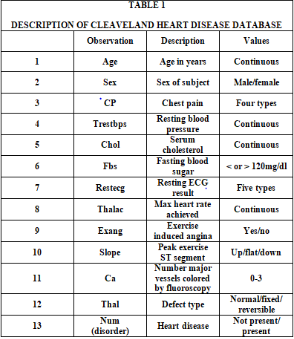
**Modelling for the Heart Stat-log dataset**

1. **Aim of analysis:**

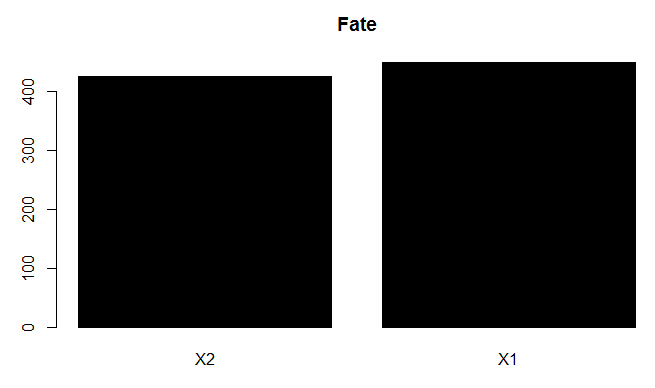
To apply different Machine learning algorithm to predict the heart disease. The best results are measured by AUC and accuracy.

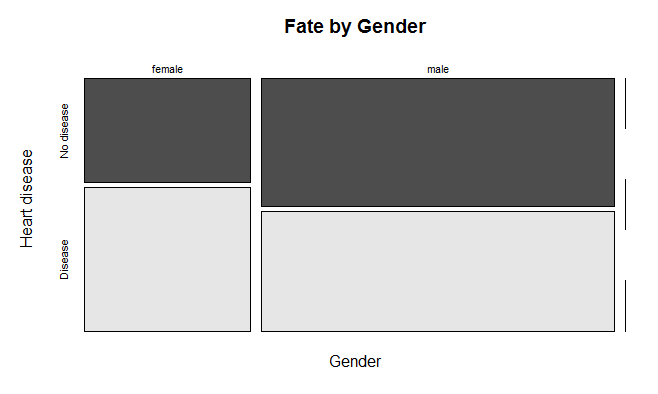
1. **Dataset:**

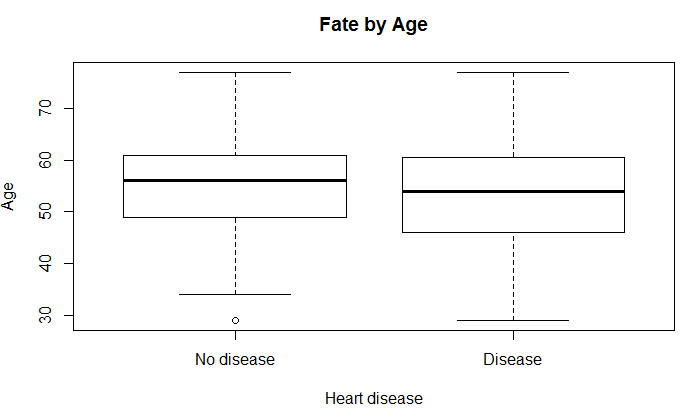


The variable we want to predict is **num** with Value 0: < 50% diameter narrowing and Value 1: > 50% diameter narrowing. We assume that every value with 0 means heart is okay, and 1,2,3,4 means heart disease.

1. **Data preparation**
2. Load heart disease data and give columns names.
3. Explore the data quickly, how many had heart attack, women or men, age, etc. Values of num > 0 are cases of heart disease. Dummify some variables.







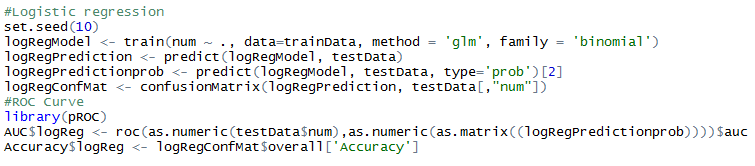
1. **Training and testing data for validation**

Split the data into Training (70%) and Testing (30%) data. Percentage of heart disease or not must be same in training and testing.

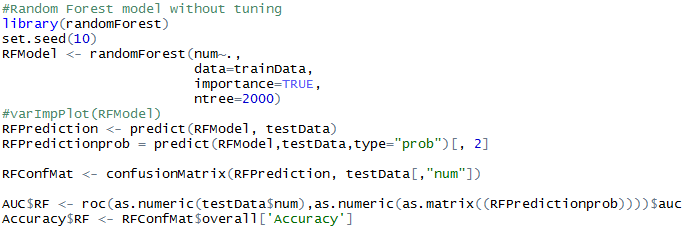
1. **Predict with 4 different methods**

Results are going to be stored in variable AUC. AUC is the area under the ROC which represents the proportion of positive data points that are correctly considered as positive and the proportion of negative data points that are mistakenly considered as positive. We also store Accuracy which is true positive and true negative divided by all results.

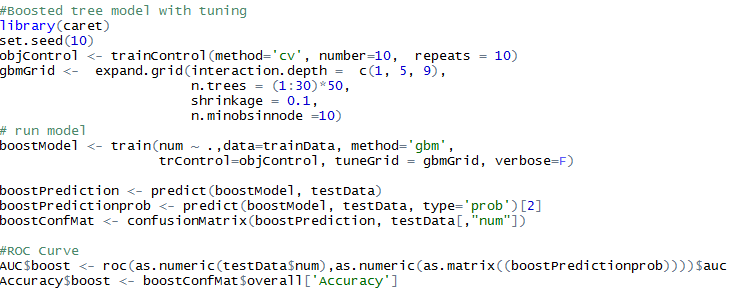
1. **Logistics Model**

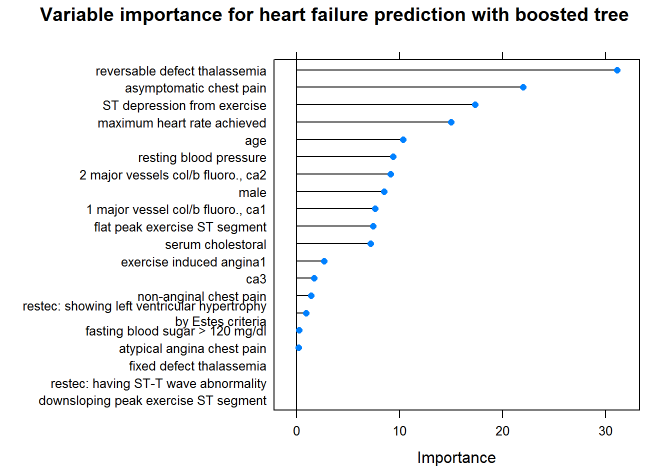
****

1. **Random Forest Model**

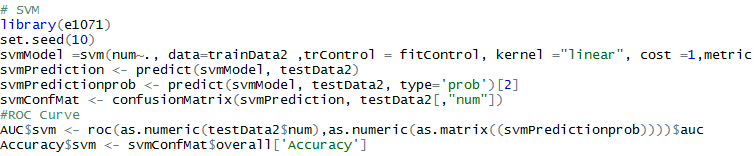


1. **Boosted Tree Model**



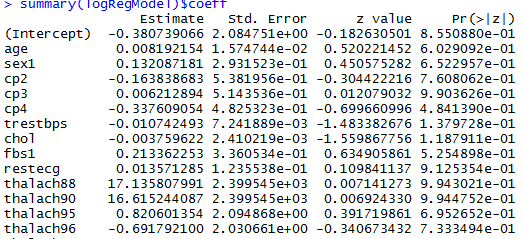


1. **Support Vector Machine (SVM)**



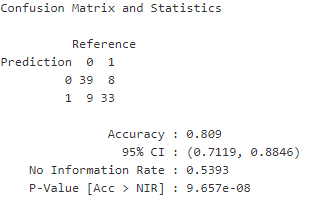
**Interpretation**

1. **Logistics Model**

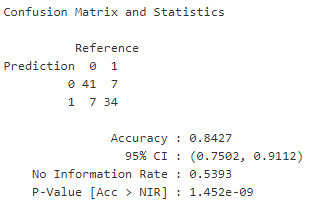


The interpretation of the coefficient for sex, for example, is: If all predictors are held at a fixed value, the odds of getting heart disease for males (males = 1) over the odds of getting heart disease for females is exp (1.32087181) = 3.746 i.e. the odds are higher.

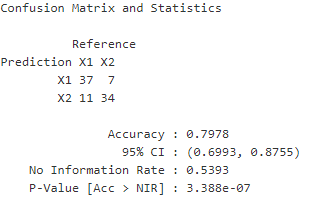
1. **Random Forest Model**



1. **Boosted Tree Model**



1. **SVM**



**Conclusion**

13 predictor variables from the UCI heart disease dataset are used to predict the diagnosis of heart disease (angiographic disease status). The performances of 4 different machine learning algorithms - logistic regression, boosted trees, random forest and support vector machines - are compared. A comparison of the area under the ROC and the accuracy of the model predictions shows that logistic regression performs best (accuracy of 0.87).  the boosted tree model was used to compare the importance of the different variables due to the easier procedure compared to logistic regression. Having a reversable defect Thalassemia is the most important predictor in the boosted tree model, followed by asymptomatic chest pain and ST depression from exercise.

**CODE**

library(MASS)

library(ISLR)

library(magic)

setwd("C:/Users/ADMIN/Downloads")

heart.data <- read.csv("heart-statlog.csv",header=FALSE,sep=",",na.strings = '?')

names(heart.data) <- c( "age", "sex", "cp", "trestbps", "chol","fbs", "restecg",

"thalach","exang","slope", "ca", "thal", "num")

head(heart.data,3)

dim(heart.data)

str(heart.data)

heart.data$num[heart.data$num > 0] <- 1

barplot(table(heart.data$num),

main="Fate", col="black")

# change a few predictor variables from integer to factors (make dummies)

chclass <-c("numeric","factor","factor","numeric","numeric","factor","factor","numeric","factor","factor","factor","factor","factor")

heart.data$age<-as.numeric(heart.data$age)

heart.data$sex<-as.factor(heart.data$sex)

heart.data$cp<-as.factor(heart.data$cp)

heart.data$trestbps<-as.numeric(heart.data$trestbps)

heart.data$chol<-as.numeric(heart.data$chol)

heart.data$fbs<-as.factor(heart.data$fbs)

heart.data$restecg<-as.numeric(heart.data$restecg)

heart.data$thalach<-as.factor(heart.data$thalach)

heart.data$exang<-as.factor(heart.data$exang)

heart.data$slope<-as.factor(heart.data$slope)

heart.data$ca<-as.factor(heart.data$ca)

heart.data$thal<-as.factor(heart.data$thal)

heart.data$num<-as.factor(heart.data$num)

str(heart.data)

heart = heart.data #add labels only for plot

levels(heart$num) = c("No disease","Disease")

levels(heart$sex) = c("female","male","")

mosaicplot(heart$sex ~ heart$num,

main="Fate by Gender", shade=FALSE,color=TRUE,

xlab="Gender", ylab="Heart disease")

boxplot(heart$age ~ heart$num,

main="Fate by Age",

ylab="Age",xlab="Heart disease")

#checking for missing values

s = sum(is.na(heart.data))

heart.data <- na.omit(heart.data)

#Training and testing data for validation

library(caret)

set.seed(10)

inTrainRows <- createDataPartition(heart.data$num,p=0.7,list=FALSE)

trainData <- heart.data[inTrainRows,]

testData <- heart.data[-inTrainRows,]

nrow(trainData)/(nrow(testData)+nrow(trainData))

#Predict with 4 different methods

AUC = list()

Accuracy = list()

#Logistic regression

set.seed(10)

logRegModel <- train(num ~ ., data=trainData, method = 'glm', family = 'binomial')

logRegPrediction <- predict(logRegModel, testData)

logRegPredictionprob <- predict(logRegModel, testData, type='prob')[2]

logRegConfMat <- confusionMatrix(logRegPrediction, testData[,"num"])

#ROC Curve

library(pROC)

AUC$logReg <- roc(as.numeric(testData$num),as.numeric(as.matrix((logRegPredictionprob))))$auc

Accuracy$logReg <- logRegConfMat$overall['Accuracy']

#Random Forest model without tuning

library(randomForest)

set.seed(10)

RFModel <- randomForest(num~.,

data=trainData2,

ntree=2000)

#varImpPlot(RFModel)

RFPrediction <- predict(RFModel, testData)

RFPredictionprob = predict(RFModel,testData,type="prob")[, 2]

RFConfMat <- confusionMatrix(RFPrediction, testData[,"num"])

AUC$RF <- roc(as.numeric(testData$num),as.numeric(as.matrix((RFPredictionprob))))$auc

Accuracy$RF <- RFConfMat$overall['Accuracy']

#Boosted tree model with tuning

library(caret)

set.seed(10)

objControl <- trainControl(method='cv', number=10, repeats = 10)

gbmGrid <- expand.grid(interaction.depth = c(1, 5, 9),

n.trees = (1:30)\*50,

shrinkage = 0.1,

n.minobsinnode =10)

# run model

boostModel <- train(num ~ .,data=trainData, method='gbm',

trControl=objControl, tuneGrid = gbmGrid, verbose=F)

boostPrediction <- predict(boostModel, testData)

boostPredictionprob <- predict(boostModel, testData, type='prob')[2]

boostConfMat <- confusionMatrix(boostPrediction, testData[,"num"])

#ROC Curve

AUC$boost <- roc(as.numeric(testData$num),as.numeric(as.matrix((boostPredictionprob))))$auc

Accuracy$boost <- boostConfMat$overall['Accuracy']

# SVM

library(e1071)

set.seed(10)

svmModel =svm(num~., data=trainData2 ,trControl = fitControl, kernel ="linear", cost =1,metric = "ROC", scale =FALSE)

svmPrediction <- predict(svmModel, testData2)

svmPredictionprob <- predict(svmModel, testData2, type='prob')[2]

svmConfMat <- confusionMatrix(svmPrediction, testData2[,"num"])

#ROC Curve

AUC$svm <- roc(as.numeric(testData2$num),as.numeric(as.matrix((svmPredictionprob))))$auc

Accuracy$svm <- svmConfMat$overall['Accuracy']

# Interpretation

summary(logRegModel)$coeff

boostImp =varImp(boostModel, scale = FALSE)

row = rownames(varImp(boostModel, scale = FALSE)$importance)

row = convert.names(row)

rownames(boostImp$importance)=row

plot(boostImp,main = 'Variable importance for heart failure prediction with boosted tree')

RFConfMat