A project report on

BRAIN STROKE PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

GANDRATH ARYAN (20BCE1736)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April,2024

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled "BRAIN STROKE PREDICTION USING MACHINE LEARNING" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering, Vellore Institute of Technology, Chennai is a record of bonafide work carried out by me under the supervision of Dr. Rajesh R

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai
Date: Signature of the Candidate

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"Brain stroke prediction using Machine Learning"** is prepared and submitted by **Gandrath Aryan (20BCE1736)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering programme** is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. Rajesh R

Date:

Signature of the Internal Examiner

Name:

Date:

Signature of the External Examiner

Name:

Date:

Approved by the Head of Department,
B.Tech. CSE

Name: Dr. Nithyanandam P
Date:

(Seal of SCOPE)

ABSTRACT

Because brain strokes may have life-altering effects, it is crucial to recognize them early so that treatment and prevention can begin immediately. To evaluate brain stroke prediction three different machine learning algorithms (XG Boost, Random Forest Classifier, and Logistic Regression) were used. Lifestyle variables, medical histories, and demographic information are just a few of the many pertinent variables included in the dataset that was used. In order to enhance the model's performance, the dataset is cleaned up using stringent preprocessing and feature engineering procedures. This involves deleting extraneous data and lowering noise. A excellent option for calculating the likelihood of a stroke based on input data is Logistic Regression, which is a linear model that is both easy and clearly interpretable. Next, the Random Forest Classifier is trained using ensemble learning to better detect and avoid overfitting as it extracts complicated relationships from the dataset. In addition, the XG Boost method, a robust gradient boosting approach, is used to enhance prediction performance via iterative learning and boosting of weak learners. To make sure the models can handle unknown data well, they are trained on a portion of the dataset and then cross-validated utilizing techniques. Each algorithm's effectiveness is thoroughly assessed using performance metrics such as accuracy, precision, recall, and area under the receiver operating characteristic curve (AUC-ROC). In this review, we look at the results and drawbacks of the research that tested several methods for predicting brain strokes using XG Boost, Random Forest Classifier, and Logistic Regression. The results of this study may pave the way for more accurate and efficient prediction tools, which in turn may help in the early detection of people at risk of stroke, their subsequent medical care, and the formulation of tailored treatment regimens.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Rajesh R., Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Machine Learning.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor i/c & Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to Dr. Nithyanandam P, Head of the Department, B.Tech. CSE and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staffs at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: Gandrath Aryan

CONTENTS

CHAPTER 5 MACHINE LEARNING MODEL TRAINING AND EVALUATION

CHAPTER 6 ENSEMBLE MODELLING AND OPTIMIZATION

::

::

::

::

::

::

::

::

LIST OF FIGURES

LIST OF ACRONYMS

XG	Extreme Gradient Boosting
RF	Random Forest

Chapter 1

Introduction

1.1 INTRODUCTION

Atrial fibrillation poses a significant risk for stroke, especially among individuals aged 65 and above, making stroke the third leading cause of death globally. Often likened to a "heart attack" for the brain, strokes result from the obstruction or reduction of blood supply, leading to ischemic or hemorrhagic events. In both developed and developing nations, stroke stands as a pervasive and potentially fatal consequence of atrial fibrillation. This project acknowledges the gravity of strokes and explores the application of data mining, specifically within the field of AI in medicine, as a promising avenue for prediction and prevention. Focusing on machine learning techniques, the project aims to analyze patient data and predict stroke risk based on factors like age, blood pressure, and glucose levels, is underscored by the project's recognition of the inadequacies in traditional methods for assessing stroke risk, which often lack the precision and efficiency necessary for timely intervention. By identifying subtle patterns and associations, the predictive models developed aim to offer a proactive healthcare approach. This allows healthcare providers to implement preventive measures and interventions before a stroke occurs, addressing the limitations of traditional methods. Ultimately, the project aspires to significantly reduce the burden of stroke-related disabilities and mortality, thereby enhancing the quality of life for individuals at risk.

One potentially fatal side effect of atrial fibrillation is stroke, which can cause mortality. For physicians, making a stroke prediction takes a lot of time and effort. A devastating condition called a stroke usually affects those over 65. It damages the brain similarly to a "coronary episode," which damages the heart and is the third leading cause of death in the United States and other agrarian nations. Hemorrhagic stroke and ischemic stroke are the two main types of strokes. Hemorrhagic stroke occurs when there is bleeding, whereas ischemic stroke occurs when there is no blood flow. Subarachnoid hemorrhage and intracerebral hemorrhage are the two types of hemorrhagic stroke. Another name for transient ischemic attack is "ministroke". When a stroke happens, the brain is deprived of oxygen and nutrients, which causes dead cells to die. In addition to being extremely costly in terms of medical care and resulting in a lifelong impairment, it may ultimately cause death. In general, data mining plays a crucial role in the medical care industry's sickness forecasting. This project uses artificial intelligence (AI) in medicine as a major topic. Using the patient's data, a machine learning model would provide a number of appropriate expectations. With the help of clinical profiles like age, blood pressure, glucose, and so on, the framework can predict patients who may become unwell and extract hidden information from a documented clinical data set. When calculating groupings for the expectation of disease, the number of attributes is taken into account. His clinical history of diseases and strokes is also included in the clinical record. Additionally, we assume that he has already experienced a stroke. Using all of that data, we train the machine using a variety of models, including decision trees, logistic regression, XGBoost, and so on.

1.2 PROJECT STATEMENT

The healthcare sector is focusing on developing accurate prediction models to identify individuals at risk of brain strokes. This involves utilizing machine learning algorithms and datasets like demographics, medical history, lifestyle factors, and genetic markers. The goal is to develop robust models that can predict stroke likelihood within a specified time frame, enabling timely intervention and personalized preventive measures, ultimately reducing stroke incidence and severity and improving patient outcomes.

1.3 OBJECTIVES

The research aims to develop an efficient and accurate machine learning model for predicting stroke risk.
Objectives include identifying key health data variables, optimizing machine learning algorithms, and evaluating the model's performance.
The research strives to contribute to early stroke detection, enabling proactive preventive measures and improved patient outcomes.
The project aims to fill existing gaps in stroke risk assessment methodologies, enhancing healthcare strategies through the integration of advanced machine learning techniques.

1.4 DOMAIN OF THE PROBLEM

The problem involves understanding stroke risk factors, symptoms, and patterns, utilizing machine learning techniques to forecast the incidence of strokes in the medical field by analyzing data, specifically neurology and medical informatics.

1.5 DOMAIN OF THE PROBLEM

1.5.1 MACHINE LEARNING

The research aims to develop an efficient and accurate machine learning model for predicting stroke risk. Objectives include identifying key health data variables, optimizing machine learning algorithms, and evaluating the model's performance. The research strives to contribute to early stroke detection, enabling proactive preventive measures and improved patient outcomes. The project aims to fill existing gaps in stroke risk assessment methodologies, enhancing healthcare strategies through the integration of advanced machine learning techniques.

1.5.2 ARCHITECTURE

The architecture for brain stroke prediction using machine learning involves a systematic approach encompassing data collection, preprocessing, model training, evaluation, deployment, and monitoring. Initially, relevant datasets containing demographic, medical, and lifestyle data are collected. Scaling numerical features, encoding categorical variables, and managing missing values are examples of preprocessing procedures. Next, methods for feature extraction and selection are used to find significant stroke risk factors. The preprocessed data is used to train machine learning models like Logistic Regression, Random Forest Classifier, and XGBoost, which are then assessed using performance measures. Interpretation of model predictions provides insights into factors influencing stroke risk. Upon deployment, models are integrated into healthcare systems with compliance to regulatory standards. Continuous monitoring and periodic retraining ensure model accuracy and adaptability. Ethical considerations like fairness and transparency are addressed throughout the process to ensure responsible deployment and use of the predictive system

Chapter 2

Background

2.1 LITERATURE SURVEY

[1] A devastating condition called a stroke usually affects those over 65. For physicians, making a stroke prediction takes a lot of time and effort. Therefore, the primary goal of the study is to use cutting-edge machine learning techniques to predict the likelihood of a stroke occurring. For improved accuracy, a comparison is performed between five alternative methods. The goal is to develop an application with an intuitive user interface that makes it simple to explore and enter data.

[2] People are now more susceptible to certain diseases like stroke because of the rapid changes in human lifestyles, which have also affected a number of biological aspects of human existence. Stroke is a potentially fatal illness that causes permanent impairment. It's currently the world's biggest cause of death. Furthermore, in Jordan, it ranks second in terms of cause of mortality, behind ischemic heart disease. Early detection of a stroke increases the likelihood of preventing complications and enhances patient care and management. Health care providers can forecast stroke disease and provide a better treatment plan by using orange software, which automatically processes data and generates data mining models. The decision tree classifier performed better than other methods, according to the results, with an accuracy level of 94.2.

[3] A stroke is a medical emergency that happens when the blood supply to a portion of the brain is interrupted. When brain cells are denied the oxygen and glucose necessary for survival, they perish. It is the second most common disease in the world and can be deadly if left untreated. A stroke could be avoided with early intervention and prediction. Algorithms for machine learning are revolutionizing healthcare and are frequently utilized in the early detection of illness. This work focuses on applying machine learning techniques to predict the early occurrence of stroke. In terms of early stroke prediction, this research proves the advantage of Ada Boost over other popular classification techniques.

[4] If blood supply to a part of the brain abruptly stops, a stroke may result. Lack of blood supply can result in handicap based on the injured area of the brain since dying brain cells cause the injury to worsen. Early symptom recognition can have a significant positive impact on both stroke prediction and healthy lifestyle promotion. This study uses a series of machine learning (ML)-developed and -evaluated models to produce a robust pattern for the long-term prediction of stroke incidence risk. This paper primarily contributes a well-performing stacking strategy that is validated by multiple metrics, including decision tree classifier, random forest classifier, XG boost, K closest neighbor, logistic regression, adaboost, catboost, etc. A computer system that mimics human intelligence is becoming more and more commonplace and is being employed in many fields, including health. Stroke medicine is one area of AI application that aims to raise the bar for patient care and diagnosis accuracy. Stroke therapy depends on a precise analysis of stroke imaging. In this study, we provide a brief overview of the application of AI in stroke imaging, emphasizing the fundamentals of technology, clinical requests, and future perspectives.

[5] Predicting a stroke's kind has grown to be a global health concern. The development of stroke deaths will continue in the upcoming year. Numerous studies have been conducted to identify stroke infections. a deep learning-based machine learning strategy for stroke and its kinds prediction. Hemorrhagic stroke, transient ischemic attack, and ischemic stroke are the three types of strokes. In the study, they have employed ML algorithms to identify the type of stroke that may have happened based on an individual's physical condition. We have gathered datasets from the medical facility. Conflicting information, missing data, and duplicate records are eliminated using the pre-processing technique. It uses deep learning to realize characterization in order to predict stroke illness. As soon as the patient subtitles are entered, a prepared model and stroke types are gauged. Better methods for predicting stroke and other types of strokes are the main focus of this effort.

[6] The medical research field nowadays has an abundance of data regarding various ailments. As a result, it's a fantastic resource for doctors to employ when looking for and analyzing early-stage disease causes. Nevertheless, stroke is one of these conditions that requires medical care in order to be treated at an extremely early stage. In general, clinical decision making has made extensive use of stroke prediction approaches. We suggest a method that predicts the occurrence and subtypes of strokes using artificial neural networks and machine learning algorithms. Following the trial, we attempt to evaluate and compare the effectiveness of the two approaches. Because the suggested system has strong real-time properties, it can be used to save time while making clinical predictions. When compared to conventional techniques already in use for the treatment of stroke illnesses, the research's findings are more precise and effective.

[7] A blockage in blood supply to many different areas of the brain causes a stroke, which is one of the leading causes of mortality globally. In an effort to lower the incidence of stroke, numerous studies have developed a deep learning (DL) algorithm-based prediction model for the illness that uses medical information. These research, however, focus less on the predictors (behavioral and demographic). In our study, we identify three important topics for implementing algorithms in the medical field: interpretability, robustness, and generalization. We suggest using random forests to forecast the incidence of strokes based on this background information. With a macro F1 score of 94%, the results of our experiment demonstrated that random forest (RF) performed better than decision tree (DT) and logistic regression (LR). According to our research, the two most important factors predicting the occurrence of stroke disease were age and body mass index (BMI).

[8] A stroke is a medical illness where a blood artery bursts, damaging the brain. Symptoms could appear if there is an irregular flow of the blood. Another cause of death and common illness is stroke. In developing nations, stroke is more common, with ischemic stroke being the most common type. By identifying many warning signs, the severity of a stroke can be reduced. The majority of the older models for predicting and detecting strokes rely on expensive and challenging to use image inspection equipment, such magnetic resonance imaging (MRI) and computed tomography (CT) scans), for real-time recognition. A subset of artificial intelligence called machine learning (ML) allows software programs to accurately predict outcomes without requiring human intervention to complete tasks. Because ML algorithms produce accurate findings in the medical domain, they have attracted a lot of attention recently. Therefore, a machine learning algorithm-based stroke illness diagnosis system is described in this work. In this study, artificial neural networks (ANNs) are the machine learning algorithm. A comparison is made between the offered ML algorithm's result analysis and that of other ML methods. To determine whether algorithm performs better for stroke identification, the performance of the suggested strategy is compared.

[9] A subset of artificial intelligence called machine learning (ML) allows software programs to accurately predict outcomes without requiring human intervention to complete tasks. The purpose of this review is to categorize and evaluate the machine learning techniques applied to stroke prediction. In order to evaluate the machine learning methods applied to stroke predictions, we have taken into account the previously published studies. It has been discovered that the majority of study has focused on the projected outcomes of mortality rate and functional outcome. Neural networks, decision trees, random forests, and support vector machines were the most often utilized approaches. Nevertheless, a small number of classifiers and predictors performed rudimentary reporting standards for medical industry instruments, and none of them turned out to be really useful.

[10] A stroke, also known as a brain attack, happens when there is insufficient blood supply to the brain or when a blood artery bursts inside the brain. Numerous studies have been conducted in an attempt to accurately anticipate the course of different diseases by examining the various parameters that are related to it and comparing the effectiveness of various predictive data mining systems. We present a comparison of different classification methods using the Stroke-Prediction-Dataset, which includes parameters such as BMI, smoking status, hypertension, and others. Here, the unbalanced dataset is balanced using ADASYN, and the missing values are filled in using Scikit-Learn's SimpleImputer. When two classification models, Random Forest and XGBoost, are compared; Random Forest's accuracy (97.67%) narrowly beats that of the XGBoost Classifier (96.894%).

[11] Stroke is the leading cause of mortality and end-stage disease in many countries. Determining ways to improve things was the aim of this study. I made advantage of Kaggle's stroke disease data set. Preprocessed data can have advantages for patients. There are two types of stroke: ischemic stroke and stroke hemorrhage. Machine learning techniques are used to categorize people into these two groups. In this inquiry, machine learning techniques were used seven times. Support Vector Machine (SVM), Random Forest, Multi-layer Perceptron (MLP), Naive Bayes, Cat Boost, Logistic Regression, and KNearest Neighbors This is why, according to our research, Cat Boost produces the best accuracy,recall,precision,andf1-Scorescores.

[12] This paper's main objective is to forecast coronary heart stroke. Heart attacks are becoming more commonplace worldwide, including in children and teenagers. Stroke prediction is a complex field that requires pre-processing of a huge number of records. To prevent strokes, it is desirable to automate the process of early identification of stroke symptoms. A dataset within the proposed model is used to predict heart attacks. To determine a person's risk level, it employs system mastering techniques such as Random Woods, Okay Nearest Neighbor (KNN), and Selection Tree. Consequently, the maximum green method is identified and an evaluation of the different algorithms is provided.

2.2 LIMITATION IN THE PRESENT MODELS

Data Quality and Availability: Finding complete, high-quality datasets to train machine learning models is still a difficulty. Datasets may be limited in size or scope, leading to potential biases or incomplete representations of stroke risk factors.

Feature Selection: Identifying relevant features for stroke prediction is complex due to the multifactorial nature of stroke. Current models may not effectively capture all relevant risk factors, leading to suboptimal performance.

Generalization to Diverse Populations: Machine learning models trained on specific populations may not generalize well to diverse demographic groups or different healthcare settings. Variations in demographics, lifestyles, and healthcare practices can affect model performance.

Interpretability: Understanding the fundamental principles behind stroke predictions is difficult due to the lack of interpretability in many ML models. This limits their clinical utility and hampers trust among healthcare professionals.

Temporal Dynamics: Stroke risk factors and patient characteristics may change over time, requiring dynamic modeling approaches. Present models may not adequately capture temporal dynamics, leading to outdated predictions and reduced accuracy over time.

Rare Events and Imbalanced Data: Since strokes are comparatively uncommon events, datasets with a disproportionately small number of stroke cases compared to non-stroke cases may be unbalanced. Models that are biased towards the majority class due to imbalanced data may perform poorly when applied to the minority class.

Handling Uncertainty: Machine learning models often provide point predictions without quantifying uncertainty. Incorporating uncertainty estimation techniques could enhance the reliability and robustness of stroke predictions, especially in critical decision-making scenarios.

Integration with Clinical Workflow: Seamless integration of machine learning models into clinical workflows poses technical and logistical challenges. Models need to be interoperable with existing healthcare systems, ensuring easy access by healthcare providers and real-time decision support.

Ethical and Legal Considerations: There are ethical concerns regarding patient privacy, consent, and the responsible use of predictive models in healthcare.

2.3 INFERENCE

Machine learning techniques like Random Forest and XGBoost have great potential in the field of drug classification and development. These algorithms can be used to analyze medication data because they are very effective at regression and classification tasks. The most effective approach in drug development is to create a variety of drugs, including small molecules, peptides, antibodies, and more modern modalities such short RNAs or cell therapies.

In order to detect illicit drugs based on a variety of parameters, such as drug reports and the makeup of substances collected, machine learning algorithms are essential. These models aid in the fight against the illegal drug trade by using data mining algorithms to categorize pharmaceuticals into appropriate and inappropriate groups. Even if there are fewer newly listed pharmaceuticals in the world, there are still a lot of risks and expenses involved with drug classification and research.

In particular, XGBoost has been compared to other algorithms such as gradient boosting machines and random forest, demonstrating its efficacy and versatility in drug-related investigations. More than ever, there is a need for sophisticated machine learning approaches to identify and classify drugs due to the introduction of new psychoactive compounds into the illicit market.

The promise of machine learning in drug discovery and repurposing has been demonstrated by the application of machine learning techniques to the repurposing of DrugBank molecules, particularly in the identification of possible medicines targeting opioid receptors. Utilizing machine learning algorithms, data-driven classification techniques provide valuable information about opioid patient profiles, facilitating the examination and comprehension of prescription medication usage trends.

In order to predict drug addiction susceptibility, researchers have created machine learning techniques that use datasets to evaluate susceptibility. This could help with preventive treatments. The adaptability of machine learning approaches has been demonstrated by the application of these frameworks to predict substance usage, including prescription opioids, alcohol, cocaine, marijuana, and methamphetamine.

Machine learning has promise for quick detection of both established and newly discovered opioid compounds because it can identify a wide range of drugs with high accuracy, including fentanyl.
Evaluation research combining prescription medication and claims data with machine learning algorithms exhibits promise in identifying the likelihood of an overdose, indicating the potential applications of these techniques in healthcare environmen

Chapter 3

Proposed System

Logistic Regression, Random Forest Classifier, and XG Boost are three important machine learning techniques that our suggested system would use to build a strong predictive model for determining an individual's probability of having a brain stroke. Starting with a broad patient dataset that includes both persons with and without a history of stroke, the process moves on to complete data collecting. This involves sourcing factors such as blood pressure, cholesterol levels, age, gender, medical history, lifestyle behaviors, and genetic predispositions. Data cleaning and preparation for analysis are goals of the preprocessing activities carried out after data gathering. To do this, you must encode categorical variables, normalize feature scales, and deal with missing data. To further improve the model's discriminative strength, we use advanced feature selection approaches to find and rank the most important predictors. An essential part of the suggested system is the deliberate choice of machine learning algorithms. While XG Boost provides iterative improvement and Random Forest Classifier is great at capturing complicated associations, Logistic Regression provides interpretability. Because each algorithm is trained separately on the labeled dataset, they may all see complex correlations and patterns in the data. Our suggested approach stands out because it integrates the separate models into an ensemble. The goal of this ensemble method is to build a better prediction model for assessing the risk of brain stroke by combining the best features of each algorithm. Metrics like F1 score, accuracy, precision, and recall are thoroughly both ensemble the individual algorithms. A detailed comprehension of the model's capacities and efficacy is guaranteed by this exhaustive evaluation. Each algorithm's hyperparameters are fine-tuned to maximize their effectiveness in the context of predicting brain strokes. The maximum potential predicted accuracy is the goal of this continual improvement procedure. The last stage of our suggested approach is to validate the final ensemble model using new data that has never been seen before. This contributes to the early intervention and preventative actions for high-risk patients and guarantees the model's generalizability and reliability in forecasting the chance of brain stroke occurrences. To sum up, our suggested methodology provides a robust and organized method for predicting brain strokes, which may improve healthcare methods via more accurate and dependable decisions.

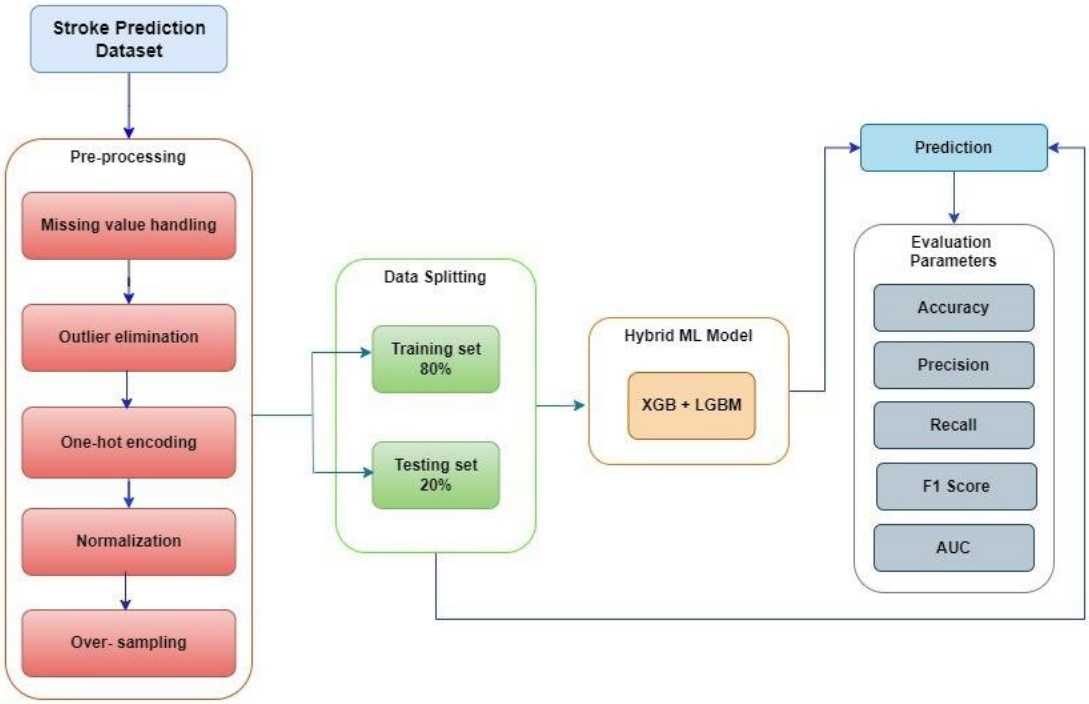


Figure 1. Proposed architecture diagram

3.1 MODULES / ALGORITHM/ FUNCTIONALITIES/ PROTOCOLS

Feature Selection Module - Identifies crucial health data variables for stroke prediction.
Machine Learning Model Module - Selects and implements machine learning algorithms for training the predictive model.
Evaluation Module - Assesses the performance, accuracy of trained model.
Deployment Module - Integrates the model into a user-friendly system for practical use.

3.2 DATA COLLECTION APPROACHES/ STRATEGIES

Advantage of Strategy:- Utilizes electronic health records for comprehensive and continuous data.
Limitation of Strategy:- May face challenges related to data quality and interoperability.
Potential Risk:- Privacy concerns and potential inaccuracies in health records.
Ethical Issues:- Ensures patient consent and addresses privacy concerns during data collection.

3.3 DATA ANALYSIS APPROACHES

3.3.1 FEATURE SELECTION MODULES

This is crucial for identifying the most relevant health data variables influencing stroke risk. This module aims to select a subset of features that contribute significantly to the predictive model. The rationale is to enhance model efficiency, reduce dimensionality, and mitigate the risk of overfitting. Ethical considerations guide the selection process, ensuring that only necessary and pertinent variables are utilized, respecting patient privacy and data protection regulations. This module plays a pivotal role in optimizing the model's performance, contributing to a more interpretable and efficient stroke prediction system in healthcare, with a focus on accuracy and ethical data utilization.

3.3.2 MACHINE LEARNING MODEL

The Machine Learning Model in the Stroke Disease Prediction Project is the core component responsible for training and predicting stroke risk based on selected health data. Leveraging algorithms like logistic regression or decision trees, it learns patterns from a labeled dataset. The model undergoes rigorous training, optimizing weights and parameters to accurately classify instances. Regularization techniques enhance generalization, preventing overfitting. Cross-validation ensures robust performance across diverse datasets. The model's interpretability is considered, addressing ethical concerns in healthcare decisions. Its ability to analyze complex relationships within health data contributes to precise stroke predictions. Continuous refinement and evaluation are vital, ensuring the model's accuracy and reliability, ultimately facilitating early intervention and improving outcomes in stroke risk assessment for preventive healthcare.

Supervised Learning - Utilizes labeled datasets for training the machine learning model.
Feature Importance Analysis - Identifies key variables contributing to stroke prediction.
Cross-validation Techniques - Ensures robustness and generalization of the model.
Interpretability Methods - Addresses ethical concerns by making the model's decisions more interpretable.

3.4 EVALUATION

A range of methods and criteria are applied to evaluate the model's ability to generalize to new, untested data. Accuracy, precision, recall, F1 score, and area under the Receiver Operating Characteristic (ROC) curve are typical evaluation criteria. These measures shed light on how well the model predicts the likelihood of stroke and prevents false positives. By evaluating the model over several dataset subsets, cross-validation methods like k-fold cross-validation guarantee robustness. Ethical considerations guide the evaluation process, emphasizing the importance of interpretability and transparency in healthcare decision-making. The Evaluation Module is essential for validating the effectiveness of the model and ensuring its practical utility in stroke risk prediction for

improved patient outcomes.

3.5 DEPLOYMENT

The Deployment focuses on the practical integration of the trained model into real-world healthcare systems. It involves creating a user-friendly interface for clinicians to interact with the model, providing them with accessible and actionable insights into stroke risk. The deployment phase ensures seamless integration, addressing compatibility issues and optimizing performance in diverse healthcare environments. Continuous monitoring and maintenance protocols are established to uphold the model's reliability. Ethical considerations, such as patient consent and privacy, are paramount during deployment, ensuring responsible and secure usage of the predictive model. The Deployment Module is pivotal in translating the research outcomes into practical healthcare applications, contributing to the early detection and proactive management of stroke risks, ultimately improving patient care and outcomes.

Chapter 4

Data Collection and Pre-Processing

4.1 DATA COLLECTION

This is integral to acquiring diverse and comprehensive health data. It involves collecting information from various sources, including electronic health records, patient demographics, medical histories, and lifestyle factors. The advantage of leveraging electronic health records ensures a continuous and real-time influx of data, facilitating timely updates to the predictive model. However, challenges may arise in data quality, interoperability, and potential biases. The strategy prioritizes patient privacy and ethical considerations, requiring informed consent for data use. By addressing these challenges, the Data Collection module ensures the development of a robust and accurate stroke prediction model, enhancing the potential for early detection and intervention, ultimately contributing to improved healthcare outcomes in stroke management.

4.2 DATA PREPROCESSING

This module is pivotal in refining raw health data for effective model training. It includes a range of activities, including encoding category variables, addressing missing values, and normalizing numerical features. Feature scaling ensures uniformity, and dimensionality reduction techniques may be applied. Addressing outliers and noise enhances data quality. Additionally, the module includes handling class imbalances in the dataset. By employing these preprocessing steps, the data becomes suitable for machine learning algorithms, contributing to model accuracy and generalization. Ethical considerations are paramount, particularly in handling sensitive health data, emphasizing privacy and adherence to regulations. Ultimately influencing the accuracy and effectiveness of the stroke prediction model in healthcare applications.

4.2.1 DATA CORRELATION HEATMAP

The correlation heatmap below depicts the correlation coefficients between different features in the solar power dataset.

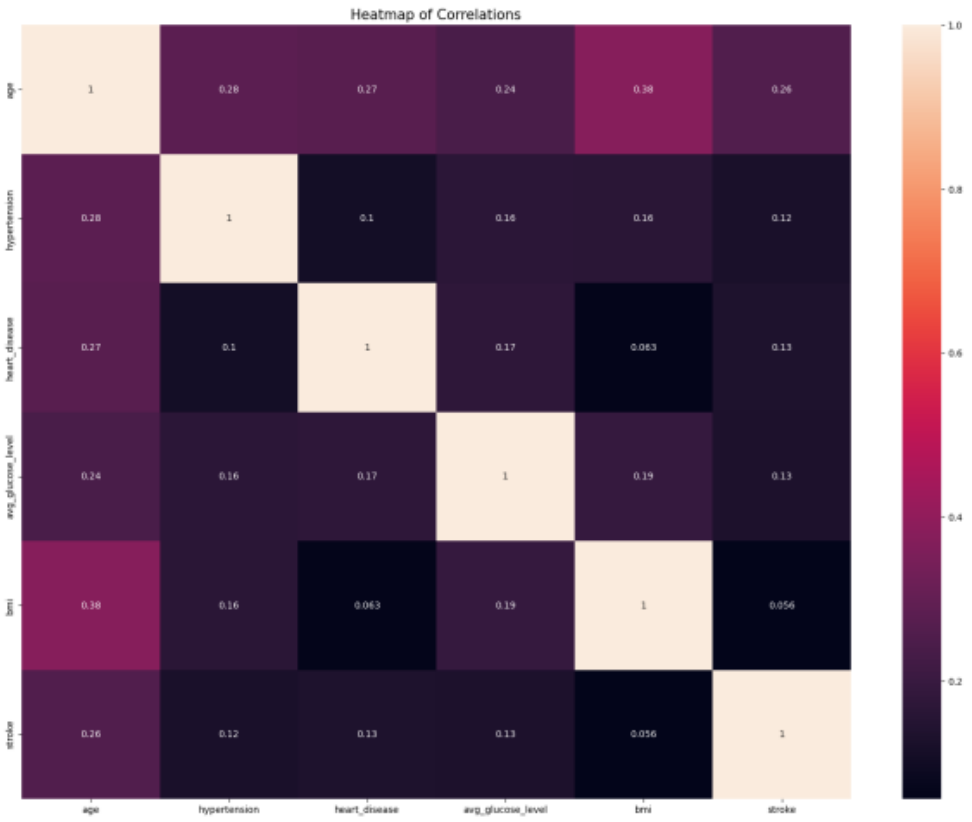


Figure 2. data correlation heatmap

Chapter 5

Machine Learning Model Training and Evaluation

5.1 REGRESSION MODELS FOR BRAIN STROKE PREDICTION

An essential part of the system for predicting brain strokes is the Machine Learning Model Training Module, which uses the preprocessed data to train and optimize the prediction models. Models reliably chance of a brain stroke in people are built in this module using a variety of methods, reduce prediction errors and maximize performance, the models are trained using the preprocessed data via repeatedly modifying their parameters. Models are often validated and checked for generalizability to new data instances using cross-validation procedures. This module also handles hyperparameter tweaking, which is used to improve the model's prediction skills by its system reliably and anticipate brain strokes via the training and optimization of machine learning models in this module.

Random Forest:

The Random Forest algorithm, which is based on machine learning, mixes several decision trees to provide an effective result. Based on data samples, the random forest algorithm creates decision trees and uses voting to determine which solution is better. It builds a tree for the data and uses that to inform its predictions. The Random Forest approach may be applied to big datasets and yield consistent results even in cases when there are significant gaps in the record values. The samples produced by the decision tree can be saved for use with other kinds of data. The two steps of random forest are to produce a random forest and then utilize the first stage's generated random forest classifier to forecast the future.

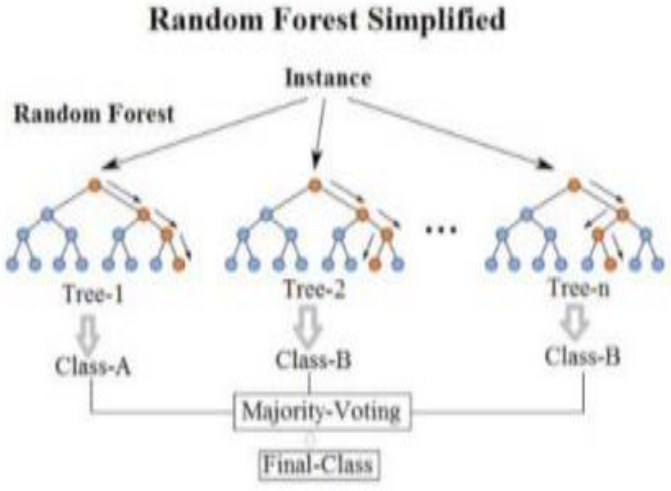


Figure 3. Random Forest Model diagram

A supervised learning system called the random forest combines several decision trees into a single "forest" at random. To increase accuracy, it is preferable to use a variety of decision models rather than just one learning model. This method's main distinction from

the conventional decision tree algorithm is the random generation of the root nodes' splitting nodes.

XG BOOST: A tree-based ensemble learning approach called XGBoost (Extreme Gradient Boosting) makes advantage of gradient boosting to enhance model performance. Here's how XGBoost's structure can be explained:
Decision Trees: XGBoost is a group of decision trees, each of which is trained one after the other to fix the mistakes of the preceding tree.

Boosting: With gradient boosting, a technique used by XGBoost, the algorithm first trains a weak learner (a decision tree in this example), and then iteratively trains other weak learners to fix the mistakes made by the earlier learners. The residuals of the preceding tree the discrepancy between the expected and actual values are used to train each new tree.

Regularization: Regularization is used by XGBoost to avoid overfitting. L1 regularization (lasso regression) and L2 regularization (ridge regression) are the two methods of regularization that are included in it. Some of the coefficients are reduced to zero by L1 regularization, which effectively eliminates those features from the model. While L2 regularization reduces the coefficients to zero, it leaves all features in place.

Split Finding: The optimal split is determined using XGBoost using a greedy approach at each decision tree node. It assesses every split that could be made and chooses the one that reduces the loss function (often the mean squared error) the most.

Parallel Processing: The purpose of XGBoost is to benefit from parallel processing. Multi-threading is used to accelerate prediction and training.
All things considered, XGBoost is a strong algorithm that works well for both regression and classification issues. To create precise and effective models, it combines the benefits of decision trees with the effectiveness of gradient boosting and regularization.

Logistic Regression: Regression analysis techniques such as logistic regression are used to forecast the likelihood of a binary result (0 or 1, True or False, Yes or No). By utilizing a logistic function to estimate probabilities.

Multiple input features, including age, blood pressure, cholesterol, smoking status, and medical history, can be included using logistic regression. Imaging data, patient surveys, and medical records can all yield these traits. A probabilistic interpretation of predictions is offered by logistic regression, which is very helpful for diagnosing medical conditions. Logistic regression can be used to predict brain strokes by estimating a person's chance of having a stroke based on their health and demographic data. These probabilities

can be used by medical practitioners to identify high-risk patients who should get preventive treatments including medication, lifestyle changes, or routine monitoring.

5.2 MODEL TRAINING PROCEDURE

The model training procedure commences with data splitting of two parts training and testing sets, typically adhering to a predefined ratio to balance model learning and evaluation. The training set is utilized to fit the model parameters through optimization techniques such as gradient descent or its variants. Simultaneously, the testing set remains unseen during training and serves as an independent dataset for evaluating model generalization to unseen data. Hyperparameter tuning is a crucial step in model training, involving strategies like grid search or randomized search to fine-tune model performance. Hyperparameters control aspects such as model complexity, regularization strength.

5.3 MODEL EVALUATION METRICS

Various criteria and approaches are used to know how effectively the model makes to new, untested data. Accuracy, precision, recall, F1 score, and area under the Receiver Operating Characteristic (ROC) curve are examples of common evaluation metrics. These measures shed light on how well the model predicts the likelihood of stroke and prevents false positives. By evaluating the model over several dataset subsets, cross-validation methods like k-fold cross-validation guarantee robustness. The review process is guided by ethical considerations, which underscore the significance of interpretability and transparency in the process of making healthcare decisions. The Assessment Module is crucial for verifying the model's efficacy and guaranteeing its usefulness in predicting stroke risk for better patient outcomes.

5.4 DATASET RESULTS

The accuracy of Logistic Regression

AUC on Test data is 0.7944421740907234

AUC on Train data is 0.8009807928075194

The accuracy of Random forest classifier

AUC on Test data is 0.9489170412750306

AUC on Train data is 1.0

The accuracy of XG boost

AUC on Test data is 0.9501430322844299

AUC on Train data is 0.9967306906416019

For XGBoost:

AUC on Test data is 0.9501430322844299

Chapter 6

Results and Discussion

A state-of-the-art technical solution, the machine learning system for brain stroke prediction analyses different patient data using sophisticated algorithms to estimate the chance of a person getting a stroke. Age, medical history, lifestyle choices, and genetic predispositions are some of the variables that this approach takes into account when calculating the likelihood of a stroke. Machine learning algorithms can analyze massive databases for trends, allowing doctors to make more informed judgments on preventative care and individualized treatment programs based on correct predictions. Another benefit of real-time monitoring is the ability to identify warning symptoms early on, which means that stroke prevention measures may be taken promptly. All things considered, this approach has a lot of potential in enhancing methods for preventing strokes, increasing patient care, and, in the end, saving lives by using AI and predictive analytics.
The models' robustness and generalizability will be guaranteed by using cross-validation procedures.

6.1 OUTPUT VISUALIZATION

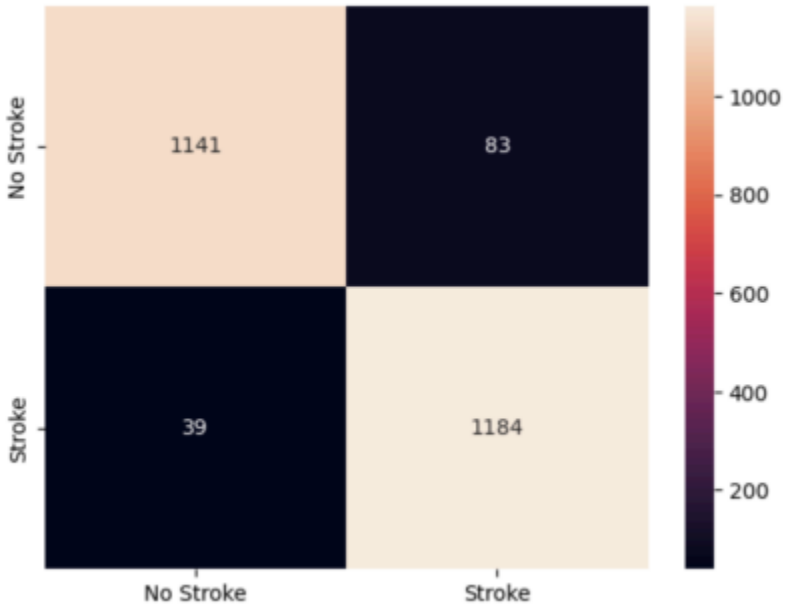


Figure 4. Confusion Matrix(Brain Stroke Prediction)

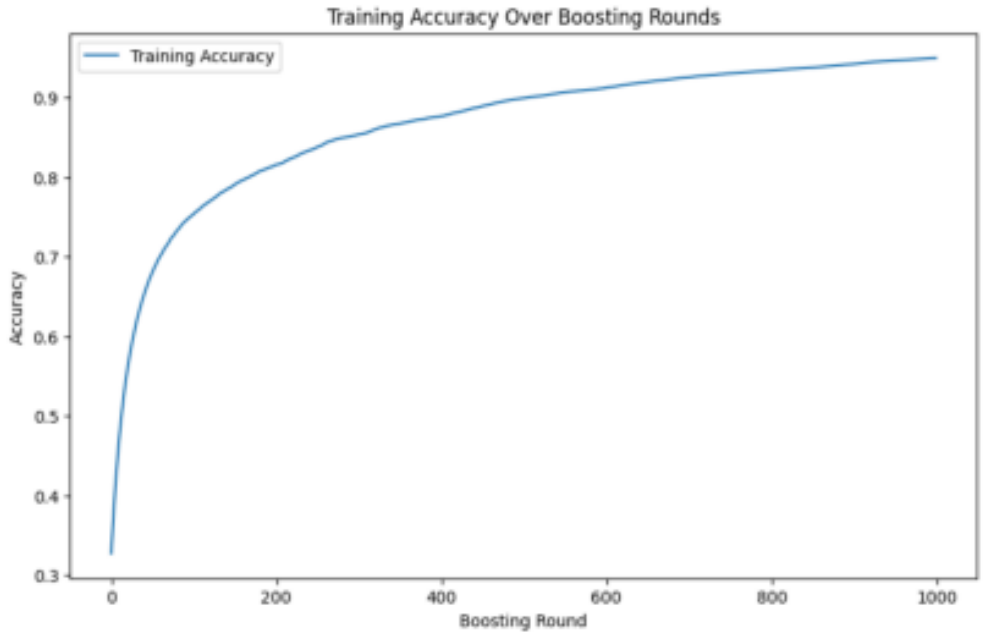


Figure 5. Training Accuracy

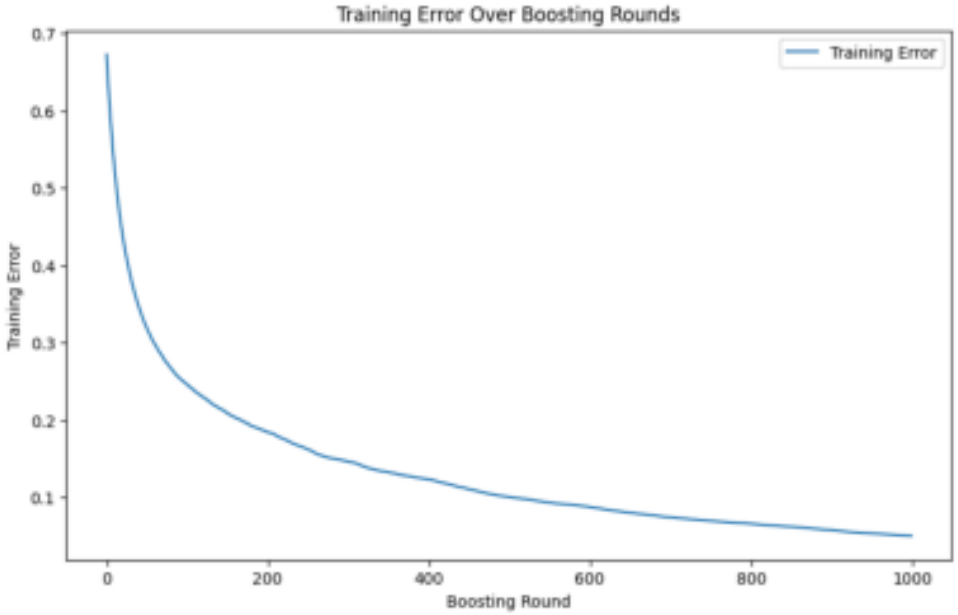


Figure 6. Training Loss

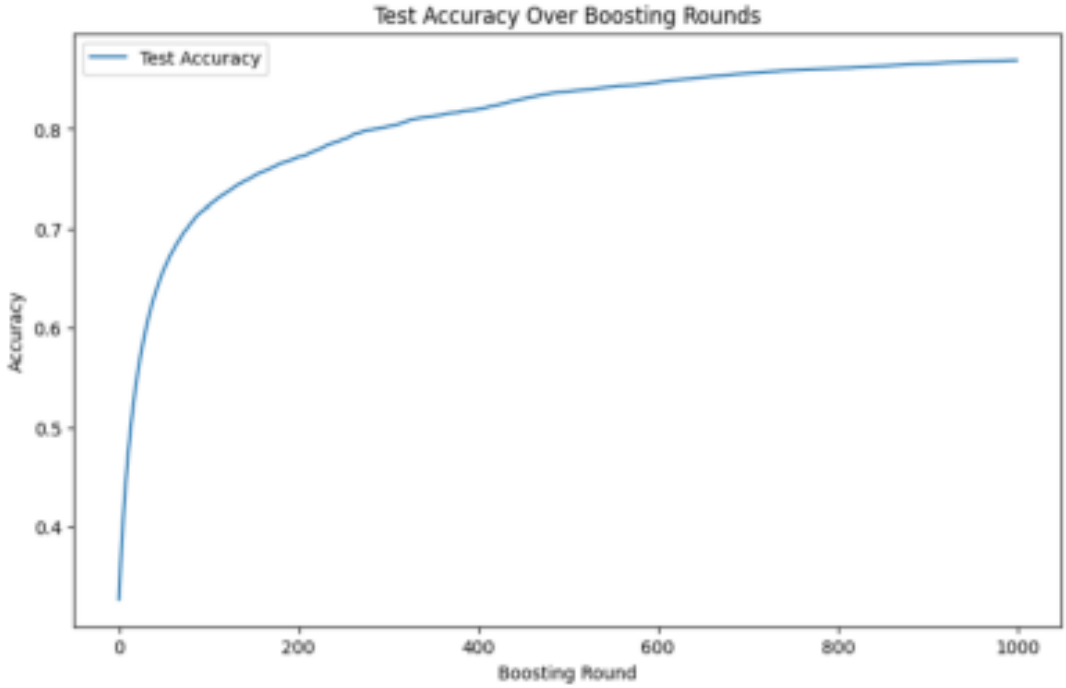


Figure 7. Test Accuracy

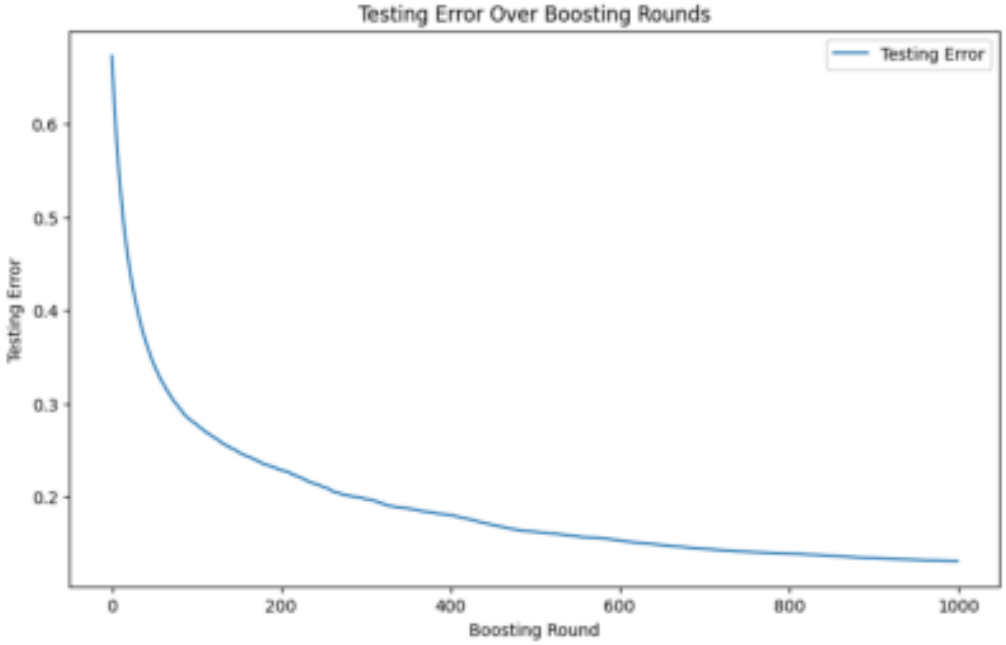


Figure 8. Test Loss

These AUC discriminatory power, XGBoost exhibits notably high AUC values on both test and train datasets, indicating strong predictive capabilities.

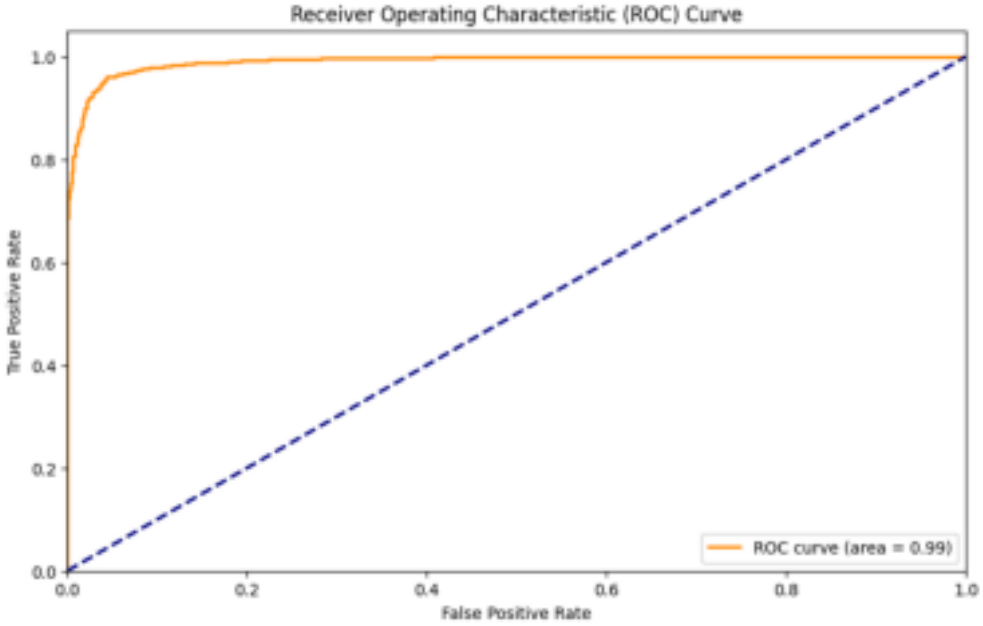


Figure 9. ROC Curve

Classification Report:	precision	recall	f1-score	support
0	0.97	0.93	0.95	1224
1	0.93	0.97	0.95	1223
accuracy			0.95	2447
macro avg	0.95	0.95	0.95	2447
weighted avg	0.95	0.95	0.95	2447

Figure 10. Classification Report

The hybrid model in this context refers to an ensemble approach that combines predictions from two distinct machine learning models: XGBoost and LightGBM. Both models are individually trained on the same dataset to detect brain strokes. Following training, their predictions on test data are combined to create a hybrid model. The combination strategy involves taking the average of the predicted probabilities from the XGBoost and LightGBM models. This blending technique aims to exploit the diverse strengths and patterns captured by each model, potentially improving overall prediction accuracy. The resulting hybrid predictions are then converted to binary outcomes using a threshold of 0.5. Ensemble models, such as this hybrid approach, often demonstrate superior performance by mitigating the weaknesses of individual models and leveraging their complementary strengths. By combining the predictive capabilities of XGBoost and LightGBM, the hybrid model seeks to provide a more robust and accurate solution for brain stroke detection. However, it's crucial to note that the effectiveness of the hybrid model may vary based on the specific characteristics of the dataset and the hyperparameters chosen during training. Regular evaluation and potential fine-tuning are essential to ensure optimal performance in real-world scenarios.

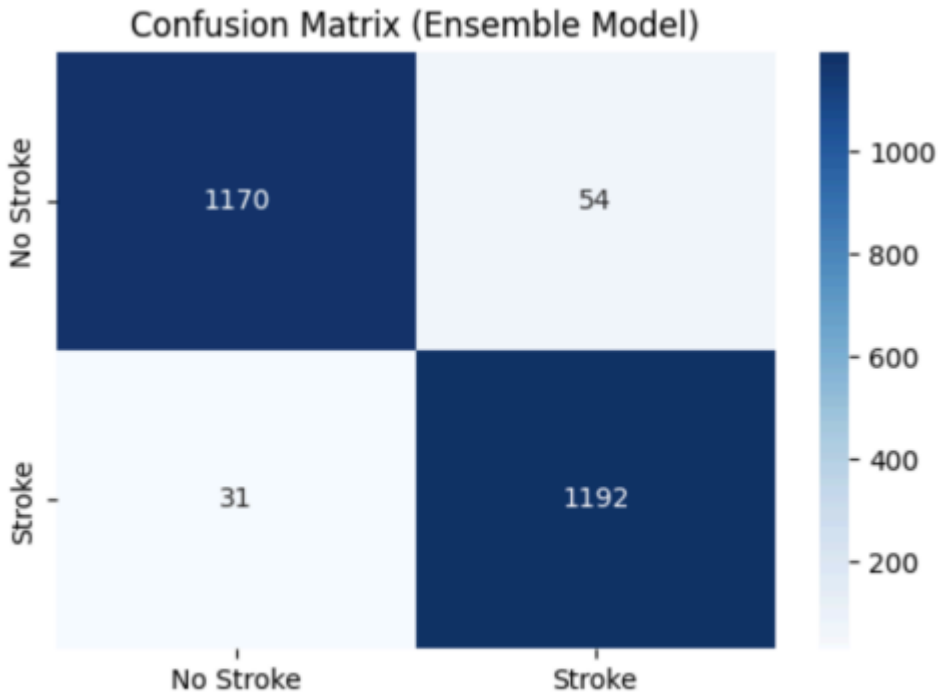


Figure 11. Confusion Matrix (Ensemble Model)

Ensemble Model Accuracy: 96.53%

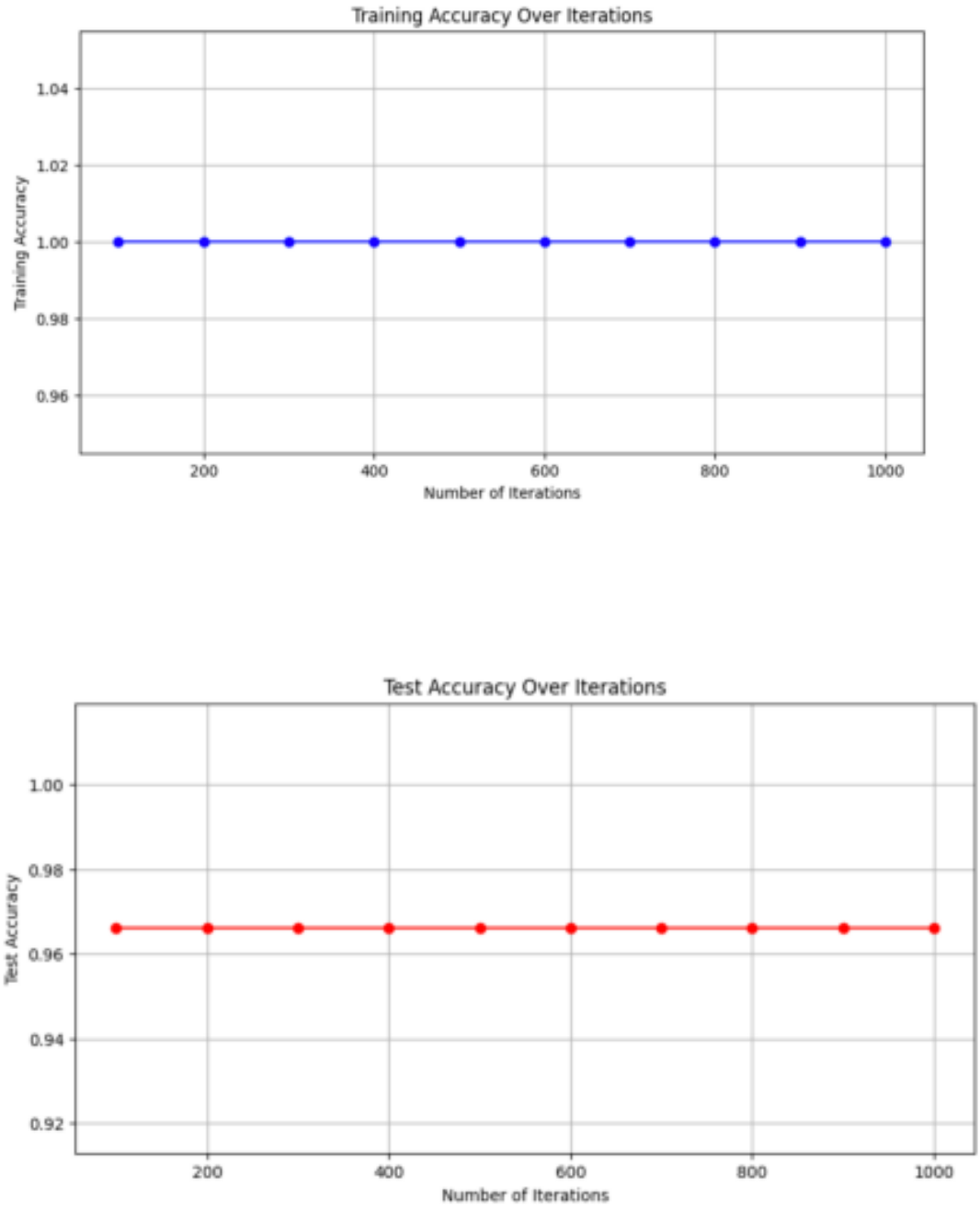


Figure 12. Training Accuracy Over Iterations

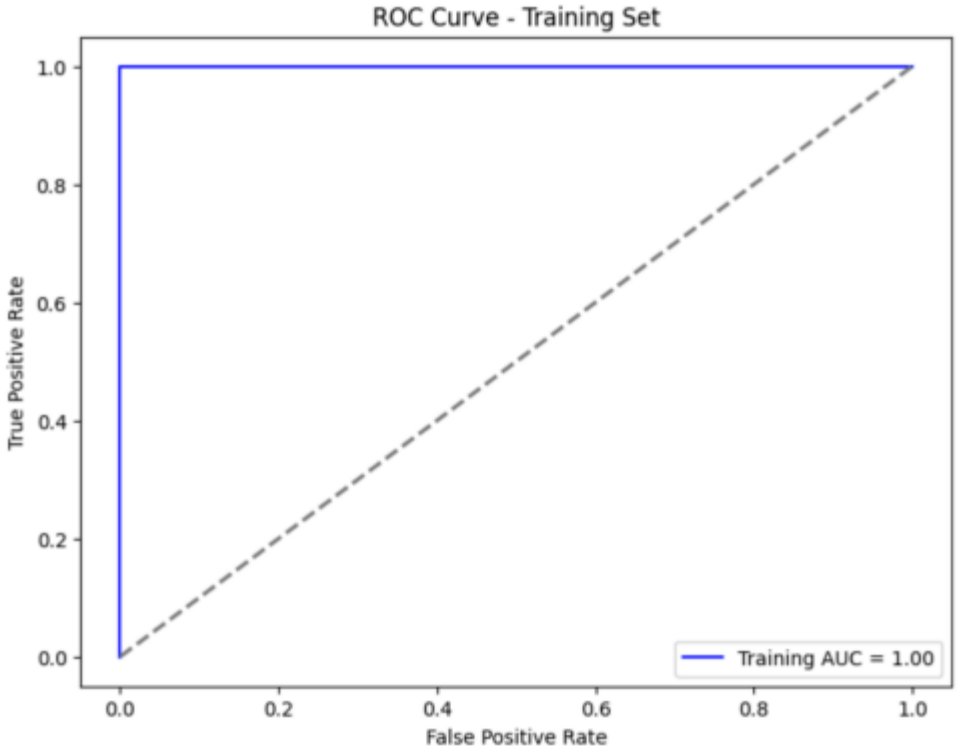


Figure 13. ROC Curve – Training Set

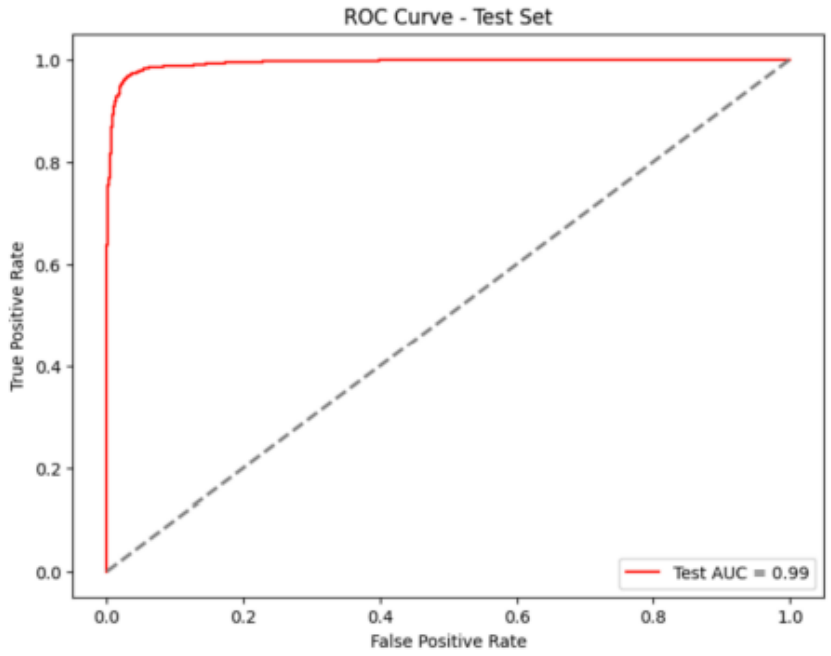


Figure 14. ROC Curve – Test Set

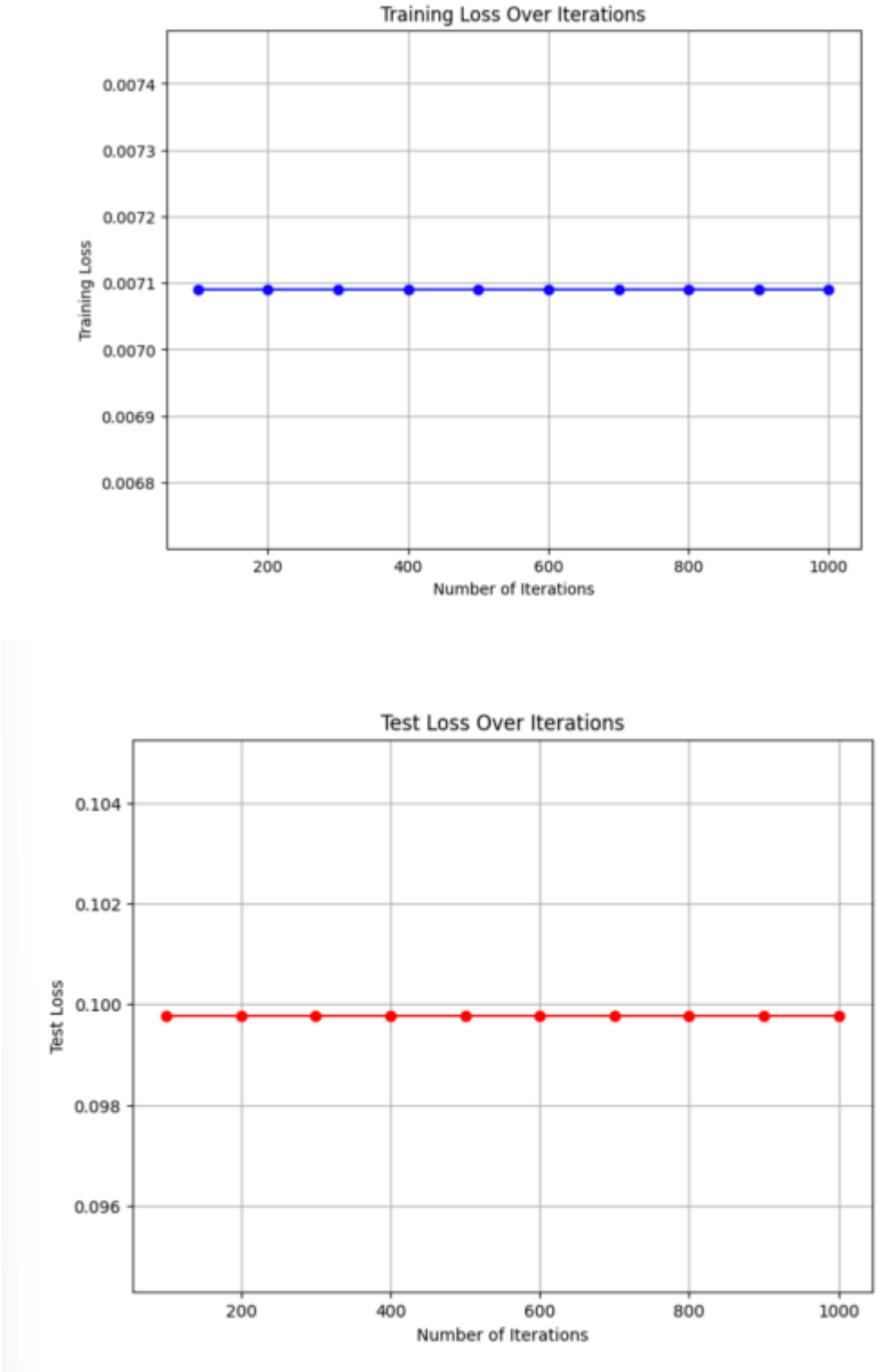


Figure 15. Training Loss Over Iterations

Ensemble Model Accuracy: 96.53%
Classification Report:

	precision	recall	f1-score	support
No Stroke	0.97	0.96	0.96	1224
Stroke	0.96	0.97	0.97	1223
accuracy			0.97	2447
macro avg	0.97	0.97	0.97	2447
weighted avg	0.97	0.97	0.97	2447

Figure 16. Classification Report

FRONTEND:

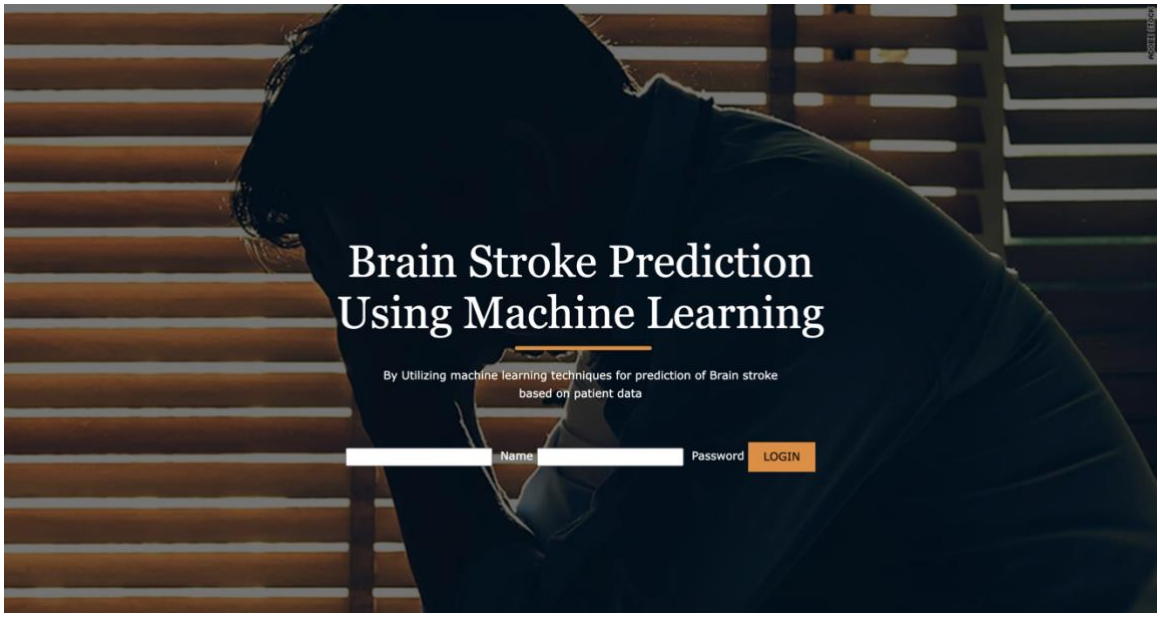


Figure 17. Login page

Figure 18. Entering inputs

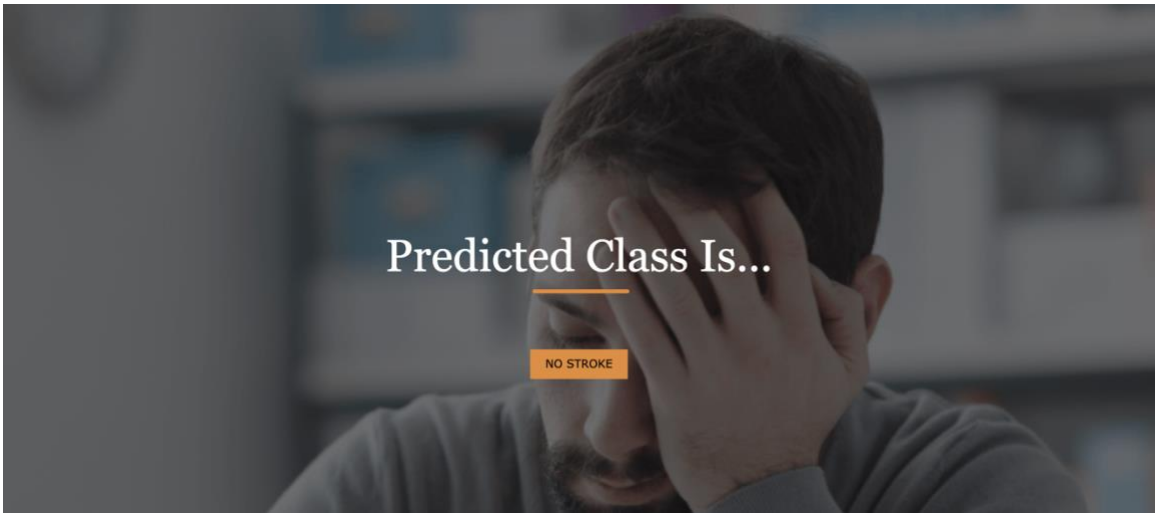


Figure 19. Output Results

Chapter 7

Conclusion and Future Work

In conclusion, the system for brain stroke prediction employing machine learning has showcased promising outcomes, particularly with the utilization of Logistic Regression, Random Forest Classifier, and XG Boost. Notably, among these algorithms, XG Boost emerged as the most accurate in identifying individuals at risk of experiencing a stroke. Through in-depth analysis of extensive datasets and intricate patterns, these machine learning algorithms offer valuable insights that can assist healthcare professionals in early detection and the formulation of effective prevention strategies. The demonstrated accuracy of XG Boost, in particular, holds substantial potential for significantly improving patient outcomes. By enabling proactive interventions and facilitating the creation of personalized treatment plans, this system contributes to a more targeted and efficient approach in addressing stroke risks. In order to improve the system's overall dependability and efficacy in clinical practice, more validation studies and real-world application are necessary, despite these encouraging findings. The importance of XG Boost in the field of brain stroke prediction is shown by the preference for it based on accuracy. Utilizing machine learning, particularly XG Boost's capabilities, might revolutionize healthcare delivery and improve patient care by allowing for more accurate risk assessment and faster actions.

To further it would be advantageous to include sophisticated algorithms like deep learning models into future work on the system for brain stroke prediction using machine learning. Conducting a comprehensive analysis of a larger dataset with diverse demographic and clinical variables would also contribute to the robustness of the predictive model. Additionally, integrating real-time monitoring and feedback capabilities into the system could enable timely interventions and personalized stroke prevention strategies for individuals at high risk. Further research on feature selection techniques, model interpretability, and validation methods would be essential for ensuring the reliability and generalizability of the predictive model. Collaboration with healthcare professionals and institutions for clinical validation and deployment of the system in real-world settings should also be considered to evaluate its impact on patient outcomes and healthcare decision-making.

Appendices

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression, RidgeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
import xgboost as xgb
from xgboost import XGBRegressor
from sklearn.metrics import accuracy_score, confusion_matrix, f1_score
```

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df=pd.read_csv(r"/content/drive/MyDrive/Brain_stroke.csv")
df
```

```
[3]:
      gender   age  hypertension  heart_disease ever_married      work_type  \
0       Male  67.0             0              1          Yes        Private
1       Male  80.0             0              1          Yes        Private
2     Female  49.0             0              0          Yes        Private
3     Female  79.0             1              0          Yes  Self-employed
4       Male  81.0             0              0          Yes        Private
...      ...   ...           ...            ...          ...            ...
5177    Male  41.0             0              0           No        Private
5178    Male  40.0             0              0          Yes        Private
5179  Female  45.0             1              0          Yes       Govt_job
5180    Male  40.0             0              0          Yes        Private
5181  Female  80.0             1              0          Yes        Private

     Residence_type  avg_glucose_level        bmi   smoking_status  stroke
0             Urban             228.69  36.600000  formerly smoked       1
1             Rural             105.92  32.500000     never smoked       1
2             Urban             171.23  34.400000           smokes       1
```

```
[ ] df.shape
```

```
(5182, 11)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5182 entries, 0 to 5181
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             5182 non-null   object 
 1   age                5182 non-null   float64
 2   hypertension       5182 non-null   int64  
 3   heart_disease      5182 non-null   int64  
 4   ever_married       5182 non-null   object 
 5   work_type          5182 non-null   object 
 6   Residence_type     5182 non-null   object 
 7   avg_glucose_level  5182 non-null   float64
 8   bmi                5182 non-null   float64
 9   smoking_status     5182 non-null   object 
 10  stroke             5182 non-null   int64  
dtypes: float64(3), int64(3), object(5)
memory usage: 445.5+ KB
```

```
[ ] df.describe()
```

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5182.000000	5182.000000	5182.000000	5182.000000	5182.000000	5182.000000
mean	43.754574	0.101505	0.059437	106.749620	28.530705	0.055577
std	22.706994	0.302026	0.236463	45.875281	6.692112	0.229125
min	0.080000	0.000000	0.000000	55.120000	14.000000	0.000000
25%	26.000000	0.000000	0.000000	77.432500	23.900000	0.000000
50%	45.000000	0.000000	0.000000	92.050000	28.300000	0.000000
75%	62.000000	0.000000	0.000000	114.340000	32.500000	0.000000
max	82.000000	1.000000	1.000000	271.740000	48.900000	1.000000

```
[ ] df.isnull().sum()
```

```
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
dtype: int64
```

```
[ ] df["work_type"].unique()
```

```
array(['Private', 'Self-employed', 'Govt_job', 'children'], dtype=object)
```

```
[ ] df["gender"].unique()
```

```
array(['Male', 'Female'], dtype=object)
```

```
df["age"].unique()
```

```
array([6.70e+01, 8.00e+01, 4.90e+01, 7.90e+01, 8.10e+01, 7.40e+01,
       6.90e+01, 7.80e+01, 6.10e+01, 5.40e+01, 5.00e+01, 6.40e+01,
       7.50e+01, 6.00e+01, 7.10e+01, 5.20e+01, 8.20e+01, 6.50e+01,
       5.70e+01, 4.20e+01, 4.80e+01, 7.20e+01, 5.80e+01, 7.60e+01,
       3.90e+01, 7.70e+01, 6.30e+01, 7.30e+01, 5.60e+01, 4.50e+01,
       7.00e+01, 5.90e+01, 6.60e+01, 4.30e+01, 6.80e+01, 4.70e+01,
       5.30e+01, 3.80e+01, 5.50e+01, 4.60e+01, 3.20e+01, 5.10e+01,
       1.40e+01, 3.00e+00, 8.00e+00, 3.70e+01, 4.00e+01, 3.50e+01,
       2.00e+01, 4.40e+01, 2.50e+01, 2.70e+01, 2.30e+01, 1.70e+01,
       1.30e+01, 4.00e+00, 1.60e+01, 2.20e+01, 3.00e+01, 2.90e+01,
       1.10e+01, 2.10e+01, 1.80e+01, 3.30e+01, 2.40e+01, 3.60e+01,
       6.40e-01, 3.40e+01, 4.10e+01, 8.80e-01, 5.00e+00, 2.60e+01,
       3.10e+01, 7.00e+00, 1.20e+01, 6.20e+01, 2.00e+00, 9.00e+00,
       1.50e+01, 2.80e+01, 1.00e+01, 1.80e+00, 3.20e-01, 1.08e+00,
       1.90e+01, 6.00e+00, 1.16e+00, 1.00e+00, 1.40e+00, 1.72e+00,
       2.40e-01, 1.64e+00, 1.56e+00, 7.20e-01, 1.88e+00, 1.24e+00,
       8.00e-01, 4.00e-01, 8.00e-02, 1.48e+00, 5.60e-01, 1.32e+00,
       1.60e-01, 4.80e-01])
```

```
[ ] df["heart_disease"].unique()
```

```
array([1, 0])
```

```
[ ] df["ever_married"].unique()
```

```
array(['Yes', 'No'], dtype=object)
```

```
[ ] df["Residence_type"].unique()
```

```
array(['Urban', 'Rural'], dtype=object)
```

```
[ ] df["smoking_status"].unique()
```

```
array(['formerly smoked', 'never smoked', 'smokes', 'Unknown'],
      dtype=object)
```

```
[ ] df["stroke"].unique()
```

```
array([1, 0])
```

```
[ ] df.corr()
```

```
<ipython-input-17-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated.
  df.corr()
```

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
age	1.000000	0.279414	0.270344	0.238526	0.376082	0.257103
hypertension	0.279414	1.000000	0.104689	0.164660	0.162426	0.119282
heart_disease	0.270344	0.104689	1.000000	0.172132	0.062912	0.131392
avg_glucose_level	0.238526	0.164660	0.172132	1.000000	0.190844	0.126957
bmi	0.376082	0.162426	0.062912	0.190844	1.000000	0.055880
stroke	0.257103	0.119282	0.131392	0.126957	0.055880	1.000000

```
[ ] df
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly smoked	1
1	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never smoked	1
2	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	smokes	1
3	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never smoked	1
4	Male	81.0	0	0	Yes	Private	Urban	186.21	29.000000	formerly smoked	1
...	...	...	...	...	...	...	...	...	...	...	...
5177	Male	41.0	0	0	No	Private	Rural	70.15	29.756631	formerly smoked	0
5178	Male	40.0	0	0	Yes	Private	Urban	191.15	31.124172	smokes	0
5179	Female	45.0	1	0	Yes	Govt_job	Rural	95.02	31.798304	smokes	0
5180	Male	40.0	0	0	Yes	Private	Rural	83.94	29.951301	smokes	0
5181	Female	80.0	1	0	Yes	Private	Urban	83.75	29.097421	never smoked	0

5182 rows × 11 columns

```
[ ]  df["age"].value_counts()
```

```
78.00    111
57.00     94
79.00     92
52.00     88
54.00     88
         ...
1.40       3
1.16       3
0.40       2
0.08       2
0.16       1
Name: age, Length: 104, dtype: int64
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5182 entries, 0 to 5181
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             5182 non-null   object 
 1   age                5182 non-null   float64
 2   hypertension       5182 non-null   int64  
 3   heart_disease      5182 non-null   int64  
 4   ever_married       5182 non-null   object 
 5   work_type          5182 non-null   object 
 6   Residence_type     5182 non-null   object 
 7   avg_glucose_level  5182 non-null   float64
 8   bmi                5182 non-null   float64
 9   smoking_status     5182 non-null   object 
 10  stroke             5182 non-null   int64  
dtypes: float64(3), int64(3), object(5)
memory usage: 445.5+ KB
```

```
[ ]  from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()
```

```
[ ]  objList = ['gender','ever_married','work_type','Residence_type','smoking_status']
     for feat in objList:
         df[feat] = le.fit_transform(df[feat].astype(str))
     print (df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5182 entries, 0 to 5181
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   gender             5182 non-null   int64
 1   age                5182 non-null   float64
 2   hypertension       5182 non-null   int64
 3   heart_disease      5182 non-null   int64
 4   ever_married       5182 non-null   int64
 5   work_type          5182 non-null   int64
 6   Residence_type     5182 non-null   int64
 7   avg_glucose_level  5182 non-null   float64
 8   bmi                5182 non-null   float64
 9   smoking_status     5182 non-null   int64
 10  stroke             5182 non-null   int64
dtypes: float64(3), int64(8)
memory usage: 445.5 KB
None
```

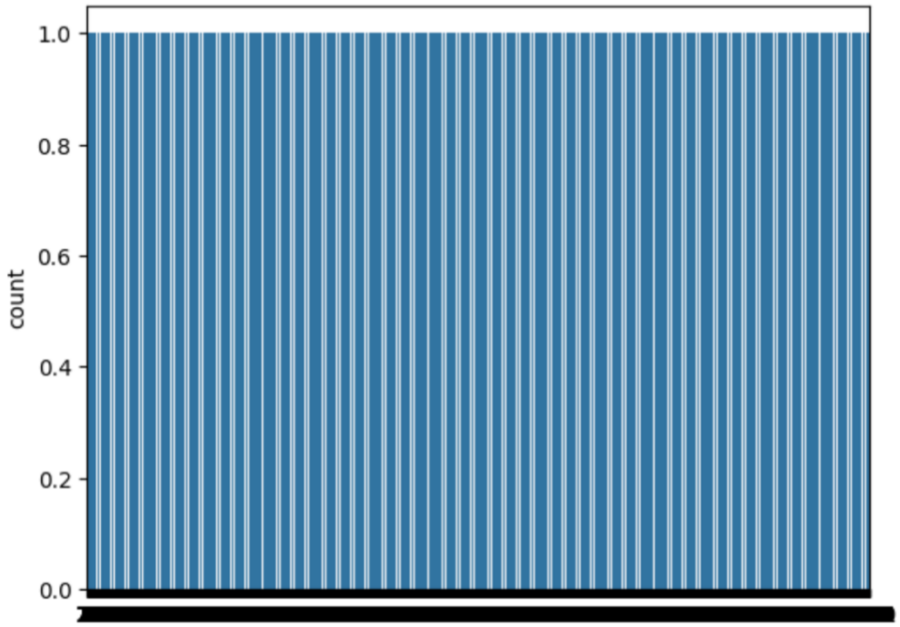
```
[ ] df
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	1	67.0	0	1	1	1	1	228.69	36.600000	1	1
1	1	80.0	0	1	1	1	0	105.92	32.500000	2	1
2	0	49.0	0	0	1	1	1	171.23	34.400000	3	1
3	0	79.0	1	0	1	2	0	174.12	24.000000	2	1
4	1	81.0	0	0	1	1	1	186.21	29.000000	1	1
...	...	...	...	...	...	...	...	...	...	...	...
5177	1	41.0	0	0	0	1	0	70.15	29.756631	1	0
5178	1	40.0	0	0	1	1	1	191.15	31.124172	3	0
5179	0	45.0	1	0	1	0	0	95.02	31.798304	3	0
5180	1	40.0	0	0	1	1	0	83.94	29.951301	3	0
5181	0	80.0	1	0	1	1	1	83.75	29.097421	2	0

5182 rows × 11 columns

```
[ ]  y = df["stroke"]
     sns.countplot(y)
     target_temp = df.stroke.value_counts()

     print(target_temp)
```

```
0    4894
1     288
Name: stroke, dtype: int64
```



```
[ ] df["stroke"].value_counts()
```

```
0    4894
1     288
Name: stroke, dtype: int64
```

```
[ ] print("Percentage of patience without stroke problems: "+str(round(target_temp[0]*100/299,2)))
    print("Percentage of patience with stroke problem : "+str(round(target_temp[1]*100/299,2)))
```

```
Percentage of patience without stroke problems: 1636.79
Percentage of patience with stroke problem : 96.32
```

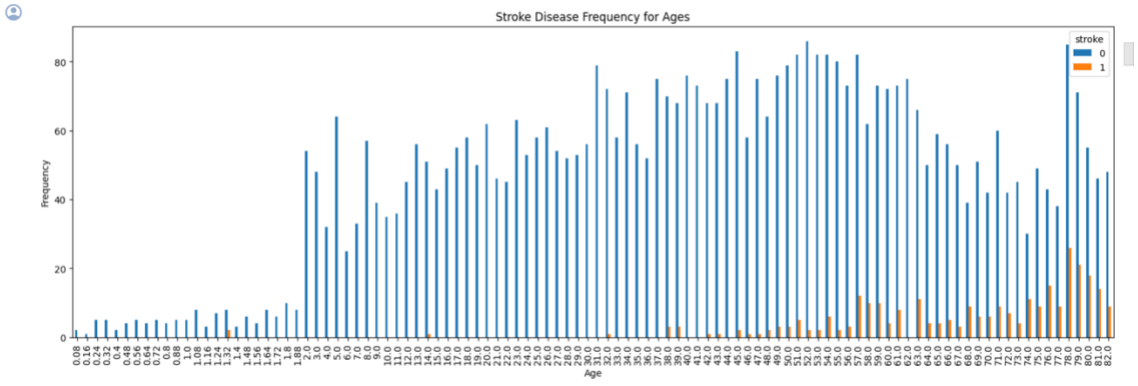
```
[ ] df["gender"].unique()
```

```
array([1, 0])
```

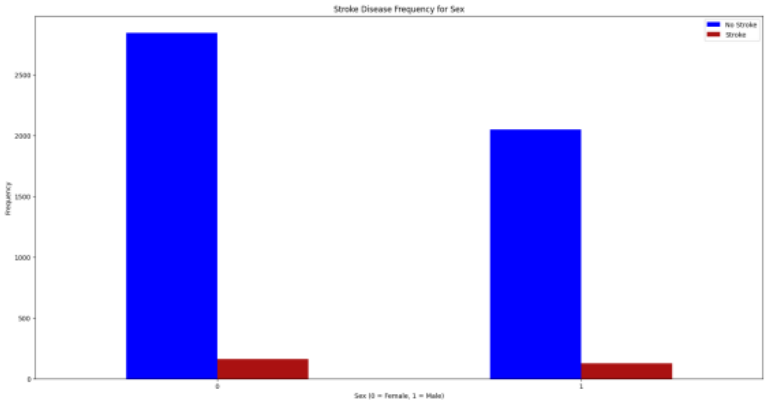
```
[ ] countFemale = len(df[df.gender == 0])
    countMale = len(df[df.gender == 1])
    print("Percentage of Female Patients:{:.2f}%".format((countFemale)/(len(df.gender))*100))
    print("Percentage of Male Patients:{:.2f}%".format((countMale)/(len(df.gender))*100))
```

```
Percentage of Female Patients:57.97%
Percentage of Male Patients:42.03%
```

```
[ ] pd.crosstab(df.age,df.stroke).plot(kind="bar",figsize=(20,6))
    plt.title('Stroke Disease Frequency for Ages')
    plt.xlabel('Age')
    plt.ylabel('Frequency')
    plt.savefig('StrokeDiseaseAndAges.png')
    plt.show()
```



```
[ ] pd.crosstab(df.gender,df.stroke).plot(kind="bar",figsize=(20,10),color=['blue','#AA1111','green','yellow','black' ])
    plt.title('Stroke Disease Frequency for Sex')
    plt.xlabel('Sex (0 = Female, 1 = Male)')
    plt.xticks(rotation=0)
    plt.legend(["No Stroke", "Stroke"])
    plt.ylabel('Frequency')
    plt.show()
```




```
[ ] df["Residence_type"].unique()
```

```
array([1, 0])
```

```
[ ] predictors = df.drop("stroke",axis=1)
    target = df["stroke"]
```

```
[ ] target.value_counts()
```

```
0    4894
1     288
Name: stroke, dtype: int64
```

```
[ ] from imblearn.over_sampling import SMOTE
```

```
[ ] from imblearn.over_sampling import SMOTE
```

```
[ ] print("Before OverSampling, counts of label '0': {}".format(sum(target == 0)))
    print("Before OverSampling, counts of label '1': {}".format(sum(target == 1)))


    # import SMOTE module from imblearn library
    # pip install imblearn (if you don't have imblearn in your system)

    sm = SMOTE(random_state = 42)
    predictors_res, target_res = sm.fit_resample(predictors,target.ravel())

    print('After OverSampling, the shape of train_X: {}'.format(predictors_res.shape))
    print('After OverSampling, the shape of train_y: {} \n'.format(target_res.shape))

    print("After OverSampling, counts of label '0': {}".format(sum(target_res == 0)))
    print("After OverSampling, counts of label '1': {}".format(sum(target_res == 1)))
```

```
Before OverSampling, counts of label '0': 4894
Before OverSampling, counts of label '1': 288
After OverSampling, the shape of train_X: (9788, 10)
After OverSampling, the shape of train_y: (9788,)

After OverSampling, counts of label '0': 4894
After OverSampling, counts of label '1': 4894
```

```
[ ] from sklearn.model_selection import GridSearchCV, train_test_split, cross_val_score
    X_train, X_test, y_train, y_test = train_test_split(predictors_res,target_res,stratify=target_res,random_state = 42)
```

```
[ ] X_test.to_csv(r"C:\Users\ST-0008\Desktop\test.csv")
```

```
[ ] target_res.shape
```

```
(9788,)
```

```
[ ] predictors_res.shape
```

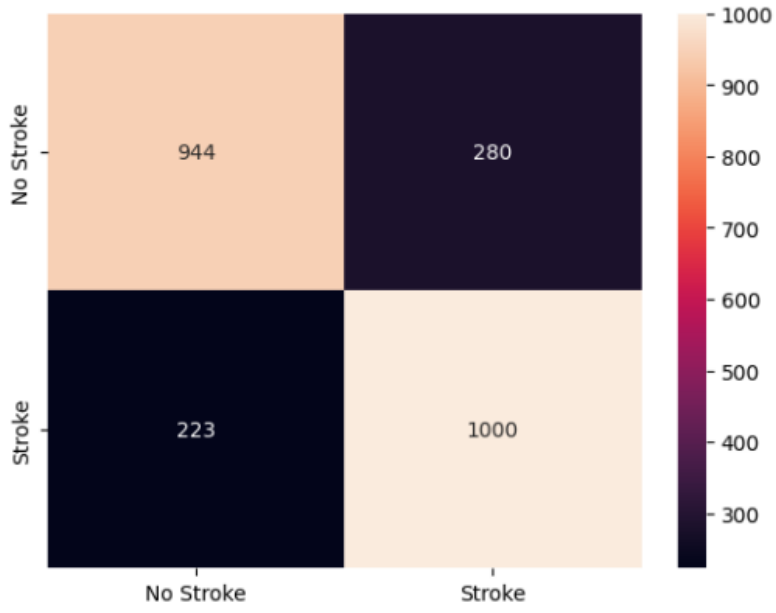
```
(9788, 10)
```

```
[ ] print(X_train.shape,X_test.shape,y_train.shape,y_test.shape)
```

```
(7341, 10) (2447, 10) (7341,) (2447,)
```

```
[ ] from sklearn.metrics import accuracy_score, confusion_matrix, f1_score
```

```
[ ] from sklearn.linear_model import LogisticRegression
    log_model=LogisticRegression()
    log_model.fit(X_train,y_train)

    import pickle
    filename = r'C:\Users\ST-0008\Desktop\LR_stroke.pkl'
    pickle.dump(log_model, open(filename, 'wb'))


    pred_test1 =log_model.predict(X_test)
    test_accuracy1 = accuracy_score(y_test, pred_test1)
    pred_train = log_model.predict(X_train)
    train_accuracy1 =accuracy_score(y_train,pred_train)

    print("AUC on Test data is " +str(accuracy_score(y_test,pred_test1)))
    print("AUC on Train data is " +str(accuracy_score(y_train,pred_train)))

    print("----------------------------")

    # Code for drawing seaborn heatmaps
    class_names =['No Stroke','Stroke']
    df_heatmap = pd.DataFrame(confusion_matrix(y_test, pred_test1.round()), index=class_names, columns=class_names )
    fig = plt.figure( )
    heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")
```

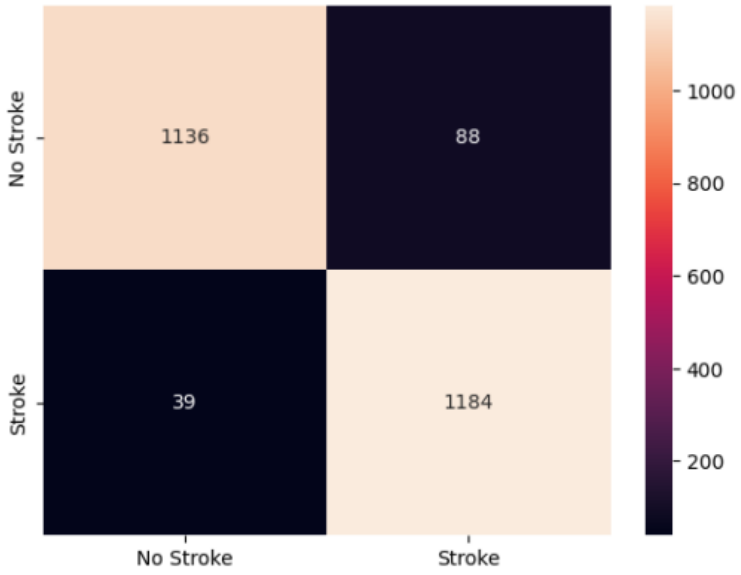


```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import GridSearchCV

dept = [1, 5, 10, 50, 100, 500, 1000]
n_estimators =  [20, 40, 60, 80, 100, 120]

param_grid={'n_estimators':n_estimators , 'max_depth':dept}
RF = RandomForestClassifier()
model = GridSearchCV(RF,param_grid,scoring='accuracy',n_jobs=-1,cv=3)
model.fit(X_train,y_train)
print("optimal n_estimators",model.best_estimator_.n_estimators)
print("optimal max_depth",model.best_estimator_.max_depth)
```

```
optimal n_estimators 100
optimal max_depth 500
```

```
[ ] optimal_n_estimators = model.best_estimator_.n_estimators
    optimal_max_depth = model.best_estimator_.max_depth
```

```
[ ] from sklearn.metrics import accuracy_score

    clf = RandomForestClassifier(max_depth = optimal_max_depth,n_estimators = optimal_n_estimators)
    clf.fit(X_train,y_train)

    import pickle
    filename = r'C:\Users\ST-0008\Desktop\rf_stroke.pkl'
    pickle.dump(clf, open(filename, 'wb'))


    pred_test3 =clf.predict(X_test)
    test_accuracy3 = accuracy_score(y_test, pred_test3)
    pred_train = clf.predict(X_train)
    train_accuracy3 =accuracy_score(y_train,pred_train)

    print("AUC on Test data is " +str(accuracy_score(y_test,pred_test3)))
    print("AUC on Train data is " +str(accuracy_score(y_train,pred_train)))

    print("----------------------------")

    # Code for drawing seaborn heatmaps
    class_names =['No Stroke','Stroke']
    df_heatmap = pd.DataFrame(confusion_matrix(y_test, pred_test3.round()), index=class_names, columns=class_names )
    fig = plt.figure( )
    heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")
```

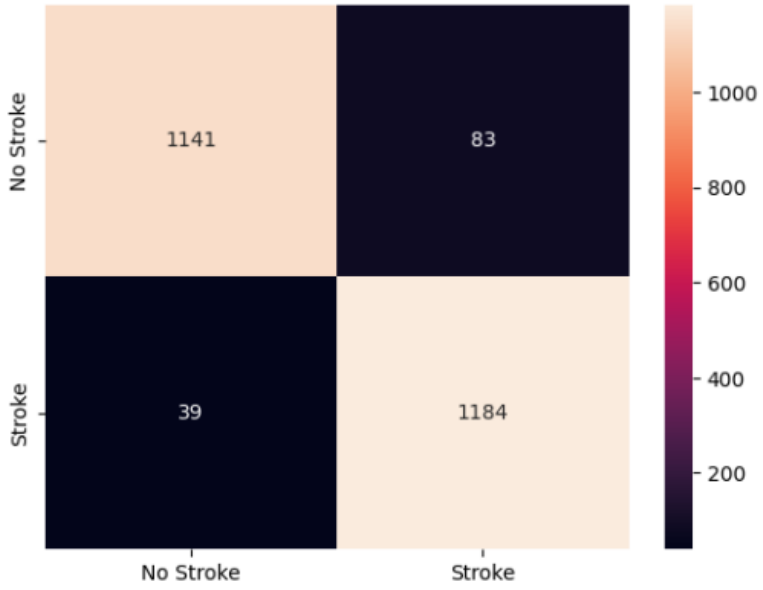



```
[ ]  results
```

	model	Classifier	Train-Accuracy	Test-Accuracy
0	LR	Logisticregression	0.794	0.801
1	RF	RandomForestClassifier	0.999	0.946

```
import xgboost as xgb
#import lightgbm as lgb
from sklearn.metrics import accuracy_score

model = xgb.XGBClassifier(n_estimators=1000, learning_rate=0.04,random_state=1)
model.fit(X_train, y_train, eval_set=[(X_train, y_train), (X_test, y_test)], verbose=100)

import pickle
filename = r'C:\Users\ST-0008\Desktop\X_gb_stroke.pkl'
pickle.dump(model, open(filename, 'wb'))


pred_test4 =model.predict(X_test)
test_accuracy4 = accuracy_score(y_test, pred_test4)
pred_train = model.predict(X_train)
train_accuracy4 =accuracy_score(y_train,pred_train)


print("AUC on Test data is " +str(accuracy_score(y_test,pred_test4)))
print("AUC on Train data is " +str(accuracy_score(y_train,pred_train)))
```

```
print("----------------------------")
# Code for drawing seaborn heatmaps
class_names =['No Stroke','Stroke']
df_heatmap = pd.DataFrame(confusion_matrix(y_test, pred_test4.round()), index=class_names, columns=class_names )
fig = plt.figure( )
heatmap = sns.heatmap(df_heatmap, annot=True, fmt="d")
```




```
from sklearn.metrics import classification_report
report = classification_report(y_test, pred_test4)

# Print the classification report
print("Classification Report:")
print(report)
```

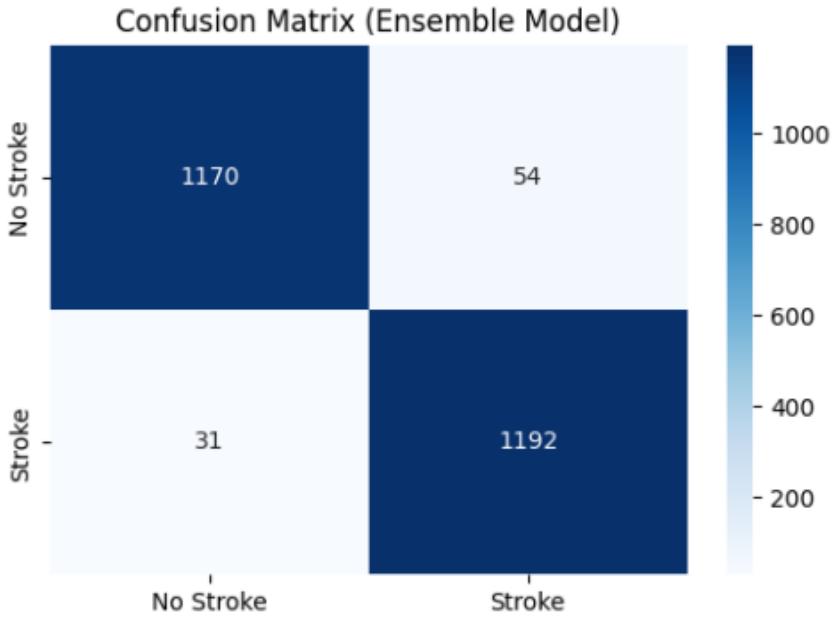
```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.93      0.95      1224
           1       0.93      0.97      0.95      1223

    accuracy                           0.95      2447
   macro avg       0.95      0.95      0.95      2447
weighted avg       0.95      0.95      0.95      2447
```

```
import pandas as pd
import xgboost as xgb
import lightgbm as lgb
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from imblearn.over_sampling import SMOTE
import seaborn as sns
import matplotlib.pyplot as plt
import pickle

# Handling class imbalance using SMOTE
smote = SMOTE(random_state=1)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# XGBoost model
xgb_model = xgb.XGBClassifier(random_state=1)
param_grid_xgb = {
    'n_estimators': [100, 500, 1000],
    'learning_rate': [0.01, 0.1, 0.2],
}
```

```
grid_search_lgb = GridSearchCV(lgb_model, param_grid_lgb, cv=5, scoring='accuracy', n_jobs=-1)
grid_search_lgb.fit(X_train_resampled, y_train_resampled)
best_lgb_model = grid_search_lgb.best_estimator_

# Save the best LightGBM model
lgb_filename = 'best_lgb_model.pkl'
pickle.dump(best_lgb_model, open(lgb_filename, 'wb'))

# Ensemble using VotingClassifier
ensemble_model = VotingClassifier(estimators=[
    ('xgb', best_xgb_model),
    ('lgb', best_lgb_model)
], voting='soft')

ensemble_model.fit(X_train_resampled, y_train_resampled)

# Make predictions
pred_ensemble = ensemble_model.predict(X_test)

# Evaluate the ensemble model
ensemble_accuracy = accuracy_score(y_test, pred_ensemble)
print("Ensemble Model Accuracy: {:.2f}%".format(ensemble_accuracy * 100))

# Confusion matrix heatmap
class_names = ['No Stroke', 'Stroke']
conf_matrix = confusion_matrix(y_test, pred_ensemble)
df_heatmap = pd.DataFrame(conf_matrix, index=class_names, columns=class_names)
```

```
    plt.figure(figsize=(6, 4))
    plt.title('Confusion Matrix (Ensemble Model)')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    sns.heatmap(df_heatmap, annot=True, fmt="d", cmap="Blues")
    plt.show()
```

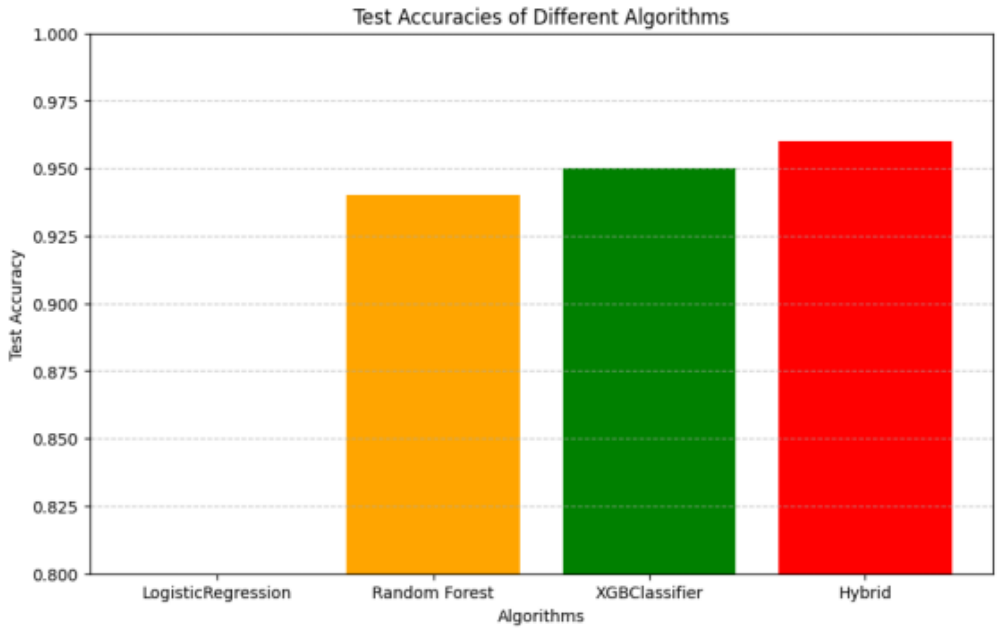
```
features: 10
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.500000 -> initscore=0.000000
Ensemble Model Accuracy: 96.53%
```



```python
from sklearn.metrics import classification_report

# Evaluate the ensemble model
ensemble_accuracy = accuracy_score(y_test, pred_ensemble)
print("Ensemble Model Accuracy: {:.2f}%".format(ensemble_accuracy * 100))

# Classification report
print("Classification Report:")
print(classification_report(y_test, pred_ensemble, target_names=['No Stroke', 'Stroke']))

# Confusion matrix heatmap
class_names = ['No Stroke', 'Stroke']
conf_matrix = confusion_matrix(y_test, pred_ensemble)
df_heatmap = pd.DataFrame(conf_matrix, index=class_names, columns=class_names)

plt.figure(figsize=(6, 4))
plt.title('Confusion Matrix (Ensemble Model)')
plt.xlabel('Predicted')
plt.ylabel('Actual')
sns.heatmap(df_heatmap, annot=True, fmt="d", cmap="Blues")
plt.show()
```

```
Ensemble Model Accuracy: 96.53%
Classification Report:
              precision    recall  f1-score   support

   No Stroke       0.97      0.96      0.96      1224
      Stroke       0.96      0.97      0.97      1223

    accuracy                           0.97      2447
   macro avg       0.97      0.97      0.97      2447
weighted avg       0.97      0.97      0.97      2447
```

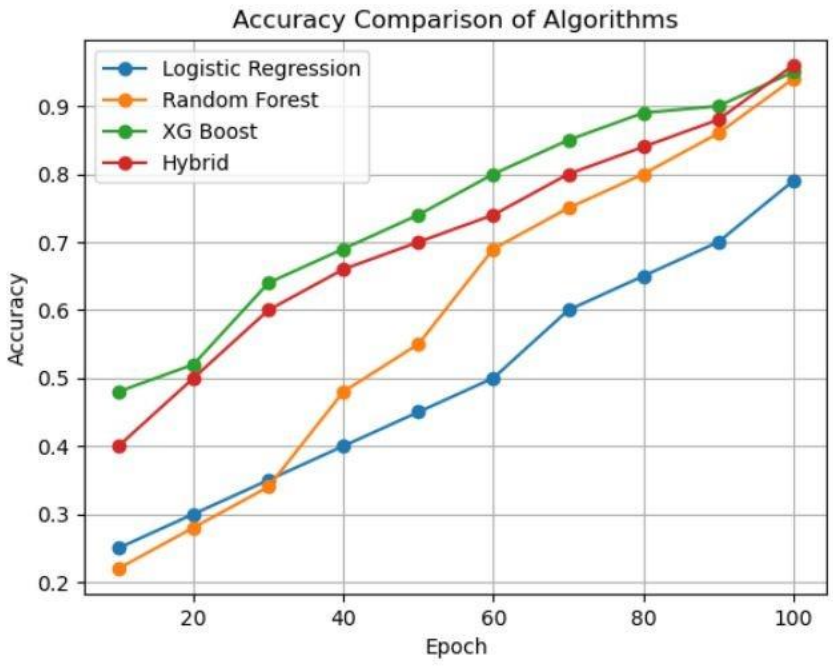
```
[ ]  MODEL COMPARISION

     import matplotlib.pyplot as plt

     # Test accuracies for each algorithm
     algorithms = ['LogisticRegression', 'Random Forest', 'XGBClassifier', 'Hybrid']
     test_accuracies = [0.79, 0.94, 0.95, 0.96]  # Replace these values with your test accuracies

     # Plotting
     plt.figure(figsize=(10, 6))
     plt.bar(algorithms, test_accuracies, color=['blue', 'orange', 'green', 'red'])
     plt.title('Test Accuracies of Different Algorithms')
     plt.xlabel('Algorithms')
     plt.ylabel('Test Accuracy')
     plt.ylim(0.8, 1.0)  # Adjust ylim if needed
     plt.grid(axis='y', linestyle='--', alpha=0.7)
     plt.show()
```



```
[ ]  from sklearn.ensemble import VotingClassifier

     def ensemble_model():
         # Define the base models
         base_models = [
             ('xgb', best_xgb_model),
             ('lgb', best_lgb_model),
         ]

         # Create the ensemble model
         ensemble_model = VotingClassifier(estimators=base_models, voting='soft')

         return ensemble_model
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Assuming you have X_train, X_test, y_train, and y_test defined for each algorithm
# Make sure to replace these with your actual data

# Initialize models
models = {
    "Logistic Regression": LogisticRegression(),
    "Random Forest": RandomForestClassifier(max_depth = 1000,n_estimators = 80),
    "XGB": xgb.XGBClassifier(n_estimators=1000, learning_rate=0.04,random_state=1),
    "HYBRID": ensemble_model()
}

# Number of epochs for training
epochs = 10

# Dictionary to store test accuracies for each algorithm
test_accuracies = {model_name: [] for model_name in models}

# Training loop
for epoch in range(epochs):
    for model_name, model in models.items():
        model.fit(X_train, y_train)
        pred_test = model.predict(X_test)
        test_accuracy = accuracy_score(y_test, pred_test)
        test_accuracies[model_name].append(test_accuracy)
        print(f"Epoch {epoch+1}, Model: {model_name}, Test Accuracy: {test_accuracy}")
```

```
# Plotting the accuracy curves
plt.figure(figsize=(10, 6))
for model_name, accuracies in test_accuracies.items():
    plt.plot(range(1, epochs + 1), accuracies, label=model_name)

plt.title('Test Accuracy Curves of Different Algorithms')
plt.xlabel('Epochs')
plt.ylabel('Test Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```



REFERENCES

[1] V. Sapra, L. Sapra, A. Vishnoi and P. Srivastava, "Identification of Brain Stroke using Boosted Random Forest," 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM), Dehradun, India, 2022, pp. 1-5, doi: 10.1109/ICACCM56405.2022.10009527.

[2] N. Felice, J. Johan, J. Natthannael, M. B. Gozal, C. Jovannie and M. S. Anggreainy, "Brain Stroke Prediction Using Random Forest Method with Tuning Parameter," 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2023, pp. 81-85, doi: 10.1109/AiDAS60501.2023.10284685.

[3] J. K and N. K. Prakash, "Prediction of Brain Strokeusing Machine Learning with Relief Algorithm," 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022, pp. 1255-1260, doi: 10.1109/ICECAA55415.2022.9936142.

[4] R. Kumari and H. Garg, "Interpretation and Analysis of Machine Learning Models for Brain Stroke Prediction," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-6, doi: 10.1109/ISCON57294.2023.10112188.

[5] N. T. Singh, A. Swetapadma and P. K. Pattnaik, "A Comparative Study of Quantum Machine Learning Enhanced SVM and Classical SVM for Brain Stroke Prediction," 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2023, pp. 1-4, doi: 10.1109/ICCCI56745.2023.10128293.

[6] P. Bathla and R. Kumar, "Artificial Intelligence based Model for Brain Stroke Prediction," 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2022, pp. 1-6, doi: 10.1109/IATMSI56455.2022.10119373.

[7] I. Almubark, "Brain Stroke Prediction Using Machine Learning Techniques," 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 2023, pp. 6104-6108, doi: 10.1109/BigData59044.2023.10386474.