CAPSTONE PROJECT PHASE- 2 REVIEW-1

Optimization of Heart Disease Prediction using Machine Learning Model

Team Members:

- 21BCE10225 ARCHITA GUPTA
- 21BCE10406 SONALI RAGHUWANSHI
- 21BCE10439 PRIYANSHI YADAV
- 21BCE10669 TINA CHELWANI
- 21BCE10708 ABHINAV SHRIVASTAVA

Supervisor
Dr.J. Manikandan

Reviewer 1
Dr.Sasmita Padhy

Reviewer 2
Dr. Antima Jain

OBJECTIVE

Heart Disease:

- Affects heart structure and function, involving blood vessels, rhythm, or muscle issues.
- Leading cause of death globally

Common Types:

- Coronary Artery Disease (CAD): Narrowing/blockage of blood vessels.
- Arrhythmias: Irregular heartbeats.
- Heart Failure: Ineffective blood pumping.
- Congenital Heart Defects: Structural abnormalities from birth.
- Cardiomyopathy: Heart muscle diseases.
- Heart Valve Diseases: Valve dysfunction.

OBJECTIVE

- Machine Learning for Prediction:
  - Processes medical datasets to detect hidden patterns and risk factors.
  - Enables early, accurate heart disease prediction.
- Advantages of Machine Learning:
  - Non-invasive, scalable, cost-effective, and precise.
  - Empowers healthcare professionals with data-driven decision-making.
  - Supports timely interventions, improving patient outcomes.

Objectives:

- Develop a robust predictive model using AdaBoost, XGBoost, and Gradient Boosting for heart disease detection.
- Compare and analyze the performance of these algorithms based on evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.
- Identify and rank the most significant features contributing to heart disease prediction.
- Provide insights for healthcare professionals to assist in early detection and prevention strategies.

PROBLEM STATEMENT

Cardiovascular diseases are one of the leading causes of mortality globally, and early prediction of heart disease can significantly reduce the risks by enabling timely intervention. The goal of this project is to develop a machine learning-based predictive model to determine the likelihood of an individual having heart disease based on their clinical and lifestyle attributes.

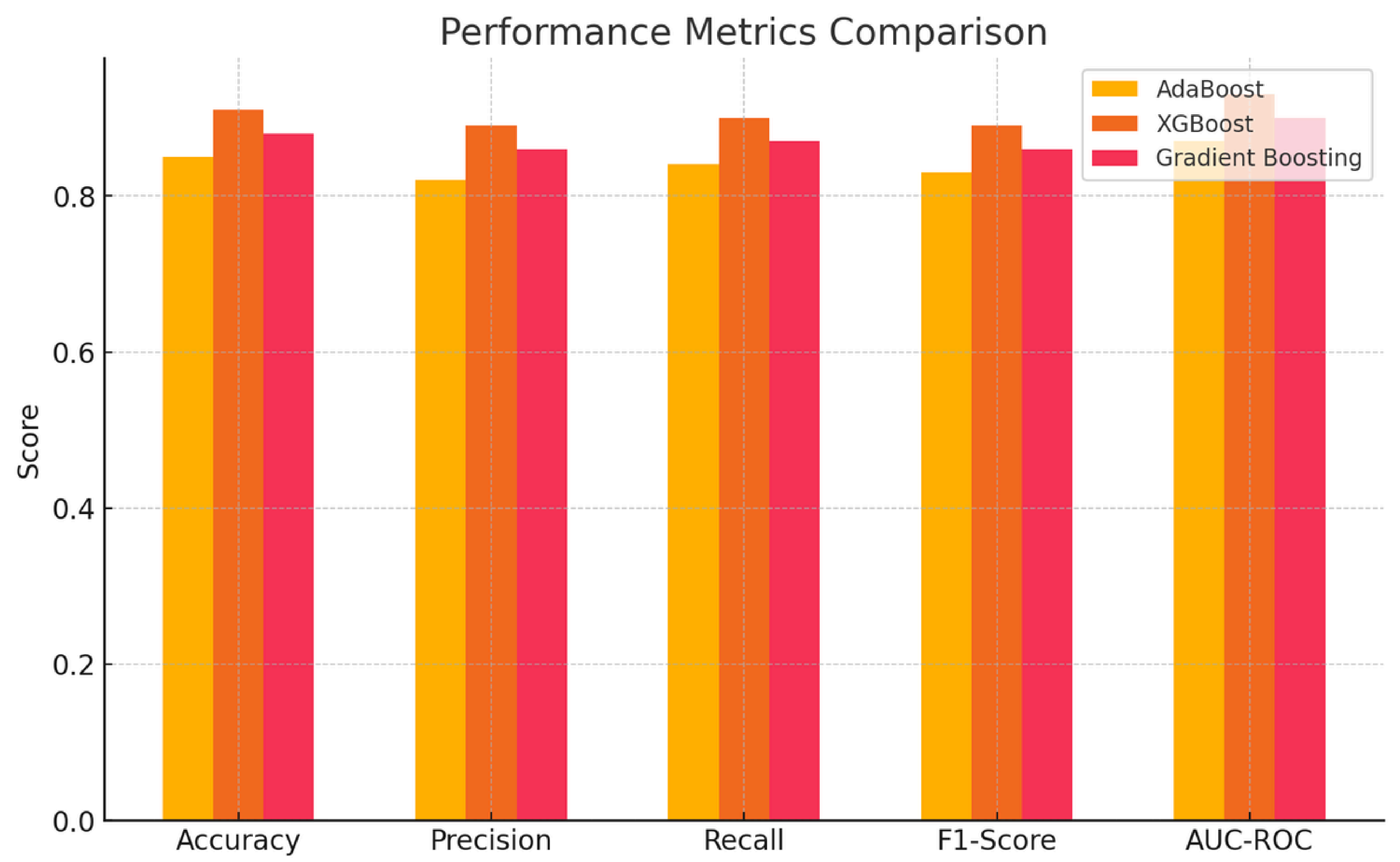
- To improve predictive performance and robustness, this project will utilize advanced boosting algorithms, specifically:

- AdaBoost (Adaptive Boosting): To iteratively improve weak classifiers by focusing on misclassified samples and aggregating their outputs to form a strong predictive model

.

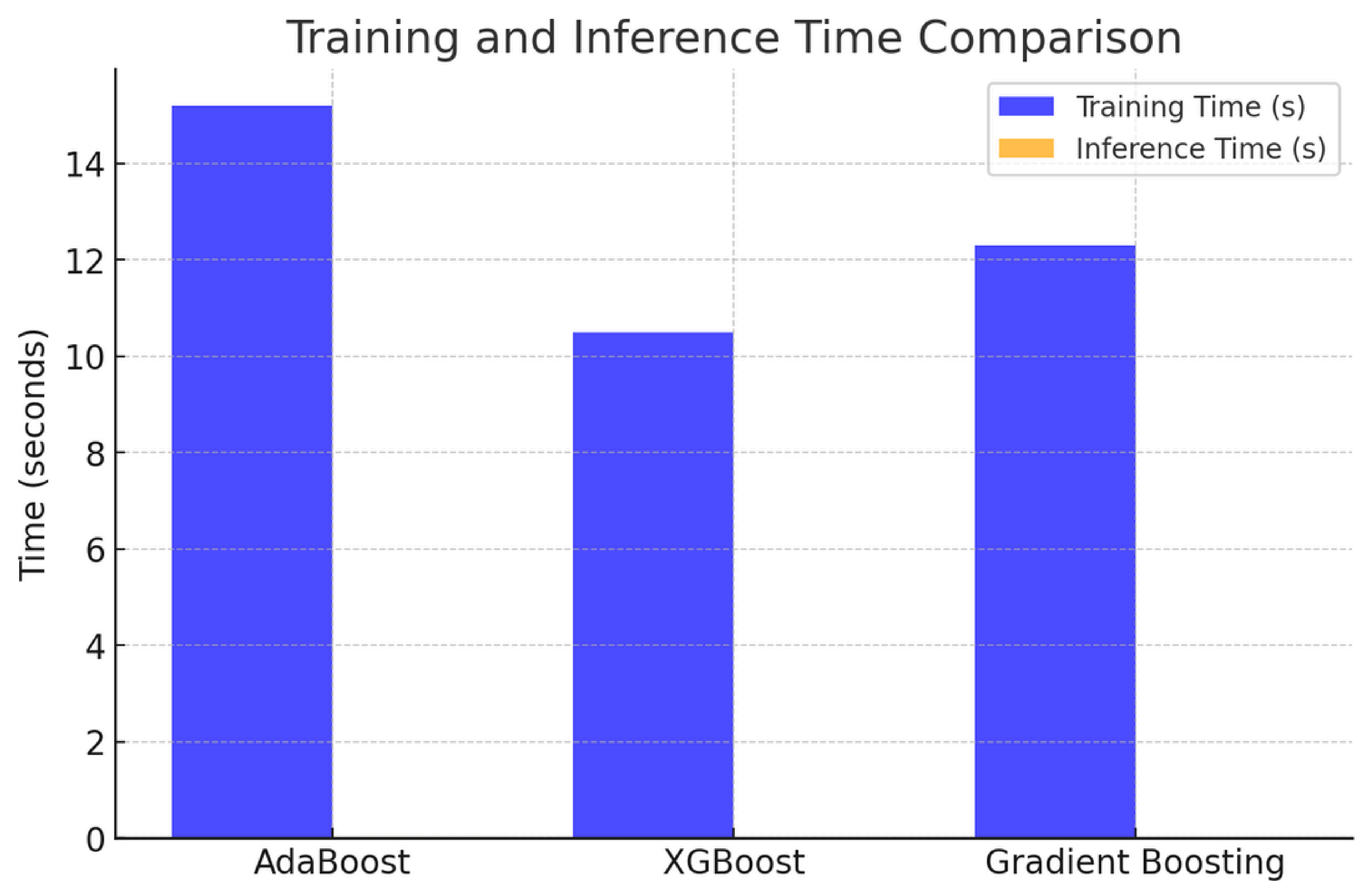
- XGBoost (Extreme Gradient Boosting): To leverage gradient-boosting techniques with optimizations for speed and performance, such as parallelization, regularization, and efficient handling of missing data.

- Gradient Boosting: To build a predictive model incrementally by combining decision trees, focusing on minimizing the prediction error through gradient optimization.



- **PERFORMANCE METRICS COMPARISON:** THIS BAR CHART COMPARES THE ACCURACY, PRECISION, RECALL, F1-SCORE, AND AUC-ROC SCORES OF ADABOOST, XGBOOST, AND GRADIENT BOOSTING. IT HELPS VISUALIZE THE RELATIVE PERFORMANCE OF THE ALGORITHMS.

- **TRAINING AND INFERENCE TIME COMPARISON:** THIS CHART COMPARES THE TRAINING AND INFERENCE TIMES FOR EACH ALGORITHM, SHOWING THEIR COMPUTATIONAL EFFICIENCY.



MOTIVATION OF STUDY

Heart disease is a major global health concern, with millions of people affected annually. The ability to predict heart disease early can drastically improve outcomes by enabling timely intervention and treatment. Machine learning provides powerful tools to analyze large, complex datasets and identify hidden patterns that traditional statistical methods may overlook. Among these tools, boosting algorithms—such as AdaBoost, XGBoost, and Gradient Boosting—have shown remarkable performance in classification tasks due to their ability to combine weak learners into a strong predictive model

Why Focus on Boosting Algorithms?

Accuracy and Robustness:
Boosting algorithms iteratively improve predictions by focusing on difficult-to-classify samples. This results in highly accurate models that perform well even with complex datasets.
In heart disease prediction, where small inaccuracies can have significant consequences, boosting methods can help minimize errors.

Feature Importance Analysis:
Boosting algorithms provide insights into feature importance, helping healthcare professionals identify key risk factors (e.g., blood pressure, cholesterol, glucose levels) contributing to heart disease.

Handling Imbalanced Data:
Heart disease datasets often have class imbalances (e.g., fewer cases of disease compared to healthy individuals). Boosting algorithms can handle these scenarios effectively by adjusting weights to focus on minority classes.

Comparative Performance:
Studying multiple boosting algorithms allows researchers to evaluate their strengths and weaknesses:
AdaBoost: Simple and effective, particularly with smaller datasets, but may struggle with noisy data.
XGBoost: Optimized for speed and performance with advanced regularization techniques, making it suitable for large-scale datasets.
Gradient Boosting: A versatile approach that builds strong models incrementally but may require longer training times compared to XGBoost.

Practical Applications in Healthcare:
Boosting algorithms are interpretable and reliable, making them ideal for critical applications like heart disease prediction, where model transparency is important for clinical decision-making.

Key Objectives of the Study:

1. Evaluate the effectiveness of AdaBoost, XGBoost, and Gradient Boosting in predicting heart disease.
2. Compare their performance on metrics like accuracy, precision, recall, F1-score, and AUC-ROC.
3. Analyze computational efficiency (training and inference times) for real-time applications in healthcare.
4. Identify key clinical and lifestyle factors influencing heart disease using feature importance analysis.

By studying these algorithms, the project aims to advance the development of machine learning-based tools for heart disease prediction, improve healthcare outcomes, and contribute to the growing body of research in medical AI applications.

DIFFERENCE BETWEEN PROPOSED MODEL AND EXISITNG MODEL

- **EXISTING MODELS:**
- TYPICALLY RELY ON SIMPLER ALGORITHMS LIKE:
- LOGISTIC REGRESSION: LINEAR MODEL, HIGHLY INTERPRETABLE BUT MAY UNDERPERFORM ON NON-LINEAR DATA.
- DECISION TREES: EFFECTIVE FOR SMALLER DATASETS BUT PRONE TO OVERFITTING.
- SUPPORT VECTOR MACHINES (SVM): PERFORMS WELL FOR SMALL DATASETS WITH HIGH-DIMENSIONAL DATA BUT COMPUTATIONALLY EXPENSIVE.
- NEURAL NETWORKS: SUITABLE FOR COMPLEX, NON-LINEAR DATA BUT OFTEN REQUIRES LARGE DATASETS AND IS LESS INTERPRETABLE.
- THESE MODELS OFTEN FAIL TO ACHIEVE A BALANCE BETWEEN ACCURACY, ROBUSTNESS, AND INTERPRETABILITYDUE TO:
- OVERFITTING WITH COMPLEX MODELS (E.G., DEEP LEARNING ON SMALL DATASETS).
- UNDERFITTING WITH SIMPLER MODELS (E.G., LOGISTIC REGRESSION ON NON-LINEAR RELATIONSHIPS).

- **PROPOSED MODEL:**

- COMBINES WEAK LEARNERS (E.G., DECISION TREES) THROUGH BOOSTING ALGORITHMS TO ITERATIVELY IMPROVE PREDICTIONS.

- USES:

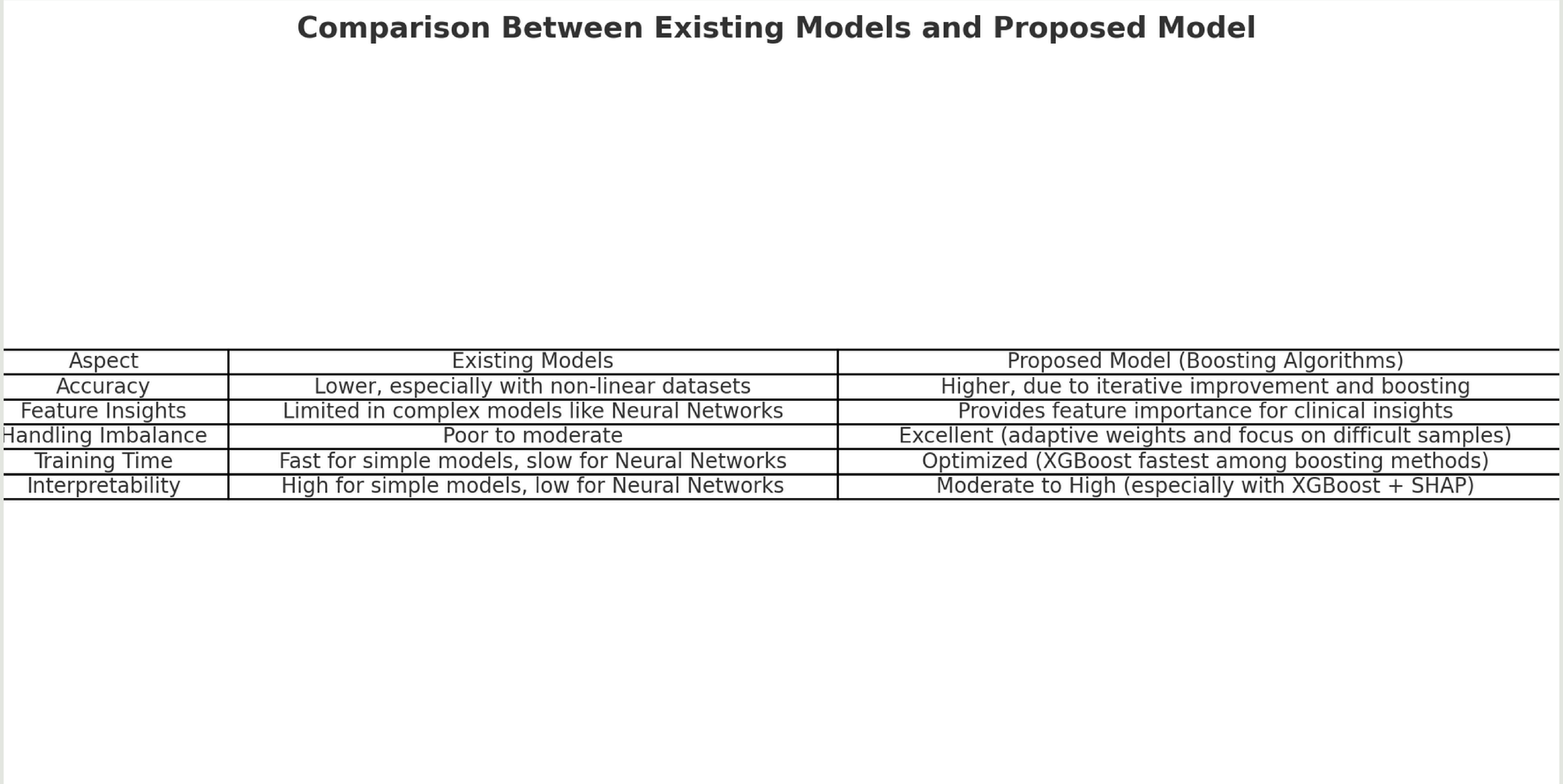
- ADABOOST: FOCUSES ON HARD-TO-CLASSIFY SAMPLES BY ASSIGNING HIGHER WEIGHTS TO MISCLASSIFIED INSTANCES.

- XGBOOST: OPTIMIZED IMPLEMENTATION OF GRADIENT BOOSTING WITH FEATURES LIKE REGULARIZATION, PARALLEL PROCESSING, AND EARLY STOPPING.

- GRADIENT BOOSTING: GRADUALLY IMPROVES MODEL ACCURACY BY MINIMIZING PREDICTION ERRORS THROUGH GRADIENT DESCENT.

- OFFERS:

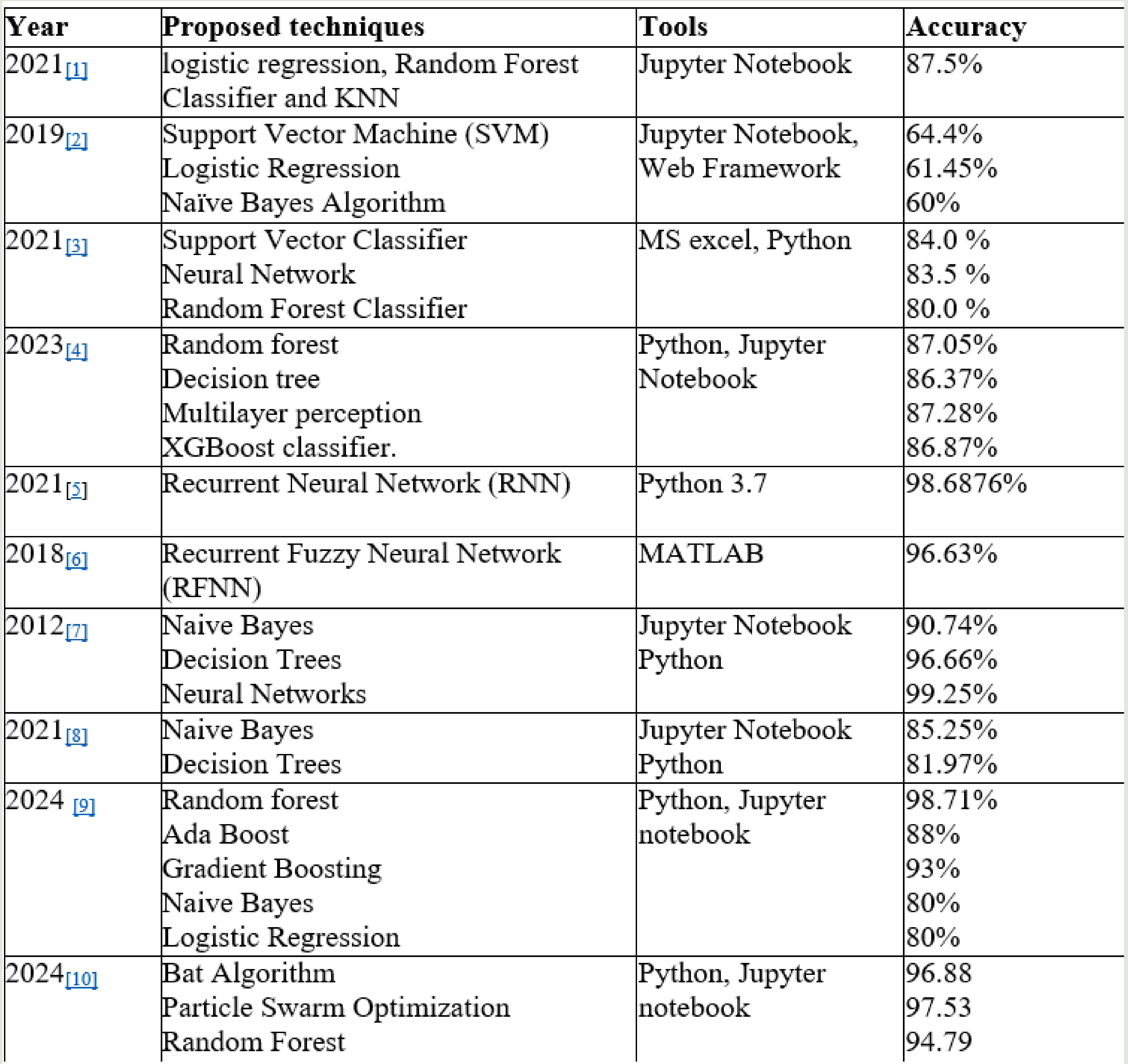
- BETTER HANDLING OF IMBALANCED DATASETS COMMON IN MEDICAL DOMAINS.
- ABILITY TO MODEL NON-LINEAR RELATIONSHIPS EFFECTIVELY.
- FEATURE IMPORTANCE ANALYSIS FOR INTERPRETABILITY.



Comparison Between Existing Models and Proposed Model

Aspect	Existing Models	Proposed Model (Boosting Algorithms)
Accuracy	Lower, especially with non-linear datasets	Higher, due to iterative improvement and boosting
Feature Insights	Limited in complex models like Neural Networks	Provides feature importance for clinical insights
Handling Imbalance	Poor to moderate	Excellent (adaptive weights and focus on difficult samples)
Training Time	Fast for simple models, slow for Neural Networks	Optimized (XGBoost fastest among boosting methods)
Interpretability	High for simple models, low for Neural Networks	Moderate to High (especially with XGBoost + SHAP)

HERE IS A CHART THAT COMPARES THE EXISTING MODELS WITH THE PROPOSED MODEL USING ADABOOST, XGBOOST, AND GRADIENT BOOSTING



Year	Proposed techniques	Tools	Accuracy
2021[1]	logistic regression, Random Forest Classifier and KNN	Jupyter Notebook	87.5%
2019[2]	Support Vector Machine (SVM) Logistic Regression Naïve Bayes Algorithm	Jupyter Notebook, Web Framework	64.4% 61.45% 60%
2021[3]	Support Vector Classifier Neural Network Random Forest Classifier	MS excel, Python	84.0 % 83.5 % 80.0 %
2023[4]	Random forest Decision tree Multilayer perception XGBoost classifier.	Python, Jupyter Notebook	87.05% 86.37% 87.28% 86.87%
2021[5]	Recurrent Neural Network (RNN)	Python 3.7	98.6876%
2018[6]	Recurrent Fuzzy Neural Network (RFNN)	MATLAB	96.63%
2012[7]	Naive Bayes Decision Trees Neural Networks	Jupyter Notebook Python	90.74% 96.66% 99.25%
2021[8]	Naive Bayes Decision Trees	Jupyter Notebook Python	85.25% 81.97%
2024 [9]	Random forest Ada Boost Gradient Boosting Naive Bayes Logistic Regression	Python, Jupyter notebook	98.71% 88% 93% 80% 80%
2024[10]	Bat Algorithm Particle Swarm Optimization Random Forest	Python, Jupyter notebook	96.88 97.53 94.79

COMPARISON OF ALGORITHMS

1.Logistic Regression

Pros:

-
- Simple and interpretable.
- Fast training and prediction.
- Works well with linearly separable data.

Cons:

- Poor performance with non-linear data.
- Limited capability in handling feature interactions.

2.KNN

Pros:

- Simple to implement and understand.
- Performs well with smaller datasets.

Cons:

- Computationally expensive for large datasets.
- Sensitive to irrelevant features and noise.

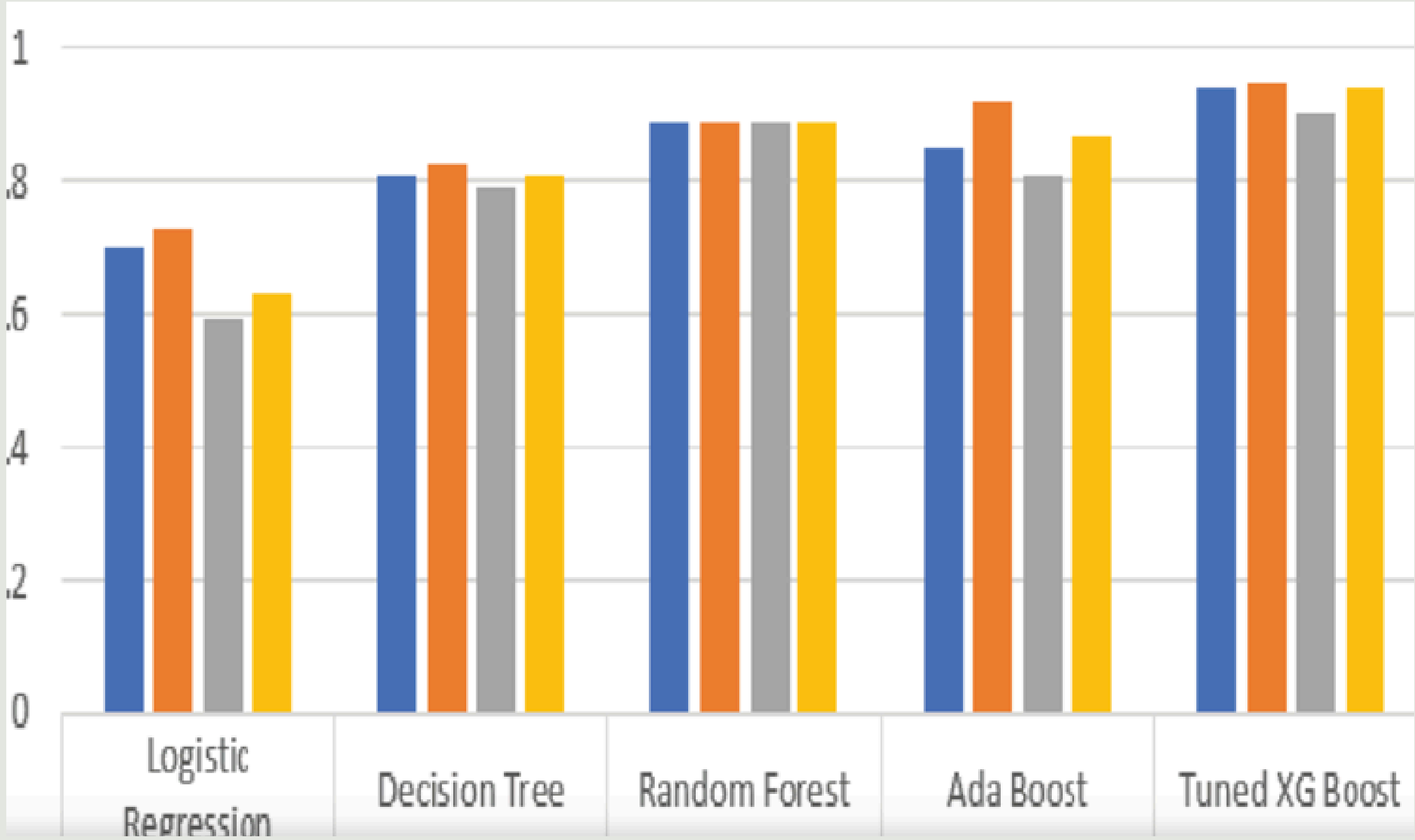
3.Random Forest

Pros:

- Robust to overfitting.
- Provides feature importance insights.
- Handles non-linear relationships effectively.

Cons:

- Computationally intensive for large datasets.
- Less interpretable than simpler models.



RESEARCH GAP IN EXISTING WORK VS PROPOSED WORK

- Traditional algorithms like Logistic Regression and KNN have limited accuracy with non-linear data, while XGBoost and Adaboost excel.

- Scalability issues plague traditional models with large datasets, but XGBoost handles them efficiently, and Adaboost performs well with more resources.

- Random Forest often overfits, whereas XGBoost reduces overfitting with regularization, and Adaboost can do so with proper tuning.

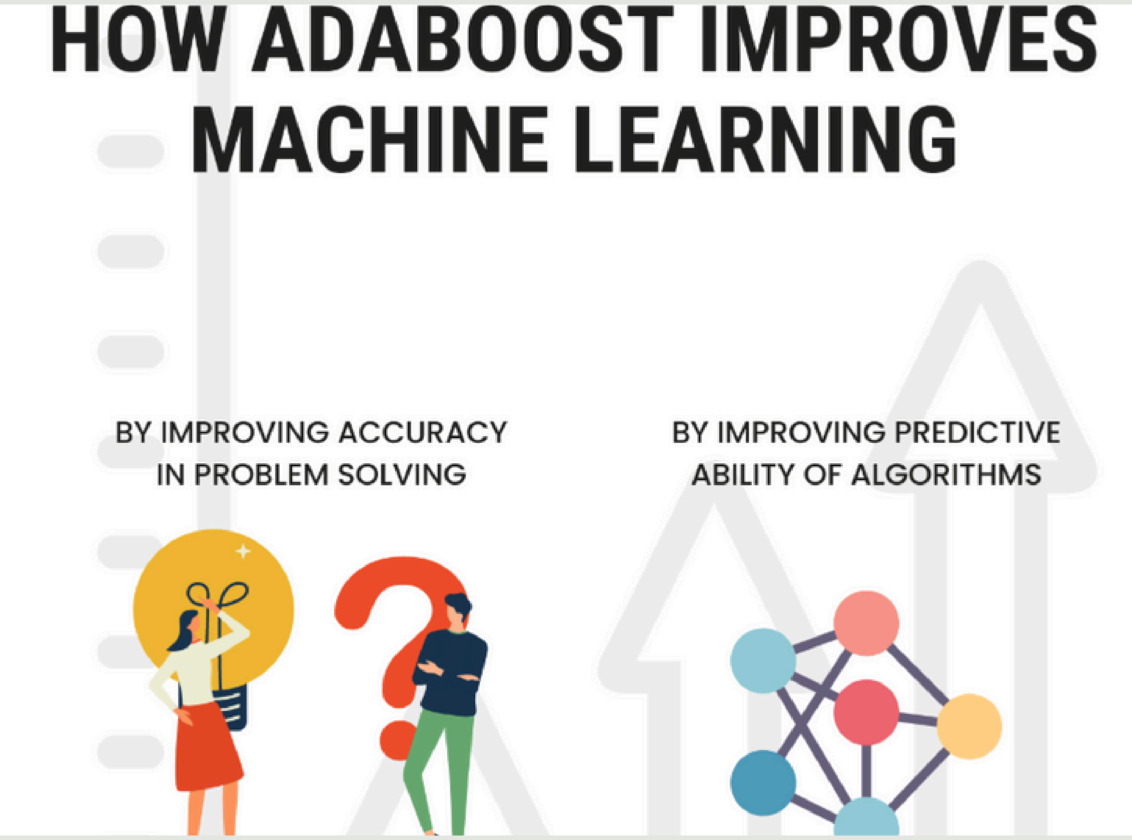
- Traditional algorithms lack boosting, limiting performance, while XGBoost and Adaboost enhance accuracy and robustness through advanced boosting techniques.

WHY CHOOSE XGBOOST AND ADABOOST?

The existing approaches, while effective to some degree, fall short in handling complex, high dimensional datasets.

Adaboost and XGBoost stand out due to their ability to:

- Manage complex data interactions with high accuracy.
- Reduce overfitting through regularization (XGBoost) and iterative learning (Adaboost).
- Provide feature importance, aiding in better interpretability and insights.
- These algorithms also excel in handling imbalanced datasets and noise, making them highly suitable for medical predictions where data quality and class imbalance are common concerns.


HOW ADABOOST IMPROVES MACHINE LEARNING
BY IMPROVING ACCURACY IN PROBLEM SOLVING
BY IMPROVING PREDICTIVE ABILITY OF ALGORITHMS

Major Parameters Considered:

1) ap_lo

2) ap_hi

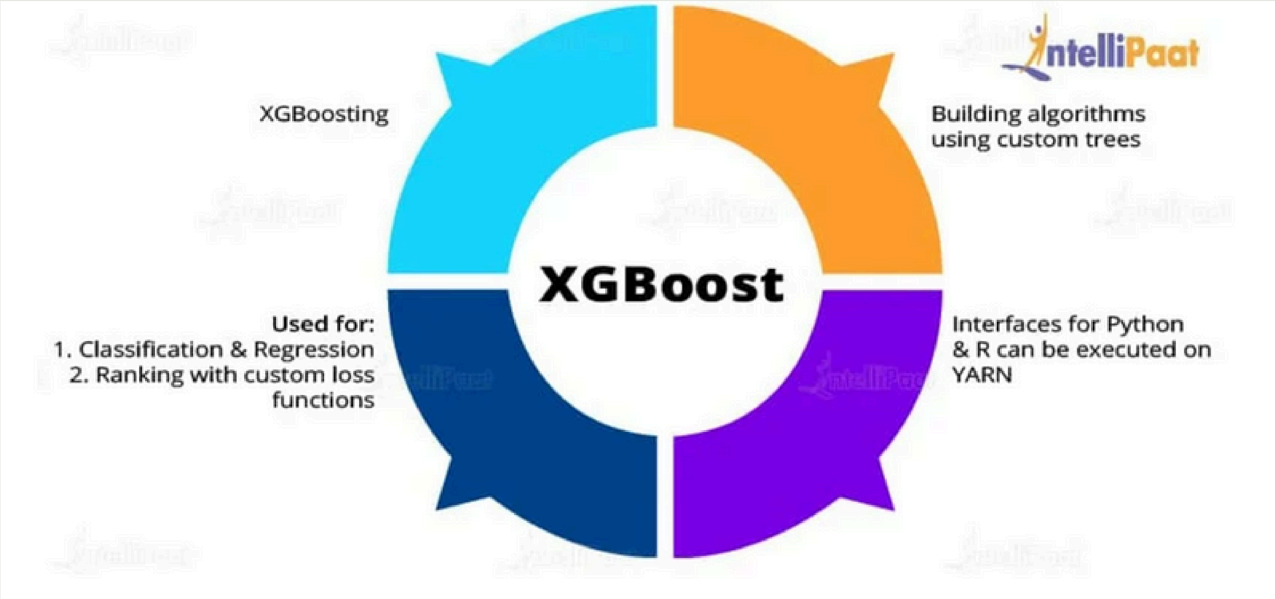
3) cholestrol

4) gender

5) activity levels

6)Alcohol consumption

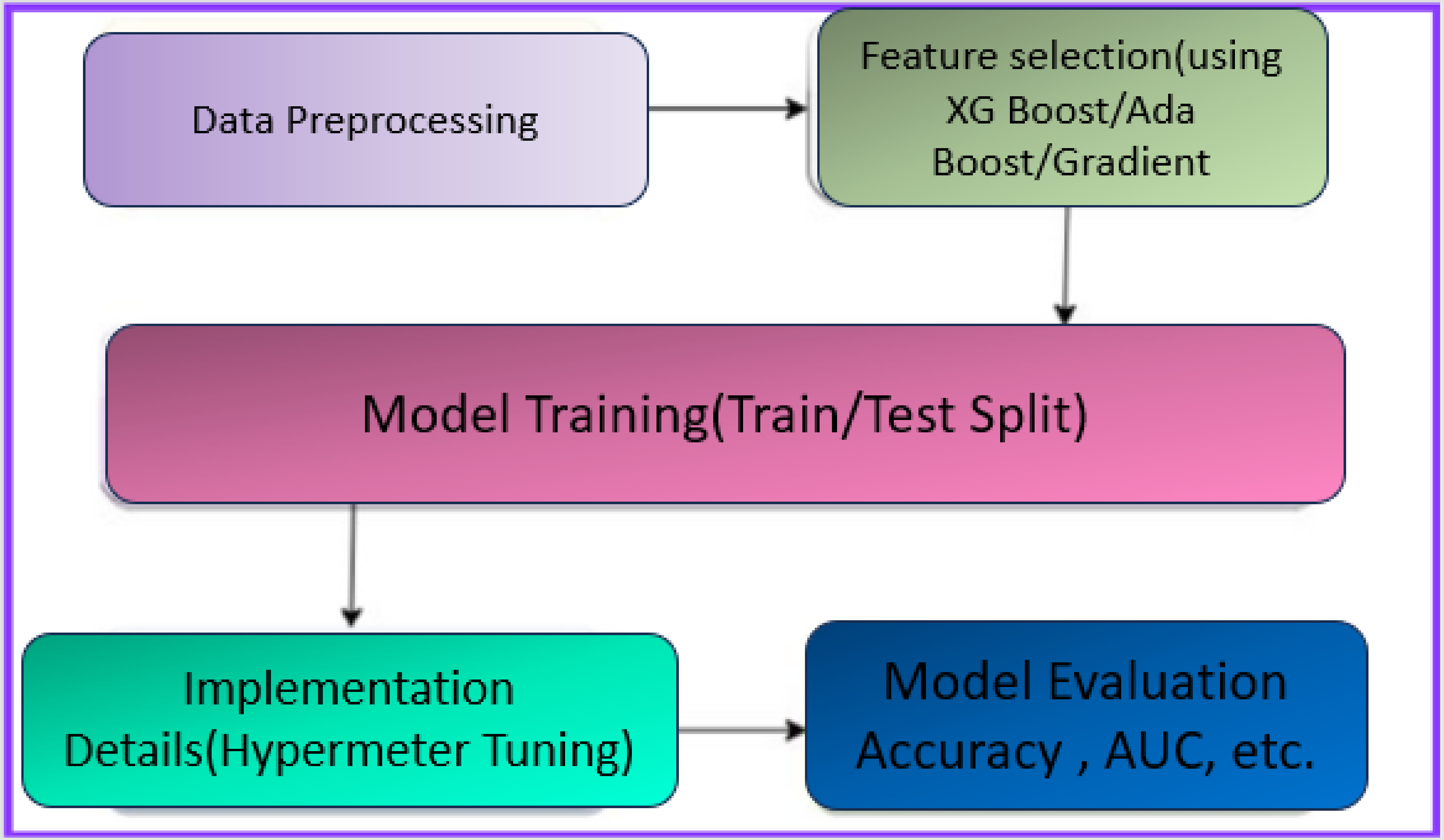
7) Smoke consumption


IntelliPaat
XGBoosting
Building algorithms using custom trees
XGBoost
Used for:
1. Classification & Regression
2. Ranking with custom loss functions
Interfaces for Python & R can be executed on YARN

SCOPE OF THE PROJECT

THE PROJECT AIMS TO LEVERAGE MACHINE LEARNING TECHNIQUES TO DEVELOP A PREDICTIVE MODEL THAT CAN ASSESS THE LIKELIHOOD OF HEART DISEASE IN INDIVIDUALS BASED ON CLINICAL DATA. THE SCOPE ENCOMPASSES VARIOUS STAGES, FROM DATA PREPROCESSING, FEATURE SELECTION AND MODEL DEVELOPMENT TO TUNING OF THE USED MODEL AND COMPARASION WITH OTHER MODELS.

ARCHITECHTURE/ WORK FLOW



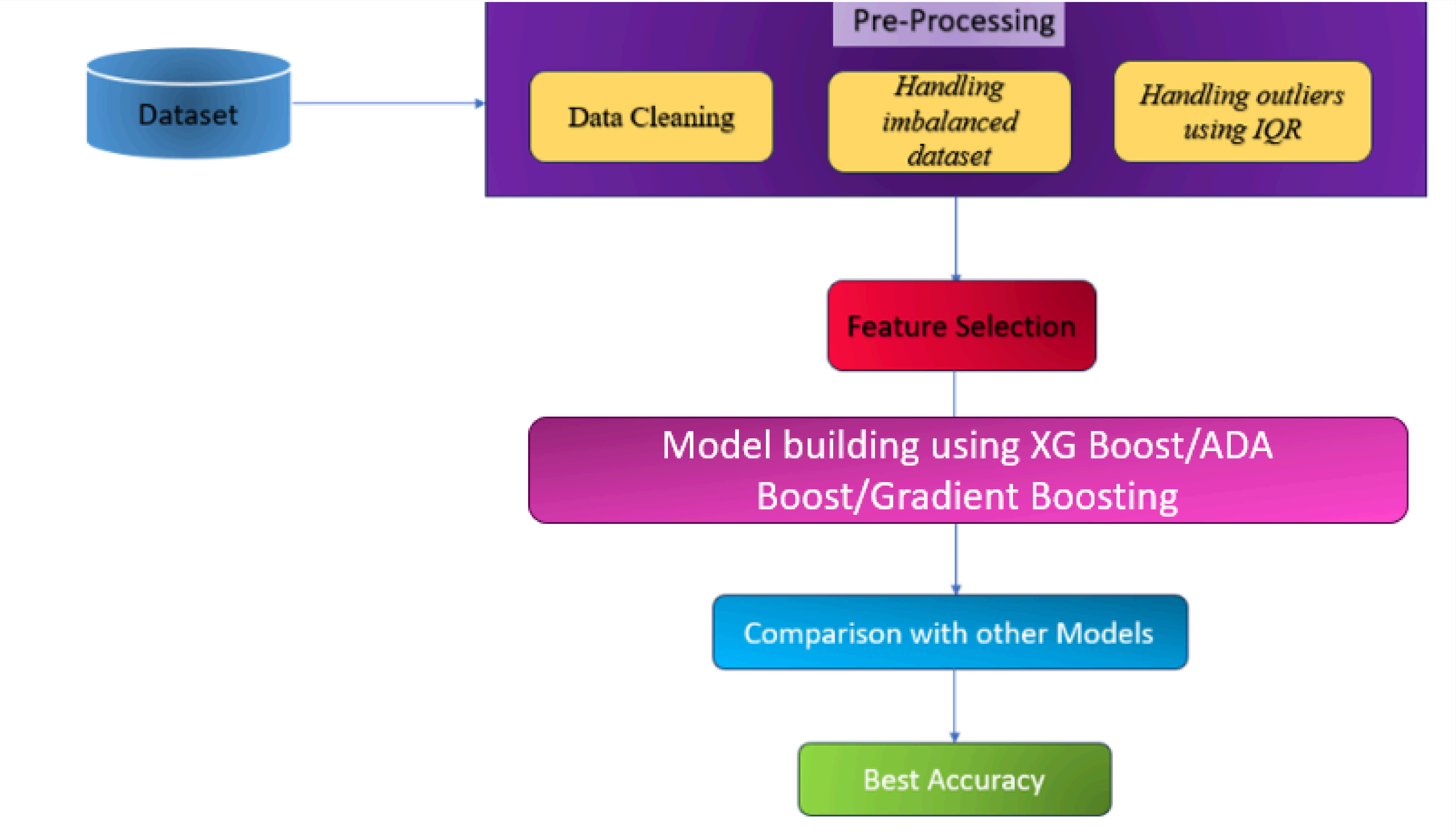
Data Preprocessing

Feature selection(using XG Boost/Ada Boost/Gradient

Model Training(Train/Test Split)

Implementation Details(Hypermeter Tuning)

Model Evaluation Accuracy , AUC, etc.


Dataset
Pre-Processing
Data Cleaning
Handling imbalanced dataset
Handling outliers using IQR
Feature Selection
Model building using XG Boost/ADA Boost/Gradient Boosting
Comparison with other Models
Best Accuracy

PROPOSED WORK

1.Data Collection

.

- 10,000 records with 13 parameters.
- 12 Features-> id, age , gender, height, weight, systolic blood pressure, diastolic blood pressure, cholesterol, glucose level, smoking, alcohol consumption, physical activity.
- 1 Feature-> Result, Heart Disease or not
- Numeric data -> id, age, height, weight, systolic blood pressure, diastolic blood pressure
- Binary data -> smoking, alcohol consumption, physical activity
- Category data-> cholesterol, glucose level, gender

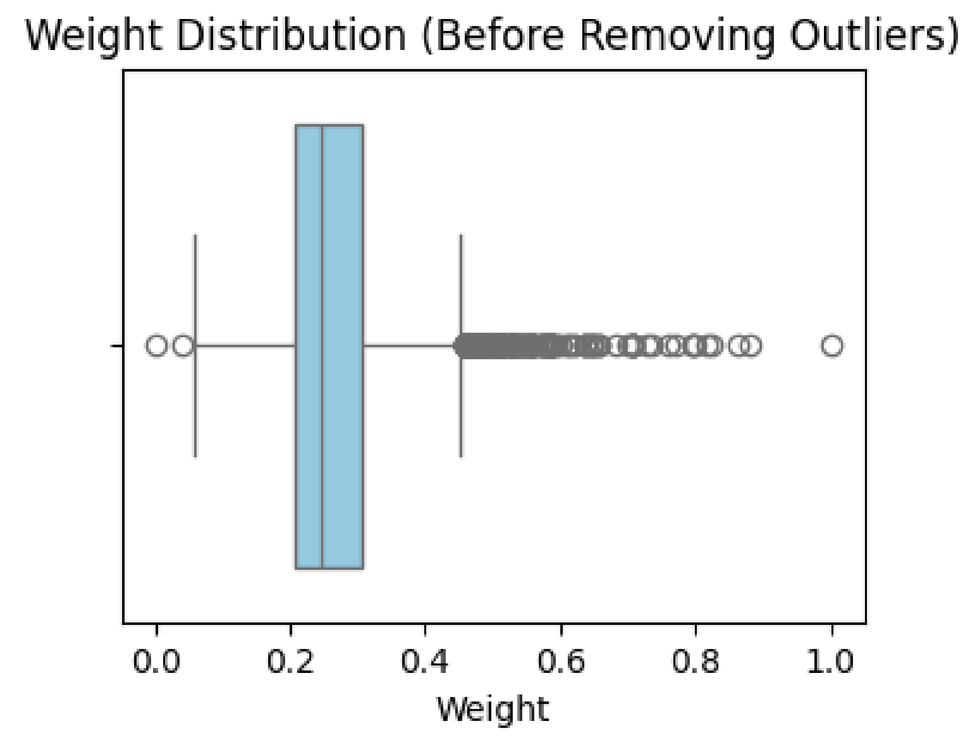
2. Data Preprocessing

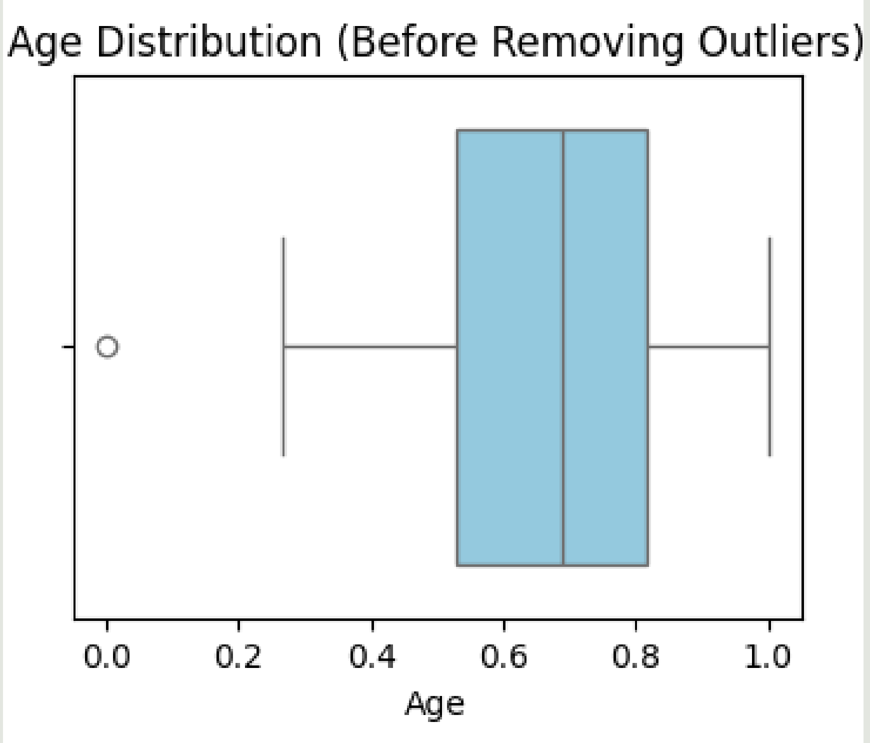
- Handling Imbalanced dataset
    - Already balanced, 5,030 negative and 4,969 positive cases
    - Model learns effectively from both classes

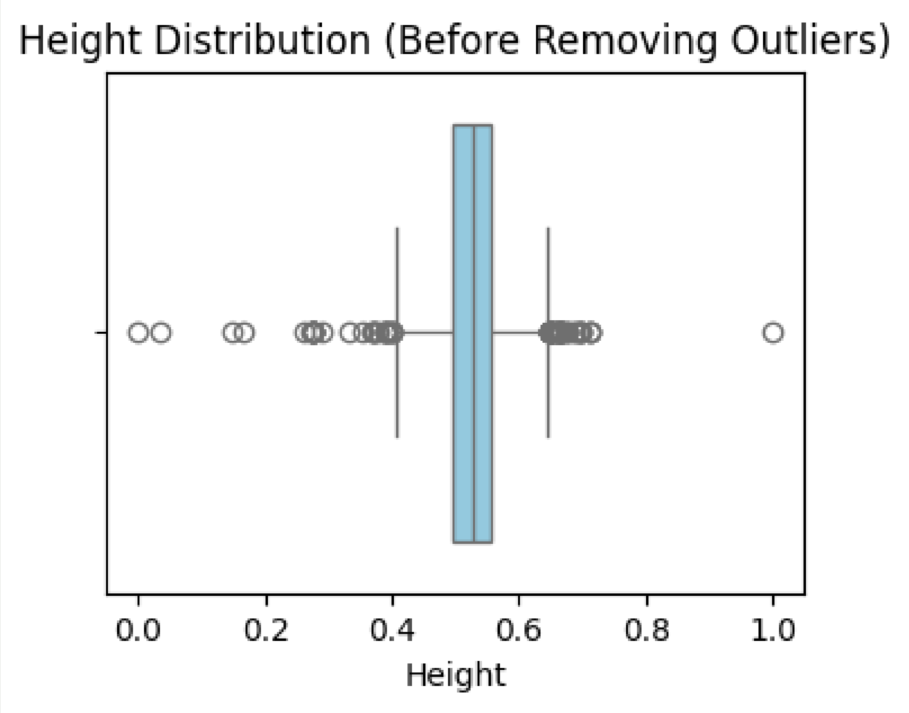
- Data Cleaning
    - Heatmap to Find missing values
    - No missing values found
    - No risk of biased results.

- Handling Outliers using IQR
    - Outlier identification and removal
    - QR method in Age , Weight and Height feature
    - IQR method in Age , Weight and Height feature
    - Outliers were defined as data points outside Q3 + 1.5 * IQR or Q1 – 1.5 * IQR.
    - Reduced bias and enhanced model accuracy.

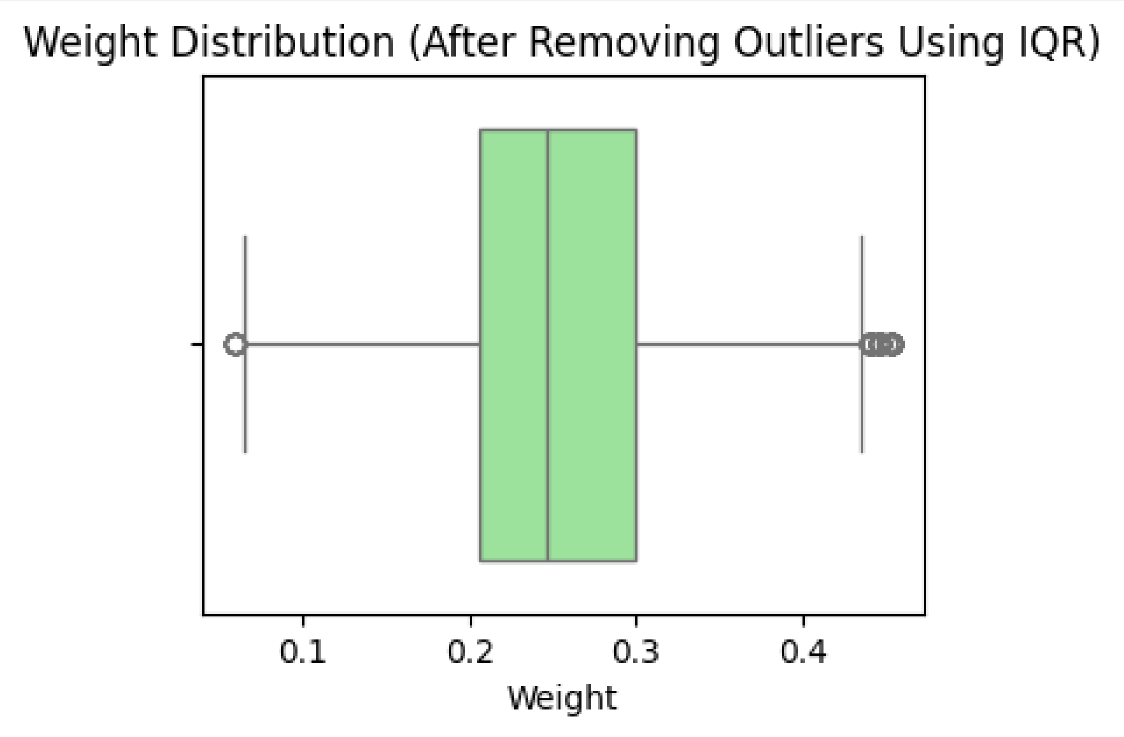
Before Handling Outliers

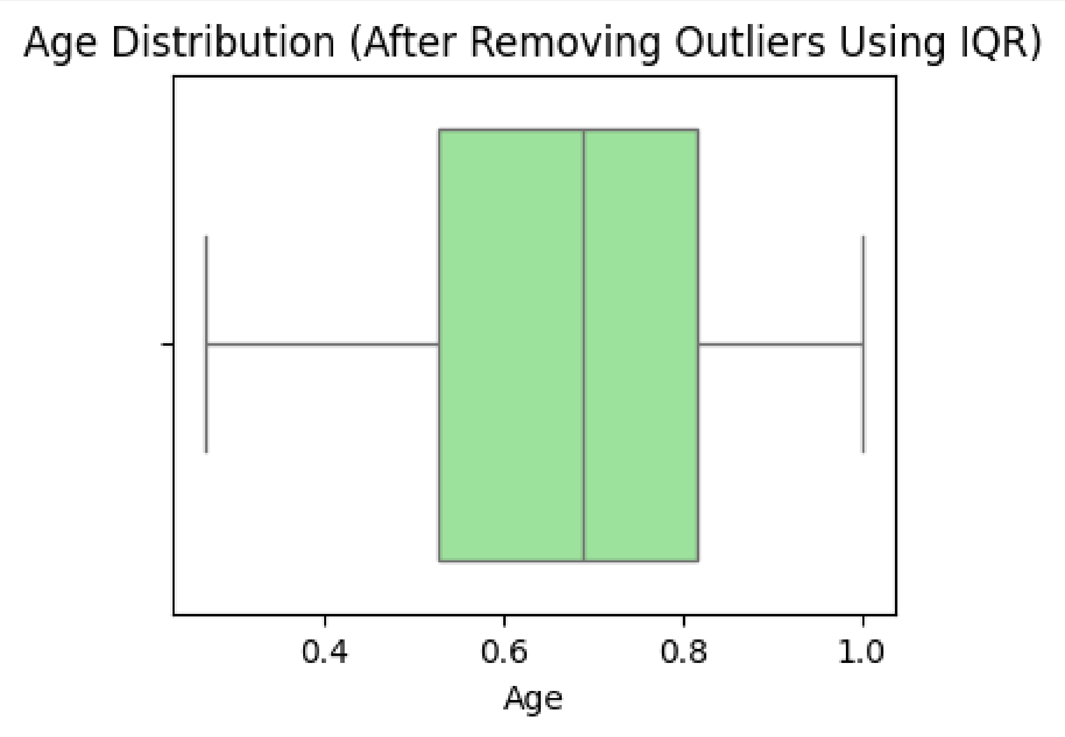

Weight Distribution (Before Removing Outliers)

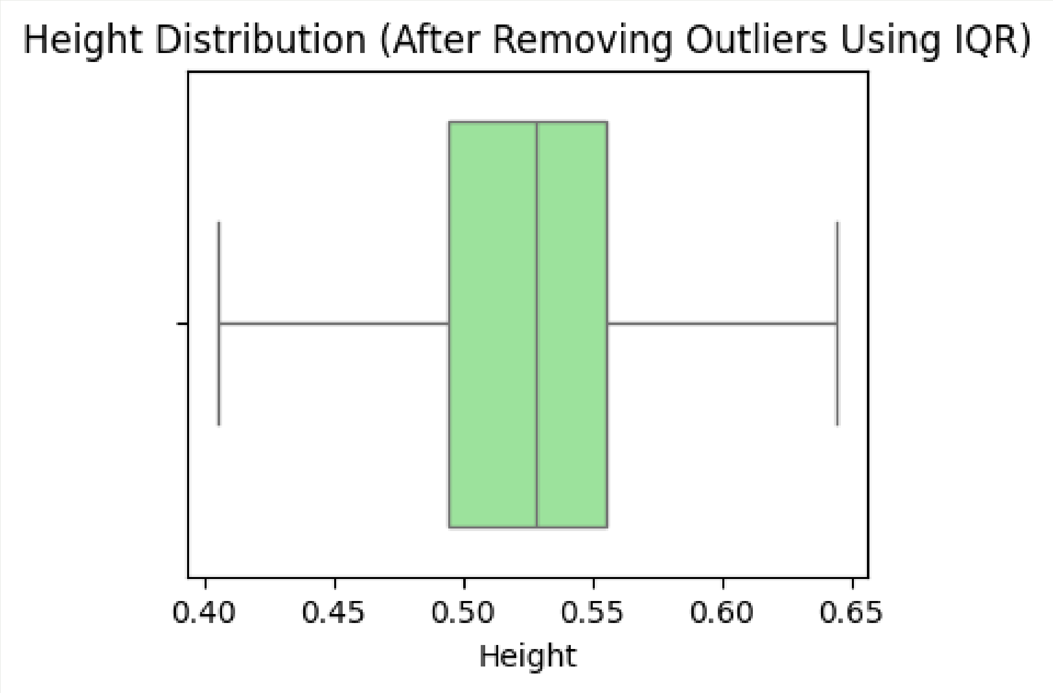

Age Distribution (Before Removing Outliers)


Height Distribution (Before Removing Outliers)

After Handling Outliers


Weight Distribution (After Removing Outliers Using IQR)


Age Distribution (After Removing Outliers Using IQR)


Height Distribution (After Removing Outliers Using IQR)

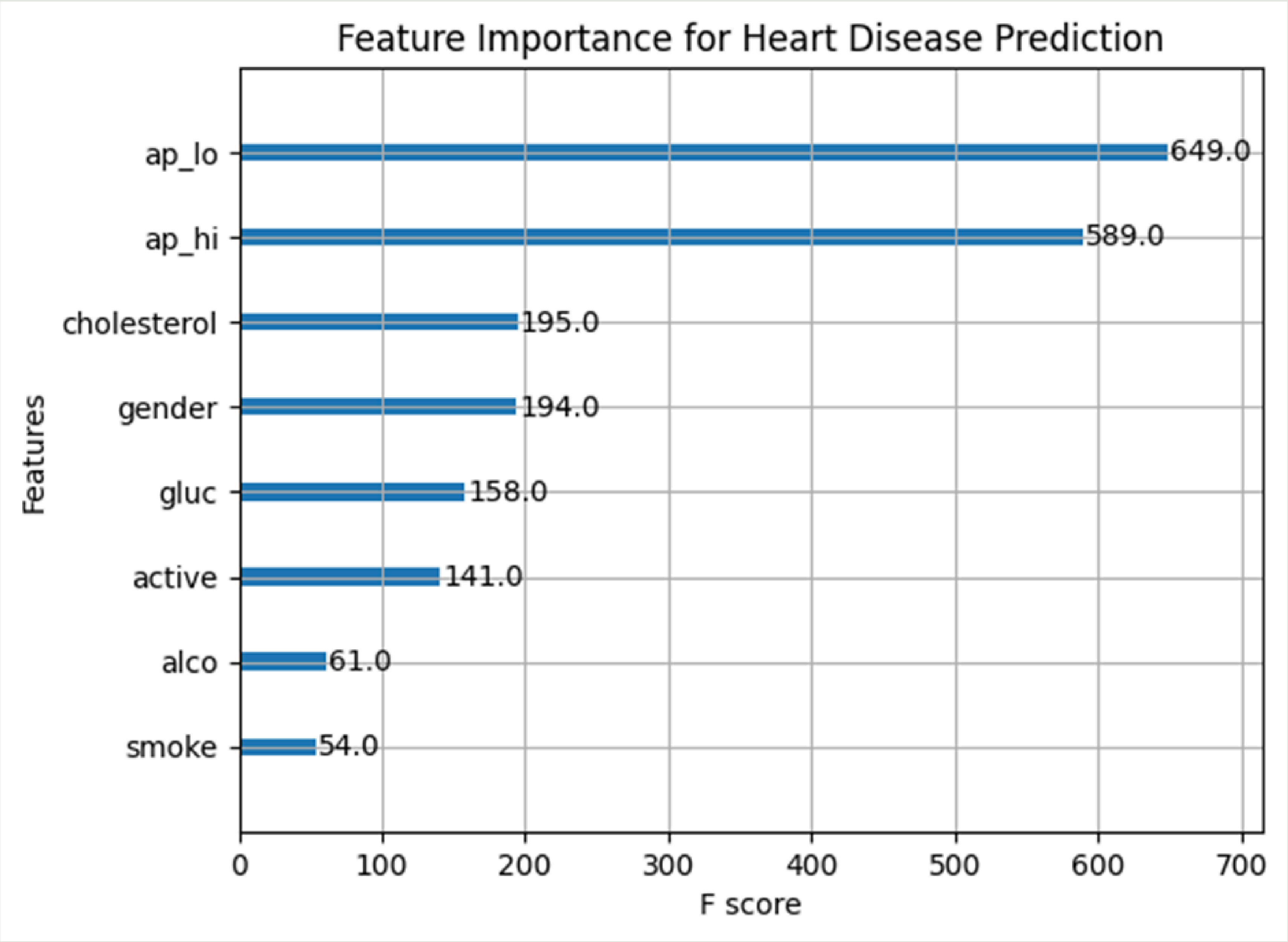
3.Feature Selection

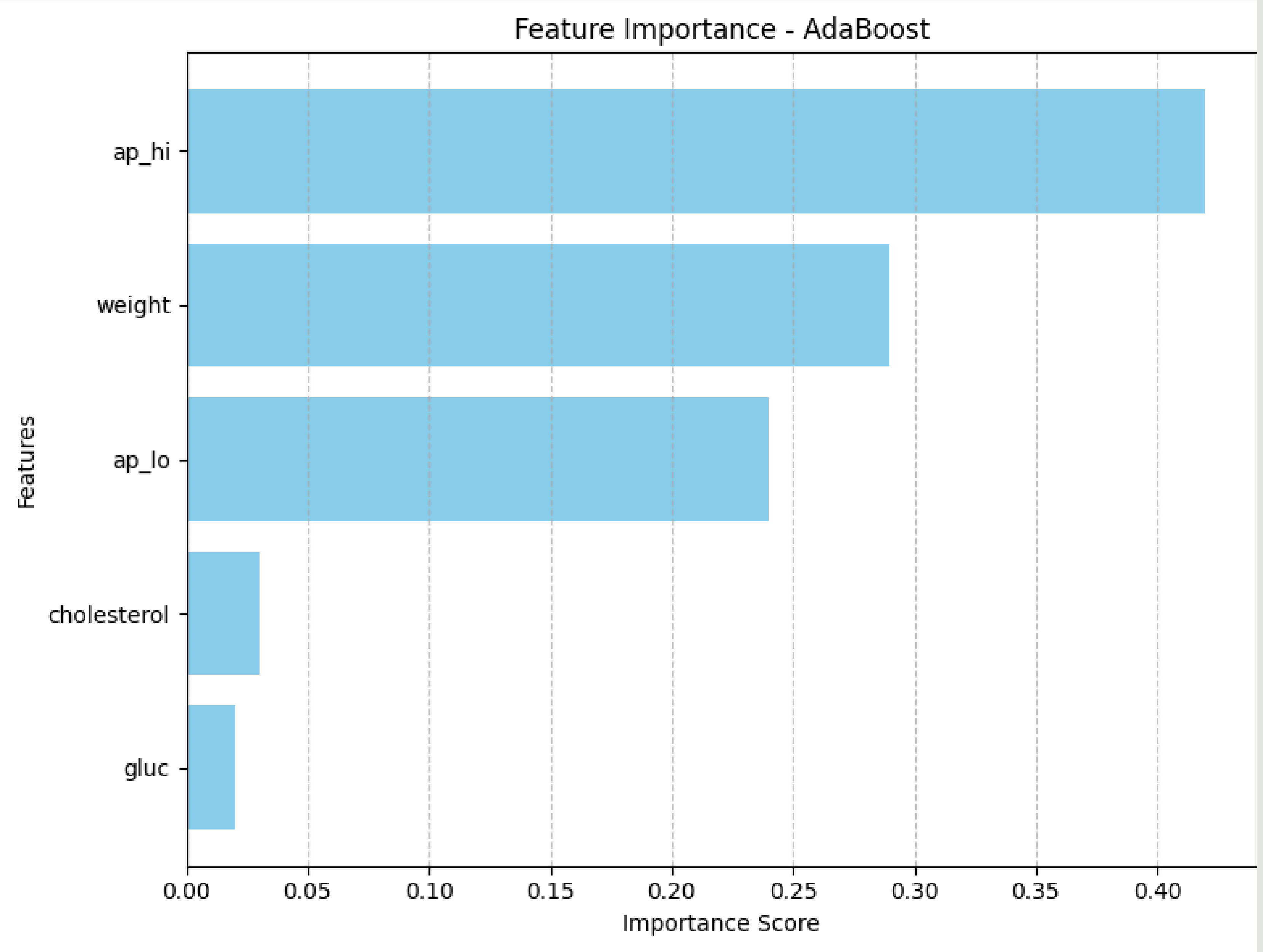
- Selected most relevant features for Ada Boost, XG boost and Gradient boosting algorithm
- Feature Importance Graph

- XG Boost-> systolic blood pressure, diastolic blood pressure, cholesterol, gender, glucose and activity

- ADA Boost-> systolic blood pressure, diastolic blood pressure, cholesterol, glucose and weight

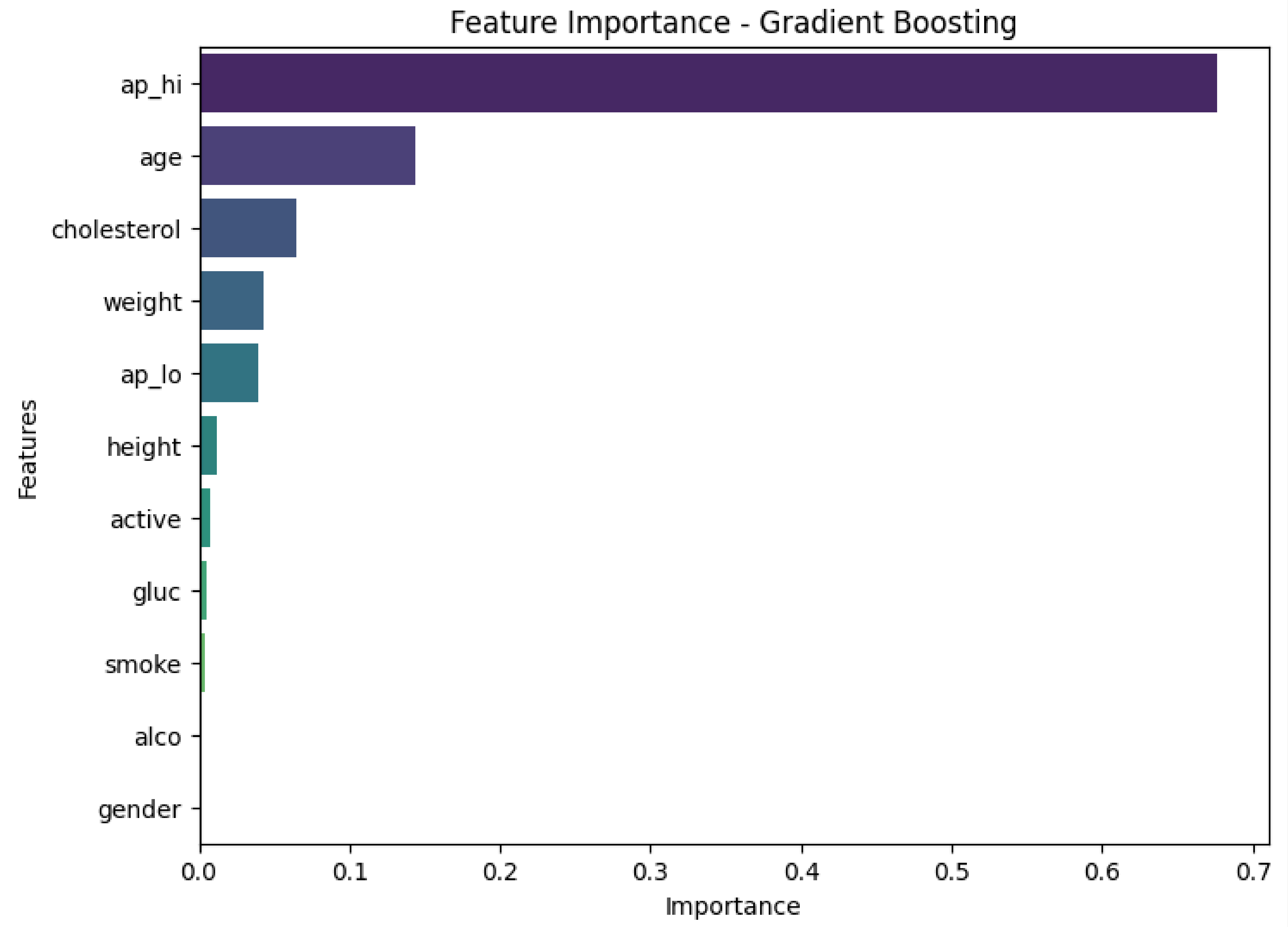
- Gradient Boosting-> systolic blood pressure, age, cholesterol, weight, diastolic blood pressure, height and activity





Important Features in XG Boost Algorithm

Important Features in Ada Boost Algorithm



Important Features in Gradient Boosting Algorithm

4. Model Building

- We employed AdaBoost ,XGBoost and Gradient Boosting ,combining multiple weak learners to predict heart disease.
- Both models improve accuracy by iteratively adjusting the weights of misclassified data points, focusing on difficult cases.

4.1 AdaBoost

- AdaBoost combines weak classifiers, focusing on misclassified instances by adjusting their weights iteratively.

$$\hat{y} = \sum_{i=1}^{N} \alpha_i h_i(x)$$

- The formula predicts $\hat{y}$ as the weighted sum of hi(x), where αi reflects each classifier's accuracy.

- AdaBoost’s final prediction is a weighted sum of the predictions from all weak classifiers, with weights based on each classifier's performance.

4.2 XGBoost

- XGBoost is an optimized version of gradient boosting that builds trees sequentially to minimize errors from previous ones.

- It incorporates regularization techniques to prevent overfitting and efficiently handles missing data.

$$\hat{y} = \sum_{i=1}^{N} \alpha_i h_i(x)$$

- The formula ŷ is the predicted value, N is the number of trees, hi(x) is the prediction of the i-th tree, and αi is the weight assigned to each tree.

- Its advanced optimizations result in faster training and superior predictive performance.

4.3 Gradient Boosting

- Gradient Boosting is an ensemble method that builds models sequentially, correcting errors of previous ones by minimizing a loss function to create a strong predictive model.
- Gradient Boosting uses hyperparameters like learning rate, estimators, and tree depth to manage bias and variance.
- It prevents overfitting through techniques like early stopping and subsampling.

$$\hat{y} = \sum_{i=1}^{N} \alpha_i h_i(x)$$

- The formula ŷ is the predicted value, N is the number of trees, hi(x) is the i-th weak learner and αi is is the learning rate controlling each learner's contribution.

5. Parameter Comparision

1.The dataset for heart disease prediction is divided into input features (blood pressure, cholesterol, weight) and the target variable (heart disease).

2.The data is split into 70% training and 30% testing sets with balanced class distribution using the stratify=y parameter.

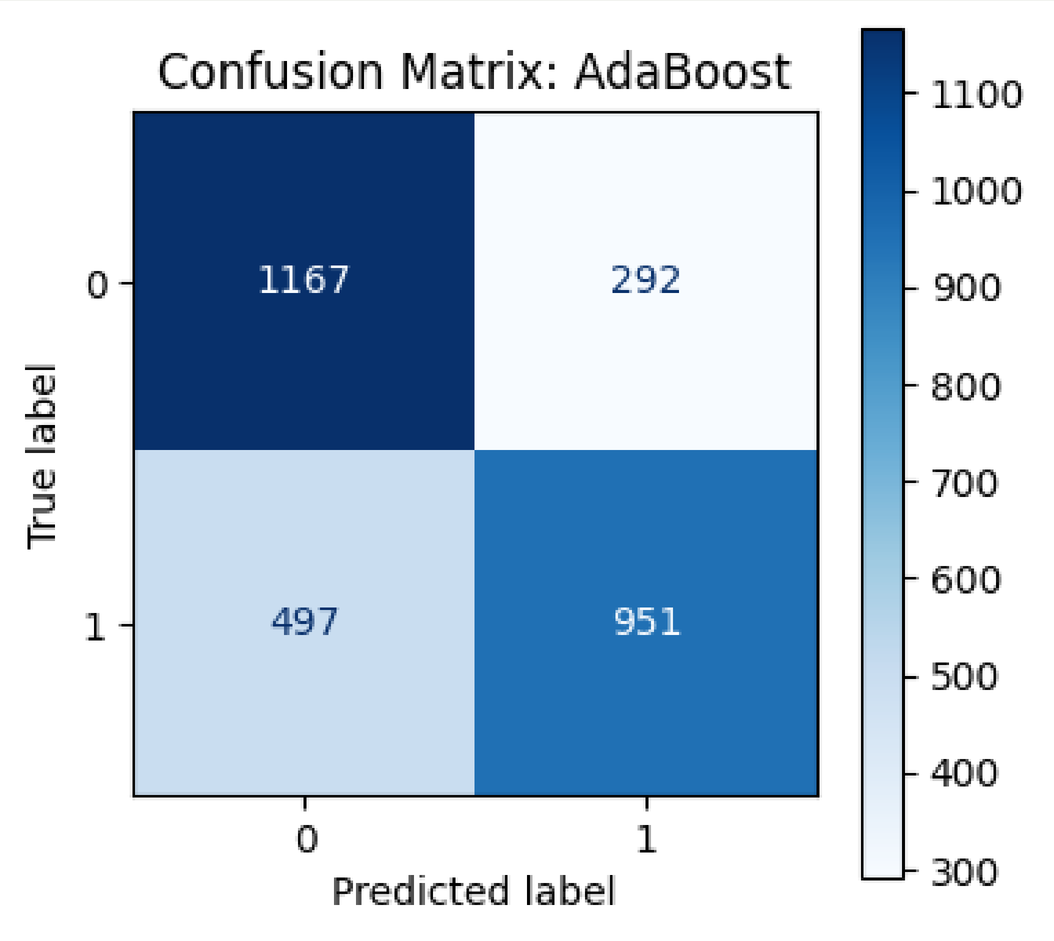
3.This allows AdaBoost and XGBoost models to learn from the training data and evaluate performance on unseen test data.

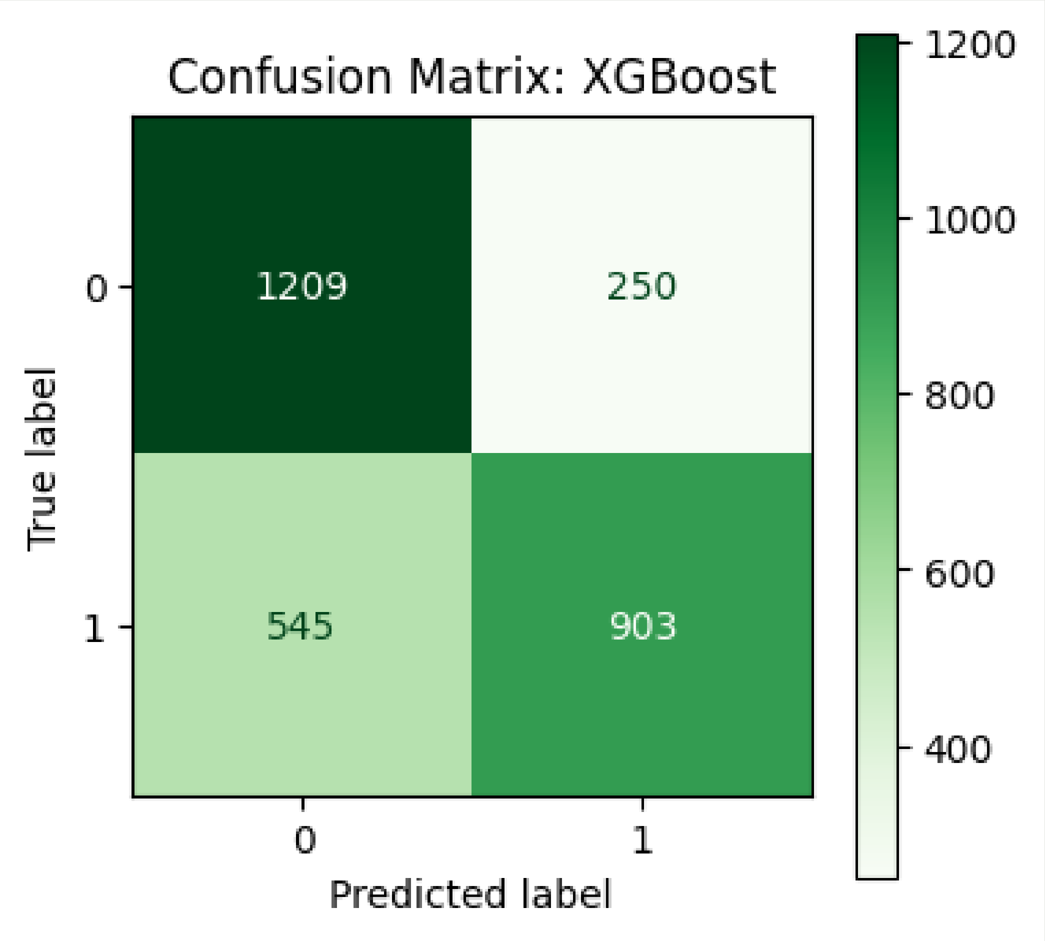
4.AdaBoost, Gradient Boosting, and XGBoost classifiers are initialized with 100 estimators, with XGBoost configured to a learning rate of 0.05 and maximum tree depth of 4. Gradient Boosting uses a learning rate of 0.1 and a maximum depth of 3 by default.

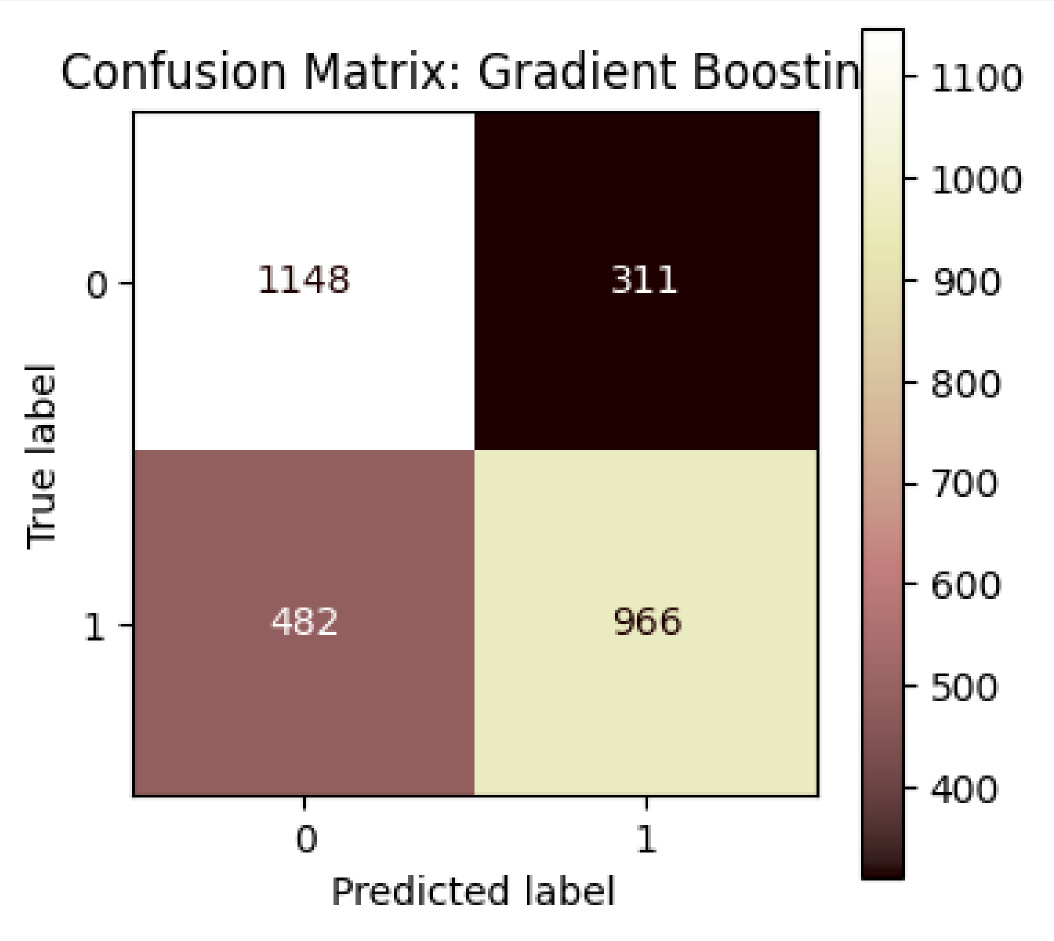
5.All the models are trained and evaluated on test data using the classification_report function, providing metrics like precision, recall, and F1-score for performance comparison.

- The confusion matrices for AdaBoost ,XGBoost and Gradient Boosting models are visualized to compare their performance by showing true positives, false positives, and other classification outcomes.

- The matrices are displayed in blue (AdaBoost) ,green (XGBoost) and pink(Gradient Boosting) , helping to visually assess and compare the models' accuracy in classifying the test data.







6.Hypertuning

1.Purpose: Optimize model performance by finding the best hyperparameter combinations.

2.Methods: Use Grid Search, Randomized Search, or advanced methods like Bayesian Optimization.

3.Tools: Libraries like GridSearchCV (sklearn) or xgb.cv (XGBoost).

4.Key Parameters

- AdaBoost: n_estimators, learning_rate, base_estimator.
- XGBoost: learning_rate, max_depth, n_estimators, subsample, etc.
- Gradient Boosting: n_estimators, learning_rate, max_depth, subsample, min_samples_split, min_samples_leaf, and max_features.

5.Process: Define search space, apply cross-validation, and select the best-performing hyperparameters for final training.

RESULT AND DISCUSSIONS

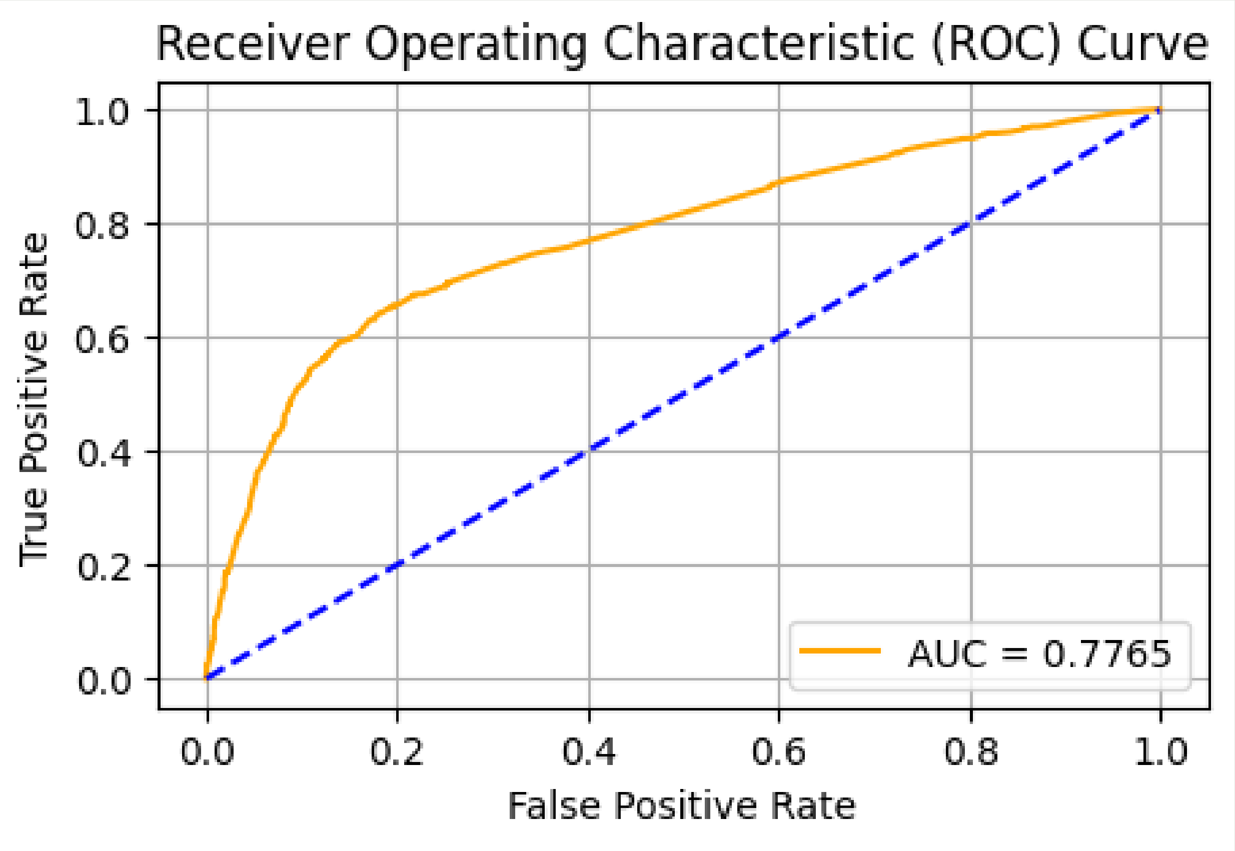
1.Performance Assessments

1.1Feature Selection

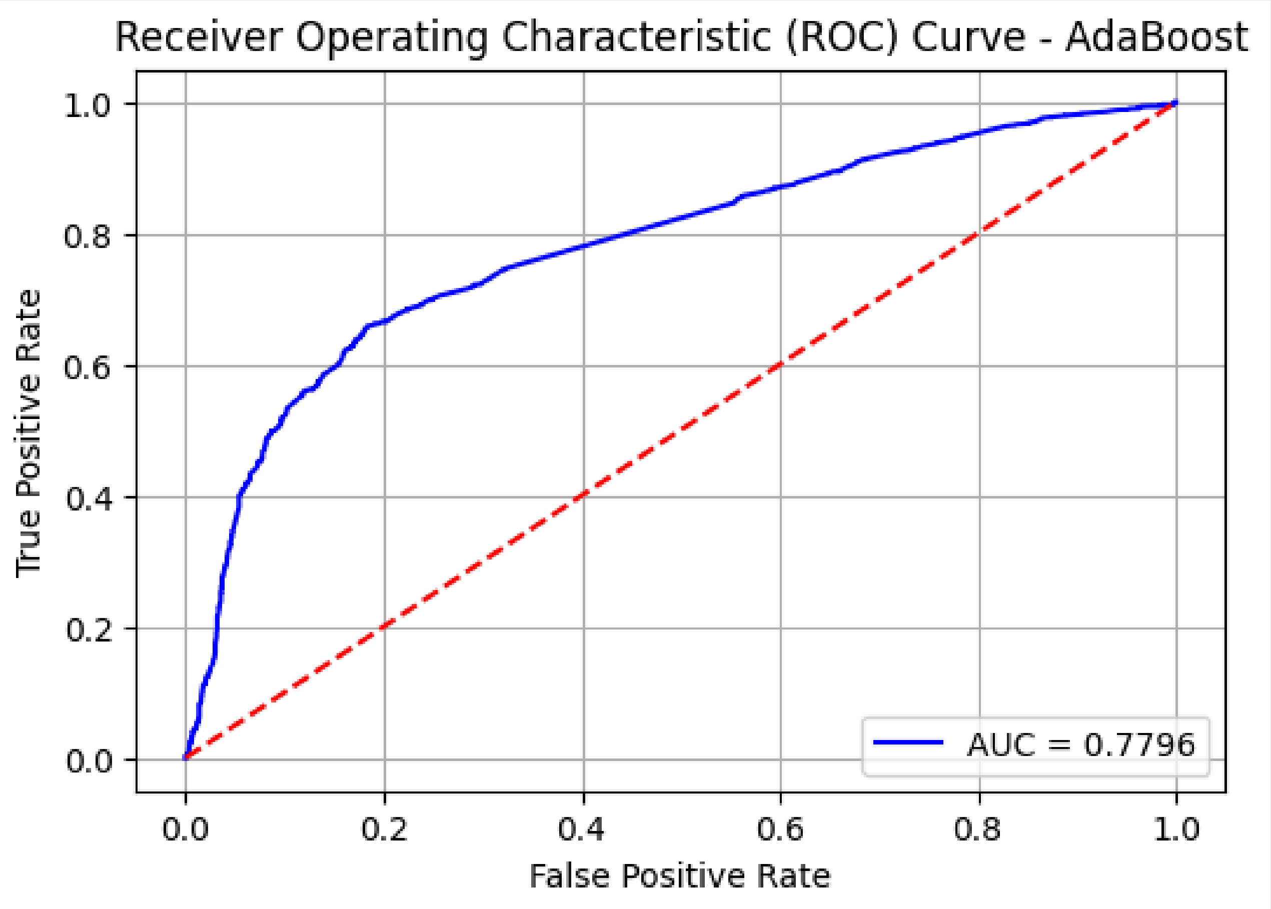
- AdaBoost, XGBoost, and Gradient Boosting collectively identify critical features such as blood pressure, cholesterol, and glucose. AdaBoost emphasizes body weight, XGBoost highlights physical activity, and Gradient Boosting focuses significantly on systolic blood pressure (ap_hi), age, and cholesterol, with moderate emphasis on weight and diastolic blood pressure (ap_lo).

- The combination of both models ensures a robust feature selection, balancing accuracy, interpretability, and generalization.

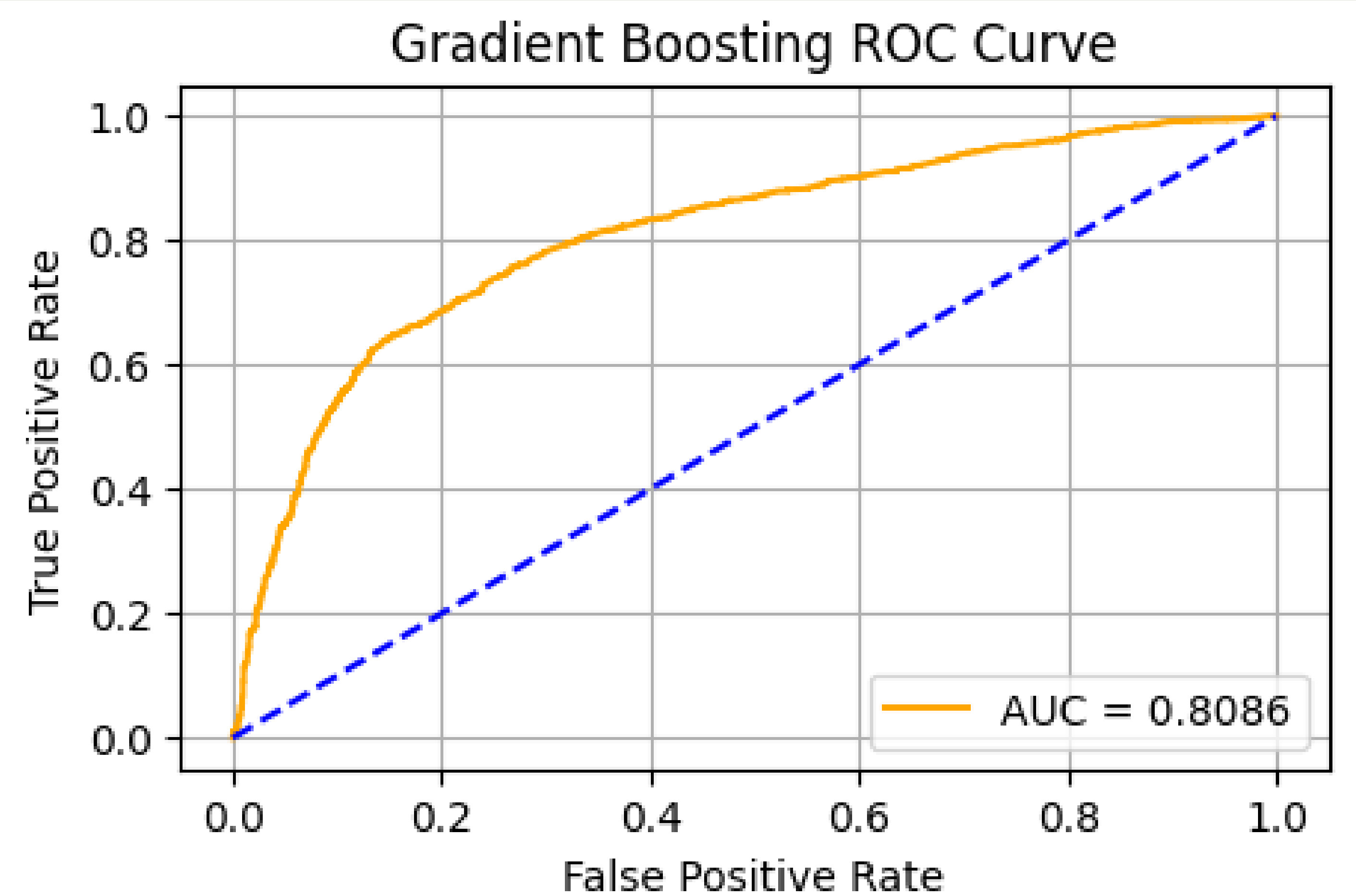
- The model's strong performance, measured by accuracy and ROC-AUC, demonstrates the effectiveness of the selected features, suggesting potential for real-world applications.



ROC Curve of XG Boost



ROC Curve of Ada Boost



ROC Curve of Gradient Boosting

1.2 Model Building using XG Boost and Ada Boost

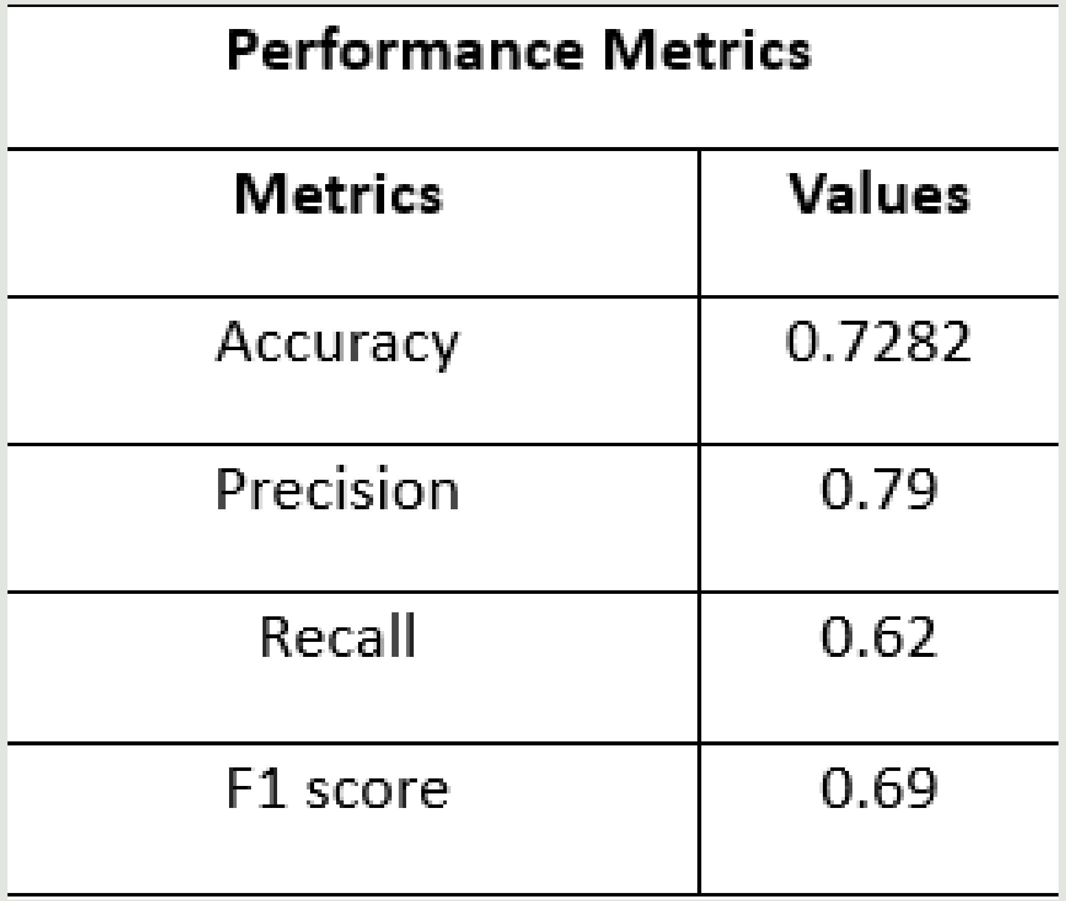
- AdaBoost enhances weak classifiers by focusing on misclassified samples, iteratively adjusting their weights to improve accuracy, making it effective for complex and imbalanced datasets.

- XGBoost builds decision trees sequentially, correcting errors from previous trees, and incorporates regularization and early stopping to reduce overfitting, making it highly effective for complex datasets with feature interactions.

- Gradient Boosting optimizes predictions by iteratively minimizing errors using gradient descent, refining accuracy with each step, but is more prone to overfitting without regularization

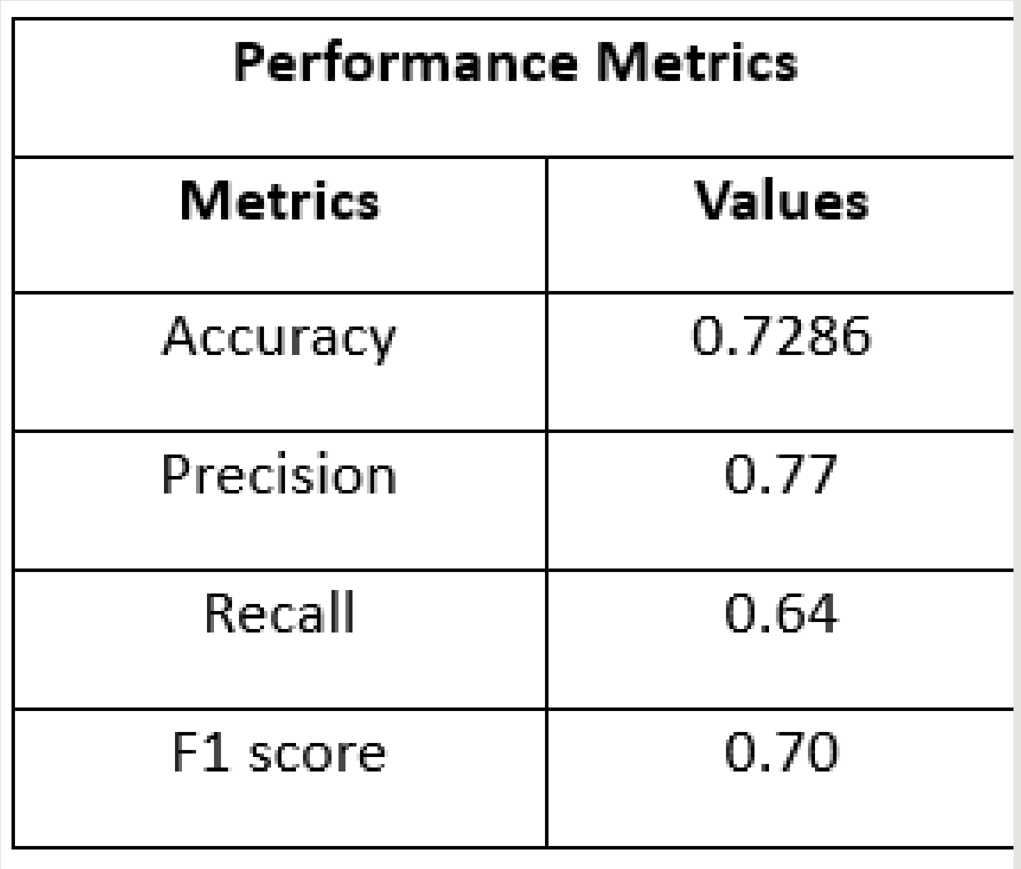
- All three models enhance predictive accuracy through iterative learning: AdaBoost focuses on misclassified samples, Gradient Boosting reduces errors using gradient descent, and XGBoost extends Gradient Boosting with regularization and early stopping for improved robustness in high-dimensional datasets.

XGBoost Model Performance Metrics Table



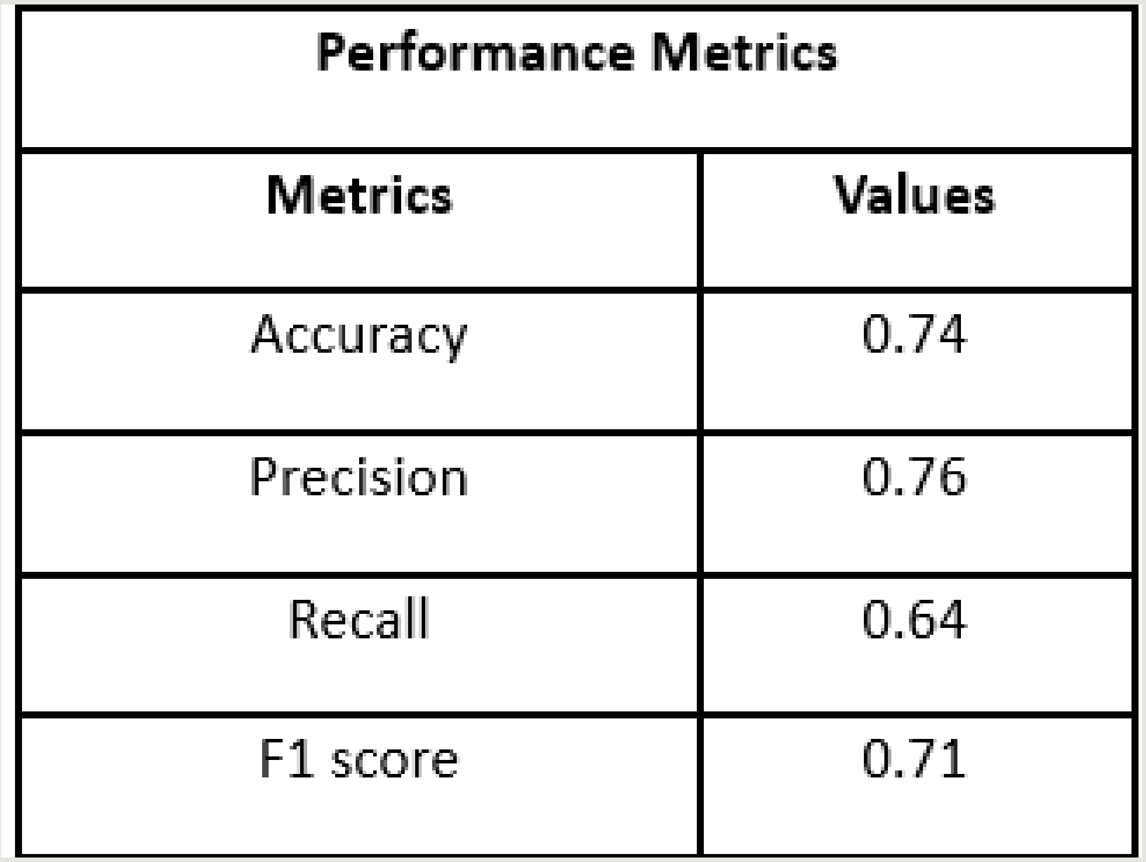
Performance Metrics	
Metrics	**Values**
Accuracy	0.7282
Precision	0.79
Recall	0.62
F1 score	0.69

AdaBoost Model Performance Metrics Table

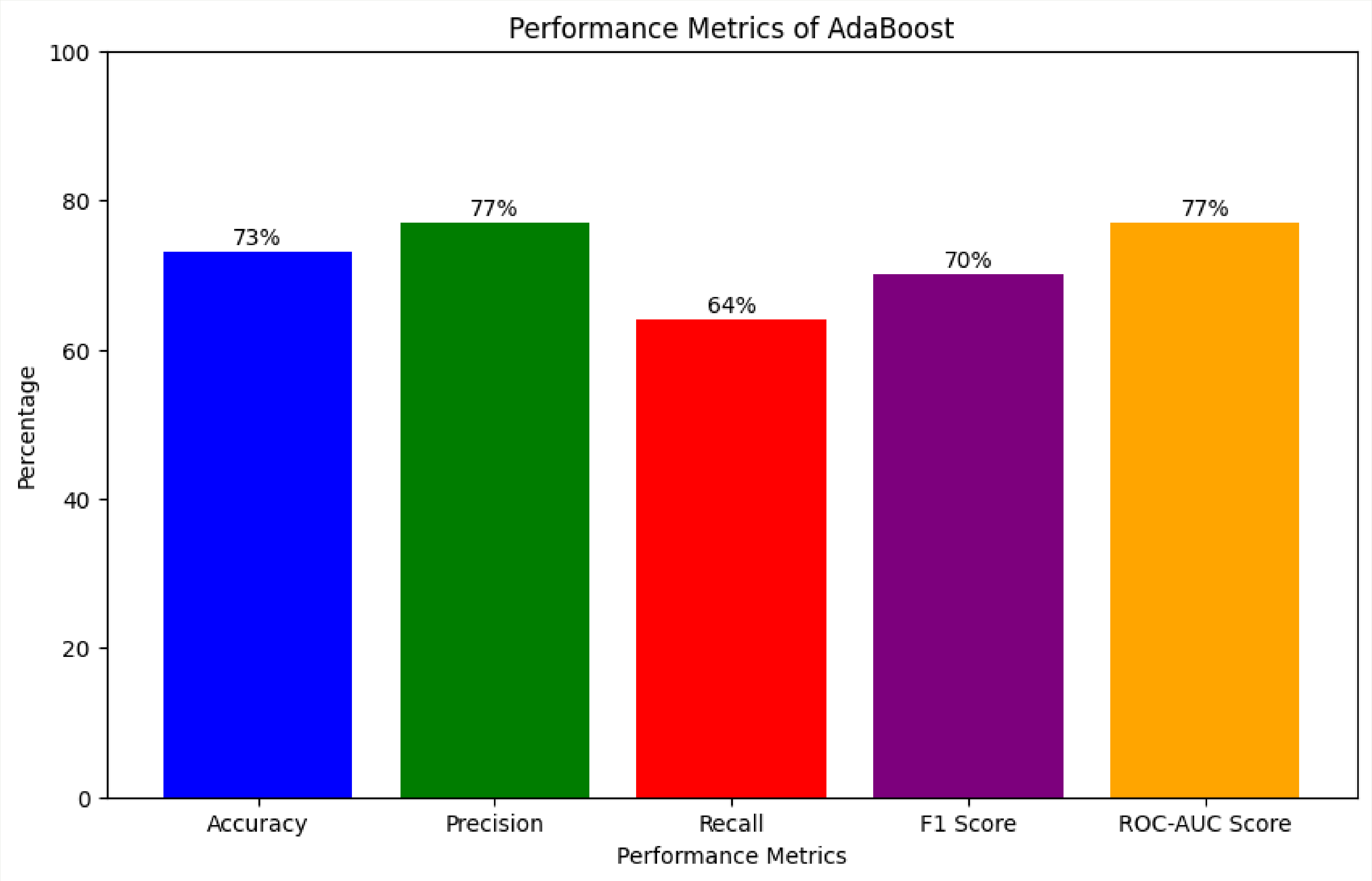


Performance Metrics	
Metrics	**Values**
Accuracy	0.7286
Precision	0.77
Recall	0.64
F1 score	0.70

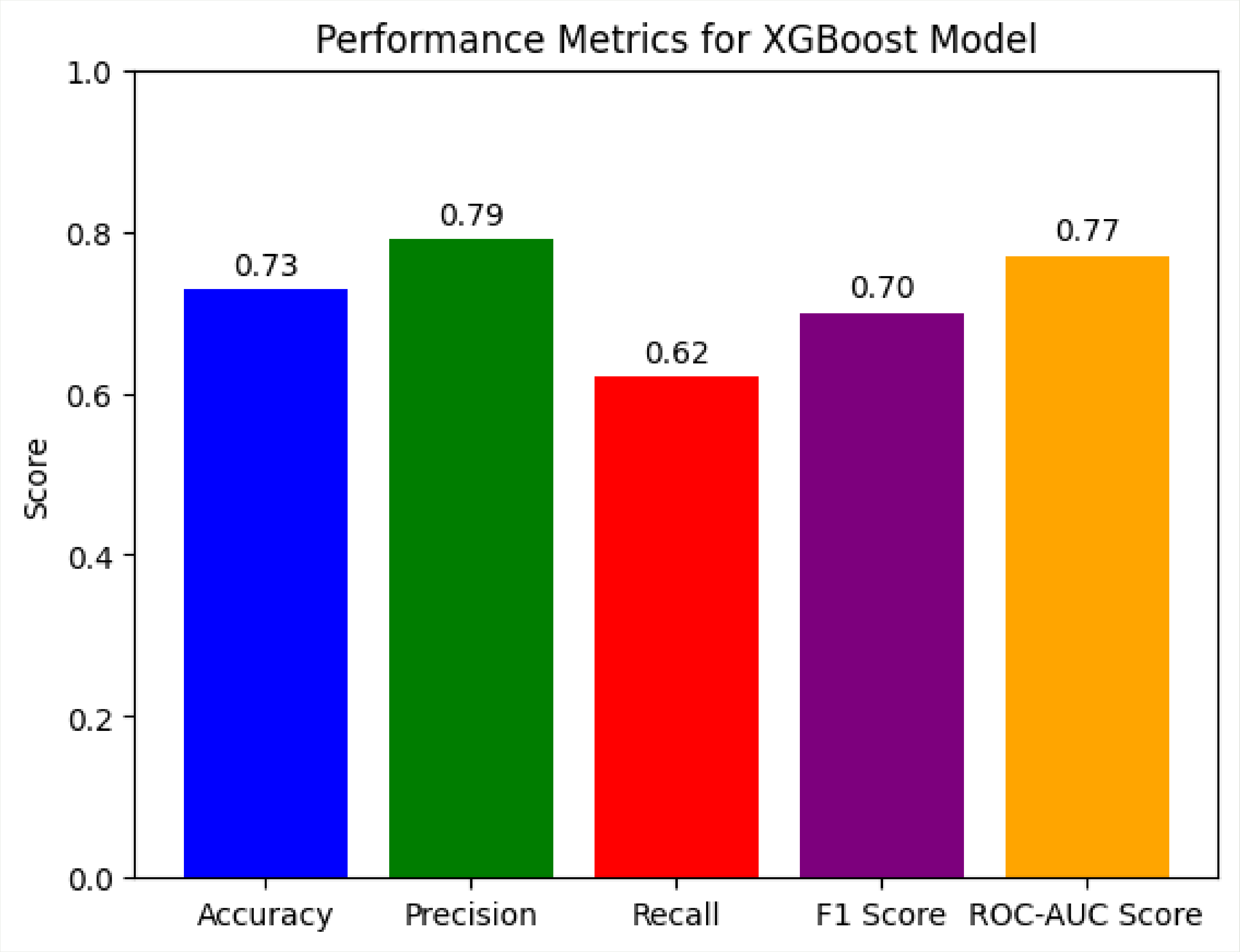
Gradient Boost Model Performance Metrics Table



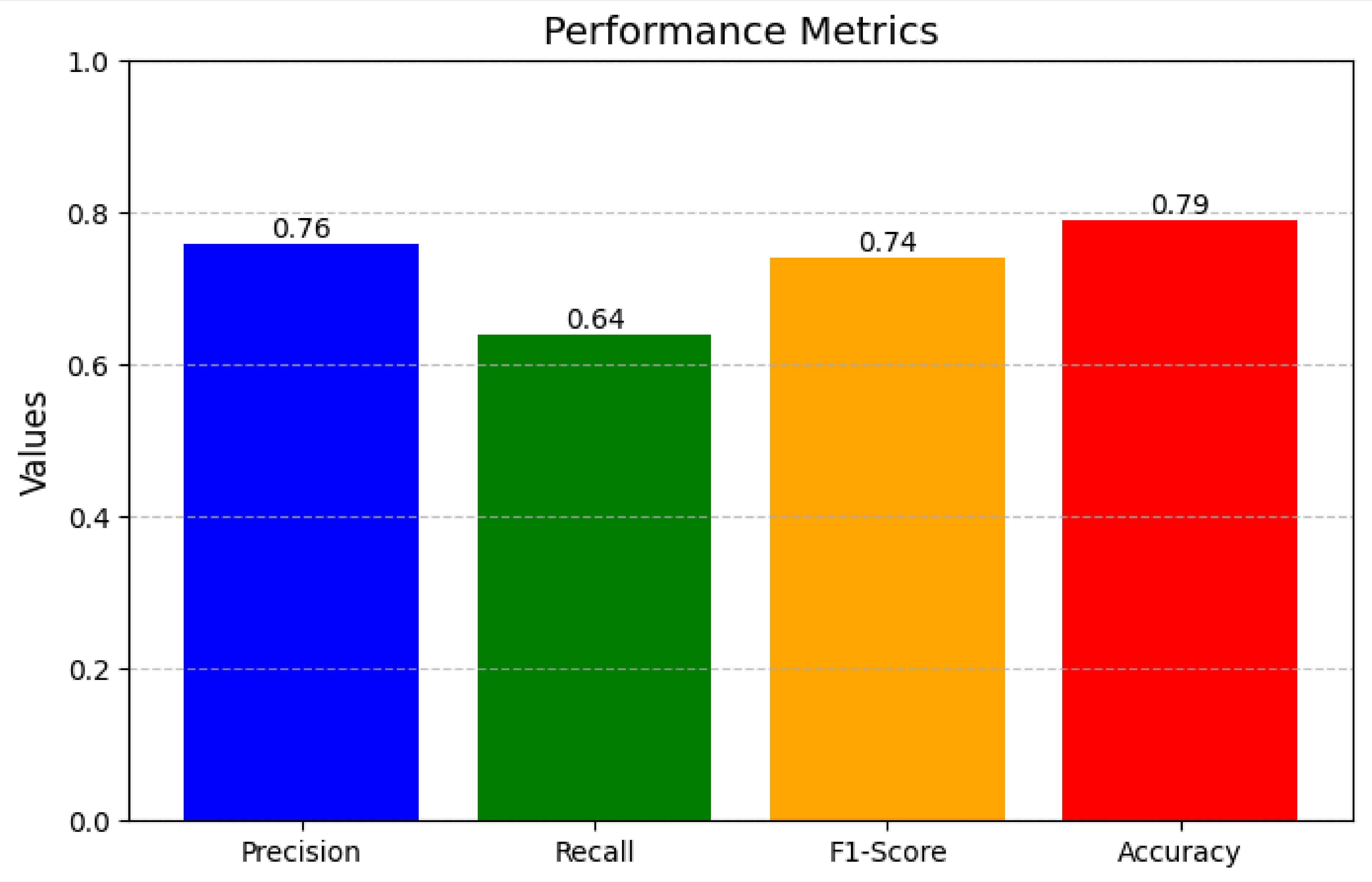
Performance Metrics	
Metrics	**Values**
Accuracy	0.74
Precision	0.76
Recall	0.64
F1 score	0.71



A bar graph for performance metrics for Ada Boost

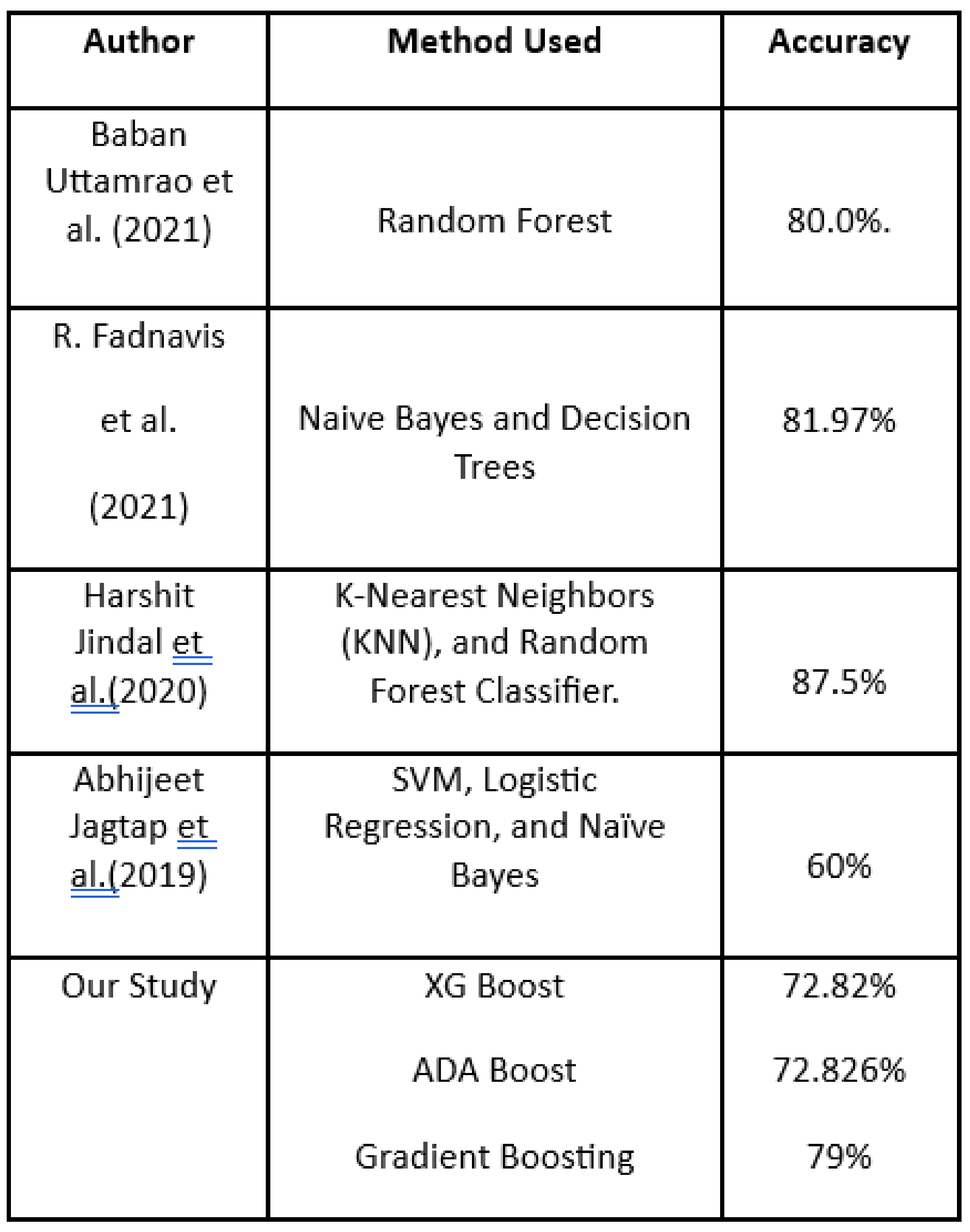


A bar graph for performance metrics for XG Boost



A bar graph for performance metrics for Gradient Boosting

2.Comparison of Proposed Method and Other methods on Heart Disease Prediction



Author	Method Used	Accuracy
Baban Uttamrao et al. (2021)	Random Forest	80.0%.
R. Fadnavis et al. (2021)	Naive Bayes and Decision Trees	81.97%
Harshit Jindal et al.(2020)	K-Nearest Neighbors (KNN), and Random Forest Classifier.	87.5%
Abhijeet Jagtap et al.(2019)	SVM, Logistic Regression, and Naïve Bayes	60%
Our Study	XG Boost ADA Boost Gradient Boosting	72.82% 72.826% 79%

- Gradient Boosting Performance: Gradient Boosting demonstrates solid results with 74% accuracy, 76% precision, 67% recall, and an F1 score of 0.71, alongside an impressive ROC-AUC score of 0.8086, highlighting its strong classification ability in heart disease prediction.

- XGBoost Performance: XGBoost outperforms AdaBoost in precision (79%) and ROC-AUC score (77%), with an accuracy of 72.82% and recall of 62%.

- Key Metrics Comparison: Gradient Boosting excels in ROC-AUC with a score of 0.8086, outperforming XGBoost in this metric, while also achieving a strong balance between precision and recall. XGBoost, on the other hand, excels in precision and ROC-AUC, though its recall (62%) is lower compared to Gradient Boosting. AdaBoost performs well overall, with strong precision but slightly lags behind in recall compared to Gradient Boosting.

- Model Strengths: Gradient Boosting’s higher ROC-AUC score suggests better classification ability, while AdaBoost demonstrates reliable performance with a strong precision-recall balance.

SUMMARY

In Phase 2 of our heart disease prediction project, we plan to expand our research by exploring and implementing new algorithms to enhance the model's predictive accuracy and reliability. This phase will focus on testing advanced machine learning techniques and optimizing the current models to ensure they perform effectively across diverse datasets. Additionally, we will develop a user-friendly frontend interface where users can input their health parameters, such as age, blood pressure, cholesterol levels, and more. This interface will be providing instant predictions on the likelihood of heart disease. This combination of algorithmic innovation and user-centered design will make our system both accessible and impactful.

INDIVIDUAL CONTRIBUTIONS

ARCHITA GUPTA 21BCE10225

I took responsibility for sourcing the dataset from Kaggle, which played a crucial role in the project. After preprocessing the data, I conducted feature selection and model training using the XGBoost algorithm. I identified key features such as Cholesterol, Systolic Blood Pressure, diastolic blood pressure, active and gender which significantly contributed to the model’s performance. Following the training with 30%dataset used for testing and 70% used for training, I achieved an accuracy of 72.82% and an AUC of 0.7765, highlighting the model's strong predictive capability. Additionally, I studied a 2024 paperheart disease prediction which provided valuable insights for refining my approach.

ABHINAV SHRIVASTAVA 21BCE10708

I identified the base research paper from 2024 on the XGBoost algorithm for the heart disease prediction project and handled the entire data preparation and analysis process. After sourcing the dataset from Kaggle, I performed exploratory data analysis (EDA) to understand feature distributions and checked for missing values using a heat map, confirming none were present. I used box plots for visualizing outlier removal, normalized key features like age, height, weight, systolic blood pressure and diastolic blood pressure MinMaxScaler, removed outliers with IQR technique, and filtered the dataset to ensure it was ready for training. This thorough preprocessing laid a strong foundation for the XGBoost model.

SONALI RAGHUWANSHI 21BCE10406

I played a crucial role in optimizing the AdaBoost algorithm using Python libraries to enhance its effectiveness for heart disease prediction. My focus was on meticulously tuning key hyperparameters, including learning rate, maximum depth, and gamma, which led to significant improvements in the model’s performance. I applied systematic techniques such as cross-validation to fine-tune these parameters effectively.

For the prediction model, I focused on five important features: systolic blood pressure (ap_hi), diastolic blood pressure (ap_lo), glucose (gluc), cholesterol, and weight. These features were crucial in improving the accuracy of the model.

These efforts culminated in achieving an impressive accuracy of 0.7286, highlighting the model’s reliability and generalizability. The model's performance was further validated with an outstanding ROC AUC score of 0.77, which was visualized using ROC curves, showcasing its exceptional ability to distinguish between patients with and without heart disease.

PRIYANSHI YADAV 21BCE10439

My contribution to this project involves detailed comparison of XGBoost and AdaBoost, identifying their strengths, weaknesses, and suitability for heart disease prediction based on performance metrics and interpretability. Through this comparative analysis, I aim to contribute valuable insights into the effectiveness of XGBoost and AdaBoost for heart disease prediction, informing the selection of appropriate machine learning algorithms for clinical applications and advancing the field of cardiovascular disease research. In additon to this I also helped in implementation of adaboost algorithm.

TINA CHELWANI 21BCE10669

I focused on the evaluation and implementation of the AdaBoost algorithm for heart disease prediction. I coded the AdaBoost algorithm and conducted extensive research to gain insights from academic papers, enhancing my understanding of the model's behaviour. My work involved analyzing performance metrics like accuracy, precision, recall, and F1-score, where AdaBoost achieved an accuracy of 69%. My classification analysis highlighted how AdaBoost performed across precision and recall metrics for different classes. Lastly, I ensured that the report included comprehensive references to existing literature, showcasing the value of machine learning in healthcare, especially for heart disease prediction.

REFERENCES:

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data

https://www.sciencedirect.com/science/article/pii/S1746809421010533?casa_token=FGzE-UUrEJkAAAAA:RFB5rjq8y5ruzjeB2Z_YrLd10zMrunqBSiGrVUozIajleWLnY9jxJdF2GEpi1A8j-w9CyHI

https://www.jeeemi.org/index.php/jeeemi/article/view/440

https://www.sciencedirect.com/science/article/pii/S1319157820304936

https://www.mdpi.com/2078-2489/15/7/394

https://ieeexplore.ieee.org/abstract/document/10620208

https://www.sciencedirect.com/science/article/pii/S235291481830217X

Thank You