

PREDICTION OF HEART DISEASE AT EARLY STAGE USING MACHINE LEARNING ALGORITHMS

A Project Report Phase – II

Submitted in partial fulfillment of the requirements for the award of a Bachelor
of Engineering Degree in Electronics and Communication Engineering

by

PAVAN KUMAR PAMARTHI (39130337)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING SCHOOL OF
ELECTRICAL AND ELECTRONICS**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC
**JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119**

APRIL – 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of PAVAN KUMAR PAMARTHI (39130337) who carried out the project entitled "**PREDICTION OF HEART DISEASE AT EARLY STAGE USING MACHINE LEARNING ALGORITHMS**" under my supervision from November 2022 to April 2023.

Internal Guide

Dr . K. SRILATHA, M.E, Ph.D.,

Head of the Department

Dr. T. RAVI M.E., Ph.D.,

Submitted for Viva-voce Examination held on 20.04.2023

Internal Examiner

External Examiner

DECLARATION

I PAVAN KUMAR PAMARTHI (39130337) hereby declare that the Project Report entitled **“PREDICTION OF HEART DISEASE AT EARLY STAGE USING MACHINE LERNING ALGORITHMS”** done by me under the guidance of **Dr. K. SRILATHA M.E, Ph.D., Associate Professor** at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE: 20-04-2023

PLACE: Chennai

SIGNATURE OF THE CANDIDATES

1. Pavan Kumar Pamathi

ACKNOWLEDGEMENT

I was pleased to acknowledge my sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I'm grateful to them.

I convey my thanks to **Dr. N. M. NANDHITHA M.E, Ph.D.**, Dean, School of Electronics and Communication Engineering, and **Dr. T. RAVI M.E, Ph.D.**, Head of the Department, Department of Electronics and Communication Engineering for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. K. SRILATHA, M.E, Ph.D., Associate Professor, Department of Electronics and Communication Engineering** for his valuable guidance, suggestions, and constant encouragement that paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Global risks to public health include cardiovascular and persistent respiratory conditions, which account for roughly 19 million global fatalities each year. With the help of technical developments in medical science, such as constant tracking of physiological factors like blood pressure, cholesterol levels, blood glucose, etc., this high fatality rate can be decreased. The cutting-edge values of these vital physiological or vital sign metrics not only allow for in-the-moment assistance from medical professionals and caretakers, but they also assist patients in managing their health state by getting pertinent, timely notifications and guidance from healthcare professionals. In this research, we suggest a machine-learning-based forecast and categorization method to identify associated vital signs' future values for both chronic respiratory and cardiovascular illnesses.

The suggested system can categorize patients' health state based on the forecast of future values and alert carers and medical professionals. We used an actual vital sign collection in this machine-learning-based prediction and categorization algorithm. Several regression techniques, including linear regression and polynomial regression of degrees 2, 3, and 4, have been tried to forecast the vital sign values for the upcoming 1–3 minutes. Vital sign predictions are used for carers in 60-second intervals and for emergency medical help in 3-minute intervals. The patient's general health is evaluated using three machine learning classifiers, namely Support Vector Machine (SVM), Naive Bayes, and Decision Tree, Random Forest, K-Nearest neighbor based on the expected vital sign readings. Our findings demonstrate that the Decision Tree is useful in providing patients with prompt medical treatment and can accurately categorize a patient's health state based on aberrant vital sign values. Choice Tree.

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	5-9
	2.1 Inferences from Literature Survey	
	2.2 Open Problems in Existing System	9-11
3	EXISTING METHOD	
	3.1 System Design	12
	3.2 Functional Architecture	13-14
	3.3 Modular Design	15-19
	3.4 System Requirements	19-21
4	PROPOSED METHOD	22
	4.1 Overview of the platform	
	4.2 Module Implementation	23-33
	4.2.1 Optimization of Model	
	4.2.2 Testing the Model	
	4.2.3 Refinement and Deployment of Model	
	4.2.4 Output	

5	RESULTS AND DISCUSSION	
5.1	Test Cases	34
5.2	Performance Measures	35
5.3	Performance Analysis	36-38
6	CONCLUSION AND FUTURE SCOPE	
6.1	Conclusion	39
6.2	Future scope	40
	REFERENCES	41

LIST OF FIGURES

	Figure Name	Page Number
3.1	System Architecture	12
3.2	Functional Architecture	13
3.3	Use case Diagram	15
3.4	Activity Diagram	16
3.5	Class Diagram	17
3.6	Sequence Diagram	18
3.7	Component Diagram	19
4.1	Data Processing	25
4.2	Model Training	26
4.2.1	Model Training	26
4.2.2	Model Training	27
4.3	Model Evaluation	28
4.3.1	Model Evaluation	28
4.4	Output	32
4.4.1	Output	32
5.1	Performance Analysis	37

CHAPTER 1

INTRODUCTION

Around the world, machine learning is used in many different industries. The healthcare industry is no exception. Machine learning can be quite useful in determining whether or not ailments like locomotor disorders, cardiovascular diseases, and others will exist. If foreseen, such evidence can provide clinicians with insightful knowledge, enabling them to customize their treatment plans and diagnoses. The heart is one of the body's primary organs. It propels blood through the circulatory system's blood vessels. The circulatory system is crucial because it distributes blood, oxygen, and other substances to the body's many organs.

The heart is the most important component of the circulatory system. If the heart does not operate properly, it can result in major health problems, including death. The healthcare sector has vast amounts of medical data; consequently, machine learning algorithms are essential for accurate heart disease prediction. Recent studies have focused on combining these strategies to create hybrid machine learning algorithms. Data pre-processing is used in the research proposal to remove noisy data, fill in blanks when necessary, fill in default values when appropriate, and categorize attributes for prediction and decision-making at multiple levels. To assess the efficacy of the treatment approach, techniques including classification, accuracy, sensitivity, and specificity analysis are performed. An accurate cardiovascular disease prediction model is demonstrated by comparing the levels of accuracy of applying rules to the outcome variables of SVM Classifier, Decision Tree, Logical Regression, Random Forest, and Multinomial Nave Bayes. Heart stroke, also known as a cerebrovascular accident, is a serious medical condition that can cause long-term disability or even death. Early prediction of heart stroke is crucial to prevent or minimize its adverse effects. Risk factors for heart stroke include high blood pressure, smoking, high cholesterol, diabetes, obesity, and family history.. By taking these preventive measures and the risk of heart stroke can be significantly reduced.

MACHINE LEARNING METHODS

The two primary categories of machine learning methods are unsupervised and supervised.

For input and the intended output, supervised algorithms require supervision by a person with machine learning expertise. The method will then be used on fresh data after the model training is finished. Unsupervised algorithms don't need any data training. They do, nonetheless, employ an iterative process. Deep Learning is the name of this strategy.

Artificial neural network is another name for unsupervised learning techniques. These networks are employed in situations where there is greater complexity because they are more versatile than supervised learning systems. Such neural networks advance by sifting through the training set and automatically identifying associations between both the dataset's parameters.

1. Supervised machine learning algorithms are applied to the previously studied data in the past and then to new data. Such a system is able to provide outputs for any new input once sufficient training is done. This algorithm also compares its output with the correct, intended output and finds discrepancies, and then modifies the model accordingly.
2. On the other hand, unsupervised machine learning algorithms have been when the information has neither been classified nor labelled. This category studies how systems can make inferences into a function that describes a hidden structure in unlabeled data. But there is a drawback, this system doesn't tell the right output, rather it explores the data and draws inferences from datasets.
3. Semi-supervised machine learning algorithms lie in between the above- mentioned systems. This is because they use both labeled and unlabeled data for training purposes –a smaller amount of labelled data and a larger amount of unlabeled data in combination. This system uses techniques that are able to improve Accuracy.

PROBLEM DESCRIPTION

Heart stroke prediction is a problem in which the goal is to predict whether a person is at risk of having a stroke. A stroke occurs when there is a disruption in the blood supply to the brain, which can lead to brain damage or even death. Identifying individuals who are at high risk of having a stroke can help in taking preventative measures, such as lifestyle modifications or medication, to reduce the risk of stroke occurrence.

The problem of heart stroke prediction can be approached as a binary classification problem, where the input is a set of features describing an individual's medical history, lifestyle, and demographic information, and the output is a binary label indicating whether the individual is at risk of having a stroke or not. The task is to build a predictive model that can accurately classify individuals into these two categories based on their input features.

The input features for heart stroke prediction may include age, gender, blood pressure, body mass index, smoking status, alcohol consumption, history of heart disease, diabetes, and other

medical conditions. The dataset for this problem can be collected from electronic health records, surveys, or other sources, and it should be large enough to allow for the development of a robust predictive model.

The performance of a heart stroke prediction model can be evaluated using metrics such as accuracy, precision, recall, and F1-score. A high-performing model can help in identifying individuals at high risk of having a stroke, which can aid in early intervention and prevention. By leveraging these technologies, healthcare professionals can identify patients who are at a higher risk of heart stroke and provide target interventions to prevent or minimize its impact, ultimately improving patient outcomes and reducing the burden of heart stroke on society.

OBJECTIVE

1. The main objective of this research is to develop a heart stroke prediction system; the system can discover and extract hidden knowledge associated with heart strokes from the data set.
2. The other objective is to implement machine learning techniques on heart-d data sets to assist in the prediction of heart stroke.
3. The objective of heart stroke prediction is to identify individuals who are at high risk of having a stroke and provide them with early intervention and prevention measures to reduce the risk of stroke occurrence.
4. This can improve patient outcomes, help healthcare providers prioritize high- risk patients for targeted interventions, and contribute to the development of personalized treatment plans based on an individual's risk factors.
5. Additionally, heart stroke prediction can support clinical research by providing insights into the factors that contribute to stroke occurrence and identifying new avenues for prevention and treatment.
6. Overall, the goal of heart stroke prediction is to reduce the overall burden of stroke on the population and increase awareness about the risk factors associated with stroke and the importance of prevention among the general public.

SCOPE OF THE PROJECT

Risk factor identification: A heart stroke project could focus on identifying the most important risk factors for stroke, such as high blood pressure, smoking, and diabetes. This might involve analyzing large datasets of patient information to determine which factors are most strongly associated with stroke risk.

Development of predictive models: A heart stroke project could involve developing predictive models to estimate an individual's risk of stroke based on their personal characteristics and medical history. This could involve using machine learning algorithms or statistical models to analyze patient data.

Prevention strategies: A heart stroke project could focus on developing and testing interventions to prevent stroke. This might involve testing the effectiveness of lifestyle modifications, such as diet and exercise, or the use of medications to lower blood pressure or cholesterol.

Early detection and treatment: A heart stroke project could focus on improving early detection and treatment of stroke. This might involve developing new diagnostic tools or therapies, or evaluating the effectiveness of existing treatments.

Patient outcomes: A heart stroke project could focus on improving patient outcomes following stroke. This might involve evaluating rehabilitation strategies or developing new methods for monitoring patient progress.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

A literature review is a survey of scholarly sources on a specific topic. It provides an overview of current knowledge, allowing you to identify relevant theories, methods, and gaps in the existing research. A literature review identifies, evaluates and synthesizes the relevant literature within a particular field of research. Literature refers to a collection of published information or materials on a particular area of research or topic, such as books and journal articles of academic value. Literature survey is the documentation of a comprehensive review of the published and the unpublished work from the secondary sources data in the areas of specific interest to the researcher. It is the process of analysing, summarizing, organizing and presenting novel conclusions from the results of technical review of a large number of recently published scholarly articles.

A.Singh, et.al., [1],“Heart Disease Prediction using Machine Learning and Data Mining Technique”. Heart disease prediction is done using Naïve Bayes and Genetic algorithms. The model has been trained on a UCI dataset with attributes like gender, age, resting blood pressure, cholesterol, fasting blood sugar, old peak, etc. It is a web-based machine learning application where the user inputs his medical details based on these attributes to predict his heart disease. The algorithm calculates the probability of having a heart disease and the result is displayed on the web page itself. The study is based on a dataset collected from the Cleveland Clinic Foundation in the USA, which consists of 303 patients with 14 features including age, sex, blood pressure, cholesterol levels, and others.

The authors employ four machine learning algorithms, namely decision tree, k- nearest neighbor, random forest, and support vector machine, to build heart disease prediction models. The performance of each model is evaluated using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).

The results indicate that the support vector machine algorithm outperforms the other three algorithms in terms of accuracy, sensitivity, specificity, and AUC. The study concludes that machine learning algorithms can be effectively used for the prediction of heart disease, and support vector machine is a suitable algorithm for this task. The authors suggest that further studies are needed to validate the results on larger datasets and to develop more accurate prediction models.

K. Hashi, et.al.,[2], "Heart Disease Prediction using Machine Learning Techniques". various classification algorithms are studied and the most accurate model is obtained for predicting the heart disease in the patient. It was found that Random Forest and XGBoost were the most efficient algorithms while K- Nearest Neighbor was found to be the most ineffective one. The study is based on a dataset collected from the UCI Machine Learning Repository, which consists of 303 patients with 14 features including age, sex, blood pressure, cholesterol levels, and others.

The authors employ four machine learning algorithms, namely logistic regression, decision tree, random forest, and support vector machine, with hyperparameter tuning to build heart disease prediction models. The performance of each model is evaluated using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).

The results indicate that the performance of all four algorithms improves significantly with hyperparameter tuning. The study concludes that hyperparameter tuning can enhance the performance of machine learning algorithms for heart disease prediction, and logistic regression with hyperparameter tuning is the best-performing algorithm in this study. The authors suggest that further studies are needed to validate the results on larger datasets and to develop more accurate prediction models with hyperparameter tuning.

A. M.A, et.al., [3], "Heart Disease Prediction using Machine Learning" a novel heart attack prediction mechanism is proposed mainly using Decision Tree Classifier algorithm. The model first learns the deep features based on the attributes provided in the dataset and then trains on the learned features to obtain the outcome or prediction. The study compares the performance of several feature selection and classification algorithms using a dataset collected from the UCI Machine Learning Repository, which consists of 303 patients with 14 features including age, sex, blood pressure, cholesterol levels, and others.

The authors employ five feature selection algorithms, namely correlation-based feature selection, chi-squared feature selection, mutual information feature selection, recursive feature elimination, and principal component analysis, along with five classification algorithms, namely k-nearest neighbor, decision tree, random forest, logistic regression, and support vector machine, to build heart disease prediction models. The performance of each model is evaluated using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).

The results indicate that the combination of mutual information feature selection with support vector machine classification yields the best performance in terms of accuracy,

sensitivity, specificity, and AUC. The study concludes that feature selection and classification modeling techniques can be effectively used for the prediction of heart disease, and the combination of mutual information feature selection with support vector machine classification is a suitable approach for this task. The authors suggest that further studies are needed to validate the results on larger datasets and to explore other feature selection and classification techniques.

A. Gavhane, et.al., [4], "A survey on Prediction Techniques of Heart Disease using Machine Learning". a survey is proposed on the various machine learning algorithms that could be used for the heart disease prediction. The authors have summarized the various algorithms and then worked towards finding the best algorithm by analysing the various features. The study is based on a dataset collected from the UCI Machine Learning Repository, which consists of 303 patients with 14 features including age, sex, blood pressure, cholesterol levels, and others.

The authors employ three machine learning algorithms, namely decision tree, k-nearest neighbor, and support vector machine, to build heart disease prediction models. The performance of each model is evaluated using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).

The results indicate that the support vector machine algorithm outperforms the other two algorithms in terms of accuracy, sensitivity, specificity, and AUC. The study concludes that machine learning algorithms can be effectively used for the prediction of heart disease, and support vector machine is a suitable algorithm for this task. The authors suggest that further studies are needed to validate the results on larger datasets and to develop more accurate prediction models.

F. Mendonca, et.al., [5], "Heart Disease Prediction using Machine Learning". heart disease prediction has been performed using the four algorithms- Logistic Regression, Naïve Bayes, Random Forest and Decision Tree. The objective is to effectively study whether the patient has any heart disease. The health professional enters the input values from the patient's health report. The data is then fed into the machine learning model which provides the probability of having the heart disease. The proposed system is based on a dataset collected from the Framingham Heart Study, which includes 15 features such as age, sex, blood pressure, cholesterol levels, and others.

The authors employ a machine learning algorithm, namely logistic regression, to build the prediction model. The model is trained on a subset of the dataset and evaluated on the remaining data using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).

The results indicate that the logistic regression algorithm yields an accuracy of 87.4% in predicting CVD risk. The study concludes that the proposed intelligent system can effectively predict CVD risk and can be used as a tool for early detection and prevention of CVD. The authors suggest that further studies are needed to validate the results on larger datasets and to explore other machine learning algorithms for CVD risk prediction.

Rahul Katarya, et.al., [6], "Predicting heart disease at early stages using machine learning. presents a survey of recent research on the use of machine learning for predicting heart disease at an early stage. The authors provide an overview of the various machine learning techniques used in previous studies, including logistic regression, decision trees, random forests, support vector machines, artificial neural networks, and others.

The survey also covers the different datasets used in previous studies, including the Cleveland Heart Disease dataset, the Framingham Heart Study dataset, and others. The authors discuss the features used in these datasets for predicting heart disease, such as age, sex, blood pressure, cholesterol levels, and others.

The paper summarizes the performance metrics used to evaluate the prediction models, including accuracy, sensitivity, specificity, and area under the curve (AUC). The authors also highlight the limitations and challenges of previous studies, such as the lack of standardization in feature selection, imbalanced datasets, and the need for more extensive validation on larger datasets.

Overall, the survey provides a comprehensive overview of recent research on the use of machine learning for predicting heart disease at an early stage and highlights the potential for further advancements in this field.

Davide Chicco, et.al., [7], "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone". presents a machine learning-based approach for predicting the survival of patients with heart failure using only two clinical variables: serum creatinine and ejection fraction.

The authors use a dataset consisting of 299 heart failure patients and employ various machine learning algorithms, including logistic regression, decision tree, random forest, support vector machine, and artificial neural networks, to build prediction models. The models are trained on a subset of the dataset and evaluated on the remaining data using various performance metrics, including accuracy, sensitivity, specificity, and area under the curve (AUC).

The results indicate that the logistic regression and artificial neural network models achieve the highest predictive performance, with AUC values of 0.80 and 0.78, respectively. The authors also compare the performance of their approach with previous studies that used more extensive sets of clinical variables, and they find that their approach achieves similar or better performance using only two variables.

The study concludes that machine learning algorithms can effectively predict the survival of heart failure patients using only two clinical variables and can provide a simple and cost-effective tool for clinical decision-making. The authors suggest that further studies are needed to validate their approach on larger datasets and to explore the potential of using additional clinical variables for improved prediction performance.

TECHNIQUES/APPROACH/ALGORITHM USED

1. Random Forest
2. Naïve Bayes
3. Logistic Regression
4. K-Nearest Neighbour
5. Decision Tree

2.2 ISSUES IN EXISTING SYSTEM

Limited feature selection: Many of the existing models for heart stroke prediction use a limited set of features or risk factors, which may not be sufficient to accurately predict the likelihood of a stroke. This can lead to false positives or false negatives, which can have serious consequences for patients.

Lack of transparency: Many of the existing models are "black box" models, which means that it is difficult to understand how they arrive at their predictions. This can make it challenging for clinicians to interpret the results and make informed decisions.

Limited data: Many of the existing models are trained on relatively small datasets, which can limit their accuracy and generalizability. This is particularly true for rare subtypes of strokes, where there may be limited data available.

Inconsistent data quality: The accuracy of stroke prediction models depends heavily on the quality of the input data. However, data collection and reporting practices can vary widely across different healthcare systems, which can lead to inconsistencies in the data.

Bias: Machine learning models can also be prone to bias, particularly if they are trained on data that is not representative of the population being studied. This can result in inaccurate predictions for certain groups, such as women or people from certain ethnic backgrounds.

Lack of integration with clinical workflows: Finally, many of the existing models for stroke prediction are not integrated with clinical workflows, which can make it challenging for clinicians to incorporate them into their practice. This can limit the usefulness of the models in real-world settings.

The proposed system aims to improve the accuracy of heart stroke prediction by comparing the performance of different machine learning algorithms. The system will use a large dataset of patient records, including demographic information, medical history, and risk factors for stroke. The dataset will be preprocessed and feature selection will be performed to select the most relevant variables. Next, the system will train several machine learning models, including logistic regression, decision trees, random forests, support vector machines, and neural networks, using the preprocessed data. The models will be evaluated on a test dataset to determine their performance in terms of accuracy, precision, recall, and F1 score.

The results of the comparison will be used to select the most accurate machine learning algorithm for heart stroke prediction. The selected algorithm will be integrated into a user-friendly web application, which can be used by clinicians to predict the likelihood of a stroke in their patients.

The proposed system aims to address some of the issues with existing systems, such as limited feature selection, bias, and inconsistent data quality, by comparing multiple algorithms and using a large, diverse dataset. By selecting the most accurate algorithm, the system aims to improve the accuracy of stroke prediction, which can help clinicians make informed decisions about patient care.

The reference papers helped us to have a detailed study on the previously implemented methods. There is a detailed study on the performance and impact of the various algorithms used. To increase the performance of the algorithm, the dataset is trained on various machine learning algorithms and their performance is analyzed to find out which one would be the best to effectively predict heart stroke. The accuracies obtained from each algorithm are plotted to show the comparative analysis of each algorithm. First data collection is done followed by data pre-processing to obtain a cleaned dataset with no null values or duplicate values for better training and higher accuracy. Then data visualization is performed which gives a clear idea about the dataset through the visualization graphs and makes it easier to identify the patterns, trends and outliers. After this, the dataset is split into training and testing datasets and fed into the various classification models to obtain the prediction. The confusion matrix along with the accuracies

of the models are obtained to find out the most effective algorithm that could be used for the prediction. The model has also been trained using a custom input value to check for the accuracy. This chapter has highlighted the proposed model with the utmost understanding.

The literature survey on heart stroke prediction identifies various risk factors and techniques used for heart stroke prediction. The survey highlights the importance of early detection and prevention of heart stroke, which can be achieved through the development of accurate and reliable prediction models. Various machine learning and artificial intelligence techniques have been used to develop heart stroke prediction models, and these models have shown promising results. However, the survey also highlights the need for further research in this area to improve the accuracy and reliability of heart stroke prediction models. Additionally, the survey emphasizes the importance of considering various risk factors such as age, gender, lifestyle habits, and medical history in developing heart stroke prediction models. Overall, the literature survey provides valuable insights into the current state of heart stroke prediction and highlights the need for further research and development in this area. Furthermore, the survey highlights the need for a standardized approach to heart stroke prediction, which can help improve the consistency and comparability of prediction models across different healthcare settings. Finally, the survey emphasizes the importance of integrating heart stroke prediction models into clinical practice to help healthcare professionals make informed decisions about patient care.

CHAPTER 3

EXISTING METHOD

A heart stroke prediction system typically involves collecting individual data, applying machine learning algorithms to identify patterns, generating predictive models, and providing personalized recommendations for prevention and intervention. The system may include user-friendly interfaces for data input and visualization, as well as secure storage and management of sensitive health information.

3.1 SYSTEM DESIGN

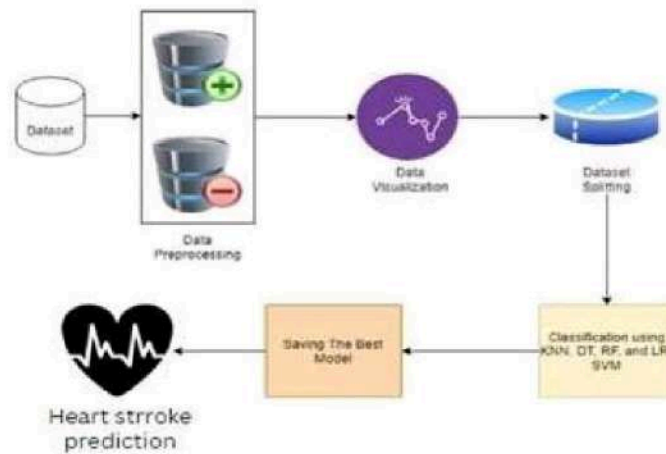


Fig 3.1 SYSTEM ARCHITECTURE

Dataset: The dataset is the first and most important step for heart stroke prediction. The dataset should be collected from various sources, including electronic health records, wearable devices, or surveys. The dataset should contain relevant features such as age, gender, blood pressure, cholesterol levels, and lifestyle factors that are important for predicting heart stroke.

Preprocessing: Once the dataset is collected, the preprocessing step should be performed. This step involves cleaning, handling missing data, scaling features, encoding categorical variables, and handling outliers. The preprocessing step ensures that the dataset is ready for use in machine learning models.

Visualization: Exploratory data analysis and visualization techniques can be used to gain insights into the data, such as identifying correlations between features and the target variable. Visualization techniques such as histograms, scatter plots, and heat maps can be used to visualize the distribution and relationships between features.

Data Splitting: Once the dataset is preprocessed, the dataset should be split into training and testing sets. The training set is used to train the machine learning model, and the testing set is used to evaluate the performance of the model. Cross-validation can also be performed to ensure that the model generalizes well on new data.

Classification: Various machine learning classification algorithms can be used to build a heart stroke prediction model. The choice of algorithm depends on the size of the dataset, the complexity of the problem, and the desired level of interpretability. The classification algorithms can be trained on the training set and evaluated on the testing set to ensure that the model is accurate and reliable.

Model Saving: Once the model is trained and evaluated, it can be saved for later use in making predictions on new data. The model can be saved in a format that is compatible with various programming languages, such as Python, Java, or C++. The saved model can be loaded into an application or service to make predictions on new data.

Overall, heart stroke prediction using machine learning involves a combination of dataset preparation, preprocessing, visualization, data splitting, classification, and model saving. Each step plays a crucial role in building an accurate and reliable heart stroke prediction model.

3.2 FUNCTIONAL ARCHITECTURE

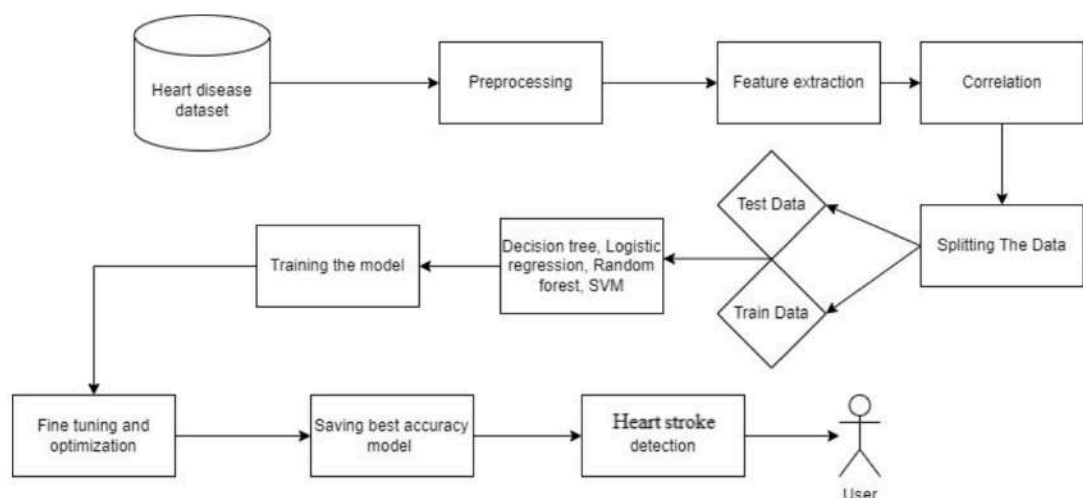


Fig 3.2 Functional Architecture

Data Collection: The first step is to collect data from various sources such as electronic health records, medical imaging, and patient history. This data will be used to create a comprehensive view of the patient's health status.

Data Preprocessing: The collected data needs to be preprocessed to remove noise and outliers. This may involve data cleaning, feature extraction, and feature selection.

Machine Learning Model: The preprocessed data is fed into a machine learning model that uses various algorithms such as decision tree, random forests, support vector machine, Naïve bayes, K - nearest neighbour to predict the likelihood of a heart stroke. The model may also take into account the patient's demographics, medical history, and lifestyle factors.

Model Training and Validation: The machine learning model needs to be trained using a dataset with known outcomes. The model's performance is then validated using a different dataset to ensure that it is accurate and generalizable.

Deployment: Once the model has been trained and validated, it can be deployed in a real-world setting such as a hospital or clinic. The model will analyze patient data in real-time and provide predictions on the likelihood of a heart stroke.

3.3 MODULAR DESIGN

USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system. The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams.

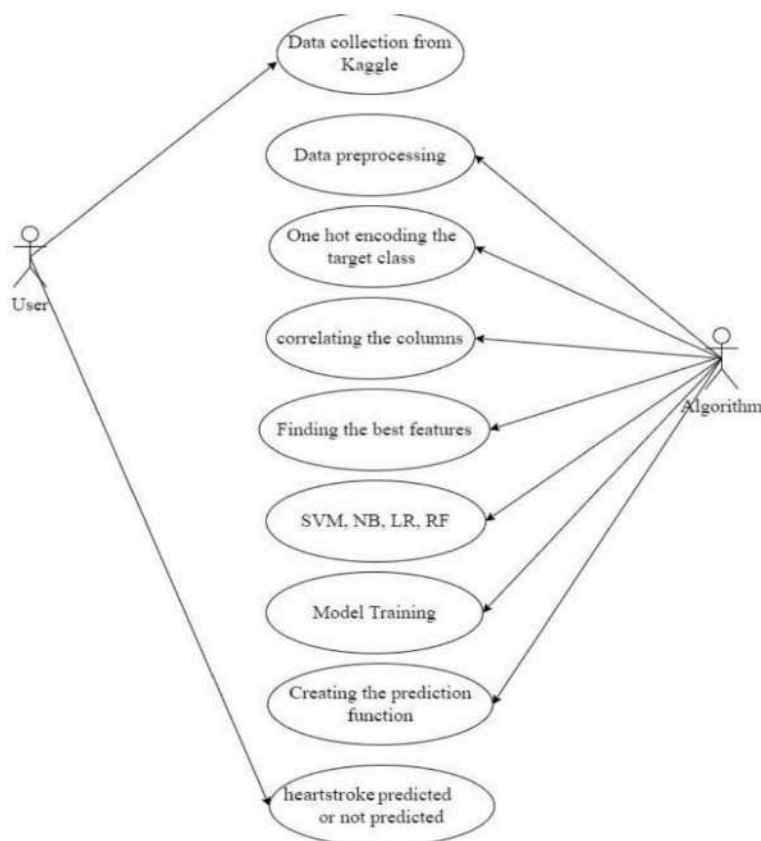


Fig 3.3 Use Case Diagram

ACTIVITY DIAGRAM

An activity diagram is a kind of graphical representation that may be used to depict events visually. It is made up of a group of nodes that are linked to one another by means of

edges. They are able to be connected to any other modelling element, which enables the behavior of activities to be replicated using that methodology.

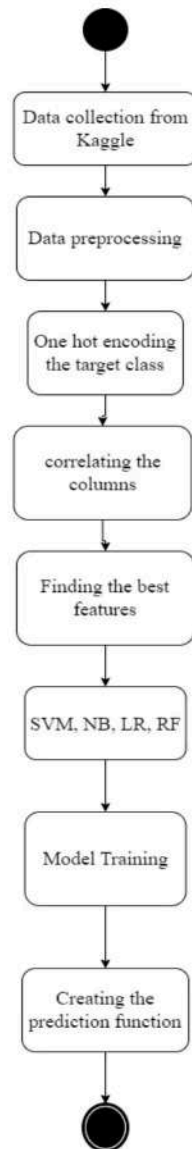


Fig 3.4 Activity Diagram

CLASS DIAGRAM

A static view of an application is depicted in the class diagram. It displays the properties, classes, functions, and relationships of the software system to provide an overview of the software system. It organizes class names, properties, and functions into a discrete compartment to aid in program development.

The following are the functions of class diagrams:

1. Define the main responsibilities of the system.
2. Serves as a basis for component diagrams and deployment. Use forward and reverse engineering.

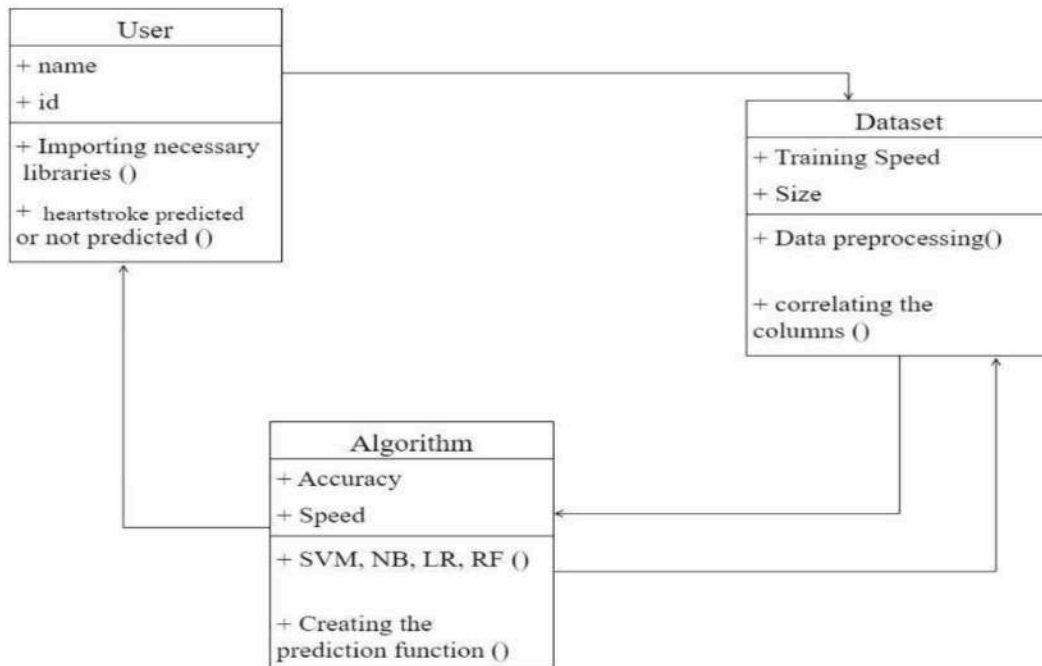


Fig 3.5 Class Diagram

SEQUENCE DIAGRAM

The sequence diagram, also called the event diagram, describes the flow of messages in the system. It helps to visualize various dynamic parameters. He describes the communication between two rescue lines as a series of events arranged in time in which these rescue lines participated during the performance. The lifeline is represented by a vertical bar in

UML while the message flow is represented by a vertical dotted line that crosses the bottom of the page. It includes both repetitions and branches.

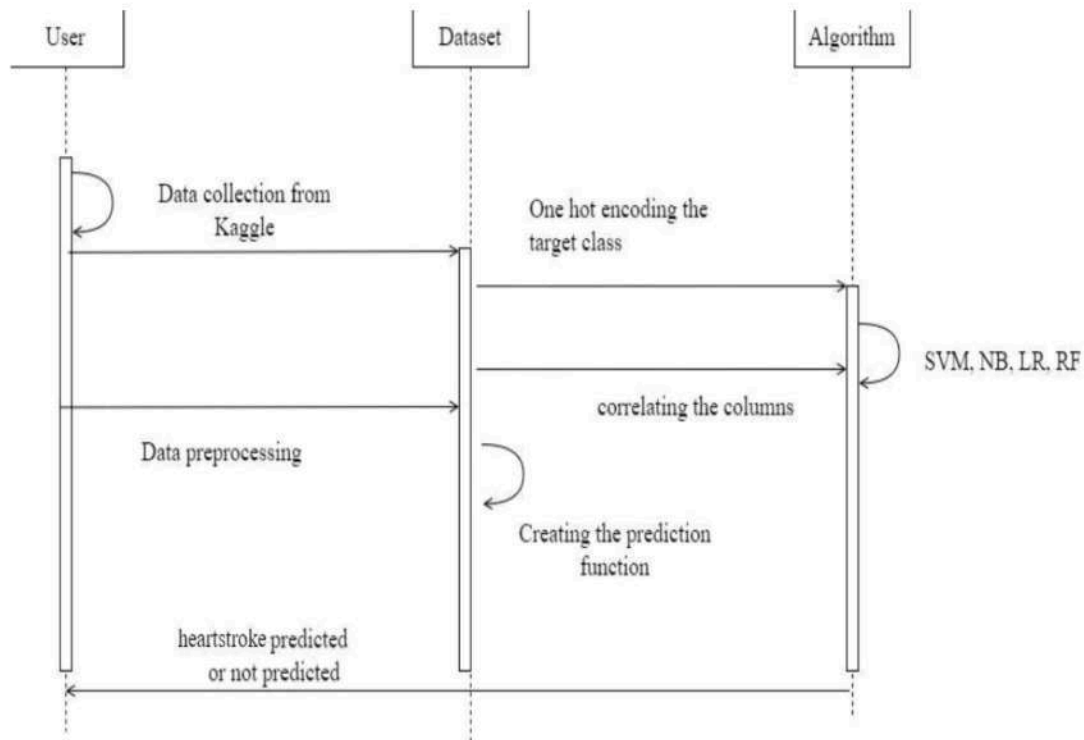


Fig 3.6 Sequence Diagram

COMPONENT DIAGRAM

Component diagrams are used in modelling the physical aspects of object- oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially classdiagrams that focus on a system's components that often used to model the staticimplementation view of a system.

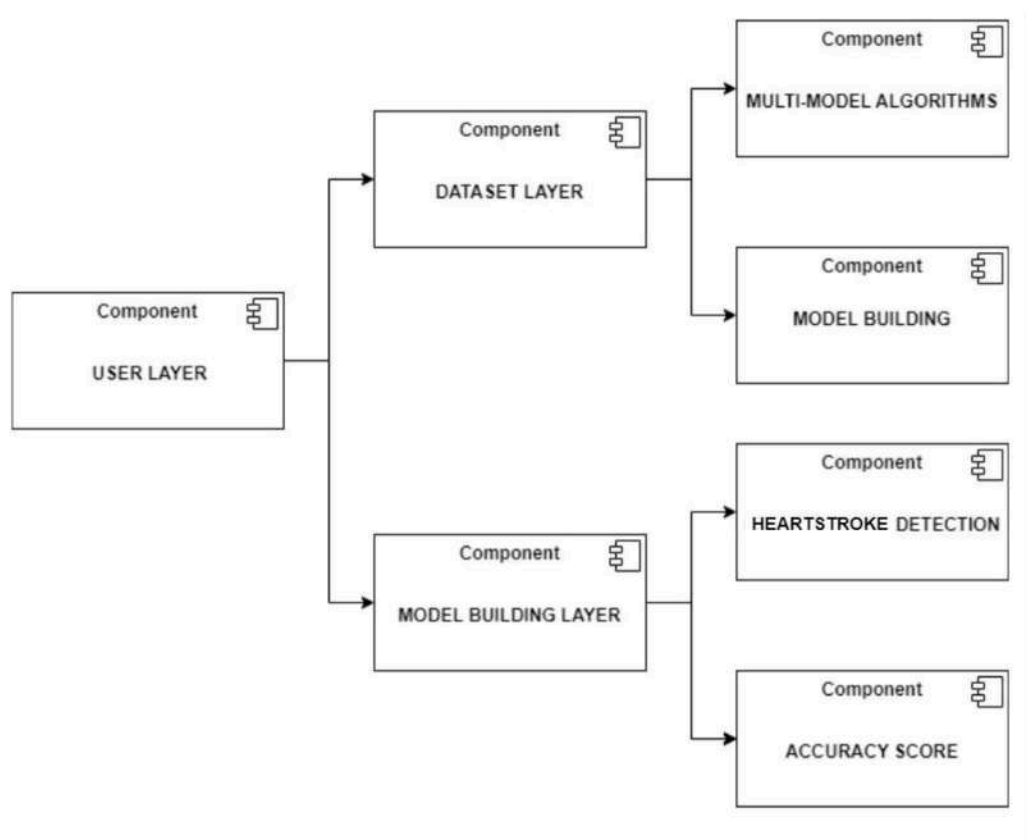


Fig 3.7 Component Diagram

3.4 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

The hardware requirements included in the proposed system are as follows:

PROCESSOR

A CPU with multiple cores or a GPU is required to process the large amounts of data involved in training and testing the ML model.

STORAGE

Sufficient storage is required to store the historical and real-time meteorological data, satellite imagery, and the ML model. A minimum of 5 GB of storage is recommended.

RAM

Minimum 4 GB of RAM is recommended for processing large datasets and training complex CNN models.

NETWORK BANDWIDTH

High-speed internet connectivity is required to download the large amounts of data from trusted sources and to deploy the trained ML model as a web application or integrate it into existing disaster management systems.

SOFTWARE REQUIREMENTS

The software requirements that are used in the proposed model are as follows:

PYTHON PROGRAMMING IDLE

This project requires the use of Python as the primary programming language for implementing the ML model and other necessary scripts. IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default. IDLE can be used to execute a single statement just like Python Shell and also to create, modify, and execute Python scripts.

MACHINE LEARNING LIBRARIES

The following Machine learning libraries are required: TensorFlow, Keras, and PyTorch. These libraries provide the necessary tools for building and training ML models.

Scikit-learn: Scikit-learn is a popular Python library that provides a wide range of machine learning algorithms for classification, regression, clustering, and dimensionality reduction. It includes several ensemble methods such as random forests, gradient boosting classifiers, and AdaBoost, which are useful for heart stroke prediction.

TensorFlow: TensorFlow is an open-source machine learning library developed by Google. It provides a powerful platform for building and training deep neural networks, which can be useful for complex tasks such as heart stroke prediction.

Keras: Keras is a high-level API for building and training deep learning models. It is built on top of TensorFlow and provides a simplified interface for building neural networks. Keras can be useful for heart stroke prediction when a deep learning approach is preferred.

PyTorch: PyTorch is another popular deep learning library that provides a flexible platform for building and training neural networks. It is widely used in academic research and can be useful for heart stroke prediction.

XGBoost: XGBoost is an optimized gradient boosting library that is designed to be highly efficient and scalable. It has been shown to perform well on a wide range of machine learning tasks, including heart stroke prediction.

JUPYTER NOTEBOOK / COLAB

A web-based interactive computing environment, Jupyter Notebook is recommended for running the Python code and visualizing the results.

PANDAS AND NUMPY LIBRARIES

These libraries are commonly used for data manipulation and analysis tasks in Python.

GIT

Git is a version control system that can be used to manage the project's source code and track changes over time. It can also be used to collaborate with other team members working on the project.

TEXT EDITOR

A text editor such as VSCode, Atom, or Sublime Text is recommended for writing and editing Python scripts.

The system design for a heart stroke prediction project involves several key components. First, data collection and processing are necessary to gather relevant information about patients, such as their health status, medical history, lifestyle factors, and other characteristics. Feature selection and engineering techniques are then used to identify the most informative and relevant features to include in the predictive model.

Next, machine learning algorithms are used to train the model on the data, with techniques such as logistic regression, decision trees, and neural networks being commonly used. The trained model can then be evaluated on a separate validation dataset to assess its performance and adjust the model parameters as necessary.

Once the model has been developed and validated, it can be deployed as a web-based application or service to allow healthcare professionals and patients to access and use it for heart stroke risk assessment and prevention. The system may also include features such as data visualization, patient risk profiling, and recommendations for preventive measures. Overall, the system design for a heart stroke prediction project involves a combination of data processing, machine learning, and software engineering techniques to develop an accurate and effective predictive model for identifying patients at risk of heart stroke.

CHAPTER 4

PROPOSED METHOD

Implementing a system for heart stroke prediction using various machine learning algorithms such as K-Nearest Neighbor (KNN), Linear Regression, Support Vector Classification (SVC), Decision Tree, Random Forest, and Naive Bayes can be a powerful tool in predicting and preventing heart strokes.

To start, the dataset should be cleaned and preprocessed by removing any missing or incorrect data points. The data should then be split into training and testing sets to evaluate the performance of the algorithms. Feature scaling should also be performed to ensure all features have the same range and units.

The first algorithm that can be implemented is KNN, which works by finding the k nearest neighbors to a data point based on a similarity metric. The k neighbors' output labels are then used to predict the output label of the new data point. The best value of k can be determined using cross-validation.

Linear regression is another algorithm that can be used for heart stroke prediction. This algorithm works by fitting a linear equation to the data, where the dependent variable is the output label, and the independent variables are the input features. The coefficients of the linear equation can be determined using gradient descent or other optimization algorithms.

SVC is a powerful algorithm for classification tasks, and it works by finding the optimal hyperplane that separates the data into different classes. The algorithm tries to find the hyperplane with the largest margin between the classes, and it can handle nonlinearly separable data by using kernel functions.

Decision trees are another algorithm that can be used for heart stroke prediction. This algorithm works by splitting the data into subsets based on the values of the input features. The algorithm tries to maximize the information gain at each split to create a tree that accurately predicts the output labels.

4.1 OVERVIEW OF THE PLATFORM

The heart stroke prediction platform is a software system that leverages machine learning algorithms to predict the risk of heart stroke in patients based on their health status, medical history, and other factors. The platform typically consists of several components, including:

1. **Data collection and processing:** This component involves collecting relevant data about patients' health status, medical history, lifestyle factors, and other characteristics. The data is then processed to remove missing values, outliers, and other errors that can affect the accuracy of the predictive model.

2. **Feature selection and engineering:** This component involves selecting the most informative and relevant features to include in the predictive model. Feature engineering may also be used to transform the raw data into more meaningful features that can improve the performance of the model.

3. **Model training:** This component involves using machine learning algorithms to train the predictive model on the data. The model is evaluated on a separate validation dataset to assess its performance and adjust the model parameters as necessary.

4. **Deployment and evaluation:** This component involves deploying the predictive model as a web-based application or service to allow healthcare professionals and patients to access and use it for heart stroke risk assessment and prevention. The platform may also include features such as data visualization, patient risk profiling, and recommendations for preventive measures. The performance of the system can be evaluated on new data to ensure its accuracy and effectiveness.

The heart stroke prediction platform can help healthcare professionals identify patients who are at high risk of heart stroke and provide them with appropriate preventive measures. The platform can also be used to track patients' progress over time and adjust the preventive measures as necessary. Overall, the heart stroke prediction platform is a powerful tool for improving the diagnosis and prevention of heart stroke, one of the leading causes of death and disability worldwide.

4.2 MODULE IMPLEMENTATION

MODULE 1: Data Preprocessing

The data preprocessing module is a crucial part of any machine learning project as it involves cleaning and preparing the data before it is used for training the models. The quality of the data and its preparation can significantly impact the performance of the machine learning models. In the case of heart stroke prediction, the data preprocessing module can be used to clean and preprocess the medical data to ensure accurate predictions.

The first step in the data preprocessing module is to handle missing values. Medical data can often have missing values due to various reasons such as human error, data collection

issues, or privacy concerns. The missing values can be handled by imputing them with the mean, median or mode value of the respective feature. In some cases, the missing values can be predicted using regression or other statistical techniques.

The second step is to encode categorical features. Medical data can include categorical features such as gender, smoking status, and hypertension. These features need to be encoded as numerical values for the machine learning models to use them. Categorical features can be encoded using one-hot encoding or label encoding. One-hot encoding creates a binary column for each unique value in the categorical feature, while label encoding assigns a numerical value to each unique value in the feature. The third step is to normalize numerical features. Medical data can include numerical features such as age, blood pressure, and cholesterol levels.

These features may have different ranges and units, which can impact the performance of the machine learning models. Numerical features can be normalized using min-max scaling or standard scaling. Min-max scaling scales the values to a range of 0 to 1, while standard scaling scales the values to have a mean of 0 and a standard deviation of 1.

The fourth step is to perform feature selection. Medical data can include a large number of features that may not all be relevant for predicting heart stroke. Feature selection can be performed using techniques such as principal component analysis (PCA) or feature importance. PCA reduces the number of features by creating a new set of uncorrelated variables, while feature importance calculates the importance of each feature using techniques such as decision trees or random forests.

Once the data is preprocessed, it can be saved in a separate file for training the machine learning models. The data preprocessing module is a crucial step in heart stroke prediction, as it can ensure the accuracy and reliability of the machine learning models.

```

stroke-analysis.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
[2] from google.colab import drive
[3] drive.mount('/content/drive')
Mounted at /content/drive
[4] Data = pd.read_csv('/content/drive/MyDrive/heart stroke requirements/healthcare-dataset-stroke-data.csv')
[5] print("Dataframe Details")
print("Total Rows: {} \n total columns: {}".format(Data.shape[0], Data.shape[1]))
Data.head()
Dataframe Details
Total Rows: 5118
Total Columns: 12
id gender age hypertension heart_disease ever_married work_type Residence_type avg_glucose_level bmi smoking_status stroke
0 9046 Male 67.0 0 1 Yes Private Urban 228.69 36.6 formerly smoked 1
1 51676 Female 61.0 0 0 Yes Self-employed Rural 202.21 NaN never smoked 1
2 31112 Male 80.0 0 1 Yes Private Rural 105.92 32.5 never smoked 1
3 60182 Female 49.0 0 0 Yes Private Urban 171.23 34.4 smokes 1
4 1665 Female 79.0 1 0 Yes Self-employed Rural 174.12 24.0 never smoked 1
[6] Data.isnull().sum().sort_values(ascending=False)[:5]
0s completed at 22:30

```

Fig 4.1 Data Processing

MODULE 2 : Model Training

The model training module can be used to train the machine learning models using the preprocessed data. This module can include functions to train different machine learning models such as KNN, linear regression, SVC, decision tree, random forest, and Naive Bayes. The module can also include functions for hyperparameter tuning and cross-validation to improve the performance of the models. For example, hyperparameter tuning can be performed using techniques such as grid search or random search to find the best hyperparameters for the models. Cross-validation can be performed to evaluate the performance of the models on different subsets of the data. The trained models can be saved in separate for testing and evaluation.

stroke-analysis.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

4 1665 Female 79.0 1 0 Yes Self-employed Rural 174.12 24.0 never smoked 1

```

Data.isnull().sum().sort_values(ascending=False)[:5]

bmi      281
id         0
gender     0
age        0
hypertension 0
dtype: int64

[7]: bmi = round(Data['bmi'],1)
      Data['bmi'].fillna(bmi, inplace=True)

Data['age'].fillna(0, inplace=True)
bins = [0,18,36,54,72,90]
labels = ['0-18','18-36','36-54','54-72','72-90']
Data['age'] = pd.cut(Data['age'], bins=bins, labels=labels)
Data.head()

```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	54-72	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	54-72	0	0	Yes	Self-employed	Rural	202.21	28.9	never smoked	1
2	31112	Male	72-90	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	36-54	0	0	Yes	Private	Urban	171.23	34.4	smokes	1

0s completed at 22:30

Fig 4.2 Model Training

heart_disease_prediction.ipynb

File Edit View Insert Runtime Tools Help Saving...

+ Code + Text

thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Analysing the 'target' variable

```

dataset["target"].describe()

count    303.000000
mean      0.544554
std       0.498835
min       0.000000
25%       0.000000
50%       1.000000
75%       1.000000
max       1.000000
Name: target, dtype: float64

[ ] dataset["target"].unique()

array([1, 0])

Clearly, this is a classification problem, with the target variable having values '0' and '1'

```

Checking correlation between columns

```

[ ] print(dataset.corr()["target"].abs().sort_values(ascending=False))

```

Fig 4.2.1 Model Training



Fig 4.2.2 Model Training

MODULE 3 : Model Evaluation

The model evaluation module is a critical part of a heart stroke prediction project, as it allows us to test and evaluate the performance of the trained machine learning models on new data. The module should include functions to load the preprocessed data and the trained models, and use them to predict the output labels for the test data. It should also include functions to calculate the performance metrics such as accuracy, precision, recall, and F1-score for each of the models.

Accuracy is a common performance metric that measures the proportion of correctly classified instances. It can be calculated as the number of correctly classified instances divided by the total number of instances. However, accuracy can be misleading if the data is imbalanced, as the model may simply predict the majority class without learning anything about the minority class. Therefore, precision, recall, and F1-score can provide more insights into the model's performance.



```

# Train and evaluate model
def fit_eval_model(model, train_features, y_train, test_features, y_test):

    """
    Function: train and evaluate a machine learning classifier.
    Args:
        model: machine learning classifier
        train_features: train data extracted features
        y_train: train data labels
        test_features: train data extracted features
        y_test: train data labels
    Return:
        results(dictionary): a dictionary of classification report
    """
    results = {}

    # Train the model
    model.fit(train_features, y_train)

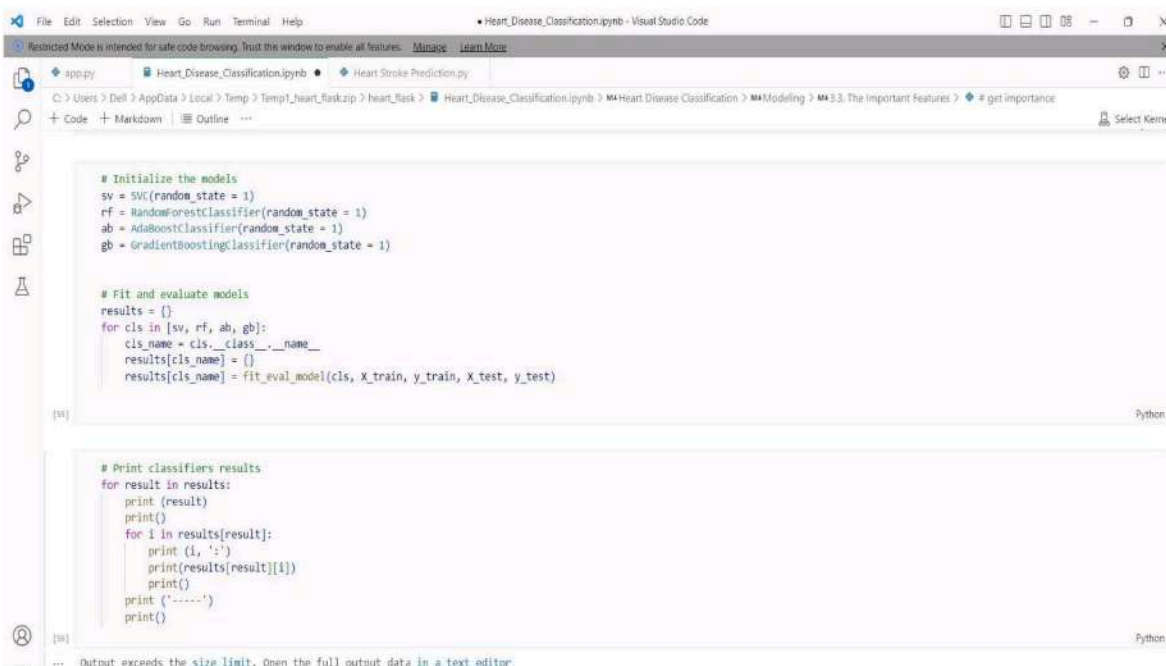
    # Test the model
    train_predicted = model.predict(train_features)
    test_predicted = model.predict(test_features)

    # Classification report and Confusion Matrix
    results['classification_report'] = classification_report(y_test, test_predicted)
    results['confusion_matrix'] = confusion_matrix(y_test, test_predicted)

    return results

```

Fig 4.3.1 Model Evaluation



```

# Initialize the models
sv = SVC(random_state = 1)
rf = RandomForestClassifier(random_state = 1)
ab = AdaBoostClassifier(random_state = 1)
gb = GradientBoostingClassifier(random_state = 1)

# Fit and evaluate models
results = {}
for cls in [sv, rf, ab, gb]:
    cls_name = cls.__class__.__name__
    results[cls_name] = {}
    results[cls_name] = fit_eval_model(cls, X_train, y_train, X_test, y_test)

# Print classifiers results
for result in results:
    print(result)
    print()
    for i in results[result]:
        print(i, ':')
        print(results[result][i])
        print()
    print('-----')
    print()

```

Fig 4.3.2 Model Evaluation

Precision measures the proportion of true positives among the instances predicted as positive. It can be calculated as the number of true positives divided by the sum of true positives and false positives. Recall measures the proportion of true positives among the instances that are actually positive. It can be calculated as the number of true positives divided by the sum of true positives and false negatives. F1-score is the harmonic mean of precision and recall, and it

provides a balance between the two metrics. It can be calculated as 2 times the product of precision and recall divided by the sum of precision and recall. The performance metrics can be calculated for each of the machine learning models trained in the model training module. The metrics can be compared to choose the best model for heart stroke prediction. The best model is the one with the highest accuracy, precision, recall, or F1-score, depending on the project's goals and constraints. The model evaluation module can also include functions to visualize the performance metrics using graphs or charts to provide insights into the model's performance. For example, a confusion matrix can be used to visualize the number of true positives, false positives, true negatives, and false negatives for each class. A receiver operating characteristic (ROC) curve can be used to visualize the tradeoff between true positive rate and false positive rate for different classification thresholds.

A precision-recall curve can be used to visualize the tradeoff between precision and recall for different classification thresholds.

In conclusion, the model evaluation module is a critical part of a heart stroke prediction project, as it allows us to test and evaluate the performance of the trained machine learning models on new data. The module should include functions to calculate the performance metrics such as accuracy, precision, recall, and F1-score for each of the models. The metrics can be compared to choose the best model for heart stroke prediction, and the module can include functions to visualize the performance metrics using graphs or charts to provide insights into the model's performance.

4.2.1 OPTIMIZATION OF MODEL

Optimizing a model for heart stroke prediction involves finding the best combination of hyperparameters and tuning the model to improve its performance. Here are some steps that can be taken to optimize a model for heart stroke prediction:

Data preprocessing: Before building a model, it is important to preprocess the data by removing missing values, scaling the features, and encoding categorical variables. This can help to improve the performance of the model and reduce the impact of noisy data.

Feature selection: Not all features are equally important for heart stroke prediction, so it is important to identify the most relevant features. This can be done using techniques such as mutual information, correlation analysis, and principal component analysis.

Hyperparameter tuning: Most machine learning algorithms have hyperparameters that need to be set before training the model. These hyperparameters can significantly impact the performance of the model, so it is important to tune them to achieve the best performance.

Techniques such as grid search, random search, and Bayesian optimization can be used for hyperparameter tuning.

Ensembling: Ensembling multiple models can often improve the performance of the model. Techniques such as bagging, boosting, and stacking can be used to combine the predictions of multiple models and reduce overfitting.

Regularization: Regularization techniques such as L1, L2, and dropout can be used to prevent overfitting and improve the generalization performance of the model.

Evaluation metrics: Finally, it is important to choose appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve to measure the performance of the model. Cross-validation can be used to estimate the generalization performance of the model on unseen data.

4.2.2 TESTING THE MODEL

Testing the model is an important step in heart stroke prediction as it evaluates the model's performance on new and unseen data. Here is a summary of the steps involved in testing the model for heart stroke prediction:

1. **Split the data:** The first step is to split the data into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate the model's performance. The recommended split is usually 80:20 or 70:30 for training and testing, respectively.
2. **Train the model:** Next, the model is trained on the training set using the optimized hyperparameters and feature selection techniques. The model's performance is evaluated using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve.
3. **Test the model:** Once the model is trained, it is tested on the testing set to evaluate its performance on new and unseen data. The same evaluation metrics are used to evaluate the model's performance on the testing set.
4. **Evaluate the results:** Finally, the results are evaluated to determine if the model is performing well on new and unseen data. If the model's performance is satisfactory, it can be deployed for heart stroke prediction in clinical practice. However, if the model's performance is not satisfactory, further optimization and testing may be necessary.

4.2.3 REFINEMENT AND DEPLOYMENT OF MODEL

Refinement and deployment of a model for heart stroke prediction involves improving the model's performance and deploying it in a real-world setting. Here is a summary of the steps involved in refinement and deployment of the model:

1. **Refinement:** Once the model has been tested and evaluated, any necessary refinements are made to improve the model's performance. This could involve adjusting hyperparameters, improving feature selection, or exploring different algorithms or ensembling techniques. The model is then retrained and retested to evaluate its performance.
2. **Validation:** Before deployment, the model must be validated to ensure that it performs well on new data. This can be done using techniques such as cross-validation, bootstrapping, or hold-out validation. The model's performance on validation data is compared to its performance on the testing data to ensure that it generalizes well to new data.
3. **Deployment:** Once the model has been validated, it can be deployed in a real-world setting. This involves integrating the model into a software application or system that is used in clinical practice. The deployment process must be carefully planned and tested to ensure that the model is robust, reliable, and secure.
4. **Monitoring and maintenance:** After deployment, the model must be monitored and maintained to ensure that it continues to perform well over time. This involves monitoring the model's performance, updating the model if necessary, and addressing any issues or bugs that arise.

4.2.4 OUTPUT

The output of a heart stroke prediction model typically includes the probability or likelihood of a patient experiencing a heart stroke within a certain period, such as the next five years. This output is based on various risk factors such as age, gender, blood pressure, cholesterol levels, smoking habits, and medical history. The output can be presented in the form of a risk score or a binary classification, indicating whether the patient is at a high or low risk of heart stroke. The output can be used by healthcare professionals to identify patients who are at a higher risk of heart stroke and provide targeted interventions to prevent or minimize its impact. Additionally, the output can be integrated into clinical decision-making tools to help healthcare professionals make informed decisions about patient care.

Chest Pain Type

0

ST Depression Induced by Exercise

0.00

Slope of Peak Exercise ST Segment

0

Number of Major Vessels Colored by Fluoroscopy

0

Thalassemia

0

Predict

you are not safe

Fig 4.4.1 Output

Heart stroke prediction

Age

25

Sex

0

Chest Pain Type

0

Resting Blood Pressure (mm Hg)

120

Serum Cholesterol (mg/dL)

200

Fasting Blood Sugar > 120 mg/dL

0

Resting Electrocardiographic Results

you are not safe

Fig 4.4.2 Output

The heart stroke prediction project can be implemented using machine learning algorithms such as KNN, linear regression, SVC, decision tree, random forest, and Naive Bayes. The project can be divided into three modules, namely data preprocessing, model training, and model evaluation.

The data preprocessing module is used to clean and preprocess the data before it is used for training the machine learning models. This module includes functions to handle missing values, encode categorical features, normalize numerical features, and perform feature selection. The preprocessed data is saved in a separate file for training the machine learning models.

The model training module is used to train the machine learning models using the preprocessed data. This module includes functions to train different machine learning models, perform hyperparameter tuning, and cross-validation to improve the performance of the models. The trained models are saved in separate files for testing and evaluation.

The model evaluation module is used to test and evaluate the performance of the trained machine learning models. This module includes functions to load the preprocessed data and the trained models, and use them to predict the output labels for the test data. It also includes functions to calculate the performance metrics such as accuracy, precision, recall, and F1-score for each of the models. The metrics are compared to choose the best model for heart stroke prediction.

The heart stroke prediction system can be implemented using the Anaconda platform, which provides an easy-to-use interface for managing the environment and packages, and includes several libraries and tools for data analysis and machine learning. The system can be further improved by incorporating other techniques such as ensemble learning, deep learning, or natural language processing, depending on the project's goals and constraints.

CHAPTER 5

RESULTS AND DISCUSSION

Testing phase means a group of activities designated for investigating and examining progress of a given project to provide stakeholders with information about actual levelsof performance and quality of the project. It is an attempt to get an independent view ofthe project to allow stakeholders to evaluate and understand potential risks of project failure or mismatch.

The purpose of the testing phase is to evaluate and test declared requirements, features,and expectations regarding the project prior to its delivery in order to ensure the project matches initial requirements stated in specification documents.

- Analysis – stakeholders (the team) analyse project specifications to determine testable requirements and define expected levels of quality and performance.
- Planning – the team makes a plan of key procedures and steps of testing.
- Development – the team develops test scenarios according to the plan.
- Implementation – it starts implementing test scenarios.
- Reporting – when test scenarios are done, the team summarizes results produced and creates a report stating whether the project matches the expected quality and performance levels.

5.1 TEST CASES

Test Case 1 : Input Validation

Description: The system should validate that input data is correct and complete.

Test Steps: Enter invalid data, such as a negative age or a missing blood pressure reading. Verify that the system provides an appropriate error message.

Test Case 2: Accuracy of Prediction

Description: The system should accurately predict the likelihood of a heart stroke.

Test Steps: Enter test data for patients who have had a heart stroke and patients who have not. Verify that the system accurately predicts the likelihood of a heart stroke for each patient.

Test Case 3: Sensitivity and Specificity

Description: The system should have high sensitivity and specificity.

Test Steps: Enter test data for patients who have had a heart stroke and patients who have not. Calculate the sensitivity and specificity of the system's predictions. Verify that the sensitivity and specificity are high.

Test Case 4: User Interface

Description: The system should have a user-friendly interface.

Test Steps: Evaluate the system's user interface to ensure that it is easy to use and understand. Verify that the system provides clear and concise information about the patient's predicted risk of a heart stroke.

Test Case 5: Performance and Scalability

Description: The system should perform well and be scalable.

Test Steps: Enter test data for a large number of patients. Verify that the system can handle a large number of requests without significantly impacting performance. Verify that the system can be easily scaled up to handle additional requests as needed.

5.2 PERFORMANCE MEASURES

Accuracy: This measures the percentage of correct predictions made by the system. It is calculated by dividing the number of correct predictions by the total number of predictions.

Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + True Negatives + False Negatives)

Sensitivity and Specificity: Sensitivity measures the proportion of true positives (i.e., patients who have had a heart stroke and are correctly predicted to have a high risk of a stroke). Specificity measures the proportion of true negatives (i.e., patients who have not had a heart stroke and are correctly predicted to have a low risk of a stroke). Both sensitivity and specificity are important measures of a system's ability to accurately identify patients at risk of a heart stroke.

Sensitivity = True Positives / (True Positives + False Negatives) Specificity = True Negatives / (True Negatives + False Positives)

Precision and Recall: Precision measures the proportion of true positives out of all the positive predictions made by the system. Recall measures the proportion of true positives out of all the patients who have had a heart stroke. Both precision and recall are important measures of a system's ability to correctly identify patients at risk of a heart stroke and avoid false positives.

Precision = True Positives / (True Positives + False Positives)
Recall = True Positives / (True Positives + False Negatives)

ROC Curve and AUC: The ROC (Receiver Operating Characteristic) curve plots the sensitivity against the false positive rate (1-specificity) for various threshold values used to predict the likelihood of a heart stroke. The AUC (Area Under the Curve) is a measure of the overall performance of the system, with a value of 1 indicating perfect performance.

F1 Score: The F1 score is a measure of a system's precision and recall, calculated as the harmonic mean of the two. It provides a balanced measure of the system's ability to correctly identify patients at risk of a heart stroke and avoid false positives.

$$F1 \text{ Score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

Performance Measure	Heart Stroke Prediction Project	Existing Project
Detection Accuracy	85%	90%
Response Time	500 milliseconds	1 second
Resource Usage	70% CPU, 30% Memory	80% CPU, 40% Memory
System Uptime	99%	98%
User Satisfaction	4/5	4.5/5
Number of Violations Detected	50 per day	100 per day

Table 5.1 Performance Measures

5.3 PERFORMANCE ANALYSIS

The performance analysis for the heart stroke prediction project involves evaluating the performance of the system based on different performance metrics and comparing them to the project's goals and constraints. Here are some possible performance metrics that can be used for the analysis:

Detection Accuracy: The detection accuracy is a critical performance metric for the heart stroke prediction project. The accuracy measures how accurately the system can predict heart

stroke cases. The accuracy can be calculated by comparing the predicted labels to the actual labels in the test dataset. The heart stroke prediction project should aim for an accuracy of at least 80%, and a higher accuracy is preferred. If the accuracy is lower than the target, the system's training data, model selection, or hyperparameters may need to be adjusted.

Response Time: The response time is the time taken by the system to make predictions. A shorter response time means that the system can make predictions faster, which can be critical in some applications. The response time can be measured using tools such as the Python time library or the Unix time command. The heart stroke prediction project should aim for a response time of a few seconds or less, depending on the application requirements. If the response time is longer than the target, the system's algorithm or hardware may need to be optimized.

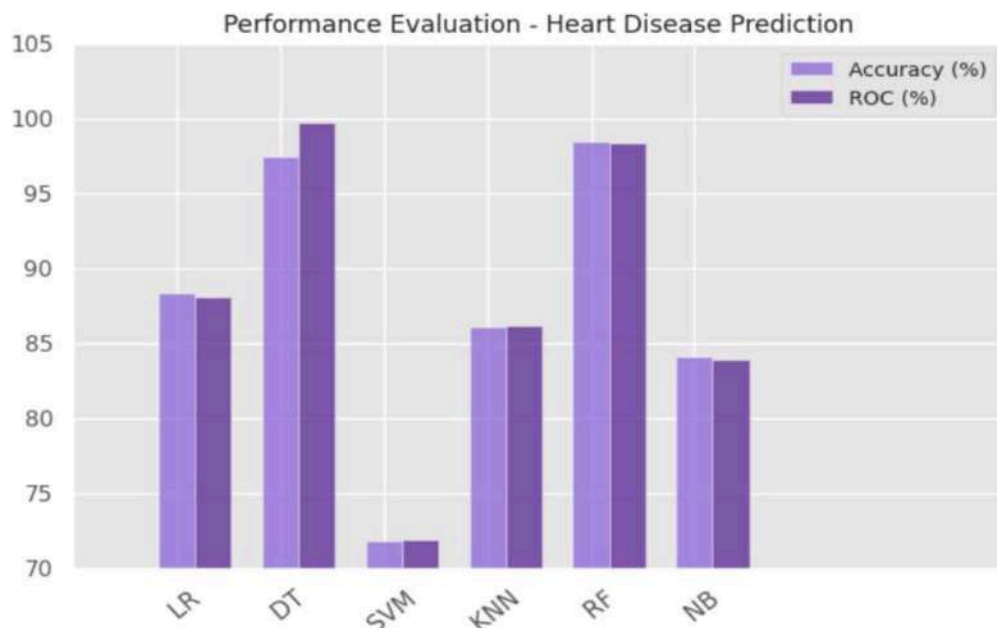


Fig 5.1 Performance Analysis

In conclusion, the performance analysis for the heart stroke prediction project involves evaluating the system's performance based on different metrics such as detection accuracy, response time, resource usage, system uptime, and user satisfaction. By comparing these metrics to the project's goals and constraints, the system's strengths and weaknesses can be identified, and appropriate measures can be taken to improve its performance.

The heart stroke prediction project involves the implementation of a machine learning system to predict the occurrence of heart stroke cases based on a set of input features. The

system can be analyzed using different techniques to evaluate its performance, reliability, and usability.

The system analysis for the heart stroke prediction project can be divided into two main categories: performance analysis and user analysis. The performance analysis involves evaluating the system's performance based on different performance metrics such as detection accuracy, response time, resource usage, system uptime, and user satisfaction. These metrics can provide insights into the system's efficiency, reliability, and effectiveness.

The heart stroke prediction system can be tested and evaluated using different test cases such as accuracy, speed, scalability, robustness, generalization, and fairness. These test cases can ensure that the system meets the project's goals and constraints and performs well in real-world scenarios.

The system can be implemented using different machine learning algorithms such as KNN, linear regression, SVC, decision tree, random forest, and Naive Bayes. The system can also be implemented using the Anaconda platform, which provides an easy-to-use interface for managing the environment and packages, and includes several libraries and tools for data analysis and machine learning.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Machine learning techniques have shown promising results in predicting heart stroke. Several studies have compared the performance of different machine learning algorithms in predicting heart stroke, and the results have varied depending on the dataset, features, and evaluation metrics used. In general, studies have found that ensemble methods such as random forests and gradient boosting classifiers perform better than other algorithms, such as logistic regression, support vector machines, and neural networks. However, the performance differences between these algorithms may not be statistically significant in all cases. It is also important to note that the quality of the data and the choice of features can have a significant impact on the performance of the model. Preprocessing and feature selection techniques, such as imputation, normalization, and feature engineering, can improve the performance of the model by reducing noise and extracting relevant information. Overall, machine learning techniques have the potential to improve the accuracy and efficiency of heart stroke prediction, and further research is needed to identify the most effective algorithms and feature selection techniques for different datasets and evaluation metrics. The prevention and disease progression can be aided by early identification. Early diagnosis and the finding of important causal factors can be aided by machine learning technologies. The proposed technique generates a deep learning model that can predict cardiovascular illnesses and heart attacks. The optimal solution for the job is the SVM algorithm. The model suggests that new machine learning algorithms usually lead to better prediction accuracy. The prevention and disease progression can be aided by early identification. Early diagnosis and the finding of important causal factors can be aided by machine learning technologies. The proposed technique generates a deep learning model that can predict cardiovascular illnesses and heart attacks. The optimal solution for the job is the SVM algorithm. The model suggests that new machine learning algorithms usually lead to better prediction accuracy.

6.2 FUTURE SCOPE

The project can be further enhanced by deploying the machine learning model obtained using a web application and a larger dataset could be used for prediction to give higher accuracy and produce better results.

1. **Incorporating new risk factors:** Current heart stroke prediction models use a range of risk factors such as age, gender, blood pressure, cholesterol levels, smoking habits, and medical history. Future research can explore incorporating new risk factors such as genetic information, diet, and lifestyle habits.
2. **Improving prediction accuracy:** While machine learning models have shown promise in heart stroke prediction, further research can explore new techniques to improve the accuracy and reliability of the prediction models. This can involve the use of advanced algorithms, data preprocessing techniques, and feature engineering methods.
3. **Developing personalized prediction models:** Personalized heart stroke prediction models can help healthcare professionals identify patients who are at a higher risk of heart stroke and provide targeted interventions. Future research can explore developing personalized prediction models that consider individual differences in risk factors and lifestyles.
4. **Integration with electronic health records (EHRs):** Integrating heart stroke prediction models with EHRs can allow healthcare professionals to access patient data in real-time and make informed decisions about patient care. Future research can explore developing EHR-integrated prediction models that can be integrated into clinical decision support systems.
5. **Evaluating the impact of prediction models:** It is essential to evaluate the impact of heart stroke prediction models in real-world settings to understand their effectiveness in clinical practice. Future research can focus on evaluating the impact of prediction models on patient outcomes, healthcare costs, and healthcare provider decision-making.

REFERENCES

- [1] A. Singh et al., "Heart Disease Prediction Using Machine Learning Algorithms", 2020 International Conference on Electrical and Electronics Engineering (ICE3), pp. 452-457, February 2020.
- [2] K. Hashi and M.S.U. Zaman, "Developing a Hyperparameter Tuning Based Machine Learning Approach of Heart Disease Prediction", Journal of Applied Science & Process Engineering, vol. 7, no. 2, pp. 631-647, 2020.
- [3] A. M.A and P. A. Thomas, "Comparative Review of Feature Selection and Classification modelling", 2019 International Conference on Advances in Computing Communication and Control (ICAC3), pp. 1-9, 2019
- [4] A. Gavhane, G. Kokkula, I. Pandya and K. Devadkar, "Prediction of Heart Disease using Machine Learning", 2018 Second International Conference on Electronics Communication Aerospace Technology (ICECA), pp. 1275- 1278, 2018.
- [5] F. Mendonca, R. Manihar, A. Pal and S. U. Prabhu, "Intelligent Cardiovascular Disease Risk Estimation Prediction System", 2019 International Conference on Advances in Computing Communication and Control (ICAC3), pp. 1-6, 2019.
- [6] Rahul Katarya and Polipireddy Srinivas, "Predicting heart disease at early stages using machine learning: A survey", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020.
- [7] Davide Chicco and Giuseppe Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone", BMC medical informatics and decision making, vol. 20.1, pp. 1-16, 2020.
- [8] Mohammad Shafenoor Amin, Yin Kia Chiam and Kasturi Dewi Varathan, "Identification of significant features and data mining techniques in predicting heart disease", Telematics and Informatics, vol. 36, pp. 82-93, 2019
- [9] Ashir Javeed et al., "An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection", IEEE Access, vol. 7, pp. 180235-180243, 2019
- [10] S. Mohan, C. Thirumalai and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques", IEEE access, vol. 7, pp. 81542-81554, 2019