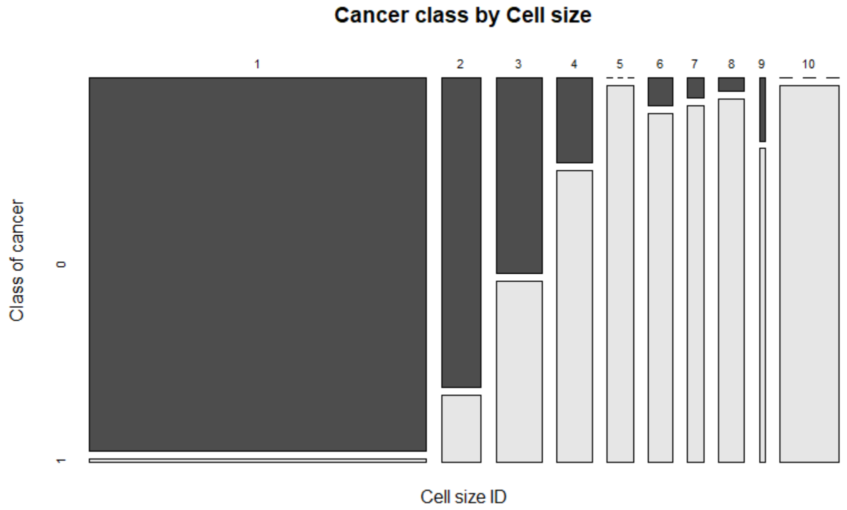
# Roll no. 33140

# Batch: L9

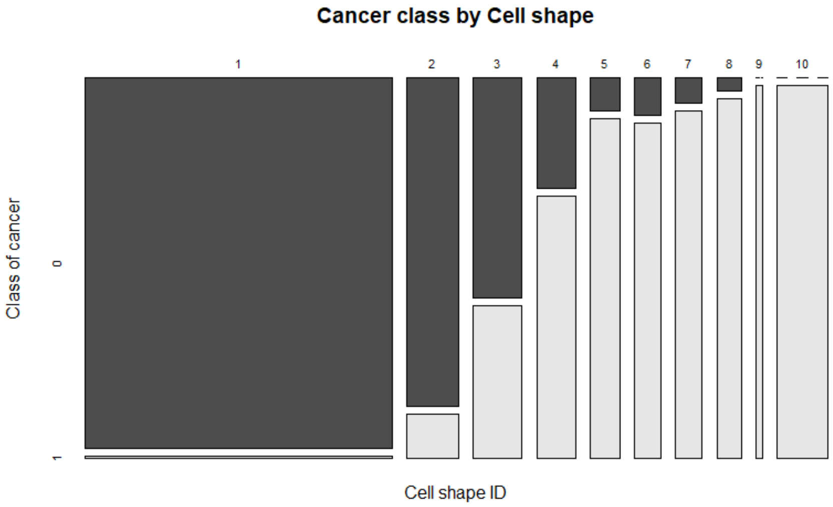
# PS. Using Naive-Bayes algorithm to predict Breast Cancer

```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from G:/College/SL6/Assignment6/.RData]

> # set the working directory

> setwd("G:/College/SL6/Assignment6")

> # Load libraries
> library('caTools')

> # Read the dataset
> breast_cancer = read.csv2("../../Sl-VI DataSets/BreastCancer/BreastCancerWc.csv",
                header = T,sep =',')

> names(breast_cancer)

 [1] "X1000025" "X5"       "X1"       "X1.1"     "X1.2"     "X2"       "X1.3"
     "X3"       "X1.4"     "X1.5"
[11] "X2.1"

> # Set the labels
> names(breast_cancer)[1] = "ID"
> names(breast_cancer)[2] = "CT" # Clump thickness
> names(breast_cancer)[3] = "CellSize"
> names(breast_cancer)[4] = "CellShape"
> names(breast_cancer)[5] = "MA" # Marginal adhesion
> names(breast_cancer)[6] = "ECellSize" # Epithelial cell size
> names(breast_cancer)[7] = "BN" # Bare nuclei
> names(breast_cancer)[8] = "BC" # Bland chromatin
> names(breast_cancer)[9] = "NN" # Normal nuclei
> names(breast_cancer)[10] = "Mit" # Mitoses
> names(breast_cancer)[11] = "Class" # class
> names(breast_cancer)
 [1] "ID"        "CT"        "CellSize"  "CellShape" "MA" "ECellSize" "BN"  "BC"
     "NN"
[10] "Mit"       "Class"

> breast_cancer$Class
  [1] 2 2 2 2 4 2 2 2 2 2 2 4 2 4 4 2 2 4 2 4 4 2 4 2 4 2 2 2 2 2 2 4 2 2 2 4 2 4 4 2 4 4
 [59] 4 4 2 4 4 2 4 2 4 4 2 2 4 2 4 4 2 2 2 2 2 2 2 2 2 4 4 4 4 2 2 2 2 2 2 2 2 2 2 4 4 4
[117] 4 2 2 2 2 4 4 4 2 4 2 4 2 2 2 4 2 2 2 2 2 2 2 2 2 4 2 2 2 4 2 2 4 2 4 4 2 2 4 2 2 2
[175] 4 2 4 2 4 2 2 2 4 4 2 4 4 4 2 4 4 2 2 2 2 2 2 2 2 4 4 2 2 2 4 4 2 2 2 4 4 2 4 4 4 2
[233] 4 2 2 4 4 4 4 2 2 2 2 2 2 4 4 2 2 2 4 2 4 4 4 2 2 2 2 4 4 4 4 4 2 4 4 4 2 4 2 4 4 2
[291] 2 4 4 2 4 2 2 2 4 4 2 4 2 4 4 2 2 4 2 2 2 4 2 2 2 4 4 2 2 4 2 2 4 2 2 4 2 4 4 4 2 2
[349] 4 2 2 2 4 2 2 4 4 4 4 4 4 2 2 2 2 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 4 2 2 2 2
[407] 2 2 2 2 2 4 2 4 2 4 2 2 2 2 4 2 2 2 4 2 4 2 2 2 2 2 2 2 4 4 2 2 2 4 2 2 2 2 2 2 2 2
[465] 4 4 4 2 2 2 2 2 2 2 2 2 2 2 4 2 2 4 4 2 2 2 4 4 4 2 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 4
[523] 4 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[581] 4 4 2 2 2 4 2 4 2 4 4 4 2 4 2 2 2 2 2 2 2 2 4 4 4 2 2 4 2 4 4 4 2 2 2 2 2 2 2 2 2 2
[639] 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 4 4 4 2 2 2 2 2 2 2 2 2 4
[697] 4 4

> # Set 1 for malignant, 0 for benign (Clean the data)
> breast_cancer$Class <- replace(breast_cancer$Class, breast_cancer$Class == 4,1)
> breast_cancer$Class <- replace(breast_cancer$Class, breast_cancer$Class == 2,0)

> breast_cancer$Class
  [1] 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1
 [59] 1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1
[117] 1 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1 1 0 0 1 0 0 0
[175] 1 0 1 0 1 0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 1 1 1 0
[233] 1 0 0 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0
[291] 0 1 1 0 1 0 0 0 1 1 0 1 0 1 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1 0 1 1 1 0 0
[349] 1 0 0 0 1 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
[407] 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0
[465] 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
[523] 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[581] 1 1 0 0 0 1 0 1 0 1 1 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0
[639] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1
[697] 1 1

> # Check for missing value
> '?' %in% breast_cancer$CT
[1] FALSE

> '?' %in% breast_cancer$CellSize
[1] FALSE

> '?' %in% breast_cancer$CellShape
[1] FALSE

> '?' %in% breast_cancer$MA
[1] FALSE

> '?' %in% breast_cancer$ECellSize
[1] FALSE
```

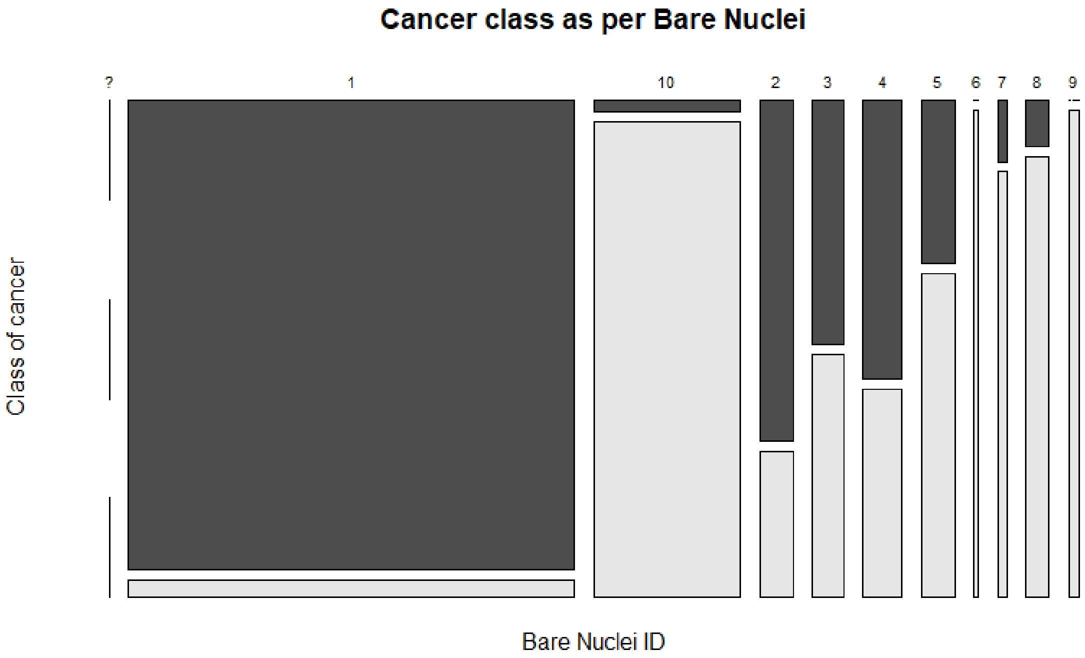
```
> '?' %in% breast_cancer$BN # Returned true (16 values are '?')
[1] TRUE  #i.e. There is a missing value here.

> # replace the NA values
> breast_cancer$BN <- replace(breast_cancer$BN, breast_cancer$BN == '?',NA)
                    # replace ? with NA
> levels(breast_cancer)[levels(breast_cancer)]
NULL

> summary(breast_cancer$CT)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   4.000   4.417   6.000  10.000 

> breast_cancer$BN[is.na(breast_cancer$BN)] <- 4.0 # Median value (replace NA)

> # Mosiac plots of some of the factors vs the class of cancer

> mosaicplot(breast_cancer$CellSize ~ breast_cancer$Class, main = "Cancer class by Cell
+             size",color = TRUE, shade = FALSE, xlab = "Cell size ID", ylab = "Class
+             of cancer")
```



```
> mosaicplot(breast_cancer$CellShape ~ breast_cancer$Class, main = "Cancer class by
+            Cell shape", color = TRUE, shade = FALSE, xlab = "Cell shape ID",
+            ylab = "Class of cancer")
```



```
> mosaicplot(breast_cancer$BN ~ breast_cancer$Class, main = "Cancer class as per
+            Bare Nuclei", color = TRUE, shade = FALSE, xlab = "Bare Nuclei ID",
+            ylab = "Class of cancer")
```



```
> # Create the dataframes for training and testing

> brcdata<-breast_cancer
> brcdata$ID=factor(brcdata$ID)
> brcdata$CT=factor(brcdata$CT)
> brcdata$TCellSize=factor(brcdata$CellSize)
> brcdata$CellShape=factor(brcdata$CellShape)
> brcdata$MA=factor(brcdata$MA)
> brcdata$ECellSize=factor(brcdata$ECellSize)
> brcdata$BN=factor(brcdata$BN)
> brcdata$BC=factor(brcdata$BC)
> brcdata$NN=factor(brcdata$NN)
> brcdata$Mit=factor(brcdata$Mit)
> brcdata$Class=factor(brcdata$Class)

> # Dividing dataset into training and testing
> split = sample.split(brcdata$Class, SplitRatio = 2/3)
> train_brcdata = subset(brcdata,split == TRUE)
> test_brcdata = subset(brcdata,split == FALSE)

> train_brcdata
        ID CT CellSize CellShape MA ECellSize BN BC NN Mit Class TCellSize
1  1002945  5        4         4  5         7 10  3  2   1     0         4
2  1015425  3        1         1  1         2  2  3  1   1     0         1
8  1033078  2        1         1  1         2  1  1  1   5     0         1
9  1033078  4        2         1  1         2  1  2  1   1     0         2
10 1035283  1        1         1  1         1  1  3  1   1     0         1
11 1036172  2        1         1  1         2  1  2  1   1     0         1
12 1041801  5        3         3  3         2  3  4  4   1     1         3
14 1044572  8        7         5 10         7  9  5  5   4     1         7
15 1047630  7        4         6  4         6  1  4  3   1     1         4
16 1048672  4        1         1  1         2  1  2  1   1     0         1
18 1050670 10        7         7  6         4 10  4  1   2     1         7
19 1050718  6        1         1  1         2  1  3  1   1     0         1
21 1054593 10        5         5  3         6  7  7 10   1     1         5
22 1056784  3        1         1  1         2  1  2  1   1     0         1
23 1057013  8        4         5  1         2  4  7  3   1     1         4
24 1059552  1        1         1  1         2  1  3  1   1     0         1
25 1065726  5        2         3  4         2  7  3  6   1     1         2
27 1066979  5        1         1  1         2  1  2  1   1     0         1
28 1067444  2        1         1  1         2  1  2  1   1     0         1
29 1070935  1        1         3  1         2  1  1  1   1     0         1
30 1070935  3        1         1  1         1  1  2  1   1     0         1
31 1071760  2        1         1  1         2  1  3  1   1     0         1
33 1074610  2        1         1  2         2  1  3  1   1     0         1
35 1079304  2        1         1  1         2  1  2  1   1     0         1
36 1080185 10       10        10  8         6  1  8  9   1     1        10
37 1081791  6        2         1  1         1  1  7  1   1     0         2
38 1084584  5        4         4  9         2 10  5  6   1     1         4
39 1091262  2        5         3  3         6  7  7  5   1     1         5
40 1096800  6        6         6  9         6  4  7  8   1     0         6
41 1099510 10        4         3  1         3  3  6  5   2     1         4
42 1100524  6       10        10  2         8 10  7  3   3     1        10
45 1103722  1        1         1  1         2  1  2  1   2     0         1
47 1105524  1        1         1  1         2  1  2  1   1     0         1
49 1106829  7        8         7  2         4  8  3  8   2     1         8
51 1108449  5        3         3  4         2  4  3  4   1     1         3
52 1110102 10        3         6  2         3  5  4 10   2     1         3
53 1110503  5        5         5  8        10  8  7  3   7     1         5
54 1110524 10        5         5  6         8  8  7  1   1     1         5
```

```
56  1112209  8       10        10  1         3  6  3  9   1     1        10
57  1113038  8        2         4  1         5  1  5  4   4     1         2
58  1113483  5        2         3  1         6 10  5  1   1     1         2
59  1113906  9        5         5  2         2  2  5  1   1     1         5
60  1115282  5        3         5  5         3  3  4 10   1     1         3
61  1115293  1        1         1  1         2  2  2  1   1     0         1
63  1116132  6        3         4  1         5  2  3  9   1     1         3
64  1116192  1        1         1  1         2  1  2  1   1     0         1
65  1116998 10        4         2  1         3  2  4  3  10     1         4
66  1117152  4        1         1  1         2  1  3  1   1     0         1
67  1118039  5        3         4  1         8 10  4  9   1     1         3
68  1120559  8        3         8  3         4  9  8  9   8     1         3
70  1121919  5        1         3  1         2  1  2  1   1     0         1
71  1123061  6       10         2  8        10  2  7  8  10     1        10
75  1131294  1        1         2  1         2  2  4  2   1     0         1
76  1132347  1        1         4  1         2  1  2  1   1     0         1
77  1133041  5        3         1  2         2  1  2  1   1     0         3
78  1133136  3        1         1  1         2  3  3  1   1     0         1
79  1136142  2        1         1  1         3  1  2  1   1     0         1
80  1137156  2        2         2  1         1  1  7  1   1     0         2
81  1143978  4        1         1  2         2  1  2  1   1     0         1
84  1147699  3        5         7  8         8  9  7 10   7     1         5
85  1147748  5       10         6  1        10  4  4 10  10     1        10
88  1152331  4        1         1  1         2  1  3  1   1     0         1
90  1156272  1        1         1  1         2  1  3  1   1     0         1
91  1156948  3        1         1  2         2  1  1  1   1     0         1
92  1157734  4        1         1  1         2  1  3  1   1     0         1
96  1165297  2        1         1  2         2  1  1  1   1     0         1
97  1165790  5        1         1  1         2  1  3  1   1     0         1
98  1165926  9        6         9  2        10  6  2  9  10     1         6
99  1166630  7        5         6 10         5 10  7  9   4     1         5
101 1167439  2        3         4  4         2  5  2  5   1     1         3
103 1168359  8        2         3  1         6  3  7  1   1     1         2
104 1168736 10       10        10 10        10  1  8  8   8     1        10
105 1169049  7        3         4  4         3  3  3  2   7     1         3
106 1170419 10       10        10  8         2 10  4  1   1     1        10
107 1170420  1        6         8 10         8 10  5  7   1     1         6
110 1171795  1        3         1  2         2  2  5  3   2     0         3
111 1171845  8        6         4  3         5  9  3  1   1     1         6
112 1172152 10        3         3 10         2 10  7  3   3     1         3
114 1173235  3        3         2  1         2  3  3  1   1     0         3
115 1173347  1        1         1  1         2  5  1  1   1     0         1
117 1173509  4        5         5 10         4 10  7  5   8     1         5
119 1173681  3        2         1  1         2  2  3  1   1     0         2
120 1174057  1        1         2  2         2  1  3  1   1     0         1
 [ reached 'max' / getOption("max.print") -- omitted 383 rows ]
```

```
> # Applying Naive Bayes claasifier on dataset
> library(e1071)
> classifier <- naiveBayes(Class ~ CT+CellSize+CellShape+MA+ECellSize+BN+BC+NN+Mit,
                           train_brcdata)
> classifier

Naive Bayes Classifier for Discrete Predictors

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

A-priori probabilities:
Y
        0         1
0.6545064 0.3454936

Conditional probabilities:
   CT
Y             1           2           3           4           5           6           7
           8           9
  0 0.318032787 0.101639344 0.196721311 0.157377049 0.173770492 0.042622951 0.003278689
0.006557377 0.000000000
  1 0.018633540 0.012422360 0.031055901 0.049689441 0.198757764 0.074534161 0.105590062
0.167701863 0.043478261
   CT
Y            10
  0 0.000000000
  1 0.298136646

   CellSize
Y       [,1]      [,2]
  0 1.334426 0.9352183
  1 6.515528 2.7502400

   CellShape
Y             1           2           3           4           5           6           7
          8           9
  0 0.777049180 0.118032787 0.055737705 0.032786885 0.003278689 0.006557377 0.006557377
0.000000000 0.000000000
  1 0.006211180 0.031055901 0.118012422 0.118012422 0.136645963 0.093167702 0.099378882
 0.136645963 0.031055901
   CellShape
Y            10
  0 0.000000000
  1 0.229813665

   MA
Y             1           2           3           4           5           6           7
           8           9
  0 0.819672131 0.081967213 0.062295082 0.013114754 0.013114754 0.006557377 0.000000000
0.000000000 0.003278689
  1 0.136645963 0.099378882 0.080745342 0.111801242 0.093167702 0.055900621 0.062111801
0.118012422 0.012422360
   MA
Y            10
```

```
  0 0.000000000
  1 0.229813665

   ECellSize
Y             1           2           3           4           5           6           7
           8           9
  0 0.104918033 0.780327869 0.072131148 0.009836066 0.013114754 0.006557377 0.006557377
0.003278689 0.000000000
  1 0.006211180 0.111801242 0.167701863 0.167701863 0.149068323 0.173913043 0.018633540
 0.080745342 0.012422360
   ECellSize
Y            10
  0 0.003278689
  1 0.111801242

   BN
Y             1          10           2           3           4           5           6
           7           8
  0 0.852459016 0.006557377 0.039344262 0.026229508 0.039344262 0.026229508 0.000000000
0.003278689 0.006557377
  1 0.068322981 0.540372671 0.055900621 0.037267081 0.062111801 0.074534161 0.018633540
0.037267081 0.062111801
   BN
Y             9
  0 0.000000000
  1 0.043478261

   BC
Y             1           2           3           4           5           6           7
          8           9
  0 0.337704918 0.344262295 0.262295082 0.022950820 0.013114754 0.003278689 0.016393443
 0.000000000 0.000000000
  1 0.012422360 0.037267081 0.130434783 0.136645963 0.130434783 0.018633540 0.310559006
0.111801242 0.043478261
   BC
Y            10
  0 0.000000000
  1 0.068322981

   NN
Y             1           2           3           4           5           6           7
          8           9
  0 0.865573770 0.075409836 0.029508197 0.003278689 0.003278689 0.003278689 0.003278689
0.013114754 0.003278689
  1 0.173913043 0.018633540 0.167701863 0.062111801 0.080745342 0.055900621 0.062111801
0.062111801 0.074534161
   NN
Y            10
  0 0.000000000
  1 0.242236025

   Mit
Y             1           2           3           4           5           6           7
          8          10
  0 0.957377049 0.026229508 0.006557377 0.000000000 0.003278689 0.000000000 0.003278689
```

```
0.003278689 0.000000000
  1 0.552795031 0.111801242 0.130434783 0.037267081 0.012422360 0.012422360 0.024844720
0.037267081 0.080745342

> #predict using trained model
> prediction <- predict(classifier, test_brcdata ,type="class")

> table(prediction,  test_brcdata[,11])   # put it in table

prediction   0   1
         0 147   0
         1   5  80

> # Displaying the accuracy using confusion Matrix
> library(e1071)
> library(caret)

> df1=confusionMatrix(test_brcdata[,11],prediction) #create the confusion matrix

> df1
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 147   5
         1   0  80

               Accuracy : 0.9784
                 95% CI : (0.9504, 0.993)
    No Information Rate : 0.6336
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.953

 Mcnemar's Test P-Value : 0.07364

            Sensitivity : 1.0000
            Specificity : 0.9412
         Pos Pred Value : 0.9671
         Neg Pred Value : 1.0000
             Prevalence : 0.6336
         Detection Rate : 0.6336
   Detection Prevalence : 0.6552
      Balanced Accuracy : 0.9706

       'Positive' Class : 0


>
```