Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20057-6

Machine learning for brain-stroke prediction: comparative analysis and evaluation

Rahul Bhowmick[1] · **Soumya Ranjan Mishra**[1] · **Sanjeeb Tiwary**[1] · **Hitesh Mohapatra**[1]

Received: 26 October 2023 / Revised: 31 July 2024 / Accepted: 7 August 2024

Abstract
This study focuses on the intricate connection between general health, blood pressure, and the occurrence of brain strokes through machine learning algorithms. To achieve this, we have thoroughly reviewed existing literature on the subject and analyzed a substantial data set comprising stroke patients. Implementing a combination of statistical and machine-learning techniques, we explored how general health indicators, including overall well-being and blood pressure, influence the risk of strokes. The findings of this study hold substantial implications for stroke prevention, treatment, and the development of novel diagnostic tools and therapies. Our ultimate aim is to gain fresh insights into the intricate interplay of general health and blood pressure, aiding in identifying individuals at risk of future brain strokes. This study entails a data-driven analysis of various algorithms across multiple datasets. Within this scope, we have thoroughly examined the behaviours and accuracy of diverse machine learning algorithms, assessing their interrelationships. This research aims to assist novice researchers in comprehending the performance of different machine learning algorithms in the context of brain stroke prediction.

Keywords Stroke diagnosis · Biomarkers · Machine learning models · Predictive analysis · Support vectors · Neural networks · K-nearest neighbors

1 Introduction

Brain stroke remains a major health concern worldwide, contributing significantly to disability and mortality rates. Understanding the factors that contribute to stroke occurrence is crucial

✉ Soumya Ranjan Mishra
soumyaranjanmishra.in@gmail.com

Rahul Bhowmick
rahulbhowmick2002@gmail.com

Sanjeeb Tiwary
sanjeebtiwary9006@gmail.com

Hitesh Mohapatra
hiteshmahapatra@gmail.com

[1] School of Computer Engineering, KIIT Deemed to Be University, Bhubaneswar 751024, Odisha, India

for effective prevention and treatment strategies [1]. Among these factors, blood pressure and general health indicators have garnered considerable attention for their potential association with stroke. Brain Stroke is a pressing issue of global concern, with a significant impact on both disability and mortality rates. As such, it is paramount to conduct thorough investigations into the various factors that contribute to the occurrence of this condition. Among the key health indicators that have been extensively studied for their potential relationship with stroke, blood pressure stands out as particularly worthy of attention [1]. Recent research by Malone et al. (2020) underscores the importance of gaining a deeper understanding of these factors, as this knowledge can inform the development of effective prevention and treatment strategies. By delving into the complexities of stroke and its contributing factors, we can work towards reducing its devastating impact on individuals and communities around the world. Elevated blood pressure, or hypertension, has long been recognized as a leading modifiable risk factor for stroke [2]. Hypertension disrupts the delicate balance of blood flow in cerebral arteries, leading to damage and potential blockages that can result in ischemic or hemorrhagic strokes [3]. However, the relationship between blood pressure and stroke is complex and multifaceted, involving various underlying health factors. In recent years, the role of general health indicators, such as cholesterol levels, has emerged as an important consideration in stroke research [2, 4]. High-density lipoprotein (HDL) and low-density lipoprotein (LDL) cholesterol levels have been specifically linked to the onset of atherosclerosis, a disorder marked by the accumulation of plaque in blood arteries, including those supplying the brain [5]. Disruptions in cholesterol profiles can contribute to arterial plaque formation and subsequent stroke events [6]. To accurately assess an individual's stroke risk, it is crucial to understand the intricate interplay between blood pressure, general health indicators, and stroke occurrence [7]. Advances in statistical and machine-learning techniques provide powerful tools to analyze large datasets and uncover hidden patterns and relationships.

This study aims to investigate the relationship between general health, blood pressure, and the occurrence of brain stroke through a comprehensive approach. Firstly, a systematic review of the existing literature will be conducted to gather insights from previous studies. This will help establish a foundation of knowledge and identify gaps in understanding [7–9]. A large dataset comprising stroke patients has been analyzed using statistical and machine-learning techniques. With the use of these cutting-edge techniques, new linkages can be found and predictive models for stroke risk can be created [9]. The findings of this study carry significant implications for stroke prevention and treatment. By identifying key factors that influence stroke risk, healthcare professionals can develop targeted interventions and personalized strategies to mitigate this devastating condition [10]. Furthermore, the results may contribute to developing innovative diagnostic tools and therapies, facilitating early detection and intervention in individuals at risk of stroke.

In this study, we aim to investigate the relationship between general health and blood pressure. We will analyze the existing literature on the subject and perform our analysis using a large dataset of patients with different types of brain strokes [6, 8]. We will use various statistical and machine-learning techniques to analyze the data and identify patterns and relationships between the variables. The determination of HDL and LDL cholesterol percentages in the bloodstream presents a valuable avenue for predicting brain stroke risk. However, further research is warranted to explore the interplay between genetic factors, lifestyle interventions, and novel biomarkers in conjunction with cholesterol percentages. Predictive models can be improved by using AI and machine learning methods, and tailored therapeutics can be discovered through interventional research that modifies these cholesterol fractions. By advancing our understanding in these areas, we can refine stroke risk assessment and develop more personalized preventive strategies, ultimately reducing the burden of

brain strokes and improving cardiovascular health outcomes. The work can be much more effective if we somehow calculate the percentage of HDL and LDL. While LDL (low-density lipoprotein) is referred to as "bad" cholesterol because a high amount of LDL causes cholesterol to accumulate in your arteries, HDL (high-density lipoprotein) is regarded as "GOOD" cholesterol because it helps remove other types of cholesterol from your system. We will have far more detailed information and comprehension for forecasting brain stroke if we can determine the percentage in the blood. This study could have produced important findings. This research seeks to provide new insights into the complex relationship between general health, blood pressure, and the occurrence of brain stroke. By combining a comprehensive literature review with advanced data analysis techniques, this study aims to enhance our understanding of stroke risk factors. Ultimately, the findings of this research have the potential to improve stroke prevention and management strategies, reducing the incidence and burden of brain stroke [11].

2 Related works

A paper published in 2010 explores about the community machine learning method for stroke prediction. This paper proposes a new automatic feature selection algorithm that selects robust features using conservative means as the heuristic. When combined with SVM, a larger area under the ROC curve is obtained compared to the Cox proportional hazards model and the L1 normal Cox feature selection algorithm. Additionally, they used a marginally-based censored regression algorithm that combines the marginal-based classifier concept with censored regression to obtain a better consistency index than the Cox model. They also discovered potential risks that were not identified with traditional methods [12, 13]. In 2016, research was conducted on developing stroke models using machine learning algorithms such as SVM and ANN. Educational supervision was discussed. Their study relied on the usage of neural networks and Support Vectors, with models that improved in accuracy by 98.1% and 91%, respectively [14].

In 2016, a study was conducted on latent visual working memory (WM), concerning the human cortex. Their real aim lies in whether WM could provide memory hints after passing certain instructions as input. Disruptions of mnemonic representations and behavioural anomalies were discovered, related to loss of memory. Reframing of FMRI (Functional Magnetic Resonance Imaging) images was used, and intuitions were gained, which said memory items may be recorded in a virtual neural code, which uses Working Memory [15]. A work has been proposed on the prediction of strokes using AI/ML where, with the help of PCA (Principal Component Analysis), dimensionality reduction was made. After performing feature selection using Decision Trees, they used Propagation Neural Networks for the stroke classification [16]. A researcher wrote an article on hitting algorithm development. They used logistic regression for model derivation. Apart from the estimation of strokes, their deployed model can develop strategies to minimize stroke, in patients having potential risks [17].

The efficacy of electronic medical records—an outstanding tool for stroke prediction—was highlighted in a published research on the prediction of strokes using electronic medical records. Notably, there is a focus on predictive analysis using EMRs. Cross-entropy loss functions, ROC curves, and other methods were used to base the results [18]. In 2022, a group of academics conducted research on stroke prediction using machine learning models. They found criteria to predict using a variety of statistical indicators. Following the comprehension and assessment of all relevant variables, Neural Networks were employed due to their ability

to generate intelligent decisions and improve estimations [19]. In 2022, a paper was published which is related to the Evaluation of drug regimen complexity scores as predictors. Their main aim was to compare scores in MRC tools concerning clinical tools. The logistic classifier was used for prediction. Conclusions were generated such as clinical scores that were less favorable performed better as compared with MRC scores [20].

A work published in 2023, predicts the efficiency of electrical modulation of the brain. FMRI images were the data, on which predictions were made. A multivoxel-driven modelling approach was used to predict memory representations based on utility prediction rules. They found that items with higher predictability were stored in memory with higher accuracy [21]. A paper on Adaptation of the Concept of Brain Reserve for the Prediction of Stroke Outcome: Proxies, Neural Mechanisms, and Significance for Research. Brain Reserve (BR) theory has been used to understand the occurrence of strokes. Performing regular MRI checkups helps in explaining individual variances in stroke recovery. One of the possibilities listed includes customized therapy, the reason behind this is that BR shows how the brain can recover from the damage. This therapeutic prevention is thus beneficial while considering research on stroke, neurodegeneration, and healthy ageing, In 2024 [22].

3 Proposed work

Review of prior research and identified technological deficiencies Over the years, medicine has made tremendous advancements in the diagnosis and treatment of various diseases. For instance, heart patients can benefit from the use of electronic devices and electronic medical records. In this article, we will discuss our research findings, as well as relevant resources and future developments in this clinical area. With personalization becoming more accessible and cost-effective, medical devices are now prevalent, and diagnostic applications incorporate multiple features. For example, the device known as AliveCor offers an affordable Electrocardiogram (ECG) device that measures a person's ECG, calculates real-time statistics and displays medical-grade heart data. To investigate one of these, a method was developed to filter data from the long-term ECG of patients with long QT syndrome (LQTS), and then advanced AI and Machine Learning Technology were applied to identify patterns that indicate risk for various symptoms such as heart rhythms, hypertension, and cardiac arrhythmias. LQTS is a disorder that usually affects ion channels in cardiomyocytes, allowing abnormal electrical impulses to occur, causing rapid and dangerous heartbeats [23]. Scalability is becoming important in medical research, and thus the analysis of big data not only leads to a deeper understanding of various diseases but also aids in self-healing [24]. Personalized, real-time surveillance system development. Routine monitoring of blood pressure and other health indicators often includes routine clinical measurements that fail to capture such events [25]. The advent of effective technologies such as smartwatches provides the opportunity to collect continuous data promptly, resulting in better monitoring and risk assessment [26]. However, algorithms and techniques are still needed to analyze and interpret the data collected by these devices.

Statistical Also, integrating statistics and machine learning into healthcare is still a challenge. While these methods show promise in identifying risk factors and predicting outcomes, they still need to be validated and integrated into existing healthcare systems well [27]. Translating research findings into practical and accessible tools for clinicians is important for their effective use in stroke prevention and treatment [28]. Addressing these different technologies

will allow the development of the right risk assessment tools, self-monitoring, and improved stroke management strategies. This study aims to fill the gap by using advanced statistical techniques and machine learning to analyze large datasets of stroke patients and provide recommendations for social interactions related to health, hypertension, and stroke incidence these technologies.

Methodology This article analyzes data on brain stroke and uses different learning models to study and predict their outcomes. This will allow us to understand the estimation of the selection criteria we decided to use to find the data. In the healthcare sector, the size of data that is generated daily is more than terabytes and petabytes. Choosing the right data, and putting cleaned data into work can help us in gaining better outcomes. Thanks to big data analysis and deep neural networks, predictions can be made based on the symptoms of various diseases.

Health management and the concept of health play an important role in people's lives. This is an important factor that makes life happy. But in today's world, exposure to illness is very common due to many factors such as daily stress, lack of sleep, malnutrition, and many other unfair factors. However, it has been observed that people often ignore or ignore these diseases or symptoms due to their busy schedules. Sometimes this neglect can lead to more serious diseases. Therefore, in this fast-moving world, the need for projects is more important than anything else. Observation plays an important role in this work. The healthcare industry generates an enormous amount of information daily, with the majority of it being in the form of unstructured data. So our main task is to first identify the correct data for our purpose, then we need to analyze the data to eliminate any inconsistencies or anomalies. If the data is not cleaned, the results will not be accurate and reliable. In the Prediction Framework, we used medical data from open-source databases. To improve the dataset's quality, we carefully worked on the features, including feature selection and elimination during the initial data preprocessing. We normalized the data with scaling techniques to help the machine/model understand it better. For this, we have used the *StandardScaler()* function, which standardizes and transforms the data by setting the Mean (μ) as 0, and the Standard Deviation (σ) as 1. Standardizing data reduces the effects of varying scales, facilitating and enhancing the model performance. Our main goal was to achieve high accuracy and minimize the loss of function because of the importance of the medical field. Once the dataset was analyzed, we identified several models that were appropriate for the classification use case. The SVM, ANN, and KNN models were among the most promising, so we pursued them for further evaluation. To ensure reliability and generalizability, we implemented a rigorous validation and testing process in each iteration, covering both the data and model components. During the validation phase, we thoroughly evaluated the model's performance and its capability to handle new data. If we noticed any indications of overfitting or underfitting, we made necessary adjustments to the data. In situations where the model's performance was inadequate, we used different model tuning methods, such as hyperparameter optimization, virtual hyperparameters, and cross-validation. This iterative approach enabled us to refine the models and make them capable of handling real-world situations.

To confirm the model's efficacy, we put it through rigorous testing using various parameters and test cases. This was a crucial step to ensure that the AI solution could consistently perform well in real-time medical scenarios. Through this thorough testing process, we were able to evaluate the model's strengths and weaknesses, pinpoint potential areas for improvement, and continuously enhance its overall performance. Our goal was to create an advanced AI model that could be highly effective, precise and sturdy in real-time medical applications. We have taken great care to conduct extensive testing and optimization to ensure that the

model meets the exacting standards of the medical field and can be trusted to provide reliable results (Fig. 1).

4 Data source

The data used were obtained from electronic health records (EHR) managed by McKinsey & Company. The database is open to the public and contains 11 features and 1 target variable (which predicts whether a patient is suffering from Stroke or not). The type of the features, (whether it is a categorical or a numerical variable) has been shown in the following distribution, as in Fig. 2.

We have used factor analysis to determine the importance of different traits in predicting stroke risk. Research has indicated that certain behaviours, such as age, diabetes, heart disease, high blood pressure, and level of physical activity, have a greater impact than others. We have used different graphs to show the dependency of these traits, indicating that age and diabetes had the highest stroke risk. Persistent high blood pressure is a medical condition

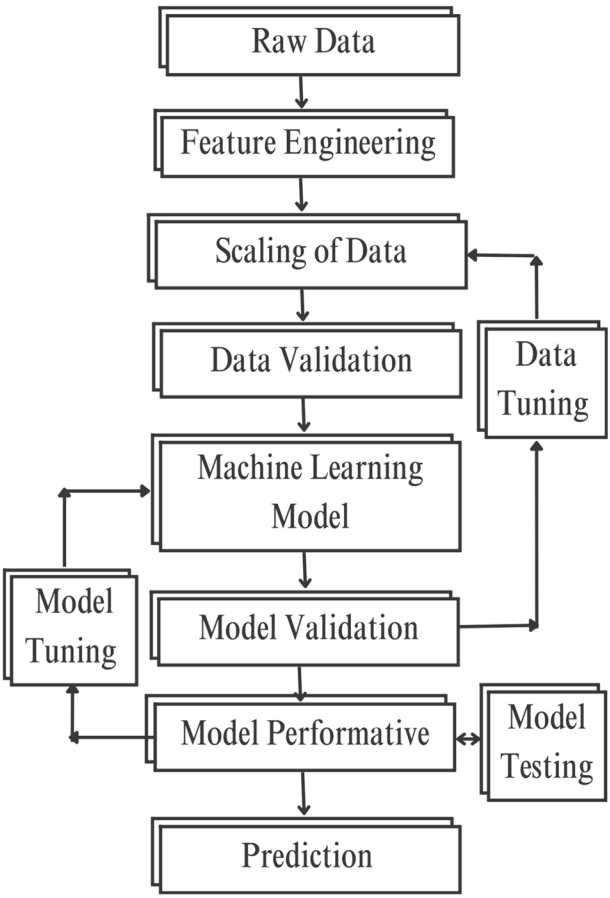


Fig. 1 Methodology Framework for Predictive Analytical

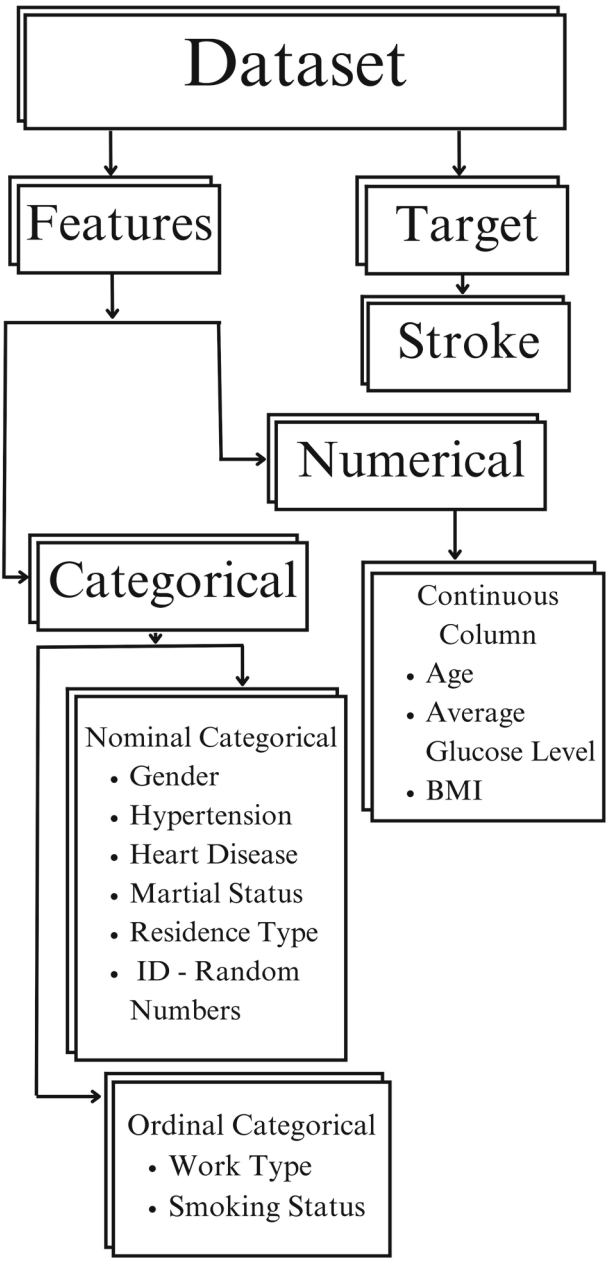


Fig. 2 Distribution of the Dataset

where the force of blood pushing against the walls of blood vessels remains elevated. This condition poses a significant risk to heart health, increasing the likelihood of heart attack, stroke, heart failure, and coronary artery disease. The data show that high blood pressure is more common in the elderly, regardless of gender, consistent with previous research. As people age, their arteries weaken and become more resistant to blood flow, which can lead

to high blood pressure. Other factors such as lifestyle, diet, and genetics can also influence the development of high blood pressure. The database also provides information about the types of jobs people hold, many of which are private. This information can be used to identify occupational health risks or develop response plans to improve the health and well-being of workers in certain industries. For example, some jobs require sitting for long periods of time, exposure to harmful chemicals or pollutants, or stress, all of which can increase the risk of high blood pressure and other health problems. By identifying these risks, interventions can be designed to improve job performance and reduce the risk of workers developing health problems.

4.1 Feature engineering and EDA

Before analyzing the effectiveness and performance of machine learning models, we should present, discuss and discuss the most important aspects of EDA (Data Analysis) and its importance. Our goal is to identify key features that contribute to a model's predictive accuracy [29].

'Objective': 'Binary-Logistic' Here the target is set to "binary: logistic" to predict the binary outcome of the stroke event, indicating that the model is trained to be a binary distribution using logistic regression. This means that the model is designed to determine whether the patient has had a stroke.

'Eval_metric': 'logloss' To gain a better understanding of the characteristics, the "wheel loss" measurement was selected. This method utilizes log loss to account for the possibility of misclassification. Specifically, binary cross-entropy loss (also known as logarithmic loss) is often applied in binary classification scenarios [30]. The loss log calculates the variance between the predicted and actual binary code, taking into account the repercussions of misclassification in order to penalize it.

The Log-Loss metric, also known as Logarithmic Loss or Binary Cross-Entropy, is a measure used to evaluate the performance of a classification model, particularly in binary classification problems. It quantifies the accuracy of the model's predicted probabilities. Here is a detailed breakdown of the formula and its components:

The general formula for **Log-Loss** metric is given by :

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[(y_i)\log(p_i) + (1 - y_i)\log(1 - p_i)] \tag{1}$$

Where,

- N Show all instances or instances in the file.
- y_i represents an actual binary tag (0 or 1) i^{th} instance.
- p_i Show estimated result i^{th} instance belonging to the positive class.

Detailed Description

Summation $(\sum_{i=1}^{N})$:

The formula sums over all N instances in the dataset. Each term inside the summation contributes to the overall log loss by calculating the error for each instance.

Actual Binary Label (y_i) y_i is the true label for the i-th instance. It can be either 0 or 1.

- If $y_i = 1$, the term $y_i \log(p_i)$ contributes to the Log-Loss.
- If $y_i = 0$, the term $(1 - y_i) \log(1 - p_i)$ contributes to the Log-Loss.

Predicted Probability (p_i) p_i is the model's predicted probability that the i-th instance belongs to the positive class (i.e., the event that $y_i = 1$).

- $\log(p_i)$ is the natural logarithm of the predicted probability for the positive class.
- $\log(1 - p_i)$ is the natural logarithm of the predicted probability for the negative class (i.e., the event that $y_i = 0$).

Terms Inside the Summation

- $y_i \log(p_i)$: This term is significant when the actual label y_i is 1. It penalizes the Log-Loss heavily if the predicted probability p_i is far from 1 (i.e., when $\log(p_i)$ is a large negative number).
- $(1 - y_i) \log(1 - p_i)$: This term is significant when the actual label y_i is 0. It penalizes the Log-Loss heavily if the predicted probability p_i is far from 0 (i.e., when $\log(1 - p_i)$ is a large negative number).

Negative Sign and Averaging

- The negative sign ensures that the Log-Loss is positive since the logarithm of a probability (which lies between 0 and 1) is always negative.
- The sum is divided by N to compute the average Log-Loss over all instances, providing a measure of the model's overall performance.

Interpretation

- A lower Log-Loss value indicates a better performing model, as it signifies that the predicted probabilities are close to the actual labels.
- A higher Log-Loss value indicates poor performance, meaning the predicted probabilities are far from the actual labels.

With reference to Fig. 3, several key observations are made:

- Age was positively correlated with stroke prediction (0.156), indicating that the incidence of stroke increases with age.
- Cardiovascular disease was positively correlated with stroke prediction (0.114), suggesting that individuals with a history of cardiovascular disease may be more likely to have a stroke.
- From the data provided, we can see that there is a 0.693 correlation between both age and marital status in the data predicting stroke. However, when looking at the relationship of each feature with the target line (stroke), Age has a lower correlation of 0.156 and Marriage 0.072. The difference in this correlation suggests that age may have a more direct relationship with marital status in predicting stroke. There is a positive correlation between age and stroke incidence with a correlation of 0.156. On the other hand, the relationship between marital status and stroke prediction is weak with a correlation of 0.072 [31].
- Measured job type correlates well with age (0.587); this is clearly seen as children (marked as -0 in the lowest category) do not go to work, while the average age of government employees does not go to work. The position (marked as the highest rank - 4) is 49. So this indicates a good relationship between the features. However, WorkType is less useful compared to Purpose (Stroke).

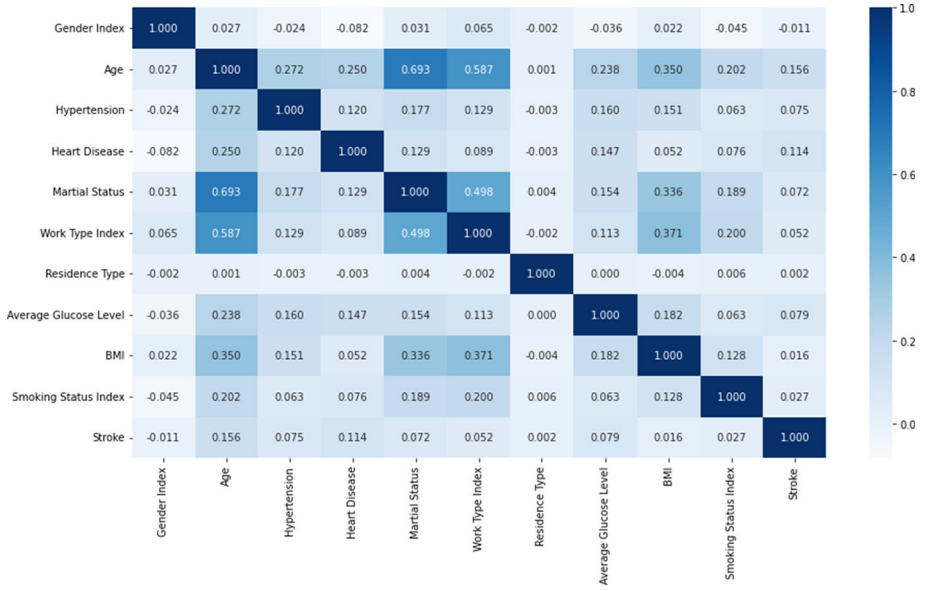


Fig. 3 Correlation Analysis

- Elevated blood pressure and mean blood glucose were weakly associated with predicting stroke (0.075 and 0.079, respectively). This means that people with high blood pressure or above-average blood sugar will have a slightly higher risk of having a stroke.

4.2 Data pre-processing

To properly understand information, preprocessing is a crucial step. After the data has undergone the exploratory data analysis (EDA) phase, it must be preprocessed to ensure that it is clean, original, and ready for analysis [32]. This process helps to identify any errors or inconsistencies in the data, making it easier to manage and analyze. The dataset is composed of both numeric and categorical data, which requires different approaches when analyzing. Numerical data can be measured and evaluated using mathematical models, while categorical data, such as frequency analysis, requires different methods. Column names have been changed and optimized for the specific model, making the data easier to identify and manage. The columns have been grouped into datasets, and the frequency of each category has been counted. This provides insight into the data distribution and helps identify patterns or trends (Figs. 4 and 5).

Null values in the "Smoking" column have been replaced with common groups (genres), which is a method of evaluating missing data in categorical variables (Table 1). Missing data in the equation can be corrected by finding the median or mean, or simply reporting the missing data. In this case, missing values in the "BMI" column were filled in with the median values as deleting data can result in missing or inconsistent data. Since the "Smoking Status" column consists of categorical data such as "non-smoker", "former smoker", and "smoker", it is not useful for data processing. Therefore, a critical step in cleaning and prioritizing data is making a good change. Keeping the order of the group seems like the right way to get to each column, so the column values "Smoking Status" - "No Smoking", "Smoking Type" and

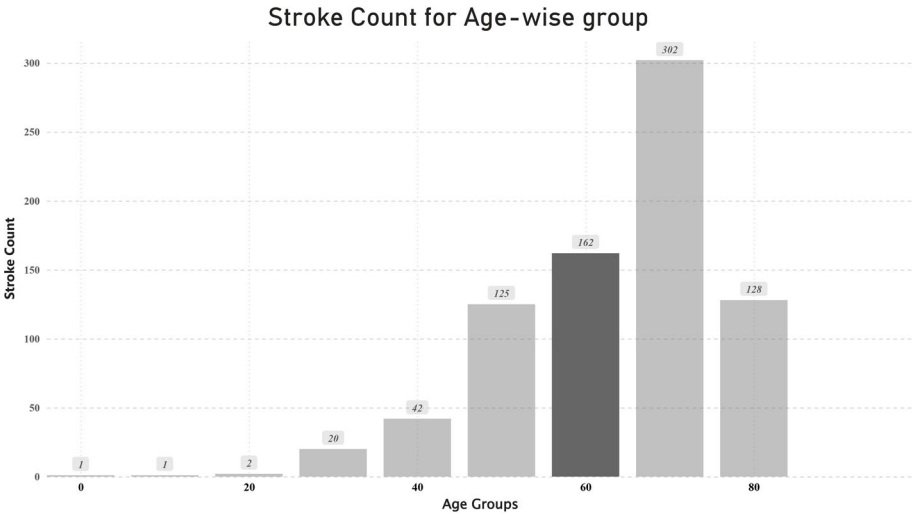


Fig. 4 Count of strokes by age groups

"Smoking" are coded as 0, 1, and 2, respectively. The same idea was used for the "Job Type" column, as it also includes the information in the list, from Children (list 0) to State Jobs (labelled 4), in ascending order. Some columns in the database represent true or false values such as "Blood Pressure", "Heart Blood", "Day Type" and "Army Force". These values will be converted to numbers - Boolean (0 or 1) [33]. The gender system consists of groups (male, female, etc.) marked (0, 1, 2) for ease of identification (Figs. 6 and 7).

5 Models

The aim of the problem is to predict the outcome of strokes, which is a binary column. Thus building distributive classification models is necessary. Furthermore, there are numerous characteristics and observations in the data, making it vital to select a model that can address these complicated issues. To assess the effectiveness of various data models, we have used three distinct classification techniques: Artificial Neural Network (ANN), Support Vector Machine (SVM), and K- Nearest neighbours (KNN). These algorithms have been selected

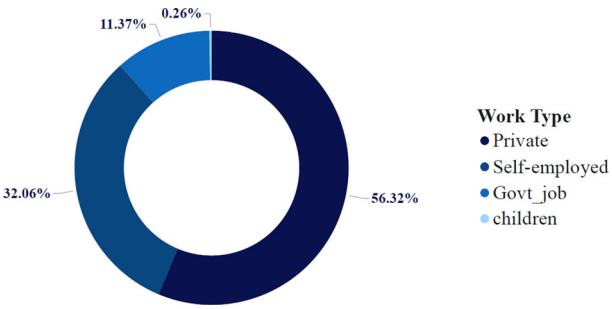


Fig. 5 Count of strokes by work type

Table 1 Age Distribution Table

SL. No.	Age group	Count	Population %
1	<=20	8511	19.611
2	20-40	10672	24.59
3	40-60	13090	30.161
4	60-80	9695	22.339
5	>=80	1432	3.2995

for their capability to handle extensive datasets and numerous features and for their excellent performance in distributing tasks. The effectiveness of these algorithms is evaluated using metrics like accuracy, errors, loss, ROC scores, etc. [34]. The results of the study show that all three algorithms have high accuracy and low efficiency, indicating that they are effective in predicting results from input devices. It was found that the ANN algorithm outperformed the other two algorithms by a small margin. It achieved higher accuracy and yielded lower results. Overall, this study demonstrates the importance of trying different models and the necessity of choosing a model that fits the specifications and data. The use of various algorithms and performance tests allows for a detailed evaluation of the working of all models and helps determine the best way to handle the task at hand.

5.1 Using neural networks(ANN)

Regarding the use of Neural Networks in previous articles [35, 36], to understand the presence of strokes, we have applied Artificial Neural Networks (ANN). The model architecture has been shown in Fig. 8.

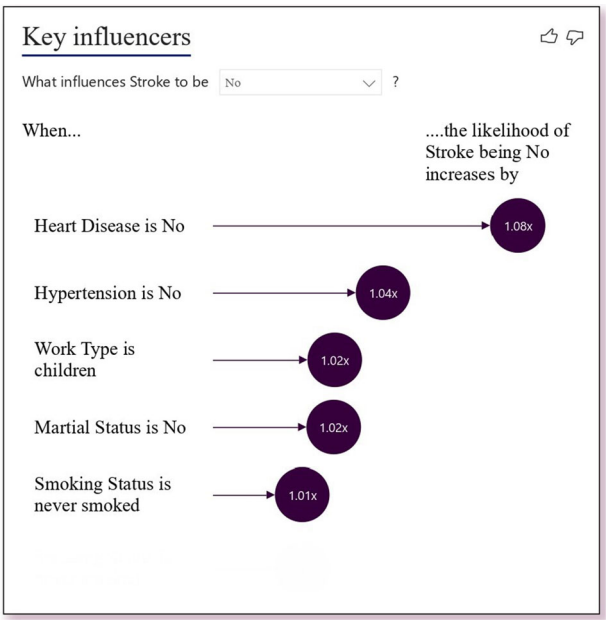


Fig. 6 Key influencer is no

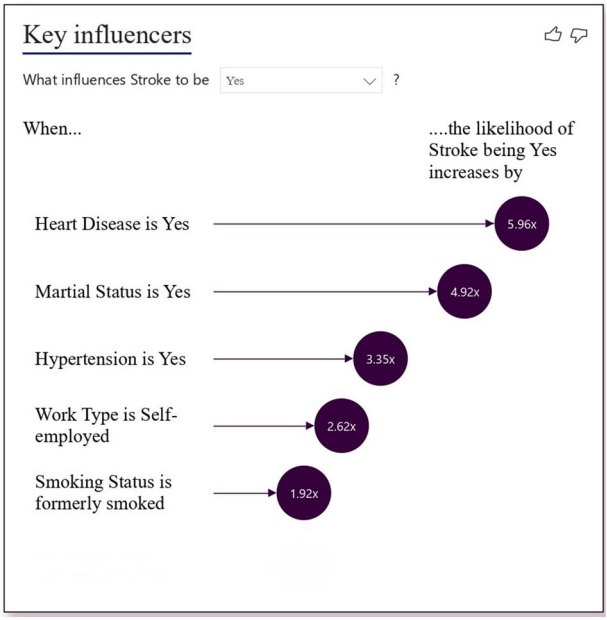


Fig. 7 Key influencer is yes

- We have created a Sequential class, which creates models layer by layer. This is to initialize our model object.
- The first layer is the input layer. There are 10 input nodes representing the 10 features used in the model.
- The second and third layers are the hidden layers, which have 16 neurons each. The activation function used by these layers is the ReLU activation function. Its advantage is that it performs well in computation and avoids the problem of gradient loss. ReLU provides the benefit of providing non-linearity in the model layers.
- In the output layer, the sigmoid function is used in predicting the class of Strokes. The single output layer is based on a binary distributed validation dataset Output layer uses a sigmoid function, which produces a binary output (0 or 1) based on the input.
- Whenever any errors are discovered, these errors are backpropagated from the output layers to input neurons. The weights and biases are updated and refined to reduce the error/ loss. This is known as **Back Propagation Algorithm (BPA)**.

Dropout is a powerful method that can help prevent problems in deep learning. The idea behind the release is to simulate the results of training a networking group with different architectures [37]. Each time a network is freed and trained, different groups of neurons are lost and additional neurons must learn to compensate for their absence. By doing this, the network will be able to acquire more resilient features that are not dependent on particular neurons.

The **Early Stopping**, as shown in Fig. 9, is an exit method that effectively prevents the integration of neurons in the network. This will stop the training process when the performance of the model goes down. This regularization technique prevents overfitting and ensures the model can continue training on generalized data without a drop in performance.

Weight Updation -

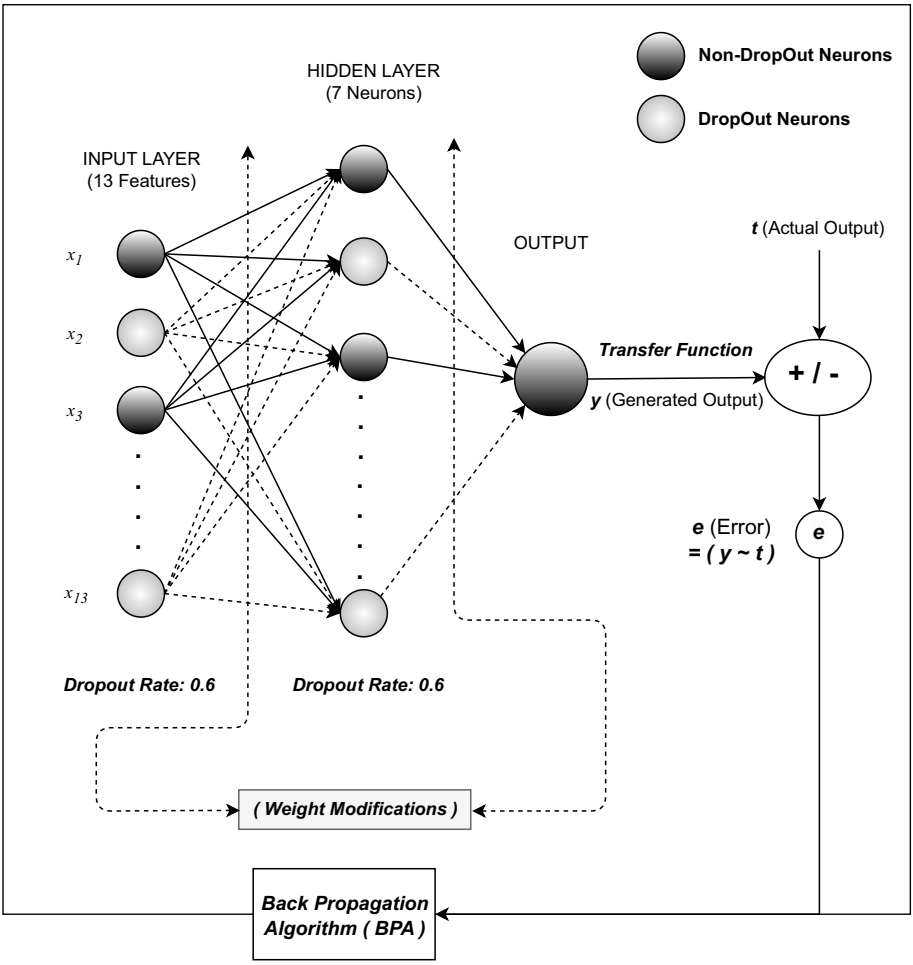


Fig. 8 Neural Network Architecture

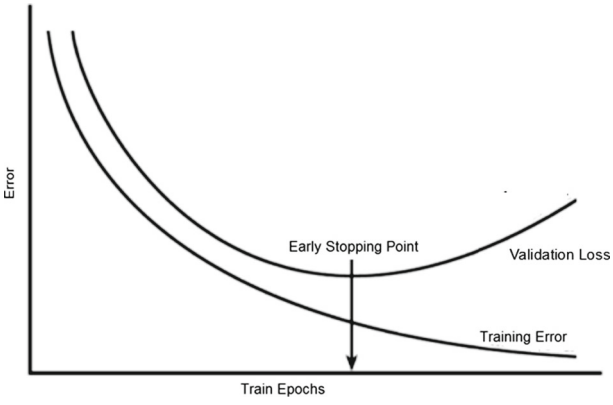


Fig. 9 Early Stopping

Algorithm 1 Updation of Weights in Output Layer.

Input: We are using Gradient Descent for the weight updating of the output layer, which is represented as $(m+1)^{th}$, as it is considered as the base weight.

$$\Rightarrow w_{kj}(m+1) = w_{kj}(m) - n\frac{\partial(W(m))}{\partial w_{kj}(m)} \tag{2}$$

1. To derive the update equation for the weight of the output layer, we used the chain rule.

$$\Rightarrow \frac{\partial E(W(m))}{\partial w^{\circ}_{kj}(m)} = \frac{\partial(W(m))}{\partial e^{\circ}_{k}(m)} \frac{\partial e^{\circ}_{k}(m)}{\partial y^{\circ}_{k}(m)} \frac{\partial y^{\circ}_{k}(m)}{\partial x^{\circ}_{k}(m)} \frac{\partial x^{\circ}_{k}(m)}{\partial w^{\circ}_{kj}(m)}$$

2. The derivative of the activation function of the output layer neuron in relation to its activation

$$\Rightarrow \frac{\partial y^{\circ}_{k}(m)}{\partial x^{\circ}_{k}(m)} = \frac{\partial f(x^{\circ}_{k}(m))}{\partial x^{\circ}_{k}(m)} = f\prime(\partial x^{\circ}_{k}(m)) \tag{3}$$

3. where,

$$\Rightarrow x^{\circ}_{k}(m) = \sum_{j=0}^{J} w_{kj}(m) y^{h}_{j}(m)$$

J is the total number of hidden neurons connected to the k^{th} output neuron. The summation interval includes the bias input ($j = 0$).
4. To calculate the activation value of the k^{th} unit in the output layer about the synaptic weight of the j^{th} unit in the hidden layer, The input from that same unit must be taken into account.

$$\Rightarrow \frac{\partial(x^{\circ}_{k}(m))}{\partial(w^{\circ}_{kj}(m))} = y^{\circ}_{k}(m)$$

5. Therefore

$$\Rightarrow \frac{\partial E(W(m))}{\prime(x^{\circ}_{kj}(m))} = [t_k(m) - y^{\circ}_{k}(m)] f\prime((x^{\circ}_{k}(m))[(y^{\circ}_{k}(m))]$$

6. By changing the weights

$$\Rightarrow \Delta w_{kj}(m) = -\eta \frac{\partial E(W(m))}{\partial(w^{\circ}_{kj}(m))} = \eta \delta^{\circ}_{k}(m) y^{h}_{j}(m) \tag{4}$$

Output: In the weight update process of a neural network, the weight value is adjusted based on the product of three key factors:

- The learning parameter (often denoted as η).
- The local error gradient of the output neuron (represented as $\delta^{\circ}_{k}(m)$).
- And the input signal from the corresponding hidden neuron (denoted as $y^{h}_{j}(m)$).

Finally, the equation of the updated weight of the output layers

$$\Rightarrow w_{kj}(m+1) = w_{kj}(m) + \Delta w_{kj}(m)$$

which is equal to,

$$w_{kj}(m) + \eta \delta^{\circ}_{k}(m) y^{h}_{j}(m) \tag{5}$$

We have analyzed how error minimization is done for weights by updation and BackPropagation (BPA), using **Gradient Descent Algorithm** (as mentioned in Algo. 1), Calculus and the application of the chain rule are used to compute the gradients of the cost/ loss function

with respect to the network parameters (weights and biases). Iteratively (Step-by-Step), the weights are modified for better performance.
where,

- m is the step count of iteration.
- k is the outer layer, which would give us the results
- j is the hidden layer. (Consider a single hidden layer for understanding)
- $w^{\circ}_{kj}(m)$ is the weight updated from j^{th} hidden neuron to be used in k^{th} neuron present in outer layer. This is the m^{th} iteration.
- η is the learning rate, to modify the weights
- $W(m)$ is the set of all weights, required to train the neurons, at the m^{th} iteration.
- $\delta^{\circ}_{k}(m)$ is the error rate of k^{th} neuron
- $y^{h}_{j}(m)$ is the output given by activation function, from j^{th} hidden neuron.
- $x^{\circ}_{k}(m)$ is the activation value of k^{th} output neuron

If we wish to represent an ANN model, which will use Sigmoid Function as activation in all the hidden layers, the mathematics behind the approach is as follows.

Initially, the output of the first hidden layer, with the initial properties x_i and w_i for the same neuron, is given as:

$$a_i = \frac{1}{1 + e^{-(x_1 w_{1i} + x_2 w_{2i} + ... + x_n w_{ni})}} \tag{6}$$

where,

- $i = [1, n]$ representing the n features, in the dataset.
- w_{ij} represents the weight given to the x^{th}_i feature, which passes through j^{th} neuron of the first hidden layer.

The output produced by each neuron in the first hidden layer (labelled a_i), is fed forwarded as the input for the next hidden layer, with new weights. This step will be repeated for all these processes. So, if (a_1, a_2, ..., a_16) are the outputs of all the neurons present in the first hidden layer, the equation for the input to a single neuron, of the second hidden layer will be: (7)

$$a_{17} = \frac{1}{1 + e^{-(a_1 w_{1,17} + a_2 w_{2,17} + ... + a_{16} w_{16,17})}} \tag{7}$$

Here, $w_{i,17}$ represents the output from a_i influenced to the first neuron, in the second layer, which is marked as the 17^{th} neuron (considered for easier counting). Similarly, the final output through the Output neuron is calculated, mentioned in (8), before being passed through the transfer function.

$$\Rightarrow y = \frac{1}{1 + e^{-(a_j w_{j,F} + a_{(j+1)} w_{(j+1),F} + ...)}} \tag{8}$$

Where j represents the count of the number of neurons in the final hidden layer, and F represents the final output neuron.

$$e^2 \Rightarrow (t - y)^2 = (t - (\vec{w}.\vec{a}))^2$$

$$\Rightarrow ErrorRate(E) = -\frac{\delta e^2}{\delta w} = 0$$

$$\Rightarrow E = -\frac{\delta}{\delta w}(t - \vec{w}.\vec{x})^2 = 0 \tag{9}$$

The main objective lies in minimising the error rate, as mentioned in (9). Weights ($\vec{w}$), and Biases ($\vec{b}$) (if present) are responsible for the error, as ($\vec{x}$), which represent the features of the dataset, are fixed.

5.2 Using K-nearest neighbors (KNN)

When it comes to classification, the primary method used involves predicting the target label of a new data point by considering the majority class of its 'k' nearest neighbours in the training set. When we want to find out which category the unknown data belongs to, we calculate the distance to other points and then select the point with the shortest distance according to the k value and choose which category the data belongs to. Cross-validation is used to divide the data into k groups [38]. Choosing an appropriate value of k is an absolute necessity for giving the best classifications of the unseen data. Refer to Section 6, and (14), for our approach to finding the optimal value of k.

The choice of an appropriate value of k is essential as it can affect the performance of the model. This has been done for the following reasons:

- To generalize the model, so it performs well for the unseen data too.
- To control the Bias-Variance trade-off, in the model.
- To find a balance between overfitting, and underfitting.
- To cross-validate the data in some scenarios, to provide a reliable estimation by dividing the data into various folds.

Cross-validation entails dividing the training data into k subsets (known as folds) as shown in the diagram of Fig. 10. The model is then trained on $(k - x)$ folds and validated on the remaining x fold(s), where x is the number of folds kept to validate the training data. This process is repeated k times while using the test data only once. The outcomes are then averaged to give a more precise estimate of the model's performance. During cross-validation of KNN, the KNN algorithm is used on each node.

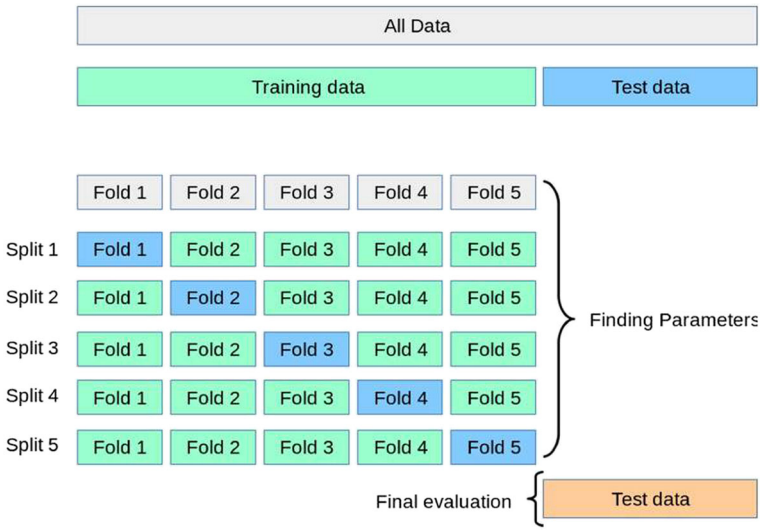


Fig. 10 K-Fold Cross Validation

Figure 10, specifies that the performance measure, that is reported for *k*-fold cross-validation, is done by folding the training set $k = 5$ times. This approach can be expensive, but it doesn't waste a lot of data (like when correcting illegal use), which is better for problems such as reverse inference where it's too small for example.

There are various types of cross-validation techniques such as k-fold cross-validation, layered *k*-fold cross-validation, and one-out cross-validation. The choice of cross-validation method depends on the nature of the data and the problem statement. CV (cross-validation) is used to evaluate the effectiveness of machine learning models, it is also an iterative technique to evaluate models with limited data. Measures should be chosen and optimized to make the number of neighbours, *k* in KNN, without using the test method, there will be over-stretching if optimists and observers are in the same dataset, so validation is most useful [39]. Here we are referring to the process with λ. The algorithm should not be fixed during cross-validation. To train the algorithm of the training process, in the training process λ will be similar to, the meaning in the data is considered as the population of unknown data, and we will find the best training for the prediction using the y class in the dataset.

Voronoi diagrams in KNN The Voronoi diagram is a mathematical tool used to divide a given area into a zone. These regions are defined as "seed points" (also called "generators" or "sites") so that all points in a region are closer to that region than other seed points. In the context of the nearest neighbour (K-NN) algorithm, the Voronoi diagram can be used to speed up the nearest neighbour search.

For the given data (population), identify the training methods and record the training points for the different units found in all the data. All classes are guaranteed with the help of competitive verification drawings Fig. 11. Show how training materials are placed. In this case, new invisible data (from test data) is taken (□) and accordingly, we find the closest, closest and best training example of class A (▲) or class B (●). K-neighbours for Training The proximity of unknown, invisible data points in the neighbourhoods (KNN) algorithm depends on the distance measure used and the selected neighbour (k value). The domain name for class A(▲) or class B(●) depending on the content of data given to it is closer to space respectively, as in Fig. 11. Specify the registered domain with the location (▲); this means that the new data (□) is closest to that class labelled P (▲). Similarly, the closest of all input points to the missing data can be obtained using the distance metric [40]. Distance

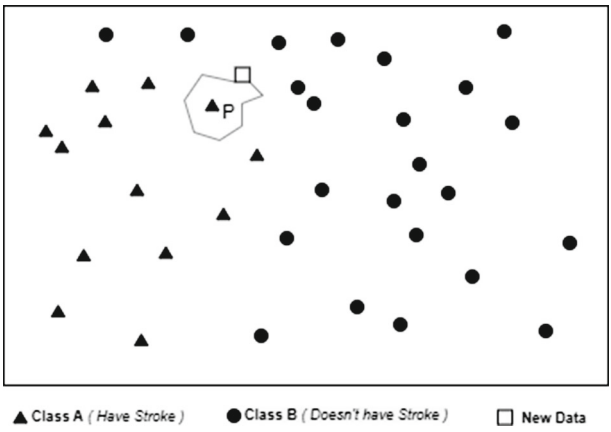


Fig. 11 Training Feature Space

measures the similarity or difference between two points. Various distance measures can be used in KNNs, based upon the problem statement, and the necessity, which includes Euclidean distance, Manhattan distance, Hamming distance, and Cosine distance [41]. The choice of distance measure affects the performance of the KNN algorithm, as some measurements may be better for some data types. The distance metric used here is the **Euclidean distance** derived from (10).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{10}$$

In (10), d is defined as the distance between two X and Y points with coordinates $X(x_1, y_1)$ and $Y(x_2, y_2)$ respectively. Other distance measures can also be used. The value of k determines the number of closest neighbours that should be taken into account when making predictions for unfamiliar data. Smaller k values will result in a more local estimate, considering only a few nearest neighbours, while larger k values will result in a more general estimate considering many neighbours [42]. The choice of the k value also affects the class prediction. Therefore, by representing all incoming data, we define these objects as a Voronoi mosaic as shown in Fig. 1 below Fig. 12.

Voronoi tessellation, also known as a Voronoi diagram, is a mathematical concept that divides space into a group of cells, where each cell is associated with a particular point or seed (called a Voronoi region or generator). The collection of all these cells is called Voronoi tessellation [43]. Voronoi subdivisions are used to define the boundaries of classes. Each Voronoi cell is associated with a unique class list that corresponds to the most common [42, 43] class of points shown in that cell. This boundary decision (marked in bold) divides data into two classes: class A (▲) (stroke patients) and class B (●) (people without stroke). The content of the border decision is equal to that of two different neighbours. This is because a boundary is defined as a point in the feature space that is closer to one neighbour than another. Therefore, they can be classified as A or B.

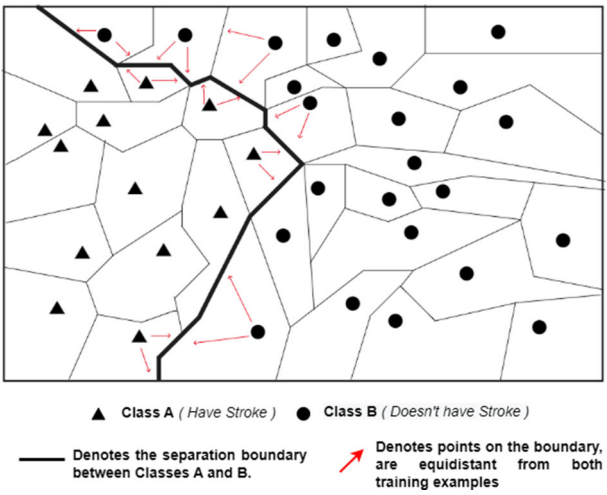


Fig. 12 Voronoi Tessellation

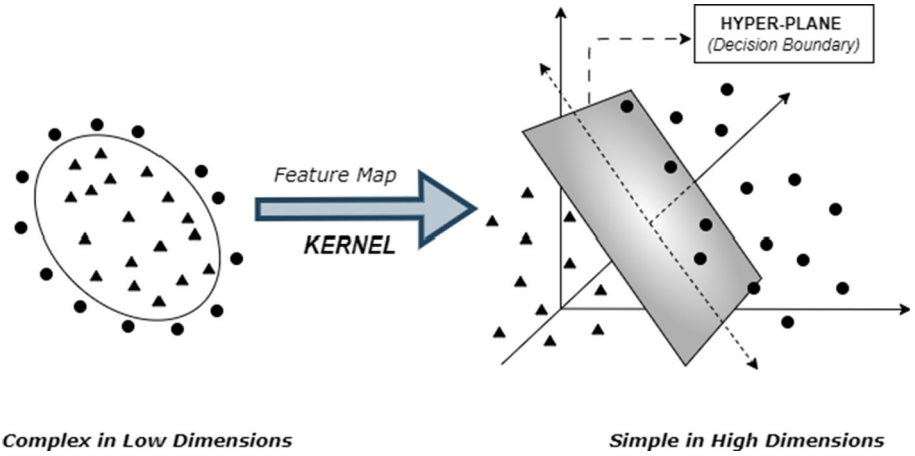


Fig. 13 Separating Hyperplane to ease classification

5.3 Support vector machine (SVM)

The objective of SVM is to create an optimal line that divides n number of dimensional spaces of different classes so that the data is in the correct class in the future. SVMs use a technique called "kernel trick" that allows them to transform non-linearly separable data into linearly separable data [44, 45] (Fig. 13).

Training, validation and testing of data is done on the scaled data. The kernel function is used to solve nonlinear problems using linear classifiers. It takes the information as input and then converts it into a desired form. The core function types are linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid [46]. When we need a transform but do not have any prior knowledge of the data, we use the RBF core function, which uses a radial operation to improve the transform. The sigmoid kernel function is equivalent to the two-layer perceptron model of a neural network [46]. Polynomial kernels are often used in SVM classification problems where the data is not linearly separated but can sometimes find a hyperplane that separates the groups by mapping the data to a higher domain. Polynomial kernels [47] help in representing the polynomial as vectors on the training data which is used in the kernel of original variables of feature space, all of this is done by the polynomial function that maps the data and transforms it into a more dimensional space [47]. It maps data to high-dimensional space using polynomial functions. Another parameter is the coefficients of the polynomial, which determine the effect of higher-order terms in the polynomial. By adjusting these parameters, the polynomial kernel can be adapted to the specific problem at hand and helps improve the accuracy of the SVM classification [47].

The kernel for polynomial degrees is defined as:

$$K(x_1, x_2) = (x_1^T x_2 + c)^d \tag{11}$$

Where,

- c is defined as the constant
- x_1, x_2 are vectors in original Space.

The c parameter used in support vector machines (SVMs), controls the balance between sample accuracy and margin size. A higher c value will reduce the training error by making

the model fit the training data more but will make the data worse, i.e. not good for new data. Conversely, a small value of c will allow larger margins and larger samples but will result in more training or poor performance [48]. In a polynomial SVM, the degree d of the polynomial is another parameter [49] that controls the complexity of the model. Higher-order polynomials fit the data more and lead to overfitting, while lower polynomials lead to simple models that do not capture the full complexity of the data, leading to underfitting.

6 Results and analysis

To understand the quality and performance of these models, we utilized two commonly used evaluation metrics: ROC curves and confusion matrices. The ROC curve is a visual representation of how well a binary classification model performs. It shows the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different threshold levels. The curve makes the trade-off between sensitivity (or Recall), and specificity visible and enables us to evaluate how well the model performs at all potential thresholds [50]. By examining the ROC curve, one can determine how well the model can distinguish between positive and negative classes. A model's success is determined by how closely its curve follows the plot's upper-left corner, indicating high sensitivity (accurate detection of positive occurrences) and low false positive rate (accurate rejection of negative entities). For example, concerning ROC interpretation, we have shown, how the ANN algorithm works on this prediction dataset, along with the ROC interpretation, as shown in Fig. 19. ROC interpretations have been done in other models too. On the other hand, a model with a curve that follows the diagonal line represents a random guess or a poorly performed model.

We have also used confusion matrices as quality matrices for our prediction. A confusion matrix is nothing but a statistical report of how accurately my predictions match the actual data or values. A model's performance can be determined by assessing various statistical metrics like Accuracy, precision, Recall (Sensitivity), F1 Score, Log-Loss, etc. These measurements provide valuable insights to classify the data based on the designed model and handle both True Positive Rate (TPR) and False Positive Rate (FPR). By utilizing ROC curves and confusion matrices, we thoroughly evaluated the classification models in terms of their overall performance, sensitivity, specificity, and other important metrics. These evaluation techniques provide a robust analysis of the model's quality and aid in the comparison of different models or approaches [50]. The plots have two parameters:

- **True Positive Rate (TPR) -**

$$TPR = \frac{TP}{TP + FN} \tag{12}$$

- **False Positive Rate (FPR) -**

$$FPR = \frac{FP}{FP + TN} \tag{13}$$

KNN Result analysis KNN is an ML algorithm that classifies data points by assigning class labels based on the majority class of its k nearest neighbours in the feature space. The value of k determines the number of neighbours considered, for which the new unseen data would go to which class. A higher value of k considers a larger number of neighbours, while a lower value of k considers a smaller number of neighbours. On the cleaned dataset, our detailed analysis has been done. We have used the $StandardScaler()$ method, which is in $sklearn.preprocessing$, to scale the data. Scaling the data is important in terms of applying

classification in KNN. KNN, being a distance-based algorithm, the classification of the data points is heavily dependent on the computation of distances. Distances can be very high, or very small depending on the features. Thus a biased result is very much prone to be developed before we predict the outcome of the model. Thus $StandardScaler()$ method is used which manipulates the data, such as Mean Value (μ) = 0, and Standard Deviation (σ) = 1. $fit\_transform$, then has been used on the features. This will ensure that the KNN model is trained on the scaled features of the dataset, and the test data will also be scaled to get optimal predictions. We took a range of k i.e. (number of neighbours), to understand and predict the outcome, more accurately. Choosing various k values, in KNN is a part of hyperparameter tuning or model selection. It is better to choose k as odd numbers, because if k, would have been even, there lies a chance that for a particular unseen data, 50% of the neighbours go to Class 0, and the rest to Class 1. So there will be a problem in identifying in which Class, would the data belong.

The optimal value of k, is given by the formula (14) :

$$k = \sqrt{N} \tag{14}$$

where, N represents the size of the sample data, used for training the model.

Our training dataset size has 70% of the total data. Thus, the number of rows for training is 30380, i.e. N = 30380. So optimal value of k = 174.298, so we would take $k(approx.)$ = 173, as k must be odd, to avoid any tie between binary classification. Thus the range of k, that has been used is all the odd numbers in between 3 and 173.

Therefore, the KNN Analysis has been done in two ways:

(i) Without Cross Validation -

The KNN model here has been trained and evaluated without doing any cross-validation. The machine's performance, along with the ROC-AUC score, has been used to determine the efficient working of KNN.

Analysis-

The KNN algorithm, which has been used to classify the occurrence of Strokes, is performing well throughout the various values of k. The model has performed well, gaining a maximum accuracy of up to 98.1951, for k = 11. After that accuracy remains constant for the rest of the k values. Predicting the optimal k value has been described further. The reason for taking large k values lies in determining the ROC-AUC Score. The best ROC-AUC Score obtained is 0.8098 (Fig. 20-II) for k = 169. For a lower k, such as k = 11, we got ROC-AUC Score = 0.635, shown in Fig. 20-I. This tells us that we are getting better results, when we take the number of neighbours, closer to an optimal value, as mentioned in (14). A higher ROC-Score (for k = 169) means that the classifier is distinguishing the class points (0s and 1s), more nearly correctly, compared to ROC-Score, for k = 11. A lower k value may be more sensitive to noise. Also, a smaller k value gives a more complex decision boundary, whereas, on the other hand, a larger k gives a smoother and simpler decision boundary. A simpler decision boundary generalizes the unseen data well. This is depicted in Fig. 20.

(ii) With 5-Fold Cross Validation

In K-Nearest Neighbours (KNN), cross-validation is a crucial technique used for analysis and evaluation with hyperparameter adjustment. Detailed descriptions can be found in Section 5.2 and Fig. 10. By averaging the findings across numerous folds of the data, this strategy aids in providing better results. For each fold of the data, the accuracy, ROC-AUC score, model fit time, and scoring time are calculated and averaged across the 5 folds. The

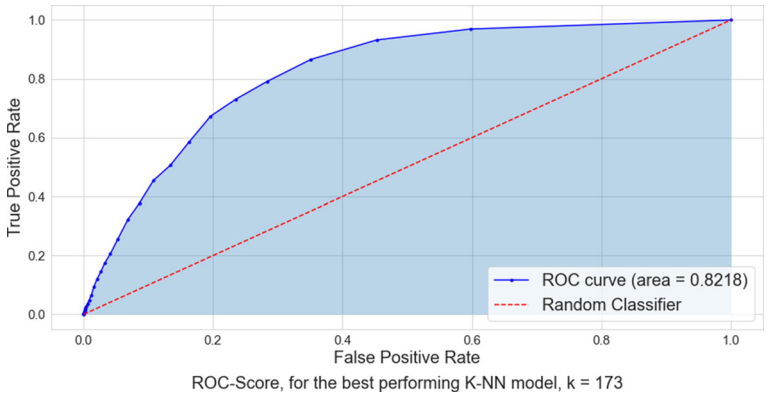


Fig. 14 ROC-AUC curve after cross validation

choice of optimal k in this scenario is similar to the previous situation, where we didn't perform any cross-validation (Fig. 14).

Analysis

Accuracy in this scenario has been refined to a very minute extent. After 5-fold cross-validation, the maximum accuracy that is generated is 98.19585, which is obtained first for $k = 11$. After that, the accuracy remains constant throughout, somewhat similar to the previous case.

In this analysis, the goal is to use the KNN algorithm to predict strokes. To determine the hyperparameter optimal value of the k, we analyze the ROC-AUC curve. KNN model performs well in distinguishing between the "Stroke" and "No Stroke" classes, with the best ROC-AUC score being approximately 0.8218 when $k = 173$. The ROC-AUC metric evaluates the model's ability to differentiate between these classes at various probability thresholds. We use 5-fold cross-validation, dividing the dataset into five subsets, to train and test the model iteratively, providing a more robust evaluation. The chosen value of k is 173, which balances model complexity and performance, making the KNN algorithm an effective predictor based on the given dataset and features for stroke occurrences (Fig. 15).

Final intuitions Thus, we can say that Accuracy cannot always be an optimal metric while predicting the results. Clearly, for our KNN model, ROC-AUC Curve Analysis is one of the better choices for interpreting the prediction of the classes. No distinct comparison can be drawn, by seeing the accuracy plots of both the KNN scenarios, shown in Fig. 15 (I) and (II). The ROC Analysis as in Fig. 16, shows that 5-fold cross-validation predicts the true and false classes, more accurately compared to the KNN model without any cross-validation. Our analysis also shows the optimal number of neighbours, i.e. k in both scenarios and how we have achieved our results.

ANN result analysis To analyse the ANN model, we first conducted an initial EDA and then implemented the model using cleaned data. Our evaluation of the ANN model architecture consisted of three layers: two hidden layers with 16 neurons each and one output layer which uses the Sigmoid Activation Function. ReLU was used for the hidden layers, resulting in an architecture of 16-10-10-1, as shown in Fig. 8. We added dropout layers with a dropout rate of 0.2 after each hidden layer to prevent overfitting. The decision to select 16 neurons in hidden layers, was grounded in the generalization of data. Hyperparameter tuning was

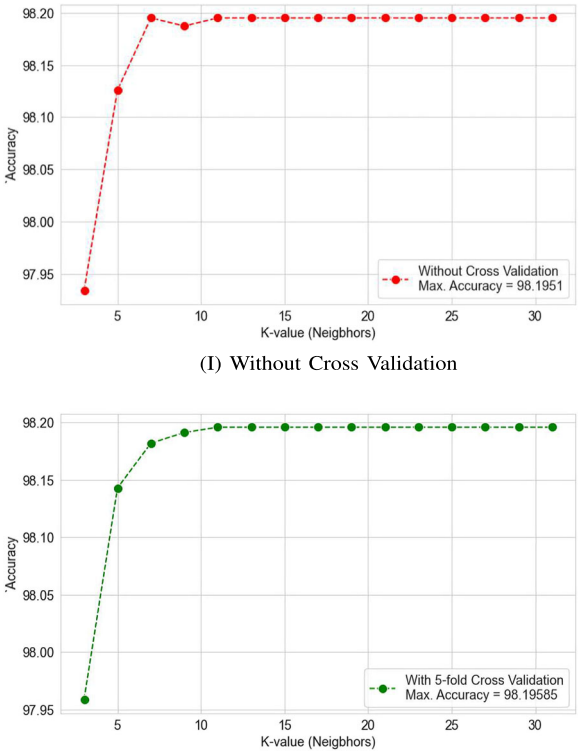


Fig. 15 Accuracy comparison of KNN with and without cross-validation

continuously fed by continuous seeding of several neurons. This allows us to minimize the overfitting of the model. The optimization of the Neural Network Architecture is heavily dependent on finding the number of neurons in a layer. The model was compiled using the Stochastic Gradient Descent Optimizer, along with binary cross-entropy loss function was used for updating the weights, to reduce the errors.

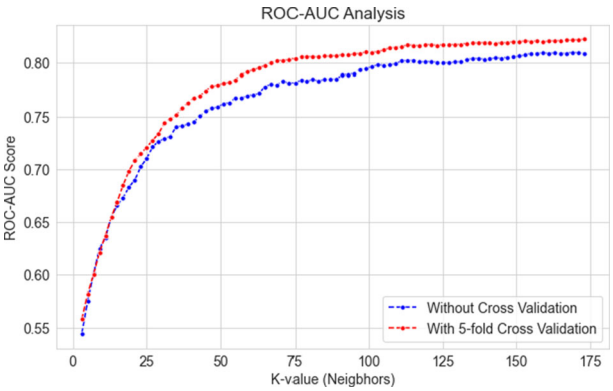


Fig. 16 ROC-AUC Comparision of KNN-Model

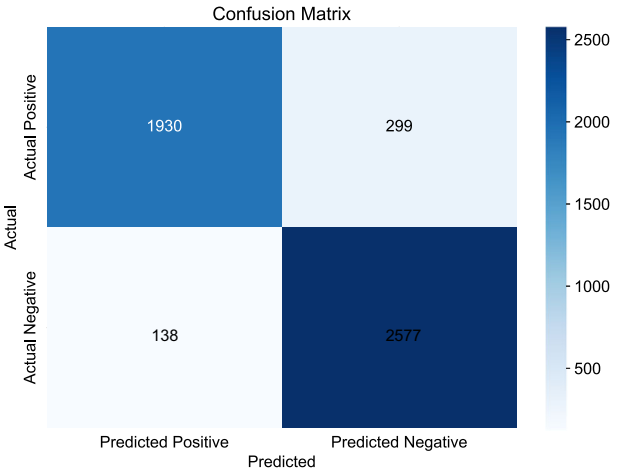


Fig. 17 Confusion Matrix

Confusion matrix Figure 17 shows the confusion matrix, which shows that the model is performing well at classifying the negative data points, but it is not performing well in the case of positive data points. this is because of the imbalance in the negative and positive data points, which means the presence of negative data is greater than the positive data points

True positives (TP): 1938 numbers and data points indicate that the model correctly classified as positive.

False positives (FP): 299 data points indicate that the model incorrectly classified as positive.

True negatives (TN): 2577 numbers and data points indicate that the model correctly classified as negative.

False negatives (FN): 200 data points indicate that the model was incorrectly classified as negative.

Some analysis of the confusion matrix Accuracy: This is the overall percentage of correct predictions calculated by adding the true positives and true negatives and then dividing by the total number of predictions.

$$\frac{(1938 + 2577)}{(1938 + 299 + 200 + 2577)} = 0.89 \sim 89\%. \tag{15}$$

Precision: This is the proportion of correct positive predictions calculated by dividing the number of true positives by the total number of positive predictions (true positives + false positives).

$$\frac{1938}{(1938 + 299)} = 0.86 \sim 86\%. \tag{16}$$

Recall: This is the proportion of actual positive cases that were correctly identified by the model. It is calculated by dividing the number of true positives by the total number of actual positive cases (true positives + false negatives).

$$\frac{1938}{(1938 + 200)} = 0.90 \sim 90\%. \tag{17}$$

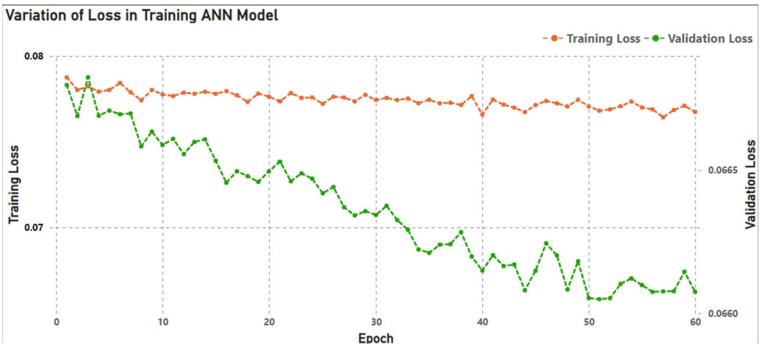


Fig. 18 Training and Validation Loss over the Epochs

Training Loss and total Validation Loss are positively correlated with each other. Training Loss and Validation Loss diverged the most when the Epoch was 1 and when Training Loss was 0.01 higher than Validation Loss. The model is thus successfully generalizing to new data if the validation loss declines more quickly than the training loss. This is typically a good sign because it shows that the model is successfully applying patterns it has learned from the training data to the validation data. Referring to Fig. 18, While a faster decline in the validation loss relative to the training loss is typically a sign of progress, it's crucial to watch how both losses behave over time. Referring to Fig. 18, testing recall and precision should ideally maintain levels comparable to those observed in training and validation. Any potential or significant deviation may indicate overfitting (high training, low testing) while maintaining good accuracy and loss in training. One can modify the model architecture or use extra regularisation approaches to avoid overfitting and improve generalisation in such circumstances.

Certain misclassifications were also generated through our confusion matrix. 437(299+138) data points resulted in predicting the Opposite class concerning the Actual Class. This contributed around 8.82% of the total dataset. This explains the error rate or the scheduled time of providing wrong predictions. Certain target adjustments like refining the model's training data, adjusting algorithms or feature engineering techniques are enabled to tune the model's performance.

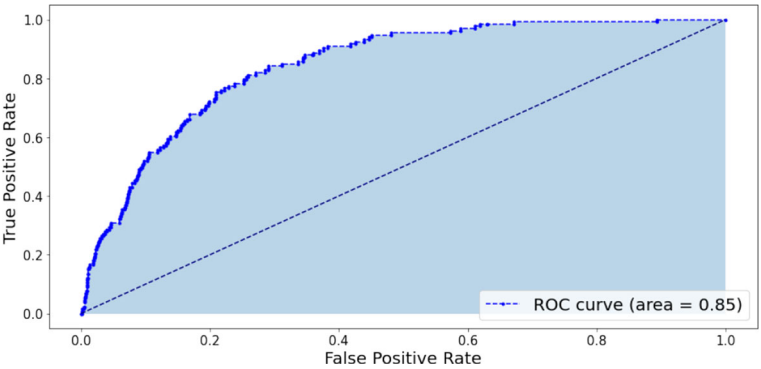


Fig. 19 ROC curve of ANN

Table 2 ANN Model Analysis

OUTCOMES	Loss	Accuracy
Training	0.0765	0.9813
Validation	0.066	0.9847

The consistency in accuracy scores across epochs suggests that the model reached a stable state and did not overfit the training data. This stability indicates that the model learned robust and generalizable patterns, allowing it to make accurate predictions on both the training and validation datasets. From Fig. 19 the model achieved an AUC (Area Under the Curve) score of 0.8451737. The AUC score is a common evaluation metric used in binary classification tasks, such as predicting the severity of road accidents. An AUC-score ϵ [0, 1], with a higher value indicating more precise and correct estimation. In this case, the obtained AUC score of 0.8451737 (approx. 0.85) suggests that the model has a good ability to distinguish between positive and negative instances, or in the context of Brain-Stroke prediction.

A graphical depiction of the true positive rate (TPR) against the false positive rate (FPR) at different categorization thresholds is denoted by the ROC curve, as shown in Fig. 19. This exhibits the balance between the sensitivity and specificity of the classification model [50]. For example, in a medical diagnostic scenario, where identifying a disease may have serious implications, a high AUC value would be crucial to minimize false positives. On the other hand, in a different domain, the importance of false positives and false negatives may vary. Therefore, while AUC provides a useful overall measure of model performance, it is essential to complement it with other metrics and consider the specific application context when interpreting the results and making decisions based on the model's predictions.

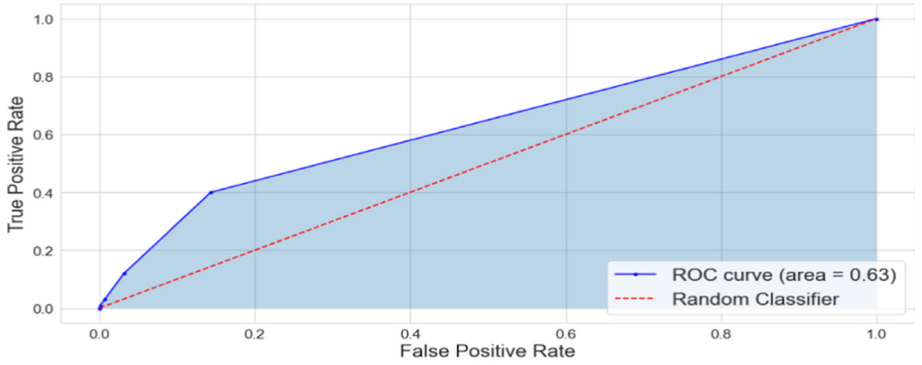
Through analysis, as displayed in Table 2 and Fig. 19, we have been able to track the progress of the model during training. Ideally, we aim to observe a declining trend in both loss and validation loss, as this indicates that the model is effectively learning and generalizing. Furthermore, we aim for high accuracy and validation accuracy levels that either increase or stabilize, which indicates improved predictive performance. By evaluating these metrics, necessary modifications can be made to the model or its training parameters to achieve optimal performance.

SVM Result Analysis We trained SVM models with all the known kernels including Linear, RBF (Radial Basis Function), Polynomial, and Sigmoid kernels, and worked on the needful results such as accuracy, Log-loss, and ROC AUC scores. The following are the results obtained: From the results of Table 3, several conclusions can be drawn:

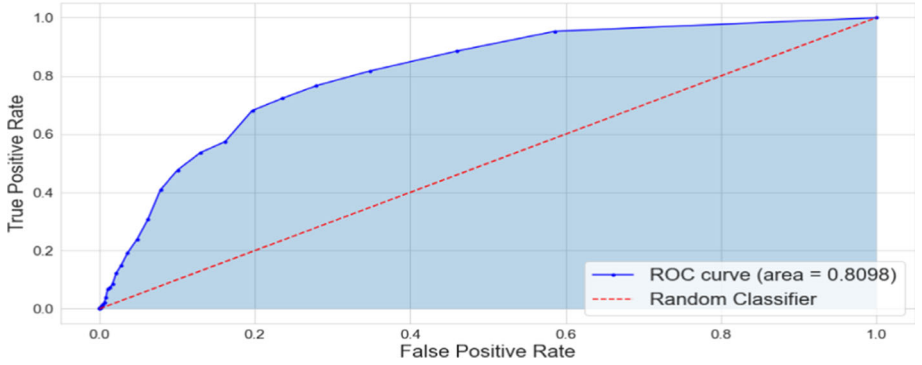
Accuracy: All SVM models achieved high accuracy, exceeding 96%, which indicates that they were generally successful in correctly classifying stroke occurrence based on the given

Table 3 SVM - Kernel Analysis

Kernel	Accuracy	Log loss	ROC-AUC score
Linear	0.9847	0.07856	0.6811
RBF	0.9847	0.07938	0.6274
Polynomial	0.9847	0.08556	0.5705
Sigmoid	0.9684	0.0771	0.5931



(I) ROC-AUC curve of KNN for smaller k (k = 11)



(II) ROC-AUC curve of KNN for larger k (k = 169)

Fig. 20 ROC-AUC curves of KNN for different values of k

dataset. The Linear, RBF, and Polynomial kernels produced identical accuracy scores, while the sigmoid kernel had a slightly lower accuracy (Fig. 20).

Log Loss: On the other hand, Sigmoid Kernel has the minimum Log-loss value out of all the models that we have used, followed by the Linear Kernel. This implies that the SVM models

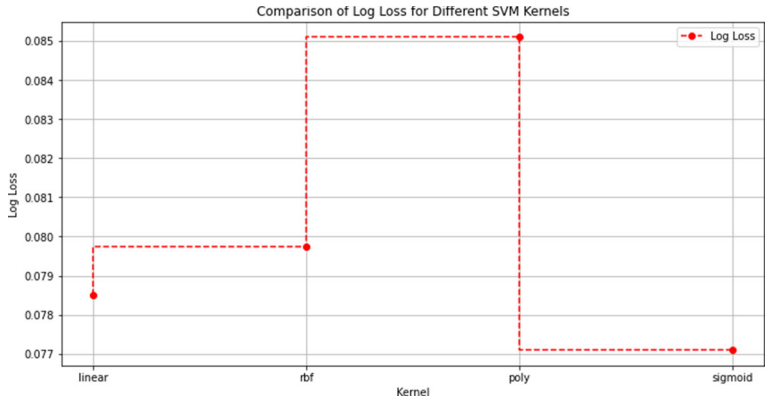


Fig. 21 Comparison of log loss for different SVM kernels

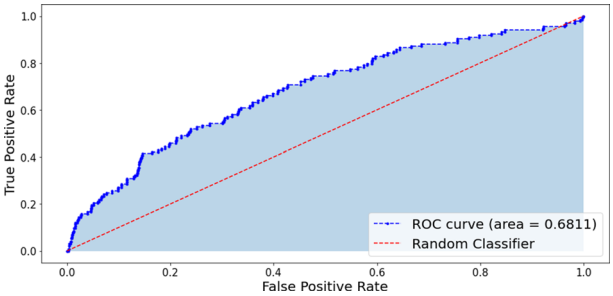


Fig. 22 ROC Curve for Linear Kernel, of SVM

were able to correctly separate the two classes with the given dataset. The estimations illustrate that SVM kernels such as Linear, RBF, and Polynomial are performing well. However, the log-loss value for the Sigmoid Kernel is lower in comparison to the other kernels. This ultimately leads to lower accuracy when compared to other kernels. This suggests that a linear hyperplane would be suitable for building a good model (Fig. 21).

ROC-AUC Score: The ROC AUC curve is displayed, as shown in Fig. 22. The linear kernel had the highest ROC AUC score of 0.6811, indicating good discrimination ability. The polynomial kernel had the lowest ROC AUC score of 0.5705, indicating that changing parameters of k (degree of polynomial) could provide better results [46].

The SVM models demonstrated high accuracy and log loss, indicating their effectiveness in stroke prediction. However, when considering the ROC AUC scores, the linear kernel outperformed the other kernels, indicating better overall discriminative power. The RBF and polynomial kernels also performed reasonably well, while the sigmoid kernel exhibited lower discriminative ability. Refer to Table 3, for the results and estimations, that we have calculated.

Polynomial Kernel Analysis -

The cross-validation scores for the SVM Classifier Model with different degrees of the polynomial kernel are shown in Fig. 23. We analyzed various degrees in the polynomial kernel and obtained the necessary results. We observe that cross-validation scores on the Polynomial Kernel were the highest at $k = 2, 3, 4$ (98.128). The scores gradually decrease as the k value increases. This suggests that higher k values, do not generalize unseen data well. This suggests that the model's performance remains relatively stable within low degrees. Thus, the potential cause of the decrease in CV Scores is overfitting and low generalization. Nonetheless, it is always necessary to conduct further experimentation and fine-tuning of the model to determine its optimal prediction.

Analysis of the Best Kernel -

Analyzing through all estimations, and model kernel performances, despite the Sigmoid kernel having the least loss score, the Linear Kernel provided better results comparatively, with high accuracy, and ROC-AUC Score. According to Table 3, the linear kernel model was able to effectively solve the binary classification issue and produced a favorable classification report. It correctly classified all samples in the dataset as either negative (N) or positive (P), with no instances of false positives (FP). This indicates that the model can be trusted to differentiate between accurate and inaccurate samples. It is important to address the problem of false negatives to ensure the effectiveness of the model in practical applications where

accurately identifying positive instances is crucial. Even though the model performs well, this issue needs to be resolved.

Final Intuitions

Choose the Best Kernel: Based on the analysis, it was found that the linear kernel had a better performance than the other kernels in terms of ROC-AUC score, achieving a high accuracy of 98.4%. It is recommended to use the linear kernel for practical applications due to its superior overall discriminative power.

Address False Negatives: Although the linear kernel had a high level of accuracy, it incorrectly classified 133 positive instances as negative. This could be a critical problem in real-world scenarios where accurately identifying positive cases is crucial. Additional investigation and improvements are necessary to address this issue.

Consider Log-Loss: The log-loss values of SVM models were low, which shows that they could effectively differentiate between the two classes with the provided dataset. It is advisable to take into account both the accuracy and log loss when assessing the model's effectiveness.

Visualize Results: To evaluate and compare model performance, it's helpful to use visualizations like ROC curves, confusion matrices, and graphs. These visual representations offer valuable insights and information.

Further Validation: To ensure the SVM models are robust and effective in different scenarios, it is important to validate them on additional datasets. This will confirm their generalization across various datasets and guarantee their effectiveness. SVM models have proven to be highly accurate and effective in predicting strokes, with low log loss as evidence. Continual evaluation and refinement of these models will lead to even better performance and usability in real-world situations.

7 Future scope and discussion

We conducted a thorough analysis of various ML models in our study to gain valuable insights. The purpose of this research is to gain new insights into the intricate association

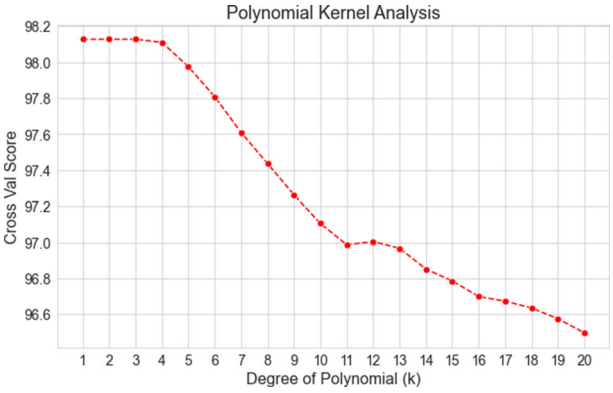


Fig. 23 SVM Model Analysis at various Poly. Degrees (k)

between general health, blood pressure, and the likelihood of experiencing a brain stroke. On ground understanding, researchers can get an overview regarding a clear comparison of stroke prediction using Neural Networks, SVM, and KNN, offering an analysis of the relative effectiveness of these long-standing techniques. We chose these base models-Neural Networks, SVM, and KNN-for their well-proven effectiveness in classification tasks, and compatibility with our dataset's characteristics. The study intends to improve our understanding of the factors that increase the risk of stroke by combining a thorough review of existing literature with advanced data analysis techniques. However, it's important to note that our findings may differ from other datasets related to brain-stroke prediction. We focused on a single dataset and utilized major ML algorithms to predict the class. These models showed remarkable accuracy and minimal loss, validating our chosen algorithms. Along with our research, we also welcome a deeper analysis of SVM using other kernel functions and the use of other algorithms such as wrapper, filter, and embedded methods to identify relevant features for perfect prediction in the future with a large dataset indicative of brain-stroke prediction.

There are paths always open for further research, especially in integrating computer-vision approaches to generate results accurately. We focused on a single dataset, which generalizes our comparisons and proof. Integrating multiple datasets related to stroke, involving blood pressure, cholesterol levels, or hypertension with other risk factors will surely help us understand the variation of the data, and the interconnection of these parameters separately. Exploring biological pathways is open for future research to enhance predictive models' performance. Exploring different biological pathways can also help in understanding how HDL and LDL cholesterol levels can cause brain strokes. Deep learning and neural network techniques can lead to better analysis.

Data Availability Data available on reasonable request

Code Availability Github Link for code

Declaration

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Malone LA, Felling RJ (2020) Pediatric stroke: Unique implications of the immature brain on injury and recovery. Pediatr Neurol 102:3–9
2. Kirton A, Westmacott R, Deveber G (2007) Pediatric stroke: rehabilitation of focal injury in the developing brain. NeuroRehabilitation 22(5):371–382
3. Amarenco P, Labreuche J, Elbaz A, Touboul P-J, Driss F, Jaillard A, Bruckert É (2006) Blood lipids in brain infarction subtypes. Cerebrovascular Diseases 22(2–3):101–108
4. de la Riva P, Zubikarai M, Sarasqueta C, Tainta M, Muñoz-Lopetegui A, Andrés-Marın N, Gonzalez F, Dıez N, de Arce A, Bergareche A et al (2017) Nontraditional lipid variables predict recurrent brain ischemia in embolic stroke of undetermined source. J Stroke Cerebrovasc Dis 26(8):1670–1677
5. Zhang X-X, Wei M, Shang L-X, Lu Y-M, Zhang L, Li Y-D, Zhang J-H, Xing Q, TuErhong ZK, Tang B-P et al (2020) Ldl-c/hdl-c is associated with ischaemic stroke in patients with non-valvular atrial fibrillation: A case-control study. Lipids Health Dis 19(1):1–11
6. Nam K-W, Kwon H-M, Jeong H-Y, Park J-H, Kwon H, Jeong S-M (2019) High triglyceride/hdl cholesterol ratio is associated with silent brain infarcts in a healthy population. BMC neurology 19(1):1–8
7. Chimowitz M, Poole R, Starling M, Schwaiger M, Gross M (1997) Frequency and severity of asymptomatic coronary disease in patients with different causes of stroke. Stroke 28(5):941–945

8. Yamamoto H, Bogousslavsky J, van Melle G (1998) Different predictors of neurological worsening in different causes of stroke. Arch Neurol 55(4):481–486
9. Maulaz AB, Bezerra DC, Michel P, Bogousslavsky J (2005) Effect of discontinuing aspirin therapy on the risk of brain ischemic stroke. Arch Neurol 62(8):1217–1220
10. Gardener H, Wright CB, Rundek T, Sacco RL (2015) Brain health and shared risk factors for dementia and stroke. Nat Rev Neurol 11(11):651–657
11. Chen Y-H, Kang J-H, Lin H-C (2011) Patients with traumatic brain injury: population-based study suggests increased risk of stroke. Stroke 42(10):2733–2739
12. Chiung AKYCC, Lin Y, Hu H-KCJ, Lee H (2010) An integrated machine learning approach to stroke prediction.
13. Khosla A, Cao Y, Lin CC-Y, Chiu H-K, Hu J, Lee H (2010) An integrated machine learning approach to stroke prediction, in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 183–192
14. Jeena R, Kumar S (2018) Machine intelligence in stroke prediction. Int J Bioinform Res Appl 14(1–2):29–48
15. Sprague TC, Ester EF, Serences JT (2016) Restoring latent visual working memory representations in human cortex. Neuron 91(3):694–707
16. Singh MS, Choudhary P (2017) Stroke prediction using artificial intelligence. In: 2017 8th Annual industrial automation and electromechanical engineering conference (IEMECON). IEEE pp 158–161
17. Min SN, Park SJ, Kim DJ, Subramaniyam M, Lee K-S (2018) Development of an algorithm for stroke prediction: a national health insurance database study in Korea. Eur Neurol 79(3–4):214–220
18. Teoh D (2018) Towards stroke prediction using electronic health records. BMC Med Inform Decis Mak 18(1):1–11
19. Dev S, Wang H, Nwosu CS, Jain N, Veeravalli B, John D (2022) A predictive analytics approach for stroke prediction using machine learning and neural networks. Healthcare Analytics 2:100 032 ISSN: 2772-4425. https://doi.org/10.1016/j.health.2022.100032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772442522000090
20. Al-Mamun MA, Strock J, Sharker Y, Shawwa K, Schmidt R, Slain D, Sakhuja A, Brothers TN (2022) Evaluating the medication regimen complexity score as a predictor of clinical outcomes in the critically ill. J Clin Med 11(16):4705
21. Levin EJ, Brissenden JA, Fengler A, Badre D (2023) Predicted utility modulates working memory fidelity in the brain. Cortex 160:115–133
22. Umarova RM, Gallucci L, Hakim A, Wiest R, Fischer U, Arnold M (2024) Adaptation of the concept of brain reserve for the prediction of stroke outcome: proxies, neural mechanisms, and significance for research. Brain Sci 14(1):77
23. Olbrich S, Sander C, Matschinger H, Mergl R, Trenner M, Schönknecht P, Hegerl U (2011) Brain and body. J Psychophysiol
24. Manikandan MS, Soman K (2012) A novel method for detecting r-peaks in electrocardiogram (ecg) signal. Biomed Signal Process Control 7(2):118–128
25. El-Hajj C, Kyriacou PA (2020) A review of machine learning techniques in photoplethysmography for the non-invasive cuff-less measurement of blood pressure. Biomed Signal Process Control 58:101870
26. Stojanova A, Koceski S, Koceska N (2019) Continuous blood pressure monitoring as a basis for ambient assisted living (aal)-review of methodologies and devices. J Med Syst 43:1–12
27. Rajula HSR, Verlato G, Manchia M, Antonucci N, Fanos V (2020) Comparison of conventional statistical methods with machine learning in medicine: diagnosis, drug development, and treatment. Medicina 56(9):455
28. Ray A, Das J, Wenzel SE (2022) Determining asthma endotypes and outcomes: Complementing existing clinical practice with modern machine learning. Cell Rep Med 3(12)
29. Zhang C, Cao L, Romagnoli A (2018) On the feature engineering of building energy data mining. Sustain Cities Soc 39:508–518
30. De Menezes FS, Liska GR, Cirillo MA, Vivanco MJ (2017) Data classification with binary response through the boosting algorithm and logistic regression. Expert Syst Appl 69:62–73
31. J. Gautam, M. Atrey, N. Malsa, A. Balyan, R. N. Shaw, and A. Ghosh, Twitter data sentiment analysis using naive bayes classifier and generation of heat map for analyzing intensity geographically, Advances in Applications of Data-Driven Computing, pp. 129–139, 2021
32. García S, Ramírez-Gallego S, Luengo J, Benítez JM, Herrera F (2016) Big data preprocessing: methods and prospects. Big Data Anal 1(1):1–22
33. Mishra P, Biancolillo A, Roger JM, Marini F, Rutledge DN (2020) New data preprocessing trends based on ensemble of multiple preprocessing techniques. TrAC, Trends Anal Chem 132:116045

34. Boateng EY, Otoo J, Abaye DA (2020) Basic tenets of classification algorithms k-nearest-neighbor, support vector machine, random forest and neural network: A review. J Data Anal Inf Process 8(4):341–357
35. Agatonovic-Kustrin S, Beresford R (2000) Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. J Pharm Biomed Anal 22(5):717–727
36. Livingstone DJ (2008) Artificial neural networks: methods and applications. Springer
37. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
38. Alarabeyyat A, Alhanahnah M et al (2016) Breast cancer detection using k-nearest neighbor machine learning algorithm. In: 2016 9th International conference on developments in esystems engineering (DeSE), IEEE, pp 35–39
39. Kramer O, Kramer O (2013) K-nearest neighbors. Dimensionality reduction with unsupervised nearest neighbors, pp 13–23,
40. Lee D-T (1982) On k-nearest neighbor voronoi diagrams in the plane. IEEE Trans Comput 100(6):478–487
41. Hu L-Y, Huang M-W, Ke S-W, Tsai C-F (2016) The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus 5(1):1–9
42. Chomboon K, Chujai P, Teerarassamee P, Kerdprasop K, Kerdprasop N (2015) An empirical study of distance metrics for k-nearest neighbor algorithm. In: Proceedings of the 3rd international conference on industrial application engineering, vol. 2
43. Tanemura M, Ogawa T, Ogita N (1983) A new algorithm for three-dimensional voronoi tessellation. J Comput Phys 51(2):191–207
44. Ma Y, Guo G (2014) Support vector machines applications. Springer, vol. 649
45. Wang L (2005) Support vector machines: theory and applications. Springer Science & Business Media, vol 177
46. Yin X, Goudriaan J, Lantinga EA, Vos J, Spiertz HJ (2003) A flexible sigmoid function of determinate growth. Ann Bot 91(3):361–371
47. Kavzoglu T, Colkesen I (2009) A kernel functions analysis for support vector machines for land cover classification. Int J Appl Earth Obs Geoinf 11(5):352–359
48. Moghaddam VH, Hamidzadeh J (2016) New hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier. Pattern Recognit 60:921–935
49. Chamasemani FF, Singh YP (2011) Multi-class support vector machine (svm) classifiers-an application in hypothyroid detection and classification. In: 2011 sixth international conference on bio-inspired computing: theories and applications. IEEE pp 351–356
50. Hajian-Tilaki K (2013) Receiver operating characteristic (roc) curve analysis for medical diagnostic test evaluation. Caspian J Intern Med 4(2):627