**Ejercicio 1**

**1.1** Con bosque aleatorio de 400 árboles y 8 variables

Resultados del modelo

Number of observations used to build the model: 3220

Missing value imputation is active.

Call:

randomForest(formula = Tipo ~ .,

data = crs$dataset[crs$sample, c(crs$input, crs$target)],

ntree = 400, mtry = 8, importance = TRUE, replace = FALSE, na.action = randomForest::na.roughfix)

Type of random forest: classification

Number of trees: 400

No. of variables tried at each split: 8

OOB estimate of error rate: 4.81%

Confusion matrix:

email spam class.error

email 1908 59 0.02999492

spam 96 1157 0.07661612

Análisis de área bajo la curva

Call:

roc.default(response = crs$rf$y, predictor = as.numeric(crs$rf$predicted))

Data: as.numeric(crs$rf$predicted) in 1967 controls (crs$rf$y email) < 1253 cases (crs$rf$y spam).

Area under the curve: 0.9467

95% CI: 0.9384-0.955 (DeLong)

Tiempo transcurrido: 43.43 segs

**1.2** Con red neuronal de 1 capa oculta con 20 neuronas

Resultado del modelo

Resumen del modelo de Red neural (construido con 'nnet'):

A 57-20-1 network with 1238 weights.

Inputs: A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15, A.16, A.17, A.18, A.19, A.20, A.21, A.22, A.23, A.24, A.25, A.26, A.27, A.28, A.29, A.30, A.31, A.32, A.33, A.34, A.35, A.36, A.37, A.38, A.39, A.40, A.41, A.42, A.43, A.44, A.45, A.46, A.47, A.48, A.49, A.50, A.51, A.52, A.53, A.54, A.55, A.56, A.57.

Output: as.factor(Tipo).

Sum of Squares Residuals: 99.8325.

Opciones de construcción de red neural: skip-layer connections; entropy fitting.

In the following table:

b represents the bias associated with a node

h1 represents hidden layer node 1

i1 represents input node 1 (i.e., input variable 1)

o represents the output node

Tiempo transcurrido: 5.81 segs

**1.3** Bosque con potenciador

Resultado del modelo

Resumen del modelo Ada Boost:

Call:

ada(Tipo ~ ., data = crs$dataset[crs$train, c(crs$input, crs$target)],

control = rpart::rpart.control(maxdepth = 30, cp = 0.01,

minsplit = 20, xval = 10), iter = 400)

Loss: exponential Method: discrete Iteration: 400

Final Confusion Matrix for Data:

Final Prediction

True value email spam

email 1965 2

spam 6 1247

Train Error: 0.002

Out-Of-Bag Error: 0.006 iteration= 391

Additional Estimates of number of iterations:

train.err1 train.kap1

378 378

Variables actually used in tree construction:

[1] "A.1" "A.10" "A.11" "A.12" "A.13" "A.14" "A.15" "A.16" "A.17" "A.18" "A.19"

[12] "A.2" "A.20" "A.21" "A.22" "A.23" "A.24" "A.25" "A.26" "A.27" "A.28" "A.29"

[23] "A.3" "A.30" "A.31" "A.33" "A.35" "A.36" "A.37" "A.39" "A.40" "A.41" "A.42"

[34] "A.43" "A.44" "A.45" "A.46" "A.48" "A.49" "A.5" "A.50" "A.51" "A.52" "A.53"

[45] "A.54" "A.55" "A.56" "A.57" "A.6" "A.7" "A.8" "A.9"

Frequency of variables actually used:

A.57 A.55 A.52 A.19 A.56 A.25 A.21 A.12 A.16 A.5 A.50 A.45 A.7 A.53 A.46 A.3

367 343 325 318 294 263 243 241 241 229 227 199 181 170 150 148

A.27 A.17 A.18 A.10 A.37 A.23 A.11 A.49 A.42 A.6 A.24 A.28 A.8 A.33 A.1 A.20

135 132 132 127 111 101 96 94 93 91 89 84 82 80 75 62

A.9 A.2 A.26 A.36 A.39 A.44 A.43 A.14 A.54 A.13 A.51 A.35 A.30 A.29 A.41 A.48

59 57 57 51 50 50 49 48 45 44 43 35 34 27 17 9

A.22 A.40 A.15 A.31

8 7 4 3

Tiempo transcurrido: 1.63 mins

**2.1** Con bosque aleatorio de 400 árboles y 8 variables

Datos de prueba

Matriz de error para el modelo Bosque aleatorio en SpamData.csv [prueba] (cuentas):

Predicho

Real email spam

email 796 25

spam 52 508

Overall error: 6%, Averaged class error: 6%

Datos completos

Matriz de error para el modelo Bosque aleatorio en SpamData.csv (cuentas):

Predicho

Real email spam

email 2762 26

spam 61 1752

Overall error: 2%, Averaged class error: 2%

**2.2** Red neuronal con 20 neuronas en la capa oculta

Datos de prueba

Matriz de error para el modelo Red neural en SpamData.csv [prueba] (cuentas):

Predicho

Real email spam

email 774 47

spam 53 507

Overall error: 7%, Averaged class error: 8%

Datos completos

Matriz de error para el modelo Red neural en SpamData.csv (cuentas):

Predicho

Real email spam

email 2680 108

spam 120 1693

Overall error: 5%, Averaged class error: 6%

**2.3** Bosque con potenciador

Datos de prueba

Matriz de error para el modelo Ada Boost en SpamData.csv [prueba] (cuentas):

Predicho

Real email spam

email 794 27

spam 39 521

Overall error: 5%, Averaged class error: 5%

Datos completos

Matriz de error para el modelo Ada Boost en SpamData.csv (cuentas):

Predicho

Real email spam

email 2759 29

spam 45 1768

Overall error: 2%, Averaged class error: 2%

Los mejores resultados se obtienen con bosque aleatorio, con y sin método de potenciación.

En este caso, el indicador que nos importa son los falsos positivos, es decir, deseamos minimizar la cantidad de email que son predichos como spam, pues si enviamos este tipo de correos directamente a la papelera de forma automatizada, basándose en algunos de los métodos anteriores, es importante no enviar emails que podrían ser importantes.

En este sentido, el método de bosque aleatorio simple es el que funciona mejor con un 1.04 %

**3.** Desde el punto de vista de la curva ROC, cualquiera de los modelos es recomendable, pues el área bajo la curva es de aproximadamente 0.99, y el idea siempre que es que sea lo más cercano a 1. Como la curva ROC compara la tasa de falsos positivos con la de verdaderos positivos, y este no es el indicador que nos interesa, no es buena idea fiarse de este criterio.