Assignment 10.2.1 Thoracic Surgery Binary Analysis

Kausik Chattapadhyay

2022-11-02

Introduction:

For this problem, you will be working with the thoracic surgery data set from the University of California Irvine machine learning repository. This data set 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.

Data Set Information:

The data was collected retrospectively at Wroclaw Thoracic Surgery Centre for patients who underwent major lung resections for primary lung cancer in the years 2007â€“2011. The Centre is associated with the Department of Thoracic Surgery of the Medical University of Wroclaw and Lower-Silesian Centre for Pulmonary Diseases, Poland, while the research database constitutes a part of the National Lung Cancer Registry, administered by the Institute of Tuberculosis and Pulmonary Diseases in Warsaw, Poland.

Attribute Information:

```
1. DGN: Diagnosis - specific combination of ICD-10 codes for primary and secondary
                    as well multiple tumours if any (DGN3,DGN2,DGN4,DGN6,DGN5,DGN8,
                    DGN1)
2. PRE4: Forced vital capacity - FVC (numeric)
3. PRE5: Volume that has been exhaled at the end of the first second of forced
         expiration - FEV1 (numeric)
4. PRE6: Performance status - Zubrod scale (PRZ2,PRZ1,PRZ0)
5. PRE7: Pain before surgery (T,F)
6. PRE8: Haemoptysis before surgery (T,F)
7. PRE9: Dyspnoea before surgery (T,F)
8. PRE10: Cough before surgery (T,F)
9. PRE11: Weakness before surgery (T,F)
10. PRE14: T in clinical TNM - size of the original tumour,
           from OC11 (smallest) to OC14 (largest) (OC11,OC14,OC12,OC13)
11. PRE17: Type 2 DM - diabetes mellitus (T,F)
12. PRE19: MI up to 6 months (T,F)
13. PRE25: PAD - peripheral arterial diseases (T,F)
14. PRE30: Smoking (T,F)
15. PRE32: Asthma (T,F)
16. AGE: Age at surgery (numeric)
17. Risk1Y: 1 year survival period - (T)rue value if died (T,F)
```

Class Distribution: the class value (Risk1Y) is binary valued.

Question A:

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.

Answer for A

```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/kausik/desktop/MS Data Science/DSC 520/dsc520-stats-r-assignments")
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.0.2
```

```
library('foreign')
```

```
## Warning: package 'foreign' was built under R version 4.0.5
```

```
set.seed(101)
thoracicSurgery_df <- read.arff("data/ThoracicSurgery.arff")
head(thoracicSurgery_df)
```

```
##    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
```

```
# Split the data into train(80%) and test(20%).
split <- sample.split(thoracicSurgery_df, SplitRatio = 0.80)
train <- subset(thoracicSurgery_df, split == TRUE)
test <- subset(thoracicSurgery_df, split == FALSE)

#logistic regression model with 80% train data
thoracicSurgery_glm <- glm(Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 +
PRE9 + PRE10 + PRE11 + PRE14 + PRE17 + PRE19 + PRE25 + PRE30 + PRE32 + AGE,
data=train, family = binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(thoracicSurgery_glm)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN + PRE4 + PRE5 + PRE6 + PRE7 + PRE8 +
##     PRE9 + PRE10 + PRE11 + PRE14 + PRE17 + PRE19 + PRE25 + PRE30 +
##     PRE32 + AGE, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6598  -0.5461  -0.4127  -0.1710   2.5202
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -15.78579 2399.54551  -0.007   0.9948
## DGNDGN2       15.18219 2399.54480   0.006   0.9950
## DGNDGN3       14.63613 2399.54475   0.006   0.9951
## DGNDGN4       15.09875 2399.54480   0.006   0.9950
## DGNDGN5       17.56260 2399.54492   0.007   0.9942
## DGNDGN6        1.38862 2873.26567   0.000   0.9996
## DGNDGN8       19.38182 2399.54532   0.008   0.9936
## PRE4          -0.11051    0.38748  -0.285   0.7755
## PRE5          -0.45294    0.43757  -1.035   0.3006
## PRE6PRZ1      -1.18516    0.69723  -1.700   0.0892 .
## PRE6PRZ2      -1.62666    1.06627  -1.526   0.1271
## PRE7T          1.03661    0.73934   1.402   0.1609
## PRE8T          0.06090    0.51094   0.119   0.9051
## PRE9T          1.09822    0.64218   1.710   0.0872 .
## PRE10T         1.02896    0.66716   1.542   0.1230
## PRE11T         0.28149    0.48760   0.577   0.5637
## PRE14OC12      0.79744    0.39595   2.014   0.0440 *
## PRE14OC13      0.46812    0.89907   0.521   0.6026
## PRE14OC14      1.73496    0.79553   2.181   0.0292 *
## PRE17T         0.89357    0.52387   1.706   0.0881 .
## PRE19T       -14.92052 1639.26029  -0.009   0.9927
## PRE25T        -0.23538    1.20967  -0.195   0.8457
## PRE30T         2.21438    0.87134   2.541   0.0110 *
## PRE32T       -14.51067 1532.04746  -0.009   0.9924
## AGE           -0.03393    0.02186  -1.553   0.1205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 290.42  on 360  degrees of freedom
## Residual deviance: 243.95  on 336  degrees of freedom
## AIC: 293.95
##
## Number of Fisher Scoring iterations: 15
```

Question B.

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

Answer for B:

The following variables had the greatest effect on the survival rate (based on P value):
1. PRE9T - Indicates whether the patient had Dyspnoea before surgery.
2. PRE14OC14 - The largest size of the original tumor.
3. PRE17T - This variable indicates whether the patient had Type 2 Diabetes.
4. PRE30T - Indicates that patient is a smoker.
5. PRE14OC13 - The second largest size of the tumor.
6. PRE5 - Volume that has been exhaled at the end of the first second of forced expiration

Question C:

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?

Answer For C

The accuracy of the model is 82% for set aside test data though train data has 87% accuracy, so we can conclude that our model is correct in predicting the outcome.

```
# Predict the train and test data with model

res_train <- predict(thoracicSurgery_glm, train, type="response")
res_test <- predict(thoracicSurgery_glm, test, type="response")

# validate the model- confusion matrix

## Train Data confusion Matrix
confusion_mat_train <- table(Actual_Value=train$Risk1Yr,
                             Predicted_Value=res_train >0.5)
confusion_mat_train
```

```
##             Predicted_Value
## Actual_Value FALSE TRUE
##            F   307    4
##            T    44    6
```

```
## Test Data Confusion Matrix
confusion_mat_test <- table(Actual_Value=test$Risk1Yr,
                            Predicted_Value=res_test >0.5)
confusion_mat_test
```

```
##             Predicted_Value
## Actual_Value FALSE TRUE
##            F    86    3
##            T    17    3
```

```r
## Train Accuracy
modelAccuracy_train <- (confusion_mat_train[[1,1]] + confusion_mat_train[[2,2]]) / sum(confusion_mat_tra
modelAccuracy_train
```

```
## [1] 0.867036
```

```r
## Test Accuracy
modelAccuracy_test <- (confusion_mat_test[[1,1]] + confusion_mat_test[[2,2]]) / sum(confusion_mat_test)
modelAccuracy_test
```

```
## [1] 0.8165138
```

Set aside Test accuracy for this model is 82% though Train accuracy is 87%. so we can conclude that our model is correct in predicting the outcome.