R Practice 3 ALY6015

R practice week 3 - Module 3 Assignment — GLM and Logistic Regression

by Mohammad Hossein Movahedi

movahed.m@northeastern.edu

8 May 2022

Table Of content

Introduction

This R practice mainly focuses on GLM and Logistic Regression, and during this assignment, I will use the `Collage` dataset to solve various problems.

Analysis

This part will go through the steps described in the assignment and solve problems

Step 1: Import the dataset and perform Exploratory Data Analysis

In this part, I create a few figures and create a summary of the data. I will also try to clean the dataset

```
# Importing the dataset
data(College)

# Exploratory Data Analysis
str(College)
summary(College)
data <- as.data.frame(College)
#replacing blank data with null
data <- data %>% mutate_all(na_if,"")
#box plot
dev.off()
```

```
boxplot(data)
#creating some plots
p1 <- ggplot(data, aes(Accept,Enroll)) + geom_point() + theme_bw()
ggMarginal(p1)

p2 <- ggplot(data, aes(F.Undergrad, P.Undergrad, colour = Private)) +
  geom_point()
ggMarginal(p2, groupColour = TRUE, groupFill = TRUE)

# Hist for Apps
hist(data$Apps, xlim = c(0, 50000),col = "green", xlab="car(year)")
# Box plot
 y<- qplot(x=data$Private,y=data$Apps, fill=data$Private,geom='boxplot')+guides(scale= "none")
 z<-qplot(x=data$Private,y=data$Enroll, fill=data$Private,geom='boxplot')+guides(scale= "none")
  grid.arrange(y,z,nrow=1)
```

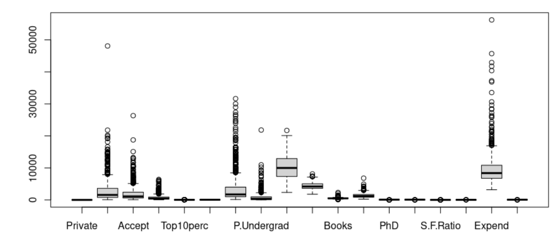
The results of `str()` and `summary()` are shown below

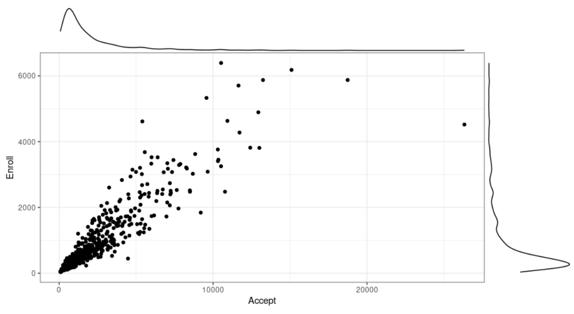
```
> str(College)
'data.frame': 777 obs. of  18 variables:
 $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ Apps       : num  1660 2186 1428 417 193 ...
 $ Accept     : num  1232 1924 1097 349 146 ...
 $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
 $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
 $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
 $ F.Undergrad: num  2885 2683 1036 510 249 ...
 $ P.Undergrad: num  537 1227 99 63 869 ...
 $ Outstate   : num  7440 12280 11250 12960 7560 ...
 $ Room.Board : num  3300 6450 3750 5450 4120 ...
 $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
 $ Personal   : num  2200 1500 1165 875 1500 ...
 $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
 $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
 $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
 $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
 $ Expend     : num  7041 10527 8735 19016 10922 ...
 $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

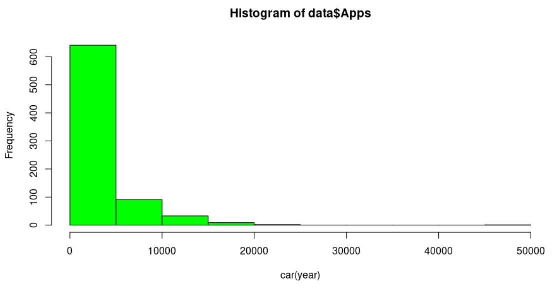
```
> summary(College)
 Private        Apps           Accept          Enroll       Top10perc
 No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
 Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
           Median : 1558   Median : 1110   Median : 434   Median :23.00
           Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
           3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
           Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
   Top25perc      F.Undergrad     P.Undergrad         Outstate       Room.Board
 Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340   Min.   :1780
 1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597
 Median : 54.0   Median : 1707   Median :  353.0   Median : 9990   Median :4200
 Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441   Mean   :4358
 3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050
 Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700   Max.   :8124
     Books           Personal          PhD            Terminal       S.F.Ratio
 Min.   :  96.0   Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50
 1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50
 Median : 500.0   Median :1200   Median : 75.00   Median : 82.0   Median :13.60
 Mean   : 549.4   Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09
```

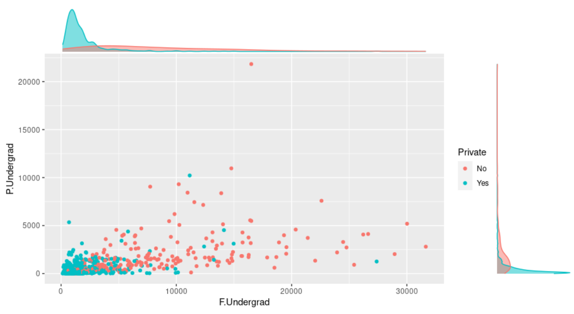
```
 3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50
 Max.   :2340.0   Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80
  perc.alumni        Expend         Grad.Rate
 Min.   : 0.00   Min.   : 3186   Min.   : 10.00
 1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00
 Median :21.00   Median : 8377   Median : 65.00
 Mean   :22.74   Mean   : 9660   Mean   : 65.46
 3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00
 Max.   :64.00   Max.   :56233   Max.   :118.00
```

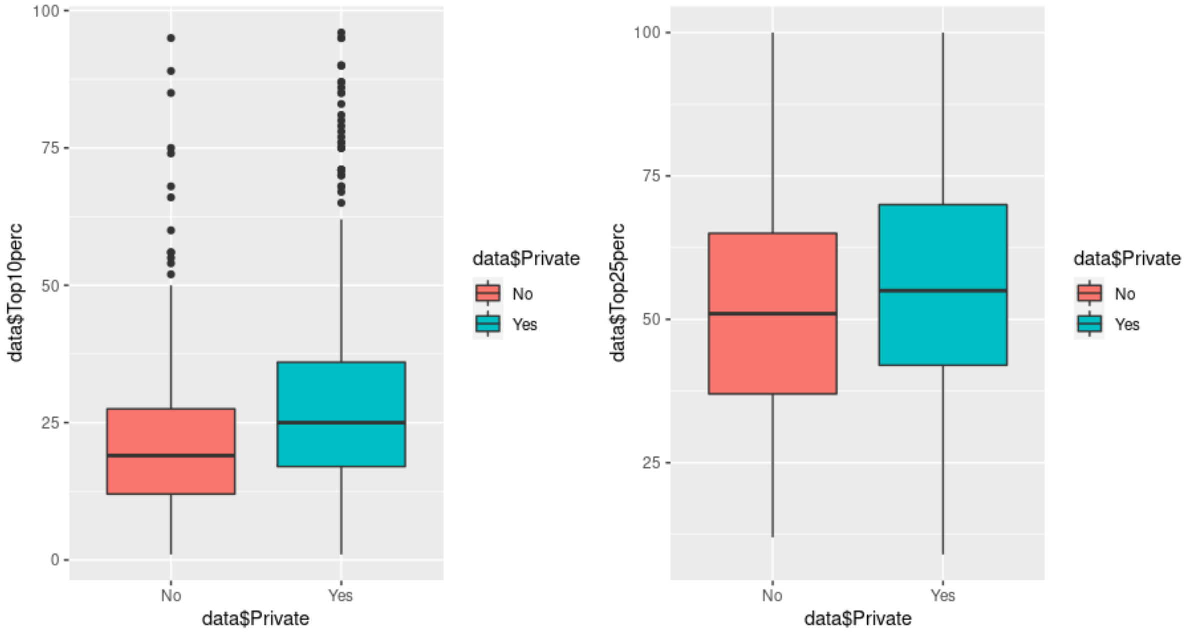
The plots that I created are shown below











Step 2: Split the data into a train and test

In this part, I divide the dataset to two-part

```
# split data to train and test
set.seed(910198135)
trainIndex<- createDataPartition(College$Private,p=0.70,list=FALSE)
train<-College[trainIndex,]
test<-College[-trainIndex,]
```

Step 3: Use the Glm() function

in this part, I use `glm()` like it is shown in the video

```
#Fit model on train data
model1<-glm(Private~.,data=train,family=binomial(link="logit"))
summary(model1)
```

the results are shown below

```
> summary(model1)

Call:
glm(formula = Private ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6775  -0.0356   0.0503   0.1469   3.4863

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  3.328e-01  2.312e+00   0.144   0.8856    
Apps        -5.086e-04  2.630e-04  -1.934   0.0531 .  
Accept      -3.192e-04  5.401e-04  -0.591   0.5544    
Enroll       2.484e-03  1.348e-03   1.843   0.0654 .  
Top10perc   -1.705e-02  3.587e-02  -0.475   0.6345    
Top25perc    3.341e-02  2.411e-02   1.386   0.1657    
F.Undergrad -4.286e-04  1.788e-04  -2.398   0.0165 *  
P.Undergrad -2.025e-05  1.544e-04  -0.131   0.8957    
Outstate     7.469e-04  1.393e-04   5.363 8.18e-08 ***
Room.Board   5.147e-04  3.219e-04   1.599   0.1098    
Books        2.607e-03  1.542e-03   1.691   0.0908 .  
Personal    -3.634e-04  3.538e-04  -1.027   0.3043    
PhD         -4.847e-02  3.628e-02  -1.336   0.1816    
Terminal    -6.036e-02  3.532e-02  -1.709   0.0875 .  
S.F.Ratio   -1.053e-01  7.669e-02  -1.373   0.1697    
perc.alumni  3.716e-02  2.442e-02   1.521   0.1282    
Expend       1.163e-04  1.471e-04   0.791   0.4291    
Grad.Rate    2.131e-03  1.525e-02   0.140   0.8888    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 639.40  on 544  degrees of freedom
Residual deviance: 166.17  on 527  degrees of freedom
AIC: 202.17

Number of Fisher Scoring iterations: 8
```

As it can be seen the `Outstate` and `F.Undergrad` showed significant impact therefore I continue with them

```
modelpfo<-glm(Private~F.Undergrad+Outstate,data=train, family=binomial(link="logit"))
summary(modelpfo)
```

the results are shown below

```
> summary(modelpfo)

Call:
glm(formula = Private ~ F.Undergrad + Outstate, family = binomial(link = "logit"), 
    data = train)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.8336  -0.0186   0.0997   0.2762   5.9521  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -3.264e+00  6.009e-01  -5.432 5.56e-08 ***
F.Undergrad -5.918e-04  6.976e-05  -8.482  < 2e-16 ***
Outstate     7.484e-04  8.409e-05   8.899  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 639.40  on 544  degrees of freedom
```

```
Residual deviance: 215.99  on 542  degrees of freedom
AIC: 221.99

Number of Fisher Scoring iterations: 7
```

As it can be seen the AICs are not so different

Step 4: Create a confusion matrix

```
#creating confusion matrix
#use model to predict probability of default
predicted <- predict(modelpfo, train, type="response")
#convert Private from "Yes" and "No" to 1's and 0's
test$Private <- ifelse(train$Private=="Yes", 1, 0)
#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(train$Private, predicted)[1]
#create confusion matrix
confusionMatrix(train$Private, predicted)
```

the result is shown below

```
> confusionMatrix(train$Private, predicted)
    0   1
0 131  12
1  18 384
```

As it can be seen thanks to `optimalCutoff()` function only a few data are mislabeled.

Step 5: Report and interpret metrics for Accuracy, Precision, Recall, and Specificity.

now I calculate the misclassification rate ,sensitivity and specificity

```
> #calculate sensitivity
> sensitivity(train$Private, predicted)
[1] 0.969697
> #calculate specificity
> specificity(train$Private, predicted)
[1] 0.8791946
> #calculate total misclassification error rate
> misClassError(train$Private, predicted, threshold=optimal)
[1] 0.055
```

As it can be seen there is only a 5% misclassification rate in this model also the specificity of the model is lower than the sensitivity but they are balanced in total

```
> #Accuracy=(IN+TP)/(TN+FP+FN+TP)
> (131+384)/(545)
```

```
[1] 0.9449541
> #Precision=TP/(FP+TP)
> 131/(131+12)
[1] 0.9160839
> #Recall=TP/(TP+FN)
> 131/(131+18)
[1] 0.8791946
```

As it can be seen all numbers are high showing the model is doing very well with train data

Step 6: Create a confusion matrix and report the results of the test set

now I repeat all the above again but with the test set the confusion matrix is shown below

```
> confusionMatrix(test$Private, predicted)
   0   1
0 53   8
1 10 161
```

As it can be seen the model works well with test data too

```
> #calculate sensitivity
> sensitivity(test$Private, predicted)
[1] 0.9526627
> #calculate specificity
> specificity(test$Private, predicted)
[1] 0.8412698
> #calculate total misclassification error rate
> misClassError(test$Private, predicted, threshold=optimal)
[1] 0.069
```

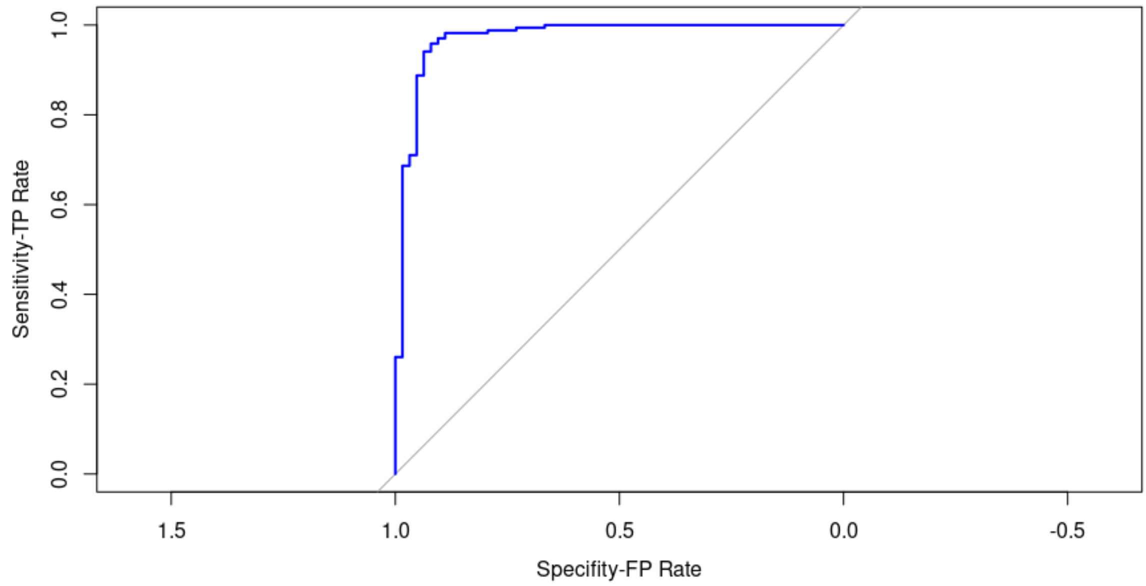
It can be seen the model works perfectly in terms of sensitivity and specificity.

Step 7: Plot and interpret the ROC curve

```
#Plot the receiver operator characteristic curve
ROC1<-roc(test$Private,probabilities.test)

plot(ROC1,col="blue",ylab="Sensitivity-TP Rate", xlab='Specifity-FP Rate')
```

the resulting plot is shown below



As it can be seen the model is far from the mid line showing how good it is comparing to other models

Step 8: Calculate and interpret the AUC

```
> auc<-auc(ROC1)
> auc
Area under the curve: 0.972
```

As it can be seen the area under the curve is almost one making it a very good model.

Conclusion

In this assignment, I learned about the confusion matrix and its uses and also how to make an effective Glm model, and I used it on a dataset.

References

Rickert, J. (2016). *Computing Classification Evaluation Metrics in R*. [online] Revolutions. Available at: https://blog.revolutionanalytics.com/2016/03/com_class_eval_metrics_r.html [Accessed 9 May 2022].

Zach (2021). *How to Create a Confusion Matrix in R (Step-by-Step)*. [online] Statology. Available at: https://www.statology.org/confusion-matrix-in-r/ [Accessed 9 May 2022].

Appendix

```
print('Mohammad Hossein Movahedi')
print('R practice 3')

#installing packages and loading them

install.packages("magrittr")
install.packages("dplyr")
install.packages("plyr")
install.packages("tidyverse")
install.packages("ggvis")
install.packages("ggplot2")
install.packages("gmodels")
install.packages("psych")
install.packages('caret')
install.packages('ggcorrplot')
install.packages('InformationValue')

library(ggcorrplot)
library(data.table)
library(FSA)
library(magrittr)
library(dplyr)
library(plyr)
library(tidyverse)
library(gmodels)
library(ggvis)
library(ggplot2)
library(psych)
library(corrplot)
library(pROC)
library(ISLR)
library(caret)
library( ggplot2)
library(gridExtra)
library(InformationValue)


# Importing the dataset
data(College)


# Exploratory Data Analysis
str(College)
summary(College)
data <- as.data.frame(College)
#replacing blank data with null
data <- data %>% mutate_all(na_if,"")
#box plot
dev.off()
boxplot(data)
#creating some plots
p1 <- ggplot(data, aes(Accept,Enroll)) + geom_point() + theme_bw()
ggMarginal(p1)

p2 <- ggplot(data, aes(F.Undergrad, P.Undergrad, colour = Private)) +
  geom_point()
ggMarginal(p2, groupColour = TRUE, groupFill = TRUE)

# Hist for Apps
hist(data$Apps, xlim = c(0, 50000),col = "green", xlab="car(year)")
```

```
# Box plot
 y<- qplot(x=data$Private,y=data$Top10perc, fill=data$Private,geom='boxplot')+guides(scale= "none")
 z<-qplot(x=data$Private,y=data$Top25perc, fill=data$Private,geom='boxplot')+guides(scale= "none")
  grid.arrange(y,z,nrow=1)

# split data to train and test
set.seed(910198135)
trainIndex<- createDataPartition(College$Private,p=0.70,list=FALSE)
train<-College[trainIndex,]
test<-College[-trainIndex,]




#Fit model on train data
model1<-glm(Private~.,data=train,family=binomial(link="logit"))
summary(model1)
modelpfo<-glm(Private~F.Undergrad+Outstate,data=train, family=binomial(link="logit"))
summary(modelpfo)
#creating confusion matrix
#use model to predict probability of train
predicted <- predict(modelpfo, train, type="response")
#convert Private from "Yes" and "No" to 1's and 0's
train$Private <- ifelse(train$Private=="Yes", 1, 0)
#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(train$Private, predicted)[1]
#create confusion matrix
confusionMatrix(train$Private, predicted)

#calculate sensitivity
sensitivity(test$Private, predicted)
#calculate specificity
specificity(test$Private, predicted)
#calculate total misclassification error rate
misClassError(test$Private, predicted, threshold=optimal)

#Accuracy=(IN+TP)/(TN+FP+FN+TP)
(131+384)/(545)
#Precision=TP/(FP+TP)
131/(131+12)
#Recall=TP/(TP+FN)
131/(131+18)

#testing
#use model to predict probability of test
predicted <- predict(modelpfo, test, type="response")
#convert Private from "Yes" and "No" to 1's and 0's
test$Private <- ifelse(test$Private=="Yes", 1, 0)
#find optimal cutoff probability to use to maximize accuracy
optimal <- optimalCutoff(test$Private, predicted)[1]
#create confusion matrix
confusionMatrix(test$Private, predicted)

#calculate sensitivity
sensitivity(test$Private, predicted)
#calculate specificity
specificity(test$Private, predicted)
#calculate total misclassification error rate
misClassError(test$Private, predicted, threshold=optimal)

## Test set predictions
probabilities.test<-predict(model1,newdata=test,type='response')
predicted.classes.min<-as.factor(ifelse(probabilities.test>=optimal, "Yes","No"))
```

```
#Plot the receiver operator characteristic curve
ROC1<-roc(test$Private,probabilities.test)

plot(ROC1,col="blue",ylab="Sensitivity-TP Rate", xlab='Specifity-FP Rate')

#Calculate the area under the ROC curve
auc<-auc(ROC1)
auc
```