Classification Metrics

Saqib Ali

May 09, 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

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

## -------------------------------------------------------------------------

## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!

## -------------------------------------------------------------------------

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

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
```

```
##     week, yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
prob = plogis(predict(logit.fit, data.joined.testing))

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
prob<-ifelse(prob> 0.5,1,0)
data.joined.testing$prob = prob


confusionMatrix(data.joined.testing$arrival_delayed, prob)

##        0     1
## 0 93445  6545
## 1  1953 14879
y = data.joined.testing$arrival_delayed
y_hat = data.joined.testing$prob

#TP <- sum((y==0)&(y==y_hat))
#TN <- sum((y==1)&(y==y_hat))
#FP <- sum((y_hat==0)&(y==1))
#FN <- sum((y_hat==1)&(y==0))

Accuracy <- function(y, y_hat){
  (sum((y==0)&(y==y_hat)) + sum((y==1)&(y==y_hat))) / (sum((y==0)&(y==y_hat)) + sum((y==1)&(y==y_hat)) +
}

TruePositiveRate <- function(y, yhat){
 sum((y==0)&(y==y_hat))/(sum((y==0)&(y==y_hat)) + sum((y_hat==1)&(y==0)))

}

FalsePositiveRate <- function(y, y_hat){
  sum((y_hat==0)&(y==1))/(sum((y_hat==0)&(y==1)) + sum((y==1)&(y==y_hat)))
}

TrueNegativeRate  <- function(y, y_hat){
  sum((y==1)&(y==y_hat))/(sum((y==1)&(y==y_hat)) + sum((y_hat==0)&(y==1)))

}

FalseNegativeRate  <- function(y, y_hat){
```

```r
  sum((y_hat==1)&(y==0))/(sum((y_hat==1)&(y==0)) + sum((y==0)&(y==y_hat)))
}

Sensitivity <- TruePositiveRate
Specificity <- TrueNegativeRate

Recall <- TruePositiveRate


Precision <- function(y, y_hat){
  sum((y==0)&(y==y_hat)) / (sum((y==0)&(y==y_hat))+sum((y_hat==0)&(y==1)))
}

Prevalence <- function(y, y_hat){
  sum(y==1)/length(y)

}

# Kappa
```

```r
Accuracy(y,y_hat)
```

```
## [1] 0.9272569
```

```r
1-misClassError(data.joined.testing$arrival_delayed, prob)
```

```
## [1] 0.9273
```

```r
TrueNegativeRate(y,y_hat)
```

```
## [1] 0.6945015
```

```r
sensitivity(data.joined.testing$arrival_delayed, prob)
```

```
## [1] 0.6945015
```

```r
TruePositiveRate(y,y_hat)
```

```
## [1] 0.9795279
```

```r
FalsePositiveRate(y,y_hat)
```

```
## [1] 0.3054985
```

```r
FalseNegativeRate(y,y_hat)
```

```
## [1] 0.02047213
```

```r
Sensitivity(y,y_hat)
```

```
## [1] 0.9795279
```

```r
Specificity(y,y_hat)
```

```
## [1] 0.6945015
```

```r
specificity(data.joined.testing$arrival_delayed, prob)
```

```
## [1] 0.9795279
```

```r
Recall(y,y_hat)
```

```
## [1] 0.9795279
```

```r
Precision(y,y_hat)
```

```
## [1] 0.9345435
```

```r
Prevalence(y,y_hat)
```

```
## [1] 0.1833901
```

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.