ML Assignment 3 - Naive Bayes Classification

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Loading the Data file in R

```r
FlightData<-read.csv("FlightDelays.csv")
str(FlightData)
```

```
## ’data.frame’:    2201 obs. of  13 variables:
##  $ CRS_DEP_TIME : int  1455 1640 1245 1715 1039 840 1240 1645 1715 2120 ...
##  $ CARRIER      : chr  "OH" "DH" "DH" "DH" ...
##  $ DEP_TIME     : int  1455 1640 1245 1709 1035 839 1243 1644 1710 2129 ...
##  $ DEST         : chr  "JFK" "JFK" "LGA" "LGA" ...
##  $ DISTANCE     : int  184 213 229 229 229 228 228 228 228 228 ...
##  $ FL_DATE      : chr  "01/01/2004" "01/01/2004" "01/01/2004" "01/01/2004" ...
##  $ FL_NUM       : int  5935 6155 7208 7215 7792 7800 7806 7810 7812 7814 ...
##  $ ORIGIN       : chr  "BWI" "DCA" "IAD" "IAD" ...
##  $ Weather      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ DAY_WEEK     : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ DAY_OF_MONTH : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TAIL_NUM     : chr  "N940CA" "N405FJ" "N695BR" "N662BR" ...
##  $ Flight.Status: chr  "ontime" "ontime" "ontime" "ontime" ...
```

```r
head(FlightData)
```

```
##   CRS_DEP_TIME CARRIER DEP_TIME DEST DISTANCE    FL_DATE FL_NUM ORIGIN Weather
## 1         1455      OH     1455  JFK      184 01/01/2004   5935    BWI       0
## 2         1640      DH     1640  JFK      213 01/01/2004   6155    DCA       0
## 3         1245      DH     1245  LGA      229 01/01/2004   7208    IAD       0
## 4         1715      DH     1709  LGA      229 01/01/2004   7215    IAD       0
## 5         1039      DH     1035  LGA      229 01/01/2004   7792    IAD       0
## 6          840      DH      839  JFK      228 01/01/2004   7800    IAD       0
##   DAY_WEEK DAY_OF_MONTH TAIL_NUM Flight.Status
## 1        4            1   N940CA        ontime
## 2        4            1   N405FJ        ontime
## 3        4            1   N695BR        ontime
## 4        4            1   N662BR        ontime
## 5        4            1   N698BR        ontime
## 6        4            1   N687BR        ontime
```

```r
#View(FlightData)
summary(FlightData)
```

```
##   CRS_DEP_TIME  CARRIER               DEP_TIME        DEST
##  Min.   : 600   Length:2201        Min.   :  10   Length:2201
##  1st Qu.:1000   Class :character   1st Qu.:1004   Class :character
##  Median :1455   Mode  :character   Median :1450   Mode  :character
##  Mean   :1372                      Mean   :1369
##  3rd Qu.:1710                      3rd Qu.:1709
##  Max.   :2130                      Max.   :2330
##     DISTANCE       FL_DATE              FL_NUM        ORIGIN
##  Min.   :169.0   Length:2201        Min.   : 746   Length:2201
##  1st Qu.:213.0   Class :character   1st Qu.:2156   Class :character
##  Median :214.0   Mode  :character   Median :2385   Mode  :character
##  Mean   :211.9                      Mean   :3815
##  3rd Qu.:214.0                      3rd Qu.:6155
##  Max.   :229.0                      Max.   :7924
##     Weather           DAY_WEEK      DAY_OF_MONTH     TAIL_NUM
##  Min.   :0.00000   Min.   :1.000   Min.   : 1.00   Length:2201
##  1st Qu.:0.00000   1st Qu.:2.000   1st Qu.: 8.00   Class :character
##  Median :0.00000   Median :4.000   Median :16.00   Mode  :character
##  Mean   :0.01454   Mean   :3.905   Mean   :16.02
##  3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:23.00
##  Max.   :1.00000   Max.   :7.000   Max.   :31.00
##  Flight.Status
##  Length:2201
##  Class :character
##  Mode  :character
##
##
##
```

Library for Naive Bayes theorem

```
library(caret)
```

```
## Loading required package: lattice
```

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

```
library(ISLR)
# install.packages("e1071") #install first
library(e1071)
```

Clean the Data

```
FlightData<-FlightData[,c(-3,-5,-6,-7,-9,-11,-12)]
str(FlightData)
```

```
## ’data.frame':    2201 obs. of  6 variables:
##  $ CRS_DEP_TIME : int  1455 1640 1245 1715 1039 840 1240 1645 1715 2120 ...
##  $ CARRIER      : chr  "OH" "DH" "DH" "DH" ...
##  $ DEST         : chr  "JFK" "JFK" "LGA" "LGA" ...
##  $ ORIGIN       : chr  "BWI" "DCA" "IAD" "IAD" ...
##  $ DAY_WEEK     : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Flight.Status: chr  "ontime" "ontime" "ontime" "ontime" ...
```

```
head(FlightData)
```

```
##   CRS_DEP_TIME CARRIER DEST ORIGIN DAY_WEEK Flight.Status
## 1         1455      OH  JFK    BWI        4        ontime
## 2         1640      DH  JFK    DCA        4        ontime
## 3         1245      DH  LGA    IAD        4        ontime
## 4          1715       DH   LGA     IAD         4         ontime
## 5          1039       DH   LGA     IAD         4         ontime
## 6           840       DH   JFK     IAD         4         ontime
```

Change the numerical variables to factors

```
FlightData$DAY_WEEK<-as.factor(FlightData$DAY_WEEK)
levels(FlightData$DAY_WEEK)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7"
```

```
#creating hourly bins for the departure time
FlightData$CRS_DEP_TIME<-as.factor(FlightData$CRS_DEP_TIME)
levels(FlightData$CRS_DEP_TIME)
```

```
##  [1] "600"  "630"  "640"  "645"  "700"  "730"  "735"  "759"  "800"  "830"
## [11] "840"  "845"  "850"  "900"  "925"  "930"  "1000" "1030" "1039" "1040"
## [21] "1100" "1130" "1200" "1230" "1240" "1245" "1300" "1315" "1330" "1359"
## [31] "1400" "1430" "1455" "1500" "1515" "1520" "1525" "1530" "1600" "1605"
## [41] "1610" "1630" "1640" "1645" "1700" "1710" "1715" "1720" "1725" "1730"
## [51] "1800" "1830" "1900" "1930" "2000" "2030" "2100" "2120" "2130"
```

```
#Outcome variable #Flight.Status
FlightData$Flight.Status<- factor(FlightData$Flight.Status, levels = c("delayed", "ontime"), labels = c

str(FlightData)
```

```
## 'data.frame':    2201 obs. of  6 variables:
##  $ CRS_DEP_TIME : Factor w/ 59 levels "600","630","640",..: 33 43 26 47 19 11 25 44 47 58 ...
##  $ CARRIER      : chr  "OH" "DH" "DH" "DH" ...
##  $ DEST         : chr  "JFK" "JFK" "LGA" "LGA" ...
##  $ ORIGIN       : chr  "BWI" "DCA" "IAD" "IAD" ...
##  $ DAY_WEEK     : Factor w/ 7 levels "1","2","3","4",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Flight.Status: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
#View(FlightData)
```

Divide into training and test

```
set.seed(123)
Index_train<-createDataPartition(FlightData$Flight.Status, p=0.6, list = FALSE)

#Training Data
TrainData<-FlightData[Index_train,]
#Test Data
TestData<-FlightData[-Index_train,]

#Data validations at the Training and Test data set
summary(TrainData)
```

```
##   CRS_DEP_TIME   CARRIER               DEST               ORIGIN
##  1455   : 82   Length:1321        Length:1321        Length:1321
##  1300   : 64   Class :character   Class :character   Class :character
##  2120   : 58   Mode  :character   Mode  :character   Mode  :character
##  700    : 57
##  1900   : 55
##  900    : 52
##  (Other):953
##  DAY_WEEK Flight.Status
##  1:202    0: 257
##  2:176    1:1064
##  3:172
##  4:232
##  5:241
##  6:145
##  7:153
```

```r
summary(TestData)
```

```
##   CRS_DEP_TIME   CARRIER              DEST               ORIGIN
##  1455   : 56   Length:880         Length:880         Length:880
##  1300   : 45   Class :character   Class :character   Class :character
##  1900   : 44   Mode  :character   Mode  :character   Mode  :character
##  1700   : 36
##  700    : 35
##  2120   : 32
##  (Other):632
##  DAY_WEEK Flight.Status
##  1:106    0:171
##  2:131    1:709
##  3:148
##  4:140
##  5:150
##  6:105
##  7:100
```

Run Naive Bayes

```r
nb_model<-naiveBayes(TrainData$Flight.Status~., data=TrainData)
nb_model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.1945496 0.8054504
##
## Conditional probabilities:
##    CRS_DEP_TIME
## Y            600          630          640          645          700
##   0 0.0000000000 0.0077821012 0.0038910506 0.0000000000 0.0466926070
##   1 0.0140977444 0.0291353383 0.0084586466 0.0112781955 0.0422932331
##    CRS_DEP_TIME
## Y            730          735          759          800          830
##   0 0.0077821012 0.0077821012 0.0000000000 0.0077821012 0.0077821012
##   1 0.0103383459 0.0084586466 0.0018796992 0.0178571429 0.0140977444
##    CRS_DEP_TIME
## Y            840          845          850          900          925
##   0 0.0155642023 0.0000000000 0.0116731518 0.0194552529 0.0000000000
##   1 0.0366541353 0.0018796992 0.0150375940 0.0441729323 0.0018796992
##    CRS_DEP_TIME
## Y            930         1000         1030         1039         1040
##   0 0.0000000000 0.0000000000 0.0233463035 0.0038910506 0.0038910506
##   1 0.0140977444 0.0159774436 0.0281954887 0.0018796992 0.0084586466
##    CRS_DEP_TIME
## Y           1100         1130         1200         1230         1240
##   0 0.0077821012 0.0000000000 0.0000000000 0.0000000000 0.0194552529
##   1 0.0263157895 0.0131578947 0.0093984962 0.0140977444 0.0150375940
##    CRS_DEP_TIME
## Y           1245         1300         1315         1330         1359
##   0 0.0505836576 0.0350194553 0.0038910506 0.0000000000 0.0116731518
##   1 0.0234962406 0.0516917293 0.0000000000 0.0122180451 0.0103383459
##    CRS_DEP_TIME
## Y           1400         1430         1455         1500         1515
##   0 0.0077821012 0.0272373541 0.1050583658 0.0350194553 0.0038910506
##   1 0.0234962406 0.0187969925 0.0516917293 0.0347744361 0.0018796992
##    CRS_DEP_TIME
## Y           1520         1525         1530         1600         1605
##   0 0.0000000000 0.0272373541 0.0233463035 0.0350194553 0.0000000000
##   1 0.0009398496 0.0084586466 0.0225563910 0.0178571429 0.0000000000
##    CRS_DEP_TIME
## Y           1610         1630         1640         1645         1700
##   0 0.0116731518 0.0155642023 0.0155642023 0.0038910506 0.0272373541
##   1 0.0103383459 0.0187969925 0.0131578947 0.0169172932 0.0291353383
##    CRS_DEP_TIME
## Y           1710         1715         1720         1725         1730
##   0 0.0194552529 0.0389105058 0.0233463035 0.0000000000 0.0350194553
##   1 0.0103383459 0.0244360902 0.0093984962 0.0009398496 0.0216165414
##    CRS_DEP_TIME
## Y           1800         1830         1900         1930         2000
##   0 0.0038910506 0.0389105058 0.0894941634 0.0077821012 0.0077821012
##   1 0.0122180451 0.0253759398 0.0300751880 0.0112781955 0.0112781955
##    CRS_DEP_TIME
## Y           2030         2100         2120         2130
##   0 0.0116731518 0.0155642023 0.0700389105 0.0038910506
##   1 0.0140977444 0.0206766917 0.0375939850 0.0000000000
##
##    CARRIER
## Y            CO          DH          DL          MQ          OH          RU
##   0 0.066147860 0.322957198 0.112840467 0.178988327 0.007782101 0.206225681
##   1 0.037593985 0.240601504 0.186090226 0.124060150 0.013157895 0.178571429
##    CARRIER
## Y            UA          US
##   0 0.011673152 0.093385214
##   1 0.015037594 0.204887218
##
##    DEST
## Y         EWR       JFK       LGA
##   0 0.3891051 0.2217899 0.3891051
##   1 0.2819549 0.1823308 0.5357143
##
##    ORIGIN
## Y          BWI        DCA        IAD
##   0 0.07392996 0.51361868 0.41245136
##   1 0.06109023 0.64849624 0.29041353
##
##    DAY_WEEK
## Y            1          2          3          4          5          6
##   0 0.18677043 0.15953307 0.11284047 0.15175097 0.17509728 0.05447471
##   1 0.14473684 0.12687970 0.13439850 0.18139098 0.18421053 0.12312030
##    DAY_WEEK
## Y            7
##   0 0.15953307
##   1 0.10526316
```

Pivot table for Flight status by destination

```
pr<-prop.table(table(TrainData$Flight.Status, TrainData$DEST), margin = 1)
pr
```

```
##
##           EWR       JFK       LGA
##   0 0.3891051 0.2217899 0.3891051
##   1 0.2819549 0.1823308 0.5357143
```

Using the model on Test set

```
# Predict probabilities Test Data

PredProb <- predict(nb_model, TestData)
head(PredProb)
```

```
## [1] 1 1 1 1 1 1
## Levels: 0 1
```

```
#Confusion Matrix on the Test Data

library("gmodels")
CrossTable(x=TestData$Flight.Status, y=PredProb, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  880
##
##
##                        | PredProb
## TestData$Flight.Status |         0 |         1 | Row Total |
## -----------------------|-----------|-----------|-----------|
##                      0 |        33 |       138 |       171 |
##                        |     0.193 |     0.807 |     0.194 |
##                        |     0.393 |     0.173 |           |
##                         |      0.037 |      0.157 |            |
## ------------------------|------------|------------|------------|
##                       1 |         51 |        658 |        709 |
##                         |      0.072 |      0.928 |      0.806 |
##                         |      0.607 |      0.827 |            |
##                         |      0.058 |      0.748 |            |
## ------------------------|------------|------------|------------|
##            Column Total |         84 |        796 |        880 |
##                         |      0.095 |      0.905 |            |
## ------------------------|------------|------------|------------|
##
##
```

```r
#Predecting probability of each class
PredProb<-predict(nb_model, TestData, type = "raw")
head(PredProb)
```

```
##                0         1
## [1,] 0.375920081 0.6240799
## [2,] 0.366764468 0.6332355
## [3,] 0.377430946 0.6225691
## [4,] 0.004975078 0.9950249
## [5,] 0.092673535 0.9073265
## [6,] 0.068785526 0.9312145
```

Plot ROC curve for Test Data Set

```r
library("pROC")
```

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

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

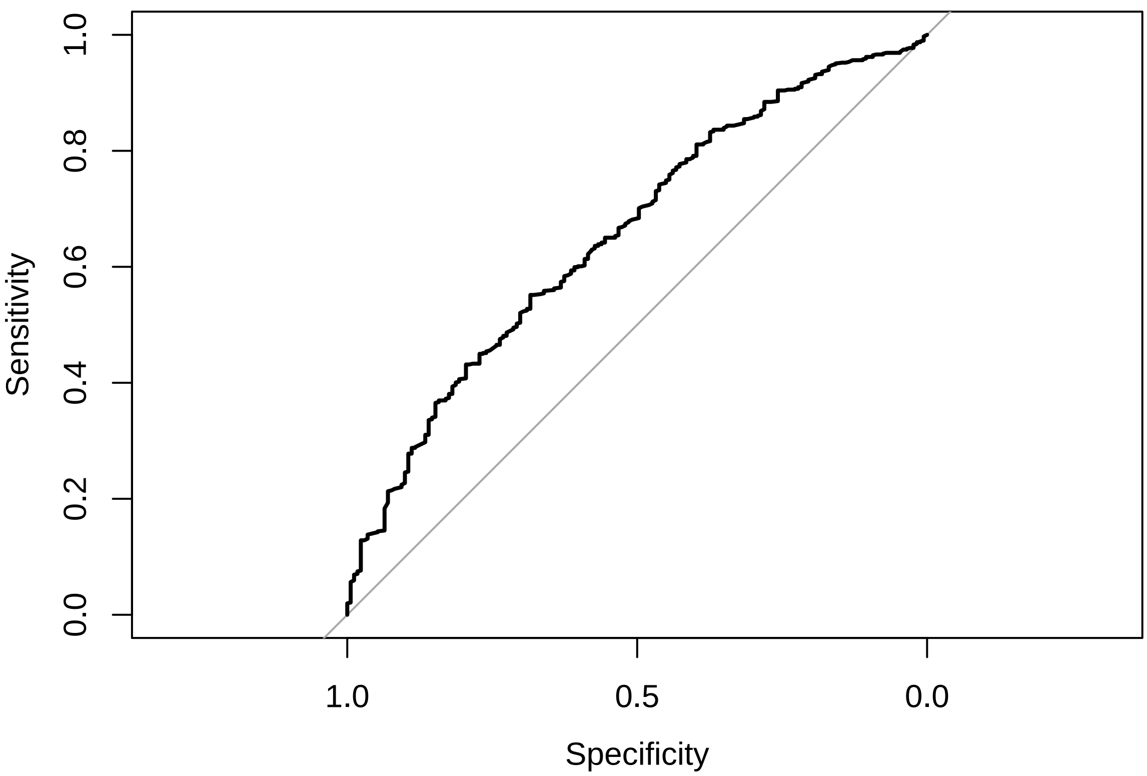
```
## The following object is masked from 'package:gmodels':
##
##     ci
```

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

```r
plot.roc(TestData$Flight.Status, PredProb[,2])
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



Output both a counts table and a proportion table outlining how many and what proportion table outlining how many and what proportion of flights were delayed and on-time at each of the three airports.

```
#Counts Table
table(FlightData$Flight.Status, FlightData$DEST)
```

```
##
##     EWR JFK LGA
##   0 161  84 183
##   1 504 302 967
```

```
#Proportion Table
prop.table(table(FlightData$Flight.Status, FlightData$DEST))
```

```
##
##            EWR        JFK        LGA
##   0 0.07314857 0.03816447 0.08314403
##   1 0.22898682 0.13721036 0.43934575
```