NB

MENG

2018楠<9e>2閺<88><88>13閺<83>

### step1 download dataset [link]（<http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/>）

setwd('d:/rworkspace/ml')

### step2 探索和准备数据

读取数据到sms\_raw中

library(readr)  
  
sms\_raw <- read\_delim("D:/rworkspace/ml/SMSSpamCollection.txt", "\t", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## type = col\_character(),  
## text = col\_character()  
## )

str(sms\_raw)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 5572 obs. of 2 variables:  
## $ type: chr "ham" "ham" "spam" "ham" ...  
## $ text: chr "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat..." "Ok lar... Joking wif u oni..." "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question("| \_\_truncated\_\_ "U dun say so early hor... U c already then say..." ...  
## - attr(\*, "spec")=List of 2  
## ..$ cols :List of 2  
## .. ..$ type: list()  
## .. .. ..- attr(\*, "class")= chr "collector\_character" "collector"  
## .. ..$ text: list()  
## .. .. ..- attr(\*, "class")= chr "collector\_character" "collector"  
## ..$ default: list()  
## .. ..- attr(\*, "class")= chr "collector\_guess" "collector"  
## ..- attr(\*, "class")= chr "col\_spec"

将sms\_raw的类型从字符型变量改成因子型变量

sms\_raw$type <- factor(sms\_raw$type)  
str(sms\_raw$type)

## Factor w/ 2 levels "ham","spam": 1 1 2 1 1 2 1 1 2 2 ...

用table查看type的数据，spam占比约13.4%

table(sms\_raw$type)

##   
## ham spam   
## 4825 747

#### 处理和分析文本数据

Corpus():创建了一个R对象来存储文本文档 VectorsSource():指示Corpus()函数使用向量sms\_raw$text

library(tm)

## Loading required package: NLP

sms\_corpus <- Corpus(VectorSource(sms\_raw$text))  
print(sms\_corpus)

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 5572

inspect():查看语料库的内容

inspect(sms\_corpus[1:3])

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 3  
##   
## [1] Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...   
## [2] Ok lar... Joking wif u oni...   
## [3] Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's

change upper letter to lower letter remove numbers remove stopwors() remove punctuation remove space between 2 words

corpus\_clean = tm\_map(sms\_corpus, tolower)  
corpus\_clean = tm\_map(corpus\_clean, removeNumbers)  
corpus\_clean = tm\_map(corpus\_clean, removeWords, stopwords())  
corpus\_clean = tm\_map(corpus\_clean, removePunctuation)  
corpus\_clean = tm\_map(corpus\_clean, stripWhitespace)  
inspect(corpus\_clean[1:3])

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 3  
##   
## [1] go jurong point crazy available bugis n great world la e buffet cine got amore wat   
## [2] ok lar joking wif u oni   
## [3] free entry wkly comp win fa cup final tkts st may text fa receive entry questionstd txt ratetcs apply s

通过**标记化**过程将消息分解成单个单词组成的组 一个记号（token）就是一个文本字符串的单个元素，本例中token就是单词 DocumentTermMatrix(): 将一个语料库作为输入，并创建一个**稀疏矩阵**的数据结构

# 创建稀疏矩阵  
sms\_dtm = DocumentTermMatrix(corpus\_clean)

1. 建立测试和训练数据集

sms\_raw\_train <- sms\_raw[1:4825, ]  
sms\_raw\_test <- sms\_raw[4826:5572, ]

输入文档-单词矩阵

sms\_dtm\_train <- sms\_dtm[1:4825, ]  
sms\_dtm\_test <- sms\_dtm[4826:5572, ]

最终语料库

sms\_corpus\_train <- corpus\_clean[1:4825 ]  
sms\_corpus\_test <- corpus\_clean[4826:5572 ]

ham和spam在test和train中占比差不多，说明分配比较均匀

prop.table(table(sms\_raw\_train$type))

##   
## ham spam   
## 0.8654922 0.1345078

prop.table(table(sms\_raw\_test$type))

##   
## ham spam   
## 0.8688086 0.1311914

1. 可视化文本数据-词云 random.order=F: 顺序排列，词频越高，越接近中心 min.freq=40: 在至少40条短信中出现过

library('wordcloud')

## Loading required package: RColorBrewer

wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F)

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## contact could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## someone could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## even could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## money could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## gud could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## per could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## many could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## friends could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## customer could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## service could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## gonna could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## help could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## dun could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## wan could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## tonight could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## sleep could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## getting could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## box could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## told could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## may could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## name could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## done could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## waiting could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## wish could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## live could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## year could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## things could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## haha could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## class could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## special could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## leave could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## man could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## best could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## people could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## lunch could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## guaranteed could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## coming could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## mins could not be fit on page. It will not be plotted.

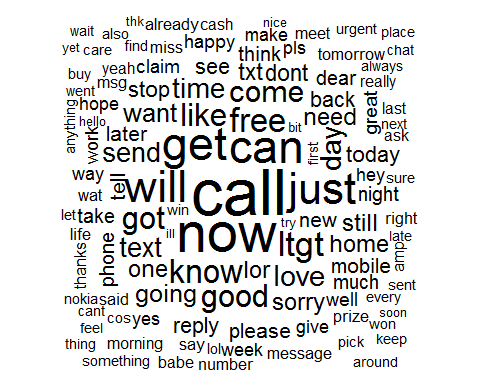
## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## days could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## end could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## ppm could not be fit on page. It will not be plotted.

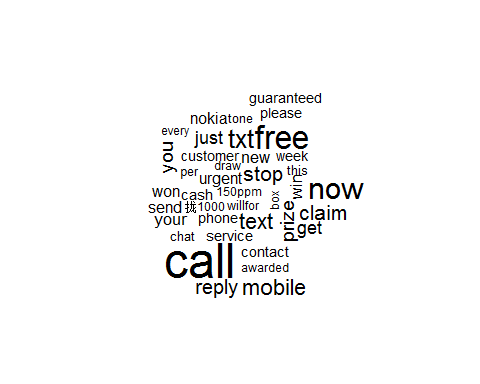
## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## holiday could not be fit on page. It will not be plotted.

## Warning in wordcloud(sms\_corpus\_train, min.freq = 40, random.order = F):  
## tone could not be fit on page. It will not be plotted.

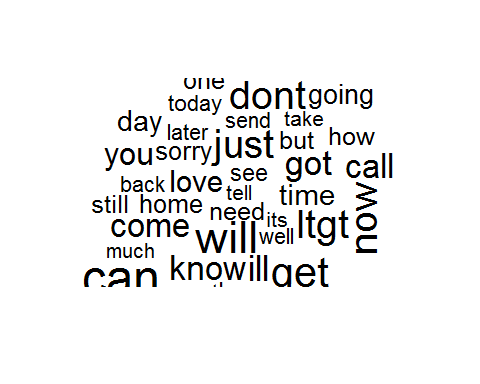


在训练集中，对标签为spam和ham分别做图，进行比较。spam中，比较多的是free，prize，推测是垃圾信息

spam = subset(sms\_raw\_train, type == 'spam')  
ham = subset(sms\_raw\_train, type == 'ham')  
wordcloud(spam$text, max.words = 40, scale = c(3, 0.5))



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

 3. 为频繁出现的单词创建指示特征 findFreqTerms(): 输入**文档-单词矩阵**， 返回一个字符向量

# 参数5表示该向量中断的单词在矩阵中至少出现5次，类型是character  
sms\_dict = findFreqTerms(sms\_dtm\_train, 5)  
  
#把数据集中所有频率超过5次的单词保存成新的数据集  
sms\_train = DocumentTermMatrix(sms\_corpus\_train, list(dictionary = sms\_dict))  
sms\_test = DocumentTermMatrix(sms\_corpus\_test, list(dictionary = sms\_dict))

朴素贝叶斯分类器通常是训练具有明确特征的数据。 因为稀疏矩阵中的元素表示一个单词出现的次数，我们需要将其改变成因子变量，根据单词是否出现，简单的表示为yes或者no

# 自定义函数convert\_counts()  
convert\_counts = function(x){   
 x = ifelse(x>0, 1, 0) #x > 0, 用1替换，否则用0替换  
 x = factor(x, levels = c(0, 1), labels = c('"No"', '"Yes"'))  
 return(x)  
 }

将convert\_counts()应用到每列稀疏矩阵。 apply()来实现，而非for或者while循环

#MARGIN = 1 按行，MARGIN = 2按列  
sms\_train <- apply(sms\_train, MARGIN = 2, convert\_counts)  
sms\_test <- apply(sms\_test, MARGIN = 2, convert\_counts)

### step3 基于数据训练模型

library(e1071)  
sms\_classifier = naiveBayes(sms\_train, sms\_raw\_train$type)  
#class(sms\_train)  
#class(sms\_classifier)  
#class(sms\_raw\_train$type)

### step4 评估模型的性能

sms\_test\_pred = predict(sms\_classifier, sms\_test)  
class(sms\_test\_pred)

## [1] "factor"

library(gmodels)  
CrossTable(sms\_test\_pred, sms\_raw\_test$type,prop.chisq = F, prop.t = F, dnn = c('predicted', 'actual'))

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 747   
##   
##   
## | actual   
## predicted | ham | spam | Row Total |   
## -------------|-----------|-----------|-----------|  
## ham | 647 | 12 | 659 |   
## | 0.982 | 0.018 | 0.882 |   
## | 0.997 | 0.122 | |   
## -------------|-----------|-----------|-----------|  
## spam | 2 | 86 | 88 |   
## | 0.023 | 0.977 | 0.118 |   
## | 0.003 | 0.878 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 649 | 98 | 747 |   
## | 0.869 | 0.131 | |   
## -------------|-----------|-----------|-----------|  
##   
##

### step5 提升模型性能

sms\_classifier2 = naiveBayes(sms\_train, sms\_raw\_train$type, laplace = 1)  
sms\_test\_pred2 = predict(sms\_classifier2, sms\_test)  
CrossTable(sms\_test\_pred2, sms\_raw\_test$type, prop.chisq = F, prop.t = F, prop.r=F, dnn = c('predicted','test'))

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Col Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 747   
##   
##   
## | test   
## predicted | ham | spam | Row Total |   
## -------------|-----------|-----------|-----------|  
## ham | 647 | 17 | 664 |   
## | 0.997 | 0.173 | |   
## -------------|-----------|-----------|-----------|  
## spam | 2 | 81 | 83 |   
## | 0.003 | 0.827 | |   
## -------------|-----------|-----------|-----------|  
## Column Total | 649 | 98 | 747 |   
## | 0.869 | 0.131 | |   
## -------------|-----------|-----------|-----------|  
##   
##