VitalPath: A Cardiovascular Risk Assessment Framework Using Bayesian-Optimized Ensembles and SHAP

Yash Choudhary[1], Raja Singh[2], Vaibhav Sharma[3], Ravindra Chauhan[4]

Department of Computer Science & Engineering, R.D. Engineering College, Ghaziabad

DOI: https://doi.org/10.51584/IJRIAS.2026.11040004

Received: 04 April 2026; Accepted: 09 April 2026; Published: 24 April 2026

ABSTRACT

Heart disease is one of the leading factors of death in whole world. Yet, predicting it early remains a huge setback. Doctors face two main problems in hospitals: patient files usually contain empty fields and complex AI models act like black boxes, making them hard to trust for users. VitalPath AI is designed to solve these issues in efficient and reliable manner. First, we tackle data gaps using the MICE algorithm. This lets us fill in missing patient details without throwing away valuable data. Next, we use SMOTE to balance the dataset classes and level the playing field, ensuring the model learns fairly and prevents any model from generating biased outcomes. Instead of just guessing settings, we used Bayesian Optimization to hunt down the optimal configurations for several machine learning models. When evaluated on the UCI Heart Disease dataset, AdaBoost came out on top by ending up attaining an AUC-ROC score of 0.963. This performance metrics surpassed what we originally hoped for, though accuracy alone isn't the only objective for this work to accomplish. To make the model easy for doctors to trust, we need transparency. That's why we integrated SHAP, which breaks down exactly why each prediction was made, letting doctors and patients see if factors like cholesterol or chest pain drove the decision.

Keywords: Bayesian Optimization, Cardiovascular diseases (CVDs), Explainable AI (XAI), Machine Learning, MICE, SHAP, SMOTE.

INTRODUCTION

Think of a doctor evaluating patients in a high-volume hospital environment. They only get a few minutes to figure out a patient's heart disease risk. But the patient's file is complicated, and some lab tests are totally missing**.** To tackle these issues, we developed the VitalPath AI framework. We built a smart assistant that helps clinicians predict heart disease accurately, while keeping the final decision incredibly easy to understand. Hospitals urgently require these kinds of automated solutions in modern healthcare systems. Cardiovascular diseases (CVDs) continue to remain a serious global health burden, contributing to roughly 19.8 million deaths yearly, making roughly 32 percent of all deaths across the globe or about one-third of all deaths worldwide. Doctors now have access to big data of digital health records. Yet, pulling actionable insights from this data remains challenging. With today's massive amounts of medical data, these older methods struggle to find hidden patterns—like how age, cholesterol, and blood pressure actually interact with each other. Machine Learning (ML) fixes this by offering a much stronger way to analyse real data.

Literature Survey

Recent research proves that stacking various ML algorithms by combining multiple models, can achieve high accuracy in heart disease prediction. These modern tools can dig up complex patterns hidden deep in patient data. They easily outsmart older methods when it comes to raw accuracy. Despite these impressive leaps in accuracy, two major challenges are holding back the use of AI in real hospitals:

The "Black Box" Problem: The strongest predictive models operate completely in the dark. They give an answer as a "High Risk" warning, but can't explain why? In medicine, the "why" is everything. A doctor cannot confidently make a treatment plan based on a blind guess from a computer.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Imperfect Data Handling: Too many researchers just chase high accuracy scores. They ignore the fact that real medical records are incomplete. Patient records often have missing values or suffer from class imbalance (where there are far more healthy patients than sick ones), which can lead to unreliable models if not handled correctly.

Work Focus	Advanced Imputation (MICE)	Handles Class Imbalance (SMOTE)	Systematic Optimization (Bayesian)	Provides Explainability (XAI)
Ensemble Accuracy Study (2021)	✗	✓	✗	✗
XAI-focused Study (2022)	✗	✗	✗	✓
Multi-Model Comparison (2023)	✗	✗	✗	✗
Hybrid Methods Study (2025)	✓	✓	✗	✗
VitalPath AI Framework	✓	✓	✓	✓

Table I: Literature Comparison Table

As the comparison table highlights, previous research tends to focus on just a single piece of the maze. It is super rare to find a system that handles both bad data and model transparency at the same time**.** That is exactly why we built VitalPath AI. We wanted to create a practical system that actually works from start to finish. We do this in three main steps:

[1] **Data First:** Instead of throwing away bad records, we actively fix them. We use MICE to fill in the blanks and SMOTE to balance the classes, giving our models a clean, unbiased slate to learn from.

[2] **Peak Performance:** Second, we push for maximum performance. Instead of guessing, we let Bayesian Optimization automatically hunt down the perfect settings across seven different algorithms.

[3] **Full Transparency:** Finally, we force the AI to be transparent. We eliminate the black box problem entirely. By deploying SHAP, our system generates a simple waterfall graph and report showing doctors and patients exactly which factors drove the final prediction.

PROPOSED METHODOLOGY

We developed a step-by-step pipeline to build a reliable and transparent heart disease prediction system with clean data, optimized models, and clear insights.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

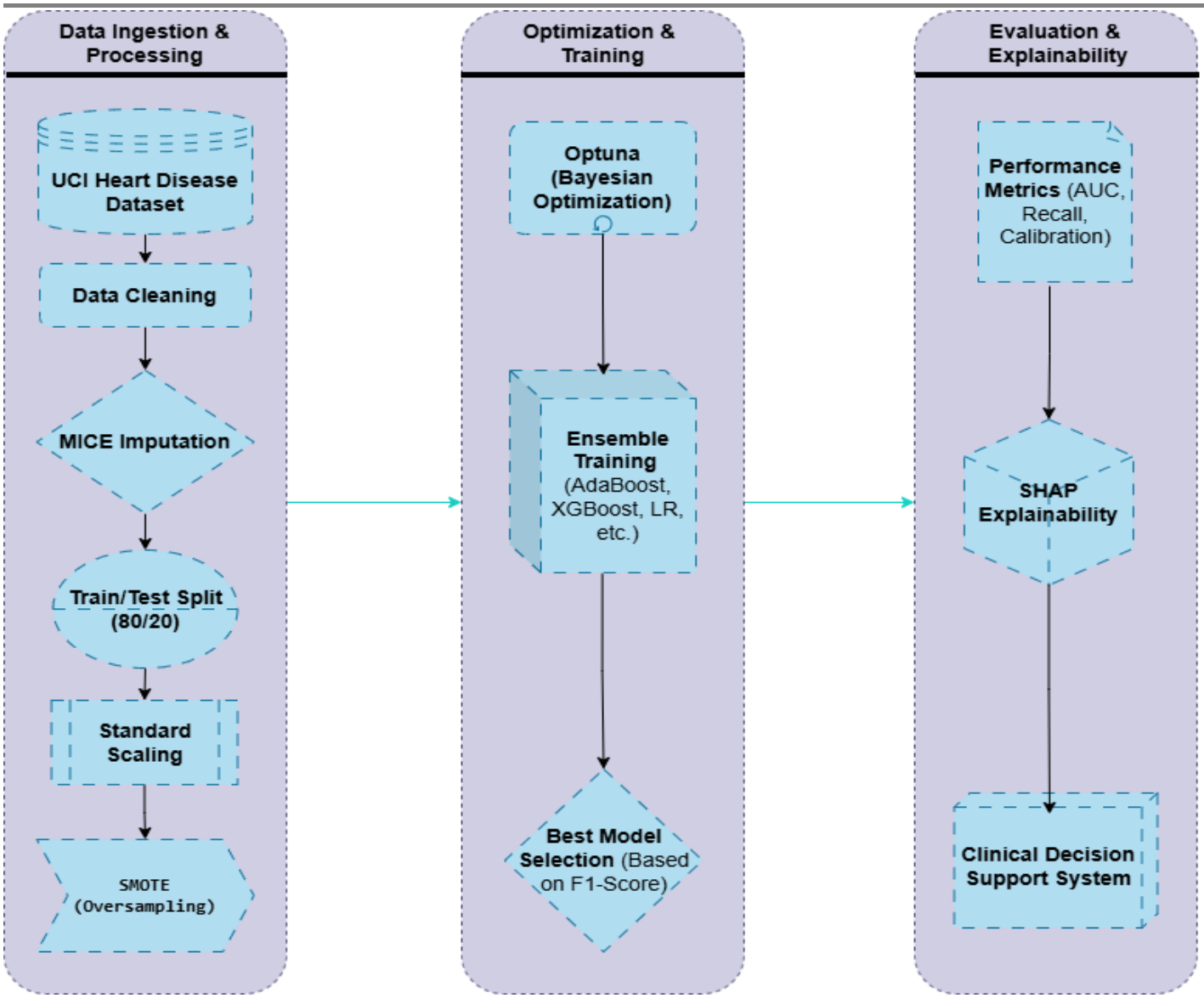

Figure 1: The Proposed 'VitalPath AI' Research Framework

Dataset Source Details

For this study, we used the well-known "Cleveland Heart Disease" dataset provided by the UCI ML open repository. It includes 303 patient records, each tracking 13 clinical features like age, cholesterol, and chest pain type. The original target variable rated disease severity from 0 to 4. We simplified this into a binary problem: 0 for "No Disease" and 1 for "Disease Present."

Data Preprocessing Pipeline

Real-world medical data is often imperfect. So, we set up a strict pipeline to ensure our models learned the right lessons. Here is how we built our robust pipeline:

Missing Value Imputation: Rather than throwing away rows or filling blank spots with arbitrary numbers, we relied on MICE to calculate the missing details mathematically based on the patient's other stats.

Data Splitting and Scaling: To prevent "data leakage," we first split the dataset into an 80% training set and a 20% hold-out test set. Next, we applied standard scaling. The Standard Scaler was fitted on the training data and then applied to the test data.

Handling Class Imbalance with SMOTE: To level the playing field, we applied SMOTE (Synthetic Minority Over-sampling Technique) to our training data to balance the classes and force the models to learn fairly.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Model Suite

To identify the ideal model for this prediction engine, we evaluated seven machine learning models.

Model	Description
Logistic Regression	A linear model that predicts how likely a specific disease result is. It's fast, highly interpretable, and serves as an excellent baseline.
Support Vector Machine	A maximal-margin classifier that finds the best possible boundary between classes. It is very effective in high-dimensional spaces.
Decision Tree	A simple, flowchart-like model that uses a series of "if-else" rules. It is easy to visualize and understand.
Random Forest	An ensemble of many Decision Trees. It combines their votes to improve accuracy and prevent the overfitting common in single trees.
AdaBoost	A sequential boosting ensemble that learns from the mistakes of previous models, focusing on the "hardest" to classify data points.
XGBoost	An advanced form of boosting that is famous for delivering high speed, scalability, and built-in handling of missing values.
CatBoost	A specialized gradient boosting algorithm designed to handle categorical features (like "Chest Pain Type") natively without complex preprocessing.

Table II: Model Suite table

Hyperparameter Optimization

Tuning a model's settings (hyperparameters), makes a massive difference in how well it performs. Instead of using the old, slow method of Grid Search to blindly test every single combination, we went with Bayesian Optimization through the Optuna framework. Optuna is much smarter. It builds a working probability map of the different settings. It learns from past trials to quickly hunt down the best possible setup.

The steps we followed for every algorithm were as follows:

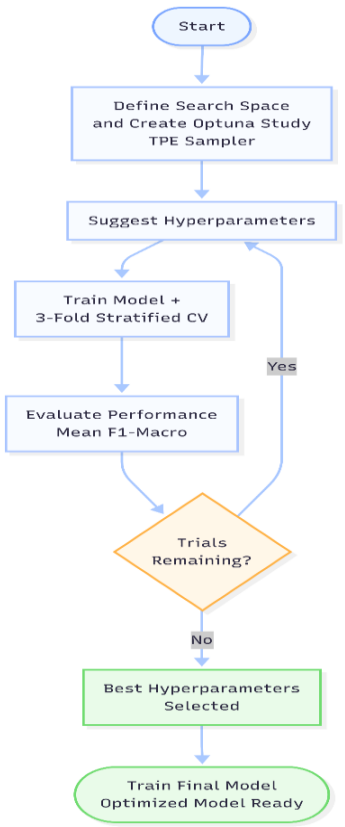

Figure 2: Bayesian Optimization Flow Chart

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Model Explainability using SHAP

Finally, we had to break open the "black box." A highly accurate model is useless in a hospital if doctors cannot see how it works. To fix this, we integrated SHAP (SHapley Additive exPlanations). Built on the math of game theory mathematics, this tool assign exact credit to each specific feature deserves for the final outcome.

Results

Here, we look at how the seven machine learning models actually performed. These scores are entirely based on the unseen data. We tested them on the SMOTE-balanced dataset and optimised their settings using Bayesian methods.

Quantitative Performance Analysis

To provide a complete picture of each model's capabilities, we evaluated them across four standard classification metrics: Accuracy, Precision, Recall, and F1-Score.

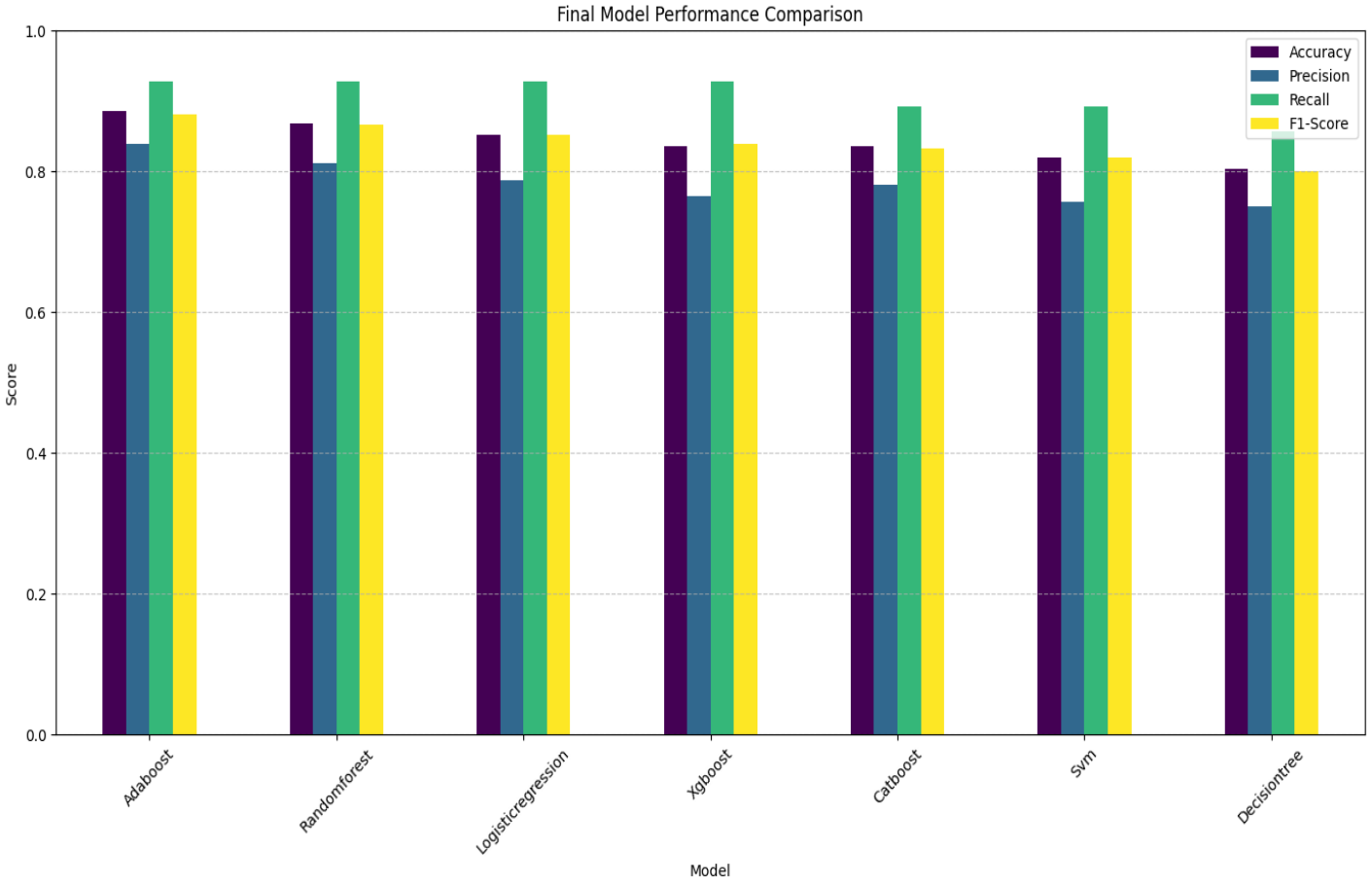

Figure 3: Bar chart comparing the final performance of all seven tuned models on the hold-out test set

The numbers show that ensemble methods—especially AdaBoost and Random Forest—easily beat out the single Decision Tree. AdaBoost clearly managed to hit the highest F1 score overall. It struck an excellent balance between precision (correctly identifying high-risk patients) and recall (making sure we find all the actual disease cases).

Analysis of Misclassifications

When you are diagnosing patients, knowing the type of mistake a model makes is far more crucial than just checking the overall score. For medical professionals, failing to identify an actually ill person is disastrous. Calling a sick person ‘healthy’ is the worst possible mistake.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

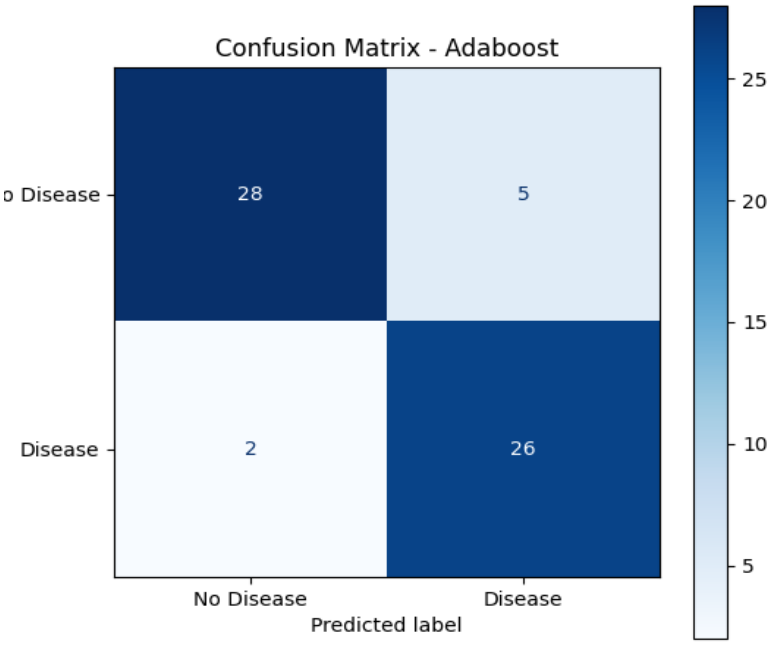

Figure 4: Confusion Matrix for the AdaBoost model, our peak-performing model

The confusion matrix for our best model, AdaBoost, shows outstanding results on the test set. Out of the 28 patients who genuinely had heart disease, the model correctly flagged 26 of them. That means, it just only misses 2 cases.

Discriminative Ability (ROC & P-R Curves)

To see how well each model could tell the difference between Disease and No Disease classes, we mapped out their ROC and Precision-Recall curves.

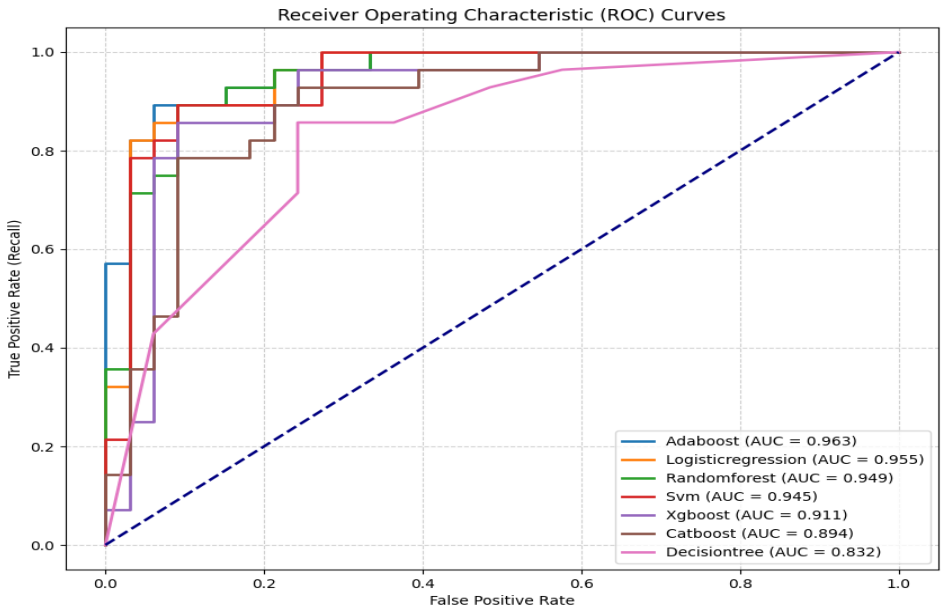

Figure 5: ROC curves for all seven models

The ROC analysis confirms that AdaBoost is the most powerful discriminator, achieving the highest Area Under the Curve (AUC-ROC) of 0.963. That number shows exactly how good it is at ranking patient by their actual risk level.

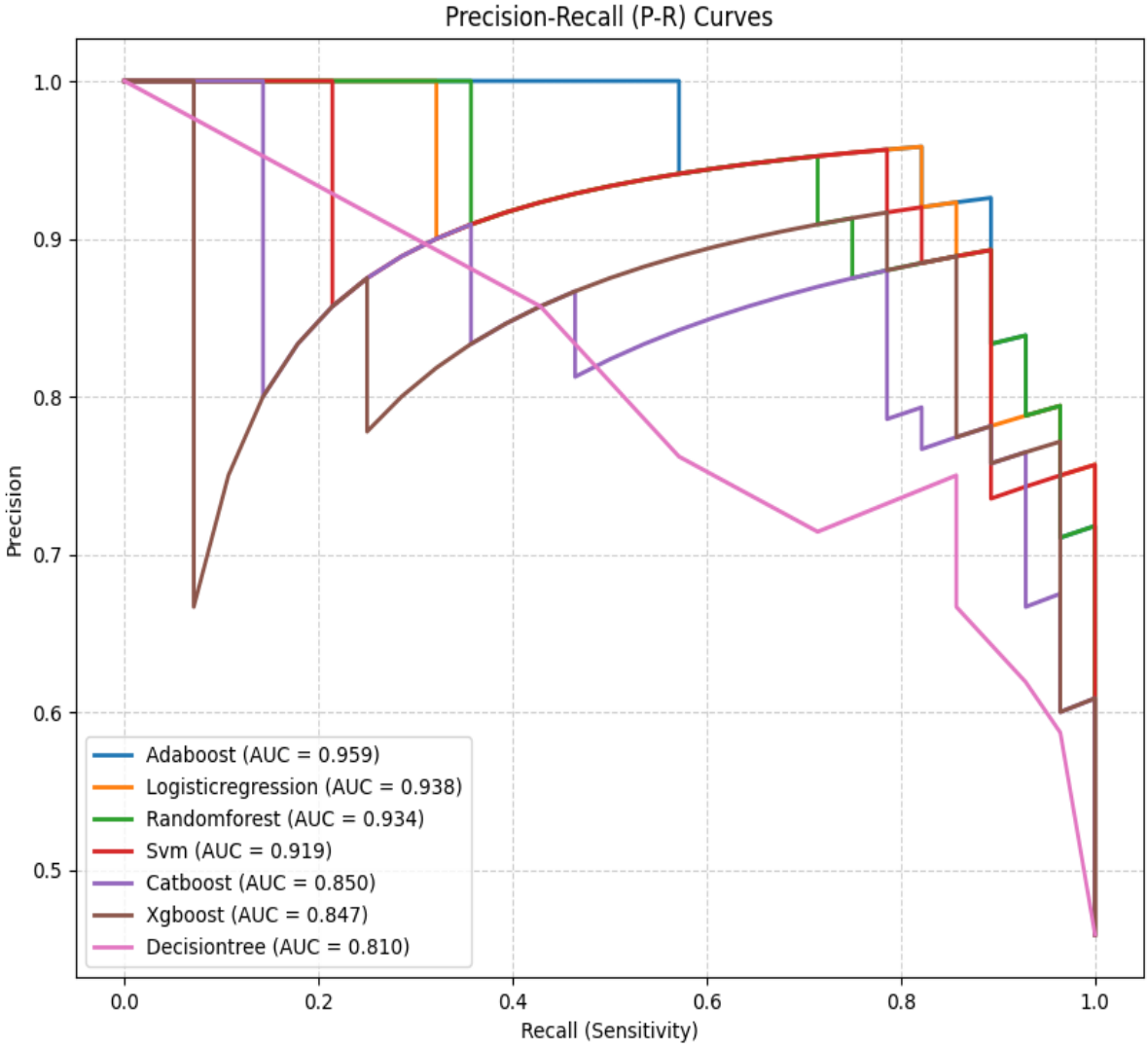

Figure 6: Precision-Recall curves

On top of that, the Precision-Recall curve analysis completely reinforces this. AdaBoost took the lead again with the highest AUC-PR of 0.959. This shows it can maintain high precision (keeping false alarms low) while still catching a high percentage of true positive cases (high recall).

Model Interpretability Analysis (SHAP)

To figure out exactly what drove the predictions inside our best-performing "black box" models, we deployed SHAP. The analysis revealed a strong consensus across the different tree-based models. The SHAP summary plot shows our algorithms are genuinely learning on clinically relevant patterns. The features consistently ranked as most impactful were:

[1] **ca** (number of major vessels)

[2] **thal** (thallium stress test result)

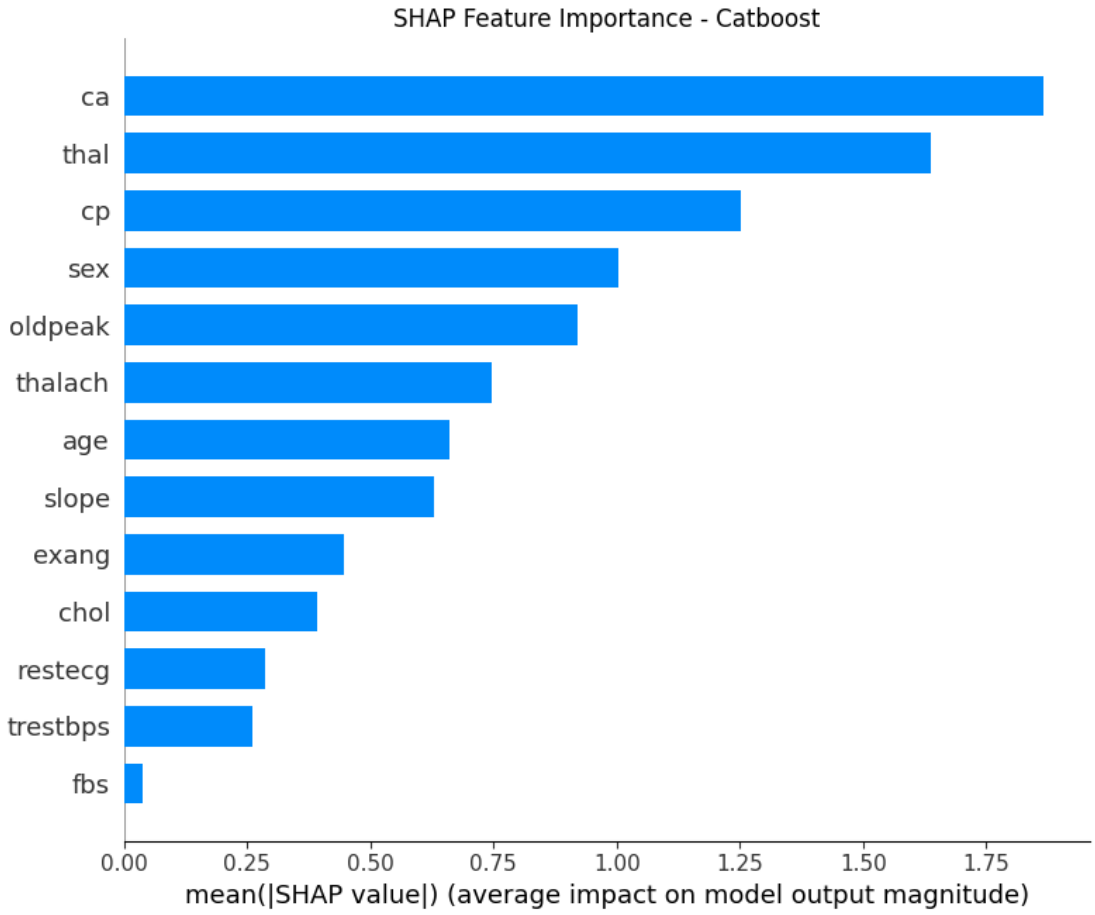

Figure 7: SHAP Summary Plot for the CatBoost model

CONCLUSION

The goal of this project was to create an intelligent and dependable tool for cardiac risk prediction and our findings prove we succeeded. With our system, VitalPath AI, we demonstrated that combining proper data preprocessing with strong machine learning methods yields highly precise predictions, even if dealing with complex medical data. Throughout our trials, it became clear that our AdaBoost model performed the best when it was trained on samples levelled out by SMOTE, achieving a high AUC-ROC score of 0.963, which means it can effectively distinguish between patients individuals facing severe danger and those who are not. At the same time, we didn't just focus on accuracy; by using SHAP, we made the model's decisions easier to understand. Instead of acting like a "black box," the system can explain its predictions using meaningful medical factors such as chest pain type and thallium stress test results, which helps build trust among doctors. When we set out to build VitalPath AI, our goal was simple but ambitious: to create a tool that a doctor could actually trust. Looking at the results, we believe we have achieved that objective.

Limitations

While our framework demonstrates strong performance, it's important to acknowledge its limitations in the context of a rapidly evolving AI landscape:

Static Learning Model: Our current system operates on a "train once, deploy forever" model. It does not learn from new patient data that comes in after deployment. This means the model's knowledge is frozen in time and it cannot adapt to new trends in disease patterns or treatments without being manually retrained.

Lack of Causal Inference: The model is a powerful pattern-matcher, but it does not understand cause and effect. For example, SHAP can tell us that a high "Thallium Stress Result" is links strongly to severe danger, yet it cannot tell us if treating the thallium defect would lower the risk. Our framework provides correlation, not causation.

Tabular Data Only: The system is exclusively designed for structured, tabular data (like an Excel sheet). It cannot incorporate unstructured data like a doctor's handwritten notes, X-ray images, or ECG signal waveforms, which contain a wealth of diagnostic information.

"Explainability" vs. "Actionability": While SHAP explains why a prediction was made, it doesn't automatically suggest the next best action. It doesn't tell a patient, "To lower the risk, focus on reducing their cholesterol by 20." The step from explanation to actionable advice is still a manual one.

Future Scope

Building on our current framework, we envision a next-generation clinical decision support system with the following advanced capabilities:

Federated and Continual Learning: Instead of training on one central dataset, we plan to deploy a Federated Learning approach. This would allow the model to learn from data across multiple hospitals without the data ever leaving the hospital, thus preserving patient privacy. This would be combined with Continual Learning, where the model is incrementally updated with new patient data on a weekly or monthly basis, allowing it to adapt and improve over time.

Causal AI Integration: We will explore integrating Causal Inference models. The goal is to move from "What factors are associated with risk?" to "If we change this factor (e.g., lower BP by 10 mmHg), by how much will the patient's risk actually decrease?" This would provide doctors with a powerful simulation tool for treatment planning.

Multi-Modal AI Architecture: The future of medical AI is multi-modal. We plan to evolve the framework to incorporate data from different sources into a single, unified prediction model. This would involve using:

a. Computer Vision to analyse cardiac MRI or CT scans.

b. Natural Language Processing (NLP) to extract insights from clinical notes.

c. Signal Processing to analyse real-time ECG data.

Generative AI for Actionable Recommendations: We will investigate using Large Language Models (LLMs), fine-tuned on medical guidelines, to translate the SHAP explanations into human-readable, actionable advice.

REFERENCES

1. Shah, P., Shukla, M., Dholakia, N. H., & Gupta, H. "Predicting cardiovascular risk with hybrid ensemble learning and explainable AI." Scientific Reports, (2025). 15(1), Article 17927.
2. Rao, C.; Li, M.; Huang, T.; Li, F. "Stroke Risk Assessment Decision-Making Using a Machine Learning Model: Logistic-AdaBoost." CMES-Comput. Model. Eng. Sci., (2024) 139, pp. 699–724.
3. Almazroi, E. A. Aldhahri, S. Bashir and S. Ashfaq. "A Clinical Decision Support System for Heart Disease Prediction Using Deep Learning," in IEEE Access, (2023). vol. 11, pp. 61646-61659.
4. Khan, M. A., & Algarni, F. "A Healthcare Monitoring System for the Diagnosis of Heart Disease in the IoMT Cloud Framework Using Explainable AI." IEEE Access, (2022). vol. 10, pp. 63583-63593.
5. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. "Optuna: A Next-generation Hyperparameter Optimization Framework," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, (2019).
6. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. "CatBoost: unbiased boosting with categorical features," in Advances in Neural Information Processing Systems (NeurIPS), (2018). vol. 31.
7. Lundberg, S. M., & Lee, S. I. "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems (NeurIPS), (2017). vol. 30.
8. Chen, T., & Guestrin, C. "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016). pp. 785-794.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

9. S. van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate Imputation by Chained Equations in R," Journal of Statistical Software, (2011). vol. 45, no. 3, pp. 1–67.
10. Janosi, A., Steinbrunn, W., Pfisterer, M., & Detrano, R. "Heart Disease Data Set." UCI Machine Learning Repository. (1988).