assignment_10.2_Venkidusamy_KesavAdithya

Kesav Adithya Venkidusamy

11/1/2021

```
knitr::opts_chunk$set(echo = TRUE)

library(farff)
library(ggplot2)
library(coefplot)
library(scales)

thoraric_df <- readARFF('E:/Personal/Bellevue University/Course/github/dsc520/data/ThoraricSurgery.arff
```

```
## Parse with reader=readr : E:/Personal/Bellevue University/Course/github/dsc520/data/ThoraricSurgery.a
```

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

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:scales':
##
##     col_factor
```

```
## header: 0.030000; preproc: 0.000000; data: 0.370000; postproc: 0.000000; total: 0.400000
```

```
head(thoraric_df,10)
```

```
##     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
## 7  DGN3 4.36 3.28 PRZ1    F    F    F     T     F  OC12     T     F     F     T
## 8  DGN2 3.19 2.50 PRZ1    F    F    F     T     F  OC11     F     F     T     T
## 9  DGN3 3.16 2.64 PRZ2    F    F    F     T     T  OC11     F     F     F     T
## 10 DGN3 2.32 2.16 PRZ1    F    F    F     T     F  OC11     F     F     F     T
##    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
## 7      F  59       T
## 8      F  66       T
## 9      F  68       F
## 10     F  54       F
```

```
dim(thoraric_df)
```

```
## [1] 470  17
```

```
colnames(thoraric_df)
```

```
##  [1] "DGN"     "PRE4"    "PRE5"    "PRE6"    "PRE7"    "PRE8"    "PRE9"
##  [8] "PRE10"   "PRE11"   "PRE14"   "PRE17"   "PRE19"   "PRE25"   "PRE30"
## [15] "PRE32"   "AGE"     "Risk1Yr"
```

```
#Creating test samples

#Total number of rows in dataset
n <- nrow(thoraric_df)

#80% of total number of records
n_test <- round(0.80 * n)

#Create a vector of indices which is an 80% random sample
set.seed(1)
train_indices <- sample(1:n, n_test)

#Subset the data frame to train indices only
train <- thoraric_df[train_indices,]

#Exclude the training indices for test set
test <- thoraric_df[-train_indices,]

#Check the dimensions
paste("Train sample ize: ", nrow(train))
```

```
## [1] "Train sample ize:  376"
```

```
paste("Train sample ize: ", nrow(test))
```

```
## [1] "Train sample ize:  94"
```

```
# 2a.Fit a binary logistic regression model to the data set that predicts whether or not the patient su

thoraric_mdl <- glm(Risk1Yr ~., family='binomial', data=thoraric_df)
summary(thoraric_mdl)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = "binomial", data = thoraric_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4929   0.2762   0.4199   0.5439   1.6084
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.604e+01  2.333e+03   0.011 0.991093
## DGNDGN2     -5.557e-01  4.128e-01  -1.346 0.178199
## DGNDGN4     -4.278e-01  4.733e-01  -0.904 0.366122
## DGNDGN6      1.377e+01  1.178e+03   0.012 0.990671
## DGNDGN5     -2.201e+00  6.113e-01  -3.600 0.000318 ***
## DGNDGN8     -3.852e+00  1.550e+00  -2.485 0.012959 *
## DGNDGN1      1.418e+01  2.400e+03   0.006 0.995285
## PRE4         2.272e-01  1.849e-01   1.229 0.219094
## PRE5         3.030e-02  1.786e-02   1.697 0.089715 .
## PRE6PRZ1     1.490e-01  5.783e-01   0.258 0.796647
## PRE6PRZ0    -2.937e-01  7.907e-01  -0.371 0.710303
## PRE7F        7.153e-01  5.556e-01   1.288 0.197884
## PRE8F        1.743e-01  3.892e-01   0.448 0.654188
## PRE9F        1.368e+00  4.868e-01   2.811 0.004942 **
## PRE10F       5.770e-01  4.826e-01   1.196 0.231855
## PRE11F       5.162e-01  3.965e-01   1.302 0.192948
## PRE14OC14   -1.653e+00  6.094e-01  -2.713 0.006675 **
## PRE14OC12   -4.394e-01  3.301e-01  -1.331 0.183177
## PRE14OC13   -1.179e+00  6.165e-01  -1.913 0.055799 .
## PRE17F       9.266e-01  4.445e-01   2.085 0.037092 *
## PRE19F      -1.466e+01  1.654e+03  -0.009 0.992928
## PRE25F      -9.789e-02  1.003e+00  -0.098 0.922273
## PRE30F       1.084e+00  4.990e-01   2.172 0.029840 *
## PRE32F      -1.398e+01  1.645e+03  -0.008 0.993219
## AGE          9.506e-03  1.810e-02   0.525 0.599442
## ---
## 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
```

```
#Creating a model with Sample data

train_mdl <- glm(Risk1Yr ~ .,data=train, family='binomial')
summary(train_mdl)
```

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

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6636   0.2547   0.3969   0.5020   1.3637
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.474e+01  3.393e+03   0.007 0.994183
## DGNDGN2     -7.190e-01  4.850e-01  -1.482 0.138214
## DGNDGN4     -4.038e-01  6.112e-01  -0.661 0.508865
## DGNDGN6      1.369e+01  1.180e+03   0.012 0.990747
## DGNDGN5     -2.194e+00  6.557e-01  -3.345 0.000822 ***
## DGNDGN8     -2.086e+01  2.400e+03  -0.009 0.993065
## DGNDGN1      1.365e+01  2.400e+03   0.006 0.995462
## PRE4         1.656e-01  2.192e-01   0.755 0.450007
## PRE5         2.784e-02  1.823e-02   1.528 0.126623
## PRE6PRZ1    -1.636e-02  7.024e-01  -0.023 0.981424
## PRE6PRZ0    -3.318e-01  9.493e-01  -0.350 0.726683
## PRE7F        7.414e-01  6.080e-01   1.219 0.222681
## PRE8F       -3.703e-01  4.998e-01  -0.741 0.458761
## PRE9F        1.790e+00  5.789e-01   3.092 0.001986 **
## PRE10F       9.275e-01  5.859e-01   1.583 0.113429
## PRE11F       5.227e-01  4.570e-01   1.144 0.252668
## PRE14OC14   -1.192e+00  6.927e-01  -1.720 0.085370 .
## PRE14OC12   -2.157e-01  3.821e-01  -0.564 0.572421
## PRE14OC13   -1.640e+00  6.579e-01  -2.494 0.012647 *
## PRE17F       1.054e+00  5.054e-01   2.084 0.037126 *
## PRE19F      -1.413e+01  2.400e+03  -0.006 0.995300
## PRE25F       1.583e-01  1.027e+00   0.154 0.877482
## PRE30F       1.094e+00  6.060e-01   1.805 0.071034 .
## PRE32F      -1.325e+01  2.400e+03  -0.006 0.995595
## AGE          9.307e-03  2.161e-02   0.431 0.666667
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 305.84  on 375  degrees of freedom
## Residual deviance: 253.98  on 351  degrees of freedom
## AIC: 303.98
##
## Number of Fisher Scoring iterations: 15
```

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

From thoraric_mdl summary, we could see DGNDGN5 variable is having smallest p-value. The variables having next two significant p-values are PRE9F and PRE14OC14. Other variables having significant p-values are PRE17F and PRE30F.The definition of these variables are given below.

1. DGNDGN5 is the diagnosis and related to multiple tumors
2. PRE9F is shortness of breath before surgery

3. PRE14OC14 is the size of original tumor

#2.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?

```
#Prediction for full data sets

pred <- predict(thoraric_mdl, type="response")
predicted <- round(pred)
conf_matrix_thoraric <- table(Predicted = predicted, Reference=thoraric_df$Risk1Yr)
accuracy_thoraric <- (conf_matrix_thoraric[1,1] + conf_matrix_thoraric[2,2]) / nrow(thoraric_df)

cat("The accuracy of the model without any sampling performed: ",percent(accuracy_thoraric))
```

```
## The accuracy of the model without any sampling performed:  84%
```

```
#Prediction for test data
test_pred <- predict(train_mdl, test, type="response")
test_predicted <- round(test_pred)
conf_matrix_test <- table(Predicted=test_predicted, Reference=test$Risk1Yr)
accuracy_test <- (conf_matrix_test[1,1] + conf_matrix_test[2,2]) /
  nrow(test)

cat("The accuracy of the model for sampling performed: ",percent(accuracy_test))
```

```
## The accuracy of the model for sampling performed:  79%
```

#Part 2

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

b.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.

```
binary_df <- read.csv("E:/Personal/Bellevue University/Course/github/dsc520/data/binary-classifier-data
dim(binary_df)
```

```
## [1] 1498    3
```

```
#Create Sample rows
binary_row <- nrow(binary_df)

#Creating 80% of the rows for training sample
binary_sample_row <- round(0.80 * binary_row)
```

```
#Create a vector of indices with 80% sample
set.seed(1)
binary_train_indices <- sample(1:binary_row, binary_sample_row)

#Subset of data frame to training indices
binary_train <-binary_df[binary_train_indices,]

#Subset excluding training indices
binary_test <- binary_df[-binary_train_indices,]

#Check the dimensions
paste("Training sample size: ",nrow(binary_train))
```

```
## [1] "Training sample size:  1198"
```

```
paste("Test sample size: ", nrow(binary_test))
```

```
## [1] "Test sample size:  300"
```

```
#Model for complete dataset
binary_mdl <- glm(label ~ x+y,data = binary_df, family=binomial(link="logit"))
summary(binary_mdl)
```

```
##
## Call:
## glm(formula = label ~ x + y, family = binomial(link = "logit"),
##     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
```

```
#Observation: y has significant p-value where as x does not have.
```

```r
#Modeling with Sample data

binary_train_mdl <- glm(label ~ ., data=binary_train, family="binomial")
summary(binary_train_mdl)
```

```
##
## Call:
## glm(formula = label ~ ., family = "binomial", data = binary_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3667  -1.1648  -0.9606   1.1661   1.3910
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.410894   0.131366   3.128 0.001761 **
## x           -0.002318   0.002057  -1.127 0.259704
## y           -0.007990   0.002089  -3.824 0.000131 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1660.0  on 1197  degrees of freedom
## Residual deviance: 1641.6  on 1195  degrees of freedom
## AIC: 1647.6
##
## Number of Fisher Scoring iterations: 4
```

```r
#2.ii.What is the accuracy of the logistic regression classifier?

#Prediction on unsample data

binary_pred <- predict(binary_mdl, type="response")
binary_predicted <- round(binary_pred)
binary_conf_matrix <- table(Predicted=binary_predicted,Reference=binary_df$label)

accuracy_binary <- (binary_conf_matrix[1,1] + binary_conf_matrix[2,2]) / nrow(binary_df)
cat("Accuracy of binary model for the whole dataset: ", percent(accuracy_binary))
```

```
## Accuracy of binary model for the whole dataset:  58%
```

```r
#Prediction on sample data

test_binary_pred <- predict(binary_train_mdl, binary_test, type="response")
test_binary_predicted <- round(test_binary_pred)
test_binary_conf_matrix <- table(Predicted=test_binary_predicted,Reference=binary_test$label)

test_accuracy_binary <- (test_binary_conf_matrix[1,1] + test_binary_conf_matrix[2,2]) / nrow(binary_tes
cat("Accuracy of binary model for the whole dataset: ", percent(test_accuracy_binary))
```

```
## Accuracy of binary model for the whole dataset:  55%
```