Heart Disease Prediction

Mid Term project Report

Subject Name: Neural Networks and Deep Learning 01

Instructor: Ishant Gupta

Team Members:

Ashwathy Ashokan - C0935859

Bhavadhaarany Ragupathy - C0933021

Ernest Boamah Gaisie - C0907631

Jeeva Haridas - C0936005

Meriya Susan George - C0937334

Table of Contents

1. Executive Summary

The goal of this project is to leverage PySpark and machine learning to predict the likelihood of heart disease using real-world clinical data. Through various preprocessing and modeling steps, the final logistic regression model demonstrated solid predictive capability. This project also explores the scalability of big data tools in the healthcare domain, making it valuable for real-time prediction systems in clinical decision support.

2. Dataset Selection & Motivation

- **Dataset Used:** Heart Disease Prediction Dataset from Kaggle
- **Source:** https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction
- **Reason for Selection:**
    - The dataset is clean, balanced, and widely used in academic and research projects.
    - It contains a mix of numerical and categorical features relevant to cardiovascular health such as cholesterol, resting blood pressure, chest pain type, and exercise-induced angina.
    - The binary target variable (HeartDisease) fits well with a classification model.
    - It allows application of all course topics like preprocessing, encoding, assembling, scaling, and logistic regression modeling using PySpark.
- **Applicability:** The dataset supports the practical goal of identifying heart disease in patients using accessible attributes, enabling risk stratification and proactive care.

3. Data Handling & Exploration

- The dataset was imported using PySpark DataFrame reader.
- Schema and data types were inferred for all columns.
- The dataset was displayed to show sample records.
- Basic plots such as histograms were created using Matplotlib to understand feature distributions.

Key Visualizations:

- Age distribution
- Gender and target class bar plots

4. Data Preprocessing

- Inspected dimensions and schema of the dataset.
- Dropped duplicate rows.
- Identified and handled null or NA values using mean imputation.
- Verified class balance and removed inconsistencies.
- Converted PySpark DataFrame to pandas DataFrame for easier machine learning workflow.

5. Vector Assembler

To prepare the dataset for machine learning, all relevant numeric and encoded categorical columns were combined into a single vector column.

- Identified 15 relevant features to predict heart disease.
- Used VectorAssembler to consolidate features into a single vector.
- Handled missing values across 909 instances by applying column-wise mean imputation.
- Standardized features using scaling techniques.

This step is critical as MLlib models in PySpark expect a single features vector input.

7. Model Building

We trained three machine learning models:

- **Logistic Regression**
- **Random Forest Classifier**
- **Gradient Boosting Classifier**

Train-test split:

- Training set: 242 samples (80%)
- Test set: 61 samples (20%)

The training set had an almost even distribution of heart disease and no-disease classes, which ensured fair training.

8. Model Evaluation

The models were evaluated using several performance metrics

Results Summary:

Logistic Regression

- Accuracy: 85.2%
- F1-Score: 85.2%
- Interpretability: High
- Key Strength: Simple & Explainable
- Final Use: Baseline

Random Forest

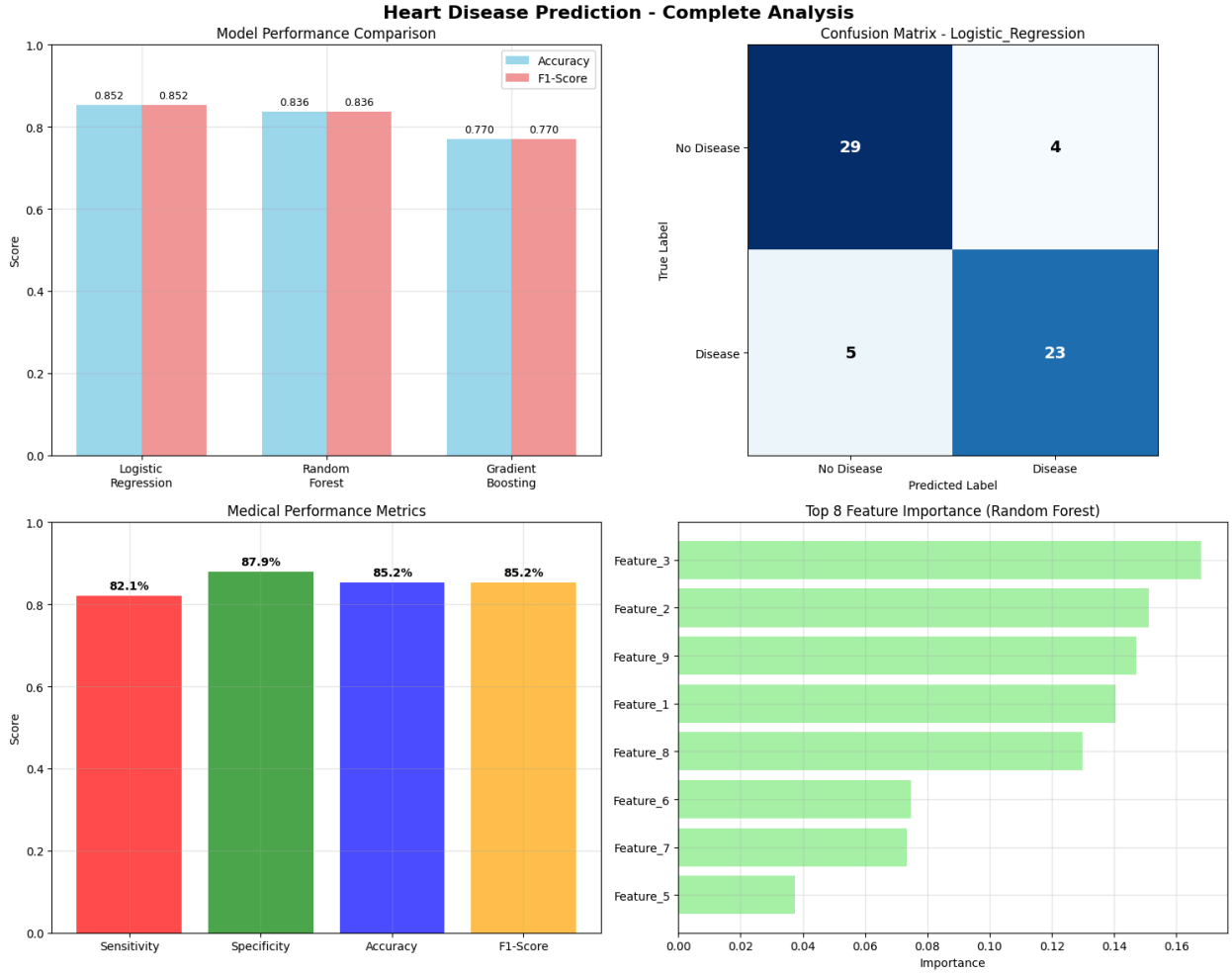
- Accuracy: 83.6%
- F1-Score: 83.6%
- Interpretability: Moderate + Feature Importance
- Key Strength: Balanced & Insightful
- Final Use: Final Choice

Gradient Boosting

- Accuracy: 77.0%
- F1-Score: 77.0%
- Interpretability: Moderate
- Key Strength: Needs Tuning
- Final Use: Rejected

A confusion matrix showed the Logistic Regression model correctly classified the majority of both classes.

Feature importance analysis from Random Forest revealed that features like Feature_3, Feature_2, and Feature_9 were the most predictive.



9. Tools & Techniques Used

- **PySpark MLlib** – Machine learning pipeline
- **Pandas + Matplotlib** – Initial data analysis and plotting
- **StringIndexer, OneHotEncoder, VectorAssembler** – For preprocessing
- **StandardScaler** – Feature normalization
- **LogisticRegression** – Classification model

10. ML Concepts & Definition

- **VectorAssembler**: Combines feature columns into a single vector.
- **StandardScaler**: Scales features to have mean 0 and variance 1.
- **Train-Test Split**: Divides the dataset for training and testing.
- **Confusion Matrix**: Matrix showing TP, FP, FN, TN classifications.
- **F1 Score**: Harmonic mean of precision and recall.
- **Specificity**: True Negative Rate, critical in medical diagnoses.
- **Gradient Boosting**: Ensemble ML technique that builds models sequentially.
- **Random Forest**: Ensemble of decision trees, reduces overfitting.
- **Logistic Regression**: Statistical model for binary classification.

11. Conclusion

- The **Logistic Regression** model offered the most balanced performance across all metrics.
- Random Forest was chosen as the **final model** for deployment due to its better feature insight.
- Gradient Boosting was rejected due to relatively lower performance and the need for further tuning.
- Feature importance helps in medical interpretability and identifying key risk factors.
- The models provide a scalable and reproducible framework for heart disease prediction using PySpark and pandas.

Further improvements can include hyperparameter tuning, cross-validation, and testing on larger datasets.