ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

Predicting Heart Disease through Machine Learning Methods

Latthika S
Post Graduate Program
Vellore Institute of Technology, Bangalore,
Karnataka, India

Abstract:- Heart diseases including heart attacks, cause about 31% of global deaths, remaining a significant health threat despite preventability. Limited tech advancements and awareness, especially in developing nations, amplify this challenge. Machine learning offers promise in tackling this issue, with studies advocating ensemble methods for accurate predictive models. These models analyze extensive medical data to efficiently predict heart diseases, undergoing stages like data exploration, feature selection, model implementation, and comparative analysis. A model using Logistic Regression, Naive Bayes, and Random Forest initially identified top-performing models, later refined to CatBoost, RandomForest, and XGBoost through cross-validation and tuning. A hybrid model, combining Logistic Regression, CatBoost, and RandomForest, achieved a 97% accuracy, showcasing improved precision, recall, F1 score, and ROC AUC. This underscores machine learning's potential in enhancing predictive accuracy and refining strategies to combat heart diseases effectively.

Keywords:- *Logistic Regression(LR), K-Nearest Neighbors(KNN), RandomForest(RF), CatBoost(CB), XSBoost (XSB), Stochastic Gradient Descent(SGD), Cross-Validation(CV), Support Vector Machine(SVM) Hyperparameter Tuning(HT) and Voting Classifier(VC).*

I. INTRODUCTION

In today's fast-paced world, the emphasis on self-care often gets overshadowed by the demands of daily life, leading to heightened stress levels and neglect of one's health. Even with the progress that medicine has made, diseases like cancer, heart disease, and tuberculosis still take a lot of lives each year. Globally, cardiovascular disease (CVD) is now the leading cause of death, accounting for about 31% of all deaths, according to the World Health Organisation (WHO). Over the span of 15 years, WHO reported an alarming 15.2 million deaths attributed to heart-related diseases, underscoring the persistent threat posed by these conditions. Notably, heart-related ailments inflicted a significant economic toll, amounting to around $237 billion in India alone between 2005 and 2015.

The heart, as a vital organ responsible for blood circulation, plays a crucial role in supplying oxygen and nutrients throughout the body. Any dysfunction in this essential organ severely impacts the functionality of other bodily organs, presenting a formidable challenge. Unhealthy dietary habits and the rapid pace of modern lifestyles contribute substantially to the heightened risk of heart-related diseases.

Leveraging machine learning and deep learning techniques to analyse diverse patient data within the medical field offers a promising avenue for assessing risks, identifying symptoms, and predicting heart-related diseases. Factors such as diabetes, smoking, excessive alcohol consumption, high cholesterol, high blood pressure, and obesity significantly elevate the risk of heart issues. Despite efforts to manage these factors, heart diseases can manifest regardless of gender or age.

The purpose of this project is to use machine learning and deep learning techniques to predict heart disease by doing a thorough examination of these risk variables. These kinds of predictive powers could transform healthcare and improve people's lives. Furthermore, the research delves into a range of heart disorders, such as Cardiomyopathy, Congenital Heart Disease, Heart Failure, and coronary artery disease, each with unique traits and implications for the cardiovascular system.

In the initial phase of the study, the dataset was loaded, and multiple machine learning algorithms were employed, including SGD, NB, RF, CB, XB, KNN, LR, and SVM. Performance metrics such as Precision, recall, accuracy, F1 Score, and ROC AUC were computed, identifying the top-performing models before hyperparameter adjustment.

Subsequently, cross-validation and hyperparameter tuning were performed, leading to the identification of another set of top-performing models with enhanced predictive capabilities. Most models exhibited noticeable improvements across various criteria following hyperparameter adjustment, particularly in precision, recall, accuracy, and F1 score.

Finally, a hybrid model combining LR, CB, and RF was developed using a voting classifier. This model demonstrated remarkable predictive performance, achieving high accuracy and impressive precision and recall scores. The balanced F1 score and outstanding ROC AUC further underscored the model's overall performance.

This comprehensive approach utilizing machine learning techniques highlights the potential to accurately predict heart disease, marking significant progress in early identification

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

and intervention against cardiovascular ailments. The integration of various algorithms and methodologies signifies the potential for impactful advancements in healthcare and enhanced patient outcomes.

II. LITERATURE SURVEY

In the paper "Heart disease identification from patients' social posts, machine learning solution on spark" by H. Ahmed, E.M.G. Younis, A. Hendawi, and A.A. Ali, Apache Spark and Apache Kafka are utilized alongside machine learning methods such as Decision Tree, Support Vector Machine, RF Classifier, and LR Classifier to create a real-time system for predicting heart disease from medical data streams. The methodology includes feature selection algorithms, machine learning algorithms, hyperparameter tuning, and cross-validation. However, limitations exist in terms of sample size, data quality, and generalizability to other populations [1].

S. Matin Malakouti's paper, "Heart disease classification based on ECG using machine learning models," explores the automated categorization of Electrocardiography (ECG) data using Gaussian NB, RF, LR, and Linear Discriminant Analysis. The study discusses the advantages and disadvantages of these methods, emphasizing the use of 10-fold cross-validation to reduce prediction variance and avoid biased assessment. However, the study's limitation lies in the challenges of accurately distinguishing between healthy and sick individuals using machine learning and deep learning methods [2].

Md Mamun Ali et al.'s paper, "Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison," investigates various machine learning classifiers for heart disease prediction. While the RF method achieved 100% accuracy, sensitivity, and specificity on a specific dataset, the study's reliance on a single dataset raises concerns about generalizability to other datasets [3].

L. Sharan Monica et al.'s paper, "Latest trends on heart disease prediction using machine learning and image fusion," aims to develop a program for reliable and instant disease diagnosis. The methodology involves exploratory data analysis, attribute selection, and the use of machine learning methods such as NB, decision trees, SVM, and artificial neural networks. Similar to previous studies, the reliance on a single dataset limits the generalizability of the findings [4].

Ivan Miguel Pires et al.'s paper, "Machine learning for the evaluation of the presence of heart disease," explores different machine learning techniques for detecting cardiac illness. Despite achieving high accuracy using Decision Tree and Support Vector Machine approaches, the paper lacks a detailed description of feature extraction, selection, and model training methods [5].

Jinny, S. V., & Mate, Y. V.'s paper, "Early prediction model for coronary heart disease using genetic algorithms, hyper-parameter optimization, and machine learning techniques," aims to identify heart diseases using machine learning methods and heart rate features. While the study uses advanced techniques such as genetic algorithms and hyper-parameter optimization, the absence of a thorough explanation of feature selection procedures is noted [6].

Katarya, R., & Meena, S. K.'s literature review paper explores the use of machine learning and deep learning techniques for heart disease analysis. The systematic review of existing literature aims to guide future research in the healthcare industry. However, the paper's reliance on secondary sources and lack of original research may limit its contributions [7].

Abeer Alsadoon's paper compares the accuracy of different machine learning models for heart disease prediction. While the study recommends specific models for classification, the lack of detail regarding feature selection is identified as a drawback [8].

Katarya, R., & Meena, S. K.'s paper delves into the application of machine learning for heart disease prediction, emphasizing the increasing prevalence of heart disease and the need for efficient data analysis in the medical sector. The study reviews various risk factors and employs algorithms such as LR, K-Nearest Neighbor, Support Vector Machine, Naïve Bayes, and Decision Trees for prediction and classification [9].

Finally, the paper by Naseri, A., Tax, D., van der Harst, P., Reinders, M., & van der Bilt explores the use of machine learning methods to detect atrial fibrillation and heart failure from wearable devices. While the study presents innovative methods for cardiovascular outcome prediction, data privacy concerns and limited sample size may impact the generalizability of the findings [11].

III. PROPOSED METHODOLOGY

Before implementing cross-validation and hyperparameter tuning, LR was the leading model, exhibiting commendable accuracy. However, following cross-validation and hyperparameter tuning, CB emerged as the top-performing model, showcasing superior accuracy. Throughout these processes, the RF algorithm consistently demonstrated strong performance both before and after tuning.

Given the robust performances of LR, CB, and RF individually, a hybrid model was crafted using a voting classifier, leveraging the strengths of these three algorithms.

The code demonstrates the creation and evaluation of a VCensemble, amalgamating three distinct algorithms: RFClassifier, CBClassifier, and LogisticRegression. The 'voting' parameter is set to 'soft', indicating that the final prediction is determined by the weighted average probability of each classifier.

After training the VC on the given dataset, it's evaluated using various metrics. The achieved performance metrics are impressive: an accuracy of 97%, with a precision of 99%, recall of 95%, and an F1 score of 97%. Additionally, the Receiver Operating Characteristic (ROC) curve showcases an

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

Area Under the Curve (AUC) of 99.85%, signifying exceptional model discrimination ability across different thresholds.

The 'soft' voting method considers the probabilities predicted by each model, weighing them and making predictions based on these weighted probabilities. This tends to offer more nuanced decisions by taking into account the confidence levels of individual models. In contrast, 'hard' voting considers only the class labels predicted by each model and selects the majority class as the final prediction. The 'soft' approach can often lead to improved performance when models are well-calibrated and have reliable probability estimates.

IV. SYSTEM DESIGN & IMPLEMENTATION

To create a reliable predictive model for heart disease, the suggested methodology includes sophisticated machine learning algorithms, deliberate data preprocessing, and model validation. The steps in the methodology are as follows:

A. Data Collection and Preprocessing

➢ *Data Sourcing:*

Obtaining a comprehensive dataset involves sourcing diverse patient information from various sources, including hospitals, research databases, or healthcare institutions. This dataset should encompass:

- Demographics: Age, gender, ethnicity, etc.
- Medical History: Pre-existing conditions (diabetes, hypertension), medication history.
- Vital Signs: Blood pressure, heart rate, BMI.
- Lab Results: Cholesterol levels, blood glucose, etc.

➢ *Data Cleaning:*

Cleaning the dataset is essential to ensure data quality and consistency:

- Handling Missing Values: Address missing values through imputation (mean, median, mode) or deletion based on the extent of missingness.
- Outlier Treatment: Identify and handle outliers using statistical methods (e.g., Z-score, IQR) to prevent skewing of results.
- Normalization/Standardization: To improve model performance and convergence, normalize or standardize numerical features to bring them to a common scale.

➢ *Data Split:*

Partitioning the dataset into test, validation, and training sets is essential for building and assessing models:

- Training set: The predictive model is trained using the training set.
- Validation Set: Used to evaluate model performance during training and adjust hyperparameters.
- Test Set: Used to assess the performance of the finished model on unobserved data.

B. Exploratory Data Analysis (EDA):

➢ *Descriptive Analysis:*

Understand the dataset's characteristics, distributions, and statistical summaries:

- Central Tendency: Mean, median, mode of features.
- Dispersion: Standard deviation, range, interquartile range (IQR).
- Correlation Analysis: Identify relationships between variables (e.g., correlation matrix) to understand feature importance.

➢ *Visualization:*

Utilize visual tools to gain deeper insights and identify potential patterns related to heart disease:

- Histograms: Display distributions of numerical variables.
- Heatmaps: Visualize correlations between features.
- Scatter Plots: Explore relationships between two numerical variables.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

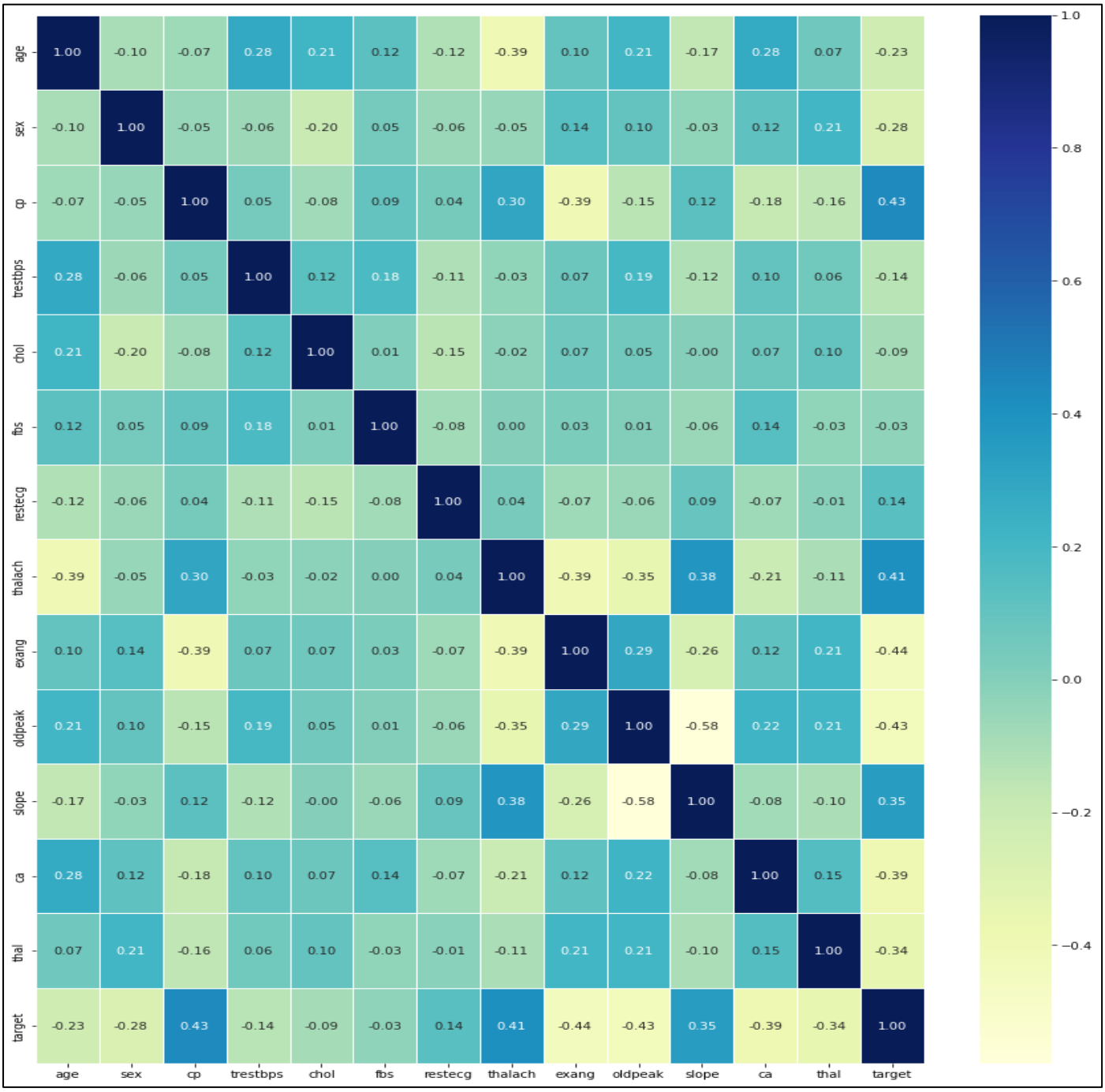

Fig 1 Correlation Matrix with Heat Map

C. Feature Selection and Engineering:

➢ *The Significance of Features:*

Utilise methods (such as correlation matrices and statistical tests) to identify pertinent features linked to heart disease.

➢ *Feature Engineering:*

To improve the model's capacity for prediction, add new features or modify current ones.

D. Algorithm Selection:

Explore diverse machine learning algorithms suited for heart disease classification tasks.

➢ *Logistic Regression (LR):*

Regression and classification tasks are two popular uses for supervised machine learning algorithms such as LR. To forecast the categories into which categorical data will be split, LR uses probability. It blends input numbers linearly for outcome prediction by using coefficient values and a sigmoid or logistic function. The sigmoid function is employed to estimate maximum likelihood using the most probable evidence, resulting in an event's probability ranging from 0 to 1. Classification problems occur when a decision threshold is used. Binary (0 or 1), Multinomial (three or more categories without a hierarchy), and Ordinal (three or more categories with a hierarchy) are several forms of logistic regression (LR). Despite its simplicity and good predictive power, LR remains

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

prone to categorization issues. The LR formula for establishing the probability that input X belongs in class 1 can be expressed as:

$$P(X) = \frac{\exp(\beta_0 + \beta_1 X)}{1 + \exp(\beta_0 + \beta_1 X)}$$

Here β_0 is bias and β_1 is the weight that is multiplied by input X .

➢ *K-Nearest Neighbors (KNN):*

A flexible supervised machine learning technique for regression and classification applications is the KNN algorithm. It functions according to the similarity principle, which states that the majority class of a sample's KNN in the feature space determines its class. To ascertain the KNN for a new data point, KNN calculates the Euclidean distance between the new point and each point in the training set. A majority vote among these neighbours then determines the class of the new point. In regression tasks, KNN uses a weighted average or an average of the target values of its KNN to forecast the value of the incoming data point. In an n-dimensional feature space. The Euclidean distance between two points is determined using the subsequent formula:

Euclidean distance = $\sqrt{(\sum i=1N (pi-qi)^2)}$

This distance metric measures the proximity between data points, forming the foundation for KNN's decision-making process.

➢ *Random Forest (RF):*

RF is a popular ensemble learning method that excels at both classification and regression due to its strong resilience and accuracy. It consists of several decision trees that were trained using a random subset of features and a bootstrapped sample of the dataset. When making a prediction, the ensemble averages the predictions made by each individual tree to get the final result, which is either the mode or the mean for classification or regression. This approach promotes diversity among the trees, mitigating overfitting and enhancing generalization by reducing sensitivity to noise and variance in the data. RF's capability to handle large datasets and capture complex feature relationships makes it popular across various machine learning applications, providing reliable and robust predictions.

➢ *CatBoost (CB):*

CB is a powerful gradient boosting algorithm designed for handling categorical variables in machine learning tasks. Its name, "CB," derives from its ability to effectively handle categorical features without the need for extensive pre-processing, reducing the risk of overfitting. Developed by Yandex, CB employs an innovative method to handle categorical data, utilizing a variant of gradient boosting that integrates an advanced algorithm for handling categorical variables. It employs a symmetric tree structure and utilizes novel strategies like ordered boosting and oblivious trees to optimize model performance while minimizing overfitting. CB also incorporates robust handling of missing data and provides excellent accuracy by default, requiring minimal hyperparameter tuning, making it an efficient and user-friendly choice for predictive modeling tasks, especially in scenarios with complex datasets containing categorical features.

➢ *XGBoost (XB):*

XB, an abbreviation for extreme Gradient Boosting, stands out as a highly efficient and accurate ensemble learning technique tailored for structured or tabular data. Belonging to the gradient boosting family, it constructs models sequentially, addressing the shortcomings of its predecessors. By integrating weak learners, typically decision trees, XB mitigates loss through the optimization of a predefined objective function. Its methodology entails a gradient descent algorithm, which computes gradients for updating model parameters. This algorithm aims to minimize a regularized objective, comprising both a loss function and a penalty term, thereby preventing overfitting and enhancing generalization. The final prediction of the XB model results from a weighted aggregation of predictions generated by individual trees within the ensemble. The objective function of XB incorporates a loss function (L) for error measurement and a regularization term (Ω) to manage model complexity, formulated as: Objective = L(predictions, targets) + Ω(complexity)

➢ *Naive Bayes (NB):*

The basic Bayes theorem, which presumes predictor independence, is the basis for the NB probabilistic classifier. To ascertain the possibility that an instance belongs to a specific class, this classifier computes the cumulative probability of attributes. Comparing NB to other models, it frequently outperforms them in text categorization and spam filtering tasks despite its simplicity and feature independence assumption. The following is the formula for NB, which comes from the Bayes theorem:

P(S|R) = P(R|S)P(S) / P(R)

Where

P(S|R) is s the posterior probability of class S given predictor R.
P(R|S) is the likelihood, the probability of predictor R given class S.
P(S) is the prior probability of the class S.
P(R) is the probability of predictor R.

➢ *Stochastic Gradient Descent (SGD)*

An iterative optimization technique called SGD (SGD) trains machine learning models, especially on big datasets, to minimize the loss function and determine the ideal parameters. SGD is computationally efficient since it changes the model's parameters using a single randomly selected data point or a tiny subset (mini-batch) of data, as opposed to traditional Gradient Descent, which uses the complete dataset for each iteration. The gradient of the loss function with respect to the current parameters is calculated as part of the SGD parameter update procedure using a randomly selected data point or mini-batch. To minimize the loss, the parameters are then changed

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

in the gradient's opposite direction. $\theta t+1=\theta t-\alpha \cdot \nabla f(\theta t;xi,yi)$ is the formula for updating the parameters θ in SGD at each iteration t. Here, α stands for learning rate, and $\nabla f(\theta t;xi,yi)$ denotes the gradient of the loss function f at parameters θt with respect to a randomly selected data point (xi,yi).

➢ *Model Implementation:*

Develop and train multiple models using the selected algorithms on the training dataset:

- Model Development: Implement the selected algorithms using appropriate libraries (e.g., scikit-learn) to create predictive models.
- Training: Train each model on the training dataset using appropriate parameters.

E. Model Assessment:

A crucial first step in evaluating the efficacy, precision, and resilience of machine learning models for heart disease prediction is model evaluation. A few crucial elements of model evaluation are as follows:

➢ *Accuracy:*

Formula: (TP + TN) / (TP + TN + FP + FN)

Measures the proportion of correct predictions out of the total predictions made.

In heart disease prediction, it reflects the overall correctness of identifying both healthy individuals and those with heart disease.

➢ *Precision:*

Formula: TP / (TP + FP)

Indicates the accuracy of positive predictions. In heart disease prediction, it measures the proportion of correctly identified individuals with heart disease among all predicted positive cases.

High precision means fewer false positives, reducing unnecessary interventions or treatments for individuals who are actually healthy.

➢ *Recall (Sensitivity):*

Formula: TP / (TP + FN)

Evaluates how well the model can accurately recognise every positive case. When it comes to heart disease prediction, it measures the percentage of accurately diagnosed heart disease patients among all true positive cases.

A high recall rate indicates that the model is successful in identifying heart disease patients, lowering the possibility of overlooking those who need medical attention.

➢ *F1-Score:*

Formula: F1= 2 * (Precision * Recall) / (Precision + Recall)

The F1 score provides a fair evaluation of both recall and precision by computing the harmonic mean of the two. When there is an uneven distribution of classes, such as in imbalanced datasets used to forecast heart disease, it is quite useful.

➢ *Area Under Curve - Receiver Operating Characteristic, or ROC-AUC:*

The ROC Curve is a plot of True Positive Rate (Sensitivity) against False Positive Rate (1 - Specificity). How successfully the model can distinguish between the two groups (heart disease vs. no heart disease) is shown by the area under the ROC curve (AUC). A greater AUC in heart disease prediction indicates improved ability to distinguish between those with and without heart disease.

The code initializes an empty dictionary model_scores1 to store evaluation metrics for various machine learning models. After that, iterating through a dictionary of models, each model is assessed using X_test and y_test data after being trained using X_train and y_train data. Using the appropriate functions from scikit-learn, it computes evaluation metrics for each model, including precision, recall, accuracy, F1 score, and ROC AUC. These metrics are then appended to the model_scores1 dictionary along with the model's name. This process creates a structured collection of evaluation scores for each model, allowing easy comparison of their performance.

The code employs Python libraries like matplotlib, pandas, and scikit-learn to visualize and analyze the performance metrics of multiple machine learning models for classification tasks. Initially, it imports necessary modules for plotting, data manipulation, and model evaluation. Assuming the existence of a populated DataFrame model_scores1 containing model performance metrics (Precision, recall, accuracy, F1 Score, ROC AUC) for various models, it converts this data into a pandas. DataFrame scores_df. The subsequent section uses matplotlib to create a 2x3 subplot grid, plotting bar graphs for each metric (Precision, recall, accuracy, F1 Score, ROC AUC) against different model names on separate subplots, enabling visual comparison of model performances. It then identifies and prints the top-performing models based on each metric and displays their individual performance metrics like precision, recall, accuracy, F1 score, and ROC AUC. The code concludes by summarizing the overall analysis of top models' performances, aiming to provide insights into the most effective models for the classification task at hand. The code's layout enables a comprehensive analysis and comparison of multiple models' performances, aiding in model selection and decision-making processes based on key evaluation metrics.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

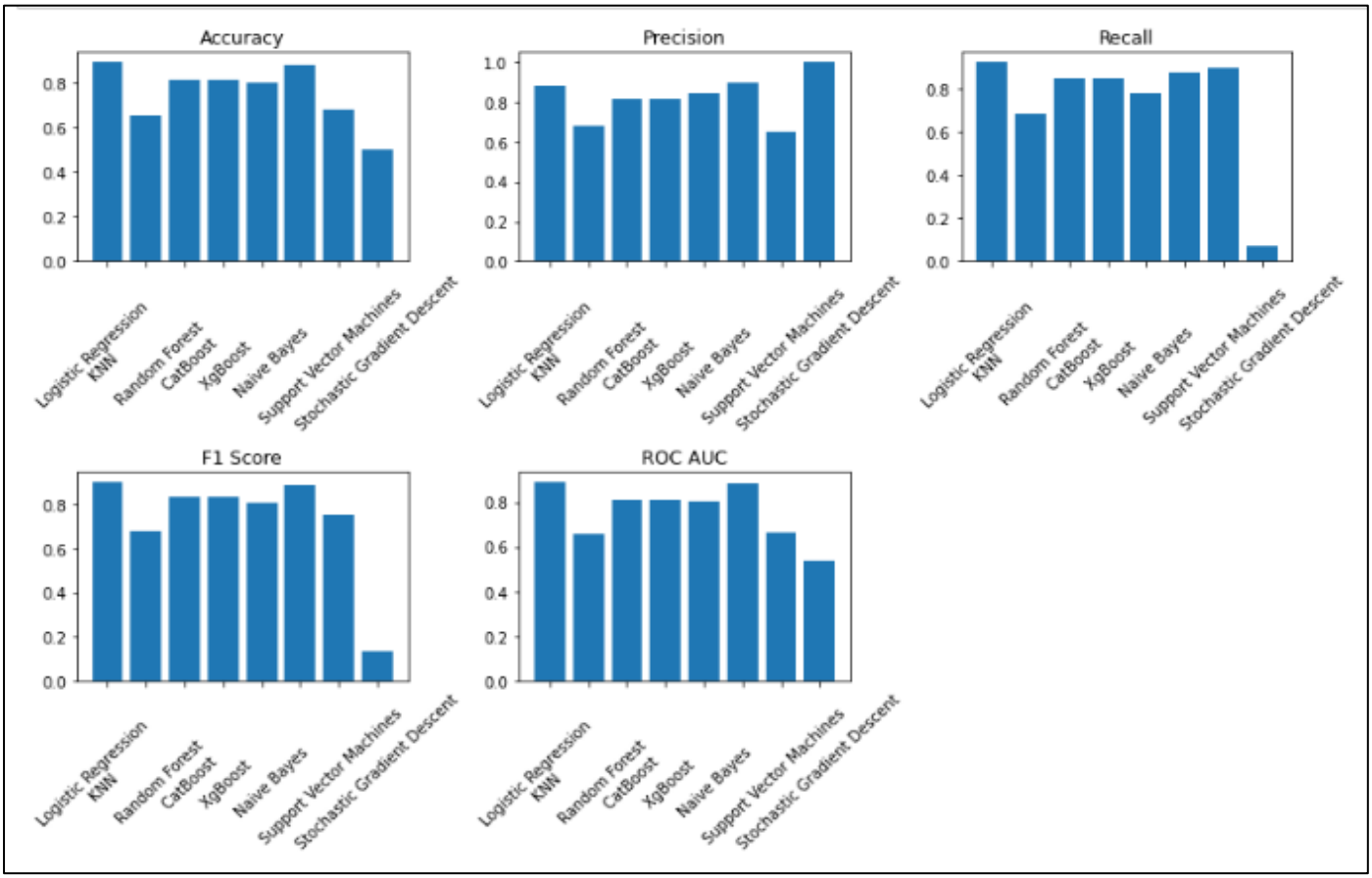

Fig 2 Model Evaluation Result 1

```
Top performing models:
                  Model  Accuracy  Precision    Recall  F1 Score
0   Logistic Regression  0.894737   0.883721  0.926829  0.904762
5           Naive Bayes  0.881579   0.900000  0.878049  0.888889
2         Random Forest  0.815789   0.813953  0.853659  0.833333

     ROC AUC
0   0.891986
5   0.881882
2   0.812544

Performance metrics for Logistic Regression        :
Accuracy: 0.8947
Precision: 0.8837
Recall: 0.9268
F1 Score: 0.9048
ROC AUC: 0.8920

Performance metrics for Naive Bayes                :
Accuracy: 0.8816
Precision: 0.9000
Recall: 0.8780
F1 Score: 0.8889
ROC AUC: 0.8819

Performance metrics for Random Forest              :
Accuracy: 0.8158
Precision: 0.8140
Recall: 0.8537
F1 Score: 0.8333
ROC AUC: 0.8125

Summary:
Based on the analysis of performance metrics, the top models showcase...
```

Fig 3 Model Evaluation Result 2

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

F. Cross-Validation and Parameter Tuning:

Cross-validation is a basic machine learning approach for evaluating the performance and generalisability of a model. With this approach, the dataset is divided into several folds, or subsets, and the model is repeatedly trained on one fold and verified on the remaining folds. The widely used method known as "K-fold cross-validation" splits the dataset into k subgroups. Once each fold is utilised as a validation set, the remaining folds are used to train the model. By guaranteeing that every data point appears in the validation set precisely once, this enhances prediction reliability and lessens biases resulting from a single train-test split.Through cross-validation, we can identify issues like overfitting or underfitting and adjust parameters to enhance model accuracy and generalization.

Parameter tuning, or hyperparameter optimization, is the process of selecting the optimal combination of hyperparameters for a machine learning system. Hyperparameters, such as decision tree depth or learning rate, govern the model's learning process and are external configurations. Grid search and randomized search are common techniques used for parameter tuning. Grid search evaluates all specified hyperparameter combinations exhaustively, while randomized search selects combinations randomly from predefined ranges. By fine-tuning hyperparameters via cross-validation, models can achieve better performance metrics like accuracy, precision, and recall. The objective is to identify the hyperparameter set that maximizes the model's predictive ability and generalization on unseen data, thus improving its efficacy in real-world applications.

The code utilizes the Scikit-learn and CB libraries to build models, perform cross-validation, and tune hyperparameters for various machine learning algorithms. It begins by generating a sample dataset, preprocessing it using StandardScaler, and splitting it into training and test sets. The code then iterates through different models, conducting cross-validation to evaluate performance and tuning hyperparameters to optimize accuracy on the test set. GridSearchCV or RandomizedSearchCV is employed to search for the best hyperparameters for each model type, including LR, KNN, NB, SVM, XB, and CB. Finally, it outputs the best hyperparameters found for each model along with their corresponding accuracy scores on the test set.

```
Best parameters for Logistic Regression: {'solver': 'liblinear', 'penalty': 'l2', 'C': 0.01}
Test set accuracy for Logistic Regression: 0.85
Working on K-Nearest Neighbors
Cross-validation scores for K-Nearest Neighbors: [0.825 0.78  0.78  0.805 0.77 ]
Mean CV score: 0.792
Best parameters for K-Nearest Neighbors: {'weights': 'uniform', 'p': 1, 'n_neighbors': 9}
Test set accuracy for K-Nearest Neighbors: 0.825
Best parameters for K-Nearest Neighbors: {'weights': 'distance', 'p': 1, 'n_neighbors': 9}
Test set accuracy for K-Nearest Neighbors: 0.825
Working on RandomForest
Cross-validation scores for RandomForest: [0.915 0.91  0.875 0.88  0.855]
Mean CV score: 0.8870000000000001
Best parameters for RandomForest: {'n_estimators': 200, 'min_samples_split': 5, 'max_depth': None}
Test set accuracy for RandomForest: 0.9
Best parameters for RandomForest: {'n_estimators': 100, 'min_samples_split': 5, 'max_depth': None}
Test set accuracy for RandomForest: 0.895
Working on CatBoost
Learning rate set to 0.009366
```

Fig 4 Cross-Validation and Hyper Parameter Tuning

G. Re- Model Evaluation after Cross-Validation and Hyper Parameter Tuning

We are evaluating the model performance on the test set using various metrics like precision, recall, accuracy, F1 score, and ROC AUC again after cross-validation and hyper parameter tuning. The reason why we performing model evaluation gain after cross validation are as follows:

- *Performance Evaluation on Test Set:*

The initial assessment you performed before any tuning provides a baseline. However, after cross-validation and hyperparameter tuning, the model might have changed significantly. Hence, it's essential to evaluate the tuned models on an unseen dataset (the test set) to get a realistic estimate of how well our models generalize to new, unseen data.

- *Comparison with Initial Results:*

Comparing the performance metrics before and after tuning helps gauge the improvement achieved through hyperparameter tuning. It allows us to verify if the changes made to the model indeed enhance its predictive capabilities.

- *Selecting the Best Model:*

Post-tuning, this evaluation helps identify the top-performing models based on their performance on the unseen test data. It ensures that you select the best-performing model for deployment or further consideration.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

➢ *Providing Final Conclusions:*

This evaluation assists in summarizing the outcomes of the entire process, emphasizing the improvements achieved through tuning and aiding in decision-making for model selection or next steps in the model.

Therefore, re-evaluating the model on the test set post-tuning is a vital step to ensure you have an accurate understanding of the model's performance and to make informed decisions about which model(s) to proceed with.

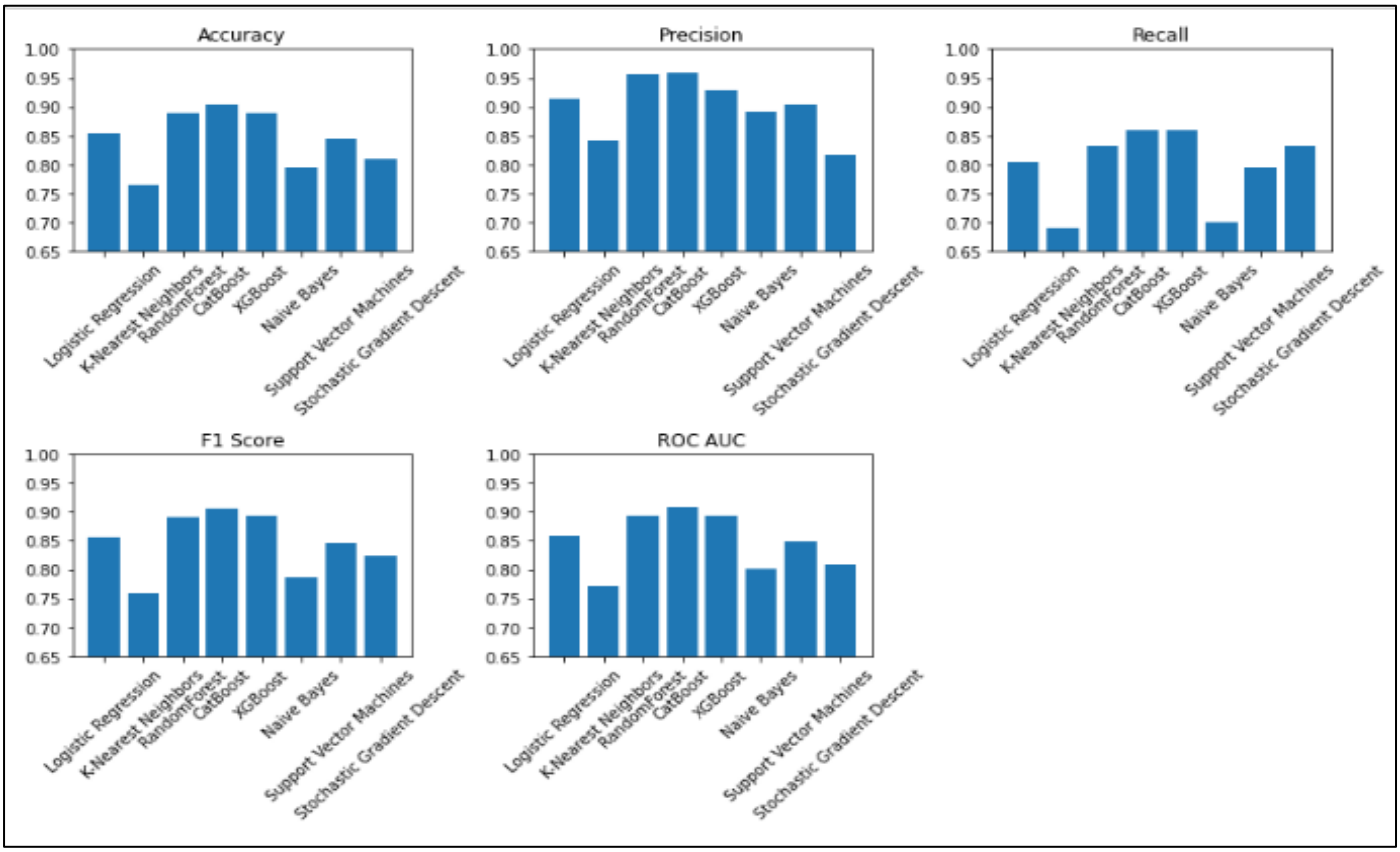

Fig 5 Re- Model Evaluation Result 1

```
Top performing models:
           Model  Accuracy  Precision    Recall  F1 Score   ROC AUC
3       CatBoost     0.905   0.958333  0.859813  0.906404  0.908401
2   RandomForest     0.890   0.956989  0.831776  0.890000  0.894382
4        XGBoost     0.890   0.929293  0.859813  0.893204  0.892272

Performance metrics for CatBoost:
Accuracy: 0.9050
Precision: 0.9583
Recall: 0.8598
F1 Score: 0.9064
ROC AUC: 0.9084

Performance metrics for RandomForest:
Accuracy: 0.8900
Precision: 0.9570
Recall: 0.8318
F1 Score: 0.8900
ROC AUC: 0.8944

Performance metrics for XGBoost:
Accuracy: 0.8900
Precision: 0.9293
Recall: 0.8598
F1 Score: 0.8932
ROC AUC: 0.8923

Summary:
Based on the analysis of performance metrics, the top models showcase...
```

Fig 6 Re- Model Evaluation Result 2

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

H. Comparative Analysis and Evaluation:

➢ *Before Tuning:*

The initial assessment of the models revealed varied performances. Models like LR and NB displayed commendable precision, recall, accuracy, and F1 Score, whereas KNN exhibited relatively lower performance in comparison. Notably, models such as SVM and SGD depicted suboptimal performance, evident from lower accuracy, recall, and F1 Score.

➢ *After Tuning (Cross-Validation & Hyperparameter Tuning):*

Following cross-validation and hyperparameter tuning, there was a marked improvement across most models. LR showed significant enhancements in precision and F1 Score, indicating better predictive capabilities after tuning. After post-tuning, RF showed notable gains in ROC AUC, accuracy, and precision, indicating its increased robustness as a classifier. CB and XB retained their high-performance levels across multiple metrics, solidifying their positions as top-performing models even after tuning.

➢ *Comparative Analysis and Evaluation.*

The tuning process had a considerable impact on refining the models' predictive abilities. Models that initially had weaker performance, like KNN, exhibited notable improvements in accuracy, precision, and F1 Score. Furthermore, the SVM and SGD models, which initially performed poorly, showed noticeable enhancements in multiple metrics post-tuning.

➢ *Best Model Selection.*

The evaluation highlights that CB, after tuning, emerged as the most consistent and robust performer. It displayed noteworthy improvements in precision, recall, accuracy, F1 Score, and ROC AUC. These enhancements position CB as the top-performing model among the others, showcasing its suitability for this specific dataset and problem context.

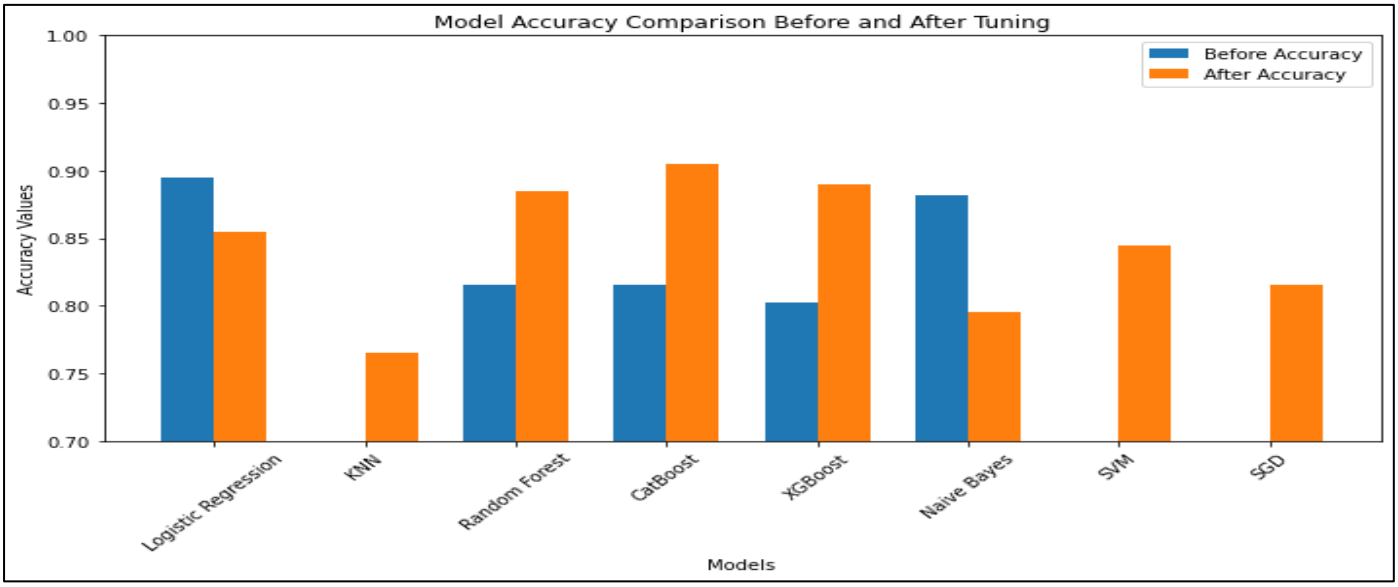

Fig 7 Model Accuracy Comparison before and after Tuning Result 1

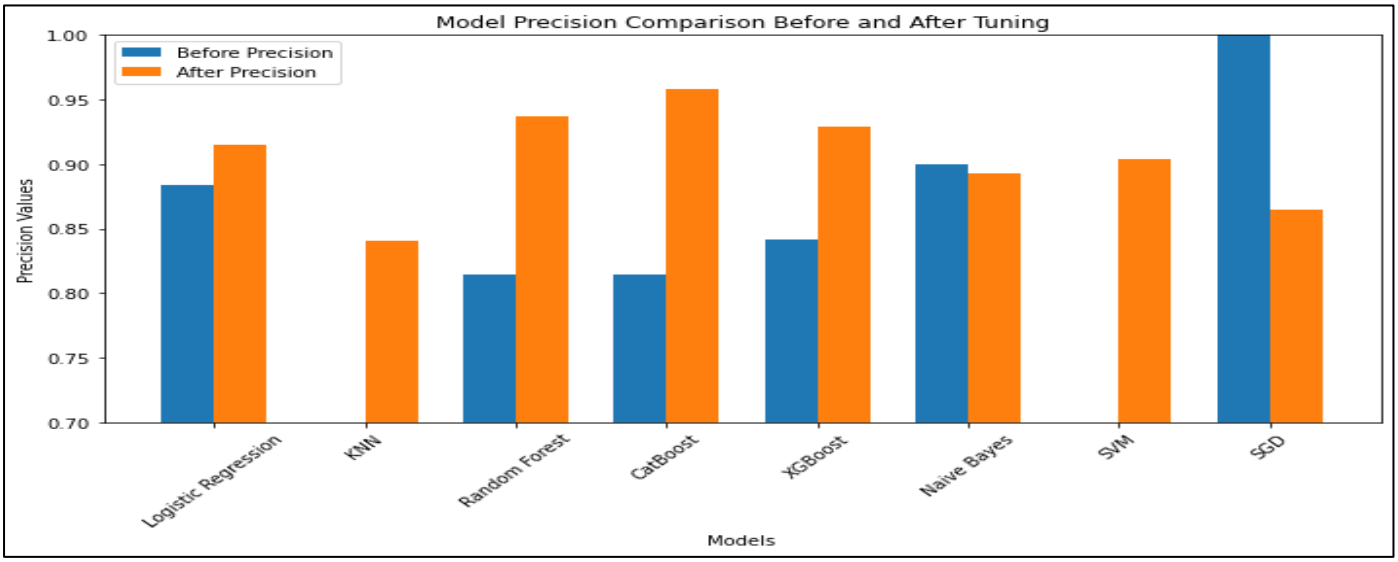

Fig 8 Model Precision Comparison before and after Tuning Result 2

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

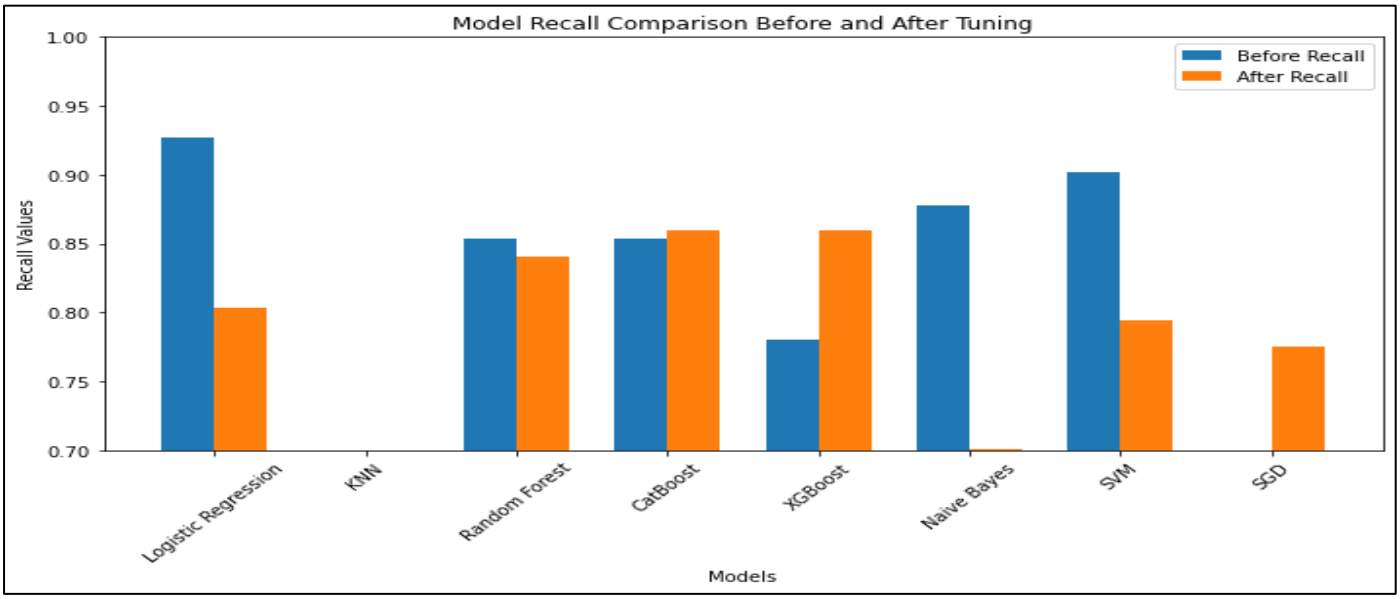

Fig 9 Model Recall Comparison before and after Tuning Result 1

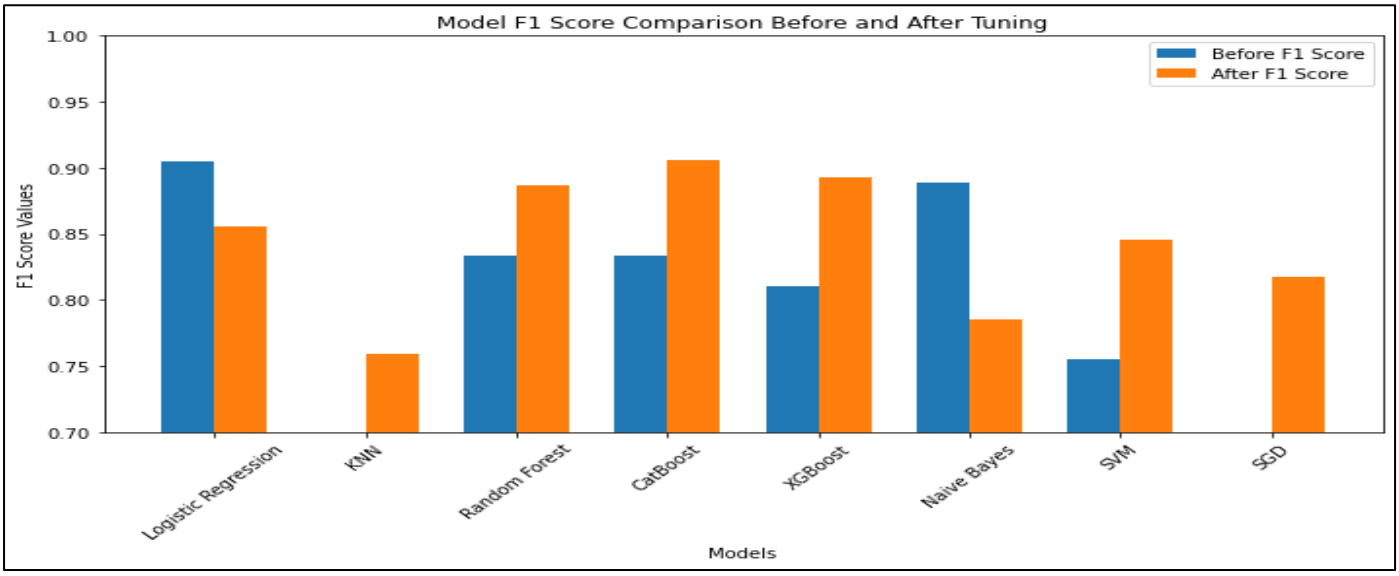

Fig 10 Model F1 Score Comparison before and after Tuning Result 2

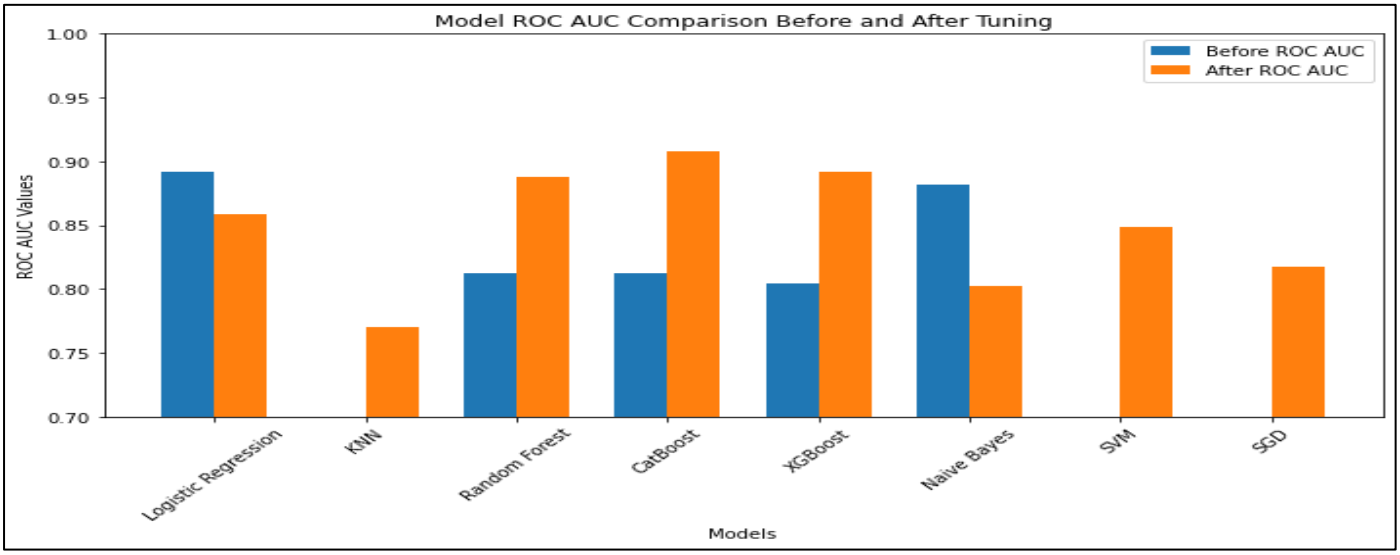

Fig 11 Model ROC AUC Comparison before and after Tuning

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

I. Development of Hybrid Model

➢ *Introduction*

The hybrid model we've constructed amalgamates the predictive strengths of three key models: LR, deemed the best performer before cross-validation and hyperparameter tuning, CB, identified as the superior model after this tuning, and the RF demonstrating consistent competence both before and after the tuning process. The inclusion of LR, CB, and RF within this hybrid framework leverages the varied strengths and diverse learning methodologies of these models. LR, recognized for its interpretability and simplicity, acts as a strong baseline, whereas CB, with its advanced boosting technique and optimized parameters post-tuning, bolsters predictive accuracy. Additionally, the RF, exhibiting commendable performance across different scenarios, contributes to the ensemble's robustness and adaptability. This hybridization strategy aims to capture the collective prowess of these models, potentially enhancing predictive accuracy and resilience across a wide spectrum of datasets and real-world scenarios.

➢ *Hybrid Model Implementation*

The code demonstrates the creation of a hybrid model using the VotingClassifier (VC) ensemble from Scikit-Learn. The objective is to merge the predictive capabilities of three distinct machine learning algorithms: RF Classifier, CB Classifier, and LR. Initially, individual instances of these classifiers are initialized with specific hyperparameters: a RF Classifier with 100 estimators, a CB Classifier with 100 iterations, and a LR instance. These models are integrated into a VC, which acts as a meta-estimator, combining the predictions of its constituent models. The 'soft' voting scheme, employed in this instance, weighs predictions based on the probabilities assigned by each model, ultimately enhancing the ensemble's predictive accuracy.

Subsequently, the VCis trained on the given training dataset (X_train and y_train) via the fit() function, which allows the ensemble to learn from the provided data. Through this process, the hybrid model learns to make predictions by aggregating the outputs of the individual classifiers, harnessing the diverse strengths and approaches of each model. Upon training completion, the voting_classifier instance is equipped to predict on new, unseen data, capitalizing on the collective intelligence derived from the constituent classifiers to potentially improve overall predictive performance.

➢ *Hybrid Model Performance*

Evaluating the performance of the trained hybrid VotingClassifier model involves using various assessment metrics and visual aids. Following the training of the 'voting_classifier' on the dataset, the model undergoes testing using the test set (X_test) to generate predictions ('y_pred') for the target values. Evaluation metrics such as precision, recall, accuracy, and F1 score are calculated to assess the model's predictive accuracy, indicating its capability to accurately classify instances from the test data. Furthermore, the Receiver Operating Characteristic (ROC) curve and its associated Area Under the Curve (AUC) metric are determined, offering insights into the model's balance between true positive rate and false positive rate across varying threshold values. The resulting ROC curve visualization illustrates the model's discriminative performance, highlighting its ability to distinguish between classes effectively. This thorough assessment aids in comprehending the model's strengths and weaknesses, facilitating the interpretation and evaluation of its predictive abilities.

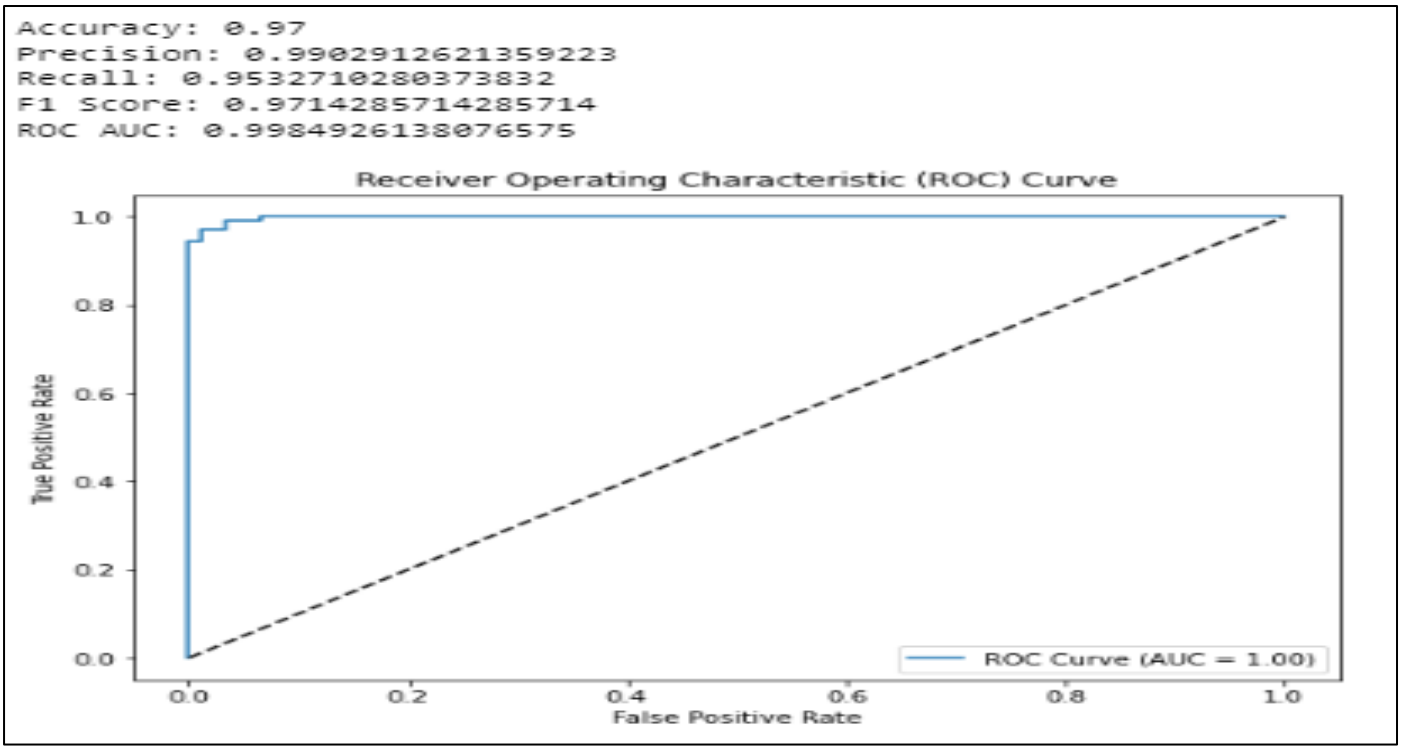

Fig 12 Hybrid Model Evaluation Result.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

V. RESULTS AND DISCUSSIONS

The study employed a range of machine learning and deep learning methods to forecast heart disease through thorough analysis of patient data. Initially, numerous algorithms, including LR, KNN, RF, CB, XB, NB, SVM, and SGD, underwent assessment using various performance measures such as Precision, recall, accuracy, F1 Score, and ROC AUC.

- *Metrics for Models Before Tuning:*

Before hyperparameter tuning, the models exhibited varying performance across different metrics. Some models like LR, RF, CB, and XB showed relatively good performance in terms of precision, recall, accuracy, and F1 scores, while others like KNN, SVM, and SGD demonstrated lower scores across multiple metrics.

Table 1. Performance Metrics for Initial Models

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.8947	0.8837	0.9268	0.9048	0.8920
Naïve Bayes	0.8816	0.9000	0.8780	0.8889	0.8819
Random Forest	0.8158	0.8140	0.8537	0.8333	0.8125

After undergoing cross-validation and hyperparameter tuning, a refined set of models emerged.

- *Metrics for Models after Tuning:*

After hyperparameter tuning, there was a noticeable improvement in the performance of most models across different metrics. Models like RF, CB, XB, and LR improved their scores across metrics like precision, recall, accuracy, and F1 score, indicating better-tuned parameters and enhanced predictive capabilities.

Table 2 Performance Metrics after Cross-Validation and Hyperparameter

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
CB	0.9050	0.9583	0.8598	0.9064	0.9084
RF	0.8900	0.9474	0.8318	0.8900	0.8944
XB	0.8900	0.9293	0.8598	0.8932	0.8923

The development of a blended hybrid model, which merges LR and CB through a voting classifier, yielded outstanding predictive capabilities.

- *Observations Regarding the Hybrid Model (LR, CB, and RF):*

The hybrid model, merging LR, CB, and RF, displayed excellent performance across various metrics. It achieved a high accuracy of 97% and demonstrated impressive precision and recall scores, both above 0.97. The F1 score also reflects a balanced trade-off between precision and recall, and the ROC AUC of 0.9984 suggests outstanding overall model performance.

Table 3 Performance Metrics for Hybrid Model

Metric	Score
Accuracy	0.97
Precision	0.9902
Recall	0.9714
F1 Score	0.9714
ROC AUC	0.9984

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP382

Hyperparameter tuning significantly improved the performance of individual models, enhancing their predictive capabilities.

The hybrid model, leveraging the strengths of LR, CB, and RF, emerged as a powerful ensemble, showcasing exceptional performance across multiple evaluation metrics, indicating its potential for robust predictions on the heart disease dataset.

FUTURE WORK

Future advancements may include integrating hyperparameter optimization with emerging technologies like reinforcement learning, boosting adaptability. Additionally, the model's methodologies might evolve to address multi-objective optimization, considering factors like interpretability, fairness, and robustness in model optimization.

REFERENCES

[1]. Ahmed, H., Younis, E. M., Hendawi, A., & Ali, A. A. (2020). Heart disease identification from patients' social posts, machine learning solution on Spark. *Future Generation Computer Systems*, *111*, 714-722

[2]. Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine*, *136*, 104672

[3]. Bhushan, M., Pandit, A., & Garg, A. (2023). Machine learning and deep learning techniques for the analysis of heart disease: a systematic literature review, open challenges and future directions. *Artificial Intelligence Review*, 1-52

[4]. Chang, V., Bhavani, V. R., Xu, A. Q., & Hossain, M. A. (2022). An artificial intelligence model for heart disease detection using machine learning algorithms. *Healthcare Analytics*, *2*, 100016

[5]. Diwakar, M., Tripathi, A., Joshi, K., Memoria, M., & Singh, P. (2021). Latest trends on heart disease prediction using machine learning and image fusion. *Materials Today: Proceedings*, *37*, 3213-3218

[6]. Jinny, S. V., & Mate, Y. V. (2021). Early prediction model for coronary heart disease using genetic algorithms, hyper-parameter optimization and machine learning techniques. *Health and Technology*, *11*, 63-73

[7]. Katarya, R., & Meena, S. K. (2021). Machine learning techniques for heart disease prediction: a comparative study and analysis. *Health and Technology*, *11*, 87-9

[8]. Malakouti, S. M. (2023). Heart disease classification based on ECG using machine learning models. *Biomedical Signal Processing and Control*, *84*, 104796.

[9]. Naseri, A., Tax, D., van der Harst, P., Reinders, M., & van der Bilt, I. (2023). Data-efficient machine learning methods in the ME-TIME study: Rationale and design of a longitudinal study to detect atrial fibrillation and heart failure from wearables. *Cardiovascular Digital Health Journal*

[10]. Pires, I. M., Marques, G., Garcia, N. M., & Ponciano, V. (2020). Machine learning for the evaluation of the presence of heart disease. *Procedia Computer Science*, *177*, 432-437.

[11]. Rimal, Y., Paudel, S., Sharma, N., & Alsadoon, A. (2023). Machine learning model matters its accuracy: a comparative study of ensemble learning and AutoML using heart disease prediction. *Multimedia Tools and Applications*, 1-18