

Otros algoritmos de regresión

1. Regresión polinomial.
2. Árbol de regresión.
3. Lasso.

Regresión polinomial

Regresión polinomial

La regresión polinómica sigue siendo lineal, ya que es lineal respecto a los parámetros del modelo.

$$y \sim \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_m x^m$$

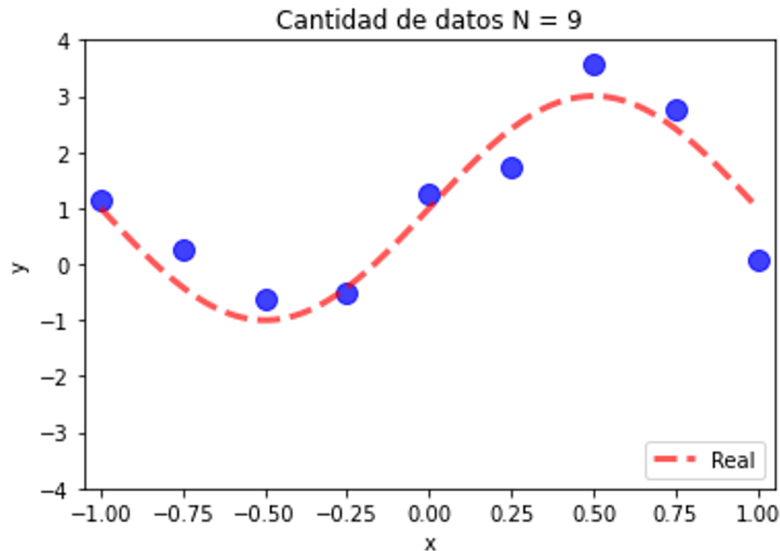
Los parámetros se encuentran minimizando la suma del cuadrado de los residuos al igual que antes:

$$RSS = \sum_i \left(y_i - \widehat{y}_i \right)^2$$

Regresión polinomial

Esta idea puede ser generalizada de la siguiente manera:

$$y \sim \beta_0 + \beta_1 \phi_1(x) + \beta_2 \phi_2(x) + \dots + \beta_m \phi_m(x)$$

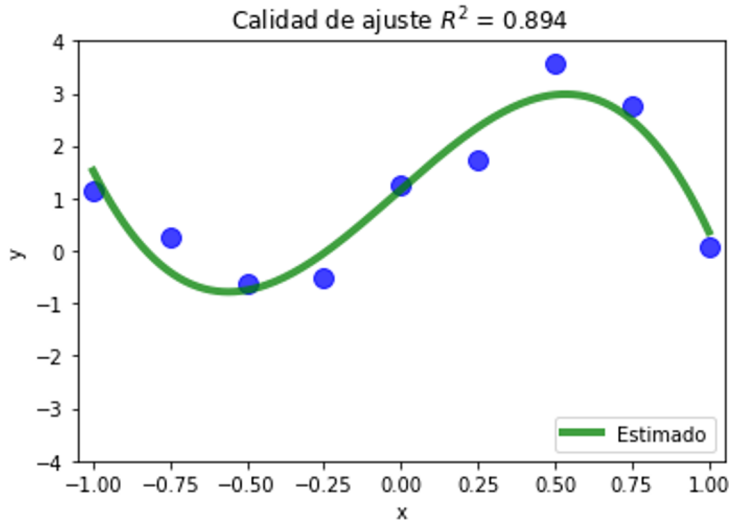


Las funciones pueden ser
exponenciales, senos, cosenos, etc.
(sin parámetros).

Regresión polinomial

Modelo de ejemplo:

Inspeccionando un poco los datos, se propone un polinomio de grado 3 (viendo por ejemplo que los datos presentan un mínimo y máximo, o un único punto de inflexión).



$$y \sim \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

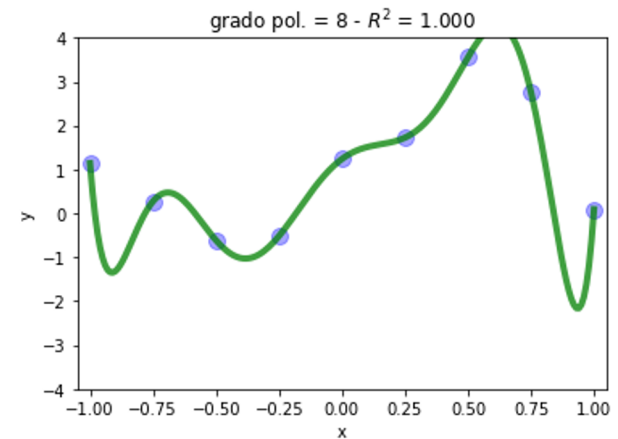
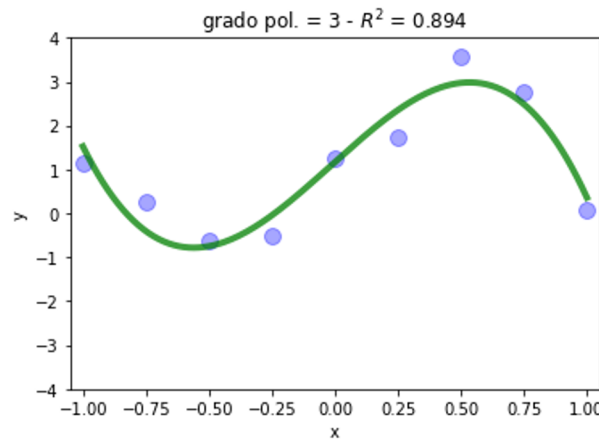
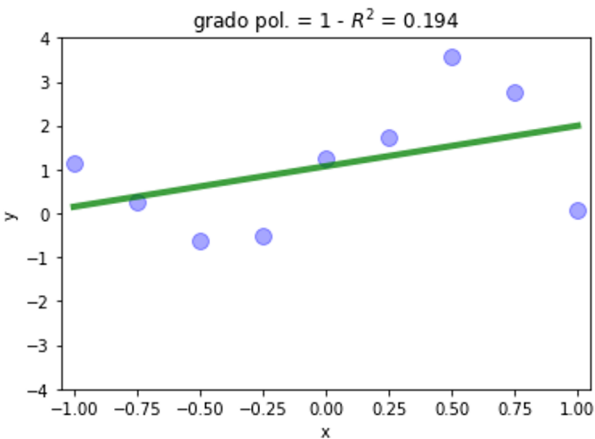
Coeficientes
estimados

Coefs.	d=3
0	1.17
1	5.16
2	-0.23
3	-5.74

Regresión polinomial

Pero, ¿cómo elegimos el grado del polinomio?

¿Qué pasa si probamos con polinomio de grado $m < N$ cualquiera?



Underfitting



Overfitting

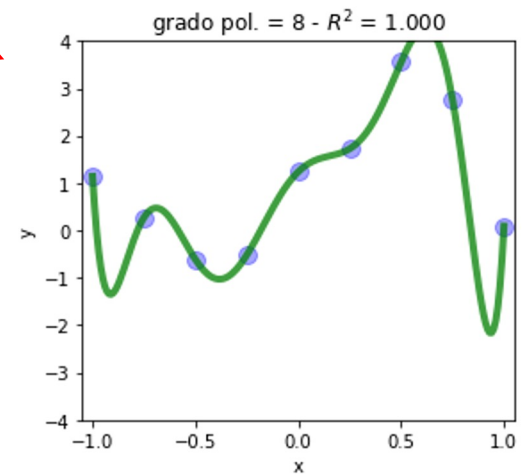
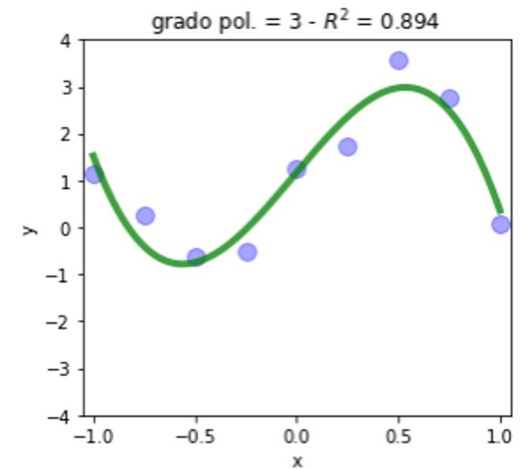
modelo más complejo

Regresión polinomial

Otras características del sobreajuste:

Coefs.	d=1	d=3	d=5	d=8
0	1.07	1.17	0.76	1.24
1	0.92	5.16	5.70	4.23
2		-0.23	3.28	-16.90
3		-5.74	-7.97	5.80
4			-3.41	121.64
5			1.74	-28.54
6				-233.17
7				17.99
8				127.80

Al sobreajustar los coeficientes suelen tomar valores muy altos.



Lasso

Lasso

Los métodos de regularización buscan mejorar el modelo.

- Al aplicar una restricción a los coeficientes del modelo, el modelo pierde “flexibilidad” (aumenta sesgo), pero disminuye la varianza.
- Es importante realizar una estandarización de los datos antes de aplicar métodos de regularización.
- Los métodos más conocidos son: Ridge y Lasso.
- Veremos Lasso por propiedades que tiene que lo hacen especial y en algunos casos muy útil.

Lasso

La idea de regularizar el polinomio es prevenir que los coeficientes no adopten valores absolutos muy altos, asociados a cambios bruscos en la curva ajustada.

$$RRS = \sum_i (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^M |\hat{\beta}_j|^q$$

Suma de los
errores al
cuadrado

Parámetro a sintonizar.
Otro ejemplo de
hiper-parámetro

Término de penalización:
se suele excluir al
parámetro β_0 .

Lasso

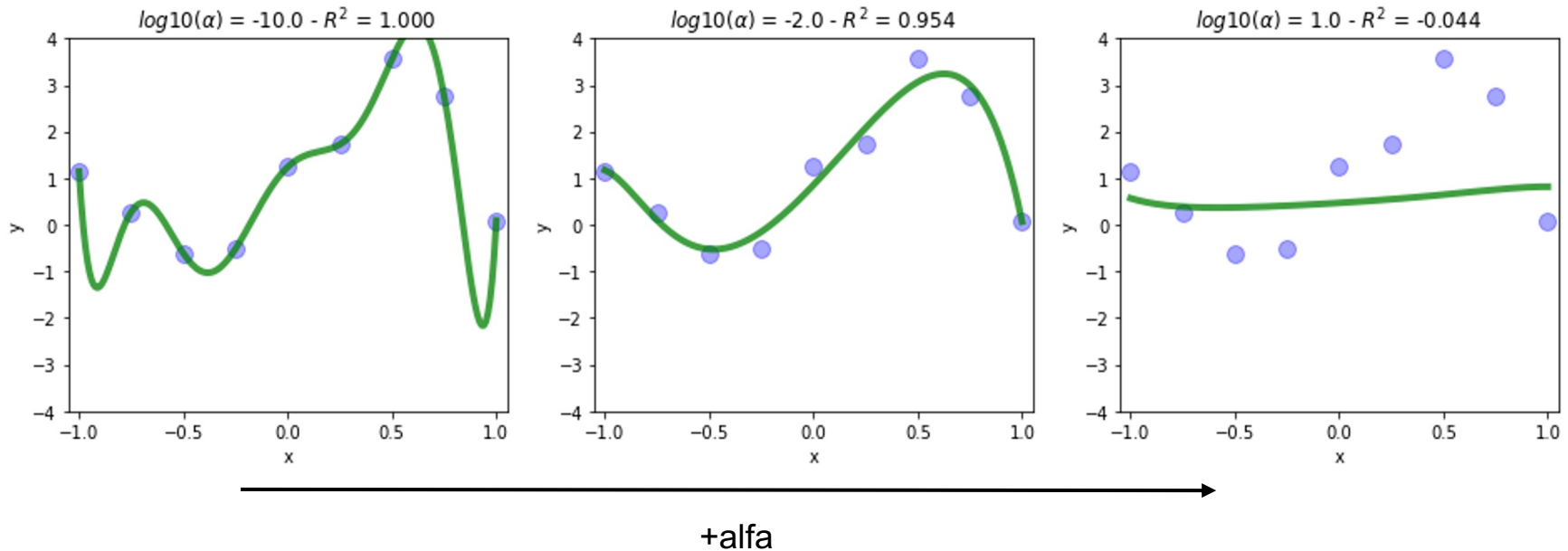
$$RRS = \sum_i \left(y_i - \hat{y}_i \right)^2 + \alpha \sum_{j=1}^M \left| \hat{\beta}_j \right|^q$$

La idea de regularizar el polinomio es prevenir que los coeficientes adopten valores absolutos muy altos, asociados a cambios bruscos en la curva ajustada.

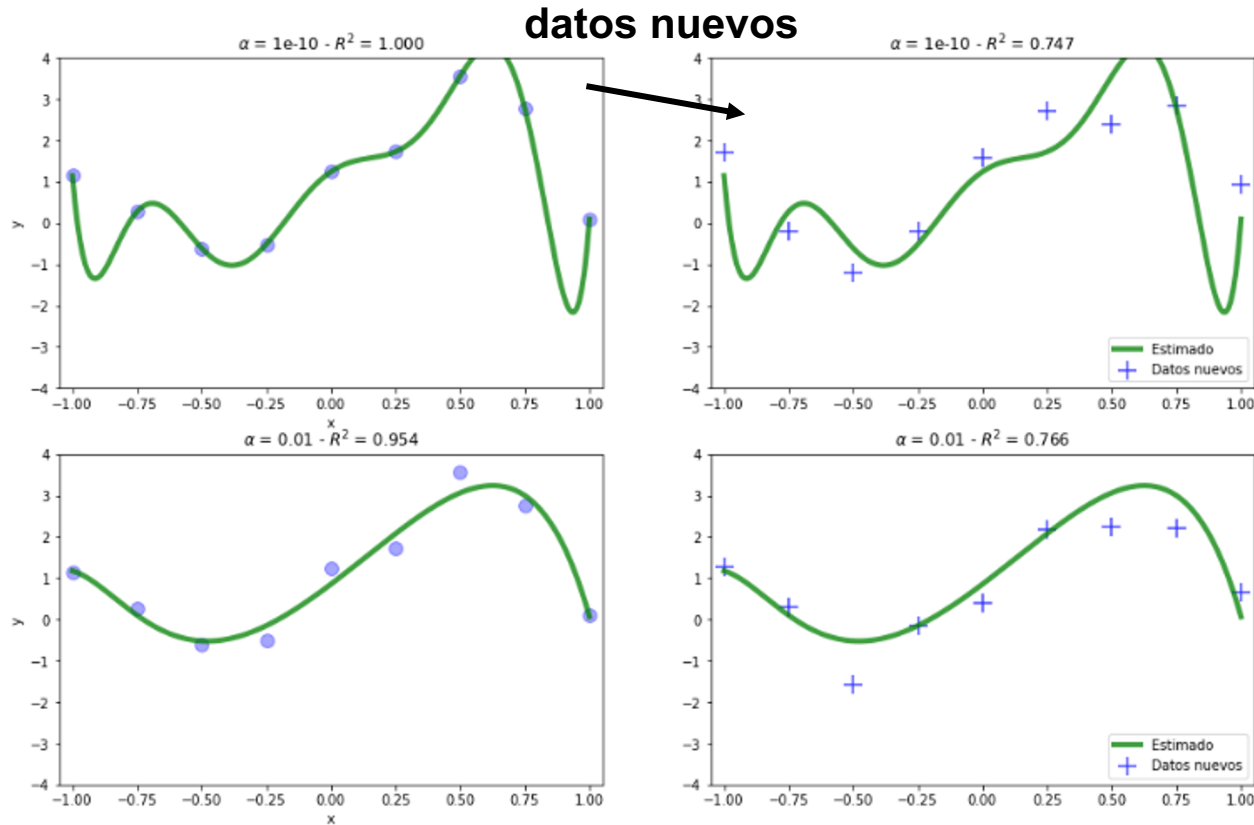
q = 1 (lasso regression): para valores de alfa altos, fuerza a que muchos coeficientes se vayan a 0, lo cual hace que el modelo se vuelva sparse (con muchos ceros). Ayuda a interpretar mejor modelo ya que actúa como un selector de las variables importantes (se queda con los términos dominantes y descarta los otros).

Lasso

Aumentar el término de penalización lleva a que el modelo sea más simple y previene el sobreajuste. Pero un valor muy alto, lleva a un polinomio de grado 0, que sabemos puede empezar a sub-ajustar.



Lasso



Con esto podemos incorporar un montón de parámetros al modelo (un grado alto en el polinomio) y el término de regularización va a ponderar solo los soportados por los datos (+ sesgo y - varianza).

Lasso

Lasso permite realizar selección de variables. Gran punto a favor!!

Si en un modelo de regresión lineal los coeficientes de los predictores tienen mucha diferencia entre sí, entonces seguramente lasso dará mejores resultados.

En scikit-learn, se puede aplicar Lasso con: **Sklearn.linear_model.Lasso**

Árboles de regresión

Árboles de regresión

» **CART (Classification and regression trees)**

- Utilidad en regresión y clasificación.
- Simples y útiles para la interpretación.

» Idea clave

- Segmentar o estratificar el espacio predictivo en un número de regiones.
- Para una nueva observación, su predicción depende de los atributos de la región a la que pertenece.

» **TDIDT (Top down induction of decision trees)**

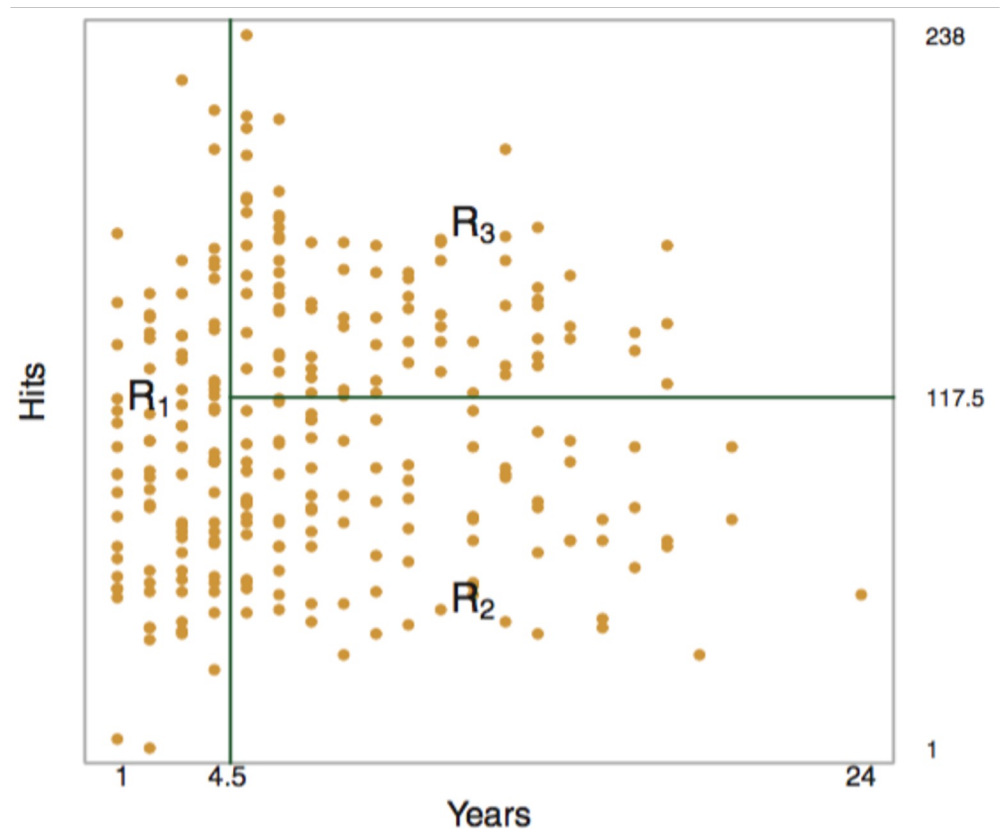
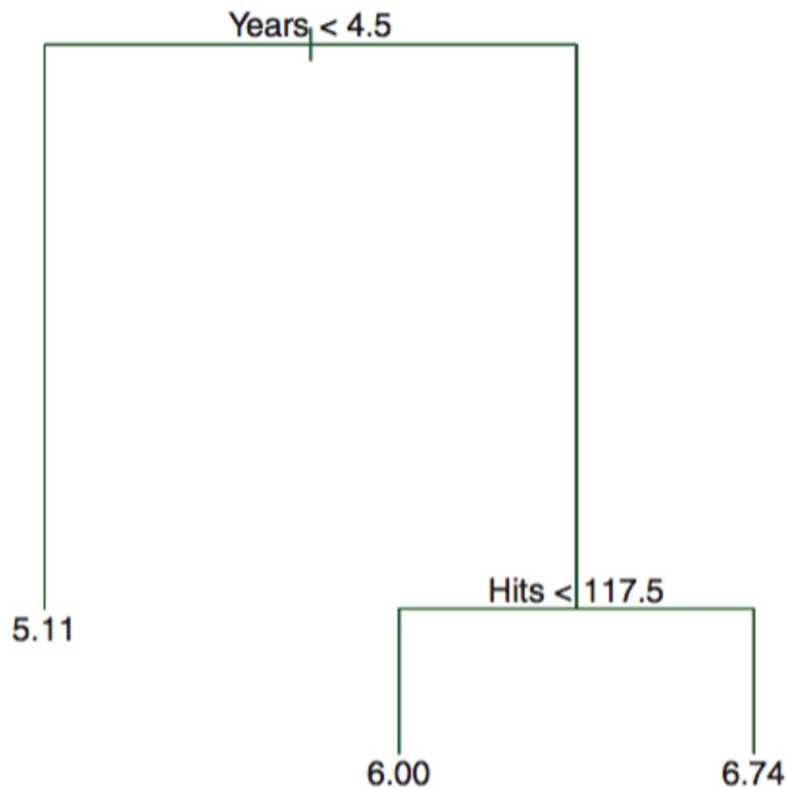
- Algoritmo básico de inducción de árboles.

» Idea del algoritmo TDIDT:

- Seleccionar la variable más informativa.
- Particionar el espacio acorde a un valor posible de esa variable.
- Construir el subárbol generado y unirlo con el árbol obtenido hasta ese momento.

Árboles de regresión

Los árboles de decisión pueden hacer varias tareas. En este curso nos enfocaremos en el problema de regresión.



Árboles de regresión

» Proceso de construcción general

- **Paso 1:** dividir el espacio predictor (de las características observables)
- **Paso 2:** Para cada observación que cae en la región R_j
 - Se realiza la misma predicción
 - **Usualmente es la media de los valores obtenidos por las observaciones del conjunto de entrenamiento en R_j**

» Paso 1: dividir el espacio predictor

- Se utilizan regiones rectangulares de alta dimensión
 - Se generan al dividir el espacio mediante hiperplanos
- Meta: Minimizar *suma de residuos cuadrática* (RSS: *residual sum of squares*)

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

- Resulta computacionalmente imposible considerar todas las posibles particiones de datos en el espacio en J regiones

Árboles de regresión

Paso 1: dividir el espacio predictor

Se elije una característica (atributo, o predictor) X_j y un punto de corte s tal que el espacio característico se divida en regiones

$$\{X|X_j < s\} \quad \text{y} \quad \{X|X_j \geq s\}$$

las cuales resultan en la mayor reducción de RSS.

Existen varias forma de hacer esto, pero una de ellas es:

Método de *partición binaria recursiva*: se buscan 2 regiones:

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

junto a valores j y s tales que minimicen la ecuación

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

donde \hat{y}_{R_1} corresponde a la predicción realizada en la región 1 (es mínima cuando \hat{y}_{R_1} es la media de los elementos de entrenamiento en la región).

Árboles de regresión

- » Este proceso permite ir generando más y más ramas del árbol.
- » Esto hace crecer mucho el árbol, por lo que se necesita un método de regule el crecimiento del árbol.
- » Existen métodos, como por ejemplo métodos de Poda.
- » Definir previamente los **hiper-parámetros** del algoritmo también sirve para controlar este fenómeno.



Son parámetros que se definen previo al entrenamiento del algoritmo

Entonces, los **parámetros del algoritmo** son los cuales se estiman durante el proceso del entrenamiento del algoritmo.

Árboles de regresión

En scikit-learn es posible usar árboles para regresión con:

sklearn.tree.DecisionTreeRegressor

Los árboles son una técnica de regresión no lineal.

Como se muestra en la figura, el parámetro `max_depth`, que regula la profundidad del árbol, permite la “curva” de regresión más o menos escalonada.

Es escalonada debido a que la estimación la realiza según el promedio de los datos que están en esa región.

