**Heart Attack Risk Predictor Project**

Project Overview

The Heart Attack Risk Predictor project is designed to assist in forecasting an individual's risk of experiencing a heart attack using various machine learning algorithms. This documentation will provide insights into the project's implementation and technical details.

Project Objectives

1. Data Analysis

2. Feature Engineering

3. Data Standardization

4. Model Building

5. Predictions

**Data Analysis**

In this section, we analyze and understand our dataset. Key points include:

Data Analysis

In this section, we analyze and understand our dataset. Key points include:

Age: Age of the patient

Sex: Sex of the patient

exang: Exercise-induced angina

ca: Number of major vessels

cp: Chest Pain type

trtbps: Resting blood pressure

chol: Cholesterol levels

fbs: Fasting blood sugar

rest\_ecg: Resting electrocardiographic results

thalach: Maximum heart rate achieved

target: 0 for less chance of a heart attack, 1 for more chance

**Data Preprocessing**

Data preprocessing is the initial step in data analysis or machine learning projects. Its purpose is to clean, organize, and prepare the data for analysis. Here's what these steps involve:

* **Data Cleaning**: This step includes handling missing data, correcting inconsistencies, and removing outliers that might negatively impact the analysis or models.
* **Data Transformation**: This involves converting data into a format that's more suitable for analysis. Common transformations include encoding categorical variables, scaling numerical features, or creating new features from existing ones.
* **Data Reduction**: If your dataset is too large or contains many irrelevant features, you may reduce its dimensionality through techniques like feature selection or extraction.

**Data Shape**

Number of Rows: This tells you how many data points or examples are in your dataset.

Number of Columns: This indicates how many features or attributes are available for analysis.

Data Types: You may check the data types of each feature, which could be numerical (integer or float) or categorical (text or nominal).

Understanding the data shape helps you get a quick overview of your dataset's size and complexity.

**Missing Value Check**

It's crucial to identify and handle missing values because they can cause issues during analysis or modeling. In this step, we will:

* Check each feature for missing values.
* Calculate the number or percentage of missing values for each feature.
* Decide on a strategy to address missing data, which may include imputing missing values or removing rows or columns with too many missing values.

Handling missing data is essential to ensure the quality of our analysis or models. Common techniques for handling missing data include mean imputation, median imputation, or using more advanced methods like regression imputation.

**Correlation Matrix**

A correlation matrix is a table showing the correlation coefficients between many variables. It helps you understand the relationships between pairs of features. Common points about the correlation matrix include:

* **Correlation Coefficient**: Each cell in the matrix contains a value between -1 and 1, which quantifies the strength and direction of the linear relationship between two variables. A positive value (closer to 1) indicates a positive correlation, while a negative value (closer to -1) indicates a negative correlation.
* **Visual Representation**: A heatmap is often used to visualize the correlation matrix, making it easier to spot patterns and relationships between variables.
* **Use Cases**: Correlation matrices are commonly used in feature selection, identifying multicollinearity (when multiple variables are highly correlated), and understanding which variables might be important for predictive modeling.

**Data Visualization**

**Age Distribution**

In this part, we explore the distribution of ages within the dataset. Visualizing the age distribution can provide insights into the age groups represented in the data. This is important because the risk of heart disease often varies with age.

**Plot Type**: A histogram is used to create a graphical representation of the age distribution.

**Insights**: By observing the histogram, it's possible to identify the predominant age groups in the dataset. In your specific dataset, it's observed that the majority of patients fall within the age group of 51 to 67 years.

**Gender Distribution**

This visualization aims to show the distribution of gender within the dataset, specifically distinguishing between male and female patients.

**Plot Type**: A countplot is used to create a bar chart to visualize the distribution of gender.

**Insights**: By analyzing the countplot, it's clear how many male and female patients are represented in the dataset. In your dataset, the code distinguishes between 0 (female) and 1 (male) gender labels.

**Chest Pain Types**

Understanding the distribution of chest pain types can provide insights into the different categories of chest pain experienced by patients.

**Plot Type**: A bar plot is created to visualize the distribution of chest pain types.

**Insights**: The bar plot represents the four types of chest pain: Typical Angina, Atypical Angina, Non-Anginal Pain, and Asymptomatic. This visualization helps in understanding the prevalence of each type of chest pain in the dataset.

**Resting ECG Results**

This visualization focuses on the distribution of resting electrocardiographic results (ECG) within the dataset. It categorizes ECG results into different classes.

**Plot Type**: A bar plot is used to display the distribution of ECG results.

**Insights**: The bar plot shows the count of patients with different ECG results, such as 'Normal,' 'ST-T Wave Abnormality,' and 'Probable or Definite Left Ventricular Hypertrophy by Estes.' This information helps understand the distribution of ECG results in the dataset.

**Pair Plot for Features**

The pair plot visualizes the relationships between pairs of features (variables) in the dataset. This is a powerful tool to gain insights into feature interactions and potential correlations.

**Plot Type**: A pairplot is created, allowing you to observe scatterplots for feature pairs and histograms for individual features.

**Insights**: By examining the pair plot, you can identify patterns and correlations between features. It's especially useful when 'output' is used as the hue parameter, allowing you to distinguish between patients with and without heart disease (0 and 1). This can reveal feature relationships concerning heart disease risk.

**Data Standardization**

In the Data Standardization section, data standardization is performed to ensure that all the features (variables) in the dataset have a common scale or distribution. This process is crucial when working with machine learning models that are sensitive to the scale of the input features. Standardization transforms the data so that it has a mean of 0 and a standard deviation of 1.

Standardization ensures that all features have a similar scale, preventing some features from dominating the modeling process due to their larger magnitude. This is particularly important for machine learning algorithms that rely on distance-based calculations or optimization, such as support vector machines (SVM) or k-nearest neighbors (KNN).

**Model Building and Evaluation**

In the Model Building and Evaluation section, various machine learning models are constructed and evaluated to predict the risk of a heart attack. The following models are employed for this purpose:

**Logistic Regression**

Logistic regression is a widely used statistical method for binary classification tasks, making it suitable for predicting whether an individual is at risk of a heart attack (binary outcome: 0 for less chance, 1 for more chance). It models the probability of the target variable (heart attack risk) as a function of the independent variables (features) in the dataset.

**Decision Tree**

A decision tree is a tree-like model that partitions the dataset into subsets based on the values of features, ultimately leading to a decision or prediction. In this case, a decision tree is used to make predictions about heart attack risk based on various patient characteristics.

**Random Forest**

Random Forest is an ensemble learning method that consists of multiple decision trees, each trained on a different subset of the data. The predictions from individual trees are combined to produce a final prediction. This approach tends to improve the model's accuracy and robustness.

**K-Nearest Neighbors (KNN)**

K-Nearest Neighbors is a simple yet effective classification algorithm. It classifies a data point by considering the "k" data points nearest to it in the feature space. In this context, KNN is applied to predict heart attack risk based on the similarity of patients' features.

**Support Vector Machine (SVM)**

A Support Vector Machine is a powerful classification algorithm that finds the optimal hyperplane to separate data into different classes. SVM can be used to create a decision boundary between patients with a higher and lower risk of heart attacks.

The primary goal of this section is to train these machine learning models using the dataset and then evaluate their performance. Evaluation typically involves assessing the accuracy and reliability of the models in making predictions. The model with the highest accuracy is considered the most suitable for predicting heart attack risk.

Each of these models will be evaluated based on specific metrics and criteria to determine their effectiveness in predicting heart attack risk. The model that demonstrates the highest accuracy or best performance will be selected for further use in predicting heart attack risk.

**Model Comparison**

|  |  |  |
| --- | --- | --- |
|  | Model | Accuracy |
| 0 | Logistic Regression | 85.714286 |
| 3 | K Nearest Neighbor | 84.615385 |
| 4 | SVM | 80.219780 |
| 2 | Random Forest | 79.120879 |
| 1 | Decision Tree | 69.230769 |

**Conclusion**

The Heart Attack Risk Predictor project successfully develops models to predict heart attack risk, with the best-performing model achieving an accuracy of 85.7%.