

```
install.packages("tm")
install.packages("wordcloud")
install.packages("e1071")
library(tm)
library(wordcloud)
library(e1071)

sms_spam_df <- read.csv(file="C:\\Users\\Hp\\Desktop\\sms_spam.csv",stringsAsFactors=F)
View(sms_spam_df)

str(sms_spam_df)
#head(sms_spam_df)
#table(sms_spam_df$category)

sms_corpus <- VCorpus(VectorSource(sms_spam_df$text))
#sms_corpus
print(sms_corpus)

inspect(sms_corpus[1:2])
#sms_corpus[1:2]

clean_corpus <- tm_map(sms_corpus, content_transformer(tolower))
#clean_corpus <- tm_map(sms_corpus, tolower)
#as.character(clean_corpus[[1]])

clean_corpus <- tm_map(clean_corpus, removeNumbers)

clean_corpus <- tm_map(clean_corpus, removePunctuation)

stopwords()[1:15]
clean_corpus <- tm_map(clean_corpus, removeWords, stopwords())
```

```
#as.character(clean_corpus[[1]])

clean_corpus <- tm_map(clean_corpus, stripWhitespace)

inspect(clean_corpus[1:3])

sms_dtm <- DocumentTermMatrix(clean_corpus)

str(sms_dtm)

spam_indices <- which(sms_spam_df$category == "spam")
ham_indices <- which(sms_spam_df$category == "ham")

#spam_indices
#ham_indices

wordcloud(clean_corpus[ham_indices], min.freq=40)
wordcloud(clean_corpus[spam_indices], min.freq=40)

sms_raw_train <- sms_spam_df[1:4169,]
sms_raw_test <- sms_spam_df[4170:5559,]

sms_dtm_train <- sms_dtm[1:4169,]
sms_dtm_test <- sms_dtm[4170:5559,]

sms_corpus_train <- clean_corpus[1:4169]
sms_corpus_test <- clean_corpus[4170:5559]

spam <- subset(sms_raw_train, category == "spam")
ham <- subset(sms_raw_train, category == "ham")
```

```
five_times_words <- findFreqTerms(sms_dtm_train, 5)
```

```
sms_train <- DocumentTermMatrix(sms_corpus_train, control=list(dictionary = five_times_words))
```

```
sms_test <- DocumentTermMatrix(sms_corpus_test, control=list(dictionary = five_times_words))
```

```
convert_count <- function(x) {
```

```
  y <- ifelse(x > 0, 1, 0)
```

```
  y <- factor(y, levels=c(0,1), labels=c("No", "Yes"))
```

```
  y
```

```
}
```

```
sms_train <- apply(sms_train, 2, convert_count)
```

```
sms_test <- apply(sms_test, 2, convert_count)
```

```
sms_classifier <- naiveBayes(sms_train, factor(sms_raw_train$category))
```

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

```
k=table(sms_test_pred, sms_raw_test$category)
```

```
k
```

```
accuracy = sum(diag(k))/sum(k)*100
```

```
accuracy
```