Arboles de Decisión

Jonnatan Arias Garcia – jonnatan.arias@utp.edu.co

David Cardenas peña - dcardenasp@utp.edu.co

Hernán Felipe Garcia - hernanf.garcia@udea.edu.co

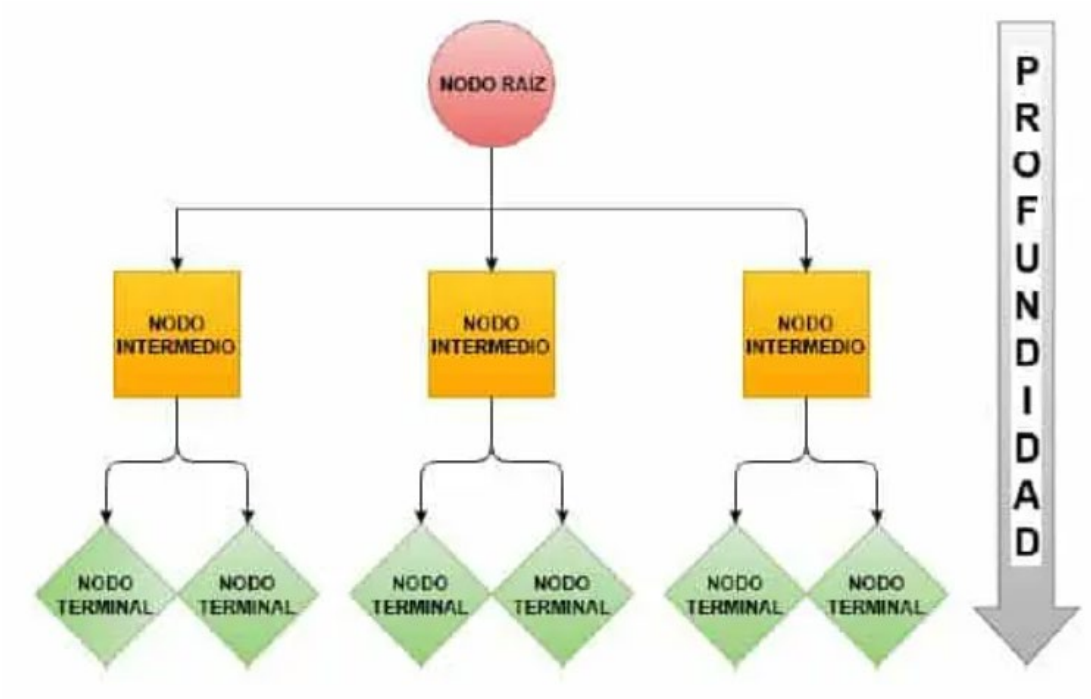
Contenido

- Introducción
- Tipos de Arboles
- Regresión Vs. Clasificación
- Como crear el árbol
  - Splitter
  - Métrica
  - Regularización
- Ventajas Vs. Desventajas

Arboles de Decisión

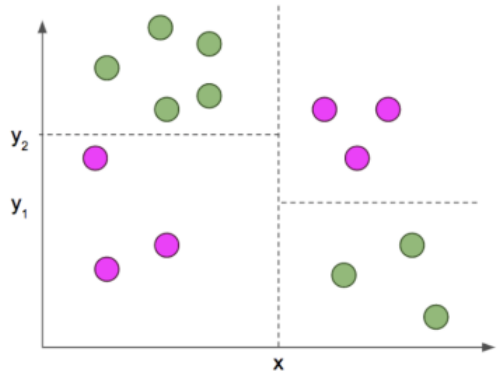
Se utiliza para tareas de clasificación como de regresión.

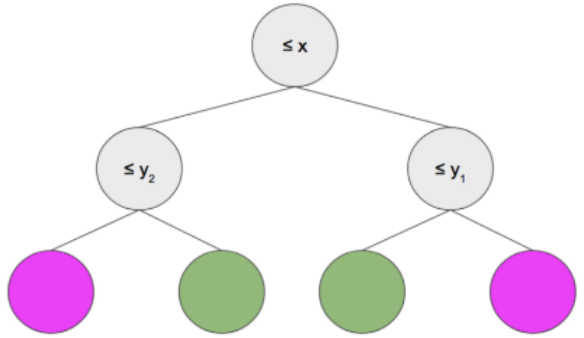
Tiene una estructura de árbol jerárquica, que consta de un nodo raíz, nodos intermedios y nodos hoja/terminal.



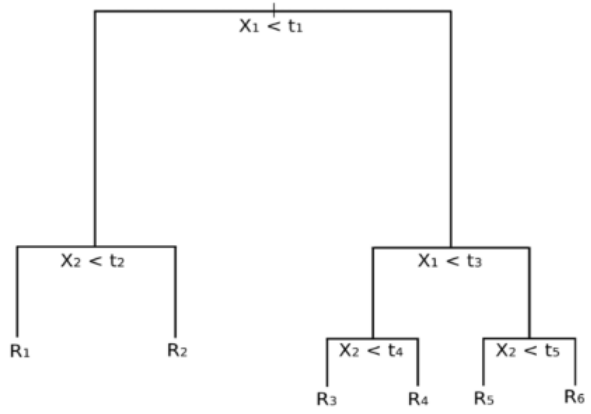
Arboles de Decisión

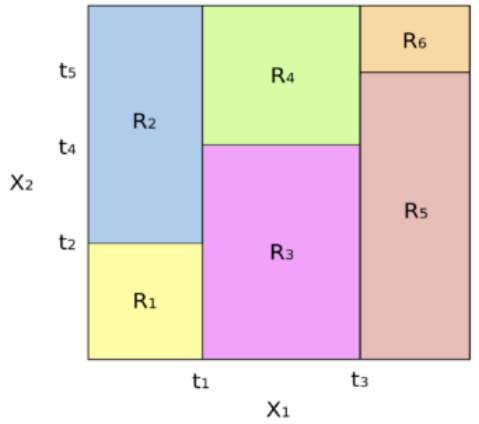
- Divide el espacio de predictores (variables independientes) en regiones distintas y no sobrepuestas.





Generalizando…





Tipos de Arboles

El primer algoritmo de Hunt, se desarrollo en la década de 1960 para modelar el aprendizaje humano en Psicología. Formando la base de los populares como:

- **ID3:** “Iterative Dichotomoser 3” Aprovecha la entropía y la ganancia de información como métrica para las divisiones.

Desarrollado por A.Ross Quinlan

Tipos de Arboles

- **C4.5:** Posterior al ID3, y desarrollado por Quinlan. Puede utilizar la ganancia de información o las proporciones de ganancia para evaluar puntos de división.

- **CART**: “Classification and Regression Trees” introducido por Leo Breiman. Y Hace uso de la impureza de Gini para identificar atributos ideales de división.
  - Impureza de Gini mide la frecuencia de clasificación incorrecta de un atributo elegido al azar. (valor bajo ideal)

DT en Regresión Vs. Clasificación

Regresión	Clasificación
Variable continua	Variable categórica
Valores de nodos terminales se reducen a la media de las observaciones en esa región	El valor en el nodo terminal se reduce a la moda de las observaciones del conjunto de entrenamiento que han “caído” en esa región

Como se crea un árbol

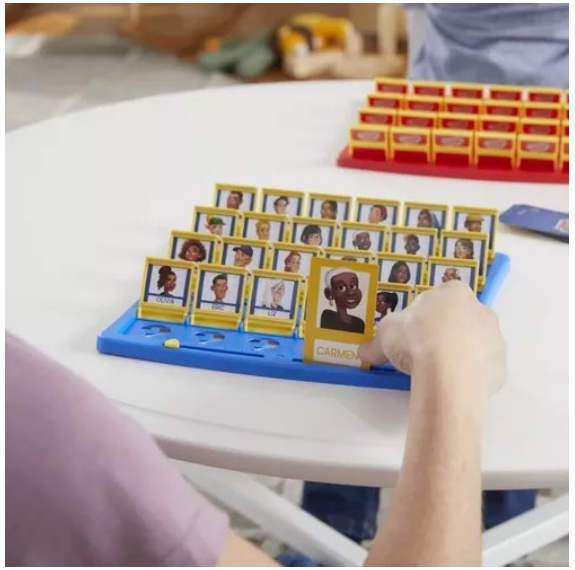
Según el **algoritmo de Hunt**, Buscamos la división en subconjuntos a través de una separación optima.

Dado unos datos, si pertenecen a la misma clase se consideran de un mismo nodo terminal, si son varias clases, se dividen en subconjuntos en función de una variable y un proceso iterativo.

Como se crea un árbol
¿Dónde ramificarse?

La idea central es buscar la mejor forma de ramificar, logrando hojas homogéneas.

- Los criterios de decisión dependen de la tarea (Clas. ó Regr.)
- La decisión de hacer divisiones afecta la precisión del árbol.
- Existen varios algoritmos para la ramificación (Elegir pregunta)
- La creación de subnodos incrementa la homogeneidad de los subnodos resultantes. (la pureza del nodo aumenta) (buscamos identificar homogeneidad entre personajes)
- Se prueba la división con diferentes variables y se escoge la que produce subnodos homogéneas. (La pregunta logro separar los personajes de la mejor forma?)



Adivina Quien?
Guess who?

Como se crea un árbol

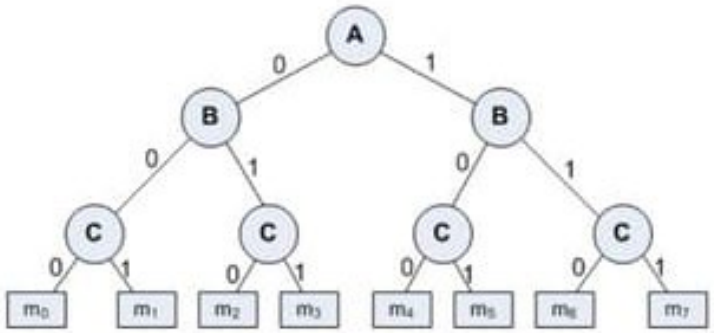
1. **División de nodos (Split)**: Seleccionarnos la característica y el umbral que mejor separan los datos en clases homogéneas en cada nodo del árbol.
2. **Criterio de impureza**: Medida (Gini o entropía) para evaluar que características y umbrales son los mas informativos al dividir los datos.
3. **Función de pérdida (Gain):** Minimizamos la perdida que cuantifica diferencias entre predicciones y Verdaderas.
4. **Regularización**: Evitar sobre ajustes, Se suele usar poda del árbol, profundidad máxima.

Como se crea un árbol (Splitter)

Para seleccionar el atributo que nos de la mejor división podemos considerar:

- Ramdon
- Best

A	B	C	f
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7



Como se crea un árbol (Splitter)

Ramdon

Elegimos aleatoriamente que atributo va primero, segundo, tercero…. Al final solo calculamos error de clase.

Best

La mejor división posible en cada nodo. Maximiza la ganancia.

Esto puede llevar a árboles más profundos y a una mayor precisión en la predicción, pero también puede aumentar el riesgo de sobreajuste.

Como se crea un árbol (Calidad del Split)

Para seleccionar el atributo que nos da la mejor división podemos considerar:

- Índice Gini
- Entropía y Ganancia de información
- Error de Clasificación (log loss)
- ...Chi cuadrado, Reducción de la varianza, ETC…

Métrica de división

Indice Gini ó impureza Gini:

Que tan impuros están los nodos de nuestro árbol.

Calcula la prob. De que un elemento elegido aleatoriamente sea etiquetado incorrectamente de acuerdo a una etiqueta dada.

$$\boldsymbol{Gini = 1 - \sum_{i=1}^{C} p_i^2}$$

Gini: impureza en el nodo

C: el numero de clases

p_i:probablidad de pertenencia a clase i

0 -> nodo puro 0.5 a 1-> nodo impuro (incorrecta división)

Métrica de división

Impureza con la Entropía:

Medida de impureza o aleatoriedad.

$$E(S) = \sum_{i=1}^{C} -p_i log_2 p_i$$

C: el numero de clases

S: Conjunto de datos donde calculamos la entropía

p_i:probablidad de pertenencia a clase i

0 -> nodo puro 1-> nodo impuro

Métrica de optimización

Ganancia de la información

Representa la diferencia entre dos entropías frente a atributos diferentes.

$$Gain(S, a) = E(S) - \sum_{V \in V_a} \frac{|S_V|}{|S|} E(S_V)$$

a: atributo especifico

E(s) Entropia del conjunto de datos S

Mas alto, mejor división de clases

Métrica de división en regresión

En casos de regresión suele usarse la medida de **sumatoria de residuos cuadrados o RSS (varianza)**

$$RSS = \sum_{i=1}^{n} (\boldsymbol{y_i} - \boldsymbol{\hat{y}_i})^2$$

$\boldsymbol{Y}_i$: Valor real

$\boldsymbol{\hat{y}_i}$: Valore predicho

Medida de discrepancia o error.

Rss bajo indica buen modelo

Métrica de optimización

Ganancia de la información en varianza

Representa la diferencia entre dos varianzas frente a atributos diferentes.

$$Gain(Rss_1, Rss_2) = Rss_1 - \sum_{Datos \in Rss} \frac{|Datos_1|}{|Datos_2|} Rss_2$$

a: atributo especifico

E(s) Entropía del conjunto de datos S

Mas alto, mejor división de clases

Regularización

El desafío más importantes en árboles de decisión.

Si no se definen límites, el árbol tendrá un 100% de precisión en el conjunto de datos de entrenamiento. En el peor caso tendrá una hoja por cada observación.

Dos formas de evitar el sobreajuste:

(a) Definir restricciones sobre el tamaño del árbol

(b) Podar el árbol.

Preprunning

Restricciones frente al tamaño del árbol

1. **Mínimo de observaciones por nodo**
   - Mínimo número de muestras
   - Valores altos previenen relaciones especifica (generaliza mejor)
   - Valores muy altos causan sobre ajuste

2. **Mínimo de observaciones en el nodo terminal**
   - Valores bajos son necesarios en clases no balanceadas

Preprunning

3. Máxima profundidad del árbol

- Mayor profundidad permite aprender relaciones mas especificas
- Se debe ajustar con validación cruzada (predict-true)

4. Máximo numero de hojas

- En lugar de máxima profundidad, numero máximo de hojas $profundidad_N = \max 2^n hojas$

5. Máximo numero de atributos por ramificación

- Selección aleatoria
- Como regla general, la raíz cuadrada funciona bien pero se debe probar hasta 30-40% del numero total de atributos

Postprunning (poda del árbol)

1.Complejidad (Cost Complexity Pruning):
Se activa al establecer el parámetro **ccp_alpha** en un valor diferente de cero.

El valor de **ccp_alpha** controla la cantidad de poda aplicada al árbol: cuanto mayor sea **ccp_alpha**, más poda se aplicará.

2.Profundidad del árbol (Depth-based Pruning):
Controlar la profundidad máxima del árbol mediante el parámetro **max_depth**.

Limitar la profundidad del árbol ayuda a prevenir el sobreajuste.

Postprunning (poda del árbol)

3.Numero mínimo de muestras en un nodo (Min Samples Split Pruning):
min_samples_split, para establecer el número mínimo de muestras requeridas para dividir un nodo interno.

Esto puede ayudar a evitar divisiones en nodos con muy pocas muestras, lo que puede conducir a un sobreajuste.

4.Número mínimo de muestras en una hoja (Min Samples Leaf Pruning):
El parámetro **min_samples_leaf** establece el número mínimo de muestras requeridas para ser una hoja.

Limitar este número puede ayudar a prevenir divisiones que resultan en hojas con muy pocas muestras.

Ventajas	Desventajas
Fácil construir, interpretar y visualizar	Tienden a tener overfit
Selección de Características importantes y no necesariamente se hace uso de todas	Muy influenciada por outliers
Si faltan datos, no llegaremos al nodo terminal pero predecimos hasta la ramificación	No suele ser buena en regresión
No es necesario (linealidad de datos)	Complejidad resta capacidad de interpretación
Sirve para datos cualitativos y cuantitativos (categóricos, numéricas…)	Arboles sesgados si hay desequilibrio de clases
Permite relaciones no lineales	Se pierde información al categorizar