Heart Attack Prediction

Corey Land

2025-12-01

For this dataset we are looking at predicting whether a person is likely to have a heart attack or not based theses predictors

Age, Gender (male = 1, female = 0), cp (Constrictive pericarditis) same as chest pain, resting blood pressure, cholesterol, fasting blood sugar, resting electrocardiogram results max heart rate, exercise induced angina (1 = yes; 0 = no), Oldpeak: ST depression induced by exercise relative to rest Slope: the slope of the peak exercise ST segment number of major blood vessels, and AHD

We are performing logistic regression to predict heart attack possibilities for each patient: Yes = 1 and No = 0

Load Libraries

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.4.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.4.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.4.3
```

```
library(corrr)
```

```
## Warning: package 'corrr' was built under R version 4.4.3
```

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.4.3
```

```
##
## Attaching package: 'plotly'
```

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

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
## Attaching package: 'dplyr'
```

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

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.4.3
```

Load Data

```
df <- read.csv("Heart Attack Data Set.csv")
```

Exploring Data

```
# View first few rows of data
head(df)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  63   1  3      145  233   1       0     150     0     2.3     0  0    1
## 2  37   1  2      130  250   0       1     187     0     3.5     0  0    2
## 3  41   0  1      130  204   0       0     172     0     1.4     2  0    2
## 4  56   1  1      120  236   0       1     178     0     0.8     2  0    2
## 5  57   0  0      120  354   0       1     163     1     0.6     2  0    2
## 6  57   1  0      140  192   0       1     148     0     0.4     1  0    1
##   target
## 1      1
## 2      1
## 3      1
## 4      1
## 5      1
## 6      1
```

```
# Check contents of data
str(df)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
# all variables are numeric

# Check shape of data
dim(df)
```

```
## [1] 303  14
```

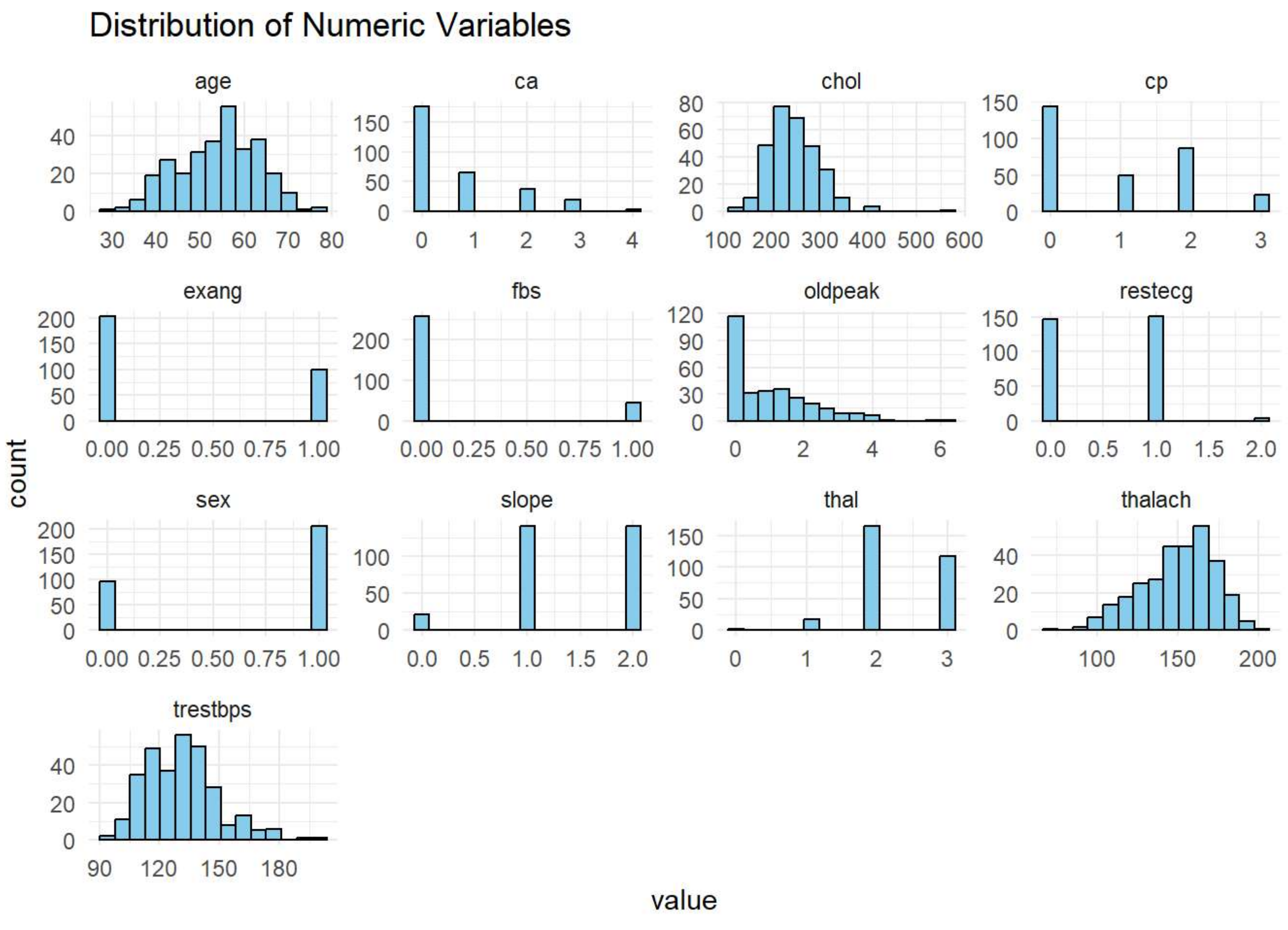
```
# dataset has 303 observations and 14 variables

# Check for missing values
colSums(is.na(df))
```

```
##      age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        0        0        0        0        0
##    exang  oldpeak    slope       ca     thal   target
##        0        0        0        0        0        0
```

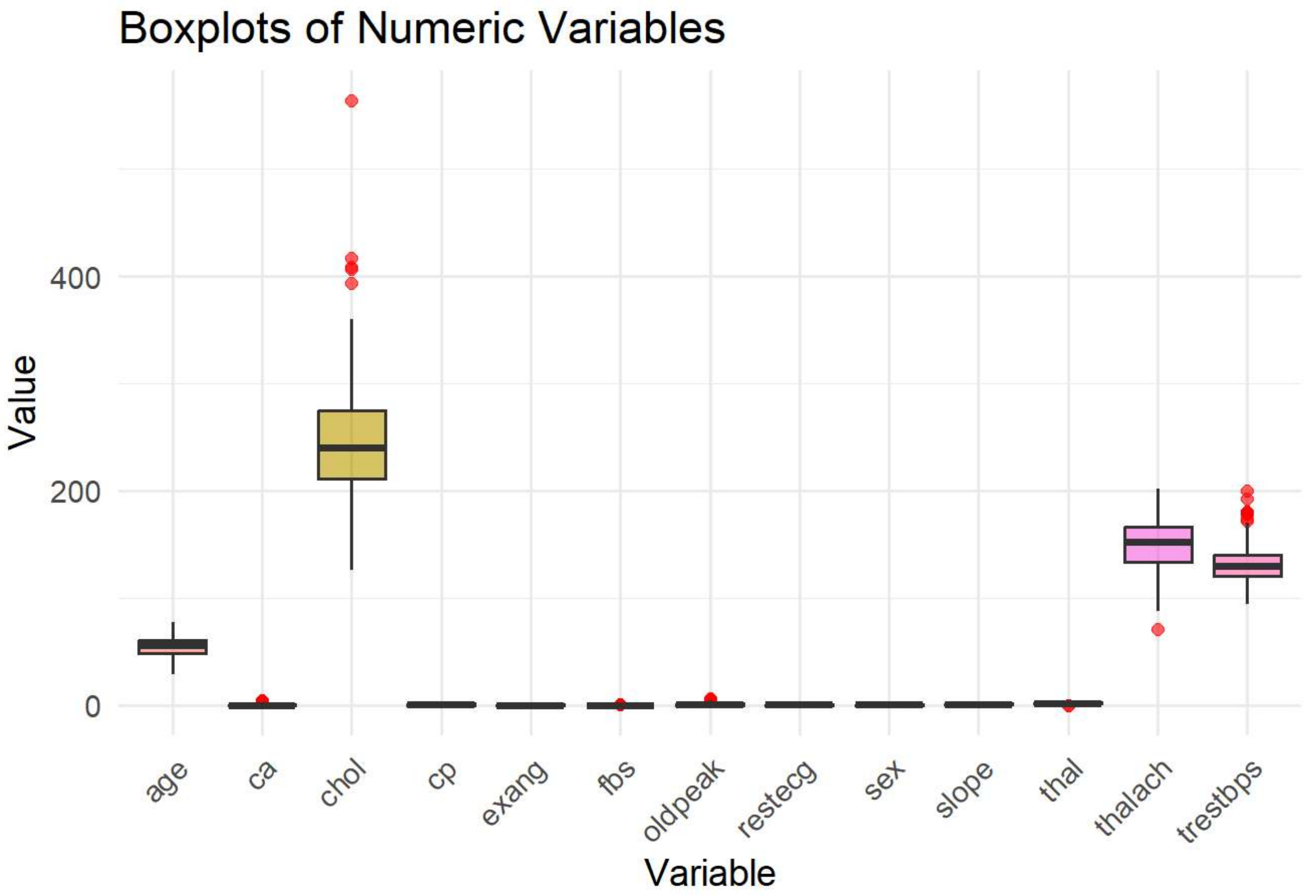
```
# data does not have any missing values

# remove any duplicates if there are any
df <- df[!duplicated(df), ]

# Dataset removed 1 duplicate, now there are 302 observations

# Check summary statistics
summary(df)
```

```
##       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 :55.50   Median :1.0000   Median :1.0000   Median :130.0
##  Mean   :54.42   Mean   :0.6821   Mean   :0.9636   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.   :126.0   Min.   :0.000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:133.2
##  Median :240.5   Median :0.000   Median :1.0000   Median :152.5
##  Mean   :246.5   Mean   :0.149   Mean   :0.5265   Mean   :149.6
##  3rd Qu.:274.8   3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.000   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   :0.3278   Mean   :1.043   Mean   :1.397   Mean   :0.7185
##  3rd Qu.:1.0000   3rd Qu.:1.600   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.200   Max.   :2.000   Max.   :4.0000
##       thal           target
##  Min.   :0.000   Min.   :0.000
##  1st Qu.:2.000   1st Qu.:0.000
##  Median :2.000   Median :1.000
##  Mean   :2.315   Mean   :0.543
##  3rd Qu.:3.000   3rd Qu.:1.000
##  Max.   :3.000   Max.   :1.000
```

```
# few examples

# average age is 54.42 years, with ages ranging from 29 to 77

# average trestbps is 130.0, with trestbps ranging from 94.0 to 200.0

# average cholesterol is 240.5 with cholesterol ranges from 126 to 564

# average thalach is 149.6 with values ranging from 71 to 202

# Let's look at our numeric variables
num_vars <- df %>% select(where(is.numeric)) %>%
  select(-target)

# Histograms for all numeric variables
if(ncol(num_vars) > 0){
  num_vars %>%
    pivot_longer(cols = everything(), names_to = "variable", values_to = "value") %>%
    ggplot(aes(x=value)) +
    geom_histogram(fill="skyblue", color="black", bins=15) +
    facet_wrap(~variable, scales="free") +
    theme_minimal() +
    labs(title="Distribution of Numeric Variables")
}
```



```
# Let's look at boxplots for numeric variables
num_vars %>%
  pivot_longer(cols = everything()) %>%
  ggplot(aes(x = name, y = value, fill = name)) +
  geom_boxplot(outlier.colour = "red", alpha = 0.6) +
  labs(title = "Boxplots of Numeric Variables", x = "Variable", y = "Value") +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), plot.margin = margin(10, 10, 20, 10))
+
  guides(fill = "none")
```



```
# There looks to be a good amount of outliers in the data

## Let's now look to see if there is correlation among variables
# View Correlation Matrix
cor_matrix <- cor(num_vars)
print(cor_matrix)
```

```
##                  age         sex          cp    trestbps         chol
## age       1.00000000 -0.09496249 -0.06310659  0.28312068  0.2072155057
## sex      -0.09496249  1.00000000 -0.05173952 -0.05764694 -0.1955714449
## cp       -0.06310659 -0.05173952  1.00000000  0.04648642 -0.0726816082
## trestbps  0.28312068 -0.05764694  0.04648642  1.00000000  0.1252562856
## chol      0.20721551 -0.19557144 -0.07268161  0.12525629  1.0000000000
## fbs       0.11949213  0.04602218  0.09601810  0.17812469  0.0114282965
## restecg  -0.11159006 -0.06035051  0.04156100 -0.11536656 -0.1476017717
## thalach  -0.39523519 -0.04643866  0.29336658 -0.04802271 -0.0053084958
## exang     0.09321585  0.14346012 -0.39293737  0.06852626  0.0640988995
## oldpeak   0.20603964  0.09832173 -0.14669225  0.19459995  0.0500860240
## slope    -0.16412423 -0.03298963  0.11685419 -0.12287293  0.0004166583
## ca        0.30226121  0.11306039 -0.19535634  0.09924834  0.0868779366
## thal      0.06531729  0.21145220 -0.16036963  0.06286958  0.0968104460
##                  fbs     restecg      thalach       exang      oldpeak
## age       0.119492128 -0.11159006 -0.395235188  0.09321585  0.206039638
## sex       0.046022181 -0.06035051 -0.046438663  0.14346012  0.098321733
## cp        0.096018104  0.04156100  0.293366582 -0.39293737 -0.146692247
## trestbps  0.178124692 -0.11536656 -0.048022712  0.06852626  0.194599950
## chol      0.011428297 -0.14760177 -0.005308496  0.06409890  0.050086024
## fbs       1.000000000 -0.08308108 -0.007169290  0.02472879  0.004514275
## restecg  -0.083081081  1.00000000  0.041209808 -0.06880655 -0.056250714
## thalach  -0.007169290  0.04120981  1.000000000 -0.37741145 -0.342200746
## exang     0.024728793 -0.06880655 -0.377411449  1.00000000  0.286766336
## oldpeak   0.004514275 -0.05625071 -0.342200746  0.28676634  1.000000000
## slope    -0.058653541  0.09040215  0.384754381 -0.25610624 -0.576314382
## ca        0.144934749 -0.08311244 -0.228311083  0.12537710  0.236560442
## thal     -0.032752381 -0.01047317 -0.094909936  0.20582566  0.209090491
##                  slope          ca        thal
## age      -0.1641242337  0.30226121  0.06531729
## sex      -0.0329896331  0.11306039  0.21145220
## cp        0.1168541942 -0.19535634 -0.16036963
## trestbps -0.1228729284  0.09924834  0.06286958
## chol      0.0004166583  0.08687794  0.09681045
## fbs      -0.0586535414  0.14493475 -0.03275238
## restecg   0.0904021525 -0.08311244 -0.01047317
## thalach   0.3847543806 -0.22831108 -0.09490994
## exang    -0.2561062438  0.12537710  0.20582566
## oldpeak  -0.5763143815  0.23656044  0.20909049
## slope     1.0000000000 -0.09223637 -0.10331367
## ca       -0.0922363668  1.00000000  0.16008543
## thal     -0.1033136653  0.16008543  1.00000000
```

```
# Looking at the correlations we see that no variables is correlated strong with one another
# no multicollinearity
```

Logistic Regression Model

```
## Prepare for logistic regression
# Split data into train and test sets
set.seed(42) # reproducibility


sample_index <- sample(c(TRUE, FALSE), nrow(df), replace = TRUE, prob = c(0.7, 0.3))
train <- df[sample_index, ]
test <- df[!sample_index, ]

cat("Training set size:", nrow(train), "\n")
```

```
## Training set size: 208
```

```
cat("Testing set size:", nrow(test), "\n")
```

```
## Testing set size: 94
```

```r
# 208 observations are used to train model
# 94 are used to test model


# Ensure categorical codes are factors
train <- train %>%
  mutate(
    sex     = factor(sex),
    cp      = factor(cp),
    fbs     = factor(fbs),
    restecg = factor(restecg),
    exang   = factor(exang),
    slope   = factor(slope),
    ca      = factor(ca),
    thal    = factor(thal)
  )
# Scale all continuous predictors at once
train <- train %>%
  mutate(across(c(age, trestbps, chol, thalach, oldpeak), scale))


# Do the same for test
test <- test %>%
  mutate(
    sex     = factor(sex),
    cp      = factor(cp),
    fbs     = factor(fbs),
    restecg = factor(restecg),
    exang   = factor(exang),
    slope   = factor(slope),
    ca      = factor(ca),
    thal    = factor(thal)
  )

test <- test %>%
  mutate(across(c(age, trestbps, chol, thalach, oldpeak), scale))

### Fit Logistic Model
model <- glm(target ~ age + sex + cp + trestbps + chol + fbs + restecg + thalach + exang + oldpeak + slope + ca + thal, data = train, family = "binomial")

# view model summary
summary(model)
```

```
##
## Call:
## glm(formula = target ~ age + sex + cp + trestbps + chol + fbs +
##     restecg + thalach + exang + oldpeak + slope + ca + thal,
##     family = "binomial", data = train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -12.2634  2399.5454  -0.005 0.995922
## age            0.3144     0.3121   1.007 0.313737
## sex1          -1.4659     0.7832  -1.872 0.061240 .
## cp1            0.3569     0.7486   0.477 0.633532
## cp2            2.2842     0.7128   3.205 0.001353 **
## cp3            2.3036     0.8524   2.702 0.006884 **
## trestbps      -0.8350     0.2754  -3.032 0.002427 **
## chol           0.2892     0.3266   0.885 0.375949
## fbs1          -0.3747     0.7057  -0.531 0.595463
## restecg1       0.6471     0.5315   1.218 0.223381
## restecg2     -11.6742  1632.5926  -0.007 0.994295
## thalach        0.8317     0.3983   2.088 0.036793 *
## exang1        -1.0274     0.5865  -1.752 0.079813 .
## oldpeak       -0.5405     0.3719  -1.454 0.146059
## slope1        -2.1961     1.3034  -1.685 0.092009 .
## slope2        -0.1996     1.4556  -0.137 0.890917
## ca1           -2.4381     0.7445  -3.275 0.001057 **
## ca2           -3.4176     0.9805  -3.485 0.000491 ***
## ca3           -0.5970     1.0561  -0.565 0.571873
## ca4            2.6193     2.0295   1.291 0.196844
## thal1         15.5680  2399.5450   0.006 0.994823
## thal2         15.2155  2399.5450   0.006 0.994941
## thal3         13.4446  2399.5450   0.006 0.995529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 288.27  on 207  degrees of freedom
## Residual deviance: 114.68  on 185  degrees of freedom
## AIC: 160.68
##
## Number of Fisher Scoring iterations: 15
```

```
# See that there are alot of dummy variables

# Interpret Coefficients
# convert coefficients to odd ratios
exp(coef(model))
```

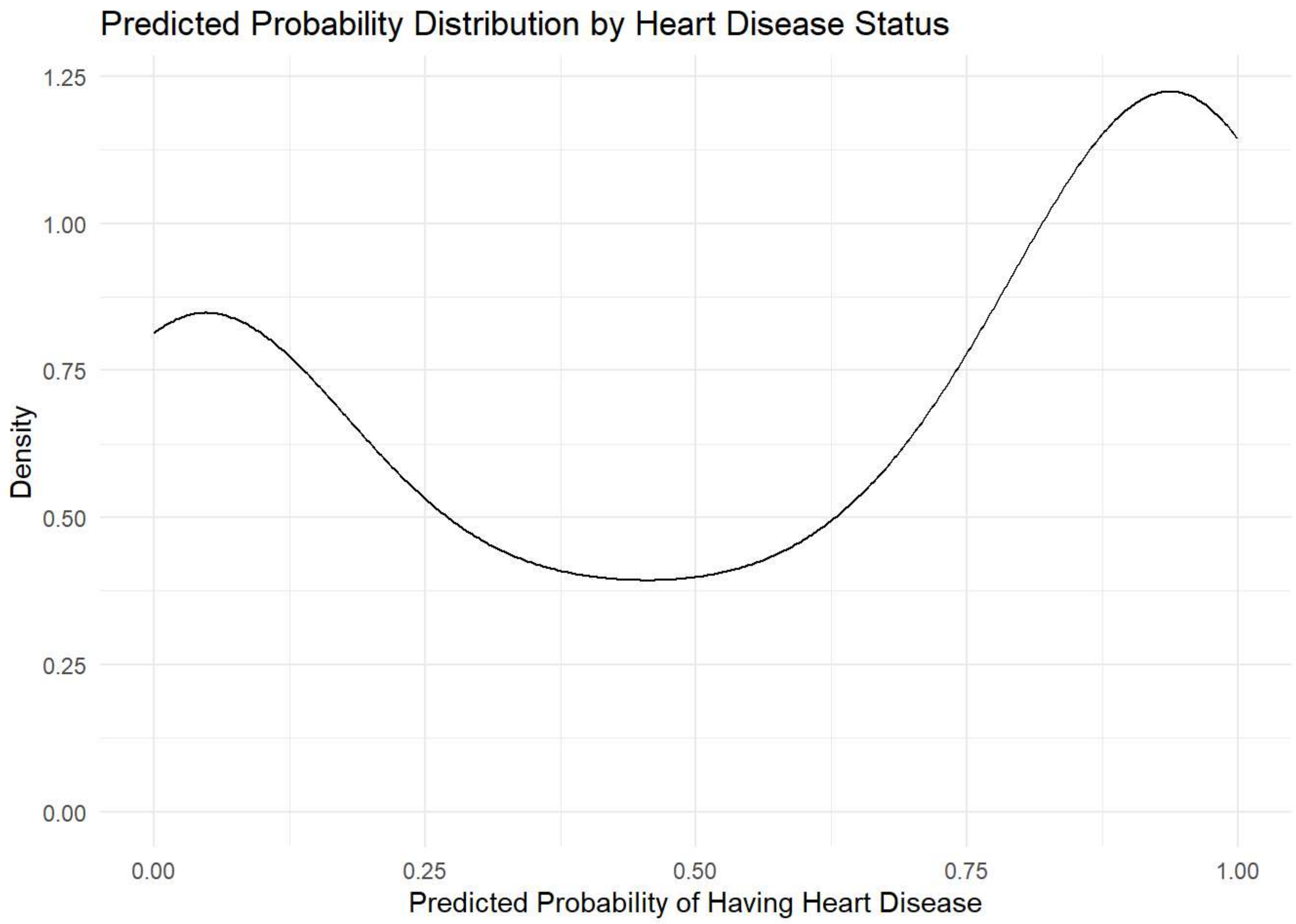
```
##  (Intercept)          age         sex1          cp1          cp2          cp3
## 4.721202e-06 1.369501e+00 2.308710e-01 1.428869e+00 9.817531e+00 1.001009e+01
##     trestbps         chol         fbs1     restecg1     restecg2      thalach
## 4.338948e-01 1.335331e+00 6.875010e-01 1.910028e+00 8.510404e-06 2.297223e+00
##       exang1      oldpeak       slope1       slope2          ca1          ca2
## 3.579484e-01 5.824414e-01 1.112358e-01 8.190326e-01 8.732483e-02 3.279132e-02
##          ca3          ca4        thal1        thal2        thal3
## 5.504527e-01 1.372566e+01 5.768730e+06 4.055267e+06 6.901145e+05
```

```
# Make predictions on test data
# predict probabilities
pred_prob <- predict(model, newdata = test, type = "response")

# Convert probabilities to class labels (1 = "Yes", 0 = "No")
pred_class <- ifelse(pred_prob > 0.5, "1", "0")



# Add predictions to test set
test$pred_prob <- pred_prob
test$pred_class <- pred_class

head(test)
```

```
##            age sex cp    trestbps       chol fbs restecg    thalach exang
## 1   0.88054758   1  3  0.83220351 -0.1876563   1       0  0.1383179     0
## 2  -2.13776698   1  2 -0.04741189  0.1523377   0       1  1.6463733     0
## 4   0.06792443   1  1 -0.63382217 -0.1276573   0       1  1.2795490     0
## 7   0.06792443   0  1  0.53899838  1.0323222   0       0  0.2605927     0
## 10  0.18401345   1  2  1.12540865 -1.4876333   0       1  1.1165160     0
## 12 -0.86078775   0  2 -0.04741189  0.6523289   0       1 -0.3100229     0
##        oldpeak slope ca thal target pred_prob pred_class
## 1   1.35092190     0  0    1      1 0.9358432          1
## 2   2.50885496     0  0    2      1 0.9790016          1
## 4  -0.09649442     2  0    2      1 0.9805557          1
## 7   0.38597769     1  0    2      1 0.7292480          1
## 10  0.67546095     2  0    2      1 0.9698030          1
## 12 -0.67546095     2  0    2      1 0.9968227          1
```

```
# Visualize Predictions
# Plot predicted by actual
ggplot(test, aes(x = pred_prob, fill = target)) +
  geom_density(alpha = 0.6) +
  labs(title = "Predicted Probability Distribution by Heart Disease Status",
       x = "Predicted Probability of Having Heart Disease", y = "Density") +
  theme_minimal()
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill.
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



```
# Evaluate Model Performance
# Confusion Matrix
conf_matrix <- table(Predicted = test$pred_class, Actual = test$target)
conf_matrix
```

```
##          Actual
## Predicted  0  1
##         0 27 13
##         1  9 45
```

```
# Confusion Matrix interpretation:

# True Negatives: 27 -> correctly predicted no heart attack

# False Positives: 13 -> predicted heart attack when actual outcome is no heart attack

# False Negatives: 9 -> predicted no heart attack when actual outcome was heart attack

# True Positive: 45 -> correctly predicted heart attack likelihood

# Accuracy
accuracy <- mean(test$pred_class == test$target)
cat("Model Accuracy:", round(accuracy, 4), "\n")
```

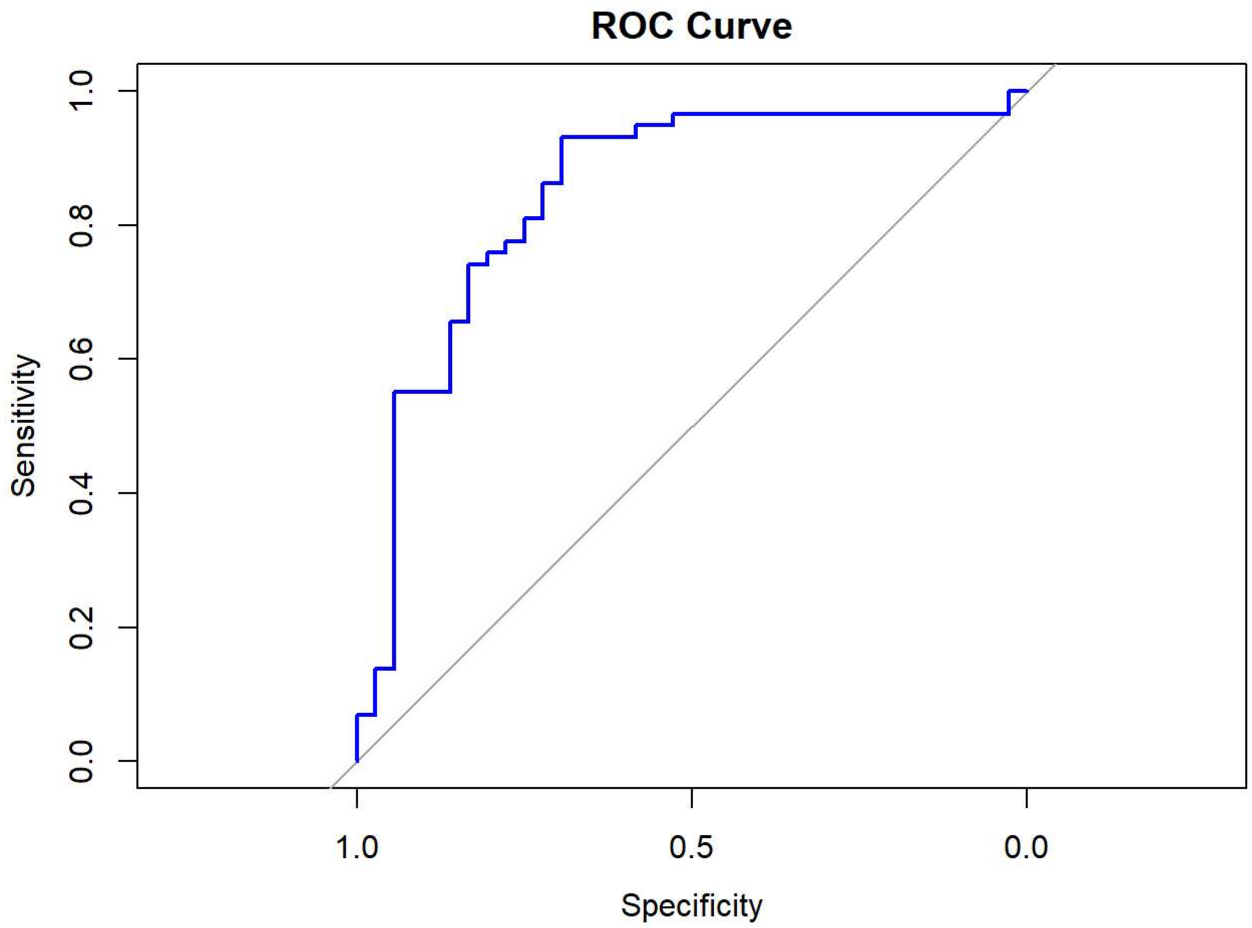
```
## Model Accuracy: 0.766
```

```
# Model is 76.6% accurate:
# correctly classifies about 77% of all cases, overall has good accurracy but can be better

# Sensitivity: TP / (TP + FN) = 45 / (45 + 9) = 83.33%
# model correctly identifies about 83% of Heart Attacks

# Specificity: TN / (TN + FP) = 27 / (27 + 13) = 67.5%
# model correctly identifies 68% of non heart attacks

# ROC Curve and AUC
roc_curve <- roc(test$target, pred_prob)
```

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

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

```
# Plot ROC Curve
plot(roc_curve, col = "blue", lwd = 2, main = "ROC Curve")
```



```
auc_value <- auc(roc_curve)
cat("AUC:", round(auc_value, 4), "\n")
```

```
## AUC: 0.8463
```

```
# AUC = 84.63

# Model has a 84.6% chance of correctly distinguishing between a randomly chosen person for hear
t attack and non heart attack -> really good discriminatory power
```