Árboles de Decisión : Clasificación y Regresión

- Un *árbol de decisión es un algoritmo de aprendizaje supervisado* porque para que aprenda el modelo necesitamos una variable dependiente en el conjunto de entrenamiento.

- El *objetivo* es *crear un modelo que prediga el valor de una variable objetivo* mediante el aprendizaje de reglas de decisión simples inferidas a partir de las características de los datos.

- Un árbol de decisión presenta una estructura similar a un diagrama de flujo donde un nodo interno representa una característica (o atributo), la rama representa una regla de decisión y cada nodo hoja representa el resultado.

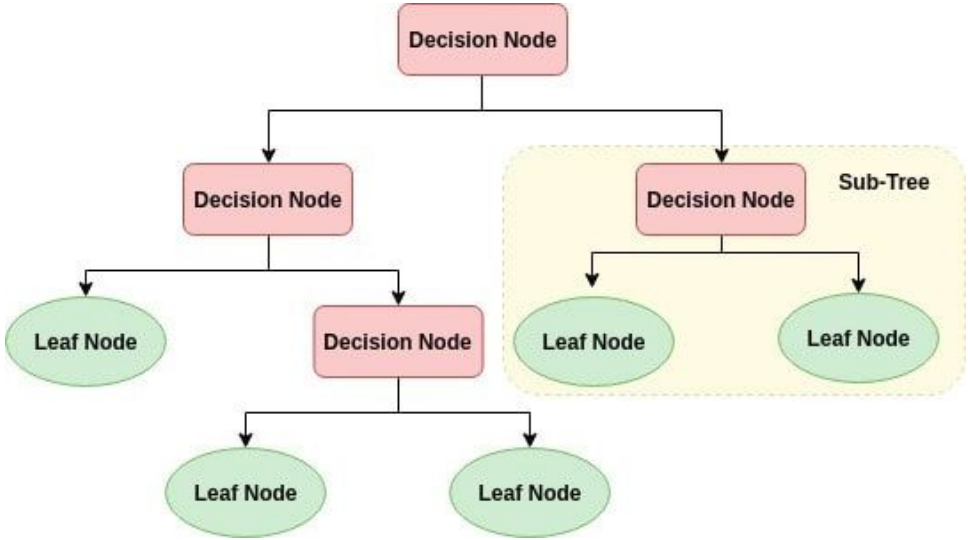
- El nodo superior en un árbol de decisión en Machine Learning se conoce como el ***nodo raíz***. Aprende a particionar en función del valor del atributo. Divide el árbol de una manera recursiva llamada partición recursiva.

- Esta estructura tipo diagrama de flujo lo ayuda a tomar decisiones. Es una visualización que imita fácilmente el pensamiento a nivel humano. Es por eso que los árboles de decisión son fáciles de entender e interpretar.

- Los árboles de decisión clasifican los ejemplos comenzando por la raíz hasta llegar a algún nodo hoja. Este enfoque se llama Enfoque de arriba hacia abajo (Top Down).

Árboles de Decisión : Clasificación y Regresión

- Cada nodo en el árbol actúa como un caso de prueba para algún atributo, y cada borde que desciende de ese nodo corresponde a una de las posibles respuestas al caso de prueba. Este proceso es recursivo y se repite para cada subárbol enraizado en los nuevos nodos.



Árboles de Decisión : Terminología

- Nodo raíz (nodo de decisión superior): Representa a toda la población o muestra y esto se divide en dos o más conjuntos homogéneos.
- División: Es un proceso de división de un nodo en dos o más subnodos.
- Nodo de decisión: Cuando un subnodo se divide en subnodos adicionales, se llama nodo de decisión.
- Nodo de hoja / terminal: Los nodos sin hijos (sin división adicional) se llaman Hoja o nodo terminal.

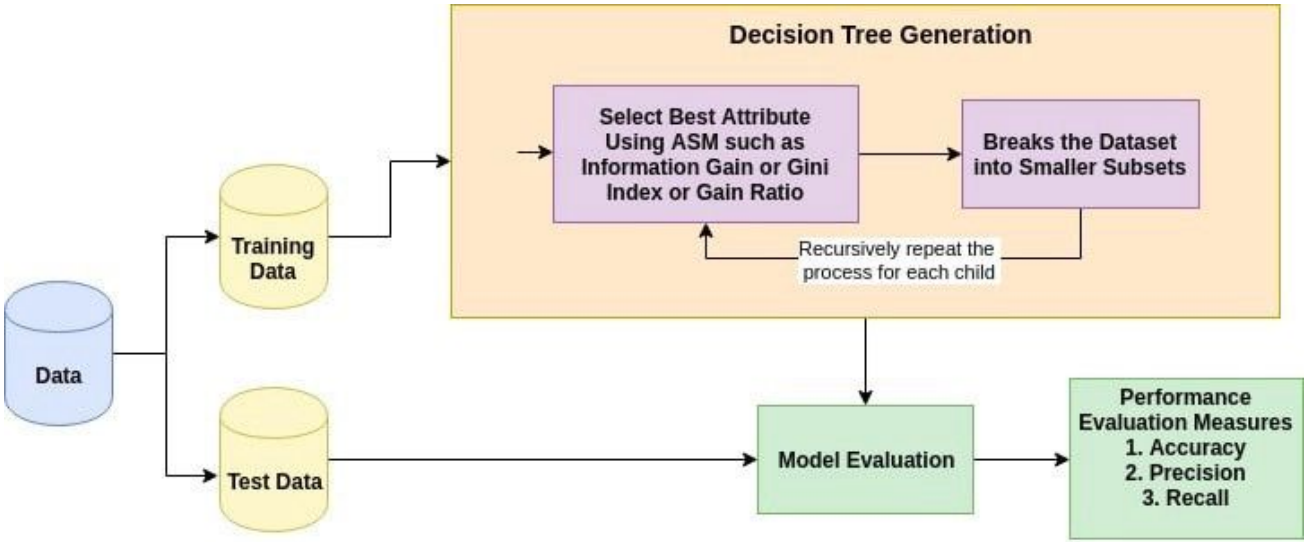
Árboles de Decisión : Terminología

- Poda: Cuando reducimos el tamaño de los árboles de decisión eliminando nodos (opuesto a la división), el proceso se llama poda.
- Rama / Subárbol: Una subsección del árbol de decisión se denomina rama o subárbol.
- Nodo padre e hijo: Un nodo, que se divide en subnodos se denomina nodo principal de subnodos, mientras que los subnodos son hijos de un nodo principal.

Árboles de Decisión : ¿Cómo funcion en Machine Learning?

- La idea detrás de cualquier algoritmo de árbol de decisión es el siguiente:
  - Seleccione el mejor atributo utilizando ***medidas de selección de atributos (ASM)*** para dividir los registros.
  - Haga que ese atributo sea un nodo de decisión y divida el conjunto de datos en subconjuntos más pequeños. Recursivamente hasta que una de las condiciones coincida:
    - Todas las tuplas pertenecen al mismo valor de atributo.
    - No quedan más atributos.
    - No hay más instancias.

Árboles de Decisión : ¿Cómo funcion en Machine Learning?



Árboles de Decisión : Algoritmo ID3

ID3 significa Iterative Dichotomizer 3. El algoritmo ID3 fue inventado por Ross Quinlan. Construye un árbol de decisiones a partir de un conjunto fijo de ejemplos y el árbol resultante se usa para clasificar muestras futuras.

La idea básica es construir el árbol de decisiones empleando una búsqueda de arriba hacia abajo a través de los conjuntos dados para probar cada atributo en cada nodo de árbol.

Suena simple, pero ¿qué nodo deberíamos seleccionar para construir el árbol de decisión correcto y más preciso? ¿Cómo decidiríamos eso?

Bueno, tenemos algunas medidas que pueden ayudarnos a seleccionar la mejor opción.

Árboles de Decisión : Medidas de Selección Atributos

La medida de selección de atributos es una heurística para seleccionar el criterio de división que divide los datos de la mejor manera posible.

ASM proporciona un rango para cada característica (o atributo) al explicar el conjunto de datos dado. El atributo de mejor puntuación se seleccionará como un atributo de división.

Árboles de Decisión : Medidas de Selección Atributos

Entropía

- *En teoría de la información, la entropía es una medida de la incertidumbre sobre una fuente de mensajes. Nos da el grado de desorganización en nuestros datos.*

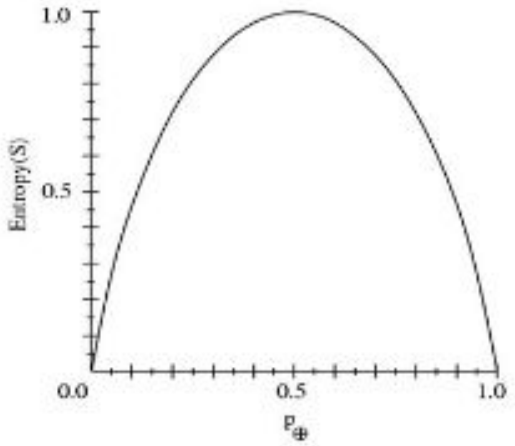
- *Dada una colección S que contiene ejemplos positivos y negativos de algún concepto objetivo, la entropía de S relativa a esta clasificación booleana es:*

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Árboles de Decisión : Medidas de Selección Atributos

Entropía

- *p + y p- son la proporción de ejemplos positivos y negativos en S.*
- *Considerar esta función de entropía relativa a una clasificación booleana, ya que la proporción de ejemplos positivos p + varía entre 0 y 1.*



Árboles de Decisión : Medidas de Selección Atributos

Entropía

- *La entropía es 0 si todos los miembros de S pertenecen a la misma clase. Por ejemplo, si todos los miembros son positivos (p + = 1), entonces p- es 0 y Entropy (S) = -1. log2 (1) - 0. log2 0 = -1. 0 - 0. log2 0 = 0.*
- *La entropía es 1 cuando la colección contiene una cantidad igual de ejemplos positivos y negativos.*
- *Si la colección contiene números desiguales de ejemplos positivos y negativos, la entropía está entre 0 y 1.*

Árboles de Decisión : Medidas de Selección Atributos

Ganancia de información

- *Mide la reducción esperada en entropía. Decide qué atributo va a un nodo de decisión. Para minimizar la profundidad del árbol de decisión, el atributo con la mayor reducción de entropía es la mejor opción.*

- *Más precisamente, la Ganancia (Gain) de información (S, A) de un atributo A con respecto a una colección de ejemplos S se define como:*

$$Gain(S, A) = \underbrace{Entropy(S)}_{\text{original entropy of S}} - \underbrace{\sum_{v \in values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)}_{\text{relative entropy of S}}$$

S = Cada valor v de todos los valores posibles del atributo A

Sv = Subconjunto de S para el cual el atributo A tiene valor v

| Sv | = Número de elementos en Sv

| S | = Número de elementos en S

Árboles de Decisión : Medidas de Selección Atributos

Ejemplo de cálculo de ganancia de información

- *Supongamos que queremos que ID3 decida si el clima es bueno para jugar al béisbol. En el transcurso de dos semanas, se recopilan datos para ayudar a ID3 a construir un árbol de decisiones.*

- *La clasificación objetivo es "¿deberíamos jugar al béisbol?", Que puede ser sí o no.*

Día	Vista	Temperatura	Humedad	Viento	¿Debo jugar al béisbol?
D1	Soleado	Caliente	Alto	Débil	No
D2	Soleado	Caliente	Alto	Fuerte	No
D3	Nublado	Caliente	Alto	Débil	Sí
D4	Lluvia	Suave	Alto	Débil	Sí
D5	Lluvia	Genial	Normal	Débil	Sí
D6	Lluvia	Genial	Normal	Fuerte	No
D7	Nublado	Genial	Normal	Fuerte	

Árboles de Decisión : Medidas de Selección Atributos

Ejemplo de cálculo de ganancia de información

- *Los atributos climáticos son la vista, la temperatura, la humedad y la velocidad del viento.*
- *Pueden tener los siguientes valores:*
  - *vista = {soleado, nublado, lluvia}*
  - *temperatura = {caliente, suave, fresco}*
  - *humedad = {alto, normal}*
  - *viento = {débil, fuerte}*

Árboles de Decisión : Medidas de Selección Atributos

Ejemplo de cálculo de ganancia de información

- *Necesitamos encontrar qué atributo será el nodo raíz en nuestro árbol de decisión.*
- *Entropía (S) = – (9/14) Log2 (9/14) – (5/14) Log2 (5/14) = 0.940*
- *Ganancia (S, Viento) = Entropía (S) – (8/14) * Entropía (Sdebil) – (6/14) * Entropía (Sfuerte)*

  *= 0.940 – (8/14) * 0.811 – (6/14) * 1.00*

  = 0.048
- *Entropía (Sdebil) = – (6/8) * log2 (6/8) – (2/8) * log2 (2/8) = 0.811*
- *Entropía (Sfuerte) = – (3/6) * log2 (3/6) – (3/6) * log2 (3/6) = 1.00*

Para cada atributo, la ganancia se calcula y la mayor ganancia se usa en la decisión.

Árboles de Decisión : Medidas de Selección Atributos

Ejemplo de cálculo de ganancia de información

- *Para cada atributo, la ganancia se calcula y la mayor ganancia se usa en la decisión.*
- *Ganancia (S, Vista) = 0.246*
- *Ganancia (S, temperatura) = 0.029*
- *Ganancia (S, Humedad) = 0.151*
- *En conclusión, el atributo Vista tiene la mayor ganancia. Por lo tanto, se usa como el atributo de decisión en el nodo raíz.*
- *Como la perspectiva tiene tres valores posibles, el nodo raíz tiene tres ramas (soleado, nublado, lluvia). La siguiente pregunta es: ¿Qué atributo debería probarse en el nodo soleado de la rama? Como hemos utilizado Vista en la raíz, sólo decidimos los tres atributos restantes: humedad, temperatura o viento.*

Árboles de Decisión : Medidas de Selección Atributos

Ejemplo de cálculo de ganancia de información

- *Ssoleado = {D1, D2, D8, D9, D11} = 5 ejemplos de la tabla con vista = soleado.*
- *Ganancia (Ssoleado, Humedad) = 0.970*
- *Ganancia (Ssoleado, temperatura) = 0.570*
- *Ganancia (Ssoleado, Viento) = 0.019*
- *La humedad tiene la mayor ganancia; por lo tanto, se usa como nodo de decisión. Este proceso continúa hasta que todos los datos se clasifiquen perfectamente o nos quedemos sin atributos.*

Árboles de Decisión : Medidas de Selección Atributos

RSS (Residual Sum of Squares)

- *En el caso de los árboles de decisión de un problema de regresión se utiliza el RSS (Residual Sum of Squares) que es una medida de la discrepancia entre los datos reales y los predichos por el modelo.*

- *Un RSS bajo indica un buen ajuste del modelo a los datos, es decir, se busca minimizar el RSS.*

- *Se define el RSS como:*

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

Árboles de Decisión : Ventajas

- *Fácil de entender e interpretar. Los árboles pueden ser visualizados.*
- *Requiere poca preparación de datos. Otras técnicas requieren a menudo la normalización de los datos, la creación de variables ficticias y la eliminación de valores en blanco.*
- *Capaz de manejar datos numéricos y categóricos. Otras técnicas suelen estar especializadas en el análisis de conjuntos de datos que sólo tienen un tipo de variable.*
- *Capaz de manejar problemas de múltiples salidas.*
- *Utiliza un modelo de caja blanca. Si una situación dada es observable en un modelo, la explicación de la condición se explica fácilmente por la lógica booleana. Por el contrario, en un modelo de caja negra (por ejemplo, en una red neuronal artificial), los resultados pueden ser más difíciles de interpretar.*

Árboles de Decisión : Desventajas

- *Tienden al sobreajuste u overfitting de los datos, por lo que el modelo al predecir nuevos casos no estima con el mismo índice de acierto.*

- *Se ven influenciados por los outliers, creando árboles con ramas muy profundas que no predicen bien para nuevos casos. Se deben eliminar dichos outliers.*

- *No suelen ser muy eficientes con modelos de regresión.*

- *Crear árboles demasiado complejos puede conllevar que no se adapten bien a los nuevos datos. La complejidad resta capacidad de interpretación.*

Árboles de Decisión : Tipos de árboles

Regresión	Clasificación
Variable dependiente es continua	Variable dependiente es categórica
Valores de los 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.

Árboles de Decisión : ¿Qué problemas puedo resolver?

- *Cada instancia del dataset se encuentra representada por pares de atributos con valores posibles que tipifican a dichos atributos .*
- *La función de predicción tiene valores de salida discretos.*
- *Modelos donde el training data puede contener errores, ya sean errores en la clasificación de las muestras de entrenamiento u errores en los valores de los atributos que describen esas muestras .*
- *Modelos donde el dataset de entrenamiento tiene valores faltantes (missing values).*

Random Forest

- *Uno de los problemas que aparecía con la creación de un árbol de decisión es que si le damos la profundidad suficiente, el árbol tiende a "memorizar" las soluciones en vez de generalizar el aprendizaje. Es decir, a padecer de overfitting.*

- *Random Forest es un técnica de aprendizaje automático supervisada basada en árboles de decisión. Su principal ventaja es que obtiene un mejor rendimiento de generalización para un rendimiento durante entrenamiento similar.*

- *Esta mejora en la generalización la consigue compensando los errores de las predicciones de los distintos árboles de decisión. Para asegurarnos que los árboles sean distintos, lo que hacemos es que cada uno se entrena con una muestra aleatoria de los datos de entrenamiento. Esta estrategia se denomina* ***bagging****.*

- *La solución para evitar esto es la de crear muchos árboles y que trabajen en conjunto.*

Random Forest : ¿Cómo funciona el algoritmo?

En forma resumida, el algoritmo sigue este proceso :

- *Selecciona individuos al azar (usando muestreo con reemplazo) para crear diferentes set de datos.*
- *Crea un árbol de decisión con cada set de datos, obteniendo diferentes árboles, ya que cada set contiene diferentes individuos y diferentes variables.*
- *Al crear los árboles se eligen variables al azar en cada nodo del árbol, dejando crecer el árbol en profundidad (sin podar).*
- *Predice los nuevos datos usando el "voto mayoritario", donde clasificará como "positivo" si la mayoría de los árboles predicen la observación como positiva.*

Random Forest : Hiperparámetros

Estos son los hiperparámetros más útiles:

- ***n_estimators**: número de árboles que va a tener el algoritmo. Normalmente cuantos más mejor, pero a partir de cierto punto deja de mejorar y sólo hace que vaya más lento. Un buen valor por defecto puede ser el uso de 100 árboles.*

- ***n_jobs**: número de cores que se pueden usar para entrenar los árboles. Cada árbol es independiente del resto, así que entrenar un bosque aleatorio es una tarea muy paralelizable. Por defecto sólo utiliza 1 core de la CPU. Para mejorar el rendimiento puedes usar tantos cores como estimes necesario. Si usas n_jobs = -1, estás indicando que quieres usar tantos cores como tenga la PC o servidor en el cual se ejecuta.*

- ***max_features**: usa forma de garantizar que los árboles son diferentes, es que todos se entrenan con una muestra aleatoria de los datos. Si queremos que todavía sean más diferentes, podemos hacer que distintos árboles usen distintos atributos. Esto puede ser útil especialmente cuando algunos atributos están relacionados entre sí.*

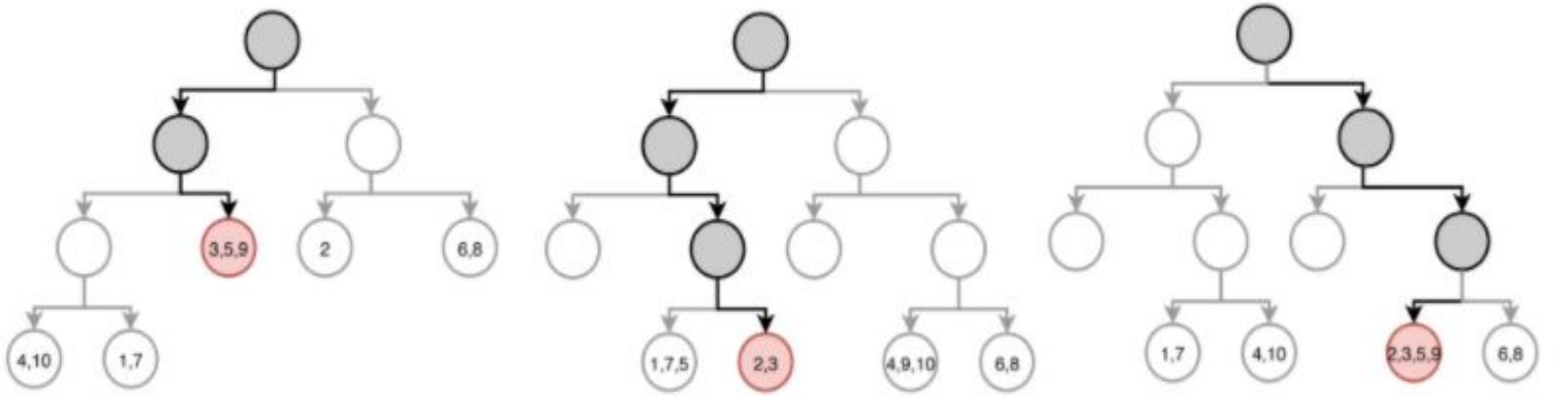
Random Forest : Hiperparámetros

Regularización (también disponibles para Decision Trees):

- ***max_depth**: la profundidad máxima del árbol.*
- ***min_samples_split**: número mínimo de muestras necesarias antes de dividir este nodo. También se puede expresar en porcentaje.*
- ***min_samples_leaf**: número mínimo de muestras que debe haber en un nodo final (hoja). También se puede expresar en porcentaje.*
- ***max_leaf_nodes**: número máximo de nodos finales.*

Random Forest : Cálculo de Predicción

- *La predicción de un modelo Random Forest es la media de las predicciones de todos los árboles que lo forman. Esto equivale a la media ponderada de todas las observaciones, empleando como pesos w la media de los vectores de pesos de todos los árboles.*



Predicción con random forest: en cada árbol, el camino hasta llegar al nodo final está resaltado. En cada nodo terminal se detalla el índice de las observaciones de entrenamiento que forman parte de él.

Random Forest : Cálculo de Predicción

- *Acorde a la imagen anterior, el vector de pesos para cada uno de los tres árboles (de izquierda a derecha) es:*

$$\mathbf{w}_{arbol_1} = (0, 0, \frac{1}{3}, 0, \frac{1}{3}, 0, 0, 0, \frac{1}{3}, 0)$$

$$\mathbf{w}_{arbol_2} = (0, \frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0, 0, 0, 0)$$

$$\mathbf{w}_{arbol_3} = (0, \frac{1}{4}, \frac{1}{4}, 0, \frac{1}{4}, 0, 0, 0, \frac{1}{4}, 0)$$

- *La media de todos los vectores de pesos es:*

$$\overline{\mathbf{w}} = \frac{1}{3}(\mathbf{w}_{arbol_1} + \mathbf{w}_{arbol_2} + \mathbf{w}_{arbol_3}) = (0, \frac{1}{4}, \frac{13}{36}, 0, \frac{7}{36}, 0, 0, 0, \frac{7}{36}, 0)$$

Random Forest : Cálculo de Predicción

- *Una vez obtenido el vector de pesos promedio, se puede calcular la predicción con la media ponderada de todas las observaciones de entrenamiento:*

$$\hat{\mu} = \sum_{i=1}^{n} \overline{\mathbf{w}}_i Y_i$$

$$\hat{\mu} = (0 \times 10) + (\frac{1}{4} \times 18) + (\frac{13}{36} \times 24) + (0 \times 8) + (\frac{1}{4} \times 2) +$$
$$(0 \times 9) + (0 \times 16) + (0 \times 10) + (\frac{1}{4} \times 20) + (0 \times 14) = 17.4$$

Bibliografía

- https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | https://www.deeplearningbook.org/