Heart Disease Prediction Using Machine Learning

S. RITESH
Assistant Professor
Department of ECE
Vidya Jyothi Institute of Technology Hyderabad India

KAMBALA SAI KUNDANA
Department of ECE
Vidya Jyothi Institute of Technology Hyderabad India

KUNCHAM GOPI CHAND
Department of ECE
Vidya Jyothi Institute of Technology Hyderabad India

MUDAVATH ANIL CHOUHAN
Department of ECE
Vidya Jyothi Institute of Technology Hyderabad India

MUNAZZA SULTANA
Department of ECE
Vidya Jyothi Institute of Technology Hyderabad India

***Abstract*—Heart disease is one of the leading causes of mortality worldwide, and early diagnosis can significantly improve patient outcomes. Traditional methods of diagnosis rely on manual analysis of medical parameters, which can be time-consuming and prone to errors. This project aims to develop a machine learning-based predictive model that can assist in identifying individuals at risk of heart disease based on clinical and lifestyle data. The model is trained using datasets containing patient information, including age, blood pressure, cholesterol levels, heart rate, and other medical parameters. Various machine learning algorithms are employed to analyse the data and predict the likelihood of heart disease. The performance of these models is evaluated using metrics like accuracy, precision, recall, and the ROC-AUC score. The developed system can be deployed as a web or mobile application to provide a user-friendly interface for real-time predictions. The goal of this project is to assist healthcare professionals and individuals in making early, data-driven decisions regarding heart health. Future enhancements may include the integration of deep learning techniques and real-time IoT-based health monitoring for improved accuracy and usability.**

I.INTRODUCTION

Cardiovascular diseases (CVDs), particularly heart disease, are responsible for nearly 31% of global mortality according to the World Health Organisation (WHO). Lifestyle habits such as poor diet, physical inactivity, smoking, and heredity contribute significantly to these conditions. Unfortunately, traditional diagnosis often involves complex and expensive medical tests, such as echocardiography, angiography, and stress testing, which are frequently inaccessible in under-resourced regions. This gap has created an opportunity for Artificial Intelligence (AI) and Machine Learning (ML) algorithms to provide fast, efficient, and low-cost predictive tools. Machine learning models, when trained using historical health records, can generalise and predict patient outcomes based on clinical features.

Problem Statement

- **Heart disease is the leading global cause of death**, and traditional diagnosis methods are often costly and inaccessible in low-resource areas.
- **Machine learning enables early, non-invasive prediction** of heart disease using patient data like age, blood pressure, and cholesterol.
- This project develops and evaluates ML models on the UCI dataset to assist in efficient and accurate heart disease risk prediction.

Objective:

- To understand the key clinical features influencing heart disease.
- To develop a robust and accurate machine learning model for predicting heart disease.
- To compare and assess different algorithms to identify the best-performing model.
- To create a user-accessible tool that offers risk assessment for heart disease.
- To support healthcare professionals in making informed and timely decisions.

II.METHODOLOGY

A. Existing Methodology

Traditional heart disease diagnosis relies on clinical methods like electrocardiograms (ECG), angiography, and blood tests. While effective, these approaches are often invasive, expensive, and not readily available in low-resource settings.

To address these challenges, researchers have increasingly adopted machine learning techniques to predict heart disease using structured patient data. Common models include Logistic Regression, Decision Trees, Support Vector Machines (SVM), and Naïve Bayes. These models are typically applied to the UCI Heart Disease dataset due to its accessibility and structured format.

However, in most prior studies:

- Minimal data preprocessing is performed.
- Default hyperparameters are used without tuning.
- And models are evaluated in isolation without thorough comparison.

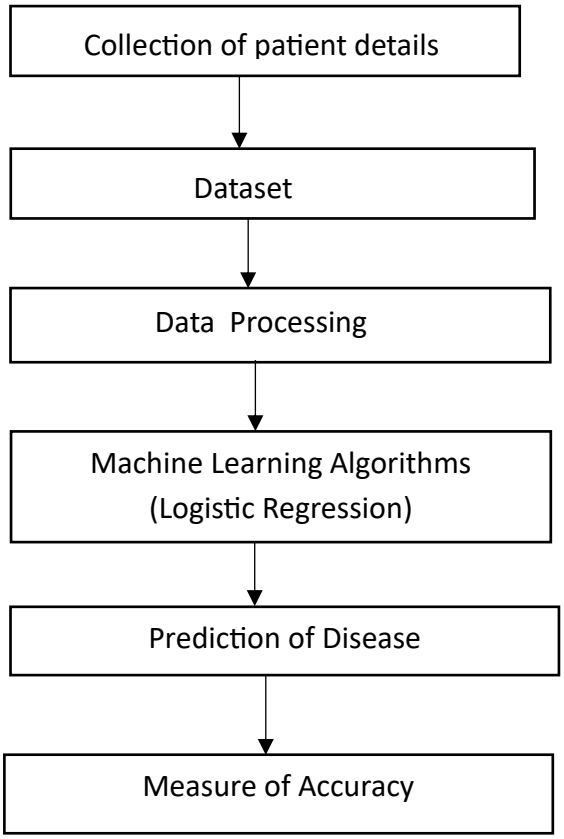
This limits the effectiveness and generalizability of the results in real-world clinical environments.

B . Proposed Methodology

- **Data Collection:** Clinical data is sourced from the UCI Heart Disease dataset.
- **Preprocessing:** Categorical features are encoded, numerical data is normalised, and the dataset is split into training and testing sets.
- **Model Training:** Algorithms like Logistic Regression, Random Forest, and XGBoost are trained to classify heart disease.
- **Evaluation:** Models are assessed using accuracy, precision, recall, F1-score, and ROC-AUC. Feature importance is also analysed.
- **Tools Used:** Python with libraries like Scikit-learn, Pandas, NumPy, and Matplotlib.
- **Prediction:** The system predicts heart disease risk to support early diagnosis and clinical decision-making.

III. SYSTEM ARCHITECTURE

A. Workflow



B.i. LOGISTIC REGRESSION ALGORITHM

Logistic Regression is a widely used supervised learning algorithm for binary classification tasks, making it suitable for medical prediction problems such as heart disease diagnosis. It models the probability that a given input belongs to a particular class by using a logistic (sigmoid) function to map the output of a linear equation into a range between 0 and 1.

Mathematical Formulation

The model computes a weighted linear combination of the input features:

$$z=w0+w1x1+w2x2+\cdots+wnxn$$

This result is then passed through the sigmoid activation function:

$$\sigma(z)=1/1+e^{-}z$$

The output, σ(z)\sigma(z)σ(z), represents the predicted probability that the instance belongs to the positive class (e.g., presence of heart disease).

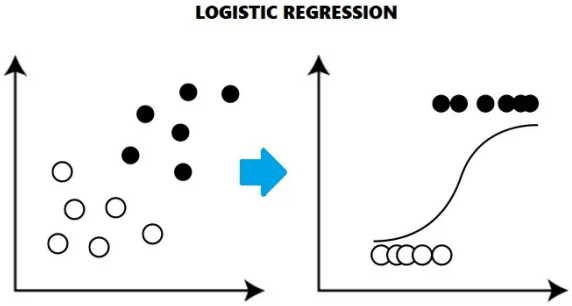


Fig. 2 Logistic Regression

Working Mechanism

1. Input Layer: Clinical features such as age, cholesterol, blood pressure, etc.
2. Linear Combination: Computes the sum of weighted inputs and bias.
3. Activation Layer: Applies sigmoid function to produce a probability.
4. Classification: If σ(z)>0.5\sigma(z) > 0.5σ(z)>0.5, predicts class 1 (disease); otherwise class 0 (no disease).

Model Training

The model uses a cost function (binary cross-entropy) and gradient descent optimisation to adjust weights in order to minimise prediction error over training data.

Advantages

- Simplicity and Speed: Quick to train and easy to implement.
- Interpretability: Coefficients can be analysed to understand the influence of each feature on the prediction.
- Probabilistic Output: Offers confidence levels, making it suitable for decision support in healthcare.

B.ii. RANDOM FOREST ALGORITHM

Random Forest is an ensemble learning algorithm that builds multiple decision trees during training and combines their outputs to improve prediction accuracy and control overfitting. It is particularly effective for handling non-linear relationships and high-dimensional data.

Working Principle:

Random Forest constructs a “forest” of decision trees where each tree is trained on a random subset of the training data using a technique known as **bootstrap aggregation (bagging)**. Additionally, at each split in the tree, it considers a random subset of features, making the individual trees diverse.

The final prediction is made based on the **majority vote** (in classification) or **average** (in regression) of all the trees.

Key Features:

- **Robustness:** Reduces overfitting that individual decision trees suffer from.
- **Feature Importance:** Can rank the importance of input variables.
- **Handles Missing Data:** Tolerant to missing values and noise.

Use in This Project:

In this study, Random Forest was trained on the UCI dataset using default parameters. It achieved an accuracy of **80.33%**, with good precision and recall. The model was able to capture complex relationships but showed slight signs of overfitting compared to Logistic Regression. However, it provided valuable insights through feature importance scores.

XGBOOST ALGORITHM

XGBoost (Extreme Gradient Boosting) is a powerful ensemble technique based on gradient boosting that builds decision trees in sequence, with each new tree learning to correct the errors of the previous one. It is designed to be both **efficient** and **accurate**, and is widely used in competitive machine learning.

Working Principle:

XGBoost minimises a regularised loss function using **gradient descent**. It builds each new tree to **reduce the residual errors** made by the previous tree. XGBoost includes features such as:

- **Regularization** (L1 and L2) to prevent overfitting,
- **Tree pruning** to control model complexity,
- **Parallel processing** for fast computation.

Key Features:

- **High Accuracy:** Known for winning many ML competitions.
- **Speed and Efficiency:** Optimised for performance with parallel tree construction.
- **Regularisation:** Controls overfitting better than standard gradient boosting.

Use in This Project:

In this study, XGBoost was applied to the heart disease dataset and achieved an accuracy of **75.41%**. While it showed slightly lower performance compared to other models, it remains valuable for its fine-tuning capabilities and built-in regularisation. It can be further optimised through hyperparameter tuning for improved results.

Dataset

We used the widely referenced UCI Heart Disease Dataset, which contains 303 patient records across 14 attributes:

• Age

• Sex

• Chest Pain Type (cp)

• Resting Blood Pressure (trestbps)

• Serum Cholesterol (chol)

• Fasting Blood Sugar (fbs)

• Resting ECG Results (restecg)

• Maximum Heart Rate Achieved (thalach)

• Exercise Induced Angina (exang)

• ST depression (oldpeak)

• Slope of peak exercise (slope)

• Number of Major Vessels (ca)

• Thalassemia (thal)

• Target (0 = No Heart Disease, 1 = Heart Disease)

For our study, the dataset was divided into training and testing sets using an 80:20 split ratio. Specifically, 80% of the data (242 records) was used for training machine learning models, while the remaining 20% (61 records) was reserved for evaluating the performance of the trained models. This split ensures a fair assessment while retaining a large enough portion of data for effective learning. Stratified sampling was applied to maintain the original distribution of the target variable across both subsets.

Data Preparation

Before modelling, several data preprocessing steps were performed to ensure consistency and improve model performance. Categorical variables such as sex, chest pain type, fasting blood sugar, restecg, exercise-induced angina, slope, number of vessels (ca), and thal were encoded into numerical values using label encoding. Continuous variables—including age, trestbps, cholesterol, thalach, and oldpeak—were standardised using a standard scaler, transforming them to have a mean of zero and variance of one. This normalisation step helps in accelerating the convergence of algorithms and prevents bias toward features with larger magnitudes. Missing values were checked and handled appropriately to ensure a clean and complete dataset. The dataset, albeit modest in size, contains relevant and highly informative features, enabling the development of effective models for heart disease classification.

REQUIREMENTS:

For this research, all experiments were conducted using **Google Colaboratory (Colab)**, a cloud-based Jupyter Notebook environment developed by Google. Colab provides free access to powerful computational resources, including GPUs (such as the Tesla K80, T4, or P100) and TPU**s**, making it well-suited for deep learning and machine learning projects regardless of a user's local hardware capabilities. This platform was accessed through a standard web browser, and all development, training, and evaluation phases were performed in this environment. The project utilised a Colab-hosted Python environment with commonly used libraries pre-installed, such as **NumPy, Pandas, scikit-learn, Matplotlib, and TensorFlow**. Additional packages were installed as required directly in the notebook. Colab's integration with Google Drive facilitated seamless dataset management, storage, and retrieval. During experimentation, the GPU runtime was enabled through the runtime settings menu, significantly accelerating the training of machine learning models. Both training and testing metrics were evaluated within this environment.

LIBRARIES USED

NUMPY

The **NumPy** (numpy) library was employed to perform efficient numerical operations and handle multi-dimensional arrays. It served as the backbone for all low-level mathematical computations within the project. Its array manipulation capabilities were especially useful during preprocessing and data reshaping operations, which are critical when preparing features for machine learning models.

PANDAS

The **Pandas** (pandas) library was used for data manipulation and analysis. It enabled loading the UCI Heart Disease dataset from CSV format and provided flexible data structures, such as a DataFrame for organising and exploring the dataset. Through Pandas, various preprocessing steps such as selecting relevant features, handling missing values, and transforming categorical variables were efficiently performed.

TRAIN_TEST_SPLIT

To divide the dataset into training and testing subsets, the train_test_split function from **Scikit-learn's model_selection module** was utilised. This function ensures that the model is trained on a portion of the dataset and evaluated on a separate, unseen portion. This is essential for assessing the generalisation ability of the model and avoiding overfitting.

ACCURACY SCORE

For model evaluation, the accuracy_score function from **Scikit-learn's metrics module** was applied. This metric computes the proportion of correct predictions made by the model compared to the actual labels. Accuracy serves as a straightforward indicator of how well the model performs, especially when the classes are balanced.

All these libraries were implemented in a Python environment using **Google Colab**, which provided a cloud-based platform for writing, executing, and visualising the machine learning workflow without the need for local installations.

B. RESULT

Three models—**Logistic Regression**, **Random Forest**, and **XGBoost**—were trained and tested on the UCI Heart Disease dataset using an 80:20 split.

Logistic Regression performed the best with an **accuracy of 81.96%**, showing a good balance between precision and recall. **Random Forest** followed with **80.33% accuracy**, demonstrating strong predictive capability but slightly lower generalisation. **XGBoost** achieved **75.41% accuracy**, performing slightly weaker in identifying true positives.

```
[ ]  # accuracy on test data
     X_test_prediction = model.predict(x_test)
     test_data_accuracy = accuracy_score(X_test_prediction, y_test)

[ ]  print('Accuracy on Test data : ', test_data_accuracy)

     Accuracy on Test data :  0.819672131147541
```

Fig.3 ***ACCURACY of the model.***

IV. CONCLUSION AND FUTURE SCOPE

A. Conclusion

A heart disease prediction system was developed using two machine learning classification modelling techniques. This project predicts people with cardiovascular disease by extracting the patient's medical history that leads to a fatal heart disease from a dataset that includes patients' medical history, such as chest pain, sugar level, blood pressure, etc. This heart disease detection system assists a patient based on his/her clinical information of them been diagnosed with a previous heart disease. The algorithms used in building the given model are Logistic regression, Random forest and XG boost.

Feature / Metric	Logistic Regression	Random Forest	XGBoost
Model Type	Linear Classifier	Ensemble (Bagging)	Ensemble (Boosting)
Test Accuracy	81.97%	80.33%	75.41%
False Positives	(Not shown)	5	7
False Negatives	(Not shown)	7	8
Interpretability	High (Coefficients)	Low	Low
Complexity	Low	High	High
Training Speed	Very Fast	Moderate	Moderate
Robustness to Outliers	Moderate	High	Moderate

Fig.4 Comparison table

B.Future Scope

- We plan to incorporate additional clinical parameters into the dataset, which is expected to enhance model accuracy and improve prediction reliability.
- Future implementations will explore more advanced and efficient classification algorithms to achieve higher performance and predictive precision.
- Expanding the size and diversity of the training data will likely improve the model’s generalisation capability and its ability to accurately predict heart disease outcomes.
- We also intend to apply advanced machine learning techniques to classify not just the presence of heart disease, but also to identify its specific type for more targeted diagnosis.

V. REFERENCES

1. World Health Organization . World Health Statistics 2021. World Health Organization; Geneva, Switzerland: 2021. [Google Scholar]

2. Chong E.K.P., Zak S.H. An Introduction to Optimization. John Wiley & Sons; Hoboken, NJ, USA: 2004. [Google Scholar]

3. Rashid T. Make Your Own Neural Network. CreateSpace Independent Publishing Platform; Scotts Valley, CA, USA: 2016. [Google Scholar]

4. https://github.com/shivam6225/HeartDiseasePrediction/blob/main/DS%20Project%20Report.pdf

5. https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/JASWANTH%20NARAYANA%20R(40738003)%20VISHESH%20K%20(40738007).pdf