Logistic Regression Assignment

David Berberena

02-18-2024

Thoracic Surgery Data Binomial Logistic Regression

```
# Assignment Start

# Upload foreign library for ARFF file importing

library(foreign)

# Set working directory for smooth file importing

setwd("C:/Users/dbzda/Documents/School/DSC 520 Statistics for Data Science")

# Import the ARFF file to view its properties

Thoracic <- read.arff("ThoracicSurgery.arff")
```

1A. 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.

```
# Load the caTools library to aid in splitting the Thoracic dataset for logistic
# regression purposes by creating a training dataset and a test dataset

library(caTools)

# A seed must be set so that the data being split will be split the same each
# time the code is run and subsequent values will be consistent

set.seed(123)

# Split the data into a training dataset and a test dataset using the
# sample.split() function in caTools and the subset() function with 80% of the
# dataset being the training dataset and the other 20% being the test dataset

split_data <- sample.split(Thoracic, SplitRatio = 0.8)
training_data <- subset(Thoracic, split_data == "TRUE")
test_data <- subset(Thoracic, split_data == "FALSE")
```

```r
# All categorical variables must be transformed into factors

Thoracic$DGN <- as.factor(Thoracic$DGN)
Thoracic$PRE6 <- as.factor(Thoracic$PRE6)
Thoracic$PRE7 <- as.factor(Thoracic$PRE7)
Thoracic$PRE8 <- as.factor(Thoracic$PRE8)
Thoracic$PRE9 <- as.factor(Thoracic$PRE9)
Thoracic$PRE10 <- as.factor(Thoracic$PRE10)
Thoracic$PRE11 <- as.factor(Thoracic$PRE11)
Thoracic$PRE14 <- as.factor(Thoracic$PRE14)
Thoracic$PRE17 <- as.factor(Thoracic$PRE17)
Thoracic$PRE19 <- as.factor(Thoracic$PRE19)
Thoracic$PRE25 <- as.factor(Thoracic$PRE25)
Thoracic$PRE30 <- as.factor(Thoracic$PRE30)
Thoracic$PRE32 <- as.factor(Thoracic$PRE32)

# Train the logistic regression model using the glm() function and the training
# dataset created earlier as the training_data variable

binary_model <- glm(Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 + PRE9 +
                      PRE10 + PRE11 + PRE14 + PRE17 + PRE19 + PRE25 + PRE30
                    + PRE32 + AGE, data = training_data, family = 'binomial')

# Output the summary of the model with the summary() function

summary(binary_model)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 +
##     PRE9 + PRE10 + PRE11 + PRE14 + PRE17 + PRE19 + PRE25 + PRE30 +
##     PRE32 + AGE, family = "binomial", data = training_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -15.68822 3956.18082  -0.004  0.99684
## DGNDGN2       15.65553 3956.18037   0.004  0.99684
## DGNDGN3       14.92123 3956.18036   0.004  0.99699
## DGNDGN4       15.60143 3956.18039   0.004  0.99685
## DGNDGN5       17.69319 3956.18042   0.004  0.99643
## DGNDGN6        0.73898 4411.93162   0.000  0.99987
## PRE4          -0.35934    0.23191  -1.549  0.12126
## PRE5          -0.03372    0.01919  -1.757  0.07885 .
## PRE6PRZ1      -0.15815    0.61653  -0.257  0.79755
## PRE6PRZ2       0.13993    0.93095   0.150  0.88052
## PRE7T          1.73385    0.63898   2.713  0.00666 **
## PRE8T         -0.36625    0.49519  -0.740  0.45954
## PRE9T          1.69685    0.55231   3.072  0.00212 **
## PRE10T         0.18413    0.53751   0.343  0.73193
## PRE11T         0.51735    0.47408   1.091  0.27516
## PRE14OC12      0.65126    0.42112   1.546  0.12199
## PRE14OC13      1.68596    0.68738   2.453  0.01418 *
## PRE14OC14      1.72044    0.72894   2.360  0.01827 *
```

```
## PRE17T         1.54993    0.51106   3.033  0.00242 **
## PRE19T       -14.54727 3956.18036  -0.004  0.99707
## PRE25T       -16.21022 1633.98543  -0.010  0.99208
## PRE30T         0.91085    0.55355   1.645  0.09988 .
## PRE32T       -14.88968 2705.63955  -0.006  0.99561
## AGE           -0.03194    0.02252  -1.418  0.15615
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 289.82  on 358  degrees of freedom
## Residual deviance: 235.00  on 335  degrees of freedom
## AIC: 283
##
## Number of Fisher Scoring iterations: 16
```

1B. According to the summary, which variables had the greatest effect on the survival rate?

Looking at each variable outlined in the summary of the logistic regression model, there are three of the variables that are annotated with the 0.001 significance code symbol. These significance code symbols are an indicator that the variable has a large effect on the outcome variable, which in this case is the survival rate variable Risk1Yr. For a variable to be annotated with a significance code, the absolute value of the estimate coefficient must be large and the p-value must be small together. The variables with the greatest effect are PRE7, PRE9, and PRE17,

1C. 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?

```r
# Running the test dataset through the created model can be done using the
# predict() function with the type argument set to "response" for fitted values

# When running this code before and after setting the seed, the number of levels
# in the DGN variable had changed from when the model was trained to this point,
# so I did some digging as to how to fix that, and I found that it was easiest
# to recall the levels of the training data variable used in the model and
# reassign those levels back to the variable using the levels() and factor()
# functions

training_levels <- levels(binary_model$DGN)
test_data$DGN <- factor(test_data$DGN, levels = training_levels)

test_values <- predict(binary_model, test_data, type = "response")
test_values <- predict(binary_model, training_data, type = "response")

# To visualize the performance of the model in its running of the test data, we
# will create a confusion matrix using the table() function

confusion_matrix <- table(act_value=training_data$Risk1Yr, pred_value=test_values>0.5)
confusion_matrix
```

```
##          pred_value
## act_value FALSE TRUE
##         F   301    8
##         T    43    7
```

```
# To compute the accuracy from the confusion matrix, we can now apply the
# accuracy formula: the sum of the true positive and true negative values
# divided by the sum of all values (true positive, true negative, false
# positive, and false negative). This can be done here with the help of the
# sum() function and the diag() function

model_accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
model_accuracy
```

```
## [1] 0.8579387
```

The accuracy of the binary logistic regression model is a rounded 85.79%.

Binary Classifier Data Binomial Logistic Regression

```
# Upload readr library for CSV file importing

library(readr)

# Import the Binary Classifier Data CSV file to view its properties

binary_dataset <- read_csv("binary_classifier_data.csv", show_col_types = FALSE)
```

2A. 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.

```
# To repeat the process of fitting a logistic regression model for this dataset,
# I will be employing most all of the same coding techniques used above for the
# first binomial logistic regression model

set.seed(123)

# When splitting the data, I needed to call the label outcome variable for
# sample.split() to function properly

split_data2 <- sample.split(binary_dataset$label, SplitRatio = 0.8)
training_data2 <- subset(binary_dataset, split_data2 == "TRUE")
test_data2 <- subset(binary_dataset, split_data2 == "FALSE")

binary_model2 <- glm(label ~ x + y, data = training_data2, family = 'binomial')

summary(binary_model2)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = "binomial", data = training_data2)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.483581   0.131812   3.669 0.000244 ***
## x           -0.002830   0.002047  -1.383 0.166805
## y           -0.009041   0.002116  -4.273 1.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1661.5  on 1198  degrees of freedom
## Residual deviance: 1637.6  on 1196  degrees of freedom
## AIC: 1643.6
##
## Number of Fisher Scoring iterations: 4
```

2B. What is the accuracy of the logistic regression classifier?

```
# The confusion matrix and accuracy will be realized the same way as above

test_values2 <- predict(binary_model2, test_data2, type = "response")
test_values2 <- predict(binary_model2, training_data2, type = "response")

confusion_matrix2 <- table(act_value=training_data2$label, pred_value=test_values2>0.5)
confusion_matrix2
```

```
##          pred_value
## act_value FALSE TRUE
##         0   344  270
##         1   220  365
```

```
model_accuracy2 <- sum(diag(confusion_matrix2)) / sum(confusion_matrix2)
model_accuracy2
```

```
## [1] 0.5913261
```

The accuracy of the logistic regression classifier is a rounded 59.13%.