```
# Assignment: ASSIGNMENT 10.2.1
# Name: Anjale, Jiteshwar
# Date: 2021-04-18
#Analysis of Thoracic Surgery Binary Dataset

#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)

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

setwd('C:/Users/anjal/OneDrive/Desktop/MS/DSC520/dsc520')

# Load the `data/ThoracicSurgery.arff` to thoraric_surgery_df
thoraric_surgery_df <- read.arff("C:/Users/anjal/OneDrive/Desktop/MS/DSC520/d
sc520/data/ThoracicSurgery.arff")

# Examine the structure of `thoraric_surgery_df` using `str()`
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 ...

# Show the top rows of thoraric_surgery_df
head(thoraric_surgery_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 P
RE30
## 1 DGN2 2.88 2.16 PRZ1    F    F    F     T     T  OC14     F     F     F
T
## 2 DGN3 3.40 1.88 PRZ0    F    F    F     F     F  OC12     F     F     F
T
## 3 DGN3 2.76 2.08 PRZ1    F    F    F     T     F  OC11     F     F     F
T
## 4 DGN3 3.68 3.04 PRZ0    F    F    F     F     F  OC11     F     F     F
F
## 5 DGN3 2.44 0.96 PRZ2    F    T    F     T     T  OC11     F     F     F
T
## 6 DGN3 2.48 1.88 PRZ1    F    F    F     T     F  OC11     F     F     F
F
##   PRE32 AGE Risk1Yr
## 1     F  60       F
## 2     F  51       F
## 3     F  59       F
## 4     F  54       F
## 5     F  73       T
## 6     F  51       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 Gener
alized Linear Models for an example. Include a summary using the summary() fu
nction in your results.

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

# View the summary of the model
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 th
e 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 survived for one year (the
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]  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE
## [13] FALSE  TRUE FALSE  TRUE  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
```

```
##            3            5           13           15           20
22
## 8.287068e-02 1.692634e-01 1.154378e-01 8.528088e-02 6.346676e-02 1.358877e
-01
##           30           32           37           39           47
49
## 5.945905e-08 3.210049e-02 1.247959e-01 5.379752e-02 8.354285e-02 1.528144e
-01
##           54           56           64           66           71
73
## 1.268064e-01 1.518051e-01 5.221406e-02 4.547291e-02 1.769600e-02 5.872367e
-02
##           81           83           88           90           98
100
## 1.007965e-01 1.092554e-01 2.220150e-01 1.389749e-01 8.663401e-08 3.001414e
-01
##          105          107          115          117          122
124
## 3.097683e-02 1.343593e-01 1.245632e-01 2.340033e-01 9.033179e-02 8.917611e
-02
##          132          134          139          141          149
151
## 1.221660e-01 8.439071e-02 1.332096e-01 1.500561e-01 8.884902e-02 4.217588e
-02
##          156          158          166          168          173
175
## 9.794784e-02 1.019523e-07 3.826184e-01 1.147794e-01 4.754743e-01 1.701133e
-01
##          183          185          190          192          200
202
## 7.236749e-02 2.770187e-02 9.786972e-02 7.315314e-02 1.827940e-01 7.811592e
-02
##          207          209          217          219          224
226
## 5.645845e-02 7.137263e-02 1.778609e-01 5.571797e-02 5.110705e-02 3.768849e
-01
##          234          236          241          243          251
253
## 1.282731e-01 8.638962e-02 4.409613e-02 4.370160e-01 9.038743e-02 9.386811e
-02
##          258          260          268          270          275
277
## 7.348739e-02 9.248713e-02 3.207561e-01 1.011537e-01 1.567863e-01 1.087993e
-01
##          285          287          292          294          302
304
```

```
## 8.066292e-02 1.148553e-01 2.422470e-01 7.516974e-02 3.501333e-02 1.532303e
-01
##          309          311          319          321          326
328
## 8.953267e-02 3.219110e-02 8.579839e-02 2.226277e-01 7.208965e-03 1.666427e
-01
##          336          338          343          345          353
355
## 8.617946e-02 1.472018e-01 1.308726e-01 9.590097e-02 1.349788e-02 5.718804e
-02
##          360          362          370          372          377
379
## 5.614757e-02 8.812173e-02 8.565278e-02 4.586356e-02 6.161964e-02 7.570812e
-02
##          387          389          394          396          404
406
## 2.795678e-01 2.464913e-01 9.711942e-02 2.298356e-01 1.132803e-01 2.519493e
-08
##          411          413          421          423          428
430
## 2.054893e-01 2.333291e-02 3.111636e-01 1.008647e-01 5.189285e-02 4.688095e
-01
##          438          440          445          447          455
457
## 1.073693e-01 1.379159e-01 1.492523e-02 5.371397e-01 5.883086e-02 1.317175e
-01
##          462          464
## 1.132793e-01 4.422608e-01

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

##            1            2            4            6            7
8
## 5.699656e-01 1.031988e-01 2.160824e-02 3.415054e-02 1.918605e-01 1.068699e
-01
##            9           10           11           12           14
16
## 1.265083e-01 9.458663e-02 8.295347e-02 4.978455e-02 4.908434e-01 7.638833e
-02
##           17           18           19           21           23
24
## 2.298384e-01 1.686594e-01 1.170482e-01 7.899455e-02 1.166706e-01 5.824619e
-02
##           25           26           27           28           29
31
## 4.628603e-01 2.759707e-01 7.223499e-02 1.044741e-01 1.225337e-01 3.730799e
-01
##           33           34           35           36           38
```

```
40
## 5.401980e-01 1.222741e-01 4.321161e-02 8.141605e-02 1.985475e-01 5.736768e
-02
##           41           42           43           44           45
46
## 3.831235e-01 1.723143e-01 1.022412e-01 6.839303e-01 1.886592e-01 7.698128e
-02
##           48           50           51           52           53
55
## 1.128335e-01 2.634907e-02 3.990471e-02 5.705188e-02 5.605594e-01 9.604222e
-02
##           57           58           59           60           61
62
## 1.040492e-01 3.868351e-01 9.091183e-02 8.436518e-02 1.882038e-01 1.775659e
-01
##           63           65           67           68           69
70
## 4.497232e-02 2.068899e-01 3.426478e-02 2.306748e-01 1.215150e-01 1.235686e
-01
##           72           74           75           76           77
78
## 2.044482e-01 1.854511e-02 5.622961e-02 3.214431e-01 1.517401e-01 1.088240e
-01
##           79           80           82           84           85
86
## 1.454896e-01 3.573413e-02 3.642241e-01 6.808071e-02 8.282431e-02 9.959463e
-02
##           87           89           91           92           93
94
## 1.516943e-01 6.230735e-01 1.475171e-01 7.598004e-02 1.018244e-01 3.580610e
-02
##           95           96           97           99          101
102
## 2.064928e-01 5.670370e-02 1.650967e-01 5.044656e-02 6.405787e-02 3.957982e
-01
##          103          104          106          108          109
110
## 1.102611e-01 2.874635e-08 1.314217e-01 1.068128e-01 2.236160e-02 2.980639e
-01
##          111          112          113          114          116
118
## 1.234449e-01 2.098142e-01 1.482006e-02 4.971735e-02 2.922307e-01 2.686309e
-01
##          119          120          121          123          125
126
## 6.225151e-02 1.764599e-01 3.945990e-02 6.199320e-01 1.457683e-01 1.099803e
-01
##          127          128          129          130          131
133
## 5.418171e-02 3.286049e-01 4.130719e-01 8.031190e-02 6.957820e-02 1.801905e
```

```
-01
##          135          136          137          138          140
142
## 7.935226e-02 7.695837e-02 2.933734e-01 3.812039e-01 2.572193e-02 9.231166e
-02
##          143          144          145          146          147
148
## 1.029460e-02 1.677159e-01 1.824691e-01 9.334413e-02 2.010585e-02 1.100579e
-01
##          150          152          153          154          155
157
## 6.588596e-02 7.084935e-02 4.472309e-02 1.399897e-01 1.027427e-01 4.854969e
-01
##          159          160          161          162          163
164
## 1.867933e-01 9.485986e-02 3.309436e-02 7.273292e-02 2.214874e-01 7.306653e
-02
##          165          167          169          170          171
172
## 4.378233e-01 1.813499e-01 1.863320e-01 3.319553e-01 8.981011e-02 3.371654e
-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
##          181          182          184          186          187
188
## 1.555587e-01 7.226418e-02 1.208968e-01 4.974416e-01 7.037954e-02 1.081729e
-01
##          189          191          193          194          195
196
## 8.370741e-02 1.071501e-07 5.107552e-02 8.899037e-02 6.161650e-02 1.414413e
-01
##          197          198          199          201          203
204
## 1.467324e-01 4.208491e-02 3.568805e-02 1.353227e-01 3.490320e-01 1.466339e
-01
##          205          206          208          210          211
212
## 3.045425e-02 1.172731e-01 8.096561e-02 3.416674e-01 4.821277e-02 1.035481e
-01
##          213          214          215          216          218
220
## 3.447902e-01 2.562132e-01 7.482114e-02 1.935358e-01 7.094838e-02 6.582535e
-02
##          221          222          223          225          227
228
## 7.270148e-01 1.194467e-01 2.586989e-01 8.371578e-02 1.733864e-01 1.206525e
-01
##          229          230          231          232          233
```

```
235
## 2.726272e-02 2.558265e-01 1.897757e-01 5.557867e-01 8.326085e-02 1.317057e
-01
##          237          238          239          240          242
244
## 1.567634e-01 1.013461e-01 4.082054e-01 1.033867e-01 6.391354e-02 3.604740e
-02
##          245          246          247          248          249
250
## 3.259522e-08 7.021216e-02 7.865337e-02 1.397018e-01 1.168226e-01 1.146856e
-01
##          252          254          255          256          257
259
## 1.235385e-01 9.485861e-02 7.640224e-02 3.947346e-02 8.482854e-02 8.010688e
-02
##          261          262          263          264          265
266
## 1.134974e-01 1.358705e-01 1.392593e-01 3.270853e-02 8.239156e-02 1.027026e
-01
##          267          269          271          272          273
274
## 8.726133e-02 4.979178e-01 1.828671e-01 3.733253e-01 4.705393e-02 3.399052e
-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
##          283          284          286          288          289
290
## 7.344261e-02 2.368618e-01 7.923320e-02 1.138796e-01 4.295451e-01 9.208997e
-02
##          291          293          295          296          297
298
## 1.361976e-01 6.389221e-08 2.834210e-01 1.088983e-01 1.352075e-01 4.421943e
-01
##          299          300          301          303          305
306
## 1.081833e-01 9.709489e-02 1.561671e-01 1.976446e-01 6.402083e-02 1.129776e
-01
##          307          308          310          312          313
314
## 6.260657e-01 1.232557e-01 7.994164e-02 9.183286e-02 2.067867e-01 1.165480e
-01
##          315          316          317          318          320
322
## 1.848784e-01 2.022857e-01 3.778067e-02 3.285881e-01 1.157016e-02 6.807046e
-02
##          323          324          325          327          329
330
## 7.937344e-02 3.651378e-01 4.155550e-02 1.526670e-01 1.462120e-01 5.928026e
```

```
-02
##          331          332          333          334          335
337
## 3.731696e-02 5.786913e-02 7.606859e-02 4.020393e-02 1.420674e-01 1.576282e
-01
##          339          340          341          342          344
346
## 5.226116e-02 1.184043e-01 5.243980e-02 8.247275e-02 1.241559e-01 5.656586e
-01
##          347          348          349          350          351
352
## 1.104491e-01 2.955094e-01 1.098571e-01 5.654319e-03 1.324475e-01 7.237318e
-02
##          354          356          357          358          359
361
## 5.923665e-02 1.025151e-01 3.593093e-01 1.182733e-01 1.279055e-01 1.310811e
-01
##          363          364          365          366          367
368
## 3.602838e-01 1.613167e-01 1.680713e-01 1.219306e-01 8.388680e-02 7.446550e
-01
##          369          371          373          374          375
376
## 9.387401e-08 1.063537e-01 8.895595e-02 7.256814e-01 1.212894e-01 6.274914e
-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
##          385          386          388          390          391
392
## 5.307208e-02 2.491018e-01 1.164616e-01 4.146143e-01 1.034826e-01 2.719705e
-01
##          393          395          397          398          399
400
## 2.534894e-01 1.678380e-01 5.616655e-02 8.124317e-02 1.166192e-01 8.003204e
-02
##          401          402          403          405          407
408
## 2.757069e-02 2.984281e-02 1.238295e-01 2.694429e-01 7.206242e-02 1.665778e
-01
##          409          410          412          414          415
416
## 2.468327e-01 7.494754e-02 2.746506e-01 1.471190e-01 1.205709e-01 2.156125e
-02
##          417          418          419          420          422
424
## 2.147515e-01 4.364347e-02 1.413123e-01 2.844515e-01 3.420630e-01 4.699953e
-02
##          425          426          427          429          431
```

```
432
## 1.966650e-01 1.228541e-01 2.471998e-01 1.736524e-01 8.261827e-02 1.122630e
-01
##          433          434          435          436          437
439
## 6.454238e-02 1.250300e-01 7.843992e-02 8.168373e-02 2.592223e-01 1.186243e
-01
##          441          442          443          444          446
448
## 1.720875e-01 4.374357e-02 1.902351e-01 3.464447e-02 7.192786e-02 2.229532e
-01
##          449          450          451          452          453
454
## 9.585091e-02 1.278963e-01 5.352113e-02 1.667358e-01 3.479825e-01 1.344147e
-01
##          456          458          459          460          461
463
## 1.580380e-01 8.141729e-02 2.703658e-02 4.519309e-02 4.462500e-02 1.270542e
-01
##          465          466          467          468          469
470
## 2.741168e-01 2.763209e-01 5.646663e-02 9.063997e-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   290    9
##            T    58    3

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

## [1] 0.8138889

#The accuracy of the model is 81.38%
```