Heart Failure Early Prediction Using Machine And Deep Learning Algorithm

Lamis F. Al-Qora'n[1] , Qusay Bsoul[2], Firas Zawaideh[3], Ala Alzoubi[4] , Silvyras Sayed[5] , Raghad W. Bsoul[6] , Diaa Salama AbdElminaam[7,8] , Nour Mostafa[9,*]

[1]Faculty of Information Technology, Department of Software Engineering, Philadelphia University, Amman, Jordan

[2]Cybersecurity Department, College of Computer Sciences and Informatics, Amman Arab University, Amman, Jordan

[3]Cybersecurity Department, Faculty of Science and Information Technology, Jadara University, Irbid, Jordan
[4]Faculty of Information Technology, Applied Science Private University, Amman, Jordan
[5]Faculty of Engineering, Misr International University, cairo , Egypt
[6]Department of Clinical Pharmacy Faculty of Pharmacy, Jordan University of Science and Technology, Irbid, Jordan

[7]MEU Research Unit, Middle East University, Amman, Jordan
[8]Jadara Research Center, Jadara University, Irbid, Jordan
[9]College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait

Emails: Lalqoran@philadelphia.edu.jo; q.bsoul@aau.edu.jo; F.zawaideh@jadara.edu.jo; a_alzoubi@asu.edu.jo; silvyras1907459@miuegypt.edu.eg; meramidebsoul@yahoo.com; diaa.salama@miuegypt.edu.eg; nour.moustafa@aum.edu.kw

Abstract

In this article, we use machine learning approaches to give a thorough investigation into the prediction of cardiac illnesses and strokes. The Stroke Prediction Dataset and the Heart Failure Prediction Dataset are the two datasets that we use. Our objective is to maximize accuracy and minimize Mean Absolute Error (MAE) and Mean Squared Error (MSE) in order to enhance predictive performance. We use a variety of machine learning methods, such as Random Forests, Naive Bayes, Decision Trees, and k-Nearest Neighbors (KNN). We also use Artificial Neural Networks (ANN) and Multi-Layer Perceptrons (MLP) as deep learning models. We use oversampling approaches to rectify the imbalance in classes. For hyperparameter tweaking, we also use Grid Search and k-Fold Cross Validation. Our goal is to deliver valuable insights into early detection and preventive measures through comprehensive testing and assessment for prevention of strokes and heart diseases.

Keywords: Heart Disease; Machine learning; Deep learning; Multi layer perceptron; Model evaluation

1 introduction

Heart disease is a critical worldwide medical problem, with different structures influencing the heart and veins. One normal sort is coronary vein sickness, which creates because of cholesterol stores (plaques) in the heart courses, causing atherosclerosis. This condition decreases blood stream to the heart and other body parts, possibly prompting cardiovascular failures, chest torment (angina), or stroke. Heart disease is brought on by a number of factors. Age is a critical variable, as more established people are bound to have harmed and limited

DOI: https://doi.org/10.54216/FPA.180113
Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

supply routes. Orientation likewise assumes a part, with men by and large at a higher gamble, and ladies' gamble expanding after menopause. Existing ailments, for example, hypertension, elevated cholesterol levels, and diabetes, can improve the probability of coronary illness. Way of life factors, including actual idleness, delayed pressure, undesirable weight control plans, and smoking, likewise add to the gamble. Last but not least, having a family history of heart disease, especially if a close relative was diagnosed at a young age, can make you more likely to get heart disease. For a proper diagnosis and treatment, it is essential to be aware of the symptoms and risk factors of heart disease and to consult a medical professional. Early prediction and way of life adjustments can essentially work on the guess for those impacted by coronary illness. As indicated by the World Health Association, cardiovascular sicknesses are the main source of death around the world, representing an expected 17.9 million fatalities every year. In addition, research reveals alarming patterns in younger demographics, highlighting the need for proactive measures and increased awareness. In light of these figures, the best way to combat this widespread health problem is to promote preventative measures, encourage healthier habits, and advance predictive technologies.

Many attempts have previously been made by researchers to use deep learning and machine learning algorithms to detect cardiac disease. These methods have been proven in recent research to improve the accuracy of heart disease prediction. For instance, a study published in the Journal of Medical Systems in 2023 used a deep learning model to predict heart disease based on patient data, achieving an accuracy of 93%. Another study published in the Journal of Biomedical Informatics in 2024 used a combination of machine learning algorithms to predict heart disease, achieving an accuracy of 90%. These research show how machine learning and deep learning algorithms may be used to increase the prediction accuracy of cardiac disease, which might ultimately result in earlier diagnosis and improved patient outcomes.

A subset of artificial intelligence (AI) called machine learning enables computer frameworks to learn and advance without explicit programming as a result. It involves enhancing computations and factual models that enable robots to analyze and draw surmising from information, recognize examples and connections, and settle on randomforest or decision in view of that data. Supervised learning, unsupervised learning, and reinforcement learning are the three primary categories into which machine learning algorithms fall. supervised learning includes preparing a model on marked dataset, where the info and result factors are known, and the model figures out how to plan contributions to yields in view of the preparation information. Unsupervised learning includes preparing a model on unlabeled dataset, where the information factors are known, however the result factors are not, and the model figures out how to distinguish examples and connections in the information. Reinforcement learning includes preparing a model to settle on choices in a unique climate, where the model figures out how to expand a prize sign by communicating with the climate. . Deep learning is a subset of machine learning that uses counterfeit brain organizations (ANNs) with various layers to learn and address complex examples and connections in information.The term "deep" learning refers to the vast number of layers in the neural network that allow the model to learn more abstract and high-level characteristics from the input data.Without the requirement for human feature engineering, deep learning models can learn and represent features directly from raw data, including text, audio, and images. This capacity empowers profound learning models to accomplish cutting edge execution in different applications, including PC vision, regular language handling, and discourse acknowledgment, among others.Deep learning models consist of layers of interconnected nodes or neurons: an input layer, one or more hidden layers, and an output layer. Deep learning models consist of interconnected nodes or neurons arranged in layers, empowering the model to learn complex and non-direct connections in the information. Deep learning models can be trained using supervised, unsupervised, or reinforcement learning techniques, depending on the application and the availability of labeled data .

The overarching objective of our study in this research paper is two datasets. The first dataset is the Dataset for Stroke Prediction. Based on several factors, such as age, gender, smoking status, and multiple illnesses, this information is used to calculate a patient's risk of stroke. We have accurately predicted the patient's heart status with the aid of both machine and deep learning models. We have utilized many classifiers such as K-Neighbors, DECISION TREE, NAÏVE BAYES, RANDOM FOREST, and convolutional neural networks (ANNs), as well as the MLP classifier.

The Heart Failure Prediction Dataset is the second dataset. Several datasets that were previously available separately but had never been combined were combined to generate this new dataset. This dataset is the largest heart disease dataset currently available for research purposes because it combines five different heart datasets with eleven shared features. In order to accurately forecast the patient's cardiac status, we have

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

employed both machine learning and deep learning models. For instance, we have used the K-Neighbors Classifier, DECISION TREE, NAÏVE BAYES, RANDOM FOREST, Artificial neural networks (ANNs), and MLP classifier.

To summarize:

- The execution includes careful data collection from clinical investigates, Stroke Prediction Dataset and Heart Failure Prediction Dataset ,providing 11 clinical features for predicting heart disease events.
- An extensive preprocessing stage guarantees dataset refinement by handling missing values in some features using linear regression prediction replacing them with NaN , label encoding for non numeric values, detect outliers using IQR , removing outliers and oversampling to balance the class distribution .
- Two data splitting methods such as KFold cross-validator and grid search with cross-validation were used to split data in order to get the models best parameters, improve performance and generalization
- The incorporation of hyperparameter advancement strategies for both standard ML methods and DL models improves model execution and intermingling, with grid search and Adam optimizer optimization.
- The implementation encompasses various recommendation algorithms to enhance heart disease prediction .They include, K-Nearest Neighbors, Decision Trees NAÏVE BAYES, RANDOM FOREST ,Artificial neural networks (ANNs), and MLP classifier. The goal is to early detect and manage heart diseases using a machine and deep learning model.
- After implementing various algorthims for prediction we used performance metrics such as Accuracy,Precision score,recall score ,F1 Score, MSE mean squared error and MAE mean absolute error.

This paper is organized as follows Section 1 consist the introduction about heart diseases , statistical analysis and Machine and Deep learning explanation . In section 2, we have shown the related work of other authors in the same topic. In section 3 we have explained the methodologies and all the techniques which we have applied for our two datasets. In section 4 we have shown the summaries with result of this work and section 5 list out the conclusion.

2 related work

Examining the state of the art in terms of heart failure prediction research and developments. This thorough analysis explores the combination of deep learning and advanced analytics together with two essential approaches: machine learning models and clinically-based assessment. This section aims to trace the evolutionary trajectory of heart failure prediction paradigms through a thoughtful synthesis of the literature. It does this by highlighting significant discoveries and technological developments that together shape the current state of personalized medicine and predictive healthcare.

In[1] , the author presented a hybrid approach in this research paper in the use of machine learning techniques to predict readmission due to heart failure is examined in the publication ”Evaluation of Machine Learning Methods for Predicting Heart Failure Readmission: A Comparative Analysis”. Through the use of wearable technology and ongoing health data monitoring for the early diagnosis and treatment of chronic diseases, the authors hope to improve patient outcomes and lower healthcare costs.The research paper outlines a five-step methodology for a proposed framework, including data set pre-processing, which involves cleaning, transforming, and organizing data for analysis and modeling, thereby improving data quality and enabling machine learning algorithms to effectively learn from it.The research uses machine learning techniques such as K-Nearest Neighbors, Artificial Neural Networks, Support Vector Machines, Decision Trees, and Naive Bayes to categorize healthy and heart disease-related citizens. It compares traditional logistic regression with alternative methods like wild-forests, boosting, random forests, and logistic regression for heart disease prediction and response.The study aimed to predict heart failure readmission using various factors, including patient characteristics, medical history, laboratory tests, medication history, hospital stay duration, symptoms, ejection fraction, ECG results, and imaging findings. The researchers created a machine learning prediction model to

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

identify high-risk individuals, aiming to improve heart failure care and reduce readmission.Machine learning approaches improved the prediction of readmission for heart insufficiency post-hospitalization, outperforming LR and providing the highest predictor range in observation duration..

In,[3] the authors of this paper introduced a study that uses machine learning techniques to develop a hybrid decision support system for early heart disease diagnosis. It analyzes previous systems, proposes a new system, discusses functionality, and recommends further work. The goal is to improve heart disease prediction accuracy.The authors prepared data for a hybrid decision support system for heart disease prediction using MICE, GA, RFE, SMOTE, and standard scalar methods. These techniques optimized the performance of machine learning models and prepared the dataset.The study developed a hybrid decision support system for heart disease prediction using machine learning techniques like SVM, Random Forest, Naive Bayes, Logistic Regression, and Adaboost. The system improved heart disease diagnosis accuracy by evaluating and contrasting various algorithms.The authors plan to use deep learning techniques in the future development of a system for heart disease diagnosis, utilizing multiple-layered neural networks to extract intricate patterns and representations from data.The study used correlation and cuckoo search algorithms to select features for predicting heart disease. The main goal was to create a hybrid decision support system for early diagnosis and prognosis using machine learning algorithms and advanced computational techniques to accurately assess clinical characteristics.The report suggests a hybrid decision support system for heart disease prediction with improved accuracy compared to current systems. However, precise percentages and metrics are lacking, and the authors should provide complete results.

In[2] ,the authors in this paper discusses a hybrid machine learning model for predicting heart disease using the Cleveland dataset. It uses AI calculations like Decision Tree and a hybrid model to improve Random Forest accuracy. The research aims to offer efficient decision-making and accurate prediction solutions for the medical sector. The paper "Heart Disease Prediction Using a Hybrid Machine Learning Model " explores AI strategies for predicting Heart disease. It uses, Decision Tree and a hybrid model to improve heart disease prediction. The model was applied to the Cleveland heart disease dataset to determine the presence or absence of heart disease.The Cleveland heart disease dataset, containing 14 characteristics, was used in a study to predict heart disease. The review categorized the presence or absence of coronary illness using a twofold characterization issue. Common features used in Random Forest models include age, sex, cholesterol levels, circulatory strain, chest pain type, pulse, and risk factors like smoking or diabetes.The review found high accuracy rates for AI models in predicting coronary disease, with 79% accuracy in decision trees and 81% and 88% accuracy in a hybrid model. The hybrid model combining Decision Tree and Arbitrary Woods achieved the highest precision.

In,[4] The paper discusses the use of AI calculations for predicting heart diseases, highlighting the importance of machine learning in improving diagnosis efficiency and accuracy. The authors compare four AI calculations - Decision tree, linear regression, k-nereast neighbor, and support vector machine - and discuss the importance of preprocessing data to ensure the best results. The paper highlights the potential of machine learning to transform heart disease diagnosis and prediction, leading to improved patient outcomes and healthcare practices.The paper discusses the importance of preprocessing information and machine learning techniques for accurate outcomes from AI calculations. It highlights the need to deal with invalid qualities in raw data before applying calculations. The review evaluates AI calculations for predicting coronary disease using the UCI vault dataset and Python programming, particularly the Boa constrictor with Jupyter Note pad. These methods enable the investigation of complex datasets and the identification of examples that contribute to precise predictions of heart-related conditions.The study examined the performance of many machine learning algorithms in predicting cardiac disease. The k-nearest neighbor (knn) algorithm demonstrated the highest accuracy, with an accuracy rate of 87 %. The study also compared the precision of Support Vector Machine, Decision Tree, and linear regression. The knn algorithm outperformed the others, demonstrating the importance of accuracy in predicting heart diseases. The decision of AI calculation significantly impacted the accuracy of heart disease forecasts, with knn showing the most significant precision.

In,[5] Using datasets like The Cleveland and IEEE Dataport, Mohammed Ahmed and Idress Husien from the University of Kirkuk examine hybrid machine learning approaches for heart disease prediction.The review discusses the preprocessing strategies used for a dataset for heart disease prediction, including information cleaning, handling missing values, highlight scaling, including choice, information change, and outlier detection. These methods ensure data quality, accuracy, and the ability to train machine learning models for accurate prediction of heart disease.The study utilized hybrid machine learning techniques to create predictive models for heart disease prediction. Key AI strategies included K-Nearest Neighbor (KNN), Decision

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

Tree, Support Vector Machine (SVM), Random Forest, and Naïve Bayes. These models were created based on data features and evaluated for accuracy. The researchers selected the most effective models for accurate predictions about cardiovascular disease. The study aimed to improve the accuracy of these models.The study found that half-breed AI strategies, including K-Nearest Neighbor (KNN), ensemble method, Gradient Boosting and Extra-Tree Classifier, and hybridized Logistic Regression and Random Forest, are effective in accurately predicting heart diseases. These models outperformed traditional methods, with KNN showing the highest accuracy at 90.8%.

In,[6] This paper focuses on developing a machine learning model for accurately predicting cardiovascular diseases, highlighting its potential in recognizing patterns in data and its importance in diagnosis and treatment. It proposes k-modes bunching and evaluates models like random forest, decision tree classifier, and XGBoost.The review used k-modes grouping and scaling to preprocess a dataset for heart disease prediction. It used Kaggle for data access and Google Colab for calculations. Different learning methods like random forest, decision tree classifier, multi-facet perceptron, and XGBoost were used. Accuracy metrics were used for evaluation.The review used k-modes grouping and scaling to preprocess a dataset for heart disease prediction. It used Kaggle for data access and Google Colab for calculations. Different learning methods like random forest, decision tree classifier, multi-facet perceptron, and XGBoost were used. Accuracy metrics were used for evaluation.

In,[7] This paper focuses on developing a machine learning-based model for predicting cardiovascular diseases, aiming to improve order precision and reduce casualties. The study uses patient records and various algorithms, including k-modes grouping, to improve accuracy. The model also considers new, hidden information and interpretability of outcomes. Further research is recommended to improve k-modes grouping for heart disease prediction.The authors discussed preprocessing methods and machine learning algorithms used to predict heart disease. They used the UCI repository dataset, which contains key elements like cholesterol levels, circulatory strain, and age, to train and test the algorithms. Python programming was used for executing the AI calculations, and the Anaconda distribution and Jupyter Notebook were used for Python programming. The authors aimed to create a viable prediction framework for heart disease using preprocessing methods and a mix of AI calculations, such as choice tree, direct relapse, k-closest neighbor, and support vector machine.In the study, machine learning algorithms for heart disease prediction were examined, revealing that the k-Nearest Neighbor (KNN) algorithm demonstrated the highest accuracy at 87%. The accuracy was measured using disarray grids, with KNN outperforming other algorithms like Support Vector Machine (SVM) and Decision Tree. The study highlights the importance of accuracy in heart disease prediction and the significance of selecting the right algorithm, with KNN demonstrating the highest accuracy among the evaluated algorithms.

In,[10]The study explores the use of machine learning algorithms for predicting heart diseases, analyzing KNN, Logistic Regression, and Random Forest Classifiers. It also discusses the use of artificial neural networks and deep learning to identify risk factors and develop a successful Coronary illness Expectation Framework for early diagnosis and treatment.The review uses preprocessing steps to remove missing information, clean it, and standardize calculations. AI methods like Calculated Relapse, K Nearest Neighbors, and Random Forest Classifier are used to analyze and categorize preprocessed data for heart disease prediction.The review discusses the use of deep learning algorithms, specifically fake brain organizations, for predicting heart diseases and identifying risk factors, emphasizing the importance of these methods in diagnosing heart conditions.The review shows that AI calculations like KNN, Logistic Regression, and Random Forest Classifier have shown promising results in predicting heart diseases, with KNN and Logistic Regression achieving 88.5% accuracy, and Calculated Relapse and KNN outperforming Random Forest Classifier in predicting coronary disease.

In,[8] The study compares four machine learning algorithms for predicting heart disease using the UCI Heart Disease dataset, aiming to identify the most accurate method.The study used the "mean of column" technique to replace missing values in the dataset, and converted it from numerical to nominal values to ensure compatibility with the machine learning methods.Four AI strategies were used to develop a coronary illness forecast model: Support Vector Machine (SVM), Random Forest, Decision Tree, and K-Nearest Neighbors (KNN). The review assessed the accuracy of these models in predicting coronary illness based on the dataset's characteristics. Decision Tree performed poorly with an accuracy level of 85%, while Random Forest achieved the highest accuracy of 99%.The review analyzed AI models for predicting coronary illness, with SVM achieving 98% accuracy, Random Forest showing almost 100% accuracy, decision Tree at 85%, and Naïve Bayes at 90.4% accuracy, with Random Forest being the most precise model in the dataset.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

In,[11] 1 the development of a machine-learning-based predictive model for non-home discharge among patients with acute heart failure (AHF). The review used patient socioeconomics, comorbidities, and treatment information to make a model that can foresee which patients are probably going to experience issues being released home. The objective is to lessen the burden placed on healthcare professionals, patients, and their families by facilitating early coordination of medical facilities for patients who may require transfer after receiving acute care. The essential AI model utilized in the review for fostering the prescient model of non-home release among intense cardiovascular breakdown patients is Rope relapse. Rope relapse is a kind of straight relapse that consolidates a punishment term (L1 regularization) to recoil the coefficients of a few non-educational factors to nothing, consequently performing variable choice and upgrading model interpretability. In particular, the review utilized L1-punished assessment, otherwise called the most un-outright shrinkage and determination administrator (Tether relapse), to make a model with better segregation capacity and clinical convenience. The specialists utilized the Tether relapse model with the 1 standard-mistake (1SE) rule to accomplish more tightfisted models by choosing fundamental factors for anticipating non-home release among AHF patients . In keeping with the research's clinical epidemiological context, the choice of Lasso regression in this study reflects the machine learning approach's emphasis on interpretability rather than complexity.It focuses on evaluating the predictive performance of a machine learning model for predicting non-home discharge among acute heart failure patients. While the study does not provide a specific accuracy value, it emphasizes the use of the c-statistic (AUC) as a measure of discrimination ability. The c-statistic assesses the model's ability to differentiate between positive and negative cases, with higher values indicating better discrimination. .

In,[10] In discussing how machine learning algorithms might be used to predict heart disease, the research highlights the importance of early identification and prevention as well as the growing global incidence of heart-related ailments. It compares several machine learning algorithms, including KNN, ID3, Naïve Bayes, Random Forest, and SVM-RFE, to assess their effectiveness in predicting heart disease. The authors evaluate different factors, datasets, and efficiency metrics to identify the most suitable algorithm for heart disease prediction. Furthermore, the paper underscores the importance of deep learning techniques in enhancing prediction accuracy and proposes future research directions in this field. Additionally, ethical considerations and conflicts of interest related to the research are also addressed in the study.The paper conducted preprocessing on the data before utilizing machine learning algorithms for heart disease prediction. The preprocessing steps included extracting necessary data from the UCI Repository, normalizing the data, selecting key features, handling missing values by replacing them with NaN , and preparing the dataset for analysis. The paper utilized various machine learning models for heart disease prediction, including K-Nearest Neighbors (KNN), ID3 algorithm, Naïve Bayes, Random Forest, and SVM-RFE (Support Vector Machine with Recursive Feature Elimination) . applying different algorithms to the preprocessed data, and comparing their results determined the most effective model for predicting heart disease.The results indicated that SVM, Logistic Regression, and Artificial Neural Network (ANN) had similar accuracy levels, while Random Forest outperformed all other algorithms with the highest accuracy. Specifically, Random Forest achieved an accuracy of 95.60

In,[9] features the significant effect of coronary illness on worldwide death rates, explicitly accentuating cardiovascular circumstances like coronary corridor sickness and cerebral stroke. It highlights the basic requirement for opportune and exact clinical determination to alleviate fatalities related with coronary illness. It also provides insights into the application of machine learning in predicting heart disease, the importance of data pre-processing, and the use of various algorithms to enhance accuracy in diagnosis and prediction. This paper included preprocessing steps as the collected data was cleaned to address noise and missing values. This step involved identifying and handling any inconsistencies, errors, or missing data points in the dataset to improve the quality and reliability of the information, transformation process included tasks such as smoothing, normalization, and aggregation to enhance the usability of the data for machine learning algorithms.This paper has also introduced data integration from different sources . All relevant data are consolidated into a unified format for processing through integration. This combination takes into consideration a more far reaching examination of the consolidated dataset, empowering scientists to determine significant bits of knowledge and make exact expectations in regards to coronary illness.The paper utilized several machine learning models to predict heart disease, including Naïve Bayes, decision tree, K-nearest neighbor, and random forest algorithms . These models were applied to the dataset to analyze and predict heart disease attributes with the aim of achieving maximum accuracy.The results of the study indicated that the K-nearest neighbor algorithm achieved the highest accuracy score among the models tested .

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

Table 1: Datasets Information. Tables should be placed in the main text near to the first time they are cited.

Dataset Name	Year	Source	Size
Stroke Prediction Dataset	2021	Kaggle	316KB
Heart Failure Prediction	2021	Kaggle	6KB

3 Methodology

- Data collection
- Data preprocessing
- Data splitting
- Optimization parameters for Machine Learning (ML) and Deep Learning (DL)
- Recommendation based on Machine Learning and Deep Learning Algorithms
- Prediction and evaluation metrics

3.1 Data collection

The data collection phase is a critical precursor to any comprehensive analysis, and in the context of our project, it involves harnessing the valuable insights embedded within three distinct yet complementary datasets focused on heart failure prediction. These datasets serve as foundational pillars, providing us with a rich tapestry of information that spans patient demographics, clinical measurements, and medical history. Each dataset contributes unique dimensions to our understanding, enhancing the depth and breadth of our exploration into the realms of predictive analytics and cardiovascular health research.

1. **Stroke Prediction Dataset** : The dataset we are working with attempts to predict the probability of a stroke in patients by taking into account a number of different input characteristics, including age, gender, heart disease, hypertension, marital status, kind of employment, type of dwelling, average glucose level, BMI, and status of smoking. It addresses the significant issue brought to light by the World Health Organization (WHO), which states that stroke is the second most common cause of death globally, accounting for around 11% of all fatalities. Every row in the collection has extensive patient data, making it possible to identify high-risk patients through thorough AI analysis. Preparing the data to manage missing values, encoding categorical features, correcting class imbalance, and choosing the right machine learning models to assess performance metrics are important factors to take into account while conducting an analysis. This dataset supports the development of predictive models that can aid healthcare professionals in early detection and personalized preventive strategies for stroke.[13]
2. **Heart Failure Prediction** : Cardiovascular disease (CVD) is the leading cause of death globally, accounting for 17.9 million fatalities per year and 31% of all deaths. Strokes and respiratory failures account for four out of every five CVD deaths, with 33% of these deaths happening in people under the age of 70. Eleven components in this dataset can be used to predict the likelihood of coronary diseases, which can help with early detection and treatment of those with high cardiovascular risk factors or CVD. Data from five distinct cardiac datasets are combined in this collection.[12]

3.2 Data Preprocessing

In the Information Preprocessing stage, we have embraced an exhaustive way to deal with refine and set up the Stroke Expectation and Heart Failure datasets for ensuing examination. This elaborate a progression of

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

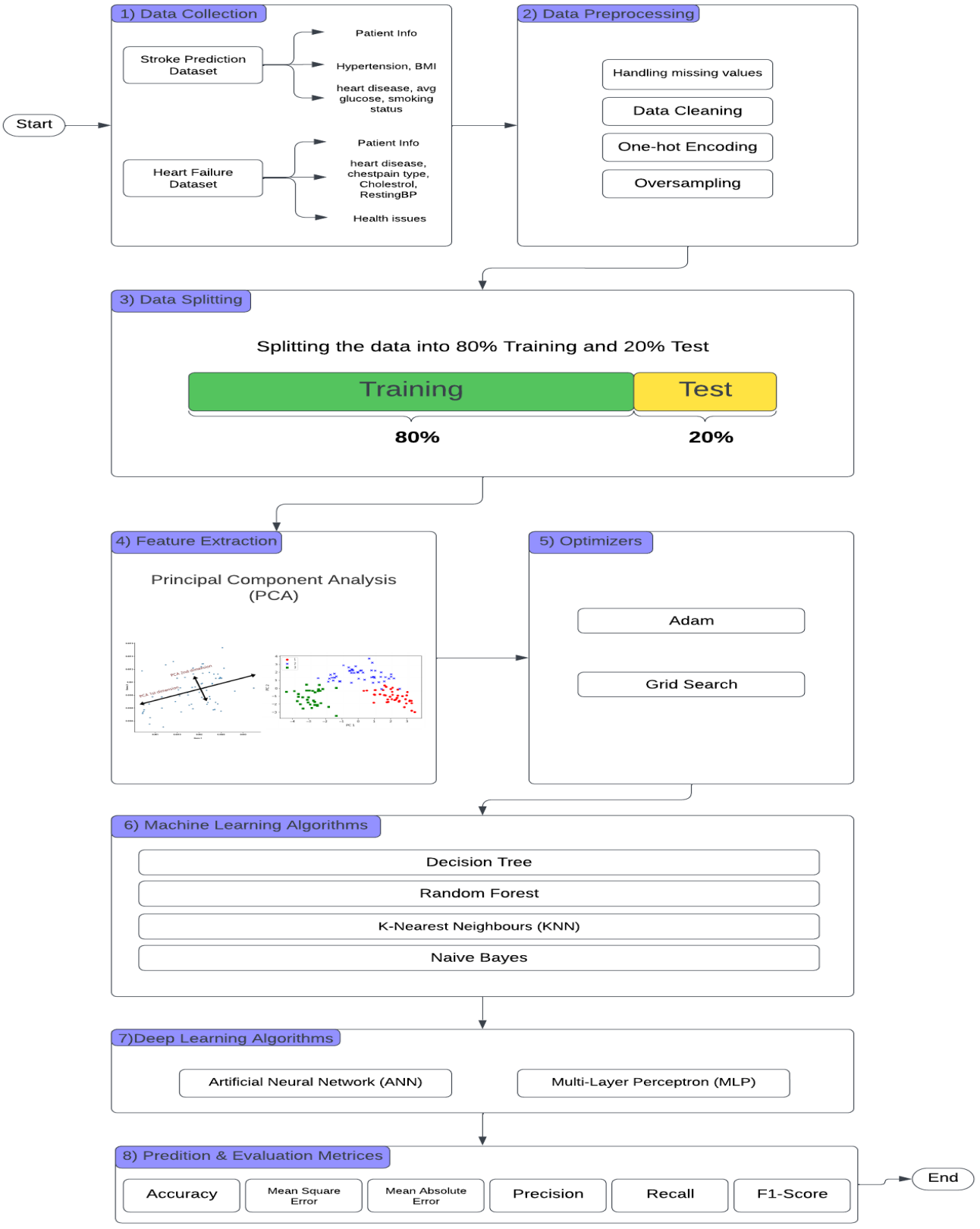


Figure 1: The Proposed System of Heart Failure Prediction .

urgent advances including Information Investigation, Taking care of Missing Qualities, Component Designing, Encoding, and Information Cleaning. The methodology applied is reliable across both datasets; any figures or

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

models displayed in this segment depend on the Stroke Expectation dataset for better comprehension.

Handling Missing Values: Missing values in a dataset can significantly impact the accuracy and reliability of predictive models, making it essential to address them effectively. In our preprocessing phase, we identified and filled missing values using two robust methods to ensure data completeness and quality. First, linear regression was employed to predict and impute missing entries based on other available data, leveraging the relationships within the dataset to provide accurate estimates. Additionally, "NaN" and "Unknown" values were handled using a RandomForest algorithm, which predicts and fills these gaps based on learned patterns from the rest of the data. By addressing missing values through these methods, we enhanced the dataset's integrity, leading to more reliable and precise heart failure prediction outcomes.

1. **Feature Selection:** Feature selection is a crucial step in the data preprocessing phase that aims to enhance the performance of predictive models by identifying and retaining the most relevant features from the dataset. By reducing the number of features, feature selection helps to simplify the model, reduce overfitting, and improve computational efficiency without compromising the model's accuracy. In our approach, we utilized a combination of Principal Component Analysis (PCA) and SelectKBest for feature selection. First, we performed PCA for dimensionality reduction, which transforms the original features into a smaller set of components that capture the most variance in the data. Specifically, we reduced the dataset to 10 principal components. Following this, we applied SelectKBest with the ANOVA F-value method to further refine our feature set by selecting the top 5 most significant features.

2. **Encoding:** Encoding is a fundamental technique in data preprocessing, especially when dealing with categorical variables. It involves converting categorical data into numerical representations that machine learning algorithms can effectively interpret. In our approach, we utilized one-hot encoding to transform every string column in the dataset into a binary format, where each unique category becomes a separate binary feature. This process ensures that the model can properly understand and utilize categorical information without imposing any ordinal relationship between the categories.

3. **Data Cleaning:** Data cleaning is a crucial preliminary step in the data analysis process, aimed at ensuring that datasets are accurate, consistent, and reliable for subsequent analysis or modeling tasks. It involves identifying and rectifying errors, inconsistencies, and missing values within the dataset to improve its quality and integrity. By addressing issues such as duplicates, outliers, and formatting discrepancies, data cleaning lays a solid foundation for meaningful insights to be derived from the data. This introductory phase sets the stage for effective data exploration, analysis, and model building, ultimately enhancing the reliability and validity of the conclusions drawn from the data.

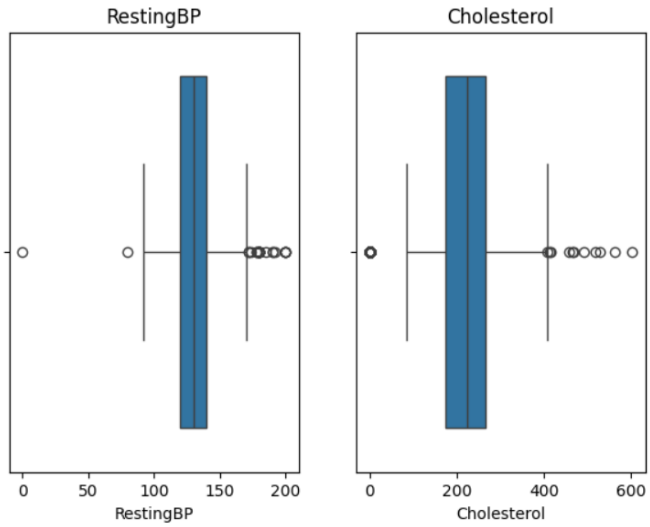
- **Removing Outliers using IQR:** Outliers, often referred to as extreme or abnormal observations, can significantly impact the analysis and interpretation of datasets by skewing statistical measures and distorting the distribution of the data. These anomalies, which deviate substantially from the majority of the data points, can arise due to measurement errors, data entry mistakes, or genuinely rare occurrences. Detecting and appropriately handling outliers is crucial to ensure the integrity and reliability of data analysis outcomes.

  The provided code snippet demonstrates an effective method for outlier detection and removal. By utilizing the Interquartile Range (IQR) method, the function `detect_outliers()` identifies outliers for specified numerical columns in the dataset. Subsequently, these outliers are visualized using boxplots, facilitating a clear understanding of their distribution across different numerical features. The removal of outliers, as executed in the code, helps mitigate their adverse effects on subsequent analyses and modeling tasks.

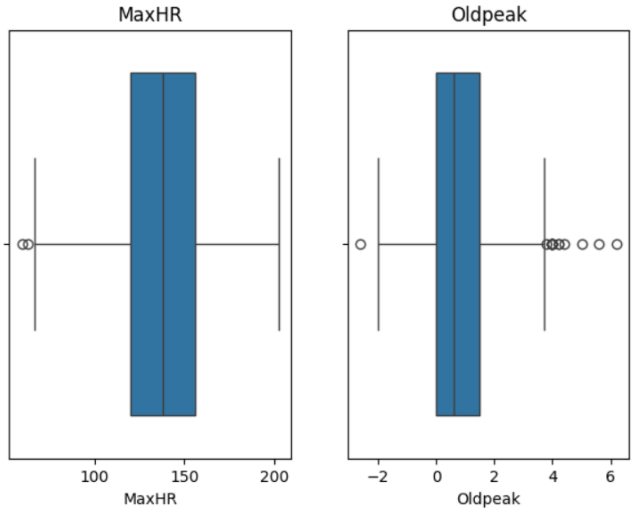
  In the process described, outliers are detected and removed from numerical columns such as 'age', 'bmi', and 'avg_glucose_level'. By iteratively applying the outlier detection function to each numerical column and consolidating the outlier indices, outliers are efficiently identified and removed from the dataset. This ensures that subsequent analyses and modeling efforts are based on a more accurate and representative dataset, ultimately leading to more reliable insights and conclusions.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024



(a) Outliers in RestingBP and Cholesterol.



(b) Outliers in MaxHR and Oldpeak.

3.3 Data splitting

The data splitting process is fundamental in machine learning, enabling the assessment of model performance on unseen data and guarding against overfitting. In the provided code snippet, we employed K-Fold cross-validation, a robust technique that partitions the dataset into 'k' equal-sized folds while ensuring that each fold serves as both a training and testing set. This approach aids in maximizing the utilization of available data for training and validation purposes, mitigating the risk of bias introduced by a single train-test split. The code establishes a KFold object with a specified number of folds, in this case, 5 folds, denoted by 'num_folds'. Each iteration of the for loop represents one fold, where the dataset is divided into training and testing subsets using the indices provided by the KFold splitter. Consequently, 'X_train' and 'y_train' contain the features and target variable, respectively, for training the model, while 'X_test' and 'y_test' represent the corresponding data for evaluation. By iterating through the folds and performing model training and evaluation within each iteration, we ensure comprehensive assessment across different subsets of the data, thereby obtaining robust estimates of model performance. This systematic approach to data splitting enhances the reliability and generalizability of the model, enabling more accurate predictions on unseen data.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

3.4 Hyperparameters Optimization Methods

3.4.1 Hyperparameters Optimization Methods for standard ML techniques

In this step, the hyper-parameter optimization technique is used to determine the optimal value for each parameter of the machine learning models, which are specified as follows. Stratified 5-fold cross-validation in Grid Search:

- **Grid search** Hyperparameter optimization through grid search is a systematic approach to finding the best combination of hyperparameters for a machine learning model. Unlike model parameters, which are learned during the training process, hyperparameters are set prior to training and can significantly impact the model's performance. Grid search involves defining a set of possible values for each hyperparameter and then exhaustively evaluating the model's performance for every possible combination of these values. By using techniques such as cross-validation, grid search ensures that the model is tested on various splits of the data to provide a robust assessment of its performance. This process helps in identifying the optimal hyperparameters that maximize the model's predictive accuracy or minimize a chosen loss function, thereby enhancing the overall effectiveness and reliability of the machine learning model.

3.4.2 Hyperparameters Optimization Methods for the proposed DL models

The performance and convergence of neural networks are significantly shaped by the hyperparameter optimization process in the complex field of deep learning. The optimizer is one of these hyperparameters that is especially important. The Adam optimizer, which is renowned for its adaptive learning rate capabilities, is one notable example in this field. Gaining an in-depth comprehension of the Adam optimizer is essential for utilizing its versatility and improving neural network training and convergence when navigating the intricacies of hyperparameter optimization for deep learning models.

- **Adam optimizer**: Adam In deep learning, optimizing model training using the Adam (Adaptive Moment Estimation) optimizer has become a popular choice due to its efficiency and effectiveness. Adam is an extension of the stochastic gradient descent (SGD) algorithm, combining the advantages of two other extensions: AdaGrad and RMSProp. It maintains per-parameter learning rates that are adapted based on the first and second moments of the gradients, which helps in accelerating convergence and handling sparse gradients. The optimizer calculates an exponentially decaying average of past gradients (momentum) and past squared gradients, allowing it to adjust the learning rate dynamically during training. This adaptability makes Adam particularly well-suited for deep learning tasks, where models often have complex architectures and are trained on large datasets. As a result, Adam tends to converge faster and more reliably compared to other optimization algorithms, making it a preferred choice for training deep neural networks

3.5 Recommendation based on ML and DL Models

The Proposal Frameworks scene has seen momentous head-ways with the combination of both machine and deep learning methods, presenting customized and applicable ideas to clients in different areas. In this section, we explain why we have recommended each ML/DL algorithm .we have introduced machine learning models such as **K-Neighbors Classifier**, DECISION TREE, NAIVE BAYES and RANDOM FOREST .Moreover, we explored various types of neural networks as convolutional neural networks (ANNs), MLP classifier. These approaches, traversing both ML and DL, are used to help in early prediction of heart disease..

DOI: https://doi.org/10.54216/FPA.180113
Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

Machine Learning Models

- **K-Nearest Neighbors (KNN)**

  1. *Distance Metric*: The distance between a query point $\mathbf{x}$ and a training point $\mathbf{x}_i$ is computed using a distance metric, such as Euclidean distance:

  $$d(\mathbf{x}, \mathbf{x}i) = \sqrt{\sum j = 1^m (x_j - x_{ij})^2}$$

  where m is the number of features.

  2. *Finding Neighbors*: Identify the k training samples that are closest to the query point $\mathbf{x}$. Let $\mathcal{N}_k(\mathbf{x})$ represent the set of the k-nearest neighbors of $\mathbf{x}$.

  3. *Classification*: For classification, the predicted class $\hat{y}$ is the mode of the classes of the k-nearest neighbors:

  $$\hat{y} = \arg\max_c \sum_{i \in \mathcal{N}_k(\mathbf{x})} \mathbb{I}(y_i = c)$$

  where $\mathbb{I}(y_i = c)$ is an indicator function that is 1 if $y_i = c$ and 0 otherwise.

  4. *Regression*: For regression, the predicted value $\hat{y}$ is the mean of the values of the k-nearest neighbors:

  $$\hat{y} = \frac{1}{k} \sum_{i \in \mathcal{N}_k(\mathbf{x})} y_i$$

  classifier is a kind of occurrence based learning calculation that is utilized for grouping errands. A straightforward and powerful calculation works by viewing as the "k" Nearest Neighbors tests to another significant piece of information and afterward grouping the new information point in light of the larger part class of those k closest neighbors. The vital thought behind KNN is that comparable cases will quite often have a place with a similar class. In this way, when we need to order another data of interest, we search for the k Neighbors tests that are generally like the new data of interest and afterward utilize the marks of those k Nearest Neighbors to decide the class of the new data of interest. The worth of k is a hyper-parameter that should be picked by the coder. A classifier with a small value of k can be too noisy and sensitive to outliers, while a classifier with a large value of k can be too smooth and ignores important data details. The most common way of choosing k Neighbors in KNN is a significant stage in the calculation. When making a prediction, the number of nearest neighbors to take into account is determined by the value of k. Here is a bit by bit clarification of the cycle Pick a scope of k Neighbors Select a scope of k Neighbors to test, e.g., k = 1, 3, 5, 7, 9, ... Split the information into preparing and testing sets Split the dataset into preparing and testing sets for example in our datasets we split the data into 80% train and 20% test. Train the KNN model for every k worth Train a KNN model for every k worth in the reach, utilizing the preparation information. Assess the model for every k worth Assess the exhibition of each KNN model utilizing measurements like exactness, accuracy, review, and F1-score. Select the ideal k worth Select the k worth that outcomes in the best execution measurements. benefits of KNN is that it is a non-parametric calculation, and that implies that it makes no suspicions about the hidden conveyance of the information. Because of this, the KNN algorithm is adaptable and can be used to solve a wide range of problems.

- **Decision Trees**

  – *Information Gain (IG)* for classification:

  $$IG(S, A) = \mathcal{I}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \mathcal{I}(S_v)$$

  where $\mathcal{I}(S)$ is the impurity measure (e.g., entropy or Gini impurity) of set S.

  – *Variance Reduction* for regression:

  $$\Delta\text{Var}(S, A) = \text{Var}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \text{Var}(S_v)$$

  where $\text{Var}(S)$ is the variance of the target values in set S.

DOI: https://doi.org/10.54216/FPA.180113
Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

- Let $\mathcal{D}$ be the dataset with n samples and m features.
- A Random Forest consists of B decision trees, h_t for $t = 1, 2, \ldots, B$.
- For each tree t:
  - A bootstrap sample $\mathcal{D}_t$ is created by sampling n times with replacement from $\mathcal{D}$.
  - The tree h_t is trained on $\mathcal{D}_t$ using a subset of features selected randomly at each split.
- For a new input $\mathbf{x}$, the Random Forest prediction $\hat{y}$ is given by:
  - For classification:
$$\hat{y} = \text{mode}\left\{h_t(\mathbf{x}) \mid t = 1, 2, \ldots, B\right\}$$
  - For regression:
$$\hat{y} = \frac{1}{B}\sum_{t=1}^{B} h_t(\mathbf{x})$$

Decision Tree is a sort of regulated learning calculation that is for the most part utilized in order issues. It works for both absolute and consistent info and result factors. In this strategy, we split the populace or test into at least two homogeneous sets in view of the main splitter / differentiator in input factors. By inferring straightforward decision rules from the characteristics of the data, the Decision Tree algorithm develops a model that can predict the value of a target variable. A kind of classifier utilizes a tree construction to characterize models. Every inside node in the tree relates to an element or characteristic, each branch addresses a choice rule, and each leaf node addresses a result. The Choice Tree calculation utilizes a hierarchical, voracious way to deal with develop the tree. It begins with the root node, which contains all the preparation information. It then, at that point, chooses the best element to divide the information in view of a rule, for example, data gain or Gini debasement. The information is then parceled into subsets in light of the chose highlight, and the cycle is recursively applied to every subset until a halting rule is met. A minimum number of samples required to split an internal node, a maximum tree depth, or a minimum improvement in the criterion are examples of factors that can serve as the basis for the stopping criterion. When the tree is developed, creating forecasts on new data can be utilized. We start at the root node and follow the branches that correspond to the values of the input features until we reach a leaf node in order to predict the class of a new example. The class mark related with the leaf node is then doled out to the new model. To improve the model we can change in some of the models hyper parameters such as

1. **max_depth**: regulates the tree's maximum depth

2. **min_samples_split**: regulates the bare minimum of samples needed to divide an internal node.

3. **min_samples_leaf**: regulates the bare minimum of samples needed to qualify as a leaf node.

4. **max_leaf_nodes**: regulates the tree's maximum number of leaf nodes

5. **min_impurity_decrease: controls** the smallest reduction in impurity needed to cause a split.

You can control the complexity of the decision tree and prevent either over-fitting or under-fitting by adjusting these hyper-parameters. It is essential to keep in mind that the ideal hyper-parameters will be determined by the particular dataset and issue at hand. You can use grid search or random search to search over a range of hyper-parameter values and evaluate the decision tree's performance on a validation set to find the best hyper-parameter values. You can then choose the hyper-parameters that give you the best exhibition on the approval set.

Decision Trees enjoy a few benefits, for example, their capacity to deal with both mathematical and unmitigated information, their interpretability, and their capacity to deal with missing qualities. In any case, they can likewise be inclined to over-fitting, particularly when the tree is permitted to become excessively profound. To resolve this issue, procedures like pruning, cross-approval, and troupe techniques can be utilized.

- **Naïve bayes:**

1. *Bayes' Theorem*: For a given class C and a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the posterior probability is given by:
$$P(C \mid \mathbf{x}) = \frac{P(C)P(\mathbf{x} \mid C)}{P(\mathbf{x})}$$

where $P(C)$ is the prior probability of class C, $P(\mathbf{x} \mid C)$ is the likelihood, and $P(\mathbf{x})$ is the evidence.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

2. *Naive Independence Assumption*: Assume that the features are conditionally independent given the class C:

$$P(\mathbf{x} \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

3. *Classification Rule*: The predicted class $\hat{C}$ is the one that maximizes the posterior probability:

$$\hat{C} = \arg\max_{C} P(C \mid \mathbf{x}) = \arg\max_{C} P(C) \prod_{i=1}^{n} P(x_i \mid C)$$

The essential thought behind Naïve Bayes is to utilize the likelihood of each component given the class mark to foresee the class name of another occurrence. The initial step is to set up the information by parting it into preparing and testing sets. Extracting features from the training data is the next step. Each element ought to be a mathematical worth that addresses some part of the information. Compute Earlier Probabilities is to ascertain the earlier probabilities of each class mark. This should be possible by counting the quantity of occurrences of each class name in the preparation information and isolating by the complete number of cases. Work out Probability Probabilities to compute the probability probabilities of each component given the class mark. This can be accomplished by dividing the total number of instances of each class label by the number of instances of each feature value the last step is compute Back Probabilities to work out the back probabilities of each class mark given the element upsides of another occurrence. This should be possible utilizing Naïve Bayes' theorem, which expresses that the back likelihood of a class mark given the component values is corresponding to the result of the earlier likelihood of the class name and the probability probabilities of the element values given the class name. Predict Class Label with the highest posterior probability is predicted after the posterior probabilities have been calculated.

Utilizing feature selection methods to select the features that are most relevant to the model is one common strategy. This can assist with diminishing the dimensionality of the information and work on the exhibition of the model. Some famous element choice techniques incorporate chi-squared test, shared data, and recursive component end. Another method is to utilize an alternate variation of Guileless Bayes, like Multinomial Gullible Bayes or Bernoulli Innocent Bayes. These variations are more qualified for various kinds of information and can work on the presentation of the model. Additionally, using a method known as "smoothing" can help Naive Bayes perform better. Smoothing increases the value of the recurrence of each component to keep away from zero probabilities. This can assist with keeping the model from making careless forecasts and work on its general execution. At last, tuning the hyper-parameters of the Naïve Bayes model can likewise assist with working on its exhibition. Some normal hyper-parameters to tune incorporate the smoothing boundary and the earlier probabilities. Lattice search or irregular pursuit can be utilized to track down the ideal mix of hyper-parameters.

- **Random forest**
  - Let $\mathcal{D}$ be the dataset with n samples and m features.
  - A Random Forest consists of B decision trees, h_t for $t = 1, 2, \ldots, B$.
  - For each tree t:
    - A bootstrap sample $\mathcal{D}_t$ is created by sampling n times with replacement from $\mathcal{D}$.
    - The tree h_t is trained on $\mathcal{D}_t$ using a subset of features selected randomly at each split.
  - For a new input $\mathbf{x}$, the Random Forest prediction $\hat{y}$ is given by:
    - For classification:

$$\hat{y} = \text{mode}\left\{h_t(\mathbf{x}) \mid t = 1, 2, \ldots, B\right\}$$

    - For regression:

$$\hat{y} = \frac{1}{B} \sum_{t=1}^{B} h_t(\mathbf{x})$$

  Random Forest is a popular machine learning algorithm that belongs to the family of ensemble methods. It is an important algorithm that can be used for both classification and regression datasets. The idea behind Random Forest is to build a collection of decision trees and combine their predictions to produce a more accurate and stable prediction. Random forest is similar to

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

the decision tree method, but it is better as it combines predictions from various decision trees. Hyperparameters like the number of trees, the maximum depth of the trees, and the amount of features to take into account at each node can all be tuned to enhance the Random Forest model's performance. The number of estimators is the most crucial parameter for enhancing the model's performance since it determines how many trees are in the forest. While adding more trees can boost the model's performance, doing so comes at a higher computational cost. Set the number of estimators to 100 as a suitable starting point, and keep increasing it until the model's performance no longer improves.

– **Deep Learning Models**

  * **Artificial Neural Network (ANN)**
  Artificial Neural Network (ANN), the way biological neural networks in the human brain process information is the model of a computing model known as an Artificial Neural Network (ANN). The layers of an artificial neural network (ANN) are made up of networked nodes, or neurons. By training, or adjusting the weights of connections between neurons to minimize prediction errors, these networks can learn from data. Classification, regression, and pattern recognition are among the tasks where artificial neural networks (ANNs) excel. They are extremely useful in areas like medical diagnosis, image and speech recognition, and financial forecasting because they can make use of vast datasets to capture intricate relationships and patterns.

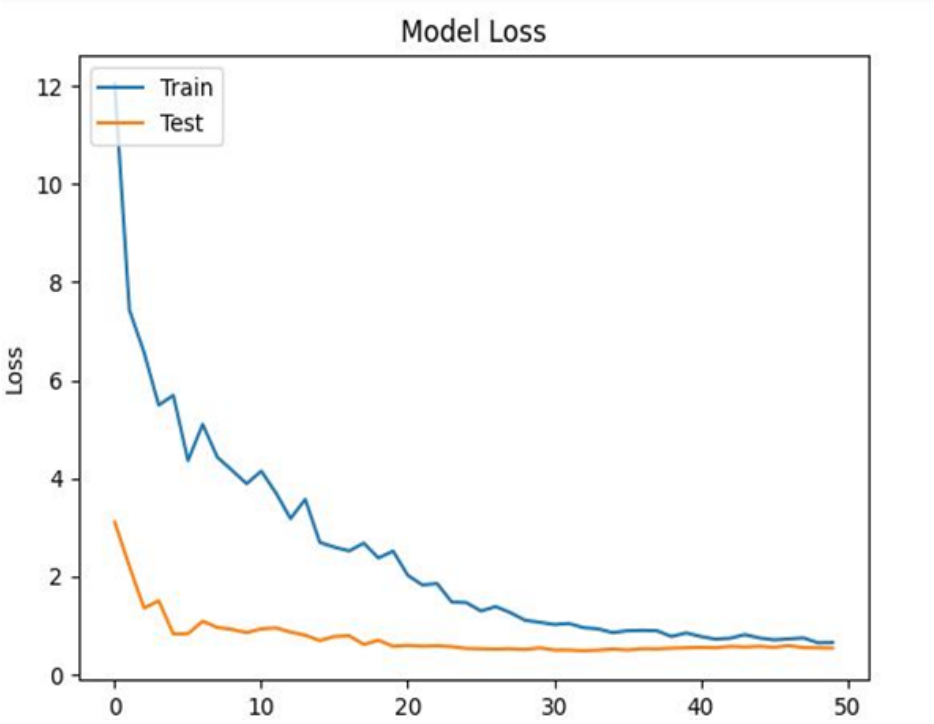


Figure 3: LOSS OF DEEP LEARNING MODEL FOR DATASET 1

  * **Input Layer** :The input layer of an ANN is the primary layer that gets the underlying information. Every neuron in the info layer addresses a component from the dataset, meaning the quantity of neurons in this layer compares to the quantity of information highlights. The input layer plays out no calculations; all things considered, it passes the information onto the following layer for handling. With regards to anticipating respiratory failure risks, the input layer takes in different clinical highlights of patients, for example, age, cholesterol levels, pulse, Blood pressure and other relevant health metrics..
  * **Hidden Layers** : Hidden layers in an ANN are moderate layers between the info and result layers where the genuine calculation and change of information happen. These layers comprise of neurons that apply initiation capabilities to the data sources they get, permitting the organization to learn non-straight connections. In the portrayed model, there are three secret layers with 16, 8, and 4 neurons separately, each utilizing the ReLU (Rectified linear Unit)

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

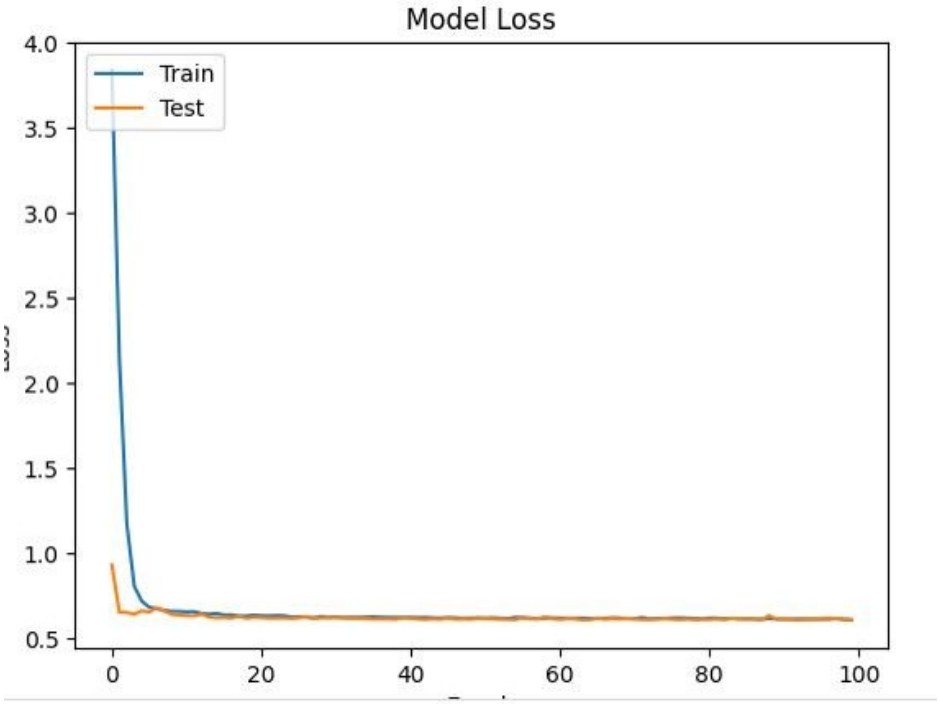


Figure 4: LOSS OF DEEP LEARNING MODEL FOR DATASET 2

actuation capability. Moreover, dropout layers are utilized after the second and third secret layers to relieve overfitting by haphazardly deactivating a small part of neurons during preparing, in this manner working on the model's capacity to sum up to new, concealed information.

* **Output Layer**: The output layer of an ANN is the final layer that creates the organization's expectations. It takes the changed information from the last hidden layer and applies an actuation capability to produce the last result. For double order undertakings, for example, anticipating the probability of a heart attack , the output layer commonly comprises of a solitary neuron with a sigmoid initiation capability. This design yields a likelihood esteem somewhere in the range of 0 and 1, which can be deciphered as the probability of the positive class (e.g., heart attack occurrence). The limit is then applied to order the result into double classes.

– **A Multi-facet Perceptron**

A Multi-facet Perceptron (MLP) is a class of feedforward artificial neural networks that comprises of various layers of neurons, each completely associated with the following. MLPs are especially powerful for taking care of intricate issues that include design acknowledgment, grouping, and relapse errands. The design commonly incorporates an input layer, at least one hidden layers, and a output layer. Every neuron in these layers processes inputs through a weighted total followed by a non-straight enactment capability. MLPs advance by changing the loads through a cycle called backpropagation during preparing. This flexibility makes MLPs integral assets for a great many applications, from picture and discourse acknowledgment to clinical finding and monetary guaging.

* **Input Layer** :The input layer of a MLP is the primary layer that gets crude information from the dataset. Every neuron in the input layer relates to a solitary element of the input information. Consequently, the quantity of neurons in this layer rises to the quantity of elements. The input layer plays out no estimations except for fills in as a conductor to pass the information to the output layers. For example, in a model intended to anticipate coronary failure gambles, the input layer would take different clinical boundaries, for example, age, cholesterol levels, circulatory strain, and other wellbeing pointers.
* **Hidden Layers** : In an MLP, hidden layers are middle layers that are located in between the input and output layers. To model complicated interactions within the data, the network needs these layers in order to function. In order to introduce non-linearities and help the network learn complex patterns, each hidden layerto their inputs. The MLP model under discussion

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

comprises three hidden layers, each containing 16, 8, or 4 neurons. The activation function of each layer is the Rectified Linear Unit (ReLU). In order to shield against overfitting, dropout layers are inserted after the second and third concealed layers. To make sure the model more effectively generalizes to unseen data, the dropout mechanism randomly disables a percentage of neurons during training.

* **Output Layers**: An MLP's output layer is the last layer that produces predictions for the network. It employs an activation function after receiving processed data from the final hidden layer to generate the desired output. Typically, the output layer of binary classification tasks has a single neuron with a sigmoid activation function, such as estimating the probability of a heart attack. The probability of the positive class is indicated by a probability score in this arrangement, which ranges from 0 to 1. After that, the model can group the data according to a threshold, classifying predictions into binary outcomes like the likelihood of a heart attack or not.

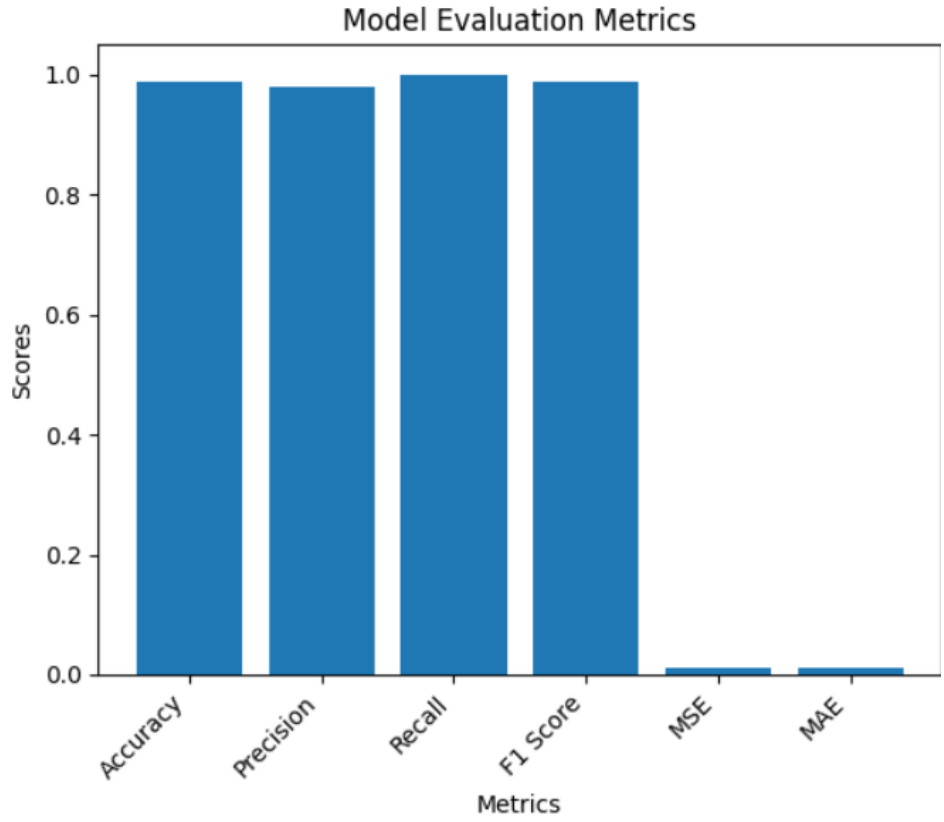


Figure 5: EVALUATION METRICS FOR MACHINE LEARNING MODEL FOR DATASET 1

3.6 Performance Evaluation Metrics

When assessing recommendation systems, three commonly utilized metrics are Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared. These metrics offer quantitative measures for gauging the accuracy and efficacy of models in predicting user preferences.

- **Mean Squared Error (MSE)** MSE quantifies the average squared disparity between predicted and actual ratings.It's computed by averaging the squared errors for each prediction. The formula for MSE is:

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

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

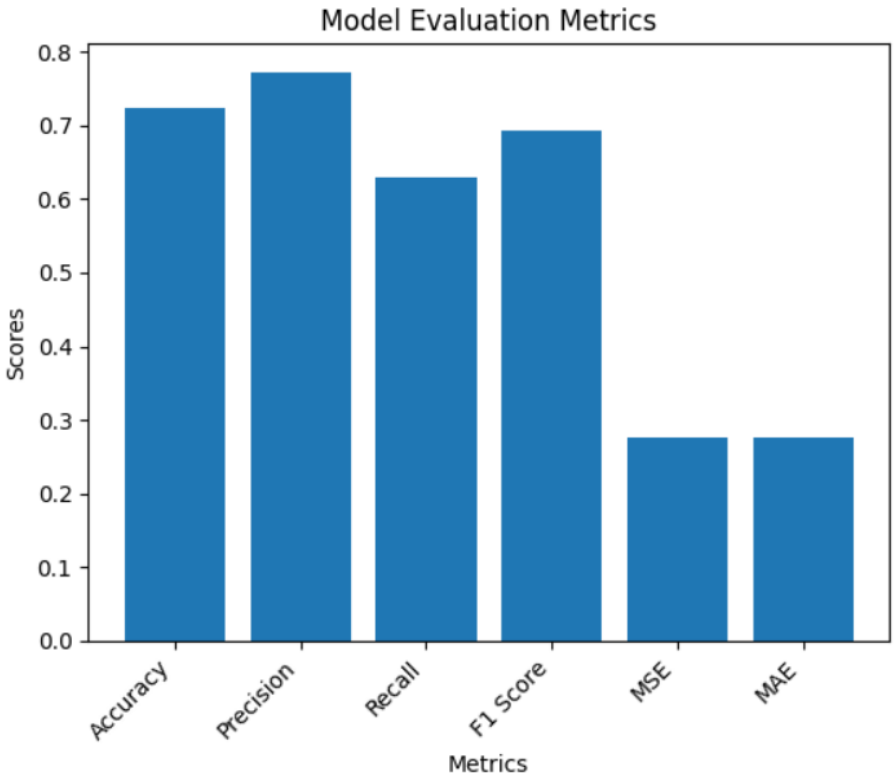


Figure 6: EVALUATION METRICS FOR MACHINE LEARNING MODEL FOR DATASET 2

where n is the number of predictions, y_i represents the actual rating, and $\hat{y}_i$ is the predicted rating. A lower MSE indicates better accuracy, with zero representing a perfect prediction.

- **Mean Absolute Error (MAE)** MAE calculates the average absolute deviation between predicted and actual ratings. It's determined by averaging the absolute differences. The formula for MAE is:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

where n is the number of predictions, y_i represents the actual rating, and $\hat{y}_i$ is the predicted rating. Similar to MSE, a lower MAE indicates better accuracy, with zero representing a perfect prediction.

- **R-squared (Coefficient of Determination)** R-squared quantifies the proportion of variance in the dependent variable (actual ratings) that can be anticipated from the independent variable (predicted ratings). It ranges from 0 to 1, with 1 indicating a perfect fit. The formula for R-squared is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3}$$

where $\hat{y}$ is the mean of the actual ratings, and n is the number of predictions. A higher R-squared value signifies a better model fit.

4 Experimental Results and Discussion

In the phase of experimental results and discussion, our predictive models for stroke prediction and heart failure exhibited robust performance across two distinct datasets: the Stroke Prediction Dataset and the Heart Failure Dataset. We employed a variety of machine learning approaches, including K-Nearest Neighbors (KNN),

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

Table 2: Results for Dataset1 and Dataset2. Tables should be placed in the main text near to the first time they are cited.

Evaluation Matrices	**RF**	**NB**	**DT**	**K-NN**
Accuracy	0.9897 , 0.7234	0.62079 , 0.7092	0.68930 , 0.62	0.94471 , 0.6738
MSE	0.01021 , 0.2765	0.379206 , 0.2907	0.31069 , 0.3870	0.05528 , 0.3262
MAE	0.01021 , 0.2765	0.379206 , 0.2907	0.31069 , 0.3870	0.05528 , 0.3262
Precision	0.9796 , 0.7719	0.7561 , 0.7101	0.6434 , 0.6271	0.89890 , 0.7068
Recall	1 , 0.6285	0.33740 , 0.7	0.82518 , 0.8031	1 , 0.5857
F1-Score	0.9897 , 0.6929	0.4666 , 0.7050	0.7230 , 0.7320	0.94675 , 0.6406

Naive Bayes, Random Forest, and Decision Tree. Additionally, in the realm of deep learning, we utilized Artificial Neural Networks (ANN) and Multilayer Perceptron (MLP). To address class imbalance, oversampling techniques were implemented. Performance evaluation was conducted using key metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared, providing a comprehensive assessment of the models' accuracy and effectiveness. The utilization of these metrics underscores their versatility and relevance in evaluating predictive models across different domains. Our investigation into these methodologies underscores their potential for further exploration and application in predictive modeling for medical conditions like stroke prediction and heart failure. The incorporation of oversampling techniques further enhances the robustness of our models in handling imbalanced datasets

4.1 Case Study I (Stroke Prediction Dataset)

The stroke prediction system underwent a comprehensive evaluation utilizing five different algorithms: K-Nearest Neighbors (KNN), Naive Bayes, Random Forest, Decision Tree, Artificial Neural Networks (ANN), and Multilayer Perceptron (MLP). Random Forest, with its best performance, achieved an MSE of 0.008 and MAE of 0.008.

These diverse performance metrics highlight the strengths and weaknesses of each algorithm, emphasizing the importance of a comprehensive evaluation when selecting an appropriate predictive model. In this comparison, Random Forest showcased the lowest MSE and MAE, suggesting superior predictive accuracy compared to other algorithms.

4.2 Case Study II (Heart Failure Dataset)

In the evaluation of the Heart Failure dataset, both machine learning and deep learning approaches were employed without oversampling. Grid search and k-folds cross-validation techniques were utilized to optimize model performance. The same set of algorithms, including K-Nearest Neighbors (KNN), Naive Bayes, Random Forest, Decision Tree, Artificial Neural Networks (ANN), and Multilayer Perceptron (MLP), were tested.

The best result achieved without k-folds was obtained with Random Forest, yielding an accuracy of 73%. With the inclusion of k-folds cross-validation and grid search, Random Forest achieved a significantly improved accuracy of 94%. Further analysis and comparison of the performance of each algorithm with and without k-folds cross-validation underscore the importance of robust evaluation techniques in optimizing predictive model accuracy for heart failure prediction.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

5 Conclusion and future work

5.1 Case Study I (Stroke Prediction Dataset)

In the comparison of algorithms for stroke prediction, the Random Forest algorithm stood out with the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE) among the evaluated methods. Despite its high predictive accuracy, it's essential to consider that the choice of the most suitable algorithm depends on various factors, including dataset characteristics and the nuanced nature of stroke risk factors. Thus, a comprehensive understanding of these factors is crucial for making informed decisions in algorithm selection.

5.2 Case Study II (Heart Failure)

The optimal method for a heart failure prediction model would be the one with the lowest Mean Squared Error (MSE). These metrics show how well the model predicts outcomes and how well it can identify underlying patterns in the data. The Random Forest model has the lowest average squared difference (MSE) between the predicted and actual results in this instance (0.027). The R-squared score also sheds light on the percentage of variation in the actual results that the model can forecast. Therefore, the Random Forest model would be regarded as the best-performing algorithm among those studied for a heart failure prediction model due to its low MSE and high R-squared.

Author contributions

References

[1] S. Shivadekar, K. Shahapure, S. Vibhute, A. Dunn, "Evaluation of Machine Learning Methods for Predicting Heart Failure Readmissions: A Comparative Analysis", International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 6s, pp. 694–699, 2024.

[2] M. Kavitha, G. Gnaneswar, R. Dinesh, Y. R. Sai, R. S. Suraj, "Heart disease prediction using hybrid machine learning model", Proc. of 2021 6th International Conference on Inventive Computation Technologies (ICICT), pp. 1329–1333, 2021, IEEE.

[3] P. Rani, R. Kumar, N. M. O. Sid Ahmed, A. Jain, "A decision support system for heart disease prediction based upon machine learning", Journal of Reliable Intelligent Environments, vol. 7, no. 3, pp. 263–275, 2021, Springer.

[4] V. Sharma, S. Yadav, M. Gupta, "Heart disease prediction using machine learning techniques", Proc. of 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 177–181, 2020, IEEE.

[5] H. Jindal, S. Agrawal, R. Khera, R. Jain, P. Nagrath, "Heart disease prediction using machine learning algorithms", IOP Conference Series: Materials Science and Engineering, vol. 1022, no. 1, p. 012072, 2021, IOP Publishing.

[6] A. Singh, R. Kumar, "Heart disease prediction using machine learning algorithms", Proc. of 2020 International Conference on Electrical and Electronics Engineering (ICE3), pp. 452–457, 2020, IEEE.

[7] D. Shah, S. Patel, S. K. Bharti, "Heart disease prediction using machine learning techniques", SN Computer Science, vol. 1, no. 6, p. 345, 2020, Springer.

[8] R. Katarya, S. K. Meena, "Machine learning techniques for heart disease prediction: A comparative study and analysis", Health and Technology, vol. 11, no. 1, pp. 87–97, 2021, Springer.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024

[9] C. M. Bhatt, P. Patel, T. Ghetia, P. L. Mazzeo, "Effective heart disease prediction using machine learning techniques", Algorithms, vol. 16, no. 2, p. 88, 2023, MDPI.

[10] A. Okada, H. Kaneko, M. Konishi, K. Kamiya, T. Sugimoto, S. Matsuoka, I. Yokota, Y. Suzuki, S. Yamaguchi, H. Itoh, et al., "A machine-learning-based prediction of non-home discharge among acute heart failure patients", Clinical Research in Cardiology, vol. 113, no. 4, pp. 522–532, 2024, Springer.

[11] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, P. Singh, et al., "Prediction of heart disease using a combination of machine learning and deep learning", Computational Intelligence and Neuroscience, vol. 2021, 2021, Hindawi.

[12] Fedesoriano, "Heart Failure Prediction", available: `https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction`, 2023.

[13] Fedesoriano, "Stroke Prediction Dataset", available: `https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset`, 2023.

DOI: https://doi.org/10.54216/FPA.180113

Received: July 05, 2024 Revised: October 04, 2024 Accepted: December 28, 2024