Proyecto : Árboles de clasificación Bayesiana.

Simulación estocástica : Teoría y Laboratorio

Gary Vidal
Jorge Sossa

December 22, 2022

Universidad de Chile
Profesor : Joaquin Fontbona
Auxiliares : Arie Wortsman, Camilo Carvajal, Pablo Zúñiga

Índice

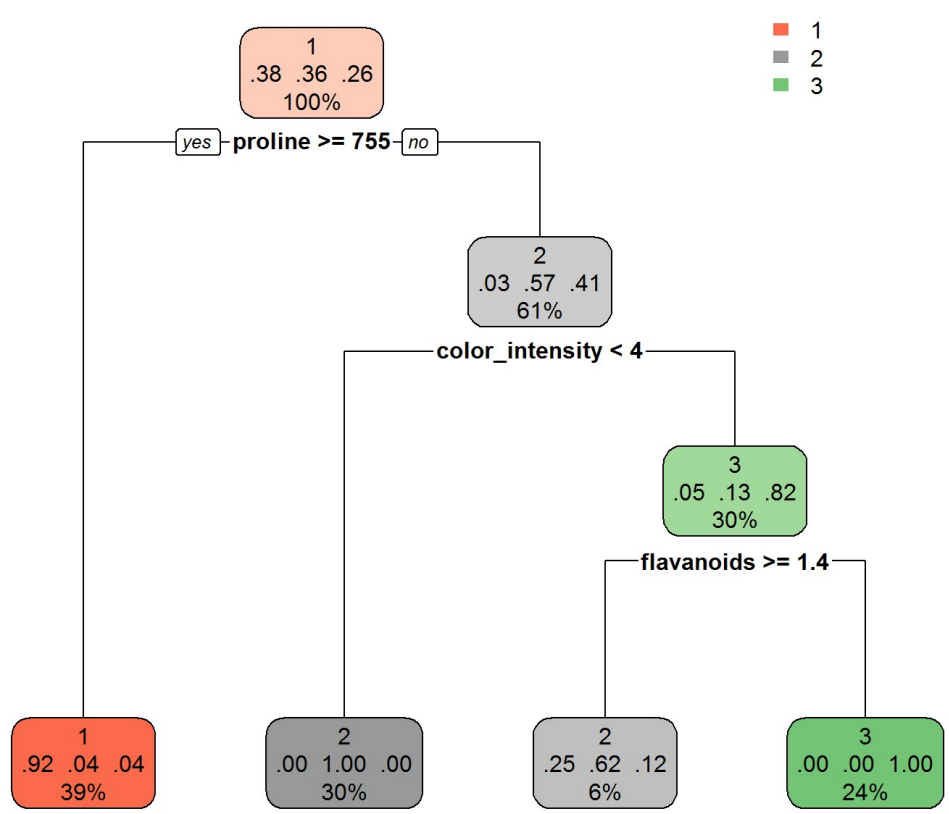
Árbol de decisión

¿QUE ES UN ÁRBOL?

Un Árbol de decisión corresponde a una estructura abstracta, la cual a partir de preguntas del tipo binarias, es capaz de clasificar los elementos de un conjunto.

Elementos de un Árbol:

- Raíz
- Nodo de Decisión
- Nodo Terminal



Objetivo

Objetivo

- Se busca ajustar un árbol de decisión T que permita clasificar/predecir datos.
- Método usual : Utilizar un algoritmo glotón que expande el árbol y luego lo corta.
- Método Bayesiano : Se especifica una distribución a priori para los árboles y se utiliza una búsqueda estocástica para buscar la distribución a posteriori.

Notación :
$X =$ conjunto de predictores.
$Y =$ conjunto de targets.
$y_{ij} =$ Observación del dato j en la hoja i

Prior $\mathbb{P}(T)$ y Posterior $\mathbb{P}(T|Y, X)$

Prior $\mathbb{P}(T)$

Se define $\mathbb{P}(T)$ implícitamente con dos funciones: $p_{split}(\eta, T)$ como la probabilidad de dividir la hoja η y $p_{rule}(\rho|\eta, T)$ como la probabilidad de asignar la regla ρ al nodo dividido.
Generamos el Prior del árbol recursivamente de este modo:

Prior $\mathbb{P}(T)$

1.- Definimos T como una hoja η

2.- Dividimos el nodo terminal η con probabilidad $p_{split}(\eta, T)$.

3.- Si el nodo se divide, se le asigna una regla ρ según $p_{rule}(\rho|\eta, T)$. Se crea una hoja a la izquierda y a la derecha de η. T se actualiza y se vuelve a la etapa 2.

ESPECIFICACIÓN p_{split} Y p_{rule}

Si ignoramos el p_{rule}, y tomamos $p_{split} = \alpha$. Para un árbol binario con b hojas, $\mathbb{P}(T) = \alpha^{b-1}(1-\alpha)^b$.
Buscamos definir p_{split} de tal manera que tengamos control sobre este.

DEFINICIÓN $p_{split}(\eta, T)$

- $p_{split} = \alpha(1+d_\eta)^{-\beta}$, donde d_η es la profundidad del nodo η y $\beta \geq 0$

Esto permite tener control sobre cuales nodos se dividen, dependiendo de α y β.

ESPECIFICACIÓN p_{split} Y p_{rule}

Definimos p_{rule} sobre el conjunto de predictores (características de los datos) de la siguiente manera :

DEFINICIÓN $p_{rule}(\rho|\eta, T)$

- Tomamos x_i al azar entre los predictores.
- Tomar s al azar entre las observaciones de x_i si este es cuantitativo.
  Tomar S al azar entre los subconjuntos posibles de x_i si este es cualitativo.

Este p_{rule} no debe generar nodos vacíos.

CÁLCULO DE $\mathbb{P}(T|Y,X)$

Tenemos la relación entre prior y posterior :

$$\mathbb{P}(T|Y,X) \propto \mathbb{P}(Y|T,X)\mathbb{P}(T)$$

Para un árbol de clasificación con K categorías, definimos $\mathbb{P}(Y|T,X)$ como :

DEFINICIÓN $\mathbb{P}(Y|T,X)$ [Posterior conjugada de Dirichlet]

$$\mathbb{P}(Y|T,X) = \left(\frac{\Gamma(\sum_k \alpha_k^p)}{\prod_k(\Gamma(\alpha_k^p))}\right)^b \prod_{i=1}^{b} \frac{\prod_k \Gamma(n_{ik}+\alpha_k^p)}{\Gamma(n_i+\sum_k \alpha_k^p)}$$

- $b =$ número de hojas.
- $\alpha^p = [\alpha_1^p, ..., \alpha_K^p]$ vector de peso de las clases.
- $n_{ik} = \sum_j 1(y_{ij} \in C_k)$
- $n_i = \sum_k n_{ik}$

Búsqueda de la Posteriori

BÚSQUEDA DE LA POSTERIORI

Usaremos Metropolis-Hastings, para generar una cadena de árboles $T^0, T^1, T^2, ...$, que converga a $\mathbb{P}(T|Y, X)$:

ALGORTIMO METROPOLIS-HASTINGS

1 Comenzando del árbol trivial T^0, simularemos transiciones de T^i a T^{i+1} en dos pasos.

1.1 Generamos un candidato T^* con probabilidad $q(T^i, T^*)$

1.2 Hacemos $T^{i+1} = T^*$ con probabilidad:

$$\alpha(T^i, T^*) = min\left\{\frac{q(T^*, T^i)\mathbb{P}(Y|X, T^*)\mathbb{P}(T^*)}{q(T^i, T^*)\mathbb{P}(Y|X, T^i)\mathbb{P}(T^i)}, 1\right\}$$

si no , $T^{i+1} = T^i$

TRANSICIONES $q(T^i, T^*)$

Consideraremos $q(T^i, T^*)$ la forma de generar T^* desde T^i usando aleatoriamente una de estas 4 operaciones:

- CRECER: Aleatoriamente elegir un nodo terminal y separarlo en 2 nuevos, asignando una regla de decisión aleatoria (p_{rule}).
- PODAR: Aleatoriamente elegir un padre con con 2 nodos terminales y volverlo terminal.
- CAMBIAR: Aleatoriamente elegir un nodo interno y reasignarle una regla de decisión (p_{rule}).
- INTERCAMBIAR: Aleatoriamente elegir una pareja padre-hijo, ambos nodos internos y cambiar sus reglas de decisión.

TRANSICIONES $q(T^i, T^*)$

Llamemos: $prob = [prob_{crecer}, prob_{podar}, prob_{cambiar}, prob_{intercambiar}]$
$N_{operación}$ = Cantidad de nodos que se puede hacer la operación
y $N_{op} = [N_{crecer}, N_{podar}, N_{cambiar}, N_{intercambiar}]$
Así elegiremos un operación con la probabilidad ponderada de
$prob \times N_{op}$
definamos $p_{split}(\eta) = \alpha_\eta =$ y $p_{prune}(\eta) = 1 - p_{split}(\eta)$ Ahora al elegir:

- CRECER: Habrá que elegir el nodo η con probabilidad $\frac{\alpha_\eta}{\sum_{j=1}^{N_{crecer}(T^i)} \alpha_j}$
- PODAR: Habrá que elegir el nodo η con probabilidad $\frac{1-\alpha_\eta}{\sum_{j=1}^{N_{podar}(T^i)} 1-\alpha_j}$

Resultados

Resultados Breast Cancer Dataset

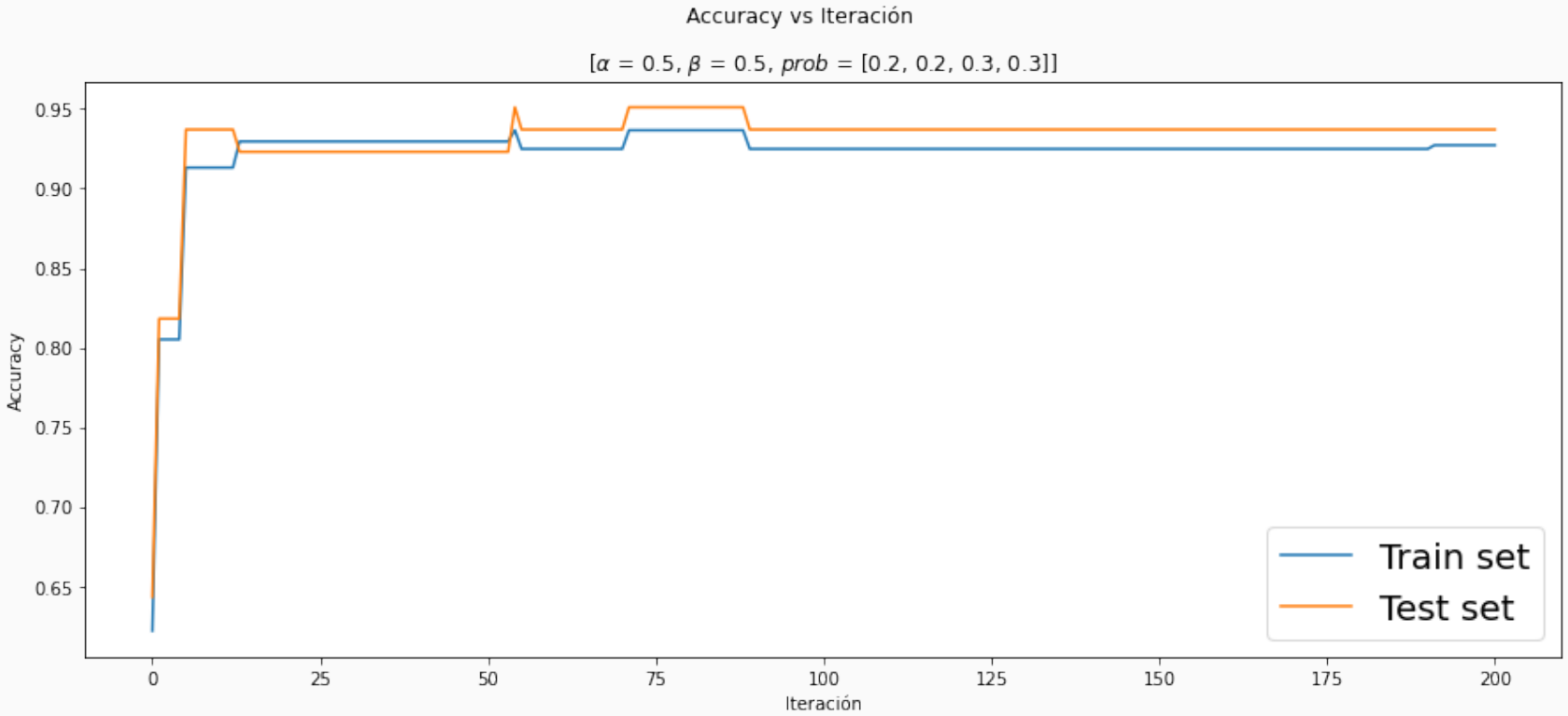


Figure 1: Precisión del modelo con $\alpha^p = [1, 1]$

Resultados Breast Cancer Dataset

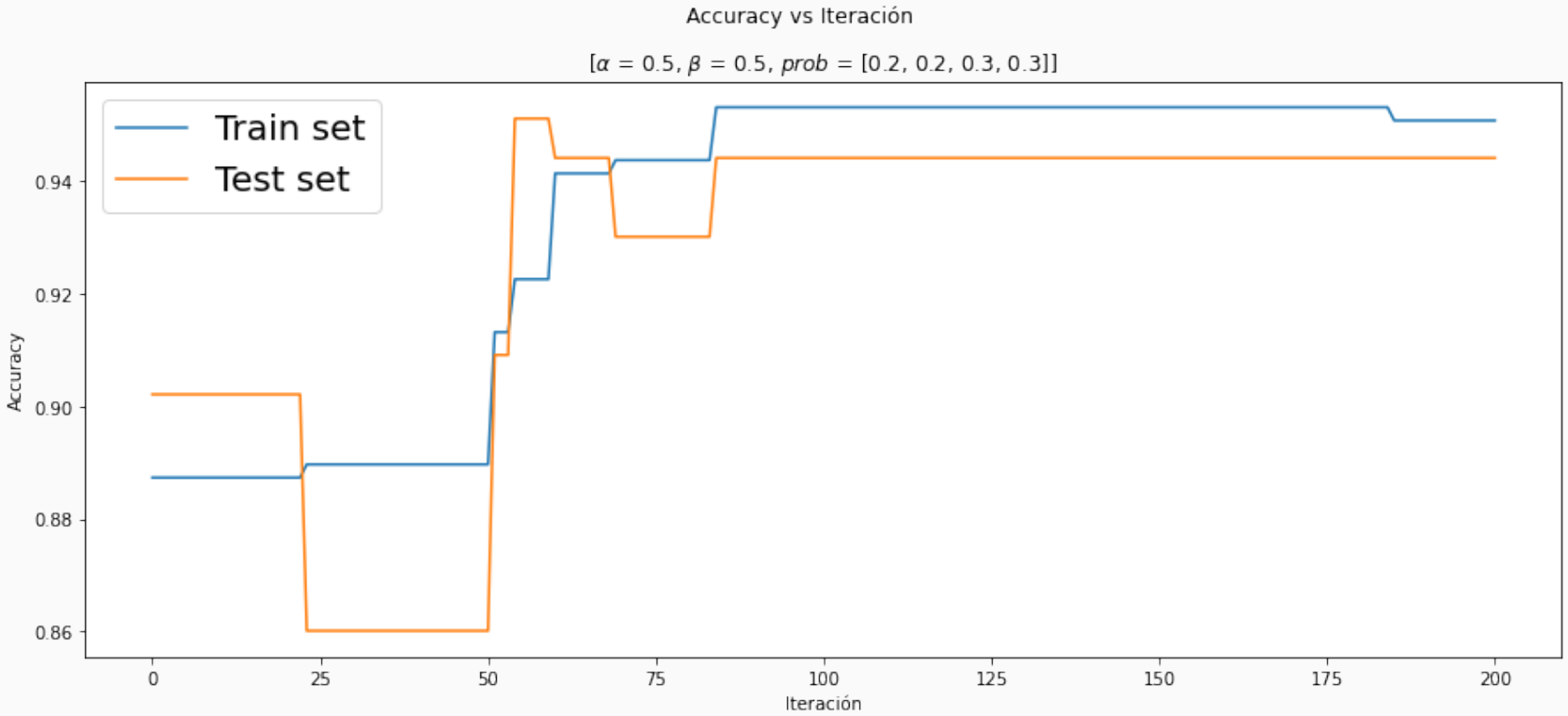


Figure 2: Precisión del modelo con $\alpha^p = [1, 2]$

Resultados Breast Cancer Dataset

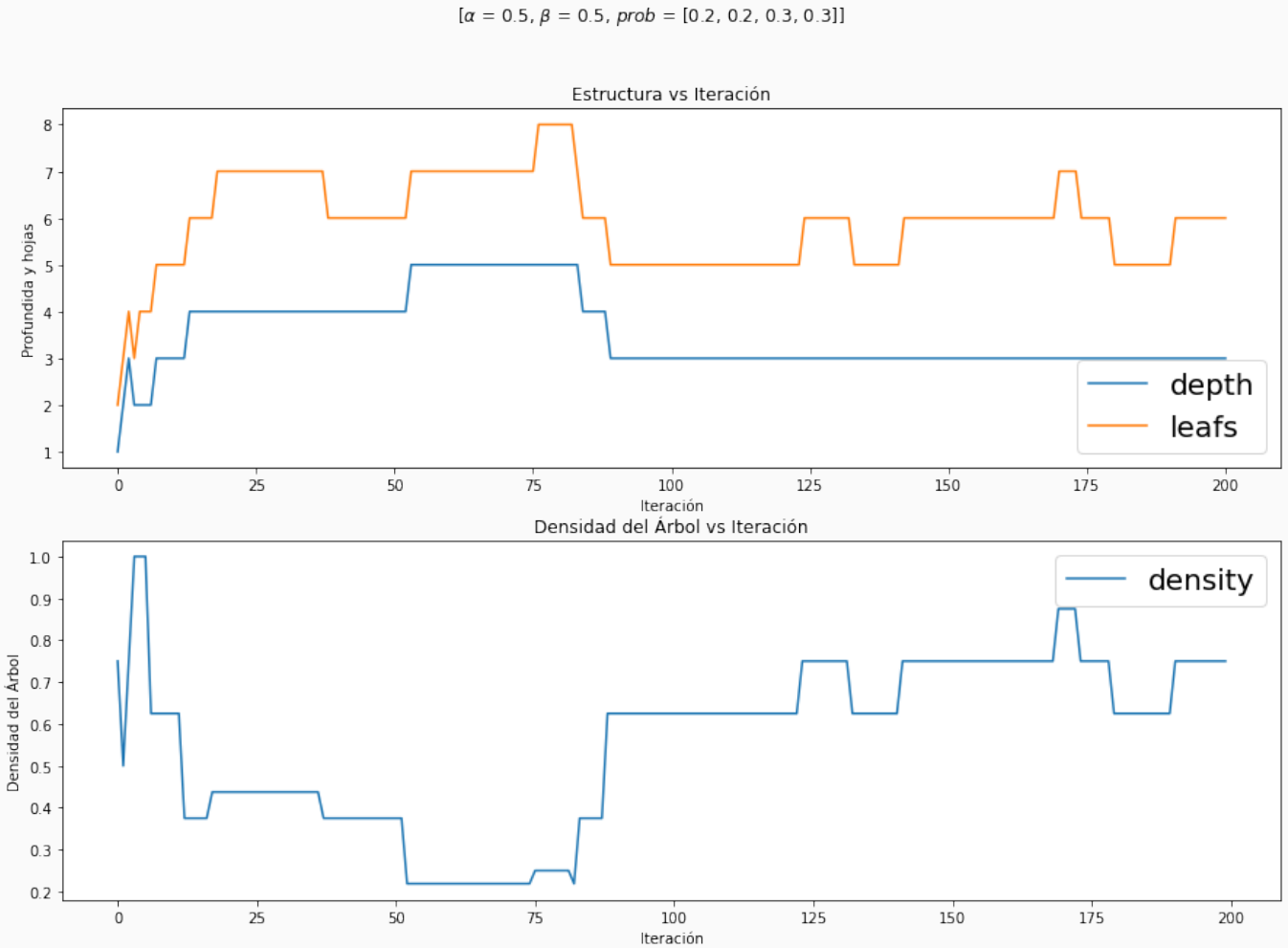


Figure 3: Estructura del árbol con $\alpha^p = [1, 2]$

Resultados Breast Cancer Dataset

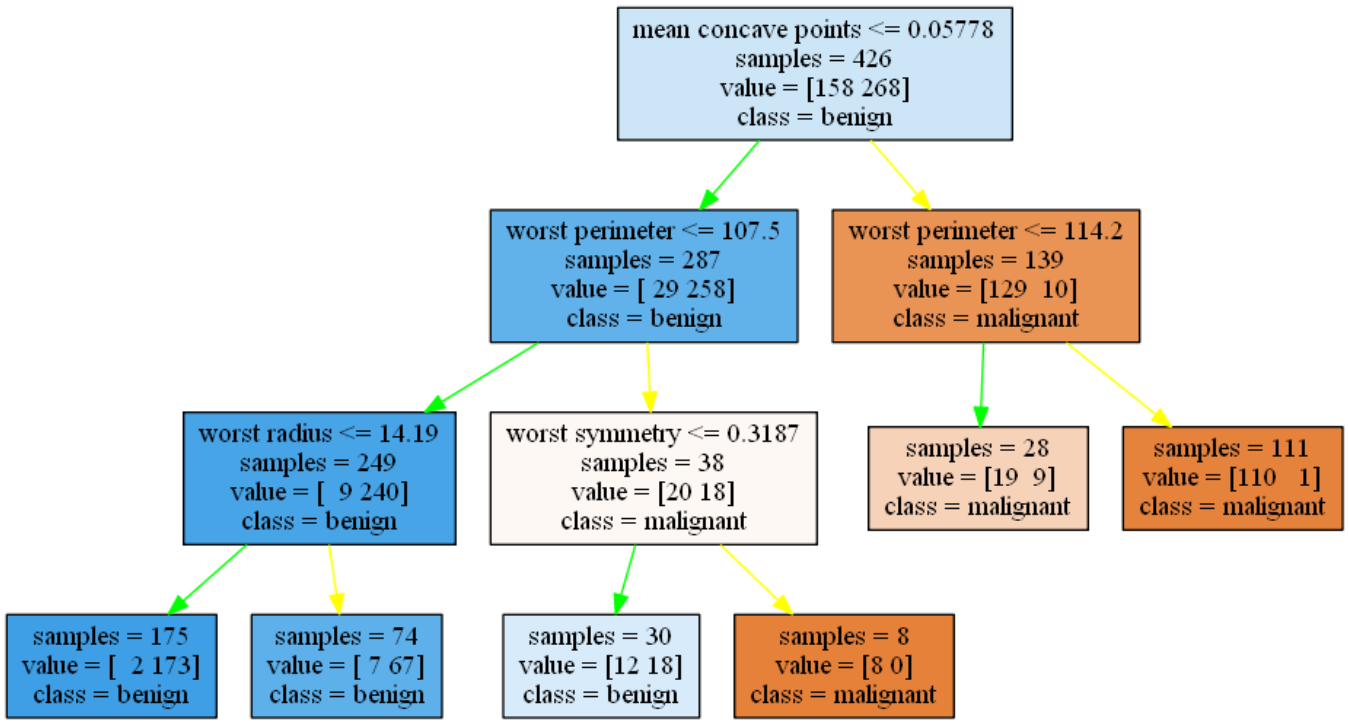


Figure 4: Árbol final $\alpha^p = [1, 1]$

Resultados Breast Cancer Dataset

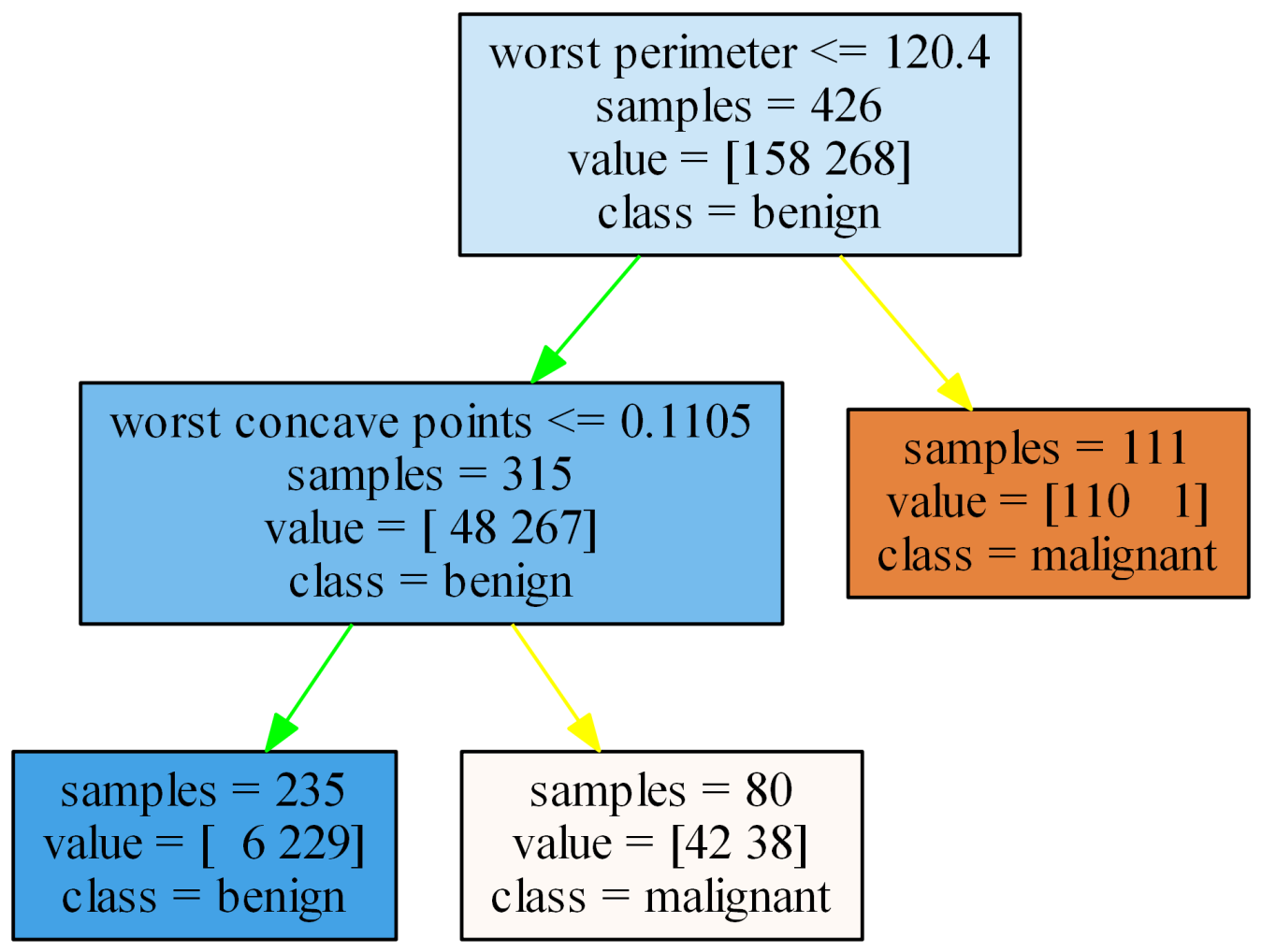


Figure 5: Árbol final $\alpha^p = [1, 2]$

Resultados Breast Cancer Dataset

	Modelo $\alpha^p = [1, 1]$	**Modelo** $\alpha^p = [1, 2]$
Precisión	0.95	0.95

Table 1: Precisión de los árboles Bayesianos

	Reg. Log. (SKL)	**Dec. Trees (SKL)**
Precisión	0.94	0.93

Table 2: Precisión de la regresión logística y del módulo Decision Trees de Sk-learn

Resultados Iris Dataset

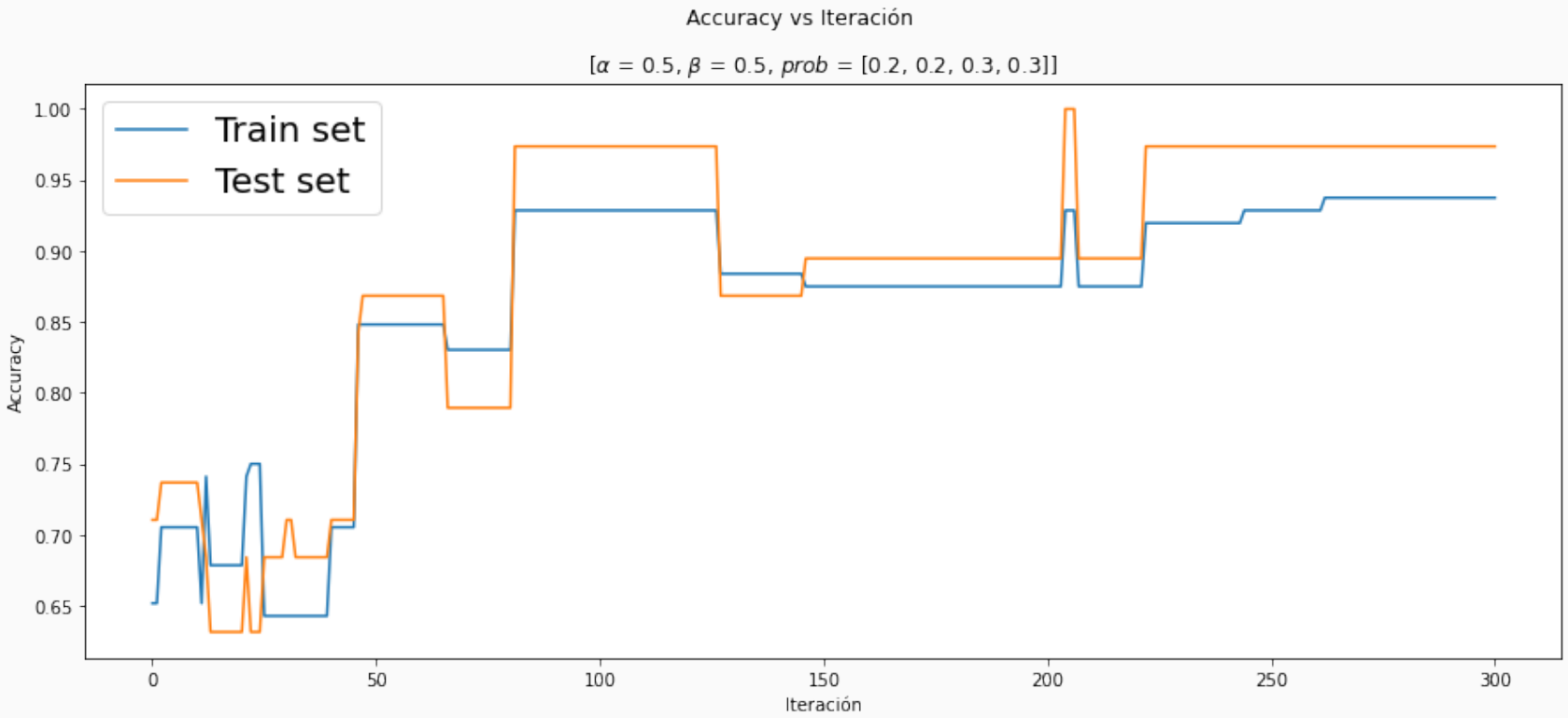


Figure 6: Precisión del modelo con $\alpha^p = [1, 1, 1]$. La precisión máxima de este modelo es 1.

Resultados Iris Dataset

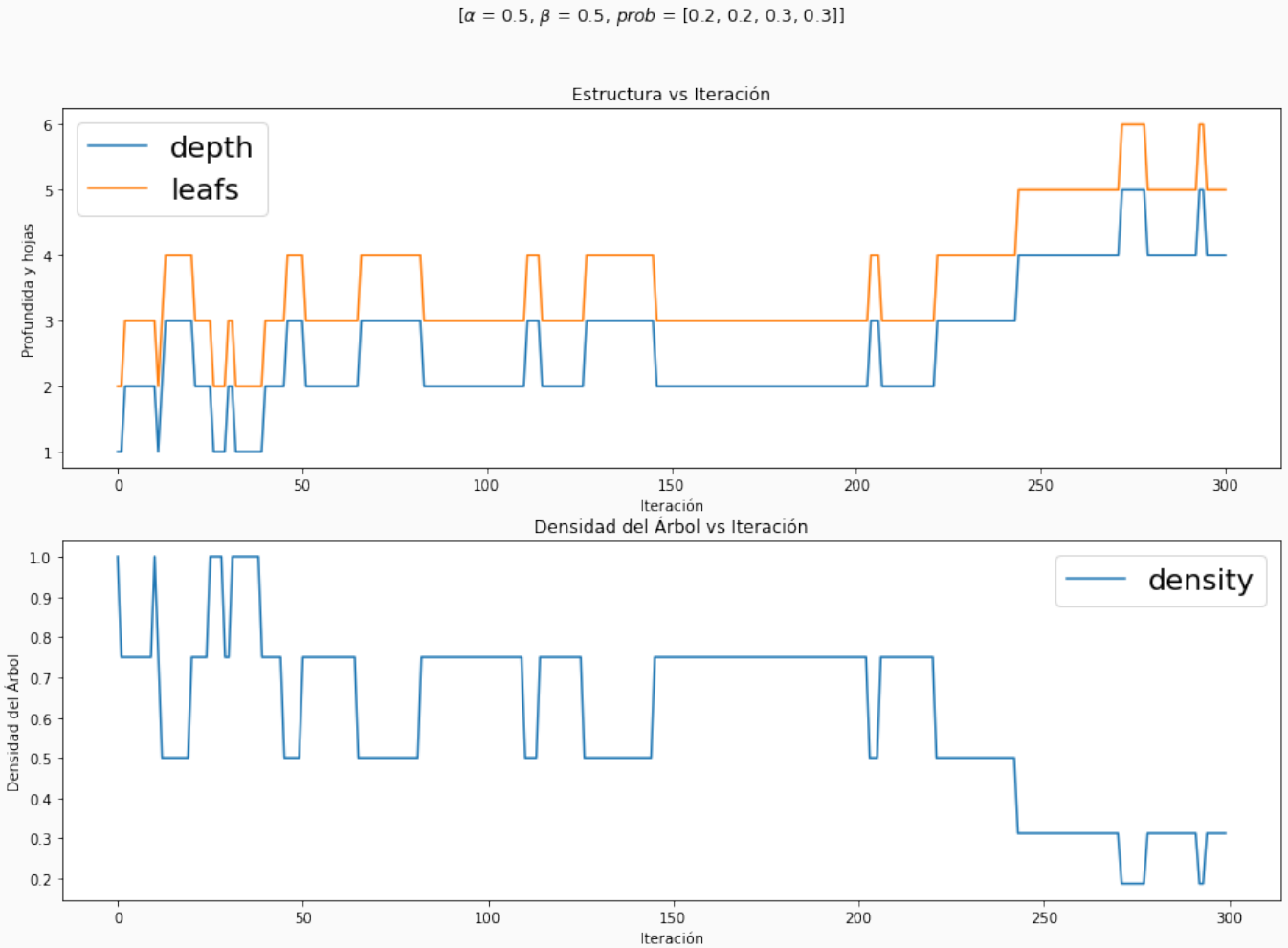


Figure 7: Estructura del árbol con $\alpha^p = [1, 1, 1]$

Resultados Digits Dataset

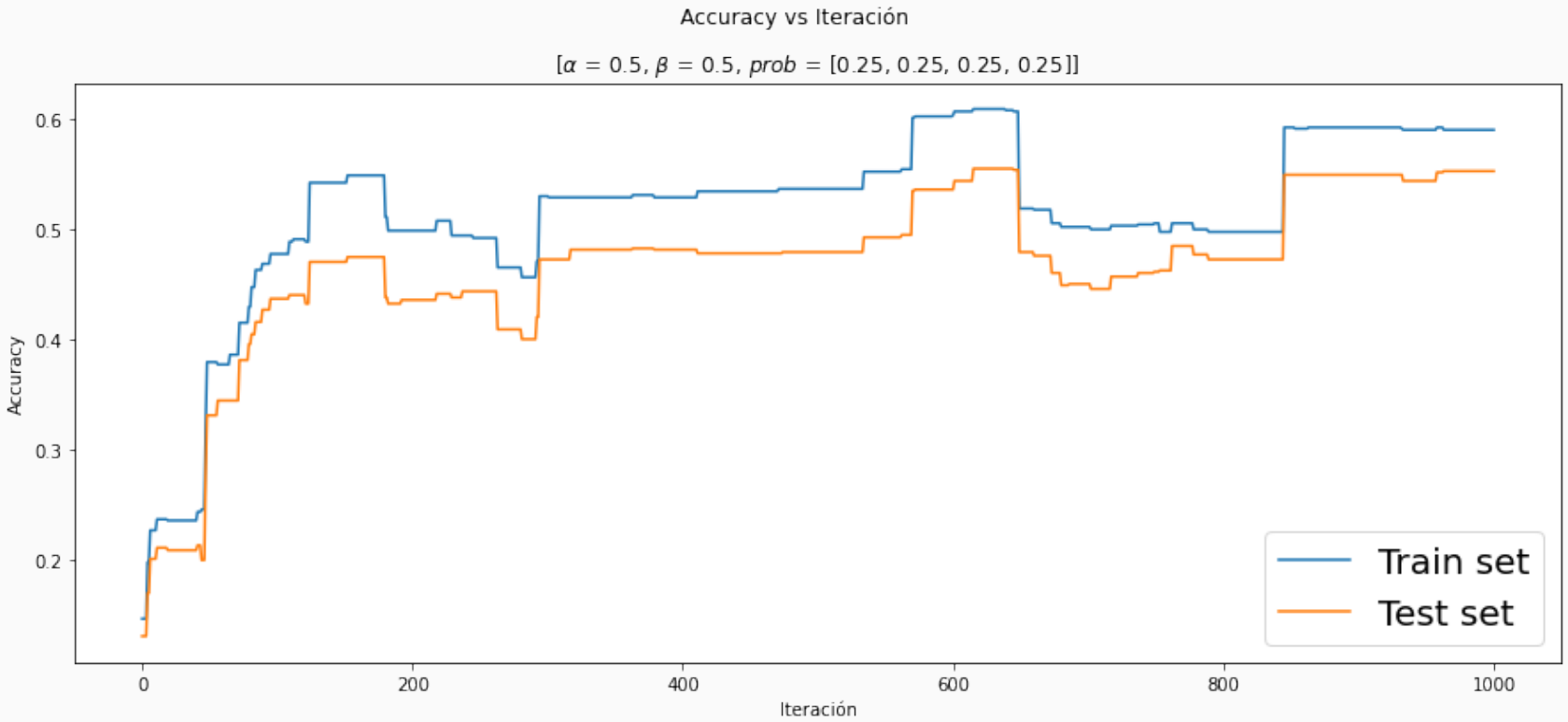


Figure 8: Precisión del modelo.

Resultados Digits Dataset

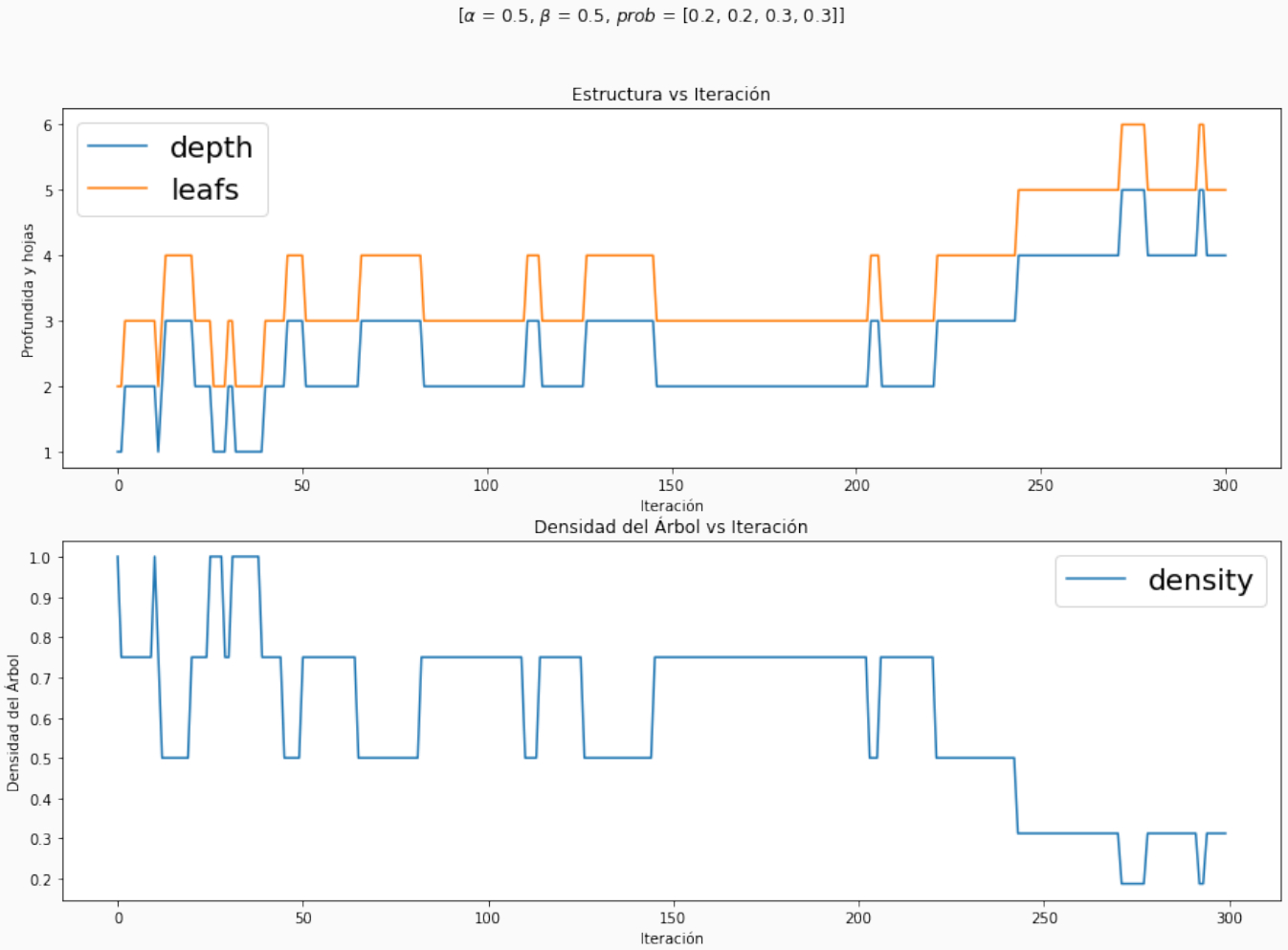


Figure 9: Estructura del modelo.

Resultados Digits Dataset solo 5 números

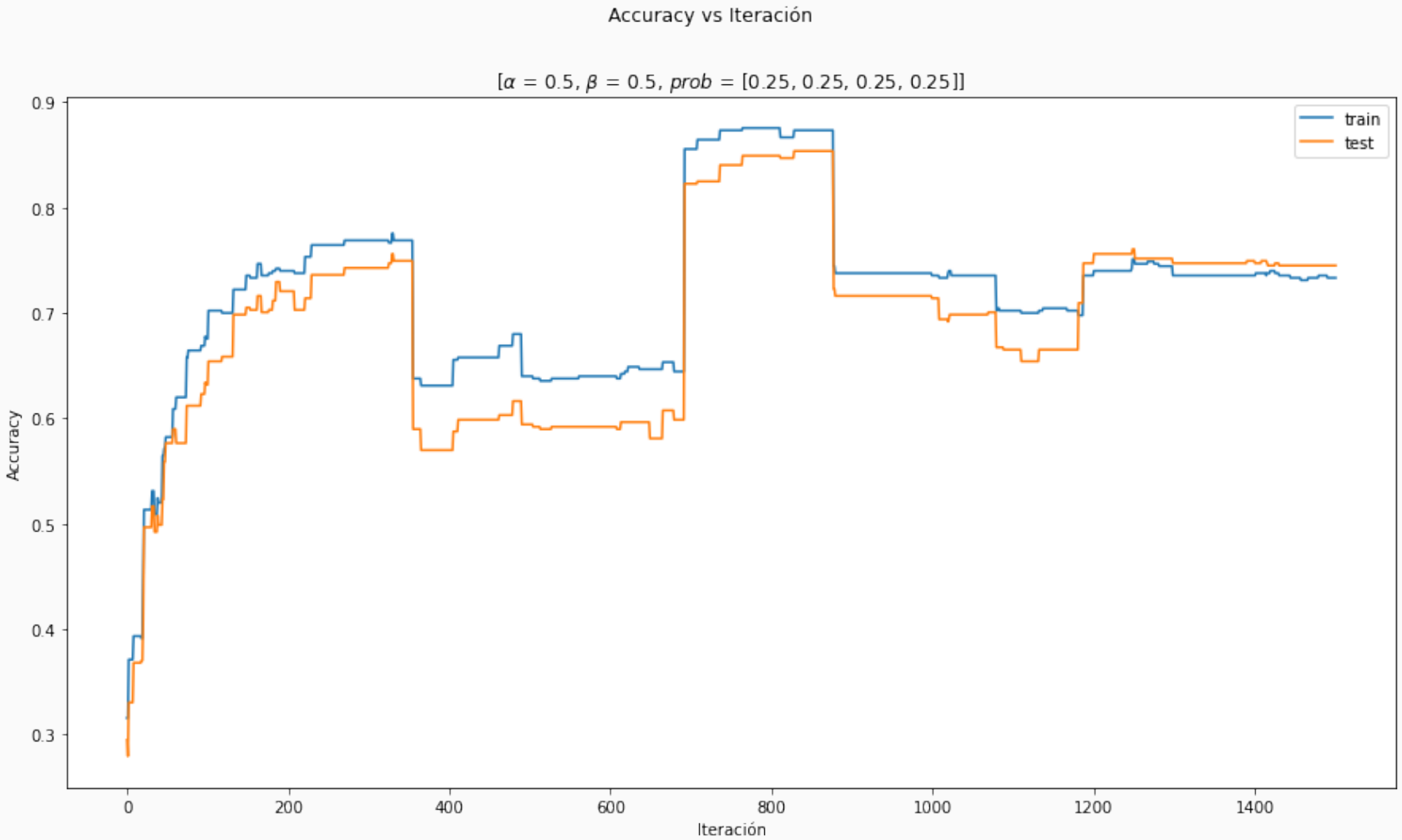


Figure 10: Precisión del modelo con 5 digitos.

Resultados Digits Dataset solo 5 números

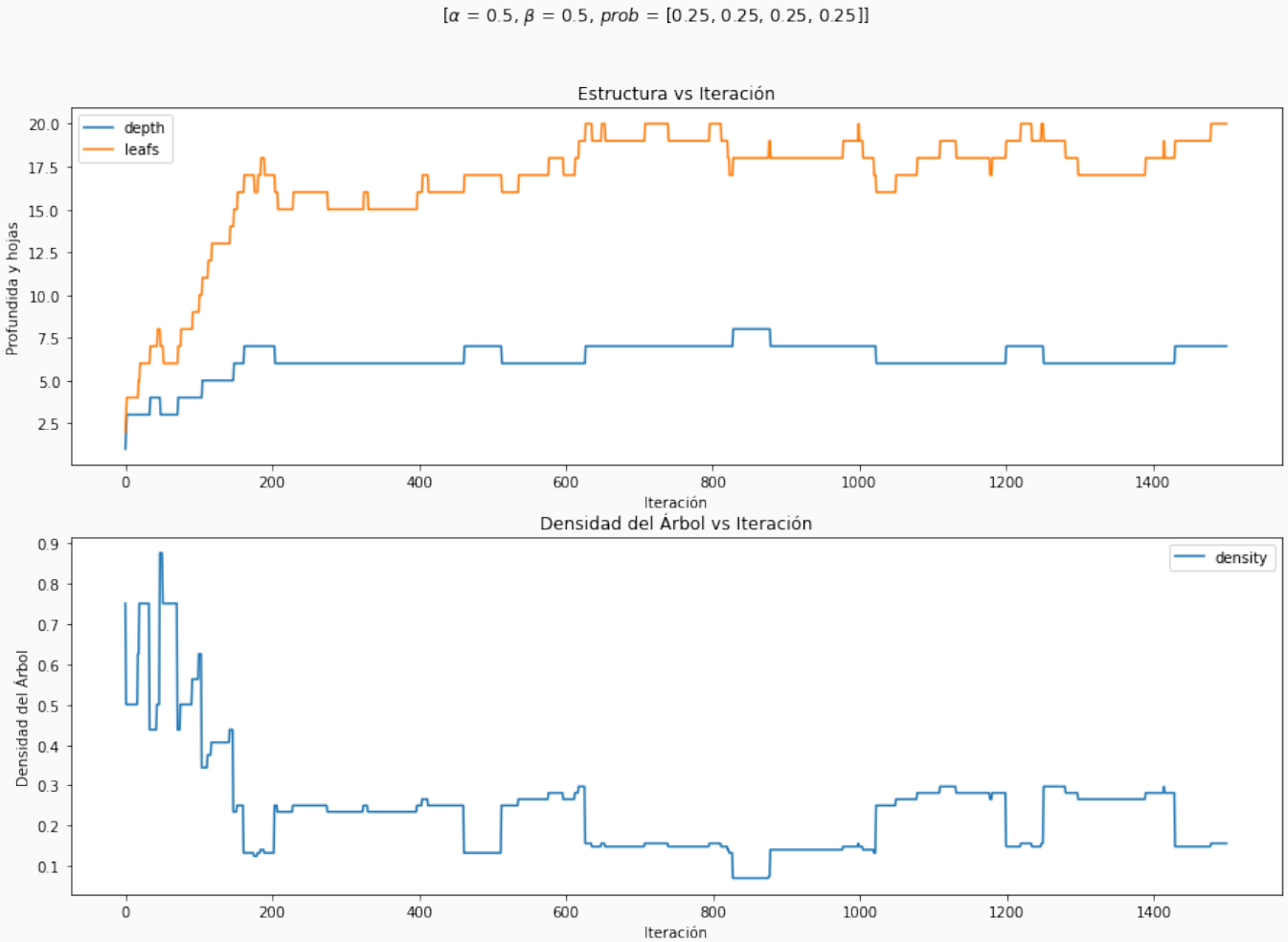


Figure 11: Estructura del modelo

Conclusión

Conclusión

- El algoritmo crea una cadena de árboles que se acerca en distribución al posteriori $\mathbb{P}(T \mid Y, X)$.
- El algoritmo desciende de manera rápida a un óptimo local, lo que impide obtener buenas soluciones para árboles más grandes.
- Esto implica que al aumentar las clases el algoritmo se demora mucho más en ajustar el modelo.
- Se podría controlar mejor el descenso posteriori del árbol restringiendo su cantidad de hojas (parecido a arboles de α-complejidad) .