**Project 2**

**By Jayanti Jain**

**Health - Breast Cancer Classification & Prediction**

For this assignment, we will be working with the BreastCancer dataset available in the “mlbench” package in Rstudio. The dataset is a default dataset.

**Dataset Description:**

This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. He assessed biopsies of breast tumours for 699 patients up to 15 July 1992; each of nine attributes has been scored on a scale of 1 to 10, and the outcome is also known. There are 699 rows and 11 columns. The variables/features are listed below:

* **Id** - Sample code number - Character
* **Cl.thickness** - Clump Thickness - Ordered Factor.
* **Cell.size** - Uniformity of Cell Size - Ordered Factor.
* **Cell.shape** - Uniformity of Cell Shape - Ordered Factor.
* **Marg.adhesion** - Marginal Adhesion - Ordered Factor.
* **Epith.c.size** - Single Epithelial Cell Size - Ordered Factor.
* **Bare.nuclei** - Bare Nuclei - Factor.
* **Bl.cromatin**- Bland Chromatin - Factor.
* **Normal.nucleoli** - Normal Nucleoli - Factor.
* **Mitoses** - Mitoses - Factor.
* **Class** - Class - Factor. Level is benign or malignant.

**Objective**: To predict if a patient has a malignant tumor from a set of predicting variables.

Classification and Prediction:

Classification problem refers to predicting the target class, here it is malignant or benign by predicting the output from a given set of predicting variables. This kind of prediction can be done by designing a classification model which is also known as a classifier. The classifier is designed by using training and test datasets from the population dataset. Different classification algorithms such as Naïve Bayes algorithm, randomforest, decision tree, neural net classifier, conditional inference tree, and support vector machine will be used to design a classification and shall be used to predict the target class.

**Step1:**

Install all relevant packages:

**Rcode:**

library(mlbench) #Package with dataset- BreastCancer

library(tidyverse) #Package for string function

library(caTools) #Package for splitting the dataset into training and test data

library(caret) #Package for functions for training and plotting models

library(mice) #Package for function to remove NA value in dataset

library(e1071) #Package for function to implement naiveBayes classification algorithm

library(rpart) #Package for function to implement tree algorithm

library(randomForest) #Package for function to implement Random Forest Algorithm

library(rpart.plot) #Package for plotting

library(nnet) #Package to implement NN classifiers

**Step2:**

Load & Explore the dataset

Findings:

The dataset contains 699 observations and the target class is “BreastCancer$Class” which specifies whether the observation is malignant or benign.

The summary of the dataset suggests that it has 16 NA values (missing values).

**Rcode:**

#loading & exploring

#package with breastcancer dataset

require(mlbench)

#loading the dataset

data("BreastCancer")

#Structure of the dataset

str(BreastCancer)

#Finding the levels of target class

levels(BreastCancer$Class)

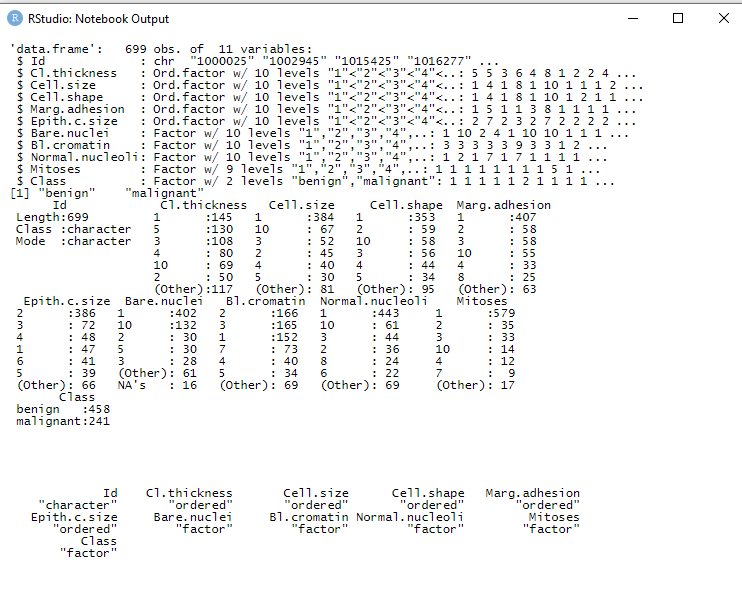
#Summary of Dataset

summary(BreastCancer)

# class of each variables

sapply(BreastCancer, function(x) class(x)[1])

**Output:**



**Step 3:**

Cleaning the Data:

It is not uncommon to have missing data in a dataset. There are several ways to remove missing values. Methods such as omitting the observations, and replacing the missing values with mean/mode of the variable are commonly used. We will use the “mice” package to overcome the 16 NA by imputing the missing values with the most suited values in consideration of all the 9 predicting columns in the dataset. The “Id” column will be filtered as it is irrelevant to our classifier algorithm.

**Rcode:**

#Cleaning the data

#Removing NA values and ID(1st column) from dataset using library mice

dataset\_impute <- mice(BreastCancer[,2:10], print = FALSE)

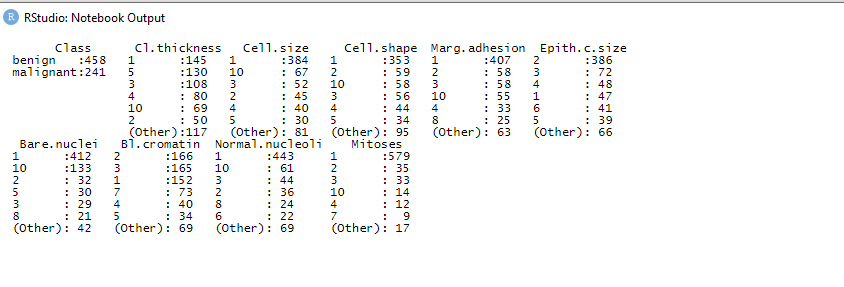
#Adding Target class to the imputed dataset without NA

BreastCancer <- cbind(BreastCancer[,11, drop = FALSE], mice::complete(dataset\_impute, 1))

#Summary of the cleaned Dataset

summary(BreastCancer)

**Ouput:**



At this stage, we will also split our data into taining and testing sets.

**RCode:**

# Create 70% training and 30% validation data

set.seed(120)

# Splitting data into training and test dataset

split=sample.split(BreastCancer, SplitRatio = 0.7)

# Training dataset

training\_set=subset(BreastCancer,split==TRUE)

# Test dataset

test\_set=subset(BreastCancer,split==FALSE)

# Dimenstions of training dataset

dim(training\_set)

# Dimesnions of test dataset

dim(test\_set)

# Removing target class

topredict\_set<-test\_set[2:10]

dim(topredict\_set)

**Step4:**

Our data is prepared at this stage to run through different classification algorithms and the accuracy of each model can be compared.

**Naïve Bayes Classification:**

A Naive Bayes classifier is a probabilistic machine learning model that’s used for the classification task. It is based on the Bayes theorem.

Bayes Theorem:

P(B/A)P(A)

P(A/B) = -------------------------

P(B)

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

**Rcode:**

#Naive Bayes Classification

#Implementing NaiveBayes

model\_naive<- naiveBayes(Class ~ ., data = training\_set)

#Predicting target class for the Validation set

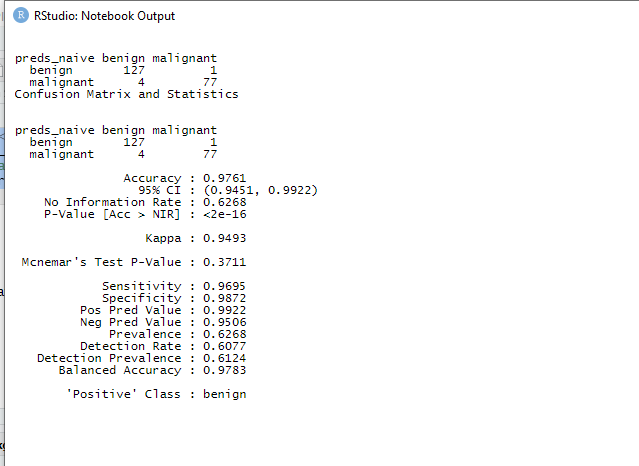
preds\_naive <- predict(model\_naive, newdata = topredict\_set)

(conf\_matrix\_naive <- table(preds\_naive, test\_set$Class))

#Confusion matrix for finding Accuracy of the model

confusionMatrix(conf\_matrix\_naive)

**Output**:



The confusion matrix shows that Naive Bayes classifier predicted 127benign cases correctly and 1 wrong prediction. Similarly, the model predicted 77 malignant cases correctly and 4 wrong predictions.

**The model accuracy is presented at 97.61%.**

**Random Forest Classification**:

Random forest, as its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction.

**Rcode:**

#Randomforest classifier

# Implementing RandomForest

model\_rf <- randomForest(Class ~ ., data = training\_set, importance=TRUE, ntree = 5)

#Predicting target class for the Validation set

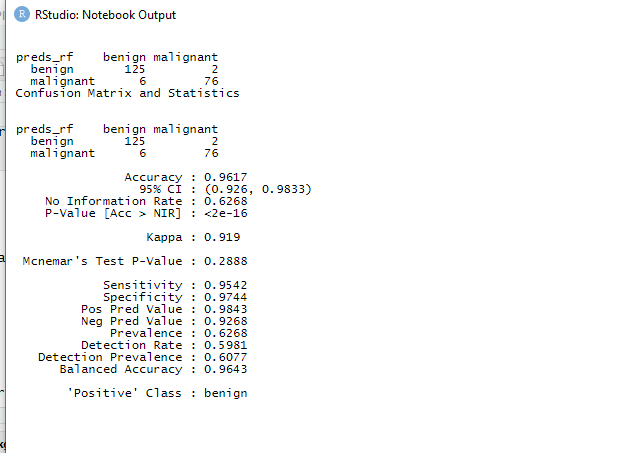
preds\_rf <- predict(model\_rf, topredict\_set)

(conf\_matrix\_forest <- table(preds\_rf, test\_set$Class))

#Confusion matrix for finding Accuracy of the model

confusionMatrix(conf\_matrix\_forest)

**Output:**



The confusion matrix shows that RandomForest classifier predicted 125benign cases correctly and 2 wrong predictions. Similarly, the model predicted 76 malignant cases correctly and 6 wrong predictions.

**The model accuracy is presented at 96.17%.**

**Decision tree classifier:**

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

**Rcode:**

#Decision tree

#Implementing Decision Tree

model\_dtree<- rpart(Class ~ ., data=training\_set)

#Predicting target class for the Validation set

preds\_dtree <- predict(model\_dtree,newdata=topredict\_set, type = "class")

(conf\_matrix\_dtree <- table(preds\_dtree, test\_set$Class))

# plot tree

prp(model\_dtree, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)

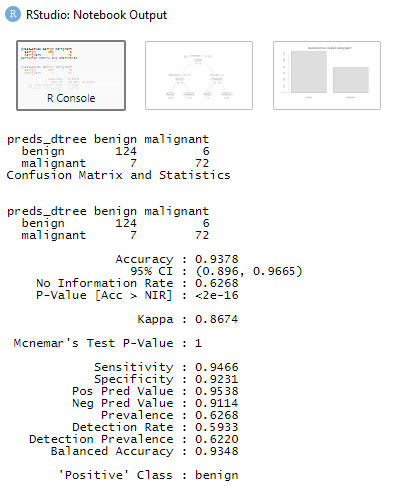
plot(preds\_dtree, main="Decision tree created using rpart")

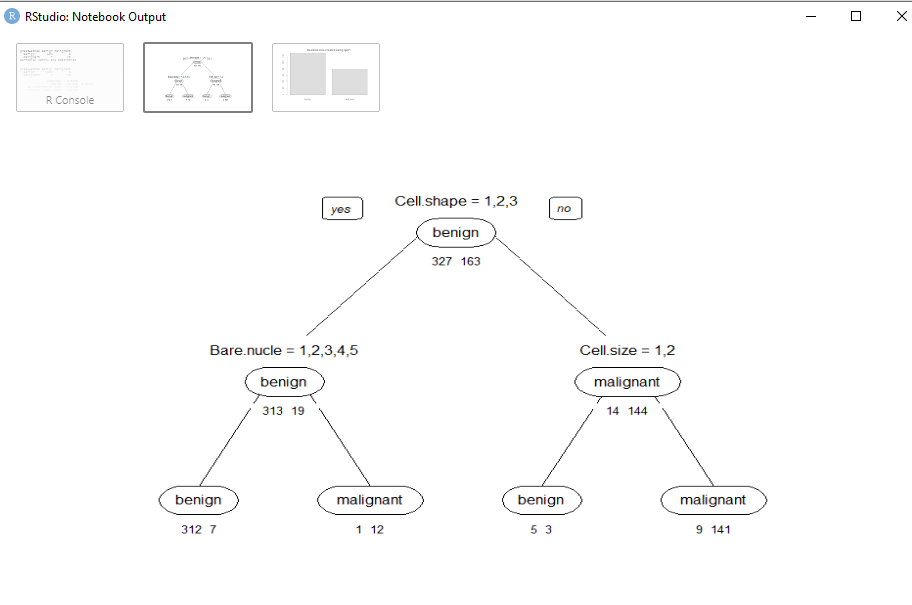
#plot(model\_dtree, main="Decision tree created using rpart")

#Confusion matrix for finding Accuracy of the model

confusionMatrix(conf\_matrix\_dtree)

**Output:**





The confusion matrix shows that decisiontree classifier predicted 124benign cases correctly and 6 wrong predictions. Similarly, the model predicted 72 malignant cases correctly and 7 wrong predictions.

**The model accuracy is presented at 93.78%.**

**Neuralnet Classifier:**

A neural network is made up of interconnected information processing units. The neural network draws from the parallel processing of information, which is the strength of this method.

**Rcode:**

#Neuralnet classifier

#Implementing Nnet Classifier

nn\_ntree<- nnet(Class ~ ., data=training\_set, size=1)

#Predicting target class for the Validation set

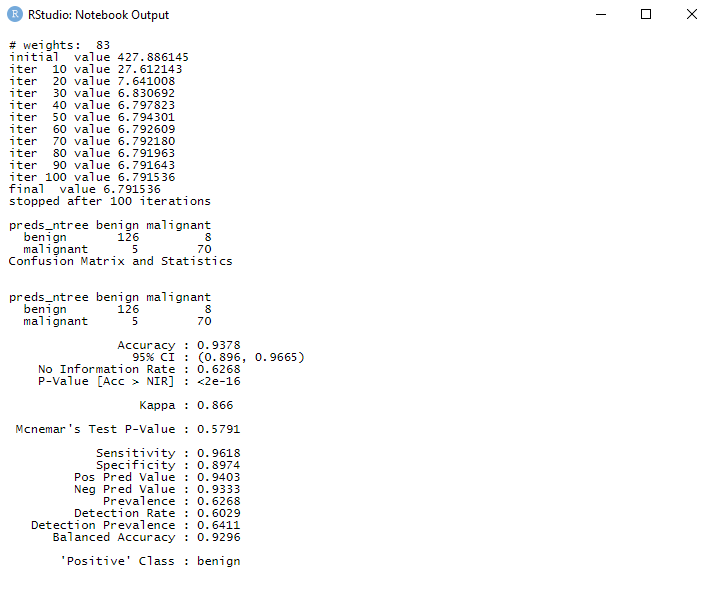
preds\_ntree <- predict(nn\_ntree,newdata=topredict\_set, type = "class")

(conf\_matrix\_ntree <- table(preds\_ntree, test\_set$Class))

#Confusion matrix for finding Accuracy of the model

confusionMatrix(conf\_matrix\_ntree)

**Output:**



The confusion matrix shows that neuralnet classifier predicted 126 benign cases correctly and 8 wrong predictions. Similarly, the model predicted 70 malignant cases correctly and 5 wrong predictions.

**The model accuracy is presented at 93.78%.**

**Conditional Inference Tree:**

Conditional Inference Trees is a different kind of decision tree that uses recursive partitioning of dependent variables based on the value of correlations. It avoids biasing just like other algorithms of classification and regression in machine learning.

**Rcode:**

# create model using conditional inference trees

require(party)

model\_ct <- ctree(Class ~ ., data=training\_set)

x.ct.pred <- predict(model\_ct, newdata=topredict\_set)

x.ct.prob <- 1- unlist(treeresponse(model\_ct, topredict\_set), use.names=F)[seq(1,nrow(topredict\_set)\*2,2)]

(conf\_matrix\_ct <- table(x.ct.pred, test\_set$Class))

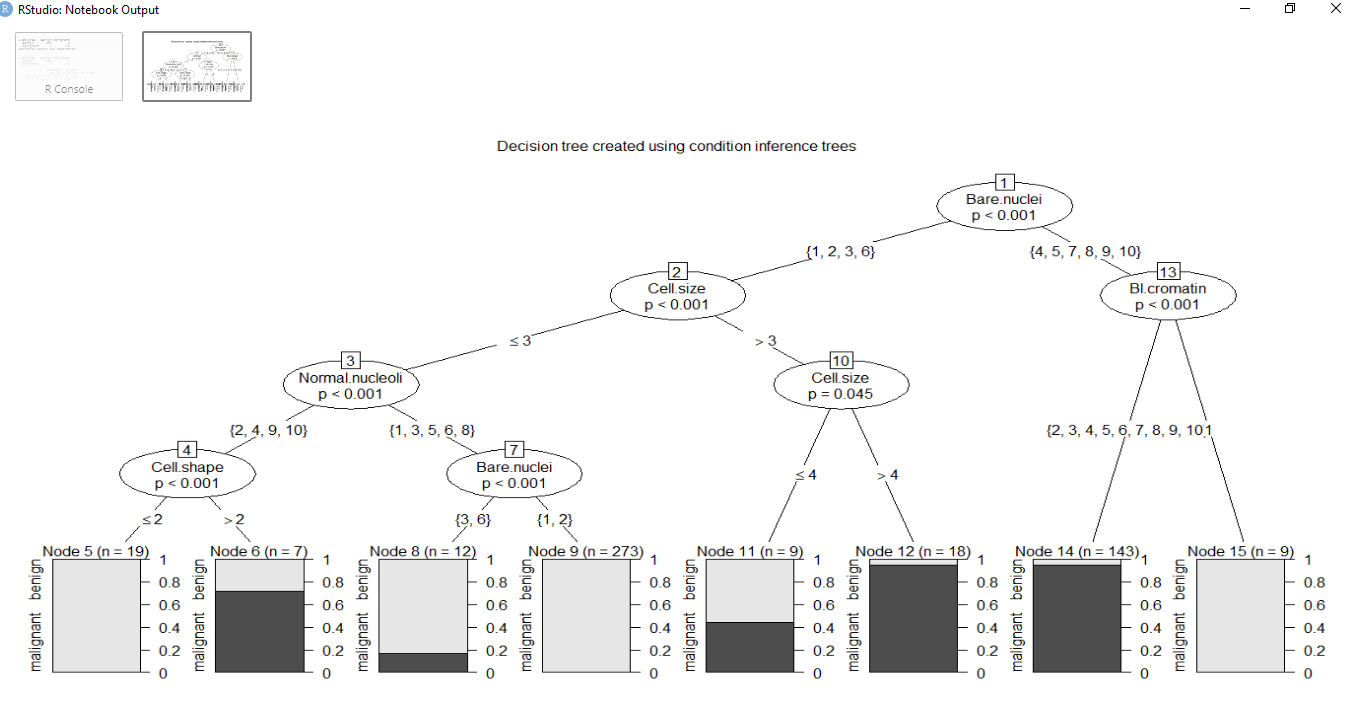
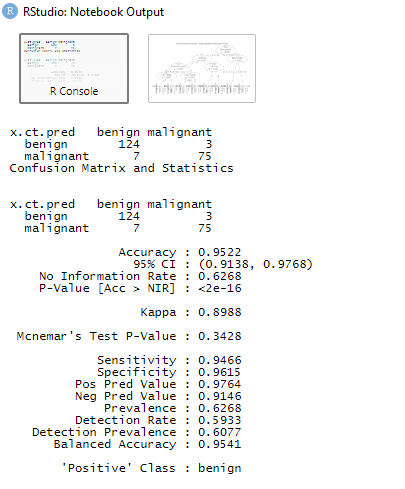
#Confusion matrix for finding Accuracy of the model

confusionMatrix(conf\_matrix\_ct)

# To view the decision tree, uncomment this line.

plot(model\_ct, main="Decision tree created using condition inference trees")

**Output:**



The confusion matrix shows that condition inference tree classifier predicted 124 benign cases correctly and 3 wrong predictions. Similarly, the model predicted 75 malignant cases correctly and 7 wrong predictions.

**The model accuracy is presented at 95.2%.**

**Support Vector Machine:**

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

**Rcode:**

## create model using svm (support vector machine)

require(e1071)

# svm requires tuning

x.svm.tune <- tune(svm, Class~., data = training\_set,

ranges = list(gamma = 2^(-8:1), cost = 2^(0:4)),

tunecontrol = tune.control(sampling = "fix"))

# display the tuning results (in text format)

x.svm.tune

# If the tuning results are on the margin of the parameters (e.g., gamma = 2^-8),

# then widen the parameters.

# I manually copied the cost and gamma from console messages above to parameters below.

x.svm <- svm(Class~., data = training\_set, cost=4, gamma=0.0625, probability = TRUE)

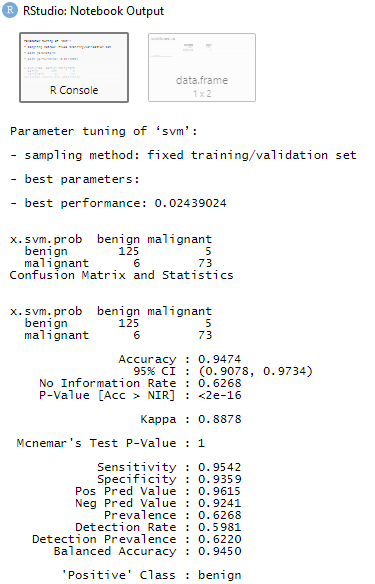
x.svm.prob <- predict(x.svm, type="prob", newdata=topredict\_set, probability = TRUE)

(conf\_matrix\_svm <- table(x.svm.prob, test\_set$Class))

#Confusion matrix for finding Accuracy of the model

confusionMatrix (conf\_matrix\_svm)

**Output:**



The confusion matrix shows that support vector machine classifier predicted 125 benign cases correctly and 5 wrong predictions. Similarly, the model predicted 73 malignant cases correctly and 6 wrong predictions.

**The model accuracy is presented at 94.74%.**

**Step 5:**

In this step, we will combine the above classifiers.

**Rcode:**

#combining classifiers

combine.classes<-data.frame(preds\_rf,preds\_dtree,preds\_ntree,x.svm.prob, x.ct.pred, preds\_naive)

head(combine.classes)

head(preds\_rf)

#head(myrda.pred)

combine.classes$preds\_rf<-ifelse(combine.classes$preds\_rf=="benign", 0, 1)

combine.classes[,2]<-ifelse(combine.classes[,2]=="benign", 0, 1)

combine.classes[,3]<-ifelse(combine.classes[,3]=="benign", 0, 1)

combine.classes[,4]<-ifelse(combine.classes[,4]=="benign", 0, 1)

combine.classes[,5]<-ifelse(combine.classes[,5]=="benign", 0, 1)

combine.classes[,6]<-ifelse(combine.classes[,6]=="benign", 0, 1)

str(combine.classes)

combine.cl<-combine.classes[, -c(7,8)]

majority.vote=rowSums(combine.classes[,-c(7,8)])

head(majority.vote)

#combine.classes[,7]<-rowSums(combine.classes[,-c(7,8)])

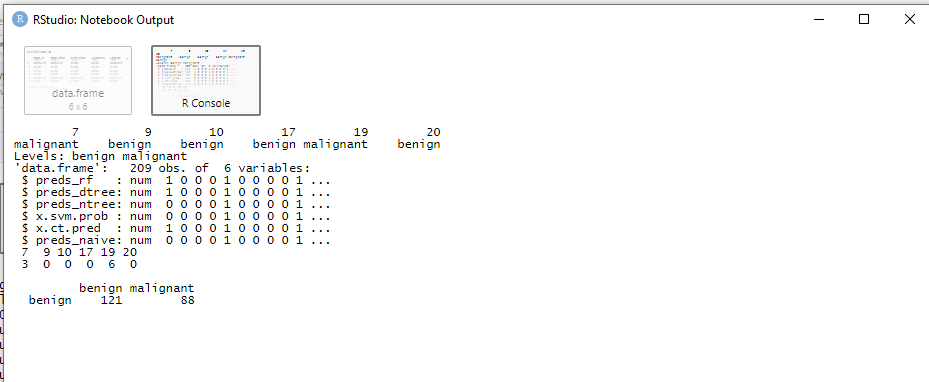
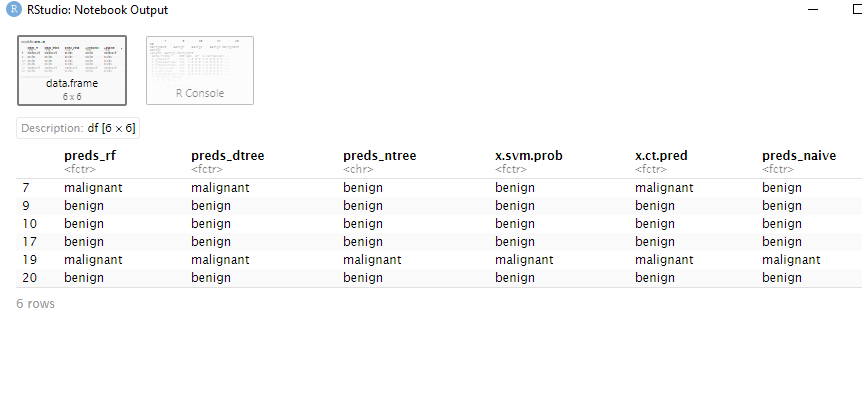
# Subset the BreastCancer data frame to only include the rows corresponding to the combine.classes data frame

breast\_cancer\_subset <- BreastCancer[1:nrow(combine.classes),]

combine.classes[,6]<-ifelse(combine.classes[,6]>=4, "malignant", "benign")

table(combine.classes[,6], breast\_cancer\_subset$Class)

**Output:**



**Conclusion:**

Comparing the above classifiers, we can conclude the Naïve Bayes Classification is the most suited classifier to predict if a patient has a malignant tumor as it has the highest accuracy.