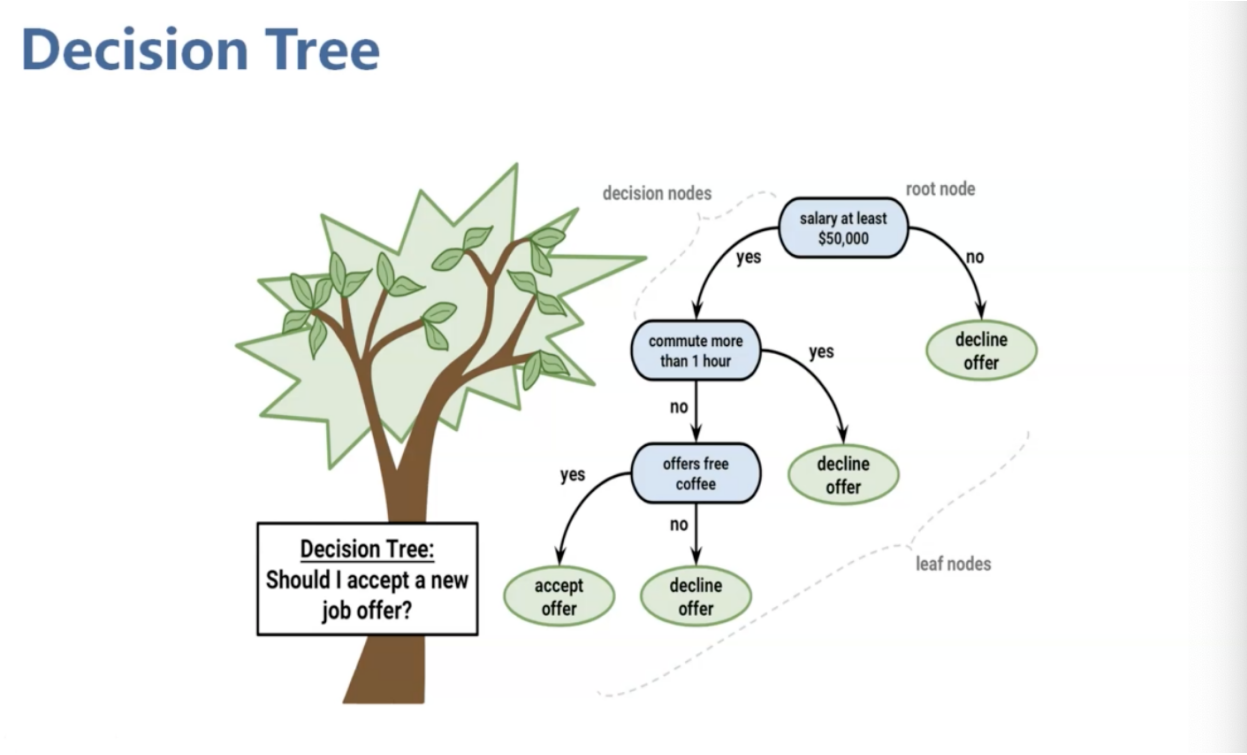
Árboles de decisión e implementación



El árbol de decisión es un tipo de modelo predictivo que es usado en cada iteración de xgboost, puede ser usado para clasificación o regresión.

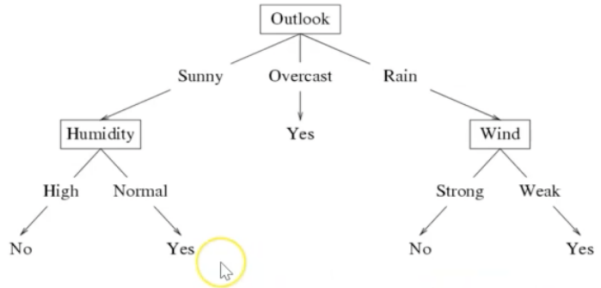
Cada nodo del arbol representa un feature, y las últimas hojas de este representan una predicción.

En si, un árbol de decisión consta de una serie de nodos y hojas.

Decision Tree

- a leaf node - indicates the value of the target attribute (class) of examples, or
- a decision node - specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.

- **Internal nodes** test the value of particular features x_j and branch according to the results of the test.
- **Leaf nodes** specify the class $h(\mathbf{x})$.



Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

Marco de trabajo

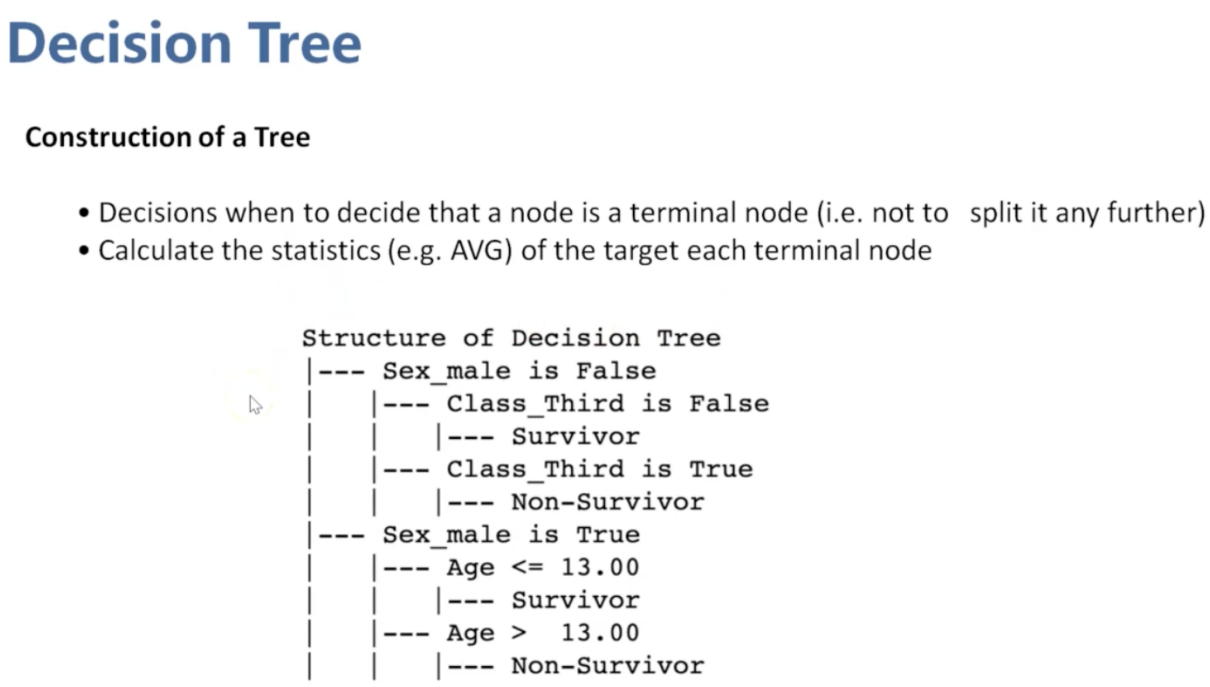
Decision Tree

- Attribute-value: Continuous or categorical variables used to split branch
- Target: Continuous or categorical variables whose values are calculated at all leaves

Atributo: Continuo o categorica variable que es usada para dividir los nodos en ramas.

Target: Continuo o variable categorica cuyos valores son calculados en las hojas.

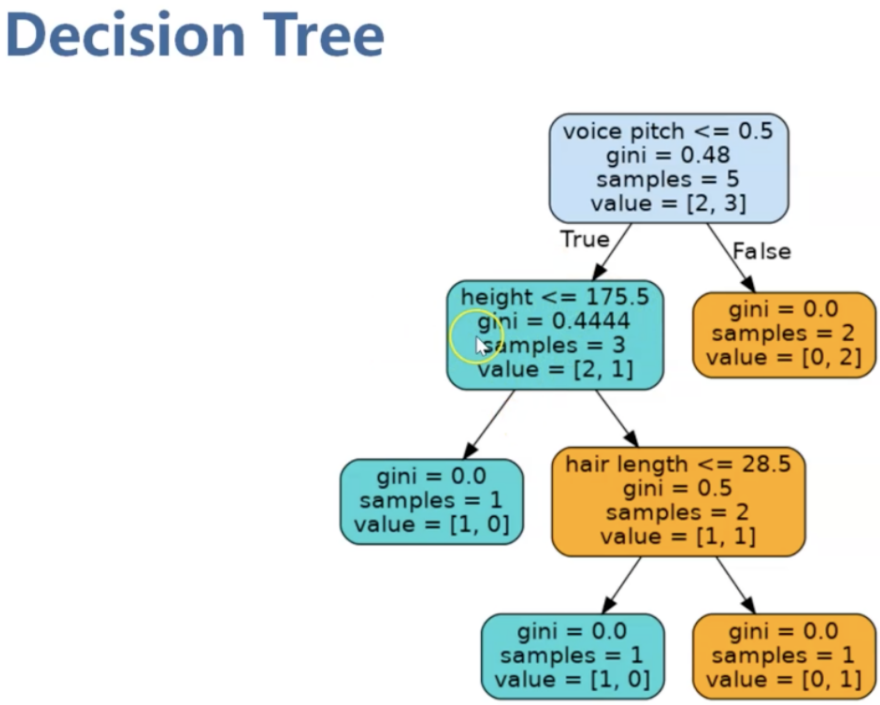
Construcción de un árbol



¿Cómo separar?

- Necesidad de un medición de impuridad de un nodo para decidir como separarlo, o saber cuál nodo separar
- La medición debe ser como máximo cuando un nodo es igualmente dividido entre todas las clases,
- La impuridad debe ser 0 (cero) si el nodo todo una clase.

La impuridad se le llama *gini.* y la manera intuitiva de interpretarlo es mientras menor sea el valor de gini, más puro será el nodo.



Gini index

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

donde,

$p(j|t)$ es la frecuencia relativa de la clase j en el nodo t.

El valor del index Gini varía desde 0 (más puro) hasta $(m-1)/m$ (todas las clases igualmente representadas).

Entropía

La segunda medida de impuridad es la entropía, y es definida como

$$Entropy(t) = -\sum_j p(j|t) log\ p(j|t)$$

donde,

$p(j|t)$ es la frecuencia relativa de la clase j en el nodo t.

La entropía medida es maximizada a $p(j|t) = 0.5.$

Recomendación sobre cual medida usar

Los resultados de ambas son relativamente parecidos, se recomienda probar con ambas y luego cruzarlas con un cross validation.

Reglas gini de separación en el modelado de árboles

Del siguiente subset de datos

Clase 1	0
Clase 2	6

Se tiene que la probabilidad o distribución de cada clase está dado por $P(C1) = 0/6 = 0$; $P(C2) = 6/6 = 1$, por lo que si calculamos su correspondiente index Gini, nos daría

$$\begin{gathered} GINI(t) = 1 - \sum_j [p(j|t)]^2 \\ = 1 - P(C1)^2 - P(C2)^2 \\ = 1 - 0 - 1 \\ = 0 \end{gathered}$$

Dandonos un impureza de 0, por lo que la separación de clases del arbol sería "perfecta”, sin embargo, si tuvieramos el siguiente subset de datos:

C1	1
C2	2

el calculo cambiaría a

$$
\begin{aligned}
GINI(t) &= 1 - \sum_j [p(j|t)]^2 \\
&= 1 - P(C1)^2 - P(C2)^2 \\
&= 1 - (\frac{1}{6})^2 - (\frac{5}{6})^2 \\
&= 0.278
\end{aligned}
$$

Finalmente, tenemos el tercer caso escenario

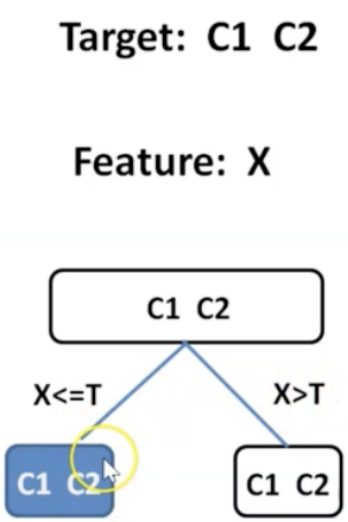
C1	2
C2	4

Dando un índice Gini:

$$
\begin{aligned}
GINI(t) &= 1 - \sum_j [p(j|t)]^2 \\
&= 1 - P(C1)^2 - P(C2)^2 \\
&= 1 - (\frac{2}{6})^2 - (\frac{4}{6})^2 \\
&= 0.444
\end{aligned}
$$

;

Asumimos que tenemos un problema de clasificación binaria, entre la clase C1 y C2, y por simplicidad, también asumiremos que tenemos un feature X y que se encontró un treshold T para separar el nodo en hojas.



En el caso del primer subset de datos, dado que el index gini es totalmente puro, asumimos que es una muy buena separación ya que podrá clasificar la predicción de manera certera

En el segundo caso, sigue teniendo un índice Gini bastante bajo pero ya no es 0, de todas formas es un indice de calidad para poder separar el nodo.

Pero para el tercr caso, no es una muy buena separación ya que el índice Gini en esa evaluación en particular no da un separador muy claro.

Obtener el índice Gini a través de los distintos nodos

Cuando un nodo p es separado en k particiones (hijos), la calidad de la separación conjunta es calculada como:

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

donde,

$n_i =$ número de registros de un hijos i,

$n =$ número de registros en un nodo p

Ejemplo

Suponiendo que tenemos una hoja que viene de la partición de un nodo que a su vez viene de la partición de otro y queremos calcular el index Gini total de la hoja,

$$\begin{aligned} Gini(N1) &= 1 - (\frac{5}{7})^2 - (\frac{2}{7})^2 \\ &= 0.194 \\ Gini(N2) &= 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 \\ &= 0.528 \end{aligned}$$

Calculando el index gini de la hoja:

$$\begin{aligned} GINI_{split} &= \sum_{i=1}^{k} \frac{n_i}{n} GINI(i) \\ &= \frac{7}{12} * 0.194 + \frac{5}{12} * 0.528 \\ &= 0.333 \end{aligned}$$

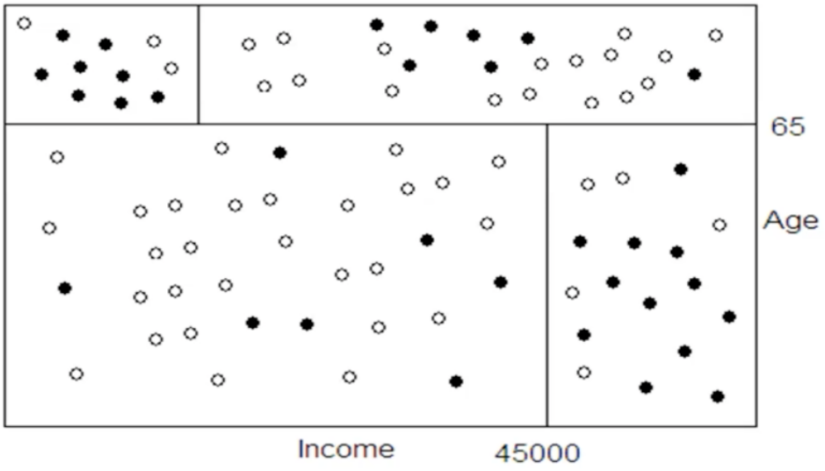
El index Gini de la hoja se interpreta como la calidad de las separaciones secuenciales que llevaron a esa hoja.

Es el calculado anterior un buen índice? hay que evaluar.

Entendiendo los árboles de decisión a través de hiperplanos dos dimensionales

Decision Tree—hyperplane plot

Classification tree rectangle: the splits are based on how much they impurity in the resulting rectangle. A pure rectangle is the one that is composed of a single class. The reduction in impurity is defined as overall impurity before the split minus the sum of the impurities for the two rectangle that result of a split

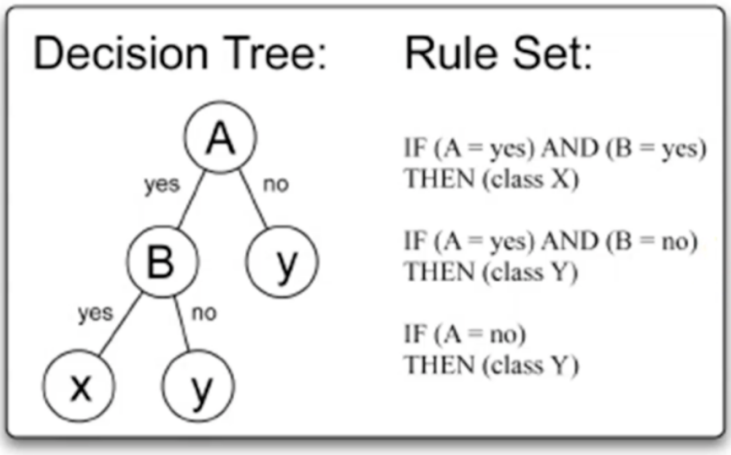


En el ejemplo anterior, se dan como features edad e ingresos para predecir 2 clases, expresados por el encode de color (blanco o negros), es un arbol binario.

Este rectangulo completo puede ser visto como la raíz del arbol, y cada división, una separación de nodos que luego se van separando más aún.

En cada nodo la idea es que la frecuencia de una clase u otra sea mayormente marcada, esto es determinado por el índice Gini.

Árbol expresado en reglas


Decision Tree:
Rule Set:
A
yes
no
B
y
yes
no
x
y
IF (A = yes) AND (B = yes)
THEN (class X)
IF (A = yes) AND (B = no)
THEN (class Y)
IF (A = no)
THEN (class Y)