IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 10, October 2026

Heart Disease Prediction Using Machine Learning with Explainable AI

Karthi Balaji M[1], Nithila K[2], Sandhya Dillybabu[3], Dr. Beenarani Manoj[4]

Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology Vadapalani, Chennai, India[1-3]

Mentor, Department of Computer Science and Engineering, SRM Institute of Science and Technology Vadapalani, Chennai, India[4]

Abstract: Heart disease is a major health issue all over the world and therefore it is important for potential risk patterns to be identified early on. Machine Learning (ML) methods are able to examine clinical data and detect patterns that can assist in putting people into one of two categories: those with heart disease and those without heart disease. The present paper describes a heart disease prediction system based on machine learning that makes use of the Cleveland portion of the UCI Heart Disease dataset. The system applies Logistic Regression and Random Forest classifiers to study patient characteristics such as age, sex, chest pain type, resting blood pressure, cholesterol level, maximum heart rate, and other clinical parameters. Before it is split into training and testing sets, the dataset goes through a preprocessing stage. The models are assessed using accuracy, precision, recall, the F1-score, and the Receiver Operating Characteristic Area Under the Curve (ROC-AUC). In order to enhance interpretability, SHAPley Additive explanations (SHAP) are employed to show the contribution of the input features to the model's predictions. A user interface based on Streamlit allows users to input patient information and see the output from the model. The system integrates prediction, model comparison, and explainability within a single application. The actual performance figures are derived from experimentation and have to be stated after the implementation has been carried out. The system is meant for educational and research use only and should not be used as a replacement for professional medical diagnosis.

Keywords: Heart disease prediction, machine learning, Logistic Regression, Random Forest, SHAP, explainable AI, UCI dataset.

1. INTRODUCTION

Heart disease is a major issue from the point of view of public health and is linked to a number of demographic, clinical, and lifestyle factors. It is useful to identify early on possible risk patterns since this can lead to further medical evaluation and to timely intervention. In the case of the traditional clinical assessment it is necessary to interpret a number of patient parameters such as blood pressure, cholesterol levels, the results of electrocardiogram measurements, and observations relating to exercise.

Machine Learning includes various methods for analyzing datasets that have a structured format and for detecting the statistical relationships between the input variables and the known results. In the case of supervised learning, the model is trained on examples that have been labelled and is then used to predict the class of records which have not been seen before. When it comes to predicting heart disease, the problem can be stated as a binary classification one in which the output indicates either the presence or the absence of heart disease in accordance with the dataset's defined target.

The project involves the development of an application for predicting heart disease by using both Logistic Regression and Random Forest. Logistic Regression is used as the basic classification method, whereas Random Forest offers an ensemble technique that is able to model nonlinear relationships. The two methods are compared on the basis of various evaluation metrics rather than just accuracy.

A further important element of the project is Explainable Artificial Intelligence (XAI). Even though a machine-learning model is able to produce a prediction, it is also useful to understand the various factors that have contributed to it. In order to show how each individual input feature affects the model's output, SHAP is used.

The system uses Python and incorporates Streamlit to provide an interactive interface for entering the patient attributes and for displaying the predictions.

 This work is licensed under a Creative Commons Attribution 4.0 International License

1.1 Objectives

The main objectives of the project are:

1. To create a machine-learning model for the purpose of classifying heart disease.
2. We need to preprocess the clinical data in order to get it ready for use in training the model.
3. We want to train and compare Logistic Regression and Random Forest.
4. We are to assess the models in terms of accuracy, precision, recall, the F1-score, and ROC-AUC.
5. To use SHAP in order to understand the contribution of each feature.
6. Create an interactive application with the use of Streamlit.

2. LITERATURE SURVEY

Research into the application of machine learning in the field of healthcare has looked at the use of classification algorithms for the purpose of detecting patterns in patient data, since prediction of heart disease is a widely studied example thereof and it involves a number of clinical variables together with a clearly defined classification target.

2.1 Heart Disease Prediction Using Clinical Datasets

The Cleveland subset of the UCI Heart Disease dataset is widely employed in machine-learning experiments that involve the classification of cardiovascular disease. It includes 303 patient records and 13 input features, these features comprising demographic information, clinical measurements, and exercise-related observations.

The dataset can be used as a useful benchmark in the development of classification models; but since it is relatively small, the results of experiments may differ depending on the train-test split, the preprocessing method, the feature encoding and the evaluation procedure. It is therefore necessary to examine the results from different studies carefully.

2.2 Logistic Regression-Based Classification

Logistic Regression is a type of supervised learning method which is often used for binary classification; it calculates the probability that an observation belongs to a certain class on the basis of its input features.

Logistic Regression is useful as a starting point in the prediction of heart disease since it is easy to train and because the coefficients can be used to explain the relationships between the input variables and the predicted outcome. The method, though, may fail when the relationships in the data are strongly nonlinear or involve complex interactions between features.

2.3 Random Forest-Based Classification

Random forest is an ensemble algorithm which averages the prediction of many decision trees. Decision trees are trained on random samples of the training data and features.

Random forests can learn non-linearity and interactions on the clinical data. Random forests can be harder to interpret the predictions so additional explanation methods needs to be used.

2.4 Explainable Artificial Intelligence Using SHAP

Lundberg and Lee. Suggested the use of SHapley Additive explanations for explaining machine-learning predictions in terms of contribution of each feature.

SHAP assigns each feature an importance value for a particular prediction in the overall model. In a heart disease prediction system, SHAP can interpret which input variables caused a certain prediction to move towards the positive class.

SHAP increases the interpretability by providing an explanation together with the predicted output. But an explanation is simply an explanation of how the model behaved, and not whether a feature causes heart disease.

2.5 Research Gap Addressed by This Project

This chosen project demonstrates how model comparison, multi-criteria evaluation, feature-level explanations, and an interactive prediction presentation can be combined in a single pipeline. Instead of simply presenting a classification result, a way to also understand its inner workings is included.

The study is constrained by the use of a small public data set and needs further validation before being used in the clinic. The primary contribution is combining these machine-learning and explainability steps within an educational demo that is reproducible.

 This work is licensed under a Creative Commons Attribution 4.0 International License

3. METHODOLOGY

The methodology framework contains dataset preparation, data preprocessing, model training, performance evaluation, explanation based on SHAP and user-interface design.

3.1 Dataset Description

It utilizes the Cleveland subset of the UCI Heart Disease dataset from the UCI Machine Learning Repository. The standard version of this data set has 303 data points (rows), 13 binary predictor variables (columns) and a label (target variable).

The input features include:

- Age
- Sex
- Type of chest pain
- Blood pressure at rest
- Serum cholesterol
- Blood sugar after fasting
- Resting electrocardiographic result
- Maximum heart rate achieved
- Exercise-induced angina
- ST depression (oldpeak)
- ST-segment slope
- Number of major vessels
- Test group for thalassemia

The initial target can be anything from 0 to 4. For binary classification experiment, the target is transformed such that 0 means no heart disease and larger than 0 is otherwise. (Encode needs to exactly match the downloaded file and implementation.)

3.2 Data Preprocessing

Before that, the data has to be prepared for the input to the training of the model. The steps of the pre-processing are as follows: check the column names, convert to numeric values of the features, look for missing values, and give shape to the target.

Missing values for a feature are imputed, e.g. with the median of the feature in the training set. Parameters of a transformer need to be fitted on the training set, not the whole data, to prevent data leakage.

Numerical Feature Scaling Logistic Regression features require feature scaling when needed. Random Forest is not require feature scaling as it depends on features thresholds to form decision rules.

Categorical values have to keep the same encodings between training and prediction. Same feature order and same way of preprocessing has to be used between training script and Streamlit application.

3.3 Train-Test Split

The data set is split into train and test samples with an 80:20 ratio. The target class proportions are approximately maintained in each subset using a stratified split.

The training data is employed to fit the machine-learning models, and the testing data is held back for when the models are tested on observations that were not previously utilized for training.

Make the experiment reproducible by setting a fixed random seed. Due to the small dataset, further work might include stratified cross-validation to get a more solid performance estimate.

3.4 Logistic Regression Model

Logistic Regression is used using the scikit-learn library. Logistic regression determines the probability that a patient record belongs to the positive class.

This is an example of the use of the method fit() to train the model on data. To make predictions to class membership from the trained model, use predict() method.

To make predictions to class membership probabilities from the trained model, use predict_proba() method.

3.5 Random Forest Model

The Random Forest classification was achieved through the use of the class Random Forest Classifier from the package scikit-learn. This classifier makes a number of decision trees, and then aggregates the result of them all.

At training time, the model has learnt patterns in the relationship between the features and the target. At prediction time, the trained model uses the new feature values to output a class prediction.

The model is compared to Logistic Regression using the same held-out test data and metrics.

3.6 Model Evaluation

The performance of the two classifiers is assessed using five commonly used metrics.
The percentage of all predictions made that are accurate.
Accuracy: Calculates the ratio of correctly predicted positive observations to the total predicted positive observations.

The ratio of accurate true positive results.
F1-score: Fusion of precision and recall by forming their harmonic mean.
ROC-AUC: Indicates the ability of a model to tell positive and negative cases apart at various decision thresholds.

A validation process is made by a table of the expected and actual responses (confusion matrix) containing, respectively, the true and false positive and the true and false negative. A ROC curve is drawn.

The final model can be compared based on the evaluation criterion pre-selected. For this project, **ROC-AUC** can be taken as the priority criterion and accuracy is a secondary consideration. The model can be obtained from the real experiment results without any a priori assumption.

3.7 SHAP-Based Explainability

The SHAP technique is used to explain the trained Random Forest model. Tree Explainer is appropriate for tree-based models and can compute the influence of single input features on model output.

The plots also can reveal the overall distribution of feature contribution over many observations. SHAP value distribution plot of features summarise the value of of each feature for each data sample. A SHAP bar plot will summarize the average across the samples in data of the absolute value of each feature contribution.

At an individual prediction level, SHAP can indicate which feature values were most responsible for the model output to be to the left or right of the decision boundary (on the way to or away from the positive class).

SHAP values but not medical conclusions or proof of causal mechanisms.

3.8 System Workflow

The complete workflow is:
UCI Heart Disease Dataset Data Preprocessing Train-Test Split Logistic Regression Model and Random Forest Model Evaluation and Comparison Best Model SHAP Explanation Streamlit User Interface

The user provides the required clinical feature values in the interface. The trained model receives these values and provides a prediction. Where supported by implementation, the interface also shows a class probability and SHAP feature contributions.

4. SYSTEM IMPLEMENTATION

Written in Python, the project can be run in Visual Studio Code on Mac with the Apple M2 processor.

4.1 Software Requirements

The primary software parts are:

- Python: Training pipeline and prediction.
- Pandas and NumPy: Use for tabular data, numerical calculations.
- Scikit-learn: has a variety of tools for preprocessing, Logistic Regression, Random Forest and evaluation.
- SHAP: Generates explanations of model predictions.
- Matplotlib: Creates performance and explanation plots.
- Joblib: Persists trained model objects for future applications.
- Streamlit: Provides the interactive user interface.

 This work is licensed under a Creative Commons Attribution 4.0 International License

4.2 Training Module

The training module loads the dataset, extracts the input feature and target variable, creates the train-test split, and then fits the algorithms.

Once trained, the module makes predictions on the test set and computes the evaluation metrics. The trained model and most of the output metrics can be saved to allow the application to use the learned model.

4.3 Prediction Interface

Use the Streamlit interface to input the 13 clinical features. These features are entered and select the relevant encoded feature categories.
Predict When the user clicks the prediction button, the app rearranges the inputs into the format the model expects. It then feeds that data into the saved model. The interface shows the predicted class and, if appropriate, its predicted probability of the positive class.
You can also render a feature-contribution plot to illustrate how SHAP explains the output of the model.
The device is intended for demonstration and experimentation purposes only. It is not intended to diagnose or treat a person.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The results of the implementation should be carried out on the chosen dataset for gathering the experimental data. The measured data must be recorded to this table.

Model	Accuracy	Precision	Recall	F1-Score	ROC_AUC
Logistic Regression	87%	81.25%	93%	86.67%	95.13
Random Forest	90%	84.38%	96.43%	90%	95.89

Table 1. Performance comparison of the machine-learning models

The comparison of the models tests to see which approach performs better when using the evaluation method that was selected. Accuracy measures the overall correctness of the model. Precision and recall give some extra metrics on the positive predictions and the positive cases that are not caught. The F1-score balances precision and recall. ROC-AUC evaluates different threshold discriminations.

Check the confusion matrix to get an idea of what type of mistakes each of the models makes. Use the SHAP summary plot to identify the features that play the most important role in the predictions of the trained Random Forest model.
The final paragraph must be related to the metrics that have been measured and the plots that have been generated. You are not allowed to write about the results of other works as if they were obtained by this implementation.

6. NOVELTY, LIMITATIONS, AND FUTURE WORK

6.1 Novelty

The primary contribution of this work is the combination of model comparison, multiple evaluation metrics, SHAP based explainability, and an interactive prediction interface in a single system.

The app adds an extra feature of running two separate machine-learning approaches and compares the results, as well as showcasing the important features of the resulting ML model. This makes it much more useful for educational purposes and model interpretation.

6.2 Limitations

The project has some limitations. The Cleveland data set is small and may not reflect the diversity of modern clinical populations. The performance estimates may further differ based on train-test split and preprocessing procedure.

The model is reliant on the input data being of a high standard and being correctly encoded. As such, a high score on this dataset does not necessarily mean that the model will generalise well to new populations. Furthermore, SHAP explains but does not justify the model's behaviour or prove the medical importance of any one variable.

The system is a prototype for research purposes and is not intended to be used for clinical diagnosis.

7. CONCLUSION

This paper is an implementation of a machine-learning system for the classification of heart disease using the Cleveland data subset of the UCI Heart Disease data set. This system provides training of and comparison between Logistic Regression and Random Forest models, and uses accuracy, precision, recall, F1-score and ROC-AUC to evaluate.

The following were implemented to show the power of classification, comparison, and explainability in this research: SHAP is used to generate feature-level interpretations of the trained model, the RF, while Streamlit is used to offer a user-friendly platform to enter clinical features and get the prediction. A combination of classification, comparison, and explainability allows a valuable framework for demonstrating machine-learning in health care.

The ultimate performance conclusions should be drawn from the metrics collected in the experiment. Additional work such as cross validation, independent data sets, and more models comparisons are needed in order to evaluate the generalization. The mentioned system is built for teaching and research purposes only and does not substitute medical professional diagnosis.

REFERENCES

[1]. Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository: Heart Disease Dataset*. University of California, Irvine. https://archive.ics.uci.edu/dataset/45/heart+disease

[2]. Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30. https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[3]. Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[4]. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html

[5]. Streamlit. *Streamlit Documentation*. https://docs.streamlit.io/

[6]. scikit-learn developers. *Scikit-learn: Machine Learning in Python — User Guide*. https://scikit-learn.org/stable/user_guide.html