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

## TAREA 1

***Lea el archivo de datos “vinos.RData”. Aplique el método k-medias para obtener dos clases utilizando como variables explicativas las concentraciones de los distintos ácidos orgánicos. Relacione la clasificación obtenida con la variedad de uva, con el fin de averiguar hasta que punto esa variedad es un criterio principal de la clasificación “natural” de los vinos o es necesario buscar criterios adicionales (como la zona, el año de la vendimia, u otros).***

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(123)

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

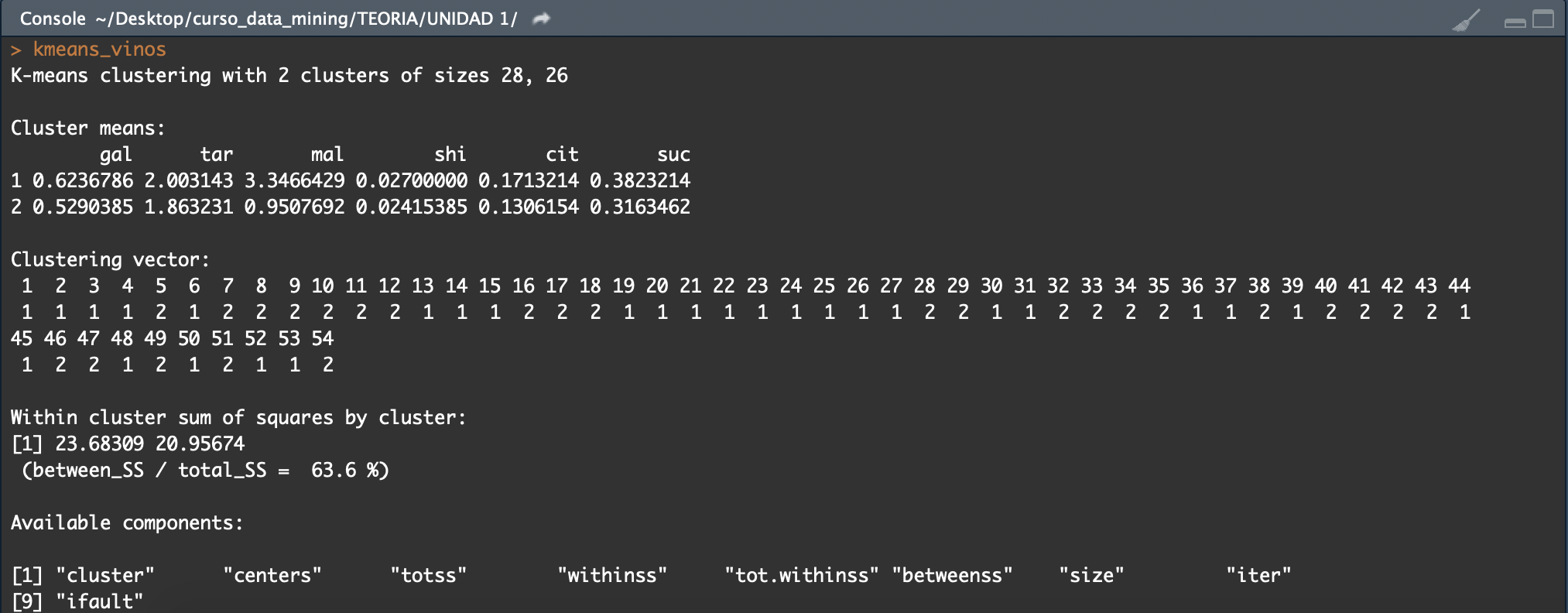
load("vinos.RData")

Realizar método k-means en el dataframe "vinos" utilizando dos clases para clasificar cada muestra en función de la información que nos dan sobre las concentraciones de los distintos ácidos orgánicos. Elegimos dos clases porque es la clasificación "natural" de nuestras muestras en este caso: Albariño y Godello. Así veremos si las variables cuantitativas son un buen criterio para clasificar los vinos en las dos clases mencionadas.

kmeans\_vinos <- kmeans(vinos[,2:ncol(vinos)], centers = 2)

Exploramos el modelo calculado:

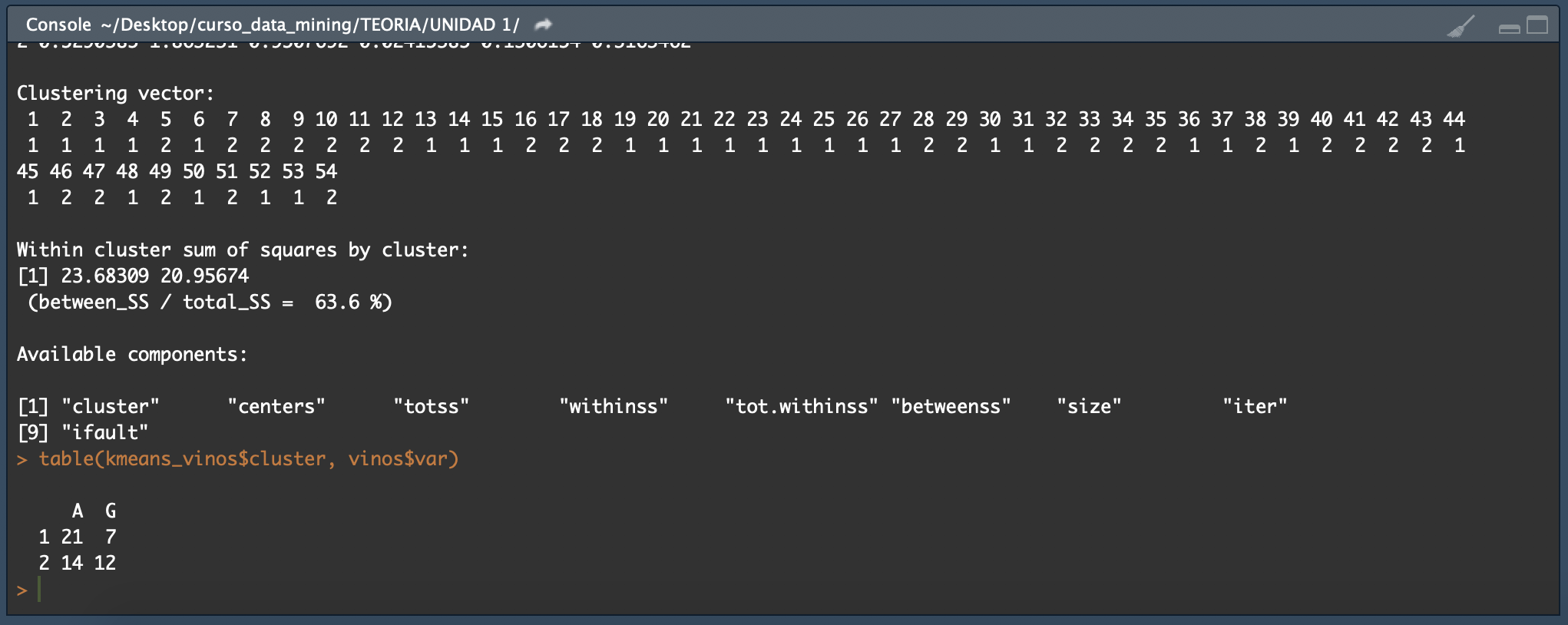
kmeans\_vinos

Aparece la siguiente salida:

Vemos que este método clasifica 28 vinos en la clase 1 y 26 vinos en la clase 2. Tenemos las medias (centros) para cada ácido orgánico en cada una de las clases. A simple vista podemos ver que algunos ácidos orgánicos (*gal, shi, cit, suc*) presentan medias parecidas para las dos clases; mientras que *tar* y *mal* tienen medias muy distintas entre las dos clases.

En este caso, el % de variabilidad explicada por esta clasificación es del 63.6%, lo que podría indicar que no es una buena clasificación.

De hecho, si escribimos:

table(kmeans\_vinos$cluster, vinos$var)

Podemos ver que 21 Albariños son clasificados en una clase y los otros 14 son clasificados en la clase 2. En relación al Godello, 7 son designados a la clase 1; 12 vinos a la clase 2. Parece ser que la concentración de ácidos orgánicos no es un buen criterio para clasificar los vinos en Albariño y Godello. Podrían ser necesarias otras variables cuantitativas para poder discernir mejor entre la clasificación natural de los vinos.

***Lea el conjunto de datos "deudas.RData". Se trata de una muestra de 100 clientes de un banco, algunos de los cuales han presentado impagos, de los que se dispone de información relativa a su nivel de ingresos, relación entre deudas e ingresos, importe de las deudas por tarjeta de crédito, e importe de otras deudas, entre otras variables. Utilice esas 4 variables (columnas 5 a 8) para obtener con el método EM una clasificación con dos grupos o clases.  Averigue si la clasificación obtenida está relacionada con la variable “Impago”.***

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")

Realizamos el método EM en el dataframe "deudas" utilizando dos clases para clasificar cada muestra en función de la información que nos dan sobre ingresos, deudas\_ingresos, deudas\_tarjeta y otras deudas. Así veremos si las variables cuantitativas son un buen criterio de clasificación de los clientes en dos categorías: presentan o no presentan impagos.

Cargamos el paquete necesario para realizar el método EM:

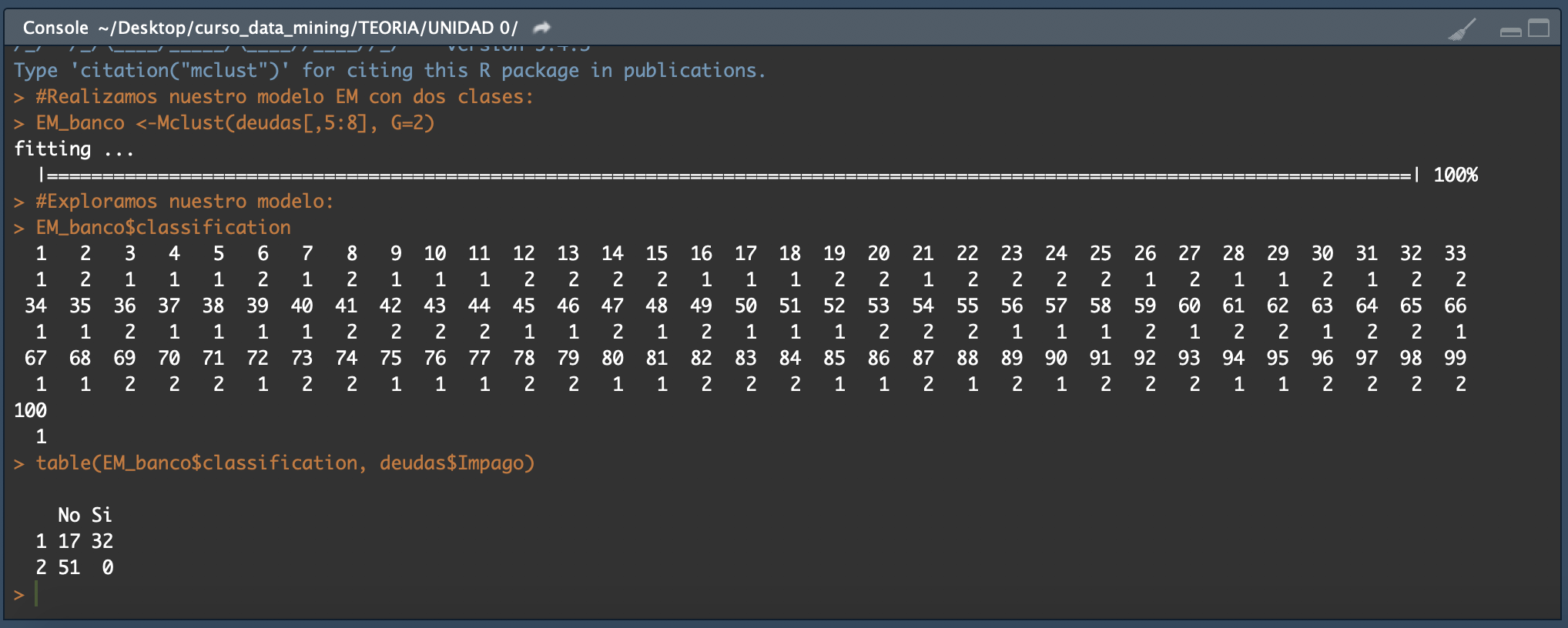
library(mclust)

Realizamos nuestro modelo EM con dos clases:

EM\_banco <-Mclust(deudas[,5:8], G=2)

Exploramos nuestro modelo y vemos como clasificación se relaciona con la variable Impago:

EM\_banco$classification

table(EM\_banco$classification, deudas$Impago)

En este caso, la clasificación es buena, 17 de las personas que no presentan impago fueron a la clase 1, mientras que 51 fueron a la clase 2. Por otro lado, la totalidad de las personas que presentan impago fueron clasificadas en una sóla clase. Se puede concluir que las variables cuantitativas usadas clasifican (al menos) perfectamente a las personas que presentan impago.

## TAREA 2

***Lea el conjunto de datos “empresas” del archivo “empresas.RData”, con una muestra de 1194 empresas, de las cuales 165 son empresas culturales, y 25 variables: la que identifica su caracter de empresa cultural y 24 adicionales obtenidas de su balance y cuenta de resultados. Aplique el método Adaboost, utilizando una muestra de entrenamiento de 1000 casos aleatorios y los restantes 194 como muestra de validación, para conocer si es posible predecir o identificar el carácter de empresa cultural (variable “CULTURAL”) a partir de las variables contables (las 24 restantes). Valide los resultados con la muestra restante. Identifique las tres variables explicativas más relevantes.***

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

setwd("/Users/jlsovaz/Desktop/curso\_data\_mining/TEORIA/UNIDAD 2/")

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("empresas.RData")

Cargamos la librería necesaria para aplicar adaboost:

library(adabag)

Creamos una muestra de entrenamiento (1000 casos seleccionados al azar) y una muestra de validación (los 194 casos restantes).

muestra\_entrenamiento <- sample(1:nrow(empresas), 1000)

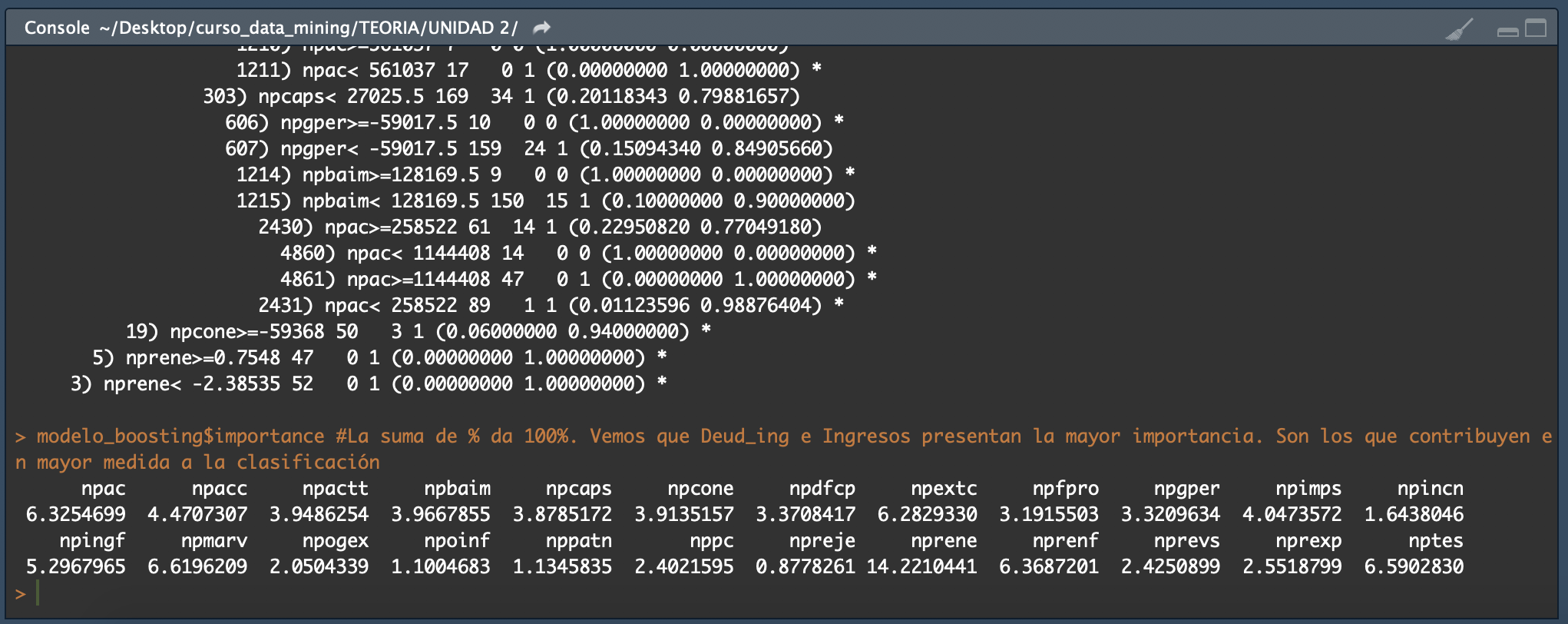
subset\_entrenamiento <- empresas[muestra\_entrenamiento,]

subset\_validacion <- empresas[-muestra\_entrenamiento,]

Realizamos un modelo de relación lineal entre la variable CULTURAL y las 24 variables restantes:

modelo\_boosting <- boosting(CULTURAL ~ ., data = subset\_entrenamiento)

Exploramos el modelo creado en relación a la importancia de cada variable:

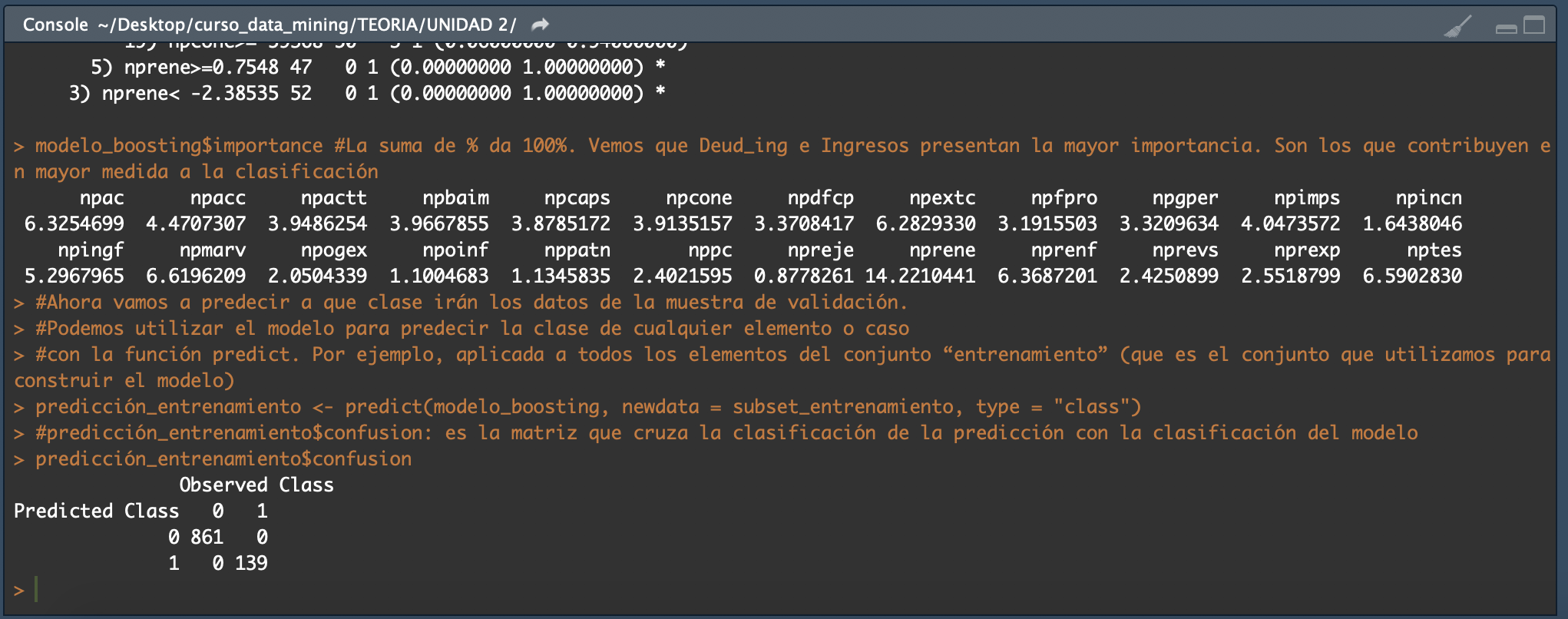
modelo\_boosting$importance

Vemos que no hay una variable que prediga con gran criterio en qué clase se clasifica cada una de las muestras. Únicamente, nprene tiene la mayor importancia (14.2210441). El resto de las variables presentan una importancia baja pero similar entre ellas. Las otras dos variables más importantes son: npmarv (6.6196) y nptes (6.59).

Ahora aplicamos este modelo sobre las mismas muestras que se utilizaron para crearlo, como si quisiéramos predecir la clase de las 1000 muestras de entrenamiento.

predicción\_entrenamiento <- predict(modelo\_boosting, newdata = subset\_entrenamiento, type = "class")

predicción\_entrenamiento$confusion

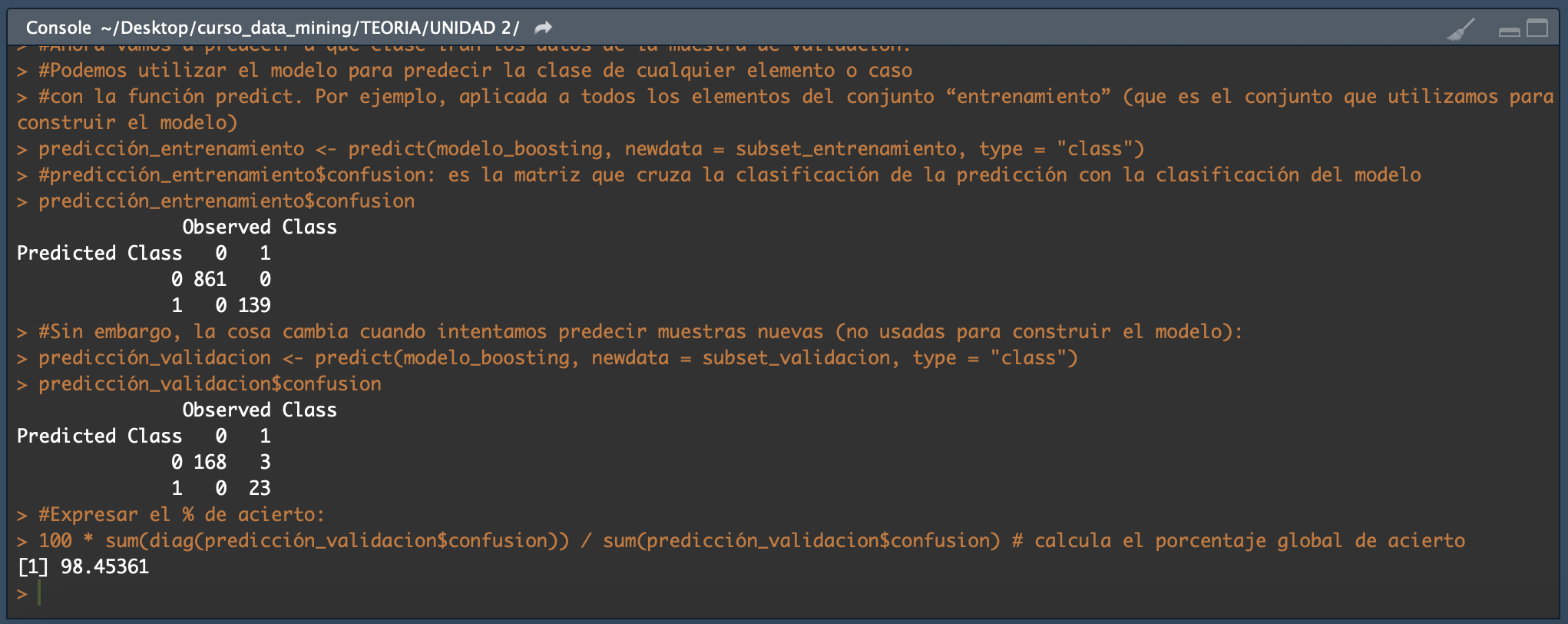


Vemos que usando el modelo es capaz de predecir correctamente (100%) a qué clase pertenecen las muestras.

Ahora utilizaremos el modelo para predecir la clase de las muestras del subset de validación:

predicción\_validacion <- predict(modelo\_boosting, newdata = subset\_validacion, type = "class")

predicción\_validacion$confusion

100 \* sum(diag(predicción\_validacion$confusion)) / sum(predicción\_validacion$confusion) # calcula el porcentaje global de acierto

Vemos que no todas las muestras fueron clasificadas con un 100% de acierto (98.45%). Hay tres muestras que fueron clasificadas en la clase 0 (CULTURAL) cuando deberían haber sido clasificadas en la clase 1 (NO CULTURAL). Con respecto a la clase 1 (CULTURAL), todas las muestras fueron clasificadas correctamente.

Ya que se ha clasificado correctamente las muestras del subset de validación, es probable que no haya problema de sobreajuste. Probaremos a crear el modelo utilizando la muestra total de empresas:

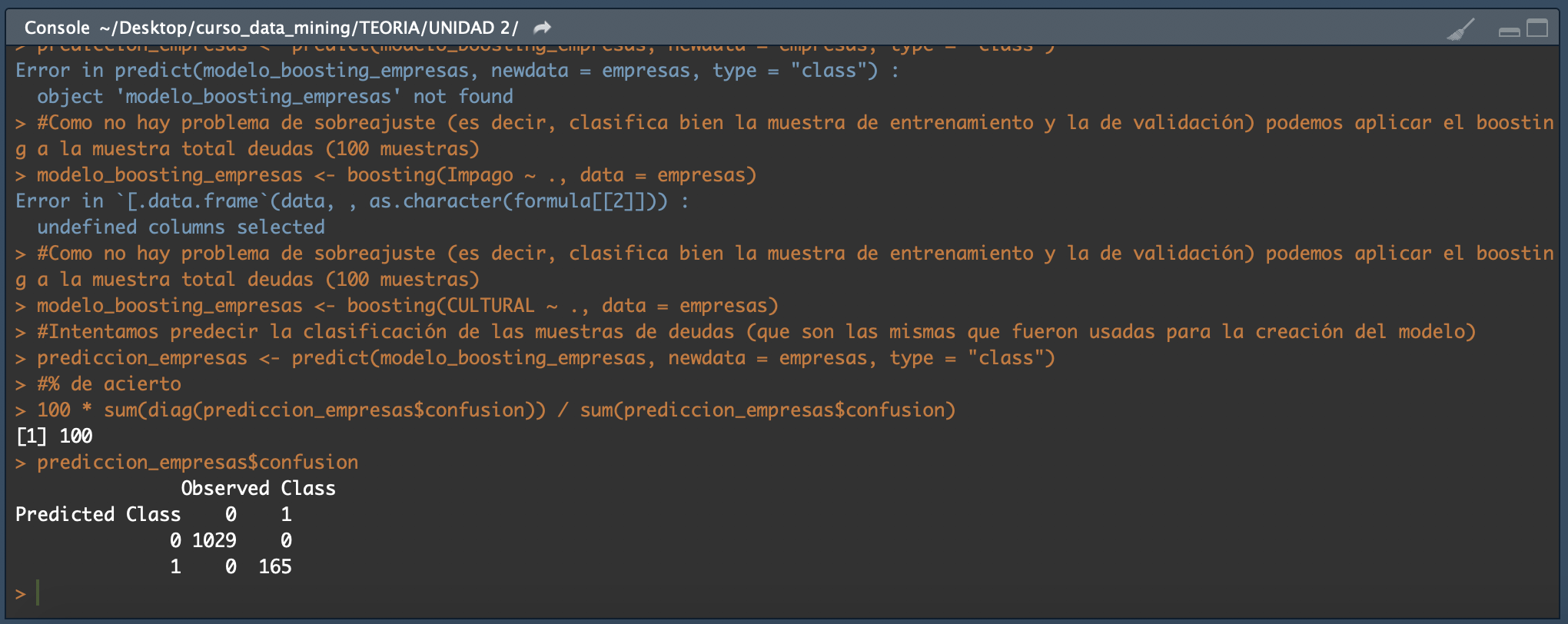
modelo\_boosting\_empresas <- boosting(CULTURAL ~ ., data = empresas)

prediccion\_empresas<- predict(modelo\_boosting\_empresas, newdata = empresas, type = "class")

#% de acierto

prediccion\_empresas$confusion

100 \* sum(diag(prediccion\_empresas$confusion)) / sum(prediccion\_empresas$confusion)

#100%

En este caso, podemos ver que el modelo mejora, ya que todas las muestras han sido clasificadas correctamente con respecto al modelo.

***Lea el conjunto de datos algas, del archivo "algas.RData, con 31 casos de 6 tipos diferentes de algas, y 19 variables con la concentración relativa de diferentes pigmentos. Aplique el método Random Forest (sustituya “vinos” por “algas” en las órdenes de la unidad) para identificar el tipo de alga (variable “clase”) a partir de las concentraciones de pigmentos. Utilice una muestra aleatoria de 20 casos y valide los resultados con los 11 restantes.***

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(123)

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

load("algas.RData")

Cargamos la librería necesaria para aplicar adaboost:

library(randomForest)

Creamos una muestra de entrenamiento (20 casos seleccionados al azar) y una muestra de validación (los 11 casos restantes).

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

entrenamiento <- algas[muestra, ]

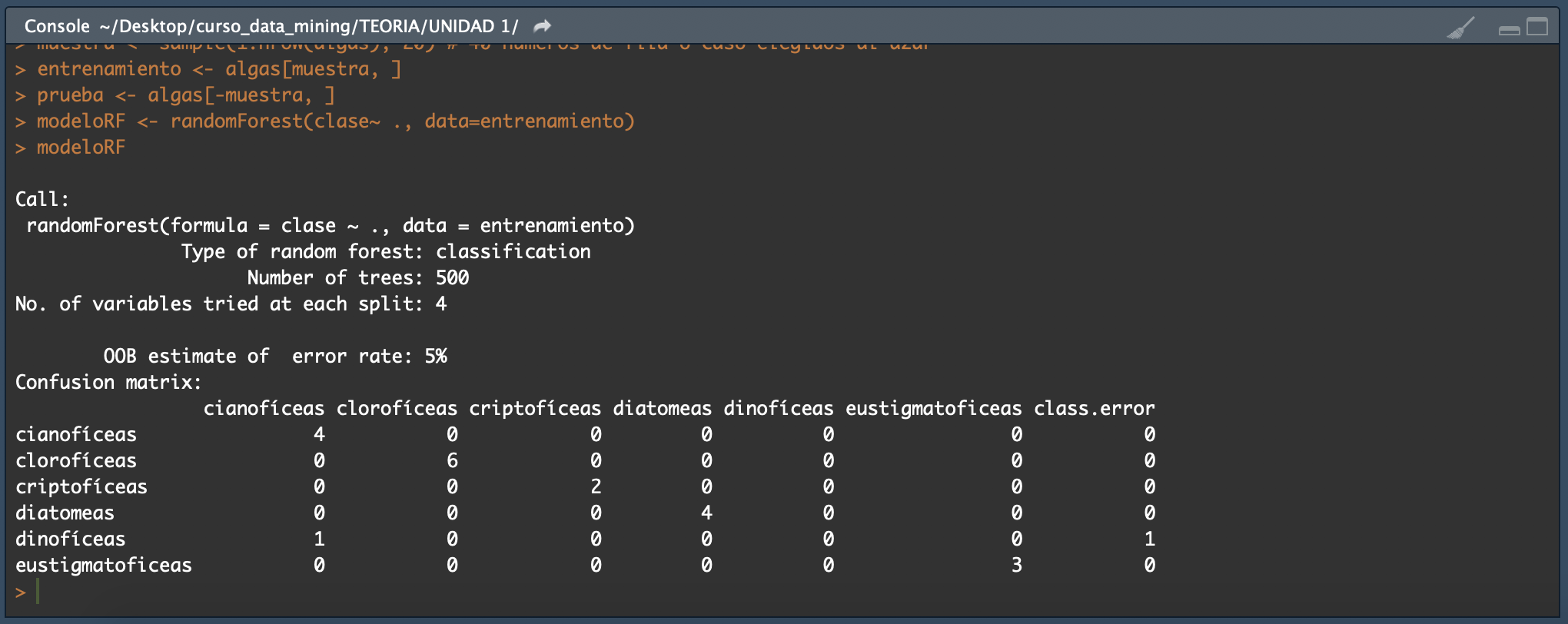
prueba <- algas[-muestra, ]

Aplicamos el método de randomForest para clasificar cada alga por la clase de alga teniendo en cuenta todas las concentraciones de los pigmentos:

modeloRF <- randomForest(clase~ ., data=entrenamiento)

Exploramos el modelo creado en relación a la importancia de cada variable:

modeloRF

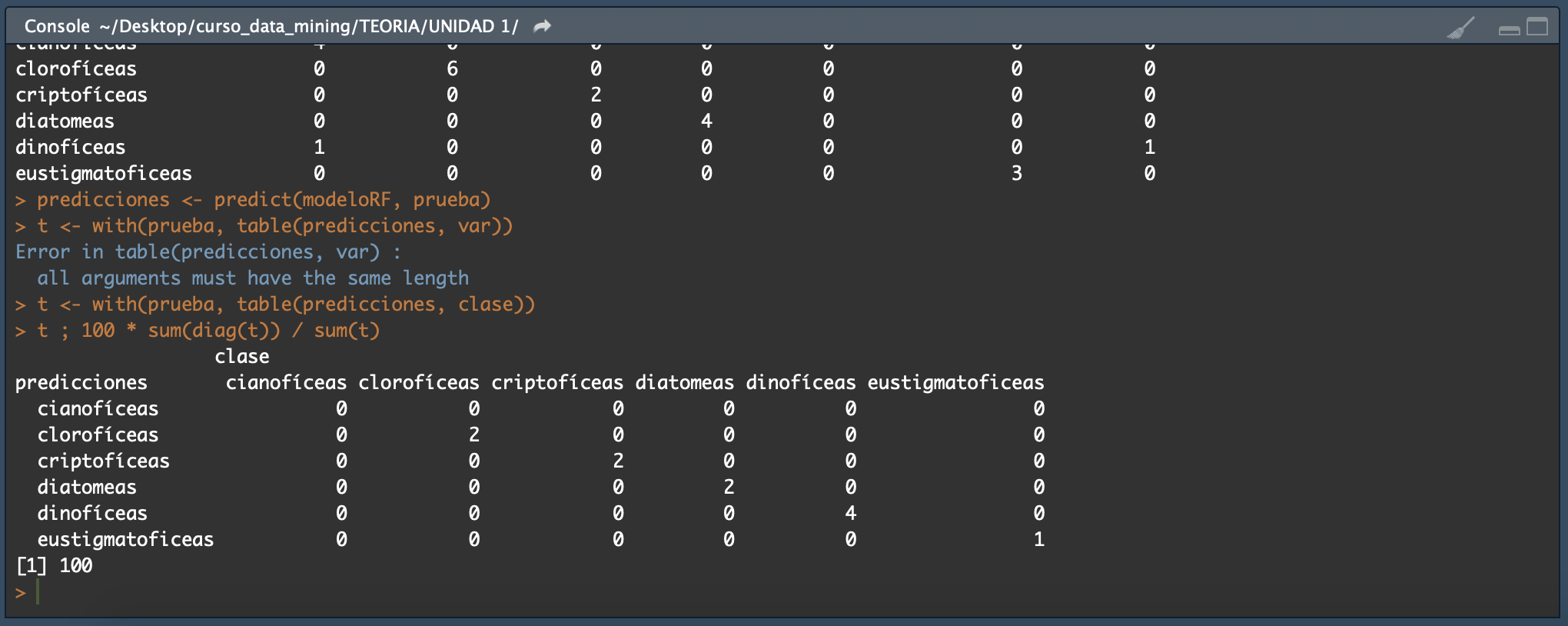


Vemos que la tasa de error en la clasificación es del 5%: sólo una muestra fue clasificada como dinofícea cuando debería haber sido clasificada como cianofícea.

Ahora utilizaremos el modelo creado para predecir la clase a la que están las 11 muestras restantes (subset de validación):

predicciones <- predict(modeloRF, prueba)

t <- with(prueba, table(predicciones, var))

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

Vemos que randomForest fue capaz de predecir la clase de las muestras del subset de validación.

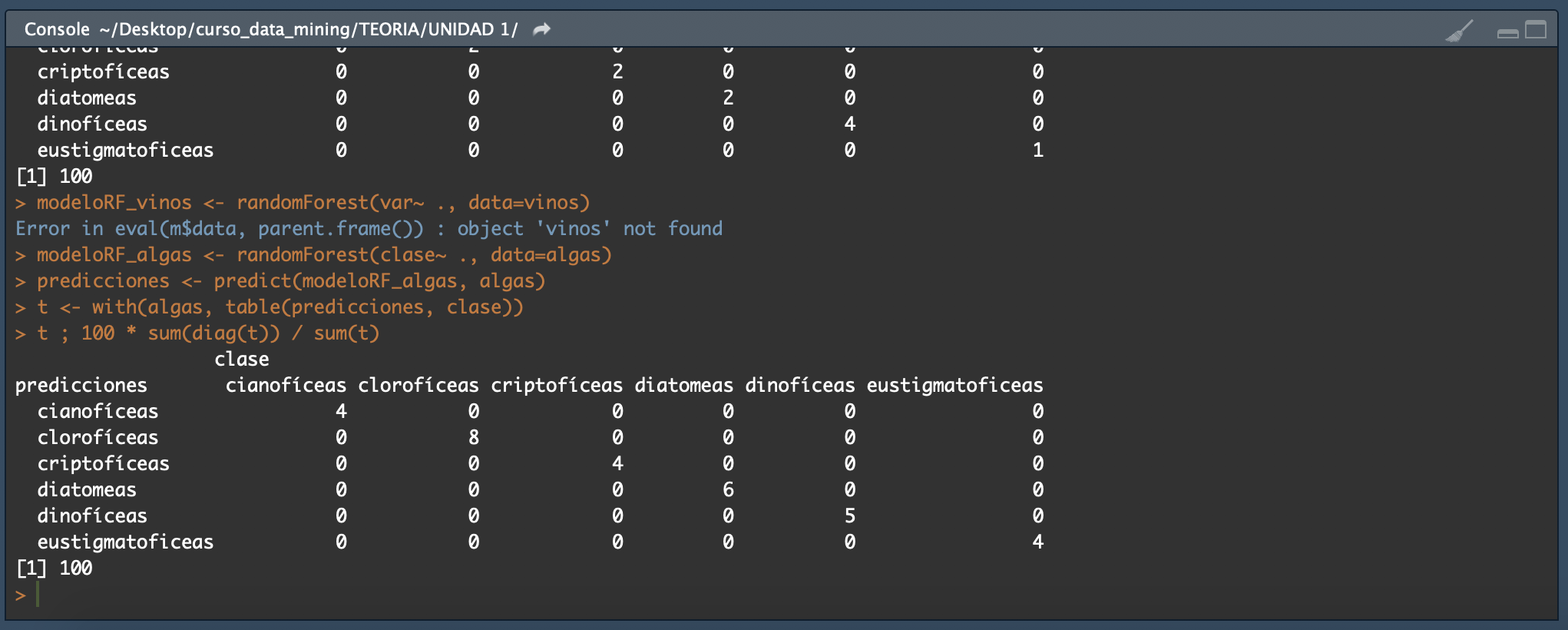
Podemos mejorar el modelo aplicando randomForest sobre el total de las muestras (algas):

modeloRF\_algas <- randomForest(clase~ ., data=algas)

predicciones <- predict(modeloRF\_algas, algas)

t <- with(algas, table(predicciones, clase))

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



Por el método de randomForest los resultados de clasificación son perfectos, es capaz de discernir la clase de cada muestra en función de las concentraciones de los pigmentos.