Lab 10: Logistic Regression - II

Problem statement:

Refer attached blooddonation.CSV dataset. Develop a logistic regression classification model as:

- Class variable: donate
- Independent variable: remaining all
- With summary command observe the results
- Display the probability of each data record
- Calculate and display the assigned class with respect to cut off value 0.5.
- Calculate and display the confusion matrix
- Calculate the accuracy of model
- Calculate the error rate of model
- Calculate the recall of model
- Calculate the precision of model
- Construct the ROC curve
- Repeat the steps from (v) to (xi) with cutoff 0.2 and 0.8

Source Code:

```
#Author: Ashish Upadhyay
#Branch: Computer Science and Engineering
#Semester: 6th
#Dr. SP Mukherjee International Institute of Information Technology, Naya Raipur
#Subject: Machine Learning Lab 8
#Task: Logistic Regression Implementation - Part II

setwd("C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs")
getwd()

train <- read.csv("blooddonation.csv")
nrow(train)
head(train)

#install.packages('caTools')
library(caTools)

set.seed(88)
split <- sample.split(train$donate, SplitRatio = 0.75)

#get training and test data
dresstrain <- subset(train, split == TRUE)
dresstest <- subset(train, split == FALSE)

#Logistic Regression Model
model <- glm (donate ~ ., data = dresstrain, family = binomial)

#Summary
summary(model)

#Probability
```

```
probability <- predict(model, type = 'response')
probability


#Confusion Matrix (Cut-off value = 0.5)
con_mat <- table(dresstrain$donate, probability > 0.5)
con_mat

#Accuracy
accuracy <- ((con_mat[1, 1] + con_mat[2, 2])/(con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
accuracy

#Precision
precision <- ((con_mat[2, 2]) / (con_mat[1, 2] + con_mat[2, 2])) * 100
precision

#Recall
recall <- (con_mat[2, 2] / (con_mat[2, 1] + con_mat[2, 2])) * 100
recall

#Error Rate
error_rate <- (con_mat[1, 1] / (con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
error_rate

#ROCR Curve
#install.packages('ROCR')
library(ROCR)
ROCRpred <- prediction(probability, dresstrain$donate)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))


#Confusion Matrix (Cut-off value = 0.2)
con_mat <- table(dresstrain$donate, probability > 0.2)
con_mat

#Accuracy
accuracy <- ((con_mat[1, 1] + con_mat[2, 2])/(con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
accuracy

#Precision
precision <- ((con_mat[2, 2]) / (con_mat[1, 2] + con_mat[2, 2])) * 100
precision

#Recall
recall <- (con_mat[2, 2] / (con_mat[2, 1] + con_mat[2, 2])) * 100
recall

#Error Rate
error_rate <- (con_mat[1, 1] / (con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
error_rate
```

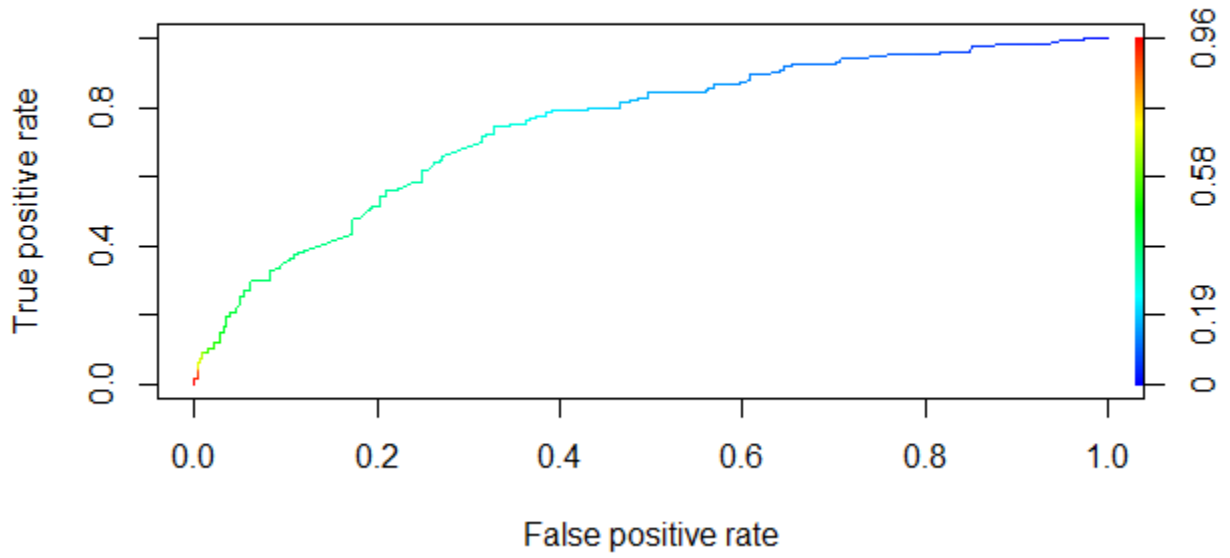
```
#ROCR Curve
#install.packages('ROCR')
library(ROCR)
ROCRpred <- prediction(probability, dresstrain$donate)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))


#Confusion Matrix (Cut-off value = 0.8)
con_mat <- table(dresstrain$donate, probability > 0.8)
con_mat

#Accuracy
accuracy <- ((con_mat[1, 1] + con_mat[2, 2])/(con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
accuracy

#Precision
precision <- ((con_mat[2, 2]) / (con_mat[1, 2] + con_mat[2, 2])) * 100
precision

#Recall
recall <- (con_mat[2, 2] / (con_mat[2, 1] + con_mat[2, 2])) * 100
recall

#Error Rate
error_rate <- (con_mat[1, 1] / (con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
error_rate

#ROCR Curve
#install.packages('ROCR')
library(ROCR)
ROCRpred <- prediction(probability, dresstrain$donate)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```

Output:

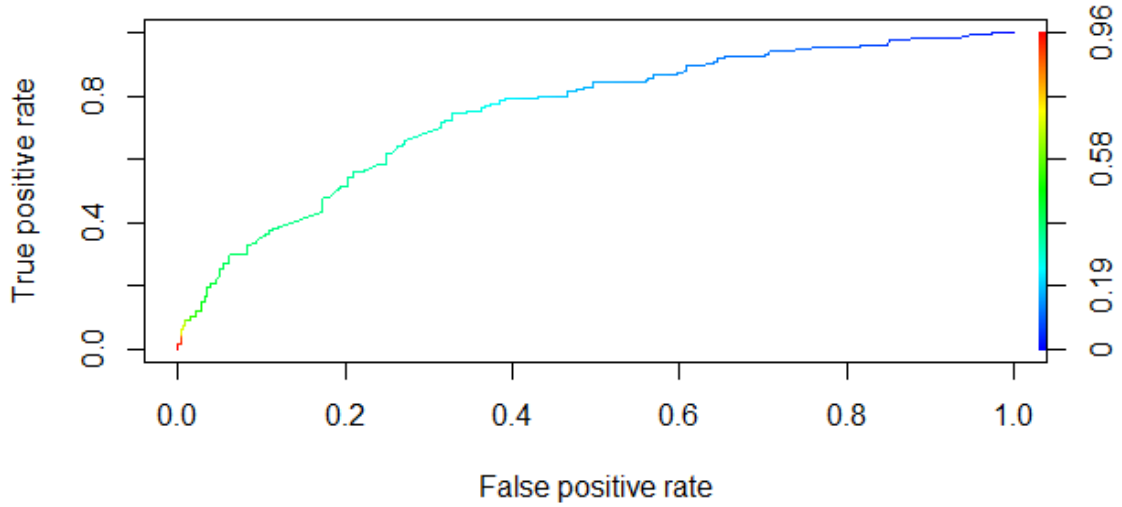
```
> #Author: Ashish Upadhyay
> #Branch: Computer Science and Engineering
> #Semester: 6th
> #Dr. SP Mukherjee International Institute of Information Technology, Naya Raipur
> #Subject: Machine Learning Lab 8
> #Task: Logistic Regression Implementation - Part II
>
> setwd("C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs")
> getwd()
[1] "C:/Users/Ashish Upadhyay/Documents/Semester6/MachineLearning/Lab Programs"
>
> train <- read.csv("blooddonation.csv")
> nrow(train)
[1] 748
```

```
> head(train)
  Recency..months. Frequency..times. Monetary..c.c..blood. Time..months. donate
1                2                50                12500             98      1
2                0                13                 3250             28      1
3                1                16                 4000             35      1
4                2                20                 5000             45      1
5                1                24                 6000             77      0
6                4                 4                 1000              4      0


> library(caTools)
>
> set.seed(88)
> split <- sample.split(train$donate, SplitRatio = 0.75)
>
> #get training and test data
> dresstrain <- subset(train, split == TRUE)
> dresstest <- subset(train, split == FALSE)
>
> #Logistic Regression Model
> model <- glm (donate ~ ., data = dresstrain, family = binomial)
>
> #Summary
> summary(model)

Call:
glm(formula = donate ~ ., family = binomial, data = dresstrain)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2778  -0.8168  -0.4871  -0.1833   2.6719

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         -0.44933    0.20315  -2.212  0.02698 *
Recency..months.    -0.10811    0.02035  -5.313 1.08e-07 ***
Frequency..times.    0.11010    0.02684   4.102 4.09e-05 ***
Monetary..c.c..blood.      NA         NA      NA       NA
Time..months.       -0.01696    0.00649  -2.614  0.00896 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 617.38  on 561  degrees of freedom
Residual deviance: 535.65  on 558  degrees of freedom
AIC: 543.65

Number of Fisher Scoring iterations: 5

>
> #Probability
> probability <- predict(model, type = 'response')
> probability
```

```
> #Confusion Matrix (Cut-off value = 0.5)
> con_mat <- table(dresstrain$donate, probability > 0.5)
> con_mat

   FALSE TRUE
 0   417   11
 1   118   16

> #Accuracy
> accuracy <- ((con_mat[1, 1] + con_mat[2, 2])/(con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
> accuracy
[1] 77.04626
> #Precision
> precision <- ((con_mat[2, 2]) / (con_mat[1, 2] + con_mat[2, 2])) * 100
> precision
[1] 59.25926
> #Recall
> recall <- (con_mat[2, 2] / (con_mat[2, 1] + con_mat[2, 2])) * 100
> recall
[1] 11.9403
> #Error Rate
> error_rate <- (con_mat[1, 1] / (con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
> error_rate
[1] 74.19929
> #ROCR Curve
> #install.packages('ROCR')
> library(ROCR)
> ROCRpred <- prediction(probability, dresstrain$donate)
> ROCRperf <- performance(ROCRpred, 'tpr','fpr')
> plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```



```
> #Confusion Matrix (Cut-off value = 0.2)
> con_mat <- table(dresstrain$donate, probability > 0.2)
> con_mat

   FALSE TRUE
 0   239  189
 1    27  107
>
> #Accuracy
> accuracy <- ((con_mat[1, 1] + con_mat[2, 2])/(con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
> accuracy
[1] 61.56584
> #Precision
> precision <- ((con_mat[2, 2]) / (con_mat[1, 2] + con_mat[2, 2])) * 100
> precision
[1] 36.14865
> #Recall
> recall <- (con_mat[2, 2] / (con_mat[2, 1] + con_mat[2, 2])) * 100
> recall
[1] 79.85075
> #Error Rate
> error_rate <- (con_mat[1, 1] / (con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
> error_rate
[1] 42.52669
>
> #ROCR Curve
> #install.packages('ROCR')
> library(ROCR)
> ROCRpred <- prediction(probability, dresstrain$donate)
> ROCRperf <- performance(ROCRpred, 'tpr','fpr')
> plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```



```
> #Confusion Matrix (Cut-off value = 0.8)
> con_mat <- table(dresstrain$donate, probability > 0.8)
> con_mat

   FALSE TRUE
 0   427    1
 1   129    5
> #Accuracy
> accuracy <- ((con_mat[1, 1] + con_mat[2, 2])/(con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
> accuracy
[1] 76.86833
> #Precision
> precision <- ((con_mat[2, 2]) / (con_mat[1, 2] + con_mat[2, 2])) * 100
> precision
[1] 83.33333
> #Recall
> recall <- (con_mat[2, 2] / (con_mat[2, 1] + con_mat[2, 2])) * 100
> recall
[1] 3.731343
> #Error Rate
> error_rate <- (con_mat[1, 1] / (con_mat[1, 1] + con_mat[1, 2] + con_mat[2, 1] + con_mat[2, 2])) * 100
> error_rate
[1] 75.97865
> #ROCR Curve
> #install.packages('ROCR')
> library(ROCR)
> ROCRpred <- prediction(probability, dresstrain$donate)
> ROCRperf <- performance(ROCRpred, 'tpr','fpr')
> plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```