

Confusion-Matrix.R

patriciamaya

2020-10-05

```
*****DRAW CONFUSION MATRIX FOR CLASSIFICATION***

#Pred_class <- predict(, newdata= test, type="response")
#actual <- test$Target

Pred_class #vector of predicted classes
Actual #vector of true labels

#Option 1 - simple confusion matrix
#table(Pred_class, Actual, dnn = c("Predictions", "Actual"))

#Option 2- more advanced confusion matrix w/ statistics
library(caret)

#confusionMatrix(Pred_class, reference = Actual, positive= "YES")
#sensitivity = recall

#Function to calculate accuracy from Option 1
my.accuracy <- function(actual, predictions)
{
  y<- as.vector(table(predictions, actual))
  names(y) <- c("TN", "FP", "FN", "TP")
  accuracy<- (y["TN"]+ y["TP"])/ sum(y)
  return(as.numeric(accuracy))
}
my.accuracy(actual, Pred_class)

#EXAMPLE
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.2

data("Carseats")
attach(Carseats)
High <- as.factor(ifelse(Sales >= 8, "YES", "NO"))
Data <- data.frame(Carseats, High)
Data <- Data[, -1]

indx <- sample(2, nrow(Data), replace=T, prob= c(0.8,0.2))
train <- Data[indx==1, ]
test <- Data[indx==2, ]
```

```

logitModel <- glm(High ~ ., data= train, family="binomial")
predictions<- predict(logitModel, newdata= test, type ="response")
predicted_class <- as.factor(ifelse(predictions >= 0.5, "YES", "NO"))
Actual <- test$High

table(predicted_class, Actual, dnn = c("Predictions", "Actual"))

##           Actual
## Predictions NO YES
##           NO  47   5
##           YES   1  27

my.accuracy(Actual, predicted_class)

## [1] 0.925

confusionMatrix(predicted_class, reference = Actual, positive= "YES")

## Confusion Matrix and Statistics
##
##           Reference
## Prediction NO YES
##           NO  47   5
##           YES   1  27
##
##               Accuracy : 0.925
##               95% CI : (0.8439, 0.972)
##       No Information Rate : 0.6
##       P-Value [Acc > NIR] : 5.34e-11
##
##               Kappa : 0.8404
##
##  Mcnemar's Test P-Value : 0.2207
##
##               Sensitivity : 0.8438
##               Specificity : 0.9792
##               Pos Pred Value : 0.9643
##               Neg Pred Value : 0.9038
##               Prevalence : 0.4000
##               Detection Rate : 0.3375
##       Detection Prevalence : 0.3500
##       Balanced Accuracy : 0.9115
##
##       'Positive' Class : YES
##

```