R Notebook

Code

Hide

```
library(caret)
library(ISLR)
library(e1071)

flight <- read.csv("FlightDelays.csv")
flight1 <- flight[,c(-3, -5, -6, -7 ,-9, -11, -12)] ## remove 3,5,6,7,9,11,12 column from dataset
head(flight1)
```

	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
5	1039	DH	LGA	IAD	4	ontime
6	840	DH	JFK	IAD	4	ontime

6 rows

Hide

```
NA
```

Hide

```
flight$Flight.Status = as.factor(flight$Flight.Status) ##Change Flightstatus to factor
flight1$DAY_WEEK = as.factor(flight1$DAY_WEEK) ##Change Day_week to factor
flight1$CRS_DEP_TIME = as.factor(flight$CRS_DEP_TIME) ##Change CRS_DEP_TIME to factor
```

QS1) Divide the data into 60% training and 40% validation

Hide

```
##Clean the data, and divide into training and Validation
set.seed(123)
Train_Index = createDataPartition(flight1$Flight.Status,p=0.6,list=FALSE)  ##divide the data into training and validation.
Train_Data = flight1[Train_Index,]
Validation_Data = flight1[-Train_Index,]


summary(Train_Data)
```

```
  CRS_DEP_TIME     CARRIER     DEST     ORIGIN
 Min.   : 600   DH     :339   EWR:400   BWI: 84
 1st Qu.:1000   RU     :243   JFK:251   DCA:822
 Median :1430   US     :242   LGA:670   IAD:415
 Mean   :1368   DL     :227
 3rd Qu.:1710   MQ     :178
 Max.   :2130   CO     : 57
                (Other): 35
    DAY_WEEK     Flight.Status
 Min.   :1.000   delayed: 257
 1st Qu.:2.000   ontime :1064
 Median :4.000
 Mean   :3.894
 3rd Qu.:5.000
 Max.   :7.000
```

Hide

```
NROW(Validation_Data)
```

```
[1] 880
```

Hide

```
prop.table(table(flight1$Flight.Status)) * 100
```

```
 delayed   ontime
19.44571 80.55429
```

Qs2) Run the Naive Bayes model to predict whether the flight is delayed or not. Use only categorical variables for the predictor variables. Note that Week and Time variables need to recoded as factors

Hide

```
# Build a naïve Bayes classifier

nb_model <-naiveBayes(Flight.Status~CRS_DEP_TIME+CARRIER+DEST+ORIGIN+DAY_WEEK,data = Train_Data)
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:
         CRS_DEP_TIME
Y                  600          630          640
  delayed 0.0000000000 0.0077821012 0.0038910506
  ontime  0.0140977444 0.0291353383 0.0084586466
         CRS_DEP_TIME
Y                  645          700          730
  delayed 0.0000000000 0.0466926070 0.0077821012
  ontime  0.0112781955 0.0422932331 0.0103383459
         CRS_DEP_TIME
Y                  735          759          800
  delayed 0.0077821012 0.0000000000 0.0077821012
  ontime  0.0084586466 0.0018796992 0.0178571429
         CRS_DEP_TIME
Y                  830          840          845
  delayed 0.0077821012 0.0155642023 0.0000000000
  ontime  0.0140977444 0.0366541353 0.0018796992
         CRS_DEP_TIME
Y                  850          900          925
  delayed 0.0116731518 0.0194552529 0.0000000000
  ontime  0.0150375940 0.0441729323 0.0018796992
         CRS_DEP_TIME
Y                  930         1000         1030
  delayed 0.0000000000 0.0000000000 0.0233463035
  ontime  0.0140977444 0.0159774436 0.0281954887
         CRS_DEP_TIME
Y                 1039         1040         1100
  delayed 0.0038910506 0.0038910506 0.0077821012
  ontime  0.0018796992 0.0084586466 0.0263157895
         CRS_DEP_TIME
Y                 1130         1200         1230
```

```
  delayed 0.0000000000 0.0000000000 0.0000000000
  ontime  0.0131578947 0.0093984962 0.0140977444
         CRS_DEP_TIME
Y                 1240         1245         1300
  delayed 0.0194552529 0.0505836576 0.0350194553
  ontime  0.0150375940 0.0234962406 0.0516917293
         CRS_DEP_TIME
Y                 1315         1330         1359
  delayed 0.0038910506 0.0000000000 0.0116731518
  ontime  0.0000000000 0.0122180451 0.0103383459
         CRS_DEP_TIME
Y                 1400         1430         1455
  delayed 0.0077821012 0.0272373541 0.1050583658
  ontime  0.0234962406 0.0187969925 0.0516917293
         CRS_DEP_TIME
Y                 1500         1515         1520
  delayed 0.0350194553 0.0038910506 0.0000000000
  ontime  0.0347744361 0.0018796992 0.0009398496
         CRS_DEP_TIME
Y                 1525         1530         1600
  delayed 0.0272373541 0.0233463035 0.0350194553
  ontime  0.0084586466 0.0225563910 0.0178571429
         CRS_DEP_TIME
Y                 1605         1610         1630
  delayed 0.0000000000 0.0116731518 0.0155642023
  ontime  0.0000000000 0.0103383459 0.0187969925
         CRS_DEP_TIME
Y                 1640         1645         1700
  delayed 0.0155642023 0.0038910506 0.0272373541
  ontime  0.0131578947 0.0169172932 0.0291353383
         CRS_DEP_TIME
Y                 1710         1715         1720
  delayed 0.0194552529 0.0389105058 0.0233463035
  ontime  0.0103383459 0.0244360902 0.0093984962
         CRS_DEP_TIME
Y                 1725         1730         1800
  delayed 0.0000000000 0.0350194553 0.0038910506
  ontime  0.0009398496 0.0216165414 0.0122180451
         CRS_DEP_TIME
Y                 1830         1900         1930
  delayed 0.0389105058 0.0894941634 0.0077821012
  ontime  0.0253759398 0.0300751880 0.0112781955
         CRS_DEP_TIME
```

```
Y                  2000         2030         2100
  delayed 0.0077821012 0.0116731518 0.0155642023
  ontime  0.0112781955 0.0140977444 0.0206766917
         CRS_DEP_TIME
Y                  2120         2130
  delayed 0.0700389105 0.0038910506
  ontime  0.0375939850 0.0000000000

         CARRIER
Y                   CO           DH           DL           MQ
  delayed 0.066147860 0.322957198 0.112840467 0.178988327
  ontime  0.037593985 0.240601504 0.186090226 0.124060150
         CARRIER
Y                   OH           RU           UA           US
  delayed 0.007782101 0.206225681 0.011673152 0.093385214
  ontime  0.013157895 0.178571429 0.015037594 0.204887218

         DEST
Y                EWR       JFK       LGA
  delayed 0.3891051 0.2217899 0.3891051
  ontime  0.2819549 0.1823308 0.5357143

         ORIGIN
Y                 BWI        DCA        IAD
  delayed 0.07392996 0.51361868 0.41245136
  ontime  0.06109023 0.64849624 0.29041353

         DAY_WEEK
Y                   1          2          3          4
  delayed 0.18677043 0.15953307 0.11284047 0.15175097
  ontime  0.14473684 0.12687970 0.13439850 0.18139098
         DAY_WEEK
Y                   5          6          7
  delayed 0.17509728 0.05447471 0.15953307
  ontime  0.18421053 0.12312030 0.10526316
```

QS3) Output both a counts table and a proportion table outlining how many and what proportion of flights were delayed and ontime at each of the three airports

Hide

```
table(flight1$Flight.Status,flight1$DEST)
```

/

```
         EWR JFK LGA
delayed 161  84 183
ontime  504 302 967
```

Hide

```
prop.table(table(flight1$Flight.Status,flight1$DEST))
```

```
                EWR        JFK        LGA
delayed 0.07314857 0.03816447 0.08314403
ontime  0.22898682 0.13721036 0.43934575
```

Qs4) Output the confusion matrix

Hide

```
##Now, use the model on the Validation set
Predicted_Valid_labels <-predict(nb_model,Validation_Data)

library("gmodels")

# Show the confusion matrix of the classifier
CrossTable(x=Validation_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
Validation_Data$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 |           |
------------------------------|-----------|-----------|-----------|
```

Hide

```
##Our results indicate that we misclassified a total of 189 cases. 138 as False Positives, and 51 as False Negatives.
```

Hide

/

```
##lets output the raw prediction probabilities rather than the predicted class. To do that, we use the raw option in the mod
el.
nb_model <- naiveBayes(Flight.Status~CRS_DEP_TIME+CARRIER+DEST+ORIGIN+DAY_WEEK,data = Train_Data)


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

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

```
         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
```

Qs 4) Output ROC for the validation data

Hide

```
## We can now output the ROC curves.
library(pROC)

#Passing the second column of the predicted probabilities
#That column contains the probability associate to 'ontime'
roc(Validation_Data$Flight.Status, Predicted_validation_labels[,2])
```

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

```


Call:
roc.default(response = Validation_Data$Flight.Status, predictor = Predicted_validation_labels[,     2])

Data: Predicted_validation_labels[, 2] in 171 controls (Validation_Data$Flight.Status delayed) < 709 cases (Validation_Data
$Flight.Status ontime).
Area under the curve: 0.6553
```

##Plot the ROC

Hide

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

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