**Assignment No. 12**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Problem Statement :** Download Pima Indians Diabetes dataset. Use Naive Bayes‟ Algorithm

for classification

* Load the data from CSV file and split it into training and test datasets.
* Summarize the properties in the training dataset so that we can calculate probabilities and make predictions.

Classify samples from a test dataset and a summarized training dataset.

#....................part 1................................

#prog2 : using diabetes data with naive bayes

library(caTools)

library(e1071)

#import dataset

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

View(mydata)

#..........part1.............

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

#70% will b in training

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

#30% will be in testing

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

#displays few samples that are used for training and testing

head(train)

head(test)

#....................part 2................................

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

my\_model

#....................part 3 ...............................

#predicting, try putting type="class" or type="raw" after the test data

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

pred1

#generate the confusion matrix

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

#To save the prediction

output<-cbind(test, pred1)

View(output)

OUTPUT:

|  |
| --- |
| R version 3.5.1 (2018-07-02) -- "Feather Spray"  Copyright (C) 2018 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.  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 D:/.RData]  > #....................part 1................................  > #prog2 : using diabetes data with naive bayes  > library(caTools)  > library(e1071)  > #prog2 : using diabetes data with naive bayes  > library(caTools)  > library(e1071)  > #import dataset  > mydata<-read.csv(file="D:\\diabetes.csv",header=TRUE,sep=",")  > View(mydata)  > library(caTools)  > library(e1071)  > #import dataset  > mydata<-read.csv(file="D:\\diabetes.csv",header=TRUE,sep=",")  > View(mydata)  > #..........part1.............  > temp\_field<-sample.split(mydata,SplitRatio=0.7)  > #70% will b in training  > train<-subset(mydata, temp\_field==TRUE)  > #30% will be in testing  > test<-subset(mydata, temp\_field == FALSE)  >  > #displays few samples that are used for training and testing  > head(train)  Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome  2 1 85 66 29 0 26.6 0.351 31 0  4 1 89 66 23 94 28.1 0.167 21 0  5 0 137 40 35 168 43.1 2.288 33 1  7 3 78 50 32 88 31.0 0.248 26 1  8 10 115 0 0 0 35.3 0.134 29 0  9 2 197 70 45 543 30.5 0.158 53 1  > head(test)  Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome  1 6 148 72 35 0 33.6 0.627 50 1  3 8 183 64 0 0 23.3 0.672 32 1  6 5 116 74 0 0 25.6 0.201 30 0  10 8 125 96 0 0 0.0 0.232 54 1  12 10 168 74 0 0 38.0 0.537 34 1  15 5 166 72 19 175 25.8 0.587 51 1  > pred1<-predict(my\_model,test[,-9])  Error in predict(my\_model, test[, -9]) : object 'my\_model' not found  > pred1  Error: object 'pred1' not found  > #....................part 1................................  > #prog2 : using diabetes data with naive bayes  > library(caTools)  > library(e1071)  > #import dataset  > mydata<-read.csv(file="D:\\diabetes.csv",header=TRUE,sep=",")  > View(mydata)  > #..........part1.............  > temp\_field<-sample.split(mydata,SplitRatio=0.7)  > #70% will b in training  > train<-subset(mydata, temp\_field==TRUE)  > #30% will be in testing  > test<-subset(mydata, temp\_field == FALSE)  >  > #displays few samples that are used for training and testing  > head(train)  Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome  2 1 85 66 29 0 26.6 0.351 31 0  4 1 89 66 23 94 28.1 0.167 21 0  6 5 116 74 0 0 25.6 0.201 30 0  7 3 78 50 32 88 31.0 0.248 26 1  8 10 115 0 0 0 35.3 0.134 29 0  9 2 197 70 45 543 30.5 0.158 53 1  > head(test)  Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome  1 6 148 72 35 0 33.6 0.627 50 1  3 8 183 64 0 0 23.3 0.672 32 1  5 0 137 40 35 168 43.1 2.288 33 1  10 8 125 96 0 0 0.0 0.232 54 1  12 10 168 74 0 0 38.0 0.537 34 1  14 1 189 60 23 846 30.1 0.398 59 1  >  > #....................part 2................................  > my\_model<-naiveBayes(as.factor(train$Outcome)~.,train)  > my\_model  Naive Bayes Classifier for Discrete Predictors  Call:  naiveBayes.default(x = X, y = Y, laplace = laplace)  A-priori probabilities:  Y  0 1  0.6457926 0.3542074  Conditional probabilities:  Pregnancies  Y [,1] [,2]  0 3.400000 3.083045  1 4.779006 3.858008  Glucose  Y [,1] [,2]  0 111.0182 25.53012  1 139.0387 32.21910  BloodPressure  Y [,1] [,2]  0 68.43333 18.02658  1 70.77348 21.66022  SkinThickness  Y [,1] [,2]  0 19.40909 14.90739  1 22.01657 17.89552  Insulin  Y [,1] [,2]  0 68.95758 96.20099  1 97.67956 133.66196  BMI  Y [,1] [,2]  0 30.18879 7.171352  1 35.44807 6.614702  DiabetesPedigreeFunction  Y [,1] [,2]  0 0.4188667 0.2994131  1 0.5294254 0.3259863  Age  Y [,1] [,2]  0 31.56364 11.82409  1 37.29282 11.12741  >  > #....................part 3 ...............................  > #predicting, try putting type="class" or type="raw" after the test data  > pred1<-predict(my\_model,test[,-9])  > pred1  [1] 1 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1  [53] 1 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1 0 0 1 0 0 0 1 0  [105] 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 0  [157] 1 1 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 0  [209] 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 1 1 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0  Levels: 0 1  > #....................part 1................................  > #prog2 : using diabetes data with naive bayes  > library(caTools)  > library(e1071)  > #import dataset  > mydata<-read.csv(file="D:\\diabetes.csv",header=TRUE,sep=",")  > View(mydata)  > #..........part1.............  > temp\_field<-sample.split(mydata,SplitRatio=0.7)  > #70% will b in training  > train<-subset(mydata, temp\_field==TRUE)  > #30% will be in testing  > test<-subset(mydata, temp\_field == FALSE)  >  > #displays few samples that are used for training and testing  > head(train)  Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome  1 6 148 72 35 0 33.6 0.627 50 1  3 8 183 64 0 0 23.3 0.672 32 1  4 1 89 66 23 94 28.1 0.167 21 0  5 0 137 40 35 168 43.1 2.288 33 1  6 5 116 74 0 0 25.6 0.201 30 0  8 10 115 0 0 0 35.3 0.134 29 0  > head(test)  Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome  2 1 85 66 29 0 26.6 0.351 31 0  7 3 78 50 32 88 31.0 0.248 26 1  9 2 197 70 45 543 30.5 0.158 53 1  11 4 110 92 0 0 37.6 0.191 30 0  16 7 100 0 0 0 30.0 0.484 32 1  18 7 107 74 0 0 29.6 0.254 31 1  >  > #....................part 2................................  > my\_model<-naiveBayes(as.factor(train$Outcome)~.,train)  > my\_model  Naive Bayes Classifier for Discrete Predictors  Call:  naiveBayes.default(x = X, y = Y, laplace = laplace)  A-priori probabilities:  Y  0 1  0.6679688 0.3320312  Conditional probabilities:  Pregnancies  Y [,1] [,2]  0 3.298246 3.024620  1 4.794118 3.655096  Glucose  Y [,1] [,2]  0 108.6637 26.40650  1 142.6882 33.28508  BloodPressure  Y [,1] [,2]  0 67.80117 18.65841  1 70.95882 21.63149  SkinThickness  Y [,1] [,2]  0 19.36550 15.02640  1 22.08235 18.08148  Insulin  Y [,1] [,2]  0 68.25146 104.2588  1 93.49412 133.5078  BMI  Y [,1] [,2]  0 30.39357 7.763375  1 34.78765 7.342284  DiabetesPedigreeFunction  Y [,1] [,2]  0 0.4341930 0.2984265  1 0.5830824 0.3974000  Age  Y [,1] [,2]  0 30.95322 11.46782  1 37.24706 11.33556  >  > #....................part 3 ...............................  > #predicting, try putting type="class" or type="raw" after the test data  > pred1<-predict(my\_model,test[,-9])  > pred1  [1] 0 0 1 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1  [53] 1 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0  [105] 1 0 0 1 0 1 1 0 1 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0  [157] 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 0 1 0 1 0 1 0 1  [209] 0 0 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0  Levels: 0 1  >  > #generate the confusion matrix  > table(pred1, test$Outcome, dnn=c("predicted", "Actual"))  Actual  predicted 0 1  0 138 43  1 20 55  >  > #To save the prediction  > output<-cbind(test, pred1)  > View(output) |