Week10

Marc Riley

5/23/2021

install.packages(foreign) install.packages("caTools")

```
library(foreign)

library(caTools)

## Warning: package 'caTools' was built under R version 4.0.5

library(caret)

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.5

library(magrittr)

## Warning: package 'magrittr' was built under R version 4.0.5

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
```

```
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::lift()      masks caret::lift()
## x purrr::set_names() masks magrittr::set_names()
```

- Set the working directory to the root of your DSC 520 directory

```
thoraric_df <- read.arff("ThoracicSurgery.arff")


str(thoraric_df)

## 'data.frame':    470 obs. of  17 variables:
##  $ DGN    : Factor w/ 7 levels "DGN1","DGN2",..: 2 3 3 3 3 3 3 2 3 3 ...
##  $ PRE4   : num  2.88 3.4 2.76 3.68 2.44 2.48 4.36 3.19 3.16 2.32 ...
##  $ PRE5   : num  2.16 1.88 2.08 3.04 0.96 1.88 3.28 2.5 2.64 2.16 ...
##  $ PRE6   : Factor w/ 3 levels "PRZ0","PRZ1",..: 2 1 2 1 3 2 2 2 3 2 ...
##  $ PRE7   : Factor w/ 2 levels "F","T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRE8   : Factor w/ 2 levels "F","T": 1 1 1 1 2 1 1 1 1 1 ...
##  $ PRE9   : Factor w/ 2 levels "F","T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRE10  : Factor w/ 2 levels "F","T": 2 1 2 1 2 2 2 2 2 2 ...
##  $ PRE11  : Factor w/ 2 levels "F","T": 2 1 1 1 2 1 1 1 2 1 ...
##  $ PRE14  : Factor w/ 4 levels "OC11","OC12",..: 4 2 1 1 1 1 2 1 1 1 ...
##  $ PRE17  : Factor w/ 2 levels "F","T": 1 1 1 1 1 1 2 1 1 1 ...
##  $ PRE19  : Factor w/ 2 levels "F","T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PRE25  : Factor w/ 2 levels "F","T": 1 1 1 1 1 1 1 2 1 1 ...
##  $ PRE30  : Factor w/ 2 levels "F","T": 2 2 2 1 2 1 2 2 2 2 ...
##  $ PRE32  : Factor w/ 2 levels "F","T": 1 1 1 1 1 1 1 1 1 1 ...
##  $ AGE    : num  60 51 59 54 73 51 59 66 68 54 ...
##  $ Risk1Yr: Factor w/ 2 levels "F","T": 1 1 1 1 2 1 2 2 1 1 ...
```

Fit a binary logistic regression model to the data set that predicts whether or not the patient survived for one year (the Risk1Y variable) after the surgery. Use the glm() function to perform the logistic regression. See Generalized Linear Models for an example. Include a summary using the summary() function in your results.

```
Model<- glm(Risk1Yr ~., data= thoraric_df, family = 'binomial')

summary (model)

##
## Call:
## glm(formula = Risk1Yr ~ ., family = "binomial", data = thoraric_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6084  -0.5439  -0.4199  -0.2762   2.4929
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.655e+01  2.400e+03  -0.007  0.99450
## DGNDGN2      1.474e+01  2.400e+03   0.006  0.99510
## DGNDGN3      1.418e+01  2.400e+03   0.006  0.99528
## DGNDGN4      1.461e+01  2.400e+03   0.006  0.99514
## DGNDGN5      1.638e+01  2.400e+03   0.007  0.99455
## DGNDGN6      4.089e-01  2.673e+03   0.000  0.99988
## DGNDGN8      1.803e+01  2.400e+03   0.008  0.99400
## PRE4        -2.272e-01  1.849e-01  -1.229  0.21909
## PRE5        -3.030e-02  1.786e-02  -1.697  0.08971 .
## PRE6PRZ1    -4.427e-01  5.199e-01  -0.852  0.39448
## PRE6PRZ2    -2.937e-01  7.907e-01  -0.371  0.71030
## PRE7T        7.153e-01  5.556e-01   1.288  0.19788
## PRE8T        1.743e-01  3.892e-01   0.448  0.65419
## PRE9T        1.368e+00  4.868e-01   2.811  0.00494 **
## PRE10T       5.770e-01  4.826e-01   1.196  0.23185
## PRE11T       5.162e-01  3.965e-01   1.302  0.19295
## PRE14OC12    4.394e-01  3.301e-01   1.331  0.18318
## PRE14OC13    1.179e+00  6.165e-01   1.913  0.05580 .
## PRE14OC14    1.653e+00  6.094e-01   2.713  0.00668 **
## PRE17T       9.266e-01  4.445e-01   2.085  0.03709 *
## PRE19T      -1.466e+01  1.654e+03  -0.009  0.99293
## PRE25T      -9.789e-02  1.003e+00  -0.098  0.92227
## PRE30T       1.084e+00  4.990e-01   2.172  0.02984 *
## PRE32T      -1.398e+01  1.645e+03  -0.008  0.99322
## AGE         -9.506e-03  1.810e-02  -0.525  0.59944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 395.61  on 469  degrees of freedom
## Residual deviance: 341.19  on 445  degrees of freedom
## AIC: 391.19
##
## Number of Fisher Scoring iterations: 15
```

According to the summary, which variables had the greatest effect on the survival rate?

- the best variable appears to be PRE9t and PRE14OC14

To compute the accuracy of your model, use the dataset to predict the outcome variable. The percent of correct predictions is the accuracy of your model. What is the accuracy of your model?

```
change <- sample.split(thoraric_df,SplitRatio = .8)
change
```

```
##  [1] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
FALSE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
train<- subset(thoraric_df, change=="TRUE")
test<- subset(thoraric_df, change=="FALSE")

res<- predict(model,train, type="response")
res
```

```
##            2            3            4            6            7
8
## 1.031988e-01 8.287068e-02 2.160824e-02 3.415054e-02 1.918605e-01
1.068699e-01
##            9           10           13           14           15
16
## 1.265083e-01 9.458663e-02 1.154378e-01 4.908434e-01 8.528088e-02
7.638833e-02
##           17           19           20           21           23
24
## 2.298384e-01 1.170482e-01 6.346676e-02 7.899455e-02 1.166706e-01
5.824619e-02
##           25           26           27           30           31
32
## 4.628603e-01 2.759707e-01 7.223499e-02 5.945905e-08 3.730799e-01
3.210049e-02
##           33           34           36           37           38
40
## 5.401980e-01 1.222741e-01 8.141605e-02 1.247959e-01 1.985475e-01
5.736768e-02
##           41           42           43           44           47
48
## 3.831235e-01 1.723143e-01 1.022412e-01 6.839303e-01 8.354285e-02
1.128335e-01
##           49           50           51           53           54
55
## 1.528144e-01 2.634907e-02 3.990471e-02 5.605594e-01 1.268064e-01
9.604222e-02
##           57           58           59           60           61
64
## 1.040492e-01 3.868351e-01 9.091183e-02 8.436518e-02 1.882038e-01
5.221406e-02
##           65           66           67           68           70
```

```
71
## 2.068899e-01 4.547291e-02 3.426478e-02 2.306748e-01 1.235686e-01
1.769600e-02
##           72           74           75           76           77
78
## 2.044482e-01 1.854511e-02 5.622961e-02 3.214431e-01 1.517401e-01
1.088240e-01
##           81           82           83           84           85
87
## 1.007965e-01 3.642241e-01 1.092554e-01 6.808071e-02 8.282431e-02
1.516943e-01
##           88           89           91           92           93
94
## 2.220150e-01 6.230735e-01 1.475171e-01 7.598004e-02 1.018244e-01
3.580610e-02
##           95           98           99          100          101
102
## 2.064928e-01 8.663401e-08 5.044656e-02 3.001414e-01 6.405787e-02
3.957982e-01
##          104          105          106          108          109
110
## 2.874635e-08 3.097683e-02 1.314217e-01 1.068128e-01 2.236160e-02
2.980639e-01
##          111          112          115          116          117
118
## 1.234449e-01 2.098142e-01 1.245632e-01 2.922307e-01 2.340033e-01
2.686309e-01
##          119          121          122          123          125
126
## 6.225151e-02 3.945990e-02 9.033179e-02 6.199320e-01 1.457683e-01
1.099803e-01
##          127          128          129          132          133
134
## 5.418171e-02 3.286049e-01 4.130719e-01 1.221660e-01 1.801905e-01
8.439071e-02
##          135          136          138          139          140
142
## 7.935226e-02 7.695837e-02 3.812039e-01 1.332096e-01 2.572193e-02
9.231166e-02
##          143          144          145          146          149
150
## 1.029460e-02 1.677159e-01 1.824691e-01 9.334413e-02 8.884902e-02
6.588596e-02
##          151          152          153          155          156
157
## 4.217588e-02 7.084935e-02 4.472309e-02 1.027427e-01 9.794784e-02
4.854969e-01
##          159          160          161          162          163
166
## 1.867933e-01 9.485986e-02 3.309436e-02 7.273292e-02 2.214874e-01
```

```
3.826184e-01
##          167          168          169          170          172
173
## 1.813499e-01 1.147794e-01 1.863320e-01 3.319553e-01 3.371654e-01
4.754743e-01
##          174          176          177          178          179
180
## 8.801868e-02 3.810037e-01 3.419036e-01 1.155253e-01 1.691160e-01
2.023070e-01
##          183          184          185          186          187
189
## 7.236749e-02 1.208968e-01 2.770187e-02 4.974416e-01 7.037954e-02
8.370741e-02
##          190          191          193          194          195
196
## 9.786972e-02 1.071501e-07 5.107552e-02 8.899037e-02 6.161650e-02
1.414413e-01
##          197          200          201          202          203
204
## 1.467324e-01 1.827940e-01 1.353227e-01 7.811592e-02 3.490320e-01
1.466339e-01
##          206          207          208          210          211
212
## 1.172731e-01 5.645845e-02 8.096561e-02 3.416674e-01 4.821277e-02
1.035481e-01
##          213          214          217          218          219
220
## 3.447902e-01 2.562132e-01 1.778609e-01 7.094838e-02 5.571797e-02
6.582535e-02
##          221          223          224          225          227
228
## 7.270148e-01 2.586989e-01 5.110705e-02 8.371578e-02 1.733864e-01
1.206525e-01
##          229          230          231          234          235
236
## 2.726272e-02 2.558265e-01 1.897757e-01 1.282731e-01 1.317057e-01
8.638962e-02
##          237          238          240          241          242
244
## 1.567634e-01 1.013461e-01 1.033867e-01 4.409613e-02 6.391354e-02
3.604740e-02
##          245          246          247          248          251
252
## 3.259522e-08 7.021216e-02 7.865337e-02 1.397018e-01 9.038743e-02
1.235385e-01
##          253          254          255          257          258
259
## 9.386811e-02 9.485861e-02 7.640224e-02 8.482854e-02 7.348739e-02
8.010688e-02
##          261          262          263          264          265
```

```
268
## 1.134974e-01 1.358705e-01 1.392593e-01 3.270853e-02 8.239156e-02
3.207561e-01
##          269          270          271          272          274
275
## 4.979178e-01 1.011537e-01 1.828671e-01 3.733253e-01 3.399052e-01
1.567863e-01
##          276          278          279          280          281
282
## 1.394679e-01 2.164656e-01 1.913885e-02 6.634443e-02 9.474987e-02
2.915087e-02
##          285          286          287          288          289
291
## 8.066292e-02 7.923320e-02 1.148553e-01 1.138796e-01 4.295451e-01
1.361976e-01
##          292          293          295          296          297
298
## 2.422470e-01 6.389221e-08 2.834210e-01 1.088983e-01 1.352075e-01
4.421943e-01
##          299          302          303          304          305
306
## 1.081833e-01 3.501333e-02 1.976446e-01 1.532303e-01 6.402083e-02
1.129776e-01
##          308          309          310          312          313
314
## 1.232557e-01 8.953267e-02 7.994164e-02 9.183286e-02 2.067867e-01
1.165480e-01
##          315          316          319          320          321
322
## 1.848784e-01 2.022857e-01 8.579839e-02 1.157016e-02 2.226277e-01
6.807046e-02
##          323          325          326          327          329
330
## 7.937344e-02 4.155550e-02 7.208965e-03 1.526670e-01 1.462120e-01
5.928026e-02
##          331          332          333          336          337
338
## 3.731696e-02 5.786913e-02 7.606859e-02 8.617946e-02 1.576282e-01
1.472018e-01
##          339          340          342          343          344
346
## 5.226116e-02 1.184043e-01 8.247275e-02 1.308726e-01 1.241559e-01
5.656586e-01
##          347          348          349          350          353
354
## 1.104491e-01 2.955094e-01 1.098571e-01 5.654319e-03 1.349788e-02
5.923665e-02
##          355          356          357          359          360
361
## 5.718804e-02 1.025151e-01 3.593093e-01 1.279055e-01 5.614757e-02
```

```
1.310811e-01
##          363          364          365          366          367
370
## 3.602838e-01 1.613167e-01 1.680713e-01 1.219306e-01 8.388680e-02
8.565278e-02
##          371          372          373          374          376
377
## 1.063537e-01 4.586356e-02 8.895595e-02 7.256814e-01 6.274914e-02
6.161964e-02
##          378          380          381          382          383
384
## 1.197857e-01 1.073616e-01 1.138013e-01 4.627649e-02 1.229746e-01
3.412311e-02
##          387          388          389          390          391
393
## 2.795678e-01 1.164616e-01 2.464913e-01 4.146143e-01 1.034826e-01
2.534894e-01
##          394          395          397          398          399
400
## 9.711942e-02 1.678380e-01 5.616655e-02 8.124317e-02 1.166192e-01
8.003204e-02
##          401          404          405          406          407
408
## 2.757069e-02 1.132803e-01 2.694429e-01 2.519493e-08 7.206242e-02
1.665778e-01
##          410          411          412          414          415
416
## 7.494754e-02 2.054893e-01 2.746506e-01 1.471190e-01 1.205709e-01
2.156125e-02
##          417          418          421          422          423
424
## 2.147515e-01 4.364347e-02 3.111636e-01 3.420630e-01 1.008647e-01
4.699953e-02
##          425          427          428          429          431
432
## 1.966650e-01 2.471998e-01 5.189285e-02 1.736524e-01 8.261827e-02
1.122630e-01
##          433          434          435          438          439
440
## 6.454238e-02 1.250300e-01 7.843992e-02 1.073693e-01 1.186243e-01
1.379159e-01
##          441          442          444          445          446
448
## 1.720875e-01 4.374357e-02 3.464447e-02 1.492523e-02 7.192786e-02
2.229532e-01
##          449          450          451          452          455
456
## 9.585091e-02 1.278963e-01 5.352113e-02 1.667358e-01 5.883086e-02
1.580380e-01
##          457          458          459          461          462
```

```
463
## 1.317175e-01 8.141729e-02 2.703658e-02 4.462500e-02 1.132793e-01
1.270542e-01
##          465          466          467          468          469
## 2.741168e-01 2.763209e-01 5.646663e-02 9.063997e-02 1.908312e-01
```

```
confmatrix <- table(Actual_value=train$Risk1Yr, Predicted_value = res > 0.50)
confmatrix
```

```
##              Predicted_value
## Actual_value FALSE TRUE
##            F   296    6
##            T    55    2
```

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

```
## [1] 0.8300836
```

- Accuracy is 83%

Fit a Logistic Regression Model

- Fit a logistic regression model to the binary-classifier-data.csv dataset**

```
binary_data <- read.csv("binary-classifier-data.csv")
binary_model <- glm(label ~ x + y, data = binary_data, family = binomial())
summary(binary_model)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = binomial(), data = binary_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3728  -1.1697  -0.9575   1.1646   1.3989
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.424809   0.117224   3.624  0.00029 ***
## x           -0.002571   0.001823  -1.411  0.15836
## y           -0.007956   0.001869  -4.257 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2075.8  on 1497  degrees of freedom
## Residual deviance: 2052.1  on 1495  degrees of freedom
## AIC: 2058.1
##
## Number of Fisher Scoring iterations: 4
```

- What is the accuracy of the logistic regression classifier?

```
## [1] 0.6012146
```