Recognising spam email with R

Vilma Stasiute

24/06/2020

The task: Using the data from the spam email data file and using logistic regresion, create a predictive model to know if an email is spam or not. Use all the variables. (What are the significant variables and which is their order of importance?)

Step 0: Importing libraries

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

Step 1: Import the data and read the documentation

Data source: https://vincentarelbundock.github.io/Rdatasets/datasets.html

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

Step 2: Exploratory analysis

Chekcing out the datasset:

```
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:

```
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
```

The sum of null values in the dataset:

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

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

Step 3: Rename "crl.tot" to "lencap"

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

```
## [1] "X"      "lencap" "dollar" "bang"   "money"  "n000"   "make"   "yesno"
```

Step 4: Splitting the data into train / test

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

Step 5: 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.

Recoding yesno variable:

```
spam$yesno<-recode(spam$yesno, " 'y'=1; 'n'=0 ")
```

Train and test the model again:

```
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")
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  -0.6063  -0.5723   0.4442   1.9575
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.7346000  0.0626185 -27.701  < 2e-16 ***
## lencap       0.0007431  0.0001175   6.324 2.55e-10 ***
## dollar       7.0001127  0.6748572  10.373  < 2e-16 ***
## bang         1.9326683  0.1410405  13.703  < 2e-16 ***
## money        1.9519363  0.2732720   7.143 9.14e-13 ***
## n000         4.0824559  0.4799183   8.507  < 2e-16 ***
## make        -0.0225479  0.1688726  -0.134    0.894
## ---
## 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: 3000.3  on 3444  degrees of freedom
## AIC: 3014.3
##
## Number of Fisher Scoring iterations: 7
```

Dollar is the strongest variable predicting whether email is spam or not, then n000, money, bang and lencap. The p-value of vairiable 'make' is above 0.05. It does not add predictive value to the model, but since it's estimate is close to zero, removing it would not make a big difference.

Step 6: Running the test data through the model

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

Step 7: Creating the confusion matrix to validate the model

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

```
##             Predicted_value
## Actual_value FALSE TRUE
##            0   658   39
##            1   149  304
```

Step 8: Calculating the accuracy of our model

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

```
## [1] 0.8365217
```

The model is accurate more than 80% of times.