Assignment 10.2

Aarti Ramani

2023-02-16

```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/Masters/GitHub/Winter2022/Ramani-DSC520")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1. Thoracic Surgery Binary Dataset

```
library(foreign)
thoraric_df <- read.arff("C:/Masters/GitHub/Winter2022/Ramani-DSC520/data/ThoraricSurgery.arff")
names(thoraric_df)
```

```
##  [1] "DGN"     "PRE4"    "PRE5"    "PRE6"    "PRE7"    "PRE8"    "PRE9"
##  [8] "PRE10"   "PRE11"   "PRE14"   "PRE17"   "PRE19"   "PRE25"   "PRE30"
## [15] "PRE32"   "AGE"     "Risk1Yr"
```

```
nrow(thoraric_df)
```

```
## [1] 470
```

```
head(thoraric_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30
## 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
```

```
#1. DGN:   Diagnosis - specific combination of ICD-10 codes for primary
#          and secondary as well multiple tumours if any
#          (DGN3,DGN2,DGN4,DGN6,DGN5,DGN8,DGN1)
#2. PRE4:  Forced vital capacity - FVC (numeric)
#3. PRE5:  Volume that has been exhaled at the end of the first second of
#          forced expiration - FEV1 (numeric)
#4. PRE6:  Performance status - Zubrod scale (PRZ2,PRZ1,PRZ0)
#5. PRE7:  Pain before surgery (T,F)
#6. PRE8:  Haemoptysis before surgery (T,F)
#7. PRE9:  Dyspnoea before surgery (T,F)
#8. PRE10: Cough before surgery (T,F)
#9. PRE11: Weakness before surgery (T,F)
#10.PRE14: T in clinical TNM - size of the original tumour,
#          from OC11 (smallest) to OC14 (largest) (OC11,OC14,OC12,OC13)
#11.PRE17: Type 2 DM - diabetes mellitus (T,F)
#12.PRE19: MI up to 6 months (T,F)
#13.PRE25: PAD - peripheral arterial diseases (T,F)
#14.PRE30: Smoking (T,F)
#15.PRE32: Asthma (T,F)
#16.AGE: Age at surgery (numeric)
#17.Risk1Y: 1 year survival period - (T)rue value if died (T,F)


#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.

result.0 <- glm(Risk1Yr ~ 1, data = thoraric_df, family = binomial())
summary(result.0)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ 1, family = binomial(), data = thoraric_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5679  -0.5679  -0.5679  -0.5679   1.9515
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7430     0.1296  -13.45   <2e-16 ***
## ---
## 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: 395.61  on 469  degrees of freedom
## AIC: 397.61
##
## Number of Fisher Scoring iterations: 4
```

```
result.1 <- glm(Risk1Yr ~ DGN + PRE4 +PRE5 +PRE6 +PRE7 +PRE8 +PRE9 +PRE10+
                PRE11 +PRE14 +PRE17 +PRE19 +PRE25 +PRE30 +PRE32+
                AGE ,data = thoraric_df, family=binomial(link="logit"))
summary(result.1)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 +
##     PRE9 + PRE10 + PRE11 + PRE14 + PRE17 + PRE19 + PRE25 + PRE30 +
##     PRE32 + AGE, family = binomial(link = "logit"), 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?
#
#Following variables have the greatest effect on the survival rate -
#PRE14OC14 - Size of the original tumour = 0.00668
#PRE9 - Dyspnoea before surgery  = 0.00494
#PRE17 - Type 2 DM - diabetes mellitus = 0.03709
#PRE30T - Smoking = 0.02984


#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?

# Add a column for T and F for predictions based on the probability above 0.5
thoraric_df$probability <- if_else(fitted(result.1)  > .5, T, F)
head(thoraric_df)
```

```
##    DGN PRE4 PRE5 PRE6 PRE7 PRE8 PRE9 PRE10 PRE11 PRE14 PRE17 PRE19 PRE25 PRE30
## 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 probability
## 1     F  60       F        TRUE
## 2     F  51       F       FALSE
## 3     F  59       F       FALSE
## 4     F  54       F       FALSE
## 5     F  73       T       FALSE
## 6     F  51       F       FALSE
```

```
# Compare predicted values with actual values
thoracic_compare <- table(actual=thoraric_df$Risk1Yr, predicted=thoraric_df$probability)
thoracic_compare
```

```
##       predicted
## actual FALSE TRUE
##      F   390   10
##      T    67    3
```

```
# Compute the accuracy
round((thoracic_compare[[1,1]] + thoracic_compare [[2,2]]) / sum(thoracic_compare),4)*100
```

```
## [1] 83.62
```

```
# the model is 83.62% accurate
```

2. binary-classifier-data.csv

```
binary_df <- read.csv("C:/Masters/GitHub/Winter2022/Ramani-DSC520/data/binary-classifier-data.csv")
names(binary_df)
```

```
## [1] "label" "x"     "y"
```

```
nrow(binary_df)
```

```
## [1] 1498
```

```
head(binary_df)
```

```
##   label        x        y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

```
binary_glm <- glm(label ~ x + y,data = binary_df, family=binomial(link="logit"))
binary_glm
```

```
## 
## Call:  glm(formula = label ~ x + y, family = binomial(link = "logit"), 
##     data = binary_df)
## 
## Coefficients:
## (Intercept)            x            y  
##    0.424809    -0.002571    -0.007956  
## 
## Degrees of Freedom: 1497 Total (i.e. Null);  1495 Residual
## Null Deviance:       2076 
## Residual Deviance: 2052  AIC: 2058
```

```
summary(binary_glm)
```

```
## 
## Call:
## glm(formula = label ~ x + y, family = binomial(link = "logit"), 
##     data = binary_df)
## 
## 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
```

```
# Add a column for T and F for predictions based on the probability above 0.5
binary_df$probability <- if_else(fitted(binary_glm)  > .5, T, F)
head(binary_df)
```

```
##   label        x        y probability
## 1     0 70.88469 83.17702       FALSE
## 2     0 74.97176 87.92922       FALSE
## 3     0 73.78333 92.20325       FALSE
## 4     0 66.40747 81.10617       FALSE
## 5     0 69.07399 84.53739       FALSE
## 6     0 72.23616 86.38403       FALSE
```

```
# Compare predicted values with actual values
binary_compare <- table(actual=binary_df$label, predicted=binary_df$probability)
binary_compare
```

```
##       predicted
## actual FALSE TRUE
##      0   429  338
##      1   286  445
```

```
# Compute the accuracy
round((binary_compare[[1,1]] + binary_compare [[2,2]]) / sum(binary_compare),4)*100
```

```
## [1] 58.34
```

```
# the model is 58.3% accurate
```