Heart disease health indicator

Abhijeet Anand(abhan872) and Hussnain Khalid(huskh803)

12/16/2021

Heart Disease is among the most prevalent chronic diseases, which impacts millions of civilians every year and exerting a significant financial burden on the economy of the country. The build up of plaques inside larger coronary arteries, molecular changes associated with aging, chronic inflammation, high blood pressure, and diabetes are all causes of and risk factors for heart disease.

The Centre for Disease Control and Prevention has identified high blood pressure, high blood cholesterol, and smoking as three key risk factors for heart disease. While, The National Heart, Lung, and Blood Institute highlights a wider array of factors such as Age, Environment and Occupation, Family History and Genetics, Lifestyle Habits, Other Medical Conditions, Race or Ethnicity, and Sex for clinicians to use in diagnosing coronary heart disease.

This dataset contains cleaned 100,638 survey responses from BRFSS 2015 to be used primarily for the binary classification of heart disease. There is good class imbalance in this dataset. 76745 respondents do not have/had heart disease while 23,893 have/had heart disease.

Below is the summary of the data that we use:

```
##  HeartDiseaseorAttack     HighBP          HighChol        CholCheck
##  0:76745              Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1:23893              1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000
##                       Median :0.0000   Median :0.0000   Median :1.0000
##                       Mean   :0.4812   Mean   :0.4724   Mean   :0.9673
##                       3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##                       Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##       BMI            Smoker           Stroke           Diabetes
##  Min.   :12.00   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:24.00   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :27.00   Median :0.0000   Median :0.00000   Median :0.0000
##  Mean   :28.62   Mean   :0.4702   Mean   :0.06043   Mean   :0.3589
##  3rd Qu.:31.00   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :98.00   Max.   :1.0000   Max.   :1.00000   Max.   :2.0000
##   PhysActivity        Fruits          Veggies       HvyAlcoholConsump
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.00000
##  Median :1.0000   Median :1.0000   Median :1.0000   Median :0.00000
##  Mean   :0.7433   Mean   :0.6276   Mean   :0.8048   Mean   :0.05312
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  AnyHealthcare     NoDocbcCost         GenHlth         MentHlth
##  Min.   :0.0000   Min.   :0.00000   Min.   :1.000   Min.   : 0.000
##  1st Qu.:1.0000   1st Qu.:0.00000   1st Qu.:2.000   1st Qu.: 0.000
##  Median :1.0000   Median :0.00000   Median :3.000   Median : 0.000
##  Mean   :0.9514   Mean   :0.08857   Mean   :2.648   Mean   : 3.392
```

```
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:3.000   3rd Qu.: 2.000
##  Max.   :1.0000   Max.   :1.00000   Max.   :5.000   Max.   :30.000
##     PhysHlth         DiffWalk           Sex              Age
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.0000   Min.   : 1.000
##  1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 6.000
##  Median : 0.000   Median :0.0000   Median :0.0000   Median : 9.000
##  Mean   : 5.026   Mean   :0.2077   Mean   :0.4627   Mean   : 8.393
##  3rd Qu.: 4.000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:11.000
##  Max.   :30.000   Max.   :1.0000   Max.   :1.0000   Max.   :13.000
##    Education         Income
##  Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:5.000
##  Median :5.000   Median :7.000
##  Mean   :5.017   Mean   :5.937
##  3rd Qu.:6.000   3rd Qu.:8.000
##  Max.   :6.000   Max.   :8.000
```

Now, we checking the data we several models namely SVM, decision tree, and Logistic Regression.

Models

Decision Tree

A decision tree is a graphical representation of all possible solutions to a decision based on certain conditions. On each step or node of a decision tree, used for classification, we try to form a condition on the features to separate all the labels or classes contained in the dataset to the fullest purity. A tree is built by splitting the source set, constituting the root node of the tree, into subsets—which constitute the successor children. The splitting is based on a set of splitting rules based on classification features. With the current dataset, a decision tree is learned on training data.

SVM(Support Vector Machine)

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space. a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin. The advantages of support vector machines are: Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.The main disadvantage of the SVM algorithm is that it has several key parameters that need to be set correctly to achieve the best classification results for any given problem

Since, we modeled using SVM and decision tree. Below are the results from the same.

Confusion Matrix from Decision Tree

```
cm_dt
```

```
##
## pred_dt     0     1
##       0 20963  4514
##       1  2032  2683
```

Confusion Matrix from SVM

```
cm_svm
```

```
##
## pred_svm     0     1
##        0 21827  4767
##        1  1168  2430
```

F1 Score from Decision Tree

```
f1_score_dt # 0.1529777
```

```
## [1] 0.1529777
```

F1 Score from SVM

```
f1_score_svm # 0.09448153
```

```
## [1] 0.09448153
```

Accuracy from Decision Tree

```
accuracy_dt # 0.7831876
```

```
## [1] 0.7831876
```

Accuracy from SVM

```
## [1] 0.8034247
```

```
## Misclassification error from decision tree:  0.21681240063593
```

```
## Misclassification error from SVM:  0.19657525172231
```

ROC Curve

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

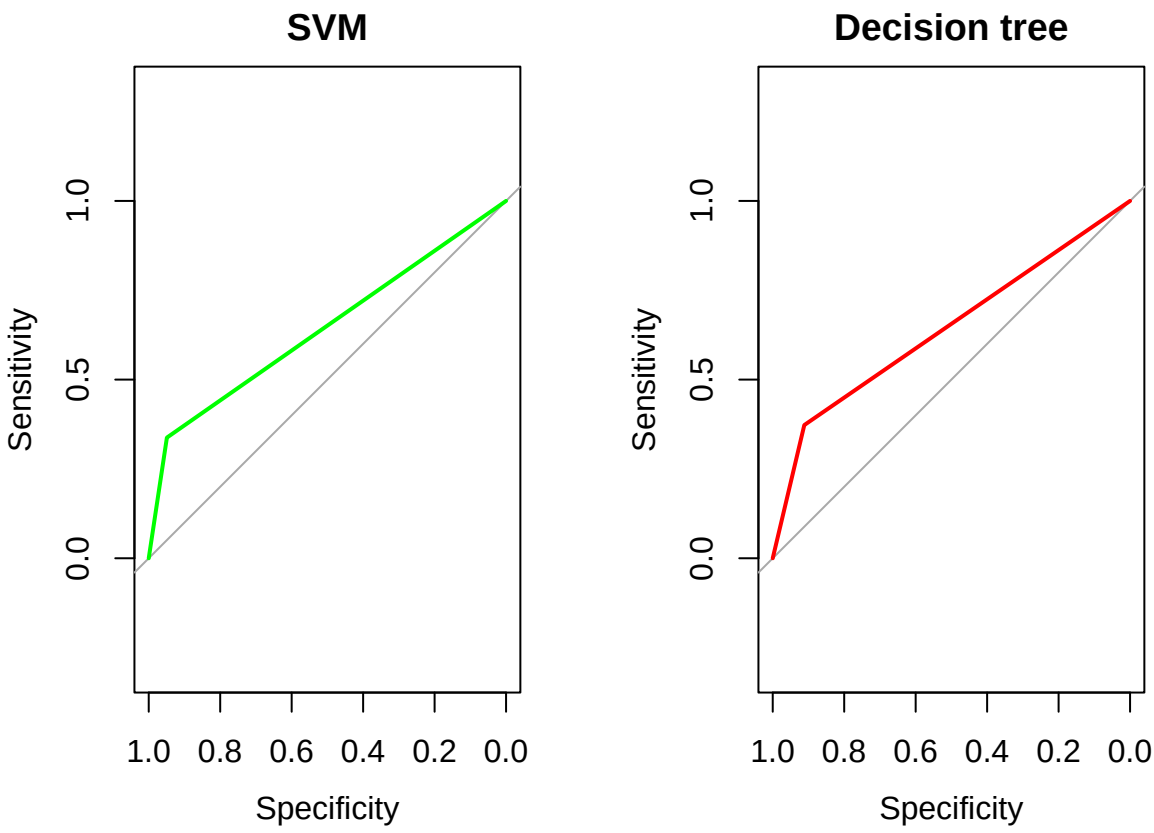
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

An ROC plot for SVM and Decision tree is plotted below separately, which is shown below.



Now, we will prune the decision tree.

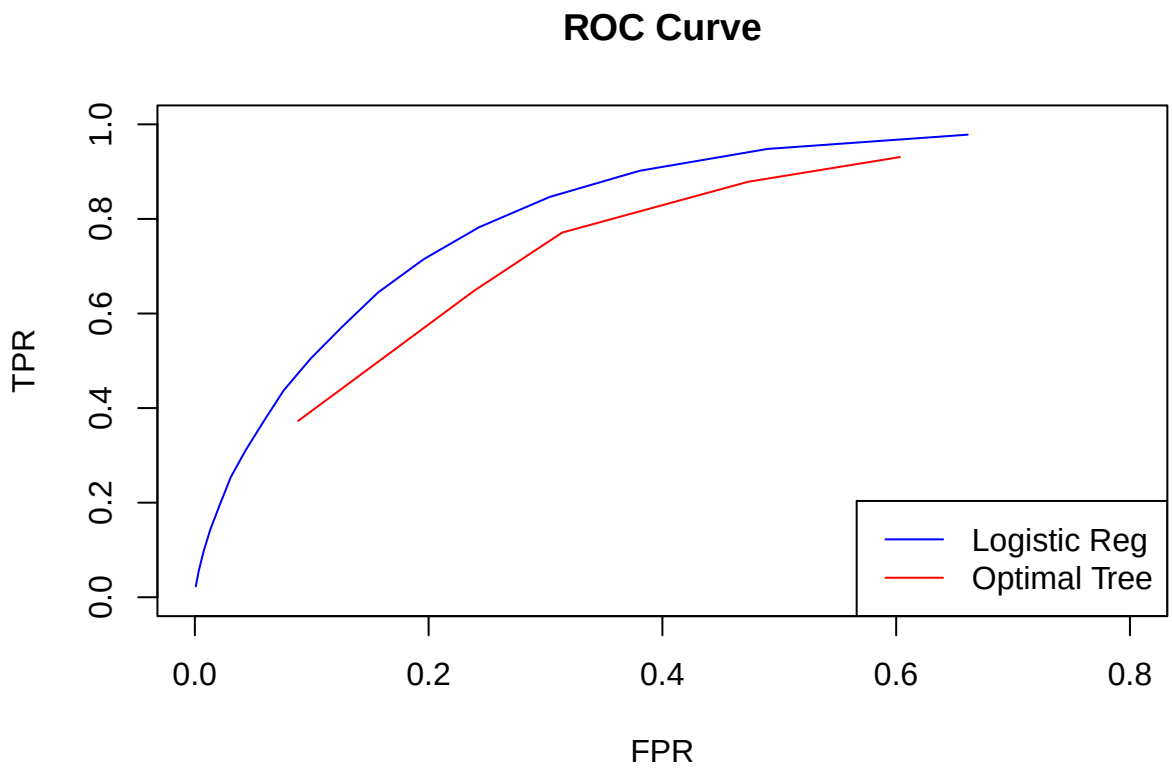
Let's see the result from the pruned tree. We print the confusion matrix from the pruned tee.

```
##    predOptimalTree
##         0     1
##   0 20913  2101
##   1  4543  2634
```

Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. It is one of the simplest ML algorithms that can be used for various classification problems. Logistic regression is easier to implement, interpret, and very efficient to train. If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.

Let's plot the ROC curve for the comparison of decision tree and Logistic regression.



Finally, we show the accuracy for all the different models.

```
## [1] 0.7831876
```

```
## Accuracy from decision tree:  0.78318759936407
```

```
## Accuracy from SVM:  0.80342474827769
```

Since, we find SVM and decision tree with better accuracy, we prefer these two models.

APPENDIX

```
# Libraries
################################################################################
library(ggplot2)
library(tree)
library(caret)
library(dplyr)
library(e1071)
library(readxl)
library(pROC)
################################################################################

################################################################################
## Load data
# Data Processing
################################################################################
```

```r
# Load data
df <- read.csv2("data3.csv", sep=",")
df[,1]<-factor(df[,1])

heart_disease_data <- read.csv2("data3.csv", sep=",")
heart_disease_data[,1]<-factor(heart_disease_data[,1])

# Summary
summary(df)


# Factorize BMI data
temp <- df$BMI
for(i in 1:length(temp)){
  if(temp[i]>31) {
    temp[i] <-4
  } else if(temp[i]>27 && temp[i]<32) {
    temp[i] <- 3
  } else if(temp[i]>24 && temp[i]<28) {
    temp[i] <- 2
  } else {
    temp[i] <- 1
  }
}
df$BMI <- temp

# Removing NA's
df <- na.omit(df)
################################################################################

# Partition data into training(40), validation(30) and test(30)
################################################################################
n=dim(df)[1]
set.seed(12345)

## Training Data
id=sample(1:n, floor(n*0.4))
train=df[id,]

id1=setdiff(1:n, id)
set.seed(12345)

## Validation data
id2=sample(id1, floor(n*0.3))
valid=df[id2,]

## Testing data
id3=setdiff(id1,id2)
test=df[id3,]
################################################################################


# Models
```

```
##############################################################################
# Decision Tree with default settings
dt_default <- tree(formula = HeartDiseaseorAttack~.,
data = train,
method="class")


# SVM with default settings
svm_default = svm(formula = HeartDiseaseorAttack~.,
                  data = train,
                  type = 'C-classification',
                  kernel = 'linear')
##############################################################################


# Predictions
##############################################################################
# decision tree
pred_dt<- predict(dt_default, test, type = "class")

# SVM
pred_svm = predict(svm_default, newdata = test)
##############################################################################


# Confusion Matrix
##############################################################################
# decision tree
cm_dt<-table(pred_dt, test$HeartDiseaseorAttack)

# SVM
cm_svm = table(pred_svm, test[,1])
##############################################################################


# Misclassification error
##############################################################################
# decision tree
mmce_dt <- 1 - sum(diag(cm_dt)) / sum(cm_dt)

# SVM
mmce_svm <- 1 - sum(diag(cm_svm)) / sum(cm_svm)
##############################################################################


# F1 Score
##############################################################################
# Decision tree
TN_dt <- cm_dt[1,1]
TP_dt <- cm_dt[2,2]
FN_dt <- cm_dt[1,2]
FP_dt <- cm_dt[2,1]
precision_dt <- (TP_dt) / (TP_dt + FP_dt) # 0.569035
```

```r
recall_score_dt <- (FP_dt) / (FP_dt + TN_dt) # 0.08836704
f1_score_dt <- 2 * ((precision_dt * recall_score_dt) / (precision_dt + recall_score_dt))

# SVM
TN_svm <- cm_svm[1,1]
TP_svm <- cm_svm[2,2]
FN_svm <- cm_svm[1,2]
FP_svm <- cm_svm[2,1]
precision_svm <- (TP_svm) / (TP_svm + FP_svm) # 0.6753752
recall_score_svm <- (FP_svm) / (FP_svm + TN_svm) # 0.05079365
f1_score_svm <- 2 * ((precision_svm * recall_score_svm) / (precision_svm + recall_score_svm))
##############################################################################


# Accuracy
##############################################################################
# Decision tree
accuracy_dt <- (TP_dt + TN_dt) / (TP_dt + TN_dt + FP_dt + FN_dt)

# SVM
accuracy_svm <- (TP_svm + TN_svm) / (TP_svm + TN_svm + FP_svm + FN_svm)
##############################################################################


# ROC Plot
##############################################################################
# fpr_dt <- FP_dt/(FP_dt+TN_dt)
# tpr_dt <- TP_dt/(TP_dt+FN_dt)
# fpr_sum <- FP_sum/(FP_sum+TN_sum)
# tpr_sum <- TP_sum/(TP_sum+FN_sum)
roc_dt <- roc(response = test$HeartDiseaseorAttack, predictor =as.numeric(pred_dt))
roc_svm <- roc(response = test$HeartDiseaseorAttack, predictor =as.numeric(pred_svm))
par(mfrow=c(1,2))
plot(roc_svm, col = "green", main = c("SVM"))
plot(roc_dt, col = "red", main = c("Decision tree"))


# Pruning Tree
##############################################################################
set.seed(12345)
trainScore = rep(0, 100)
testScore = rep(0, 100)

for(i in 2:100) {
  prunedTree = prune.tree(dt_default, best=i)
  pd = predict(prunedTree, newdata = valid, type= "tree")
  trainScore[i] = deviance(prunedTree)
  testScore[i] = deviance(pd)
}

optimalLeaves <- which.min(testScore[2:100])

optimalTree = prune.tree(dt_default, best=optimalLeaves)
```

```r
predOptimalTree = predict(optimalTree, newdata= valid, type="class")

cnfMatrixOptimalTree <- table(valid$HeartDiseaseorAttack, predOptimalTree)
cnfMatrixOptimalTree


# Logistic Regression
#############################################################################################
pi_generator <- seq(0.05, 0.95, 0.05)
logiReg <- glm(formula = HeartDiseaseorAttack~., data = train, family = "binomial")
logiRegPred <- predict(logiReg, select(test, -c(HeartDiseaseorAttack)), type = "response")

confList <- list()
for(i in pi_generator) {
  a <- as.factor(ifelse(logiRegPred>i, 'yes', 'no'))
  b <- table(a, test$HeartDiseaseorAttack)
  confList <- c(confList, list(b))
}

tpr_logR <- c()
fpr_logR <- c()
total_loop <- length(pi_generator)-1
for (iter in 1:18) {
  tpr_value <- confList[[iter]][4]/(confList[[iter]][3]+confList[[iter]][4])
  tpr_logR <- c(tpr_logR, tpr_value)

  fpr_value <- confList[[iter]][2]/(confList[[iter]][1]+confList[[iter]][2])
  fpr_logR <- c(fpr_logR, fpr_value)
}


# Classify test data with Optimal Tree
optimalTreePred = predict(optimalTree, newdata= test, type="vector")
confListOptTree <- list()
for(i in pi_generator) {
  k <- as.factor(ifelse(optimalTreePred[,2]>i, 'yes', 'no'))
  l <- table(k, test$HeartDiseaseorAttack)
  confListOptTree <- c(confListOptTree, list(l))
}

tprOptTree <- c()
fprOptTree <- c()

for (iter1 in 1:19) {
  TP <- confListOptTree[[iter1]][4]
  P <- (confListOptTree[[iter1]][3]+confListOptTree[[iter1]][4])
  tpr_value <- TP/P
  tprOptTree <- c(tprOptTree, tpr_value)

  FP <- confListOptTree[[iter1]][2]
  N <- (confListOptTree[[iter1]][1]+confListOptTree[[iter1]][2])
  fpr_value <- FP/N
  fprOptTree <- c(fprOptTree, fpr_value)
```

```
}
tprOptTree[16:19] <- 0.0
fprOptTree[16:19] <- 0.0

plot(fpr_logR, tpr_logR, type="l", col="blue",
     xlab="FPR", ylab="TPR", xlim=c(0,0.8), ylim=c(0,1))
lines(fprOptTree, tprOptTree, col="red", type="l")
title("ROC Curve")
legend("bottomright", legend=c("Logistic Reg","Optimal Tree"),
       col=c("blue","red"), lty = 1)


# Accuracies
#################################################################################
accuracy_dt = 1-mmce_dt
accuracy_dt
cat(paste("Accuracy from decision tree: ", accuracy_dt))
cat(paste("Accuracy from SVM: ", accuracy_svm))
```