Anexo 3

Proceso de análisis - Árbol de decisión

# Introducción a los árboles de decisión

En este anexo mostramos el proceso de análisis utilizado

# Selección de datos de análisis

Seleccionamos las variables que incluíremos en nuestra base de datos. Por un lado, eliminaremos las variables numéricas, en tanto utilizaremos el árbol de decisión como método de clasificación y análisis. Por otro lado, las variables numéricas que nos interesen, las recodificaremos como categóricas.

Eliminaremos aquellas variables que añaden ruido innecesario, así como las que resulten redundantes con la variable dependiente.

# Edad por franjas   
  
foessa2 <- mutate(foessa2, edad\_franjas = case\_when(  
 edad <= 17 ~ "0-17 años",   
 edad >= 18 & edad <= 25 ~ "18-25 años",   
 edad >= 26 & edad <= 35 ~ "26-35 años",   
 edad >= 36 & edad <= 50 ~ "36-50 años",   
 edad >= 51 & edad <= 64 ~ "51-64 años",   
 edad >= 65 ~ "Más de 65 años"  
))  
  
foessa2$edad\_franjas <- as.factor(foessa2$edad\_franjas)  
  
  
# Eliminamos las variables que no nos interesan (103 iniciales)  
  
foessat <- select(foessa2, -c(CCAA,   
 PROVINCIA,   
 edad,   
 tamano\_hogar,   
 ingresos\_UC,   
 ingresos\_hogar,   
 gasto\_energia,   
 gasto\_agua,   
 rehab\_cocina,   
 rehab\_baño,   
 rehab\_instal,   
 rehab\_calef,   
 rehab\_ventana,   
 rehab\_tabiques,  
 rehab\_suelo,  
 rehab\_barreras,   
 ingresos\_calidad,   
 gasto\_energia\_num,   
 share\_energy,  
 TWO\_M\_Sí,  
 HEP\_Sí,  
 temp\_adecuada\_No\_w,  
 retrasos1\_Sí\_w,  
 TWO\_M\_Sí\_w,  
 HEP\_Sí\_w,   
 Vulnerabilidad\_num,  
 retrasos1\_Sí,   
 temp\_adecuada\_r,   
 temp\_adecuada\_No,   
 TWO\_M,   
 HEP,   
 retrasos,   
 temp\_adecuada,   
 retrasos1,   
 Vulnerabilidad\_Energetica,  
 PE,   
 dinero\_gastoscasa,   
 avisos\_cortes  
 ))  
  
# Eliminamos las etiquetas de las variables  
  
# install.packages("sjlabelled")  
library(sjlabelled)  
  
foessat <- remove\_all\_labels(foessat)  
  
str(foessat) # El resultado final nos da una base de datos con 69 variables categóricas

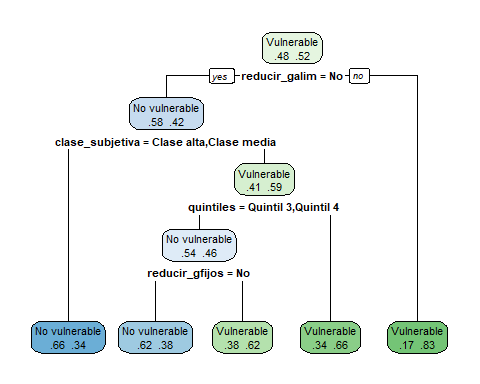
## 'data.frame': 29953 obs. of 66 variables:  
## $ sexo : Factor w/ 2 levels "Varón","Mujer": 2 1 2 1 2 1 2 1 2 1 ...  
## $ educacion : Factor w/ 11 levels "01. No sabe leer o escribir",..: 10 10 10 9 7 5 5 11 5 5 ...  
## $ salud : Factor w/ 5 levels "Muy buena","Bastante buena",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ dependencia : Factor w/ 7 levels "Sí, gran dependencia (Grado 3)",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ excl4grupos : Factor w/ 4 levels "Integracion plena",..: 3 3 3 1 1 3 3 3 2 2 ...  
## $ exclusion : Factor w/ 2 levels "No exclusion",..: 2 2 2 1 1 2 2 2 1 1 ...  
## $ etnia : Factor w/ 3 levels "Todos españoles o UE15",..: 2 2 2 1 1 1 1 1 1 1 ...  
## $ monoparental : Factor w/ 2 levels "Hogar monoparental",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ anciano : Factor w/ 2 levels "No","Si": 1 1 1 1 1 1 1 1 2 2 ...  
## $ menor : Factor w/ 2 levels "Hogar sin menores de 18 años",..: 1 1 1 1 1 2 2 2 1 1 ...  
## $ joven : Factor w/ 2 levels "Hogar sin jovenes 18-24 años",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ discapacidad : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ barrio\_dummy : Factor w/ 2 levels "Barrio buenas condiciones",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ barrio : Factor w/ 4 levels "Zona marginal",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ alojamiento : Factor w/ 7 levels "Chabola","Cueva",..: 5 5 5 6 6 5 5 5 5 5 ...  
## $ tamano\_municipio : Factor w/ 5 levels "Más de 100.000",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ ocupado : Factor w/ 2 levels "Hogar sin ocupados según D35",..: 2 2 2 2 2 2 2 2 1 1 ...  
## $ parado : Factor w/ 2 levels "Hogar sin parados EPA",..: 2 2 2 2 2 2 2 2 1 1 ...  
## $ clase\_subjetiva : Factor w/ 4 levels "Clase alta","Clase media",..: 3 3 3 2 2 3 3 3 2 2 ...  
## $ evolucion12 : Factor w/ 6 levels "Con mucha dificultad",..: 1 1 1 4 4 1 1 1 4 4 ...  
## $ reducir\_gfijos : Factor w/ 2 levels "Sí","No": 1 1 1 2 2 1 1 1 2 2 ...  
## $ reducir\_galim : Factor w/ 2 levels "Sí","No": 2 2 2 2 2 1 1 1 2 2 ...  
## $ dieta\_inadec : Factor w/ 2 levels "Sí","No": 2 2 2 2 2 1 1 1 2 2 ...  
## $ reducir\_ocio : Factor w/ 2 levels "Sí","No": 1 1 1 2 2 1 1 1 1 1 ...  
## $ perdida\_relaciones : Factor w/ 2 levels "Sí","No": 2 2 2 2 2 2 2 2 1 1 ...  
## $ insalubridad : Factor w/ 3 levels "No","Sí","No Consta": 1 1 1 1 1 2 2 2 1 1 ...  
## $ entorno\_degradado : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 2 2 2 1 1 ...  
## $ barrio\_conflictivo : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ necesidad\_vivienda : Factor w/ 4 levels "Necesitan cambiar de vivienda",..: 4 4 4 4 4 1 1 1 4 4 ...  
## $ dispone\_agua : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_agua\_cal : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_elect : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_calef : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ ppalocupacion : Factor w/ 4 levels "Trabaja","Parado",..: 1 1 1 1 1 1 1 1 3 3 ...  
## $ tenencia : Factor w/ 11 levels "Por compra, totalmente pagada",..: 7 7 7 7 7 6 6 6 1 1 ...  
## $ dispone\_baño : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_cocina : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_frigo : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_lava : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ dispone\_pc : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 2 2 ...  
## $ dispone\_internet : Factor w/ 2 levels "Si","No": 1 1 1 1 1 2 2 2 2 2 ...  
## $ dispone\_tv : Factor w/ 2 levels "Si","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ ruina : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ discrim\_etnia : Factor w/ 2 levels "No","Sí": 2 2 2 1 1 2 2 2 1 1 ...  
## $ discrim\_mujer : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ discrim\_fisico : Factor w/ 2 levels "No","Sí": 2 2 2 1 1 1 1 1 1 1 ...  
## $ discrim\_sexual : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ discrim\_nunca : Factor w/ 2 levels "No","Sí": 1 1 1 2 2 1 1 1 2 2 ...  
## $ discrim\_nosabe : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ discrim\_nocont : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 1 1 ...  
## $ proteina : Factor w/ 2 levels "Si","No": 1 1 1 1 1 2 2 2 1 1 ...  
## $ Vulnerabilidad\_dummy: Factor w/ 2 levels "No vulnerable",..: 2 2 2 1 1 2 2 2 1 1 ...  
## $ tamano\_hogarf : Factor w/ 5 levels "1 miembro","2 miembros",..: 3 3 3 2 2 3 3 3 2 2 ...  
## $ dos\_adult\_65 : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 2 2 ...  
## $ monomarental : Factor w/ 2 levels "Hogar monomarental",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ monoparental\_h : Factor w/ 2 levels "Hogar monoparental (h)",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ quintiles : Factor w/ 5 levels "Quintil 1","Quintil 2",..: 1 1 1 5 5 1 1 1 3 3 ...  
## $ rehab : Factor w/ 2 levels "Necesita rehabilitación",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ rehab\_tipo : Factor w/ 10 levels "Baño","Barreras arquitectónicas",..: 6 6 6 6 6 6 6 6 6 6 ...  
## $ def\_hogar : chr "No" "No" "No" "No" ...  
## $ tenencia\_rec : Factor w/ 4 levels "Alquiler","Gratuita/Semigratuita",..: 1 1 1 1 1 1 1 1 4 4 ...  
## $ alojamiento\_rec : Factor w/ 3 levels "Otros","En piso",..: 2 2 2 3 3 2 2 2 2 2 ...  
## $ salud\_rec : Factor w/ 3 levels "Buena o muy buena",..: 1 1 1 1 1 1 1 1 2 2 ...  
## $ discriminacion : Factor w/ 2 levels "No","Sí": 2 2 2 1 1 2 2 2 1 1 ...  
## $ vuln\_familiar : Factor w/ 2 levels "No","Sí": 1 1 1 1 1 1 1 1 2 2 ...  
## $ edad\_franjas : Factor w/ 6 levels "0-17 años","18-25 años",..: 3 4 5 4 3 4 3 1 6 6 ...

Primero, dividimos nuestros datos en dos partes como datos de test i entrenamiento:

set.seed(123)  
sample\_data = sample.split(foessat, SplitRatio = 0.75)  
train\_data <- subset(foessat, sample\_data == TRUE)  
test\_data <- subset(foessat, sample\_data == FALSE)

A partir de aquí, podemos crear nuestro primer árbol de decision con el paquete rpart:

# Creamos el arbol de decisión   
  
rtree <- rpart(Vulnerabilidad\_dummy ~ ., train\_data,   
 method = "class")  
arbol1 <- rpart.plot(rtree,   
 extra = 4)



Estudiamos la evolución del error a medida que el árbol va creciendo

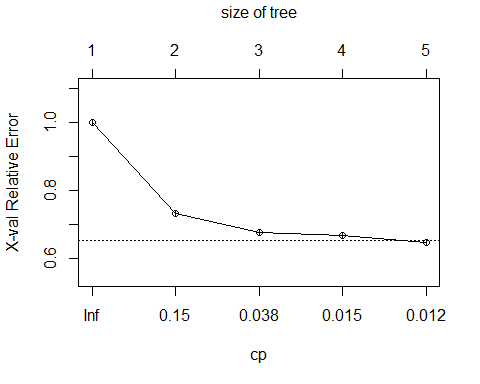
summary(rtree) # estadísticas detalladas de cada nodo

## Call:  
## rpart(formula = Vulnerabilidad\_dummy ~ ., data = train\_data,   
## method = "class")  
## n= 22237   
##   
## CP nsplit rel error xerror xstd  
## 1 0.24186003 0 1.0000000 1.0000000 0.006970852  
## 2 0.09421781 1 0.7581400 0.7342814 0.006667458  
## 3 0.01553144 2 0.6639222 0.6757111 0.006533665  
## 4 0.01450225 3 0.6483907 0.6669162 0.006511292  
## 5 0.01000000 4 0.6338885 0.6482972 0.006461896  
##   
## Variable importance  
## reducir\_galim dieta\_inadec evolucion12 clase\_subjetiva   
## 34 14 13 12   
## perdida\_relaciones proteina excl4grupos quintiles   
## 8 5 5 3   
## reducir\_gfijos reducir\_ocio ppalocupacion   
## 3 1 1   
##   
## Node number 1: 22237 observations, complexity param=0.24186  
## predicted class=Vulnerable expected loss=0.4806404 P(node) =1  
## class counts: 10688 11549  
## probabilities: 0.481 0.519   
## left son=2 (17055 obs) right son=3 (5182 obs)  
## Primary splits:  
## reducir\_galim splits as RL, improve=1323.998, (7 missing)  
## clase\_subjetiva splits as LLRR, improve=1315.873, (88 missing)  
## evolucion12 splits as RRLLLL, improve=1285.178, (72 missing)  
## reducir\_gfijos splits as RL, improve=1118.495, (42 missing)  
## reducir\_ocio splits as RL, improve=1049.980, (27 missing)  
## Surrogate splits:  
## dieta\_inadec splits as RL, agree=0.861, adj=0.404, (3 split)  
## evolucion12 splits as RLLLLL, agree=0.835, adj=0.292, (4 split)  
## perdida\_relaciones splits as RL, agree=0.822, adj=0.234, (0 split)  
## proteina splits as LR, agree=0.805, adj=0.162, (0 split)  
## excl4grupos splits as LLLR, agree=0.798, adj=0.132, (0 split)  
##   
## Node number 2: 17055 observations, complexity param=0.09421781  
## predicted class=No vulnerable expected loss=0.4242158 P(node) =0.766965  
## class counts: 9820 7235  
## probabilities: 0.576 0.424   
## left son=4 (11448 obs) right son=5 (5607 obs)  
## Primary splits:  
## clase\_subjetiva splits as LLRR, improve=459.5636, (77 missing)  
## quintiles splits as RRLLR, improve=373.5922, (2454 missing)  
## evolucion12 splits as RRRLLL, improve=358.6691, (69 missing)  
## reducir\_gfijos splits as RL, improve=340.5999, (28 missing)  
## reducir\_ocio splits as RL, improve=231.3717, (27 missing)  
## Surrogate splits:  
## evolucion12 splits as RRLLLL, agree=0.768, adj=0.296, (61 split)  
## reducir\_gfijos splits as RL, agree=0.712, adj=0.127, (14 split)  
## reducir\_ocio splits as RL, agree=0.708, adj=0.115, (0 split)  
## ppalocupacion splits as LRLL, agree=0.697, adj=0.083, (2 split)  
## excl4grupos splits as LLRR, agree=0.692, adj=0.065, (0 split)  
##   
## Node number 3: 5182 observations  
## predicted class=Vulnerable expected loss=0.1675029 P(node) =0.233035  
## class counts: 868 4314  
## probabilities: 0.168 0.832   
##   
## Node number 4: 11448 observations  
## predicted class=No vulnerable expected loss=0.3431167 P(node) =0.5148176  
## class counts: 7520 3928  
## probabilities: 0.657 0.343   
##   
## Node number 5: 5607 observations, complexity param=0.01553144  
## predicted class=Vulnerable expected loss=0.4102015 P(node) =0.2521473  
## class counts: 2300 3307  
## probabilities: 0.410 0.590   
## left son=10 (1998 obs) right son=11 (3609 obs)  
## Primary splits:  
## quintiles splits as RRLLR, improve=130.88390, (711 missing)  
## reducir\_gfijos splits as RL, improve= 99.87623, (19 missing)  
## excl4grupos splits as LLRR, improve= 90.91891, (0 missing)  
## exclusion splits as LR, improve= 90.91891, (0 missing)  
## evolucion12 splits as RRLLLL, improve= 78.89266, (15 missing)  
## Surrogate splits:  
## evolucion12 splits as RRRLRR, agree=0.624, adj=0.034, (706 split)  
## excl4grupos splits as LRRR, agree=0.623, adj=0.030, (5 split)  
## alojamiento splits as R-RLRRR, agree=0.612, adj=0.003, (0 split)  
## insalubridad splits as RRL, agree=0.611, adj=0.001, (0 split)  
## tenencia splits as RRRLRRRRRRR, agree=0.611, adj=0.001, (0 split)  
##   
## Node number 10: 1998 observations, complexity param=0.01450225  
## predicted class=No vulnerable expected loss=0.4584585 P(node) =0.08985025  
## class counts: 1082 916  
## probabilities: 0.542 0.458   
## left son=20 (1349 obs) right son=21 (649 obs)  
## Primary splits:  
## reducir\_gfijos splits as RL, improve=50.50283, (2 missing)  
## tenencia splits as LRRRLRRRLRL, improve=21.29737, (0 missing)  
## reducir\_ocio splits as RL, improve=13.07042, (1 missing)  
## proteina splits as LR, improve=12.07506, (19 missing)  
## rehab\_tipo splits as L--RRLL--L, improve=11.29564, (0 missing)  
## Surrogate splits:  
## reducir\_ocio splits as RL, agree=0.696, adj=0.063, (2 split)  
## barrio splits as LLLR, agree=0.678, adj=0.008, (0 split)  
## alojamiento splits as --LLLLR, agree=0.677, adj=0.003, (0 split)  
## tenencia splits as LLLLLLLLRRL, agree=0.677, adj=0.003, (0 split)  
##   
## Node number 11: 3609 observations  
## predicted class=Vulnerable expected loss=0.3374896 P(node) =0.1622971  
## class counts: 1218 2391  
## probabilities: 0.337 0.663   
##   
## Node number 20: 1349 observations  
## predicted class=No vulnerable expected loss=0.381023 P(node) =0.06066466  
## class counts: 835 514  
## probabilities: 0.619 0.381   
##   
## Node number 21: 649 observations  
## predicted class=Vulnerable expected loss=0.3805855 P(node) =0.02918559  
## class counts: 247 402  
## probabilities: 0.381 0.619

printcp(rtree) # estadísticas de resultados

##   
## Classification tree:  
## rpart(formula = Vulnerabilidad\_dummy ~ ., data = train\_data,   
## method = "class")  
##   
## Variables actually used in tree construction:  
## [1] clase\_subjetiva quintiles reducir\_galim reducir\_gfijos   
##   
## Root node error: 10688/22237 = 0.48064  
##   
## n= 22237   
##   
## CP nsplit rel error xerror xstd  
## 1 0.241860 0 1.00000 1.00000 0.0069709  
## 2 0.094218 1 0.75814 0.73428 0.0066675  
## 3 0.015531 2 0.66392 0.67571 0.0065337  
## 4 0.014502 3 0.64839 0.66692 0.0065113  
## 5 0.010000 4 0.63389 0.64830 0.0064619

plotcp(rtree) # evolución del error a medida que se incrementan los nodos



Validamos la capacidad de predicción del árbol con el fichero de validación

testPredRpart <- predict(rtree, newdata = test\_data, type = "class")  
  
# Visualizamos una matriz de confusión  
table(testPredRpart, test\_data$Vulnerabilidad\_dummy)

##   
## testPredRpart No vulnerable Vulnerable  
## No vulnerable 2924 1552  
## Vulnerable 819 2421

Calculamos el % de aciertos de nuestro modelo, utilizando los datos de test:

sum(testPredRpart == test\_data$Vulnerabilidad\_dummy)/ length(test\_data$Vulnerabilidad\_dummy)\*100

## [1] 69.27164

El resultado nos clasifica correctamente un 69,2% de los registros, por lo que es bastante mejorable.