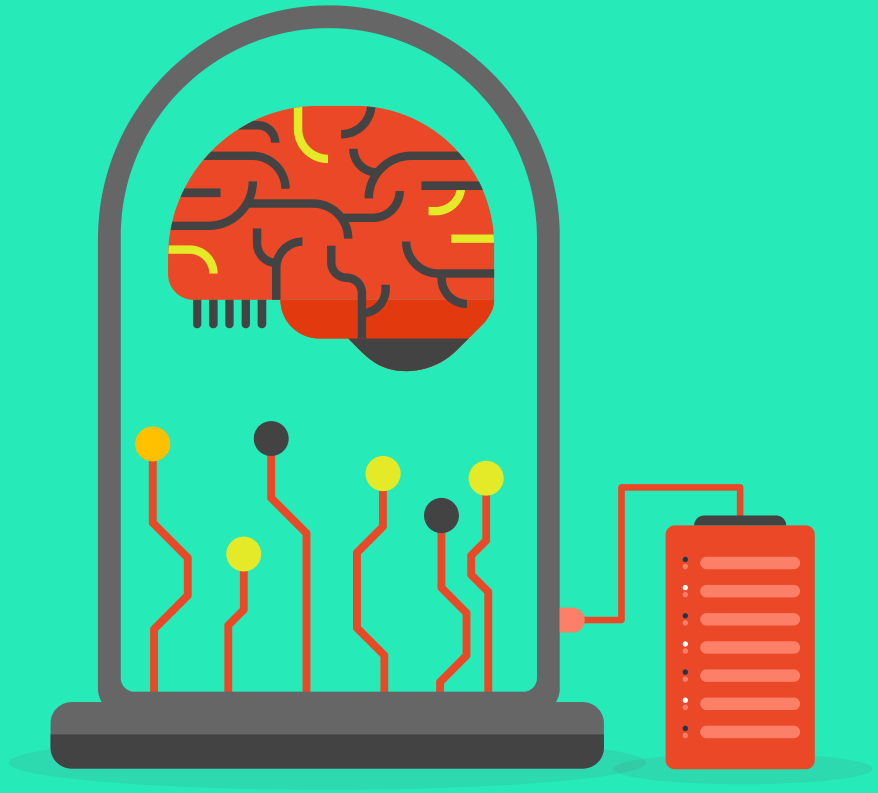
8. Regresión con Árboles de Decisión



Árboles de Decisión

Los **Árboles de Decisión** son uno de los algoritmos de Machine Learning más versátiles que existen, capaces de resolver problemas tanto de clasificación como regresión, y además pueden ajustar datos muy complejos o con múltiples salidas. Por otro lado, representan los bloques fundamentales de los **Bosques Aleatorios**, los cuales están entre los algoritmos de Machine Learning más poderosos existentes.

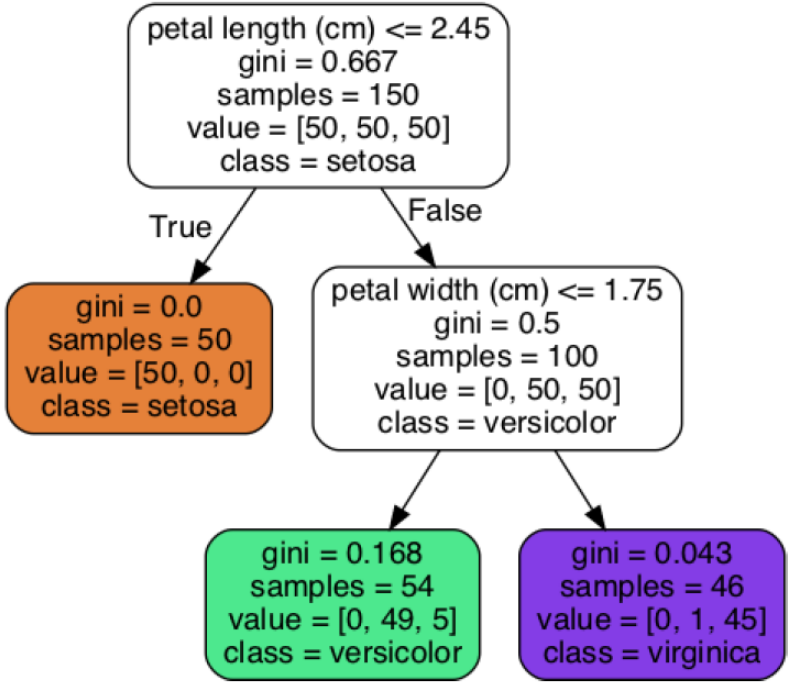
A fin de entender de manera más clara el cómo funciona un Árbol de Decisión, veamos el ejemplo de un modelo ya entrenado aplicado, en primer lugar, a un problema de Clasificación. Por ejemplo, supongamos que entrenamos uno para clasificar los datos del conocido dataset ***Iris***, el cual contiene 50 muestras para 3 especies distintas de flores Iris: setosa, versicolor y virginica, y cuyas variables son la longitud y ancho del sépalo, y longitud y ancho del pétalo.

Árboles de Decisión

Largo Sépalo	Ancho Sépalo	Largo Pétalo	Ancho Pétalo	Especie
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
6.5	3.0	5.2	2.0	virginica
6.2	3.4	5.4	2.3	virginica
5.9	3.0	5.1	1.8	virginica

Árboles de Decisión

Al completar el entrenamiento con los 150 datos totales de dicho dataset, podemos obtener un Árbol de Decisión como el siguiente:



Algoritmo CART

La librería *Scikit-Learn* usa el algoritmo CART (Classification and Regression Tree) para entrenar los Árboles de Decisión. El algoritmo primero subdivide el conjunto de entrenamiento en dos conjuntos usando una única variable k y un umbral t_k. El algoritmo escoge el par de valores (k, t_k) que producen los subconjuntos más puros (gini), minimizando la función de costo:

$$J(k, t_k) = \frac{m_l}{m} G_l + \frac{m_r}{m} G_r$$

$G_{l/r}$ *mide la impureza de los subconjuntos de la izquierda/derecha*

$m_{l/r}$ *es el numero de instancias en los subconjutos de la izquierda/derecha*

Una vez se subdivide este conjunto, se repite de manera recursiva la misma lógica para cada nuevo subconjunto hasta alcanzar una profundidad preestablecida para cada nodo (hiperparámetro) o bien hasta que la pureza ya no disminuye.

Regularización en CART

En el caso de los Árboles de Decisión, es posible regularizar un modelo (evitar sobre entrenamiento) al restringir la libertad que tiene el algoritmo para ajustar los datos de entrenamiento. Un parámetro básico de regularización es limitar la *profundidad* del árbol, es decir, la cantidad de niveles de nodos que el mismo puede crear. Otros pueden ser la cantidad mínima de registros que debe tener un nodo para dividirse, la cantidad mínima de registros que un nodo hoja puede tener, o al número máximo de nodos hoja que el Árbol admite. Todos estos son parámetros ajustables en la librería Scikit-Learn.

El valor de dichos parámetros dependerá mucho de los datos y de la experimentación/ajuste a partir de los resultados obtenidos con cada problema.

Regresión con Árboles de Decisión

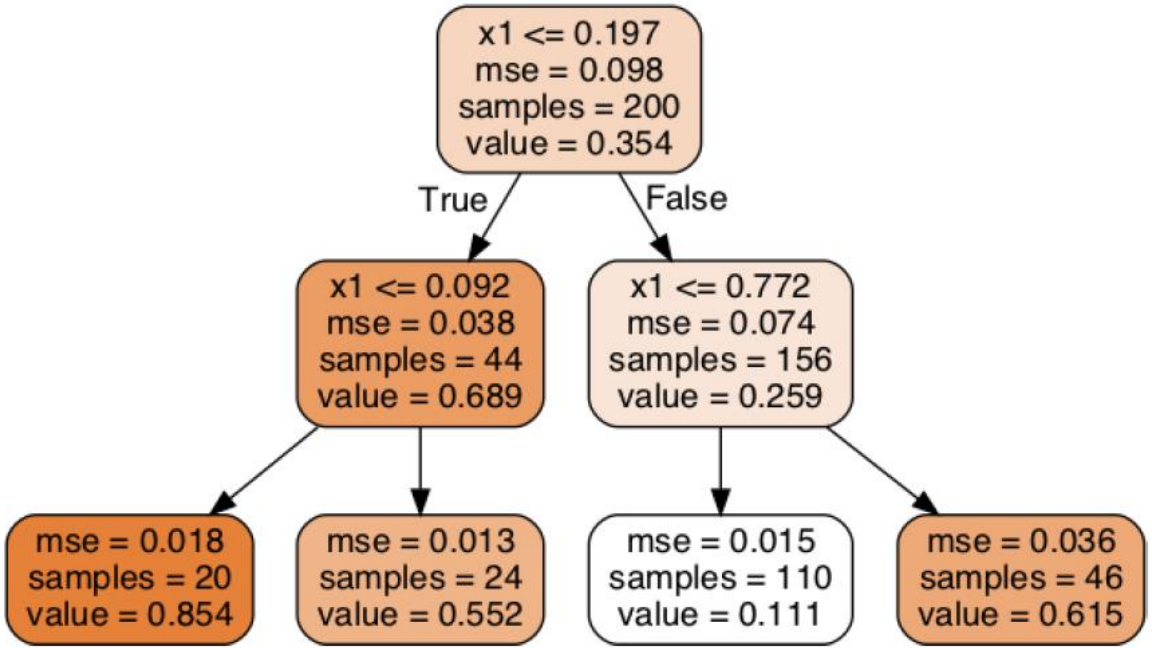
De manera similar, es posible llevar a cabo regresión con los Árboles de Decisión. En este caso, las hojas del árbol darán lugar a predicciones de valores numéricos y no clases, mientras que el objetivo del entrenamiento será minimizar la función de costo que dependerá, ahora, del MSE:

$$J(k, t_k) = \frac{m_l}{m} MSE_l + \frac{m_r}{m} MSE_r$$

La manera como se recorre el árbol para generar predicciones de nuevos datos será idéntica al caso de clasificación, hasta alcanzar un nodo hoja en donde se obtiene el valor final de la regresión.

Regresión con Árboles de Decisión

Por ejemplo, al entrenar datos de una función cuadrática con ruido, podemos obtener un resultado como el de la figura:



Noveno Notebook Práctico

Como veremos en la implementación práctica, los Árboles de Decisión son modelos de Machine Learning que tienen la cualidad de ser intuitivos y útiles cuando queremos explicar su forma de predecir datos. Sin embargo, son propensos al sobre entrenamiento, y sus ajustes, por la cualidad que tienen de dividir de manera iterativa los conjuntos de datos, producen bordes de decisión irregulares.

Veamos los ejemplos.

Noveno Notebook Práctico