iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 12 **Issue:** III **Month of publication:** March 2024

**DOI:** https://doi.org/10.22214/ijraset.2024.58888

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 12 Issue III Mar 2024- Available at www.ijraset.com*

# An Efficient Ensemble Machine Learning Model for Cardiovascular Disease Prediction Using Digital Health Records

B. Sharanya[1], V. Srividya[2], B. Prajnaya[3], Bandari Gayathri[4]
[1, 2, 3]*UG Student Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*
[4]*Assistant Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana*

***Abstract:*** ***Detecting cardiovascular problems early is crucial for timely treatment. In our study, we employed machine learning to analyze a diverse set of information about individuals' lives and health, to predict cardiovascular disease. Ensuring data accuracy and addressing missing information were prioritized in our approach. Experimenting with different solo ML & ensemble ML methods, comprised of Random Forest and XGBoost with tuning, we achieved a notable 92% accuracy in identifying potential heart issues. Remarkably, combining multiple machine learning methods through ensemble learning proved even more effective than individual methods. Expanding our methodology to include Light GBM, Extra Tree, Decision Tree, SVM, Naive Bayes, QDA, & Adaboost enhanced the comprehensiveness of our analysis. Additionally, delving into ensemble learning methods such as bagging, boosting, tuning, & stacking further pushed the boundaries of predictive accuracy. In essence, our research outstands the potency of diverse ensemble machine-learning techniques and algorithms in early cardiovascular prediction. Ensemble methods, which combine different algorithms, emerged as powerful tools without relying on complex terminology.***

***Keywords: Cardiovascular Disease (CVD), Artificial Intelligence(AI), Machine Learning (ML), Deep Learning (DL), Ensemble Learning, Heart disease.***

## I. INTRODUCTION

As per WHO CVD is the deadliest disease almost a third of all deaths are caused by cardiovascular disease it takes over 17 million lives over a year. In the year 2023, it took over 20.5 million lives [1]. Advanced knowledge can help us to prevent deaths caused by cardiac arrest. To identify cardiac diseases there are symptoms like chest pain, chest pressure, shortness of breathing, and fainting. Heart diseases are identified by clinical examinations, medical history, and diagnostic tests. Cardiovascular disease (CVD) is a disease that affects on heart and blood vessels. There are distinct types of cardiovascular diseases. They are Coronary heart disease, Peripheral heart disease, stroke, and Heart failure. Cardiovascular disease is caused due raised blood pressure, high cholesterol levels, lack of exercise, diabetes, smoking, tobacco, and improper Diet. Healthcare professionals or providers can use Artificial Intelligence to detect heart diseases. Leads to diagnosis at the advanced stage of the disease. Artificial Intelligence is predominantly used in identifying cardiovascular disease. In the presented study we hypothesize that ensemble, ML functionalities are superior to solo and deep learning algorithms due to the scarcity of available clinical data. This paper is structured across 5 different modules. module 2 concisely scrutinizes literature related to CVD prediction using ML and DL algorithms. module 3 describes the materials and methods used in the prediction of CVD using ML and DL algorithms. module 4 embraces the results. In module 5 we present a comparative study of implemented model with the state-of-art techniques for predicting CVD.

## II. LITERATURE REVIEW

In this literature review, we explore the latest research on machine learning-based approaches for CVD prediction. We examine the different machine learning algorithms employed in CVD risk prediction, the datasets used for model development and validation, evaluation metrics for assessing model performance, and recent advancements in the field. By synthesizing findings from the literature, we attempt to contribute insights into the potential of machine learning to revolutionize CVD risk stratification and inform personalized preventive strategies in clinical practice.

Muktevi Sri Venkatesh et al[2] In their discussion of the early methods for forecasting cardiovascular illness, suggested a prediction using an SVM, Naïve Bayes classifier, Random Forest RF, and logistic regression. In comparison to other machine learning algorithms, logistic regression has a higher accuracy rate (77.06%), according to his research.

©IJRASET: All Rights are Reserved 

Ramdas Kapila et al[3] introduced the Mcclusky Binary Classifier (QMBC) model, which makes use of an ensemble of seven models, an adopted set of machine learning
techniques. They all used ANOVA and chi-square methods to determine the top ten traits. The Cleveland dataset and the CV dataset were used in this investigation. It provided a 98.36% accuracy rate.

Liaqat Ali et al [4] focus on issues related to overfitting and underfitting. The chi-square statistical model, which they devised, is used to remove unnecessary features when searching with an exhaustive search method for the best-configured deep neural network. The presented mixed diagnostic system's temporal complexity was not examined in this study. The expected accuracy of the suggested model is 93.3%.

Hemantha Kumar Kalluri et al [5] implemented a model for disease forecasting that leverages Convolutional Neural Networks (CNNs) to achieve predictive accuracy. Comprising two convolution layers, 2 dropout layers, and a single resulted layer, this model demonstrates a reported accuracy of 94.78%. Through the strategic integration of convolutional layers, the model captures hierarchical patterns in input data, crucial for disease prediction. Dropout layers enhance model generalization by reducing overfitting, while the output layer synthesizes learned features into actionable predictions. With its high accuracy, this CNN-based model presents a promising avenue for disease forecasting, offering valuable insights into potential diagnoses and treatment strategies.

Mayank Sharma et al [6] proposed a model that gains performance in detecting cardiac disease. They used a Tree classifier, hybrid CNN, and Bi-LSTM to predict heart illness. In this study, comparative studies are also examined. This approach is capable of producing an accuracy of 96.66%.

James Meng et al [7] presented the first clinical knowledge-enhanced ML model for predicting IHD. They included key steps such as statistical analysis, preprocessing, feature selection, and model learning evaluation. This model based on SVM achieved an accuracy of 94.4%.

Abu Yazid et al[8] researched the ANN and used the Cleveland dataset to achieve 90.9% of accuracy and also worked with the statlog dataset and achieved an accuracy of 90%

B.B.Gupta et al [9] created a model for accurately predicting cardiovascular illness. IoT models and machine learning were employed. With an accuracy of 87.72%, it seems that more complicated classifiers, such as SVM and Random Forest, produced superior results.

Ankur Gupta et al [10] suggested a framework for machine intelligence MIFH is used to diagnose heart disease. They suggested a framework called MIFH, which can be utilized to forecast the occurrences of either heart patients or normal persons. Their sensitivity was 92.8%, compared to 89.28% for MIFH.

Talha Javed et al [11] suggested deep learning and machine learning methods based on ensembles to forecast cardiovascular illness. The models' performance was evaluated based on how accurate they were. Their accuracy rate was 88.70%.

Pradhan et al [12] considered the UCI repository dataset and five methods (support vector machine, logistic regression,
main component analysis, multi-layer perceptron classifier, and achieved approximately 90% accuracy.

Vicky Singh et al [13] used machine learning algorithms in this examination, and a recommendation system based on variables like age, blood pressure, and so on was presented. They concluded that SVM and decision tree classifiers provide 85% accuracy.

R. Karthikeyan et al [14] suggested utilizing a convolution neural network and deep learning to predict cardiovascular illness.

Uma Maheshwari et al[15] engaged a unique method for predicting cardiac disease by combining neural networks with logistic regression analysis. Initially, the foremost risk indicators for illness prediction are chosen using logistic regression. The statistical p-value is produced. With an accuracy of 84%, the combination of logistic regression and neural network is used to predict cardiac disease.

Senthil Kumar et al[16] By using machine learning approaches, created an excellent approach that improves the accuracy of cardiovascular disease detection by identifying important aspects. He achieved an improved performance level of 88.7% using HRFLM.

## III. MATERIALS & METHODS

### *A. Description of the Data Set*

In machine learning, data is paramount for accuracy. This collected dataset contains 19 variables of which 12 are arithmetical and 7 are categorical. The number of instances is 308854 and the dataset does not contain missing values.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 12 Issue III Mar 2024- Available at www.ijraset.com*

Table 1: CVD Dataset

| Serial Number | Attribute | Description |
|---|---|---|
| 1 | General Health | Well-being, fitness |
| 2 | Check-up | Examination to ensure health or wellness |
| 3 | Exercise | Activity for fitness and health. |
| 4 | Heart Disease | Cardiac condition |
| 5 | Skin_cancer | Describe different parts of your skin or conditions that can affect it. |
| 6 | Other_Cancer | Those who indicated they have experienced any other forms of cancer |
| 7 | Depression | Feeling sad, hopeless, or down for a long time. |
| 8 | Diabetics | Having too much sugar in your blood for a long time. |
| 9 | Arthritis | Pain and swelling in your joints makes it hard to move. |
| 10 | Gender | 0-Female,1-Male |
| 11 | Age | In days |
| 12 | Height | In Cent Meter |
| 13 | Weight | Kilograms |
| 14 | BMI | It is a number that shows if a person is a healthy weight for their height. |
| 15 | Smoking History | Whether Patient Smokes or Not |
| 16 | Alcohol consumption | Whether patient smokes or not |
| 17 | Fruit consumption | Recording patients' fruit intake |
| 18 | Green Vegetables consumption | Recording patients' green vegetable intake |
| 19 | Friedpotato vegetable consumption | Recording patients' potato intake |

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

*B. Methods*

*1) Solo Machine Learning Algorithms*

[17 ]Algorithms for machine learning allow computers to recognize patterns and connections in data without the need for explicit programming. Based on input data, these algorithms employ statistical approaches to find patterns and provide predictions or choices. Three main categories can be used to group them

Supervised Learning: In this method, an algorithm is trained using a labeled dataset in which every input has a matching output. To forecast or categorize newly discovered data, the algorithm gains knowledge from this labeled dataset. Neural networks, decision trees, SVmachine, logistic regression, and linear regression are common techniques in supervised learning.

Unsupervised Learning: When there are no labels on the input data, unsupervised learning algorithms are applied. Without any indication, the algorithm looks for facts in the statistics. This learning comprises approaches, clustering algorithms like k-means, and hierarchical clustering. In unsupervised learning, dimensionality reduction methods like (SVD) and (PCA) are frequently employed.

Reinforcement learning: Teaching an agent through reinforcement learning involves guiding it to interact with its environment to maximize rewards. The agent learns through trial and error receiving feedback in the form of rewards or penalties based on its performance. Algorithms, like Q learning and deep Q networks (DQN) are commonly used in robotics, gaming and autonomous systems for reinforcement learning tasks.

*a) Logistic Regression*

In the domain of learning algorithms logistic regression stands out as a choice in machine learning. It is utilized for predicting outcomes based on a set of variables. Unlike regression, which is used for regression tasks logistic regression tackles classification challenges by providing values between 0 and 1 instead of definite values of 0 or 1. The core concept behind regression is fitting a S" shaped function instead of a straight line to predict binary outcomes that indicate the likelihood of an event occurring.

*b) Ridge Classifier*

To combat overfitting issues in machine learning models methods like Ridge Regression and Ridge Classifier are employed. Overfitting occurs when a model performs well on training data but poorly, on data.In Ridge Regression, an additional term known as L2 regularization is included in the linear regression equation to avoid overfitting. This term penalizes coefficients helping to manage the complexity of the model. Likewise a Ridge Classifier employs L2 regularization to prevent overfitting, in tasks involving class classification.

*c) Support Vector Machine*

Machine learning is principally built upon SVM, which is an abbreviation of Support Vector Machine. It can also be used as a regression tool, though classification is its main area. At its very essence, SVM looks for a decision horizon that splits n-dimensional space into well-defined groups; this decision horizon is usually referred to as a hyperplane. Thereafter, when new data points are encountered in the future, it will be easy to place them into different classes if there was proper interpretation of the hyperplane initially made. The selection of critical elements called support vectors that define the hyperplane is central to how SVM functions. Consequently, suport vectors are those which have great impact on where the line passes through and what direction it takes on the graph.In reality, these support vectors determine where the line goes and in what direction it would slant; thus giving birth to SV machine as its name suggests.The algorithm exhibits great efficiency in classifying data points when using these support vectors via optimization of the hyperplane itself.

*d) K-Nearest Neighbour*

Regression problems and classification tasks are addressed by utilizing K-Nearest Neighbors (KNN) algorithm within machine learning.The following sentence provides an explanation for KNN:Think about some points that you have plotted on a graph with each one being assigned either category or value.Whereas KNN examines the closest neighbors.Then a vote is taken (for classification) or averages (for regression) obtain their labels or values in order to find the label or value of the new point. KNN is an easy to use and adaptable technique making it applicable in different areas like pattern recognition, data mining and intrusion detection. Another thing why KNN is considered good because it is “non-parametric.” This means that it does not presume anything about how data are distributed.This flexibility makes it handy for real-life situations where data can be messy or irregular. To use KNN, you start with some known data (called training data), which has points already labeled. Then, when you get a new point, KNN compares it to the known points and makes predictions based on their proximity.

*e) Decision Tree*

Although it is frequently used for classification, a DT is a useful tool in supervised learning; it would handle both regression & classification tasks. It resembles a tree-shaped flowchart in which decisions are made at each stage depending on a particular data attribute.

Nodes: Decision nodes and leaf nodes are the 2 foremost types. Decisions are taken at decision nodes, which then lead to branches. The ultimate results, or leaf nodes, have no offshoots. Making Decisions: The characteristics of the data are used to inform decision-making. For instance, we might choose a fruit based on its size or color to determine if it is an orange or an apple. Visual Representation: Consider it as a tree that branches out from a root node. Every branch symbolizes a choice made in response to a characteristic, with results appearing on the leaves. Building the Tree: To build the tree, we employ a technique known as CART (Classification and Regression Tree). The selection of features to employ and the timing of decisions are made easier by this algorithm. As a result, a decision tree is a visual tool for determining possible results or solutions for an issue by making decisions based on variables in the data. Because it begins with a root node and grows into stems to form a structure corresponding to a tree, it is known as a DT.

*f) Naive Bayes*

One sort of supervised learning utilized for classification tasks—particularly for text classification with huge datasets—is the Naïve Bayes functionality. WKT being easy to use but efficient in creating prediction models quickly. Probabilistic Classifier: Naïve Bayes makes predictions based on how likely it is that an object will fall into a specific class. For example, it might determine whether an email is spam or not by looking at the likelihood of specific terms showing up in spam emails. "Naïve" The assumption: Because it presumes that features are independent of one another, it is referred to as "naïve". For example, if we're trying to identify fruits based on color, shape, and taste, Naïve Bayes treats each feature (color, shape, taste) separately. So, even though these features might be related (like red apples being more likely to be sweet), Naïve Bayes assumes they're independent.

*2) Ensemble Learning Techniques*

Ensemble learning is a powerful technique in machine learning where many models are joined to enhance predictive accuracy and robustness. The underlying principle is that while individual models may have limitations or biases, combining their predictions can mitigate these weaknesses and yield better overall performance. One common approach in ensemble learning is through voting methods, where predictions from multiple models are aggregated, such as through major voting for classification tasks. Boosting is another widely used technique, wherein models are trained orderly, with each successive model concentrating on eradicating the errors made by its predecessors. This iterative process often results in highly accurate predictions. Stacking, on the other hand, involves combining predictions from a variety of models through a meta-learner. a meta-learner learns how to effectively integrate the expectations of base models to produce the final output. In this study, we employed various ensemble learning algorithms for our predictive modeling.[18]

*3) Gradient Boosting*

Gradient Boosting is a powerful method in machine learning that creates strong predictive models by combining multiple weaker models. Here's how it works: Combining Weak Models: It starts with simple models, often decision trees, and gradually improves them. Minimizing Loss: Each new model is trained to reduce the errors of the previous model. It does this by minimizing a loss function, like mean squared error or cross-entropy. Gradient Descent: The algorithm calculates the gradient (slope) of the loss function concerning the predictions of the current ensemble. This gradient guides the training of the new model. Training New Models: A new weak model is then trained to minimize this gradient. It focuses on correcting the errors made by the current ensemble. Adding Predictions: The predictions of the new model are added to the ensemble, improving the overall predictions. Iterative Process: This process repeats until a stopping point is reached, like when no further improvements are seen. Unlike AdaBoost, which adjusts the weights of training instances, Gradient Boosting uses residuals of the previous model as labels for training the next one. One popular form of Gradient Boosting is Gradient Boosted Trees, where each weak learner is DT (specifically CART - Classification and Regression Trees).

*4) LightGBM*

LightGBM offers a sizeable advancement in terms of efficiency and memory footprint due to its innovative techniques: GOSS and EFB.

GOSS redefines sample selection by assigning priority to instances with substantial gradient contributions while subsampling those with high gradients, thus hastening training without compromising model performance. On the other hand, EFB proposes another way of feature representation whereby exclusive features are packed into one bundle reducing dimensionality and conserving memory consumption. In summary, these techniques form the characteristic traits of LightGBM which differentiate it from traditional GBDT frameworks. By going beyond histogram-based algorithms, LightGBM paves way for efficient and effective gradient boosting which has led to superior machine learning performance

*5) XGBoost*

XGBoost, which is an abbreviation for Extreme Gradient Boosting, represents the state of the art in optimized distributed GBoosting libraries designed for efficient and scalable training of ML models. This ensemble learning approach merges knowledge from many weak models to produce a robust and powerful prediction machine.It is known for being able to handle massive datasets effectively and perform well across a wide range of machine learning tasks, making XGBoost one of the most widely used and useful tools in the field of machine learning. One key quality that makes it stand out is its ability to handle missing values in real world data sets. Consequently, this tool can work with such data directly reducing unnecessary preprocessing complexities; thus improving modeling pipeline speediness.In addition, XGBoost comes with built-in support for parallel processing which speeding up model training on large-scale data sets as opposed to traditional methods that are time-consuming

*6) Ensemble Methods*

After our first model did not make great predictions we applied various ensemble methods on it.Level 2 Models are created by combining Level 1 Model's predictions through weighted averages or simple pooling techniques. These techniques were employed to boost the overall predictive power of the model.

*a) Bagging*

Firstly, bagging involves the creation of numerous subsets of the original dataset through a process known as bootstrap sampling. This entails randomly selecting instances from the dataset with replacements, thereby generating subsets of the same shape as the original result set. Due to the nature of sampling with replacement, these subsets may contain duplicate instances, and each subset represents a slightly different perspective of the overall dataset.Secondly, a base model, typically a decision tree although other models can also be used, is trained on each of these bootstrap samples. As a result of the variations in the training data introduced by bootstrap sampling, each model trained on a different subset will inherently be slightly different from the others. Finally, once all the models have been trained, predictions are made for unseen data using each model. In regression tasks, the final forecast may entail averaging the predictions of all models, whereas, in classification tasks, the final values may be decided by a majority vote among the predictions of all models. The ensemble model's generalization performance is eventually enhanced by this combination of predictions from other models, which helps to minimize overfitting and lower variance. Bagging is particularly effective at reducing overfitting because it leverages the diversity introduced by training models on different subsets of result sets. By averaging the predictions of multiple models, the ensemble model tends to exhibit lower error rates compared to individual models, thus enhancing predictive accuracy. Popular algorithms that utilize bagging include Random Forests, XGBoost, LightGBM, and AdaBoost which employ bagging with decision trees as base learners, and Bagged Trees, which can utilize any base learning algorithm. Bagging's versatility in ensemble learning extends beyond decision trees, making it a widely applicable technique across various domains of machine learning.

*b) Stacking*

Other words used interchangeably with stacking include stacked generalizations or stacked.A good example would be ensemble as another effective method of group instruction. Instead of just averaging the predictions of several systems, stacking involves training a meta-model or meta-learner to figure out how to optimally combine the predictions of the base models. By using the underlying models' predictions as input features, this metamodel learns to produce predictions based on these inputs. In essence, it figures out how best to combine or weigh the underlying models' predictions to get the final one. Stacking is an ensemble technique that is more advanced than majority voting or simple averaging, which are employed in bagging methods. Enhancing predictive performance has been demonstrated to be highly beneficial, particularly when the underlying models are complementary and diversified. Nevertheless, stacking might need more precise hyperparameter adjustment and involve more computational work. Despite these difficulties, stacking is nevertheless a well-liked and effective method in the ensemble learning toolkit. e learning toolbox.

*c) Boosting*

Boosting is Additionally well-known as an ensemble learning method in machine learning, which emphasizes training several weak learners in turn giving a powerful ensemble system. In contrast to independent model training, boosting trains models iteratively by having each new model place greater emphasis on cases that the preceding model misclassified. Essentially, boosting is the process of combining several weak models—usually shallow decision trees, or "weak learners"—to produce a strong and precise predictive model. Boosting is well renowned for its capacity to generate extremely accurate models, frequently surpassing the performance of single models and even other ensemble techniques like bagging. Boosting algorithms, however, may be susceptible to noisy data and outliers, and to avoid overfitting, hyperparameters may need to be carefully adjusted.

*d) Tuning*

In machine learning parlance, "tuning" is modifying a few variables, or "hyperparameters," to maximize a machine learning model's performance. In contrast to a model's parameters, which are determined during training, hyperparameters are predetermined and have an impact on the learning process. Hyperparameter tuning is essential for getting the most out of a machine-learning model, regardless of the method employed.It often involves a trade-off between computational resources, such as time and hardware, and the quality of the resulting model. Effective hyperparameter tuning can lead to improved model accuracy, generalization, and robustness across different datasets and applications.

## IV. RESULTS & DISCUSSIONS

*A. Data Preprocessing*

Dataset Contains 12 Categorical values so we owned a Label Encoder to transfigure Categorical to Numerical.

*1) Label Encoder*

In machine learning, a label encoder is a preprocessing tool that transforms textual or category input into an indexed representation. In particular, this is crucial when collaborating with distinct machine learning algorithms that need quantitative input because a large number of algorithms are built to work with numerical data. An instance of the label encoder is created at initialization. The training dataset's categorical labels are "fitted" to the encoder. The encoder learns the mapping between each distinct label and a corresponding numerical value in this step. The dataset's categorical labels can be converted into numerical representations using the encoder once it has been fitted. Using the mapping that was discovered during the fitting process, each distinct label is substituted with its assigned numerical value.

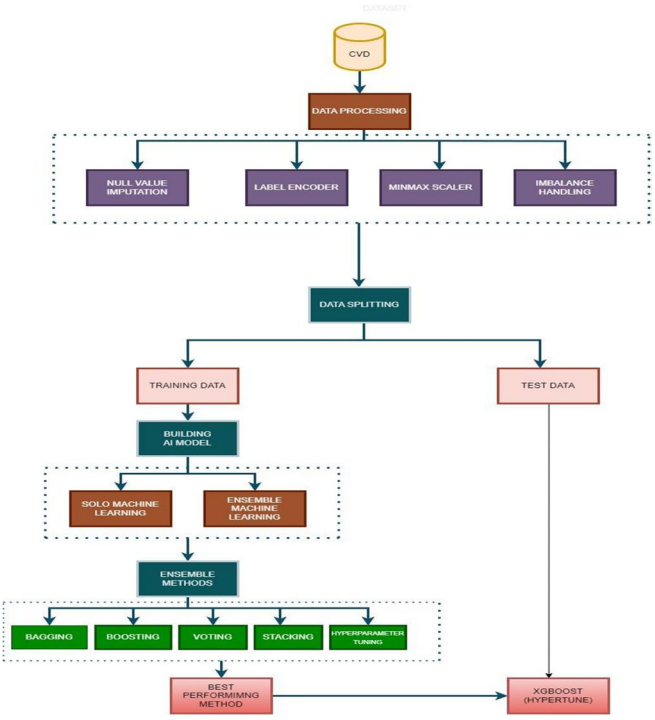



Fig1: Architecture Diagram

*2) Min Max Scaler*

MIN MAX Scaler is widely used to proportionate the nominal value from 0 to 1. To scale numerical characteristics inside a given range, usually between 0 and 1, machine learning uses the preprocessing approach known as min-max scaling. Our objective is to standardize the data and assign a uniform scale to every characteristic so that variations in their magnitudes do not lead to one feature predominating over others. This normalization might be especially crucial for algorithms, such gradient based optimization methods, that depend on gradients or distance measurements.

*3) Imbalance Data Handling*

In handling imbalanced data, we experimented with two techniques, SMOTE and ADASYN.

*a) SMOTE*

Machine learning uses the SMOTE data augmentation technique to solve the issue of class imbalance in classification jobs. When one class in the target variable has much fewer instances than another, it is said to be imbalanced. As a result, the minority lesson will see inefficient performance from one-sided models. To stabilize the distribution of classes, SMOTE generates synthetic examples with a concentration on the minority class.

*b) ADASYN*

ADASYN (Adaptive Synthetic Sampling) was created to solve some of its shortcomings. Similar to SMOTE, ADASYN is pre owned to manage class imbalance in datasets used for ML, especially in cases of classifying the input data into two mutually exclusive categories where one class is greatly underrepresented. We tried both approaches and discovered that the best accuracy was obtained when combining SMOTE with the XGBoost algorithm. This indicates that the best overall performance in predicting the target variable was obtained by using XGBoost to train our model and SMOTE to generate synthetic instances for the minority class.

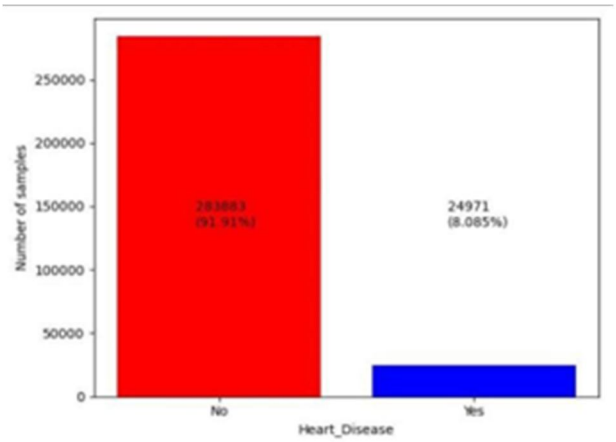



Fig2: Target Variable distribution before using SMOTE

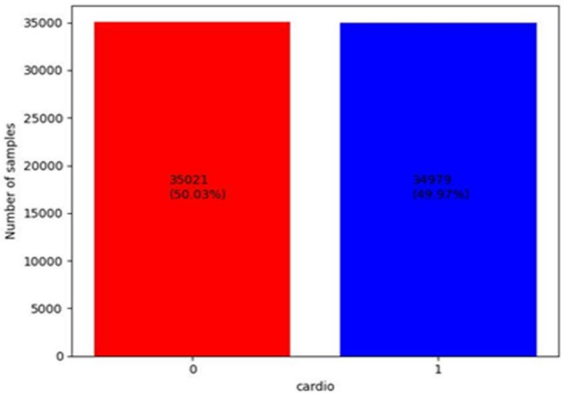



Fig3: Target Variable Distribution after using SMOTE

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 12 Issue III Mar 2024- Available at www.ijraset.com*

### *B. Data Splitting*

We used the K fold Cross Validation technique and tested our set of results with 10 folds,4 folds,2folds,5 folds and took the model with the validation that gives more performance metrics and k=10 with Extreme Gradient Boosting gives superlative rightness.

| Model | Accuracy | AUC | Recall | Precision | F1 Score |
|---|---|---|---|---|---|
| xgboost | 0.917 | 0.8288 | 0.9013 | 0.8858 | 0.883 |
| lightgbm | 0.9009 | 0.8182 | 0.836 | 0.8884 | 0.8863 |
| random forest | 0.8832 | 0.8064 | 0.9171 | 0.8805 | 0.8865 |
| ada | 0.8502 | 0.8065 | 0.9 | 0.8801 | 0.8835 |
| et | 0.8256 | 0.7929 | 0.854 | 0.8781 | 0.8883 |
| gbc | 0.8256 | 0.8304 | 0.8456 | 0.8894 | 0.8883 |
| dt | 0.8613 | 0.5739 | 0.8613 | 0.8719 | 0.897 |
| dummy | 0.7892 | 0.6595 | 0.7892 | 0.8448 | 0.8664 |
| svm | 0.7508 | 0 | 0.7592 | 0.9074 | 0.8804 |
| Lr | 0.7476 | 0.8358 | 0.7508 | 0.9125 | 0.7951 |
| ridge | 0.7413 | 0 | 0.7476 | 0.9127 | 0.803 |
| lda | 0.7412 | 0.8348 | 0.7413 | 0.9127 | 0.7984 |
| knn | 0.6839 | 0.5606 | 0.6839 | 0.862 | 0.7532 |
| nb | 0.6091 | 0.801 | 0.6091 | 0.911 | 0.696 |
| qda | 0.5701 | 0.787 | 0.5701 | 0.9116 | 0.6615 |

Table2: Performance Metrics of different algorithms(k=10 folds)

### *C. Model Selection*

After separating our result set into training and testing sets and preprocessing it, choosing XGBoost as our model. The next stage is to train the model on the training set and assess its effectiveness. Since you suggested utilizing K Fold Cross Validation by having K=10, we used cross validation to obtain a reliable performance estimate for our model. To determine which machine learning algorithm would work best for our dataset, we investigated a variety of models in this study. LightGBM, XGBoost, GB Classifier, Random Forest (RF), Extra Trees, AdaBoost (ADA), Decision Tree (DT), SVM, Logistic Regression (LR), Ridge Classifier (Ridge), Linear Discriminant Analysis (LDA), K-nearest neighbors (KNN), Naive Bayes (NB), and Quadratic Discriminant Analysis (QDA) are among the models we have developed. Among them, XGBoost appeared to be the most appropriate choice for our purposes based on such factors as accuracy, robustness, and interpretability. It has been shown to possess some of the best characteristics regarding accuracy and reliability. Therefore, we will now refine this model by modifying hyperparameters examining feature importance and preparing it for production. Henceforth it is crucial that we validate carefully comprehending these results in order to ascertain if the XGboost that we selected aligns with our customized machine learning task. Because of this, in addition to hyperparameter tuning, Bagging, Boosting, Stacking and Voting will increase the performance of our chosen model XGBoost. We intend to further enhance the system's robustness as well as its predictability using bagging , boosting , stacking , voting which build on top of ensemble approach.

©IJRASET: All Rights are Reserved 

Bagging will involve training top 5 models on different subsets from dataset; thus reducing overfitting while improving stability. Boosting concatenates a sequential training process, enabling the model to compare and progressively correct its errors, holding intricate relationships in the data. Stacking embraces diverse model predictions as inputs into a new meta-learner, refining a final output. Voting involves consolidating the predictions of multiple models, contributing to a more comprehensive decision map. Hyperparameter tuning involves an iterative adjustment of configuration setting, of the model, attempting to discern the order that yields the best results. By systematically tuning the model, we can reduce the system's performance on our specific dataset. We aim to extract the maximum potential from XGBoost, through numerous adjustments.

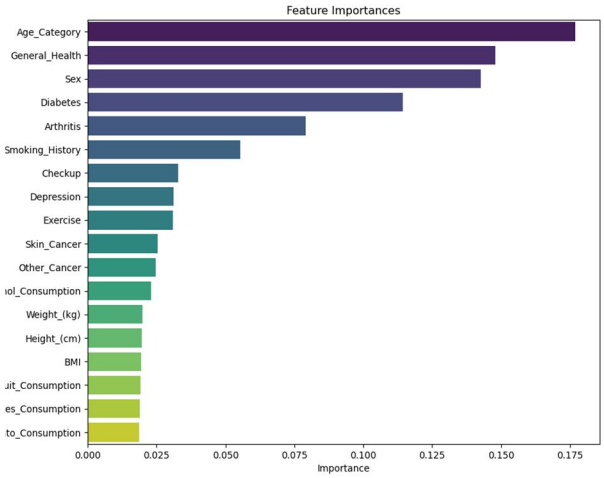



Fig 4: Feature importance

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| lightgbm | 0.9190 | 0.8838 | 0.923 | 0.8900 |
| xgboost | 0.9189 | 0.8827 | 0.9156 | 0.8888 |
| Random Forest | 0.9065 | 0.8829 | 0.9073 | 0.8934 |
| Adaboost | 0.82192 | 0.8847 | 0.8009 | 0.8541 |
| gbc | 0.8256 | 0.8900 | 0.8456 | 0.8883 |

Table 3: Performance Metrics obtained using Bagging

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| stacking_model | 0.90704 | 0.8835 | 0.9132 | 0.8918 |

Table 4: Performance Metrics obtained using stacking

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| voting_model | 0.91031 | 0.8818 | 0.9090 | 0.8868 |

Table 5: Performance Metrics obtained using voting

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| xgboost | 0.9200 | 0.8900 | 0.9200 | 0.8900 |
| lightgbm | 0.9114 | 0.8835 | 0.9110 | 0.8866 |
| gbc | 0.9065 | 0.8629 | 0.8973 | 0.8834 |
| Adaboost | 0.8956 | 0.8847 | 0.8909 | 0.8541 |
| Random Forest | 0.8256 | 0.8900 | 0.8256 | 0.8183 |

Table 6: Performance Metrics obtained using Hyperparameter Tuning

*D. Model Evaluation*

Following our XGBoost hyperparameter tuning model, we obtained outstanding model performance outcomes. The accuracy was a remarkable 92% demonstrating the general accuracy of our predictions.Additionally, the % of real positive predictions among all positive predictions is indicated by the precision of 0.89. This measure is very useful when trying to reduce false positives.The model's recall of 0.92 shows that it can accurately identify the majority of true positive cases.It highlights how well the model detects positive situations by showing a low percentage of false negatives.The F1-Score in our case was 0.89, indicating a balanced performance in recall and precision. Together, these assessment indicators show how reliable and efficient our adjusted XGBoost model is. These outcomes, in our opinion, demonstrate the model's capacity to fulfill the goals set out in our machine learning job.

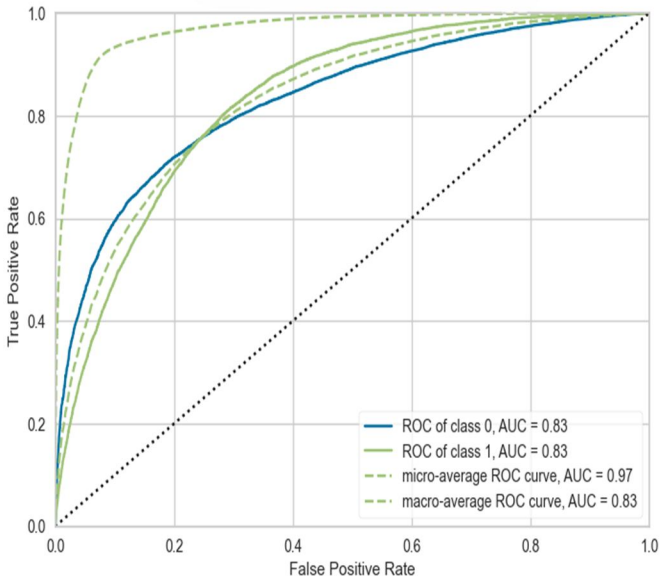



Fig 5: AUROC performance curve of XGBoost Hypertuning.

## V. CONCLUSION

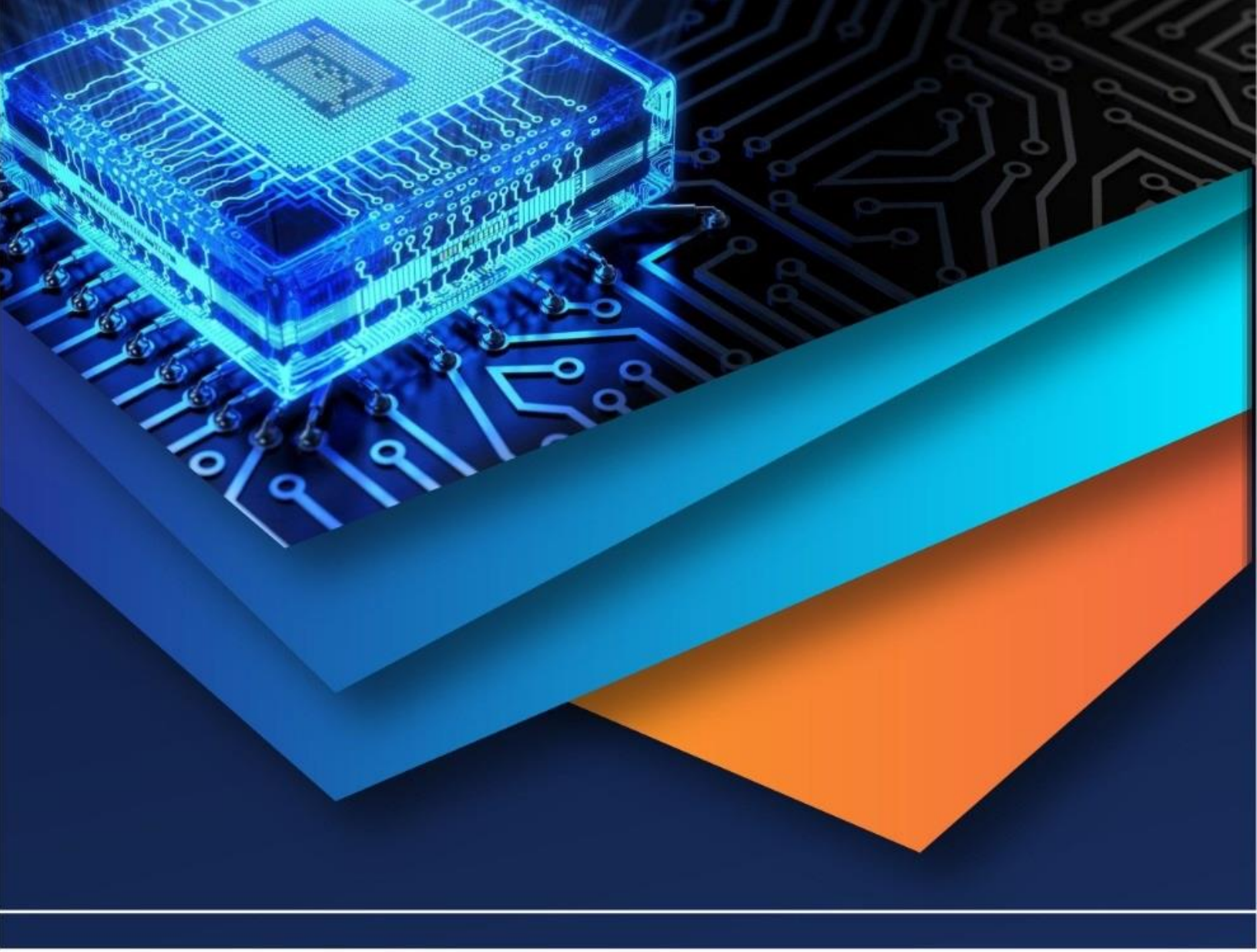
In conclusion, our study on predicting Cardiovascular Disease (CVD) using advanced ensemble techniques leads us to the hypothesis that combining multiple machine learning models results in higher accuracy. Our main aim was to develop a reliable tool for early detection of CVD and intervention for those at risk. Through our exploration of various ensemble methods, including Random Forests, AdaBoost, Gradient Boosting, and XGBoost, we successfully created an ensemble model with an accuracy of 92%. This finding suggests that the hypothesis holds – ensembling techniques contribute to improved predictive accuracy in the context of CVD. The ensemble approach, blending the strengths of different models, plays a crucial role in achieving this high accuracy. By avoiding overfitting and demonstrating effectiveness in real-world healthcare scenarios, our ensemble model stands out as a promising tool for identifying individuals at risk of cardiovascular complications.

©IJRASET: All Rights are Reserved 

As we move forward, it becomes evident that our hypothesis aligns with the outcomes of this study. Ensembling machine learning models, as demonstrated in our research, offers a practical avenue for healthcare professionals and policymakers to enhance the accuracy of predictive tools. This, in turn, contributes to proactive healthcare interventions and the prevention of cardiovascular diseases in at-risk populations.

## REFERENCES

[1] WHO, Geneva. "WHO methods and data sources for country-level causes of death." (2014)
[2] Singirikonda, Bhagyalaxmi, and Muktevi Srivenkatesh. "An Approach to Prediction of Cardiovascular Diseases using Machine and Deep Learning Models." International Journal of Intelligent Systems and Applications in Engineering 10
[3] Kapila, Ramdas, T. Ragunathan, Sumalatha Saleti, T. Jaya Lakshmi, and Mohd Wazih Ahmad. "Heart Disease Prediction using Novel Quine McCluskey Binary Classifier (QMBC)."
[4] Ali, Liaqat, Atiqur Rahman, Aurangzeb Khan, Mingyi Zhou, Ashir Javeed, and Javed Ali Khan. "An automated diagnostic system for heart disease prediction based on ${\chi^{2}} $ statistical model and optimally configured deep neural network."
[5] Sajja, Tulasi Krishna, and Hemantha Kumar Kalluri. "A Deep Learning Method for Prediction of Cardiovascular Disease Using a Convolutional Neural Network."
[6] Shrivastava, Prashant Kumar, Mayank Sharma, and Avenash Kumar. "HCBiLSTM: A hybrid model for predicting heart disease using CNN and BiLSTM algorithms." Measurement: Sensors 25 (2023): 100657.
[7] Meng, James, and Ruiming Xing. "Inside the "black box": Embedding clinical knowledge in data-driven machine learning for heart disease diagnosis."
[8] Yazid, M. Haider Abu, Muhammad Haikal Satria, Shukor Talib, and Novi Azman. "Artificial neural network parameter tuning framework for heart disease classification.
[9] Ahamed, Jameel, Abdul Manan Koli, Khaleel Ahmad, Alam Jamal, and B. B. Gupta. "CDPS-IoT: cardiovascular disease prediction system based on IoT using machine learning." (2022).
[10] Gupta, Ankur, Rahul Kumar, Harkirat Singh Arora, and Balasubramanian Raman. "MIFH: A machine intelligence framework for heart disease diagnosis." IEEE access 8 (2019)
[11] Alqahtani, Abdullah, Shtwai Alsubai, Mohammed Sha, Lucia Vilcekova, and Talha Javed. "Cardiovascular disease detection using ensemble learning." Computational Intelligence and Neuroscience 2022 (2022).
[12] Pradhan, M. R. "Cardiovascular disease prediction using various machine learning algorithms." Journal of Computer Science 18, no. 10 (2022): 993-1004.
[13] Singh, Vicky, and Brijesh Pandey. "Prediction of Cardiac Arrest and Recommending Lifestyle Changes to Prevent It Using Machine Learning." In International Conference on Intelligent Technologies & Science, pp. 1-6. 2021.
[14] Karthikeyan, R., D. Vijendra Babu, R. Suresh, M. Nalathambi, and S. Dinakaran. "Cardiac Arrest Prediction using Machine Learning Algorithms." In Journal of Physics: Conference Series, vol. 1964, no. 6, p. 062076. IOP Publishing, 2021
[15] Arun Kumar, N., and P. Uma Maheshwari. "Neural Network Based Approach in Identifying Cardio Vascular Disease-A Survey."
[16] Mohan, Senthilkumar, Chandrasegar Thirumalai, and Gautam Srivastava. "Effective heart disease prediction using hybrid machine learning techniques." IEEE access 7 (2019): 81542-81554..
[17] El Naqa, Issam, and Martin J. Murphy. What is machine learning?. Springer International Publishing, 2015.
[18] Zhang, Cha, and Yunqian Ma, eds. Ensemble machine learning: methods and applications. Springer Science & Business Media, 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)