Heart Disease Analysis

P.Srinith Reddy, M. Abhideep, P. Vishnu teja

Introduction

This analysis aims to explore the dataset related to heart disease and build predictive models to identify the presence of heart disease based on several features. We will start by loading the necessary libraries, then move on to data cleaning and exploratory data analysis, and finally, we will build predictive models using logistic regression, SVM, and kNN.

Install & load required packages

We begin by installing and loading the necessary libraries for our analysis. These libraries include packages for data manipulation, visualization, and machine learning.

```
install.packages(c("dplyr", "ggplot2", "forcats", "rsample", "tidyverse",
"tidymodels", "gridExtra",
"PROC", "tidyr", "readr", "caret", "gplots", "GGally", "dslabs", "lubridate",
"tidytext",
"RColorBrewer", "randomForest", "tictoc", "e1071", "ggpubr"))
```

```
library(dslabs)
library(lubridate)
library(tidytext)
library("RColorBrewer")
library(randomForest)
library(tictoc)
library(e1071)
library(ggpubr)
library(dplyr)
library(ggplot2)
library(forcats)
library(rsample)
library(tidyverse)
library(gridExtra)
library(pROC)
library(tidyr)
library(readr)
library(caret)
library(gplots)
library(GGally)
```

Load Data

We load the heart disease dataset and perform an initial inspection to understand its structure and content. This step involves reading the dataset, displaying the first few rows, and summarizing the data.

```
heart <- read.csv("C:/Users/Abhideep/Movies/project/heart.csv")
```

Understanding Dataset

We need to understand the dataset using head, summary and str to learn about no.of columns, no.of rows and do the required cleaning and transformations.

```
head(heart)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  52   1  0      125  212   0       1     168     0     1.0     2  2    3
## 2  53   1  0      140  203   1       0     155     1     3.1     0  0    3
## 3  70   1  0      145  174   0       1     125     1     2.6     0  0    3
## 4  61   1  0      148  203   0       1     161     0     0.0     2  1    3
## 5  62   0  0      138  294   1       1     106     0     1.9     1  3    2
## 6  58   0  0      100  248   0       0     122     0     1.0     1  0    2
##   heart_disease
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             1
```

```
summary(heart)
```

```
##       age             sex               cp            trestbps    
##  Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.   : 94.0  
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.0  
##  Median :56.00   Median :1.0000   Median :1.0000   Median :130.0  
##  Mean   :54.43   Mean   :0.6956   Mean   :0.9424   Mean   :131.6  
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.0000   3rd Qu.:140.0  
##  Max.   :77.00   Max.   :1.0000   Max.   :3.0000   Max.   :200.0  
##       chol           fbs            restecg          thalach     
##  Min.   :126   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0  
##  1st Qu.:211   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:132.0  
##  Median :240   Median :0.0000   Median :1.0000   Median :152.0  
##  Mean   :246   Mean   :0.1493   Mean   :0.5298   Mean   :149.1  
##  3rd Qu.:275   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0  
##  Max.   :564   Max.   :1.0000   Max.   :2.0000   Max.   :202.0  
##      exang           oldpeak          slope             ca        
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.800   Median :1.000   Median :0.0000  
##  Mean   :0.3366   Mean   :1.072   Mean   :1.385   Mean   :0.7541  
##  3rd Qu.:1.0000   3rd Qu.:1.800   3rd Qu.:2.000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :6.200   Max.   :2.000   Max.   :4.0000  
##       thal       heart_disease   
##  Min.   :0.000   Min.   :0.0000  
##  1st Qu.:2.000   1st Qu.:0.0000  
##  Median :2.000   Median :1.0000  
##  Mean   :2.324   Mean   :0.5132  
##  3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :3.000   Max.   :1.0000
```

```
str(heart)
```

```
## '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 ...
##  $ heart_disease: int  0 0 0 0 0 1 0 0 0 0 ...
```

```
names = c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg",
"thalach", "exang", "oldpeak",
          "slope", "ca", "thal", "heart_disease")
colnames(heart) <- names
```

Data Cleaning and Transformation

In this section, we clean and transform the dataset. We rename columns for better readability, recode categorical variables, convert them into factors, and handle any missing values. Finally, we select the relevant columns for our analysis.

(i) Encoding of numerical values to string values:

```
heart <- heart %>%
  mutate(sex = case_when(sex == 0 ~ "female",
                         sex == 1 ~ "male")) %>%
  mutate(cp = case_when(cp == 1 ~ "typical angina",
                        cp == 2 ~ "atypical angina",
                        cp == 3 ~ "non-anginal pain",
                        cp == 4 ~ "asymptomatic")) %>%
  mutate(fbs = case_when(fbs == 1 ~ "high",
                         fbs == 0 ~ "low")) %>%
  mutate(exang = case_when(exang == 0 ~ "no",
                           exang == 1 ~ "yes")) %>%
  mutate(heart_disease = case_when(heart_disease == 0 ~ "absence",
                                   TRUE ~ "presence"))
```

(ii) Transforming specific columns into Factors:

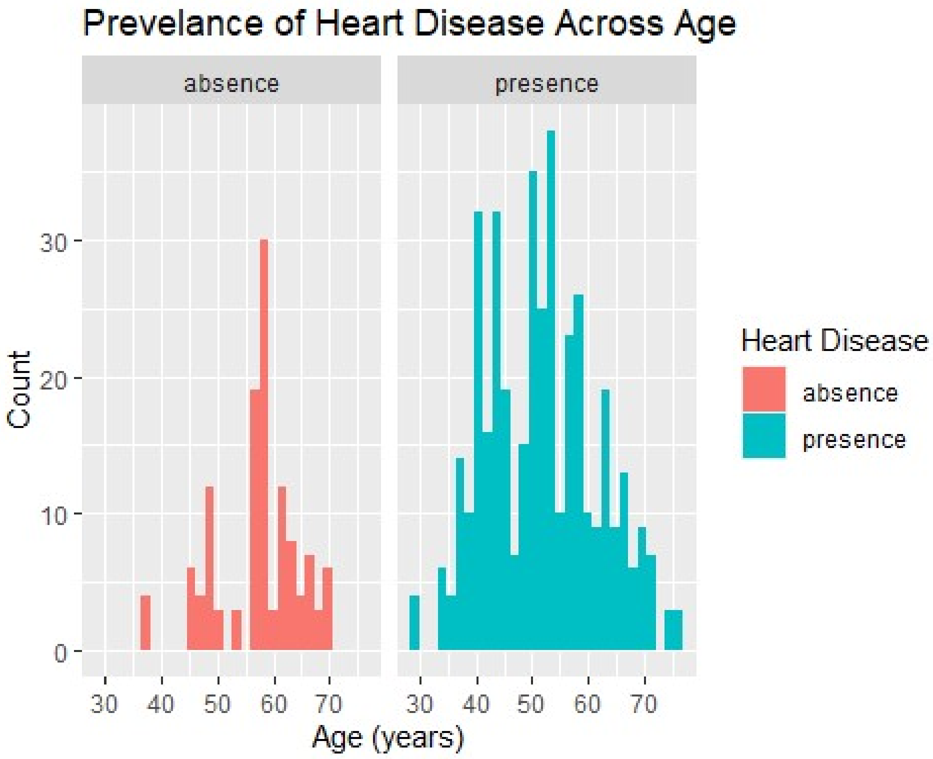
```
heart <- heart %>%
  mutate(sex = as.factor(sex)) %>%
  mutate(cp = as.factor(cp)) %>%
  mutate(fbs = as.factor(fbs)) %>%
  mutate(exang = as.factor(exang)) %>%
  mutate(heart_disease = as.factor(heart_disease))
```

(iii) Renaming specific columns:

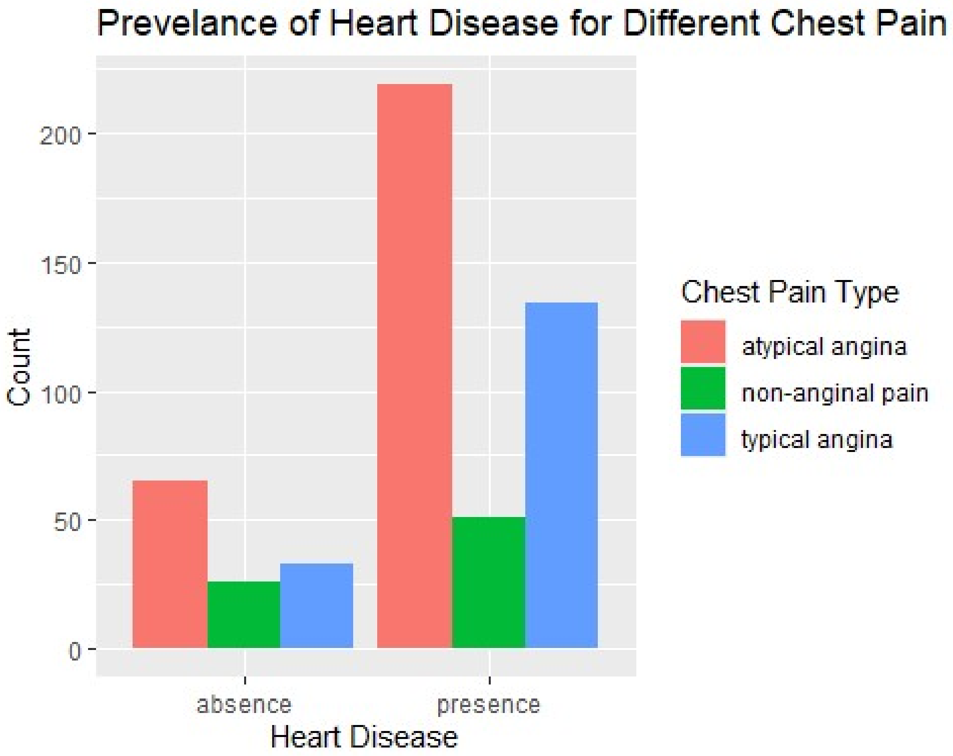
```
heart <- heart %>%
  select(age, sex, cp, trestbps, chol, fbs, thalach, exang, heart_disease)
%>%
  rename("max_hr" = "thalach",
         "exercise_angina" = "exang") %>%
  drop_na()
```

Exploratory Data Analysis

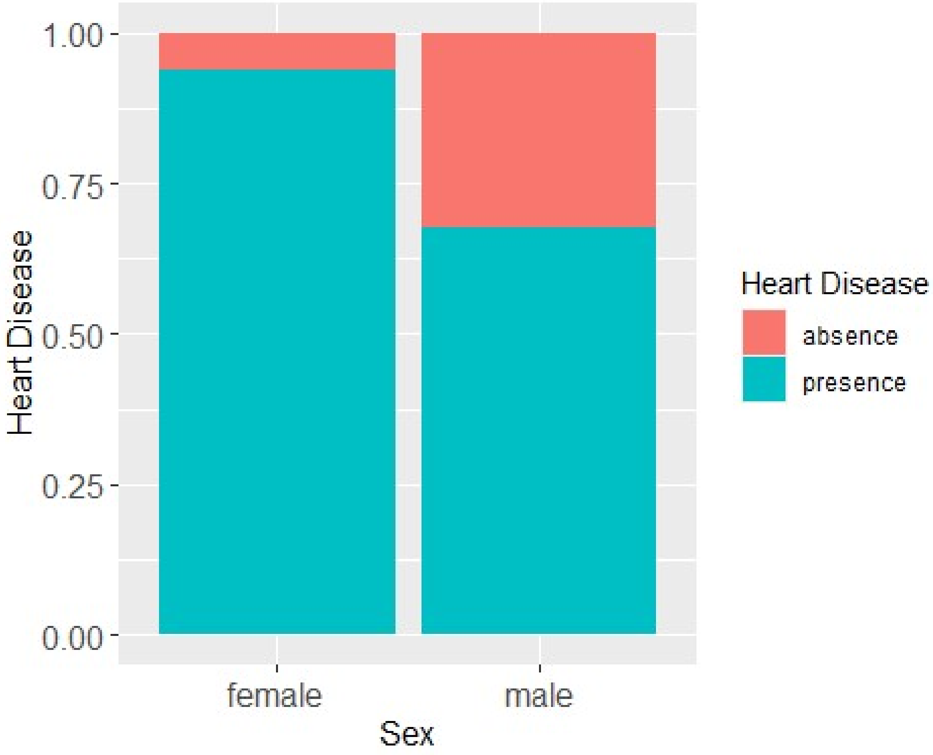
(i)We visualize the age distribution of patients with and without heart disease to understand the prevalence of heart disease across different age groups.



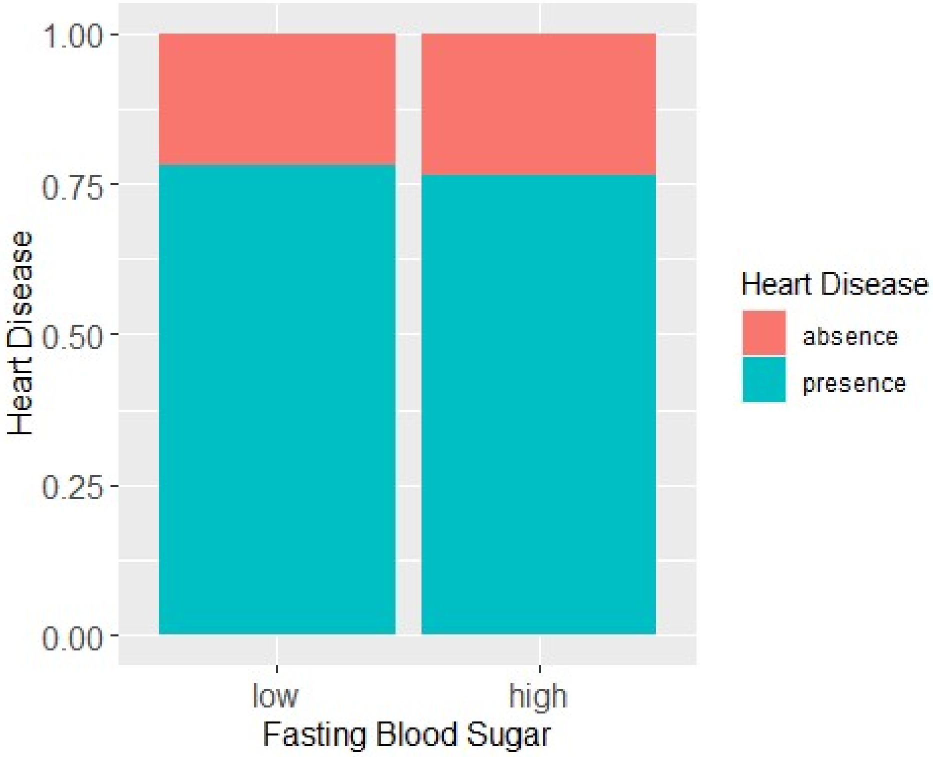
(ii)We examine the distribution of different types of chest pain among patients with and without heart disease to identify any patterns.



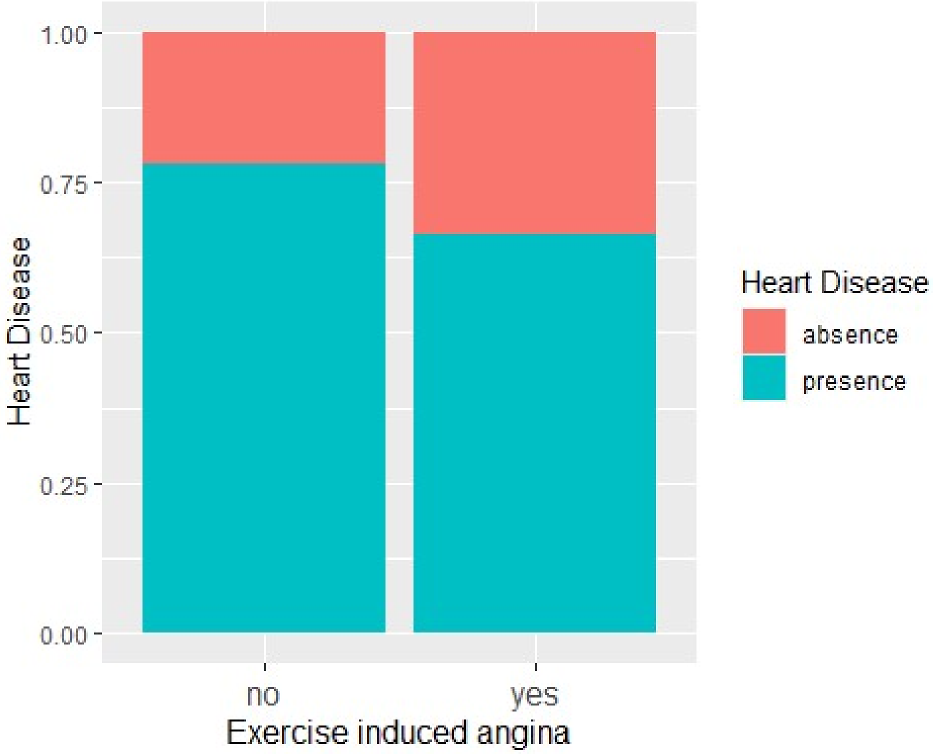
(iii)We compare the prevalence of heart disease between males and females to see if there is a significant difference based on sex.



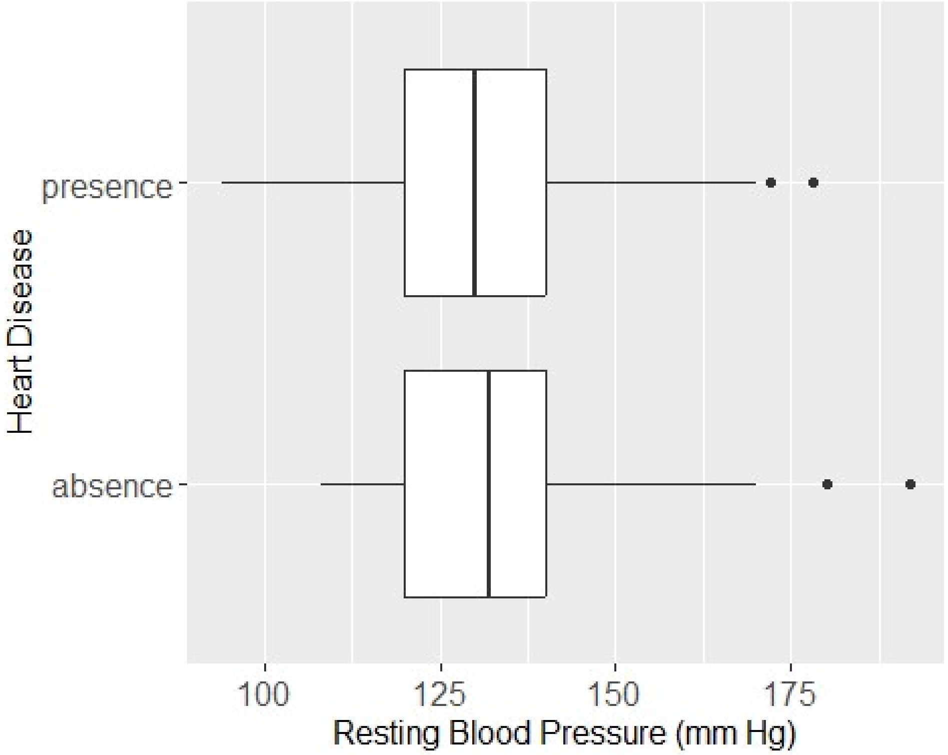
(iv)We analyze the relationship between fasting blood sugar levels and the presence of heart disease.



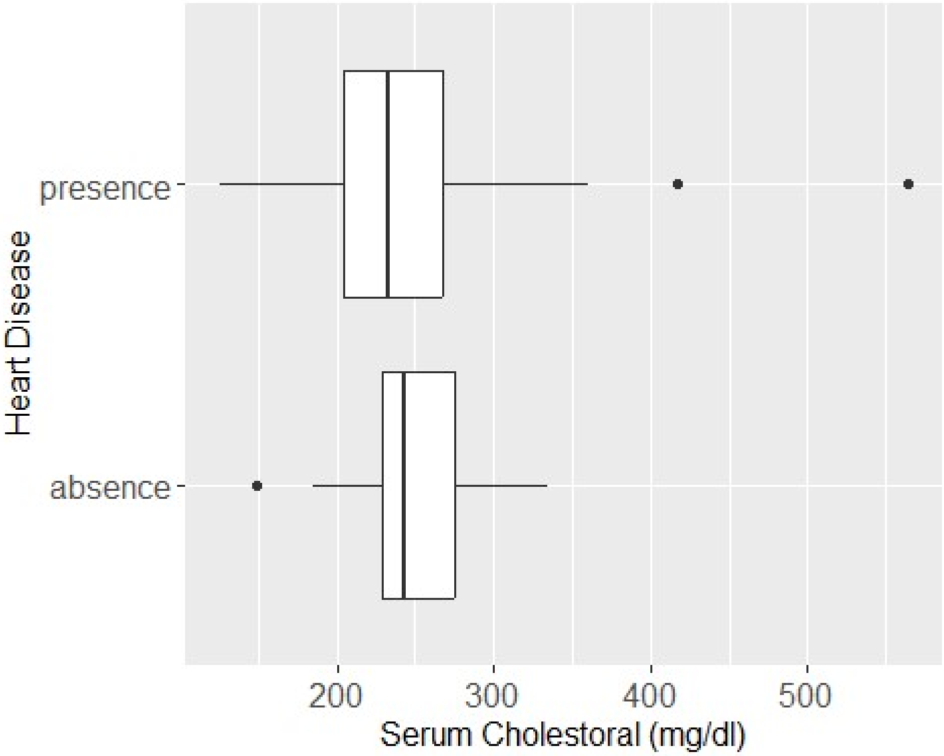
(v)We explore the impact of exercise-induced angina on heart disease prevalence.



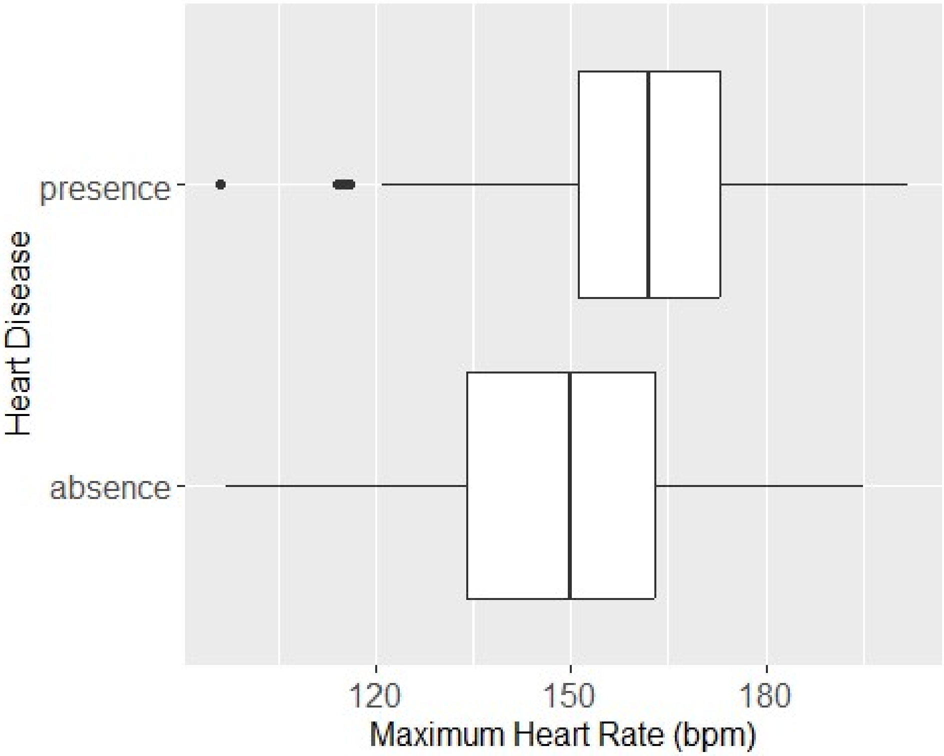
(vi)We visualize the relationship between resting blood pressure and heart disease.



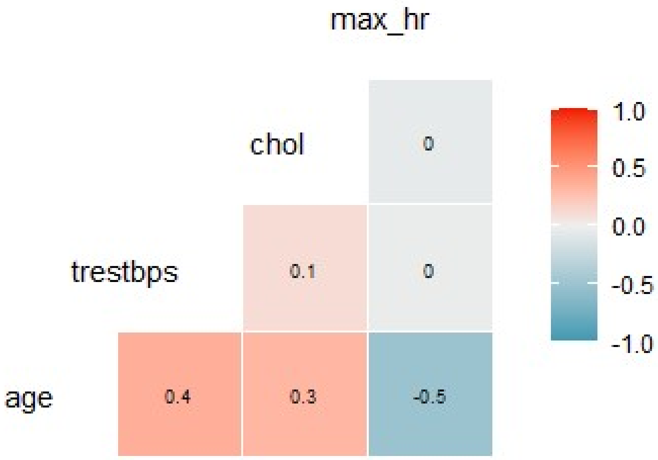
(vii)We examine the distribution of serum cholesterol levels among patients with and without heart disease.



(viii)We analyze the relationship between maximum heart rate and heart disease.



We create acorrelation matrix to understand the relationships between different variables in the dataset.



Predictive Modeling

We built a total of 3 models Logistic Regression model, SVM Model & KNN Model. *(i) Train Test Split:*

```
heart$heart_disease <- as.factor(heart$heart_disease)

set.seed(123)
splitIndex <- createDataPartition(heart$heart_disease, p = 0.8, list = FALSE)
train_set <- heart[splitIndex, ]
test_set <- heart[-splitIndex, ]
```

(ii) Model-1 Building:

```
logistic_fit <- train(heart_disease ~ ., data = train_set, method = "glm",
family = binomial)
print(logistic_fit)
```

```
## Generalized Linear Model
##
## 424 samples
##   8 predictor
##   2 classes: 'absence', 'presence'
##
```

```
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 424, 424, 424, 424, 424, 424, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.7770756  0.278278

log_predictions <- predict(logistic_fit, newdata = test_set)
```

(iii) Model-2 Building:

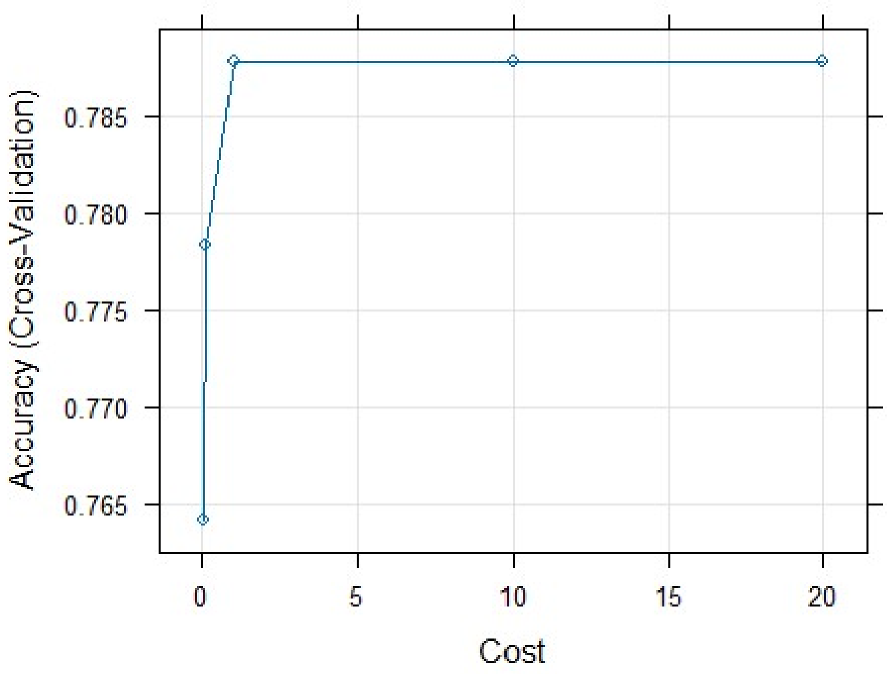
```
ctrl <- trainControl(method = "cv", verboseIter = FALSE, number = 5)

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

svm_fit <- train(heart_disease ~ .,data = train_set,
                 method = "svmLinear", preProcess = c("center","scale"),
                 tuneGrid = grid_svm, trControl = ctrl)
print(svm_fit)

## Support Vector Machines with Linear Kernel
##
## 424 samples
##   8 predictor
##   2 classes: 'absence', 'presence'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 339, 340, 339, 339, 339
## Resampling results across tuning parameters:
##
##   C      Accuracy   Kappa
##    0.01  0.7641457  0.0000000
##    0.10  0.7783193  0.1449890
##    1.00  0.7877591  0.2584493
##   10.00  0.7877591  0.2584493
##   20.00  0.7877591  0.2584493
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.

plot(svm_fit)
```



```
svm_predict <- predict(svm_fit, newdata = test_set)
```

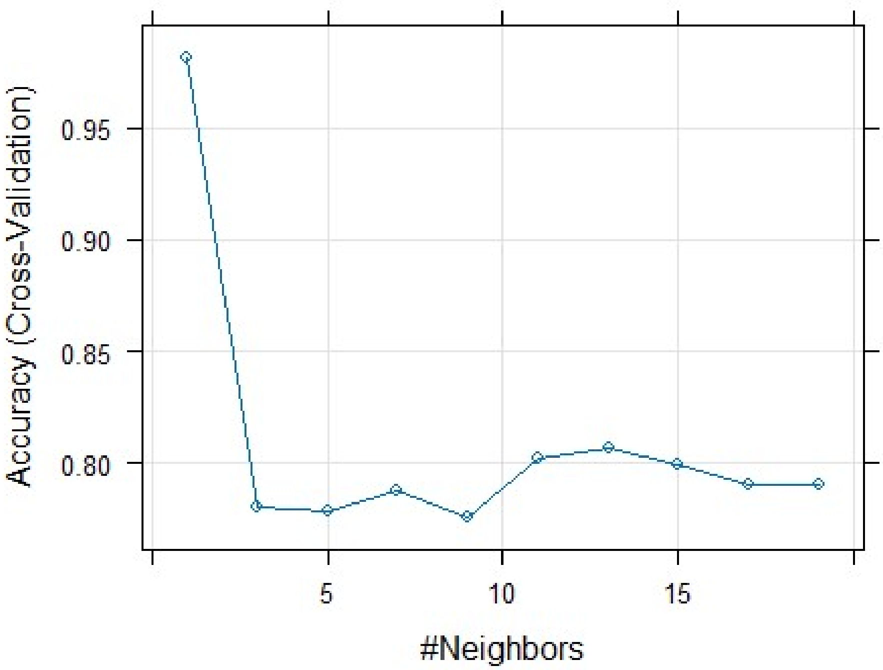
(iv) Model-3 Building:

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

knnFit <- train(heart_disease ~ .,
                data = train_set, method = "knn", preProcess =
c("center","scale"),
                trControl = ctrl , tuneGrid = expand.grid(k = seq(1, 20, 2)))
print(knnFit)
```

```
## k-Nearest Neighbors
##
## 424 samples
##   8 predictor
##   2 classes: 'absence', 'presence'
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 339, 339, 339, 339, 340
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9811765  0.9474389
##    3  0.7805322  0.4110455
##    5  0.7782073  0.3638559
##    7  0.7877031  0.3361050
##    9  0.7759384  0.2743740
##   11  0.8019608  0.3776836
##   13  0.8066106  0.3808000
##   15  0.7995238  0.3804404
##   17  0.7901120  0.3372687
##   19  0.7901120  0.2964636
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
plot(knnFit)
```



```
knn_predictions <- predict(knnFit,newdata = test_set )
```

Model- 1,2,3 Evaluation

Confusion matrices:

```
logistic_cm <- confusionMatrix(log_predictions, test_set$heart_disease)
print(logistic_cm)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction absence presence
##   absence        9        2
##   presence      15       78
```

```
##
##                 Accuracy : 0.8365
##                   95% CI : (0.7512, 0.9018)
##      No Information Rate : 0.7692
##      P-Value [Acc > NIR] : 0.061160
##
##                    Kappa : 0.4319
##
##   Mcnemar's Test P-Value : 0.003609
##
##              Sensitivity : 0.37500
##              Specificity : 0.97500
##           Pos Pred Value : 0.81818
##           Neg Pred Value : 0.83871
##               Prevalence : 0.23077
##           Detection Rate : 0.08654
##     Detection Prevalence : 0.10577
##        Balanced Accuracy : 0.67500
##
##         'Positive' Class : absence
##
```

```r
svm_cm <- confusionMatrix(svm_predict, test_set$heart_disease)
print(svm_cm)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction absence presence
##   absence        9        2
##   presence      15       78
##
##                 Accuracy : 0.8365
##                   95% CI : (0.7512, 0.9018)
##      No Information Rate : 0.7692
##      P-Value [Acc > NIR] : 0.061160
##
##                    Kappa : 0.4319
##
##   Mcnemar's Test P-Value : 0.003609
##
##              Sensitivity : 0.37500
##              Specificity : 0.97500
##           Pos Pred Value : 0.81818
##           Neg Pred Value : 0.83871
##               Prevalence : 0.23077
##           Detection Rate : 0.08654
##     Detection Prevalence : 0.10577
##        Balanced Accuracy : 0.67500
##
```

```
##         'Positive' Class : absence
##
```

```r
knn_cm <- confusionMatrix(knn_predictions, test_set$heart_disease )
print(knn_cm)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction absence presence
##   absence       24        0
##   presence       0       80
##
##                Accuracy : 1
##                  95% CI : (0.9652, 1)
##     No Information Rate : 0.7692
##     P-Value [Acc > NIR] : 1.412e-12
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.2308
##          Detection Rate : 0.2308
##    Detection Prevalence : 0.2308
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : absence
##
```

Accuracy:

```r
logistic_accuracy <- logistic_cm$overall["Accuracy"]
svm_accuracy <- svm_cm$overall["Accuracy"]
knn_accuracy <- knn_cm$overall["Accuracy"]

print(paste("Logistic Regression Accuracy: ", logistic_accuracy*100))
```

```
## [1] "Logistic Regression Accuracy:  83.6538461538462"
```

```r
print(paste("SVM Accuracy: ", svm_accuracy*100))
```

```
## [1] "SVM Accuracy:  83.6538461538462"
```

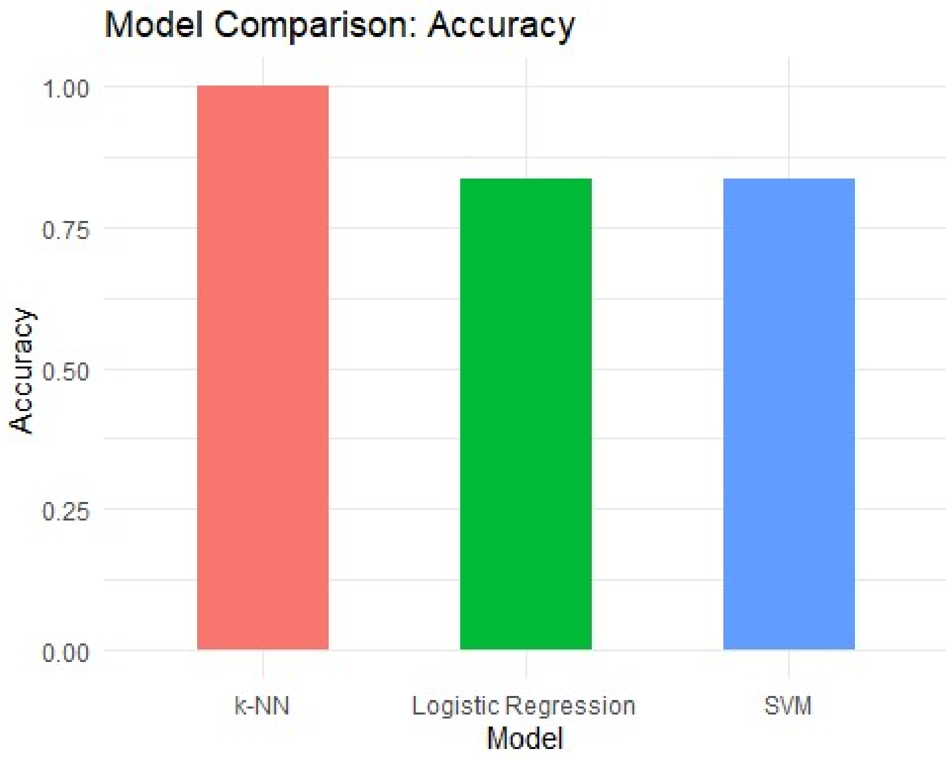
```r
print(paste("k-NN Accuracy: ", knn_accuracy*100))
```

```
## [1] "k-NN Accuracy:  100"
```

Plotting accuracy:

```r
results <- data.frame(
  Model = c("Logistic Regression", "SVM", "k-NN"),
  Accuracy = c(logistic_accuracy, svm_accuracy, knn_accuracy)
)

ggplot(results, aes(x = Model, y = Accuracy, fill = Model)) +
  geom_bar(stat = "identity", width = 0.5) +
  ylim(0, 1) +
  labs(title = "Model Comparison: Accuracy",
       x = "Model",
       y = "Accuracy") +
  theme_minimal() +
  theme(legend.position = "none")
```


Model Comparison: Accuracy
Accuracy
k-NN
Logistic Regression
SVM
Model