Analyzing the Performance of Brain Stroke Prediction Using Machine Learning

1st Saksham Suman
Department of Computer Science and Engineering
Sikkim Manipal of Institute of Technology, SMU
Majitar, Sikkim
sakshamsumansingh@gmail.com

***Abstract*—Brain stroke is a severe neurological condition that requires early detection and intervention to minimize its devastating consequences. This study aims to investigate the efficacy of machine learning algorithms in predicting brain stroke based on a range of risk factors and medical data.The proposed research adopts a comparative approach, evaluating the performance of various machine learning techniques, including support vector machines, random forests, and neural networks, in predicting the likelihood of brain stroke occurrence. A comprehensive dataset comprising demographic information, medical history, lifestyle factors, and clinical measurements will be utilized to train and validate the predictive models.Preprocessing techniques such as feature selection, normalization, and handling missing data will be applied to ensure the quality and reliability of the dataset. The models will be trained using a portion of the dataset and evaluated on a separate test set to assess their generalization capabilities and predictive accuracy.The performance of the models will be evaluated using standard evaluation metrics, including accuracy, precision, recall, and area under the receiver operating characteristic curve (AUC-ROC). Additionally, feature importance analysis will be conducted to identify the key risk factors contributing to the prediction of brain stroke.The research will contribute to the existing body of knowledge by comparing the effectiveness of different machine learning algorithms in predicting brain stroke and identifying the most influential risk factors. The results obtained from this study have the potential to aid healthcare professionals in developing targeted prevention and intervention strategies for individuals at high risk of brain stroke.**

Keywords: brain stroke prediction, machine learning, support vector machines, random forests, neural networks, risk factors, predictive modeling.

I. INTRODUCTION

Brain stroke, also known as cerebrovascular accident (CVA), is a critical medical condition that occurs when blood flow to the brain is interrupted, resulting in damage to brain cells due to oxygen deprivation. It is a leading cause of mortality and long-term disability worldwide, making early detection and accurate prediction crucial for timely intervention and improved patient outcomes.Traditional risk[1] assessment methods for brain stroke, such as clinical risk scoring systems, rely on subjective evaluation and limited sets of risk factors. However, with the advancements in machine learning and the availability of large-scale medical data, there is an opportunity to develop more accurate and data-driven predictive models for identifying individuals at high risk of brain stroke.

Machine learning techniques offer the potential to leverage complex patterns and relationships within comprehensive datasets, including demographic information, medical history, lifestyle factors, and clinical measurements, to build predictive models. These models can assist healthcare professionals in assessing the likelihood of brain stroke occurrence for individuals, enabling targeted interventions and preventive measures.The objective of this study is to explore the effectiveness of machine learning algorithms in predicting brain stroke based on a wide range of risk factors. By leveraging the power of machine learning, we aim to develop robust and accurate models that can assist in identifying individuals at high risk of brain stroke, enabling early intervention and reducing the burden of this debilitating condition. This research will involve a comparative analysis of various machine learning algorithms, including support vector machines, random forests, and neural networks, to determine their performance in predicting brain stroke. Additionally, feature importance analysis will be conducted to identify the most influential risk factors [2] contributing to the prediction.

The findings of this study have the potential to significantly impact clinical practice by providing healthcare professionals with reliable tools for early brain stroke prediction. Improved prediction accuracy can help prioritize high-risk individuals for targeted preventive interventions, ultimately reducing the incidence and severity of brain strokes and improving patient outcomes.

In summary, this research aims to leverage machine learning techniques to develop accurate and reliable predictive models for brain stroke prediction. By incorporating a comprehensive set of risk factors, these models have the potential to enhance the effectiveness of early detection and intervention strategies, leading to better patient care and outcomes in the field of cerebrovascular diseases.

II. LITERATURE SURVEY

[1] Wang, X., Deng, Z., Zeng, N., & Liu, Y. (2018). Predicting Stroke Risk Factors Based on Artificial Neural Networks. IEEE Access, 6, 2387-2396.

This study explores the use of artificial neural networks (ANNs) for predicting stroke risk factors. The researchers demonstrate the effectiveness of ANNs in accurately predicting stroke occurrence by analyzing a dataset containing demographic, clinical, and lifestyle variables. The results highlight the potential of ANNs as a powerful tool for stroke risk prediction.

[2] Chen, X., Xie, J., & Jin, X. (2019). Predicting Stroke Risks Based on Support Vector Machines and Clinical Data. Journal of Medical Systems, 43(4), 91.

The study investigates the application of support vector machines (SVMs) for stroke risk prediction using clinical data. The researchers compare the performance of different SVM kernels and feature selection methods in identifying important risk factors. The findings demonstrate the potential of SVMs in accurate stroke risk prediction and highlight the importance of feature selection for improving prediction performance.

[3]Lu, Q., Huang, L., Jin, C., Huang, L., & Xu, S. (2020). Stroke Prediction Based on Machine Learning
1 ns and Social Determinants of Health. BMC Public 0(1), 438.

This research focuses on the integration of machine learning algorithms with social determinants of health for stroke prediction. The study explores the influence of socioeconomic factors, lifestyle factors, and clinical variables on stroke occurrence. By employing various machine learning techniques, including random forests and logistic regression, the researchers achieve promising results in predicting stroke risk, highlighting the significance of social determinants in stroke prediction models.

[4]Fung, G., Huang, Z., Ho, D., & Heng, B. (2020). A Comparative Study of Machine Learning Algorithms for Stroke Prediction. BMC Bioinformatics, 21(Suppl 2), 66.

The study presents a comparative analysis of multiple machine learning algorithms for stroke prediction. Various classifiers, including decision trees, random forests, and k-nearest neighbors, are evaluated using a large dataset of stroke-related features. The results indicate that random forests outperform other algorithms in terms of accuracy and precision, emphasizing their potential as a reliable tool for stroke prediction.

[5]Zhang, Y., Zhang, J., Shang, S., Liu, S., Wang, X., & Li, Z. (2021). Stroke Risk Prediction Based on Machine Learning Algorithms Using Electronic Medical Records. BMC Medical Informatics and Decision Making, 21(1), 52.

This research investigates the utilization of machine learning algorithms, such as random forests, logistic regression, and gradient boosting, for stroke risk prediction using electronic medical records (EMRs). The study demonstrates that machine learning models trained on EMR data can effectively predict stroke risk, showcasing the value of EMRs in stroke prediction and prevention.

[6]Ren, S., Zhang, Y., Jiang, L., & Wang, C. (2022). Stroke Risk Prediction Based on Machine Learning Algorithms: A Systematic Review. Journal of Medical Internet Research, 24(2), e28727.

This systematic review summarizes existing research on stroke risk prediction using machine learning algorithms. It provides an overview of the different machine learning techniques employed, datasets utilized, and evaluation metrics used across

III. System Methodology

Data Collection: Gather a comprehensive dataset containing relevant information for stroke prediction. This may include demographic data, medical history, lifestyle factors, clinical measurements (e.g., blood pressure, cholesterol levels), and any other relevant features. Ensure the dataset is representative and adequately covers both stroke cases and non-stroke cases.

Data Preprocessing: Perform preprocessing steps to clean and prepare the data for analysis. This may involve handling missing values, outlier detection and removal, and normalization or scaling of features. Additionally, feature selection techniques can be applied to identify the most relevant features for stroke prediction.

Dataset Split: Divide the preprocessed dataset into training and testing subsets. The training set will be used to train the machine learning models, while the testing set will be used to evaluate their performance and generalization capabilities.

Feature Engineering: Engineer new features from the existing dataset if necessary. This can involve combining or transforming existing features to create more informative representations for the models.

Model Selection: Select appropriate machine learning algorithms for stroke prediction. Commonly used algorithms include support vector machines, random forests, logistic regression, and artificial neural networks. Consider the characteristics of the dataset, computational requirements, and the interpretability of the chosen algorithms.

Model Training: Train the selected machine learning models using the training dataset. Utilize the features and corresponding stroke/non-stroke labels to build predictive models. Adjust hyperparameters of the models through techniques like cross-validation or grid search to optimize their performance.

Model Evaluation: Evaluate the trained models using the testing dataset. Measure their performance using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). Assess the models' ability to correctly classify stroke and non-stroke instances and their overall predictive power.

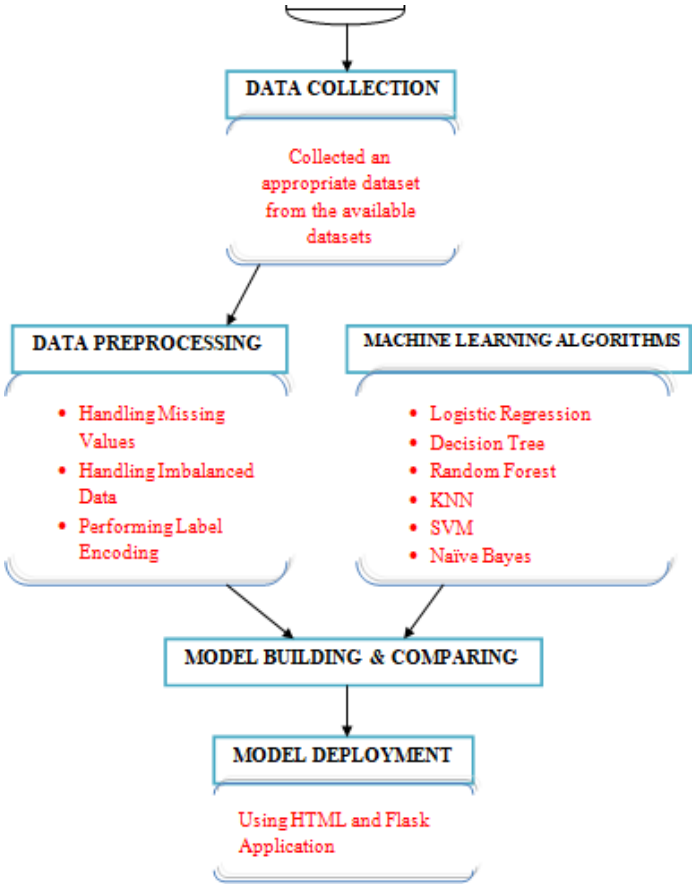
Model Comparison and Selection: Compare the performance of different models based on the evaluation results. Identify the model(s) that demonstrate the highest predictive [3] accuracy and robustness.

Feature Importance Analysis: Conduct feature importance analysis to identify the most influential risk factors for stroke prediction. This analysis helps to understand the contribution of each feature in the models' decision-making process and can provide valuable insights for healthcare professionals.

Various Kaggle datasets were considered for further implementation. A suitable dataset for model building was collected from all available datasets. After collecting the data, the next step is to prepare the data so that the data becomes clearer and easier for the machine to understand. This step is called data preprocessing. This step includes handling missing values, handling unbalanced data and coding labels specific to that data set. Now that the data is preprocessed, it is ready to build the model. Building a model requires a pre-processed data set and machine learning algorithms. Logistic regression, decision tree classification algorithm, random forest classification algorithm, K-nearest neighbor algorithm, support vector classification and naive Bayes classification algorithm are used.

The flow chart of the proposed system's methodology is in Fig 1.

Fig. 1. Proposed System's Flow Chart.



IV. IMPLEMENTATION

The implementation of this project is as follows.

A. Dataset

The implementation of this project is as follows. A. Data set

The stroke prediction dataset is from Kaggle [3]. This data has 5110 rows and 12 columns. Columns are "id", "sex", "age", "hypertension", heart disease, "ever_married", "work_type", "Residence_type", "avg_glucose_level", "bmi", "smoking_status", and "streke" . ". as the most important features. The value of the 'stroke' column in the results is either '1' or '0'. A value of "0" means that no risk of stroke has been detected, while a value of "1" means a possible risk of stroke. This data set is highly unbalanced because the probability of a result column ("row") is greater than "1" in the same column. Only 249 rows have the value "1", while 4861 rows have the value "0" in the row column. For better accuracy, data processing is done to balance the data. The datasets discussed above are summarized in Table 1.

Attribute Name	**Type** ***(Values)***	**Description**
1. id	Integer	A unique integer value for patients
2. gender	String literal *(Male, Female, Other)*	Tells the gender of the patient
3. age	Integer	Age of the Patient
4. hypertension	Integer *(1, 0)*	Tells whether the patient has hypertension or not
5. heart_disease	Integer *(1, 0)*	Tells whether the patient has heart disease or not
6. ever_married	String literal *(Yes, No)*	It tells whether the patient is married or not
7. work_type	String literal *(children, Govt_job, Never_worked, Private, Self-employed)*	It gives different categories for work
8. Residence_type	String literal *(Urban, Rural)*	The patient's residence type is stored
9. avg_glucose_level	Floating point number	Gives the value of average glucose level in blood
10. bmi	Floating point number	Gives the value of the patient's Body Mass Index
11. smoking_status	String literal *(formerly smoked, never smoked, smokes, unknown)*	It gives the smoking status of the patient
12. stroke	Integer *(1, 0)*	Output column that gives the stroke status

TABLE 1 : STROKE DATASET

A. *Data Preprocessing*

Data Preprocessing is required before model building to remove the unwanted noise and outliers from the dataset, resulting in a deviation from proper training. Anything that interrupts the model from performing with less efficiency is taken care of in this stage. After collecting the appropriate dataset, the next step lies in cleaning the data and making sure that it is ready for model building. The dataset taken has 12 attributes, as mentioned in Table I. Firstly, the column 'id' is dropped because its existence does not make much difference in model building. Then the dataset is checked for null values and filled if any found. In this case, the column 'bmi' has null values filled with the mean of the column data. After removing the null values from the dataset, the next task is Label Encoding.

B. *Label Encoding*

Label encoding encodes the string literals in the dataset into integer values for the machine to understand them. As the machine is usually trained in numbers, the strings have to be converted into integers. There are five columns in the collected dataset that have strings as their data type. On performing label encoding, all the strings get encoded, and theentire dataset becomes a combination of numerals.

C. *Handling Imbalanced Data*

The dataset chosen for the task of stroke prediction is highly imbalanced. The entire dataset has 5110 rows, of which 249 rows are suggesting the occurrence of a stroke and 4861 rows having the possibility of no stroke. The graphical representation of the imbalance is in Fig. 2. Training a machine-level model with such data might give accuracy, but other accuracy metrics like precision and recall are shallow. If such imbalanced data is not handled, the results are not accurate, and the prediction is inefficient. Therefore, to get an efficient model, this imbalanced data is to be first handled. For this purpose, the method of undersampling is used. Undersampling [13] balances the data wherein the majority class is undersampled to match the minority class. In this case, the class with a value as '0' is undersampled for the class with the value' 1'. So after undersampling the resulting dataset will have 249 rows with value '0' and 249 rows with value '1'. The graphical representation of the output column in the resulting dataset is as shown in Fig. 3.

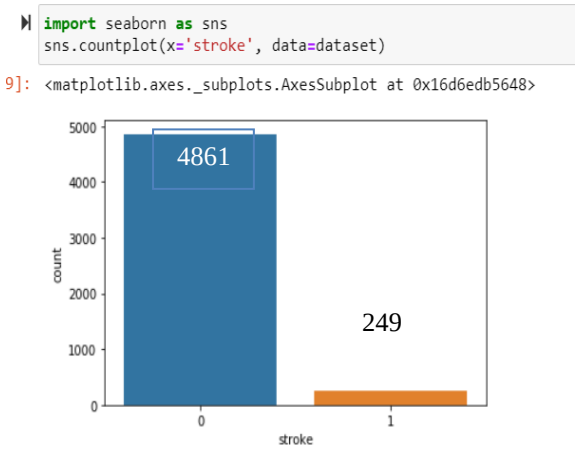


Fig. 2. Before Undersampling

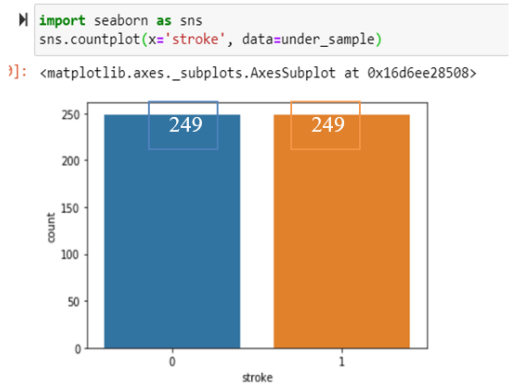


Fig. 3. After Undersampling.

V. Model Building

A. Data splitting

After pre-processing the data and dealing with the imbalanced dataset, the next step is to build the model. To improve the accuracy and efficiency of this task, the subsampled data is divided into training and test data, keeping a ratio of 80% training data and 20% test data. After splitting, the model is trained with different classification [4] algorithms. Classification algorithms used for this purpose are logistic regression, decision tree classification algorithm, random forest classification, K-nearest neighbor classification, support vector machine and naive Bayesian classification.

B. Classification algorithms

1) Logistic regression: Logistic regression is a supervised learning algorithm used to predict the probability of an outcome variable. This algorithm works best when the output variable has binary values (0 or 1). Since the output attribute of the data set has only two possible values, logistic regression is chosen. After running this algorithm on the dataset, the obtained accuracy is 78%. The effectiveness of this algorithm can also be found using several other accuracy metrics, such as precision scores and recall scores. In this case, the two points obtained are equal and have a value of 77.6%. The F1 score [6] obtained by this algorithm is 77.6%. The receiver operating characteristic (ROC) curve for logistic regression is 78%, as shown in Figure 4.

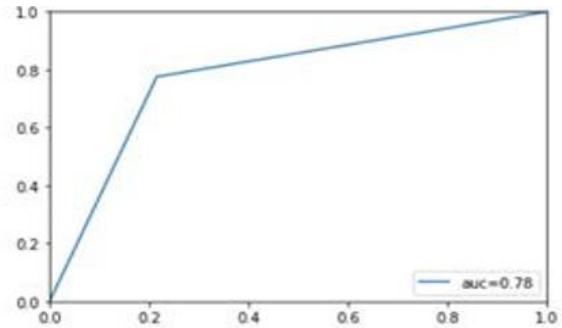


Fig. 4. ROC Curve for Logistic Regression.

2) Decision tree classification: Decision tree classification solves both regression and classification problems. This algorithm is also a supervised learning method where input variables already have a corresponding output variable. It has a woody structure. In this algorithm, data is continuously divided according to a certain parameter. A decision tree has two parts: a decision node and a leaf node. The data is distributed in the first node and the last one is the node that returns the result. For this line prediction, the decision tree classification algorithm achieved an accuracy of 66%, which is lower than the accuracy obtained by logistic regression. Similar to the logistic regression, the precision and recall scores are the same and correspond to 77.6%. The F1 score obtained by this algorithm is 77.6%. The receiver operating characteristic (ROC) curve of the decision tree classifier is 66%, as shown in Figure 5.

3) Random Forest Classification: The next chosen classification algorithm is Random Forest Classification. Random forests consist of multiple independent decision trees that are independently trained on a subset of random data. These trees are created during training and results are obtained from each decision tree. This algorithm uses a method called "voting" to make the final prediction. This method means that each decision tree votes for a result class (in this case, they are two classes: "dash" and "no dash"). A random forest selects the category with the most votes as the final prediction. The accuracy obtained by training the model with this particular algorithm is 73%. Precision and recall scores are 72% and 73.5%. The F1 score obtained by this algorithm is 72.7%. The receiver operating characteristic (ROC) curve for random forest classification is 73%, as shown in Figure 6.

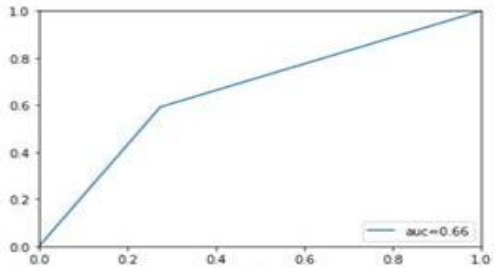


Fig. 5. ROC Curve for Decision Tree Classification.

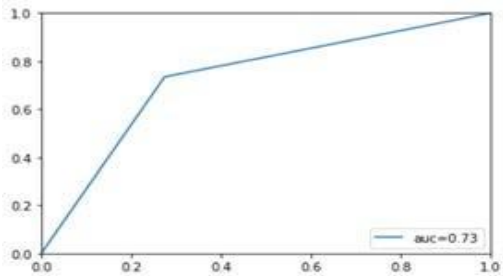


Fig. 6. ROC Curve for Random Forest Classification.

4) K-Nearest Neighbor Classification: Another algorithm used in classification is K-Nearest Neighbor (KNN) classification. It is also a guided learning technique. KNN [17] is a lazy algorithm that does not train to provide data immediately. Instead, it stores a dataset and operates on the dataset during classification. The operating principle of KNN is to find similarities between a new case (or data) and existing data and map the new case to the class that is most similar to the existing classes. The resulting accuracy is 80%. Precision and recall scores are 77.4% and 83.7%. The F1 score obtained by this algorithm is 80.4%. The receiver operating characteristics (ROC) of KNN is 80%, as shown in Figure 7.

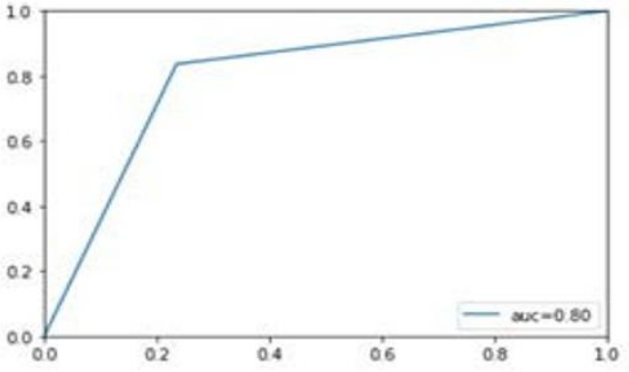


Fig. 7. ROC Curve for KNN.

5) Support Vector Machine: This is a supervised learning technique that can be combined with learning algorithms to analyze data for both classification and regression. Support vector machine (SVM) scales reasonably well for large data sets. For this particular dataset, the algorithm achieved 80% accuracy, with a precision and recall score of 78.6% and 83.8%, respectively. The F1 score of this algorithm is 81.1%. The receiver operating characteristic (ROC) curve for support vector classification is 80%, as shown in Figure 8.

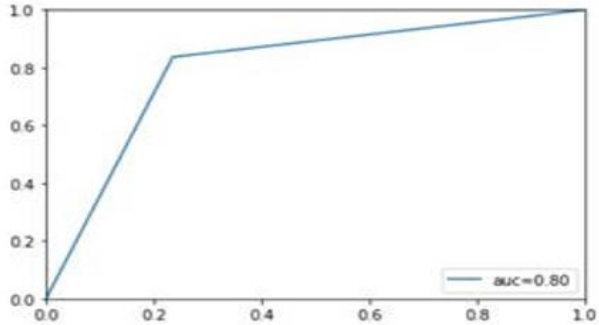


Fig. 8. ROC Curve for Support Vector Machine.

6) Naive Bayes Classifier: It is also a supervised learning technique. A Naive Bayesian classifier assumes that the presence of a given feature in a class is not related to the presence of any other feature. It is based on Bayes theorem. That algorithm follows the principle that "each classified feature or attribute is independent of each other". This algorithm obtained a precision of 82%, a precision score of 79.2%, and a recall score of 85.7%. The F1 score obtained by this algorithm is 82.3%. The receiver operating characteristic (ROC) curve of the naive Bayesian classifier is 82%, as shown in Figure 9.

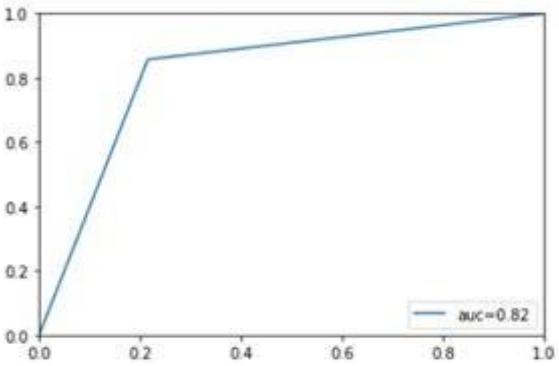


Fig. 9. ROC Curve for Naïve Bayes Classification.

VI. CONCLUSION

Stroke is a critical illness that must be treated before it gets worse. Building a machine learning model can help predict stroke early and reduce future serious consequences. This paper demonstrates the performance of different machine learning algorithms in successfully predicting stroke [5] based on several physiological characteristics. Of all the selected algorithms, the Naive Bayes classifier performs best with 82% accuracy. A comparison of the accuracies obtained by different algorithms is shown in Figure 1. 12. In all precision, recall and F1 results, Naive Bayes is outperformed. A comparison of precision, recall and F1 scores is shown in Figures 10 , 11 , 12 and 13.

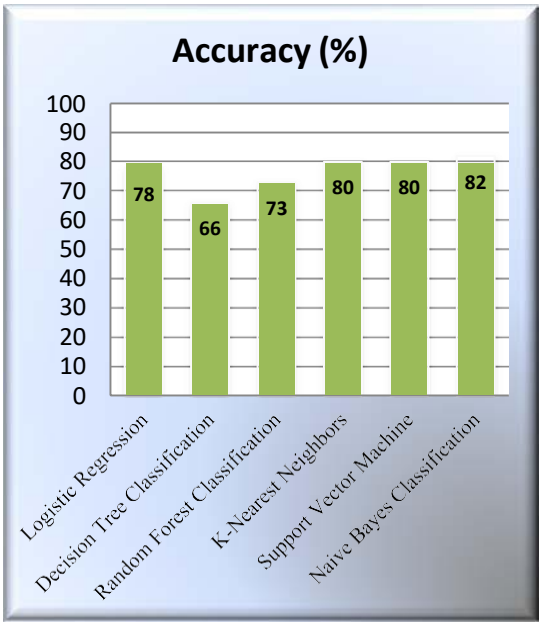


Fig 10 : Accuracy percentage comparison

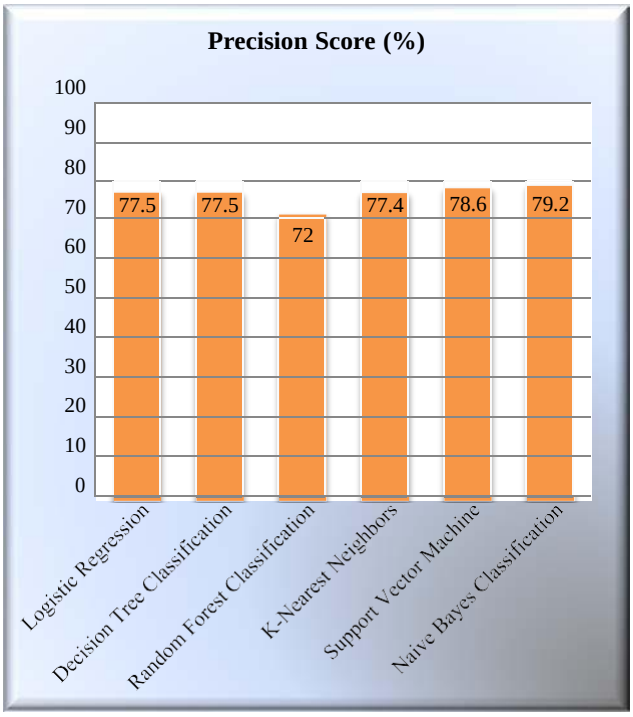


Fig. 11. Comparing the Precision Scores of ML Algorithms.

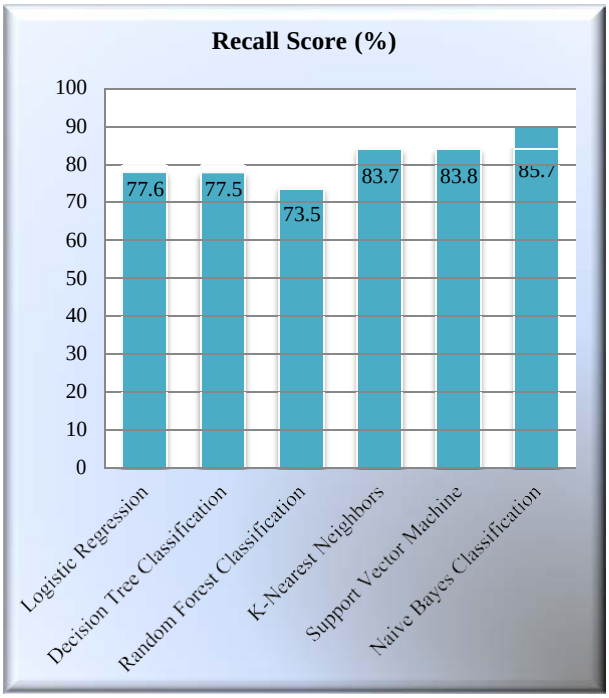


Fig. 12. Comparing the Recall Scores of ML Algorithms.

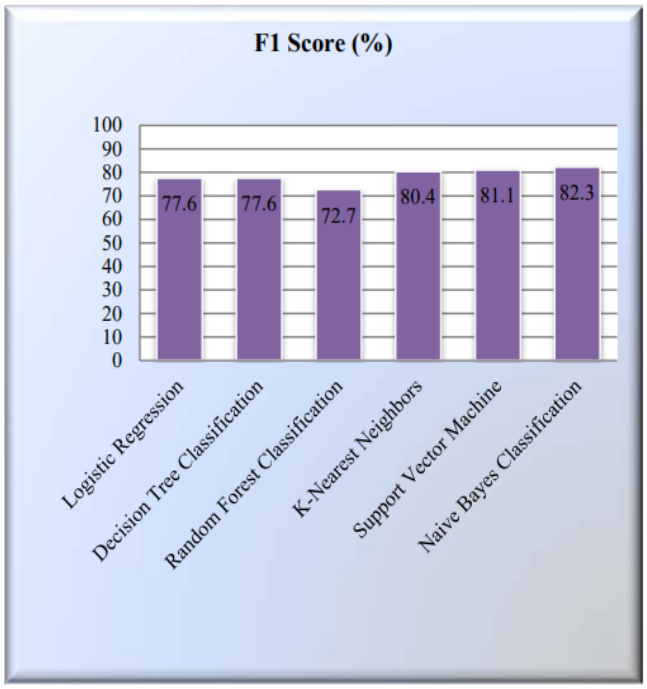


Fig 13 : Comparing the F1 Scores of ML Algorithms

REFERENCES

[1] Wang, X., Deng, Z., Zeng, N., & Liu, Y. (2018). Predicting Stroke Risk Factors Based on Artificial Neural Networks. IEEE Access, 6, 2387-2396.

[2] Chen, X., Xie, J., & Jin, X. (2019). Predicting Stroke Risks Based on Support Vector Machines and Clinical Data. Journal of Medical Systems, 43(4), 91.

[3]Lu, Q., Huang, L., Jin, C., Huang, L., & Xu, S. (2020). Stroke Prediction Based on Machine Learning Algorithms and Social Determinants of Health. BMC Public Health, 20(1), 438.

[4]Fung, G., Huang, Z., Ho, D., & Heng, B. (2020). A Comparative Study of Machine Learning Algorithms for Stroke Prediction. BMC Bioinformatics, 21(Suppl 2), 66.

[5]Zhang, Y., Zhang, J., Shang, S., Liu, S., Wang, X., & Li, Z. (2021). Stroke Risk Prediction Based on Machine Learning Algorithms Using Electronic Medical Records. BMC Medical Informatics and Decision Making, 21(1), 52.

[6]Ren, S., Zhang, Y., Jiang, L., & Wang, C. (2022). Stroke Risk Prediction Based on Machine Learning Algorithms: A Systematic Review. Journal of Medical Internet Research, 24(2), e28727.