



SEMANA 3

## Regresión y clasificación con Árboles de Decisión y Random Forest

Prof. Luis Torrejón



---

Al finalizar, el alumno conocerá sobre la regresión y clasificación con regresión logística, árboles de decisión y random forest.

---





## TEMARIO

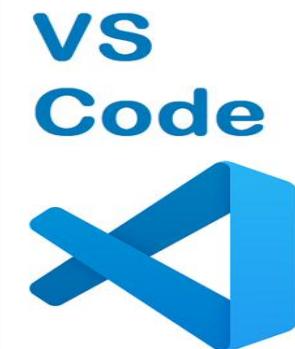
- 1 — Regresión logística
- 2 — Regresión y clasificación con Árboles de Decisión
- 3 — Regresión y clasificación con Árboles de Decisión





# Revisión de actividad individual

Revisión de Laboratorio Regresión Lineal





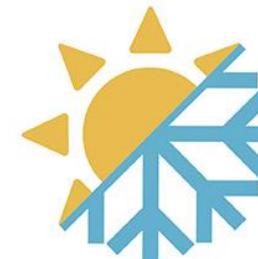
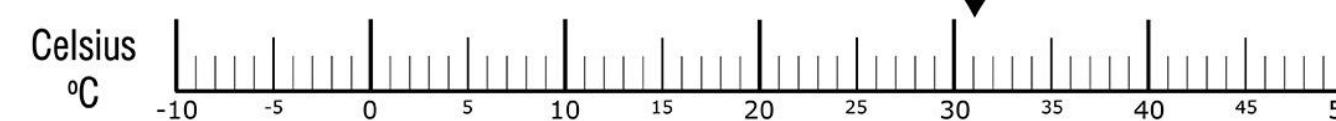
# Recordemos - Algoritmos de clasificación vs regresión



**Regresión**

¿Qué temperatura habrá mañana?

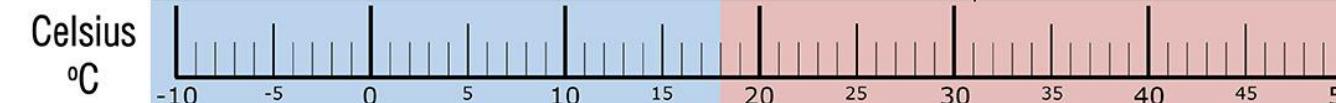
PREDICCIÓN  
31°



**Clasificación**

¿Mañana será un día frío o caluroso?

PREDICCIÓN  
Caluroso





# Recordemos - Algoritmos de clasificación vs regresión

| Característica         | Clasificación   | Regresión  |
|------------------------|---|--|
| Variable Objetivo      | Categórica (clases o categorías)                        | Continua (valores numéricos)                             |
| Ejemplos de Salidas    | "Aprobado" vs. "Reprobado", "Spam" vs. "No Spam"        | Precio de una vivienda, temperatura en °C                |
| Tipo de Problema       | Determinar una categoría o grupo                        | Predecir un valor cuantitativo                           |
| Función de Error       | Funciones como entropía o índice de Gini                | Error cuadrado medio (MSE), error absoluto medio         |
| Ejemplos de Algoritmos | Árboles de decisión, Naive Bayes, SVM, redes neuronales | Regresión lineal, árboles de regresión, redes neuronales |
| Aplicaciones Comunes   | Detección de fraude, clasificación de imágenes          | Estimación de precios, predicción de series temporales   |

# Regresión logística

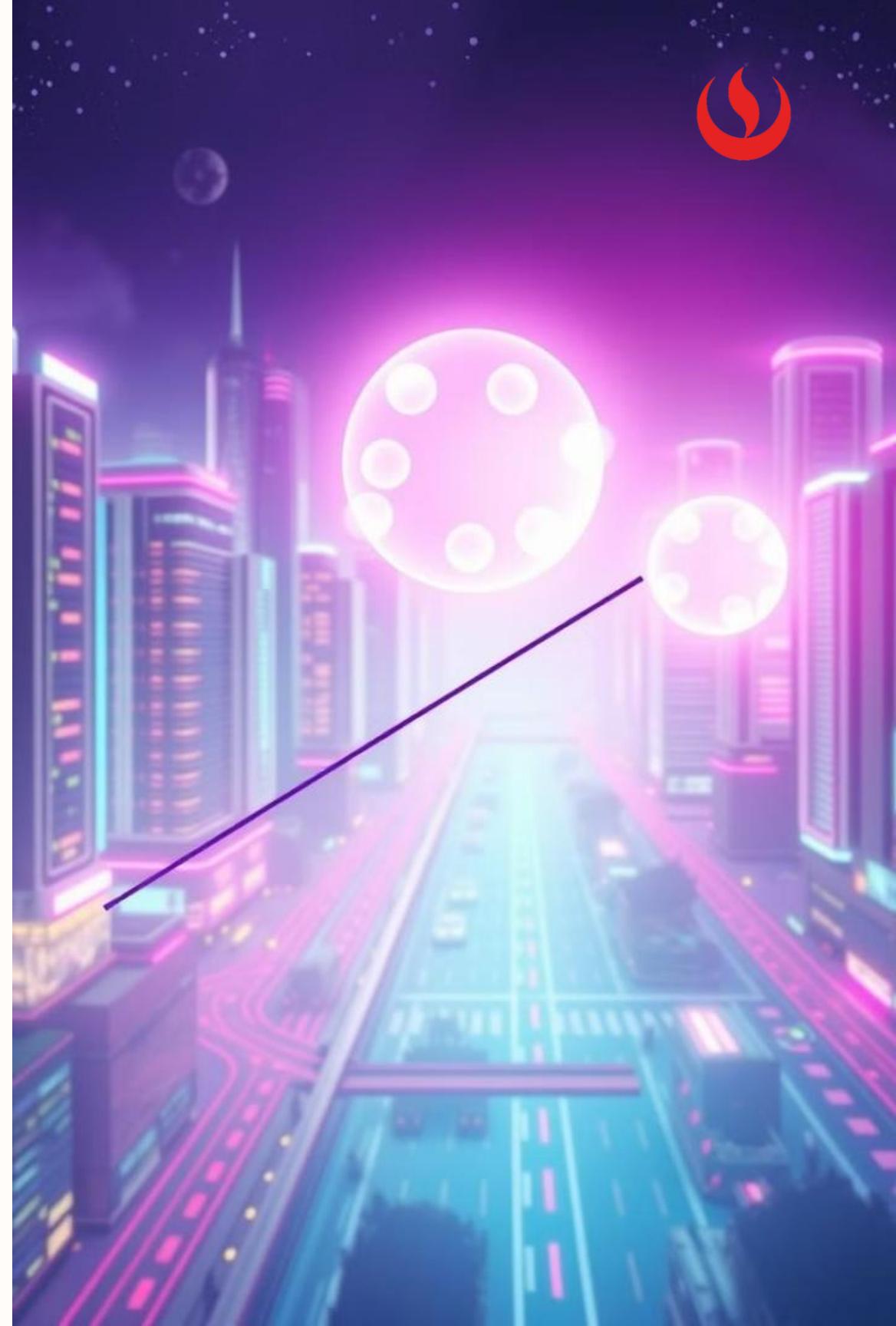




# Regresión logística

La regresión logística es una técnica estadística que se utiliza para predecir la probabilidad de un evento binario (sí/no, verdadero/falso).

Es una herramienta esencial en el aprendizaje supervisado, donde buscamos construir un modelo que pueda aprender de datos históricos para hacer predicciones sobre nuevos datos.





# ¿Qué es la Regresión Logística?

La regresión logística es un **algoritmo de clasificación supervisado**, no un método de regresión a pesar de su nombre engañoso. Esta técnica estadística se utiliza para modelar la probabilidad de que un evento discreto ocurra, produciendo una salida que siempre está acotada entre 0 y 1.

A diferencia de los métodos de regresión tradicionales que predicen valores continuos, la regresión logística está diseñada específicamente para **problemas de clasificación**, donde el objetivo es asignar observaciones a categorías discretas predefinidas.

## Entrada

Variables predictoras continuas o categóricas

## Proceso

Transformación logística

## Salida

Probabilidad entre 0 y 1

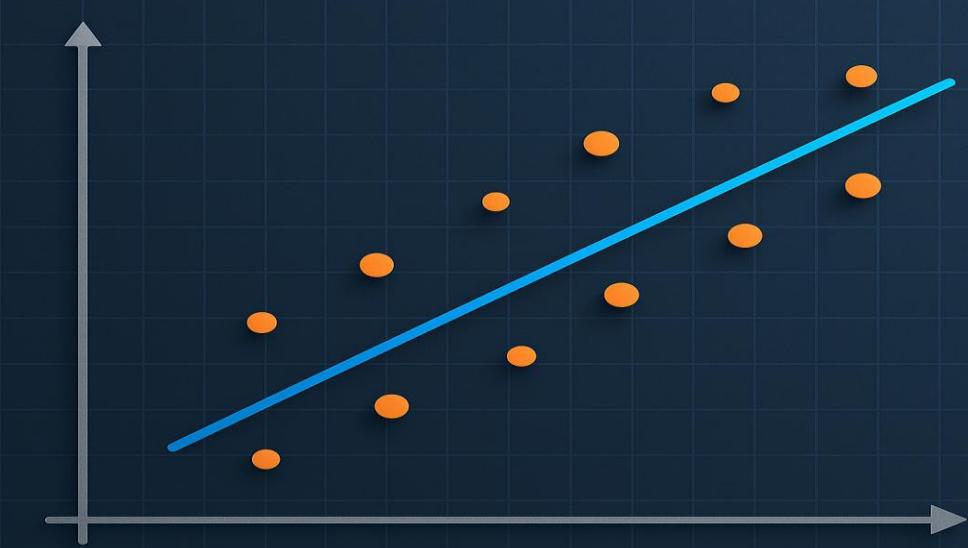


# Diferencia Clave con la Regresión Lineal

## Regresión Lineal

Predice valores **continuos** sin límites definidos. Por ejemplo, puede estimar el precio de una vivienda que puede tomar cualquier valor positivo, desde miles hasta millones de dólares. La salida es un número en un rango potencialmente infinito.

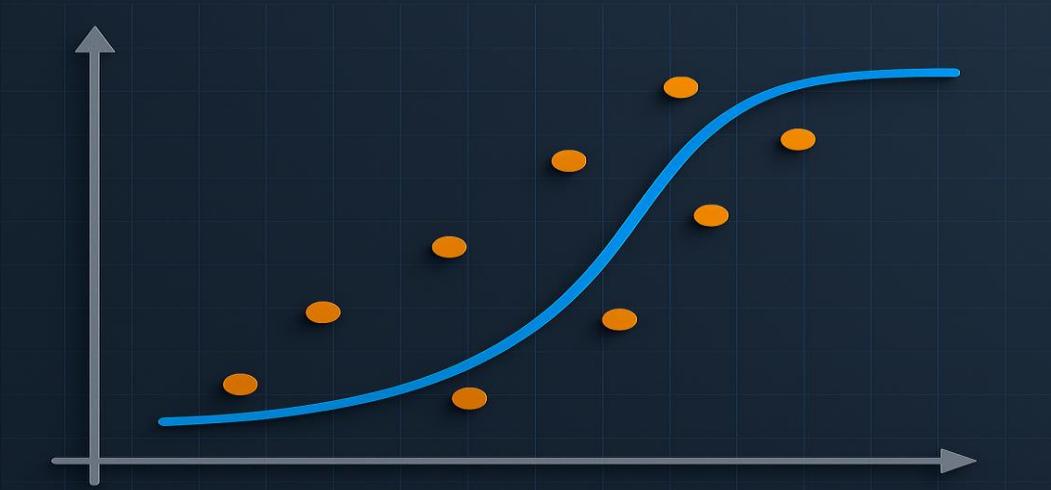
- Predice cantidades numéricas
- Rango:  $(-\infty, +\infty)$
- Ejemplo: precio, temperatura, ingresos



## Regresión Logística

Predice la **probabilidad** de pertenencia a una categoría discreta. La salida siempre está acotada entre 0 y 1, representando la certeza de que una observación pertenezca a una clase específica (como "Sí" o "No", "Fraude" o "Legítimo").

- Predice probabilidades de clases
- Rango:  $[0, 1]$
- Ejemplo: Sí/No, Éxito/Fracaso, 0/1



La función de enlace logística es la clave que permite transformar una combinación lineal ilimitada en una probabilidad válida.



# La Función Logit (Log-Odds)

En el corazón de la regresión logística se encuentra la **función logit**, también conocida como logaritmo de las probabilidades o log-odds. Esta función establece la relación lineal fundamental del modelo.

La regresión logística no modela directamente la probabilidad P, sino que trabaja con la transformación logit de esa probabilidad. Esta transformación convierte el rango restringido [0,1] de las probabilidades en el rango ilimitado  $(-\infty, +\infty)$ , permitiendo el uso de una ecuación lineal estándar:

$$\ln \left( \frac{P}{1 - P} \right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

**P / (1-P)**

**Odds Ratio** - La razón de momios representa cuántas veces es más probable que ocurra el evento versus que no ocurra

**ln(Odds)**

**Log-Odds** - El logaritmo natural de los odds, que puede tomar cualquier valor real y permite la modelización lineal

**$\beta_0, \beta_1 \dots \beta_n$**

**Coeficientes** - Parámetros que el modelo aprende y que cuantifican el efecto de cada variable predictora



# La Función Sísmoide

Una vez que hemos modelado los log-odds con una ecuación lineal, necesitamos **convertir esa salida ilimitada en una probabilidad válida** entre 0 y 1. Aquí es donde entra la función sísmoide, también llamada función logística.

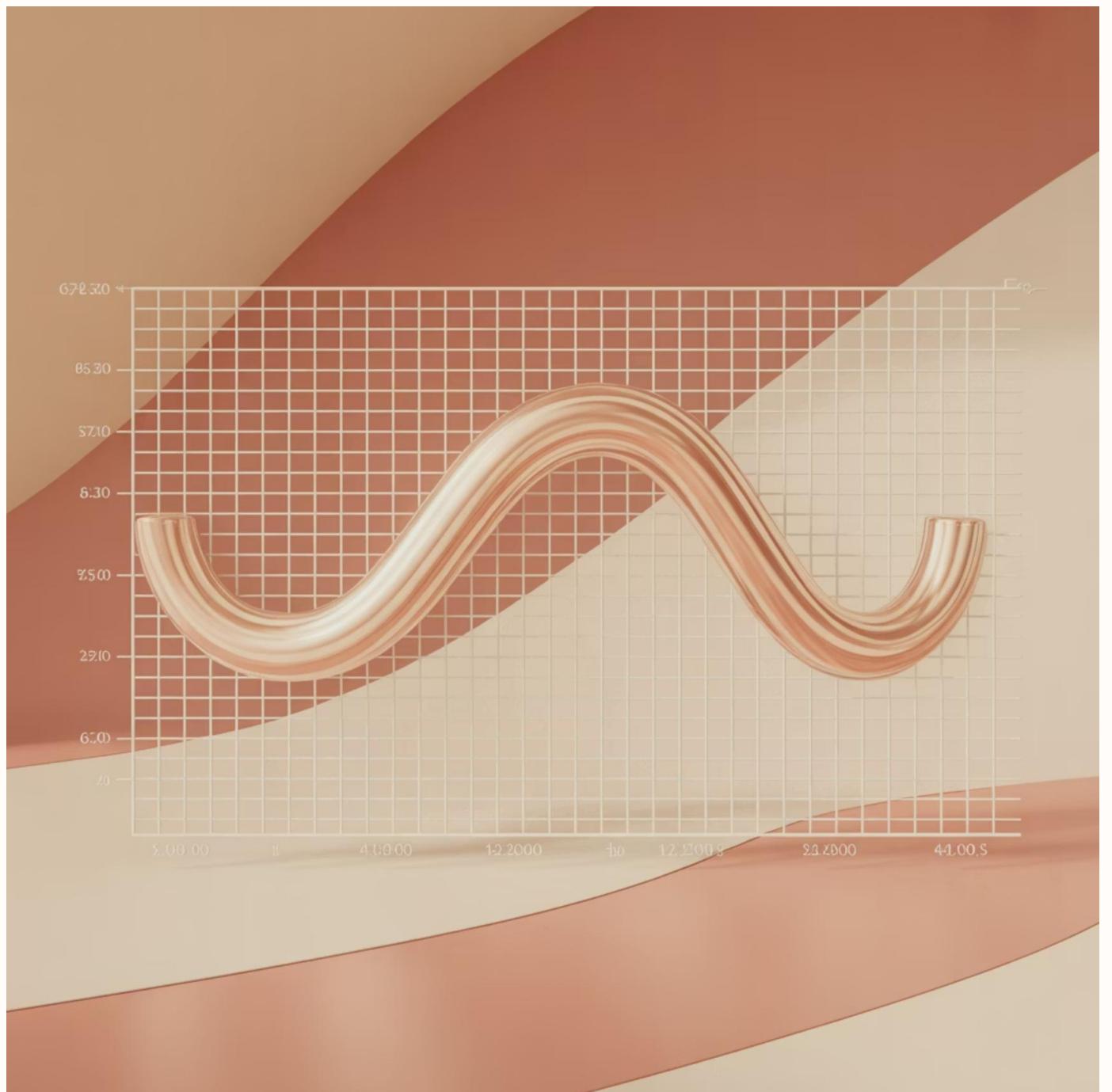
La función sísmoide ( $\sigma$ ) tiene una forma característica de "S" que suavemente transforma cualquier número real en un valor entre 0 y 1. Esta transformación es invertible respecto a la función logit, creando una correspondencia perfecta entre probabilidades y log-odds.

$$P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \dots + \beta_n X_n)}}$$

Donde  $e$  es la constante de Euler (aproximadamente 2.718) y el exponente negativo garantiza la forma correcta de la curva.

$$\sigma(0) = \frac{1}{1 + e^0} = \frac{1}{2} = 0.5$$

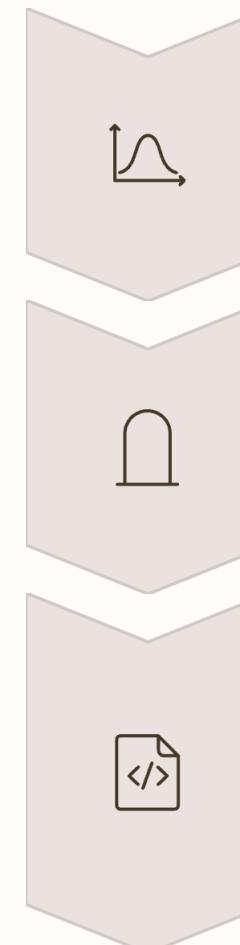
Esto indica que hay un 50% de probabilidad de que el evento ocurra.





# El Umbral de Decisión

La salida del modelo es una probabilidad ( $P$ ), pero necesitamos convertirla en una clasificación concreta. Para ello, aplicamos un **umbral de decisión**.



## Probabilidad Calculada

El modelo genera  $P$  entre 0 y 1

## Aplicar Umbral

Típicamente 0.5 (ajustable según el caso)

## Clasificación Final

Si  $P \geq 0.5 \rightarrow$  Clase 1 Si  $P < 0.5 \rightarrow$  Clase 0



# Supuesto 1: Variable Dependiente Binaria

El resultado ( $Y$ ) debe ser **categórico y dicotómico**, es decir, tener exactamente dos categorías posibles. Este es el supuesto fundamental de la regresión logística binaria.

## Ejemplos Comunes:

- **Fraude:** Sí/No
- **Enfermedad:** Presente/Ausente
- **Conversión:** Compra/No compra
- **Aprobación:** Aprobado/Rechazado

**Nota:** Este modelo se extiende a casos Multinomial u Ordinal cuando hay más de dos categorías.



# Supuesto 2: Independencia de Observaciones



## Requisito Fundamental

Cada observación o punto de datos debe ser **independiente** de los demás. No debe existir correlación entre las observaciones.

## Prevención de Autocorrelación

Este supuesto asegura que los errores del modelo no estén relacionados entre sí, lo cual es crítico para estimaciones precisas.

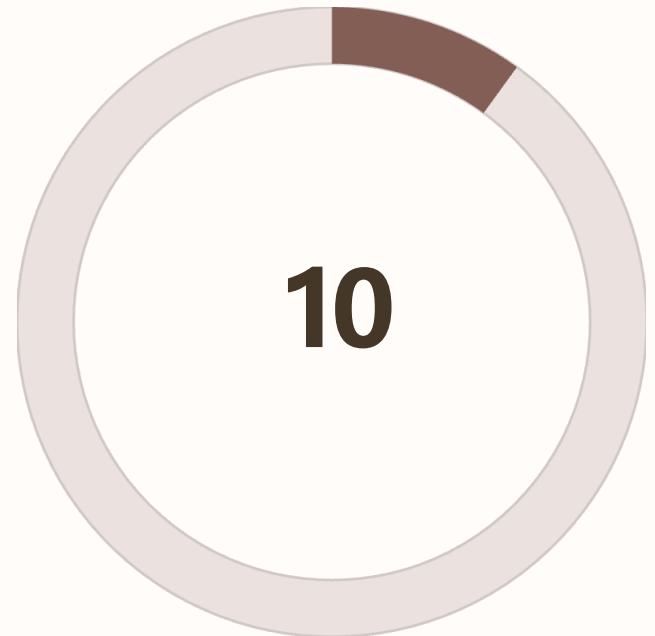
## Impacto de Violación

Si se viola este supuesto, los errores estándar de los coeficientes serán incorrectos, comprometiendo la validez de las inferencias estadísticas.



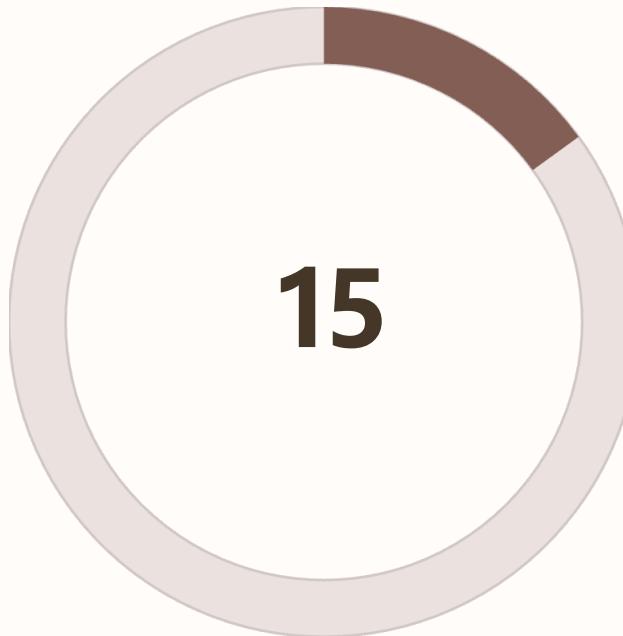


# Supuesto 3: Muestra Grande y Datos Suficientes



**Regla General**

Mínimo de 10 eventos por variable predictora



**Ideal**

Al menos 15 casos por predictor para mayor estabilidad

## Tamaño Muestral

Se requiere una muestra de tamaño suficiente para obtener estimaciones de coeficientes fiables y estables. Muestras pequeñas pueden producir coeficientes inestables.

## Multicolinealidad

Las variables predictoras no deben estar altamente correlacionadas entre sí. Se recomienda verificar con el **Factor de Inflación de Varianza (VIF)**.



# Aplicaciones: Sector Financiero y Crédito



## Predicción de Impago

Evaluación del riesgo crediticio mediante *Credit Scoring*. ¿El cliente incumplirá un préstamo (Sí/No)? Análisis basado en historial de pagos, ingresos y comportamiento financiero.



## Aprobación de Crédito

Determinar la probabilidad de aprobación de préstamos o tarjetas de crédito en base a historial crediticio, puntaje FICO, y otros factores de riesgo.



## Detección de Fraude

Clasificar transacciones en tiempo real como fraudulentas o legítimas, analizando patrones de comportamiento, ubicación, monto y frecuencia de operaciones.



# Aplicaciones: Sector Salud y Medicina



## Diagnóstico Binario

¿El paciente tiene la enfermedad X (Sí/No)? Clasificación basada en síntomas, resultados de laboratorio y factores de riesgo del paciente.



## Riesgo de Enfermedad

Predecir la probabilidad de desarrollar condiciones médicas (diabetes, enfermedades cardíacas) en función de edad, IMC, presión arterial y otros indicadores.



## Eficacia de Tratamientos

Clasificar si un paciente responderá positivamente a un nuevo medicamento o terapia, optimizando decisiones de tratamiento personalizado.





# Aplicaciones: Marketing y Negocios



- **Churn Prediction**

¿El cliente abandonará el servicio (Sí/No)? Identificación temprana de clientes en riesgo de cancelación para implementar estrategias de retención.

- **Propensión de Compra**

¿El usuario hará clic en un anuncio o comprará un producto? Optimización de campañas publicitarias mediante predicción de conversión.

- **Segmentación de Clientes**

Clasificar clientes en grupos de alto/bajo valor para personalizar ofertas, servicios y estrategias de marketing dirigido.



# Aplicaciones: Ciencia de Datos y Ecología



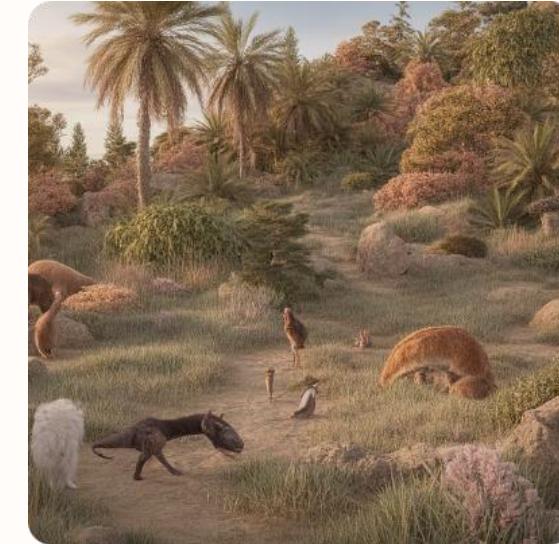
## Reconocimiento de Imágenes

Clasificación inicial de objetos simples (gato vs. perro) como punto de partida en visión por computadora.



## Predicción de Eventos

Modelar la probabilidad de eventos geológicos (terremotos) o climáticos (tormentas severas) basándose en datos históricos.



## Distribución de Especies

Predecir la probabilidad de encontrar una especie en una ubicación dada según variables ambientales y climáticas.



# Ventaja 1: Interpretación y Simplicidad

1

## Fácil de Entender

El modelo es matemáticamente transparente y sus fundamentos son accesibles para profesionales de diversas disciplinas.

2

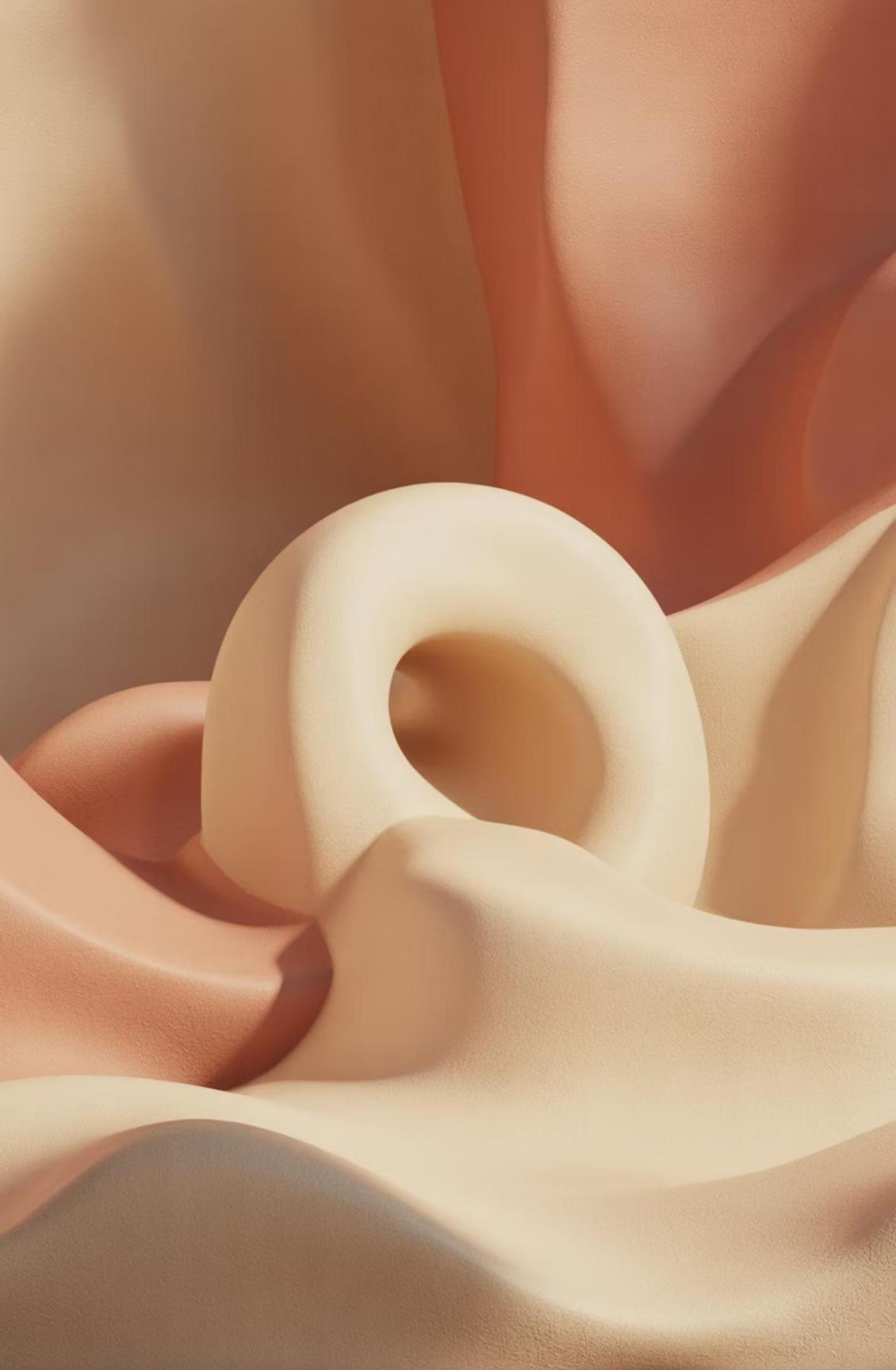
## Interpretación de Coeficientes

Los coeficientes  $\beta$  pueden interpretarse como el cambio en el log-odds por unidad de cambio en X. Un coeficiente positivo indica mayor probabilidad.

3

## Modelo Base

Excelente punto de partida (*baseline*) en problemas de clasificación binaria antes de probar modelos más complejos.





# Ventaja 2: Eficiencia y Velocidad



## Características de Rendimiento

### Bajo Costo Computacional

Es muy eficiente y rápida de entrenar, incluso con grandes conjuntos de datos que contienen millones de observaciones.

### Regularización

Admite técnicas de regularización (L1/L2, Ridge/Lasso) para evitar el sobreajuste y mejorar la generalización del modelo.



# Desventaja 1: Asume Relación Lineal

La principal limitación de la regresión logística es su **asunción fundamental de linealidad** en la escala logit. Cuando la relación real entre las características y la probabilidad del evento es altamente no lineal o compleja, el modelo puede tener un rendimiento subóptimo.

## Restricción Principal

La Regresión Logística puede tener un bajo rendimiento si la relación real entre las características y la probabilidad es altamente no lineal o contiene interacciones complejas.

## Posibles Soluciones

Se puede mitigar creando características no lineales (transformaciones polinómicas, interacciones) o utilizando modelos más complejos como Redes Neuronales, SVM o árboles de decisión.



# Desventaja 2: Sensibilidad y Precisión

## Sensibilidad a Outliers

Es sensible a los valores atípicos (*outliers*) que pueden distorsionar significativamente los coeficientes estimados y afectar las predicciones.

## Multicolinealidad

La presencia de predictores altamente correlacionados puede inflar los errores estándar y hacer que los coeficientes sean inestables o difíciles de interpretar.

## Separación Completa

Si las clases son perfectamente separables, los coeficientes pueden volverse infinitos (problema de separación completa o quasi-completa).

## Modelos Más Potentes

En muchos casos, modelos como XGBoost, Random Forest o Redes Neuronales superan su rendimiento en términos de precisión predictiva.



# Regresión Logística Binaria

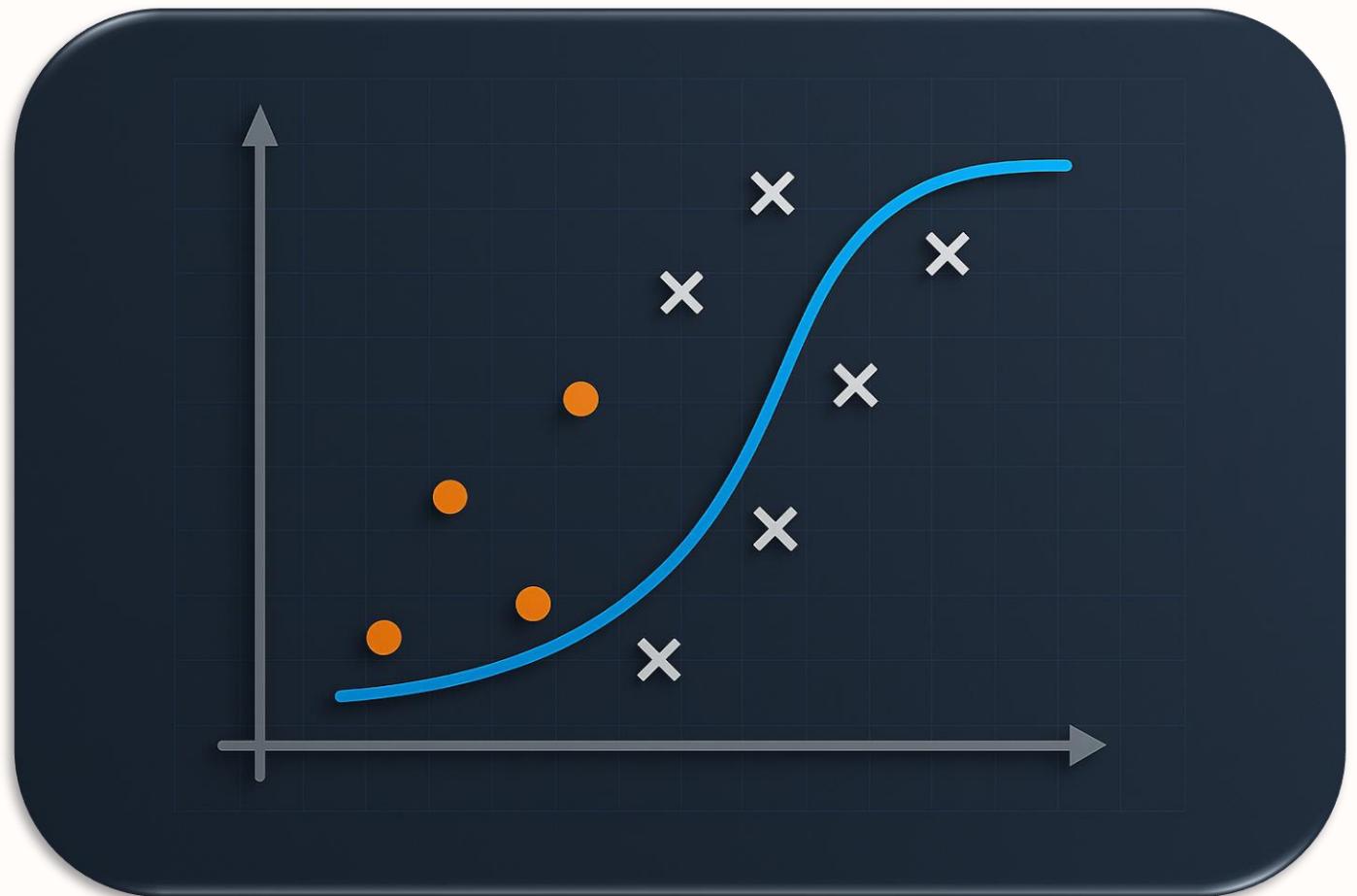
## Características Principales

- **Objetivo:** Clasificar entre dos resultados mutuamente excluyentes (0 o 1)
- **Modelo:** Usa una única función sigmoide
- **Salida:** Probabilidad  $P(Y=1 | X)$

## Ejemplo Clásico

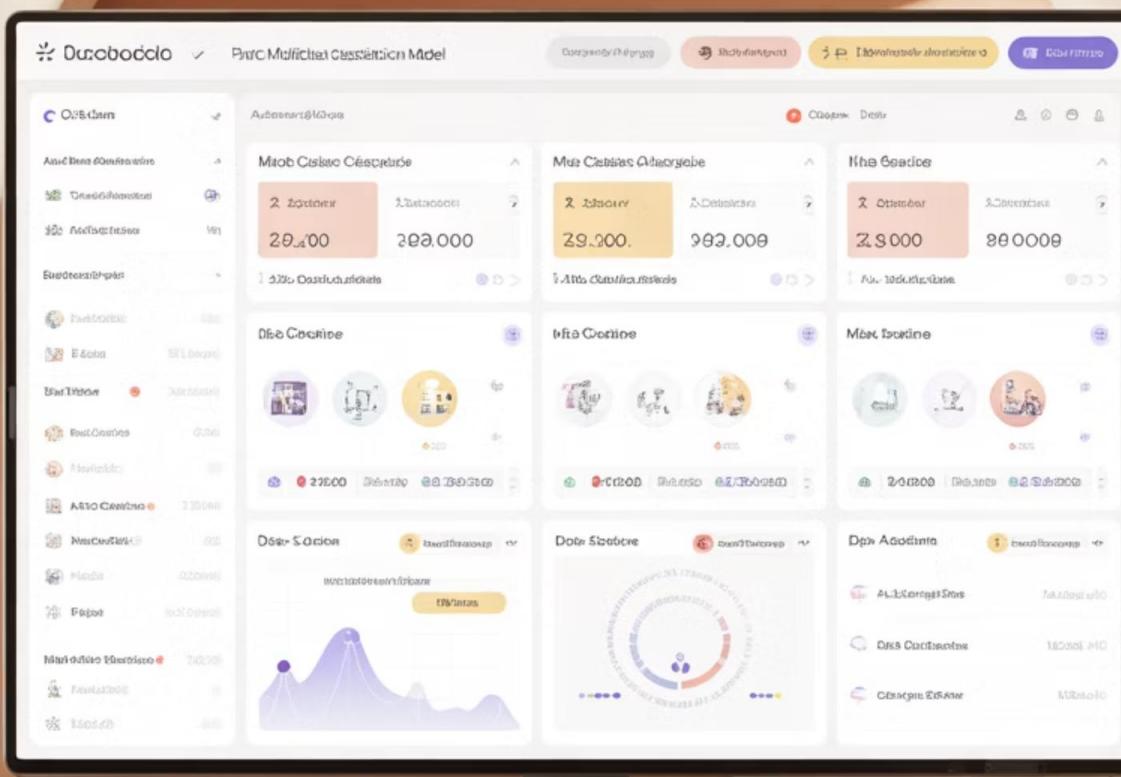
¿El correo electrónico es **Spam** o **No Spam**?

El modelo analiza características como palabras clave, remitente, enlaces sospechosos y genera una probabilidad de spam.





# ¿Qué pasa con más de dos clases?



1

## El Problema

La Regresión Logística Binaria no puede manejar resultados con tres o más categorías (ej: Clase A, B, C) directamente.

2

## Solución 1: Multinomial

Para categorías nominales sin orden inherente

3

## Solución 2: Ordinal

Para categorías con orden natural



# Regresión Logística Multinomial

También conocida como **Logit Polinomial**, esta variante maneja clasificaciones con múltiples categorías nominales.



## Objetivo

Clasificar en más de dos clases nominales (sin orden inherente)



## Mecanismo del Modelo

Estima  $K-1$  modelos binarios independientes, donde  $K$  es el número de clases



## Clase de Referencia

Selecciona una clase base y modela el log-odds de cada clase restante contra ella

**Ejemplo:** Clasificar tipos de uva en una bodega: Malbec, Cabernet Sauvignon, Merlot, Chardonnay.



# Regresión Logística Ordinal

## Logit de Probabilidades Proporcionales

Esta variante está diseñada específicamente para variables de respuesta con categorías ordenadas.

### Características:

- **Objetivo:** Clasificar en más de dos clases ordinales (con orden inherente)
- **Modelo:** Estima un único conjunto de coeficientes para los predictores
- **Puntos de Corte:** Utiliza múltiples umbrales (*cutpoints*) entre categorías

### Ejemplo Típico

#### Calificaciones de Satisfacción:

1. Muy Malo
2. Malo
3. Bueno
4. Muy Bueno





# Multinomial vs. Ordinal: La Decisión

## Usar Multinomial

**Cuando:** Las categorías no tienen un orden lógico o natural

**Ejemplos:** Color (rojo, azul, verde), profesión (médico, ingeniero, profesor), tipo de producto

## Usar Ordinal

**Cuando:** Las categorías tienen un orden natural y significativo

**Ejemplos:** Nivel educativo (primaria, secundaria, universidad), ranking (bajo, medio, alto), gravedad de enfermedad

- ❑ **Advertencia:** Usar Multinomial en datos ordinales ignora la valiosa información de orden y reduce la potencia estadística del análisis.



# Pasos para la Implementación

## Cargar Datos

Usar Pandas para leer el conjunto de datos desde archivos CSV, Excel, bases de datos SQL o APIs.

```
import pandas as pd  
df = pd.read_csv('datos.csv')
```

## Exploración Inicial

Analizar la estructura de datos, tipos de variables, valores faltantes y distribuciones usando `df.info()`, `df.describe()` y `df.head()`.

## Manejo de Valores Faltantes

Imputar o eliminar datos faltantes según la estrategia adecuada para tu problema (media, mediana, moda, o eliminación).

## Codificación de Variables

Convertir variables categóricas en numéricas mediante One-Hot Encoding (`pd.get_dummies()`) o Label Encoding.

## Escalado de Características

Normalizar las variables numéricas usando `StandardScaler` o `MinMaxScaler`. Aunque no es estrictamente obligatorio, mejora la convergencia del modelo.



# Python IA





## TEMARIO

- 1 — Regresión logística
- 2 — Regresión y clasificación con Árboles de Decisión
- 3 — Regresión y clasificación con Árboles de Decisión



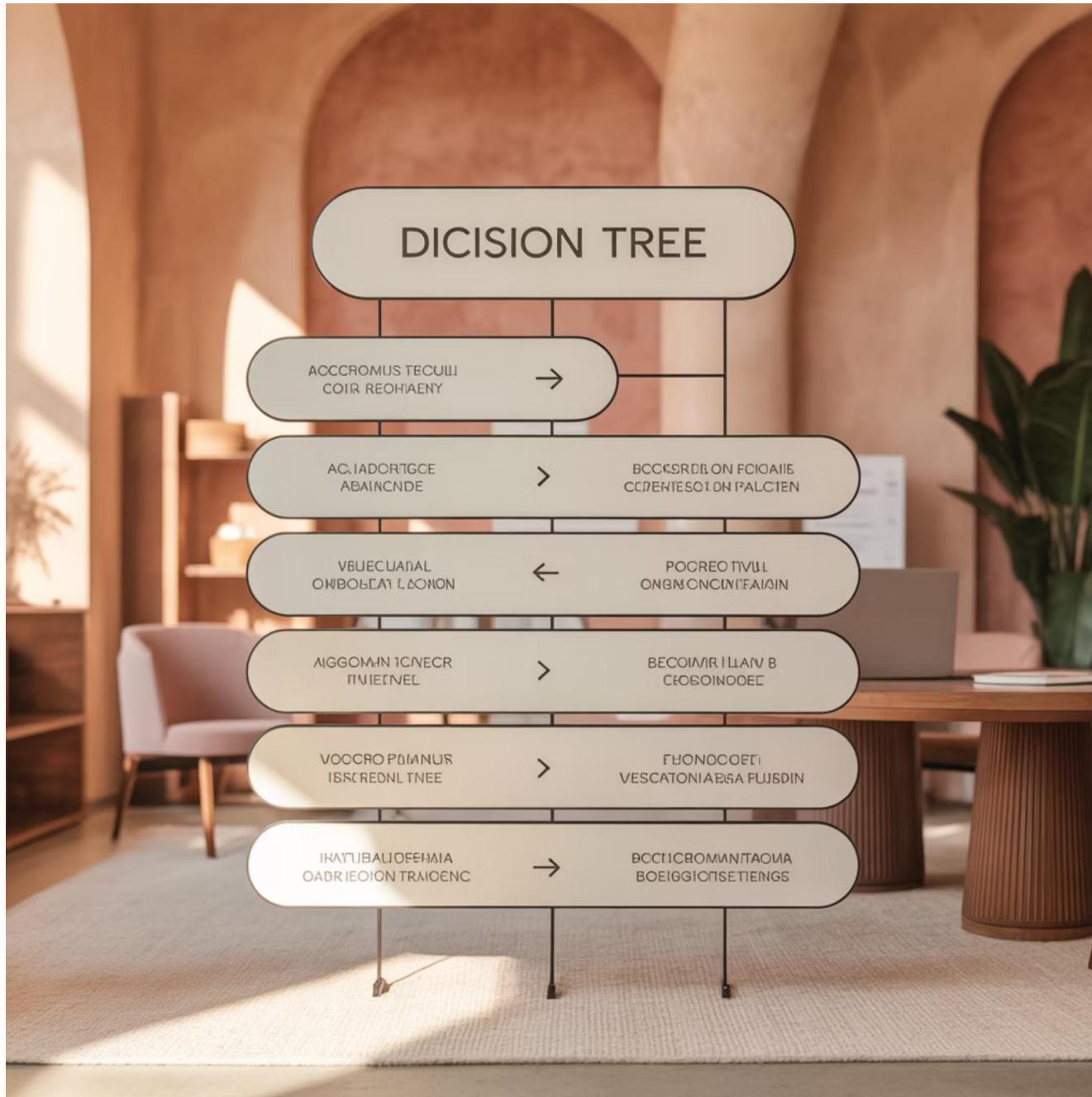


# Regresión y Clasificación con Árboles de Decisión

Un viaje completo desde los fundamentos hasta la aplicación práctica



# ¿Qué es un Árbol de Decisión?



Un **árbol de decisión** es un algoritmo de aprendizaje supervisado que utiliza una estructura jerárquica similar a un diagrama de flujo para tomar decisiones basadas en características de los datos.

Funciona dividiendo iterativamente el conjunto de datos en subconjuntos más pequeños basándose en las características más informativas, creando un modelo interpretable y visual.



# Clasificación vs. Regresión

## Clasificación

### Predicción de categorías

- Salida: Etiquetas discretas
- Ejemplo: Clima → "Frío" o "Caluroso"
- Las hojas contienen clases

## Regresión

### Predicción de valores numéricos

- Salida: Valores continuos
- Ejemplo: Temperatura → 31°C
- Las hojas contienen promedios



## Componentes Clave del Árbol

1

### Nodo Raíz

El punto de partida que representa el conjunto completo de datos antes de cualquier división

2

### Nodos Internos

Realizan pruebas sobre atributos específicos para determinar la siguiente ramificación

3

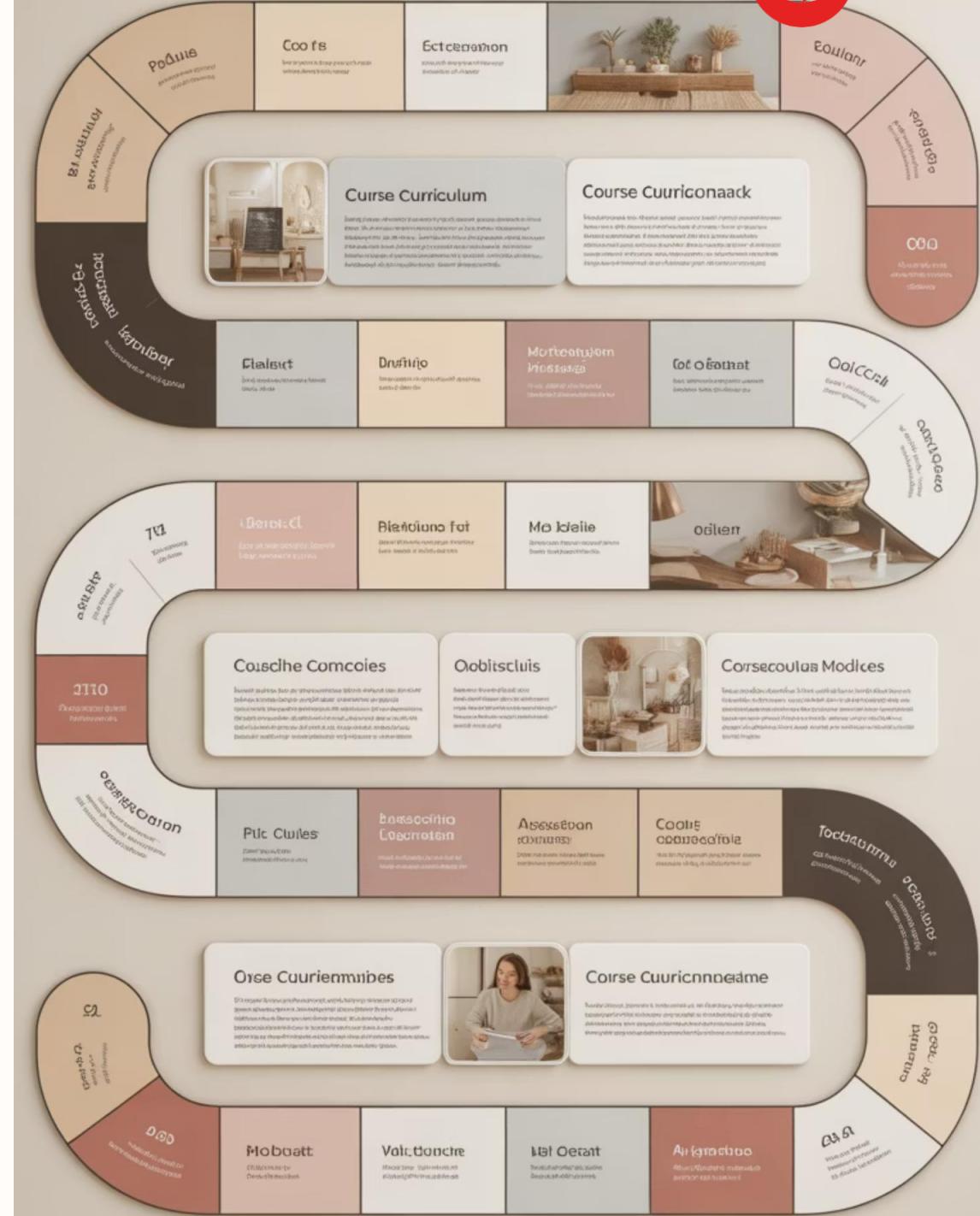
### Ramas

Representan los posibles resultados de las pruebas realizadas en los nodos internos

4

### Hojas

Nodos terminales que contienen las predicciones finales del modelo



# Árboles de decisión - Clasificación

# Árboles de decisión - Clasificación – Definición - ¿Cómo Funciona?



Ataque



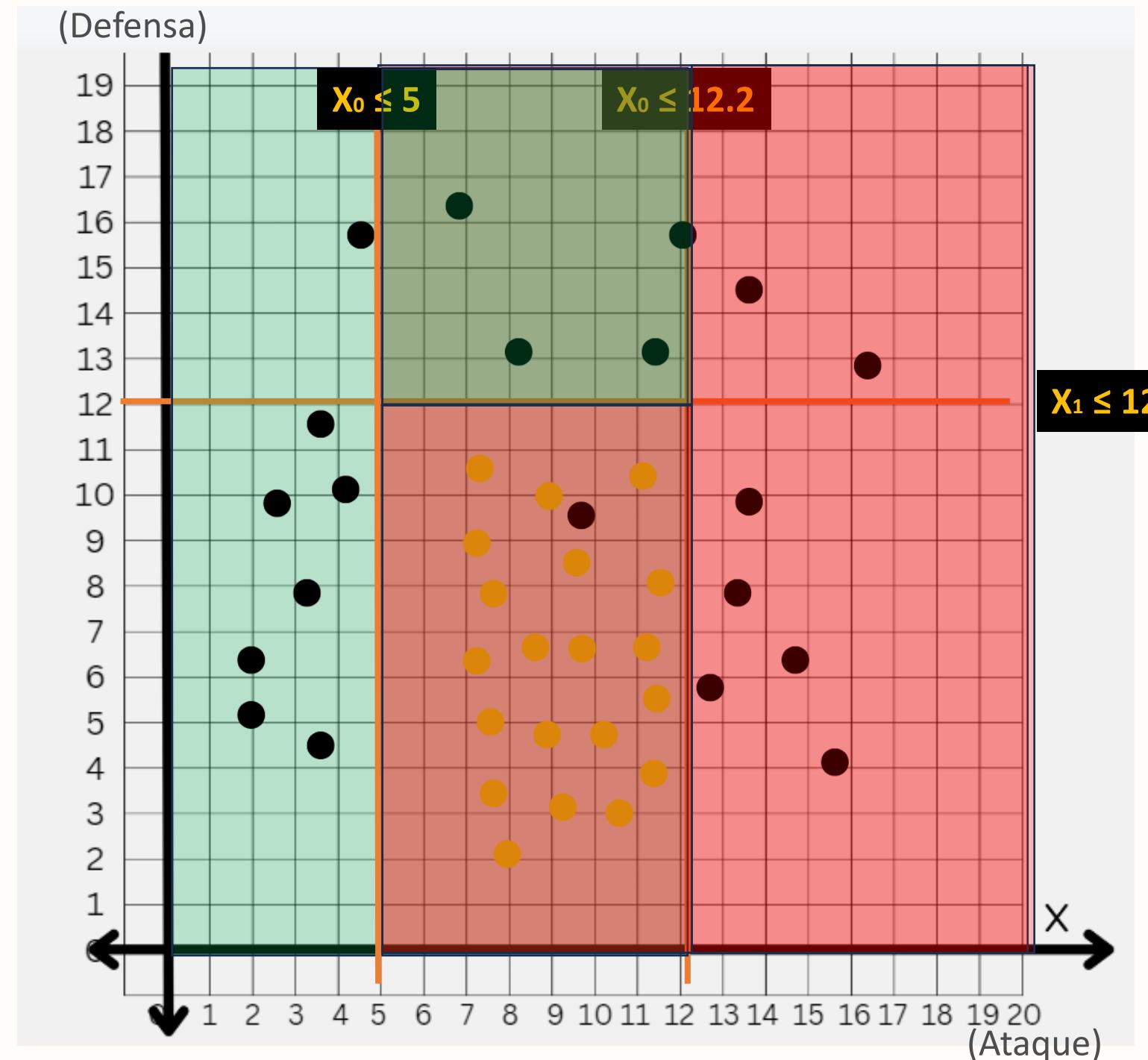
Ataque

Defensa

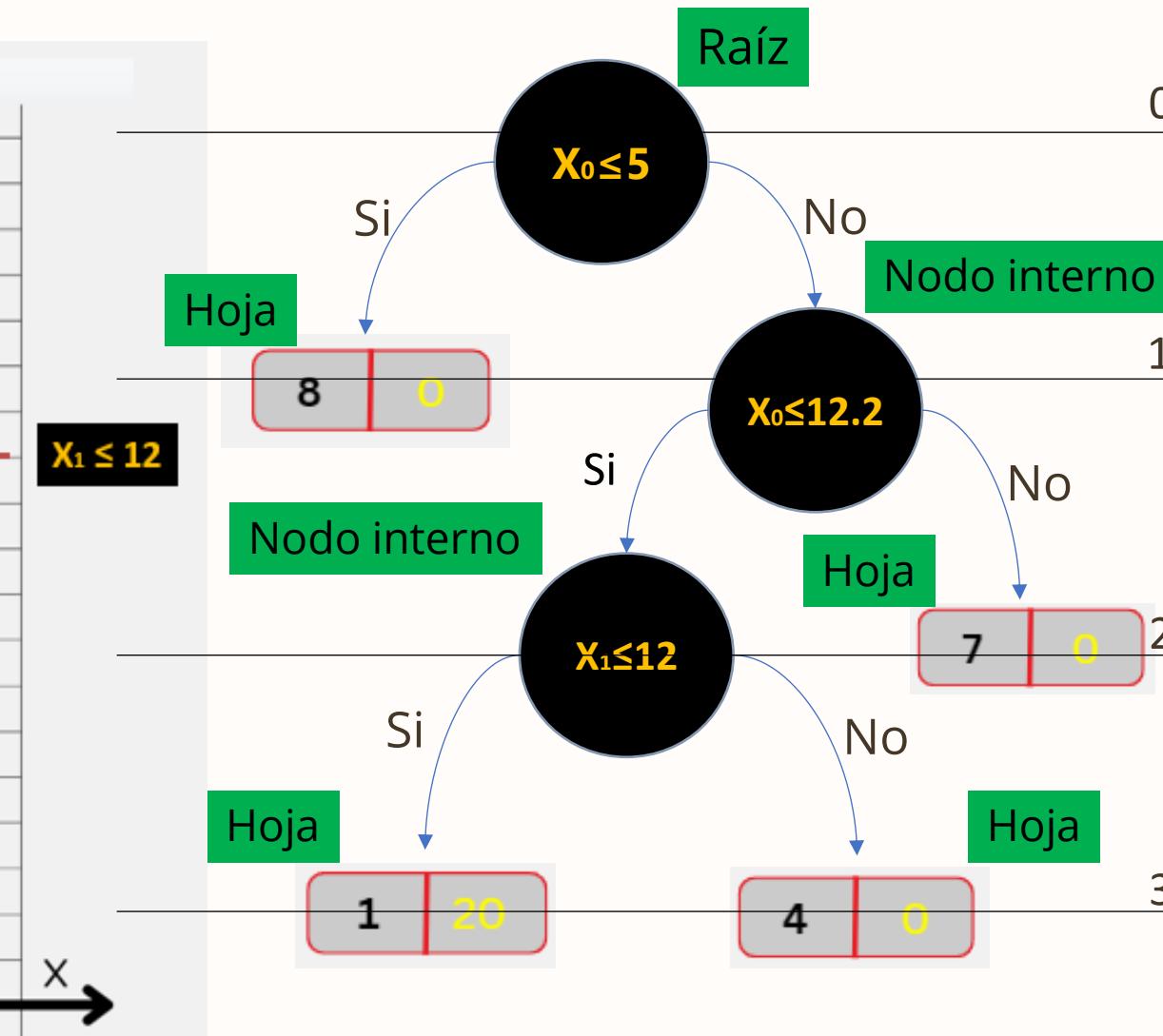
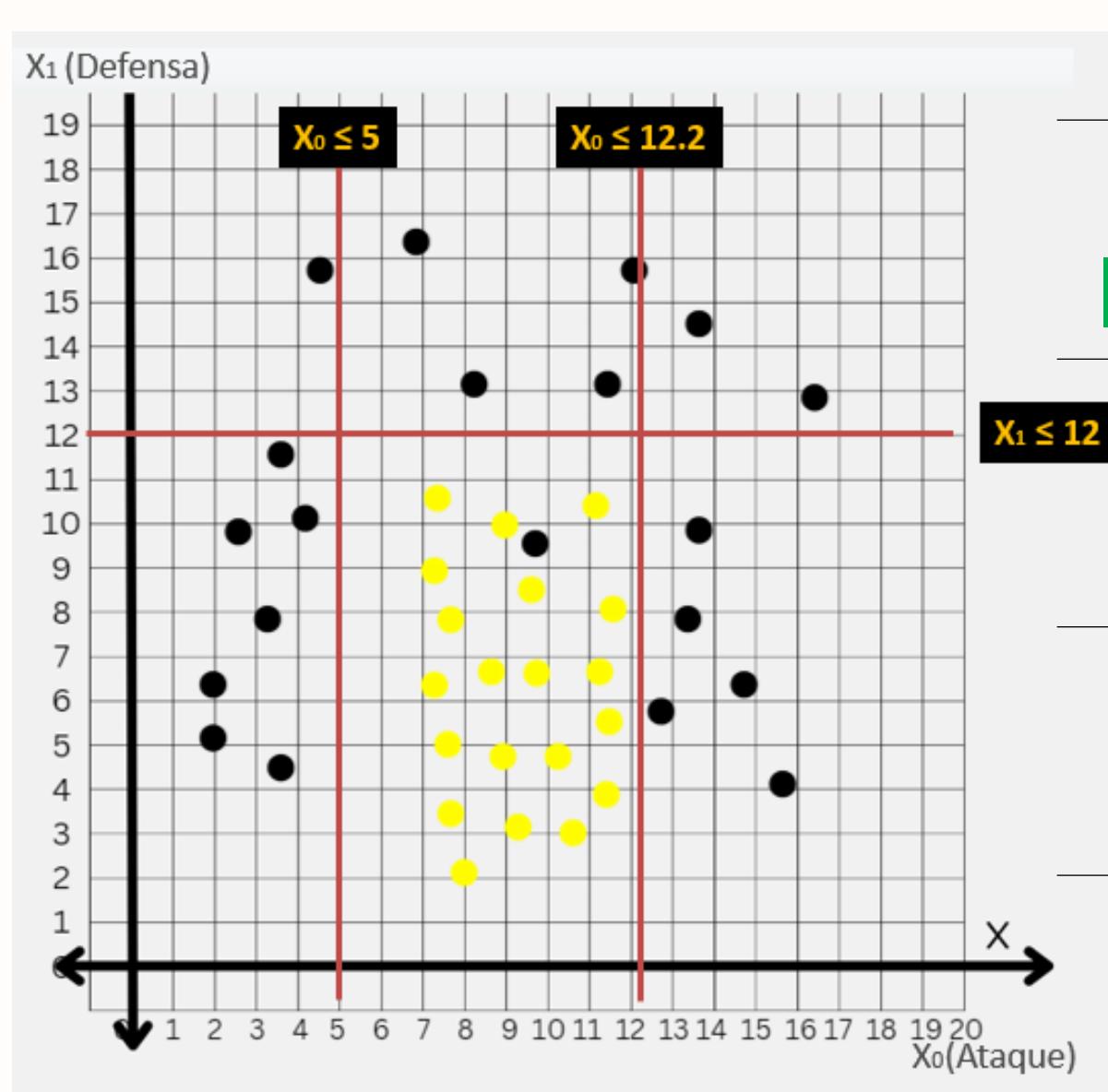


Defensa

# Árboles de decisión – Clasificación – Definición - ¿Cómo Funciona?

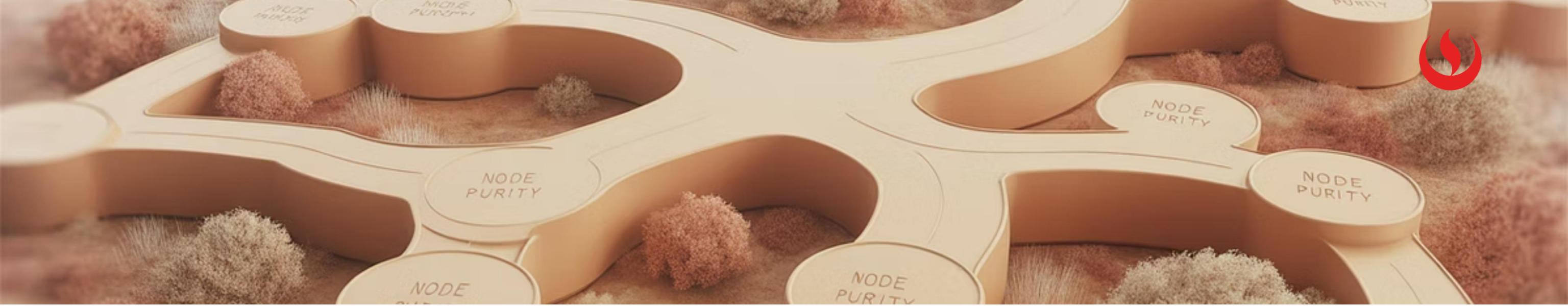


# Árboles de decisión – Clasificación – Definición - ¿Cómo Funciona?



## Selección de nodos

- Ganancia de información
- Gini index



## Pureza de Nodos en Clasificación

La clave para construir un árbol efectivo es saber **cómo dividir los datos** en cada nodo. Buscamos crear subconjuntos lo más "puros" posible, donde la mayoría de las muestras pertenezcan a una misma clase.



### Nodo Puro

Todas las muestras pertenecen a una única clase  
(100% homogéneo)



### Nodo Impuro

Las muestras están mezcladas entre diferentes clases  
(heterogéneo)



# Entropía: Midiendo el Desorden

La **entropía** es una medida de la impureza o desorden en un conjunto de datos. Cuanto mayor es la entropía, más mezcladas están las clases.

$$H(D) = - \sum_{i=1}^c p_i \log_2(p_i)$$

Donde  $p_i$  es la proporción de muestras de la clase  $i$  y  $c$  es el número total de clases.

## Ejemplo de Entropía

Imagina un conjunto de datos de clasificación con 3 "Ángeles" y 3 "Demonios":

$$H(D) = - \left( \frac{3}{6} \log_2 \left( \frac{3}{6} \right) + \frac{3}{6} \log_2 \left( \frac{3}{6} \right) \right) = 1$$

**Entropía = 0:** El conjunto de datos está completamente puro, todos son de una sola clase.

**Entropía = 1:** El conjunto de datos tiene la máxima impureza, distribución equilibrada de clases.



**Valores de Entropía:**  $H = 0$  indica pureza total (todas las muestras de una clase), mientras que  $H$  es máxima cuando las clases están uniformemente distribuidas.



## Ganancia de Información

La **Ganancia de Información (Information Gain)** es una medida que se utiliza para determinar qué atributo dividir en un nodo, buscando maximizar la reducción de la incertidumbre (Entropía) e sobre la variable objetivo.

Medida de la reducción de la entropía al dividir el conjunto de datos en función de un atributo.



### Entropía Padre

Desorden antes de la división

### Entropía Hijos

Desorden promedio después de dividir

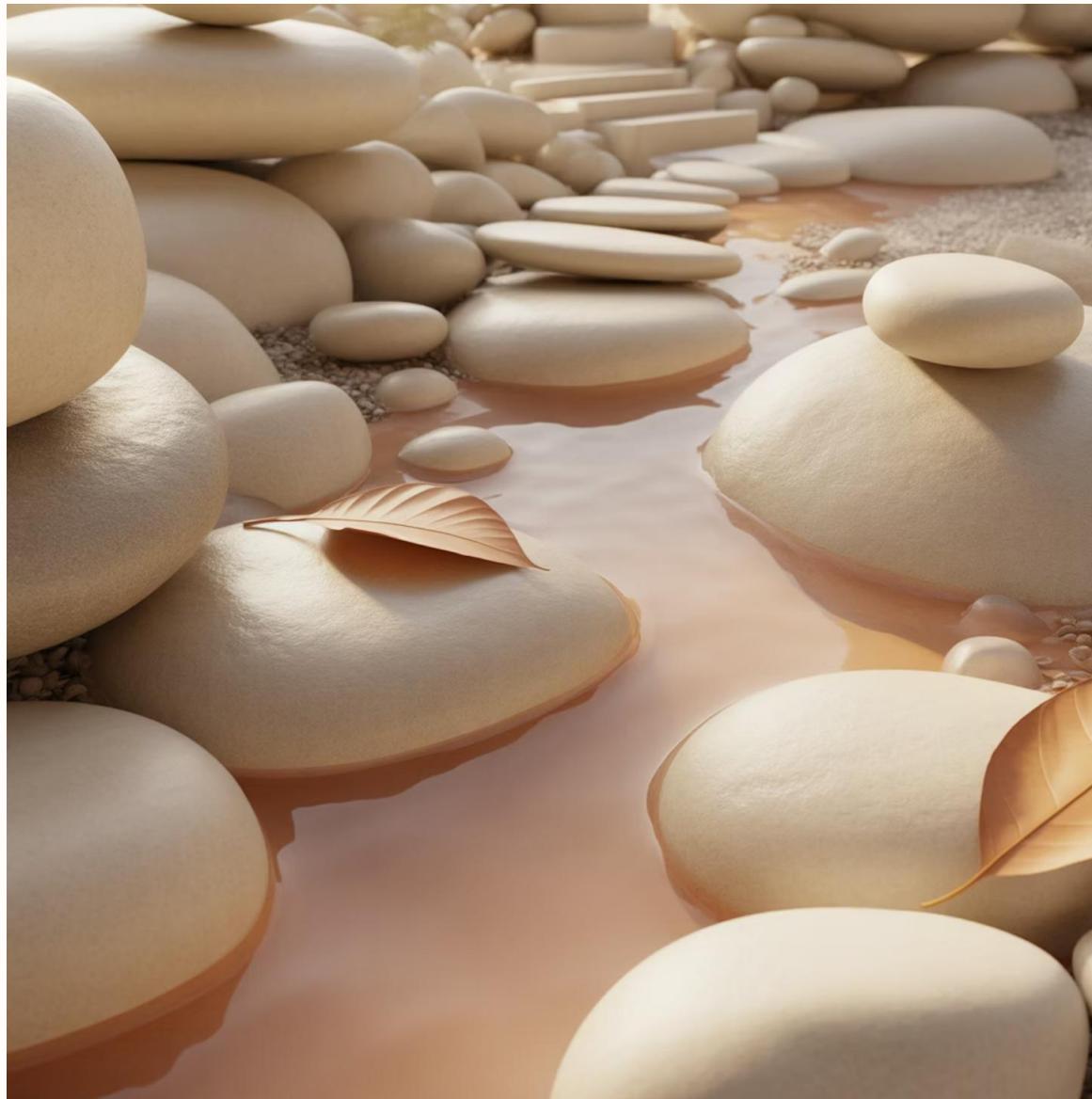
### Ganancia

Diferencia = Reducción del desorden

La característica con mayor ganancia de información se convierte en el nodo de decisión.



# Índice Gini: Alternativa a la Entropía



El **Índice Gini** mide la probabilidad de clasificar incorrectamente una muestra elegida aleatoriamente. Es computacionalmente más eficiente que la entropía.

$$Gini(D) = 1 - \sum_{i=1}^c p_i^2$$

**p:** es la proporción de observaciones de cada clase en el nodo.

**Gini = 0:** Indica máxima pureza (todas las muestras en el nodo pertenecen a la misma clase).

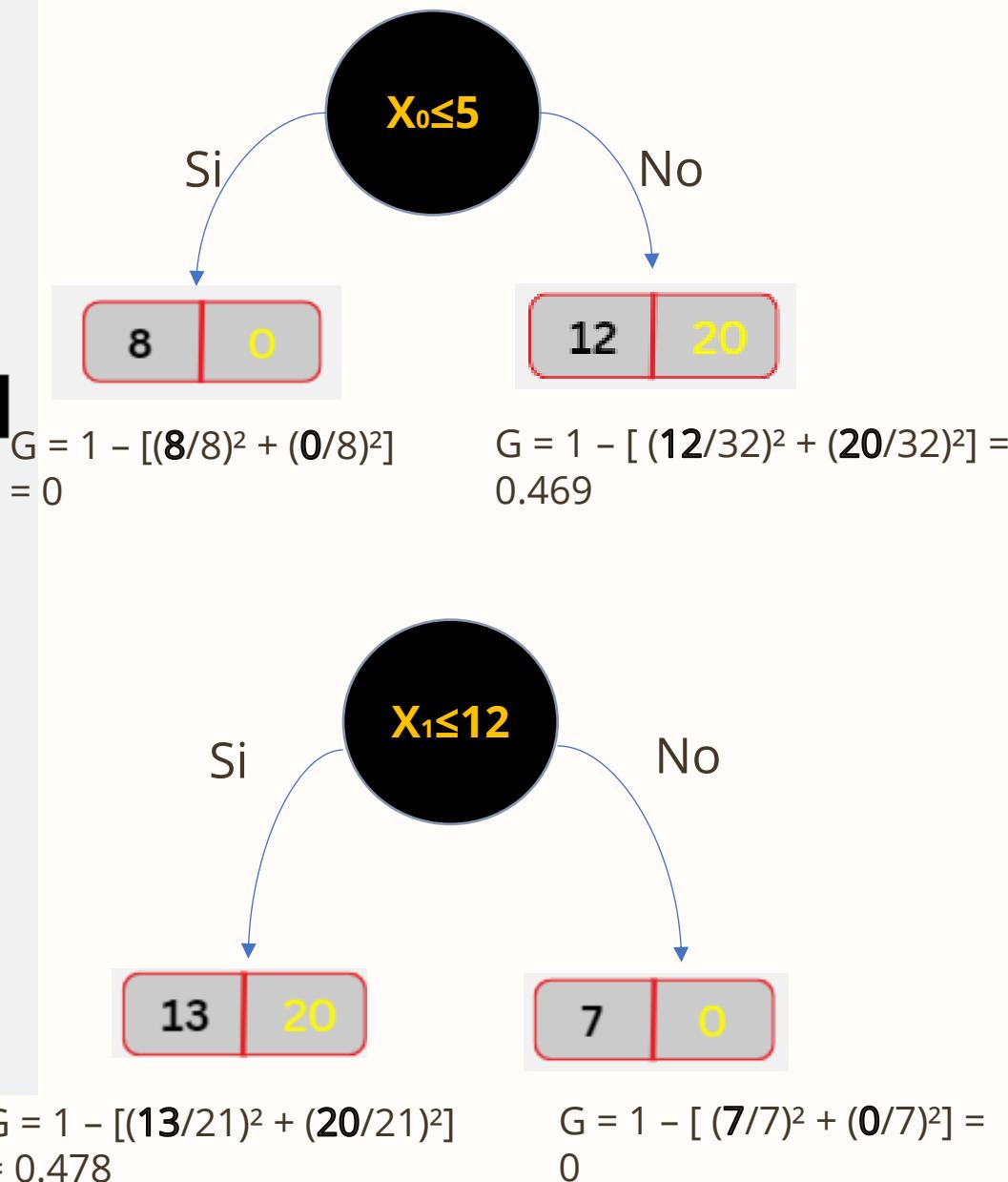
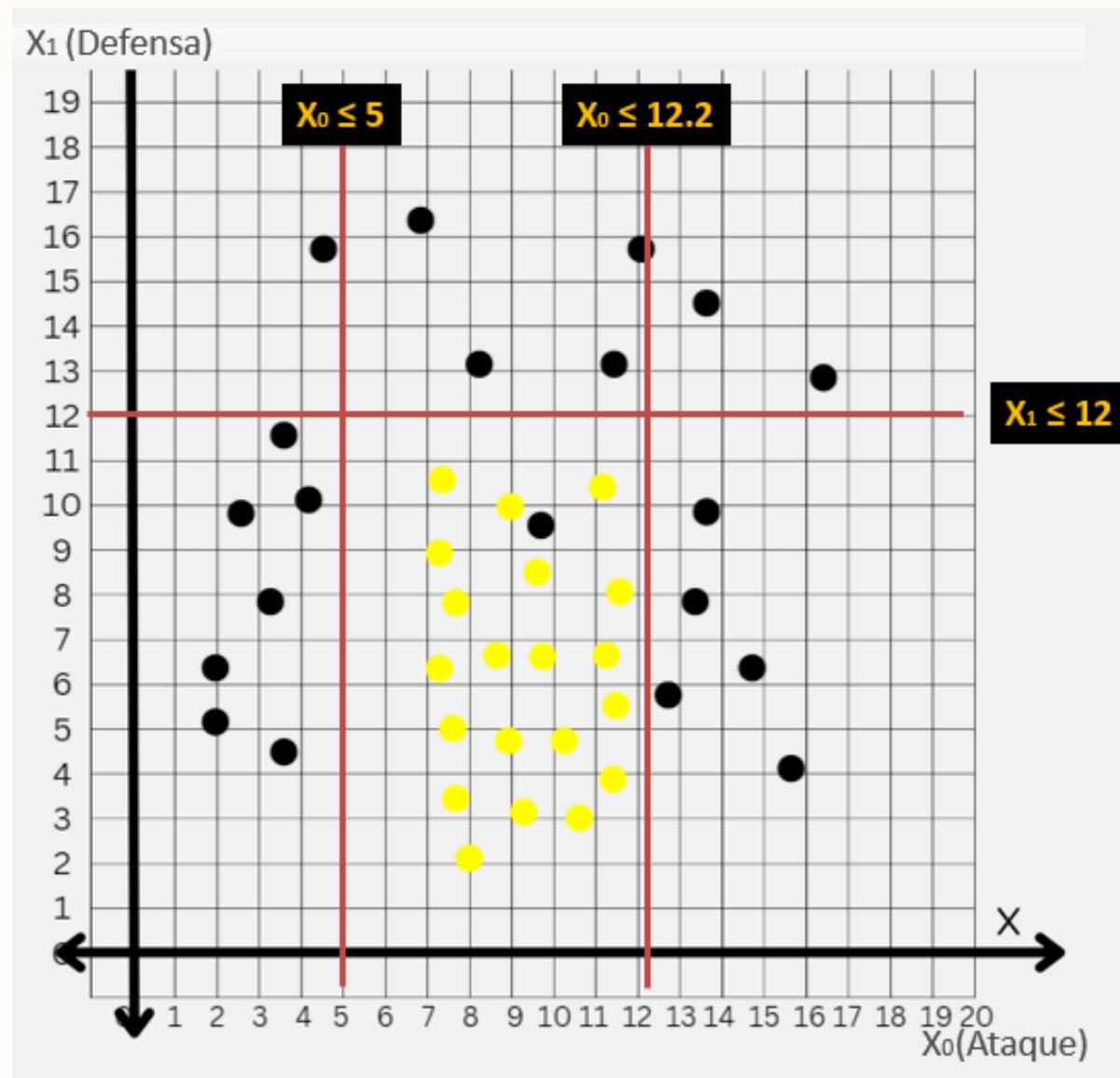
**Gini cercano a 0.5:** Representa una mezcla equilibrada de clases.

**Gini = 1:** Indica una mezcla total y desorganizada de clases

Valores cercanos a 0 indican alta pureza, mientras que valores más altos indican mayor impureza en el nodo.



# ¿Cómo funciona el Índice Gini?

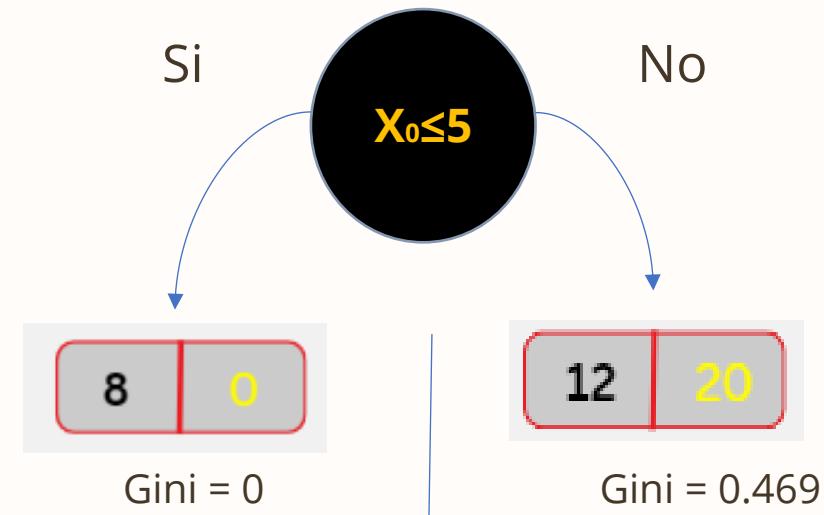




# ¿Cómo funciona el Índice Gini? - Función de costo - Total

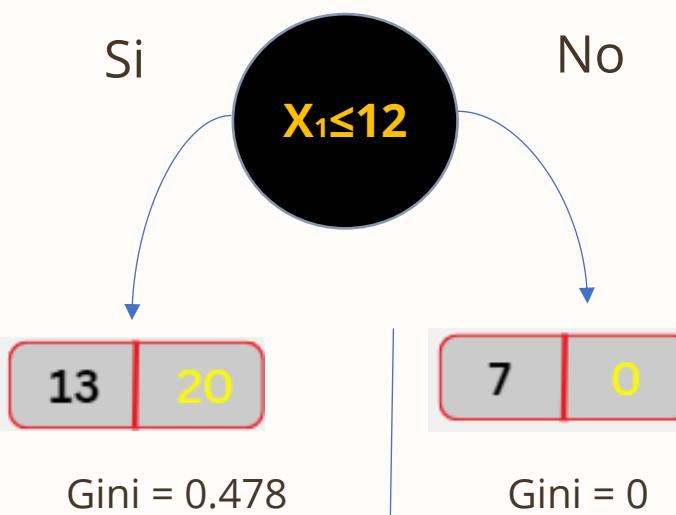
Impureza = Gini\*((Datos 1 + Datos 2)/número total datos)

Impureza ponderada =  $0 + 0.381 = 0.381$



$$\text{Impureza} = 0 * ((8+0)/40) = 0$$

Impureza ponderada =  $0 + 0.40 = 0.40$



$$\text{Impureza} = 0.478 * ((13+20)/40) = 0.4$$

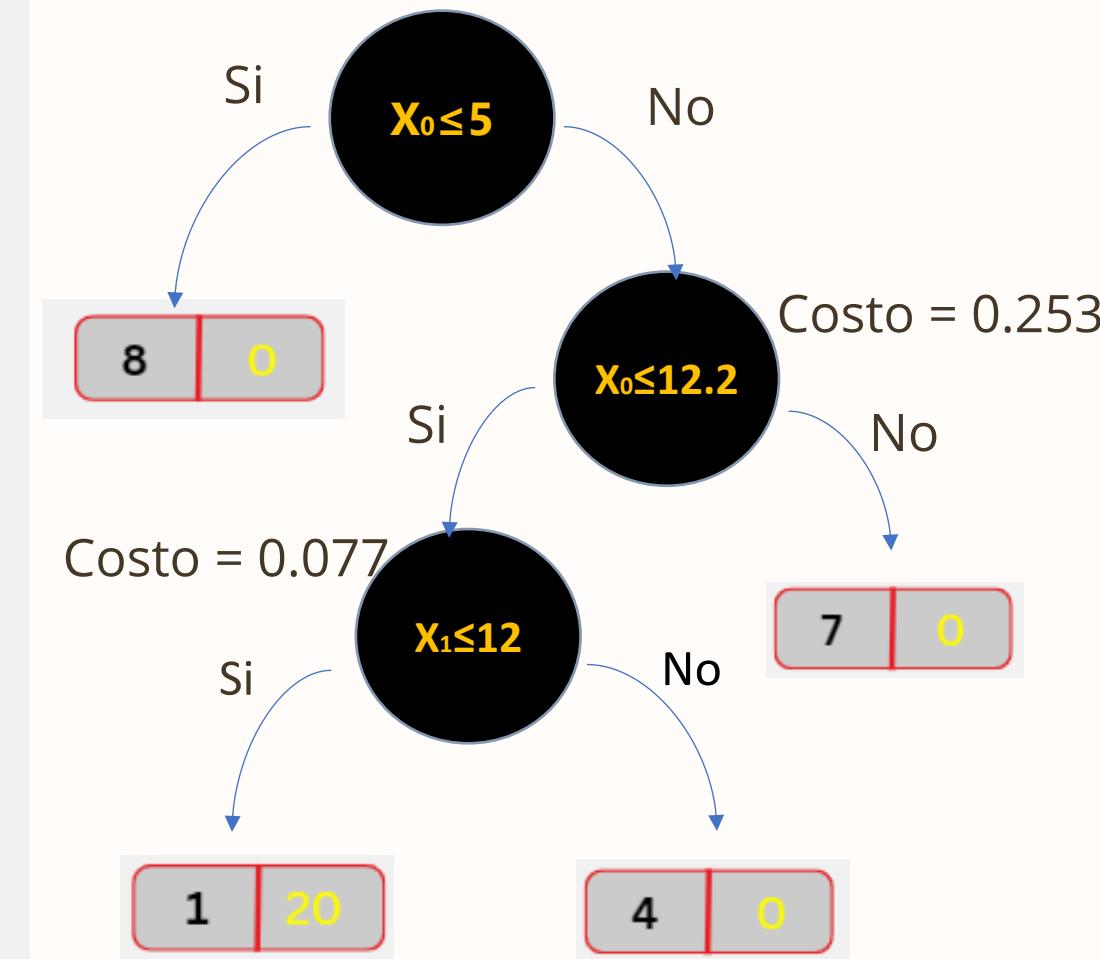
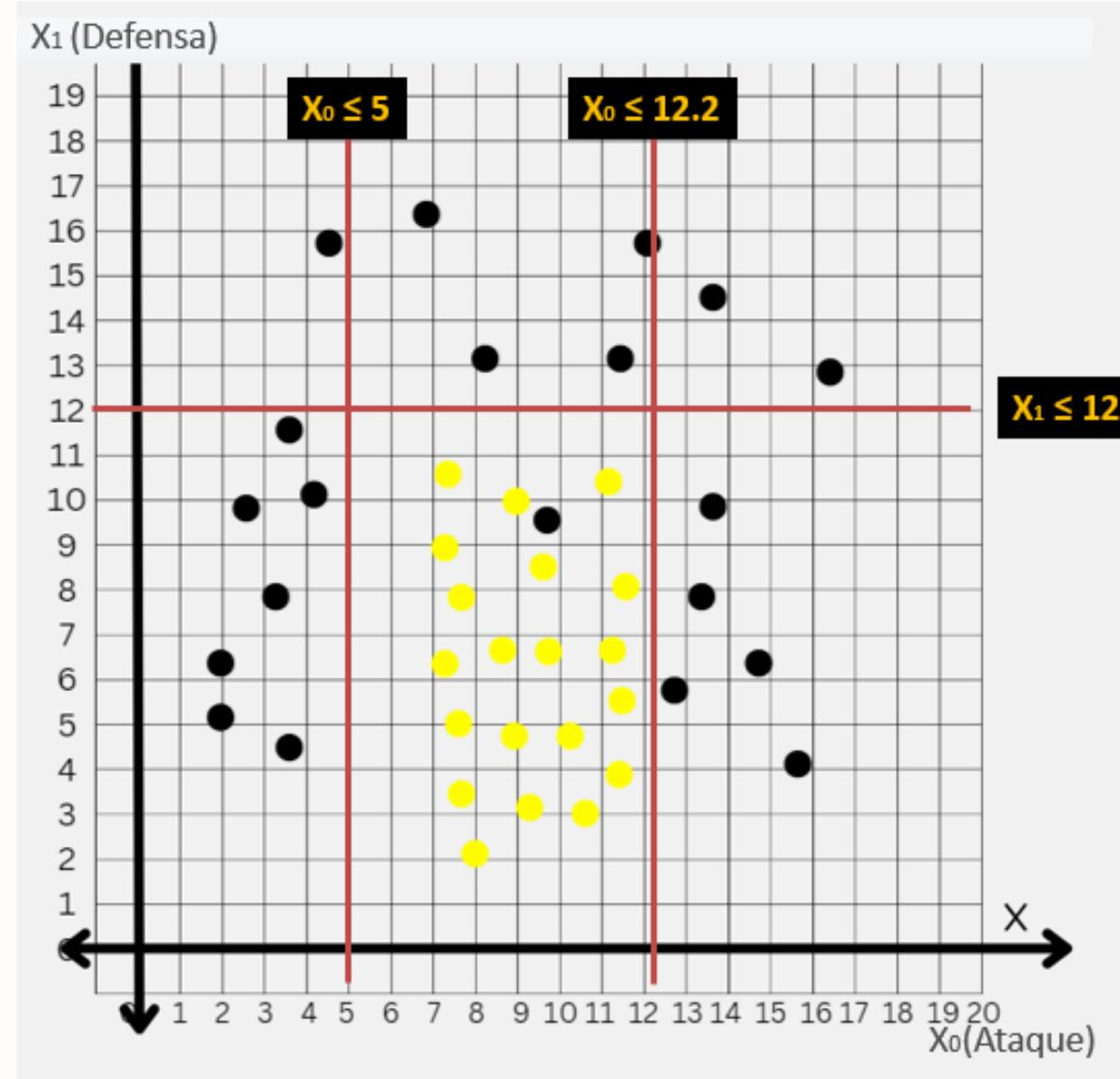
$$\text{Impureza} = 0 * ((7+0)/40) = 0$$



# ¿Cómo funciona el Índice Gini? - Función de costo cada nodo

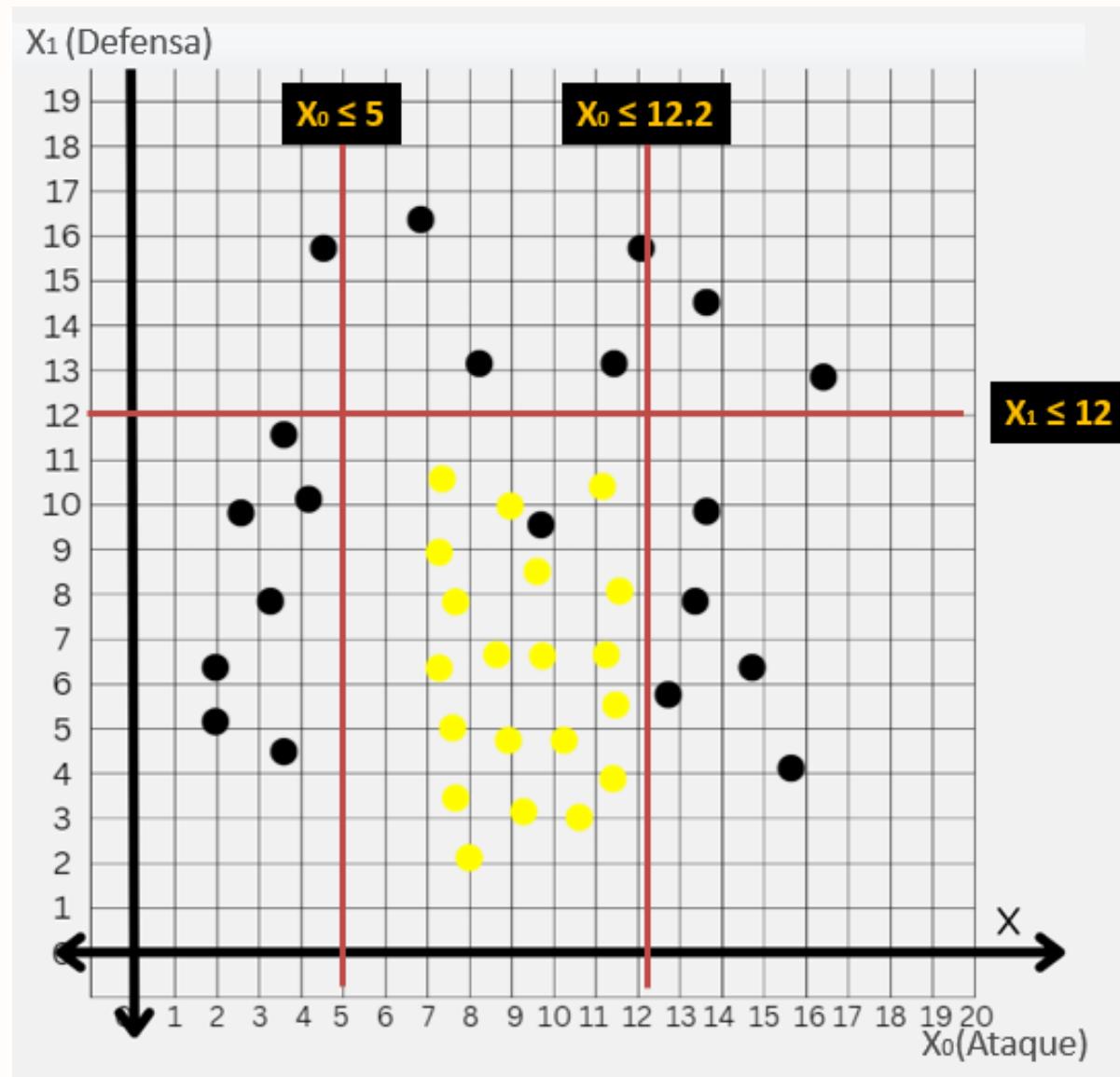
Impureza = Gini\*((Datos 1 + Datos 2)/ **Datos de cada nodo**)

Impureza ponderada (Costo) = 0.381





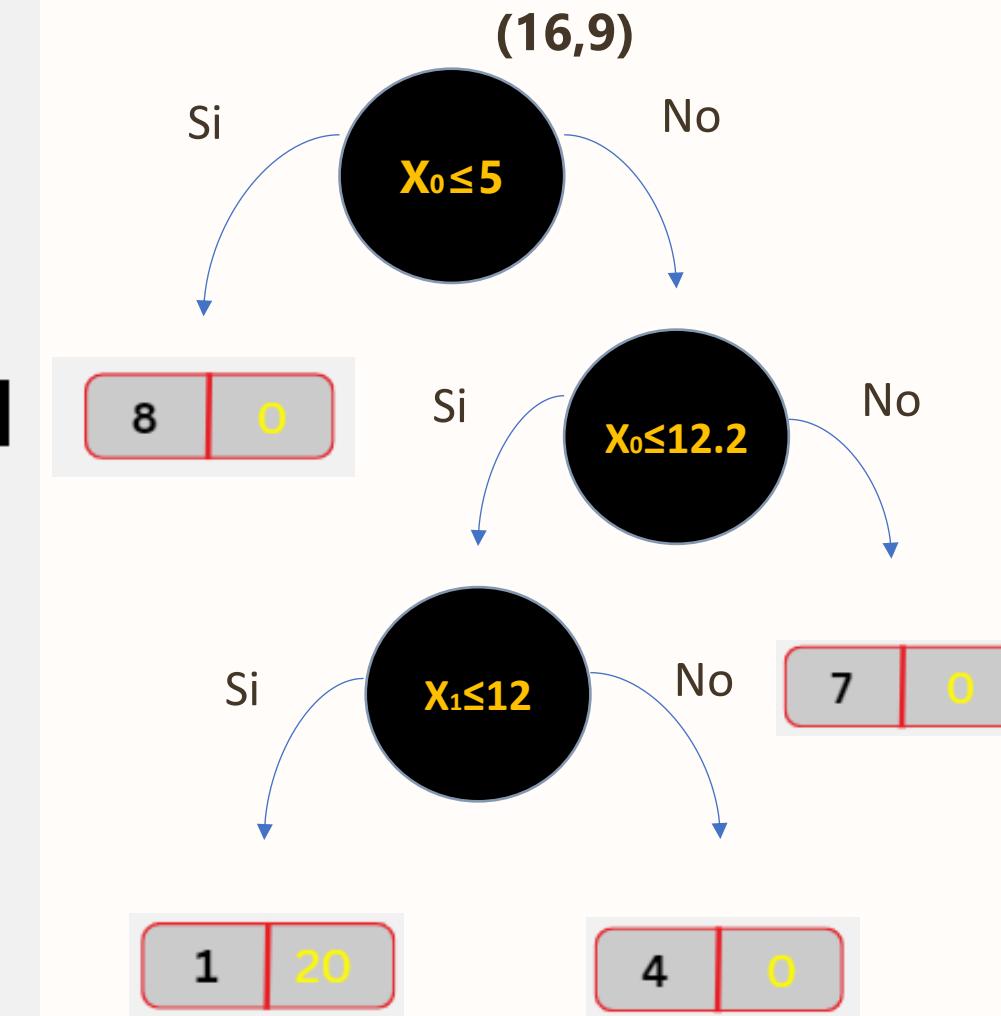
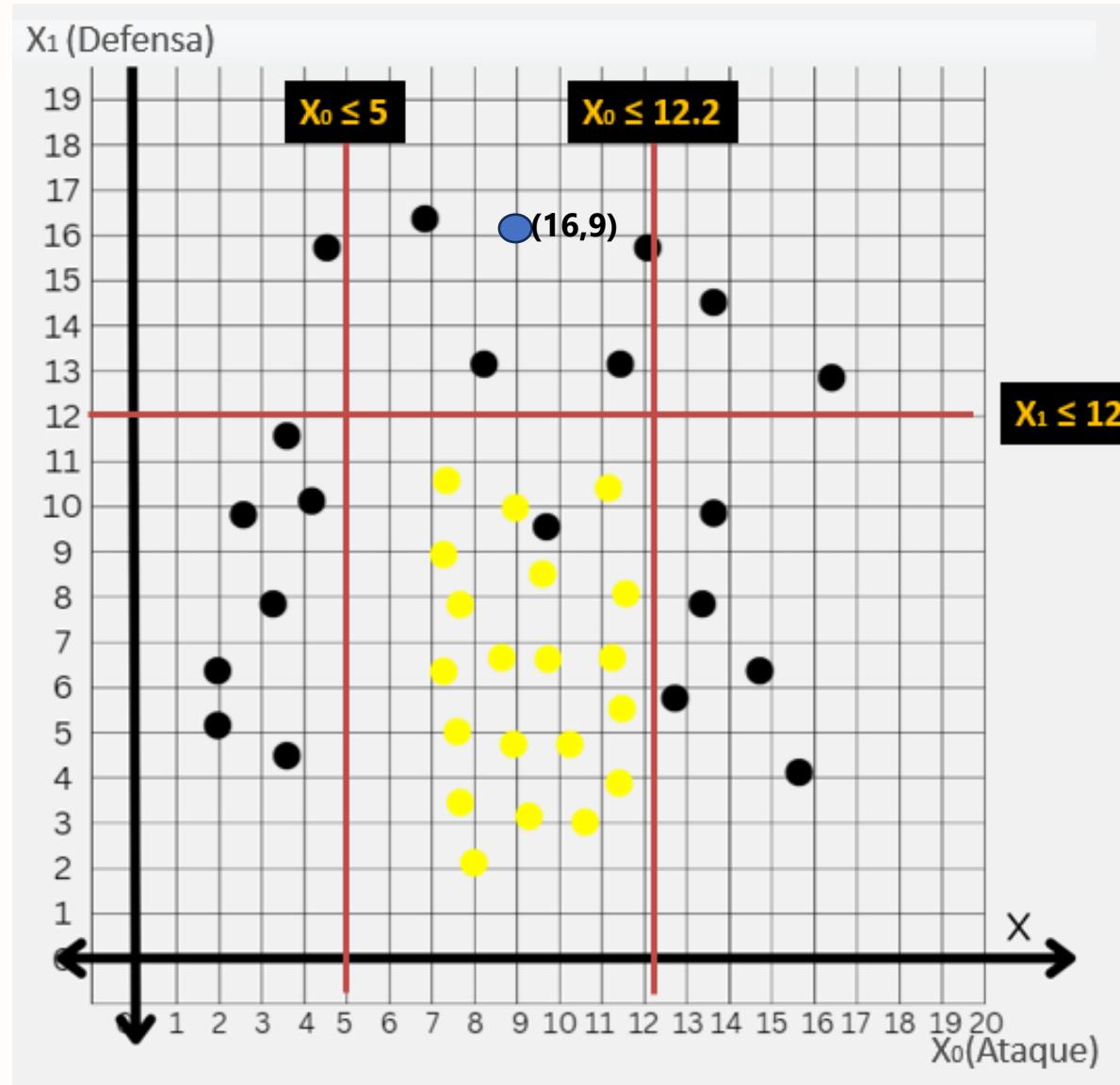
# ¿Cómo funciona el Índice Gini? - Función de costo cada nodo



1. Calcular todos los posibles umbrales disponibles (valores intermedios entre cada par de características adyacentes).  $40 \text{ datos} - 1 = 39 * 2 \text{ características} = 78 \text{ umbrales}$ .
2. Calcular Índices Gini de los hijos, con el objetivo de hallar el costo del nodo del padre.
3. Escoger el umbral que genera el menor costo (el más homogéneo)
4. Se repite hasta 3 veces, según la configuración de parada de los nodos resultantes.



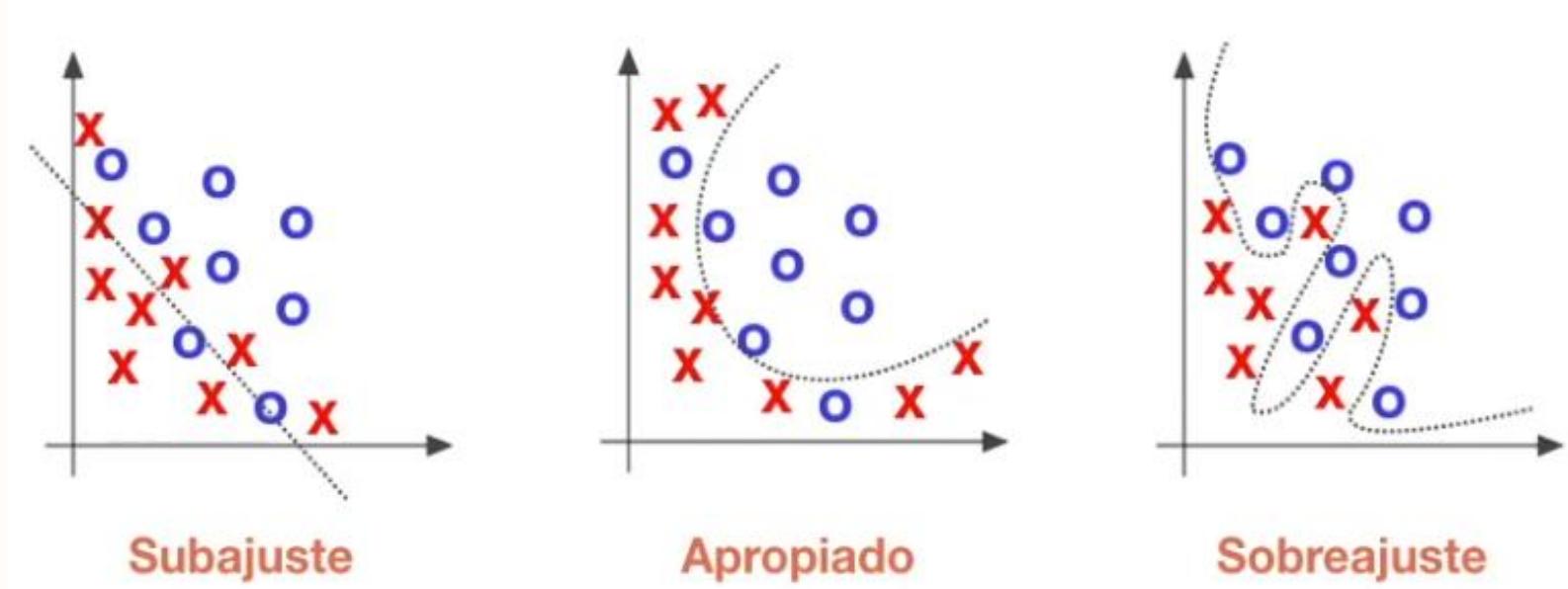
# ¿Cómo funciona el Índice Gini? - Función de costo cada nodo





# Underfitting - Overfitting

El underfitting (o subajuste) ocurre cuando un modelo es demasiado simple para capturar la relación subyacente en los datos. Esto significa que el modelo no logra aprender los patrones importantes de los datos de entrenamiento, resultando en un rendimiento pobre tanto en el conjunto de entrenamiento como en el conjunto de prueba.



El overfitting, o sobreajuste, es cuando un modelo se ajusta demasiado a los datos de entrenamiento, incluyendo el ruido y las variaciones aleatorias, haciendo que no pueda generalizar bien a nuevos datos.

Esto significa que, aunque el modelo tiene un alto rendimiento en el conjunto de entrenamiento, su desempeño disminuye al enfrentarse a datos que no ha visto antes.



# Underfitting - Overfitting

|                | Underfitting   | Just right                                      | Overfitting  |
|----------------|--|---|--|
| Symptoms       | - High training error<br>- Training error close to test error<br>- High bias | - Training error slightly lower than test error | - Low training error<br>- Training error much lower than test error<br>- High variance |
| Regression     |  |   |  |
| Classification |  |   |  |
| Deep learning  |  |   |  |
| Remedies       | - Complexify model<br>- Add more features<br>- Train longer                  |   | - Regularize<br>- Get more data  |

Decision boundaries for overfitted regression and classification models (Source: [Wonseok Shin](#))





# Tratar el Overfitting

- **Profundidad máxima:** limitar cuántos nodos se pueden crear en la estructura del árbol.
- **Mínimo número de datos que debe tener un nodo:** limitar cuántos nodos se pueden crear en la estructura del árbol.
- **Mínimo número de datos que debe tener una hoja:** número mínimo de instancias que un nodo debe contener para considerar realizar una división

## Tratar el Overfitting:

- **Aumentar la Complejidad del Modelo:** Utiliza un modelo más complejo que tenga más capacidad para capturar patrones en los datos.
- **Añadir Más Características:** Incorporar más variables o características relevantes al modelo.
- **Mejorar el Preprocesamiento de Datos** Asegúrate de que los datos están correctamente preprocesados.



# Casos de Estudio



## Caso 1: Aprobación de Crédito

La evaluación del riesgo crediticio es fundamental para las instituciones financieras. Los árboles de decisión permiten automatizar y estandarizar las decisiones de aprobación basándose en múltiples factores del solicitante.



### Objetivo

Predecir si un solicitante debe recibir o no un crédito



### Tipo de Problema

Clasificación Binaria: Aprobar o Rechazar



# Caso 1: Variables del Conjunto de Datos

1

## Ingresos Mensuales

Capacidad de pago del solicitante (variable continua en unidades monetarias)

2

## Historial Crediticio

Categoría: Excelente, Bueno, Regular, Malo (basado en comportamiento de pagos previos)

3

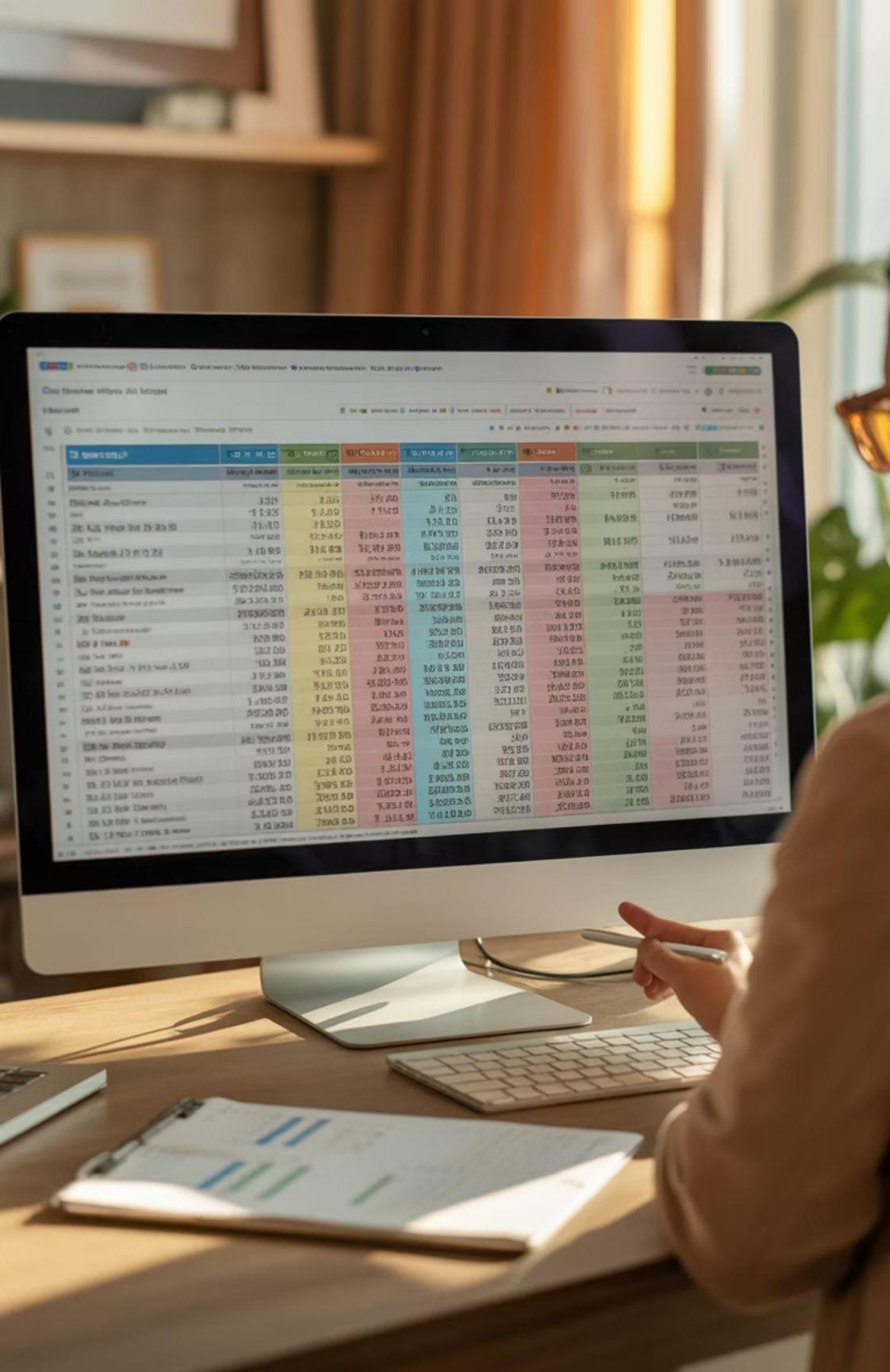
## Deudas Actuales

Monto total de obligaciones financieras existentes que impactan la capacidad de endeudamiento

4

## Plazo Solicitado

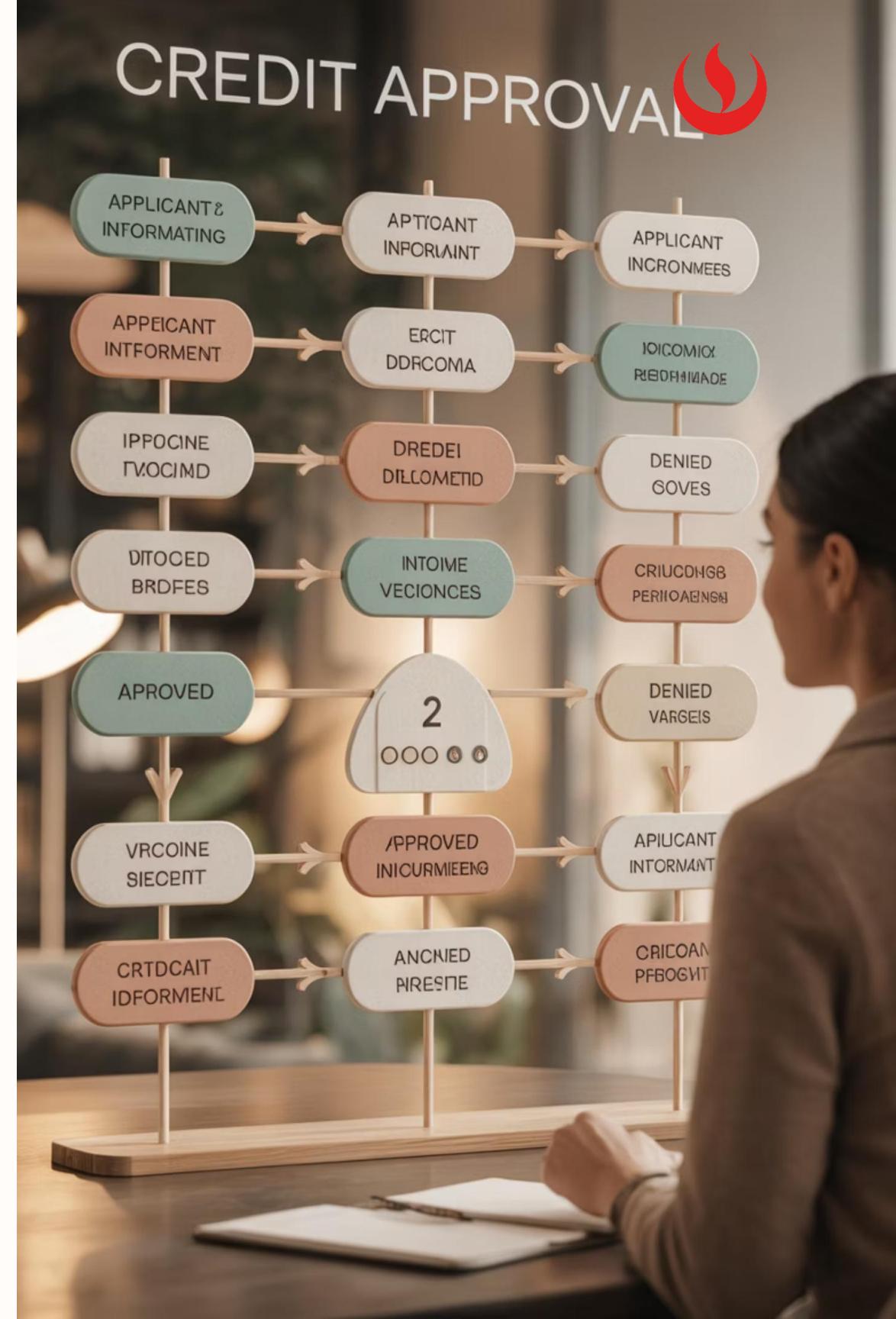
Duración del préstamo en meses: a mayor plazo, mayor riesgo percibido



## Caso 1: Árbol de Decisión Resultante

El árbol identifica que el **Historial Crediticio** es el factor más determinante, seguido por el nivel de Ingresos para casos intermedios.

- 1 Raíz: Historial Crediticio**  
Primera división: si es "Malo" → Rechazo automático
- 2 Nivel 2: Ingresos > \$3,000**  
Para historial "Regular" o mejor, evaluar capacidad de pago
- 3 Nivel 3: Ratio Deuda/Ingreso**  
Si deudas > 40% de ingresos → Evaluación adicional
- 4 Hojas: Decisión Final**  
Aprobar (historial excelente + ingresos altos) o Rechazar





# Caso 1: Reglas de Decisión Extraídas

## → Regla 1: Rechazo Inmediato

**SI** Historial\_Crediticio = "Malo" **ENTONCES** Rechazar (sin importar otros factores)

## → Regla 2: Aprobación Directa

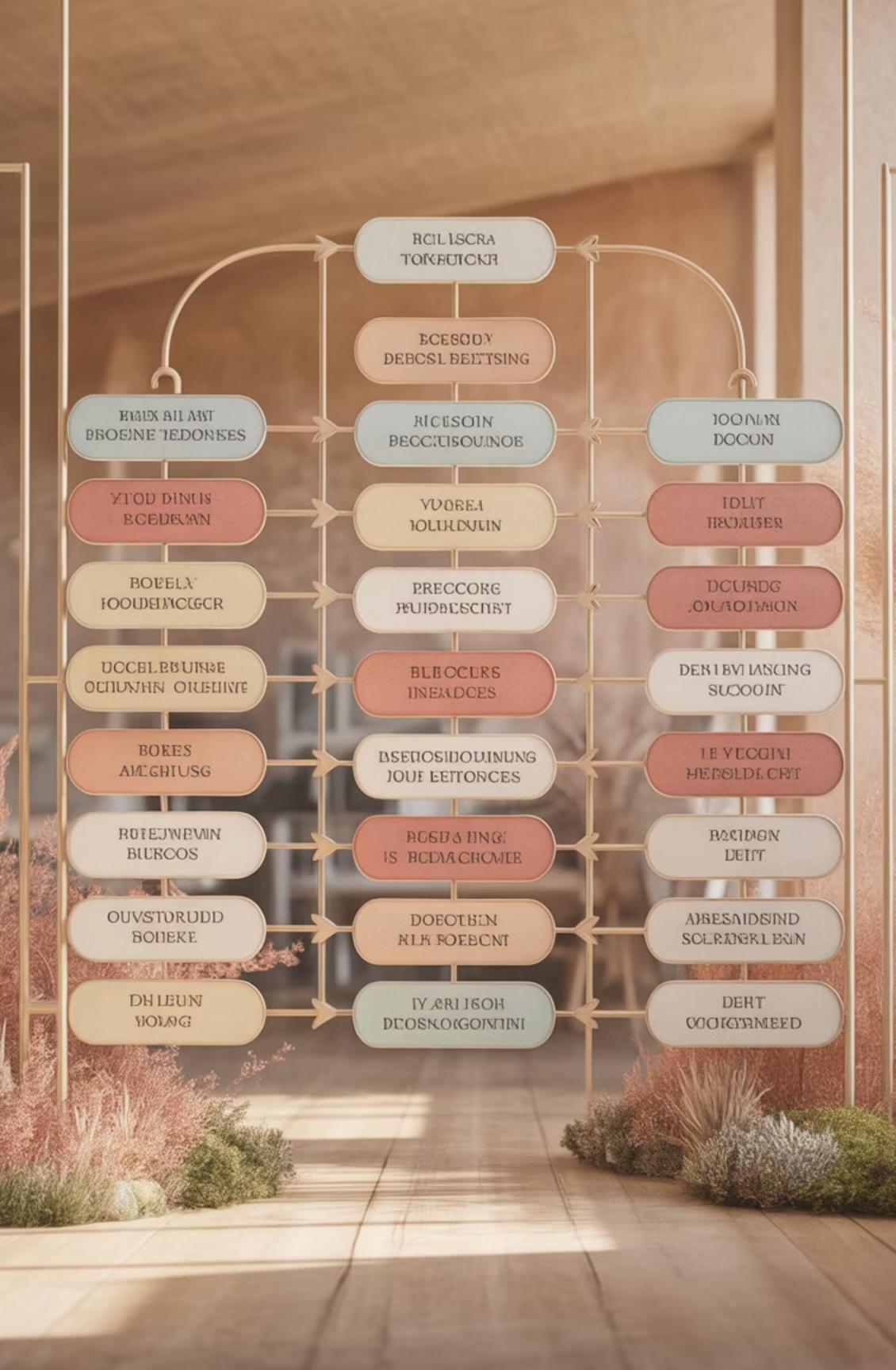
**SI** Historial = "Excelente" Y Ingresos > \$5,000 **ENTONCES** Aprobar

## → Regla 3: Evaluación Detallada

**SI** Historial = "Bueno" Y Ingresos > \$3,000 Y Deudas/Ingresos < 40% **ENTONCES** Aprobar

## → Regla 4: Rechazo por Sobreendeudamiento

**SI** Deudas/Ingresos > 60% **ENTONCES** Rechazar (alto riesgo de impago)





## Caso 2: Diagnóstico Médico Básico

Los árboles de decisión son herramientas valiosas en medicina para apoyar el diagnóstico temprano de enfermedades como la diabetes, combinando múltiples indicadores clínicos en reglas interpretables.

### Problema Clínico

Predecir si un paciente tiene alto riesgo de desarrollar diabetes tipo 2

### Clasificación Binaria

Riesgo Alto vs. Riesgo Bajo



# Caso 2: Datos Clínicos del Paciente



## Variables Numéricas

- **Nivel de Glucosa en Sangre:** mg/dL en ayunas (umbral crítico: 126 mg/dL)
- **Edad:** Años del paciente (riesgo aumenta con edad)
- **Presión Arterial:** Sistólica en mmHg
- **Índice de Masa Corporal (IMC):** kg/m<sup>2</sup> como indicador de obesidad

## Contexto Adicional

El modelo también puede incorporar:

- Historial familiar de diabetes
- Nivel de actividad física
- Perímetro abdominal
- Niveles de triglicéridos

# Caso 2: Árbol de Decisión Médico

El árbol identifica **Glucosa en Ayunas** como el predictor más importante, seguido por IMC y edad.



## Nodo Raíz: Glucosa

Si Glucosa  $\geq 126$  mg/dL  $\rightarrow$  Riesgo Alto inmediato (criterio diagnóstico estándar)



## Segunda División: IMC

Para glucosa 100-125 mg/dL: Si IMC  $> 30$  (obeso)  $\rightarrow$  Riesgo Alto



## Tercera División: Edad

Para IMC 25-30: Si Edad  $> 45$  años  $\rightarrow$  Riesgo Moderado-Alto





# Caso 2: Interpretación Clínica

## Factor Más Decisivo

La **glucosa en ayunas** es el biomarcador crítico, concordando con guías clínicas establecidas por la ADA (American Diabetes Association)

## Factores de Riesgo Secundarios

El **IMC elevado** y la **edad avanzada** actúan como amplificadores de riesgo en pacientes prediabéticos

## Aplicación Preventiva

Permite identificar tempranamente pacientes para programas de intervención lifestyle (dieta, ejercicio)

- ❑ **Nota Importante:** Este árbol es una herramienta de apoyo diagnóstico, no reemplaza el juicio clínico profesional ni pruebas confirmatorias.





## Caso 3: Clasificación de Especies de Iris

El conjunto de datos Iris es un clásico en machine learning. Contiene medidas de tres especies de flores, donde el objetivo es clasificar cada muestra en su especie correcta basándose en dimensiones de pétalos y sépalos.

### Especie 1: Setosa

Características distintivas y fácilmente separable

### Especie 3: Virginica

La más grande, similar a Versicolor en algunas medidas

### Especie 2: Versicolor

Características intermedias con cierta superposición



## Caso 3: Características Medidas

### Dimensiones del Pétalo

- **Largo del Pétalo:** Medido en cm, rango típico 1-7 cm
- **Ancho del Pétalo:** Medido en cm, rango típico 0.1-2.5 cm

### Dimensiones del Sépalo

- **Largo del Sépalo:** Medido en cm, rango típico 4-8 cm
- **Ancho del Sépalo:** Medido en cm, rango típico 2-4.5 cm

El conjunto contiene 150 muestras (50 de cada especie) con estas 4 características numéricas.



## Caso 3: Árbol Multinomial Resultante

El árbol logra una clasificación casi perfecta usando principalmente las medidas de **pétalo**.

### Primera División: Largo de Pétalo $\leq 2.45$ cm

Identifica perfectamente la especie **Setosa** (100% precisión)



### Segunda División: Ancho de Pétalo $\leq 1.75$ cm

Separa Versicolor de Virginica con alta precisión

### División Terciaria: Largo de Pétalo $\leq 4.95$ cm

Refina la clasificación en casos ambiguos entre las dos especies restantes



## Caso 3: Fronteras de Decisión

Visualizando solo dos dimensiones (Largo vs. Ancho de Pétalo), podemos ver cómo el árbol divide el espacio en regiones rectangulares:

### Región Setosa

Cluster bien separado en la esquina inferior izquierda (pétales pequeños)

### Región Versicolor

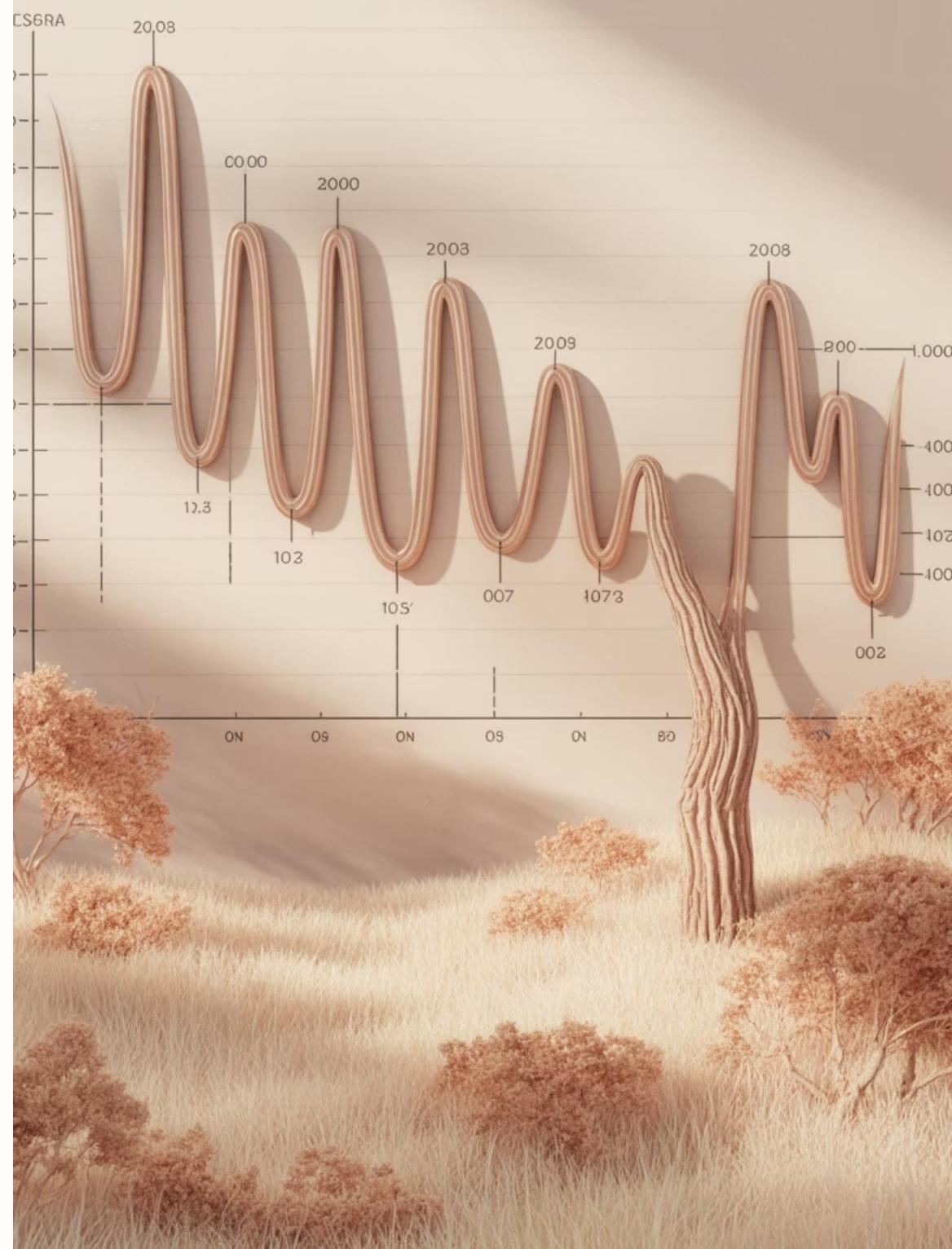
Zona central con ligera superposición hacia Virginica

### Región Virginica

Esquina superior derecha (pétales grandes y anchos)

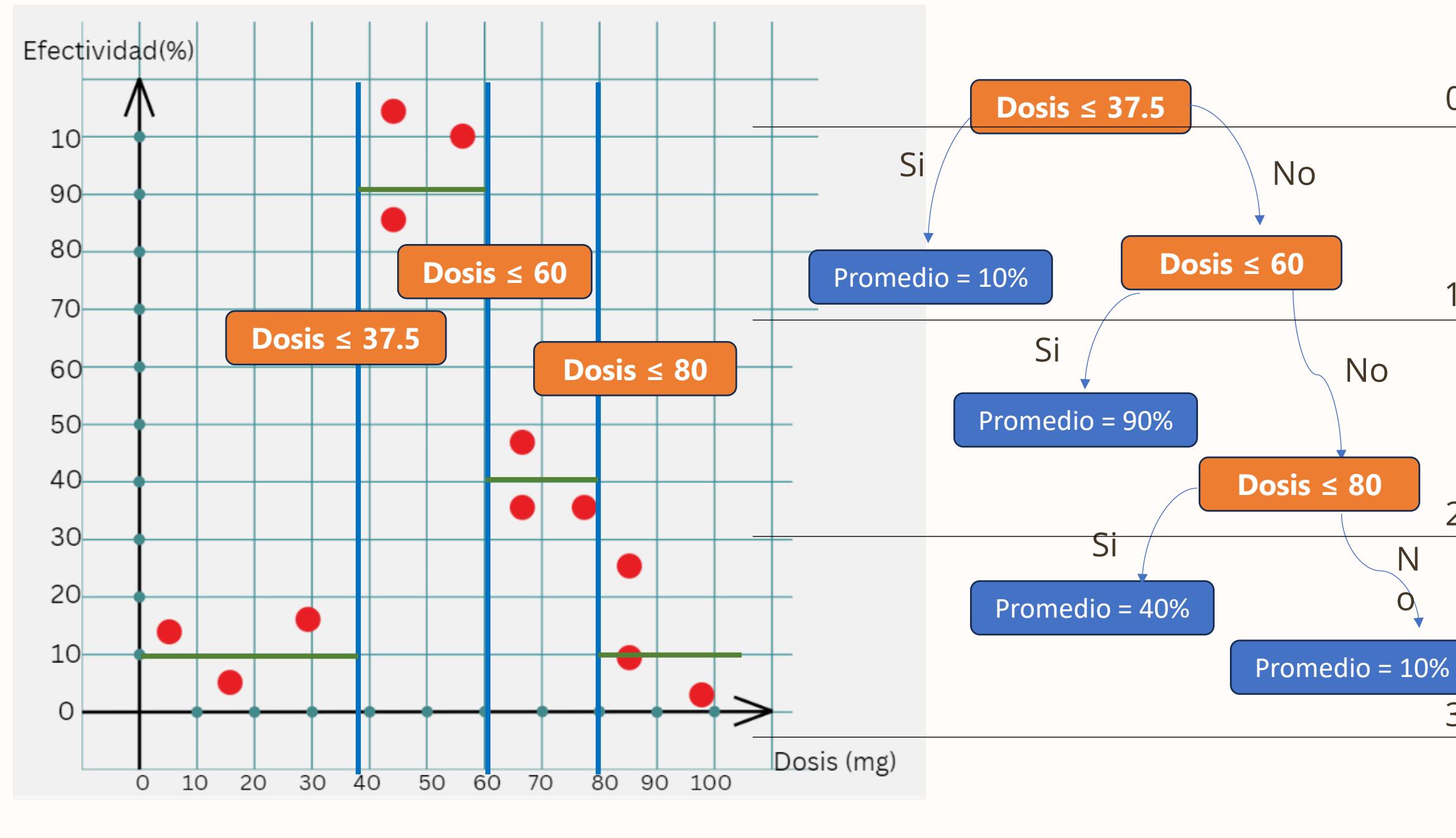


# Árboles de decisión - Regresión



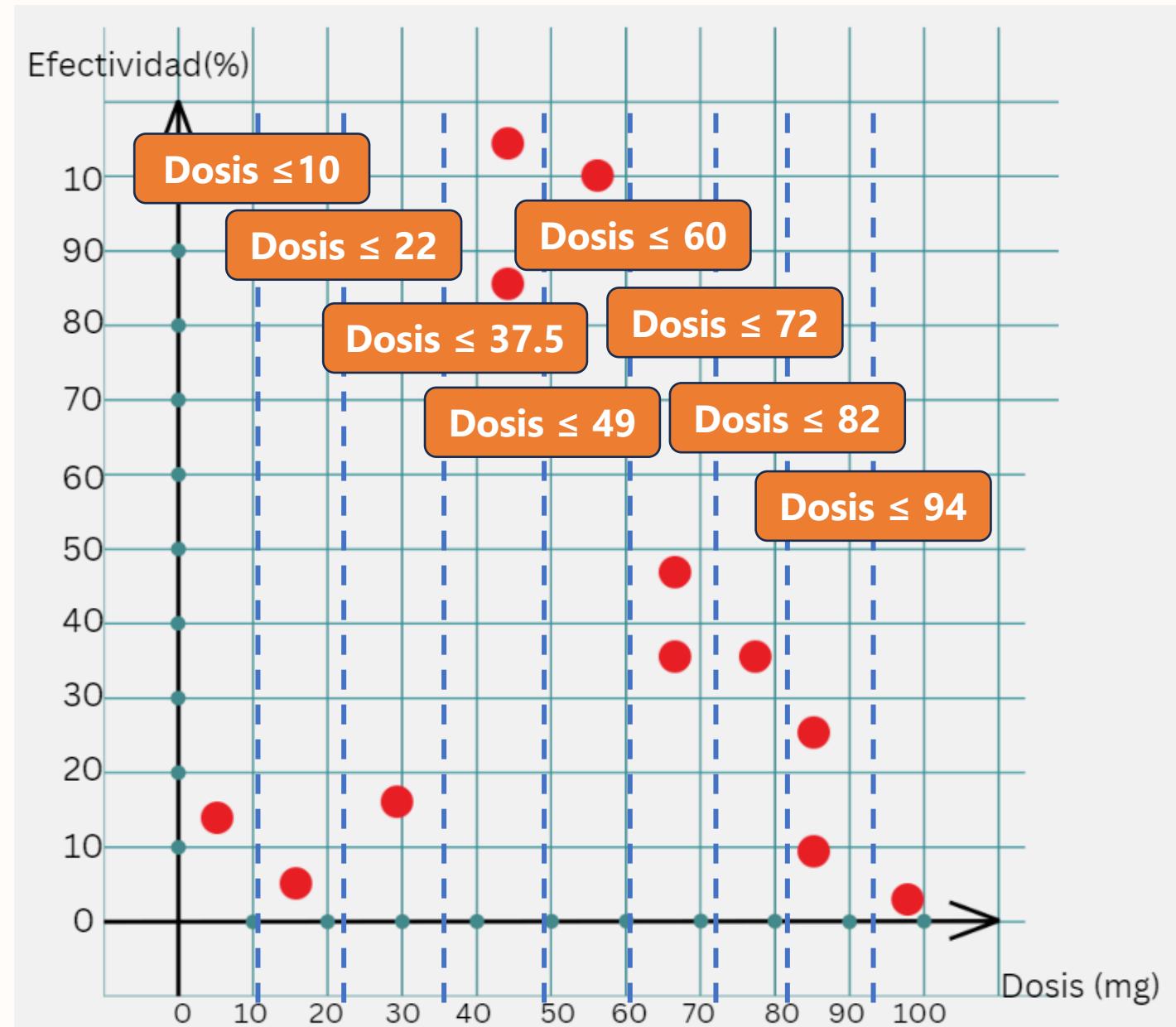


# Árboles de decisión - Regresión – Definición





# Árboles de decisión - Regresión – Definición



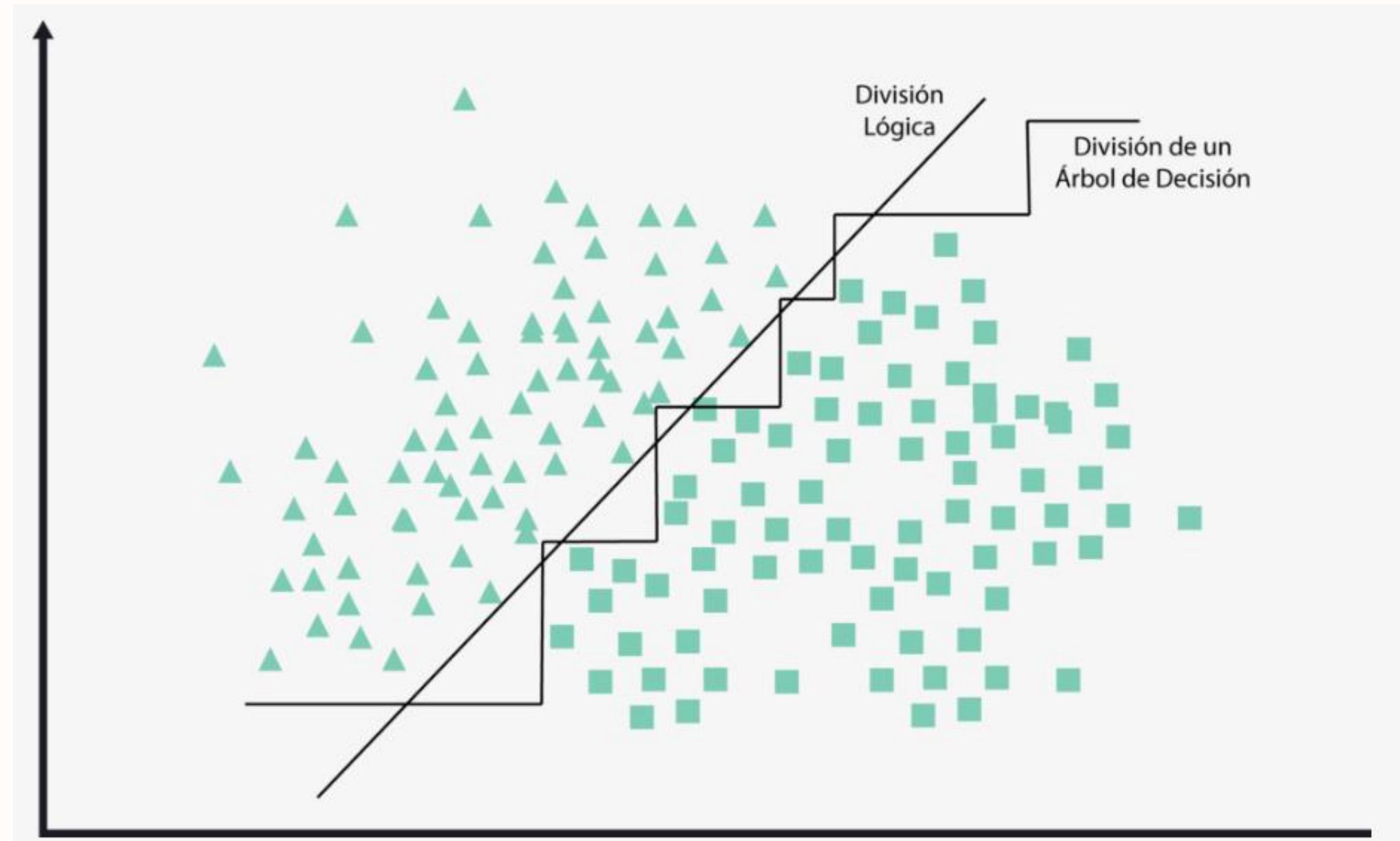
Para hallar el nodo principal, se utiliza el error cuadrático medio y la función de costo.

|              | Costo  |
|--------------|--------|
| Dosis ≤ 10   | 1033.4 |
| Dosis ≤ 22   | 922    |
| Dosis ≤ 37.5 | 850    |
| Dosis ≤ 49   | 1052   |
| Dosis ≤ 60   | 1010   |
| Dosis ≤ 72   | 1050   |
| Dosis ≤ 82   | 1023   |
| Dosis ≤ 94   | 1015   |

Para los siguientes nodos se realiza el mismo procedimiento a partir del inicial y los que son seleccionados con el costo menor, hasta cumplir con el criterio de parada.



# Árboles de decisión - Regresión – Definición





# Árboles de decisión - Regresión – Definición

## Independencia de las Observaciones

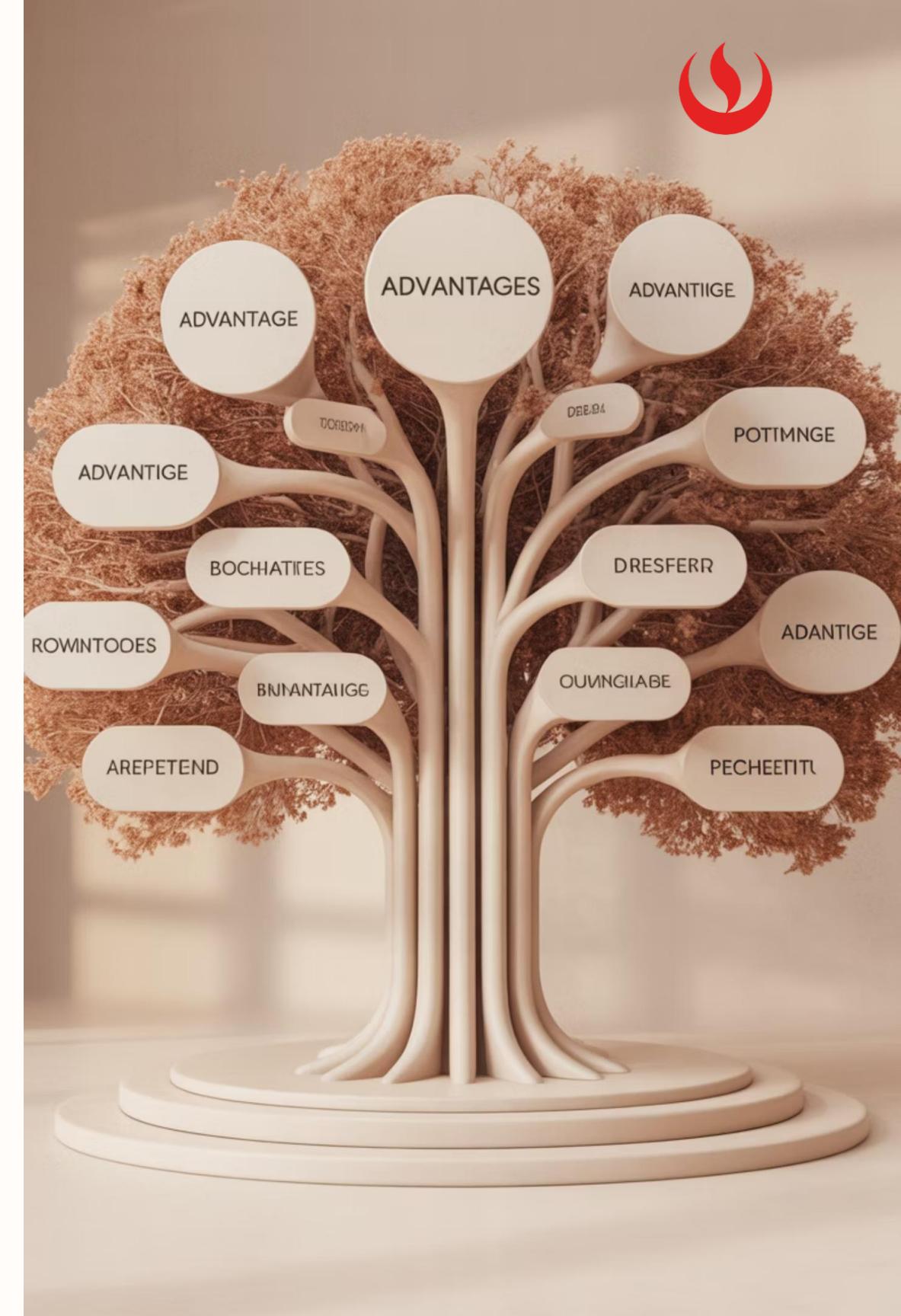
Se asume que las observaciones en el conjunto de datos son independientes entre sí.

## Relaciones No Lineales

Los árboles de decisión no requieren relaciones lineales entre las variables predictoras (características) y la variable objetivo

## Importancia de Características Predictivas

Se asume que hay ciertas características en los datos que tienen relevancia o poder predictivo.





# Caso de Estudio



# Caso 4: Predicción de Precios de Vivienda

Los árboles de regresión son ideales para estimar valores de propiedades, segmentando el mercado en categorías homogéneas basadas en características clave que determinan el precio.

## Problema de Regresión

Estimar el valor de mercado de una propiedad en unidades monetarias

## Variable Objetivo

Precio de venta en miles de dólares (valor continuo)



# Caso 4: Variables del Mercado Inmobiliario

## Características Físicas

- **Metros Cuadrados (m<sup>2</sup>):** Área construida total
- **Número de Habitaciones:** Dormitorios principales
- **Número de Baños:** Completos y medios baños
- **Antigüedad:** Años desde construcción

## Características de Ubicación

- **Zona/Barrio:** Prestigio y demanda del área
- **Proximidad a Servicios:** Escuelas, transporte, comercios
- **Estrato Socioeconómico:** Categoría de la zona

REAL ESTATE  
Projestioy

|   |                                   |                               |                                |
|---|-----------------------------------|-------------------------------|--------------------------------|
| 2700                                      | 502                               | 55                            | 90                             |
| SQUARE                                    | BEATROURE                         | BATHROOMS                     | BASHROWATIE                    |
| SQUARE FOOTARE<br>DDOGEGO                 | NUMEROUS AL<br>OMBOEAGUEA         | MTORAKSI<br>APBOOICOBRI       | WATERNOULL<br>EOEADKONOSNIOGSI |
| NE I HERO/OPTAUE<br>DE IDOT<br>NEOTRISMAN | NUMBER OF HEIGRI<br>ABEO BTERSEO. | THAI CAMIT<br>CIVEERB OUSIKOB | NUJWM FIBAM<br>MOSHAT          |

# Caso 4: Árbol de Regresión

El árbol segmenta el mercado jerárquicamente, generando **promedios de precio** en cada hoja.

## Primera División: $m^2 \geq 100$

Separa propiedades grandes (precio promedio: \$180K) de pequeñas (\$95K)

## Segunda División: Ubicación Premium

Para propiedades grandes, zona premium aumenta precio a \$250K vs. \$140K

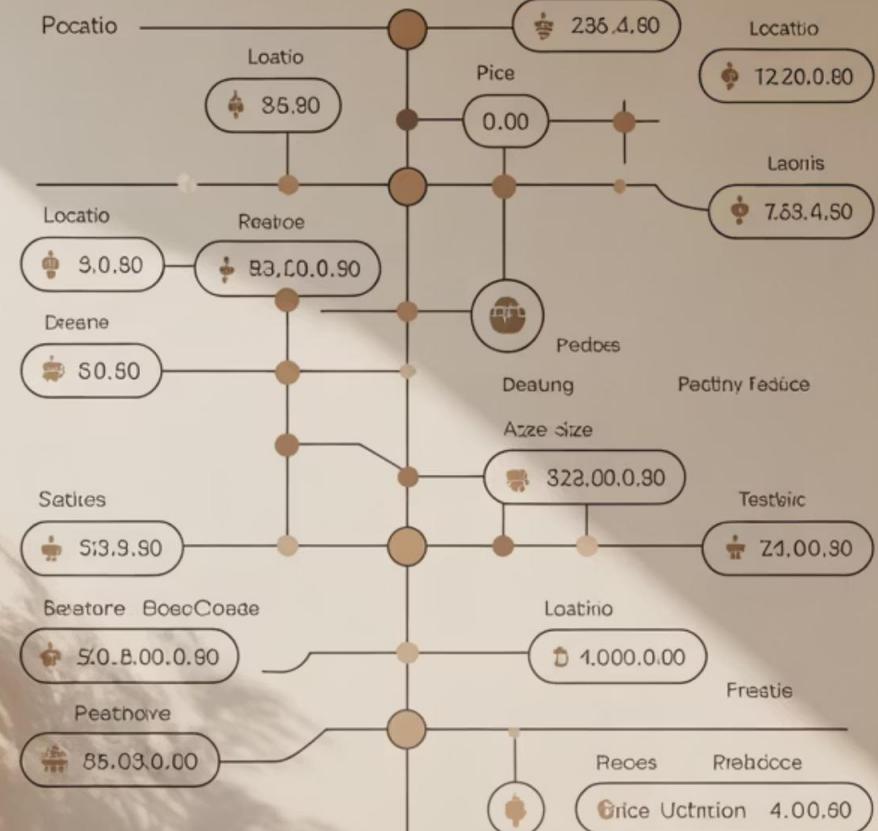
## Tercera División: Antigüedad < 10 años

En zona premium, propiedades nuevas alcanzan \$320K vs. \$200K para antiguas

## División Final: N° Habitaciones $\geq 4$

Refinamiento final: casas familiares grandes llegan hasta \$380K

Progression Tree Prices





# Caso 4: Segmentos de Mercado Identificados



**\$85K**

**Segmento Básico**  
Propiedades < 80m<sup>2</sup>, zona  
periférica, > 20 años  
antigüedad

**\$145K**

**Segmento Medio**  
80-120m<sup>2</sup>, zona  
intermedia, 10-20 años, 2-3  
habitaciones

**\$220K**

**Segmento Premium**  
120-180m<sup>2</sup>, zona exclusiva,  
< 10 años, 3-4 habitaciones

**\$380K**

**Segmento Lujo**  
> 180m<sup>2</sup>, zona premium,  
nueva construcción, ≥ 4  
habitaciones

Estos segmentos permiten a agentes inmobiliarios posicionar propiedades y a compradores tomar decisiones informadas.



# Ventajas de los Árboles de Decisión



## Interpretabilidad Transparente

Funcionan como "cajas blancas" donde cada decisión es clara y trazable, facilitando la explicación a stakeholders no técnicos



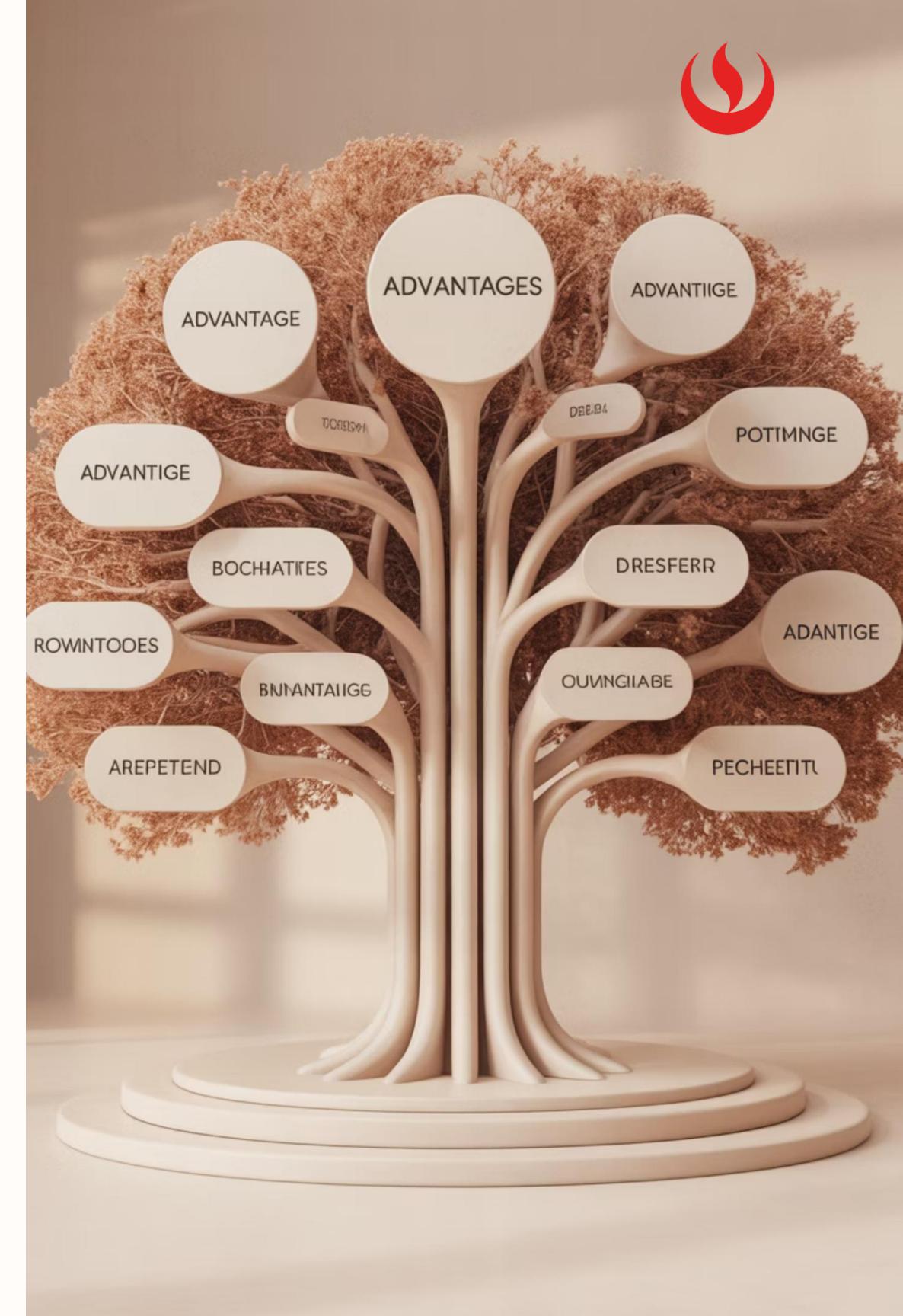
## Simplicidad de Uso

Requieren poca preparación de datos, no necesitan normalización y funcionan bien con pocas muestras de entrenamiento



## Versatilidad de Datos

Maneján naturalmente tanto características numéricas continuas como variables categóricas sin transformaciones adicionales





# Desventajas y Limitaciones

## Propensión al Sobreajuste

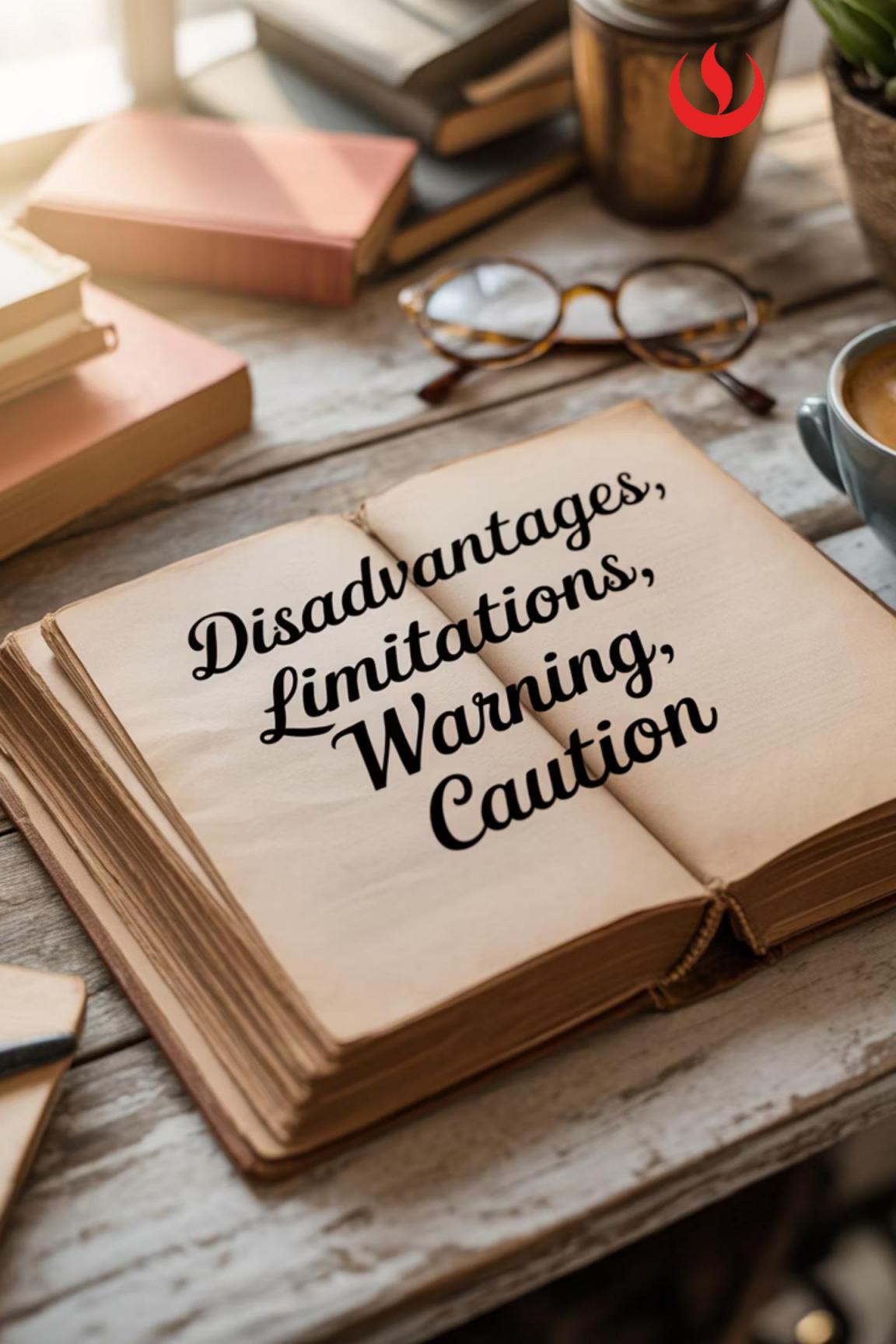
Los árboles pueden crecer demasiado profundos, memorizando el ruido del conjunto de entrenamiento en lugar de aprender patrones generalizables

## Alta Inestabilidad

Pequeños cambios en los datos de entrenamiento pueden resultar en árboles completamente diferentes, afectando la reproducibilidad

## Sesgos en Divisiones

Favorecen características con muchos valores únicos y pueden tener dificultades con relaciones lineales complejas





# Python IA





## TEMARIO

- 
- The background image shows the exterior of a modern, multi-story building at night. The building has large glass windows and a dark facade. In the foreground, there are palm trees and a paved walkway. A small sign on the building reads "Intertank".
- 1 — Regresión logística
  - 2 — Regresión y clasificación con Árboles de Decisión
  - 3 — Regresión y clasificación con Árboles de Decisión





# Regresión y Clasificación con Random Forest

Un viaje completo desde los fundamentos hasta aplicaciones avanzadas en el mundo real





# El Problema: ¿Por qué no usar un solo Árbol de Decisión?

## Sobreajuste Crítico

Los Árboles de Decisión individuales tienden a memorizar los datos de entrenamiento en lugar de aprender patrones generalizables. Esto resulta en un rendimiento excelente durante el entrenamiento pero pobre con datos nuevos.

- Alta varianza en las predicciones
- Sensibilidad extrema a pequeños cambios
- Precisión engañosa en entrenamiento

## Inestabilidad Estructural

Un pequeño cambio en los datos puede producir un árbol completamente diferente. Esta inestabilidad hace que los modelos sean poco confiables para aplicaciones críticas donde la consistencia es fundamental.

- Estructura variable con datos similares
- Dificulta la reproducibilidad
- Baja confiabilidad en producción



# La Solución: Métodos de Ensamble

## La Sabiduría de la Multitud

En lugar de confiar en un solo modelo, los métodos de ensamble combinan las predicciones de múltiples modelos para obtener resultados más robustos y precisos.

## Reducción de Error

Al promediar o votar entre múltiples modelos, los errores individuales tienden a cancelarse, resultando en predicciones más estables y confiables.

## Mayor Generalización

Los ensambles capturan diferentes aspectos de los datos, creando un modelo más completo que generaliza mejor a situaciones nuevas.



# Concepto de Ensamble: Bagging vs. Boosting

Bagging (Bootstrap Aggregating)



Boosting





# Definición de Random Forest

Random Forest es un algoritmo de ensamble que construye múltiples Árboles de Decisión mediante la técnica de Bagging, introduciendo aleatoriedad adicional en la selección de características durante la construcción de cada árbol.

## Ensamble de Árboles

Combina cientos o miles de árboles de decisión entrenados de forma independiente

## Doble Aleatoriedad

Introduce variación tanto en los datos (bootstrap) como en las características disponibles

## Predictión Colectiva

Agrega las predicciones mediante votación (clasificación) o promedio (regresión)



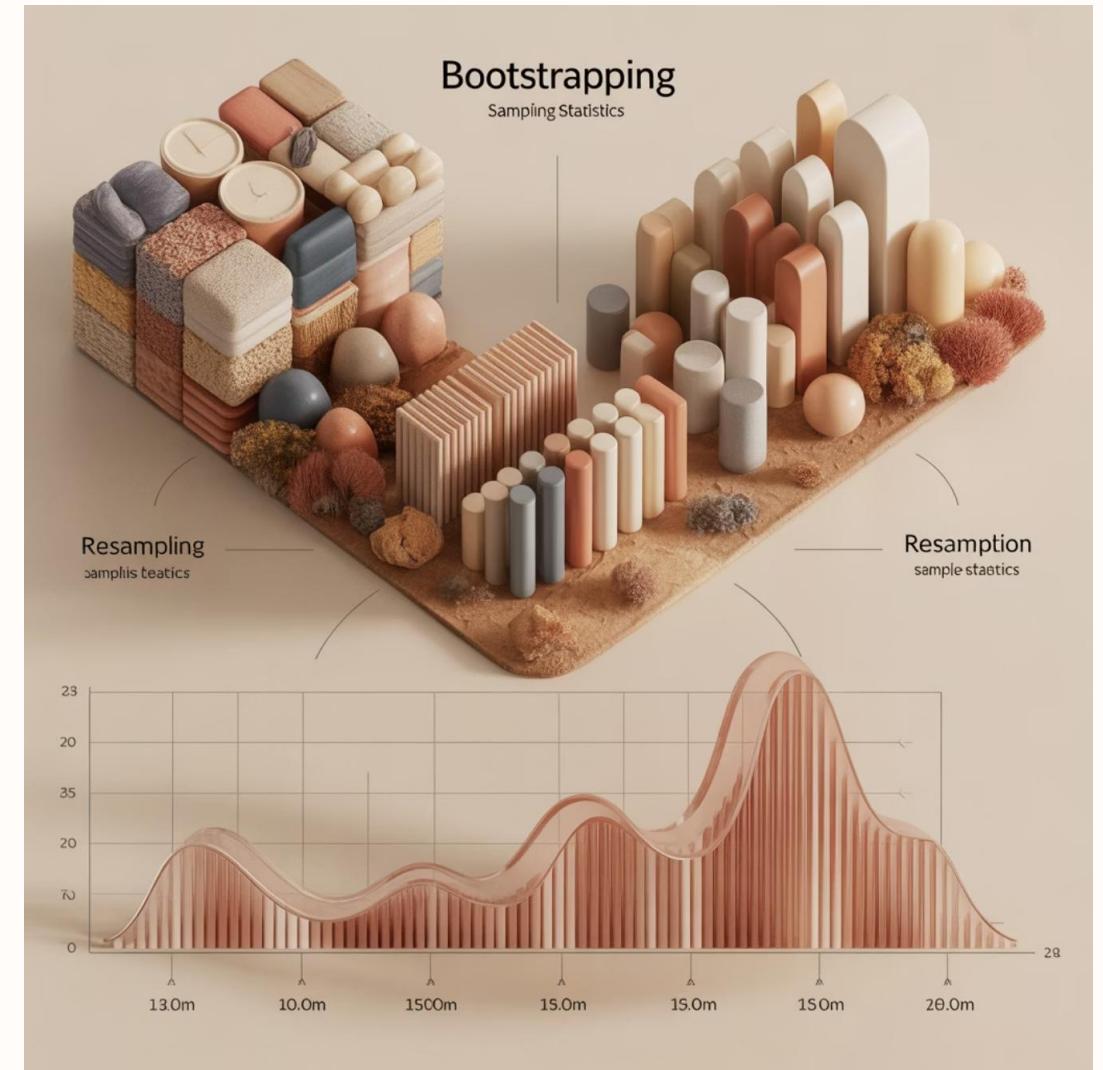


# Paso 1: Bootstrap (Muestreo con Reemplazo)

El bootstrap es la técnica fundamental que permite crear diversos conjuntos de entrenamiento a partir de un único dataset original. Para cada árbol en el bosque, se genera un nuevo conjunto de datos seleccionando muestras **con reemplazo** del dataset original.

## ¿Cómo funciona?

1. Se toma el dataset original de N observaciones
2. Se seleccionan N muestras aleatoriamente CON reemplazo
3. Algunas observaciones aparecen múltiples veces
4. Aproximadamente 63% de datos únicos por muestra
5. El 37% restante se usa para validación (OOB)



**Nota clave:** El muestreo con reemplazo significa que  después de seleccionar una observación, esta vuelve al conjunto y puede ser seleccionada nuevamente.



## Paso 2: Aleatoriedad de Características

Además del bootstrap en las observaciones, Random Forest introduce una segunda capa de aleatoriedad: en cada división del árbol, solo se considera un **subconjunto aleatorio** de  $m$  características en lugar de todas las características disponibles.



### Selección Aleatoria

En cada nodo del árbol, se eligen aleatoriamente  $m$  características de las  $p$  totales disponibles



### Decorrelación

Esta técnica asegura que los árboles sean diferentes entre sí, evitando que todos usen las mismas características dominantes



### Valor Típico

Clasificación:  $m = \sqrt{p}$  características. Regresión:  $m = p/3$  características



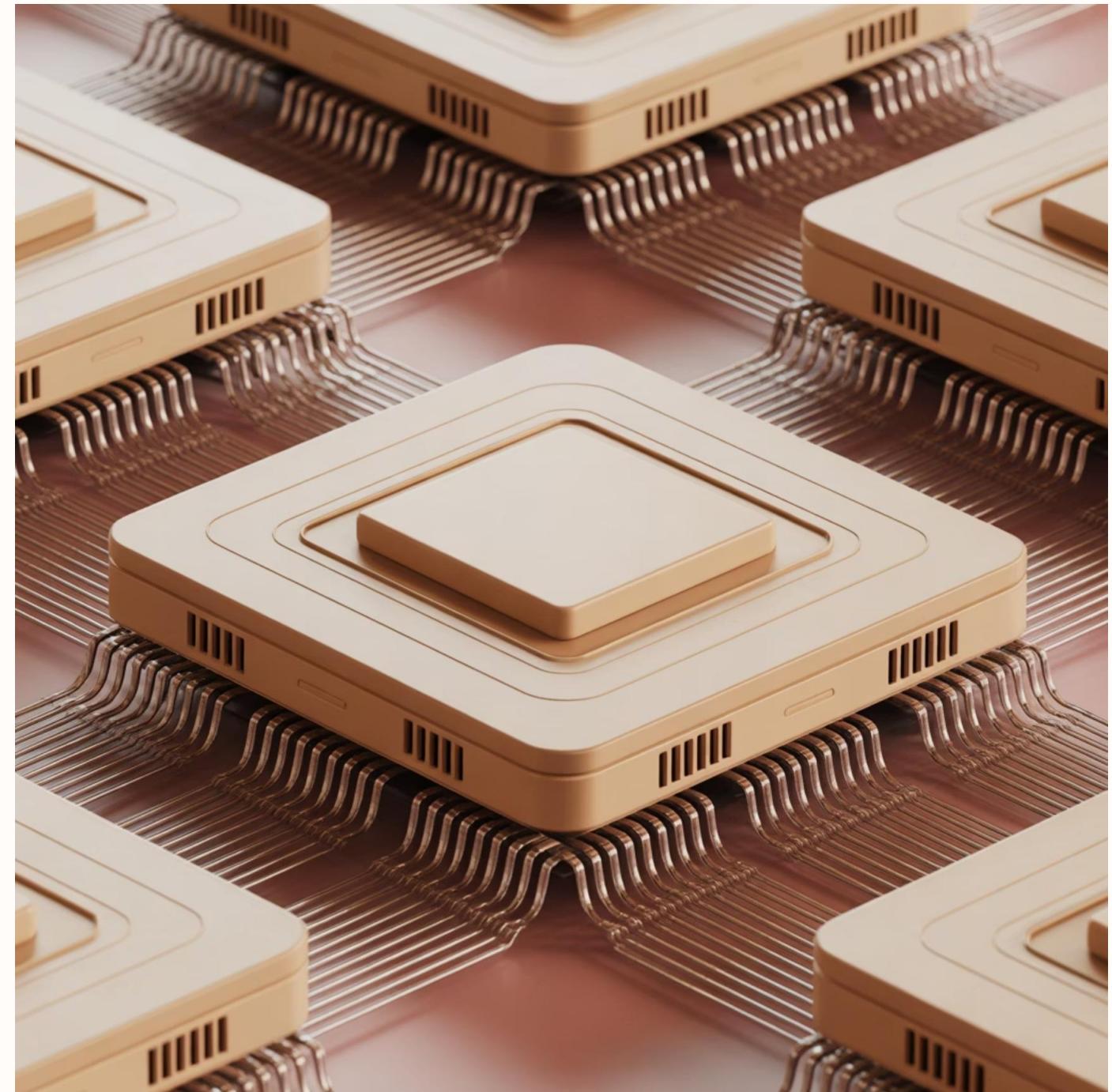
# Paso 3: Construcción de Múltiples Árboles

## Cada Árbol es Único

Debido a la combinación del bootstrap y la selección aleatoria de características, cada árbol en el bosque ve una perspectiva diferente de los datos. Esta diversidad es la clave del poder predictivo de Random Forest.

- Árbol 1: Datos A, Características {X1, X3, X7}
- Árbol 2: Datos B, Características {X2, X5, X7}
- Árbol 3: Datos C, Características {X1, X4, X6}
- ...y así sucesivamente para cientos de árboles

## Entrenamiento Paralelo





# Paso 4: Agregación en Clasificación

## Votación Mayoritaria

Para problemas de clasificación, cada árbol en el bosque emite un "voto" por la clase que predice. La predicción final del Random Forest es la clase que recibe más votos.

01

### Predictión Individual

Cada uno de los n árboles predice una clase para la nueva observación

02

### Conteo de Votos

Se cuentan cuántos árboles votaron por cada clase posible

03

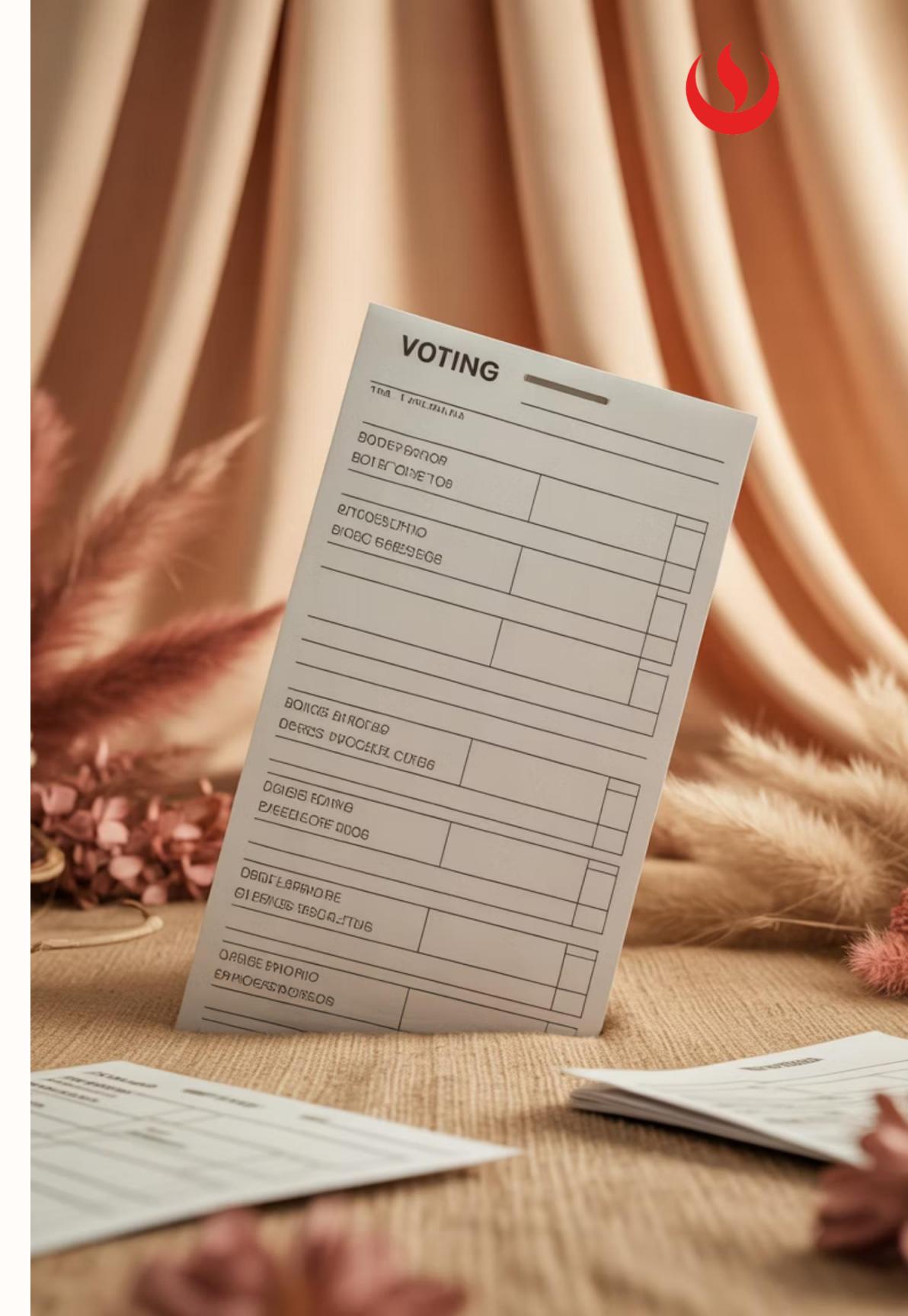
### Clase Ganadora

La clase con mayor número de votos se selecciona como predicción final

04

### Probabilidades

Opcionalmente, el porcentaje de votos puede interpretarse como probabilidad de clase





# Paso 4: Agregación en Regresión

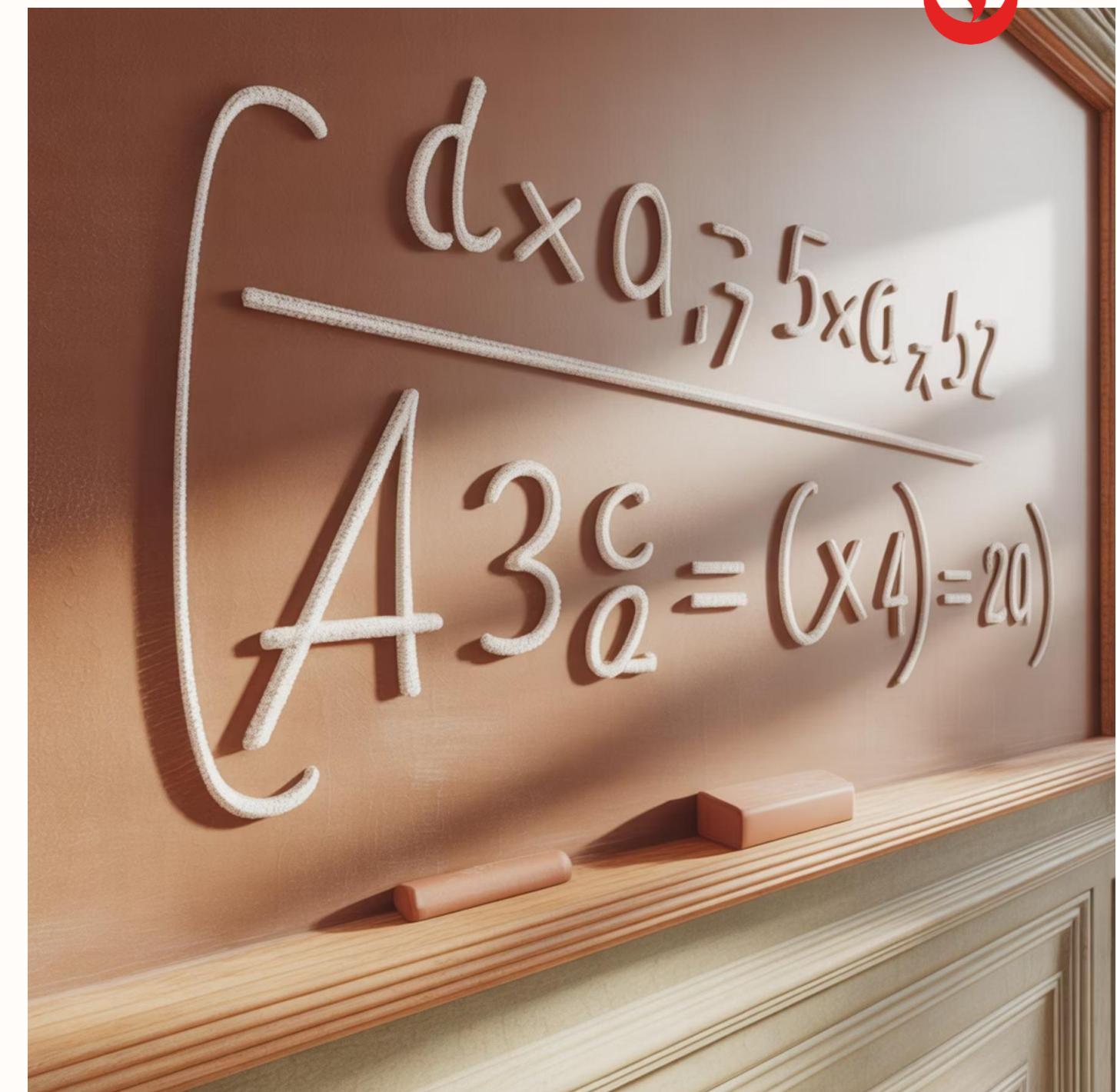
## Promedio de Predicciones

Para problemas de regresión, cada árbol predice un valor numérico continuo. La predicción final del Random Forest es simplemente el promedio de todas las predicciones individuales de los árboles.

### Ejemplo Numérico

- Árbol 1 predice: 45.3
- Árbol 2 predice: 47.8
- Árbol 3 predice: 46.1
- Árbol 4 predice: 48.2
- Árbol 5 predice: 46.9

**Predicción final:**  $(45.3 + 47.8 + 46.1 + 48.2 + 46.9) / 5 = 46.86$

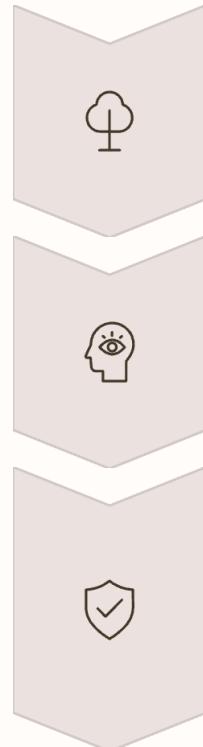


El promedio suaviza las predicciones extremas de árboles individuales, resultando en estimaciones más estables y confiables.



# Ventaja 1: Reducción Drástica del Sobreajuste

El poder de Random Forest radica en su capacidad para controlar el sobreajuste, uno de los problemas más críticos de los Árboles de Decisión individuales.



## Árbol Individual

Puede memorizar ruido y patrones espurios del conjunto de entrenamiento

## Promedio/Votación

Los errores aleatorios de árboles individuales se cancelan entre sí

## Modelo Robusto

Solo persisten los patrones verdaderos presentes en la mayoría de los árboles



"Mientras más árboles en el bosque, más robusto es el modelo frente al sobreajuste. La diversidad es la clave."



# Ventaja 2: Alta Precisión

## Superioridad Consistente

En la mayoría de problemas prácticos, Random Forest supera a los Árboles de Decisión individuales y a muchos otros algoritmos clásicos de machine learning. Esta ventaja se debe a su capacidad de capturar relaciones complejas sin sobreajustar.

## Comparación típica de precisión:

- Árbol de Decisión: 75-80%
- Random Forest: 85-92%
- Ganancia: +10-12 puntos porcentuales



- ❑ **En competencias de Kaggle:** Random Forest frecuentemente aparece en soluciones ganadoras, ya sea como modelo principal o como parte de un ensamble más grande.



# Ventaja 3: Importancia de Características

## ¿Cómo funciona?

Random Forest proporciona una medida natural de la importancia de cada característica en el modelo. Esto se calcula observando cuánto mejora cada característica la pureza de las divisiones a lo largo de todos los árboles del bosque.

01

### Medición en Cada Árbol

Para cada árbol, se registra la reducción de impureza (Gini o Entropía) que produce cada característica

02

### Promedio Global

Se promedian las contribuciones de cada característica a través de todos los árboles del bosque

03

### Normalización

Los valores se normalizan para que sumen 100%, creando un ranking interpretable





# Ventaja 4: Robustez



## Datos Faltantes

Random Forest puede manejar valores faltantes de manera elegante, usando técnicas como imputación interna o ignorando características ausentes en divisiones específicas. No requiere imputación previa obligatoria.



## Sin Escalado

A diferencia de algoritmos como SVM o regresión logística, Random Forest no requiere normalización o estandarización de características. Las decisiones se basan en umbrales, no en distancias.



## Resistente a Outliers

Los árboles individuales pueden acomodar valores extremos sin que estos dominen el modelo completo, ya que su efecto se diluye en el promedio del ensamble.

## Datos Heterogéneos

Maneja naturalmente mezclas de características numéricas y categóricas sin necesidad de codificación one-hot extensiva, simplificando el preprocesamiento.



# Desventaja 1: Costo Computacional

## Recursos Intensivos

Entrenar cientos o miles de árboles completos requiere significativamente más recursos computacionales que un modelo simple. El costo se multiplica por el número de árboles en el bosque.



## Consideraciones Prácticas

- **Tiempo de entrenamiento:** 10-100x más lento que un árbol individual
- **Memoria:** Debe almacenar todos los árboles en RAM
- **Predicción:** Más lenta, debe consultar todos los árboles
- **Mitigación:** Entrenamiento paralelo, poda de árboles

- En datasets grandes (millones de filas), considerar alternativas como Gradient Boosting o sampleo estratégico.



# Desventaja 2: Pérdida de Interpretabilidad

Mientras un Árbol de Decisión único puede ser visualizado y comprendido completamente, un Random Forest con cientos de árboles se convierte en un modelo de "caja negra". Es prácticamente imposible entender el proceso de decisión completo.

## Regulaciones y Auditoría

En sectores altamente regulados (banca, salud, seguros), la falta de explicabilidad puede ser un obstáculo legal o ético significativo.

## Confianza del Usuario

Los stakeholders pueden desconfiar de predicciones que no pueden ser explicadas en términos simples paso a paso.

## Depuración Difícil

Cuando el modelo falla, es muy complicado identificar qué salió mal o cómo corregir el comportamiento específico.





# Casos de Estudio



# Caso 1: Detección de Fraude

## El Problema

Las instituciones financieras procesan millones de transacciones diariamente, y solo una fracción minúscula (0.1-0.5%) son fraudulentas. Detectar fraude requiere **alta precisión** para minimizar falsos positivos (transacciones legítimas bloqueadas) mientras se capturan verdaderos fraudes.

## Desafíos Específicos:

- Desbalance extremo de clases (99.5% legítimas vs. 0.5% fraude)
- Patrones de fraude en constante evolución
- Costo asimétrico: fraude no detectado es muy costoso
- Datos con ruido y comportamientos atípicos legítimos
- Necesidad de respuesta en tiempo real



- **Contexto:** Un sistema de detección debe procesar miles de transacciones por segundo con latencia < 100ms.



# Caso 1: ¿Por qué Random Forest?



## Manejo del Desbalance

RF puede usar técnicas como `class_weight='balanced'` para dar mayor peso a la clase minoritaria (fraude), mejorando la sensibilidad sin necesitar SMOTE complejo.



## Robusto al Ruido

El promediado de múltiples árboles filtra transacciones anómalas legítimas que podrían confundir a modelos más simples.



## Predicción Rápida

Aunque el entrenamiento es costoso, la predicción es eficiente y puede optimizarse para cumplir requisitos de latencia en producción.



## Probabilidades

RF proporciona probabilidades de fraude, permitiendo ajustar el umbral de decisión según el trade-off deseado entre precisión y recall.

"En detección de fraude, no buscamos perfección, sino el balance óptimo entre detectar fraudes reales y no molestar a clientes legítimos."



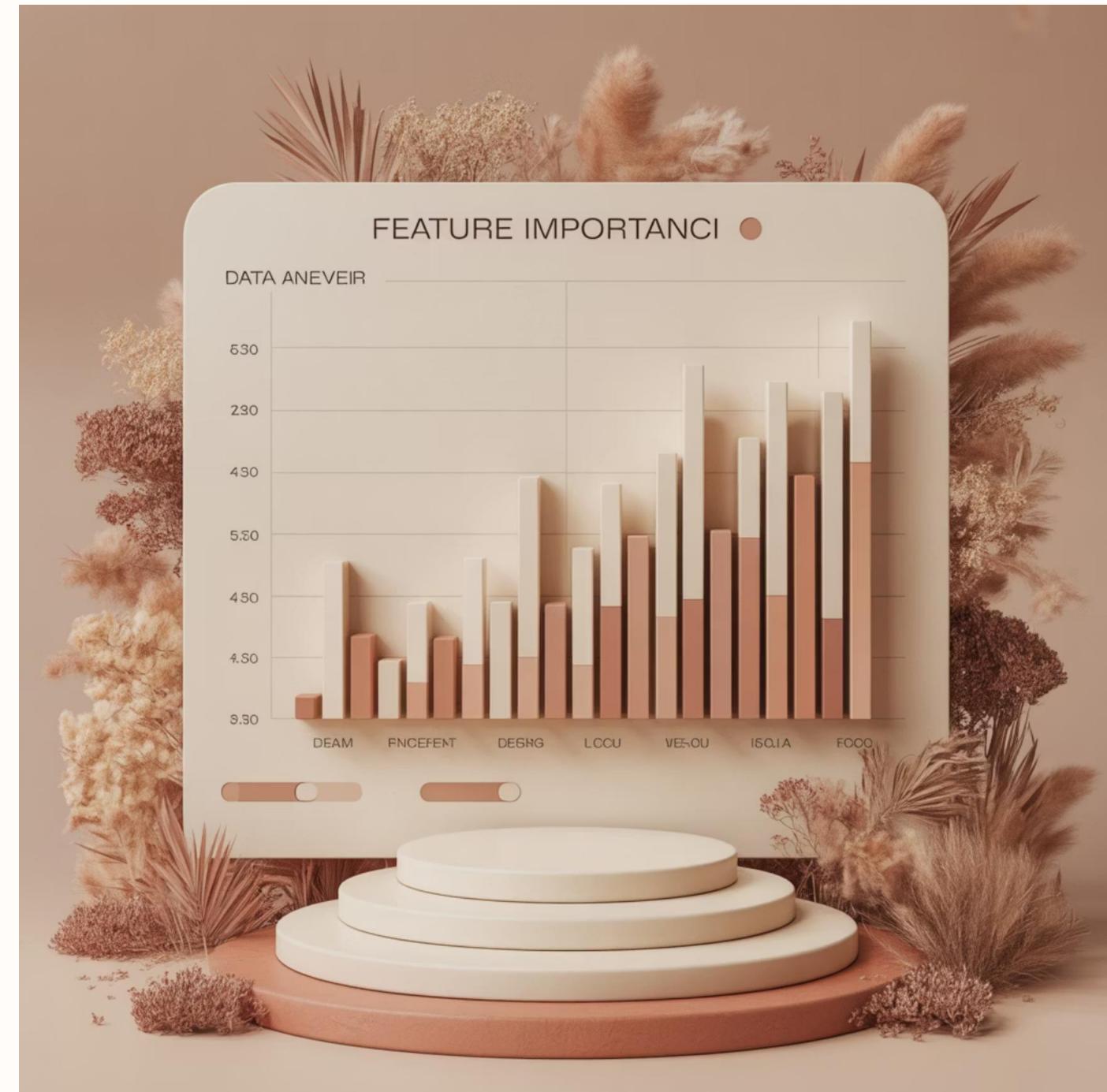
# Caso 1: Importancia de Características

## Variables que Alertan Sobre Fraude

El análisis de importancia de características revela qué aspectos de una transacción son más predictivos de fraude, proporcionando insights valiosos para el negocio.

### Top 10 Características Predictivas:

1. **Monto de transacción** (especialmente montos inusuales)
2. **Ubicación geográfica** (distancia del patrón habitual)
3. **Hora del día** (transacciones nocturnas son más riesgosas)
4. **Tipo de comercio** (categorías de alto riesgo)
5. **Velocidad de transacciones** (múltiples en corto tiempo)
6. **Cambio de patrón** (desviación del comportamiento histórico)
7. **Tipo de tarjeta** (débito vs. crédito)
8. **Método de entrada** (chip, contactless, manual)
9. **País del comercio** (transacciones internacionales)
10. **Historial reciente** (transacciones previas rechazadas)



Insight clave: Las características temporales y de patrones de comportamiento son más predictivas que características individuales de la transacción aislada.



# Caso 1: Resultados y Métricas



## Evaluación con Matriz de Confusión

En problemas desbalanceados como detección de fraude, la precisión (accuracy) es engañosa. Un modelo que predice "no fraude" siempre tendría 99.5% de precisión pero sería inútil. Necesitamos métricas más sofisticadas.

## Trade-off Crítico:

**Falsos Positivos (FP):** Bloquear transacciones legítimas frustra clientes y genera pérdida de ventas.

**Falsos Negativos (FN):** Fraudes no detectados generan pérdidas directas y daño reputacional.

- El umbral de decisión se ajusta según el costo de negocio: típicamente se prefiere más FP que FN.

## Métricas Clave:

- **Recall (Sensibilidad):** 87% de fraudes detectados
- **Precision:** 76% de alertas son fraudes reales
- **F1-Score:** 0.81 (balance recall-precision)
- **AUC-ROC:** 0.94 (excelente discriminación)



# Caso 2: Diagnóstico Médico Avanzado

## El Problema: Datos Genómicos

El diagnóstico de enfermedades como el cáncer usando datos genómicos presenta desafíos únicos. Los microarrays genéticos miden la expresión de **miles o decenas de miles de genes** simultáneamente ( $p \gg n$ : más características que observaciones).

### Alta Dimensionalidad

20,000-30,000 genes medidos vs. solo 100-500 pacientes en estudios típicos. Riesgo extremo de sobreajuste.

### Multicolinealidad

Genes relacionados funcionalmente muestran patrones de expresión correlacionados, complicando la interpretación.

### Datos Costosos

Obtener muestras y realizar secuenciación es extremadamente caro, limitando el tamaño del dataset.

### Heterogeneidad

El cáncer del mismo tipo puede tener perfiles genéticos muy diferentes entre pacientes.



# Caso 2: ¿Por qué Random Forest?

## Excelente para Alta Dimensionalidad

Random Forest maneja naturalmente problemas donde  $p \gg n$  gracias a su mecanismo de selección aleatoria de características. El parámetro `max_features` es crucial aquí.

### Configuración Óptima:

- **max\_features:**  $\sqrt{p}$  o  $\log_2(p)$  en lugar de  $p/3$
- **min\_samples\_leaf:** 3-5 para evitar hojas con un solo paciente
- **n\_estimators:** 500-1000 para estabilidad
- **class\_weight:** 'balanced' si hay desbalance entre casos/controles



- **Ventaja crítica:** Al considerar solo  $\sqrt{p}$  características en cada división, RF evita que unos pocos genes dominantes capturen toda la señal, permitiendo descubrir biomarcadores alternativos.



# Caso 2: Identificación de Biomarcadores



## Feature Importance en Genómica

La importancia de características en Random Forest no solo mejora las predicciones, sino que identifica **biomarcadores genéticos** potencialmente causales de la enfermedad.

### Ejemplo: Cáncer de Mama

Un estudio identificó 25 genes con alta importancia predictiva:

- **ESR1, PGR:** Receptores hormonales conocidos
- **ERBB2 (HER2):** Target terapéutico establecido
- **TP53, BRCA1:** Genes supresores tumorales
- **Nuevos candidatos:** 12 genes sin asociación previa

**Impacto:** Tres de estos genes nuevos fueron validados como dianas terapéuticas potenciales en estudios posteriores.

### Proceso de Descubrimiento:

1. Entrenar RF con todos los genes
2. Extraer importancias de características
3. Identificar top 50-100 genes
4. Validación biológica de los genes
5. Estudios funcionales en laboratorio



# Caso 2: Resultados Clínicos

## Alta Precisión y Curva ROC

El modelo Random Forest entrenado con datos de expresión génica alcanzó un rendimiento diagnóstico comparable o superior a métodos tradicionales de patología.

### Métricas de Desempeño:

**94%**

#### Precisión Global

Clasificación correcta de casos cancerosos vs.  
benignos

**92%**

#### Sensibilidad

Detección de verdaderos casos de cáncer

**96%**

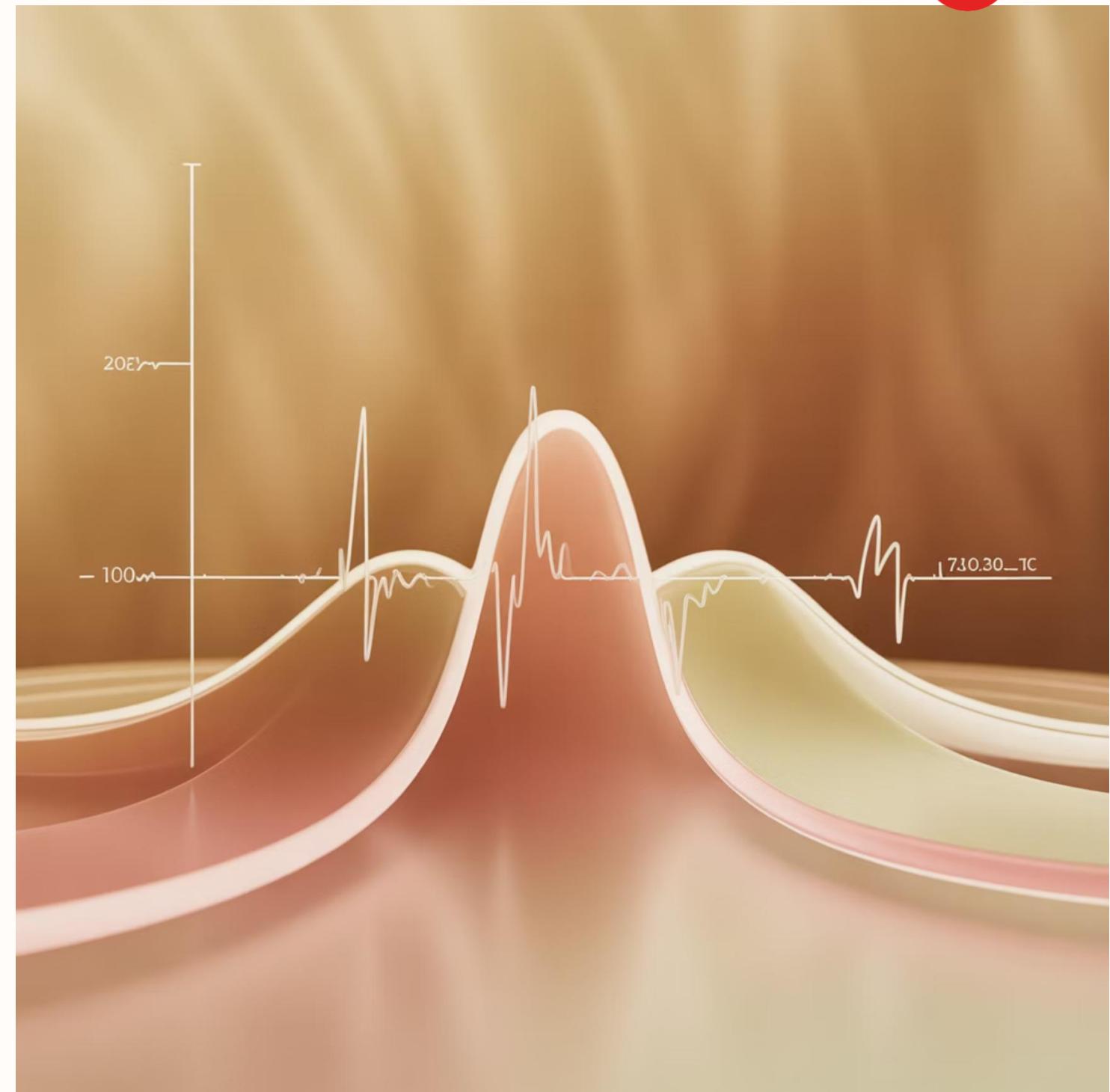
#### Especificidad

Identificación correcta de casos benignos

**0.97**

#### AUC-ROC

Excelente capacidad discriminativa





# Caso 3: Clasificación de Texto



## El Problema: Análisis de Sentimiento

El análisis de sentimiento busca determinar si un texto expresa una opinión positiva, negativa o neutral. Esto es fundamental para monitorear la reputación de marca, analizar feedback de clientes, y entender la opinión pública en redes sociales.

### Preprocesamiento Típico:

1. Tokenización (dividir en palabras)
2. Limpieza (minúsculas, puntuación)
3. Stopwords (eliminar palabras comunes)
4. Stemming/Lemmatization
5. Vectorización TF-IDF o Bag-of-Words
6. Resultado: matriz sparse de 10,000+ características

### Desafíos del Procesamiento de Texto:

- **Alta dimensionalidad:** Miles de palabras únicas en el vocabulario
- **Sparsidad:** Cada documento contiene solo una pequeña fracción de palabras
- **Contexto:** El significado depende del contexto ("no está mal" es positivo)
- **Sarcasmo e ironía:** Difíciles de detectar automáticamente



# Caso 3: ¿Por qué Random Forest?

## Manejo de "Bag of Words"

Después de vectorizar texto con TF-IDF, obtenemos miles de características (una por palabra o n-grama). Random Forest es ideal para este escenario de alta dimensionalidad con datos sparse.

### Selección Automática de Palabras Clave

max\_features asegura que cada árbol considere solo un subconjunto de palabras, previniendo que términos muy frecuentes dominen todas las decisiones.

### Robustez a Vocabulario Ruidoso

Errores ortográficos, jerga y variaciones lingüísticas no afectan dramáticamente el rendimiento, ya que el ensamble promedia sobre múltiples perspectivas.

### Sin Necesidad de Ingeniería Compleja

A diferencia de modelos lineales, RF no requiere selección manual de características o regularización sofisticada. La aleatoriedad inherente proporciona regularización natural.





# Caso 3: Palabras Clave Más Influuyentes

## Feature Importance en Texto

El análisis de importancia revela qué palabras o n-gramas son más predictivos del sentimiento, proporcionando insights sobre el lenguaje emocional en el dominio específico.

### Top Palabras para Sentimiento POSITIVO:

- "excelente" (importancia: 0.089)
- "increíble" (importancia: 0.076)
- "recomiendo" (importancia: 0.072)
- "encantó" (importancia: 0.068)
- "perfecto" (importancia: 0.061)
- "buena calidad" (importancia: 0.055)
- "vale la pena" (importancia: 0.049)
- "satisfecho" (importancia: 0.045)

### Top Palabras para Sentimiento NEGATIVO:

- "terrible" (importancia: 0.094)
- "decepcionante" (importancia: 0.081)
- "no funciona" (importancia: 0.077)
- "mala calidad" (importancia: 0.073)
- "dinero perdido" (importancia: 0.062)
- "no recomiendo" (importancia: 0.058)
- "defectuoso" (importancia: 0.051)
- "frustración" (importancia: 0.047)

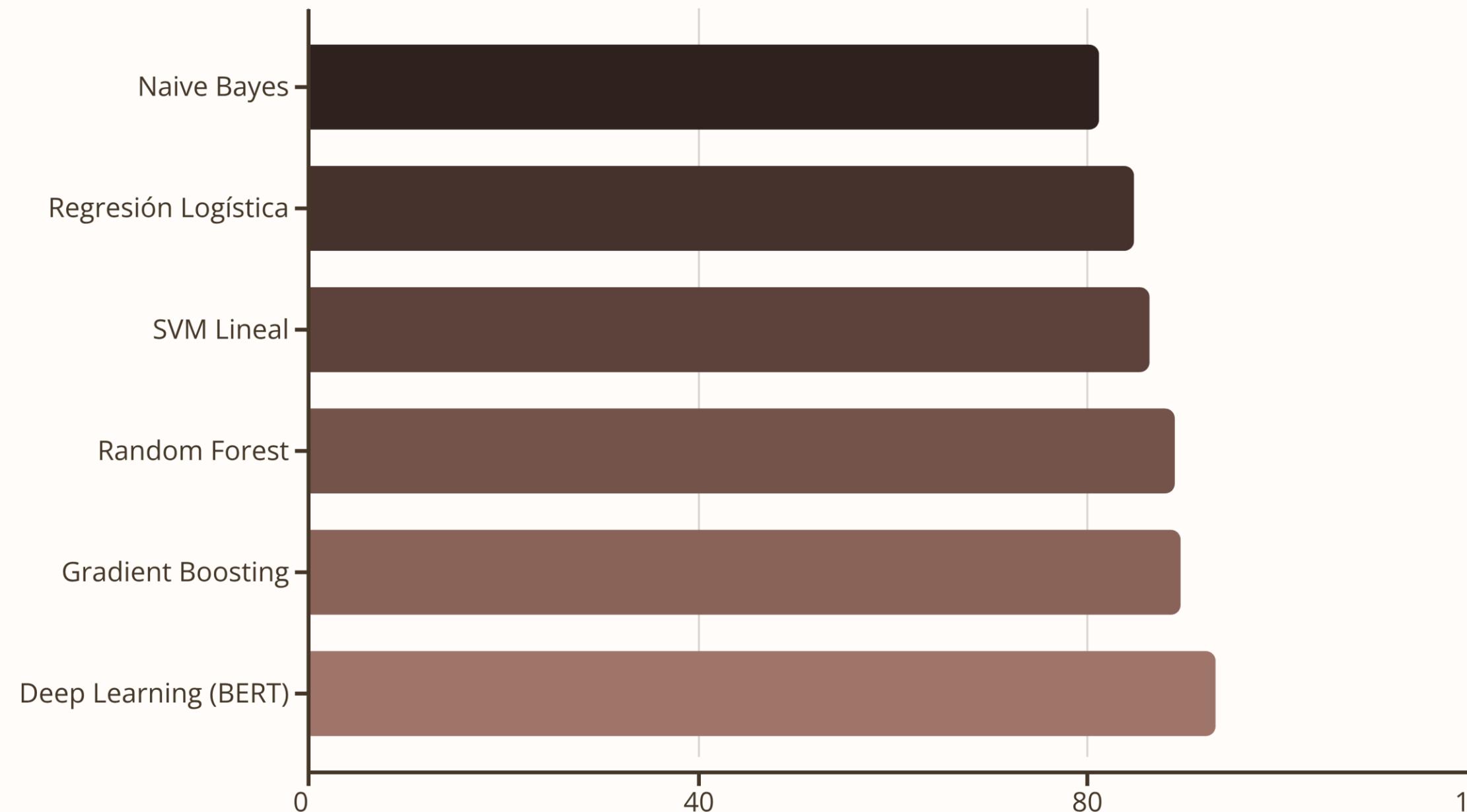
Nota: Los bigramas ("no funciona", "mala calidad") suelen ser más informativos que palabras individuales.



# Caso 3: Comparación con Otros Modelos

## Benchmarking de Algoritmos

Evaluamos Random Forest contra otros algoritmos comunes en clasificación de texto usando el mismo conjunto de datos de 50,000 reseñas de productos.





# Caso 4: Predicción del Precio de Acciones



## El Problema: Mercados Financieros

Predecir el precio futuro de acciones es uno de los problemas más desafiantes en machine learning debido a la naturaleza caótica y ruidosa de los mercados financieros. No buscamos predecir el precio exacto, sino capturar tendencias y patrones que mejoren las decisiones de inversión.

### Features Técnicos Utilizados:

1. Medias móviles (7, 14, 30, 60 días)
2. RSI (Relative Strength Index)
3. MACD (Moving Average Convergence Divergence)
4. Bollinger Bands
5. Volumen relativo
6. Ratios de precio (máximo/mínimo)
7. Momentum y ROC
8. Volatilidad histórica
9. Correlación con índices

### Desafíos Únicos:

- **Ruido extremo:** Señal débil oculta en volatilidad
- **No estacionariedad:** Patrones cambian constantemente
- **Eventos inesperados:** Noticias, políticas, crisis
- **Eficiencia del mercado:** Información pública ya está en el precio



# Caso 4: ¿Por qué Random Forest?

## Captura de Relaciones No Lineales

Los indicadores técnicos financieros tienen interacciones complejas y no lineales que los modelos lineales no pueden capturar eficientemente.

### Interacciones de Indicadores

RF captura automáticamente interacciones como: "Si RSI > 70 Y MACD cruza hacia abajo Y volumen está alto → Señal de venta"

### Régimenes de Mercado

El modelo aprende diferentes patrones para mercados alcistas vs. bajistas vs. laterales sin necesidad de especificarlos manualmente

### Robustez a Outliers

Eventos extremos (flash crashes, gaps de apertura) no distorsionan el modelo completo gracias al promediado

"En finanzas cuantitativas, no buscamos predicciones perfectas sino ventajas estadísticas pequeñas pero consistentes."

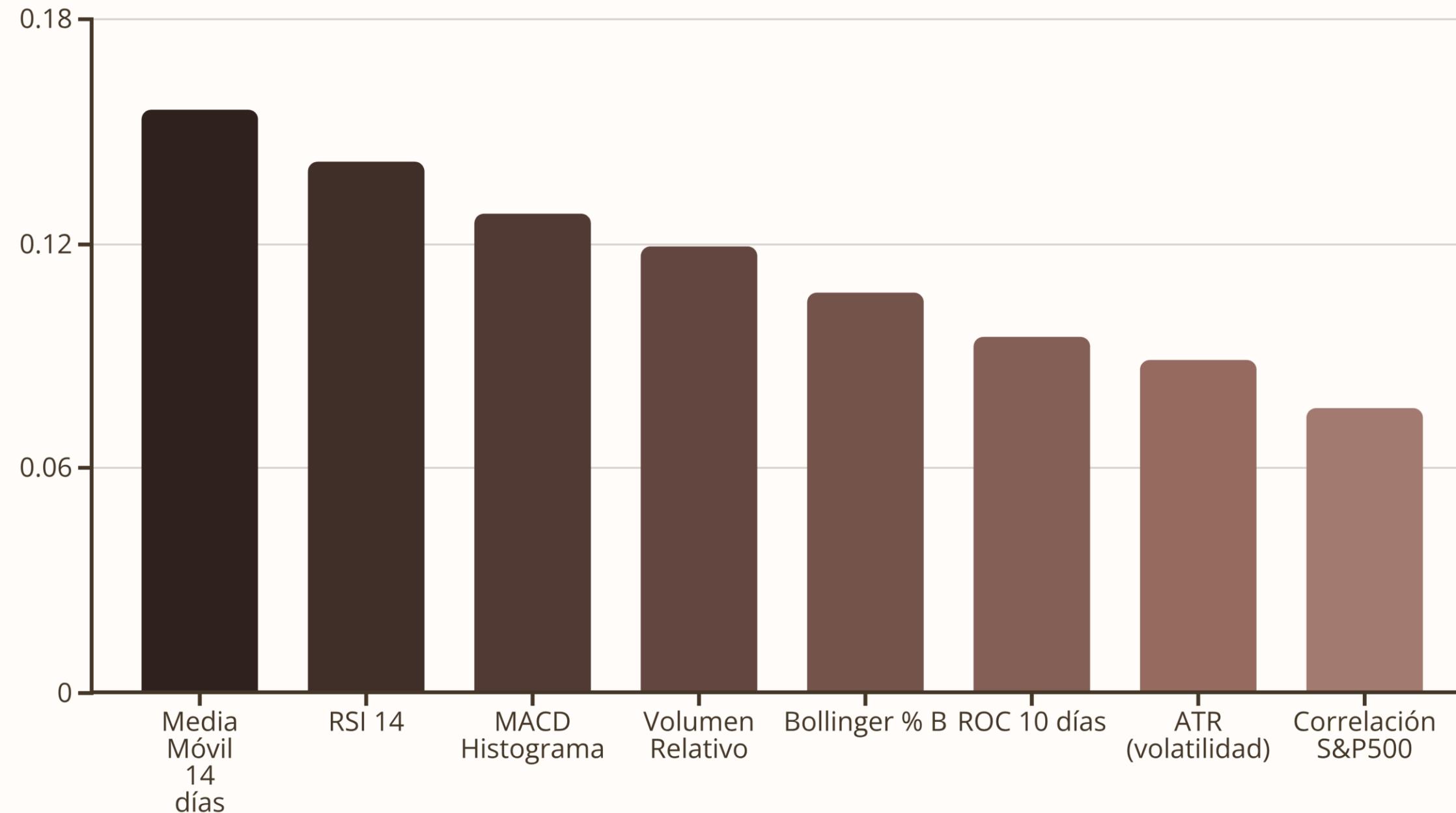




# Caso 4: Indicadores Técnicos Más Predictivos

## Feature Importance en Trading

El análisis de importancia revela cuáles indicadores técnicos contienen más información predictiva sobre los movimientos futuros del precio.





# Caso 4: Predicción vs. Valor Real

## Evaluación del Rendimiento

Evaluamos el modelo en el precio de cierre de Apple (AAPL) durante 2023, usando una ventana deslizante para simular trading real.

### Métricas de Error:

**\$2.84**

**MAE**

Error absoluto medio por predicción

**\$4.12**

**RMSE**

Penaliza errores grandes más fuertemente

**2.1%**

**MAPE**

Error porcentual medio

**0.87**

**R<sup>2</sup>**

Proporción de varianza explicada





# Python IA





# CONCLUSIONES

01

La regresión logística es ideal para problemas de clasificación binaria.

02

La regresión logística permite interpretar fácilmente la influencia de cada variable mediante coeficientes.

03

Los árboles de decisiones capturan relaciones no lineales sin requerir preprocesamiento extenso de datos.

04

Los árboles de decisiones son propensos al sobreajuste, especialmente con árboles profundos.



# CONCLUSIONES

05

Random Forest reduce el sobreajuste al combinar múltiples árboles.

06

Random Forest tiene un alto costo computacional, especialmente con muchos árboles.

07

La regresión logística ofrece interpretabilidad a costa de simplificar relaciones, mientras que los árboles de decisión y Random Forest aumentan la precisión y complejidad, pero sacrifican claridad en la interpretación.



## Bibliografía

- Bastián Gutiérrez , Macarena Dehnhardt , Roberto Cortés , Alexis Matheu , Cristián Cornejo. Modelo logístico de deserción mediante técnicas de regresión y árbol de decisión para la eficiencia en la destinación de recursos: el caso de una universidad privada chilena. Revista Ibérica de Sistemas y Tecnologías de la Información.
- Aurélien Géron. Aprendizaje automático práctico con Scikit-Learn, Keras y TensorFlow, 3<sup>a</sup> edición. O'Reilly Media, Inc.
- Revisa esta lectura, haciendo clic en este enlace:  
<https://wwwproquest.upc.elogim.com/docview/3085715684/fulltextPDF/7B98A6A5D07247C2PQ/1?accountid=43860&sourcetype=Scholarly%20Journals>