Week 10 Exercise

Sarah Theriot

2024-10-30

Exercise 1

Step 1: Install and Load Required Packages

```r
# Install packages
if (!require(foreign)) install.packages("foreign")
```

```
## Loading required package: foreign
```

```r
if (!require(dplyr)) install.packages("dplyr")
```

```
## Loading required package: 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
```

```r
if (!require(caret)) install.packages("caret")
```

```
## Loading required package: caret
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
# Load necessary libraries
library(foreign)  # For reading ARFF files
library(dplyr)    # For data manipulation
library(caret)
```

Step 2: Load the Data

```r
# Load the dataset
thoracic_data <- read.arff("C:/Users/sarah/Desktop/MSDS/Statistics for Data Science/Week 10/ThoraricSurg

# Display the structure of the dataset
str(thoracic_data)
```

```
## '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 ...
```

```r
# Display the first few rows
head(thoracic_data)
```

```
##    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
```

Step 3: Fit the Logistic Regression Model

```r
# Fit the logistic regression model
model <- glm(Risk1Yr ~ ., data = thoracic_data, family = binomial)
```

```r
# View the summary of the model
summary(model)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = binomial, data = thoracic_data)
##
## 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
```

Step 4: Analyze the Results

I fit a binary logistic regression model to predict whether patients survived for one year after surgery using the Risk1Yr variable. The model summary showed that the variables PRE9T, PRE14OC14, PRE17T, and PRE30T had the greatest impact on survival rates, which indicates that these factors are important in determining patient outcomes after thoracic surgery.

Step 5: Make Predictions and Compute Accuracy

```
# Make predictions
predictions <- ifelse(predict(model, type = "response") > 0.5, 1, 0)

# Create a confusion matrix to compare predicted vs actual outcomes
confusion_matrix <- table(Predicted = predictions, Actual = thoracic_data$Risk1Yr)

# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy: ", round(accuracy * 100, 2), "%", sep = ""))
```

```
## [1] "Accuracy: 83.62%"
```

Conclusion:

The accuracy of my logistic regression model is 83.62%, meaning it correctly predicted the survival outcome for about 84% of the patients in the dataset. This suggests that the model is fairly reliable in determining which patients are likely to survive for one year after surgery.

Exercise 2

Step 1: Load the Dataset

```
# Load necessary libraries
library(dplyr)

# Read the dataset
binary_data <- read.csv("C:/Users/sarah/Desktop/MSDS/Statistics for Data Science/Week 10/binary-classif

# View the first few rows of the dataset
head(binary_data)
```

```
##   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
```

Step 2: Fit the Logistic Regression Model

```
# Fit the logistic regression model
model_binary <- glm(label ~ x + y, data = binary_data, family = binomial)

# View the summary of the model
summary(model_binary)
```

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

Step 3: Calculate the Accuracy

```
# Make predictions
predicted_probs_binary <- predict(model_binary, type = "response")
predicted_classes_binary <- ifelse(predicted_probs_binary > 0.5, 1, 0)

# Create a confusion matrix
confusion_matrix_binary <- table(Actual = binary_data$label, Predicted = predicted_classes_binary)

# Calculate accuracy
accuracy_binary <- sum(diag(confusion_matrix_binary)) / sum(confusion_matrix_binary)
print(paste("Accuracy: ", round(accuracy_binary * 100, 2), "%", sep = ""))
```

```
## [1] "Accuracy: 58.34%"
```

Conclusion

The accuracy of my logistic regression model is 58.34%, meaning it correctly predicted the outcome for about 58% of the cases in the dataset. This suggests that the model may need further improvement, as it isn’t very reliable in distinguishing between the two classes.