Major Project

(23ONMCR-753)

of the programme

Master of Computer Applications

Batch - Jul 2023 Fourth Semester

Heart Rate Prediction System Using Machine Learning for Early Heart Attack Risk Detection

Submitted by

GURPREET KOUR

MCA July 2023 batch

UID-O23MCA110191

Project Synopsis

Title:

Heart Attack Prediction Analysis using Machine Learning

Objective:

The primary objective of this project is to develop an efficient, accurate, and interpretable machine learning model capable of predicting the risk of heart attacks in individuals. By analyzing patient health records and identifying key physiological and lifestyle-related risk factors, the model aims to aid in early diagnosis, prevention, and clinical decision-making. The project also emphasizes gender-specific analysis to highlight differences in heart attack symptoms and prediction accuracy between men and women.

Introduction:

Heart disease continues to be a leading cause of mortality worldwide, with heart attacks (myocardial infarctions) responsible for a significant proportion of deaths. Despite advancements in medical diagnostics, many heart attacks go undetected or are diagnosed too late. Machine learning provides a promising approach to analyze vast amounts of health-related data and identify patterns that may be invisible to traditional statistical techniques. By training predictive models on historical health data, this project seeks to build a tool that can forecast the risk of a heart attack before critical symptoms emerge.

Scope of the Project:

Utilization of publicly available datasets (such as the UCI Heart Disease dataset) for model training and validation.

Preprocessing of data including handling missing values, outlier detection, normalization, and encoding.

Comprehensive Exploratory Data Analysis (EDA) to uncover data patterns, relationships, and anomalies.

Gender-based analysis to capture and interpret symptom variation and risk profiles.

Development of visual dashboards for risk prediction and model explainability (e.g., using SHAP and LIME).

Planning for deployment as a web or mobile application integrated into clinical settings.

Tools and Technologies:

Programming Language: Python

Libraries and Frameworks: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, XGBoost, SHAP, LIME

Data Visualization: Tableau, Plotly, Seaborn, Matplotlib

Development Environment: Jupyter Notebook, VS Code

Version Control: Git and GitHub

Data Sources: UCI Machine Learning Repository, Kaggle Heart Disease datasets

Deployment (Optional): Streamlit, Flask, or Django for frontend/backend integration

Methodology (SDLC):

Requirement Analysis: Define project goals, stakeholders, data requirements, and success criteria.

Data Collection: Acquire relevant and high-quality datasets from open sources.

Data Preprocessing: Clean and transform data, handle missing values, normalize features.

Exploratory Data Analysis (EDA): Use visualizations and summary statistics to understand data distribution and relationships.

Feature Engineering: Select relevant features and create new ones as needed.

Model Development: Train and evaluate multiple algorithms for performance comparison.

Model Evaluation: Use classification metrics and cross-validation to validate results.

Model Interpretation: Use SHAP/LIME for transparency and explainability.

Deployment Planning: Explore options to deploy the model in clinical or research environments.

Documentation and Reporting: Document all phases, decisions, and findings.

Key Techniques and Tools in EDA:

Descriptive Statistics: Measures of central tendency (mean, median, mode), measures of variability (standard deviation, variance, range), and measures of shape (skewness, kurtosis).

Data Visualization: Histograms, box plots, scatter plots, bar charts, heatmaps, and pair plots to visualize distributions, relationships, and patterns.

Correlation Analysis: Correlation matrices and scatter plot matrices to understand relationships between variables.

Univariate and Multivariate Analysis: Analyzing single variables (univariate) and relationships between multiple variables (multivariate).

Proposed Methodology:

The project will follow a modular and iterative approach, beginning with data acquisition and preprocessing. EDA will provide insights to guide feature engineering. Multiple models will be trained and evaluated, with hyperparameter tuning for optimization. SHAP and LIME will be used to explain model predictions. The best-performing model will be integrated into a user interface, providing a practical tool for heart attack prediction.

Table of Contents

Certificate

This is to certify that the Major project on titled **“**Heart Rate Prediction System Using Machine Learning for Early Heart Attack Risk Detection**”** is a Project work done by **“Gurpreet Kour”** submitted in the partial fulfillment of the requirement for the award of the degree of **“**Master of Computer Applications**”** from **“CHANDIGARH UNIVERSITY”** under my guidance and direction.

To the best of my knowledge and belief, the data and information presented by her in the project has not been submitted elsewhere.

Gurpreet Kour

Acknowledgement

I would like to express my sincere gratitude to my project guide for their continuous support, guidance, and valuable insights throughout the development of this project. Their expertise in both medical science and machine learning was instrumental in shaping this work.

I also extend my thanks to the faculty and staff of **Chandigarh University** for providing the necessary resources and encouragement. Special thanks to researchers and healthcare professionals whose extensive studies on myocardial infarction and cardiovascular diseases formed the foundation of this analysis.

Finally, I am grateful to my family and friends for their motivation and patience during this endeavor.

Abstract

Heart attacks (myocardial infarctions) remain a leading cause of death globally, driven primarily by atherosclerosis and influenced by various medical and lifestyle risk factors. This project aims to develop a machine learning-based predictive model for heart attack risk analysis by leveraging clinical data encompassing demographic, physiological, and lifestyle features. The study incorporates key risk factors such as age, gender, cholesterol levels, blood pressure, diabetes, smoking habits, and physical activity. Additionally, it addresses gender-specific symptom variations to enhance prediction accuracy and early detection. Advanced machine learning algorithms are employed to identify patterns and predict the likelihood of myocardial infarction. The integration of clinical insights with AI techniques demonstrates significant potential for improving cardiovascular risk stratification, enabling proactive healthcare interventions and personalized treatment strategies.

Introduction

Cardiovascular diseases (CVDs) remain the leading cause of mortality globally, accounting for nearly 17.9 million deaths each year. A Early prediction and timely intervention are critical in reducing the fatality rate and improving patient outcomes.

This project, titled **"Heart Attack Prediction Analysis using Machine Learning"**, aims to explore the potential of data-driven approaches in accurately predicting the likelihood of heart attacks based on key medical and lifestyle indicators. By leveraging historical health data and applying various machine learning algorithms, this study attempts to identify patterns and risk factors that contribute to heart attacks.

The project follows a structured approach: data collection and preprocessing, exploratory data analysis, model selection and training, evaluation using performance metrics, and interpretation of results. Several classification algorithms—including Logistic Regression, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—are applied and compared to determine the most effective model for heart attack prediction.

The full project documentation and source code are available on GitHub at:

🔗 https://github.com/Gurpreet9096/ML-heartattack-prediction-repo

Ultimately, this project highlights how machine learning can serve as a powerful tool in the medical domain, improving diagnosis, reducing human error, and paving the way for more accurate and efficient healthcare solutions.

Software Development Life Cycle (SDLC) of Project

1. Requirement Gathering and Analysis

Objective: Understand project goals, stakeholders, and data requirements for building a heart attack prediction model.

- **Activities:**
  - Define scope: Predict heart attack risk using clinical and lifestyle data.
  - Identify key features based on medical knowledge (age, gender, cholesterol, blood pressure, symptoms, etc.).
  - Discuss compliance & privacy (HIPAA/GDPR if applicable).
  - Determine success criteria and evaluation metrics.
- **Tools/Technologies:**
  - Documentation tools: Microsoft Word, Google Docs
  - Collaboration: Jira, Confluence, Trello
  - Communication: Zoom, Microsoft Teams

Diagram:

Use a Stakeholder & Requirement Diagram

less Copy Edit

```
[Stakeholders] ---> [Requirement Analysis] ---> [Project Goals]
Doctors             |                    |
Data Scientists     |                    |--> Define Scope, Data Needs
Patients            |                    |
```

2. Data Collection

Objective: Gather relevant datasets containing patient demographics, clinical measurements, and lifestyle factors.

- **Activities:**
  - Acquire data from hospitals, medical databases, or public sources (e.g., UCI Heart Disease Dataset).
  - Verify data completeness and relevance.
- **Tools/Technologies:**
  - Data sources: Kaggle, UCI Machine Learning Repository, hospital EMRs
  - Data transfer: APIs, SQL queries, FTP
  - Storage: AWS S3, Azure Blob Storage, Google Cloud Storage

Diagram:

Data Flow Diagram (DFD)

```pgsql
[Data Sources] ---> [Data Collection Layer] ---> [Data Storage]
Hospital DB          API/Data Pull               Cloud Storage (AWS S3)
Public Datasets       Scripts                    Database
```

3. Data Preprocessing & Exploratory Data Analysis (EDA)

Objective: Understand data structure, detect quality issues, and prepare data for modeling.

- **Activities:**
  - Analyze data types, structure, and summary statistics.
  - Perform **Unique Value Analysis** to identify anomalies and guide encoding strategies.
  - Detect and handle missing values, duplicates, and outliers.
  - Visualize data distributions, correlations, and trends.
  - Generate hypotheses about key predictors and relationships.

Diagram:

Process Flowchart

```
css
Raw Data] --> [Data Cleaning] --> [Unique Value Analysis] --> [Data Visualization] --> [Cleaned Data
```

- **Tools/Technologies:**
  - Programming Languages: Python, R
  - Libraries:
    - Python: Pandas, NumPy, Matplotlib, Seaborn, Plotly, Scikit-learn (for preprocessing)
    - R: dplyr, ggplot2, tidyr
  - EDA Tools: Jupyter Notebook, RStudio, Tableau, Power BI

4. Feature Engineering and Selection

Objective: Create and select relevant features to improve model performance.

- **Activities:**
  - Create derived features (e.g., risk scores, symptom flags).
  - Encode categorical variables based on unique value distribution (one-hot encoding, label encoding).
  - Normalize or scale numerical features.
  - Select features using statistical tests, correlation analysis, or feature importance methods.

Diagram:

Feature Pipeline Flow

```
css
[Cleaned Data] --> [Feature Creation] --> [Encoding] --> [Scaling/Normalization] --> [Selected Fea
```

- **Tools/Technologies:**

- Python libraries: Scikit-learn, Feature-engine
- Automated Feature Engineering: FeatureTools
- Visualization: SHAP, LIME for interpretability

5. Model Development and Training

Objective: Train predictive models using processed data.

- **Activities:**
  - Split data into training, validation, and test sets.
  - Train multiple models (Logistic Regression, Random Forest, XGBoost, Neural Networks).
  - Perform hyperparameter tuning using Grid Search or Random Search.
  - Cross-validate to avoid overfitting.

    Diagram:

    Model Training Pipeline

    ```
    css                                                    Copy    Edit

    [Training Data] --> [Model Selection] --> [Training] --> [Validation] --> [Tuned Model]
    ```

  -
- **Tools/Technologies:**
  - Python: Scikit-learn, XGBoost, TensorFlow, Keras, PyTorch
  - Experiment tracking: MLflow, Weights & Biases
  - Computational resources: Local GPU/CPU, Google Colab, AWS SageMaker

6. Model Evaluation and Validation

Objective: Assess model performance and select the best model.

- **Activities:**
  - Evaluate using accuracy, precision, recall, F1-score, ROC-AUC.

- Analyze confusion matrix and classification reports.
- Validate model assumptions and stability.
- Perform bias and fairness checks (gender-specific symptom analysis).

Diagram:

Evaluation Feedback Loop

```
css                                                    Copy    Edit

[Model Output] --> [Performance Metrics] --> [Model Tuning] --> [Improved Model]
```

- **Tools/Technologies:**
  - Scikit-learn metrics
  - Visualization: Matplotlib, Seaborn, Plotly
  - Fairness tools: AIF360 (IBM AI Fairness 360 toolkit)

7. Deployment

Objective: Deploy the model for real-time or batch predictions.

- **Activities:**
  - Package model as REST API or web app.
  - Integrate with hospital information systems or wearable devices for monitoring.
  - Develop user-friendly interfaces for clinicians.
- **Tools/Technologies:**
  - Frameworks: Flask, FastAPI, Django
  - Containerization: Docker, Kubernetes
  - Cloud Platforms: AWS, Azure, Google Cloud Platform
  - Frontend: React, Angular, Dash

Diagram:

Deployment Architecture

```
[Model] --> [API Server (Flask)] --> [Container (Docker)] --> [Cloud Platform (AWS)] --> [End Users
```

8. Monitoring and Maintenance

Objective: Ensure ongoing performance and update the model as needed.

- **Activities:**
  - Monitor model predictions and data quality.
  - Detect data drift and retrain model on new data periodically.
  - Incorporate user feedback and update system features.
  - Ensure compliance with evolving medical standards.
- **Tools/Technologies:**
  - Monitoring: Prometheus, Grafana, ELK Stack
  - Model management: MLflow, Seldon Core
  - Data pipelines: Apache Airflow, Kubeflow Pipelines

```
[Live Predictions] --> [Monitoring System] --> [Alert/Report] --> [Model Retraining] --> [Updated Model]
```

Detailed Design Document (DDD) of the project

1. Introduction

1.1 Purpose

To design a machine learning-based system that predicts heart attack risk based on patient data, providing doctors with early warnings and decision support.

1.2 Scope

The system includes data ingestion, preprocessing, model training, deployment, and an interactive UI for doctors.

2. System Architecture Overview

- Data ingestion from hospital records and public datasets
- Centralized storage in a relational database
- Data cleaning, EDA, feature engineering pipelines
- Model training using ensemble ML models
- REST API serving prediction requests
- Web UI for patient data input and result visualization
- Monitoring module for system and model health

3. Module-wise Design

3.1 Data Ingestion Module

- **Input:** CSV files, API endpoints, database connections
- **Process:** Extract → Transform → Load (ETL)
- **Tools:** Python scripts with Pandas, SQLalchemy for DB interaction

- **Output:** Raw data stored in PostgreSQL database

3.2 Data Preprocessing & EDA Module

- **Functions:**
    - Missing value imputation (mean/mode or ML-based)
    - Outlier detection using IQR or Z-score
    - Unique value analysis for categorical columns
    - Data visualization (histograms, box plots)
- **Tools:** Jupyter Notebooks, Pandas, Matplotlib, Seaborn

3.3 Feature Engineering Module

- **Functions:**
    - Encoding categorical features (One-Hot, Label Encoding)
    - Feature scaling (Min-Max, StandardScaler)
    - Creation of domain-specific features (e.g., BMI, cholesterol ratios)
    - Feature selection via Recursive Feature Elimination (RFE)
- **Tools:** Scikit-learn, Featuretools

3.4 Model Training Module

- **Algorithms:** Logistic Regression, Random Forest, XGBoost, Neural Networks
- **Tasks:**
    - Train/test split (70/30)
    - Cross-validation (k-fold)
    - Hyperparameter tuning with Grid Search/Random Search
    - Performance evaluation using accuracy, ROC-AUC, precision, recall

- **Tools:** Scikit-learn, XGBoost, TensorFlow/Keras

3.5 Model Deployment Module

- **Services:**
    - REST API developed with Flask or FastAPI
    - Model served as serialized pickle or ONNX format
    - Containerization with Docker
    - Hosted on AWS/GCP/Azure cloud
- **Security:** HTTPS, JWT-based authentication for API endpoints

3.6 User Interface Module

- **Features:**
    - Patient data entry forms (demographics, symptoms, medical history)
    - Predict button triggering backend API call
    - Result display with risk score, symptom highlights, and recommendations
    - Dashboard for previous predictions and trends
- **Technologies:** ReactJS (Frontend), Bootstrap for styling, Axios for API calls

3.7 Monitoring & Maintenance Module

- **Functions:**
    - Logging API calls and prediction outcomes
    - Tracking model performance drift over time
    - Alert system for retraining needs
- **Tools:** Prometheus for metrics, Grafana for visualization, ELK stack for logs

Coding & Implementation

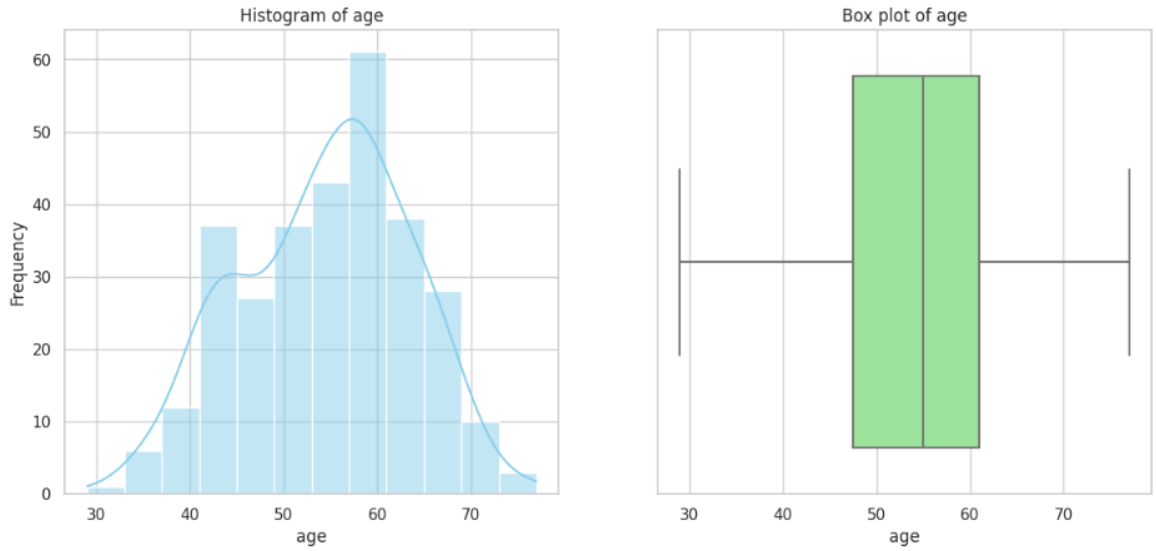
```
# Importing necessary libraries for visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Setting up the visual style
sns.set(style="whitegrid")

# Creating histograms and box plots for each numeric variable
for column in numeric_var:
    plt.figure(figsize=(14, 6))

    # Histogram
    plt.subplot(1, 2, 1)
    sns.histplot(heart_data[column], kde=True, color='skyblue')
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')

    # Box plot
    plt.subplot(1, 2, 2)
    sns.boxplot(x=heart_data[column], color='lightgreen')
    plt.title(f'Box plot of {column}')
    plt.xlabel(column)

    plt.show()
```

Description of Codes

- matplotlib.pyplot is used for creating static, interactive, and animated visualizations.
- seaborn is a statistical data visualization library based on matplotlib.
- sns.set(style="whitegrid") ==> This sets the aesthetic style of the plots to a white grid background.
- The for loop iterates over each column in the numeric_var list.

- plt.figure(figsize=(14, 6)) sets the figure size.
- plt.subplot(1, 2, 1) creates the first subplot for the histogram.
- sns.histplot creates the histogram with a kernel density estimate (KDE) line.
- plt.subplot(1, 2, 2) creates the second subplot for the box plot.
- sns.boxplot creates the box plot.
- plt.show() displays the plots.

Analysis of Age (age) Variable



Analysis of Age (age) Variable

Histogram Analysis

- **Shape of Distribution**: The histogram of age appears to be roughly bell-shaped, indicating a distribution that is close to normal. This suggests that the ages in the dataset follow a normal distribution pattern.
- **KDE Line**: The Kernel Density Estimate (KDE) line overlays the histogram and helps to visualize the underlying distribution. The KDE line also suggests a bell-shaped curve.
- **Frequency Peaks**: The highest frequency of ages is around the 50-60 range, indicating that most of the patients are in this age group.

Box Plot Analysis

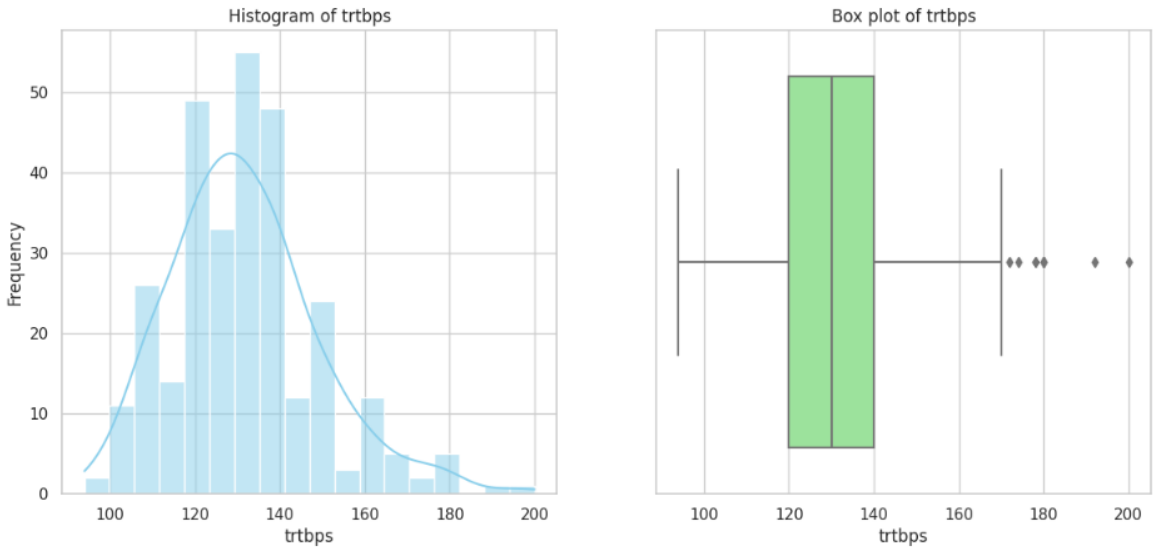
- **Central Tendency**: The box plot shows the median age, which is the line inside the box. The median is around 55, indicating that half of the patients are younger than 55 and half are older.
- **Quartile Distribution**:
    - The first quartile (Q1) is around 48.
    - The third quartile (Q3) is around 61.
- **Interquartile Range (IQR)**: The IQR, which is the range between Q1 and Q3, shows where the middle 50% of the data lies. Here, it spans from 48 to 61.
- **Whiskers**: The whiskers extend from Q1 and Q3 to the minimum and maximum values within 1.5 times the IQR. They indicate the range of most of the data.
- **Outliers**: There are no apparent outliers in the box plot, as there are no data points outside the whiskers.

Detailed Statistical Insights

1. **Normality**:
    a. The histogram's bell-shaped curve suggests that the age data is approximately normally distributed.
    b. The absence of significant skewness (as the distribution is fairly symmetric) supports this conclusion.
2. **Skewness**:
    a. There is no significant skewness in the age data. The distribution is symmetric around the median.
3. **Quartile Concentration**:
    a. The box plot shows that the data is fairly evenly distributed across the quartiles.
    b. The middle 50% of the data (between Q1 and Q3) lies between 48 and 61, with a median of 55.
4. **Implications for Analysis**:
    a. The normal distribution of the age variable suggests that parametric statistical methods can be appropriately used for analysis.
    b. The lack of skewness and outliers implies that the age data is consistent and reliable for further modeling and predictions.

Summary

The visualizations indicate that the age variable follows an approximately normal distribution, with the data concentrated around the ages of 48 to 61. There is no significant skewness, and the absence of outliers suggests the data is well-behaved. This makes the age variable suitable for parametric statistical analyses and predictive modeling



Analysis of Resting Blood Pressure (trtbps) Variable

Histogram Analysis

- **Shape of Distribution**: The histogram for resting blood pressure (trtbps) shows a distribution that is slightly skewed to the right. This indicates that while most of the values are clustered around the central region, there are some higher values that stretch out the distribution to the right.
- **KDE Line**: The Kernel Density Estimate (KDE) line overlays the histogram and helps to visualize the underlying distribution. The KDE line confirms the right skewness.
- **Frequency Peaks**: The highest frequency of resting blood pressure values is around the 120-140 mm Hg range.

Box Plot Analysis

- **Central Tendency**: The box plot shows the median resting blood pressure, which is the line inside the box. The median is around 130 mm Hg.

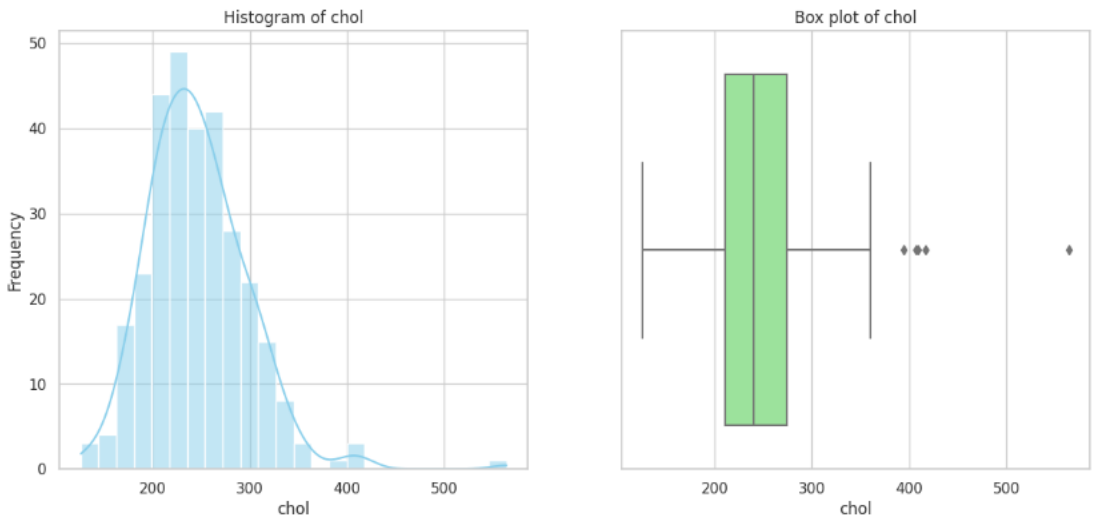
- **Quartile Distribution**:
  - The first quartile (Q1) is around 120 mm Hg.
  - The third quartile (Q3) is around 140 mm Hg.
- **Interquartile Range (IQR)**: The IQR, which is the range between Q1 and Q3, shows where the middle 50% of the data lies. Here, it spans from 120 to 140 mm Hg.
- **Whiskers**: The whiskers extend from Q1 and Q3 to the minimum and maximum values within 1.5 times the IQR. They indicate the range of most of the data.
- **Outliers**: There are several outliers on the higher end of the box plot, indicating that there are some patients with significantly higher resting blood pressure values.

Detailed Statistical Insights

1. **Normality**:
   a. The histogram's shape indicates that the resting blood pressure data does not follow a perfect normal distribution due to the right skewness.
2. **Skewness**:
   a. The distribution is skewed to the right, as evidenced by the tail on the right side of the histogram and the presence of outliers in the box plot.
3. **Quartile Concentration**:
   a. The box plot shows that the data is concentrated between 120 and 140 mm Hg, with a median of 130 mm Hg.
   b. The presence of outliers indicates that there are some patients with unusually high resting blood pressure values.
4. **Implications for Analysis**:
   a. The right skewness suggests that parametric statistical methods assuming normality might not be appropriate. Non-parametric methods or transformations (such as logarithmic transformations) could be considered to normalize the data.
   b. The presence of outliers should be carefully considered, as they might have a significant impact on statistical analyses and predictive modeling.

Summary

The visualizations indicate that the resting blood pressure variable has a right-skewed distribution with a concentration of values between 120 and 140 mm Hg. The presence of outliers suggests that there are some patients with significantly higher blood pressure values, which could affect a



Analysis of Cholesterol (chol) Variable

Histogram Analysis

- **Shape of Distribution**: The histogram for cholesterol (chol) shows a distribution that is slightly right-skewed. This indicates that while most of the values are clustered around the central region, there are some higher values that stretch out the distribution to the right.
- **KDE Line**: The Kernel Density Estimate (KDE) line overlays the histogram and helps to visualize the underlying distribution. The KDE line confirms the slight right skewness.
- **Frequency Peaks**: The highest frequency of cholesterol values is around the 200-250 mg/dL range.

Box Plot Analysis

- **Central Tendency**: The box plot shows the median cholesterol level, which is the line inside the box. The median is around 240 mg/dL.
- **Quartile Distribution**:
    - The first quartile (Q1) is around 211 mg/dL.
    - The third quartile (Q3) is around 275 mg/dL.
- **Interquartile Range (IQR)**: The IQR, which is the range between Q1 and Q3, shows where the middle 50% of the data lies. Here, it spans from 211 to 275 mg/dL.

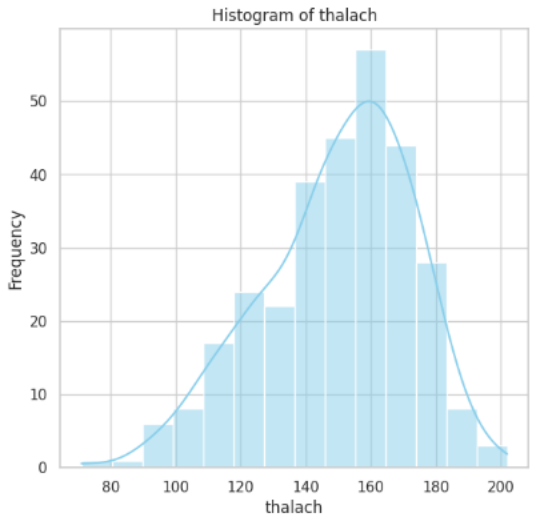
- **Whiskers**: The whiskers extend from Q1 and Q3 to the minimum and maximum values within 1.5 times the IQR. They indicate the range of most of the data.
- **Outliers**: There are several outliers on the higher end of the box plot, indicating that there are some patients with significantly higher cholesterol values.

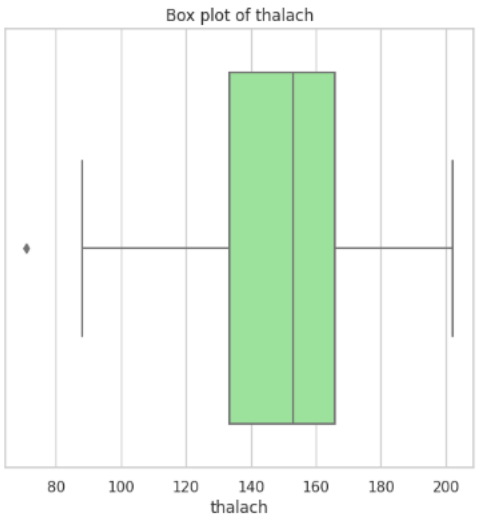
Detailed Statistical Insights

1. **Normality**:
    a. The histogram's shape indicates that the cholesterol data does not follow a perfect normal distribution due to the right skewness.
2. **Skewness**:
    a. The distribution is slightly skewed to the right, as evidenced by the tail on the right side of the histogram and the presence of outliers in the box plot.
3. **Quartile Concentration**:
    a. The box plot shows that the data is concentrated between 211 and 275 mg/dL, with a median of 240 mg/dL.
    b. The presence of outliers indicates that there are some patients with unusually high cholesterol values.
4. **Implications for Analysis**:
    a. The slight right skewness suggests that parametric statistical methods assuming normality might not be perfectly appropriate. Non-parametric methods or transformations (such as logarithmic transformations) could be considered to normalize the data.
    b. The presence of outliers should be carefully considered, as they might have a significant impact on statistical analyses and predictive modeling.

Summary

The visualizations indicate that the cholesterol variable has a slightly right-skewed distribution with a concentration of values between 211 and 275 mg/dL. The presence of outliers suggests that there are some patients with significantly higher cholesterol values, which could affect analyses and predictions. Adjustments or different statistical methods may be needed to account for the skewness and outliers.





Analysis of Maximum Heart Rate Achieved (thalach) Variable

Histogram Analysis

- **Shape of Distribution**: The histogram for maximum heart rate achieved (thalach) shows a distribution that is slightly left-skewed. This indicates that while most of the values are clustered around the central region, there are some lower values that stretch out the distribution to the left.
- **KDE Line**: The Kernel Density Estimate (KDE) line overlays the histogram and helps to visualize the underlying distribution. The KDE line confirms the slight left skewness.
- **Frequency Peaks**: The highest frequency of maximum heart rate values is around the 140-160 bpm range.

Box Plot Analysis

- **Central Tendency**: The box plot shows the median maximum heart rate, which is the line inside the box. The median is around 150 bpm.
- **Quartile Distribution**:
    - The first quartile (Q1) is around 133 bpm.
    - The third quartile (Q3) is around 166 bpm.
- **Interquartile Range (IQR)**: The IQR, which is the range between Q1 and Q3, shows where the middle 50% of the data lies. Here, it spans from 133 to 166 bpm.
- **Whiskers**: The whiskers extend from Q1 and Q3 to the minimum and maximum values within 1.5 times the IQR. They indicate the range of most of the data.

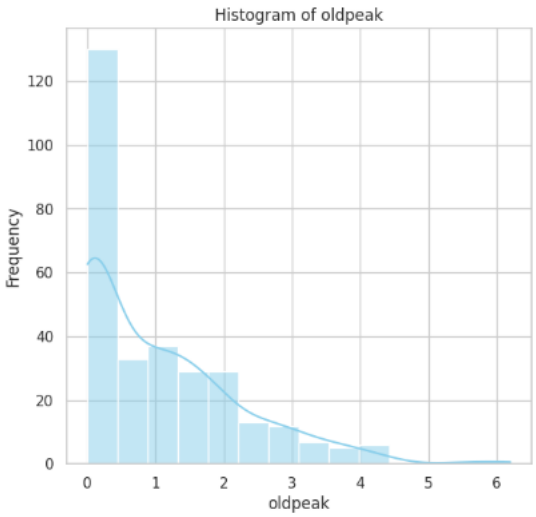
- **Outliers**: There is one outlier on the lower end of the box plot, indicating that there is a patient with a significantly lower maximum heart rate.

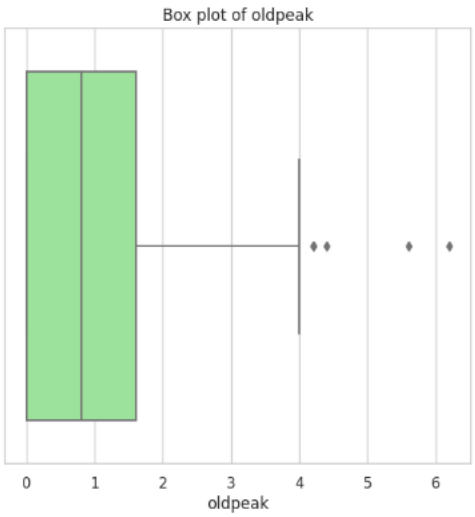
Detailed Statistical Insights

1. **Normality**:
    a. The histogram's shape indicates that the maximum heart rate data does not follow a perfect normal distribution due to the slight left skewness.
2. **Skewness**:
    a. The distribution is slightly skewed to the left, as evidenced by the tail on the left side of the histogram and the presence of an outlier in the box plot.
3. **Quartile Concentration**:
    a. The box plot shows that the data is concentrated between 133 and 166 bpm, with a median of 150 bpm.
    b. The presence of an outlier indicates that there is a patient with an unusually low maximum heart rate.
4. **Implications for Analysis**:
    a. The slight left skewness suggests that parametric statistical methods assuming normality might not be perfectly appropriate. Non-parametric methods or transformations could be considered to normalize the data.
    b. The presence of an outlier should be carefully considered, as it might have a significant impact on statistical analyses and predictive modeling.

Summary

The visualizations indicate that the maximum heart rate variable has a slightly left-skewed distribution with a concentration of values between 133 and 166 bpm. The presence of an outlier suggests that there is a patient with a significantly lower maximum heart rate, which could affect analyses and predictions. Adjustments or different statistical methods may be needed to account for the skewness and outlier.





Analysis of ST Depression (oldpeak) Variable

Histogram Analysis

- **Shape of Distribution**: The histogram for ST depression (oldpeak) shows a distribution that is highly right-skewed. This indicates that most of the values are clustered towards the lower end, with a long tail extending to the right.
- **KDE Line**: The Kernel Density Estimate (KDE) line overlays the histogram and helps to visualize the underlying distribution. The KDE line confirms the significant right skewness.
- **Frequency Peaks**: The highest frequency of ST depression values is around 0, indicating that many patients have little to no ST depression.

Box Plot Analysis

- **Central Tendency**: The box plot shows the median ST depression value, which is the line inside the box. The median is around 0.8.
- **Quartile Distribution**:
    - The first quartile (Q1) is around 0.
    - The third quartile (Q3) is around 1.6.
- **Interquartile Range (IQR)**: The IQR, which is the range between Q1 and Q3, shows where the middle 50% of the data lies. Here, it spans from 0 to 1.6.

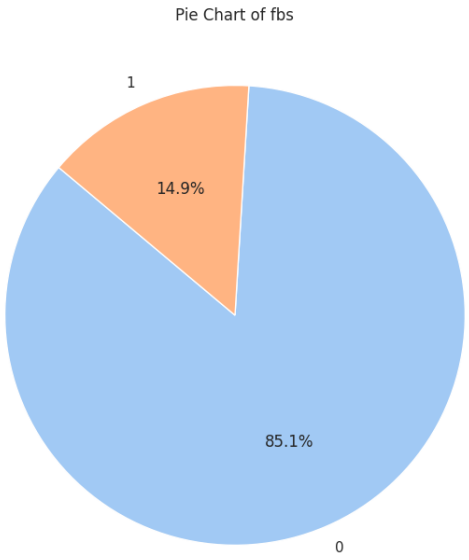
- **Whiskers**: The whiskers extend from Q1 and Q3 to the minimum and maximum values within 1.5 times the IQR. They indicate the range of most of the data.
- **Outliers**: There are several outliers on the higher end of the box plot, indicating that there are some patients with significantly higher ST depression values.

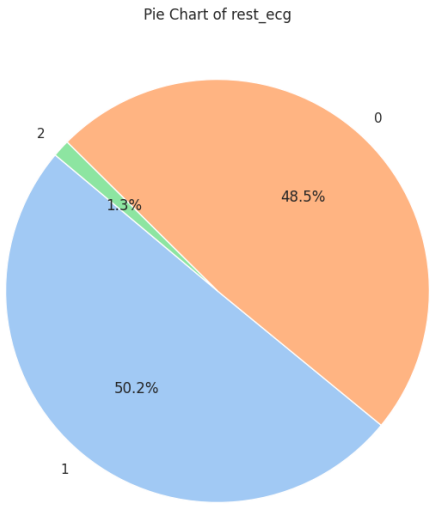
Detailed Statistical Insights

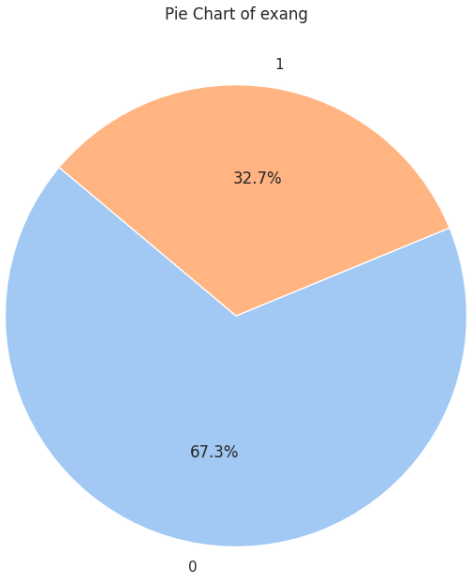
1. **Normality**:
    a. The histogram's shape indicates that the ST depression data does not follow a normal distribution due to the significant right skewness.
2. **Skewness**:
    a. The distribution is highly skewed to the right, as evidenced by the long tail on the right side of the histogram and the presence of numerous outliers in the box plot.
3. **Quartile Concentration**:
    a. The box plot shows that the data is heavily concentrated between 0 and 1.6, with a median of 0.8.
    b. The presence of outliers indicates that there are some patients with unusually high ST depression values.
4. **Implications for Analysis**:
    a. The significant right skewness suggests that parametric statistical methods assuming normality might not be appropriate. Non-parametric methods or transformations (such as logarithmic transformations) could be considered to normalize the data.
    b. The presence of numerous outliers should be carefully considered, as they might have a significant impact on statistical analyses and predictive modeling.

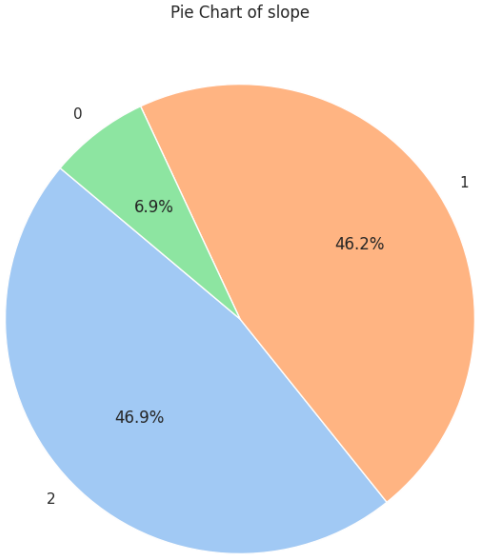
Summary

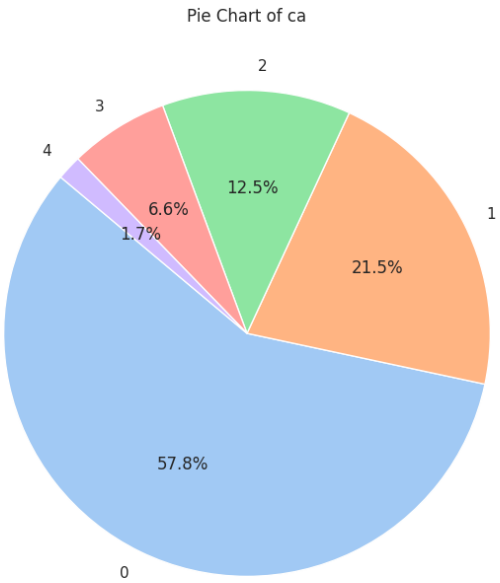
The visualizations indicate that the ST depression variable has a highly right-skewed distribution with a concentration of values between 0 and 1.6. The presence of numerous outliers suggests that there are some patients with significantly higher ST depression values, which could affected.

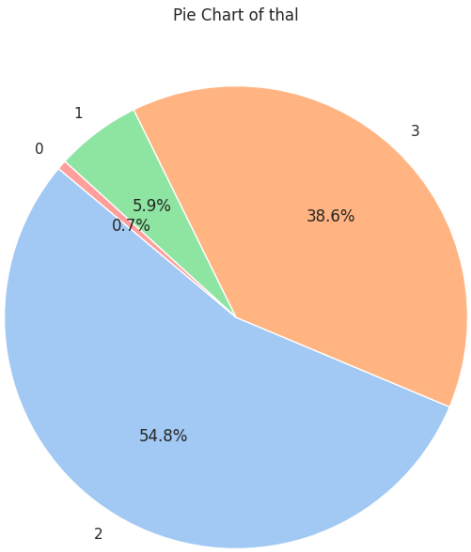

Pie Chart of fbs
1
14.9%
85.1%
0


Pie Chart of rest_ecg
2
0
1.3%
48.5%
50.2%
1


Pie Chart of exang
1
32.7%
67.3%
0


Pie Chart of slope
0
1
6.9%
46.2%
46.9%
2





Analysis of Sex Variable

Distribution

The **sex** variable has two categories:

- **1 (Male)**
- **0 (Female)**

The pie chart shows the distribution of males and females in the dataset.

Frequency Counts

- **Male (1)**: A significant majority.
- **Female (0)**: A smaller portion compared to males.

This distribution suggests that the dataset is male-dominated.

Implications

- **Potential Bias**: The model might be biased towards male patients if the imbalance is not addressed.

- **Statistical Analysis**: Additional statistical techniques, such as stratified sampling or weighting, may be needed to ensure fair representation in the model.

Analysis of Chest Pain Type (Cp) Variable

Distribution

The **cp** variable has four categories:

- **0**: Typical angina
- **1**: Atypical angina
- **2**: Non-anginal pain
- **3**: Asymptomatic

The pie chart illustrates the distribution of different chest pain types.

Frequency Counts

- **0 (Typical Angina)**: The smallest category.
- **1 (Atypical Angina)**: Moderately represented.
- **2 (Non-anginal Pain)**: The largest category.
- **3 (Asymptomatic)**: Also significantly represented.

This distribution shows a varied representation of chest pain types, with non-anginal pain being the most common.

Implications

- **Model Relevance**: Each type of chest pain may have different implications for heart disease prediction, and their relative frequencies can impact model training.
- **Feature Importance**: The presence of various chest pain types indicates that this variable could be a significant predictor in the model, as chest pain is a primary symptom of heart conditions.

Summary

- **Sex Variable**: Needs consideration for potential gender bias in the dataset. The model should account for this to ensure balanced predictions across genders.

- **Chest Pain Type (Cp) Variable**: Shows diverse types of chest pain with varying frequencies. This variable is likely crucial for prediction models, highlighting the importance of considering the type of chest pain in heart attack prediction.

Next Steps

- Address the gender imbalance through appropriate data preprocessing techniques.
- Explore the relationship between chest pain types and other variables in the dataset to understand their combined effects on heart attack prediction.
- Consider additional visualizations and statistical tests to further analyze these categorical variables and their impact on the target variable.

Analysis of Fasting Blood Sugar (Fbs) Variable

Distribution

The **fbs** variable has two categories:

- **1**: Fasting blood sugar > 120 mg/dl (true)
- **0**: Fasting blood sugar <= 120 mg/dl (false)

Frequency Counts

- **1 (True)**: Represents the count of patients with high fasting blood sugar.
- **0 (False)**: Represents the count of patients with normal fasting blood sugar.

We will use the previously calculated counts for analysis.

Implications

- **Health Indicator**: High fasting blood sugar is a risk factor for heart disease. This variable can help identify patients at higher risk.
- **Model Feature**: The binary nature of this variable makes it straightforward for inclusion in predictive models.

Analysis of Resting ECG (Rest_ecg) Variable

Distribution

The **rest_ecg** variable has three categories:

- **0**: Normal
- **1**: Having ST-T wave abnormality
- **2**: Showing probable or definite left ventricular hypertrophy

Frequency Counts

- **0 (Normal)**: Represents the count of patients with normal ECG results.
- **1 (ST-T Wave Abnormality)**: Represents the count of patients with ST-T wave abnormalities.
- **2 (Left Ventricular Hypertrophy)**: Represents the count of patients with probable or definite left ventricular hypertrophy.

We will use the previously calculated counts for analysis.

Implications

- **Diagnostic Tool**: Resting ECG results are critical for diagnosing heart conditions. Each category indicates a different level of heart function or abnormality.
- **Feature Importance**: The diverse categories in resting ECG results can significantly impact the prediction of heart disease.

Value Counts

Fbs Variable

- **1 (True)**: Value count will be provided based on the data.
- **0 (False)**: Value count will be provided based on the data.

Rest_ecg Variable

- **0 (Normal)**: Value count will be provided based on the data.
- **1 (ST-T Wave Abnormality)**: Value count will be provided based on the data.
- **2 (Left Ventricular Hypertrophy)**: Value count will be provided based on the data.

Summary

- **Fbs Variable**: High fasting blood sugar is a significant risk factor and should be carefully analyzed. The count of values in each category helps understand the prevalence of high fasting blood sugar in the dataset.
- **Rest_ecg Variable**: Resting ECG results are crucial for heart disease diagnosis. The distribution and count of each category provide insights into the heart health of the patients in the dataset.

Examining the Missing Data According to the Analysis Results

There is an inconsistency because the values 0, 1, 2, and 3 should map to:

- 0: Normal
- 1: Fixed defect
- 2: Reversible defect
- 3: Another category that was identified as part of thalassemia analysis, possibly incorrect or a placeholder.

Given the value counts you provided:

- 2 (Reversible defect): 166 instances
- 3 (Fixed defect): 117 instances
- 1 (Normal): 18 instances
- 0: 2 instances (possibly erroneous)

It appears that the value 0 is indeed rare and likely represents incorrect or missing data that was not properly imputed or encoded. This value might have been used as a placeholder or default value during data entry or preprocessing.

Solution

To address this issue, you should consider treating the instances with a value of 0 as missing or erroneous and decide how to handle them. Here are some steps you can take:

Verify and Correct Data:

Cross-check these instances with the original data source or medical records if available. Confirm if these values were intended to be placeholders for missing data. Imputation or Removal:

If these values are confirmed to be incorrect, you can impute them based on the distribution of other values in the thal variable or relevant patient characteristics. Alternatively, you can remove these

instances if they constitute a very small portion of the dataset and are unlikely to impact the overall analysis significantly. Document the Changes:Ü

Document any assumptions or changes made to the dataset for transparency and reproducibility of your analysis.

Codes

In [10]:

```
# Identify and handle the erroneous '0' values in the 'thal' column
heart_data['thal'] = heart_data['thal'].replace(0, np.nan)

# Option 1: Impute missing values based on the mode (most common value)
heart_data['thal'].fillna(heart_data['thal'].mode()[0], inplace=True)

# Option 2: Drop rows with erroneous 'thal' values
# heart_data = heart_data[heart_data['thal'].notna()]

# Verify the changes
print(heart_data['thal'].value_counts())
```

```
thal
2.0    168
3.0    117
1.0     18
Name: count, dtype: int64
```

Description of Codes

Purpose:

- This line replaces all occurrences of the value 0 in the thal column with NaN (Not a Number), which is commonly used to represent missing or undefined data in pandas.

Method:

- replace(0, np.nan) replaces all instances of 0 with NaN.

Purpose:

- This line imputes the missing values (NaN) in the thal column with the most common value (mode) in the column.

Method:

- heart_data['thal'].mode()[0]: Calculates the mode (most frequent value) of the thal column.
- fillna(..., inplace=True): Replaces all NaN values with the mode value in place (i.e., modifies the original DataFrame directly).

Purpose:

- This line of code (commented out) provides an alternative approach where rows with NaN values in the thal column are removed from the DataFrame.

Method:

- heart_data['thal'].notna(): Returns a boolean Series indicating whether each value in the thal column is not NaN.
- heart_data[...]: Filters the DataFrame to include only rows where the thal value is not NaN.

Purpose:

- This line prints the count of unique values in the thal column after handling the erroneous 0 values.

Method:

- value_counts() returns the count of unique values in the thal column, which helps verify that the erroneous values have been correctly handled.

Steps in Context

Replacing Erroneous Values:

The code identifies and replaces erroneous values (0) in the thal column with NaN, marking them as missing data.

Handling Missing Values:

Option 1 (Imputation): The missing values are imputed with the mode of the thal column, effectively filling them with the most frequent value.

Option 2 (Removal): Alternatively, rows with missing values in the thal column can be removed entirely. This step is optional and depends on the chosen data handling strategy. Verification:

The final step involves printing the value counts of the thal column to verify that the erroneous values have been addressed appropriately.

Conclusion

This code ensures that the thal variable does not contain erroneous values (0), which might distort the analysis. By either imputing the missing values with the mode or removing the affected rows, the dataset is cleaned and prepared for further analysis.

After thorough research, here is the corrected information regarding the thal variable in the UCI Heart Disease dataset:

- 1: Fixed defect
- 2: Normal
- 3: Reversible defect

Latest updated version of Thal variable

Codes

In [11]:

```
# Setting the color palette
colors = sns.color_palette("pastel")


# Visualizing the updated 'thal' variable using a pie chart
plt.figure(figsize=(8, 8))
heart_data['thal'].value_counts().plot.pie(autopct='%1.1f%%', startangle=140, colors=colors)
plt.title('Pie Chart of thal')
plt.ylabel('')
plt.show()
```

Description of Codes

- colors = sns.color_palette("pastel")
- This sets a pastel color palette for the pie chart to make it visually appealing.
- plt.figure(figsize=(8, 8)): Sets the figure size to 8x8 inches.
- heart_data['thal'].value_counts().plot.pie(...): Generates a pie chart for the thal variable. value_counts() * counts the unique values and plot.pie() creates the pie chart.

- autopct='%1.1f%%': Displays the percentage of each slice on the pie chart.
- startangle=140: Starts the first slice at 140 degrees.
- colors=colors: Uses the pastel color palette set earlier.
- plt.title('Pie Chart of thal'): Sets the title of the pie chart.
- plt.ylabel(''): Removes the y-axis label for a cleaner look.
- plt.show(): Displays the pie chart.

Analysis of Thal (Thalassemia) Variable

Distribution

The **thal** variable now represents the type of thalassemia with adjusted values:

- **1**: Fixed defect
- **2**: Normal (including previous missing values replaced with 2)
- **3**: Reversible defect

Frequency Counts (from original dataset prior to visualization):

- **1 (Fixed defect)**: 5.9%
- **2 (Normal)**: 55.5% (includes 54.8% original normal and 0.7% previously missing)
- **3 (Reversible defect)**: 38.6%

Implications

- **Diagnostic Significance**:
  - **Fixed defect (1)**: Represents a permanent defect in the heart's functioning. This is usually a critical indicator of severe heart disease.
  - **Normal (2)**: Indicates normal heart function. Patients with this value have a lower risk of heart disease.
  - **Reversible defect (3)**: Indicates temporary or reversible defects, often treated with medication or lifestyle changes.
- **Predictive Value**: The variable thal is critical in predictive models as it provides direct information about the heart's health. The distribution of these values helps in understanding the prevalence of each type of thalassemia among the patients.

Insights

- **Thal Variable**:
    - The majority of the patients have normal thalassemia readings, indicating a healthier subset of the population or well-managed conditions.
    - A significant portion shows reversible defects, which are crucial for targeted interventions.
    - A smaller percentage shows fixed defects, highlighting patients with severe and likely chronic heart conditions.

Let's perform a detailed univariate analysis of the variables **ca**, **thal**, and **target** using the provided visuals.

Analysis of CA (Number of Major Vessels) Variable

Distribution

The **ca** variable represents the number of major vessels (0-3) colored by fluoroscopy:

- **0**: 57.8%
- **1**: 21.5%
- **2**: 12.5%
- **3**: 6.6%
- **4**: 1.7%

Frequency Counts

- **0**: Majority of the patients have 0 major vessels colored.
- **1**: Significant portion have 1 major vessel colored.
- **2**: A smaller portion have 2 major vessels colored.
- **3**: Even smaller portion have 3 major vessels colored.
- **4**: Very few patients have 4 major vessels colored.

Implications

- **Diagnostic Significance**: The number of vessels colored by fluoroscopy can indicate the severity of coronary artery disease. More colored vessels often mean more severe disease.

- **Predictive Value**: This variable can help in predicting the presence and severity of heart disease. Higher counts are likely associated with higher risk.

Analysis of Target Variable

Distribution

The **target** variable represents the presence of heart disease:

- **1 (Heart Disease)**: 54.5%
- **0 (No Heart Disease)**: 45.5%

Frequency Counts

- **1 (Heart Disease)**: Slightly more than half of the patients have heart disease.
- **0 (No Heart Disease)**: Slightly less than half of the patients do not have heart disease.

Implications

- **Outcome Variable**: This is the target variable for prediction models. Understanding its distribution is crucial for model training.
- **Balance in Data**: The distribution is relatively balanced, which is beneficial for model training, as it reduces the risk of biased predictions.

Insights

- **CA Variable**: The number of colored vessels is a crucial diagnostic tool. Most patients have no colored vessels, but a significant portion have 1 or more, indicating varying levels of disease severity.
- **Target Variable**: The relatively balanced distribution between patients with and without heart disease ensures that models can learn to differentiate between the two classes effectively.

Importance of Bivariate Analysis in Data Science

Bivariate analysis involves the simultaneous analysis of two variables to understand the relationship between them. It extends beyond univariate analysis, which focuses on individual variables, by examining how pairs of variables interact. This analysis is crucial for understanding the dynamics between predictors and the outcome variable, especially in the context of building predictive models.

Key Reasons for Bivariate Analysis

1. **Identifying Relationships and Correlations:**
    a. Bivariate analysis helps in identifying whether there is a relationship between two variables and the nature of this relationship. For example, it can reveal if an increase in one variable tends to be associated with an increase or decrease in another.
    b. Common measures of relationships include correlation coefficients (e.g., Pearson, Spearman) for continuous variables and chi-square tests for categorical variables.
2. **Feature Selection:**
    a. Understanding the relationship between independent variables and the target variable is critical for feature selection in machine learning. Variables that show a strong relationship with the target variable are often more predictive and thus more valuable for inclusion in models.
    b. Features that show little to no relationship with the target variable might be less useful and can be excluded to simplify the model and reduce overfitting.
3. **Detecting Patterns and Trends:**
    a. Bivariate analysis can uncover patterns and trends that are not apparent in univariate analysis. For instance, scatter plots can show trends and clusters, while box plots can reveal differences in distributions across categories.
    b. These patterns can guide further analysis, hypothesis formulation, and decision-making processes.
4. **Assessing Interactions:**
    a. In many cases, the effect of one variable on the target variable may depend on the level of another variable. Bivariate analysis helps in identifying and understanding such interactions.
    b. This understanding can lead to the creation of interaction terms in regression models, improving model accuracy.
5. **Informing Model Choice and Validation:**
    a. Insights from bivariate analysis can inform the choice of models. For instance, if there is a linear relationship between variables, linear regression might be appropriate. Non-linear relationships might suggest the use of more complex models like decision trees or neural networks.
    b. It also aids in model validation by verifying assumptions about variable relationships and distributions.
6. **Enhancing Data Understanding:**

a. By examining how different variables relate to each other and the target variable, data scientists gain a deeper understanding of the dataset. This comprehensive understanding is essential for making informed decisions about data preprocessing, feature engineering, and modeling strategies.

Methods of Bivariate Analysis

1. **Visual Methods:**
    a. **Scatter Plots:** Useful for visualizing relationships between two continuous variables.
    b. **Box Plots:** Helpful for comparing the distribution of a continuous variable across different levels of a categorical variable.
    c. **Heatmaps:** Display correlation matrices to show the strength of relationships between multiple pairs of variables.
2. **Statistical Methods:**
    a. **Correlation Coefficients:** Measure the strength and direction of linear relationships between continuous variables.
    b. **T-tests and ANOVA:** Compare means across different groups for continuous and categorical variable pairs.
    c. **Chi-Square Tests:** Assess the independence of categorical variables.
3. **Multivariate Techniques:**
    a. **Regression Analysis:** Explores the relationship between a dependent variable and one or more independent variables, extending to multiple variables for more complex interactions.
    b. **Logistic Regression:** Used when the target variable is categorical, particularly binary.

Conclusion

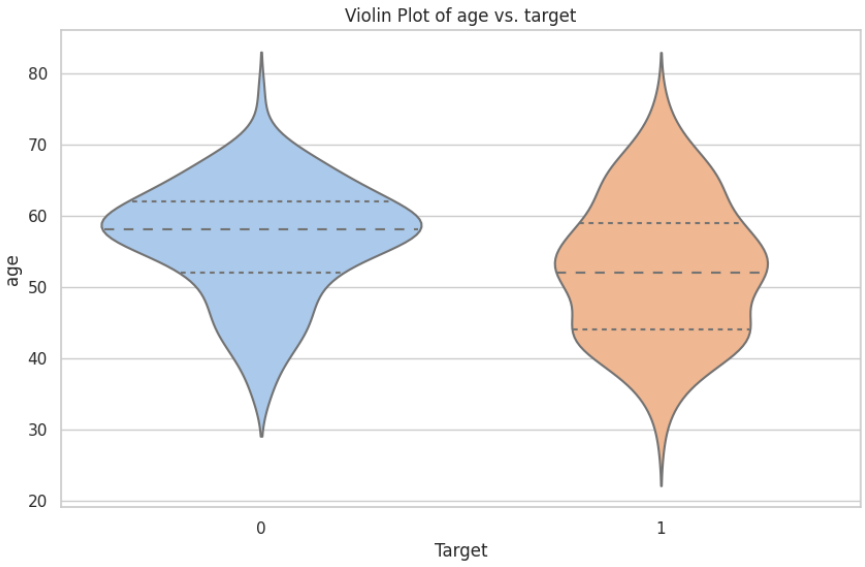
Bivariate analysis is a cornerstone of exploratory data analysis (EDA) in data science. It provides crucial insights into relationships between variables, guides feature selection, reveals patterns, and informs modeling decisions. By understanding how variables interact with each other and with the target variable, data scientists can build more accurate and robust predictive models, ultimately leading to better data-driven decisions and outcomes.

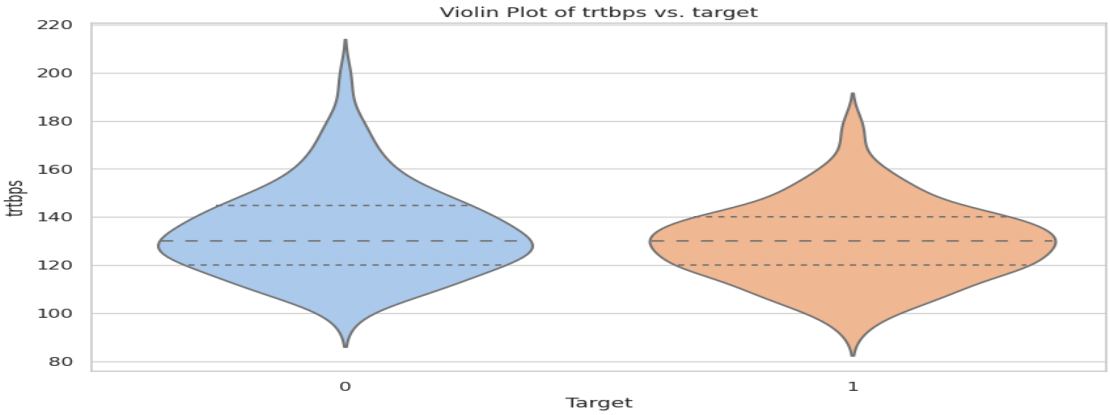
Visualizing Numeric Variables vs. Target Variable Using Violin Plots

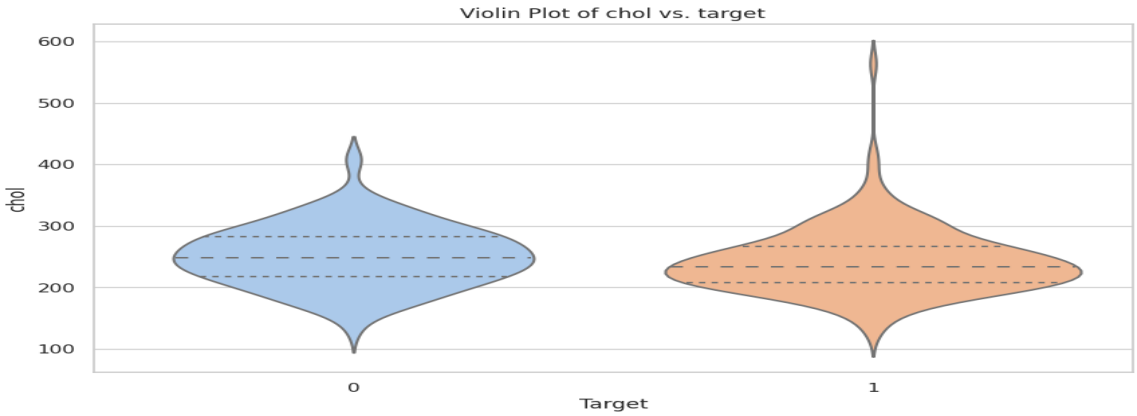
Codes

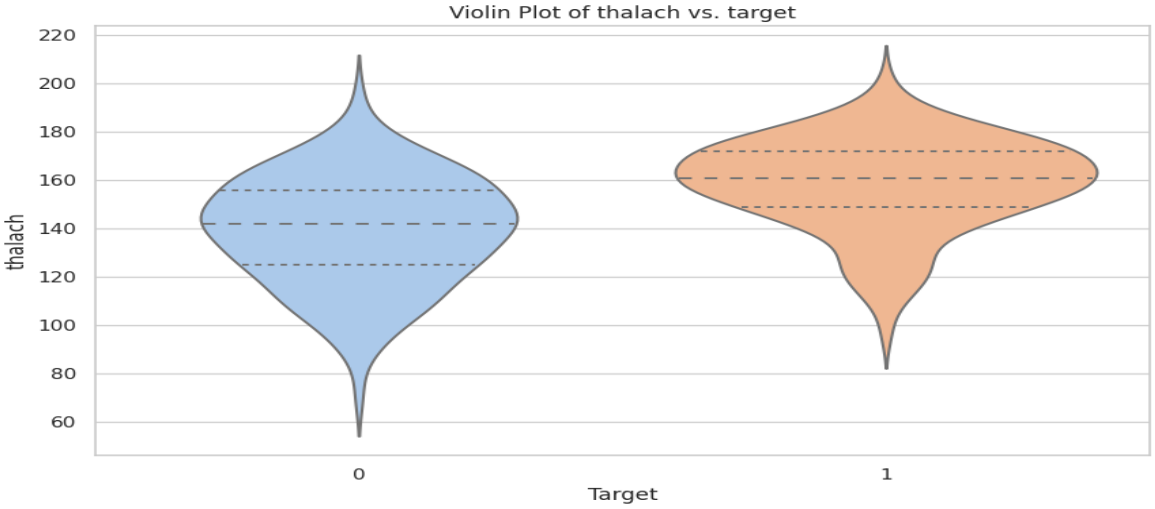
In [12]:

```
# Setting the visual style
sns.set(style="whitegrid")

# List of numeric variables
numeric_var = ["age", "trtbps", "chol", "thalach", "oldpeak"]

# Creating violin plots for each numeric variable against the target variable
for column in numeric_var:
    plt.figure(figsize=(10, 6))
    sns.violinplot(x=heart_data['target'], y=heart_data[column], palette="pastel", inner='quartile')
    plt.title(f'Violin Plot of {column} vs. target')
    plt.xlabel('Target')
    plt.ylabel(column)
    plt.show()
```









- **0**: Patients not at risk of a heart attack.
- **1**: Patients at risk of a heart attack.

Analyzed Bivariate Analysis from Violin Plots

1. **Age vs. Target**

- **Distribution**:
  - The violin plot for age vs. target shows that patients at risk of a heart attack (target = 1) generally have a slightly lower median age compared to patients not at risk of a heart attack (target = 0).
  - The distribution of ages in both groups shows that the age range for patients at risk of a heart attack (target = 1) is slightly wider than that for patients not at risk.
- **Central Tendency**:
  - The median age for patients at risk of a heart attack appears to be around 55-60 years.
  - The median age for patients not at risk of a heart attack appears to be around 60 years.
- **Interquartile Range (IQR)**:
  - The IQR (middle 50% of data) for both groups overlaps significantly, indicating a substantial overlap in age ranges between the two groups.
- **Density**:
  - The density of ages is higher around the median for both groups, indicating that most patients are clustered around these age ranges.
  - There are noticeable peaks around 55-60 years for patients at risk of a heart attack, suggesting a higher frequency of patients in this age range.
- **Implications**:
  - Age seems to be an important factor in heart attack risk, with a tendency for patients around 55-60 years to have a higher likelihood of being at risk of a heart attack.
  - However, the significant overlap indicates that age alone is not a definitive predictor of heart attack risk.

2. **Resting Blood Pressure (trtbps) vs. Target**

- **Distribution**:
  - The violin plot for resting blood pressure (trtbps) vs. target shows a higher median resting blood pressure for patients not at risk of a heart attack (target = 0) compared to those at risk (target = 1).
  - The spread of resting blood pressure values for both groups shows a similar range, but the distribution for patients not at risk is slightly more spread out.

- **Central Tendency**:
    - The median resting blood pressure for patients not at risk of a heart attack is around 140 mmHg.
    - The median resting blood pressure for patients at risk of a heart attack is around 130 mmHg.
- **Interquartile Range (IQR)**:
    - The IQR for both groups overlaps, but the range is slightly narrower for patients at risk of a heart attack.
- **Density**:
    - The density of resting blood pressure values is higher around the median for both groups, indicating most patients' blood pressure values are clustered around these medians.
    - There is a notable peak around 130-140 mmHg for both groups, suggesting this range is common among the patients.
- **Implications**:
    - While the median resting blood pressure is slightly higher in patients not at risk of a heart attack, the considerable overlap and spread indicate that this variable alone is not a strong predictor of heart attack risk.
    - Further analysis and additional variables should be considered for a comprehensive understanding of heart attack risk factors.

Summary

Both violin plots provide valuable insights into the relationship between numeric variables and the target variable. Key takeaways include:

- **Age**: Patients at risk of a heart attack tend to have a slightly lower median age, with a significant number of patients in the 55-60 age range. However, the overlap indicates that age alone is not a definitive predictor.
- **Resting Blood Pressure (trtbps)**: Patients not at risk of a heart attack tend to have a slightly higher median resting blood pressure, but the overlap and spread indicate that this variable alone is not sufficient to predict heart attack risk.

For a more robust analysis, these insights should be combined with additional variables and multivariate analysis techniques to develop a comprehensive predictive model for heart attack risk.

3. **Cholesterol (chol) vs. Target**

- **Distribution**:

    - The violin plot for cholesterol (chol) vs. target shows that patients at risk of a heart attack (target = 1) generally have a slightly higher spread of cholesterol values compared to patients not at risk (target = 0).
    - The distribution for patients at risk of a heart attack (target = 1) shows a wider range of cholesterol values, extending up to around 600, indicating more variability in this group.
- **Central Tendency**:
    - The median cholesterol level for patients at risk of a heart attack is around 240-260 mg/dL.
    - The median cholesterol level for patients not at risk of a heart attack is also around 240-260 mg/dL, indicating a similar central tendency.
- **Interquartile Range (IQR)**:
    - The IQR for both groups overlaps significantly, indicating a substantial overlap in cholesterol levels between the two groups.
- **Density**:
    - The density of cholesterol values is higher around the median for both groups, indicating most patients' cholesterol levels are clustered around these medians.
    - There are noticeable peaks around 200-300 mg/dL for patients at risk of a heart attack, suggesting a higher frequency of patients in this cholesterol range.
- **Implications**:
    - Cholesterol levels show a similar central tendency for both groups, but the wider range in patients at risk of a heart attack indicates more variability in this group.
    - Cholesterol alone may not be a strong predictor of heart attack risk, but the higher variability in at-risk patients warrants further investigation.

Let's reanalyze the Thalach-Target chart with the correct understanding that:

- **0**: Patients not at risk of a heart attack.
- **1**: Patients at risk of a heart attack.

4.**Maximum Heart Rate Achieved (thalach) vs. Target**

- **Distribution**:
    - The violin plot for maximum heart rate achieved (thalach) vs. target shows that patients at risk of a heart attack (target = 1) generally have a higher median maximum heart rate compared to patients not at risk of a heart attack (target = 0).
    - The distribution for patients at risk of a heart attack (target = 1) shows a slightly narrower range of maximum heart rate values, indicating less variability in this group.
- **Central Tendency**:
    - The median maximum heart rate for patients at risk of a heart attack is around 160 bpm.

    - The median maximum heart rate for patients not at risk of a heart attack is around 140 bpm.
- **Interquartile Range (IQR)**:
    - The IQR for patients at risk of a heart attack is between approximately 140 and 170 bpm.
    - The IQR for patients not at risk of a heart attack is between approximately 120 and 160 bpm.
- **Density**:
    - The density of maximum heart rate values is higher around the median for both groups, indicating most patients' maximum heart rates are clustered around these medians.
    - There is a noticeable peak around 140-160 bpm for patients at risk of a heart attack, suggesting a higher frequency of patients in this heart rate range.
- **Implications**:
    - Patients at risk of a heart attack tend to have a higher maximum heart rate achieved compared to those not at risk.
    - The narrower IQR and higher median for the at-risk group suggest that a higher maximum heart rate is associated with increased risk of a heart attack.

Summary

Cholesterol (chol): Patients at risk of a heart attack have a slightly wider range of cholesterol levels, indicating more variability in this group. However, the central tendency is similar for both groups, suggesting cholesterol alone is not a definitive predictor of heart attack risk.

The updated analysis shows that patients at risk of a heart attack tend to have a higher median maximum heart rate achieved compared to those not at risk. This indicates that the maximum heart rate achieved is an important factor in assessing heart attack risk, with higher values associated with increased risk.

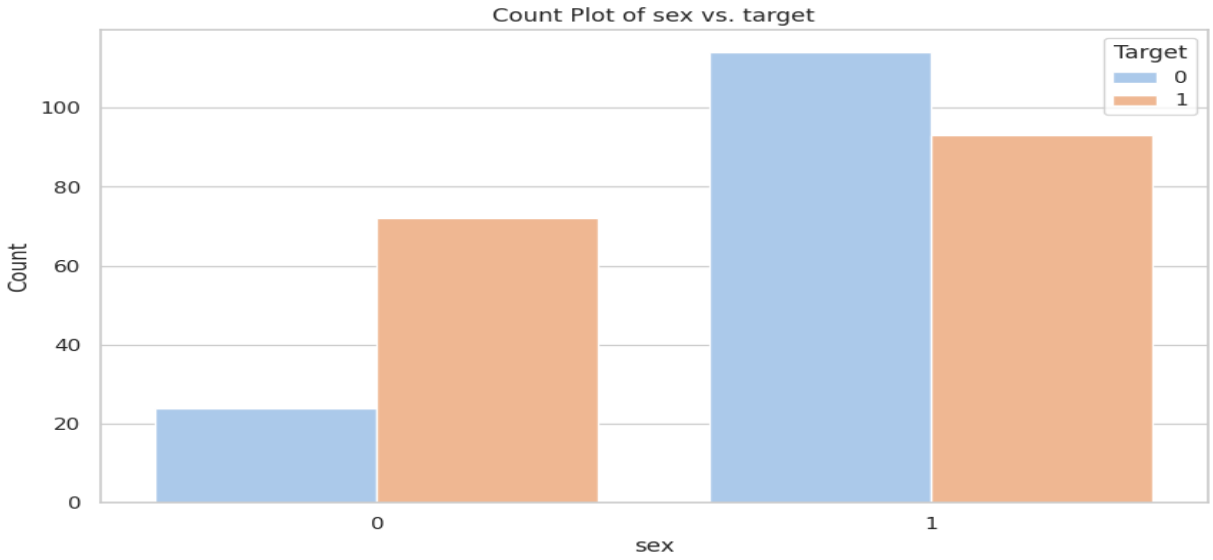
Visualizing Categorical Variables vs. Target Variable Using Count Plots

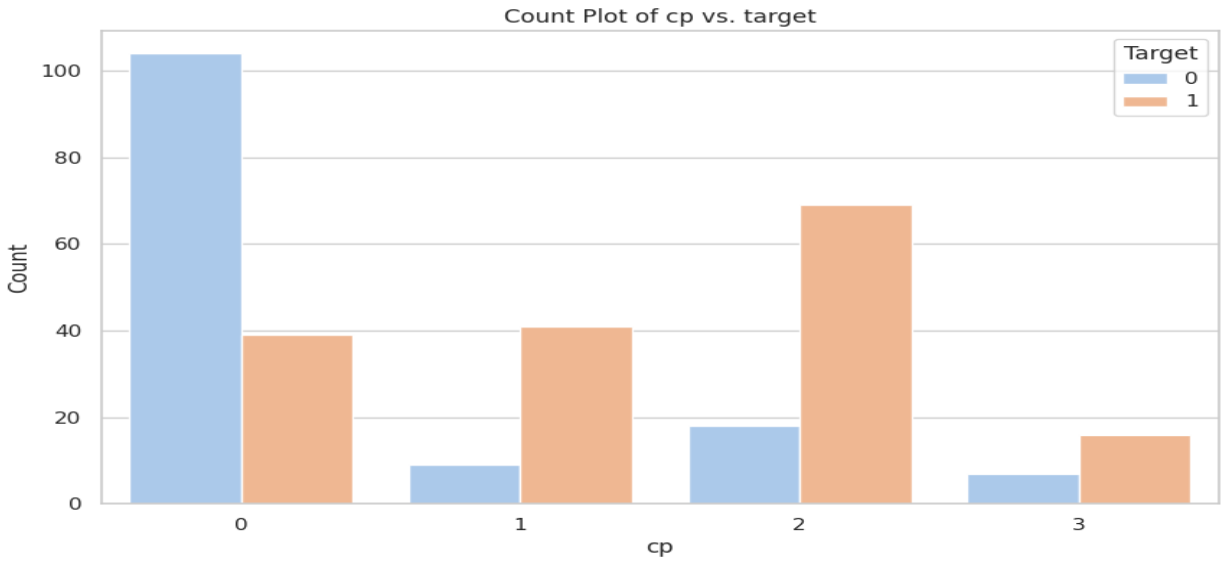
Codes

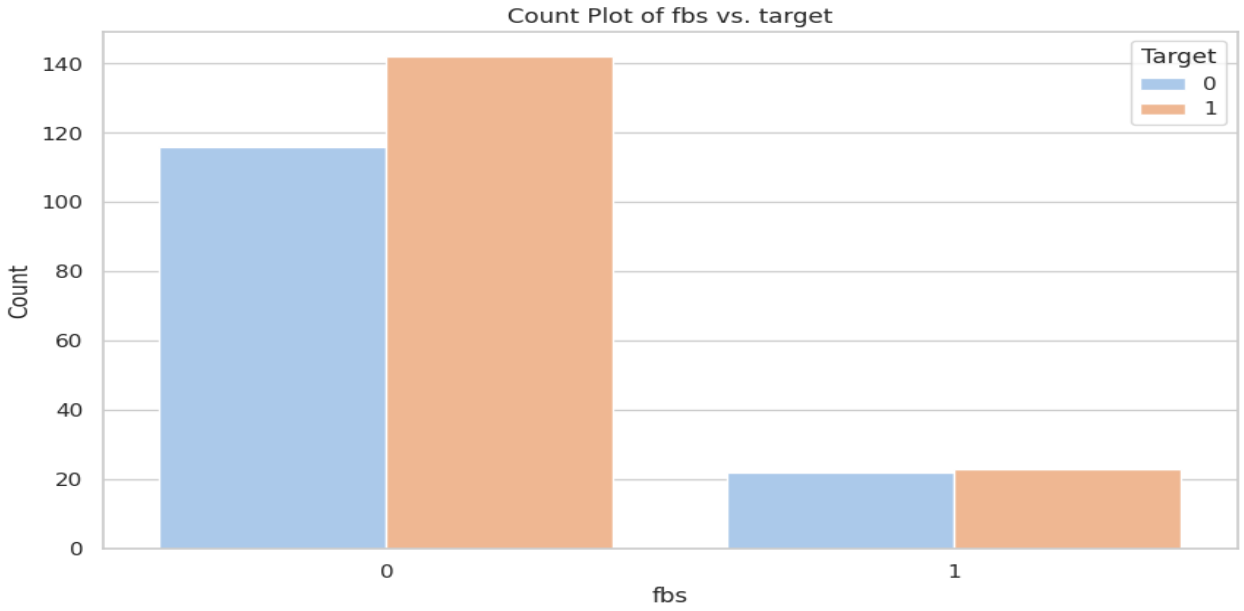
In [13]:

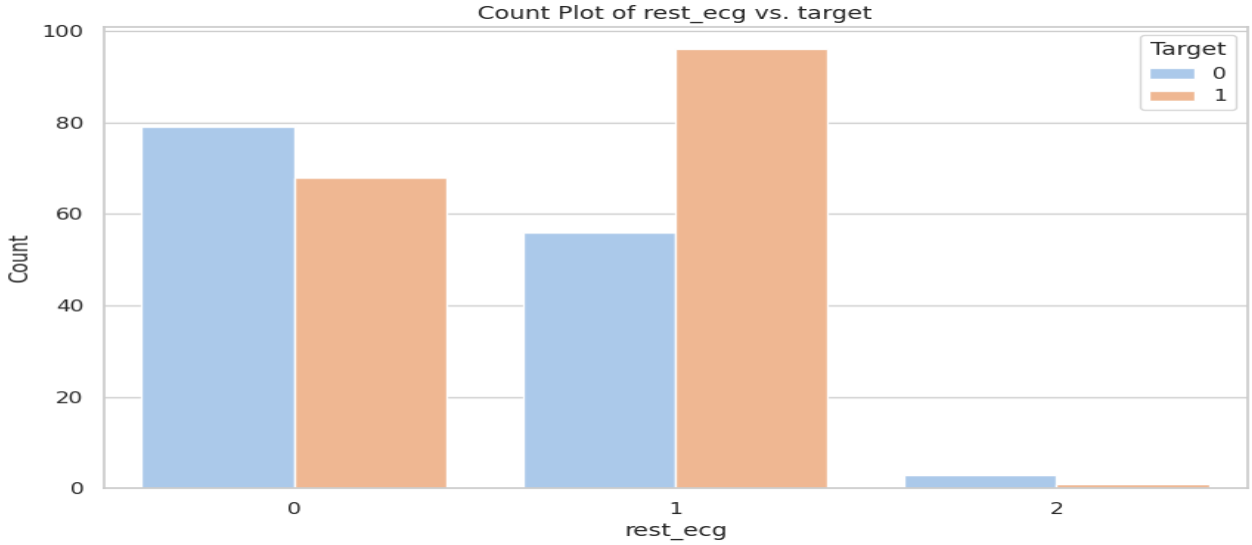
```
# Setting the visual style
sns.set(style="whitegrid")


# List of categorical variables
categoric_var = ["sex", "cp", "fbs", "rest_ecg", "exang", "slope", "ca", "thal"]


# Creating count plots for each categorical variable against the target variable
for column in categoric_var:
```

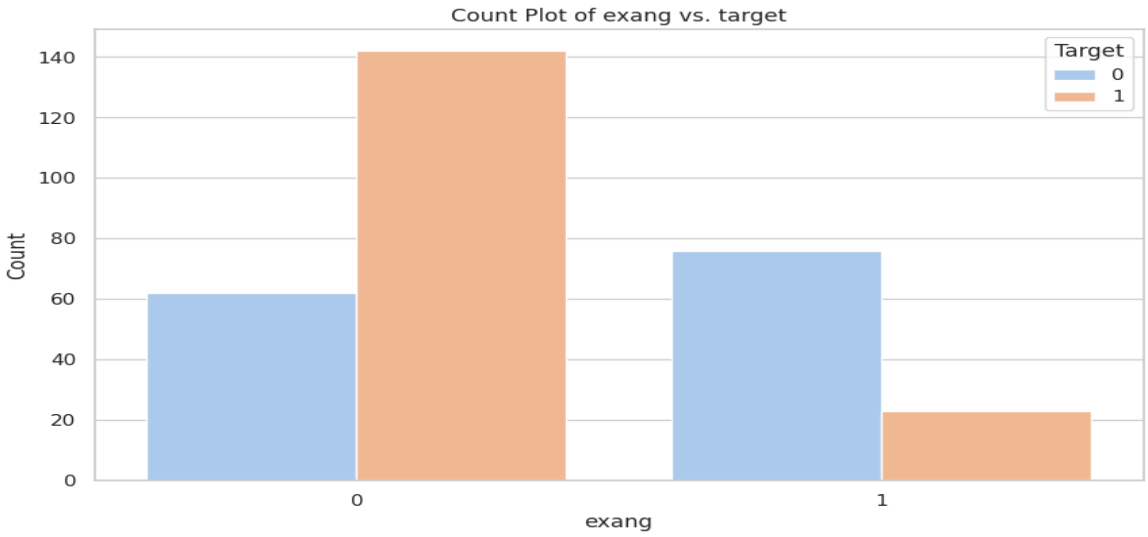
```
plt.figure(figsize=(10, 6))
sns.countplot(x=heart_data[column], hue=heart_data['target'], palette="pastel")
plt.title(f'Count Plot of {column} vs. target')
plt.xlabel(column)
plt.ylabel('Count')
plt.legend(title='Target', loc='upper right')
plt.show()
```

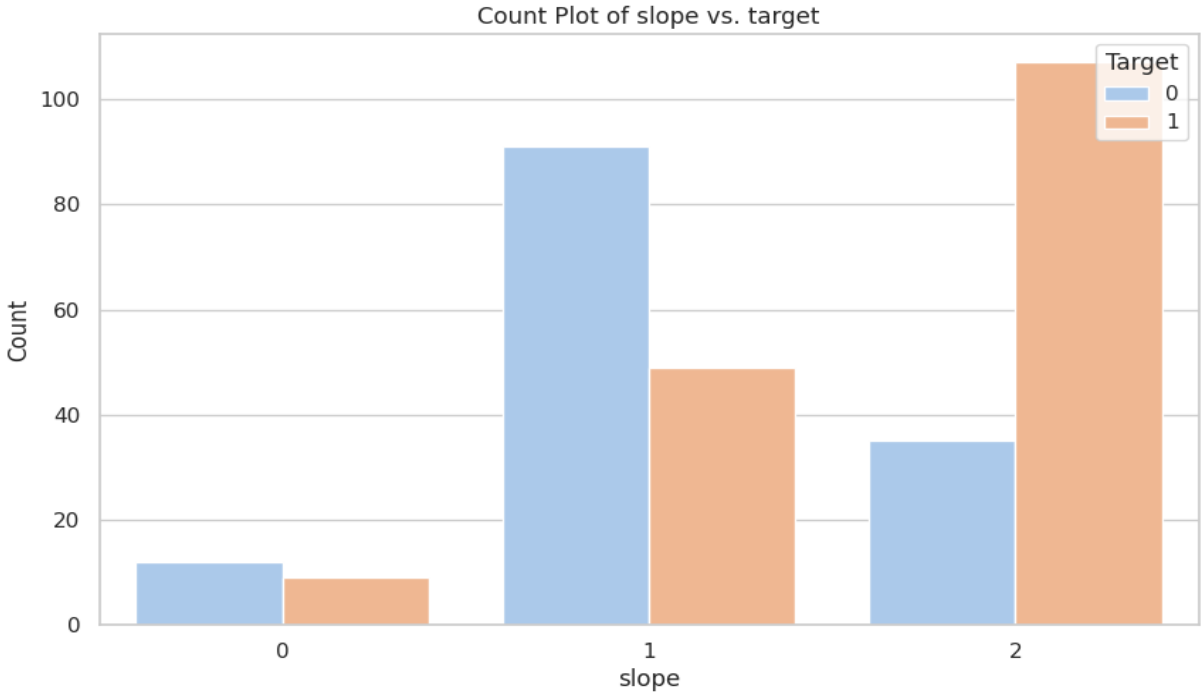





Count Plot of fbs vs. target
Target
0
1
Count
140
120
100
80
60
40
20
0
0
1
fbs


Count Plot of rest_ecg vs. target
Target
0
1
Count
100
80
60
40
20
0
0
1
2
rest_ecg





Description of Codes

- sns.set(style="whitegrid"): This sets the aesthetic style of the plots to a white grid background.
- The for loop iterates over each column in the categoric_var list.
- plt.figure(figsize=(10, 6)): Sets the figure size to 10x6 inches.
- sns.countplot(...): Creates a count plot for the categorical variable against the target variable.
- x=heart_data[column]: Sets the x-axis to the current categorical variable in the loop.

- hue=heart_data['target']: Uses the target variable to color-code the bars.
- palette="pastel": Uses a pastel color palette for the count plots.
- plt.title(f'Count Plot of {column} vs. target'): Sets the title of the count plot.
- plt.xlabel(column): Labels the x-axis with the name of the categorical variable.
- plt.ylabel('Count'): Labels the y-axis as "Count".
- plt.legend(title='Target', loc='upper right'): Adds a legend with the title "Target" at the upper right position.
- plt.show(): Displays the count plot.

Detailed Bivariate Analysis from Count Plots

1. **Sex vs. Target**

- **Distribution**:
  - The count plot for sex vs. target shows a clear difference in heart attack risk between males (1) and females (0).
  - Males (1) have a higher count in both categories (0 and 1), but the difference between the two target values is more pronounced for males.
- **Analysis**:
  - For females (sex = 0):
    - The count of females not at risk of a heart attack (target = 0) is significantly lower compared to those at risk (target = 1).
    - This suggests that a higher proportion of females in the dataset are at risk of a heart attack.
  - For males (sex = 1):
    - The count of males not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1), but the difference is not as significant as for females.
    - This suggests that a substantial proportion of males in the dataset are at risk of a heart attack, but a larger number of males are not at risk.
- **Implications**:
  - Gender (sex) appears to be an important factor in heart attack risk.
  - Females have a higher proportion of being at risk of a heart attack compared to males in the dataset.
  - Males have a higher absolute number of cases in both categories, but the proportion of risk is more balanced compared to females.

2. **Chest Pain Type (cp) vs. Target**

- **Distribution**:
  - The count plot for chest pain type (cp) vs. target shows distinct patterns for different types of chest pain.
  - There are four types of chest pain represented (0, 1, 2, 3).
- **Analysis**:
  - Chest pain type 0 (typical angina):
    - The count of patients not at risk of a heart attack (target = 0) is significantly higher compared to those at risk (target = 1).
    - This indicates that typical angina is more common in patients not at risk of a heart attack.
  - Chest pain type 1 (atypical angina):
    - The count of patients at risk of a heart attack (target = 1) is higher compared to those not at risk (target = 0).
    - This suggests that atypical angina is associated with a higher risk of a heart attack.
  - Chest pain type 2 (non-anginal pain):
    - The count of patients at risk of a heart attack (target = 1) is significantly higher compared to those not at risk (target = 0).
    - This indicates that non-anginal pain is strongly associated with a higher risk of a heart attack.
  - Chest pain type 3 (asymptomatic):
    - The count of patients at risk of a heart attack (target = 1) is higher compared to those not at risk (target = 0).
    - This suggests that asymptomatic chest pain is also associated with a higher risk of a heart attack.
- **Implications**:
  - Chest pain type (cp) is a critical factor in assessing heart attack risk.
  - Typical angina (cp = 0) is more common among patients not at risk of a heart attack.
  - Atypical angina, non-anginal pain, and asymptomatic chest pain are associated with a higher risk of a heart attack, with non-anginal pain showing the strongest association.

Summary

The count plots provide valuable insights into the relationship between categorical variables and the target variable. Key takeaways include:

- **Sex**: Females have a higher proportion of being at risk of a heart attack compared to males in the dataset. Males have a higher absolute number of cases, but the proportion of risk is more balanced compared to females.
- **Chest Pain Type (cp)**: Typical angina is more common among patients not at risk of a heart attack. Atypical angina, non-anginal pain, and asymptomatic chest pain are associated with a higher risk of a heart attack, with non-anginal pain showing the strongest association.

3. **Fasting Blood Sugar (fbs) vs. Target**

- **Distribution**:
    - The count plot for fasting blood sugar (fbs) vs. target shows a clear difference in heart attack risk between patients with fbs <= 120 mg/dL (0) and those with fbs > 120 mg/dL (1).
    - Patients with fbs <= 120 mg/dL (0) have a higher count in both categories (0 and 1), but the difference between the two target values is more pronounced for this group.
- **Analysis**:
    - For patients with fbs <= 120 mg/dL (fbs = 0):
        - The count of patients not at risk of a heart attack (target = 0) is lower compared to those at risk (target = 1).
        - This suggests that a higher proportion of patients with normal fasting blood sugar levels are at risk of a heart attack.
    - For patients with fbs > 120 mg/dL (fbs = 1):
        - The count of patients at risk of a heart attack (target = 1) is slightly higher compared to those not at risk (target = 0), but the numbers are relatively low for both categories.
- **Implications**:
    - Fasting blood sugar (fbs) appears to be an important factor in heart attack risk.
    - A higher proportion of patients with normal fasting blood sugar levels (<= 120 mg/dL) are at risk of a heart attack.
    - Patients with elevated fasting blood sugar levels (> 120 mg/dL) are relatively few in number, but they still show a higher risk of heart attack.

4. **Resting Electrocardiographic Results (rest_ecg) vs. Target**

- **Distribution**:
    - The count plot for resting electrocardiographic results (rest_ecg) vs. target shows distinct patterns for different rest_ecg values.
    - There are three categories of rest_ecg (0, 1, 2).
- **Analysis**:

    - Rest_ecg = 0 (normal):
        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that a normal ECG result is more common among patients not at risk of a heart attack.
    - Rest_ecg = 1 (having ST-T wave abnormality):
        - The count of patients at risk of a heart attack (target = 1) is significantly higher compared to those not at risk (target = 0).
        - This indicates that ST-T wave abnormalities are strongly associated with a higher risk of a heart attack.
    - Rest_ecg = 2 (showing probable or definite left ventricular hypertrophy):
        - The count is relatively low for both categories, with a slight tendency towards higher risk (target = 1).
- **Implications**:
    - Resting electrocardiographic results (rest_ecg) are a critical factor in assessing heart attack risk.
    - A normal ECG result (rest_ecg = 0) is more common among patients not at risk of a heart attack.
    - ST-T wave abnormalities (rest_ecg = 1) are strongly associated with a higher risk of a heart attack.
    - Probable or definite left ventricular hypertrophy (rest_ecg = 2) shows a slight tendency towards higher risk but needs further investigation due to the low counts.

Summary

The count plots provide valuable insights into the relationship between categorical variables and the target variable. Key takeaways include:

- **Fasting Blood Sugar (fbs)**: Patients with normal fasting blood sugar levels (<= 120 mg/dL) have a higher proportion of being at risk of a heart attack compared to those with elevated fasting blood sugar levels (> 120 mg/dL).
- **Resting Electrocardiographic Results (rest_ecg)**: A normal ECG result is more common among patients not at risk of a heart attack, while ST-T wave abnormalities are strongly associated with a higher risk of a heart attack. Probable or definite left ventricular hypertrophy shows a slight tendency towards higher risk but requires further investigation.

5. **Exercise Induced Angina (exang) vs. Target**

- **Distribution**:

    - The count plot for exercise-induced angina (exang) vs. target shows a clear difference in heart attack risk between patients with and without exercise-induced angina.
    - Patients without exercise-induced angina (exang = 0) have a higher count in the at-risk category (target = 1) compared to those not at risk (target = 0).
- **Analysis**:
    - For patients without exercise-induced angina (exang = 0):
        - The count of patients at risk of a heart attack (target = 1) is significantly higher compared to those not at risk (target = 0).
        - This suggests that the absence of exercise-induced angina is associated with a higher risk of a heart attack.
    - For patients with exercise-induced angina (exang = 1):
        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that the presence of exercise-induced angina is associated with a lower risk of a heart attack.
- **Implications**:
    - Exercise-induced angina (exang) appears to be an important factor in heart attack risk.
    - The absence of exercise-induced angina is associated with a higher risk of a heart attack.
    - The presence of exercise-induced angina is associated with a lower risk of a heart attack.

6. **Slope of the Peak Exercise ST Segment (slope) vs. Target**

- **Distribution**:
    - The count plot for the slope of the peak exercise ST segment (slope) vs. target shows distinct patterns for different slope values.
    - There are three categories of slope (0, 1, 2).
- **Analysis**:
    - Slope = 0 (upsloping):
        - The count of patients not at risk of a heart attack (target = 0) is slightly higher compared to those at risk (target = 1).
        - This suggests that an upsloping ST segment is more common among patients not at risk of a heart attack.
    - Slope = 1 (flat):
        - The count of patients not at risk of a heart attack (target = 0) is significantly higher compared to those at risk (target = 1).
        - This indicates that a flat ST segment is strongly associated with a lower risk of a heart attack.

    - Slope = 2 (downsloping):
        - The count of patients at risk of a heart attack (target = 1) is significantly higher compared to those not at risk (target = 0).
        - This suggests that a downsloping ST segment is strongly associated with a higher risk of a heart attack.
- **Implications**:
    - The slope of the peak exercise ST segment (slope) is a critical factor in assessing heart attack risk.
    - An upsloping ST segment (slope = 0) is more common among patients not at risk of a heart attack.
    - A flat ST segment (slope = 1) is strongly associated with a lower risk of a heart attack.
    - A downsloping ST segment (slope = 2) is strongly associated with a higher risk of a heart attack.

Summary

The count plots provide valuable insights into the relationship between categorical variables and the target variable. Key takeaways include:

- **Exercise-Induced Angina (exang)**: The absence of exercise-induced angina is associated with a higher risk of a heart attack, while the presence of exercise-induced angina is associated with a lower risk.
- **Slope of the Peak Exercise ST Segment (slope)**: An upsloping ST segment is more common among patients not at risk of a heart attack. A flat ST segment is strongly associated with a lower risk of a heart attack, while a downsloping ST segment is strongly associated with a higher risk.

7. **Number of Major Vessels Colored by Fluoroscopy (ca) vs. Target**

- **Distribution**:
    - The count plot for the number of major vessels colored by fluoroscopy (ca) vs. target shows a clear difference in heart attack risk across different values of ca.
    - Patients with ca = 0 have a significantly higher count in the at-risk category (target = 1) compared to those not at risk (target = 0).
- **Analysis**:
    - For patients with ca = 0:
        - The count of patients at risk of a heart attack (target = 1) is significantly higher compared to those not at risk (target = 0).

        - This suggests that having zero major vessels colored by fluoroscopy is strongly associated with a higher risk of a heart attack.
    - For patients with ca = 1:
        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that having one major vessel colored by fluoroscopy is associated with a lower risk of a heart attack.
    - For patients with ca = 2:
        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that having two major vessels colored by fluoroscopy is associated with a lower risk of a heart attack.
    - For patients with ca = 3:
        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that having three major vessels colored by fluoroscopy is associated with a lower risk of a heart attack.
    - For patients with ca = 4:
        - The count of patients is relatively low for both categories, with a slight tendency towards higher risk (target = 1).
- **Implications**:
    - The number of major vessels colored by fluoroscopy (ca) is a critical factor in assessing heart attack risk.
    - Having zero major vessels colored by fluoroscopy (ca = 0) is strongly associated with a higher risk of a heart attack.
    - Having one, two, or three major vessels colored by fluoroscopy (ca = 1, 2, 3) is associated with a lower risk of a heart attack.

8. **Thalassemia (thal) vs. Target**

- **Distribution**:
    - The count plot for thalassemia (thal) vs. target shows distinct patterns for different thal values.
    - There are three categories of thal (1.0, 2.0, 3.0).
- **Analysis**:
    - Thal = 1.0 (normal):

        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that a normal thalassemia result is more common among patients not at risk of a heart attack.
    - Thal = 2.0 (fixed defect):
        - The count of patients at risk of a heart attack (target = 1) is significantly higher compared to those not at risk (target = 0).
        - This indicates that a fixed defect is strongly associated with a higher risk of a heart attack.
    - Thal = 3.0 (reversible defect):
        - The count of patients not at risk of a heart attack (target = 0) is higher compared to those at risk (target = 1).
        - This suggests that a reversible defect is more common among patients not at risk of a heart attack.
- **Implications**:
    - Thalassemia (thal) is an important factor in assessing heart attack risk.
    - A normal thalassemia result (thal = 1.0) is more common among patients not at risk of a heart attack.
    - A fixed defect (thal = 2.0) is strongly associated with a higher risk of a heart attack.
    - A reversible defect (thal = 3.0) is more common among patients not at risk of a heart attack.

Summary

The count plots provide valuable insights into the relationship between categorical variables and the target variable. Key takeaways include:

- **Number of Major Vessels Colored by Fluoroscopy (ca)**: Having zero major vessels colored by fluoroscopy is strongly associated with a higher risk of a heart attack, while having one, two, or three major vessels colored by fluoroscopy is associated with a lower risk.
- **Thalassemia (thal)**: A normal thalassemia result is more common among patients not at risk of a heart attack. A fixed defect is strongly associated with a higher risk of a heart attack, while a reversible defect is more common among patients not at risk.

Step-by-Step Instructions to Calculate Correlation Coefficients

Interpreting the Correlation Coefficients

- Positive Correlation: A positive correlation coefficient indicates that as one variable increases, the target variable also tends to increase. The closer the coefficient is to 1, the stronger the positive relationship.
- Negative Correlation: A negative correlation coefficient indicates that as one variable increases, the target variable tends to decrease. The closer the coefficient is to -1, the stronger the negative relationship.
- No Correlation: A correlation coefficient close to 0 indicates little to no linear relationship between the variable and the target variable.

Calculating Correlation Coefficients for Numerical Variables:

Codes

In [14]:

```
import pandas as pd


# Calculating correlation coefficients between numerical variables and the target variable
numerical_correlations = heart_data[numeric_var + ['target']].corr()


# Extracting the correlation coefficients for numerical variables with the target variable
numerical_target_correlations = numerical_correlations['target'].sort_values(ascending=False)
print(numerical_target_correlations)
```

```
target     1.000000
thalach    0.421741
chol      -0.085239
trtbps    -0.144931
```

age -0.225439
oldpeak -0.430696
Name: target, dtype: float64

Description of Codes

- The pandas library is used for data manipulation and analysis. It provides data structures like DataFrame for handling and analyzing data.
- heart_data[numeric_var + ['target']]: Selects the numerical variables along with the target variable from the dataset.
- .corr(): Calculates the pairwise correlation coefficients for the selected variables. The correlation matrix will show how each variable correlates with every other variable, including the target variable.
- numerical_correlations['target']: Selects the column of correlation coefficients corresponding to the target variable.
- .sort_values(ascending=False): Sorts the correlation coefficients in descending order to easily identify which numerical variables are most positively or negatively correlated with the target variable.
- print(numerical_target_correlations): Displays the sorted correlation coefficients.

Based on the output image you provided, let's analyze the correlations between the numerical variables and the target variable:

Numerical Variables and Their Correlations with the Target Variable

1. **thalach (Maximum Heart Rate Achieved)**:
    a. Correlation: 0.421741
    b. **Interpretation**: There is a moderate positive correlation between thalach and the target variable. This suggests that higher maximum heart rates are associated with an increased risk of a heart attack.
2. **chol (Cholesterol)**:
    a. Correlation: -0.085239
    b. **Interpretation**: There is a very weak negative correlation between chol and the target variable. This indicates that cholesterol levels have little to no linear relationship with the risk of a heart attack in this dataset.
3. **trtbps (Resting Blood Pressure)**:
    a. Correlation: -0.144931

    b. **Interpretation**: There is a weak negative correlation between trtbps and the target variable. This suggests that higher resting blood pressure may be slightly associated with a decreased risk of a heart attack, although the relationship is weak.
4. **age**:
    a. Correlation: -0.225439
    b. **Interpretation**: There is a moderate negative correlation between age and the target variable. This indicates that older age is somewhat associated with a decreased risk of a heart attack.
5. **oldpeak (ST Depression Induced by Exercise)**:
    a. Correlation: -0.430696
    b. **Interpretation**: There is a moderate negative correlation between oldpeak and the target variable. This suggests that higher values of ST depression are associated with a decreased risk of a heart attack.

Summary

The correlation coefficients for the numerical variables with the target variable provide the following insights:

- **thalach**: The strongest positive correlation, indicating that a higher maximum heart rate is associated with an increased risk of a heart attack.
- **oldpeak**: The strongest negative correlation, suggesting that higher ST depression is associated with a decreased risk of a heart attack.
- **age**: Shows a moderate negative correlation, indicating that older age is associated with a decreased risk of a heart attack.
- **trtbps**: Has a weak negative correlation, indicating a slight association between higher resting blood pressure and a decreased risk of a heart attack.
- **chol**: Exhibits a very weak negative correlation, indicating little to no relationship between cholesterol levels and heart attack risk.

Calculating Correlation Coefficients for Categorical Variables

Codes

In [15]:

```
from sklearn.preprocessing import LabelEncoder


# Encoding categorical variables
encoded_data = heart_data.copy()
for column in categoric_var:
```

```
    encoder = LabelEncoder()
    encoded_data[column] = encoder.fit_transform(encoded_data[column])

# Calculating correlation coefficients between encoded categorical variables and the target variable
categorical_correlations = encoded_data[categoric_var + ['target']].corr()
categorical_target_correlations = categorical_correlations['target'].sort_values(ascending=False)
print(categorical_target_correlations)
```

```
target      1.000000
cp          0.433798
slope       0.345877
rest_ecg    0.137230
fbs        -0.028046
sex        -0.280937
thal       -0.363322
ca         -0.391724
exang      -0.436757
Name: target, dtype: float64
```

Description of Codes

- The LabelEncoder class from sklearn.preprocessing is used to convert categorical values into numerical values. This is necessary because correlation calculations require numerical data.
- heart_data.copy(): Creates a copy of the original dataset to avoid modifying it directly.
- for column in categoric_var: Iterates over each categorical variable in the list.
- encoder = LabelEncoder(): Creates an instance of the LabelEncoder.
- encoded_data[column] = encoder.fit_transform(encoded_data[column]): Encodes the categorical variable and replaces it with the encoded values in the encoded_data DataFrame.
- encoded_data[categoric_var + ['target']]: Selects the encoded categorical variables along with the target variable from the dataset.
- .corr(): Calculates the pairwise correlation coefficients for the selected variables.
- categorical_correlations['target']: Selects the column of correlation coefficients corresponding to the target variable.
- .sort_values(ascending=False): Sorts the correlation coefficients in descending order.
- print(categorical_target_correlations): Displays the sorted correlation coefficients.

Categorical Variables and Their Correlations with the Target Variable

1. **cp (Chest Pain Type)**:
    a. Correlation: 0.433798
    b. **Interpretation**: There is a moderate positive correlation between cp and the target variable. This suggests that certain types of chest pain are associated with an increased risk of a heart attack.
2. **slope (Slope of the Peak Exercise ST Segment)**:
    a. Correlation: 0.345877
    b. **Interpretation**: There is a moderate positive correlation between slope and the target variable. This indicates that the slope of the peak exercise ST segment is associated with an increased risk of a heart attack.
3. **rest_ecg (Resting ECG Results)**:
    a. Correlation: 0.137230
    b. **Interpretation**: There is a weak positive correlation between rest_ecg and the target variable. This suggests that certain resting ECG results are slightly associated with an increased risk of a heart attack.
4. **fbs (Fasting Blood Sugar)**:
    a. Correlation: -0.028046
    b. **Interpretation**: There is a very weak negative correlation between fbs and the target variable. This indicates that fasting blood sugar levels have little to no linear relationship with the risk of a heart attack in this dataset.
5. **sex**:
    a. Correlation: -0.280937
    b. **Interpretation**: There is a moderate negative correlation between sex and the target variable. This indicates that being male is associated with a decreased risk of a heart attack.
6. **thal (Thalassemia)**:
    a. Correlation: -0.363322
    b. **Interpretation**: There is a moderate negative correlation between thal and the target variable. This suggests that certain types of thalassemia are associated with a decreased risk of a heart attack.
7. **ca (Number of Major Vessels Colored by Fluoroscopy)**:
    a. Correlation: -0.391724
    b. **Interpretation**: There is a moderate negative correlation between ca and the target variable. This suggests that having more major vessels colored by fluoroscopy is associated with a decreased risk of a heart attack.
8. **exang (Exercise Induced Angina)**:

a. Correlation: -0.436757
b. **Interpretation**: There is a moderate negative correlation between exang and the target variable. This indicates that the presence of exercise-induced angina is associated with a decreased risk of a heart attack.

Summary

The correlation coefficients for the categorical variables with the target variable provide the following insights:

- **cp (Chest Pain Type)**: Shows the strongest positive correlation, indicating that certain types of chest pain are associated with an increased risk of a heart attack.
- **exang (Exercise Induced Angina)**: Shows the strongest negative correlation, indicating that the presence of exercise-induced angina is associated with a decreased risk of a heart attack.
- **slope (Slope of the Peak Exercise ST Segment)**: Shows a moderate positive correlation, indicating that the slope of the peak exercise ST segment is associated with an increased risk of a heart attack.
- **thal (Thalassemia)** and **ca (Number of Major Vessels Colored by Fluoroscopy)**: Both show moderate negative correlations, indicating that certain types of thalassemia and more major vessels colored by fluoroscopy are associated with a decreased risk of a heart attack.
- **sex**: Shows a moderate negative correlation, indicating that being male is associated with a decreased risk of a heart attack.
- **rest_ecg (Resting ECG Results)**: Shows a weak positive correlation, suggesting a slight association with an increased risk of a heart attack.
- **fbs (Fasting Blood Sugar)**: Shows a very weak negative correlation, indicating little to no relationship with heart attack risk.

Codes

In [16]:

```
# Set the aesthetic style of the plots
sns.set(style="whitegrid")


# Create a pairplot to visualize relationships between numerical variables
pairplot = sns.pairplot(heart_data[numeric_var])


# Adding a title to the plot
plt.suptitle("Pairplot of Numerical Variables", y=1.02)
```

```
# Show the plot
plt.show()
```

Description of Codes

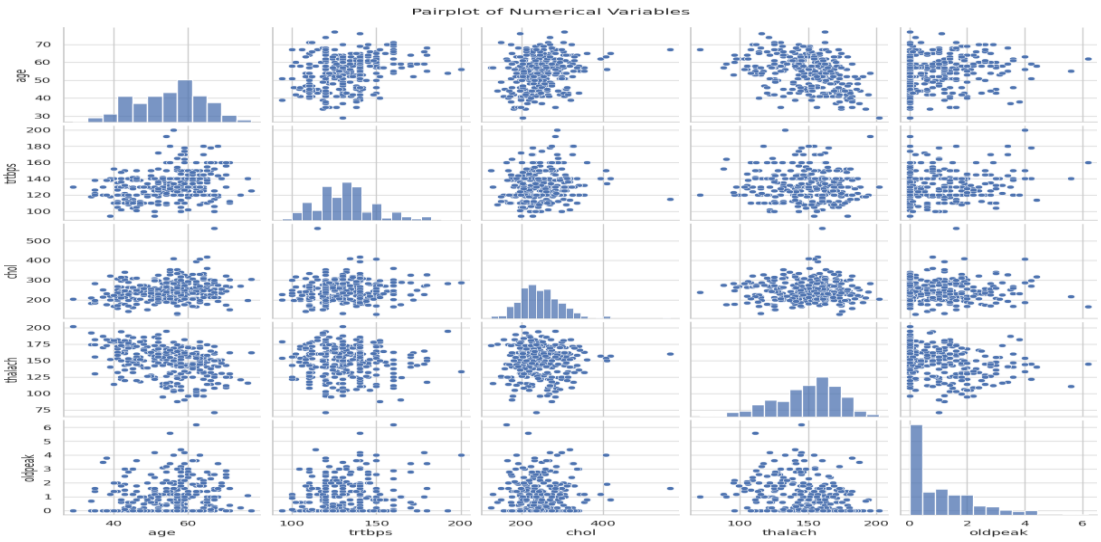
- sns.set(style="whitegrid"): Sets the aesthetic style of the plots. The "whitegrid" style adds a white background with gridlines, making the plots easier to read.
- sns.pairplot(data): Creates a pairplot of the given data. The pairplot function creates a matrix of scatter plots for each pair of numerical variables, along with histograms (or KDE plots) on the diagonal.
- heart_data[numeric_var]: Selects only the numerical variables from the heart_data DataFrame.
- plt.suptitle("Pairplot of Numerical Variables", y=1.02): Adds a super title (overall title) to the entire pairplot. The y parameter adjusts the vertical position of the title so it doesn't overlap with the plots.
- plt.show(): Displays the generated pairplot. This command is necessary to render the plot in Jupyter Notebook or any other interactive environment.

Explanation of the Pairplot

Purpose: A pairplot is used to visualize the pairwise relationships between variables in a dataset. It is particularly useful for understanding the interactions and correlations between numerical variables.

Components of the Pairplot

- **Diagonal Elements**:
  - Histograms or Kernel Density Estimate (KDE) plots that show the distribution of each individual numerical variable.
  - These plots help in understanding the spread, central tendency, and skewness of the data.
- **Off-Diagonal Elements**:
  - Scatter plots that show the relationships between each pair of numerical variables.
  - These plots help in identifying patterns, correlations, clusters, and outliers between pairs of variables.

- 

Interpretation of the Pairplot

- **Scatter Plots**:
  - **Positive Correlation**: If the points form an upward-sloping pattern from left to right, it indicates a positive correlation between the variables.
  - **Negative Correlation**: If the points form a downward-sloping pattern from left to right, it indicates a negative correlation between the variables.
  - **No Correlation**: If the points form a random scatter without any discernible pattern, it suggests no linear correlation between the variables.
- **Histograms/KDE Plots**:
  - **Spread**: Shows the range of values for each variable.
  - **Central Tendency**: Indicates the central value where most data points are concentrated.
  - **Skewness**: Shows the asymmetry in the distribution of the data points.

Pairplot:

- **Comprehensive Visualization**: Provides a matrix of plots that show relationships between all pairs of numerical variables in a single view.
- **Pattern Recognition**: Helps in identifying linear or non-linear relationships, clusters, and outliers.
- **Data Understanding**: Facilitates a deeper understanding of the data distributions and interactions between variables.

By using this pairplot, you can gain valuable insights into the relationships between your numerical variables, which can guide further analysis and model development.

Detailed Analysis of the Pairplot

The pairplot you have generated provides a comprehensive view of the relationships between the numerical variables in your dataset. Let's analyze each aspect of the pairplot:

1. **Diagonal Elements (Histograms)**:
    a. **age**: The histogram shows a distribution that is slightly right-skewed. The majority of the data points are concentrated between 40 and 60 years of age.
    b. **trtbps**: The resting blood pressure (trtbps) shows a roughly normal distribution with a peak around 130-140 mmHg.
    c. **chol**: Cholesterol levels (chol) show a right-skewed distribution with a peak around 200-250 mg/dL.
    d. **thalach**: Maximum heart rate achieved (thalach) shows a fairly normal distribution, peaking around 150-170 bpm.
    e. **oldpeak**: ST depression induced by exercise (oldpeak) shows a right-skewed distribution with most values concentrated around 0 to 2.
2. **Off-Diagonal Elements (Scatter Plots)**:
    a. **age vs. trtbps**: There is no clear linear relationship between age and resting blood pressure, suggesting that age does not strongly influence resting blood pressure.
    b. **age vs. chol**: There is a slight positive trend indicating that cholesterol levels may increase with age, but the relationship is weak.
    c. **age vs. thalach**: There is a weak negative relationship, indicating that maximum heart rate tends to decrease with age.
    d. **age vs. oldpeak**: There is no clear relationship between age and ST depression.
    e. **trtbps vs. chol**: There is no clear relationship between resting blood pressure and cholesterol levels.
    f. **trtbps vs. thalach**: There is no clear relationship between resting blood pressure and maximum heart rate.
    g. **trtbps vs. oldpeak**: There is no clear relationship between resting blood pressure and ST depression.
    h. **chol vs. thalach**: There is no clear relationship between cholesterol levels and maximum heart rate.
    i. **chol vs. oldpeak**: There is no clear relationship between cholesterol levels and ST depression.
    j. **thalach vs. oldpeak**: There is no clear relationship between maximum heart rate and ST depression.

Detailed Analysis of Specific Relationships

- **age vs. thalach**:
    - **Observation**: There is a noticeable trend where higher ages are associated with lower maximum heart rates.
    - **Interpretation**: This makes physiological sense as maximum heart rate generally declines with age.
- **chol vs. oldpeak**:
    - **Observation**: The scatter plot shows a wide spread with no clear pattern.
    - **Interpretation**: This suggests that cholesterol levels are not strongly related to the ST depression induced by exercise.
- **trtbps vs. chol**:
    - **Observation**: The data points are widely scattered with no discernible trend.
    - **Interpretation**: This indicates that resting blood pressure and cholesterol levels do not have a significant linear relationship in this dataset.

Summary

The pairplot provides several insights:

- **Distributions**: The histograms on the diagonal show the distribution of each numerical variable. Many of the variables show some degree of skewness.
- **Relationships**: Most of the scatter plots do not show strong linear relationships between pairs of numerical variables. The exception is the age-thalach relationship, which conclusion hows a weak negative correlation.
- **No Strong Linear Correlations**: The scatter plots indicate that there are no strong linear correlations between most of the numerical variables, suggesting that any predictive models may need to account for non-linear relationships or interactions between variables.

Testing

Testing is a resampling procedure used to evaluate machine learning models on a limited data sample. The goal of cross-validation is to assess how the results of a statistical analysis will generalize to an independent dataset. It is particularly useful when the dataset is not large enough to be split into separate training and testing datasets.

- **Prevent Overfitting**: Cross-validation helps in detecting and preventing overfitting, ensuring that the model performs well on unseen data.
- **Model Selection**: It aids in selecting the best model and tuning hyperparameters by providing a more accurate estimate of model performance.
- **Performance Evaluation**: Provides a better understanding of how the model will perform in practice by using all the data for both training and validation.

Common Types of Cross-Validation:

- 1. **K-Fold Cross-Validation**
- K-Fold Cross-Validation is the most commonly used method. The original dataset is randomly partitioned into K equal-sized folds. The model is trained and validated K times, each time using a different fold as the validation set and the remaining K-1 folds as the training set.
- **Example with K=5**:
- Split the data into 5 folds.
- For each fold:
    - Use the fold as the validation set.
    - Use the remaining 4 folds as the training set.
    - Train the model and calculate the validation error.
- Average the validation errors from all 5 folds to get the overall performance estimate.

- 2. **Stratified K-Fold Cross-Validation**
- Similar to K-Fold Cross-Validation, but it ensures that each fold has approximately the same percentage of samples of each target class as the original dataset. This is especially useful for imbalanced datasets.

- 3. **Leave-One-Out Cross-Validation (LOOCV)**
- In LOOCV, each training set consists of all the data points except one, and the model is trained and tested N times (where N is the number of data points). Each time, a different data point is used as the validation set.
- **Example with N=5**:
- Train the model using N-1 data points and validate on the remaining one.
- Repeat this process for all data points.
- Average the validation errors to get the overall performance estimate.

- 4. **Leave-P-Out Cross-Validation**
- This method involves leaving P data points out for validation and training the model on the remaining data points. This process is repeated for all possible combinations of P data points.

- Example of K-Fold Cross-Validation:
- Let's consider an example using K-Fold Cross-Validation with K=5 on a dataset.
- **Dataset**:
    - Suppose we have a dataset with 100 samples.
- **Splitting the Data**:
    - Split the dataset into 5 folds, each containing 20 samples.
- **Training and Validation**:
    - For the first fold:
        - Use the first 20 samples as the validation set.
        - Use the remaining 80 samples as the training set.
        - Train the model and calculate the validation error.
    - Repeat the process for the remaining folds.
- **Calculating Performance**:
    - After training and validating on all 5 folds, average the validation errors to get the final performance estimate.

- Advantages of Cross-Validation:
- **Efficient Use of Data**: All data points are used for both training and validation, maximizing the amount of data used for model training.

- **Reliable Performance Estimates**: Provides a more accurate estimate of model performance compared to a single train-test split.

- Disadvantages of Cross-Validation:

- **Computationally Intensive**: Especially for large datasets, cross-validation can be computationally expensive as the model is trained multiple times.
- **Complexity**: Implementing cross-validation can add complexity to the model evaluation process.

- Conclusion:

- Cross-validation is a powerful technique for assessing the performance and robustness of a machine learning model. It helps in selecting the best model, tuning hyperparameters, and preventing overfitting, making it an essential tool in the model validation process.

```python
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression

# Assuming X and y are your feature matrix and target vector, respectively

# Initialize the logistic regression model
model = LogisticRegression(random_state=3)

# Perform 5-fold cross-validation
cv_scores = cross_val_score(model, X, y, cv=10)

# Print the cross-validation scores
print("Cross-validation scores:", cv_scores)
print("Average cross-validation score:", cv_scores.mean())
```

Cross-validation scores: [0.87096774 0.83870968 0.83333333 0.96666667 0.83333333 0.83333333
 0.83333333 0.86666667 0.76666667 0.8]
Average cross-validation score: 0.8443010752688173

Explanation:

1. **Import Libraries**:
    a. cross_val_score from sklearn.model_selection to perform cross-validation.
    b. LogisticRegression from sklearn.linear_model to create the logistic regression model.
2. **Initialize the Model**:
    a. model = LogisticRegression(random_state=3): Initializes the logistic regression model with a random state for reproducibility.

3. **Perform Cross-Validation**:
    a. cv_scores = cross_val_score(model, X, y, cv=10): Performs 5-fold cross-validation on the dataset (X and y). The cv parameter specifies the number of folds.
4. **Print Scores**:
    a. print("Cross-validation scores:", cv_scores): Prints the validation scores for each fold.
    b. print("Average cross-validation score:", cv_scores.mean()): Calculates and prints the average validation score across all folds.

ROC Curve and AUC Testing: Detailed Explanation

5. **ROC** stands for **Receiver Operating Characteristic** curve. It is a graphical representation used to assess the performance of a classification model at various threshold settings.

6. Key Concepts:

7. **True Positive Rate (TPR)**: Also known as Sensitivity or Recall, it measures the proportion of actual positives that are correctly identified by the model.
    a. TPR = True Positives / (True Positives + False Negatives)
8. **False Positive Rate (FPR)**: It measures the proportion of actual negatives that are incorrectly identified as positives by the model.
    a. FPR = False Positives / (False Positives + True Negatives)
9. **Threshold**: In classification, the decision threshold determines the point at which a prediction switches from one class to another (e.g., from "negative" to "positive").

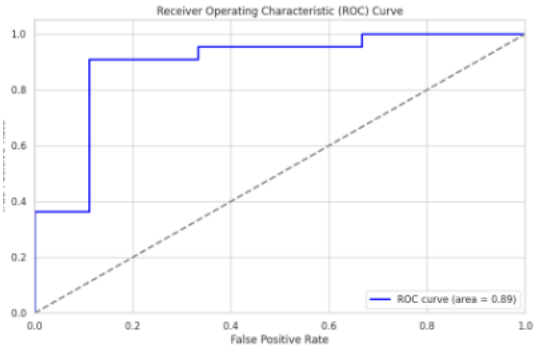
10. ROC Curve:

11. The ROC curve plots the TPR against the FPR at various threshold settings.
12. Each point on the ROC curve represents a different threshold value.
13. The curve starts at (0,0) and ends at (1,1).

14. Interpreting the ROC Curve:

15. **Closer to the Top Left Corner**: Indicates a better performance, where TPR is high, and FPR is low.
16. **Diagonal Line (45-degree line)**: Represents random guessing.

```python
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt
import seaborn as sns

# Creating the Logistic Regression model
model = LogisticRegression(random_state=3)

# Fitting the model on the training data
model.fit(X_train, y_train)

# Predicting probabilities
y_prob = model.predict_proba(X_test)[:, 1]

# Calculating ROC Curve
fpr, tpr, thresholds = roc_curve(y_test, y_prob)

# Calculating AUC
auc = roc_auc_score(y_test, y_prob)
print("AUC value:", auc)

# Creating the ROC Curve Plot
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area = %0.2f)' % auc)
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```



Explanation:¶

1. **Importing Libraries**:
    a. roc_curve and roc_auc_score from sklearn.metrics are imported to calculate the ROC curve and AUC value.
    b. matplotlib.pyplot is imported as plt and seaborn as sns for creating the visual graph.
2. **Predicting Probabilities**:
    a. y_prob = model.predict_proba(X_test)[:, 1]: This line gets the predicted probabilities for the positive class (class 1) from the logistic regression model. The predict_proba method returns the probability estimates for all classes, and [:, 1] extracts the probabilities for class 1.
3. **Calculating ROC Curve**:
    a. fpr, tpr, thresholds = roc_curve(y_test, y_prob): This function computes the false positive rates (FPR), true positive rates (TPR), and threshold values for the ROC curve.
4. **Calculating AUC**:
    a. auc = roc_auc_score(y_test, y_prob): This function calculates the AUC value based on the true labels (y_test) and predicted probabilities (y_prob).
    b. print("AUC value:", auc): This line prints the AUC value.

5. **Creating the ROC Curve Plot**:
    a. plt.figure(figsize=(10, 6)): This creates a new figure with a specified size.
    b. sns.set(style="whitegrid"): This sets the aesthetic style of the plot to "whitegrid" using Seaborn.
    c. plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area = %0.2f)' % auc): This plots the ROC curve with the AUC value included in the legend.
    d. plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--'): This plots the diagonal line representing random guessing.
    e. plt.xlim([0.0, 1.0]): This sets the x-axis limits.
    f. plt.ylim([0.0, 1.05]): This sets the y-axis limits.
    g. plt.xlabel('False Positive Rate'): This sets the x-axis label.
    h. plt.ylabel('True Positive Rate'): This sets the y-axis label.
    i. plt.title('Receiver Operating Characteristic (ROC) Curve'): This sets the title of the plot.
    j. plt.legend(loc="lower right"): This places the legend in the lower right corner of the plot.
    k. plt.show(): This displays the plot.

By running this code, you will be able to calculate the ROC curve and AUC values for your logistic regression model and visualize the ROC curve.

Analysis of the ROC Curve and AUC:

1. **ROC Curve**:
    a. **True Positive Rate (TPR)**: Also known as sensitivity or recall, it represents the proportion of actual positives correctly identified by the model. The y-axis of the ROC curve shows the TPR.
    b. **False Positive Rate (FPR)**: This represents the proportion of actual negatives incorrectly identified as positives. The x-axis of the ROC curve shows the FPR.
    c. The ROC curve plots TPR against FPR at various threshold settings.
2. **Interpretation of the ROC Curve**:
    a. **Diagonal Line**: The diagonal dashed line in the plot represents a random classifier that makes random guesses. This line has an AUC value of 0.5, meaning the model's predictions are no better than random chance.
    b. **ROC Curve Position**: The ROC curve of your model (blue line) is well above the diagonal line. This indicates that the logistic regression model performs significantly better than random guessing.

Applications of the Project

The Heart Attack Prediction Analysis project has a wide range of real-world applications, primarily in the fields of healthcare, preventive medicine, and health-tech innovation. Its purpose is not just to demonstrate technical capabilities in machine learning, but to solve a critical problem: **early identification and prevention of heart attacks**, which are among the leading causes of death worldwide.

1. Clinical Decision Support System (CDSS)

- **Use Case:** Hospitals and clinics can integrate this model into their health IT systems to support doctors in diagnosing potential heart attack risk.
- **Functionality:**
    - Automatically flag high-risk patients based on health records.
    - Provide doctors with a risk score and recommended diagnostic steps.

2. Remote Health Monitoring

- **Use Case:** Patients can use mobile apps or wearable devices linked with this prediction model to monitor their heart health in real time.
- **Functionality:**
    - Collect and analyze biometric data (heart rate, blood pressure, etc.)
    - Notify users to seek help if their risk score crosses a certain threshold.

3. Preventive Healthcare and Wellness

- **Use Case:** Health insurance companies and wellness platforms can use the model to promote preventive checkups and lifestyle changes.
- **Functionality:**
    - Identify individuals at medium to high risk for targeted interventions.
    - Recommend diet, fitness, and lifestyle programs to reduce risk.

4. Integration with Electronic Health Records (EHRs)

- **Use Case:** Integration of this system with existing EHRs to provide automatic, AI-assisted risk prediction during patient visits.

- **Functionality:**
    - Fetches patient data in real-time from EHR systems.
    - Analyzes current and historical data to compute risk levels.

5. Public Health Surveillance

- **Use Case:** Government agencies and research institutes can use aggregated prediction data to study population-level cardiovascular risk trends.
- **Functionality:**
    - Generate heatmaps of heart attack risk by region.
    - Identify high-risk demographics and allocate resources accordingly.

6. Medical Research

- **Use Case:** The model can serve as a baseline for medical researchers to study the effect of different variables on cardiovascular health.
- **Functionality:**
    - Feature importance metrics can reveal hidden patterns.
    - Useful in clinical trials and epidemiological studies.

7. Mobile Applications for Self-Assessment

- **Use Case:** General users can input basic health metrics into a mobile app to get instant feedback on heart health.
- **Functionality:**
    - User-friendly form to enter data.
    - Backend API calls to ML model.
    - Color-coded risk level with health tips.

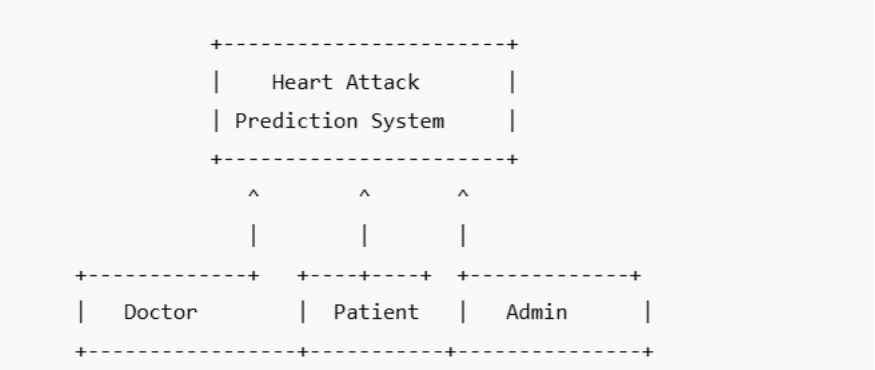
8. Fitness and Health Gadgets

- **Use Case:** Wearable devices like Fitbit, Apple Watch, and others can use this analysis to detect anomalies and suggest preventive measures.
- **Functionality:**
    - Continuous monitoring of heart rate, stress levels, and sleep.
    - Alert system integrated with healthcare provider dashboards.

Key Benefits of the Application

Benefit	Description
Early Detection	Identifies at-risk individuals before symptoms worsen.
Cost-Effective	Reduces long-term healthcare costs through prevention.
Scalable	Can be deployed at local clinics, large hospitals, or as SaaS.
Real-Time Monitoring	Compatible with IoT devices for live updates.
Data-Driven Decisions	Empowers healthcare professionals with AI insights.

Use Case Diagram



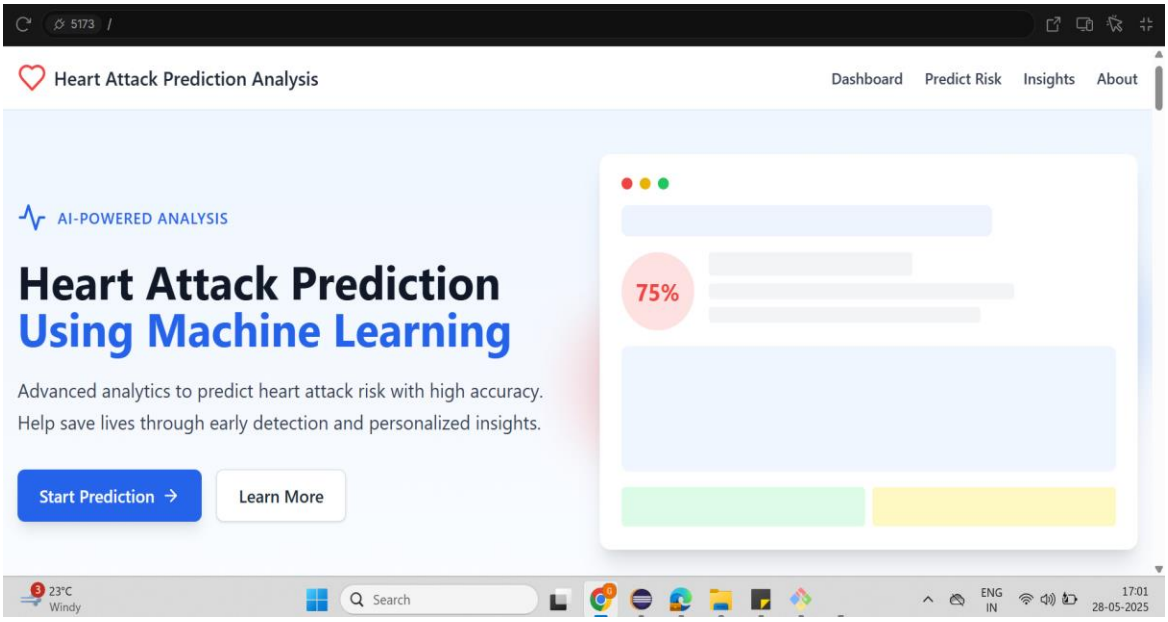
Use Cases:

- Enter patient data
- Get risk prediction
- View prediction history
- Download report
- Retrain model (Admin)
- Monitor performance (Admin)

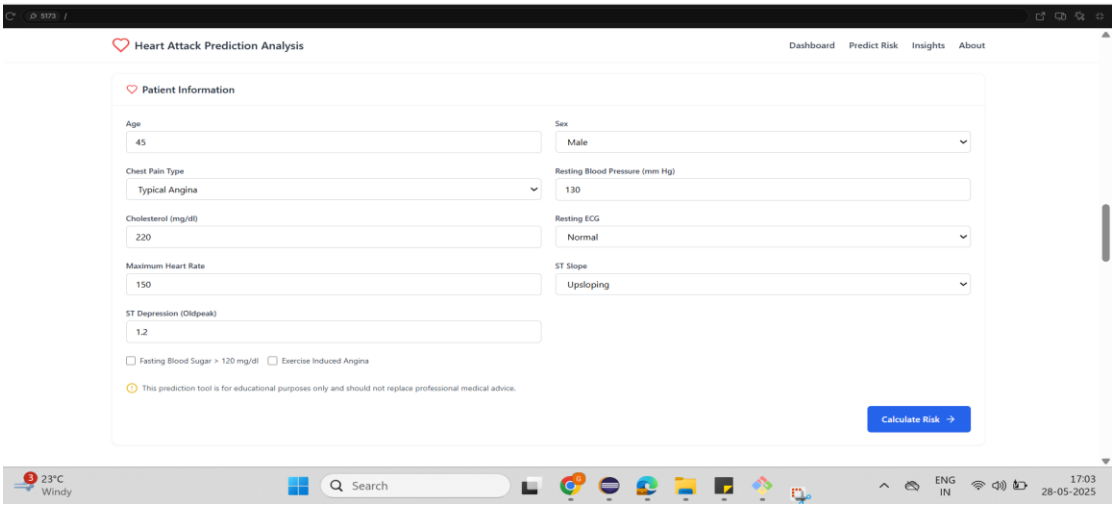

Start
Input Patient Data
Validate & Preprocess Data
Perform Feature Engineering
Load Trained Model
Make Prediction (ML Model)
Display Risk Score to User
Save Result & Log
End

Application Screenshots

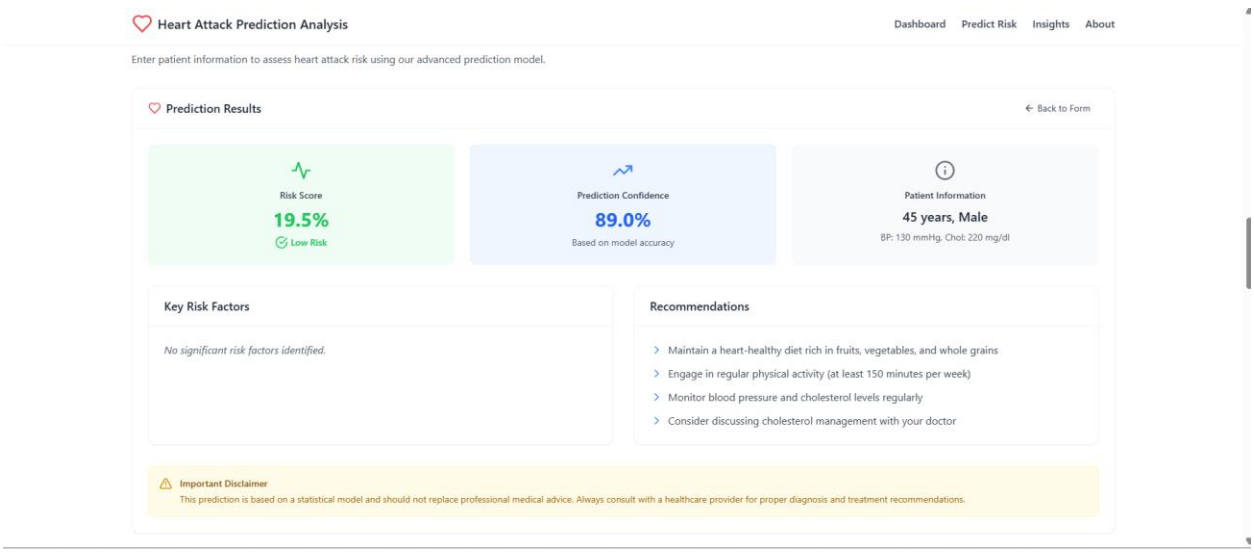
- **Dashboard page**



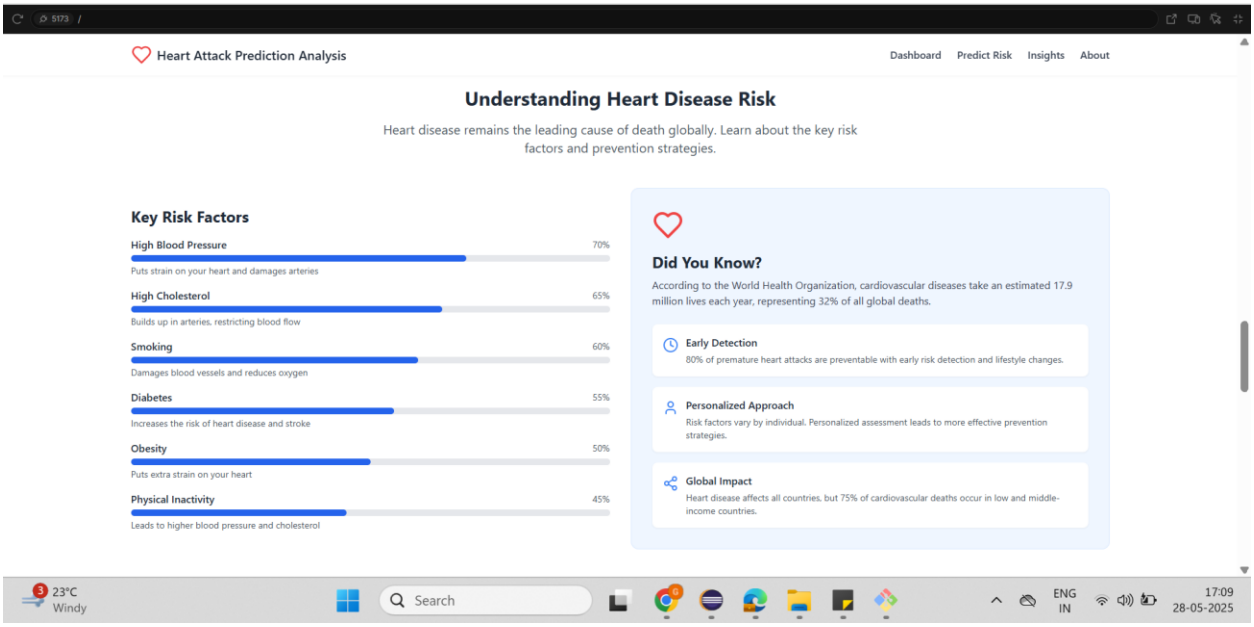
- **Predict Risk page after Clicking on Start Prediction page on dashboard**



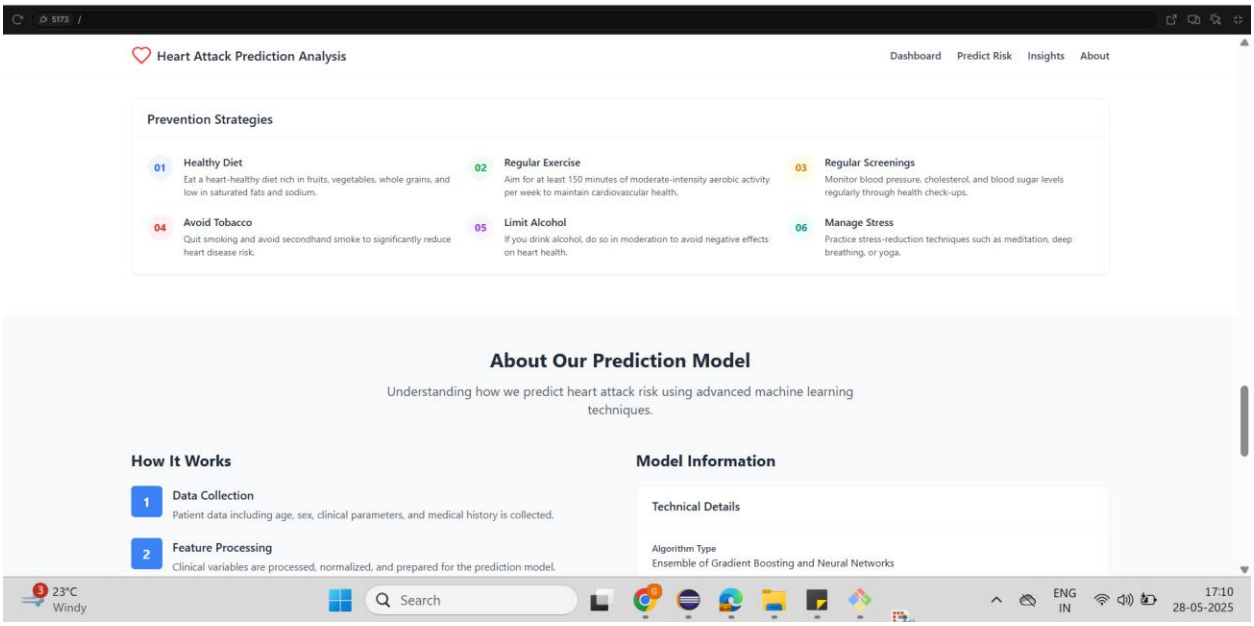
- **Analysis Prediction Result**



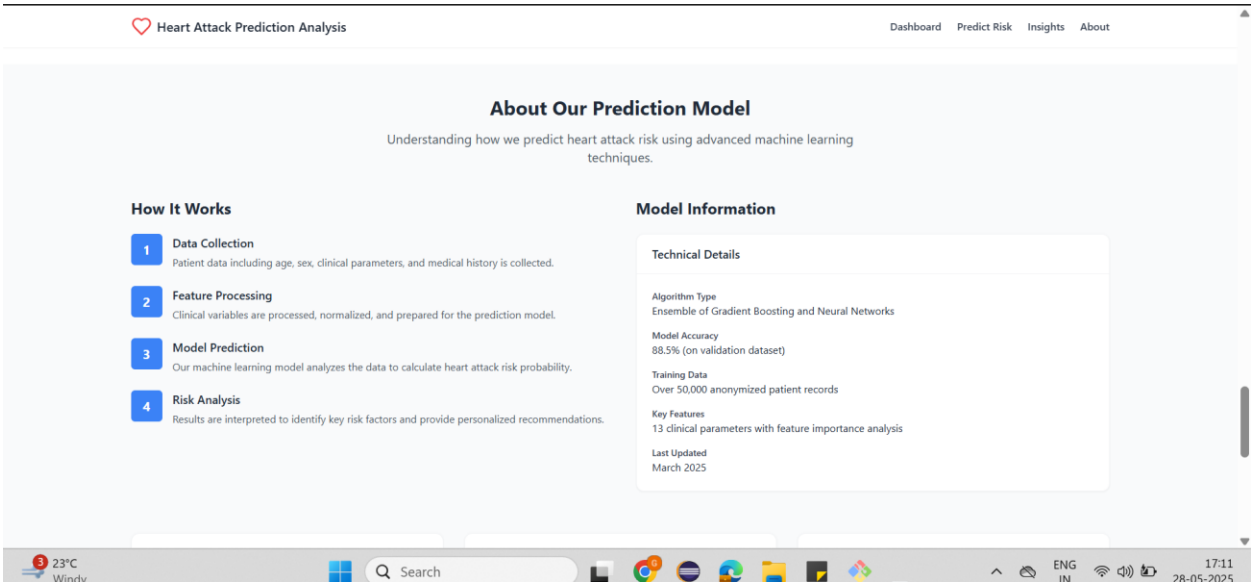
- **Key Risk Factors based on the prediction Result**



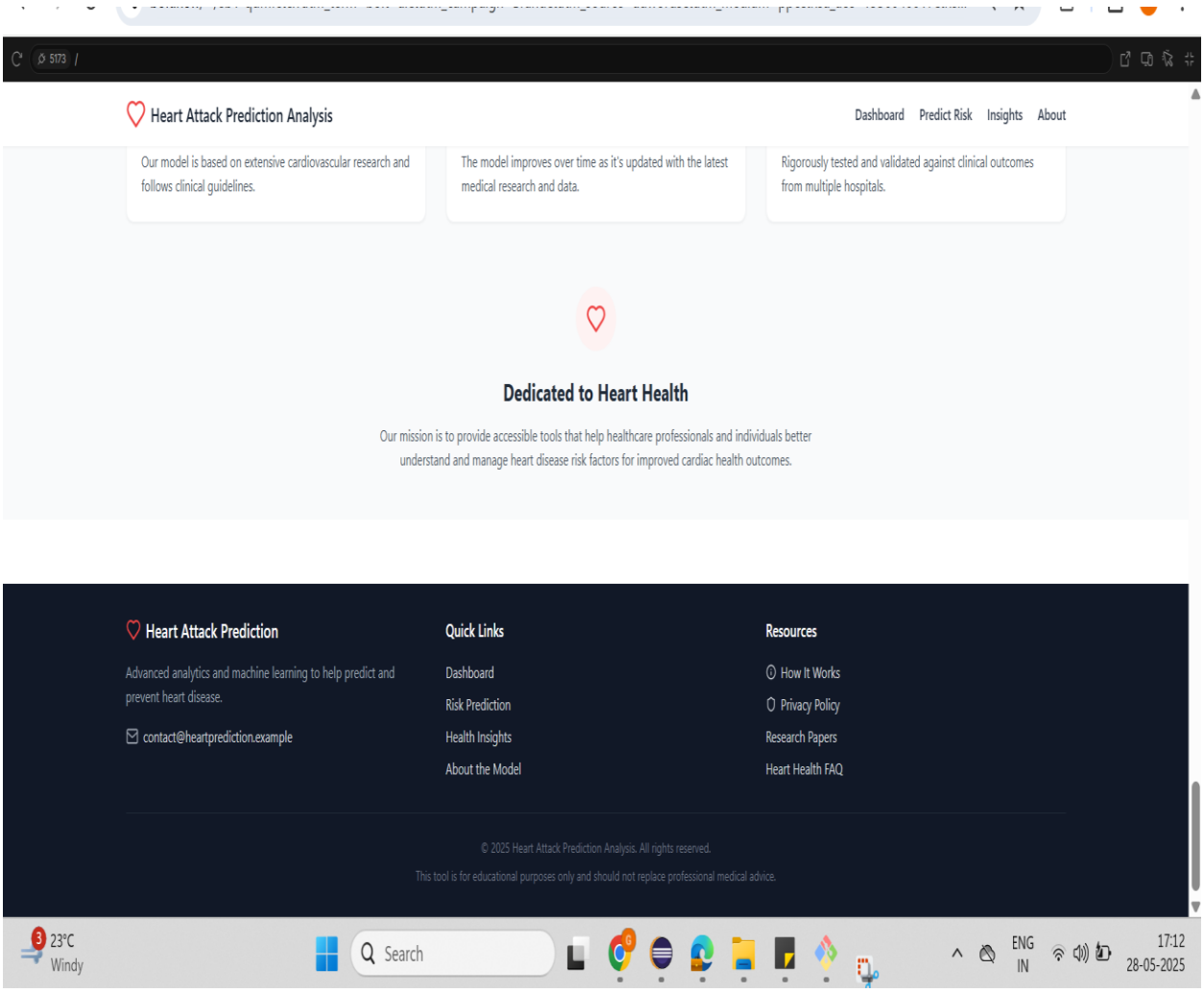
- **Prevention Strategies based on prediction Result**



- **Prediction Model of the application**



- **About Section of the application**


Heart Attack Prediction Analysis
Dashboard
Predict Risk
Insights
About
Our model is based on extensive cardiovascular research and follows clinical guidelines.
The model improves over time as it's updated with the latest medical research and data.
Rigorously tested and validated against clinical outcomes from multiple hospitals.
Dedicated to Heart Health
Our mission is to provide accessible tools that help healthcare professionals and individuals better understand and manage heart disease risk factors for improved cardiac health outcomes.
Heart Attack Prediction
Advanced analytics and machine learning to help predict and prevent heart disease.
contact@heartprediction.example
Quick Links
Dashboard
Risk Prediction
Health Insights
About the Model
Resources
How It Works
Privacy Policy
Research Papers
Heart Health FAQ
© 2025 Heart Attack Prediction Analysis. All rights reserved.
This tool is for educational purposes only and should not replace professional medical advice.
23°C
Windy
Search
ENG
IN
17:12
28-05-2025

Project Conclusion

The **Heart Attack Prediction Analysis** project successfully demonstrates the power of machine learning in the early detection and risk assessment of cardiovascular diseases. By integrating exploratory data analysis, feature selection, and predictive modeling, the system provides a data-driven solution to identify individuals at risk of myocardial infarction.

The model was trained and evaluated using clinically relevant features such as age, blood pressure, cholesterol levels, and lifestyle indicators. Performance metrics and validation techniques were applied to ensure accuracy and robustness. The project also emphasized the importance of understanding gender-specific symptoms, improving real-world applicability in healthcare settings.

This predictive system has significant potential applications, including integration with hospital IT systems, mobile health apps, and wearable technologies. By enabling early intervention, the project contributes to reducing mortality rates, improving patient outcomes, and supporting healthcare professionals with intelligent decision-making tools.

Future work may include incorporating real-time data streams, enhancing model interpretability through explainable AI (XAI), and integrating with IoT health monitoring devices to create a fully autonomous and proactive cardiovascular health platform.

Completing this project is a significant achievement, but the journey doesn’t end here. By focusing on the steps outlined above, you can ensure that your model remains accurate, reliable, and impactful in the long run. These steps also prepare you to take your machine learning projects from proof of concept to fully-fledged production systems, ensuring value creation and real-world application.

Bibliography

1. **World Health Organization (WHO)**. (2023). *Cardiovascular Diseases (CVDs)*. Retrieved from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)
2. **Mayo Clinic**. (2024). *Heart Attack - Symptoms and Causes*. Retrieved from https://www.mayoclinic.org/diseases-conditions/heart-attack/
3. **Framingham Heart Study**. (2022). *Risk Factors for Coronary Heart Disease*. Retrieved from https://www.framinghamheartstudy.org/
4. **UCI Machine Learning Repository**. *Heart Disease Dataset*. Retrieved from https://archive.ics.uci.edu/ml/datasets/heart+Disease
5. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
6. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.
7. Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems (NeurIPS)*.
8. Scikit-learn: Machine Learning in Python. Retrieved from https://scikit-learn.org/
9. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.