HEART DISEASE PREDICTION USING LOGISTIC REGRESSION IN R

Brief explanation of the R code:

1. Load Essential Libraries:
    - Import the necessary libraries for data manipulation, visualization, and modeling: ggplot2, gridExtra, reshape2, grid, DT, dplyr, and cowplot.
2. Load and Examine Data:
    - Load the heart disease dataset from the CSV file: data <- read.csv("C:\\dataset\\HeartFailure\\heart.csv")
    - Explore the first few rows and structure of the data: head(data, n=5), str(data)
    - Check for missing values: missing_val <- colSums(is.na(data))
3. Data Visualization:
    - Create a custom theme for consistent plot styling.
    - Generate a histogram of patient ages: ggplot(data, aes(x=Age)) + geom_histogram(...)
    - Create a bar plot showing the distribution of heart disease cases: ggplot(heart_counts_df, aes(x=Disease_Status, y=Count, fill=Disease_Status)) + geom_bar(...)
    - Create a pie chart illustrating gender distribution: pie(category_counts, labels=..., col=custom_colors, main="Pie Chart of Gender")
4. Prepare Data for Modeling:
    - Separate features (X) from the target variable (y): X <- select(data, -'HeartDisease'), y <- data$'HeartDisease'
    - Split data into training (80%) and testing (20%) sets: sample_indices <- sample(...), train_data <- data[sample_indices,], test_data <- data[-sample_indices,]
5. Build Logistic Regression Model:
    - Train a logistic regression model using the training data: model <- glm(HeartDisease ~ ., data=train_data, family=binomial)
6. Evaluate Model Performance:
    - Generate predictions on the testing set: predictions <- predict(model, newdata=test_data, type="response")
    - Apply a threshold (0.5) to convert probabilities to binary predictions: predicted_classes <- ifelse(predictions > threshold, 1, 0)
    - Calculate accuracy using a confusion matrix: accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
7. Make New Predictions:
    - Create a new dataset with patient information for prediction: new_data <- data.frame(...)
    - Generate predictions for the new dataset: new_predictions <- predict(model, newdata=new_data, type="response")
    - Convert probabilities to binary predictions using the threshold: new_predicted_classes <- ifelse(new_predictions > threshold, 1, 0)