# TDVI: Inteligencia Artificial

## Árboles de decisión

UTDT - LTD

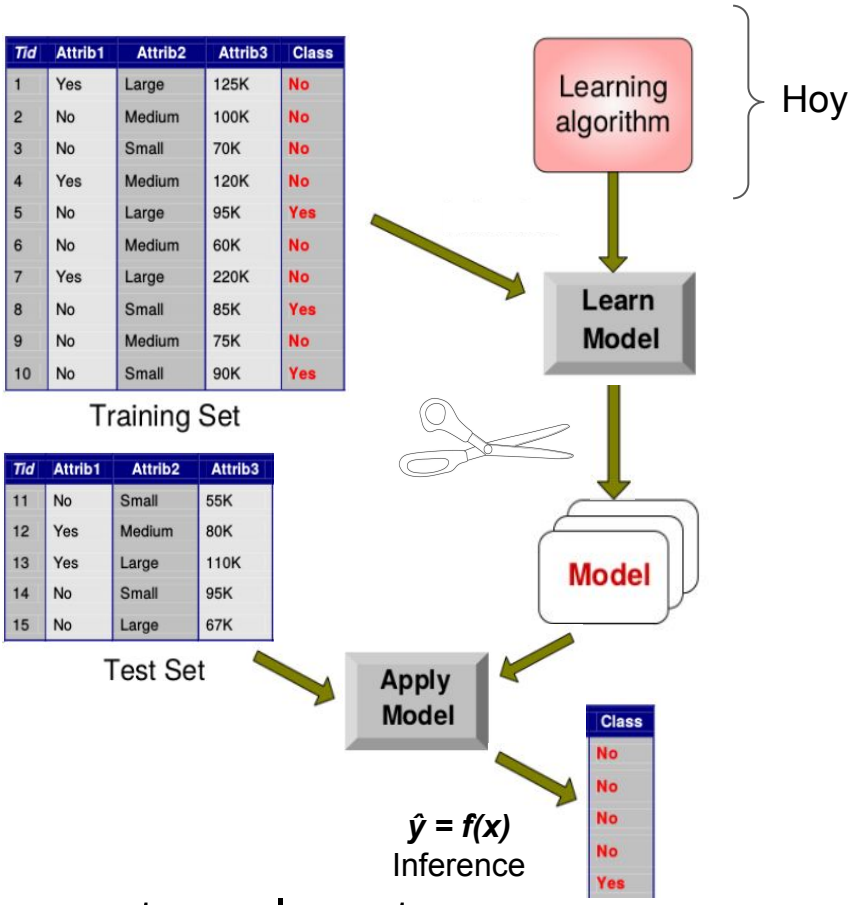
# Estructura de la clase

- Repaso de Aprendizaje Supervisado
- Noción de espacio de atributos
- Árboles de decisión

# Estructura de la clase

- Repaso de Aprendizaje Supervisado
- Noción de espacio de atributos
- Árboles de decisión

# Repaso de Aprendizaje Supervisado




Estos conjuntos son mutuamente excluyentes para asegurar que el modelo se pruebe con datos que no ha visto durante el entrenamiento, evitando sesgos.

# Estructura de la clase

- Repaso de Aprendizaje Supervisado
- Noción de espacio de atributos
- Árboles de decisión

# Noción de espacio de atributos

Durante la primera mitad del curso trabajaremos casi exclusivamente con conjuntos de datos "rectangulares":

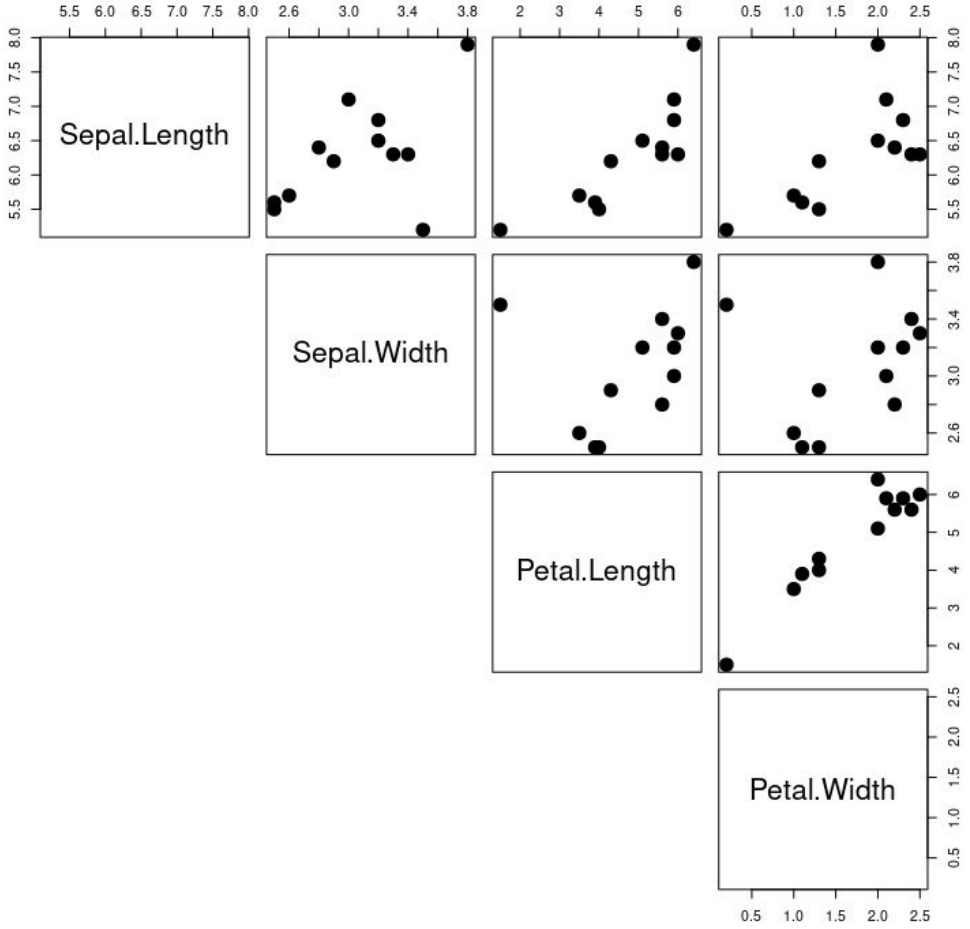
- Cada fila representa una observación
- Cada columna un atributo medido para cada observación

Ejemplo: una muestra de doce observaciones de "iris"

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|---|---|---|---|
| 5.2 | 3.5 | 1.5 | 0.2 | setosa |
| 5.7 | 2.6 | 3.5 | 1.0 | versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | virginica |
| 6.5 | 3.2 | 5.1 | 2.0 | virginica |
| 6.3 | 3.4 | 5.6 | 2.4 | virginica |
| 6.4 | 2.8 | 5.6 | 2.2 | virginica |
| 6.8 | 3.2 | 5.9 | 2.3 | virginica |
| 7.9 | 3.8 | 6.4 | 2.0 | virginica |
| 6.2 | 2.9 | 4.3 | 1.3 | versicolor |
| 7.1 | 3.0 | 5.9 | 2.1 | virginica |
| 5.5 | 2.5 | 4.0 | 1.3 | versicolor |
| 5.6 | 2.5 | 3.9 | 1.1 | versicolor |

# Noción de espacio de atributos

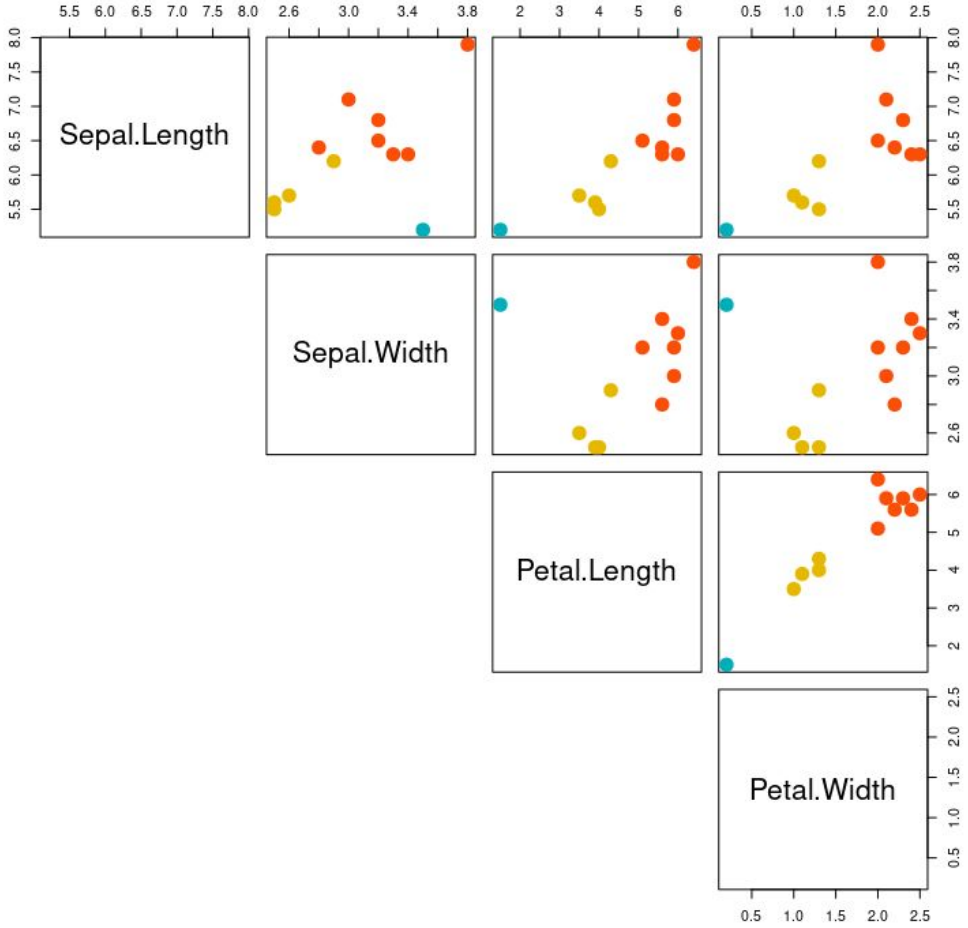
Los posibles valores de las variables predictoras definen lo que se conoce como el ”espacio de atributos”. Cada observación se corresponde con una coordenada (i.e., un vector) en el mismo



# Noción de espacio de atributos

¿Las observaciones de la misma clase tienden a estar concentradas en ciertas regiones?

¿Qué atributo parece bueno para predecir las iris virginica?




setosa

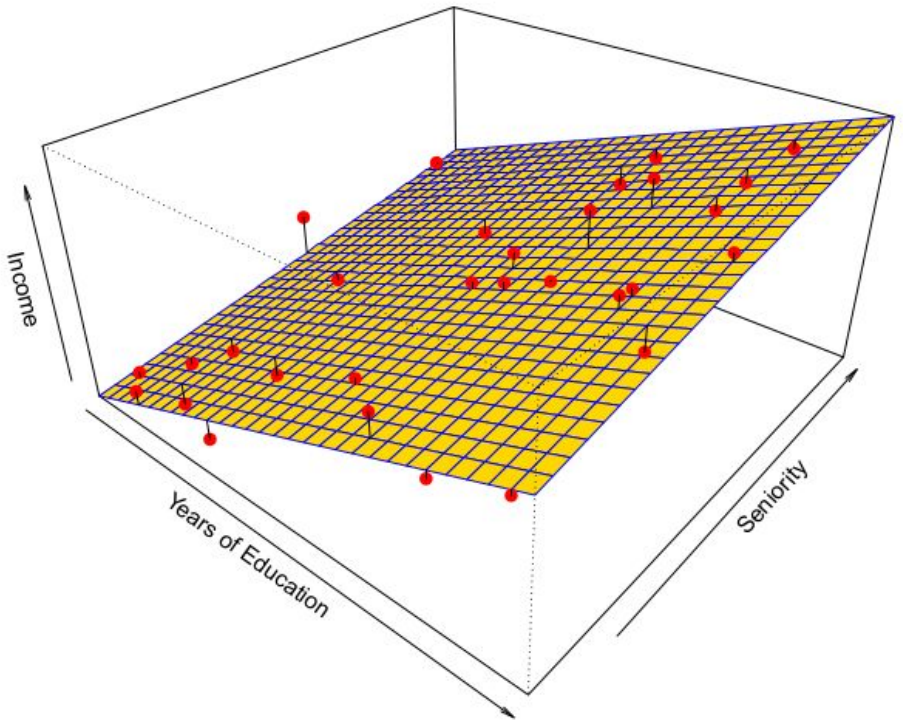
versicolor

virginica

# Noción de espacio de atributos

Para regresión la noción de espacio de atributos es idéntica. Recuerden que sólo depende de las X

Ahora la variable a predecir se puede visualizar como una dimensión adicional (que no pertenece al espacio de atributos)



# Estructura de la clase

- Repaso de Aprendizaje Supervisado
- Noción de espacio de atributos
- Árboles de decisión

# Árboles de decisión

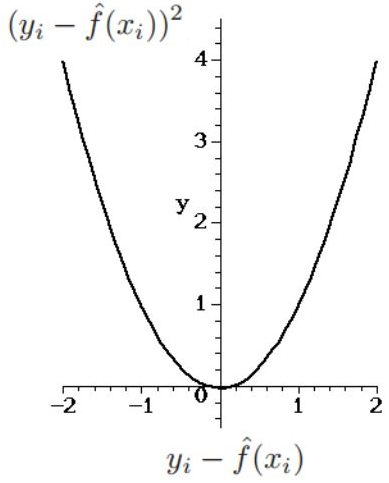
Un primer modelo de aprendizaje supervisado: árboles de decisión

Sirve tanto para atacar problemas de clasificación (binaria y multiclase) como de regresión

Primero veremos el problema de regresión (predecir una variable continua), luego el de clasificación

Definimos una medida de "calidad de la predicción" (función de costo / función objetivo) Sin pérdida de generalidad, la suma de los errores al cuadrado (SSE)

$$\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$



# Árboles de decisión

La construcción de un árbol estará guiada por minimizar el error/costo (SSE) sobre los datos de entrenamiento

Esta idea se aplica a muchos modelos de aprendizaje supervisado, por ej.:

- Regresión lineal
- Regresiones ridge y lasso
- Regresión logística
- Redes neuronales
- Support vector machines
- Gradient Boosting Machines
- XGBoost

# Árboles de decisión

Un árbol dado particiona el espacio de atributos en regiones. Las regiones vendrán definidas por si cumplen o no una serie de reglas

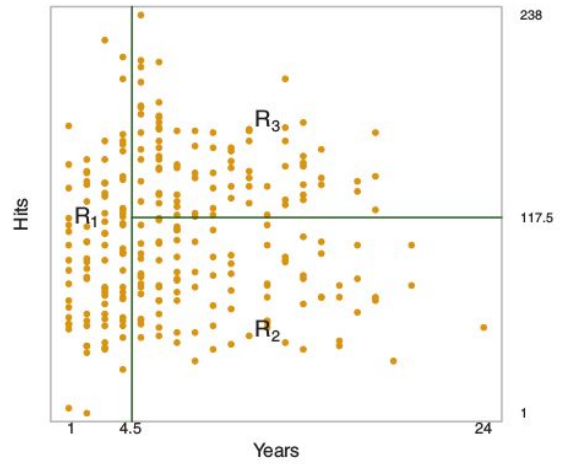
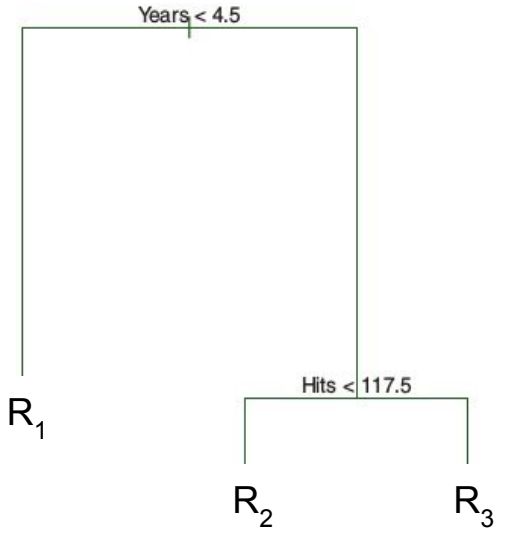



FIGURE 8.2. *The three-region partition for the* Hitters *data set from the regression tree illustrated in Figure 8.1.*




Un árbol tiene:

- Nodos internos: dividen el espacio de atributos
- Hojas: indican la región en la que se ubica una observación

¿En qué región caerá una observación con Years = 6 y Hits = 80?

# Árboles de decisión

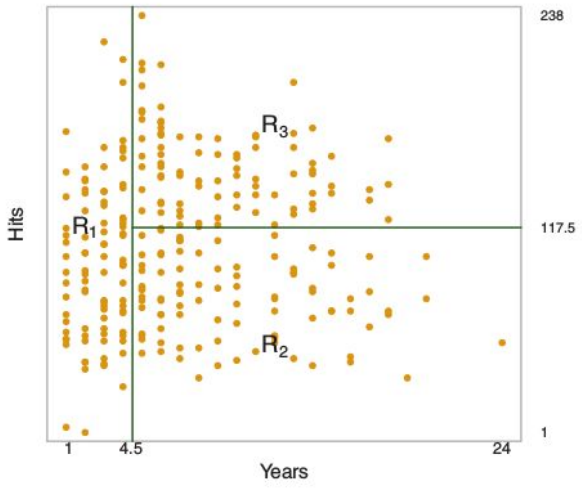
Primero veamos cómo usar una partición ya armada para predecir

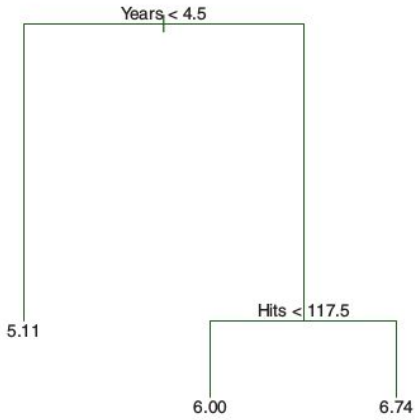
### Prediction via Stratification of the Feature Space

We now discuss the process of building a regression tree. Roughly speaking, there are two steps.

1. We divide the predictor space—that is, the set of possible values for $X_1, X_2, \ldots, X_p$—into $J$ distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$.

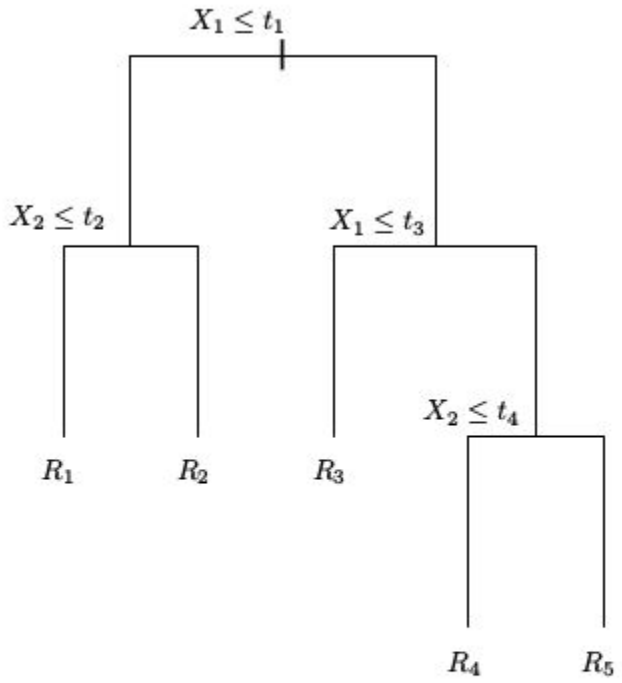
2. For every observation that falls into the region $R_j$, we make the same prediction, which is simply the mean of the response values for the training observations in $R_j$.

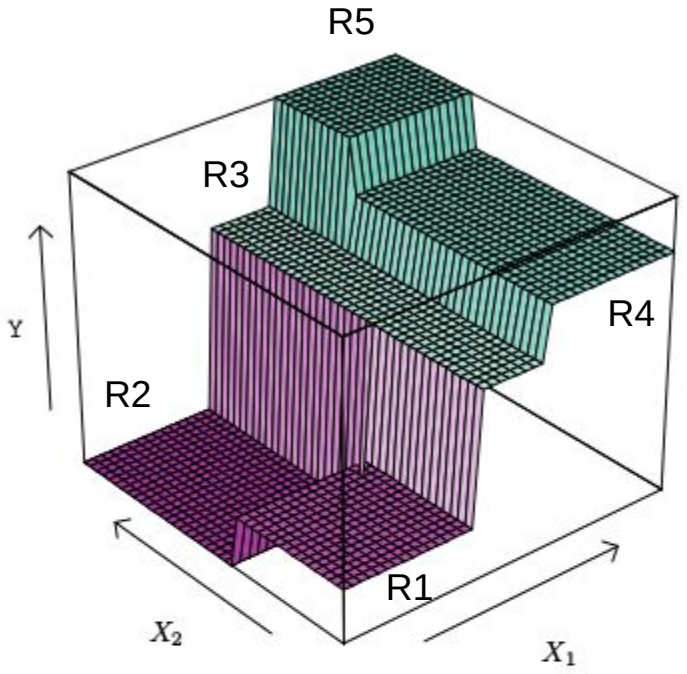


# Árboles de decisión

¿Pueden identificar en la figura de la derecha cada región?



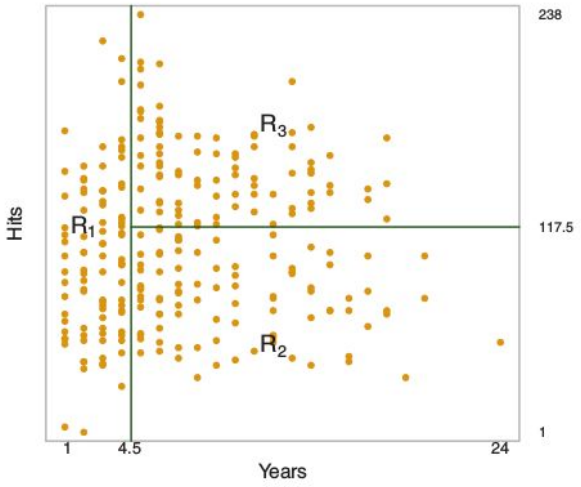

Las regiones son los "pisos"

# Árboles de decisión

El algoritmo de árboles de decisión divide el espacio de atributos en rectángulos "jerárquicos" paralelos a los ejes (este es el "sesgo inductivo" del algoritmo, más adelante en la materia ahondaremos sobre esto)

Dada una partición, uno tiene un error de predicción (SSE en entrenamiento), el mismo viene dado por:

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

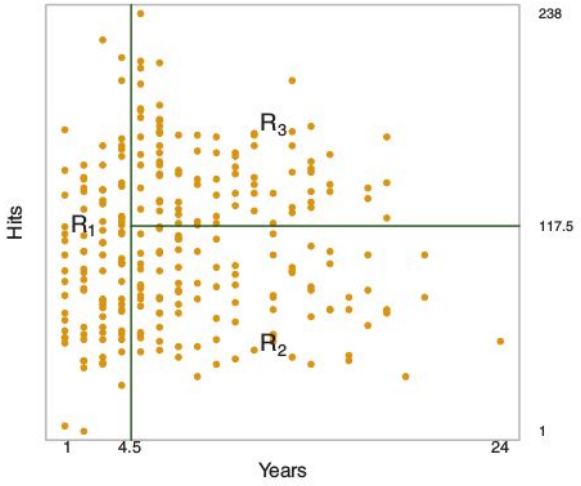



El objetivo será encontrar las particiones que minimicen este error (IMPORTANTE: no olviden que es el error de entrenamiento)

# Árboles de decisión

¿Cuántas posibles particiones hay? ¿Cómo podremos encontrar buenas particiones?

Vamos a seguir la estrategia de dividir recursivamente al problema en subproblemas más pequeños que conservan la misma estructura



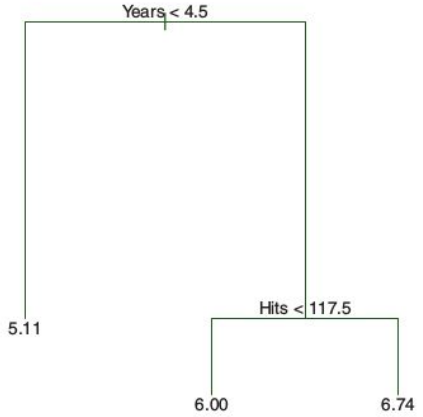




**FIGURE 8.2.** *The three-region partition for the* `Hitters` *data set from the regression tree illustrated in Figure 8.1.*

# Árboles de decisión

Seguiremos una estrategia de particiones recursivas (*top-down* y *greedy*).

Para cada variable *j* y cada punto de corte *s*, definimos las siguientes regiones:

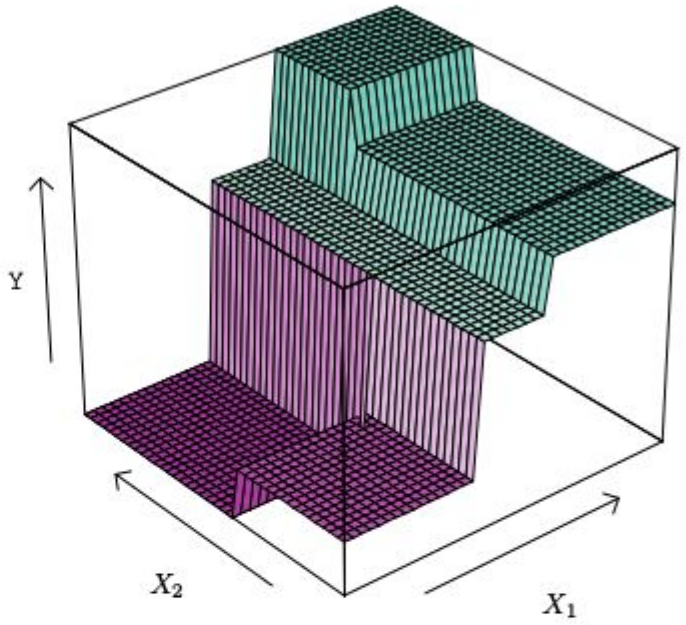
$$R_1(j,s) = \{X | X_j < s\} \text{ and } R_2(j,s) = \{X | X_j \geq s\}$$

Se elegirán como corte los valores de *j* y *s* que minimicen la siguiente expresión (a la reducción del error por un split se le llama ganancia)

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

Luego, se repite el proceso para cada subregión (paso recursivo) hasta cumplir algún criterio de parada (por ej., que en la región quede una única observación)

# Árboles de decisión




Propiedades interesantes de los árboles:

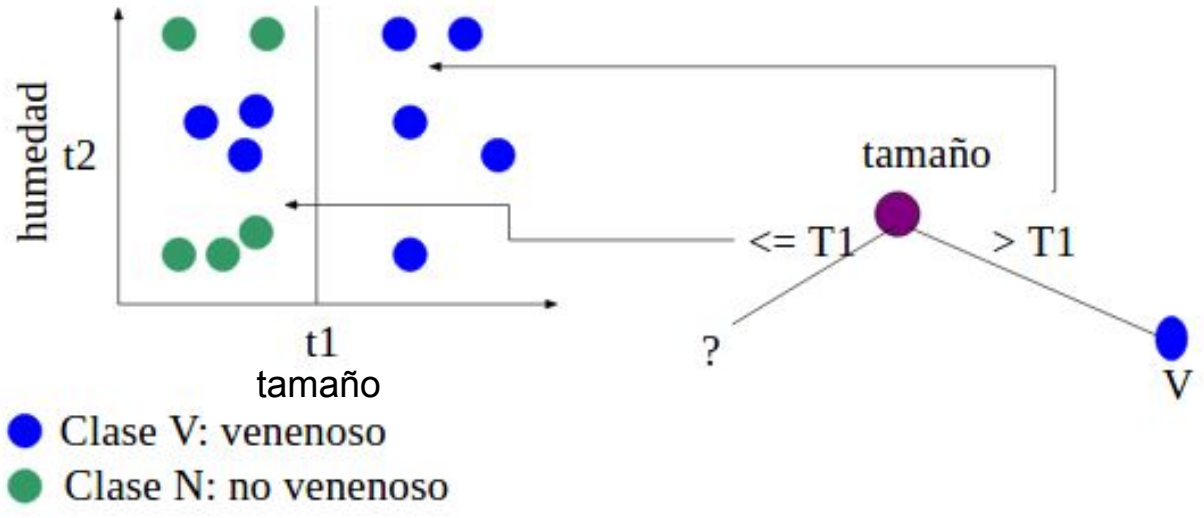
- Detectan interacciones entre los predictores
- Detectan no linealidades
- Teóricamente sus predicciones no se ven alteradas por transformaciones monótonas de los predictores de entrenamiento

# Árboles de decisión

La estrategia se adapta fácilmente a clasificación

Ahora vamos a predecir la probabilidad de que la observación *i* pertenezca a la clase *k* dado los valores de sus atributos *x*, es decir: $P( y_i = k \mid x_i)$

Estimaremos dicha probabilidad como la proporción de observaciones de la clase *k* dentro la región correspondiente a $x_i$

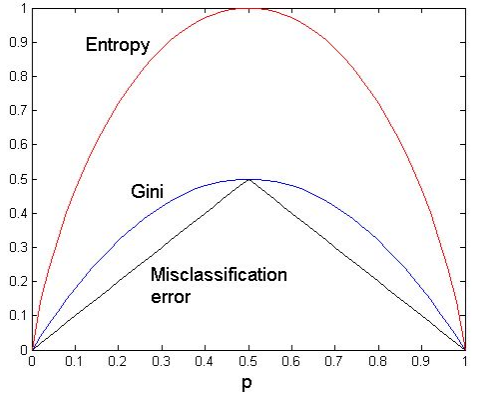


# Árboles de decisión

En clasificación, en vez de guiar la construcción de los árboles usando SSE, se utiliza medidas de impureza

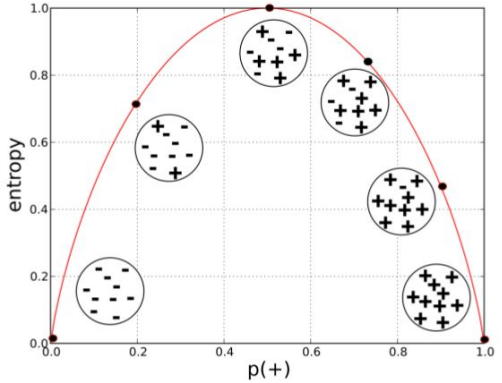
The *Gini index* is defined by

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




An alternative to the Gini index is *cross-entropy*, given by

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



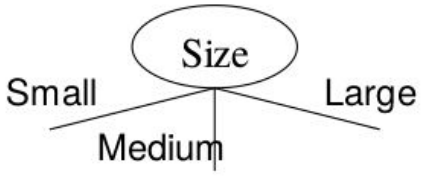

Dos aclaraciones:

1. Estas figuras solo aplican para el caso de $K = 2$
2. El costo del árbol se obtiene como la suma de la impureza de cada hoja, ponderando cada hoja por la fracción de observaciones que caen en la misma (también si se usa MSE)

# Árboles de decisión

Los árboles pueden trabajar de manera muy natural con variables categóricas

- Cortes múltiples:

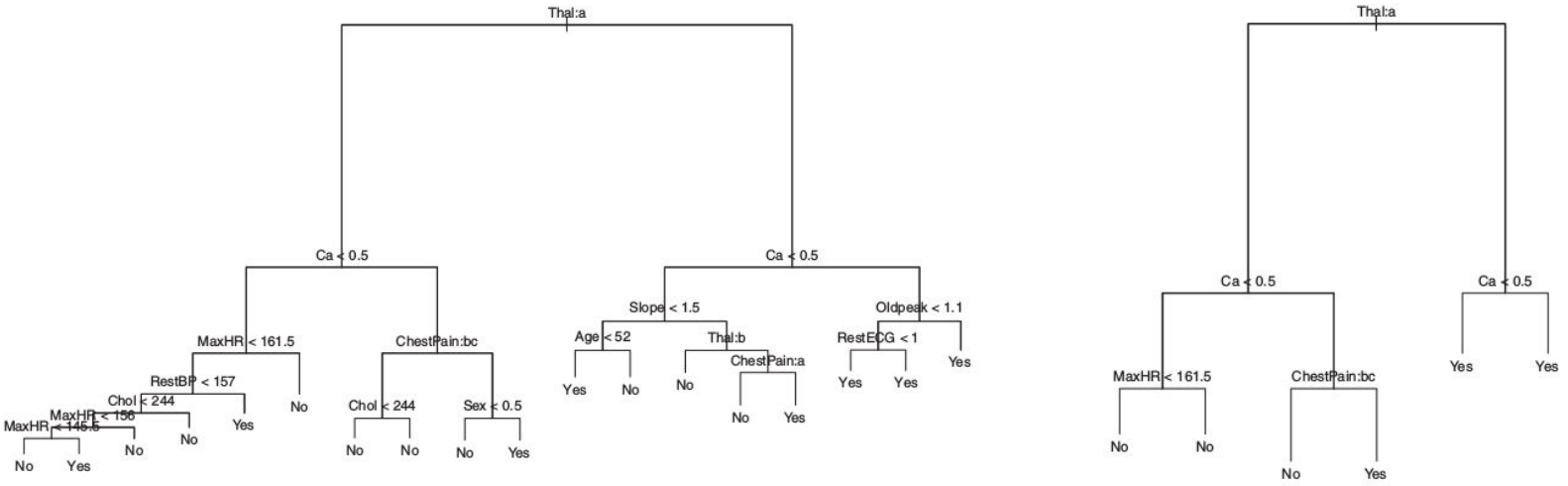



- Cortes binarios (lo más común):


No todas las librerías tienen esto implementado. Por ej. scikit-learn y XGBoost tienen implementado árboles que sólo trabajan con atributos numéricos (XGBoost incorporó el manejo de predictores categóricos en modo prueba no hace mucho)

# Árboles de decisión

¿Cuál de estos dos árboles ajusta más los datos?




Llamaremos flexibilidad del modelo a su capacidad de adaptarse a los datos de entrenamiento. Por ejemplo, en la comparación de los dos árboles, el de la izquierda es más flexible que el de la derecha

IMPORTANTE: No es cierto que un modelo que ajusta perfecto los datos de entrenamiento será el mejor al momento de predecir sobre datos nuevos (más sobre esto las próximas clases)

# Árboles de decisión

Hay muchos criterios para evitar que los árboles sean excesivamente profundos/flexibles. Por ejemplo:

- Definir una profundidad máxima (*max_depth* en sklearn / *maxdepth* en rpart)
- Definir un número mínimo de observaciones en un nodo de decisión para realizar un split (*min_samples_split* en sklearn / *minsplit* en rpart)
- Definir un número mínimo de observaciones que deben tener las hojas (*min_samples_leaf* en sklearn / *minbucket* en rpart)

¿Tiene sentido que min_samples_leaf >= min_samples_split? ¿Por qué?

# Árboles de decisión

Post-prunning:

Dado un valor de α, se hace crecer al árbol con profundidad máxima y luego se busca el sub-árbol que minimiza la siguiente expresión (ISLR, algoritmo 8.1).

$$\sum_{m=1}^{|T|} \sum_{i:\, x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

¿Cómo afecta α la flexibilidad del modelo?

Se van podando las hojas que no reducen el error en al menos α. Noten que α quita flexibilidad al modelo (*ccp_alpha* en sklearn / *cp* en rpart)

Pre-prunning (más amigable computacionalmente, ¿por qué?):

Sólo se acepta un split como válido si reduce el error en al menos α (*min_impurity_decrease* en sklearn)

# Árboles de decisión

*maxdepth*, *min_samples_split*, *min_samples_leaf* y *min_impurity_decrease* afectan a la flexibilidad del árbol resultante

Sus valores deben definirse antes de entrenar el modelo

A esto parámetros del algoritmo de aprendizaje, cuyo valor se debe definir antes de entrenar el modelo, se los conoce como hiperparámetros

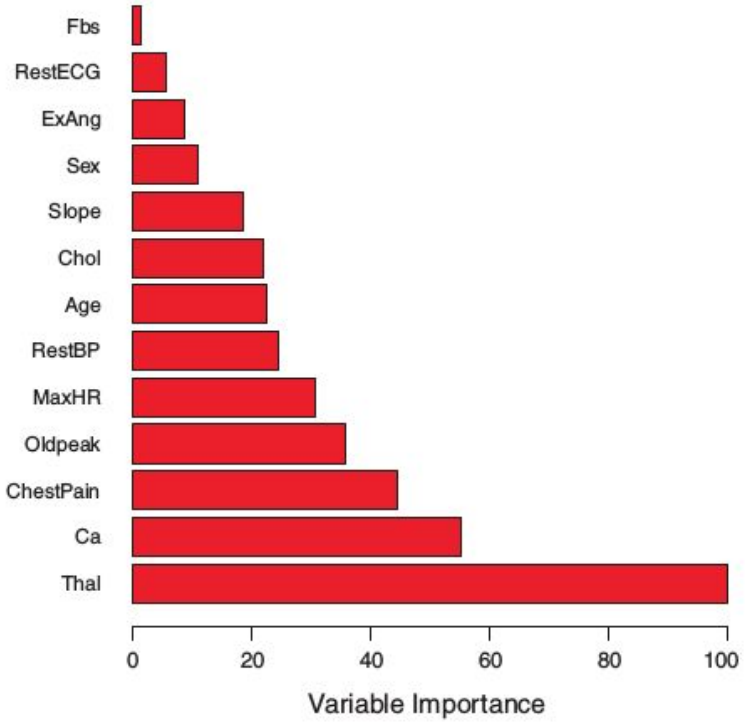
Prácticamente todos los algoritmos de aprendizaje automático tienen hiperparámetros

# Árboles de decisión

## Importancia de atributos

Estrategia típica en árboles:

1) Asignar una importancia igual a 0 a cada variable
2) Construir el árbol de decisiones utilizando los datos de entrenamiento
3) Para cada split del árbol construido:
   a) Identificar la ganancia (*g*) que se obtuvo al dividir el nodo mediante el atributo utilizado (*j*)
   b) Sumar *g* al valor de importancia de *j*
4) Reescalar todas las importancias respecto al máxima importancia (opcional)

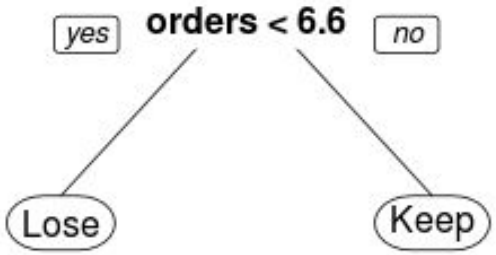


# Árboles de decisión

¿Qué ocurre si un predictor tiene valores faltantes?

**1. Durante la construcción del árbol.** Al calcular la ganancia de información para un posible split:

- Las observaciones con valores faltantes se ignoran
- Esto asegura que la selección del mejor split se base sólo en datos completos



# Árboles de decisión

**2. Al clasificar nuevas observaciones.** Cuando una observación tiene un valor faltante en la variable de split, se utilizan "surrogate splits" (particiones sustitutas):

**a) Creación de surrogate splits:**

- Se usa el conjunto de datos que llega al nodo actual
- Se construye un mini-árbol de profundidad 1 para cada variable predictora restante
- Cada mini-árbol intenta replicar la partición del split original

**b) Ranking de surrogate splits:**

- Se ordenan según su capacidad para imitar el split original
- El ranking se basa en la calidad de la predicción

**c) Clasificación de observaciones con valores faltantes:**

- Se usa el mejor surrogate split disponible (sin valor faltante) para esa observación
- La observación se dirige según la predicción de este surrogate split

**d) Regla de la mayoría:**

- La regla de la mayoría envía la observación a la rama con más datos en el nodo actual
- Los surrogate splits deben superar la performance de la "regla de la mayoría". Si no, se aplica la regla de la mayoría

# Árboles de decisión

Puntos a favor y en contra:

▲ Son muy fáciles de explicar (más que una regresión lineal)

▲ Algunos creen que los árboles de decisión reflejan más de cerca la toma de decisiones humanas que otros enfoques de predicción

▲ Pueden mostrarse gráficamente y son interpretados fácilmente incluso por personas inexpertas (especialmente si son pequeños).

▲ Manejo elegante de valores faltantes en los predictores

▲ Pueden manejar fácilmente predictores categóricos sin necesidad de alterar el conjunto de datos (no sklearn)

▼ Los árboles generalmente se ven superados en performance predictiva por otros algoritmos

▼ Pueden ser poco robustos. Un pequeño cambio en los datos de entrenamiento puede llevar a cambios drásticos en la estructura del árbol (dificultando su interpretabilidad)

# Bibliografía

- ISLP. Sección 8.1

- Hastie, Tibshirani & Friedman, "The Elements of Statistical Learning", Sección 9.2

Avanzada

- Therneau & Atkinson, "An Introduction to Recursive Partitioning Using the RPART Routines", Sección 5 (sobre datos faltantes)