mchris26_3

Mark Christian

10/18/2020

```
library(e1071)
FD <- read.csv("FlightDelays.csv")
```

```
FD$DAY_WEEK <- factor(FD$DAY_WEEK)
FD$DEP_TIME <- factor(FD$DEP_TIME)
FD$CRS_DEP_TIME <- factor(round(FD$CRS_DEP_TIME/100))
```

#I had to change the numerical to catagorical and also creat bins for hourly departure time.

```
#install.packages("ISLR")
library(ISLR)
library(caret)
FD_Pred <- FD[,c(10, 1, 8, 4, 2, 13)]
train.index <- createDataPartition(FD_Pred$Flight.Status,p=0.6,list = FALSE)
train.df <- FD_Pred[train.index, ]
valid.df <- FD_Pred[-train.index, ]
```

#here I created training and validation sets.

```
nb_delays <- naiveBayes(Flight.Status ~ ., data = train.df)
head(nb_delays)
```

```
## $apriori
## Y
## delayed  ontime
##     257    1064
##
## $tables
## $tables$DAY_WEEK
##          DAY_WEEK
## Y                 1         2         3         4         5         6
##   delayed 0.17898833 0.16342412 0.14785992 0.14007782 0.17509728 0.04669261
##   ontime  0.11748120 0.13721805 0.14285714 0.18703008 0.17951128 0.12687970
##          DAY_WEEK
## Y                 7
##   delayed 0.14785992
##   ontime  0.10902256
##
## $tables$CRS_DEP_TIME
##          CRS_DEP_TIME
```

```
## Y                   6           7           8           9          10
##   delayed 0.038910506 0.054474708 0.070038911 0.027237354 0.011673152
##   ontime  0.069548872 0.056390977 0.083646617 0.064849624 0.050751880
##          CRS_DEP_TIME
## Y                  11          12          13          14          15
##   delayed 0.007782101 0.054474708 0.038910506 0.050583658 0.182879377
##   ontime  0.032894737 0.056390977 0.062969925 0.064849624 0.110902256
##          CRS_DEP_TIME
## Y                  16          17          18          19          20
##   delayed 0.097276265 0.143968872 0.031128405 0.089494163 0.019455253
##   ontime  0.069548872 0.105263158 0.041353383 0.042293233 0.029135338
##          CRS_DEP_TIME
## Y                  21
##   delayed 0.081712062
##   ontime  0.059210526
##
## $tables$ORIGIN
##          ORIGIN
## Y                BWI        DCA        IAD
##   delayed 0.07003891 0.53696498 0.39299611
##   ontime  0.05827068 0.66823308 0.27349624
##
## $tables$DEST
##          DEST
## Y               EWR       JFK       LGA
##   delayed 0.3696498 0.1906615 0.4396887
##   ontime  0.2922932 0.1503759 0.5573308
##
## $tables$CARRIER
##          CARRIER
## Y                  CO         DH         DL         MQ         OH         RU
##   delayed 0.06225681 0.31517510 0.11673152 0.19455253 0.01167315 0.20233463
##   ontime  0.04041353 0.20864662 0.19548872 0.12781955 0.01503759 0.18233083
##          CARRIER
## Y                  UA         US
##   delayed 0.01556420 0.08171206
##   ontime  0.01691729 0.21334586
##
##
## $levels
## NULL
##
## $isnumeric
##     DAY_WEEK CRS_DEP_TIME       ORIGIN         DEST      CARRIER
##        FALSE        FALSE        FALSE        FALSE        FALSE
##
## $call
## naiveBayes.default(x = X, y = Y, laplace = laplace)
```

#Naive Bayes to see if the flights are delayed or not

```
prop.table(table(train.df$Flight.Status, train.df$DEST), margin = 1)
```

```
##
```

```
##                 EWR       JFK       LGA
##   delayed 0.3696498 0.1906615 0.4396887
##   ontime  0.2922932 0.1503759 0.5573308
```

```
nb_pred <- predict(nb_delays, newdata = valid.df, type = "raw")
head(nb_pred)
```

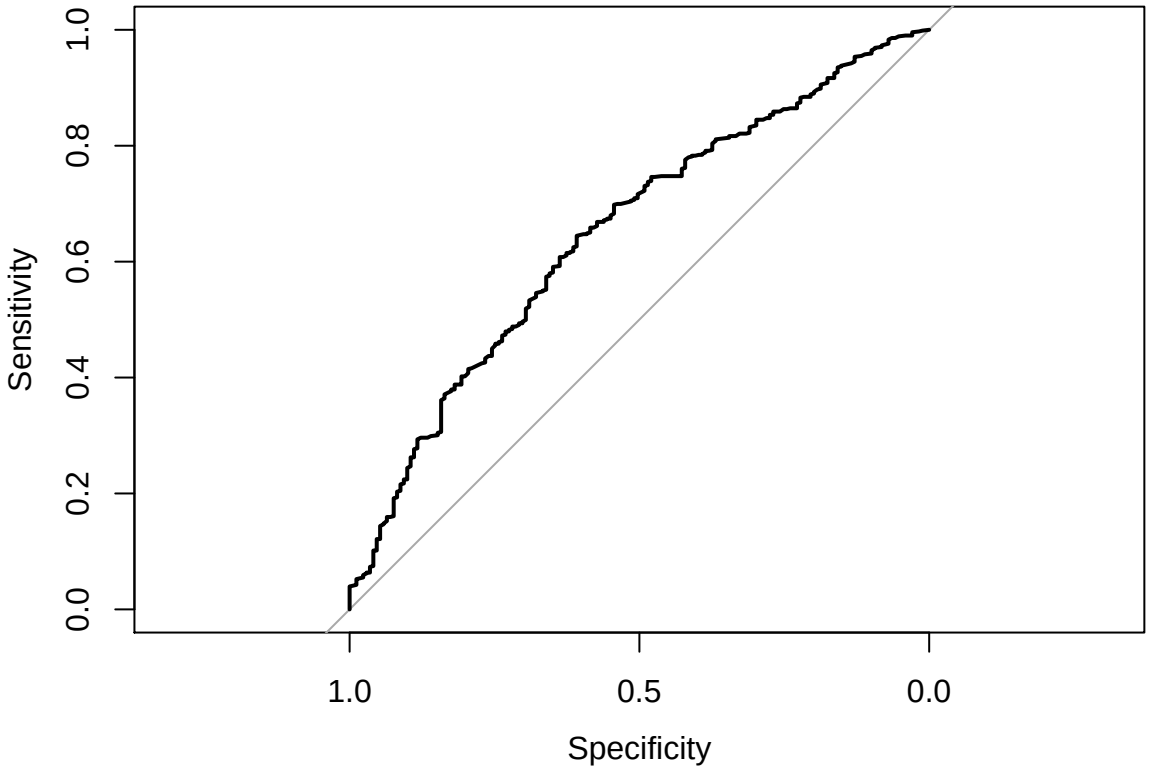
```
##         delayed    ontime
## [1,] 0.28027109 0.7197289
## [2,] 0.06651255 0.9334874
## [3,] 0.32475615 0.6752438
## [4,] 0.41049939 0.5895006
## [5,] 0.15360815 0.8463918
## [6,] 0.02795816 0.9720418
```

```
pred_class <- predict(nb_delays, newdata = valid.df)
```

```
library(caret)
# training
pred_class1 <- predict(nb_delays, newdata = train.df)
#confusionMatrix(pred_class1, train.df$Flight.Status)
# validation
pred_class2 <- predict(nb_delays, newdata = valid.df)
#confusionMatrix(pred_class2, valid.df$Flight.Status)
#AUC Value and ROC Curves III
library(pROC)
roc(valid.df$Flight.Status, nb_pred[,2])
```

```
##
## Call:
## roc.default(response = valid.df$Flight.Status, predictor = nb_pred[,     2])
##
## Data: nb_pred[, 2] in 171 controls (valid.df$Flight.Status delayed) < 709 cases (valid.df$Flight.Sta
## Area under the curve: 0.6503
```

```
plot.roc(valid.df$Flight.Status, nb_pred[,2])
```


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity