MLAssignment3

#Install package if not already installed

```
#install.packages("caret")
#install.packages("lattice")
#install.packages("ggplot2")
#install.packages("e1071")
#install.packages("klaR")
#install.packages("psych")
#install.packages("rpivotTable")
```

#load all the required libraries

```
library(caret)
```

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

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

```
library(readr)
library(gmodels)
library(ISLR)
library(e1071)
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 object is masked from 'package:gmodels':
##
##     ci
```

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

```
library(rpivotTable)
```

Import the FlightDelays Dataset

```
FlightData <- read.csv("FlightDelays.csv")
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
##
##
##
```

Record Week columns as fators

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

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

Create bins for Scheduled Departure Time column and record it as factors

```
FlightData$CRS_DEP_TIME <- floor(FlightData$CRS_DEP_TIME/100)
FlightData$CRS_DEP_TIME <- factor(FlightData$CRS_DEP_TIME)
levels(FlightData$CRS_DEP_TIME)
```

```
##  [1] "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15" "16" "17" "18" "19" "20"
## [16] "21"
```

Label the outcome variable in two classes (1 = Delayed, 0 = Ontime)

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

```
## [1] "0" "1"
```

Partition the data into training (60%) and validation (40%) sets

```
set.seed(123)
Train_Index=createDataPartition(FlightData$CRS_DEP_TIME, p=0.60, list=FALSE)
Train_Data = FlightData[Train_Index,]
Validation_Data = FlightData[-Train_Index,]
```

Now, run the Naive Bayes model on the training dataset

```
nb_model <- naiveBayes(Train_Data$Flight.Status~CARRIER+DEST+ORIGIN+DAY_WEEK+CRS_DEP_TIME, data = Train_Data)
nb_model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.7980407 0.2019593
##
## Conditional probabilities:
##    CARRIER
## Y            CO          DH          DL          MQ          OH          RU
##   0 0.035882908 0.241737488 0.180358829 0.124645892 0.012275732 0.169971671
##   1 0.063432836 0.291044776 0.138059701 0.205223881 0.007462687 0.212686567
##    CARRIER
## Y            UA          US
##   0 0.012275732 0.222851747
##   1 0.011194030 0.070895522
##
##    DEST
## Y         EWR       JFK       LGA
##   0 0.2766761 0.1661945 0.5571294
##   1 0.3656716 0.1977612 0.4365672
##
##    ORIGIN
## Y          BWI        DCA        IAD
##   0 0.05382436 0.64683664 0.29933900
##   1 0.07835821 0.53731343 0.38432836
##
##    DAY_WEEK
## Y            1          2          3          4          5          6
##   0 0.11709160 0.13786591 0.15391879 0.17469311 0.18508026 0.12181303
##   1 0.21641791 0.17537313 0.13432836 0.11567164 0.17537313 0.04477612
##    DAY_WEEK
## Y            7
##   0 0.10953730
##   1 0.13805970
##
##    CRS_DEP_TIME
## Y            6          7          8          9         10         11
##   0 0.06232295 0.06232295 0.07648725 0.05571294 0.04910293 0.03493862
##   1 0.03731343 0.05597015 0.06343284 0.02238806 0.02985075 0.01492537
##    CRS_DEP_TIME
## Y           12         13         14         15         16         17
##   0 0.07082153 0.07837583 0.09442871 0.06326723 0.08120869 0.10576015
##   1 0.04104478 0.04477612 0.15671642 0.09701493 0.07835821 0.12313433
##    CRS_DEP_TIME
## Y           18         19         20         21
##   0 0.03871577 0.04249292 0.02455146 0.05949008
##   1 0.03731343 0.10074627 0.02238806 0.07462687
```

The output from the Naive Bayes model predicts that the probability of Delayed flight is 0.2019593 and the probability for On time flights is 0.7980407

Output Counts table and proportion table for flights that are delayed and ontime from Origin

```
#Counts table
rpivotTable(FlightData, rows = "Flight.Status", cols = "ORIGIN", width = "100%", height = "400px")
```

Table | Count | ORIGIN

CRS_DEP_TIME, CARRIER, DEP_TIME, DEST, DISTANCE, FL_DATE, FL_NUM, Weather, DAY_WEEK, DAY_OF_MONTH, TAIL_NUM

Flight.Status

Flight.Status \ ORIGIN	BWI	DCA	IAD	Totals
0	108	1,149	516	1,773
1	37	221	170	428
Totals	145	1,370	686	2,201

```
#Proportion Table
prop.table(table(FlightData$Flight.Status, FlightData$ORIGIN),margin = 1)
```

```
##
##            BWI        DCA        IAD
##   0 0.06091371 0.64805415 0.29103215
##   1 0.08644860 0.51635514 0.39719626
```

Output Counts table and proportion table for ontime and delayed flights at Destination

```
#Counts table
rpivotTable(FlightData, rows = "Flight.Status", cols = "DEST", width = "100%", height = "400px")
```

Table | Count | DEST

CRS_DEP_TIME, CARRIER, DEP_TIME, DISTANCE, FL_DATE, FL_NUM, ORIGIN, Weather, DAY_WEEK, DAY_OF_MONTH, TAIL_NUM

Flight.Status

Flight.Status \ DEST	EWR	JFK	LGA	Totals
0	504	302	967	1,773
1	161	84	183	428
Totals	665	386	1,150	2,201

```
#Proportion Table
prop.table(table(FlightData$Flight.Status, FlightData$DEST),margin = 1)
```

```
##
##           EWR       JFK       LGA
##   0 0.2842640 0.1703328 0.5454033
##   1 0.3761682 0.1962617 0.4275701
```

Confusion matrix for the validation data

```
#Make predictions and return probability
PredictData <-predict(nb_model,Validation_Data)

#show the first few values
head(PredictData)
```

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

```
#Confusion Matrix
confusionMatrix(PredictData, Validation_Data$Flight.Status)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 691 144
##          1  23  16
##
##                Accuracy : 0.8089
##                  95% CI : (0.7813, 0.8345)
##     No Information Rate : 0.8169
##     P-Value [Acc > NIR] : 0.7458
##
##                   Kappa : 0.0959
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9678
##             Specificity : 0.1000
##          Pos Pred Value : 0.8275
##          Neg Pred Value : 0.4103
##              Prevalence : 0.8169
##          Detection Rate : 0.7906
##    Detection Prevalence : 0.9554
##       Balanced Accuracy : 0.5339
##
##        'Positive' Class : 0
##
```

The Confusion matrix for the validation data set shows the Accuracy of 0.8089

Output the ROC for the validation data

```
#Make predictions and return probability of each class
PredictData <-predict(nb_model,Validation_Data, type = "raw")

#show the first few values
head(PredictData)
```

```
##              0          1
## [1,] 0.7734716 0.22652837
## [2,] 0.8947474 0.10525262
## [3,] 0.8088533 0.19114668
## [4,] 0.8901611 0.10983894
## [5,] 0.7970548 0.20294517
## [6,] 0.9675379 0.03246205
```

```
#ROC Curve for validation Data set
roc(Validation_Data$Flight.Status, PredictData[,2])
```

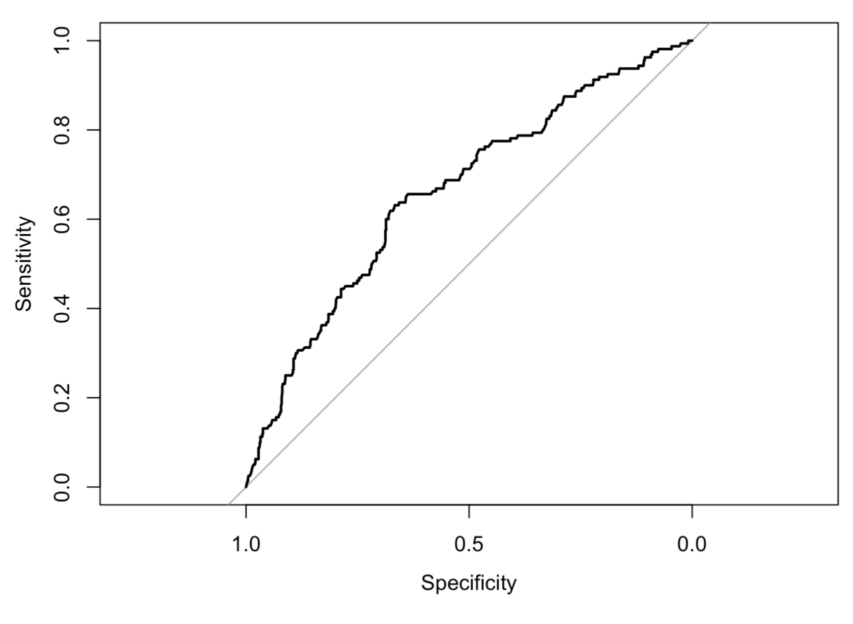
```
## Setting levels: control = 0, case = 1
```

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

```
##
## Call:
## roc.default(response = Validation_Data$Flight.Status, predictor = PredictData[,     2])
##
## Data: PredictData[, 2] in 714 controls (Validation_Data$Flight.Status 0) < 160 cases (Validation_Data$Flight.S
tatus 1).
## Area under the curve: 0.661
```

```
plot.roc(Validation_Data$Flight.Status,PredictData[,2])
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



The Area Under the Curve is 0.661