*I. J. Education and Management Engineering,* **2023, 2, 27-35**
Published Online on April 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijeme.2023.02.04

# Analyzing the Performance of the Machine Learning Algorithms for Stroke Detection

**Trailokya Raj Ojha***
Assistant Professor, Department of Computer Science and Engineering, Nepal Engineering College, 44800 Bhaktapur, Nepal
Email: trailokyaro@nec.edu.np
ORCID iD: https://orcid.org/0000-0001-7554-1731
*Corresponding Author

**Ashish Kumar Jha**
Assistant Professor, Department of Computer Science and Engineering, Nepal Engineering College, 44800 Bhaktapur, Nepal
Email: ashishkj@nec.edu.np
ORCID iD: https://orcid.org/0000-0003-4530-1942

Received: 20 September, 2022; Revised: 26 October, 2022; Accepted: 25 November, 2022; Published: 08 April, 2023

**Abstract:** A brain stroke is a condition with an insufficient blood supply to the brain, which causes cell death. Due to the lack of blood supply, the brain cells die, and disabilities occurs in different parts of the brain. Strokes have become one of the major causes of death and disability in recent years. Investigating the affected individuals has shown several risk factors that are considered to be causes of stroke. Considering such risk factors, many research works have been performed to classify and predict stroke. In this research, we have applied five machine learning algorithms to identify and classify the stroke from the individual's medical history and physical activities. Different physiological factors have are considered and applied to machine learning algorithms such as Naïve Bayes, AdaBoost, Decision Table, k-NN, and Random Forest. The algorithm Decision Table performed the best to predict the stroke based on different physiological factors in the applied dataset with an accuracy of 82.1%. The machine learning algorithms can be a helpful for clinical prediction of stroke against individual's medical history and physical activities in a better way.

**Index Terms:** Brain stroke, machine learning, data analysis, prediction

## 1. Introduction

A stroke, also known as a brain attack, happens when a blood vessel in the brain breaks or when something stops the flow of blood to a specific area of the brain. The common types of stroke are ischemic and hemorrhagic strokes. With permanent or transient damage, it can range from minor to extremely severe. Hemorrhages are rare and entail the rupture of a blood artery, which can result in brain bleeding. The most frequent type of stroke, ischemic strokes, is characterized by the suspension of blood flow to a portion of the brain as a result of arterial narrowing or blockage.

According to World Stroke Organization [1], globally 25% of adults over the age of 25 will have a stroke in their lifetime. More than twelve million people experience their first stroke yearly, and more than six million of them die due to stroke. Stroke has an impact on the patient as well as on their social status, family, and workplace. Many medical studies have been carried out to find effective stroke predictors. Some of the studies reported the risk factors for stroke as age, systolic blood pressure, the use of anti-hypertensive therapy, diabetes mellitus, cigarette smoking, prior cardiovascular disease, atrial fibrillation, and left ventricular hypertrophy by electrocardiogram [2,3].

Stroke causes a significant number of fatalities and is on the rise in developing nations [4]. Several stroke risk factors are responsible for different types of strokes. Different types of prediction algorithms made it easy to know the relationship between risk factors and type of stroke. The algorithms used in machine learning are beneficial in providing accurate analysis and producing accurate predictions. In this research, we have used different machine learning algorithms to predict the possibility of the occurrence of a stroke based on the patients' health reports and statistical data. The different machine learning algorithms used to conduct the research are Naïve Bayes, AdaBoost, Decision Table, k-NN, and Random Forest.

The dataset collected from Kaggle [5] with various attributes is used in this study. The dataset collected from a secondary source is not always ready to process by the machine learning algorithm. So the dataset was processed to be used with machine learning algorithms. This process is called data preprocessing. In this stage, checking and adjusting the null values and data balancing is performed. The undersampling technique was used to handle the imbalanced distribution of data between the stroke and non-stroke classes.

After making the data ready to be used by machine learning algorithms, the dataset is divided into train and test data. The new data is then used to create a model using a variety of classification algorithms. For each of these methods, accuracy is calculated and compared to obtain the most accurate prediction model.

The main objective of this study is to predict the possibility of stroke based on stroke from the individual's medical history and physical activities. To achieve the result, we have applied different machine learning algorithms like Naïve Bayes, AdaBoost, Decision Table, k-NN, and Random Forest. The result obtained by training the different models is compared and the best model is suggested.

This paper is organized as follows: Section 2 describes the overview of the relevant works. In section 3, data description and working methodology is presented. Experimental results are described in section 4. Finally, section 5 concludes the results.

## 2. Related Works

One person dies from a stroke every four to five minutes, according to World Health Organization (WHO) estimates of the fifteen million individuals who experience them globally each year. Stroke is the sixth most common cause of death in the United States, according to the Centers for Disease Control and Prevention (CDC) [6]. About 11% of people die from non-communicable diseases like stroke each year. Approximately 795,000 Americans experience the incapacitating symptoms of strokes regularly [7]. To categorize stroke conditions in 507 people, Govindarajan et al. [8] employed text mining and a machine learning classifier. They investigated several artificial neural networks (ANN)-based machine learning techniques for training purposes and discovered that the SGD algorithm delivered the highest value, 95%.

Research on stroke prognosis was conducted by Amini et al. [9]. Fifty risk factors for stroke, diabetes, cardiovascular disease, smoking, hyperlipidemia, and alcohol consumption were categorized by them in 807 healthy and unhealthy people. The c4.5 decision tree algorithm and the K-nearest neighbor algorithm, both of which have accuracy rates of 95%, were employed by the researchers (94 percent accuracy).

A study on determining the prognosis of an ischemic stroke was given by Cheng et al. [10]. They used two ANN models, 82 ischemic stroke patient data sets, and accuracy values of 79 and 95 percent in their study. To ascertain whether a stroke patient's death may be predicted, Cheon et al. [11] conducted research. In their study, they calculated the stroke incidence using 15,099 people. They used a deep neural network technique to find strokes. To extract data from the medical records and forecast strokes, the authors used PCA. They have an area under the curve of 83 percent (AUC).

Artificial intelligence was used in the study by Singh et al. [12] to forecast strokes. They used the cardiovascular health study (CHS) dataset in their research and applied a novel method for predicting stroke. Additionally, they performed a feature extraction followed by a principal component analysis using the decision tree method. In this instance, a neural network classification method was used to build the model, and it had a 97 percent accuracy rate.

To ascertain the efficacy of automated early ischemic stroke detection, Chin et al. [13] conducted research. Their research's main goal was to develop a Convolutional Neural Network technique for automating primary ischemic stroke (CNN). For the goal of developing and testing the CNN model, the author gathered 256 images. To increase the gathered picture for their system's image preprocessing, used the data lengthening technique. CNN method had an accuracy rate of 90%.

The research was done by Sung et al. [14] to create a stroke severity index they collected information on 3577 people who suffered an acute ischemic stroke. They built their predictive models using several data mining techniques, including linear regression. The k-nearest neighbor algorithm fared worse than their ability to anticipate (95% confidence interval).

Machine learning was utilized by Monteiro et al. [15] to forecast the functional outcome of an ischemic stroke. They used a patient who passed away three months after admission to test this procedure. They achieved an AUC value of over 90. The research was undertaken to ascertain the risk of stroke by Kansadub et al. The authors of the study used Naive Bayes, decision trees, and neural networks to analyze the data and predict strokes. In their study, they evaluated the accuracy and AUC of their pointer. All of these algorithms were characterized by them as decision trees, with naive Bayes producing the most precise outcomes.

To identify the classification of an ischemic stroke, Adam et al. [16] undertook a study. They used the decision tree method and the k-nearest neighbor method to categorize ischemic strokes. Medical professionals found the decision tree method to be more helpful in their study when categorizing strokes. 90% accuracy was regarded to be a good accuracy rate for the majority of investigations.

With a classification accuracy of 96%, Khan et al. [17] use random forest classification that exceeds the other investigated techniques. According to the study, the random forest method performs better than other methods when forecasting brain strokes using cross-validation measures.

In this study, similar to the previous studies, different machine learning algorithms were trained and the best performing model is selected for the used data set.

## 3. Methodology

This section contains four sub-sections such as data description, data preprocessing, machine learning models, and implementation procedure. Each subsection is described below.

### *3.1 Data Description*

The dataset used in this research was obtained from the Kaggle data repository. The total number of participants was 4981 among which 2074 were male and 2907 were female. The dataset has 10 attributes as input for machine learning models and one target class. The attributes gender, age, hypertension, heart_disease, ever_married, work_type, residence_type, avg_glucose_level, bmi, and smking_status are the main attributes that are used as input for the machine learning model. The attribute stroke is used as the output variable. The number '0' denotes the absence of any stroke risk, while the number '1' denotes the possibility of stroke risk. The dataset used in this research is highly imbalanced as it has 248 rows with a value of '1' whereas 4733 rows have a value of '0' in the stroke column. To attain better accuracy, the data pre-processing technique is used to balance the dataset. The details of the dataset are described in table 1.

Table 1. Brain Stroke Dataset Attribute Description

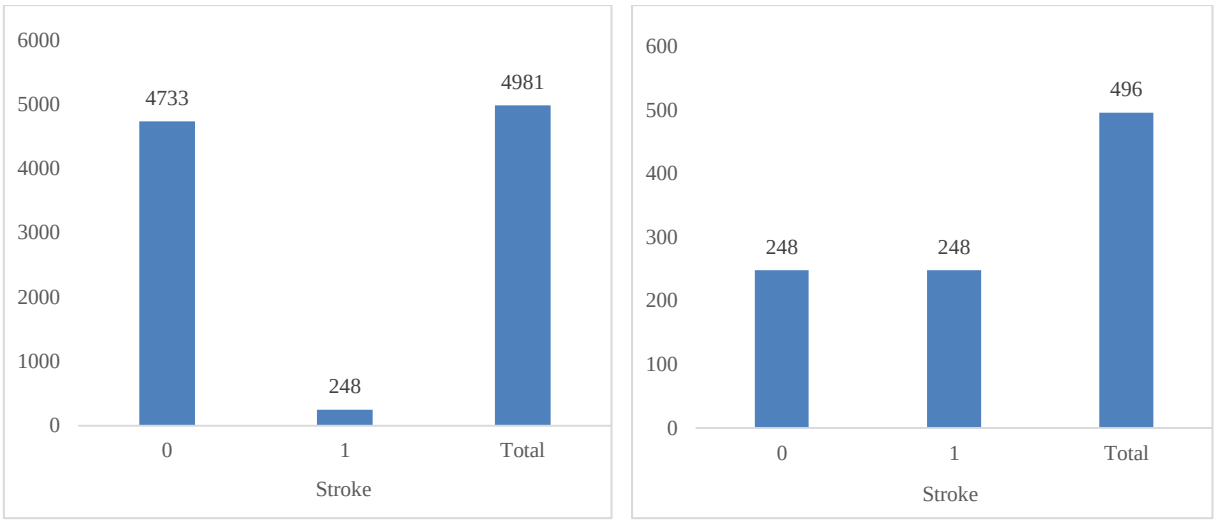
| Attribute Name | Type (Possible Values) | Description |
|---|---|---|
| Gender | String(Male, Female) | Describes the gender of the participant |
| Age | Floating point number (0.08 to 82) | Age of the participant |
| Hypertension | Numeric (0, 1) | Participants hypertension status |
| Heart_disease | Numeric (0, 1) | Participant's heart disease status |
| Ever_married | Nominal (Yes, No) | Tells whether the participant is ever married or not. |
| Work_type | String (private, self-employed, govt_job, children) | Describes the nature of the work of the participants |
| Residence_type | Nominal (Urban, Rural) | Tells the residence type of the participants |
| Avg_glucose_level | Floating point number (55.12 to 271.74) | Shows the average glucose level of the participants. |
| Bmi | Floating point number (14 to 48.9) | Gives the body mass index of the participant |
| Smoking_status | String (formerly smoked, never smoked, smokes, unknown) | Shows the smoking status of the participant |
| Stroke | Numeric (0, 1) | Response variable which describes the stroke status |

### *3.2 Data Preprocessing*

The raw data might contain noise and/or missing values affecting negatively in the final prediction. Hence, preprocessing of the data is necessary. This stage deals with anything preventing the model from operating more effectively. Preprocessing of the dataset includes feature selection, values reduction, and discretization. The dataset taken for the research has 11 attributes including the response variable. Firstly the dataset is checked for null values and if occurs it is filled. After adjusting null values, the string values are converted to nominal as WEKA cannot process string values.

The dataset used for this study is very unbalanced. The entire dataset contains 4981 rows, among which 248 rows in the stroke column have the value '1' whereas 4733 rows have the value '0'. If such uneven data is not managed, the predictions and outcomes are ineffective. The undersampling technique was used in this research to handle the imbalanced distribution of data between the stroke and non-stroke classes. By implementing undersampling the majority class is undersampled to match the minority class. More specifically, the majority of class 'stroke' with value '0' was undersampled for the class 'stroke' with value '1' to equally distribute the participants. After implementing undersampling, the dataset contains 248 rows with value '0' and 248 rows with value '1' resulting in 496 total rows. Figure 1 displays a graphical depiction of the response variable before and after applying undersampling in the dataset.

The next step after handling the imbalanced dataset and finishing data preprocessing is to develop a model. The dataset obtained after under-sampling is split into training and testing data. In this research, we have used a 10-fold cross-validation technique to get a better result. In the 10-fold cross-validation technique, 90% of the total training dataset is randomly selected as training data and the remaining 10% data as test data. After splitting the dataset, we used classification algorithms to train the model. The different machine learning algorithms used to train the model are described below.




(a) (b)

Fig. 1. (a). Dataset before Under-sampling, (b). Dataset after Under-sampling

### *3.3 Machine Learning Models*

#### *a. Na ïve Bayes*

The Naive Bayes model is used in supervised learning algorithms and relies on the Bayes theorem. The Bayes theorem tells the conditional probability of occurrence of an event against another event which is already occurred. The conditional probability of Naive Bayes can be represented as follows:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \quad (1)$$

A Naive Bayes classifier is implacable in the dataset containing many attributes. Additionally, it is good at handling small datasets which require less training data, is highly scalable, adept at handling both continuous and discrete data, and is insensitive to irrelevant features [18].

#### *b. Adaptive Boosting (AdaBoost)*

The AdaBoost algorithm is the most common and commonly used ensemble learning approach. Boosting is a method that combines all weak classifiers into a single, effective classifier. The character of AdaBoost is that it creates a weak learner from the initial training data and then modifies the distribution of the training data for the subsequent weak learner training round based on the predicted performance. The samples with low prior prediction accuracy in previous steps are will get more attention in the next step. Finally, the weaker learners are combined with varying weights to design a powerful learner [19].

#### *c. Decision Table*

Decision tables (DTs) present an alternate method for describing rule-based classification models in a user-friendly way [20]. DTs are represented in a tabular form and are used to describe and analyze decision circumstances (such as the evaluation of credit risk), in which the existence of several criteria affects the execution of a series of actions [21]. In our context, the conditions correspond to the outcome of stroke or not stroke. A decision table consists of four quadrants as shown in table 2. The horizontal line is used to divide the table into a condition part (above) and an action part (below). The vertical line is used to separate subjects (left) from entries (right).

Table 2. Decision Table quadrants

| | |
|---|---|
| Condition Subjects | Condition Entries |
| Action Subjects | Action Entries |

The criteria that are important to the decision-making process are the condition subjects. They stand for the characteristics of the rule preceding for which information is required to categorize whether or not the participant has a stroke. For each condition subject (attribute), each condition entry defines a relevant subset of values (referred to as a state), or it contains the dash sign ('-') if the value is unimportant about that column. The values allocated to the associated action subject (class) are then stored in the action entries, with the "x" entry designating the value that corresponds to a certain set of circumstances. Thus, each column in the DT's entry section has a classification rule.

*d. Random Forest*

Random Forest classifier follows the theory of ensemble learning. It is a process that uses a variety of classifiers to improve how this technology is implemented to address repeated drawbacks. Random forests consist of multiple independent decision trees which are trained independently on a random subset of data. With this technique, each decision tree casts a vote for a certain output class, in our case 'stroke' or 'no stroke'. The class with the most votes is selected as the final forecast by the random forest [22]. The following formula can be used to determine the weight of each feature in a decision tree.

$$fi_i = \frac{\sum ni_j}{\frac{n{:}node\ j\ splits\ on\ feature\ i}{\sum ni_k}} \quad (2)$$
$$k \in all\ nodes$$

Where '$fi_i$' indicates the '$i^{th}$' feature importance and '$ni_j$' defines the importance of node 'j'.

*e. K-nearest Neighbors Classification (k-NN)*

KNN is a non-parametric supervised learning algorithm and it is categorized as a lazy learning approach. It does not train the given dataset immediately but stores the dataset and performs the action at the time of classification [22]. The basic working principle of k-NN is to find the similarities between the new data and the existing data, which then maps the new example into the category that best matches the existing categories. To make predictions for a new instance, the entire training set is searched for the k most similar instances which are called the neighbors. These k instances determine the prediction of the output.

Different distance metrics such as Euclidean distance, Minkowski distance, and Manhattan distance can be used with k-NN. In this research, we have used Euclidean distance for the model. Euclidean distance is a direct path that connects two points. The Euclidean distance between two points can be calculated using the following formula [23].

$$d(A,B) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (3)$$

*3.4 Implementation Procedure*

To predict stroke in participants, this study uses the machine learning method to build a prediction model. The data sets collected from the Kaggle repository were examined using the WEKA toolkit to determine the best and most likely association between them. Preprocessing and data manipulation utilizing data mining methods were the next steps. The knowledge representation of the results was used after implementing several data mining (classification) techniques, such as Naïve Bayes, Adaptive Boosting (AdaBoost), Decision Table, Random Forest, and K-nearest Neighbors Classification (k-NN). After building the models, each model is compared based on the different accuracy matrices such as accuracy score, precision score, recall score, F-measure, and ROC curve. The working procedure of the research is shown in figure 2.

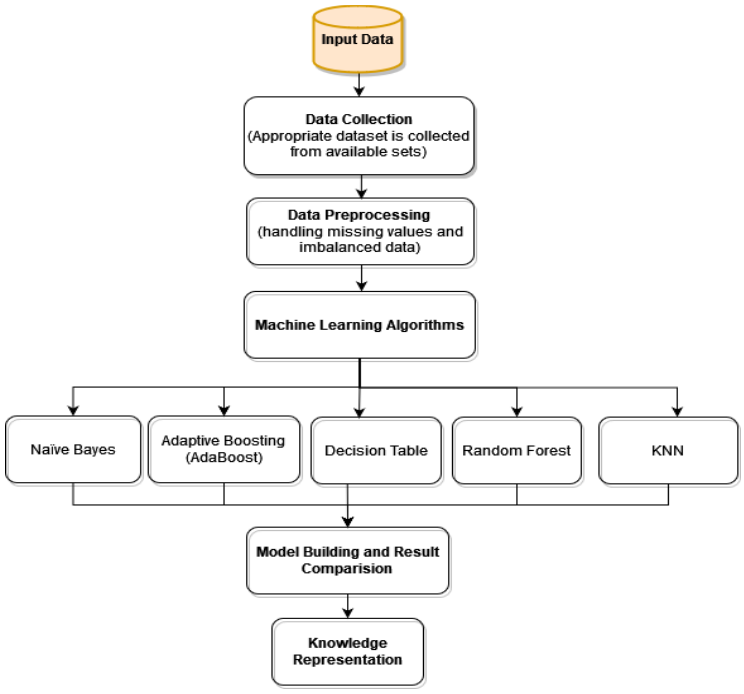



Fig. 2. System Flow Diagram

The proposed methodology involves different steps from data acquisition to model prediction. To achieve the objective of the study the proposed helped by allowing to adopt data from the repository, balance the unbalanced dataset and clean the data set, to choose different machine learning models, to train different proposed machine learning models and finally to predict the best model.

## 4. Result and Discussion

In this section, the performance of machine learning models is evaluated using WEKA 3.8.6 [24] environment. WEKA is a free data mining tool. GNU General Public License was used to create and distribute WEKA. It contains a rich library of different models for data preprocessing, classification, association, visualization, clustering, and many more.

### *4.1 Evaluation Metrics*

Several performance indicators were logged during the consideration and evaluation of the ML models. In this research, we have considered the most commonly considered parameters such as accuracy score, precision score, recall score, F-Measure, and ROC.

The number of positive class forecasts that fall within the positive class is measured by precision. Recall measures how many correct class predictions were produced using all of the positive cases in the dataset. F-Measure provides a value that balances both precision and recall in one number. Accuracy, precision, recall, and F-measure can be calculated with the formula shown in equation 4 to 7.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{4}$$

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

$$\text{F} - \text{Measure} = 2 * \frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \tag{7}$$

Here, TP denotes true positive, TN indicates true negative, FP indicates false positive and FN indicates false negative.

### *4.2 Performance evaluation of machine learning models*

In this study, we have a 10-fold cross-validation technique to train and validate the models. After applying the 10-fold cross-validation technique, we have used classification algorithms to train the model. The performance result obtained from different machine learning algorithms used to train the model is demonstrated in table 3.

Table 3. Performance results of Machine learning algorithms

| Model | Accuracy | Precision | Recall | F-Measure |
|---|---|---|---|---|
| **Naïve Bayes** | 0.756 | 0.719 | 0.756 | 0.734 |
| **AdaBoost** | 0.801 | 0.756 | 0.801 | 0.755 |
| **Decision Table** | 0.821 | 0.796 | 0.821 | 0.784 |
| **Random Forest** | 0.801 | 0.751 | 0.801 | 0.744 |
| **k-NN** | 0.756 | 0.719 | 0.756 | 0.734 |

From table 3 we can see that the Naïve Bayes classifier and k-NN classifier gives 75.6% of accuracy. The accuracy of the AdaBoost classifier and Random Forest classifier is 80.1%. The Decision Table classifier has the highest accuracy of 82.1%. The precision, recall, and F-Measure for Naïve Bayes classifier are 71.9%, 75.6%, and 73.4% respectively. The AdaBoost classifier has 75.6%, 80.1%and 75.5% precision, recall, and F-Measure respectively. Similarly, the precision, recall, and F-Measure for Decision table are 79.6%, 82.1%, and 78.4% respectively. The Random Forest classifier got the 75.1%, 80.1%, and 74.4 % of precision, recall, and F-Measure. The k-NN classifier has 71.9%, 75.6%, and 73.4% of precision, recall, and F-Measure.

After model building, it is concluded that the Decision Table classifier has the best performance with 82.1% accuracy compared to other classifiers for the dataset used in this research. Additionally, if we compare the precision and recall, Naïve Bayes and k-NN have the same performance but Decision Table performs better. The maximum difference with these parameters among different algorithms is 7.7%. While considering F-Measure, Decision Table is 5%

higher than Na ïve Bayes and k-NN, 4% higher than Random Forest, and 2.9% higher than AdaBoost. The accuracy of different machine learning models used in this research is shown in figure 3.

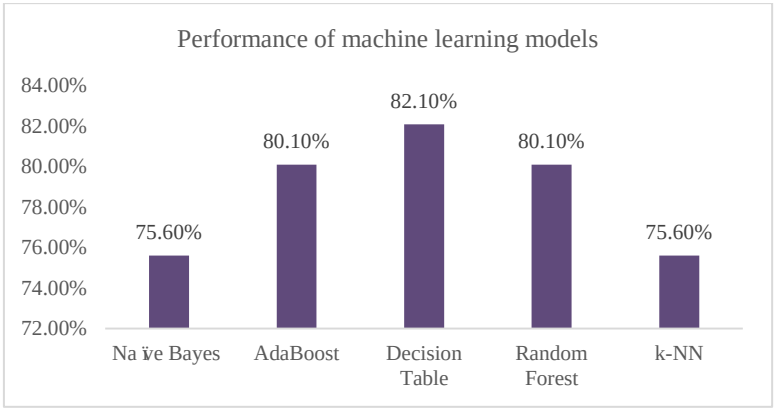



Fig. 3. Performance analysis of machine learning models

The comparison of different performance parameters such as precision, recall, and F-Measure obtained after training different models used in this research is demonstrated in figure 4. It can be seen that the decision table performed the best in each parameter compared to other models.

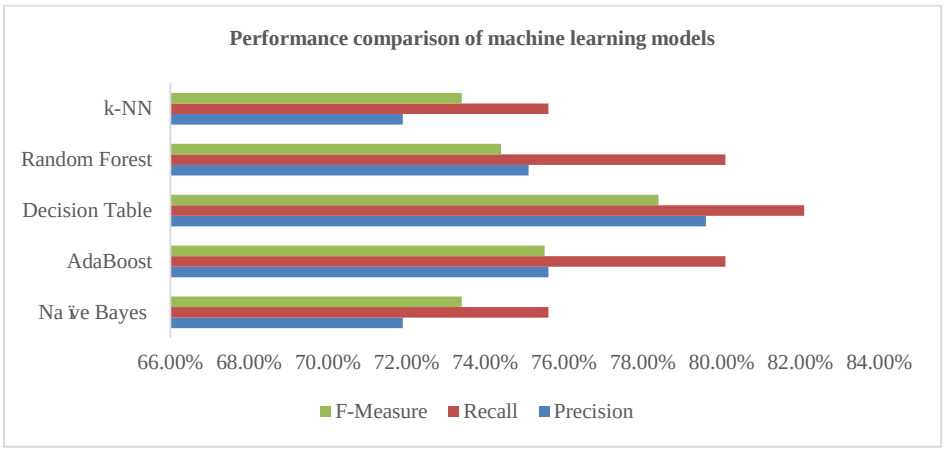



Fig. 4. Performance comparison of different machine learning models

The Receiver operating characteristic (ROC) curve for the Decision Table classifier is shown in figure 5.

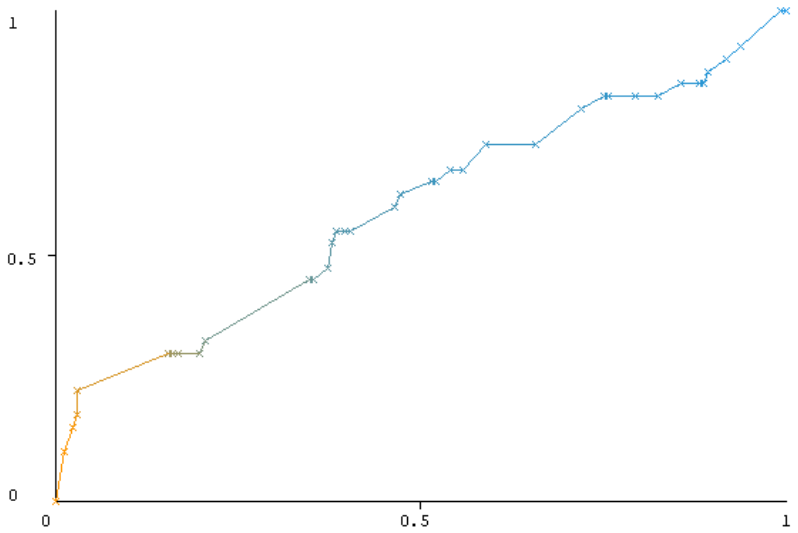



Fig. 5. The ROC curve for Decision Table classification

A measurement tool for binary classification issues is the Receiver Operator Characteristic (ROC) curve. In essence, it separates the "signal" from the "noise" by plotting the true positive rate (TPR) against the false positive rate (FPR) at different threshold values. The capacity of a classifier to differentiate between classes is measured by the Area Under the Curve (AUC), which is used as a summary of the ROC curve. From figure 5, we can see that AUC is in between 0.5 and 1. This indicates that the classifier has performed to predict true positive and true negative in better way.

A limitation of this study is that it was conducted based on publicly available datasets. The dataset has limited features and size which might not predict the actual result. If real data from the hospital or medical institution are available with the patient's detailed health profile, a more accurate model can be developed to get better predictions.

## 5. Conclusion

Stroke causes a significant number of fatalities and is increasing every day. Several stroke risk factors are responsible for different types of strokes. The design of a machine learning model can aid in the early detection of stroke and minimize its severe effects. These performance indicators of different machine learning algorithms used in this research show the successful prediction of stroke. Five machine learning algorithms such as Naïve Bayes, AdaBoost, Decision Table, k-NN, and Random Forest were used to detect the stroke. All these machine learning models were trained using WEKA environment to achieve the performance of the models. The result obtained after training all five models shows that Decision Table performs better than other methods. The model can be helpful for the clinical prediction of stroke in a better way.

In the future, we can extend the research by applying multiple classification techniques and designing a framework to display the predicted results.

## References


[1] "World Stroke Organization," Available: *https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learnabout-stroke* . [Accessed: August 10, 2022]

[2] T. R. Dawber, G. F. Meadors, and F. E. Moore, "Epidemiological Approaches to Heart Disease: The Framingham Study*."

[3] P. A. Wolf, R. B. D, A. J. Belanger, and W. B. Kannel, "Probability of Stroke: A Risk Profile From the Framingham Study." [Online]. Available: http://ahajournals.org

[4] M. S. Donaldson, J. M. Corrigan, and L. T. Kohn, "To err is human: building a safer health system," *National academy press Washington, DC*, vol. 6, 2000.

[5] "Brain stroke prediction dataset," *https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset*.

[6] "Concept of stroke by healthline," *https://www.cdc.gov/stroke/index.htm*. [Accessed: August 12, 2022]

[7] "Statistics of stroke by Centers for disease control and prevention," *https://www.cdc.gov/stroke/facts.htm*. [Accessed: August 12, 2022]

[8] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, and R. Manikandan, "Classification of stroke disease using machine learning algorithms," *Neural Computing & Applications*, vol. 32, no. 3, pp. 817–828, 2020.

[9] L. Amini, R. Azarpazhouh, and M. T. Farzadfar, "Prediction and control of stroke by data mining," *International Journal of Preventive Medicine*, vol. 4, no. 2, pp. S245–S249, 2013.

[10] C. A. Cheng, Y. C. Lin, and H. W. Chiu, "Prediction of the prognosis of ischemic stroke patients after intravenous thrombolysis using artificial neural networks," *Studies in Health Technology and Informatics*, vol. 202, pp. 115–118, 2014.

[11] S. Cheon, J. Kim, and J. Lim, "The use of deep learning to predict stroke patient mortality," *International Journal of Environmental Research and Public Health*, vol. 16, no. 11, 2019.

[12] M. S. Singh and P. Choudhary, "Stroke prediction using artificial intelligence," in *Proceedings of the 2017 8th Annual Industrial Automation And Electromechanical Engineering Conference (IEMECON)*, Aug. 2017, pp. 158–161.

[13] C.-L. Chin, B.-J. Lin, and G.-R. Wu et al., "An automated early ischemic stroke detection system using CNN deep learning algorithm," in *Proceedings of the 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST)*, Nov. 2017, pp. 368–372.

[14] S.-F. Sung, C.-Y. Hsieh, and Y.-H. Kao Yang et al., "Developing a stroke severity index based on administrative data was feasible using data mining techniques," *Journal of Clinical Epidemiology*, vol. 68, no. 11, pp. 1292–1300, 2015.

[15] M. Monteiro, A. C. Fonseca, and A. T. Freitas et al., "Using machine learning to improve the prediction of functional outcome in ischemic stroke patients," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1953–1959, 2018.

[16] S. Y. Adam, A. Yousif, and M. B. Bashir, "Classification of ischemic stroke using machine learning algorithms," *International Journal of Computer Application*, vol. 149, no. 10, pp. 26–31, 2016.

[17] T. Tazin, M. N. Alam, N. N. Dola, M. S. Bari, S. Bourouis, and M. Monirujjaman Khan, "Stroke Disease Detection and Prediction Using Robust Learning Approaches," *Journal of Healthcare Engineering*, vol. 2021, 2021, doi: 10.1155/2021/7633381.

[18] 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 *In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE.*, 2019, pp. 1–6.

[19] D.-C. Feng *et al.*, "Machine learning-based compressive strength prediction for concrete: an adaptive boosting approach."

[20] G. Wets, J. Vanthienen, and S. Piramuthu, "Extending a tabular knowledge-based framework with feature selection," *Expert Systems with Applications*, vol. 13, no. 2, pp. 109–119, 1997.

[21] J. Vanthienen and E. Dries, "Illustration of a decision table tool for specifying and implementing knowledge based systems," *International Journal on Artificial Intelligence Tools*, vol. 3, no. 2, pp. 267–288, 1994.
[22] G. Sailasya and G. L. Aruna Kumari, "Analyzing the Performance of Stroke Prediction using ML Classification Algorithms." [Online]. Available: www.ijacsa.thesai.org
[23] A. Pandey and A. Jain, "Comparative Analysis of KNN Algorithm using Various Normalization Techniques," *International Journal of Computer Network and Information Security*, vol. 9, no. 11, pp. 36–42, Nov. 2017, doi: 10.5815/ijcnis.2017.11.04.
[24] "WEKA Tool," *Available Online: https://www.weka.io/*. [Accessed: August 8, 2022]

## Authors' Profiles

**Trailokya Raj Ojha** is working as an Assistant Professor at Department of Computer Science and Engineering at Nepal Engineering College, Changunarayan Bhaktapur, Nepal. He has received Master's degree in Software Systems from Tampere University of Technology in 2014. He has been involved in software development and academic profession since 2008. His research fields include software engineering, process improvement, software project management, Internet of Things, and machine learning.

**Ashish Kumar Jha** received his Master's degree in Computer Science Specialization in Networking from Sharda University in 2017. He has involved in software development and teaching profession since 2013, currently working as assistant professor in Nepal Engineering College. His research interest includes, Internet of Things, Image Processing and pattern Recognition.

**How to cite this paper:** Trailokya Raj Ojha, Ashish Kumar Jha, "Analyzing the Performance of the Machine Learning Algorithms for Stroke Detection ", International Journal of Education and Management Engineering (IJEME), Vol.13, No.2, pp. 27-35, 2023. DOI:10.5815/ijeme.2023.02.04