Exercise 9.2 Alan Donahue

Alan Donahue

8/8/2021

Part 1

Question 1

```
#setting the working directory
setwd("C:/Users/Alan Donahue/Documents/data science masters/DSC 520 Stats/GIT/dsc520")

#load the library
library(foreign)
library(caTools)

#load the data
surgery_df <- read.arff("data/ThoraricSurgery.arff")

head(surgery_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
```

```
#Question 1
#build the binary logistic regression model
surgery_logmod.1 <- glm(Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 + PRE9 + PRE10 + PRE11 + PRE14
                          + PRE19 + PRE25 + PRE30 + PRE32 + AGE, data = surgery_df, family = binomial())

#summary of model
summary(surgery_logmod.1)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 +
##     PRE9 + PRE10 + PRE11 + PRE14 + PRE17 + PRE19 + PRE25 + PRE30 +
##     PRE32 + AGE, family = binomial(), data = 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
```

Question 2

Based off the results, it looks like PRE9T, PRE14OC14, PRE17T, and PRE30T had the greatest effect on the survival rate.

Question 3

```
split <- sample.split(surgery_df, SplitRatio = 0.8)
split
```

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

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

surgery_logmod.2 <- glm(Risk1Yr ~ PRE9 + PRE14 + PRE17 + PRE30, data = train, family = "binomial")

res <- predict(surgery_logmod.2, test, type = "response")
head(res)
```

```
##          9         14         16         17         26         31
## 0.09982776 0.40389110 0.13916568 0.13916568 0.13916568 0.40389110
```

```
res <- predict(surgery_logmod.2, train, type = "response")
head(res)
```

```
##          1          2          3          4          5          6
## 0.40389110 0.13916568 0.09982776 0.01940042 0.09982776 0.01940042
```

```
confmatrix <- table(Actual_Value=train$Risk1Yr, Predicted_Value = res > 0.5)
confmatrix
```

```
##             Predicted_Value
## Actual_Value FALSE TRUE
##            F   314    1
##            T    46    0
```

```
accuracy <- ((confmatrix[[1,1]] + confmatrix[[2,2]]) / sum(confmatrix)) * 100
print(accuracy)
```

```
## [1] 86.98061
```

Part 2

```
#setting the working directory
setwd("C:/Users/Alan Donahue/Documents/data science masters/DSC 520 Stats/GIT/dsc520")

binary_df = read.csv("data/binary-classifier-data.csv")

#logistic regression model
binary_logmod.1 <- glm(label ~ x + y, data = binary_df, family = "binomial")

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

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

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

summary(binary_logmod.1)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = "binomial", 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
```

```
res <- predict(binary_logmod.1, test, type = "response")
head(res)
```

```
##         2         5         8        11        14        17
## 0.3852176 0.3952460 0.3637058 0.3943309 0.3844039 0.4003614
```

```
res <- predict(binary_logmod.1, train, type = "response")
head(res)
```

```
##         1         3         4         6         7         9
## 0.3967211 0.3779152 0.4034378 0.3898045 0.3842859 0.3782162
```

```
confmatrix.2 <- table(Actual_Value=train$label, Predicted_Value = res > .5)
confmatrix.2
```

```
##             Predicted_Value
## Actual_Value FALSE TRUE
##            0   285  226
##            1   188  300
```

```r
accuracy.2 <- ((confmatrix.2[[1,1]] + confmatrix.2[[2,2]]) / sum(confmatrix.2)) * 100
print(accuracy.2)
```

```
## [1] 58.55856
```