**Prediction of Myocardial Infarction Risk in Cardiovascular Disease Using Machine Learning Algorithms**

**Aim:**

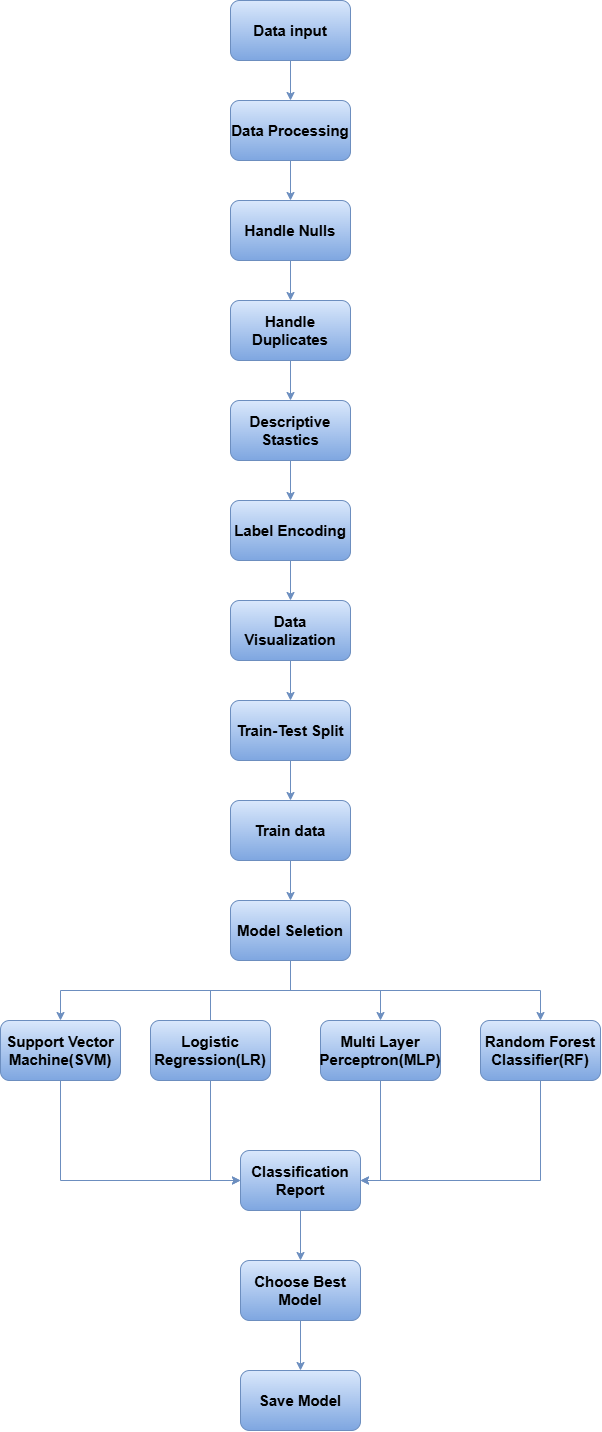
To develop an intelligent and efficient machine learning-based system capable of predicting the risk of myocardial infarction (heart attack) by analyzing patient-specific clinical and lifestyle data, enabling early detection, personalized prevention strategies, and improved healthcare outcomes.

**Dataset Details**:

Collecting the dataset containing features like Age, Marital Status, Sleep Patterns, Depression Levels, Smoking Habits, Diabetes Status, Blood Pressure, Hypersensitivity, Chest Pain (cp), Resting Blood Pressure (trestbps), Cholesterol Levels (chol), Fasting Blood Sugar (fbs), Resting ECG Results (restecg), Maximum Heart Rate Achieved (thalach), Exercise-Induced Angina (exang), Depression (oldpeak), Slope of the Peak Exercise ST Segment (slope), Number of Major Vessels (ca), Thalassemia (thal), and Mortality Status for predicting myocardial infarction.

**Software Tools:**

Visual Studio Code, Python



### 

### 1. Data Input

The dataset contains various features such as Age, Marital Status, Sleep Patterns, Depression Levels, Smoking Habits, and medical parameters like Blood Pressure, Cholesterol Levels, and Heart Rate, aimed at predicting the likelihood of myocardial infarction. The target variable, Mortality Status, indicates whether the patient survived or succumbed to the condition.

### 2. Data Processing

#### **2.1 Handle Nulls**

Identify and handle missing values by either imputing with the mean/median (for numerical data) or mode (for categorical data), or removing rows/columns with excessive nulls.

#### **2.2 Handle Duplicates**

Detect and remove duplicate rows to ensure the dataset is clean and avoid bias in model training.

#### **2.3 Descriptive Statistics**

Generate summary statistics (mean, median, standard deviation, min, max) to understand the distribution of numerical features and detect any outliers or skewed data.

### 3. Label Encoding

Label Encoding is used to convert categorical variables into numerical values. The LabelEncoder from sklearn.preprocessing is applied to the specified columns (Marital\_status, Sleep, Depression, Smoking, Hypersensitivity). Each categorical value in these columns is transformed into a unique integer, making the data suitable for machine learning models. The .astype(int) ensures the encoded values are in integer format.

### 4. Data Visualization

#### 4.1 Countplot for Mortality

Shows the distribution of patients who survived vs. those who did not, based on the "Mortality" variable.

#### **4.2 Histogram for Age**

Visualizes the distribution of ages in the dataset, indicating how frequently different age groups occur.

#### **4.3 Histogram for Cholesterol (Chol)**

Displays the distribution of cholesterol levels, showing how common different cholesterol values are in the dataset.

#### **4.4 Pie Chart for Smoking**

Illustrates the proportion of smokers vs. non-smokers in the dataset, providing insight into smoking habits.

#### **4.5 Countplot for Mortality by Age**

Visualizes the count of survivors vs. non-survivors, broken down by different age groups, highlighting age-related trends in mortality.

#### **4.6 Pairplot**

Shows pairwise relationships and distributions of all numerical features, helping to identify patterns and correlations between variables.

#### **4.7 Heatmap of Correlations**

Displays a heatmap to show correlations between numerical features, helping to identify highly correlated variables.

### 5. Train-Test Split

Splits the dataset into training (70%) and testing (30%) sets, ensuring proportional distribution of the target variable using stratify=Y. The value\_counts() on y\_train shows the distribution of the target in the training set. Prints the number of samples in the training and test sets, along with the total dataset size.

### 6. Model Selection

* **SVM**: Best for binary classification problems, especially when the decision boundary is not easily separable.
* **Logistic Regression**: A simple, interpretable model for binary classification, especially if the relationship between input features and the target is linear.
* **MLP**: A neural network that can model complex relationships in data but requires more data and computational resources.
* **Random Forest**: A robust ensemble model that combines multiple decision trees to improve accuracy and reduce overfitting. It’s especially effective when dealing with complex, noisy data.

### 7. Model Evaluation

#### **7.1 Classification Report**

The classification report provides performance metrics for the model, including precision, recall, F1-score, and support for each class.

#### **7.2 Accuracy**

* The overall accuracy of the model is 99%, indicating that 99% of the predictions were correct.

#### **7.3 Precision**

* For class 0, precision is 0.99, meaning 99% of the predicted positives were actually correct.
* For class 1, precision is 1.00, meaning all predicted positives were correct.

#### **7.4 Recall**

* For class 0, recall is 1.00, meaning all actual positives of class 0 were correctly identified.
* For class 1, recall is 0.96, meaning 96% of the actual positives of class 1 were correctly identified.

#### **7.5 F1-Score**

* The F1-score for class 0 is 0.99 and for class 1 is 0.98, balancing precision and recall for each class.

#### **7.6 Support**

* Support refers to the number of true instances for each class in the test set (87 for class 0 and 24 for class 1).

### 7.7 Confusion Matrix

The confusion matrix visualizes true positives, false positives, true negatives, and false negatives, providing insights into how well the model is distinguishing between classes.

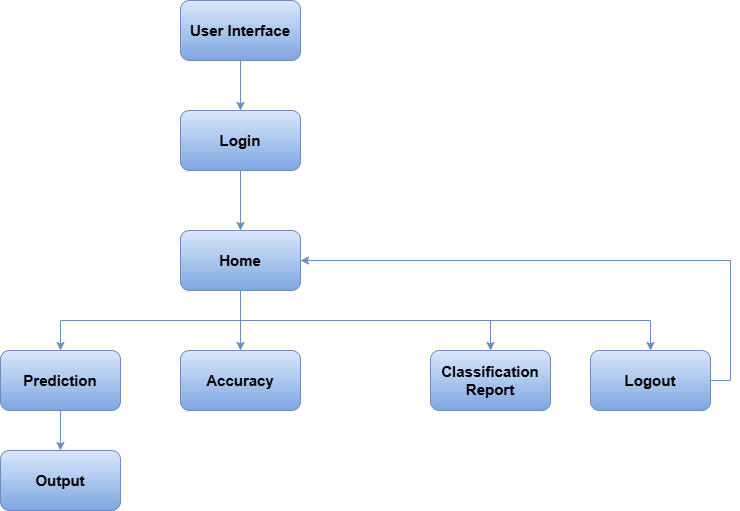
### 7.8 Choose Best Model

* **Comparison**: Based on the classification report, the model that gives the best performance is chosen.
* **Consideration**: If false negatives (missed heart attack cases) are critical, a model prioritizing recall may be selected.

### 8. Save Model

* **Serialization**: Once the best model is chosen, it is saved using joblib or pickle, so it can be loaded and used for future predictions without retraining.
* **Deployment**: The saved model is integrated into a system where new patient data can be input, and the model will output whether the patient is at risk of a heart attack.

**USER INTERFACE:**



1. **User Interface and Login**

The process begins with a user interface where the user enters their credentials to log in. This ensures data security and privacy.

1. **Home Page**

After successful login, the user lands on the home page. This page serves as the central hub for accessing different functionalities.

* 1. **Prediction**

This is the core step where the ML model predicts the MI risk. Here's a breakdown of the process:

The user's relevant medical data (age, blood pressure, cholesterol levels, smoking habits, etc.) is collected.The chosen ML algorithm is **Random Forest**, an ensemble method that combines multiple decision trees to improve accuracy and robustness.The Random Forest model is trained on historical data of patients with known MI outcomes. The model learns to identify patterns and relationships between the input features and the target variable.The trained model is then used to predict the MI risk for the new user based on their input data.The predicted MI risk is presented to the user in a clear and understandable format. This could be:

**Output:**

**Heart stroke will occur**

**Heart stroke will not occur**

1. **Accuracy :**

These sections provide insights into the performance of the ML model:

* **Accuracy**: Measures the overall correctness of the predictions accuracy percentages of each model.

**4**. **Classification Report:**

* **Classification Report**: Shows various metrics like precision, recall, F1-score, and support for different risk categories. This helps assess the model's ability to correctly identify patients at different risk levels.

1. **Logout:**

The user can log out of the system to ensure data security and privacy and moves to home page.

**Reference links:**

**Logistic Regression:** [**https://www.geeksforgeeks.org/understanding-logistic-regression/**](https://www.geeksforgeeks.org/understanding-logistic-regression/)

**SVM:** [**https://www.geeksforgeeks.org/support-vector-machine-algorithm/**](https://www.geeksforgeeks.org/support-vector-machine-algorithm/)

**MLP:** [**https://www.geeksforgeeks.org/classification-using-sklearn-multi-layer-perceptron/**](https://www.geeksforgeeks.org/classification-using-sklearn-multi-layer-perceptron/)

**Random forest:** [**https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/**](https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/)