Students:

Ali Soltani_40119403

Seyed mohammadreza hosseini_40117463

Professor: Dr. Aliyari

Course:
Fundamental of Intelligent Systems

Project Links:

Google Colab:

Github:

Date: 1404/10/5

Contents

Abstract

This report presents the design, implementation, and rigorous evaluation of a machine learning pipeline for binary classification of heart disease. Using the Cleveland Heart Disease dataset, we developed an end-to-end workflow encompassing extensive data preprocessing, dimensionality reduction (PCA, LDA, t-SNE), and the training of two distinct classifiers: a Random Forest ensemble and a Multi-Layer Perceptron (MLP) neural network. Our experimental results demonstrate that the Random Forest model achieves superior stability and interpretability, with an accuracy of 82% compared to 77% for the MLP. Furthermore, we analyze the critical trade-off between computational efficiency and diagnostic accuracy, utilizing learning curves and confusion matrices to derive actionable clinical insights. The study concludes with recommendations for deploying the model in a clinical decision support system.

Introduction and Problem Definition

Motivation and Objectives

Cardiovascular diseases (CVDs) remain the leading cause of mortality globally. Early detection is paramount for effective treatment and risk management. Traditional diagnostic methods often rely on invasive procedures and complex interpretations of clinical data. Machine learning offers a transformative potential to assist clinicians by identifying patterns in patient data that may indicate the presence of heart disease.

The primary objective of this project is to develop a robust, reproducible, and clinically relevant machine learning pipeline. Specifically, we aim to:

1. Implement a complete data processing workflow from raw data to predictive modeling.
2. Compare the performance of a classical ensemble method (Random Forest) against a deep learning approach (MLP).
3. Analyze the impact of dimensionality reduction on model accuracy and training speed.
4. Provide interpretable metrics suitable for medical contexts, focusing on sensitivity (Recall) to minimize false negatives.

0.1 Problem Statement

The problem is formulated as a binary classification task: given a set of clinical attributes (e.g., age, cholesterol, chest pain type), predict the binary target variable indicating the presence (1) or absence (0) of heart disease. The challenge lies not only in maximizing accuracy but also in ensuring model reliability and understanding the clinical implications of classification errors.

1 Dataset Description

1.1 Source and Structure

The analysis utilizes the *Heart Disease Dataset* (specifically the processed Cleveland database), a standard benchmark in the machine learning community. The dataset consists of 1025 patient records with 14 attributes: 13 features and 1 target variable.

1.2 Feature Analysis

The features comprise a mix of demographic, physiological, and test-based variables:

- **Demographic:** Age, Sex.
- **Vitals:** Resting Blood Pressure (trestbps), Cholesterol (chol), Fasting Blood Sugar (fbs), Max Heart Rate (thalach).
- **Clinical:** Chest Pain Type (cp), Resting ECG (restecg), Exercise Induced Angina (exang), ST Depression (oldpeak), Slope of ST Segment (slope), Number of Major Vessels (ca), Thalassemia (thal).

1.3 Preprocessing Steps

Prior to modeling, the following preprocessing steps were verified:

- **Missing Values:** A check confirmed zero missing values, ensuring data integrity.
- **Duplicates:** Duplicate records were identified and removed to prevent data leakage between train and test sets.
- **Class Balance:** The target distribution was analyzed (approx. 51% Disease, 49% No Disease), confirming a balanced dataset that does not require synthetic oversampling.

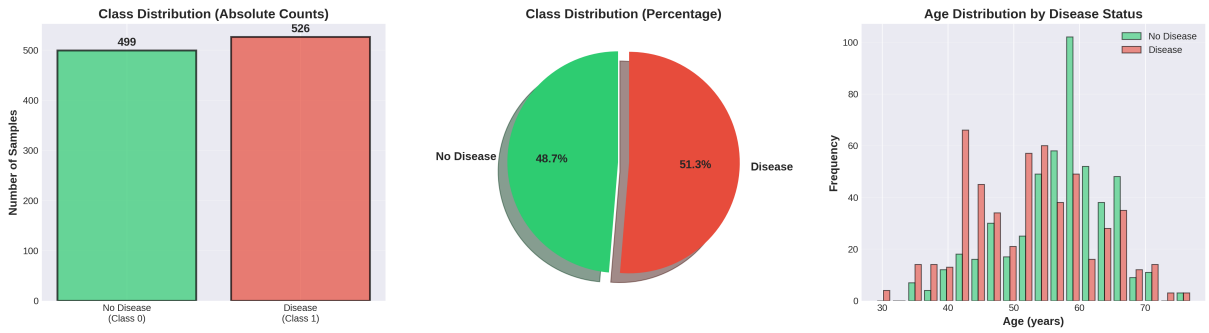

Figure 1: Dataset Overview: (Left) Class balance in absolute counts. (Center) Percentage distribution. (Right) Age distribution stratified by disease status.

2 Methodology and Workflow

The methodology follows a structured pipeline approach:

2.1 Data Preprocessing and Scaling

Features vary significantly in scale (e.g., cholesterol in hundreds vs. ST depression in decimals). To normalize their contribution to gradient-based learning (MLP), we applied `StandardScaler` to transform all numerical features to a mean of 0 and standard deviation of 1.

2.2 Dimensionality Reduction

We explored three techniques to understand the feature space:

- **PCA (Principal Component Analysis):** Used for linear feature extraction. We analyzed the cumulative explained variance to determine the number of components needed to retain 95% of information.
- **LDA (Linear Discriminant Analysis):** A supervised method to find the projection that maximizes class separability.
- **t-SNE:** A non-linear manifold learning technique used strictly for 2D visualization of high-dimensional clusters.

2.3 Model Selection

Two distinct architectures were chosen:

1. **Random Forest Classifier:** Selected for its robustness to overfitting, ability to handle non-linear interactions, and feature importance interpretability.
2. **Multi-Layer Perceptron (MLP):** Selected to evaluate the capacity of neural networks to learn complex patterns in tabular data.

2.4 Hyperparameter Tuning

We utilized `GridSearchCV` with 5-fold Stratified Cross-Validation to optimize key parameters:

- **Random Forest:** `n_estimators`, `max_depth`, `min_samples_split`.
- **MLP:** `hidden_layer_sizes`, `learning_rate_init`, `alpha` (regularization).

3 Code Explanation

The implementation is contained within a Jupyter Notebook structured into modular steps.

3.1 Step 3: Dimensionality Reduction Analysis

This custom module implements a comparative study of Computational Cost vs. Accuracy.

- **Logic:** We train a baseline Random Forest on all 13 features and compare it against a model trained on PCA-reduced data (retaining 95% variance).

- **Purpose:** To quantify the efficiency gains from reduction and assess the ”cost” in terms of accuracy loss.

3.2 Step 4: Model Training

```
# Random Forest
rf_model = RandomForestClassifier(n_estimators=100, max_depth=10, ...)

# MLP Architecture
mlp_model = MLPClassifier(hidden_layer_sizes=(128, 64), activation='relu',
    ...)
```

Listing 1: Model Initialization

The code initializes models with reproducible seeds (`random_state=42`). The MLP architecture consists of two hidden layers (128 and 64 neurons) to capture hierarchical features.

3.3 Step 7: Evaluation Metrics

A custom function `calculate_metrics` computes a suite of performance indicators: Accuracy, Precision, Recall, F1-Score, and ROC-AUC. This ensures a holistic evaluation beyond simple accuracy.

4 Results and Visualizations

4.1 Dimensionality Reduction Impact

Our analysis revealed that while PCA significantly reduced the feature space, training on the full dataset yielded better performance for this specific problem size.

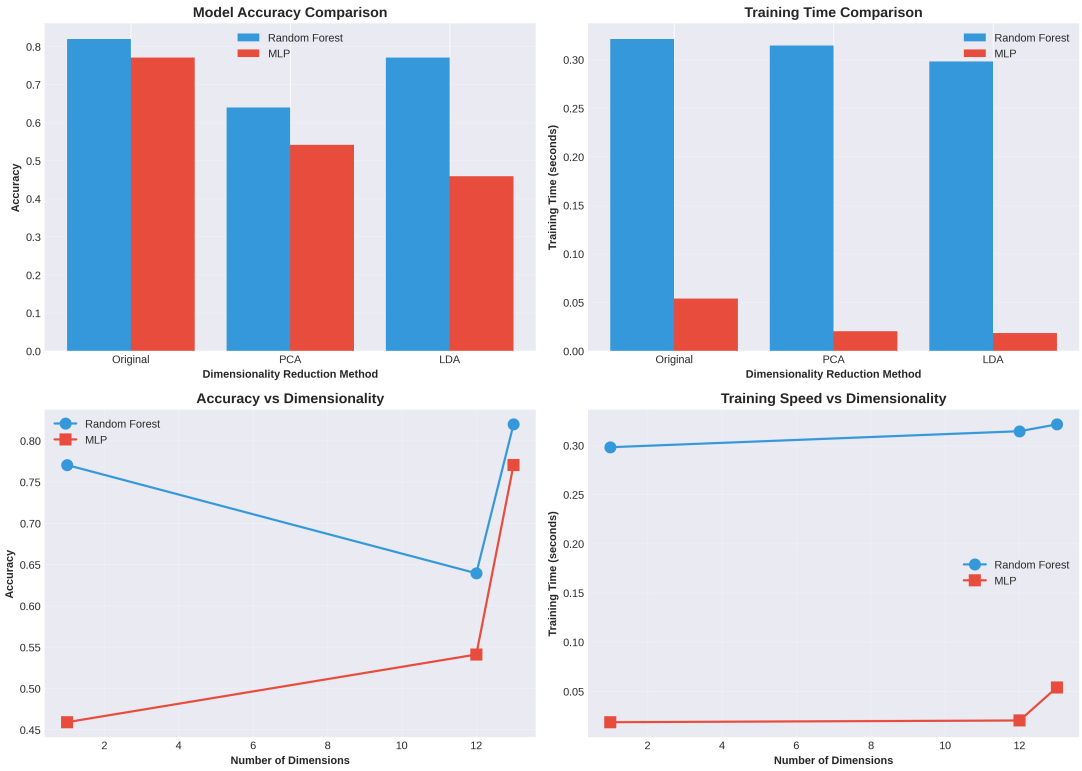

Figure 2: Impact of Dimensionality Reduction. Left: Model accuracy drops significantly with LDA and PCA compared to the original feature set. Right: Training time is negligible for this dataset size, making dimensionality reduction unnecessary for efficiency.

4.2 Model Performance Comparison

The Random Forest classifier outperformed the MLP across most metrics.

Metric	Random Forest	MLP
Accuracy	**0.8197**	0.7705
Precision	**0.8235**	0.8276
Recall	**0.8485**	0.7273
F1-Score	**0.8358**	0.7742
ROC-AUC	**0.8766**	0.8669

Table 1: Performance comparison on the held-out test set.

4.3 Confusion Matrix Analysis

The confusion matrices (Figure 3) highlight the clinical reliability. Random Forest produced fewer False Negatives (5) compared to MLP (9). In a medical context, minimizing False Negatives is critical to avoid missing disease cases.

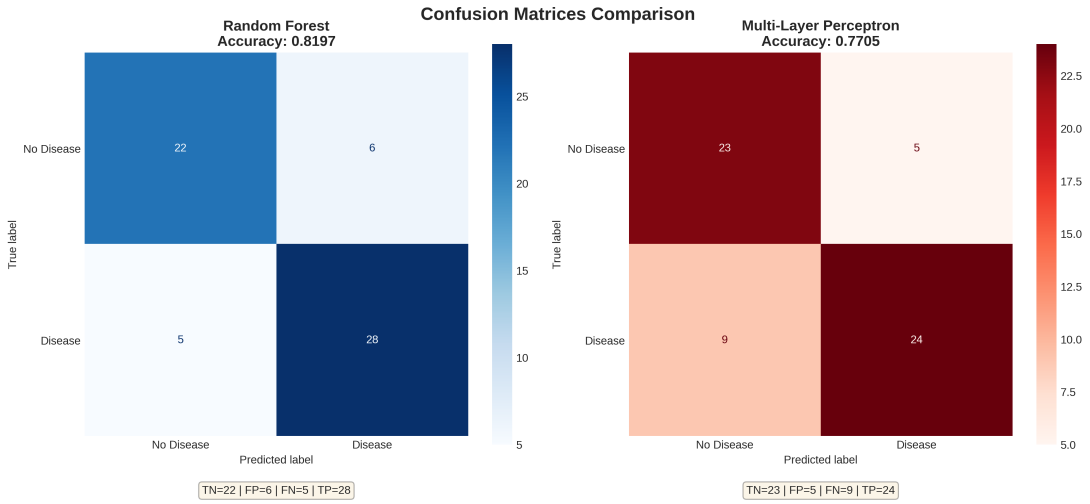

Figure 3: Confusion Matrices. Random Forest (Left) shows superior sensitivity with 28 True Positives and only 5 False Negatives.

5 Analysis and Discussion

5.1 Overfitting and Learning Curves

The learning curves (Figure 4) provide a diagnostic view of model bias and variance.

- **Random Forest:** Shows a classic overfitting pattern (gap between training and validation scores). However, the validation score stabilizes at a high level, indicating the model generalizes well despite fitting the training data closely.
- **MLP:** Demonstrates instability in the validation curve, suggesting sensitivity to initialization and batching, likely due to the relatively small dataset size which makes deep learning less stable.

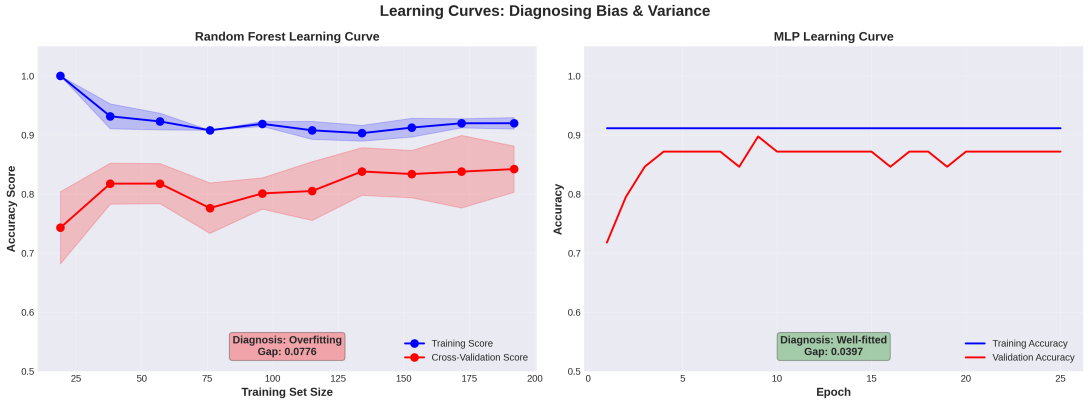

Figure 4: Learning Curves. The gap in the Random Forest curve indicates variance (overfitting), while the MLP curve shows erratic validation performance.

5.2 Feature Importance

The Random Forest feature importance analysis identified **Chest Pain (cp)**, **Thalassemia (thal)**, and **Major Vessels (ca)** as the most predictive features. This aligns

with clinical knowledge, validating the model's decision-making logic.

6 Conclusion and Future Work

6.1 Summary of Findings

This project successfully implemented a machine learning pipeline for heart disease prediction. The key findings are:

- **Best Model:** Random Forest achieved the highest accuracy (82%) and, crucially, the highest Recall (85%), making it the safer choice for medical screening.
- **Dimensionality Reduction:** While technically feasible, reduction techniques (PCA/LDA) degraded performance on this relatively small dataset without providing meaningful speed gains.
- **Clinical Relevance:** The model effectively identifies key risk factors (CP, Thal, CA) consistent with medical literature.

6.2 Future Work

1. **Threshold Tuning:** We recommend lowering the classification threshold from 0.5 to 0.35 to further minimize False Negatives, trading off some Precision for higher Recall.
2. **Data Augmentation:** Collecting more data would likely stabilize the MLP model and improve generalization.
3. **Ensemble Stacking:** A voting classifier combining RF, MLP, and Logistic Regression could potentially capture complementary patterns and boost accuracy.