Project Proposal

Angel Chen (5344999) Cindy Wong (5050422) Jalen Souksamlane (7451404)

February 13, 2020

Research Question

Is there a relationship between the predictors (frequency of characters, length of email, certain common key words, etc.) and whether an email is considered spam or not? If so, which predictors affect the response?

It is important in constructing algorithms for email systems that automatically consider certain emails as spam; if there is direct correlation between the predictors and the response, it would be more convenient for email users in a way that their spam emails will be filtered out before reaching them.

Why is it important? What is already known?

It is known that while there is a pattern of indicators of spam and non-spam email, and that it is possible to construct personalized spam filters based on each email user's habits, there could be false positives (non-spam emails considered spam) which is undesirable.

Data

Which is your response variable?

Spam type (0 indicates non-spam email, 1 indicates spam email)

How many predictor variables are there? How many of each type (e.g. numeric, categorical, binary).

There are 57 predictor variables, they are all numeric variables since the predictors are mainly frequency of words/characters and length.

Analysis Plan

Since there are a lot of variables, we plan to find the mean and median frequencies on the variables that seem more important to our model. To decide whether a predictor has a large effect on the response, we can first do some exploratory analysis. We can include scatterplots and boxplots to see the effect each predictor has on the response, then we can note the predictors that seem to have a large effect. For this dataset, we will be doing supervised machine learning using logistic regression, decision trees, and K nearest neighbors. To train our classifier, we can do cross-validation. We plan to hold-out half of the observations

References

Mark Hopkins, Erik Reeber, George Forman, and Jaap Suermondt of Hewlett-Packard Labs (1999). Spambase Data Set. http://archive.ics.uci.edu/ml/datasets/Spambase

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Data Overview

```
## Read in data
# adding column names to the data
names <- c('make','address','all','3d','our','over','remove','internet','order','mail','receive',
           'will','people','report','addresses','free','business','email','you','credit','your',
           'font','000','money','hp','hpl','george','650','lab','labs','telnet','857','data','415',
           '85','technology','1999','parts','pm','direct','cs','meeting','original','project','re',
           'edu','table','conference','semi.colon','parenthesis','bracket','exclamation','dollar.sign',
           'capital_run_length_average','capital_run_length_longest','capital_run_length_total','spam')

data <- read.table('~/Documents/PSTAT131/Project/spambase.data', sep = ',', col.names = names)

dim(data)
```

```
## [1] 4601   58
```

```
## Missingness
apply(is.na(data), 2, sum)
```

```
##                        make                     address
##                           0                           0
##                         all                        X3d
##                           0                           0
##                         our                        over
##                           0                           0
##                      remove                    internet
##                           0                           0
##                       order                        mail
##                           0                           0
##                     receive                        will
##                           0                           0
##                      people                      report
##                           0                           0
##                   addresses                        free
##                           0                           0
##                    business                       email
##                           0                           0
##                         you                      credit
##                           0                           0
##                        your                        font
##                           0                           0
##                        X000                       money
##                           0                           0
##                          hp                         hpl
##                           0                           0
##                      george                        X650
##                           0                           0
##                         lab                        labs
##                           0                           0
##                      telnet                        X857
##                           0                           0
##                        data                        X415
##                           0                           0
##                         X85                  technology
```

```
##                         0                         0
##                      X1999                     parts
##                         0                         0
##                        pm                    direct
##                         0                         0
##                        cs                   meeting
##                         0                         0
##                  original                   project
##                         0                         0
##                        re                       edu
##                         0                         0
##                     table                conference
##                         0                         0
##                 semi.colon                parenthesis
##                         0                         0
##                    bracket                exclamation
##                         0                         0
##                dollar.sign                     pound
##                         0                         0
## capital_run_length_average capital_run_length_longest
##                         0                         0
##   capital_run_length_total                      spam
##                         0                         0
```

```
# it appears that there isn't any missing observations

## Split the data
set.seed(1)
# sample 50% of observations as training data
data.sample <- sample(1:nrow(data), 0.50*nrow(data))
train <- data[data.sample,]
dim(train)
```

```
## [1] 2300   58
```

```
# the other 50% as test data
test <- data[-data.sample,]
dim(test)
```

```
## [1] 2301   58
```

```
## Fitting the training set with two variables
fit.train <- glm(spam~remove+free, data = data, family = 'binomial')
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fit.train)
```

```
##
## Call:
## glm(formula = spam ~ remove + free, family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.7594  -0.7594   0.7919   1.6639
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.09593    0.03916  -27.98   <2e-16 ***
## remove       6.54643    0.42235   15.50   <2e-16 ***
## free         1.58308    0.10545   15.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6170.2  on 4600  degrees of freedom
## Residual deviance: 4828.9  on 4598  degrees of freedom
## AIC: 4834.9
##
## Number of Fisher Scoring iterations: 7
```

```
## k-NN
set.seed(2)
# creating response vector for training set
y.train <- train$spam
# creating design matrix for training set with two variables
x.train <- train %>% select(-spam)
x.train <- scale(x.train,center=TRUE,scale=TRUE)
# creating response vector and design matrix for test set
meanvec <- attr(x.train,'scaled:center')
sdvec <- attr(x.train,'scaled:scale')
y.test <- test$spam
x.test <- test %>% select(-spam) %>% scale(center=meanvec,scale=sdvec)
# training the classifier and making predictions on the training set
pred.y.train <- knn(train=x.train,test=x.train,cl=y.train,k=5)
# calculating the confusion matrix
conf.train <- table(predicted=pred.y.train,observed=y.train)
conf.train
```

```
##          observed
## predicted    0    1
##         0 1313  107
##         1   60  820
```

```
# train accuracy rate
sum(diag(conf.train)/sum(conf.train))
```

```
## [1] 0.9273913
```