Aprendizaje supervisado

Maestría en Estadística

Pablo Vena

Universidad de Buenos Aires

12 de noviembre de 2021

Referencias

- Lecturas

ISLR2 Capítulo 8.
ESL Capítulo 10.

- Material
  - Slides Hastie

Árboles de decisión

Veremos métodos basados en **árboles** para regresión y clasificación.

- Consisten en la **estratificación** o **segmentación** del espacio de *features* en una serie de regiones simples.
- El conjunto de reglas de división utilizadas para segmentar el espacio predictor puede describirse con un árbol.

Árboles de decisión

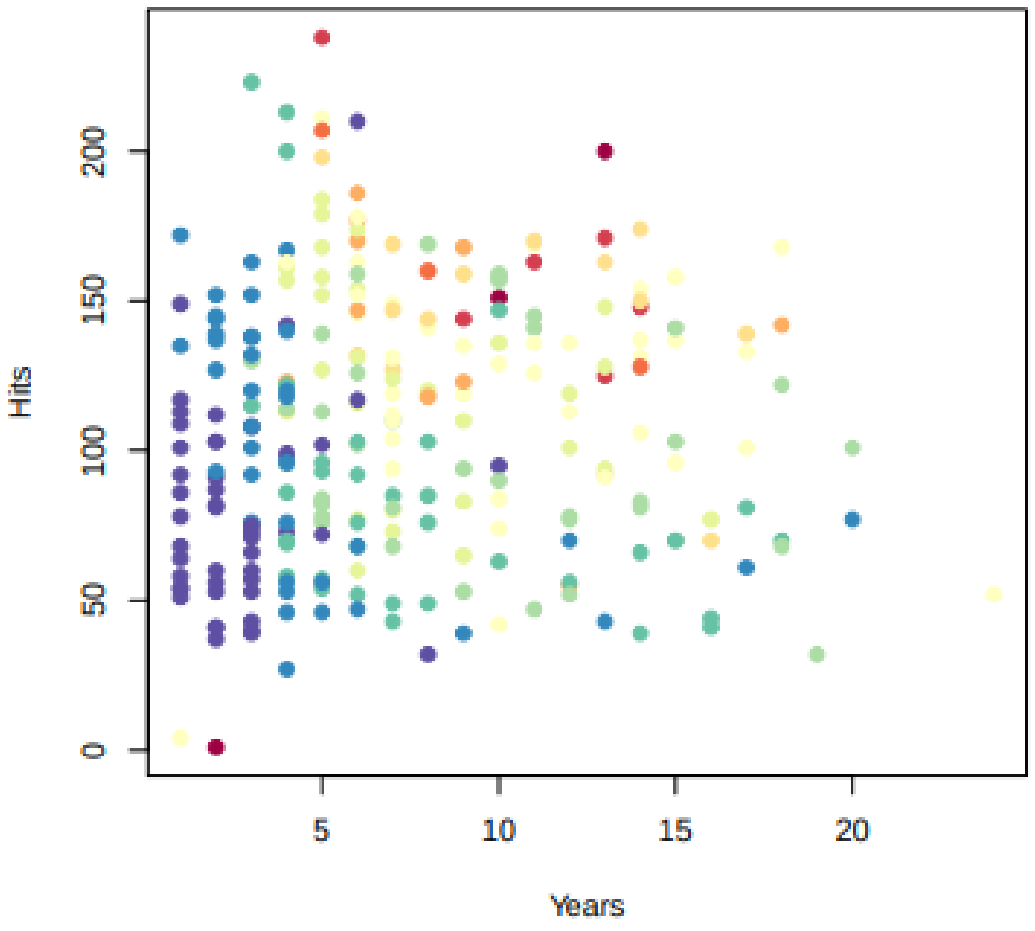
Veremos métodos basados en **árboles** para regresión y clasificación.

- Consisten en la **estratificación** o **segmentación** del espacio de *features* en una serie de regiones simples.
- El conjunto de reglas de división utilizadas para segmentar el espacio predictor puede describirse con un árbol.

Ventajas y desventajas

- Sencillos y útiles para interpretación.
- No suelen ser competitivos con las mejores técnicas aprendizaje supervisado en términos de predicción.
- Veremos **bagging**, **random forests** y **boosting**: se generan y combinan múltiples árboles para obtener una única predicción.
- La combinación de un gran número de árboles a menudo puede mejorar la predicción a expensas de la interpretación.

Árboles de decisión



Construyamos el árbol de regresión

$$\texttt{log(Salary)} \sim \texttt{Years} + \texttt{Hits}$$

Es decir, predecir el *logaritmo del salario* de un jugador de béisbol, basado en el número de

- *años* que ha jugado en las grandes ligas y
- *hits* convertidos en el año anterior.

Árboles de decisión

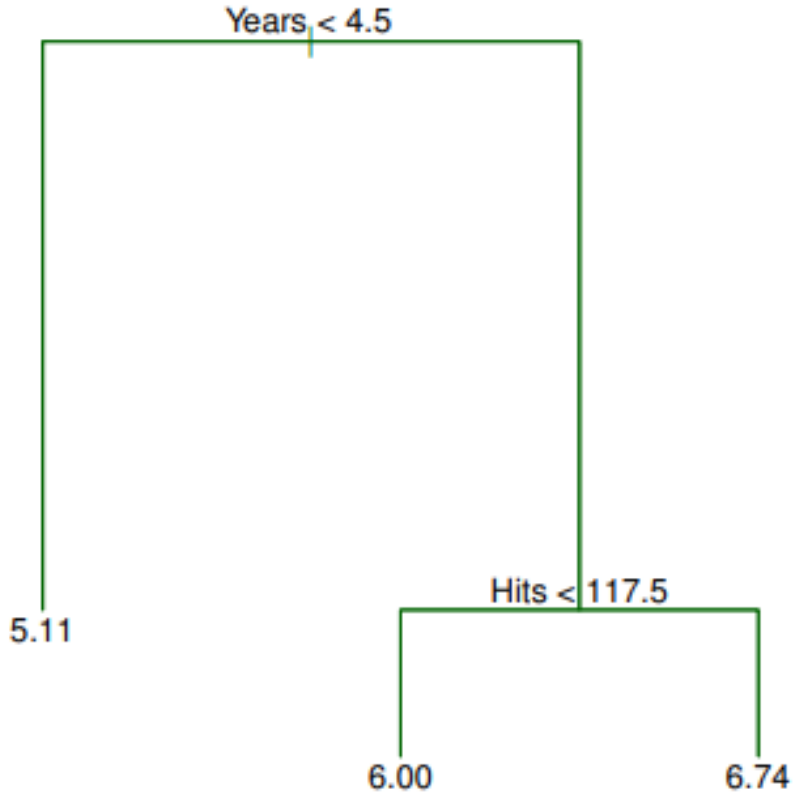


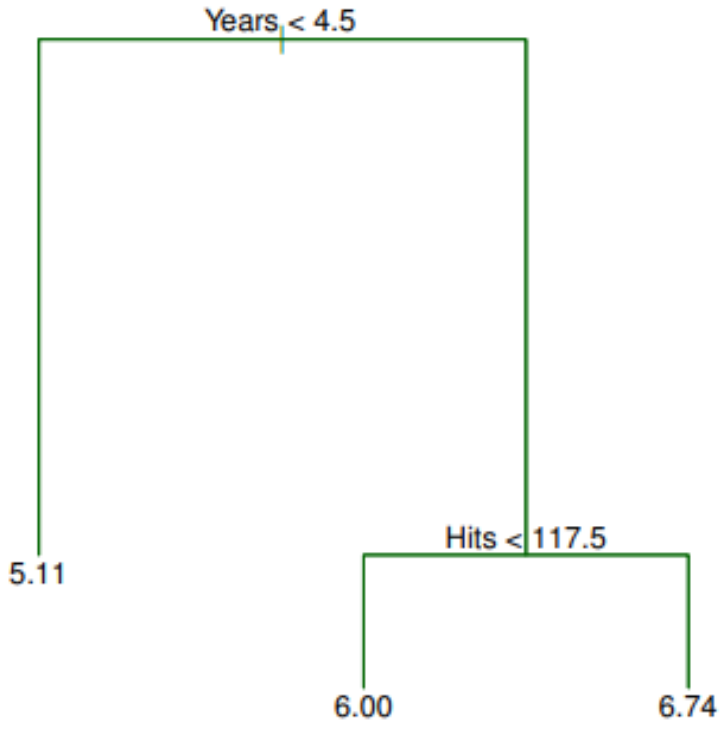
Figura: Árbol de decisión para las variabes `Years` y `Hits`.

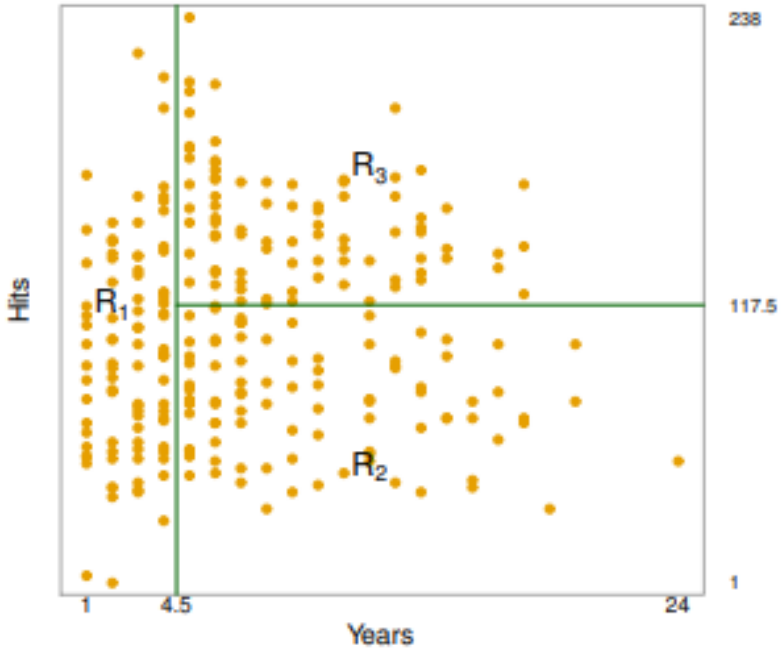
Árboles de decisión

En un nodo interno, la etiqueta (de la forma $X_j < t_k$) indica la rama izquierda que surge de esa división, y la rama derecha corresponde a $Xj \geq t_k$. Por ejemplo, la parte superior del árbol da lugar a dos grandes ramas. La rama de la izquierda corresponde a $Years < 4{,}5$, y la derecha corresponde a $Years \geq 4,5$.

El árbol tiene dos nodos internos y tres nodos terminales, u hojas. El número de cada hoja es la media de la respuesta para las observaciones que caen allí.

Árboles de decisión





Árboles de decisión

El árbol segmenta a los jugadores en tres regiones:

$$
\begin{aligned}
R_1 &= \{X/\texttt{Years} < 4{,}5\},\\
R_2 &= \{X/\texttt{Years} \geq 4{,}5, \texttt{Hits} < 117{,}5\},\\
R_3 &= \{X/\texttt{Years} \geq 4{,}5, \texttt{Hits} \geq 171{,}5\}\,.
\end{aligned}
$$

Definiciones

- La regiones R_1, R_2 y R_3 se conocen como **nodos terminales**.
- Los árboles de decisión suelen dibujarse al revés, en el sentido de que las hojas están en la parte inferior del árbol.
- Los puntos del árbol en los que se divide el espacio de predicción se denominan **nodos internos**.
- En el árbol de bateadores, los dos nodos internos se indican con el texto Years $<$ 4,5 y Hits $<$ 117,5.

Interpretación

- Los años (`Years`) son el factor más importante para determinar el salario, y los jugadores con menos experiencia ganan sueldos más bajos que que los jugadores más experimentados.
- Dado que un jugador tiene menos experiencia, el número de `Hits` que hizo en el año anterior parece tener poco peso en su salario.
- Pero entre los jugadores que llevan cinco o más años en las grandes ligas, el número de hits realizados en el año año anterior sí afecta al salario, y los jugadores que hicieron más hits el año pasado tienden a tener salarios más altos.
- Seguramente es una simplificación excesiva, pero en comparación con un modelo de regresión es fácil de mostrar, interpretar y explicar

Construcción

1. Dividimos el espacio de las covariables, es decir, el conjunto de posibles valores de
$$X_1, X_2, \ldots, X_p$$
en J regiones distintas y disjuntas,
$$R_1, R_2, \ldots, R_J\,.$$
2. Para cada observación que cae en la región R_j hacemos la misma predicción: simplemente la media de los valores de respuesta de las observaciones de entrenamiento en R_j.

Construcción

- Las regiones podrían tener cualquier forma. Sin embargo, elegimos dividir el espacio de predictores en rectángulos, o cajas, para simplificar y facilitar la interpretación del modelo predictivo resultante.
- El objetivo es encontrar las cajas

$$R_1, \ldots, R_J$$

que minimicen el RSS, dado por

$$\sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \widehat{y}_{R_j}\right)^2$$

donde $\widehat{y}_{R_j}$ es la respuesta media de las observaciones de entrenamiento dentro de la j-ésima caja.

Construcción

- Es inviable desde el punto de vista computacional considerar todas las posibles particiones del espacio de *features* en J cajas.
- Adoptamos un enfoque **descendente** y **codicioso** (*greedy*) que conocido como división binaria recursiva.
- Comienza en la parte superior del árbol y luego divide sucesivamente el espacio de predicción; cada división se indica mediante dos nuevas ramas más abajo en el árbol.
- Es **codicioso** porque en cada paso del proceso de construcción del árbol, se realiza la **mejor** división en ese paso en particular, en lugar de mirar hacia adelante y elegir una división que conduzca a un árbol mejor en algún paso futuro.

Construcción

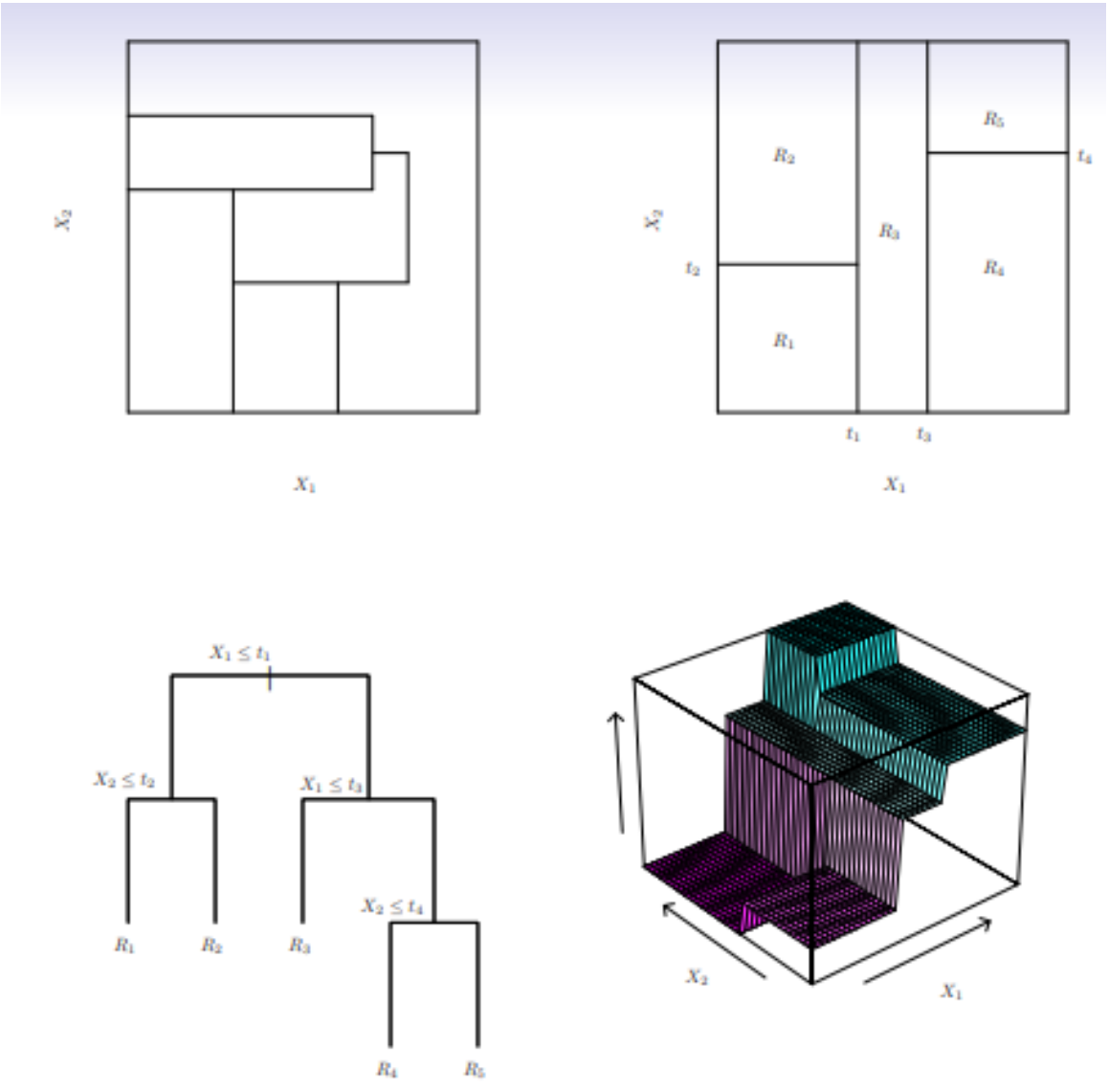
- Primero seleccionamos el predictor X_j y el punto de corte s de forma que que la división del espacio del predictor en las regiones $R_1(j,s) = \{X|X_j < s\}$ y $R_2(j,s) = \{X|X_j \geq s\}$ conduzca a la mayor reducción posible reducción del RSS.

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_1} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]$$

- A continuación, repetimos el proceso, buscando el mejor predictor y el mejor punto de corte para dividir los datos aún más para minimizar el RSS dentro de cada una de las regiones resultantes.

$$\widehat{f}(x) = \sum_{i=1}^{M} \widehat{c}_m I(x \in R_m)$$

Predicciones



Predicciones

Arriba a la izquierda:	una partición del espacio de características bidimensional que no podría resultar de una división binaria recursiva.
Arriba a la derecha:	el resultado de la división binaria recursiva en un ejemplo bidimensional.
Abajo a la izquierda:	un árbol correspondiente a la partición en el panel superior arriba a la derecha.
Abajo a la derecha:	un gráfico en perspectiva de la superficie de predicción correspondiente a ese árbol

Podar un árbol

- El proceso anterior puede producir buenas predicciones en el conjunto de entrenamiento, pero es probable que se **sobreajuste** a los datos.
- Un árbol más pequeño con menos divisiones (es decir, menos regiones $R_1, \ldots, R_j$) podría tener menor varianza y una mejor interpretación a costa de un pequeño sesgo.
- Una posible alternativa al proceso descrito anteriormente es crecer el árbol sólo mientras la disminución del RSS debida a cada división supere algún umbral (alto).
- Esta estrategia dará lugar a árboles más pequeños. Sin embargo, un corte aparentemente inútil al principio del árbol árbol puede ir seguido de una división muy buena, es decir, un corte que conduce a una gran reducción del RSS más adelante.

Podar un árbol

Idea

Crecemos un árbol grande T_0 y lo podamos para obtener un subárbol.

Poda

Realizamos una poda por **complejidad**. Consideramos una sucesión de árboles indexados por un parámetro de ajuste $\alpha \geq 0$. Para cada valor de α se obtiene un subárbol $T \subset T_0$ tal que

$$C_\alpha(T) = \sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \widehat{y}_{R_m})^2 + \alpha \cdot |T|$$

es lo más pequeño posible.

- $|T|$ indica el número de nodos terminales del árbol,
- R_m es la región correspondiente al m-ésimo nodo nodo terminal,
- $\widehat{y}_{R_m}$ es la media de las observaciones de entrenamiento en R_m.

Podar un árbol

Elección del mejor subárbol

- El parámetro de ajuste α controla el equilibrio entre la complejidad del subárbol y su ajuste a los datos de entrenamiento.
- Seleccionamos un valor óptimo $\widehat{\alpha}$ mediante validación cruzada.
- Volvemos al conjunto de datos completo y obtenemos el subárbol correspondiente a $\widehat{\alpha}$.

Algoritmo de árbol

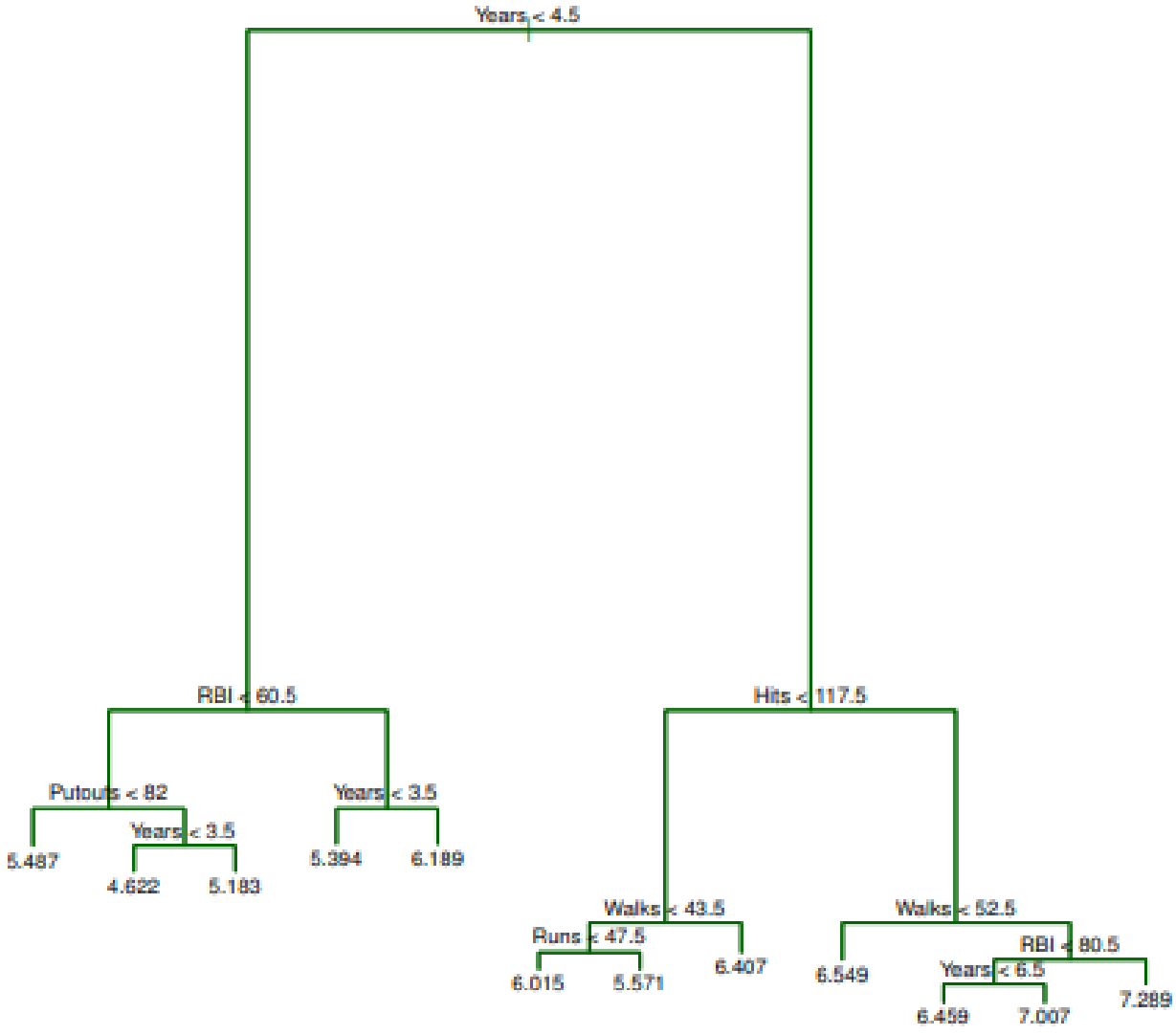
1. Usar la *división binaria recursiva* para construir un gran árbol para los datos de entrenamiento, deteniéndose sólo cuando cada nodo terminal tiene menos de un número mínimo de observaciones.
2. Aplicar una poda por complejidad al árbol grande para obtener una sucesión de los mejores subárboles, en función de α.
3. Utilizar la validación cruzada de K veces para elegir α. Para cada $k = 1, \ldots, K$:
    1. Repita los pasos anteriores en los $\frac{K-1}{K}$ de los datos de entrenamiento datos de entrenamiento, excluyendo el k-ésimo grupo.
    2. Evalúe el error medio de predicción al cuadrado en los datos del de los datos del grupo k, en función de α.
    3. Promedie los resultados y elija α para minimizar el error medio.
4. Devolver el subárbol del paso 2 que corresponde al valor elegido de α.

Ejemplo

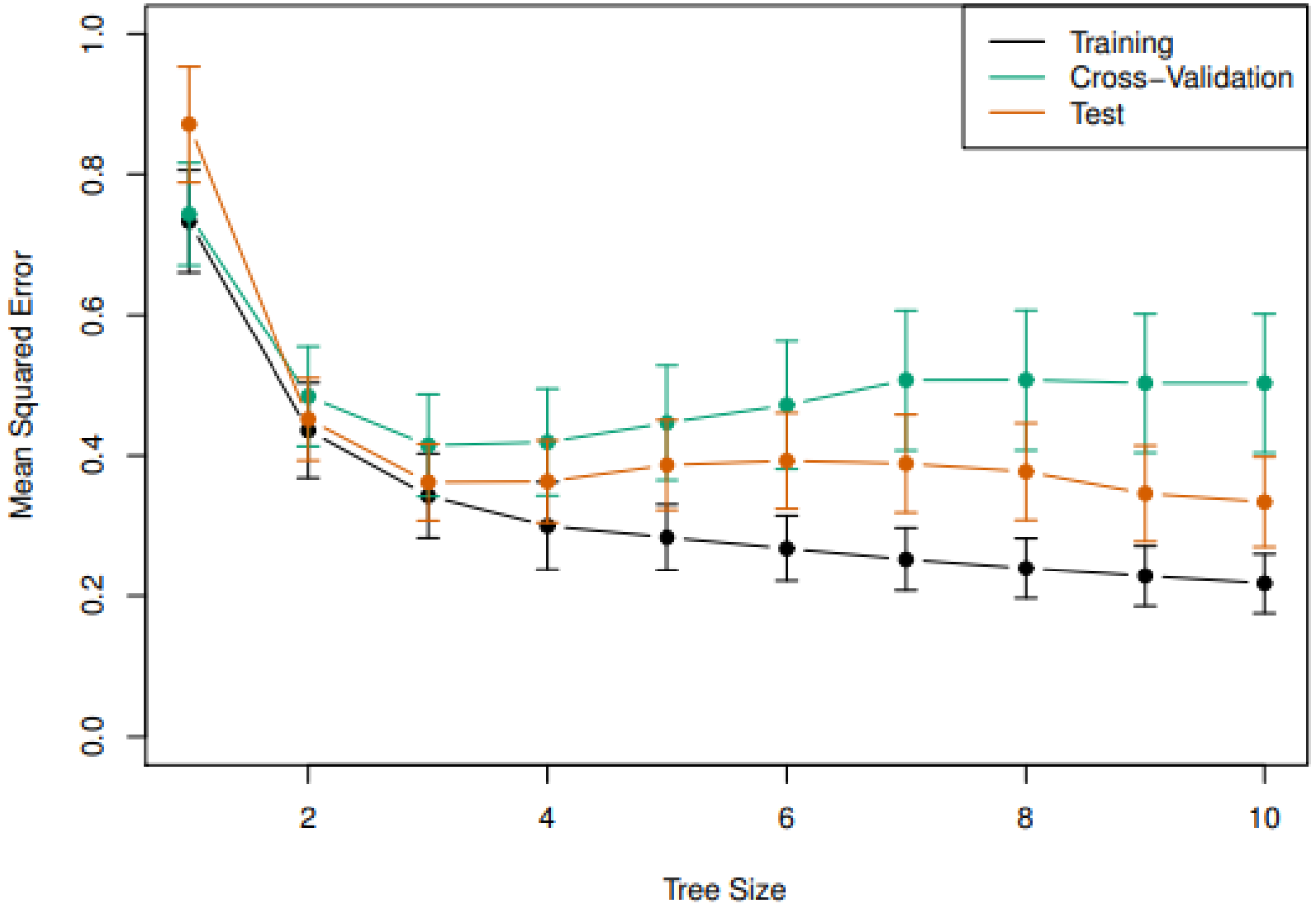
Hitters

- En primer lugar, dividimos aleatoriamente el conjunto de datos por la mitad, obteniendo 132 observaciones en el conjunto de entrenamiento y 131 observaciones en el conjunto de prueba.
- A continuación, construimos un gran árbol de regresión con los datos de entrenamiento y variamos α para crear subárboles con diferentes números de nodos terminales.
- Por último, realizamos una validación cruzada con 6 grupo para estimar el *MSE* validado cruzado de los árboles en función de α.

Ejemplo


Years < 4.5
RBI < 60.5
Hits < 117.5
Putouts < 82
Years < 3.5
Years < 3.5
Walks < 43.5
Walks < 52.5
Runs < 47.5
RBI < 80.5
Years < 6.5
5.487
4.622
5.183
5.394
6.189
6.015
5.571
6.407
6.549
6.459
7.007
7.289

Ejemplo


Training
Cross−Validation
Test
Mean Squared Error
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
Tree Size

Árboles de clasificación

- Muy similar a un árbol de regresión, excepto que se utiliza para predecir una respuesta cualitativa en lugar de una cuantitativa cuantitativa.
- En un árbol de clasificación, se predice que cada observación pertenece a la clase **más frecuente** de las observaciones en la región a la que pertenece.

Árboles de clasificación

- Utilizamos la *división binaria recursiva* para crear un árbol de clasificación.
- En el ámbito de la clasificación, el *RSS* no puede utilizarse como criterio para realizar las divisiones binarias.
- Consideramos la tasa de error de clasificación, la fracción de las observaciones de entrenamiento en que no pertenecen a la clase más común:

$$E = 1 - \max_k(\widehat{p}_{mk}).$$

  donde $\widehat{p}_{mk}$ representa la proporción de observaciones de entrenamiento en la m-ésima región que son de la k-ésima clase.
- Sin embargo, el error de clasificación no es lo suficientemente sensible para el crecimiento del árbol. En la práctica son preferibles otras dos medidas.

Índice de Gini

- El **índice de Gini** se define como

$$G = \sum_{k=1}^{K} \widehat{p}_{mk} \cdot (1 - \widehat{p}_{mk})$$

una medida de la varianza total entre las K clases. El índice de Gini toma un valor pequeño si todos los $\widehat{p}_{mk}$ son cercanos a cero o uno.

- El índice de Gini se conoce como una medida de pureza del nodo: un valor pequeño indica que un nodo contiene un nodo contiene predominantemente observaciones de una sola clase.
- Una alternativa es la **entropía cruzada** (*cross-entropy*), dada por

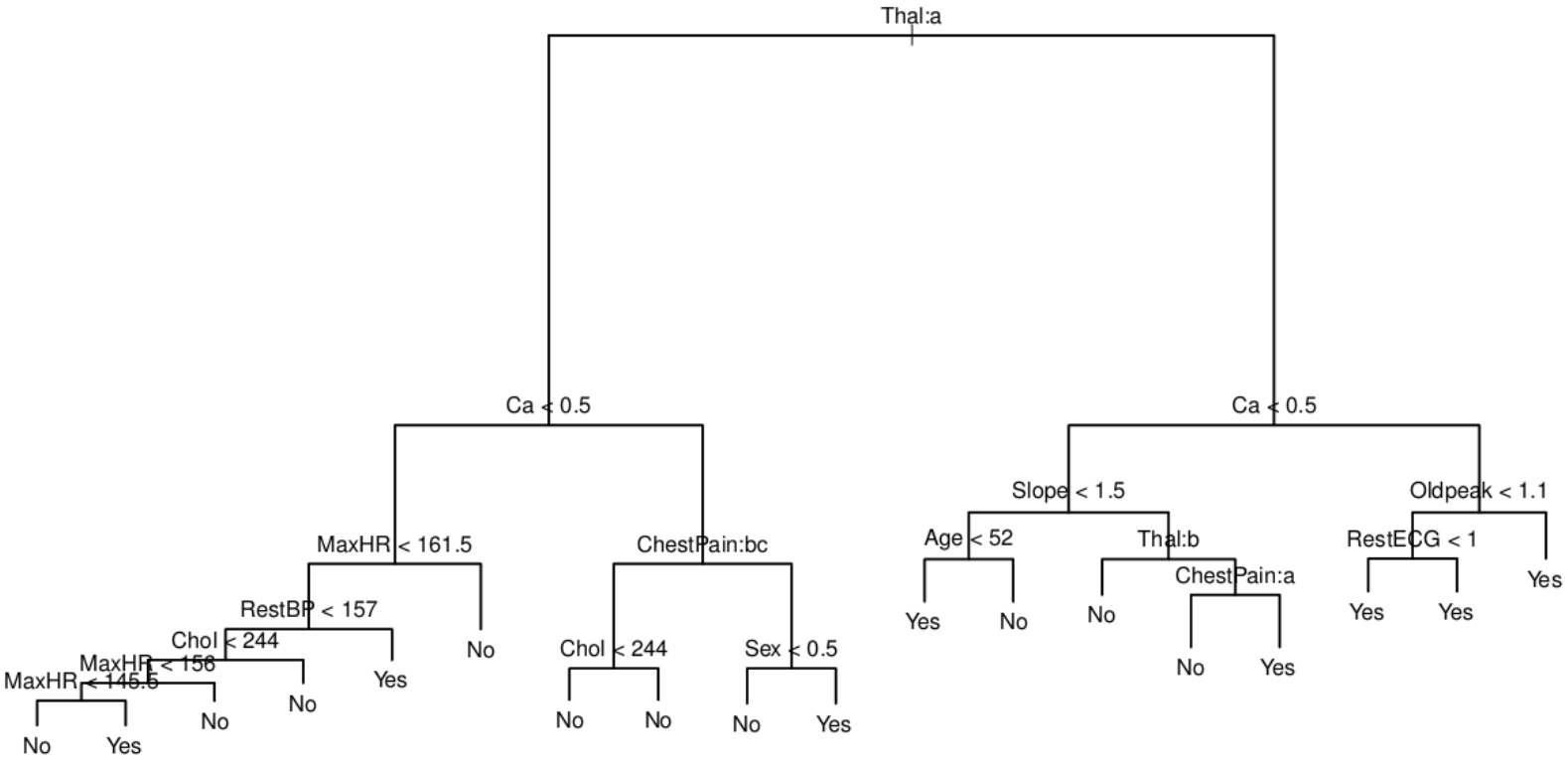
$$D = -\sum_{k=1}^{K} \widehat{p}_{mk} \log \widehat{p}_{mk}$$

Ejemplo

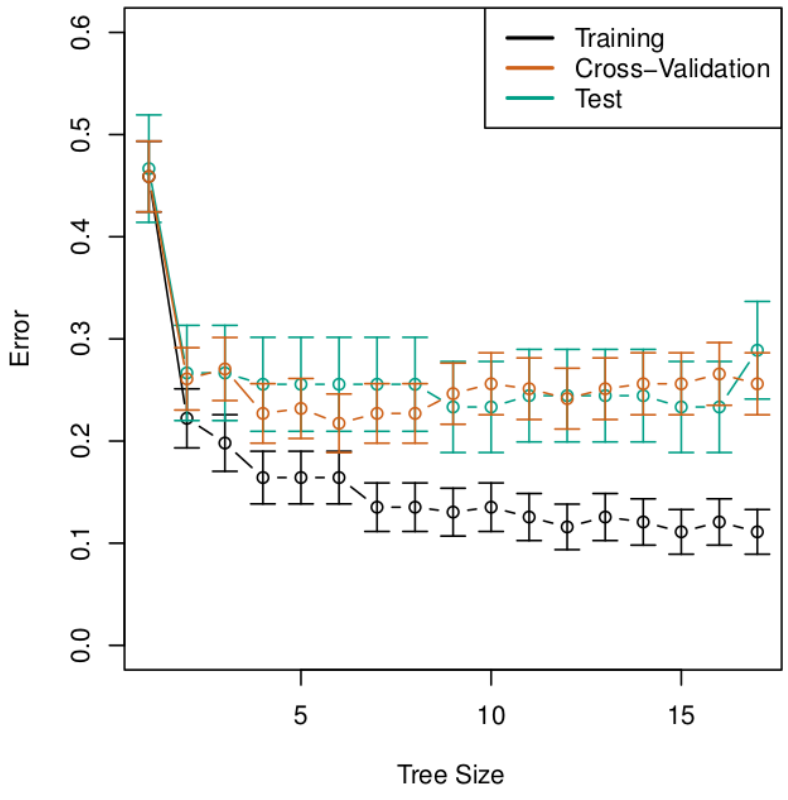
heart

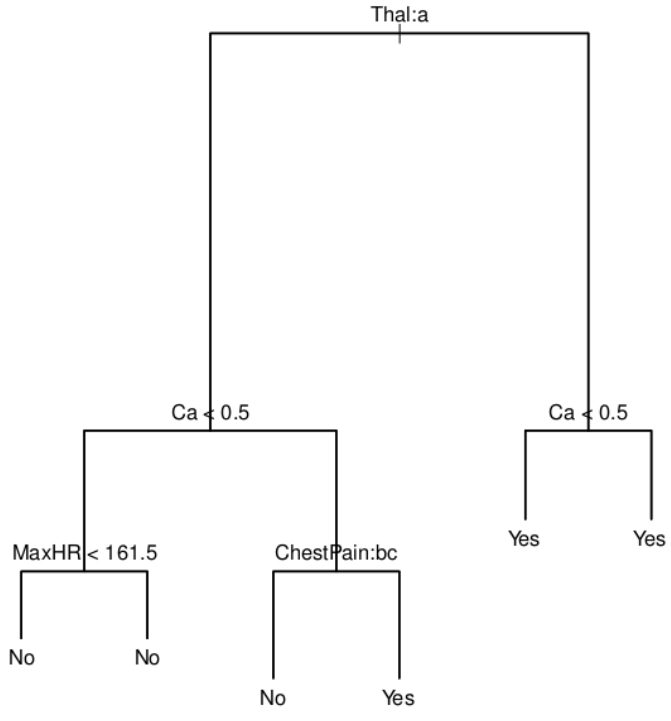
- Estos datos contienen un resultado binario `HD` para 303 pacientes que presentaron dolor torácico.
- Un valor de resultado de `Yes` indica la presencia de enfermedad cardíaca basada en una prueba angiográfica, mientras que `No` significa que no hay enfermedad cardíaca.
- Hay 13 predictores que incluyen edad (`Age`), el sexo (`Sex`), el colesterol (`Chol`) y otras mediciones de la función cardíaca y pulmonar.
- La validaciíon cruzada da como resultado un árbol con seis nodos terminales.

Ejemplo

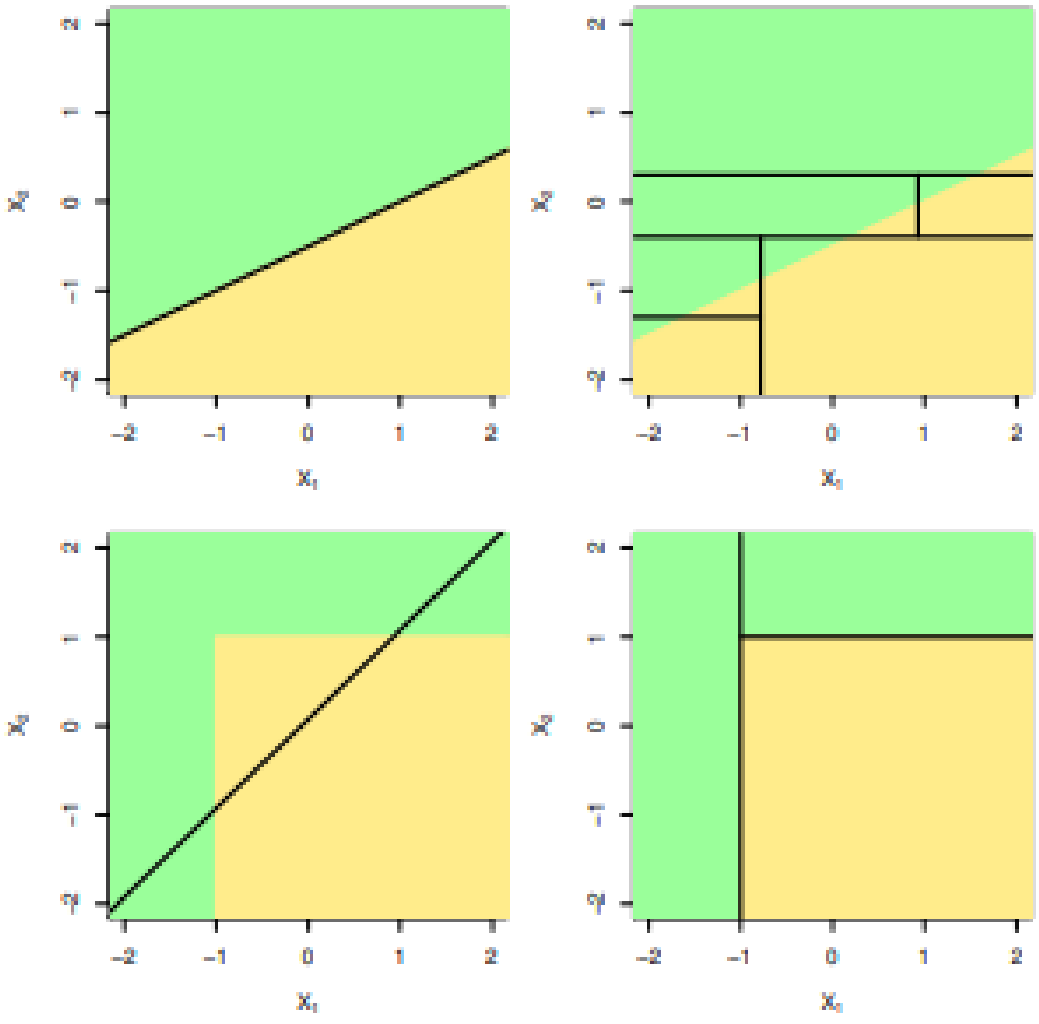

Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Slope < 1.5
Oldpeak < 1.1
Age < 52
Thal:b
RestECG < 1
RestBP < 157
Chol < 244
Chol < 244
Sex < 0.5
ChestPain:a
MaxHR < 156
MaxHR < 145.5
No
Yes
No
No
No
Yes
No
No
No
Yes
Yes
No
No
Yes
No
Yes
Yes
Yes
Yes

Ejemplo


Training
Cross−Validation
Test
Error
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
Tree Size


Thal:a
Ca < 0.5
Ca < 0.5
MaxHR < 161.5
ChestPain:bc
Yes
Yes
No
No
No
Yes

Árboles contra el modelo lineal



1. Frontera lineal real,
2. frontera no lineal real,
3. modelo lineal,
4. modelo basado en árboles

Ventajas y desventajas

- Los árboles son muy fáciles de explicar, incluso más que la regresión lineal.
- Los árboles de decisión reflejan mejor la toma de decisiones.
- Pueden mostrarse gráficamente y son fáciles de interpretar incluso por un no experto (especialmente si son pequenõs).
- Los árboles pueden manejar predictores cualitativos sin necesidad de crear variables *dummy*.

- Por desgracia, los árboles no suelen tener el mismo nivel de precisión predictiva que otros métodos de regresión.

Bagging

La **agregación Bootstrap**, o **bagging**, es un procedimiento de propósito general para reducir la varianza de un método de aprendizaje estadístico.

- Dado un conjunto de n observaciones independientes

$$Z_1, \ldots, Z_n$$

con varianza σ^2, la varianza de la media muestral $\bar{Z}$ es $\frac{\sigma^2}{n}$.

- Promediar un conjunto de observaciones reduce la varianza. Por supuesto, esto no es práctico porque generalmente no tenemos acceso a múltiples conjuntos de entrenamiento.
- En su lugar, podemos hacer **bootstrap**: tomamos muestras repetidas del (único) conjunto de datos de entrenamiento.

Bagging

Regresión

1. Generamos B conjuntos de datos de entrenamiento diferentes.
2. Entrenamos nuestro método en el b-ésimo conjunto de entrenamiento para obtener $\widehat{f}^{*b}(x)$, la predicción en un punto x.
3. Promediamos todas las predicciones para obtener

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

Clasificación

Para cada observación de test registramos la clase predicha por cada uno de los árboles B y tomamos un voto mayoritario: la predicción global es la clase más frecuente entre las predicciones de los árboles B.

Bagging

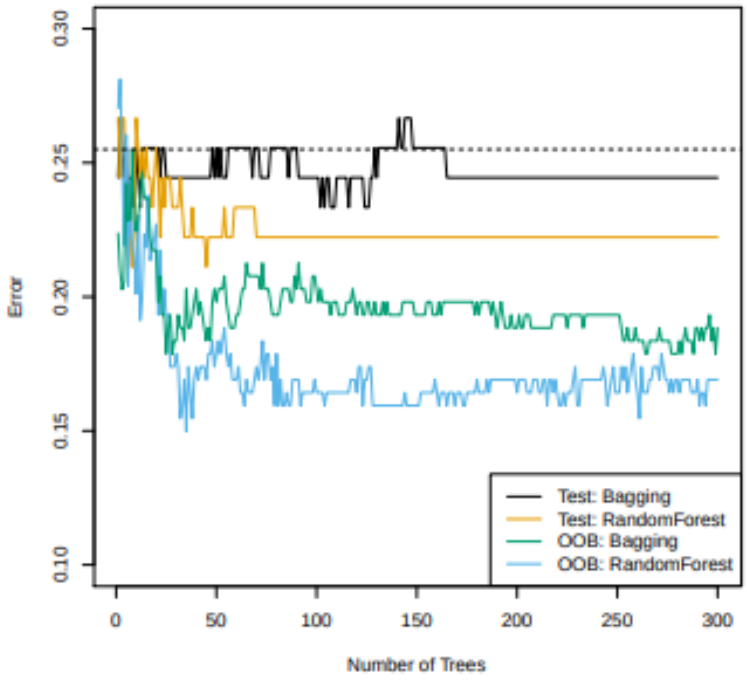


Figura: Error de test en función de la cantidad de árboles (`Heart`).

- La linea punteada indica el error de test para un solo árbol de clasificación.
- *Random forests* con $m = \sqrt{p}$.
- La líneas verde y azul muestran el error OOB.

Estimación del error *out of bag*

- Recordemos que los árboles se ajustan repetidamente a subconjuntos de las observaciones. Se puede demostrar que, en promedio, cada árbol utiliza alrededor de dos tercios de las observaciones.
- El tercio restante de las observaciones que no se utilizan para ajustar un determinado árbol se denominan observaciones *out of bag* (OOB)
- Podemos predecir la respuesta de la i-ésima observación utilizando cada uno de los árboles en los que esa observación estaba fuera de bolsa. Este se obtienen alrededor de $B/3$ predicciones para la i-ésima observación, que promediamos.
- Esta estimación es esencialmente el error de validación cruzada *LOO* para *bagging*, si B es grande.

Bosques aleatorios (random forest)

- Los **bosques aleatorios** ofrecen una mejora con respecto a *bagging* por medio de un pequeño retoque que **descorrelaciona** los árboles. Esto reduce la varianza cuando promediamos los árboles.
- Al igual que en el embolsado, construimos una serie de árboles de decisión en muestras de entrenamiento.
- Pero al construir estos árboles de decisión, cada vez que se considera una división en un árbol, se eligen al azar m predictores como candidatos a la división del conjunto completo de p predictores. La división sólo puede utilizar uno de esos m predictores.
- Se toma una nueva selección de m predictores en cada división, y normalmente elegimos $m \approx \sqrt{p}$.

Ejemplo: datos de expresión genética

- Aplicamos *random forest* a un conjunto de datos biológicos de de alta dimensión que consisten en mediciones de expresión de 4718 genes medidos en muestras de tejido de 349 pacientes.
- En los seres humanos hay unos 20,000 genes, y cada uno de ellos genes tienen diferentes niveles de actividad, o expresión, en determinadas células, tejidos y condiciones biológicas.
- Cada una de las muestras de pacientes tiene una etiqueta cualitativa con 15 niveles diferentes: normal o uno de los 14 tipos diferentes de cáncer.
- Utilizamos *random forest* para predecir el tipo de cáncer basándonos en los 500 genes que tienen la mayor varianza en el conjunto de entrenamiento.
- Dividimos aleatoriamente las observaciones en un conjunto de entrenamiento y otro de prueba, y ajustamos el conjunto de entrenamiento para tres valores diferentes del número de variables de división m.

Ejemplo: datos de expresión genética

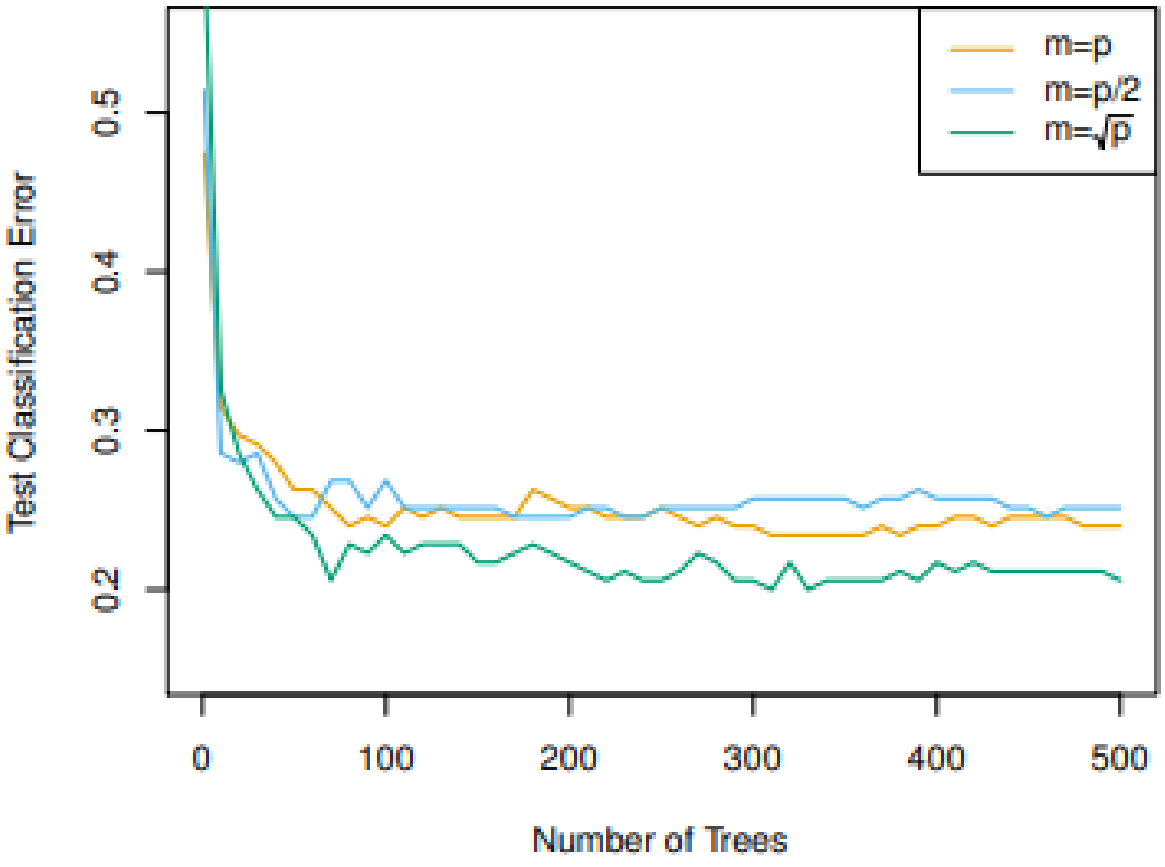


Figura: $p = 500$ predictores y $K = 15$ clases.

Boosting

- Al igual que el *bagging*, el **boosting** es un enfoque general que puede aplicarse a muchos métodos de aprendizaje estadístico para regresión o clasificación.
- Recordemos que el *bagging* implica la creación de múltiples copias del conjunto e datos de entrenamiento original utilizando el *bootstrap*, ajustando un de decisión a cada copia y, a continuación, combinar todos los de los árboles para crear un único modelo predictivo.
- En particular, cada árbol se construye sobre un conjunto de datos bootstrap independiente de los demás árboles.
- El *boosting* funciona de forma similar, salvo que los árboles se generan secuencialmente: cada árbol se construye utilizando la información de los árboles anteriores.

Boosting

1. Fijamos $\widehat{f}(x) = 0$ y $r_i = y_i$ para todos los i en el conjunto de entrenamiento.
2. Para $b = 1, 2, \ldots, B$, repetir:
   1. Ajustamos un árbol $\widehat{f}^b$ con d cortes ($d + 1$ nodos terminales) al conjunto de entrenamiento (X, r).
   2. Actualizar $\widehat{f}$ agregando una versión encogida del nuevo árbol:
   $$\widehat{f}(x) \leftarrow \widehat{f} + \lambda \widehat{f}^b(x).$$
   3. Actualizar los residuos,
   $$r_i \leftarrow r_i + \lambda \widehat{f}^b(x_i).$$
3. Devolver el modelo,
$$\widehat{f}(x) = \sum_{b=1}^{B} \lambda \widehat{f}^b(x)\,.$$

¿Cuál es la idea de este procedimiento?

- A diferencia de ajustar un único árbol de decisión de gran tamaño a los datos, que equivale a un ajuste duro de los datos y potencialmente a un sobreajuste, el enfoque de *boosting* aprende lentamente.
- Dado el modelo actual, ajustamos un árbol de decisión a los residuos del modelo. A continuación, añadimos este nuevo árbol de decisión la función ajustada para actualizar los residuos.
- Cada uno de estos árboles puede ser bastante pequeño, con sólo unos pocos nodos terminales, determinados por el parámetro d en el algoritmo.
- Al ajustar pequeños árboles a los residuos, mejoramos lentamente $\widehat{f}$ en las áreas en las que no funciona bien. El parámetro λ ralentiza el proceso, permitiendo que más árboles y de diferente forma ataquen a los residuos.

Boosting

- El boosting para la clasificación es similar en espíritu al boosting para la regresión, pero es un poco más complejo.
- Pueden conocer los detalles en Elements of Statistical Learning, capítulo 10.
- El paquete de R `gbm` (*gradient boosted models*) maneja una variedad de problemas de regresión y clasificación.

Ejemplo: datos de expresión genética

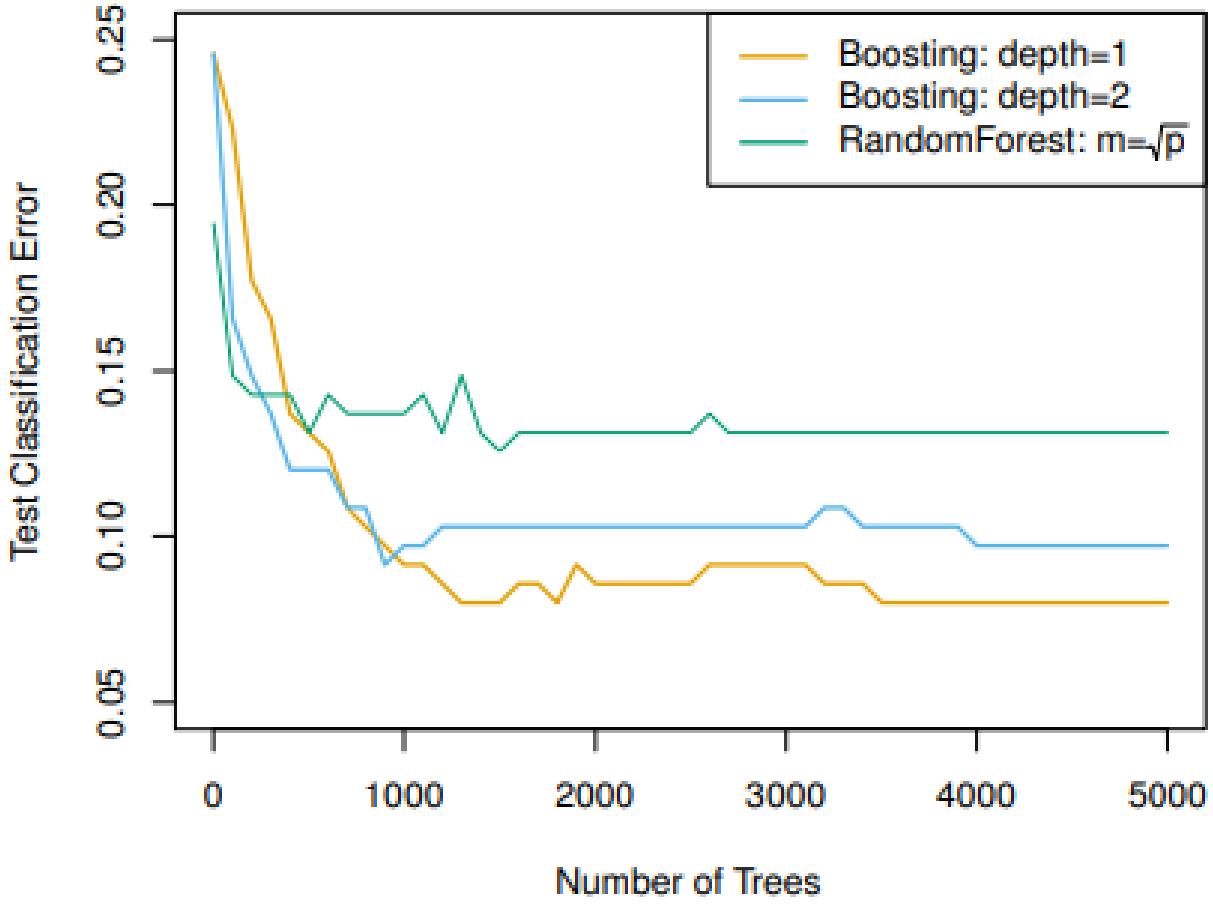
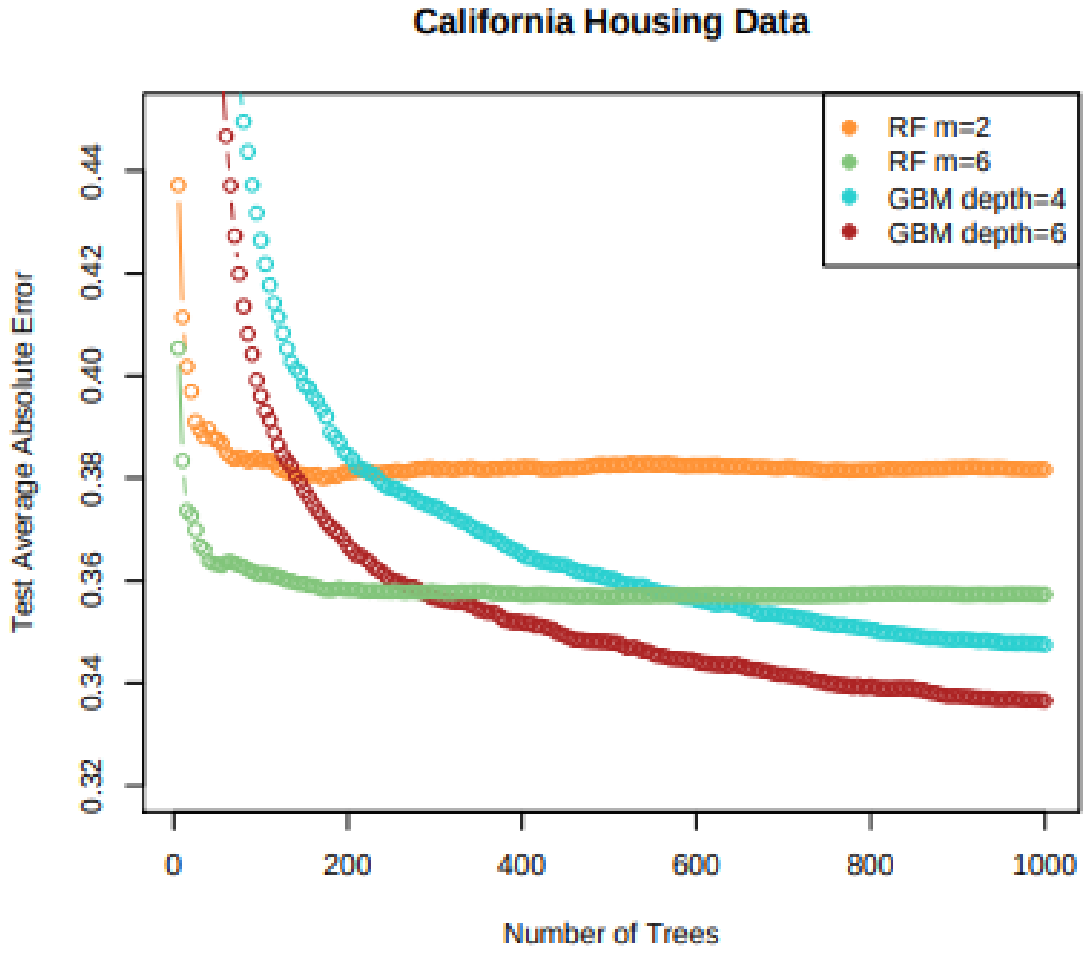


Figura: La tasa de error de test para un solo árbol es del 24 %.

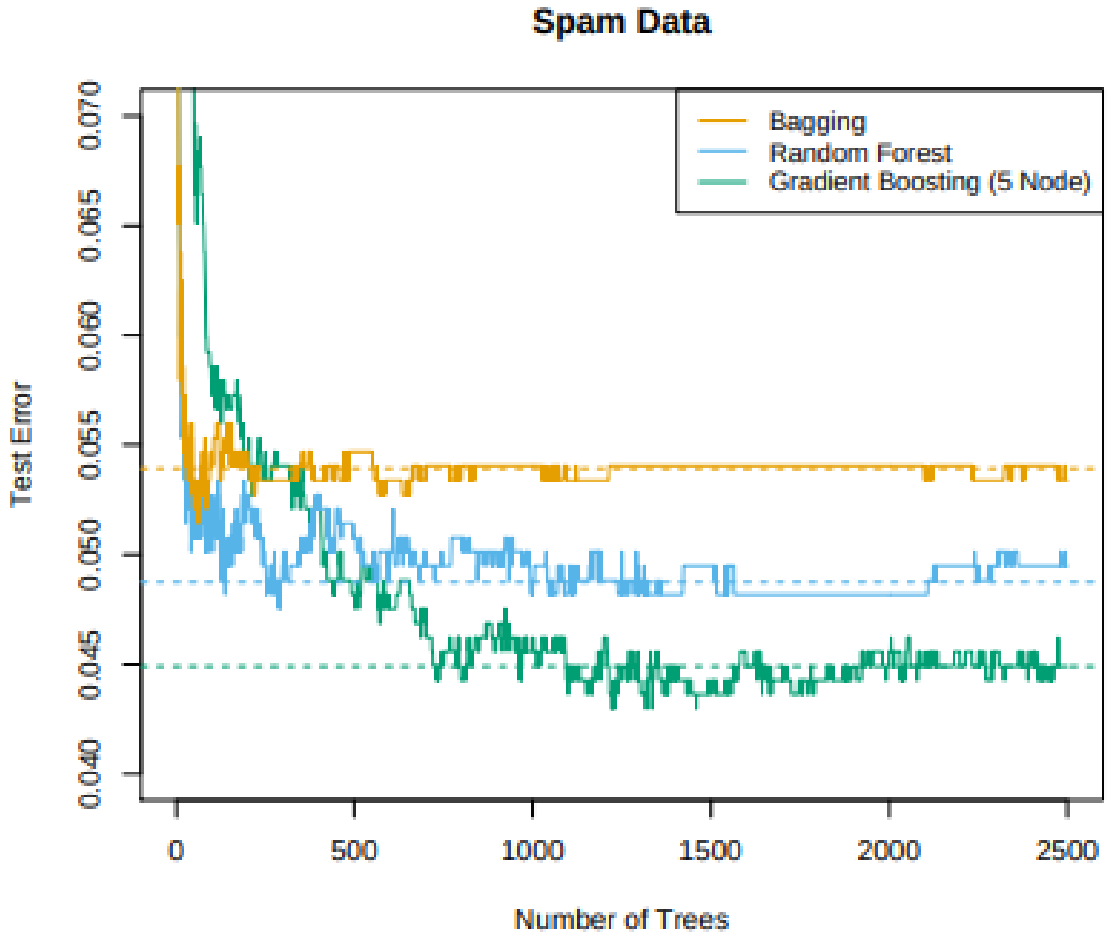
Parámetros de ajuste para el boosting

1. El número de árboles B. Puede haber sobreajuste si B es demasiado grande. Utilizamos validación cruzada para seleccionar B.
2. El parámetro λ. Controla la velocidad de aprendizaje de boosting. Los valores típicos son 0,01 o 0,001, y la elección correcta puede depender del problema. Un λ muy pequeño puede requerir el uso de un valor muy grande de B para conseguir un buen rendimiento.
3. El **número de divisiones** d en cada árbol, que controla la complejidad del ensamble. A menudo $d = 1$ funciona bien, en cuyo caso cada árbol consiste en una de una sola división y que da lugar a un modelo aditivo. En general, d es la profundidad de la interacción y controla el orden de orden de interacción del modelo ya que las divisiones d pueden implicar a lo sumo d variables.

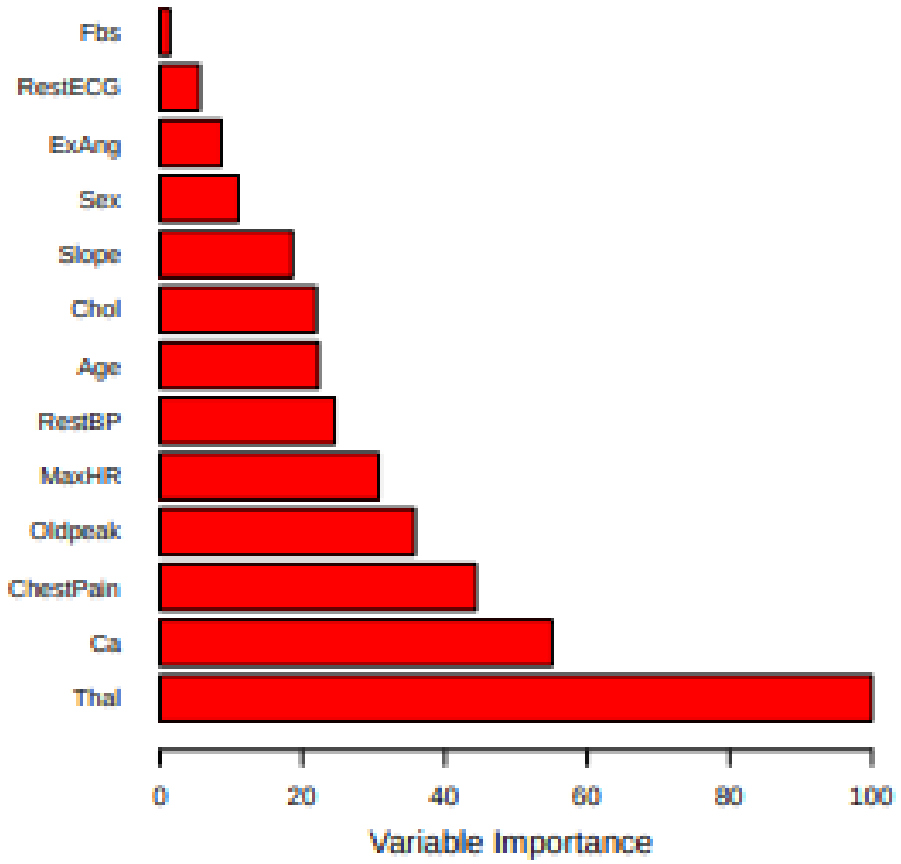
Ejemplo: Boston



Ejemplo: Spam



Importancia de las variables



- Para árboles de regresión (bagged o RF), registramos el RSS disminuido por los cortes de un predictor dado, promediamos sobre los B árboles. Un valor alto indica que es un predictor importante.
- Similarmente, para clasificación, sumamos el Gini disminuido por los cortes de un predictor dado, promediamos sobre los B árboles.

Resumen

- Los árboles de decisión son modelos simples e interpretables para regresión y clasificación
- Sin embargo, a menudo no son competitivos con otros métodos en términos de precisión de predicción
- El *bagging*, *random forest* y el *boosting* son buenos métodos para mejorar la precisión de predicción de los árboles. Funcionan funcionan creciendo muchos árboles en los datos de entrenamiento y combinando las predicciones del conjunto de árboles resultante.
- Los dos últimos métodos, *random forest* y *boosting*se encuentran entre los métodos más avanzados de aprendizaje supervisado. Sin embargo, sus resultados pueden ser difíciles de interpretar.