www.nataliejulian.com
Natalie Julian

Ejercicios Árboles de decisión

Contexto

Es sabido que, una entidad que presta servicios o productos (pudiera ser una empresa, un banco, una tienda, etcétera) puede mejorar la experiencia de cliente desarrollando productos personalizados en pos de las preferencias y necesidades de cada uno de sus clientes.
La base de datos `potencial` contiene información sobre clientes de una institución financiera:

- `Customer ID` ID asociado al cliente
- `Age` Edad en años del cliente
- `Income` Ingreso anual del cliente
- `Family` Tamaño del grupo familiar del cliente
- `CCAvg` Cupo promedio mensual utilizado en tarjetas de crédito
- `Education` Nivel educacional (1 si no es graduado, 2 graduado y 3 si posee estudios especializadoss (magister, doctorado, etcétera)
- `Mortgage` Monto de la hipoteca (0 indica que no posee)
- `ZIP Code` Código postal del domicilio

En la última campaña a cada cliente se le ofreció un producto personalizado en base a su comportamiento financiero, preferencias, capacidad de pago y necesidades. La variable target corresponde a `Personal Loan` el cual indica si el cliente tomó o no tomó este producto (*¿El cliente aceptó o no el préstamo propuesto?*), donde 0 indica que el cliente no adquirió el producto y 1 indica que sí lo adquirió.

Es de interés analizar cuáles pudieran ser los perfiles de clientes que tienen mayor probabilidad a aceptar el producto sugerido, de manera de, identificar a los clientes con dichas características y priorizarlos a ellos en las próximas campañas.

Entrenamiento del árbol

i) Entrene un árbol de decisión. Justifique sus pasos para determinar el árbol final. Utilice como semilla `2021`. Recuerde:

- Cargar la base de datos correctamente, verifique que la información se ha leido como corresponde al importarla en R.

```
#Carga la data

library(readxl)
potencial <- read_excel(file.choose())

print(potencial)
# A tibble: 5,000 x 9
      ID   Age Income `ZIP Code` Family CCAvg Education Mortgage
   <dbl> <dbl>  <dbl>      <dbl>  <dbl> <dbl>     <dbl>    <dbl>
 1     1    25     49      91107      4   1.6         1        0
 2     2    45     34      90089      3   1.5         1        0
 3     3    39     11      94720      1   1           1        0
 4     4    35    100      94112      1   2.7         2        0
 5     5    35     45      91330      4   1           2        0

 str(potencial)  #Revisar formato de las variables
 tbl_df,     tbl     and 'data.frame':   5000 obs. of  9 variables:
$ ID            : num  1 2 3 4 5 6 7 8 9 10 ...
$ Age           : num  25 45 39 35 35 37 53 50 35 34 ...
$ Income        : num  49 34 11 100 45 29 72 22 81 180 ...
$ ZIP Code      : num  91107 90089 94720 94112 91330 ...
$ Family        : num  4 3 1 1 4 4 2 1 3 1 ...
$ CCAvg         : num  1.6 1.5 1 2.7 1 0.4 1.5 0.3 0.6 8.9 ...
$ Education     : num  1 1 1 2 2 2 2 3 2 3 ...
$ Mortgage      : num  0 0 0 0 0 155 0 0 104 0 ...
$ Personal Loan: num  0 0 0 0 0 0 0 0 0 1 ...
```

- Determinar cuáles son las variables que va a utilizar en el árbol de decisión, de ser necesario recodificar aquellas que estime conveniente (*Variables categóricas en formato numérico*) o realizar otras transformaciones.

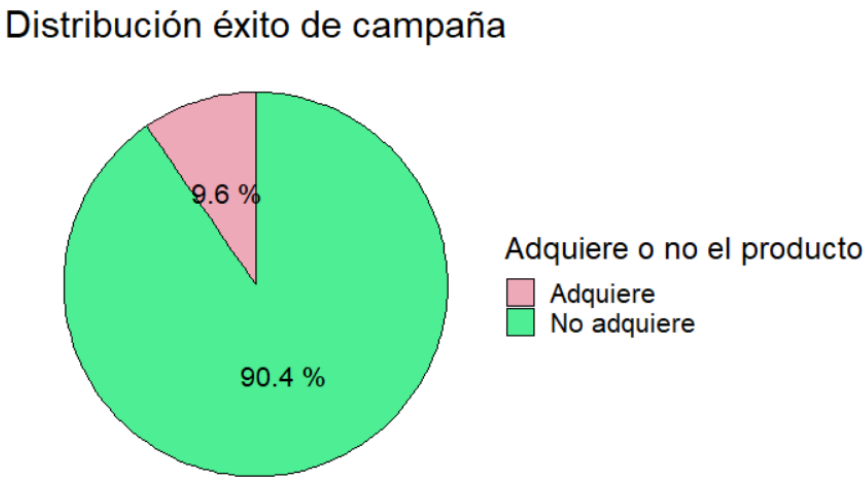
  La variable ZIP no tiene mucho sentido incluirla en el árbol, pues indica el código postal del domicilio, funciona como una especie de ID del domicilio.
  Recodificar variables como `Personal Loan` o `Education`.
  Analizar que no hay datos faltantes o registros duplicados.

```
#Las variables Education y Personal Loan deben recodificarse:

potencial$`Personal Loan`<-ifelse(potencial$`Personal Loan`=="0",
"No adquiere","Adquiere")

potencial$Education<-ifelse(potencial$Education=="1","Undergraduated",
```

```
ifelse(potencial$Education=="2","Graduated","Advanced"))

#Verificar que no hay NA:
summary(potencial)


#Verificar que no hay observaciones repetidas:

table(table(potencial$ID))   #Efectivamente no se repiten ID

   1
5000
```

- Analizar la variable de interés.

```
(table(potencial$`Personal Loan`)/nrow(potencial))*100

   Adquiere No adquiere
        9.6         90.4

df<-data.frame((table(potencial$`Personal Loan`)/nrow(potencial))*100)
df[,3]<-df[,2]*10

colnames(df)<-c("class","prop","n")

#install.packages("dplyr")

library(dplyr)

df <- df %>%
  arrange(desc(class)) %>%
  mutate(lab.ypos = cumsum(prop) - 0.5*prop)

library(ggplot2)

ggplot(df, aes(x = "", y = prop, fill = class)) +
  geom_bar(width = 2, stat = "identity", color = "black") +
  coord_polar("y", start = 0)+
  geom_text(aes(y = lab.ypos, label = paste(prop,"%")),size=5.5,
  color = "black")+
  scale_fill_manual(values = c("pink2","seagreen2")) +
  theme_void()+labs(fill="Adquiere o no el producto",
  title = "Distribucion exito de campa a")+
  theme(plot.title = element_text(hjust=0.5,size=22),
  legend.title = element_text(size=18), legend.text = element_text(size=16))

#el 9.6% adquiere el producto. Hay que determinar caracter sticas de
# estos clientes. El 90.4% no adquiere el producto
```



Analizar las variables `Age` y `Income`. Preguntarse si considerar `Zip Code` como variable, tiene sentido. Por supuesto, no utilizar el ID como variable.

```
#Income

summary(potencial$Income)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   8.00   39.00   64.00   73.77   98.00  224.00

#Los valores de Income no son muy altos

#columnas que no se utilizaran:

# - ID
# - ZIP CODE
```

- Plantee los sets de entrenamiento y testeo. Verifique que estos sets de entrenamiento son los adecuados para estudiar el desempeño del árbol posteriormente.

  La idea es observar que las categorías se distribuyan en ambos splits.

```
#Set de entrenamiento

#install.packages("caret")
library(caret)

set.seed(2021) #Semilla de aleatoriedad para el split


#split de 75% entrenamiento

index <- createDataPartition(potencial$`Personal Loan`, p = 0.75,
list = FALSE)
Train <- potencial[index,]
Test <- potencial[-index,]
```

```
table(Train$`Personal Loan`)

   Adquiere No adquiere
        360         3390

table(Test$`Personal Loan`)
   Adquiere No adquiere
        120         1130
```

Podar el árbol

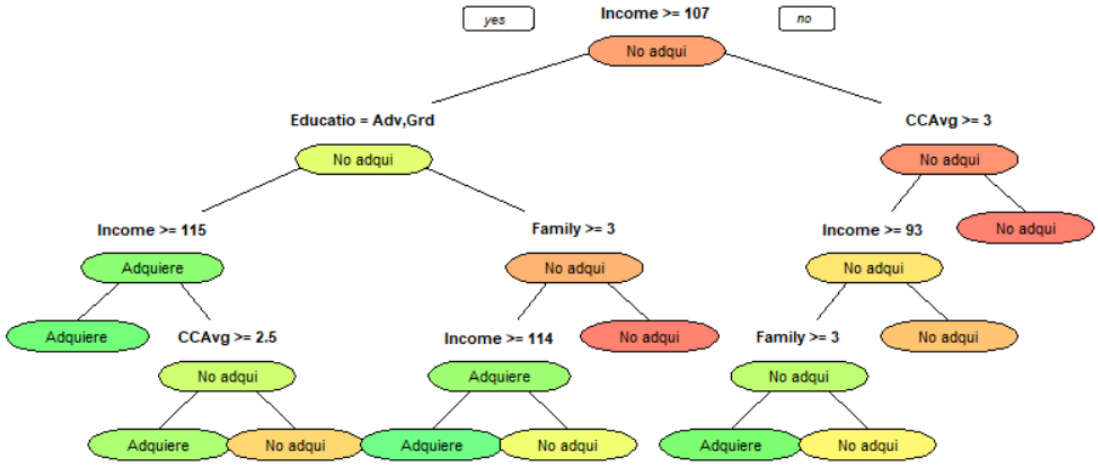
ii) Una vez realizado el árbol completo (sin podar) determine:

- ¿Cuál es la variable que más importancia tiene al momento de discernir sobre si la campaña fue efectiva (el cliente tomó el producto)?

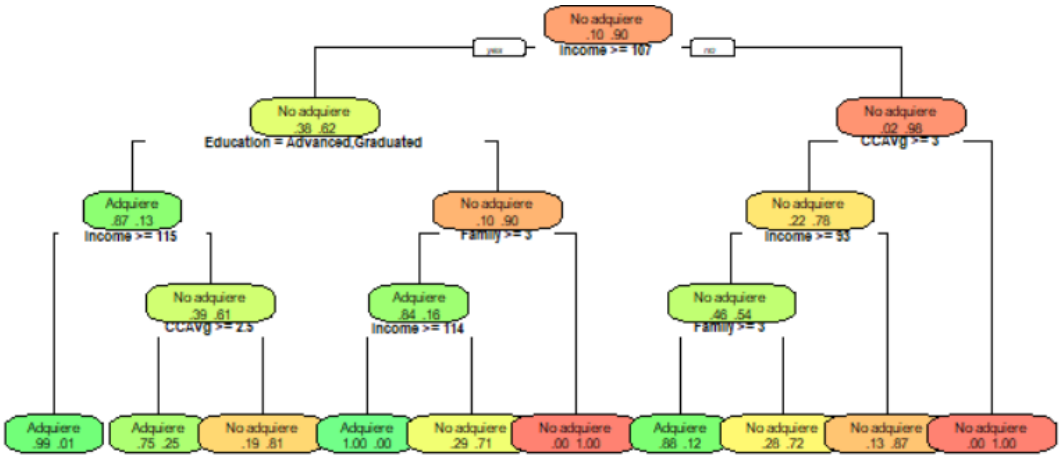
```
# Creando el arbol

#install.packages("rpart")
library(rpart)

model <- rpart(`Personal Loan` ~., data=Train[,-c(1,4)], method="class",
model=TRUE)

model$variable.importance
 Education     Income     Family      CCAvg   Mortgage        Age
229.910879 228.230119 144.210513  93.944442  19.769877   2.756952

#La variable con mas importancia calculada es Education
```

- ¿Qué obtiene si poda el árbol en términos de menor costo de complejidad asociado al mínimo `xerror`? ¿qué criterio utilizaría usted para podar el árbol?

```
#install.packages("rpart.plot")
library("rpart.plot")


#El arbol completo:
prp(model,type=1, box.palette = "-RdYlGn",legend.x=NA,cex=0.7)
```

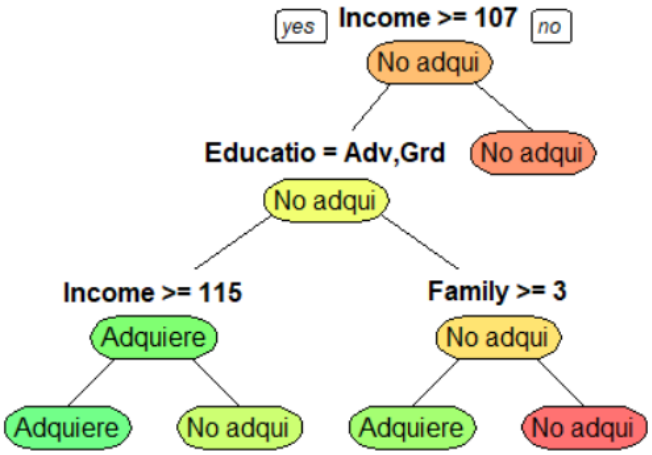



```
rpart.plot(model,extra=4,box.palette = "-RdYlGn",cex=0.5)
```




```
model$cptable
          CP nsplit rel error    xerror       xstd
1 0.31111111      0 1.0000000 1.0000000 0.05011099
2 0.11666667      2 0.3777778 0.3833333 0.03202545
3 0.03611111      3 0.2611111 0.2500000 0.02603417
4 0.02777778      4 0.2250000 0.2444444 0.02575030
5 0.01666667      5 0.1972222 0.2416667 0.02560707
6 0.01203704      6 0.1805556 0.2277778 0.02487731
7 0.01000000      9 0.1444444 0.1750000 0.02186194

#El menor xerror se tiene en el arbol mas grande
# por lo tanto, podar utilizando el cp con minimo xerror no resulta
# efectivo.

# Podria podarse en terminos de parsimonia, y fijarse
# otro cp
```

```
# Aqu  dependera del criterio de cada uno, lo importante es
# especificar

# Se puede podar utilizando otro cp, fijando un xerror tope
# (por ejemplo, de 25%)

model_dt.pruned2 <- prune(model, cp = 0.02777778) #arbol podado

prp(model_dt.pruned2,type=1,box.palette = "-RdYlGn",legend.x=NA)
```



- Evalúe el desempeño del árbol en el conjunto de datos de testeo. ¿Cuál es la tasa de clasificación correcta? Comente.

```
predpod <- unname(predict(model_dt.pruned2, Test[,-c(1,4)], type = "class"))

table(predpod==Test$`Personal Loan`)

FALSE  TRUE
   20  1230

addmargins(table(predpod,Test$`Personal Loan`),margin=1)

predpod        Adquiere No adquiere
  Adquiere          104           4
  No adquiere        16        1126
  Sum               120        1130

#Tasa de clasificacion correcta:

sum(diag(addmargins(table(predpod,Test$`Personal Loan`),margin=1)))/nrow(Test)
[1] 0.984
```

Pregunta de aplicación

iii) En base al árbol elegido, usted le recomendaría a la institución financiera invertir recursos y realizarle la campaña a un cliente con las siguientes características?:

- `Customer ID` 6001
- `Age` 54
- `Income` 114
- `Family` 3
- `CCAvg` 1.63333
- `Education` 2
- `Mortgage` 0
- `ZIP Code` 95032

No se recomienda pues en base a sus características no adquiriría el producto.