Week10_assignment_RiazAhmed_TamimAnsari.R

Riaz

2023-12-28

```r
# title: "Week_10_Excercise_Riaz"
# author: "Riaz Ahmed Tamim Ansari"
# date: "2023-11-05"


#Loading the data using foreign library

library(foreign)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Loading required package: lattice
```

```r
dframe <-
  read.arff("C:\\Users\\Riaz\\Desktop\\MSDS\\Introduction to Statistics\\Week10\\thoracicsurgerydata\\Th

#Quick summary using structure function
str(dframe)
```

```
## '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
#Checking if the dependent variable is factor,
is.factor(dframe$Risk1Yr)
```

```
## [1] TRUE
```

```r
head(dframe)
```

```
##    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
```

```r
#The variable explanation from the website says as for Risk1Yr, "1 year survival period - (T)rue value
#if died (T,F).  The question is to find out the survival rate, which means False.  For that reason,
#we will take True as baseline category.  Alphabetically F will be taken as baseline and we need to
#use relevel to change.

dframe$Risk1Yr <- relevel(dframe$Risk1Yr, ref = "T")

#Split the given data into Test and Train by using base R package.
set.seed(2)
sample <- sample(c(TRUE,FALSE),nrow(dframe),replace=TRUE,prob=c(0.6,0.4))
train <- dframe[sample,]
test <- dframe[!sample,]

table(dframe$DGN)
```

```
##
## DGN1 DGN2 DGN3 DGN4 DGN5 DGN6 DGN8
##    1   52  349   47   15    4    2
```

```r
table(train$DGN)
```

```
##
## DGN1 DGN2 DGN3 DGN4 DGN5 DGN6 DGN8
##    0   33  210   26    8    2    1
```

```r
table(test$DGN)
```

```
##
## DGN1 DGN2 DGN3 DGN4 DGN5 DGN6 DGN8
##    1   19  139   21    7    2    1
```

```r
#Initially, I have created model1 with all the variables and then used
#stepwise model selection with both the directions and arrived at model5,

model1 <- glm(Risk1Yr~.,data=train,family=binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ ., family = binomial, data = train)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.97738    2.06381   1.443  0.14912
## DGNDGN3         0.45483    0.57672   0.789  0.43032
## DGNDGN4        -0.12440    0.81288  -0.153  0.87837
## DGNDGN5        -1.72088    0.96540  -1.783  0.07466 .
## DGNDGN6        13.67455 1688.28839   0.008  0.99354
## DGNDGN8       -21.03566 2399.54497  -0.009  0.99301
## PRE4            0.06488    0.24393   0.266  0.79027
## PRE5            0.03344    0.02044   1.636  0.10184
## PRE6PRZ1       -0.21389    0.75131  -0.285  0.77588
## PRE6PRZ2        0.19072    1.10515   0.173  0.86299
## PRE7T          -0.57623    0.77866  -0.740  0.45929
## PRE8T          -1.04582    0.48058  -2.176  0.02954 *
## PRE9T          -1.59337    0.72758  -2.190  0.02853 *
## PRE10T         -0.16259    0.63358  -0.257  0.79747
## PRE11T         -0.66909    0.50256  -1.331  0.18307
## PRE14OC12      -0.65481    0.46360  -1.412  0.15782
## PRE14OC13      -2.65974    0.86772  -3.065  0.00218 **
## PRE14OC14      -1.94907    0.90019  -2.165  0.03037 *
## PRE17T         -1.07204    0.58545  -1.831  0.06708 .
## PRE19T         15.18925 2399.54478   0.006  0.99495
## PRE25T          1.81153    1.39521   1.298  0.19415
## PRE30T         -1.73061    0.77160  -2.243  0.02490 *
## PRE32T         14.13635 2399.54481   0.006  0.99530
## AGE             0.01905    0.02448   0.778  0.43643
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 240.16  on 279  degrees of freedom
## Residual deviance: 189.62  on 256  degrees of freedom
## AIC: 237.62
##
## Number of Fisher Scoring iterations: 15
```

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

```r
stepwise_model <- stepAIC(model1, direction = "both", trace = FALSE)
summary(stepwise_model)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN + PRE5 + PRE8 + PRE9 + PRE14 + PRE17 +
##     PRE25 + PRE30, family = binomial, data = train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.85977    0.97853   3.944    8e-05 ***
## DGNDGN3       0.48664    0.54482   0.893  0.37175
## DGNDGN4      -0.07424    0.78690  -0.094  0.92483
## DGNDGN5      -1.91000    0.96375  -1.982  0.04750 *
## DGNDGN6      13.36803 1029.10260   0.013  0.98964
## DGNDGN8     -19.54090 1455.39786  -0.013  0.98929
## PRE5          0.03596    0.01973   1.822  0.06843 .
## PRE8T        -1.16598    0.45014  -2.590  0.00959 **
## PRE9T        -1.62657    0.71741  -2.267  0.02337 *
## PRE14OC12    -0.52276    0.44029  -1.187  0.23510
## PRE14OC13    -2.68656    0.85541  -3.141  0.00169 **
## PRE14OC14    -1.80634    0.84948  -2.126  0.03347 *
## PRE17T       -1.13021    0.57049  -1.981  0.04758 *
## PRE25T        2.15920    1.48839   1.451  0.14687
## PRE30T       -1.76377    0.75985  -2.321  0.02028 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 240.16  on 279  degrees of freedom
## Residual deviance: 193.31  on 265  degrees of freedom
## AIC: 223.31
##
## Number of Fisher Scoring iterations: 14
```

```r
# Using the DGN + PRE5 + PRE8 + PRE9 + PRE14 + PRE17 + PRE25 + PRE30 variables as suggested by
#the above stepwise calculation,

model5 <- glm(Risk1Yr ~ DGN + PRE5 + PRE8 + PRE9 + PRE14 + PRE17 + PRE25 + PRE30,data=train,family=bino

summary(model5)
```

```
##
## Call:
## glm(formula = Risk1Yr ~ DGN + PRE5 + PRE8 + PRE9 + PRE14 + PRE17 +
##     PRE25 + PRE30, family = binomial, data = train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.85977    0.97853   3.944    8e-05 ***
## DGNDGN3       0.48664    0.54482   0.893  0.37175
```

```
## DGNDGN4       -0.07424    0.78690  -0.094  0.92483
## DGNDGN5       -1.91000    0.96375  -1.982  0.04750 *
## DGNDGN6       13.36803 1029.10260   0.013  0.98964
## DGNDGN8      -19.54090 1455.39786  -0.013  0.98929
## PRE5           0.03596    0.01973   1.822  0.06843 .
## PRE8T         -1.16598    0.45014  -2.590  0.00959 **
## PRE9T         -1.62657    0.71741  -2.267  0.02337 *
## PRE14OC12     -0.52276    0.44029  -1.187  0.23510
## PRE14OC13     -2.68656    0.85541  -3.141  0.00169 **
## PRE14OC14     -1.80634    0.84948  -2.126  0.03347 *
## PRE17T        -1.13021    0.57049  -1.981  0.04758 *
## PRE25T         2.15920    1.48839   1.451  0.14687
## PRE30T        -1.76377    0.75985  -2.321  0.02028 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 240.16  on 279  degrees of freedom
## Residual deviance: 193.31  on 265  degrees of freedom
## AIC: 223.31
##
## Number of Fisher Scoring iterations: 14
```

```
#Analysis:
#=========

#Deviance improvement and chisquare statistic:
chisquaremodel5 <- model5$null.deviance - model5$deviance
chisquaremodel5
```

```
## [1] 46.84578
```

```
#Calculating the degrees of freedom,
chisquaredfmodel5 <- model5$df.null - model5$df.residual
chisquaredfmodel5
```

```
## [1] 14
```

```
#Calculate Chisqprobability,

chisqprobmodel5 <- 1 - pchisq(chisquaremodel5,chisquaredfmodel5)
chisqprobmodel5
```

```
## [1] 2.037695e-05
```

```
#As the chisquare probability is less than .05, we can reject the Null hypothesis

#Coeffecients of the model
model5$coefficients
```

```
##  (Intercept)      DGNDGN3      DGNDGN4      DGNDGN5      DGNDGN6      DGNDGN8
##   3.85976969   0.48664037  -0.07424226  -1.91000026  13.36803113 -19.54090467
##         PRE5        PRE8T        PRE9T     PRE14OC12    PRE14OC13    PRE14OC14
##   0.03595836  -1.16597557  -1.62656877  -0.52276416  -2.68656105  -1.80633607
##       PRE17T       PRE25T       PRE30T
##  -1.13020931   2.15919551  -1.76376617
```

```
#Odds ratio for the predictors in the model,
exp(model5$coefficients)
```

```
##  (Intercept)      DGNDGN3      DGNDGN4      DGNDGN5      DGNDGN6      DGNDGN8
## 4.745442e+01 1.626841e+00 9.284467e-01 1.480803e-01 6.392374e+05 3.262067e-09
##         PRE5        PRE8T        PRE9T     PRE14OC12    PRE14OC13    PRE14OC14
## 1.036613e+00 3.116185e-01 1.966030e-01 5.928795e-01 6.811478e-02 1.642549e-01
##       PRE17T       PRE25T       PRE30T
## 3.229656e-01 8.664165e+00 1.713981e-01
```

```
#Looking at the odds ratio, anything above 1 signifies, as the independent value increases
#the dependent variable also increases. Anything below 0 says as the  independent value increases
#dependedent variable decreases. Looking at the output of model5, I can say DGN3,DGN6,PRE5,PRE25
#are above 1 and positive.  Rest are all inversely proportional.
#Also, they denote for every one unit increase in that variable corresponding values displayed are
#the increase in dependent variable.
#For eg for every 1 unit increase in PRE25T variable, the odds of Survival rate increases 8.66 times an
#1 unit of PRE5, the odds of survival rate increases by 1.037 times

#And the following variables are having the great effect on survival rate.
#DGN6
#DGN3
#PRE5
#PRE25T

#Lesson learnt to solve the error:

#Change to NA to prevent the following error from occuring.  In order to fix this below error,
#I have also tried to increase the sample size from 70/30 to 60/40.  But it has not solved
#the problem.  So I had to do this way by substituting NA for new levels.  I could have ignored and
#deleted this one row, as it was very minor when compared to the entire dataset, however for learning
#purpose I didnt do that and just used NA instead.

#"Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels) :
#  factor DGN has new levels DGN1"

data_test_new <- test                                      # Duplicate test data set
data_test_new$DGN[which(!(data_test_new$DGN %in% unique(train$DGN)))] <- NA  # Replace new levels by NA

#Predicting using the model5

data_test_new$predicted_probability <- predict(model5, data_test_new, type="response")

data_test_new$final_prediction <- ifelse(data_test_new$predicted_probability >= 0.5, 'F', 'T')

#Confusion matrix to calculate accuracy:
```

```r
confusion_1st <- confusionMatrix(data = factor(data_test_new$final_prediction, levels = c('T', 'F')),
                              reference = factor(data_test_new$Risk1Yr, levels = c('T', 'F')))
# Print the confusion matrix
print(confusion_1st)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   T   F
##          T   4  11
##          F  23 151
##
##                Accuracy : 0.8201
##                  95% CI : (0.7578, 0.8721)
##     No Information Rate : 0.8571
##     P-Value [Acc > NIR] : 0.93683
##
##                   Kappa : 0.0985
##
##  Mcnemar's Test P-Value : 0.05923
##
##             Sensitivity : 0.14815
##             Specificity : 0.93210
##          Pos Pred Value : 0.26667
##          Neg Pred Value : 0.86782
##              Prevalence : 0.14286
##          Detection Rate : 0.02116
##    Detection Prevalence : 0.07937
##       Balanced Accuracy : 0.54012
##
##        'Positive' Class : T
##
```

```r
#Accuracy is at 82%

#Question 2:
#===========
#Fit a logistic regression model to the binary-classifier-data.csv dataset

#Load the data
binclass <- read.delim("C:\\Users\\Riaz\\Desktop\\MSDS\\Introduction to Statistics\\Week10\\binary-class

#There are no NA values in the dataset and there are a total of 1498 rows,
sum(is.na(binclass))
```

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

```r
nrow(binclass)
```

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

```r
names(binclass)
```

```
## [1] "label" "x"     "y"
```

```r
#We need to convert the binclass$label to factor,
is.factor(binclass$label)
```

```
## [1] FALSE
```

```r
binclass$label <- as.factor(binclass$label)

#Splitting the data to train and test

set.seed(2)
sample <- sample(c(TRUE,FALSE),nrow(binclass),replace=TRUE,prob=c(0.7,0.3))
train_binclass <- binclass[sample,]
test_binclass <- binclass[!sample,]

head(train_binclass)
```

```
##    label        x        y
## 1      0 70.88469 83.17702
## 3      0 73.78333 92.20325
## 4      0 66.40747 81.10617
## 7      0 70.92514 89.73168
## 9      0 72.75624 92.37422
## 10     0 69.03660 91.74529
```

```r
head(test_binclass)
```

```
##    label        x        y
## 2      0 74.97176 87.92922
## 5      0 69.07399 84.53739
## 6      0 72.23616 86.38403
## 8      0 77.57454 98.63425
## 13     0 68.01709 83.81533
## 16     0 69.23680 89.98705
```

```r
model_binclass <- glm(label~.,data=train_binclass,family=binomial)

summary(model_binclass)
```

```
##
## Call:
## glm(formula = label ~ ., family = binomial, data = train_binclass)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.364286   0.142415   2.558   0.0105 *
## x           -0.002982   0.002186  -1.364   0.1726
```

```
## y            -0.006694   0.002278  -2.938   0.0033 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1421.1  on 1025  degrees of freedom
## Residual deviance: 1408.6  on 1023  degrees of freedom
## AIC: 1414.6
##
## Number of Fisher Scoring iterations: 4
```

```
#Analysis:
#=========

#As the Pr(>|z|) is less than .05 for y, it is statistically significant.
#For the predictor x Pr(>|z|) is more than .05, meaning it is not statisfically significant.
#Both the coeffecients are negative, which means there is a negative relation with y for label.

#Residual deviance is 12.47 less than Null deviance, which means that the model is predicting
#outcomes more accurately than the base model.

chisq_model_binclass <- model_binclass$null.deviance - model_binclass$deviance

#Calculating probability with this chisquare statistic,

df_model_binclass <- model_binclass$df.null - model_binclass$df.residual

chisq_prob_model_binclass <- (1 - pchisq(chisq_model_binclass,df_model_binclass))
chisq_prob_model_binclass
```

```
## [1] 0.001962515
```

```
#As the probability of chisquare statistic is .001 which is lesser than .05, we can say
#that it is statistically significant and the model we created works.

#Calculating R2 by Hosmer and Lemeshow's measure

R2.model_binclass <- chisq_model_binclass/model_binclass$null.deviance
R2.model_binclass
```

```
## [1] 0.008772978
```

```
#Calculating the Odds ratio for the predictors in the model,
oddsration_model_binclass <- exp(model_binclass$coefficients)
oddsration_model_binclass
```

```
## (Intercept)           x           y
##   1.4394852   0.9970226   0.9933279
```

```r
#As the odds ratio are less than 1, the independent and dependent variables are inversely proportional.
#In this case, as y decreases 1, there is a corresponding increase in the odds of x .99 times

#Beta values:
#============
#Constant is 0.36429, SE is 0.14
#y dependent variable -0.006, SE is .002
#Odds ratio:
#==========
#For y dependent variable is 0.99

#Predicting on the test dataset,

test_binclass$predicted_probability <- predict(model_binclass, test_binclass, type="response")

test_binclass$final_prediction <- ifelse(test_binclass$predicted_probability >= 0.5, 1, 0)

#Creating confusion matrix and predicting accuracy,


confusion <- confusionMatrix(data = factor(test_binclass$final_prediction, levels = c(0, 1)),
                             reference = factor(test_binclass$label, levels = c(0, 1)))
# Print the confusion matrixlibrary(caret)
print(confusion)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 135 116
##          1 101 120
##
##                Accuracy : 0.5403
##                  95% CI : (0.4941, 0.5859)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.04422
##
##                   Kappa : 0.0805
##
##  Mcnemar's Test P-Value : 0.34192
##
##             Sensitivity : 0.5720
##             Specificity : 0.5085
##          Pos Pred Value : 0.5378
##          Neg Pred Value : 0.5430
##              Prevalence : 0.5000
##          Detection Rate : 0.2860
##    Detection Prevalence : 0.5318
##       Balanced Accuracy : 0.5403
##
##        'Positive' Class : 0
##
```

```
#Accuracy is at 54%
```