


IMT2200 Introducción a la Ciencia de Datos



Rodrigo A. Carrasco
Instituto de Ingeniería Matemática y Computacional
Escuela de Ingeniería

 R⁶ Rodrigo_carrasco2

 @_rax

 @rodrigo_a_Carrasco

 www.raxlab.science




Avisos Importantes

- **Tarea 4**

- Quedó disponible la Tarea 4 para su desarrollo, no la dejen para último momento.
- Ojo que les dejé un par de días más para su entrega.

- **Interrogación 2**

- El próximo viernes 14 es la I2 a las 16:30 en la sala C201.
 - Entra todo el material hasta la clase de este jueves 6 (clase 23 en Canvas). El foco estará en las clases 12 a 23.
 - Al igual que para la I1, habrá un set de preguntas de Armas de Destrucción Matemática (capítulos 4 a 6).
 - La parte de conceptos y el libro será sin apuntes; para la de desarrollo podrán usar todo el material visto en clases.
- 

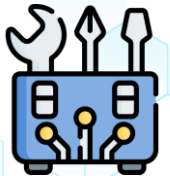


01

Repaso

Temas vistos la clase pasada

- Vimos cómo elegir entre modelos mediante “validación cruzada”.
- Además, revisamos otra herramienta de regresión: la regresión logística.
- A diferencia de las herramientas anteriores, esta nos permite clasificar y conectar una variable categórica (dependiente) con variables independientes.
- Ahora veremos otras herramientas para clasificar, que son distintos tipos de algoritmos de aprendizaje supervisado.



A decorative graphic on the left side of the slide. It consists of a grid of hexagons, some of which are outlined in white and others in a light blue. Some hexagons are filled with a light blue color, while others are empty. The pattern is layered, with some hexagons appearing to be in front of others, creating a 3D effect. The overall color scheme is teal and blue.

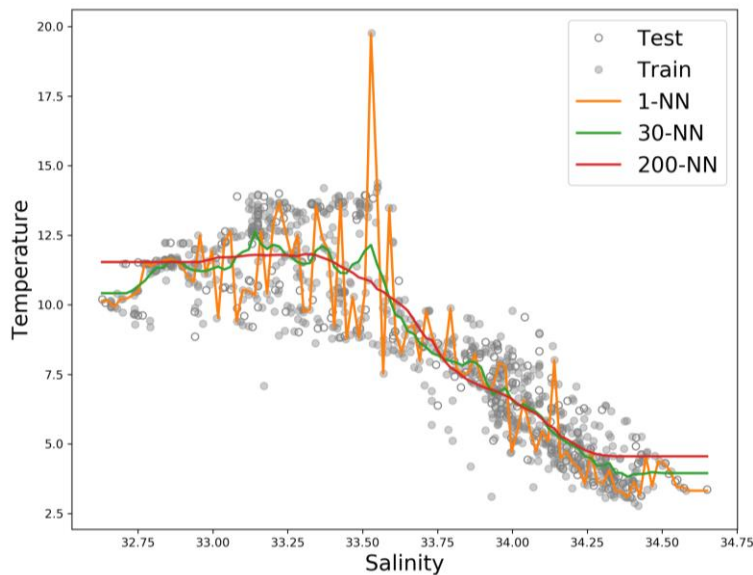
02

Machine Learning

Clasificación - kNN

Clasificación kNN

kNN para regresión: usamos como predictores, las observaciones disponibles (x,y) más similares a la observación (x) que queremos predecir.

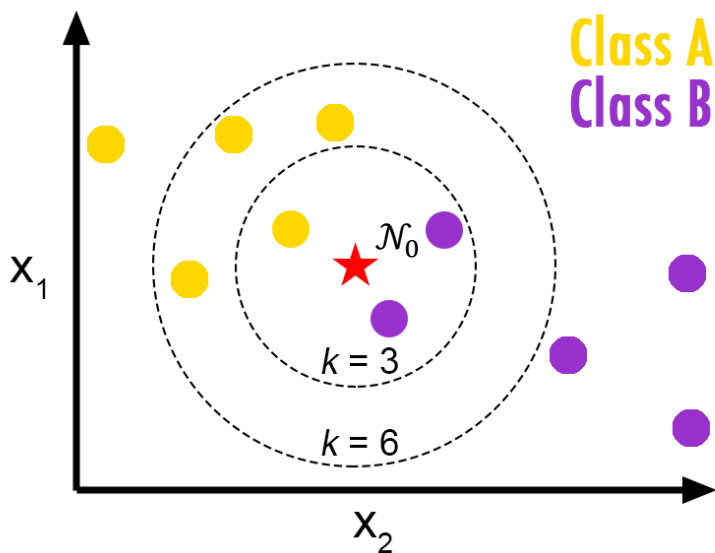


$$\hat{y}_i = \frac{1}{k} \sum_{j=1}^k y_{i_j}$$

y_{i_j} son los k vecinos más cercanos a (x_i, y_i)

Clasificación kNN

kNN para clasificación: clasificamos una observación específica, con base en las categorías de sus vecinos más cercanos.



Para un dato x_0 :

1. Se calcula la distancia a todos los demás puntos x_i :

$$D^2(x_i, x_0) = \sum_{j=1}^P (x_{i,j} - x_{0,j})^2$$

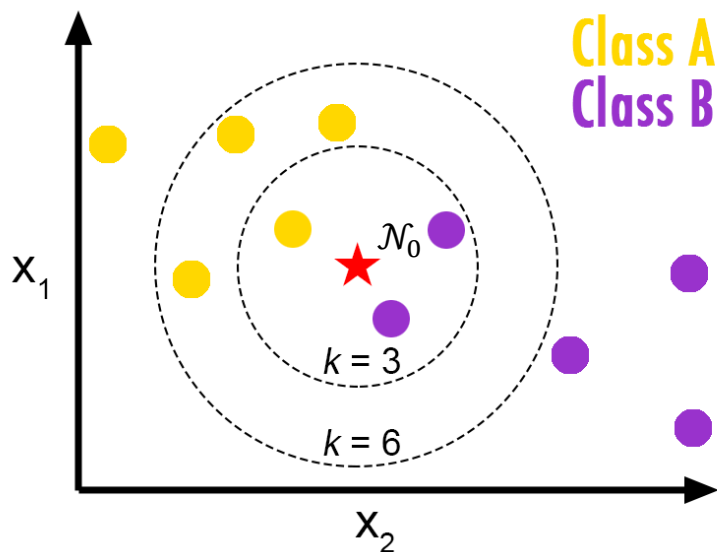
2. Se identifican los k puntos del dataset de entrenamiento más cercanos a $x_0 \rightarrow \mathcal{N}_0$

Clasificación kNN

$k = 3$: para ★

$$P(Y = A|X_1, X_2) = \frac{1}{3}, P(Y = B|X_1, X_2) = \frac{2}{3} \rightarrow Y = B$$

kNN para clasificación: clasificamos una observación específica, con base en las categorías de sus vecinos más cercanos.



3. Se estima la probabilidad condicional de la clase j , como la fracción de puntos en \mathcal{N}_0 cuyas respuestas son j

$$P(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

4. Se aplica la regla de Bayes y se clasifica la observación de prueba x_0 a la clase con la mayor probabilidad estimada.

Normalización

Si hay múltiples predictores: se define una medida de distancia multidimensional para identificar las observaciones más similares o “vecinos”.

- Distancia Euclideana: $D(x_i, x_0) = \sqrt{\sum_{j=1}^P (x_{i,j} - x_{0,j})^2}$
- Si los predictores tienen diferentes escalas y variabilidad → se introducen efectos de escala en la medición de distancia.
- Por lo tanto, para $p > 1$, es necesario estandarizar los predictores.
- **Normalización z:** se resta la media, y se divide por la desviación estándar.

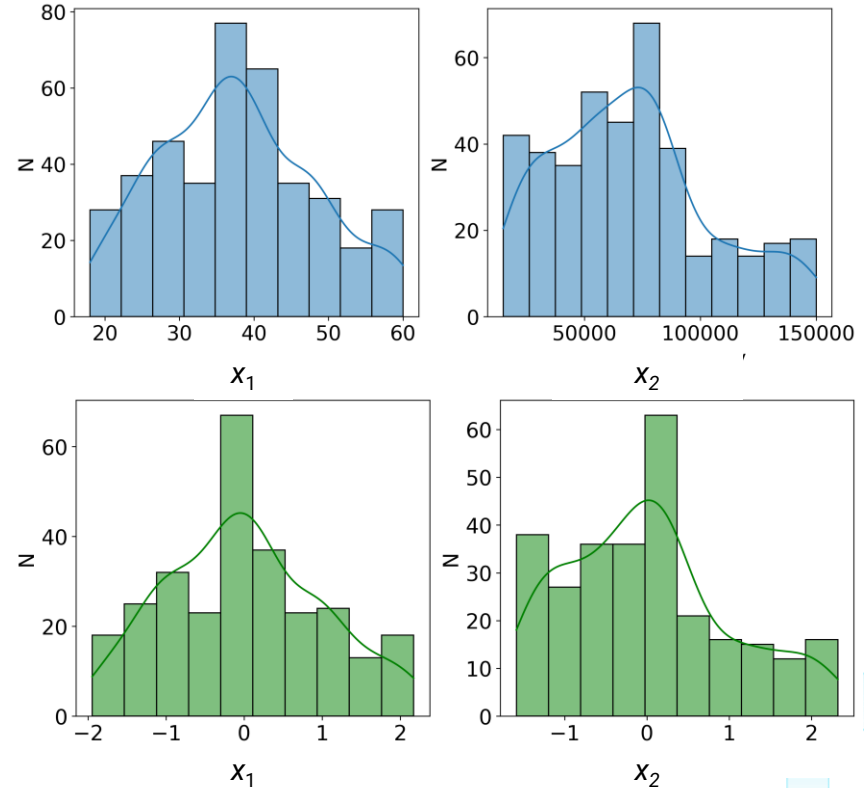
$$\Rightarrow x_{scaled} = \frac{x - \mu}{\sigma}$$

Ejemplo de normalización

Ejemplo: Predicción de comportamiento de compra de clientes de una RS en base a su edad e ingresos.

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...

**Datos
normalizados**

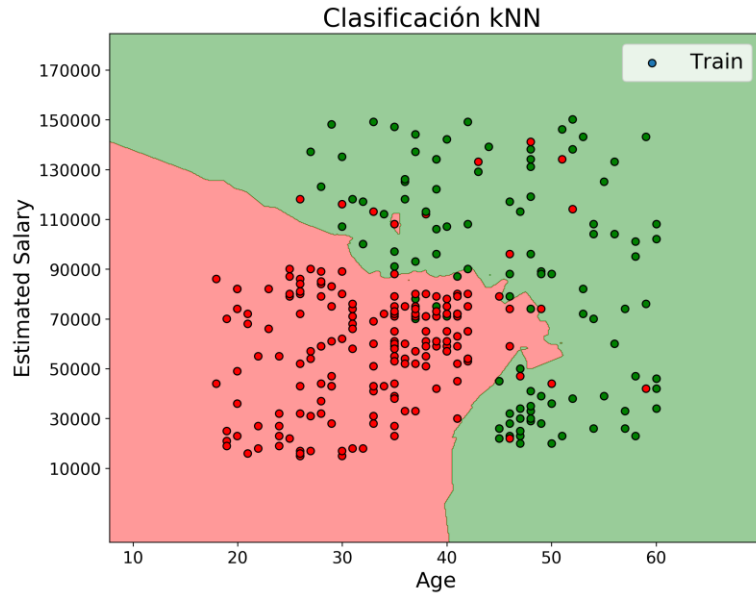




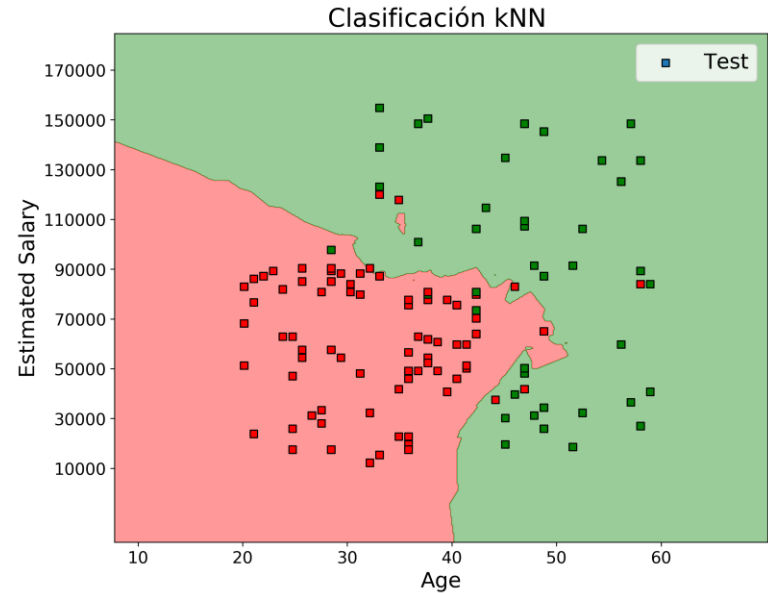
¿Cómo evaluamos un
modelo de clasificación?

Clasificación kNN: Ejemplo

¿Cómo evaluamos qué tan buena es la clasificación?



Purchased = 0 (Negative)
Purchased = 1 (Positive)



Evaluación de un modelo

Matriz de confusión: es usada para evaluar los resultados de la clasificación

- C_{ij} : número de observaciones que se sabe están en el grupo i , y son clasificadas en el grupo j

Real	0	$C_{0,0}$ Verdaderos negativos (tn)	$C_{0,1}$ Falso positivo (fp)
	1	$C_{1,0}$ Falso negativo (fn)	$C_{1,1}$ Verdaderos positivos (tp)
		0	1
		Predicción	

Accuracy/ Exactitud: fracción de aciertos (en el dataset de prueba)

$$\text{accuracy} = \frac{tp + tn}{tp + fp + tn + fn}$$

Precision/Precisión: capacidad de no clasificar como “positivo” un negativo

$$\text{precision} = \frac{tp}{tp + fp}$$

Evaluación del modelo

Matriz de confusión: es usada para evaluar los resultados de la clasificación

- $C_{i,j}$: número de observaciones que se sabe están en el grupo i , y son clasificadas en el grupo j

Real	0	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$
	1	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$
	2	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$
		0	1	2
		Predicción		

Evaluación del modelo

Matriz de confusión: es usada para evaluar los resultados de la clasificación

- $C_{i,j}$: número de observaciones que se sabe están en el grupo i , y son clasificadas en el grupo j

$C_{0,0}$ Verdaderos negativos (tn)	$C_{0,1}$ Falso positivo (fp)
$C_{1,0}$ Falso negativo (fn)	$C_{1,1}$ Verdaderos positivos (tp)

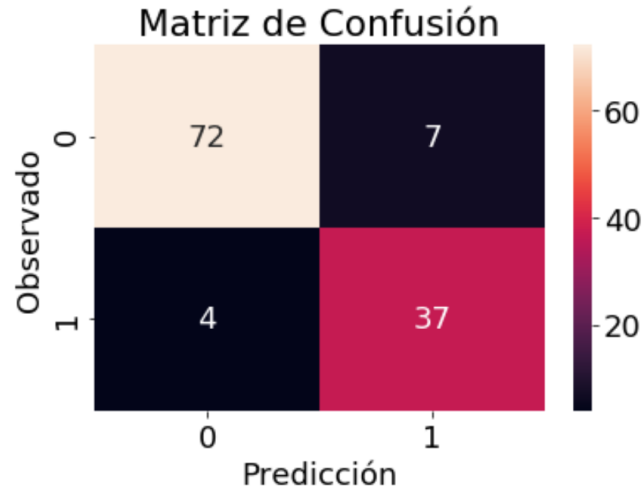
Recall / Sensibilidad: capacidad del clasificador de identificar todos los “positivos”

$$\text{recall} = \frac{tp}{tp + fn}$$

F-score: promedio ponderado de precisión y recall

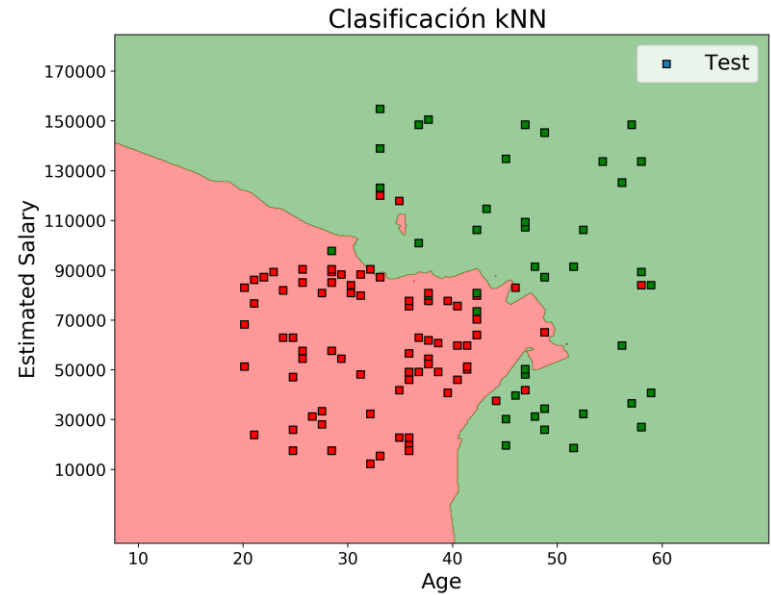
$$F = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Ejemplo




Purchased = 0 (Negative)

Purchased = 1 (Positive)





Clasificación kNN: consideraciones

- ✓ Método intuitivo y simple de entender e implementar.
 - ✓ Funciona bien para clasificaciones binarias, o multiclase.
 - ✓ Sólo tiene un hiperparámetro (k)
 - ✗ El algoritmo es lento para grandes datasets.
 - ✗ Funciona bien con pocas variables predictoras, pero falla para problemas de muchas dimensiones.
 - ✗ Requiere normalizar los features para evitar problemas de escala.
 - ✗ No funciona bien sobre datasets imbalanceados.
- 

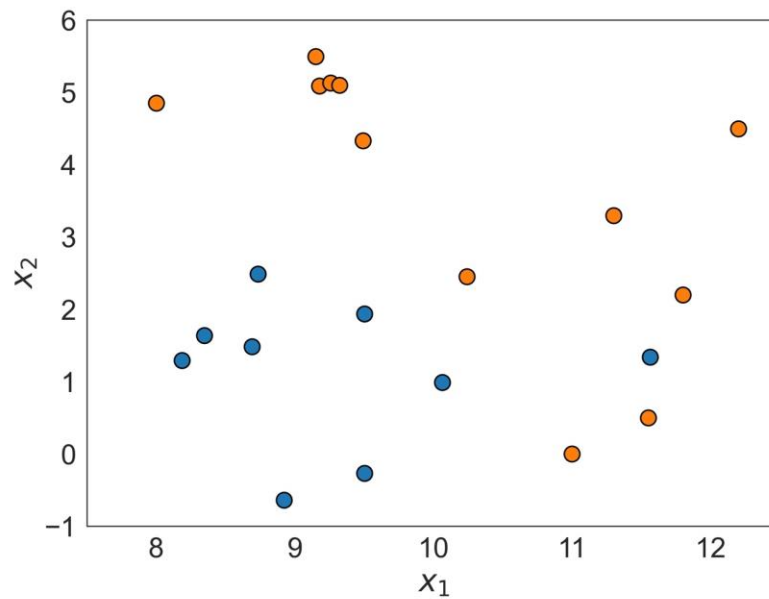
A decorative graphic on the left side of the slide. It consists of a grid of hexagons, some of which are outlined in white and others in a lighter teal. Some hexagons are filled with a gradient from teal to dark blue. Small teal dots are placed at the vertices of the hexagons, and thin teal lines connect some of them, creating a network-like structure.

03

Machine Learning

Clasificación - Árboles

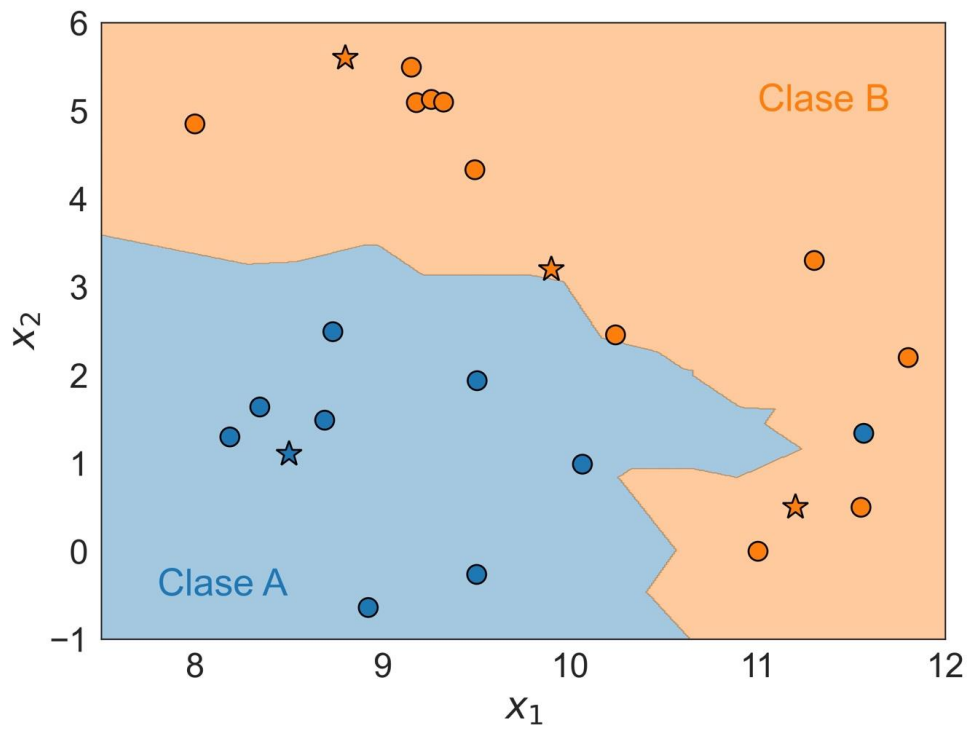
Ejemplo



● D_{train} (clase A)

● D_{train} (clase B)

Ejemplo



Clasificación kNN

($k=3$)

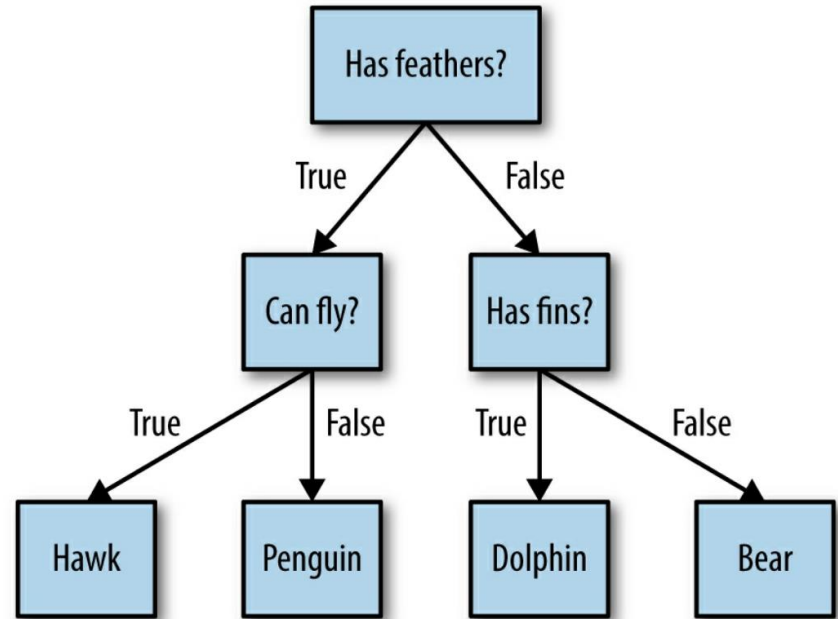
Árboles de decisión

- Los árboles de decisión son modelos de **regresión y clasificación** que se basan en aprender una **jerarquía de preguntas (tests) de tipo if/else** para llegar de la forma más rápida posible a una decisión acertada respecto al valor de la etiqueta de una nueva observación.



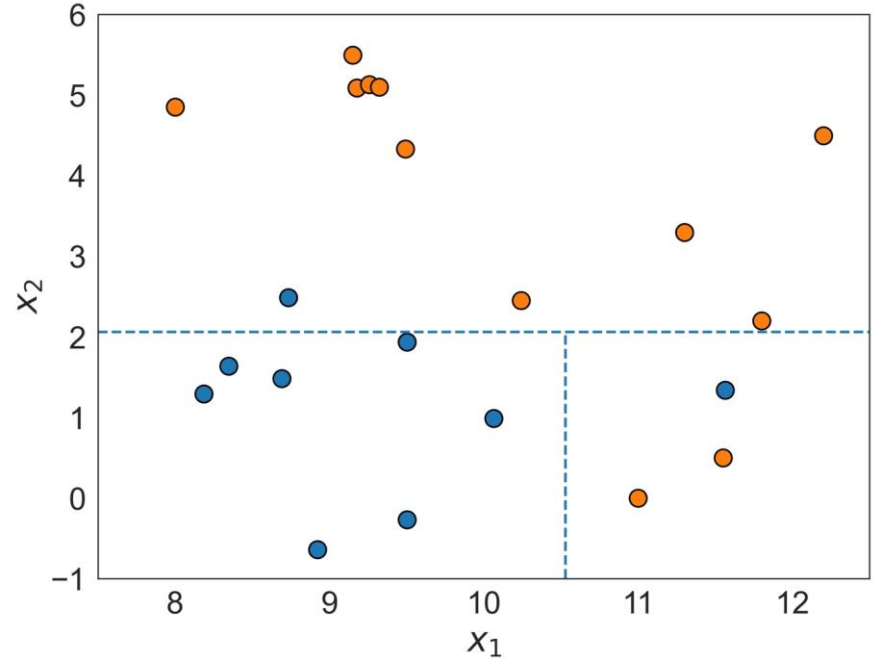
Árboles de decisión

- Las preguntas pueden ser relativas al valor de una variable numérica o categórica.
- El algoritmo busca sobre todas las posibles pruebas, y selecciona la que es **más informativa** respecto a la variable objetivo.



Ejemplo

- ¿es x_2 mayor a 2.06? y luego,
- ¿es x_1 menor a 10.5?

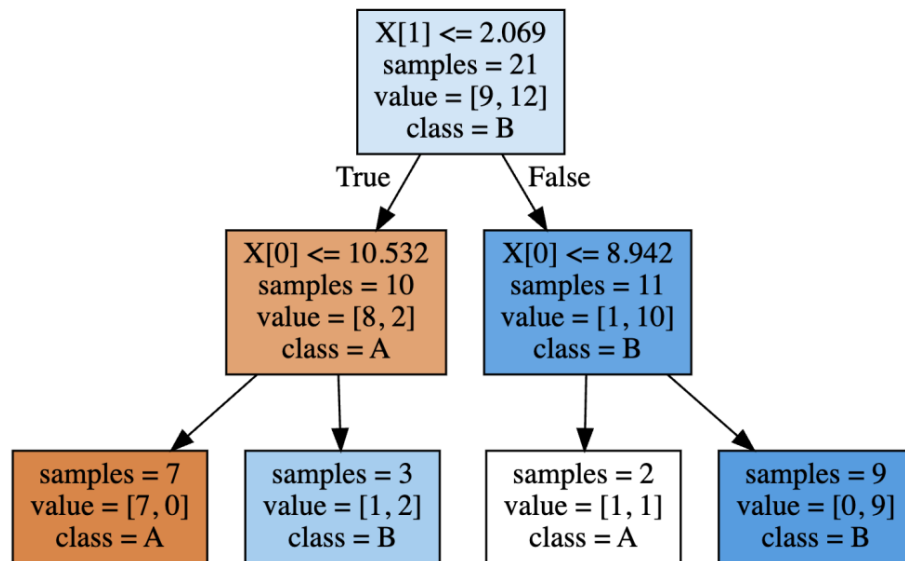


● \mathcal{D}_{train} (clase A)

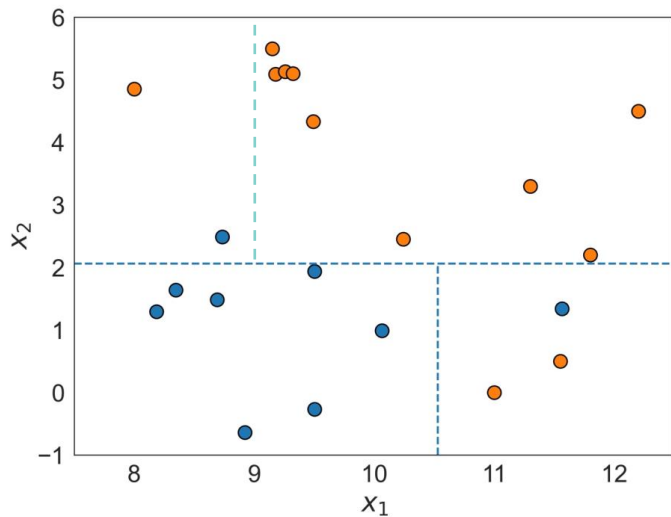
● \mathcal{D}_{train} (clase B)

Ejemplo

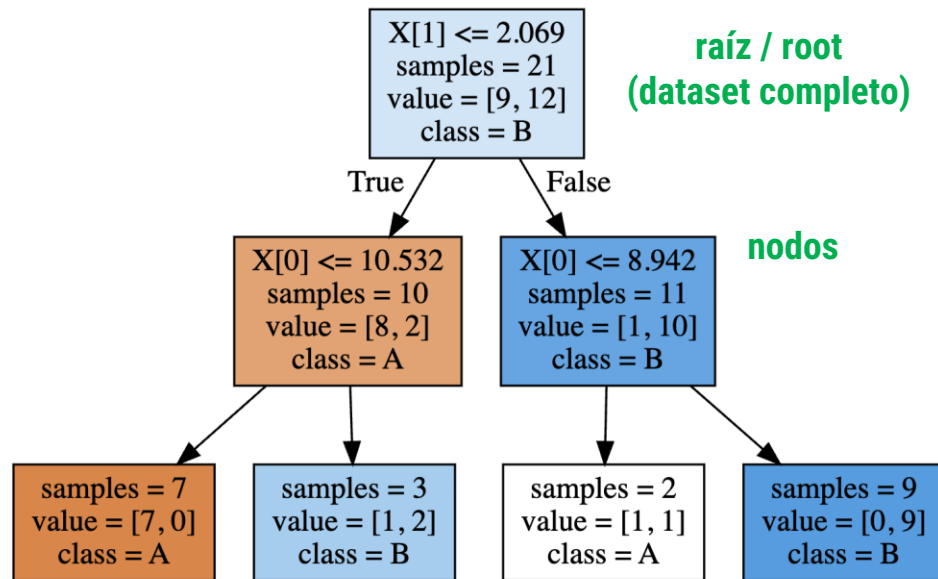
- Para los datos de ejemplo, un árbol de decisión para clasificar las observaciones en clases A o B podría tener la siguiente estructura de tres niveles.
- Cada pregunta concierne **sólo a una variable**, por lo tanto, cada pregunta particiona los datos a lo largo de un eje.



Ejemplo

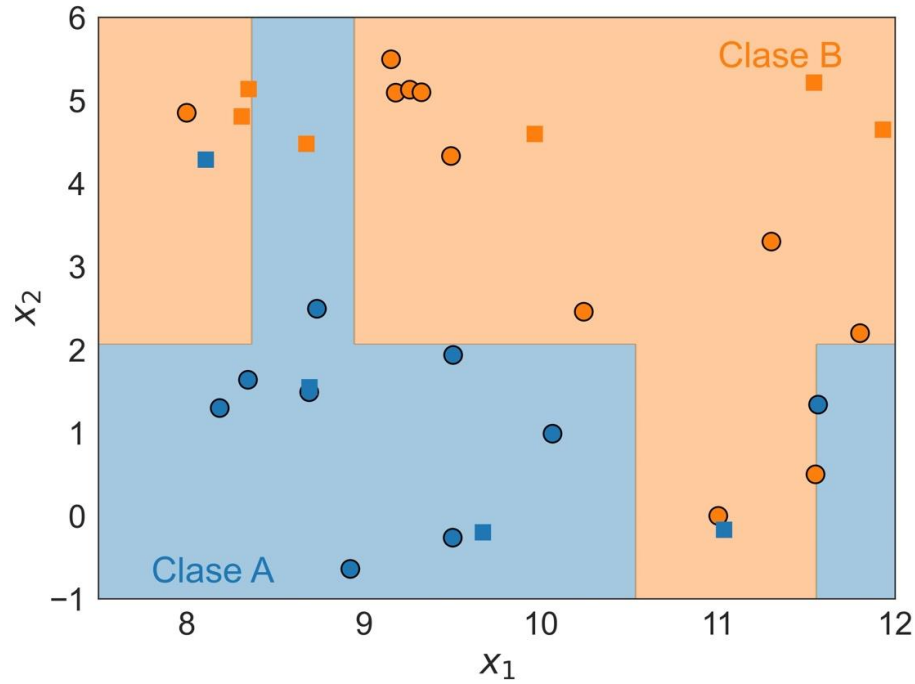


- D_{train} (clase A)
- D_{train} (clase B)



hoja/leaf
(región en la
partición)

Ejemplo



Frontera de decisión en el
plano x_1 - x_2

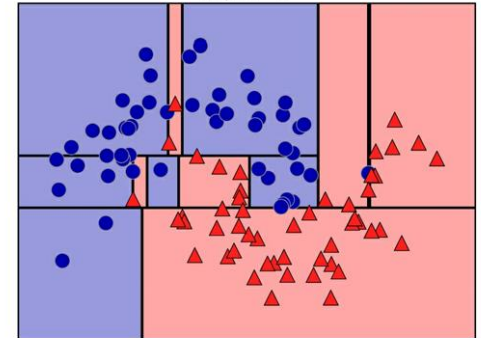
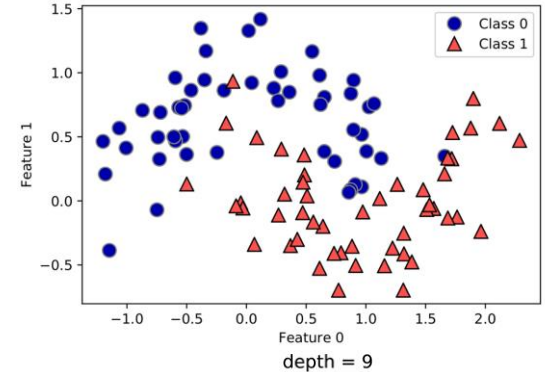
(distinta a la obtenida con el
clasificador kNN)

Complejidad del árbol de decisión

Típicamente, la construcción del árbol de decisión continua hasta que todas las hojas son **“puras”**: **contienen sólo una clase objetivo.**

Esto puede llevar a modelos muy complejos: **overfitting.**

- hojas puras → 100% precisión para datos de entrenamiento
- Estrategias para prevenir overfitting:
- **Pre-pruning:** cortar tempranamente la creación del árbol
 - Limitar profundidad
 - Limitar nº de hojas
 - Requerir mínimo de puntos en un nodo para dividir
- **Post-pruning:** crear el árbol completo, y colapsar nodos que aportan poca información.



A decorative graphic on the left side of the slide. It features a grid of hexagons in various shades of teal and blue. Some hexagons are solid, while others are outlined. Small teal dots are placed at the vertices of the hexagonal grid.

04

Notebook de Ejemplos

Está disponible en el GitHub del curso