Ciencia de Datos

Métodos basados en árboles

Roberto Ponce López

Tecnológico de Monterrey

rpl@tec.mx

10 de marzo del 2021

Agenda

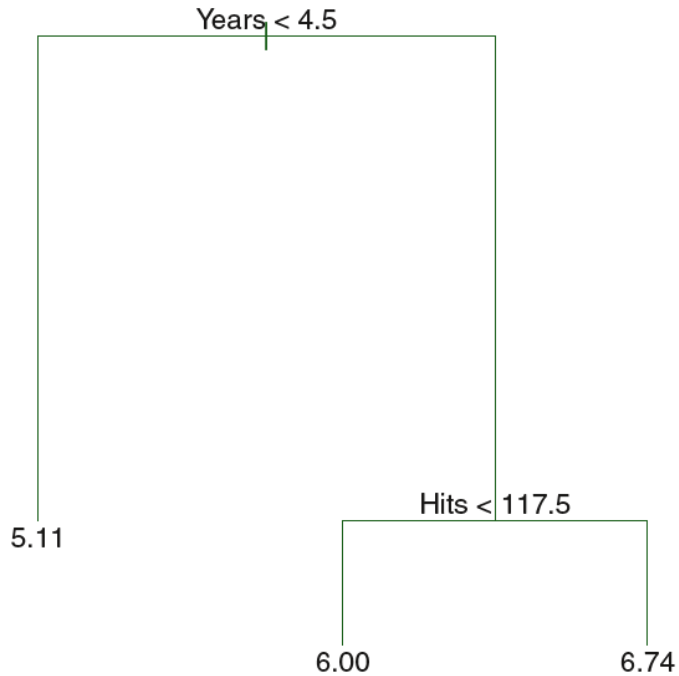
1. Árboles de Regresión
2. Árboles de Clasificación
3. Bagging y Random Forrest

Árboles de Decisión

- Pueden aplicarse a problemas de regresión o de clasificación
- La idea general es que segmentaremos el espacio de predicción en una serie de regiones simples
- Para hacer una predicción para una observación dada, normalmente se utiliza la media de los datos de entrenamiento en la región a la que pertenece
- Dado que el conjunto de reglas de división utilizadas para segmentar el espacio del predictor se puede resumir en un árbol, estos enfoques se denominan métodos de árbol de decisión
- Estos métodos son simples y útiles para la interpretación

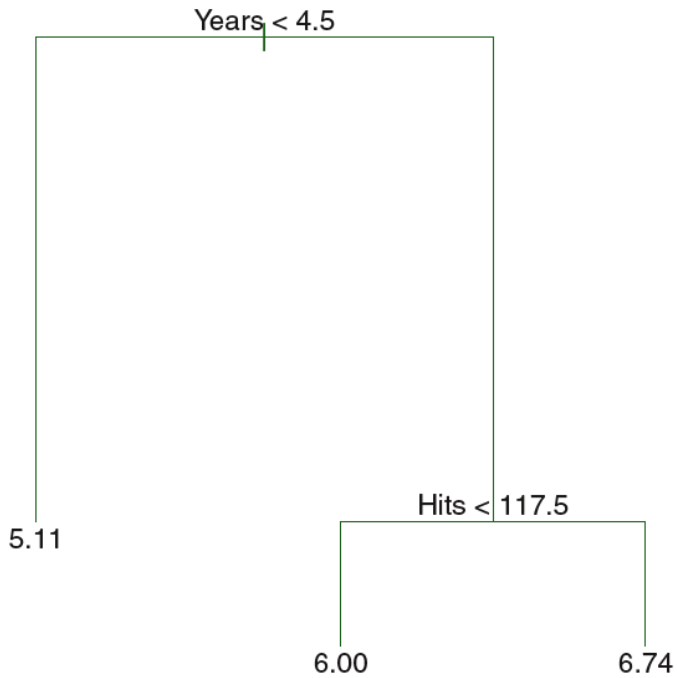
Árboles de Regresión

1. Ejemplo del jugador de béisbol
2. Buscamos predecir su salario con base en su desempeño y años de experiencia en las grandes ligas

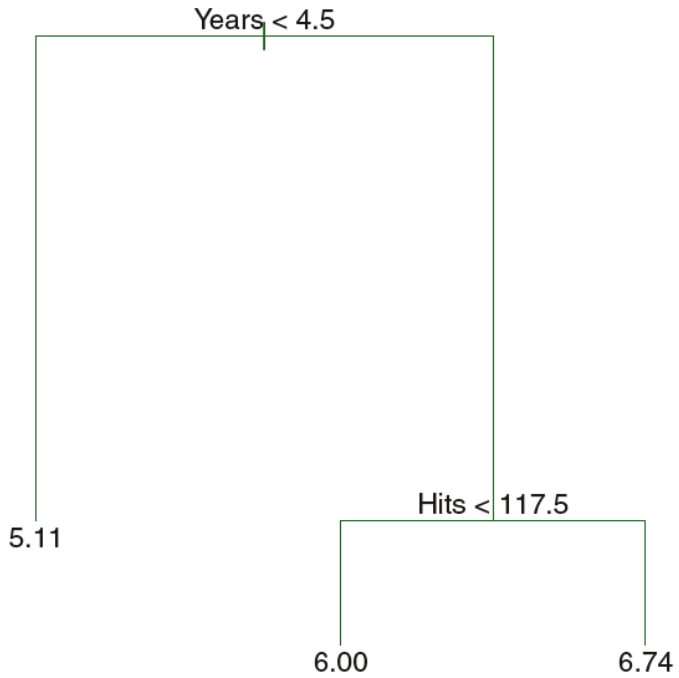


Árboles de Regresión

- Es una serie de reglas de partición con recursión, comenzando por la parte alta
- Dos ramas: $Years < 4{,}5$ $(X_j \leq t_k)$ y $Years > 4{,}5$ $(X_j \geq t_k)$
- El árbol tiene dos nodos y tres hojas

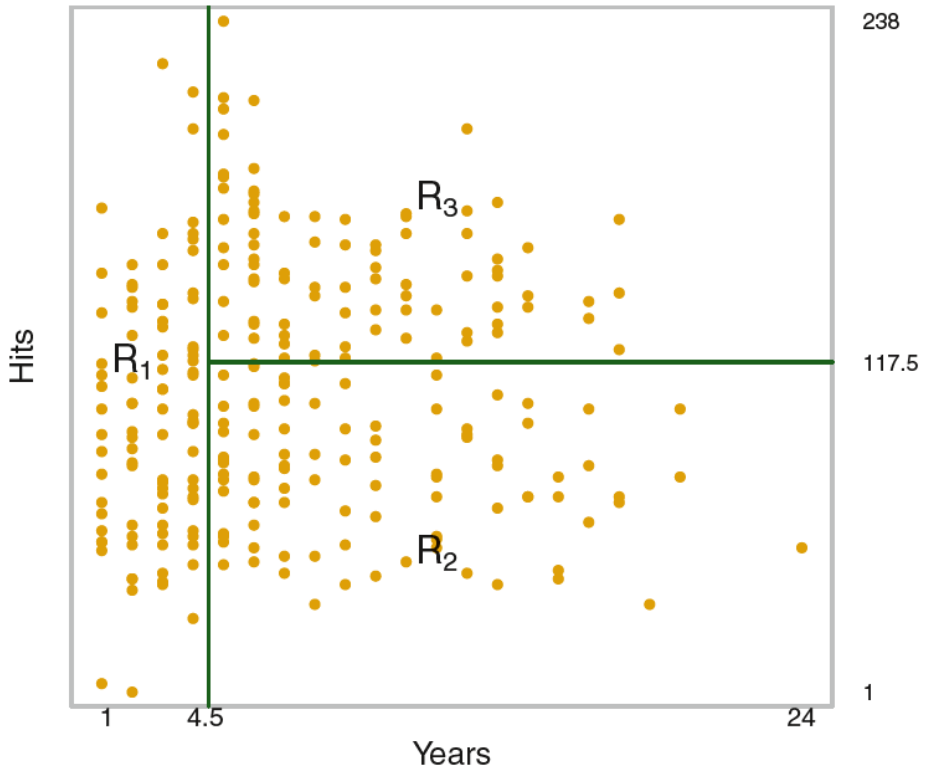


Predicción de Salarios



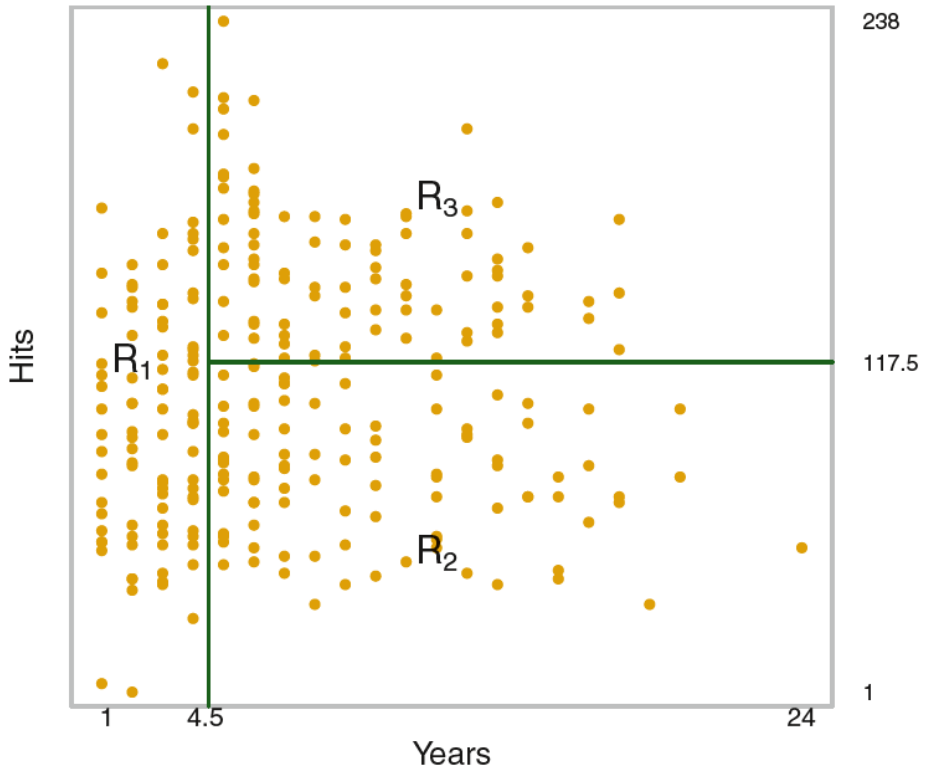
- $e^{5,107} = \$165,174$
- $e^{5,999} = \$402,834$
- $e^{6,740} = \$845,346$

Tres regiones



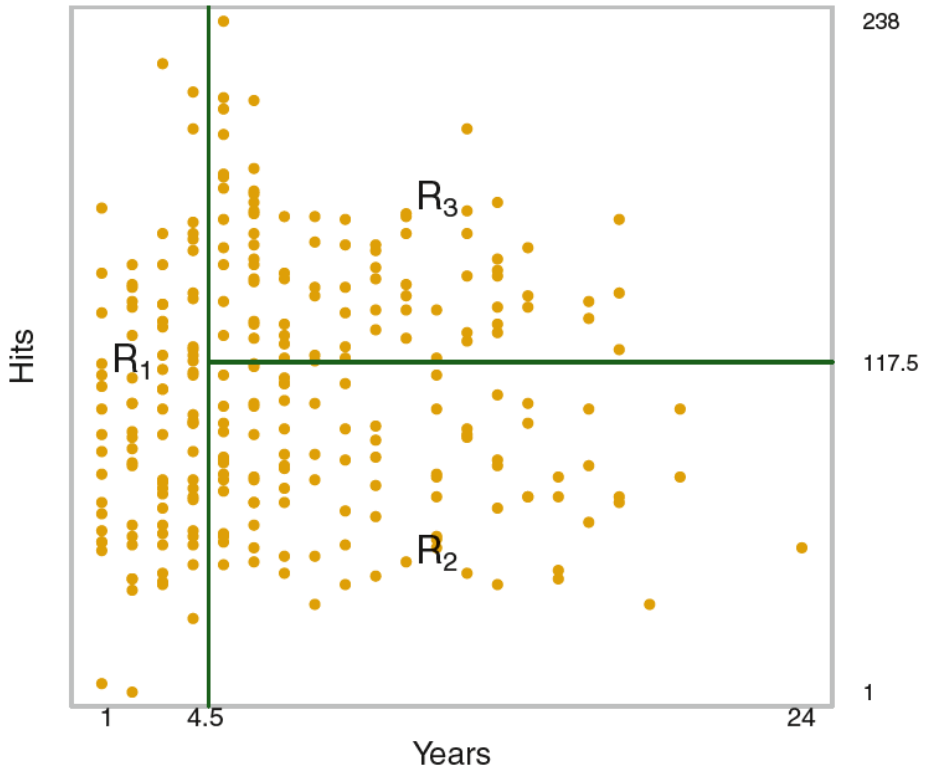
- $R_1 = \{X \,|\, Years < 4{,}5\}$
- $R_2 = \{X \,|\, Years \geq 4{,}5, Hits < 117{,}5\}$
- $R_3 = \{X \,|\, Years \geq 4{,}5, Hits \geq 117{,}5\}$

Tres regiones



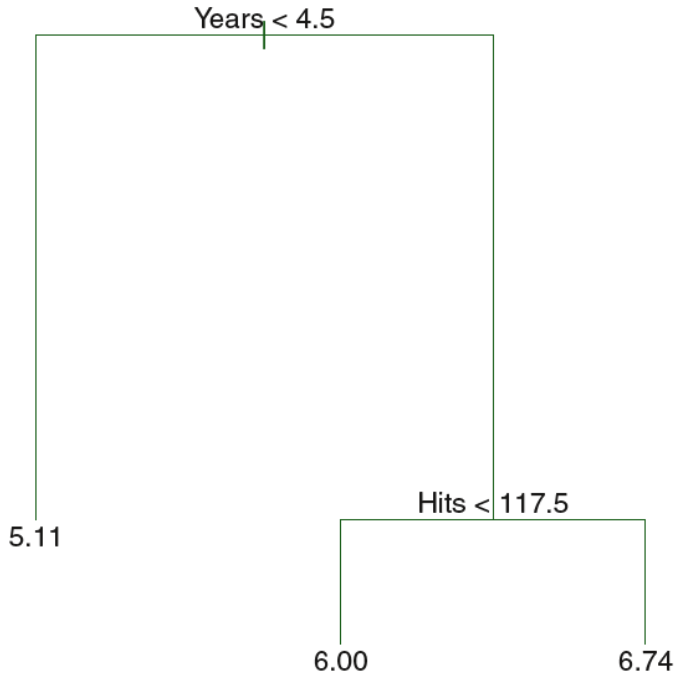
- $R_1 = \{X \,|\, Years < 4{,}5\}$
- $R_2 = \{X \,|\, Years \geq 4{,}5, Hits < 117{,}5\}$
- $R_3 = \{X \,|\, Years \geq 4{,}5, Hits \geq 117{,}5\}$
- *Years* es el factor más importante para determinar salario. Los jugadores con menor experiencia ganan salarios más bajos que los

Tres regiones



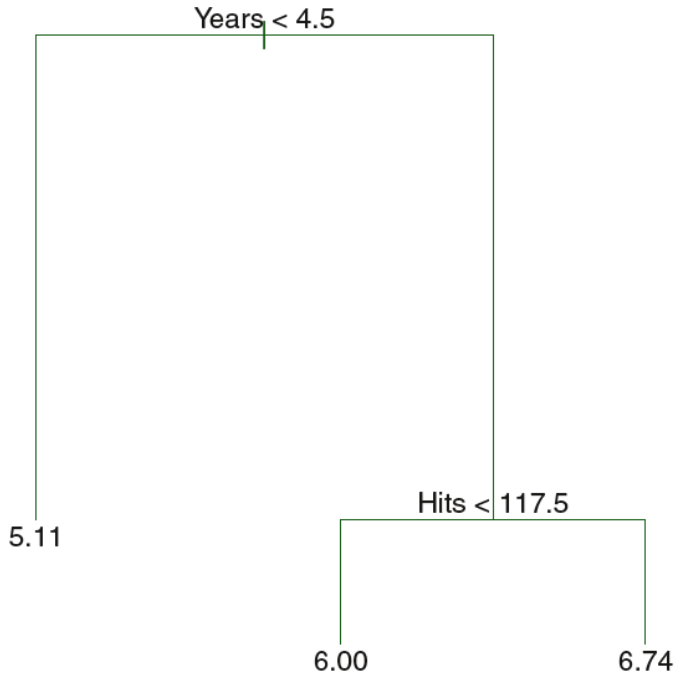
- $R_1 = \{X \,|\, Years < 4{,}5\}$
- $R_2 = \{X \,|\, Years \geq 4{,}5, Hits < 117{,}5\}$
- $R_3 = \{X \,|\, Years \geq 4{,}5, Hits \geq 117{,}5\}$
- *Years* es el factor más importante para determinar salario. Los jugadores con menor experiencia ganan salarios más bajos que los

Tres regiones



- Dado que un jugador es menos experimentado, el número de hits que hizo en su año previo parece jugar un papel menor en su salario

Tres regiones



- Dado que un jugador es menos experimentado, el número de hits que hizo en su año previo parece jugar un papel menor en su salario
- Pero el número de hits en el año previo es un predictor importante de salario para aquéllos que han estado más de 5 años en las grandes ligas. A mayor número de hits, mayor salario.

Construcción de un Árbol de Regresión

Hay dos étapas:

1. Dividimos el espacio de predicción, esto es el set de posibles valores para $X_1, X_2, ...X_p$, en J regiones distintas regiones (no superpuestas), $R_1, R_2, ..., R_J$
2. Para cada observación que cae dentro de la región R_j, hacemos la misma predicción, que es el simple promedio en los valores de respuesta para las observaciones de entrenamiento en R_j.

¿Cómo construimos las regiones $R_1, R_2, ..., R_J$?

Hay dos étapas:

1. Dividimos el espacio de predicción en rectángulos de alta dimensionalidad
2. El objetivo es minimizar el Residual Sums of Squares (RSS):

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

donde $\hat{y}_{R_j}$ es la respuesta promedio para las observaciones de entrenamiento dentro del rectángulo jth

¿Cómo construimos las regiones $R_1, R_2, ..., R_J$?

- Es imposible tomar cada posible partición del espacio en J rectángulos
- Por esta razón, tomamos una aproximación *greedy* que se conoce como *recursive binary splitting*

¿Cómo construimos las regiones $R_1, R_2, ..., R_J$?

- Es imposible tomar cada posible partición del espacio en J rectángulos
- Por esta razón, tomamos una aproximación *greedy* que se conoce como *recursive binary splitting*
- El algoritmo es *greedy* porque en cada etapa del árbol de construcción del proceso, la mejor partición es la que ocurre en esa etapa, sin considerar los resultados de las particiones subsecuentes por ocurrir

¿Cómo construimos las regiones $R_1, R_2, ..., R_J$?

- Es imposible tomar cada posible partición del espacio en J rectángulos
- Por esta razón, tomamos una aproximación *greedy* que se conoce como *recursive binary splitting*
- El algoritmo es *greedy* porque en cada etapa del árbol de construcción del proceso, la mejor partición es la que ocurre en esa etapa, sin considerar los resultados de las particiones subsecuentes por ocurrir
- Seleccionamos el predictor X_j y el valor s del valor de corte de la variable para partir la región, el cual conlleva a una reducción en el RSS

Recursive binary splitting

1. Seleccionamos el predictor X_j y el valor de corte s, de tal forma que partimos el espacio de predicción en las regiones $\{X|X_j < s\}$ y $\{X|X_j \geq s$ que conlleva a la reducción máxima del RSS
2. Para cualquier j y s, definimos el par de semiplanos

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

y

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

3. y buscamos el valor de j y s que minimiza la ecuación:

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

Recursive binary splitting

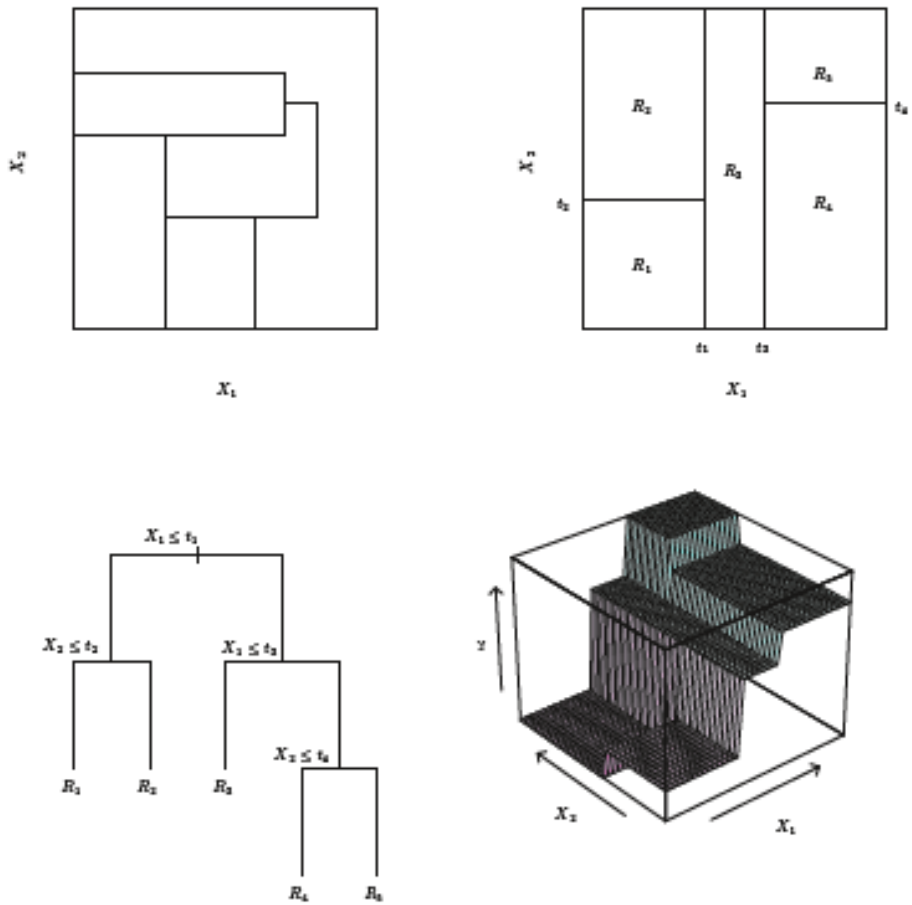


FIGURE 8.3. Top Left: *A partition of two-dimensional feature space that could not result from recursive binary splitting.* Top Right: *The output of recursive binary splitting on a two-dimensional example.* Bottom Left: *A tree corresponding to the partition in the top right panel.* Bottom Right: *A perspective plot of the prediction surface corresponding to that tree.*

Tree Pruning

- Este proceso muy seguramente sobreajusta a nuestros datos (*overfit*)
- Estrategia: construye el árbol más grande posible y despúes poda las ramas que no aportan

Árboles de Clasificación

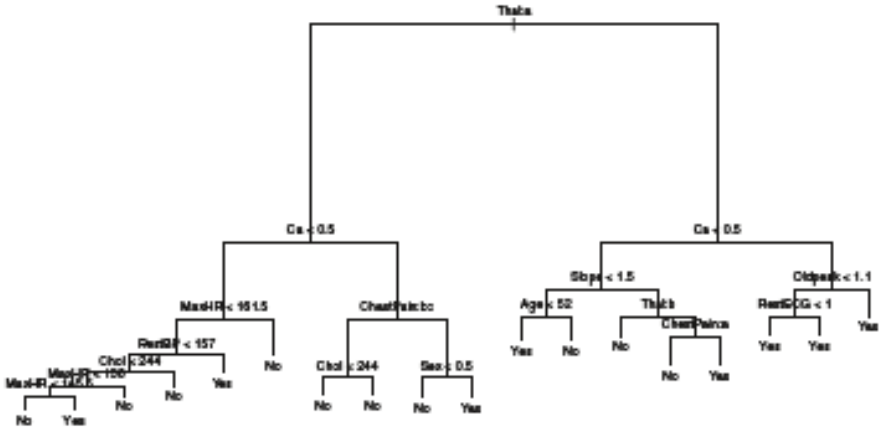
- Similares a los árboles de regresión, pero se utilizan para predecir una respuesta cualitativa o categórica en lugar de cuantitativa
- Cada observación en una región dada es asignada a la clase qu ocurre más frecuentemente

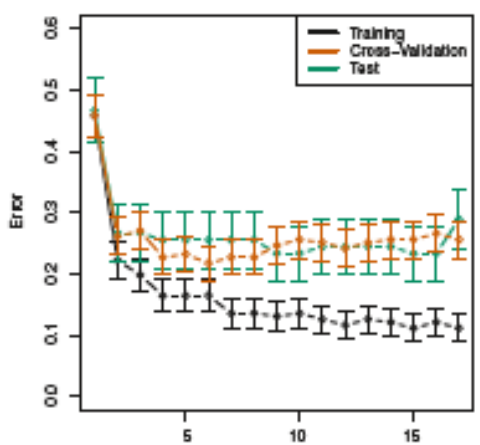
Árboles de Clasificación

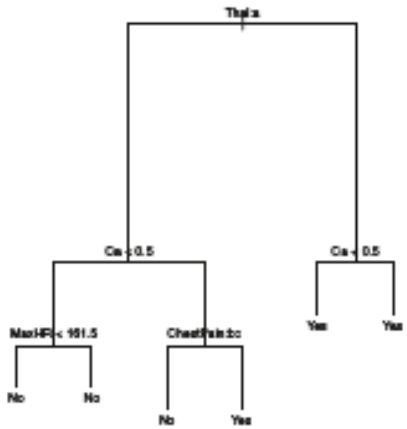
- Similares a los árboles de regresión, pero se utilizan para predecir una respuesta cualitativa o categórica en lugar de cuantitativa
- Cada observación en una región dada es asignada a la clase qu ocurre más frecuentemente
- La función de pérdida no es el RSS porque se trata de una variable categórica
- La función de pérdida es la tasa de clasificación del error
- Esto es la tasa de casos en la base de entrenamiento que son mal clasificados por la asignación en la región dada
- El Gini Index o un índice de Entropía son alternativas para calcular el error (más sensitivos al número de nodos)

Ejemplo: Heart data set

- Respuestas binarias de 303 pacientes con paro cardíaco que presentaron dolor en el pecho
- Hay 13 predictores, incluyendo: edad, sexo y niveles de colesterol







Árboles versus Módelos Lineales

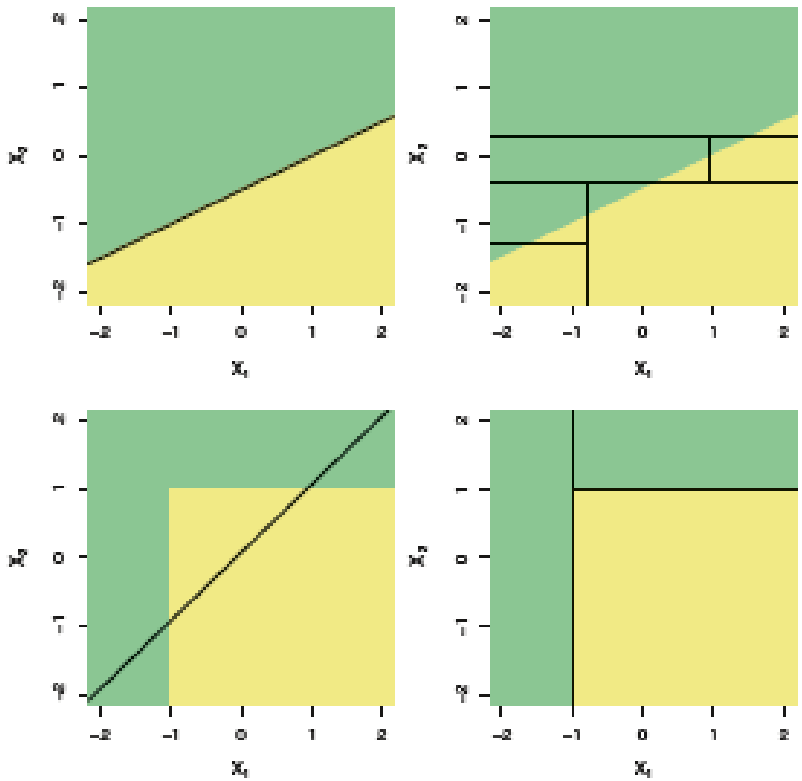


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

Ventajas de los Árboles

- Fáciles de explicar al público; más fácil que una regresión lineal
- Algunos expertos consideran que los árboles aproximan de mejor manera el proceso de decisión humano
- Tienen una representación visual fácil de interpretar, incluso por audiencias no expertas
- Funcionan con variables númericas o categóricas, tanto en las independientes como en la dependiente

Desventajas de los Árboles

- Desafortunadamente, los árboles generalmente no tienen el mismo nivel de exactitud en las predicciones que la regresión lineal o un logit multinomial
- Adicionalmente, los árboles no son muy robustos. Un pequeño cambio en los datos puede cambiar en el árbol estimado

Desventajas de los Árboles

- Desafortunadamente, los árboles generalmente no tienen el mismo nivel de exactitud en las predicciones que la regresión lineal o un logit multinomial
- Adicionalmente, los árboles no son muy robustos. Un pequeño cambio en los datos puede cambiar en el árbol estimado
- Esto nos lleva a los siguientes dos métodos...

Bagging

- Los árboles de decisión sufren de varianzas grandes

Bagging

- Los árboles de decisión sufren de varianzas grandes
- Esto significa que si partimos los datos en dos mitades y ajustamos un árbol de decisión en cada mitad, el resultado que vamos a obtener en el par va a ser muy diferente
- Sin embargo, el resultado tendría poca varianza si aplicado repetidamente en distintas bases de datos
- Tomar el promedio de un set de observaciones reduce la varianza
- Podemos calcular $\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f}^B(x)$ utilizando B training data sets separados, sacando el promedio para obtener un modelo de aprendizaje estadístico con una varianza pequeña

Bagging

- El problema es que solamente tenemos una base de datos y no varias
- La solución es generar B bootstrapeped training data sets
- Entrenando nuestro método en el bth bootstrapped training data set para obtener $\hat{f}^{*b}(x)$, sacando el promedio de las predicciones para obtener:

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

Bagging

- El problema es que solamente tenemos una base de datos y no varias
- La solución es generar B bootstrapeped training data sets
- Entrenando nuestro método en el bth bootstrapped training data set para obtener $\hat{f}^{*b}(x)$, sacando el promedio de las predicciones para obtener:

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

- Ejemplo: toma dos tercios de los datos aleatoriamente y corre un árbol; repite el procedimiento una gran cantidad de veces

Bagging

- El problema es que solamente tenemos una base de datos y no varias
- La solución es generar B bootstrapeped training data sets
- Entrenando nuestro método en el bth bootstrapped training data set para obtener $\hat{f}^{*b}(x)$, sacando el promedio de las predicciones para obtener:

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

- Ejemplo: toma dos tercios de los datos aleatoriamente y corre un árbol; repite el procedimiento una gran cantidad de veces
- Con bagging ganas en predicción, pero pierdes en interpretabilidad

Ejemplo de Bagging

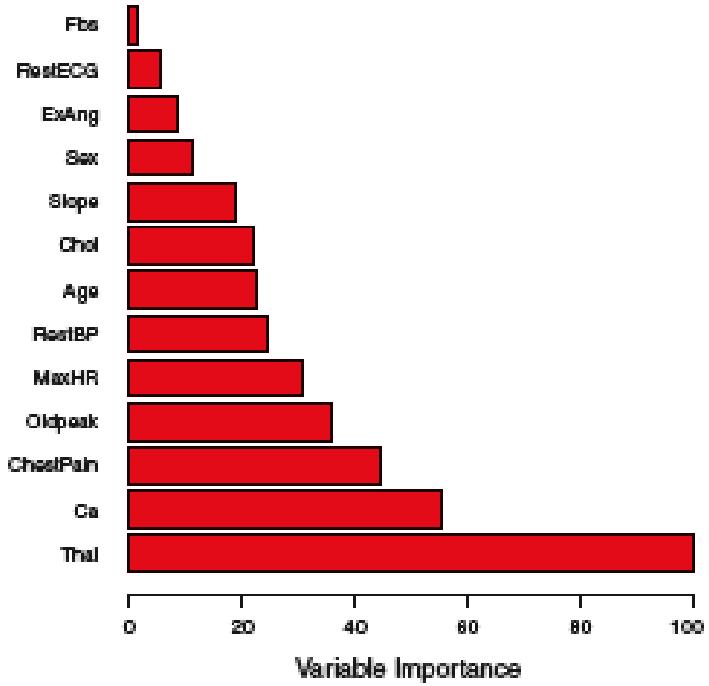


FIGURE 8.9. *A variable importance plot for the* `Heart` *data. Variable importance is computed using the mean decrease in Gini index, and expressed relative to the maximum.*

FIGURE 8.7. *Top Row: A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

Random Forrest

- Similar a Bagging, también se fundamenta en Bootstrapping
- Toma muestras aleatorias no solamente de casos, sino de predictores j
- El algoritmo no toma todos los predictores, sino una selección aleatoria del set de variables disponibles como variable independiente
- El efecto es una decorrelación de los árboles porque tienen distintos predictores

Random Forrest

- Similar a Bagging, también se fundamenta en Bootstrapping
- Toma muestras aleatorias no solamente de casos, sino de predictores j
- El algoritmo no toma todos los predictores, sino una selección aleatoria del set de variables disponibles como variable independiente
- El efecto es una decorrelación de los árboles porque tienen distintos predictores
- Previene overfitting

Random Forrest

- Similar a Bagging, también se fundamenta en Bootstrapping
- Toma muestras aleatorias no solamente de casos, sino de predictores j
- El algoritmo no toma todos los predictores, sino una selección aleatoria del set de variables disponibles como variable independiente
- El efecto es una decorrelación de los árboles porque tienen distintos predictores
- Previene overfitting
- Hay una reducción en la varianza porque muchas de las cantidades no están correlacionadas
- El promedio es el predictor