Capstone project, Heart Attack Prediction

Eyad Alkronz

May 20th, 2021

Table of Contents

Overview

This project is part of the HarvardX course PH125.9x Data Science: Capstone project. Heart disease, alternatively known as cardiovascular disease, encases various conditions that impact the heart and is the primary basis of death worldwide over the span of the past few decades. It associates many risk factors in heart disease and a need of the time to get accurate, reliable, and sensible approaches to make an early diagnosis to achieve prompt management of the disease. Data mining is a commonly used technique for processing enormous data in the healthcare domain. Researchers apply several data mining and machine learning techniques to analyses huge complex medical data, helping healthcare professionals to predict heart disease. This report presents various attributes related to heart disease, and the model on basis of supervised learning algorithms as decision tree, K-nearest neighbor, and random forest algorithm. It uses the existing dataset from the UCI Machine Learning Repository of heart disease patients and data available on this link https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset. The dataset comprises 303 instances and 76 attributes. Of these 76 attributes, only 14 attributes are considered for testing. This report aims to envision the probability of developing heart disease in the patients. results will be explained. Finally, the report ends with some concluding remarks.

Introduction

Over the last decade, heart disease or cardiovascular remains the primary basis of death worldwide. An estimate by the World Health Organization, that over 17.9 million deaths occur every year worldwide because of cardiovascular disease, and of these deaths, 80% are because of coronary artery disease and cerebral stroke. The vast number of deaths is common amongst low and middle-income countries. Many predisposing factors such as personal and professional habits and genetic predisposition accounts for heart disease. Various habitual risk factors such as smoking, overuse of alcohol and caffeine, stress, and physical inactivity along with other physiological factors like obesity, hypertension, high blood cholesterol, and pre-existing heart conditions are predisposing factors for heart disease. The efficient and accurate and early medical diagnosis of heart disease plays a crucial role in taking preventive measures to prevent death.

Data mining refers to the extraction of required information from huge datasets in various fields such as the medical field, business field, and educational field. Machine learning is one of the most rapidly evolving domains of artificial intelligence. These algorithms can analyze huge data from various fields, one such important field is the medical field. It is a substitute to routine prediction modeling approach using a computer to gain an understanding of complex and non-linear interactions among different factors by reducing the errors in predicted and factual outcomes. Data mining is exploring huge datasets to extract hidden crucial decision-making information from a collection of a past repository for future analysis. The medical field comprises tremendous data of patients. These data need mining by various machine learning algorithms. Healthcare professionals do analysis of these data to achieve effective diagnostic decision by healthcare professionals. Medical data mining using classification algorithms provides clinical aid through analysis. It tests the classification algorithms to predict heart disease in patients.

Data mining is the process of extracting valuable data and information from huge databases. Various data mining techniques such as regression, clustering, association rule and classification techniques like decision tree, random forest and K-nearest neighbor are used to classify various heart disease attributes in predicting heart disease. A comparative analysis of the classification techniques is used. In this report, I have taken dataset from the UCI repository. The classification model is developed using classification algorithms for prediction of heart disease.

Background

Heart disease affects millions of people, and it remains the chief cause of death in the world. Medical diagnosis should be proficient, reliable, and aided with computer techniques to reduce the effective cost for diagnostic tests. Data mining is a software technology that helps computers to build and classify various attributes. This report uses classification techniques to predict heart disease.

Machine Learning

Machine learning is an emerging subdivision of artificial intelligence. Its primary focus is to design systems, allow them to learn and make predictions based on the experience. It trains machine learning algorithms using a training dataset to create a model. The model uses the new input data to predict heart disease. Using machine learning, it detects hidden patterns in the input dataset to build models. It makes accurate predictions for new datasets. The model uses the new input data to predict heart disease and then tested for accuracy.

Classification Machine Learning Techniques

The classification task is used for prediction of subsequent cases dependent on past information. Many data mining techniques neural network, decision tree have been applied by researchers to have a precision diagnosis in heart disease. The accuracy given by different techniques varies with number of attributes. This research provides diagnostic accuracy score for improvement of better health results. We have used RStudio tool in this report for pre-processing the dataset, Only 14 attributes of all 76 different attributes have been considered for analysis to get precise results. By comparison and analysis using different algorithms with RStudio tool heart disease can be predicted and treated early and prompt.

The Data

The data used for the project is the Heart Attack Analysis & Prediction Dataset A dataset for heart attack classification, collected by UCI Machine Learning Repository and data avilable on this link https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

The data contains

Variable name	Description
age	Age of the person
sex	Gender of the person
cp	Chest Pain type chest pain type
trtbps	resting blood pressure (in mm Hg)
chol	cholestoral in mg/dl fetched via BMI sensor
fbs	(fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
restecg	resting electrocardiographic results

thalachh	maximum heart rate achieved
exng	exercise induced angina (1 = yes; 0 = no)
oldpeak	Previous peak

The following code splits the data to a training set and a validation set. The training set, is used to train the algorithms. The validation set, test in the code, is used to test the algorithms on new data ("the real world").

```
##########################################################
# Load Data then Split raw data set into train and test set
##########################################################

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-pro
ject.org")
 if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.or
g")
library(rpart)
library(caret)
#We Read in the heart.csv file and set it to a data frame called data
data <- read.csv('heart.csv')

# Split raw data set into train and test set: Validation set will be 25% of the Set
# Validation set will be 25% of  data
set.seed(1, sample.kind="Rounding")

test_index <- createDataPartition(y = data$output, times = 1, p = 0.25, list = FALS
E)
train_set <- data[-test_index,]
test_set <- data[test_index,]
```

Methods and Analysis

Data Analysis

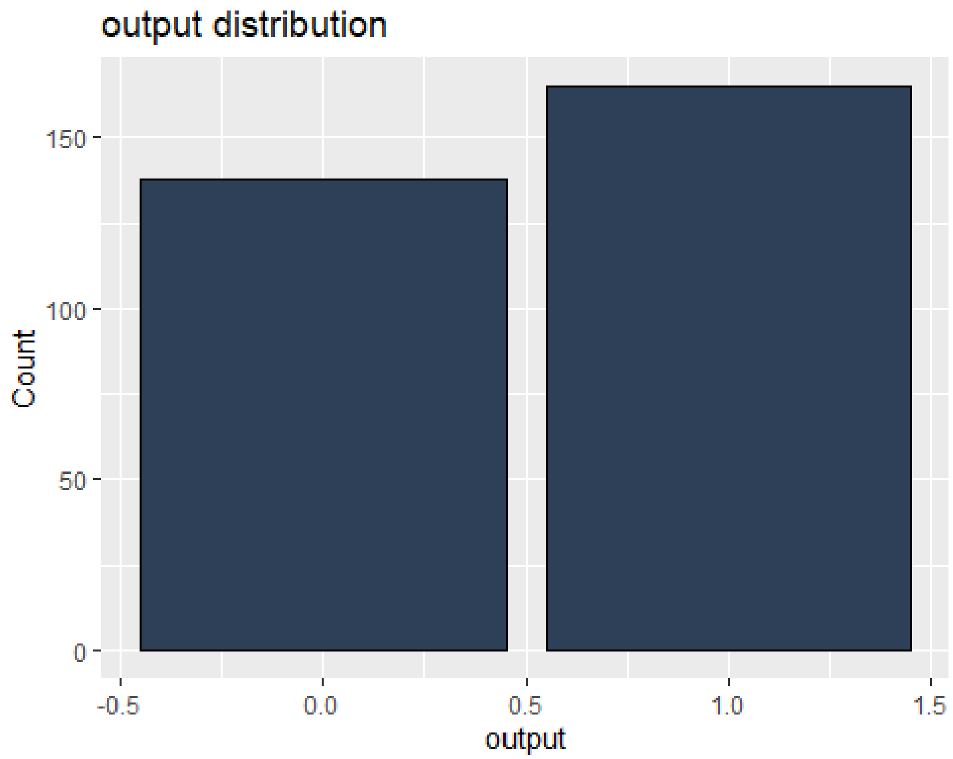
Looking at the first few rows of the “data”, we can see the features which are “age”, “sex”, “cp”, “trtbps”, “chol”, “fbs”, “restecg”, “thalachh”, “exng”, “oldpeak”, “slp”, “caa”, “thall”, “output”, Each row represents a single patient information.

```
##   age sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall output
## 1  63   1  3    145  233   1       0      150    0     2.3   0   0     1      1
## 2  37   1  2    130  250   0       1      187    0     3.5   0   0     2      1
## 3  41   0  1    130  204   0       0      172    0     1.4   2   0     2      1
## 4  56   1  1    120  236   0       1      178    0     0.8   2   0     2      1
## 5  57   0  0    120  354   0       1      163    1     0.6   2   0     2      1
## 6  57   1  0    140  192   0       1      148    0     0.4   1   0     1      1
```

A summary of the data can confirm that there are no missing values.

```
##       age             sex               cp            trtbps
##  Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0
##  1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :1.000   Median :130.0
##  Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0
##       chol            fbs            restecg          thalachh
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :246.3   Mean   :0.1485   Mean   :0.5281   Mean   :149.6
##  3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##       exng           oldpeak          slp             caa
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000
##      thall           output
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:2.000   1st Qu.:0.0000
##  Median :2.000   Median :1.0000
##  Mean   :2.314   Mean   :0.5446
##  3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :3.000   Max.   :1.0000
```

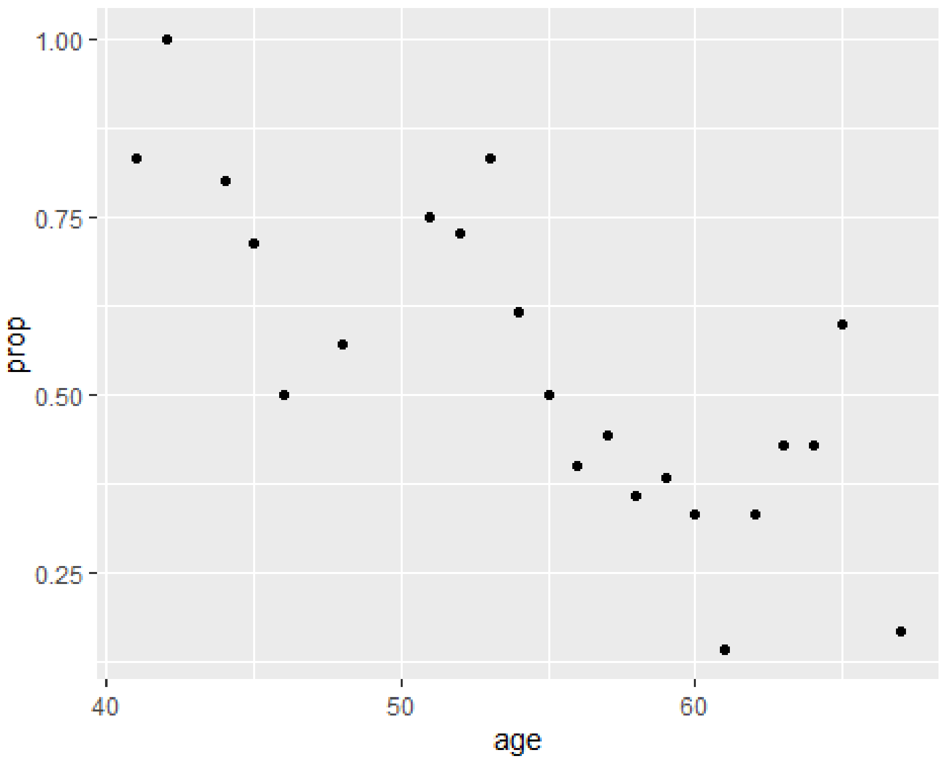
Output distribution



Model Building

Logistic regression

```
train_set %>%
  filter(age < mean(age) ) %>%
  summarize(y_hat = mean(output == 1))

##       y_hat
## 1 0.7247706

train_set %>%
   group_by(age) %>%
  filter(n() >= 5) %>%
  summarize(prop = mean(output == 1)) %>%
  ggplot(aes(age, prop)) +
  geom_point()
```

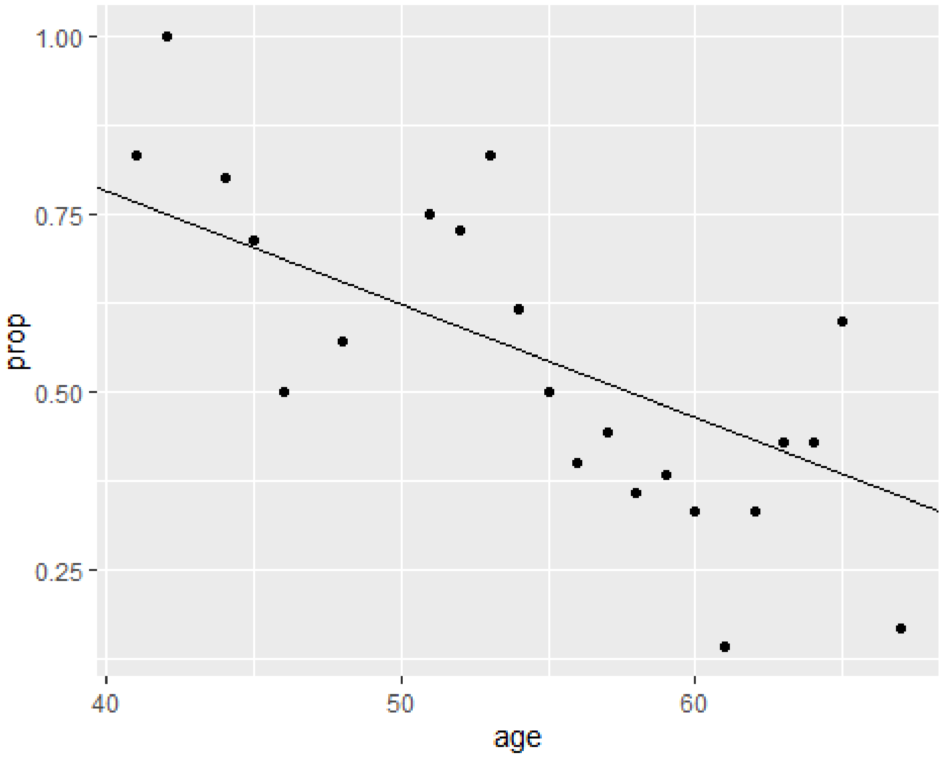


```
lm_fit <- lm(output ~ age, data = train_set)
p_hat <- predict(lm_fit, test_set)
y_hat <- ifelse(p_hat > 0.5, 1, 0)%>% factor()
confusionMatrix(y_hat, test_set$output%>%factor())$overall[["Accuracy"]]

## [1] 0.5394737
```

We see this method does substantially better than guessing.

Generalized linear models

```
train_set %>%
   group_by(age) %>%
  filter(n() >= 5) %>%
  summarize(prop = mean(output == 1)) %>%
  ggplot(aes(age, prop)) +
  geom_point() +
  geom_abline(intercept = lm_fit$coef[1], slope = lm_fit$coef[2])
```



```
glm_fit <-  glm(output ~ age, data = train_set, family = "binomial")
p_hat_logit <- predict(glm_fit, newdata = test_set, type = "response")
y_hat_logit <- ifelse(p_hat_logit > 0.5, 1, 0) %>% factor
confusionMatrix(y_hat_logit, test_set$output %>% factor)$overall[["Accuracy"]]

## [1] 0.5394737
```

Generalized linear models with more than one predictor

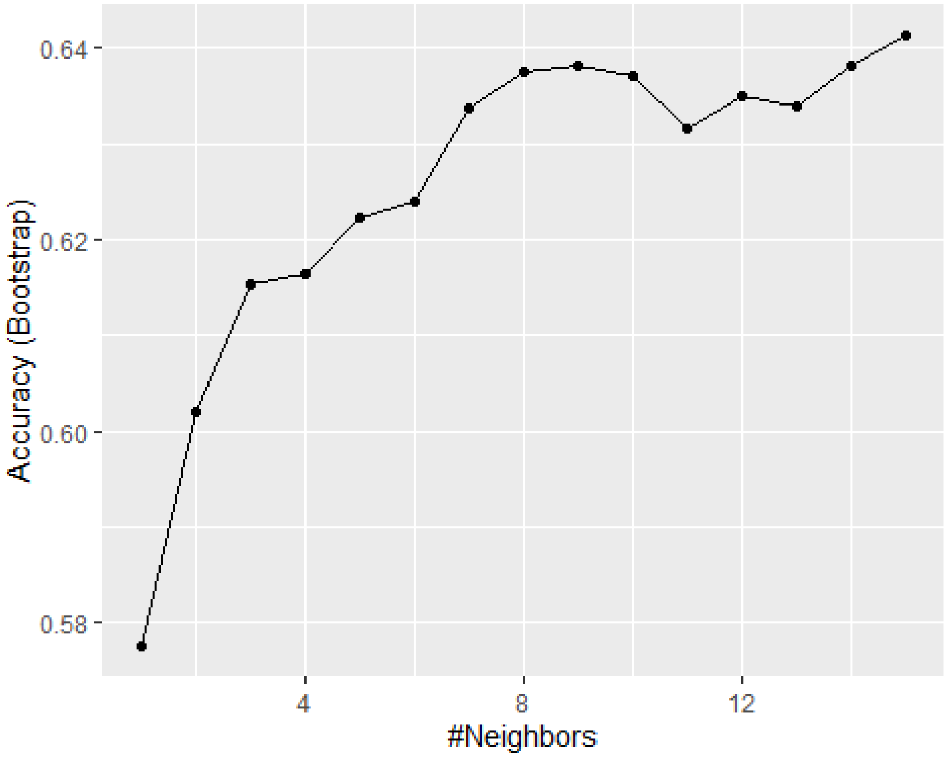
```
glm_fit_more_than_one <-  glm(output ~ age + trtbps + cp + chol +thalachh + oldpeak
, data = train_set, family = "binomial")
p_hat_logit_more_than_one <- predict(glm_fit_more_than_one, newdata = test_set, typ
e = "response")
y_hat_logit_more_than_one <- ifelse(p_hat_logit_more_than_one > 0.5, 1, 0) %>% fact
or
confusionMatrix(y_hat_logit_more_than_one, test_set$output %>% factor)$overall[["Ac
curacy"]]

## [1] 0.7631579
```

k-nearest neighbors

The K-nearest neighbors' algorithm is a supervised classification algorithm method. It classifies objects dependent on nearest neighbor. It is a type of instance-based learning. The calculation of distance of an attribute from its neighbors is measured using Euclidean distance . It uses a group of named points and uses them on how to mark another point. The data are clustered based on similarity amongst them, and is possible to fill the missing values of data using K-NN. Once the missing values are filled, various prediction techniques apply to the data set. It is possible to gain better accuracy by utilizing various combinations of these algorithms.

K-NN algorithm is simple to carry out without creating a model or making other assumptions. This algorithm is versatile and is used for classification, regression, and search. Even though K-NN is the simplest algorithm, noisy and irrelevant features affect its accuracy

```
train_knn <- train(output%>% as.factor() ~ ., method = "knn",
                   data = train_set,
                   tuneGrid = data.frame(k = seq(1, 15, 1)))
ggplot(train_knn)
```



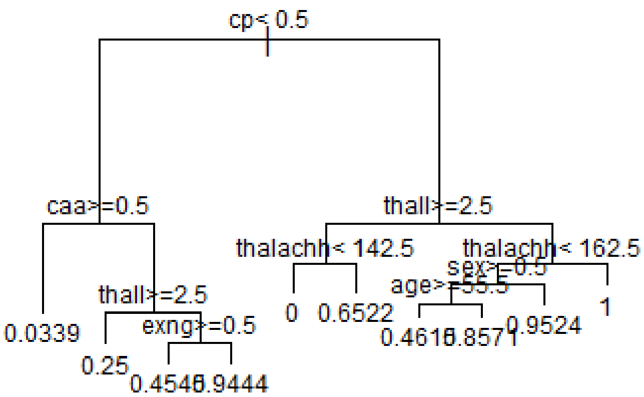
```
train_knn$bestTune

##     k
## 15 15

confusionMatrix(predict(train_knn, test_set, type = "raw"),
                test_set$output%>% as.factor())$overall["Accuracy"]

##  Accuracy
## 0.6052632
```

regression trees

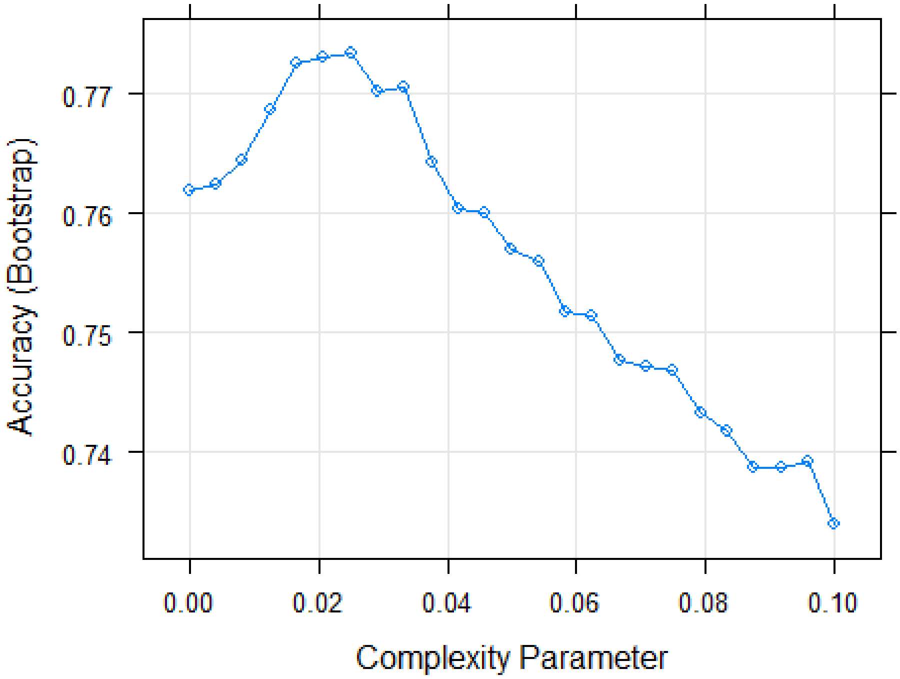
Decision tree is a classification algorithm that works on categorical as well as numerical data. Decision tree is used for creating tree-like structures. Decision tree is simple and widely used to handle medical dataset. It is easy to implement and analyses the data in tree-shaped graph. The decision tree model makes analysis based on three nodes.

Root node: main node, based on this all-other node's functions.
Interior node: handles various attributes.
Leaf node: represent the result of each test.
This algorithm splits the data into two or more analogous sets based on the most important indicators. The entropy of each attribute is calculated and then the data are divided, with predictors having maximum information gain or minimum entropy.
The results obtained are easier to read and interpret

```
fit <- rpart(output  ~ ., data = train_set)
plot(fit, margin = 0.1)
text(fit, cex = 0.75)
```



```
train_rpart <- train(output %>% as.factor() ~ .,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)),
                     data = train_set)
plot(train_rpart)
```

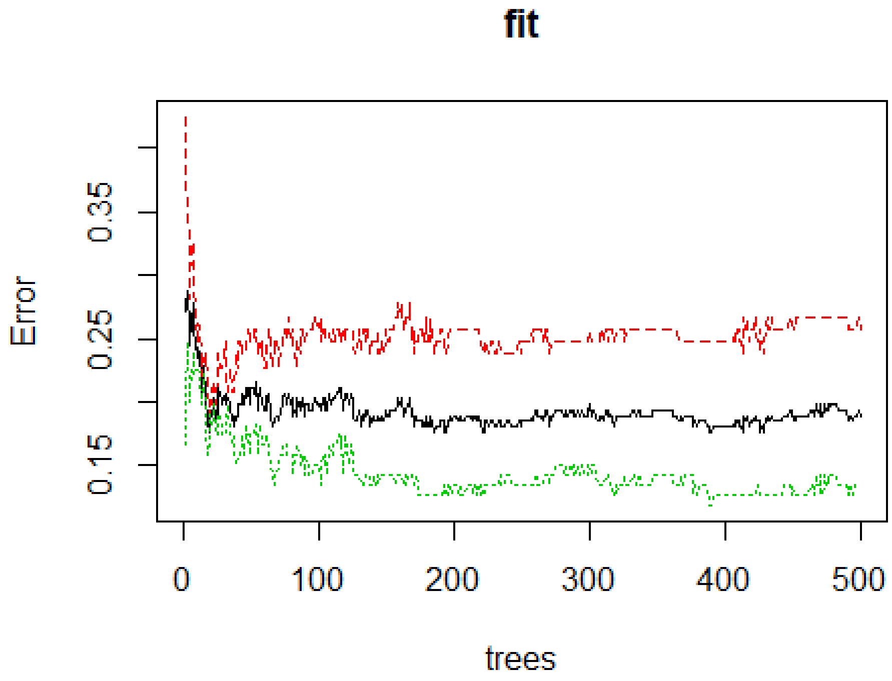


```
train_rpart$bestTune

##      cp
## 7 0.025

y_hat_rpart <- predict(train_rpart, test_set)
confusionMatrix(y_hat_rpart, test_set$output %>% as.factor())$overall["Accuracy"]

##  Accuracy
## 0.7894737
```

Random forests

Random forest algorithm is a supervised classification algorithmic technique. In this algorithm, several trees create a forest. Each individual tree in random forest lets out a class expectation and the class with most votes turns into a model's forecast. In the random forest classifier, the more number of trees give higher accuracy.

```
library(randomForest)

fit <- randomForest(output %>% as.factor() ~., data = train_set)
plot(fit)
```



```
confusionMatrix(predict(fit, test_set),
                test_set$output %>% as.factor())$overall["Accuracy"]

##  Accuracy
## 0.8421053
```

Conclusion

The overall aim is to define various data mining techniques useful in effective heart disease prediction. Efficient and accurate prediction with a lesser number of attributes and tests is our goal. In this study, I consider only 14 essential attributes. I applied four data mining classification techniques, K-nearest neighbor, decision tree, and random forest. The data were pre-processed and then used in the model. Random forest is the algorithm showing the best results in this model. I found the accuracy after implementing four algorithms to be highest in Random forest algorithm with accuracy =0.8421053.

Appendix

Environment

Operating System:

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          6.2
## year           2019
## month          12
## day            12
## svn rev        77560
## language       R
## version.string R version 3.6.2 (2019-12-12)
## nickname       Dark and Stormy Night
```