HarvardX PH125.9x Data Science: Capstone Heart Desease

Irina Shevchuk

2024-12-09

1. Introduction

1.1 The project information

The goal of this project is to create a model that could predict the likelihood of heart disease based on patient data. This project was created as part of the professional certification program in Data Science offered by HarvardX. Calculations are performed using R (a programming language for statistical computing and data visualization). Note: It is an student project created by an individual without formal medical education. The project result cannot be used to make a decision about treatment tactics. Nevertheless, during the analysis, there was an attempt to look at the data from the point of view of a doctor. What information should be interesting and useful when assessing the likelihood of a disease to a practicing physician.

1.2 Dataset Information

In this project was used the data set "Heart Disease" available in open source on the website Kaggle This data set dates from 1988 and consists of one database: Cleveland and contains 14 attributes. The "condition" field refers to the presence of heart disease in the patient.

PREDICTORS (FEATURES) 1. **age**: age in years Integer: year 2. **sex**: the biological trait that determines whether a sexually reproducing organism produces male or female gametes
Categorical: 1 = MALE, 0 = FEMALE 3. **cp**: chest pain type Categorical: 0 = TYPICAL ANGINA, 1 = ATYPICAL ANGINA, 2 = NON-ANGINAL PAIN, 3 = ASYMPTOMATIC 4. **trestbps**: resting blood pressure (on admission to the hospital) Integer: mm Hg 5. **chol**: serum cholestoral Integer: mg/dl 6. **fbs**: fasting blood sugar > 120 mg/dl Categorical: 1 = YES, 0 = NO 7. **restecg**: resting electrocardiographic results Categorical: 0 = NORMAL, 1 = having ST-T wave ABNORMALITY (T wave inversions and/or ST elevation or depression of > 0.05 mV), 2 = showing PROBABLE OR DEFINITE left ventricular hypertrophy by Estes' criteria 8. **thalach**: maximum heart rate achieved Integer: heart rate 9. **exang**: exercise induced angina Categorical: 1 = YES, 0 = NO 10. **oldpeak**: ST depression induced by exercise relative to rest Integer: 11. **slope**: the slope of the peak exercise ST segment Categorical: 0 = UPSLOPING, 1 = FLAT, 2 = DOWNSLOPING 12. **ca**: number of major vessels (0-3) colored by flourosopy, number of vessels Integer: number of vessels 13. **thal**: Thalassemia is an inherited blood disorder that affects hemoglobin production as well as heart function Categorical: 0 = NORMAL, 1 = FIXED DEFECT, 2 = REVERSABLE DEFECT

TARGET 14. **condition**: diagnosis of heart disease Categorical: 1 = YES, 0 = NO

The original data set available in open source on the website The UCI Machine Learning Repository. Consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. The database contains 76 attributes. The "TARGET" field refers to the presence of heart disease in the patient. The data from the original data set was not used in this project.

```r
# Specify the URL of the dataset in the repository
dataset_url <- "https://raw.githubusercontent.com/Irina-Mikhailovna/HarvardX-PH125.9x-Data-Science-Caps
# Loaded the database
data <- read.csv(dataset_url, na.strings = "?", header = TRUE)
```

1.3 Exploratory Data Analysis

The necessary libraries was downloaded for analysis and calculations

```r
# Installation and download of packages

if (!require(tidyverse)) install.packages("tidyverse") # Data tidying
if (!require(ggplot2)) install.packages("ggplot2") # Data Visualization
if (!require(ggcorrplot)) install.packages("ggcorrplot") # Correlation matrix
if (!require(dplyr)) install.packages("dplyr") # Data manipulation
if (!require(rpart)) install.packages("rpart") # Building classification and regression trees
if (!require(tidyr)) install.packages("tidyr") # Create tidy data
if (!require(patchwork)) install.packages("patchwork") # Make plot composition
if (!require(corrplot)) install.packages("corrplot") # Correlation matrix
if (!require(caret)) install.packages("caret") # Complex regression and classification problems
if (!require(rpart)) install.packages("rpart") # Decision Trees
if (!require(randomForest)) install.packages("randomForest") # Random Forest
if (!require(xgboost)) install.packages("xgboost") # Gradient Boosting
if (!require(pROC)) install.packages("pROC") # ROC curves
if (!require(stargazer)) install.packages("stargazer") # creates LATEX code
if (!require(tinytex)) install.packages("tinytex") # 'LaTeX' documents
if (!require(gtsummary)) install.packages("gtsummary") # publication-ready analytical and  tables
if (!require(knitr)) install.packages("knitr") # global settings for R Markdown script

library(tidyverse) # Data tidying
library(ggplot2) # Data Visualization
library(ggcorrplot) # Correlation matrix
library(dplyr) # Data manipulation
library(rpart) # Building classification and regression trees
library(tidyr) # Create tidy data
library(patchwork) # Make plot composition
library(corrplot) # Correlation matrix
library(caret) # Complex regression and classification problems
library(rpart) # Decision Trees
library(randomForest) # Random Forest
library(xgboost) # Gradient Boosting
library(pROC) # ROC curves
library(stargazer) # creates LATEX code
library(tinytex) # 'LaTeX' documents
library(gtsummary) # publication-ready analytical and  tables
library(knitr) # global settings for R Markdown script
```

Was check the quality data set: The dataset was examined for any missing values. The result = 0 it mean that there are no missing values in the set

```r
# Checked for missing values
sum(is.na(data))
```

```
## [1] 0
```

The data structure was analyzed. The dataset does contain 14 features.

```
#  Summary of the data set structure
str(data)
```

```
## 'data.frame':    297 obs. of  14 variables:
##  $ age      : int  69 69 66 65 64 64 63 61 60 59 ...
##  $ sex      : int  1 0 0 1 1 1 1 1 0 1 ...
##  $ cp       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ trestbps : int  160 140 150 138 110 170 145 134 150 178 ...
##  $ chol     : int  234 239 226 282 211 227 233 234 240 270 ...
##  $ fbs      : int  1 0 0 1 0 0 1 0 0 0 ...
##  $ restecg  : int  2 0 0 2 2 2 2 0 0 2 ...
##  $ thalach  : int  131 151 114 174 144 155 150 145 171 145 ...
##  $ exang    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak  : num  0.1 1.8 2.6 1.4 1.8 0.6 2.3 2.6 0.9 4.2 ...
##  $ slope    : int  1 0 2 1 1 1 2 1 0 2 ...
##  $ ca       : int  1 2 0 1 0 0 0 2 0 0 ...
##  $ thal     : int  0 0 0 0 0 2 1 0 0 2 ...
##  $ condition: int  0 0 0 1 0 0 0 1 0 0 ...
```

Analysis of the set structure shows that some data belong to categories, some are continuous. To facilitate the analysis, a new set of identical data was created **data_text**. The numerical value of these categories has been changed to a test value. The details of the features is given in the description of the date set Kaggle. **Data_text** is used only for visual data analysis.

The initial set **data** will be used for modeling.

```
# In category type data, change the numeric value to a text value to facilitate data analysis

data_text <- data %>%
  mutate(sex = if_else(sex == 1, "MALE", "FEMALE"),
         cp = if_else(cp == 0,"TYPICAL ANGINA", if_else(cp == 1, "ATYPICAL ANGINA",
                      if_else(cp == 2, "NON-ANGINAL PAIN", "ASYMPTOMATIC"))),
         fbs = if_else(fbs == 1, ">120", "<=120"),
         restecg = if_else(restecg == 0, "NORMAL",
                           if_else(restecg == 1, "ABNORMALITY", "PROBABLE OR DEFINITE")),
         exang = if_else(exang == 1, "YES" ,"NO"),
         slope = if_else(slope == 0, "UPSLOPING",
                         if_else(slope == 1, "FLAT", "DOWNSLOPING")),
         thal = if_else(thal == 0, "NORMAL",
                        if_else(thal == 1, "FIXED DEFECT", "REVERSABLE DEFECT")),
         condition = if_else(condition == 1, "Disease", "No Disease")
         ) %>%
  mutate_if(is.character, as.factor) %>% dplyr::select(sex, cp, fbs,restecg, exang, slope, ca, thal, co
```

The data structure was checked.

```
summary(data_text)
```

```
##      sex                      cp         fbs                      restecg
##  FEMALE: 96   ASYMPTOMATIC    :142   <=120:254   ABNORMALITY         :  4
##  MALE  :201   ATYPICAL ANGINA : 49   >120 : 43   NORMAL              :147
##               NON-ANGINAL PAIN: 83               PROBABLE OR DEFINITE:146
##               TYPICAL ANGINA  : 23
##
##
##  exang             slope           ca                        thal
##  NO :200   DOWNSLOPING: 21   Min.   :0.0000   FIXED DEFECT     : 18
##  YES: 97   FLAT       :137   1st Qu.:0.0000   NORMAL           :164
##            UPSLOPING  :139   Median :0.0000   REVERSABLE DEFECT:115
##                              Mean   :0.6768
##                              3rd Qu.:1.0000
##                              Max.   :3.0000
##       condition        age           trestbps          chol
##  Disease   :137   Min.   :29.00   Min.   : 94.0   Min.   :126.0
##  No Disease:160   1st Qu.:48.00   1st Qu.:120.0   1st Qu.:211.0
##                   Median :56.00   Median :130.0   Median :243.0
##                   Mean   :54.54   Mean   :131.7   Mean   :247.4
##                   3rd Qu.:61.00   3rd Qu.:140.0   3rd Qu.:276.0
##                   Max.   :77.00   Max.   :200.0   Max.   :564.0
##     thalach         oldpeak
##  Min.   : 71.0   Min.   :0.000
##  1st Qu.:133.0   1st Qu.:0.000
##  Median :153.0   Median :0.800
##  Mean   :149.6   Mean   :1.056
##  3rd Qu.:166.0   3rd Qu.:1.600
##  Max.   :202.0   Max.   :6.200
```

Analysis of the sample structure showed that the proportion of men is almost twice as large as women. Checked the percentage of men and women have disease

```
# Counted the total number of men and women
total_count_sex <- data_text %>% group_by(sex) %>% summarize(Observations = n())

# Counted men and women have disease
Observations <- data_text %>%
  filter(condition == "Disease") %>%
  group_by(sex) %>%
  summarize(Disease_cases = n())

# Combined data and calculated the percentage
percent_disease <- total_count_sex %>%
  left_join(Observations, by = "sex") %>%
  mutate(Disease_cases = replace_na(Disease_cases, 0),
         "% disease" = (round((Disease_cases/Observations) * 100,2)))

percent_disease
```

```
## # A tibble: 2 x 4
##   sex    Observations Disease_cases `% disease`
##   <fct>         <int>         <int>       <dbl>
## 1 FEMALE           96            25        26.0
## 2 MALE            201           112        55.7
```

The percentage of the disease in the group of men is also 2 times higher than in the group of women. Such a difference in the percentage of men and women and the percentage of diseases can be explained as a sampling principle, difference in the biology of the genders, or social and cultural reasons[6,7]. Note: As part of this work, only dataset data was used, other sources were not analyzed.

1.4 Preparation of data for further analysis stages

The data set is relatively small (297 observations), so the division into training and test sets: in the proportion of 70/30 `p = 0.7`.

The training set will be used to train the model, and the test set is used to test its performance on new data.

```
# The division into training and test sets
set.seed(123)
train_index <-  createDataPartition(data$condition, times = 1, p = 0.7, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]
```

The statistics, both sets have about the same percentage of disease. This means that the data set was divided correctly.

```
# Summary train_data and test_data
Summary_sets_table <- tibble(
  Set = c("train_data", "test_data"),
  Observations = c(nrow(train_data), nrow(test_data)),
  "Disease cases" = c(sum(train_data$condition), sum(test_data$condition)),
  "% disease" = round(c(sum(train_data$condition)/nrow(train_data)*100, sum(test_data$condition)/nrow(t
  )

Summary_sets_table %>% knitr::kable()
```

Set	Observations	Disease cases	% disease
train_data	208	97	46.63
test_data	89	40	44.94

2. Models

Since the task of predicting the presence of heart disease is a classification task (presence of disease = 1, no disease = 0), 3 models will be used in this project:

1. Logistic regression: Model all predictors (features)
2. Logistic regression: Model significant predictors (features)
3. Decision Trees: Model all predictors (features)

The models are trained on **train_data** data and check on **test_data**.

The following indicators are used to evaluate the model:

1. Root Mean Squared Error (RMSE)

the residuals' standard deviation, or the average difference between the projected and actual values produced by a statistical model. The lower the RMSE, the closer the predicted values are to the actual ones given by the model.

$$RMSE = \sqrt{\frac{1}{n}\sum (y_i \ - \hat{y}_i)^2}$$

2. Accuracy

The proportion of correct predictions. The higher this indicator, the more correct predictions the model gives.

$$Accuracy = \frac{TP + TN}{\text{Total prediction}}$$

3. Precision

The proportion of correctly predicted positive cases among all predicted positives. The higher this indicator, the more correct predictions the model gives.

$$Precision = \frac{TP}{TP + FP}$$

4. Recall

The proportion of correctly predicted positive cases among all actual positives. The higher this indicator, the more correct predictions the model gives.

$$Recall = \frac{TP}{TP + FN}$$

5. F1 Score

The harmonic mean of Precision and Recall, providing a balance between the two metrics

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

2.1 Logistic regression

2.1.1 Modeling approach

Logistic regression estimates the probability that the target variable (condition) will take the value 1 (disease) based on the values of other variables. The result of the model predicts the probability, and then it can be converted into a prediction for the class. If the probability exceeds the threshold value (usually 0.5), the result is classified as 1 (disease), otherwise as 0 (no disease).

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + ... + \beta_n \cdot X_n)}}$$

where:

$$e - Euler's\ number (e \approx 2.718)$$

$$\beta_0, \beta_1 ... \beta_n - feature\ coefficients$$

$$X_1, X_2 ... X_n - feature$$

2.1.2 Logistic regression: Model all predictors (MAP)

2.1.2.1 Created the logistic regression: Model all predictors (MAP)

Creating a model logistic regression wiht all predictors (features) except condition are used to predict the target = `condition ~ .`.

```
model_log_MAP <- glm(condition ~ ., data = train_data, family = binomial)
summary(model_log_MAP)
```

```
##
## Call:
## glm(formula = condition ~ ., family = binomial, data = train_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.5235326  3.4044265  -2.210 0.027110 *
## age          0.0144584  0.0280616   0.515 0.606387
## sex          1.6978012  0.5853695   2.900 0.003727 **
## cp           0.6125655  0.2417992   2.533 0.011297 *
## trestbps     0.0006434  0.0127262   0.051 0.959681
## chol         0.0093762  0.0050590   1.853 0.063829 .
## fbs         -0.9898786  0.6603357  -1.499 0.133860
## restecg      0.3384178  0.2178811   1.553 0.120370
## thalach     -0.0052691  0.0123053  -0.428 0.668509
## exang        1.3449196  0.5113010   2.630 0.008529 **
## oldpeak      0.3273509  0.2522740   1.298 0.194425
## slope        0.6893088  0.4275284   1.612 0.106894
## ca           1.1530729  0.3030671   3.805 0.000142 ***
## thal         0.5608630  0.2424066   2.314 0.020683 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 287.41  on 207  degrees of freedom
## Residual deviance: 146.80  on 194  degrees of freedom
## AIC: 174.8
##
## Number of Fisher Scoring iterations: 6
```

2.1.2.2 Practical application of the logistic regression: Model all predictors (MAP)

As a result, a formula was obtained by which the presence of the heart disease can be predicted. For practical application and assessment of the probability of a patient having heart disease, instead of the X feature, the value of a specific patient's feature must be substituted into the formula.

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + ... + \beta_{13} \cdot X_{13})}}$$

where:

$$e - Euler's\ number (e \approx 2.718)$$

$$\beta_0 = -7.5235326$$

$$\beta_1 \cdot X_1 = 0.0144584 \cdot X_{age}$$

$$\beta_2 \cdot X_2 = 1.6978012 \cdot X_{sex}$$

$$\beta_3 \cdot X_3 = 0.6125655 \cdot X_{cp} - chest\ pain\ type\ feauture$$

$$\beta_4 \cdot X_4 = 0.0006434 \cdot X_{trestbps} - resting\ blood\ pressure\ feauture$$

$$\beta_5 \cdot X_5 = 0.0093762 \cdot X_{chol} - serum\ cholestoral\ feauture$$

$$\beta_6 \cdot X_6 = -0.9898786 \cdot X_{fbs} - fasting\ blood\ sugar\ feauture$$

$$\beta_7 \cdot X_7 = 0.3384178 \cdot X_{restecg} - resting\ electrocardiographic\ results\ feauture$$

$$\beta_8 \cdot X_8 = -0.0052691 \cdot X_{thalach} - maximum\ heart\ rate\ achieved\ feauture$$

$$\beta_9 \cdot X_9 = 1.3449196 \cdot X_{exang} - exercise\ induced\ angina\ feauture$$

$$\beta_{10} \cdot X_{10} = 0.3273509 \cdot X_{oldpeak} - ST\ depression\ induced\ by\ exercise\ relative\ to\ rest\ feauture$$

$$\beta_{11} \cdot X_{11} = 0.6893088 \cdot X_{slope} - the\ slope\ of\ the\ peak\ exercise\ ST\ segment\ feauture$$

$$\beta_{12} \cdot X_{12} = 1.1530729 \cdot X_{ca} - number\ of\ major\ vessels\ (0-3)\ colored\ by\ flourosopy\ feauture$$

$$\beta_{13} \cdot X_{13} = 0.5608630 \cdot X_{thal} - thalassemia\ (an\ inherited\ blood\ disorder)\ feauture$$

2.1.2.3 Efficiency assessment the logistic regression: Model all predictors (MAP)

The confusion matrix:

```
# Predict probabilities for the test data
predictions_log_MAP <- predict(model_log_MAP, newdata = test_data, type = "response")

# Convert probabilities to binary class predictions
predicted_classes_log_MAP <- ifelse(predictions_log_MAP > 0.5, 1, 0)

# Generate the confusion matrix
confusion_matrix_log_MAP <- table(Predicted = predicted_classes_log_MAP, Actual = test_data$condition)

# Print the confusion matrix
print_confusion_matrix_log_MAP <- paste(
  "The confusion matrix:\n\n",
  paste(capture.output(print(confusion_matrix_log_MAP)), collapse = "\n"), "\n\n",
  sprintf("TN: True Negative (correctly predicted 0)       %d", confusion_matrix_log_MAP[1, 1]), "\n",
  sprintf("FP: False Positive (incorrectly predicted 1)    %d", confusion_matrix_log_MAP[1, 2]), "\n",
  sprintf("FN: False Negative (incorrectly predicted 0)    %d", confusion_matrix_log_MAP[2, 1]), "\n",
  sprintf("TP: True Positive (correctly predicted 1)       %d", confusion_matrix_log_MAP[2, 2]), "\n",
  sep = ""
)

cat(print_confusion_matrix_log_MAP)
```

```
## The confusion matrix:
##
##           Actual
## Predicted  0  1
##         0 40  7
##         1  9 33
##
```

```
## TN: True Negative (correctly predicted 0)       40
## FP: False Positive (incorrectly predicted 1)    7
## FN: False Negative (incorrectly predicted 0)    9
## TP: True Positive (correctly predicted 1)       33
```

```r
# RMSE
rmse_log_MAP <- sqrt(mean((test_data$condition - predictions_log_MAP)^2))
# Accuracy
accuracy_log_MAP <- sum(diag(confusion_matrix_log_MAP)) / sum(confusion_matrix_log_MAP)
# Precision
precision_log_MAP <- confusion_matrix_log_MAP[2, 2] / sum(confusion_matrix_log_MAP[2, ])
# Recall
recall_log_MAP <- confusion_matrix_log_MAP[2, 2] / sum(confusion_matrix_log_MAP[, 2])
# F1 Score
f1_score_log_MAP <- 2 * ((precision_log_MAP * recall_log_MAP) / (precision_log_MAP + recall_log_MAP))
```

2.1.2.4 Result and conclusion on the effectiveness of the model: logistic regression: Model all predictors (MAP)

```r
results_table <- tibble(
  Model = c("Logistic regression: Model all predictors"),
  RMSE = round(c(rmse_log_MAP),5),
  Accuracy = round(c(accuracy_log_MAP),5),
  Precision = round(c(precision_log_MAP),5),
  Recall = c(recall_log_MAP),
  F1_Score = round(c(f1_score_log_MAP),5)
)
results_table %>% knitr::kable()
```

Model	RMSE	Accuracy	Precision	Recall	F1_Score
Logistic regression: Model all predictors	0.35678	0.82022	0.78571	0.825	0.80488

The model has achieved **accuracy** 0.82022, which indicates a high ability to classify data correctly as a whole. This shows that the model has correctly classified the 82% of all cases. **The RMSE** value was 0.35678, which indicates a low) error in the predicted probability values. For a positive class, the model showed **Precision** 0.78571, which means that 79% of the cases predicted as positive were indeed positive. **Recall** was 0.825, which shows that the model successfully detected 83% of all real positive cases. **False Negative** (incorrectly predicted 0) accounted for 9 cases or 10% of the total number of observations, which can be critical in assessing the presence of the disease in cases where the patient needs urgent medical care.

2.1.3 Logistic regression: Model significant predictors (MSP)

2.1.3.1 Created the logistic regression: Model significant predictors (MSP)

Because the coefficients of many factors are close to zero. The model has been recalculated taking into account significant factors. To search for significant predictors, the value Pr(>|z|) was used to indicate the probability of obtaining the desired result by chance. Predictors with values of $p<0.05$ are considered statistically significant. If $p>0.05$, such predictors can be excluded from the model.

```r
# Obtaining coefficients and p-values
coeffs_log_MAP <- summary(model_log_MAP)$coefficients

# Significant predictors (p-value < 0.05)
significant_predictors_log_MAP <- rownames(coeffs_log_MAP)[coeffs_log_MAP[, 4] < 0.05]
cat("Significant predictors:", paste(significant_predictors_log_MAP, collapse = ", "), "\n")
```

```
## Significant predictors: (Intercept), sex, cp, exang, ca, thal
```

```r
# Insignificant predictors (p-value > 0.05)
insignificant_predictors_log_MAP <- rownames(coeffs_log_MAP)[coeffs_log_MAP[, 4] > 0.05]
cat("Insignificant predictors:", paste(insignificant_predictors_log_MAP, collapse = ", "), "\n")
```

```
## Insignificant predictors: age, trestbps, chol, fbs, restecg, thalach, oldpeak, slope
```

Significant predictors: 1. sex: 1 = MALE, 0 = FEMALE 2. cp (chest pain type): 0 = TYPICAL ANGINA, 1 = ATYPICAL ANGINA, 2 = NON-ANGINAL PAIN, 3 = ASYMPTOMATIC 3. exang (exercise induced angina): 1 = YES, 0 = NO 4. ca (number of major vessels (0-3) colored by flourosopy): number of vessels 5. thal (thalassemia is an inherited blood disorder): 0 = NORMAL, 1 = FIXED DEFECT, 2 = REVERSABLE DEFECT

A new dataset **train_data_MSP** has been created with only significant predicates

```r
train_data_MSP <- train_data %>% select(condition, sex, cp, exang, ca, thal)
```

The Logistic regression with significant predictors (MSP) model is calculated

```r
model_log_MSP <- glm(condition ~ ., data = train_data_MSP, family = binomial)
summary(model_log_MSP)
```

```
##
## Call:
## glm(formula = condition ~ ., family = binomial, data = train_data_MSP)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.9229     0.6444  -6.088 1.14e-09 ***
## sex           1.2421     0.4896   2.537 0.011189 *
## cp            0.5286     0.2081   2.540 0.011084 *
## exang         1.6413     0.4647   3.532 0.000412 ***
## ca            1.1293     0.2510   4.500 6.81e-06 ***
## thal          0.6702     0.2174   3.083 0.002052 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 287.41  on 207  degrees of freedom
## Residual deviance: 166.14  on 202  degrees of freedom
## AIC: 178.14
##
## Number of Fisher Scoring iterations: 5
```

2.1.3.2 Practical application of the logistic regression: Model significant predictors (MSP)

As a result, a formula was obtained by which the presence of the heart disease can be predicted. For practical application and assessment of the probability of a patient having heart disease, instead of the X feature, the value of a specific patient's feature must be substituted into the formula.

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + ... + \beta_5 \cdot X_5)}}$$

where:

$$e - Euler's\ number (e \approx 2.718)$$

$$\beta_0 = -3.9229$$

$$\beta_1 \cdot X_1 = 1.2421 \cdot X_{sex}$$

$$\beta_2 \cdot X_2 = 0.5286 \cdot X_{cp} - chest\ pain\ type\ feauture$$

$$\beta_3 \cdot X_3 = 1.6413 \cdot X_{exang} - exercise\ induced\ angina\ feauture$$

$$\beta_4 \cdot X_4 = 1.1293 \cdot X_{ca} - number\ of\ major\ vessels\ (0 - 3)\ colored\ by\ flourosopy\ feauture$$

$$\beta_5 \cdot X_5 = 0.6702 \cdot X_{thal} - thalassemia\ (an\ inherited\ blood\ disorder)\ feauture$$

2.1.3.3 Efficiency assessment the logistic regression: Model significant predictors (MSP)

```
# Predict probabilities for the test data
predictions_log_MSP <- predict(model_log_MSP, newdata = test_data, type = "response")

# Convert probabilities to binary class predictions
predicted_classes_log_MSP <- ifelse(predictions_log_MSP > 0.5, 1, 0)

# Generate the confusion matrix
confusion_matrix_log_MSP <- table(Predicted = predicted_classes_log_MSP, Actual = test_data$condition)

# Print the confusion matrix
print_confusion_matrix_log_MSP <- paste(
  "The confusion matrix:\n\n",
  paste(capture.output(print(confusion_matrix_log_MSP)), collapse = "\n"), "\n\n",
  sprintf("TN: True Negative (correctly predicted 0)       %d", confusion_matrix_log_MSP[1, 1]), "\n",
  sprintf("FP: False Positive (incorrectly predicted 1)    %d", confusion_matrix_log_MSP[1, 2]), "\n",
  sprintf("FN: False Negative (incorrectly predicted 0)    %d", confusion_matrix_log_MSP[2, 1]), "\n",
  sprintf("TP: True Positive (correctly predicted 1)       %d", confusion_matrix_log_MSP[2, 2]), "\n",
  sep = ""
)

cat(print_confusion_matrix_log_MSP)
```

```
## The confusion matrix:
##
##           Actual
## Predicted  0  1
##         0 40  8
##         1  9 32
##
```

```
## TN: True Negative (correctly predicted 0)       40
## FP: False Positive (incorrectly predicted 1)    8
## FN: False Negative (incorrectly predicted 0)    9
## TP: True Positive (correctly predicted 1)       32
```

```r
# Accuracy
accuracy_log_MSP <- sum(diag(confusion_matrix_log_MSP)) / sum(confusion_matrix_log_MSP)
# Precision
precision_log_MSP <- confusion_matrix_log_MSP[2, 2] / sum(confusion_matrix_log_MSP[2, ])
# Recall
recall_log_MSP <- confusion_matrix_log_MSP[2, 2] / sum(confusion_matrix_log_MSP[, 2])
# F1 Score
f1_score_log_MSP <- 2 * ((precision_log_MSP * recall_log_MSP) / (precision_log_MSP + recall_log_MSP))
# RMSE
rmse_log_MSP <- sqrt(mean((test_data$condition - predictions_log_MSP)^2))
```

2.1.3.4 Result and conclusion on the effectiveness of the model: logistic regression: Model significant predictors (MSP)

```r
results_table <- tibble(
  Models = c("Logistic regression: Model all predictors","Logistic regression: Model significant predic
  RMSE = round(c(rmse_log_MAP,rmse_log_MSP),5),
  Accuracy = round(c(accuracy_log_MAP,accuracy_log_MSP),5),
  Precision = round(c(precision_log_MAP,precision_log_MSP),5),
  Recall = c(recall_log_MAP,recall_log_MSP ),
  F1_Score = round(c(f1_score_log_MAP,f1_score_log_MSP),5)
)
results_table %>% knitr::kable()
```

Models	RMSE	Accuracy	Precision	Recall	F1_Score
Logistic regression: Model all predictors	0.35678	0.82022	0.78571	0.825	0.80488
Logistic regression: Model significant predictors	0.35434	0.80899	0.78049	0.800	0.79012

The model has achieved **Accuracy** 0.80899, which indicates a high ability to classify data correctly as a whole. This shows that the model has correctly classified the 80% of all cases. **The RMSE** value was 0.35434, which indicates a low) error in the predicted probability values. For a positive class, the model showed **Precision** 0.78049, which means that 78% of the cases predicted as positive were indeed positive. **Recall** was 0.800, which shows that the model successfully detected 80% of all real positive cases. **False Negative** (incorrectly predicted 0) accounted for 9 cases or 10% of the total number of observations, which can be critical in assessing the presence of the disease in cases where the patient needs urgent medical care.

The RMSE indicator improved slightly, the indicators **Accuracy**, **Precision**, **Recall** deteriorated slightly, the indicator **False Negative** did not change relative to the Logistic regression model with all predictors.

2.2 Decision Trees

2.2.1 Created the Decision Trees model

A decision tree model was built, within which a hierarchical decision-making structure was created, where each node corresponds to the verification of a certain function. This allows the model to explain why it came to a certain conclusion.

```r
model_tree_MAP <- rpart(condition ~ ., data = train_data, method = "class")
summary(model_tree_MAP)
```

```
## Call:
## rpart(formula = condition ~ ., data = train_data, method = "class")
##   n= 208
##
##           CP nsplit rel error    xerror       xstd
## 1 0.47422680      0 1.0000000 1.0000000 0.07417268
## 2 0.06185567      1 0.5257732 0.6804124 0.06920106
## 3 0.04123711      3 0.4020619 0.5670103 0.06557279
## 4 0.01000000      5 0.3195876 0.5051546 0.06309475
##
## Variable importance
##       ca     thal       cp  oldpeak  thalach      age    exang    slope
##       28       19       14       12       10        8        5        3
## trestbps      sex
##        1        1
##
## Node number 1: 208 observations,    complexity param=0.4742268
##   predicted class=0  expected loss=0.4663462  P(node) =1
##     class counts:   111    97
##    probabilities: 0.534 0.466
##   left son=2 (122 obs) right son=3 (86 obs)
##   Primary splits:
##       ca      < 0.5   to the left,  improve=26.58527, (0 missing)
##       thal    < 0.5   to the left,  improve=25.58940, (0 missing)
##       cp      < 2.5   to the left,  improve=24.63004, (0 missing)
##       exang   < 0.5   to the left,  improve=19.15402, (0 missing)
##       oldpeak < 1.7   to the left,  improve=16.04107, (0 missing)
##   Surrogate splits:
##       age     < 55.5  to the left,  agree=0.678, adj=0.221, (0 split)
##       thalach < 132.5 to the right, agree=0.678, adj=0.221, (0 split)
##       cp      < 2.5   to the left,  agree=0.644, adj=0.140, (0 split)
##       thal    < 1.5   to the left,  agree=0.644, adj=0.140, (0 split)
##       oldpeak < 1.85  to the left,  agree=0.639, adj=0.128, (0 split)
##
## Node number 2: 122 observations,    complexity param=0.06185567
##   predicted class=0  expected loss=0.2540984  P(node) =0.5865385
##     class counts:    91    31
##    probabilities: 0.746 0.254
##   left son=4 (87 obs) right son=5 (35 obs)
##   Primary splits:
##       thal    < 1.5   to the left,  improve=9.884654, (0 missing)
##       exang   < 0.5   to the left,  improve=8.391990, (0 missing)
##       oldpeak < 1.7   to the left,  improve=8.326637, (0 missing)
##       thalach < 160.5 to the right, improve=6.279671, (0 missing)
##       slope   < 0.5   to the left,  improve=5.392713, (0 missing)
##   Surrogate splits:
##       trestbps < 165   to the left,  agree=0.738, adj=0.086, (0 split)
##       exang    < 0.5   to the left,  agree=0.738, adj=0.086, (0 split)
##       oldpeak  < 2.4   to the left,  agree=0.738, adj=0.086, (0 split)
##       thalach  < 114.5 to the right, agree=0.730, adj=0.057, (0 split)
```

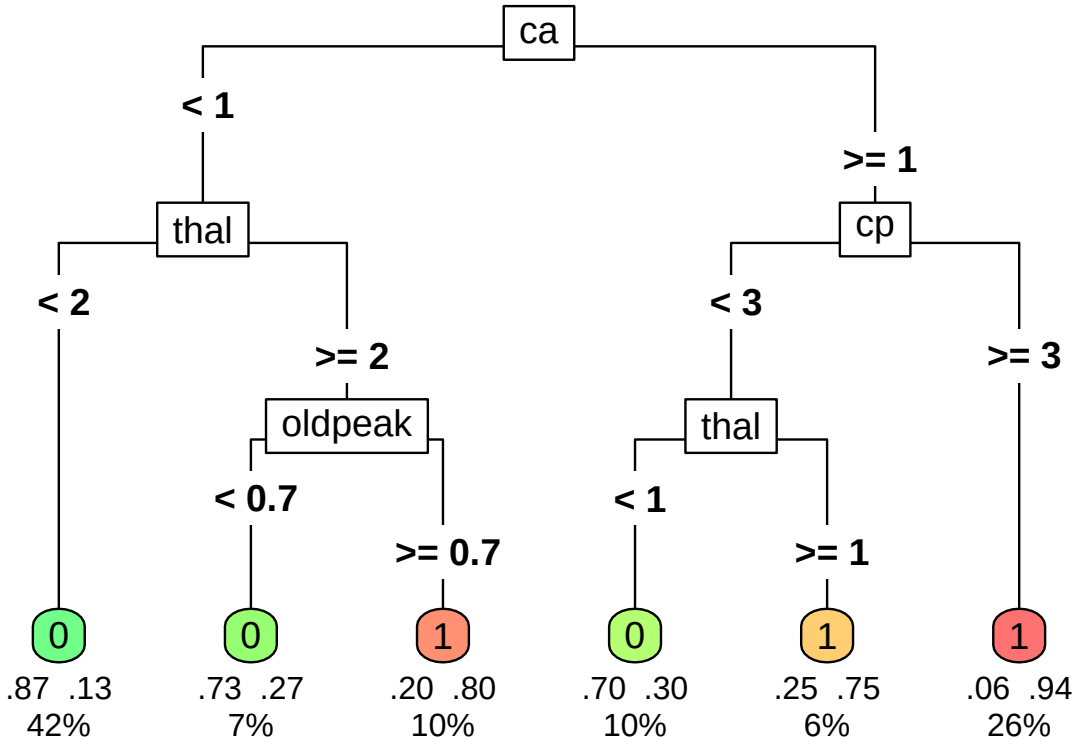
```
##
## Node number 3: 86 observations,    complexity param=0.04123711
##   predicted class=1  expected loss=0.2325581  P(node) =0.4134615
##     class counts:    20    66
##    probabilities: 0.233 0.767
##   left son=6 (32 obs) right son=7 (54 obs)
##   Primary splits:
##       cp      < 2.5   to the left,  improve=9.093508, (0 missing)
##       thal    < 0.5   to the left,  improve=5.277434, (0 missing)
##       oldpeak < 0.85  to the left,  improve=4.193327, (0 missing)
##       slope   < 0.5   to the left,  improve=3.934953, (0 missing)
##       exang   < 0.5   to the left,  improve=3.879897, (0 missing)
##   Surrogate splits:
##       thal    < 0.5   to the left,  agree=0.709, adj=0.219, (0 split)
##       thalach < 145.5 to the right, agree=0.698, adj=0.188, (0 split)
##       exang   < 0.5   to the left,  agree=0.698, adj=0.188, (0 split)
##       oldpeak < 0.85  to the left,  agree=0.698, adj=0.188, (0 split)
##       age     < 67.5  to the right, agree=0.674, adj=0.125, (0 split)
##
## Node number 4: 87 observations
##   predicted class=0  expected loss=0.1264368  P(node) =0.4182692
##     class counts:    76    11
##    probabilities: 0.874 0.126
##
## Node number 5: 35 observations,    complexity param=0.06185567
##   predicted class=1  expected loss=0.4285714  P(node) =0.1682692
##     class counts:    15    20
##    probabilities: 0.429 0.571
##   left son=10 (15 obs) right son=11 (20 obs)
##   Primary splits:
##       oldpeak < 0.7   to the left,  improve=4.876190, (0 missing)
##       slope   < 0.5   to the left,  improve=3.952068, (0 missing)
##       exang   < 0.5   to the left,  improve=3.425752, (0 missing)
##       thalach < 160   to the right, improve=2.862554, (0 missing)
##       cp      < 2.5   to the left,  improve=2.469655, (0 missing)
##   Surrogate splits:
##       slope   < 0.5   to the left,  agree=0.800, adj=0.533, (0 split)
##       exang   < 0.5   to the left,  agree=0.714, adj=0.333, (0 split)
##       thalach < 152.5 to the right, agree=0.657, adj=0.200, (0 split)
##       age     < 55.5  to the right, agree=0.600, adj=0.067, (0 split)
##       cp      < 1.5   to the left,  agree=0.600, adj=0.067, (0 split)
##
## Node number 6: 32 observations,    complexity param=0.04123711
##   predicted class=0  expected loss=0.46875  P(node) =0.1538462
##     class counts:    17    15
##    probabilities: 0.531 0.469
##   left son=12 (20 obs) right son=13 (12 obs)
##   Primary splits:
##       thal  < 0.5   to the left,  improve=3.037500, (0 missing)
##       sex   < 0.5   to the left,  improve=2.101136, (0 missing)
##       slope < 0.5   to the left,  improve=1.508929, (0 missing)
##       chol  < 260.5 to the right, improve=1.035539, (0 missing)
##       age   < 53.5  to the left,  improve=1.020833, (0 missing)
##   Surrogate splits:
```

```
##       thalach < 120.5 to the right, agree=0.719, adj=0.250, (0 split)
##       oldpeak < 0.9   to the left,  agree=0.719, adj=0.250, (0 split)
##       sex     < 0.5   to the left,  agree=0.688, adj=0.167, (0 split)
##       exang   < 0.5   to the left,  agree=0.688, adj=0.167, (0 split)
##       slope   < 0.5   to the left,  agree=0.688, adj=0.167, (0 split)
##
## Node number 7: 54 observations
##   predicted class=1  expected loss=0.05555556  P(node) =0.2596154
##     class counts:     3    51
##    probabilities: 0.056 0.944
##
## Node number 10: 15 observations
##   predicted class=0  expected loss=0.2666667  P(node) =0.07211538
##     class counts:    11     4
##    probabilities: 0.733 0.267
##
## Node number 11: 20 observations
##   predicted class=1  expected loss=0.2  P(node) =0.09615385
##     class counts:     4    16
##    probabilities: 0.200 0.800
##
## Node number 12: 20 observations
##   predicted class=0  expected loss=0.3  P(node) =0.09615385
##     class counts:    14     6
##    probabilities: 0.700 0.300
##
## Node number 13: 12 observations
##   predicted class=1  expected loss=0.25  P(node) =0.05769231
##     class counts:     3     9
##    probabilities: 0.250 0.750
```

2.2.2 Practical application of the Decision Trees model

The decision tree is easier to interpret visually through a graphical representation of the tree structure.

```
rpart.plot::rpart.plot(
  model_tree_MAP,
  type = 5,
  extra = 104,
  under = TRUE,
  tweak = 1.2,
  clip.facs = TRUE,
  box.palette = "GnYlRd",
  branch.lty = 1,
)
```



The factors taken into account in the leaves (end nodes) of the decision tree are: 1. ca (number of major vessels (0-3) colored by flourosopy): number of vessels 2. thal (thalassemia is an inherited blood disorder): 0 = NORMAL, 1 = FIXED DEFECT, 2 = REVERSABLE DEFECT 3. cp (chest pain type): 0 = TYPICAL ANGINA, 1 = ATYPICAL ANGINA, 2 = NON-ANGINAL PAIN, 3 = ASYMPTOMATIC 4. oldpeak (ST depression induced by exercise relative to rest)

Example of leafe analysis:

Node 13: 12 patients: 9 (75%) have the disease. 3 (25%) do not have the disease. The model predicted "sick" with an error of 25%.

This analysis allows the doctor to:

1.Understand the size and characteristics of the patient group. 2.Evaluate the reliability of predictions. 3.Decide to be careful when interpreting this node.

The value of the features in the final leaves can be viewed in the table

```
# Calculating the probability of error
error_rate <- 1 - apply(leaves$yval2[, -1], 1, max)

leaf_table <- tibble(
  "Leaf Number" = rownames(leaves),
  "Features" = node_paths,
  "Observations" = leaves$n,
  "Predicted Class" = if_else(leaves$yval==1,"No Disease","Disease"),
  "Disease Probability" = round(leaves$yval2[, 3] / leaves$n,2),
  "Healthy Probability" = round(leaves$yval2[, 2] / leaves$n,2)
  )
```

```
leaf_table %>% knitr::kable()
```

Leaf Number	Features	Observations	Predicted Class	Disease Probability	Healthy Probability
4	root , ca< 0.5 , thal< 1.5	87	No Disease	0.13	0.87
10	root , ca< 0.5 , thal>=1.5 , oldpeak< 0.7	15	No Disease	0.27	0.73
11	root , ca< 0.5 , thal>=1.5 , oldpeak>=0.7	20	Disease	0.80	0.20
12	root , ca>=0.5 , cp< 2.5 , thal< 0.5	20	No Disease	0.30	0.70
13	root , ca>=0.5 , cp< 2.5 , thal>=0.5	12	Disease	0.75	0.25
7	root , ca>=0.5, cp>=2.5	54	Disease	0.94	0.06

2.2.3 Efficiency assessment the Decision Trees

The confusion matrix:

```
# Predict probabilities for the test data
predictions_tree_MAP <- predict(model_tree_MAP, newdata = test_data, type = "class")

# Convert predictions to numeric
predictions_tree_MAP_numeric <- as.numeric(as.character(predictions_tree_MAP))

# Create the confusion matrix
confusion_matrix_tree_MAP <- table(Predicted = predictions_tree_MAP, Actual = test_data$condition)

# Print the confusion matrix
print_confusion_matrix_tree_MAP <- paste(
  "The confusion matrix:\n\n",
  paste(capture.output(print(confusion_matrix_tree_MAP)), collapse = "\n"), "\n\n",
  sprintf("TN: True Negative (correctly predicted 0)       %d", confusion_matrix_tree_MAP[1, 1]), "\n",
  sprintf("FP: False Positive (incorrectly predicted 1)    %d", confusion_matrix_tree_MAP[1, 2]), "\n",
  sprintf("FN: False Negative (incorrectly predicted 0)    %d", confusion_matrix_tree_MAP[2, 1]), "\n",
  sprintf("TP: True Positive (correctly predicted 1)       %d", confusion_matrix_tree_MAP[2, 2]), "\n",
  sep = ""
)

cat(print_confusion_matrix_tree_MAP)
```

```
## The confusion matrix:
##
##           Actual
## Predicted  0  1
##         0 43  8
##         1  6 32
##
## TN: True Negative (correctly predicted 0)       43
## FP: False Positive (incorrectly predicted 1)    8
## FN: False Negative (incorrectly predicted 0)    6
## TP: True Positive (correctly predicted 1)       32
```

```r
#  Accuracy
accuracy_tree_MAP <- sum(diag(confusion_matrix_tree_MAP)) / sum(confusion_matrix_tree_MAP)
#  Precision
precision_tree_MAP <- confusion_matrix_tree_MAP["1", "1"] / sum(confusion_matrix_tree_MAP["1", ])
#  Recall
recall_tree_MAP <- confusion_matrix_tree_MAP["1", "1"] / sum(confusion_matrix_tree_MAP[, "1"])
#  F1 Score
f1_score_tree_MAP <- 2 * ((precision_tree_MAP * recall_tree_MAP) / (precision_tree_MAP + recall_tree_MAI
# Convert predicted class and condition to numeric
predicted_classes_tree_MAP <- as.numeric(as.character(predict(model_tree_MAP, newdata = test_data, type
true_classes_tree_MAP <- as.numeric(as.character(test_data$condition))
#  RMSE
rmse_tree_MAP <- sqrt(mean((test_data$condition - predictions_tree_MAP_numeric)^2))
```

2.2.4 Result and conclusion on the effectiveness of the Decision Trees:

```r
results_table <- tibble(
  Models = c("Logistic regression: Model all predictors","Logistic regression: Model significant predic
  RMSE = round(c(rmse_log_MAP, rmse_log_MSP, rmse_tree_MAP),5),
  Accuracy = round(c(accuracy_log_MAP, accuracy_log_MSP, accuracy_tree_MAP),5),
  Precision = round(c(precision_log_MAP, precision_log_MSP, precision_tree_MAP),5),
  Recall = c(recall_log_MAP,recall_log_MSP, recall_tree_MAP),
  F1_Score = round(c(f1_score_log_MAP, f1_score_log_MSP, f1_score_tree_MAP),5)
)
results_table %>% knitr::kable()
```

Models	RMSE	Accuracy	Precision	Recall	F1_Score
Logistic regression: Model all predictors	0.35678	0.82022	0.78571	0.825	0.80488
Logistic regression: Model significant predictors	0.35434	0.80899	0.78049	0.800	0.79012
Decision Trees: Model all predictors	0.39661	0.84270	0.84211	0.800	0.82051

The model has achieved **Accuracy** 0.84270, which indicates a high ability to classify data correctly as a whole. This shows that the model has correctly classified the 84% of all cases. **The RMSE** value was 0.39661, which indicates a low) error in the predicted probability values. For a positive class, the model showed **Precision** 0.84211, which means that 84% of the cases predicted as positive were indeed positive. **Recall** was 0.800, which shows that the model successfully detected 80% of all real positive cases. **False Negative** (incorrectly predicted 0) accounted for 6 cases or 7% of the total number of observations, which can be critical in assessing the presence of the disease in cases where the patient needs urgent medical care.

The RMSE has deteriorated , the indicators **Accuracy**, **Precision**, **False Negative** have improved relative to the Logistic regression models.

3. Conclusion

From a practical point of view, **the decision tree** is better suited to understand and explain each step of classification for a particular patient. **Logistic regression** describes risk more mathematically strictly and is suitable for general conclusions about the significance of features. Doctors are advised to use both approaches: a decision tree for visual analysis and logistic regression for detailed risk calculation.

4. References

1. Data set “Heart Disease” available in open source on the website Kaggle
2. Rafael A. Irizarry introduction to Data Science
3. R for Data Science (2e)
4. R Documentation
5. Authoring Books and Technical Documents with R Markdown
6. AHEM Maas, YEA Appelman. Gender differences in coronary heart disease
7. Julie Corliss, The heart disease gender gap