Mineria de Datos

Arboles de Decision

Dr. Edgar Acuna
Departmento de Matematicas

Universidad de Puerto Rico- Mayaguez

academic.uprm.edu/eacuna

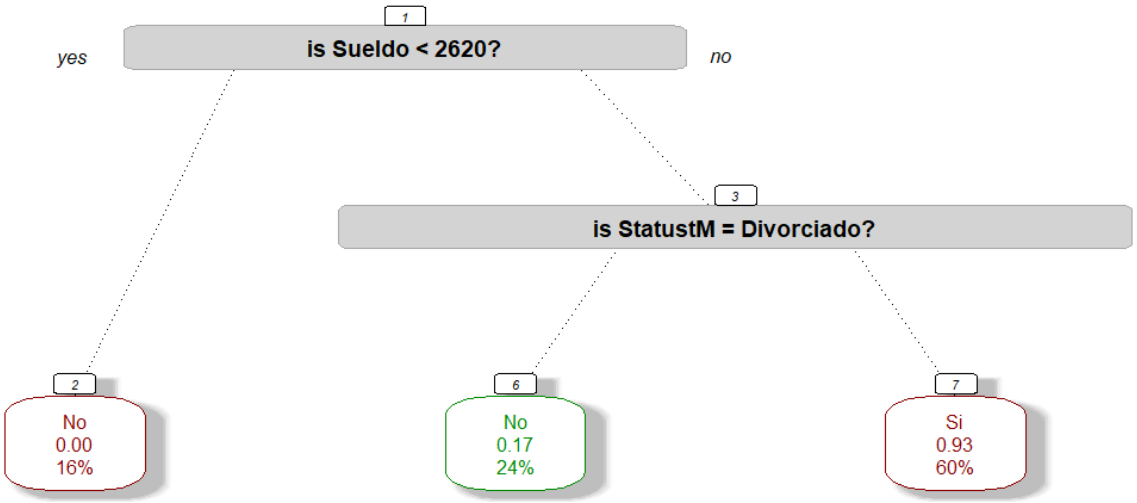
El uso de árboles de decisión tuvo su origen en las ciencias sociales con los trabajos de Sonquist y Morgan (1964) y Morgan y Messenger (1979) realizado en el Survey Research Center del Institute for Social Research de la Universidad de Michigan. El programa THAID (Theta Automatic Interaction Detection), de Sonquist, Baker y Morgan (1971), fue uno de los primeros métodos de ajuste de los datos basados en árboles de clasificación.

En estadística, Kass (1980) introdujo un algoritmo recursivo de clasificación no binario, llamado CHAID (Chi-square automatic interaction detection). Más tarde, Breiman, Friedman, Olshen y Stone (1984) introdujeron un nuevo algoritmo para la construcción de arboles y los aplicaron a problemas de regresión y clasificación. El método es conocido como CART (Classification and regression trees) por sus siglas en inglés. Casi al mismo tiempo el proceso de indución mediante árboles de decisión comenzó a ser usado por la comunidad de "Machine Learning" (Michalski, (1973), Quinlan (1983)). En el área de "Pattern Recognition" , Henrichon y Fu (1969) escribieron un paper en particionamiento noparametrico que guarda cierta relacion con arboles de decisión, pero usa hiperplanos que no son paralelos a los ejes coordenados.

Ejemplo: Prestamo para comprar carro

Sexo	Familia	CasPropia	AnosEmpleo	Sueldo	StatustMarital	Prestamo
Hombre	3	No	17	2500	Soltero	No
Mujer	5	Si	10	3000	Casado	Si
Mujer	4	No	15	2000	Viudo	No
Hombre	3	Si	16	2800	Soltero	Si
Hombre	6	Si	11	4000	Viudo	Si
Mujer	4	Si	26	3200	Soltero	Si
Mujer	2	Si	14	1800	Soltero	No
Hombre	5	Si	10	3750	Casado	Si
Hombre	6	No	18	2970	Divorciado	No
Hombre	4	Si	12	3350	Divorciado	No
Hombre	1	No	23	1950	Soltero	No
Mujer	2	Si	25	2740	Soltero	Si
Mujer	3	No	7	3100	Soltero	Si
Hombre	5	Si	5	3845	Divorciado	No
Hombre	3	No	13	3200	Casado	Si
Mujer	3	Si	9	2800	Soltero	No
Hombre	2	No	6	3200	Soltero	Si
Hombre	3	Si	7	3815	Viudo	Si
Mujer	2	Si	11	2980	Divorciado	No
Hombre	4	Si	15	2850	Viudo	Si
Mujer	1	No	6	3125	Divorciado	No
Hombre	1	No	8	3500	Soltero	Si
Hombre	4	No	22	4500	Divorciado	Si
Hombre	2	Si	10	3200	Casado	Si
Hombre	3	Si	9	3000	Casado	Si

Ejemplo: Prestamo para comprar carro



El tamano el arbol puede cambiarse cambiando parametros

Algoritmos para arboles de Decisión

C4.5. Introducido por Quinlan (1993) dentro de la comunidad de "Machine Learning". Es decendiente del ID3 (Quinlan, 1986).

CHAID. Significa "Chi-square automatic interaction detection", fue introducido por Kass (1980) y es un derivado del THAID: "A sequential search program for the analysis of nominal scale dependent variables" (Morgan and Messenger, 1973). El criterio para particionar está basado en $\chi 2$.

NewId. (Boswell, 1990). Es descendiente del ID3 (Quinlan, 1986)

CART. Introducido por Breiman et al. (1984), propiamente es un algoritmo de árboles de decisión binario. Existe una versión similar llamada IndCART y que está disponible en el paquete IND distribuido por la NASA. RPART (PARTicionamiento Recursivo), una versión de CART esta disponible en R.

Arboles Bayesianos: Está basado en aplicación de métodos Bayesianos a arboles de decisión. Buntine (1992). Disponible en el paquete IND distribuido por la NASA.

CN2. Introducido por Clark and Niblett (1989).

Construcción de arboles de decisión

Un árbol de decisión particiona el espacio de variables predictoras en un conjunto de hiper-rectángulos y en cada uno de ellos ajusta un modelo sencillo, generalmente una constante. Es decir y=c, donde y es la variable de respuesta.

La construcción de un árbol de decisión se basa en cuatro elementos:

a) Un conjunto de preguntas binarias Q de la forma { $x \in A$?} donde A es un subconjunto del espacio muestral de la variable x.

b) El método usado para particionar los nodos.

c) La estrategia requerida para parar el crecimiento del árbol.

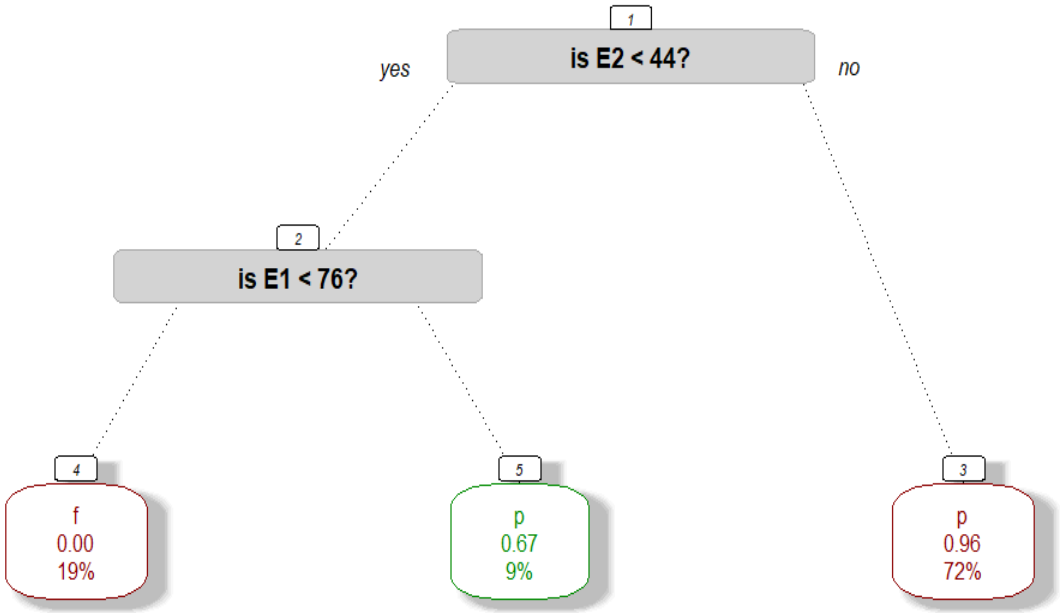
d) La asignación de cada nodo terminal a un valor de la variable de respuesta (regresión) o a una clase (clasificación).

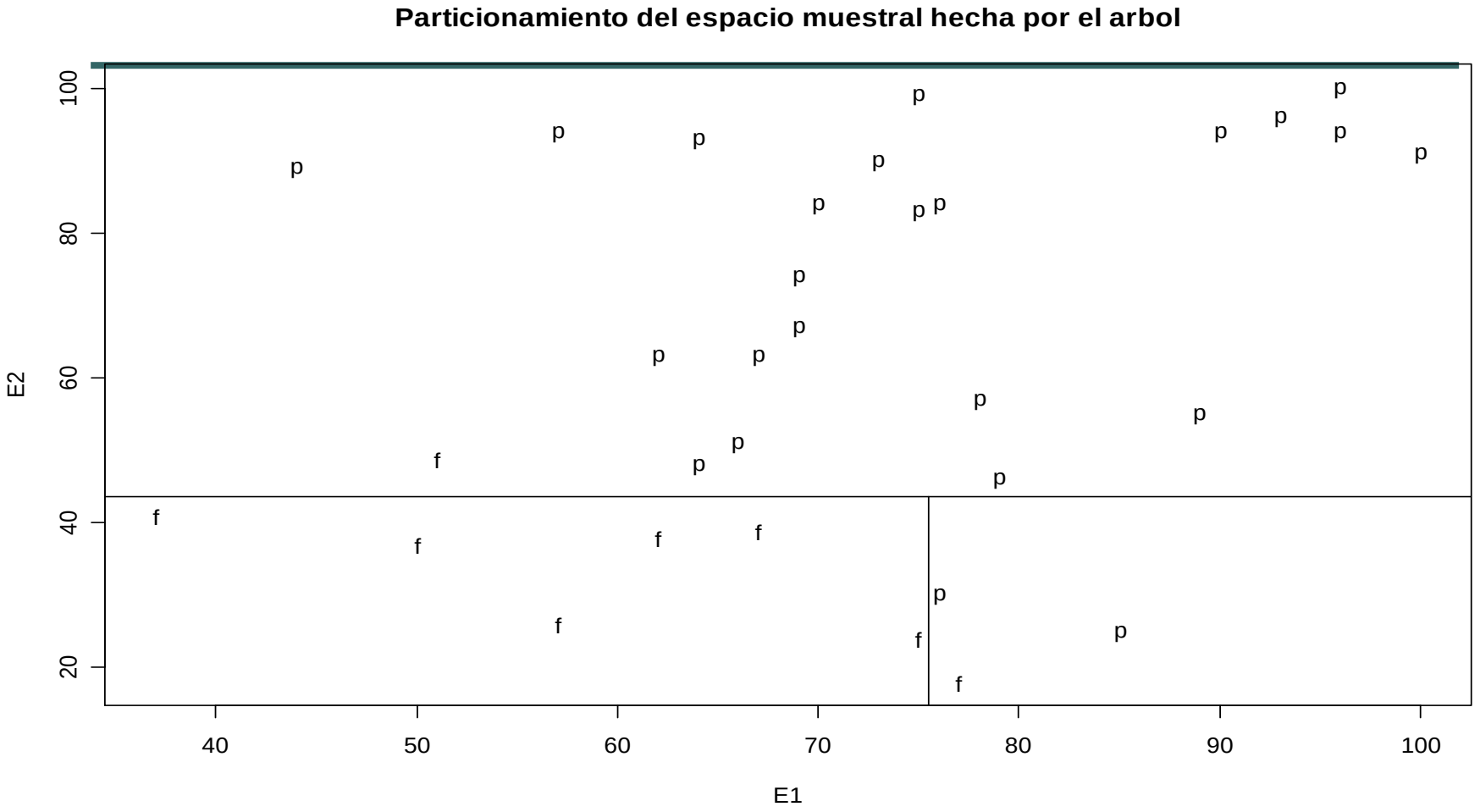
Las diferencias principales entre los algoritmos para construir árboles se hallan en la regla para particionar los nodos, la estrategia para podar los árboles, y el tratamiento de valores perdidos (“missing values”).

Ejemplo de arbol para clasificacion supervisada

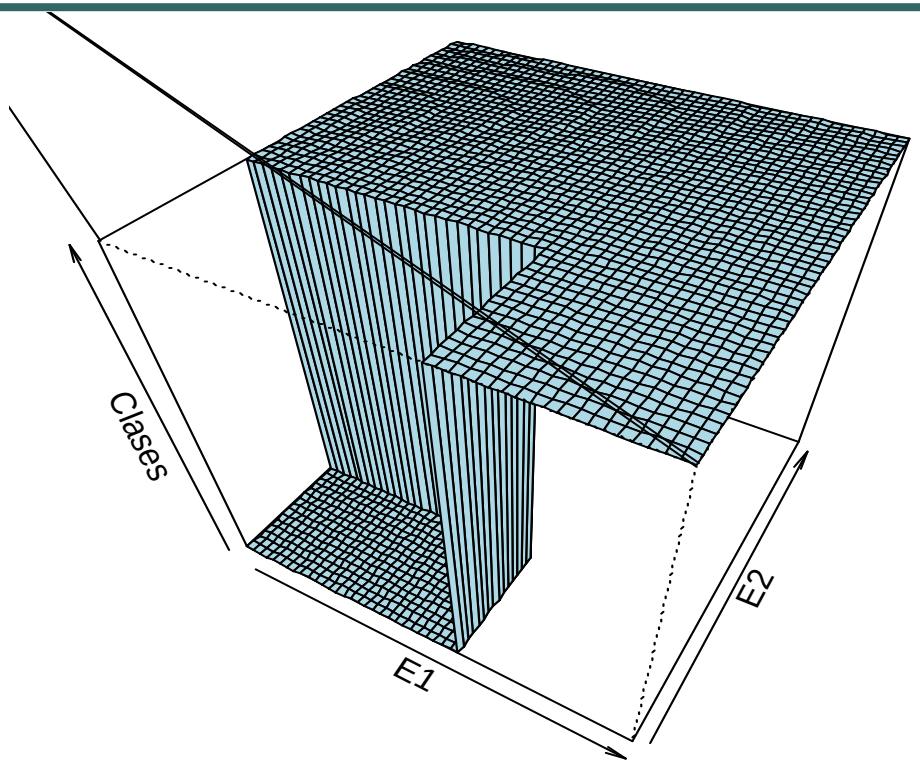
```
library(rpart)
eje1dis=read.table("http://academic.uprm.edu/eacuna/eje1disc.dat",header=T)
arbol=rpart(Nota~E1+E2,
     data=eje1dis,method="class",control=rpart.control(minbucket=2))
 print(arbol)
n= 32

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 32 8 p (0.25000000 0.75000000)
  2) E2< 43.5 9 2 f (0.77777778 0.22222222)
    4) E1< 75.5 6 0 f (1.00000000 0.00000000) *
    5) E1>=75.5 3 1 p (0.33333333 0.66666667) *
  3) E2>=43.5 23 1 p (0.04347826 0.95652174) *
```

ARBOL para el ejemplo anterior

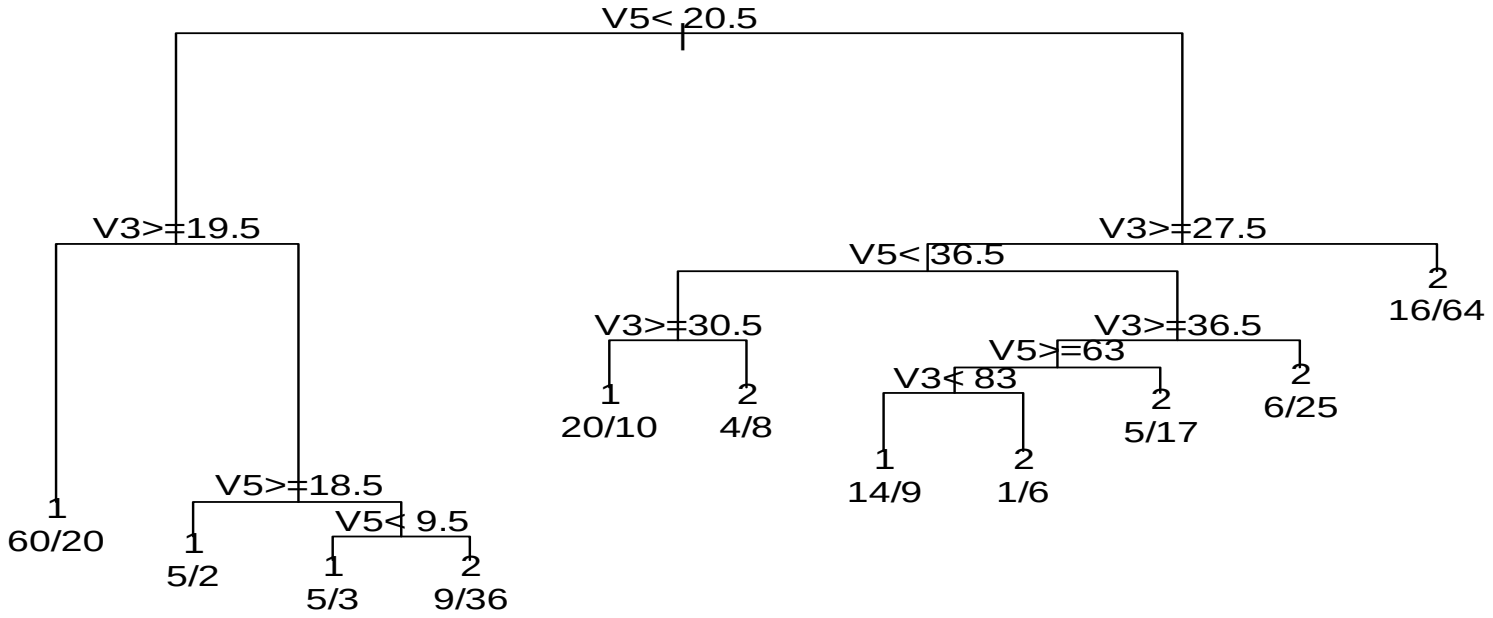



Particionamiento del espacio muestral hecha por el arbol
E2
E1
100
80
60
40
20
40
50
60
70
80
90
100

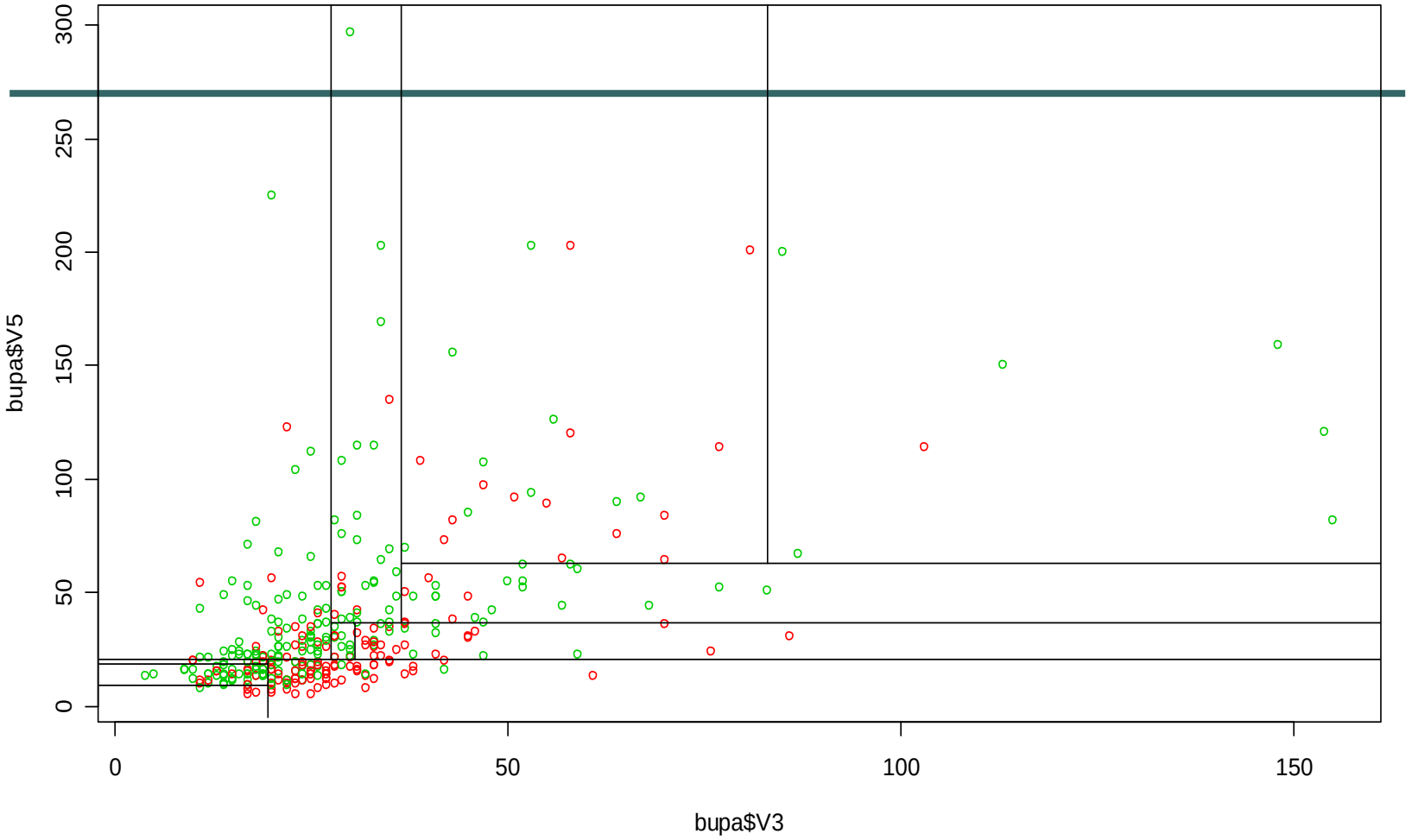
superficie de decision generada por el arbol



Ejemplo: clasificacion de bupa con atributos V3 y V5




```
> plot(arbolbupa,margin=.25)
> text(arbolbupa,use.n=T)
```



Particion del espacio muestral segun arbol de decision

I-El conjunto de preguntas binarias Q

- Supongamos que el vector de variables predictoras es de la forma $\mathbf{x}=(x_1,\ldots..x_p)$, donde algunas de las variables x_i son discretas y otras son continuas. Entonces, el conjunto Q de preguntas binarias en los nodos debe tener las siguientes características
  - a) cada división de los nodos depende del valor de una sola variable predictora
  - b) Si la variable x_k es continua entonces Q incluye todas las preguntas de la forma {Es $x_k \leq c$}, donde c es cualquier número real. Usualmente c es el punto medio entre dos valores consecutivos de un atributo.

c) Si la variable x_k es categórica tomando valores en $\{b_1,b_2,\ldots\ldots b_m\}$ entonces Q incluye todas las preguntas de la forma $\{ x_k \in A?\}$ donde A es un subconjunto cualquiera de $\{b_1,b_2,\ldots\ldots b_m\}$. En total se pueden considerar $2^{m-1}-1$

- Por ejemplo si x2, x3 y x4 son variables predictoras continuas y x1 es categórica con valores 0, 1 y 2, entonces Q incluye preguntas de la siguiente forma:

Es x3≤ 4.5?

Es x4≤ -1.4?

Es x1= 0 ó 1?

- También se puede usar divisiones en mas de dos nodos, pero no se recomienda porque el conjunto de datos se dividiría muy rápido dejando muy pocos datos para las subsiguientes divisiones.

II-Procedimiento para particionar los nodos

La idea fundamental es que los nodos hijos sean más puros que los nodos padres. La partición de un nodo t del árbol T se hace de acuerdo a un criterio que es diseñado para producir nodos hijos que produzcan una suma de cuadrados de errores menor que la del nodo padre en el caso de regresión o que separen mejor las clases que el nodo padre en el caso de clasificación.

En árboles de clasificación sean $p(s)=\{\#\ i\leq N: Xi\in s\}/N$ la proporción de observaciones en el nodo s, y

$p(j/s)=\{\#\ i\leq N: Xi \in s \text{ y } Yi=j\}/\{\#\ i\leq N: Xi \in s\}$

la proporción de observaciones en el nodo s que pertenecen a la clase j (j=1,….,J), donde J es el número de clases.

El indice de la impureza del nodo s se define como $i(s)=\varphi(p(1/s),p(2/s),\ldots.p(J/s))$ donde φ es una función de impureza, la cual debe satisfacer ciertas propiedades.

Entonces la regla para particionar el nodo t es como sigue:

Formar el nodo hijo derecho t_R y el nodo hijo izquierdo t_L tal que la disminución de la impureza dada por

$\Delta i(t)= i(t)-\{p(t_L)i(t_L)+p(t_R)i(t_R)\}$

sea máxima.

Medidas de Impureza

Para árboles de clasificación se pueden usar las siguientes medidas de impureza.

a) El ***Coeficiente de Gini***. *Para el nodo t* ***se*** *define por*

$i_G(t) = \sum_j \sum_k p(j/t)\,p(k/t) = \sum_{j=1}^{J} p(j/t)(1-p(j/t))$

Si hay dos clases (J=2) presentes entonces

$i_G(t)=2p(1-p)$,

donde p es la proporción de observaciones en la primera clase.

El coeficiente de Gini se puede interpretar como uno en donde se clasifica cada observación de un nodo a una clase j con probabilidad p(j/t) en lugar de clasificar todas las observaciones a la clase con mayor número de observaciones.

b) La ***Entropia*** **Cruzada o Devianza o Impureza de Información** definida por

$i_E(t)= -\Sigma_j\, p(j/t)\log[p(j/t)]$

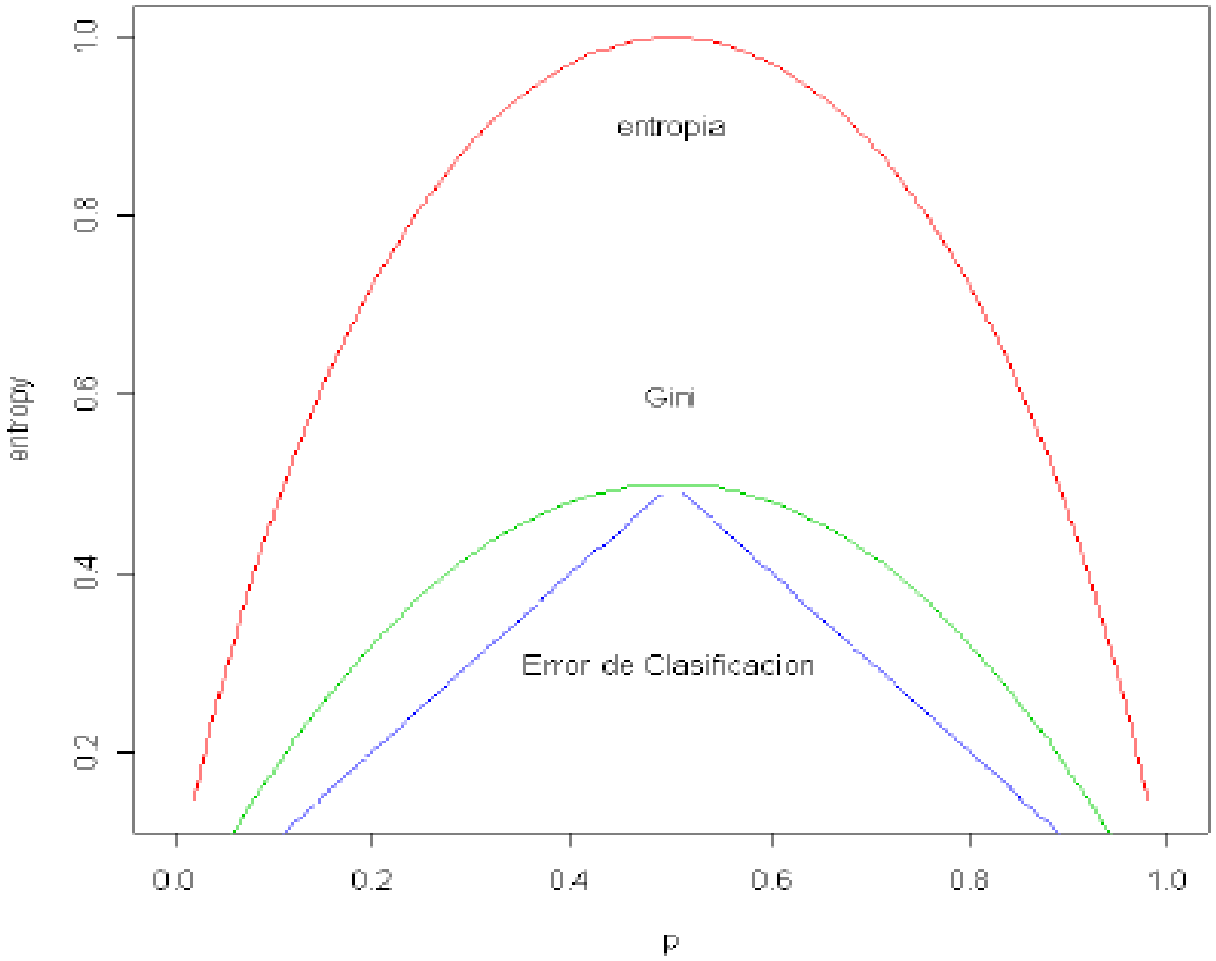
El logaritmo es tomado en base 2. Cuando se aplica entropía a distribuciones continuas se aplica logaritmo natural. Para dos clases se tiene

$i_E(t)= -p*\log(p) -(1-p)*\log(1-p)$

En regresion, La Devianza es equivalente a la suma de cuadrados de los errores y está dada por el negativo del doble del logaritmo de la función de verosimilitud.

Rpart usa por default la impureza Gini, pero tiene la opción de usar la impureza de información. C4.5 usa esta ultima.

- c) La ***tasa de Mala clasificación***, definida por $i_{MC}(t)=1-\max_j p(j/t)$ Para dos clases sería: $i_{MC}(t)=1-\max(p,1-p)$



Ejemplo 1: calculo de la impureza

En el ejemplo de prestamo para comprar carros si consideramos la particion en los nodos hijos

t_R:Status Marital= Soltero tenemos que a 4 No se le dio prestamo y a 6 si. Por otro lado,

t_L: Status Marital= Casado, Viudo o Divorciado, tenemos que a 6 No se les presto y a 9 si.

La impureza de t_R sera usando:

Entropia: -4/10*log2(4/10)-6/10*log2(6/10)=.9709

El coeficiente de Gini: 2(4/10)(6/10)=48/100=.48

Error de mala clasificacion:1-max(4/10,6/10)=4/10=.4

La impureza de t_L sera usando:

Entropia: -6/15*log2(6/15)-9/15*log2(9/15)=.9709

El coeficiente de Gini: 2(6/15)(9/15)=108/225=.48

Error de mala clasificacion:1-max(6/15,9/15)=6/15=.4

Ejemplo 1: calculo de la impureza

La impureza del nodo padre, donde hay 15 que se les dio prestamo y 10 que no se les dio prestamo es

Entropia= -(10/25)*log2(10/25)-(15/25)*log2(15/25)=.9709

Gini=2*10/25*15/25=.48

Error de Mala clasificacion:1-max(10/25,15/25)=.4

Por lo tanto, la reduccion de impureza con cada una de las funciones sera

Δi(t)= .9709-(10/25*.9709+15/25*.9709)=0(Entropia)

Δi(t)= .48-(10/25*.48+15/25*.48)=0(Gini)

Δi(t)= .4-(10/25*.4+15/25*.4)=0 (ME)

Sin embargo la particion en el los nodos hijos t_R: Sueldo menor que 2990 (7 no y 3 Si) y t_L: Sueldo mayor o igual que 2990 (3 No y 12 Si), producira la mayor reduccion en la impureza. Aqui solo se muestra la reduccion por el metodo de la entropia

Δi(t)= .9709-(10/25*.8812+15/25*.7219)=.1852

La reduccion en la impureza es mayor que cuando se usa la particion anterior, asi que es una mejor particion.

Ejemplo 2: calculo de la impureza

En el ejemplo de las notas si consideramos dividir en E2=47, entonces los nodos hijos seran:

t_R:E2<47, tenemos 3 que pasaron y 7 que no pasaron. Por otro lado,
t_L:E2>47, tenemos 21 que pasaron y 1 que no pasaron. Luego,

La impureza de t_R sera usando:
Entropia: -3/10*log2(3/10)-7/10*log2(7/10)=.8812
El coeficiente de Gini: 2(3/10)(7/10)=.42
Error de mala clasificacion:1-max(3/10,7/10)=3/10=.3

La impureza de t_L sera usando:
Entropia: -21/22*log2(21/22)-1/22*log2(1/22)=.2667
El coeficiente de Gini: 2(21/22)(1/22)=.086
Error de mala clasificacion:1-max(21/22,1/22)=1/22=.045

Ejemplo 2: calculo de la impureza

La impureza del nodo padre, donde hay 8 que fracasaron y 24 que pasaron es usando

Entropia= -(8/32)*log2(8/32)-(24/32)*log2(24/32)=.8112

Gini=2*8/32*24/32=.375

Error de Mala clasificacion:1-max(8/32,24/32)=8/32=.25

Por lo tanto, la reduccion de impureza con cada una de las funciones sera

Δi(t)= .8112-(10/32*.8812+22/32*.2667)=.3524(Entropia)

Δi(t)= .375-(10/32*.42+22/32*.086)=.184 (Gini)

Δi(t)= .25-(10/32*.3+22/32*.045)=.125 (ME)

Sin embargo la particion en el punto E2=43.5 producira la mayor reduccion. Aqui solo se muestra la reduccion por el metodo de la entropia

Δi(t)= .8112-(9/32*.764+23/32*.2580)=.4108.

La reduccion en la impureza es mayor que cuando se usa el nodo E2=47, asi que es una mejor particion.

Para arboles de regresion donde la variable de respuesta y es continua, se pueden usar las siguientes medidas de impureza

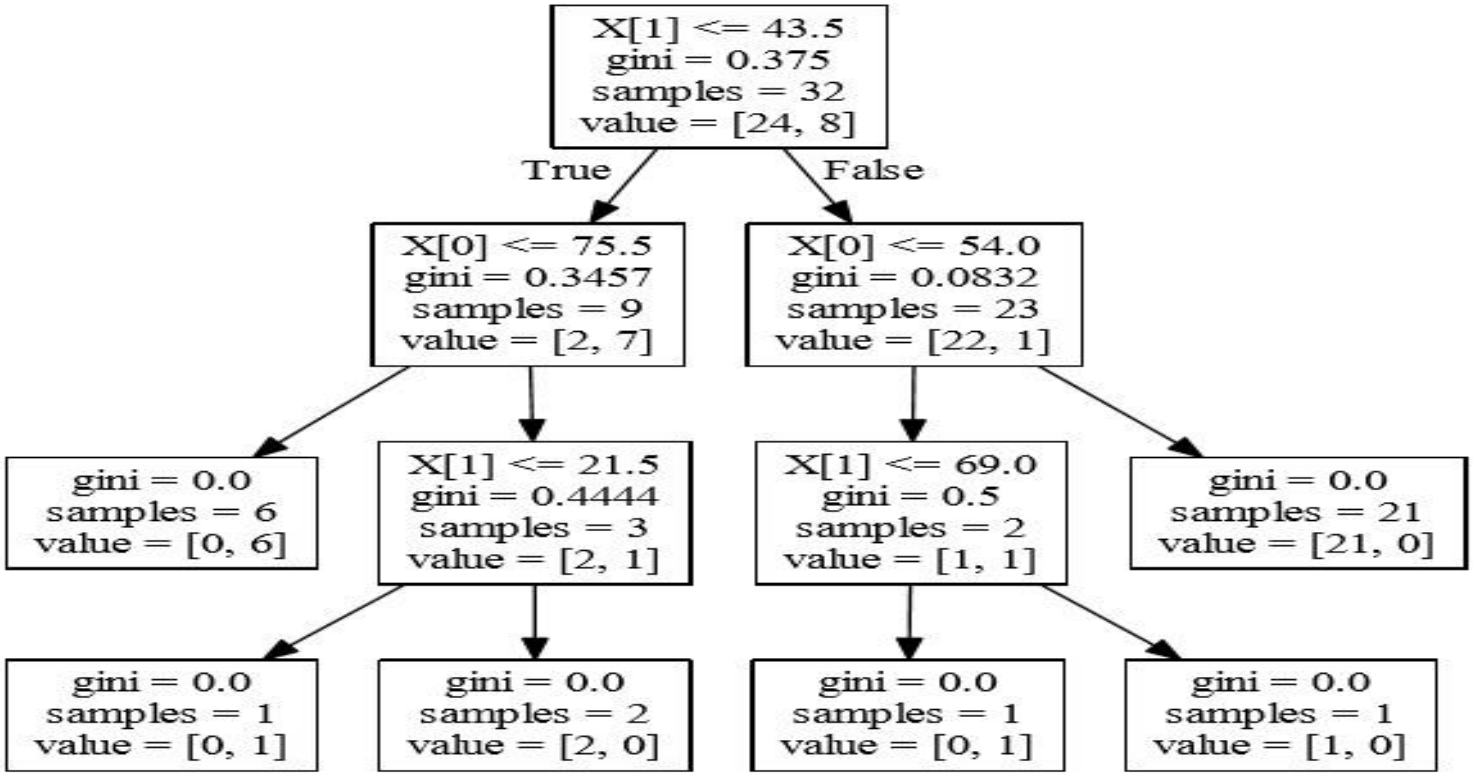
i) La Varianza, definida por

$$i(t) = \sum_{j \in t} \frac{(y_j - \bar{y}_t)^2}{n_t}$$

Donde $\bar{y}_t$ es la media de la variable de respuesta y en el nodo t.
ii) La desviacion absoluta mediana

$$i(t) = \sum_{j \in t} \frac{(y_j - \bar{y}_t)^2}{n_t}$$

Arbol con Python y GraphViz



III-Criterios para parar el crecimiento del árbol.

El crecimiento del arbol termina cuando es imposible continuar. Esto es,

(1) Hay una observacion en cada uno de los nodos hijos,

(2) Todas las observaciones en cada nodo hijo pertenecen a una misma clase, o

3) Se alcanza el numero maximo de niveles (“depth”) que debe tener el arbol, segun acordado por el usuario.

El arbol “mas grande” que se crea generalmente “sobreajusta”, es decir sigue mucho la tendencia de los datos. Las ultimas divisiones del arbol son las que generalmente causan el sobreajuste. Por lo tanto se requiere recortar el arbol.

Criterios para parar el crecimiento del árbol.

La función **rpart** de R tiene varios criterios de parada que son aplicados simultáneamente y son controlados con opciones de la función **rpart.control** .

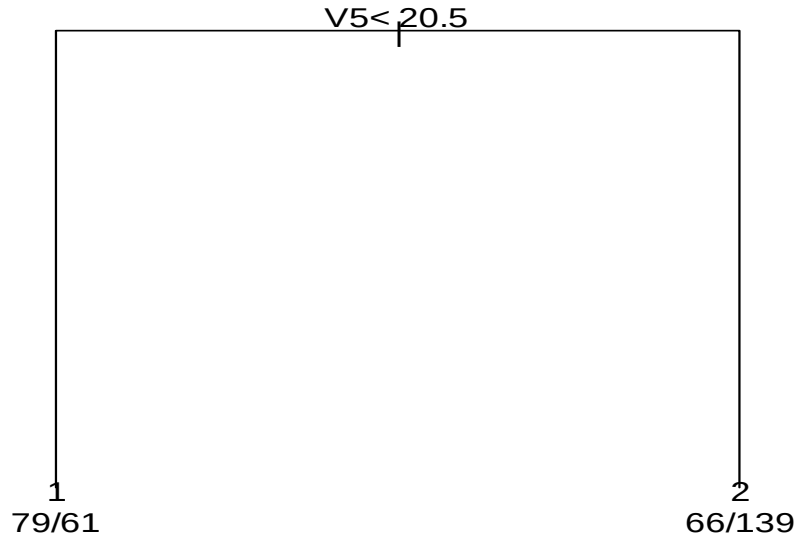
La opción **minsplit** fija el número mínimo de observaciones en un nodo para para que este sea dividido. Esta opción por defecto es 20.

La opción **minbucket** indica el número mínimo de observaciones en cualquier nodo terminal. Por defecto esta opción es el valor redondeado de minsplit/3.

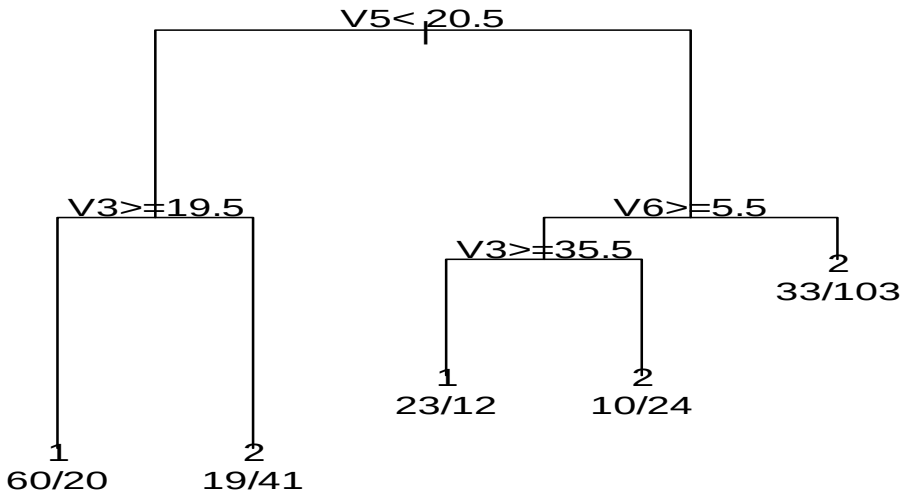
La opción **cp**; parámetro de complejidad. Indica que si el criterio de impureza no es reducido en mas de cp*100% entonces se para..Por defecto cp=.01. Es decir, la reducción en la impureza del nodo terminal debe ser menor del 1% de la impureza inicial.

Tambien hay un parámetro **maxdepth** que condiciona la profundidad máxima del arbol. Por defecto está establecida como 30.

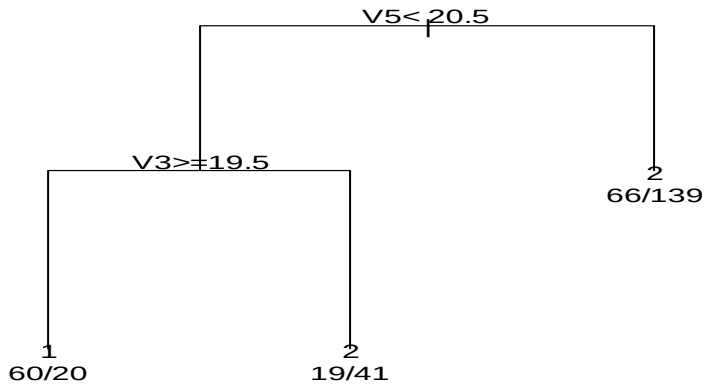
```
arbolbupa=rpart(V7~.,data=bupa, method="class",minbucket=50)
plot(arbolbupa,margin=.25)
text(arbolbupa,use.n=T)
```



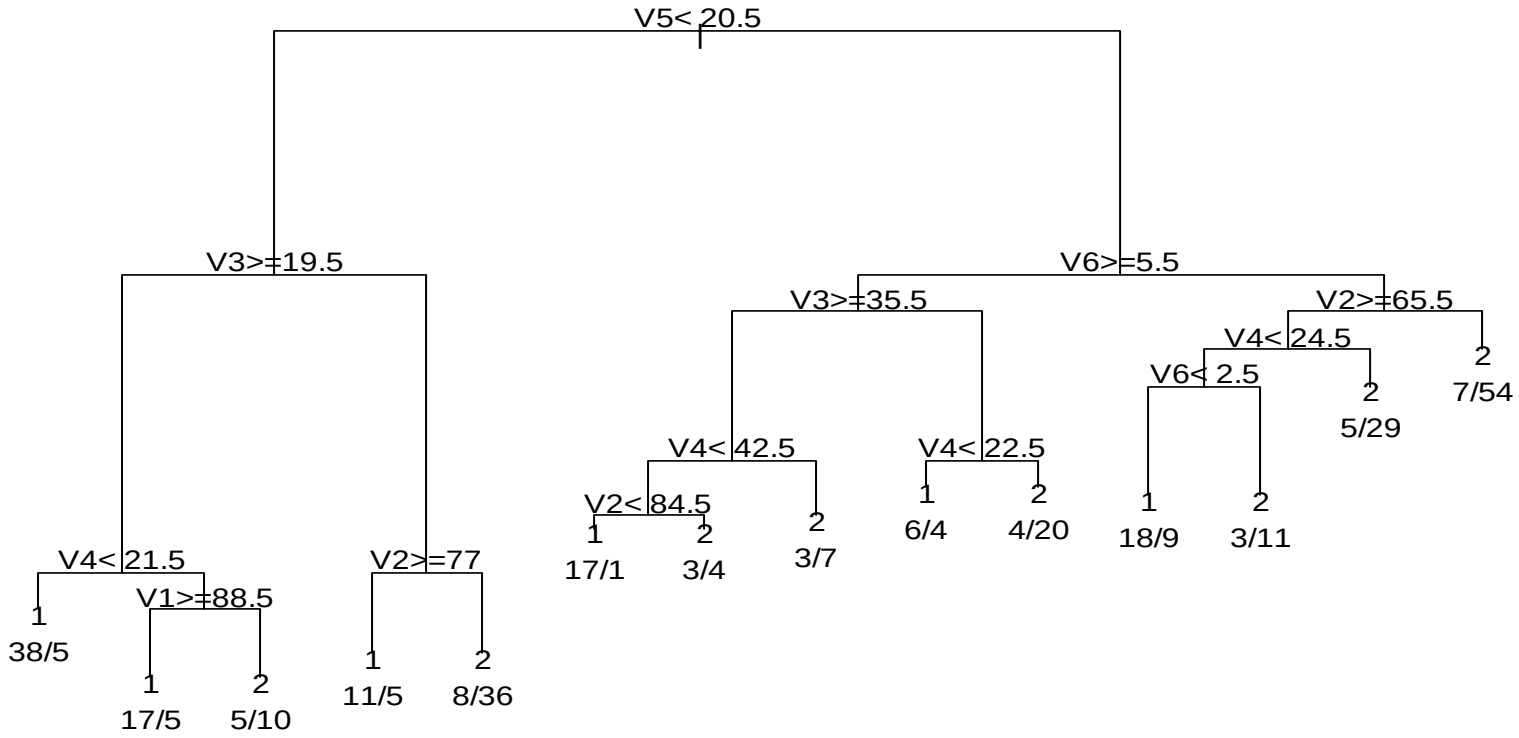
```
arbolbupa=rpart(V7~.,data=bupa, method="class",minbucket=20)
plot(arbolbupa,margin=.25)
 text(arbolbupa,use.n=T)
```



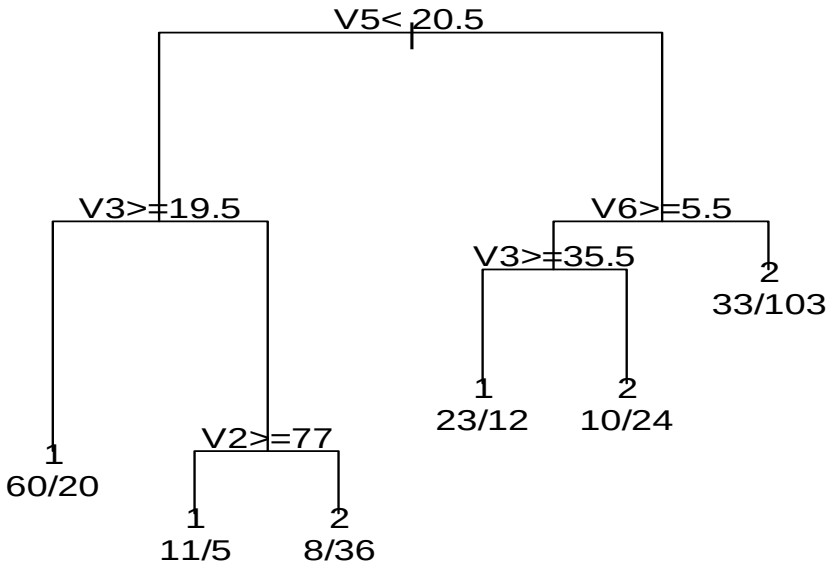
```
arbolbupa=rpart(V7~.,data=bupa, method="class",cp=.05)
plot(arbolbupa,margin=.25)
text(arbolbupa,use.n=T)
```



```
arbolbupa=rpart(V7~.,data=bupa, method="class",cp=.0001)
plot(arbolbupa,margin=.25)
text(arbolbupa,use.n=T)
```



```
arbolbupa=rpart(V7~.,data=bupa, method="class",maxdepth=3)
plot(arbolbupa,margin=.25)
text(arbolbupa,use.n=T)
```



Arboles de Decision en Rapidminer

Se sigue la secuencia de operadores: Modeling>Classification and Regression>Tree Induction>Regression Trees

Recortando ("prunning") un árbol.

- Hacer crecer un árbol demasiado grande puede crear problemas de "sobreajuste" , es decir el modelo puede seguir más al ruido que a la señal.

Para cualquier árbol T y cualquier $\alpha \geq 0$ (α es llamado el parámetro de complejidad), una medida del mérito del árbol T (o medida de costo-complejidad) está dada por:

$$R_\alpha(T) = Resub(T) + \alpha|T|$$

donde Resub(T) es estimado por resubsitución de la tasa de clasificación errada de T, $|T|$ es el número de nodos terminales de T. Cuando $\alpha=0$ se obtiene el árbol más grande y cuando $\alpha=\propto$ se obtiene un árbol con un solo nodo. Es decir, cuando α se incrementa se poda cada vez mas el arbol. El árbol óptimo T_α es el árbol mas pequeño que miminiza $R_\alpha(T)$.

- La función **prune** de la librería **rpart** ejecuta recorte de un árbol. La opción cp (α) de prune es llamado el parámetro de complejidad. El cp indica que se descartará cualquier partición que no disminuye la impureza por un factor igual a cp.
- prune(arbolbupa,cp=.05)

Estimación del Error de Clasificación

Breiman, et al (1984) recomiendan usar validación cruzada 10 para estimar el error de clasificación. Ellos no recomiendan "bootstrapping" porque han mostrado que el sesgo estimado subestima en menos del 40% al verdadero sesgo. Solo recomiendan su uso para muestras pequeñas.

La función **xpred.rpart** da las predicciones de la variable de respuesta usando validación cruzada para valores dados del parámetro de complejidad.

```
arbolexa=rpart(Class~E1+E2,data=eje1dis,method="class")
cvpred=xpred.rpart(arbolexa,xval=10)
error=mean(cvpred !=as.numeric(eje1dis[,3]))
[1] 0.2343
arbolbupa=rpart(V7~.,data=bupa,method="class")
cvpred=xpred.rpart(arbolbupa,xval=10)
error=mean(cvpred !=bupa[,7])
[1] 0.3391
```

Tratamiento de valores perdidos en clasificación por árboles

El procedimiento para tratar los casos con observaciones perdidas es como sigue:

Se determina la mejor partición del nodo basado en una variable digamos x_k con los datos que se tiene disponible. Si cuando se desea clasificar una observación del conjunto de entrenamiento o de prueba no hay el valor correspondiente de la variable x_k entonces se usa la “partición sustituta“ y si no hubiera un valor

observado de la variable envuelta en la variable sustituta entonces se usa una segunda partición sustituta y asi sucesivamente. Esto es algo similar a cuando en un modelo lineal se reemplaza el valor perdido de una variable predictora por el valor predicho por su regresion con otra variable predictora que esta más altamente correlacionada con ella.

```
bupamiss <- unlist(bupa); bupamiss[sample(2070,100)] = NA
b=as.data.frame(matrix(bupamiss, ncol=7))
arbolbupa=rpart(V7~.,data=as.data.frame(b),method="class")
```

Ventajas y desventajas de clasificación por arboles

Ventajas:

- Puede ser aplicado a cualquier tipo de variables predictoras: continuas y categóricas
- Los resultados son fáciles de entender e interpretar.
- No tiene problema de trabajar con datos perdidos.
- Hace automáticamente selección de variables.
- Es invariante a transformaciones de las variables predictoras.
- Es robusto a la presencia de "outliers".
- Es un clasificador noparamétrico, es decir que no requiere suposciones.
- Es rápido de calcular.

Desventajas:

- El proceso de selección de variables es sesgado hacia las variables con mas valores diferentes.
- Dificultad para elegir el árbol óptimo
- La superficie de predicción no es muy suave, ya que son conjuntos de planos.
- Requiere un gran numero de datos para asegurarse que la cantidad de observaciones en los nodos terminales es significativa.
- Ausencia de una función global de las variables y como consecuencia pérdida de la representación geométrica.
- No toma en cuenta las interacciones que puede existir entre las variables predictoras.