CURSO DE FORMACIÓN
"MINERÍA DE DATOS CON 'R' "
UNIVERSIDAD DE VIGO

TAREA 2 resuelta de forma orientativa

1) 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

{ notas: sustituya "deudas" por "empresas" en las órdenes del ejemplo de la unidad.
Puede obtener la importancia ordenada con sort(modelo$importance) }

```
setwd("C:/CURSO DM")
load("empresas.RData")      #  carga el conjunto de datos

names(empresas)             #  listado de variables
```

```
 [1] "CULTURAL" "npac"     "npacc"    "npactt"   "npbaim"   "npcaps"   "npcone"
 [8] "npdfcp"   "npextc"   "npfpro"   "npgper"   "npimps"   "npincn"   "npingf"
[15] "npmarv"   "npogex"   "npoinf"   "nppatn"   "nppc"     "npreje"   "nprene"
[22] "nprenf"   "nprevs"   "nprexp"   "nptes"
```

```
summary(empresas)           #  resumen descriptivo
```

```
 CULTURAL      npac                npacc                npactt
 0:1029   Min.   :     16615   Min.   :       229   Min.   :     40879
 1: 165   1st Qu.:    255194   1st Qu.:     98685   1st Qu.:    432775
          Median :    658630   Median :    245198   Median :   1196432
          Mean   :   5693995   Mean   :   2892705   Mean   :  13630905
          3rd Qu.:   1919394   3rd Qu.:    752458   3rd Qu.:   3135912
          Max.   :2421517000   Max.   :1467502000   Max.   :5501603000
     npbaim              npcaps             npcone                npdfcp
 Min.   :  -4132944   Min.   :    2971   Min.   :-3736509000   Min.   :        0
 1st Qu.:     -8442   1st Qu.:   12000   1st Qu.:   -1881934   1st Qu.:    21030
 Median :      7258   Median :   54090   Median :    -509448   Median :    89893
 Mean   :   1612010   Mean   :  638138   Mean   :   -8317870   Mean   :   896782
 3rd Qu.:     55804   3rd Qu.:  162632   3rd Qu.:    -167099   3rd Qu.:   347338
 Max.   :1485515000   Max.   :93500000   Max.   :     780862   Max.   :116743000
     npextc              npfpro              npgper               npimps
 Min.   :    -8241   Min.   :  -3180689   Min.   :-130811000   Min.   :-70053000
 1st Qu.:    40000   1st Qu.:    132143   1st Qu.:   -594990   1st Qu.:   -13609
 Median :   167886   Median :    411654   Median :   -235288   Median :    -1870
 Mean   :  1500006   Mean   :   7421842   Mean   :   -985653   Mean   :  -137112
 3rd Qu.:   530968   3rd Qu.:   1232870   3rd Qu.:   -111509   3rd Qu.:     1714
 Max.   :416970000   Max.   :3227989000   Max.   :     -3359   Max.   :  2848000
     npincn              npingf              npmarv               npogex
 Min.   :        52   Min.   :     -47   Min.   :-46.94530   Min.   :-634379000
 1st Qu.:    421716   1st Qu.:      53   1st Qu.:  0.00035   1st Qu.:   -443658
 Median :   1065598   Median :     652   Median :  0.02225   Median :   -148974
 Mean   :  12013710   Mean   :   63971   Mean   :  0.84694   Mean   :  -1497239
```

```
 3rd Qu.:   3265299   3rd Qu.:    6076   3rd Qu.:  0.06705   3rd Qu.:    -59719
 Max.   :5363837000   Max.   :27458749   Max.   : 54.94870   Max.   :     -2463
     npoinf              nppatn              nppc               npreje
 Min.   :     -42   Min.   :      1521   Min.   :      7212   Min.   : -5723801
 1st Qu.:      50   1st Qu.:    133538   1st Qu.:    159101   1st Qu.:     -6983
 Median :     632   Median :    425298   Median :    399854   Median :      5182
 Mean   :   63869   Mean   :   7465947   Mean   :   4628631   Mean   :   1474897
 3rd Qu.:    5732   3rd Qu.:   1300917   3rd Qu.:   1211212   3rd Qu.:     43680
 Max.   :27458749   Max.   :3204901000   Max.   :2272580000   Max.   :1415462000
     nprene              nprenf              nprevs             nprexp
 Min.   :-44.97920   Min.   :-78.50060   Min.   :         1   Min.   : -2708285
 1st Qu.:  0.00032   1st Qu.: -0.01738   1st Qu.:     49475   1st Qu.:     -2676
 Median :  0.02655   Median :  0.02530   Median :    240272   Median :     20260
 Mean   :  0.96188   Mean   :  0.86629   Mean   :   4430496   Mean   :   1666783
 3rd Qu.:  0.06782   3rd Qu.:  0.10110   3rd Qu.:    825920   3rd Qu.:     90318
 Max.   : 56.03620   Max.   : 53.00130   Max.   :2288286000   Max.   :1496275000
     nptes
 Min.   :  -105060
 1st Qu.:    13911
 Median :    45949
 Mean   :   741334
 3rd Qu.:   155702
 Max.   :612926000
```

```
library(adabag)          #  carga el paquete adabag que contiene la función boosting
set.seed(123)                                  # inicializa el arranque del generador aleatorio
muestra <- sample(1:nrow(empresas), 1000)          # muestra de 1000 números entre 1 y 1194
entrenamiento <- empresas[muestra, ]          # muestra de entrenamiento
prueba <- empresas[-muestra, ]                # muestra de validación

modelo <- boosting(CULTURAL ~ ., data = entrenamiento)
```

Una vez construido el modelo lo utilizamos para predecir o identificar a las empresas culturales:

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

```
               Observed Class
Predicted Class   0   1
              0 861   0
              1   0 139
```

El modelo ha identificado correctamente a todas las empresas del conjunto de entrenamiento. Como podría haber sobreajuste, validaremos el modelo con los datos no utilizados para su construcción:

```
resultados.prueba <- predict(modelo, newdata = prueba, type = "class")
t <- resultados.prueba$confusion
t  ;  100 * sum(diag(t)) / sum(t)     #  calcula el porcentaje global de acierto
```

```
               Observed Class
Predicted Class   0   1
              0 168   3
              1   0  23
[1] 98.45361
```

El porcentaje de acierto es muy elevado, 98,45%, por lo que el problema de sobreajuste es irrelevante. Construimos el modelo de nuevo utilizando toda la muestra disponible:

```
modelo <- boosting(CULTURAL ~ ., data = empresas)

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

t <- resultados$confusion

t  ;  100 * sum(diag(t)) / sum(t)     #  calcula el porcentaje global de acierto
```

```
               Observed Class
Predicted Class    0    1
              0 1029    0
              1    0  165
[1] 100
```

Identifica correctamente, de nuevo, a todas las empresas.

Veamos ahora cuales son las variables más importantes en el modelo. Para ello utilizamos el estadístico de importancia, del que podemos obtener los valores ordenados de menor a mayor con sort, de forma que las más importantes aparecerán al final:

```
sort(round(modelo$importance, 3))     # round permite elegir el número de decimales
```

```
> sort(round(modelo$importance, 3))
nppatn npoinf npreje npincn   nppc nprexp npogex npbaim nprevs npactt npimps npgper
 0.974  1.121  1.635  1.680  2.136  2.553  2.648  2.666  2.894  3.204  3.403  3.500
npcaps npfpro  npacc npcone npdfcp  nptes nprenf   npac npingf npextc npmarv nprene
 3.516  3.777  3.889  4.353  4.972  5.407  5.654  5.857  6.073  6.701  8.248 13.140
```

Las 3 variables más importantes son: nprene, npmarv, npextc.

2) 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.

```
setwd("C:/CURSO DM")
load("algas.RData")     #  lee el conjunto "algas"
library(randomForest)    #  carga el paquete randomForest
set.seed(12345)
muestra <- sample(1:nrow(algas), 20)      #  20 números de fila o caso elegidos al azar
entrenamiento <- algas[muestra, ]
prueba <- algas[-muestra, ]
set.seed(12345)
modelo <- randomForest(clase~ ., data=entrenamiento)
modelo
```

```
Call:
 randomForest(formula = clase ~ ., data = entrenamiento)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 0%
Confusion matrix:
           cianofíceas clorofíceas criptofíceas diatomeas dinofíceas eustigm. class.error
cianofíceas       2           0           0         0          0         0           0
clorofíceas       0           6           0         0          0         0           0
criptofíceas      0           0           2         0          0         0           0
diatomeas         0           0           0         5          0         0           0
dinofíceas        0           0           0         0          2         0           0
eustigmatoficeas  0           0           0         0          0         3           0
```

Todas las algas han sido identificadas correctamente en la muestra de entrenamiento; el error de clasificación en cada clase (class.error = 0 en todos los grupos) y global (error rate) es 0%.

Veamos que ocurre con la muestra de validación, para averiguar si el modelo construido es capaz de identificar correctamente los casos no utilizados para su construcción:

```
predicciones <- predict(modelo, prueba)
t <- with(prueba, table(predicciones, clase))    # Matriz de confusión
t  ;  100 * sum(diag(t)) / sum(t)
```

```
                                    clase
predicciones     cianofíceas clorofíceas criptofíceas diatomeas dinofíceas eustigm.
cianofíceas            2           0           0         0          0         0
clorofíceas            0           2           0         0          0         0
criptofíceas           0           0           2         0          0         0
diatomeas              0           0           0         1          0         0
dinofíceas             0           0           0         0          3         0
eustigmatoficeas       0           0           0         0          0         1

[1] 100
```

Tambien se obtiene el 100% de acierto.
Podemos construir el modelo con nuestra muestra completa:

```
modelo <- randomForest(clase~ ., data=algas)      # con toda la muestra
predicciones <- predict(modelo, algas)
t <- with(algas, table(predicciones, clase))    # Matriz de confusión
t  ;  100 * sum(diag(t)) / sum(t)
```

```
                                    clase
predicciones    cianofíceas clorofíceas criptofíceas diatomeas dinofíceas eustigm.
cianofíceas           4           0            0         0          0         0
clorofíceas           0           8            0         0          0         0
criptofíceas          0           0            4         0          0         0
diatomeas             0           0            0         6          0         0
dinofíceas            0           0            0         0          5         0
eustigmatoficeas      0           0            0         0          0         4

[1] 100
```

De nuevo tenemos el 100% de acierto.