Métodos Basados en Arboles.

César Gómez

19 de octubre de 2020

Resumen

Estos métodos envuelven *estratificar* o *segmentar* el espacio de predictores en un número simple de regiones.
Como el conjunto de reglas utilizadas para segmentar el espacio de predictores, puede resumirse en un diagrama de arbol, estos métodos resiven el nombre de **árboles de decisión**.
En este capítulo vamos a cubrir entre otros conceptos:

- *Bagging*.
- *Random Forest*.
- *Boosting*.

Arboles de regresión

Vamos a ajustar un árbol de regresión para predecir el salario de un jugador de baseball basandose en el número de años que ha jugado y el número de hits (batazos) que ha realizado

RStudio Source Editor

hitters1

Filter

	Hitters.Salary	Hitters.Years	Hitters.Hits
-Andy Allanson	*NA*	1	66
-Alan Ashby	475.000	14	81
-Alvin Davis	480.000	3	130
-Andre Dawson	500.000	11	141
-Andres Galarraga	91.500	2	87
-Alfredo Griffin	750.000	11	169
-Al Newman	70.000	2	37
-Argenis Salazar	100.000	3	73
-Andres Thomas	75.000	2	81
-Andre Thornton	1100.000	13	92
-Alan Trammell	517.143	10	159
-Alex Trevino	512.500	9	53
-Andy VanSlyke	550.000	4	113

El conjunto de datos `Hitters` contiene originalmente 20 variables entre ellas `Salary` (salario) en 322 observaciones.

Arboles de regresión

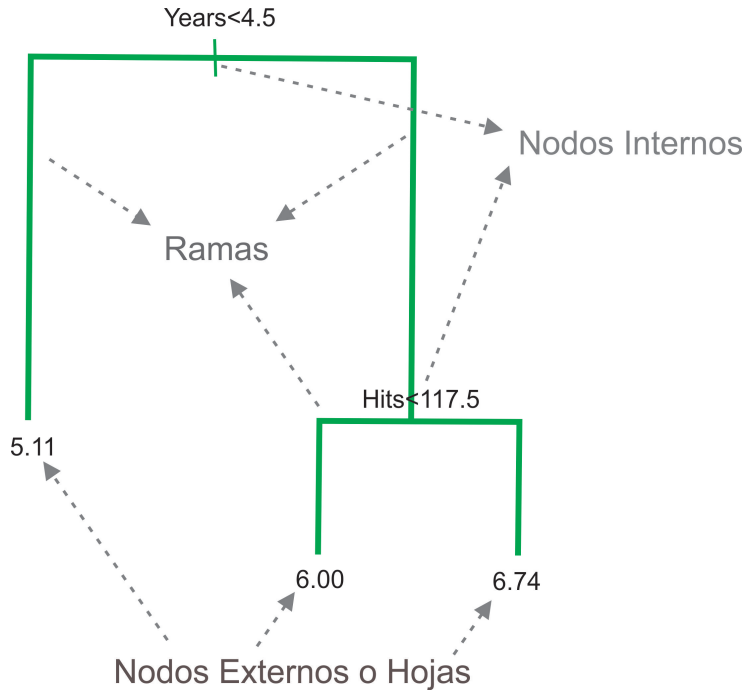

Figura 1: Para el conjunto de datos Hitters, se ilustra un arbol de regresión para predecir el log del salario de un jugador de béisbol, basado en el número de años que él ha jugado en grandes ligas y en el número de hits que ha hecho en el año anterior.

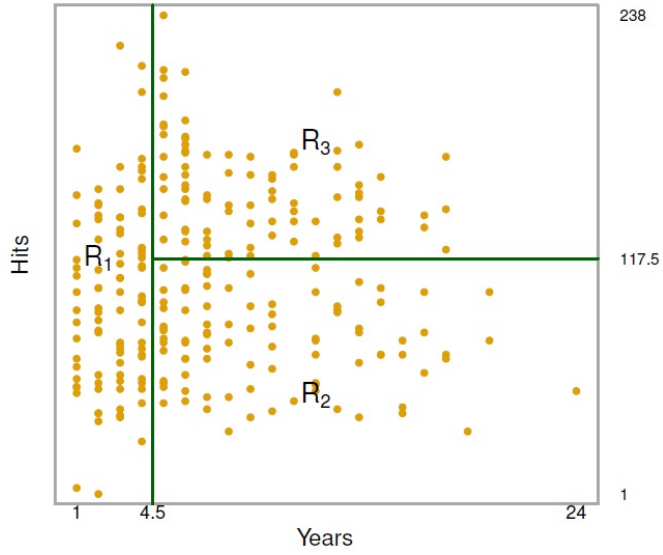

Figura 2: Acá la partición del espacio de predictores correspondiente al arbol de la figura 1.

Las Regiones R_1, R_2 y R_3. Se especifican como

$$R_1 = \{X | \text{Years} < 4.5\}.$$

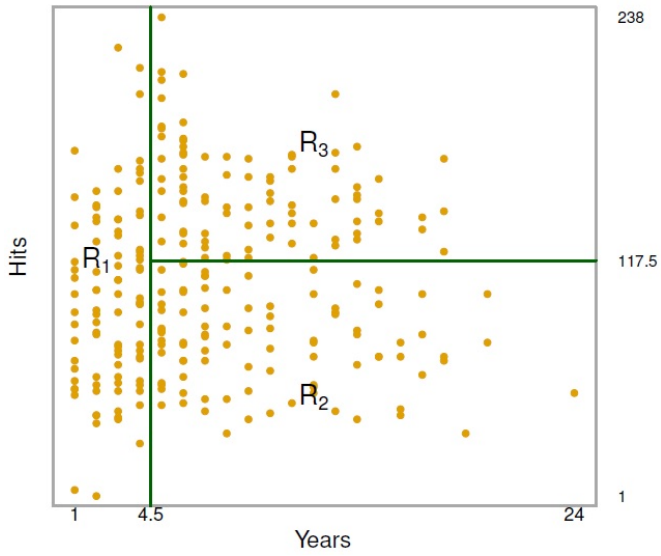

Figura 2: Acá la partición del espacio de predictores correspondiente al arbol de la figura 1.

Las Regiones R_1, R_2 y R_3. Se especifican como

$$R_1 = \{X | \text{Years} < 4.5\}.$$

- La predicción para el Log-Salario de una observación en la región R_1 corresponde a la media en esta región y es según el árbol es 5.11.

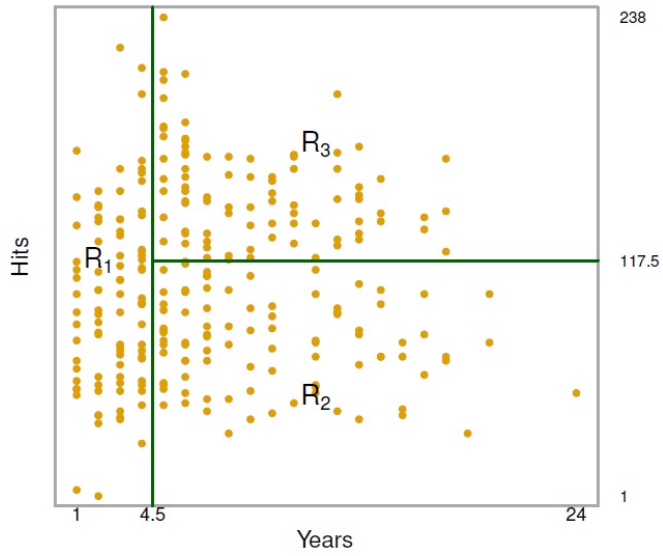

Figura 2: Acá la partición del espacio de predictores correspondiente al arbol de la figura 1.

Las Regiones R_1, R_2 y R_3. Se especifican como

$$R_1 = \{X | \text{Years} < 4.5\}.$$

- La predicción para el Log-Salario de una observación en la región R_1 corresponde a la media en esta región y es según el árbol es 5.11.
- La predicción del salario para una observación en esta región, sería

$$1000 \times e^{5.11} \approx \$165670.$$

- La predicción del Log-Salario para un punto en la región R_2

$$R_2 = \{X | \text{Years} \geq 4.5, \text{Hits} < 117.5\}.$$

corresponde a la media del Log-Salario para los puntos en esta región, según el árbol está media tendría el valor de 6.00. El Salario promedio para los puntos en esta región sería entonces

$$1000 \times e^{6.00} \approx \$403428.$$

- La predicción del Log-Salario para un punto en la región R_3

$$R_3 = \{X | \text{Years} \geq 4.5, \text{Hits} \geq 117.5\}.$$

corresponde a la media del Log-Salario para los puntos en esta región, según el árbol está media tendría el valor de 6.74. El Salario promedio para los puntos en esta región sería entonces

$$1000 \times e^{6.74} \approx \$845560.$$

Interpretación del árbol

El árbol de regresión ajustado puede ser interpretado de la siguiente manera:

- `years` (años de experiencia) es el factor más importante para determinar el salario y los jugadores con menos experiencia gana salarios más bajos que los jugadores más experimentados.
- Para un jugador con menos años de experiencia, el número de `Hits` (batazos) que hizo en el año anterior parece jugar poco papel en su salario. Pero entre jugadores que han estado en las ligas mayores durante cinco o más años, el número de `Hits` realizado en el año anterior afecta el salario, y los jugadores que realizaron más `Hits` el año pasado tienden a tener salarios más altos.
- El árbol de regresión es probablemente una simplificación excesiva de la verdadera relación entre Éxitos, años y salario. Sin embargo, posee ventajas sobre otros tipos de modelos de regresión: Es fácil interpretar, y tiene una representación gráfica facil de leer.

En General

1. Se particiona el espacio de predictores, es decir el conjunto de todos los posibles valores de

$$(X_1, X_2, \ldots, X_p)$$

en J regiones disjuntas $R_1, R_2, \ldots, R_J$.

En General

1. Se particiona el espacio de predictores, es decir el conjunto de todos los posibles valores de

$$(X_1, X_2, \ldots, X_p)$$

en J regiones disjuntas $R_1, R_2, \ldots, R_J$.

2. La predicción de cada observación en la región R_i es simplemente la media de la **variable respuesta** de las observaciones en esta región.

Predicción a través de la estratificación del espacio de características

(La construcción de un árbol)

¿Como se construyen las regiones $R_1, R_2, \ldots, R_J$?

- En teoría las regiones pueden tener cualquier forma, pero por simplicidad e interpretabilidad conviene considerar cajas o rectángulos p-dimensionales.

Predicción a través de la estratificación del espacio de características

(La construcción de un árbol)

¿Como se construyen las regiones $R_1, R_2, \ldots, R_J$?

- En teoría las regiones pueden tener cualquier forma, pero por simplicidad e interpretabilidad conviene considerar cajas o rectángulos p-dimensionales.
- El **objetivo** es encontrar cajas $R_1, R_2, \ldots, R_J$ que minimicen RSS (*suma de cuadrados de los residuales*)

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_i})^2 \tag{1}$$

Algunas consideraciones sobre las reglas de segmentación

- Computacionalmente es inviable considerar cada partición del espacio de predictores en J cajas.

Algunas consideraciones sobre las reglas de segmentación

- Computacionalmente es inviable considerar cada partición del espacio de predictores en J cajas.
- Por esta razón se procede por medio de un procedimiento "*codicioso*" (greedy) de arriba a abajo conocido como **partición binaria recursiva**.

Algunas consideraciones sobre las reglas de segmentación

- Computacionalmente es inviable considerar cada partición del espacio de predictores en J cajas.
- Por esta razón se procede por medio de un procedimiento "*codicioso*" (greedy) de arriba a abajo conocido como **partición binaria recursiva**.
- Es "Codicioso" porque en cada paso del proceso de construcción del árbol, el mejor **particionamiento**(*split*) es realizado en dicho paso, en vez de mirar hacia adelante y seleccionar otro particionamiento que conduzca a un mejor árbol (uno con menor SSR en (1)) en otro paso futuro.

- Para llevar a cabo *particionamiento binario recursivo* se selecciona cada posible predictor X_j y cada posible valor de corte s que reduzcan al máximo el RSS por medio de la partición del espacio predictor generada por los 2 semiplanos

$$R_1(j,s) = \{X | X_j < s\} \quad , \quad R_2(j,s) = \{X | X \geq s\}$$

Es decir, se buscan $j \in \{1, 2, \ldots, J\}$ y s que minimicen la ecuación

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \qquad (2)$$

- Luego se itera este procedimiento en cada una de las 2 regiones que se hayan determinado con anterioridad.
- El proceso continua hasta que se satisfaga un criterio de parada como un número máximo de observaciones en cada una de las regiones finales que corresponden a las hojas o nodos terminales del árbol.

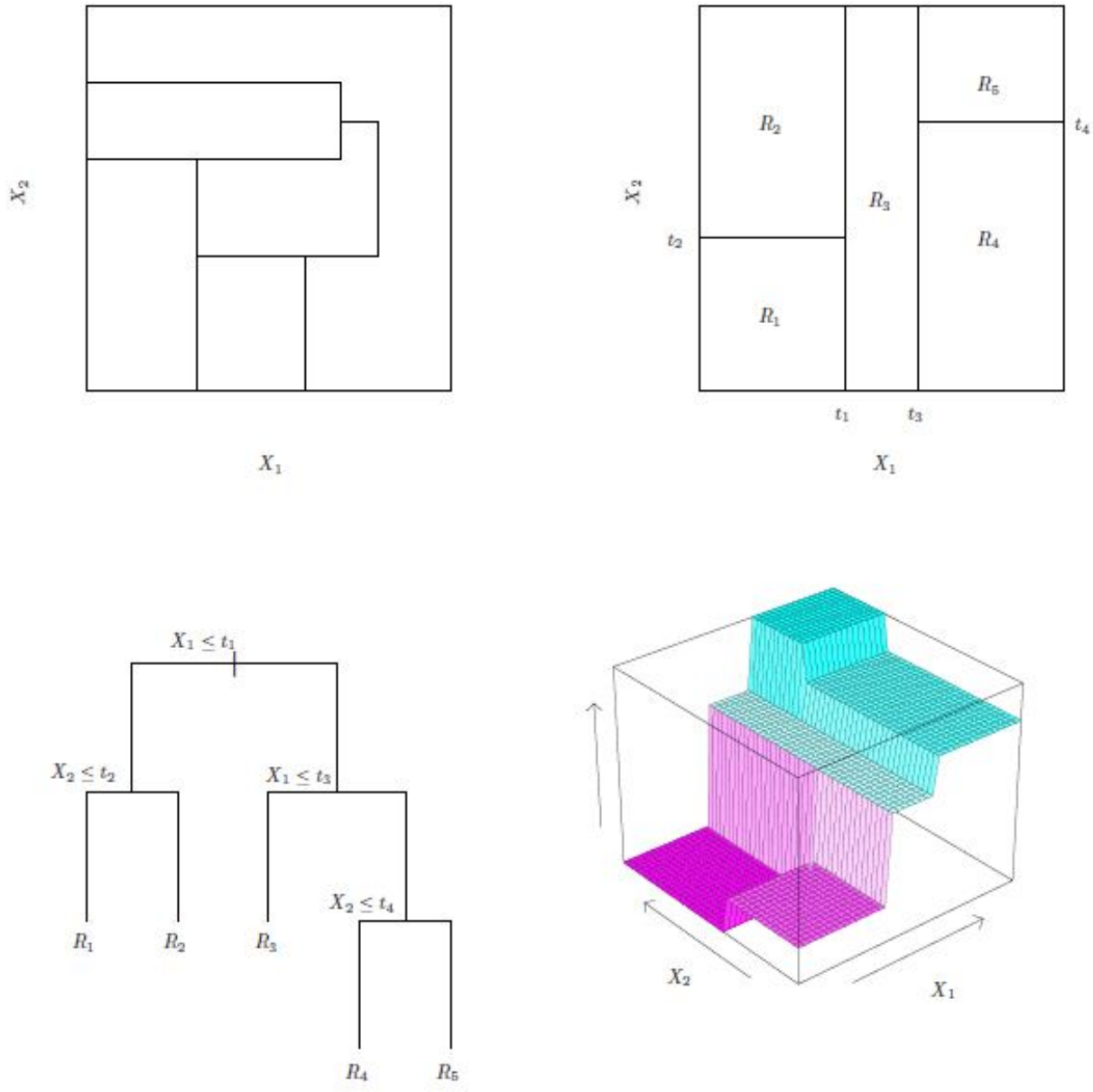
X_2
X_1
R_1
R_2
R_3
R_4
R_5
t_1
t_2
t_3
t_4
$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5
X_2
X_1

Figura 3

Poda de árboles (Tree pruning)

- El proceso descrito para la creación de árboles es propenso a sobre ajustar los datos debido a que los árboles así generados pueden resultar bastante complejos.
- Un árbol más pequeño, (es decir uno con menos regiones $R_1, \ldots, R_j$) puede resultar en un árbol con menos varianza y mejor interpretación al costo de un pequeño incremento en el sesgo.

- Una estrategia para controlar de una forma predecible y consistente la complejidad (tamaño) de los árboles generados consiste en el PRUNING o poda de árboles.
- ¿Cómo determinar la mejor manera de podar un arbol?.
  Cost complexity pruning ("*poda por costo de complejidad*") también conocido como **weakest link pruning** ("*poda del enlace más debil*") proporciona una forma de hacer esto.

- En vez de considerar cada sub-arbol, se considera una secuencia de árboles indexada por un parámetro no negativo α.
  A cada valor de α le corresponde un sub-arbol $T \subset T_0$ tal que la cantidad

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|. \tag{3}$$

  sea tan pequeña como sea posible.

- En (3), cuando $\alpha = 0$, el sub-arbol correspondiente es simplemente igual a T_0, por que (3) es simplemente el error de entrenamiento. Pero a medida que el valor de α aumenta hay un costo a pagar por subárboles con muchas hojas o nodos terminales.

- En (3), cuando $\alpha = 0$, el sub-arbol correspondiente es simplemente igual a T_0, por que (3) es simplemente el error de entrenamiento. Pero a medida que el valor de α aumenta hay un costo a pagar por subárboles con muchas hojas o nodos terminales.
- Así cuando α comienza a incrementarse a partir de 0. Las ramas del arbol comienzan a ser podadas en una forma predecible y jerárquica, por lo que obtener la secuencia de subárboles como una función de α es fácil.

- En (3), cuando $\alpha = 0$, el sub-arbol correspondiente es simplemente igual a T_0, por que (3) es simplemente el error de entrenamiento. Pero a medida que el valor de α aumenta hay un costo a pagar por subárboles con muchas hojas o nodos terminales.
- Así cuando α comienza a incrementarse a partir de 0. Las ramas del arbol comienzan a ser podadas en una forma predecible y jerárquica, por lo que obtener la secuencia de subárboles como una función de α es fácil.
- El valor adecuado de α puede ser seleccionado utilizando validación cruzada.

Algoritmo 8.1 Construcción de un árbol de regresión.

1. Utilícese particionamiento recursivo binario para generar un árbol grande para los datos de entrenamiento, parando solo cuando cada nodo terminal posea menos de un número mínimo de observaciones.
2. Aplíquese *cost complexity pruning* al árbol grande para obtener una secuencia de sub arboles como una función de α.

3. Utilícese K-fold validación cruzada para seleccionar el valor de α. Es decir divida el conjunto de entrenamiento en K lotes y para cada $k = 1, \ldots, K$:
   (a) repítase los pasos 1 y 2 en todos los lotes a excepción del k-ésimo lote de los datos de entrenamiento.
   (b) Evalúe el **error de predicción de media cuadrática** en el k-ésimo lote que se ha dejado aislado, como una función de α .
4. Retórnese el sub-árbol de el paso 2 correspondiente al valor seleccionado de α.

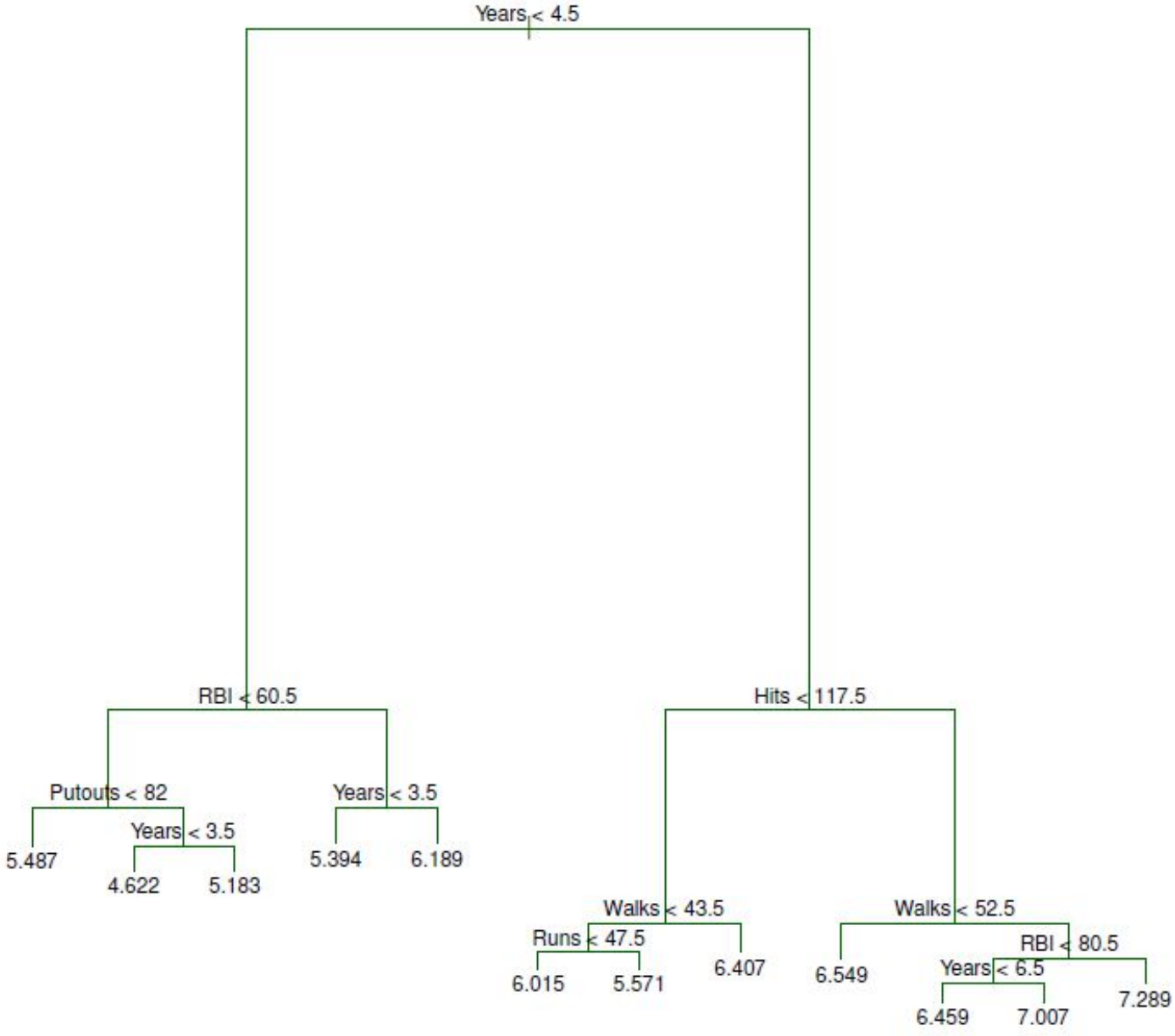
Years < 4.5
RBI < 60.5
Hits < 117.5
Putouts < 82
Years < 3.5
Years < 3.5
Walks < 43.5
Walks < 52.5
Runs < 47.5
RBI < 80.5
Years < 6.5
5.487
4.622
5.183
5.394
6.189
6.015
5.571
6.407
6.549
6.459
7.007
7.289

Figura 4

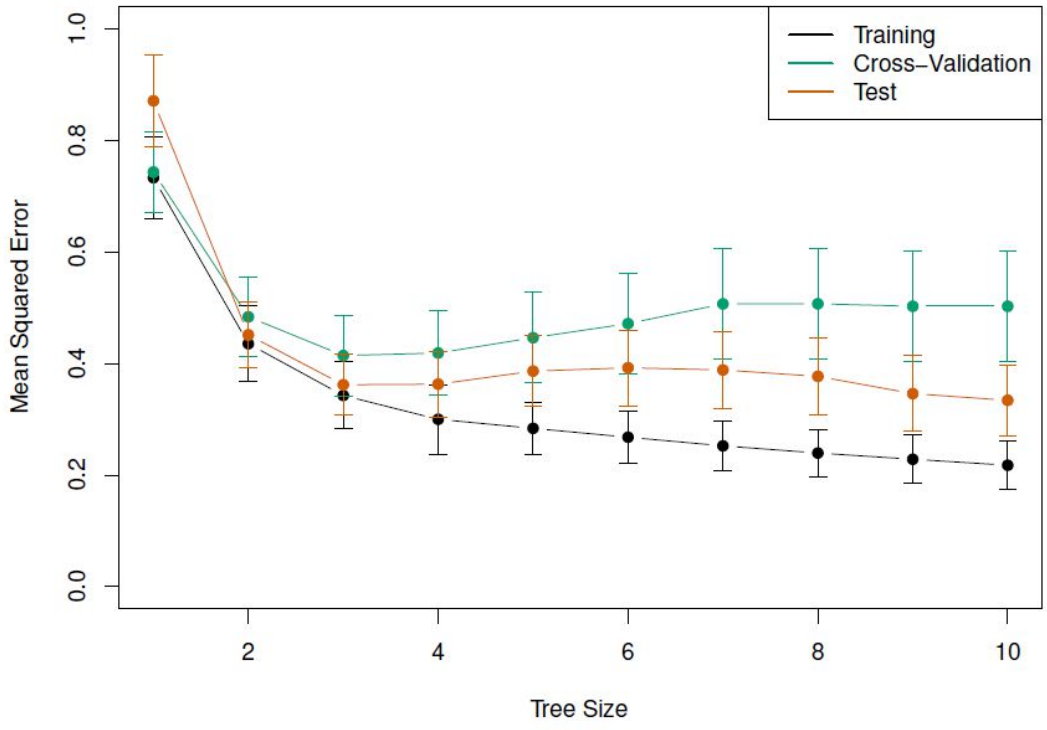

Figura 5: MSE, sobre los conjuntos de, entrenamiento, validación cruzada y de prueba, mostrados como función del número de nodos terminales del correspondiente arbol podado. El mínimo del error en el conjunto v-c ocurre en un tamaño de 3.