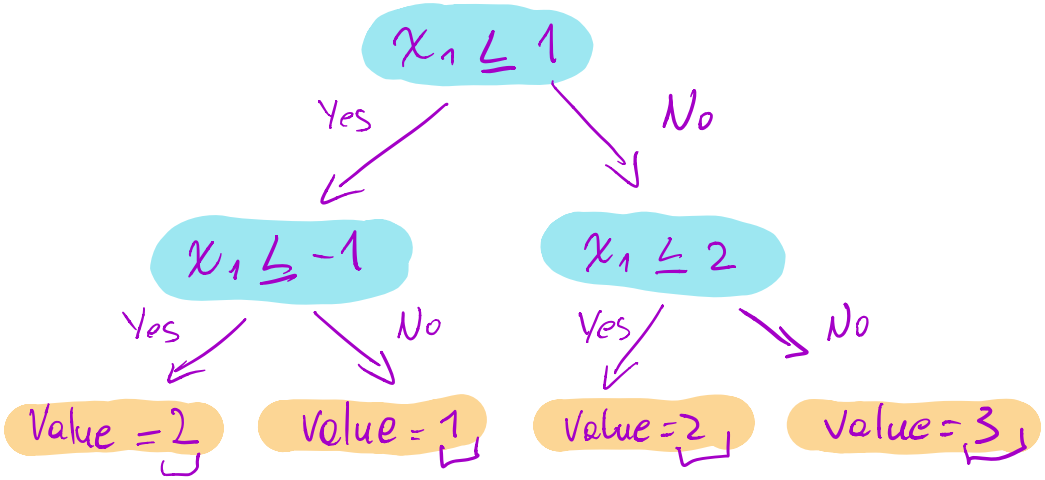
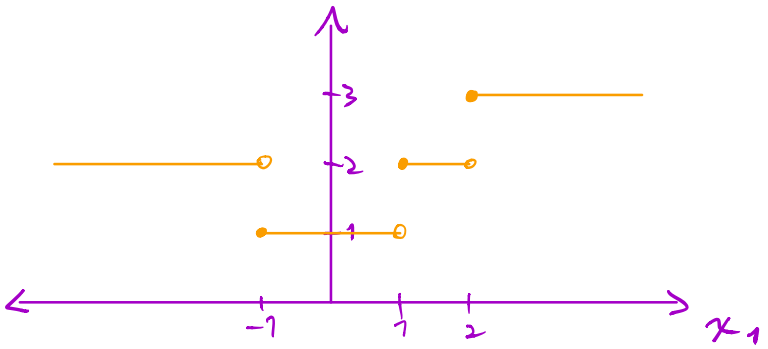
Árboles de regresión

La idea de usar regresión con un árbol es similar a lo que hacíamos antes, pero en vez de retornar una clase retornamos un valor numérico.

Por ejemplo, el siguiente árbol de decisión en base a x_1 se ve de la siguiente forma:



Represente la siguiente función:

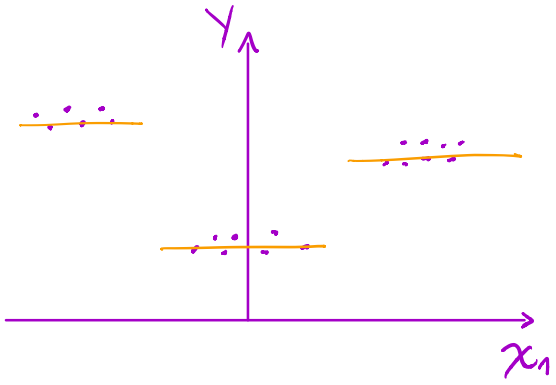


Ahora bien, ¿cómo entrenamos un árbol para regresión? El algoritmo es el mismo, pero en vez de impureza de Gini vamos a usar el MSE:

$$J(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right}$$

Donde $MSE_{node} = \sum_{i \in node} (\underbrace{\bar{y}_{node}}_{\text{Promedio de los elementos en el nodo}} - y_i)^2$

Entonces, según nuestros datos buscamos hacer un "split" que minimice el MSE:



Aquí hay una división que minimiza el MSE y cada parte retorna el promedio

Ojo! Como vimos en el árbol de ejemplo, la feature x_1 aparece en más de un nivel. Esto pasa con las features no binarias en Clasificación también

¿Y para más de una variable?

Podemos ir buscando el par (k, t_k), con k feature y t_k threshold, de forma greedy al igual que antes.

Para evitar el overfitting vamos a:

1) Limitar la profundidad del árbol
2) Definir el mínimo número de elementos en un nodo para hacer split.

Para entender esto mejor lo vamos a ver en código.