Data Science Capstone Project: Predicting Heart Disease

Catherine Edis

March 2022

Contents

1 Introduction **1**
- 1.1 Purpose . . . 1
- 1.2 Heart Disease Data . . . 1
- 1.3 Goal of the Project . . . 2
- 1.4 Key Steps Performed . . . 2
- 1.5 Acknowledgments . . . 3

2 Analysis **3**
- 2.1 Preparation . . . 3
- 2.2 Initial Examination of the Data . . . 4
- 2.3 Reformatting the Dataset . . . 5
- 2.4 Further Exploration of the Data . . . 7
- 2.5 Modelling Approach . . . 16

3 Results **18**

4 Conclusion **19**

---

1 Introduction

1.1 Purpose

This project was undertaken to fulfill the requirements of the "Data Science: Capstone" course, which is part of the Professional Certificate in Data Science program offered by Harvard University through edX. The purpose of the project was to solve a chosen problem using a publicly-available dataset and the machine learning techniques explained in the Professional Certificate program. The machine learning techniques used must include more than just standard linear regression.

1.2 Heart Disease Data

The project used a dataset of heart disease observations that was originally collected by the Cleveland Clinic Foundation (principal investigator: Robert Detrano, M.D., Ph.D.). The dataset was published on the UC Irvine Machine Learning Repository by David Aha in 1988. It is available at: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

The *processed.cleveland.data* dataset was used for this project, because it was relatively clean and complete. (It contains relatively few missing or "dummy" values compared to the other datasets published at the above site.)

As described in the *heart-disease.names* file provided with the dataset, the dataset includes 303 observations, with 14 attributes (described below). Each observation relates to an individual patient, and includes physiological information collected while they underwent supervised tests, including a thallium stress test (also known as a nuclear stress test). A thallium stress test is a nuclear imaging test that shows how well blood flows around the patient's heart while they are exercising or at rest.[1]

Attribute	Description	Type
age	Age of the patient, in years	Continuous
sex	Sex of the patient	Categorical
cp	Type of chest pain	Categorical
trestbps	Resting blood pressure	Continuous
chol	Serum cholesterol	Continuous
fbs	Fasting blood sugar	Categorical
restecg	Resting electrocardiographic (ECG) results	Categorical
thalach	Maximum heart rate achieved	Continuous
exang	Exercise-induced angina	Categorical
oldpeak	ST depression induced by exercise relative to rest	Continuous
slope	Slope of peak exercise ST segment	Categorical
ca	Number of major vessels colored by fluoroscopy	Continuous
thal	Thallium stress test result	Categorical
target	Heart disease diagnosis	Categorical

The attributes are discussed in more detail in the *Analysis* section below.

1.3 Goal of the Project

The aim of this project was to explore a number of different models that might be used to predict the presence of heart disease, and identify the model that produces the most accurate predictions, using the Cleveland heart disease dataset.

1.4 Key Steps Performed

The following key steps were performed.

1. Load the dataset.
2. Initial examination of the structure and content of the dataset.
3. Perform some basic checks on the quality of the data (for example, check for any missing values).
4. Where required, reformat the data to make it easier to work with.
5. Explore the dataset further, to better understand the nature of the data and identify attributes that could be used as potential predictors in the modelling.
6. Set up the objects required for modelling, including a training and test set generated from the original dataset.
7. Use the training set to develop and train various prediction models, and run them against the test set.
8. Compare the accuracy of the models when run against the test set, and identify the one that produces the most accurate results.

These steps and the results that were produced are described in more detail in the following sections.

[1] https://www.healthline.com/health/thallium-stress-test

1.5 Acknowledgments

The author wishes to acknowledge the following:

- for the collection of the original data: Cleveland Clinic Foundation, Principal Investigator Robert Detrano, M.D., Ph.D., et al.
- for the UC Irvine Machine Learning Repository: Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- for the Heart Disease dataset published on the UC Irvine Machine Learning Repository: David Aha.

2 Analysis

2.1 Preparation

Before beginning the analysis, any required libraries were installed and loaded.

```
##########################################################################################
# Install any packages required.
##########################################################################################

if(!require(tidyverse))
  install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret))
  install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table))
  install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(scales))
  install.packages("scales", repos = "http://cran.us.r-project.org")
if(!require(lubridate))
  install.packages("lubridate", repos = "http://cran.us.r-project.org")

library(tidyverse)
library(caret)
library(data.table)
library(scales)
library(lubridate)
```

A copy of the processed Cleveland heart disease dataset was then loaded, and the *cleveland_heart_disease* set/table created for use in the analysis.

```
##########################################################################################
#
# Load the dataset.
#
# The original dataset is available in the UCI Machine Learning Repository, at:
# https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease
#
# This project uses a copy of the dataset, which was uploaded to a GitHub repository,
# along with the project report and the associated scripts.
#
##########################################################################################

# Read in the data from the "processed.cleveland.data" file.
```

```
cleveland_heart_disease <- read.csv("processed.cleveland.data",
                                    header = FALSE, sep = ",", stringsAsFactors = FALSE)

# The data file does not contain a header row, so the attribute/column names need to be
# set, using the information provided with the dataset in the "heart-disease.names" file.

column_names <- c("age", "sex", "cp",   "trestbps", "chol", "fbs", "restecg",
                  "thalach", "exang", "oldpeak", "slope", "ca", "thal", "target")
colnames(cleveland_heart_disease) <- column_names
```

2.2 Initial Examination of the Data

The dataset was examined to understand and confirm its structure, including the:

- class of the dataset;
- number of observations (rows);
- number of attributes (columns); and,
- name and class of each attribute.

```
# Examine the structure of the dataset.
str(cleveland_heart_disease)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : num  1 1 1 1 0 1 0 0 1 1 ...
##  $ cp      : num  1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : num  1 0 0 0 0 0 0 0 0 1 ...
##  $ restecg : num  2 2 2 0 2 0 2 0 2 2 ...
##  $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : num  0 1 1 0 0 0 0 1 0 1 ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ slope   : num  3 2 2 3 1 1 3 1 2 3 ...
##  $ ca      : chr  "0.0" "3.0" "2.0" "0.0" ...
##  $ thal    : chr  "6.0" "3.0" "7.0" "3.0" ...
##  $ target  : int  0 2 1 0 0 0 3 0 2 1 ...
```

The first few rows of the dataset were examined to view some examples of content.

```
# To see some example data, view the first few rows of the sets.
knitr::kable(head(cleveland_heart_disease, 10))
```

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	2
67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0
56	1	2	120	236	0	0	178	0	0.8	1	0.0	3.0	0
62	0	4	140	268	0	2	160	0	3.6	3	2.0	3.0	3
57	0	4	120	354	0	0	163	1	0.6	1	0.0	3.0	0

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	4	130	254	0	2	147	0	1.4	2	1.0	7.0	2
53	1	4	140	203	1	2	155	1	3.1	3	0.0	7.0	1

The dataset was then checked for missing values. The documentation that was supplied with the original dataset indicated that there were some missing values, and each of them was set to "?". The dataset was therefore checked for values of both "NA" and "?".

```
# Check for any missing ("NA") values in any column.
apply(cleveland_heart_disease, 2, function(x) any(is.na(x)))
```

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

```
# Check for any values containing a "?".
apply(cleveland_heart_disease, 2, function(x) any(x=="?"))
```

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

The initial examination confirmed that:

1. The dataset contained 303 observations and 14 variables.
2. Each row appears to contain an observation for one patient.
3. The dataset contains no "NA" values.
4. The dataset contains some "?" values (used to denote missing values) in the *ca* and *thal* columns.

2.3 Reformatting the Dataset

After the initial examination, the dataset was reformatted to make it easier to work with, as follows:

- Rows that contained "?" values were removed.
- The *target* attribute (the attribute the modelling attempted to predict), was converted to a binary value that simply indicated whether or not heart disease had been diagnosed in the patient.
- Columns that corresponded to categorical attributes were converted to factors.
- Labels were defined for factor levels, based on the information provided in the documentation that accompanied the original dataset.

```
#############################################################################################
# Reformat the data to make it easier to work with.
#############################################################################################

# Remove the rows with "?" values.
cleveland_heart_disease <- cleveland_heart_disease %>% filter(ca!="?" & thal !="?")

# Because the "ca" and "thal" columns contained "?", they were both loaded as type
# "character" initially. They can now be converted to numerics.
cleveland_heart_disease$ca <- as.numeric(cleveland_heart_disease$ca)
```

```r
cleveland_heart_disease$thal <- as.numeric(cleveland_heart_disease$thal)

# Convert the "target" attribute to a binary value where:
#    0 indicates no heart disease is present
#    1 indicates some heart disease is present
#
# Note: This is the attribute that the models will be trained to predict.

cleveland_heart_disease <- cleveland_heart_disease %>%
  mutate(target = ifelse(target >= 1, 1, 0))

# Several of the attributes in the dataset are categorical.
# Convert the categorical attributes from numeric to factor.
categorical_column_names <- c("sex", "cp", "fbs", "restecg", "exang", "slope", "thal",
                              "target")
cleveland_heart_disease[categorical_column_names] <-
  lapply(cleveland_heart_disease[categorical_column_names], factor)

# Define level names for the "sex" attribute, to make plotting easier.
levels(cleveland_heart_disease$sex)[levels(cleveland_heart_disease$sex)==0] <-
  "Female"
levels(cleveland_heart_disease$sex)[levels(cleveland_heart_disease$sex)==1] <-
  "Male"

# Define level names for the "cp" attribute, to make plotting easier.
levels(cleveland_heart_disease$cp)[levels(cleveland_heart_disease$cp)==1] <-
  "Typical Angina"
levels(cleveland_heart_disease$cp)[levels(cleveland_heart_disease$cp)==2] <-
  "Atypical Angina"
levels(cleveland_heart_disease$cp)[levels(cleveland_heart_disease$cp)==3] <-
  "Non-Anginal Pain"
levels(cleveland_heart_disease$cp)[levels(cleveland_heart_disease$cp)==4] <-
  "Asymptomatic"

# Define level names for the "fbs" attribute, to make plotting easier.
levels(cleveland_heart_disease$fbs)[levels(cleveland_heart_disease$fbs)==0] <-
  "<= 120 mg/dl"
levels(cleveland_heart_disease$fbs)[levels(cleveland_heart_disease$fbs)==1] <-
  "> 120 mg/dl"

# Define level names for the "restecg" attribute, to make plotting easier.
levels(cleveland_heart_disease$restecg)[levels(cleveland_heart_disease$restecg)==0] <-
  "Normal"
levels(cleveland_heart_disease$restecg)[levels(cleveland_heart_disease$restecg)==1] <-
  "ST-T Wave Abnormality"
levels(cleveland_heart_disease$restecg)[levels(cleveland_heart_disease$restecg)==2] <-
  "Left Ventricular Hypertrophy"

# Define level names for the "exang" attribute, to make plotting easier.
levels(cleveland_heart_disease$exang)[levels(cleveland_heart_disease$exang)==0] <-
  "No"
levels(cleveland_heart_disease$exang)[levels(cleveland_heart_disease$exang)==1] <-
  "Yes"
```

```
# Define level names for the "slope" attribute, to make plotting easier.
levels(cleveland_heart_disease$slope)[levels(cleveland_heart_disease$slope)==1] <-
  "Upsloping"
levels(cleveland_heart_disease$slope)[levels(cleveland_heart_disease$slope)==2] <-
  "Flat"
levels(cleveland_heart_disease$slope)[levels(cleveland_heart_disease$slope)==3] <-
  "Downsloping"

# Define level names for the "thal" attribute, to make plotting easier.
levels(cleveland_heart_disease$thal)[levels(cleveland_heart_disease$thal)==3] <-
  "Normal"
levels(cleveland_heart_disease$thal)[levels(cleveland_heart_disease$thal)==6] <-
  "Fixed Defect"
levels(cleveland_heart_disease$thal)[levels(cleveland_heart_disease$thal)==7] <-
  "Reversible Defect"

# Define level names for the "target" attribute, to make plotting easier.
levels(cleveland_heart_disease$target)[levels(cleveland_heart_disease$target)==0] <-
  "No Heart Disease"
levels(cleveland_heart_disease$target)[levels(cleveland_heart_disease$target)==1] <-
  "Heart Disease"
```

The structure of the reformatted dataset was then examined and confirmed.

```
# Examine and confirm the structure of the reformatted dataset.
str(cleveland_heart_disease)
```

```
## 'data.frame':    297 obs. of  14 variables:
##  $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 1 2 2 ...
##  $ cp      : Factor w/ 4 levels "Typical Angina",..: 1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : Factor w/ 2 levels "<= 120 mg/dl",..: 2 1 1 1 1 1 1 1 1 2 ...
##  $ restecg : Factor w/ 3 levels "Normal","ST-T Wave Abnormality",..: 3 3 3 1 3 1 3 1 3 3 ...
##  $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 2 1 2 ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ slope   : Factor w/ 3 levels "Upsloping","Flat",..: 3 2 2 3 1 1 3 1 2 3 ...
##  $ ca      : num  0 3 2 0 0 0 2 0 1 0 ...
##  $ thal    : Factor w/ 3 levels "Normal","Fixed Defect",..: 2 1 3 1 1 1 1 1 3 3 ...
##  $ target  : Factor w/ 2 levels "No Heart Disease",..: 1 2 2 1 1 1 2 1 2 2 ...
```

After reformatting, the dataset had:

- 297 rows (6 rows originally contained "?" values and had therefore been removed);
- 8 attributes with a type of *factor*, and with the factor levels labelled appropriately (for the 8 categorical attributes); and,
- 6 attributes with a type of *number* (for the 6 continuous attributes).

2.4 Further Exploration of the Data

The dataset was then explored further, to better understand the nature of the data.

Note: In the interests of readability, the R code used to generate the following tables and graphs was not displayed in this report. It can be viewed in the associated R and R Markdown scripts.

First, the number of patients with and without heart disease was calculated.

Table 3: Number of Patients with and without Heart Disease

Presence of Heart Disease	Count
No Heart Disease	160
Heart Disease	137

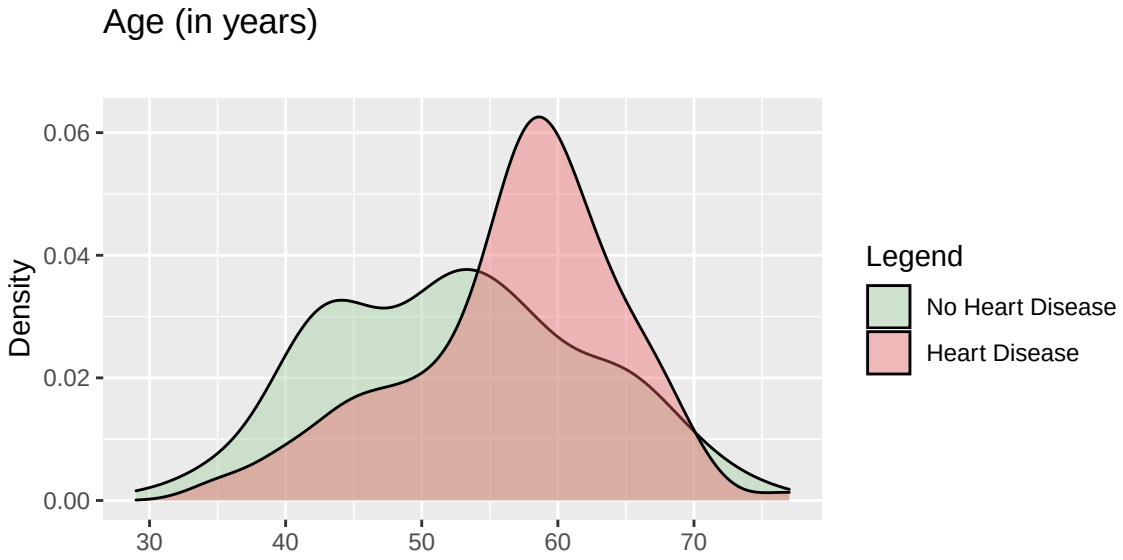
There was a reasonable number of patients in each category, which was useful for modelling purposes.

Next, each attribute was examined for any obvious correlation between it and the presence of heart disease. The method used to do this differed depending on whether the attribute was continuous or categorical.

For each continuous attribute, a density plot was generated that showed the distribution of patients with and without heart disease (in red and green respectively), overlaid on each other. This visually showed whether there were any obvious differences across the values.

For each categorical attribute:

1. A bar graph was generated to show the proportion of patients with and without heart disease (in red and green respectively), grouped by each value of the categorical attribute. This visually showed whether there were any obvious differences in proportions across the categories.
2. The number of patients for each value of the categorical attribute was calculated. This assisted the interpretation of the associated bar graph, because a noticeable variation across categories might be due to an individual category having a very small number of observations.



The density plot for age showed some distinction between the plots for patients with and without heart disease. This suggested there may have been some correlation between the two.

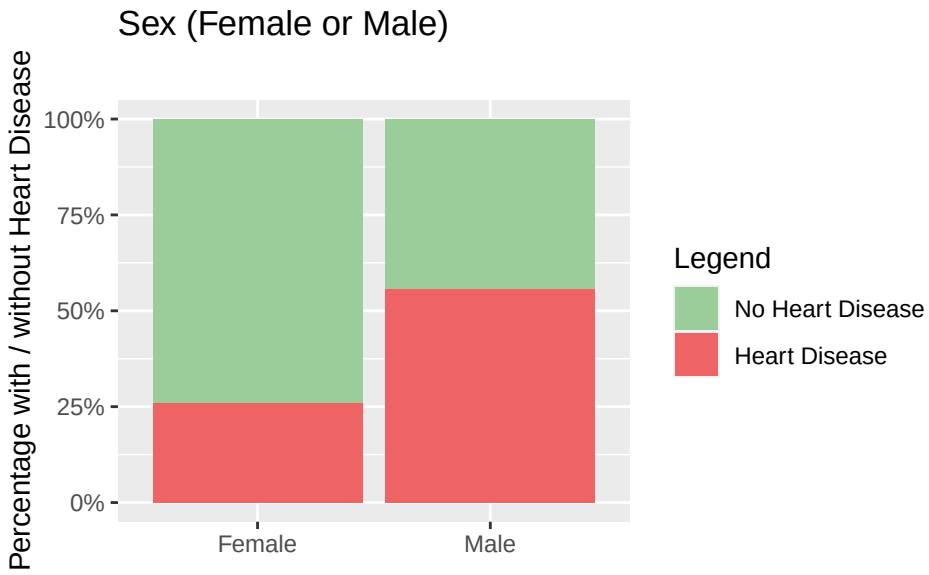


Table 4: Number of patients grouped by sex (female or male).

Sex	Count
Female	96
Male	201

The bar graph above indicated that male patients in the Cleveland dataset were around twice as likely to have heart disease than the female patients. While there were more than double the number of men than women, the number of female patients (96) was not insignificant. This suggested that *sex* had some predictive power in relation to heart disease.

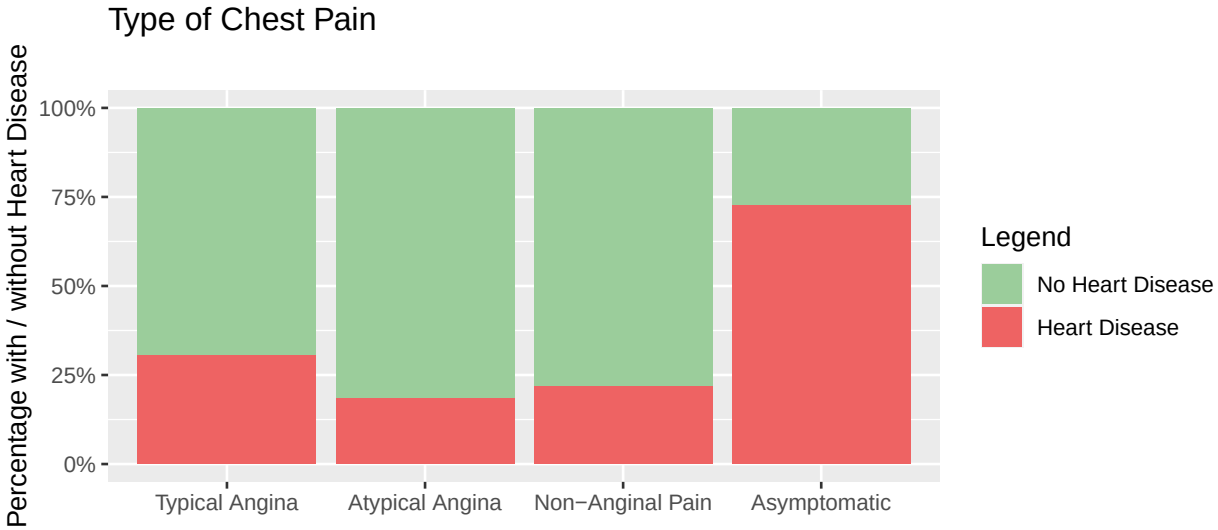
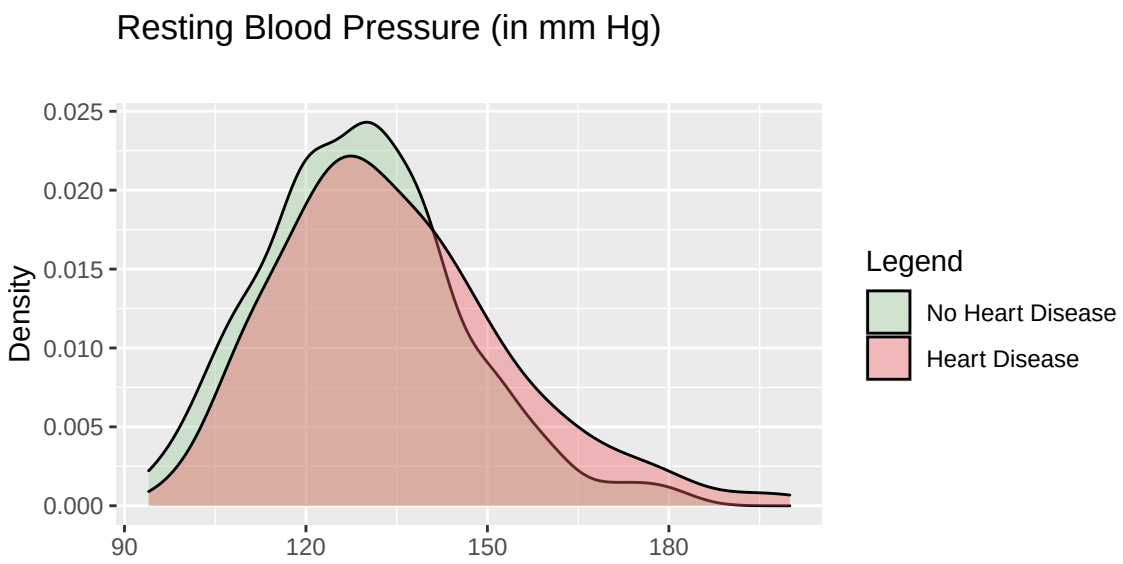


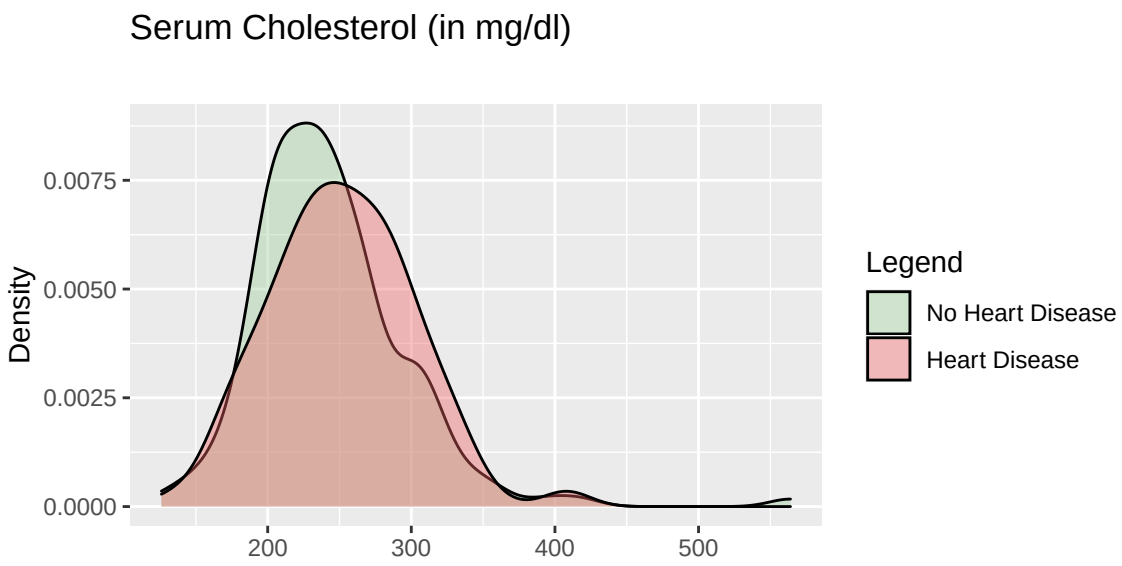
Table 5: Number of patients grouped by chest pain category.

Chest Pain	Count
Typical Angina	23
Atypical Angina	49
Non-Anginal Pain	83
Asymptomatic	142

There appeared to be some correlation between the type of chest pain experienced by a patient and the presence of heart disease. Notably, patients who reported **no** chest pain were much **more** likely to have heart disease than patients who reported some form of chest pain; and the number of patients in the dataset who were asymptomatic was relatively high.



The density plot for resting blood pressure showed no significant distinction between patients who did and did not have heart disease. This attribute therefore appeared to have little, if any, correlation with heart disease.



Similarly, the density plot for serum cholesterol showed little distinction between patients who did and did not have heart disease. This attribute therefore appeared to have little correlation with heart disease.

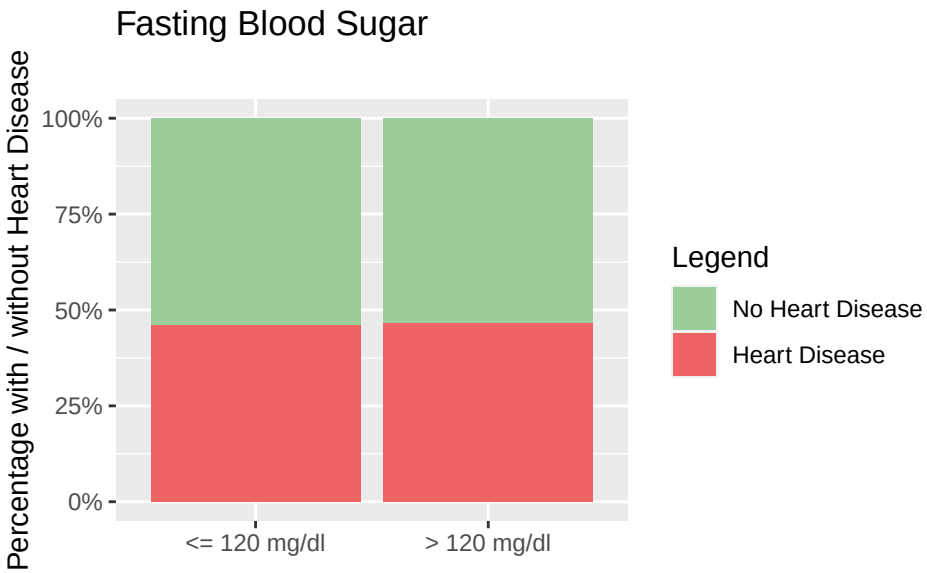


Table 6: Number of patients grouped by fasting blood sugar category.

Fasting Blood Sugar	Count
<= 120 mg/dl	254
> 120 mg/dl	43

The proportion of patients who did/did not have heart disease was virtually identical, regardless of their fasting blood sugar category. This attribute therefore appeared to have no correlation with heart disease.

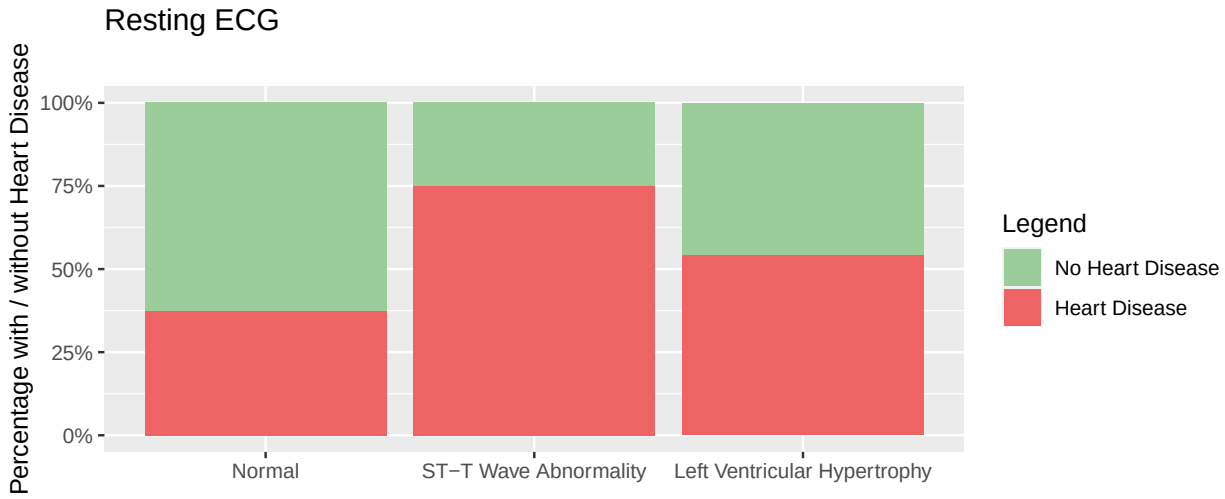
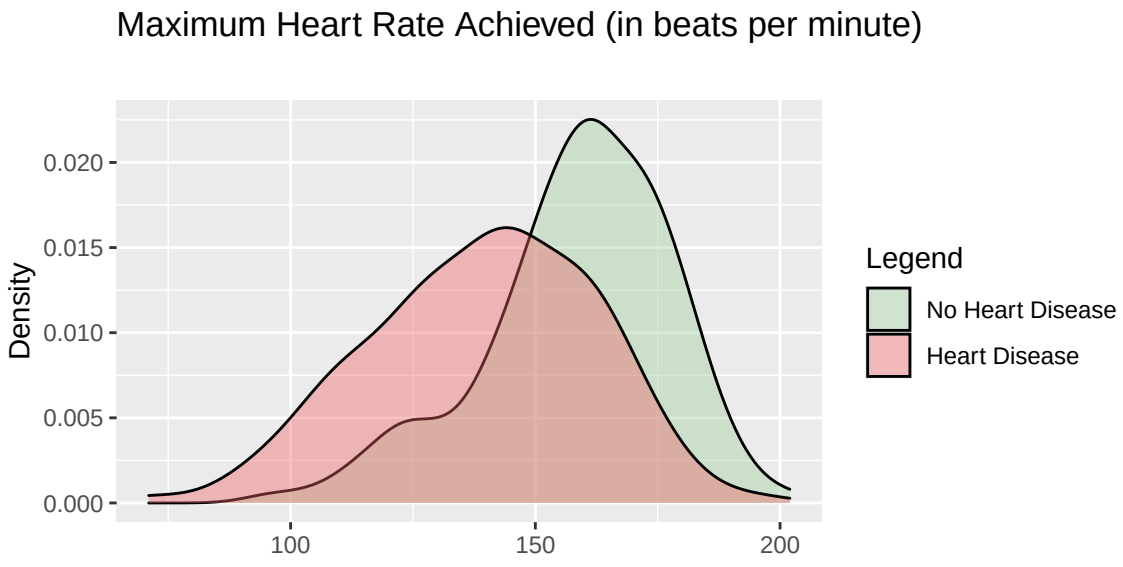


Table 7: Number of patients grouped by resting ECG category.

Resting ECG	Count
Normal	147
ST-T Wave Abnormality	4
Left Ventricular Hypertrophy	146

The bar graph above suggested that patients who had a resting ECG result that indicated an ST-T wave abnormality were significantly more likely to have heart disease than patients with a "normal" resting ECG result. However, there were only four patients in the former group, so this inference should be treated with caution. The number of patients with a result that indicated left ventricular hypertrophy was more significant, and based on the bar graph above, this did appear to have some correlation with the presence of heart disease.



The density plot for maximum heart rate achieved showed some distinction between patients who did and did not have heart disease. This attribute therefore appeared to have some correlation with heart disease.

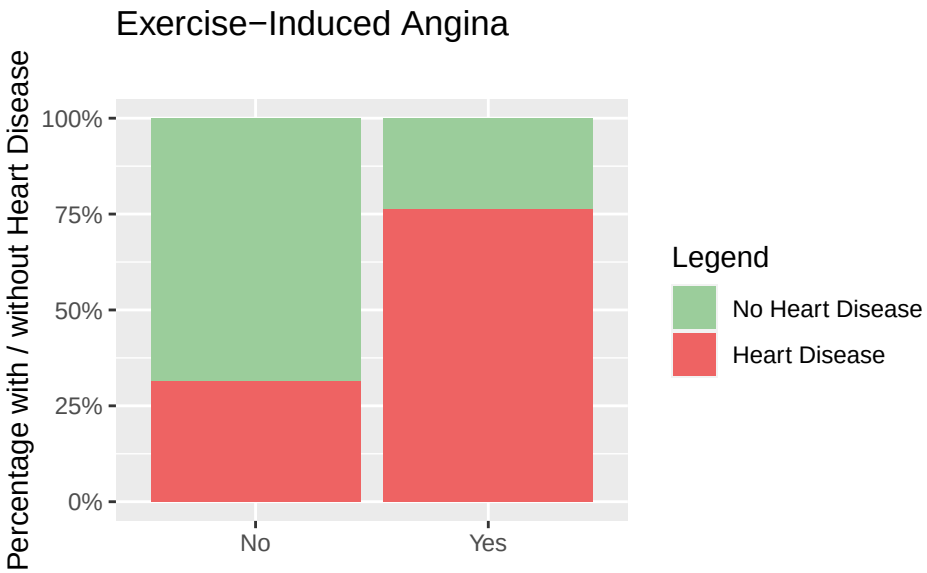
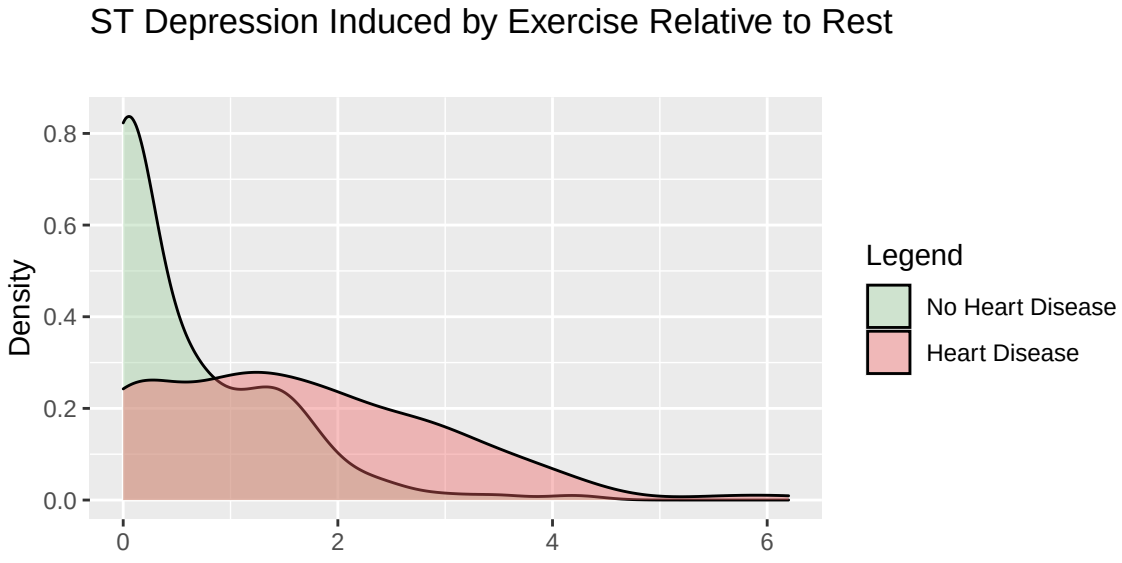


Table 8: Number of patients grouped by exercise-induced angina category.

Exercise-Induced Angina	Count
No	200
Yes	97

The bar graph above indicated that patients who experienced exercise-induced angina were more than twice as likely to have heart disease. This indicated considerable correlation between exercise-induced angina and heart disease.



The density plot for ST depression induced by exercise relative to rest above showed some distinction between patients who did and did not have heart disease. This attribute therefore appeared to have some correlation

with heart disease.

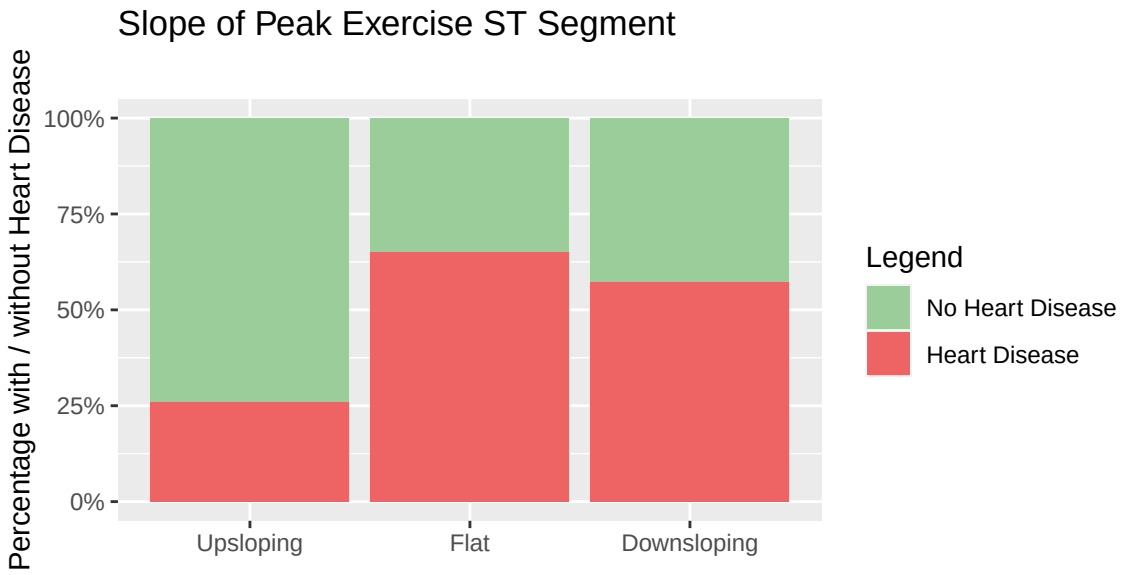
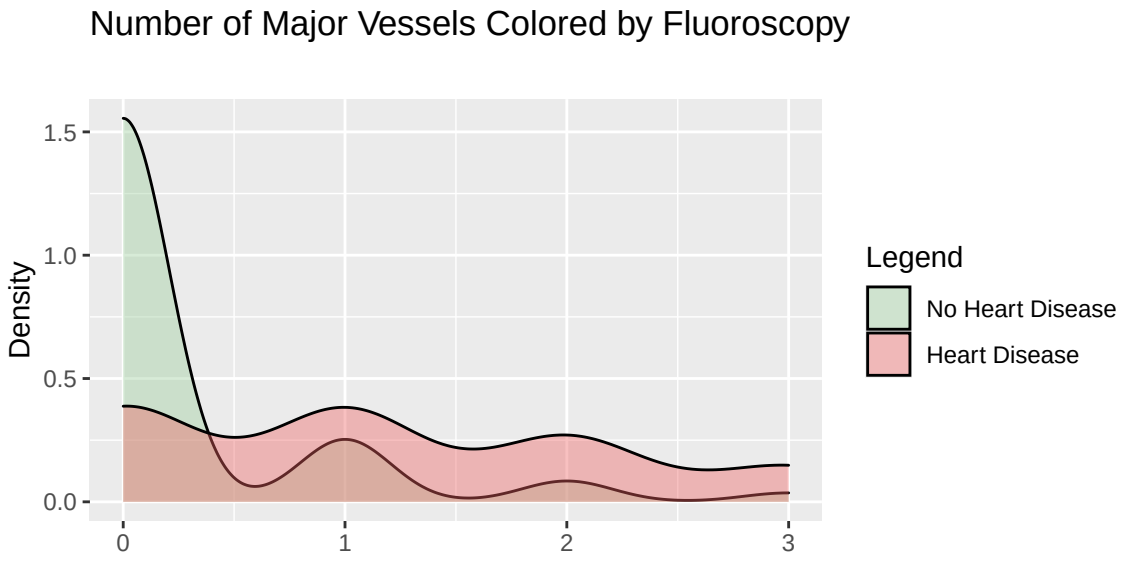


Table 9: Number of patients grouped by slope of peak exercise ST segment.

Slope of Peak Exercise ST Segment	Count
Upsloping	139
Flat	137
Downsloping	21

There appeared to be considerable correlation between the slope of peak exercise ST segment and the presence of heart disease - patients for whom it was upsloping were less than half as likely to have heart disease than other patients.



The density plot for the number of major vessels colored by fluoroscopy showed some distinction between patients who did and did not have heart disease. This attribute therefore appeared to have some correlation with heart disease.

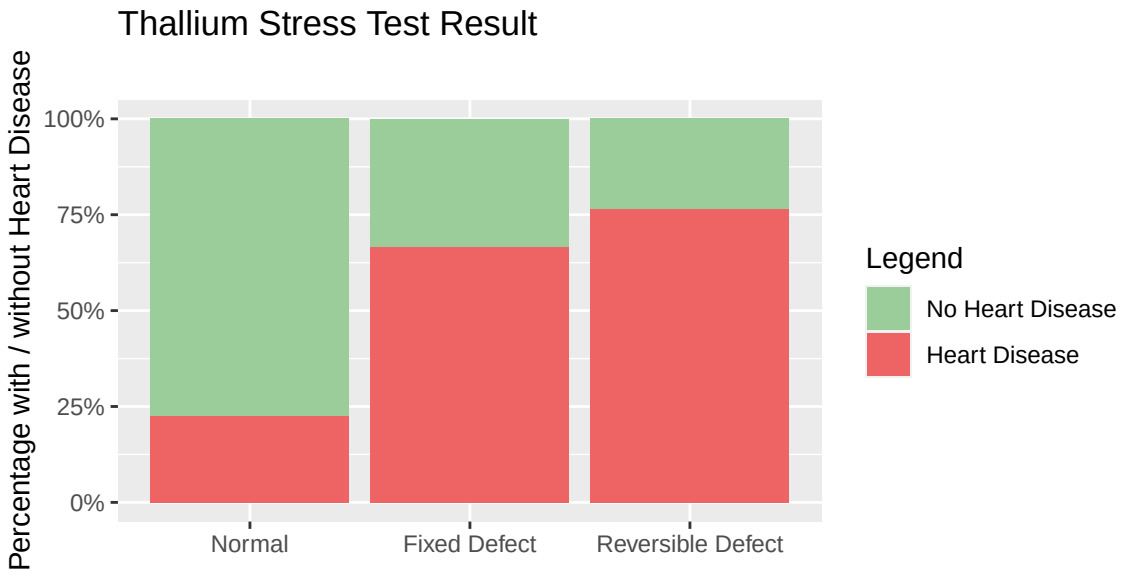


Table 10: Number of patients grouped by thallium stress test result.

Thallium Stress Test Result	Count
Normal	164
Fixed Defect	18
Reversible Defect	115

Finally, there appeared to be considerable correlation between a patient's thallium stress test results (in relation to any defects found) and the presence of heart disease. Patients for whom it was "normal" were almost two thirds **less** likely to have heart disease than other patients.

After the above exploration of the data was completed, it appeared that the following attributes were most likely to be potential predictors for the presence of heart disease:

- Age of the patient (*age* - continuous)
- Sex of the patient (*sex* - categorical)
- Type of chest pain (*cp* - categorical)
- Resting electrocardiographic results (*restecg* - categorical)
- Maximum heart rate achieved (*thalach* - continuous)
- Exercise-induced angina (*exang* - categorical)
- ST depression induced by exercise relative to rest (*oldpeak* - continuous)
- Slope of the peak exercise ST segment (*slope* - categorical)
- Number of major vessels colored by fluoroscopy (*ca* - continuous)
- Thallium stress test result (*thal* - categorical)

2.5 Modelling Approach

Several types of modelling were then explored to determine how accurately they predicted heart disease. Each model was first trained using a training subset of the *cleveland_heart_disease* dataset. Then, each model was applied to a separate, test, subset of the *cleveland_heart_disease* dataset, and the accuracy of its predictions assessed.

The accuracy of each model's predictions was determined using the "overall accuracy" calculated in a confusion matrix. "Overall accuracy" reported the proportion of observations in the test set that were correctly predicted by the model.

The following types of models were explored:

- Generalized Linear Model (glm)
- Linear Discriminant Analysis (lda)
- k-Nearest Neighbors (knn)
- Classification and Regression Tree (rpart)
- Random Forest (rf)

The training and test sets were created by partitioning the reformatted *cleveland_heart_disease* dataset into a *heart_train_set* and *heart_test_set*, each of which comprised 50% of the reformatted dataset. A 50/50 split was chosen to ensure there were enough observations in the test set to adequately validate the models.

```
##############################################################################################
# Set up the objects required for the modelling.
##############################################################################################

# Create training and test sets from the "cleveland_heart_disease" set.

# The following line of code assumes that R 3.6 or later is being used.
set.seed(1, sample.kind = "Rounding")

heart_test_index <- createDataPartition(y = cleveland_heart_disease$target,
                                        times = 1, p = 0.5, list = FALSE)
heart_train_set <- cleveland_heart_disease %>% slice(-heart_test_index)
heart_test_set <- cleveland_heart_disease %>% slice(heart_test_index)
```

```
# Display the structure of the new training set.
str(heart_train_set)
```

```
## 'data.frame':    148 obs. of  14 variables:
##  $ age     : num  63 67 56 62 57 53 57 56 56 52 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 2 2 1 2 2 ...
##  $ cp      : Factor w/ 4 levels "Typical Angina",..: 1 4 2 4 4 4 4 2 3 3 ...
##  $ trestbps: num  145 160 120 140 120 140 140 140 130 172 ...
##  $ chol    : num  233 286 236 268 354 203 192 294 256 199 ...
##  $ fbs     : Factor w/ 2 levels "<= 120 mg/dl",..: 2 1 1 1 1 2 1 1 2 2 ...
##  $ restecg : Factor w/ 3 levels "Normal","ST-T Wave Abnormality",..: 3 3 1 3 1 3 1 3 3 1 ...
##  $ thalach : num  150 108 178 160 163 155 148 153 142 162 ...
##  $ exang   : Factor w/ 2 levels "No","Yes": 1 2 1 1 2 2 1 1 2 1 ...
##  $ oldpeak : num  2.3 1.5 0.8 3.6 0.6 3.1 0.4 1.3 0.6 0.5 ...
##  $ slope   : Factor w/ 3 levels "Upsloping","Flat",..: 3 2 1 3 1 3 2 2 2 1 ...
##  $ ca      : num  0 3 0 2 0 0 0 0 1 0 ...
##  $ thal    : Factor w/ 3 levels "Normal","Fixed Defect",..: 2 1 1 1 1 3 2 1 2 3 ...
##  $ target  : Factor w/ 2 levels "No Heart Disease",..: 1 2 1 2 1 2 1 1 2 1 ...
```

```
# Display the structure of the new test set.
str(heart_test_set)
```

```
## 'data.frame':    149 obs. of  14 variables:
##  $ age     : num  67 37 41 63 44 57 48 54 64 58 ...
##  $ sex     : Factor w/ 2 levels "Female","Male": 2 2 1 2 2 2 2 2 2 2 ...
##  $ cp      : Factor w/ 4 levels "Typical Angina",..: 4 3 2 4 2 3 2 4 1 2 ...
##  $ trestbps: num  120 130 130 130 120 150 110 140 110 120 ...
##  $ chol    : num  229 250 204 254 263 168 229 239 211 284 ...
##  $ fbs     : Factor w/ 2 levels "<= 120 mg/dl",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ restecg : Factor w/ 3 levels "Normal","ST-T Wave Abnormality",..: 3 1 3 3 1 1 1 1 3 3 ...
##  $ thalach : num  129 187 172 147 173 174 168 160 144 160 ...
##  $ exang   : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
##  $ oldpeak : num  2.6 3.5 1.4 1.4 0 1.6 1 1.2 1.8 1.8 ...
##  $ slope   : Factor w/ 3 levels "Upsloping","Flat",..: 2 3 1 2 1 1 3 1 2 2 ...
##  $ ca      : num  2 0 0 1 0 0 0 0 0 0 ...
##  $ thal    : Factor w/ 3 levels "Normal","Fixed Defect",..: 3 1 1 3 3 1 3 1 1 1 ...
##  $ target  : Factor w/ 2 levels "No Heart Disease",..: 2 1 1 2 1 1 2 1 1 2 ...
```

Having created the training and test sets, each model was then run, and the results recorded.

```
##########################################################################################
# Apply different models to try to predict whether or not the patient has heart disease.
##########################################################################################

# Generalized Linear Model (glm)
fit_glm <- train(target ~ ., method ="glm", data = heart_train_set)
y_hat_glm <- predict(fit_glm, heart_test_set, type = "raw")
cm_glm <- confusionMatrix(data = y_hat_glm, reference = heart_test_set$target)

model_results <- tibble(Method = "Generalized Linear Model",
                        Accuracy = cm_glm$overall[["Accuracy"]],
                        Sensitivity = cm_glm$byClass[["Sensitivity"]],
                        Specificity = cm_glm$byClass[["Specificity"]])

# Linear Discriminant Analysis (lda)
fit_lda <- train(target ~ ., method ="lda", data = heart_train_set)
y_hat_lda <- predict(fit_lda, heart_test_set, type = "raw")
cm_lda <- confusionMatrix(data = y_hat_lda, reference = heart_test_set$target)

model_results <- bind_rows(model_results,
                           tibble(Method="Linear Discriminant Analysis",
                                  Accuracy = cm_lda$overall[["Accuracy"]],
                                  Sensitivity = cm_lda$byClass[["Sensitivity"]],
                                  Specificity = cm_lda$byClass[["Specificity"]]))

# k-Nearest Neighbors (knn)
fit_knn <- train(target ~ ., method ="knn", data = heart_train_set)
y_hat_knn <- predict(fit_knn, heart_test_set, type = "raw")
cm_knn <- confusionMatrix(data = y_hat_knn, reference = heart_test_set$target)

model_results <- bind_rows(model_results,
                           tibble(Method="k-Nearest Neighbors",
                                  Accuracy = cm_knn$overall[["Accuracy"]],
```

```
                                Sensitivity = cm_knn$byClass[["Sensitivity"]],
                                Specificity = cm_knn$byClass[["Specificity"]]))

# Classification and Regression Tree (rpart)
fit_rpart <- train(target ~ ., method ="rpart", data = heart_train_set)
y_hat_rpart <- predict(fit_rpart, heart_test_set, type = "raw")
cm_cart <- confusionMatrix(data = y_hat_rpart, reference = heart_test_set$target)

model_results <- bind_rows(model_results,
                           tibble(Method="Classification and Regression Tree",
                                Accuracy = cm_cart$overall[["Accuracy"]],
                                Sensitivity = cm_cart$byClass[["Sensitivity"]],
                                Specificity = cm_cart$byClass[["Specificity"]]))

# Random Forest (rf)
fit_rf <- train(target ~ ., method ="rf", data = heart_train_set)
y_hat_rf <- predict(fit_rf, heart_test_set, type = "raw")
cm_rf <- confusionMatrix(data = y_hat_rf, reference = heart_test_set$target)

model_results <- bind_rows(model_results,
                           tibble(Method="Random Forest",
                                Accuracy = cm_rf$overall[["Accuracy"]],
                                Sensitivity = cm_rf$byClass[["Sensitivity"]],
                                Specificity = cm_rf$byClass[["Specificity"]]))
```

The results produced by each model are presented in the *Results* section below.

3 Results

The results for each model were as follows. Note that the "overall accuracy" of each model was used to assess its accuracy; but sensitivity and specificity were also reported to better understand the performance of each model.

```
# Display the results from all of the models.
knitr::kable(model_results)
```

Method	Accuracy	Sensitivity	Specificity
Generalized Linear Model	0.7919463	0.8250	0.7536232
Linear Discriminant Analysis	0.8187919	0.8875	0.7391304
k-Nearest Neighbors	0.6308725	0.7375	0.5072464
Classification and Regression Tree	0.6711409	0.6625	0.6811594
Random Forest	0.8120805	0.8625	0.7536232

In descending order of accuracy, the results were as follows.

```
# Display the models sorted in descending order of accuracy.
knitr::kable(model_results %>% arrange(desc(Accuracy)))
```

Method	Accuracy	Sensitivity	Specificity
Linear Discriminant Analysis	0.8187919	0.8875	0.7391304
Random Forest	0.8120805	0.8625	0.7536232
Generalized Linear Model	0.7919463	0.8250	0.7536232
Classification and Regression Tree	0.6711409	0.6625	0.6811594
k-Nearest Neighbors	0.6308725	0.7375	0.5072464

As shown above, the Linear Discriminant Analysis model produced the most accurate predictions, with an overall accuracy of approximately 81.9%. It also exhibited a reasonable level of sensitivity of 88.75%.

4 Conclusion

This project explored the use of various models to predict the presence of heart disease, and identified the model that produced the most accurate predictions, using the Cleveland heart disease dataset.

The most accurate predictions were generated using Linear Discriminant Analysis, which had an overall accuracy of approximately 81.9% when applied to the test set. The Random Forest model had a slightly lower overall accuracy of 81.2%.

There were some limitations on the analysis that could be performed, including the following:

- The *cleveland_heart_disease* dataset used in the modelling was relatively small (297 observations), and partitioned into training and test sets of only 148 and 149 observations respectively. Similar datasets were available from other locations (Long Beach, Hungary, and Switzerland) but they contained a large number of missing values. This made them much less suitable for modelling purposes compared to the Cleveland dataset, which was relatively clean and complete.
- It is likely that most of the patients lived in the same, relatively small (in global terms) geographic region. The modelling might not accurately predict heart disease in patients from other regions or countries, where various environmental, economic and lifestyle factors could impact people's cardiovascular health very differently.

While the Linear Discriminant Analysis and Random Forest models produced moderately accurate predictions, it is questionable that they would be sufficient on their own to be used for clinical purposes. At best, they might be able to be used in combination with other tools or techniques to assist in predicting or diagnosing the presence of heart disease.