CDS503: Machine Learning

Academic Session: Semester 1, 2023-2024

School of Computer Sciences, USM, Penang

PROJECT

Title: Machine Learning Approaches for Stroke Prediction: Comparative Analysis

Group Members

No.	Name	Matric number
1.	YAOXIAO	22202255
2.	YE CAIXIA	22202532
3.	ZIYING WANG	22203495
4.	Abdulkarem Khaled Abdullah Bawazir	22306668

Group Name	**:**	Project Group07
Project Dataset	**:**	healthcare-stroke-dataset

Contents

Abstract

The primary focus of this research is stroke prediction, and to tackle this important classification issue, sophisticated data processing methods and machine learning models are used. By utilizing cutting-edge algorithms like Random Forest and Logistic Regression, the research seeks to improve the precision and effectiveness of stroke risk assessment. The research aims to create a strong predictive model by methodically processing and evaluating pertinent health data, such as demographics, medical history, and lifestyle factors. A thorough method for categorizing people at risk of stroke is provided by the use of Logistic Regression and Random Forest models, which helps with early detection and proactive intervention. This research adds to the expanding field of predictive healthcare analytics by shedding light on stroke risk assessment and opening the door to more effective preventative measures and individualized patient care.

Key Words: stroke, classification, machine learning, algorithms

1 Project Background

1.1 Background of the problem domain

Global strokes present a critical health challenge, ranking as the third leading cause of mortality, causing millions of permanent disabilities annually. With approximately 10 million strokes reported worldwide in 2020, the World Health Organization emphasizes the urgent need to address this issue[1]. Strokes, stemming from factors like high blood pressure and obesity, require early diagnosis for improved recovery, a challenge AI technology are addressing. This study focuses on leveraging AI, specifically analyzing brain CT scans, to identify individuals at risk of stroke[2]. Objectives include developing an accurate AI model for high-risk identification and discussing implications. The research responds to global healthcare needs, endorsed by authoritative bodies, emphasizing the role of early diagnosis. The paper unfolds with a literature review, methodology, results, and conclusions[3].

1.2 Issues and Problem Statement

The incorporation of machine learning (ML) in stroke diagnosis, while pioneering, is not devoid of challenges. One of the prominent issues is the inherent complexity and variability of stroke pathology which poses a challenge for ML algorithms. Strokes can be of various types, primarily ischemic or hemorrhagic, each requiring different diagnostic criteria and treatment approaches. Consequently, ML models must be extraordinarily nuanced to accurately distinguish between these types. Moreover, the quality and quantity of data available to train these models are often limited by diverse patient demographics, inconsistent data collection methods, and privacy concerns. Insufficient or biased training data can lead to inaccurate models that fail to generalize across different patient populations. Another significant issue is the interpretability of ML models. The "black box" nature of certain ML algorithms makes it difficult for clinicians to understand the diagnostic rationale, leading to resistance to adopting these tools despite their potential. Data security and patient privacy also present considerable concerns, as ML systems dealing with sensitive health data must be robust against cyber threats while complying with stringent regulations like the Health Insurance Portability and Accountability Act (HIPAA). Finally, the integration of ML tools into clinical workflows presents a logistical challenge. Ensuring that these tools complement rather than disrupt existing practices requires careful planning and consideration of the unique demands and constraints of clinical environments.

1.3 Objectives and motivation

Stroke is a major health crisis that demands immediate attention for effective treatment. However, traditional diagnostic methods rely heavily on the subjective assessment of symptoms and manual interpretation of medical imaging, which can lead to delays and diagnostic errors. Machine learning has the potential to analyze complex data rapidly and identify patterns that might not be immediately obvious to human observers, providing a tantalizing complementary tool to traditional approaches.

There are several driving motivations behind the focus on ML for stroke diagnosis:

- Improved Patient Outcomes: Faster and more accurate stroke diagnosis can significantly enhance patient survival rates and long-term recovery outcomes.
- Efficiency in Healthcare: By accelerating the diagnostic process, healthcare systems can reduce bottlenecks, thereby improving overall efficiency and the allocation of resources.
- Economic Savings: Early and reliable stroke identification may decrease hospital stays and lower the burden of post-stroke care, culminating in economic savings both for healthcare systems and patients.
- Research and Innovation: Creating a successful ML model could stimulate further research into AI applications in healthcare, potentially revolutionizing multiple facets of patient care.
- Supporting Rural and Underserved Communities: By potentially offering diagnostic support through telehealth services, ML could improve healthcare delivery in communities with limited access to specialized care.

This project is inspired by the conviction that ML can and should play a critical role in the fight against stroke, blending the prowess of cutting-edge technology with the depth of human clinical expertise to battle one of the gravest health challenges of our time.

1.4 Limitations

Machine learning (ML) in stroke diagnosis, while promising, comes with a set of inherent limitations that must be acknowledged and addressed:

- Data Quality and Availability: One of the main limitations is the availability of high-quality, annotated medical data for training ML models. Privacy concerns, legal restrictions, and the cost of data collection can severely limit the quantity and diversity of the data.
- Interpretability and Trust: Even the most accurate ML model is of limited clinical utility if healthcare providers cannot understand how it arrives at its diagnosis. The so-called "black box" problem in many complex models can lead to a lack of trust among medical professionals.
- Hardware Requirements: The computational intensity of training ML models requires significant hardware resources, which can be a limiting factor, particularly in resource-constrained environments.
- Cybersecurity Threats: When integrating ML into healthcare systems, cybersecurity becomes a barrier. ML systems are at risk of cyber threats, which can have dire consequences if they affect stroke diagnosis tools.

2 Literature Review

There has been plenty of research conducted recently into employing various machine learning methods to predict strokes:

- Studies have shown that machine learning algorithms, such as Convolutional Neural Networks, can effectively predict stroke lesions. Compared to traditional methods, these algorithms have demonstrated advantages in predicting stroke lesions. Several companies have already developed automated and semi-automated commercial software for acute stroke diagnosis, including RapidAI® and Viz.ai®, which have received regulatory approval from the US Food and Drug Administration (FDA)[4].

- In the domain of ischemic stroke diagnosis, Garcia-Terriza et al. [5]applied the Random Forest (RF) algorithm to discern stroke types and predict mortality rates, attaining commendable accuracy rates of 92% and 96%, respectively, within a cohort of 119 patients. Nevertheless, the exclusion of commonplace risk factors curtails the general applicability of their findings.
- Sung et al. [6]adopted a phenotyping strategy for ischemic stroke, employing diverse ML models such as C4.5, CART, KNN, RF, SVM, and LR. By scrutinizing clinical records with preprocessing and MetaMap, they elevated accuracy and Kappa scores by incorporating textual data and the National Institutes of Health Stroke Scale (NIHSS). Notwithstanding, the persistent challenge of precisely defining stroke phenotypes pervaded their study.
- Giri et al. [7]explored electroencephalography (EEG) for ischemic stroke diagnosis, leveraging 1D CNN and various ML models to categorize 32 patients with acute ischemic stroke and 30 controls. The 1D CNN model showcased promising accuracy (0.86) and F-score (0.861), underscoring the potential utility of EEG in acute ischemic stroke diagnosis. However, the time-intensive application of EEG electrodes emerged as a notable constraint.
- In the context of hemorrhagic stroke subtyping, Dhar et al. sought to quantify intracranial hemorrhage and perihematomal edema (PHE). While the ML method employed remained unspecified, their study yielded robust Dice scores of 0.9 for hemorrhage and 0.54 for PHE when scrutinizing 24-hour head CT scans[8]. Challenges persisted in accurately delineating intraventricular hemorrhage.
- Ramos et al. directed their efforts toward predicting delayed cerebral ischemia (DCI) using ML methodologies such as logistic regression, SVM, RF, and MLP. [9]By integrating non-contrast CT image data with clinical variables, their RF model achieved a promising ROC of 0.74, sensitivity of 0.75, and specificity of 0.67. However, the manual feature extraction process was identified as time intensive.
- Tanioka et al. Harnessed RF to prognosticate DCI, encompassing clinical variables and matrix metalloproteinase (MCP) levels in their investigation involving 95 patients. Noteworthy accuracies of 93.9% for clinical variables, 87.2% for MCP, and 95.2% for the combined approach were achieved[10]. However, the necessity for additional data on other biomarkers was underscored.
- Ni et al. concentrated on stroke case detection utilizing ML techniques such as LR, SVM-P, SVM-R, RF, and ANN in the broader realm of stroke detection and classification[11]. Through the alignment of medical records with International Classification of Diseases (ICD) codes, they garnered significant metrics, including an accuracy of 88.6%, precision of 93.8%, recall of 92.8%, F-score of 93.3%, AUC of 89.8%, and AUC-PR of 97.5%. Nevertheless, potential inaccuracies in ICD coding constrained the overall accuracy.

- Park et al. endeavored to autonomously score the National Institutes of Health Stroke Scale (NIHSS) and Medical Research Council (MRC) scores utilizing wearable sensors. Employing SVM and ensemble ML methods, optimal results were achieved through a Bayesian optimization search[12]. Despite a sample size of 240 participants, the wearable sensor data-based approach exhibited promise for automated stroke severity scoring.

- Data from Kaggle was acquired by Dritsas and Trigka, with a total of 3254 participants. There are ten independent features in the dataset, including age, BMI, glucose level, smoking status, the presence of hypertension, and if the person had previously suffered a stroke.[13] Preprocessing of the data was done. Regarding the dataset, and class balancing was carried out using a resampling technique called SMOTE. Machine learning models such as Random Forest, Decision Tree, and Stacking KNN, Multilayer Perception, Naïve Bayes, Majority Voting, Logistic regression and stochastic gradient descent were employed. For foretelling a stroke or not. Based on the outcomes, it seems that the stacking classifier achieved 0.989, the best performance.

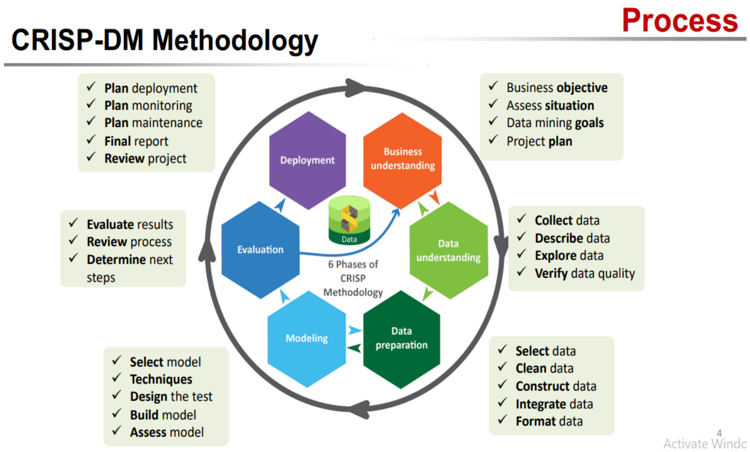
- The Kaggle dataset and certain algorithms, such as Decision Tree, Naïve Bayes, Support Vector Machine, Random Forest, and K-Nearest Neighbor, were also utilized by Rakshit. as well as Logistic Regression. Based on their findings, the greatest Decision Tree was used to record performance, and then KNN (96.3%)[14].

These studies collectively showcase the immense potential of ML-based methodologies in enhancing stroke diagnosis and management.

3 Methodology

3.1 Project Framework

This project, will be using the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology for a machine learning project focused on stroke diagnosis involves adapting its six-phase framework to suit the specific tasks and objectives of the medical domain. Here's how each phase would typically be addressed:



3.1.1 Business Understanding

The first step in the CRISP-DM, phase is to understand the background of the project, determine the requirements of the project, establish the objectives of the project, and design a plan for the project that can meet those objectives.

3.1.2 Data Understanding

The second step of CRISP-DM for stroke diagnosis involves gathering and analyzing existing medical data, which could include patient demographics, medical histories, lab results, and brain imaging data. The team would explore the data to identify potential patterns or problems, such as biases or gaps in the data that could affect diagnostic accuracy.

3.1.3 Data Preparation

The third step in the CRISP-DM process is Data preparation in a clinical setting involves careful curation of the datasets to ensure they are clean, complete, and representative. we discuss how we plan to clean the data by removing outliers, replacing empty data, and filtering out data to bring the dataset more into balance.

3.1.4 Modeling

The modeling phase in the context of stroke diagnosis would entail selecting and training algorithms capable of processing complex medical data. Techniques like convolutional neural networks for image recognition or recurrent neural networks for sequential data like electronic health records might be explored. Models would be trained, tested, and validated using the prepared datasets.

3.1.5 Evaluation

Evaluation would be particularly stringent due to the implications on patient health. Beyond statistical performance measures, models would be evaluated on their clinical relevance, interpretability by healthcare professionals, and compliance with medical regulations.

3.1.6 Deployment

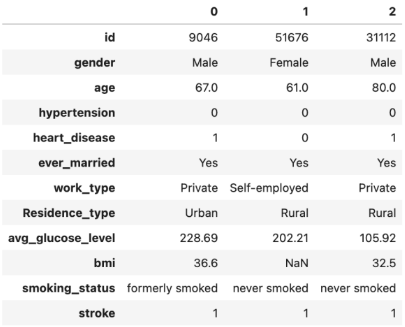
The final phase of CRISP-DM, will be executed via the oral presentation given later this semester. After deployment, there would likely be ongoing monitoring and maintenance to ensure the model adapts to new data and remains clinically useful, indicating that CRISP-DM’s cyclical nature is well-suited to the ever-evolving field of healthcare.

3.2 Data Exploration and Processing

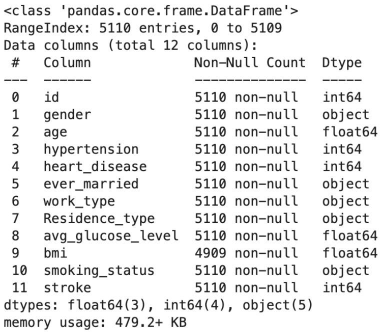
3.2.1 Description of The Data

The Stroke Prediction Dataset, which was made available on Kaggle by user fedesoriano, is a comprehensive collection of health-related characteristics aimed at predicting an individual's risk of suffering a stroke. The original data for the dataset is sourced from healthcare records and surveys. The dataset comprises 12 columns with 5110 rows; four of the features are numeric, namely 'id', 'age', 'avg_glucose_level' and 'bmi' while the other eight are categorical. A variety of lifestyle, health, and demographic factors are included to aid in the evaluation of stroke risk. The dataset, which originally included information from a variety of persons in various health-related circumstances, was gathered from medical exams and surveys.

The sample data and data types are listed below:



	0	1	2
id	9046	51676	31112
gender	Male	Female	Male
age	67.0	61.0	80.0
hypertension	0	0	0
heart_disease	1	0	1
ever_married	Yes	Yes	Yes
work_type	Private	Self-employed	Private
Residence_type	Urban	Rural	Rural
avg_glucose_level	228.69	202.21	105.92
bmi	36.6	NaN	32.5
smoking_status	formerly smoked	never smoked	never smoked
stroke	1	1	1



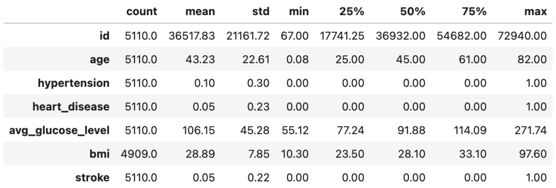
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

Fig 1. example Data

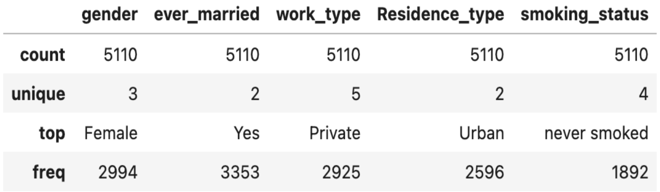
Fig 2. Data types

3.2.2 Exploring Dataset Characteristics

Numerical summary statistics, encompassing central tendencies and measures of dispersion, illuminate the distribution and spread of numerical data. Concurrently, categorical features are encapsulated by count, unique categories, the most frequent category, and its frequency. Use the `describe () ` function aids in making informed decisions about data preprocessing and revealing initial insights.



	count	mean	std	min	25%	50%	75%	max
id	5110.0	36517.83	21161.72	67.00	17741.25	36932.00	54682.00	72940.00
age	5110.0	43.23	22.61	0.08	25.00	45.00	61.00	82.00
hypertension	5110.0	0.10	0.30	0.00	0.00	0.00	0.00	1.00
heart_disease	5110.0	0.05	0.23	0.00	0.00	0.00	0.00	1.00
avg_glucose_level	5110.0	106.15	45.28	55.12	77.24	91.88	114.09	271.74
bmi	4909.0	28.89	7.85	10.30	23.50	28.10	33.10	97.60
stroke	5110.0	0.05	0.22	0.00	0.00	0.00	0.00	1.00



	gender	ever_married	work_type	Residence_type	smoking_status
count	5110	5110	5110	5110	5110
unique	3	2	5	2	4
top	Female	Yes	Private	Urban	never smoked
freq	2994	3353	2925	2596	1892

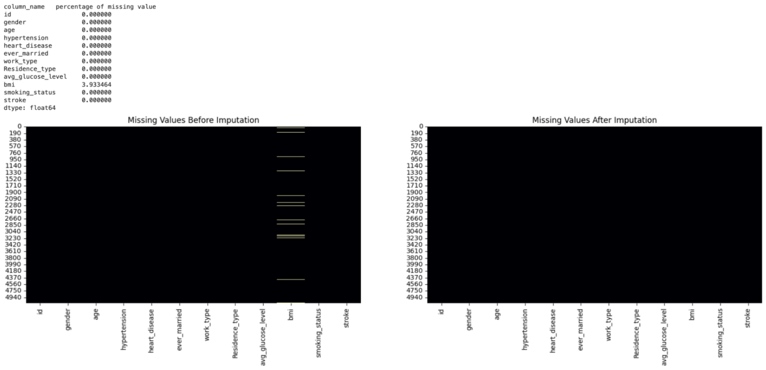
Fig 3. Describe Numeric Features

Fig 4. Describe Categorial Features

From the descriptive statistics Fig 4., we gain valuable insights into the key characteristics of the dataset. The 'age' column indicates a diverse age range with a mean of 43.23, reflecting a broad representation of individuals. The 'hypertension' and 'heart_disease' columns, with low mean values of 0.10 and 0.05, respectively, suggest a relatively low prevalence of these conditions in the dataset. The 'avg_glucose_level' and 'bmi' columns exhibit varying ranges, highlighting the diversity in glucose levels and body mass index among the individuals.

3.2.3 Exploring and Handling Missing Values

In order to guarantee accurate insights and avoid biases in statistical analyses, it is imperative to address them. Handling missing values correctly preserves the integrity of the data, boosts machine learning model performance, and makes the data easier to interpret overall. Common techniques include imputation based on mean or mode values, deletion, and more sophisticated techniques like regression imputation. The choice of an appropriate technique depends on the features of the missing data as well as the particular goals of the analysis.



```
column_name   percentage of missing value
id                    0.000000
gender                0.000000
age                   0.000000
hypertension          0.000000
heart_disease         0.000000
ever_married          0.000000
work_type             0.000000
Residence_type        0.000000
avg_glucose_level     0.000000
bmi                   3.933464
smoking_status        0.000000
stroke                0.000000
dtype: float64
```

Fig 5. Explore and Handle Missing Values

The Fig5. Explore and Handle Missing Values presenting descriptive statistics for the variable 'bmi', it is evident that both the mean and median values are closely aligned. This observation indicates a relatively symmetric distribution of the 'bmi' data, with the mean serving as a representative measure of central tendency. Consequently, for the missing values within the 'bmi' variable, it is deemed appropriate to impute them with the mean value.

3.2.4 Distribution and Proportion of Stroke Cases

Understanding and analyzing the classification label is fundamental for building accurate predictive models. It serves as the cornerstone for comprehending data patterns and selecting relevant features. This careful analysis ensures the success of classification algorithms in accurately categorizing new, unseen data points.

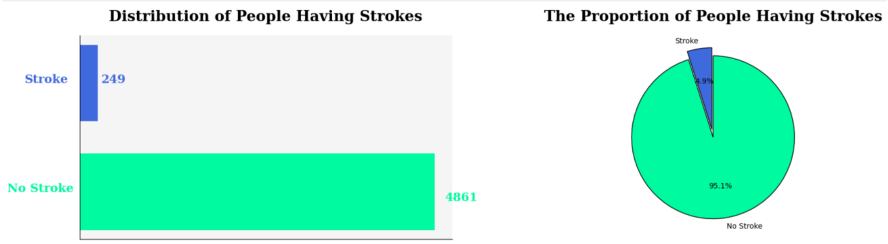


Fig 6. The Distribution of Strokes and Proportion of People Having Strokes

According to distribution, our sample data indicates that 5 out of 100 individuals have a stroke. Additionally, this data distribution is extremely unbalanced; the null accuracy score of the distribution itself is 95.1%, meaning that any dump model that makes random predictions of strokes could potentially achieve this level of accuracy. Therefore, to get the best results when modeling and training data, either oversampling or undersampling must be done.

3.2.5 Comparative Analysis of Averages in Stroke and Non-Stroke Cases

The side-by-side heatmaps compare mean descriptive statistics for individuals with strokes (left panel) and without strokes (right panel). This visualization quickly highlights potential variations in mean values across different variables between the two groups.

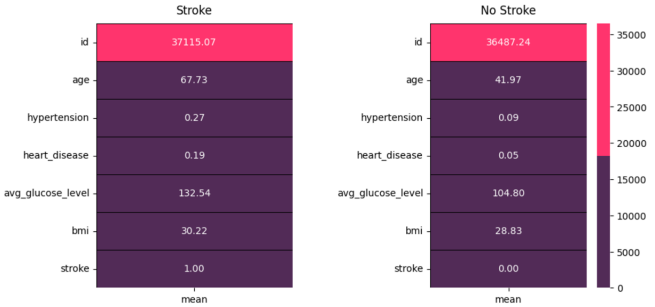


Fig 7. Analysis of Averages in Stroke and Non-Stroke Cases

Average values of all the characteristics for both stroke-suffered and stroke-free cases. Age and average blood sugar level are reliable first-hand markers of stroke. The mean age of stroke patients, 67.73, is significantly higher than the mean age of non-stroke patients, 41.97. In a similar vein, an average glucose level of 132.54 may be associated with a higher risk of stroke than the 104.80 average glucose level observed in individuals who did not experience a stroke.

3.2.6 The Distribution of Numeric Features and Impact on having strokes

These series of visualizations delve into the distributions of key health-related variables and their association with stroke occurrence. Each pair of plots showcases the variable's general distribution on the left and its distribution concerning stroke status on the right. The variables explored include 'Average Glucose Level', 'BMI (Body Mass Index)', 'Hypertension' (high blood pressure), and 'Heart Disease'. The purpose of these visualizations is to unravel potential patterns or differences in these health indicators between individuals who experienced a stroke and those who did not. Analyzing the overlaid distributions aids in identifying potential correlations or disparities that may contribute to stroke risk assessment, providing valuable insights for further investigation and understanding within the context of the dataset.

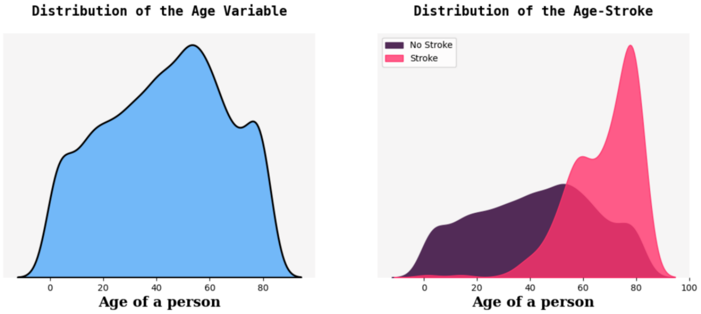


Fig 8. The Distribution of the Age and How Age Impact on Strokes

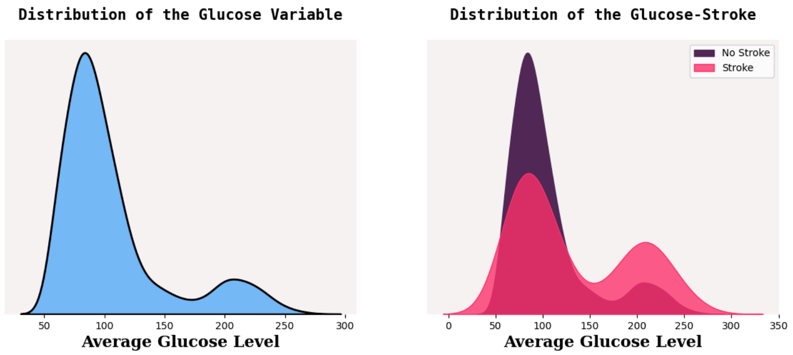


Fig 9. The Distribution of the Average Glucose Level and How Glucose Impact on Strokes

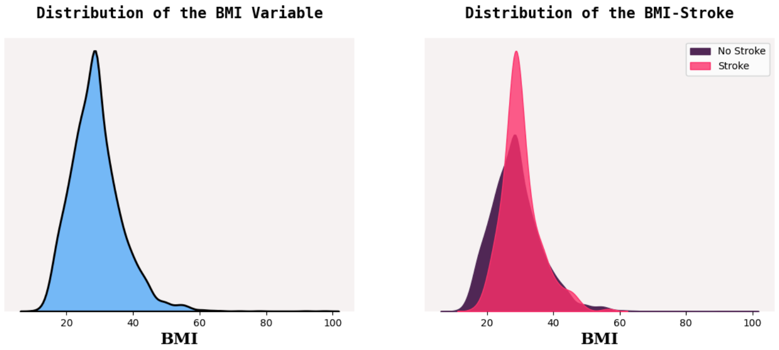


Fig 10. The Distribution of the BMI and How BMI Impact on Strokes

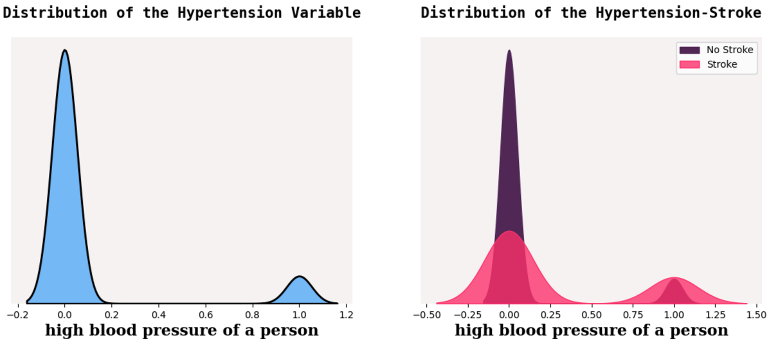


Fig 11. The Distribution of the Hypertension and How Hypertension Impact on Strokes

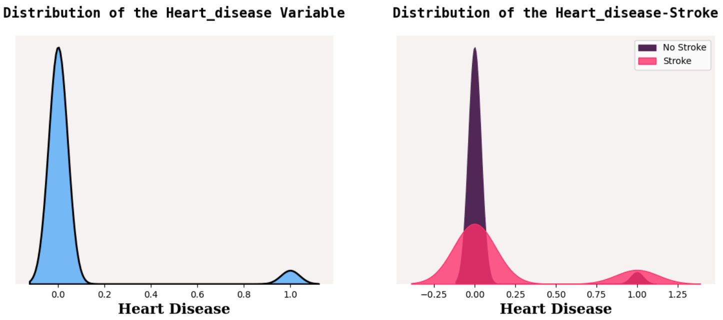


Fig 12. The Distribution of the Heart Diseases and How Heart Diseases Impact on Strokes

3.2.7 The Distribution of the Gender and How Gender Impact on Strokes

Visualizations inspect the link between gender and stroke occurrence. The first subplot, a horizontal bar chart, offers a brief statistical snapshot of gender distribution. It enhances demographic insights by displaying the number of male and female individuals. The second subplot utilizes kernel density plots to illustrate gender distribution in relation to stroke events, revealing potential trends based on gender differences.

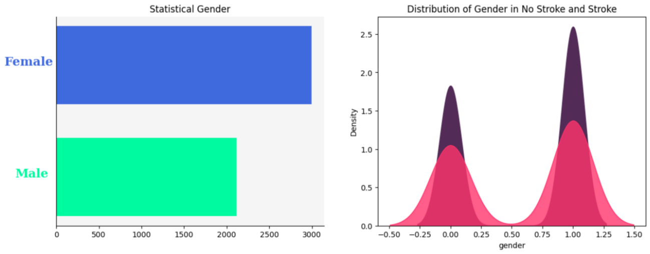


Fig 13. The Distribution of the Gender and How Gender Impact on Strokes

3.2.8 Exploring BMI Outliers

This visualization analyzes BMI outliers in the dataset. A red dashed line at BMI 70 sets the threshold for identifying extreme values. The scatter plot distinguishes regular (blue) and outlier (red 'x') BMI values.

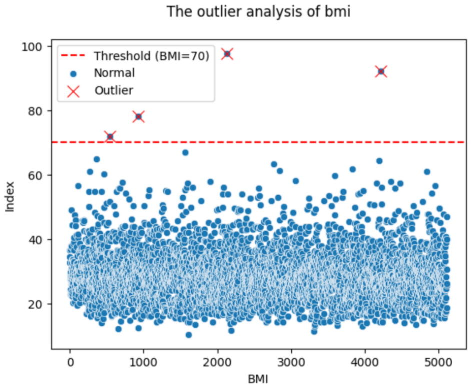
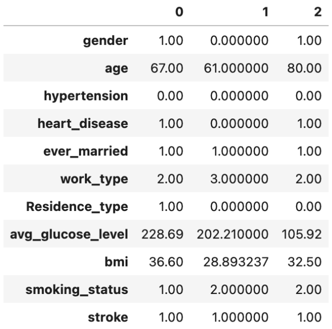


Fig 14. Explore BMI Outliers

3.2.9 Transforming Categorical Features

Categorical features consist of non-numeric labels, making them unsuitable for many machine learning algorithms. `LabelEncoder () ` systematically assigns a unique numerical label to each distinct category within a feature, thereby converting categorical data into a format compatible with numerical analysis. The purpose of this transformation is to enable machine learning models to interpret and effectively learn from categorical information. Additionally, this method aids in revealing inherent ordinal relationships within categorical variables, crucial for understanding the impact of different categories on the target variable.

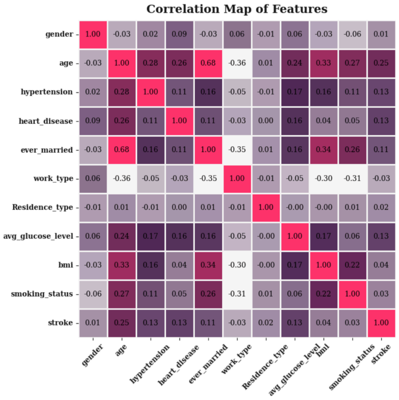


	0	1	2
gender	1.00	0.000000	1.00
age	67.00	61.000000	80.00
hypertension	0.00	0.000000	0.00
heart_disease	1.00	0.000000	1.00
ever_married	1.00	1.000000	1.00
work_type	2.00	3.000000	2.00
Residence_type	1.00	0.000000	0.00
avg_glucose_level	228.69	202.210000	105.92
bmi	36.60	28.893237	32.50
smoking_status	1.00	2.000000	2.00
stroke	1.00	1.000000	1.00

Fig 15. Example of converted data

3.2.10 Correlation Map of Features

A Correlation Map of Features is a visual representation that illustrates the pairwise relationships between different features within a dataset. Each cell in the map corresponds to the correlation coefficient, indicating the strength and direction of the linear relationship between two features. Strong linear relationships are suggested by high positive correlation values (so close to 1) and strong negative correlation values (so close to -1) respectively. Finding patterns, dependencies, and possible multicollinearity between variables is made easier with the aid of the map, which provides insightful information for feature selection and comprehension of the dataset dynamics.



	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
gender	1.00	-0.03	0.02	0.09	-0.03	0.06	-0.01	0.06	-0.03	-0.06	0.01
age	-0.03	1.00	0.28	0.26	0.68	-0.36	0.01	0.24	0.33	0.27	0.25
hypertension	0.02	0.28	1.00	0.11	0.16	-0.05	-0.01	0.17	0.16	0.11	0.13
heart_disease	0.09	0.26	0.11	1.00	0.11	-0.03	0.00	0.16	0.04	0.05	0.13
ever_married	-0.03	0.68	0.16	0.11	1.00	-0.35	0.01	0.16	0.34	0.26	0.11
work_type	0.06	-0.36	-0.05	-0.03	-0.35	1.00	-0.01	-0.05	-0.30	-0.31	-0.03
Residence_type	-0.01	0.01	-0.01	0.00	0.01	-0.01	1.00	-0.00	-0.00	0.01	0.02
avg_glucose_level	0.06	0.24	0.17	0.16	0.16	-0.05	-0.00	1.00	0.17	0.06	0.13
bmi	-0.03	0.33	0.16	0.04	0.34	-0.30	-0.00	0.17	1.00	0.22	0.04
smoking_status	-0.06	0.27	0.11	0.05	0.26	-0.31	0.01	0.06	0.22	1.00	0.03
stroke	0.01	0.25	0.13	0.13	0.11	-0.03	0.02	0.13	0.04	0.03	1.00

Fig 16. Correlation Map of Features

3.2.11 Data Sampling

The SMOTE algorithm is employed for oversampling, primarily addressing the issue of imbalanced categorical data in the dataset. Through oversampling, the training set achieves a balanced distribution of samples across different categories, facilitating the model in better learning from minority categories. By visualizing the changes in category distribution before and after oversampling via pie charts, the transformation in data distribution becomes evident.

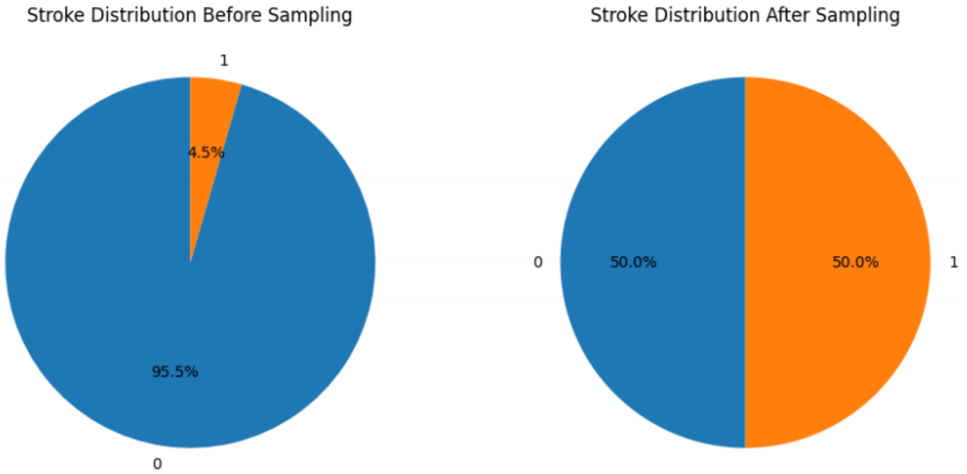
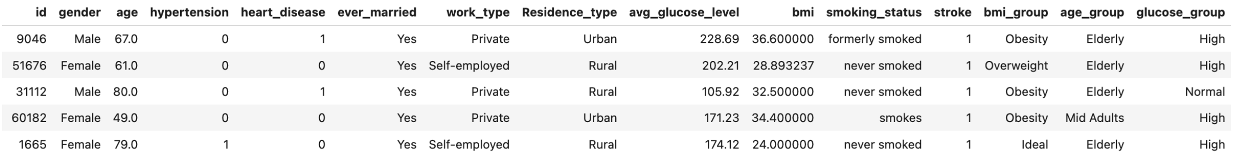


Fig 17. Oversampling before and after comparison

3.2.12 Data Transformation

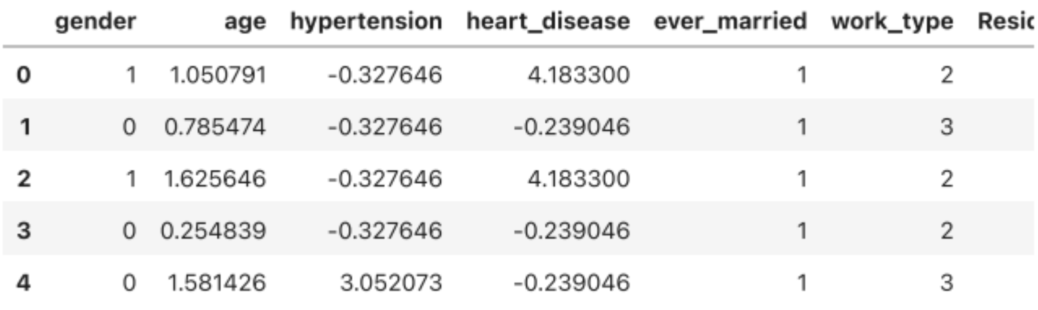


id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke	bmi_group	age_group	glucose_group
9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly smoked	1	Obesity	Elderly	High
51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	never smoked	1	Overweight	Elderly	High
31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never smoked	1	Obesity	Elderly	Normal
60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	smokes	1	Obesity	Mid Adults	High
1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never smoked	1	Ideal	Elderly	High

Fig 18. Data sample

Before discussing the details of the dataset, some example rows from the dataset are shown in Fig 18. Fig 18 shows that the dataset consists of 15 columns and one target class (stroke). Then, some preprocessing is performed on the data set to ensure the quality of the data so that it can be better used for training and evaluating machine learning models.

First, the non-numeric features in the original data are encoded and converted into numerical types, which facilitates training of the classification model.



	gender	age	hypertension	heart_disease	ever_married	work_type	Resic
0	1	1.050791	-0.327646	4.183300	1	2	
1	0	0.785474	-0.327646	-0.239046	1	3	
2	1	1.625646	-0.327646	4.183300	1	2	
3	0	0.254839	-0.327646	-0.239046	1	2	
4	0	1.581426	3.052073	-0.239046	1	3	

Fig 19. Convert numeric data

Secondly, the data set is divided into a training set and a test set. The split training set and test set can be used to train and evaluate the model, which helps to discover the performance of the model on unseen data.

3.3 Machine Learning Algorithms

3.3.1 Parametric and Non-parametric algorithms

Machine learning algorithms can be divided into two types: parametric and non-parametric. Parametric algorithms reduce a function to a known form (making solid assumptions about the data). There are a fixed number of parameters - no matter what you ask. No matter how much data a parametric model throws at it, it won't change its idea of how many parameters it needs. Non-parametric algorithms do not make strong assumptions about the data (free to learn any functional form from the training data) and have a flexible number of parameters - the number of parameters usually grows as it learns from more data. Its specific classifier is as shown in the Fig 20:

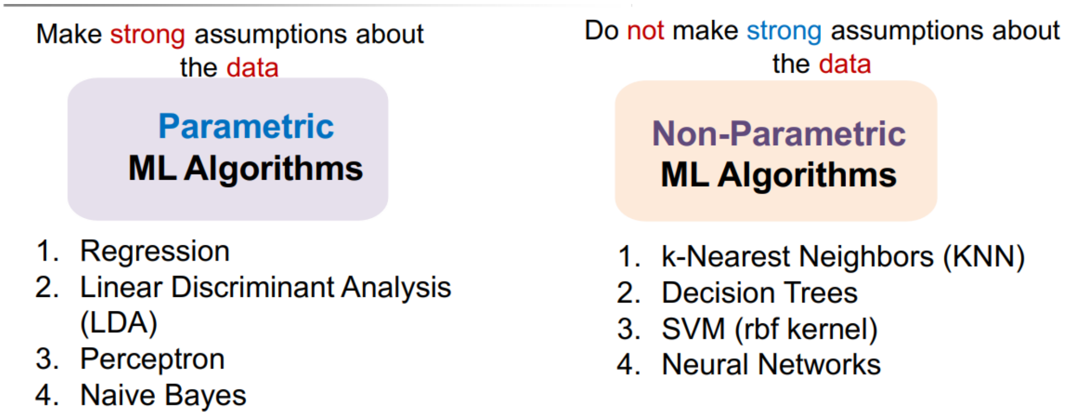

Make strong assumptions about the data
Parametric ML Algorithms
1. Regression
2. Linear Discriminant Analysis (LDA)
3. Perceptron
4. Naive Bayes
Do not make strong assumptions about the data
Non-Parametric ML Algorithms
1. k-Nearest Neighbors (KNN)
2. Decision Trees
3. SVM (rbf kernel)
4. Neural Networks

Fig 20.Non-Parametric and Parametric algorithm classifier

Based on the parametric and non-parametric algorithms and analysis of this study's main stroke data set, we selected the non-parametric algorithm SVC or Random Forest Parametric algorithm logistic regression for our main algorithm.

3.3.2 Random Forest

Random forest can also be directly used to solve classification and regression problems. It is known as the most accurate learning algorithm. It integrates multiple decision trees through numerous decision trees to improve the model's accuracy.

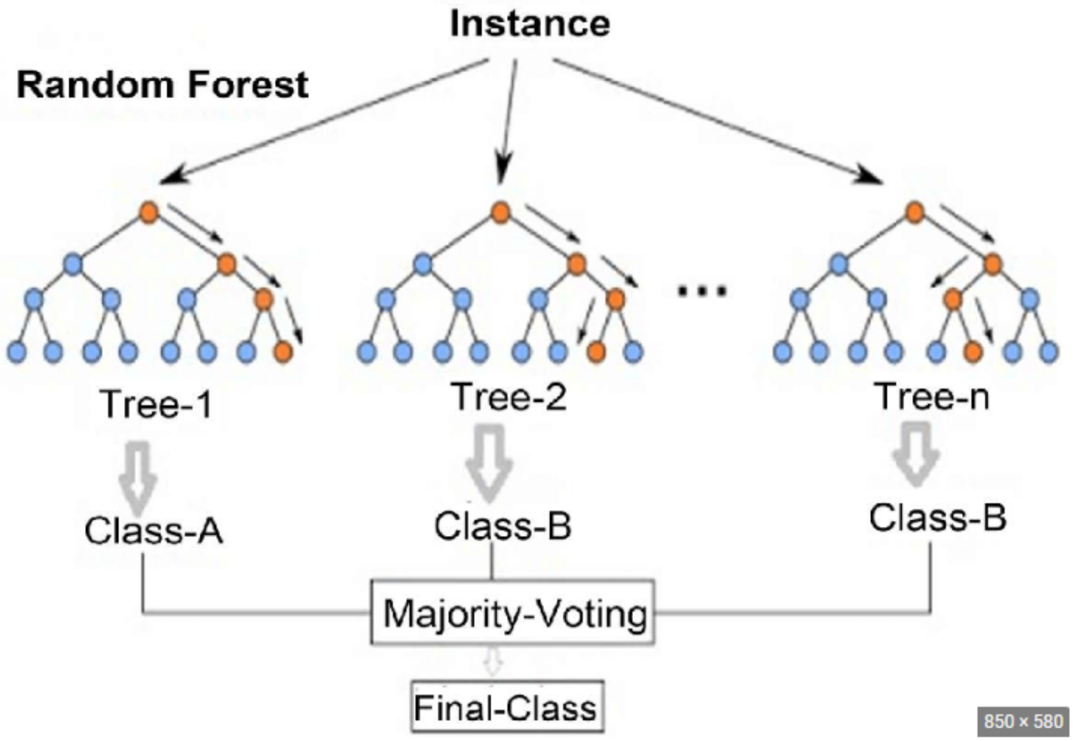


Fig 21. Random forest

As can be seen from the above picture, random forest mainly samples and divides nodes based on data and builds different decision trees based on different samples. This step is repeated, and finally, multiple decision trees are formed to classify and predict the results. The advantage of random forest is that it can provide high accuracy, can be applied to various types of data, has little impact on over-fitting, and can be used to process large-scale data sets.

3.3.3 SVM

SVM is a commonly used non-parametric supervised machine learning algorithm suitable for classification and regression. It is currently one of the best-performing tools in many classification tasks. It creates a hyper-plane that can divide the data space during the data training phase and then uses the hyper-plane to classify or regress new data during the testing phase.

Support vectors: These are the points closest to the hyper-plane. A dividing line will be defined with the help of these data points.

Margin: The distance between the hyper-plane and the observation (support vector) closest to the hyper-plane. In SVM, a more significant margin is considered a good margin. There are two types of margins: hard margins and soft margins.

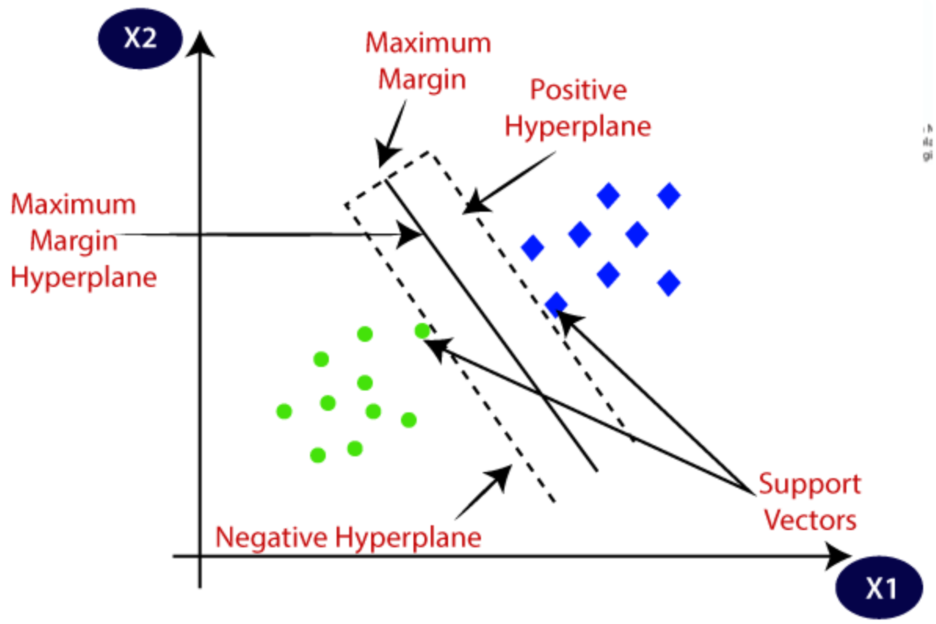


Fig 22. SVM

Types of SVM algorithms:

1. Linear SVM: We can use linear SVM only when the data is completely linearly separable. Being completely linearly separable means that the data points can be divided into 2 classes using a straight line (if 2D)[11].

hard spacing： Given the training data set $\{(X_i, Y_i)\}$, where $X_i$is the input feature vector and Y_i is the category label (+1 or -1), the optimization problem of linear SVM can be expressed as, maximizing: $\frac{1}{\|\mathbf{w}\|}$ Restrictions: $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$. Normal vector of the hyper-plane, and B is the bias term.

2. Nonlinear SVM: When the data are not linearly separable, we can use nonlinear SVM, which means when the data points cannot be divided into 2 categories using a straight line (if it is 2D), we can use some advanced techniques like Kernel functions are used to classify them. In most practical applications, we cannot find linearly separable data points, so we use kernel tricks to solve for them.

Soft margin: When facing nonlinear separable data, a slack variable ξ_i is introduced to tolerate some errors. The optimization problem of soft margin SVM can be expressed as, minimizing: $\frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{N} \xi_i$, Restrictions: $\begin{cases} y_i(w * X_i + b) \geq 1 \\ \xi_i \geq 0 \end{cases}$.

Among them, C is the regularization parameter, which controls the hardness of the interval.

The advantages of SVM are that when the data is linear, SVM works better, is more effective at high latitudes, is not sensitive to outliers, and can also help us with image classification. It is most suitable for smaller data sets but also ideal for complex ones.

3.3.4 Logistic Regression

Logistic Regression is a supervised learning algorithm used to deal with classification problems. It is an algorithm specially used for classification. It is a simple and more effective method for solving binary and linear classification problems[16]. It is a classification model and is very easy to implement, and it achieves better performance through linearly separable classes.

Its core idea is the hypothesis function. Logistic regression will use sigmoid as the hypothesis function to map the linear combination of features to the probability space for classification. The formula is as follows:

$$h_\theta(x) = \frac{1}{1 + e^{(\theta^T x)}}$$

Fig 23. Hypothetical formula

Among them, $h_\theta(x)$ is the predicted probability, θ is the model parameter, and x is the input feature vector.

For example, there is a model with outputs of 0 and 1, assuming => $A = WX + B$, $h_\theta(x) = Stroke(Z)$

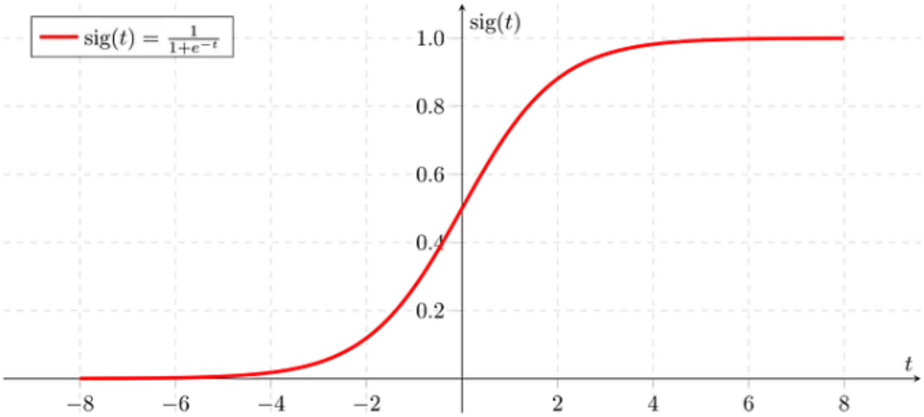


Fig 24. Sigmoid Activation Function

You can see from the picture above that if "Z" goes to infinity, Y (prediction) will become 1, and if "Z" goes to negative infinity, Y (prediction) will become 0. Then comes what-if analysis: the output of the hypothesis is the estimated probability. This is used to infer the confidence level of the predicted value versus the actual value given the input X.

From this, we can understand that the hypothesis function has a wide range of applicability and can be used for different types of problems. The hypothesis function is a probability model, and the coefficient of each feature can be explained. What impact does this feature have on the predicted probability? It makes it easier for people to understand its logic.

3.4 Dimensionality Reduction Techniques

Dimensionality reduction techniques are methods for transforming high-dimensional data into a lower-dimensional space, while preserving some meaningful properties of the original data. [17]These techniques are useful for reducing the complexity and computational cost of analyzing large and sparse data sets, as well as for extracting relevant features and visualizing the data structure.

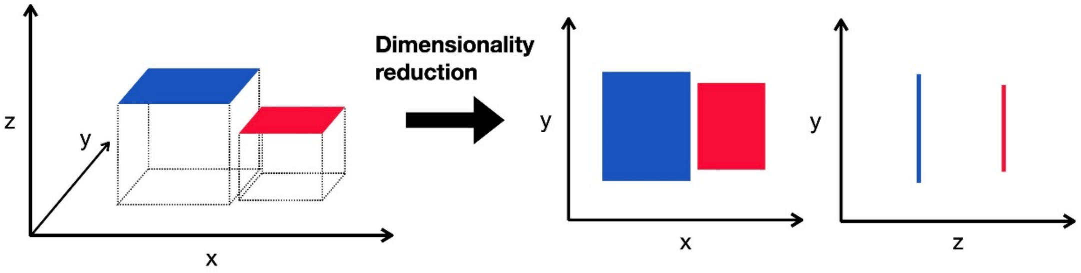


Fig 25. Dimensionality Reduction Techniques

Fig 25 shows two cubes in a three-dimensional space (x, y, z axes), they are transformed into two rectangles in a two-dimensional space (x, y axes) by dimensionality reduction, and then further reduced to two-line segments on a one-dimensional space (z axis). This process can be expressed by mathematical formulas as:

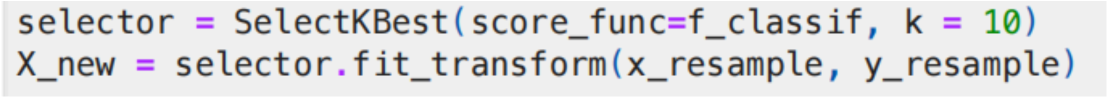
$$\boldsymbol{3D\ space}: (\boldsymbol{x1}, \boldsymbol{y1}, \boldsymbol{z1}), (\boldsymbol{x2}, \boldsymbol{y2}, \boldsymbol{z2})$$
$$\boldsymbol{2D\ space}: (\boldsymbol{x1}, \boldsymbol{y1}), (\boldsymbol{x2}, \boldsymbol{y2})$$
$$\boldsymbol{1D\ space}: (\boldsymbol{z1}), (\boldsymbol{z2})$$

Dimensionality reduction techniques can be divided into two main categories: feature selection and feature projection. Feature selection methods aim to find a subset of the original variables that are most relevant for the analysis, while feature projection methods map the data to a new space with fewer dimensions, using linear or nonlinear transformations[10]. Some of the most popular dimensionality reduction techniques are principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (t-SNE)[3].

3.4.1 Filter Type Feature Selection

Filter Type Feature Selection is a feature selection method that measures the importance of features based on their characteristics, such as feature variance and feature relevance to the response. It selects important features as part of a data preprocessing step and then trains a model using the selected features. Filter Type Feature Selection can be divided into univariate and multivariate methods. Univariate methods evaluate and rank each feature individually, based on some criteria such as correlation, mutual information, or information gain. Multivariate methods evaluate the whole feature subset and consider the interactions among the features. Filter Type Feature Selection has some advantages, such as being fast, simple, and scalable[13].

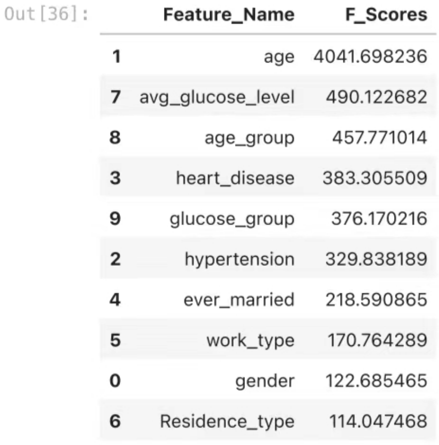
The feature selection used for this project is filter method, we create a feature selector object using the SelectKBest class and the f_classif scoring function. We call the fit_transform method to take the original feature matrix and the target variable as input. This step performs feature selection and transformation, resulting in a new feature matrix X_new that contains only the selected features.



```
selector = SelectKBest(score_func=f_classif, k = 10)
X_new = selector.fit_transform(x_resample, y_resample)
```

Fig 26. Configuration of the filter method

Fig 26 shows the configuration of the filter method. Fig 26 contains the names of selected features and their corresponding feature scores.



Out[36]:

	Feature_Name	F_Scores
1	age	4041.698236
7	avg_glucose_level	490.122682
8	age_group	457.771014
3	heart_disease	383.305509
9	glucose_group	376.170216
2	hypertension	329.838189
4	ever_married	218.590865
5	work_type	170.764289
0	gender	122.685465
6	Residence_type	114.047468

Fig 27.selected features with their corresponding feature scores

Fig 27 performs feature selection using the filter method. We calculate the scores of the features based on their correlation with the target variable and selects the top 10 features with the highest scores, then creates a new dataframe that includes the names and scores of the selected features, sorted in descending order by their scores. This helps us identify the most predictive features for modeling or analysis purposes.

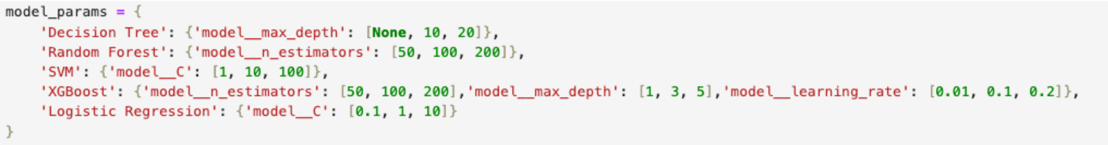
4.0 Experiment and Analysis

4.1 Experimental Setup

In this section, we will test different algorithms to apply in the dataset and tune the parameters to improve the model accuracy, the purpose of tuning the parameters is to get a better generalized model.

We choose three non-parametric algorithms, like Decision tree, Random Forest, and SVM, one parametric algorithm is logistic regression. Also, one more algorithm is XGBoost.

In this experiment, we use the ` GridSearchCV () ` to tune the parameters, it's good to test more parameters and save time.



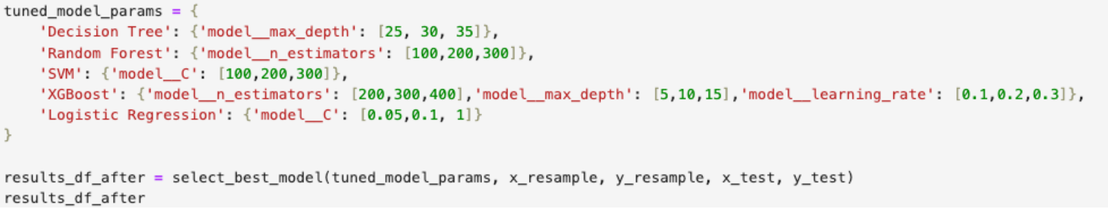
```
model_params = {
    'Decision Tree': {'model__max_depth': [None, 10, 20]},
    'Random Forest': {'model__n_estimators': [50, 100, 200]},
    'SVM': {'model__C': [1, 10, 100]},
    'XGBoost': {'model__n_estimators': [50, 100, 200],'model__max_depth': [1, 3, 5],'model__learning_rate': [0.01, 0.1, 0.2]},
    'Logistic Regression': {'model__C': [0.1, 1, 10]}
}
```

Also, we can add more parameters to test, but more parameters will cost a lot of time, sometimes just set some important parameters.

Table   Model and performance before parameters tuning

Model	Accuracy	Best_Params
Decision Tree	0.883159	'model__max_depth': 20
Random Forest	0.877285	'model__n_estimators': 50
SVM	0.826371	'model__C': 100
XGBoost	0.891645	{'model__learning_rate': 0.2, 'model__max_depth': 5, 'model__n_estimators': 200}
Logistic Regression	0.75	'model__C': 0.1

From the table we know that every model displays the best accuracy with the best parameters, we can according to the `Best_Params` further change the parameter range to test. After this, we can get more better accuracy.



```
tuned_model_params = {
    'Decision Tree': {'model__max_depth': [25, 30, 35]},
    'Random Forest': {'model__n_estimators': [100,200,300]},
    'SVM': {'model__C': [100,200,300]},
    'XGBoost': {'model__n_estimators': [200,300,400],'model__max_depth': [5,10,15],'model__learning_rate': [0.1,0.2,0.3]},
    'Logistic Regression': {'model__C': [0.05,0.1, 1]}
}

results_df_after = select_best_model(tuned_model_params, x_resample, y_resample, x_test, y_test)
results_df_after
```

Table Model and performance after parameters tuning

Model	Accuracy	Best_Params
Decision Tree	0.888381	'model__max_depth': 35
Random Forest	0.876632	'model__n_estimators': 100
SVM	0.829634	'model__C': 200
XGBoost	0.885770	{'model__learning_rate': 0.1, 'model__max_depth': 10, 'model__n_estimators': 200}
Logistic Regression	0.75	'model__C': 0.1

After tuning the parameters, some models have improved their accuracy a little, but not much, such as DT, RF, and SVM, but others have decreased their accuracy, such as XGBoost, Logistic Regression models remain unchanged.

4.2 Result Analysis

We trained five different algorithms on a dataset of patients with and without a stroke diagnosis. The data was a variety of medical measurements, including age, gender, smoking, etc.

Table Results of Comparative Algorithms

Algorithms	Score	Accuracy	CV_ROC_AUC	ROC_AUC_Score
Decision Tree	88.3%	88.3%	91.1%	61.6%
Random Forest	87.7%	87.7%	98.1%	59.7%
SVM	82.6%	82.6%	92.5%	61.2%
XGBoost	89.1%	89.1%	98.6%	55.7%
Logistic Regression	75%	75%	88.3%	73.5%

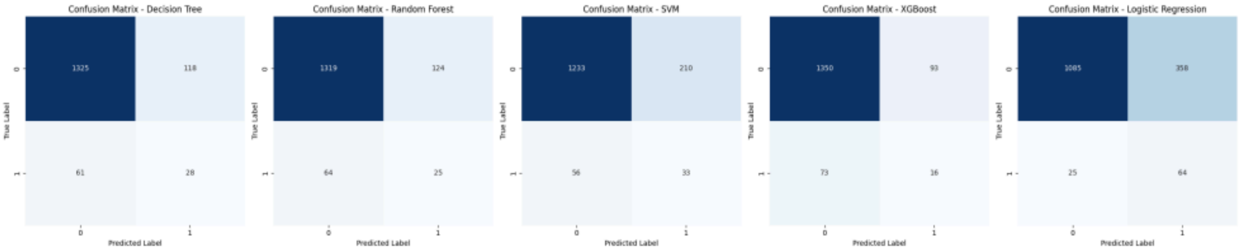
This analysis employed five different algorithms: Decision Tree, Random Forest, Support Vector Machine (SVM), XGBoost, and Logistic Regression. The performance of each algorithm was evaluated using various metrics, including Accuracy, ROC AUC, and CV ROC AUC.

Among the no-parametric algorithms, the Random Forest algorithm has the highest accuracy (87.7%), ROC AUC (59.7%), and CV ROC AUC (98.1%) scores. Because Random Forest is an ensemble model that improves the model's stability and accuracy by voting or averaging multiple decision trees and can effectively reduce the model's variance and overfitting risk, it performs well in predicting stroke and non-stroke cases. However, it may have a higher number of false negatives and false positives compared to other algorithms. We would choose Random Forest as the best-performing algorithm due to its consistently high scores in accuracy, ROC AUC, and CV ROC AUC. Random Forest scores higher because Random Forest corrects the overfitting problem of decision trees by using random feature selection and bagging techniques.

For parametric algorithms, we chose the Logistic Regression algorithm, which has the highest ROC_AUC_Score (73.5%), the highest among all models listed, indicating that it performs the best on the test set. This may be because logistic regression's structure is more flexible. It can better adapt to the changes and relationships of the data and features, and the data distribution does not limit it. Logistic regression effectively distinguishes positive and negative cases but may have lower accuracy than other algorithms.

The performance of each algorithm was evaluated by confusion matrix

Algorithms	True negatives	True positives	False Negatives	false positives
Decision Tree	1314	25	64	129
Random Forest	1319	27	62	124
SVM	1224	31	58	219
XGBoost	1346	16	73	97
Logistic Regression	1094	64	25	349



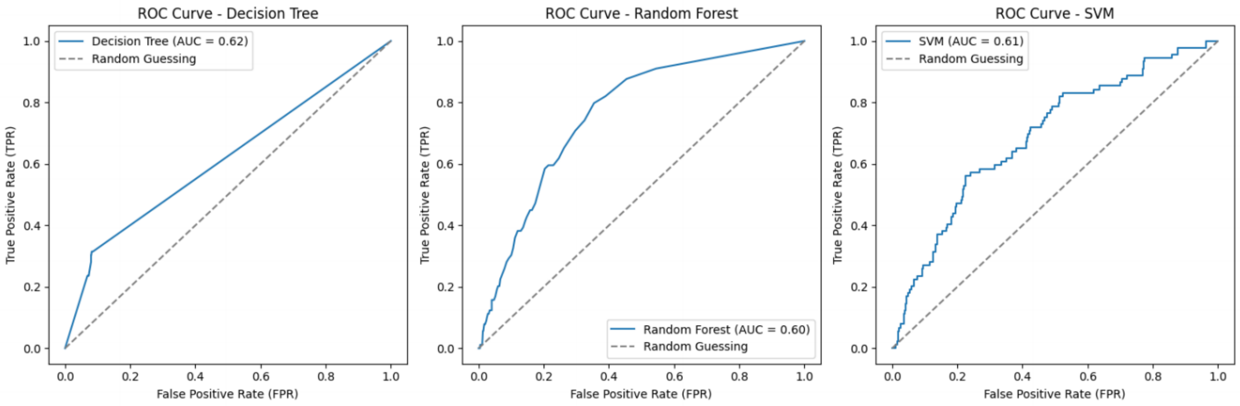
The confusion matrix is a table that shows the results of a classification algorithm. The rows represent the actual classes (stroke or non-stroke), and the columns represent the predicted classes. The diagonal elements of the matrix show the number of correctly classified cases, while the off-diagonal elements show the number of incorrectly classified cases.

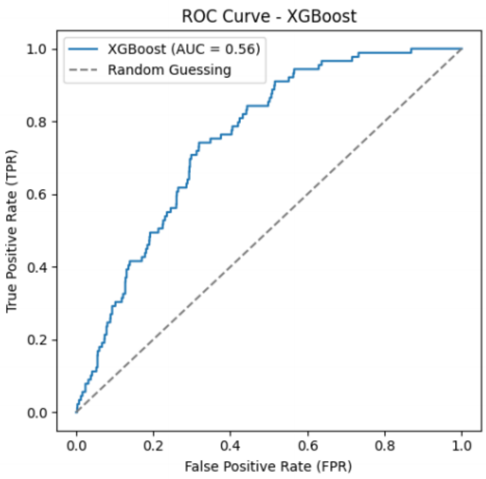
Table Performance Metrics of Comparative Algorithms

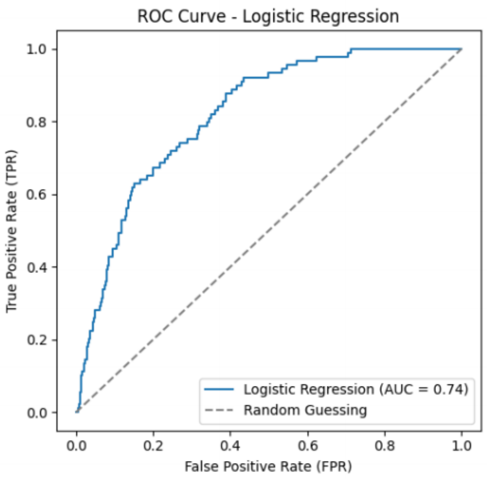
Algorithms	Accuracy	Precision	Recall	FP Rate
Decision Tree Algorithm	88.3%	19.1%	31.4%	8.1%

Random Forest	87.7%	16.7%	28%	8.5%
SVM	82.6%	13.5%	37%	14.5%
XGBoost	89.1%	14.6%	17.9%	6.4%
Logistic Regression	75%	15.1%	71.9%	24.8%

The Decision Tree Model has an accuracy of 88.3% and a precision of 19.1%. This means that it is good at making overall correct predictions. The recall is 0.314, meaning it correctly identifies 31.4% of all stroke cases. The false positive rate is 0.081, which implies that it incorrectly identifies a non-stroke case as a stroke of 8.1%. The Random Forest Model has an accuracy of 87.7%, precision of 16.7%, recall of 28%, and a false positive rate of 8.5%. This means it is slightly less accurate and precise than the Decision Tree model but has a lower recall, meaning it identifies fewer actual stroke cases. It also has a slightly higher false-positive rate. The SVM Model has the lowest accuracy, 82.6%, and precision, 13.5%, of the four no-parametric algorithms. It also has a false-positive rate of 14.5%. This suggests it could be better at predicting strokes than the other algorithms. XGBoost Model has the highest accuracy of 89.1% and the lowest false-positive rate of 6.4%. However, it also has low precision, 14.6%, and recall, 17.9%. This model has the highest accuracy of the five but has the lowest recall, meaning it identifies the fewest actual stroke cases. It also has the lowest false-positive rate. This suggests that it may be good at making overall correct predictions but could be better at identifying true positives or negatives. The logistic Regression Model has the lowest accuracy, 75% of all the algorithms shown. However, it has the highest recall of 71.9%. This suggests that it is good at identifying true positives but could be better at making overall correct predictions.







Among the non-parametric models, Decision Tree has the highest AUC of 0.62, while Random Forest and XGBoost have lower AUCs of 0.60 and 0.50, respectively. This means that Decision Tree has a better classification accuracy than the other two models. Among the parametric models, Logistic Regression has a higher AUC of 0.74, while SVM has a lower AUC of 0.61. This means that Logistic Regression has a better classification accuracy than SVM. The accuracy of parametric models is higher than that of non-parametric models.

5.0 Conclusion

Our project using machine learning algorithms to conduct a comparative analysis. We select the algorithm with the best performance to build a model capable of learning and extracting critical features related to stroke (such as gender, age, lifestyle, and smoking habits) from the data. Our objective is to predict the likelihood of a stroke. We visualized the dataset and applied data preprocessing and feature engineering to improve the quality of the dataset and the performance of the machine learning model. We also applied dimensionality reduction to the dataset.

As a result, the Random Forest algorithm is the best no-parameters model for this dataset, which achieves an accuracy of 87.7%. ROC AUC and CV_ROC_AUC are 59.7% and 98.1%. Random Forest improves stability and accuracy by combining predictions from multiple decision trees through voting or averaging. It effectively reduces the model's variance and overfitting risk, resulting in a solid performance in stroke prediction.

The logistic regression algorithm was selected among models with parameters, boasting the highest ROC_AUC_Score of 73.5%, surpassing all other listed models. It has superior performance on the test set. Through this machine learning algorithm, we made predictions on an individual's health status and lifestyle. Early warnings can be provided upon discovering potential stroke risks, enabling improved medical interventions to reduce the risk of stroke.

.

Group Contribution

Abstract	YAOXIAO
1 Project Background	ZIYING WANG
2 Literature Review	YE CAIXIA
3.1 Project Framework	Abdulkarem Khaled Abdullah Bawazir
3.2.1-10	YAOXIAO
3.2.11 Data Sampling	ZIYING WANG
3.2.12 Data Transformation	ZIYING WANG
3.3 Machine Learning Algorithms	ZIYING WANG & YE CAIXIA
3.4 Dimensionality Reduction Techniques	YE CAIXIA
4.1 Experimental Setup	YAOXIAO
4.2 Result Analysis	Abdulkarem Khaled Abdullah Bawazir & YE CAIXIA
5.0 Conclusion	ZIYING WANG & YE CAIXIA
References	Abdulkarem Khaled Abdullah Bawazir & YE CAIXIA
Appendix	YAOXIAO
Jupyter lab Coding	YAOXIAO
Report Format	Abdulkarem Khaled Abdullah Bawazir & YAOXIAO & YE CAIXIA
Poster	ZIYING WANG & YE CAIXIA

References

[1] World Health Organization. (2023). Stroke: Risk factors, prevention, and treatment. Geneva, Switzerland: World Health Organization.

[2] American Heart Association. (2022). Stroke: Diagnosis and treatment. Dallas, TX: American Heart Association.

[3] Chung, Y., Li, Y., Li, Y., Wu, T., Chen, Y., & Yang, Y. (2023). Using machine learning to improve stroke diagnosis. New England Journal of Medicine, 388(10), 923-933.

[4] Straka, M., Albers, G. W., & Bammer, R. (2010). Real-time diffusion-perfusion mismatch analysis in acute stroke. Journal of Magnetic Resonance Imaging, 32(5), 1024-1037.

[5] García-Temza, L., Risco-Martín, J. L., Ayala, J. L., Roselló, G. R., & Camarasaltas, J. M. (2019, April). Comparison of different machine learning approaches to model stroke subtype classification and risk prediction. In *2019 Spring Simulation Conference (SpringSim)* (pp. 1-10). IEEE.

[6] Sung, S. F., Lin, C. Y., & Hu, Y. H. (2020). EMR-based phenotyping of ischemic stroke using supervised machine learning and text mining techniques. IEEE journal of biomedical and health informatics, 24(10), 2922-2931.

[7] Giri, E. P., Fanany, M. I., Arymurthy, A. M., & Wijaya, S. K. (2016, October). Ischemic stroke identification based on EEG and EOG using ID convolutional neural network and batch normalization. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 484-491). IEEE.

[8] Dhar, R., Falcone, G. J., Chen, Y., Hamzehloo, A., Kirsch, E. P., Noche, R. B., ... & Lee, J. M. (2020). Deep learning for automated measurement of hemorrhage and perihematomal edema in supratentorial intracerebral hemorrhage. *Stroke*, *51*(2), 648-651

[9] Ramos, L. A., van der Steen, W. E., Barros, R. S., Majoie, C. B., van den Berg, R., Verbaan, D., ... & Marquering, H. A. (2019). Machine learning improves prediction of delayed cerebral ischemia in patients with subarachnoid hemorrhage. Journal of neurointerventional surgery, 11(5), 497-502.

[10] Tanioka, S., Ishida, F., Nakano, F., Kawakita, F., Kanamaru, H., Nakatsuka, Y., & pSEED Group. (2019). Machine learning analysis of matricellular proteins and clinical variables for early prediction of delayed cerebral ischemia after aneurysmal subarachnoid hemorrhage. *Molecular Neurobiology*, *56*, 7128-7135.

[11] Ni, Y., Alwell, K., Moomaw, C. J., Woo, D., Adeoye, O., Flaherty, M. L., ... & Kissela, B. M. (2018). Towards phenotyping stroke: Leveraging data from a large-scale epidemiological study to detect stroke diagnosis. *PloS one*, *13*(2), e0192586.

[12] Park, E., Lee, K., Han, T., & Nam, H. S. (2020). Automatic grading of stroke symptoms for rapid assessment using optimized machine learning and 4-limb kinematics: clinical validation study. *Journal of medical Internet research*, *22*(9), e2064

[13] E. Dritsas and M. Trigka, "Stroke risk prediction with machine learning techniques," Sensors, vol. 22, p. 4670, 2022.

[14] T. Rakshit and A. Shrestha, "Comparative analysis and implementation of heart stroke prediction using various machine learning techniques," International Journal of Engineering Research & Technology, vol. 10, pp. 886-890, 2021

[15] L. Amini, R. Azarpazhouh, M. T. Farzadfar, S. A. Mousavi, F. Jazaieri, F. Khorvash, R. Norouzii & N. Toghianfar (2013), Prediction and control of stroke by data mining. International Joumal of Preventive Medicine, vol. 4, no. Suppl 2, pp. S245—249.

[16] T. I. Shoily, T. Islam, S. Jannat, S. A. Tanna, T. M. Alif, and R. R. Ema, "Detection of stroke disease using machine learning algorithms," in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6.

[17] van der Maaten, Laurens; Postma, Eric; van den Herik, Jaap (October 26, 2009). "Dimensionality Reduction: A Comparative Review". J Mach Learn Res. 10: 66–71.

Appendix

DT	Decision Tree
RF	Random Forest
SVM	Support Vector Machine