NATIONAL ECONOMICS UNIVERSITY

FACULTY OF ECONOMICS AND MATHEMATICS

-------***-------

★ ★ ★

BACHELOR THESIS

★ ★ ★

PREDICTING STROKE USING MACHINE LEARNING

Student : **Tran Thi Thu Trang**

Student ID : **11208164**

Class : **DSEB 62**

Supervisor : **Ph.D. Nguyen Manh The**

Ha Noi, March 2024.

Acknowledgement

I'd want to take this occasion to offer my heartfelt gratitude to everyone who helped me complete this bachelor thesis.

First and foremost, I want to thank my supervisor, Ph.D. Nguyen Manh The, for providing me with invaluable direction, assistance, and advice throughout the study process. His insightful remarks, constructive criticism, and unwavering support were invaluable in shaping and improving this thesis. I genuinely appreciate his devotion to my academic development.

I would also want to express my heartfelt gratitude to the teachers at the Faculty of Mathematical Economics; their zeal for teaching and willingness to join in stimulating conversations has motivated me to work harder on my studies.

My heartfelt appreciation goes to my family, friends, and coworkers at FPT Software for their unwavering support, understanding, and love. This journey was made more enjoyable and satisfying due to their excellent insights and persistent contributions.

Finally, no matter how large or little your contribution to this study has been, I want to extend my heartfelt appreciation to you all. Your aid, motivation, and contributions have been invaluable, and I am grateful that you have been a part of my academic journey.

Thank you.

Tran Thi Thu Trang

Contents

LIST OF ABBREVIATIONS

ML	Machine Learning
EDA	Exploratory Data Analysis
LR	Logistics Regression
RF	Random Forest
DT	Decision Tree
KNN	K-Nearest Neighbor
TIA	Transient Ischemic Attack
DALY	Disability Adjusted Life Years
ANN	Artificial Neural Network
SGD	Stochastic Gradient Descent
ROC	Receiver Operating Characteristics
AUC	Area Under the Curve
MLE	Maximum Likelihood Estimation
SMOTE	Synthetic Minority Oversampling Technique
MI	Mutual Information

LIST OF TABLE

LIST OF FIGURES

ABSTRACT

Stroke is the second greatest cause of death worldwide and the third leading cause of disability. One in every four people is at risk of having a stroke in their lifetime. Early awareness of the various stroke hazards can reduce the risk of a stroke. Predicting a stroke is critical for avoiding serious health consequences or death. The objective of my research paper is to detect stroke risk in patients by using machine learning classification algorithms on features derived from available dataset. It will help us determine which algorithm is best suited and accurate for stroke prediction. Models used for stroke prediction included Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, and Random Forest, and experimental findings were analyzed using Jupyter Notebook. Among the supervised machine learning methods described, the Random Forest technique has the best classification accuracy (CA) of 94.9%. As a result, Random Forest is an almost perfect classifier for stroke prediction, allowing doctors and patients to prescribe and recognize potential strokes early on.

Keywords: Machine Learining, healthy, stroke, Logistic Regression, KNN, Decision Tree, Random Forest.

CHAPTER 1: INTRODUCTION

Human existence relies on many bodily components and their functions. Stroke is a serious disease that takes a person's life. A stroke happens when a blockage or hemorrhage in the blood arteries disrupts or lowers the flow of blood to the brain. When this occurs, the brain does not receive enough oxygen or nutrients, causing brain cells to die. Stroke is a cerebrovascular disorder. This implies it affects the blood arteries that transport oxygen to the brain. If the brain does not receive enough oxygen, harm may begin to develop. Although many strokes are curable, some might result in disability or death.

Stroke symptoms appear suddenly and can vary between individuals. Indicators of a stroke include dizziness, trouble speaking or comprehending, blurred or lost vision in one or both eyes, numbness in the arms, legs, or face, typically on one side of the body, or issues with coordination or balance, walking, or mobility. Other signs include fainting or seizures and severe, unexplained headaches. Other unusual stroke symptoms may involve sudden nausea, temporary loss of consciousness, fainting, confusion, seizures, or coma.

Human state after stroke varies on the type of stroke, which is classified into three categories: Transient Ischemic Attack (TIA), Ischemic Stroke, Hemorrhagic Stroke.

A transient ischemic attack (TIA) is a condition in which blood supply to the brain is momentarily disrupted. Patients often recover from this form of stroke in minutes. According to the Centers for Disease Control and Prevention (CDC), one-third of persons who experience a transient ischemic attack (TIA) have suffered a stroke within a year.

Ischemic stroke affects 87% of stroke sufferers. This is the most prevalent form of stroke. It happens when a major blood artery in the brain becomes clogged, which can

be caused by a blood clot. It may also get clogged due to an accumulation of fat and cholesterol.

A hemorrhagic stroke occurs when a brain artery ruptures and spills blood into nearby tissues. Damage to brain cells and tissue occurs when the blood pressure of the arteries is greater than that of the skull. About 13% of strokes are hemorrhagic.

According to the 2022 Global Stroke Factsheet, the lifetime risk of experiencing a stroke has risen by 50% over the past 17 years, with one in four individuals now expected to suffer from a stroke. Between 1990 and 2019, there was a 70% increase in stroke incidence, a 43% rise in stroke-related fatalities, a 102% increase in stroke prevalence, and a 143% increase in Disability Adjusted Life Years (DALYs) attributed to stroke. Particularly noteworthy is the disproportionate burden of stroke borne by low- and lower-middle-income countries, where 86% of stroke fatalities and 89% of DALYs are recorded. This disproportionate burden faced by low-income households in lower and lower-middle-income nations presents an unprecedented challenge.

There are several reasons why patients have stoke. According to the National Heart, Lung, and Blood Institute, food, inactivity, tobacco, alcohol, personal history, health history, and complications are the most common variables that contribute to strokes.

The foundation of the modern era is machine learning, which is used to predict a variety of issues in their early stages. For instance, as stroke is one of the major diseases that can be cured if anticipated in its early stages, many diseases can be prevented if predicted early.

Overall, machine learning is crucial to the health care industry when it comes to disease diagnosis and prediction. My goal is to train the algorithm by providing it with a training dataset and detect patients at risk of stroke using different types of classification algorithms. Then these algorithms can predict the data by comparing the

provided data with the training datasets. In this study, I choose the following machine learning methods: Logistic regression, KNN, Decision Tree, Random Forest to train with the stroke prediction dataset.

CHAPTER 2: LITERATURE SURVEY

The research efforts focus on information discovery through the use of machine learning classifiers. To date, numerous studies have explored stroke prediction using various machine learning techniques and methodologies. The accomplishments of some research projects are detailed below:

Leila Amini et al. (2013) conducted a study to predict the incidence of stroke at Esfahan Al-Zahra and Mashhad Ghaem hospitals during 2010-2011. They classified a data set containing 50 risk factors for stroke such as history of cardiovascular disease, diabetes, hyperlipidemia, smoking and alcohol consumption of 807 healthy and sick subjects collected using how to use a standard checklist. To analyze the data, they used data mining techniques, K -nearest neighbors, and C4.5 decision trees using WEKA. The accuracy of the two models is 94.18% and 95.42% respectively.

Focusing on predicting stroke, Chun-An Cheng et al. (2014) conducted a study to predict the outcome of ischemic stroke patients after intravenous thrombolysis. They collected retrospective data of 82 ischemic stroke patients at Tri-service General Hospital and used STATISTICA 10 software to select the best artificial neural network. The accuracy of ANN model 1 is 79.27% and that of ANN model 2 is 95.12%.

In another study, Priya Govindarajan et al. (2020) used a combined of text mining tools and machine learning algorithms to classify stroke in 507 individuals collected from Sugaram General Hospital. The processed data is then fed into various machine learning algorithms such as artificial neural networks, support vector machines, boosting and bagging, and random forests. Among these algorithms, the artificial neural network trained using stochastic gradient descent (SGD) algorithm outperforms other algorithms with classification accuracy higher than 95%.

Teerapat Kansadub et al. (2015) conducted a study to determine the risk of stroke. In this study, they used three Naive Bayes classification algorithms, decision trees, and neural networks to evaluate the accuracy and area under the ROC curve (AUC) in their study. They classify all these algorithms into Decision Tree which is most accurate and Naive Bayes which is best in AUC.

Adam et al. (2016) conducted a study to determine the classification of ischemic stroke based on a dataset of 400 cases collected from different hospitals in Sudan. They classified using two models: k-nearest neighbor (KNN) method and decision tree technique. The results showed that the decision tree method was found by medical experts to be more useful when used to classify strokes.

Inheriting and learning from the above research, this project will focus on building a machine learning model including Logistic Regression, Decision Tree, KNN among selected features to predict the patient's risk of stroke. Multiply by the highest predicted result.

CHAPTER 3: THEORY OF MODEL

3.1. Logistic Regression

Logistic regression is a supervised machine learning method utilized for binary classification tasks. It estimates the likelihood of a particular outcome or observation. The model generates a binary outcome with two potential values, such as yes/no, 0/1, or true/false.

This technique assesses the relationship between one or more independent variables and categorizes data into discrete classes. Logistic regression is extensively employed in predictive modeling, where it calculates the mathematical probability of whether an instance belongs to a specific category or not.

Logistic regression derives its name from the logistic function, which lies at the heart of this method. Also known as the sigmoid function, the logistic function was originally developed by statisticians to model population growth dynamics in ecology. It exhibits a characteristic S-shaped curve, rising rapidly and plateauing at the carrying capacity of the environment. The sigmoid function is capable of transforming any real-valued number into a value between 0 and 1, but it never precisely reaches these limits.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Where e is the base of the natural logarithms (Euler's number or the EXP() function in your spreadsheet) and z is the actual numerical value that you want to transform.

The graph of the sigmoid function illustrates this behavior, mapping input values to a probability range between 0 and 1.

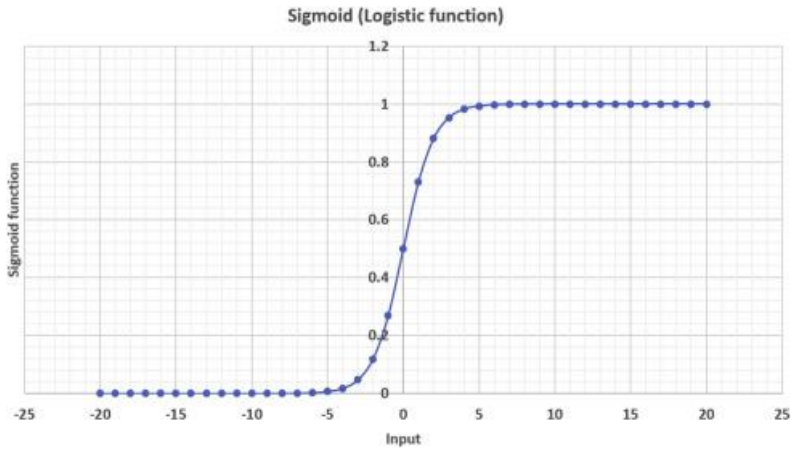


Figure 1. Sigmoid function

The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables.

In linear regression, the Mean Squared Error (MSE) is utilized, representing the difference between y_predicted and y_actual. This metric is derived from the maximum likelihood estimator. However, in logistic regression, the relationship between the predicted values ($\hat{y}$) and the input variables (z) is nonlinear, as expressed by the sigmoid function ($\hat{Y} = \frac{1}{1+e^{-z}}$). If this nonlinear function is employed within the MSE equation, it results in a non-convex graph with numerous local minima, as illustrated.

This cost function poses a significant challenge as it leads to results with local minima, potentially causing us to overlook the global minimum and resulting in increased error. To address this issue, logistic regression employs a different cost function known as log loss, which is also derived from the Maximum Likelihood Estimation (MLE) method. In machine learning, MLE's primary goal, especially in logistic regression, is to determine parameter values that maximize the likelihood function.

Assuming that the probability of a data point belonging to a specific class follows a Bernoulli distribution. It can be expressed as follows, with $y_i \in \{0,1\}$:

$$P(y_i|x_i, w) = P(y = y_i)^{y_i} * \big(1 - P(y = y_i)\big)^{1-y_i}$$

Direct optimization of the likelihood function is challenging. To overcome this, the logarithm is applied to convert the product to a sum, simplifying the optimization process. This changes the problem to the optimization of the Log Likelihood function.

$$L = Log(P(y|x, w) = \sum_{i=1}^{n} \log\big(P(y_i|x_i, w)\big)$$

$$L = \sum_{i=1}^{n} y_i * \log\big(P(y = y_i)\big) + (1 - y_i) * \log\big(1 - P(y = y_i)\big)$$

$$L = \sum_{i=1}^{n} y_i * \log(\hat{y_i}) + (1 - y_i) * \log(1 - \hat{y_i})$$

We need to *w* that maximizes the log-likelihood function:

$$argmax_w \sum_{i=1}^{n} y_i * \log\Big(\sigma(w^T x^i)\Big) + (1 - y_i) * \log\Big(1 - \sigma(w^T x^i)\Big)$$

where $\sigma(w^T x^i)$ is the sigmoid function.

In order to determine the appropriate value for w, the Gradient Descent optimization method is commonly employed. Gradient descent is a crucial algorithm used in logistic regression to find the optimal parameters for classification. By iteratively adjusting these parameters based on the gradient of the cost function, logistic regression can efficiently classify data points into two or more classes.

To implement the Gradient Descent method, the initial w value is randomly generated, and it is updated after each iteration using the following formula (with L as learning rate):

$$w_{new} = w_{old} - \alpha x^T \frac{dL}{dw}$$

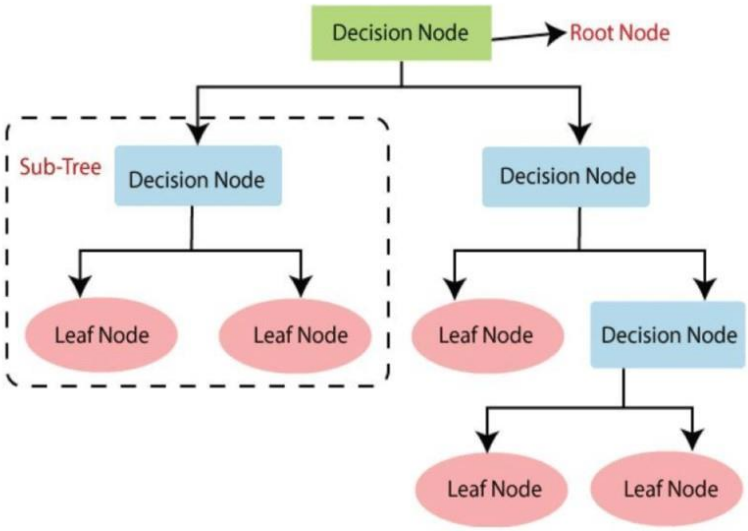
$$w_{new} = w_{old} - \alpha\, x^T (y - \hat{y})$$

3.2. Decision tree

A decision tree is a non-parametric supervised learning algorithm for classification and regression tasks. It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes , forming a hierarchical, tree-like structure.

A decision tree represents a hierarchical model employed in decision support systems to visualize decisions and their possible outcomes, encompassing chance events, resource costs, and utility considerations. This algorithmic model utilizes conditional control statements.

Figure 2. Decision Tree algorithm



In the decision tree, the green, blue, and red cells in *Figure 2* are called nodes. The nodes representing the output (blue and red) in a decision tree are referred to as leaf nodes or terminal nodes. Nodes representing questions are non-leaf nodes. The top non-leaf node, also known as the first question, is called the root node. Non-leaf nodes

typically have two or more child nodes, which can either be leaf nodes or other non-leaf nodes. Child nodes sharing the same parent are called sibling nodes. When all non-leaf nodes have only two child nodes, the decision tree is referred to as a binary decision tree. Questions in a binary decision tree can be framed as true or false questions. Decision trees with leaf nodes having multiple child nodes can also be transformed into binary decision trees, as most questions can be simplified into true or false questions.

The resulting tree structure represents a series of if-else rules that can be used for classification or regression tasks. A visual representation of a decision tree is shown in the following figure:

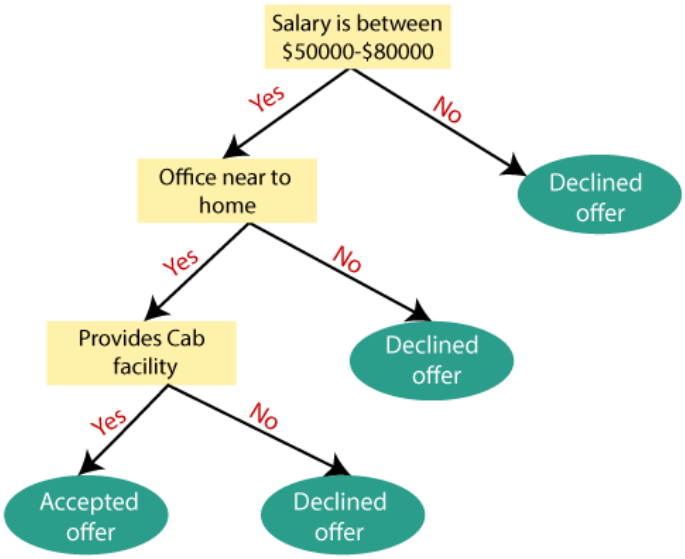


Figure 3. Example of a Decision Tree

The process of creating a decision tree involves the following steps:

In a decision tree, the process for predicting the class of a given dataset begins at the root node. The algorithm compares the value of the root attribute with the attribute in the record (real dataset) and, based on the comparison, follows the appropriate branch to the next node. At each subsequent node, the algorithm again compares the attribute

value with the corresponding sub-node and proceeds further. This process continues until a leaf node is reached. The detailed steps of the process are as follows:

Step1: Begin the tree with the root node, denoted as S, which encompasses the entire dataset.

Step2: Determine the best attribute in the dataset using an Attribute Selection Measure (ASM).

Step3: Partition the dataset S into subsets based on the possible values of the best attribute.

Step4: Create a decision tree node containing the best attribute.

Step5: Recursively construct new decision trees using the subsets of the dataset generated in step 3. Continue this process until a stage is reached where further classification is not possible, and designate the final node as a leaf node.

One critical question is how to select the optimum characteristic and split point for each node. The Gini index and Entropy are two often used measures for this purpose.

Entropy

Entropy is a term in Thermodynamics, a measure of change, chaos or randomness. In 1948, Shannon expanded the concept of Entropy to the field of research and statistics with the following formula:

Given a probability distribution of a discrete variable x can get n different values $x_1, x_2, \dots, x_n$. Suppose that the probability to x get these values $p_i = p(x = x_i)\ with\ 0 \le p_i \le 1, \sum_{i=1}^{n} p_i = 1$. The symbol for this distribution is $p = (p_1, p_2, \dots, p_n)$. The entropy of this distribution is defined as:

$$H(p) = -\sum_{i=1}^{n} p_i * \log(p_i)$$

Consider a problem with different C classes. Suppose we are working with a non-leaf node containing a set S of data points, with the number of elements denoted as $|S| = N$. Suppose further that out of these N data points, N_c (where c=1,2,… ,c) points belong to class C. The probability that each data point falls into class C is approximately equal $\frac{N_c}{N}$ (using maximum likelihood estimation). Therefore, the entropy at this node is calculated by the following formula:

$$H(S) = -\sum_{c=1}^{C} \frac{N_c}{N} \log\left(\frac{N_c}{N}\right)$$

Next, let's say the selected attribute is x. Based on x, the data points in S are divided into K child node $S_1, S_2, \dots, S_K$ with the number of points in each child node respectively $m_1, m_2 \dots, m_K$. We define the weighted entropy of the child nodes as follows:

$$H(x, S) = \sum_{k=1}^{K} \frac{m_k}{N} H(S_k)$$

This weighting is important because nodes often have different numbers of points.

Gini index

The Gini index is another measure used to evaluate the degree of inequality in the distribution of classes, besides the entropy function. It quantifies the impurity of a node and is calculated as follows.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

With $\sum_{i=1}^{C} p_i = 1$ where p_i is the probability of an instance being classified into a particular class. The Gini index subtracts the sum of the squared probabilities of each class from 1. This index ranges from 0 (perfectly pure) to 1 (maximum impurity), where a lower Gini index indicates a more homogenous node.

Similar to the entropy function, the purity after each split scenario is determined by calculating the Gini index:

$$Gini(x,S) = \sum_{k=1}^{K} \frac{m_k}{N} \times Gini(S_k)$$

Decision trees are a basic yet common algorithm. This method is commonly utilized because to its advantages. The model provides easy-to-understand rules for the reader, resulting in a collection of rules with each leaf branch representing a rule of the tree. There is no need to normalize or generate fake variables for input data that contains missing values. Can deal with numerical and categorical data. Capable of handling large amounts of data.

However, decision trees do have limitations. Your data is critically important to the decision tree model. The structure of the decision tree model can vary substantially even if the data set is just slightly altered. Overfitting issues are common with decision trees. If a deep enough tree is constructed, it is possible for each leaf node to contain a small number of samples from a specific class, resulting in excessive model complexity. To mitigate overfitting, termination criteria such as maximum depth, minimum sample split, or pruning can be employed to reduce the risk.

Nevertheless, the introduction of such criteria may increase the bias error of the model.

3.3. Random Forest

Random forest, a popular machine learning algorithm developed by Leo Breiman and Adele Cutler, merges the outputs of numerous decision trees to produce a single outcome. Its popularity stems from its user-friendliness and versatility, making it suitable for both classification and regression tasks.

The Random Forest algorithm's popularity stems from its user-friendly nature and versatility, allowing it to effectively address both classification and regression

problems. Its strength lies in its capability to manage complex datasets and alleviate overfitting, rendering it invaluable for a range of predictive tasks in machine learning. The fundamental concept behind Random Forest is depicted *Figure 4*.

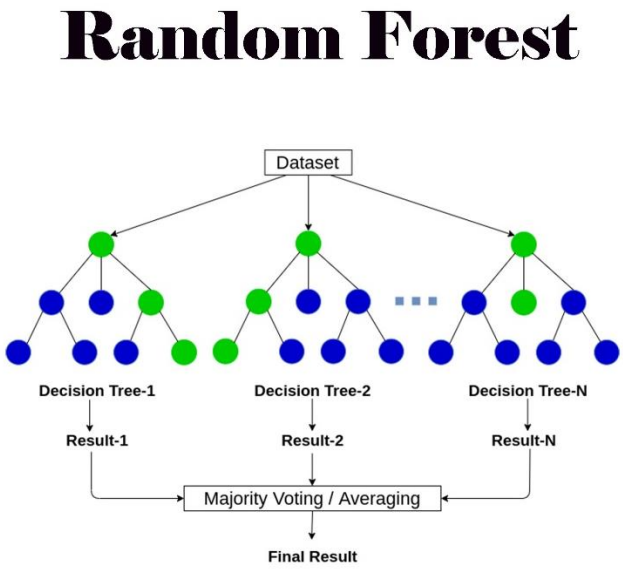


Figure 4. Example of a Random Forest model's workflow

The Random Forest algorithm operates through several key steps:

- Ensemble of Decision Trees: Random Forest utilizes ensemble learning by creating a collection of Decision Trees. Each tree functions as an individual expert, specializing in a particular aspect of the data.
- Random Feature Selection: To ensure diversity among the decision trees, Random Forest employs random feature selection. During the training of each tree, a random subset of features is chosen.
- Bootstrap Aggregating or Bagging: Bagging is a core technique in the training strategy of Random Forest. It involves generating multiple bootstrap samples from the original dataset, where instances are sampled with replacement. This process creates varied subsets of data for each decision tree, introducing diversity in the training process and increasing the model's robustness.

- Decision Making and Voting: When making predictions, each decision tree in the Random Forest casts its vote. For classification tasks, the final prediction is determined by the mode (most frequent prediction) across all the trees. In regression tasks, the average of the individual tree predictions is taken. This internal voting mechanism ensures a balanced and collective decision-making process.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. Random Forest is renowned for its effectiveness in both classification and regression tasks. Here are some important features of Random Forest:

- Diversity: Each individual tree in the Random Forest is constructed using a random subset of features, leading to diversity among the trees.
- Immune to the curse of dimensionality: Random Forest is robust to high-dimensional data because each tree only considers a random subset of features, reducing the dimensionality of the feature space.
- Parallelization: The construction of each tree in the Random Forest can be done independently and in parallel. This allows for efficient utilization of computational resources, as multiple trees can be built simultaneously.
- Train-Test split: In Random Forest, there is no need to explicitly segregate the data into training and testing sets as the out-of-bag (OOB) samples (approximately 30% of the data) can be used for validation during training.
- Stability: The final prediction in Random Forest is based on the majority voting (for classification) or averaging (for regression) of predictions from individual

trees. This ensemble approach enhances the stability of the model and reduces the impact of noise in the data.

3.4. K-Nearest Neighbors Classifier (KNN)

The K-Nearest Neighbors (KNN) algorithm is a fundamental classification algorithm in machine learning. It falls under the supervised learning domain and finds extensive application in pattern recognition, data mining, and intrusion detection.

One of the key advantages of KNN is its non-parametric nature. Unlike some other algorithms that make assumptions about the underlying distribution of data (e.g., Gaussian Mixture Models), KNN does not make any such assumptions. This makes it widely applicable in real-life scenarios where the data distribution may not be known beforehand.

In KNN, we are provided with prior data, often referred to as training data. This training data consists of coordinates associated with specific attributes or labels. The algorithm works by classifying new data points based on their proximity to the nearest neighbors in the training dataset. The "K" in KNN represents the number of nearest neighbors considered when making a classification decision.

Overall, KNN is valued for its simplicity, flexibility, and ability to handle non-linear decision boundaries. However, it may not perform optimally in high-dimensional spaces or with large datasets due to its computational complexity.

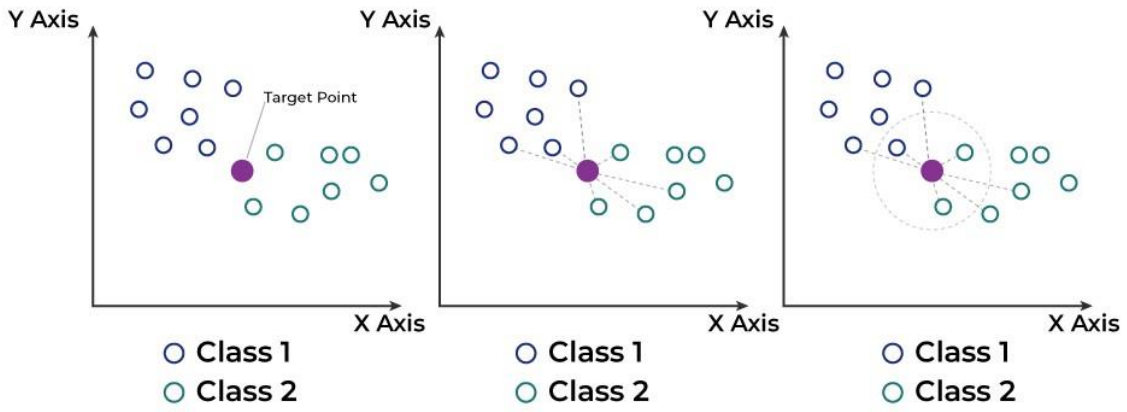


Figure 5. Example of a K-NN algorithms

The working of the K-Nearest Neighbors (KNN) algorithm can be outlined as follows:

Step 1: Selecting the optimal value of K

- K represents the number of nearest neighbors to be considered when making a prediction.
- The choice of K can significantly impact the performance of the algorithm and is often determined through techniques like cross-validation.

Step 2: Calculating distance

- To measure the similarity between the target data point and the training data points, distances are calculated.
- The most commonly used distance metric is Euclidean distance, although other metrics like Manhattan distance or Minkowski distance can also be employed.
- For each data point in the training dataset, the distance to the target data point is computed using the chosen distance metric.
- Let X represent the training dataset with n data points, where each data point is represented by a d-dimensional feature vector X_i and Y represents the corresponding labels or values for each data point in X. Given a new data point x, the algorithm calculates the distance between x and each data point X_i in X using a distance metric, such as Euclidean distance:

$$Distance(x, X_i) = \sqrt{\sum_{j=1}^{d} \left(x_i - X_{i_j}\right)^2}$$

Example: In 2 dimensions, the distance between $A(x_1, y_1), B(x_2, y_2)$ is

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

Step 3: Finding the K nearest neighbors

- Once distances are calculated, the K nearest neighbors to the target data point are identified.
- This is typically achieved by sorting the distances in ascending order and selecting the K data points with the smallest distances.

Step 4: Making predictions

- For classification tasks, the predicted class for the target data point is determined by majority voting among its K-Nearest Neighbors.
- For regression tasks, the predicted value is often computed as the mean or median of the target values of the K nearest neighbors.

Step 5: Output

- Finally, the predicted class or value is returned as the output of the algorithm.

KNN is a basic and understandable algorithm. It is unnecessary to make assumptions about data distribution or to learn a sophisticated model. KNN is capable of handling unstructured data and makes no assumptions about data structure. This allows it to be used with a wide range of data formats, including text, graphics, and audio. Also works effectively on tiny datasets with a limited number of samples. It is not necessary to do extensive calculations prior to making forecasts. Easily fine-tune parameters with K being an important parameter in the KNN algorithm. The algorithm's performance can be affected by changes to the K value.

Although the KNN algorithm offers several benefits, it also has the following drawbacks. KNN performs poorly when the data set is huge because computing the distance between data points takes a lengthy time. KNN is not an appropriate method for handling huge datasets. KNN is subject to noise in the data and data points in various classes that are near together. This can result in erroneous or unstable classifications. KNN does not manage many variables and requires data balance in different levels. If one class has more samples than another, KNN can quickly become biased and generate incorrect results.

3.5. Metrics

3.5.1. Precision, Recall and F1 score

In the evaluation of classification models, there are four fundamental metrics that serve as the basis for most other metrics. In this study, the primary evaluation metrics utilized to assess the performance of the ML classifier include accuracy, recall, precision, and F-measure.

Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome. Accuracy was calculated by dividing the number of correct predictions by the total number of predictions. The mathematical formula is:

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

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Precision is a metric that measures how often a machine learning model correctly predicts the positive class. It was calculated by dividing the number of correct positive predictions (true positives) by the total number of instances the model predicted as positive (both true and false positives).

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

Recall is calculated as the ratio between the number of *Positive* samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect Positive samples. The higher the recall, the more positive samples detected.

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

The balance between precision and recall can cause the model's outputs to have either high precision and low recall or low precision and high recall. It becomes challenging to determine which is a good model since it can be unclear which evaluation – precision or recall – is more appropriate. To overcome this challenge, both precision and recall are combined to form a new metric called the F1 score, which is the harmonic mean of both.

F1 score or *F-measure* is described as the harmonic mean of the precision and recall of a classification model. The two metrics contribute equally to the score, ensuring that the F1 metric correctly indicates the reliability of a model.

$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1 score, ranging from 0 to 1, represents the model's performance, with 0 indicating the lowest possible result and 1 representing a perfect outcome, indicating that every label was correctly predicted by the model. A high F1 score typically reflects a well-balanced performance, indicating the model's ability to achieve both high precision and high recall simultaneously. Conversely, a low F1 score suggests a trade-off between recall and precision, indicating challenges in achieving a balanced performance. As a general rule of thumb, the F1 score value can be interpreted as follows:

F1 score	Interpretation
> 0.9	Very good
0.8 - 0.9	Good
0.5 - 0.8	OK
< 0.5	Not good

3.5.2. ROC and AUC

The AUC-ROC curve, which stands for the Area Under the Receiver Operating Characteristic curve, provides a visual depiction of a binary classification model's performance across different classification thresholds. It serves as a widely used tool in machine learning for evaluating a model's capability to differentiate between two classes, typically representing the positive class (e.g., presence of a disease) and the negative class (e.g., absence of a disease).

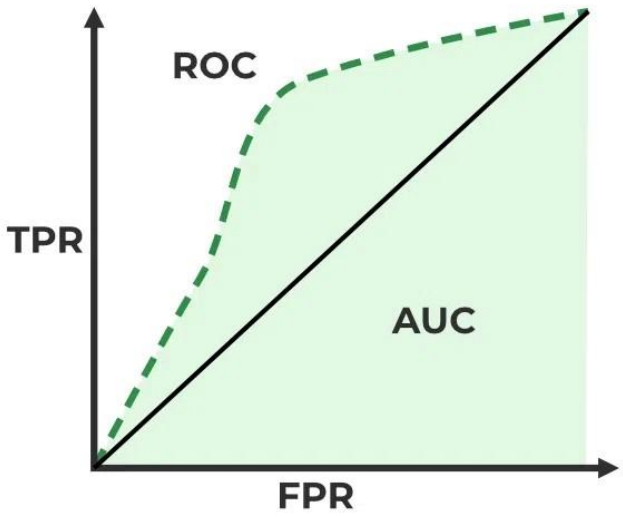


Figure 6. ROC-AUC Classification Evaluation Metric

ROC stands for Receiver Operating Characteristics, and the ROC curve is the graphical representation of the effectiveness of the binary classification model. It

plots the true positive rate (TPR) vs the false positive rate (FPR) at different classification thresholds.

True Rositive Rate / Sensitivity / Recall is the ratio of positive examples that are correctly identified

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

The False Positive Rate (FPR) is the ratio of negative examples that are incorrectly classified as positive by a binary classification model. It is calculated as the number of false positives divided by the sum of true negatives and false positives.

$$FPR = 1 - TPR = 1 - \frac{TP}{TP + FN} = \frac{FN}{TP + FN}$$

AUC stands for Area Under the Curve, and it refers to the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve illustrates the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) at various classification thresholds. AUC is a metric used to evaluate the overall performance of a binary classification model. Since both TPR and FPR range from 0 to 1, the AUC value also falls within this range. A higher AUC value indicates better model performance, with a value of 1 indicating perfect classification.

CHAPTER 4: CONSTRUCTION FOR MODEL

4.1. Exploratory Data Analysis – EDA

In earlier parts, we learnt about stroke, the need of developing a stroke risk prediction model, and the methods used to apply that model. This chapter focuses on the more difficult aspects of model creation. First, a detailed description of the data set will be supplied, as well as relevant information gleaned from the data. Then, do exploratory data analysis and preparation. The variable selection procedure will then be carried out, resulting in the building of four distinct models: Logistic Regression, Decision Tree, Random Forest, and K-Nearest Neighbors (KNN). Additionally, optimization approaches will be employed to improve the performance of these models. Finally, the models' outputs will be compared to form judgments regarding their usefulness in predicting stroke risk.

4.1.1. Data Overview

The dataset used in this study was sourced from Kaggle, a publicly available platform, focusing on stroke prediction. It contains data relevant to strokes, considering factors such as gender, age, various medical conditions, and smoking status to determine a patient's risk of stroke. Each entry in the dataset provides crucial information about the patient's health status. *Table 1* presents the features of the dataset along with their descriptions.

Columns Name	Meaning
id	Unique identifier
gender	Male, "Female" or "Other"
age	Age of the patient
hypertension	0 if the patient doesn't have hypertension
heart_disease	0 if the patient doesn't have any heart diseases
ever_married	No or "Yes"
work_type	Children, "Govt_jov", "Never_worked", "Private" or "Self-employed"
Residence_type	Rural or "Urban"
avg_glucose_level	Average glucose level in blood
bmi	Body mass index
smoking_status	Formerly smoked, "never smoked", "smokes" or "Unknown"
stroke	1 if the patient had a stroke or 0 if not

Table 1. A summary of the dataset's characteristics.

The dataset includes information from patients at risk of stroke. This dataset comprises 5110 patients and 12 attributes. This dataset divides the stroke variable into two categories: "stroke" and "no stroke." his category is a binary-valued property that shows whether or not the patient exhibits stroke symptoms. Out of 5110 individuals, 249 suffered a stroke, whereas 4869 were non-stroke. The "BMI" property contains 201 null values. Because the median is unaffected by outliers, it was utilized to impute the missing values for continuous variables.

Below is a representation of the appearance of the sample data:

Table 2. Shows sample data

id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	1

4.1.2. Exploratory data analysis

Before being subjected to analysis and evaluation, each dataset underwent the process of Exploratory Data Analysis (EDA) and visualization, ensuring that the final dataset aligned with the objectives of the report and contributed to accurate predictive results.

Stroke – Target feature

In the stroke's column, we found that 249 strokes occurred out of a total of 5110 cases. We represented'stroke' as 1 and 'non-stroke' as 0, with 4861 patients being normal in total. As indicated in *Figure 7*, the proportion of patients who did not have a stroke was 95.13%, whereas the rate of patients who did have one was 4.87%.

Figure 7. Stroke Status

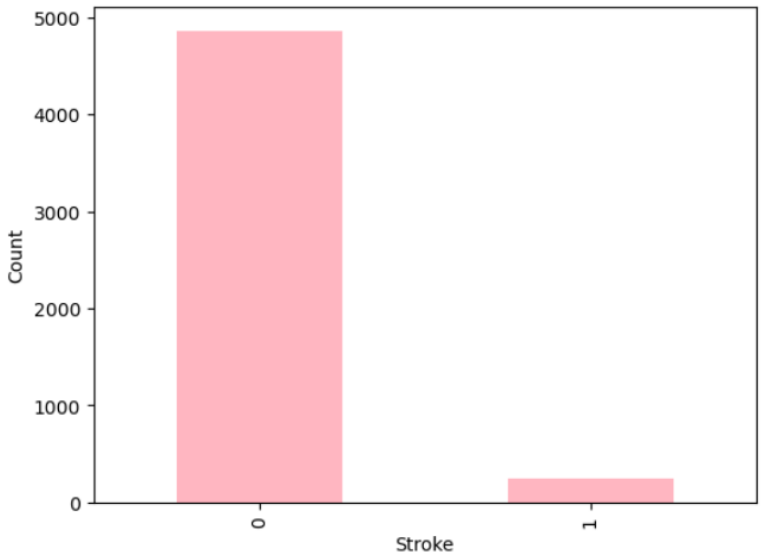


Figure 8 and *Figure 9* presents a boxplot for parameters such as age, average_glucose_level. With *Figure 8* the age of patients with strokes varied from approximately 40 to 80 years old, while the age of patients without strokes ranged from 0 to 80.

In *Figure 9.* the typical blood glucose levels for the majority of patients ranged from 50 to 130. The remaining patients had glucose levels ranging from 130 to 300, and

none of these patients experienced a stroke. For stroke patients, most had average blood glucose levels between 60 and 130, with only a small fraction having levels between 170 and 300.

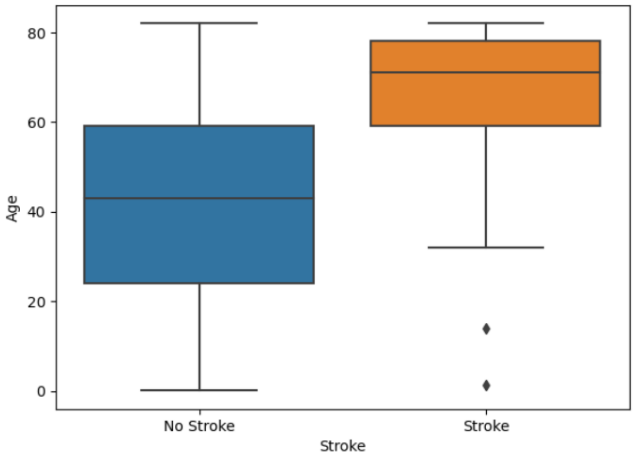


Figure 8. Boxplot of age by Stroke Status

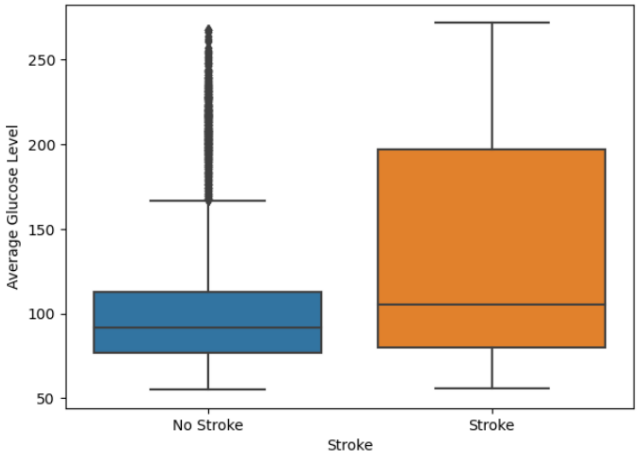
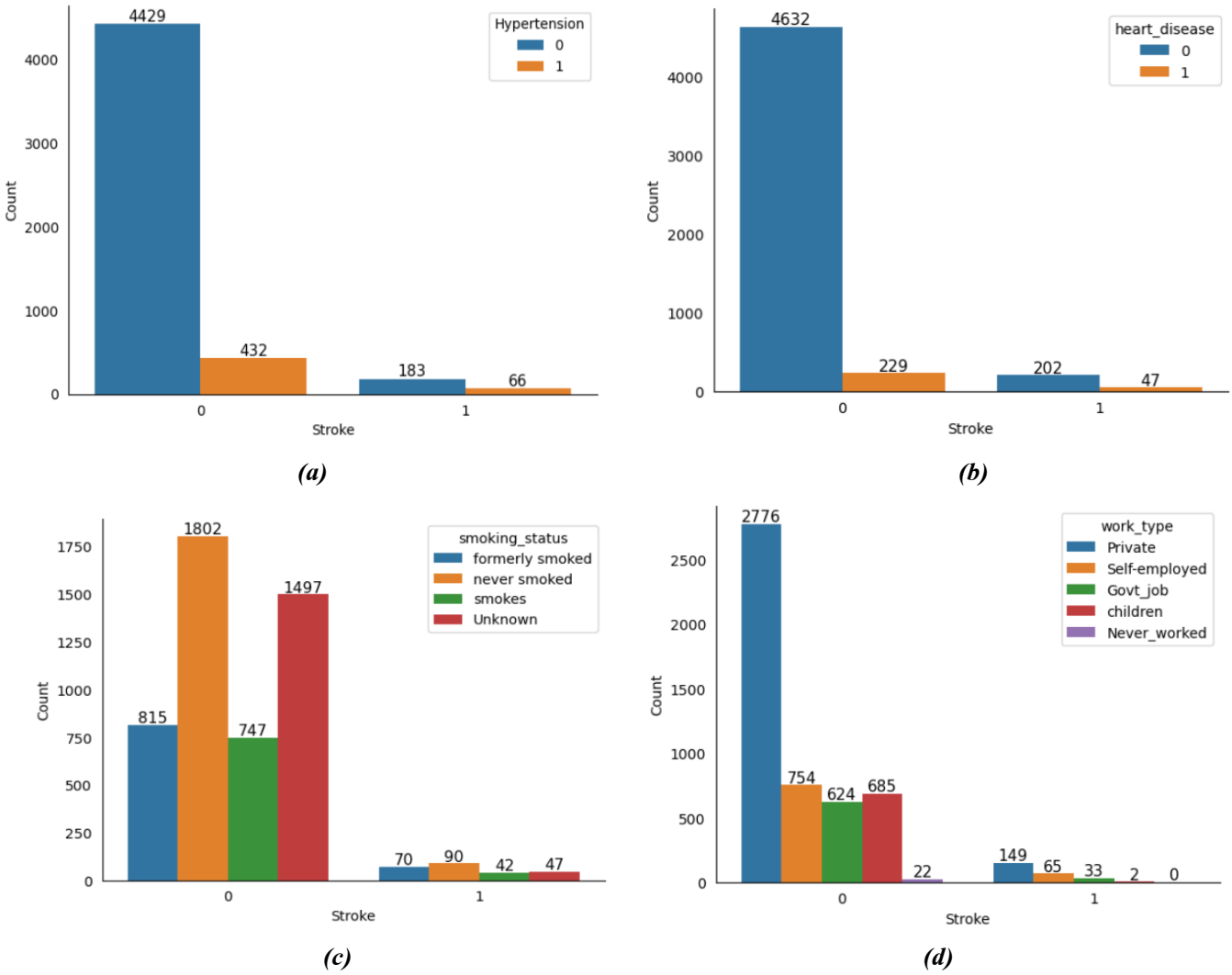


Figure 9. Boxplot of average_glucose_level by Stroke Status

Figure 10 illustrates the variations of attributes concerning the target variable stroke. In *Figure 10.(a)*, it is shown that 183 individuals experienced a stroke despite not having hypertension, whereas 4429 patients had neither hypertension nor a stroke. Additionally, 66 patients had both hypertension and stroke, while 432 patients had hypertension but did not suffer a stroke. In addition, *Figure 10.(b)* shows that a total of 202 patients had a stroke but no heart disease, while 229 patients had heart disease but no stroke. Forty-seven patients had both heart disease and a stroke, whereas 4632 patients had neither condition. This plotting shows that the number of "people with strokes but no heart disease" is approximately 6 to 8 times the number of "people with strokes and also heart disease". This shows most of the people with no heart disease are suffering with strokes compared to the once who have heart disease.

Figure 10. Bar plots depict certain categorical attributes as follows (a)hypertension and (b)heart_disease, (c)smoking_status and (d)work_type



In *Figure 10.(c)*, 1497 patients with an unknown smoking status did not experience a stroke, while 47 patients with the same status did experience a stroke. Among those who formerly smoked, 70 individuals suffered strokes, whereas 815 did not. For patients who had never smoked, 90 experienced strokes, while 1802 did not. Additionally, 42 current smokers experienced strokes, compared to 747 current smokers who did not. Through this, we can see people who never smoked got more strokes The people who smoked and unknown has a somewhat same probability of getting stroke.

Finally, *Figure 10.(d)* shows 2776 patients who work privately without having a stroke, while 149 persons who work privately and have a stroke. While 65 self-employed persons suffered a stroke, 754 people working in the same job did not. A stroke affects 33 out of every 657 government employees. In terms of children, nearly none of them have had a stroke; just two patients are children. There are no strokes among individuals who do not work. It is clear that persons who work privately are at a much higher risk of stroke.

4.2. Data Preprocessing

This process involves cleaning the data to ensure its quality and reliability. Real-world data often contains redundant values and substantial noise, including duplicates, redundancies, and missing values. When such data is used to build a model, it can lead to inaccurate results. Therefore, it's essential to clean the data before feeding it into the model to ensure more accurate outcomes. This cleaning process typically involves removing duplicate and redundant values from the dataset and correcting any missing values.

4.2.1. Fill missing values

When dealing with missing data, there are two options: remove the column/row that contains the missing values or replace the missing values with another value (popular ways include filling in values like 0, median, average, or more advanced emptying procedures). The choice of replacement value will depend on the type of missing value.

In *table 3*, of the 201 NULL values in the bmi property, 40 Nan values out of 249 BMI values had a stroke. As a result, we are unable to discard the NULL value. We cannot calculate the mean because the outliers have an effect on the mean. Hence we impute it with median values.

Table 3. Null values of each feature

Feature	Total of null values
bmi	201
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
smoking_status	0
stroke	0

4.2.2. Categorical features

Most machine learning methods are unable to analyze categorical data directly (with the exception of Decision Trees). As a result, categorical characteristics must be converted into numerical values before the models can be trained. Categorical data is generally classified into two types: ordinal and nominal variables. As a result, there are many approaches to dealing with categorical data: Ordinal Encoding, Single Hot Encoding, and Dummy Encoding.

In this case, we chose to use dummy encoding for the remaining category variables. Because these variables have several categories in 1 variable, the use of dummy encoding produces numerous categorical variables, each indicating a distinct value of the original category variable. Therefore, preventing misunderstandings when the model understands that there is a hierarchical relationship between the values of a category variable.

4.2.3. Data Scaling

Differences in units, distributions, and scales between features have a significant impact on model classification efficiency and convergence of algorithms. Furthermore,

differences in values of features (variables) cause distortions in assessing the importance of variables due to differences in their magnitude. Scaling helps to avoid these issues by ensuring that the feature values are within a manageable range.

The two feature scaling methods are normalization and standardization.

1. Standardization (Z-score normalization): This technique rescales the data so that it has a mean of 0 and a standard deviation of 1. It's particularly useful when the data follows a Gaussian distribution.
2. Min-Max Scaling (Normalization): This technique rescales the data to a fixed range, typically 0 to 1. It's useful when the data does not necessarily follow a Gaussian distribution.

The data in the article are scaled according to the first method with a mean of 0 and a variance of 1. This standardization supports algorithms such as linear regression and logistic regression, as they are often assume that the input data follows a Gaussian distribution.

4.2.4. Balance Data using SMOTE

Balancing the dataset facilitates model training by preventing it from becoming biased toward one class. Data balancing can be achieved through undersampling or oversampling. Although undersampling is easy to perform and can improve model run-time, it has certain drawbacks. Removing samples from the original dataset can lead to the loss of important information and may cause overfitting if there are insufficient data. Therefore, oversampling was proposed in this study. To minimize the risk of inappropriate model training, we used the Borderline-Synthetic Minority Oversampling Technique (SMOTE) to balance the training dataset. This method generates synthetic points from the minority class. A comparison of the borderline-

SMOTE algorithm's performance before and after implementation is shown in Figure 10. The testing dataset was left unbalanced to preserve its integrity.

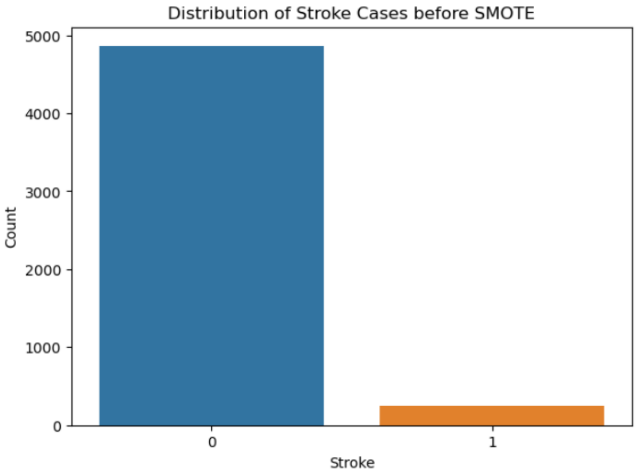


Figure 11.Unbalanced training dataset

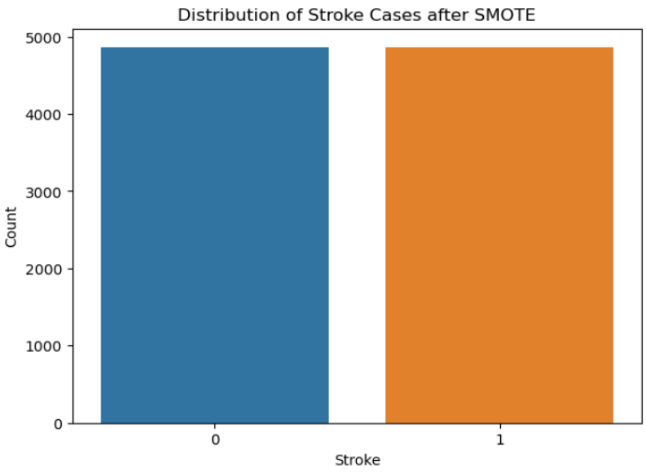


Figure 12. Balanced training dataset using Borderline-SMOTE

4.2.5. Features Selection

Pearson's Correlation and Mutual Information were utilized in this study to select the most relevant attributes. These techniques helped in extracting essential features and reducing the amount of data.

Pearson's Correlation

Confirm and visualize the correlation between features. Statistical analysis appears trivial, yet it's a crucial activity. A heat map provides you with the correlation between each feature on the other values, which is equivalent to a multicollinearity correlation matrix. This graph provides a visual depiction of the correlation matrix that is displayed in *Figure 13*.

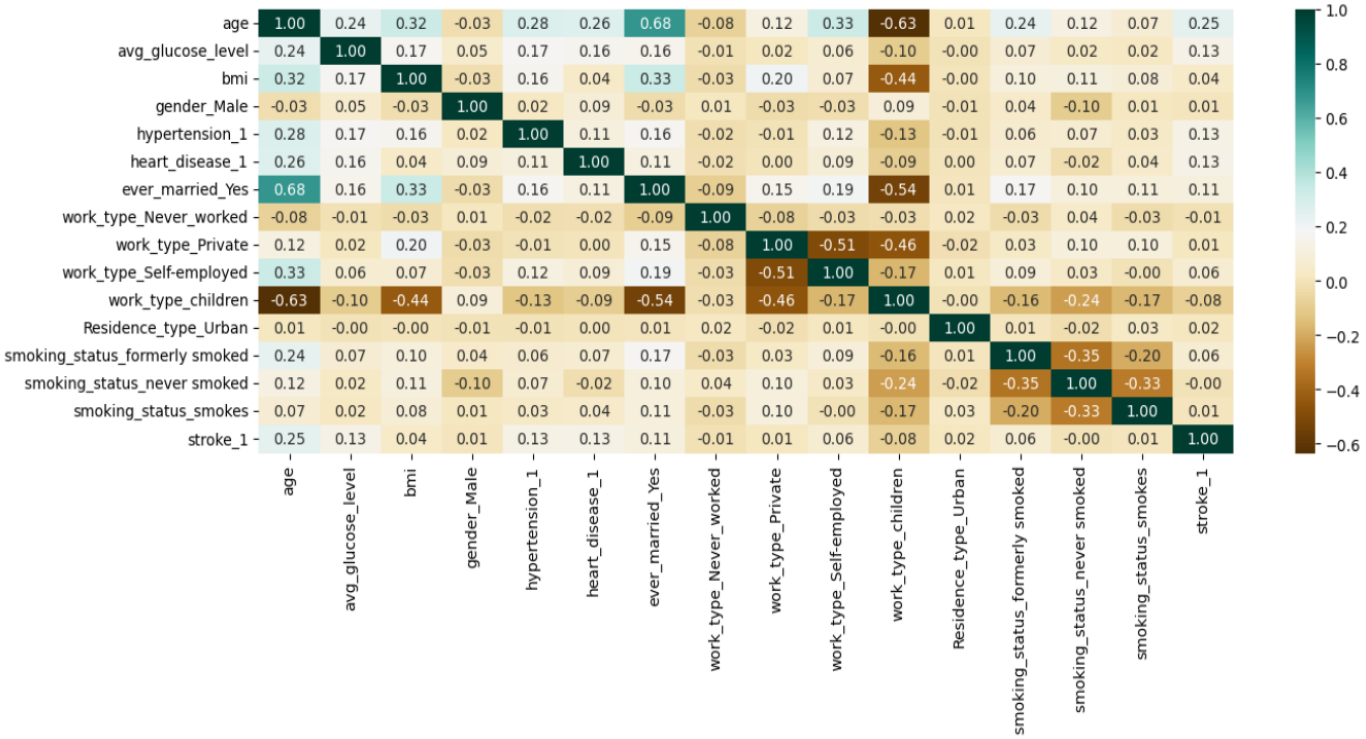


Figure 13. Correlation matrix

Mutual Information

MI can be used to measure the importance of features to the target variable during the feature selection process. The attributes were ranked based on how much each attribute contributes to the target variable, as seen in *Figure 14*.

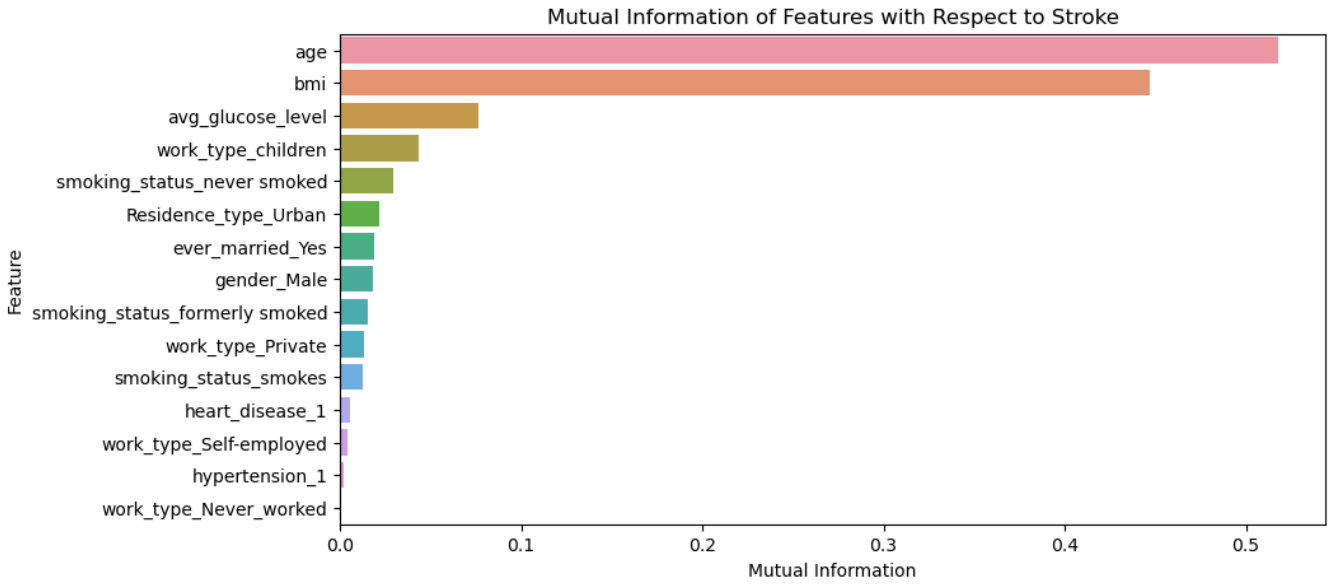


Figure 14. MI of features with respect to Stroke

Important features

The important features identified through Mutual Information and Pearson's correlation analysis, as shown in the figures above, include age, BMI, average glucose level, and smoking status. These consistent characteristics have been chosen for further analysis.

CHAPTER 5 : RESULT

5.1. Model Evaluation

ML algorithms can accurately estimate an individual's risk of stroke and provide personalized recommendations for prevention and therapy by identifying trends and risk factors. Furthermore, they can assist medical professionals in diagnosing patients more swiftly and accurately, thereby enhancing patient outcomes. A training-to-testing ratio of 80:20 was employed to train all the models. The comparison results are displayed in *Table 4* to provide a clearer understanding of the performance.

Table 4. Performance Metrics using 80-20 Ration

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	79,38%	0,785	0,808	0,796
Decision Tree	90,4%	0,887	0,826	0,906
Random Forest	94,7%	0,93	0,966	0,948
KNN	92,40%	0,855	0,984	0,915

Table 5 represents the results compare all models with 5-fold cross validation.

Table 5.Performance Metrics using 5-fold cross validation

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	86,32%	0.856	0,872	0,864
Decision Tree	92,2%	0,909	0,938	0,923
Random Forest	94,9%	0,94	0,959	0,949
KNN	90,85%	0,855	0,984	0,915

The table allows for the comparison of different machine learning models like Logistic Regression, KNN, Decision Tree, Random Forest based on their accuracy. This can help choose the most suitable model for a specific problem. The accuracy of all the models listed is above 80%, suggesting they are all performing well on the task. Random Forest has the best-performed algorithm in terms of both accuracy and F1-score, reaching 94.9% and 0.949 respectively using 5-fold cross validation. This

suggests that it is the most effective at correctly classifying data points and has the best ability to distinguish between classes.

5.2. Hyperparameter tuning

To avoid overfitting, hyperparameter tuning was conducted on all the algorithms using 5-fold cross-validation and the Grid Search technique. *Table 6* outlines the hyperparameters selected by the models.

Table 6.Hyperparameters used for Grid Search

Algorithm	Hyperparameters
Logistic Regression	{'C': 0.2, 'penalty': 'l1', 'solver': 'liblinear'}
Decision Tree	{'criterion': 'entropy', 'max_depth': 20, 'min_samples_leaf': 1}
Random Forest	{'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
KNN	{'metric': 'minkowski', 'n_neighbors': 3, 'weights': 'distance'}

The results in Table 04, 05 and 06 showed that method Random Forest to improvements in the model's scores, increasing the accuracy score from 0.947 (Table 04) to 0.949.

5.3. ROC_AUC

The ROCs of these effective techniques such as LG, RF, DT, and KNN are represented in *Figure 15*, *Figure 16*, *Figure 17*, and *Figure 18,* respectively.

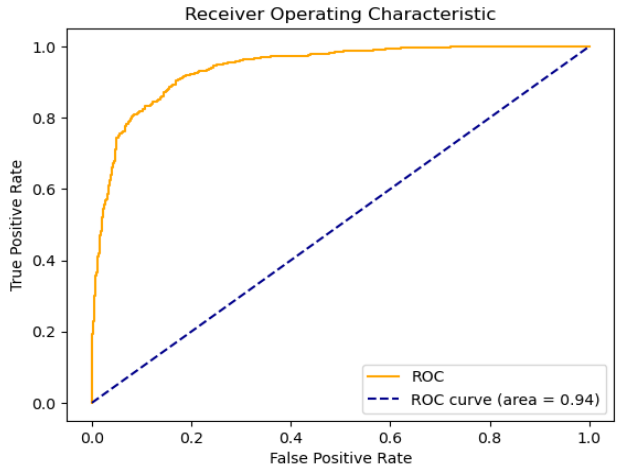


Figure 15. Logistic Regression algorithm ROC

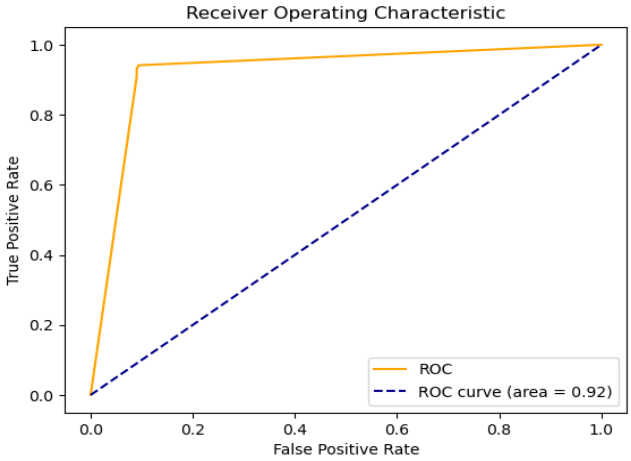


Figure 16. Decision Tree algorithm ROC

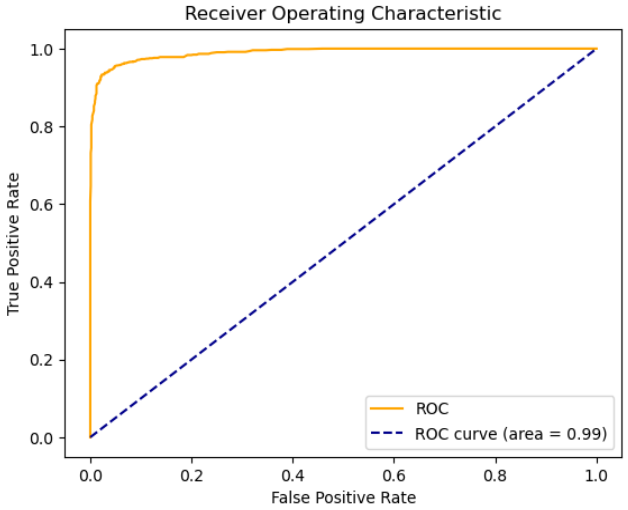


Figure 17. Random Forest Classifiers ROC

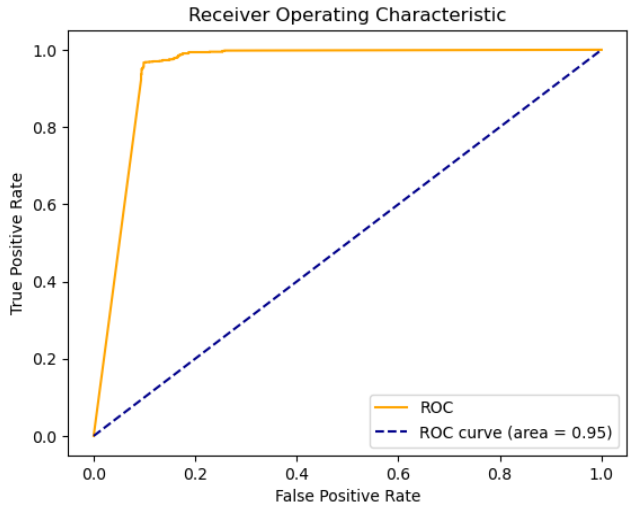


Figure 18. K-NN algorithm ROC

We may evaluate the models' effectiveness in predicting stroke using the AUC (Area Under the Curve) values and the ROC chart above. The Random Forest model performed the best, with an AUC of 0.99, indicating nearly flawless classification between positive and negative classes. Random Forest's ROC curve nearly reaches the

chart ceiling, indicating a very low false positive rate and a very highrate. Therefore the classification model is quite accurate. The K-Nearest Neighbors (KNN) and Logistic Regression models also performed well, with AUCs of 0.95 and 0.94, respectively. The ROC curves for these two models are both around the upper left corner of the graph, indicating good prediction accuracy. Although the Decision Tree model has a little lower AUC of 0.92 than other models, it still performs well in classification. To summarize, the Random Forest model is the best solution for this stroke prediction problem because to its superior performance, although KNN and Logistic Regression are other viable options, depending on the individual needs and processing resources.

CHAPTER 6: CONCLUSION

As life progresses, humans face numerous health concerns, with strokes being a significant issue. Early intervention for strokes can greatly enhance the chances of a full recovery and reduce the risk of complications. Therefore, we utilize machine learning to predict strokes, aiming to facilitate timely intervention and improve patient outcomes.

In this research paper, we present a comparative study of various machine learning techniques for predicting stroke in the patients. We explore Logistic Regression, Decision Trees, Random Forest, and KNN. Our experimental results demonstrate that ensemble learning technique, namely Random Forest, achieve the highest accuracy compared to the other models. They obtain an impressive AUC score of 99% for predicting stroke, outperforming other algorithms in terms of accuracy, precision, F-measure, recall, and AUC score. This underscores the effectiveness of the Random Forest as a promising approach for predicting stroke. This demonstrates the utility of Random Forests as a viable tool for stroke prediction. The value of early identification cannot be overstated, since it can reduce the risk of serious complications related to stroke.

Lifestyle changes can't avert all strokes. When it comes to reducing your risk of stroke, though, many of these modifications can make a significant difference. Quitting smoking today will minimize your stroke risk. Alcohol use should be limited. Alcohol can elevate your blood pressure, putting you at greater risk of stroke.

Maintain a healthy weight. They are at an increased risk of stroke because they are overweight or obese. To keep your weight in check, consume a wellbalanced diet and engage in regular physical activity. Both methods can also help lower cholesterol and

blood pressure. In addition, ensure that you have regular checks. Doctors can help you make these alterations to your way of life by offering encouragement and support.

REFERENCES

[1] James McIntosh, (2024). *Everything you need to know about stroke.* URL: https://www.medicalnewstoday.com/articles/7624

[2] The Johns Hopkins University, The Johns Hopkins Hospital, and Johns Hopkins Health System, (2024)."Johns Hopkins Medicine." URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/stroke

[3] Dr Poonam Khetrapal Singh, (2021). "World Stroke Day,". URL: https://www.who.int/southeastasia/news/detail/28-10-2021-world-stroke-day

[4] (WHO)Health OrganizationWorld, (2022).“World Stroke Day 2022”. URL: https://www.who.int/srilanka/news/detail/29-10-2022-world-stroke-day-2022

[5] L. Amini, R. Azarpazhouh, M. T. Farzadfar, S. A. Mousavi, F. Jazaieri, F. Khorvash, R. Norouzi, & N. Toghianfar, (2013). “Prediction and control of stroke by data mining,” (*International Journal of Preventive Medicine)* vol. 4, no. 2, pp. S245–S249. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3678226/

[6] C. A. CHENG, Y. C. LIN, and H. W. CHIU,(2014). “Prediction of the prognosis of ischemic stroke patients after intravenous thrombolysis using artificial neural networks” vol. 202, pp. 115–118. URL:https://books.google.com/books?hl=vi&lr=&id=ApzYBAAAQBAJ&oi=fnd&pg=PA115&ots=ci7yrIVk3G&sig=ASJNa3vTf8stUEGxCje8VqGvLiI

[7] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, & R. Manikandan, (2020). "Classification of stroke disease using machine learning algorithms", vol. 32, no. 3, pp. 817–828.
URL: https://link.springer.com/article/10.1007/s00521-019-04041-y

[8] T. Kansadub, S. Thammaboosadee, S. Kiattisin, and C. Jalayondeja, (2015). "Stroke risk prediction model based on demographic data," in *Proceedings of the 2015 8th Biomedical Engineering International Conference (BMEiCON)*, pp. 1–3, Pattaya, Thailand. URL: https://ieeexplore.ieee.org/abstract/document/7399556

[9] S. Y. Adam, A. Yousif, and M. B. Bashir, (2016). "Classification of ischemic stroke using machine learning algorithms," *International Journal of Computer Application*, vol. 149, no. 10, pp. 26–31.
URL: https://www.researchgate.net/Classification-of-Ischemic-Stroke-using-Machine-Learning

[10] Anshul Saini, (2024). "A Beginner's Guide to Logistic Regression". URL: What is Logistic Regression: A Complete Guide (analyticsvidhya.com)

[11] Sonoo Jaiswal , (2021). "Decision Tree Classification Algorithm".
URL: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[12] "Cây Quyết Định (Decision Tree)", *Trí tuệ nhân tạo*, (2019). URL: https://trituenhantao.io/kien-thuc/decision-tree/

[13] "Decision Trees (1): Iterative Dichotomiser 3", Machine Learning cơ bản (2018).
URL: https://machinelearningcoban.com/2018/01/14/id3/

[14] "Random Forest Algorithm in Machine Learning" (2024). URL: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

[15] Sruthi E R, (2024). “Understand Random Forest Algorithms With Examples (Updated 2024)”.
URL: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[16] “K-Nearest Neighbor(KNN) Algorithm”, (2024). URL: https://www.geeksforgeeks.org/k-nearest-neighbours/

[17] “F1 Score in Machine Learning”, (2023). URL: https://encord.com/blog/f1-score-in-machine-learning/

[18] Evidently AI Team , “Accuracy vs. precision vs. recall in machine learning: what's the difference?”. URL: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall

[19] “AUC ROC Curve in Machine Learning”, (2024). URL: https://www.geeksforgeeks.org/auc-roc-curve/

[20] Susmita S, K. Chadaga, N. Sampathila, S. Prabhu, R. Chadaga, & Swathi Katta S, (2023). "Multiple Explainable Approaches to Predict the Risk of Stroke Using Artificial Intelligence". URL: https://doi.org/10.3390/info14080435

Document Viewer

Turnitin Originality Report

Processed on: 21-Jun-2024 14:46 +07
ID: 2406182289
Word Count: 8254
Submitted: 1

11208164_Tran Thi Thu Trang_Predicting Stroke...
By Trần Thị Thu Trang

Similarity Index	Similarity by Source	
18%	Internet Sources:	20%
	Publications:	16%
	Student Papers:	11%

include quoted | include bibliography | excluding matches < 50 words | mode: quickview (classic) report | print | download

4% match (Internet from 15-Jan-2024)
https://WWW.MDPI.COM/2078-2489/14/8/435

2% match (Chetan Sharma, Shamneesh Sharma, Mukesh Kumar, Ankur Sodhi. "Early Stroke Prediction Using Machine Learning", 2022 International Conference on Decision Aid Sciences and Applications (DASA), 2022)
Chetan Sharma, Shamneesh Sharma, Mukesh Kumar, Ankur Sodhi. "Early Stroke Prediction Using Machine Learning", 2022 International Conference on Decision Aid Sciences and Applications (DASA), 2022

1% match (Internet from 20-Apr-2024)
https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/?ref=ml_lbp

1% match (Internet from 07-Jun-2024)
https://www.geeksforgeeks.org/k-nearest-neighbours/?id=141822&type=article

1% match (Internet from 28-Nov-2022)
https://www.researchgate.net/publication/364359247_Stroke_Prediction_with_Logistic_Regression_and_assessing_it_using_Confusion_M

1% match (Internet from 21-Oct-2022)
https://www.researchgate.net/publication/359518651_Early_Stroke_Prediction_Using_Machine_Learning

1% match (Internet from 06-Feb-2023)
http://www.uet.vnu.edu.vn

1% match (Susmita S, Krishnaraj Chadaga, Niranjana Sampathila, Srikanth Prabhu, Rajagopala Chadaga, Swathi Katta S. "Multiple Explainable Approaches to Predict the Risk of Stroke Using Artificial Intelligence", Information, 2023)
Susmita S, Krishnaraj Chadaga, Niranjana Sampathila, Srikanth Prabhu, Rajagopala Chadaga, Swathi Katta S. "Multiple Explainable Approaches to Predict the Risk of Stroke Using Artificial Intelligence", Information, 2023

1% match (Internet from 06-Jun-2024)
https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

1% match (student papers from 01-Jun-2021)
Submitted to Birkbeck College on 2021-06-01

1% match (Internet from 26-Jul-2023)
https://www.hindawi.com/journals/jhe/2021/7633381/

1% match (student papers from 04-Sep-2023)
Submitted to Queen's College on 2023-09-04

1% match ("Sustainable Development through Machine Learning, AI and IoT", Springer Science and Business Media LLC, 2023)
"Sustainable Development through Machine Learning, AI and IoT", Springer Science and Business Media LLC, 2023

1% match (Internet from 02-Dec-2023)
https://vdoc.pub/documents/master-machine-learning-algorithms-58d6n05h7ea0