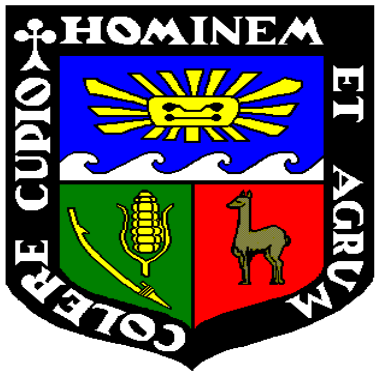
Técnicas Multivariadas Avanzadas

Métodos basados en árboles

Ms Carlos López de Castilla Vásquez

Universidad Nacional Agraria La Molina

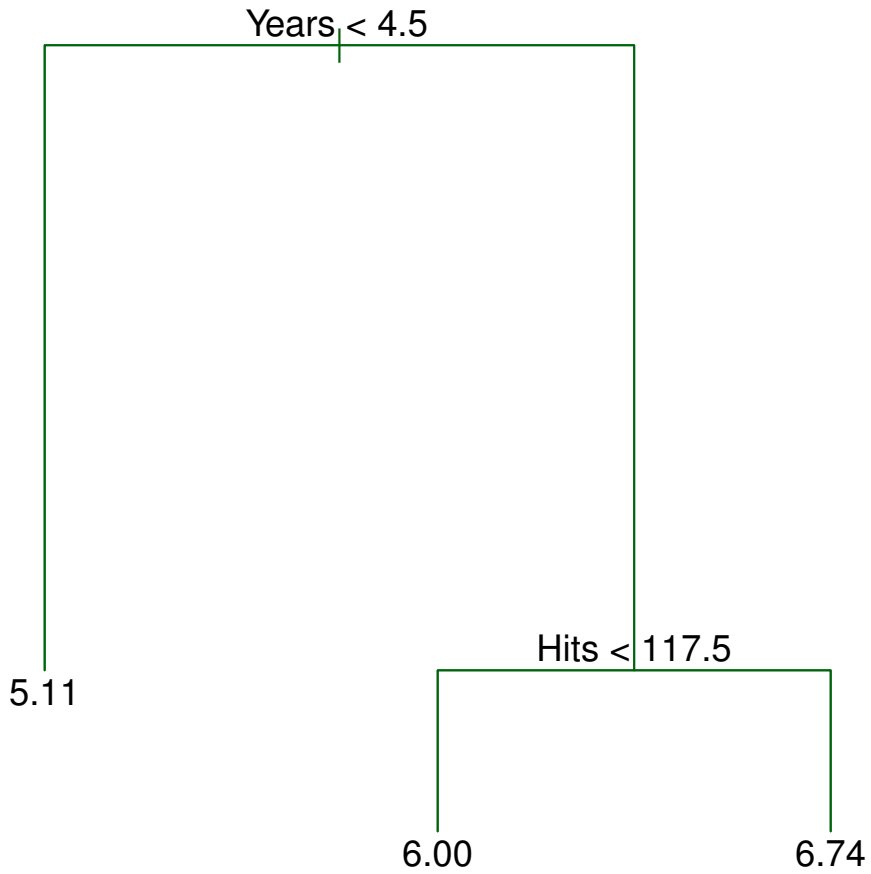
2014-2



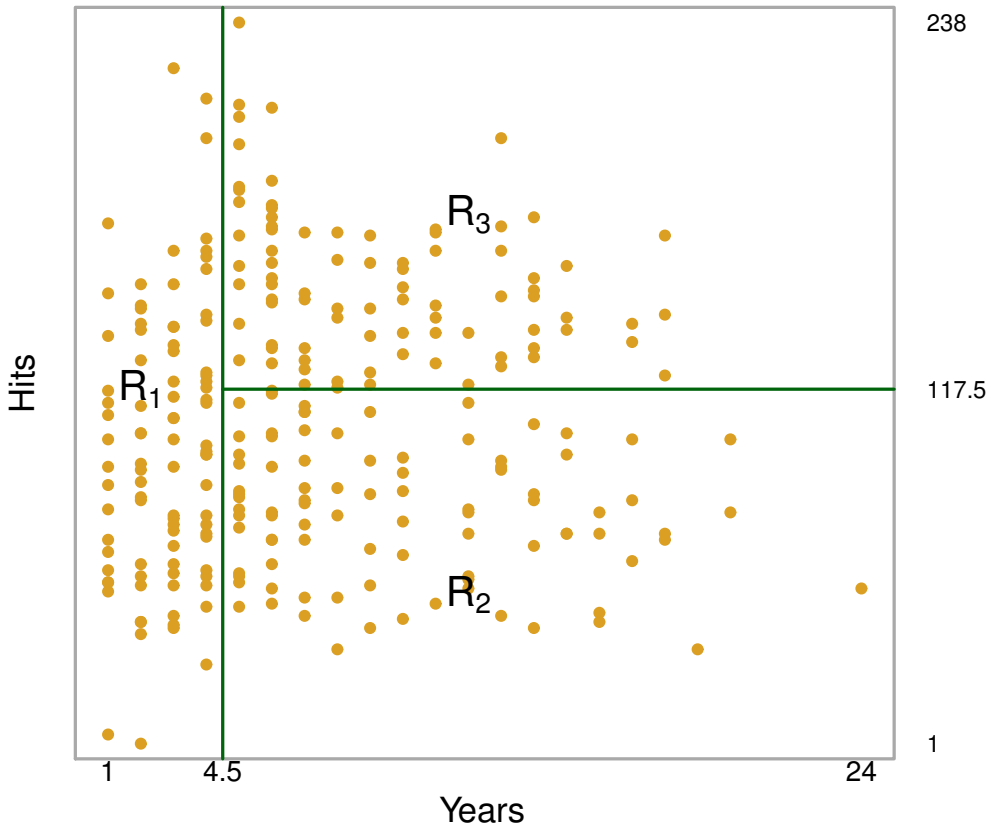
Introducción

- Se describen métodos *basados en árboles* para regresión y clasificación.
- Estos métodos requieren *estratificar* o *segmentar* el espacio de los predictores en un determinado número de regiones.
- Como el conjunto de reglas de separación usadas pueden ser resumidas en un árbol, esta metodología es conocida como *métodos de decisión basados en árboles*.
- Los métodos basados en arboles son simples y útiles para propósitos de interpretación.
- Sin embargo no son competitivos con los mejores métodos de aprendizaje supervisado.

Arbol de regresión para la data Baseball



Arbol de Regresión para la data Baseball



Terminología

- Las regiones R_1, R_2 y R_3 son conocidas como *nodos terminales*.
- Los árboles de decisión tienen un crecimiento inverso al que le conocemos.
- Los puntos en los que el árbol divide el espacio de los predictores son llamados *nodos internos*.
- En el árbol de regresión para la data `Baseball` los nodos internos se indican usando:

$$\texttt{Years} < 4{,}5 \qquad \texttt{Hits} < 117{,}5.$$

Interpretación

- `Years` es el factor más importante para determinar `Salary`. Los jugadores con menos experiencia ganan menos que los más experimentados.
- Si un jugador tienen menos experiencia entonces el número de `Hits` juega un pequeño rol en el `Salary`.
- Pero entre jugadores que estuvieron en las grandes ligas por cuatro años y medio o más, el número de `Hits` si afecta el `Salary` ya que los jugadores con más `Hits` ganan más.
- Se trata de una sobresimplificación pero en comparación con la regresión es una herramienta fácil de observar, interpretar y explicar.

Construcción del árbol

- Se divide el espacio de los predictores en J regiones $R_1, R_2, \cdots, R_J$ que no se traslapan.
- Para cada observación que se encuentra en la región R_j se tiene la misma predicción obtenida como la media de los valores de la variable respuesta para las observaciones en la muestra de entrenamiento dentro de R_j.
- En teoría las regiones podrían tener cualquier forma sin embargo es preferible dividir el espacio de los predictores en *rectángulos multidimencionales* o *cajas* por simplicidad y facilidad en la interpretación.

Construcción del árbol

- El objetivo es encontrar $R_1, R_2, \cdots, R_J$ que minimicen:

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

donde $\hat{y}_{R_j}$ es la media de la variable respuesta para las observaciones en la data de entrenamiento dentro de la j-ésima caja.

- Desafortunadamente no es computacionalmente viable considerar cada posible partición del espacio de los predictores en J cajas.

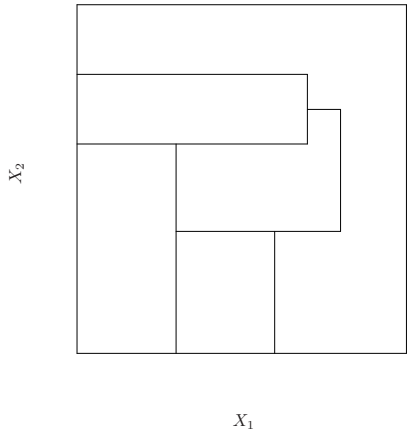
Construcción del árbol

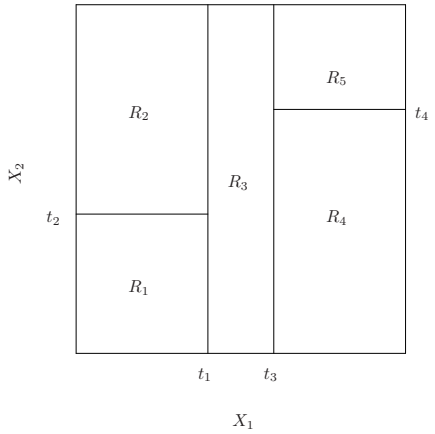
- Por esta razón se utiliza un método *greedy top-down* llamado de *división recursiva binaria*.
- El método es *top-down* por que empieza en la parte superior del árbol y luego se divide sucesivamente el espacio de los predictores. En cada división hay dos ramas que se abren hacia abajo.
- El método es *greedy* por que en el proceso de construcción del árbol, la mejor división se realiza en cada paso sin considerar en la elección aquella que permita obtener un mejor árbol en un paso sucesivo.

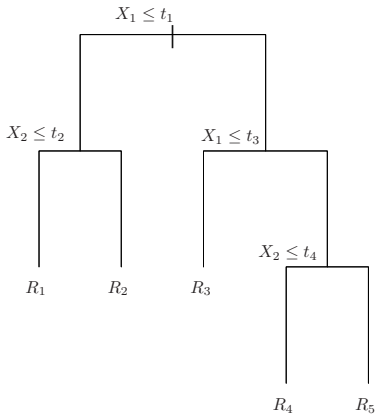
Construcción del árbol

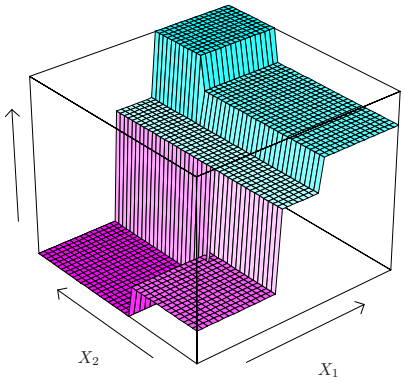
- Se elige el predictor X_j y el punto de corte s tal que la división en $\{X|X_j < s\}$ y $\{X|X_j \geq s\}$ permite obtener la mayor reducción en RSS.
- Se repite el proceso buscando el mejor predictor y el mejor punto de corte que permita dividir la data y minimizar RSS dentro de cada una de las regiones obtenidas previamente.
- Nuevamente se busca dividir las regiones de tal forma que se minimice RSS. El proceso continua hasta que se cumpla cierto criterio de parada o cuando cada región tenga como máximo cinco observaciones.

Construcción del árbol









Poda de un árbol

- Una estrategia adecuada es construir un árbol completo T_0 y luego *podarlo* para obtener un sub-árbol.
- Se usa un *costo de complejidad por poda* también llamado *weakest link pruning*.
- Se considera una secuencia de árboles indexados por un parámetro de sintonización $\alpha > 0$. Para cada valor de α le corresponde un subconjunto $T \subset T_0$ tal que:

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

sea lo menor posible.

Poda de un árbol

- El parámetro de sintonización α controla el intercambio entre la complejidad del sub-árbol y la bondad de ajuste obtenida con la data de entrenamiento.
- Se puede elegir el valor óptimo de $\hat{\alpha}$ usando validación cruzada.
- Luego de la elección se estima el sub-árbol correspondiente a $\hat{\alpha}$ usando la data completa.

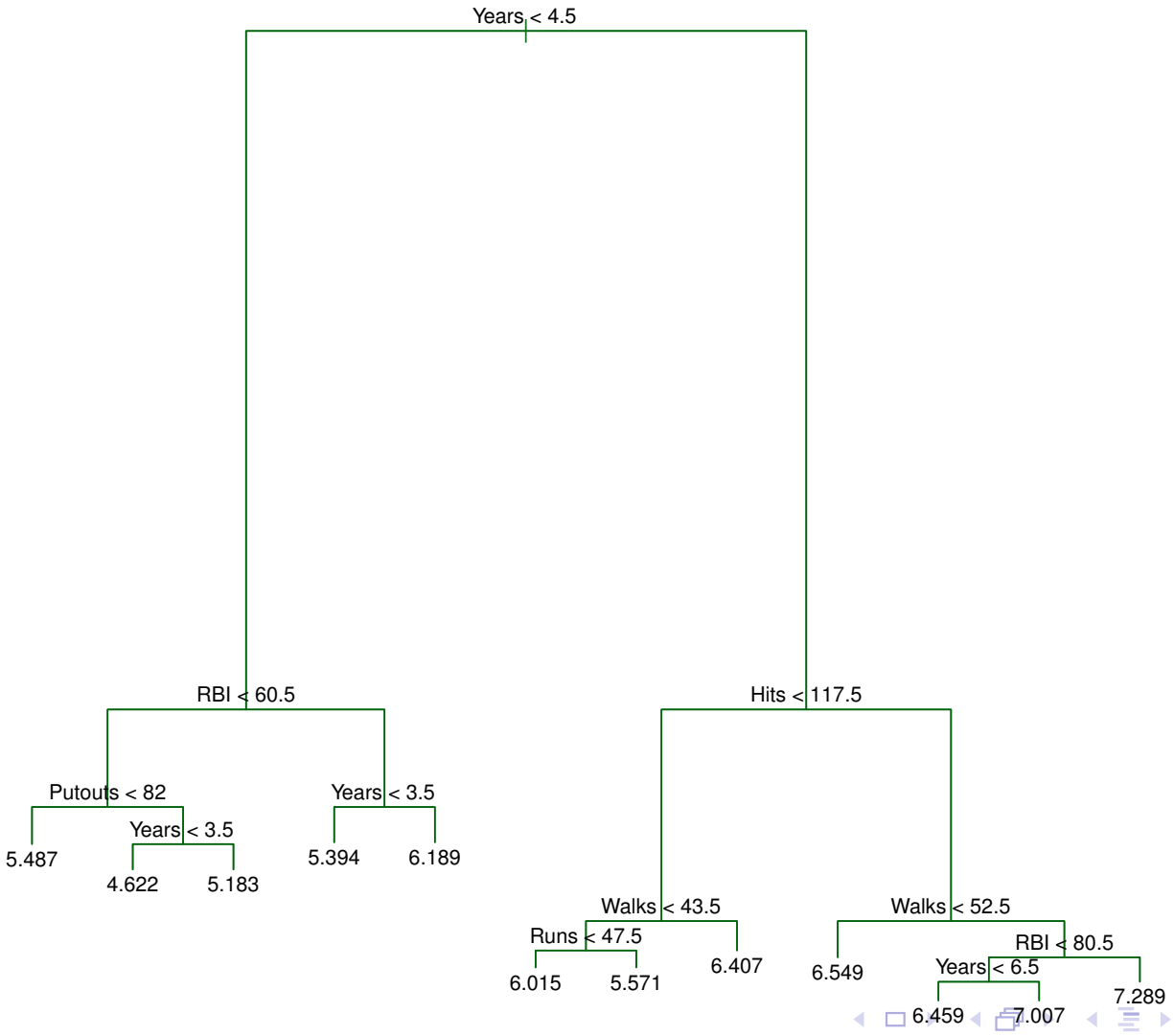
Resumen del algoritmo

1. Se usa *división recursiva binaria* para obtener un árbol completo con la data de entrenamiento.
2. Se aplica el *costo de complejidad por poda* para obtener una secuencia de sub-árboles como función de α.
3. Usar CV K-fold para elegir α. Para $k = 1, \cdots, K$:
   - 3.1 Repetir los pasos 1 y 2 sobre la fracción $\frac{K-1}{K}$ de la data de entrenamiento.
   - 3.2 Evaluar MSPE en el K-ésimo fold dejado fuera como función de α.
4. Elegir α que minimice el error promedio.
5. Se estima el sub-árbol correspondiente a $\hat{\alpha}$ usando la data completa.

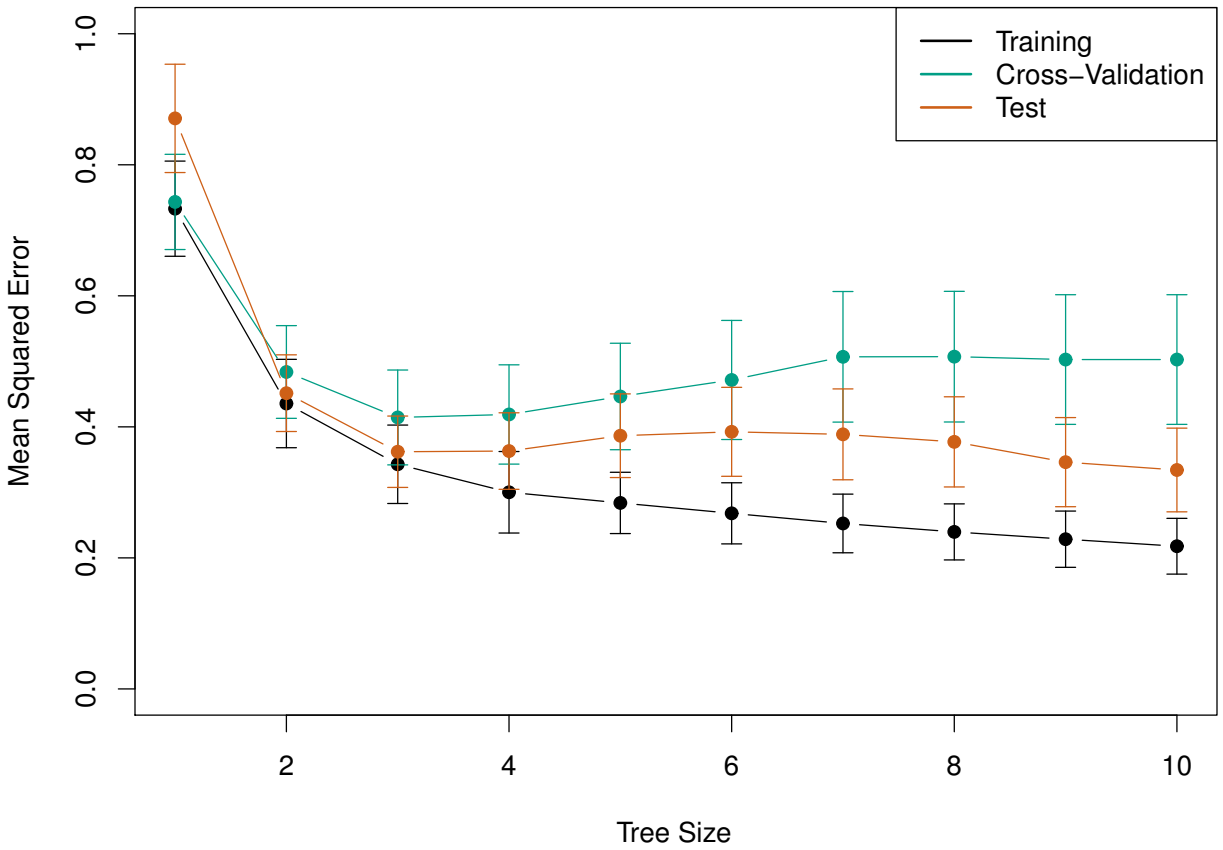
Ejemplo Baseball

- Primero se divide aleatoriamente la data en 132 observaciones en la data de entrenamiento y 131 observaciones en la data de prueba.
- Se construye un árbol de regresión completo con la data de entrenamiento considerando diferentes valores para α que permitan obtener sub-árboles con diferentes números de nodos terminales.
- Finalmente se realiza CV $K = 6$ para estimar MSE por validación cruzada para los arboles como función de α.

Ejemplo Baseball



Ejemplo Baseball



Arboles de clasificación

- Un árbol de clasificación es muy parecido a un árbol de regresión. La diferencia esta en que en el primer caso el árbol es usado para clasificar una observación en alguna de las clases correspondientes a la variable respuesta.
- Para un árbol de clasificación la predicción se realiza hacia la clase *más común* para las observaciones en la data de entrenamiento correspondientes a la región de la cual proviene.
- Para el proceso de construcción también se usa un método de división recursiva binaria, sin embargo ya no es posible usar RSS como criterio para determinar los puntos de división.

Construcción del árbol

- Una alternativa natural a RSS es la *tasa de error de clasificación* definida como la fracción de las observaciones en la data de entrenamiento que no coinciden con la clase más común:

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

  donde $\hat{p}_{mk}$ representa la proporción de las observaciones en la data de entrenamiento en la $m-$ésima región que pertenecen a la $k-$ésima clase.

- Sin embargo el error de clasificación no es lo suficientemente sensible para el proceso de construcción y en la práctica es preferible usar otro tipo de indicador.

Indice de Gini y Devianza

- El *indice de Gini* se define por:

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

  y mide el total de varianza en las K clases.
- Este indicador toma un valor pequeño si todos los $\hat{p}_{mk}$ toman valores cercanos a cero o uno.
- Por esta razón se considera una medida de la *pureza* del nodo ya que un valor pequeño indica que el nodo contiene observaciones donde predomina una clase.

Indice de Gini y Devianza

- Una alternativa al indice de Gini es la *entropía cruzada* definida por:

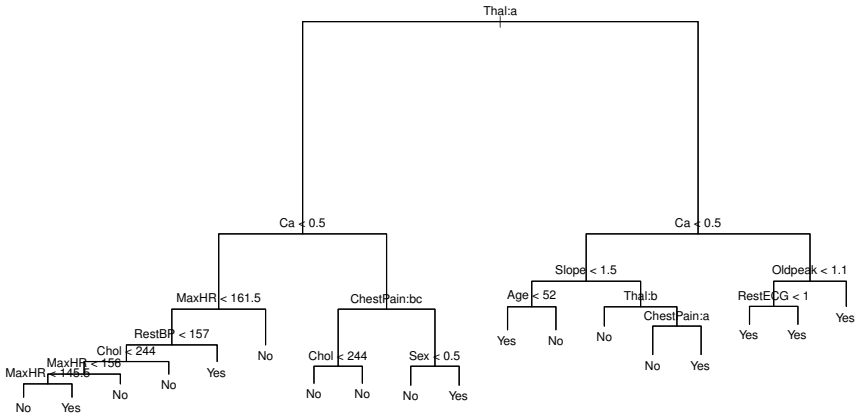
$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

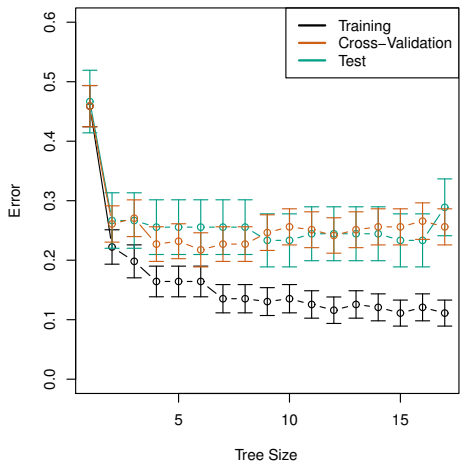
- El indice de Gini y la entropía cruzada son indicadores muy parecidos en términos numéricos.

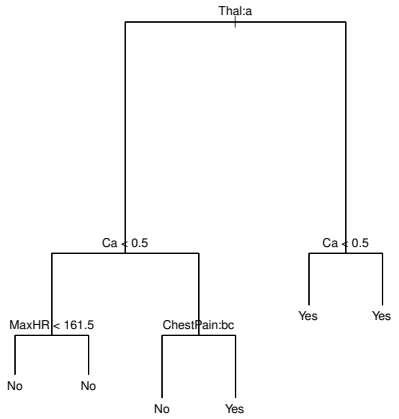
Ejemplo data Heart

- La data contiene una variable respuesta binaria `HD` para 303 pacientes que presentaron dolor de pecho.
- La clase `Yes` indica la presencia de enfermedad del corazón basado en pruebas angiográficas mientras que la clase `No` indica la no presencia de la enfermedad.
- Se tienen 13 predictores como `Age`, `Sex`, `Chol` (una medida del colesterol), etc.
- Se realizó validación cruzada obteniendo un árbol con seis nodos terminales.

Ejemplo data Heart







Bagging

- *Bagging* o *bootstrap aggregation* es un procedimiento usado para reducir la variancia de un método statistical learning.
- Recordar que dado n observaciones $Z_1, Z_2, \cdots, Z_n$ cada una con variancia σ^2 entonces la media $\overline{Z}$ tiene varianza igual a σ^2/n.
- En otras palabras promediando un conjunto de observaciones se reduce la variancia. Podría parecer nada práctico ya que por lo general no se tiene acceso a múltiples conjuntos de entrenamiento.
- Sin embargo es posible aplicar bootstrap, tomando muestras repetidas a partir del conjunto de entrenamiento.

Bagging

- Se generan B diferentes muestras de entrenamiento usando bootstrap.
- Al aplicar el método sobre $b-$ésima muestra de entrenamiento bootstrap se obtiene la predicción $\hat{f}^{*b}(x)$ en el punto x.
- Se promedian las predicciones y se obtiene:

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

conocido como bagging.

Arboles bagging

- El procedimiento anterior puede ser aplicado a los árboles.
- En un árbol de regresión se construyen B árboles usando B muestras bootstrap de entrenamiento y luego promediando las predicciones resultantes.
- En un árbol de clasificación para cada observación en la data de prueba se registra la clase predecida por cada uno de los B árboles y se considera un *voto mayoritario*, es decir la predicción se hace hacia la clase más frecuente obtenida en las B predicciones.

Estimación del error out-of-bag

- La idea del bagging es estimar repetidamente los árboles usando las muestras boostrap. Se puede demostrar que cada árbol usa aproximadamente dos tercios de las observaciones.
- El tercio restante no usado es llamado *observaciones out-of-bag* (OOB).
- Se puede predecir la respuesta para la $i-$ésima observación usando los árboles en los que la observación fue OOB. Lo anterior permite aproximadamente $B/3$ predicciones que luego se promedian.
- Esta estimación es, en esencia, el *error por validación cruzada LOO* para bagging cuando B es grande.

Random Forest

- *Random Forest* permite una mejora que reduce la correlación de los árboles y a la vez reduce la variancia cuando se promedia.
- Así como en bagging se construye una cantidad de árboles de decisión sobre las muestras bootstrap.
- En cada paso se elige al azar m predictores. La separación se realiza solo con uno de los m predictores.
- Se realiza una nueva selección de m predictores en cada separación.
- Se suele usar $m \approx \sqrt{p}$.