# Heart Disease Prediction - Project Specification

## 1. Project Overview

### 1.1 Problem Statement

Heart disease remains the leading cause of death globally. Early detection and risk assessment can significantly improve patient outcomes. This project develops a machine learning system to predict the likelihood of heart disease based on patient health metrics and lifestyle factors.

### 1.2 Objectives

- Build a binary classification model to predict heart disease (Yes/No)
- Achieve minimum 85% accuracy with good precision and recall balance
- Identify key risk factors contributing to heart disease
- Create an interpretable, deployable model for healthcare applications
- Provide comprehensive documentation for educational purposes

### 1.3 Target Audience

- High school students learning ML fundamentals
- Bachelor's degree students in Computer Science, Data Science, or Health Informatics
- Healthcare professionals interested in AI applications
- Researchers studying cardiovascular disease prediction

## 2. Dataset Description

### 2.1 Source

- **Dataset Name**: Heart Disease Health Indicators
- **Records**: 10,000 patient records
- **Features**: 20 independent variables + 1 target variable
- **Class Distribution**: Imbalanced (80% No, 20% Yes)

### 2.2 Features

#### Demographic Features

- **Age**: Patient age (18-80 years)
- **Gender**: Male/Female

**Clinical Measurements**

- **Blood Pressure**: Systolic blood pressure (120-180 mmHg)
- **Cholesterol Level**: Total cholesterol (mg/dL)
- **BMI**: Body Mass Index
- **Fasting Blood Sugar**: Blood glucose level (mg/dL)
- **Triglyceride Level**: Triglycerides (mg/dL)
- **CRP Level**: C-Reactive Protein (inflammation marker)
- **Homocysteine Level**: Amino acid level

**Binary Health Indicators**

- **High Blood Pressure**: Yes/No
- **Low HDL Cholesterol**: Yes/No (good cholesterol)
- **High LDL Cholesterol**: Yes/No (bad cholesterol)
- **Diabetes**: Yes/No
- **Family Heart Disease**: Yes/No

**Lifestyle Factors**

- **Exercise Habits**: Low/Medium/High
- **Smoking**: Yes/No
- **Alcohol Consumption**: None/Low/Medium/High
- **Stress Level**: Low/Medium/High
- **Sleep Hours**: Daily sleep duration
- **Sugar Consumption**: Low/Medium/High

**Target Variable**

- **Heart Disease Status**: Yes/No (binary classification)

## 3. Project Scope

### 3.1 In Scope

✅ Exploratory Data Analysis (EDA)
✅ Data cleaning and preprocessing
✅ Feature engineering and selection

✅ Multiple ML model training and comparison
✅ Hyperparameter tuning
✅ Model evaluation and interpretation
✅ Model deployment (Flask API + Docker)
✅ Comprehensive documentation

### 3.2 Out of Scope

❌ Deep learning models (focus on classical ML)
❌ Real-time streaming data processing
❌ Integration with Electronic Health Records (EHR)
❌ Mobile application development
❌ Clinical validation and regulatory approval

## 4. Success Criteria

### 4.1 Model Performance Metrics

- **Accuracy**: ≥ 85%
- **Precision**: ≥ 80% (minimize false positives)
- **Recall**: ≥ 75% (minimize false negatives)
- **F1-Score**: ≥ 77%
- **ROC-AUC**: ≥ 0.88

### 4.2 Technical Requirements

- Clean, well-documented Python code
- Reproducible results (random seed management)
- Version control with Git
- Modular, maintainable code structure
- Comprehensive unit tests

### 4.3 Educational Outcomes

- Clear step-by-step explanations
- Visualizations for key insights
- Practical ML concepts demonstration
- Best practices for production ML systems

## 5. Technology Stack

### 5.1 Programming Language

- **Python 3.8+**

### 5.2 Core Libraries

- **Data Manipulation**: pandas, numpy
- **Visualization**: matplotlib, seaborn, plotly
- **Machine Learning**: scikit-learn, xgboost, lightgbm
- **Model Interpretation**: SHAP, lime
- **Deployment**: Flask, FastAPI, Docker

### 5.3 Development Tools

- **Version Control**: Git, GitHub
- **Environment Management**: virtualenv, conda
- **Notebooks**: Jupyter Lab
- **Testing**: pytest
- **Code Quality**: pylint, black

## 6. Project Deliverables

### 6.1 Code Repository

1. Source code for data processing
2. Model training scripts
3. Evaluation and visualization scripts
4. Deployment code (API + Docker)
5. Unit tests

### 6.2 Documentation

1. README.md with quick start guide
2. Project specification (this document)
3. Design document
4. Data dictionary

5. Model performance report
6. API documentation

### 6.3 Models

1. Trained model artifacts (.pkl files)
2. Model performance metrics
3. Feature importance analysis
4. Confusion matrices and ROC curves

## 7. Project Timeline

**Phase 1: Data Understanding (Week 1)**

- Load and explore dataset
- Perform statistical analysis
- Create visualizations
- Document findings

**Phase 2: Data Preparation (Week 1-2)**

- Handle missing values
- Encode categorical variables
- Feature scaling
- Address class imbalance
- Train-test split

**Phase 3: Model Development (Week 2-3)**

- Train baseline models
- Feature selection
- Hyperparameter tuning
- Cross-validation
- Model comparison

**Phase 4: Evaluation & Interpretation (Week 3)**

- Evaluate on test set
- Feature importance analysis

- Error analysis
- Model interpretation with SHAP

**Phase 5: Deployment (Week 4)**

- Create Flask API
- Dockerize application
- Write deployment documentation
- Create demo examples

## 8. Risks and Mitigation

### 8.1 Data Quality Issues

- **Risk**: Missing values, outliers, errors
- **Mitigation**: Robust data cleaning pipeline, validation checks

### 8.2 Class Imbalance

- **Risk**: Model bias toward majority class
- **Mitigation**: SMOTE, class weights, stratified sampling

### 8.3 Overfitting

- **Risk**: Poor generalization to new data
- **Mitigation**: Cross-validation, regularization, ensemble methods

### 8.4 Interpretability

- **Risk**: Black-box model decisions
- **Mitigation**: Feature importance, SHAP values, simple model comparison

## 9. Ethical Considerations

### 9.1 Privacy

- No personally identifiable information (PII)
- Anonymized dataset
- Secure data handling

### 9.2 Fairness

- Monitor for gender bias
- Ensure equitable performance across demographics
- Report fairness metrics

### 9.3 Medical Disclaimer

⚠️ **This model is for educational purposes only and should NOT be used for actual medical diagnosis. Always consult qualified healthcare professionals for medical advice.**

## 10. References and Resources

### 10.1 Academic Resources

- American Heart Association Guidelines
- WHO Cardiovascular Disease Statistics
- Scikit-learn Documentation
- Machine Learning Mastery

### 10.2 Learning Materials

- Python for Data Science Handbook
- Hands-On Machine Learning with Scikit-Learn
- Feature Engineering and Selection Book
- Interpretable Machine Learning Book

---

**Document Version**: 1.0
**Last Updated**: January 2026
**Author**: ML Education Team