Research Paper

Early Brain Stroke Detection Using Artificial Intelligence

RESEARCH SUPERVISOR

Dr. Md. Ashikur Rahman Khan
Professor & Chairman
Department of ICE, NSTU

SUBMITTED BY

Md. Khalilur Rahman
Roll No: ASH2111MSc118M
Session: 2020-2021
Year II, Term I
Department of ICE, NSTU

DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING

NOAKHLAI SCIENCE AND TECHNOLOGY UNIVERSITY

APRIL 2024

Early Brain Stroke Detection Using Artificial Intelligence

MD. Khalilur Rahman

This research paper submitted to the Department of Information and Communication Engineering, Noakhali Science and Technology University, in partial fulfillment of the requirements for the degree of M.Sc. (Eng.) in Information and Communication Engineering.

DEPARTMENT OF INFORMATION AND COMMUNICATION ENGINEERING

NOAKHLAI SCIENCE AND TECHNOLOGY UNIVERSITY

APRIL 2024

DECLARATION

I hereby declare that this research paper has not been submitted elsewhere for the requirement of any kind of Degree, Diploma or Publication.

Md. Khalilur Rahman
Roll No: ASH2111MSc118M
Session: 2020-2021
Year II, Term I
Department of Information and Communication Engineering
Noakhali Science & Technology University
Noakhali - 3814.

ACCEPTANCE

This research paper is submitted to the Department of Information & Communication Engineering, Noakhali Science & Technology University, Noakhali - 3814, Noakhali in partial fulfillment of the requirements for having the M.Sc. (Eng.) degree in ICE.

Dr. Md. Ashikur Rahman Khan
Professor & Chairman
Department of Information and Communication Engineering
Noakhali Science & Technology University
Noakhali - 3814.

ABSTRACT

Due to the current environmental conditions and human lifestyle choices, people are impacted by a wide range of diseases nowadays. If such diseases are to be prevented from reaching their terminal phases, early detection and prediction are essential. An enormous financial burden is placed on individuals who suffer from stroke, a cerebrovascular disorder that is one of the main causes of mortality. One of the main risk factors for stroke is health-related behavior, which is gaining importance as a prevention strategy. Many machine learning algorithms that incorporate lifestyle characteristics as predictors to autonomously diagnose stroke have been used to predict the risk of stroke.

This work uses five supervised machine learning classifiers to predict strokes: K-Nearest Neighbor Algorithm, Decision Tree, Random Forest, Support Vector Machine, and Naïve Bayes. The aforementioned classifiers are trained on the dataset, which consists of 5110 items with 10 attributes, and their performance is assessed using the confusion matrix. The dataset is preprocessed to make it acceptable for prediction. The RF method surpassed all other algorithms in the employed dataset for predicting strokes based on many physiological characteristics, with an accuracy of 95.8%. In contrast to an individual's medical history and level of physical activity, machine learning algorithms may be more useful for the clinical estimation of stroke.

Stroke patients need continuous critical care in addition to all of these diagnoses, which can be provided by an interdisciplinary team.

Keyword: Artificial Intelligence, Confusion Matrix, Random Forest Classifier, Stroke

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 01

INTRODUCTION

1.1 Introduction

A brain attack, or stroke, is a medical condition caused by a broken blood vessel or a blockage in the blood supply to a particular part of the brain. Human body functions are controlled by the brain, which also stores memories and produces our thoughts, emotions, and spoken language. The brain also controls a number of other body functions, such as digestion and breathing. Our brains cannot function properly without oxygen. Rich in oxygen, blood from the arteries nourishes every part of our brain. A stroke is caused by brain cells that are deprived of oxygen and start to die minutes after there is a stoppage in blood flow. There could be irreversible brain damage, death, or long-term incapacity from this.

Stroke is the leading cause of disability and the second highest cause of mortality worldwide. A person's lifetime risk of suffering a stroke has increased by 50% in the last 17 years, with 1 in 4 people being at risk, according to the Global Stroke Factsheet, which was published in 2022. Between 1990 and 2019, there was an increase in stroke incidence, stroke fatalities, prevalence, and Disability Adjusted Life Years (DALY) of 70%, 43%, 102%, and 143%, respectively. The most important finding is that 86% of stroke-related deaths and 89% of DALYs occur in lower- and lower-middle-income countries worldwide. These countries also have the highest burden of stroke. This disproportionate financial stress experienced by lower- and lower-middle-income countries is posing a new challenge for low-income families. (2022, World Stroke Day).

Stroke is the most common cause of adult disability in Europe, and it can affect many aspects of everyday life. By 2040, there will likely be 12 million stroke victims in Europe, up from the present nine million. Families, healthcare providers, social services, and the medical system would all be even more burdened by this. The inaugural European Life After Stroke Forum was held in person

in Barcelona on March 10, 2023. Its objective is to bring attention to this neglected area of the stroke treatment pathway in order to raise awareness of stroke care and support, as well as to elevate it to the same level as acute care and rehabilitation. A blocked artery or a blood vessel that leaks or bursts is the main cause of stroke. In certain people, there may only be a transient decrease in blood flow to the brain; there may be no long-term consequences.

A stroke can happen to anyone, at any age, however certain people are more susceptible than others. Strokes are more common in later life, occurring in those over 65 in about two thirds of cases. Heart disease, diabetes, high blood pressure, smoking, binge drinking, and various other health problems are among the conditions that increase the risk of stroke (Centers for Disease Control and Prevention, 2023). There are three types of stroke, some more dangerous than others and more likely to result in permanent disability. These comprise transient ischemic attacks (mini-strokes), hemorrhagic strokes, and ischemic strokes. Holek (n.d.)

An ischemic stroke, which is the most common kind in the general population, is typically brought on by a blood clot that obstructs or blocks a blood vessel in the brain. This causes a disruption in the blood stream, which keeps oxygenated blood from getting to specific parts of the brain. Nerve cells malfunction and eventually die when there is a disruption in their blood and oxygen supply. These injured brain regions are the source of the different difficulties a stroke survivor may encounter. A burst or released blood clot in a brain artery causes a hemorrhagic stroke. Blood leakage causes the brain to enlarge or hemorrhage, puts excessive strain on brain cells, and sharply raises intracranial pressure. Because they often cause less damage and danger than other forms of strokes, transient ischemic attacks are also referred to as "mini-strokes". In this case, the brain's blood supply is blocked for a maximum of five minutes. Nonetheless, it is seen as a warning sign for more strokes and must not be disregarded.

Brain cell death causes loss of brain function. It's possible that the patient won't be able to carry out activities requiring that particular part of the brain. A stroke, for example, can cause impairments in a person's ability to walk, speak, eat, think, and remember in addition to their emotional regulation and other vital body processes. The patient may struggle to raise both arms and keep them there, have trouble smiling, have his face slump to one side, and speak with a slurred voice. The most typical signs of a stroke are these three. A sudden and intense headache

accompanied by a blinding pain unlike anything previously felt, loss of consciousness, sudden loss or blurring of vision, dizziness, confusion, trouble understanding what others are saying, problems with balance and coordination, difficulty swallowing (dysphagia), and total paralysis of one side of the body are some additional symptoms that may be experienced.

FIGURE 1 lists a number of the strokes.

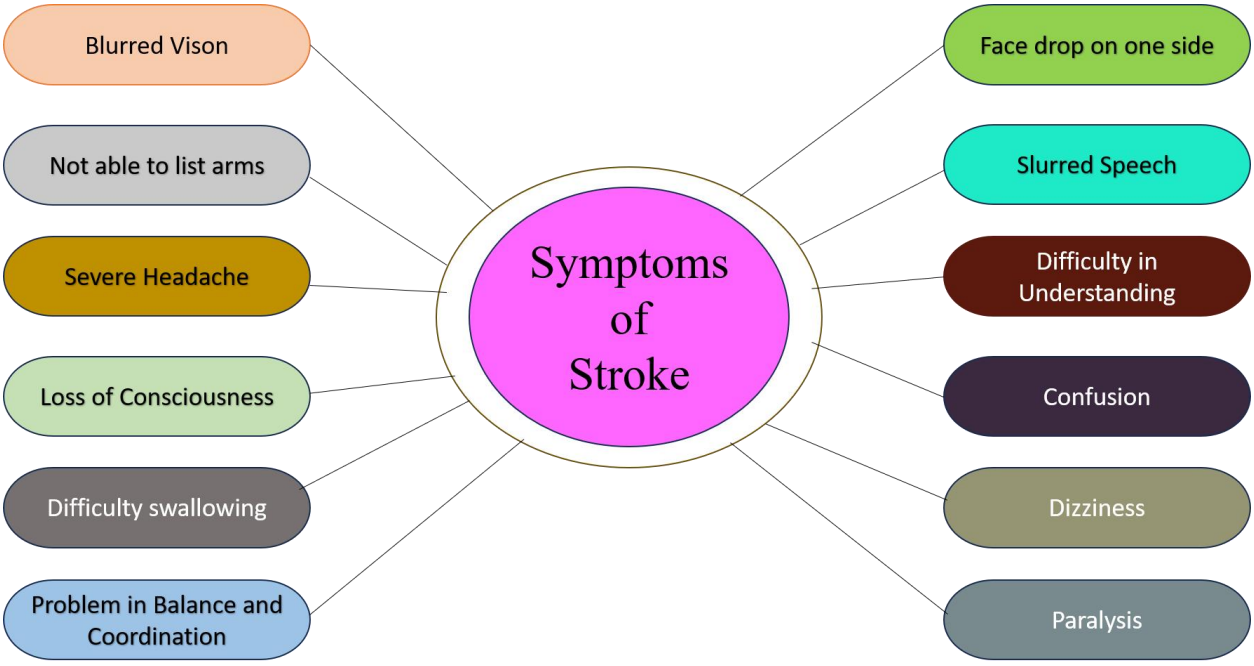

Figure 1. 1: Symptoms of stroke

A stroke can happen to anyone at any time. When the blood supply to a portion of the brain is interrupted, a stroke occurs, which is a serious medical illness that can be fatal (NHS 2022). But some people are more prone to having a stroke than others. Certain stroke risk factors are modifiable or controllable, while others are not. Obesity, high blood lipid and cholesterol levels, cardiac disease, diabetes, smoking, birth control pills (oral contraceptives), high blood pressure, and a history of transient ischemic events Risk factors for stroke that are modifiable include inactivity, obesity, excessive alcohol consumption, irregular pulse, structural abnormalities of the heart, older age, and family history of stroke. The medical emergency team at the hospital will try to determine what kind of stroke the patient is experiencing. A number of blood tests may be performed on the patient, including ones to assess the patient's blood clotting speed, blood sugar levels, and infection status. Using a succession of X-rays, a computed tomography (CT) scan

creates an exact image of the brain. It might show signs of a malignancy, ischemic stroke, cerebral haemorrhage, or other illnesses. Magnetic resonance imaging (MRI) uses powerful radio waves and magnets to produce a precise picture of the brain. It can detect brain tissue damage associated with ischemic stroke and cerebral haemorrhage. To acquire accurate images of the carotid arteries' inside, utilize carotid ultrasonography. Both blood flow and the buildup of fatty deposits (plaques) in the carotid arteries are shown by this test. The exact images of the heart produced by an echocardiography aid in the identification of the source of cardiac clots. The movement of blood clots from the heart to the brain may have caused strokes. FIGURE 2 lists the most prevalent conventional stroke diagnosis.

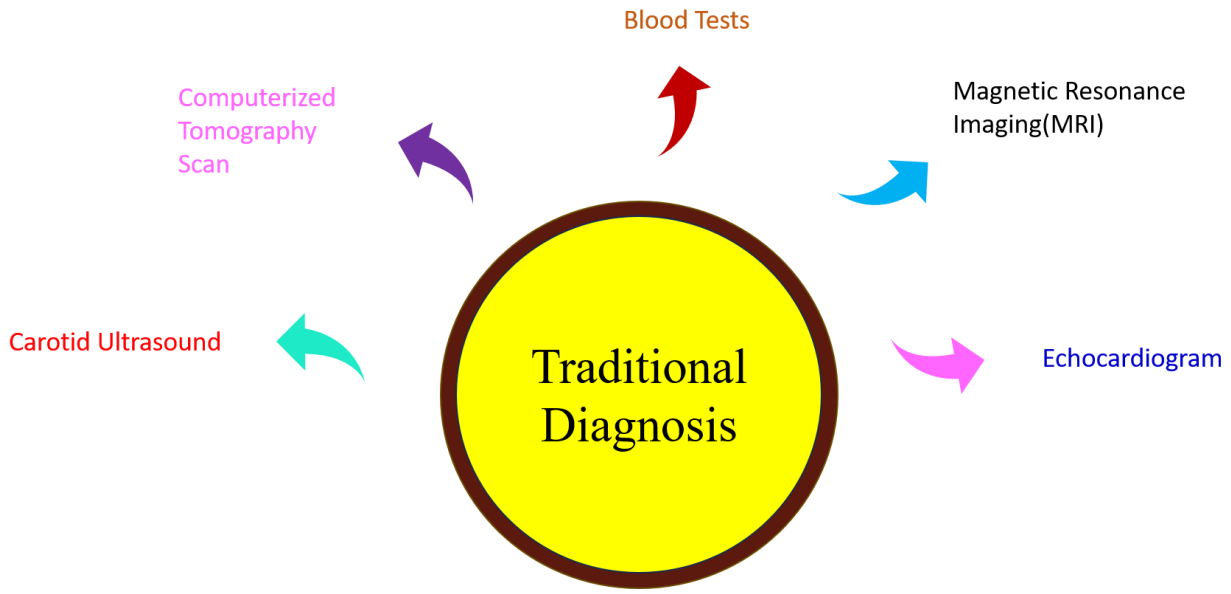

Figure 1. 2: Traditional Diagnosis

1.2 Problem Statement

The problem statement centers on the requirement for early diagnosis of brain strokes utilizing a mix of genetic engineering and artificial intelligence (AI). A brain stroke, often referred to as a cerebrovascular accident, happens when blood flow to the brain is disrupted, which can have serious repercussions and the loss of brain cells. A stroke's early identification is essential for prompt medical attention and better patient outcomes.

- **Limited methods of detection:** Magnetic resonance imaging (MRI) and computed tomography (CT) scans, both of which are used to diagnose strokes, visualize structural abnormalities in the brain. Particularly in the early phases of stroke development, these procedures could not always produce reliable results.

- **Delayed diagnosis and treatment:** Strokes are frequently detected later than necessary due to the limits of current detection techniques, which leads to delayed treatment. This delay may have an adverse effect on the patient's recovery and raise the possibility of long-term disability or even death.

- **Biomarkers and genetic predisposition:** Genetic factors may increase a person's risk of developing a stroke. Finding particular genetic markers linked to a higher risk of stroke could help with early detection and allow for preventative actions. However, the accuracy and dependability of existing methods in genetic testing are sometimes lacking.

- **Complex data Analyzing:** medical data is necessary for brain stroke identification. This analysis includes looking at a variety of clinical characteristics, patient history, genetic data, and medical imaging. It may be difficult for traditional analysis techniques to efficiently handle and interpret this data in order to make precise predictions.

- **Lack of predictive models:** It is difficult to create reliable predictive models that can combine genetic engineering data with AI algorithms to detect early indicators of stroke. Extensive investigation, data gathering, and validation are necessary for the development of an accurate and trustworthy predictive model.

1.3 Research Gap

- **Limited application of genetic algorithms in stroke detection:** Although genetic algorithms have been widely applied to a variety of optimization issues, their use specifically in the early diagnosis of brain strokes is very limited. In order to maximize the detection accuracy and effectiveness of brain strokes, it is necessary to investigate and assess the efficacy of incorporating genetic algorithms into AI models.

- **Lack of comprehensive evaluation and comparison:** Although there are numerous AI-based systems for stroke detection, there may not be enough thorough assessment and comparison studies that compare the effectiveness of these systems with one another and with tried-and-true techniques. The proposed method needs to be thoroughly compared to existing approaches, such as conventional image processing algorithms and other AI-based techniques, in order to ascertain its advantages and disadvantages.

- **Insufficient diversity in training datasets:** Effective training of AI models requires access to a wide variety of representative datasets. It's possible that there aren't enough publicly accessible databases with a wide variety of stroke cases, including various demographics, stroke kinds, and imaging modalities. By gathering and curating larger, more varied datasets specifically for early brain stroke diagnosis utilizing AI and genetic algorithms, this research gap could be filled.

- **Interpretability and explainability of AI models:** Deep neural networks in particular demonstrate a "black-box" aspect that makes it difficult to comprehend the underlying assumptions that underlie their predictions. By creating methods to clarify the decision-making process of AI models, it is critical to close the research gap of interpretability and explainability in the context of stroke detection. This would increase user confidence in the system and make it easier for clinical settings to implement it.

- **Real-world deployment and validation:** Developing and validating AI models for early stroke detection is crucial, but there may be a dearth of research on actually putting these systems into use and validating them in the real world. The suggested AI-based system needs to be integrated into ordinary clinical practice, but more study is required to evaluate its viability, dependability, and clinical impact. This research should take into account

aspects like computing efficiency, integration with current medical infrastructure, and user experience.

By filling in these knowledge gaps, scientists can progress the use of AI and genetic algorithms for early brain stroke detection, resulting in more precise and effective diagnosis, better patient outcomes, and a better understanding of stroke pathogenesis.

1.4 Objectives

The objectives of this research work are as follows:

i. To develop an AI-based system for early brain stroke detection.

ii. To explore the integration of genetic algorithms to optimize the performance of the AI model.

iii. To validate the proposed approach through comprehensive experiments and evaluations.

iv. To compare the performance of the developed system with existing methods and techniques.

v. To contribute to medical advancements in the field of stroke detection and diagnosis.

CHAPTER 02

LITERATURE REVIEW

2.1 Related Works

Numerous academics have already employed machine learning-based methods to forecast strokes. The stroke was estimated using the machine learning classification techniques K closest Neighbors, Support Vector Machine, Logistic Regression, Decision Tree, Random Forest, and Random Forest. In addition to the web application, an HTML page and a Flask were created to allow users to enter values for forecasts. This model's accuracy was 82%. One drawback of this work was that the researchers trained the model on textual data rather than actual brain scans (Ojhai & Jha 2023). After an ischemic stroke, deep neural networks, random forests, and logistic regression were used to predict patients' long-term prognosis. Of the 2604 patients in this study, twenty-43 (or 78%) achieved positive outcomes. Compared to the ASTRAL score, the deep neural network model's area under the curve was noticeably larger. Yoon, Kim, and Park, 2019)

When working with uneven data, machine learning algorithms that use data balancing approaches can be useful tools for stroke prediction. Wu and Fang (2020) used random forests (RF), support vector machines (SVM), and regularized logistic regression (RLR) with a 78% accuracy rate. However, as the study's outcome variable was a self-reported stroke, there may be some subjective bias.. It was suggested to develop a weighted voting classifier for stroke prediction using factors including age, body mass index, blood pressure, heart disease, smoking status, blood sugar levels, and prior strokes. The weighted voting classifier fared better than the base classifiers, with an accuracy rate of 97%. (2020, Emon, Meghla, and Keya.)

An alternative approach to creating Explainable Artificial Intelligence (XAI) models involves superimposing reasoning on top of machine learning. People can understand the explanations produced by XAI, and it operates efficiently. The accuracy was 78%, according to the authors' comparison of their findings with Random Forests and the SVM classifier that was deemed to be

the best for the same dataset (Prentzas, Nicolaides, & Kyriacou 2019). A national disease registry was used to apply machine learning techniques for the purpose of predicting the fate of a 90-day stroke. Implemented and assessed were SVM, RF, ANN, and a hybrid artificial neural network (HANN) using a 10-time repeated hold-out with 10-fold cross-validation. Using preadmission and inpatient data, machine learning algorithms yield over 0.94 AUC in both ischemic and hemorrhagic stroke cases. With the addition of data, the prediction accuracy increased to 0.97 AUC. In 2020, Hsu, Johnson, and Lin In another study, a stroke prediction system based on artificial intelligence (AI) and real-time biosignals was created. Real-time EMG (Electromyography) bio-signals from the thighs and calves were gathered, the prominent features identified, and prediction models based on daily activities created. The relevant information was extracted from the raw data using maximum entropy and labeling algorithms. A novel stemming method was used to acquire the processed dataset. By emphasizing the diversity in the dataset, the kind of stroke was identified with a respectable degree of accuracy. ANN, SVM, DT, Logistic Regression (LR), Bagging, and Boosting are the methods employed. Artificial neural networks trained with stochastic gradient descent performed better than the other algorithms, with a 95% classification accuracy and a 14.69 standard deviation. (Gandomir, Soundarapandian, & Govindarajan, 2019).

Using multilayer perceptron (MLP) neural networks, the death rate of 584 stroke patients is estimated. Quick Propagation (QP), Levenberg-Marquardt (LM), Back Propagation (BP), Quasi-Newton (QN), Delta Bar Delta (DBD), and Conjugate Gradient Descent (CGD) are the six different MLP algorithms that were used. According to SÜT and ÇELİK (2012), QP exhibited the highest levels of specificity (81.3%), sensitivity (78.4%), and accuracy (80.7%). A stroke prediction model was created using data mining on a sample of 294 people who were not affected by a stroke and 147 stroke sufferers. The model made use of demographic and medical screening data. Three categorization techniques were used in the study: Naive Bayes, Decision Trees, and Artificial Neural Networks (ANN). The algorithm that worked best was the ANN with integrated data. The overall accuracy rate, AUC, FPR, and FNR for this approach were 0.84, 0.90, 0.12, and 0.25.

(Kansadub & Thammaboosadee, 2019) A new approach to segment the stroke using cranial CT scans is presented to aid in medical diagnosis. A nonparametric estimate approach based on the

Parzen window is used to segment the stroke, and the level set procedure is automatically started within the stroke region. The fuzzy C-means approach is used to compare the output of the level set algorithms with that of the suggested method. The highest accuracy mean was 99.84%, while the lowest standard deviation was 0.08%. Boubouças, Marques, and Braga (2018) A deep neural network (DNN) can predict the motor outcome in the upper and lower limbs six months following a stroke. The model made use of two well-liked machine learning methods: random forest and logistic regression. The upper limb function predicted by the DNN model in this case has an Area Under the Curve (AUC) of 0.906. The AUC of the random forest model was 0.882, whereas the logistic regression model's was 0.874. For the purpose of predicting lower limb function, the AUCs for the DNN, logistic regression, and random forest models were, respectively, 0.822, 0.768, and 0.802. (Choo, Kim, and Chang, 2021).

In a different study, machine learning algorithms were used to predict Early Neurological Deterioration (END) in individuals who had suffered a minor stroke. The accuracy of the results obtained with deep neural networks, boosted neural networks, and logistic regression was 96.6%. (2020, Kang, Cho, and Sung.) To predict functional outcomes in patients who experienced an acute ischemic stroke following endovascular therapy, the predictive accuracy of machine learning algorithms was assessed. The Random Forest (RF), Classification and Regression Tree (CART), C5.0 Decision Tree (DT), Support Vector Machine (SVM), Adaptive Boost Machine (ABM), Least Absolute Shrinkage and Selection Operator (LASSO), and logistic regression models were among the machine learning and regression models that were used to train the dataset. AUC range = 0.65-0.72, MCC range = 0.29-0.42, and external validation range = 0.66-0.71, MCC range = 0.34-0.42 were found to be valid. Similar predictive accuracy was shown by machine learning and logistic regression models. (Menon, Alaka, and Brobbey, 2020.)

In a different study, machine learning was utilized to develop a general methodological pipeline for future research and to determine the most reliable and efficient stroke prediction algorithm in a hypertensive population in China.The Random Under Sampling (RUS)-applied RF model with laboratory variables showed the best model performance. In the current investigation, RUS (sensitivity = 63.9, specificity = 53.7, and mean AUCs = 0.624) showed a more satisfactory outcome, with data balancing approaches improving overall performance with RUS compared to

null models (sensitivity = 0, specificity = 100, and mean AUCs = 0.643). (Cao, Huang, and Chen, 2022).

2.2 Summery:

Table 1: Summary of the related work in prediction of stroke using machine learning techniques.

Paper Reference	Contribution	Methodology	Results
(Ojhai & Jha 2023.)	We developed an HTML website, a Flask, and an online application that let users enter values for forecasts.	K-Nearest Neighbors, Decision Tree Classification, Random Forest Classification, Naive Bayes Classification, Support Vector Machine, and Logistic Regression	82% accuracy
(Yoon et al. 2019)	It was anticipated how ischemic stroke patients would fare in the long run.	Logistic regression, random forests, and deep neural networks.	AUC of Deep Neural Network (DNN) ASTRAL score
(Wu &Fang 2020)	When dealing with uneven data, data balancing techniques are employed to forecast stroke.	Random Forest (RF), Support Vector Machine (SVM), and Regularized Logistic Regression (RLR)	78% accuracy
(Emon et al. 2020)	A weighted voting classifier for predicting stroke was	Voting classifier	97% accuracy

	suggested		
(Prentzas et al. 2019.)	Explainable Artificial Intelligence (XAI) models were proposed as a way to build reasoning on top of machine learning (ML).	The SVM classifier and Random Forests	78% accuracy
(Lin et al. 2020.)	Utilizing machine learning techniques, a national sickness registry was utilized to predict the fate of a stroke after 90 days.	A hybrid artificial neural network (HANN), SVM, RF, and ANN	0.97 AUC
(Yu et al. 2020.)	A stroke prediction system was developed that recognizes stroke using real-time biosignals and artificial intelligence (AI).	Long Short Term Memory (LSTM), Random Forest	98.958% accuracy
(Govindarajan et al. 2019.)	Stochastic gradient descent-trained artificial neural networks outperformed the other techniques.	ANN, SVM, DT, Bagging, Boosting, and Logistic Regression (LR)	95% accuracy

(SÜT & ÇELİK, 2012)	Mortality in stroke patients are predicted using ML algorithms	Six types of MLP Neural Networks: Quick Propagation (QP), Levenberg-Marquardt (LM), Back Propagation (BP), Quasi Newton (QN), Delta Bar Delta (DBD), and Conjugate Gradient Descent (CGD	80.7% accuracy, 78.4% sensitivity and 81.3% specificity
(Thammaboosadee & Kansadub, 2019.)	A stroke prediction model was developed using data mining techniques.	Decision trees, naive bayes, and artificial neural networks (ANNs). Flexible	84% accuracy, 0.90 AUC, 0.12 FPR, and 0.25 FNR
(Rebouças et al. 2018).	A new approach to stroke segmentation with cranial CT scans is presented.	Fuzzy Cmeans, Parzen window	99.84% accuracy

(Kim et al. 2021.)	Six months after the stroke, the motor outcome in the lower and upper limbs was anticipated.	Random Forest, Logistic Regression, and Deep Neural Network (DNN) Model	Upper Limb: Area Under the Curve (AUC) for DNN,LR and RF are 0.906, 0.874 and 0.882 respectively. Lower Limb: The (AUC) for DNN,LR and RF are 0.822, 0.768, and 0.802, respectively
(Sung et al. 2020.)	Patients with acute neurological impairment (END)	Logistic regression, deep neural networks, and boosted trees	96% accuracy
	mild stroke was forecasted using ML		

(Alaka et al. 2020.)	The ability of machine-learning algorithms to forecast functional outcomes following endovascular therapy in individuals suffering from acute ischemic stroke was assessed.	Logistic regression models, Random Forest (RF), Adaptive Boost Machine (ABM), Classification and Regression Tree (CART), C5.0 Decision Tree (DT), Support Vector Machine (SVM), and Least Absolute Shrinkage and Selection Operator (LASSO)	internally (AUC range = [0.65-0.72]; MCC range = [0.29-0.42]) and externally (AUC range = [0.66-0.71]; MCC range = [0.34-0.42])
(Huang et al. 2022.)	Machine learning is used to construct a generic methodological pipeline for future study and the most accurate stroke prediction methodology in a Chinese hypertensive population.	RF was applied via Random Under Sampling (RUS).	RUS: sensitivity = 63.9, specificity = 53.7, and mean AUCs = 0.624

CHAPTER 03

METHODOLOGY

3.1 Methodology

This chapter is divided into five sections: working flowchart, evaluation matrices, machine learning methods, data preprocessing, and data description. This section contains the necessary figures as well as a thorough explanation of the implantation procedure.

3.2 Artificial Intelligence

Artificial intelligence and predictive analytics are two prediction techniques that forecast future events based on past data. Then, using fresh data, the effectiveness of the trained prediction models is assessed by contrasting the estimated results with the actual values. The decision support technique termed a decision tree, which is akin to a tree, is one of the algorithms employed in this study. Decision nodes, leaf nodes, and a root node make up its three components. Using a decision tree approach, a training dataset is divided into branches, which are then further divided into other branches. This pattern is followed until a leaf node is reached. There is no way to split the leaf node further. The traits that were utilized to forecast the outcome are represented by the decision tree's nodes. The leaves are linked by the decision nodes. (Engineering Education Program Section, EngEd), n.d.)

We have utilized a variety of machine learning models in our research that are appropriate for brain stroke detection. Traditional machine learning methods, deep learning architectures (such as

i. Logistic Regression (LR)
ii. Random Forest Classification (RF)
iii. Decision Tree (DT)
iv. Support Vactor Machine (SVM)
v. K-Nearest Neighbours (KNN)

3.2.1 Logistic regression

When attempting to forecast the probability that an instance will belong to a specific class, classification problems are the main application for the supervised machine learning technique known as logistic regression. This statistical method assesses the relationship between a set of binary dependent variables and a collection of independent factors. Because it uses a set of independent factors to predict the categorical dependent variable, it is a powerful tool for decision-making. A discrete or categorical value must be the result of using logistic regression to predict a categorical dependent variable. Instead of providing precise numbers between 0 and 1, it offers probabilistic values in the range of 0 to 1. True or False, 0 or 1, Yes or No, and so forth can be displayed for this. Rather of fitting a regression line to 0 or 1, logistic regression uses a sigmoid-shaped logistic function to predict two maximum values. The logistic function's curve shows that an event's weight determines its likelihood. Logistic regression is a potent machine learning technique that may be used to categorize incoming data and generate probabilities from discrete and continuous datasets.

a. **Terminologies involved in Logistic Regression:** The following are some terminology frequently used in logistic regression:
    i. **Independent variables:** Also referred to as predictor variables or features, these are the variables used to predict the dependent variable. They can be continuous, categorical, or binary.
    ii. **Dependent variable:** Also known as the response variable or target variable, it is the variable being predicted in logistic regression. It is typically categorical or binary.
    iii. **Logistic function:** Also called the sigmoid function, it is used to transform the linear combination of independent variables and coefficients into probabilities between 0 and 1.
    iv. **Logit:** The natural logarithm of the odds ratio. In logistic regression, the relationship between the independent variables and the dependent variable is modelled using the logit function.
    v. **Odd ratio:** the relationship between an event's likelihood of happening and its likelihood of not happening. The change in the chances of the dependent variable with

a one-unit change in the independent variable is represented by the coefficients in logistic regression.

vi. **Log-odds:** The natural logarithm of the chances is called the log-odds, or logit function. The log odds of the dependent variable in logistic regression are represented as a linear mixture of the intercept and the independent factors.

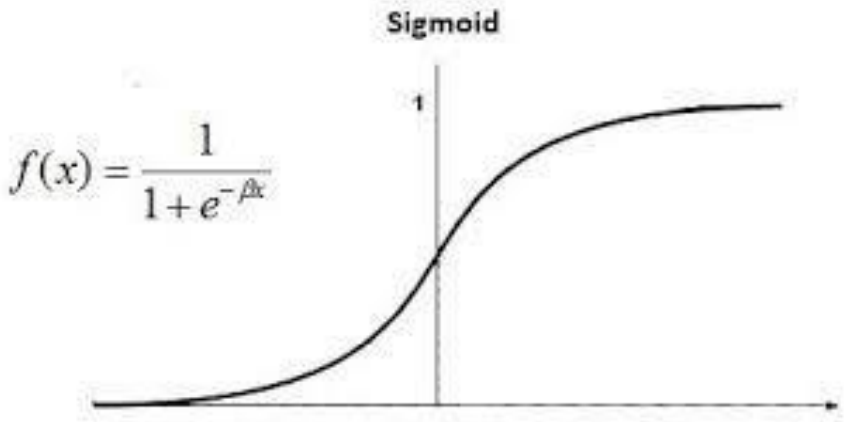

Figure 3. 1: Logistic regression

b) Figure 1: Logistic function (sigmoid function)

i. **Coefficient:** Also known as beta coefficients or weights, these represent the strength and direction of the relationship between each independent variable and the dependent variable.

ii. **Intercept:** The constant term in the logistic regression equation, representing the value of the log-odds when all independent variables are zero.

iii. **Maximum Likelihood Estimation (MLE):** The method used to estimate the parameters of the logistic regression model by maximizing the likelihood function, which measures the probability of observing the data given the model parameters.

iv. **Confusion Matrix:** a table that displays the counts of true positives, true negatives, false positives, and false negatives and is used to assess how well a classification model performs.

b. **Types of Logistic Regression:** Logistic Regression can be classified into 3 categories. These are: -

i. **Binomial:** There are only two conceivable forms of dependent variables in binomial logistic regression, such as 0 or 1, Pass or Fail, etc.

ii. **Multinomial:** One of three unordered categories—such as "cat," "dogs," or "sheep"—may be the dependent variable in a multinomial logistic regression.

iii. **Ordinal:** Ordinal logistic regression allows for the possibility of three or more ordered types of dependent variables, such as "low," "medium," or "high."

3.2.2 Random Forest Classification

Among supervised learning methods, Random Forest is a well-known machine learning algorithm. It can be used for machine learning problems involving both regression and classification. Its basis is the idea of ensemble learning, which is a technique that combines multiple classifiers to solve a difficult problem and improve the performance of the model.

The Random Forest classifier "contains a number of decision trees on various subsets of the given

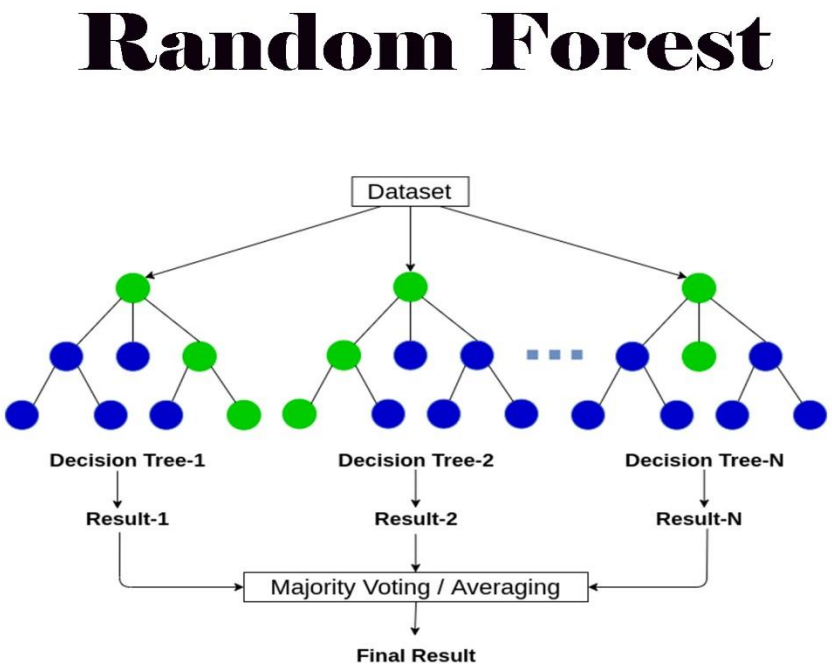

Figure 3. 2: Random Forest Classification

dataset and takes the average to improve the predictive accuracy of that dataset."

Rather than depending only on one decision tree's forecast, the random forest aggregates them all and makes its prediction based on the majority vote of projections. Bigger tree counts in the forest are linked to improved accuracy and a lower risk of overfitting. Figure 2 depicts how the Random Forest algorithm works.

3.2.3 Decision Tree

a) The decision tree is one of the finest supervised learning strategies for issues involving both regression and classification. It creates a tree structure similar to a flowchart, with each internal node representing a test on an attribute, each branch reflecting the test's result, and each leaf node (terminal node) carrying a class name. Subsets of the training data are split iteratively based on attribute values until a stopping condition is met, such as the minimum number of samples required to split a node or the maximum tree depth.

b) **Terminologies involved in Decision Tree:** The following are a few terms that are frequently used in decision trees:

- **Root Node:** The root node in a decision tree is the highest node, representing the whole dataset and divided into child nodes according to feature values.
- **Internal Nodes:** Decision tree nodes that show splits based on feature values and from where child nodes branch.
- **Leaf Node (Terminal Node):** Nodes in a decision tree that indicate final decision outcomes or classifications and from which no child nodes branch off are called leaf nodes, sometimes known as terminal nodes.
- **Splitting Criterion:** The information gain, entropy, or Gini impurity criterion that is applied at each node to divide the data. It establishes the dividing feature and threshold.
- **Decision Rule:** Regulations at every internal node that specify how the data should be divided according to the values of attributes.
- **Feature:** Attributes or variables used for splitting the data at each node in the decision tree.
- **Branch:** The path from the root node to a leaf node, representing the sequence of decisions made based on feature values.
- **Pruning:** A technique used to prevent overfitting by removing unnecessary branches or nodes from the decision tree.

- **Entropy:** A measure of impurity or disorder in a dataset, used as a splitting criterion to maximize information gain.
- **Gini Impurity:** A measure of impurity or uncertainty in a dataset, used as a splitting criterion to minimize impurity in child nodes.
- **Information Gain:** A metric used to evaluate the effectiveness of a split in reducing uncertainty in classification tasks, calculated as the difference between the impurity of the parent node and the weighted impurity of the child nodes.
- **Decision Tree Depth:** The number of levels or layers in a decision tree, representing the complexity of the model and its ability to capture intricate patterns in the data.

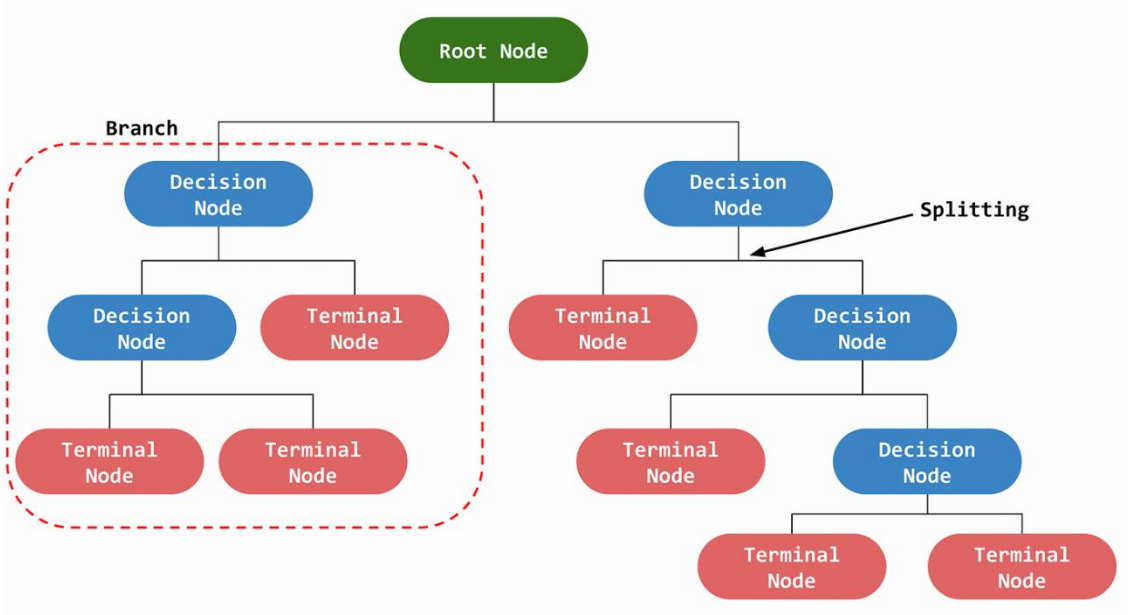

Figure 3. 3: Decision Tree

3.2.4 Support Vector Machine (SVM): A well-liked supervised machine learning approach for both regression and classification applications is called Support Vector Machine (SVM). SVM searches a high-dimensional feature space for the ideal hyperplane or decision boundary to divide various classes.

Support Vector Machine (SVM) is a widely used supervised machine learning technique that is applied to regression and classification problems. SVM looks for the optimal hyperplane or decision boundary in a high-dimensional feature space to split different classes.

To swiftly classify new data points in the future, the SVM algorithm seeks to determine the optimal line or decision boundary that can split n-dimensional space into classes. This ideal decision boundary is called a hyperplane.

SVM chooses the extreme vectors and points that help build the hyperplane. Support vectors are used in this technique, which is also known as the Support Vector Machine. Take a look at the illustration below, which displays a decision boundary or hyperplane between two distinct

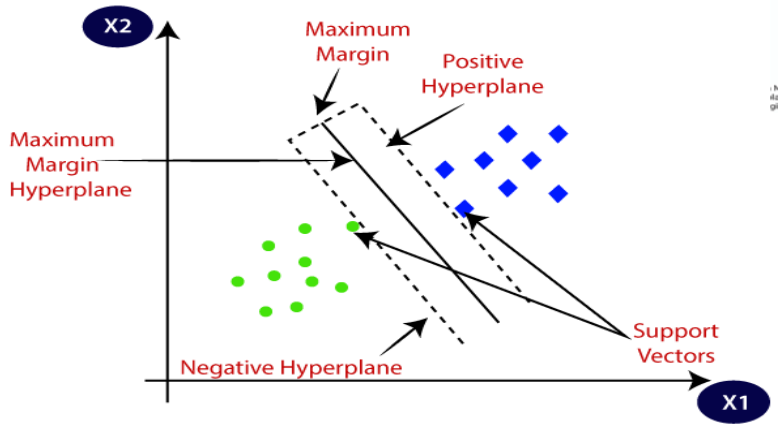

Figure 3. 4: Support Vector Machine (SVM)

Popular supervised learning algorithms for classification and regression tasks include Support Vector Machine (SVM). The steps for implementing SVM in machine learning are as follows:

a) **Data Preparation:** Collect and preprocess your data before using it. The data will need to be cleaned, normalized, and transformed as necessary. Make sure your data is in a format that SVM can use.

b) **Feature Selection:** Choose the pertinent characteristics that will be utilized to train the SVM model in the feature selection step. Remove any unnecessary or duplicated features, then pick features that actually affect the target variable.

c) **Data Partitioning:** Partitioning your data into training and test sets will help you analyze it better. The SVM model will be trained using the training set, and its performance will be assessed using the test set.

d) **Training:** The SVM model is trained using the training set in this step. Finding the best hyperplane to maximize the margin between the various classes is the objective. Different versions, like linear SVM or kernel SVM, can be used to train SVM.

3.2.5 K-Nearest Neighbor (KNN):

For classification and regression applications, the k-Nearest Neighbors (KNN) algorithm is a straightforward and understandable machine learning technique. This is an example of instance-based learning, in which the algorithm finds the "k" nearest data points (neighbors) in the feature space to the input sample, memorizing the full training dataset and using that information to create predictions. Since KNN is non-parametric, it makes no assumptions regarding the distribution of the underlying data.

Here's how the KNN algorithm works:

a) **Store Training Data**: The algorithm starts by storing the entire training dataset in memory. Each data point consists of features (attributes) and a corresponding class label (for classification tasks) or a numerical value (for regression tasks).

b) **Calculate Distance**: When making predictions for a new input sample, the algorithm calculates the distance between the input sample and every data point in the training dataset. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance. Euclidean distance is commonly used for continuous features, while other metrics may be more suitable for categorical features.

c) **Find Nearest Neighbors**: After calculating distances, the algorithm identifies the "k" nearest neighbors to the input sample based on the calculated distances. The value of "k" is a hyperparameter that needs to be specified by the user. It determines the number of neighbors considered when making predictions. Typically, larger values of "k" result in smoother decision boundaries but may lead to increased computational complexity.

d) **Majority Voting (Classification)** or **Average (Regression)**: For classification tasks, KNN predicts the class label of the input sample by taking a majority vote among the class labels of its "k" nearest neighbors. The class label with the highest frequency among the neighbors is assigned to the input sample. For regression tasks, KNN predicts the numerical value by taking the average of the target values of its "k" nearest neighbors.

e) **Prediction**: After determining the class label or numerical value based on the majority vote or average, KNN assigns this prediction to the input sample.

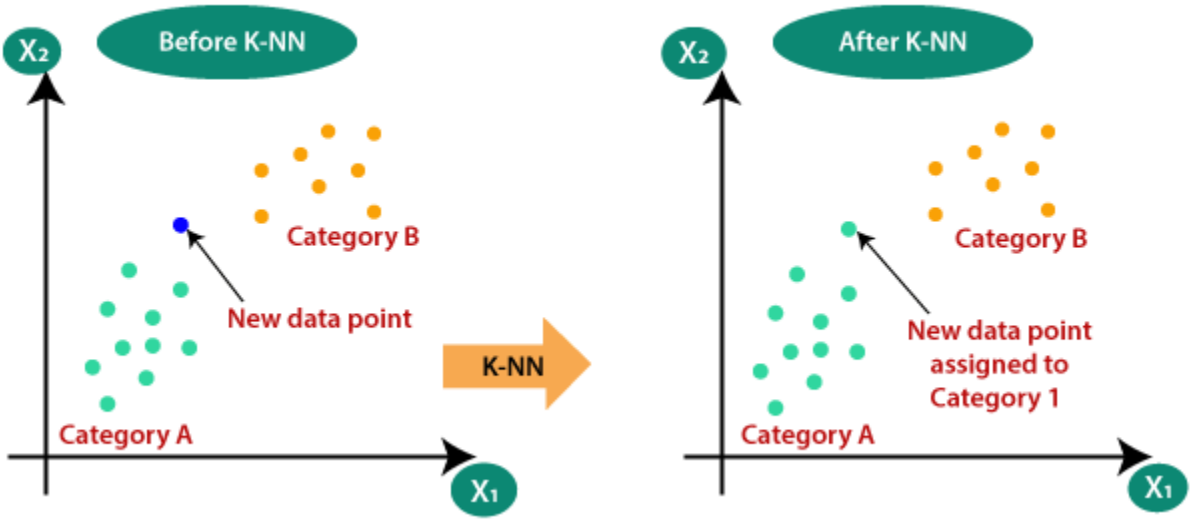

Figure 3. 5: K-NN

f) Evaluation: Another validation or test dataset is used to assess the KNN algorithm's performance. For classification tasks, accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) are common evaluation metrics. Metrics like mean absolute error (MAE) and mean squared error (MSE) are frequently employed for regression jobs.

3.3 Evaluation Metrics

Model performance and accuracy are assessed using metrics for model evaluation. These indicators are useful for assessing the model's effectiveness and contrasting it with alternative algorithms or models. The model is assessed in this work utilizing the Confusion Matrix and its useful metrics, including F1-score, Accuracy, Precision, and Recall. Confusion Matrix: A confusion matrix is used to summarize how well a machine learning model performed on a set of test data. It is widely used to gauge the effectiveness of categorization models. With each input event, these models attempt to predict a categorical label. The number of false positives (FP), false negatives (TN), true positives (TP), and false negatives (FN) that the model produced using the test data is displayed in the matrix (GeeksforGeeks, 2018). True indicates a value forecast that is accurate, whereas False indicates a value prediction that is off. In the n.d., SimpleLearn.com

The amount of times our genuine positive values agree with our positive predictions is called True Positive. The frequency that a model misinterprets positive values as negatives is known as False Positive. Despite expectations of a negative number, the outcome is positive. True Negative is the percentage that our actual negative values agree with our negative forecasts. False Negative is the number of times a model predicts positive results for negative variables when it shouldn't. It's better than what I had thought it would be. The percentage of values that were successfully classified is computed using accuracy. This is the result of dividing the total of all true values by all values. n.d. (Simplilearn.com)

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

where True Positive, True Negative, False Positive, and False Negative are represented, respectively, by the letters TP, TN, FP, and FN. Precision is what determines how accurately the model classifies positive values. It's the ratio of all the expected positive values to the genuine positives. n.d. (Simplilearn.com)

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

where FP and TP, respectively, stand for False Positive and True Positive.

The percentage of real positive cases, or actual positive cases, that were accurately predicted to be positive is measured in Recall. The recall is also known as sensitivity. This implies that there will be an additional fraction of true positive cases that are incorrectly predicted as negative (henceforth known as the false negative). This can also be shown with a false negative rate.

n.d. (Simplilearn.com)

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

where FN and TP stand for False Negative and True Positive, respectively.

more recall numbers are indicative of reduced false negatives and more true positives. Larger false negative values and lower real positive values would result from lower recall. For the banking and healthcare sectors, highly sensitive models are favored. Recall and precision work together well to create the F1-Score. It is useful when taking into account both sensitivity and precision (Simplilearn.com, n.d.).

$$F1 - Score = \frac{Precision * Recall}{Precision + Recall}$$

The relationships between one or more variables are explained by the idea of correlation. These elements may have been present in the raw data that we used to predict our target variable. A precise correlation analysis enhances comprehension of the data. It is crucial for identifying the crucial variables. The correlation coefficient can be calculated using a variety of formulae, but one of the most used is Pearson's correlation, or Pearson's R, which is widely used in linear regression. The coefficient of Pearson's correlation is represented by the symbol "R". The outcome of the correlation coefficient formula falls between -1 and 1. (2022; GeeksforGeeks.)

A really bad relationship is indicated by a score of -1. A score of one denotes incredibly strong bonds. There is absolutely no relationship when the score is 0. It assesses the degree of accuracy of a classification model and evaluates how well it performs when it makes predictions using test data. It not only recognizes the type of error, such as type-I or type-II error, but also the particular kind of error. The model's accuracy and precision can be calculated using the confusion matrix, among other attributes. Javatpoint (n.d.)

CHAPTER 04

Implementation

4.1 Implementation

To address the problem of early brain stroke detection using artificial intelligence, we have developed a study technique that consists of five parts.

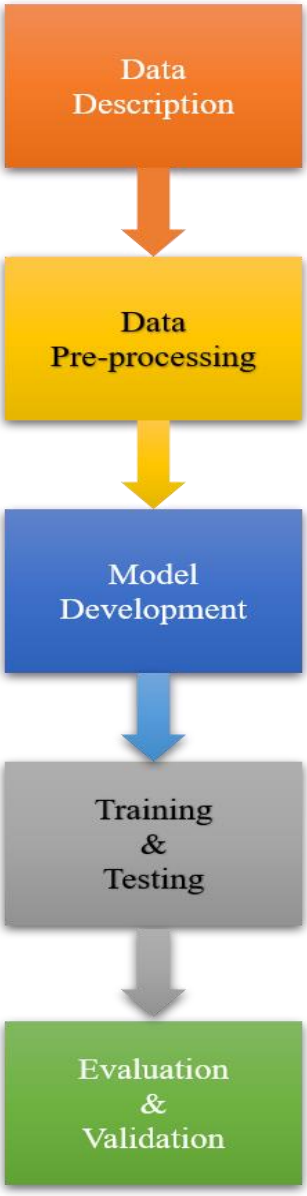

Figure 4. 1: Process of Brain Stroke Detection

4.2 Data Description

This dataset was gathered from the Kaggle website in order to determine the probability that a patient will experience a stroke. This document contains the personal data of 5110 individuals, encompassing 10 qualities (FEDESORIANO 2021). Gender is one of the characteristics. This characteristic speaks to an individual's gender. This data is classified as either Male, Female, or Other. Age is an additional one. This quality relates to an individual's age. These are numerical data. Hypertension is an additional one. This characteristic indicates whether or not a person has hypertension. A patient's score is either 0 or 1, depending on whether they have hypertension or not. These are numerical data. Heart illness is another. This characteristic indicates whether or not a person has heart disease. A patient's score is divided into two categories: zero for no heart disease and one for heart illness. This data is numerical. One more is ever married; this characteristic denotes an individual's marital status. This data is classified as either Yes or No. Work type is an additional one. This characteristic reflects the individual's working environment. It's category data about kids, government jobs, never working, private employment, or self-employment.

The Residence type is an additional one. This characteristic reflects the individual's daily circumstances. This data is classified as either rural or urban. The glucose level is another. This characteristic indicates the patient's average blood glucose level. This data is numerical. The condition of smoking is another. This characteristic indicates whether or not a person smokes. It is categorical data about people who have smoked in the past, never smoked, smoke, or are unknown. Stroke is an additional one. This trait indicates whether or not a person has had a stroke in the past. This data is numerical. A stroke occurred if the patient's score was 1, and not if it was 0. The stroke attribute is the decision class among all of these qualities, while the remaining attributes are the response class.

4.3 Data Preprocessing

Null checking is the first of numerous stages that must be completed in data preprocessing. There may be situations in programming where a variable is not given a value. These kinds of events are usually denoted by the unique value null. The Pandas method is null can be used to verify if any values in the dataset are missing or null. Since the null values are set to True values, only the rows containing null values are displayed. The dataset in this study contains a few null values. To eliminate these null values, the dopa function is employed. Transforming categorical data into numerical data is an additional step. The categorical variables in the data collection ought to be converted to numerical values. While many machine learning algorithms are not natively able to accommodate categorical variables, some of them can. For these transformation processes, Label Encoding and the One Hot Encoding technique are employed.

There is an extra step called the Train-Test Split. The Train-Test Split method was used to evaluate the model's performance. It divides the data into two parts: the training piece and the testing portion, so that it may compare my machine learning model with actual machine learning. The model selection module and the function tra in_test_split are both found in the sci-kit-learn package (Scikit-learn.org, 2018). In our model, the test set comprises 20% of the dataset, whereas the training set comprises 80% of it. Stage four is feature scaling. A technique for standardizing the number of independent variables, or features, in data is feature scaling. It is often finished during the data preparation step and is sometimes referred to as data normalization in the context of data processing. Each characteristic's data values are standardized to have a unit variance and zero mean. The primary computation procedure is to find the distribution mean and standard deviation for each feature and use the formula below to create a new data point.

4.4 Model Development

Utilizing past data to forecast future events, predictive analytics and machine learning are two approaches to prediction. By contrasting the estimated results with the actual values, the performance of the trained prediction models is then assessed using fresh data. The study's algorithms include: Decision Tree: Decision trees are decision assistance techniques that have a tree-like appearance. There are three nodes in it: a root node, leaf nodes, and decision nodes. A decision tree method divides a training dataset into branches, and then further divides the branches into more branches. This pattern continues until it comes to a leaf node. The leaf node cannot be further separated. The decision tree's nodes represent the characteristics that were used to predict the result. The decision nodes give links to the leaves. (Program Section for Engineering Education (EngEd), n.d.)

When creating a machine learning model, the most crucial thing to remember is to choose the best approach for the given dataset and job. Adopting a decision tree has two key advantages: it is easy to grasp and often resembles human decision-making processes; also, due of its tree-like form, the decision tree's explanation is obvious. JavaTpoint (2021). The model is fitted to the training set by importing the DecisionTreeClassifier class from the sklearn.tree package. Below is the code.

Here is the code.

```
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy',
random_state=0)   classifier.fit(x_train, y_train)
```

Code 1.

The code above created the classifier object and handed it two important parameters. Criterion = "entropy" is one. Entropy-acquired information determines the quality of split, which is measured by the criterion. For generating random states, another one is random_state='0' (javaTpoint, 2021). Random Forest Classifier is an additional algorithm. Decision trees in a random forest algorithm come in a wide variety. A forest that is trained using bootstrap aggregation or bagging is produced by the random forest algorithm.

This algorithm decides the result based on the decision trees' predictions. It averages the outcomes from several trees to generate predictions. The more trees there are, the more accurate the outcome

is. A random forest eliminates the drawbacks of the decision tree algorithm. It reduces dataset overfitting and increases precision. (Program | Section | Engineering Education (EngEd), n.d.)

Many decision trees are used in random forest systems. A decision tree has three nodes: the decision, the leaf, and the root. Every decision tree's leaf node indicates the result generated by that particular tree. A majority vote determines the result. The ultimate output that the system chooses in this case is the one that the majority of the decision trees choose, according to the Engineering Education (EngEd) Program Section (n.d.). To diagnose patients, medical experts employ Random Forest techniques. Patients are diagnosed by analyzing their past medical experiences. To determine the appropriate dosage for every patient, historical medical records are examined. Importing the Random Forest Classifier allows the Random Forest technique to be fitted to the training set of data. Here is the code.

```
from sklearn.ensemble import RandomForestClassifier
classifier                RandomForestClassifier(n_estimators=        10,
criterion="entropy")    classifier.fit(x_train, y_train)
```

Code 2.

The code above created the classifier object and handed it two important parameters.
The first is the minimal number of trees needed for the Random Forest, or n_estimators=. Ten is the default value. I have to deal with the overfitting problem even though I can choose any number I choose. An additional criterion is = Its objective is to evaluate the split's accuracy. "Entropy" has been used here as a proxy for knowledge acquisition (JavaTpoint, n.d.). The Naïve Bayes classifier is an extra algorithm that is widely used in supervised machine learning for text classification and other classification applications. This method is predicated on the idea that, given the class, the qualities of the input data are conditionally independent, enabling the algorithm to predict outcomes with high speed and precision. In challenging situations when the data distribution is illdefined, this technique improves the classifier's performance by using a kernel function to estimate the probability density function of the input data. As a result, the naive Bayes classifier is a helpful machine learning tool, particularly for text categorization, spam filtering, and sentiment analysis applications. It is very beneficial for very large data sets and is simple to create. The Naive

Bayes algorithm is well-known for its ease of use, effectiveness, and superior results in numerous real-world scenarios. Chauhan (2022). The Naïve Bayes model fits the training set.

Here is the code.

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)
```

Code 3.

The GaussianNB classifier is used to fit it to the training dataset. (JavaTpoint, n.d.)

An additional algorithm used in this research was Support Vector Machine. One class of machine learning is support vector machines. SVM creates a hyperplane in multidimensional space to distinguish between several classes. After constructing the ideal hyperplane iteratively, it is employed to reduce error. Support vector machines' main goal is to identify the maximum marginal hyperplane (MMH) that best splits the dataset into classes (SVM). A linear hyperplane is not always the best solution for a problem. The kernel trick, a method used by SVM, adds dimensions to nonseparable issues in order to make them separable. When handling non-linear separation problems, it performs admirably. A kernel trick can be used to create a classifier that is more accurate (Datacamp.com, n.d.). The SVM classifier is fitted with the training set. Import the SVC class from the Sklearn.svm package in order to build the SVM classifier. This is the code.

```
from sklearn.svm import SVC # "Support vector classifier"
classifier = SVC(kernel='linear', random_state=0)
classifier.fit(x_train, y_train
```

Code 4.

Since I am building an SVM for data that can be separated linearly, I used kernel='linear' in the code above. I am able to change it for non-linear data. Next, the classifier is fitted using the x_train and y_train training datasets. JavaTPoint (n.d.)

Some people refer to the K-Nearest Neighbors (KNN) family of classification and regression algorithms as memory-based learning approaches. Given how accurate its predictions are, the KNN algorithm might even be able to compete with the most accurate models. When great accuracy is needed but a human-readable model is not needed, the KNN technique can be applied. "Feature similarity" is a technique used by the KNN algorithm to forecast the values of any new data points. This implies that a value is assigned to the new point based on how closely it resembles the points in the training set. The KNN parameter, "k," denotes the number of nearest neighbors. The distance between each training point and the new point is initially determined using the Euclidean, Manhattan, and Hamming distances, depending on whether the distance is continuous or categorical. Based on the average of these data points, the final prognosis for the new point is made. Vidhya Analytics, 2019. . By importing the KNeighborsClassifier class from the Sklearn Neighbours library, the K-NN classifier is fitted to the training set of data

```
from sklearn.neighbors import KNeighborsClassifier   classifier=
KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )
classifier.fit(x_train, y_train)
```

Code 5.

The class's Classifier object is created following the import of the class. This class's parameters are n_neighbors: It lists the required neighbors of the algorithm. Usually, five are needed. metric='minkowski': This default setting and p=2 determine the distance between the points: That is equivalent to the classical Euclidean metric. (JavaTpoint, 2021)

4.5 Working Flowchart

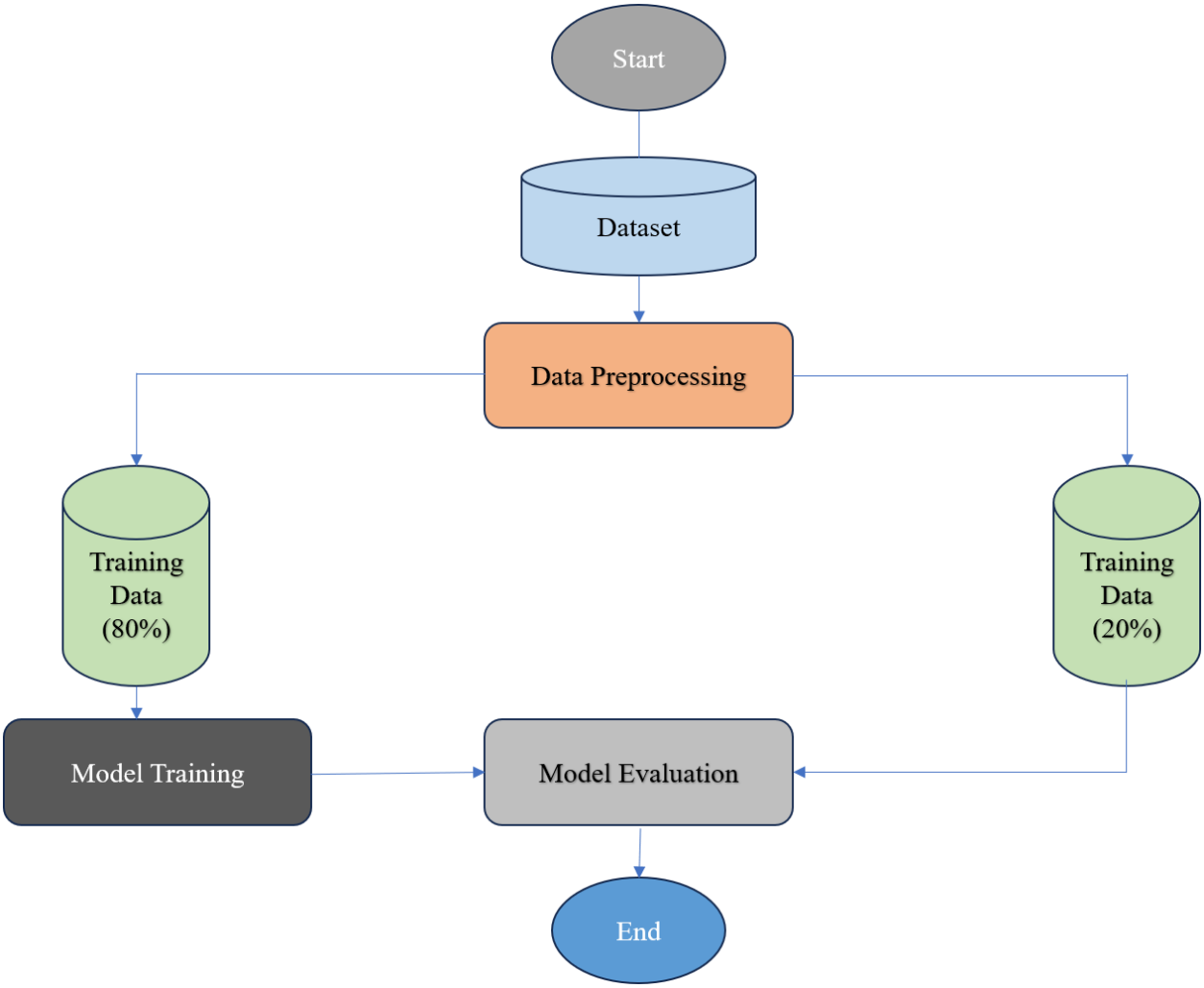

Figure 4. 2: Working Flowchart of this study

Python was used to code this portion of the project. As a result, all required Scikit-learn and Python libraries are used. The preprocessing step involves preparing the data for future use after it has been collected. The dataset is then divided into two sections: testing and training data. The five classifiers are used to train the training dataset. The performance of the testing data is assessed using the same classifiers. The optimal algorithm is determined when the outcome is analyzed.

CHAPTER 05

RESULT AND DISCUSSION

5.1 Result Analysis and Discussion

Our evaluation of the model will be based on several metrics, such as each classifier's F-1 score, accuracy, precision, and recall. For testing, we used 25% of our dataset, and for training, we used the remaining 75%. The performance that each classifier demonstrated throughout testing will be thoroughly analysed in the sections that follow.

5.2 Logistic Regression

The accuracy, precision, recall and F1-score of Logistic Regression algorithm are shown in below table 7.

Table 2: Accuracy, Precision, Recall and F1-score of Logistic Regression

Accuracy	Precision	Recall	F1-score
94.68 %	89.65 %	94.68 %	92.1 %

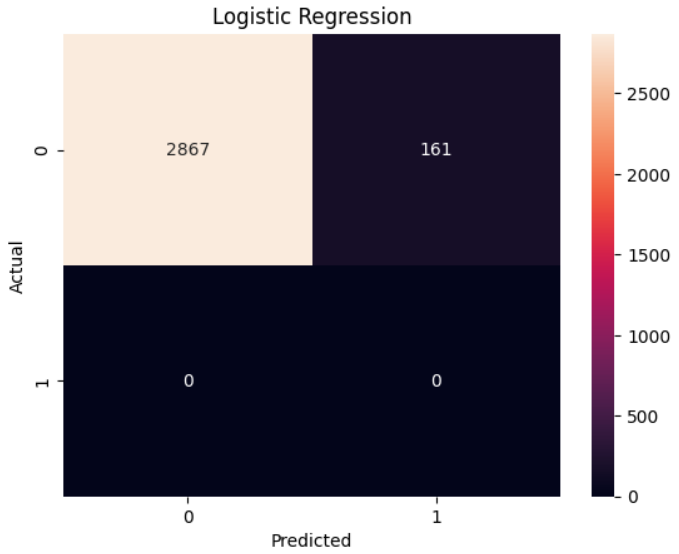

Figure 5. 1: Confusion Matrix for Logistic Regression

Based on Figure 9, the accuracy is around 94.68%. A precision score of 89.65% was given. 2867 valid data were appropriately labelled as authentic parameters. Furthermore, the classifier's 94.68% recall and 92.1% F-1 score prove its validity.

- True Positive (TP): 0 instances were correctly classified as "Positive".
- False Positive (FP): 161 instances were incorrectly classified as "Positive" when they were actually "Negative".
- False Negative (FN): 2867 instances were incorrectly classified as "Negative" when they were actually "Positive".
- True Negative (TN): 0 instances were correctly classified as "Negative".

5.3 Decision Tree

The accuracy, precision, recall and F1-score of Decision tree algorithm are shown in below table 7.

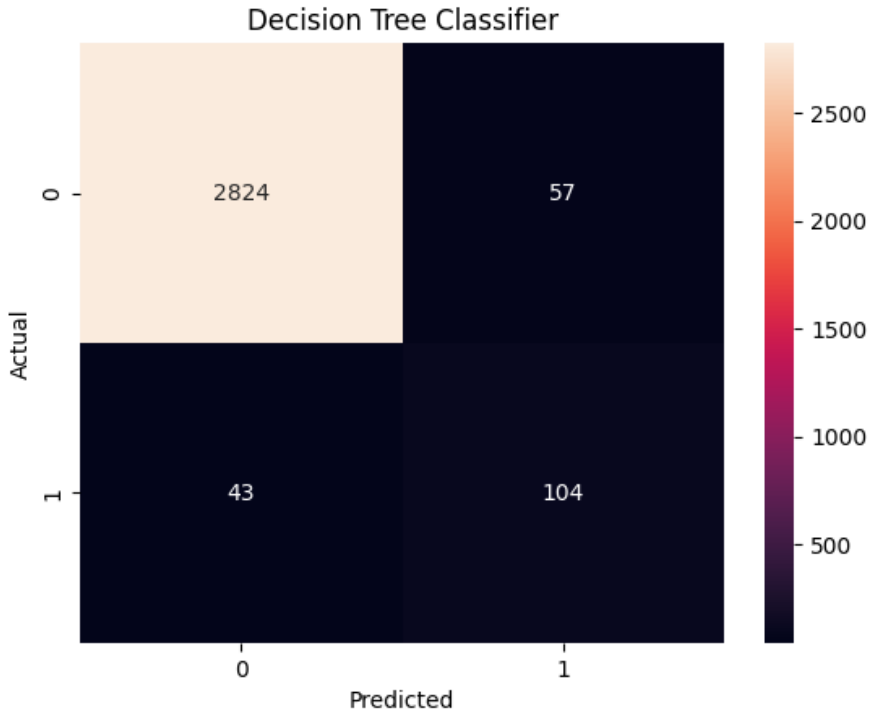

Figure 5. 2: Confusion Matrix for Decision Tree

Table 3: Accuracy, Precision, Recall and F1-score of Decision Tree

Accuracy	Precision	Recall	F1-score
96.7 %	96.57 %	96.7 %	96.63 %

Based on Figure 9, the accuracy is around 96.7%. A precision score of 96.57% was given. 10091 valid data were appropriately labelled as authentic parameters. Furthermore, the classifier's 96.7% recall and 96.63% F-1 score prove its validity.

- True Positive (TP): 104 instances were correctly classified as "Positive".
- False Positive (FP): 57 instances were incorrectly classified as "Positive" when they were actually "Negative".
- False Negative (FN): 2824 instances were incorrectly classified as "Negative" when they were actually "Positive".
- True Negative (TN): 43 instances were correctly classified as "Negative".

5.4 Random Forest

The accuracy, precision, recall and F1-score of Random Forest algorithm are shown in below table 7.

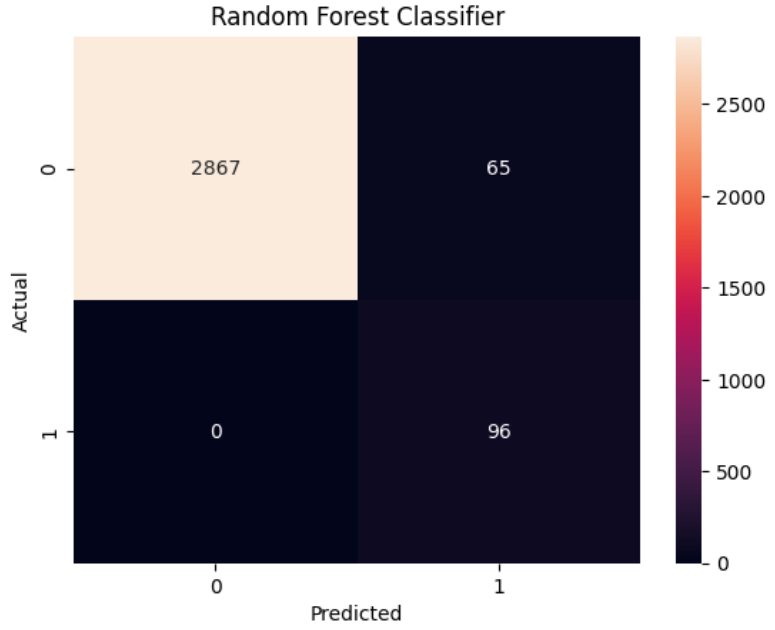

Figure 5. 3: Confusion Matrix for Random Forest

Table 4: Accuracy, Precision, Recall and F1-score of Random Forest

Accuracy	Precision	Recall	F1-score
97.85 %	97.9 %	97.85 %	97.59 %

Based on Figure 9, the accuracy is around 97.85%. A precision score of 97.9 % was given. 10091 valid data were appropriately labelled as authentic parameters. Furthermore, the classifier's 97.85% recall and 97.59 % F-1 score prove its validity.

- True Positive (TP): 96 instances were correctly classified as "Positive".
- False Positive (FP): 65 instances were incorrectly classified as "Positive" when they were actually "Negative".
- False Negative (FN): 2867 instances were incorrectly classified as "Negative" when they were actually "Positive".
- True Negative (TN): 0 instances were correctly classified as "Negative".

5.5 Support Vector Machines (SVM)

The accuracy, precision, recall and F1-score of SVM algorithm are shown in below table 7.

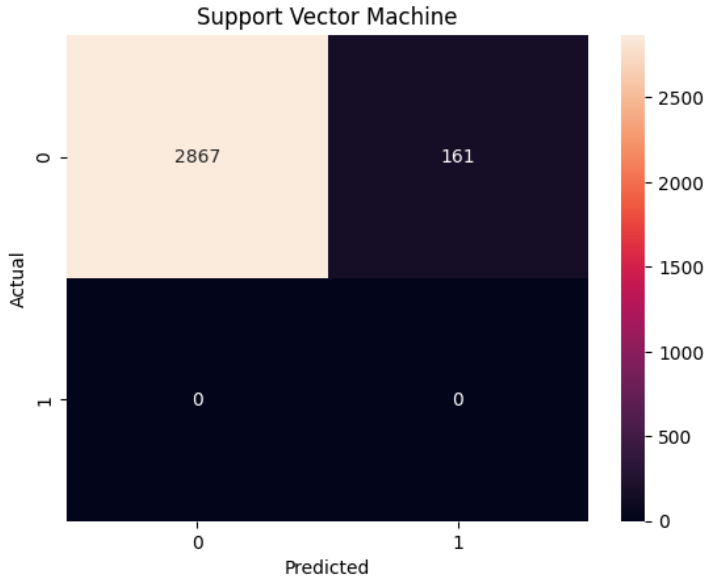

Figure 5. 4: Confusion Matrix for SVM

Table 5: Accuracy, Precision, Recall and F1-score of SVM

Accuracy	Precision	Recall	F1-score
94.68 %	89.65 %	94.68 %	92.1 %

Based on Figure 9, the accuracy is around 94.68%. A precision score of 89.65 % was given. 10091 valid data were appropriately labelled as authentic parameters. Furthermore, the classifier's 94.68% recall and 92.1 % F-1 score prove its validity.

- True Positive (TP): 0 instances were correctly classified as "Positive".
- False Positive (FP): 161 instances were incorrectly classified as "Positive" when they were actually "Negative".
- False Negative (FN): 2867 instances were incorrectly classified as "Negative" when they were actually "Positive".
- True Negative (TN): 0 instances were correctly classified as "Negative".

5.6 K-Nearest Neighbors (KNN)

The accuracy, precision, recall and F1-score of KNN algorithm are shown in below table 7.

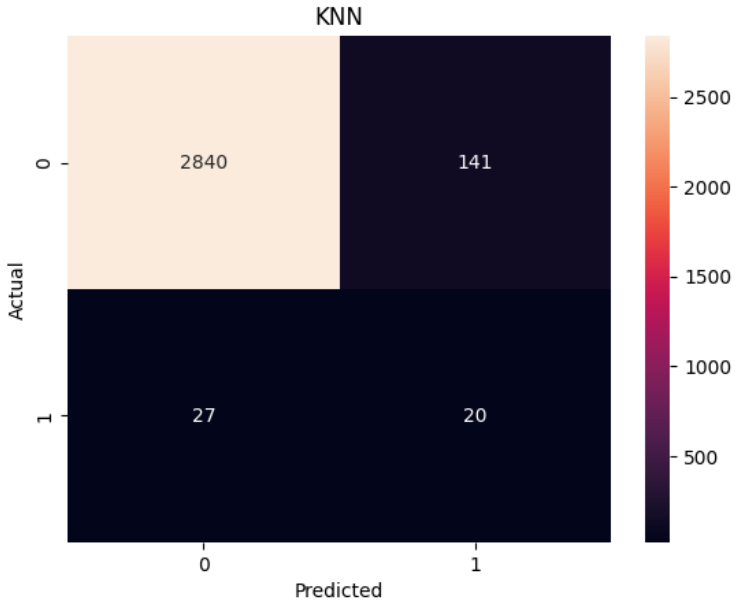

Figure 5. 5: Confusion Matrix for KNN

Table 6: Accuracy, Precision, Recall and F1-score of KNN

Accuracy	Precision	Recall	F1-score
94.45 %	92.47 %	94.45 %	92.99 %

Based on Figure 9, the accuracy is around 94.45 %. A precision score of 92.47 % was given. 10091 valid data were appropriately labelled as authentic parameters. Furthermore, the classifier's 94.45 % recall and 92.99 % F-1 score prove its validity.

- True Positive (TP): 20 instances were correctly classified as "Positive".
- False Positive (FP): 141 instances were incorrectly classified as "Positive" when they were actually "Negative".
- False Negative (FN): 2840 instances were incorrectly classified as "Negative" when they were actually "Positive".
- True Negative (TN): 47 instances were correctly classified as "Negative".

5.7 Comparison

The summery of these classifiers is shown the below table 12.

Table 7: Comparison summary of the classifiers

Model Name	Accuracy	Precision	Recall	F1-score
Decision Tree	96.70%	96.57%	96.70%	96.63%
Random Forest	**97.85%**	**97.90%**	**97.85%**	**97.59%**
Logistic Regression	94.68%	89.65%	94.68%	92.10%
Support Vector Machines (SVM)	94.68%	89.65%	94.68%	92.10%
K-Nearest Neighbors (KNN)	94.45%	92.47%	94.45%	92.99%

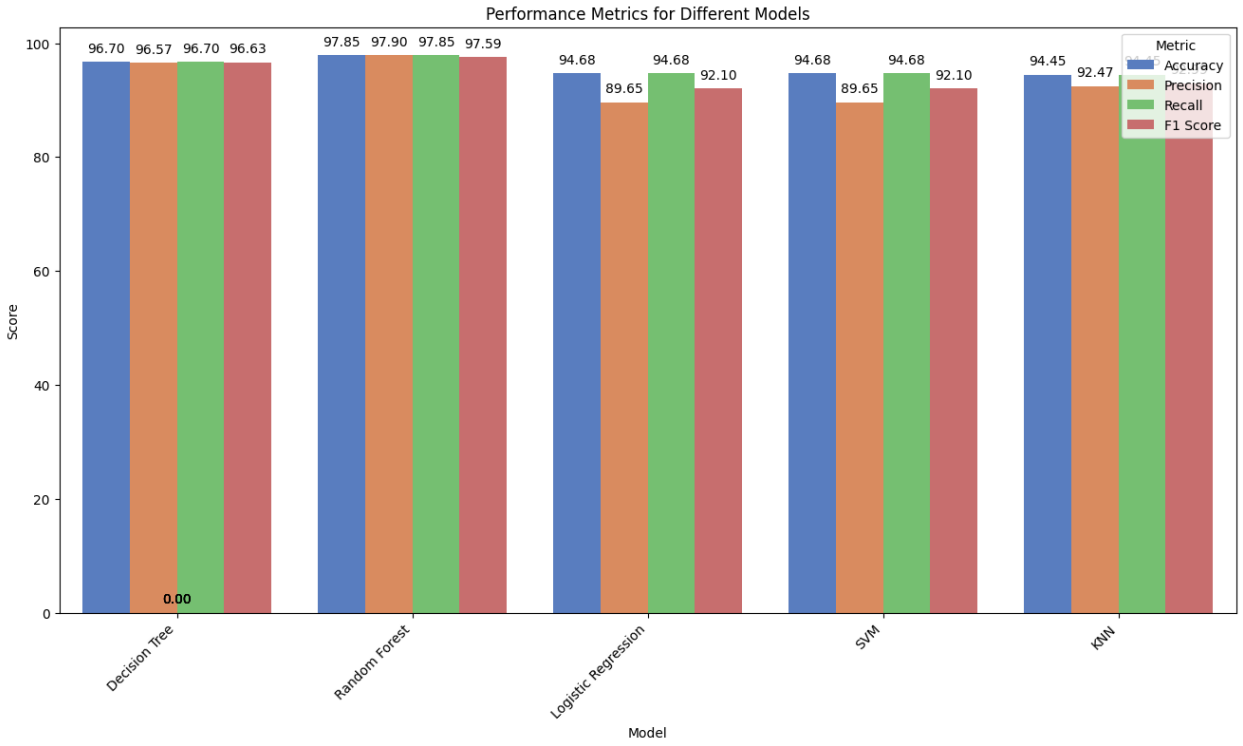

Figure 5. 6: Histogram of the Classifiers

Table 12 indicates that the Decision Tree Classifier achieved an accuracy of 96.70%. SVM and Logistic Regression Classifier also showed outstanding accuracy of 94.68% and 94.68% respectively. **However, with 97.85% accuracy, the Random Forest Classifier had the highest performance**. We may also observe that, due to the algorithm's design, the KNN Classifier had the lowest accuracy of 94.45%. The column chart in Figure 14 illustrates the results we obtained using the classifiers.

CHAPTER 06

CONCLUSION

6.1 Conclusion

Stroke mortality rates are high and rising daily. Different types of strokes are caused by a number of stroke risk factors. The creation of a machine learning model can lessen the severity of stroke and help detect it earlier. A substantial amount of information may be acquired based on the thorough analysis of all available studies to better comprehend how machine learning techniques can aid in the very accurate prediction of this disease. In this thesis work, five classifiers are used: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), and K-Nearest and Neighbors (KNN) Algorithm. The dataset was trained with 5110 data, which included the attributes gender, age, hypertension, heart disease, glucose level, smoking status, marital status, job type, and residence type. After that, a confusion matrix and measures like accuracy, precision, recall, and F1-score are used to evaluate the model. According to this study, RF and DT classifiers are more accurate than the others at making predictions, while KNN shows the least ineffective result.

The study's conclusions suggest that AI is capable of predicting stroke. The combination of the RF and DT classifiers, which produces the highest accuracy RF (97.70%), followed by DT (96.70%), is the most suitable strategy for stroke prediction. By detecting the illness early, the suggested method is probably going to lower the risk of stroke and save costs associated with medical consultations, diagnosis, and treatment. My ambition is to combine an existing model that will improve performance indicators with deep learning-based imaging, like brain MRIs and CT scans.

REFERENCES

1. Alaka, S.A., Menon, B.K., Brobbey, A., Williamson, T., Goyal, M., Demchuk, A.M., Hill, M.D. and Sajobi, T.T. 2020. Functional Outcome Prediction in Ischemic Stroke: A Comparison of Machine Learning Algorithms and Regression Models. Frontiers in Neurology, 11. doi:https://doi.org/10.3389/fneur.2020.00889.

2. Analytics Vidhya. 2020. Categorical Encoding One Hot Encoding vs Label Encoding. Available at:https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding vshttps://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/labelencoding-using-scikit-learn/.

3. Scikit-learn.org. 2018. sklearn.model_selection.train_test_split — scikit-learn 0.20.3 documentation. Available at: https://scikithttps://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.htmllearn.org/stable/modules/gen erated/sklearn.model_selection.train_test_split.html.

4. Analytics Vidhya. 2019. A Practical Introduction to K-Nearest Neighbor for Regression. Available at: https://www.analyticsvidhya.com/blog/2018/08/khttps://www.analyticsvidhya.com/blog/2018/08/k-nearestneighbor-introduction-regression-python/nearest-neighbor-introduction-regression-python/.

5. Chauhan, N. 2022. Naïve Bayes Algorithm: Everything You Need to Know. Available at: https://www.kdnuggets.com/2020/06/naive-bayes algorithmhttps://www.kdnuggets.com/2020/06/naivebayes-algorithm-everything.htmleverything.html.

6. Centers for Diseases Control and Prevention. 2023. Know Your Risk for Stroke. Available at: https://www.cdc.gov/stroke/risk_factors.htm#:~:text=Not%20getting%20enough%20physical%20activity, lower%20your%20chances%20for%20stroke

Datacamp.com. (n.d.). Scikit-learn SVM Tutorial with Python (Support Vector Machines). Available at: https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python

Emon, M.U., Keya, M.S., Meghla, T.I., Rahman, Md.M., Mamun, M.S.A. and Kaiser, M.S. 2020. Performance Analysis of Machine Learning Approaches in Stroke Prediction. doi:https://doi.org/10.1109/ICECA49313.2020.9297525

Engineering Education EngEd Program Section. (n.d.). Introduction to Random Forest in Machine Learning. Available at: https://www.section.io/engineeringeducation/introduction-to-random-forest-inmachinelearning/#:~:text=The%20(random%20forest)%20algorithm%20establishes Accessed 26 Jun.
2023.

FEDESORIANO. 2021. Stroke Prediction Dataset. Available at: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Govindarajan, P., Soundarapandian, R.K., Gandomi, A.H., Patan, R., Jayaraman, P. and Manikandan, R.
2019. Classification of stroke disease using machine learning algorithms. Neural Computing and Applications, 32(3), pp.817–828. doi:https://doi.org/10.1007/s00521-019-04041-y Accessed 26 Jun 2023.

GeeksforGeeks. 2018. Confusion Matrix in Machine Learning – GeeksforGeeks. Available at: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

GeeksforGeeks. 2022. Pearson Correlation Coefficient. Available at: https://www.geeksforgeeks.org/pearson- correlation-coefficient/

Huang, X., Cao, T., Chen, L., Li, J., Tan, Z., Xu, B.Y., Richard Huan Xu, Song, Y.S., Zhou, Z., Wang, Z.,
Wei, Y., Zhang, Y., Li, J., Huo, Y., Qin, X., Wu, Y., Gold, R., Wang, H., Cheng, X. and Xu, X. 2022. Novel Insights on Establishing Machine Learning-Based Stroke Prediction Models Among Hypertensive Adults.
9. doi:https://doi.org/10.3389/fcvm.2022.901240

Holek, I.M. (n.d.). What Are the 3 Types of Stroke? - CBC Health cbchealth.de. Available at: https://cbchealth.de/en/three-types-of-stroke/

Heo, J., Yoon, J.G., Park, H., Kim, Y.D., Nam, H.S. and Heo, J.H. 2019. Machine Learning– Based Model for Prediction of Outcomes in Acute Stroke. Stroke, 50(5), pp.1263–1265. doi:https://doi.org/10.1161/strokeaha.118.024293

JavaTPoint (n.d.). Support Vector Machine (SVM) Algorithm – Javatpoint. Available at: https://www.javatpoint.com/machine-learning-supporthttps://www.javatpoint.com/machine-learning-supportvector-machine-algorithmvector-machine-algorithm

JavaTpoint. 2021. K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint. Available at:
https://www.javatpoint.com/k-nearest-neighboralgorithm-for-machine-learning

JavaTpoint. 2021. Machine Learning Decision Tree Classification Algorithm - Javatpoint. Available at: https://www.javatpoint.com/machine-learninghttps://www.javatpoint.com/machine-learning-decision-treeclassification-algorithmdecision-tree-classification-algorithm

JavaTpoint (n.d.). Machine Learning Random Forest Algorithm - Javatpoint. Available at:
https://www.javatpoint.com/machine-learning-randomhttps://www.javatpoint.com/machine-learning-randomforest-algorithmforest-algorithm

JavaTpoint (n.d.). Naive Bayes Classifier in Machine Learning - Javatpoint. Available at:
https://www.javatpoint.com/machine-learning-naive-bayeshttps://www.javatpoint.com/machine-learningnaive-bayes-classifier

JavaTpoint (n.d.). Confusion Matrix in Machine Learning - Javatpoint. Available at: https://www.javatpoint.com/confusion-matrix-in-machine-learning

Kim, J.K., Choo, Y.J. and Chang, M.C. 2021. Prediction of Motor Function in Stroke Patients Using
Machine Learning Algorithm: Development of Practical Models. Journal of Stroke and Cerebrovascular
Diseases, 30(8), p.105856. Available at: doi:https://doi.org/10.1016/j.jstrokecerebrovasdis.2021.105856

Lin, C.-H., Hsu, K.-C., Johnson, K.R., Fann, Y.C., Tsai, C.-H., Sun, Y., Lien, L.-M., Chang, W.L., Chen, P.-L., Lin, C.-L. and Hsu, C.Y. 2020. Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry. Computer Methods and Programs in Biomedicine, 190, p.105381. doi:https://doi.org/10.1016/j.cmpb.2020.105381

NHS. 2022. Overview-Stroke. Available at: https://www.nhs.uk/conditions/stroke/ Accessed 14 December 2023

Prentzas, N., Nicolaides, A., Kyriacou, E., Kakas, A. and Pattichis, C. 2019. Integrating Machine Learning with Symbolic Reasoning to Build an Explainable AI Model for Stroke Prediction.
doi:https://doi.org/10.1109/BIBE.2019.00152

Raghav Vashisht 2021. Machine Learning: When to perform a Feature Scaling? - Atoti Community. Available at: https://atoti.io/articles/when-to-perform-afeature-scaling/#:~:text=What%20is%20Feature%20Scaling%3F Accessed 26 Jun 2023

Rebouças, E. de S., Marques, R.C.P., Braga, A.M., Oliveira, S.A.F., de Albuquerque, V.H.C. and Rebouças Filho, P.P. 2018. New level set approach based on Parzen estimation for stroke segmentation in skull CT images. Soft Computing, 23(19), pp.9265–9286. doi:https://doi.org/10.1007/s00500-018-3491-4

Simplilearn.com. (n.d.). What is a Confusion Matrix in Machine Learning? Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrixmachinehttps://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning

SÜT, N. and ÇELİK, Y. 2012. Prediction of mortality in stroke patients using multilayer perceptron neural networks. Turkish Journal of Medical Sciences. doi:https://doi.org/10.3906/sag-1105-20

Sung, S.M., Kang, Y.J., Cho, H.J., Kim, N.R., Lee, S.M., Choi, B.K. and Cho, G. 2020. Prediction of early neurological deterioration in acute minor ischemic stroke by machine learning algorithms. Clinical Neurology and Neurosurgery, 195, p.105892.
Available at: doi:https://doi.org/10.1016/j.clineuro.2020.105892

Trailokya Raj Ojha and Ashish Kumar Jha 2023. Analyzing the Performance of the Machine Learning Algorithms for Stroke Detection. 13 (2), pp.27–35. Available at: doi:https://doi.org/10.5815/ijeme.2023.02.04

Thammaboosadee, S. and Kansadub, T. 2019. Data mining model and application for stroke prediction: A combination of demographic and medical screening data Approach. Interdisciplinary Research Review,
14(4), pp. 61–69. Available at: https://ph02.tcithaijo.org/index.php/jtir/article/view/221565. Accessed: 26
June 2023

WHO.int.(n.d.). World Stroke Day 2022. Available at: https://www.who.int/srilanka/news/detail/29-102022-world-stroke-dayhttps://www.who.int/srilanka/news/detail/29-10-2022-world-stroke-day20222022#:~:text=Stroke%20is%20the%20leading%20cause Accessed 25 June 2023.

World Stroke Organization. (n.d.). European Life After Stroke Forum 2023. Available at: https://www.world-stroke.org/news-and-blog/blogs/european-life-after-stroke-forum-2023 Accessed 26 Jun 2023.

Wu, Y. and Fang, Y. 2020. Stroke Prediction with Machine Learning Methods among Older Chinese.
International Journal of Environmental Research and Public Health, 17(6), p.1828.
doi:https://doi.org/10.3390/ijerph17061828

Yu, J., Park, S., Kwon, S.-H., Ho, C.M.B., Pyo, C.-S. and Lee, H. (2020). AI-Based Stroke Disease Prediction System Using Real-Time Electromyography Signals. Applied Sciences, 10(19), p.6791.
doi:https://doi.org/10.3390/app10196791