logistic Regression

Samiksha Borade

2022-11-11

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
getwd()
```

```
## [1] "C:/Users/HP/Documents"
```

```
diabetes <- read.csv("C:/Users/HP/Downloads/archive (3)/diabetes.csv")

View(diabetes)

head(diabetes)
```

```
##   Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
## 1           6     148            72            35       0 33.6
## 2           1      85            66            29       0 26.6
## 3           8     183            64             0       0 23.3
## 4           1      89            66            23      94 28.1
## 5           0     137            40            35     168 43.1
## 6           5     116            74             0       0 25.6
##   DiabetesPedigreeFunction Age Outcome
## 1                    0.627  50       1
## 2                    0.351  31       0
## 3                    0.672  32       1
## 4                    0.167  21       0
## 5                    2.288  33       1
## 6                    0.201  30       0
```

```
names(diabetes)
```

```
## [1] "Pregnancies"              "Glucose"
## [3] "BloodPressure"            "SkinThickness"
## [5] "Insulin"                  "BMI"
## [7] "DiabetesPedigreeFunction" "Age"
## [9] "Outcome"
```

```
str(diabetes)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ Pregnancies              : int  6 1 8 1 0 5 3 10 2 8 ...
##  $ Glucose                  : int  148 85 183 89 137 116 78 115 197 125 ...
```

```
##  $ BloodPressure           : int  72 66 64 66 40 74 50 0 70 96 ...
##  $ SkinThickness           : int  35 29 0 23 35 0 32 0 45 0 ...
##  $ Insulin                 : int  0 0 0 94 168 0 88 0 543 0 ...
##  $ BMI                     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ Age                     : int  50 31 32 21 33 30 26 29 53 54 ...
##  $ Outcome                 : int  1 0 1 0 1 0 1 0 1 1 ...
```

```
diabetes$Outcome <- as.factor(diabetes$Outcome)

class(diabetes$Outcome)
```

```
## [1] "factor"
```

```
table(diabetes$Outcome)
```

```
##
##   0   1
## 500 268
```

```
## Missing Values

colSums(is.na(diabetes))
```

```
##              Pregnancies                  Glucose            BloodPressure
##                        0                        0                        0
##            SkinThickness                  Insulin                      BMI
##                        0                        0                        0
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                        0
```

```
dim(diabetes)
```

```
## [1] 768   9
```

```
# Split the data
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.2.1
```

```
require(caTools)
set.seed(3)
sample = sample.split(diabetes$Outcome, SplitRatio=0.75)
train = subset(diabetes, sample==TRUE)
test = subset(diabetes, sample==FALSE)

nrow(diabetes)
```

```
## [1] 768
```

```
nrow(train)
```

```
## [1] 576
```

```
nrow(test)
```

```
## [1] 192
```

```
table(train$Age_Cat)
```

```
## < table of extent 0 >
```

```
str(train)
```

```
## 'data.frame':    576 obs. of  9 variables:
##  $ Pregnancies             : int  6 1 1 0 5 3 10 2 8 10 ...
##  $ Glucose                 : int  148 85 89 137 116 78 115 197 125 168 ...
##  $ BloodPressure           : int  72 66 66 40 74 50 0 70 96 74 ...
##  $ SkinThickness           : int  35 29 23 35 0 32 0 45 0 0 ...
##  $ Insulin                 : int  0 0 94 168 0 88 0 543 0 0 ...
##  $ BMI                     : num  33.6 26.6 28.1 43.1 25.6 31 35.3 30.5 0 38 ...
##  $ DiabetesPedigreeFunction: num  0.627 0.351 0.167 2.288 0.201 ...
##  $ Age                     : int  50 31 21 33 30 26 29 53 54 34 ...
##  $ Outcome                 : Factor w/ 2 levels "0","1": 2 1 1 2 1 2 1 2 2 2 ...
```

```
table(diabetes$Outcome)
```

```
##
##   0   1
## 500 268
```

```
baseline <- round(500/nrow(diabetes),2)
baseline
```

```
## [1] 0.65
```

```
AllVar <- glm(Outcome ~ ., data = train, family = binomial)
summary(AllVar)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4130  -0.7573  -0.4507   0.7763   2.8596
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -7.757693   0.793274  -9.779  < 2e-16 ***
## Pregnancies                0.113159   0.035379   3.199  0.00138 **
## Glucose                    0.033104   0.004113   8.049 8.34e-16 ***
## BloodPressure             -0.012949   0.005963  -2.172  0.02989 *
## SkinThickness              0.001993   0.007638   0.261  0.79411
## Insulin                   -0.001729   0.001005  -1.721  0.08530 .
## BMI                        0.078239   0.016830   4.649 3.34e-06 ***
## DiabetesPedigreeFunction   1.049963   0.338686   3.100  0.00193 **
## Age                        0.014089   0.010329   1.364  0.17255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 745.11  on 575  degrees of freedom
## Residual deviance: 562.31  on 567  degrees of freedom
## AIC: 580.31
##
## Number of Fisher Scoring iterations: 5
```

```r
PredictTrain <- predict(AllVar, type = "response")
summary(PredictTrain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00291 0.13356 0.27896 0.34896 0.52451 0.99210
```

```r
tapply(PredictTrain, train$Outcome, mean)
```

```
##         0         1
## 0.2455600 0.5418656
```

```r
threshold_0.5 <- table(train$Outcome, PredictTrain > 0.5)
threshold_0.5
```

```
##
##     FALSE TRUE
##   0   333   42
##   1    93  108
```

```r
accuracy_0.5 <- round(sum(diag(threshold_0.5))/sum(threshold_0.5),2)
sprintf("Accuracy is %s",accuracy_0.5)
```

```
## [1] "Accuracy is 0.77"
```

```r
MC_0.5 <- 1-accuracy_0.5
sprintf("Mis-classification error is %s",MC_0.5)
```

```
## [1] "Mis-classification error is 0.23"
```

```r
sensitivity0.5 <- round(118/(83+118),2)
specificity0.5 <- round(333/(333+42),2)
sprintf("Sensitivity at 0.5 threshold: %s", sensitivity0.5)
```

```
## [1] "Sensitivity at 0.5 threshold: 0.59"
```

```r
sprintf("Specificity at 0.5 threshold: %s", specificity0.5)
```

```
## [1] "Specificity at 0.5 threshold: 0.89"
```

```r
PredictTest <- predict(AllVar, type = "response", newdata = test)

test_tab <- table(test$Outcome, PredictTest > 0.5)
test_tab
```

```
##
##     FALSE TRUE
##   0   115   10
##   1    24   43
```

```r
accuracy_test <- round(sum(diag(test_tab))/sum(test_tab),2)
sprintf("Accuracy on test set is %s", accuracy_test)
```

```
## [1] "Accuracy on test set is 0.82"
```