

# Classification Metrics

*Abhishek Chaturvedi*

*May 17, 2017*

KNIT YOUR DOCUMENT AS *HTML* AND SUBMIT IT AND THE `Rmd` file.

## Classification Metrics Functions

Write functions of `y` and `y_hat` to calculate the following classification metrics

- True Positive Rate
- False Positive Rate
- True Negative Rate
- False Negative Rate
- Sensitivity
- Specificity
- Recall
- Precision
- Prevalence
- Accuracy
- Kappa

```
# test data
x <- c(1,1,0,1,1,0,1)
x1 <- c(1,1,1,0,0,1,1)

# function to get the true positive rate
# function accepts two vectors of observed and predicted values
tpr <- function(yo, yp){
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }
  return(sum(yo & yp)/sum(yo))
}
tpr(x,x1)
```

```
## [1] 0.6
```

```
fpr <- function(yo, yp) {
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }
  return(sum((yp != yo) & yp)/(length(yo)-sum(yo)))
}
fpr(x,x1)
```

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

```
tnr <- function(yo, yp) {
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }
  return(sum(!yo & !yp)/(length(yo)-sum(yo)))
}
tnr(x,x1)
```

```
## [1] 0
```

```
fnr <- function(yo, yp) {
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }
  return(sum(!yp & yo)/sum(yo))
}
fnr(x,x1)
```

```
## [1] 0.4
```

```
# sensitivity is true positive rate
sensitivity <- function (yo, yp){
  return(tpr(yo,yp))
}
sensitivity(x, x1)
```

```
## [1] 0.6
```

```
#specificity is true negative rate
specificity <- function(yo, yp){
  return(tnr(yo,yp))
}
specificity(x, x1)
```

```
## [1] 0
```

```
# recall is same as true positive rate
recall <- function(yo,yp){
  return(tpr(yo,yp))
}
recall(x,x1)
```

```
## [1] 0.6
```

```
# is true positive compare to predictied positives
precision <- function(yo,yp){
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }
  return(sum(yo & yp)/sum(yp))
}
precision(x,x1)
```

```
## [1] 0.6
```

```
# number of observed +ve cases in given data set
prevalence <- function (yo){
  if (is.null(yo)) {
    stop("NULL 'yo' value")
  }
  return(sum(yo)/length(yo))
}

prevalence(x1)
```

```
## [1] 0.7142857
```

```

# accuracy is total True +ve plus -ve / Observed +ve plus -ve
accuracy <- function(yo, yp){
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }

  return((sum(yo & yp) + sum(!yo & !yp))/length(yo))
}

accuracy(x, x1)

```

```
## [1] 0.4285714
```

```

kappa <- function(yo,yp){
  if (is.null(yo) || is.null(yp)) {
    stop("NULL 'yo' or 'yp' value")
  }
  if(length(yo) != length(yp)){
    stop(" number of predicted and observed responses are not same ")
  }

  # calculating matrix a,b,c,d

  # when observed and predicted both are true
  a <- sum(yo & yp)
  # when observed false and predicted is true
  b <- sum(!yo & yp)
  # when observed true and predicted is false
  c <- sum(yo & !yp)
  # when observed and predicted both are false
  d <- sum(!yo & !yp)

  # observed proportionate agreement
  p0 <- (a+d)/(a+b+c+d)

  # the probability of random agreement
  pe <- (((a+b)/(a+b+c+d))*((a+c)/(a+b+c+d))) + (((c+d)/(a+b+c+d))*((b+d)/(a+b+c+d)))

  return((p0-pe)/(1-pe))
}

kappa(x,x1)

```

```
## [1] -0.4
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

# Usage

Use the function(s) above to calculate the metrics for your late arrival model that you created last week.

```
#install.packages("biomod2")  
library("biomod2")
```

```
## Loading required package: sp
```

```
## Loading required package: raster
```

```
## Loading required package: parallel
```

```
## Loading required package: reshape
```

```
## Loading required package: ggplot2
```

```
## biomod2 3.3-7 loaded.  
##  
## Type browseVignettes(package='biomod2') to access directly biomod2 vignettes.
```

```
library('data.table')
```

```
##  
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:reshape':  
##  
##      melt
```

```
## The following object is masked from 'package:raster':  
##  
##      shift
```

```
flightsDataJoined <- readRDS("flightsDataJoined.rds")

y <- "arr_delay"
xs <- c('humid','dep_time', 'sched_dep_time','sched_arr_time','dep_delay','origin')
yx <- flightsDataJoined[,c(y,xs),with=FALSE]
yx <- na.omit(yx)
yx <- within(yx, gt22 <- arr_delay>=22)
model <- glm(formula = gt22 ~ . - arr_delay, family=binomial, data=yx)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#summary(model)
```

```
y1 <- yx$gt22
y2 <- BinaryTransformation(predict(model,yx, type="response"),0.5)
```

```
##
## *** in setMethod('BinaryTransformation', signature(data='numeric')
## *** in setMethod('BinaryTransformation', signature(data='data.frame')
```

```
tpr(y1,y2)
```

```
## [1] 0.6469822
```

```
tnr(y1,y2)
```

```
## [1] 0.983425
```

```
fpr(y1,y2)
```

```
## [1] 0.016575
```

```
fnr(y1,y2)
```

```
## [1] 0.3530178
```

```
sensitivity(y1,y2)
```

```
## [1] 0.6469822
```

```
specificity(y1,y2)
```

```
## [1] 0.983425
```

```
recall(y1,y2)
```

```
## [1] 0.6469822
```

```
precision(y1,y2)
```

```
## [1] 0.8705814
```

```
prevalence(y1)
```

```
## [1] 0.1470016
```

```
kappa(y1,y2)
```

```
## [1] 0.7053809
```