

# Chapter 4: Classification using Naive Bayes

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

## Example: Filtering spam SMS messages

### Step 1: Download the data

```
URL <- "http://cox.csueastbay.edu/~esuess/classes/Statistics_6620/Presentations/ml6/sms_spam.csv"
download.file(URL, destfile = "./sms_spam.csv", method="curl")
```

### Step 2: Exploring and preparing the data —

```
# read the sms data into the sms data frame
sms_raw <- read.csv("sms_spam.csv", stringsAsFactors = FALSE)

# examine the structure of the sms data
str(sms_raw)

## 'data.frame':    5559 obs. of  2 variables:
## $ type: chr  "ham" "ham" "ham" "spam" ...
## $ text: chr  "Hope you are having a good week. Just checking in" "K..give back my thanks." "Am also
# convert spam/ham to factor.
sms_raw$type <- factor(sms_raw$type)

# examine the type variable more carefully
str(sms_raw$type)

##  Factor w/ 2 levels "ham","spam": 1 1 1 2 2 1 1 1 2 1 ...
table(sms_raw$type)

##
##   ham  spam
## 4812   747

# build a corpus using the text mining (tm) package
library(tm)
```

```

## Loading required package: NLP
sms_corpus <- VCorpus(VectorSource(sms_raw$text))

# examine the sms corpus
print(sms_corpus)

## <<VCorpus>>
## Metadata: corpus specific: 0, document level (indexed): 0
## Content: documents: 5559
inspect(sms_corpus[1:2])

## <<VCorpus>>
## Metadata: corpus specific: 0, document level (indexed): 0
## Content: documents: 2
##
## [[1]]
## <<PlainTextDocument>>
## Metadata: 7
## Content: chars: 49
##
## [[2]]
## <<PlainTextDocument>>
## Metadata: 7
## Content: chars: 23
as.character(sms_corpus[[1]])

## [1] "Hope you are having a good week. Just checking in"
lapply(sms_corpus[1:2], as.character)

## $`1`
## [1] "Hope you are having a good week. Just checking in"
##
## $`2`
## [1] "K..give back my thanks."
# clean up the corpus using tm_map()
sms_corpus_clean <- tm_map(sms_corpus, content_transformer(tolower))

# show the difference between sms_corpus and corpus_clean
as.character(sms_corpus[[1]])

## [1] "Hope you are having a good week. Just checking in"
as.character(sms_corpus_clean[[1]])

## [1] "hope you are having a good week. just checking in"
sms_corpus_clean <- tm_map(sms_corpus_clean, removeNumbers) # remove numbers
sms_corpus_clean <- tm_map(sms_corpus_clean, removeWords, stopwords()) # remove stop words
sms_corpus_clean <- tm_map(sms_corpus_clean, removePunctuation) # remove punctuation

# tip: create a custom function to replace (rather than remove) punctuation
removePunctuation("hello...world")

## [1] "helloworld"

```

```

replacePunctuation <- function(x) { gsub("[[:punct:]]+", " ", x) }
replacePunctuation("hello...world")

## [1] "hello world"
# illustration of word stemming
library(SnowballC)
wordStem(c("learn", "learned", "learning", "learns"))

## [1] "learn" "learn" "learn" "learn"
sms_corpus_clean <- tm_map(sms_corpus_clean, stemDocument)

sms_corpus_clean <- tm_map(sms_corpus_clean, stripWhitespace) # eliminate unneeded whitespace

# examine the final clean corpus
lapply(sms_corpus[1:3], as.character)

## $`1`
## [1] "Hope you are having a good week. Just checking in"
##
## $`2`
## [1] "K..give back my thanks."
##
## $`3`
## [1] "Am also doing in cbe only. But have to pay."
lapply(sms_corpus_clean[1:3], as.character)

## $`1`
## [1] "hope good week just check"
##
## $`2`
## [1] "kgive back thank"
##
## $`3`
## [1] "also cbe pay"
# create a document-term sparse matrix
sms_dtm <- DocumentTermMatrix(sms_corpus_clean)

# alternative solution: create a document-term sparse matrix directly from the SMS corpus
sms_dtm2 <- DocumentTermMatrix(sms_corpus, control = list(
  tolower = TRUE,
  removeNumbers = TRUE,
  stopwords = TRUE,
  removePunctuation = TRUE,
  stemming = TRUE
))

# alternative solution: using custom stop words function ensures identical result
sms_dtm3 <- DocumentTermMatrix(sms_corpus, control = list(
  tolower = TRUE,
  removeNumbers = TRUE,
  stopwords = function(x) { removeWords(x, stopwords()) },
  removePunctuation = TRUE,

```

```

    stemming = TRUE
))

# compare the result
sms_dtm

## <<DocumentTermMatrix (documents: 5559, terms: 6559)>>
## Non-/sparse entries: 42147/36419334
## Sparsity           : 100%
## Maximal term length: 40
## Weighting          : term frequency (tf)
sms_dtm2

## <<DocumentTermMatrix (documents: 5559, terms: 6961)>>
## Non-/sparse entries: 43221/38652978
## Sparsity           : 100%
## Maximal term length: 40
## Weighting          : term frequency (tf)
sms_dtm3

## <<DocumentTermMatrix (documents: 5559, terms: 6559)>>
## Non-/sparse entries: 42147/36419334
## Sparsity           : 100%
## Maximal term length: 40
## Weighting          : term frequency (tf)

# creating training and test datasets
sms_dtm_train <- sms_dtm[1:4169, ]
sms_dtm_test  <- sms_dtm[4170:5559, ]

# also save the labels
sms_train_labels <- sms_raw[1:4169, ]$type
sms_test_labels  <- sms_raw[4170:5559, ]$type

# check that the proportion of spam is similar
prop.table(table(sms_train_labels))

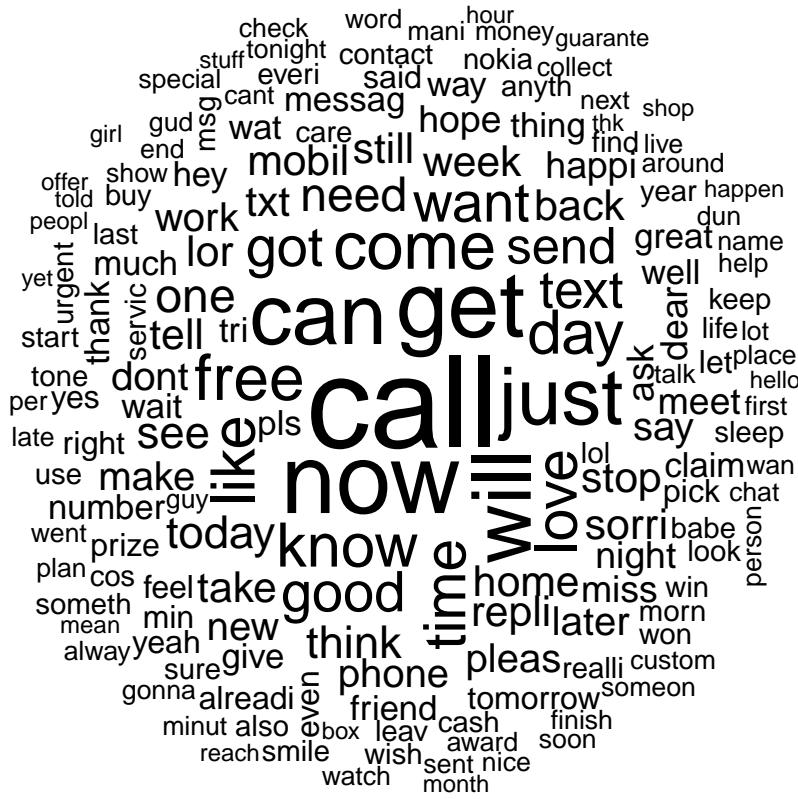
## sms_train_labels
##      ham      spam
## 0.8647158 0.1352842
prop.table(table(sms_test_labels))

## sms_test_labels
##      ham      spam
## 0.8683453 0.1316547

# word cloud visualization
library(wordcloud)

## Loading required package: RColorBrewer
wordcloud(sms_corpus_clean, min.freq = 50, random.order = FALSE)

```



```
# subset the training data into spam and ham groups
spam <- subset(sms_raw, type == "spam")
ham <- subset(sms_raw, type == "ham")

wordcloud(spam$text, max.words = 40, scale = c(3, 0.5))

## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents

## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents

urgent
customer won
txt send mins service
stop chat you mobile
stop will latest your prize
get contact reply per guaranteed
get just line £1000
150ppm phone win cash text
nokia awarded week
claim new
please call

wordcloud(ham$text, max.words = 40, scale = c(3, 0.5))

## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

can want how now night  
home time dont get  
later still its back you come  
cant see got day tell  
think much well ill but  
love going lor today sorry  
just will take send  
good call one need  
like know

```
  sms_dtm_freq_train <- removeSparseTerms(sms_dtm_train, 0.999)  
  sms_dtm_freq_train
```

```
## <<DocumentTermMatrix (documents: 4169, terms: 1104)>>
## Non-/sparse entries: 24827/4577749
## Sparsity           : 99%
## Maximal term length: 19
## Weighting          : term frequency (tf)

# indicator features for frequent words
findFreqTerms(sms_dtm_train, 5)
```

```

## [1] "èwk"
## [4] "abiola"
## [7] "accept"
## [10] "across"
## [13] "actual"
## [16] "admir"
## [19] "aft"
## [22] "ago"
## [25] "aight"
## [28] "aiyo"
## [31] "alon"
## [34] "also"
## [37] "announc"
## [40] "anymor"
## [43] "anytim"
## [46] "app"
## [49] "arcad"
## [52] "argu"
## [55] "around"
## [58] "asap"
## [61] "attempt"
## [64] "ave"
## [67] "awak"
## [70] "awesom"
## [73] "awful"
## [76] "awfully"
## [79] "awfully"
## [82] "awfully"
## [85] "awfully"
## [88] "awfully"
## [91] "awfully"
## [94] "awfully"
## [97] "awfully"
## [100] "awfully"
## [103] "awfully"
## [106] "awfully"
## [109] "awfully"
## [112] "awfully"
## [115] "awfully"
## [118] "awfully"
## [121] "awfully"
## [124] "awfully"
## [127] "awfully"
## [130] "awfully"
## [133] "awfully"
## [136] "awfully"
## [139] "awfully"
## [142] "awfully"
## [145] "awfully"
## [148] "awfully"
## [151] "awfully"
## [154] "awfully"
## [157] "awfully"
## [160] "awfully"
## [163] "awfully"
## [166] "awfully"
## [169] "awfully"
## [172] "awfully"
## [175] "awfully"
## [178] "awfully"
## [181] "awfully"
## [184] "awfully"
## [187] "awfully"
## [190] "awfully"
## [193] "awfully"
## [196] "awfully"
## [199] "awfully"
## [202] "awfully"
## [205] "awfully"
## [208] "awfully"
## [211] "awfully"
## [214] "awfully"
## [217] "awfully"
## [220] "awfully"
## [223] "awfully"
## [226] "awfully"
## [229] "awfully"
## [232] "awfully"
## [235] "awfully"
## [238] "awfully"
## [241] "awfully"
## [244] "awfully"
## [247] "awfully"
## [250] "awfully"
## [253] "awfully"
## [256] "awfully"
## [259] "awfully"
## [262] "awfully"
## [265] "awfully"
## [268] "awfully"
## [271] "awfully"
## [274] "awfully"
## [277] "awfully"
## [280] "awfully"
## [283] "awfully"
## [286] "awfully"
## [289] "awfully"
## [292] "awfully"
## [295] "awfully"
## [298] "awfully"
## [301] "awfully"
## [304] "awfully"
## [307] "awfully"
## [310] "awfully"
## [313] "awfully"
## [316] "awfully"
## [319] "awfully"
## [322] "awfully"
## [325] "awfully"
## [328] "awfully"
## [331] "awfully"
## [334] "awfully"
## [337] "awfully"
## [340] "awfully"
## [343] "awfully"
## [346] "awfully"
## [349] "awfully"
## [352] "awfully"
## [355] "awfully"
## [358] "awfully"
## [361] "awfully"
## [364] "awfully"
## [367] "awfully"
## [370] "awfully"
## [373] "awfully"
## [376] "awfully"
## [379] "awfully"
## [382] "awfully"
## [385] "awfully"
## [388] "awfully"
## [391] "awfully"
## [394] "awfully"
## [397] "awfully"
## [400] "awfully"
## [403] "awfully"
## [406] "awfully"
## [409] "awfully"
## [412] "awfully"
## [415] "awfully"
## [418] "awfully"
## [421] "awfully"
## [424] "awfully"
## [427] "awfully"
## [430] "awfully"
## [433] "awfully"
## [436] "awfully"
## [439] "awfully"
## [442] "awfully"
## [445] "awfully"
## [448] "awfully"
## [451] "awfully"
## [454] "awfully"
## [457] "awfully"
## [460] "awfully"
## [463] "awfully"
## [466] "awfully"
## [469] "awfully"
## [472] "awfully"
## [475] "awfully"
## [478] "awfully"
## [481] "awfully"
## [484] "awfully"
## [487] "awfully"
## [490] "awfully"
## [493] "awfully"
## [496] "awfully"
## [499] "awfully"
## [502] "awfully"
## [505] "awfully"
## [508] "awfully"
## [511] "awfully"
## [514] "awfully"
## [517] "awfully"
## [520] "awfully"
## [523] "awfully"
## [526] "awfully"
## [529] "awfully"
## [532] "awfully"
## [535] "awfully"
## [538] "awfully"
## [541] "awfully"
## [544] "awfully"
## [547] "awfully"
## [550] "awfully"
## [553] "awfully"
## [556] "awfully"
## [559] "awfully"
## [562] "awfully"
## [565] "awfully"
## [568] "awfully"
## [571] "awfully"
## [574] "awfully"
## [577] "awfully"
## [580] "awfully"
## [583] "awfully"
## [586] "awfully"
## [589] "awfully"
## [592] "awfully"
## [595] "awfully"
## [598] "awfully"
## [601] "awfully"
## [604] "awfully"
## [607] "awfully"
## [610] "awfully"
## [613] "awfully"
## [616] "awfully"
## [619] "awfully"
## [622] "awfully"
## [625] "awfully"
## [628] "awfully"
## [631] "awfully"
## [634] "awfully"
## [637] "awfully"
## [640] "awfully"
## [643] "awfully"
## [646] "awfully"
## [649] "awfully"
## [652] "awfully"
## [655] "awfully"
## [658] "awfully"
## [661] "awfully"
## [664] "awfully"
## [667] "awfully"
## [670] "awfully"
## [673] "awfully"
## [676] "awfully"
## [679] "awfully"
## [682] "awfully"
## [685] "awfully"
## [688] "awfully"
## [691] "awfully"
## [694] "awfully"
## [697] "awfully"
## [700] "awfully"
## [703] "awfully"
## [706] "awfully"
## [709] "awfully"
## [712] "awfully"
## [715] "awfully"
## [718] "awfully"
## [721] "awfully"
## [724] "awfully"
## [727] "awfully"
## [730] "awfully"
## [733] "awfully"
## [736] "awfully"
## [739] "awfully"
## [742] "awfully"
## [745] "awfully"
## [748] "awfully"
## [751] "awfully"
## [754] "awfully"
## [757] "awfully"
## [760] "awfully"
## [763] "awfully"
## [766] "awfully"
## [769] "awfully"
## [772] "awfully"
## [775] "awfully"
## [778] "awfully"
## [781] "awfully"
## [784] "awfully"
## [787] "awfully"
## [790] "awfully"
## [793] "awfully"
## [796] "awfully"
## [799] "awfully"
## [802] "awfully"
## [805] "awfully"
## [808] "awfully"
## [811] "awfully"
## [814] "awfully"
## [817] "awfully"
## [820] "awfully"
## [823] "awfully"
## [826] "awfully"
## [829] "awfully"
## [832] "awfully"
## [835] "awfully"
## [838] "awfully"
## [841] "awfully"
## [844] "awfully"
## [847] "awfully"
## [850] "awfully"
## [853] "awfully"
## [856] "awfully"
## [859] "awfully"
## [862] "awfully"
## [865] "awfully"
## [868] "awfully"
## [871] "awfully"
## [874] "awfully"
## [877] "awfully"
## [880] "awfully"
## [883] "awfully"
## [886] "awfully"
## [889] "awfully"
## [892] "awfully"
## [895] "awfully"
## [898] "awfully"
## [901] "awfully"
## [904] "awfully"
## [907] "awfully"
## [910] "awfully"
## [913] "awfully"
## [916] "awfully"
## [919] "awfully"
## [922] "awfully"
## [925] "awfully"
## [928] "awfully"
## [931] "awfully"
## [934] "awfully"
## [937] "awfully"
## [940] "awfully"
## [943] "awfully"
## [946] "awfully"
## [949] "awfully"
## [952] "awfully"
## [955] "awfully"
## [958] "awfully"
## [961] "awfully"
## [964] "awfully"
## [967] "awfully"
## [970] "awfully"
## [973] "awfully"
## [976] "awfully"
## [979] "awfully"
## [982] "awfully"
## [985] "awfully"
## [988] "awfully"
## [991] "awfully"
## [994] "awfully"
## [997] "awfully"
## [1000] "awfully"

```

```

## [73] "back"          "bad"           "bag"
## [76] "bank"          "bare"          "basic"
## [79] "bath"          "batteri"       "bcoz"
## [82] "bday"          "beauti"        "becom"
## [85] "bed"           "bedroom"       "beer"
## [88] "begin"         "believ"        "best"
## [91] "better"         "bid"           "big"
## [94] "bill"          "bird"          "birthday"
## [97] "bit"           "black"         "blank"
## [100] "bless"         "blue"          "bluetooth"
## [103] "bold"          "bonus"         "boo"
## [106] "book"          "boost"         "bore"
## [109] "boss"          "bother"        "bout"
## [112] "box"           "boy"           "boytoy"
## [115] "break"         "breath"        "bring"
## [118] "brother"       "bslvyl"        "btnationalr"
## [121] "buck"          "bus"           "busi"
## [124] "buy"           "cabin"         "call"
## [127] "caller"        "callertun"     "camcord"
## [130] "came"          "camera"        "campus"
## [133] "can"           "cancel"        "cancer"
## [136] "cant"          "car"           "card"
## [139] "care"          "carlo"         "case"
## [142] "cash"          "cashbal"       "catch"
## [145] "caus"          "celebr"        "cell"
## [148] "centr"         "chanc"         "chang"
## [151] "charg"         "chat"          "cheap"
## [154] "cheaper"       "check"         "cheer"
## [157] "chennai"       "chikku"        "childish"
## [160] "children"      "choic"         "choos"
## [163] "christma"      "claim"         "class"
## [166] "clean"         "clear"         "close"
## [169] "club"          "code"          "coffe"
## [172] "cold"          "colleagu"      "collect"
## [175] "colleg"        "colour"        "come"
## [178] "comin"         "comp"          "compani"
## [181] "competit"      "complet"       "complimentari"
## [184] "comput"        "condit"        "confirm"
## [187] "congrat"       "congratul"     "connect"
## [190] "contact"       "content"       "contract"
## [193] "cook"          "cool"          "copi"
## [196] "correct"       "cos"           "cost"
## [199] "cost&pm"      "costa"         "coupl"
## [202] "cours"         "cover"         "coz"
## [205] "crave"         "crazi"         "creat"
## [208] "credit"        "cri"           "cross"
## [211] "cuddl"         "cum"           "cup"
## [214] "current"       "custcar"       "custom"
## [217] "cut"           "cute"          "cuz"
## [220] "dad"           "daddi"         "darl"
## [223] "darlin"        "darren"        "dat"
## [226] "date"          "day"           "dead"
## [229] "deal"          "dear"          "decid"
## [232] "decim"         "decis"         "deep"

```

```

## [235] "definit"
## [238] "deliveri"
## [241] "detail"
## [244] "diet"
## [247] "digit"
## [250] "direct"
## [253] "discuss"
## [256] "doc"
## [259] "dog"
## [262] "done"
## [265] "doubl"
## [268] "dream"
## [271] "drop"
## [274] "due"
## [277] "dvd"
## [280] "earth"
## [283] "eatin"
## [286] "els"
## [289] "end"
## [292] "enjoy"
## [295] "entitl"
## [298] "etc"
## [301] "even"
## [304] "everybodi"
## [307] "exact"
## [310] "excit"
## [313] "experi"
## [316] "eye"
## [319] "fact"
## [322] "fanci"
## [325] "far"
## [328] "father"
## [331] "feel"
## [334] "fight"
## [337] "fill"
## [340] "find"
## [343] "finish"
## [346] "flag"
## [349] "flower"
## [352] "food"
## [355] "forgot"
## [358] "freak"
## [361] "freephon"
## [364] "friday"
## [367] "frm"
## [370] "full"
## [373] "funni"
## [376] "game"
## [379] "gave"
## [382] "get"
## [385] "girl"
## [388] "glad"
## [391] "goin"
## [394] "good"
"del"
"den"
"didnt"
"iffer"
"din"
"dis"
"disturb"
"doctor"
"doin"
"dont"
"download"
"drink"
"drug"
"dun"
"earli"
"easi"
"egg"
"email"
"energi"
"enough"
"entri"
"euro"
"ever"
"everyon"
"exam"
"excus"
"expir"
"face"
"fall"
"fantasi"
"fast"
"fault"
"felt"
"figur"
"film"
"fine"
"first"
"flat"
"follow"
"forev"
"forward"
"free"
"fren"
"friend"
"frnd"
"fullonsmscom"
"futur"
"gap"
"gay"
"gettin"
"girlfrnd"
"god"
"gone"
"goodmorn"
"deliv"
"depend"
"die"
"difficult"
"dinner"
"discount"
"dnt"
"doesnt"
"don"
"door"
"draw"
"drive"
"dude"
"dunno"
"earlier"
"eat"
"either"
"embarass"
"england"
"enter"
"envelop"
"eve"
"everi"
"everyth"
"excel"
"expect"
"extra"
"facebook"
"famili"
"fantast"
"fat"
"feb"
"fetch"
"file"
"final"
"finger"
"fix"
"flight"
"fone"
"forget"
"found"
"freemsg"
"fri"
"friendship"
"frnds"
"fun"
"gal"
"gas"
"gentl"
"gift"
"give"
"goe"
"gonna"
"goodnight"

```

```

## [397] "got"
## [400] "great"
## [403] "gud"
## [406] "gym"
## [409] "hai"
## [412] "hand"
## [415] "happen"
## [418] "hate"
## [421] "head"
## [424] "heart"
## [427] "hell"
## [430] "hey"
## [433] "hiya"
## [436] "hmv"
## [439] "holder"
## [442] "hook"
## [445] "horni"
## [448] "hotel"
## [451] "how"
## [454] "hrs"
## [457] "huh"
## [460] "hurt"
## [463] "identifi"
## [466] "immedi"
## [469] "includ"
## [472] "inform"
## [475] "interest"
## [478] "irrit"
## [481] "issu"
## [484] "januari"
## [487] "john"
## [490] "joy"
## [493] "just"
## [496] "keep"
## [499] "kid"
## [502] "kinda"
## [505] "knew"
## [508] "ladi"
## [511] "laptop"
## [514] "late"
## [517] "laugh"
## [520] "lead"
## [523] "leav"
## [526] "leh"
## [529] "lesson"
## [532] "liao"
## [535] "life"
## [538] "like"
## [541] "list"
## [544] "live"
## [547] "loan"
## [550] "log"
## [553] "long"
## [556] "lookin"
"goto"
"grin"
"guess"
"haf"
"hair"
"handset"
"happi"
"hav"
"hear"
"heavi"
"hello"
"hgsuiteland"
"hm"
"hol"
"holiday"
"hop"
"hospit"
"hour"
"howev"
"httpwwwurawinnercom"
"hungri"
"ice"
"ignor"
"import"
"india"
"insid"
"invit"
"ish"
"ive"
"jay"
"join"
"jst"
"juz"
"kept"
"kill"
"king"
"know"
"land"
"lar"
"later"
"lazi"
"learn"
"lect"
"lei"
"let"
"librari"
"lift"
"line"
"listen"
"lmao"
"local"
"lol"
"longer"
"lor"
"gotta"
"guarante"
"guy"
"haha"
"half"
"hang"
"hard"
"havent"
"heard"
"hee"
"help"
"hit"
"hmmp"
"hold"
"home"
"hope"
"hot"
"hous"
"howz"
"hug"
"hurri"
"idea"
"ill"
"inc"
"info"
"instead"
"ipod"
"island"
"izzit"
"job"
"joke"
"jus"
"kate"
"kick"
"kind"
"kiss"
"knw"
"landlin"
"last"
"latest"
"ldn"
"least"
"left"
"less"
"letter"
"lie"
"light"
"link"
"littl"
"load"
"locat"
"london"
"look"
"lose"

```

```

## [559] "lost"          "lot"           "lovabl"
## [562] "love"          "lover"         "loyalti"
## [565] "ltd"           "luck"          "lucki"
## [568] "lunch"          "luv"           "mad"
## [571] "made"          "mah"           "mail"
## [574] "make"           "malaria"       "man"
## [577] "mani"          "march"         "mark"
## [580] "marri"          "match"         "mate"
## [583] "matter"         "maxim"         "maxmin"
## [586] "may"            "mayb"          "meal"
## [589] "mean"           "meant"         "med"
## [592] "medic"          "meet"          "meetin"
## [595] "meh"            "member"        "men"
## [598] "merri"          "messag"        "met"
## [601] "mid"            "midnight"      "mighth"
## [604] "min"            "mind"          "mine"
## [607] "minut"          "miracl"        "miss"
## [610] "mistak"         "moan"          "mob"
## [613] "mobil"          "mobileupd"     "mode"
## [616] "mom"            "moment"        "mon"
## [619] "monday"         "money"         "month"
## [622] "morn"           "mother"        "motorola"
## [625] "move"           "movi"          "mrng"
## [628] "mrt"            "mrw"           "msg"
## [631] "msgs"           "mths"          "much"
## [634] "mum"            "murder"        "music"
## [637] "must"           "muz"           "nah"
## [640] "nake"           "name"          "nation"
## [643] "natur"          "naughti"       "near"
## [646] "need"           "net"           "network"
## [649] "neva"           "never"         "new"
## [652] "news"           "next"          "nice"
## [655] "nigeria"        "night"         "nite"
## [658] "nobodi"         "noe"           "nokia"
## [661] "noon"           "nope"          "normal"
## [664] "normpton"       "noth"          "notic"
## [667] "now"            "num"           "number"
## [670] "nyt"            "obvious"       "offer"
## [673] "offic"          "offici"        "okay"
## [676] "oki"            "old"           "omg"
## [679] "one"             "onlin"         "onto"
## [682] "oop"             "open"          "oper"
## [685] "opinion"         "opt"           "optout"
## [688] "orang"          "orchard"       "order"
## [691] "oredi"           "oso"           "other"
## [694] "otherwis"        "outsid"        "pack"
## [697] "page"            "paid"          "pain"
## [700] "paper"          "parent"        "park"
## [703] "part"           "parti"         "partner"
## [706] "pass"           "passion"       "password"
## [709] "past"            "pay"           "peopl"
## [712] "per"             "person"        "pete"
## [715] "phone"          "photo"         "pic"
## [718] "pick"           "pictur"        "pin"

```

```

## [721] "piss"           "pix"          "pizza"
## [724] "place"          "plan"         "play"
## [727] "player"         "pleas"        "pleasur"
## [730] "plenti"         "pls"          "plus"
## [733] "plz"             "pmin"        "pmsg"
## [736] "pobox"          "point"        "poli"
## [739] "polic"          "poor"         "pop"
## [742] "possess"         "possibl"      "post"
## [745] "pound"           "power"        "ppm"
## [748] "pray"            "present"     "press"
## [751] "pretti"          "previous"    "price"
## [754] "princess"        "privat"       "prize"
## [757] "prob"            "probabl"     "problem"
## [760] "project"         "promis"       "pub"
## [763] "put"              "qualiti"    "question"
## [766] "quick"           "quit"         "quiz"
## [769] "quot"            "rain"         "random"
## [772] "rang"            "rate"         "rather"
## [775] "rcvd"            "reach"        "read"
## [778] "readi"           "real"         "reali"
## [781] "realli"          "reason"       "receipt"
## [784] "receiv"          "recent"       "record"
## [787] "refer"           "regard"       "regist"
## [790] "relat"           "relax"        "remain"
## [793] "rememb"          "remind"       "remov"
## [796] "rent"             "rental"       "repli"
## [799] "repres"          "request"     "respond"
## [802] "respons"         "rest"         "result"
## [805] "return"          "reveal"       "review"
## [808] "reward"          "right"        "ring"
## [811] "rington"         "rite"         "road"
## [814] "rock"             "role"         "room"
## [817] "roommat"         "rose"         "round"
## [820] "rowwjhl"         "rpli"         "rreveal"
## [823] "run"              "rush"         "sad"
## [826] "sae"              "safe"         "said"
## [829] "sale"             "sat"          "saturday"
## [832] "savamob"         "save"         "saw"
## [835] "say"              "sch"          "school"
## [838] "scream"          "sea"          "search"
## [841] "sec"              "second"      "secret"
## [844] "see"              "seem"         "seen"
## [847] "select"          "self"         "sell"
## [850] "semest"          "send"         "sens"
## [853] "sent"             "serious"     "servic"
## [856] "set"              "settл"        "sex"
## [859] "sexi"             "shall"        "share"
## [862] "shd"              "ship"         "shirt"
## [865] "shop"             "short"        "show"
## [868] "shower"          "sick"         "side"
## [871] "sigh"             "sight"        "sign"
## [874] "silent"          "simpl"        "sinc"
## [877] "singl"           "sipix"        "sir"
## [880] "sis"              "sister"      "sit"

```

```

## [883] "situat"           "skxh"          "skype"
## [886] "slave"             "sleep"          "slept"
## [889] "slow"              "slowli"         "small"
## [892] "smile"             "smoke"          "sms"
## [895] "smth"              "snow"           "sofa"
## [898] "sol"               "somebodi"       "someon"
## [901] "someth"            "sometim"        "somewher"
## [904] "song"               "soni"           "sonyericsson"
## [907] "soon"              "sorri"          "sort"
## [910] "sound"             "south"          "space"
## [913] "speak"             "special"         "specialcal"
## [916] "spend"             "spent"          "spoke"
## [919] "spree"             "stand"          "start"
## [922] "statement"         "station"         "stay"
## [925] "std"               "step"           "still"
## [928] "stockport"         "stone"          "stop"
## [931] "store"              "stori"          "street"
## [934] "student"            "studi"          "stuff"
## [937] "stupid"             "style"          "sub"
## [940] "subscrib"           "success"         "suck"
## [943] "suit"               "summer"         "sun"
## [946] "sunday"             "sunshin"        "sup"
## [949] "support"            "suppos"         "sure"
## [952] "surf"               "surpris"         "sweet"
## [955] "swing"              "system"         "take"
## [958] "talk"               "tampa"          "tariff"
## [961] "tcs"                "tea"            "teach"
## [964] "tear"               "teas"           "tel"
## [967] "tell"               "ten"            "tenerif"
## [970] "term"               "test"           "text"
## [973] "thank"              "thanx"          "that"
## [976] "thing"              "think"          "thinkin"
## [979] "thk"                "tho"            "though"
## [982] "thought"            "throw"          "thru"
## [985] "tht"                "thur"           "tick"
## [988] "ticket"             "til"            "till"
## [991] "time"               "tire"           "titl"
## [994] "tmr"                "toclaim"        "today"
## [997] "togeth"             "told"           "tomo"
## [1000] "tomorrow"          "tone"           "tonight"
## [1003] "tonit"              "took"           "top"
## [1006] "torch"              "tot"            "total"
## [1009] "touch"              "tough"          "tour"
## [1012] "toward"             "town"           "track"
## [1015] "train"              "transact"       "travel"
## [1018] "treat"              "tri"            "trip"
## [1021] "troubl"             "true"           "trust"
## [1024] "truth"              "tscs"           "ttyl"
## [1027] "tuesday"            "turn"           "twice"
## [1030] "two"                "txt"            "txting"
## [1033] "txts"              "type"           "ufind"
## [1036] "ugh"               "ull"            "uncl"
## [1039] "understand"        "unless"         "unlimit"
## [1042] "unredeem"           "unsub"          "unsubscrib"

```

```

## [1045] "updat"          "ure"           "urgent"
## [1048] "urself"          "use"            "user"
## [1051] "usf"             "usual"          "uve"
## [1054] "valentin"         "valid"          "valu"
## [1057] "via"              "video"          "vikki"
## [1060] "visit"            "vodafon"        "voic"
## [1063] "vomit"            "voucher"        "wait"
## [1066] "wake"             "walk"           "wan"
## [1069] "wana"             "wanna"          "want"
## [1072] "wap"              "warm"           "wast"
## [1075] "wat"              "watch"          "water"
## [1078] "way"              "weak"           "wear"
## [1081] "weather"          "wed"            "wednesday"
## [1084] "weed"              "week"           "weekend"
## [1087] "welcom"           "well"           "wen"
## [1090] "went"              "what"           "whatev"
## [1093] "whenev"            "whole"          "wid"
## [1096] "wif"               "wife"           "wil"
## [1099] "will"              "win"            "wine"
## [1102] "winner"            "wish"           "wit"
## [1105] "within"            "without"        "wiv"
## [1108] "wkli"              "wks"            "wnt"
## [1111] "woke"              "won"            "wonder"
## [1114] "wont"              "word"           "work"
## [1117] "workin"            "world"          "worri"
## [1120] "wors"              "worth"          "wot"
## [1123] "wow"               "write"          "wrong"
## [1126] "wwq"               "wwwgetzedcouk" "xmas"
## [1129] "xxx"               "yahoo"          "yar"
## [1132] "yeah"              "year"           "yep"
## [1135] "yes"               "yesterday"      "yet"
## [1138] "yoga"              "yup"            "yet"

# save frequently-appearing terms to a character vector
sms_freq_words <- findFreqTerms(sms_dtm_train, 5)
str(sms_freq_words)

## chr [1:1139] "£wk" "€~m" "€~s" "abiola" "abl" "abt" "accept" "access" ...
# create DTM's with only the frequent terms
sms_dtm_freq_train <- sms_dtm_train[ , sms_freq_words]
sms_dtm_freq_test <- sms_dtm_test[ , sms_freq_words]

# convert counts to a factor
convert_counts <- function(x) {
  x <- ifelse(x > 0, "Yes", "No")
}

# apply() convert_counts() to columns of train/test data
sms_train <- apply(sms_dtm_freq_train, MARGIN = 2, convert_counts)
sms_test <- apply(sms_dtm_freq_test, MARGIN = 2, convert_counts)

```

### Step 3: Training a model on the data —

```
library(e1071)
sms_classifier <- naiveBayes(sms_train, sms_train_labels)
```

### Step 4: Evaluating model performance —

```
sms_test_pred <- predict(sms_classifier, sms_test)

head(sms_test_pred)

## [1] ham ham ham ham spam ham
## Levels: ham spam

library(gmodels)
CrossTable(sms_test_pred, sms_test_labels,
           prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE,
           dnn = c('predicted', 'actual'))

##
##
##      Cell Contents
## |-----|
## |             N |
## |     N / Col Total |
## |-----|
## 
## 
## Total Observations in Table:  1390
##
##
##          | actual
##   predicted |    ham |    spam | Row Total |
## -----|-----|-----|-----|
##         ham |    1201 |      30 |    1231 |
##         |  0.995 |  0.164 |      |
## -----|-----|-----|-----|
##         spam |       6 |    153 |    159 |
##         |  0.005 |  0.836 |      |
## -----|-----|-----|-----|
## Column Total |    1207 |    183 |    1390 |
##         |  0.868 |  0.132 |      |
## -----|-----|-----|-----|
## 
##
```

### Step 5: Improving model performance —

```
sms_classifier2 <- naiveBayes(sms_train, sms_train_labels, laplace = 1)
sms_test_pred2 <- predict(sms_classifier2, sms_test)
CrossTable(sms_test_pred2, sms_test_labels,
```

```

prop.chisq = FALSE, prop.t = FALSE, prop.r = FALSE,
dnn = c('predicted', 'actual'))

##
##
##      Cell Contents
## |-----|
## |                   N |
## |             N / Col Total |
## |-----|
##
##
## Total Observations in Table:  1390
##
##
##           | actual
##   predicted |    ham |     spam | Row Total |
## -----|-----|-----|-----|
##       ham |    1202 |      28 |    1230 |
##           |  0.996 |  0.153 |      |
## -----|-----|-----|-----|
##       spam |      5 |    155 |    160 |
##           |  0.004 |  0.847 |      |
## -----|-----|-----|-----|
## Column Total |    1207 |    183 |    1390 |
##           |  0.868 |  0.132 |      |
## -----|-----|-----|-----|
##
##

```