3

February 1, 2024

1 Logistic Regression In R

[21]:
```
# Summary of dataset in package
summary(mtcars)
```

```
      mpg             cyl             disp             hp       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
      drat             wt             qsec             vs        
 Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
 1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
 Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
 Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
 3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
 Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
       am              gear            carb      
 Min.   :0.0000   Min.   :3.000   Min.   :1.000  
 1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
 Median :0.0000   Median :4.000   Median :2.000  
 Mean   :0.4062   Mean   :3.688   Mean   :2.812  
 3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

Logistic regression is implemented in R using glm() by training the model using features or variables in the dataset.

[22]:
```
# Installing the package

# For Logistic regression
install.packages("caTools")

# For ROC curve to evaluate model
install.packages("ROCR")
```

```
# Loading package
library(caTools)
library(ROCR)
```

```
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

[23]:
```
# Splitting dataset
split <- sample.split(mtcars, SplitRatio = 0.8)
split

train_reg <- subset(mtcars, split == "TRUE")
test_reg <- subset(mtcars, split == "FALSE")

# Training model
logistic_model <- glm(vs ~ wt + disp,
                                        data = train_reg,
                                        family = "binomial")
logistic_model

# Summary
summary(logistic_model)
```

1. FALSE 2. TRUE 3. TRUE 4. TRUE 5. TRUE 6. FALSE 7. TRUE 8. TRUE 9. TRUE 10. TRUE 11. FALSE

```
Call:  glm(formula = vs ~ wt + disp, family = "binomial", data = train_reg)

Coefficients:
(Intercept)           wt         disp
    1.44440      1.54534     -0.03262

Degrees of Freedom: 23 Total (i.e. Null);  21 Residual
Null Deviance:            33.1
Residual Deviance: 15.8         AIC: 21.8


Call:
glm(formula = vs ~ wt + disp, family = "binomial", data = train_reg)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.44440    3.35050   0.431   0.6664
wt           1.54534    1.94984   0.793   0.4280
```

```
disp        -0.03262    0.01797  -1.815   0.0695 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 33.104  on 23  degrees of freedom
Residual deviance: 15.801  on 21  degrees of freedom
AIC: 21.801

Number of Fisher Scoring iterations: 6
```

1.1 Predict test data based on model

[24]:
```
predict_reg <- predict(logistic_model,
                                        test_reg, type = "response")
predict_reg
```

Mazda RX4 0.56811772067142 **Valiant** 0.366336432517721 **Merc 280C** 0.784730885831396 **Merc 450SE** 0.220569990126755 **Chrysler Imperial** 0.00948925420350044 **Dodge Challenger** 0.0296325009949015 **AMC Javelin** 0.0405629141446895 **Lotus Europa** 0.663838948193045

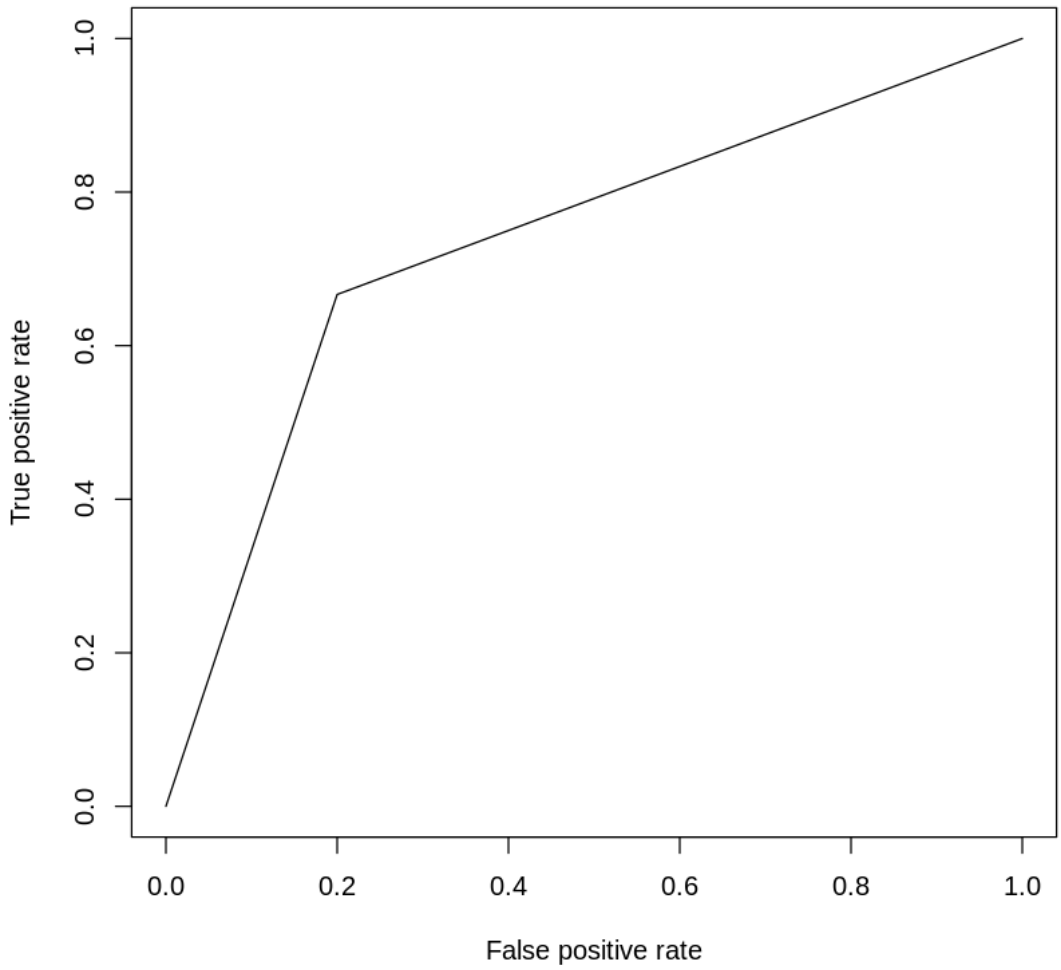
[26]:
```
# Changing probabilities
predict_reg <- ifelse(predict_reg >0.5, 1, 0)

# Evaluating model accuracy
# using confusion matrix
table(test_reg$vs, predict_reg)

missing_classerr <- mean(predict_reg != test_reg$vs)
print(paste('Accuracy =', 1 - missing_classerr))

# ROC-AUC Curve
ROCPred <- prediction(predict_reg, test_reg$vs)
ROCPer <- performance(ROCPred, measure = "tpr",
                                        x.measure = "fpr")

auc <- performance(ROCPred, measure = "auc")
auc <- auc@y.values[[1]]
auc

# Plotting curve
plot(ROCPer)
plot(ROCPer, colorize = TRUE,
        print.cutoffs.at = seq(0.1, by = 0.1),
        main = "ROC CURVE")
abline(a = 0, b = 1)
```

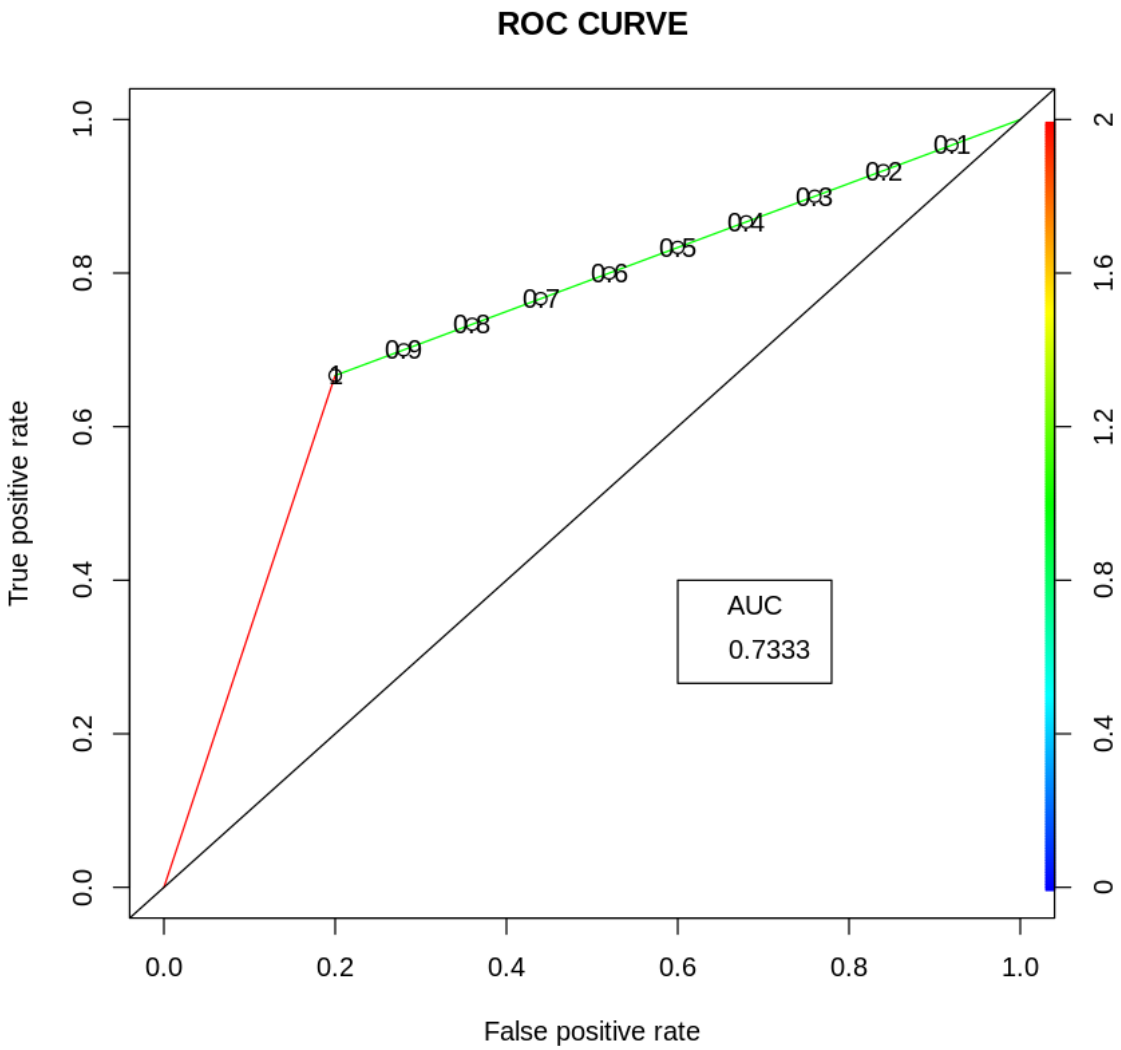
```
auc <- round(auc, 4)
legend(.6, .4, auc, title = "AUC", cex = 1)
```

```
   predict_reg
    0 1
  0 4 1
  1 1 2
```

```
[1] "Accuracy = 0.75"
```

0.733333333333333




ROC CURVE
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0
0.4
0.8
1.2
1.6
2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
AUC
0.7333