Arboles de Decisión

Arboles de Decisión

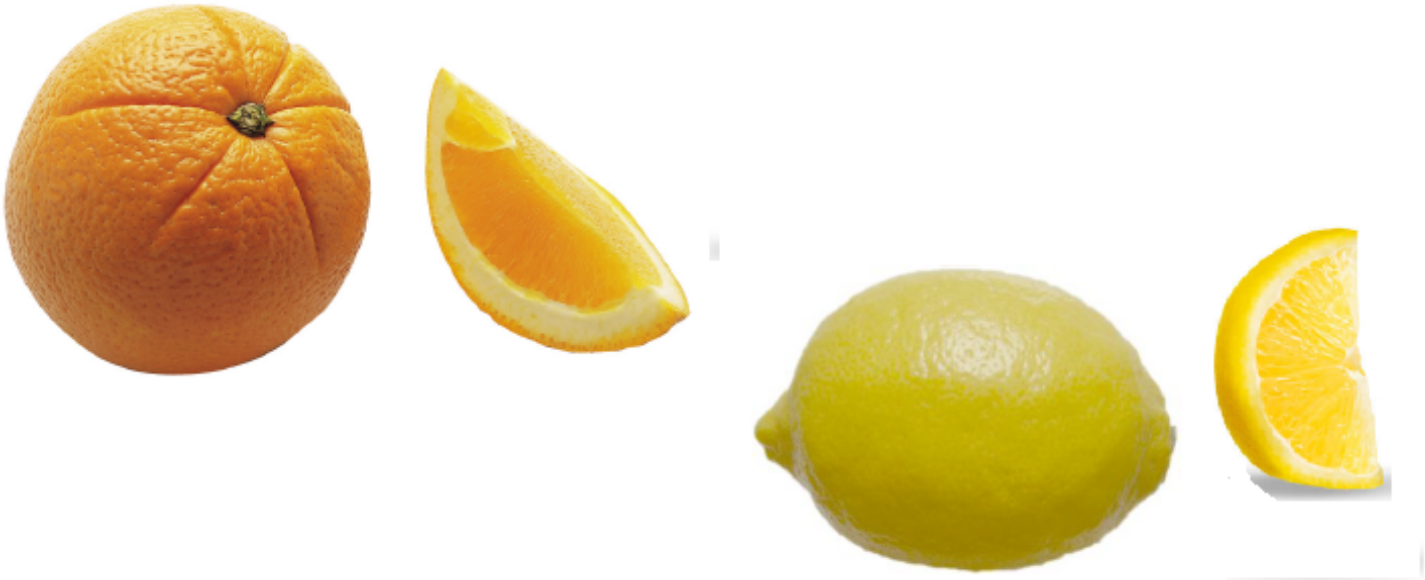
Idea Básica

Seguimos en el problema de predecir la clase Y a partir de un conjunto de covariables o atributos $\mathbf{X}$.

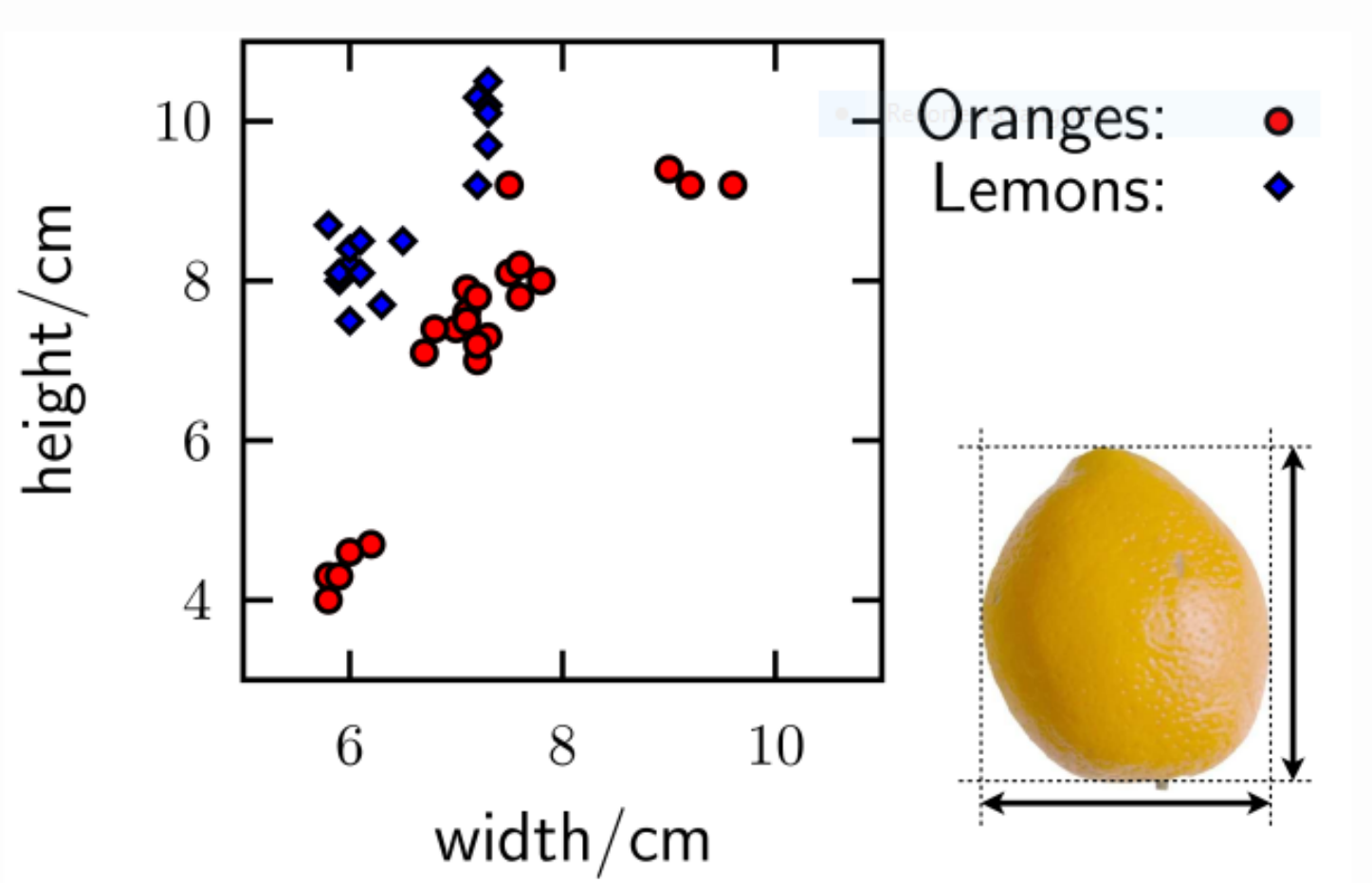
Hemos visto KNN, Bayes Naive, alguna otra idea?

- Los árboles son una herramienta muy flexible, de fácil interpretación, que partiendo el espacio de los atributos en rectángulos permite predecir la clase Y.
- Permiten trabajar con no-linealidades e interacciones.
- CART: Classification And Regression Trees (Breiman, Friedman, Olshen & Stone, 1984)

Toy Example: Limones y Naranjas



Toy Example (datos reales)



(Gracias https://homepages.inf.ed.ac.uk/imurray2/teaching/oranges_and_lemons/)

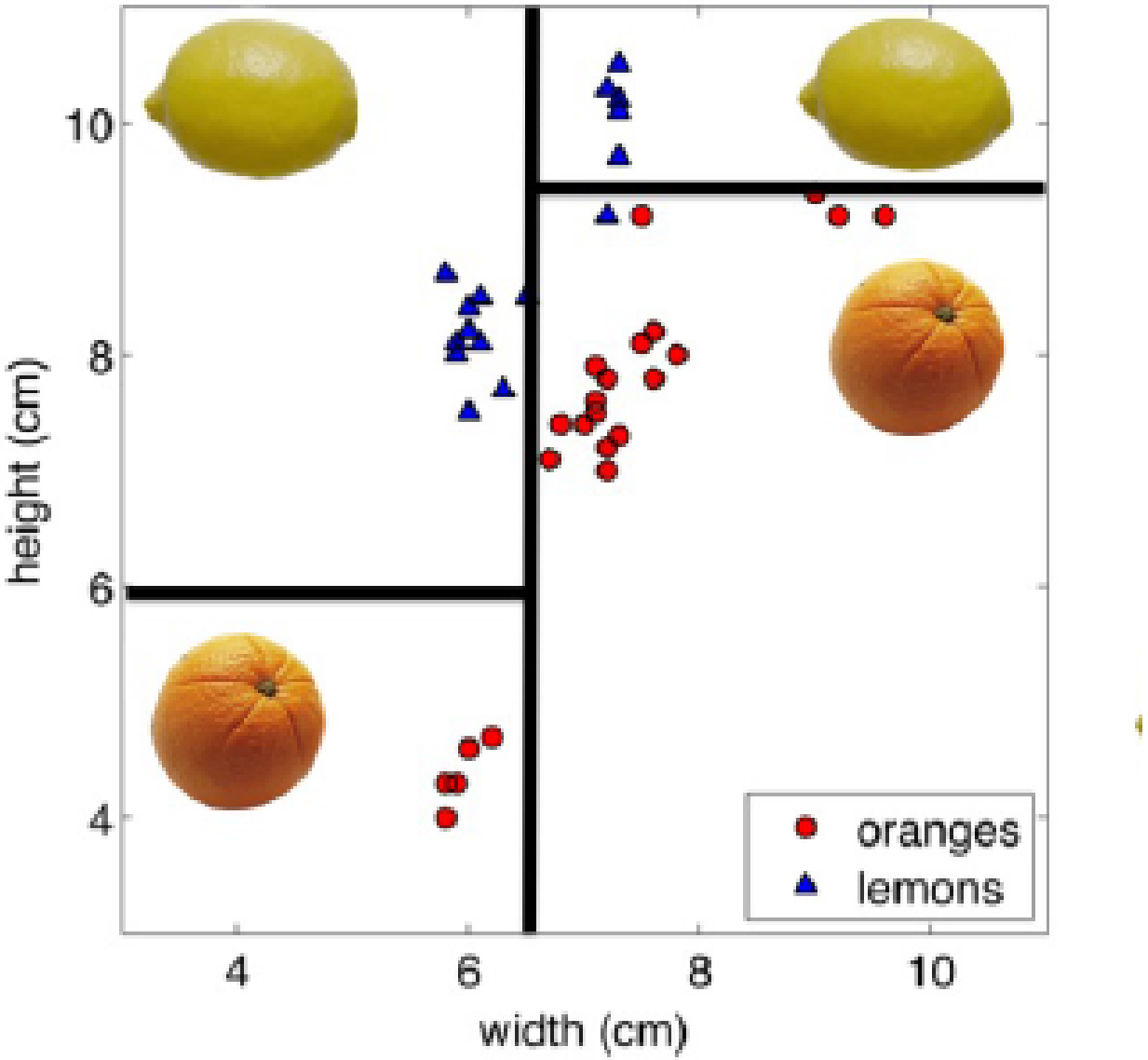
Idea Básica de CART

Vamos a concentrarnos en el caso de

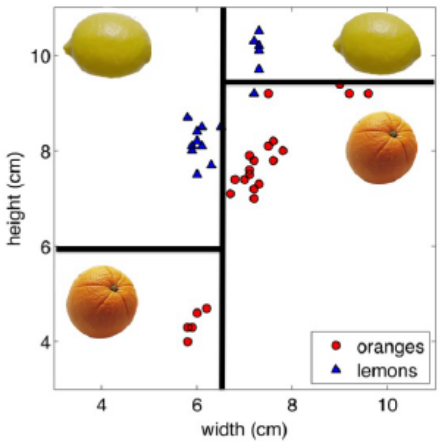
Particiones recursivas binarias del espacio de atributos

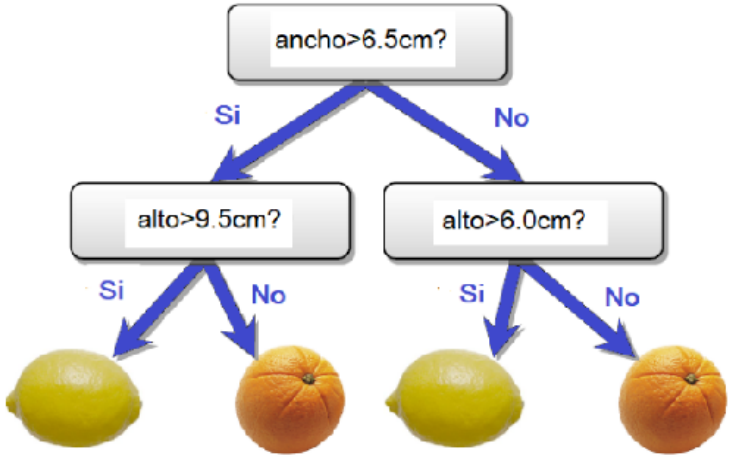
- Y indica la clase $(1, 2, \ldots, K)$ y X_1 y X_2 son los atributos.
- Partimos el espacio (X_1, X_2) en dos regiones considerando solo una de las variables: partición horizontal o vertical.
- En cada región predecimos con la regla de la mayoría.
- Continuamos partiendo las regiones, con el mismo criterio.

Toy Example: Limones y Naranjas

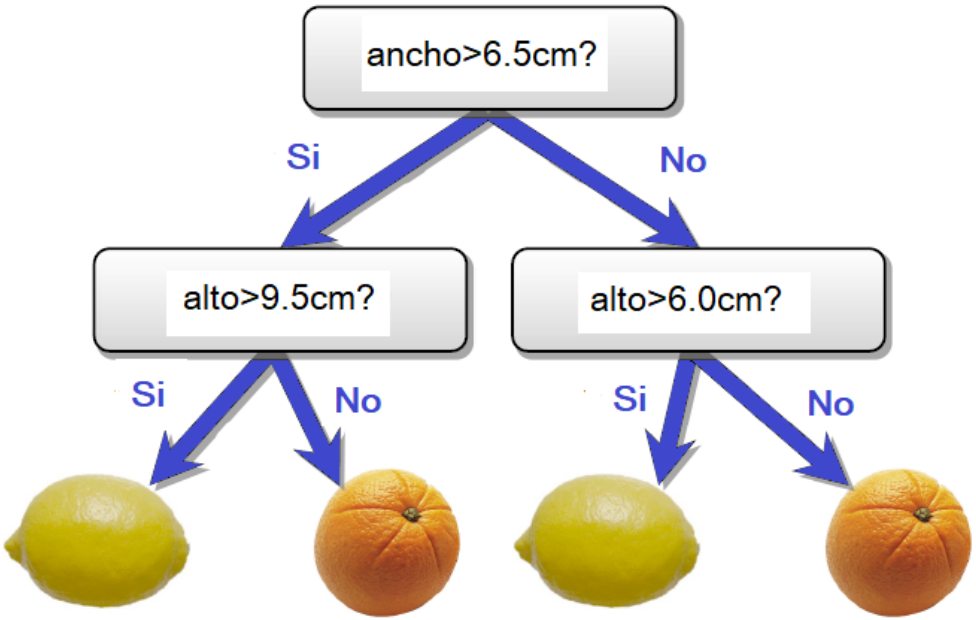


Toy Example: Limones y Naranjas

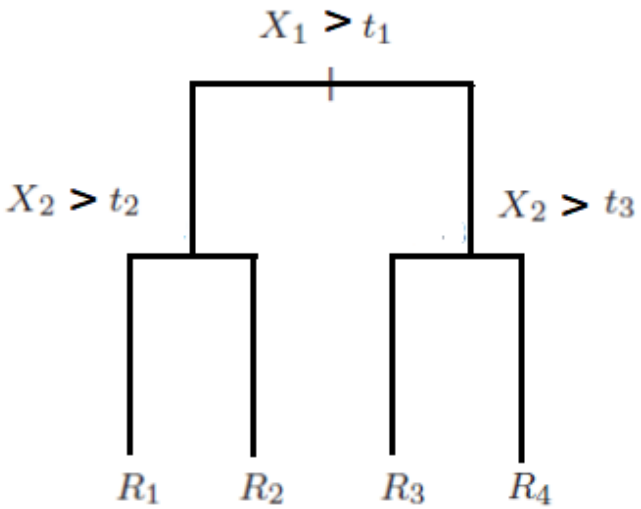




Toy Example: Limones y Naranjas



Arbol: un poco de jerga



Arbol: Estructura

- Nodos $\longleftrightarrow$ Tests en las Variables
- Ramas $\longleftrightarrow$ Resultados de los Tests
- Hojas $\longleftrightarrow$ Clasificación

Ejemplo Real

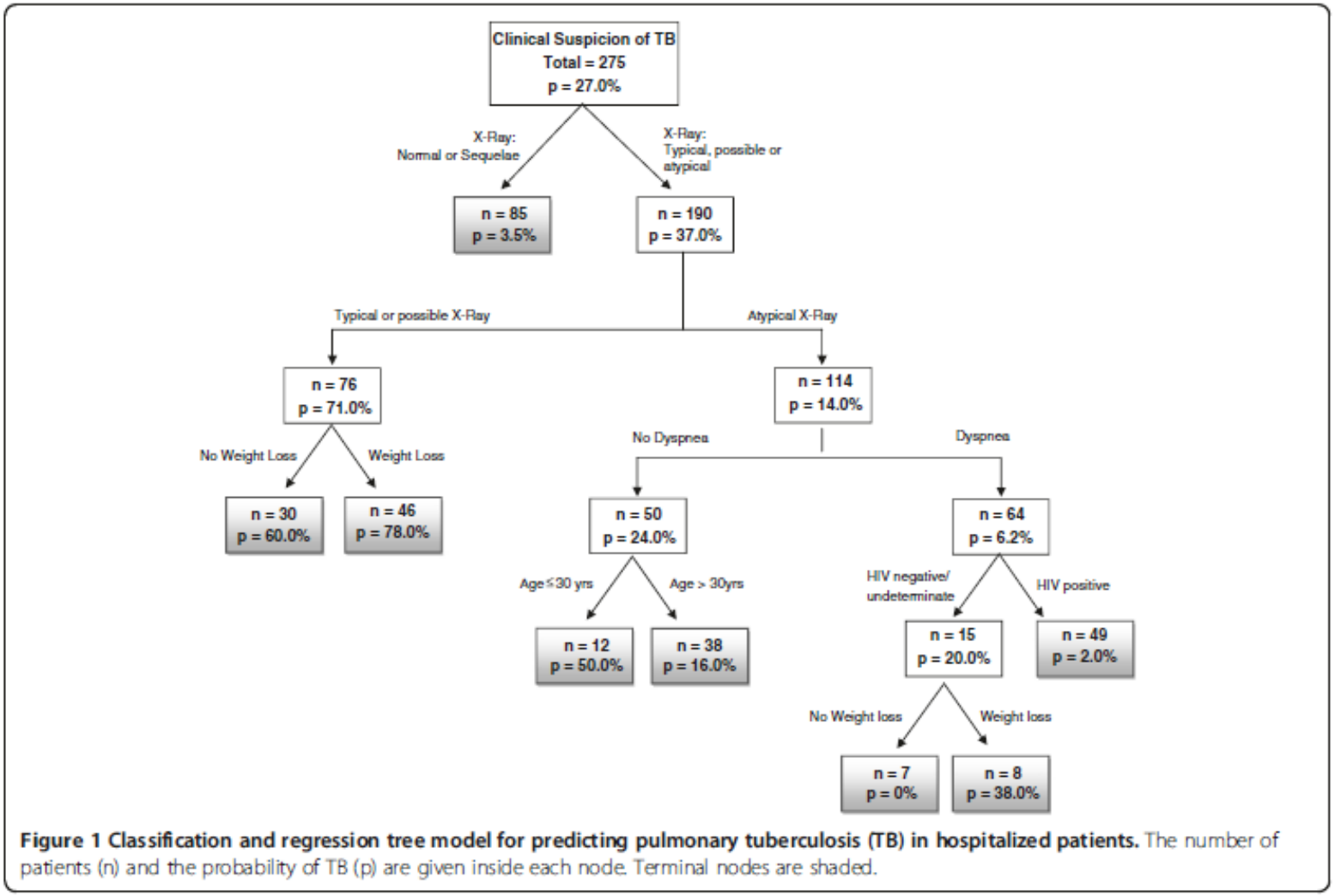


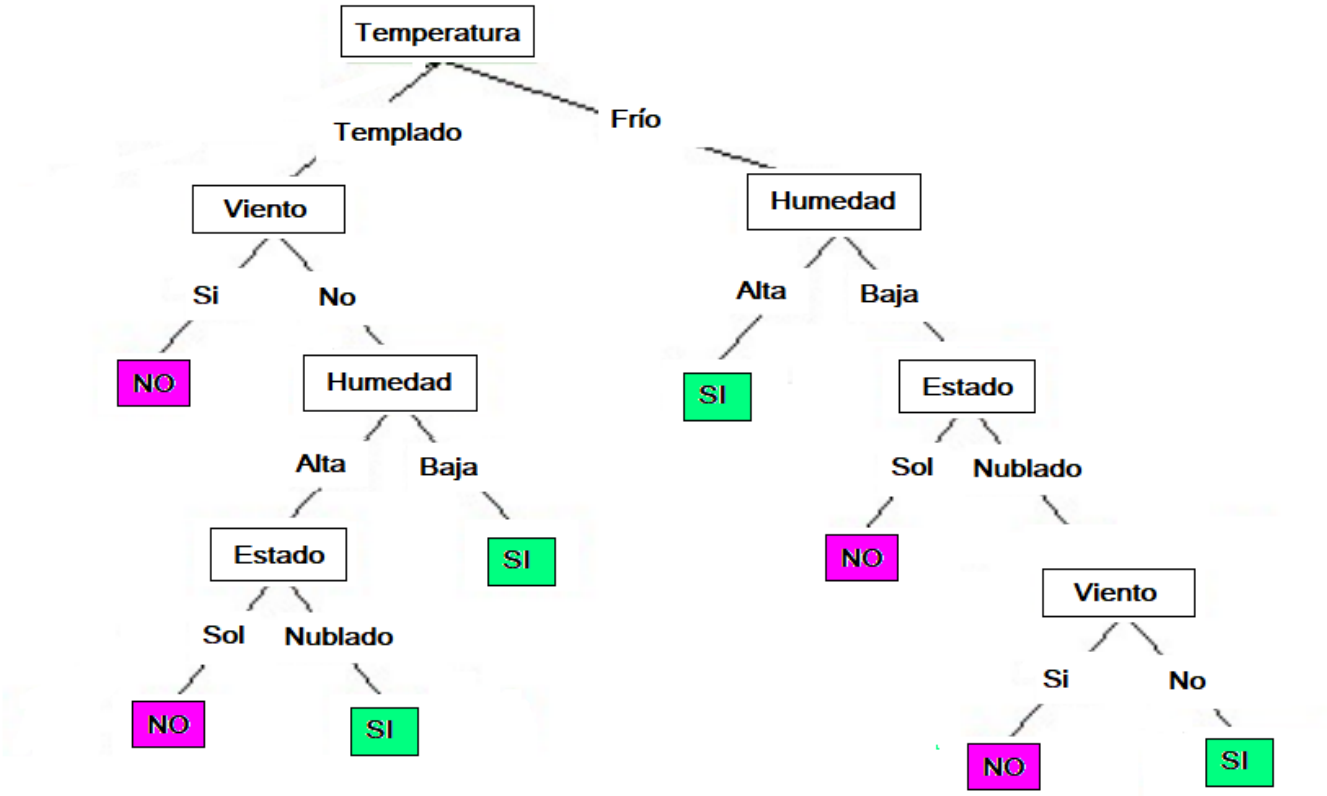
Figure 1 Classification and regression tree model for predicting pulmonary tuberculosis (TB) in hospitalized patients. The number of patients (n) and the probability of TB (p) are given inside each node. Terminal nodes are shaded.

Fuente: Classification and regression tree (CART) model to predict pulmonary tuberculosis in hospitalized patients, Aguiar et al. (2012)

No llueve, ¿hacemos un pic-nic?

- Temperatura (Templado, Frío)
- Humedad (Alta, Baja)
- Viento (Si, No)
- Estado (Soleado, Nublado)

No lueve, ¿hacemos un pic-nic?



Arbol de Decisión: Algoritmo

- Elegimos un atributo (¿cuál?). Realizamos un test (¿cuál?)
- Condicional a esta elección, elegimos otra condición, realizamos un test. . .
- En cada hoja predecimos con la regla de la mayoría.
- Procedemos igual en cada rama.

Construcción de un árbol: estrategias

1. En cada nivel debemos elegir:

   - Qué variable dividir: X_j

   - Dónde dividirla: $\tilde{x}_j$

Construcción de un árbol: estrategias

1. En cada nivel debemos elegir:

   - Qué variable dividir: X_j

   - Dónde dividirla: $\tilde{x}_j$

Para cada variable, ¿cuántas divisiones posibles existen?

- *Variable discreta o continua*: $\#$ $\{$ valores posibles$\}$ -1
- *Variable categórica* ($M = \#$ $\{$categorías$\}$): $2^{M-1} - 1$

Construcción de un árbol

1. En cada nivel debemos elegir:

   - Qué variable dividir: X_j

   - Dónde dividirla: $\tilde{x}_j$

2. Elegirlos en base a un criterio de ganancia:
   (elijamos los atributos y los cortes que dan mayor ganancia!!)

Error de Clasificación

Indexamos los nodos con τ.

En el nodo τ asociado a la región R_τ, con $n_\tau = |\tau|$ observaciones, consideramos la proporción de observaciones de la clase k que yacen en dicha región.

$$\hat{p}_{\tau k} = \frac{1}{n_\tau} \sum_{x_i \subset R_\tau} I\left(y_i = k\right)$$

Regla de la mayoría: $k^* = \arg\max_k \hat{p}_{\tau k}$

Error de Clasificación

$$E = 1 - \hat{p}_{\tau k^*} = 1 - \max_k \hat{p}_{\tau k} \tag{1}$$

fracción de observaciones del nodo que no pertenecen a la clase mayoritaria.

Alternativas: Impureza

Dentro de cada nodo y cada variable, buscamos una división tal que cada región sea lo más homogénea posible. Si fuera posible, todas de una misma clase. . . tratamos de hacer pequeña la **impureza**.

En Clasificación en cada región se usa el concepto de **impureza**. Se trata de minimizar la impureza.

$i(p_1, p_2, \ldots, p_K) : \mathbb{R}^K_{\geq 0} \to \mathbb{R}$:

- $p_1 + p_2 + \cdots + p_K = 1, \quad p_i \geq 0$
- $i(p_1, p_2, \ldots, p_K) = 0$ si (1, 0,. . . ,0), (0,1,. . . ,0),etc.
- $i(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K})$ máxima.

Impureza

Indice de Gini: especie de **varianza total** en el nodo

$$G = \sum_{k=1}^{K} p_k \left(1 - p_k\right)$$

Deviance o Entropía (Cross-Entropy): log–verosimilitud binomial

$$D = -\sum_{k=1}^{K} p_k \log p_k$$

Impureza

Indice de Gini

$$G = \sum_{k=1}^{K} \hat{p}_{\tau k} \left(1 - \hat{p}_{\tau k}\right)$$

Cross-Entropy

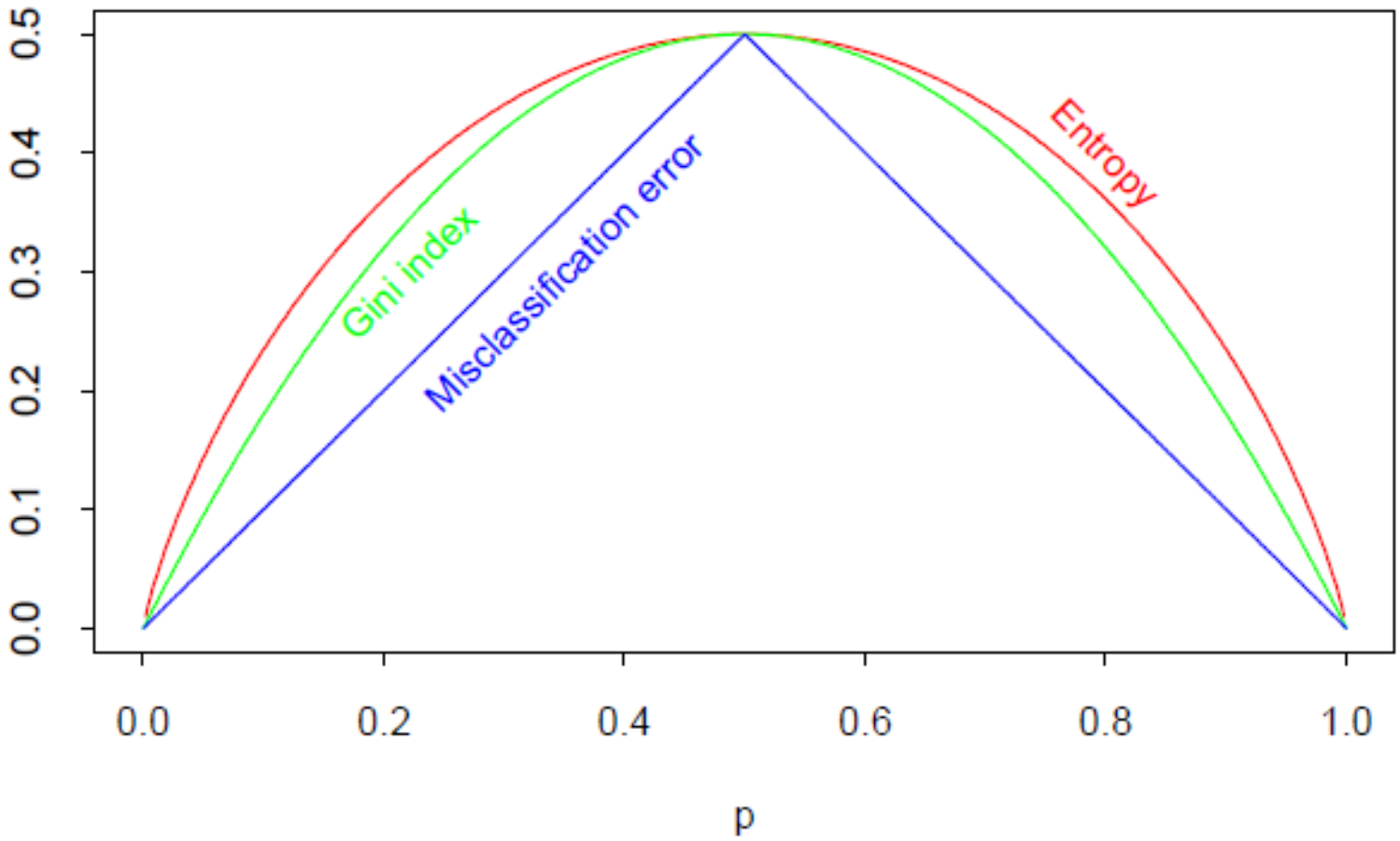
$$D = -\sum_{k=1}^{K} \hat{p}_{\tau k} \log \hat{p}_{\tau k}$$

\# Para 2 clases

Si p es la proporción de la clase 1

- Error de Clasificación: $1 - \max(p, 1-p)$
- Cross-Entropy: $-p \log p - (1-p) \log(1-p)$
- Gini: $2p(1-p)$

Para 2 clases



Obs.: Cross-entropy fue reescalada para pasar por el (0.5,0.5)

Minimizando la Impureza

- Para cada variable (X_j) y en cada nodo (τ), buscamos una división, de modo que la impureza total entre los dos nodos hijos (τ_L y τ_R) sea lo más pequeña posible.
- La bondad de la división se mide por la reducción de impurezas generada por la división s en la variable j-ésima, a partir de los dos nodos hijos

$$\Delta i(\tau, j, s) := i(\tau) - [p_L \, i\,(\tau_L) + p_R \, i\,(\tau_R)]$$

  donde p_L y p_R son la fracción de datos del nodo τ que cae en cada nodo hijo e $i(\tau) = G(\tau)$ o $i(\tau) = D(\tau)$
- Cuanto más grande $\Delta i(\tau, j, s)$, más se reduce la impureza.
- Como $i(\tau)$ no depende de s, en lugar de maximizar $\Delta i(\tau, j, s)$ buscamos

$$\min_s [p_L \, i\,(\tau_L) + p_R \, i\,(\tau_R)]$$

Crecimiento recursivo

- En cada nodo, primero elegimos la mejor división posible para cada variable, y luego seleccionamos la variable que minimiza la impureza total de los nodos hijos.
- Dividimos de forma recursiva todos los nodos (incluidos los nodos hijos) hasta que haya solo una observación en cada nodo: *árbol saturado*. Stop. sobre-especificación / sobreajuste

Criterios de detención anticipada:

- Dejamos de dividir un nodo si $n_\tau \leq n_{min}$ observaciones. (i. e., nodo muy pequeño).
- Dejamos de dividir si $\Delta i(\tau, j, s) < \delta$ (i. e., la mejoría es muy pequeña).

Crecimiento recursivo

- En cada nodo, primero elegimos la mejor división posible para cada variable, y luego seleccionamos la variable que minimiza la impureza total de los nodos hijos.
- Dividimos de forma recursiva todos los nodos (incluidos los nodos hijos) hasta que haya solo una observación en cada nodo: *árbol saturado*. Stop. sobre-especificación / sobreajuste

Criterios de detención anticipada:

- Dejamos de dividir un nodo si $n_\tau \leq n_{min}$ observaciones. (i. e., nodo muy pequeño).
- Dejamos de dividir si $\Delta i(\tau, j, s) < \delta$ (i. e., la mejoría es muy pequeña).

Criterio de Breiman

Breiman et al. (1984) muestran que es mejor criterio dejar crecer el árbol hasta la saturación y luego podarlo usando una función que tenga en cuenta costo y complejidad, complejidad se refiere al número de nodos terminales.

- prune (poda)
- cost–complexity function

Criterio de Breiman

Breiman et al. (1984) muestran que es mejor criterio dejar crecer el árbol hasta la saturación y luego podarlo usando una función que tenga en cuenta costo y complejidad, complejidad se refiere al número de nodos terminales.

- prune (poda)
- cost–complexity function

1. Arboles grandes son penalizados por su gran tamaño, pero harán predicciones perfectas en la muestra.
2. Arboles pequeños recibirán una penalización pequeña por su tamaño, pero sus habilidades de predicción serán limitadas.

$\Longrightarrow$ Compromiso 1–2

Función de costo-complejidad

- $[T]$: número total de nodos de un árbol

Función Costo-complejidad

$$C_\alpha(T) = \sum_{\tau=1}^{[T]} n_\tau \, i_T(\tau) + \alpha \, [T]$$

- El primer término penaliza la impureza (heterogeneidad dentro del nodo)
- El segundo término penaliza la cantidad de nodos.
- Dado un α, buscamos encontrar la poda que minimiza $C_\alpha(T)$. (**Smallest minimizing subtree, Breiman et al, 1984**)
- La elección de α suele hacerse por CV.

Detalles del Algoritmo

1. Utilice la división binaria recursiva para hacer crecer el árbol hacia uno maximal con los datos de entrenamiento, deteniéndose solo cuando cada nodo terminal tiene menos de un número mínimo de observaciones.
2. Aplique la poda de que minimiza la función de costo-complejidad del árbol maximal para obtener una secuencia de los mejores subárboles, en función de α.
3. Utilice la validación cruzada de M-fold para elegir α. Para cada $m = 1, ..., M$:

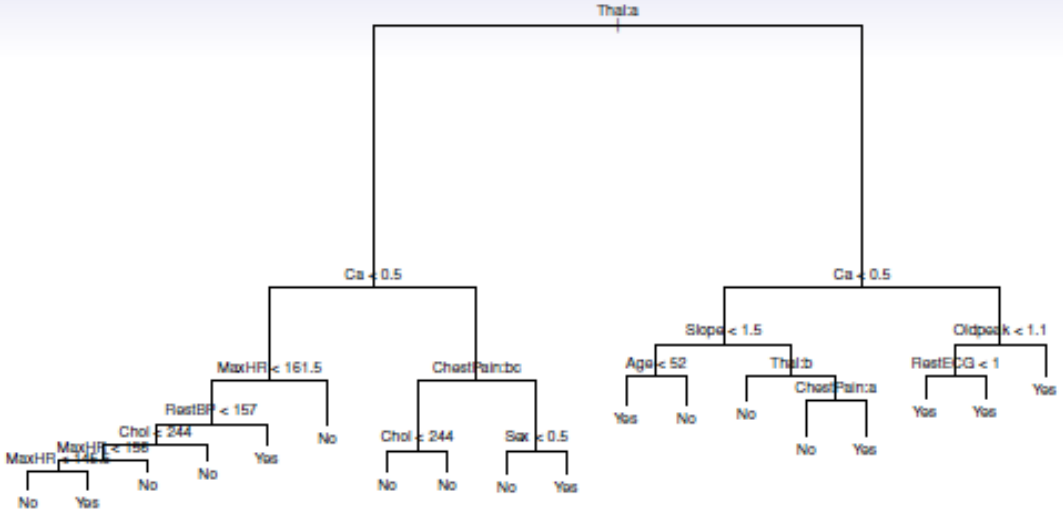
   - 3.1 Para cada fold m repita los pasos 1 y 2 sobre los restantes $M - 1$ folds.
   - 3.2 Calcule el error de clasificación en el $m-$ésimo fold.
   - 3.3 Promedie estos resultados sobre los M folds y elija el α que minimiza el error promedio de clasificación.

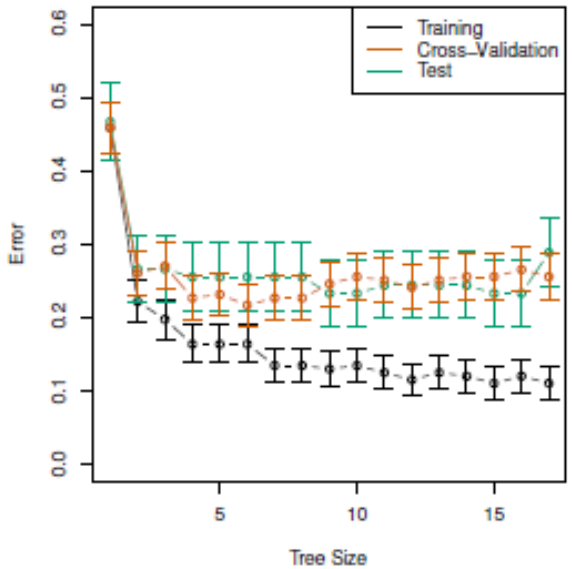
4. Elija del paso 2 el árbol que corresponde al α elegido.

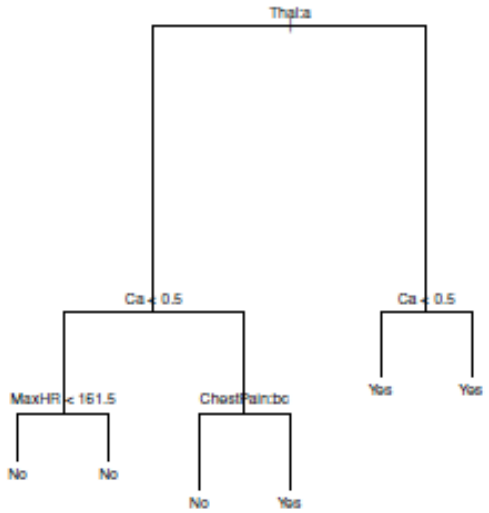
Ejemplo: Heart Data

- Respuesta binaria (enfermedad de corazón si o no) 303 pacientes con angina de pecho.
- Si o No resulta de una angioplastía
- Hay 13 predictores: Edad, Sexo, Colesterol, y otras variables que reflejan el funcionamiento del corazón y pulmones.
- La CV indica un árbol con 6 nodos.

Ejemplo: Heart Data







Ventajas y desventajas de los Arboles

- Son fáciles de explicar e interpretar.
- Se asemejan al proceso de decisión humano.
- Tiene una representación gráfica simple y esto facilita su interpretación por parte de no expertos (en especial si son de tamaño moderado).
- Manejan con facilidad distintos tipos de variables y de interacciones.
- No son tan precisos para la predicción como otras técnicas de regresión y pueden ser bastante inestables.
- Sin embargo, hay alternativas como *bagging* que mejoran mucho su performance mediante la combinación de distintos árboles.

Bagging

- Bagging o Bootstrap aggregation se usa en general para reducir varianza.
- Tengamos en cuenta que si tenemos $V_1, \ldots, V_n$ observaciones independientes con varianza σ^2, su promedio $\overline{V}$ tiene varianza σ^2/n.
- Con lo cual, promediando reducimos variabilidad.

Bootstrap

El bootstrap es una herramienta extremadamente poderosa y de amplia aplicación.

Se puede utilizar para **cuantificar la incertidumbre** asociada a un estimador o método de aprendizaje estadístico dado.

El poder del bootstrap radica en el hecho de que puede ser aplicado a una vasta gama de métodos de aprendizaje estadístico, incluyendo algunos para los que una medida de variabilidad es difícil de obtener y no es generado automáticamente por software estadístico.

Bootstrap: Ejemplo

Definición: llamamos error de una estimación a la estimación del desvío (exacto o aproximado) del estimador con el cual estimamos.

- Estimador: $\widehat{\theta}_n$
- Error Standard del Estimador: se $= \sqrt{V(\widehat{\theta}_n)}$ o se $\approx \sqrt{V(\widehat{\theta}_n)}$

Error de Estimación

$Z_1, Z_2, \ldots, Z_n$ i.i.d $Z_i \sim F$, $\mathbb{E}(Z_i) = \theta, \mathbb{V}(Z_i) = \sigma^2$

- Estimador: $\widehat{\theta}_n = \overline{Z}_n$.
- $se^2 = \mathbb{V}ar(\widehat{\theta}_n) = \mathbb{V}ar(\overline{Z}_n) = \sigma^2/n$
- $\widehat{se}^2 = S^2/n$
- $\widehat{se} = \sqrt{S^2/n}$

Error de estimación para la θ =mediana

- Estimador: $\widehat{\theta}_n = \text{med}(Z_1, \ldots, Z_n)$.
- Error Standard del Estimador:

  $\text{se} = \text{se}(\text{med}(Z_1, \ldots, Z_n)) = \sqrt{\mathbb{V}\{\text{med}(Z_1, \ldots, Z_n)\}}$ =???
- $\widehat{\text{se}}$ =??
- Bootstrap!! $\widehat{\text{se}}_{boot}$

Bootstrap

El enfoque bootstrap nos permite usar una computadora para emular el proceso de obtención de nuevos conjuntos de muestras, de modo que podamos estimar la variabilidad que nos interesa sin generar muestras adicionales.

En lugar de obtener de la población repetidamente conjuntos de datos independientes, obtenemos distintos conjuntos de datos por remuestreo de observaciones del conjunto de datos original que tenemos en la mano.

Bootstrap

1. Seleccionamos aleatoriamente n observaciones del conjunto de datos, obtenemos así la muestra bootstrapeada $\mathcal{Z}^{*1} = \{Z_1^*, Z_2^*, \ldots, Z_n^*\}$.
2. El remuestreo se realiza con reposición, lo que significa que la misma observación puede ocurrir más de una vez en el conjunto de datos original.
3. Usamos $\mathcal{Z}^{*1}$ para producir una nueva estimación, la llamamos $\widehat{\theta}_1^*$.
4. Repetimos los pasos anteriores B veces.

Bootstrap

1. Con las B repeticiones obtenemos las muestras bootstrap $\mathcal{Z}^{*1}, \ldots \mathcal{Z}^{*B}$ y las correspondientes estimaciones $\widehat{\theta}^*_1, \ldots, \widehat{\theta}^*_B$.
2. Computamos el error de estimación a partir de las estimaciones bootstrap

$$\widehat{se}_{boot}(\widehat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\widehat{\theta}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\widehat{\theta}^{*r'}\right)^2} \tag{2}$$

3. El valor computado se usa como estimador del Error Standard del Estimador en cuestión.

Bagging

- Hacemos bootstrap tomando B muestras con reposición de la muestra de entrenamiento.
- Con cada muestra bootstrapeada entrenamos nuestro método y obtenemos un árbol.
- Para cada muestra de testeo, registramos la clase predicha por cada uno de los B árboles y tomamos el voto por mayoría: **predecimos con la clase más votada a partir de las B predicciones**.

Bagging

$\hat{G}(x)$: es un clasificador para una respuesta con K clases.

Asociado, tenemos un vector indicador $\hat{f}(x)$ de la clase predicha (vector de $K-1$ 0 y un 1): $(0,0,1,0,\ldots,0)$ y nos fijamos en que posición está el 1:

$$\hat{G}(x) = \arg\max_k \hat{f}(x)$$

A través del bagging obtenemos

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

que es un vector de proporciones $(p_1,\ldots,p_K)$.

Bagging

Tendremos

$$\hat{G}_{\text{bag}}(x) = \arg\max_k \hat{f}_{\text{bag}}(x)$$

Otra alternativa es estimar las probabilidades de las clases en el punto x a partir de cada árbol y promediarlas.

Curvas de Error

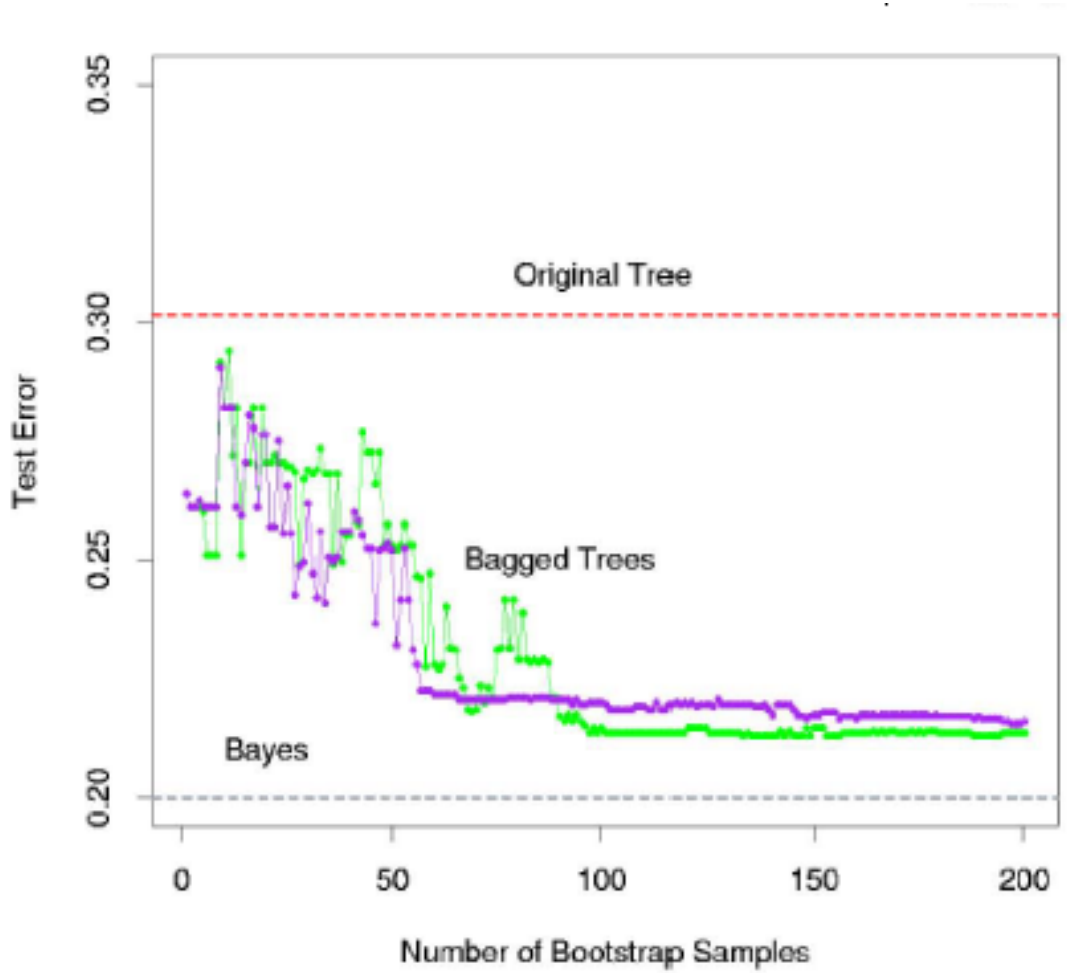


FIGURE 8.10. *Error curves for the bagging example of Figure 8.9. Shown is the test error of the original tree and bagged trees as a function of the number of bootstrap samples. The green points correspond to majority vote, while the purple points average the probabilities.*

Bagging

- Los árboles se dejan crecer profusamente y no se podan.
- Por lo tanto, cada árbol individual tiene una alta varianza, pero un sesgo bajo. Por lo tanto, promedio de estos B árboles reduce la varianza.
- Bagging puede proporcionar grandes mejorias en la predicción, si bien a la hora de interpretar es mucho más dificil.

Importancia de las variables

Con el bagging ganamos en precisión, pero hemos perdido la fácil interpretación de un árbol: ahora tenemos muchos.

precisión vs. interpretabilidad

Sin embargo, podemos computar la cantidad total que el índice de Gini decrece en cada split para cada variable y esto promediarlo sobre los B árboles.

Una representación gráfica de esta medida se muestra en el siguiente gráfico para el ejemplo de Heart Data.

Medida de Inportancia

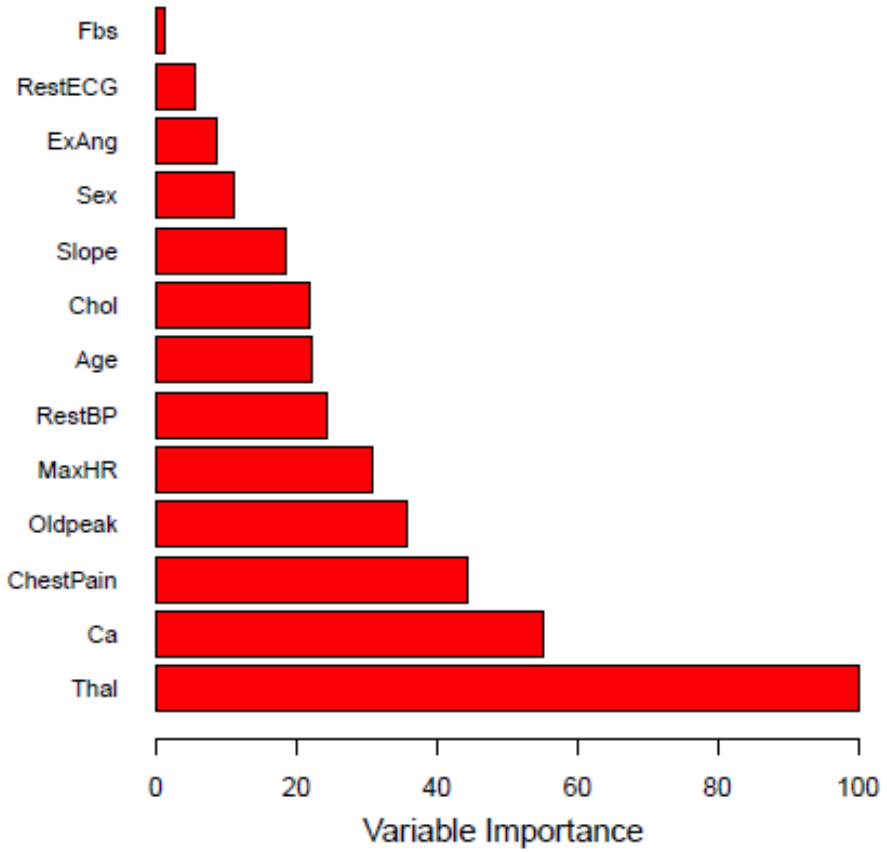


FIGURE 8.9. *A variable importance plot for the* `Heart` *data. Variable importance is computed using the mean decrease in Gini index, and expressed relative to the maximum.*