Week 3 Project

Kevin McBeth

January 29, 2018

Introduction

Naive Bayes classification methods leverage Bayes Law, a mathematical relationship indicating that one can find the probability of a given event occuring if another event occured, based on the probability of that other event occuring given the event and the probability of each occuring in general. This project was to create a simple spam detector, identifying whether or not an item is spam based upon it's feature vector.

On my machine, the predictor had it's best accuracy when classifying everything as not spam. This indicates that the features either were not well defined or large enough in number to give good predictions based upon the feature vector. I attempted to clean up the data using the raw text and the packages "tm" and "dplyr" to create the vectors into words.

This was mostly done because of an error requiring the names to be substituted, as they included punctuations.

```
## Loading required package: MASS

## Loading required package: lattice

## Loading required package: ggplot2

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate

##
## Attaching package: 'naivebayes'

## The following object is masked from 'package:data.table':
##
##     tables
```

```
## Classes 'data.table' and 'data.frame':   5572 obs. of  5 variables:
##  $ v1: chr  "ham" "ham" "spam" "ham" ...
##  $ v2: chr  "Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cin
##  $ V3: chr  "" "" "" "" ...
##  $ V4: chr  "" "" "" "" ...
##  $ V5: chr  "" "" "" "" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Cleaning

First, I removed the empty columns from the raw csv file. Then, I added column names, and converted the classifier to a factor. Following this, I used the Corpus function (from TM) and the pipe operator to convert all words to lowercase, remove numbers, remove stop words, remove punctuation, and strip whitespaces. At the conclusion, I created a DocumentTermMatrix object from this text. Note that this code, and some of the following code comes from the following source: https://rpubs.com/jesuscastagnetto/caret-naive-bayes-spam-ham-sms

Data Partioning

Following this work, I partioned the data into test and train sets, using the corpus objects and document term matrix options in addition to the raw values. Then, using guidance, created a pander library table to show the distribution of classifiers

```r
train.indices <- createDataPartition(full$type, p = 0.8, list=FALSE)
raw_train <- full[train.indices,]
raw_test <- full[-train.indices,]
corpus_train <- full_corpus_clean[train.indices]
corpus_test <- full_corpus_clean[-train.indices]
dtm_train <- full_dtm[train.indices,]
dtm_test <- full_dtm[-train.indices,]

frqtab <- function(x, caption) {
    round(100*prop.table(table(x)), 1)
}

ft_orig <- frqtab(full$type)
ft_train <- frqtab(raw_train$type)
ft_test <- frqtab(raw_test$type)
ft_df <- as.data.frame(cbind(ft_orig, ft_train, ft_test))
colnames(ft_df) <- c("Original", "Training set", "Test set")
pander(ft_df, style="rmarkdown",
       caption=paste0("Comparison of SMS type frequencies among datasets"))
```

Table 1: Comparison of SMS type frequencies among datasets ##Model Fitting To continue the project, I used a recommended text dictionary to find all the elements in the training set which occurred five times. This narrows the list of words to more common words, and make it such that the data set is not too empty. Here, the defined apply function is returning a list of values of counts by applying a function to the columns of the Document Term Matrix object. An issue with the caret function is that non basic characters come up as zero vectors, which fails the model. To fix this, I found which item was the issue, and then removed it from the training test set.

	Original	Training set	Test set
ham	86.6	86.6	86.6
spam	13.4	13.4	13.4

Using R markdown was difficult in this area, because, of buffer overflows due to the excessive amount of warnings written.

```
## callsÃ¥
##    1289
```

```
##                nah              std            melle
##               1107             1108             1109
##          searching            stock              egg
##               1110             1111             1112
##                tea        hopefully             kept
##               1113             1114             1115
##               weak            liked              ice
##               1116             1117             1118
##                red             song            plane
##               1119             1120             1121
##             simply           wishes            eatin
##               1122             1123             1124
##                bak              dvd         sunshine
##               1125             1126             1127
##              rooms          replied            dream
##               1128             1129             1130
##            arrange            waste          discuss
##               1131             1132             1133
##              smoke           doesnt           joined
##               1134             1135             1136
##              touch             kate       connection
##               1137             1138             1139
##           semester         romantic         wwwtcbiz
##               1140             1141             1142
##         appreciate          lessons             died
##               1143             1144             1145
##             laptop          website              cal
##               1146             1147             1148
##                wks          disturb            swing
##               1149             1150             1151
##              sense             high            babes
##               1152             1153             1154
```

```
##        selection     christmas        access
##             1155          1156          1157
##              via       surfing           num
##             1158          1159          1160
##        basically        urself           ago
##             1161          1162          1163
##        insurance        posted           air
##             1164          1165          1166
##        bluetooth  sonyericsson           gbp
##             1167          1168          1169
##           brings       mistake         guide
##             1170          1171          1172
##             slow       current      facebook
##             1173          1174          1175
##         pictures       putting  fullonsmscom
##             1176          1177          1178
##         poboxwwq        hostel       respond
##             1179          1180          1181
##             vary        ticket       dollars
##             1182          1183          1184
##            group          loan         gives
##             1185          1186          1187
##          charges       depends        within
##             1188          1189          1190
##           hoping       married       english
##             1191          1192          1193
##            cheap        barely        smiles
##             1194          1195          1196
##         marriage          kids  announcement
##             1197          1198          1199
##          present       stopped        ladies
##             1200          1201          1202
##         somethin         daily       results
##             1203          1204          1205
##        valentine        drinks          paid
##             1206          1207          1208
##             area        gentle         earth
##             1209          1210          1211
##          bedroom          bold         torch
##             1212          1213          1214
##              law           wer        tonite
##             1215          1216          1217
##            realy         holla    polyphonic
##             1218          1219          1220
##         lttimegt     normptone           wwq
##             1221          1222          1223
##     wwwgetzedcouk           bid       charity
##             1224          1225          1226
##            polys           wed           cds
##             1227          1228          1229
##           matter        travel           mid
##             1230          1231          1232
##         midnight         teach       cheaper
##             1233          1234          1235
```

```
##                 gay              places               heavy
##                1236                1237                1238
##               brand                 ull                site
##                1239                1240                1241
##              asleep                hiya              flower
##                1242                1243                1244
##            revealed              convey             regards
##                1245                1246                1247
##             digital               sipix            awaiting
##                1248                1249                1250
## httpwwwurawinnercom                onto             italian
##                1251                1252                1253
##              energy              choice            complete
##                1254                1255                1256
##             callsÃ¥                tour             meaning
##                1257                1258                1259
##               tells             totally           difficult
##                1260                1261                1262
##              london                buzz              inside
##                1263                1264                1265
##               sight                doin              social
##                1266                1267                1268
##              slowly             selling                buns
##                1269                1270                1271
##              arcade                list               moral
##                1272                1273                1274
##           obviously               boost               caken
##                1275                1276                1277
##                latr              minuts                mood
##                1278                1279                1280
##               empty              remind           ringtones
##                1281                1282                1283
##                 sky              budget              random
##                1284                1285                1286
##             deliver              ignore              common
##                1287                1288                1289
```

```
## Naive Bayes
##
## 4458 samples
## 1288 predictors
##    2 classes: 'ham', 'spam'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 4458, 4458, 4458, 4458, 4458, 4458, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9804872  0.9129294
##    TRUE      0.9804872  0.9129294
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
```

```
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE
##  and adjust = 1.
```

Confusion Matrix

The final portion of this spam classifier is to look at model performance. To do this, we use a confusion matrix on the test set. Looking below, see a test accuracy of 98% and a training accuracy of 79%, with only a small portion of spam being correctly classified.

```r
confusionMatrix(predict(model1, text_train), raw_train$type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  ham spam
##       ham  3849   64
##       spam   11  534
##
##                Accuracy : 0.9832
##                  95% CI : (0.979, 0.9867)
##     No Information Rate : 0.8659
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9248
##  Mcnemar's Test P-Value : 1.92e-09
##
##             Sensitivity : 0.9972
##             Specificity : 0.8930
##          Pos Pred Value : 0.9836
##          Neg Pred Value : 0.9798
##              Prevalence : 0.8659
##          Detection Rate : 0.8634
##    Detection Prevalence : 0.8777
##       Balanced Accuracy : 0.9451
##
##        'Positive' Class : ham
##
```

```r
confusionMatrix(predict(model1, text_test), raw_test$type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ham spam
##       ham  846  119
##       spam 119   30
##
##                Accuracy : 0.7864
##                  95% CI : (0.7611, 0.8101)
##     No Information Rate : 0.8662
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.078
```

```
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8767
##             Specificity : 0.2013
##          Pos Pred Value : 0.8767
##          Neg Pred Value : 0.2013
##              Prevalence : 0.8662
##          Detection Rate : 0.7594
##    Detection Prevalence : 0.8662
##       Balanced Accuracy : 0.5390
##
##        'Positive' Class : ham
##
```