# Lab 4

In [1]:
```
library(tm)
library(e1071)
```

```
Loading required package: NLP
```

In [2]:
```
sms_raw <-read.csv("F:/dataMining600/Lab4/sms_spam.csv",stringsAsFactors = FALSE)
print(sms_raw$type[239])
### this is important to the " duplicated row.names" error
#colnames(sms_raw) <- c(colnames(sms_raw)[-1],"x")
#sms_raw$x <- NULL
###
head(sms_raw)
print(dim(sms_raw))
sms_raw$type[239]
```

```
[1] "ham"
```

| type | text |
|---|---|
| ham | Hope you are having a good week. Just checking in |
| ham | K..give back my thanks. |
| ham | Am also doing in cbe only. But have to pay. |
| spam | complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your URGENT collection. 09066364349 NOW from Landline not to lose out! Box434SK38WP150PPM18+ |
| spam | okmail: Dear Dave this is your final notice to collect your 4* Tenerife Holiday or #5000 CASH award! Call 09061743806 from landline. TCs SAE Box326 CW25WX 150ppm |
| ham | Aiya we discuss later lar... Pick u up at 4 is it? |

```
[1] 5559    2
```

'ham'

In [3]:
```
sms_corpus <- VCorpus(VectorSource(sms_raw$text))
sms_corpus

## the following code is to clean the text by delete all the non-ascii charact
ers
for(i in 1:length(sms_raw$text)){
     sms_raw$text[i] = iconv(sms_raw$text[i], "latin1", "ASCII", sub="")
}

## have to remvoe all the non-ascii chars before excuting the following:
sms_corpus_clean <- tm_map(sms_corpus, removeNumbers)
sms_corpus_clean <- tm_map(sms_corpus_clean,removePunctuation)
sms_corpus_clean <- tm_map(sms_corpus_clean,content_transformer(tolower))
```

```
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 5559
```

In [11]:
```
sms_dtm <- DocumentTermMatrix(sms_corpus_clean)
inspect(sms_dtm)
print (sms_corpus_clean[1:4])
```

```
<<DocumentTermMatrix (documents: 5559, terms: 8390)>>
Non-/sparse entries: 57560/46582450
Sparsity           : 100%
Maximal term length: 40
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   and are call for have now that the you your
  1628   4   3    0   1    0   0    1   2   5    0
  2046   4   0    0   0    4   0    2  10  13    0
  2993   3   1    0   2    2   0    1   2   4    1
  313    4   0    0   6    1   0    2   8   0    0
  3522   1   0    0   1    0   0    0   2   0    0
  399    1   0    0   1    0   0    0   2   0    0
  4493   1   0    0   2    0   0    2   2   5    0
  5279   0   0    0   3    1   0    1   1   0    0
  64     1   0    0   0    0   0    1   3   4    0
  808    1   0    0   1    3   0    1   2   3    0
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 4
```

In [23]:
```
train_size <- nrow(sms_dtm)*0.8
test_size <- nrow(sms_dtm)*0.2
print(train_size)
print(test_size)

sms_dtm_train <- sms_dtm[1:train_size, ]
sms_dtm_test <- sms_dtm[train_size+1:test_size, ]
inspect(sms_dtm)

sms_train_labels = sms_raw$type[1:train_size]
sms_test_labels = sms_raw$type[train_size+1:test_size]
```

```
[1] 4447.2
[1] 1111.8
<<DocumentTermMatrix (documents: 5559, terms: 8390)>>
Non-/sparse entries: 57560/46582450
Sparsity           : 100%
Maximal term length: 40
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   and are call for have now that the you your
  1628   4   3    0   1    0   0    1   2   5    0
  2046   4   0    0   0    4   0    2  10  13    0
  2993   3   1    0   2    2   0    1   2   4    1
  313    4   0    0   6    1   0    2   8   0    0
  3522   1   0    0   1    0   0    0   2   0    0
  399    1   0    0   1    0   0    0   2   0    0
  4493   1   0    0   2    0   0    2   2   5    0
  5279   0   0    0   3    1   0    1   1   0    0
  64     1   0    0   0    0   0    1   3   4    0
  808    1   0    0   1    3   0    1   2   3    0
```

In [24]:
```
freq_terms <- findFreqTerms(sms_dtm_train, 10)
# create DTMs with only the frequent terms
sms_dtm_freq_train <- sms_dtm_train[ , freq_terms]
sms_dtm_freq_test <- sms_dtm_test[ , freq_terms]
```

In [25]:
```
inspect(sms_dtm_freq_train)
```

```
<<DocumentTermMatrix (documents: 4447, terms: 763)>>
Non-/sparse entries: 33434/3359627
Sparsity           : 99%
Maximal term length: 15
Weighting          : term frequency (tf)
Sample             :
      Terms
Docs   and are call for have now that the you your
  1613   3   0    0   0    0   0    3   3   8    0
  1628   4   3    0   1    0   0    1   2   5    0
  2046   4   0    0   0    4   0    2  10  13    0
  2993   3   1    0   2    2   0    1   2   4    1
  313    4   0    0   6    1   0    2   8   0    0
  3522   1   0    0   1    0   0    0   2   0    0
  3854   1   1    0   0    0   0    1   3   4    0
  399    1   0    0   1    0   0    0   2   0    0
  64     1   0    0   0    0   0    1   3   4    0
  808    1   0    0   1    3   0    1   2   3    0
```

In [27]:
```
convert_counts <- function(x)
{
x <- ifelse(x > 0, "Yes", "No")
}
sms_train <- apply(sms_dtm_freq_train, MARGIN = 2,convert_counts)
sms_test <- apply(sms_dtm_freq_test, MARGIN = 2,convert_counts)

sms_classifier <- naiveBayes(as.matrix(sms_train),as.factor(sms_train_labels))
```

In [28]:
```
sms_test_pred <- predict(sms_classifier,as.matrix(sms_test))
```

In [29]:
```
table("Predicted" = sms_test_pred, "Actual" = sms_test_labels)
```

```
         Actual
Predicted ham spam
     ham  956   22
     spam   5  128
```

In [ ]:

In [ ]: