Heart Attack Analysis & Prediction

Jessica Stapleton, Zach Chase, Ahmed Ayoubi

5/27/2022

Background and Description

Heart disease (HD) is one of the most common diseases nowadays. An early diagnosis of such a disease is crucial for many healthcare providers to prevent their patients from such a disease and save lives. We will be predicting whether a person will have a heart attack (HA) or not using classification trees and logistic regression with the attributes listed below.

This dataset was obtained from kaggle.com (https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset). The data consists of 14 variables and 1025 observations.

Variables:

- age: Age of the patient
- sex: Sex of the patient (1 = male, 0 = female)
- exang: exercise induced angina (1 = yes; 0 = no)
- ca: number of major vessels (0-4)
- cp: Chest Pain type chest pain type(Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic)
- trestbps: 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(Value 0: normal, Value 1: having ST-T , Value 2: hypertrophy)
- thalach: maximum heart rate achieved
- oldpeak: ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot)
- slope: the slope of the peak exercise ST segment (Value 0: upsloping, Value 1: flat, Value 2: downsloping)

- thal: A blood disorder called thalassemia (Value 0: NULL “dropped from the dataset previously” , Value 1: fixed, Value 2: normal, Value 3: reversible defect)

- target: 0 = less chance of heart attack. 1 = more chance of heart attack

```
#loading the data
dat <- read.csv("heart.csv")

#checking the types of variables are in the correct form
str(dat)

## 'data.frame':    1025 obs. of  14 variables:
##  $ age     : int  52 53 70 61 62 58 58 55 46 54 ...
##  $ sex     : int  1 1 1 1 0 0 1 1 1 1 ...
##  $ cp      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ trestbps: int  125 140 145 148 138 100 114 160 120 122 ...
##  $ chol    : int  212 203 174 203 294 248 318 289 249 286 ...
##  $ fbs     : int  0 1 0 0 1 0 0 0 0 0 ...
##  $ restecg : int  1 0 1 1 1 0 2 0 0 0 ...
##  $ thalach : int  168 155 125 161 106 122 140 145 144 116 ...
##  $ exang   : int  0 1 1 0 0 0 0 1 0 1 ...
##  $ oldpeak : num  1 3.1 2.6 0 1.9 1 4.4 0.8 0.8 3.2 ...
##  $ slope   : int  2 0 0 2 1 1 0 1 2 1 ...
##  $ ca      : int  2 0 0 1 3 0 3 1 0 2 ...
##  $ thal    : int  3 3 3 3 2 2 1 3 3 2 ...
##  $ target  : int  0 0 0 0 0 1 0 0 0 0 ...
```

Data Cleaning

We have to convert the categorical variables into factors. We will convert sex, cp, fbs, restecg, exang, slope, thal, and target and label them for convenience.

```
#factor and label
dat$sex<-as.factor(dat$sex)
levels(dat$sex)<-c("Female","Male")

dat$cp<-as.factor(dat$cp)
levels(dat$cp)<-c("typical","atypical","non-anginal","asymptomatic")

dat$fbs<-as.factor(dat$fbs)
levels(dat$fbs)<-c("False","True")

dat$restecg<-as.factor(dat$restecg)
levels(dat$restecg)<-c("normal","stt","hypertrophy")

dat$exang<-as.factor(dat$exang)
levels(dat$exang)<-c("No","Yes")

dat$slope<-as.factor(dat$slope)
levels(dat$slope)<-c("upsloping","flat","downsloping")
```

```r
#number of vessels here so no need to label
dat$ca<-as.factor(dat$ca)

dat$thal<-as.factor(dat$thal)
levels(dat$thal)<-c("fixed", "normal","reversable")

dat$target<-as.factor(dat$target)
levels(dat$target)<-c("No HA", "Yes HA")
```

Checking the changes.

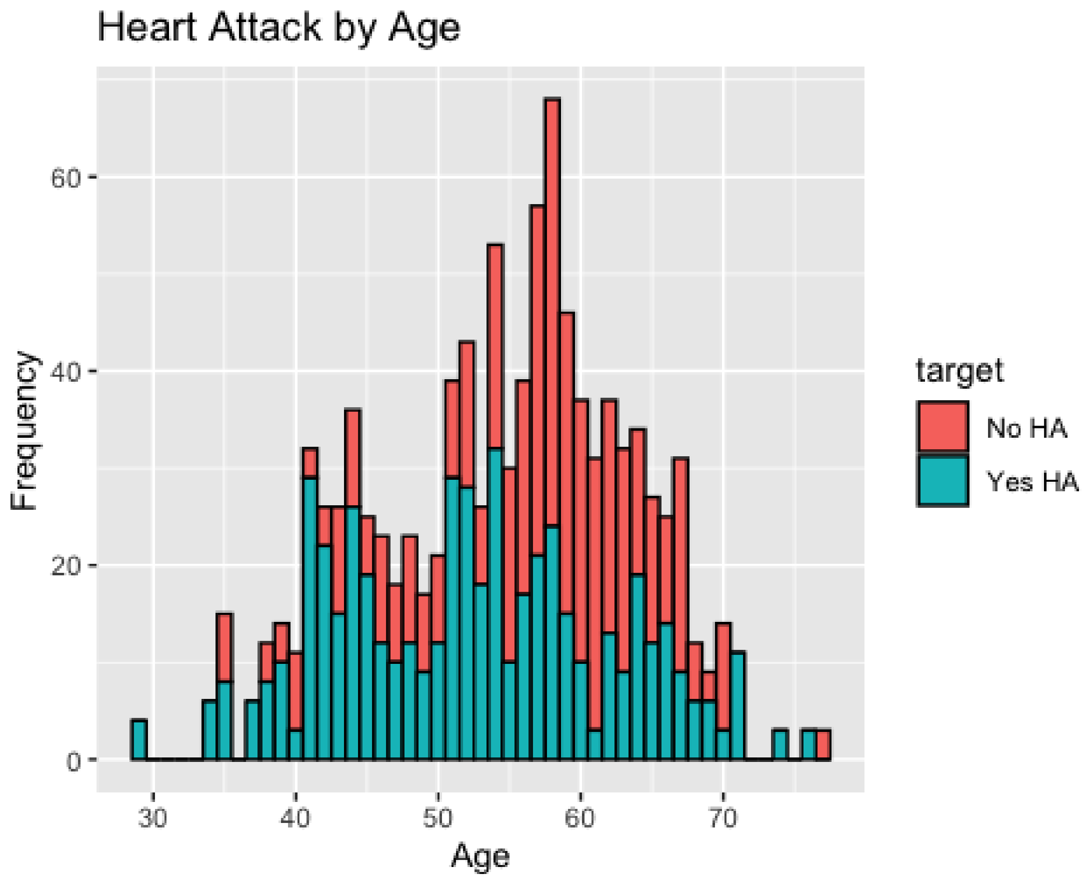
```r
str(dat)
```

```
## 'data.frame':    1025 obs. of  14 variables:
##  $ age     : int  52 53 70 61 62 58 58 55 46 54 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 1 2 2 2 2 ...
##  $ cp      : Factor w/ 4 levels "typical","atypical",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ trestbps: int  125 140 145 148 138 100 114 160 120 122 ...
##  $ chol    : int  212 203 174 203 294 248 318 289 249 286 ...
##  $ fbs     : Factor w/ 2 levels "False","True": 1 2 1 1 2 1 1 1 1 1 ...
##  $ restecg : Factor w/ 3 levels "normal","stt",..: 2 1 2 2 2 1 3 1 1 1 ...
##  $ thalach : int  168 155 125 161 106 122 140 145 144 116 ...
##  $ exang   : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 2 1 2 ...
##  $ oldpeak : num  1 3.1 2.6 0 1.9 1 4.4 0.8 0.8 3.2 ...
##  $ slope   : Factor w/ 3 levels "upsloping","flat",..: 3 1 1 3 2 2 1 2 3 2
...
##  $ ca      : Factor w/ 5 levels "0","1","2","3",..: 3 1 1 2 4 1 4 2 1 3
...
##  $ thal    : Factor w/ 3 levels "fixed","normal",..: 3 3 3 3 2 2 1 3 3 2
...
##  $ target  : Factor w/ 2 levels "No HA","Yes HA": 1 1 1 1 1 2 1 1 1 1 ...
```

Exploratory Data Analysis (EDA)

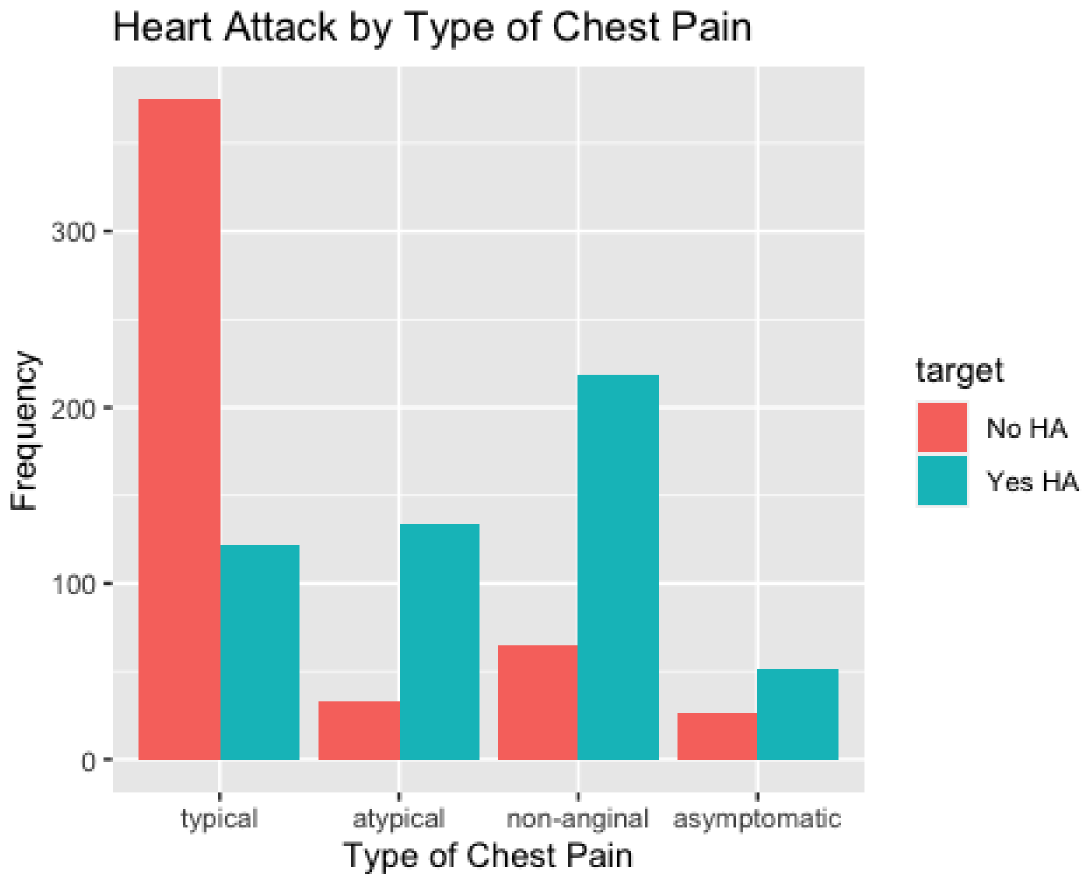
Let’s explore the data a bit before getting into the analysis.

```r
ggplot(dat,aes(x=age,fill=target, color = target)) + geom_histogram(binwidth
= 1,color="black") + labs(x = "Age",y = "Frequency", title = "Heart Attack by
Age")
```



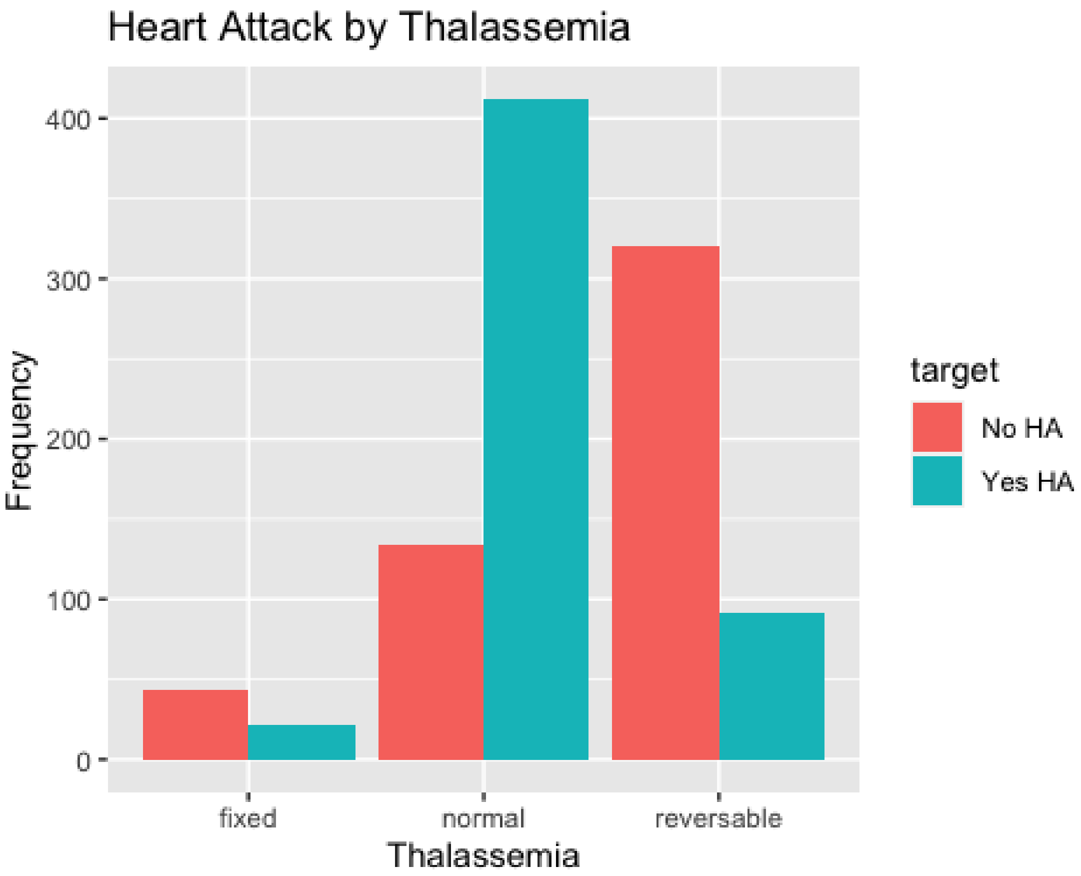
We can conclude that likeliness of a heart attack is more likely to occur in a person’s 40’s-50’s than in their 60’s or later.

```
#bar plot - position = "dodge" makes them side by side
ggplot(dat,aes(x=cp,fill=target)) + geom_bar(position = "dodge") + labs(x = 
"Type of Chest Pain", y = "Frequency", title = "Heart Attack by Type of Chest 
Pain")
```



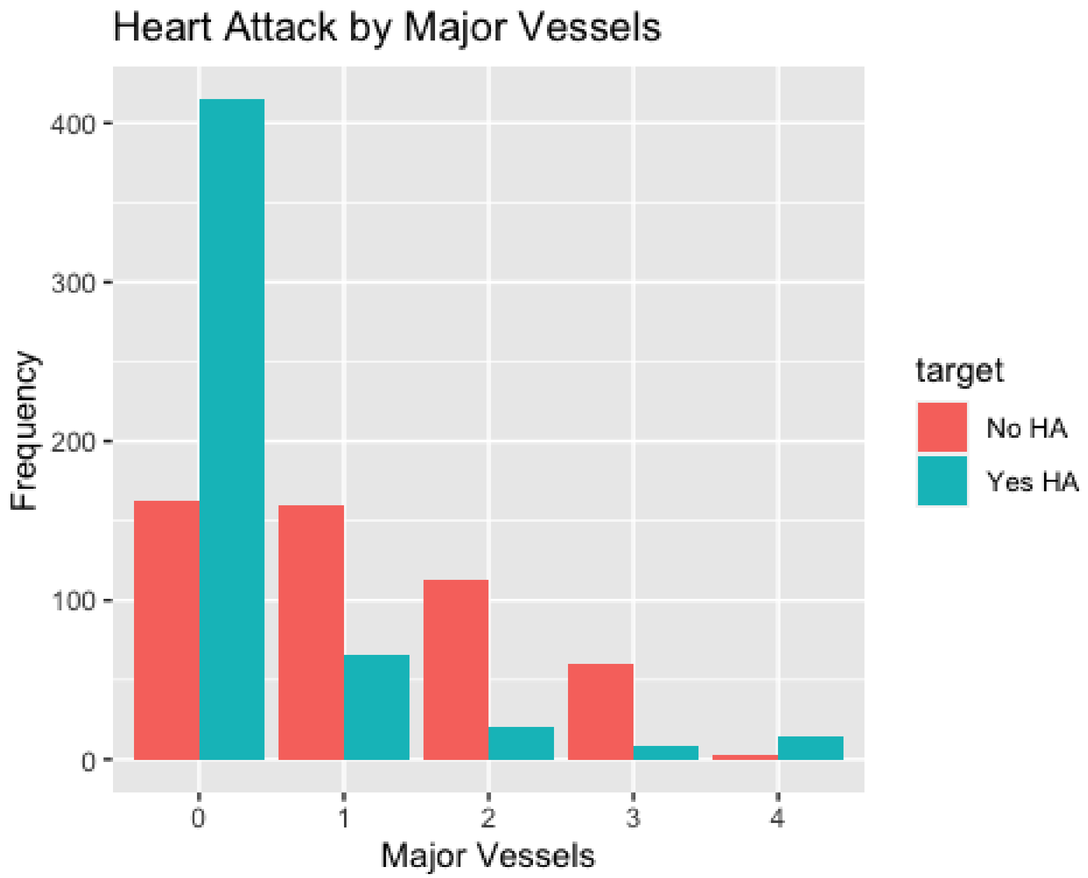
From this above graph we see that a cp value of ‘typical’ has more outcomes of no likely heart disease, while ‘atypical’, ‘non-anginal’, and ‘asymptomatic’ has a higher frequency of heart disease.

```
#bar plot - position = "dodge" makes them side by side
ggplot(dat,aes(x=thal,fill=target)) + geom_bar(position = "dodge") + labs(x =
"Thalassemia", y = "Frequency", title = "Heart Attack by Thalassemia")
```



From the above graph we see that an individual who has a ‘normal’ thal has a higher frequency of heart disease, while an individual who has ‘reversable’ has a higher frequency of No heart disease.

```
#bar plot - position = "dodge" makes them side by side
ggplot(dat,aes(x=ca,fill=target)) + geom_bar(position = "dodge") + labs(x =
"Major Vessels", y = "Frequency", title = "Heart Attack by Major Vessels")
```



We can see if someone doesn't have any major vessels, they are far more likely to have a heart attack.

Train and Test Data Split

First, we will divide our data into training (70%) and test data (30%).

```
#save the number of rows in the data set for use in the splitting code
n <- nrow(dat)

set.seed(202205) #make a split that's reproducible

#0's are test set and 1's are training set
tv.split <- sample(rep(0:1,c(round(n*.3), n-1*round(n*.3))),n)
table(tv.split)

## tv.split
##   0   1
## 308 717

dat.train <- dat[tv.split==1,]
dat.test <- dat[tv.split==0,]
```

When you run this, you should get 308 zeros and 717 ones.

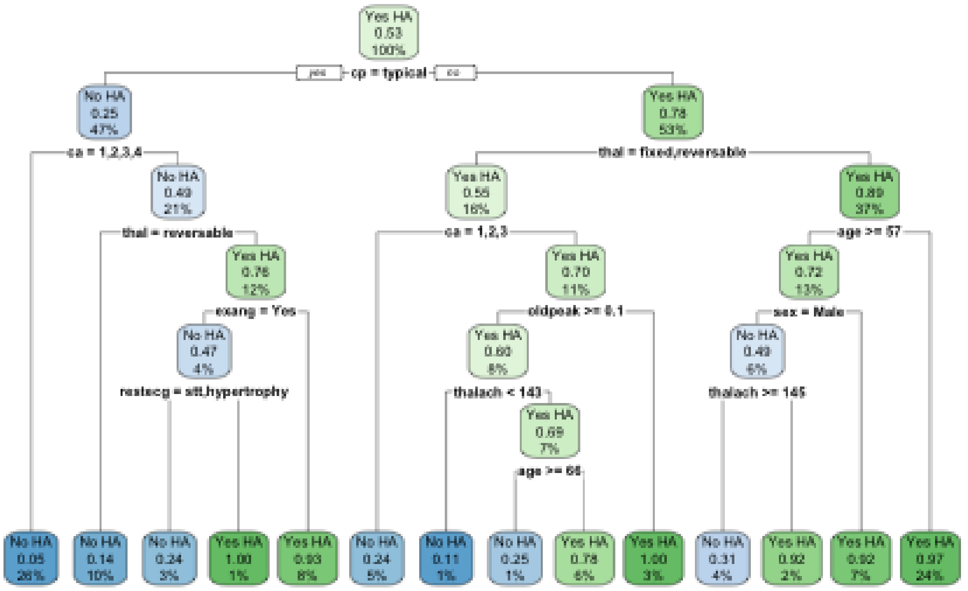
Classifiation Tree

Next, we will conduct a decision tree analysis. Inside the rpart library is the rpart() function that we will use for our analysis. rpart knows automatically when to perform regression or classification based on if the outcome variables is continuous or categorical. We will be creating a classification tree here. The key arguments of the rpart () function are:

- Formula: formula to specify the variable of interest, and the variables used for prediction (e.x. formula = Survived ~ Sex + Age).
- Method: Type of prediction you want, here using "class" is for categorical variables.
- Data: The data set to build the decision tree. We will use the training data here.

The function rpart.plot() in the rpart.plot library is used to create a fancy looking tree.

```
#first creating a tree with all the explanatory variables
tree <- rpart(target~., method = "class", data = dat.train)

rpart.plot(tree, sub="Classification tree for Heart Attack Prediction")
```



Classification tree for Heart Attack Prediction

Decision trees provide good visualization for showing the explanatory variables that have the greatest impact on the outcome variable. By looking at the classification tree here, we can see that the significant variables are cp, thal, ca, chol, age, exang, thalach, and oldpeak.

We can interpret the tree as such:

- The root node shows the proportion of people in the training data that actually have have a high chance of having a heart attack (target = 1). It is 53% of people here, hence the 0.53 in the node. The other number, 100%, means that 100% of our training data is considered at this node.

-The condition to split the root node is whether chest pain, “cp”, is “typical” or not. Any time a condition on the splitting is met, we go down the left branch of that decision node.

-Say, in this case, chest pain is “typical”, then we go down to the root’s left child node. We can interpret this node as 47% of our training data have chest pain as “typical” and their chance of heart attack is 25%.

-If a node has less than 50% chance of having a heart attack, the node’s title is “No HA” and the color gets changed to blue, though the interpretation of the node is still the percentage of people up who are more likely, “Yes HA”, to have a heart attack up until that point.

-From this node, we ask if a person had 1-4 major vessels “ca” and if they did, we move to down to the left most leaf node. We can interpret this node as people who have typical chest pain and 1-4 major vessels make up 26% of our data and their chance of heart attack is 5%. We can continue this analysis in a similar fashion throughout the entire tree.

Prediction

Now that we’ve had a chance to visualize the significant explanatory variables, we will conduct further analysis. Here, we are validating our model that was fit with training data to the test data. We do this to make our model more generalizable.

We want to predict which people are more likely to have a heart attack from the test set. We will use the predict() function here.

```
predicted <- predict(tree, dat.test, type = "class")
```

Confusion Matrix

We will now create a confusion matrix to see which people of the test data are actually more likely to have a heart attack, and which ones were predicted to be as such.

```
confusion.matrix <- table(Actual = dat.test$target, Predicted = predicted)
confusion.matrix

##          Predicted
## Actual    No HA Yes HA
##   No HA    147     18
##   Yes HA    25    118
```

Our model produced 118 true positives, 147 true negatives, 18 false positives, and 25 false negatives.

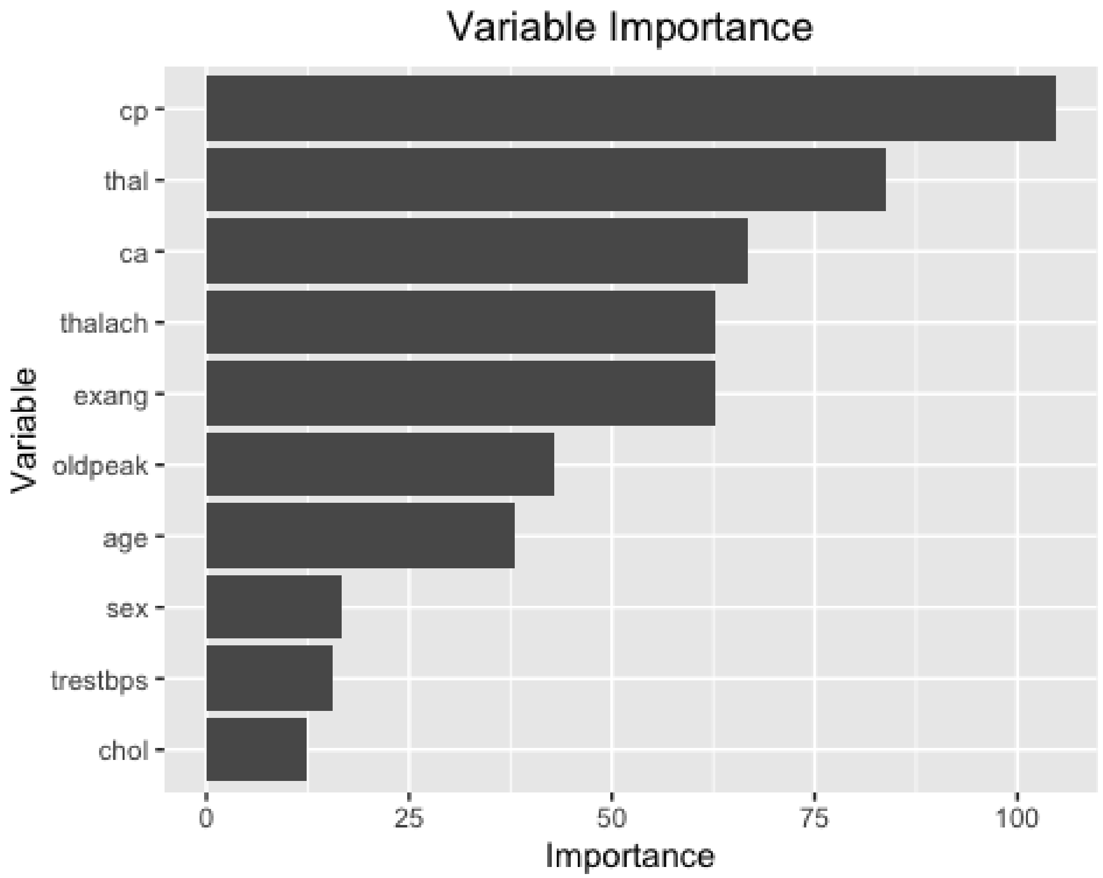
Now that we've conducted our confusion matrix, we will use it to compute the four indices of model fit.

```
#accuracy
accuracy <- sum(diag(confusion.matrix))/sum(confusion.matrix)
accuracy

## [1] 0.8603896

#precision
precision <- confusion.matrix[2,2]/sum(confusion.matrix[,2])
precision

## [1] 0.8676471

#recall
recall <- confusion.matrix[2,2]/sum(confusion.matrix[2,])
recall

## [1] 0.8251748

#F1 score
F1score <- 2*((precision*recall)/(precision+recall))
F1score

## [1] 0.8458781
```

Accuracy is 86%, precision is 86.8%, recall is 82.5% and F1 score is 84.6%. Precision is the ratio of true positives to the total true positives and false positives. This is pretty high, so the proportion of false positives is low here. Recall, on the other hand, is the ratio of true positives to the total true positives and false negatives. Recall is lower than precision because there are more false negatives in our model than false positives. We would rather have more false negatives than false positives because it would be better to be put into more likely to have HA group and not actually have a chance of HA, than to actually have a high chance of HA and not know or do anything about it. Thus, our tree model is a reasonable fit for our data.

Variable Importance

```
#feature importance using vip
vip(tree) + ylab('Importance') + xlab("Variable") +
  ggtitle("Variable Importance") + theme(plot.title = element_text(hjust =
0.5))
```



From the above graph we see that the most important variables that our model uses is cp, thal, and ca. Note that this can be seen in the actual decision tree visualization where these variables are some of the first the model looks at when predicting output.

Conclusion

Using the classification tree and variable importance plot above helped us to visualize the significant explanatory variables on whether someone is likely or not to have a heart attack. These variables are chest pain, thalassemia, and number of major vessels. If someone had chest pain other than "typical" they would be more likely to have a heart attack. For someone with "normal" thalassemia, they are more likely to have a heart attack. And for someone who doesn't have any major vessels, they are far more likely to have a heart attack. Additionally, we used the tree model that was fit with training data to the test data to predict the likeliness of a heart attack in the test data participants. After analyzing the confusion matrix, we conclude that our tree model is 86% accurate in predicting the likeliness of heart attack and because precision is higher than recall, our model is reasonable.