"Analysis of Thoracic Surgery Binary Dataset"

"Ghanta, Madhavi"[*]

May 18, 2023

The results below are generated from an R script.

```
#date: "2023-05-18"

# a. For this problem, you will be working with the thoracic surgery data set
# from the University of California Irvine machine learning repository. This da
# taset contains information on life expectancy in lung cancer patients after s
# urgery. The underlying thoracic surgery data is in ARFF format. This is a tex
# t-based format with information on each of the attributes. You can load this
# data using a package such as foreign or by cutting and pasting the data secti
# on into a CSV file

## Load the foreign package
library(foreign)
library(caTools)

## Set the working directory to the root of your DSC 520 directory Week10 folder
setwd('C:/Users/mghan/Documents/dsc520/week10')

thoraric_surgery_df <- read.arff("C:/Users/mghan/Documents/dsc520/week10/ThoraricSurgery.arff")
str(thoraric_surgery_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 ...

head(thoraric_surgery_df)
```

[*] This report is automatically generated with the R package **knitr** (version 1.42).

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30 PRE32 AGE
## 1 DGN2 2.88 2.16 PRZ1    F    F    F     T     T  OC14     F     F     F     T     F  60
## 2 DGN3 3.40 1.88 PRZ0    F    F    F     F     F  OC12     F     F     F     T     F  51
## 3 DGN3 2.76 2.08 PRZ1    F    F    F     T     F  OC11     F     F     F     T     F  59
## 4 DGN3 3.68 3.04 PRZ0    F    F    F     F     F  OC11     F     F     F     F     F  54
## 5 DGN3 2.44 0.96 PRZ2    F    T    F     T     T  OC11     F     F     F     T     F  73
## 6 DGN3 2.48 1.88 PRZ1    F    F    F     T     F  OC11     F     F     F     F     F  51
##   Risk1Yr
## 1       F
## 2       F
## 3       F
## 4       F
## 5       T
## 6       F
```

```
# i.Fit a binary logistic regression model to the data set that predicts whet
# her 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.

#Fit the binary logistic regression model to the data set
mymodel <-glm(Risk1Yr ~ .,data = thoraric_surgery_df, family = 'binomial')

summary(mymodel)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = "binomial", data = thoraric_surgery_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
```

```
# ii. According to the summary, which variables had the greatest
# effect on the survival rate?
# As all the below variables have less p-value, it looks like
# below are the g  ood predictors for the whether or not the
# patient Risk1Y variable) after the surgery.
# PRE5,PRE9T,PRE14OC13,PRE14OC14,PRE17T,PRE30T

# iii. 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?

#Split the data into test and train datasets
split <- sample.split(thoraric_surgery_df,SplitRatio = 0.8)
split
```

```
##  [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## [15]  TRUE FALSE  TRUE
```

```
train<- subset(thoraric_surgery_df,split=="TRUE")
test<- subset(thoraric_surgery_df,split=="FALSE")

#run the test data through model
res<- predict(mymodel,test,type="response")
res
```

```
##          1          9         14         16         18         26         31         33
## 0.56996561 0.12650827 0.49084339 0.07638833 0.16865938 0.27597072 0.37307988 0.54019801
##         35         43         48         50         52         60         65         67
## 0.04321161 0.10224121 0.11283347 0.02634907 0.05705188 0.08436518 0.20688994 0.03426478
##         69         77         82         84         86         94         99        101
## 0.12151498 0.15174010 0.36422406 0.06808071 0.09959463 0.03580610 0.05044656 0.06405787
##        103        111        116        118        120        128        133        135
## 0.11026111 0.12344491 0.29223071 0.26863091 0.17645993 0.32860489 0.18019052 0.07935226
##        137        145        150        152        154        162        167        169
## 0.29337338 0.18246914 0.06588596 0.07084935 0.13998974 0.07273292 0.18134988 0.18633203
##        171        179        184        186        188        196        201        203
## 0.08981011 0.16911596 0.12089683 0.49744155 0.10817293 0.14144135 0.13532270 0.34903196
##        205        213        218        220        222        230        235        237
## 0.03045425 0.34479024 0.07094838 0.06582535 0.11944674 0.25582654 0.13170567 0.15676337
##        239        247        252        254        256        264        269        271
## 0.40820539 0.07865337 0.12353852 0.09485861 0.03947346 0.03270853 0.49791784 0.18286705
##        273        281        286        288        290        298        303        305
## 0.04705393 0.09474987 0.07923320 0.11387957 0.09208997 0.44219425 0.19764461 0.06402083
##        307        315        320        322        324        332        337        339
## 0.62606566 0.18487837 0.01157016 0.06807046 0.36513781 0.05786913 0.15762821 0.05226116
##        341        349        354        356        358        366        371        373
## 0.05243980 0.10985711 0.05923665 0.10251512 0.11827333 0.12193064 0.10635368 0.08895595
##        375        383        388        390        392        400        405        407
## 0.12128941 0.12297463 0.11646164 0.41461433 0.27197049 0.08003204 0.26944288 0.07206242
##        409        417        422        424        426        434        439        441
## 0.24683265 0.21475154 0.34206296 0.04699953 0.12285413 0.12503003 0.11862427 0.17208748
##        443        451        456        458        460        468
## 0.19023509 0.05352113 0.15803800 0.08141729 0.04519309 0.09063997
```

```
#run the train data through model
res<- predict(mymodel,train,type="response")
res
```

```
##            2            3            4            5            6            7
## 1.031988e-01 8.287068e-02 2.160824e-02 1.692634e-01 3.415054e-02 1.918605e-01
##            8           10           11           12           13           15
## 1.068699e-01 9.458663e-02 8.295347e-02 4.978455e-02 1.154378e-01 8.528088e-02
##           17           19           20           21           22           23
## 2.298384e-01 1.170482e-01 6.346676e-02 7.899455e-02 1.358877e-01 1.166706e-01
##           24           25           27           28           29           30
## 5.824619e-02 4.628603e-01 7.223499e-02 1.044741e-01 1.225337e-01 5.945905e-08
##           32           34           36           37           38           39
## 3.210049e-02 1.222741e-01 8.141605e-02 1.247959e-01 1.985475e-01 5.379752e-02
##           40           41           42           44           45           46
## 5.736768e-02 3.831235e-01 1.723143e-01 6.839303e-01 1.886592e-01 7.698128e-02
##           47           49           51           53           54           55
## 8.354285e-02 1.528144e-01 3.990471e-02 5.605594e-01 1.268064e-01 9.604222e-02
##           56           57           58           59           61           62
## 1.518051e-01 1.040492e-01 3.868351e-01 9.091183e-02 1.882038e-01 1.775659e-01
##           63           64           66           68           70           71
## 4.497232e-02 5.221406e-02 4.547291e-02 2.306748e-01 1.235686e-01 1.769600e-02
##           72           73           74           75           76           78
## 2.044482e-01 5.872367e-02 1.854511e-02 5.622961e-02 3.214431e-01 1.088240e-01
##           79           80           81           83           85           87
## 1.454896e-01 3.573413e-02 1.007965e-01 1.092554e-01 8.282431e-02 1.516943e-01
##           88           89           90           91           92           93
## 2.220150e-01 6.230735e-01 1.389749e-01 1.475171e-01 7.598004e-02 1.018244e-01
##           95           96           97           98          100          102
## 2.064928e-01 5.670370e-02 1.650967e-01 8.663401e-08 3.001414e-01 3.957982e-01
##          104          105          106          107          108          109
## 2.874635e-08 3.097683e-02 1.314217e-01 1.343593e-01 1.068128e-01 2.236160e-02
##          110          112          113          114          115          117
## 2.980639e-01 2.098142e-01 1.482006e-02 4.971735e-02 1.245632e-01 2.340033e-01
##          119          121          122          123          124          125
## 6.225151e-02 3.945990e-02 9.033179e-02 6.199320e-01 8.917611e-02 1.457683e-01
##          126          127          129          130          131          132
## 1.099803e-01 5.418171e-02 4.130719e-01 8.031190e-02 6.957820e-02 1.221660e-01
##          134          136          138          139          140          141
## 8.439071e-02 7.695837e-02 3.812039e-01 1.332096e-01 2.572193e-02 1.500561e-01
##          142          143          144          146          147          148
## 9.231166e-02 1.029460e-02 1.677159e-01 9.334413e-02 2.010585e-02 1.100579e-01
##          149          151          153          155          156          157
## 8.884902e-02 4.217588e-02 4.472309e-02 1.027427e-01 9.794784e-02 4.854969e-01
##          158          159          160          161          163          164
## 1.019523e-07 1.867933e-01 9.485986e-02 3.309436e-02 2.214874e-01 7.306653e-02
##          165          166          168          170          172          173
## 4.378233e-01 3.826184e-01 1.147794e-01 3.319553e-01 3.371654e-01 4.754743e-01
##          174          175          176          177          178          180
## 8.801868e-02 1.701133e-01 3.810037e-01 3.419036e-01 1.155253e-01 2.023070e-01
##          181          182          183          185          187          189
## 1.555587e-01 7.226418e-02 7.236749e-02 2.770187e-02 7.037954e-02 8.370741e-02
##          190          191          192          193          194          195
## 9.786972e-02 1.071501e-07 7.315314e-02 5.107552e-02 8.899037e-02 6.161650e-02
##          197          198          199          200          202          204
## 1.467324e-01 4.208491e-02 3.568805e-02 1.827940e-01 7.811592e-02 1.466339e-01
##          206          207          208          209          210          211
## 1.172731e-01 5.645845e-02 8.096561e-02 7.137263e-02 3.416674e-01 4.821277e-02
##          212          214          215          216          217          219
## 1.035481e-01 2.562132e-01 7.482114e-02 1.935358e-01 1.778609e-01 5.571797e-02
##          221          223          224          225          226          227
## 7.270148e-01 2.586989e-01 5.110705e-02 8.371578e-02 3.768849e-01 1.733864e-01
##          228          229          231          232          233          234
## 1.206525e-01 2.726272e-02 1.897757e-01 5.557867e-01 8.326085e-02 1.282731e-01
##          236          238          240          241          242          243
## 8.638962e-02 1.013461e-01 1.033867e-01 4.409613e-02 6.391354e-02 4.370160e-01
##          244          245          246          248          249          250
## 3.604740e-02 3.259522e-08 7.021216e-02 1.397018e-01 1.168226e-01 1.146856e-01
##          251          253          255          257          258          259
## 9.038743e-02 9.386811e-02 7.640224e-02 8.482854e-02 7.348739e-02 8.010688e-02
##          260          261          262          263          265          266
## 9.248713e-02 1.134974e-01 1.358705e-01 1.392593e-01 8.239156e-02 1.027026e-01
##          267          268          270          272          274          275
## 8.726133e-02 3.207561e-01 1.011537e-01 3.733253e-01 3.399052e-01 1.567863e-01
##          276          277          278          279          280          282
## 1.394679e-01 1.087993e-01 2.164656e-01 1.913885e-02 6.634443e-02 2.915087e-02
##          283          284          285          287          289          291
## 7.344261e-02 2.368618e-01 8.066292e-02 1.148553e-01 4.295451e-01 1.361976e-01
##          292          293          294          295          296          297
## 2.422470e-01 6.389221e-08 7.516974e-02 2.834210e-01 1.088983e-01 1.352075e-01
##          299          300          301          302          304          306
## 1.081833e-01 9.709489e-02 1.561671e-01 3.501333e-02 1.532303e-01 1.129776e-01
##          308          309          310          311          312          313
## 1.232557e-01 8.953267e-02 7.994164e-02 3.219110e-02 9.183286e-02 2.067867e-01
##          314          316          317          318          319          321
## 1.165480e-01 2.022857e-01 3.778067e-02 3.285881e-01 8.579839e-02 2.226277e-01
##          323          325          326          327          328          329
## 7.937344e-02 4.155550e-02 7.208965e-03 1.526670e-01 1.666427e-01 1.462120e-01
##          330          331          333          334          335          336
## 5.928026e-02 3.731696e-02 7.606859e-02 4.020393e-02 1.420674e-01 8.617946e-02
##          338          340          342          343          344          345
## 1.472018e-01 1.184043e-01 8.247275e-02 1.308726e-01 1.241559e-01 9.590097e-02
##          346          347          348          350          351          352
## 5.656586e-01 1.104491e-01 2.955094e-01 5.654319e-03 1.324475e-01 7.237318e-02
##          353          355          357          359          360          361
## 1.349788e-02 5.718804e-02 3.593093e-01 1.279055e-01 5.614757e-02 1.310811e-01
##          362          363          364          365          367          368
## 8.812173e-02 3.602838e-01 1.613167e-01 1.680713e-01 8.388680e-02 7.446550e-01
##          369          370          372          374          376          377
## 9.387401e-08 8.565278e-02 4.586356e-02 7.256814e-01 6.274914e-02 6.161964e-02
##          378          379          380          381          382          384
## 1.197857e-01 7.570812e-02 1.073616e-01 1.138013e-01 4.627649e-02 3.412311e-02
##          385          386          387          389          391          393
## 5.307208e-02 2.491018e-01 2.795678e-01 2.464913e-01 1.034826e-01 2.534894e-01
##          394          395          396          397          398          399
## 9.711942e-02 1.678380e-01 2.298356e-01 5.616655e-02 8.124317e-02 1.166192e-01
##          401          402          403          404          406          408
## 2.757069e-02 2.984281e-02 1.238295e-01 1.132803e-01 2.519493e-08 1.665778e-01
##          410          411          412          413          414          415
## 7.494754e-02 2.054893e-01 2.746506e-01 2.333291e-02 1.471190e-01 1.205709e-01
##          416          418          419          420          421          423
## 2.156125e-02 4.364347e-02 1.413123e-01 2.844515e-01 3.111636e-01 1.008647e-01
##          425          427          428          429          430          431
## 1.966650e-01 2.471998e-01 5.189285e-02 1.736524e-01 4.688095e-01 8.261827e-02
##          432          433          435          436          437          438
## 1.122630e-01 6.454238e-02 7.843992e-02 8.168373e-02 2.592223e-01 1.073693e-01
##          440          442          444          445          446          447
## 1.379159e-01 4.374357e-02 3.464447e-02 1.492523e-02 7.192786e-02 5.371397e-01
##          448          449          450          452          453          454
## 2.229532e-01 9.585091e-02 1.278963e-01 1.667358e-01 3.479825e-01 1.344147e-01
##          455          457          459          461          462          463
## 5.883086e-02 1.317175e-01 2.703658e-02 4.462500e-02 1.132793e-01 1.270542e-01
##          464          465          466          467          469          470
## 4.422608e-01 2.741168e-01 2.763209e-01 5.646663e-02 1.908312e-01 7.494837e-02
```

```
#Validate the model - confusion Matrix
confmatrix <- table(Actual_Value=train$Risk1Yr,Predicted_Value = res >0.5)
confmatrix
```

```
##             Predicted_Value
## Actual_Value FALSE TRUE
##            F   305    7
##            T    45    3
```

```
#Accuracy of the model
(confmatrix[[1,1]] + confmatrix[[2,2]]) / sum(confmatrix)
```

```
## [1] 0.8555556
```

```
#The accuracy of the model is 84.95%
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8  LC_CTYPE=C
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] caTools_1.18.2 foreign_0.8-83
##
## loaded via a namespace (and not attached):
##  [1] rstudioapi_0.14 knitr_1.42      magrittr_2.0.3  hms_1.1.3       R6_2.5.1
##  [6] rlang_1.1.0     fastmap_1.1.1   fansi_1.0.4     highr_0.10      tools_4.2.2
## [11] xfun_0.38       tinytex_0.45    utf8_1.2.3      cli_3.6.1       htmltools_0.5.5
## [16] yaml_2.3.7      digest_0.6.31   tibble_3.2.1    lifecycle_1.0.3 readr_2.1.4
## [21] tzdb_0.3.0      vctrs_0.6.1     bitops_1.0-7    glue_1.6.2      evaluate_0.20
## [26] rmarkdown_2.21  compiler_4.2.2  pillar_1.9.0    pkgconfig_2.0.3
```

```
Sys.time()
```

```
## [1] "2023-05-18 15:02:52 PDT"
```