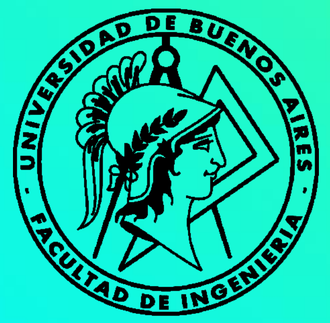


ÁRBOLES DE DECISIÓN

APRENDIZAJE DE MAQUINA - CEIA - FIUBA

Dr. Ing. Facundo Adrián Lucianna

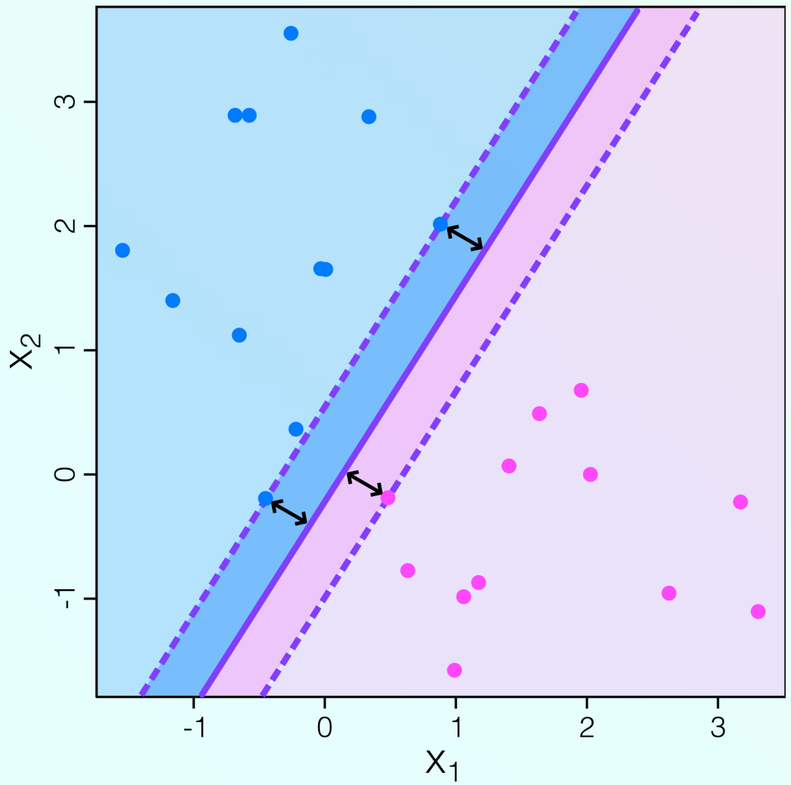
Esp. Lic. María Carina Roldán

REPASO CLASE ANTERIOR

- Maximal Margin Classifier
- Clasificador de vector de soportes
- Support Vector Machines para clasificar
- Support Vector Machines para regresión

MAXIMAL MARGIN CLASSIFIER

Si nuestros datos son linealmente separables, puede existir un número infinito de hiperplanos que funcionen



Por lo tanto, necesitamos algún criterio de selección.

El caso que estamos analizando busca el hiperplano que se encuentre lo más alejado posible de los datos de entrenamiento.

Es decir, calculamos la distancia mínima de cada observación de entrenamiento al hiperplano y obtenemos la más pequeña de esas distancias, a la que llamamos margen.

El objetivo es encontrar el hiperplano que tenga el **mayor margen posible**. El algoritmo que realiza esta tarea se conoce como Maximal Margin Classifier

Podemos pensar que este clasificador busca el máximo grosor de la "franja" o "banda" que puede pasar entre las clases.

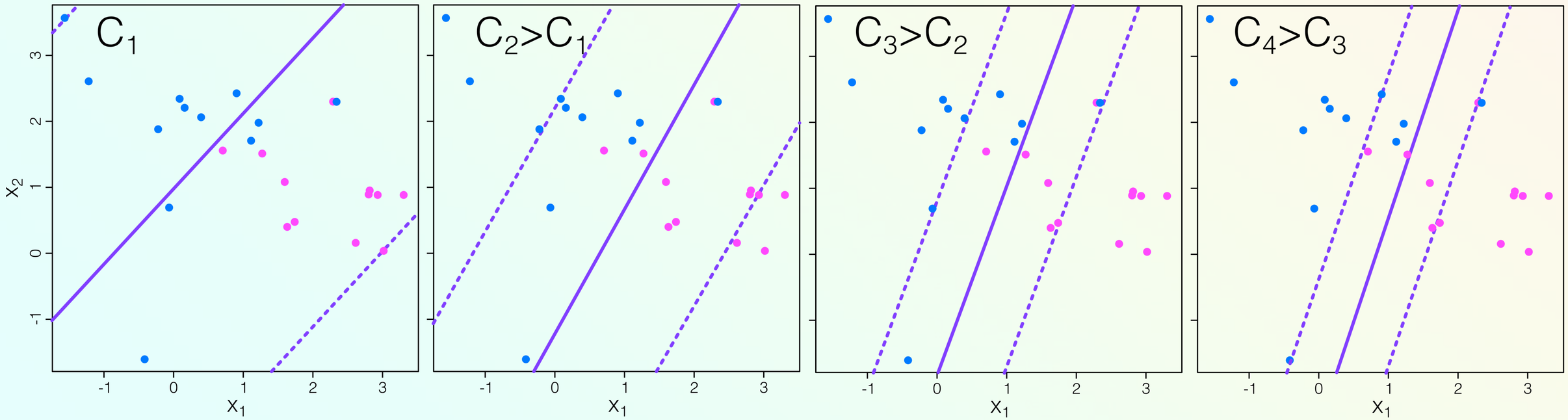
CLASIFICADOR DE VECTOR DE SOPORTES

Por lo que vimos, si queremos seguir utilizando un hiperplano, debemos relajar las exigencias:

- Mayor robustez frente a observaciones individuales.
- Mejor clasificación de la mayoría (aunque no de todas) las observaciones de entrenamiento.

Es decir, podría valer la pena clasificar erróneamente algunas observaciones de entrenamiento para poder clasificar mejor las restantes.

CLASIFICADOR DE VECTOR DE SOPORTES



Hay un pequeño número de observaciones que se encuentran en el margen o dentro de él, y son las que determinan el hiperplano. A estas se las llama vectores de soporte.

Las demás observaciones **no tienen importancia para el modelo**.

SUPPORT VECTOR MACHINE

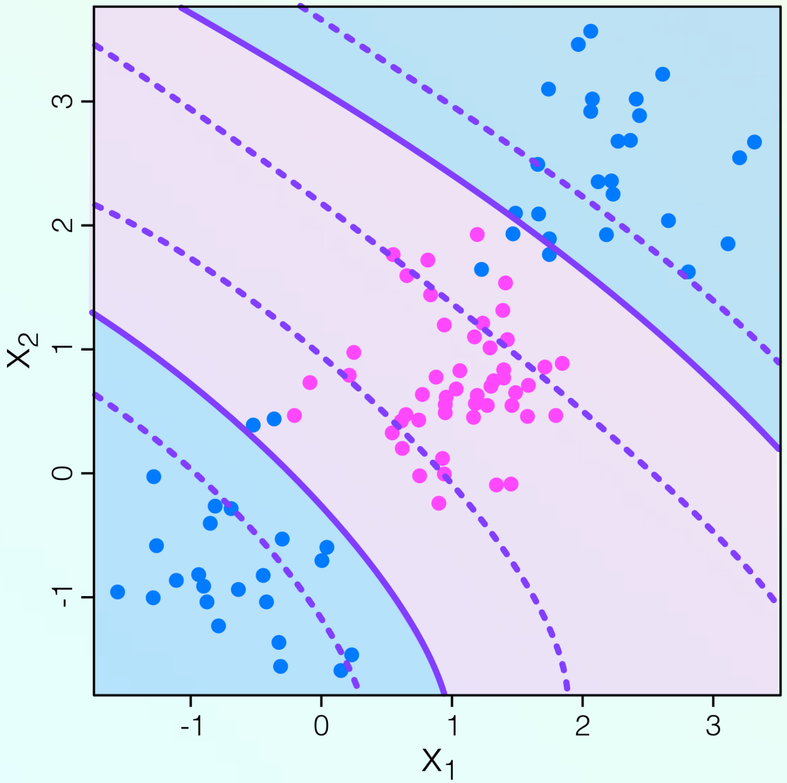
El modelo llamado Support Vector Machine (SVM), o Máquina de Vectores de Soporte, extiende al Clasificador de Vectores de Soporte permitiendo ampliar el espacio de features mediante el uso de funciones kernel.

La frontera de decisión se puede describir como:

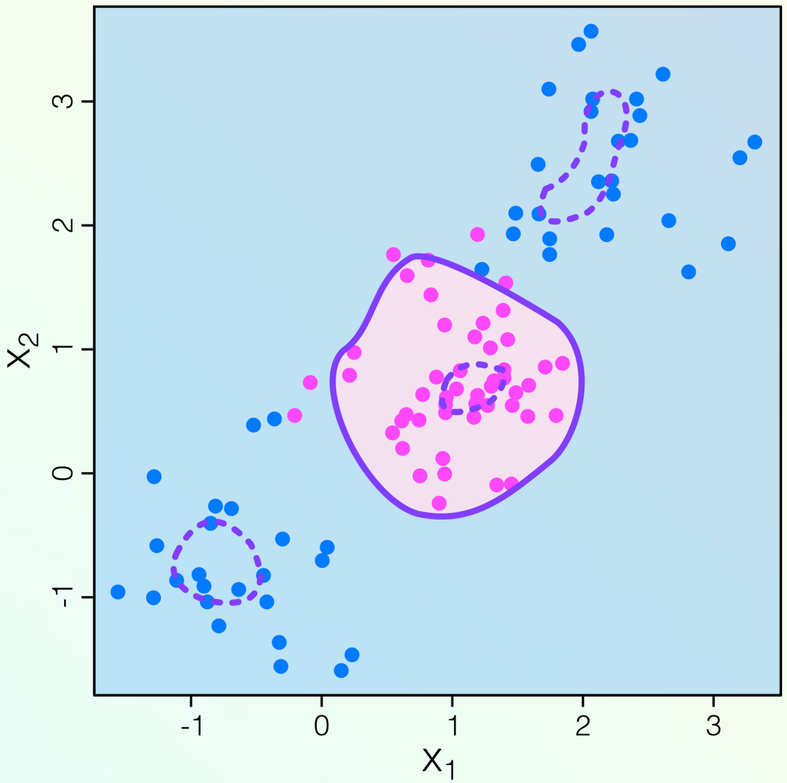
$$f(X) = \beta_0 + \sum_{i \in \mathcal{H}} \alpha_i K(X_i, X)$$

La forma de entrenamiento, la importancia de los vectores de soporte y el hiperparámetro C se mantienen. La gran diferencia es que ahora las fronteras de decisión no son necesariamente lineales, sino que están determinadas por la función kernel elegida

SUPPORT VECTOR MACHINE

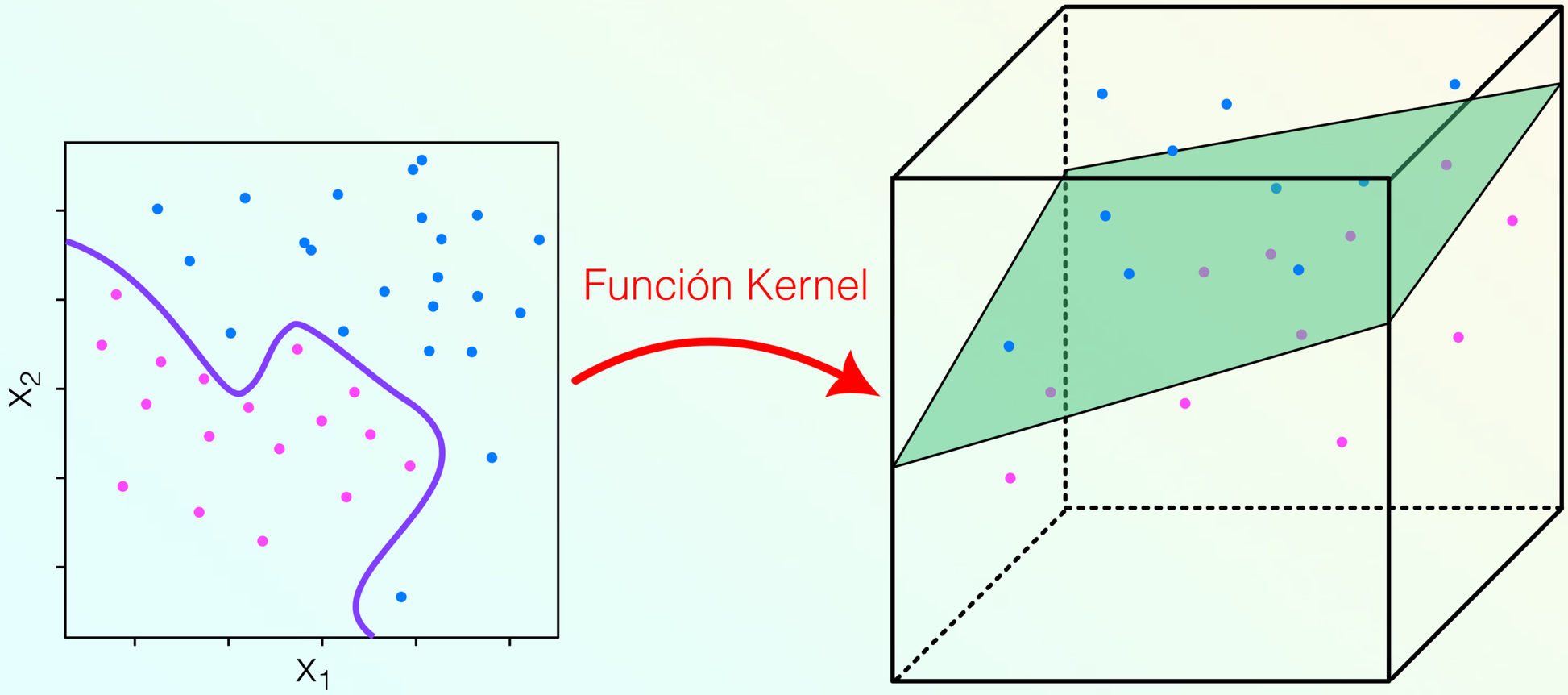


Kernel polinomial de orden 3



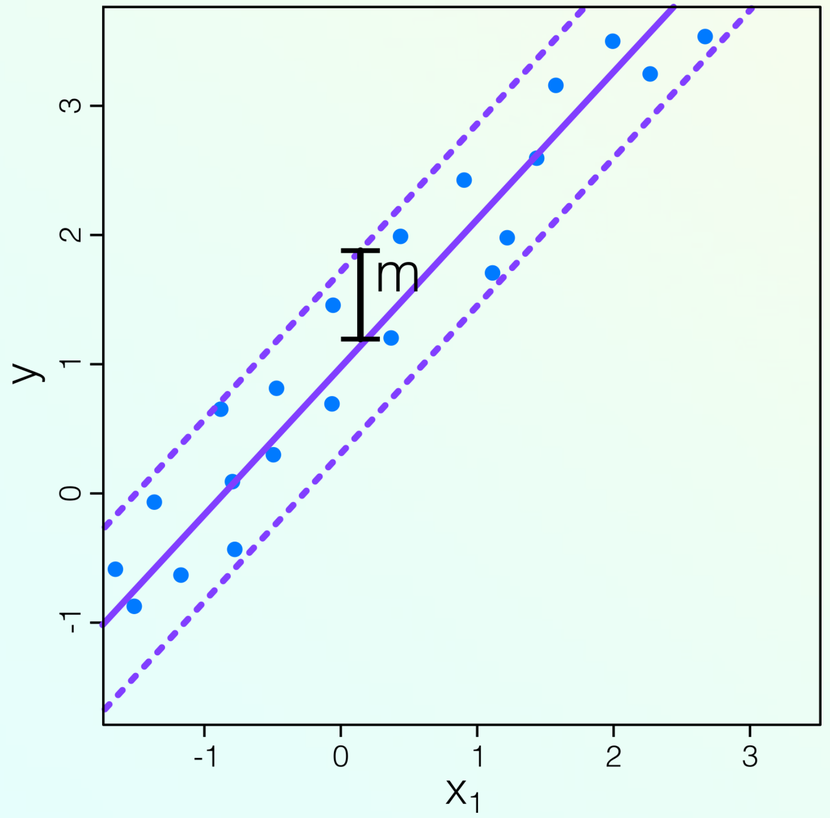
Kernel radial

SUPPORT VECTOR MACHINE

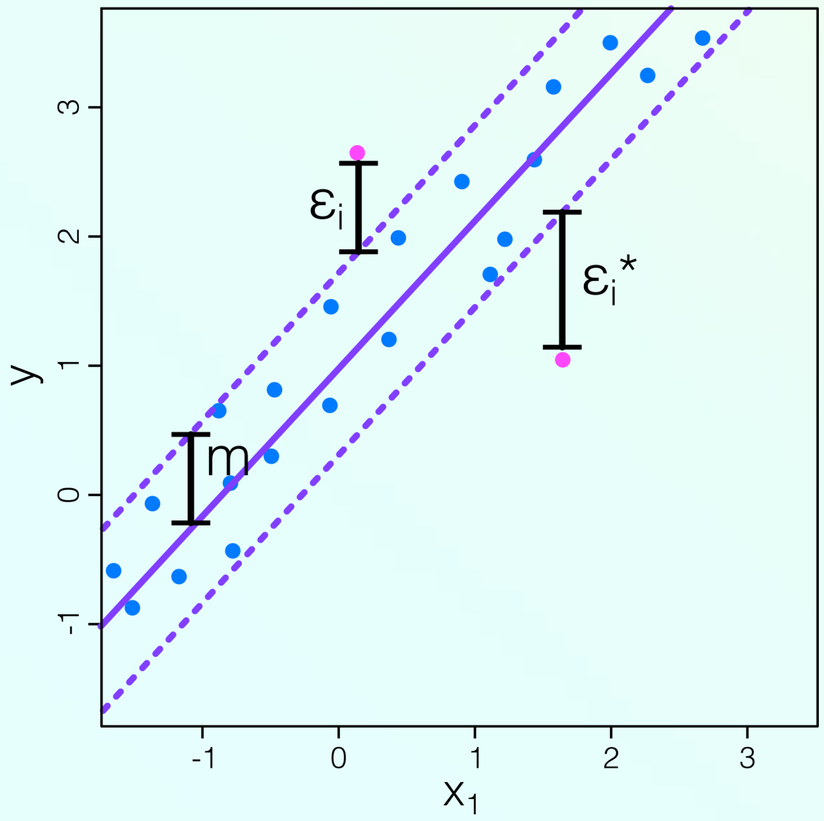


SUPPORT VECTOR MACHINE EN REGRESIÓN

Lo que se optimiza ahora es el hiperplano que mejor logra incluir a todos los puntos de entrenamiento dentro del margen, minimizando el valor de dicho margen.



SUPPORT VECTOR MACHINE EN REGRESIÓN



$$\sum_{i=1}^{n}(\epsilon_i + \epsilon_i^*) \leq \frac{1}{C} \qquad \epsilon_i, \epsilon_i^* \geq 0 \qquad \forall i = 1, \dots, n$$

La restricción es:

$$y_i - \left(b_0 + < \vec{b}, X >\right) \leq m + \epsilon_i$$

$$\left(b_0 + < \vec{b}, X >\right) - y_i \leq m + \epsilon_i^* \qquad \forall i = 1, \dots, n$$

ÁRBOLES DE DECISIÓN

ÁRBOLES DE DECISIÓN

Los árboles de clasificación y regresión, conocidos como CART (Classification and Regression Trees), son una poderosa técnica de aprendizaje automático que se utiliza ampliamente para resolver problemas tanto de clasificación como de regresión.

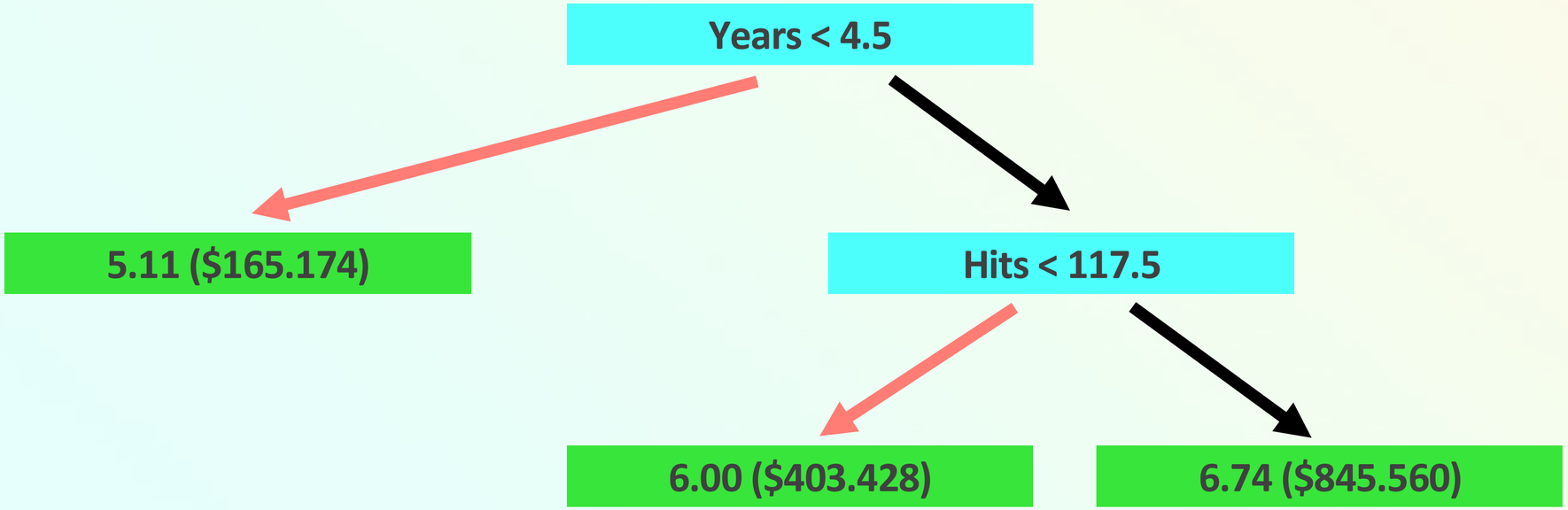
Los árboles CART son modelos de decisión que emplean una estructura jerárquica en forma de árbol para realizar predicciones basadas en reglas lógicas sencillas y fáciles de interpretar.

Arboles de clasificación

Arboles de regresión

ÁRBOLES DE REGRESIÓN

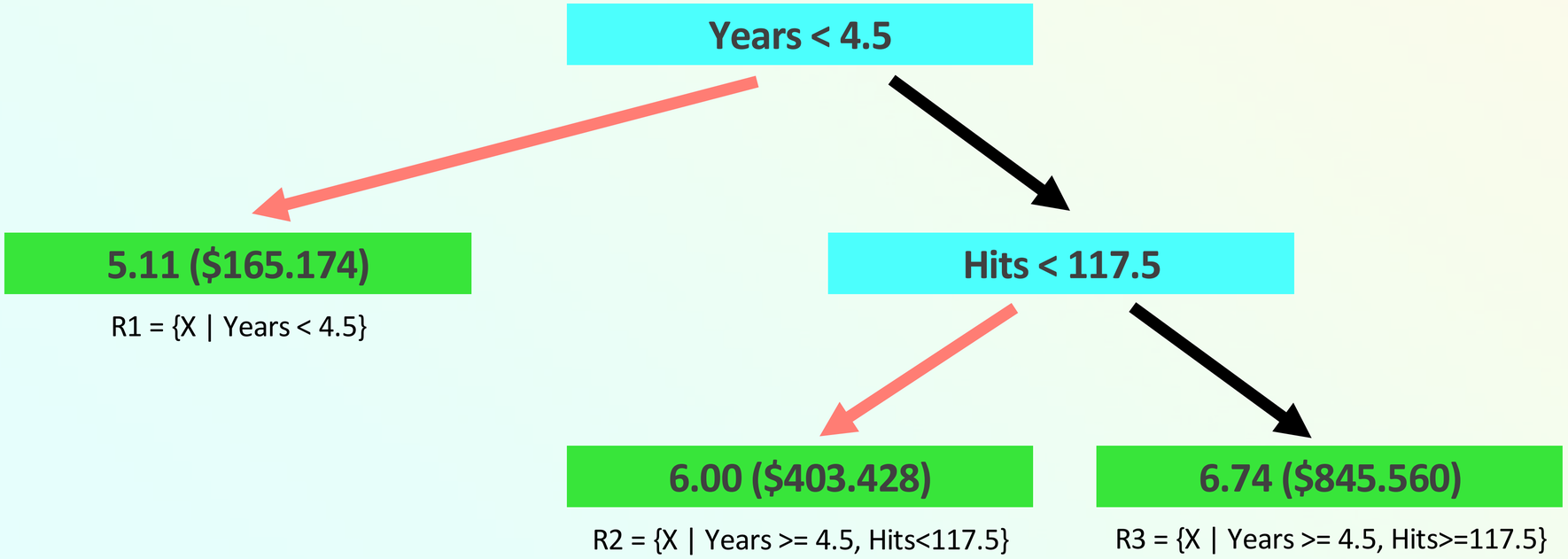
Empecemos con un ejemplo sencillo utilizando el dataset Hitters, que contiene datos de salarios de jugadores de béisbol de 1987 y sus estadísticas deportivas de 1986.



En este caso, se predice el logaritmo del salario (ya que su distribución es más cercana a una forma de campana). Years representa los años jugando en las ligas, mientras que Hits indica el número de hits logrados durante el año anterior.

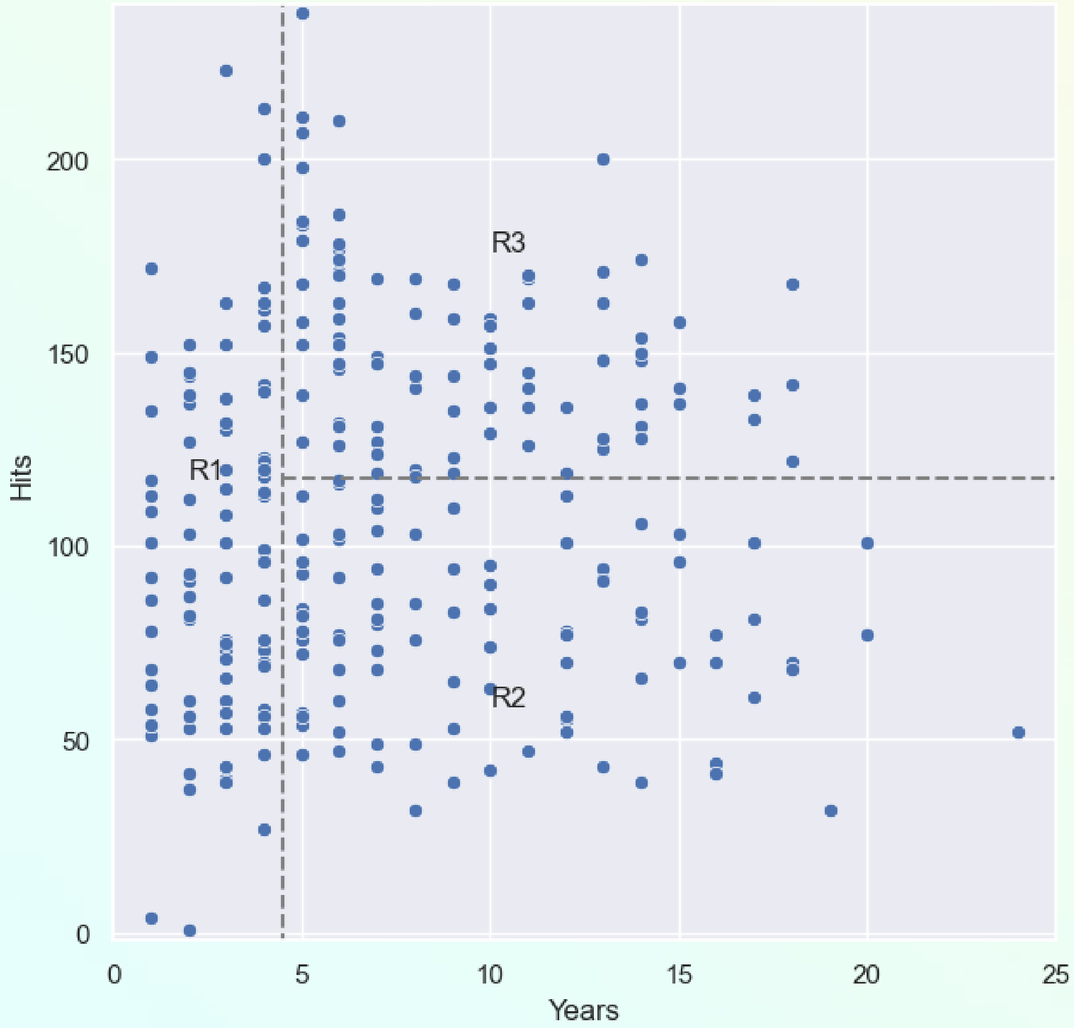
ÁRBOLES DE REGRESIÓN

Empecemos con un ejemplo sencillo utilizando el dataset Hitters, que contiene datos de salarios de jugadores de béisbol de 1987 y sus estadísticas deportivas de 1986.



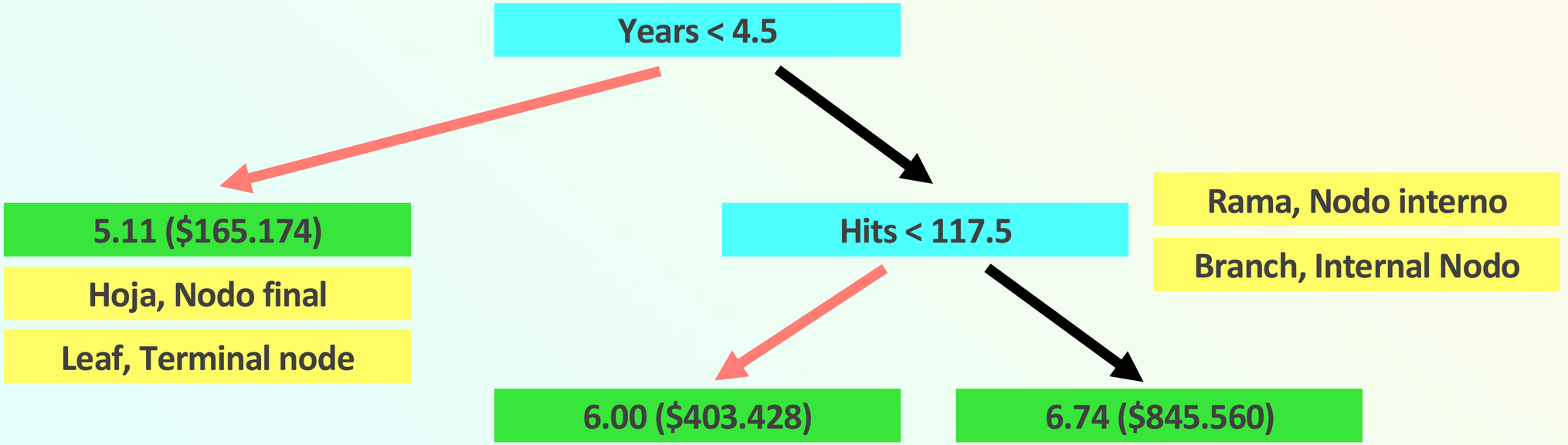
En este caso, se predice el logaritmo del salario (ya que su distribución es más cercana a una forma de campana). Years representa los años jugando en las ligas, mientras que Hits indica el número de hits logrados durante el año anterior.

ÁRBOLES DE REGRESIÓN



ÁRBOLES DE REGRESIÓN

Empecemos con un ejemplo sencillo, usando el dataset Hitters (Dato de salarios de jugadores de Beisbol de 1987 y estadísticas deportivas de 1986)



Este árbol de regresión es una simplificación excesiva de la verdadera relación de regresión entre Salary, Years y Hits. Sin embargo, tiene sus ventajas: es más fácil de interpretar y ofrece una representación gráfica más intuitiva.

ÁRBOLES DE REGRESIÓN

Proceso de construcción de un árbol de regresión:

1. Dividimos el espacio de observaciones —es decir, el conjunto de los valores posibles de $X_1, X_2, \ldots, X_p$ — en J regiones distintas y no solapadas, $R_1, R_2, \ldots, R_J$.

2. Para cada observación que cae en una región R_j, realizamos la misma predicción, que consiste simplemente en la media de la variable respuesta de las observaciones de entrenamiento dentro de esa región R_j.

   También podríamos utilizar otra medida de tendencia central si fuera conveniente.

ÁRBOLES DE REGRESIÓN

¿Cómo dividimos el espacio de observaciones?

En teoría, podríamos dividir el espacio en cualquier tipo de regiones, pero se eligen regiones *rectangulares* para simplificar el modelo.

El objetivo es encontrar las regiones R_1, …, R_J que minimicen la suma de los residuos al cuadrado (RSS), dada por:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

ÁRBOLES DE REGRESIÓN

¿Cómo dividimos el espacio de observaciones?

En teoría, podríamos dividir el espacio en cualquier tipo de regiones, pero se eligen regiones *rectangulares* para simplificar el modelo.

El objetivo es encontrar las regiones R_1, …, R_J que minimicen la suma de los residuos al cuadrado (RSS), dada por:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

Es la media de y en la región R_J

ÁRBOLES DE REGRESIÓN

¿Cómo se lleva a cabo esta división?

Buscar todas las combinaciones posibles de valores para minimizar el RSS es computacionalmente imposible.

Por eso, utilizamos un algoritmo de optimización heurística denominado “recursive binary splitting” (división binaria recursiva).

- Top-down: comenzamos desde el tronco del árbol y avanzamos hacia las ramas.
- Greedy: en cada paso, se elige la mejor bifurcación local (la que más reduce el RSS en ese paso), sin considerar el efecto global.

ÁRBOLES DE REGRESIÓN

Recursive binary splitting

Se elige una variable X_j y un punto de corte ***s*** de tal forma que el espacio de features se divide en dos regiones:

$$R_1(j,s) = \{X | X_j < s\} \qquad R_2(j,s) = \{X | X_j \geq s\}$$

Buscamos los valores de j y s que minimizan la suma de los residuos al cuadrado (RSS) según la siguiente ecuación:

$$\sum_{i:x_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2$$

ÁRBOLES DE REGRESIÓN

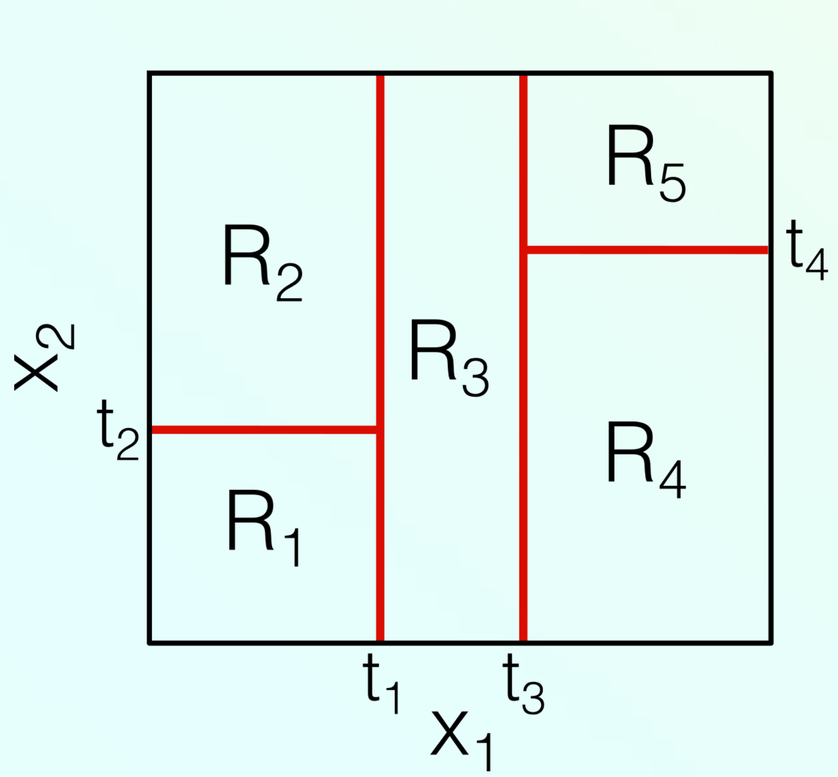
Recursive binary splitting

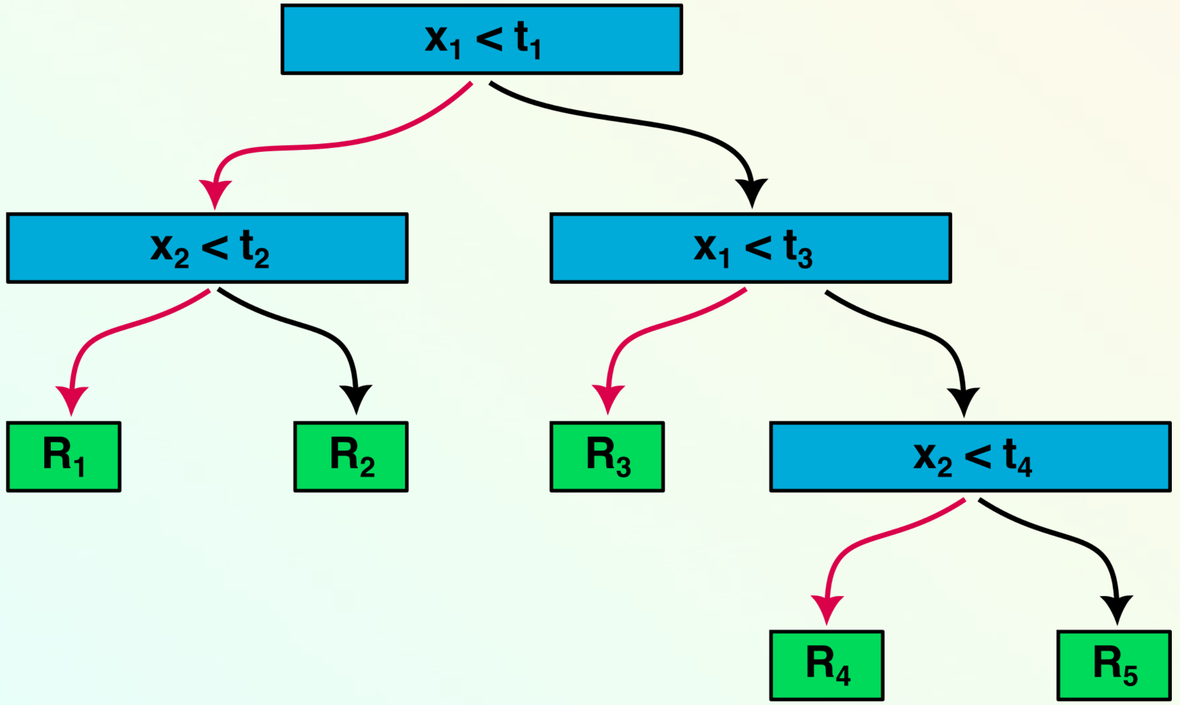
Recursivamente repetimos este proceso para los segmentos generados, pero ahora tomando las regiones formadas y aplicando el mismo procedimiento, dividiéndolas en nuevas subregiones.

Este proceso continúa hasta que se alcanza un criterio de detención (stopping criterion).

ÁRBOLES DE REGRESIÓN

Recursive binary splitting





ÁRBOLES DE REGRESIÓN

Podando los arboles

El proceso descrito puede producir buenas predicciones sobre el conjunto de entrenamiento, pero con mucha facilidad puede generar overfitting, lo que lleva a un mal desempeño en el conjunto de validación.

El caso más extremo es un árbol que tiene **una hoja por cada punto** del conjunto de entrenamiento.

Esto ocurre porque el árbol es demasiado complejo.

Un árbol más pequeño, con menos regiones, puede tener menor **varianza** y ofrecer mejor interpretabilidad, a expensas de un leve aumento en el **sesgo**.

ÁRBOLES DE REGRESIÓN

Podando los arboles

La estrategia más obvia sería construir el árbol únicamente mientras la disminución del RSS sea mayor que un valor umbral (relativamente alto).

Esta estrategia da como resultado árboles más pequeños, pero es demasiado miope:

Una división que parece poco útil en las primeras etapas podría ser seguida por otra división muy valiosa, que reduzca significativamente el RSS más adelante.

ÁRBOLES DE REGRESIÓN

Una mejor estrategia: eliminación hacia atrás

Una mejor alternativa es realizar un proceso de poda hacia atrás. Primero construimos un árbol grande T_0 y luego lo vamos podando para obtener un subárbol más simple.

¿Cómo lo hacemos?

Intuitivamente, buscamos el subárbol que minimice el error de validación.

Sin embargo, este es un problema demasiado complejo para resolverlo mediante simple iteración, por lo que se introduce un enfoque más sistemático.

ÁRBOLES DE REGRESIÓN

Podando los árboles con penalización

Introducimos un valor de penalización α en nuestra fórmula del RSS.

Dado un valor de α, existe un subárbol T⊆T0 que minimiza:

$$\sum_{m=1}^{L} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_m}\right)^2 + \alpha L$$

Donde:

L es el número de hojas del subárbol.

$\alpha \geq 0$ controla el compromiso entre la complejidad del árbol y su capacidad de ajuste (regularización).

Si $\alpha = 0$, el árbol elegido es T_0.

A medida que α aumenta, el costo por tamaño del árbol crece, favoreciendo modelos más simples.

ÁRBOLES DE REGRESIÓN

Comportamiento al aumentar α

Algo interesante es que, a medida que aumentamos α desde cero, las ramas se podan del árbol de forma anidada y predecible.

Esto permite obtener fácilmente la secuencia completa de subárboles en función de α.

El valor óptimo de α puede seleccionarse mediante un proceso de búsqueda de hiperparámetros (por ejemplo, validación cruzada).

ÁRBOLES DE REGRESIÓN

Algoritmo completo

1. Utilizando **recursive binary splitting**, se crea el árbol más grande posible con el conjunto de entrenamiento, deteniendo el proceso cuando cada hoja contenga menos de un número mínimo de observaciones.

2. Se aplica la técnica de poda del árbol para obtener un conjunto de subárboles en función del parámetro α, utilizando validación cruzada para determinar el mejor valor de α.

3. Se elige el subárbol del paso anterior que corresponde al valor de α que minimiza el error de validación.

VAMOS A PRÁCTICAR UN POCO…

ÁRBOL DE CLASIFICACIÓN

ÁRBOL DE CLASIFICACIÓN

Un árbol de clasificación es muy similar a un árbol de regresión, pero se utiliza para predecir una variable cualitativa.

En el caso de regresión, al llegar a una hoja, obteníamos el valor promedio de las observaciones en esa hoja.

Ahora, se obtiene la clase más frecuente entre las observaciones que caen en la hoja.

Al interpretar los resultados de un árbol de clasificación, a menudo nos interesa no solo la predicción de clase correspondiente a un nodo terminal, sino también las proporciones de cada clase entre las observaciones de entrenamiento que caen en esa región.

ÁRBOL DE CLASIFICACIÓN

La forma más simple de crear un árbol de clasificación es muy similar a la del árbol de regresión, usando la estrategia top-down y greedy.

Sin embargo, no contamos con el **error cuadrático**.

Una primera métrica que podemos usar es la tasa de error de clasificación:

$$E = 1 - \max_k(\hat{p}_{mk})$$

Donde $\hat{p}_{mk}$ representa la proporción de observaciones de entrenamiento en la región m que pertenecen a la clase k.

Esta métrica indica la fracción de observaciones en la región que no pertenecen a la clase más frecuente

⚠ El error de clasificación **no es suficientemente sensible para crecer árboles complejos**, por lo que en la práctica se prefieren otras dos medidas: índice de Gini y entropía.

ÁRBOL DE CLASIFICACIÓN

Índice de Gini

El índice de Gini es una medida de desigualdad que originalmente se usó para medir la desigualdad económica de países:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Se puede interpretar como una medida de la varianza entre las clases.

El índice de Gini toma valores pequeños si todas $\hat{p}_{mk}$ son cercanas a 0 o 1, es decir, si la región es “pura”.

Por eso, se dice que el índice de Gini mide la pureza de un nodo terminal.

ÁRBOL DE CLASIFICACIÓN

Entropía

En un árbol de clasificación, queremos encontrar divisiones que proporcionen mucha información.

Una rama es excelente si separa claramente las clases: todas las observaciones de una clase van a un lado y todas las de otra clase al otro.

La entropía mide la cantidad de desorden o incertidumbre:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

- Si todas las observaciones de entrenamiento en una hoja pertenecen a una sola clase, la entropía es cero.
- Si las clases están repartidas uniformemente, la entropía es grande.
- Valores de $\hat{p}_{mk}$ cercanos a 0 o 1 producen entropía baja, mientras que proporciones similares generan entropía alta.

ÁRBOL DE CLASIFICACIÓN

ARBOL DE CLASIFICACIÓN

Similitudes con los árboles de regresión

El resto del procedimiento es igual al árbol de regresión: división recursiva, poda y selección de subárboles óptimos.

VENTAJAS Y DESVENTAJAS DE LOS ÁRBOLES

✅ Fáciles de explicar a personas, incluso más que la regresión lineal.

✅ Se aproximan a la forma de pensar humana.

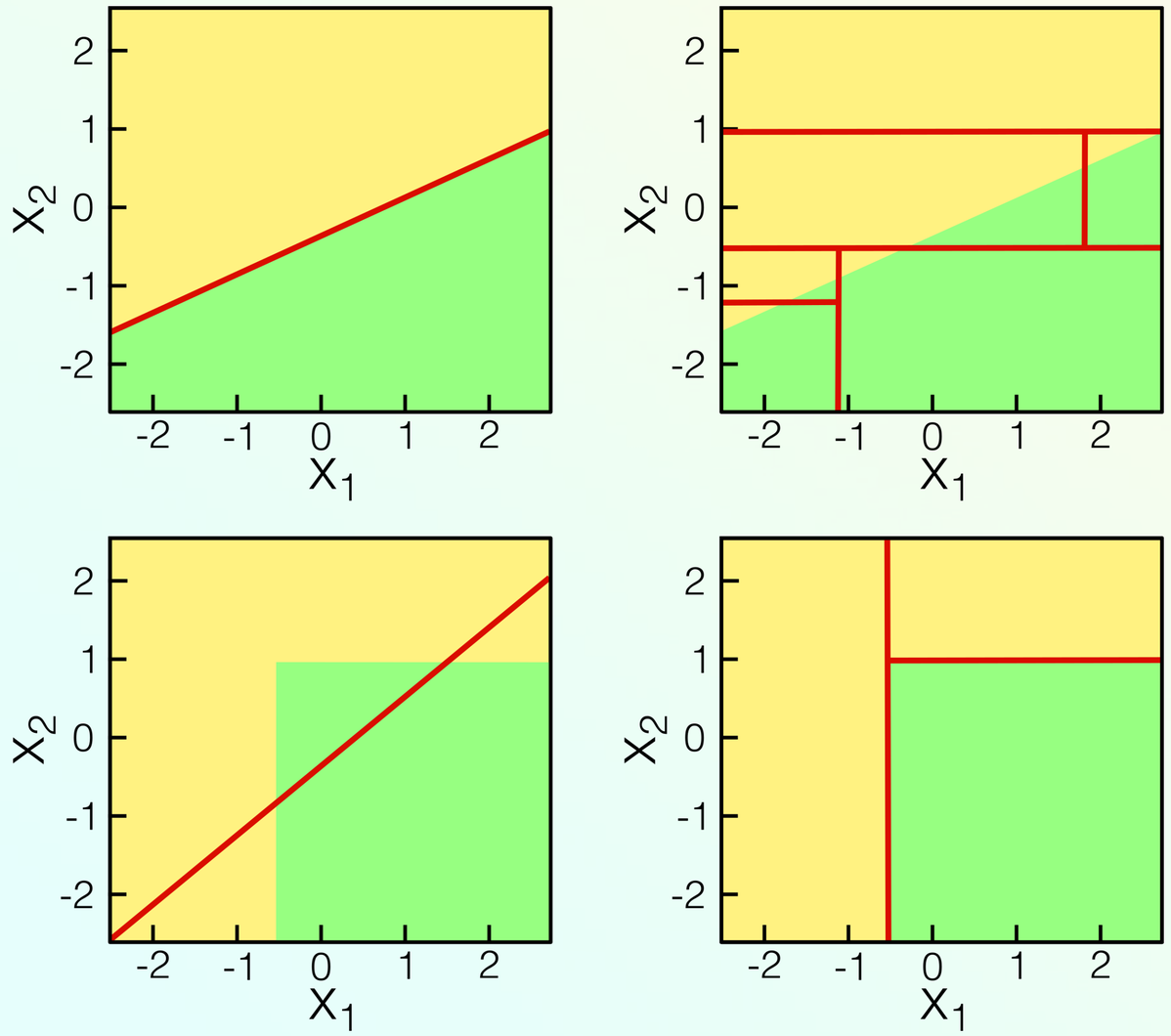
✅ Se pueden representar gráficamente.

✅ Pueden manejar variables cualitativas sin necesidad de crear dummies.

🚫 No tienen el mismo nivel de precisión predictiva que otros modelos.

🚫 No son robustos: pequeños cambios en los datos pueden generar grandes variaciones en la predicción.

VENTAJAS Y DESVENTAJAS DE LOS ÁRBOLES



VAMOS A PRÁCTICAR UN POCO…