Spam Email Prediction

Yusi Chen

24/06/2020

Project Purpose

Tocreate a predictive model to evaluate whether an email is spam or not.

Language

R

Model

Logistic Regression

Preparing data

```r
install.packages("caTools")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
install.packages("questionr")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
install.packages("car")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```r
library(caTools)
library(questionr)
library(car)
```

```
## Loading required package: carData
```

```r
spam<-read.csv('/cloud/project/spam7.csv')
```

```r
describe(spam)
```

```
## [4601 obs. x 8 variables] tbl_df tbl data.frame
##
## $X:
## integer: 1 2 3 4 5 6 7 8 9 10 ...
```

```
## min: 1 - max: 4601 - NAs: 0 (0%) - 4601 unique values
##
## $crl.tot:
## integer: 278 1028 2259 191 191 54 112 49 1257 749 ...
## min: 1 - max: 15841 - NAs: 0 (0%) - 919 unique values
##
## $dollar:
## numeric: 0 0.18 0.184 0 0 0 0.054 0 0.203 0.081 ...
## min: 0 - max: 6.003 - NAs: 0 (0%) - 504 unique values
##
## $bang:
## numeric: 0.778 0.372 0.276 0.137 0.135 0 0.164 0 0.181 0.244 ...
## min: 0 - max: 32.478 - NAs: 0 (0%) - 964 unique values
##
## $money:
## numeric: 0 0.43 0.06 0 0 0 0 0 0.15 0 ...
## min: 0 - max: 12.5 - NAs: 0 (0%) - 143 unique values
##
## $n000:
## numeric: 0 0.43 1.16 0 0 0 0 0 0 0.19 ...
## min: 0 - max: 5.45 - NAs: 0 (0%) - 164 unique values
##
## $make:
## numeric: 0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...
## min: 0 - max: 4.54 - NAs: 0 (0%) - 142 unique values
##
## $yesno:
## character: "y" "y" "y" "y" "y" "y" "y" "y" "y" "y" ...
## NAs: 0 (0%) - 2 unique values
```

```
summary(spam)
```

```
##        X           crl.tot            dollar             bang
##  Min.   :   1   Min.   :    1.0   Min.   :0.00000   Min.   : 0.0000
##  1st Qu.:1151   1st Qu.:   35.0   1st Qu.:0.00000   1st Qu.: 0.0000
##  Median :2301   Median :   95.0   Median :0.00000   Median : 0.0000
##  Mean   :2301   Mean   :  283.3   Mean   :0.07581   Mean   : 0.2691
##  3rd Qu.:3451   3rd Qu.:  266.0   3rd Qu.:0.05200   3rd Qu.: 0.3150
##  Max.   :4601   Max.   :15841.0   Max.   :6.00300   Max.   :32.4780
##      money              n000             make            yesno
##  Min.   : 0.00000   Min.   :0.0000   Min.   :0.0000   Length:4601
##  1st Qu.: 0.00000   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median : 0.00000   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   : 0.09427   Mean   :0.1016   Mean   :0.1046
##  3rd Qu.: 0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :12.50000   Max.   :5.4500   Max.   :4.5400
```

```
str(spam)
```

```
## 'data.frame':    4601 obs. of  8 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ crl.tot: int  278 1028 2259 191 191 54 112 49 1257 749 ...
##  $ dollar : num  0 0.18 0.184 0 0 0 0.054 0 0.203 0.081 ...
##  $ bang   : num  0.778 0.372 0.276 0.137 0.135 0 0.164 0 0.181 0.244 ...
##  $ money  : num  0 0.43 0.06 0 0 0 0 0 0.15 0 ...
```

```
##  $ n000   : num  0 0.43 1.16 0 0 0 0 0 0 0.19 ...
##  $ make   : num  0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...
##  $ yesno  : chr  "y" "y" "y" "y" ...
```

```
sum(is.na(spam))
```

```
## [1] 0
```

```
names(spam)[names(spam)=='crl.tot'] <-'lencap'
```

Building model

Step 1: Splitting the data into train / test

```
split<-sample.split(spam, SplitRatio = 0.8)
train<-subset(spam,split=='TRUE')
test <-subset(spam,split=='FALSE')
```

Step 2: Training the model. yesno is the dependant variable and the others are the independant.We need to recode yesno, redo the train / test and then run the model.

```
spam$yesno <- recode(spam$yesno,"'y'=1;'n'=0")
split<-sample.split(spam, SplitRatio = 0.8)
train<-subset(spam,split=='TRUE')
test <-subset(spam,split=='FALSE')
mymodel <- glm(yesno ~ lencap+dollar+bang+money+n000+make, data=train, family='binomial')
```

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

```
summary(mymodel)
```

```
##
## Call:
## glm(formula = yesno ~ lencap + dollar + bang + money + n000 +
##     make, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.6128  -0.5820   0.4120   1.9318
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.6990416  0.0617979 -27.494  < 2e-16 ***
## lencap       0.0006587  0.0001096   6.012 1.83e-09 ***
## dollar       8.9066554  0.7331337  12.149  < 2e-16 ***
## bang         1.3153743  0.1196920  10.990  < 2e-16 ***
## money        2.1659613  0.2775783   7.803 6.04e-15 ***
## n000         4.3436394  0.5229124   8.307  < 2e-16 ***
## make         0.0208952  0.1684914   0.124    0.901
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4628.1  on 3450  degrees of freedom
## Residual deviance: 3024.8  on 3444  degrees of freedom
```

```
## AIC: 3038.8
##
## Number of Fisher Scoring iterations: 8
```

Oberservation: I use all the variables. According to Significant level, Dollar is the strongest variable, followed by n000.

Step 3: Running the test data through the model

```r
res <- predict(mymodel,test,type='response')
```

Step 4: Creating the confusion matrix to validate the model

```r
confmatrix <- table(Actual_value=test$yesno, Predicted_value=res>0.5)
confmatrix
```

```
##             Predicted_value
## Actual_value FALSE TRUE
##            0   663   34
##            1   183  270
```

Step 5: Calculating the accuracy of our model

```r
(confmatrix[[1,1]]+confmatrix[[2,2]])/sum(confmatrix)
```

```
## [1] 0.8113043
```

Conclusion:

Based on the variables selected, there will be a accuracy level of more than 80% to assess whether this email is spam or not spam.