Statistical Machine Learning Assignment 2

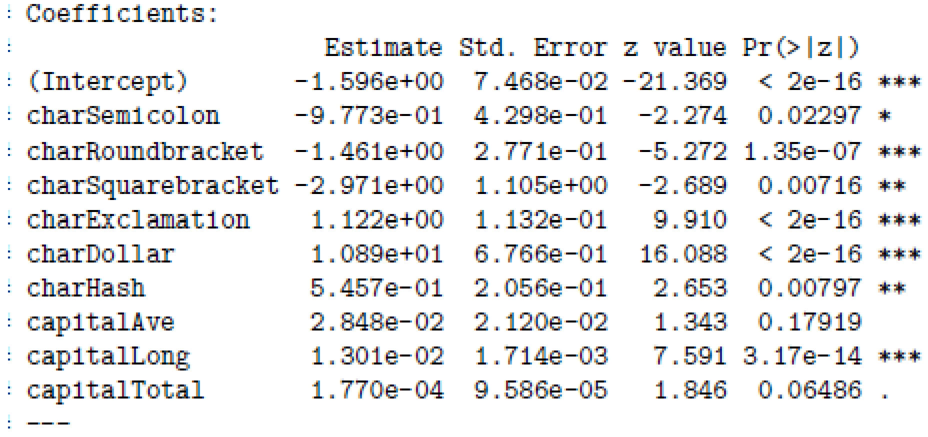
-Shubhang Periwal

Abstract

The dataset spam available in package kernlab contains 57 features extracted from the content of emails which were classified as spam or nonspam. The variables 49-54 indicate the frequency of the special characters ;, (, [, !, \$, and \#. The variables 55-57 contain the average, longest and total run-length of capital letters. The last variable, type indicates if an email is spam or nonspam. Our aim is to predict whether an email is spam or not. We need to use a subset of features related to frequencies.

Methodology and Observation

- Extracting the required columns out of the total columns.
- Dividing the data into training and testing set. Training set is also further used for tau's validation.
- We need to see the structure of data, and we can see that that all variables are numerical.
- Test set is to show that with change in value of tau, the results in the testing set improve.
- Next step is to use glm function to train the model for type with training data passed within the model.
- Summary of our model shows the coefficients estimated for the model.
- We can see that both charDollar and charExclamation have 3 stars (this shows that they have significant importance in prediction of spam or not spam).



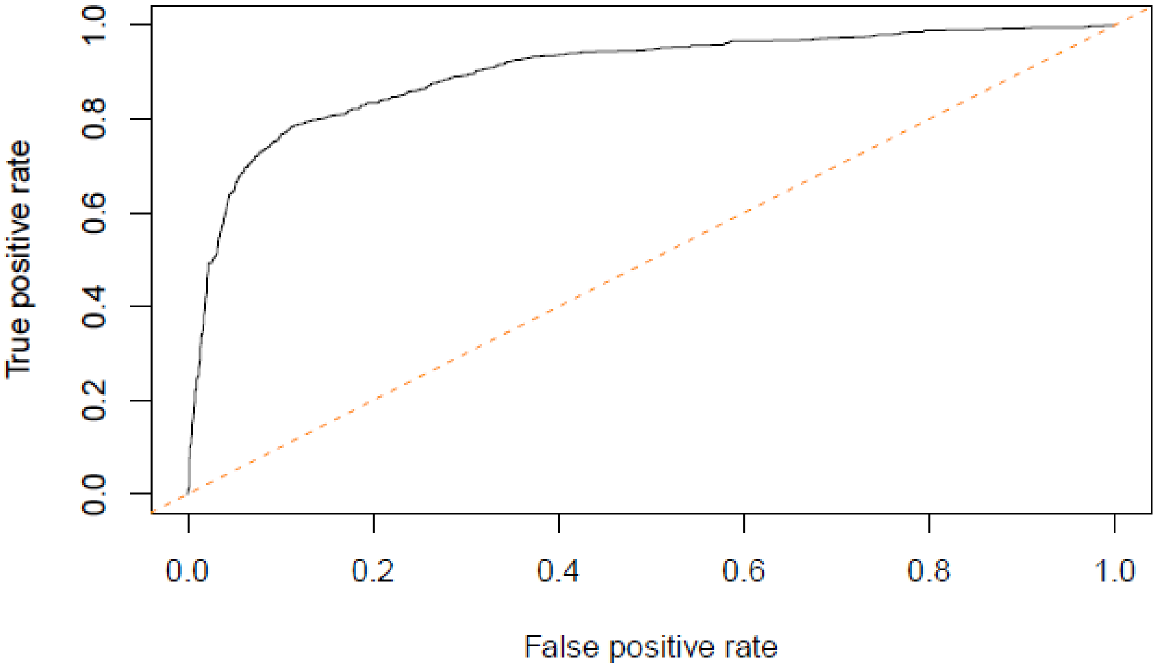
```
: Coefficients:
:                    Estimate Std. Error z value Pr(>|z|)
: (Intercept)       -1.596e+00  7.468e-02 -21.369  < 2e-16 ***
: charSemicolon     -9.773e-01  4.298e-01  -2.274  0.02297 *
: charRoundbracket  -1.461e+00  2.771e-01  -5.272 1.35e-07 ***
: charSquarebracket -2.971e+00  1.105e+00  -2.689  0.00716 **
: charExclamation    1.122e+00  1.132e-01   9.910  < 2e-16 ***
: charDollar         1.089e+01  6.766e-01  16.088  < 2e-16 ***
: charHash           5.457e-01  2.056e-01   2.653  0.00797 **
: capitalAve         2.848e-02  2.120e-02   1.343  0.17919
: capitalLong        1.301e-02  1.714e-03   7.591 3.17e-14 ***
: capitalTotal       1.770e-04  9.586e-05   1.846  0.06486 .
: ---
```

- With increase in a single exclamation mark, the estimate increases by 1.122, which is positive, so having an more exclamation marks might lead to spam classification.
- For each dollar sign used, the estimate increases by 10.89 (extremely high), this could also lead to a spam classification.
- 95% interval for charExclamation is [0.8999612,1.343729].
- 95% interval for charDollar is [9.559288,12.2116].
- Both charExclamation and charDollar have significant impact on the probability of an email being spam.

Inferential Problems with dollar character and Exclamation mark

- Financial emails or receipts have high use of dollar sign, this could lead to misclassification of emails.
- Urgent mails could have a high number of exclamation marks which could easily be misclassified to spam due to high coefficient of charExclamation.
- There is a possibility that some data is not present in this dataset and we might have to investigate other features of data. But both variables have a high coefficient which might impact result in case of extreme cases.

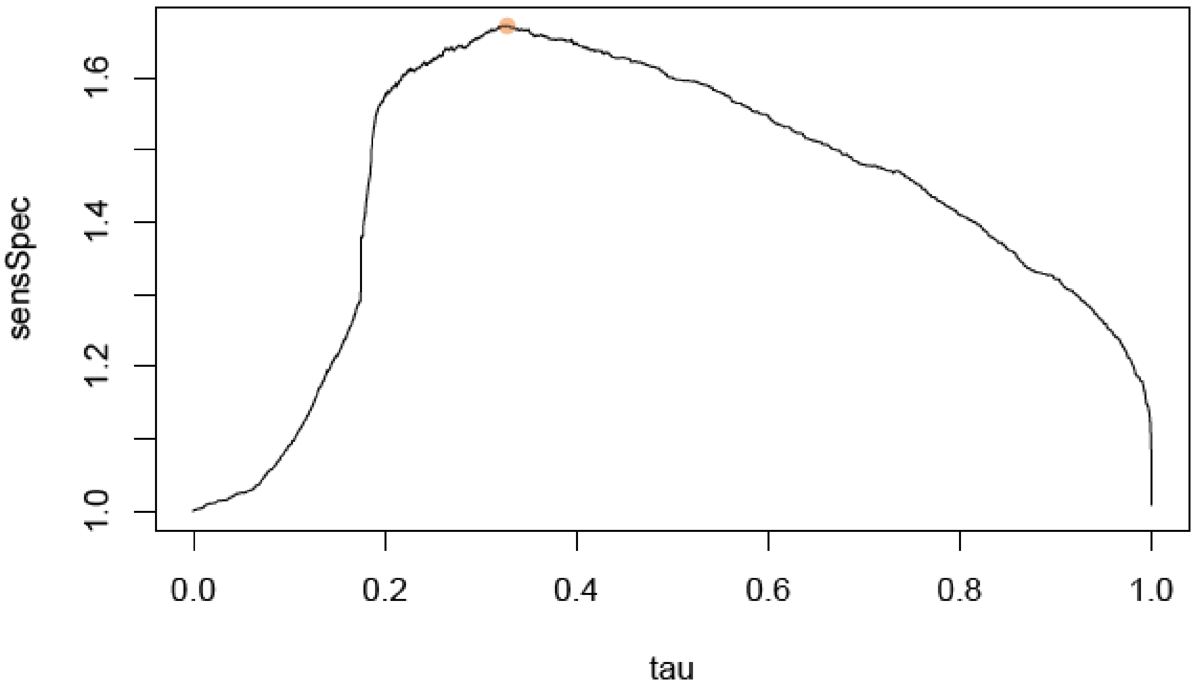
We then apply performance function to calculate different performance measures. We then use this function to plot ROC curve.



Model Performance with tau initialized to 0.5

The model gives an accuracy of 78.39% on the test set. On training set, it gives a high accuracy of 89.96763%.

The optimal threshold tao can be found by maximizing the sum of sensitivity and specificity for different values of tao.



The value of tau comes out to be 0.3277191.

We then calculate accuracy of the model with new value of tau. The accuracy increases from 78.39 to 79.805%.

When we train the model and test it for entire data, we get different figures as specified below.

We get an accuracy of 80% without updating tau and an accuracy of 90% when we update tau. The results are so accurate because the model is trained and tested on the same data.

Appendix

~~~
---
title: "SML Assignment 2"
author: "Shubhang Periwal 19201104"
date: "3/31/2020"
output: pdf_document
---

```{r}
library(kernlab)
library(ROCR)
data(spam)
dt <- spam[,49:58]
# breaking data into test and train sets
set.seed(19201104)
~~~

~~~
tot <- nrow(dt)
n <- floor(tot*0.8)
# training indices
train <- sample((1:tot),n)
#testing indices
test <- setdiff(1:tot,c(train))
str(dt)
```



```{r}
fit = glm(type ~ ., data = dt[train,], family = "binomial")
summary(fit)
```

Coefficient analysis
```{r}
chd <- fit$coefficients["charDollar"]
chex <- fit$coefficients["charExclamation"]

exp(chd)
exp(chex)

sm <- summary(fit)
se_chd <- sm$coef["charDollar",2]
se_chex <- sm$coeff["charExclamation",2]

#calculating 95% interval for char Dollar
chd_l <- chd - 1.96*se_chd
chd_u <- chd + 1.96*se_chd
chd_l
chd_u
#calculating 95% interval for char exclamation
chex_l <- chex - 1.96*se_chex
chex_u <- chex + 1.96*se_chex
chex_l
chex_u

```



```{r}
p <- predict(fit, dt[test,])
tau <- 0.5
~~~

~~~~
pred = ifelse(p > tau, "SPAM", "NON-SPAM")
p <- fitted(fit)
res <- table(dt[test,]$type, pred)
res
res<- as.matrix(res)
accuracy <- (res[1,1] + res[2,2])/sum(res)
accuracy
```




```{r}
predobj <- prediction(fitted(fit), dt[train,]$type)
perf <- performance(predobj, "tpr", "fpr")
plot(perf)
abline(0, 1, col = "darkorange", lty = 2)
auc = performance(predobj, "auc")
auc@y.values
```

```{r}
sens <- performance(predobj, "sens")
spec <- performance(predobj, "spec")
tau <- sens@x.values[[1]]
sensSpec <- sens@y.values[[1]] + spec@y.values[[1]]
best <- which.max(sensSpec)
plot(tau, sensSpec, type = "l")
points(tau[best], sensSpec[best], pch = 19, col = adjustcolor("darkorange2", 0.5))
```

```{r}
p <- predict(fit, dt[test,])
tau[best]
pred = ifelse(p > tau[best], "spam", "nonspam")
res<- table(dt[test,]$type, pred)
res
res<- as.matrix(res)
accuracy <- (res[1,1] + res[2,2])/sum(res)
accuracy
```
```{r}
fit = glm(type ~ ., data = dt, family = "binomial")
summary(fit)
p <- predict(fit,dt)
tau <- 0.5
~~~~

~~~
pred = ifelse(p > tau, "SPAM", "NON-SPAM")
p <- fitted(fit)
res <- table(dt$type, pred)
res
res<- as.matrix(res)
accuracy <- (res[1,1] + res[2,2])/sum(res)
accuracy
sens <- performance(predobj, "sens")
spec <- performance(predobj, "spec")
tau <- sens@x.values[[1]]
sensSpec <- sens@y.values[[1]] + spec@y.values[[1]]
best <- which.max(sensSpec)
plot(tau, sensSpec, type = "l")
points(tau[best], sensSpec[best], pch = 19, col = adjustcolor("darkorange2", 0.5))

predobj <- prediction(fitted(fit), dt$type)
perf <- performance(predobj, "tpr", "fpr")
plot(perf)
abline(0, 1, col = "darkorange", lty = 2)
auc = performance(predobj, "auc")
auc@y.values
```
~~~