Week 10 assignment

Allison Forte

2022-05-21

Assignment 10.2

Part 1

Fit a Logistic Regression Model to Thoracic Surgery Binary Dataset

```
# load package
library(foreign)

# load data
surgery_df = read.arff('data/ThoracicSurgery.arff')

# split data into 2 sets to have a training and validating data set
library(caTools)
split <- sample.split(surgery_df, SplitRatio = 0.8)
surgery_train <- subset(surgery_df, split == "TRUE")
surgery_validate <- subset(surgery_df, split == "FALSE")
```

Assignment Instructions:

- 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.

```
library(glm2)
surgery_model <- glm(Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 +
            PRE7 + PRE8 + PRE9 + PRE10 + PRE11 +
            PRE14 + PRE17 + PRE19 + PRE25 + PRE30 +
            PRE32 + AGE, data = surgery_train, family = binomial()  )
summary(surgery_model)
```

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

```
##     Min       1Q   Median       3Q      Max
## -1.5659  -0.5353  -0.4248  -0.2228   2.5240
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -16.99267 2399.54550  -0.007  0.99435
## DGNDGN2       14.59008 2399.54481   0.006  0.99515
## DGNDGN3       14.32882 2399.54477   0.006  0.99524
## DGNDGN4       14.41328 2399.54484   0.006  0.99521
## DGNDGN5       16.83083 2399.54484   0.007  0.99440
## DGNDGN6        1.29321 2754.70946   0.000  0.99963
## DGNDGN8       35.31340 3393.46878   0.010  0.99170
## PRE4          -0.14428    0.31892  -0.452  0.65097
## PRE5          -0.12171    0.31179  -0.390  0.69627
## PRE6PRZ1      -0.11848    0.62136  -0.191  0.84877
## PRE6PRZ2      -1.67124    1.30406  -1.282  0.19999
## PRE7T          0.78079    0.64627   1.208  0.22699
## PRE8T          0.29258    0.46004   0.636  0.52478
## PRE9T          1.63187    0.57537   2.836  0.00456 **
## PRE10T         0.35792    0.57421   0.623  0.53307
## PRE11T         0.35601    0.46137   0.772  0.44033
## PRE14OC12      0.57616    0.38563   1.494  0.13516
## PRE14OC13      0.31001    0.81313   0.381  0.70301
## PRE14OC14      1.05017    0.96406   1.089  0.27602
## PRE17T         0.89770    0.54202   1.656  0.09768 .
## PRE19T       -14.11015 2399.54475  -0.006  0.99531
## PRE25T        -0.33446    1.09254  -0.306  0.75951
## PRE30T         1.53256    0.64049   2.393  0.01672 *
## PRE32T       -14.66567 2399.54477  -0.006  0.99512
## AGE           -0.01279    0.02199  -0.582  0.56082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 290.12  on 359  degrees of freedom
## Residual deviance: 242.55  on 335  degrees of freedom
## AIC: 292.55
##
## Number of Fisher Scoring iterations: 15
```

- According to the summary, which variables had the greatest effect on the survival rate?
  - According to the summary the variable with the greatest effect was PRE14 when the value was OC14
- 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?
  - The accuracy of the model is 84%

```r
# Compute the accuracy of your model

# Use validation data in model based on training data
val_res <- predict(surgery_model, surgery_validate, type = "response")
```

```r
# Compare results from validation data to results from training data (that the model was based on)
train_res <-predict(surgery_model, surgery_train, type = "response")

#Validate model using confusion matrix
confmatrix <- table(Actual_Value = surgery_train$Risk1Yr, Predicted_Value = train_res >0.5)
confmatrix
```

```
##              Predicted_Value
## Actual_Value FALSE TRUE
##            F   304    6
##            T    43    7
```

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

```
## [1] 0.8638889
```

Part 2

- Fit a logistic regression model to the binary-classifier-data.csv dataset
  - The dataset (found in binary-classifier-data.csv) contains three variables; label, x, and y. The label variable is either 0 or 1 and is the output we want to predict using the x and y variables.

```r
# load data
binary_df = read.csv('data/binary-classifier-data.csv')

# split data into 2 sets to have a training and validating data set
split <- sample.split(binary_df, SplitRatio = 0.8)
binary_train <- subset(binary_df, split == "TRUE")
binary_validate <- subset(binary_df, split == "FALSE")

#  fit a logistic regression model
binary_model <- glm(label ~ x + y, data = binary_train, family = binomial()  )
summary(binary_model)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = binomial(), data = binary_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3766  -1.1693  -0.9522   1.1648   1.3896
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.433172   0.143853   3.011 0.002602 **
## x           -0.002722   0.002231  -1.220 0.222475
## y           -0.008017   0.002286  -3.507 0.000453 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1384.3  on 998  degrees of freedom
## Residual deviance: 1368.0  on 996  degrees of freedom
## AIC: 1374
##
## Number of Fisher Scoring iterations: 4
```

- What is the accuracy of the logistic regression classifier?
  - The accuracy of this model is 58% with 80/20 split for training/validating data

```
# Compute the accuracy of your model

# Use validation data in model based on training data
val_res2 <- predict(binary_model, binary_validate, type = "response")

# Compare results from validation data to results from training data (that the model was based on)
train_res2 <-predict(binary_model, binary_train, type = "response")

#Validate model using confusion matrix
confmatrix <- table(Actual_Value = binary_train$label, Predicted_Value = train_res2 >0.5)
confmatrix
```

```
##             Predicted_Value
## Actual_Value FALSE TRUE
##            0   283  229
##            1   190  297
```

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

```
## [1] 0.5805806
```