assignment_10_01_PothineniKalyan

PothineniKalyan

2023-05-18

10.2 Exercise 1

1

> 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(foreign)

# Download and load the dataset
# Library Foregin
# ref.https://stat.ethz.ch/R-manual/R-devel/library/foreign/html/read.arff.html
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/00277/ThoracicSurgery.arff"
download.file(url, "ThoracicSurgery.arff")
thoracic_df <- read.arff("ThoracicSurgery.arff")

# Rename the columns
colnames(thoracic_df) <- c('DGN', 'PRE4', 'PRE5', 'PRE6', 'PRE7', 'PRE8', 'PRE9', 'PRE10',
                  'PRE11', 'PRE14', 'PRE17', 'PRE19', 'PRE25', 'PRE30', 'PRE32', 'AGE', 'Risk1Y')

# Convert non-numeric columns to factors
thoracic_df$DGN <- as.factor(thoracic_df$DGN)
thoracic_df$Risk1Y <- as.factor(thoracic_df$Risk1Y)


# Fit a logistic regression model using 'glm
model <- glm(Risk1Y ~ ., data = thoracic_df, family = binomial(link = "logit"))

# Display the summary of the model
summary(model)
```

```
##
## Call:
## glm(formula = Risk1Y ~ ., family = binomial(link = "logit"),
##     data = thoracic_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?

To determine which variables had the greatest effect on the survival rate, we can look at the estimated coefficients of the logistic regression model. In the provided summary output, the "Coefficients" section displays the estimates, standard errors, z-values, and p-values for each predictor variable.

Variables with the significant effect on the survival rate typically have coefficients with a large magnitude (far from zero) and low p-values

Based on the summary output we got, it seems that the variables "PRE9T" and "PRE14OC14" have the greatest effect on the survival rate. These variables have estimated coefficients of 1.368 and 1.653, respectively, which are relatively large compared to other variables. Additionally, their

p-values are below the commonly used threshold of 0.05, indicating that these coefficients are statistically significant.

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?

To compute the accuracy of the logistic regression model, we need to compare the predicted outcomes (y_pred) with the actual outcomes in the dataset

```
# Compute accuracy
# Predict the outcome variable for the dataset
y_pred <- predict(model, type = "response")

# new vector, predicted probability in to binary values, with threshold of 0.5
y_pred <- ifelse(y_pred >= 0.5, 1, 0)

# Compute accuracy is caluclated by comparing predicted vs actual values
accuracy <- mean(y_pred == thoracic_df$Risk1Y) * 100

# Display the accuracy
accuracy
```

```
## [1] 0
```

Accuracy of zero means that the model did not make any correct predictions. It indicates that the predicted values (y_pred) did not match the actual values (thoracic_df$Risk1Y) at all.

A zero accuracy can occur due to various reasons, such as a poorly trained or ill-fitted model, imbalance in the dataset, or incorrect implementation of the evaluation code

To check if the implementation of the evaluation code is wrong, lets compare the values and calculate the percentage of correct predictions. By calculating the accuracy in this way, we can ensure that we obtain the correct percentage of correct prediction made by the logistic regression model

```
correct_predictions <- sum(y_pred == thoracic_df$Risk1Y)
total_predictions <- length(y_pred)
accuracy <- correct_predictions / total_predictions * 100

# Print the accuracy
# Print the accuracy
print(paste("Accuracy:", accuracy))
```

```
## [1] "Accuracy: 0"
```

2.Fit a Logistic Regrssion Model for binary classified data

To fit the Logistic Regrssion Model, we use 'glm()'

```r
# Read the dataset
binary_df <- read.csv('data/binary-classifier-data.csv')

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

# Print the summary of the model
summary(binary_model)
```

```
##
## 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
```

Accuracy

```r
binary_predictions <- ifelse(predict(binary_model, type = 'response') >= 0.5, 1, 0)

# Calculate the accuracy
binary_accuracy <- sum(binary_predictions == binary_df$label) / nrow(binary_df)

# Print the accuracy
print(paste("Accuracy:", binary_accuracy))
```

```
## [1] "Accuracy: 0.583444592790387"
```

The accuracy of the logistic regression classifier on binary-classifier-data.csv dataset is approximately 0.5834, or 58.34%. This means that the classifier correctly predicted the label (0 or 1) for 58.34% of the instances in the dataset