**1. Load Necessary Packages**

library(tidyverse)

library(broom)

library(Metrics)

* **tidyverse**: A collection of R packages for data manipulation, visualization, and analysis.
* **broom**: Converts statistical analysis objects into tidy data frames.
* **Metrics**: Provides functions to calculate model performance metrics like accuracy and AUC.

**2. Load the Dataset**

hd\_data <- read.csv("C:/Users/Nelson/Downloads/Cleveland\_hd (2).csv")

head(hd\_data, 5)

* Reads the CSV file into a data frame called hd\_data.
* head(hd\_data, 5) displays the first 5 rows to check the data loaded correctly.

**Your file** (Heartrate.csv) has columns like: age, gender, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, class1.

**3. Data Cleaning and Transformation**

**Create Binary Outcome Variable**

hd\_data <- hd\_data %>%

mutate(hd = ifelse(class > 0, 1, 0))

* Creates a new column hd:
  + hd = 1 if class > 0 (any heart disease present)
  + hd = 0 if class == 0 (no disease)

**Recode Gender from Numeric to Factor**

hd\_data <- hd\_data %>%

mutate(gender = factor(sex, levels = 0:1, labels = c("Female", "Male")))

* Converts the sex column (0 = Female, 1 = Male) to a categorical variable called gender with labels "Female" and "Male".
* **Note:** In your file, the column is named gender, not sex. You should use gender instead:

mutate(gender = factor(gender, levels = 0:1, labels = c("Female", "Male")))

**4. Statistical Tests**

**Chi-Squared Test: Gender vs Disease**

hd\_gender <- chisq.test(hd\_data$hd, hd\_data$gender)

* Tests if there is a significant association between gender and heart disease (hd).

**T-Test: Age vs Disease**

hd\_age <- t.test(age ~ hd, data = hd\_data)

* Compares the mean age between those with and without heart disease.

**T-Test: Max Heart Rate (thalach) vs Disease**

hd\_heartrate <- t.test(thalach ~ hd, data = hd\_data)

* Compares the mean max heart rate (thalach) between those with and without heart disease.

**Print Test Results**

print(hd\_gender)

print(hd\_age)

print(hd\_heartrate)

* Displays the results of the above tests.

**5. Label Outcome for Plotting**

hd\_data <- hd\_data %>%

mutate(hd\_labelled = ifelse(hd == 0, "No Disease", "Disease"))

* Adds a new column hd\_labelled for use in plots: "No Disease" or "Disease".

**6. Visualizations**

**Boxplot: Age vs Disease**

ggplot(data = hd\_data, aes(x = hd\_labelled, y = age)) +

geom\_boxplot() +

ggtitle("Age vs Heart Disease")

* Shows the distribution of age for each disease group.

**Barplot: Gender vs Disease**

ggplot(data = hd\_data, aes(x = hd\_labelled, fill = gender)) +

geom\_bar(position = "fill") +

ylab("Proportion") +

ggtitle("Gender Distribution by Heart Disease")

* Shows the proportion of males and females in each disease group.

**Boxplot: Max Heart Rate vs Disease**

ggplot(data = hd\_data, aes(x = hd\_labelled, y = thalach)) +

geom\_boxplot() +

ggtitle("Max Heart Rate vs Heart Disease")

* Shows the distribution of max heart rate (thalach) for each disease group.

**7. Logistic Regression Model**

**Fit the Model**

model <- glm(hd ~ age + gender + thalach, data = hd\_data, family = "binomial")

* Fits a logistic regression model predicting heart disease (hd) using age, gender, and max heart rate.

**Model Summary**

print(summary(model))

* Prints coefficients, standard errors, and significance for each predictor.

**Tidy Model Output and Compute Odds Ratios**

tidy\_m <- tidy(model)

tidy\_m$OR <- exp(tidy\_m$estimate)

tidy\_m$lower\_CI <- exp(tidy\_m$estimate - 1.96 \* tidy\_m$std.error)

tidy\_m$upper\_CI <- exp(tidy\_m$estimate + 1.96 \* tidy\_m$std.error)

print(tidy\_m)

* tidy(model): Converts model summary to a tidy data frame.
* Calculates Odds Ratios (OR) and 95% Confidence Intervals for each predictor.

**8. Predict and Classify**

**Predicted Probabilities**

pred\_prob <- predict(model, hd\_data, type = "response")

* Predicts the probability of heart disease for each patient.

**Classification Using 0.5 Cutoff**

hd\_data$pred\_hd <- ifelse(pred\_prob >= 0.5, 1, 0)

* Classifies as 1 (disease) if predicted probability ≥ 0.5, else 0.

**Predict for a New Patient**

newdata <- data.frame(age = 45, gender = "Female", thalach = 150)

p\_new <- predict(model, newdata, type = "response")

print(paste("Predicted probability for new case:", round(p\_new, 3)))

* Predicts the probability of heart disease for a new patient (age 45, female, thalach 150).

**9. Model Evaluation**

**AUC, Accuracy, Classification Error**

auc <- auc(hd\_data$hd, hd\_data$pred\_hd)

accuracy <- accuracy(hd\_data$hd, hd\_data$pred\_hd)

classification\_error <- ce(hd\_data$hd, hd\_data$pred\_hd)

print(paste("AUC =", round(auc, 3)))

print(paste("Accuracy =", round(accuracy, 3)))

print(paste("Classification Error =", round(classification\_error, 3)))

* **AUC**: Measures the model's ability to distinguish between classes.
* **Accuracy**: Proportion of correct predictions.
* **Classification Error**: Proportion of incorrect predictions.

**Confusion Matrix**

print(table(hd\_data$hd, hd\_data$pred\_hd, dnn = c("True Status", "Predicted Status")))

* Shows the count of true positives, false positives, true negatives, and false negatives.

**Summary Table: Key Columns Used**

| **Variable** | **Meaning** | **Example Value** |
| --- | --- | --- |
| age | Age of patient | 63 |
| gender | 0 = Female, 1 = Male (needs recoding) | 1 |
| thalach | Maximum heart rate achieved | 150 |
| class | Heart disease diagnosis (0 = no disease, >0 = disease) | 0, 1, 2, ... |
| hd | Binary outcome (created in code) | 0 or 1 |