Statistical Machine Learning

Nisarga G-19203753

02/04/2020

Download all the required packages: kernlab , this package contains “spam” dataset and load the dataset.The dataset has 58 variables, but for the purpose of assignment we include 49-58 variables where 58th variable is response variable “type”.

```
#install.packages("kernlab")
library(kernlab)
data("spam")
spam <- spam[,49:58]
```

Fit the logistic regression model:We fit the logistic regression where the response variable “type” is the function of all the other variables.

```
fit <- glm(type ~ ., data = spam, family = "binomial")

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

summary(fit)

##
## Call:
## glm(formula = type ~ ., family = "binomial", data = spam)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -0.6403  -0.5211   0.5177   3.6202
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.677e+00  7.038e-02 -23.835  < 2e-16 ***
## charSemicolon     -1.055e+00  4.117e-01  -2.562 0.010419 *
## charRoundbracket  -1.441e+00  2.513e-01  -5.733 9.87e-09 ***
## charSquarebracket -3.878e+00  1.085e+00  -3.574 0.000351 ***
## charExclamation    1.312e+00  1.100e-01  11.931  < 2e-16 ***
## charDollar         1.059e+01  6.007e-01  17.622  < 2e-16 ***
## charHash           3.553e-01  1.445e-01   2.459 0.013924 *
## capitalAve         5.560e-02  2.195e-02   2.533 0.011308 *
## capitalLong        1.385e-02  1.653e-03   8.377  < 2e-16 ***
## capitalTotal       1.687e-04  8.902e-05   1.895 0.058034 .
## ---
## 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: 4042.6  on 4591  degrees of freedom
## AIC: 4062.6
##
## Number of Fisher Scoring iterations: 8
```

charExclamation and charDollar are significantly contributing to the response variable.The three stars *** in the summary of the fit indicates that particular varaible is higly significant.By hypothesis testing,β1 and β2 are the coefficiencts-Null hypothesis: H0 : β1=β2=0, alternative hypothesis: Ha: β1=β2!= 0.pvalue is 0 so we reject the null hypothesis and conclude that both the variables are significantly different from 0 hence these two varaibles are highly significant.

Inferential problems related to these two variables:The problem is the perfect speration that gives higer values of coefficients and standard errors,which means they might be unreliable. Also,in reality any email with more number of $ and ! characters would be classified as spam, but that email may not be spam. For 1 unit increase in variables charExclamation and charDollar , there is 1.312 and 10.59 respective increase in the probability of that email being spam.Hence more the number of ! and $ characters more likely the email is going to be classified as spam.

PREDICTIVE PERFORMANCE: We can classify the observations for $\tau = 0.5$ running the following lines of code.

```
tau <- 0.5
p <- fitted(fit)
pred <- ifelse(p > tau, 1, 0)
table(spam$type, pred)
```

```
##          pred
##              0    1
##   nonspam 2645  143
##   spam     604 1209
```

Package ROCR can be used to calculate many performance measures. To use the functionalities of the package, we first need to create a prediction object, providing in input the estimated probabilities and the actual class values of the response variable.

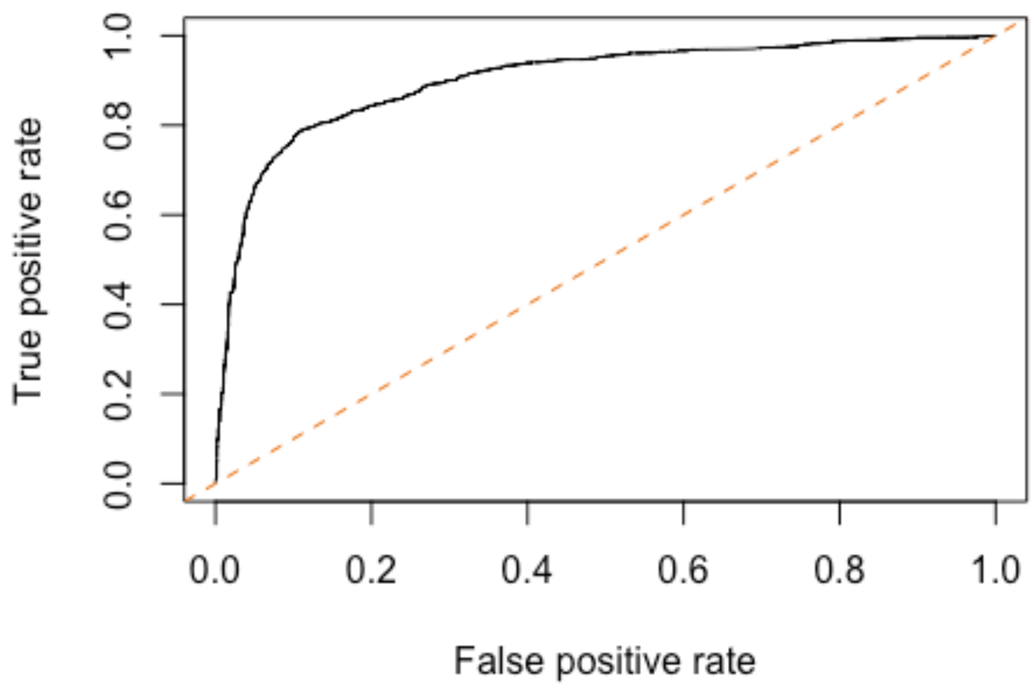
```
#install.packages("ROCR")
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
predObj <- prediction(fitted(fit), spam$type)
perf <- performance(predObj, "tpr", "fpr")
plot(perf)
abline(0,1, col = "darkorange2", lty = 2) # add bisect line
```



The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.In the graph above, the curve is not closer to the straight line hence the accuracy is more.

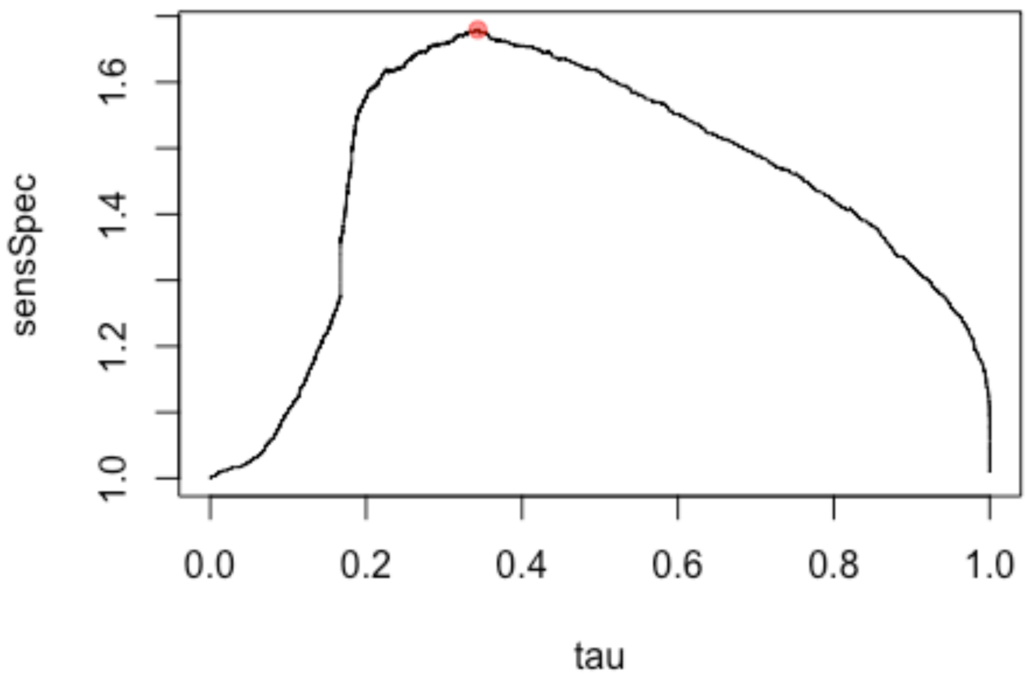
```r
auc <- performance(predObj, "auc")
auc@y.values

## [[1]]
## [1] 0.9020404
```

Accuracy is approximately 90.20%, which indicates that my prediction is 90.20% right.

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

```r
sens <- performance(predObj, "sens")
spec <- performance(predObj, "spec")
tau <- sens@x.values[[1]]
sensSpec <- sens@y.values[[1]] + spec@y.values[[1]]
```

```r
best <- which.max(sensSpec)
plot(tau, sensSpec, type = "l")
points(tau[best], sensSpec[best], pch = 19, col = adjustcolor("red", 0.5))
```




```r
tau[best]

##       195
## 0.3432798

pred <- ifelse(fitted(fit) > tau[best], 1, 0)
table(spam$type, pred)

##          pred
##              0    1
##   nonspam 2488  300
##   spam     388 1425
```

The optimal threshold value indicated by red dot in the graph above is found out to be 0.34, this inidcates that this provides better accuracy than tau=0.5 .