Prima Indians Diabetes Outcome Logistics regression

Arinze Francis

2022-06-21

```
## function (..., list = character(), pos = -1, envir = as.environment(pos),
##     inherits = FALSE)
## {
##     dots <- match.call(expand.dots = FALSE)$...
##     if (length(dots) && !all(vapply(dots, function(x) is.symbol(x) ||
##         is.character(x), NA, USE.NAMES = FALSE)))
##         stop("... must contain names or character strings")
##     names <- vapply(dots, as.character, "")
##     if (length(names) == 0L)
##         names <- character()
##     list <- .Primitive("c")(list, names)
##     .Internal(remove(list, envir, inherits))
## }
## <bytecode: 0x0000000014d0bda0>
## <environment: namespace:base>
```

Load R packages

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

Dataset loading and manipulations

```
data <- read.table("pima-indians-diabetes.csv", header = T, sep ="," )
head(data)
```

```
##   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
```

```
any(is.na(data))
```

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

```
# Converting the dependent variable to factor
data$Outcome <- as.factor(data$Outcome)
str(data)
```

```
## '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                 : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 2 1 2 2 ...
```

Fit Logistics Regression model

```
data_result <- lrm(Outcome~ Age +DiabetesPedigreeFunction+BMI+ Insulin +SkinThickness +BloodPres
sure +Glucose+Pregnancies ,x=T,y=T,data = data )
data_result
```

```
## Logistic Regression Model
##
##  lrm(formula = Outcome ~ Age + DiabetesPedigreeFunction + BMI +
##      Insulin + SkinThickness + BloodPressure + Glucose + Pregnancies,
##      data = data, x = T, y = T)
##
##                        Model Likelihood    Discrimination    Rank Discrim.
##                              Ratio Test           Indexes          Indexes
##  Obs           768    LR chi2     270.04    R2       0.408    C       0.839
##   0            500    d.f.             8    g        1.811    Dxy     0.679
##   1            268    Pr(> chi2) <0.0001    gr       6.116    gamma   0.679
##  max |deriv| 6e-08                          gp       0.302    tau-a   0.309
##                                             Brier    0.153
##
##                           Coef    S.E.   Wald Z Pr(>|Z|)
##  Intercept                -8.4047 0.7166 -11.73 <0.0001
##  Age                       0.0149 0.0093   1.59 0.1112
##  DiabetesPedigreeFunction  0.9452 0.2991   3.16 0.0016
##  BMI                       0.0897 0.0151   5.95 <0.0001
##  Insulin                  -0.0012 0.0009  -1.32 0.1861
##  SkinThickness             0.0006 0.0069   0.09 0.9285
##  BloodPressure            -0.0133 0.0052  -2.54 0.0111
##  Glucose                   0.0352 0.0037   9.48 <0.0001
##  Pregnancies               0.1232 0.0321   3.84 0.0001
##
```

```
# For the coefficient table; The coefficient table showed that glucose, pregnancies, body mass i
ndex,  Blood pressure and Diabetes pedigree function variable has significant positive influence
(p-values < 0.05) on diabetes.

# Each one-unit change in glucose will increase the log odds of having diabetes by 0.035, and it
s p-value indicates that it is significant in determining diabetes. Also, each unit increase in
 BMI increases the log odds of having diabetes by 0.0897 and p-value is significant too and etc.

# Interpretation: On the top right, you see several discrimination indices. The C denotes the c-
index (AUC), In this case, the c-index is 0.839 (>0.8) meaning it is good enough for predicting
 the outcomes of individuals.
```

to obtain odds ratio and inter-quartile range

```
# Estimate odd ratios
data_result$coefficients %>% exp()
```

```
##                Intercept                      Age DiabetesPedigreeFunction
##                0.0002238                1.0149801                2.5732759
##                      BMI                  Insulin            SkinThickness
##                1.0938471                0.9988090                1.0006192
##            BloodPressure                  Glucose              Pregnancies
##                0.9867924                1.0357893                1.1310906
```

```
# Interpretation: The odds of being diabetes positive increases by 1.03 with an increase in Gluc
ose, The odds of being diabetes positive increases by 1.09 with an increase in Body mass Index a
nd etc

# Estimate inter-quartile range
di <- datadist(data) #
options(datadist='di')
data_result %>% summary()
```

```
##              Effects              Response : Outcome
##
##  Factor                   Low     High     Diff.    Effect   S.E.   Lower 0.95
##  Age                     24.0000  41.0000  17.0000  0.25277 0.1587 -0.05826
##   Odds Ratio             24.0000  41.0000  17.0000  1.28760     NA  0.94341
##  DiabetesPedigreeFunction  0.2437   0.6262   0.3825  0.36153 0.1144  0.13726
##   Odds Ratio             0.2437   0.6262   0.3825  1.43550     NA  1.14710
##  BMI                     27.3000  36.6000   9.3000  0.83422 0.1403  0.55921
##   Odds Ratio             27.3000  36.6000   9.3000  2.30300     NA  1.74930
##  Insulin                  0.0000 127.2500 127.2500 -0.15164 0.1147 -0.37641
##   Odds Ratio              0.0000 127.2500 127.2500  0.85929     NA  0.68632
##  SkinThickness            0.0000  32.0000  32.0000  0.01981 0.2208 -0.41291
##   Odds Ratio              0.0000  32.0000  32.0000  1.02000     NA  0.66172
##  BloodPressure           62.0000  80.0000  18.0000 -0.23932 0.0942 -0.42396
##   Odds Ratio             62.0000  80.0000  18.0000  0.78716     NA  0.65445
##  Glucose                 99.0000 140.2500  41.2500  1.45050 0.1530  1.15070
##   Odds Ratio             99.0000 140.2500  41.2500  4.26530     NA  3.16030
##  Pregnancies              1.0000   6.0000   5.0000  0.61591 0.1604  0.30156
##   Odds Ratio              1.0000   6.0000   5.0000  1.85130     NA  1.35200
##  Upper 0.95
##   0.56380
##   1.75730
##   0.58580
##   1.79640
##   1.10920
##   3.03200
##   0.07313
##   1.07590
##   0.45253
##   1.57230
##  -0.05468
##   0.94679
##   1.75030
##   5.75660
##   0.93027
##   2.53520
```

Diagnostics

Outliers

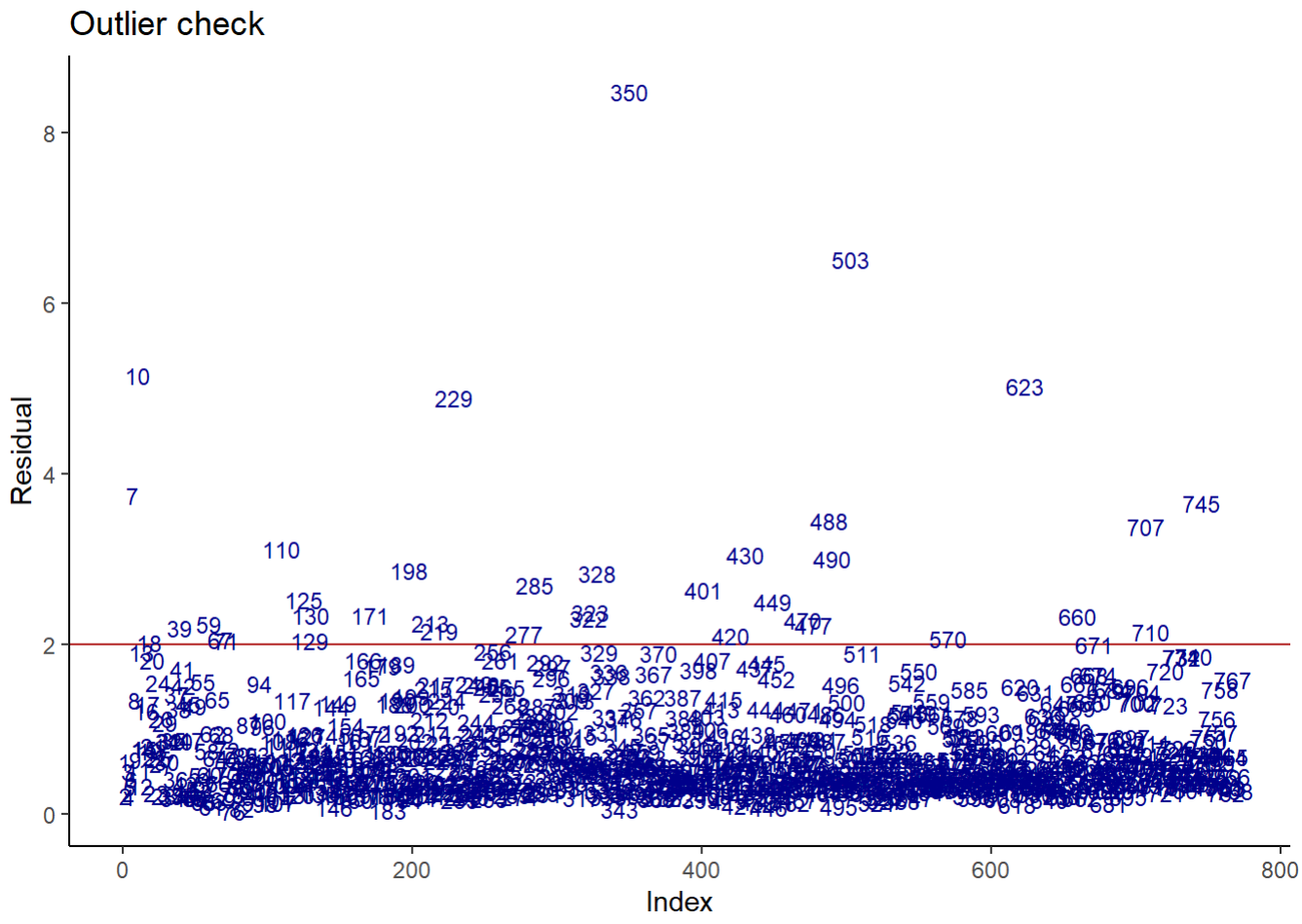
```
data_1 <- data %>% mutate(Residuals = residuals(data_result, type = 'pearson'),
                      Index=1:nrow(data))
head(data_1)
```

```
##   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 Residuals Index
## 1                    0.627  50       1    0.6209     1
## 2                    0.351  31       0   -0.2261     2
## 3                    0.672  32       1    0.5051     3
## 4                    0.167  21       0   -0.2084     4
## 5                    2.288  33       1    0.3293     5
## 6                    0.201  30       0   -0.4145     6
```

```
# Visualization of the Outliers
data_2 <- data_1 %>% ggplot(aes(x=Index, y=abs(Residuals))) + geom_hline(yintercept = 2,
                                                                         col='firebrick') + geom_text
(aes(label = Index),

col = 'darkblue', size = 3) + labs(title = 'Outlier check',

y = "Residual", x = "Index") + theme_classic()
data_2
```



Multicollinearity

```
data_result %>% vif()
```

```
##                     Age DiabetesPedigreeFunction                     BMI
##                   1.502                   1.034                 1.220
##                 Insulin           SkinThickness           BloodPressure
##                   1.468                   1.522                 1.175
##                 Glucose             Pregnancies
##                   1.214                   1.408
```

 # Interpreation: There is no case of multicollinearity as the values were below 5

Validate Model Using Bootstrap

```
model_validity <- validate(data_result, method="boot", B=1000)
model_validity
```

```
##           index.orig training    test optimism index.corrected    n
## Dxy           0.6789   0.6854  0.6703   0.0151          0.6637 1000
## R2            0.4085   0.4191  0.3983   0.0207          0.3878 1000
## Intercept     0.0000   0.0000 -0.0187   0.0187         -0.0187 1000
## Slope         1.0000   1.0000  0.9540   0.0460          0.9540 1000
## Emax          0.0000   0.0000  0.0135   0.0135          0.0135 1000
## D             0.3503   0.3616  0.3399   0.0217          0.3286 1000
## U            -0.0026  -0.0026  0.0009  -0.0035          0.0009 1000
## Q             0.3529   0.3642  0.3390   0.0252          0.3277 1000
## B             0.1527   0.1504  0.1551  -0.0047          0.1574 1000
## g             1.8108   1.8613  1.7693   0.0920          1.7188 1000
## gp            0.3025   0.3055  0.2984   0.0072          0.2953 1000
```

 # Interpreation: Using the Dxy, the bias-corrected Dxy is a bit smaller (0.6632) than the origin al (0.6789). The bias-corrected c-index (AUC) is c=1+Dxy2 which equals 0.8316.