mydata <-read.csv(file = "H:\\LP1\\diabetes.csv", header = TRUE, sep=",")

#the above reads the file diabetes.csv into a data frame that it creates called MyData. header=TRUE specifies that this data includes a header row and sep=”,” specifies that the data is separated by commas

View(mydata)

library(caTools)

# Contains several basic utility functions including: moving (rolling, running) window statistic functions, read/write for GIF and ENVI binary files, fast calculation of AUC, LogitBoost classifier, base64 encoder/decoder, round-off-error-free sum and cumsum, etc.

library(e1071)

# Functions for latent class analysis, short time Fourier transform, fuzzy clustering, support vector machines, shortest path computation, bagged clustering, naive Bayes classifier

# Part-1 caTools

temp\_field <- sample.split(mydata,SplitRatio=0.7)

#70% willbe in training

train <- subset(mydata, temp\_field==TRUE)

# 30 % will be in testing

# Return subsets of vectors, matrices or data frames which meet conditions.

test <- subset(mydata, temp\_field==FALSE)

# display few samples that are used for training & testing

head(train)

# Returns the first or last parts of a vector, matrix, table, data frame or function. Since head() and tail() are generic functions, they may also have been extended to other classes. (first 6 enties)

head(test)

# Part-2 e1071

# building naive Bayes

my\_model <-naiveBayes(as.factor(train$Outcome)~.,train)

# Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.

# Outcome is the response variable as a function of all other variables so use ~.

# for developing the model we use train data set

# To see summaries of the probabilities calculated

# displays A-priori probabilities: (i.e it displays probability of having diabetes and probability of not having diabetes alongwith mean and standard deviation for each attribute with respect to 0 and 1.

my\_model

#predicting probabilities, for my\_model and type=class oy raw after the test data

pred1<-predict(my\_model,test[,-9])

pred1

pred1<-predict(my\_model,test[,-9],type="class")

pred1

# character string denoting the type of predicted value returned. If type = "class": (for a classification tree) a factor of classifications based on the responses.

pred1<-predict(my\_model,test[,-9],type="raw")

pred1

# If "raw", the conditional a-posterior probabilities for each class are returned

#generate the confusion matrix

pred1<-predict(my\_model,test[,-9])

pred1

table(pred1,test$Outcome,dnn = c("predicted","Actual"))

#Calculates a cross-tabulation of observed and predicted classes with associated statistics. **dnn**

a character vector of dimnames for the table

# to save the prediction

output <- cbind(test,pred1)

# Take a sequence of vector, matrix or data-frame arguments and combine by *c*olumns or *r*ows, respectively.

View(output)