Logistic Regression

Justin Besteman

April 8, 2017

Clean up the past

```r
# Clean up
rm(list = ls())
```

Loading Libraries

```r
# Used to read in the data est
library(readr)

# Used to make the graph of the ROCR
library(ROCR)
```

Reading in the data of PassFail.dat

```r
# Loading in Data
theData <- read_delim(
  "~/Desktop/code/topics/logistic-regression/PassFail.dat",
  " ",
  escape_double = FALSE,
  col_names = FALSE,
  col_types =
    cols(
      X1 = col_skip(),
      X10 = col_skip(),
      X12 = col_skip(),
      X14 = col_skip(),
      X2 = col_skip(),
      X4 = col_skip(),
      X6 = col_skip(),
      X8 = col_skip()
    ),
  na = "null",
  trim_ws = TRUE
)

# Making the theData into a data.frame
PassFail <- data.frame(theData)

# Renaming columns
colnames(PassFail) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6")
```

Number of Observations

```r
# Number of Observation will hold the number of observation
numberOfObservation <-  nrow(PassFail)

numberOfObservation
```

```
## [1] 10000
```

Making sample data and test data

```r
# Setting the set seed of the random number generator for consistent testing
set.seed(123321)

# Making index a vector of that holds the number of observation
# I.E (1,2,3, ..... , 10000)
index <- c(1:numberOfObservation)

# Setting random6000 to the random sample of 6000
# Using sample() to grab 6000 random sample
random6000 <- sample(index, numberOfObservation * .60)

# trainingSamplePassFail to random sample of indices of index and
# pulling then from PassFail data
trainingSamplePassFail <- PassFail[random6000 ,]

# testData will hold the rest of the 4000 data sampl
testData <-  PassFail[-random6000 ,]
```

Running GLM

```r
# Running glm on the trainSamplePassFail data
# Formula y = x1 + x2 + x3 + x4 + x5 + x6
model <- glm(y  ~ . , data = trainingSamplePassFail , family = binomial(link = "logit"))

model
```

```
##
## Call:  glm(formula = y ~ ., family = binomial(link = "logit"), data = trainingSamplePassFail)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4
##    -1.60329      1.57422      0.81495      0.40369      0.19858
##          x5           x6
##     0.07576      0.07100
##
## Degrees of Freedom: 5999 Total (i.e. Null);  5993 Residual
## Null Deviance:       8132
## Residual Deviance: 7608  AIC: 7622
```

Observation scoring the test data

```
# Testing the prediction

# Testing the model against the testData
results <- predict(model, testData ,type='response')

# Using ifelse to test it best on .5
results <- ifelse(results > 0.5,1,0)

# Finding the error
error <- mean(results != testData$y)

error
```

```
## [1] 0.34825
```

```
# Showing the Accuracy of the model
accuracy <- 1 - error

accuracy
```

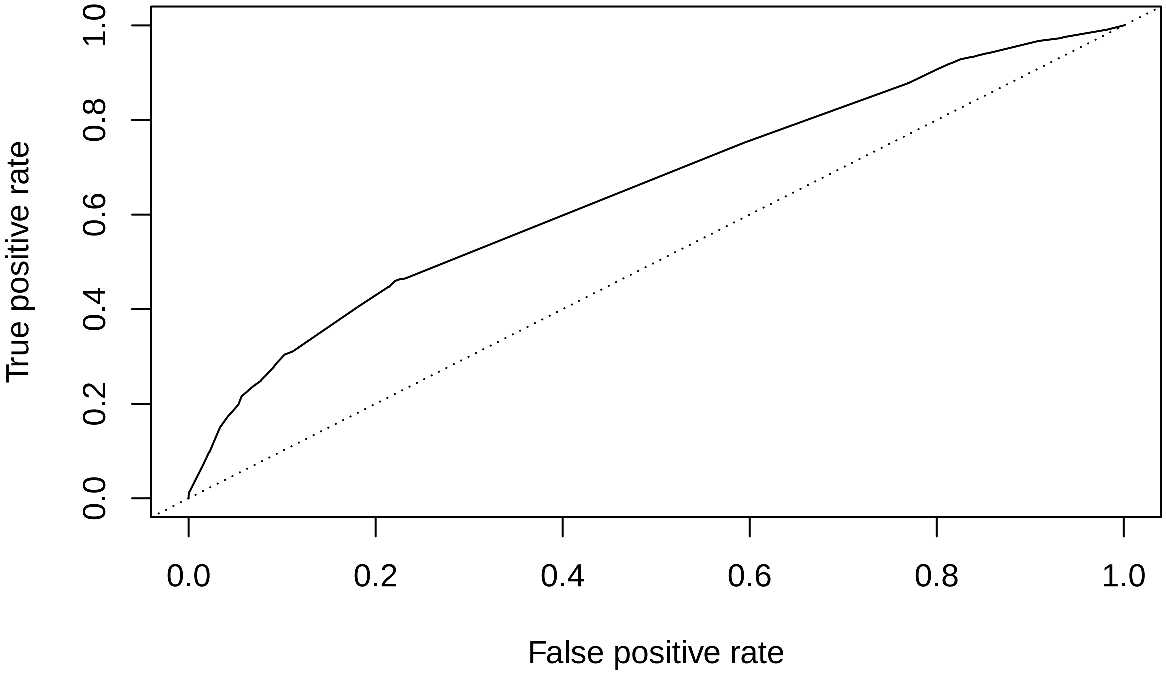
```
## [1] 0.65175
```

Making the plot of the curve of the ROCR

```
# Using the ROCR Package here



p <- predict(model,testData, type="response")
pr <- prediction(p, testData$y)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")

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

```
## [1] 0.650838
```

```
# Ploting the curve
plot(prf)

# Plotting the random guessing line
abline(a = 0 , b = 1 , lty = 3)
```


1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate