Building a Decision Tree Predictive Model

Wangjun SHEN

2023-05-27

Introduction

The goal of this assignment is to create a decision tree model that predicts heart disease. The quality of work will be improved by enhancing the presentation of results. To achieve this, more visually appealing representations, such as tables and graphics, will be used instead of code snippets.

Machine learning algorithms have become increasingly popular in the field of heart disease detection due to their potential to improve accuracy and enable early diagnosis. This assignment aims to explore the dataset and extract relevant features for analysis, while considering feedback on the selected features to ensure optimal predictive power. The data exploration process will be conducted using R and RStudio, and will involve the use of descriptive statistics and visualizations such as value distributions, histograms, bar plots, and box plots. These techniques will help us gain insights into the dataset and investigate correlations between important variables. Correlation plots or correlograms will also be used to analyze these relationships. Next, decision tree models will be built using R, taking into account parameters such as maximum depth, minimum number of samples for splitting, and complexity. Multiple models with different parameter values will be fit and their performance will be compared using metrics such as accuracy, precision, recall, F-score, and AUC.

In conclusion, this assignment aims to develop a decision tree model for heart disease prediction while addressing previous feedback on result presentation. The assignment requirements will be followed and improvements will be incorporated to deliver a comprehensive analysis of heart disease detection using decision trees.

Related Work

To be Continuous for this part

Data exploration

Explain the features you decided to extract.

The same feature set that was extracted in Assignment 1 will be used to predict heart disease. Based on feedback received, the background section was the main area of focus and no specific issues were raised with the selected feature set. Therefore, the previously chosen feature set will be maintained as it includes relevant variables associated with heart disease. These features encompass demographic information such as age, sex, and chest pain type, as well as physiological indicators like blood pressure, cholesterol levels, and maximum heart rate.

Provide descriptive statistics (using tables and figures) of your feature set

Import data set:

```
# Read the CSV file into a data frame
heart_data <- read.csv("heart_features_selected.csv")

sapply(heart_data, class)
```

```
##            id           age           sex            cp       thalach 
##     "integer"     "integer"   "character"   "character"     "integer" 
##         exang       oldpeak major_vessels        restwm        target 
##     "logical"     "numeric"     "integer"   "character"   "character" 
```

When examining a dataset, it's crucial to verify for any absent values. This is because absent values can influence the precision and credibility of the analysis, resulting in prejudiced outcomes and less efficient models. Spotting and resolving absent values can enhance the quality of the analysis.

```
# Identify missing values represented as "nan", "none", or "null" (case-insensitive)
missing_values <- c("nan", "none", "null", "NA", "N/A")
missing_count <- sapply(heart_data, function(x) sum(toupper(x) %in% toupper(missing_values), na.rm = TRU

# Create a table to display the missing value count
missing_table <- data.frame(Feature = names(missing_count), Count = missing_count)
print(missing_table)
```

```
##                     Feature Count
## id                       id     0
## age                     age     0
## sex                     sex     0
## cp                       cp     0
## thalach             thalach     0
## exang                 exang     0
## oldpeak             oldpeak     0
## major_vessels major_vessels     0
## restwm               restwm     7
## target               target     0
```

The majority of features in the dataset are complete, except for the restwm feature which contains 7 missing values. It is important to further investigate these missing values in order to maintain the accuracy and completeness of the data.

Drop those missing values:

```
# Remove rows with missing values in restwm column
heart_data <- heart_data[heart_data$restwm != "none", ]

# Verify the changes
print(head(heart_data))
```

```
##   id age    sex             cp thalach exang oldpeak major_vessels
## 1  1  52   male typical angina     168 FALSE     1.0             2
## 2  2  53   male typical angina     155  TRUE     3.1             0
## 3  3  70   male typical angina     125  TRUE     2.6             0
## 4  4  61   male typical angina     161 FALSE     0.0             1
## 5  5  62 female typical angina     106 FALSE     1.9             3
## 6  6  58 female typical angina     122 FALSE     1.0             0
##                 restwm     target
## 1 akinesis or dyskmem no disease
## 2 akinesis or dyskmem no disease
## 3 akinesis or dyskmem no disease
## 4 akinesis or dyskmem no disease
## 5  moderate or severe no disease
## 6  moderate or severe    disease
```

The majority of features in the dataset are complete, with the exception of the restwm feature, which has 7 missing values. These values will be removed directly, as the dataset is sufficiently large, and their removal will have no impact.

Drop id variable, and then convert sex, cp, exang, restwm and target into categories/factor type:

```
# Drop the "id" variable
heart_data <- heart_data[, -which(names(heart_data) == "id")]

# Convert variables to factor type
convert_vars <- c("sex", "cp", "exang", "restwm", "target")
heart_data[convert_vars] <- lapply(heart_data[convert_vars], as.factor)

str(heart_data)
```

```
## 'data.frame':    1018 obs. of  9 variables:
##  $ age          : int  52 53 70 61 62 58 58 55 46 54 ...
##  $ sex          : Factor w/ 2 levels "female","male": 2 2 2 2 1 1 2 2 2 2 ...
##  $ cp           : Factor w/ 4 levels "asymptomatic",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ thalach      : int  168 155 125 161 106 122 140 145 144 116 ...
##  $ exang        : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 1 1 1 1 2 1 2 ...
##  $ oldpeak      : num  1 3.1 2.6 0 1.9 1 4.4 0.8 0.8 3.2 ...
##  $ major_vessels: int  2 0 0 1 3 0 3 1 0 2 ...
##  $ restwm       : Factor w/ 3 levels "akinesis or dyskmem",..: 1 1 1 1 3 3 2 1 1 3 ...
##  $ target       : Factor w/ 2 levels "disease","no disease": 2 2 2 2 2 1 2 2 2 2 ...
```

Now create two sub data set for categories features and numeric features:

```
# Subset categorical variables
categories_df <- heart_data[, sapply(heart_data, is.factor)]

# Subset numeric variables
numeric_df <- heart_data[, sapply(heart_data, is.numeric)]
```

To begin, an examination will be made of how the categories are distributed:

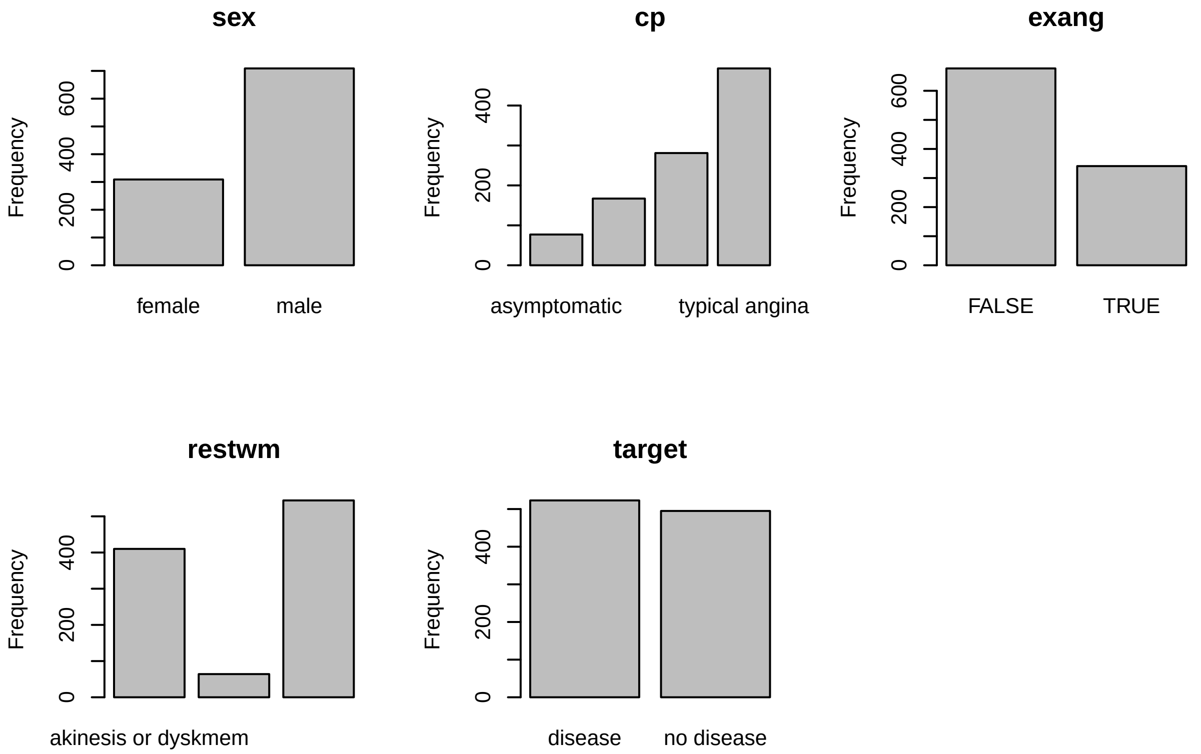
```
# Descriptive statistics for categorical variables
summary(categories_df)
```

```
##      sex                     cp        exang                     restwm
##  female:309   asymptomatic    : 77   FALSE:677   akinesis or dyskmem:410
##  male  :709   atypical angina :167   TRUE :341   mild or moderate   : 64
##               non-anginal pain:281               moderate or severe :544
##               typical angina  :493
##         target
##  disease   :523
##  no disease:495
##
##
```

```
# Value distributions
table_data <- lapply(categories_df, table)
print(table_data)
```

```
## $sex
##
## female   male
##    309    709
##
## $cp
##
##     asymptomatic  atypical angina non-anginal pain   typical angina
##               77              167              281              493
##
## $exang
##
## FALSE  TRUE
##   677   341
##
## $restwm
##
## akinesis or dyskmem    mild or moderate  moderate or severe
##                 410                  64                 544
##
## $target
##
##    disease no disease
##        523        495
```

```
# Bar plots
par(mfrow = c(2, 3))  # Set the layout for multiple plots

for (i in 1:ncol(categories_df)) {
  barplot(table_data[[i]], main = names(table_data)[i], xlab = "", ylab = "Frequency")
}

# Additional visualizations (if desired)
# ...

# Reset the layout for plots
par(mfrow = c(1, 1))
```



This document contains data on various factors related to heart disease. The sex variable has 309 samples for females and 709 for males. The cp variable has four categories, with typical angina being the most common at 493 occurrences. The exang variable has two categories, with 677 instances of FALSE and 341 instances of TRUE. The restwm variable has three categories, with moderate or severe being the most frequent at 544 occurrences. The target variable has two categories, with 523 cases of disease and 495 cases of no disease.

Then explore Numeric Features:

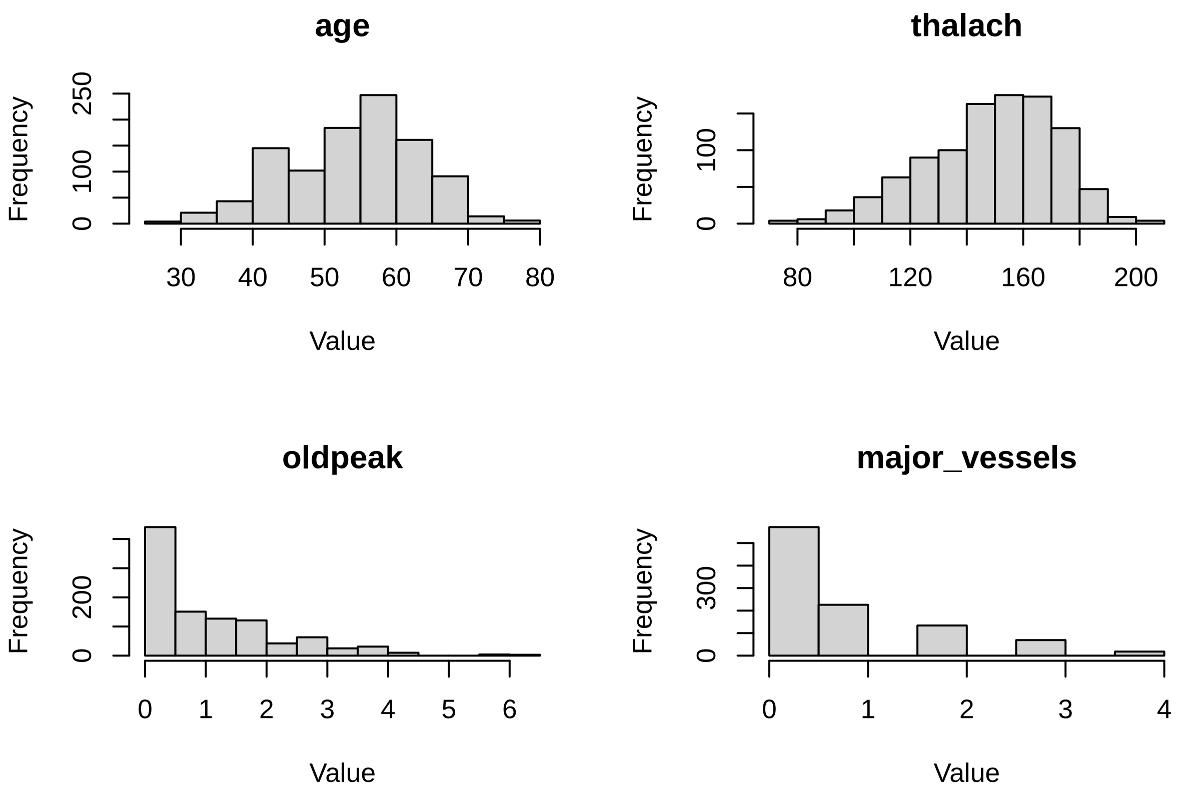
```r
# Descriptive statistics for numeric variables
summary(numeric_df)
```

```
##       age           thalach         oldpeak      major_vessels   
##  Min.   :29.00   Min.   : 71.0   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:48.00   1st Qu.:132.0   1st Qu.:0.000   1st Qu.:0.0000  
##  Median :56.00   Median :152.0   Median :0.800   Median :0.0000  
##  Mean   :54.45   Mean   :149.2   Mean   :1.075   Mean   :0.7593  
##  3rd Qu.:61.00   3rd Qu.:166.0   3rd Qu.:1.800   3rd Qu.:1.0000  
##  Max.   :77.00   Max.   :202.0   Max.   :6.200   Max.   :4.0000
```

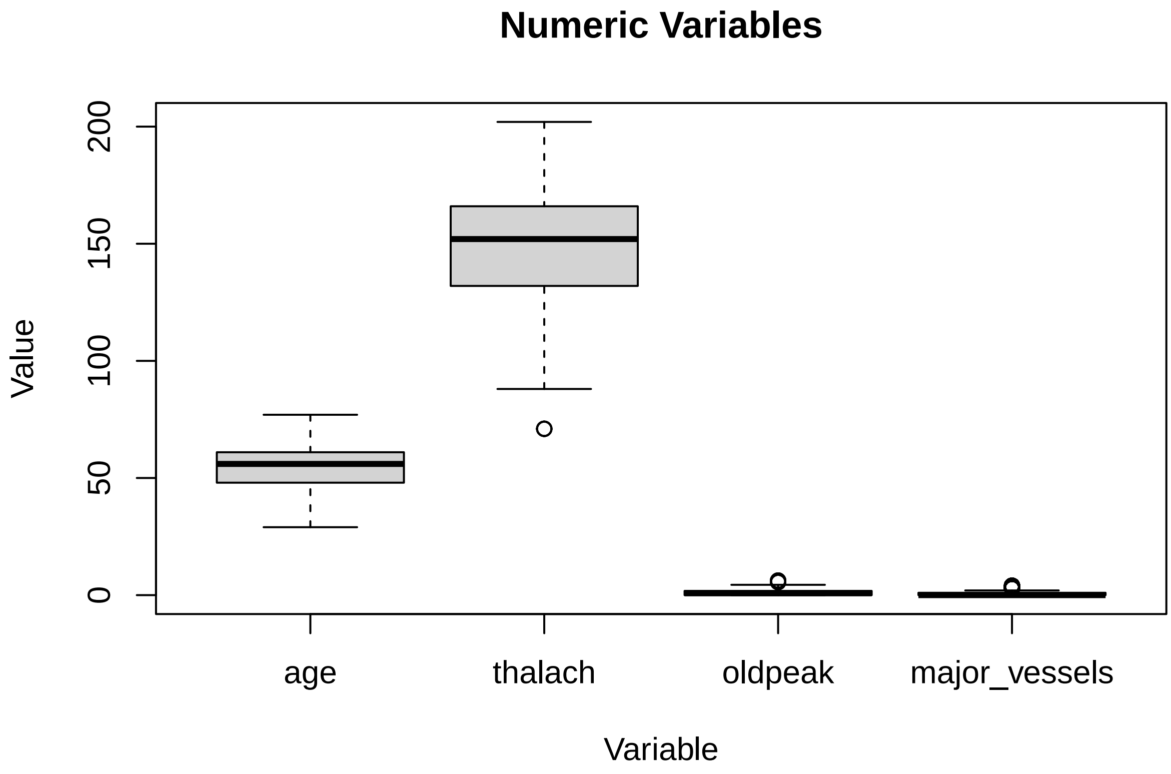
```r
# Value distributions
for (i in 1:ncol(numeric_df)) {
  cat("Variable:", names(numeric_df)[i], "\n")
  cat(table(numeric_df[, i]), "\n\n")
}
```

```
## Variable: age
## 4 6 15 6 12 14 11 32 26 26 36 25 23 18 23 17 21 39 39 23 53 30 39 57 68 46 37 31 37 32 34 27 25 31 1
##
## Variable: thalach
## 4 3 3 4 7 4 3 8 10 3 8 7 10 7 3 11 6 7 4 4 11 3 12 6 4 25 14 4 3 4 15 14 26 7 4 7 3 11 7 21 10 19 23
##
## Variable: oldpeak
## 326 23 37 10 30 15 47 3 44 10 47 6 58 3 44 16 37 36 16 32 3 14 7 11 7 21 22 3 17 4 8 10 3 15 4 12 6
##
## Variable: major_vessels
## 571 226 134 69 18
```

```
# Histograms
par(mfrow = c(2, 2))  # Set the layout for multiple plots

for (i in 1:ncol(numeric_df)) {
  hist(numeric_df[, i], main = names(numeric_df)[i], xlab = "Value", ylab = "Frequency")
}
```



```
# Box plots
par(mfrow = c(1, 1))  # Reset the layout for plots
boxplot(numeric_df, main = "Numeric Variables", xlab = "Variable", ylab = "Value")
```



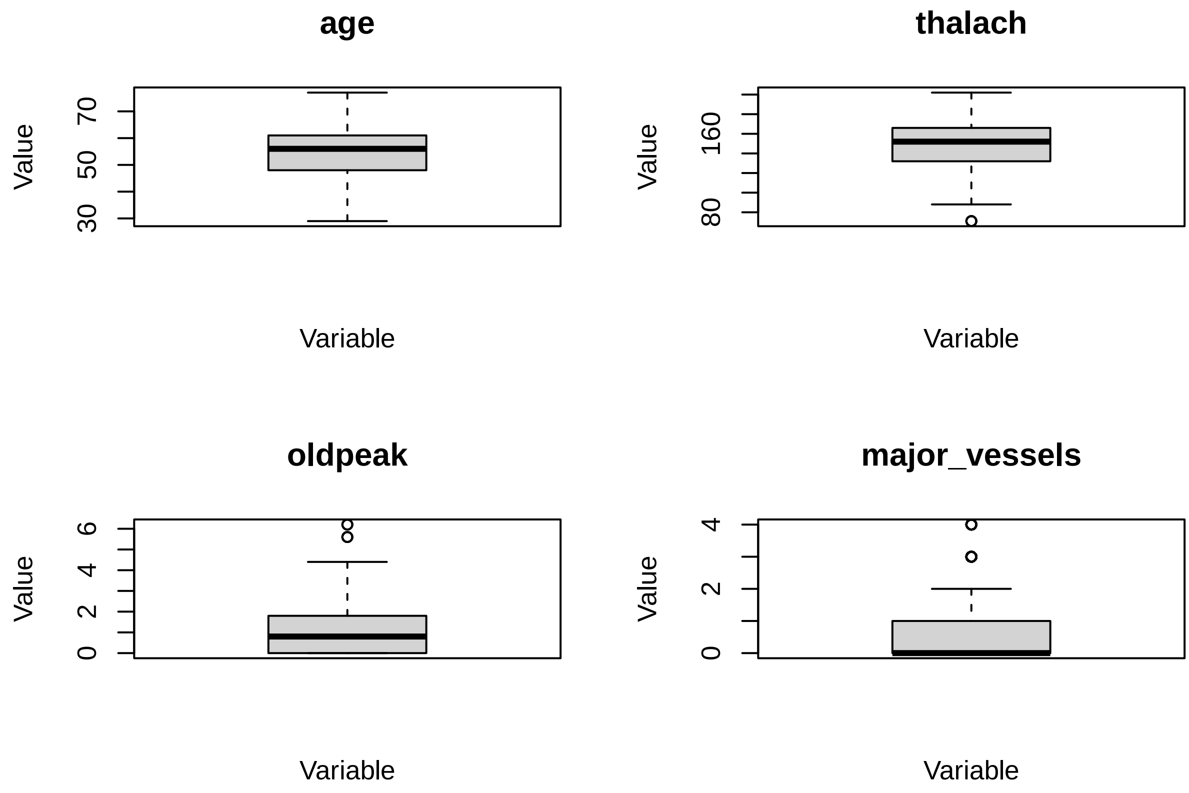
The age variable ranges from 29 to 77 years, with a median age of 56 years and a mean age of 54.45 years. The majority of individuals fall between the ages of 48 to 61 years. The maximum heart rate achieved (thalach) ranges from 71 to 202 beats per minute (bpm). The median heart rate is 152 bpm, with a mean heart rate of 149.2 bpm. Most individuals have a heart rate between 132 and 166 bpm. The ST depression induced by exercise relative to rest (oldpeak) ranges from 0 to 6.2. The median value is 0.8, indicating a mild ST depression on average. The mean value is 1.075, indicating a slightly higher overall ST depression. The number of major vessels colored by fluoroscopy (major_vessels) ranges from 0 to 4. Most individuals have 0 major vessels colored, with a median value of 0 and a mean value of 0.7593. However, there is some variability, as some individuals have up to 4 major vessels colored.

The above figure shows the visualization results of the distribution of numeric features, which is convenient for intuitive observation.

Box plots provide a visual representation of the data distribution, including any potential outliers, and can be used to identify points located outside the whiskers or outside the box. The age data does not contain any apparent outliers, as both the minimum and maximum values fall within a reasonable range for human age. However, for thalach, there are some values that exceed both the upper quartile (Q3) and the maximum value. These values could potentially be considered outliers. No extreme or outlier values are present for the ST depression caused by exercise relative to rest (oldpeak) or the number of major vessels stained by fluoroscopy (major_vessels).

```r
# Detect outliers in numeric variables
outliers <- sapply(numeric_df, function(x) {
  lower <- quantile(x, probs = 0.25) - 1.5 * IQR(x)
  upper <- quantile(x, probs = 0.75) + 1.5 * IQR(x)
  x[x < lower | x > upper]
})
```

```
# Create a combined plot for outlier visualization
par(mfrow = c(2, 2))  # Set the layout for multiple plots

# Plot boxplots with outliers highlighted
for (i in 1:ncol(numeric_df)) {
  boxplot(numeric_df[, i], main = names(numeric_df)[i], xlab = "Variable", ylab = "Value", outline = !is
}
```



```
# Reset the layout for plots
par(mfrow = c(1, 1))
```

Each numerical variable will then be examined for any values that fall outside the upper and lower limits, which are considered outliers and may indicate the presence of extreme data values.

```
# Identify rows with outliers in numeric variables
outlier_rows <- apply(sapply(numeric_df, function(x) {
  lower <- quantile(x, probs = 0.25) - 1.5 * IQR(x)
  upper <- quantile(x, probs = 0.75) + 1.5 * IQR(x)
  x < lower | x > upper
}), 1, any)

# Remove rows with outliers from heart_data
heart_data <- heart_data[!outlier_rows, ]
```

```
# Verify the updated dataset
str(heart_data)
```

```
## ’data.frame':    923 obs. of  9 variables:
##  $ age          : int  52 53 70 61 58 55 46 54 71 43 ...
##  $ sex          : Factor w/ 2 levels "female","male": 2 2 2 2 1 2 2 2 1 1 ...
##  $ cp           : Factor w/ 4 levels "asymptomatic",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ thalach      : int  168 155 125 161 122 145 144 116 125 136 ...
##  $ exang        : Factor w/ 2 levels "FALSE","TRUE": 1 2 2 1 1 2 1 2 1 2 ...
##  $ oldpeak      : num  1 3.1 2.6 0 1 0.8 0.8 3.2 1.6 3 ...
##  $ major_vessels: int  2 0 0 1 0 1 0 2 0 0 ...
##  $ restwm       : Factor w/ 3 levels "akinesis or dyskmem",..: 1 1 1 1 3 1 1 3 3 1 ...
##  $ target       : Factor w/ 2 levels "disease","no disease": 2 2 2 2 1 2 2 2 1 2 ...
```

The Correlation Between Variables

For features in categories_df:

```
# Convert categorical variables to numeric representation
numeric_categories_df <- sapply(categories_df, as.numeric)

# Compute the correlation matrix
cor_matrix <- cor(numeric_categories_df)

# Display the correlation values
cor_matrix
```

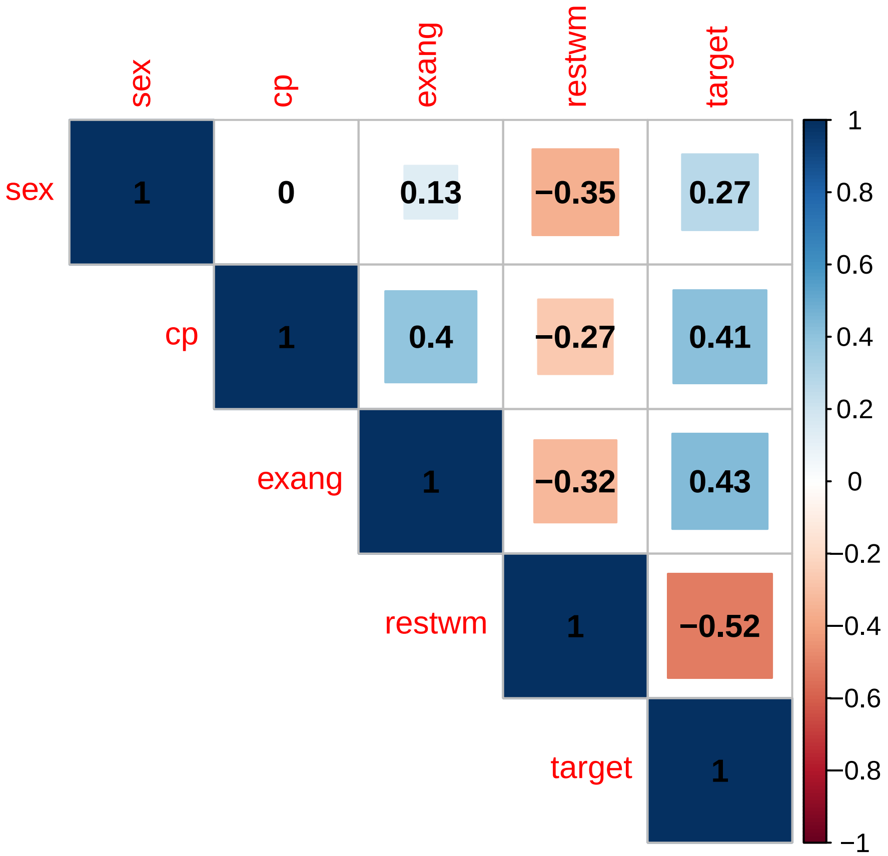
```
##                 sex           cp      exang     restwm     target
## sex     1.000000000  0.004912486  0.1335619 -0.3526975  0.2746429
## cp      0.004912486  1.000000000  0.3974181 -0.2662407  0.4141313
## exang   0.133561926  0.397418085  1.0000000 -0.3230982  0.4338575
## restwm -0.352697478 -0.266240739 -0.3230982  1.0000000 -0.5187868
## target  0.274642933  0.414131286  0.4338575 -0.5187868  1.0000000
```

```
library(corrplot)
```

```
## Warning: package ’corrplot’ was built under R version 4.2.3
```

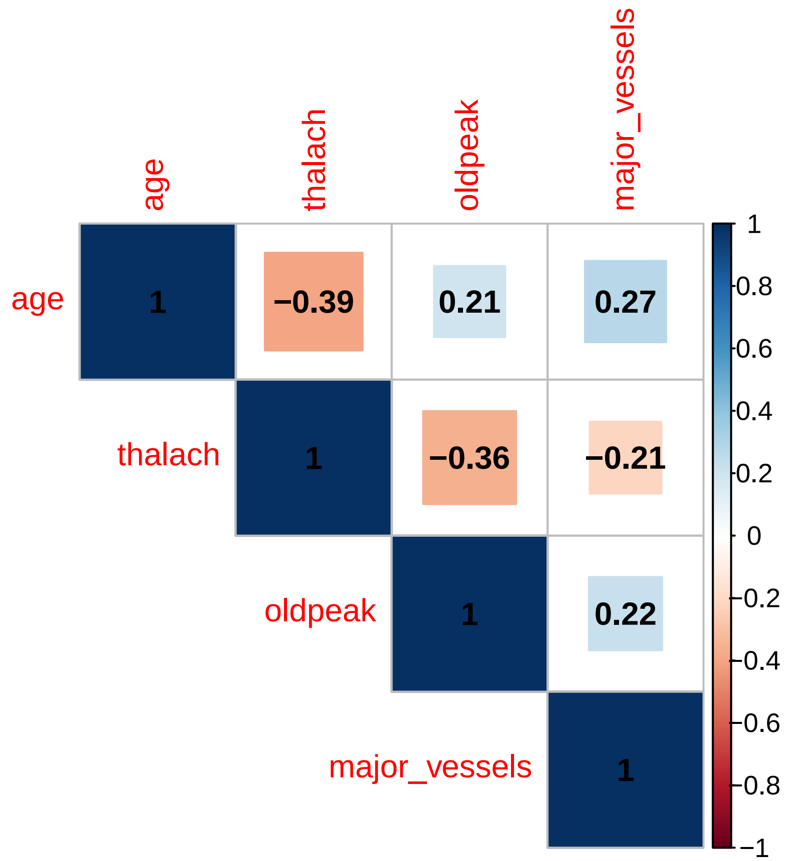
```
## corrplot 0.92 loaded
```

```
# Convert categorical variables to numeric representation
numeric_categories_df <- sapply(categories_df, as.numeric)

# Compute the correlation matrix
cor_matrix <- cor(numeric_categories_df)

# Create a correlation matrix plot with values displayed in squares
corrplot(cor_matrix, method = "square", type = "upper", addCoef.col = "black")
```



Several correlation coefficients were observed between different variables and the likelihood of heart attack. Notably, a positive correlation of 0.27 was found between sex and target, suggesting a potential association between gender and the probability of heart attack. Additionally, a moderate positive correlation of 0.41 was observed between cp and target, indicating a potential link between chest pain type and the likelihood of heart attack. Moreover, a moderate positive correlation of 0.43 was present between exang and target, implying a possible connection between exercise-induced angina and the probability of heart attack. Finally, a strong negative correlation of -0.52 was observed between restwm and target, suggesting a significant association between abnormal resting electrocardiogram and the probability of heart attack.

```
cor_matrix <- cor(numeric_df)

library(corrplot)
corrplot(cor_matrix, method = "square", type = "upper", addCoef.col = "black")
```



```
cor_matrix
```

```
##                     age    thalach    oldpeak major_vessels
## age           1.0000000 -0.3920101  0.2079390     0.2709852
## thalach      -0.3920101  1.0000000 -0.3553421    -0.2113357
## oldpeak       0.2079390 -0.3553421  1.0000000     0.2203498
## major_vessels 0.2709852 -0.2113357  0.2203498     1.0000000
```

Now import the data set:

```
write.csv(heart_data, file = "features_prepare_for_modelling.csv", row.names = FALSE)
```

Building Decision Tree Models

sdfsdafasdf