R Notebook

Reading file into program

```
library(readr)
FlightDelays <- read_csv("C:/Trading detail/STUDY/01_MSBA/02 MSBA ML/02/FlightDelays.csv")
```

```
## Parsed with column specification:
## cols(
##   CRS_DEP_TIME = col_double(),
##   CARRIER = col_character(),
##   DEP_TIME = col_double(),
##   DEST = col_character(),
##   DISTANCE = col_double(),
##   FL_DATE = col_character(),
##   FL_NUM = col_double(),
##   ORIGIN = col_character(),
##   Weather = col_double(),
##   DAY_WEEK = col_double(),
##   DAY_OF_MONTH = col_double(),
##   TAIL_NUM = col_character(),
##   `Flight Status` = col_character()
## )
```

```
summary(FlightDelays)
```

```
##   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(caret)
```

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

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

```
library(ISLR)
library(e1071)
str(FlightDelays)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2201 obs. of  13 variables:
##  $ CRS_DEP_TIME : num  1455 1640 1245 1715 1039 ...
##  $ CARRIER      : chr  "OH" "DH" "DH" "DH" ...
##  $ DEP_TIME     : num  1455 1640 1245 1709 1035 ...
##  $ DEST         : chr  "JFK" "JFK" "LGA" "LGA" ...
##  $ DISTANCE     : num  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       : num  5935 6155 7208 7215 7792 ...
##  $ ORIGIN       : chr  "BWI" "DCA" "IAD" "IAD" ...
##  $ Weather      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DAY_WEEK     : num  4 4 4 4 4 4 4 4 4 4 ...
##  $ DAY_OF_MONTH : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ TAIL_NUM     : chr  "N940CA" "N405FJ" "N695BR" "N662BR" ...
##  $ Flight Status: chr  "ontime" "ontime" "ontime" "ontime" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   CRS_DEP_TIME = col_double(),
##   ..   CARRIER = col_character(),
##   ..   DEP_TIME = col_double(),
##   ..   DEST = col_character(),
##   ..   DISTANCE = col_double(),
##   ..   FL_DATE = col_character(),
##   ..   FL_NUM = col_double(),
##   ..   ORIGIN = col_character(),
##   ..   Weather = col_double(),
##   ..   DAY_WEEK = col_double(),
##   ..   DAY_OF_MONTH = col_double(),
##   ..   TAIL_NUM = col_character(),
##   ..   `Flight Status` = col_character()
##   .. )
```

2. categorical variables for the predictor variables -> Week and Time variables need to recoded as factors

```
FlightDelays$DAY_WEEK<-as.factor(FlightDelays$DAY_WEEK)
FlightDelays$CRS_DEP_TIME<-as.factor(FlightDelays$CRS_DEP_TIME)
FlightDelays$`Flight Status` <- as.factor(FlightDelays$`Flight Status`)

##FlightDelays$`Flight Status` <- factor(FlightDelays$`Flight Status`,levels = c("delayed","ontime"),labels = c(0,1))
str(FlightDelays)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2201 obs. of  13 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" ...
##  $ DEP_TIME     : num  1455 1640 1245 1709 1035 ...
##  $ DEST         : chr  "JFK" "JFK" "LGA" "LGA" ...
##  $ DISTANCE     : num  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       : num  5935 6155 7208 7215 7792 ...
##  $ ORIGIN       : chr  "BWI" "DCA" "IAD" "IAD" ...
##  $ Weather      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DAY_WEEK     : Factor w/ 7 levels "1","2","3","4",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ DAY_OF_MONTH : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ TAIL_NUM     : chr  "N940CA" "N405FJ" "N695BR" "N662BR" ...
##  $ Flight Status: Factor w/ 2 levels "delayed","ontime": 2 2 2 2 2 2 2 2 2 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   CRS_DEP_TIME = col_double(),
##   ..   CARRIER = col_character(),
##   ..   DEP_TIME = col_double(),
##   ..   DEST = col_character(),
##   ..   DISTANCE = col_double(),
##   ..   FL_DATE = col_character(),
##   ..   FL_NUM = col_double(),
##   ..   ORIGIN = col_character(),
##   ..   Weather = col_double(),
##   ..   DAY_WEEK = col_double(),
##   ..   DAY_OF_MONTH = col_double(),
##   ..   TAIL_NUM = col_character(),
##   ..   `Flight Status` = col_character()
##   .. )
```

filter data as per required coloumn

```
MyData<-FlightDelays[,c(10,1,8,4,2,13)]
summary(MyData)
```

```
##  DAY_WEEK  CRS_DEP_TIME     ORIGIN              DEST          
##  1:308    1455   : 138   Length:2201        Length:2201       
##  2:307    1300   : 109   Class :character   Class :character  
##  3:320    1900   :  99   Mode  :character   Mode  :character  
##  4:372    700    :  92                                        
##  5:391    2120   :  90                                        
##  6:250    900    :  77                                        
##  7:253    (Other):1596                                        
##    CARRIER          Flight Status
##  Length:2201        delayed: 428 
##  Class :character   ontime :1773 
##  Mode  :character                
##                                  
##                                  
##                                  
## 
```

```
str(MyData)
```

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

```
str(FlightDelays)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2201 obs. of  13 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" ...
##  $ DEP_TIME     : num  1455 1640 1245 1709 1035 ...
##  $ DEST         : chr  "JFK" "JFK" "LGA" "LGA" ...
##  $ DISTANCE     : num  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       : num  5935 6155 7208 7215 7792 ...
##  $ ORIGIN       : chr  "BWI" "DCA" "IAD" "IAD" ...
##  $ Weather      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ DAY_WEEK     : Factor w/ 7 levels "1","2","3","4",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ DAY_OF_MONTH : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ TAIL_NUM     : chr  "N940CA" "N405FJ" "N695BR" "N662BR" ...
##  $ Flight Status: Factor w/ 2 levels "delayed","ontime": 2 2 2 2 2 2 2 2 2 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   CRS_DEP_TIME = col_double(),
##   ..   CARRIER = col_character(),
##   ..   DEP_TIME = col_double(),
##   ..   DEST = col_character(),
##   ..   DISTANCE = col_double(),
##   ..   FL_DATE = col_character(),
##   ..   FL_NUM = col_double(),
##   ..   ORIGIN = col_character(),
##   ..   Weather = col_double(),
##   ..   DAY_WEEK = col_double(),
##   ..   DAY_OF_MONTH = col_double(),
##   ..   TAIL_NUM = col_character(),
##   ..   `Flight Status` = col_character()
##   .. )
```

Divide the data into 60% training and 40% validation apart form training and validation data dividing subset data into test , travel for further check of accuracy of the model

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

Train <-MyData[Index_Train,]  ## 60 % data
valid  <-MyData[-Index_Train,]  ## 40 %data
test_index <- createDataPartition(MyData$`Flight Status`, p=0.2, list=FALSE)
testdata <- MyData[test_index,]
traveldata <- MyData[-test_index,]
```

Prepareing NaiveBayes model

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

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

Now, use the model on the test set

```
Predicted_Test_labels <-predict(nb_model,testdata)

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

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  441 
## 
## 
##                          | Predicted_Test_labels 
## testdata$`Flight Status` |   delayed |    ontime | Row Total | 
## -------------------------|-----------|-----------|-----------|
##                  delayed |        22 |        64 |        86 | 
##                          |     0.256 |     0.744 |     0.195 | 
##                          |     0.500 |     0.161 |           | 
##                          |     0.050 |     0.145 |           | 
## -------------------------|-----------|-----------|-----------|
##                   ontime |        22 |       333 |       355 | 
##                          |     0.062 |     0.938 |     0.805 | 
##                          |     0.500 |     0.839 |           | 
##                          |     0.050 |     0.755 |           | 
## -------------------------|-----------|-----------|-----------|
##             Column Total |        44 |       397 |       441 | 
##                          |     0.100 |     0.900 |           | 
## -------------------------|-----------|-----------|-----------|
## 
## 
```

3.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.

```
nb_model$apriori
```

```
## Y
## delayed  ontime 
##     257    1064
```

```
prop.table(nb_model$apriori)
```

```
## Y
##   delayed    ontime 
## 0.1945496 0.8054504
```

It is sometimes useful to output the raw prediction probabilities rather than the predicted class. To do that, we use the raw option in the model.

```
Predicted_Test_labels1 <-predict(nb_model,testdata, type = "raw")
head (Predicted_Test_labels1)
```

```
##         delayed    ontime
## [1,] 0.37743095 0.6225691
## [2,] 0.34265540 0.6573446
## [3,] 0.30103396 0.6989660
## [4,] 0.33291545 0.6670846
## [5,] 0.05065374 0.9493463
## [6,] 0.44199749 0.5580025
```

ROC Curves

```
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
```

```
roc(testdata$`Flight Status`, Predicted_Test_labels1[,2])
```

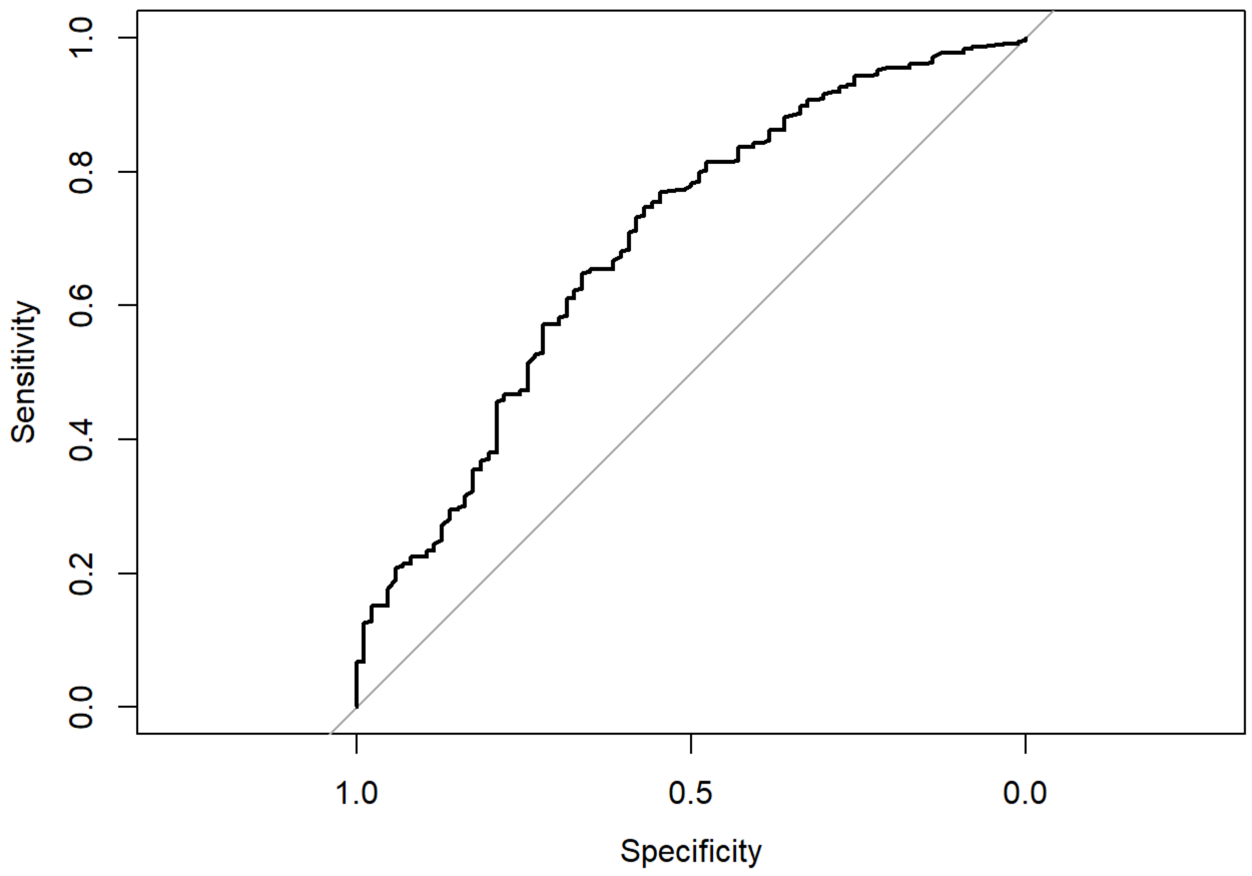
```
## Setting levels: control = delayed, case = ontime
```

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

```
##
## Call:
## roc.default(response = testdata$`Flight Status`, predictor = Predicted_Test_labels1[,
2])
##
## Data: Predicted_Test_labels1[, 2] in 86 controls (testdata$`Flight Status` delayed) < 355
cases (testdata$`Flight Status` ontime).
## Area under the curve: 0.6938
```

```
plot.roc(testdata$`Flight Status`,Predicted_Test_labels1[,2])
```

```
## Setting levels: control = delayed, case = ontime
## Setting direction: controls < cases
```



running iterations on data to check on validation , travel+validation and then entire data

1. validation

```
Predicted_valid_labels <-predict(nb_model,valid)


CrossTable(x=valid$`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$`Flight Status` |   delayed |    ontime | Row Total | 
## ----------------------|-----------|-----------|-----------|
##               delayed |        33 |       138 |       171 | 
##                       |     0.193 |     0.807 |     0.194 | 
##                       |     0.393 |     0.173 |           | 
##                       |     0.037 |     0.157 |           | 
## ----------------------|-----------|-----------|-----------|
##                ontime |        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 |           | 
## ----------------------|-----------|-----------|-----------|
## 
## 
```

```
### accuracy :  (658+39)/880 = .7855
#recall/sensitvity : 658/709 = .928
## precision : 658/(658+138)  = .82
###specificity : 39/171 = .228




Predicted_valid_labels1 <-predict(nb_model,valid, type = "raw")
head (Predicted_valid_labels1)
```

```
##          delayed    ontime
## [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
```

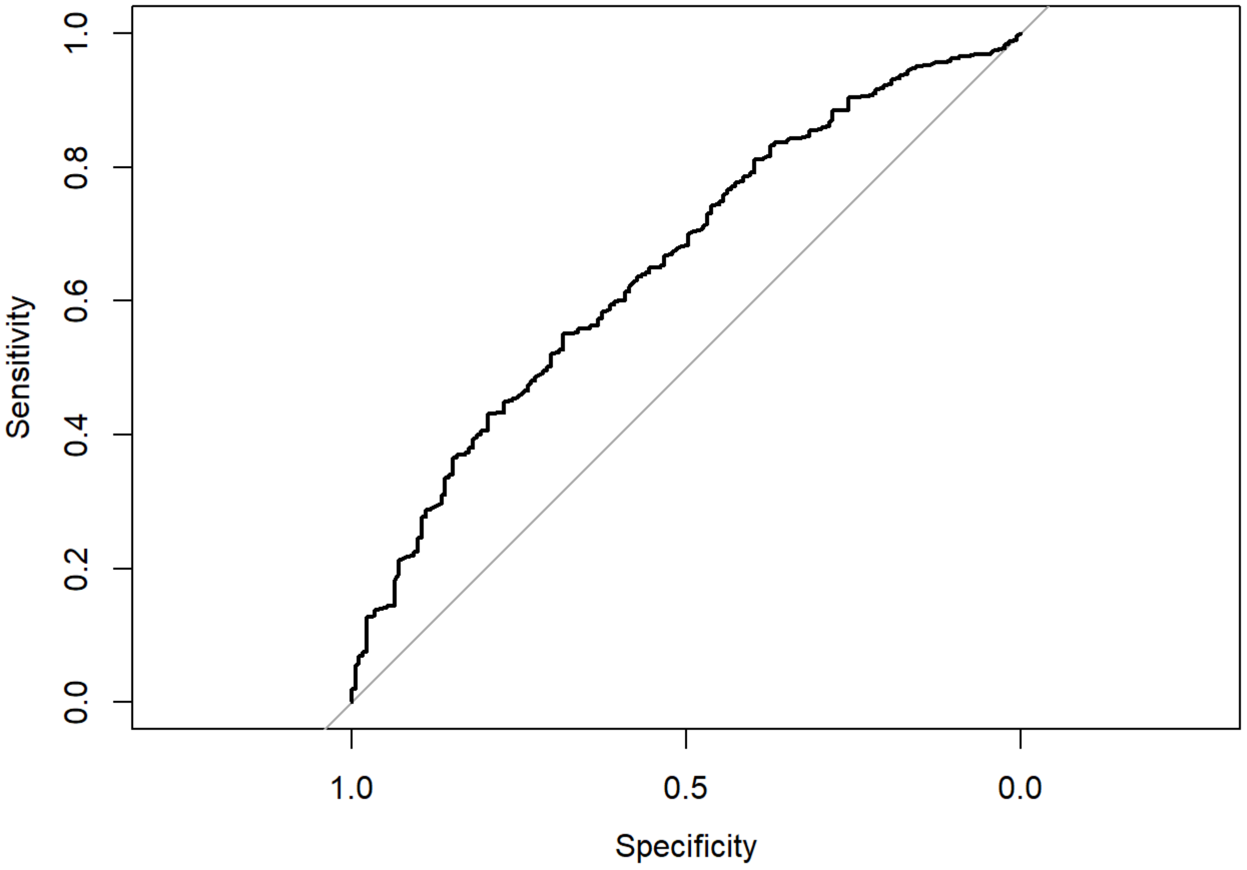
```
roc(valid$`Flight Status`, Predicted_valid_labels1[,2])
```

```
## Setting levels: control = delayed, case = ontime
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = valid$`Flight Status`, predictor = Predicted_valid_labels1[,
2])
##
## Data: Predicted_valid_labels1[, 2] in 171 controls (valid$`Flight Status` delayed) < 709 c
ases (valid$`Flight Status` ontime).
## Area under the curve: 0.6553
```

```
plot.roc(valid$`Flight Status`,Predicted_valid_labels1[,2])
```

```
## Setting levels: control = delayed, case = ontime
## Setting direction: controls < cases
```



2. travel data set

```
Predicted_travel_labels <-predict(nb_model,traveldata)


CrossTable(x=traveldata$`Flight Status`,y=Predicted_travel_labels, prop.chisq = FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  1760 
## 
## 
##                            | Predicted_travel_labels 
## traveldata$`Flight Status` |   delayed |    ontime | Row Total | 
## ---------------------------|-----------|-----------|-----------|
##                    delayed |        68 |       274 |       342 | 
##                            |     0.199 |     0.801 |     0.194 | 
##                            |     0.400 |     0.172 |           | 
##                            |     0.039 |     0.156 |           | 
## ---------------------------|-----------|-----------|-----------|
##                     ontime |       102 |      1316 |      1418 | 
##                            |     0.072 |     0.928 |     0.806 | 
##                            |     0.600 |     0.828 |           | 
##                            |     0.058 |     0.748 |           | 
## ---------------------------|-----------|-----------|-----------|
##               Column Total |       170 |      1590 |      1760 | 
##                            |     0.097 |     0.903 |           | 
## ---------------------------|-----------|-----------|-----------|
## 
## 
```

```
Predicted_travel_labels1 <-predict(nb_model,traveldata, type = "raw")
head (Predicted_travel_labels1)
```

```
##        delayed    ontime
## [1,] 0.2633882 0.7366118
## [2,] 0.2361207 0.7638793
## [3,] 0.3759201 0.6240799
## [4,] 0.3082130 0.6917870
## [5,] 0.3667645 0.6332355
## [6,] 0.1659532 0.8340468
```

```
roc(traveldata$`Flight Status`, Predicted_travel_labels1[,2])
```

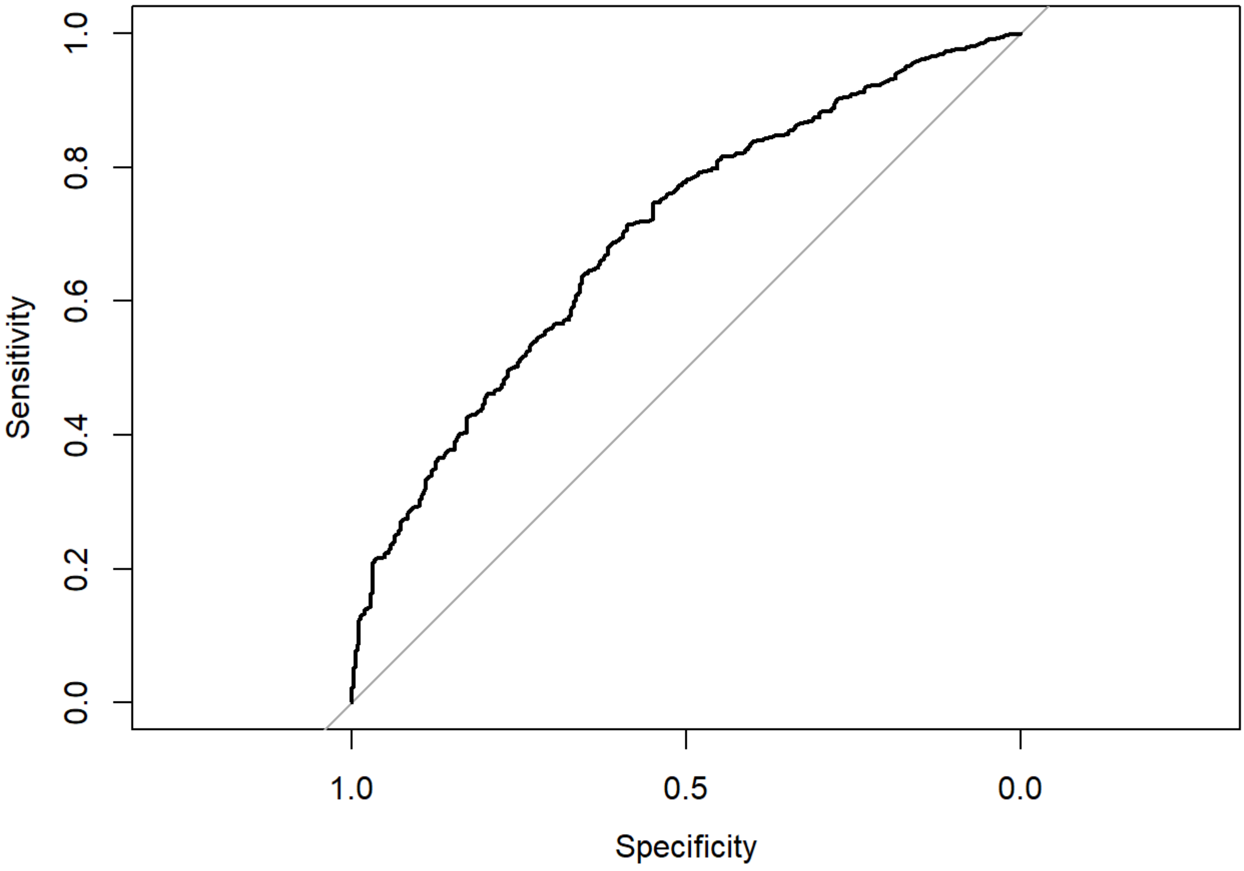
```
## Setting levels: control = delayed, case = ontime
```

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

```
##
## Call:
## roc.default(response = traveldata$`Flight Status`, predictor = Predicted_travel_labels1[,
2])
##
## Data: Predicted_travel_labels1[, 2] in 342 controls (traveldata$`Flight Status` delayed) <
1418 cases (traveldata$`Flight Status` ontime).
## Area under the curve: 0.6974
```

```
plot.roc(traveldata$`Flight Status`,Predicted_travel_labels1[,2])
```

```
## Setting levels: control = delayed, case = ontime
## Setting direction: controls < cases
```



3. entire dataset created Mydata

```
Predicted_MyData_labels <-predict(nb_model,MyData)


CrossTable(x=MyData$`Flight Status`,y=Predicted_MyData_labels, prop.chisq = FALSE)
```

```
## 
##  
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
##  
## Total Observations in Table:  2201 
## 
##  
##                        | Predicted_MyData_labels 
## MyData$`Flight Status` |   delayed |    ontime | Row Total | 
## -----------------------|-----------|-----------|-----------|
##                delayed |        90 |       338 |       428 | 
##                        |     0.210 |     0.790 |     0.194 | 
##                        |     0.421 |     0.170 |           | 
##                        |     0.041 |     0.154 |           | 
## -----------------------|-----------|-----------|-----------|
##                 ontime |       124 |      1649 |      1773 | 
##                        |     0.070 |     0.930 |     0.806 | 
##                        |     0.579 |     0.830 |           | 
##                        |     0.056 |     0.749 |           | 
## -----------------------|-----------|-----------|-----------|
##           Column Total |       214 |      1987 |      2201 | 
##                        |     0.097 |     0.903 |           | 
## -----------------------|-----------|-----------|-----------|
## 
## 
```

```
Predicted_MyData_labels1 <-predict(nb_model,MyData, type = "raw")
head (Predicted_MyData_labels1)
```

```
##        delayed    ontime
## [1,] 0.2633882 0.7366118
## [2,] 0.2361207 0.7638793
## [3,] 0.3759201 0.6240799
## [4,] 0.3082130 0.6917870
## [5,] 0.3667645 0.6332355
## [6,] 0.1659532 0.8340468
```

```
roc(MyData$`Flight Status`, Predicted_MyData_labels1[,2])
```

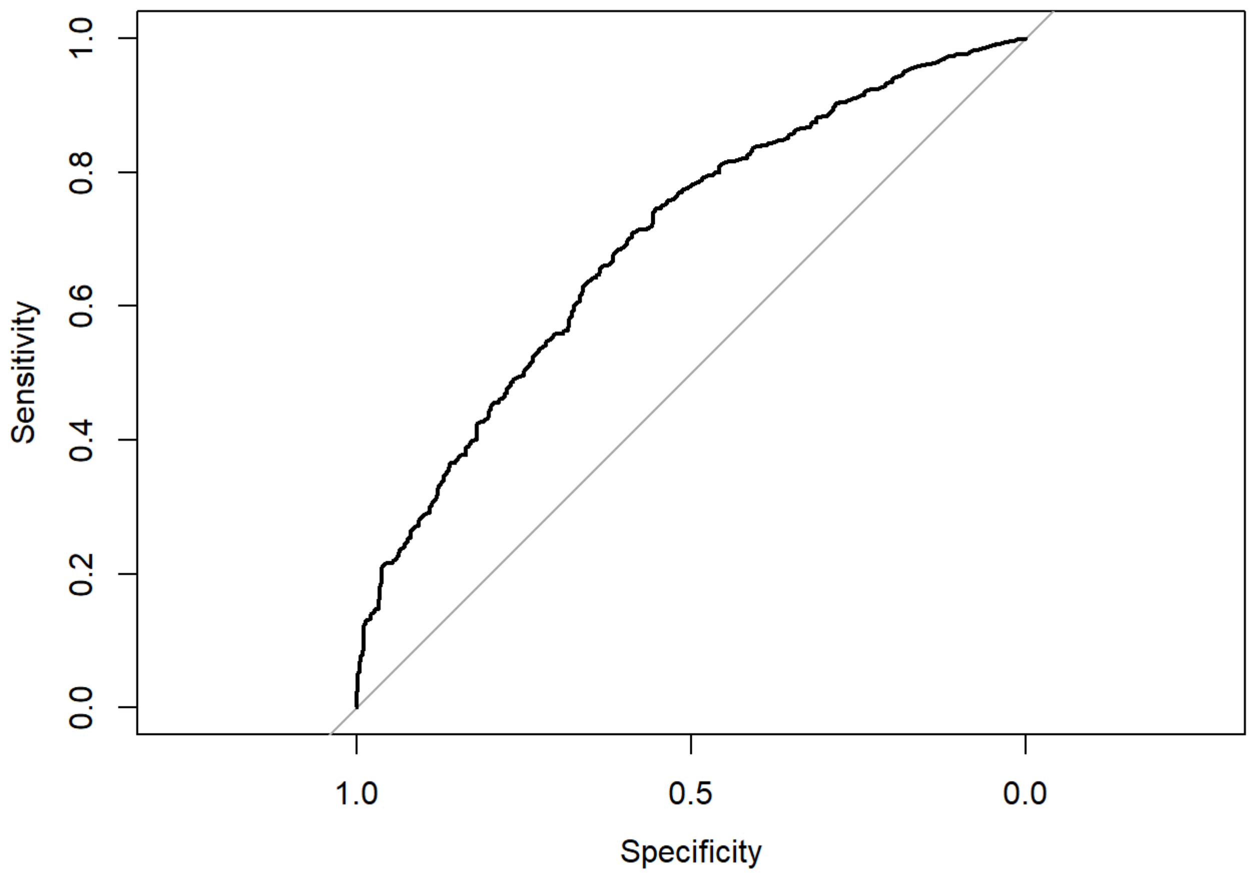
```
## Setting levels: control = delayed, case = ontime
```

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

```
##
## Call:
## roc.default(response = MyData$`Flight Status`, predictor = Predicted_MyData_labels1[,
2])
##
## Data: Predicted_MyData_labels1[, 2] in 428 controls (MyData$`Flight Status` delayed) < 177
3 cases (MyData$`Flight Status` ontime).
## Area under the curve: 0.6965
```

```
plot.roc(MyData$`Flight Status`,Predicted_MyData_labels1[,2])
```

```
## Setting levels: control = delayed, case = ontime
## Setting direction: controls < cases
```


Sensitivity
Specificity