Predicting Delayed Flights

```
library(readr)
library(caret)
```

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

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

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

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(pROC)
```

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

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

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

Read the data and Summary

```
MyData <- read.csv("FlightDelays.csv")
summary(MyData)
```

```
##   CRS_DEP_TIME     CARRIER       DEP_TIME     DEST         DISTANCE
##  Min.   : 600   DH     :551   Min.   :  10   EWR: 665   Min.   :169.0
##  1st Qu.:1000   RU     :408   1st Qu.:1004   JFK: 386   1st Qu.:213.0
##  Median :1455   US     :404   Median :1450   LGA:1150   Median :214.0
##  Mean   :1372   DL     :388   Mean   :1369              Mean   :211.9
##  3rd Qu.:1710   MQ     :295   3rd Qu.:1709              3rd Qu.:214.0
##  Max.   :2130   CO     : 94   Max.   :2330              Max.   :229.0
##                 (Other): 61
##       FL_DATE         FL_NUM     ORIGIN        Weather
##  1/22/2004 :  86   Min.   : 746   BWI: 145   Min.   :0.00000
##  01/06/2004:  85   1st Qu.:2156   DCA:1370   1st Qu.:0.00000
##  01/08/2004:  85   Median :2385   IAD: 686   Median :0.00000
##  1/13/2004 :  85   Mean   :3815              Mean   :0.01454
##  1/20/2004 :  85   3rd Qu.:6155              3rd Qu.:0.00000
##  1/21/2004 :  85   Max.   :7924              Max.   :1.00000
##  (Other)   :1690
##     DAY_WEEK      DAY_OF_MONTH      TAIL_NUM    Flight.Status
##  Min.   :1.000   Min.   : 1.00   N225DL :  65   delayed: 428
##  1st Qu.:2.000   1st Qu.: 8.00   N242DL :  56   ontime :1773
##  Median :4.000   Median :16.00   N223DZ :  50
##  Mean   :3.905   Mean   :16.02   N221DL :  45
##  3rd Qu.:5.000   3rd Qu.:23.00   N241DL :  36
##  Max.   :7.000   Max.   :31.00   N722UW :  36
##                                  (Other):1913
```

Clean the data

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

```
## 'data.frame':    2201 obs. of  6 variables:
##  $ CRS_DEP_TIME : int  1455 1640 1245 1715 1039 840 1240 1645 1715 2120 ...
##  $ CARRIER      : Factor w/ 8 levels "CO","DH","DL",..: 5 2 2 2 2 2 2 2 2 2 ...
##  $ DEST         : Factor w/ 3 levels "EWR","JFK","LGA": 2 2 3 3 3 2 2 2 2 2 ...
##  $ ORIGIN       : Factor w/ 3 levels "BWI","DCA","IAD": 1 2 3 3 3 3 3 3 3 3 ...
##  $ DAY_WEEK     : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Flight.Status: Factor w/ 2 levels "delayed","ontime": 2 2 2 2 2 2 2 2 2 2 ...
```

```
head(MyData)
```

	CRS_DEP_TIME <int>	CARRIER <fctr>	DEST <fctr>	ORIGIN <fctr>	DAY_WEEK <int>	Flight.Status <fctr>
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

	CRS_DEP_TIME <int>	CARRIER <fctr>	DEST <fctr>	ORIGIN <fctr>	DAY_WEEK <int>	Flight.Status <fctr>
5	1039	DH	LGA	IAD	4	ontime
6	840	DH	JFK	IAD	4	ontime

6 rows

```
set.seed(123)
```

Week and Time Variables to be recorded as Factors

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

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

```
MyData$CRS_DEP_TIME <- as.factor(MyData$CRS_DEP_TIME)
levels(MyData$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"
```

The outcome variable is whether the flight was delayed, and thus it has two classes (1 = delayed and 0 = on time)

```
MyData$Flight.Status <- factor(MyData$Flight.Status,levels = c("delayed","ontime"),labels = c(0,
1))
```

Divide the data into training and Validation

```
# 60% reserved for Training
Train_Index <- createDataPartition(MyData$Flight.Status, p=0.6, list=FALSE)
Training <- MyData[Train_Index,]
# Validation is the rest 40%
Valid_Data  <- MyData[-Train_Index,]
```

Run the Naive Bayes model to predict whether the flight is delayed or not. Use only categorical variables for the predictor variables.

```
nb_model <-naiveBayes(Training$Flight.Status~CARRIER+DEST+ORIGIN+DAY_WEEK+CRS_DEP_TIME, data = T
raining)
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:
##    CARRIER
## Y            CO          DH          DL          MQ          OH
##   0 0.066147860 0.322957198 0.112840467 0.178988327 0.007782101
##   1 0.037593985 0.240601504 0.186090226 0.124060150 0.013157895
##    CARRIER
## Y            RU          UA          US
##   0 0.206225681 0.011673152 0.093385214
##   1 0.178571429 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
##
##    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
```

Output the confusion matrix and ROC for the validation data

```r
# Predicting the delayed status on Validation dataSet
Predicted_Valid_labels <-predict(nb_model,Valid_Data)
library("gmodels")
```

```
##
## Attaching package: 'gmodels'
```

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

```
# Show the confusion matrix of the classifier
CrossTable(x=Valid_Data$Flight.Status,y=Predicted_Valid_labels, prop.chisq = FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  880 
## 
## 
##                         | Predicted_Valid_labels 
## Valid_Data$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 |           | 
## -------------------------|-----------|-----------|-----------|
## 
## 
```

```
nb_model <- naiveBayes(Training$Flight.Status~CARRIER+DEST+ORIGIN+DAY_WEEK+CRS_DEP_TIME,data = T
raining)
#Make predictions and return probability of each class
Predicted_Valid_labels <-predict(nb_model,Valid_Data, type = "raw")
#show the first few values
head(Predicted_Valid_labels)
```

```
##                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
```

ROC Curve for Validation Data Set

```
roc(Valid_Data$Flight.Status, Predicted_Valid_labels[,2])
```

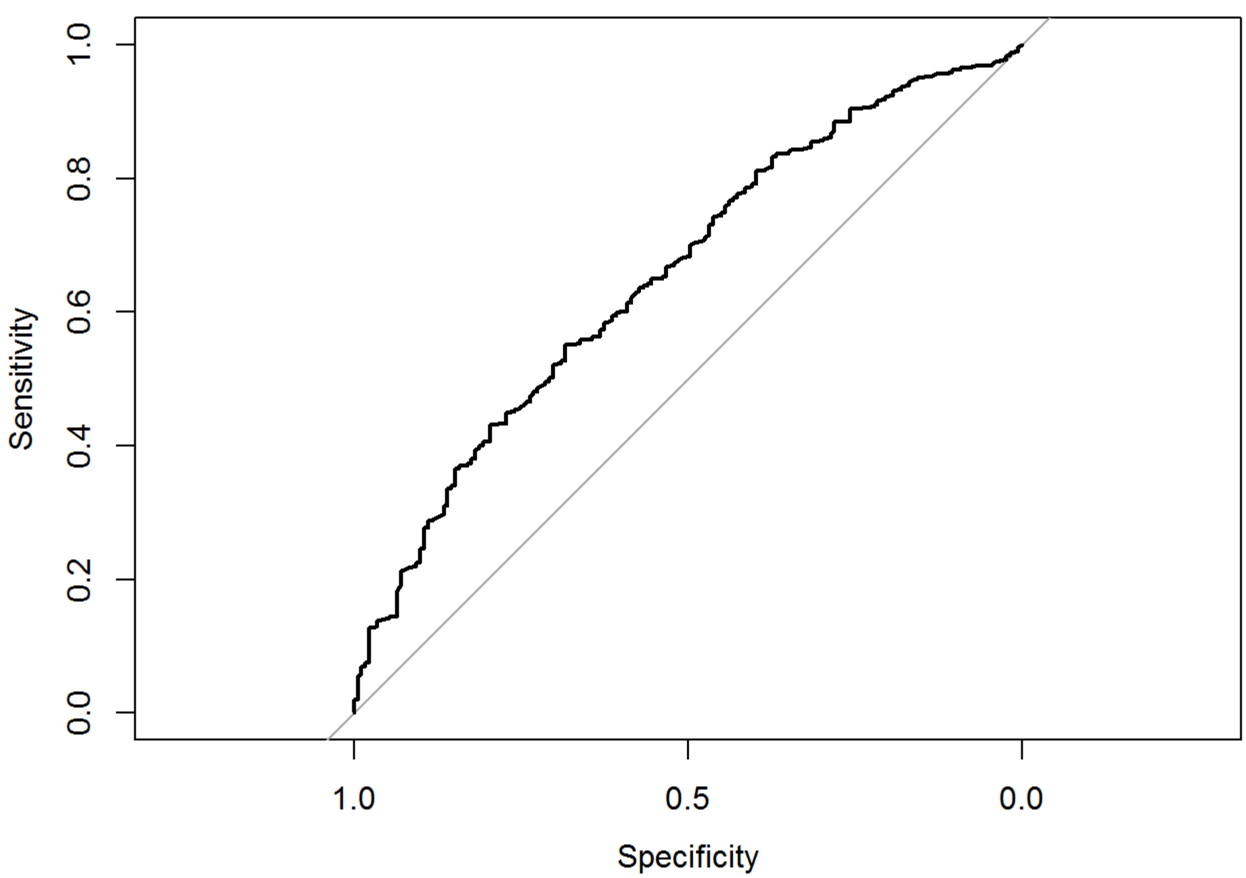
```
## Setting levels: control = 0, case = 1
```

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

```
##
## Call:
## roc.default(response = Valid_Data$Flight.Status, predictor = Predicted_Valid_labels[,     2])
##
## Data: Predicted_Valid_labels[, 2] in 171 controls (Valid_Data$Flight.Status 0) < 709 cases (V
alid_Data$Flight.Status 1).
## Area under the curve: 0.6553
```

```
plot.roc(Valid_Data$Flight.Status,Predicted_Valid_labels[,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 of flights were delayed and on-time at each of the three airports.

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

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

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

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