Lab4.R

2020-03-31

```r
install.packages("kernlab", repos = "http://cran.us.r-project.org")

library(kernlab)
data("spam")
tibble::as.tibble(spam)

is.factor(spam$type)

## [1] TRUE

levels(spam$type)

## [1] "nonspam" "spam"

set.seed(42)
# spam_idx = sample(nrow(spam), round(nrow(spam) / 2))
spam_idx = sample(nrow(spam), 1000)
spam_trn = spam[spam_idx, ]
spam_tst = spam[-spam_idx, ]


fit_caps = glm(type ~ capitalTotal,
               data = spam_trn, family = binomial)
fit_selected = glm(type ~ edu + money + capitalTotal + charDollar,
                   data = spam_trn, family = binomial)

fit_additive = glm(type ~ .,
                   data = spam_trn, family = binomial)

fit_over = glm(type ~ capitalTotal * (.),
               data = spam_trn, family = binomial, maxit = 50)

# training misclassification rate
mean(ifelse(predict(fit_caps) > 0, "spam", "nonspam") != spam_trn$type)

## [1] 0.339

mean(ifelse(predict(fit_selected) > 0, "spam", "nonspam") != spam_trn$type)

## [1] 0.224

mean(ifelse(predict(fit_additive) > 0, "spam", "nonspam") != spam_trn$type)

## [1] 0.066

mean(ifelse(predict(fit_over) > 0, "spam", "nonspam") != spam_trn$type)
```

```
## [1] 0.136

library(boot)

set.seed(1)
# Model 1
cv.glm(spam_trn, fit_caps, K = 5)$delta[1]

## [1] 0.2166961

# Model 2
cv.glm(spam_trn, fit_selected, K = 5)$delta[1]

## [1] 0.1587043

# Model 3
cv.glm(spam_trn, fit_additive, K = 5)$delta[1]

# Model 4
cv.glm(spam_trn, fit_over, K = 5)$delta[1]

## Exercise 1

## 1 - Most underfit to most overfit

# 1. Model 1 - "fit_caps"
# 2. Model 2 - "fit_selected"
# 3. Model 4 - "fit_over"
# 4. Model 3 - "fit_additive"

## 2

set.seed(3)
# Model 1
cv.glm(spam_trn, fit_caps, K = 100)$delta[1]

## [1] 0.2168394

# Model 2
cv.glm(spam_trn, fit_selected, K = 100)$delta[1]

# Model 3
cv.glm(spam_trn, fit_additive, K = 100)$delta[1]

# Our conclusion stays the same as the results were not any different than ou
r initial analysis

# Create 4 confusion matrices

make_conf_mat = function(predicted, actual) {
  table(predicted = predicted, actual = actual)
}
```

```r
spam_add_pred = ifelse(predict(fit_additive, spam_tst, type = "response") > 0
.5,
                       "spam",
                       "nonspam")
spam_caps_pred = ifelse(predict(fit_caps, spam_tst, type = "response") > 0.5,
                        "spam",
                        "nonspam")
spam_over_pred = ifelse(predict(fit_over, spam_tst, type = "response") > 0.5,
                        "spam",
                        "nonspam")
spam_select_pred = ifelse(predict(fit_selected, spam_tst, type = "response")
> 0.5,
                          "spam",
                          "nonspam")

(conf_mat_addi = make_conf_mat(predicted = spam_add_pred, actual = spam_tst$t
ype))

##          actual
## predicted nonspam spam
##   nonspam    2057  157
##   spam        127 1260

(conf_mat_caps = make_conf_mat(predicted = spam_caps_pred, actual = spam_tst$
type))

##          actual
## predicted nonspam spam
##   nonspam    2022 1066
##   spam        162  351

(conf_mat_over = make_conf_mat(predicted = spam_over_pred, actual = spam_tst$
type))

##          actual
## predicted nonspam spam
##   nonspam    1725  103
##   spam        459 1314

(conf_mat_selc = make_conf_mat(predicted = spam_select_pred, actual = spam_ts
t$type))

##          actual
## predicted nonspam spam
##   nonspam    2073  615
##   spam        111  802

table(spam_tst$type) / nrow(spam_tst)
```

```
##
##   nonspam      spam
## 0.6064982 0.3935018

## Exercise 2 ##

bank <- read.csv('https://msudataanalytics.github.io/SSC442/Labs/data/bank.cs
v')

set.seed(42)
bank_idx <- sample(nrow(bank), 1000)
bank_trn = bank[bank_idx, ]
bank_tst = bank[-bank_idx, ]

bank_log <- glm(y ~., data = bank_trn, family = binomial)

cv <- cv.glm(bank_trn, bank_log, K=10)$delta[1]

#summary(bank_log)

Call:
glm(formula = y ~ ., family = binomial, data = bank_trn)

Deviance Residuals:
    Min       1Q   Median       3Q
-3.9762  -0.4005  -0.2493  -0.1355
    Max
 2.8376

Coefficients:
                     Estimate Std. Error
(Intercept)        -2.720e+00  1.126e+00
age                -5.544e-03  1.557e-02
jobblue-collar      3.286e-01  5.486e-01
jobentrepreneur    -5.631e-01  9.767e-01
jobhousemaid        1.475e+00  8.330e-01
jobmanagement       2.029e-01  5.510e-01
jobretired          1.711e+00  6.446e-01
jobself-employed   -8.042e-01  8.372e-01
jobservices         3.578e-01  6.390e-01
jobstudent          1.832e+00  8.625e-01
jobtechnician       3.194e-01  5.239e-01
jobunemployed      -4.103e-01  9.021e-01
jobunknown          2.612e+00  1.021e+00
maritalmarried     -5.955e-01  3.608e-01
maritalsingle      -6.406e-01  4.375e-01
educationsecondary  4.198e-01  4.511e-01
educationtertiary   9.755e-01  5.043e-01
educationunknown   -1.003e+00  9.832e-01
defaultyes          9.149e-01  8.344e-01
```

```
balance          -4.218e-05  4.325e-05
housingyes       -6.792e-01  3.177e-01
loanyes          -2.769e-01  3.708e-01
contacttelephone -2.499e-01  5.026e-01
contactunknown   -1.927e+00  5.012e-01
day               1.114e-02  1.823e-02
monthaug         -8.167e-01  5.356e-01
monthdec          1.115e-01  1.277e+00
monthfeb         -2.255e-01  6.159e-01
monthjan         -1.318e+00  7.964e-01
monthjul         -1.306e+00  5.446e-01
monthjun          3.694e-01  6.767e-01
monthmar          7.788e-01  8.963e-01
monthmay         -1.343e-01  4.793e-01
monthnov         -7.416e-01  5.447e-01
monthoct          1.690e+00  7.152e-01
monthsep          9.775e-01  9.172e-01
duration          4.321e-03  4.384e-04
campaign         -6.471e-02  5.593e-02
previous          1.142e-01  5.235e-02
```