**Nombre del Algoritmo:** Decision Tree Learning

**Descripción del Problema Computacional para la cual el algoritmo es una solución.**

**Delimitación del Problema:**

El algoritmo de aprendizaje de árboles de decisión busca dividir un conjunto de datos 𝑋 en grupos homogéneos con respecto a una variable objetivo 𝑦, generando una estructura jerárquica en forma de árbol. Cada nodo representa una decisión basada en una característica, y las hojas representan clases o valores predichos.

**Comprensión del Problema:**

El objetivo es construir un árbol de decisión que pueda ser utilizado para clasificar nuevas instancias o predecir valores. Se basa en criterios como la ganancia de información o el índice Gini para seleccionar las divisiones más significativas.

**Clasificación del Problema:**

* Problema supervisado: Entrenamiento con datos etiquetados.
* Clasificación y regresión: Aplicable a tareas discretas y continuas.
* Algoritmo greedy: Utiliza una estrategia codiciosa para construir el árbol.

**Viabilidad Computacional:**

El aprendizaje de árboles de decisión es computacionalmente eficiente en la mayoría de los casos, pero puede ser costoso para grandes conjuntos de datos si no se optimizan los cálculos.

**Contexto del Problema:**

**Análisis de Datos y Estructura del Problema:**

* Datos de entrada: Matriz de características 𝑋 y vector de etiquetas 𝑦.
* Estructura del problema:
  + Divisiones recursivas basadas en criterios como entropía, ganancia de información o Gini.
  + Árbol binario o de múltiples ramas dependiendo de la implementación.

**Evaluación de Complejidad:**

* Entrenamiento:
  + Complejidad promedio: 𝑂(𝑛⋅𝑚⋅log(𝑛)), donde 𝑛 es el número de instancias y 𝑚 el número de características.
  + En casos extremos (sin poda), puede ser 𝑂 (𝑛 ²⋅𝑚).
* Predicción:
  + Complejidad: 𝑂(𝑑) donde 𝑑 es la profundidad del árbol.

**Análisis del Contexto del Problema para la cual el algoritmo es una solución**.

**Contexto general del Problema:**

Árboles de decisión son aplicados en:

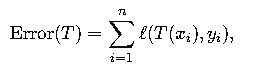
* Diagnóstico médico.
* Clasificación de texto e imágenes.
* Sistemas de recomendación.
* Modelos de toma de decisiones en finanzas.

**Identificación de los conjuntos y estructuras de datos:**

* Conjunto de características 𝑋∈𝑅𝑛×𝑚 .
* Etiquetas 𝑦 categóricas o continuas.
* Estructuras:
  + Árboles binarios para divisiones sencillas.
  + Árboles multiclase para problemas complejos.

**Formalización del problema:**

Dado un conjunto de entrenamiento 𝐷=(𝑋, 𝑦), encontrar un árbol 𝑇 que minimice el error de predicción:



donde ℓ es una función de pérdida.

**Análisis de la Dinámica del Problema:**

* Proceso recursivo que divide el espacio de datos según condiciones de partición.
* Terminación: Cuando se alcanza un criterio de parada como la profundidad máxima o el número mínimo de ejemplos en un nodo.

**Evaluación de la Complejidad del Contexto:**

* Limitaciones en datos desequilibrados: El modelo puede sobreajustarse a clases dominantes.
* Datos con ruido: Árboles profundos pueden sobreajustarse.

**Identificación de los casos límites y excepciones:**

* Caso límite: Todos los datos pertenecen a la misma clase en un nodo.
* Excepciones: Características categóricas con demasiados valores únicos.

**Interacción con otros sistemas:**

* Integración con sistemas de clasificación por conjunto (e.g., random forests).
* Preprocesamiento: Normalización o codificación de datos categóricos.

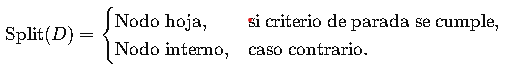
**Especificación Formal:**

**Definición de las entradas y salidas:**

* Entradas:
  + Matriz de características 𝑋.
  + Vector de etiquetas 𝑦.
  + Parámetros como profundidad máxima y tamaño mínimo de nodos.
* Salidas:
  + Un árbol de decisión 𝑇.

**Definición formal del problema mediante funciones:**

El árbol de decisión 𝑇 se construye mediante una función recursiva Split(𝐷):



**Relaciones y Restricciones:**

* La división debe maximizar la pureza de los nodos hijos.
* Restricción de profundidad máxima o número mínimo de ejemplos en nodos.

**Invariantes y propiedades:**

* Todos los ejemplos en un nodo hoja pertenecen a la misma clase (en clasificación pura).
* Las divisiones en los nodos internos no se solapan.

**Control de Flujo:**

* Evaluar el conjunto de datos en el nodo actual.
* Si cumple un criterio de parada, marcarlo como hoja.
* Si no, dividir el conjunto de datos y crear subárboles.

**Pseudocodigo del Algoritmo**

Algoritmo AprendizajeArbolDecision

Definir datos\_X como Matriz[n][m] // Conjunto de características

Definir etiquetas\_Y como Vector[n] // Etiquetas asociadas

Definir profundidad\_maxima como Entero

Definir tamanio\_minimo\_nodo como Entero

Funcion ConstruirArbol(datos\_X, etiquetas\_Y, profundidad\_actual)

Si profundidad\_actual >= profundidad\_maxima o tamaño(datos\_X) <= tamanio\_minimo\_nodo Entonces

Retornar CrearHoja(etiquetas\_Y)

FinSi

mejor\_division <- EncontrarMejorDivision(datos\_X, etiquetas\_Y)

Si mejor\_division no existe Entonces

Retornar CrearHoja(etiquetas\_Y)

FinSi

nodo <- CrearNodo(mejor\_division.condicion)

nodo.hijo\_izquierdo <- ConstruirArbol(mejor\_division.izquierda\_X, mejor\_division.izquierda\_Y, profundidad\_actual + 1)

nodo.hijo\_derecho <- ConstruirArbol(mejor\_division.derecha\_X, mejor\_division.derecha\_Y, profundidad\_actual + 1)

Retornar nodo

FinFuncion

// Proceso principal

Escribir "Construyendo el árbol de decisión..."

arbol <- ConstruirArbol(datos\_X, etiquetas\_Y, 0)

Escribir "Árbol de decisión construido exitosamente"

FinAlgoritmo

**Caso de Usos Posibles:**

* Diagnóstico médico:
  + Clasificación de pacientes con base en síntomas.
* Clasificación de texto:
  + Determinar el tema de un documento.
* Análisis financiero:
  + Predecir si una transacción es fraudulenta.
* Clasificación de imágenes:
  + Distinguir objetos en imágenes simples.