**Heart Attack Prediction Project Documentation**

**Project Overview**

**Project Title**

Heart Attack Prediction using Machine Learning

**Author**

Aadip Thapaliya,omar,anspreet,muhamed

**Project Description**

This project implements machine learning algorithms to predict the likelihood of heart attacks based on various health parameters and risk factors. The system aims to provide early warning signs and assist healthcare professionals in making informed decisions about patient care.

**Repository Information**

* **Repository URL**: https://github.com/Aadip-Thapaliya/Heart-Attack-Prediction-UE
* **Project Type**: Machine Learning Classification
* **Domain**: Healthcare/Medical Prediction
* **Programming Language**: Python

**Table of Contents**

**1. Project Structure**

**2. Installation and Setup**

**3. Dataset Information**

**4. Features and Attributes**

**5. Machine Learning Models**

**6. Data Preprocessing**

**7. Model Training and Evaluation**

**8. Results and Performance**

**9. Usage Instructions**

**10. Dependencies**

**11. Contributing**

**12. License**

**Installation and Setup**

**Prerequisites**

* Python 3.7 or higher
* pip package manager
* Virtual environment (recommended)

**Installation Steps**

1. **Clone the repository:**
2. git clone https://github.com/Aadip-Thapaliya/Heart-Attack-Prediction-UE.git
3. cd Heart-Attack-Prediction-UE
4. **Create virtual environment:**
5. python -m venv venv
6. source venv/bin/activate # On Windows: venv\Scripts\activate
7. **Install dependencies:**
8. pip install -r requirements.txt
9. **Run the application:**
10. python main.py

**Features and Attributes**

**Input Features**

| **Feature Name** | **Data Type** | **Description** | **Range/Values** |
| --- | --- | --- | --- |
| age | Numeric | Patient age in years | 18-80 |
| sex | Categorical | Patient gender | 0 = Female, 1 = Male |
| cp | Categorical | Chest pain type | 0-3 |
| trestbps | Numeric | Resting blood pressure | 80-200 mmHg |
| chol | Numeric | Serum cholesterol | 100-600 mg/dl |
| fbs | Binary | Fasting blood sugar > 120 mg/dl | 0 = No, 1 = Yes |
| restecg | Categorical | Resting electrocardiogram results | 0-2 |
| thalach | Numeric | Maximum heart rate achieved | 60-220 bpm |
| exang | Binary | Exercise-induced angina | 0 = No, 1 = Yes |
| oldpeak | Numeric | ST depression induced by exercise | 0-7 |
| slope | Categorical | Slope of peak exercise ST segment | 0-2 |
| ca | Numeric | Number of major vessels colored by fluoroscopy | 0-4 |
| thal | Categorical | Thalassemia | 0-3 |

**Target Variable**

* **output**: Binary classification (0 = No heart attack risk, 1 = Heart attack risk)

**Machine Learning Models**

**Implemented Algorithms**

1. **Logistic Regression**
   * Linear classification algorithm
   * Good baseline model for binary classification
   * Provides probability estimates
2. **Random Forest**
   * Ensemble learning method
   * Handles non-linear relationships
   * Feature importance analysis
3. **Support Vector Machine (SVM)**
   * Effective for high-dimensional data
   * Good generalization capability
   * Kernel trick for non-linear separation
4. **Decision Tree**
   * Interpretable model
   * Handles both numerical and categorical features
   * Easy to visualize decision paths

**Model Selection Criteria**

* Cross-validation performance
* Generalization ability
* Interpretability requirements
* Computational efficiency

**Data Preprocessing**

**Data Cleaning Steps**

1. **Missing Value Treatment**
   * Identification of missing values
   * Imputation strategies (mean, median, mode)
   * Removal of records with excessive missing data
2. **Outlier Detection and Treatment**
   * Statistical methods (IQR, Z-score)
   * Visualization techniques (box plots, scatter plots)
   * Outlier removal or transformation
3. **Feature Engineering**
   * Creation of new features
   * Feature combinations
   * Domain-specific transformations
4. **Data Normalization**
   * Scaling numerical features
   * Standardization (Z-score normalization)
   * Min-Max scaling
5. **Categorical Encoding**
   * One-hot encoding for nominal variables
   * Label encoding for ordinal variables
   * Target encoding where appropriate

**Data Splitting Strategy**

* **Training Set**: 80% of the data
* **Test Set**: 20% of the data
* **Stratified sampling** to maintain class distribution

**Model Training and Evaluation**

**Training Process**

1. **Cross-Validation**
   * K-fold cross-validation (k=5)
   * Stratified cross-validation for balanced evaluation
   * Performance consistency across folds
2. **Hyperparameter Tuning**
   * Grid search for optimal parameters
   * Random search for efficiency
   * Bayesian optimization for complex models
3. **Feature Selection**
   * Correlation analysis
   * Recursive feature elimination
   * Feature importance from tree-based models

**Evaluation Metrics**

1. **Classification Metrics**
   * Accuracy: Overall correctness
   * Precision: True positive rate
   * Recall (Sensitivity): Detection rate
   * F1-Score: Harmonic mean of precision and recall
   * Specificity: True negative rate
2. **Advanced Metrics**
   * AUC-ROC: Area under ROC curve
   * Confusion Matrix: Detailed classification results
   * Classification Report: Comprehensive metric summary
3. **Cross-Validation Scores**
   * Mean CV accuracy
   * Standard deviation of CV scores
   * Statistical significance tests

**Dependencies**

**Core Libraries**

pandas>=1.3.0

numpy>=1.21.0

scikit-learn>=1.0.0

matplotlib>=3.4.0

seaborn>=0.11.0

jupyter>=1.0.0

**Optional Libraries**

plotly>=5.0.0

streamlit>=1.0.0

flask>=2.0.0

tensorflow>=2.6.0

pytorch>=1.9.0

**Development Dependencies**

pytest>=6.0.0

black>=21.0.0

flake8>=3.9.0

mypy>=0.910

**Project Timeline**

**Development Phases**

1. **Phase 1**: Data Collection and Exploration
2. **Phase 2**: Data Preprocessing and Feature Engineering
3. **Phase 3**: Model Development and Training
4. **Phase 4**: Model Evaluation and Selection
5. **Phase 5**: System Integration and Testing
6. **Phase 6**: Documentation and Deployment

**Future Enhancements**

**Planned Improvements**

1. **Model Enhancements**:
   * Deep learning models (CNN, RNN)
   * Ensemble methods
   * Automated machine learning (AutoML)
2. **Feature Engineering**:
   * Advanced feature selection techniques
   * Time-series features for longitudinal data
   * External data integration
3. **System Improvements**:
   * Real-time prediction API
   * Model monitoring and retraining
   * Performance optimization
4. **User Interface**:
   * Enhanced web interface
   * Mobile application
   * Integration with healthcare systems

**Research Opportunities**

* Explainable AI for medical decisions
* Federated learning for privacy-preserving training
* Multi-modal data integration
* Personalized risk assessment

**Contributing**

**How to Contribute**

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Add tests for new functionality
5. Submit a pull request

**Contribution Guidelines**

* Follow PEP 8 style guidelines
* Add appropriate documentation
* Include unit tests
* Update README if necessary

**Code Review Process**

* All contributions require code review
* Automated testing must pass
* Documentation must be updated
* Performance impact assessment

**License**

MIT License

Copyright (c) 2025 Aadip Thapaliya

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.**Contact Information**

**Project Maintainer**

* **Name**: Aadip Thapaliya
* **Email**: aadipthapaliya@gmail.com
* **GitHub**: [@Aadip-Thapaliya](https://github.com/Aadip-Thapaliya)