**CURSO DE FORMACIÓN EN DATA MINING CON "R"**

## TAREA 3

***Lea el archivo de datos “deudas”. Aplique el método CHAID para analizar el perfil de los clientes con impagos utilizando todas las variables como explicativas excepto la variable dependiente Impago. Es necesario transformar algunas variables previamente, ya que todas deben ser cualitativas para aplicar este método; utilice summary(deudas) para ver cuales son las 4 variables cualitativas y las 4 cuantitativas:***

***Las 4 variables cualitativas ordinales deben ser declaradas como tales (se muestra un ejemplo para la primera):***

En primer lugar, seleccionamos nuestro directorio de trabajo. En mi caso:

setwd("/Users/jlsovaz/Desktop/curso\_data\_mining/TEORIA/UNIDAD 0/")

Para poder aplicar CHAID, necesario instalar y cargar el paquete "CHAID":

library(CHAID)

Establecemos una semilla para lograr reproducibilidad en los resultados obtenidos:

set.seed(123)

Importamos el dataset del primer problema en formato .RData con la función load:

load("deudas.RData")

Realizamos summary del dataset para poder ver qué variables son cualitativas y qué variables son cuantitativas. Las cuantitativas hay que transformarlas en cualitativas estableciendo rangos.

Primero, declarar las variables ordinales y ordenarlas en función creciente, ya que R ordena los niveles de estas variables alfabéticamente:

deudas$Edad <- ordered(deudas$Edad, levels=c(‘menos de 30, entre 30 y 40, mas de 40’))

deudas$Formacion <- ordered(deudas$Formacion, levels=c(‘elemental, bachillerato, estudios universitarios’))

deudas$Empleo <- ordered(deudas$Empleo, levels=c(‘menos de 5 a\xf1os, entre 5 y 10 a\xf1os, m\xe1s de 10 a\xf1os’))

deudas$Residencia <- ordered(deudas$Residencia, levels=c(‘menos de 5 a\xf1os, entre 5 y 10 a\xf1os, m\xe1s de 10 a\xf1os’))

***b) Las 4 variables cuantitativas deben ser recodificadas y a continuación (si se forman más de dos niveles) declaradas como ordinales. Por ejemplo podemos utilizar los centiles 33 y 67 para definir sistematicamente los niveles bajo/medio/alto en todas las variables cuantitativas, como en el ejemplo (con la variable Ingreso; solo es necesario cambiar el nombre de la variable las 6 veces que aparece para aplicarlo a las restantes):***

Ahora, convertimos las variables cuantitativas en cualitativas y las ordenamos de la misma forma que las 4 anteriores. Recodificaremos las variables utilizando los centiles 33 y 67:

q1 <- quantile(deudas$Ingreso,0.33)

q3 <- quantile(deudas$Ingreso,0.67)

deudas$Ingreso <- recode(deudas$Ingreso, 'lo:q1="bajo"; q1:q3="medio"; q3:hi="alto" ', as.factor=TRUE)

deudas$Ingreso <- ordered(deudas$Ingreso, levels=c('bajo' ,  'medio', 'alto'))

Hacemos lo mismo con las otras 3:

q1 <- quantile(deudas$Deud\_ing,0.33)

q3 <- quantile(deudas$Deud\_ing,0.67)

deudas$Deud\_ing <- recode(deudas$Deud\_ing, 'lo:q1="bajo"; q1:q3="medio"; q3:hi="alto" ', as.factor=TRUE)

deudas$Deud\_ing <- ordered(deudas$Deud\_ing, levels=c('bajo' ,  'medio', 'alto'))

q1 <- quantile(deudas$Deud\_tarj,0.33)

q3 <- quantile(deudas$Deud\_tarj,0.67)

deudas$Deud\_tarj <- recode(deudas$Deud\_tarj, 'lo:q1="bajo"; q1:q3="medio"; q3:hi="alto" ', as.factor=TRUE)

deudas$Deud\_tarj <- ordered(deudas$Deud\_tarj, levels=c('bajo' ,  'medio', 'alto'))

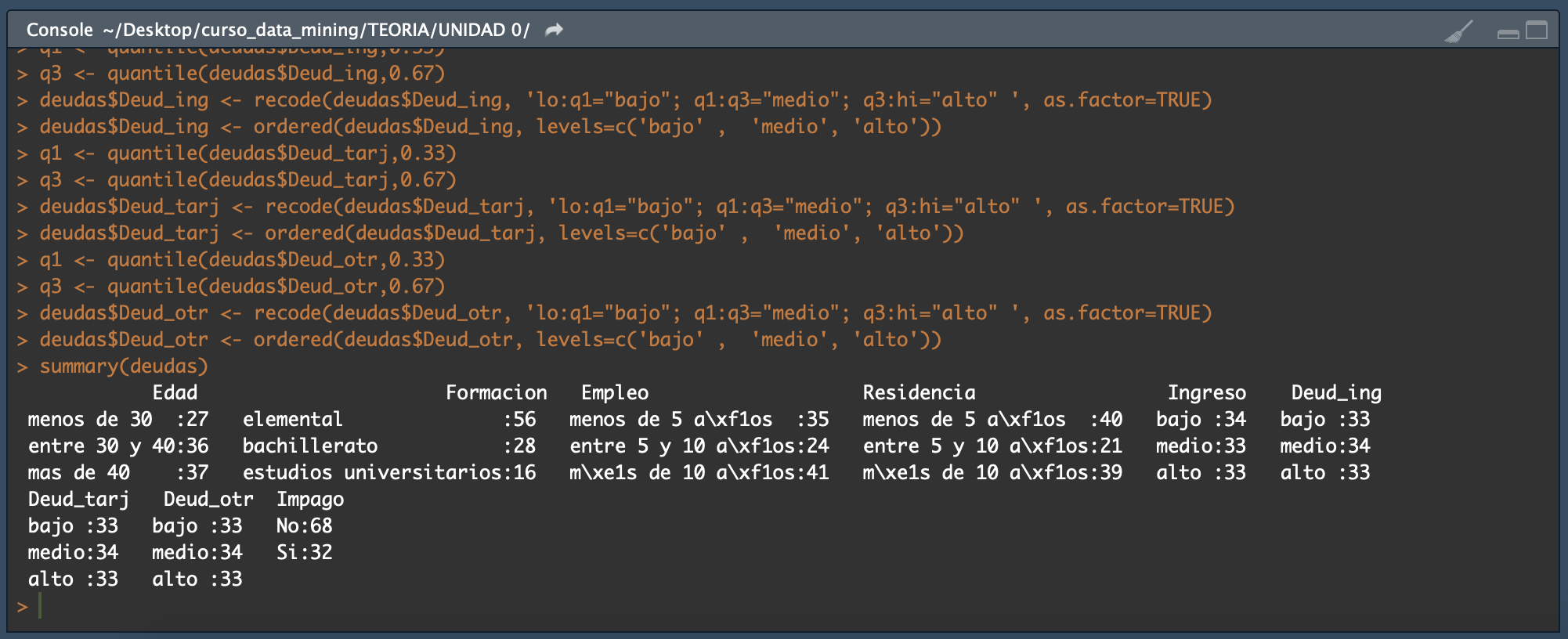
q1 <- quantile(deudas$Deud\_otr,0.33)

q3 <- quantile(deudas$Deud\_otr,0.67)

deudas$Deud\_otr <- recode(deudas$Deud\_otr, 'lo:q1="bajo"; q1:q3="medio"; q3:hi="alto" ', as.factor=TRUE)

deudas$Deud\_otr <- ordered(deudas$Deud\_otr, levels=c('bajo' ,  'medio', 'alto'))

Hacemos summary(deudas) y vemos que todas las variables están correctamente transformadas:



Aplicamos el método CHAID, configuramos algunos parámetros antes:

chaid\_control(alpha2 = 0.05, alpha3 = -1, alpha4 = 0.05,minsplit = 20, minbucket = 7,

minprob = 0.01,stump = FALSE, maxheight = -1)

Aplicamos el método:

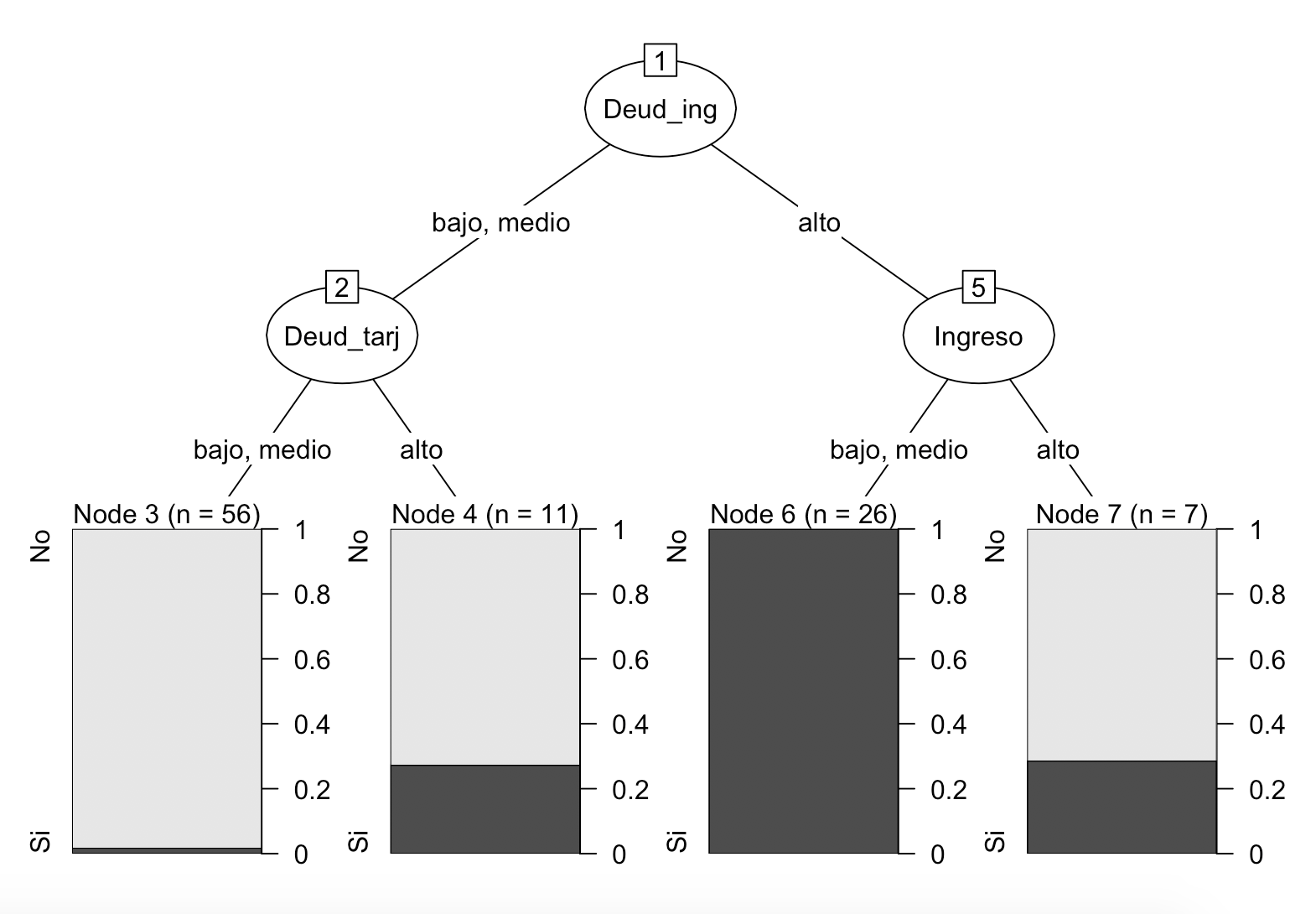
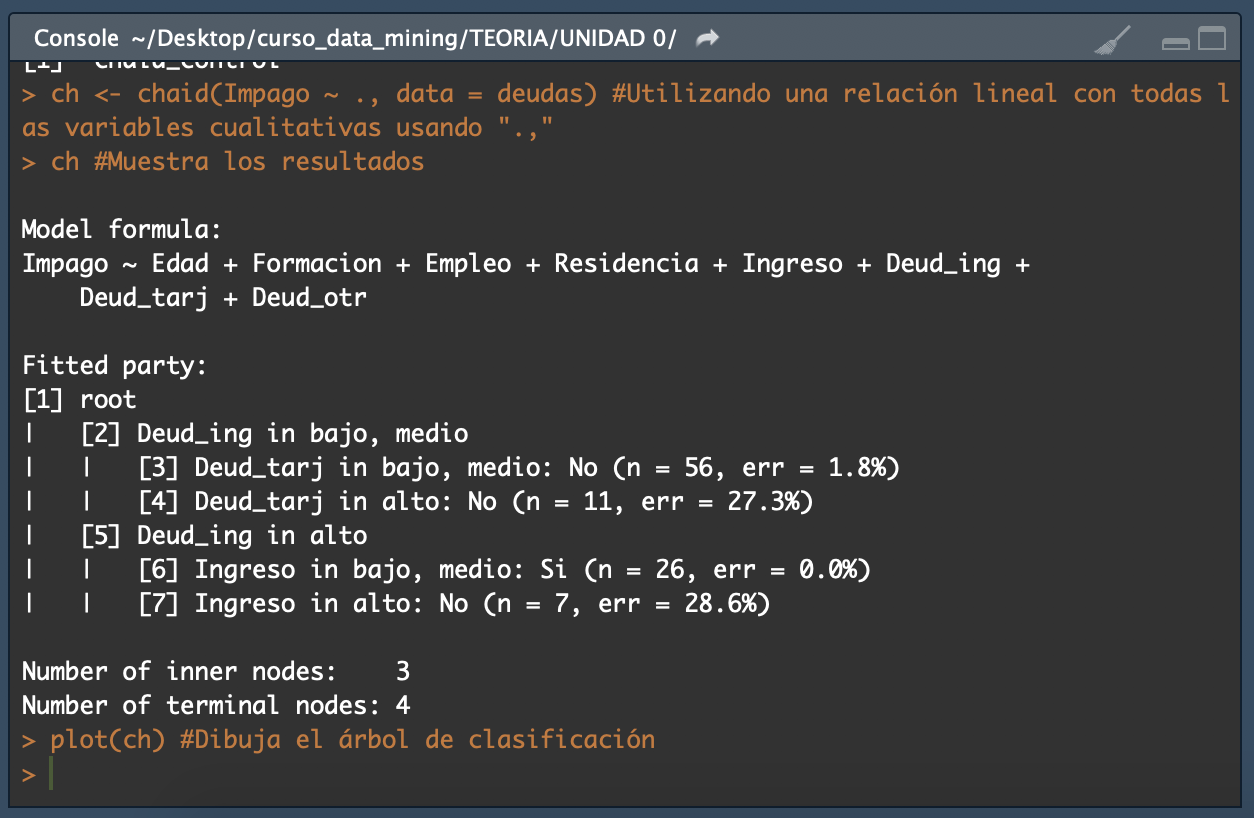
ch <- chaid(comprador ~ ., data = compra) #Utilizando una relación lineal con todas las variables cualitativas usando ".,"

Muestra los resultados:

Ch

Dibuja el árbol de clasificación

plot(ch)



Vemos que la variable explicativa más significativa para clasificar a las personas en relación con la variable "Impago" es la Deuda por ingresos. Empieza dividiendo entonces por Deuda ingreso. Luego se emplean las siguientes variables para seguir realizando separaciones de grupo.

Analizando el árbol creado, podemos ver que la totalidad de las personas que presentan deudas por ingreso altas y deudas por tarjeta medias o bajas, son clientes que presentan impago. El ~30% de clientes con deudas por ingreso altas y deudas por tarjeta altas presentan Impago.

Por otro lado, el 30% de los clientes con deudas por ingreso bajas o medias y deudas por tarjeta altas presentan Impago. Por último, la mayor parte de los clientes que presentan bajas o medias deudas por ingreso y bajas o medias deudas por tarjeta, no presentan Impago (como es de esperar).

La variable otras deudas no se ha utilizado, no parece ser importante para establecer los grupos y para la creación del árbol.

***Lea el conjunto de datos "algas", que tiene 31 casos y 20 variables. Aplique el método Naive Bayes para identificar el tipo de alga (variable “clase”) utilizando las restantes variables (concentraciones de pigmentos) como variables explicativas. Utilice una muestra aleatoria de 20 casos como conjunto de entrenamiento y el resto para validación. Construya la matriz de confusión (tabla de predicción y clase original) para el conjunto de prueba y para todo el conjunto "algas".***

En primer lugar, seleccionamos nuestro directorio de trabajo. En mi caso:

setwd("/Users/jlsovaz/Desktop/curso\_data\_mining/TEORIA/UNIDAD 1/")

Para poder aplicar CHAID, necesario instalar y cargar el paquete "CHAID":

library(e1071)

Establecemos una semilla para lograr reproducibilidad en los resultados obtenidos:

set.seed(12345)

Importamos el dataset del primer problema en formato .RData con la función load:

load("algas.RData")

A partir de este dataset, creamos nuestras dos muestras: la de entrenamiento y la de validación:

muestra <- sample(1:nrow(algas), 20)

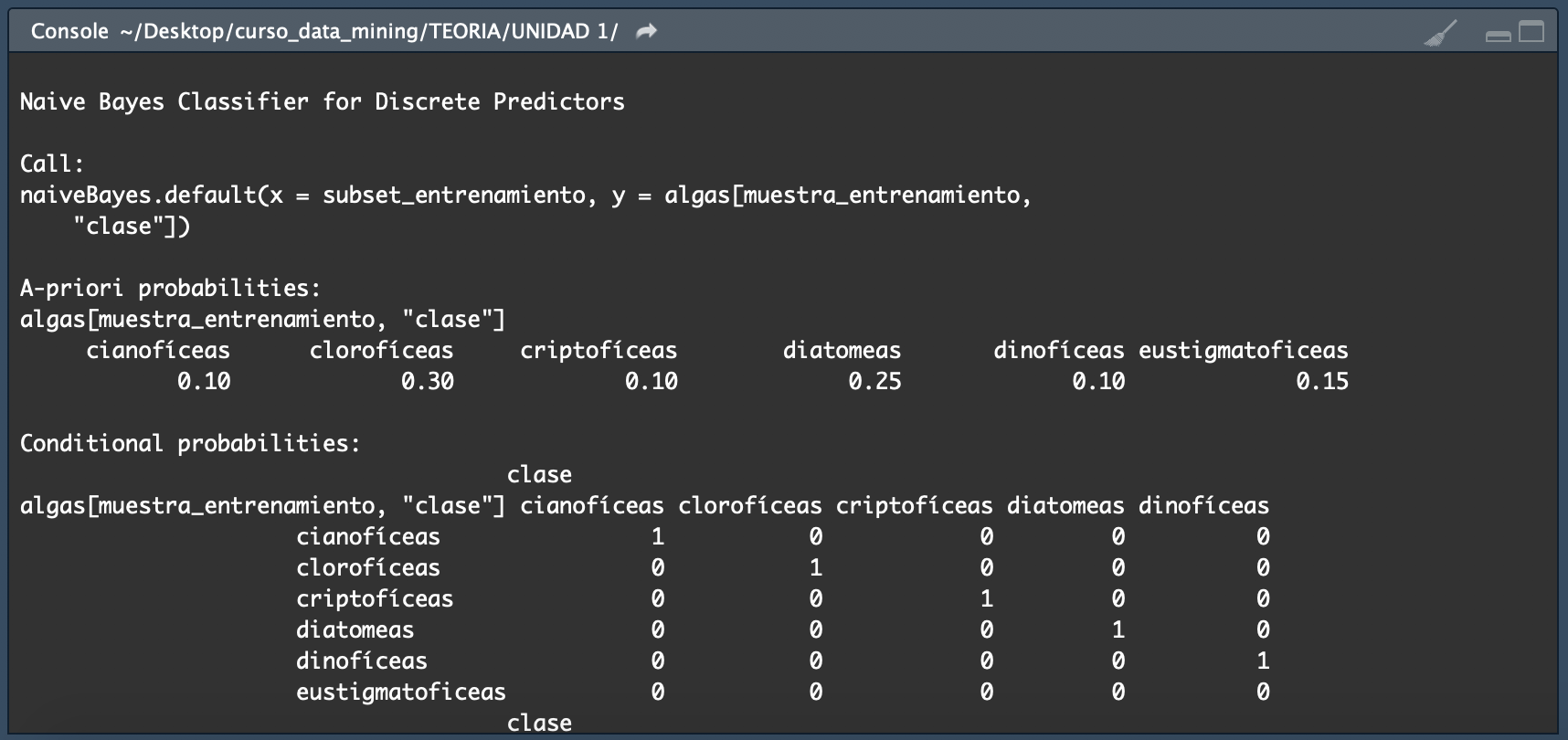
entrenamiento <- algas[muestra,]

prueba <- algas[-muestra,]

Aplicamos el método:

modelo <- naiveBayes(x = entrenamiento, y = algas[muestra,"clase"])

modelo



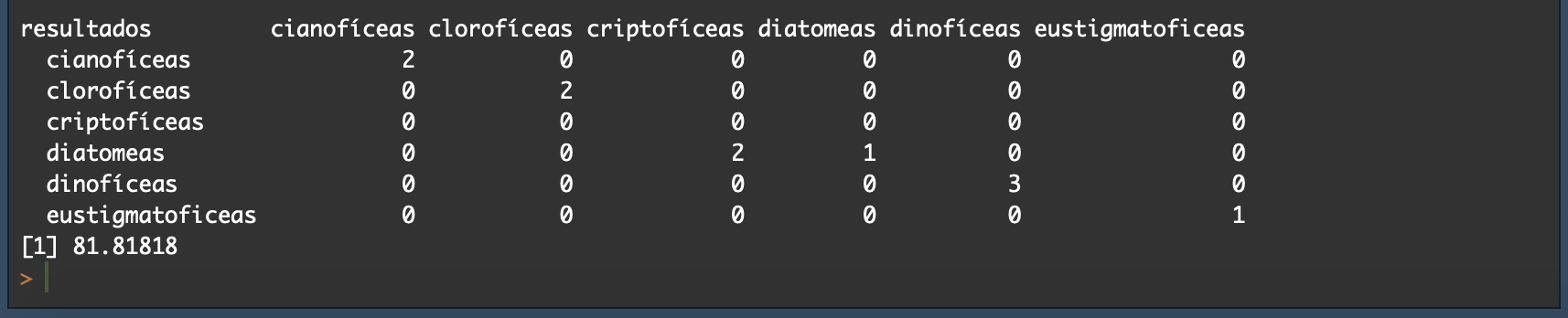
Podemos ver la proporción de algas de cada clase. Por ejemplo, el 30% de las algas son clorofíceas

Establecemos la predicción, con los datos del conjunto de validación:

resultados <- predict(object = modelo, newdata = prueba, type = "class")

t <- table(resultados, algas[-muestra,"clase"])

t ; 100 \* sum(diag(t)) / sum(t)



Podemos ver que el % de acierto en la predicción es del 81.81%

Mejoramos entonces el modelo aplicando el método a todo el dataset:

modelo <- naiveBayes(x = algas, y = algas[,"clase"])

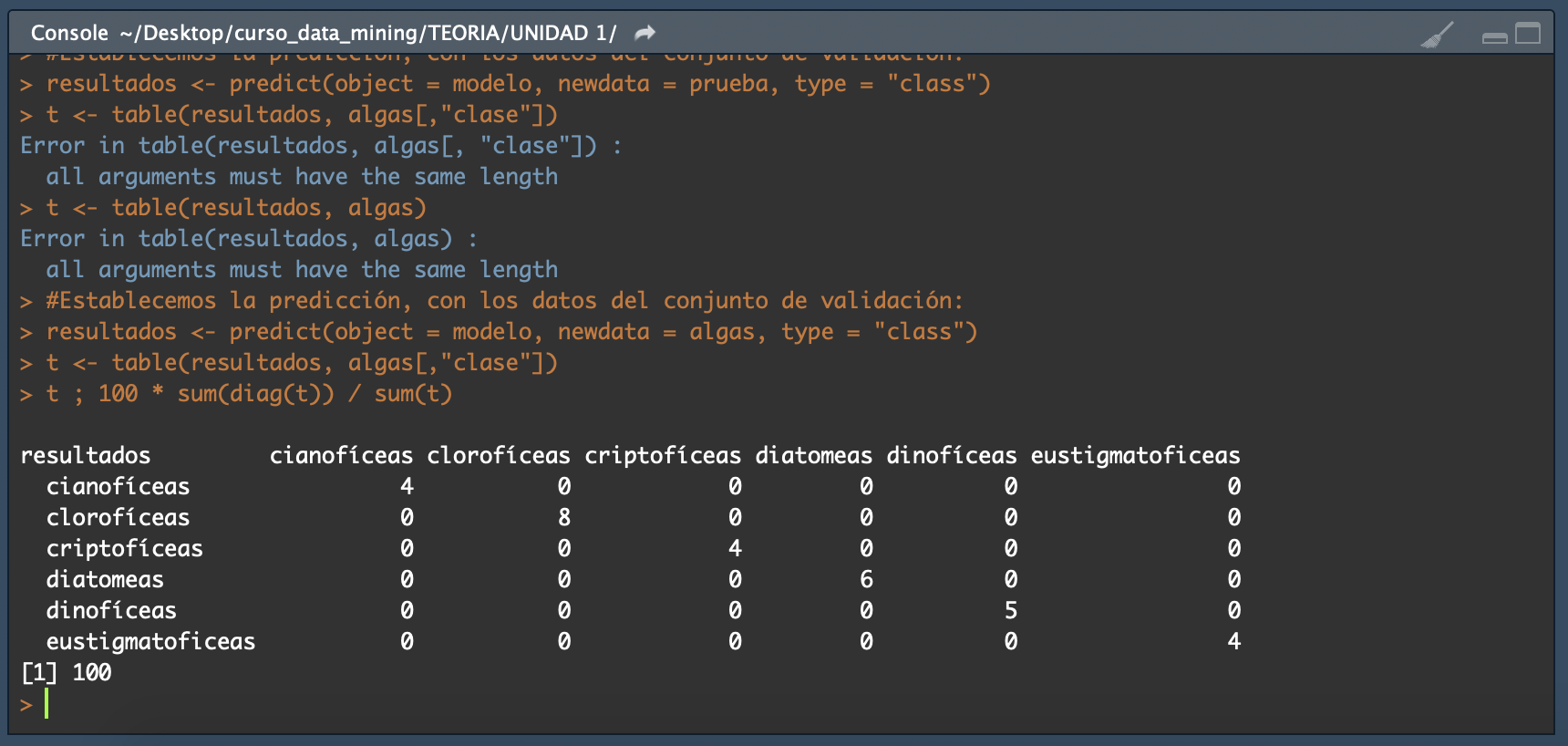
modelo

Establecemos la predicción::

resultados <- predict(object = modelo, newdata = algas, type = "class")

t <- table(resultados, algas[,"clase"])

t ; 100 \* sum(diag(t)) / sum(t)



Vemos que el modelo mejora, ya que esta vez es capaz de clasificar correctamente todas las algas en la clase correcta con un 100% de acierto.

## TAREA 4

***Lea el archivo de datos “vinos”. Aplique el método CART para identificar la variedad de uva utilizada en la elaboración del vino a partir de las concentraciones de ácidos orgánicos determinadas analíticamente. Utilice una muestra de entrenamiento de 40 casos aleatorios y los 16 restantes como muestra de validación.***

***{ Nota: sustituya en las órdenes del ejemplo de la unidad***

***“deudas” por “vinos”, e “Impago” por “var” (en el punto 2 de la tarea debe hacerse la transformación inversa) }***

En primer lugar, seleccionamos nuestro directorio de trabajo. En mi caso:

setwd("/Users/jlsovaz/Desktop/curso\_data\_mining/TEORIA/UNIDAD 1/")

Establecemos una semilla para lograr reproducibilidad en los resultados obtenidos:

set.seed(12345)

Importamos el dataset del primer problema en formato .RData con la función load:

load("vinos.RData")

Construimos la muestra que servirá de entrenamiento con 40 casos y la de validación con los restantes:

muestra <- sample(1:nrow(vinos), 40)

entrenamiento <- vinos[muestra, ]

prueba <- vinos[-muestra, ]

Cargamos el paquete necesario para aplicar el método CART:

library(rpart)

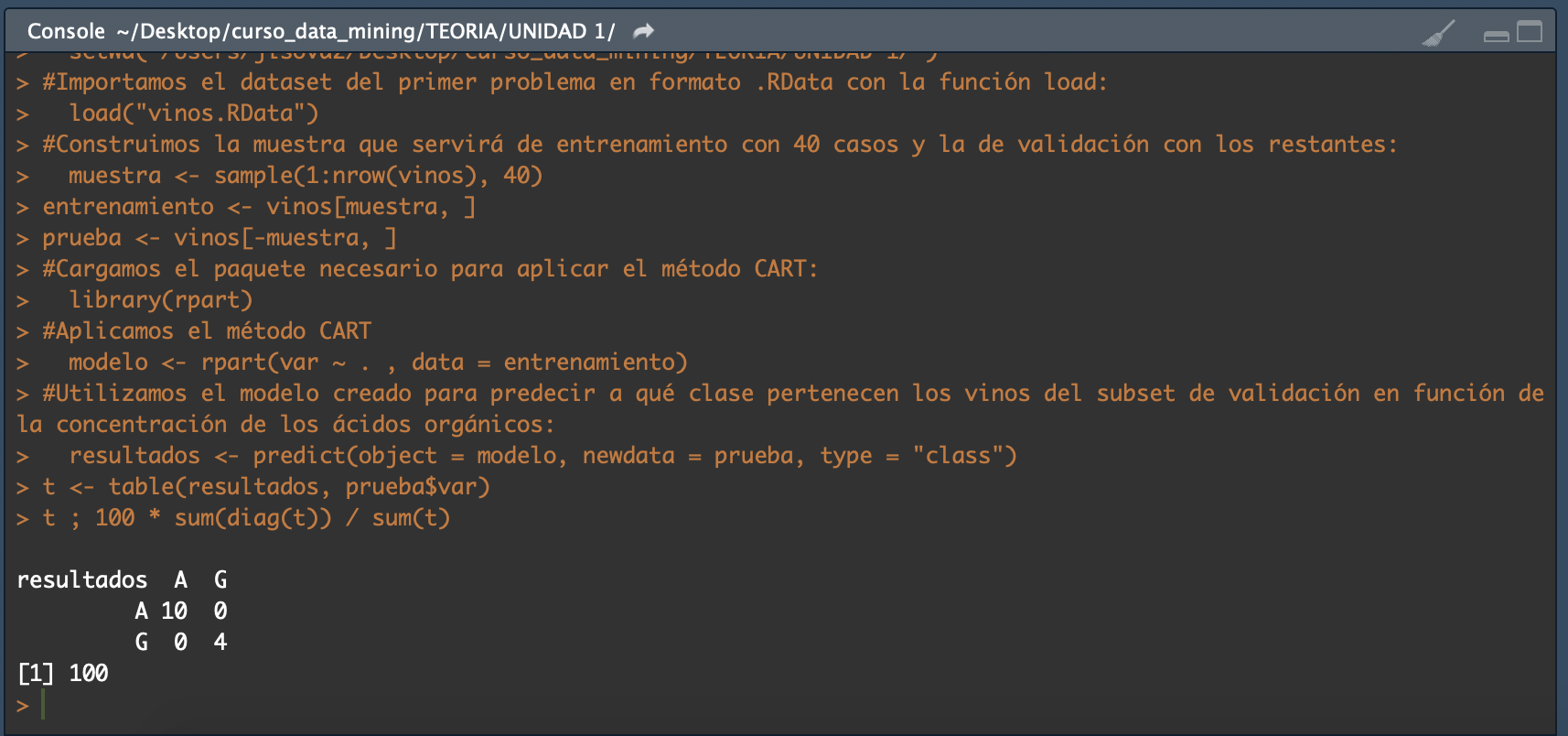
Aplicamos el método CART

modelo <- rpart(var ~ . , data = entrenamiento)

Utilizamos el modelo creado para predecir a qué clase pertenecen los vinos del subset de validación en función de la concentración de los ácidos orgánicos:

resultados <- predict(object = modelo, newdata = prueba, type = "class")

t <- table(resultados, prueba$var)

 t ; 100 \* sum(diag(t)) / sum(t)

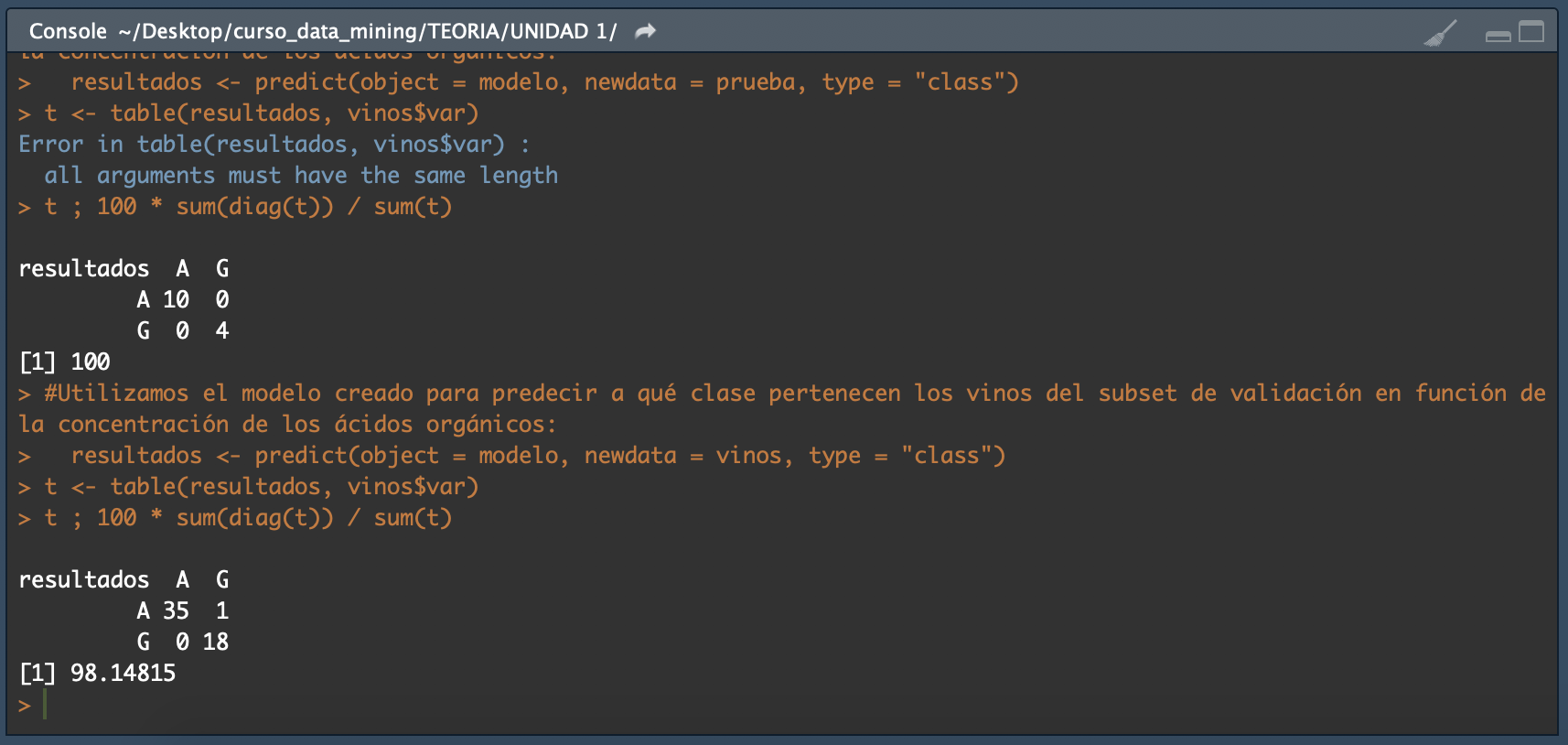
Como funciona muy bien la predicción, podemos repetir el modelo utilizando esta vez como muestra el dataset entero:

modelo <- rpart(var ~ . , data = vinos)

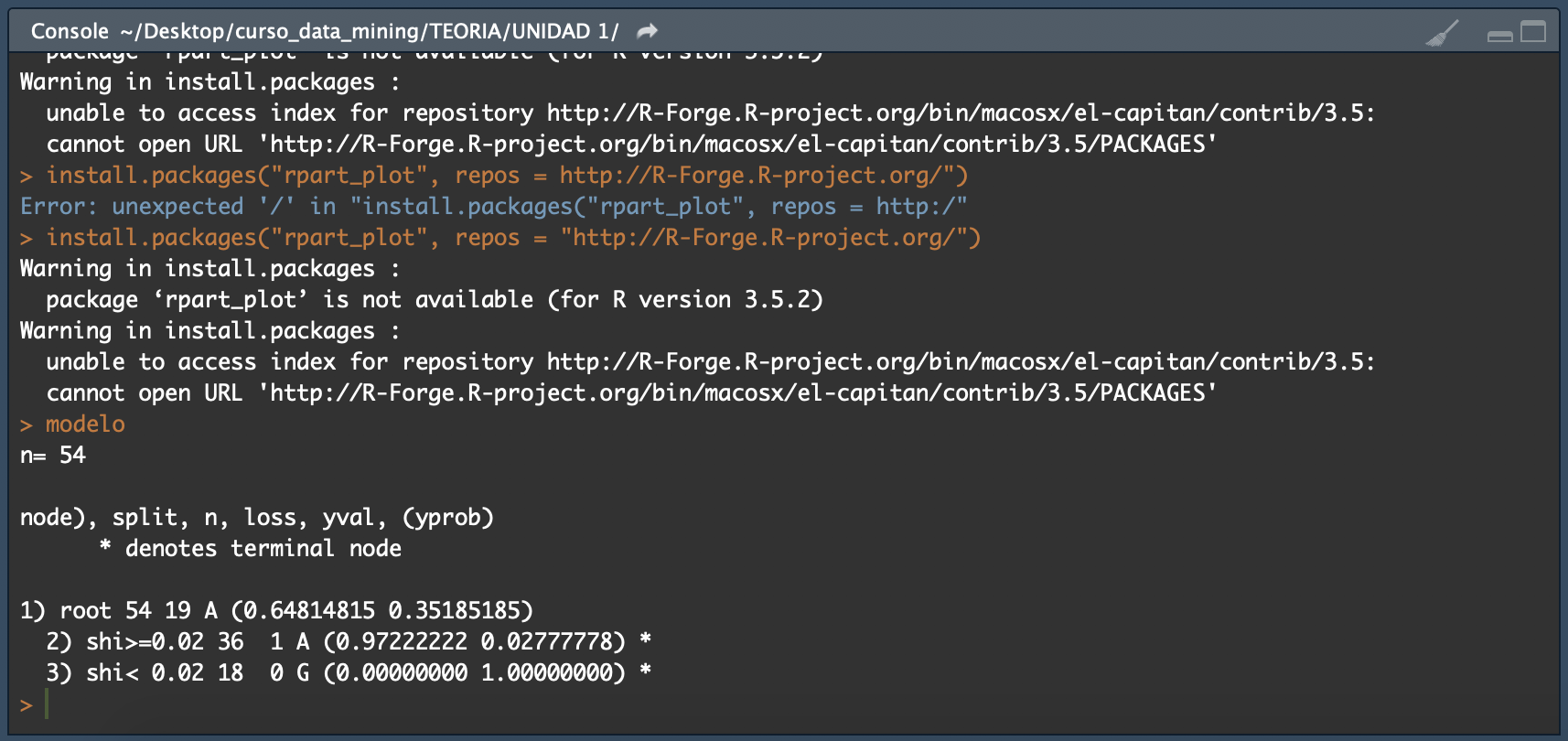
resultados <- predict(object = modelo, newdata = vinos, type = "class")

t <- table(resultados, vinos$var)

t ; 100 \* sum(diag(t)) / sum(t)

******

Vemos que el modelo no funciona correctamente en todos los casos. Es capaz de predecir de forma correcta todos los vinos excepto uno.

******Si vemos el modelo, podemos ver que de 54 vinos, 19 pertenecen a la clase A (Albariño, un 35%).

En el segundo nodo se utiliza la variable shi con el corte mayor o igual que 0.02 o menor que 0.02 .En ese punto, de 37 vinos, solamente 1 es Albariño; el resto son Godello.

***Lea el conjunto de datos "deudas". Aplique el método C5.0 para identificar a los clientes que tendrán impagos a partir de sus características conocidas. Utilice una muestra de entrenamiento de 80 casos aleatorios y los 20 restantes como muestra de validación.  Represente gráficamente el árbol con la regla de clasificación. Construya la matriz de confusión para la muestra de entrenamiento y para la de prueba***.

En primer lugar, seleccionamos nuestro directorio de trabajo. En mi caso:

setwd("/Users/jlsovaz/Desktop/curso\_data\_mining/TEORIA/UNIDAD 0/")

Establecemos una semilla para lograr reproducibilidad en los resultados obtenidos:

set.seed(123)

Importamos el dataset del primer problema en formato .RData con la función load:

load("deudas.RData")

Construimos la muestra que servirá de entrenamiento con 40 casos y la de validación con los restantes:

muestra <- sample(1:nrow(deudas), 80)

entrenamiento <- deudas[muestra, ]

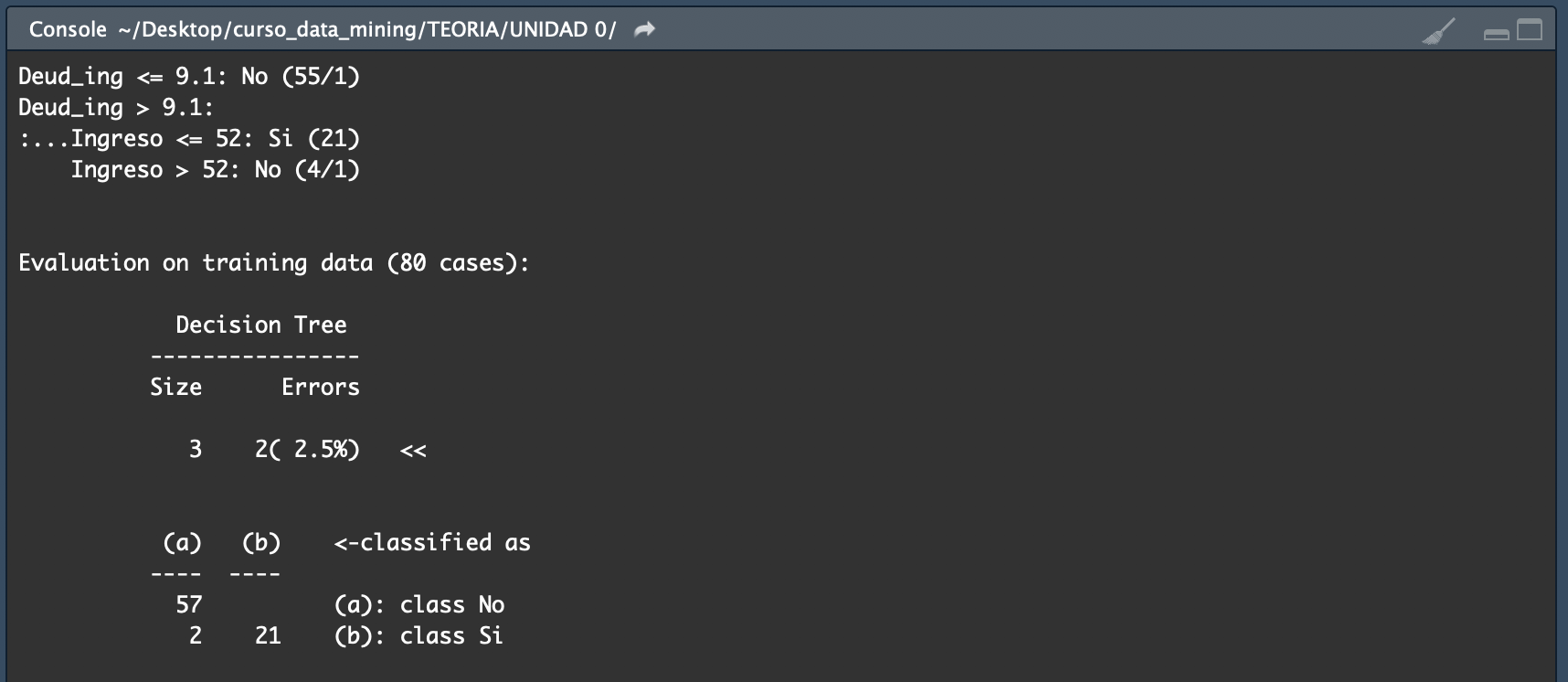
prueba <- deudas[-muestra, ]

Cargamos el paquete necesario para aplicar el método C5.0:

library(C50)

Aplicamos el método C5.0

modelo <- C5.0(Impago ~ Ingreso+Deud\_ing+Deud\_tarj+Deud\_otr, data = entrenamiento)

Exploramos el modelo construido con summary(modelo):

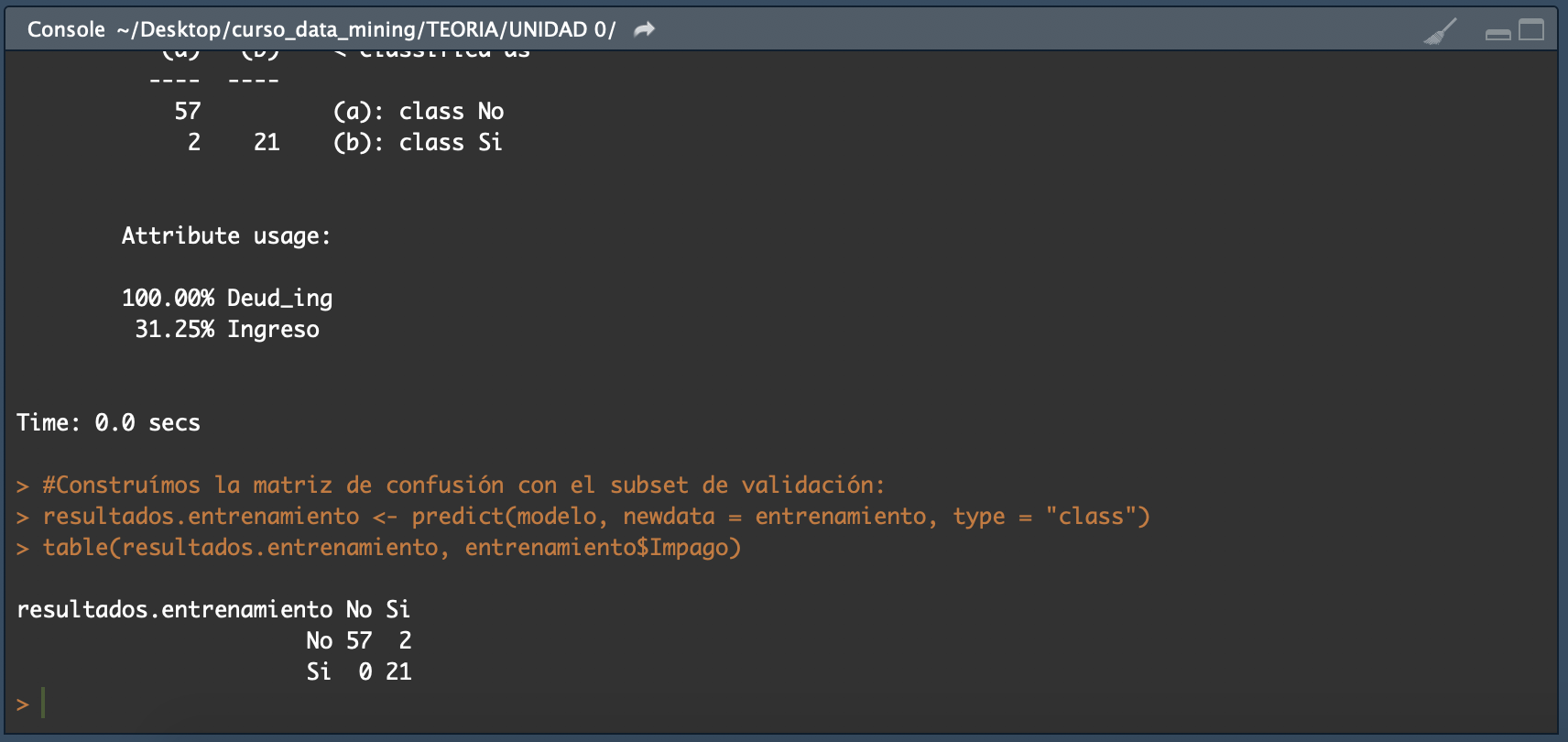
El modelo tiene en cuenta dos variables. En primer lugar la variable deuda ingreso con el punto de corte 9.1. Los clientes con punto de corte menor o igual a 9.1, no presentan impago excepto 1.

Se utiliza una segunda variable para los clientes con deuda ingreso mayor que 9.1. La segunda variable es ingreso con punto de corte 52.

Construímos la matriz de confusión con el subset de validación:

resultados.entrenamiento <- predict(modelo, newdata = entrenamiento, type = "class")

table(resultados.entrenamiento, entrenamiento$Impago)



Vemos que solamente hay dos clientes que fueron clasificados incorrectamente.

Ya que funcionó bastante bien, podemos realizar el modelo con toda la muestra inicial:

modelo <- C5.0(Impago ~ Ingreso+Deud\_ing+Deud\_tarj+Deud\_otr, data = deudas)

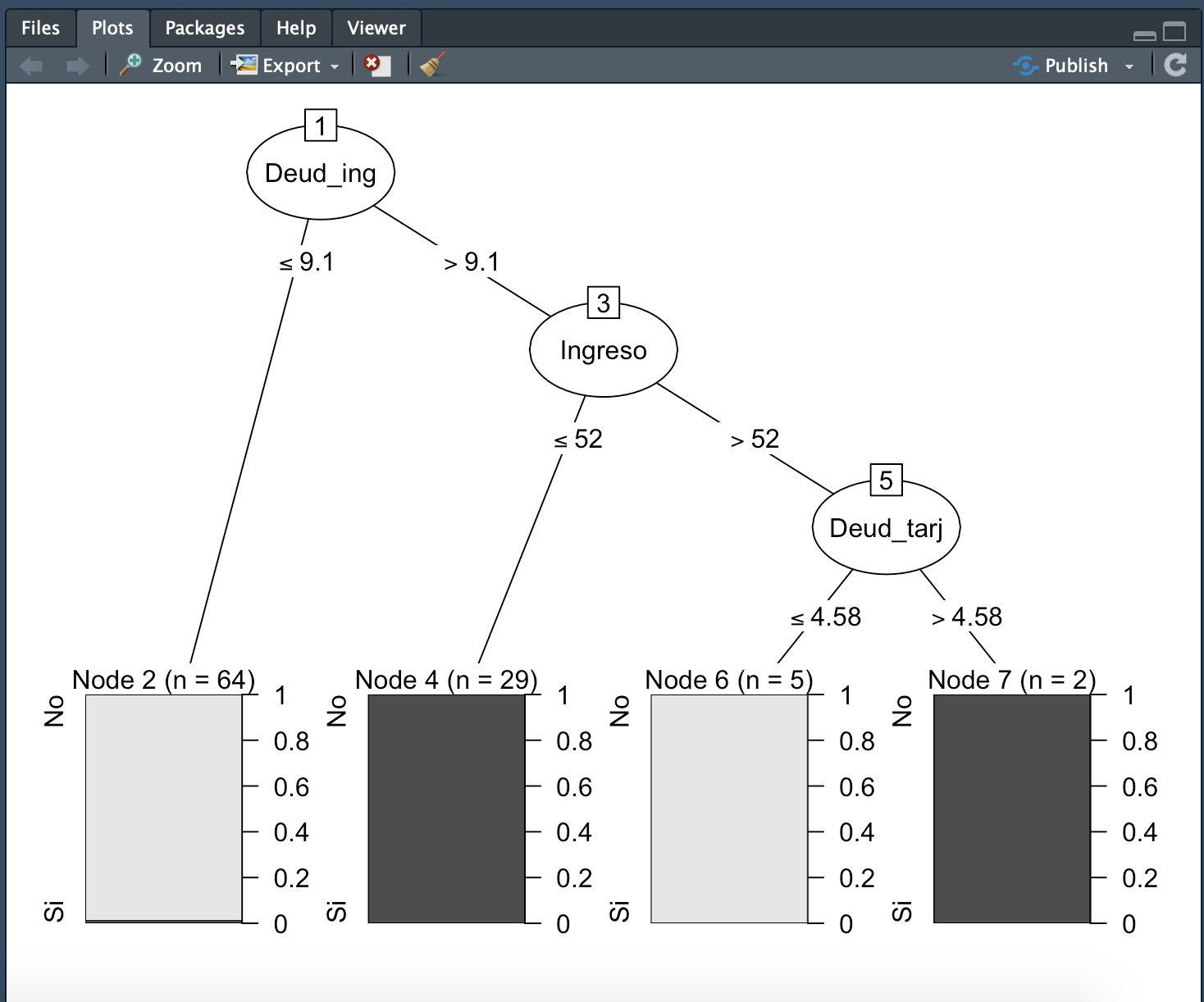
resultados.entrenamiento <- predict(modelo, newdata = deudas, type = "class")

table(resultados.entrenamiento, deudas$Impago)

El modelo parece que mejora: en este caso solamente un cliente es clasificado incorrectamente

## 

El árbol se puede representar mediante:

plot(modelo)