**Practical 10**

**AIM:** Perform the logistic regression on the given data warehouse data.

**Solution:**

To perform this download the data from the link given below:

<https://github.com/TarekDib03/Analytics/blob/master/Week3%20-%20Logistic%20Regression/Data/quality.csv>

Code:

> quality <- read.csv('C:/Users/Anon/Desktop/BI Practical/quality.csv')

> str(quality)

Output:

'data.frame': 131 obs. of 14 variables:

$ MemberID : int 1 2 3 4 5 6 7 8 9 10 ...

$ InpatientDays : int 0 1 0 0 8 2 16 2 2 4 ...

$ ERVisits : int 0 1 0 1 2 0 1 0 1 2 ...

$ OfficeVisits : int 18 6 5 19 19 9 8 8 4 0 ...

$ Narcotics : int 1 1 3 0 3 2 1 0 3 2 ...

$ DaysSinceLastERVisit: num 731 411 731 158 449 ...

$ Pain : int 10 0 10 34 10 6 4 5 5 2 ...

$ TotalVisits : int 18 8 5 20 29 11 25 10 7 6 ...

$ ProviderCount : int 21 27 16 14 24 40 19 11 28 21 ...

$ MedicalClaims : int 93 19 27 59 51 53 40 28 20 17 ...

$ ClaimLines : int 222 115 148 242 204 156 261 87 98 66 ...

$ StartedOnCombination: logi FALSE FALSE FALSE FALSE FALSE FALSE ...

$ AcuteDrugGapSmall : int 0 1 5 0 0 4 0 0 0 0 ...

$ PoorCare : int 0 0 0 0 0 1 0 0 1 0 ...

> table(quality$PoorCare)

0 1

98 33

> 98/131

[1] 0.7480916

> install.packages("caTools")

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/Anon/AppData/Local/R/win-library/4.2’

(as ‘lib’ is unspecified)

also installing the dependency ‘bitops’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/bitops\_1.0-7.zip'

Content type 'application/zip' length 31679 bytes (30 KB)

downloaded 30 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caTools\_1.18.2.zip'

Content type 'application/zip' length 245985 bytes (240 KB)

downloaded 240 KB

package ‘bitops’ successfully unpacked and MD5 sums checked

package ‘caTools’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Anon\AppData\Local\Temp\RtmpCyEvnb\downloaded\_packages

> library(caTools)

> set.seed(88)

> split = sample.split(quality$PoorCare, SplitRatio = 0.75)

> split

[1] TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE

[13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[25] FALSE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE

[37] FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE

[49] FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[61] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE

[73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE

[85] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE

[97] TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

[109] TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE

[121] FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

> qualityTrain = subset(quality, split == TRUE)

> qualityTest = subset(quality, split == FALSE)

> nrow(qualityTrain)

[1] 99

> nrow(qualityTest)

[1] 32

> QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics, data = qualityTrain, family = binomial)

> summary(QualityLog)

Call:

glm(formula = PoorCare ~ OfficeVisits + Narcotics, family = binomial,

data = qualityTrain)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.06303 -0.63155 -0.50503 -0.09689 2.16686

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.64613 0.52357 -5.054 4.33e-07 \*\*\*

OfficeVisits 0.08212 0.03055 2.688 0.00718 \*\*

Narcotics 0.07630 0.03205 2.381 0.01728 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 111.888 on 98 degrees of freedom

Residual deviance: 89.127 on 96 degrees of freedom

AIC: 95.127

Number of Fisher Scoring iterations: 4

> predictTrain = predict(QualityLog, type = "response")

> summary(predictTrain)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.06623 0.11912 0.15967 0.25253 0.26765 0.98456

> tapply(predictTrain, qualityTrain$PoorCare, mean)

0 1

0.1894512 0.4392246

> table(qualityTrain$PoorCare, predictTrain > 0.5)

FALSE TRUE

0 70 4

1 15 10

> 10/25

[1] 0.4

> 70/74

[1] 0.9459459

> table(qualityTrain$PoorCare, predictTrain > 0.7)

FALSE TRUE

0 73 1

1 17 8

> 8/25

[1] 0.32

> 73/74

[1] 0.9864865

> table(qualityTrain$PoorCare, predictTrain > 0.2)

FALSE TRUE

0 54 20

1 9 16

> 16/25

[1] 0.64

> 54/74

[1] 0.7297297

> install.packages("ROCR")

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/Anon/AppData/Local/R/win-library/4.2’

(as ‘lib’ is unspecified)

also installing the dependencies ‘gtools’, ‘gplots’

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/gtools\_3.9.4.zip'

Content type 'application/zip' length 359411 bytes (350 KB)

downloaded 350 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/gplots\_3.1.3.zip'

Content type 'application/zip' length 603183 bytes (589 KB)

downloaded 589 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/ROCR\_1.0-11.zip'

Content type 'application/zip' length 453649 bytes (443 KB)

downloaded 443 KB

package ‘gtools’ successfully unpacked and MD5 sums checked

package ‘gplots’ successfully unpacked and MD5 sums checked

package ‘ROCR’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

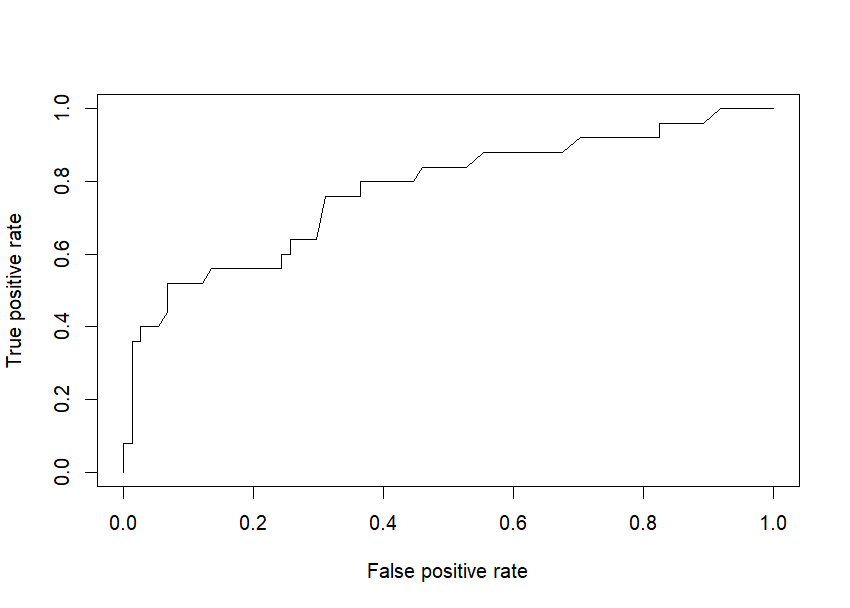
C:\Users\Anon\AppData\Local\Temp\RtmpCyEvnb\downloaded\_packages

> library(ROCR)

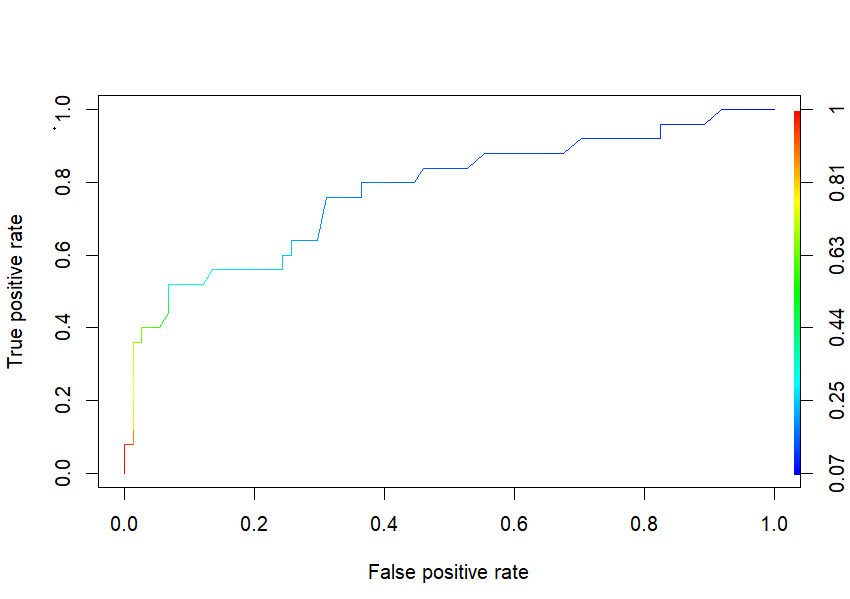
> ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)

> ROCRpref = performance(ROCRpred, "tpr", "fpr")

> plot(ROCRpref)



> plot(ROCRpref, colorize=TRUE)



Complete code:

quality <- read.csv('C:/Users/Anon/Desktop/BI Practical/quality.csv')

str(quality)

table(quality$PoorCare)

98/131

install.packages("caTools")

library(caTools)

set.seed(88)

split = sample.split(quality$PoorCare, SplitRatio = 0.75)

split

qualityTrain = subset(quality, split == TRUE)

qualityTest = subset(quality, split == FALSE)

nrow(qualityTrain)

nrow(qualityTest)

QualityLog = glm(PoorCare ~ OfficeVisits + Narcotics, data = qualityTrain, family = binomial)

summary(QualityLog)

predictTrain = predict(QualityLog, type = "response")

summary(predictTrain)

tapply(predictTrain, qualityTrain$PoorCare, mean)

table(qualityTrain$PoorCare, predictTrain > 0.5)

10/25

70/74

table(qualityTrain$PoorCare, predictTrain > 0.7)

8/25

73/74

table(qualityTrain$PoorCare, predictTrain > 0.2)

16/25

54/74

install.packages("ROCR")

library(ROCR)

ROCRpred = prediction(predictTrain, qualityTrain$PoorCare)

ROCRpref = performance(ROCRpred, "tpr", "fpr")

plot(ROCRpref)

plot(ROCRpref, colorize=TRUE)

plot(ROCRpref, colorize=TRUE, print, print.cutoffs.at = seq(0,1,by = 0.1), text.adj = c(-0.2,0.7))