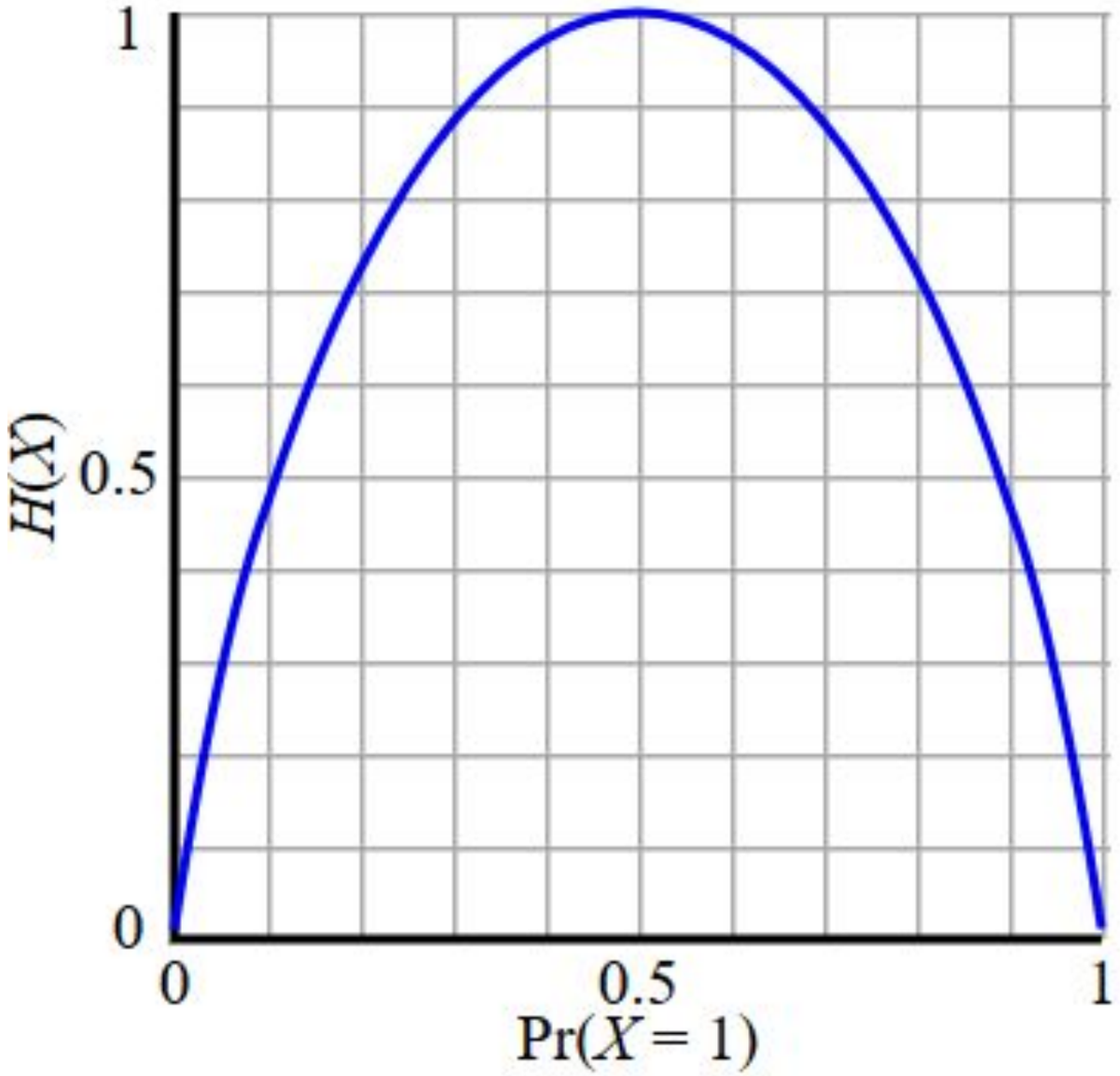
- **Entropía**: información promedio requerida para codificar un evento dado una distribución de probabilidad (viene de la teoría de información de Claude Shannon).
- La entropía nos entrega la información esperada en bits (puede ser una fracción).

Fórmula para calcular la entropía:

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

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

Entropía para dos Clases con distintas Proporciones



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

Ejemplo: atributo *outlook*

- *Outlook* = *Sunny* :

info([2,3]) = entropy(2/5,3/5)= -2/5 log (2/5) - 3/5 log (3/5) = 0.971 bits

- *Outlook* = *Overcast* :

info([4,0]) = entropy(1,0) = -1 log (1) - 0 log (0) = 0 bits

Nota: esto normalmente queda indefinido

- *Outlook* = *Rainy* :

info([2,3]) = entropy(3/5,2/5)=-3/5 log (3/5) - 2/5 log (2/5) = 0.971 bits

- Información esperada para el atributo

info([3,2],[4,0],[3,2]) = (5/14) x 0.971 + (4/14) x 0 + (5/14) x 0.971 = 0.693 bits

Calculando information gain

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

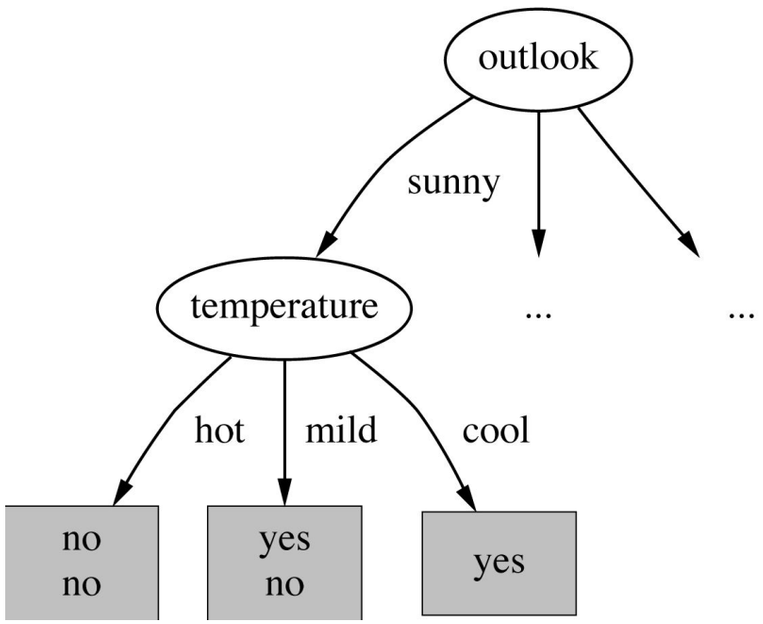
$$Gain(S, D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V)$$

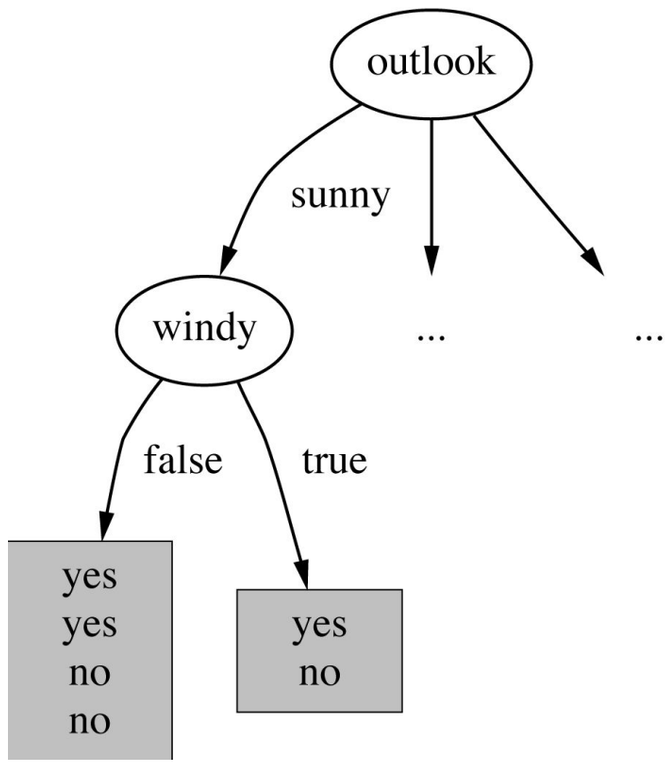
gain(*Outlook*) = info([9,5]) – info([2,3],[4,0],[3,2])
= 0.940 – 0.693
= 0.247 bits

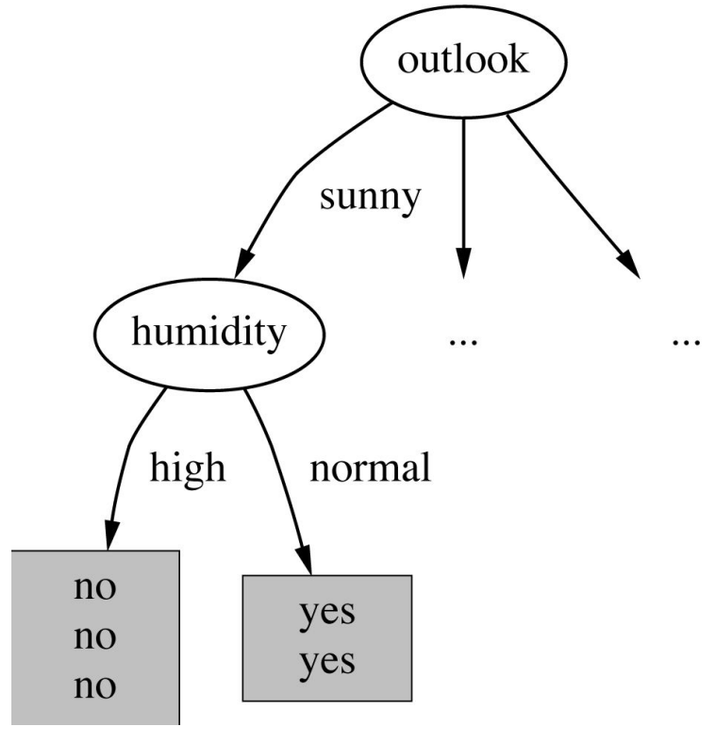
- Information gain para los atributos de los datos de weather:

gain(*Outlook*) = 0.247 bits
gain(*Temperature*) = 0.029 bits
gain(*Humidity*) = 0.152 bits
gain(*Windy*) = 0.048 bits

Seguimos particionando





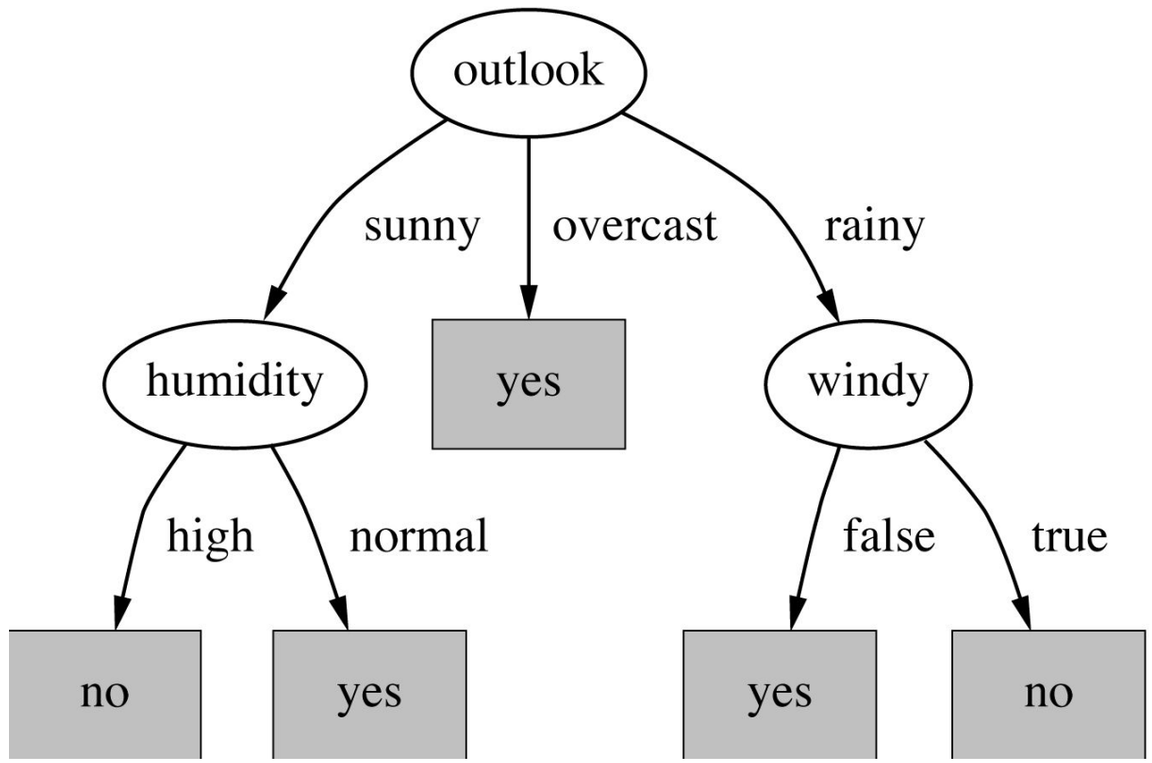


gain(*Temperature*) = 0.571 bits

gain(*Humidity*) = 0.971 bits

gain(*Windy*) = 0.020 bits

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.

Otras cosas sobre árboles

- 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).

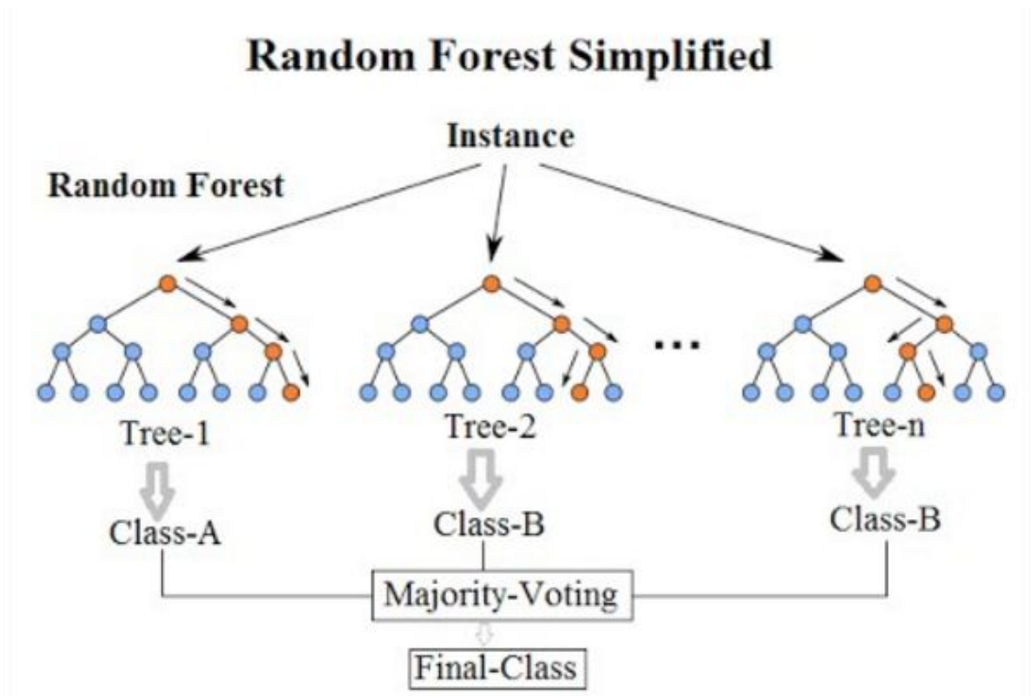
- Existen otras métricas para medir pureza distintas a entropía como el índice de **Gini** =1-Pr(Sacar dos ejemplos de la misma clase).

- 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**.

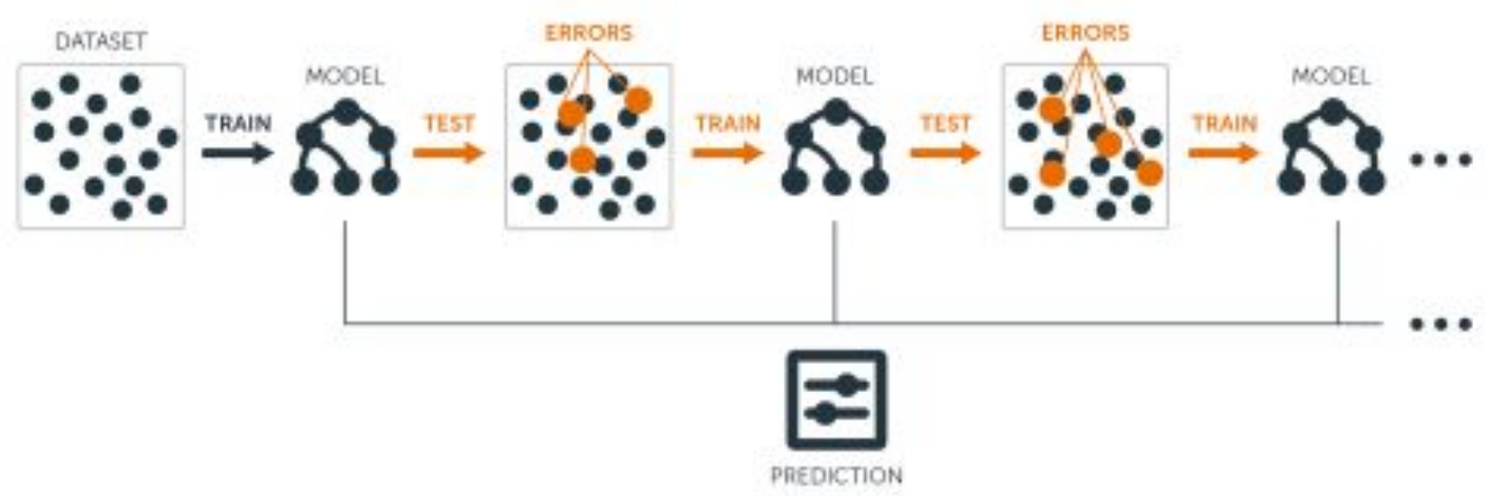
Random Forest

- Es posible combinar varios árboles de decisión para tener modelos con mayor capacidad predictiva.
- Random Forest:
  - un **ensamble** de árboles de decisión (a veces de un sólo nivel).
  - Cada árbol se entrena con distintas muestras aleatorias (con reemplazo) del dataset de entrenamiento.
  - Luego los distintos árboles votan para hacer predicciones finales



Gradient Boosting

- Boosting es otro enfoque para combinar varios clasificadores
- Idea: se entrenan modelos de manera secuencial donde cada modelo siguiente trata de corregir los errores del modelo anterior
- En Gradient Boosting los modelos son **árboles de decisión**.
- Cada árbol se entrena para predecir los errores del anterior.
- La predicción final es una suma ponderada de todos los árboles.
- Existen implementaciones eficientes de la idea como XGBoost y CatBoost para atributos categóricos.

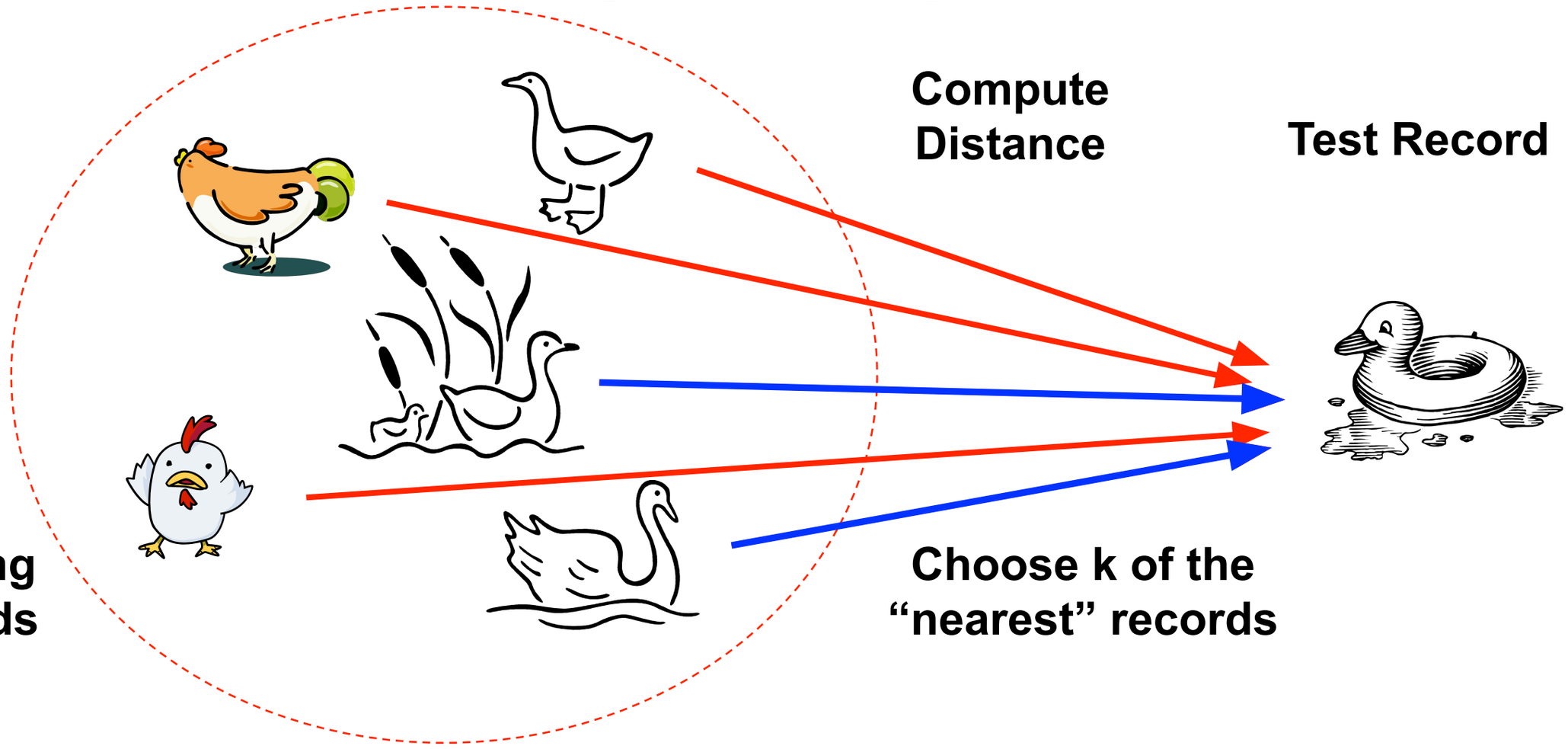


Clasificador KNN

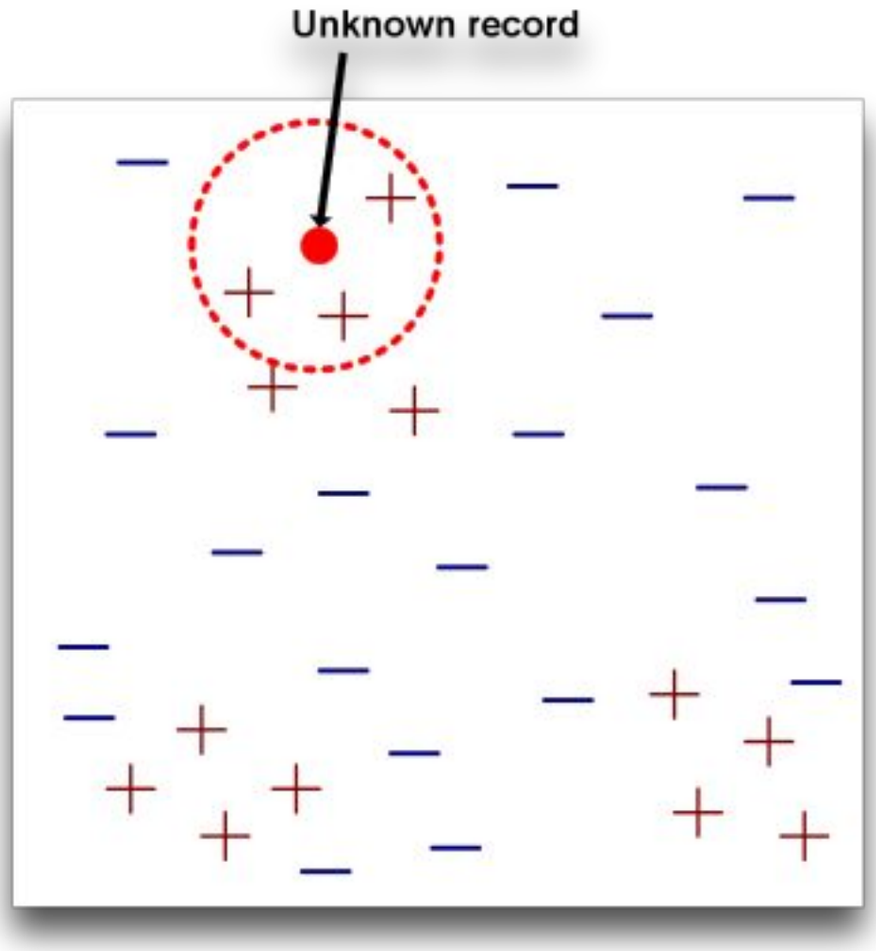
- Nearest Neighbor Classifier (o k-nn)
- Es un clasificador basado en **instancias**
- Conocido como **lazy**
  - Usa los **k** puntos más cercanos (nearest neighbors) para realizar la clasificación

Clasificadores KNN

- Idea:
  - **If it walks like a duck, quacks like a duck, then it's probably a duck**



Clasificadores KNN



- Necesita 3 cosas
  - –Set de records almacenados.
  - –Métrica de distancia para calcular la distancia entre records.
  - –Valor de *k*, el número de vecinos cercanos a obtener.
- Para clasificar un récord nuevo
  - –Calcular la distancia de los récords almacenados.
  - –Identificar *k* nearest neighbors .
  - – Utilizar la clase de los knn para asignar la clase al record nuevo (e.j. voto de la mayoría).

Métricas de distancia

- Para atributos numéricos usamos la distancia euclidiana:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2},$$

- Una versión más general es la distancia de **Minkowsky** (r=1 => distancia Manhattan, r=2 => distancia euclidiana)

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

- Es muy importante que los atributos estén normalizados.

Escalando atributos

- Problemas de escalas

  - Atributos deben ser escalados para prevenir que algún atributo domine la métrica de distancia

  - Ejemplos:

    - La altura de una persona puede variar entre 1.5m a 1.8m

    - El peso puede variar entre 40 kg a 150 kg

    - El ingreso de una persona puede variar entre $150K a $10M

Técnicas para escalar atributos

$$\frac{x-\mu_x}{\sigma_x}$$

- Normalización a media cero y varianza unitaria:

$$\frac{x-min_x}{max_x-min_x}$$

Distancia y similitud

- Es importante distinguir entre métricas de similitud y métricas de distancia.
- Similitud: entre más cerca dos objetos mayor el valor de la métrica.
- Distancia: entre más lejos dos objetos mayor el valor de la métrica.

Attribute Type	**Dissimilarity**	**Similarity**
Nominal	$d = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$	$s = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$
Ordinal	$d = \lvert x - y \rvert / (n-1)$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - d$
Interval or Ratio	$d = \lvert x - y \rvert$	$s = -d$, $s = \frac{1}{1+d}$, $s = e^{-d}$, $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Similitud Coseno

- Cuando nuestros objetos son vectores sparse (muchas columnas con cero) es conveniente usar la similitud coseno.
- Corresponde al coseno del ángulo entre los dos vectores.

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \, \|\mathbf{y}\|} \qquad \|\mathbf{x}\| = \sqrt{\sum_{k=1}^{n} x_k^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}.$$

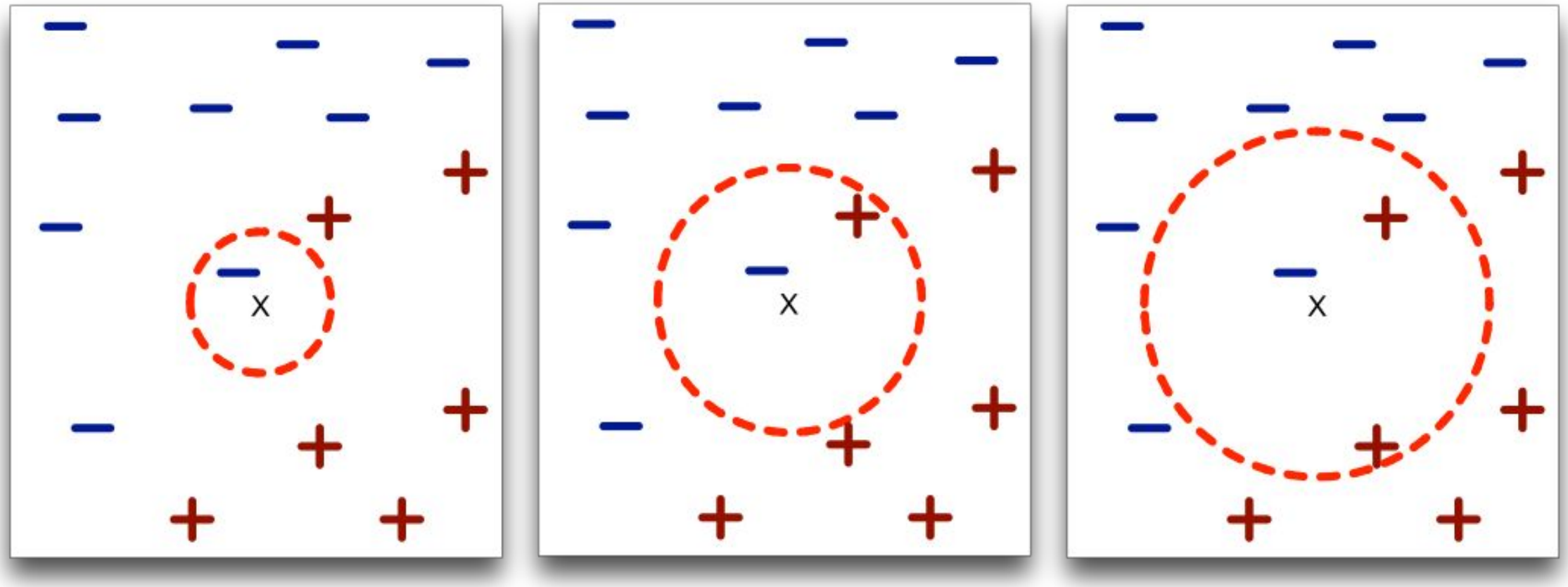
- Un ejemplo común es cuando tratamos documentos como bolsas de palabras (cada columna es una palabra del vocabulario).

Ejemplo:

$$\mathbf{x} = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$
$$\mathbf{y} = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

$$\mathbf{x} \cdot \mathbf{y} = 3 * 1 + 2 * 0 + 0 * 0 + 5 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 1 + 0 * 0 + 0 * 2 = 5$$
$$\|\mathbf{x}\| = \sqrt{3 * 3 + 2 * 2 + 0 * 0 + 5 * 5 + 0 * 0 + 0 * 0 + 0 * 0 + 2 * 2 + 0 * 0 + 0 * 0} = 6.48$$
$$\|\mathbf{y}\| = \sqrt{1 * 1 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 2 * 2} = 2.24$$
$$\mathbf{cos}(\mathbf{x}, \mathbf{y}) = \mathbf{0.31}$$

Definición de NN



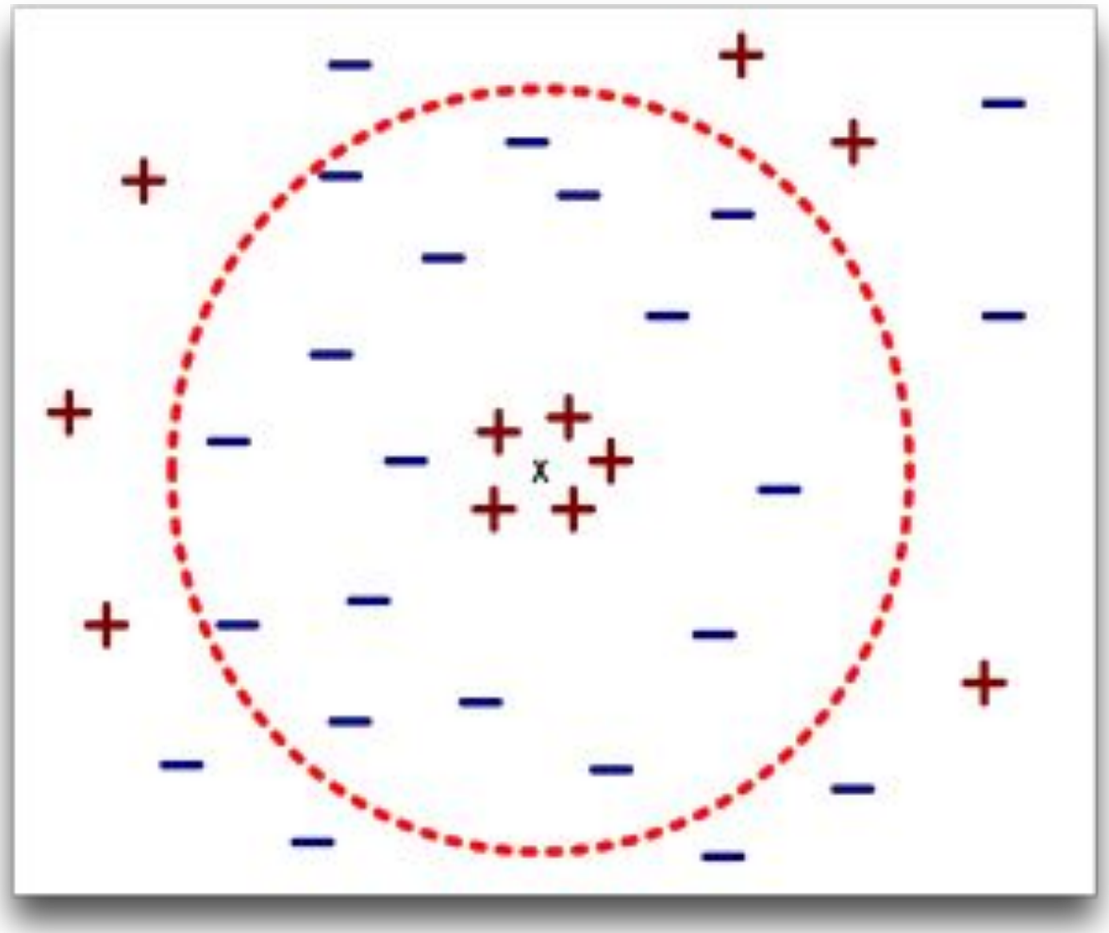
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-NN de un récord x son los puntos que tienen las k menores distancias a x

Eligiendo el valor de K

- k muy pequeño es susceptible a ruido

- k muy grande puede incluir puntos de otra clase



Clasificación kNN

- Los clasificadores k-NN son **lazy learners.**
  - No construyen modelos explícitos, es más flexible ya que no necesita comprometerse con un modelo global a priori.
  - Al contrario de otros **eager learners** como los árboles de decisión o clasificadores basados en reglas.
  - Es independiente del nro. de clases.
  - La clasificación es más costosa (memoria y tiempo).

La Maldición de la Dimensionalidad

- Cuando los datos tienen una alta dimensionalidad KNN está sujeta a la **Maldición de la Dimensionalidad**.

- Fenómeno en que muchos tipos de análisis de datos se vuelven significativamente más difíciles a medida que aumenta la dimensionalidad de los datos.

- Para la clasificación, esto puede significar que no haya suficientes ejemplos para crear un modelo que asigne de forma confiable una clase a todos los ejemplos posibles.

- Para técnicas basadas en distancias (KNN, K-means) las distancias entre objetos se vuelven menos claras cuando hay muchas dimensiones.

dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl

/ DCCUCHILE