```
> #https://rstudio-pubs-static.s3.amazonaws.com/347013_371e59e8867549ddb922ea7b530e5c6a.html#fn2
> #Riaz Khan, South Dakota State University
>
> library(faraway)
> library(caret)
Loading required package: lattice

Attaching package: ‘lattice’

The following object is masked from ‘package:faraway’:

    melanoma

Loading required package: ggplot2
Want to understand how all the pieces fit together? Read R for Data Science: https://r4ds.had.co.nz/
Warning message:
package ‘ggplot2’ was built under R version 3.6.3
> library(e1071)
Warning message:
package ‘e1071’ was built under R version 3.6.3
> library(naivebayes)
naivebayes 0.9.7 loaded
Warning message:
package ‘naivebayes’ was built under R version 3.6.3
> library(psych)

Attaching package: ‘psych’

The following objects are masked from ‘package:ggplot2’:

    %+%, alpha

The following object is masked from ‘package:faraway’:

    logit

>
> #Reading data into R
> mydata <- read.csv("C:/Users/jmard/OneDrive/Desktop/Computing and Graphics in Applied
Statistics2020/Bayes_Material/hsbdemo.csv")
```

```
> mydata$sesf <- as.factor(mydata$ses)
>
> #data set contains information on 200 student scores in different subjects and educational choices
(prog = general, academic or vocational).
> #interested in classifying new students into educational choices based on their scores.
>
> head(mydata)
  obs  id female ses schtyp     prog read write math science socst honors awards cid sesf
1   1  45 female   1 public vocation   34    35   41      29    26      0      0   1    1
2   2 108   male   2 public  general   34    33   41      36    36      0      0   1    2
3   3  15   male   3 public vocation   39    39   44      26    42      0      0   1    3
4   4  67   male   1 public vocation   37    37   42      33    32      0      0   1    1
5   5 153   male   2 public vocation   39    31   40      39    51      0      0   1    2
6   6  51 female   3 public  general   42    36   42      31    39      0      0   1    3
> str(mydata)
'data.frame':   200 obs. of  15 variables:
 $ obs    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ id     : int  45 108 15 67 153 51 164 133 2 53 ...
 $ female : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 2 2 1 2 ...
 $ ses    : int  1 2 3 1 2 3 2 2 2 2 ...
 $ schtyp : Factor w/ 2 levels "private","public": 2 2 2 2 2 2 2 2 2 2 ...
 $ prog   : Factor w/ 3 levels "academic","general",..: 3 2 3 3 3 2 3 3 3 3 ...
 $ read   : int  34 34 39 37 39 42 31 50 39 34 ...
 $ write  : int  35 33 39 37 31 36 36 31 41 37 ...
 $ math   : int  41 41 44 42 40 42 46 40 33 46 ...
 $ science: int  29 36 26 33 39 31 39 34 42 39 ...
 $ socst  : int  26 36 42 32 51 39 46 31 41 31 ...
 $ honors : int  0 0 0 0 0 0 0 0 0 0 ...
 $ awards : int  0 0 0 0 0 0 0 0 0 0 ...
 $ cid    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ sesf   : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 2 2 2 2 ...
> describe(mydata)
        vars   n   mean    sd median trimmed   mad min max range  skew kurtosis   se
obs        1 200 100.50 57.88  100.5  100.50 74.13   1 200   199  0.00    -1.22 4.09
id         2 200 100.50 57.88  100.5  100.50 74.13   1 200   199  0.00    -1.22 4.09
female*    3 200   1.46  0.50    1.0    1.44  0.00   1   2     1  0.18    -1.98 0.04
ses        4 200   2.06  0.72    2.0    2.07  1.48   1   3     2 -0.08    -1.10 0.05
schtyp*    5 200   1.84  0.37    2.0    1.92  0.00   1   2     1 -1.84     1.40 0.03
prog*      6 200   1.73  0.84    1.0    1.66  0.00   1   3     2  0.55    -1.37 0.06
read       7 200  52.23 10.25   50.0   52.03 10.38  28  76    48  0.19    -0.66 0.72
```

```
write      8 200  52.77  9.48   54.0   53.36 11.86  31  67    36 -0.47    -0.78 0.67
math       9 200  52.65  9.37   52.0   52.23 10.38  33  75    42  0.28    -0.69 0.66
science   10 200  51.85  9.90   53.0   52.02 11.86  26  74    48 -0.19    -0.60 0.70
socst     11 200  52.41 10.74   52.0   52.99 13.34  26  71    45 -0.38    -0.57 0.76
honors    12 200   0.26  0.44    0.0    0.21  0.00   0   1     1  1.06    -0.89 0.03
awards    13 200   1.67  1.82    1.0    1.38  1.48   0   7     7  1.17     0.83 0.13
cid       14 200  10.43  5.80   10.5   10.42  8.15   1  20    19  0.02    -1.21 0.41
sesf*     15 200   2.06  0.72    2.0    2.07  1.48   1   3     2 -0.08    -1.10 0.05
>
> attach(mydata)
The following object is masked _by_ .GlobalEnv:

    sesf

>
> numx <- data.frame(mydata$prog,mydata$read,mydata$write,mydata$math,mydata$science,mydata$socst)
> ## Correlation matrix across all programs
> cor(numx[,2:6],method="pearson")
              mydata.read mydata.write mydata.math mydata.science mydata.socst
mydata.read     1.0000000    0.5967765   0.6622801      0.6301579    0.6214843
mydata.write    0.5967765    1.0000000   0.6174493      0.5704416    0.6047932
mydata.math     0.6622801    0.6174493   1.0000000      0.6307332    0.5444803
mydata.science  0.6301579    0.5704416   0.6307332      1.0000000    0.4651060
mydata.socst    0.6214843    0.6047932   0.5444803      0.4651060    1.0000000
>
> subdata1 <- subset(numx,prog=='academic')
> subdata2 <- subset(numx,prog=='general')
> subdata3 <- subset(numx,prog=='vocation')
>
> ## Correlation matrix within academic
> cor(subdata1[,2:6],method="pearson")
              mydata.read mydata.write mydata.math mydata.science mydata.socst
mydata.read     1.0000000    0.5608413   0.6917634      0.6250391    0.5851566
mydata.write    0.5608413    1.0000000   0.6130255      0.5128848    0.4538175
mydata.math     0.6917634    0.6130255   1.0000000      0.6410174    0.4591657
mydata.science  0.6250391    0.5128848   0.6410174      1.0000000    0.4383806
mydata.socst    0.5851566    0.4538175   0.4591657      0.4383806    1.0000000
>
> ## Correlation matrix within general
> cor(subdata2[,2:6],method="pearson")
```

```
               mydata.read mydata.write mydata.math mydata.science mydata.socst
mydata.read      1.0000000    0.4739121   0.3945974      0.6586988    0.5418732
mydata.write     0.4739121    1.0000000   0.3586417      0.5629392    0.6505204
mydata.math      0.3945974    0.3586417   1.0000000      0.5752819    0.3787115
mydata.science   0.6586988    0.5629392   0.5752819      1.0000000    0.4222026
mydata.socst     0.5418732    0.6505204   0.3787115      0.4222026    1.0000000
>
> ## Correlation matrix within vocational
> cor(subdata3[,2:6],method="pearson")
               mydata.read mydata.write mydata.math mydata.science mydata.socst
mydata.read      1.0000000    0.4615702   0.4570520      0.5132068    0.4325037
mydata.write     0.4615702    1.0000000   0.5090928      0.5225355    0.4926333
mydata.math      0.4570520    0.5090928   1.0000000      0.5706508    0.3769207
mydata.science   0.5132068    0.5225355   0.5706508      1.0000000    0.3348232
mydata.socst     0.4325037    0.4926333   0.3769207      0.3348232    1.0000000
>
> set.seed(7267166)
> trainIndex=createDataPartition(mydata$prog, p=0.7)$Resample1
> train=mydata[trainIndex, ]
> test=mydata[-trainIndex, ]
>
> ## check the balance
> print(table(mydata$prog))

academic  general vocation
     105       45       50
>
> print(table(train$prog))

academic  general vocation
      74       32       35
>
> NBclassfier=naiveBayes(prog~read+write+math+science+socst, data=train)
> print(NBclassfier)

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)
```

```
A-priori probabilities:
Y
 academic   general  vocation
0.5248227 0.2269504 0.2482270

Conditional probabilities:
          read
Y              [,1]     [,2]
  academic 56.32432 9.315046
  general  48.90625 7.985307
  vocation 46.37143 8.135202

          write
Y              [,1]      [,2]
  academic 56.47297  7.996957
  general  51.71875 10.167102
  vocation 46.91429  9.726772

          math
Y              [,1]     [,2]
  academic 57.04054 8.386942
  general  51.06250 7.568771
  vocation 45.54286 6.976190

          science
Y              [,1]     [,2]
  academic 54.09459 9.064712
  general  52.62500 9.641142
  vocation 45.80000 9.420878

          socst
Y              [,1]      [,2]
  academic 56.41892  9.353055
  general  50.81250  9.959393
  vocation 43.88571 10.615669

>
> printALL=function(model){
+   trainPred=predict(model, newdata = train, type = "class")
+   trainTable=table(train$prog, trainPred)
```

```
+   testPred=predict(NBclassfier, newdata=test, type="class")
+   testTable=table(test$prog, testPred)
+   trainAcc=(trainTable[1,1]+trainTable[2,2]+trainTable[3,3])/sum(trainTable)
+   testAcc=(testTable[1,1]+testTable[2,2]+testTable[3,3])/sum(testTable)
+   message("Contingency Table for Training Data")
+   print(trainTable)
+   message("Contingency Table for Test Data")
+   print(testTable)
+   message("Accuracy")
+   print(round(cbind(trainAccuracy=trainAcc, testAccuracy=testAcc),3))
+ }
> printALL(NBclassfier)
Contingency Table for Training Data
          trainPred
           academic general vocation
  academic       57       4       13
  general        15       3       14
  vocation        9       3       23
Contingency Table for Test Data
          testPred
           academic general vocation
  academic       22       4        5
  general         6       3        4
  vocation        4       2        9
Accuracy
     trainAccuracy testAccuracy
[1,]         0.589        0.576
>
> ## Results after converting ses as factor  ses=Social Economic Status 1=low 2=middle 3=high
>
> ## Contingency Table for Training Data
>
> newNBclassifier=naive_bayes(prog~sesf+read+write+math+science+socst,usekernel=T, data=train)
> printALL(newNBclassifier)
Contingency Table for Training Data
          trainPred
           academic general vocation
  academic       62       0       12
  general        15       8        9
  vocation        9       2       24
```

```
Contingency Table for Test Data
          testPred
           academic general vocation
  academic       22       4        5
  general         6       3        4
  vocation        4       2        9
Accuracy
     trainAccuracy testAccuracy
[1,]         0.667        0.576
Warning message:
predict.naive_bayes(): more features in the newdata are provided as there are probability tables in the
object. Calculation is performed based on features to be found in the tables.
>
```