UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Una Introducción al Aprendizaje de Máquina basado en Árboles de Clasificación

MONOGRAFÍA DE TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE MATEMÁTICO
PROYECTO CURRICULAR DE MATEMÁTICAS

Oscar David Guzmán Barrera, Santiago Vargas Gonzalez
Dirigido por: Luis Alejandro Masmela Caita

Bogotá DC
Octubre de 2022

Resumen

En el área de Machine Learning, el modelo de Árboles de Clasificación es uno de los más populares en el ámbito de Aprendizaje Supervisado. Por lo tanto, basados el trabajo de (Breiman et al., 1984), se plantea describir de manera matemática la construcción de un modelo de árbol de clasificación ilustrando cada concepto a través de la base de datos IRIS (Fisher, 1936). Además, se tratarán temas como el rendimiento del modelo, el podado y la validación cruzada con el fin de obtener un modelo de árbol eficiente.

Palabras clave: Árboles de Clasificación, Machine Learning, Aprendizaje Supervisado.

Clasificación AMS: 68T01

Agradecimientos: Un agradecimiento especial a nuestras familias; al docente Luis Alejandro Masmela por su apoyo y paciencia; también a la Universidad Distrital quien nos acogió en este proceso.

Introducción

El aprendizaje supervisado es una de las principales ramas del Machine Learning. Este consiste en el diseño de algoritmos matemáticos que en un conjunto de datos dado relacionen un grupo de variables independientes con una variable dependiente. Uno de los métodos más conocidos son los Árboles de Clasificación, pues su funcionamiento es simple y además se han usado para modelar diferentes aplicaciones tal como la clasificación de letras en una imagen, predecir la enfermedad de un paciente basado en sus síntomas, etc.

En este trabajo se estudiarán los árboles de clasificación planteados por Leo Breiman (Breiman et al., 1984) acompañado con las lecturas de (Choi, 2017). Cabe resaltar que según (Loh, 2014), dichos árboles de clasificación están basados en dos artículos que son los cimientos de los árboles de regresión en (Morgan y Sonquist, 1963), y clasificación en (Messenger y Mandell, 1972) respectivamente.

Ahora bien, el principio fundamental de los árboles de clasificación de Breiman es crear particiones en un conjunto de datos a partir de "preguntas" que tienen una respuesta cerrada, si o no. Por ejemplo, el siguiente árbol clasifica en una muestra de agua su potabilidad usando preguntas acerca de condiciones químicas en el líquido, como:

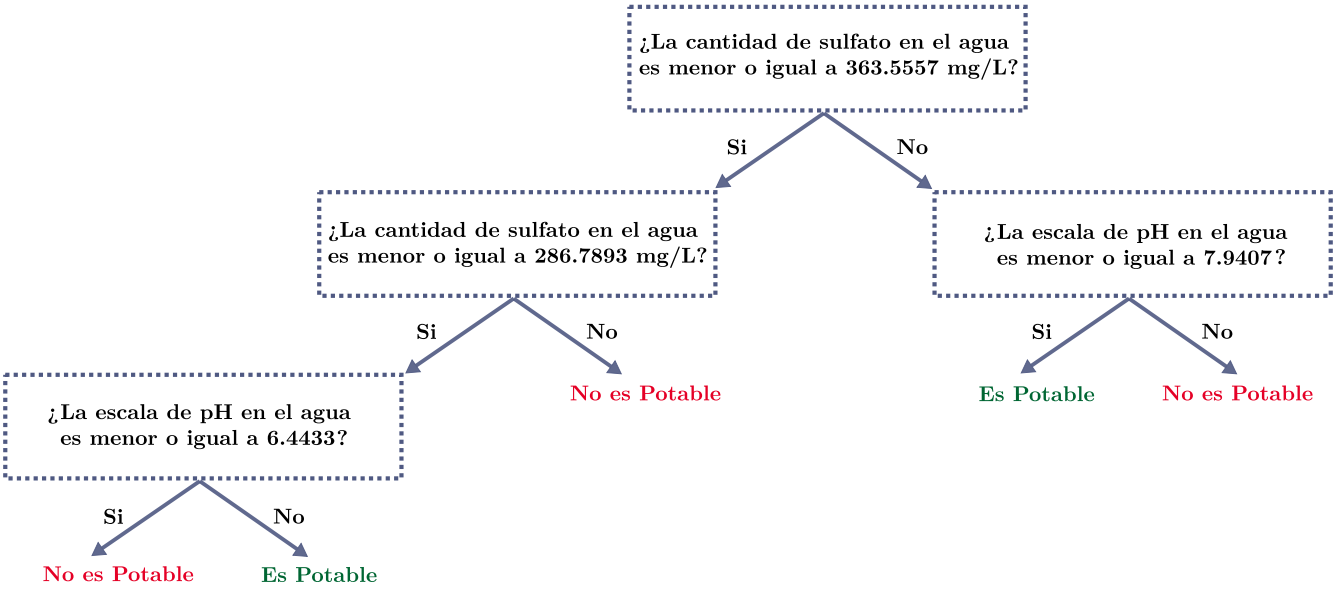


Figura 1: Árbol de Decisión creado a partir de los datos extraidos en: https://www.kaggle.com/datasets/adityakadiwal/water-potability

Aunque pueda parecer sencilla la construcción de los árboles, surgen interrogantes acerca de: las preguntas formuladas, la cantidad de preguntas y el rendimiento del árbol. La solución a dichas cuestiones son los objetivos principales de este trabajo, que además se abordarán de manera matemática.

Por lo tanto, esta monografía se llevará a cabo en el siguiente orden: en la sección 1 se introducirá la base datos IRIS, y se construirá un primer árbol de clasificación basado en una porción de dicha base de datos; en la sección 2 se planteará el "podado", el cual crea un conjunto de árboles cada uno basado en el árbol de la sección 1; en la sección 3 seleccionará el mejor árbol de la sección 2 usando la validación cruzada; y por último, en la sección 4 se hallará el mejor árbol para la base de datos IRIS completa.

Algunas demostraciones fueron omitidas en este documento, sin embargo, en **esta extensión** del trabajo se puede consultar cada una de ellas. Además, se creó un programa en Python para modelar un árbol de clasificación el cual puede ser encontrado **Aquí**.

1. Árbol Binario de Clasificadores Estructurados

1.1. Base de Datos

El conjunto de datos IRIS (Fisher, 1936) consta de 150 observaciones del ancho y alto, del pétalo y el sépalo de tres tipos de plantas: Iris Setosa, Iris Virginica e Iris Versicolor.

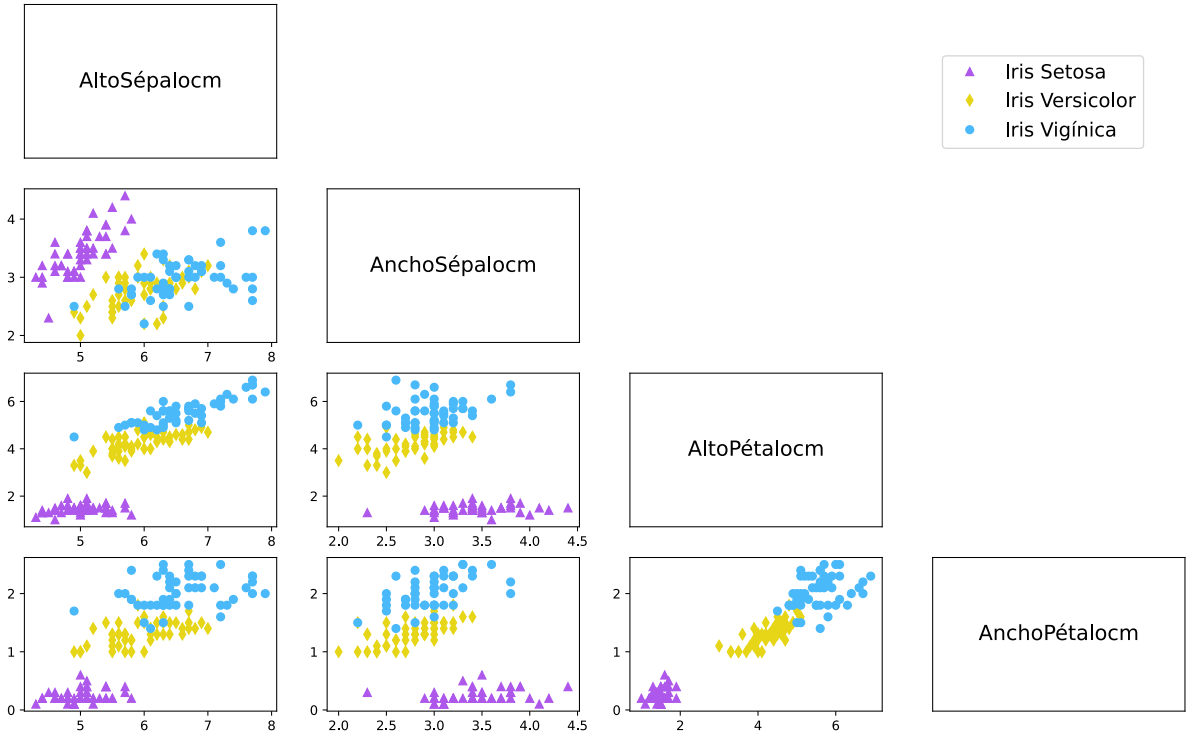


Figura 2: Digrama de dispersión por pares entre las variables que conforman la base de datos Iris.

Con el fin de ilustrar algunos ejemplos de las definiciones que se verán a lo largo del documento, se usará la base de datos IRIS con las variables alto y ancho del pétalo. La siguiente gráfica muestra cómo se relaciona dicho subconjunto de la base de datos.

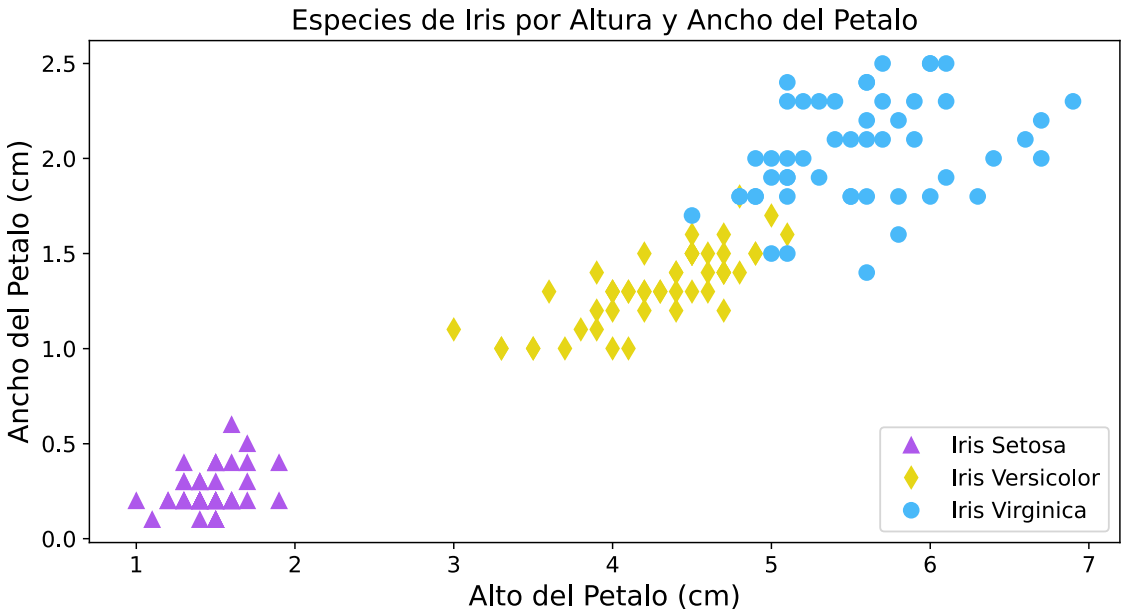


Figura 3: Gráfica de IRIS usando las variables de ancho y alto del pétalo.

En la Figura 3 se puede identificar como los valores de alto y ancho del pétalo se asocian a uno de los tres tipos de flor. Este conjunto de observaciones en el cual se relacionan las variables con una clase se conoce como muestra de aprendizaje. La función de dicha muestra es entrenar el modelo de clasificación. Esto se define formalmente como:

Definición 1. *Una muestra de aprendizaje es un conjunto de la forma:*

$$\mathcal{L} = \left\{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\right\},$$

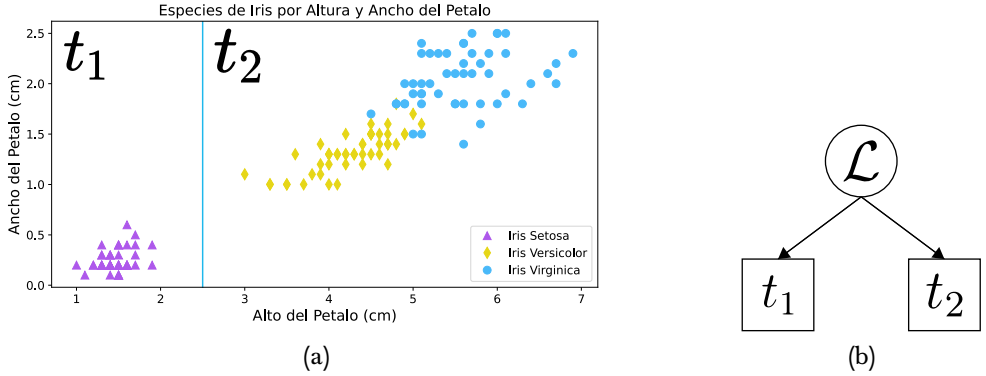
donde $\vec{x}_i$ pertenece al conjunto de variables independientes X, y y_i a su asociación en el conjunto de la variable dependiente $C = \{1, 2, \ldots, J\}$.

En nuestro caso, cada vector de $\mathcal{L}$ tiene como primera componente una dupla con las variables alto y ancho del pétalo en el conjunto X de variables independientes; y como segunda componente, a una asociación numérica de las clases de flor Iris (Setosa, Virginica y Versicolor) en el conjunto C.

Ahora bien, con la muestra de aprendizaje, se procederá a mostrar cómo formular preguntas que creen buenas divisiones en $\mathcal{L}$ para construir un árbol de clasificación.

1.2. La reducción de impureza y las divisiones más óptimas

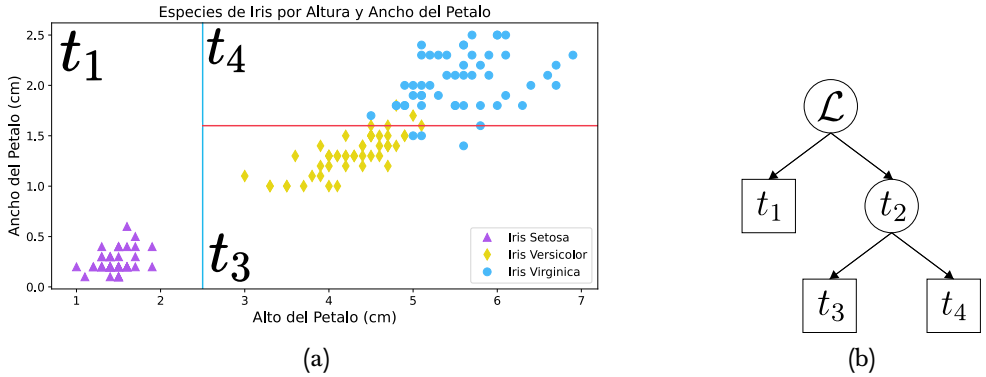
Para comenzar, en la Figura 3, se puede observar que algunas clases pueden ser separadas de manera evidente. Por ejemplo, la clase Iris Setosa es agrupada en el conjunto t_1 que toma las observaciones con el alto de pétalo menor o igual a $2.5cm$. Por otro lado, el conjunto restante t_2 conformado por las observaciones con alto del pétalo mayor a $2.5cm$ tiene dos flores involucradas como se puede ver en la Figura 4 (a).



(a) (b)

Figura 4: En (a) se muestra el gráfico de los datos con la división realizada. En (b) se muestra la visualización de las divisiones como un grafo de árbol. Donde el nodo raíz corresponde a $\mathcal{L}$; las hojas (o nodos terminales), corresponden a los nodos t_1, t_2.

A su vez, puede realizarse una partición sobre t_2 conformada por dos nodos t_3, t_4, donde t_3 corresponde a las observaciones de t_2 que tienen un ancho del pétalo menor o igual a $1.6cm$ y t_4 mayor a $1.6cm$, esto se puede ver representado en la Figura 5 (a).



(a) (b)

Figura 5: En (a) el gráfico de los datos con las divisiones realizadas. En (b) la visualización de las divisiones como un grafo de árbol. Donde el nodo raíz corresponde a $\mathcal{L}$; las hojas (o nodos terminales), corresponden a los nodos t_1, t_3, t_4; y los nodos intermedios, en este caso corresponde al nodo t_2.

En la Figura 5 (a) se puede notar que las observaciones han quedado divididas por los subconjuntos t_1, t_3, t_4. Es claro que el subconjunto t_1 contiene solo observaciones clasificadas dentro de la flor Iris Setosa. Por otra parte, los subconjuntos t_3 y t_4 contienen observaciones clasificadas en dos flores diferentes.

Uno de los principales objetivos en la construcción de los árboles de clasificación es generar divisiones en la muestra de aprendizaje, de tal manera que, en la mayoría de particiones resida una especie de Iris. Por lo tanto, es importante conocer cuáles son las proporciones de las clases correspondientes a las observaciones dentro de los nodos, y en general en el árbol. Así pues, es necesario mostrar las siguientes definiciones:

Definición 2. *La cantidad de observaciones en $\mathcal{L}$ clasificadas en $j \in C$ van a ser denotadas como:*

$$N_j = |\{(x_i, y_i) \in \mathcal{L} \mid y_i = j\}|,$$

y la cantidad de datos en el nodo t clasificados en $j \in C$:

$$N_j(t) = |\{(x_i, y_i) \in t \mid y_i = c_j\}|.$$

Definición 3. *Las observaciones de $\mathcal{L}$ contenidas en el nodo del árbol t van a ser denotadas como:*

$$\mathcal{L}(t) = \{(x_i, y_i) \in \mathcal{L} \mid (x_i, y_i) \in t\},$$

y la cantidad de observaciones de $\mathcal{L}$ contenidas en un nodo t se notará como $|\mathcal{L}(t)| = N(t)$.

Definición 4. *La proporción de observaciones en $\mathcal{L}$ clasificadas en $j \in C$ es:*

$$\pi(j) = \frac{N_j}{N}.$$

Definición 5. *La proporción de observaciones en $\mathcal{L}$ clasificadas en $j \in C$ que están en un nodo t es:*

$$p(j, t) = \frac{N_j(t)}{N}.$$

Definición 6. *La proporción de observaciones que están en un nodo t es:*

$$p(t) = \sum_{j=1}^{k} p(j, t) = \frac{N(t)}{N}$$

Definición 7. *La proporción de observaciones en un nodo t clasificadas en $j \in C$ es:*

$$p(j \mid t) = \frac{N_j(t)}{N(t)}$$

Ahora, supongamos que el árbol definido en la Figura 5 (b) va a ser el usado para clasificar las observaciones. Se puede notar que algunas datos dentro del subconjunto t_3 no son Iris Versicolor. Dado que se necesita buscar la homogeneidad de los nodos del árbol, una métrica cuantitativa de homogeneidad es la noción de impureza, la cual mide la siguiente función:

Definición 8. *Una función:*

$$\begin{aligned} \phi : \mathbb{R}^J &\longrightarrow \mathbb{R} \\ (p_1, p_2, \ldots, p_J) &\longmapsto x \in \mathbb{R}, \end{aligned}$$

tal que $p_j \geq 0$; $j = 1, 2, \ldots, J$ *y* $\sum_{j=1}^{J} p_j = 1$, *es una* ***función de impureza*** *si:*

(i) *La función* ϕ *toma el valor máximo únicamente en el punto* $\left(\frac{1}{J}, \frac{1}{J}, \ldots, \frac{1}{J}\right)$.

(ii) *Para dos números enteros* i, j *en* $\{1, 2, \ldots, J\}$ *se cumple que:* $\phi(\ldots, p_i, \ldots, p_j, \ldots) = \phi(\ldots, p_j, \ldots, p_i, \ldots)$.

(iii) *La función* ϕ *toma el valor mínimo únicamente en los puntos* $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)$.

Teorema 1. *La función:*

$$G(p_1, p_2, \ldots, p_J) = 1 - \sum_{i=1}^{J} (p_i)^2,$$

es una función de impureza (conocida como ***Índice de Impureza de Gini****)*

Demostración:
Sea $(p_1, p_2, \ldots, p_J)$ un vector que cumple con las condiciones dadas en la Definición 8, entonces:

(i) Para hallar el máximo de la función, se va a utilizar el método de multiplicadores de Lagrange. Donde la función objetivo f esta definida como:

$$f(p_1, p_2, \ldots, p_J) = 1 - \textstyle\sum_{i=1}^{J} p_i^2,$$

y la función condición g que restringe a f es:

$$g(p_1, p_2, \ldots, p_J) = p_1 + p_2 + \ldots p_J.$$

Desde luego se puede aplicar Lagrange en $(0,1]^n$ pues en esa región las funciones f, g son continuas y su primera derivada es continua. Entonces, se hallará el punto $(p_1, p_2, \ldots, p_J)$ que satisfaga:

$$\begin{cases} \nabla(f(p_1, p_2, \ldots, p_J)) = \lambda \cdot \nabla(g(p_1, p_2, \ldots, p_J)) & ; \lambda \in \mathbb{R}, \\ g(p_1, p_2, \ldots, p_J) = 1. \end{cases}$$

Reemplazando:

$$\begin{cases} \Big(-2p_1, -2p_2, \ldots, -2p_J\Big) = \lambda \cdot (1, 1, \ldots, 1), \\ p_1 + p_2 + \cdots + p_J = 1. \end{cases}$$

Por lo tanto:

$$-2p_j = \lambda \implies p_j = -\frac{\lambda}{2}; \quad \forall j \in \{1, 2, \ldots, J\}.$$

Así $p_1 = p_2 = \cdots = p_J = -\frac{\lambda}{2}$, entonces:

$$p_1 + p_2 + \cdots + p_J = 1 \implies p_1 = \frac{1}{J}.$$

Dando como conclusión que $\left(\frac{1}{J}, \frac{1}{J}, \cdots, \frac{1}{J}\right)$ es un punto máximo o mínimo. Sin embargo, dicho punto es máximo pues se cumple que $G\left(\frac{1}{J}, \frac{1}{J}, \cdots, \frac{1}{J}\right) > 0$ si $J > 1$, y además en la condición (iii) se mostrará un punto que en su evaluación en G no sobrepasa al encontrado anteriormente.

(ii) Sean i, j en $\{1, 2, \ldots, J\}$, entonces:

$$G(\ldots, p_i, \ldots, p_j, \ldots) = 1 - (\cdots + p_j + \cdots + p_i + \ldots) = G(\ldots, p_j, \ldots, p_i, \ldots).$$

(iii) La función G es no negativa, pues para cualquier componente p_i del vector $(p_1, p_2, \ldots, p_J)$ se cumple que:

$$0 \leq 1 - \sum_{j=1}^{J} (p_j)^2 \implies 0 \leq G(p_1, \ldots, p_i, \ldots, p_J).$$

Por lo tanto, el mínimo valor posible que puede tomar G es 0. Por otra parte, se tiene que:

$$\begin{aligned} G(1, 0, \ldots, 0) &= 1 - \sum_{i=1}^{J} (p_i)^2 \\ &= 1 - \Big((1)^2 + 0^2 + \cdots + (0)^2\Big) \\ G(1, 0, \ldots, 0) &= 0. \end{aligned}$$

Entonces, la condición 2 implica que:

$$G(1,0,\ldots,0) = G(0,1,\ldots,0) = \cdots = G(0,0,\ldots,1) = 0.$$

Así que los vectores $(1,0,\ldots,0),(0,1,\ldots,0),\cdots,(0,0,\ldots,1)$ son mínimos en G. ■

De manera similar a la anterior prueba, se puede mostrar que el índice de entropía de Shannon también es una función de impureza:

Teorema 2. *La función H de Entropía de Shannon dada por:*

$$H(p_1,p_2,\ldots,p_J) = -\sum_{i=1}^{J} g(p_i),$$

donde $g(p_i)$ esta definida como:

$$g(p_i) = \begin{cases} 0 & si \quad p_i = 0, \\ p_i \log_2(p_i) & si \quad p_i \neq 0. \end{cases}$$

Es una función de impureza.

Demostración: Sea $(p_1,p_2,\ldots,p_J)$ un vector con las condiciones dadas en la definición Definición 8. Entonces, se demostrará que la función H cumple con cada una de las condiciones dadas:

(i) Primeramente, se supondrá que $p_i \neq 0; i \in \{1,2,\ldots,J\}$. Entonces, usando el método de multiplicadores de Lagrange la función f que se quiere optimizar es:

$$f(p_1,p_2,\ldots,p_J) = -\sum_{i=1}^{J} p_i \cdot \log_2(p_i),$$

y la función g que la restringe es:

$$g(p_1,p_2,\ldots,p_J) = p_1 + p_2 + \ldots p_J,$$

donde $p_1 + p_2 + \ldots p_J = 1$.

Desde luego se puede aplicar Lagrange en $(0,1]^J$ pues en esa región las funciones f,g son continuas y su primera derivada es continua.

Entonces, se necesita hallar $(p_1, p_2, \ldots, p_J)$ del sistema de ecuaciones:

$$\begin{cases} \nabla(f(p_1, p_2, \ldots, p_J)) & = \lambda \cdot \nabla(g(p_1, p_2, \ldots, p_J)) \qquad \lambda \in \mathbb{R}, \\ g(p_1, p_2, \ldots, p_J) = 1. \end{cases}$$

Entonces:

$$\begin{cases} \left(\frac{\ln(p_1)+1}{\ln(2)}, \frac{\ln(p_2)+1}{\ln(2)}, \ldots, \frac{\ln(p_J)+1}{\ln(2)}\right) & = \lambda \cdot (1, 1, \ldots, 1) \qquad \lambda \in \mathbb{R}, \\ p_1 + p_2 + \cdots + p_j = 1. \end{cases}$$

Por lo tanto, para cualquier $p_j, j = 1, \ldots, J$:

$$\begin{aligned} \frac{\ln(p_j)+1}{\ln(2)} &= \lambda \\ \ln(p_j) &= \ln(2) \cdot \lambda - 1 \\ p_j &= \exp(\ln(2) \cdot \lambda - 1). \end{aligned}$$

Entonces $p_1 = p_2 = \cdots = p_J = \exp(\ln(2) \cdot \lambda - 1)$, por lo tanto:

$$\begin{aligned} p_1 + p_2 + \cdots + p_j &= 1 \\ p_1 + p_1 + \cdots + p_1 &= 1 \\ p_1 &= \frac{1}{J}. \end{aligned} \tag{1.2.1}$$

Así, $p_1 = p_2 = \cdots = p_J = \frac{1}{J}$, entonces $\left(\frac{1}{J}, \frac{1}{J}, \ldots, \frac{1}{J}\right)$ es un punto crítico de H. Sin embargo, para mostrar que el punto anteriormente encontrado es un máximo de f con la restricción g, se realizará el procedimiento anterior para el vector de probabilidades $(p_1, p_2, \ldots, p_J)$ en el caso donde m componentes del vector son distintos 0 y $m \neq J$.

Ahora bien, sea $p_{k1}, p_{k2}, \ldots, p_{km}$ los componentes del vector tal que $p_{ki} \neq 0; i = 1, \ldots, m$, entonces se desarrolla el método de multiplicadores de Lagrange donde la función f que se quiere optimizar:

$$f(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{m} p_{ki} \cdot \log_2(p_{ki}),$$

y la función g que la restringe es:

$$g(p_1, p_2, \ldots, p_J) = p_1 + p_2 + \ldots p_J$$

donde $p_1 + p_2 + \dots p_J = p_{k1} + p_{k2} + \dots p_{km} = 1$.

Desde luego se puede aplicar Lagrange en $(0,1]^J$ pues en esa región las funciones f, g son continuas y su primera derivada es continua.

Entonces, se necesita hallar $(p_1, p_2, \dots, p_J)$ del sistema de ecuaciones:

$$\begin{cases} \nabla(f(p_1, p_2, \dots, p_J)) & = \lambda \cdot \nabla(g(p_1, p_2, \dots, p_J)) \qquad \lambda \in \mathbb{R}, \\ g(p_1, p_2, \dots, p_J) = 1. \end{cases}$$

Entonces:

$$\begin{cases} \left(\dots, \frac{\ln(p_{k1})+1}{\ln(2)}, \frac{\ln(p_{k2})+1}{\ln(2)}, \dots, 0, \dots, \frac{\ln(p_{km})+1}{\ln(2)}, \dots\right) & = \lambda \cdot (\dots, 1, 1, \dots, 0, \dots, 1, \dots) \qquad \lambda \in \mathbb{R}, \\ p_{k1} + p_{k2} + \dots p_{km} = 1. \end{cases}$$

Por lo tanto, para cualquier $p_{kj}; j \in \{1, 2, \dots, m\}$:

$$\begin{aligned} \frac{\ln(p_{kj}) + 1}{\ln(2)} &= \lambda \\ \ln(p_{kj}) &= \ln(2) \cdot \lambda - 1 \\ p_{kj} &= \exp(\ln(2) \cdot \lambda - 1). \end{aligned}$$

Entonces $p_{k1} = p_{k2} = \dots = p_{km} = \exp(\ln(2) \cdot \lambda - 1)$, por lo tanto:

$$\begin{aligned} p_{k1} + p_{k2} + \dots + p_{km} &= 1 \\ p_{k1} + p_{k1} + \dots + p_{k1} &= 1 \\ p_{k1} &= \frac{1}{m}. \end{aligned} \tag{1.2.2}$$

Así, $p_{k1} = p_{k2} = \dots = p_{km} = \frac{1}{m}$, entonces $\left(\dots, \frac{1}{m}, \frac{1}{m}, \dots, 0, \dots, \frac{1}{m}, \dots\right)$ es un punto crítico de H bajo el supuesto sobre el vector de probabilidad. Sin embargo, el anterior punto encontrado no es un punto máximo, pues se mostrará que:

$$H\left(\dots, \frac{1}{m}, \frac{1}{m}, \dots, 0, \dots, \frac{1}{m}, \dots\right) < H\left(\frac{1}{J}, \frac{1}{J}, \dots, \frac{1}{J}\right)$$

Entonces, se tiene que $m < J$ consecuentemente:

$$\begin{aligned}
\log_2(m) &< \log_2(J) \\
-\log_2\left(\frac{1}{m}\right) &< -\log_2\left(\frac{1}{J}\right) \\
-\frac{m}{m}\log_2\left(\frac{1}{J-m}\right) &< -\frac{J}{J}\log_2\left(\frac{1}{J}\right) \\
-\sum_{i=1}^{m}\frac{1}{m}\log_2\left(\frac{1}{m}\right) &< -\sum_{i=1}^{J}\frac{1}{J}\log_2\left(\frac{1}{J}\right) \\
H\left(\ldots,\frac{1}{m},\frac{1}{m},\ldots,0,\ldots,\frac{1}{m},\ldots\right) &< H\left(\frac{1}{J},\frac{1}{J},\ldots,\frac{1}{J}\right).
\end{aligned}$$

Por lo tanto el punto $(\frac{1}{J},\frac{1}{J},\ldots,\frac{1}{J})$ es máximo en la función H.

(ii) Se mostrará que para un vector de probabilidad $(p_1,p_2,\ldots,p_J)$ cualquiera se cumple que:

$$H(\ldots,p_i,\ldots,p_j,\ldots) = H(\ldots,p_j,\ldots,p_i,\ldots)$$

Para todo i,j en $\{1,2,\ldots,J\}$, entonces usando la conmutatividad de la suma:

$$\begin{aligned}
H(\ldots,p_i,\ldots,p_j,\ldots) &= \cdots - p_i\cdot\log_2 p_i - \cdots - p_j\cdot\log_2 p_j - \ldots \\
&= \cdots - p_j\cdot\log_2 p_j - \cdots - p_i\cdot\log_2 p_i - \ldots \\
H(\ldots,p_i,\ldots,p_j,\ldots) &= H(\ldots,p_j,\ldots,p_i,\ldots).
\end{aligned}$$

(iii) La función H es no negativa, pues para cualquier componente p_i del vector de probabilidades $(p_1,p_2,\ldots,p_J)$ se cumple que:

$$\begin{cases} 0 & \text{si} \quad p_i = 0, \\ p_i\log_2(p_i) & \text{si} \quad p_i \neq 0. \end{cases}$$

Sin embargo, en cualquier caso los resultados son no negativos, entonces una suma finita como la que es expuesta en la definición de H debe ser también no negativa, esto es:

$$H(p_1,p_2,\ldots,p_J) \geq 0.$$

Por lo tanto, el mínimo valor posible que puede tomar H es 0. Por otra parte, se tiene que:

$$H(1,0,\ldots,0) = -1\cdot\log_2(1) + 0 + \cdots + 0 = 0.$$

Entonces, la condición 2 implica que:

$$H(1,0,\ldots,0) = H(0,1,\ldots,0) = \cdots = H(0,0,\ldots,1) = 0.$$

Por lo tanto, los vectores $(1,0,\ldots,0),(0,1,\ldots,0),\cdots,(0,0,\ldots,1)$ son mínimos en H.

Puesto que H cumple las 3 condiciones de la definición de función de impureza entonces H es una función de impureza.

■

En particular, si se tiene el vector de probabilidad $(p, 1-p)$ con $0 \leq p \leq 1$, los valores de las anteriores funciones en ese vector se pueden visualizar como sigue:

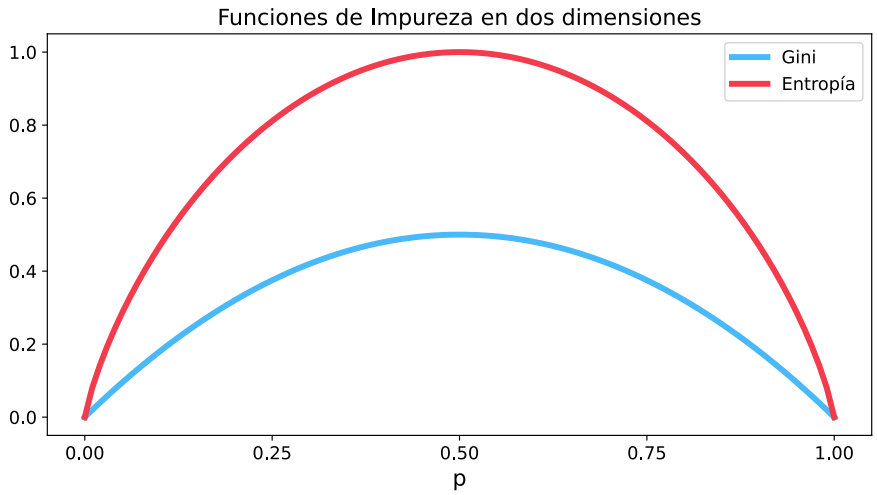


Figura 6: Comparación de funciones de impureza en el vector $(p, 1-p)$, donde p cumple las condiciones de la Definición 8.

En la Figura 6, se puede notar que el punto máximo en ambas funciones se encuentra ubicado cuando $p = \frac{1}{2}$. Esto se debe a que si en un nodo residen dos clases con la misma proporción, entonces en gran medida deja de existir la homogeneidad de las clases.

Ahora bien, la función de impureza aplicada a las proporciones de las clases en los nodos de un árbol, se define como:

Definición 9. *Dada una función de impureza ϕ, se define la medida de impureza $i(t)$ de un nodo t por:*

$$i(t) = \phi(p(1 \mid t), p(2 \mid t), \ldots, p(J \mid t)).$$

Por ejemplo, se hallará la impureza de t_1 y t_2 en la Figura 4. Usando el lenguaje de programación Python, se halló el número de observaciones en cada nodo: en el nodo t_1 hay 50 datos con clase Iris Setosa; en t_2 hay 50 datos con la flor Versicolor y otros 50 con Virginica. Entonces, haciendo la asociación númerica de las clases: 1 corresponde a Setosa, 2 corresponde a Versicolor y 3 corresponde a Virginica, y aplicando la función de impureza de Gini. Se obtiene:

$$i(t_1) = G(p(1 \mid t_1), p(2 \mid t_1), p(3 \mid t_1)) = 1 - \left(\left(\frac{50}{50}\right)^2 + 0 + 0\right) = 0,$$

$$i(t_2) = G(p(1 \mid t_2), p(2 \mid t_2), p(3 \mid t_2)) = 1 - \left(0 + \left(\frac{50}{100}\right)^2 + \left(\frac{50}{100}\right)\right)^2 = 0.5.$$

Como se tenia previsto, los cálculos anteriores muestran que el nodo t_1 es más homogéneo en las observaciones que el nodo t_2. Usualmente, a los nodos con impureza baja se les dice **puros**; en cambio, los nodos con mayor impureza son **impuros**.

Con la anterior función se tiene una forma de medir la homogeneidad que hay en un nodo. A continuación, se mostrará una función que mide el rendimiento de una partición observando la pureza de los nodos resultantes.

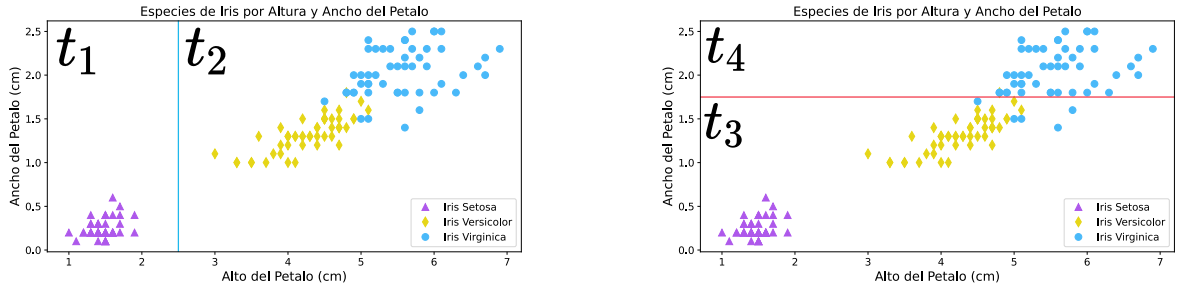
Definición 10. *Sea t un nodo y s una condición (pregunta) que provoca una división de t en dos nodos t_L,t_R. Se define la **disminución de la impureza** como:*

$$\Delta i(s, t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R),$$

donde $p_L = \frac{p(t_L)}{p(t)}$, $p_R = \frac{p(t_R)}{p(t)}$.

Con ayuda de la anterior función se puede comparar el rendimiento de dos o mas preguntas en el mismo nodo t, solo basta con evaluarlas y escoger la que tenga mayor valor bajo la función.

Para mostrar un ejemplo de la anterior definición, se hallará la disminución de la impureza en las siguientes divisiones del mismo nodo $\mathcal{L}$.



(a) División de $\mathcal{L}$ basado en el alto del pétalo

(b) División de $\mathcal{L}$ basado en el ancho del pétalo

Figura 7: Dos divisiones para el nodo $\mathcal{L}$

Primeramente, se calculó en Python que: en el nodo t_1 hay 50 flores Setosa; en t_2 hay 50 Versicolor y 50 Virginica; por otro lado, en t_3 hay 50 Setosas, 49 versicolores y 5 Virginicas; y en t_4 hay 45 flores Virginica y una Versicolor.

Usando como función de impureza el índice de Gini sobre los datos representados en la Figura 4, se halló la reducción de impureza en cada caso:

- Para la condición s_1 en la variable de alto del pétalo que dio como resultado los nodos t_1, t_2:

$$\begin{aligned}\Delta i(s_1, \mathcal{L}) &= i(\mathcal{L}) - \frac{p(t_1)}{p(\mathcal{L})} \cdot i(t_1) - \frac{p(t_2)}{p(\mathcal{L})} \cdot i(t_2) \\ &= 0.6667 - \left(\frac{50}{150}\right) \cdot (0) - \left(\frac{100}{150}\right) \cdot (0.5) \\ \Delta i(s_1, \mathcal{L}) &= 0.3334.\end{aligned}$$

- Para la condición s_2 en la variable ancho del pétalo que dio como resultado t_3, t_4:

$$\begin{aligned}\Delta i(s_2, \mathcal{L}) &= i(\mathcal{L}) - \frac{p(t_3)}{p(\mathcal{L})} \cdot i(t_3) - \frac{p(t_4)}{p(\mathcal{L})} \cdot i(t_4) \\ &= 0.6667 - \left(\frac{104}{150}\right)\left(1 - \left(\frac{50}{104}\right)^2 - \left(\frac{49}{104}\right)^2 - \left(\frac{5}{104}\right)^2\right) - \left(\frac{46}{150}\right)\left(1 - \left(\frac{45}{46}\right)^2 - \left(\frac{1}{46}\right)^2\right) \\ \Delta i(s_2, \mathcal{L}) &= 0.2761\end{aligned}$$

Como se puede observar la disminución de la impureza es mas alta con la división que da como resultado t_1, t_2, que con la división que da como resultado t_3, t_4. Pues, una división logra separar toda una clase, mientras que la otra no lo hace muy bien.

La función de reducción de impureza es una pieza importante en la construcción de árboles, pues esta juzga la utilidad de una división en un nodo. No obstante, la fórmula dada en Definición 10 tiene un origen del cual se hablará a continuación, partiendo desde la siguiente definición.

Definición 11. *Sea T un árbol. Se denotará por $\widetilde{T}$ el **conjunto de nodos terminales** (hojas) de T.*

A su vez, para construir la función de disminución de impureza, es necesario observar la impureza de las hojas en el árbol. Para ello se define un promedio de las impurezas en los nodos terminales del árbol:

Definición 12. *La **impureza de un árbol** esta dada por:*

$$I(T) = \sum_{t \in \widetilde{T}} I(t),$$

donde $I(t) = i(t) \cdot p(t)$.

Entonces, si se tienen dos árboles T, T' donde sus hojas están marcadas como t y t' tal como se ve en la siguiente figura:

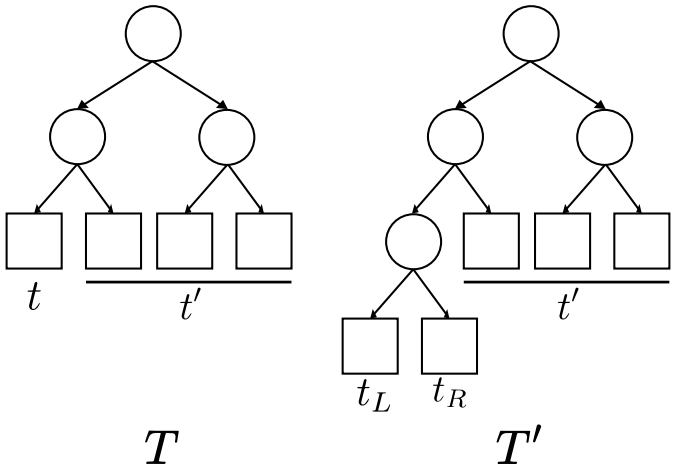


Figura 8: A la izquierda un árbol T. A derecha una extensión del árbol T notada como T'

Se puede esperar que si la división que separó el nodo t es buena, la impureza del árbol $I(T')$ es menor que $I(T)$. Entonces usando la Definición 12:

$$\begin{cases} I(T) = I(t) + \sum_{t' \in \widetilde{T} - \{t\}} I(t'), \\ I(T') = I(t_L) + I(t_R) + \sum_{t' \in \widetilde{T} - \{t\}} I(t') \end{cases}$$
$$\implies I(T) - I(T') = I(t) - I(t_L) - I(t_R).$$

Con la idea anterior, las divisiones óptimas deben maximizar $I(T) - I(T')$. Lo cual da cabida a la siguiente definición:

Definición 13. *Sea t un nodo y sea $\mathtt{s}$ una pregunta que particiona el nodo t en dos nodos t_L, t_R. Entonces, el cambio total de impureza debido a s está dado por:*

$$\Delta I(\mathtt{s}, t) = I(t) - I(t_L) - I(t_r).$$

De lo anterior se puede notar que:

$$\begin{aligned} \Delta I(\mathtt{s}, t) &= i(t) \cdot p(t) - i(t_L) \cdot p(t_L) - i(t_R) \cdot p(t_R) \\ &= i(t) \cdot p(t) - i(t_L) \cdot p(t_L) \cdot \frac{p(t)}{p(t)} - i(t_R) \cdot p(t_R) \cdot \frac{p(t)}{p(t)} \\ &= p(t) \left(i(t) - \frac{p(t_L)}{p(t)} \cdot i(t_L) - \frac{p(t_R)}{p(t)} \cdot i(t_R) \right) \\ &= p(t) \left(i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R) \right) \\ \Delta I(\mathtt{s}, t) &= p(t) \cdot \Delta i(\mathtt{s}, t). \end{aligned}$$

Por lo tanto la pregunta s que maximiza $\Delta I(s,t)$ también maximiza $\Delta i(s,t)$.

Cabe agregar que la función de reducción de impureza es siempre mayor o igual que 0. En (Breiman et al., 1984, p. 126) hay una breve prueba de ello bajo la suposición de que la función de impureza (Definición 8) es cóncava en el espacio de vectores de probabilidad.

1.3. Decidiendo la clase representante en las hojas

En la Figura 4 se observa que los nodos terminales en el árbol conforman una partición del conjunto de variables independientes X. Dado que en todas las particiones una clase sobresale por su proporción, es usual definir que la clase representante en cada partición sea aquella que tenga mayor población en el nodo. Sin embargo, el anterior criterio puede ser generalizado usando la siguiente función cuyo propósito es evaluar la relación entre clases.

Definición 14. *Sean i y j clases que pertenecen a el conjunto C. Se dice que $C(i \mid j)$ es el costo de clasificar erróneamente la clase j como si fuese la clase i. Además, $C(i \mid j) \geq 0$ y $C(j \mid j) = 0$.*

Por ejemplo, $C(i \mid j) = 1 - \delta_{ij}$ es una función de costo donde:

$$\delta_{ij} = \begin{cases} 1 & \text{si} \quad i = j, \\ 0 & \text{si} \quad i \neq j. \end{cases}$$

Ahora bien, supongamos que en un nodo t de un árbol T se afirma que la clase i representa dicho nodo. Por lo tanto, empleando la Definición 14, se define **el costo esperado de clasificar erróneamente el nodo t como una clase i**, igual a:

$$\sum_j C(i \mid j) \cdot p(j \mid t). \tag{1.3.1}$$

En particular, si $C(i \mid j) = 1 - \delta_{ij}$ entonces:

$$\sum_j C(i \mid j) \cdot p(j \mid t) = \sum_j p(j \mid t) - \sum_j \delta_{ij} p(j \mid t) = 1 - (0 + \cdots + 1 \cdot p(i \mid t) + \cdots + 0) = 1 - p(i \mid t).$$

Se espera que la clase representante de un nodo terminal tenga el menor costo esperado de clasificación en dicho nodo. Lo cual da cabida a la siguiente definición:

Definición 15. *Sea t un nodo. La clase $j^*(t)$ que representa un nodo t está definida como:*

$$j^*(t) = arg\min_i \left(\sum_j C(i \mid j) \cdot p(j \mid t) \right).$$

Por ejemplo, si $C(i \mid j) = 1 - \delta_{ij}$ entonces:

$$j^*(t) = arg\,\min_i\Big(1 - p(i \mid t)\Big) = arg\,\max_i\Big(p(i \mid t)\Big) = arg\,\max_i\left(\frac{N_i(t)}{N(t)}\right) = arg\,\max_i\Big(N_i(t)\Big).$$

En el anterior ejemplo, se obtuvo la regla usual para determinar la clase representante de un nodo; tomar aquella que tiene el mayor número de elementos dentro del mismo. Por otra parte, puede suceder que existan dos o más clases en un nodo t tal que cada una de ellas sea igual a $j^*(t)$. No obstante, en ese caso (Choi, 2017) recomienda usar una regla arbitraria para elegir alguna de dichas clases.

Hasta este punto, se tienen todos los elementos para construir un primer árbol de clasificación con nuestra base de datos. En la siguiente sub-sección se mostrarán los detalles de este proceso.

1.4. Construcción del Primer Árbol de Clasificación

La construcción se llevará a cabo de la siguiente manera: primero, se necesita buscar la variable que forme parte de la pregunta para dividir la raíz $\mathcal{L}$. Para ello, se formularán diferentes preguntas con cada una de las variables. Luego, se tomará la que tenga mayor puntaje en la función de disminución de impureza vista en la Definición 10. Con la primera división obtenida, se realiza de nuevo el procedimiento en los nodos resultantes impuros. Al árbol resultante se le notará como $T_{\text{máx}}$.

Para comenzar, en cada variable se pueden crear infinitas preguntas; no obstante, en este caso las variables de entrada son cuantitativas. Por lo tanto, la estrategia es formular preguntas de la forma:

¿Se cumple qué $\mathit{Variable} \leq a$?

Donde a es un número cualquiera, también conocido como *Umbral* o *Threshold*. Dado que cada umbral produce una partición al nodo, se necesita generar un conjunto finito de umbrales para probar cuál divide mejor el nodo.

Una forma de generar dichos umbrales es de la siguiente manera: supongamos que una variable está compuesta por los datos $\{1, 3, 4, 5\}$, de los cuales se omitieron los duplicados y se organizaron de menor a mayor. Entonces el conjunto de umbrales, son los puntos medios que se pueden generar de la anterior lista. Dando como resultado el conjunto $\{2, 3.5, 4.5\}$.

Por lo tanto, generando el conjunto de umbrales para cada una de las variables y observando la disminución de la impureza en cada uno de los casos. Se encontró que la variable "Alto del Pétalo" con el umbral 2.45 tiene el mayor puntaje. De tal manera, el árbol comenzará de la siguiente forma:

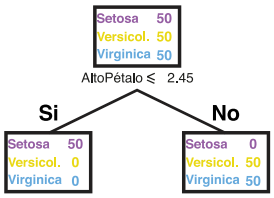


Figura 9: Árbol de decisión para los datos mostrados en la Figura 3

Repitiendo el proceso anterior en las hojas con impureza. Se obtuvo el siguiente resultado.

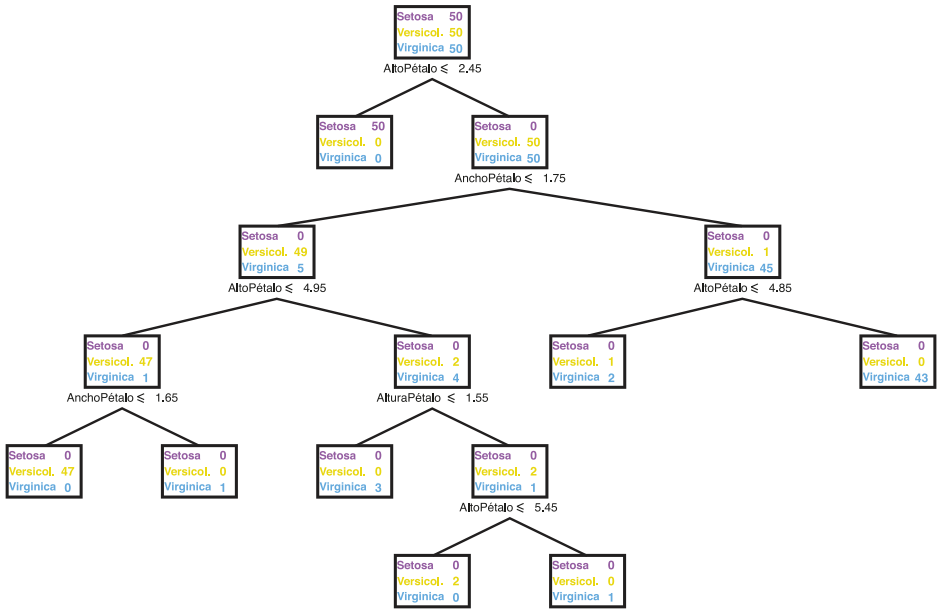


Figura 10: Árbol de decisión $T_{\text{máx}}$ de la base de datos Figura 3.

La siguiente gráfica muestra como las hojas del anterior árbol genera una partición en el conjunto de variables independientes X.

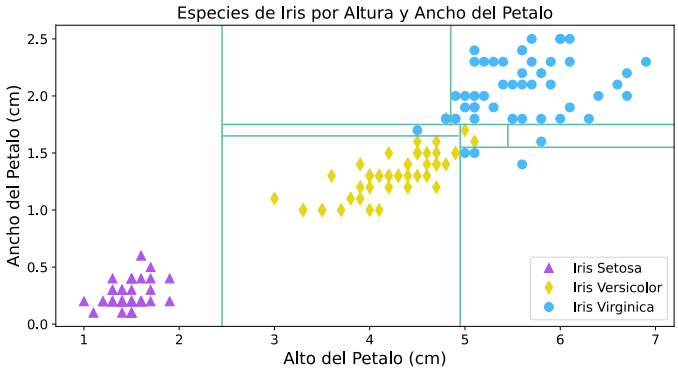


Figura 11: Visualización de la partición de la base de datos causada por las hojas del árbol en la Figura 10

La Figura 11 muestra como el árbol anteriormente construido logra separar las clases en diferentes conjuntos.

Ahora bien, se ha creado el primer árbol de clasificación, notado como $T_{\text{máx}}$, pues es el árbol más grande que se puede construir usando el procedimiento anterior. En él modelo se observa que la mayoría de sus hojas son puras, por lo tanto, la clase representante de cada nodo terminal puede ser la que más alta proporción tiene.

No obstante, el árbol construido en la Figura 10 se ajusta a los datos en $\mathcal{L}$, lo cual indica que el modelo tiene un error de sobreajuste. Para mitigar dicho error se introducirá el podado de un árbol.

En pocas palabras, el término "podar el árbol" hace referencia a encontrar versiones reducidas de $T_{\text{máx}}$. Aunque se pueda pensar que un árbol más pequeño no clasificará los datos de manera correcta, por lo menos reducirá el error de sobre ajuste.

Por consiguiente, para juzgar el rendimiento de los árboles, en la siguiente sección se mostrará una métrica que observa el rendimiento de un árbol. Además se acompañará de algunas definiciones y un teorema que muestra que, al expandir un árbol el error de clasificación es menor.

1.5. El Costo en el Error de Clasificación

Algunos árboles pueden estar sujetos a errores al clasificar los datos. Dado que el objetivo es construir un modelo con un error de clasificación bajo, se necesita una métrica que evalué dicho error en un árbol. Puesto que los nodos terminales del árbol son los que toman la decisión en la clasificación, entonces la siguiente función observará el rendimiento de cualquier nodo para clasificar.

Definición 16. *Sea t un nodo en un árbol T. El costo de clasificación errónea $r(t)$ en el nodo t es:*

$$r(t) = \min_i \left(\sum_j C(i \mid j) \cdot p(j \mid t) \right).$$

Como se decía anteriormente, las hojas del árbol toman la decisión de la clasificación. Por lo tanto, evaluar el error de clasificación de un árbol es evaluar el error con las hojas del mismo. Entonces, para extender la Definición 16 a árboles, se toma $R(t) = r(t) \cdot p(t)$, y con el se define el **costo de clasificación errónea** $R(T)$ de un árbol T como:

$$R(T) = \sum_{t \in \widetilde{T}} R(t) = \sum_{t \in \widetilde{T}} r(t) \cdot p(t). \tag{1.5.1}$$

Con las anteriores definiciones, es fácil mostrar que cualquier división en un nodo hace decrecer el costo de clasificación errónea. En otras palabras, entre más se expanda el árbol, menor error de clasificación tendrá.

Teorema 3. *Sea t un nodo y t_L, t_R los nodos resultantes de una división de t. Entonces*

$$R(t) \geq R(t_l) + R(t_R).$$

Asimismo, la igualdad se obtiene si $j^*(t) = j^*(t_L) = j^*(t_R)$.

Demostración: Para la demostración se supondrá que la etiqueta de clase del nodo t es $j^*(t)$ esto es:

$$r(t) = \min_i \left(\sum_j C(i \mid j) \cdot p(j \mid t) \right) = \sum_j C(j^*(t) \mid j) \cdot p(j \mid t).$$

Entonces:

$$\begin{aligned}
R(t) &= r(t) \cdot p(t) \\
&= \left(\sum_j C(j^*(t) \mid j) \cdot p(j \mid t) \right) p(t) \\
&= \sum_j C(j^*(t) \mid j) \cdot p(j \mid t) \cdot p(t) \\
&= \sum_j C(j^*(t) \mid j) \cdot p(j, t) \\
&= \sum_j C(j^*(t) \mid j) \cdot (p(j, t_L) + p(j, t_R)) \\
R(t) &= \sum_j C(j^*(t) \mid j) \cdot p(j, t_L) + \sum_j C(j^*(t) \mid j) \cdot p(j, t_R).
\end{aligned}$$

Por otro lado se puede afirmar que:

$$\begin{cases} \sum_j C(j^*(t) \mid j) \cdot p(j, t_L) \geq \min_i \sum_j C(i \mid j) \cdot p(j, t_L), \\ \sum_j C(j^*(t) \mid j) \cdot p(j, t_R) \geq \min_i \sum_j C(i \mid j) \cdot p(j, t_R). \end{cases}$$

$$\Longrightarrow \begin{cases} \sum_j C(j^*(t) \mid j) \cdot p(j, t_L) - \min_i \sum_j C(i \mid j) \cdot p(j, t_L) \geq 0, \\ \sum_j C(j^*(t) \mid j) \cdot p(j, t_R) - \min_i \sum_j C(i \mid j) \cdot p(j, t_R) \geq 0. \end{cases}$$

Entonces:

$$\begin{aligned}
R(t) - R(t_L) - R(t_R) = &\sum_j C(j^*(t) \mid j) \cdot p(j, t_L) + \sum_j C(j^*(t) \mid j) \cdot p(j, t_R) - \min_i \sum_j C(i \mid j) \cdot p(j, t_L). \\
&- \min_i \sum_j C(i \mid j) \cdot p(j, t_R) \qquad (1.5.2) \\
R(t) - R(t_L) - R(t_R) &\geq 0 \\
R(t) &\geq R(t_L) + R(t_R).
\end{aligned}$$

Por lo tanto, se demostró la primera desigualdad. Ahora bien, es fácil demostrar que

$R(t) = R(t_l) + R(t_R)$ si $j^*(t) = j^*(t_L) = j^*(t_R)$ pues usando la Definición 15 se puede afirmar que:

$$\begin{cases} \min\limits_i \sum\limits_j C(i \mid j) \cdot p(j, t_L) = \sum\limits_j C(j^*(t_L) \mid j) \cdot p(j, t_L), \\ \min\limits_i \sum\limits_j C(i \mid j) \cdot p(j, t_R) = \sum\limits_j C(j^*(t_R) \mid j) \cdot p(j, t_R). \end{cases}$$

Entonces, reemplazando en la ecuación (1.5.2):

$$\begin{aligned} R(t) - R(t_L) - R(t_R) = &\sum_j C(j^*(t) \mid j) \cdot p(j, t_L) + \sum_j C(j^*(t) \mid j) \cdot p(j, t_R) - \sum_j C(j^*(t_L) \mid j) \cdot p(j, t_L) \\ &- \sum_j C(j^*(t_R) \mid j) \cdot p(j, t_R). \end{aligned}$$

Puesto que $j^*(t) = j^*(t_L) = j^*(t_R)$, entonces:

$$\begin{aligned} R(t) - R(t_L) - R(t_R) = &\sum_j C(j^*(t) \mid j) \cdot p(j, t_L) + \sum_j C(j^*(t) \mid j) \cdot p(j, t_R) - \sum_j C(j^*(t) \mid j) \cdot p(j, t_L) \\ &- \sum_j C(j^*(t) \mid j) \cdot p(j, t_R). \\ R(t) - R(t_L) - R(t_R) &= 0 \\ R(t) &= R(t_L) + R(t_R). \end{aligned}$$

■

Aunque el error de clasificación de un árbol decrezca cuando este se extiende, el sobre ajuste del modelo a la muestra de aprendizaje crece. No obstante, construir el árbol extendido es uno de los primeros pasos para crear nuevos árboles óptimos, pues estos árboles no son más que versiones reducidas que son obtenidas quitando algunas hojas del árbol extendido. A continuación se observará este método a detalle.

2. El Tamaño Adecuado del Árbol

Dado que un árbol de clasificación extendido puede causar problemas de sobre ajuste, su tamaño se debe ser regularizado. Para esto, se crearán nuevos árboles quitando algunas hojas de $T_{\text{máx}}$ con el fin de hacer el árbol más pequeño. El tamaño del nuevo árbol es el tema de discusión en esta sección.

Para comenzar se necesita crear una métrica que penalice el tamaño del árbol. Esto es, entre más grande el árbol, mayor debe ser la penalización. Por lo tanto, se necesita la siguiente definición.

Definición 17. *Sea α un número no negativo. Para cualquier nodo t, se define:*

$$R_\alpha(t) = R(t) + \alpha.$$

Y a su vez se define para un árbol T:

$$R_\alpha(T) = R(T) + \alpha\left|\widetilde{T}\right|.$$

Es la medida $R_\alpha(T)$ la que define hasta qué punto se debe extender el árbol. En comparación de $R(T)$, la medida $R_\alpha(T)$ no necesariamente disminuye cuando se extiende el árbol, pues el término $\left|\widetilde{T}\right|$ crecerá.

Antes, se mencionaba que al árbol $T_{\text{máx}}$ se le debían remover algunas hojas. Para este proceso, se definirán las siguientes notaciones.

Si se tiene un árbol T como en la Figura 12.a) y t_2 es un nodo de T, entonces el sub-árbol T_{t_2} es el árbol formado por todos los descendientes de t_2 incluido el mismo, tal como se ve en la Figura 12.b). Ahora bien, si se quiere cortar el sub-árbol T_{t_2} de T, se removerán todos los descendientes de t_2 en T dando como resultado el árbol notado como $T - T_{t_2}$, ilustrado en la Figura 12.c).

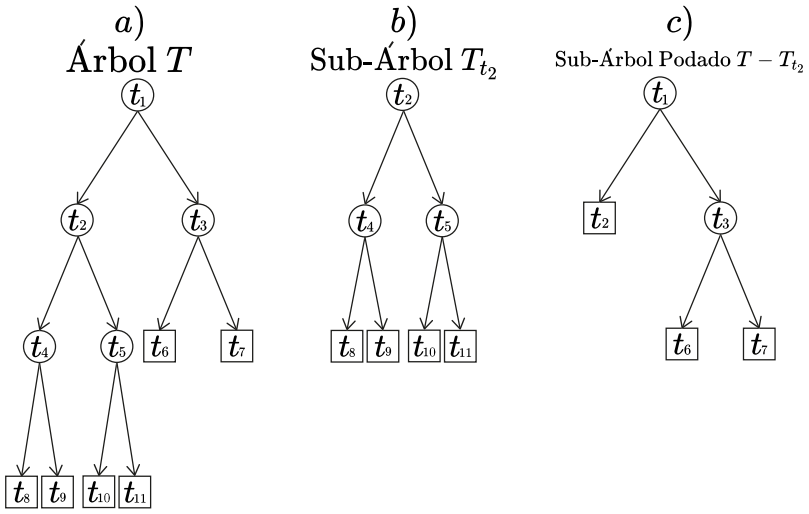


Figura 12: Ejemplo del proceso de podado para el modelo de árbol. Tomado y adaptado de (Breiman et al., 1984, p. 64)

Son los sub-árboles podados aquellos con los que se pueden crear nuevos modelos menos sobre ajustados a comparación de $T_{\text{máx}}$. Pues comparten la misma raíz que el árbol original y no se extienden tanto. Por lo tanto, el proceso de generar un sub-árbol podado puede ser formalizado como sigue:

Definición 18. *Un **sub-árbol podado** T_1 de un árbol T es un sub-árbol obtenido por el siguiente proceso:*

1. *Tomar un nodo t,*

2. *remover todos los descendientes de t en el árbol T, a excepción de t mismo.*

De hecho, si T_1 es un sub-árbol podado de T entonces se notará como $T_1 \prec T$, o $T_1 \preceq T$ en el caso que T_1 pueda ser igual a T.

Ahora bien, se construirá un conjunto de sub árboles podados del árbol extendido $T_{\text{máx}}$; no obstante, dichos sub árboles serán cuidadosamente escogidos. En la siguiente definición se observará que cada árbol del anterior conjunto mencionado es generado con un número α no negativo. Además, tendrán una medida R_α menor a los demás árboles.

Definición 19. *Dado un árbol T y un número α no negativo, el **sub-árbol podado óptimo** de T notado como $T(\alpha)$ debe cumplir las siguientes condiciones:*

1. $T(\alpha) \preceq T$.
2. *Para cualquier sub-árbol podado $T' \preceq T$ se cumple que $R_\alpha(T(\alpha)) \leq R_\alpha(T')$.*
3. *Si un sub-árbol podado $T' \preceq T$ satisface $R_\alpha(T(\alpha)) = R_\alpha(T')$ entonces $T(\alpha) \preceq T'$.*

En consecuencia a la anterior definición, se demostrará la existencia y unicidad de un sub-árbol podado óptimo del árbol T para un α no negativo fijo. Sin embargo, primero se dará una definición y un lema para dicho fin.

Definición 20. *Sea T un árbol tal que $\left|\widetilde{T}\right| > 1$. Donde t_1 es el nodo raíz del árbol T y t_L,t_R los nodos subyacentes a t_1. Se definen las **ramas principales** de T como los sub-árboles T_{t_L}, T_{t_R}.*

Lema 1. *Sea T un árbol con ramas principales T_{t_L}, T_{t_R}. Entonces para cualquier α no negativo se cumple que:*

$$R_\alpha(T) = R_\alpha(T_{t_L}) + R_\alpha(T_{t_R}).$$

Demostración: Sea T un árbol con ramas principales T_{t_L}, T_{t_R}. Entonces:

$$\begin{aligned} R_\alpha(T) &= R(T) + \alpha \cdot \left|\widetilde{T}\right| \\ &= \sum_{t \in \widetilde{T}} r(t) \cdot p(t) + \alpha \cdot \left|\widetilde{T}\right| \\ &= \left(\sum_{t \in \widetilde{T_{t_L}}} r(t) \cdot p(t) + \sum_{t \in \widetilde{T_{t_R}}} r(t) \cdot p(t) \right) + \alpha \cdot \left(\left|\widetilde{T_{t_L}}\right| + \left|\widetilde{T_{t_R}}\right| \right) \\ R_\alpha(T) &= R_\alpha(T_{t_L}) + R_\alpha(T_{t_R}). \end{aligned}$$

■

Teorema 4. *Sea T un árbol y α un número no negativo. El sub-árbol podado óptimo $T(\alpha) \preceq T$ existe y es único.*

Demostración: La demostración se hará con una inducción completa sobre el número de hojas del árbol T. Si $\left|\widetilde{T}\right| = 1$ entonces el árbol solo consiste en el nodo raíz, y por ende $T(\alpha)$ es el nodo raíz. Ahora bien, supongamos que la hipótesis se cumple para todo número natural i tal que $i \leq n$ donde n es un natural cualquiera. Se probará la hipótesis para $n+1$.

Sea $\left|\widetilde{T}\right| = n+1$ y t_1 el nodo raíz de T. Puesto que el número de nodos terminales es mayor que 1, existen los nodos subyacentes t_L, t_R de t_1. Por lo tanto, las ramas principales T_{t_L}, T_{t_R} de T también existen. Dado que $\left|\widetilde{T}\right| = \left|\widetilde{T_{t_L}}\right| + \left|\widetilde{T_{t_R}}\right|$, se puede afirmar:

$$\left|\widetilde{T_{t_L}}\right| < \left|\widetilde{T}\right|, \qquad \left|\widetilde{T_{t_R}}\right| < \left|\widetilde{T}\right|.$$

Usando la hipótesis de inducción sobre los árboles T_{t_L}, T_{t_R}, existen los sub-árboles podados óptimos $T_{t_L}(\alpha), T_{t_R}(\alpha)$ de T_{t_L}, T_{t_R} respectivamente. Ahora bien, se mostrará que el árbol T_1 formado por la raíz t_1 y las ramas principales T_{t_L}, T_{t_R} es igual a $T(\alpha)$. Para probar lo anterior se necesita considerar dos casos:

Supongamos que $R_\alpha(\{t_1\}) \leq R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha))$. Sea $T' \preceq T$ tal que $\left|\widetilde{T'}\right| > 1$, y T'_{t_L}, T'_{t_R} son sus ramas principales. Entonces, usando la Definición 19 se tiene que: $R_\alpha(T_{t_L}(\alpha)) \leq R_\alpha(T'_{t_L})$ y $R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_R})$, por lo tanto:

$$R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_L}) + R_\alpha(T'_{t_R}),$$

usando el Lema 1:

$$R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T').$$

Dado que $R_\alpha(\{t_1\}) \leq R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha))$ entonces:

$$R_\alpha(\{t_1\}) \leq R_\alpha(T').$$

Puesto que no puede haber un sub-árbol podado de $\{t_1\}$ y este último es un sub-árbol podado de T, se afirma que $T(\alpha) = \{t_1\}$.

Por otro lado, si $R_\alpha(\{t_1\}) > R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha))$, entonces $T(\alpha)$ no puede ser el nodo raíz de T. Sea T' un sub-árbol podado de T no trivial, donde T'_{t_L}, T'_{t_R} son sus ramas principales. De tal modo que usando la Definición 19 se tiene: $R_\alpha(T_{t_L}(\alpha)) \leq R_\alpha(T'_{t_L})$ y $R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_R})$, así que:

$$R_\alpha(T_{t_L}(\alpha)) + R_\alpha(T_{t_R}(\alpha)) \leq R_\alpha(T'_{t_L}) + R_\alpha(T'_{t_R}).$$

Por ende, usando el Lema 1:

$$R_\alpha(T_1) \leq R_\alpha(T')$$

Ahora bien, se demostrará que para cualquier sub-árbol podado T' de T tal que $T_1 \prec T'$ entonces $R_\alpha(T_1(\alpha)) \neq R_\alpha(T')$.

Sea T' un sub-árbol podado de T con ramas principales T'_L, T'_R, tal que $T_1 \not\preceq T'$, sin perdida de generalidad se puede asumir que $T'_L \prec T_{t_L}(\alpha)$ y $T_{t_R}(\alpha) \preceq T'_R$, entonces por la Definición 19 se tiene que:

$$R_\alpha\left(T_{t_L}(\alpha)\right) < R_\alpha(T'_L), \qquad R_\alpha\left(T_{t_R}(\alpha)\right) \leq R_\alpha(T'_R).$$

Por lo tanto:

$$R_\alpha\left(T_{t_L}(\alpha)\right) + R_\alpha\left(T_{t_R}(\alpha)\right) < R_\alpha(T'_L) + R_\alpha(T'_R).$$

Así, usando el Lema 1 se tiene:

$$R_\alpha(T_1) < R_\alpha(T').$$

Entonces, se puede concluir que $T_1 = T(\alpha)$.

Puesto que se la hipótesis se cumple para un árbol de $n+1$ hojas, queda demostrado que $T(\alpha)$ existe.

Por otro lado, para demostrar la unicidad supongamos que existe un sub-árbol podado óptimo T' de T distinto de $T(\alpha)$. Por la Definición 19 se tiene que $R_\alpha(T(\alpha)) \leq R_\alpha(T')$ y a su vez $R_\alpha(T') \leq R_\alpha(T(\alpha))$, entonces $R_\alpha(T') = R_\alpha(T(\alpha))$. Lo anterior implica que $T(\alpha) \preceq T'$ y $T' \preceq T(\alpha)$; consecuentemente, $T(\alpha) = T'$. Lo cual contradice la no unicidad de $T(\alpha)$. ■

El anterior teorema mostró que existen árboles generados por un α no negativo, tal que su error de clasificación R_α es menor que otros sub árboles podados. Sin embargo, se puede pensar que al ser α arbitrario, se generará un conjunto infinito de sub árboles podados. No obstante, a partir del árbol óptimo $T(0)$ se pueden una cantidad finita de valores α y, por lo tanto, un número finito de sub - árboles podados óptimos. A continuación se mostrará como construir dichos árboles óptimos.

Primeramente del árbol $T_{\text{máx}}$ se halla el sub-árbol podado óptimo $T(0)$, la forma de hacerlo es la siguiente: sean t_L, t_R los nodos descendientes de un nodo t cualquiera en el árbol $T_{\text{máx}}$, se sabe por el Teorema 3 que $R(t) \geq R(t_L) + R(t_R)$, por lo tanto, se podará de $T_{\text{máx}}$ los nodos t tal que $R(t) = R(t_L) + R(t_R)$, al árbol resultante se le notará T_1. De hecho, el árbol T_1 es igual a $T(0)$, lo cual puede ser postulado como un teorema.

Teorema 5. *El sub-árbol* T_1 *del resultado anterior es igual al sub-árbol podado óptimo* $T(0)$ *de* $T_{\text{máx}}$.

Demostración: Sea $T_{\text{máx}} = T$ Para la demostración solo hace falta mostrar que T_1 satisface la Definición 19 y por unicidad de $T(0)$ se llegará que $T_1 = T(0)$

Desde luego T_1 cumple que $T_1 \preceq T$, pues su construcción es derivada del podado del árbol.

Ahora bien, se mostrará que para cualquier sub-árbol podado $T' \preceq T$ se tiene la desigualdad $R(T_1) \leq R(T')$. Entonces, sea T' un sub-árbol podado de T, no necesariamente este debe ser un sub-árbol de T_1 o viceversa.

Los nodos terminales de los árboles T_1, T' pueden satisfacer alguna de las siguientes condiciones:

En el primer caso, puede existir un sub-árbol (no trivial) de T' donde la raíz es un nodo terminal x de T_1 y sus nodos terminales $y_{11}, y_{12}, \ldots$ pertenezcan al conjunto de nodos terminales de T'. Por lo tanto, usando la hipótesis en la construcción de T_1 se tiene que:

$$R(x) = R(y_{11}) + R(y_{12}) + \ldots$$

Por otro lado, puede ocurrir lo contrario a lo anterior, esto es que puede existir un sub-árbol (no trivial) de T_1 donde la raíz es un nodo terminal y de T' y sus nodos terminales $x_{21}, x_{22}, \ldots$ pertenezcan al conjunto de nodos terminales de T_1, entonces usando la condición sobre la construcción de T_1 se tiene que:

$$R(y) > R(x_{21}) + R(x_{22}) + \ldots$$

Juntando los dos casos anteriores se debe cumplir que:

$$R(T') \geq R(T_1)$$

Similar a la anterior demostración sí para un sub-árbol podado de T' distinto de T_1 ocurre que $R(T_1) = R(T')$, entonces se puede observar que $T_1 \leq T'$. ■

Ahora bien, el propósito ahora es tratar de generalizar el teorema anterior con el fin de podar árboles y que sean equivalentes a un árbol $T(\alpha)$. Para ello, se mostrará el siguiente teorema:

Teorema 6. *Para cualquier nodo no terminal* $t \in T(0)$ *se cumple que:*

$$R(T_t) < R(t).$$

Demostración: Sea t un nodo cualquiera no terminal en $T(0)$. Se hará una inducción completa sobre el número de nodos terminales de T_t, entonces:

Sí $\left|\widetilde{T}_t\right| = 2$, entonces se notará a los dos nodos terminales de T_t como t_L, t_R. Por lo tanto, usando el Teorema 3 se tiene que:

$$R(t) \geq R(t_L) + R(t_R).$$

Sin embargo, dado que t tiene descendencia entonces por la construcción de $T(0)$ se tiene que $R(t) > R(t_L) + R(t_R)$.

Ahora bien, se supone que para cualquier natural $i > 2$ tal que $i \leq n$ donde n es un natural cualquiera se cumple la condición.

Entonces, se probará que la hipótesis se cumple para $\left|\widetilde{T}_t\right| = n + 1$. Sea T_L, T_R las ramas principales de T_t, puesto que $\left|\widetilde{T}_t\right| = \left|\widetilde{T_R}\right| + \left|\widetilde{T}_L\right|$ entonces se puede afirmar que:

$$\left|\widetilde{T}_L\right| < \left|\widetilde{T}_t\right|, \qquad \left|\widetilde{T_R}\right| < \left|\widetilde{T}_t\right|.$$

Por lo tanto, usando la hipótesis de inducción sobre T_L, T_R se tiene que:

$$R(T_L) < R(t_L), \qquad R(T_R) < R(t_R).$$

Donde t_L, t_R son las raíces de T_L, T_R respectivamente (o los descendientes de t), entonces:

$$\begin{aligned} R(T_L) + R(T_R) &< R(t_L) + R(t_R) \\ R(T_t) &< R(t_L) + R(t_R) \\ R(T_t) &< R(t) \qquad \text{(Teorema 3)}. \end{aligned}$$

■

El anterior teorema muestra que para cualquier nodo no terminal t en $T(0)$, el error de clasificación del nodo es mayor que el error del sub-arból T_t. Entonces, se espera que del anterior teorema también se cumpla para un nodo t del árbol $T(0)$ que:

$$R_\alpha(T_t) < R_\alpha(t). \tag{2.0.1}$$

De ser verdad la anterior desigualdad, se tiene:

$$\begin{aligned} R(T_t) + \alpha\left|\widetilde{T}_t\right| &< R(t) + \alpha \\ \alpha &< \frac{R(t) - R(T_t)}{\left|\widetilde{T}_t\right| - 1}. \end{aligned} \tag{2.0.2}$$

Ahora bien, si α toma valores muy grandes, habrá nodos que no satisfagan la desigualdad (2.0.2). Para encontrar dichos nodos, se definirá la siguiente función $g_1(t)$:

$$g_1(t) = \begin{cases} \frac{R(t)-R(T_t)}{\left|\widetilde{T}_t\right|-1} & si \quad t \notin \widetilde{T(0)}, \\ \infty & si \quad t \in \widetilde{T(0)}. \end{cases} \tag{2.0.3}$$

Entonces, se define:

$$\alpha_2 = \min_{t \in T(0)} g_1(t).$$

Por lo tanto, si un número α cumple que $\alpha < \alpha_2$, la desigualdad (2.0.2) se satisface para todos los nodos $t \in T(0)$ no terminales. En otra mano, sí $\alpha = \alpha_2$, existirán algunos nodos que conviertan la inecuación (2.0.1) en una igualdad, dichos nodos se llamarán ***nodos de enlace más débiles***.

En consecuencia, sí t_1' es un nodo de enlace débil en $T(0)$, entonces:

$$g_1(t_1') = \alpha_2 = \min_{t \in T(0)} g_1(t) \tag{2.0.4}$$

Por otro lado, de (2.0.4) se puede concluir que $R_{\alpha_2}\left(T_{t_1'}\right) = R_{\alpha_2}\left(t_1'\right)$, entonces:

$$\begin{aligned}
R_{\alpha_2}\Big(T(0)\Big) &= \sum_{t\in\widetilde{T(0)}} r(t)\cdot p(t) + \alpha_2\left|\widetilde{T(0)}\right| \\
&= \sum_{t\in\widetilde{T(0)},\, t\notin\widetilde{T_{t_1'}}} r(t)\cdot p(t) + \alpha_2\left(\left|\widetilde{T(0)}\right| - \left|\widetilde{T_{t_1'}}\right|\right) + \sum_{t\in\widetilde{T_{t_1'}}} r(t)\cdot p(t) + \alpha_2\left(\left|\widetilde{T_{t_1'}}\right|\right) \\
&= \sum_{t\in\widetilde{T(0)},\, t\notin\widetilde{T_{t_1'}}} r(t)\cdot p(t) + \alpha_2\left(\left|\widetilde{T(0)}\right| - \left|\widetilde{T_{t_1'}}\right|\right) + R_{\alpha_2}\left(T_{t_1'}\right) \\
&= \sum_{t\in\widetilde{T(0)},\, t\notin\widetilde{T_{t_1'}}} r(t)\cdot p(t) + \alpha_2\left(\left|\widetilde{T(0)}\right| - \left|\widetilde{T_{t_1'}}\right|\right) + R_{\alpha_2}\left(t_1'\right) \\
&= \sum_{t\in\widetilde{T(0)},\, t\notin\widetilde{T_{t_1'}}} r(t)\cdot p(t) + \alpha_2\left(\left|\widetilde{T(0)}\right| - \left|\widetilde{T_{t_1'}}\right|\right) + r\left(t_1'\right)\cdot p\left(t_1'\right) + \alpha_2 \\
R_{\alpha_2}\Big(T(0)\Big) &= R_{\alpha_2}\left(T(0) - T_{t_1'}\right).
\end{aligned}$$

Por lo tanto, el árbol $T_2 = T(0) - T_{t_1'}$ será parte del conjunto de árboles podados óptimos. De hecho $T_2 = T(\alpha_2)$.

Además, si se repite el anterior proceso con el árbol T_2, se obtendrá un α_3 y un sub-árbol de $T(0)$ notado como $T_3 = T(\alpha_3)$. Consecutivamente, se genera el conjunto de árboles:

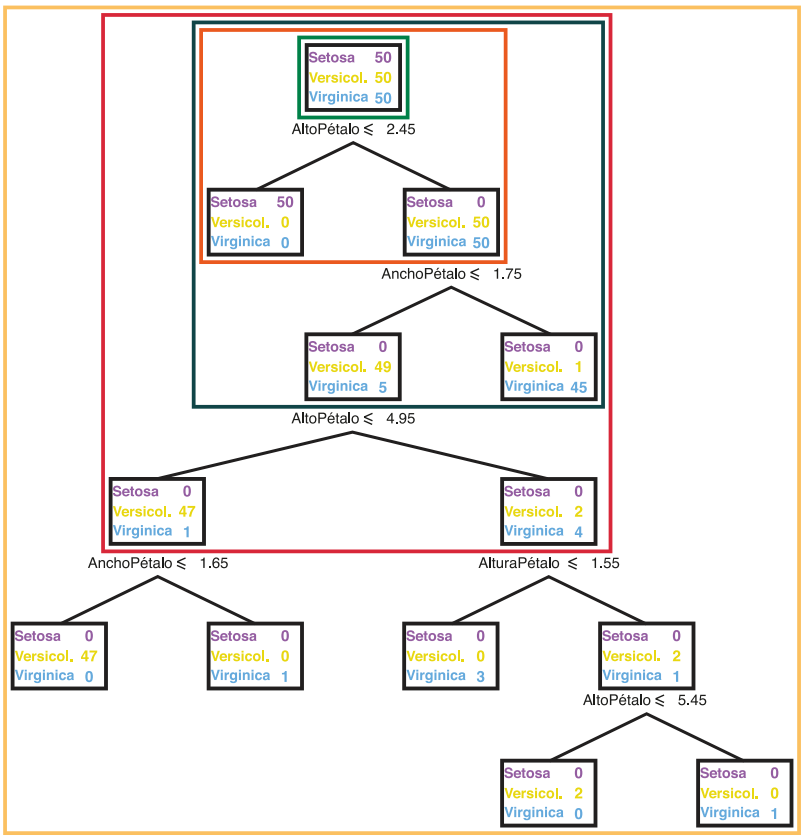
$$T(0) = T_1 \geq T_2 \geq T_3 \geq \cdots \geq \{t_1\}.$$

Y la relación entre los números:

$$0 = \alpha_1 < \alpha_2 < \alpha_3 < \ldots.$$

Ahora bien, usando lo visto en esta sección en la base de datos Iris, se obtuvo el siguiente conjunto

de sub-árboles podados óptimos:



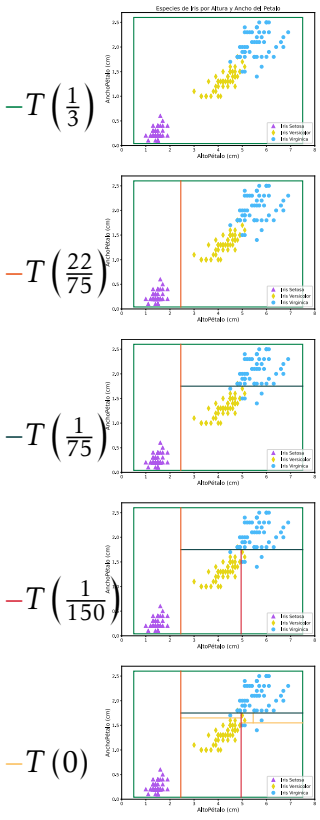


Figura 13: A izquierda, los sub-árboles optimos. A derecha, las divisiones que cada uno de los sub-árboles construyen respectivamente.

Por ende, solo hace falta observar cuál de los árboles $T_1, T_2, \ldots, \{t_1\}$ tiene el mejor rendimiento. Para ello se utilizará la validación cruzada. Sin embargo, el funcionamiento de dicho método depende del siguiente teorema:

Teorema 7. *Sea* $T_k = T(\alpha_k)$

1. *Si* $\alpha_1 \leq \alpha_2$ *entonces* $T(\alpha_2) \leq T(\alpha_1)$.
2. *Si existe* α *tal que* $\alpha_k \leq \alpha < \alpha_{k+1}$ *entonces* $T(\alpha_k) = T(\alpha)$.

Demostración:

1. Sea T un árbol y α_1, α_2 dos números no negativos tal que $\alpha_1 \leq \alpha_2$, entonces por el Teorema 4 se tiene que los árboles $T(\alpha_1), T(\alpha_2)$ existen. Por lo tanto, para la demostración se hará inducción sobre el número de hojas de $T(\alpha_1)$ y se mostrará que se cumple que $T(\alpha_2)$.

Entonces, si $\left|\widetilde{T(\alpha_1)}\right| = 1$ se tiene que $T(\alpha_1) = \{t_1\}$. Consecuentemente, usando la Definición 19 de su-árbol podado óptimo se afirma que para cualquier sub-árbol podado T' de T se cumple que:

$$R_\alpha(\{t_1\}) \leq R_\alpha(T').$$

Por lo tanto:

$$\begin{aligned} R_{\alpha_2}(\{t_1\}) &= R(\{t_1\}) + \alpha_2 \cdot 1 \\ &= R(\{t_1\}) + \alpha_2 + (\alpha_1 - \alpha_1) \\ &= R(\{t_1\}) + \alpha_1 + (\alpha_2 - \alpha_1) \\ &= R_{\alpha_1}(\{t_1\}) + (\alpha_2 - \alpha_1) \\ &\leq R_{\alpha_1}(T') + (\alpha_2 - \alpha_1). \end{aligned}$$

Puesto que por hipótesis $\alpha_1 \leq \alpha_2$, entonces $0 \leq (\alpha_2 - \alpha_1) \leq \left|\widetilde{T'}\right|(\alpha_2 - \alpha_1)$, así:

$$\begin{aligned} &\leq R_{\alpha_1}(T') + \left|\widetilde{T'}\right|(\alpha_2 - \alpha_1) \\ &\leq \left(R(T') + \alpha_1\left|\widetilde{T'}\right|\right) + \left|\widetilde{T'}\right|\alpha_2 - \left|\widetilde{T'}\right|\alpha_1 \\ &\leq R(T') + \alpha_2\left|\widetilde{T'}\right| \\ R_{\alpha_2}(\{t_1\}) &\leq R_{\alpha_2}(T'). \end{aligned}$$

Se acabó de mostrar que la raíz $\{t_1\}$ cumple con la segunda condición de la Definición 19 de sub-árbol podado óptimo con $\alpha = \alpha_2$, y además dicha raíz también cumple con la condición 3 pues para cualquier sub-árbol podado $T' \preceq T$ distinto de la raíz tal que $R_{\alpha_2}(\{t_1\}) = R_{\alpha_2}(T')$, se cumplirá que $\{t_1\} \preceq T'$, pues la ráiz es la misma para cualquier sub-arbol podado. Por lo tanto $T(\alpha_2) = \{t_1\}$, y consecuentemente $T(\alpha_2) = T(\alpha_1)$. Entonces la hipótesis se cumple para $\left|\widetilde{T(\alpha_1)}\right| = 1$.

Ahora bien, supongamos que la hipótesis $\left|\widetilde{T(\alpha_1)}\right| = n$ se cumple para todo número natural i tal que $i \leq n$ donde n es un natural cualquiera, entonces se probará la hipótesis para $n+1$, esto es:

Sea $\left|\widetilde{T(\alpha_1)}\right| = n+1$ y $T_L(\alpha_1), T_R(\alpha_1)$ las ramas principales de $T(\alpha_1)$ donde T_L, T_R son las ramas principales de T. Puesto que $\left|\widetilde{T(\alpha_1)}\right| > 1$ y:

$$\left|\widetilde{T_L(\alpha_1)}\right| < \left|\widetilde{T(\alpha_1)}\right|, \qquad \left|\widetilde{T_R(\alpha_1)}\right| < \left|\widetilde{T(\alpha_1)}\right|.$$

Entonces, usando la hipótesis de inducción sobre $T_L(\alpha_1), T_R(\alpha_1)$ se puede afirmar que:

$$T_L(\alpha_2) \leq T_L(\alpha_1), \qquad T_R(\alpha_2) \leq T_R(\alpha_1).$$

Desde luego $T(\alpha_2)$ es el sub-árbol podado óptimo con raíz $\{t_1\}$ y ramas principales $T_L(\alpha_2), T_R(\alpha_2)$, entonces por las desigualdades anteriores se puede inferir que $T(\alpha_2) \leq T(\alpha_1)$, por otro lado puede ocurrir que $T(\alpha_2) = \{t_1\}$ pero claramente la desigualdad se mantiene, pues la raíz esta presente en todos los sub-árboles podados de T.

2. La segunda condición se demostrará por reducción al absurdo. Supongamos que para cualesquiera $\alpha_k, \alpha, \alpha_{k+1}$ números no negativos se cumple que $\alpha_k \leq \alpha < \alpha_{k+1}$, y además se supone que $T(\alpha_k) \neq T(\alpha)$. Por lo tanto, usando lo demostrado en el item anterior se tiene que $T(\alpha) < T(\alpha_k)$.

Entonces, debe existir un nodo t' en $T(\alpha_k)$ tal que el sub-árbol $T(\alpha_k)_{t'}$ con raíz t' que contiene a todos los nodos descendientes esté podado en el árbol $T(\alpha)$.

Sin embargo, si dicho sub-árbol de $T(\alpha_k)$ está podado de $T(\alpha)$ debe ocurrir lo siguiente:

$$\alpha_{k+1} = \min_{t \in T(\alpha_k)} g_1(t) \leq \alpha.$$

Por lo tanto $\alpha_{k+1} \leq \alpha$ lo cual es una contradicción. ■

A continuación, se observará qué árbol ilustrado en la Figura 13 tiene el mejor rendimiento. Se usará el Teorema 7 para mostrar el método de validación cruzada en la siguiente sección.

3. Seleccionando al mejor árbol podado

Para evaluar el rendimiento de un árbol, se utilizan dos subconjuntos disjuntos de los datos: un conjunto de entrenamiento y un conjunto de prueba. Con el conjunto de entrenamiento se construye el clasificador (en este caso el árbol); y, por otro lado, el conjunto de prueba se evalúa en el árbol anterior para observar la precisión del clasificador. Por lo tanto, se definirán algunas cantidades teóricas asociadas al clasificador d formado por el árbol:

Definición 21. *La probabilidad teórica de clasificar la clase j como una clase i está dada por:*

$$Q^*(i \mid j) = P\left(d(X) = i \mid C = j\right).$$

Definición 22. *El valor esperado teórico para el costo del error de clasificación de la clase j es:*

$$R^*(j) = \sum_i C(i \mid j)\, Q^*(i \mid j).$$

Definición 23. *El costo general teórico del clasificador d, está dado por:*

$$R^*(d) = \sum_j R^*(j)\, \pi(j).$$

Para esta última definición, cabe resaltar la siguiente observación. Si $C(i \mid j) = 1 - \delta_{ij}$ entonces:

$$
\begin{aligned}
R^*(d) &= \sum_j R^*(j)\,\pi(j) \\
&= \sum_j \left(\sum_i C(i \mid j)\, Q^*(i \mid j) \right) \pi(j) \\
&= \sum_j \sum_i (1 - \delta_{ij})\, Q^*(i \mid j)\, \pi(j) \\
&= \sum_j \sum_i Q^*(i \mid j)\pi(j) - \sum_j \sum_i \delta_{ij} Q^*(i \mid j)\pi(j) \\
&= \sum_j \sum_i Q^*(i \mid j)\, \pi(j) - \sum_i Q^*(i \mid i)\, \pi(i) \\
&= \sum_j \pi(j) - \sum_i Q^*(i \mid i)\, \pi(i) \\
&= \sum_i \pi(i) - \sum_i Q^*(i \mid i)\, \pi(i) \\
&= \sum_i (1 - Q^*(i \mid i))\, \pi(i) \\
&= \sum_i P(d(\mathcal{X}) \neq Y \mid Y = i)\, \pi(i) \\
&= \sum_i P(d(\mathcal{X}) \neq Y \mid Y = i)\, P(Y = i) \\
&= \sum_i P(d(\mathcal{X}) \neq Y \cap Y = i) \\
R^*(d) &= P(d(\mathcal{X}) \neq Y)
\end{aligned}
$$

De lo anterior, se concluye que el error de costo general teórico con $C(i \mid j) = 1 - \delta_{ij}$, no es más que la probabilidad de que el árbol cometa errores al clasificar. Con lo anterior en mente se procederá a mostrar el método de validación cruzada.

3.1. Validación Cruzada

Para usar la validación cruzada de V-iteraciones, se dividen los datos $\mathcal{L}$ en V conjuntos disjuntos $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_V$. Se notará $\mathcal{L}^{(v)} = \mathcal{L} - \mathcal{L}_v$ donde $v \in \{1, 2, \ldots, V\}$.

Primeramente, usando $\mathcal{L}$ hacemos crecer el árbol $T_{\text{máx}}$ para luego obtener los valores $0 = \alpha_1 < \alpha_2 < \ldots$ y sus respectivos árboles podados $T_1 \geq T_2 \geq \ldots \{t_1\}$ donde $T_k = T(\alpha_k)$. Desde luego los árboles $T_1, T_2, \ldots$ no se pueden utilizar en la validación cruzada, pues se empleó la muestra de aprendizaje para construirlos.

Entonces, se define $\alpha'_k = \sqrt{\alpha_k \cdot \alpha_{k+1}}$. Para cada $v \in \{1, 2, \ldots, V\}$ y $\alpha \in \{\alpha'_1, \alpha'_2, \ldots\}$, se usará $\mathcal{L}^{(v)}$ para hacer crecer $T^{(v)}_{\text{máx}}$ y luego se construirá $T^{(v)}(\alpha)$. Un algoritmo para construir $T^{(v)}(\alpha)$ puede ser visto en (Ripley, 1996, Proposición 7.2 en p. 223).

Dado que $\mathcal{L}_v$ no se usó para formar los árboles $T^{(v)}(\alpha)$, entonces se usará como conjunto de prueba en el clasificador.

Usando $\mathcal{L}_v$ como conjunto de prueba en los árboles $T^{(v)}(\alpha)$, se calculará:

- $N^{(v)}_{ij}$: El número de clases j en $\mathcal{L}_v$ clasificados por $T^{(v)}(\alpha)$ como i.
- $N_{ij} = \sum_i N^{(v)}_{ij}$: El número total de clases j que fueron clasificadas como i en el proceso de validación cruzada.

Ahora bien, se espera que para un valor de V grande, $T^{(v)}(\alpha)$ y $T(\alpha)$ tengan una precisión similar. Entonces:

$$
\begin{aligned}
Q^{CV}(i \mid j) &= \frac{N_{ij}}{N_j} \approx Q^*(i \mid j), \\
R^{CV}(j) &= \sum_i C(i \mid j) Q^{CV}(i \mid j) \approx R^*(j), \\
R^{CV}\left(T(\alpha)\right) &= \sum_j R^{CV}(j)\, \pi(j) \approx R^*\left(T(\alpha)\right).
\end{aligned}
$$

Por lo tanto, usando el Teorema 7 se tiene que $T_{\alpha' k} = T_{\alpha_k} = T_k$. Entonces el estimado de $R^{CV}(T_k)$ está dado por:

$$R^{CV}(T_k) = R^{CV}(T_{\alpha_k}).$$

Finalmente, el sub árbol podado óptimo $T_{k'}$ será alguno de los árboles $T_1, T_2, \ldots, \{t_1\}$ tal que:

$$R^{CV}(T_{k'}) = \min_k R^{CV}(T_k).$$

Retomando el problema de la base de datos que se esta manejando. Se hicieron los cálculos en computadora para obtener los α_k en la Figura 13, dando como resultado:

$$\alpha_1 = 0, \quad \alpha_2 = \frac{1}{150}, \quad \alpha_3 = \frac{1}{75}, \quad \alpha_4 = \frac{22}{75}, \quad \alpha_5 = \frac{1}{3}.$$

Entonces los α'_k serán:

$$\alpha'_1 = 0, \quad \alpha'_2 = 0.00943, \quad \alpha'_3 = 0.06254, \quad \alpha'_4 = 0.31269.$$

Con diferentes números de divisiones V de la base de datos $\mathcal{L}$, se construyeron los sub-árboles podados óptimos usando los alfas anteriores. Dando como resultado:

		$R^{CV}(T_k)$ (Promedio ± Desv. Estándar)			
k	$\lvert\widetilde{T_k}\rvert$	$V = 2$	$V = 5$	$V = 25$	$V = 150$
1	7	0.04933 ± 0.0098	0.04667 ± 0.009	0.04667 ± 0.007	0.04666
2	4	0.052 ± 0.00643	0.06267 ± 0.015	0.06267 ± 0.007	0.05333
3	3	0.06533 ± 0.009	0.07333 ± 0.004	0.05467 ± 0.0049	0.04666
4	2	0.4533 ± 0.098	0.4853 ± 0.0316	0.484 ± 0.027	0.6667

Tabla 1: Resultados de R^{CV} para cada uno de los árboles T_k.

Dado que los errores R^{CV} mas bajos los obtuvo el árbol asociado a $k = 1$ para cada valor de V. Entonces se puede deducir que el árbol $T_1 = T(0)$ es el más óptimo para este caso.

4. Construyendo un árbol de clasificación para la base de datos Iris

El propósito ahora es emplear la base de datos Iris usando todas sus variables para crear un árbol óptimo con la teoría usada en los anteriores capítulos. Con fines prácticos se notará la base de datos antes mencionada como $\mathcal{L}$. Entonces, la construcción y podado del árbol para la muestrade aprendizaje dio como resultado:

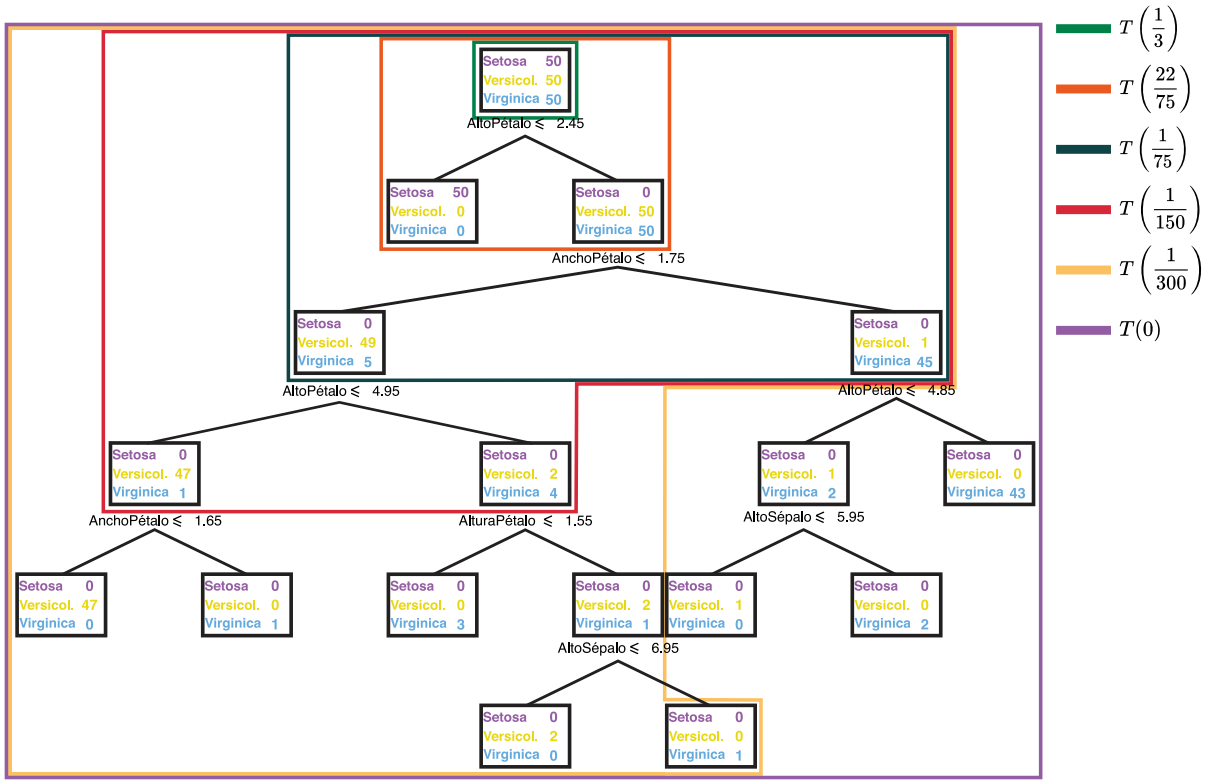


Figura 14: Sub-árboles podados óptimos resultantes.

Empleando la validación cruzada para encontrar el sub-árbol que mejor se desempeña, se obtuvo que los α'_k son:

$$\alpha'_1 = 0, \quad \alpha'_2 = 0.004714, \quad \alpha'_3 = 0.009428, \quad \alpha'_4 = 0.062538, \quad \alpha'_5 = 0.31269.$$

Dando como resultado la siguiente tabla:

		$R^{CV}(T_k)$ (Promedio ± Desv. Estándar)				
k	$\lvert\widetilde{T}_k\rvert$	$V = 6$	$V = 15$	$V = 50$	$V = 75$	$V = 150$
1	9	0.06 ± 0.0072	0.06267 ± 0.011	0.06267 ± 0.0032	0.06267 ± 0.0032	0.06
2	7	0.0533 ± 0.004	0.05733 ± 0.0067	0.056 ± 0.0032	0.05467 ± 0.0034	0.0533
3	4	0.06533 ± 0.0096	0.064 ± 0.005	0.06533 ± 0.0076	0.05867 ± 0.0049	0.0625
4	3	0.07333 ± 0.0072	0.064 ± 0.015	0.06133 ± 0.015	0.04933 ± 0.0052	0.0466
5	2	0.468 ± 0.06	0.456 ± 0.03	0.508 ± 0.0196	0.5387 ± 0.022	0.6667

Tabla 2: Resultados de R^{CV} para cada uno de los árboles T_k construidos a partir de la base de datos Iris.

Aunque el árbol asociado a $k = 4$ muestra un buen rendimiento cuando $V = 150$. El árbol asociado a $k = 2$ tiene mejores resultados de R^{CV} para cada valor de V. Entonces, el árbol de clasificación más óptimo es $T_2 = T\left(\frac{1}{130}\right)$.

5. Conclusión

Como se ha visto en el desarrollo de este trabajo, el modelo de árbol de clasificación tiene un amplio contexto matemático, tanto en su construcción como la validación del clasificador. Además, son sencillos de entender y aplicar.

No obstante, debido a su funcionamiento simple, en algunas base de datos complejas su rendimiento puede ser bajo. Sin embargo, los árboles de clasificación son los cimientos de otros métodos más potentes (Bosques Aleatorios, Gradient Boosting).

Referencias

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.

Choi, H. I. (2017). Classification and Regression Tree (CART). https://www.math.snu.ac.kr/~hichoi/machinelearning/lecturenotes/CART.pdf

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179-188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Loh, W.-Y. (2014). Fifty years of classification and regression trees: fifty years of classification and regression trees. *International Statistical Review*, *82*(3), 329-348. https://doi.org/10.1111/insr.12016

Messenger, R., & Mandell, L. (1972). A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American Statistical Association*, *67*(340), 768-772. https://doi.org/10.1080/01621459.1972.10481290

Morgan, J. N., & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, *58*(302), 415-434. https://doi.org/10.1080/01621459.1963.10500855

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge University Press. https://doi.org/10.1017/CBO9780511812651