Building a Heart Disease Prediction Model

Harini Lakshmanan

Mohammad Mahmoudighaznavi

Verity Pierson

ADS 503: Applied Predictive Modeling

Summer 2023

Abstract

Heart disease, also known as cardiovascular disease, is a significant global concern which comes with both high morbidity and mortality rates. There are several conditions affecting the heart and the body's blood vessels, including coronary artery disease, heart failure, arrhythmia, valvular heart disease and congenital heart defects. The risk factors that contribute to heart disease are age, gender, family history, smoking, obesity, diabetes, unhealthy diet or high blood pressure or cholesterol levels. 47% of Americans live with at least one of these risk factors. (National Center for Chronic Disease Prevention and Health Promotion, 2023).

Machine learning (ML) is computational algorithms, which allow patterns to be identified in collected data. It is these patterns that will help medical professionals to know what to look for in order to diagnose and treat patients faster. ML helps to scan records of biometric data, analyze them and determine the risk factors of each patient (Javaid, Zghyer, Chang, Spaulding, Isakadze, Ding, Kargillis, Gao, Rahman, Brown, Saria, Martin, Kramer, Blumenthal, & Marvel, 2022)

This project shows how machine learning (ML) and artificial intelligence (AI) can help to diagnose people with heart disease conditions faster than traditional ways of diagnosis, based on the symptoms and results of tests. Twelve models were developed to include, Linear Discriminant Analysis (LDA), Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machine (Linear) SVM(Linear), Support Vector Machine (Radial Kernel) SVM (Radial Kernel), Random Forest (RF), Quadratic Discriminant Analysis (QDA), Gradient Boosting Machines (GBM), Bagged Trees, Neural Network, Nearest Shrunken Centroids (NSC), and Mixture Discriminant Analysis (MDA). Of those twelve models the Support Vector Machine (Radial Kernel) had the highest accuracy rate of 85.2%.

Keywords: Machine learning, artificial intelligence, model, conditions, diagnosis

Table of Contents

Abstract……………………………………………………………………………………….2

List of Tables……………………………………………………………………………..…4

List of Figures. ……………………………………………………………………………….5

List of Equations……………………………………………………………………….. 6

GitHub Repository Link……………….………………………….…………………….…...7

Problem Statement..……………………………………………………………………...7

Data description………………………………………………………………………….7

Exploratory Data Analysis (EDA)..9

Data Pre-Processing…...……………………………………………………………………17

Data Splitting………………………………………………………………………………18

Model Building Strategies……………………………………………………………………18

Linear Discriminant Analysis (LDA)…………………………………………………18

Logistic Regression…………………………………………………………………19

K-Nearest Neighbor (KNN)…………………………………………………………...19

Support Vector Machine (SVM) (Linear)…………………………………………….……19

Support Vector Machine (SVM) (Radial Kernel) …...…………………………………..20

Random Forest………………………………………………………………………...20

Quadratic Discriminant Analysis (QDA)……………………………………………...20

Gradient Boosting Machine (GBM)……………………………………………………..20

Bagging Tree……………………………………………………………………………….21

Neural Network……………………………………………………………………….21

Nearest Shrunken Centroids (NSC)…………………………………………………...21

Mixture Discriminant Analysis (MDA)……………………………………………………22

Model Performance and Hyperparameter Tuning ………………………………………………22

Results…………………………………………………………………………………...23

Discussion and Conclusion……………………………………………………………...24

Strengths and Weaknesses of this Study………………………………………………...25

References……………………………………………………………………………….26

Appendix………………………………………………………………………………...28

List of Tables

Table 1: Model Performance ………………………………………………………..……….22

List of Figures

Figure 1: *Histogram plot depicting the correlation between age and number of patients with and without heart disease*.……………………………………………...……………………….….……10

Figure 2: *Box plot displays the age distribution of heart diagnosis*...………………………….11

Figure 3: *Mosaic plot displays gender vs heart disease*.……..……………………….……..……12

Figure 4: *Geom_bar plot displays heart disease distributions by chest pain type*….…………….13

Figure 5: *Box plot displays heart disease distribution by resting blood pressure*……..……..……14

Figure 6: *Geom_bar plot displays heart disease distributions by number of major vessels*...….15

Figure 7: *Box plot displays heart disease distribution by serum cholesterol*.………...……...…16

Figure 8: *Box plot displays heart disease distributions by number of major vessels*.……….....…17

Figure 9: *Top Important Variables of SVM*……………………………………………….…..…24

List of Equations

Equations: Appendix………………..…………………………………..…..………………..……..26

GitHub Repository Link

Our GitHub repository can be found at: https://github.com/harinigautham/ADS_503_APM_Heart-Predictions. *It includes all* instructions to reproduce and deploy the models.

Problem Statement

Heart disease is a major health concern worldwide, with high mortality rates, which in turn has a significant impact on the patient's quality of life. Heart disease is the leading cause of death in the US, causing 25% of deaths each year. In other words, every 36 seconds one person dies due to cardiovascular disease (https://www.cdc.gov/heartdisease/facts.htm). Being able to detect heart disease early, with an accurate prediction can greatly help identify preventive measures, personalized treatments, and improved patient outcomes.

There are many factors that can increase the risk of getting heart disease. Some of the factors are out of control such as age, sex, family history, or heart shape. There are some factors that are controllable such as blood pressure, cholesterol level, smoking, and diabetes. This project intended to help us find out the important features that lead to heart disease based on available health and demographic information.

Data Description

The Heart Disease dataset was downloaded from the Kaggle datasets (https://www.kaggle.com/datasets/utkarshx27/heart-disease-diagnosis-dataset) and had 270 instances and 14 different variable features including the patient's heart activity, demographic/risk factors, and target variable. All the variables were taken into consideration initially for heart disease prediction. The attributes and brief information of each one is listed below:

age: age in years

sex: sex (1 = male; 0 = female)

chest.pain.type: chest pain type:

- Value 1: typical angina
- Value 2: atypical angina
- Value 3: non-anginal pain
- Value 4: asymptomatic

resting.blood.pressure (in mm Hg on admission to the hospital)

serum.cholestoral (in mg/dl)

fasting.blood.sugar (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

Resting.electrocardiographic.results:

- Value 0: normal
- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria.

max.heart.rate: maximum heart rate achieved

Exercise.induced.angina (1 = yes; 0 = no)

Oldpeak: ST depression induced by exercise relative to rest.

ST.segment: the slope of the peak exercise ST segment

- Value 1: upsloping
- Value 2: flat
- Value 3: downsloping

Major.vessels: number of major vessels (0-3) colored by fluoroscopy.

Thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

Heart.disease: diagnosis of heart disease (Absence (1) or presence (2) of heart disease.

Exploratory Data Analysis (EDA)

The heart disease prediction data was extracted from Kaggle.com, which is an online community for data scientists and machine learning engineers. Kaggle provides datasets for the creation of Artificial Intelligence (AI) models, allows collaboration and holds competitions for the solution of data science challenges. It is the data science competition that started Kaggle back in 2010 (Mahmoud, 2022) and is a subsidiary of Google. This data set was chosen because of its insight into real-world solutions to a real-world medical problem.

The first step required for producing our EDA is to understand the data and identify which predictors are not only required but provide meaningful information. This in turn will help to develop charts that help to tell a story. Through this process it was determined that the dataset did not have any missing data, nor did it have any NAs.

Figure 1 shows the breakdown of the number of patients who have heart disease, which are colored in blue and those without heart disease, colored in pink. The blue columns show patients who have heart disease based on their age. The data shows a moderate skew to the left, as the majority of patients with heart disease are between the ages of 55 and 70 years of age. The data was grouped by age and heart disease and shows a positive correlation between the variables. Patients without heart disease have a more normal distribution with only a slight skew to the left.

Figure 1

Histogram plot depicting the correlation between age and number of patients with and without heart disease.

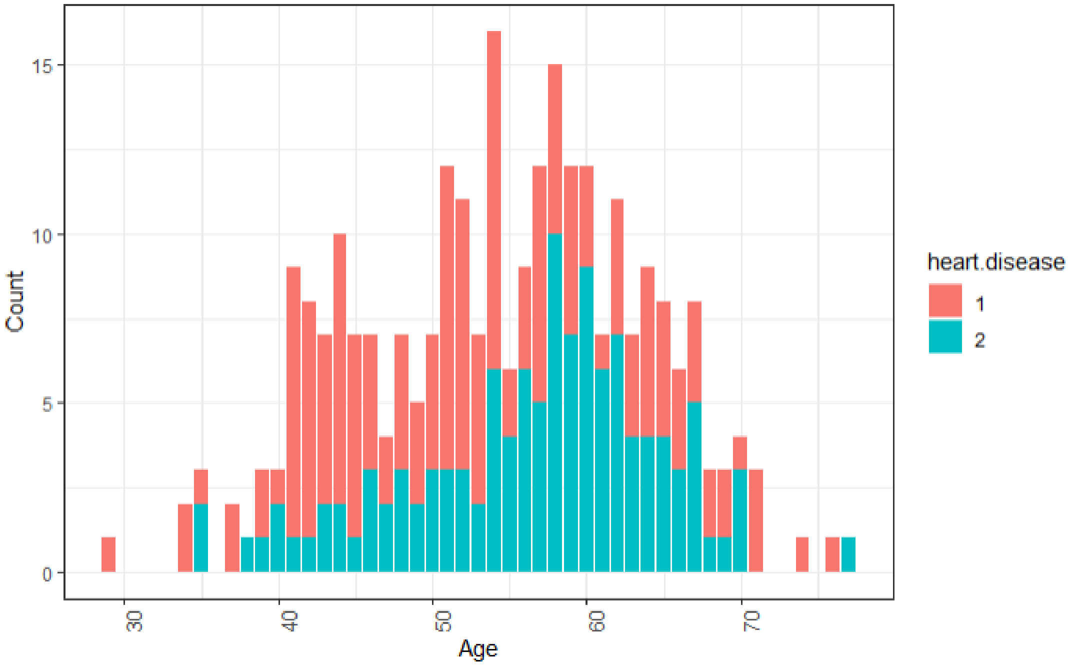
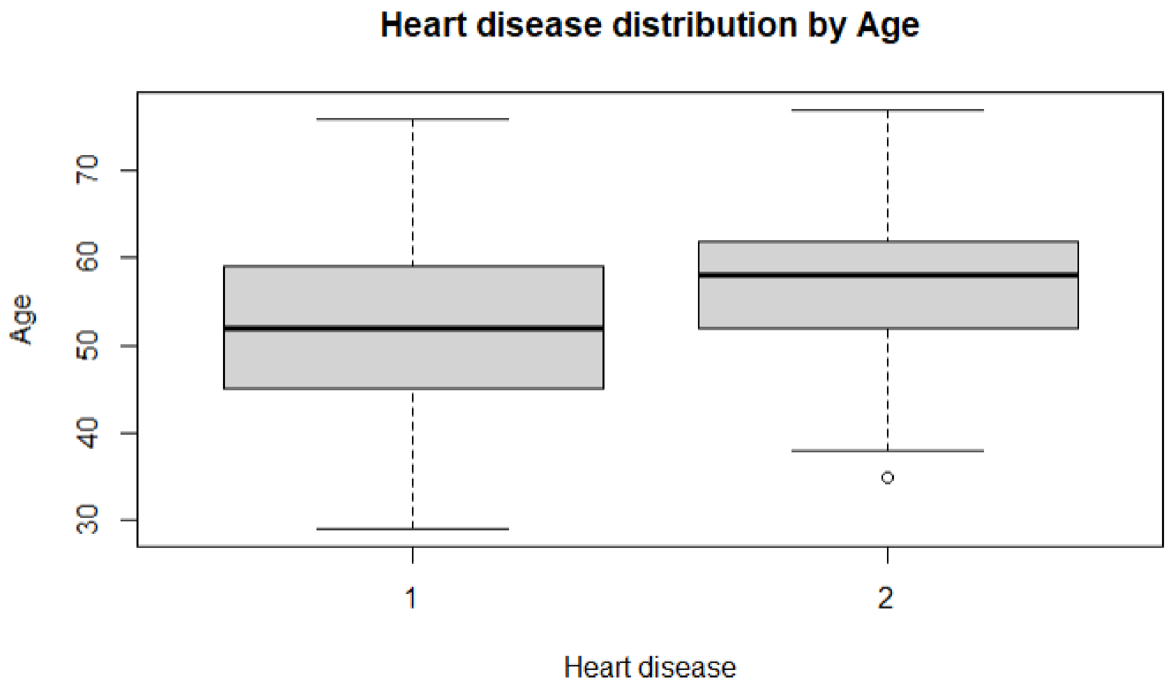


Figure 2 shows a box plot which displays how age is distributed between those with and without heart disease. The box plot shows the presence of outliers in the “with heart disease” data, showing some evidence that younger people in the dataset have been diagnosed with heart disease. The mean of those with heart disease are around 55 years of age and we can see the skewing found in the histogram is also present in the box plot.

Figure 2

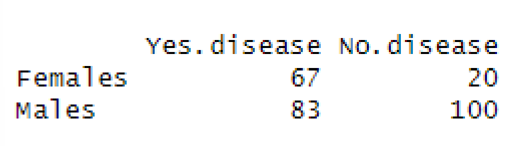
Box plot displays the age distribution of heart diagnosis.



Additional breakdown of heart disease and gender is shown in Figure 3. The mosaic plot breaks down the heart disease vs gender spread of data. This plot shows more males (1) than females (0) have been positively diagnosed with heart disease (2).

Figure 3

Mosaic plot and Contingency Table displays gender vs heart disease.



```
           Yes.disease No.disease
Females             67         20
Males               83        100
```

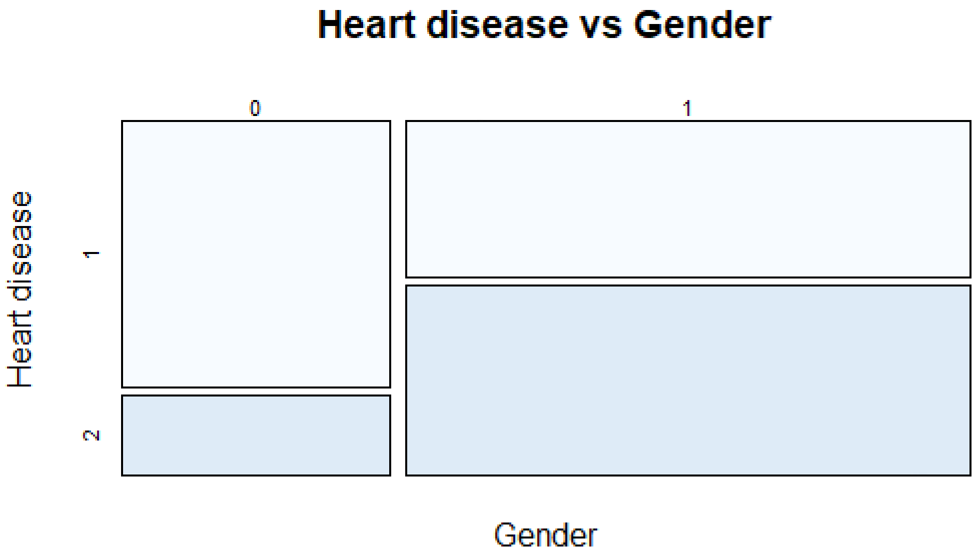


Figure 4 is a geom bar chart, which provides insight into the breakdown of chest pain type and the heart disease severity. Through the original dataset, the chest pain types are as follows, 1 = typical angina, 2 = atypical angina, 3 = non-anginal pain and 4 = asymptomatic. The chart shows approximately 80% of the time heart disease patients have asymptomatic chest pain.

Figure 4

Geom_bar plot displays heart disease distributions by chest pain type.

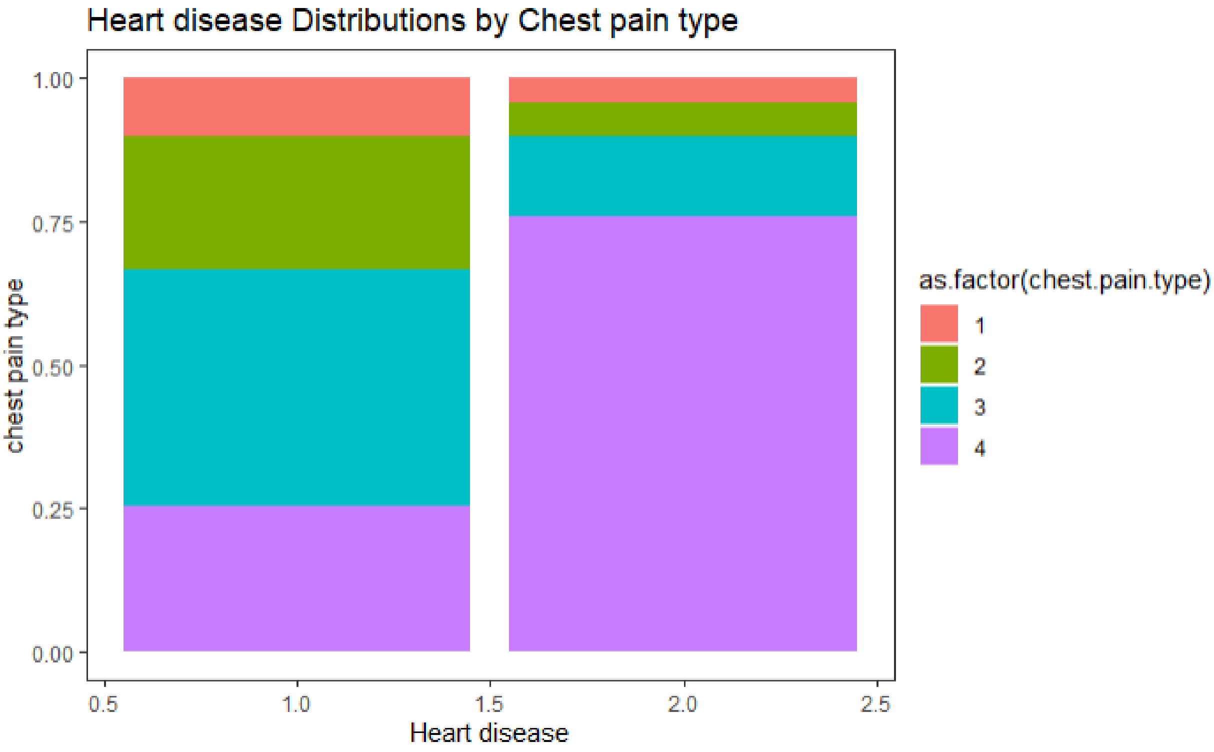


Figure 5 uses a box plot to explain the relationship between heart disease diagnosis and resting blood pressure. Here it is observed that outliers can be seen in both those with and those without heart disease. The resting blood pressure in both cases shows a mean of approximately 130 bpm. Resting blood pressure is shown to be higher in patients with heart disease, which is to be expected.

Figure 5

Box plot displays heart disease distribution by resting blood pressure.

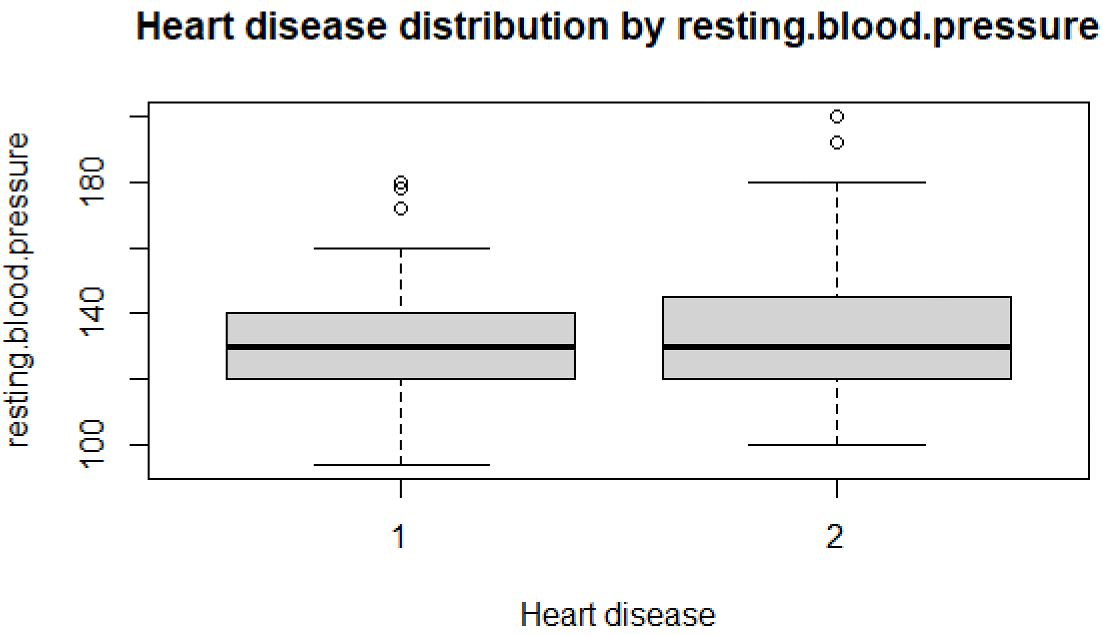


Figure 6 provides a look at the breakdown between heart disease distributions by number of major vessels colored by fluoroscopy during diagnosis. As expected, most patients with no heart disease have no major vessels blocked and therefore are colored pink in this figure, while on the Green = 1 major vessel are blocked, Blue = 2 major vessels are blocked and Purple = 3 major vessels are blocked. The higher number of major vessels colored shows the increased risk of heart disease.

Figure 6

Geom_bar plot displays heart disease distributions by number of major vessels.

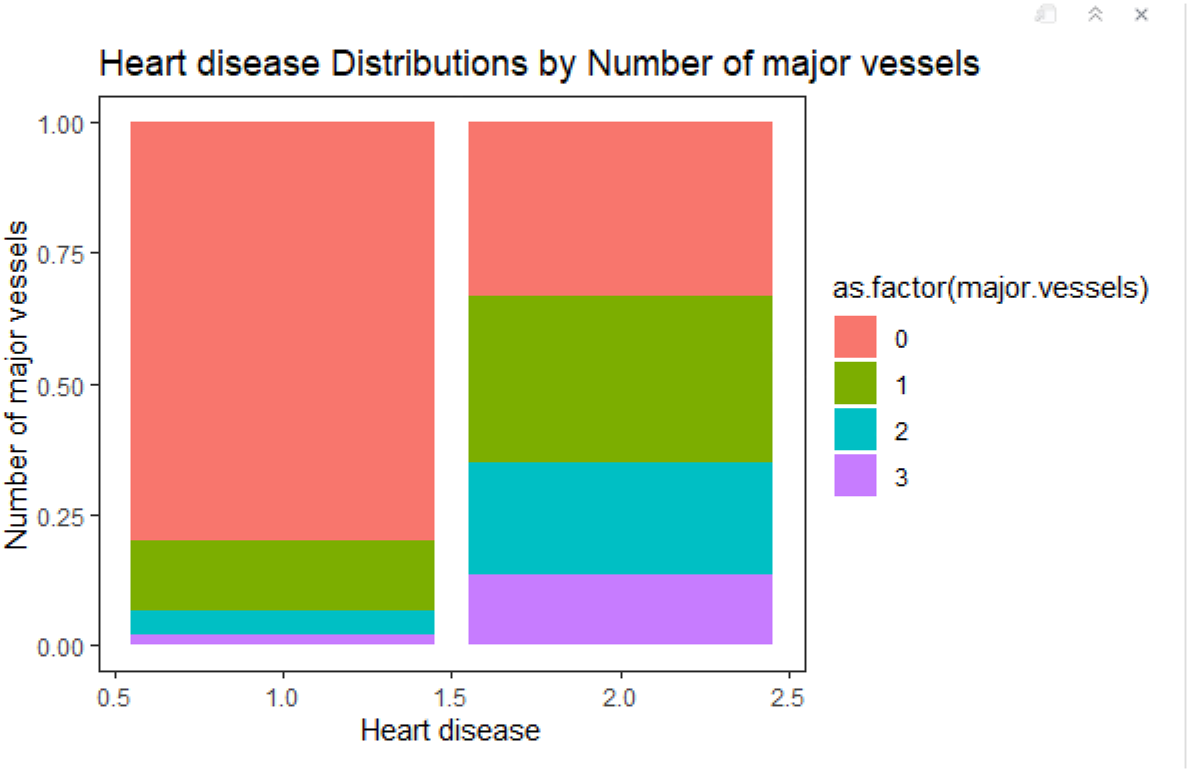


Figure 7 shows the relationship between heart disease and serum cholesterol. Studies have shown that when there are high levels of cholesterol in the blood there is a higher risk of heart disease. High levels of serum cholesterol showed additional need for treatment to reduce the risk for heart disease (Jousilahti, Vartiainen, Pekkanen, Tuomilehto, Sundvall, & Puska, 1998). The figure 7, shows some outliers, especially in the group without heart disease, which might be a precursor to those patients being positively diagnosed in the future due to the high levels of serum cholesterol.

Figure 7

Box plot displays heart disease distribution by serum cholesterol.

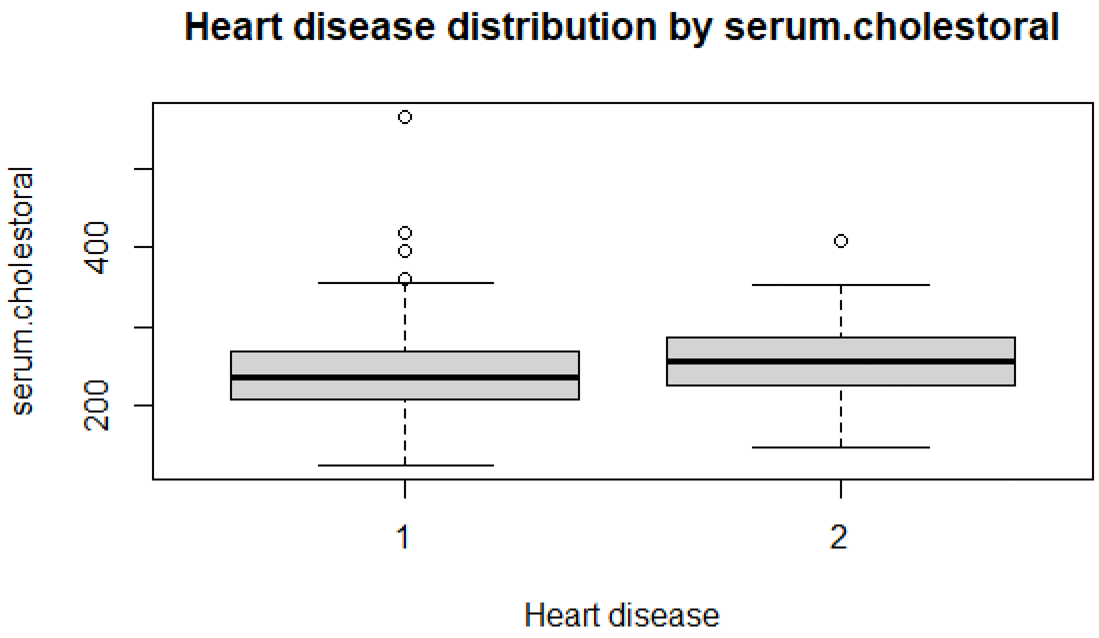
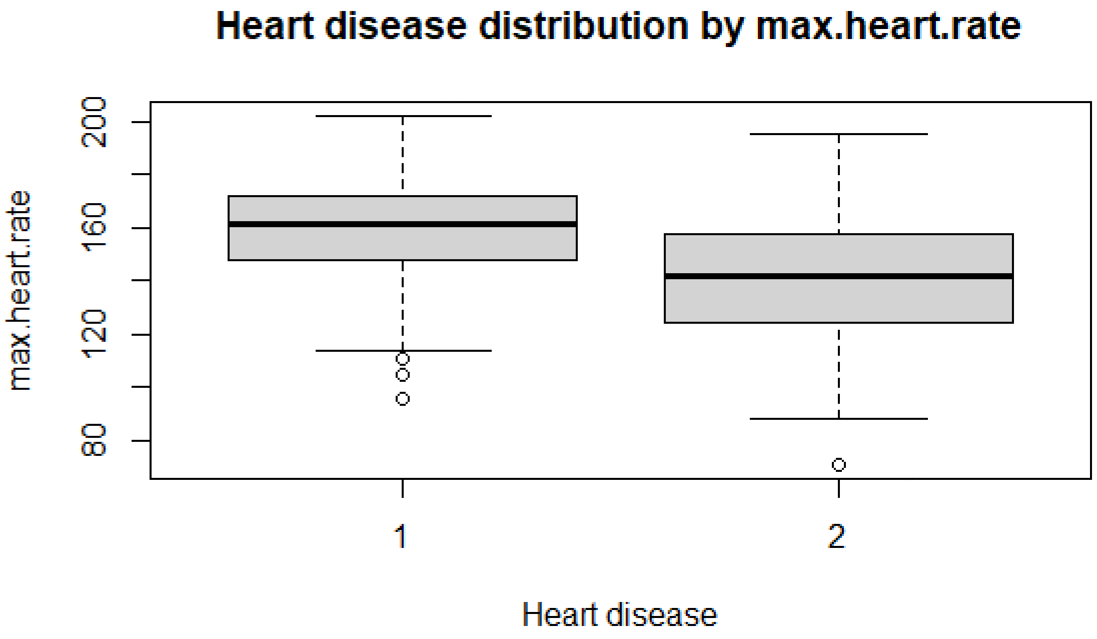


Figure 8 shows the distribution of max heart rate for those patients with and without heart disease. As the plot shows there are some outliers for both those with and those without heart disease. For patients with heart disease, the heart rate is less than those who do not have heart problems. There are some observed outliers in the data, but they are insignificant and were chosen to leave them in.

Figure 8

Box plot displays heart disease distributions by number of major vessels.



Data Pre-Processing

The raw Heart Disease Prediction dataset was downloaded from Kaggle.com (*https://www.kaggle.com/datasets/utkarshx27/heart-disease-diagnosis-dataset*) brought into RStudio as heart_df. The dataset was then checked to see if there are any nulls or NAs. No nulls or NAs were found. During review of the information on the dataset from Kaggle.com, it was found the dataset had been pre-processed prior to extraction. The dataset was cleaned up, nulls and NAs taken care of, and scale and centering was completed. A summary was run on the data to see the number of variables for each column. This allowed for the EDA charts to be planned out and identified.

Data Splitting

To find the best model, the dataset was split into training and test sets. The test set acts as an evaluator and indicator of potential overfitting issues. This dataset is split into 80% training records and 20% testing records. The stratified method is used to split the data to ensure both train and test datasets get enough records of both sides.

Model Building Strategies

The goal of our project is to have the model with the highest accuracy and sensitivity. Sensitivity is equally important because the objective is to predict as many true positives as possible and low false negatives. Higher the sensitivity lowers the chance that we end up in a situation where the prediction says that a patient is predicted that they are not at risk for heart disease, but actually get one.

Linear Discriminant Analysis (LDA):

The first model we experimented with was Linear Discriminant analysis. By calculating the likelihood that a fresh batch of data belongs to each class, LDA creates predictions. A forecast is made for the output class that has the highest likelihood. Since our problem is a pure classification problem, LDA is one of the great tools for prediction of heart disease. The test data resulted in an Accuracy of 0.8519 and sensitivity is 0.9032.

Logistic Regression:

Logistic regression is also a classification model we model the likelihood of a discrete result or output variable given an input variable and classified based on probability and it is a simple model to use and we can understand the impact of each predictor variable may have on heart disease. This model returned an Accuracy of 0.8519 and a Sensitivity of 0.9032.

K Nearest Neighbor (KNN):

KNN classification is a nonlinear classification problem opposed to our first two classification problems. It classifies the data based on distance metrics like Euclidean and k-nearest samples. In our algorithm we tried a range of k values from 1 to 19 for the classification and found out that the k value of 19 had the best accuracy of 0.8704 and sensitivity is 0.9355.

Support vector machine (SVM) (Linear):

SVM works similar to LDA and the basic concept behind both are the same, that is to find an optimal hyperplane to classify the data and here has only binary classification required for our problem. We first experimented with the linear SVM method in R which uses a linear hyperplane to classify similar to LDA to train our model. The amount that you want to prevent misclassifying each training example is specified by the C hyperparameter, which is used in SVM optimization, and we used the C value of 0.01 which gives the best accuracy of 0.8519 and sensitivity of 0.9032.

Support vector machine (SVM) (Radial Kernel):

SVM Radial creates the hyperplane boundary using a radial basis function instead of a linear plane in the SVM Linear model. It creates a much more complex boundary to classify, and we had an improvement in accuracy from 0.8519 and sensitivity of 0.9355.

Random Forest:

A popular supervised machine learning algorithm for classification issues is random forest. On several samples, it constructs decision trees and uses the majority decision to classify the data. This test data models gives Accuracy of 0.7778 and sensitivity of 0.8387.

Quadratic Discriminant Analysis (QDA):

Unlike LDA, QDA creates a quadratic decision boundary instead of linear using a bayes classifier. Using our 13 predictors we got an accuracy of 0.87 and sensitivity of 0.861 using Quadratic discriminant analysis, this model returned Accuracy of 0.8333 and sensitivity of 0.9032.

Gradient Boosting Machines (GBM):

A machine learning method called gradient boosting is used, among other things, for classification and regression tasks. For classification, a powerful predicting model is created using the gradient boosting classifier by combining many weak learning models like decision trees. The model gives Accuracy of 0.7222 and sensitivity of 0.7097.

Bagged Tree:

The next model we used is the bagged tree classification. Making bootstrap samples from the training data set, building trees on those samples, aggregating the results from all the trees, and forecasting the results are the steps involved in decision tree bagging. The test model gives Accuracy of 0.8333 and Sensitivity of 0.8710.

Neural Network:

Neural networks have a collection of hidden units or variables that are linear combinations of predictors. These are then subjected to a nonlinear transformation before being connected to the result by yet another linear combination. The number of hidden units/nodes and decay are the two key hyperparameters since these have a high likelihood of overfitting the data. Decay reduces the magnitude of each parameter estimate utilized in the equation, resulting in less overfitting and more rounded decision limits. Based on iteration we arrived at number of hidden units to be 2 and decay to be 0.4 for the best accuracy of 0.8148 and sensitivity of 0.8387.

Nearest Shrunken Centroids (NSC):

A straightforward classifier called NSC operates under the premise that samples from the same class must roughly lie on the same subspace. It uses a hyperparameter called the shrinkage threshold and for NSC classifiers, the threshold parameter is crucial since it controls how many variables are employed in the classification rule and how much the centroids are decreased. Based on iteration in R the optimal shrinkage threshold chosen was 0.862 and it obtained an accuracy of 0.8333 and a sensitivity of 0.9032 in the test data set.

Mixture Discriminant Analysis (MDA):

Each class is thought to originate from a single normal (or Gaussian) distribution by the LDA classifier and hence this is limiting. Each class in MDA is assumed to be a Gaussian mixture of subclasses, and each data point is assigned a probability of being a member of each class. The assumption remains that all classes' covariance matrices are equal similar to LDA. For MDA we had an accuracy of 0.8519 for the test, not too different from LDA as we are solving only a binary classification problem.

Model Performance and Hyperparameter Tuning

Table 1 provides the results for the train and test set. Having results in one table, helps us see which models has the best performances and which ones have potential of overfitting.

Table 1

Model Performance.

Models	Train Accuracy	Kappa	Test Accuracy	Kappa	Sensitivity
LDA	0.8140843	0.6223315	0.8519	0.6936	0.9032
Logistic Regression	0.8151988	0.6263616	0.8519	0.6936	0.9032
KNN	0.7821856	0.5599854	0.8704	0.7304	0.9355
SVM(Linear)	0.8422430	0.6750859	0.8519	0.6936	0.9032
SVM (Radial Kernel)	0.8513289	0.6966396	0.8519	0.7304	0.9355
Random Forest	0.8195158	0.6304878	0.7778	0.5404	0.8387
QDA	0.7769824	0.5515716	0.8333	0.6534	0.9032

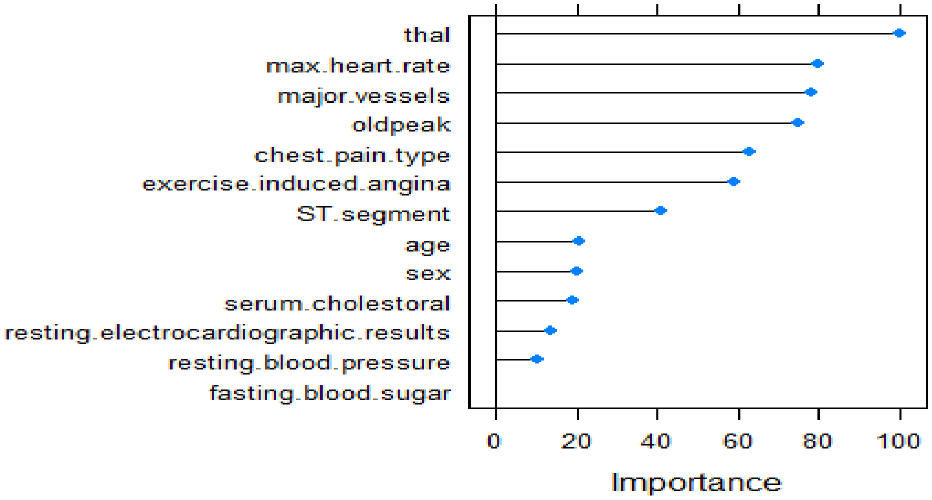
GBM	0.8658210	0.7249885	0.7222	0.4414	0.7097
Bagged Trees	0.8144317	0.6221302	0.8333	0.6573	0.8710
Neural Network	0.8423437	0.6797597	0.8148	0.6213	0.8387
NSC	0.8422380	0.6752968	0.8333	0.6534	0.9032
MDA	0.8422430	0.6791175	0.8519	0.6936	0.9032

Results

The main goal of our project is to classify patients based on the thirteen different metrics and our primary goal in our model is to have the best accuracy and the next is to have the least false negatives, that is high sensitivity. LDA, logistic regression, KNN, SVM models and MDA had accuracy greater than 85% and two of the models that stood out were KNN and SVM radial due to their high sensitivity value of 93%. Even though KNN had better test accuracy than SVM, its train accuracy was poor. When we iterated with different seed values to randomize the test train split, SVM always had better sensitivity and accuracy overall compared to KNN and hence our final chosen model is SVM radial for this classification problem. Since sensitivity is critical for this type of medical classification problem, SVM radial had a better overall model compared to all the classification models that we performed for this project. Figure 9 arranges the predictors of SVM radial, based on importance. Thalassemia is the most important predictor in this case.

Figure 9

Top Important Variables of SVM.



Discussion and Conclusion

Based on the available data that we used for this project it is highly recommended to use SVM radial model for initial classification of patients for heart disease as it had an accuracy of 85% and a sensitivity of 93%. One of the limitations of this dataset that we observed was the low sample size. This led to variation of model accuracies based on the randomization of the test train split at random. For future improvements, a larger sample size would help in determining a more accurate model as we move forward and minimize the impact of test train split randomization on model performance. SVM radial model performed the best despite this limitation and provided consistently the best result irrespective of the randomization and hence we recommend using this model for screening patients based on risk of a potential heart disease.

Strengths and Weaknesses of this Study

The strength of this study is the models performed well with the model with the highest accuracy level being SVM radial model at 85% with a sensitivity of 93%. A model which has an accuracy of between 80 – 90 % is considered an excellent model. Anything over 90% would suggest that there is overfitting (Wilame, 2020).

The weakness we have in our data set is the small number of observations. The number of observations in the data set for this study was 270 with 14 variables. With this small number of observations, the data set is good for proof of concept, but a larger number of observations would be needed for production (Gonfalonieri, 2019). There is also room for additional variables such as the patient’s BMI, smoking and family history, to be included. The dataset would also benefit from the inclusion of data from different hospitals, different cultures and different demographics to include location, race, and ethnicities.

References

Gonfalonieri, A. (2019, June 3). *Dealing with the lack of data in machine learning*. Medium. https://medium.com/predict/dealing-with-the-lack-of-data-in-machine-learning-725f2abd2b92

Javaid, A., Zghyer, F., Chang, K., Spaulding, E., Isakadze, N., Ding, J., Kargillis, D., Gao, Y., Rahman, F., Brown, D., Saria, S., Martin, S., Kramer, C., Blumenthal, R., & Marvel, F. (2022, August 29). *Medicine 2032: The Future of Cardiovascular Disease Prevention with Machine Learning and Digital Health Technology*. American Journal of Preventive Cardiology. https://www.sciencedirect.com/science/article/pii/S2666667722000630

Jousilahti, P., Vartiainen, E., Pekkanen, J., Tuomilehto, J., Sundvall, J., & Puska, P. (1998, March 24). *Serum cholesterol distribution and coronary heart disease risk: Observations and predictions among middle-aged population in Eastern Finland*. Circulation. https://pubmed.ncbi.nlm.nih.gov/9531256/

Kuhn, M., & Johnson, K. (2016). *Applied Predictive Modeling.* springer.

Mahmoud, M. H. (2022). *What is a kaggle?: Data Science and Machine Learning*. Kaggle. https://www.kaggle.com/general/328265

National Center for Chronic Disease Prevention and Health Promotion, D. f. (2023, May). *Heart Disease Facts*. Retrieved from Centers for Disease Control and Prevention: https://www.cdc.gov/heartdisease/facts.htm

Wilame. (2020, June 15). *Why you should not trust only in accuracy to measure machine learning performance*. Medium. https://medium.com/@limavallantin/why-you-should-not-trust-only-in-accuracy-to-measure-machine-learning-performance-a72cf00b4516#:~:text=If%20your%20’X’%20value%20is%20between%2080%25%20and%2090,a%20probably%20an%20overfitting%20case.

Appendix

Heart Disease Predictions

Team 2: Verity Pierson, Harini Lakshmanan, Mohammad Mahmoudighaznavi

2023-06-03

```
knitr::opts_chunk$set(echo = TRUE)

library(mlbench)
library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units

library(e1071)

##
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':
##
##     impute

library(caret)

##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
##library(tidyr)
library(corrplot)

## corrplot 0.92 loaded

library(AppliedPredictiveModeling)
library(car)

## Loading required package: carData

library(lattice)
library(lars)

## Loaded lars 1.3

library(stats)
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:corrplot':
##
##     corrplot

## The following object is masked from 'package:caret':
##
##     R2

## The following object is masked from 'package:stats':
##
##     loadings

library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:Hmisc':
##
##     src, summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(gbm)
```

```
## Loaded gbm 2.1.8.1
```

```r
library(earth)
```

```
## Loading required package: plotmo
```

```
## Loading required package: plotrix
```

```
## Loading required package: TeachingDemos
```

```
##
## Attaching package: 'TeachingDemos'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     cnvrt.coords, subplot
```

```
library(plotmo)
library(plotrix)
library(TeachingDemos)

#load heart disease dataset
heart_df <- read.table(file.choose(), header=TRUE, sep=",")
head(heart_df)
```

```
##   age sex chest.pain.type resting.blood.pressure serum.cholestoral
## 1  70   1               4                    130               322
## 2  67   0               3                    115               564
## 3  57   1               2                    124               261
## 4  64   1               4                    128               263
## 5  74   0               2                    120               269
## 6  65   1               4                    120               177
##   fasting.blood.sugar resting.electrocardiographic.results max.hear
t.rate
## 1                   0                                    2
109
## 2                   0                                    2
160
## 3                   0                                    0
141
## 4                   0                                    0
105
## 5                   0                                    2
121
## 6                   0                                    0
140
##   exercise.induced.angina oldpeak ST.segment major.vessels thal hea
rt.disease
## 1                       0     2.4          2             3    3
2
## 2                       0     1.6          2             0    7
1
## 3                       0     0.3          1             0    7
2
## 4                       1     0.2          2             1    7
1
## 5                       1     0.2          1             1    3
1
## 6                       0     0.4          1             0    7
1
```

```
#lets see how dataset look like
str(heart_df)
```

```
## 'data.frame':    270 obs. of  14 variables:
##  $ age                                 : int  70 67 57 64 74 65 56 
59 60 63 ...
##  $ sex                                 : int  1 0 1 1 0 1 1 1 1 0 .
..
##  $ chest.pain.type                     : int  4 3 2 4 2 4 3 4 4 4 .
..
##  $ resting.blood.pressure              : int  130 115 124 128 120 1
20 130 110 140 150 ...
##  $ serum.cholestoral                   : int  322 564 261 263 269 1
77 256 239 293 407 ...
##  $ fasting.blood.sugar                 : int  0 0 0 0 0 0 1 0 0 0 .
..
##  $ resting.electrocardiographic.results: int  2 2 0 0 2 0 2 2 2 2 .
..
##  $ max.heart.rate                      : int  109 160 141 105 121 1
40 142 142 170 154 ...
##  $ exercise.induced.angina             : int  0 0 0 1 1 0 1 1 0 0 .
..
##  $ oldpeak                             : num  2.4 1.6 0.3 0.2 0.2 0
.4 0.6 1.2 1.2 4 ...
##  $ ST.segment                          : int  2 2 1 2 1 1 2 2 2 2 .
..
##  $ major.vessels                       : int  3 0 0 1 1 0 1 1 2 3 .
..
##  $ thal                                : int  3 7 7 7 3 7 6 7 7 7 .
..
##  $ heart.disease                       : int  2 1 2 1 1 1 2 2 2 2 .
..
```

```
summary(heart_df)
```

```
##       age             sex          chest.pain.type resting.blood.pre
ssure
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0  
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0  
##  Median :55.00   Median :1.0000   Median :3.000   Median :130.0  
##  Mean   :54.43   Mean   :0.6778   Mean   :3.174   Mean   :131.3  
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0  
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0  
##  serum.cholestoral fasting.blood.sugar resting.electrocardiographic
.results
##  Min.   :126.0     Min.   :0.0000      Min.   :0.000              
##  1st Qu.:213.0     1st Qu.:0.0000      1st Qu.:0.000              
##  Median :245.0     Median :0.0000      Median :2.000              
##  Mean   :249.7     Mean   :0.1481      Mean   :1.022              
##  3rd Qu.:280.0     3rd Qu.:0.0000      3rd Qu.:2.000  
##  Max.   :564.0     Max.   :1.0000      Max.   :2.000  
##  max.heart.rate  exercise.induced.angina    oldpeak       ST.segmen
t
##  Min.   : 71.0   Min.   :0.0000          Min.   :0.00   Min.   :1.0
00
##  1st Qu.:133.0   1st Qu.:0.0000          1st Qu.:0.00   1st Qu.:1.0
00
##  Median :153.5   Median :0.0000          Median :0.80   Median :2.0
00
##  Mean   :149.7   Mean   :0.3296          Mean   :1.05   Mean   :1.5
85
##  3rd Qu.:166.0   3rd Qu.:1.0000          3rd Qu.:1.60   3rd Qu.:2.0
00
##  Max.   :202.0   Max.   :1.0000          Max.   :6.20   Max.   :3.0
00
##  major.vessels         thal         heart.disease  
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:1.000  
##  Median :0.0000   Median :3.000   Median :1.000  
##  Mean   :0.6704   Mean   :4.696   Mean   :1.444  
##  3rd Qu.:1.0000   3rd Qu.:7.000   3rd Qu.:2.000  
##  Max.   :3.0000   Max.   :7.000   Max.   :2.000
```

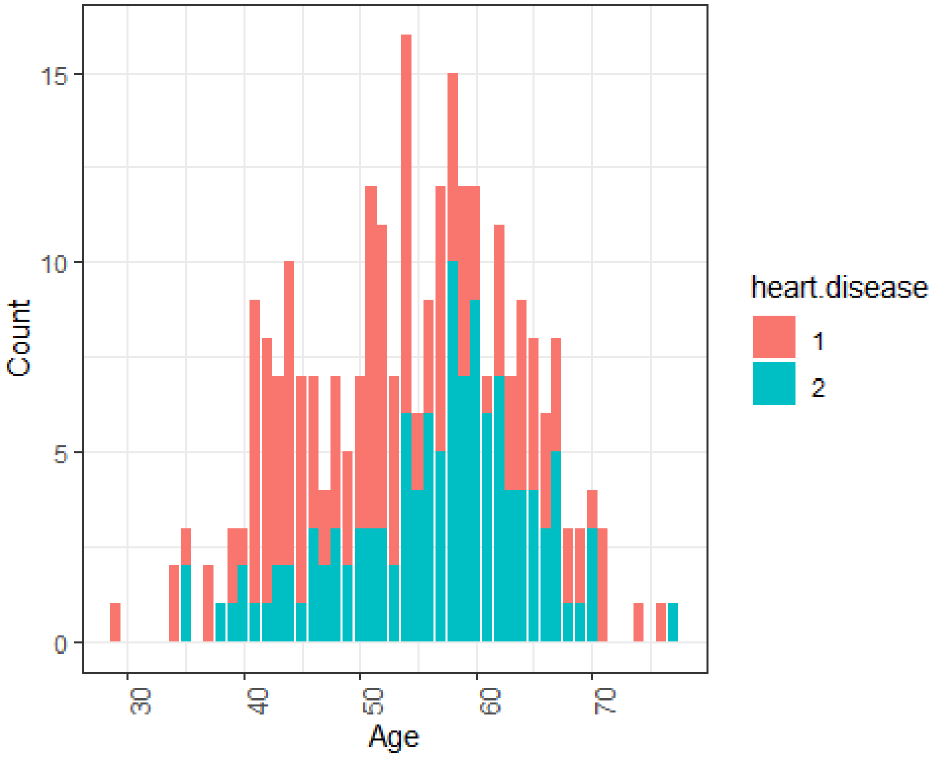
```
#checking the NA values
sum(is.na(heart_df))
```

```
## [1] 0
```

```
# check Distinct values


heart_df %>%
  summarise(n_age = n_distinct(age), n_sex = n_distinct(sex),
            n_chestpain = n_distinct(chest.pain.type),
            n_restbp=n_distinct(resting.blood.pressure),
            n_chol = n_distinct(serum.cholestoral),
            n_fastbs = n_distinct(fasting.blood.sugar),
            n_restecg = n_distinct(resting.electrocardiographic.result
s),
            n_HR= n_distinct(max.heart.rate),
            n_exercise = n_distinct(exercise.induced.angina),
            n_oldpeak = n_distinct(oldpeak),
            n_STsegment = n_distinct(ST.segment),
            n_mvessels = n_distinct(major.vessels),
            n_thal = n_distinct(thal),
            n_heartdisease = n_distinct(heart.disease))
```

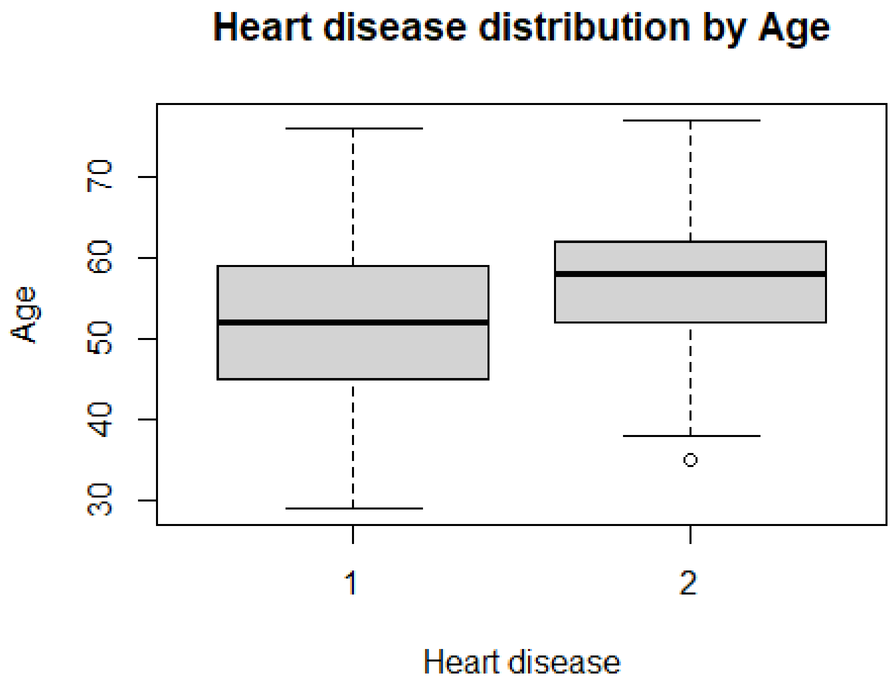
```
##   n_age n_sex n_chestpain n_restbp n_chol n_fastbs n_restecg n_HR n
_exercise
## 1    41     2           4       47    144        2         3   90
2
##   n_oldpeak n_STsegment n_mvessels n_thal n_heartdisease
## 1        39           3          4      3              2
```

```
#age distribution vs heart disease plot
heart_df %>% group_by(age, heart.disease) %>% summarise(count = n()) %
>%
  ggplot() + geom_bar(aes(age, count,   fill = as.factor(heart.disease
)), stat = "Identity") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, size = 10)) +
  ylab("Count") + xlab("Age") + labs(fill = "heart.disease")

## `summarise()` has grouped output by 'age'. You can override using t
he `.groups`
## argument.
```



Blue plot which is the presence of heart disease is left skewed which shows age has a positive correlation with heart disease.

```
#boxplot to displays the age distribution of heart diagnosis
boxplot(heart_df$age ~ heart_df$heart.disease,
        main="Heart disease distribution by Age",
         ylab="Age",xlab="Heart disease")
```



```
#Gender analysis
#The proportion of females and males patients in the dataset.

heart_df %>%
    group_by( sex ) %>%
    summarise( percent = 100 * n() / nrow( heart_df ))

## # A tibble: 2 x 2
##     sex percent
##   <int>   <dbl>
## 1     0    32.2
## 2     1    67.8
```

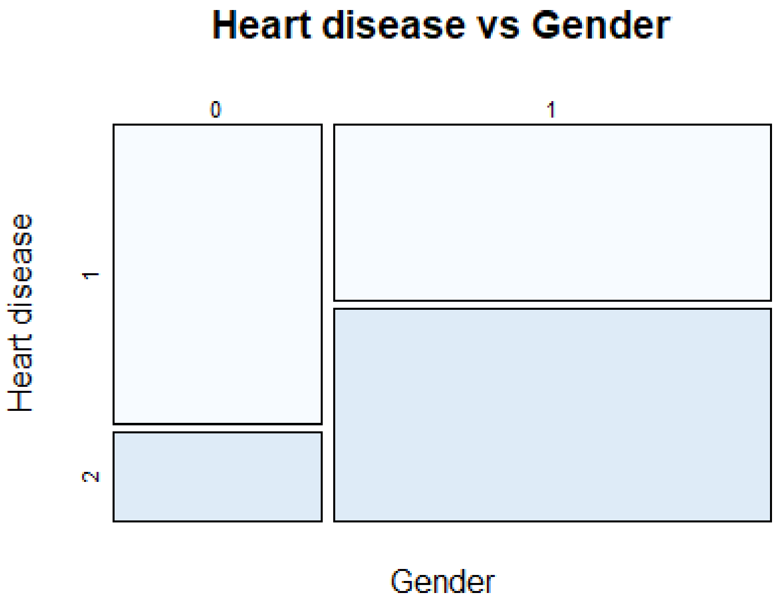
There are 32.2 % females and 67.8% males in the dataset

```
#Check the percentage of males and females with heart disease

female_yes <- table(heart_df[heart_df$sex==0,]$heart.disease)
male_yes <- table(heart_df[heart_df$sex==1,]$heart.disease)
FMcombine_yes <- rbind(female_yes,male_yes)

#Rename columns names and rows names.
colnames(FMcombine_yes) <- c("Yes.disease", "No.disease")
```

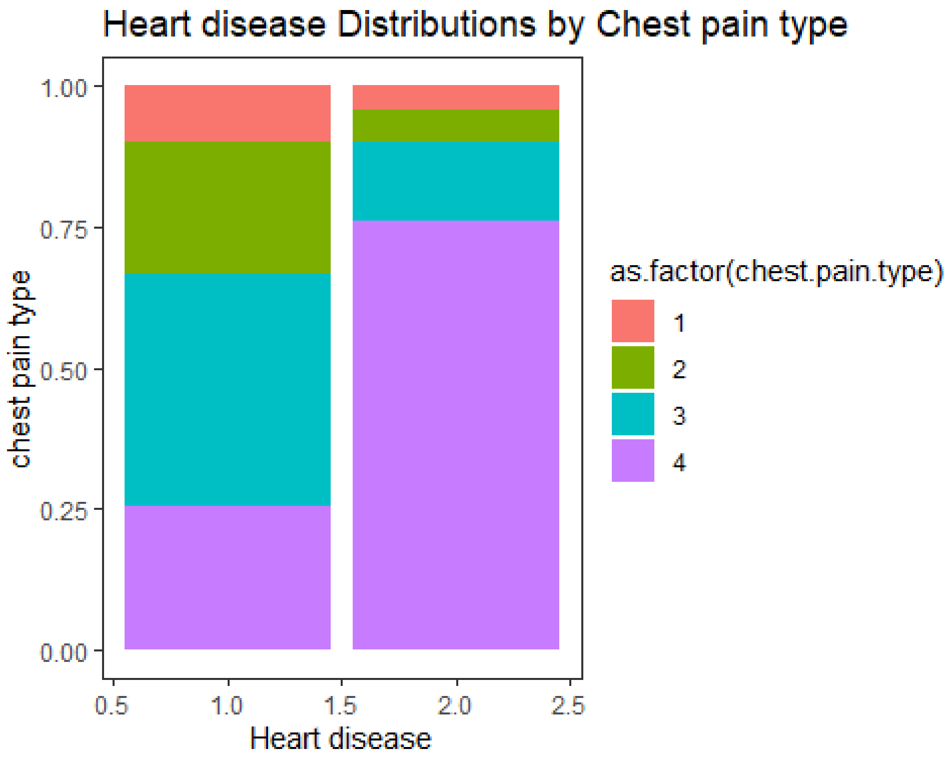
```
rownames(FMcombine_yes) <- c("Females", "Males")

#Display the table
FMcombine_yes

##          Yes.disease No.disease
## Females           67         20
## Males             83        100
```

There are 67 females out of 87 who have diagnosed with heart disease and 83 males out of 183 were diagnosed with heart disease.

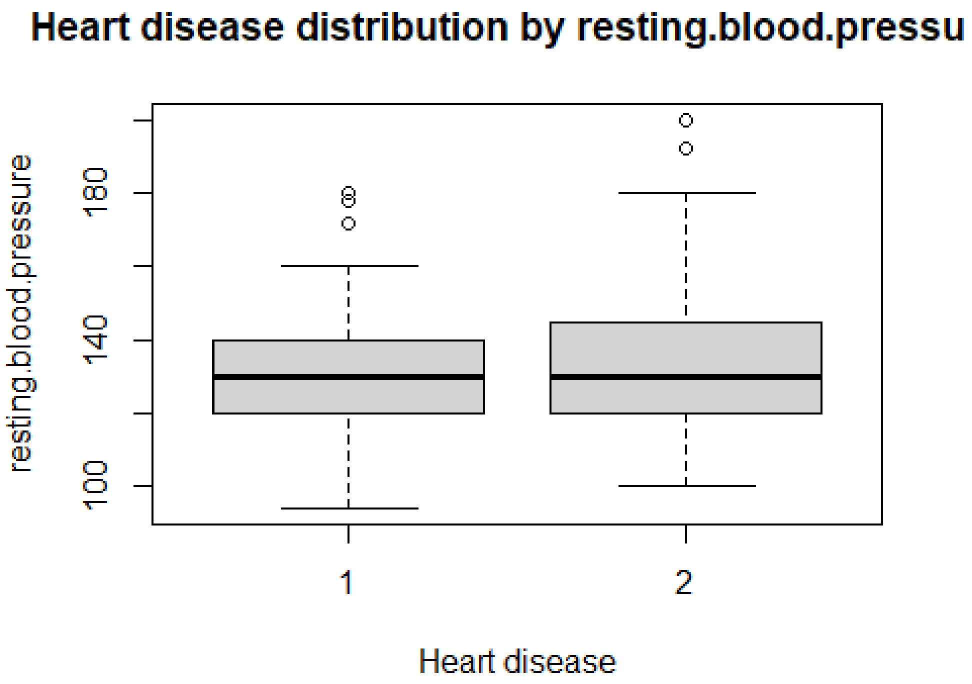
```
#mosaic plot for gender vs heart.disease
mosaicplot(heart_df$sex ~ heart_df$heart.disease,
           main="Heart disease vs Gender", shade=FALSE,color=blues9,
           xlab="Gender", ylab="Heart disease")
```



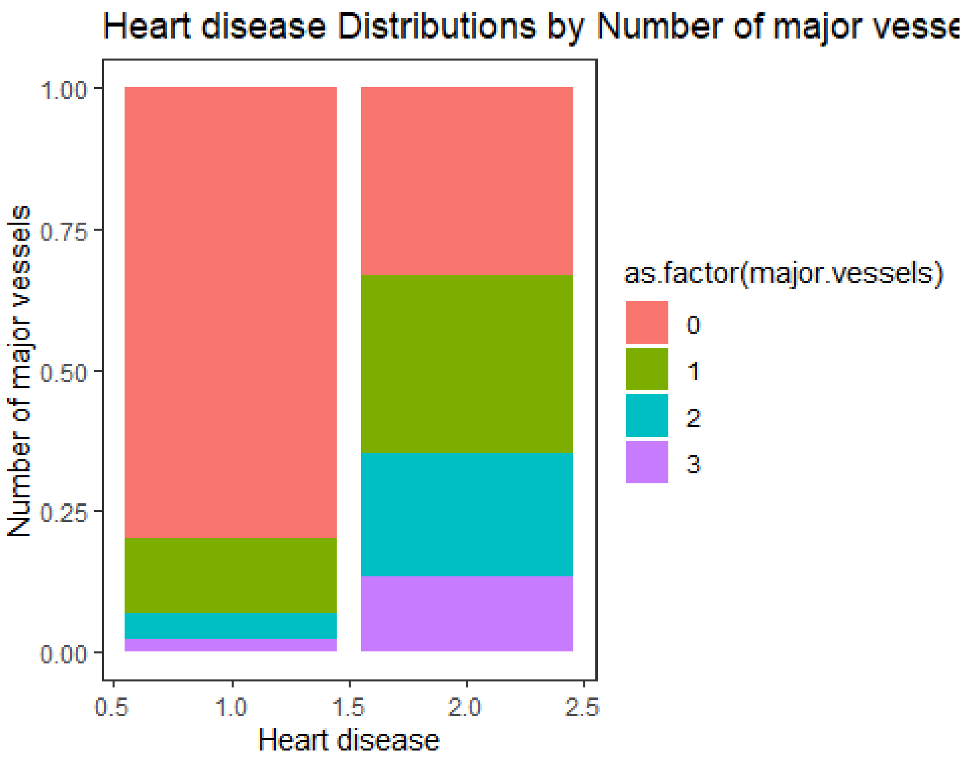
```
#chest pain type analysis
#heart_df$chest.pain.type <- as.factor(heart_df$chest.pain.type)

ggplot(data = heart_df, aes(x = heart.disease, fill = as.factor(chest.
pain.type))) +
  geom_bar(position = "fill") +
  labs(title = "Heart disease Distributions by Chest pain type",
       x = "Heart disease",
       y = "chest pain type") +
  theme_test()
```



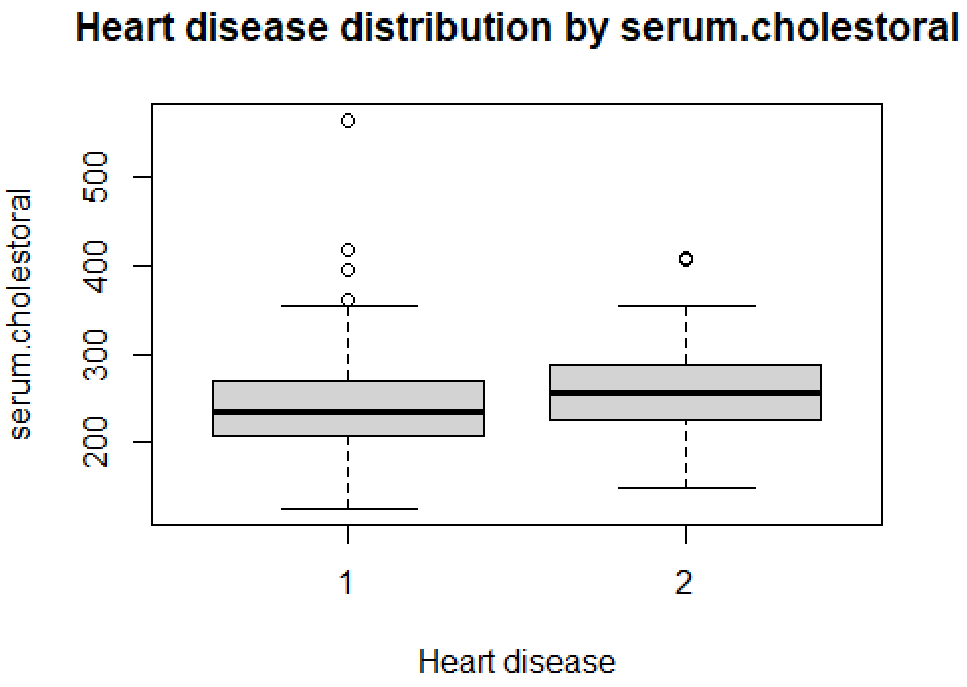
```
#Rest Blood Pressure distribution vs heart disease plot
boxplot(heart_df$resting.blood.pressure ~ heart_df$heart.disease,
        main="Heart disease distribution by resting.blood.pressure",
         ylab="resting.blood.pressure",xlab="Heart disease")
```



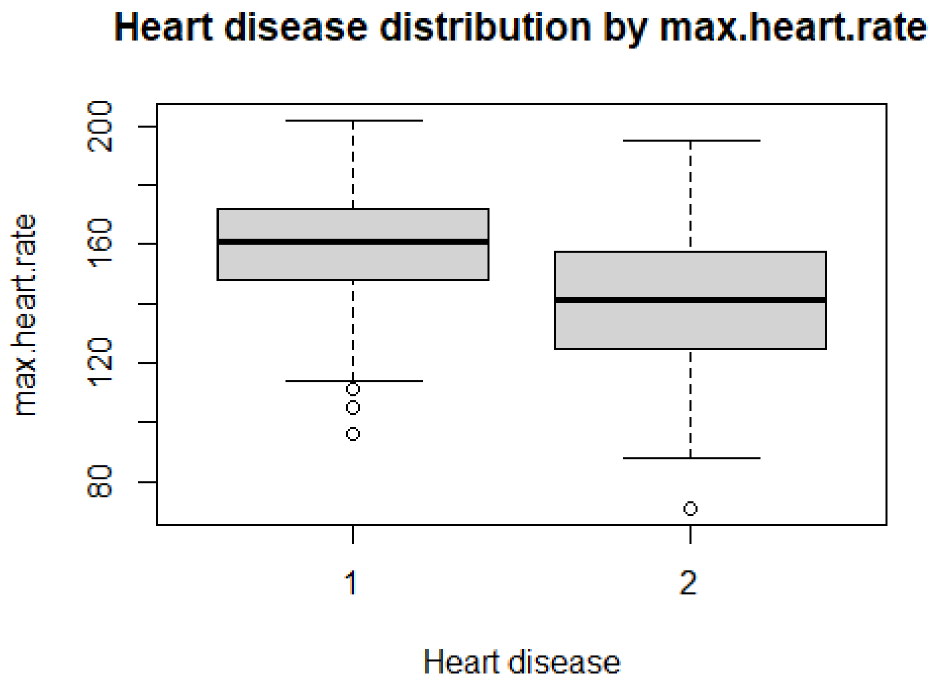
```
#major blood vessels vs heart.disease analysis
#heart_df$major.vessels <- as.factor(heart_df$major.vessels)

ggplot(data = heart_df, aes(x = heart.disease, fill = as.factor(major.
vessels))) +
  geom_bar(position = "fill") +
  labs(title = "Heart disease Distributions by Number of major vessels
",
       x = "Heart disease",
       y = "Number of major vessels") +
  theme_test()
```



```
#serum.cholestoral distribution vs heart disease plot

boxplot(heart_df$serum.cholestoral ~ heart_df$heart.disease,
        main="Heart disease distribution by serum.cholestoral",
         ylab="serum.cholestoral",xlab="Heart disease")
```



```
#max.heart.rate distribution vs heart disease plot

boxplot(heart_df$max.heart.rate ~ heart_df$heart.disease,
        main="Heart disease distribution by max.heart.rate",
         ylab="max.heart.rate",xlab="Heart disease")
```

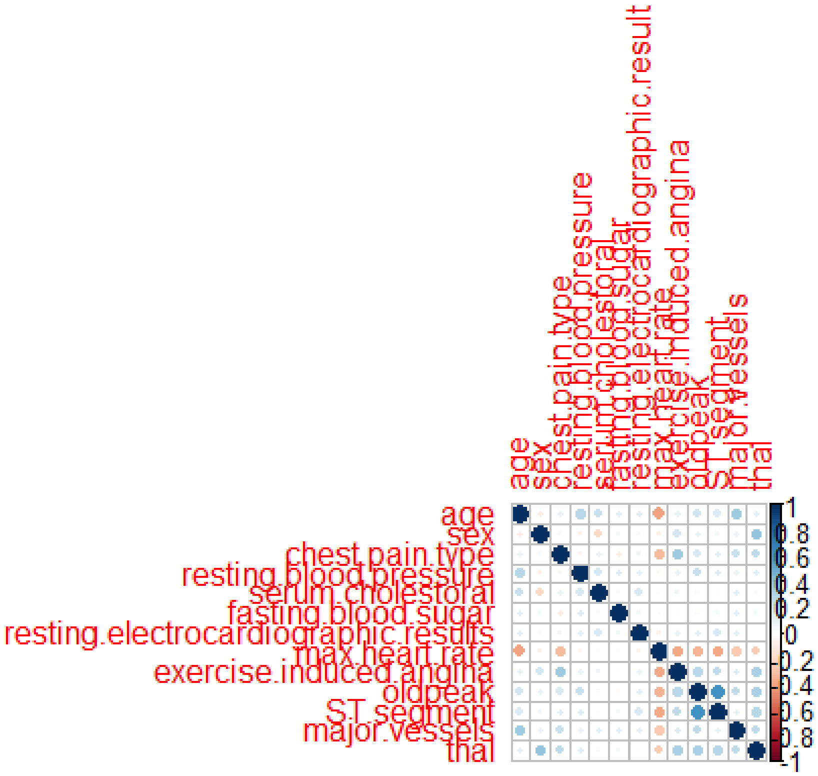


```
#correlations
corr <- cor(heart_df[,1:13])

round(corr,2)

##                                          age   sex chest.pain.type
## age                                   1.00 -0.09            0.10
## sex                                  -0.09  1.00            0.03
## chest.pain.type                       0.10  0.03            1.00
## resting.blood.pressure                0.27 -0.06           -0.04
## serum.cholestoral                     0.22 -0.20            0.09
## fasting.blood.sugar                   0.12  0.04           -0.10
## resting.electrocardiographic.results  0.13  0.04            0.07
## max.heart.rate                       -0.40 -0.08           -0.32
## exercise.induced.angina               0.10  0.18            0.35
## oldpeak                               0.19  0.10            0.17
## ST.segment                            0.16  0.05            0.14
## major.vessels                         0.36  0.09            0.23
## thal                                  0.11  0.39            0.26
##                                      resting.blood.pressure serum.c
holestoral
## age                                                    0.27
0.22
## sex                                                   -0.06
-0.20
```

```
## chest.pain.type                                       -0.04
0.09
## resting.blood.pressure                                 1.00
0.17
## serum.cholestoral                                      0.17
1.00
## fasting.blood.sugar                                    0.16
0.03
## resting.electrocardiographic.results                   0.12
0.17
## max.heart.rate                                        -0.04
-0.02
## exercise.induced.angina                                0.08
0.08
## oldpeak                                                0.22
0.03
## ST.segment                                             0.14
-0.01
## major.vessels                                          0.09
0.13
## thal                                                   0.13
0.03
##                                      fasting.blood.sugar
## age                                               0.12
## sex                                               0.04
## chest.pain.type                                  -0.10
## resting.blood.pressure                            0.16
## serum.cholestoral                                 0.03
## fasting.blood.sugar                               1.00
## resting.electrocardiographic.results              0.05
## max.heart.rate                                    0.02
## exercise.induced.angina                           0.00
## oldpeak                                          -0.03
## ST.segment                                        0.04
## major.vessels                                     0.12
## thal                                              0.05
##                                      resting.electrocardiographic.r
esults
## age
0.13
## sex
0.04
## chest.pain.type
0.07
## resting.blood.pressure
0.12
```

```
## serum.cholestoral
0.17
## fasting.blood.sugar
0.05
## resting.electrocardiographic.results
1.00
## max.heart.rate
-0.07
## exercise.induced.angina
0.10
## oldpeak
0.12
## ST.segment
0.16
## major.vessels
0.11
## thal
0.01
##                                      max.heart.rate exercise.induce
d.angina
## age                                           -0.40
0.10
## sex                                           -0.08
0.18
## chest.pain.type                               -0.32
0.35
## resting.blood.pressure                        -0.04
0.08
## serum.cholestoral                             -0.02
0.08
## fasting.blood.sugar                            0.02
0.00
## resting.electrocardiographic.results          -0.07
0.10
## max.heart.rate                                 1.00
-0.38
## exercise.induced.angina                       -0.38
1.00
## oldpeak                                       -0.35
0.27
## ST.segment                                    -0.39
0.26
## major.vessels                                 -0.27
0.15
## thal                                          -0.25
0.32
```

```
##                                      oldpeak ST.segment major.vesse
ls  thal
## age                                     0.19       0.16          0.
36  0.11
## sex                                     0.10       0.05          0.
09  0.39
## chest.pain.type                         0.17       0.14          0.
23  0.26
## resting.blood.pressure                  0.22       0.14          0.
09  0.13
## serum.cholestoral                       0.03      -0.01          0.
13  0.03
## fasting.blood.sugar                    -0.03       0.04          0.
12  0.05
## resting.electrocardiographic.results    0.12       0.16          0.
11  0.01
## max.heart.rate                         -0.35      -0.39         -0.
27 -0.25
## exercise.induced.angina                 0.27       0.26          0.
15  0.32
## oldpeak                                 1.00       0.61          0.
26  0.32
## ST.segment                              0.61       1.00          0.
11  0.28
## major.vessels                           0.26       0.11          1.
00  0.26
## thal                                    0.32       0.28          0.
26  1.00

#plot correlations
corrplot::corrplot(cor(heart_df[, 1:13]))
```



```
#split dataset
set.seed(502)
trainingrows <- createDataPartition(heart_df$heart.disease, p=0.8, lis
t=FALSE)
heart_train <- heart_df[trainingrows,]
heart_test <- heart_df[-trainingrows,]

#preprocess including center and scale
heart_trainimp <- preProcess(heart_train, "knnImpute")
heart_trainpredict <- predict(heart_trainimp, heart_train)
heart_testpredict <- predict(heart_trainimp, heart_test)


summary(heart_trainpredict)

##       age                 sex          chest.pain.type   resting.blo
od.pressure
##  Min.   :-2.89006   Min.   :-1.5375   Min.   :-2.2568   Min.   :-2.
1064
##  1st Qu.:-0.75103   1st Qu.:-1.5375   1st Qu.:-0.1691   1st Qu.:-0.
6760
##  Median : 0.07605   Median : 0.6474   Median :-0.1691   Median :-0.
1258
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.
0000
```

```
##  3rd Qu.: 0.67498   3rd Qu.: 0.6474   3rd Qu.: 0.8747   3rd Qu.: 0.
4243
##  Max.   : 2.47176   Max.   : 0.6474   Max.   : 0.8747   Max.   : 3.
7254
##  serum.cholestoral  fasting.blood.sugar resting.electrocardiographi
c.results
##  Min.   :-2.56329   Min.   :-0.4237     Min.   :-0.9931
##  1st Qu.:-0.70670   1st Qu.:-0.4237     1st Qu.:-0.9931
##  Median :-0.09481   Median :-0.4237     Median :-0.4919
##  Mean   : 0.00000   Mean   : 0.0000     Mean   : 0.0000
##  3rd Qu.: 0.59552   3rd Qu.:-0.4237     3rd Qu.: 1.0117
##  Max.   : 3.52421   Max.   : 2.3494     Max.   : 1.0117
##  max.heart.rate    exercise.induced.angina    oldpeak          ST.s
egment
##  Min.   :-3.3497   Min.   :-0.7055         Min.   :-0.9184   Min.
:-0.9674
##  1st Qu.:-0.6304   1st Qu.:-0.7055         1st Qu.:-0.9184   1st Qu
.:-0.9674
##  Median : 0.1495   Median :-0.7055         Median :-0.2521   Median
: 0.6037
##  Mean   : 0.0000   Mean   : 0.0000         Mean   : 0.0000   Mean
: 0.0000
##  3rd Qu.: 0.7503   3rd Qu.: 1.4109         3rd Qu.: 0.5806   3rd Qu
.: 0.6037
##  Max.   : 2.1731   Max.   : 1.4109         Max.   : 4.2448   Max.
: 2.1748
##  major.vessels          thal         heart.disease
##  Min.   :-0.7203   Min.   :-0.8958   Min.   :-0.9008
##  1st Qu.:-0.7203   1st Qu.:-0.8958   1st Qu.:-0.9008
##  Median :-0.7203   Median :-0.8958   Median :-0.9008
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3169   3rd Qu.: 1.1518   3rd Qu.: 1.1050
##  Max.   : 2.3914   Max.   : 1.1518   Max.   : 1.1050
```

##Linear Discriminant Analysis

```
#LDA
lda_fit <- train(as.factor(heart.disease) ~ ., method = "lda", data =
heart_train)
lda_fit

## Linear Discriminant Analysis
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
```

```
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 216, 216, 216, 216, 216, 216, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8140843  0.6223315
```

```
lda_predict <- predict(lda_fit, heart_test)
```

```
confusionMatrix(lda_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 28  5
##          2  3 18
##
##                Accuracy : 0.8519
##                  95% CI : (0.7288, 0.9338)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 1.182e-05
##
##                   Kappa : 0.6936
##
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.9032
##             Specificity : 0.7826
##          Pos Pred Value : 0.8485
##          Neg Pred Value : 0.8571
##              Prevalence : 0.5741
##          Detection Rate : 0.5185
##    Detection Prevalence : 0.6111
##       Balanced Accuracy : 0.8429
##
##        'Positive' Class : 1
##
```

##Logistic Regression

```
set.seed(503)
lr_fit <- train(as.factor(heart.disease) ~ ., method = "glm", data = h
eart_train)
lr_fit
```

```
## Generalized Linear Model
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 216, 216, 216, 216, 216, 216, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8151988  0.6263616
```

```
lr_predict <- predict(lr_fit, heart_test)
confusionMatrix(lr_predict, as.factor(heart_test$heart.disease))
```

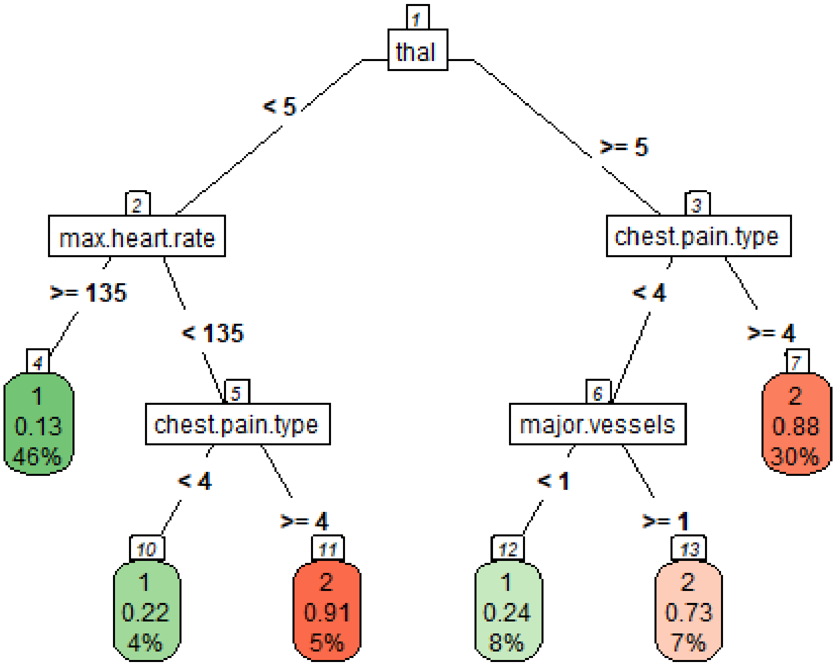
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 28  5
##          2  3 18
##
##                 Accuracy : 0.8519
##                   95% CI : (0.7288, 0.9338)
##      No Information Rate : 0.5741
##      P-Value [Acc > NIR] : 1.182e-05
##
##                    Kappa : 0.6936
##
##  Mcnemar's Test P-Value : 0.7237
##
##              Sensitivity : 0.9032
##              Specificity : 0.7826
##           Pos Pred Value : 0.8485
##           Neg Pred Value : 0.8571
##               Prevalence : 0.5741
##           Detection Rate : 0.5185
##     Detection Prevalence : 0.6111
##        Balanced Accuracy : 0.8429
##
##         'Positive' Class : 1
##
```

```
#rpart
set.seed(503)
rpart_fit <- train(as.factor(heart.disease) ~ ., method = "rpart", dat
a = heart_train)
rpart_fit
```

```
## CART
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 216, 216, 216, 216, 216, 216, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.02577320  0.7500499  0.4935970
##   0.04639175  0.7398428  0.4724009
##   0.48453608  0.6605814  0.3077145
##
## Accuracy was used to select the optimal model using the largest val
ue.
## The final value used for the model was cp = 0.0257732.
```

```
rpart_predict <- predict(rpart_fit, heart_test)
confusionMatrix(rpart_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 26  8
##          2  5 15
##
##                Accuracy : 0.7593
##                  95% CI : (0.6236, 0.8651)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 0.003636
##
##                   Kappa : 0.4993
##
##  Mcnemar's Test P-Value : 0.579100
##
##             Sensitivity : 0.8387
##                 Specificity : 0.6522
##              Pos Pred Value : 0.7647
##              Neg Pred Value : 0.7500
##                  Prevalence : 0.5741
##              Detection Rate : 0.4815
##        Detection Prevalence : 0.6296
##           Balanced Accuracy : 0.7454
##
##            'Positive' Class : 1
##

library(rpart)

library(rpart.plot)
rpart.plot(rpart_fit$finalModel,
           type=5,
           fallen.leaves = FALSE,
           box.palette = "GnRd",
           nn=TRUE)
```



##KNN

```
ctrl <- trainControl(method = "cv", verboseIter = FALSE, number = 5)
set.seed(503)
knn_fit <- train(as.factor(heart.disease) ~ .,
                data = heart_train, method = "knn", preProcess = c("ce
nter","scale"),
                trControl = ctrl , tuneGrid = expand.grid(k = seq(1, 2
0, 2)))
```

```
knn_fit
```

```
## k-Nearest Neighbors
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results across tuning parameters:
##
##    k   Accuracy   Kappa
##     1  0.7821856  0.5599854
##     3  0.7961492  0.5855293
##     5  0.8197272  0.6338094
##     7  0.8425602  0.6810507
##     9  0.8564079  0.7082992
##    11  0.8379140  0.6700623
##    13  0.8425652  0.6791184
##    15  0.8423487  0.6780794
##    17  0.8379090  0.6694955
##    19  0.8378033  0.6684487
##
## Accuracy was used to select the optimal model using the largest val
ue.
## The final value used for the model was k = 9.
```

```
knn_predict <- predict(knn_fit, heart_test)
confusionMatrix(knn_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  1  2
##          1 29  5
##          2  2 18
##
##                Accuracy : 0.8704
##                  95% CI : (0.751, 0.9463)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 2.608e-06
##
##                   Kappa : 0.7304
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9355
##             Specificity : 0.7826
##          Pos Pred Value : 0.8529
##          Neg Pred Value : 0.9000
##              Prevalence : 0.5741
##          Detection Rate : 0.5370
##    Detection Prevalence : 0.6296
##       Balanced Accuracy : 0.8590
##
##        'Positive' Class : 1
##
```

##Support Vector Machine (SVMLinear)

```
ctrl <- trainControl(method = "cv", verboseIter = FALSE, number = 5)
set.seed(503)

grid_svm <- expand.grid(C = c(0.01, 0.1, 1, 10, 20))


svm_fit <- train(as.factor(heart.disease) ~ .,data = heart_train,
                 method = "svmLinear", preProcess = c("center","scale"
),
                 tuneGrid = grid_svm, trControl = ctrl)
svm_fit
```

```
## Support Vector Machines with Linear Kernel
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results across tuning parameters:
##
##   C      Accuracy   Kappa
##    0.01  0.8422430  0.6750859
##    0.10  0.8375919  0.6718875
##    1.00  0.8144317  0.6252399
##   10.00  0.8145424  0.6264769
##   20.00  0.8145424  0.6264769
## 
## Accuracy was used to select the optimal model using the largest val
ue.
## The final value used for the model was C = 0.01.
```

```
svm_predict <- predict(svm_fit, heart_test)
confusionMatrix(svm_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  1  2
##          1 28  5
##          2  3 18
##                                           
##                Accuracy : 0.8519          
##                  95% CI : (0.7288, 0.9338)
##     No Information Rate : 0.5741          
##     P-Value [Acc > NIR] : 1.182e-05       
##                                           
##                   Kappa : 0.6936          
##                                           
##  Mcnemar's Test P-Value : 0.7237          
##                                           
##             Sensitivity : 0.9032          
##             Specificity : 0.7826          
##          Pos Pred Value : 0.8485          
##          Neg Pred Value : 0.8571          
##              Prevalence : 0.5741          
##          Detection Rate : 0.5185          
##    Detection Prevalence : 0.6111          
##       Balanced Accuracy : 0.8429          
##                                           
##        'Positive' Class : 1               
## 
```

Support Vector Machines with Radial kernel

```
sigmaEst <- kernlab::sigest(as.matrix(heart_train[,1:13]))
svmgrid <- expand.grid(sigma = sigmaEst, C = 2^seq(-4,+4))

set.seed(503)
svmR_fit <- train(as.factor(heart.disease) ~ .,data = heart_train,
                 method = "svmRadial", preProcess = c("center","scale"
),
                 tuneGrid = svmgrid, trControl = ctrl)
svmR_fit
```

```
## Support Vector Machines with Radial Basis Function Kernel
## 
## 216 samples
##  13 predictor
##   2 classes: '1', '2' 
## 
## Pre-processing: centered (13), scaled (13) 
## Resampling: Cross-Validated (5 fold) 
## Summary of sample sizes: 172, 174, 173, 172, 173 
## Resampling results across tuning parameters:
## 
##   sigma       C        Accuracy   Kappa    
##   0.02506020   0.0625  0.7870533  0.5528577
##   0.02506020   0.1250  0.8328300  0.6557309
##   0.02506020   0.2500  0.8422430  0.6750859
##   0.02506020   0.5000  0.8513289  0.6966396
##   0.02506020   1.0000  0.8424595  0.6796107
##   0.02506020   2.0000  0.8285010  0.6518432
##   0.02506020   4.0000  0.8147589  0.6245500
##   0.02506020   8.0000  0.7963606  0.5899247
##   0.02506020  16.0000  0.7823971  0.5610870
##   0.04088519   0.0625  0.8099970  0.6044989
##   0.04088519   0.1250  0.8375919  0.6659338
##   0.04088519   0.2500  0.8374811  0.6657046
##   0.04088519   0.5000  0.8468942  0.6867949
##   0.04088519   1.0000  0.8238498  0.6420363
##   0.04088519   2.0000  0.8194100  0.6344370
##   0.04088519   4.0000  0.8102084  0.6170805
##   0.04088519   8.0000  0.7732004  0.5414910
##   0.04088519  16.0000  0.7638931  0.5228505
##   0.07053588   0.0625  0.7688765  0.5131599
##   0.07053588   0.1250  0.8281788  0.6466808
##   0.07053588   0.2500  0.8329357  0.6567316
##   0.07053588   0.5000  0.8425652  0.6789357
##   0.07053588   1.0000  0.8333686  0.6616707
##   0.07053588   2.0000  0.8056629  0.6083322
##   0.07053588   4.0000  0.7733112  0.5415856
##   0.07053588   8.0000  0.7775345  0.5495941
##   0.07053588  16.0000  0.7822964  0.5587234
## 
## Accuracy was used to select the optimal model using the largest val
ue.
```

```
## The final values used for the model were sigma = 0.0250602 and C =
0.5.
```

```
svmR_predict <- predict(svmR_fit, heart_test)
confusionMatrix(svmR_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 29  5
##          2  2 18
##
##                Accuracy : 0.8704
##                  95% CI : (0.751, 0.9463)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 2.608e-06
##
##                   Kappa : 0.7304
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9355
##             Specificity : 0.7826
##          Pos Pred Value : 0.8529
##          Neg Pred Value : 0.9000
##              Prevalence : 0.5741
##          Detection Rate : 0.5370
##    Detection Prevalence : 0.6296
##       Balanced Accuracy : 0.8590
##
##        'Positive' Class : 1
##
```

##Random Forest

```
control<- trainControl(method = "cv", number = 5, verboseIter = FALSE)
grid <-data.frame(mtry = seq(1, 10, 2))
set.seed(503)
rf_fit <- train(as.factor(heart.disease) ~ ., method = "rf", data = he
art_train, ntree = 20, trControl = control,
                  tuneGrid = grid)
rf_fit
```

```
## Random Forest
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
## 
## No pre-processing
## Resampling: Cross-Validated (5 fold) 
## Summary of sample sizes: 172, 174, 173, 172, 173 
## Resampling results across tuning parameters:
## 
##   mtry  Accuracy   Kappa    
##   1     0.8143310  0.6182560
##   3     0.8007953  0.5918303
##   5     0.8102134  0.6133067
##   7     0.8195158  0.6304878
##   9     0.7915937  0.5737053
## 
## Accuracy was used to select the optimal model using the largest val
ue.
## The final value used for the model was mtry = 7.
```

```
rf_predict <- predict(rf_fit, heart_test)
confusionMatrix(rf_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  1  2
##          1 26  7
##          2  5 16
##                                           
##                Accuracy : 0.7778          
##                  95% CI : (0.644, 0.8796)
##     No Information Rate : 0.5741          
##     P-Value [Acc > NIR] : 0.00143         
##                                           
##                   Kappa : 0.5404          
##                                           
##  Mcnemar's Test P-Value : 0.77283         
##                                           
##             Sensitivity : 0.8387          
##             Specificity : 0.6957          
##          Pos Pred Value : 0.7879          
##          Neg Pred Value : 0.7619          
##              Prevalence : 0.5741          
##          Detection Rate : 0.4815          
##    Detection Prevalence : 0.6111          
##       Balanced Accuracy : 0.7672          
```

```
##
##        'Positive' Class : 1
##
```

##QDA

```
set.seed(503)
qda_fit <- train(as.factor(heart.disease) ~ ., method = "qda", data =
heart_train)
qda_fit
```

```
## Quadratic Discriminant Analysis
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 216, 216, 216, 216, 216, 216, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7769824  0.5515716
```

```
qda_predict <- predict(qda_fit, heart_test)
confusionMatrix(qda_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 28  6
##          2  3 17
##
##                Accuracy : 0.8333
##                  95% CI : (0.7071, 0.9208)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 4.676e-05
##
##                   Kappa : 0.6534
##
##  Mcnemar's Test P-Value : 0.505
##
##             Sensitivity : 0.9032
##             Specificity : 0.7391
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.8500
##              Prevalence : 0.5741
##          Detection Rate : 0.5185
##    Detection Prevalence : 0.6296
##       Balanced Accuracy : 0.8212
##
##        'Positive' Class : 1
##
```

##Gradient Boosting Machine

```
gbmGrid <-  expand.grid(interaction.depth = c(1, 5, 10, 25, 30),
                        n.trees = c(5, 10, 25, 50),
                        shrinkage = c(0.1, 0.2, 0.3,  0.4, 0.5),
                        n.minobsinnode = 20)

set.seed(503)
gbm_fit <- train(as.factor(heart.disease) ~ ., method = "gbm", data =
heart_train,  trControl = control, verbose = FALSE, tuneGrid = gbmGrid
)
gbm_fit
```

```
## Stochastic Gradient Boosting
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results across tuning parameters:
##
##   shrinkage  interaction.depth  n.trees  Accuracy   Kappa
##   0.1         1                  5      0.7776402  0.5365405
##   0.1         1                 10      0.8193043  0.6298612
##   0.1         1                 25      0.8332629  0.6596724
##   0.1         1                 50      0.8467834  0.6873135
##   0.1         5                  5      0.7872747  0.5598093
##   0.1         5                 10      0.8148646  0.6199736
##   0.1         5                 25      0.8332578  0.6595553
##   0.1         5                 50      0.8288231  0.6516591
##   0.1        10                  5      0.7591463  0.5004990
##   0.1        10                 10      0.8375868  0.6663273
##   0.1        10                 25      0.8374862  0.6692311
##   0.1        10                 50      0.8331521  0.6586575
##   0.1        25                  5      0.7775395  0.5393272
##   0.1         25                  10      0.8284959  0.6485442
##   0.1         25                  25      0.8468942  0.6876187
##   0.1         25                  50      0.8471056  0.6886368
##   0.1         30                   5      0.7918152  0.5684438
##   0.1         30                  10      0.8331471  0.6552971
##   0.1         30                  25      0.8466777  0.6874748
##   0.1         30                  50      0.8288181  0.6501902
##   0.2          1                   5      0.8190879  0.6307112
##   0.2          1                  10      0.8236333  0.6413314
##   0.2          1                  25      0.8472163  0.6883805
##   0.2          1                  50      0.8287124  0.6540508
##   0.2          5                   5      0.8372647  0.6676143
##   0.2          5                  10      0.8471056  0.6883803
##   0.2          5                  25      0.8474278  0.6890738
##   0.2          5                  50      0.8516511  0.6965949
##   0.2         10                   5      0.8239555  0.6394651
##   0.2         10                  10      0.8468892  0.6887835
##   0.2         10                  25      0.8424544  0.6782864
##   0.2         10                  50      0.8145374  0.6229499
##   0.2         25                   5      0.7963606  0.5849170
##   0.2         25                  10      0.8147589  0.6248387
##   0.2         25                  25      0.8378033  0.6682596
##   0.2         25                  50      0.8331521  0.6598252
##   0.2         30                   5      0.7917044  0.5740819
##   0.2         30                  10      0.8288181  0.6502626
##   0.2         30                  25      0.8330464  0.6585990
##   0.2         30                  50      0.8195158  0.6310048
##   0.3          1                   5      0.7917095  0.5757303
##   0.3          1                  10      0.8146532  0.6223108
##   0.3          1                  25      0.8426759  0.6823183
##   0.3          1                  50      0.8198429  0.6358059
##   0.3          5                   5      0.8098913  0.6132680
##   0.3          5                  10      0.8238498  0.6409152
##   0.3          5                  25      0.8331521  0.6590869
##   0.3          5                  50      0.8150861  0.6215947
##   0.3         10                   5      0.8009111  0.5934089
##   0.3         10                  10      0.8515504  0.6985367
##   0.3         10                  25      0.8471106  0.6902036
##   0.3         10                  50      0.8145374  0.6245767
##   0.3         25                   5      0.8190929  0.6322239
##   0.3         25                  10      0.8375919  0.6703315
##   0.3         25                  25      0.8515454  0.6976958
##   0.3         25                  50      0.8151868  0.6244794
##   0.3         30                   5      0.8379090  0.6684201
##   0.3         30                  10      0.8193043  0.6302632
##   0.3         30                  25      0.8239656  0.6418297
##   0.3        30                 50       0.7871690  0.5636472
##   0.4         1                  5       0.7966828  0.5917100
##   0.4         1                 10       0.8334692  0.6630059
##   0.4         1                 25       0.8334743  0.6637227
##   0.4         1                 50       0.8146532  0.6227456
##   0.4         5                  5       0.8189872  0.6324495
##   0.4         5                 10       0.8194100  0.6337379
##   0.4         5                 25       0.8014497  0.5991760
##   0.4         5                 50       0.7965821  0.5887921
##   0.4        10                  5       0.8285010  0.6488496
##   0.4        10                 10       0.8516561  0.6964464
##   0.4        10                 25       0.7961542  0.5856654
##   0.4        10                 50       0.7780731  0.5489088
##   0.4        25                  5       0.8148646  0.6228855
##   0.4        25                 10       0.8097856  0.6113681
##   0.4        25                 25       0.7864089  0.5682204
##   0.4        25                 50       0.7918152  0.5769888
##   0.4        30                  5       0.8240612  0.6424101
##   0.4        30                 10       0.8332629  0.6628194
##   0.4        30                 25       0.8197372  0.6330701
##   0.4        30                 50       0.8237491  0.6414819
##   0.5         1                  5       0.8658210  0.7249885
##   0.5         1                 10       0.8379140  0.6705967
##   0.5         1                 25       0.8331521  0.6612942
##   0.5         1                 50       0.7913873  0.5802623
##   0.5         5                  5       0.8422430  0.6797641
##   0.5         5                 10       0.8331521  0.6609814
##   0.5         5                 25       0.7917095  0.5783567
##   0.5         5                 50       0.7961593  0.5830955
##   0.5        10                  5       0.8242726  0.6451090
##   0.5        10                 10       0.8241720  0.6435216
##   0.5        10                 25       0.8286117  0.6509143
##   0.5        10                 50       0.8096748  0.6134107
##   0.5        25                  5       0.8005889  0.5941704
##   0.5        25                 10       0.8149703  0.6257908
##   0.5        25                 25       0.7965771  0.5882871
##   0.5        25                 50       0.7871640  0.5682824
##   0.5        30                  5       0.8057787  0.6071936
##   0.5        30                 10       0.8282845  0.6495855
##   0.5        30                 25       0.8101027  0.6149082
##   0.5        30                 50       0.7959327  0.5859065
## 
## Tuning parameter 'n.minobsinnode' was held constant at a value of 2
0
## Accuracy was used to select the optimal model using the largest val
ue.
```

```
## The final values used for the model were n.trees = 5, interaction.d
epth =
##  1, shrinkage = 0.5 and n.minobsinnode = 20.
```

```
gbm_predict <- predict(gbm_fit, heart_test)
confusionMatrix(gbm_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  1  2
##          1 22  6
##          2  9 17
##                                           
##                Accuracy : 0.7222          
##                  95% CI : (0.5836, 0.8354)
##     No Information Rate : 0.5741          
##     P-Value [Acc > NIR] : 0.01791         
##                                           
##                   Kappa : 0.4414          
##                                           
##  Mcnemar's Test P-Value : 0.60558         
##                                           
##             Sensitivity : 0.7097          
##             Specificity : 0.7391          
##          Pos Pred Value : 0.7857          
##          Neg Pred Value : 0.6538          
##              Prevalence : 0.5741          
##          Detection Rate : 0.4074          
##    Detection Prevalence : 0.5185          
##       Balanced Accuracy : 0.7244          
##                                           
##        'Positive' Class : 1               
## 
```

##Bagged trees

```
set.seed(503)
bagged_fit <- train(as.factor(heart.disease) ~ ., method = "treebag",
nbagg=50,
                    data = heart_train,  trControl = control, metric="
Accuracy")

bagged_fit
```

```
## Bagged CART 
## 
```

```
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results:
##
##   Accuracy   Kappa
##   0.8144317  0.6221302
```

```
bagged_predict <- predict(bagged_fit, heart_test)
confusionMatrix(bagged_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 27  5
##          2  4 18
##
##                Accuracy : 0.8333
##                  95% CI : (0.7071, 0.9208)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 4.676e-05
##
##                   Kappa : 0.6573
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8710
##             Specificity : 0.7826
##          Pos Pred Value : 0.8438
##          Neg Pred Value : 0.8182
##              Prevalence : 0.5741
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5926
##       Balanced Accuracy : 0.8268
##
##        'Positive' Class : 1
##
```

##Neural network

```
set.seed(503)
```

```
nnetGrid <- expand.grid(size=1:3, decay=c(0,0.1,0.2,0.3,0.4,0.5,1,2))

nnet_fit <- train(as.factor(heart.disease) ~ ., method = "nnet",
                    data = heart_train, tuneGrid=nnetGrid,
                  trace=FALSE, maxit=2000, trControl = control, metric
="Accuracy")

nnet_fit

## Neural Network
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results across tuning parameters:
##
##   size  decay  Accuracy   Kappa
##   1     0.0    0.6665459  0.2672235
##   1     0.1    0.8329357  0.6607511
##   1     0.2    0.8423437  0.6797597
##   1     0.3    0.8376925  0.6706021
##   1     0.4    0.8375868  0.6715644
##   1     0.5    0.8375868  0.6715644
##   1     1.0    0.8330464  0.6611183
##   1     2.0    0.8286117  0.6504003
##   2     0.0    0.6009614  0.1443406
##   2     0.1    0.8010168  0.5972284
##   2     0.2    0.8142152  0.6246645
##   2     0.3    0.8282845  0.6537589
##   2     0.4    0.8004782  0.5928086
##   2     0.5    0.8376976  0.6714324
##   2     1.0    0.8332629  0.6602823
##   2     2.0    0.8285010  0.6509686
##   3     0.0    0.6541730  0.2659360
##   3     0.1    0.8055572  0.6076564
##   3     0.2    0.7911708  0.5740845
##   3     0.3    0.8241720  0.6453645
##   3     0.4    0.8238498  0.6431288
##   3     0.5    0.8333686  0.6616189
##   3     1.0    0.8331521  0.6593204
##   3     2.0    0.8331521  0.6588559
##
```

```
## Accuracy was used to select the optimal model using the largest val
ue.
## The final values used for the model were size = 1 and decay = 0.2.
```

```
nnet_predict <- predict(nnet_fit, heart_test)
confusionMatrix(nnet_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  1  2
##          1 26  5
##          2  5 18
##                                           
##                Accuracy : 0.8148          
##                  95% CI : (0.6857, 0.9075)
##     No Information Rate : 0.5741          
##     P-Value [Acc > NIR] : 0.0001634       
##                                           
##                   Kappa : 0.6213          
##                                           
##  Mcnemar's Test P-Value : 1.0000000       
##                                           
##             Sensitivity : 0.8387          
##             Specificity : 0.7826          
##          Pos Pred Value : 0.8387          
##          Neg Pred Value : 0.7826          
##              Prevalence : 0.5741          
##          Detection Rate : 0.4815          
##    Detection Prevalence : 0.5741          
##       Balanced Accuracy : 0.8107          
##                                           
##        'Positive' Class : 1               
## 
```

Support vector machine with svmlinear kernel has the best performance among all.

```
nscGrid <- data.frame(threshold = seq(0,25, length=30))
set.seed(503)

nsc_fit <- train(as.factor(heart.disease) ~ .,method = "pam",data = he
art_train,
                 preProc = c("center", "scale"), tuneGrid = nscGrid,tr
Control = ctrl,metric = "Accuracy")
```

```
## 111111
```

```
nsc_fit

## Nearest Shrunken Centroids
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results across tuning parameters:
##
##   threshold  Accuracy   Kappa
##    0.000000  0.8421323  0.6752230
##    0.862069  0.8422380  0.6752968
##    1.724138  0.8146481  0.6165775
##    2.586207  0.7270059  0.4192687
##    3.448276  0.5509614  0.0000000
##    4.310345  0.5509614  0.0000000
##    5.172414  0.5509614  0.0000000
##    6.034483  0.5509614  0.0000000
##    6.896552  0.5509614  0.0000000
##    7.758621  0.5509614  0.0000000
##    8.620690  0.5509614  0.0000000
##    9.482759  0.5509614  0.0000000
##   10.344828  0.5509614  0.0000000
##   11.206897  0.5509614  0.0000000
##   12.068966  0.5509614  0.0000000
##   12.931034  0.5509614  0.0000000
##   13.793103  0.5509614  0.0000000
##   14.655172  0.5509614  0.0000000
##   15.517241  0.5509614  0.0000000
##   16.379310  0.5509614  0.0000000
##   17.241379  0.5509614  0.0000000
##   18.103448  0.5509614  0.0000000
##   18.965517  0.5509614  0.0000000
##   19.827586  0.5509614  0.0000000
##   20.689655  0.5509614  0.0000000
##   21.551724  0.5509614  0.0000000
##   22.413793  0.5509614  0.0000000
##   23.275862  0.5509614  0.0000000
##   24.137931  0.5509614  0.0000000
##   25.000000  0.5509614  0.0000000
##
## Accuracy was used to select the optimal model using the largest val
```

```
ue.
## The final value used for the model was threshold = 0.862069.
```

```
nsc_predict <- predict(nsc_fit, heart_test)
confusionMatrix(nsc_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  1  2
##          1 28  6
##          2  3 17
##
##                Accuracy : 0.8333
##                  95% CI : (0.7071, 0.9208)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 4.676e-05
##
##                   Kappa : 0.6534
##
##  Mcnemar's Test P-Value : 0.505
##
##             Sensitivity : 0.9032
##             Specificity : 0.7391
##          Pos Pred Value : 0.8235
##          Neg Pred Value : 0.8500
##              Prevalence : 0.5741
##          Detection Rate : 0.5185
##    Detection Prevalence : 0.6296
##       Balanced Accuracy : 0.8212
##
##        'Positive' Class : 1
##
```

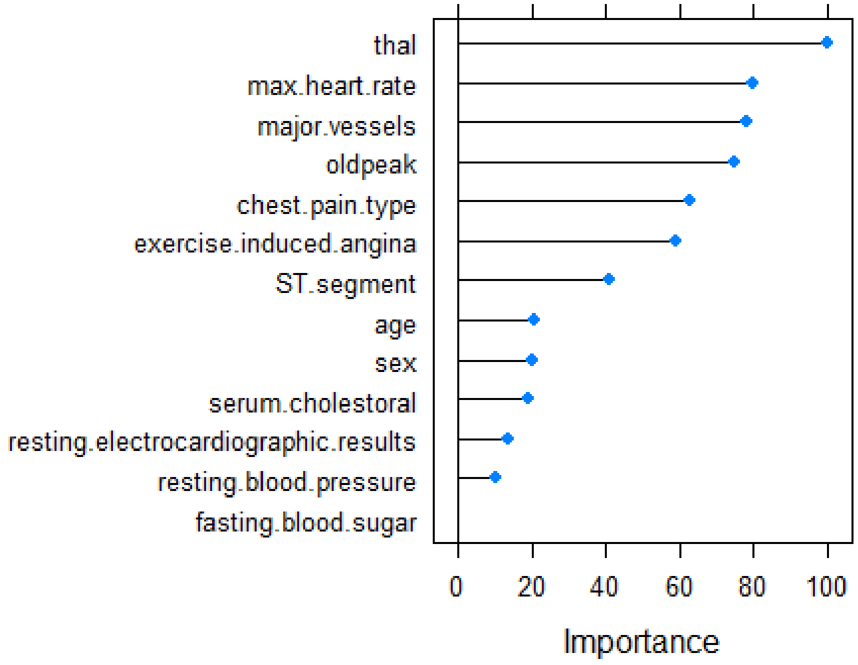
```
set.seed(503)
mda_fit <- train(x = heart_train[, 1:13],
                 y = as.factor(heart_train$heart.disease),
                 method = "mda",
                 tuneGrid = expand.grid(subclasses = 1:3),
                 metric = "Accuracy",
                 trControl = ctrl)
mda_fit
```

```
## Mixture Discriminant Analysis
##
## 216 samples
##  13 predictor
##   2 classes: '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 172, 174, 173, 172, 173
## Resampling results across tuning parameters:
##
##   subclasses  Accuracy   Kappa
##   1           0.8422430  0.6791175
##   2           0.8141146  0.6223301
##   3           0.8093476  0.6127467
##
## Accuracy was used to select the optimal model using the largest val
ue.
## The final value used for the model was subclasses = 1.
```

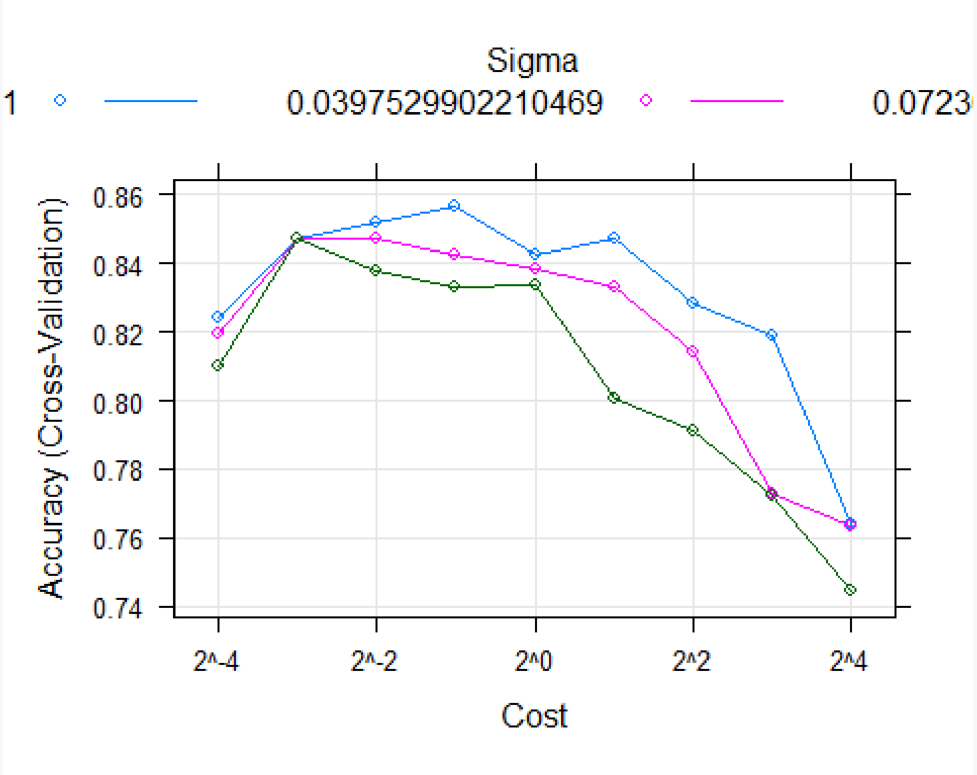
```
mda_predict <- predict(mda_fit, heart_test)
confusionMatrix(mda_predict, as.factor(heart_test$heart.disease))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 28  5
##          2  3 18
##
##                Accuracy : 0.8519
##                  95% CI : (0.7288, 0.9338)
##     No Information Rate : 0.5741
##     P-Value [Acc > NIR] : 1.182e-05
##
##                   Kappa : 0.6936
##
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.9032
##             Specificity : 0.7826
##          Pos Pred Value : 0.8485
##          Neg Pred Value : 0.8571
##              Prevalence : 0.5741
##          Detection Rate : 0.5185
##    Detection Prevalence : 0.6111
##       Balanced Accuracy : 0.8429
##
##        'Positive' Class : 1
##
```

```
#top important variables of SVM
varImp(svm_fit, top=5)

## loess r-squared variable importance
##
##                                      Overall
## thal                                  100.00
## max.heart.rate                         79.68
## major.vessels                          78.45
## oldpeak                                74.98
## chest.pain.type                        62.88
## exercise.induced.angina                58.92
## ST.segment                             41.13
## age                                    20.67
## sex                                    20.16
## serum.cholestoral                      19.19
## resting.electrocardiographic.results   13.39
## resting.blood.pressure                 10.03
## fasting.blood.sugar                     0.00

plot(varImp(svm_fit, top=5))
```



```r
plot(svmR_fit, scales = list(x = list(log = 2)))
```


Sigma
Accuracy (Cross-Validation)
2^-4
2^-2
2^0
2^2
2^4
Cost