# Data621_Homework2

*Jenny*

*October 14, 2018*

install.packages("caret") install.packages("lattice") install.packages("rlang") install.packages("ggplot2") install.packages("pROC") install.packages("caret")

## Download the dataset and read data.

```
classification_output_data <- read.csv("https://raw.githubusercontent.com/JennierJ/DATA621/master/Homew

head(classification_output_data)
```

```
##   pregnant glucose diastolic skinfold insulin  bmi pedigree age class
## 1        7     124        70       33     215 25.5    0.161  37     0
## 2        2     122        76       27     200 35.9    0.483  26     0
## 3        3     107        62       13      48 22.9    0.678  23     1
## 4        1      91        64       24       0 29.2    0.192  21     0
## 5        4      83        86       19       0 29.3    0.317  34     0
## 6        1     100        74       12      46 19.5    0.149  28     0
##   scored.class scored.probability
## 1            0         0.32845226
## 2            0         0.27319044
## 3            0         0.10966039
## 4            0         0.05599835
## 5            0         0.10049072
## 6            0         0.05515460
```

## Raw Confusion Matrix

```
Target <- classification_output_data$class
Model <- classification_output_data$scored.class

confusion_matrix <- table(Model, Target)


colnames(confusion_matrix) <- c("Target Negative", "Target Positive")
rownames(confusion_matrix) <- c("Model Negative", "Model Positive")
#confusion_matrix <- table(classification_output_data$class, classification_output_data$scored.class)

confusion_matrix
```

```
##                   Target
## Model              Target Negative Target Positive
##   Model Negative               119              30
##   Model Positive                 5              27
```

**The rows in the confusion matrix represent the predicted class, and the columns represent the actual class.**

## Function for accuracy of the predictions.

```
accuracy <- function(con_df){
  Target <- con_df$class
  Model <- con_df$scored.class
  confusion_matrix <- table(Model, Target)
  TN <- confusion_matrix[1,1]
  TP <- confusion_matrix[2,2]
  FN <- confusion_matrix[1,2]
  FP <- confusion_matrix[2,1]

  con_accuracy <- (TP + TN)/ (TP + FP + TN + FN)
  return(con_accuracy)
}

(accuracy <- accuracy(classification_output_data))
```

```
## [1] 0.8066298
```

## Funcion for classification error rate of the predictions

```
classificaion_Error_Rate <- function(con_df){
  Target <- con_df$class
  Model <- con_df$scored.class
  confusion_matrix <- table(Model, Target)
  TN <- confusion_matrix[1,1]
  TP <- confusion_matrix[2,2]
  FN <- confusion_matrix[1,2]
  FP <- confusion_matrix[2,1]

  error_rate <- (FP + FN)/ (TP + FP + TN + FN)
  return(error_rate)
}

(classificaion_Error_Rate <- classificaion_Error_Rate(classification_output_data))
```

```
## [1] 0.1933702
```

```
# Verify the sum of accuracy and error
accuracy + classificaion_Error_Rate
```

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

## Function for the precision of the predictions

```
precision <- function(con_df){
  Target <- con_df$class
  Model <- con_df$scored.class
  confusion_matrix <- table(Model, Target)
  TN <- confusion_matrix[1,1]
  TP <- confusion_matrix[2,2]
  FN <- confusion_matrix[1,2]
  FP <- confusion_matrix[2,1]

  precision_value <- ((TP)/ (TP + FP))
  return(precision_value)
}

(precision <- precision(classification_output_data))
```

```
## [1] 0.84375
```

### Function for the sensitivity

```
sensitivity <- function(con_df){
  Target <- con_df$class
  Model <- con_df$scored.class
  confusion_matrix <- table(Model, Target)
  TN <- confusion_matrix[1,1]
  TP <- confusion_matrix[2,2]
  FN <- confusion_matrix[1,2]
  FP <- confusion_matrix[2,1]

  sensitivity_value <- ((TP)/ (TP + FN))
  return(sensitivity_value)
}

(sensitivity <- sensitivity(classification_output_data))
```

```
## [1] 0.4736842
```

### Function for the specificity

```
specificity <- function(con_df){
  Target <- con_df$class
  Model <- con_df$scored.class
  confusion_matrix <- table(Model, Target)
  TN <- confusion_matrix[1,1]
  TP <- confusion_matrix[2,2]
  FN <- confusion_matrix[1,2]
  FP <- confusion_matrix[2,1]

  specificity_value <- ((TN)/ (TN + FP))
  return(specificity_value)
}

(specificity <- specificity(classification_output_data))
```

```
## [1] 0.9596774
```

## F1 scores function

```
F1_scores <- function(con_df){
  Target <- con_df$class
  Model <- con_df$scored.class
  confusion_matrix <- table(Model, Target)
  TN <- confusion_matrix[1,1]
  TP <- confusion_matrix[2,2]
  FN <- confusion_matrix[1,2]
  FP <- confusion_matrix[2,1]

  precision_value <- ((TP)/ (TP + FP))
  sensitivity_value <- ((TP)/ (TP + FN))
  F1_scores_value <- 2 * precision_value * sensitivity_value / (precision_value + sensitivity_value)
  return(F1_scores_value)
}

(F1_scores <- F1_scores(classification_output_data))
```

```
## [1] 0.6067416
```

## F1 Range

```
Precisions <- runif(100000, min = 0, max = 1)
Sensitivities <- runif(100000, min = 0, max = 1)

max( 2* Precisions * Sensitivities / (Precisions + Sensitivities) )
```

```
## [1] 0.9981111
```

## Caret package

```
library(lattice)
library(ggplot2)
library(caret)

df_caret <- confusionMatrix(factor(classification_output_data$scored.class),
                            factor(classification_output_data$class)
)

df_caret
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 119  30
##          1   5  27
##
##                Accuracy : 0.8066
##                  95% CI : (0.7415, 0.8615)
##     No Information Rate : 0.6851
##     P-Value [Acc > NIR] : 0.0001712
##
##                   Kappa : 0.4916
##  Mcnemar's Test P-Value : 4.976e-05
##
##             Sensitivity : 0.9597
##             Specificity : 0.4737
##          Pos Pred Value : 0.7987
##          Neg Pred Value : 0.8438
##              Prevalence : 0.6851
##          Detection Rate : 0.6575
##    Detection Prevalence : 0.8232
##       Balanced Accuracy : 0.7167
##
##        'Positive' Class : 0
##
```

## pROC package

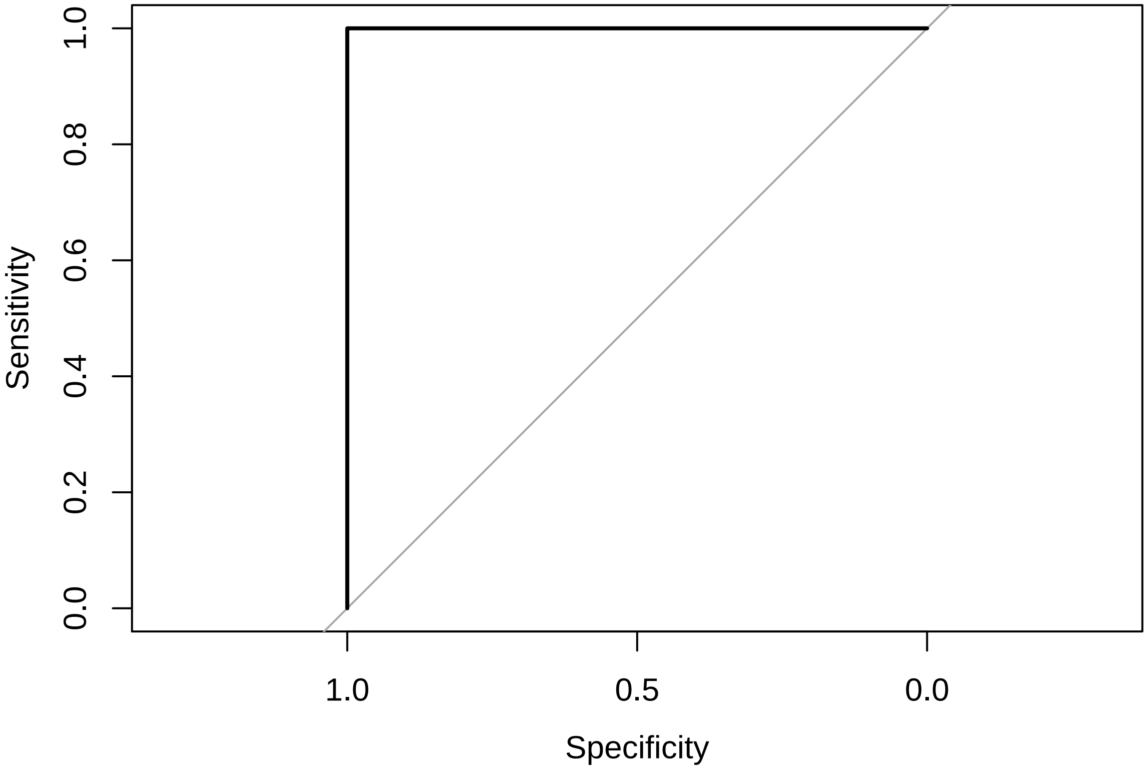
```r
#library(ggplot2)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
roc(classification_output_data$scored.class, classification_output_data$scored.probability, plot=T)
```



```
##
## Call:
## roc.default(response = classification_output_data$scored.class,     predictor = classification_outpu
##
## Data: classification_output_data$scored.probability in 149 controls (classification_output_data$scor
## Area under the curve: 1
```