statproject

Deekshitha Reddy Kalluru

2024-12-10

# Load necessary libraries  
library(ggplot2) # For data visualization

## Warning: package 'ggplot2' was built under R version 4.4.2

library(rpart) # For decision trees (optional, not used directly here)  
library(rpart.plot) # For plotting decision trees (optional, not used directly here)

## Warning: package 'rpart.plot' was built under R version 4.4.2

library(caret) # For model evaluation (e.g., confusion matrix, accuracy)

## Loading required package: lattice

library(readxl) # For reading Excel files  
library(e1071) # For SVM and other machine learning models (optional, not used directly here)

## Warning: package 'e1071' was built under R version 4.4.2

# Load the dataset from an Excel file  
data <- read\_excel("dataset.xlsx")  
  
# View the cleaned dataset to ensure it's in the correct format  
head(data)

## # A tibble: 6 × 12  
## age sex chest\_pain resting\_bp cholestrol fasting\_bloodsugar resting\_ecg  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 40 1 2 140 289 0 0  
## 2 49 0 3 160 180 0 0  
## 3 37 1 2 130 283 0 1  
## 4 48 0 4 138 214 0 0  
## 5 54 1 3 150 195 0 0  
## 6 39 1 3 120 339 0 0  
## # ℹ 5 more variables: max\_heartrate <dbl>, exercise\_angina <dbl>,  
## # oldpeak <dbl>, ST\_slope <dbl>, `target(presence of heart disease)` <dbl>

# Display the column names to understand the dataset structure  
colnames(data)

## [1] "age" "sex"   
## [3] "chest\_pain" "resting\_bp"   
## [5] "cholestrol" "fasting\_bloodsugar"   
## [7] "resting\_ecg" "max\_heartrate"   
## [9] "exercise\_angina" "oldpeak"   
## [11] "ST\_slope" "target(presence of heart disease)"

# Rename the 12th column to "target" (assuming this is the target variable indicating heart disease)  
colnames(data)[c(12)] <- c("target")  
  
# Check if the column name was updated correctly  
colnames(data)

## [1] "age" "sex" "chest\_pain"   
## [4] "resting\_bp" "cholestrol" "fasting\_bloodsugar"  
## [7] "resting\_ecg" "max\_heartrate" "exercise\_angina"   
## [10] "oldpeak" "ST\_slope" "target"

# Check for missing values in the relevant columns: 'age', 'cholestrol', 'max\_heartrate', and 'target'  
# This helps us understand if any cleaning is needed before analysis  
colSums(is.na(data[c("age", "cholestrol", "max\_heartrate", "target")]))

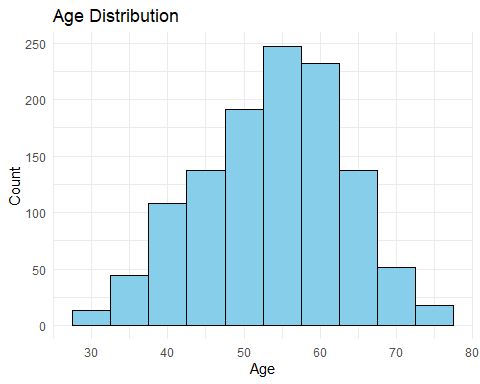
## age cholestrol max\_heartrate target   
## 0 0 0 0

# Remove rows with NA (missing values) in the relevant columns to ensure clean data for modeling  
# We only keep rows where 'age', 'cholestrol', 'max\_heartrate', and 'target' are not missing  
clean\_data <- na.omit(data[c("age", "cholestrol", "max\_heartrate", "target")])  
  
# Display the cleaned dataset  
head(clean\_data)

## # A tibble: 6 × 4  
## age cholestrol max\_heartrate target  
## <dbl> <dbl> <dbl> <dbl>  
## 1 40 289 172 0  
## 2 49 180 156 1  
## 3 37 283 98 0  
## 4 48 214 108 1  
## 5 54 195 122 0  
## 6 39 339 170 0

#BASIC VISSUALIZATIONS

#Age Distribution  
ggplot(data, aes(x = age)) +  
 geom\_histogram(binwidth = 5, fill = "skyblue", color = "black") +  
 labs(title = "Age Distribution", x = "Age", y = "Count") +  
 theme\_minimal()



#RANDOM FOREST #1. Can age, cholesterol levels, and maximum heart rate predict the likelihood of heart disease?

# Load necessary libraries  
library(randomForest)

## Warning: package 'randomForest' was built under R version 4.4.2

## randomForest 4.7-1.2

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

##   
## Attaching package: 'randomForest'

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

library(caret)  
library(ggplot2)  
  
# Assuming 'clean\_data' is your dataset, with columns 'age', 'cholestrol', 'max\_heartrate', and 'target'  
  
# 1. Ensure the 'target' variable is a factor (for classification)  
clean\_data$target <- factor(clean\_data$target, levels = c(0, 1))  
  
# 2. Split the data into training and testing sets  
set.seed(123) # For reproducibility  
train\_index <- createDataPartition(clean\_data$target, p = 0.8, list = FALSE)  
train\_data <- clean\_data[train\_index, ]  
test\_data <- clean\_data[-train\_index, ]  
  
# 3. Fit a Random Forest model (ensure it's a classification model)  
rf\_model <- randomForest(target ~ age + cholestrol + max\_heartrate, data = train\_data, importance = TRUE, ntree = 100)  
  
# 4. Make predictions using the random forest model  
rf\_predictions <- predict(rf\_model, newdata = test\_data)  
  
# 5. Ensure the predicted values and actual target values match  
rf\_predictions <- factor(rf\_predictions, levels = c(0, 1))  
actual <- factor(test\_data$target, levels = c(0, 1))  
  
# 6. Evaluate the model using confusion matrix and accuracy  
cm <- confusionMatrix(rf\_predictions, actual)  
accuracy <- cm$overall["Accuracy"]  
  
# Print the accuracy and confusion matrix for the Random Forest model  
cat("\nRandom Forest Model\n")

##   
## Random Forest Model

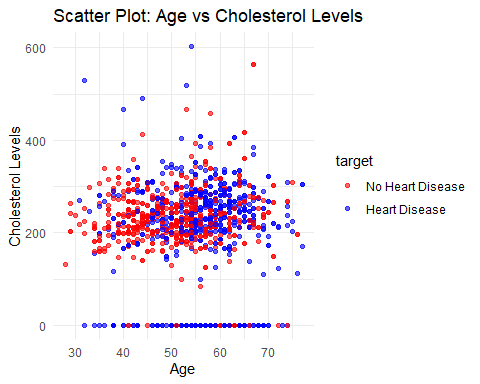
cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.8333333

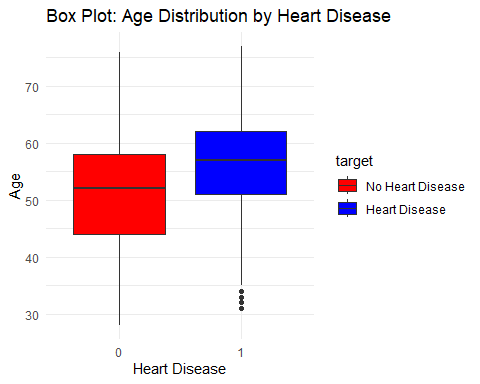
print(cm)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 88 16  
## 1 23 107  
##   
## Accuracy : 0.8333   
## 95% CI : (0.7793, 0.8787)  
## No Information Rate : 0.5256   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.6648   
##   
## Mcnemar's Test P-Value : 0.3367   
##   
## Sensitivity : 0.7928   
## Specificity : 0.8699   
## Pos Pred Value : 0.8462   
## Neg Pred Value : 0.8231   
## Prevalence : 0.4744   
## Detection Rate : 0.3761   
## Detection Prevalence : 0.4444   
## Balanced Accuracy : 0.8314   
##   
## 'Positive' Class : 0   
##

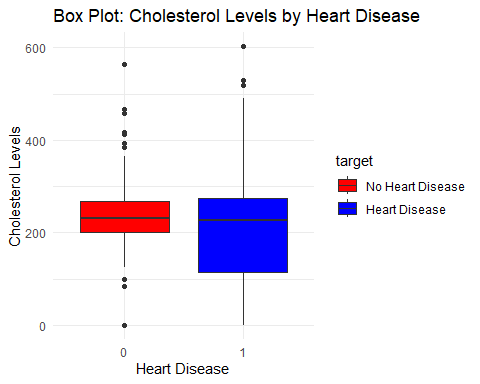
# 7. Visualize the relationship between each predictor and the likelihood of heart disease  
  
# Scatter plots for continuous variables  
ggplot(clean\_data, aes(x = age, y = cholestrol, color = target)) +  
 geom\_point(alpha = 0.6) +  
 labs(title = "Scatter Plot: Age vs Cholesterol Levels",  
 x = "Age", y = "Cholesterol Levels") +  
 theme\_minimal() +  
 scale\_color\_manual(values = c("red", "blue"), labels = c("No Heart Disease", "Heart Disease"))



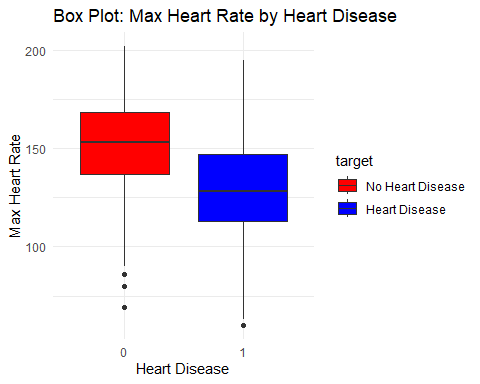
# Box plots to show feature distributions by heart disease status  
ggplot(clean\_data, aes(x = target, y = age, fill = target)) +  
 geom\_boxplot() +  
 labs(title = "Box Plot: Age Distribution by Heart Disease",  
 x = "Heart Disease", y = "Age") +  
 scale\_fill\_manual(values = c("red", "blue"), labels = c("No Heart Disease", "Heart Disease")) +  
 theme\_minimal()



ggplot(clean\_data, aes(x = target, y = cholestrol, fill = target)) +  
 geom\_boxplot() +  
 labs(title = "Box Plot: Cholesterol Levels by Heart Disease",  
 x = "Heart Disease", y = "Cholesterol Levels") +  
 scale\_fill\_manual(values = c("red", "blue"), labels = c("No Heart Disease", "Heart Disease")) +  
 theme\_minimal()



ggplot(clean\_data, aes(x = target, y = max\_heartrate, fill = target)) +  
 geom\_boxplot() +  
 labs(title = "Box Plot: Max Heart Rate by Heart Disease",  
 x = "Heart Disease", y = "Max Heart Rate") +  
 scale\_fill\_manual(values = c("red", "blue"), labels = c("No Heart Disease", "Heart Disease")) +  
 theme\_minimal()



#2. Is there a significant association between chest pain type and the presence of heart disease?

# Check for any missing values in the relevant columns  
sum(is.na(data$chest\_pain)) # Check for NAs in chest\_pain column

## [1] 0

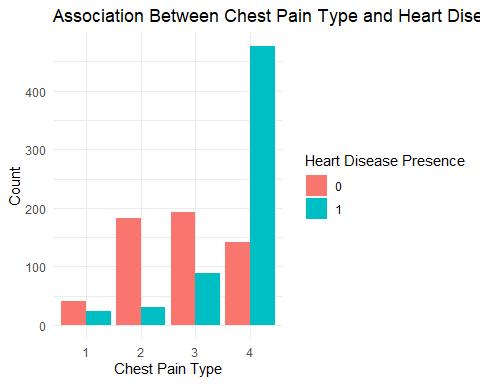
sum(is.na(data$target)) # Check for NAs in target column

## [1] 0

# If any NAs are found, remove them  
data <- na.omit(data)  
  
# Relevel the factors to ensure no extra levels like NA are present  
data$chest\_pain <- factor(data$chest\_pain, levels = c(1, 2, 3, 4)) # Adjust to match your chest pain type categories  
data$target <- factor(data$target, levels = c(0, 1)) # Adjust to match your target variable categories  
  
# Create a contingency table of chest pain type and heart disease presence (target variable)  
chisq\_test <- chisq.test(table(data$chest\_pain, data$target))  
  
# Print the results of the Chi-squared test  
print(chisq\_test)

##   
## Pearson's Chi-squared test  
##   
## data: table(data$chest\_pain, data$target)  
## X-squared = 331.02, df = 3, p-value < 2.2e-16

# Visualize the association between chest pain type and heart disease using a bar plot  
library(ggplot2)  
  
ggplot(data, aes(x = as.factor(chest\_pain), fill = as.factor(target))) +  
 geom\_bar(position = "dodge") +  
 labs(x = "Chest Pain Type", fill = "Heart Disease Presence", y = "Count",  
 title = "Association Between Chest Pain Type and Heart Disease") +  
 theme\_minimal() # Use a minimal theme for the plot



# Fit a logistic regression model with 'max\_heartrate' as the predictor and 'target' as the response variable  
# This model will help us understand if there is any significant threshold for maximum heart rate influencing the risk of heart disease  
threshold\_analysis <- glm(target ~ max\_heartrate, data = data, family = "binomial")  
  
# Display the summary of the logistic regression model for threshold analysis  
# This will show the coefficient for 'max\_heartrate' and its statistical significance  
summary(threshold\_analysis)

##   
## Call:  
## glm(formula = target ~ max\_heartrate, family = "binomial", data = data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.476387 0.415784 13.17 <2e-16 \*\*\*  
## max\_heartrate -0.038233 0.002904 -13.16 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1630.0 on 1177 degrees of freedom  
## Residual deviance: 1412.3 on 1176 degrees of freedom  
## AIC: 1416.3  
##   
## Number of Fisher Scoring iterations: 4

# Create a threshold for maximum heart rate (e.g., categorizing it into "Above 140" and "Below 140")  
data$threshold\_category <- ifelse(data$max\_heartrate < 140, "Below 140", "Above 140")  
  
# Fit the logistic regression model again using the new categorical variable 'threshold\_category'  
# This will help assess if having a maximum heart rate below 140 bpm significantly increases the likelihood of heart disease  
threshold\_model <- glm(target ~ threshold\_category, data = data, family = "binomial")  
  
# Display the summary of the threshold model  
# The coefficients will show if the "Below 140" category has a higher risk compared to the "Above 140" category  
summary(threshold\_model)

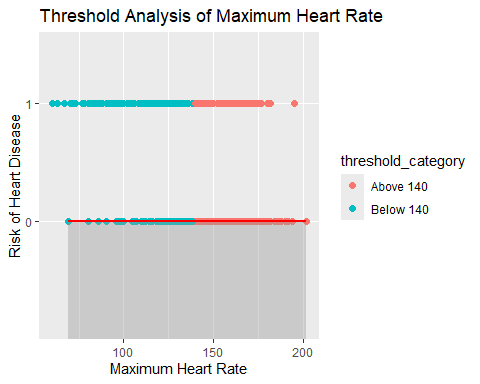
##   
## Call:  
## glm(formula = target ~ threshold\_category, family = "binomial",   
## data = data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.54971 0.08245 -6.667 2.61e-11 \*\*\*  
## threshold\_categoryBelow 140 1.45189 0.12550 11.568 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1630.0 on 1177 degrees of freedom  
## Residual deviance: 1486.5 on 1176 degrees of freedom  
## AIC: 1490.5  
##   
## Number of Fisher Scoring iterations: 4

# Plot the predicted probability of heart disease as a function of maximum heart rate with respect to the threshold  
# This will visualize how the likelihood of heart disease changes across different ranges of heart rate  
ggplot(data, aes(x = max\_heartrate, y = target)) +  
 geom\_point(aes(color = threshold\_category), size = 2) + # Points colored by threshold category  
 stat\_smooth(method = "glm", method.args = list(family = "binomial"), color = "red") + # Fit logistic regression line  
 labs(title = "Threshold Analysis of Maximum Heart Rate",  
 x = "Maximum Heart Rate",  
 y = "Risk of Heart Disease") # Removed the theme\_minimal() for a default theme

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: glm.fit: algorithm did not converge

## Warning: Failed to fit group 2.  
## Caused by error:  
## ! y values must be 0 <= y <= 1



# Split data into groups based on max\_heartrate  
threshold\_analysis <- glm(target ~ max\_heartrate, data = data, family = "binomial")  
# Summary of the logistic regression model  
summary(threshold\_analysis)

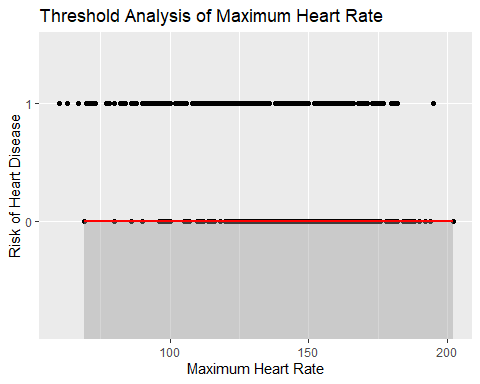
##   
## Call:  
## glm(formula = target ~ max\_heartrate, family = "binomial", data = data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 5.476387 0.415784 13.17 <2e-16 \*\*\*  
## max\_heartrate -0.038233 0.002904 -13.16 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1630.0 on 1177 degrees of freedom  
## Residual deviance: 1412.3 on 1176 degrees of freedom  
## AIC: 1416.3  
##   
## Number of Fisher Scoring iterations: 4

# Plot risk against max\_heartrate  
ggplot(data, aes(x = max\_heartrate, y = target)) +  
 geom\_point() +  
 stat\_smooth(method = "glm", method.args = list(family = "binomial"), color = "red") +  
 labs(title = "Threshold Analysis of Maximum Heart Rate",  
 x = "Maximum Heart Rate",  
 y = "Risk of Heart Disease")

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: glm.fit: algorithm did not converge

## Warning: Failed to fit group 2.  
## Caused by error:  
## ! y values must be 0 <= y <= 1



#3. What is the combined effect of resting blood pressure and cholesterol levels on the likelihood of heart disease?

# Load necessary libraries  
library(ggplot2)  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.2

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

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

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

# Assuming 'data' is your dataset  
# Convert target to a factor for classification  
data$target <- factor(data$target, levels = c(0, 1))  
  
# Fit a logistic regression model  
logistic\_model <- glm(target ~ resting\_bp + cholestrol,   
 data = data,   
 family = binomial(link = "logit"))  
  
# Summary of the logistic regression model (for p-values and confidence intervals)  
summary(logistic\_model)

##   
## Call:  
## glm(formula = target ~ resting\_bp + cholestrol, family = binomial(link = "logit"),   
## data = data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.1255089 0.4532112 -2.483 0.013 \*   
## resting\_bp 0.0167095 0.0034208 4.885 1.04e-06 \*\*\*  
## cholestrol -0.0045934 0.0006473 -7.096 1.28e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1630.0 on 1177 degrees of freedom  
## Residual deviance: 1557.1 on 1175 degrees of freedom  
## AIC: 1563.1  
##   
## Number of Fisher Scoring iterations: 4

confint(logistic\_model) # Confidence intervals for coefficients

## Waiting for profiling to be done...

## 2.5 % 97.5 %  
## (Intercept) -2.021360096 -0.24292866  
## resting\_bp 0.010078235 0.02350095  
## cholestrol -0.005885464 -0.00334525

# Predict probabilities and classes  
data <- data %>%  
 mutate(predicted\_prob = predict(logistic\_model, newdata = data, type = "response"),  
 predicted\_class = ifelse(predicted\_prob > 0.5, 1, 0)) # Assign classes based on 0.5 threshold  
  
# Bin the continuous variables (resting\_bp and cholestrol) into categories for grouping  
data <- data %>%  
 mutate(bp\_category = cut(resting\_bp, breaks = c(0, 90, 120, 140, 160, Inf),   
 labels = c("Low", "Normal", "Elevated", "High", "Very High")),  
 chol\_category = cut(cholestrol, breaks = c(0, 200, 240, 300, Inf),   
 labels = c("Desirable", "Borderline High", "High", "Very High")))  
  
  
  
# Assuming 'data' is the dataset  
# Ensure target is a factor and that resting\_bp and cholestrol are properly binned  
data$target <- factor(data$target, levels = c(0, 1))  
  
# Remove NA values and group by BP and Cholesterol categories  
heatmap\_data <- data %>%  
 filter(!is.na(resting\_bp) & !is.na(cholestrol)) %>% # Remove rows with NA in resting\_bp and cholestrol  
 mutate(bp\_category = cut(resting\_bp, breaks = c(0, 90, 120, 140, 160, Inf),   
 labels = c("Low", "Normal", "Elevated", "High", "Very High")),  
 chol\_category = cut(cholestrol, breaks = c(0, 200, 240, 300, Inf),   
 labels = c("Desirable", "Borderline High", "High", "Very High"))) %>%  
 filter(!is.na(bp\_category) & !is.na(chol\_category)) %>% # Ensure newly created categories have no NAs  
 group\_by(bp\_category, chol\_category) %>%  
 summarise(actual = sum(as.numeric(target) == 1),  
 predicted = sum(predicted\_class == 1),  
 total = n(),  
 .groups = "drop") %>%  
 mutate(predicted\_percentage = predicted / total \* 100)  
  
# Create the heatmap  
ggplot(heatmap\_data, aes(x = bp\_category, y = chol\_category, fill = predicted\_percentage)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient(low = "lightblue", high = "red", name = "Predicted %") +  
 labs(title = "Heatmap: Predicted Heart Disease by BP and Cholesterol Categories",  
 x = "Resting BP Category",  
 y = "Cholesterol Category") +  
 theme\_minimal()

