Week 10

Javier Corpus

2025-02-11

Fit a Logistic Regression Model to Thoracic Surgery Binary Dataset

For this problem, you will be working with the thoracic surgery data set from the University of California Irvine machine learning repository. This dataset contains information on life expectancy in lung cancer patients after surgery. The underlying thoracic surgery data is in ARFF format. This is a text-based format with information on each of the attributes. You can load this data using a package such as foreign or by cutting and pasting the data section into a CSV file.

Assignment Instructions:

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 required to read a file in ARFF format
library(foreign)

# Loading the data into a dataframe
df_ts <- read.arff("ThoracicSurgery.arff")

# Creating the model to predict Risk1Yr
model_ts <- glm(Risk1Yr ~ ., data = df_ts, family = binomial)

# Displaying a summary of the model
summary(model_ts)
```

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

2) According to the summary, which variables had the greatest effect on the survival rate?

```r
# Getting the odds ratios from the coefficients
odds_ratios <- exp(coef(model_ts))

# Creating a dataframe with the results, getting the name of the variable,
# coefficients, odd rations and the p-value
results <- data.frame(
  Variable = names(coef(model_ts)),
  Coefficient = coef(model_ts),
  Odds_Ratio = odds_ratios,
  P_Value = summary(model_ts)$coefficients[, 4]
)

# Sorting the `results` dataframe by the absolute value of
# coefficients, from greater to smaller.
results <- results[order(-abs(results$Coefficient)), ]
```

```r
# Removing the "(Intercept)" row.
results <- subset(results, Variable != "(Intercept)")

print(results)
```

```
##            Variable   Coefficient   Odds_Ratio     P_Value
## DGNDGN8     DGNDGN8  18.032862288 6.785355e+07 0.994003861
## DGNDGN5     DGNDGN5  16.381321125 1.301120e+07 0.994553008
## DGNDGN2     DGNDGN2  14.736275610 2.511211e+06 0.995099999
## PRE19T       PRE19T -14.655378361 4.317676e-07 0.992928420
## DGNDGN4     DGNDGN4  14.608328798 2.209615e+06 0.995142542
## DGNDGN3     DGNDGN3  14.180551971 1.440574e+06 0.995284782
## PRE32T       PRE32T -13.983294646 8.455364e-07 0.993218971
## PRE14OC14 PRE14OC14   1.652972956 5.222483e+00 0.006675215
## PRE9T         PRE9T   1.368216429 3.928338e+00 0.004941570
## PRE14OC13 PRE14OC13   1.179207421 3.251796e+00 0.055799144
## PRE30T       PRE30T   1.083997008 2.956473e+00 0.029840120
## PRE17T       PRE17T   0.926593415 2.525890e+00 0.037091709
## PRE7T         PRE7T   0.715340997 2.044884e+00 0.197883787
## PRE10T       PRE10T   0.576957854 1.780613e+00 0.231854799
## PRE11T       PRE11T   0.516180804 1.675616e+00 0.192947882
## PRE6PRZ1   PRE6PRZ1  -0.442714960 6.422903e-01 0.394477609
## PRE14OC12 PRE14OC12   0.439363853 1.551720e+00 0.183177019
## DGNDGN6     DGNDGN6   0.408853479 1.505091e+00 0.999877960
## PRE6PRZ2   PRE6PRZ2  -0.293700721 7.454996e-01 0.710303466
## PRE4           PRE4  -0.227244839 7.967257e-01 0.219094469
## PRE8T         PRE8T   0.174336579 1.190456e+00 0.654187785
## PRE25T       PRE25T  -0.097894464 9.067446e-01 0.922272899
## PRE5           PRE5  -0.030303535 9.701510e-01 0.089714907
## AGE             AGE  -0.009505656 9.905394e-01 0.599441514
```

The summary shows that variables PRE14OC14, PRE9T, PRE30T and PRE17T are all statistically significant. But the variables with the greatest effect on the survival date, according to the absolute value of their coefficients, are:

Variable	Coefficient
DGNDGN8	18.032862288
DGNDGN5	16.381321125
DGNDGN2	14.736275610
PRE19T	-14.655378361
DGNDGN4	14.608328798
DGNDGN3	14.180551971
PRE32T	-13.983294646

3) 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?

```
# Making predictions on the training data (the model)
predicted_probabilities <- predict(model_ts, type = "response")

# Converting probabilities to binary outcomes (using a threshold of 0.5)
predicted_classes <- ifelse(predicted_probabilities > 0.5, 1, 0)

# Creating a confusion matrix
confusion_matrix <- table(Predicted = predicted_classes, Actual =
df_ts$Risk1Yr)

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

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

Fit a Logistic Regression Model

1) Fit a logistic regression model to the binary-classifier-data.csv dataset

```
df_bcd <- read.csv("binary-classifier-data.csv")

model_bcd <- glm(label ~ ., data = df_bcd, family = binomial)
summary(model_bcd)
```

```
##
## Call:
## glm(formula = label ~ ., family = binomial, data = df_bcd)
##
## 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
```

2) 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.

a) What is the accuracy of the logistic regression classifier?

```
# Making predictions on the training data (the model)
predicted_probabilities <- predict(model_bcd, type = "response")

# Converting probabilities to binary outcomes (using a threshold of 0.5)
predicted_classes <- ifelse(predicted_probabilities > 0.5, 1, 0)

# Creating a confusion matrix
confusion_matrix <- table(Predicted = predicted_classes, Actual =
df_bcd$label)

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

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

b) Keep this assignment handy, as you will be comparing your results from this week to next week.