Assignment 6_2

33140 (Sahil Naphade)

23/04/2020

Load the libraries and read the file

```
# Roll no. 33140
# Batch: L9
# PS. Using Naive-Bayes algorithm and SVM to predict Breast Cancer

# set the working directory
setwd("G:/College/SL6/Assignment6")

# Load libraries
library('caTools')
library(e1071)
library(caret)
```

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

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

```
# 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"
##  [7] "X1.3"     "X3"       "X1.4"     "X1.5"     "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"
##  [7] "BN"        "BC"        "NN"        "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
```

```
##  [38] 4 4 2 4 4 4 4 2 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 2 4 2 4 4 2 2 4 2 4 4
##  [75] 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 4 2 4 4 4 4 4 2 4 2 4
## [112] 4 4 2 2 2 4 2 2 2 2 4 4 4 2 4 2 4 2 2 2 4 2 2 2 2 2 2 2 2 4 2 2 2 4 2 2
## [149] 4 2 4 4 2 2 4 2 2 2 4 4 2 2 2 2 2 4 4 2 2 2 2 2 4 4 4 2 4 2 4 2 2 2 4 4 2
## [186] 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 2 4 2 2 4 4
## [223] 4 4 2 4 4 2 4 4 4 2 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
## [260] 4 4 4 4 4 2 4 4 4 2 4 2 4 4 2 2 2 2 2 4 2 2 4 4 4 4 4 2 4 4 2 2 4 4 2 4 2
## [297] 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 4
## [334] 4 2 4 2 2 4 4 2 2 2 4 2 2 2 4 4 2 2 2 4 2 2 4 4 4 4 4 4 2 2 2 2 4 4 2 2 2
## [371] 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 4 2 2 2 2 4 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2
## [408] 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
## [445] 2 2 2 2 4 2 2 2 2 4 2 4 4 4 2 2 2 2 2 2 2 2 4 4 4 2 2 2 2 2 2 2 2 2 2 2 4 2 2
## [482] 4 4 2 2 2 4 4 4 2 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 4 4 2 2 2
## [519] 4 2 2 4 4 2 2 2 2 2 2 4 2 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
## [556] 2 2 2 2 2 2 2 2 2 4 2 2 4 4 4 4 2 2 4 2 2 2 2 2 2 4 4 2 2 2 4 2 4 2 4 4 4
## [593] 2 4 2 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 2 2 2 4 2 2 2
## [630] 2 2 2 4 2 2 4 2 2 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
## [667] 2 4 4 4 2 2 2 2 2 2 2 2 2 2 2 4 4 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 4 4 4
```

Clean the data

```
# 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 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
##  [38] 1 1 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 0 0 1 0 1 1
##  [75] 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 1 0 1 1 1 1 1 0 1 0 1
## [112] 1 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0
## [149] 1 0 1 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 1 0
## [186] 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 0 1 0 0 1 1
## [223] 1 1 0 1 1 0 1 1 1 0 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
## [260] 1 1 1 1 1 0 1 1 1 0 1 0 1 1 0 0 0 0 0 1 0 0 1 1 1 1 1 0 1 1 0 0 1 1 0 1 0
## [297] 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 1
## [334] 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## [408] 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
## [445] 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
## [482] 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0
## [519] 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
## [556] 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 1
## [593] 0 1 0 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 0 0 0 1 0 0 0
## [630] 0 0 0 1 0 0 1 0 0 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
## [667] 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 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
```

```
# 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
```

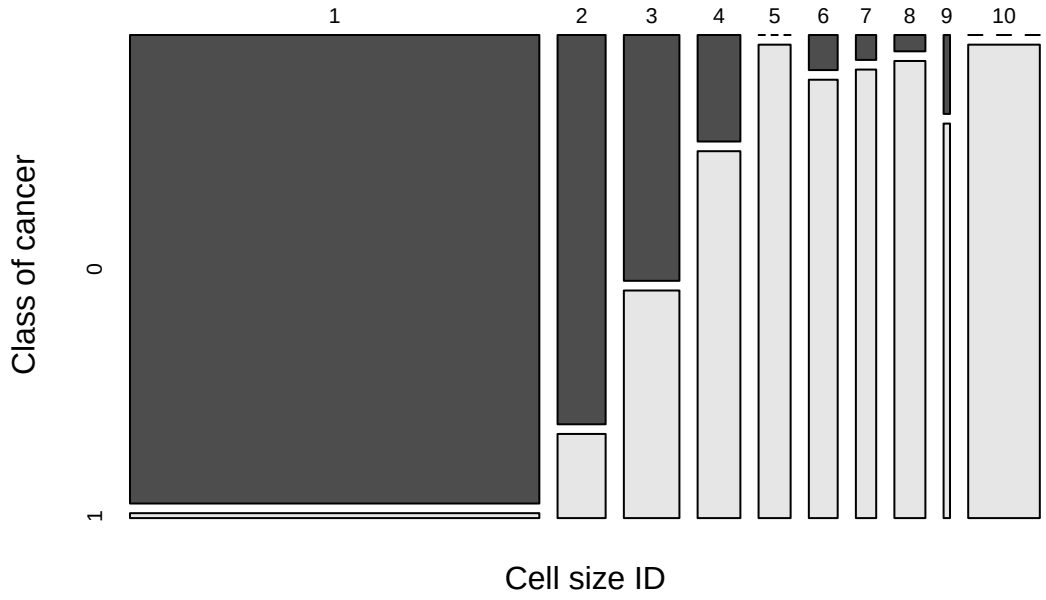
```
summary(breast_cancer$CT)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   4.000   4.417   6.000  10.000
```

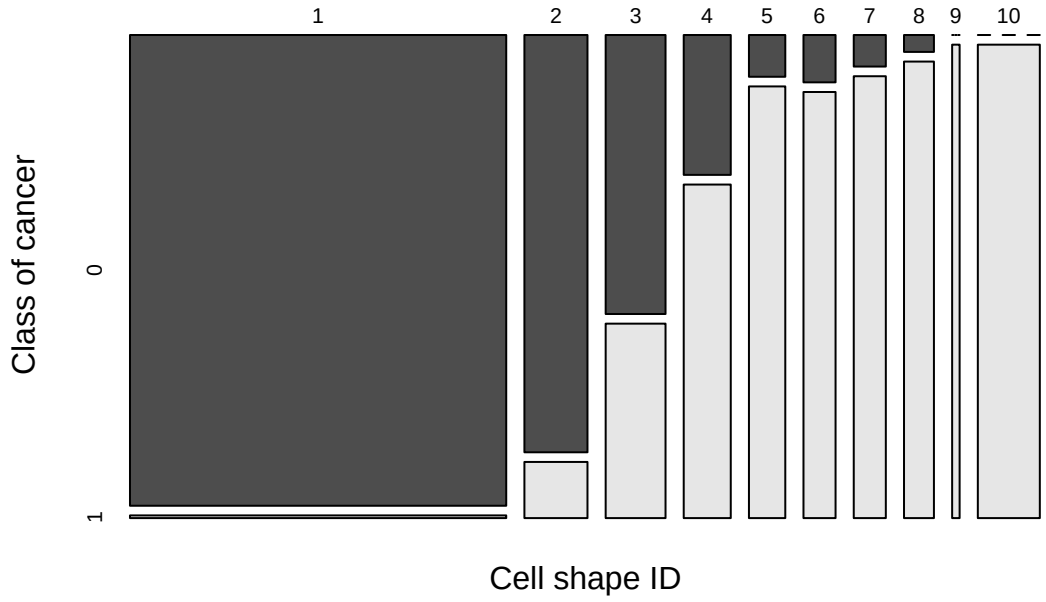
```
breast_cancer$BN[is.na(breast_cancer$BN)] <- 4.0 # Median value (replace NA)
```

Visualize

```
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")
```



Subset the data to create training and testing, and convert the data to factors.

```
# 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)
str(train_brcdata)
```

```
## 'data.frame':    466 obs. of  12 variables:
##  $ ID       : Factor w/ 644 levels "61634","63375",..: 175 179 180 182 186 186 188 193 194 195 ...
##  $ CT       : Factor w/ 10 levels "1","2","3","4",..: 5 4 8 2 2 4 2 1 8 7 ...
##  $ CellSize : int  4 1 10 1 1 2 1 1 7 4 ...
##  $ CellShape: Factor w/ 10 levels "1","2","3","4",..: 4 1 10 2 1 1 1 1 5 6 ...
```

```
##  $ MA       : Factor w/ 10 levels "1","2","3","4",..: 5 3 8 1 1 1 1 1 10 4 ...
##  $ ECellSize: Factor w/ 10 levels "1","2","3","4",..: 7 2 7 2 2 2 2 2 7 6 ...
##  $ BN       : Factor w/ 10 levels "1","10","2","3",..: 2 1 2 1 1 1 1 4 10 1 ...
##  $ BC       : Factor w/ 10 levels "1","2","3","4",..: 3 3 9 3 1 2 2 3 5 4 ...
##  $ NN       : Factor w/ 10 levels "1","2","3","4",..: 2 1 7 1 1 1 1 1 5 3 ...
##  $ Mit      : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 5 1 1 1 4 1 ...
##  $ Class    : Factor w/ 2 levels "0","1": 1 1 2 1 1 1 1 1 2 2 ...
##  $ TCellSize: Factor w/ 10 levels "1","2","3","4",..: 4 1 10 1 1 2 1 1 7 4 ...
```

APPLYING NAIVE-BAYES

```
# Applying-Naive Bayes claasifier on dataset (Training)
library(e1071) # import library
classifier <- naiveBayes(Class ~ CT+CellSize+CellShape+MA+ECellSize+BN+BC+NN+Mit,train_brcdata) # train
classifier    # check the prediction model for Naive Bayes
```

```
##
## 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
##   0 0.334426230 0.095081967 0.200000000 0.134426230 0.203278689 0.026229508
##   1 0.012422360 0.012422360 0.043478261 0.062111801 0.192546584 0.080745342
##    CT
## Y             7           8           9          10
##   0 0.000000000 0.006557377 0.000000000 0.000000000
##   1 0.105590062 0.180124224 0.062111801 0.248447205
##
##    CellSize
## Y       [,1]      [,2]
##   0 1.281967 0.8306442
##   1 6.813665 2.6697494
##
##    CellShape
## Y             1           2           3           4           5           6
##   0 0.773770492 0.114754098 0.062295082 0.032786885 0.003278689 0.006557377
##   1 0.006211180 0.012422360 0.099378882 0.124223602 0.111801242 0.130434783
##    CellShape
## Y             7           8           9          10
##   0 0.006557377 0.000000000 0.000000000 0.000000000
##   1 0.130434783 0.111801242 0.024844720 0.248447205
##
##    MA
## Y             1           2           3           4           5           6
##   0 0.806557377 0.095081967 0.062295082 0.013114754 0.009836066 0.009836066
##   1 0.105590062 0.099378882 0.124223602 0.111801242 0.086956522 0.068322981
```

```
##    MA
## Y             7           8           9          10
##   0 0.000000000 0.000000000 0.000000000 0.003278689
##   1 0.062111801 0.105590062 0.012422360 0.223602484
##
##    ECellSize
## Y             1           2           3           4           5           6
##   0 0.101639344 0.786885246 0.062295082 0.022950820 0.013114754 0.000000000
##   1 0.006211180 0.068322981 0.149068323 0.192546584 0.142857143 0.180124224
##    ECellSize
## Y             7           8           9          10
##   0 0.003278689 0.006557377 0.000000000 0.003278689
##   1 0.049689441 0.093167702 0.006211180 0.111801242
##
##    BN
## Y             1          10           2           3           4           5
##   0 0.836065574 0.003278689 0.055737705 0.032786885 0.039344262 0.022950820
##   1 0.062111801 0.559006211 0.043478261 0.062111801 0.062111801 0.068322981
##    BN
## Y             6           7           8           9
##   0 0.000000000 0.003278689 0.006557377 0.000000000
##   1 0.012422360 0.018633540 0.074534161 0.037267081
##
##    BC
## Y             1           2           3           4           5           6
##   0 0.331147541 0.340983607 0.291803279 0.013114754 0.009836066 0.000000000
##   1 0.006211180 0.031055901 0.155279503 0.130434783 0.118012422 0.037267081
##    BC
## Y             7           8           9          10
##   0 0.013114754 0.000000000 0.000000000 0.000000000
##   1 0.273291925 0.111801242 0.037267081 0.099378882
##
##    NN
## Y             1           2           3           4           5           6
##   0 0.872131148 0.068852459 0.032786885 0.003278689 0.003278689 0.009836066
##   1 0.118012422 0.031055901 0.130434783 0.062111801 0.055900621 0.086956522
##    NN
## Y             7           8           9          10
##   0 0.003278689 0.006557377 0.000000000 0.000000000
##   1 0.068322981 0.099378882 0.074534161 0.273291925
##
##    Mit
## Y             1           2           3           4           5           6
##   0 0.970491803 0.022950820 0.000000000 0.000000000 0.003278689 0.000000000
##   1 0.552795031 0.093167702 0.155279503 0.062111801 0.024844720 0.012422360
##    Mit
## Y             7           8          10
##   0 0.003278689 0.000000000 0.000000000
##   1 0.031055901 0.024844720 0.043478261
```

Display accuracy

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

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

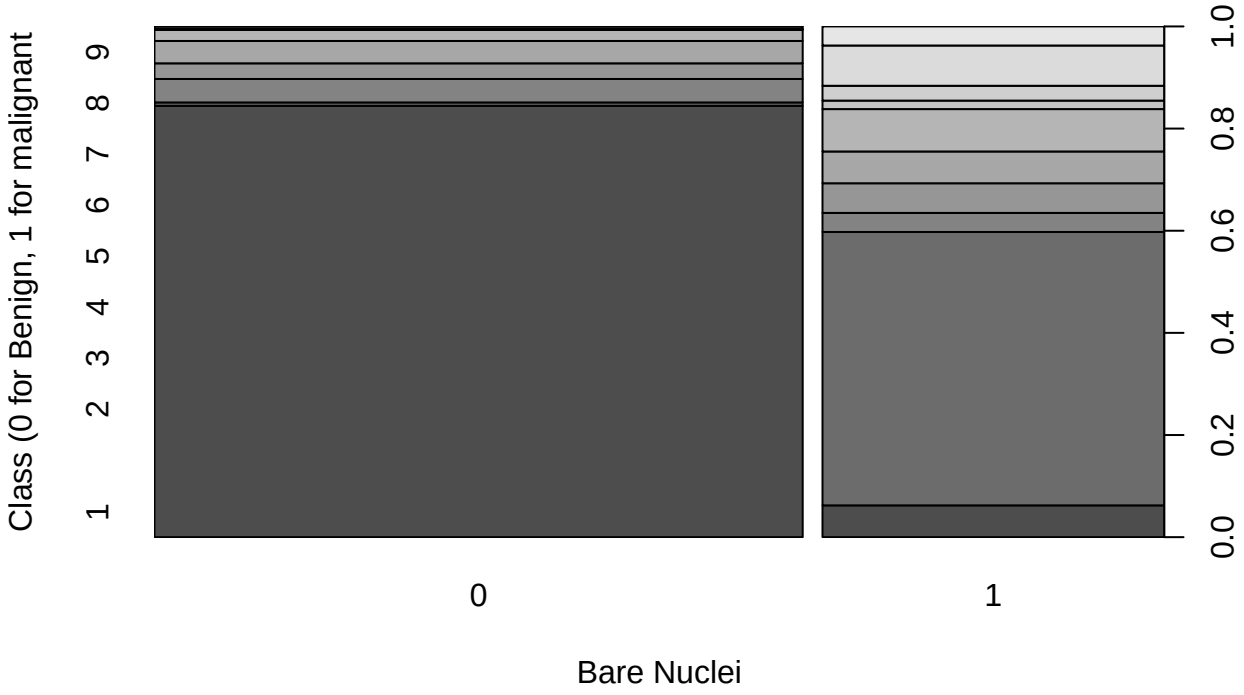
```
##
## prediction   0   1
##          0 145   2
##          1   7  78
```

```r
# Displaying the accuracy using confusion Matrix
nb_accuracy <- confusionMatrix(test_brcdata[,11],prediction) #create the confusion matrix
nb_accuracy
```

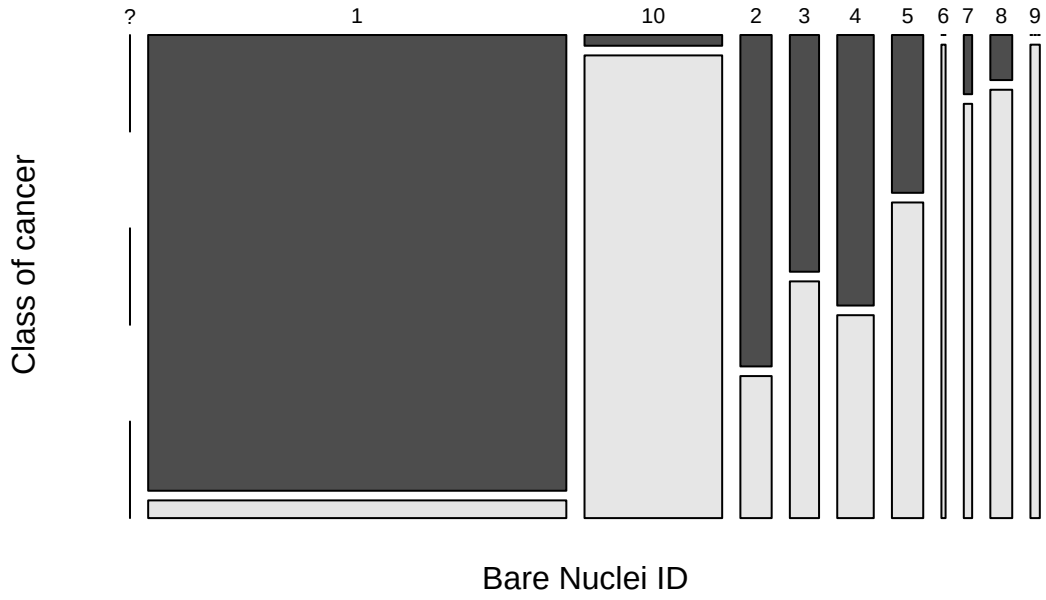
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 145   7
##          1   2  78
##
##                Accuracy : 0.9612
##                  95% CI : (0.9276, 0.9821)
##     No Information Rate : 0.6336
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9154
##
##  Mcnemar's Test P-Value : 0.1824
##
##             Sensitivity : 0.9864
##             Specificity : 0.9176
##          Pos Pred Value : 0.9539
##          Neg Pred Value : 0.9750
##              Prevalence : 0.6336
##          Detection Rate : 0.6250
##    Detection Prevalence : 0.6552
##       Balanced Accuracy : 0.9520
##
##        'Positive' Class : 0
##
```

Applying SVM

```r
# Applying SVM on the dataset (Bare nuclei vs Class)
# Plot Class vs Bare Nuclei values
# Line plot
plot(x = brcdata$Class,y = brcdata$BN, xlab = "Bare Nuclei", ylab = "Class (0 for Benign, 1 for malignan
```




```
# Mosaic plot
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")
```




```
# Train the SVM model
svm_model <- svm(Class ~ BN,train_brcdata)

# Predict using SVM model
prediction_svm <- predict(svm_model,test_brcdata)

svm_accuracy <- confusionMatrix(test_brcdata[,11],prediction_svm)
svm_accuracy
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 135  17
##          1   7  73
##
##                Accuracy : 0.8966
##                  95% CI : (0.85, 0.9326)
##     No Information Rate : 0.6121
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7776
##
##  Mcnemar's Test P-Value : 0.06619
##
##             Sensitivity : 0.9507
```

```
##             Specificity : 0.8111
##          Pos Pred Value : 0.8882
##          Neg Pred Value : 0.9125
##              Prevalence : 0.6121
##          Detection Rate : 0.5819
##    Detection Prevalence : 0.6552
##       Balanced Accuracy : 0.8809
##
##        'Positive' Class : 0
##
```