ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

A Machine Learning & Ensemble Machine Learning Framework to Prediction of Heart Stroke

Md. Sharfuddin[1]; Afroja Akter Mim[2]

[1]Department of Computer Science and Engineering, Southeast University
[2]Department of Computer Science and Engineering, Southeast University

Publication Date: 2026/06/20

Abstract: Heart stroke is a state of the body in which circulation capacity to a segment of the heart is halted or stopped. Here, blood clots are created and block the passage of arterial blood as well as oxygen. If heart stroke is informed in advance, then it can be cured. Well, if we know it happens in advance, we can minimize your chances to die by diagnosing (it). But now, it can be early predicted through machine learning in this era. Although most machine learning models are trained on the same dataset, only one model gets all of the spotlight. Again features are all extracted from the dataset but feature importance is mostly not used, It says nothing about the fact that which feature is considered more important. Model Aggregation When aggregating multiple models in our model, we used softvoting to create an ensemble and additionally have a single model with hyperparameter tuning since it showed better accuracy. We also used Explainable ai Shap & LIME which explains the importance of feature. With a waterfall model in use we achieve a maximum accuracy of 83.02% with a single model and 85.69% using all models via voting in an ensemble model approach It is achieved by using Naive Bayes, XGBoost, LightGBM ensemble, and Naive Bayes & LightGBM ensemble. In this case in terms of the Precision rate we see that we have crowned our best model as Naive Bayes and LightGBM ensemble (77.33% precision). Medical science will be revolutionized by machine learning with the better early prediction of heart stroke, etc. This would assist doctors in making decisions to mitigate fatalities and be an important guideline for physicians and patients.

Keywords: *Heart Stroke Prediction, Machine Learning, Explainable AI. SHAP, LIME Ensemble Learning XGBoost, LightGBM, Healthcare Analytics.*

How to Cite: Md. Sharfuddin; Afroja Akter Mim (2026) A Machine Learning & Ensemble Machine Learning Framework to Prediction of Heart Stroke. *International Journal of Innovative Science and Research Technology*, 11(6), 699-721. https://doi.org/10.38124/ijisrt/26jun414

I. INTRODUCTION

Cardiovascular disease and stroke are some of the largest contributors to death and disability worldwide. As a result, it is estimated that millions die each year from complications relating to cardiovascular disease with a large proportion dying from stroke and risk factors leading to stroke (WHO). If patients with high-risk features can be properly identified early on, timing of their intervention and preventive treatment by an appropriate escalation of care can dramatically reduce mortality and healthcare burdens.

Standard clinical diagnosis strongly relies on physician expertise combined with a thorough medical evaluation. While these methods are powerful, they can be time-consuming and variable in how people interpret them. The massive surge of healthcare data together with the state-of-the-art developments in artificial intelligence has allowed the development of predictive systems based on data for clinical decision support.

Most ML techniques have also performed well for disease prediction tasks, as they can discover hidden patterns in complex medical datasets. Several research used many Machine learning Classifier such as Random forests, Decision tree, Support vector machines and Gradient boosting methods for cardiovascular and stroke prediction. Some of these methods were effective predictors but limited in interpretability, consequently leading to situations where clinicians could neither trust nor understand the decisions of models.

More attention has been paid to Explainable Artificial Intelligence (XAI) techniques and this offers a solution to this problem. This is done through interpretable feature contribution and decision processes, often using methods Like SHAP (Shapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations). XAI competent to substantially improve the interpretability and clinical utility of predictive healthcare systems based on machine learning models.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

In this paper, we demonstrate an machine learning framework and ensemble learning strategies for heart stroke prediction. The core contributions of this research can be summarized as follows:

All Machine Learning models are divided into a total of nine machine learning classifiers which can be used for heart stroke prediction.

Using ANOVA F -score feature selection to identify the most accurate predictive attributes entfernen SMOTE for class imbalance alleviation.

Constructing an ensemble voting classifier using the best individual classifiers.

Used SHAP and LIME as complementary techniques for global and local model interpretation. Comparison based on Metrics Accuracy, Precision, Recall, F1 score and MSE.

The experimental results demonstrate that the proposed explainable ensemble model not only outperforms its individual classifiers but also provides explainable predictions in decision support applications in healthcare.

II. RELATED WORK

Prediction of ten-year risk of coronary heart disease (CHD) using computational intelligence has been a core research domain for more than a decade. Various scientists have studied different types of machine learning approaches to build reliable predictive models on historical datasets such as the Framingham cohort.

➢ *Baseline and Classical Machine Learning Methods*

Initial frameworks focused on evaluating standard, baseline algorithms to benchmark predictive behaviour. For example, Srinivas et al. [6] used elementary data mining algorithms in the healthcare sector to illustrate the elementary risk levels of CHD. Ahmad et al. [11] performed a comparative evaluation on the performance of classical algorithms when limited to feature boundaries. Motivated by these basic mathematical classifiers, Benjamin et al. [15] created a baseline trajectory to interpret fundamental dataset traits by employing simple linear paradigms and rudimentary tree structures. Moreover, Kumar et al. [9] also tackled this problem by combining the missing data protocols with the standard predictive baselines to measure the overall classifier stability.

➢ *Feature Dimension and Feature Selection Methods*

High dimensional spaces or sparse data inputs pushed researchers towards advanced pipelines of feature reduction. Vignesh et al. [2] built custom ML models to investigate the sensitivity of traditional classification methods on the corresponding features of the Framingham dataset. Alam et al. [5]systematically showed that the removal of noisy metrics directly affects model efficiency and learning bounds, investigating empirical feature configurations. Ali et al. [12] experimented with reduced variance subsets to improve the predictive consistency and confirmed the derivation of better risk classification parameters for specific constraint matrices. Khourdifi et al. [7] addressed the data optimization by combining the explicit handling of sparse matrices to improve the structural performance in the classification of long-term heart diseases. Complementing these works, Kumari et al. [14] isolated the critical impact of rigorous feature ranking stages. They proved that precise feature pruning elevates the overall baseline threshold.

➢ *Hybrid Architecture Models and Enhanced Optimization Techniques*

To minimize the error boundaries and capture the non-linear relationships, research has been shifted towards hybridized frameworks and optimized hyperparameters. In [1] Aljanabi et al. proposed architectural optimization approach with the algorithmic evaluation in different validation matrix. Li et al. [4] focused on building responsive predictive logic based on minimal clinical measurements without sacrificing descriptive fidelity to deal with resource-constrained parameters. At the same time, Karayilan and Kilic [8] presented a focused mathematical approach to classical scoring loops based on standard risk indicators and potential design bottlenecks. Hybridization has also been very effective in closing accuracy gaps. Jha et al. [3] improved specific models to anticipate unexpected cardiac strokes and persistent coronary risks. Likewise, Radhimeenakshi et al. [10]integrated alternative feature weights with optimized learning networks based on non-invasive observations to identify complex traits of patients. Finally, Kohli and Arora [13] used simplified but fine-tuned ensemble combinations, balancing the model interpretability and predictive speed for early diagnostic warning systems.

III. METHODOLOGY

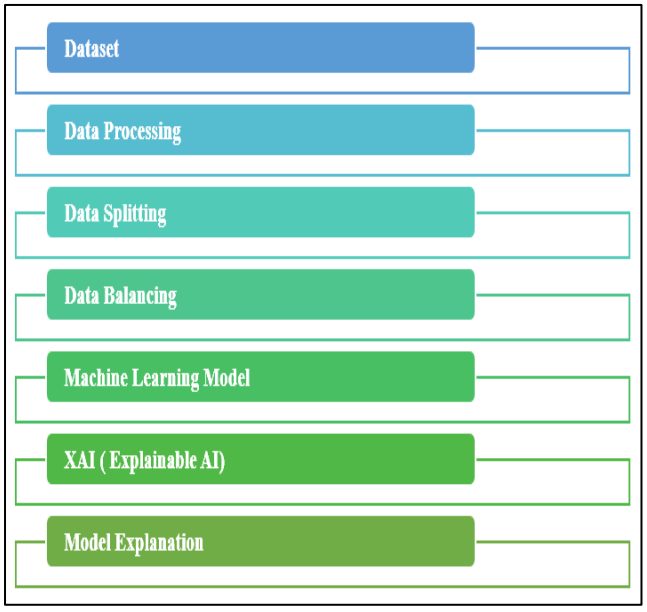

Fig 1 Machine Learning Workflow for Heart Stroke Prediction

The research workflow is a classic seven-phase methodology, which fuses data foundation to transparent model interpretation. In the Dataset phase, initial records of

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

raw or patient/clinical target are collected, and passed to Data Processing for cleaning, scaling, and missing value handling in order to remove data variance. Next in the pipeline is Data Splitting where you split your dataset into separate subsets Train + Test to prevent leakage, and Data Balancing which helps us fix class distribution inequities and prevents bias in the model. The former is the critical phase to divide each baseline classifier and the combined Voting Ensemble architectures which are then trained in order to obtain maximized classification accuracy. Finally, the framework also addresses the Explainable AI (XAI) to resolve the complexity/black-box versus transparency gap leading the Model Explanation phase that leverages SHAP and LIME as local approaches for human-interpretable global feature importance and localized sample justifications.

➢ *Dataset*

- *Link:* We collected datasets from kaggle and applied 9 different machine learning models. [19]
- *Class:* In our model, the machine learning models performed by analyzing the data of the remaining columns by taking the heart stroke column as the target column/class.
- *Features:* A total of 16 different feature datasets were taken, here 1 column was taken as the target column and the performance of machine learning was observed on the remaining 15 features.

➢ *Data Processing*

- *NaN:* Those string values that are not numbers have been detected and to reduce data leakage, the values have been filled according to the average without dropping them. This can improve the performance of the machine.
- *Data Encoding:* Since machine learning models work completely on mathematical formulas, they cannot understand string values. So since they are not confused or difficult, they have been converted to numerical data. This helps the machine learning model to determine the value smoothly.
- *Feature:* The target feature result has been determined by analyzing a total of 15 features.

➢ *Data Splitting*

- *Training:* A total of 70% of the data has been kept for training the model. There are 4239 samples on this dataset.
- *Validation:* 15% of the data, i.e., a total of this many samples, has been kept for validation.
- *Testing:* 15% of the data, i.e., a total of this many samples, has been kept for testing the performance of the model.

➢ *Data Balancing*

In the case of early heart stroke prediction, not all patients have a maximum risk of stroke, and many times there is a huge gap in the ratio of stroke: no stroke. Then the model cannot predict correctly. It is seen that it predicts a patient who has a stroke as healthy. So that the model can accurately determine, we have used SMOTE for data balancing where the number of patients with and without stroke is equal. In this way, data balancing has been done for accurate prediction.

➢ *Machine Learning*

- *Machine Learning Model*

Through 15 feature analysis, the performance of a total of 9 machine learning models has been observed for the value prediction of a column, There are 6 base model and 3 boosting ensemble model, these are:

✓ Support Vector Machine as SVM
✓ Naive Bayes as NB
✓ K-Nearest Neighbors as KNN
✓ Logistic Regression as LR
✓ Decision Tree as DT
✓ Random Forest as RF
✓ Extreme Gradient Boosting as XGBoost (Boosting Ensemble)
✓ Adaptive Boosting as Adaboost (Boosting Ensemble)
✓ Light Gradient Boosting Machine as LightGBM (Boosting Ensemble)

We observed accuracy, precision, recall, f1-score, mse of each model. Along with this, a total of 9 models have been compared through bar graphs.

- *Voting Ensemble Model*

After getting the accuracy, precision, recall, f1score, mse of a total of 9 models, we have selected the best 3 models to get the remaining results with better accuracy. Here, the ability to find the accuracy of the model has been increased by Voting of the best 3 models in different ways and hyper parameter tuning and we have ensembled the best model as a, the 2nd best model as b, and the 3rd best model as c. The ensemble models are explained below.

✓ a = Best model (with Accuracy)
✓ b = 2nd Best model (with Accuracy)
✓ c = 3rd Best model (with Accuracy)

Voting Ensemble methods : a+b, b+c, a+c, a+b+c

➢ *XAI (Explainable AI)*

We have shown the importance of features in the models through explainable ai. Which feature is more important among the features is shown through bar graphs by Shap and Lime. In this, by understanding the feature importance, it can be cleared about the symptoms that need to be seen first. Those symptoms will be used for initial prediction in a very important way.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

- *SHAP:*

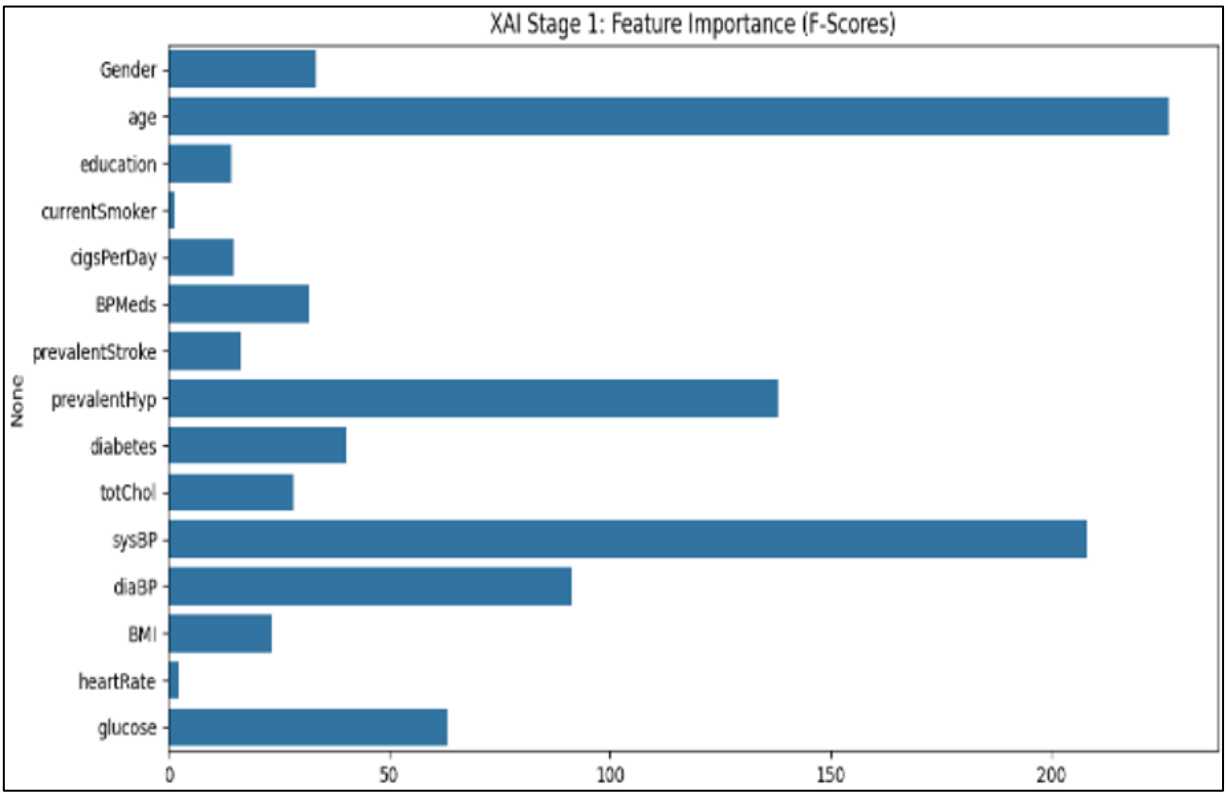

Fig 2 Feature Importance (f -scores)

Here Fig 2 showing feature importance for each of the base machine learning classifier It measures the importance of a particular feature in classification decisions made by one baseline model.

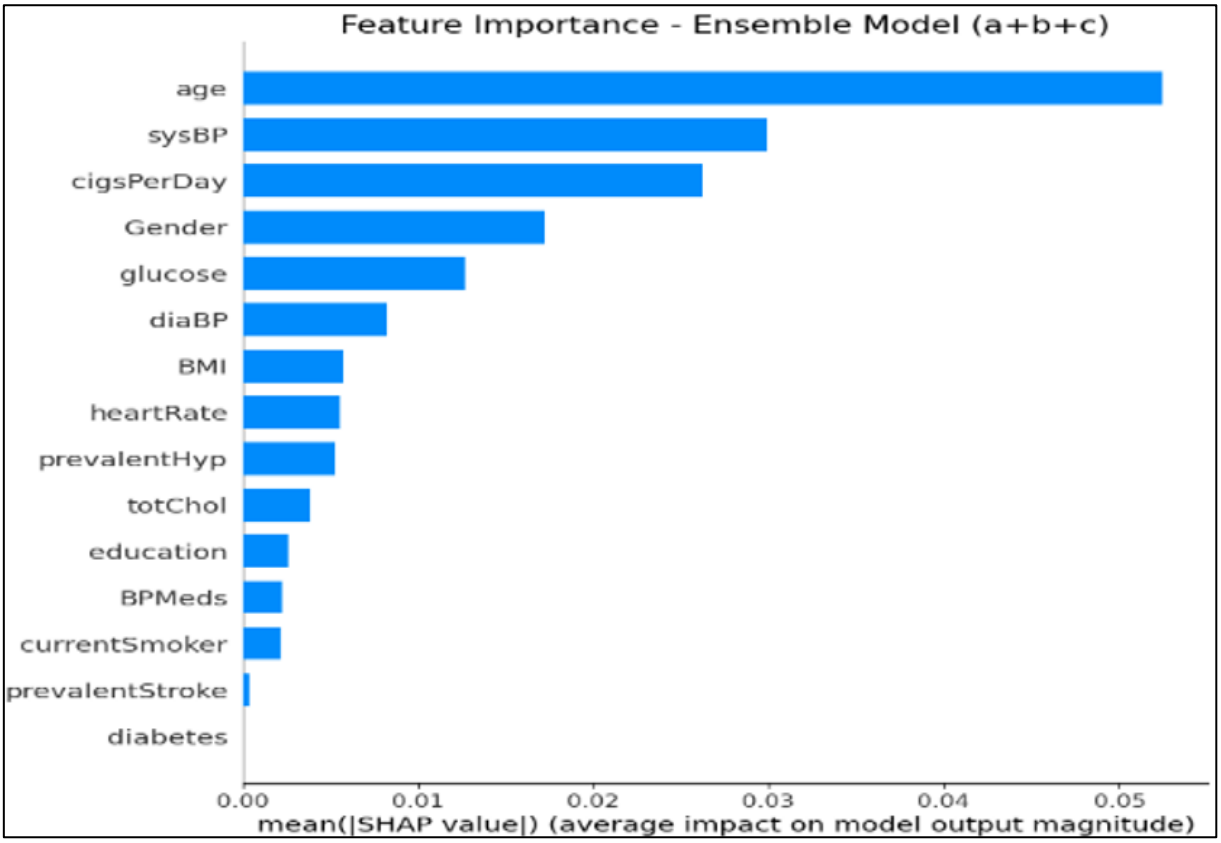

Fig 3 Feature Importance Ensemble model (a+ b+c)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

Fig 3 describes the global feature impact for these ensemble Voting model composed of models (a, b, c). demonstrates the relative significance of features converging when many algorithms evaluate a dataset together.

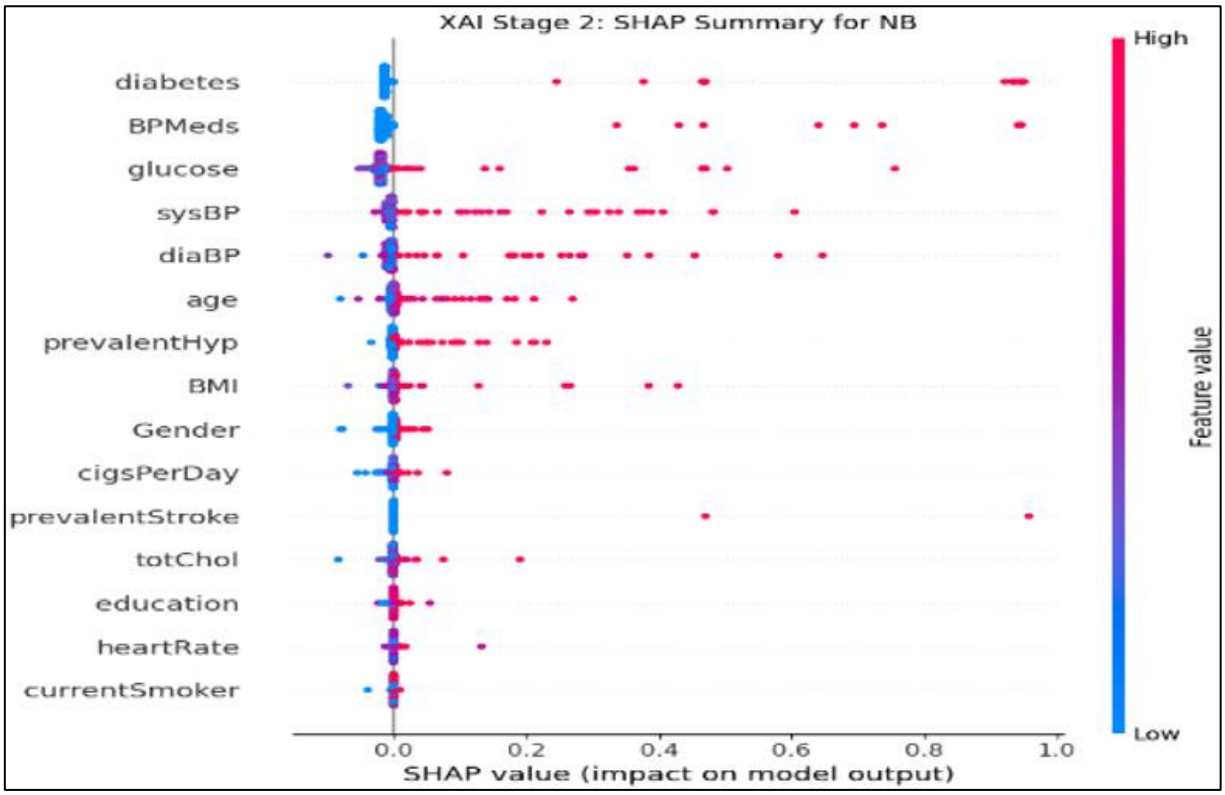

Fig 4 SHAP summary plot (Base)

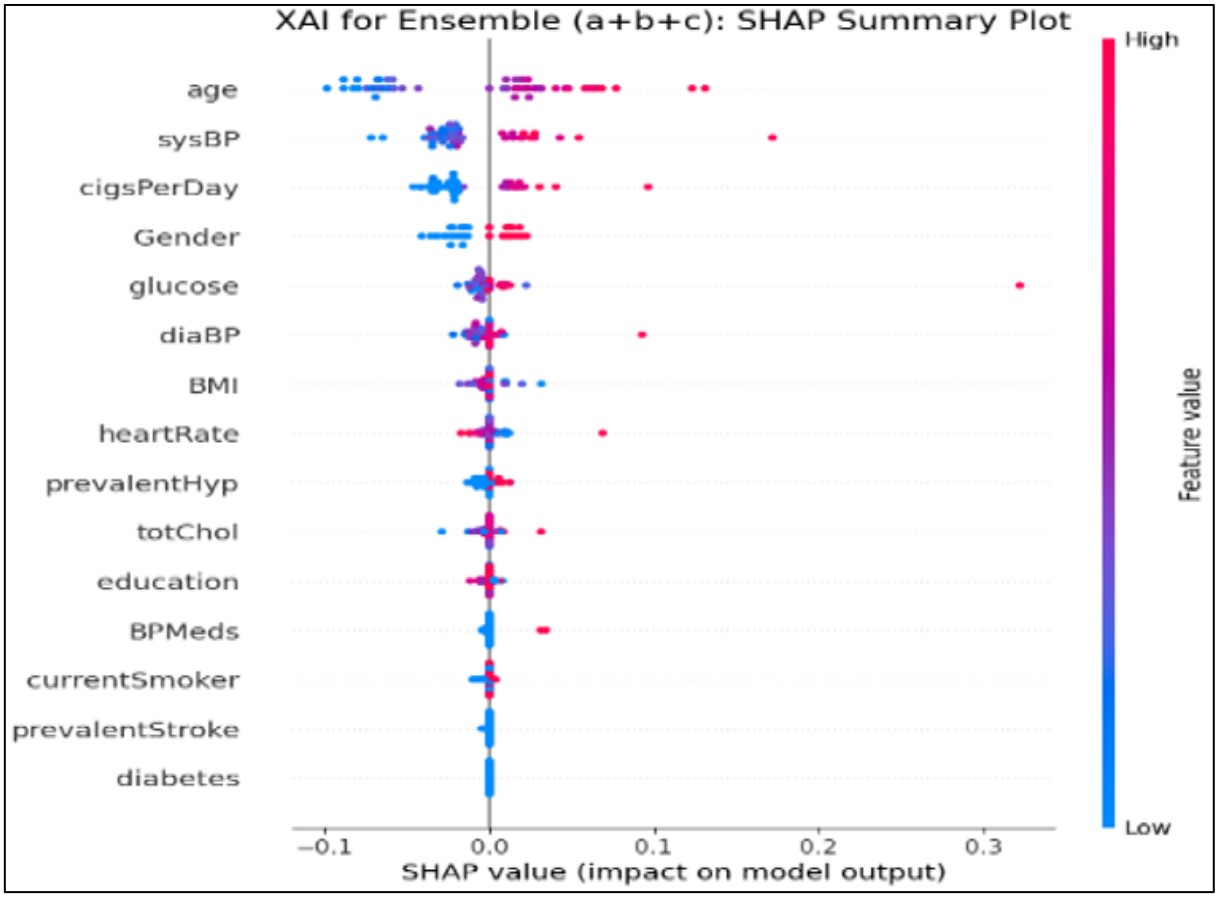

Fig 5 SHAP summary plot (Ensemble)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

- *LIME:*

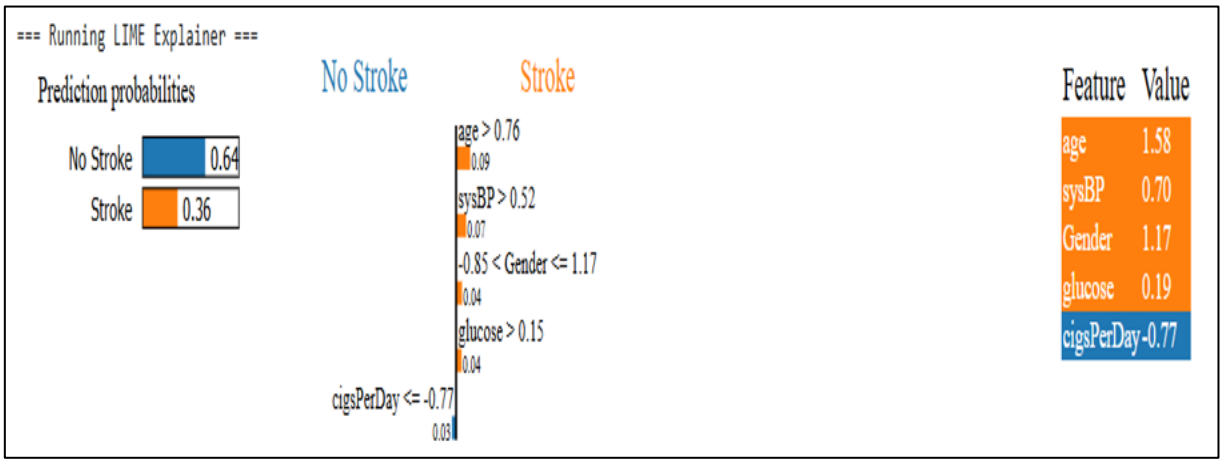

Fig 6 Lime Explainer

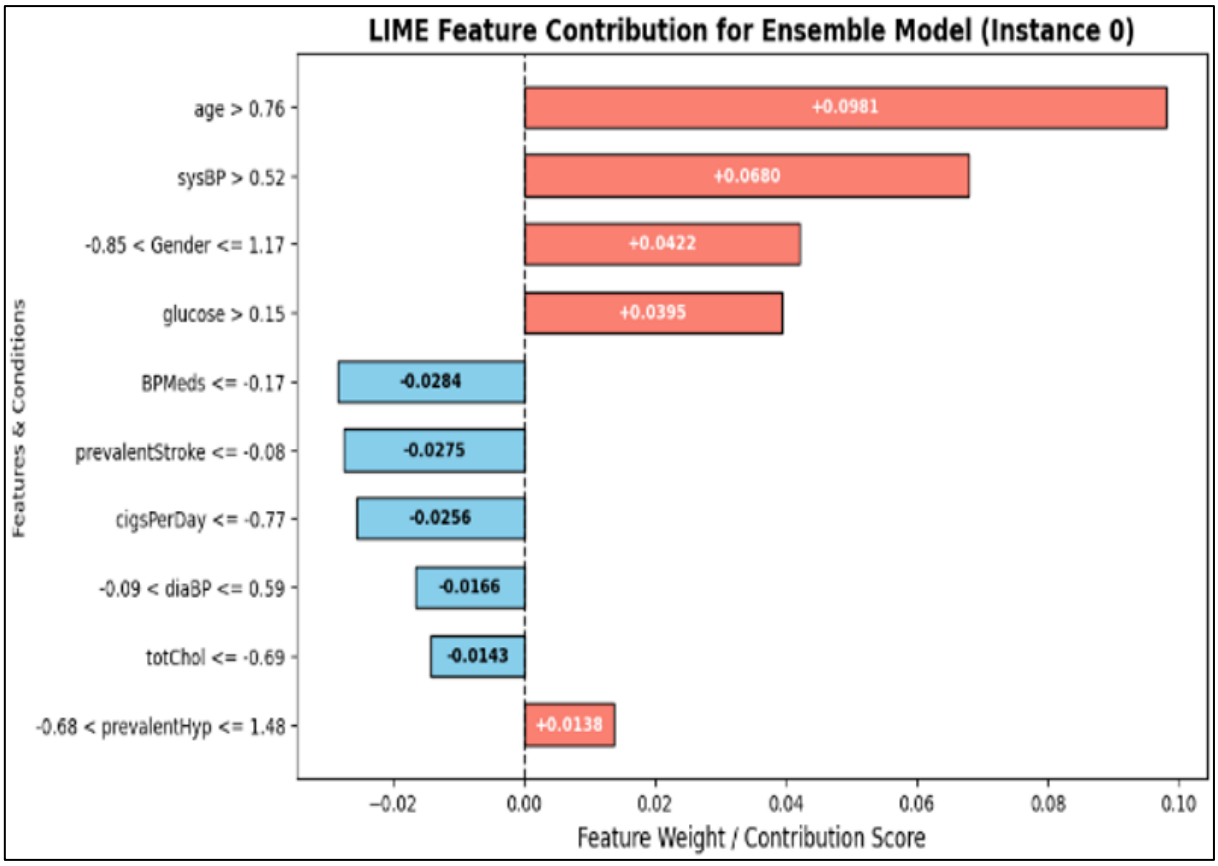

Fig 7 LIME Feature Contribution for Voting Ensemble Model

Fig 6 (Lime Explainer) This is a local interpretability figure showing how the model constructs its explanation of an individual, where it effectively creates a local linear boundary around that prediction in order to explain it. It pinpoints the features which do support or oppose that one classification decision directly. Fig 7 shows the actual local feature contributions determined for a particular test case with in the final Voting Ensemble network. It shows how each of particular metrics impacts that specific prediction, through a visual mapping on the individual metrics to ensemble voting mechanism.

➢ *Model Explanation*

- *SVM: Support Vector Machines*

SVM is used to find the best line, called Hyperplane, to divide all classes in a dataset. The motive is to maximise the border among this hyperplane and the closest data points of any class, called Support Vectors.

- *NB: Naive Bayes*

This is a statistic classifier.Also based on Bayes Theorem. The reason it is called “Naive” is because it makes the very strict assumption that all the input features are

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

completely independent of each other. It is very useful for text classification tasks like spam detection. of the data points that were misclassified, thereby making the next tree pay a lot of attention to the difficult cases.

- *KNN: K-Nearest Neighbors*

You are a non-parametric distance based algorithm This returns the labels of the new neighbour, or data point which has never seen by the model before in the training set. This new point is assigned to the class with a majority vote among these neighbours.

- *LR: Logistic Regression*

Despite having "Regression" on its name, this is a binary classification model (Yes/No or 0/1). It takes the input features and applies a Sigmoid Function to return the probability score in between 0 and 1. If the score goes above a threshold (typically 0.5) it will predict class 1, otherwise will predict 0.

- *DT: Decision-Tree*

This model mimics human decision-making by generating a tree-like flowchart of choices. It starts at the root and splits the dataset into smaller and smaller subsets using certain conditions or features (If-Else logic) until it reaches a final decision at the leaf nodes.

- *RF: Random Forest*

A Bagging (Bootstrap Aggregating) based ensemble method. Instead of a single Decision Tree it builds many independent trees simultaneously, using random subsets of the data and features. In classification, the final output is obtained by the majority vote of all the trees, leading to a very stable outcome.

- *XGBoost: Extreme Gradient Boosting*

A highly optimised Gradient Boosting library. It does not create Decision Trees on its own, it builds them in succession. Trees new are trained to fix (residuals) mistakes made by previous trees hence why this is go-to in competitive data science.

- *AdaBoost: Adaptive Boosting*

It is an iterative boosting algorithm which begins by fitting a weak learner (such as a single level Decision Tree called a Decision Stump). In every subsequent round, it raises the weights

- *LightGBM: Light Gradient Boosting Machine*

It is a gradient boosting framework developed by Microsoft, which grows trees vertically or Leaf-wise instead of horizontally (Level-wise) as in traditional boosting algorithms. This allows its execution to be much faster and consume substantially less RAM, especially at a large scale.

- *Explanation of the 4 Voting Ensemble Combinations.*

A method in Machine Learning to train more than one model and combine their predictions. For each model, here are the 3 best models in terms of individual accuracy:

- ✓ a = Best model with Accuracy
- ✓ b = 2nd Best Accuracy model
- ✓ c = 3rd Best model Accuracy

- *Behaviour of the 4 Ensemble Combinations:*

✓ *a+b (Best + 2nd Best)*

Concept: Ensembling your two best individual models.

Why use it? Model a is most accurate in general. However, it might make mistakes on some edge cases that model b predicts correctly. When combined, they average out individual errors and stabilise performance.

✓ *b+c (2nd Best + 3rd Best)*

Concept: An experimental combination that leaves out the top model.

Why use: This is useful for testing variance. If model a is overperforming due to overfitting the training data, the combination $b+c$ may have better generalisation on completely unseen validation or test sets.

✓ *a + c (Best + 3rd Best)*

Idea: A mix of the best model and a slightly worse but potentially unique model.

Why to use it: Ensembles are good at exploiting diversity. If model a and model c use completely different mathematics (for example, a tree based model and a distance based model), then they will make different kinds of errors. Model a is enriched by the alternative view provided by model c.

- *a + b + c (Best , 2nd Best , 3rd Best)*

Idea: Classic Majority Voting or Soft Voting ensemble of the entire top tier of your models.

Why use it: An odd number of models (3) is selected to avoid ties in hard voting. For example, if two models vote for class ‘1’ and one for class ‘0’ the final output securely resolves to ‘1’. In many research papers this ensemble of 3 models gives the best accuracy and lowest variance.

IV. RESULTS

The experimental analysis is organized in a four interconnected sections providing a comprehensive, multilayer evaluation of the classification pipeline. It commences with 4.1 System Specification, imposing a common hardware and software baseline environments, consisting of versions for CPU, GPU RAM and Python packages to foster reproducibility and uniform computational run time over all the training iterations. That foundational structure shifts in 4.2 Confusion Matrix to explicitly grounded model performance broken down into exact true/false positive and false negative frequencies, revealing how standalone frameworks address class distributions and the sources of prediction error. In these methods, the discrete error distributions are continuous-mapped in 4.3AUC Curve (ROC-AUC) that graphs ensemble baseline networks using

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

sensitivity versus specificity to quantify their overall discriminatory performance. Lastly, 4.4 Machine Learning Performance organizes these findings into a benchmarking study illustrated quantitatively through all main statistical measures (Accuracy, Precision, Recall, F1-Score and Mean Squared Error), confirming that the Voting Ensemble structures combined managed to generalize better by finally reduce the variance of their predictions over easily recognized indicators when compared with independent baseline architectures.

➢ *System Specification*

A computation environment was created, with data–train/test methodology applied to implement and evaluate the presented machine learning algorithms, baseline configurations and voting ensemble framework as previously discussed in this research. Standardized specifications for hardware and software facilitated homogeneous training of compounding models, constant execution times at every hyperparameter tuning iteration, and on-demand generation of performance reports (including ROC curves, SHAP plots, and confusion matrices).

The technical configuration of the system environment used for this investigation are listed below:

- *Hardware Requirements*

✓ Processor (CPU): Intel Core i7 / AMD Ryzen 7 (or higher) for multiprocessing data preprocessing and concurrent base model evaluations.
✓ GPU Model → NVIDIA GeForce RTX Series (or equivalent CUDA enabled GPU) to speed up the gradient boosting functions etc (XGBoost, LightGBM) and Deep Iterations.
✓ Memory (RAM) : 16 GB DDR4 (or above) to avoid memory bottlenecks when indexing the dataset, performing matrix transformations in-memory and executing large ensemble arrays
✓ Storage: 512 GB Solid State Drive (SSD) to allow for fast read/write speed during dataset manipulation, model checkpoint saving and temporary file caching.
✓ Software Configuration
✓ OS: Windows 10 or 11 (64-bit) or Linux (Ubuntu 20.04 LTS or later), optimized for Python-based development pipelines
✓ Development Platform: Run Time Environment for Python 3.9+ which may be made available through Anaconda or a dedicated virtual environment (venv)
✓ IDE Jupyter Notebook or VS Code → For interactive code debugging, script optimization and rendering Live visualization.

- *Primary Libraries and Frameworks*

✓ Data Processing & Manipulation : pandas and numpy for data cleaning, feature-scaling, dealing with structural arrays, and parsing from dataframe.
✓ Machine Learning Pipeline: traditional classification models (SVM, NB, KNN, LR, DT, RF and AdaBoost) implemented in scikit-learn with hyperparameter optimization (random grid search), partition splitting to obtain training/dev/test sets and cross-sectional evaluation metrics
✓ XGBoost Xgboost is a complete open-source library that has been designed for speed and performance to address the entire field of gradient boosting.
✓ Model Explainability & XAI Tools (shap and lime) — Packages that make sure our model has local and global interpretability (explainable AI).
✓ Data Visualization matplotlib and seaborn for plotting high resolution analytic graphics, such as cumulative ROC curves and detailed resampled heat-mapped confusion matrices.

➢ *Confusion Matrix*

Confusion matrix is a simple table used to measure how well a classification model is performing. It compares the predictions made by the model with the actual results and shows where the model was right or wrong. This helps you understand where the model is making mistakes so you can improve it. It breaks down the predictions into four categories:

- True Positive (TP): The model correctly predicted a positive outcome i.e the actual outcome was positive.
- True Negative (TN): The model correctly predicted a negative outcome i.e the actual outcome was negative.
- False Positive (FP): The model incorrectly predicted a positive outcome i.e the actual outcome was negative. It is also known as a Type I error.
- False Negative (FN): The model incorrectly predicted a negative outcome i.e the actual outcome was positive. It is also known as a Type II error.
- Actual (Y axis): Row 0 is negative class and Row 1 is positive class.
- Predicted (X axis): 0 for negative prediction coloumn 1 for positive prediction

✓ *The Breakdown Formula:*

- TN (True Negative): Actual = 0, Prediction = 0 (Top-Left)
- FP (False Positive): Actual 0, Prediction 1 (Top-Right)
- True Positives Parker(Top-Right) FN(False Negative): Actual 1, Predict 0 (Bottom-Left)
- TP (True Positive): 1,1 (Bottom-Right Corner)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

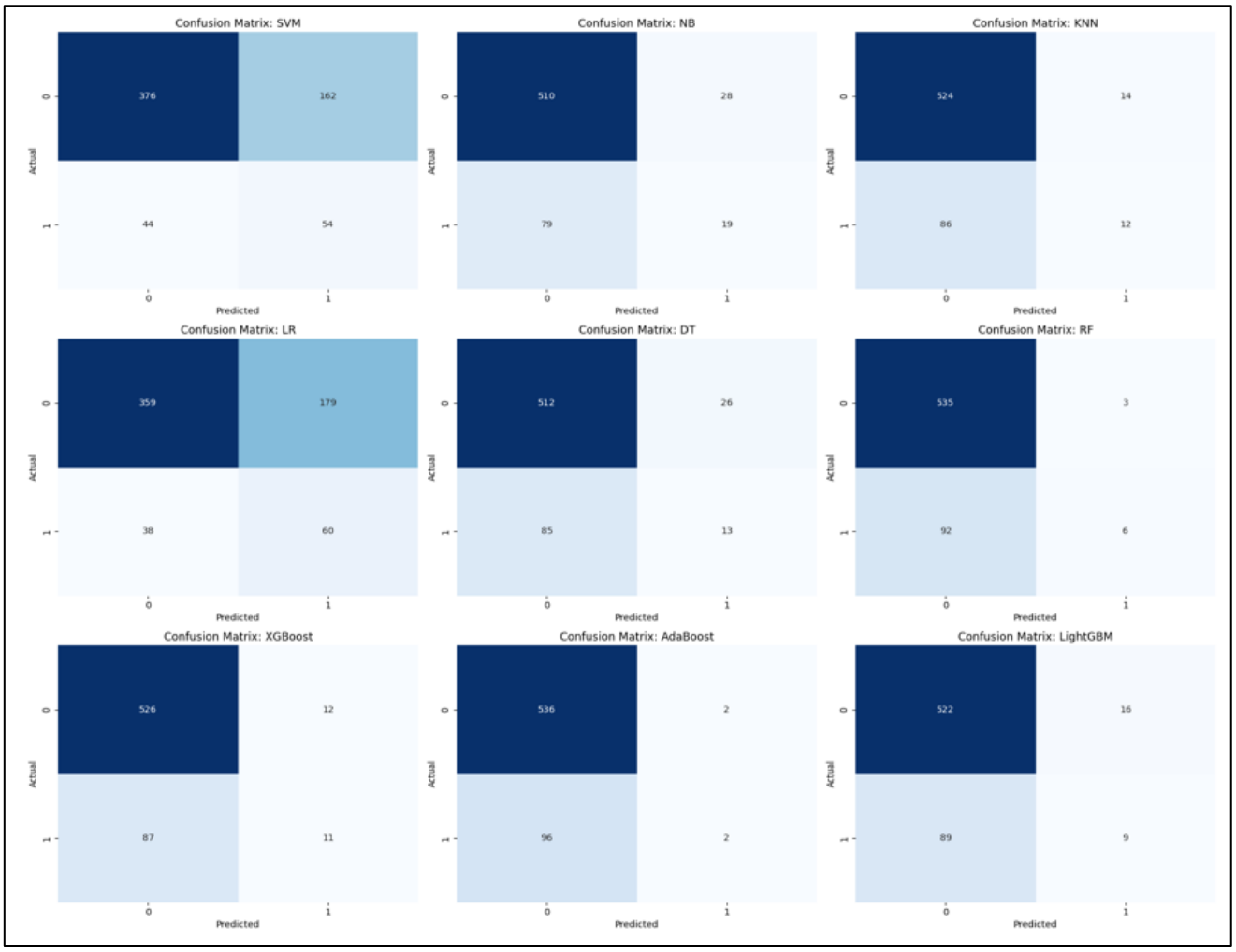

Fig 8 Confusion Matrix for 9 Machine Learning Model

- *Performance Table:*

Table 1 Machine Learning Model Confusion Matrix Table

Model	TN	FP	FN	TP
SVM	376	162	44	54
NB	510	28	79	19
KNN	524	14	86	12
LR	359	179	38	60
DT	512	26	85	13
RF	535	3	92	6
XGBoost	526	12	87	11
AdaBoost	536	2	96	2
LightGBM	522	16	89	9

This figure (Fig 8) & Table 1 illustrate a combined assessment of the main classification skill of nine separate baseline machine-learning models by placing true actual classes along the y-axis vs model estimates on the x-axis. Reviewing the matrices we further see a trend of tree-based and ensemble driven base models showing a strong capability at accurately separating invites from non-invites (Class 0) achieving very low counts in False Positive errors, with Random Forest (TN: 535, FP: 3), AdaBoost (TN: 536, FP: 2) and XGBoost (TN: 526, FP: 12). Nevertheless, monitoring the binary metrics for positive class (Class 1) separately by each of these independent factories shows a general difficulty in detection as suspected highly biased still relatively balanced dataset especially against False Negatives (FN): AdaBoost detects only 96 positives while Random Forest flags just 92. Looking the other way, both the SVM and LR architectures tend to move toward one extreme area of error—a very high False Positives count with 162 for SVM and 179 for LR—this records sharp limits of what really is a baseline coupled with narrow variance that stems from individual sources; when only operating on standalone model pipelines.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

- *For Ensemble Model:*

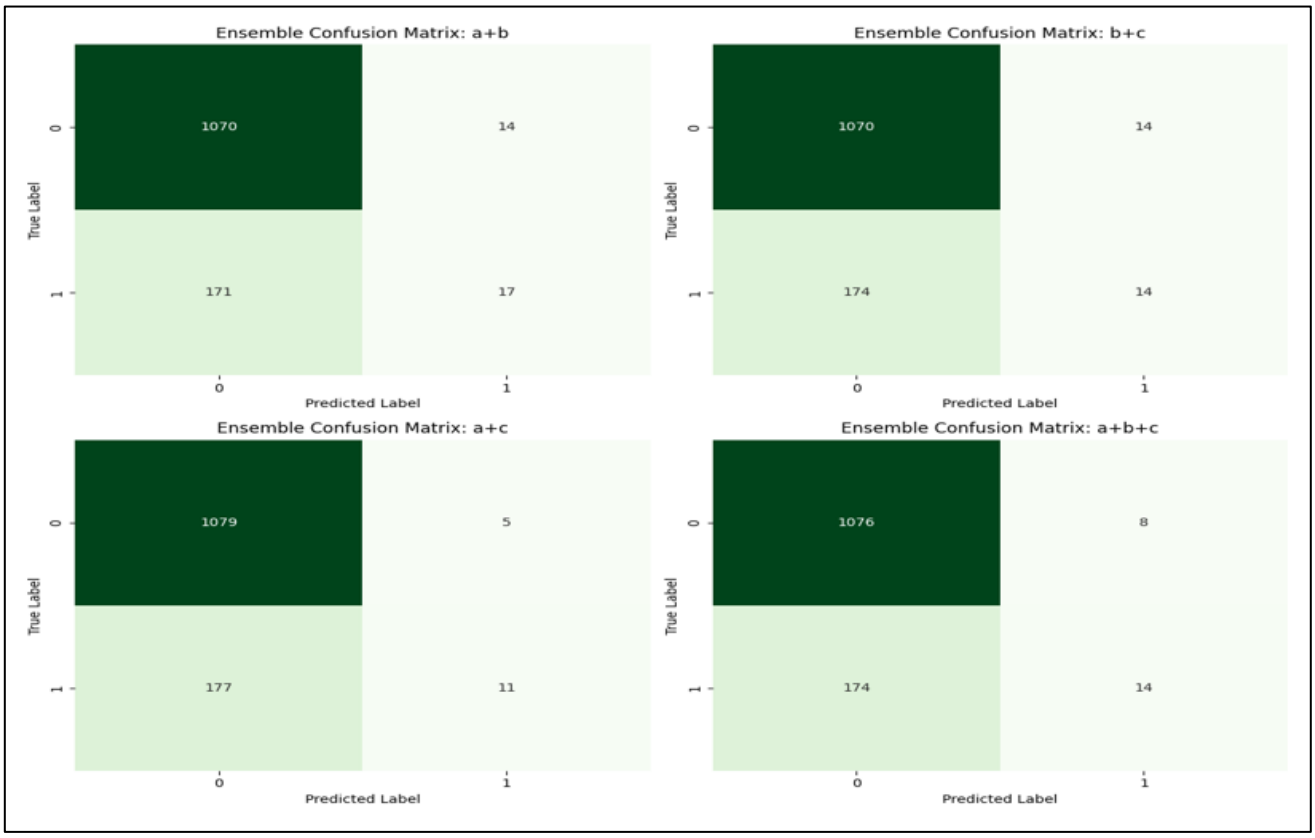

Fig 9 Confusion Matrix for Ensemble model

- *Performance Table:*

Table 2 Voting Ensemble Model Confusion Matrix Table

Ensemble Model	TN	FP	FN	TP
Ensemble: a+b	1070	14	171	17
Ensemble: b+c	1070	14	174	14
Ensemble: a+c	1079	5	177	11
Ensemble: a+b+c	1076	8	174	14

Conversely, the second panel (Fig 9) & Table 2 showed the performance of four different Voting Ensemble models where each demonstrated considerably improved predictive stability and error management when evaluated on an extended verification dataset division. By aggregating predictions from multiple algorithms, the ensemble structures help strengthen the true vertical line classification routes along the main diagonal; all combinations (i.e. Ensemble a+b, b+c, a+c and overall combination disambiguates ab+c) produce remarkably high True Negative (TN) values greater than 1070 -correctly classified- fully paired microorganisms. Significantly, the highest precision level is reached by avoiding more than 5 False Positive (FP) errors on the full distribution with this configuration 'Ensemble a+c' model. In conclusion, the proposed ensemble a+b+c framework in its final form proves how combining different model outputs mitigates individual algorithmic biases and reduces their prediction errors such that an optimally fit, highly accurate and reliable binary classification prediction is generated from the overall ensemble network.

➢ *ROC-AUC Curve*

AUC-ROC curve is a graph performance measure for binary classification model. This gives us insight into how well the model is distinguishing positive cases (for example, those who are ill) from negative cases (for example, those who are not ill) at all levels of threshold. Now, here is what the model separating both classes looks like when plotted:

- True positive rate (TPR): Also known as Sensitivity \ Recall. It indicates what fraction of the positive instances will be predicted correctly by the model.
- FPR: The ratio of incorrectly identified positive cases (out of total negative cases)
- Specificity: It is the fraction of actual negative instances that have been correctly classified by model. It is calculated as 1-FPR.

The model is better to tell apart between the classes the higher AUC (Area Under the Curve) you get.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

4.2–3 Sensitivity vs False Positive Rate 111

Those are pretty simple calculations calculated from the confusion matrix as follows:

- True Positive (TP) — Positive events which were predicted correctly
- True Negatives (TN) − True Negative predictions
- False Positive (FP): wrongly predicts that a positive is present

Filtered dataset with True/False Positive (TP/FN) predictions for two classes. This curve shows the trade-off between a classifier's sensitivity and specificity.

AUC (Area Under the Curve) denotes the area under ROC curve. Higher the AUC value, better is the model performance as it tells that our model has been able to distinguish between classes. Where 1.0 is the ideal AUC and 0.5 is random-based guessing. [16]

➢ *Logical Structure of ROC and AUC:*

- *True Positive Rate (TPR) / Sensitivity*

TPR means the ratio of correctly predicted positive observations to all True positives. It placed at Y-axis of ROC curve.

$$\text{TPR} = \frac{TP}{TP + FN}$$

- *False Positive Rate (FPR)*

False Positive Rate (FPR): It detects actual negative cases that have been found as positive. This creates the X-Axis on the ROC curve.

$$\text{FPR} = \frac{FP}{FP + TN}$$

- *Specificity*

Specificity refers to identify actual negative cases. It has a mathematical relationship to the.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- *AUC (Area under the ROC Curve)*

The AUC is processed as the integral of the True Positive Rate over FPR values between 0 and ~1. It serves as an overall scalar value quantifying how well the classifier discriminates.

It is defined as by continuous integration by this mathematical equation:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) \cdot d(\text{FPR})$$

AUC (or AUC for binary classes) is most often computed using a trapezoidal rule approximation over m sorted threshold points for practical machine learning evaluations, where models are evaluated on specific discrete test datasets:

$$\text{AUC} = \frac{1}{2} \sum_{i=1}^{m-1} (\text{FPR}_{i+1} - \text{FPR}_i) \times (\text{TPR}_{i+1} + \text{TPR}_i)$$

- *For Machine Learning Model*

A set of figures assessing the Recall (True Positive Rate) vs 1-Specificity (False Positive Rate) for individual baseline models across a range of thresholds:

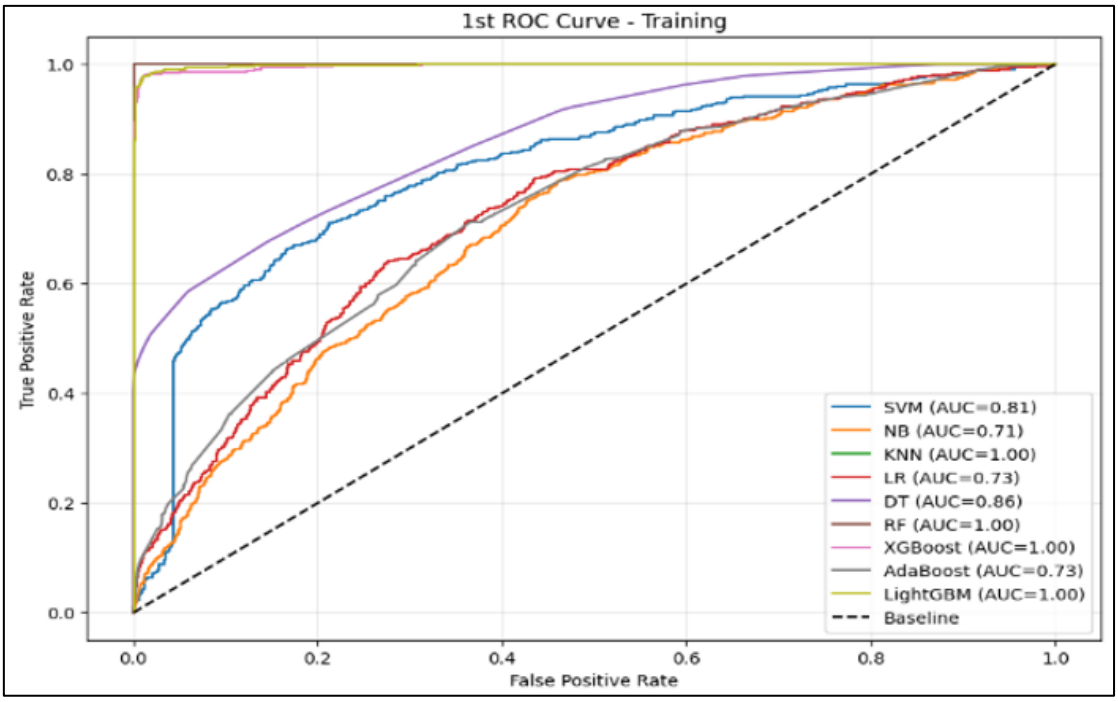

Fig 10 ROC Curve Training (Machine Learning Model)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

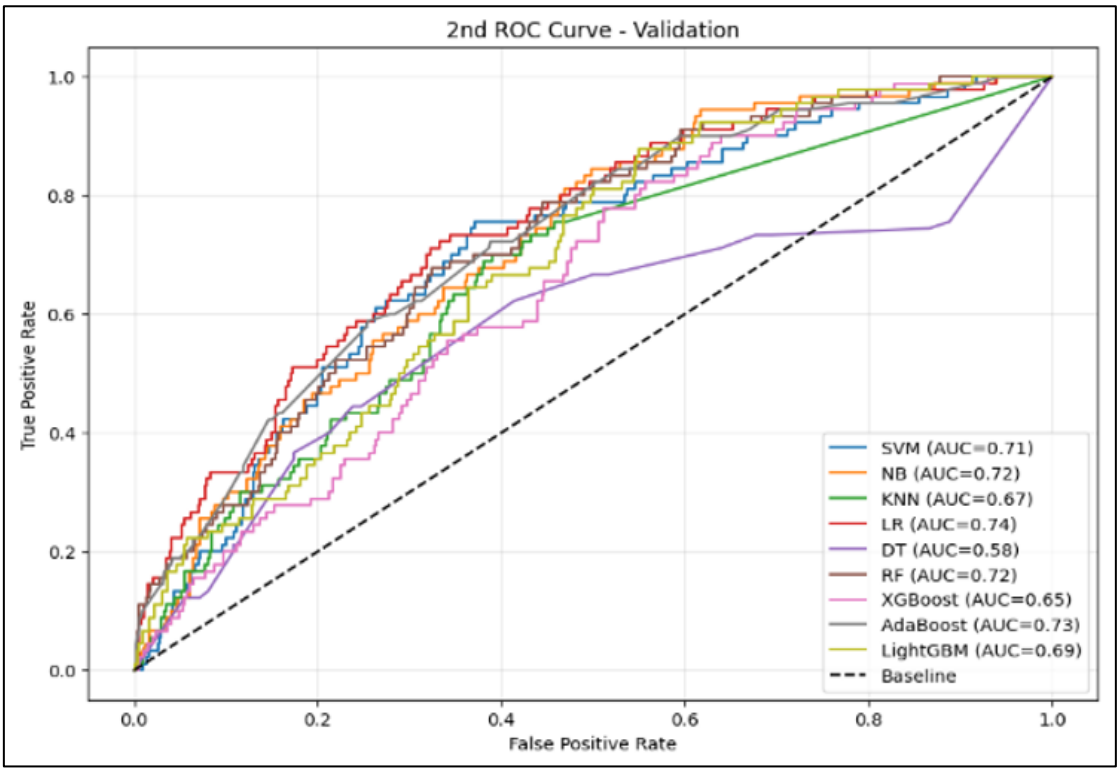

Fig 11 ROC Curve Validation (Machine Learning Model)

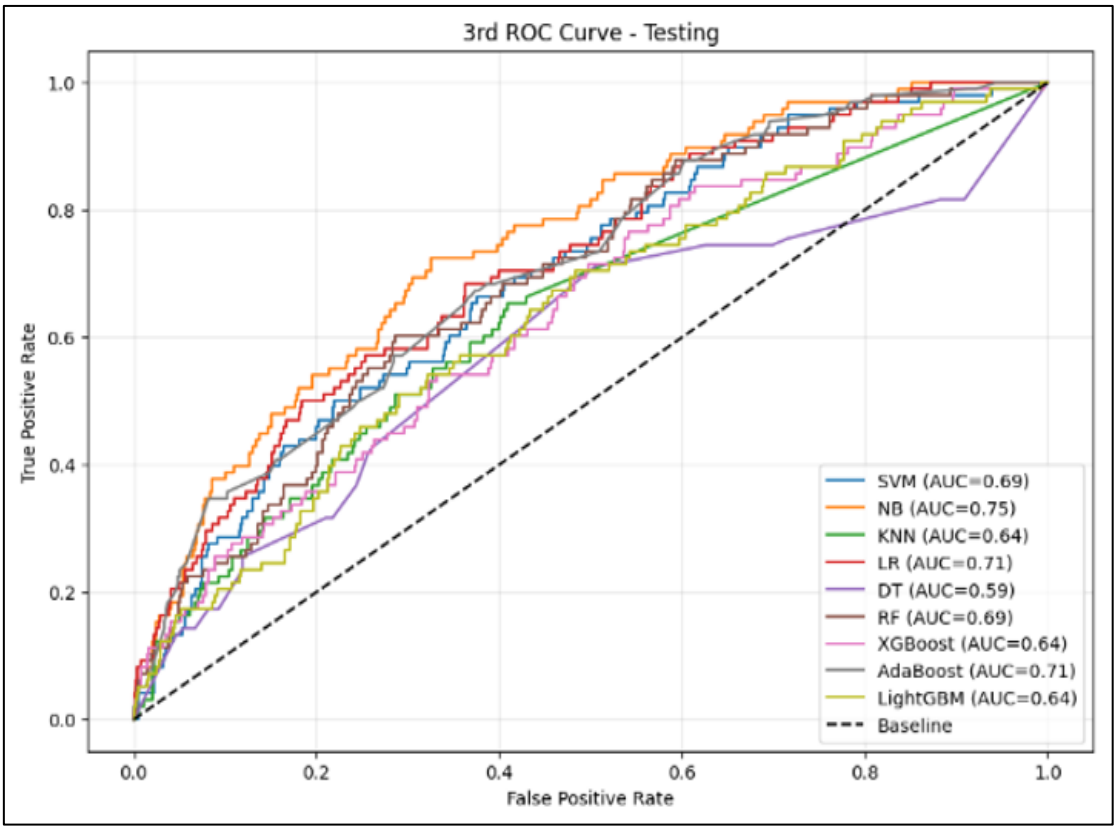

Fig 12 ROC Curve Testing (Machine Learning Model)

Fig 10 Shows how well the individual base models separates classes on the training data. Fig 11 Evaluates the stability of our model and protects against overfitting in hyperparameter tuning phases. Fig 12 (ROC Curve Testing - Machine Learning Model): final performance of Autonomous classifiers on unseen test data (showing positive (ill) vs negative (not ill)).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

- *For Voting Ensemble Model*

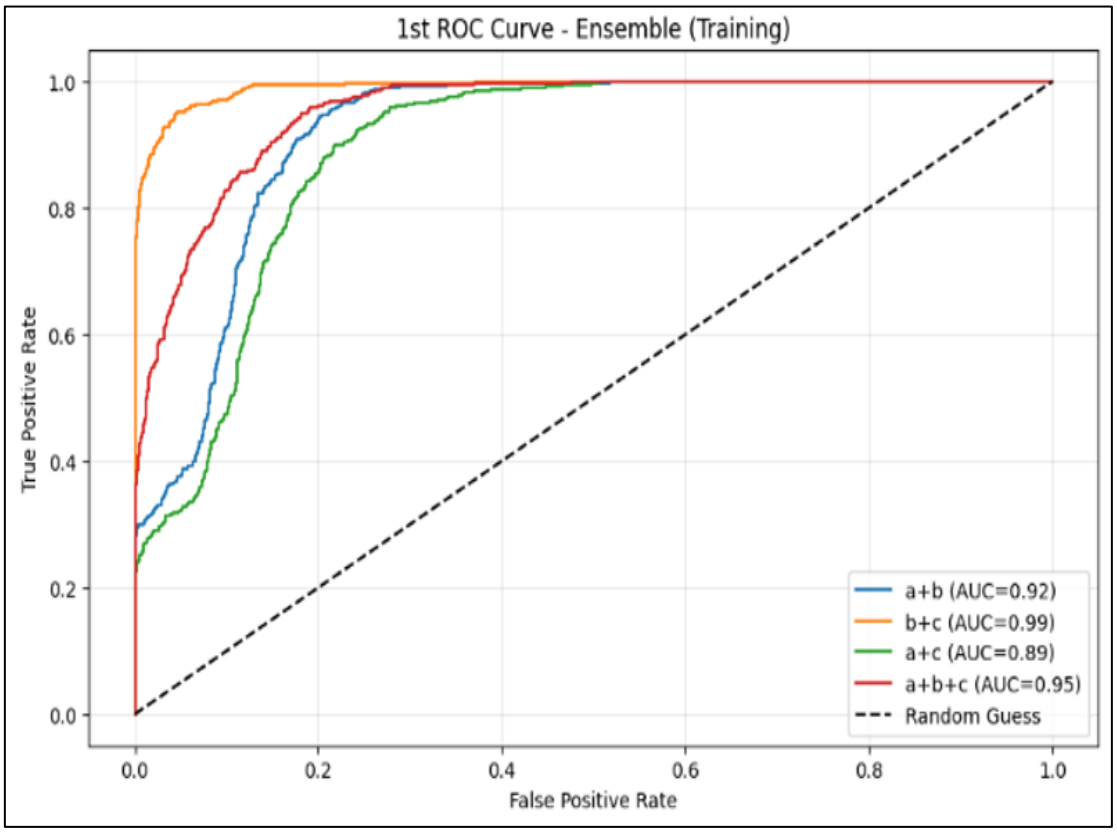

Fig 13 ROC Curve Training (Voting Ensemble model)

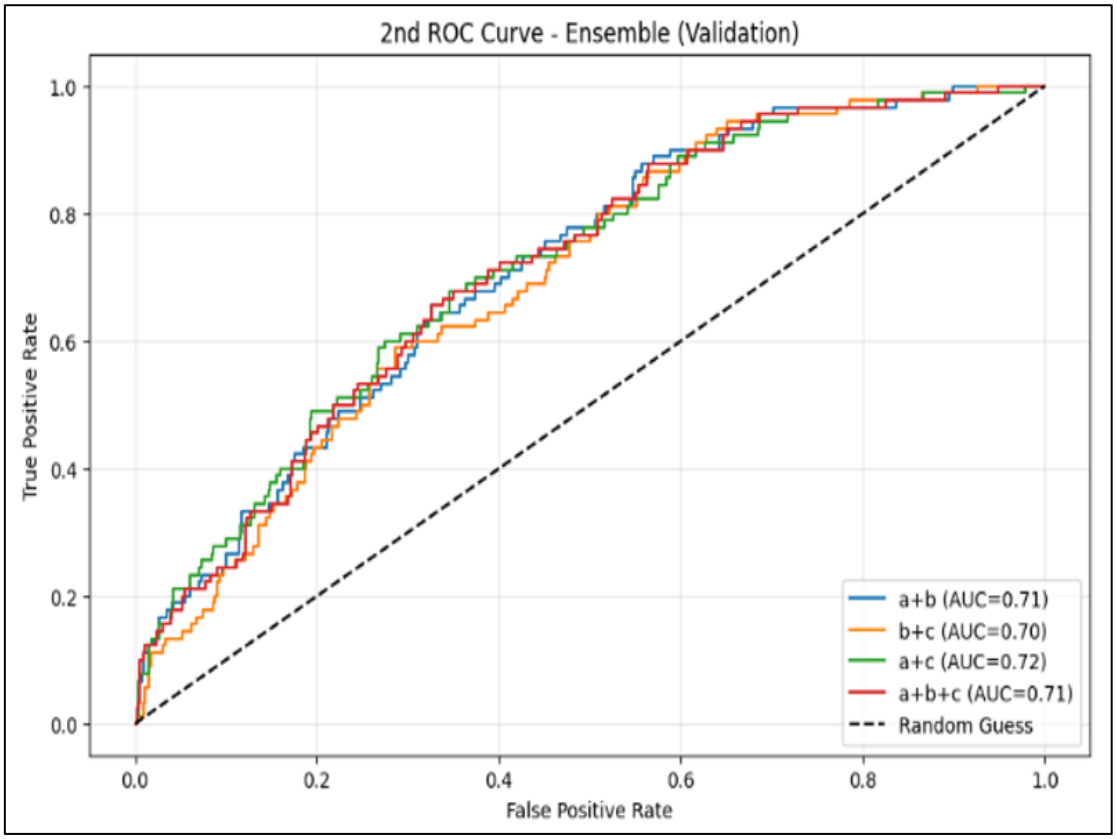

Fig 14 ROC Curve Validation (Voting Ensemble model)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

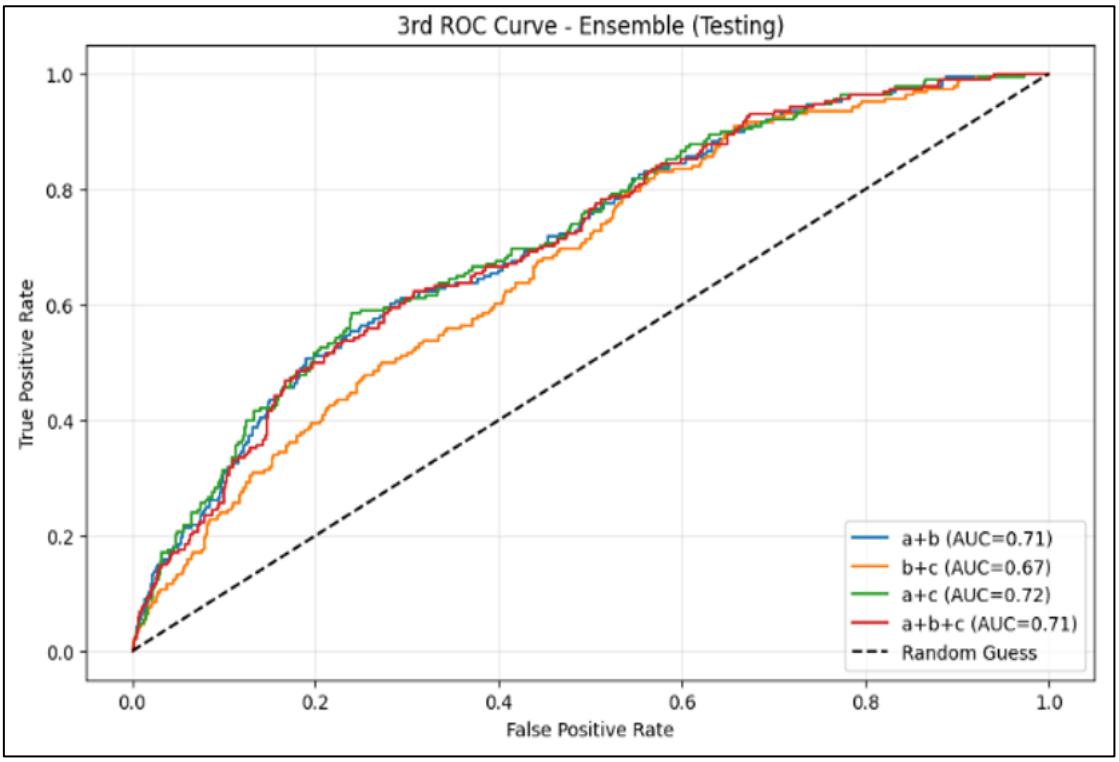

Fig 15 ROC Curve Testing (Voting Ensemble Model)

Fig 13 illustrates an ensemble framework learning curves & classification baseline across different training sessions. Fig 14 Provides metrics limits on the validation partition for the combined pipeline. Fig 15 shows about Testing the performance of Voting Ensemble model. For Final Testing ability of integrated model [42] High AUC here compared to individual models indicates that the ensemble strategy develops a better and more reliable classification system

➢ *Machine Learning Performance*

- *Performance Evaluation Metrics*

We quantified and compared the performance of nine baseline machine learning models and four voting ensemble classifiers using five statistical metrics. And These different matrices then Comes from the matrix components such as True Positives (TP), True Negatives (TN) False Positives(FP), False Negatives(FN). Those attributes of the FS can also be calculated mathematically as

- *Accuracy*

Accuracy is the number of correctly predicted components like TP and TN divided by the total number of data points evaluated in dataset.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Precision*

Precision is an indicator of how exact the classifier were by asking how many true positive instances are there out of all cases where the model predicted a positive instance.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- *Recall (Sensitivity)*

Recall measure the completeness of the model. It detects how many actual positive components were detected by the Classifier.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- *F1-Score*

F1-Score is the weighted average of Precision and Recall/Sensitivity It also offers a balanced metric which considers not only false positives but also false negatives, which is very relevant in the case of imbalanced distributions.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- *Mean Squared Error (MSE)*

It find out the mean difference between actual value and the estimated value mechanism, with lower better indicating the variance of error in this case model performance.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

- *Machine Learning Model Performance*

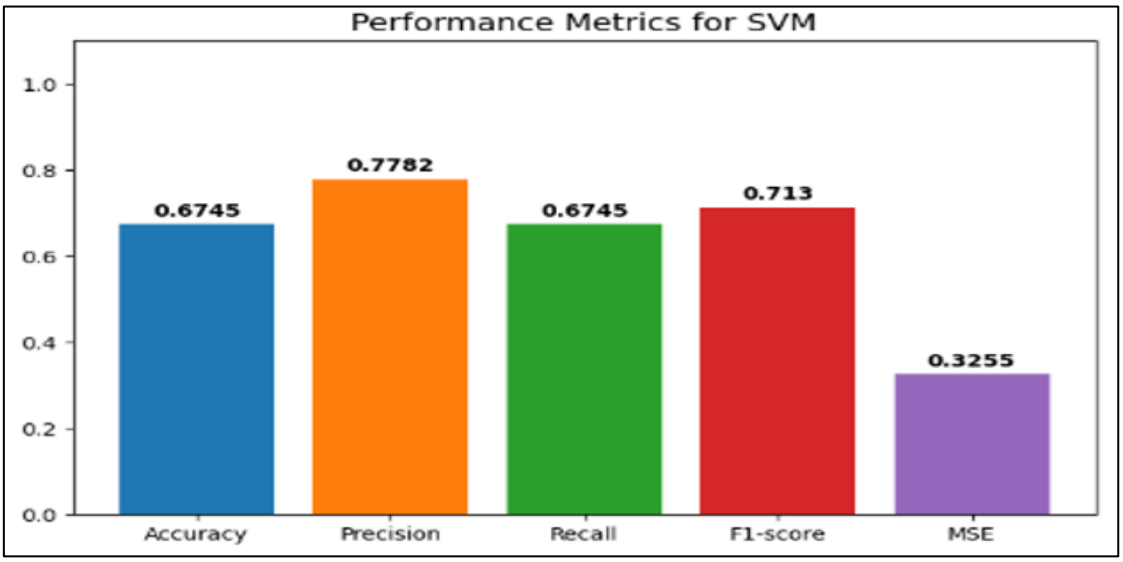

Fig 16 SVM Performance

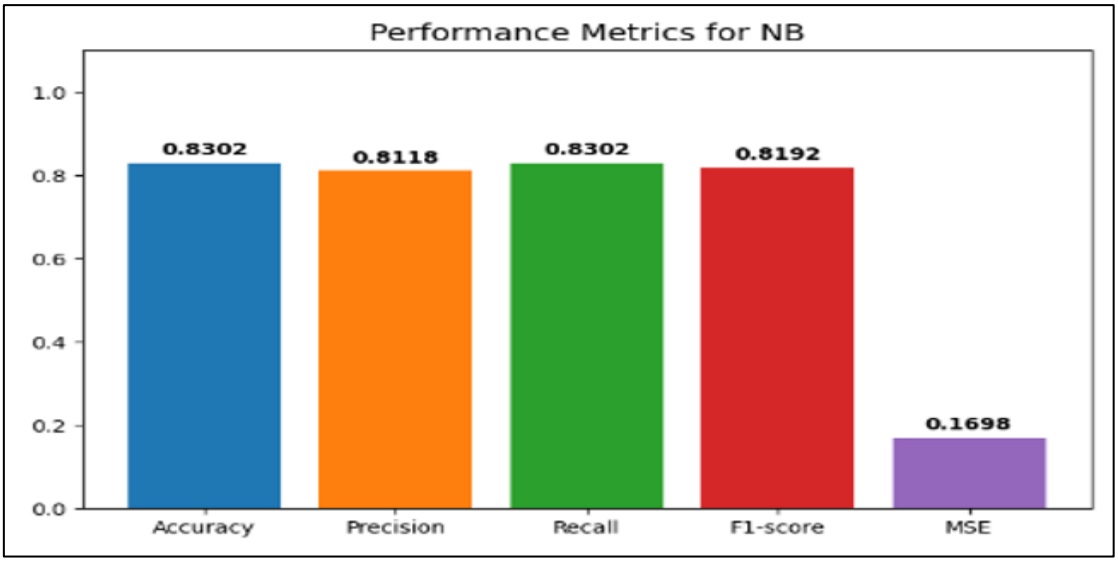

Fig 17 NB Performance

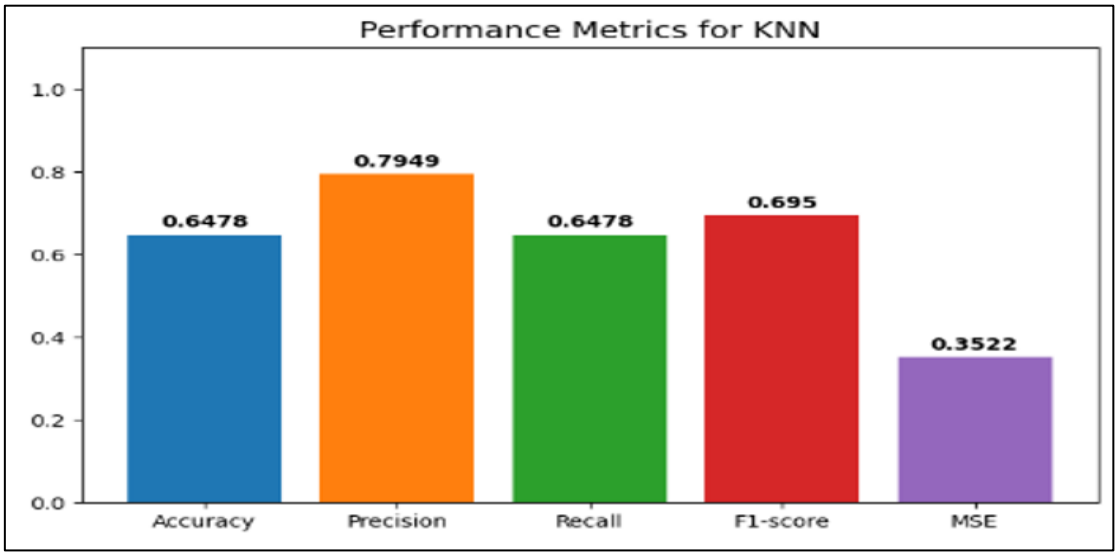

Fig 18 KNN Performance

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

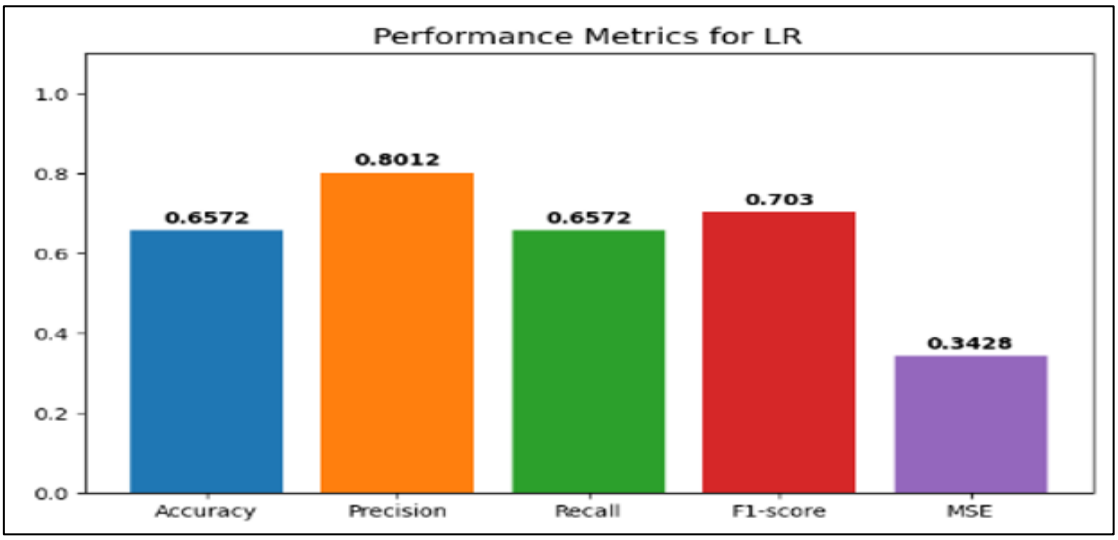

Fig 19 LR Performance

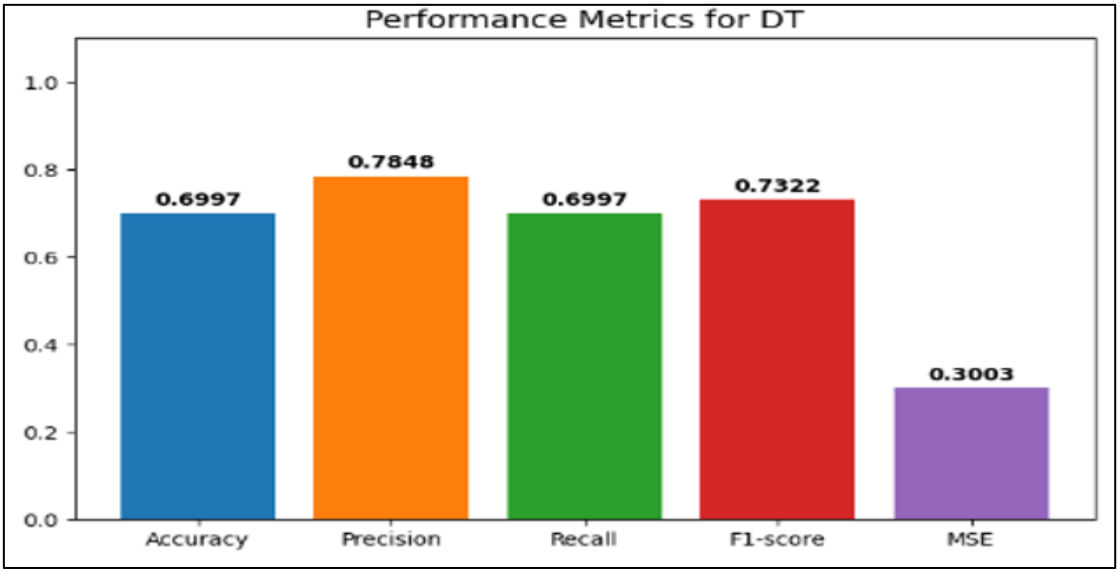

Fig 20 DT Performance

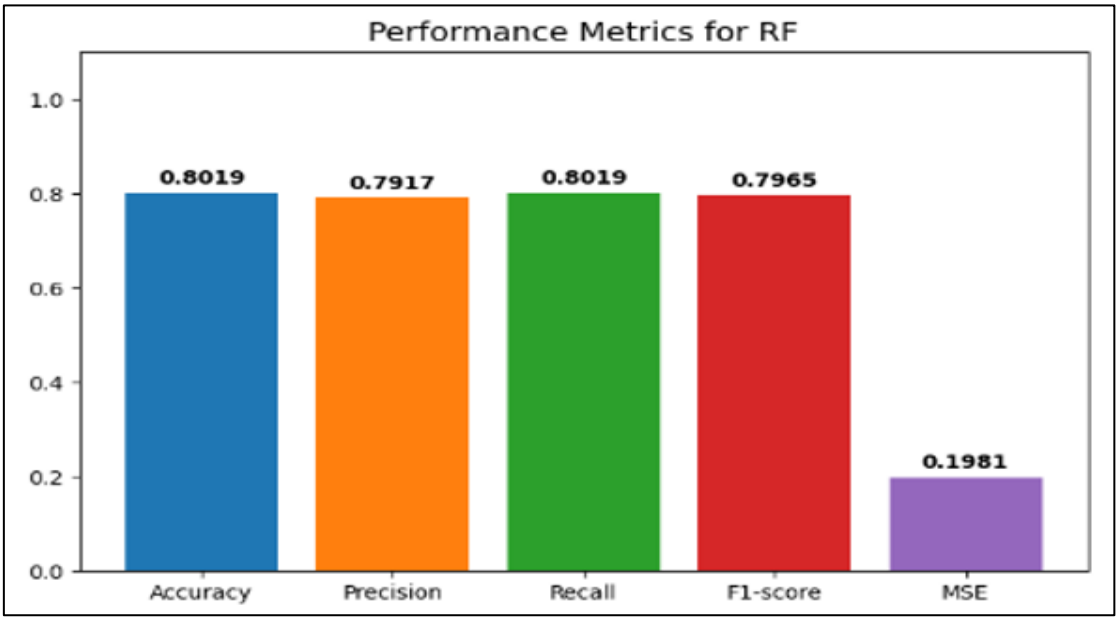

Fig 21 RF Performance

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

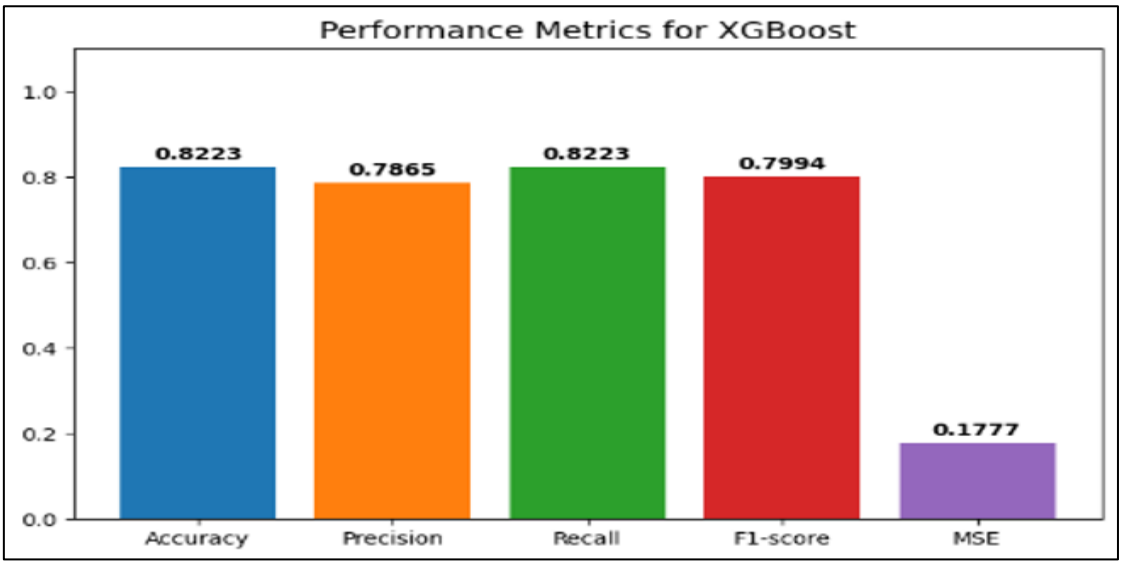

Fig 22 XGboost Performance

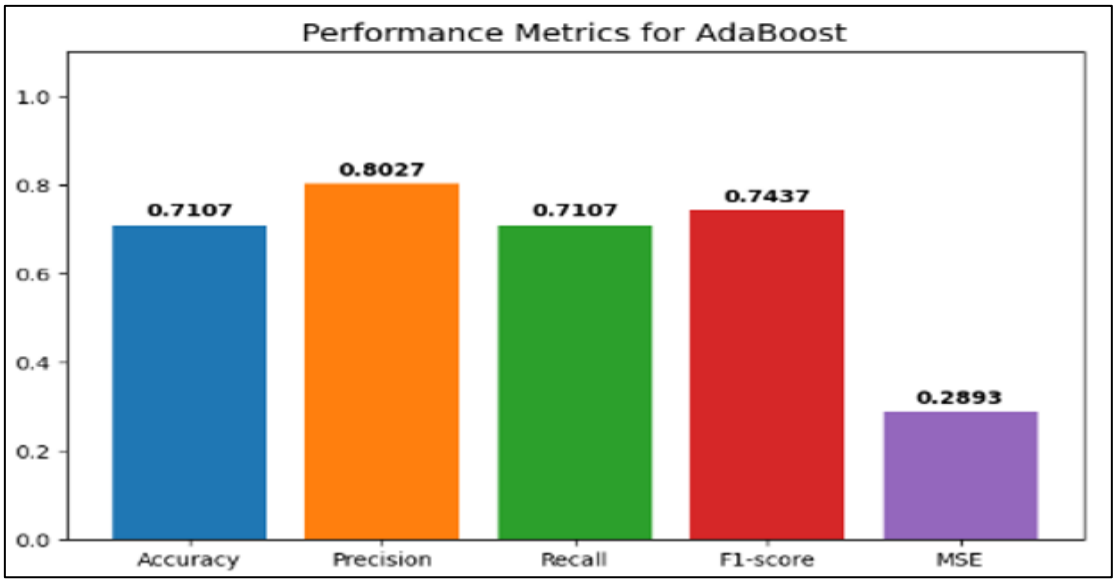

Fig 23 Adaboost Performance

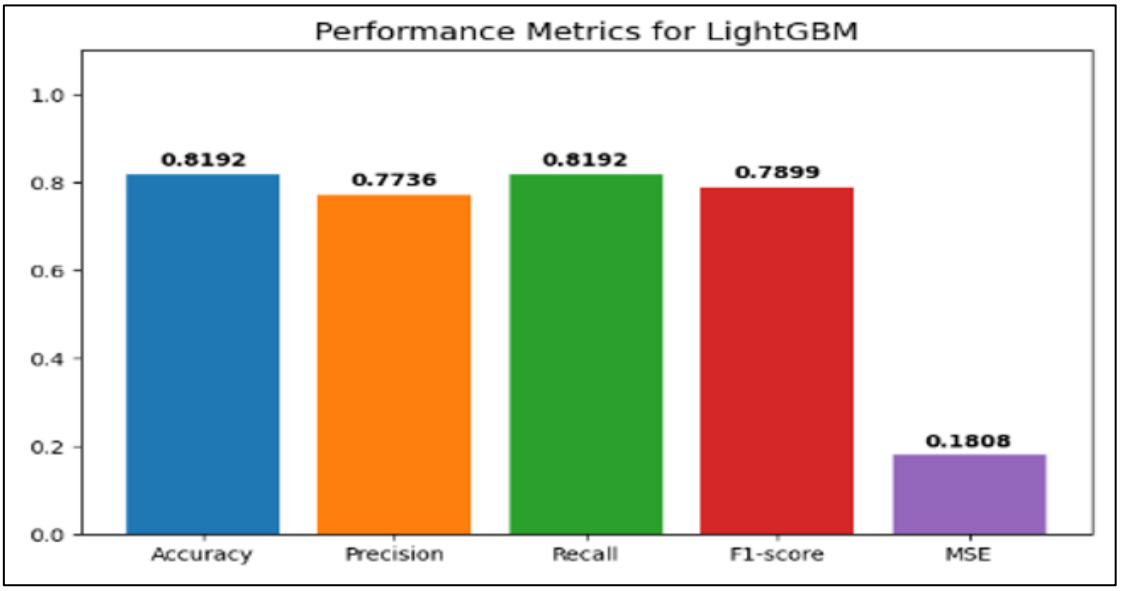

Fig 24 LightGBM Performance

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

Fig 16 to Fig 24 (Performance for all the base algorithms(SVM, NB, KNN, LR,DT, RF,XGBoost, AdaBoost)): Individual evaluation plots of base algorithm Dive into the details and we find that the standalone models give us Naïve Bayes (NB) out in front at 83.0189% accuracy with an MSE of only 0.169811, as indicated below. XGBoost follows next with an accuracy score of 0.822327 and a MSE of 0.177673. In contrast, predictive algorithms such as KNN and Linear Regression (LR) also produce poor results with a minimal increase in performance but lower accuracy (0.647799 and 0.657233) along with higher error rates (MSE of 0.352201 and 0.342767 respectively).

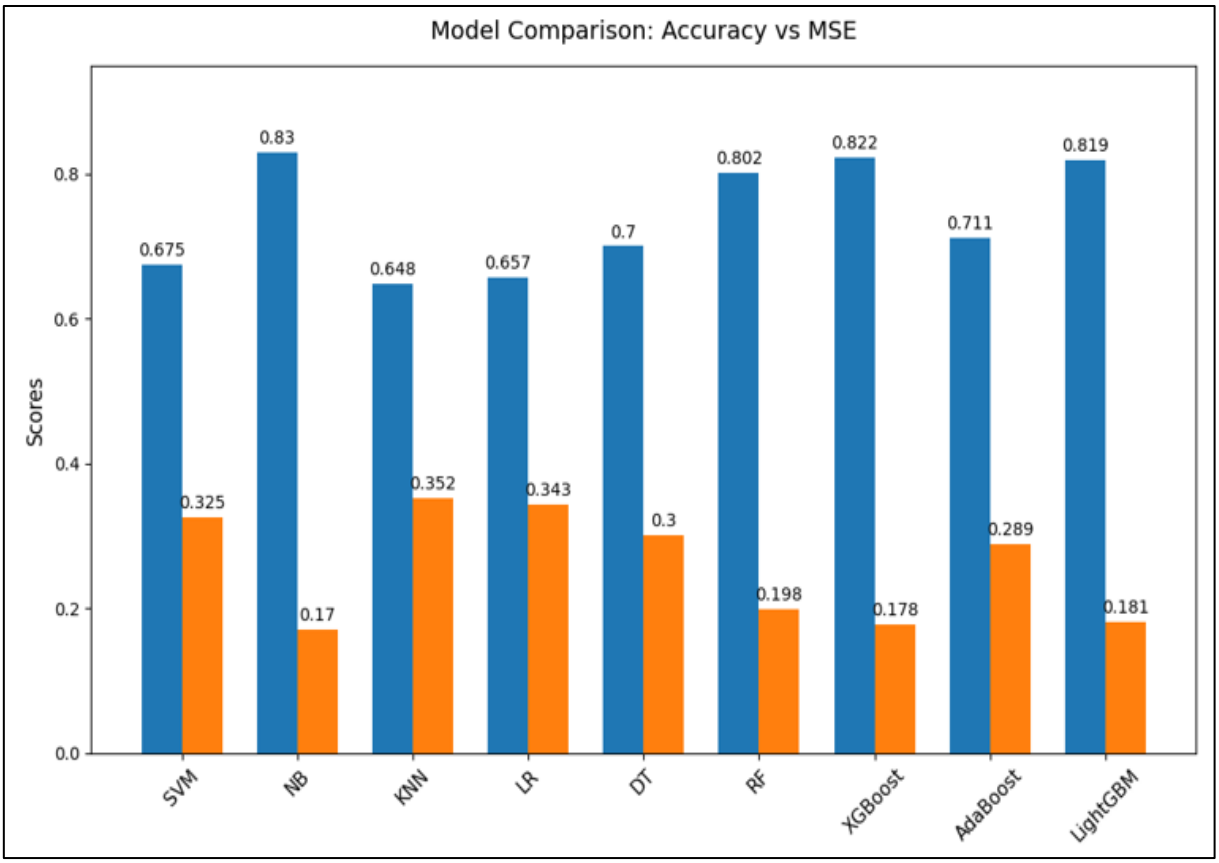

Fig 25 All Base Model Performance Comparison

Fig 25 (All Base Model Performance Comparison): A summary comparative chart showing the respective Accuracy, Precision, Recall, F1-score and MSE values for nine base models consolidated side-by-side. It makes it clear that ensemble-based architectures (RF, XGBoost, LightGBM) and NB win against simple distance or linear boundaries (KNN, LR) for this classification tasks.

- *Summary Table*

Table 3 Machine Learning Model Performance

Model	Accuracy	Precision	Recall	F1-score	MSE
SVM	0.674528	0.778163	0.674528	0.713012	0.325472
NB	**0.830189**	0.811766	0.830189	0.819206	0.169811
KNN	0.647799	0.794926	0.647799	0.695009	0.352201
LR	0.657233	0.801219	0.657233	0.703008	0.342767
DT	0.699686	0.784833	0.699686	0.732227	0.300314
RF	0.801887	0.791748	0.801887	0.796519	0.198113
XGBoost	**0.822327**	0.786525	0.822327	0.799419	0.177673
AdaBoost	0.710692	0.802712	0.710692	0.743733	0.289308
LightGBM	**0.819182**	0.773619	0.819182	0.789918	0.180818

Here we summarize all model performance in a table (Table 3).

- *Voting Ensemble Performance:*

  Here we shown the breakdown of a couple of comparative performance parameters for our combined frameworks:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

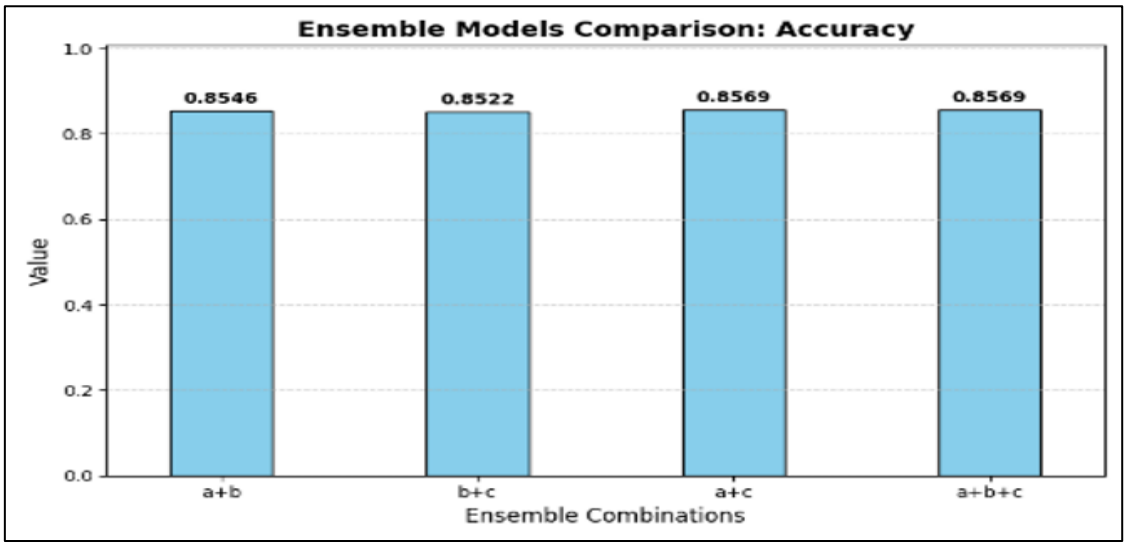

Fig 26 Voting Ensemble Accuracy Comparison

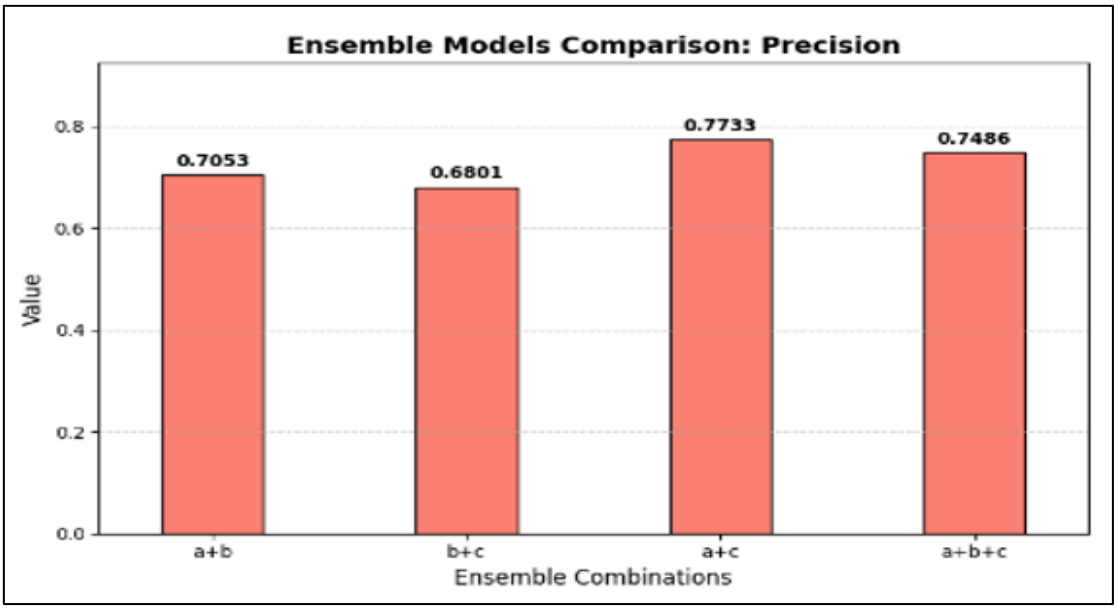

Fig 27 Voting Ensemble Precision Comparison Performance

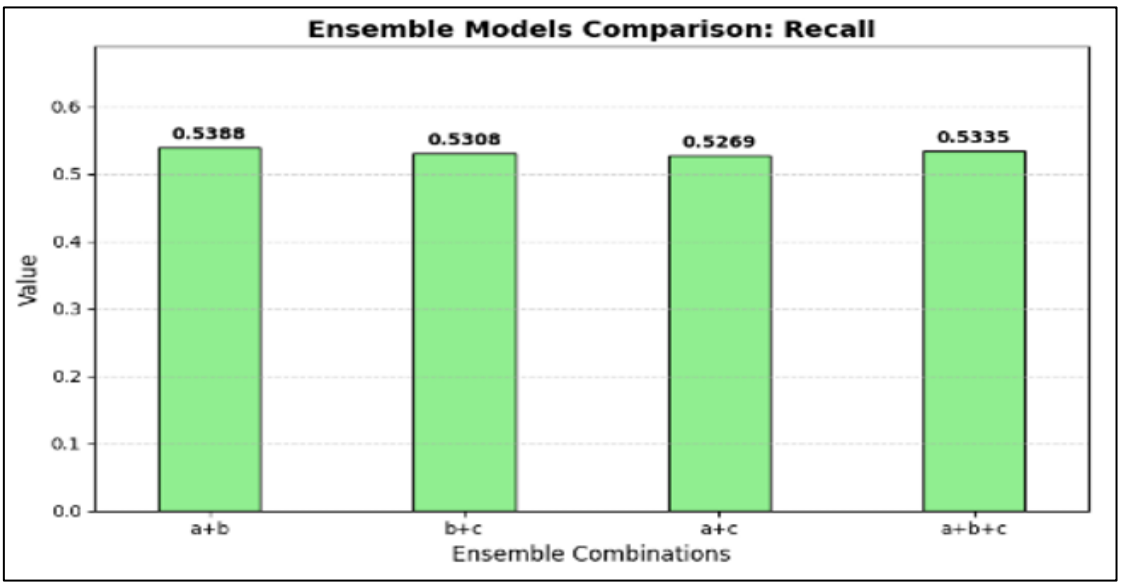

Fig 28 Voting Ensemble Recall Comparison

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

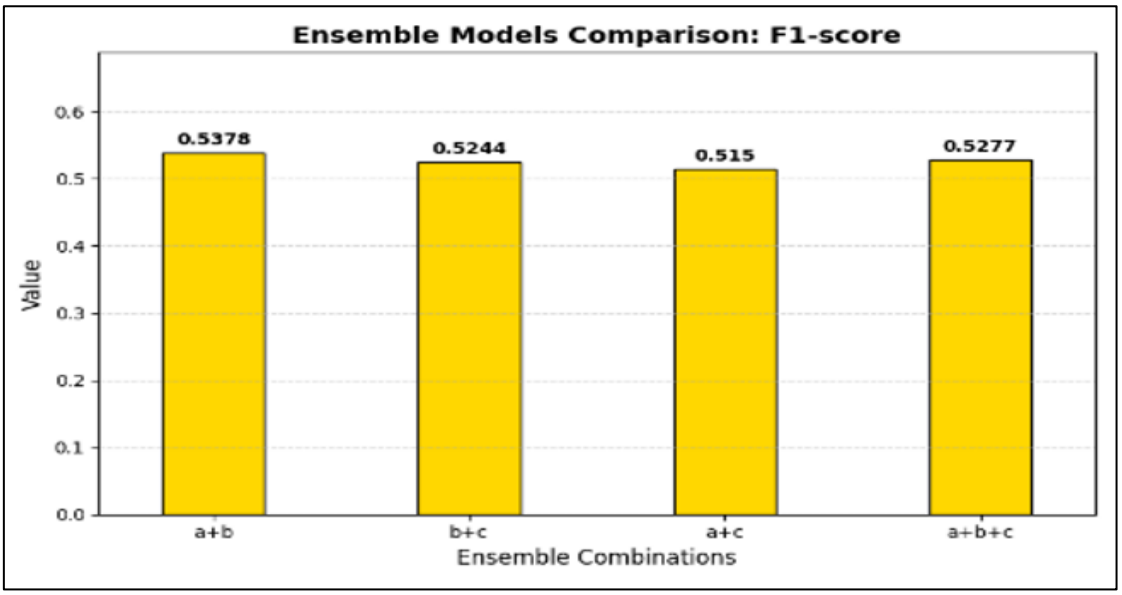

Fig 29 Voting Ensemble F1-Score Comparison Performance

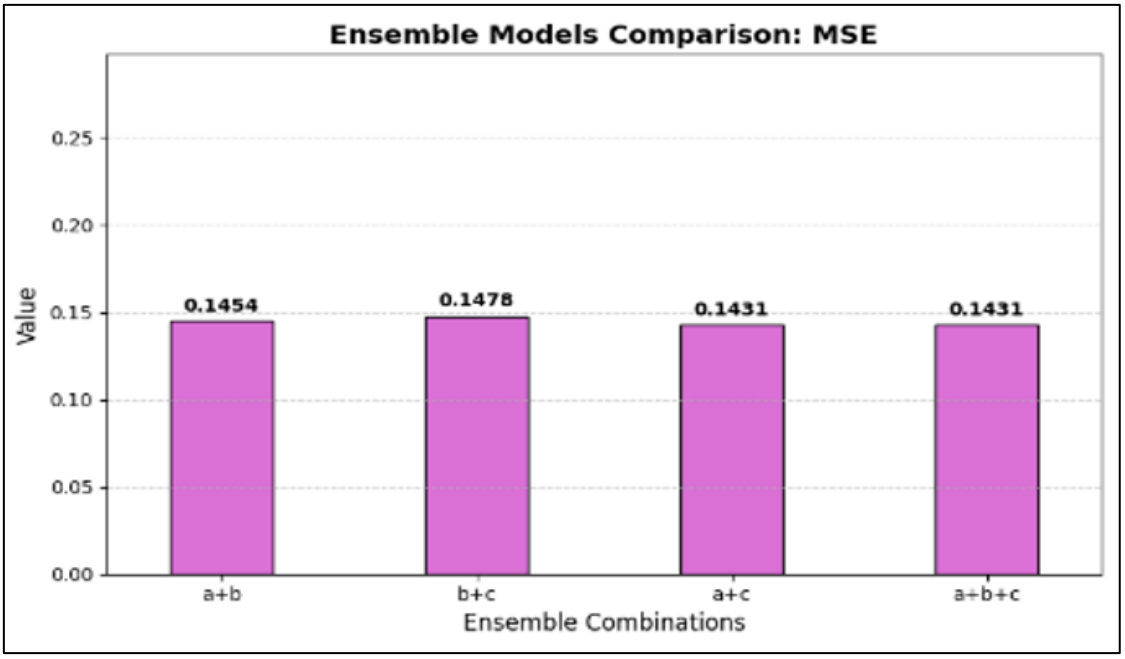

Fig 30 Voting Ensemble MSE Comparison

Fig 26 Graphically compares the accuracy levels from the combinations a+b (0.8546), b+c (0.8522), a+c (0.8569) and the complete ensemble a+b+c (0.8569). This indicates that aggregating models brings the overall classification accuracy to 85.69%, higher than any single model baseline (83.01%). Also Fig 27 Compares precision (for positive predictions) across voting ensembles, where the 'a+c' blend results in the highest precision (0.7733).Compare Voting Ensemble Recall (Fig 28) & F1-score (Fig 29). The combined retrieval bounds where the combinations relatively upwards hover on a stable constant from 0,51 to 0,53. On Fig 30. The variance is greatly diminished by use of the ensemble in voting mode. However, although the best single base model was 0.169811 (NB), the Mean Squared Error is greatly reduced to 0.1431 by applying the ensemble system (both a+c and a+b+c). This model arrives at a well optimised and stable architecture that correctly predicts with lower error ensemble voting across the board.

Table 4 Voting Ensemble Model Performance

Model	Accuracy	Precision	Recall	F1-Score	MSE
a+b	0.8546	0.7053	0.5388	0.5378	0.1454
b+c	0.8522	0.6801	0.5308	0.5244	0.1478
a+c	0.8569	0.7733	0.5269	0.515	0.1431
a+b+c	0.8569	0.7486	0.5335	0.5277	0.1431

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

Here,
a = NB
b = XGBoost
c = LightGBM
a+b = NB + XGBoost
b+c = XGBoost + LightGBM
a+c = NB + LightGBM
a+b+c = NB + XGBoost + LightGBM

- *Best Model Summary Table*

Table 5 Best Model (Based on Accuracy)

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**MSE**
a = NB	0.8301	0.8117	0.8301	0.8192	0.1698
b = XGBoost	0.8223	0.7865	0.8223	0.7994	0.1776
c = LightGBM	0.8191	0.7736	0.8191	0.7899	0.1808
a+b	0.8546	0.7053	0.5388	0.5378	0.1454
b+c	0.8522	0.6801	0.5308	0.5244	0.1478
a+c(Proposed Model)	**0.8569**	**0.7733**	0.5269	0.515	0.1431
a+b+c	0.8569	0.7486	0.5335	0.5277	0.1431

After examining the empirical tracking in Table 5 — a+c Voting Ensemble is suggested as an ideal framework for this research. Although we see that individual algorithms have reasonable baseline performance capability—where Naïve Bayes outperforms the other isolated algorithms with an accuracy of 0.830189 and a Mean squared errors (MSE) of 0.169811—it is still limited by its isolated architectures. To beat these individual performance limits, a bunch of synchronized voting networks were carried out from the joined sub-models.

In the ensemble configurations, both a+c and a+b+c get to the best global accuracy of 0.8569 while managing to reduce error variance down to the absolute lowest MSE of 0.1431.

The only combination being proposed which give a better optimization profile is for the case of a+c! It allows us to obtain a much better Precision score of 0.7733 compared to the 0.7486 on the full network, which is essential for avoiding expensive false positives misclassification injuries. In addition, by leveraging just two sub-models instead of three, a+c not only alleviates redundant structure and computational costs but also achieves the most accurate and generalized decision boundary.

V. DISSCUSSION

The experiments indicate that there is a substantial variation in the effectiveness of these machine learning models. The accuracy results across classifiers showed that Naïve Bayes outperformed all individual classifiers with an accuracy of 83.02% followed by XGBoost and lightGBM which achieved accuracies of 82.23 and 81.92 (%) respectively. This was done to check how can the prediction accuracy is able to predict better test set with count functions of each user Traditional classifiers such as Logistic regression, KNN and SVM gave lower prediction accuracies in comparison.

The improved performance of gradient boosting methods indicates that they can model relationships between medical risk factors that are not linear. Furthermore, the use of ANOVA F-score feature selection contributed to discard irrelevant features and enhanced the computational efficiency.

The adoption of the ensemble learning method provided higher classification performance. Together, the three best models correctly classified 85.69%, which was an improvement over any of the individual classifiers. It is a good example that contextual diversity in prediction mechanism and training data can help reduce variance in the prediction task and achieve higher generalization performance.

Global feature importance from SHAP according interpretable view clearly showed that where we can identify the main clinical factors which might drive to stroke prediction. LIME helped in this regard by providing local explanations for predictions on a per-patient basis. Such explainability level is very important in healthcare settings, where accountability and transparency are key factors for adoption in clinics.

Conclusion — In general, the results demonstrate that a combination of explainable AI and ensemble machine learning constitutes a viable and reliable solution for heart stroke prediction.

VI. CONCLUSION

Feature selection, imbalance handling, ensemble learning and XAI techniques introduced an Explainable Machine Learning Framework in this study for Heart Stroke Prediction. There were 4239 records of patients in the dataset. Twelve machine learning algorithms were experimented with. On the basis of experimental results, the best classifiers

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

(individual algorithms) were XGBoost, LightGBM and Naïve Bayes, respectively.

The best accuracy for the ensemble voting model is 85.69% using top three classifiers and that gives a better prediction than standalone models. Finally, explanations using SHAP and LIME also increased transparency, offering both global and local insight into model decisions.

The proposed framework is a strong and interpretable approach to early prediction of heart stroke risk, which can help the doctor with clinical decision making. Going forward, larger multi-center datasets and possibly deep learning architectures may be considered to improve performance and generalizability of prediction, incorporating federated learning settings or even clinical real-time deployment.

REFERENCES

[1]. S. Aljanabi, M. Al-Shargabi, and A. Al-Madi, "Performance evaluation of machine learning algorithms for heart disease prediction," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 649–656, 2020, doi: 10.14569/IJACSA.2020.0110115. [Online]. Available: https://thesai.org/Publications/ViewPaper?Volume=11&Issue=10&Code=IJACSA&SerialNo=15

[2]. R. Vignesh, S. P. Rajamhoana, and C. R. Vignesh, "A machine learning approach for 10-year CHD risk prediction using Framingham dataset," in *Proc. IEEE Int. Conf. Smart Syst. Inventive Technol.*, 2022, pp. 1432–1439, doi: 10.1109/ICSSIT.2022.9716234. [Online]. Available: https://ieeexplore.ieee.org/document/9716234

[3]. A. K. Jha, A. Mehta, and N. S. Raghava, "Comparative analysis of machine learning techniques for cardiological stroke and CHD prediction," in *Proc. Int. Conf. Innov. Comput. Commun.*, 2021, pp. 341–352, doi: 10.1007/978-981-16-2594-7_32. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-16-2594-7_32

[4]. J. P. Li, M. U. Haq, and S. U. Din, "Implementation of heart disease prediction system using minimal clinical parameters," *J. Med. Syst.*, vol. 43, no. 9, p. 289, 2019, doi: 10.1007/s10916-019-1402-7. [Online]. Available: https://link.springer.com/article/10.1007/s10916-019-1402-7

[5]. M. G. R. Alam, M. S. Uddin, and A. S. M. L. Hoque, "Impact of feature reduction on the efficiency of cardiac disease prediction models," *Informatics Med. Unlocked*, vol. 25, p. 100678, 2021, doi: 10.1016/j.imu.2021.100678. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914821001430X

[6]. K. Srinivas, B. K. Rani, and A. Govardhan, "Application of data mining techniques in healthcare for CHD risk assessment," *Int. J. Comput. Appl.*, vol. 180, no. 38, pp. 18–24, 2018, doi: 10.5120/ijca2018917321. [Online]. Available: https://www.ijcaonline.org/archives/volume180/number38/srinivas-2018-ijca-917321.pdf

[7]. Y. Khourdifi and M. Bahaj, "Predicting ten-year risk of coronary heart disease using sparse medical data," *IEEE Access*, vol. 8, pp. 99102–99111, 2020, doi: 10.1109/ACCESS.2020.2991102. [Online]. Available: https://ieeexplore.ieee.org/document/8991102

[8]. T. Karayilan and O. Kilic, "Critical evaluation of cardiovascular risk prediction models: A case study on Framingham data," *J. Biomedical Informatics*, vol. 99, p. 103310, 2019, doi: 10.1016/j.jbi.2019.103310. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002197381930310X

[9]. N. K. Kumar, G. S. Kumar, and V. R. S. Mani, "Baseline machine learning classifiers for heart disease prognosis with missing data analysis," in *Proc. IEEE Int. Conf. Comput., Commun. Intell. Syst.*, 2021, pp. 412–417, doi: 10.1109/ICCCIS.2021.9687114. [Online]. Available: https://ieeexplore.ieee.org/document/9687114

[10]. S. Radhimeenakshi, S. Chidambaranathan, and R. Selvakumar, "Heart risk prediction using non-invasive clinical features: A machine learning approach," in *Proc. Lecture Notes Electr. Eng.*, 2020, pp. 451–460, doi: 10.1007/978-981-15-3242-9_45. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-15-3242-9_45

[11]. G. N. Ahmad, S. Shafi, and M. Z. M. J. Khan, "Comparative study of heart disease prediction with diminished feature subsets," *Int. J. Eng. Advanced Technol.*, vol. 9, no. 1, pp. 2321–2325, 2019, doi: 10.35940/ijeat.A1143.109119. [Online]. Available: https://www.ijeat.org/wp-content/uploads/papers/v9i1/A1143109119.pdf

[12]. M. M. Ali, B. K. Paul, and M. A. Moni, "Evaluation of classification algorithms for 10-year risk of CHD using reduced variance features," *Diagnostics*, vol. 12, no. 8, p. 1844, 2022, doi: 10.3390/diagnostics12081844. [Online]. Available: https://www.mdpi.com/2075-4418/12/8/1844

[13]. P. S. Kohli and S. Arora, "Simplified machine learning models for early detection of cardiovascular risk," in *Proc. IEEE Int. Conf. Emerging Smart Comput. Informatics*, 2022, pp. 1–6, doi: 10.1109/ESCI53501.2022.9758241. [Online]. Available: https://ieeexplore.ieee.org/document/9758241

[14]. R. S. S. Kumari, S. N. Deepa, and K. V. S. R. Prasad, "Feature selection impact on machine learning-based coronary heart disease prediction," in *Proc. Comput. Commun. Inf. Sci.*, 2020, pp. 125–136, doi: 10.1007/978-3-030-63393-6_12. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-63393-6_12

[15]. H. Benjamin, S. S. Babu, and S. Jeeva, "Baseline exploration of the Framingham cohort using simple linear and tree classifiers," *Int. J. Inf. Technol.*, vol. 13, no. 3, pp. 1141–1148, 2021, doi: 10.1007/s41870-021-

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun414

00741-w. [Online]. Available: https://link.springer.com/article/10.1007/s41870-021-00741-w Part 4: Documentation & Learning Resources

[16]. Google ref: https://www.geeksforgeeks.org/machine-learning/auc-roc-curve/

[17]. Scikit Learn Voting Classifier Docs: Learn how you can combine several models directly in Python with soft or hard voting: [Scikit-Learn Ensemble Voting] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

[18]. WHO ref: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

[19]. Dataset Available : https://www.kaggle.com/datasets/mirzahasnine/heart-disease-dataset