DATA VISUALIZATION OF DIABETES DATA SET AND DETERMINING THE BEST TECHNIQUE FOR ITS PREDICTION

FINAL REVIEW REPORT

Submitted by

Urja Mehta (15BCE0309)

Tanvi Pareek (15BCE0764)

Anisha Gupta (15BCE2078)

Prepared For

MACHINE LEARNING (CSE4020) – PROJECT COMPONENT

Submitted To

Vijayasherly V

School of Computer Science and Engineering

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ■ CHENNAI
www.vit.ac.in

1) Table of Contents

2. ABSTRACT

Machine learning techniques have been extensively applied in bioinformatics to analyze biomedical data. In this project, we choose the R programing as our tool to analyze a Diabetes Data Set, which collects the information of patients with and without developing diabetes. The discussion follows the machine learning process. The focus will be on the data preprocessing, including attribute identification and selection, outlier removal, data normalization and numerical discretization, visual data analysis, hidden relationships discovery, and a diabetes prediction model construction.

3. INTRODUCTION

In machine learning we can extract hidden knowledge from large volumes of raw data. It is one of the tasks in the process of knowledge discovery from the database. The knowledge must be new, not obvious, and one must be able to use it. It has been defined as “the nontrivial extraction of previously unknown, implicit and potentially useful information from data. It is “the science of extracting useful information from large databases”. It is one of the tasks in the process of knowledge discovery from the database.
Machine learning technique is used to discover knowledge out of data and presenting it in a form that is easily understand to humans. It is a process to examine large amounts of data routinely collected. Machine learning is most useful in an exploratory analysis because of nontrivial information in large volumes of data. It is a cooperative effort of humans and computers. Best results are achieved by balancing the knowledge of human experts in describing problems and goals with the search capabilities of computers. There are two primary goals of data mining tend to be prediction and description. Prediction involves some variables or fields in the data set to predict unknown or future values of other variables of interest. On the other hand Description focuses on finding patterns describing the data that can be interpreted by humans.

The Disease Prediction plays an important role in machine learning. This paper analyzes Diabetes disease predictions.

4. PROJECT RESOURCE REQUIREMENTS

4.1 Software Resource Requirements

- R script module

5. STEPS TO BE FOLLOWED: DATA ANALYSIS

5.1 Data Preprocessing

Most of the data sets used in process were not necessarily gathered with a specific goal in mind. Some of them may contain errors, outliers or missing values. In order to use those data sets in the mining process, the data needs to undergo preprocessing, using data cleaning, normalization and data transformation. It has been estimated that data preparation alone accounts for 60% of all the time and effort expanded in the entire mining process.

5.2 Feature Identification and categorization

Attributes are usually described by a set of corresponding values. Features described by both numerical and symbolic values can be either discrete (categorical) or continuous. Discrete features concern a situation in which the total number of values is relatively small (finite), while with continuous features the total number of values is very large (infinite) and covers a specific interval (range). The following attributes can be gathered from the data set.

- Cholesterol
- Hdl
- Bp1
- Stab.glucose
- Pregnant: Number of times of pregnant
- Plasma-Glucose: Plasma glucose concentration
- tolerance test. Blood sugar level.
- DiastolicBP: Diastolic blood pressure (mmHg)
- TricepsSFT: Triceps skin fold thickness (mm)
- Serum-Insulin: 2-hour serum insulin (mu U/mt)
- BMI: Body mass index (w in kg/h in m)
- DPF: Diabetes pedigree function
- Age: Age of the patient (years)
- Class: Diabetes onset within five years (0 or 1)

These characteristics need to be kept in mind as the data set is cleaned. Most of this work can be done in R script itself. After importing samples of the Data Set, changing default attribute titles, and renaming the values of attribute Class from (0, 1) to (No, Yes), one can obtain a proper categorized attribute table.

5.3 Empty Values removal and feature selection

Using R script, the attributes can be sorted in different ways to view patterns and values. Those rows with missing value should be removed for the attributes where values are absent. We can observe various graphs after removal of these attributes and selecting required features. R script can also be used to view the data using different plotters. We also plot graph for correlation of each attribute against every other attribute. The attributes could be removed based on the number of missing values and its relationship to other attributes. The Histogram Plotter can be used to check the rest of the attributes for correlations.

6. PREDICTION TECHNIQUES USED

We compare the performance for the following classifiers:

1. Logistic Regression

It is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Mathematically, logistic regression estimates a multiple linear regression function defined as:

$$\text{logit}(p) = \log\left(\frac{p(y=1)}{1-(p=1)}\right) = \beta_0 + \beta_1 x_{i2} + \beta_2 \cdot x_{i2} + \_ + \beta_p \cdot x_{in}$$

for i = 1…n .

2. Support Vector Machine (SVM)

Support Vector Machine" (SVM) is a supervised machine-learning algorithm, which can be, used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well.

3. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results.

7. CODE FOR VISUALIZATION (3 METHODS) TO PRINT ACCURACY

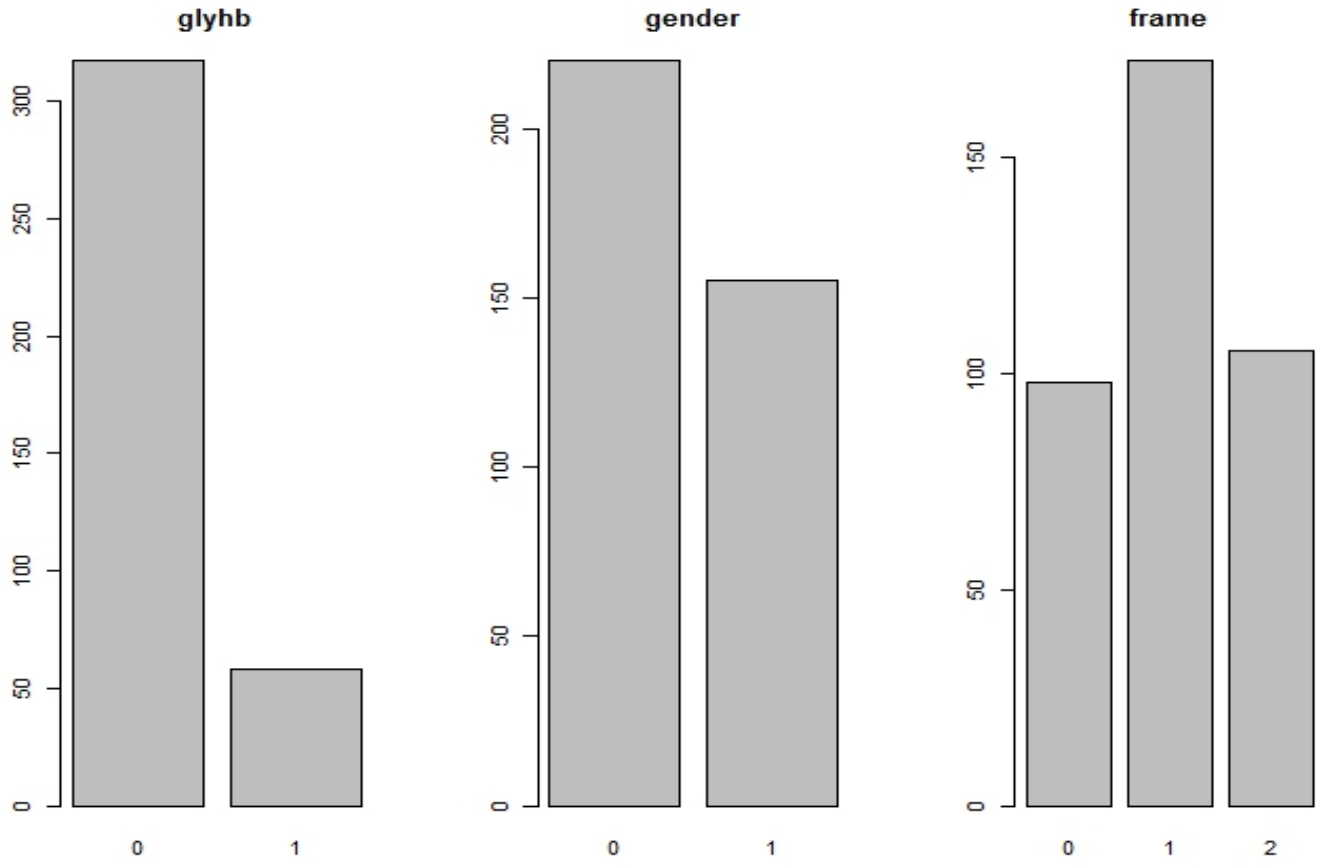
```
# Load
libraries
            library(randomForest)
            library(caret)

            # load the data
            filename="C:\\data\\diabetes_data.csv"
            datasetRaw = read.csv(filename)
            print(head(datasetRaw))

            # clean the data
            numColumns = dim(datasetRaw)[2]
            vector_NAs = rep(0, numColumns)
            for (i in 1:numColumns) {
              vector_NAs[i] = sum(is.na(datasetRaw[,i]))
```

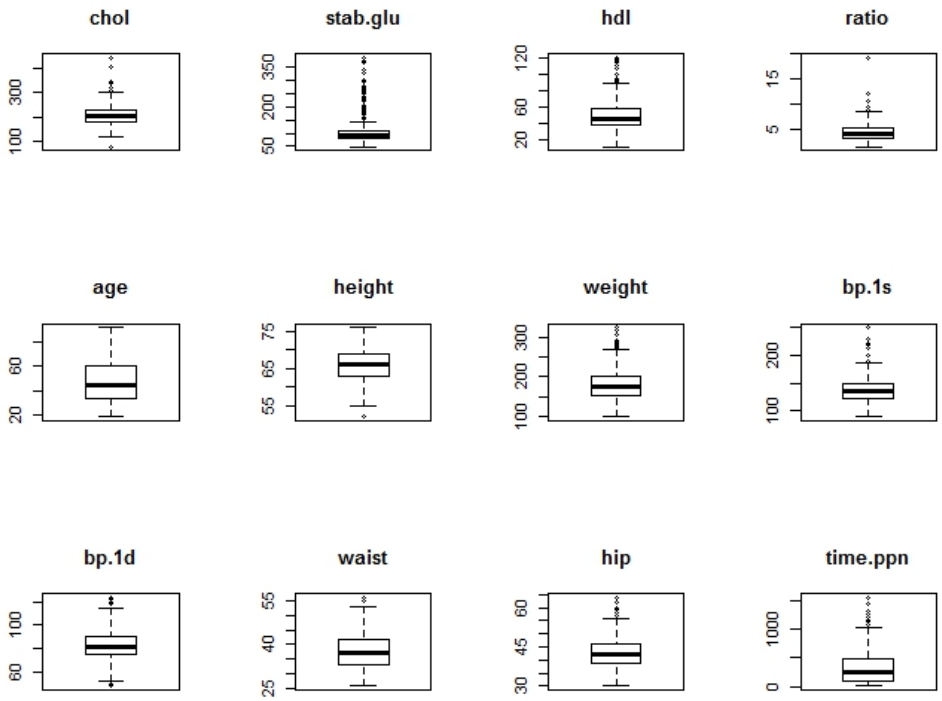
```
}
print("The missing values in each column:")
print(vector_NAs)

# delete columns 15 and 16 due to many missing values
# delete column 1 (id), column 7 (location) because they contain no useful information
dataset = datasetRaw[,-c(1,7,15,16)]
print(dim(dataset))

# remove the row with missing values
row.has.na <- apply(dataset, 1, function(x){any(is.na(x))})
dataset = dataset[!row.has.na,]
print(dim(dataset))
print(head(dataset))

# encode the class label (column 5): Glycosolated hemoglobin  > 7.0 is taken as a
positive diagnosis of diabetes.
dataset[,5] = ifelse(dataset[,5] >= 7.0, 1, 0)
dataset[,5] = factor(dataset[,5]) # class label must be factor type

# encode the categorical data (column-7 gender)
dataset[,7] = ifelse(dataset[,7] == "female", 0, 1)
dataset[,7] = factor(dataset[,7])
# encode the categorical data (column-10 frame)
dataset[,10] = ifelse(dataset[,10] == "small", 0, ifelse(dataset[,10] == "medium", 1,2) )
dataset[,10] = factor(dataset[,10])

# split the data into training and validation sets
set.seed(7)
validation_index = createDataPartition(dataset$glyhb, p=0.90, list=FALSE)
validationData = dataset[-validation_index,]
trainingData = dataset[validation_index,]

# comparison among different classifiers
#1. Logistic Regression
set.seed(7)
control.glm = trainControl(method = "cv", number = 5)
fit.glm = train(glyhb~., data = trainingData, method = "glm", preProc =
c("center","scale"), trControl = control.glm)
print(fit.glm$results) # accuracy = 0.9172205

#2. Support Vector Machine
set.seed(7)
control.svmRadial = trainControl(method="cv", number=5)
```

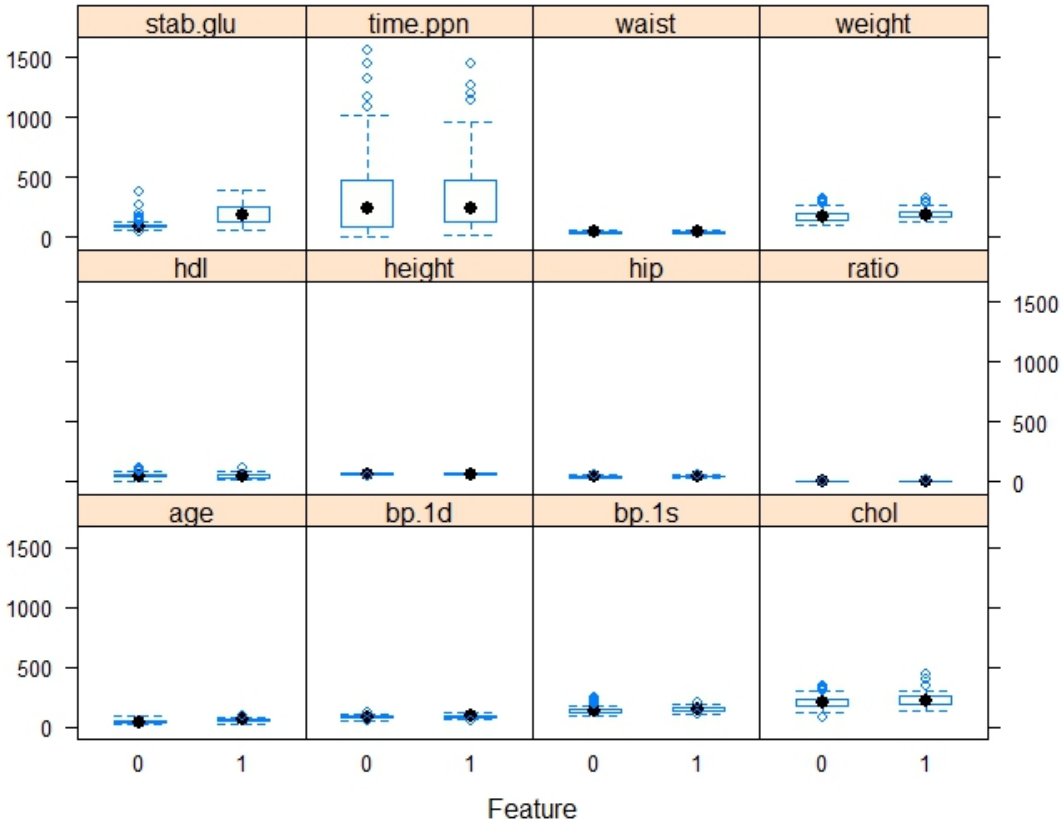
```
fit.svmRadial <- train(glyhb~., data=trainingData, method="svmRadial", metric="Accuracy",
preProc=c("center","scale"), trControl=control.svmRadial)
# summarize fit
print(fit.svmRadial$results) #accuracy = 0.9231731

#3. random forest
control.rf = trainControl(method="cv", number=5)
set.seed(7)
metric = "Accuracy"
mtry = 7 # mtry=7 (number of variables to try)
tunegrid <- expand.grid(.mtry=mtry)
fit.rf_default <- train(glyhb~., data=trainingData, method="rf", metric=metric,
tuneGrid=tunegrid, preProc=c("center","scale"), trControl=control)
print(fit.rf_default$results) # accuracy = 0.9114924

#4. parameter tunning via grid search for random forest
control.rf_search <- trainControl(method="repeatedcv", number=5, repeats=3,
search="grid")
set.seed(7)
tunegrid <- expand.grid(.mtry=c(1:15))
fit.rf_gridsearch <- train(glyhb~., data=trainingData, method="rf", metric=metric,
tuneGrid=tunegrid, trControl=control.rf_search, ntree=1000)
print(fit.rf_gridsearch) # accuracy = 0.9204355 when mtry = 12
print(fit.rf_gridsearch$finalModel)
plot(fit.rf_gridsearch)

# make predictions on the validation set
set.seed(7)
predictions = predict(fit.rf_gridsearch, newdata=validationData)
confusionMatrix = confusionMatrix(predictions, validationData$glyhb)
# confusion matrix
print(confusionMatrix$table)

# save the final classifier model into disk
saveRDS(fit.rf_gridsearch, "C:\\data\\diabetes_classification")

# load the model from the disk
final_model <- readRDS("C:\\data\\diabetes_classification")
print(final_model)
# make predictions using the loaded model
set.seed(7)
predictions = predict(final_model, newdata=validationData)
```

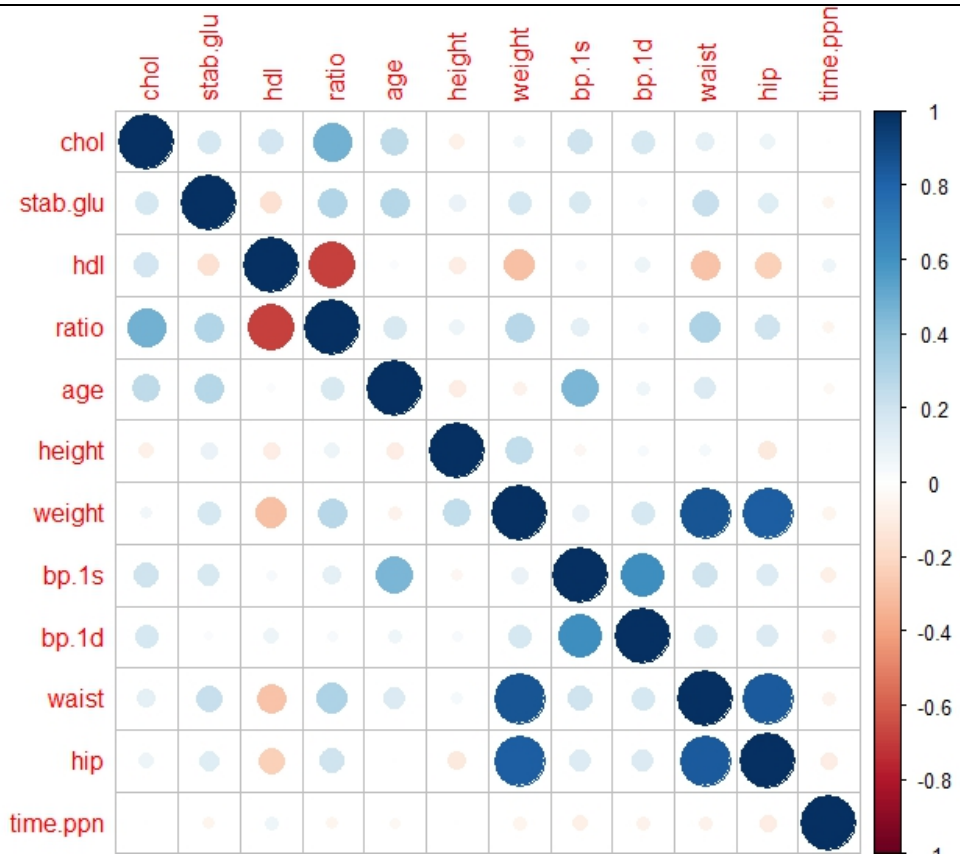
8. OUTPUT

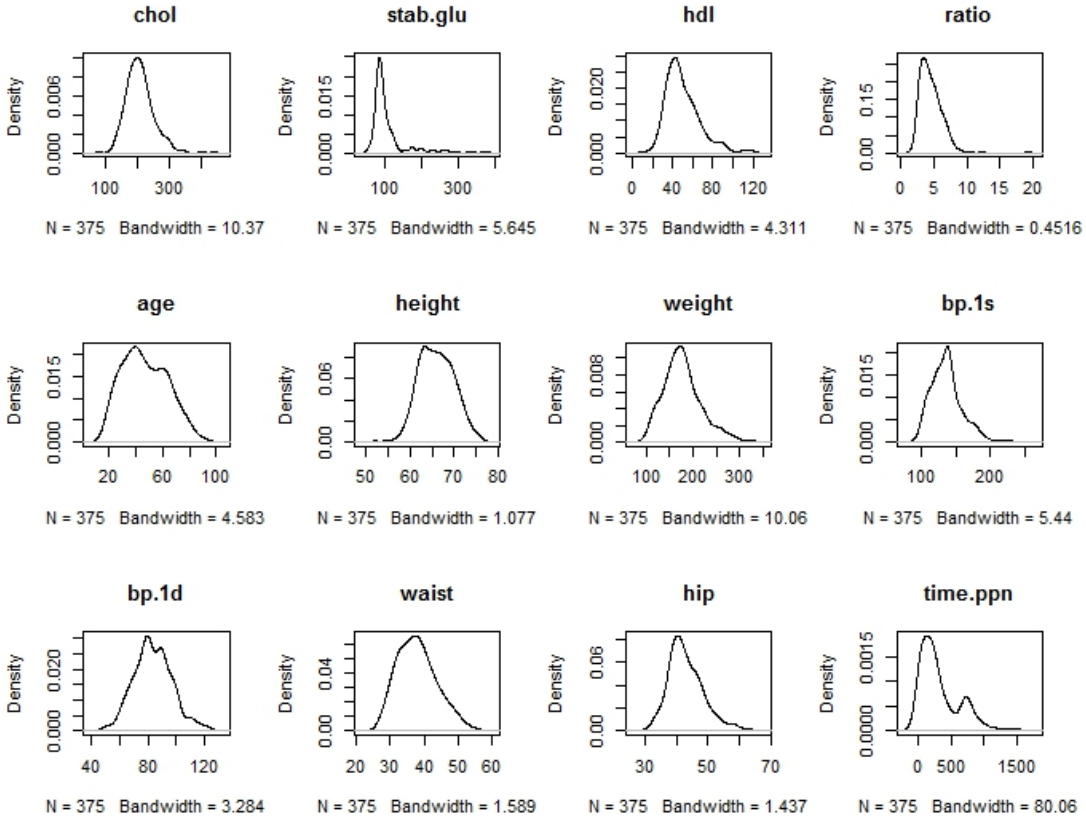
- Data visualization
- Histogram plot
- Density plot
- Box and Whisker plot
- Bar plot
- Missing data map
- Pair-wise correlation plot

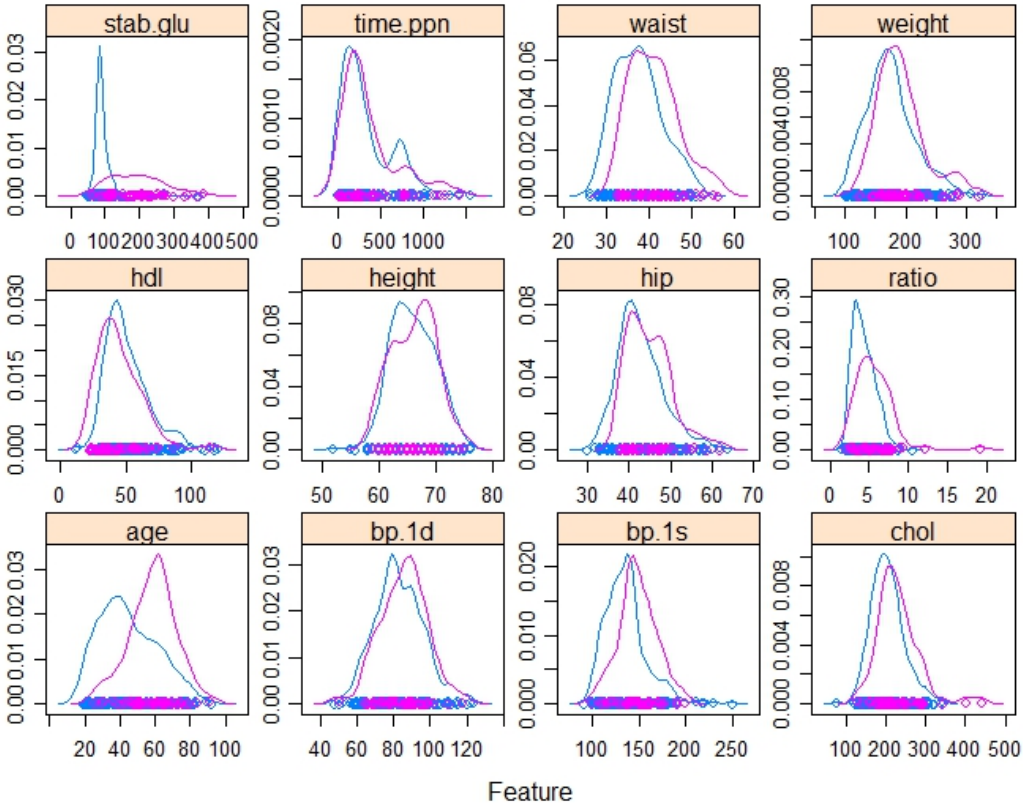


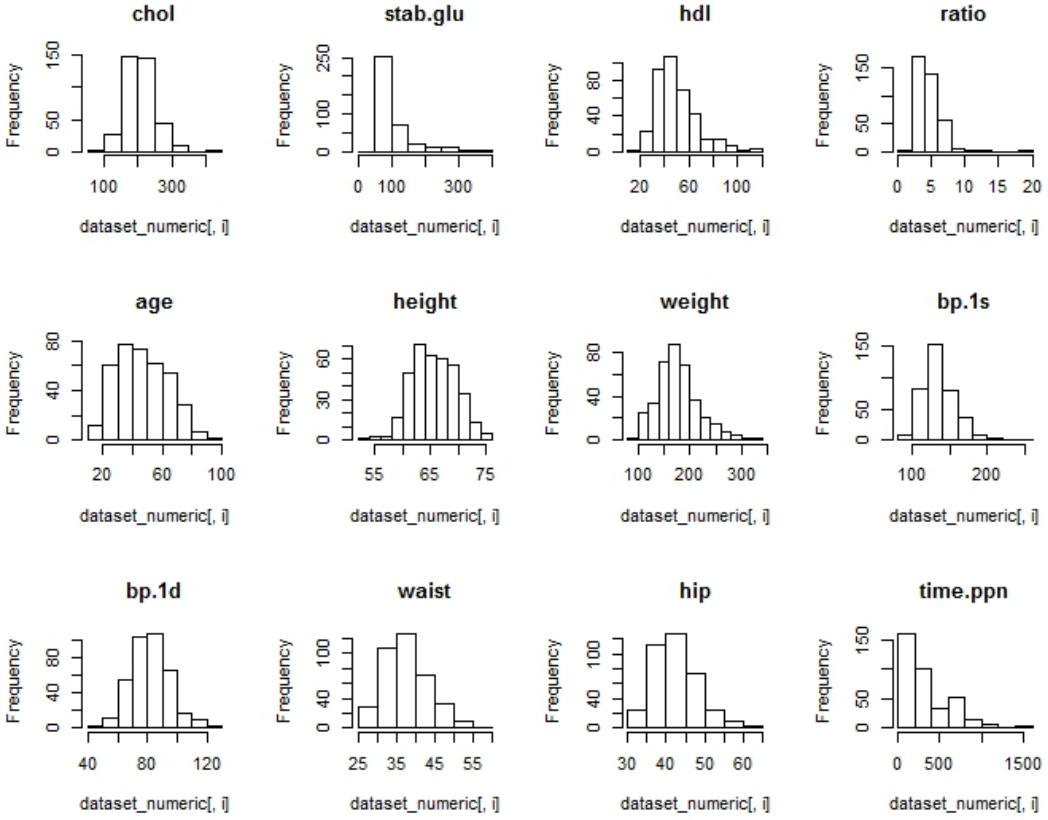

chol
stab.glu
hdl
ratio
age
height
weight
bp.1s
bp.1d
waist
hip
time.ppn

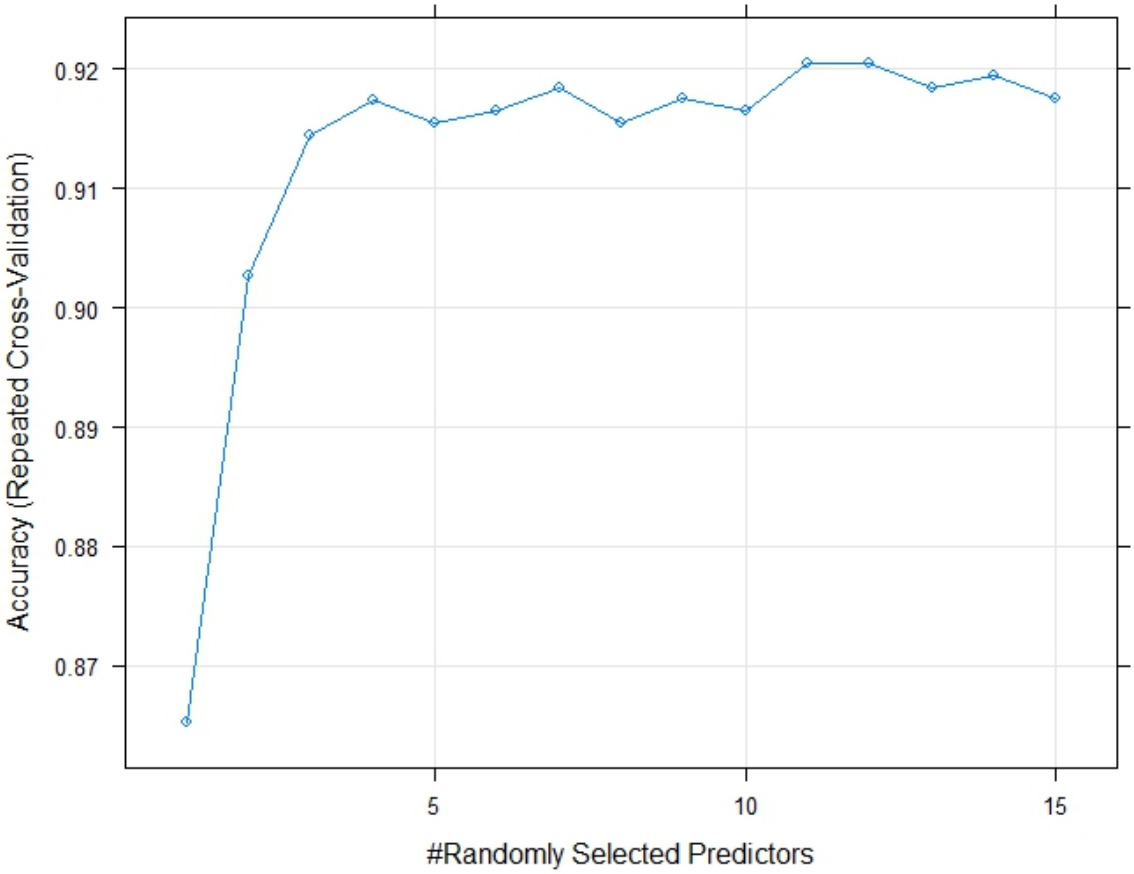

stab.glu
time.ppn
waist
weight
hdl
height
hip
ratio
age
bp.1d
bp.1s
chol
Feature

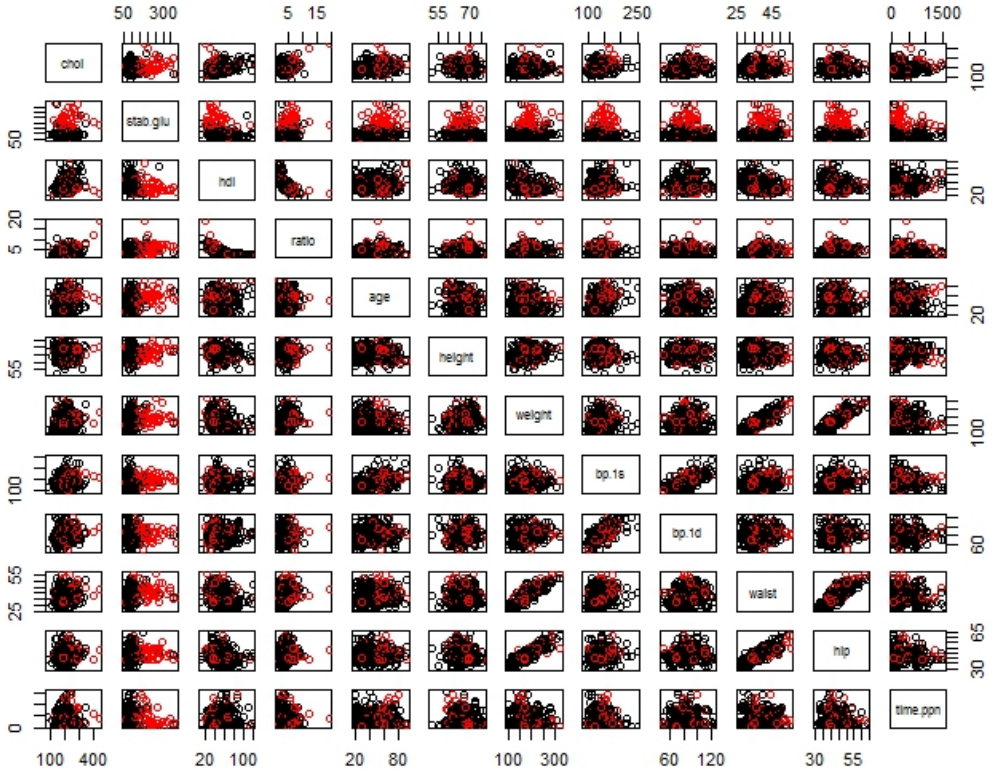

chol
stab.glu
hdl
ratio
age
height
weight
bp.1s
bp.1d
waist
hip
time.ppn
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8

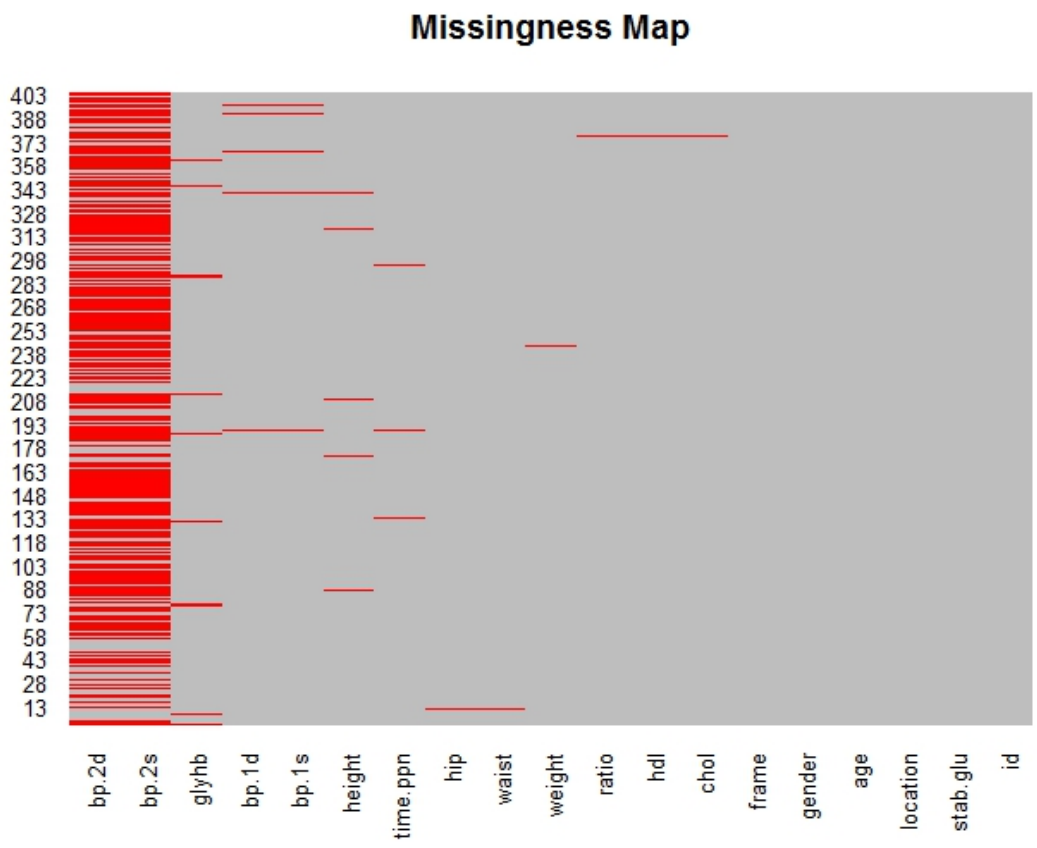

chol
Density
N = 375 Bandwidth = 10.37
stab.glu
N = 375 Bandwidth = 5.645
hdl
N = 375 Bandwidth = 4.311
ratio
N = 375 Bandwidth = 0.4516
age
N = 375 Bandwidth = 4.583
height
N = 375 Bandwidth = 1.077
weight
N = 375 Bandwidth = 10.06
bp.1s
N = 375 Bandwidth = 5.44
bp.1d
N = 375 Bandwidth = 3.284
waist
N = 375 Bandwidth = 1.589
hip
N = 375 Bandwidth = 1.437
time.ppn
N = 375 Bandwidth = 80.06


stab.glu
time.ppn
waist
weight
hdl
height
hip
ratio
age
bp.1d
bp.1s
chol
Feature


chol
stab.glu
hdl
ratio
age
height
weight
bp.1s
bp.1d
waist
hip
time.ppn
Frequency
dataset_numeric[, i]


Accuracy (Repeated Cross-Validation)
0.92
0.91
0.90
0.89
0.88
0.87
5
10
15
#Randomly Selected Predictors


chol
stab.glu
hdl
ratio
age
height
weight
bp.1s
bp.1d
waist
hip
time.ppn



9. CONCLUSION

CLASSIFICATION TECHNIQUE	ACCURACY
LOGISTIC REGRESSION	0.9172205
SVM	0.9231731
RANDOM FOREST	0.9114929

After following the procedure and extracting information from the data set followed by the data analysis and prediction using 3 algorithms, we find out that Support Vector Machine (SVM) techniques predicts with the highest accuracy.