Final Project

MATH 40028/50028: Statistical Learning

May 5, 2024

Introduction

Heart disease remains one of the leading causes of death worldwide, prompting extensive research efforts aimed at early diagnosis and effective treatment.Therfore, the availability of comprehensive datasets plays a crucial role in advancing machine learning for clinical diagnosis. The Heart Disease Dataset presented here represents a significant contribution to this field, as it combines information from five popular heart disease datasets into a single resource.

Dataset Overview

This dataset has been created by combining information from various independent sources, including the Cleveland, Hungarian, Switzerland, Long Beach VA, and Statlog (Heart) datasets. This comprehensive collection includes 1,190 entries with 11 important characteristics. By pooling data from these different sources, the dataset provides a wider range of opportunities for research and analysis than would be possible with any single dataset alone.

Features and Attributes

The dataset encompasses a range of demographic, clinical, and physiological attributes, including age, sex, chest pain type, resting blood pressure, serum cholesterol levels, fasting blood sugar, resting electrocardiogram results, maximum heart rate achieved, exercise-induced angina, old peak depression, and the slope of the peak exercise ST segment. Each attribute offers valuable insights into the various risk factors and indicators associated with heart disease.

The Attributes and There Possible values:

Attribute	Code given	Unit	Data type
age	Age	years	Numeric
sex	Sex	1, 0	Binary
chest pain type	Chest pain type	1,2,3,4	Nominal
resting blood pressure	Resting bp	mm Hg	Numeric
serum cholesterol	Cholesterol	mg/dl	Numeric
fasting blood sugar	Fasting blood	1,0 > 120 mg/dl	Binary
resting electrocardiogram results	Resting ecg	0,1,2	Nominal
maximum heart rate achieved	Max heart rate	71–202	Numeric
exercise-induced angina	Exercise angina	0,1	Binary
oldpeak	ST oldpeak		Numeric
the slope of the peak exercise ST segment	ST slope	0,1,2	Nominal
class	Target	0,1	Binary

Heart Disease Dataset Attributes Description.

Attribute	Description
age	Age of the individual in years
sex	Gender of the individual (1 = male, 0 = female)
chest pain type	Type of chest pain (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)
resting blood pressure	Resting blood pressure in mm Hg
serum cholesterol	Serum cholesterol level in mg/dl
fasting blood sugar	Fasting blood sugar level (> 120 mg/dl: 1=true, <= 120 mg/dl: 0=false)
resting electrocardiogram results	Resting electrocardiogram results (0: normal, 1: abnormal ST-T wave, 2: probable left ventricular hypertrophy)
maximum heart rate achieved	Maximum heart rate achieved
exercise-induced angina	Exercise-induced angina (1=yes, 0=no)
oldpeak	ST depression induced by exercise relative to rest
the slope of the peak exercise ST segment	Slope of the peak exercise ST segment (1: upsloping, 2: flat, 3: downsloping)
class	Presence of heart disease (1=heart disease, 0=normal)

Loading Data and checking for Nulls

```
# Load the CSV data
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(visdat)
```

```
## Warning: package 'visdat' was built under R version 4.3.3
```

```
library(cowplot)
```

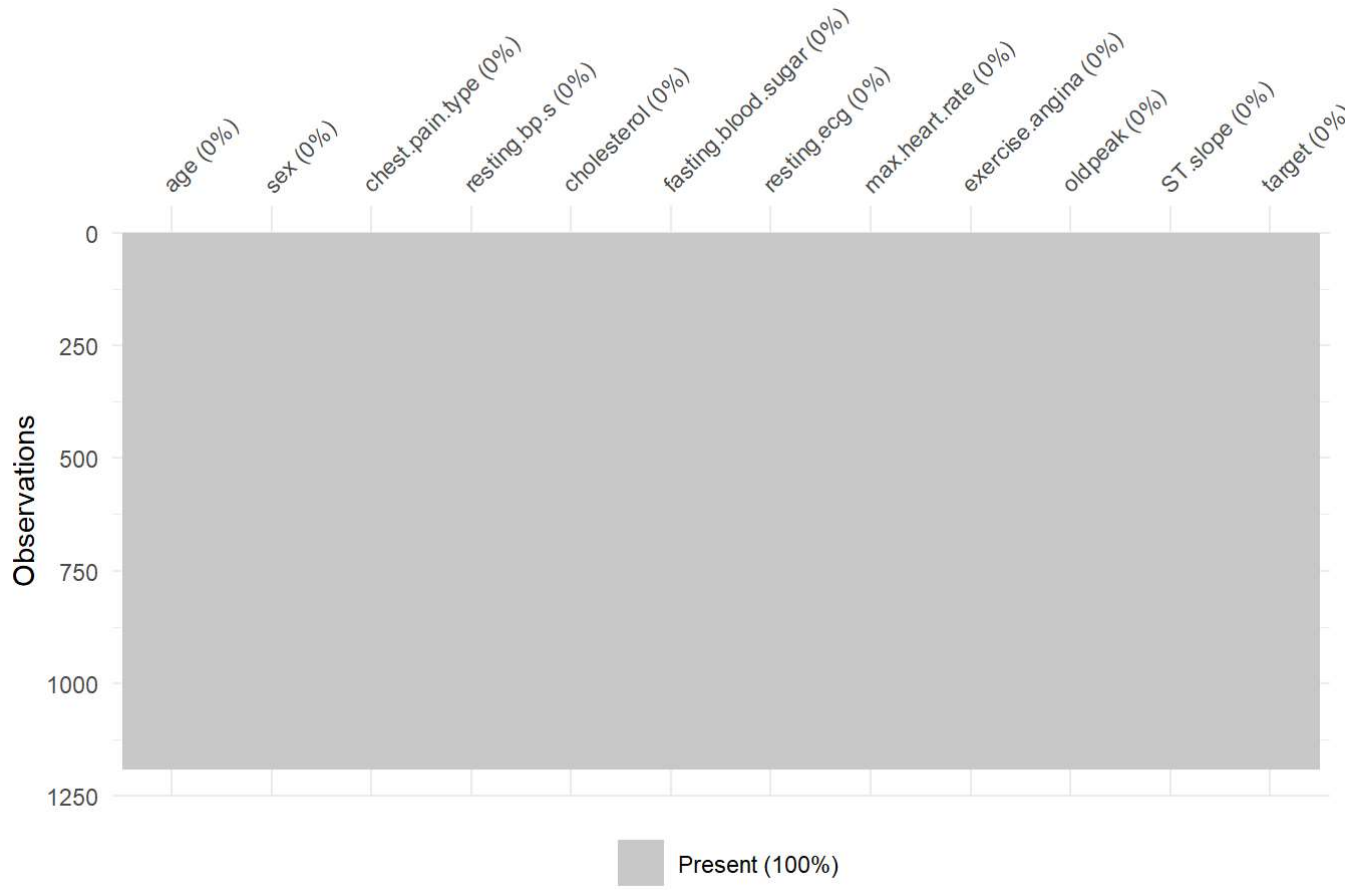
```
## Warning: package 'cowplot' was built under R version 4.3.3
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```
heart_data <- read.csv("C:/Users/A U C/Downloads/statistical_final_project/heart_statlog_clevela
nd_hungary_final.csv")

vis_miss(heart_data)
```



Identify and defining Pridiction Problems

In this project, we aim to explore the Heart Disease Dataset to perform predictive analysis using statistical learning methods. By investigating the relationships between various features and the presence of heart disease, we seek to develop models capable of accurately predicting the likelihood of an individual having heart disease (where target=1 indicates the presence of heart disease).

Checking for Invalid Observation and Remove them (Out of the

allowed range)

```
valid_ranges <- list(
  age = c(0, 120),  # Assum age is between 0 and 120
  `resting.bp.s` = c(0, 300),  # Assum resting bp s is between 0 and 300
  cholesterol = c(0, Inf),  # Assumcholesterol is non-negative
  `max.heart.rate` = c(70, 202)  # Assum max heart rate is between 0 and 202
 )

# Check validity of each variable separately
for (var in names(valid_ranges)) {
  invalid_indices <- which(heart_data[[var]] <= valid_ranges[[var]][1] | heart_data[[var]] > val
id_ranges[[var]][2])
  num_invalid <- length(invalid_indices)
  if (num_invalid > 0) {
    cat(paste("Invalid observations found for variable", var, ":", num_invalid, "\n"))
    cat("Reason for invalidity:\n")
    if (any(heart_data[[var]] <= valid_ranges[[var]][1])) {
      cat(paste(" - Values below minimum valid range less than or equal(", valid_ranges[[var]]
[1], ")\n", sep = ""))
    }
    if (any(heart_data[[var]] > valid_ranges[[var]][2])) {
      cat(paste(" - Values above maximum valid range (", valid_ranges[[var]][2], ")\n", sep =
""))
    }
  } else {
    cat(paste("All observations for variable", var, "are within their valid range.\n"))
  }
}
```

```
## All observations for variable age are within their valid range.
## Invalid observations found for variable resting.bp.s : 1
## Reason for invalidity:
##  - Values below minimum valid range less than or equal(0)
## Invalid observations found for variable cholesterol : 172
## Reason for invalidity:
##  - Values below minimum valid range less than or equal(0)
## Invalid observations found for variable max.heart.rate : 5
## Reason for invalidity:
##  - Values below minimum valid range less than or equal(70)
```

```r
# Define valid ranges for each variable
valid_ranges <- list(
  age = c(0, 100),  # Assuming age is between 0 and 100
  `resting.bp.s` = c(0, 300),  # Assuming resting bp s is between 0 and 300
  cholesterol = c(0, Inf),  # Assuming cholesterol is non-negative
  `max.heart.rate` = c(70, 500)  # Assuming max heart rate is between 70 and 202
)

# check validity of each variable separately and remove invalid rows
valid_rows <- rep(TRUE, nrow(heart_data))
for (var in names(valid_ranges)) {
  valid_rows <- valid_rows &
    (heart_data[[var]] > valid_ranges[[var]][1] &
       heart_data[[var]] <= valid_ranges[[var]][2])
}

# remove invalid rows
heart_data <- heart_data[valid_rows, ]
```

```r
valid_ranges <- list(
  age = c(0, 100),
  `resting.bp.s` = c(0, 300),
  cholesterol = c(0, Inf),
  `max.heart.rate` = c(70, 500)
 )


for (var in names(valid_ranges)) {
  invalid_indices <- which(heart_data[[var]] <= valid_ranges[[var]][1] | heart_data[[var]] > val
id_ranges[[var]][2])
  num_invalid <- length(invalid_indices)
  if (num_invalid > 0) {
    cat(paste("Invalid observations found for variable", var, ":", num_invalid, "\n"))
    cat("Reason for invalidity:\n")
    if (any(heart_data[[var]] <= valid_ranges[[var]][1])) {
      cat(paste(" - Values below minimum valid range less than or equal(", valid_ranges[[var]]
[1], ")\n", sep = ""))
    }
    if (any(heart_data[[var]] > valid_ranges[[var]][2])) {
      cat(paste(" - Values above maximum valid range (", valid_ranges[[var]][2], ")\n", sep =
""))
    }
  } else {
    cat(paste("All observations for variable", var, "are within their valid range.\n"))
  }
}
```

```
## All observations for variable age are within their valid range.
## All observations for variable resting.bp.s are within their valid range.
## All observations for variable cholesterol are within their valid range.
## All observations for variable max.heart.rate are within their valid range.
```

```
#heart_data <- heart_data[heart_data$sex != 0, ]
summary(heart_data)
```

```
##       age             sex         chest.pain.type  resting.bp.s
##  Min.   :28.00   Min.   :0.0000   Min.   :1.000   Min.   : 92.0
##  1st Qu.:46.00   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:120.0
##  Median :54.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :53.29   Mean   :0.7345   Mean   :3.164   Mean   :132.6
##  3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##   cholesterol    fasting.blood.sugar  resting.ecg     max.heart.rate
##  Min.   : 85.0   Min.   :0.0000      Min.   :0.0000   Min.   : 71.0
##  1st Qu.:209.0   1st Qu.:0.0000      1st Qu.:0.0000   1st Qu.:125.0
##  Median :240.0   Median :0.0000      Median :0.0000   Median :144.0
##  Mean   :245.7   Mean   :0.1613      Mean   :0.7414   Mean   :142.8
##  3rd Qu.:276.0   3rd Qu.:0.0000      3rd Qu.:2.0000   3rd Qu.:161.0
##  Max.   :603.0   Max.   :1.0000      Max.   :2.0000   Max.   :202.0
##  exercise.angina     oldpeak           ST.slope         target
##  Min.   :0.0000   Min.   :-0.1000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:1.000   1st Qu.:0.000
##  Median :0.0000   Median : 0.6000   Median :2.000   Median :0.000
##  Mean   :0.3707   Mean   : 0.9391   Mean   :1.585   Mean   :0.469
##  3rd Qu.:1.0000   3rd Qu.: 1.6000   3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   :1.0000   Max.   : 6.2000   Max.   :3.000   Max.   :1.000
```

Expalnation of What we did

**Some observations have 0 cholesterol, which is impossible in real life, so we removed these observations because most probably there is a problem with the machine that was used to take these observations.

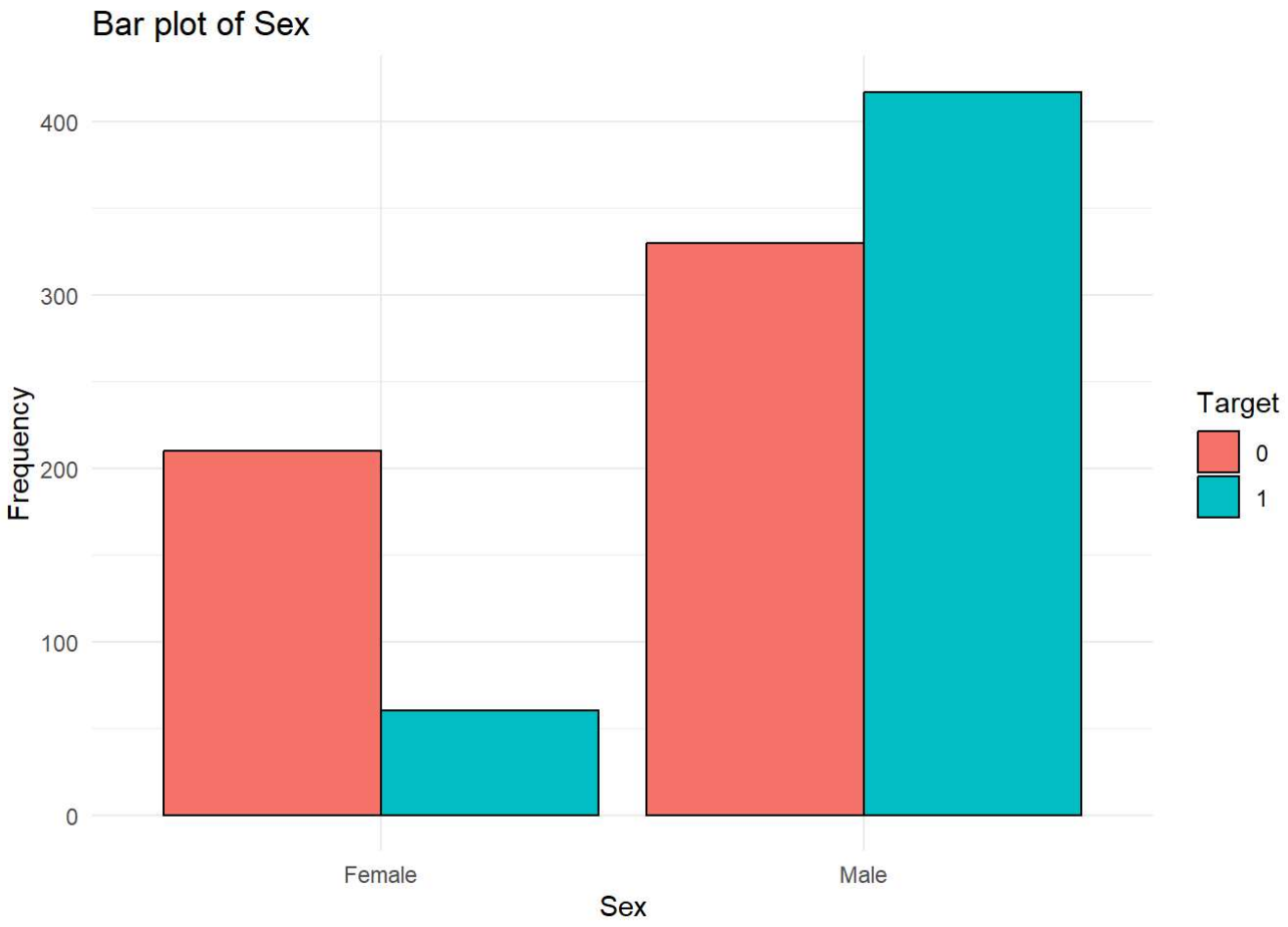
**One of the observations has zero resting blood pressure, which means that he/she is dead or the measurement is wrong.

**Five of the observations have a max heart rate out of the specified range in the dataset itself.

**We removed these rows to make sure that the dataset is cleaned and all observations are correct.

Potential Bias and a Suggested Solution

```
heart_data$sex <- factor(heart_data$sex, levels = c(0, 1), labels = c( "Female", "Male"))
target_0_data <- heart_data[heart_data$target == 0, ]
target_1_data <- heart_data[heart_data$target == 1, ]

# combine data for both target values
combined_data <- rbind(
  transform(target_0_data, target = "0"),
  transform(target_1_data, target = "1")
)

# bar plot for the sex variable
barplot_sex <- ggplot(combined_data, aes(x = sex, fill = target)) +
  geom_bar(position = "dodge", color = "black") +
  labs(x = "Sex", y = "Frequency", fill = "Target") +
  ggtitle("Bar plot of Sex") +
  theme_minimal()


print(barplot_sex)
```



Observations from the Sex Bar Plot

Data Imbalance

- **The number of males is much higher than females**, which can significantly influence the outcomes of the machine learning model.

Gender Disparity in Heart Disease Prevalence

- **Males**: A significant number of males are shown to have heart disease (Blue bar), outnumbering those without it (red bar). This indicates a higher prevalence of heart disease among males in the dataset.
- **Females**: The plot shows fewer females with heart disease compared to males, and a lower ratio of heart disease presence among females, suggesting a lower prevalence.

Rationale for Separate Analyses

The bar plot not only highlights disparities in heart disease prevalence between genders but also shows differences in the population sizes of males and females. These variations can potentially bias statistical analyses and outcomes.The larger sample size of males could dominate overall statistics and outcomes when combined with the smaller sample size of females.

proposed Analytical Strategy

In light of the observed discrepancies in population sizes and disease prevalence rates, we propose the following tailored approach: - **Creation of Separate Datasets**: Split the dataset into two groups based on gender. This separation facilitates targeted analysis that can more accurately reflect the unique conditions of each group.

- **Distinct Data Analyses**: Conduct separate exploratory data analyses (EDA) for each gender. This step will uncover specific attributes and their correlations within each group.
- **Different Statistical Methods**: Depending on the unique characteristics and insights gained from the EDA, apply potentially different statistical methods for each group. These could range from logistic regression to decision trees, or even advanced machine learning algorithms.

Benefits of This Approach

- **Enhanced Model Accuracy**: Separate analyses enable the development of more precise models, specifically tuned to the dynamics of each gender group.
- **Fairness and Equity in Analysis**: By analyzing the data separately by gender, we ensure that the analysis outcomes are not biased by the predominance of one group over another, promoting fairness and equity.

Data Splitting

** First we will divide the dataset into two datasets(one for males and other for females) to explore and train each data separately.

** We will split each dataset randomly to have a training and testing datasets to perform CV and estimate the classfiers' performance and then confirm this by predicting the testing dataset targets (75% for training and 25% for testing).

```
# create a dataset for males
male_data <- heart_data[heart_data$sex == "Male", ]

# create a dataset for females
female_data <- heart_data[heart_data$sex == "Female", ]
```

```r
set.seed(123)

# proportion of data for training and testing
train_prop <- 0.75

# generate random indices for splitting
train_indices <- sample(nrow(female_data), round(train_prop * nrow(female_data)))

# training and testing datasets
train_data_f <- female_data[train_indices, ]
test_data_f <- female_data[-train_indices, ]
```

```r
set.seed(123)

#  proportion of data for training and testing
train_prop <- 0.75  # 75% for training, 20% for testing

# random indices for splitting
train_indices <- sample(nrow(male_data), round(train_prop * nrow(male_data)))

# training and testing datasets
train_data_m <- male_data[train_indices, ]
test_data_m <- male_data[-train_indices, ]
```

```r
summary(train_data_f)
```

```
##       age            sex      chest.pain.type  resting.bp.s    cholesterol
##  Min.   :30.00   Female:202   Min.   :1.00    Min.   : 94.0   Min.   :141.0
##  1st Qu.:46.00   Male  :  0   1st Qu.:2.00    1st Qu.:120.0   1st Qu.:213.2
##  Median :54.00                Median :3.00    Median :130.0   Median :249.5
##  Mean   :53.15                Mean   :2.99    Mean   :132.1   Mean   :258.0
##  3rd Qu.:61.00                3rd Qu.:4.00    3rd Qu.:140.0   3rd Qu.:292.5
##  Max.   :76.00                Max.   :4.00    Max.   :200.0   Max.   :564.0
##  fasting.blood.sugar  resting.ecg     max.heart.rate  exercise.angina
##  Min.   :0.0000      Min.   :0.0000   Min.   : 90.0   Min.   :0.0000
##  1st Qu.:0.0000      1st Qu.:0.0000   1st Qu.:133.5   1st Qu.:0.0000
##  Median :0.0000      Median :0.0000   Median :152.0   Median :0.0000
##  Mean   :0.1188      Mean   :0.7327   Mean   :148.7   Mean   :0.2228
##  3rd Qu.:0.0000      3rd Qu.:2.0000   3rd Qu.:163.0   3rd Qu.:0.0000
##  Max.   :1.0000      Max.   :2.0000   Max.   :192.0   Max.   :1.0000
##     oldpeak          ST.slope        target
##  Min.   :0.0000   Min.   :1.00   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:1.00   1st Qu.:0.0000
##  Median :0.2000   Median :1.00   Median :0.0000
##  Mean   :0.6639   Mean   :1.45   Mean   :0.2228
##  3rd Qu.:1.0750   3rd Qu.:2.00   3rd Qu.:0.0000
##  Max.   :6.2000   Max.   :3.00   Max.   :1.0000
```

```r
summary(train_data_m)
```

```
##       age            sex      chest.pain.type  resting.bp.s    cholesterol
##  Min.   :28.00   Female:  0   Min.   :1.000   Min.   : 92.0   Min.   :100.0
##  1st Qu.:46.00   Male  :560   1st Qu.:3.000   1st Qu.:120.0   1st Qu.:207.0
##  Median :54.00                Median :4.000   Median :130.0   Median :235.5
##  Mean   :53.11                Mean   :3.225   Mean   :132.7   Mean   :242.2
##  3rd Qu.:59.00                3rd Qu.:4.000   3rd Qu.:140.0   3rd Qu.:275.0
##  Max.   :77.00                Max.   :4.000   Max.   :200.0   Max.   :603.0
##  fasting.blood.sugar  resting.ecg     max.heart.rate  exercise.angina
##  Min.   :0.000       Min.   :0.0000   Min.   : 71.0   Min.   :0.0000
##  1st Qu.:0.000       1st Qu.:0.0000   1st Qu.:124.0   1st Qu.:0.0000
##  Median :0.000       Median :0.0000   Median :140.0   Median :0.0000
##  Mean   :0.175       Mean   :0.7196   Mean   :140.8   Mean   :0.4232
##  3rd Qu.:0.000       3rd Qu.:2.0000   3rd Qu.:160.0   3rd Qu.:1.0000
##  Max.   :1.000       Max.   :2.0000   Max.   :202.0   Max.   :1.0000
##     oldpeak         ST.slope         target
##  Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.000
##  Median :0.800   Median :2.000   Median :1.000
##  Mean   :1.014   Mean   :1.636   Mean   :0.575
##  3rd Qu.:1.725   3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   :5.600   Max.   :3.000   Max.   :1.000
```

Statistical learning strategies and methods

- Perform exploratory data analysis using the training set.
- Describe the statistical learning approaches and other strategies for feature engineering (transformation, selection, etc.).
- Based on the conditions assumed by the statistical learning methods, discuss their applicability to the prediction problem.

Analysis of Numaric Variables

Males

```
library(gridExtra)
```

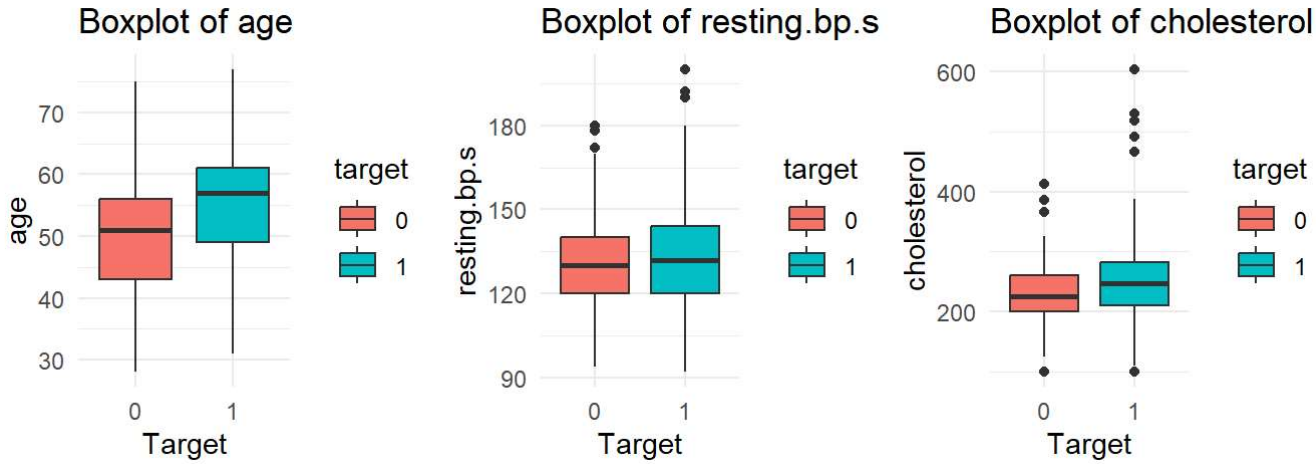
```
## Warning: package 'gridExtra' was built under R version 4.3.3
```

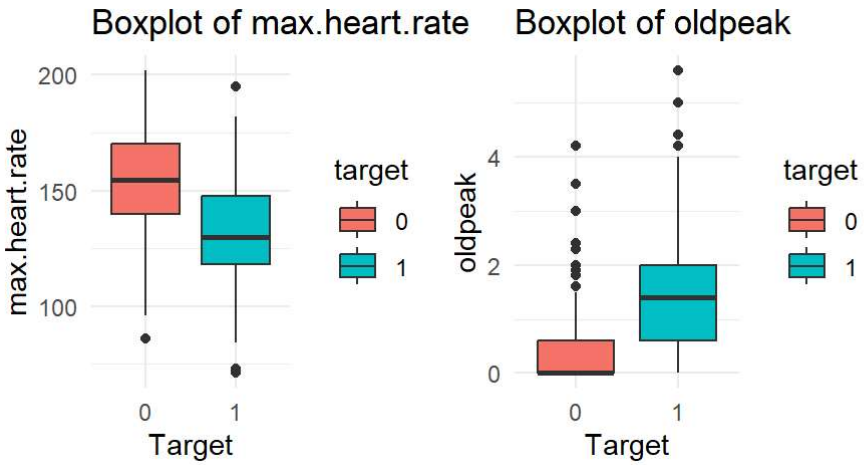
```
target_0_data_m <- train_data_m[train_data_m$target == 0, ]
target_1_data_m <- train_data_m[train_data_m$target == 1, ]
# combine data for both target values and add a new column for target labels
combined_data <- rbind(
  transform(target_0_data_m, target = "0"),
  transform(target_1_data_m, target = "1")
)
continuous_vars <- train_data_m[, c('age', 'resting.bp.s', 'cholesterol', 'max.heart.rate', 'old
peak')]
# List to store histograms
histograms_continuous <- lapply(names(continuous_vars), function(var) {

  # Histogram for both target values
  hist_combined <- ggplot(combined_data, aes_string(x = var, fill = "target")) +
    geom_histogram(color = "black", bins = 20, alpha = 0.5) +
    labs(x = var, y = "Frequency", fill = "Target") +
    ggtitle(paste("Histogram of", var)) +
    theme_minimal()

  # Return histogram
  hist_combined
})
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
#arrange histograms in a grid
grid.arrange(grobs = histograms_continuous, ncol = 3)
```

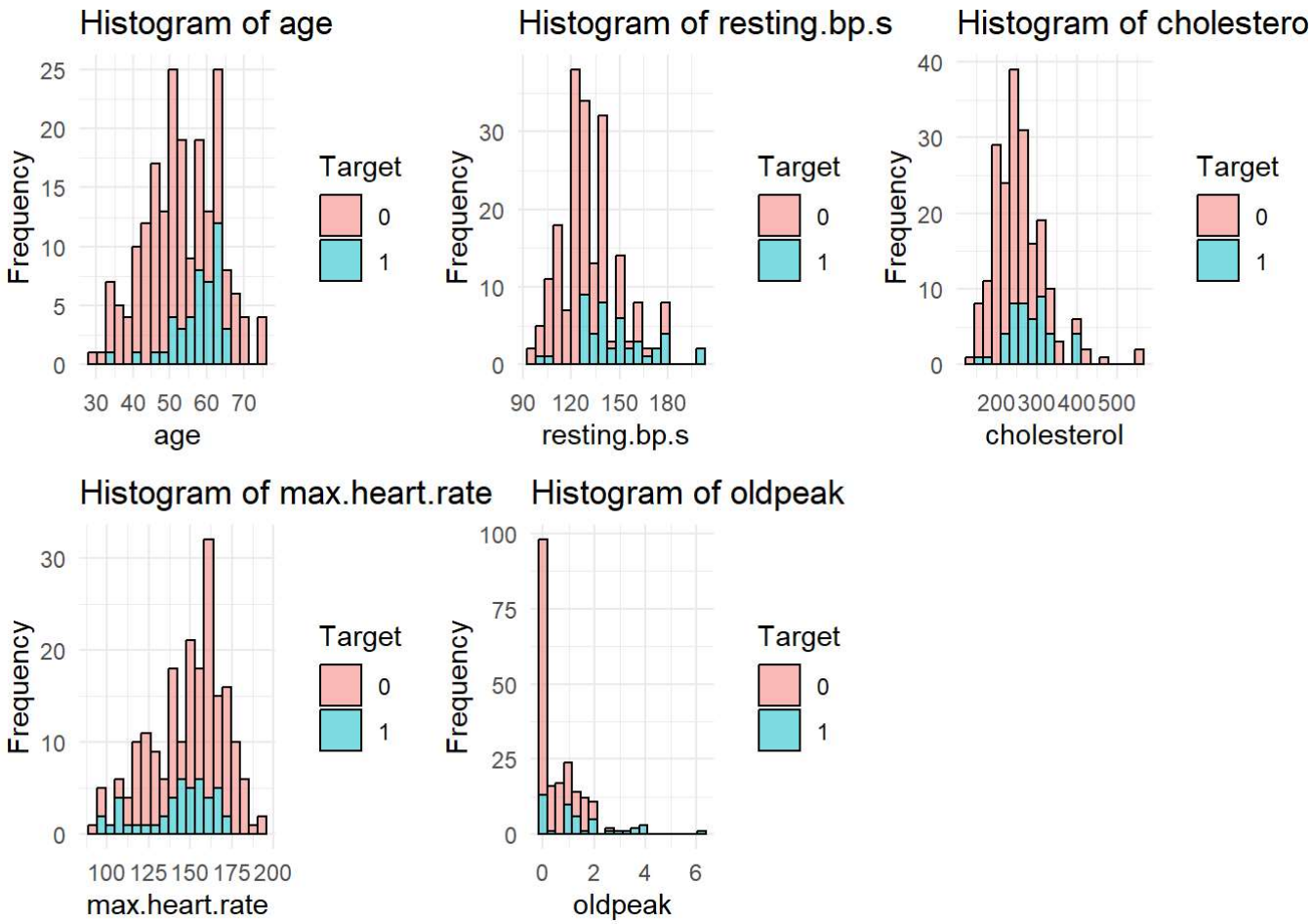

```
#boxplots for continuous variables
boxplots_continuous <- lapply(names(continuous_vars), function(var) {
  ggplot(combined_data, aes(x = target, y = !!as.name(var), fill = target)) +
    geom_boxplot() +
    labs(x = "Target", y = var) +
    ggtitle(paste("Boxplot of", var)) +
    theme_minimal()
})

# arrange histograms and boxplots in a grid
grid.arrange(grobs = boxplots_continuous, ncol = 3)
```


Boxplot of age
age
70
60
50
40
30
0
1
Target
target
0
1
Boxplot of resting.bp.s
resting.bp.s
180
150
120
90
0
1
Target
target
0
1
Boxplot of cholesterol
cholesterol
600
400
200
0
1
Target
target
0
1

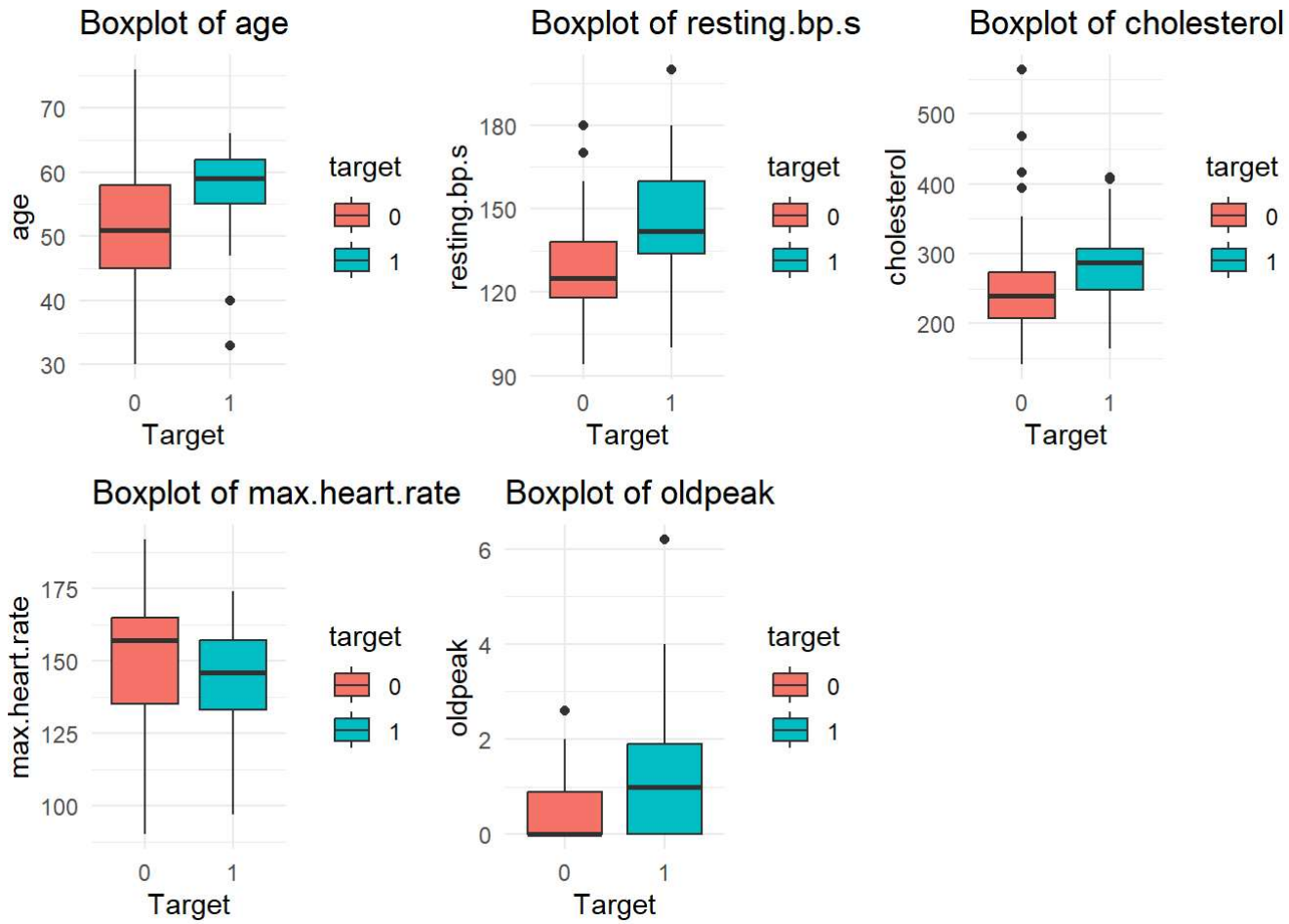

Boxplot of max.heart.rate
max.heart.rate
200
150
100
0
1
Target
target
0
1
Boxplot of oldpeak
oldpeak
4
2
0
0
1
Target
target
0
1

Females

```
target_0_data_f <- train_data_f[train_data_f$target == 0, ]
target_1_data_f <- train_data_f[train_data_f$target == 1, ]
# combine data for both target values and add a new column for target labels
combined_data <- rbind(
  transform(target_0_data_f, target = "0"),
  transform(target_1_data_f, target = "1")
)
continuous_vars <- train_data_f[, c('age', 'resting.bp.s', 'cholesterol', 'max.heart.rate', 'old
peak')]
# list to store histograms
histograms_continuous <- lapply(names(continuous_vars), function(var) {

  # histogram for both target values
  hist_combined <- ggplot(combined_data, aes_string(x = var, fill = "target")) +
    geom_histogram(color = "black", bins = 20, alpha = 0.5) +
    labs(x = var, y = "Frequency", fill = "Target") +
    ggtitle(paste("Histogram of", var)) +
    theme_minimal()

  # return histogram
  hist_combined
})

# arrange histograms in a grid
grid.arrange(grobs = histograms_continuous, ncol = 3)
```




```
boxplots_continuous <- lapply(names(continuous_vars), function(var) {
  ggplot(combined_data, aes(x = target, y = !!as.name(var), fill = target)) +
    geom_boxplot() +
    labs(x = "Target", y = var) +
    ggtitle(paste("Boxplot of", var)) +
    theme_minimal()
})

# arrange histograms and boxplots in a grid
grid.arrange(grobs = boxplots_continuous, ncol = 3)
```



Some Observations from the Numaric data(Males)

Age:

- The histogram of the `age` variable shows that heart disease appears more frequently in older age groups. This distribution highlights age as a significant risk factor.

Max Heart Rate (max.heart.rate):

- Interestingly, the histogram for `max.heart.rate` suggests that lower maximum heart rates are associated with a higher occurrence of heart disease.

Oldpeak (ST depression induced by exercise relative to rest):

- The analysis reveals that increased values of `oldpeak` are visibly associated with the presence of heart disease. on average, the oldpeak is higher in people who have heart disease.

Others:

-We did not notice any huge difference in other measurements between the people how have heart disease and who do not.

Some Observations from the Numeric Data (Females)

- **Age**:
    - Age is an important predictor for both genders. In females, the gap between those who have heart disease and those who do not is more pronounced, highlighting age as a crucial factor in diagnosing heart disease in women.
- **Resting Blood Pressure**:

    - Unlike in males, where resting blood pressure does not show a direct effect on heart disease, higher levels are significantly correlated with the presence of heart disease in females. This indicates that resting blood pressure is a more critical diagnostic indicator in women.
- **Cholesterol**:
    - Cholesterol levels do not show significant diagnostic importance in males but are more critical in females. Higher cholesterol levels are associated with an increased risk of heart disease in women.
- **Max Heart Rate**:
    - Lower maximum heart rates are strongly correlated with heart disease in females, more so than in males.
- **Oldpeak**:
    - Increased oldpeak values are closely linked to heart disease. This trend is crucial for diagnosis in both genders, with females showing a slightly higher average oldpeak when heart disease is present.

Reason for Choosing Boxplots and Histogram

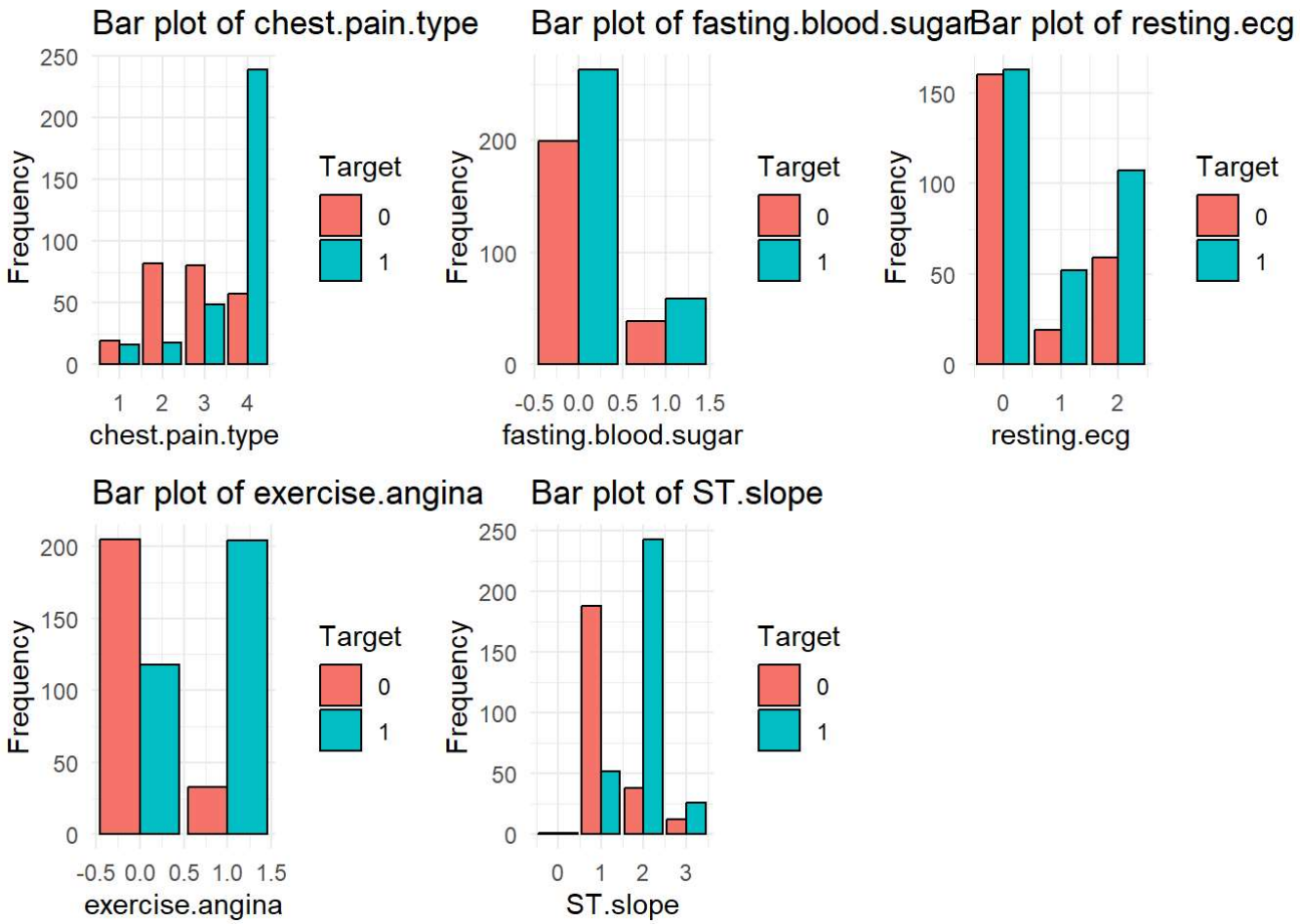
-**Histogram**: It helps us to the the difference in the distribution between the two classes, so we can perdict which predictors might be a strong indecator for heart disease.

-**Boxplots**:In addition to showing the distribution like the histogram, it also highlights outliers in the data and any skewness that may exist. Moreover, it displays the median, allowing us to understand where most of the data points lie

Analysis of Discrete Variables

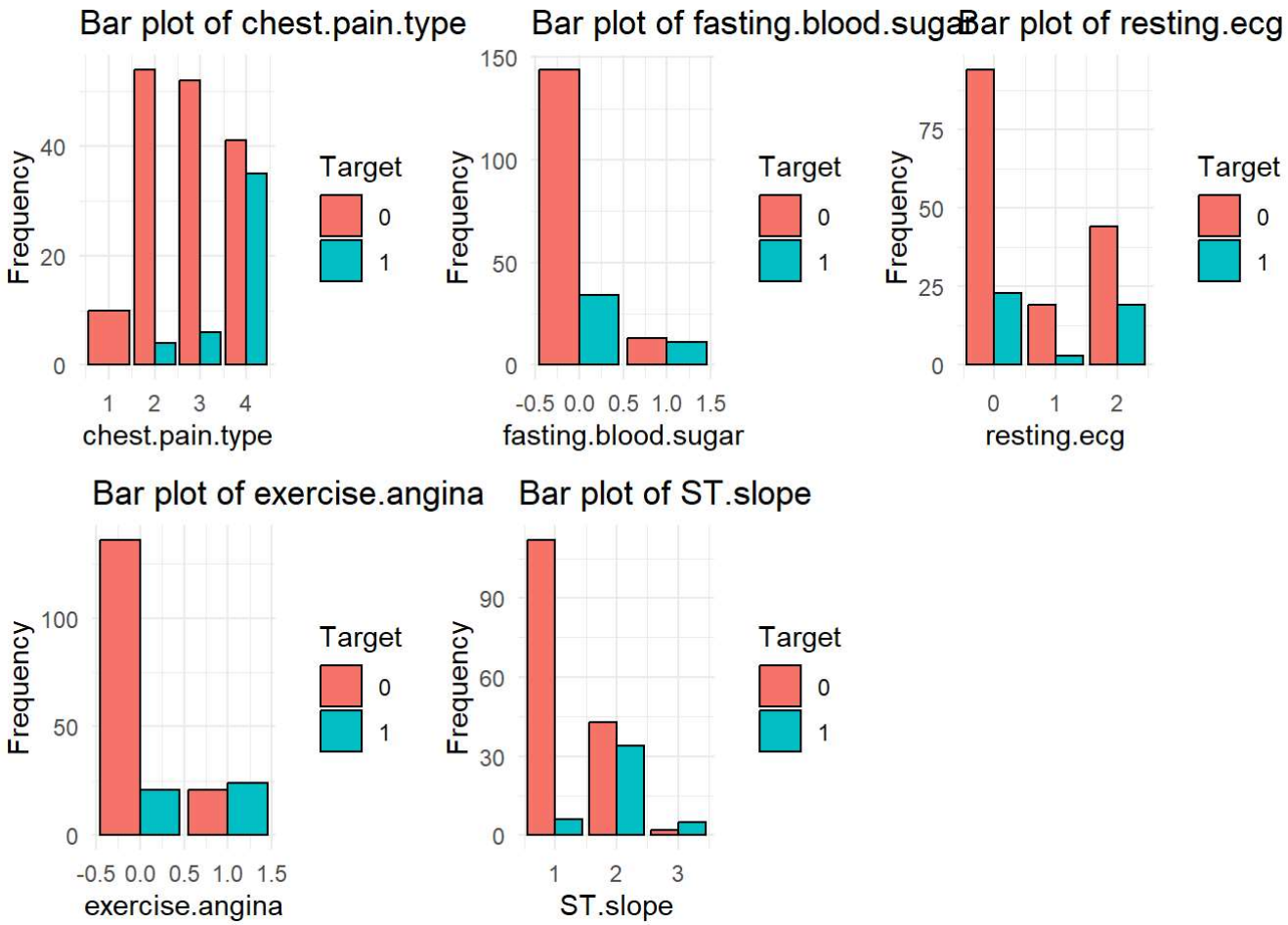
Males

```
# combine data for both target values
combined_data_m <- rbind(
  transform(target_0_data_m, target = "0"),
  transform(target_1_data_m, target = "1")
)
discrete_vars_m <- train_data_m[, c('chest.pain.type', 'fasting.blood.sugar', 'resting.ecg', 'ex
ercise.angina','ST.slope')]
# list to store bar plots
barplots_discrete_m <- lapply(names(discrete_vars_m), function(var) {
  ggplot(combined_data_m, aes_string(x = var, fill = "target")) +
    geom_bar(position = "dodge", color = "black") +
    labs(x = var, y = "Frequency", fill = "Target") +
    ggtitle(paste("Bar plot of", var)) +
    theme_minimal()
})

# arrange bar plots in a grid
grid.arrange(grobs = barplots_discrete_m, ncol = 3)
```



Females

```
# combine data for both target values
combined_data_f <- rbind(
  transform(target_0_data_f, target = "0"),
  transform(target_1_data_f, target = "1")
)
discrete_vars_f <- train_data_f[, c('chest.pain.type', 'fasting.blood.sugar', 'resting.ecg', 'ex
ercise.angina','ST.slope')]
# list to store bar plots
barplots_discrete_f <- lapply(names(discrete_vars_f), function(var) {
  ggplot(combined_data_f, aes_string(x = var, fill = "target")) +
    geom_bar(position = "dodge", color = "black") +
    labs(x = var, y = "Frequency", fill = "Target") +
    ggtitle(paste("Bar plot of", var)) +
    theme_minimal()
})

# arrange bar plots in a grid
grid.arrange(grobs = barplots_discrete_f, ncol = 3)
```



Some Observations from the discrete data(Males)

Chest Pain Type:

- The distribution of chest pain types shows varying frequencies of heart disease occurrence across different types of chest pain. Typically, certain types of pain, like asymptomatic, may be more indicative of heart disease than others like typical angina.

Exercise Angina:

- Exercise-induced angina is another critical factor. The plot showing its relationship with heart disease can highlight how symptoms during exercise relate to underlying heart conditions (when Exercise Angina="yes or 1").

ST Slope

- The slope of the peak exercise ST segment in an ECG can indicate heart health, with different slopes (upsloping, flat, downsloping) . The bar plot shows that the flat slop (ST.slop=2) can show an important indication for having heart disease.

Some Observations from the discrete data(Females)

- **Chest Pain Type**:
    - asymptomatic type is significant indicators of heart disease in females, a more focused predictive pattern compared to males where a broader range of pain types are involved.
- **ST Slope**:

    - While a flat ST slope is associated with heart disease in females, its predictive importance is somewhat lower compared to males, where it serves as a more critical diagnostic indicator.
- **Exercise Angina**:
    - Although occurring less frequently in females, exercise-induced angina remains a significant indicator of heart disease, paralleling its predictive value in males.
- **Most of the discrete variables seems to have a distinct impact on the target outcome especially for males, which suggests that they are useful predictors in the dataset. Their inclusion in predictive modeling would likely improve the model's performance.**

Reason for Choosing Barplots

They provide a good representation of data when dealing with discrete variables. Bar plots can show distinct categories, such as the type of chest pain that is most common among individuals with heart disease.

Summary

- **Numeric Data**: Shows stronger correlations with heart disease in females, especially for predictors like resting blood pressure and cholesterol, highlighting more pronounced physiological markers in females than in males.
- **Discrete Data**: Exhibits more variation and diagnostic relevance in males, particularly with variables like chest pain type and ST slope, indicating a broader range of symptomatic expressions in males compared to females.
- In general, the features for detecting heart disease is somewhat different between males and females.

Corelation between the predictors

We will use a heatmap because it can show the dependencies between the features in the data.

Males

```
# select only the required columns
selected_data_m <- train_data_m[, c('age', 'resting.bp.s', 'cholesterol', 'max.heart.rate', 'old
peak', 'target')]

# convert target to numeric if it's a factor
selected_data_m$target <- as.numeric(selected_data_m$target)

# calculate the correlation matrix
correlation_matrix_m <- cor(selected_data_m, use = "complete.obs")

#plot it
corrplot(correlation_matrix_m, method = "color", addCoef.col = "black", tl.col = "black", tl.srt
= 0.5)
```

Females

```
#select only the required columns
selected_data_f <- train_data_f[, c('age', 'resting.bp.s', 'cholesterol', 'max.heart.rate', 'old
peak', 'target')]

#convert target to numeric if it's a factor
selected_data_f$target <- as.numeric(selected_data_f$target)

correlation_matrix <- cor(selected_data_f, use = "complete.obs")

#plot it
corrplot(correlation_matrix, method = "color", addCoef.col = "black", tl.col = "black", tl.srt =
0.5)
```

Insights from the Correlation Matrices

- The correlation matrices for both males and females demonstrate direct relationships between various predictors and the target variable (indicating the presence of heart disease).
- However, the matrices also highlight correlations among the predictors themselves, suggesting multicollinearity. This multicollinearity can complicate predictive models since predictors are not independent of each other. For instance, higher cholesterol might correlate with increased blood pressure, and both might increase with age.
- Addressing these interdependencies requires sophisticated statistical learning methods capable of capturing these complex relationships without oversimplifying the model or losing important information.

Choosing the Statistical learning methods

The dataset for predicting heart disease in males includes several complexities:

- **Size of the Dataset**: Both datasets is relatively small, which increases the risk of overfitting.
- **Presence of Outliers**: Numerous outliers in numerical variables add complexity to the predictive modeling process (Refer to the box plots).
- **Categorical Variables**: There are multiple categorical predictors that show clear associations with heart disease outcomes In both datasets.
- **Numerical Variables**: These do not provide clear class separations and exhibit significant correlations with the target variable and among themselves In both datasets.

Chosen Techniques: SVM and Random Forest

Support Vector Machines (SVM)

Complex Relationship Modeling

SVM is capable of modeling complex, non-linear interactions through kernel functions, making it particularly effective for datasets where class boundaries are not linearly separable. We use the Radial Basis Function (RBF) kernel to handle the non-linearity in the data.

Robustness to Overfitting

By tuning the regularization parameter (`C`), SVM can balance between margin maximization and loss minimization, crucial for preventing overfitting in smaller datasets.

Random Forest

- **Robustness to Overfitting**: Valuable for datasets with complex and non-linear relationships, helping to prevent model overfitting while capturing intricate patterns in the data.
- **Handling of Mixed Data Types**: Random Forest handles both categorical and numerical data efficiently, making it ideal for datasets with diverse predictors.
- **Feature Importance**: Provides insights on which features are most impactful in predicting heart disease, essential for interpretative analysis and guiding further research.

Cross Validation

We will use 10-fold cross-validation to determine all the required parameters for these models. Additionally, this approach will allow us to estimate their performance accurately so we can choose the final model for each dataset (males and females).

Predictive analysis and results

Prepare the folds for the CV

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

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

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

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

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

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

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
```

```
library(rsample)
```

```
## Warning: package 'rsample' was built under R version 4.3.3
```

```
set.seed(1)
folds_f <- vfold_cv(train_data_f, v = 10)
folds_f
```

```
## #  10-fold cross-validation
## # A tibble: 10 × 2
##    splits           id
##    <list>           <chr>
##  1 <split [181/21]> Fold01
##  2 <split [181/21]> Fold02
##  3 <split [182/20]> Fold03
##  4 <split [182/20]> Fold04
##  5 <split [182/20]> Fold05
##  6 <split [182/20]> Fold06
##  7 <split [182/20]> Fold07
##  8 <split [182/20]> Fold08
##  9 <split [182/20]> Fold09
## 10 <split [182/20]> Fold10
```

```
set.seed(1)
folds <- vfold_cv(train_data_m, v = 10)
folds
```

```
## #  10-fold cross-validation
## # A tibble: 10 × 2
##    splits           id
##    <list>           <chr>
##  1 <split [504/56]> Fold01
##  2 <split [504/56]> Fold02
##  3 <split [504/56]> Fold03
##  4 <split [504/56]> Fold04
##  5 <split [504/56]> Fold05
##  6 <split [504/56]> Fold06
##  7 <split [504/56]> Fold07
##  8 <split [504/56]> Fold08
##  9 <split [504/56]> Fold09
## 10 <split [504/56]> Fold10
```

10-Folds CV for SVM and RF

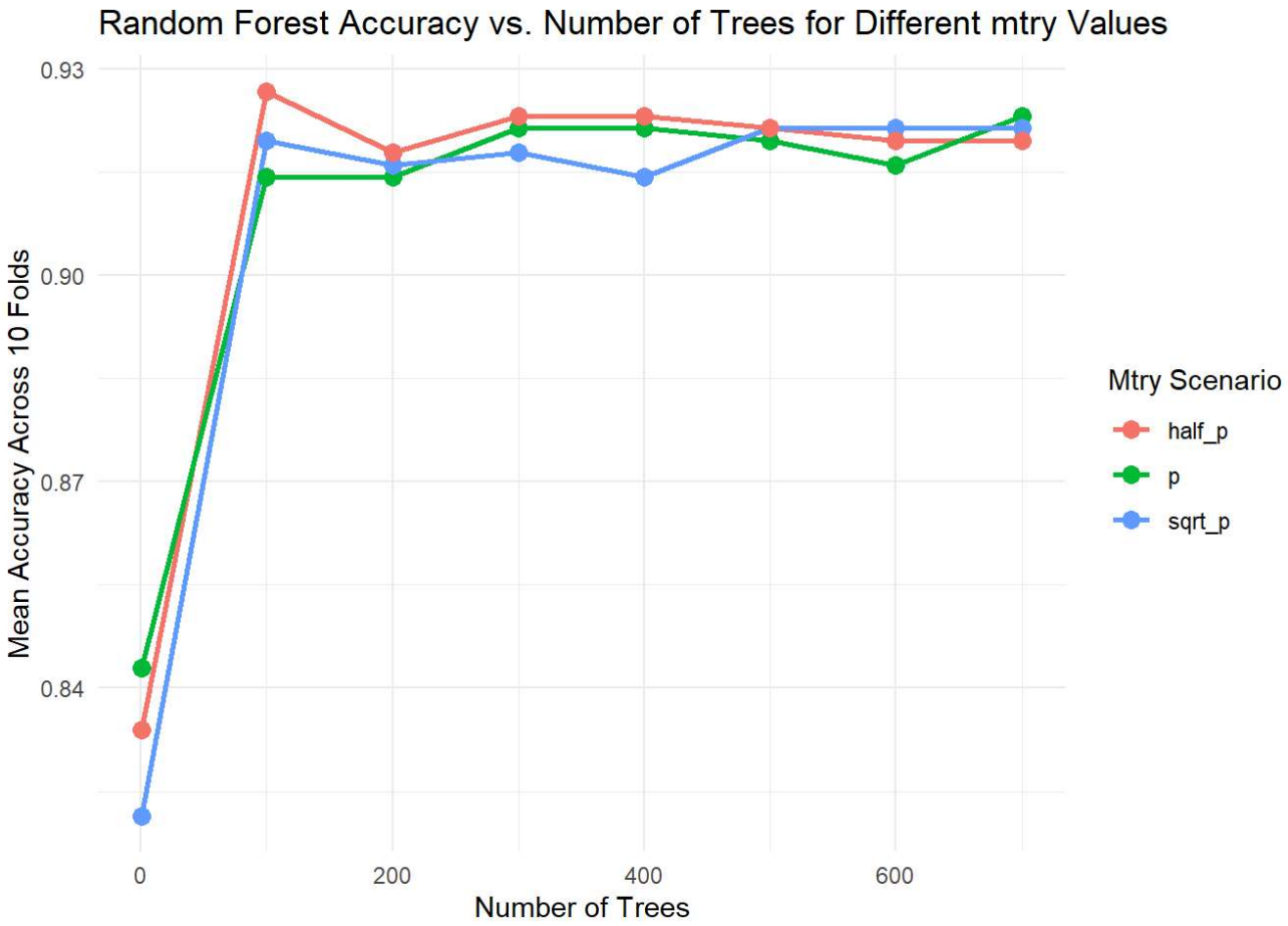
RF(Males Dataset)

```r
# number of folds
k <- 10
train_data_m$target <- as.factor(train_data_m$target)

# number of predictors in the model, excluding the target and 'sex'
num_predictors <- ncol(train_data_m) - 2  # Adjust if more columns are excluded

# list of tree numbers to try
tree_numbers <- c(1, 100, 200, 300, 400, 500, 600, 700)

#initialize data frames to store results for each mtry scenario
results_p <- results_sqrt_p <- results_half_p <- numeric(length(tree_numbers))
data_frames <- list(p = numeric(length(tree_numbers)),
                    sqrt_p = numeric(length(tree_numbers)),
                    half_p = numeric(length(tree_numbers)))

names(data_frames) <- c("p", "sqrt_p", "half_p")
mtry_values <- c(p = num_predictors, sqrt_p = sqrt(num_predictors), half_p = num_predictors / 2)
results_rf_m <- data.frame(ntree = integer(), mtry = integer(), mean_accuracy = numeric())
# Loop through each mtry scenario
for (scenario in names(mtry_values)) {
    mtry_val <- round(mtry_values[[scenario]])

    # Loop through each number of trees
    for (idx in seq_along(tree_numbers)) {
        ntree <- tree_numbers[idx]
        rf_accuracies <- numeric(k)

        # Perform k-fold cross-validation
        for (i in 1:k) {
            # Extract the training and validation sets for this fold
            fold_train <- train_data_m[folds$splits[[i]]$in_id, ]
            fold_valid <- train_data_m[-folds$splits[[i]]$in_id, ]

            # Train the Random Forest model on the training set for this fold
            rf_model_fold <- randomForest(target ~ . - sex, data = fold_train, ntree = ntree, mt
ry = mtry_val)

            # Make predictions on the validation set for this fold
            predictions_rf <- predict(rf_model_fold, newdata = fold_valid)

            # Calculate accuracy for this fold
            correct_predictions_RF <- sum(predictions_rf == fold_valid$target)
            total_predictions_rf <- length(predictions_rf)
            accuracy <- correct_predictions_RF / total_predictions_rf

            # Store the accuracy for this fold
            rf_accuracies[i] <- accuracy
        }

        # Calculate the mean accuracy across all folds and store in the appropriate scenario
        mean_accuracy <- mean(rf_accuracies)
```

```
        results_rf_m <- rbind(results_rf_m, data.frame(ntree = ntree, mtry = mtry_val, mean_accu
racy = mean_accuracy))
        data_frames[[scenario]][idx] <- mean_accuracy
    }
}

# Convert lists to data frames for plotting
accuracy_data <- data.frame(ntree = rep(tree_numbers, times = 3),
                            mean_accuracy = unlist(data_frames),
                            scenario = rep(names(data_frames), each = length(tree_numbers)))

# Plotting the results
accuracy_plot <- ggplot(accuracy_data, aes(x = ntree, y = mean_accuracy, color = scenario)) +
  geom_line(size = 1) +  # Line graph
  geom_point(size = 3) +  # Points
  labs(title = "Random Forest Accuracy vs. Number of Trees for Different mtry Values",
       x = "Number of Trees",
       y = "Mean Accuracy Across 10 Folds",
       color = "Mtry Scenario") +
  theme_minimal()  # Minimal theme
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# Display the plot
print(accuracy_plot)
```




```
optimal_results_rf_m <- results_rf_m[which.max(results_rf_m$mean_accuracy), ]
```

SVM(Males Dataset)

NOTE: The Used library e1071 and the function perform 10-Folds Cross_Validation to get the best parameters and the accuracy.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:rsample':
##
##     permutations
```

```
#tune the SVM model parameters
  tune_results <- e1071::tune(svm,target ~ ., data = train_data_m, kernel = "radial",
                                  ranges = list(cost = 10^(-1:3), gamma = c(0.5, 1, 2, 3, 4)))

  #extract the best parameters
  best_parameters <- tune_results$best.parameter
  print(best_parameters)
```

```
##   cost gamma
## 2    1   0.5
```

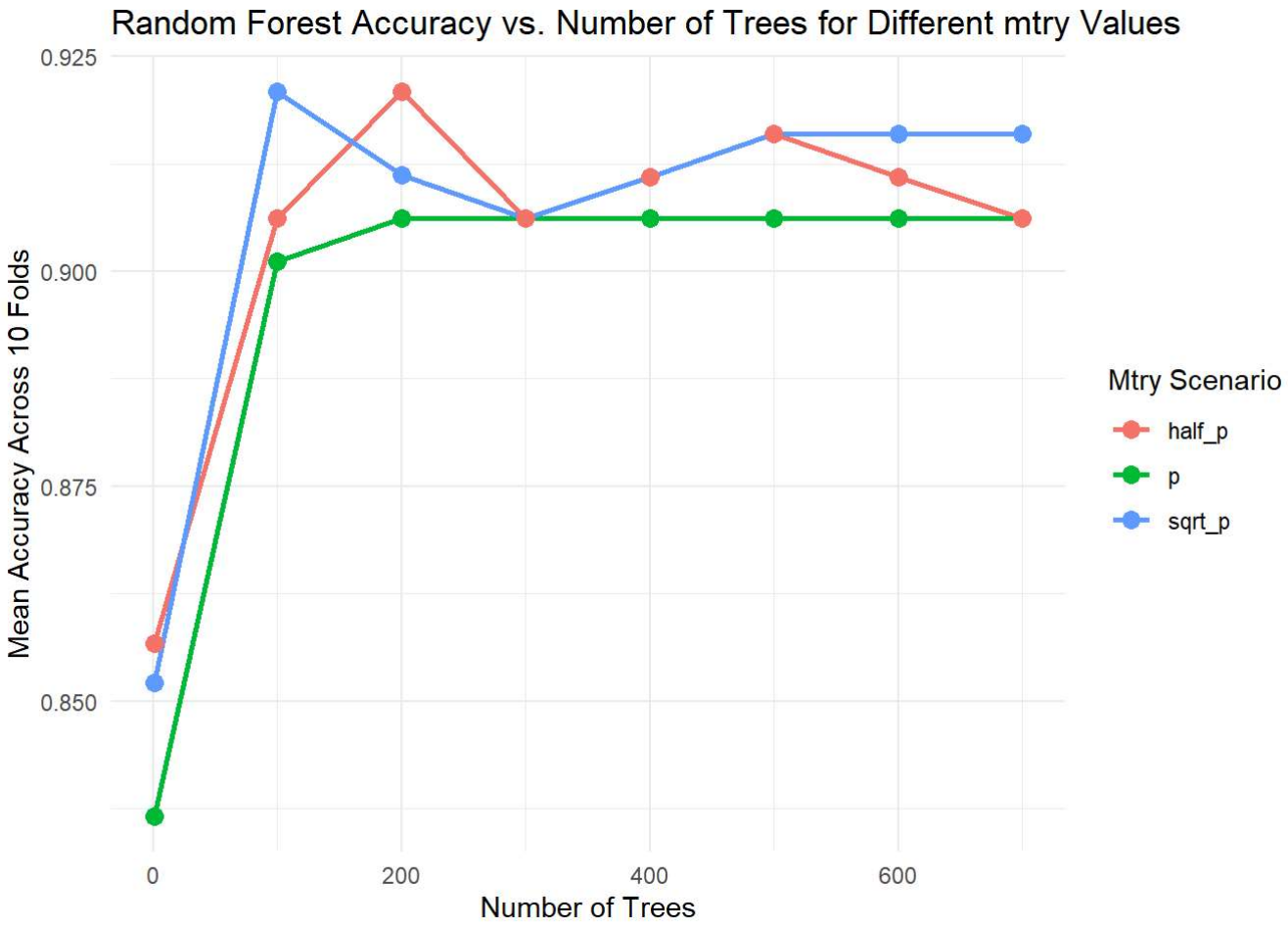
```
tune_summary <- summary(tune_results)

best_accuracy <- 1 - tune_summary$best.performance
print(paste("Best Accuracy:", best_accuracy))
```

```
## [1] "Best Accuracy: 0.883928571428571"
```

RF(Females Dataset)

```
# number of folds
k <- 10
train_data_f$target <- as.factor(train_data_f$target)

# Number of predictors in the model, excluding the target and 'sex'
num_predictors <- ncol(train_data_f) - 2  # Adjust if more columns are excluded

#list of tree numbers to try
tree_numbers <- c(1, 100, 200, 300, 400, 500, 600, 700)

#initialize data frames to store results for each mtry scenario
results_p <- results_sqrt_p <- results_half_p <- numeric(length(tree_numbers))
data_frames <- list(p = numeric(length(tree_numbers)),
                    sqrt_p = numeric(length(tree_numbers)),
                    half_p = numeric(length(tree_numbers)))

names(data_frames) <- c("p", "sqrt_p", "half_p")
mtry_values <- c(p = num_predictors, sqrt_p = sqrt(num_predictors), half_p = num_predictors / 2)
results_rf_f <- data.frame(ntree = integer(), mtry = integer(), mean_accuracy = numeric())
# Loop through each mtry scenario
for (scenario in names(mtry_values)) {
    mtry_val <- round(mtry_values[[scenario]])

    # Loop through each number of trees
    for (idx in seq_along(tree_numbers)) {
        ntree <- tree_numbers[idx]
        rf_accuracies <- numeric(k)

        # Perform k-fold cross-validation
        for (i in 1:k) {
            # Extract the training and validation sets for this fold
            fold_train <- train_data_f[folds_f$splits[[i]]$in_id, ]
            fold_valid <- train_data_f[-folds_f$splits[[i]]$in_id, ]

            # Train the Random Forest model on the training set for this fold
            rf_model_fold <- randomForest(target ~ . - sex, data = fold_train, ntree = ntree, mt
ry = mtry_val)

            # Make predictions on the validation set for this fold
            predictions_rf_f <- predict(rf_model_fold, newdata = fold_valid)

            # Calculate accuracy for this fold
            correct_predictions_rf_f <- sum(predictions_rf_f == fold_valid$target)
            total_predictions <- length(predictions_rf_f)
            accuracy <- correct_predictions_rf_f / total_predictions

            # Store the accuracy for this fold
            rf_accuracies[i] <- accuracy
        }

        # Calculate the mean accuracy across all folds and store in the appropriate scenario
        mean_accuracy <- mean(rf_accuracies)
```

```
        results_rf_f <- rbind(results_rf_f, data.frame(ntree = ntree, mtry = mtry_val, mean_accu
racy = mean_accuracy))
        data_frames[[scenario]][idx] <- mean_accuracy
    }
}

# Convert lists to data frames for plotting
accuracy_data <- data.frame(ntree = rep(tree_numbers, times = 3),
                            mean_accuracy = unlist(data_frames),
                            scenario = rep(names(data_frames), each = length(tree_numbers)))

# Plotting the results
accuracy_plot <- ggplot(accuracy_data, aes(x = ntree, y = mean_accuracy, color = scenario)) +
  geom_line(size = 1) +  # Line graph
  geom_point(size = 3) +  # Points
  labs(title = "Random Forest Accuracy vs. Number of Trees for Different mtry Values",
       x = "Number of Trees",
       y = "Mean Accuracy Across 10 Folds",
       color = "Mtry Scenario") +
  theme_minimal()  # Minimal theme

# Display the plot
print(accuracy_plot)
```




```
optimal_results_rf_f <- results_rf_f[which.max(results_rf_f$mean_accuracy), ]
```

SVM(Females Dataset)

NOTE: The Used library e1071 and the function perform 10-Folds Cross_Validation to get the best parameters and the accuracy.

```
  #tune the SVM model parameters
  tune_results <- e1071::tune(svm,target ~ ., data = train_data_f, kernel = "radial",
                                ranges = list(cost = 10^(-1:3), gamma = c(0.5, 1, 2, 3, 4)))

  #extract the best parameters
  best_parameters_F <- tune_results$best.parameter
  print(best_parameters_F)
```

```
##   cost gamma
## 8   10     1
```

```
tune_summary <- summary(tune_results)

best_accuracy <- 1 - tune_summary$best.performance
print(paste("Best Accuracy:", best_accuracy))
```

```
## [1] "Best Accuracy: 0.891428571428571"
```

Overview of Cross-Validation Performance:

- **Parameters** The cross-validation process helped us to determine all the required parameters for the SVM and Random Forest models that we are going to use to find the test accuracy in the next section.
- **Model Comparison:** For both genders, cross-validation revealed that both Random Forest and SVM performed robustly, where there accurcies were above 88% . However, Random Forest slightly outperformed SVM for both genders so we will select them as the final models with the specfied parameters (ntree and mtyr) in the CV.

Performance on Test data (Final models)

RF (Males)

```
rf_model_male_test <- randomForest(target ~ . - sex, data = train_data_m, ntree = optimal_result
s_rf_m$ntree, mtry = optimal_results_rf_m$mtry)

            # predicit on the validation set for this fold
            predictions_test_rf <- predict(rf_model_male_test, newdata = test_data_m)

            #calculate accuracy for this fold
            correct_predictions_RF_m_test <- sum(predictions_test_rf == test_data_m$target)
            total_predictions_rf_test <- length(predictions_test_rf)
            accuracy <- correct_predictions_RF_m_test / total_predictions_rf_test
            print (accuracy)
```

```
## [1] 0.9304813
```

RF (Females)

```
rf_model_female_test <- randomForest(target ~ . - sex, data = train_data_f, ntree = optimal_resu
lts_rf_f$ntree, mtry = optimal_results_rf_f$mtry)

            #predictions on the validation set for this fold
            predictions_test_rf_f <- predict(rf_model_female_test, newdata = test_data_f)

            #calculate accuracy for this fold
            correct_predictions_RF_f_test <- sum(predictions_test_rf_f == test_data_f$target)
            total_predictions_rf_test_f <- length(predictions_test_rf_f)
            accuracy <- correct_predictions_RF_f_test / total_predictions_rf_test_f
            print (accuracy)
```

```
## [1] 0.8823529
```

results Discussion

Male Dataset results

The Random Forest model achieved a test accuracy of approximately 92% when applied to the male test dataset.

Female Dataset results

For females, the Random Forest model achieved a test accuracy of about 90%.

Discussion

We had a dataset then we decided to divide it to avoid the imbalance between males and females and prevent bias towards the male subset because it is larger. After exploring the data, we decided to apply RF and SVM, and they both showed good performance using cross-validation (CV). According to the CV results, RF was better for both subsets, so we chose it. When we actually applied the model to unseen data, it performed well, which emphasizes the practicality of our procedure.

Conclusion

The analysis demonstrates that both Random Forest and SVM are capable statistical learning methods for predicting heart disease from the dataset used. The use of cross-validation ensured that the models are both robust and reliable, and it showed that RF is better approach for both datasets. In addition, the test results confirmed their ability to generalize well to new data. Moreover, the separation of the dataset into male and female subsets at the beginning proved very successful. This approach allowed us to discover that each subset has distinct characteristics and should be treated and trained separately to achieve better accuracies. This method ensures that no gender imbalances the data, preventing bias towards males, who constitute the larger portion of the dataset.

Scope and Generalizability

The models developed show good generalizability to new data, indicating that the statistical learning methods applied can effectively handle the complexity of the heart disease prediction problem. However, the generalizability of the findings might still be limited by the characteristics of the dataset, which primarily includes data from specific populations. Therefore, applying these models to broader populations may require adjustments and revalidation.

Limitations and Possibilities for Improvement

Limitations

- **Sample Size**: One of the primary limitations is the sample size, particularly in the female subset, which may not fully capture the diversity and complexity of heart disease presentations in broader populations.
- **Model Complexity**: While SVM and Random Forest provide robust predictions, their complexity can make interpretation difficult, especially for clinical applications where understanding the decision-making process is crucial.

Recommendations for Future Work

- **Data Collection**: To decrease the limitations posed by the current dataset size, future studies should focus on collecting more data across diverse demographics. This expansion would help improve the robustness and applicability of the predictive models.
- **Inclusion of Additional Predictors**: Integrating more predictors, such as lifestyle factors or genetic information, could enhance the models' accuracy.
- **Use of Simpler Models for Interpretability**: Considering the applicability of less complex statistical learning classifiers, such as logistic regression because it could enhance interpretability. Such models provide clearer insights into how each predictor influences the risk of heart disease, and this valuable for clinical decision.

By addressing these areas, future research can enhance the predictive accuracy, reliability, and applicability of models designed to predict heart disease risk.