SEPARATING SPAM FROM HAM

Techniques involved: Logistic regression, CART, Random Forest, prob statement in random forest

As a result of the huge number of spam emails being sent across the Internet each day, most email providers offer a spam filter that automatically flags likely spam messages and separates them from the ham. Though these filters use a number of techniques (e.g. looking up the sender in a so-called “Blackhole List” that contains IP addresses of likely spammers), most rely heavily on the analysis of the contents of an email via text analytics.

In this homework problem, we will build and evaluate a spam filter using a publicly available dataset. One source of spam messages in this dataset is the SpamAssassin corpus, which contains hand-labeled spam messages contributed by Internet users. The remaining spam was collected by Project Honey Pot, a project that collects spam messages and identifies spammers by publishing email address that humans would know not to contact but that bots might target with spam. The full dataset we will use was constructed as roughly a 75/25 mix of the ham and spam messages.

emails <- read.csv("emails.csv", stringsAsFactors = F)

The nchar() function counts the number of characters in a piece of text. How many characters are in the longest email in the dataset (where longest is measured in terms of the maximum number of characters)?

max(nchar(emails$text))

## [1] 43952

preparing the corpus

**library**(tm)

## Loading required package: NLP

corpus <- Corpus(VectorSource(emails$text))

corpus <- tm\_map(corpus, tolower)

corpus <- tm\_map(corpus, PlainTextDocument)

corpus <- tm\_map(corpus, removePunctuation)

corpus <- tm\_map(corpus, removeWords, stopwords("english"))

corpus <- tm\_map(corpus, stemDocument)

dtm <- DocumentTermMatrix(corpus)

dtm

## <<DocumentTermMatrix (documents: 5728, terms: 28687)>>

## Non-/sparse entries: 481719/163837417

## Sparsity : 100%

## Maximal term length: 24

## Weighting : term frequency (tf)

spdtm <- removeSparseTerms(dtm, 0.95)

spdtm

## <<DocumentTermMatrix (documents: 5728, terms: 330)>>

## Non-/sparse entries: 213551/1676689

## Sparsity : 89%

## Maximal term length: 10

## Weighting : term frequency (tf)

Build a data frame called emailsSparse from spdtm, and use the make.names function to make the variable names of emailsSparse valid.

emailsSparse <- as.data.frame(as.matrix(spdtm))

colnames(emailsSparse) <- make.names(colnames(emailsSparse))

colSums() is an R function that returns the sum of values for each variable in our data frame. Our data frame contains the number of times each word stem (columns) appeared in each email (rows). Therefore, colSums(emailsSparse) returns the number of times a word stem appeared across all the emails in the dataset. What is the word stem that shows up most frequently across all the emails in the dataset?

which.max(colSums(emailsSparse))

## enron

## 92

Add a variable called “spam” to emailsSparse containing the email spam labels.

emailsSparse$spam <- emails$spam

How many word stems appear at least 5000 times in the ham emails in the dataset?

sum(colSums(emailsSparse[emailsSparse$spam == 0,]) >= 5000)

## [1] 6

How many word stems appear at least 1000 times in the spam emails in the dataset?

sum(colSums(emailsSparse[emailsSparse$spam == 1,]) >= 1000)

## [1] 4

**The lists of most common words are significantly different between the spam and ham emails. This tells us that, the frequencies of these most common words are likely to help differentiate between spam and ham.**

**Several of the most common word stems from the ham documents, such as “enron”, “hou” (short for Houston), “vinc” (the word stem of “Vince”) and “kaminski”, are likely specific to Vincent Kaminski’s inbox. That means the models we build are personalized, and would need to be further tested before being used as a spam filter for another person.**

Building machine learning models

For each model, obtain the predicted spam probabilities for the training set.

*Be careful to obtain probabilities instead of predicted classes, because we will be using these values to compute training set AUC values.*

convert the dependent variable to a factor

class(emailsSparse$spam)

## [1] "integer"

emailsSparse$spam <- as.factor(emailsSparse$spam)

Split dataset

**library**(caTools)

set.seed(123)

split <- sample.split(emailsSparse$spam, SplitRatio = 0.7)

train <- subset(emailsSparse, split == T )

test <- subset(emailsSparse, split == F)

Build a logistic regression model

spamLog <- glm(spam ~., data = train, family = binomial)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(spamLog)

##

## Call:

## glm(formula = spam ~ ., family = binomial, data = train)

##

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -1.011 0.000 0.000 0.000 1.354

##

## Coefficients:

## Estimate Std. Error z value Pr(>|z|)

## (Intercept) -3.082e+01 1.055e+04 -0.003 0.998

## X000 1.474e+01 1.058e+04 0.001 0.999

## X2000 -3.631e+01 1.556e+04 -0.002 0.998

## X2001 -3.215e+01 1.318e+04 -0.002 0.998

## X713 -2.427e+01 2.914e+04 -0.001 0.999

## X853 -1.212e+00 5.942e+04 0.000 1.000

## abl -2.049e+00 2.088e+04 0.000 1.000

## access -1.480e+01 1.335e+04 -0.001 0.999

## account 2.488e+01 8.165e+03 0.003 0.998

## addit 1.463e+00 2.703e+04 0.000 1.000

## address -4.613e+00 1.113e+04 0.000 1.000

## allow 1.899e+01 6.436e+03 0.003 0.998

## alreadi -2.407e+01 3.319e+04 -0.001 0.999

## also 2.990e+01 1.378e+04 0.002 0.998

## analysi -2.405e+01 3.860e+04 -0.001 1.000

## anoth -8.744e+00 2.032e+04 0.000 1.000

## applic -2.649e+00 1.674e+04 0.000 1.000

## appreci -2.145e+01 2.762e+04 -0.001 0.999

## approv -1.302e+00 1.589e+04 0.000 1.000

## april -2.620e+01 2.208e+04 -0.001 0.999

## area 2.041e+01 2.266e+04 0.001 0.999

## arrang 1.069e+01 2.135e+04 0.001 1.000

## ask -7.746e+00 1.976e+04 0.000 1.000

## assist -1.128e+01 2.490e+04 0.000 1.000

## associ 9.049e+00 1.909e+04 0.000 1.000

## attach -1.037e+01 1.534e+04 -0.001 0.999

## attend -3.451e+01 3.257e+04 -0.001 0.999

## avail 8.651e+00 1.709e+04 0.001 1.000

## back -1.323e+01 2.272e+04 -0.001 1.000

## base -1.354e+01 2.122e+04 -0.001 0.999

## begin 2.228e+01 2.973e+04 0.001 0.999

## believ 3.233e+01 2.136e+04 0.002 0.999

## best -8.201e+00 1.333e+03 -0.006 0.995

## better 4.263e+01 2.360e+04 0.002 0.999

## book 4.301e+00 2.024e+04 0.000 1.000

## bring 1.607e+01 6.767e+04 0.000 1.000

## busi -4.803e+00 1.000e+04 0.000 1.000

## buy 4.170e+01 3.892e+04 0.001 0.999

## call -1.145e+00 1.111e+04 0.000 1.000

## can 3.762e+00 7.674e+03 0.000 1.000

## case -3.372e+01 2.880e+04 -0.001 0.999

## chang -2.717e+01 2.215e+04 -0.001 0.999

## check 1.425e+00 1.963e+04 0.000 1.000

## click 1.376e+01 7.077e+03 0.002 0.998

## com 1.936e+00 4.039e+03 0.000 1.000

## come -1.166e+00 1.511e+04 0.000 1.000

## comment -3.251e+00 3.387e+04 0.000 1.000

## communic 1.580e+01 8.958e+03 0.002 0.999

## compani 4.781e+00 9.186e+03 0.001 1.000

## complet -1.363e+01 2.024e+04 -0.001 0.999

## confer -7.503e-01 8.557e+03 0.000 1.000

## confirm -1.300e+01 1.514e+04 -0.001 0.999

## contact 1.530e+00 1.262e+04 0.000 1.000

## continu 1.487e+01 1.535e+04 0.001 0.999

## contract -1.295e+01 1.498e+04 -0.001 0.999

## copi -4.274e+01 3.070e+04 -0.001 0.999

## corp 1.606e+01 2.708e+04 0.001 1.000

## corpor -8.286e-01 2.818e+04 0.000 1.000

## cost -1.938e+00 1.833e+04 0.000 1.000

## cours 1.665e+01 1.834e+04 0.001 0.999

## creat 1.338e+01 3.946e+04 0.000 1.000

## credit 2.617e+01 1.314e+04 0.002 0.998

## crenshaw 9.994e+01 6.769e+04 0.001 0.999

## current 3.629e+00 1.707e+04 0.000 1.000

## custom 1.829e+01 1.008e+04 0.002 0.999

## data -2.609e+01 2.271e+04 -0.001 0.999

## date -2.786e+00 1.699e+04 0.000 1.000

## day -6.100e+00 5.866e+03 -0.001 0.999

## deal -1.129e+01 1.448e+04 -0.001 0.999

## dear -2.313e+00 2.306e+04 0.000 1.000

## depart -4.068e+01 2.509e+04 -0.002 0.999

## deriv -4.971e+01 3.587e+04 -0.001 0.999

## design -7.923e+00 2.939e+04 0.000 1.000

## detail 1.197e+01 2.301e+04 0.001 1.000

## develop 5.976e+00 9.455e+03 0.001 0.999

## differ -2.293e+00 1.075e+04 0.000 1.000

## direct -2.051e+01 3.194e+04 -0.001 0.999

## director -1.770e+01 1.793e+04 -0.001 0.999

## discuss -1.051e+01 1.915e+04 -0.001 1.000

## doc -2.597e+01 2.603e+04 -0.001 0.999

## don 2.129e+01 1.456e+04 0.001 0.999

## done 6.828e+00 1.882e+04 0.000 1.000

## due -4.163e+00 3.532e+04 0.000 1.000

## ect 8.685e-01 5.342e+03 0.000 1.000

## edu -2.122e-01 6.917e+02 0.000 1.000

## effect 1.948e+01 2.100e+04 0.001 0.999

## effort 1.606e+01 5.670e+04 0.000 1.000

## either -2.744e+01 4.000e+04 -0.001 0.999

## email 3.833e+00 1.186e+04 0.000 1.000

## end -1.311e+01 2.938e+04 0.000 1.000

## energi -1.620e+01 1.646e+04 -0.001 0.999

## engin 2.664e+01 2.394e+04 0.001 0.999

## enron -8.789e+00 5.719e+03 -0.002 0.999

## etc 9.470e-01 1.569e+04 0.000 1.000

## even -1.654e+01 2.289e+04 -0.001 0.999

## event 1.694e+01 1.851e+04 0.001 0.999

## expect -1.179e+01 1.914e+04 -0.001 1.000

## experi 2.460e+00 2.240e+04 0.000 1.000

## fax 3.537e+00 3.386e+04 0.000 1.000

## feel 2.596e+00 2.348e+04 0.000 1.000

## file -2.943e+01 2.165e+04 -0.001 0.999

## final 8.075e+00 5.008e+04 0.000 1.000

## financ -9.122e+00 7.524e+03 -0.001 0.999

## financi -9.747e+00 1.727e+04 -0.001 1.000

## find -2.623e+00 9.727e+03 0.000 1.000

## first -4.666e-01 2.043e+04 0.000 1.000

## follow 1.766e+01 3.080e+03 0.006 0.995

## form 8.483e+00 1.674e+04 0.001 1.000

## forward -3.484e+00 1.864e+04 0.000 1.000

## free 6.113e+00 8.121e+03 0.001 0.999

## friday -1.146e+01 1.996e+04 -0.001 1.000

## full 2.125e+01 2.190e+04 0.001 0.999

## futur 4.146e+01 1.439e+04 0.003 0.998

## gas -3.901e+00 4.160e+03 -0.001 0.999

## get 5.154e+00 9.737e+03 0.001 1.000

## gibner 2.901e+01 2.460e+04 0.001 0.999

## give -2.518e+01 2.130e+04 -0.001 0.999

## given -2.186e+01 5.426e+04 0.000 1.000

## good 5.399e+00 1.619e+04 0.000 1.000

## great 1.222e+01 1.090e+04 0.001 0.999

## group 5.264e-01 1.037e+04 0.000 1.000

## happi 1.939e-02 1.202e+04 0.000 1.000

## hear 2.887e+01 2.281e+04 0.001 0.999

## hello 2.166e+01 1.361e+04 0.002 0.999

## help 1.731e+01 2.791e+03 0.006 0.995

## high -1.982e+00 2.554e+04 0.000 1.000

## home 5.973e+00 8.965e+03 0.001 0.999

## hope -1.435e+01 2.179e+04 -0.001 0.999

## hou 6.852e+00 6.437e+03 0.001 0.999

## hour 2.478e+00 1.333e+04 0.000 1.000

## houston -1.855e+01 7.305e+03 -0.003 0.998

## howev -3.449e+01 3.562e+04 -0.001 0.999

## http 2.528e+01 2.107e+04 0.001 0.999

## idea -1.845e+01 3.892e+04 0.000 1.000

## immedi 6.285e+01 3.346e+04 0.002 0.999

## import -1.859e+00 2.236e+04 0.000 1.000

## includ -3.454e+00 1.799e+04 0.000 1.000

## increas 6.476e+00 2.329e+04 0.000 1.000

## industri -3.160e+01 2.373e+04 -0.001 0.999

## info -1.255e+00 4.857e+03 0.000 1.000

## inform 2.078e+01 8.549e+03 0.002 0.998

## interest 2.698e+01 1.159e+04 0.002 0.998

## intern -7.991e+00 3.351e+04 0.000 1.000

## internet 8.749e+00 1.100e+04 0.001 0.999

## interview -1.640e+01 1.873e+04 -0.001 0.999

## invest 3.201e+01 2.393e+04 0.001 0.999

## invit 4.304e+00 2.215e+04 0.000 1.000

## involv 3.815e+01 3.315e+04 0.001 0.999

## issu -3.708e+01 3.396e+04 -0.001 0.999

## john -5.326e-01 2.856e+04 0.000 1.000

## join -3.824e+01 2.334e+04 -0.002 0.999

## juli -1.358e+01 3.009e+04 0.000 1.000

## just -1.021e+01 1.114e+04 -0.001 0.999

## kaminski -1.812e+01 6.029e+03 -0.003 0.998

## keep 1.867e+01 2.782e+04 0.001 0.999

## kevin -3.779e+01 4.738e+04 -0.001 0.999

## know 1.277e+01 1.526e+04 0.001 0.999

## last 1.046e+00 1.372e+04 0.000 1.000

## let -2.763e+01 1.462e+04 -0.002 0.998

## life 5.812e+01 3.864e+04 0.002 0.999

## like 5.649e+00 7.660e+03 0.001 0.999

## line 8.743e+00 1.236e+04 0.001 0.999

## link -6.929e+00 1.345e+04 -0.001 1.000

## list -8.692e+00 2.149e+03 -0.004 0.997

## locat 2.073e+01 1.597e+04 0.001 0.999

## london 6.745e+00 1.642e+04 0.000 1.000

## long -1.489e+01 1.934e+04 -0.001 0.999

## look -7.031e+00 1.563e+04 0.000 1.000

## lot -1.964e+01 1.321e+04 -0.001 0.999

## made 2.820e+00 2.743e+04 0.000 1.000

## mail 7.584e+00 1.021e+04 0.001 0.999

## make 2.901e+01 1.528e+04 0.002 0.998

## manag 6.014e+00 1.445e+04 0.000 1.000

## mani 1.885e+01 1.442e+04 0.001 0.999

## mark -3.350e+01 3.208e+04 -0.001 0.999

## market 7.895e+00 8.012e+03 0.001 0.999

## may -9.434e+00 1.397e+04 -0.001 0.999

## mean 6.078e-01 2.952e+04 0.000 1.000

## meet -1.063e+00 1.263e+04 0.000 1.000

## member 1.381e+01 2.343e+04 0.001 1.000

## mention -2.279e+01 2.714e+04 -0.001 0.999

## messag 1.716e+01 2.562e+03 0.007 0.995

## might 1.244e+01 1.753e+04 0.001 0.999

## model -2.292e+01 1.049e+04 -0.002 0.998

## monday -1.034e+00 3.233e+04 0.000 1.000

## money 3.264e+01 1.321e+04 0.002 0.998

## month -3.727e+00 1.112e+04 0.000 1.000

## morn -2.645e+01 3.403e+04 -0.001 0.999

## move -3.834e+01 3.011e+04 -0.001 0.999

## much 3.775e-01 1.392e+04 0.000 1.000

## name 1.672e+01 1.322e+04 0.001 0.999

## need 8.437e-01 1.221e+04 0.000 1.000

## net 1.256e+01 2.197e+04 0.001 1.000

## new 1.003e+00 1.009e+04 0.000 1.000

## next. 1.492e+01 1.724e+04 0.001 0.999

## note 1.446e+01 2.294e+04 0.001 0.999

## now 3.790e+01 1.219e+04 0.003 0.998

## number -9.622e+00 1.591e+04 -0.001 1.000

## offer 1.174e+01 1.084e+04 0.001 0.999

## offic -1.344e+01 2.311e+04 -0.001 1.000

## one 1.241e+01 6.652e+03 0.002 0.999

## onlin 3.589e+01 1.665e+04 0.002 0.998

## open 2.114e+01 2.961e+04 0.001 0.999

## oper -1.696e+01 2.757e+04 -0.001 1.000

## opportun -4.131e+00 1.918e+04 0.000 1.000

## option -1.085e+00 9.325e+03 0.000 1.000

## order 6.533e+00 1.242e+04 0.001 1.000

## origin 3.226e+01 3.818e+04 0.001 0.999

## part 4.594e+00 3.483e+04 0.000 1.000

## particip -1.154e+01 1.738e+04 -0.001 0.999

## peopl -1.864e+01 1.439e+04 -0.001 0.999

## per 1.367e+01 1.273e+04 0.001 0.999

## person 1.870e+01 9.575e+03 0.002 0.998

## phone -6.957e+00 1.172e+04 -0.001 1.000

## place 9.005e+00 3.661e+04 0.000 1.000

## plan -1.830e+01 6.320e+03 -0.003 0.998

## pleas -7.961e+00 9.484e+03 -0.001 0.999

## point 5.498e+00 3.403e+04 0.000 1.000

## posit -1.543e+01 2.316e+04 -0.001 0.999

## possibl -1.366e+01 2.492e+04 -0.001 1.000

## power -5.643e+00 1.173e+04 0.000 1.000

## present -6.163e+00 1.278e+04 0.000 1.000

## price 3.428e+00 7.850e+03 0.000 1.000

## problem 1.262e+01 9.763e+03 0.001 0.999

## process -2.957e-01 1.191e+04 0.000 1.000

## product 1.016e+01 1.345e+04 0.001 0.999

## program 1.444e+00 1.183e+04 0.000 1.000

## project 2.173e+00 1.497e+04 0.000 1.000

## provid 2.422e-01 1.859e+04 0.000 1.000

## public -5.250e+01 2.341e+04 -0.002 0.998

## put -1.052e+01 2.681e+04 0.000 1.000

## question -3.467e+01 1.859e+04 -0.002 0.999

## rate -3.112e+00 1.319e+04 0.000 1.000

## read -1.527e+01 2.145e+04 -0.001 0.999

## real 2.046e+01 2.358e+04 0.001 0.999

## realli -2.667e+01 4.640e+04 -0.001 1.000

## receiv 5.765e-01 1.585e+04 0.000 1.000

## recent -2.067e+00 1.780e+04 0.000 1.000

## regard -3.668e+00 1.511e+04 0.000 1.000

## relat -5.114e+01 1.793e+04 -0.003 0.998

## remov 2.325e+01 2.484e+04 0.001 0.999

## repli 1.538e+01 2.916e+04 0.001 1.000

## report -1.482e+01 1.477e+04 -0.001 0.999

## request -1.232e+01 1.167e+04 -0.001 0.999

## requir 5.004e-01 2.937e+04 0.000 1.000

## research -2.826e+01 1.553e+04 -0.002 0.999

## resourc -2.735e+01 3.522e+04 -0.001 0.999

## respond 2.974e+01 3.888e+04 0.001 0.999

## respons -1.960e+01 3.667e+04 -0.001 1.000

## result -5.002e-01 3.140e+04 0.000 1.000

## resum -9.219e+00 2.100e+04 0.000 1.000

## return 1.745e+01 1.844e+04 0.001 0.999

## review -4.825e+00 1.013e+04 0.000 1.000

## right 2.312e+01 1.590e+04 0.001 0.999

## risk -4.001e+00 1.718e+04 0.000 1.000

## robert -2.096e+01 2.907e+04 -0.001 0.999

## run -5.162e+01 4.434e+04 -0.001 0.999

## say 7.366e+00 2.217e+04 0.000 1.000

## schedul 1.919e+00 3.580e+04 0.000 1.000

## school -3.870e+00 2.882e+04 0.000 1.000

## secur -1.604e+01 2.201e+03 -0.007 0.994

## see -1.120e+01 1.293e+04 -0.001 0.999

## send -2.427e+01 1.222e+04 -0.002 0.998

## sent -1.488e+01 2.195e+04 -0.001 0.999

## servic -7.164e+00 1.235e+04 -0.001 1.000

## set -9.353e+00 2.627e+04 0.000 1.000

## sever 2.041e+01 3.093e+04 0.001 0.999

## shall 1.930e+01 3.075e+04 0.001 0.999

## shirley -7.133e+01 6.329e+04 -0.001 0.999

## short -8.974e+00 1.721e+04 -0.001 1.000

## sinc -3.438e+00 3.546e+04 0.000 1.000

## sincer -2.073e+01 3.515e+04 -0.001 1.000

## site 8.689e+00 1.496e+04 0.001 1.000

## softwar 2.575e+01 1.059e+04 0.002 0.998

## soon 2.350e+01 3.731e+04 0.001 0.999

## sorri 6.036e+00 2.299e+04 0.000 1.000

## special 1.777e+01 2.755e+04 0.001 0.999

## specif -2.337e+01 3.083e+04 -0.001 0.999

## start 1.437e+01 1.897e+04 0.001 0.999

## state 1.221e+01 1.677e+04 0.001 0.999

## still 3.878e+00 2.622e+04 0.000 1.000

## stinson -4.345e+01 2.697e+04 -0.002 0.999

## student -1.815e+01 2.186e+04 -0.001 0.999

## subject 3.041e+01 1.055e+04 0.003 0.998

## success 4.344e+00 2.783e+04 0.000 1.000

## suggest -3.842e+01 4.475e+04 -0.001 0.999

## support -1.539e+01 1.976e+04 -0.001 0.999

## sure -5.503e+00 2.078e+04 0.000 1.000

## system 3.778e+00 9.149e+03 0.000 1.000

## take 5.731e+00 1.716e+04 0.000 1.000

## talk -1.011e+01 2.021e+04 -0.001 1.000

## team 7.940e+00 2.570e+04 0.000 1.000

## term 2.013e+01 2.303e+04 0.001 0.999

## thank -3.890e+01 1.059e+04 -0.004 0.997

## thing 2.579e+01 1.341e+04 0.002 0.998

## think -1.218e+01 2.077e+04 -0.001 1.000

## thought 1.243e+01 3.023e+04 0.000 1.000

## thursday -1.491e+01 3.262e+04 0.000 1.000

## time -5.921e+00 8.335e+03 -0.001 0.999

## today -1.762e+01 1.965e+04 -0.001 0.999

## togeth -2.355e+01 1.869e+04 -0.001 0.999

## trade -1.755e+01 1.483e+04 -0.001 0.999

## tri 9.278e-01 1.282e+04 0.000 1.000

## tuesday -2.808e+01 3.959e+04 -0.001 0.999

## two -2.573e+01 1.844e+04 -0.001 0.999

## type -1.447e+01 2.755e+04 -0.001 1.000

## understand 9.307e+00 2.342e+04 0.000 1.000

## unit -4.020e+00 3.008e+04 0.000 1.000

## univers 1.228e+01 2.197e+04 0.001 1.000

## updat -1.510e+01 1.448e+04 -0.001 0.999

## use -1.385e+01 9.382e+03 -0.001 0.999

## valu 9.024e-01 1.360e+04 0.000 1.000

## version -3.606e+01 2.939e+04 -0.001 0.999

## vinc -3.735e+01 8.647e+03 -0.004 0.997

## visit 2.585e+01 1.170e+04 0.002 0.998

## vkamin -6.649e+01 5.703e+04 -0.001 0.999

## want -2.555e+00 1.106e+04 0.000 1.000

## way 1.339e+01 1.138e+04 0.001 0.999

## web 2.791e+00 1.686e+04 0.000 1.000

## websit -2.563e+01 1.848e+04 -0.001 0.999

## wednesday -1.526e+01 2.642e+04 -0.001 1.000

## week -6.795e+00 1.046e+04 -0.001 0.999

## well -2.222e+01 9.713e+03 -0.002 0.998

## will -1.119e+01 5.980e+03 -0.002 0.999

## wish 1.173e+01 3.175e+04 0.000 1.000

## within 2.900e+01 2.163e+04 0.001 0.999

## without 1.942e+01 1.763e+04 0.001 0.999

## work -1.099e+01 1.160e+04 -0.001 0.999

## write 4.406e+01 2.825e+04 0.002 0.999

## www -7.867e+00 2.224e+04 0.000 1.000

## year -1.010e+01 1.039e+04 -0.001 0.999

##

## (Dispersion parameter for binomial family taken to be 1)

##

## Null deviance: 4409.49 on 4009 degrees of freedom

## Residual deviance: 13.46 on 3679 degrees of freedom

## AIC: 675.46

##

## Number of Fisher Scoring iterations: 25

spamLog\_pred <- predict(spamLog)

table(train$spam, spamLog\_pred >= 0.5)

##

## FALSE TRUE

## 0 3052 0

## 1 4 954

**library**(ROCR)

## Loading required package: gplots

##

## Attaching package: 'gplots'

##

## The following object is masked from 'package:stats':

##

## lowess

ROCRpred <- prediction(spamLog\_pred, train$spam)

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9999959

Build a CART model

**library**(rpart)

**library**(rpart.plot)

spamCART <- rpart(spam~., data = train, method = "class")

prp(spamCART)

A picture containing schematic

Description automatically generated

spamCART\_pred <- predict(spamCART)

table(train$spam, spamCART\_pred[,2] >= 0.5)

##

## FALSE TRUE

## 0 2885 167

## 1 64 894

ROCRpred <- prediction(spamCART\_pred[, 2], train$spam)

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9696044

Build a random forest model

**library**(randomForest)

## randomForest 4.6-10

## Type rfNews() to see new features/changes/bug fixes.

set.seed(123)

spamRF <- randomForest(spam~., data = train)

spamRF\_pred <- predict(spamRF)

table(train$spam, spamRF\_pred)

## spamRF\_pred

## 0 1

## 0 3013 39

## 1 44 914

What is the training set AUC of spamRF? (Remember to pass the argument type=“prob” to the predict function to get predicted probabilities for a random forest model. The probabilities will be the second column of the output.)

spamRF\_pred <- predict(spamRF, type = "prob")

ROCRpred <- prediction(spamRF\_pred[,2], train$spam)

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9979116

Conclusion on the training set

Logistic regression model has the best training set performance, in terms of accuracy and AUC.

Evaluating on the test set

spamLog\_pred <- predict(spamLog, newdata = test)

spamCART\_pred <- predict(spamCART, newdata = test)

spamRF\_pred <- predict(spamRF, newdata = test, type = "prob")

Logistic regression model

table(test$spam, spamLog\_pred >= 0.5)

##

## FALSE TRUE

## 0 1258 50

## 1 34 376

ROCRpred <- prediction(spamLog\_pred, test$spam)

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9767994

CART model

table(test$spam, spamCART\_pred[,2] >= 0.5)

##

## FALSE TRUE

## 0 1228 80

## 1 24 386

ROCRpred <- prediction(spamCART\_pred[,2], test$spam)

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.963176

Random Forest model

table(test$spam, spamRF\_pred[,2]>=0.5)

##

## FALSE TRUE

## 0 1290 18

## 1 24 386

ROCRpred <- prediction(spamRF\_pred[,2], test$spam)

as.numeric(performance(ROCRpred, "auc")@y.values)

## [1] 0.9975656

Conclusion on the test set

Random Forest model has the best test set performance, in terms of accuracy and AUC.

Both CART and random forest had very similar accuracies on the training and testing sets. However, logistic regression obtained nearly perfect accuracy and AUC on the training set and had far-from-perfect performance on the testing set. This is an indicator of overfitting.