Stroke Prediction using Machine Learning Technique

Iftehaz Newaz
Dept. of Computer Science and Engineering,
International Islamic University Chittagong
Chittagong, Bangladesh

***Abstract*—Disruption of blood supply in the brain is known as stroke. The rate of stroke has been gradually increasing, if one shows negligence in this matter then one might lose their life. Therefore it's one of the most important concerns in our life. If we can identify the stroke possibility in the early stage, we can reduce its rate of occurrence. Machine Learning algorithms play a vital role in the medical field for the betterment of human beings. In this paper we illustrate how machine learning algorithms help to identify stroke possibility and a comparison between different machine learning algorithms.**

***Index Terms*—Medical, Stroke, Machine Learning.**

I. Introduction

Stoke is one of the forefront concerns in today's world. From 2006 till 2023 the rate of stroke has increased about 50%. One out of four people have the probability to have a stroke in their lifetime. Stroke can occur when blood arteries get blocked. Depending on the position of neuronal damage, other symptoms, considering loss of consciousness, may occur [1]. hemorrhagic and thrombotic are the two vital types of stroke. A burst in cerebral blood vessels may lead to hemorrhagic stroke and one or more blockages in cerebral blood vessels may lead to thrombotic stroke. [1].

Numerous medical studies have been conducted in order to detect the possibility of having a stroke at an early stage. There are many risk factors such as hypertension, fatigue, cardiac disease etc. Lack of technical support, medical resources and the prediction of stroke is a matter of concern. The issue can be resolved by using machine learning techniques over potential dataset.

Here we consider a few machine learning methods such as Logistic Regression, K-Nearest Neighbor and Decision Tree which help to identify stroke at an early stage. The dataset we picked was an imbalance dataset. Thus, we had to do pre-processing to balance the datasets and for other matters.

II. Problem and The Dataset

Nowadays machine learning techniques can be beneficial to reduce the rate of stroke by detecting it at an early stage. Therefore, we have obtained the stroke dataset from Kaggle [2] which has 2510 upvotes. Dataset contains 5110 instances and 11 clinical features. Here 251 bmi data was missing.

TABLE I
Summary of Dataset

Features	Type	Range of Values
id	Integer	Arbitary
Gender	String (Male - 0, Female - 1)	0,1
Age	Float	0.08-82
Hypertension	Integer	0,1
Heart_disease	Integer	0,1
Ever_married	String (Yes - 0, No - 1)	0,1
Work_type	String (Private - 0, self-employed - 1)	0,1
Residence_type	String (Urban - 0, Rural - 1)	0,1
Avg_glucose_level	Float	55 - 291.0
BMI	Float	10.1 - 97.6
Smoking_status	String (Former - 0, Never Smoked - 1, Smokes - 2, Unknown - 3)	0, 1 ,2, 3
Stroke	Integer(Stroke 1 else 0)	0, 1

III. Methodology

We picked the classifiers which perform best in this domain, those are Logistic Regression, K-Nearest Neighbor, Decision Tree.

A. Logistic Regression

It's a classification method where it takes input X to train itself in order to predict binary outcomes.

$$log(p(X))/(1-p(X)) = \beta_0 + \beta_1.X \tag{1}$$

Here X is the independent variable and p(X) is the dependent variable. β_0 is the intercept and β_1 is the slope of coefficient [3], [4].

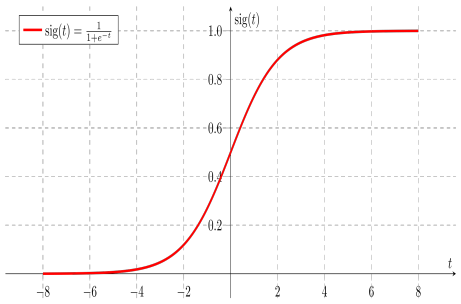

Fig. 1. Curve of Sigmoid Function

It uses a sigmoid function which takes any range of values and maps it into a range of 0 to 1. L2 regularization is used for penalty.

B. K-Nearest Neighbor

It's a supervised learning method. At first the train data will be fed to the classifier, From the test data, an instance will be fed to to check the distance between test data points and all the train instance's data points. It will pick k number of instances from the train set which has the shortest distance. The majority voting will be taken from those k instances and the classifier will output the result of majority vote.

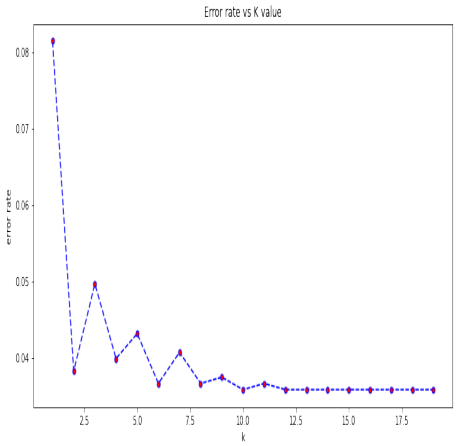

Fig. 2. K value vs Error rate

We examined the error rate for 1 to 20 k value and we can see that for first 10 k values the error rate fluctuates, after that it decreases and becomes flat. looking at the diagram we can pick the best k for KNN classifier. Thus we picked 11 as our k value

C. Decision Tree

It's a supervised learning technique where internal nodes show the attribute of a dataset, branches illustrate the decision rules and each leaf node depicts the outcome.

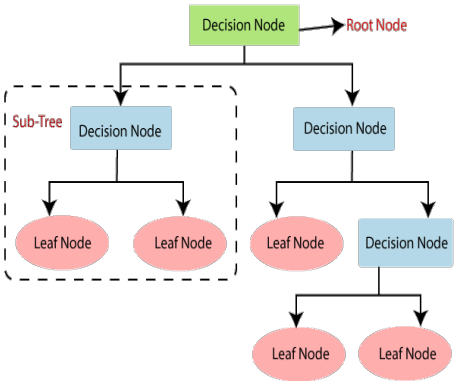

Fig. 3. Diagram of Decision Tree

Root Node: It is the starting point. It represents the entire dataset, which later gets divided into two or more homogeneous sub-tree/sets.
Leaf Node: Leaf nodes are end-node/output-node, and the tree cannot be segregated further after getting it.
Splitting: It is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
Branch/Sub Tree: A tree formed by splitting the tree.
Pruning: It is the process of removing the unwanted branches from the tree.
Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

D. Our approach

We load the dataset and do some preprocessing before feeding the data to classifiers. Firstly, we deal with all the missing values then we check whether the performance gets better with dropping the missing values or with replacing the missing values. The result showed that dropping the missing value yields better performance. Secondly, we know that classifiers can not deal with string type data. Therefore, we had to convert the data to numeric type. Lastly we have balanced the dataset as it was imbalanced.

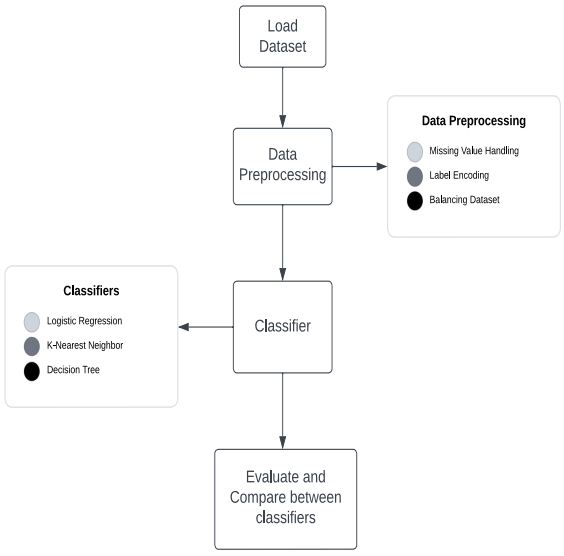

Fig. 4. Diagram of Our Approach

After preprocessing the dataset, we feed it to classifiers Logistic Regression, KNN, and Decision Tree. We then evaluate its performance and compare the performance between using SMOTE and before using SMOTE for balancing the datasets. The confusion Matrix and ROC Curve depicts which one performs better.

IV. Experimental Setup

A. Data Preprocessing

We had to preprocess the data to transform it into a shape, in order to feed the data into the classifier. Thus, we had to do missing value handling, label encoding, handling imbalance data.

Missing Value: There are 251 missing values in the BMI feature, we checked the performance by replacing those values by median and by dropping it. The experiment showed it performs better when we drop the missing values.

Label Encoding: There are few features consisting of string data type, as the classifier can't deal with string value. We had to convert it into a numeric type by using pd.factorize() which is a function of the pandas library. Here it identifies all distinct values and later can replace the string value with a numeric one.

Imbalance Data: Initially the data is imbalanced. Here the number of patients not having strokes is 4700 and having strokes is 209.

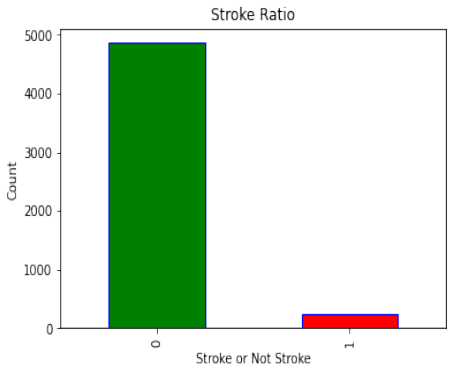

Fig. 5. Imabalance Ratio

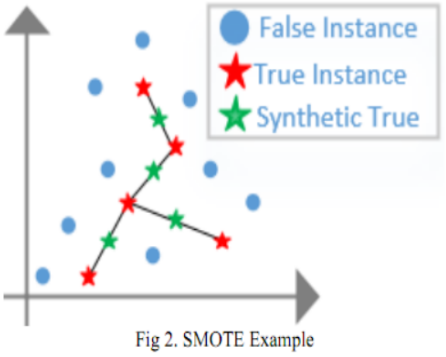

Fig. 6. How SMOTE works

We feed the data to the classifiers and shows that it performs poorly and can't detect those patients who have the potential

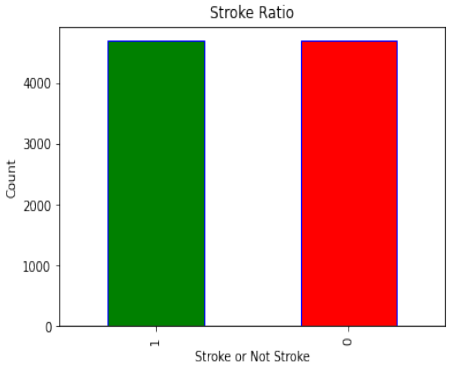

Fig. 7. Imbalance Ratio

to have a stroke. Therefore we used SMOTE to balance the dataset.

It's an oversampling method where new instances are obtained by synthesizing the existing sample. After using SMOTE the dataset is balanced. Now the dataset contains 4700 patients each having the stroke and not having stroke.

Normalization: In order to converge the classifier faster we used normalization technique to scale all the features between 0 to 1. We used a normalization function from sklearn library for this which uses L2 norms.

V. Results and Discussion

After balancing the dataset using SMOTE we did 5 fold cross validation of classifier and accuracies of the classifiers are:

TABLE II
Accuracy of Classifiers with 5 Fold CV Before using SMOTE

Classifier Name	Accuracy
Logistic Regression	95.7420%
K-Nearest Neighbor	95.7424%
Decision Tree	92.3200%

TABLE III
Accuracy of Classifiers with 5 Fold CV After using SMOTE

Classifier Name	Accuracy
Logistic Regression	70.7234%
K-Nearest Neighbor	84.9680%
Decision Tree	90.1596%

TABLE IV
Precision, Recall and F1 measures of Classifiers Before using SMOTE

Classifier Name	Precision	Recall	F1
Logistic Regression	0.48	0.50	0.49
K-Nearest Neighbor	0.48	0.50	0.49
Decision Tree	0.0.50	0.50	0.50

TABLE V
PRECISION, RECALL AND F1 MEASURES OF CLASSIFIERS AFTER USING SMOTE

Classifier Name	Precision	Recall	F1
Logistic Regression	0.70	0.70	0.70
K-Nearest Neighbor	0.86	0.84	0.84
Decision Tree	0.94	0.94	0.94

From table II, table III, table IV, and table V we can see that before balancing the dataset, the precision, recall and F1 score is poor for all the classifiers, the F1 score is 0.50 for Decision Tree. After balancing the dataset we can see that classifiers perform better, the precision, recall and F1 score is significant enough to consider. Among the three classifiers the F1 score is highest for Decision Tree which is 0.94.

TABLE VI
PARAMETERS

Classifier Name	Parameters
Logistic Regression	random_state = 0, iteration = 1000
K-Nearest Neighbor	neighbor = 11
Decision Tree	criterion='entropy', max_depth = 20

A. SMOTE & Confusion Matrix

The Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique for balancing the dataset when the difference between the amount of positive and negative label is immense. To use SMOTE there are few parameters to look at, we picked random_state parameter as 0, train test split random_state as 114 and test_size as 0.25. Before sampling classifier is not working properly, it is unable to detect class 1.

Before balancing the dataset the confusion matrix looks as:

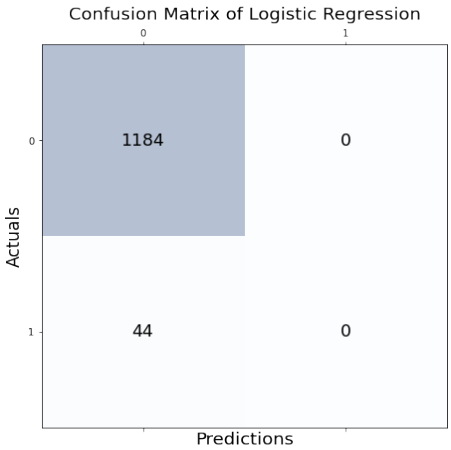

Fig. 8. Before SMOTE the Confusion Matrix

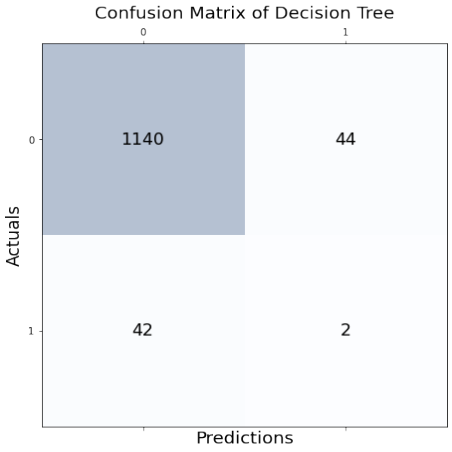

Fig. 9. Before SMOTE the Confusion Matrix

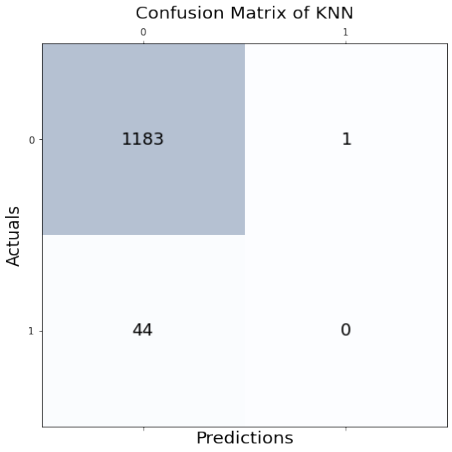

Fig. 10. Before SMOTE the Confusion Matrix

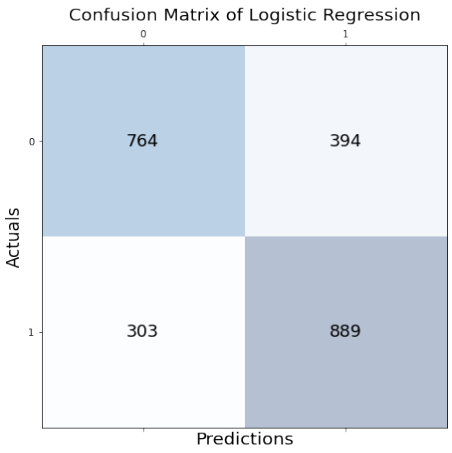

Fig. 11. After SMOTE the Confusion Matrix

After balancing the datasets, we see that the classifier works better to detect class 1 properly.

The analysis of confusion matrix shows that the Logistic Regression, K-Nearest Neighbor and Decision Tree classifiers are unable to predict class 1 if the dataset is imbalanced. After balancing the datasets the classifiers is able to detect label 1 and not being biased. The experiment illustrates that the decision tree performs better among Logistic Regression and

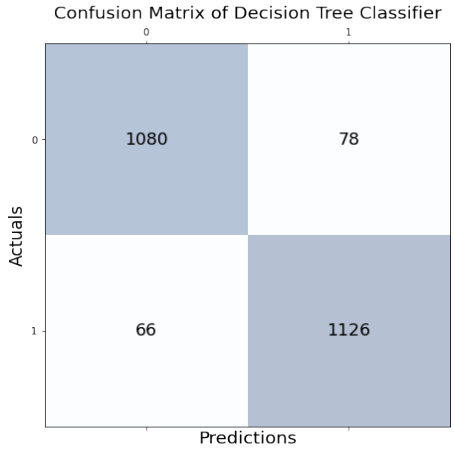

Fig. 12. After SMOTE the Confusion Matrix

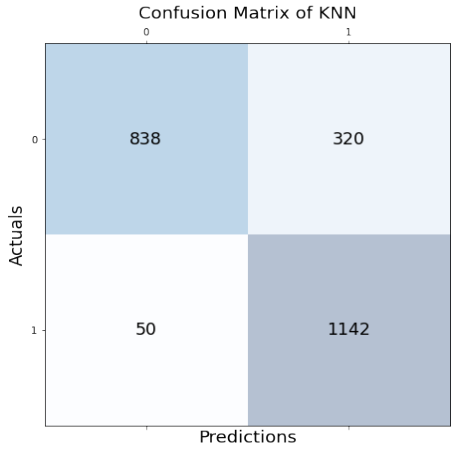

Fig. 13. After SMOTE the Confusion Matrix

K-Nearest Neighbor after the dataset is balanced.

B. ROC Curve

This plot is to show the behavior of two variable True Positive Rate and False Positive Rate for every threshold value between 0 to 1. True Positive Rate is also known as Recall/Sensitivity.

$TPR = Recall = Sensitivity = TP/(TP + FN)$
& $FPR = (1 - Specificity) = FP/(TN + FP)$

In the ROC Curve there's a green line starting from bottom left to top right, which indicates the classifier has the same predictive power as flipping a coin. The closer the blue line to the green line, the worse the classifier is. On the other hand, the farther the blue line to the green line, the better the classifier is.

The class distribution has no effect on the ROC curve. This makes it valuable for assessing classifiers that forecast uncommon occurrences like diseases or natural catastrophes.

After balancing the dataset, the ROC curve looks like:

ROC Curve depicts that among the three classifiers Logistic Regression, K-Nearest Neighbor and Decision Tree, Logistic

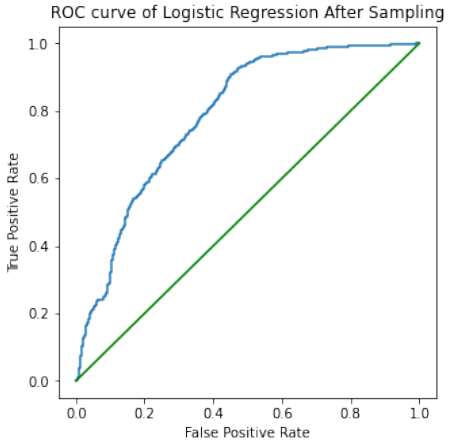

Fig. 14. After SMOTE the ROC Curve

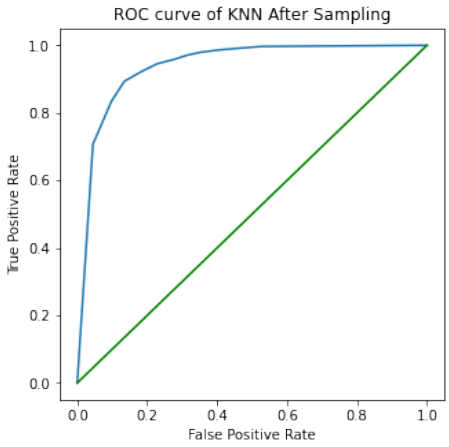

Fig. 15. After SMOTE the ROC Curve

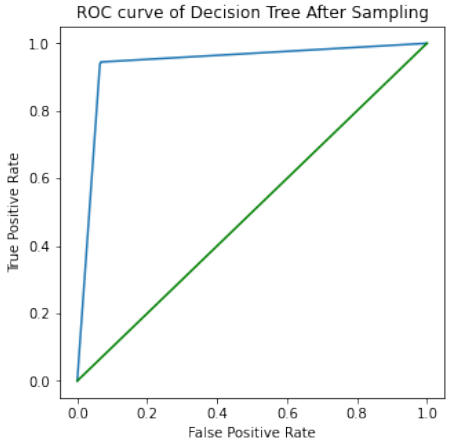

Fig. 16. After SMOTE the ROC Curve

Regression performs poorly and Decision Tree performs better.

VI. Conclusion

We can conclude that the occurrence of stroke can be reduced at an early stage with the help of machine learning techniques if we have proper dataset. It's a matter of concern nowadays Thus more research focus should be given in this domain. This paper depicts how machine learning can be utilized in order to reduce stroke rate. In future hybrid machine learning techniques with feature extraction and deep learning techniques can be implemented to extend the work.

References

[1] Azam, M. S., Habibullah, M., & Rana, H. K. (2020). Performance analysis of various machine learning approaches in stroke prediction. International Journal of Computer Applications, 175(21), 11-15.

[2] Kaggle. "Healthcare stroke Patients in Python"https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset"

[3] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley Sons.

[4] Couronné, R., Probst, P., & Boulesteix, A. L. (2018). Random forest versus logistic regression: a large-scale benchmark experiment. BMC bioinformatics, 19, 1-14.