Comparative study of various ML Algorithms to Predict Brain Stroke

ABSTRACT

In recent years, the number of people experiencing strokes has been rising rapidly worldwide, with the current figures nearly doubling in comparison to that two decades ago. Stroke remains a leading cause of mortality and morbidity worldwide, emphasising the critical need for accurate and timely prediction models to identify individuals at risk. Timely identification significantly mitigates the severity of stroke and in turn, lessens the mortality rate caused by stroke. This paper aims to introduce a model to predict the likelihood of stroke based on several factors including age, sex, cholesterol levels, BMI, smoking status, history of heart disease or attack, physical activity, diet, alcohol consumption, general health, physical health, motor abilities, high blood pressure, and diabetes. In this study, we compare different algorithms with and without random oversampling (ROS) to choose the best machine-learning techniques for the prediction of brain stroke. A brain stroke dataset from Kaggle was employed to build up the model. In the training and testing procedure, Classification and regression tree (CART), Support Vector Machine (SVM), Linear Logistic Regression (LLR), K-nearest neighbours (KNN), and Random Forest (RF) were utilised. The performance of each classifier with and without random oversampling has been estimated by various evaluation metrics such as accuracy score, precision score, sensitivity, specificity, recall score, F1 score, and ROC-AUC curve. The Random Forest Algorithm after ROS achieved the best performance among the algorithms with an accuracy of 98.38%.

1. INTRODUCTION

1.1 BRAIN STROKE

The brain, a complex organ, encased within the protective skull, facilitates intricate neural connections to coordinate movements, regulate vital functions, and process complex thoughts. Its adaptability, resilience, and susceptibility to disorders highlight the ongoing quest to understand and optimise brain health. Impaired brain health can significantly impact an individual's life, resulting in diminished cognitive and physical functioning. Physical symptoms like headaches, fatigue, and coordination issues can also arise. Severe cases can lead to debilitating conditions such as dementia or stroke, impacting quality of life and increasing dependency on others for daily tasks. In extreme situations, diminished brain health can even result in fatal outcomes due to complications or inability to perform vital bodily functions.

A stroke happens when blood flow to the brain is disrupted, depriving brain cells of oxygen and nutrients. This can occur due to either a blockage in an artery (ischemic stroke) or a rupture of a blood vessel (hemorrhagic stroke). Strokes are urgent medical situations as they can lead to serious complications like paralysis, speech problems, and cognitive issues. Risk factors for stroke include high blood pressure, smoking, diabetes, obesity, and family history. Immediate attention is crucial to minimise the impact of a stroke, and treatments may involve clot-busting drugs or surgery. Increasing awareness about stroke symptoms and preventive measures is vital for reducing its occurrence and severity. Early prediction of stroke may lessen stroke-related mortality.

According to the World Health Organization (WHO) [1], stroke is the second leading cause of death globally and the leading cause of disability.WHO estimated that globally, approximately 15 million people suffer from stroke each year. Additionally, around 5 million people die from stroke annually, and many survivors are left with long-term disabilities. Stroke can occur at any age, but the risk increases with age. While strokes are more common in older adults, particularly those over the age of 65, they can also affect younger individuals, including children and young adults.

According to our data set, from Kaggle [2], it was found that individuals in the age category of 65-69 are more likely to experience stroke.

1.2 INTRODUCTION TO ML ALGORITHMS

Machine learning, an integral part of artificial intelligence, empowers computers to learn from data without being explicitly programmed. It revolves around the development of algorithms that enable machines to recognize patterns within data and make decisions or predictions based on that analysis. These algorithms are the backbone of machine learning systems, allowing them to improve their performance over time as they are exposed to more data.

The importance of machine learning and its algorithms is multifaceted and far-reaching. Firstly, they have revolutionised various industries by enabling more accurate predictions and smarter decision-making processes. For instance, in healthcare, machine learning algorithms can analyse medical data to assist in diagnosing diseases, predicting patient outcomes, and personalising treatment plans. This has led to improved patient care and better health outcomes.

Machine learning algorithms play a major role in enhancing user experience and personalization in various of its applications. In e-commerce, recommendation systems powered by machine learning algorithms analyse user behaviour and preference to suggest products tailored to individual tastes, thus increasing sales and customer satisfaction.

The versatility of machine learning algorithms extends to their ability to automate repetitive tasks and optimise processes across various domains. In manufacturing, for example, predictive maintenance algorithms can analyse sensor data to anticipate equipment failures, minimising downtime and reducing maintenance costs. Similarly, in supply chain management, machine learning algorithms optimise inventory management, route planning, and demand forecasting, improving operational efficiency and reducing waste.

Overall, machine learning and its algorithms are indispensable tools in the modern technological landscape. Their ability to extract valuable insights from vast datasets, automate complex tasks, and drive innovation across various domains underscores their importance in shaping the present and future of society. As the volume and complexity of data continue to grow, the role of machine learning algorithms will only become more prominent, influencing nearly every aspect of our lives.

2. LITERATURE REVIEW

[3] The authors have implemented an improvised random forest algorithm. The improvised Random forest algorithm uses a decision tree, logistic regression, and SVM as the base classifiers. This approach enhanced the effectiveness of the random forest algorithm. The data was collected from the database of the National Institutes of Health Stroke Scale (NIHSS) for the study. Out of the total sample, 80% of subjects were used for training and 20% of subjects were used for testing. This helped improve the accuracy to 96.97% compared to the existing models.

[4] This research suggests that machine learning techniques outpowers deep neural networks for predicting brain stroke occurring at an early stage. The authors have proposed Random Forest as the most optimal algorithm for brain stroke prediction. The dataset was preprocessed using the Random Oversample (Ros approach), MinMaxScaler was used to scale the features to between -1 and 1 to normalise them, and the Principal component analysis (PCA) was utilised which chooses the minimum number of principal components to retain a variance of 95%. Out of the total sample, 80% of subjects were used for training, while 20% for testing. The Random Forest classifier achieved the highest classification accuracy at 99% among all machine learning classifiers.

[5] This study utilised a dataset consisting of medical, physiological, and environmental tests related to strokes, focusing on assessing the effectiveness of machine learning, deep learning, and a hybrid technique for analysing Magnetic Resonance Imaging (MRI) data concerning cerebral haemorrhage. New features, namely diabetes and obesity, were derived based on corresponding values. The t-distributed Stochastic Neighbour Embedding algorithm and Recursive Feature Elimination(RFE) were applied to favorably modify the dataset. Various classification algorithms, including Support Vector Machine (SVM), K Nearest Neighbours (KNN), Decision Tree, Random Forest, and Multilayer Perceptron, were applied, with Random Forest outperforming others, achieving an outstanding 99% overall accuracy. The MRI image dataset was assessed using the AlexNet model and the hybrid AlexNet + SVM technique. The hybrid model demonstrated superior performance to AlexNet, achieving 99.9% accuracy, 100% sensitivity, 99.80% specificity, and a 99.86% Area Under the Curve (AUC).

[6] This research focuses on using machine learning algorithms to identify risk variables of stroke. Efficient data collection, data pre-processing, and data transformation methods have been applied to the dataset. Three different classifiers were used- Random forest(RF) Support Vector Machine (SVM), and Decision Tree (DT). To assess their performance, different

parameters like accuracy, sensitivity (SEN), error rate, false-positive rate (FPR), false-negative rate (FNR), root mean square error, and log loss. The random forest tree classifier showed the best results with an accuracy of 95.30%.

[7] This study aimed to utilise diverse machine learning (ML) techniques to predict 90-day stroke outcomes, leveraging a nationwide disease registry. The Taiwan Stroke Registry (TSR) provided data from stroke patients since 2006. Three established ML models (support vector machine, random forest, and artificial neural network), alongside a hybrid artificial neural network, were employed and assessed using a 10-time repeated hold-out method with 10-fold cross-validation. ML techniques demonstrated AUCs exceeding 0.94 for both ischemic and hemorrhagic stroke using pre-admission and inpatient data. Incorporating follow-up data further enhanced prediction with a 0.97 AUC. Among 206 clinical variables screened, 17 important features were identified from the ischemic stroke dataset and 22 from the hemorrhagic stroke dataset, preserving performance. This research demonstrated that machine learning techniques trained on extensive, cross-regional registry datasets could accurately predict functional outcomes following a stroke.

[8] This study focuses on identifying the most important factors for stroke prediction by examining the diverse risk factors found in Electronic Health Records (EHR) of patients. Their findings suggest that age, heart disease, average glucose level, and hypertension, independently, are the most important factors for detecting stroke in patients. Furthermore, they found that no two features are highly correlated to each other. They used a dimensionality reduction technique to transform the high-dimensional feature space into a low-dimensional feature space. This study compares the performance of 3 classification approaches: Convolutional Neural Network (CNN), Decision tree (DT), and Random forest (RF). Additionally, a random downsampling technique was used to reduce the adverse impact caused by the unbalanced nature of the dataset. The ratio of the number of training observations to testing observations is 70:30. They found that the best-performing algorithm is the Convolutional Neural Network(CNN) with an accuracy rate of 78% and a miss rate of 19%.

[9] This article outlines the implementation of six distinct machine-learning classification algorithms to predict stroke. The dataset is sourced from Kaggle. The dataset is preprocessed by removing null values, and label encoding, and the dataset is balanced by an undersampling method. This preprocessed data is split in such a way that 80% is training data and 20% is testing data. The training data is then fitted into six machine learning algorithms: Decision Tree Classification, Logistic Regression, Random Forest Classification, K-Nearest Neighbors Classification, Support Vector Machine, and Naïve Bayes Classification. Naïve Bayes emerged as the top-performing algorithm for this task, achieving an accuracy of approximately 82%.

[10] In this research endeavour, machine learning (ML) techniques were leveraged to develop and assess multiple models aimed at crafting a resilient framework for predicting the long-term risk of stroke occurrence. The research utilised a dataset sourced from Kaggle, focusing specifically on participants aged over 18 years old. The dataset comprised 3254 participants,

with 10 features serving as inputs to the machine learning models and 1 feature representing the target class. The primary contribution of this study lies in the implementation of a stacking method that demonstrates high performance, validated through diverse metrics including AUC, precision, recall, F-measure, and accuracy. The stacking classification method demonstrates superior performance compared to other methods, achieving an AUC of 98.9%, F-measure, precision, and recall of 97.4%, and an accuracy of 98%.

[11] In this study, the authors use three different data selection methods (without data resampling, with data imputation, and with data resampling). Four machine learning classifiers, namely: naïve Bayes, BayesNet, J48(Java implementation of C4.5 algorithm), and random forest. Recall or Sensitivity, specificity, positive predictive value(PPV), negative predictive value(NPV), accuracy, and area under the curve(AUC) are the six evaluation measures used to assess performance. The data sets were obtained from the National Health and Nutrition Examination Survey. They concluded that the data resampling approach performed the best out of the data selection techniques. They also found that the random forest algorithm was the best-performing algorithm with an accuracy score of 0.96, a sensitivity score of 0.97, a specificity score of 0.96, a positive prediction value of 0.75, a negative prediction value of 0.99, and an area under the curve of 0.97 when all the attributed were used.

[12] In this study, the authors run CT scan images through different machine learning algorithms. The CT scan images of stroke patients were preprocessed to reduce noise and increase the quality of the images. They further used machine learning algorithms to classify the stroke into two types, namely Ischemic, and haemorrhage. The machine learning algorithms are K-Nearest Neighbors, Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Multi-layer Perceptron (MLP-NN), Deep Learning and Support Vector Machine. Amongst these algorithms, they found that the Random tree achieved an accuracy of 95.97% which was the highest, it also had a precision percentage of 94.39%, a recall value of 96.12%, and an f1-Measure of 95.39%.

3. MATERIALS AND METHODS

3.1 DESCRIPTION OF DATASET

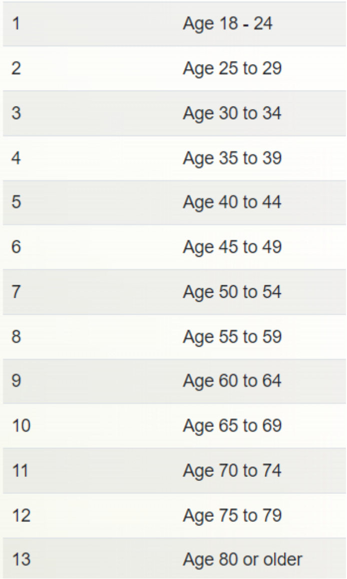
The research was carried out based on the dataset obtained from the Kaggle website [2]. The description of the dataset is given in Table 1. The output column consists of values “1” and “0”. If the column had an output “1”, then a chance of having a stroke was detected, and if it had a “0”, then a chance of not having a stroke was detected.

Table 1: Description of the dataset used

Attribute Name	Description
Sex	1=male ; 0= female
HighChol	0 = no high cholesterol 1 = high cholesterol
CholCheck	0 = no cholesterol check in 5 years 1 = Yes cholesterol check in 5 years
BMI	Body Mass Index
Smoking status	Have you smoked at least 100 cigarettes in your entire life? 0 = no ; 1 = yes
HeartDiseaseorAttack	Coronary heart disease (CHD) or Myocardial infarction (MI)? 0 = no ; 1 = yes
PhysActivity	Physical activity in the past 30 days - not including a job? 0 = no ; 1 = yes
Fruits	Consume Fruit 1 or more times per day? 0 = no ; 1 = yes
Veggies	Consume Vegetables 1 or more times per day? 0 = no ; 1 = yes
HvyAlcoholConsump	(adult men >=14 drinks per week and adult women>=7 drinks per week)? 0 = no ; 1 = yes
GenHlth	Would you say that in general your health is: scale 1-5 1 = excellent 2 = very good 3 = good 4 = fair 5 = poor
MentHlth	Days of poor mental health scale 1-30 days
PhysHlth	Physical illness or injury days in past 30 days scale 1-30

DiffWalk	Do you have serious difficulty walking or climbing stairs? 0 = no ; 1 = yes
HighBP	0 = no high BP ; 1 = high BP
Diabetes	0 = no diabetes ; 1 = diabetes
Stroke	0=no stroke;1=had stroke

Table 2: Age category



1	Age 18 - 24
2	Age 25 to 29
3	Age 30 to 34
4	Age 35 to 39
5	Age 40 to 44
6	Age 45 to 49
7	Age 50 to 54
8	Age 55 to 59
9	Age 60 to 64
10	Age 65 to 69
11	Age 70 to 74
12	Age 75 to 79
13	Age 80 or older

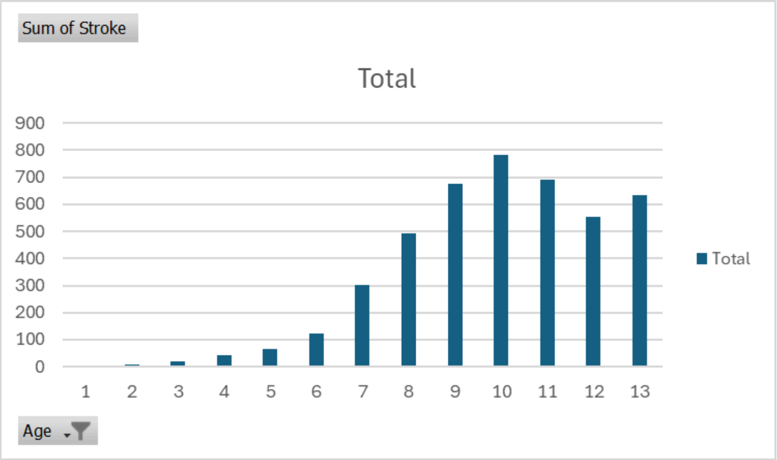


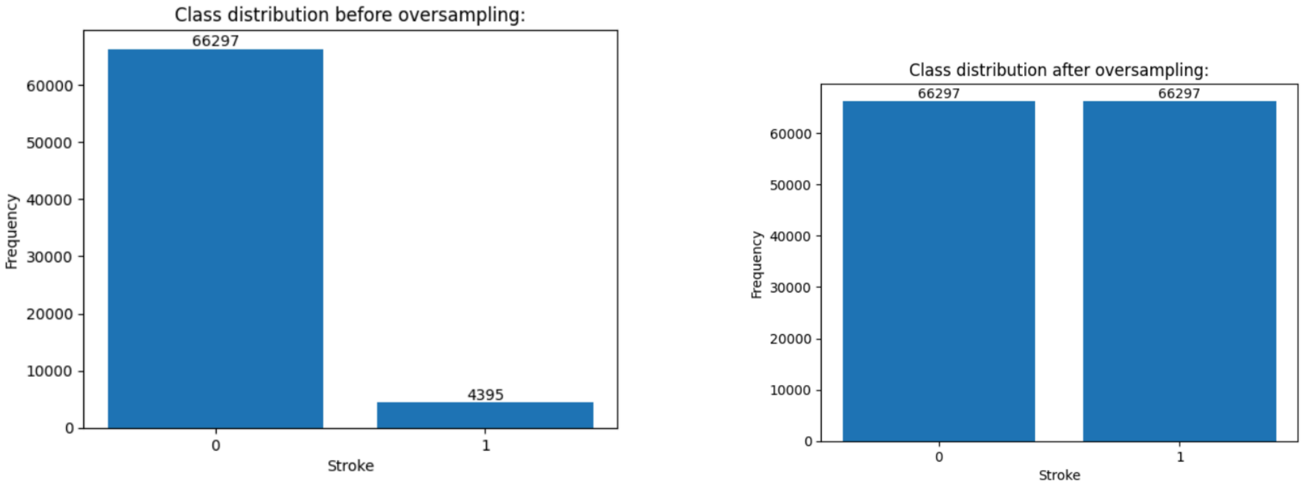
Fig 1

3.2 DATA PRE-PROCESSING

Data preprocessing ensures data quality, enhances model performance, handles missing values, normalised data, encodes categorical variables, reduces noise, and prevents overfitting in machine learning and analysis, improving overall accuracy and reliability.

Random oversampling addresses the class imbalance in classification. When one class is underrepresented, models struggle to classify minority instances, causing biased predictions. By duplicating minority class samples, random oversampling rebalances data, improving model performance.

The change of imbalance in the dataset before and after Random Over Sampling is shown in Fig 1(A) and Fig 1(B).



(A): Count of stroke before oversampling

(B): Count of stroke after oversampling

Fig 2

Oversampling is preferred over undersampling because it retains information, enhances model performance, reduces overfitting, and preserves original class distribution, offering flexibility in sampling techniques to address class imbalance effectively in classification tasks.

Feature scaling is a preprocessing step used to standardise or normalise the range of features in a dataset. However, for this dataset, we found that feature scaling did not make a difference in the evaluation matrices.

The dataset was divided into 70% for training and 30% for testing.

3.3 CLASSIFICATION OF ALGORITHMS

3.3.1 CART(Classification and Regression Tree) decision tree algorithm

Cart, short for Classification and Regression Trees, represents a machine learning algorithm employed for tasks involving both classification and regression. This algorithm operates by iteratively dividing the input space (also known as feature space) into smaller segments, guided by the values of input features, with the aim of making predictions. [13]

Algorithm 1
CART(Classification and Regression Tree) decision tree algorithm

1. Procedure CART (Classification and Regression Tree) decision tree algorithm
2. Input : csv data file with features ‘Age’, ‘Sex’, ’HighChol’, ‘CholCheck’, ’BMI’, ’Smoker’, ‘HeartDiseaseorAttack’, ‘PhysActivity’, ‘Fruits’, ‘Veggies’, ‘HvyAlcoholConsump’, ‘GenHlth’, ‘MentHlth’, ‘PhysHlth’, ‘DiffWalk’, ‘HighBP’, ‘Diabetes’ and output feature ‘stroke’
3. Output : predicting if a patient has a stroke or not based on the data sets
4. Import pandas, import tree from sklearn, import accuracy_score, precision_score, recall_score, f1_score from sklearn.metrics, import DecisionTreeClassifier from sklearn.tree, import train_test_split from sklearn.model_selection
5. Read csv file using pandas
6. Split the data set in ratio 7:3 as a training set and testing set respectively
7. Create an instance of DecisionTreeClassifier()
8. Fit the decision tree algorithm to training set
9. Predict the stroke values using test data
10. Display accuracy score, precision score, f1 score, recall score, specificity and sensitivity using stroke values of test data set and predicted stroke values
11. Plot decision tree using tree.plot_tree
12. End procedure

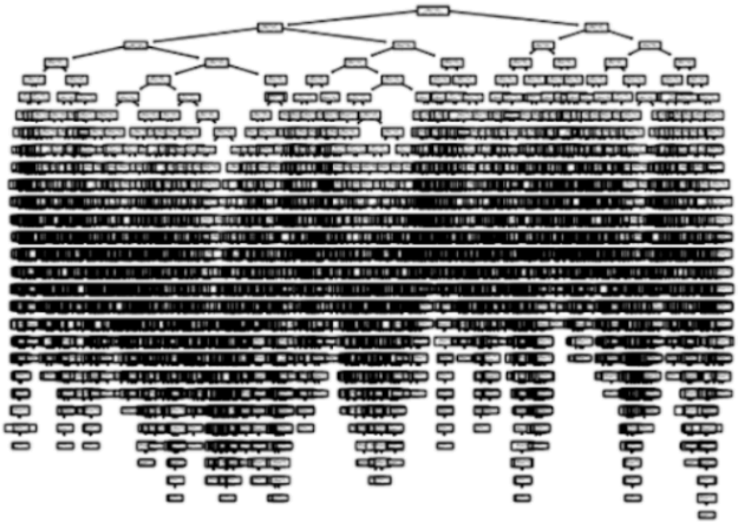


Fig 3: Decision Tree Regression Graph

3.3.2. Support Vector Machine(SVM)

Support Vector Machine (SVM) is a supervised learning technique utilised for classification and regression assignments. Its principal aim is to identify the most suitable hyperplane that distinguishes between different classes within the feature space. SVM achieves this by transforming input data into a higher-dimensional feature space, where it determines the ideal hyperplane that maximises the margin between classes. [14]

Algorithm 2
Support Vector Machine(SVM)

1. Procedure Support Vector Machine(SVM) algorithm
2. Input:csv data file with features ‘Age’, ‘Sex’, ’HighChol’, ‘CholCheck’, ’BMI’, ’Smoker’, ‘HeartDiseaseorAttack’, ‘PhysActivity’, ‘Fruits’, ‘Veggies’, ‘HvyAlcoholConsump’, ‘GenHlth’, ‘MentHlth’, ‘PhysHlth’, ‘DiffWalk’, ‘HighBP’, ‘Diabetes’ and output feature ‘stroke’
3. Output : predicting if a patient has a stroke or not based on the data sets
4. Import pandas, import SVC from sklearn.svm, import accuracy_score, precision_score, recall_score,f1_score from sklearn.metrics,import train_test_split from sklearn.model_selection
5. Read csv file using pandas
6. Split the data set in ratio 7:3 as a training set and testing set respectively
7. Create an instance of SVC()
8. Fit the svm model to training set
9. Predict the stroke values using test data
10. Display accuracy score, precision score, f1 score, recall score, specificity and sensitivity using stroke values of test data set and predicted stroke values
11. End procedure

3.3.3. Logistic regression algorithm

Logistic regression is a statistical method for binary classification. It models the probability of the outcome using a logistic function. By fitting the data to a linear equation, it estimates the probability of a binary outcome, making it widely used in fields such as medicine and marketing for prediction tasks. [15]

Algorithm 3
Logistic regression algorithm

1. Procedure Logistic regression algorithm
2. Input:csv data file with features ‘Age’, ‘Sex’, ’HighChol’, ‘CholCheck’, ’BMI’, ’Smoker’, ‘HeartDiseaseorAttack’, ‘PhysActivity’, ‘Fruits’, ‘Veggies’, ‘HvyAlcoholConsump’, ‘GenHlth’, ‘MentHlth’, ‘PhysHlth’, ‘DiffWalk’, ‘HighBP’, ‘Diabetes’ and output feature ‘stroke’
3. Output: predicting if a patient has a stroke based on the data sets

4. Import pandas, import train_test_split from sklearn.model_selection, import accuracy_score, precision_score, recall_score, f1_score from sklearn.metrics, import LogisticRegression from sklearn.linear_model

5. Read CSV file using pandas

6. Split the data set in ratio 7:3 as a training set and testing set respectively

7. Define an instance LogisticRegression()

8. Fit the Logistic Regression algorithm to the training set

9. Predict the stroke values using test data

10. Display accuracy score, precision score, f1 score, recall score,specificity and sensitivity using stroke values of test data set and predicted stroke values

11. End procedure

3.3.4. KNN Algorithm

K-Nearest Neighbors (KNN) is an uncomplicated yet potent supervised learning method applied in classification and regression. It predicts the class of a data point by discerning the predominant class among its K nearest neighbours within the feature space, determined by a predefined distance measure. [16]

Algorithm 4
KNN Algorithm

1. Procedure: KNN algorithm

2. Input:csv data file with features ‘Age’, ‘Sex’, ’HighChol’, ‘CholCheck’, ’BMI’, ’Smoker’, ‘HeartDiseaseorAttack’, ‘PhysActivity’, ‘Fruits’, ‘Veggies’, ‘HvyAlcoholConsump’, ‘GenHlth’, ‘MentHlth’, ‘PhysHlth’, ‘DiffWalk’, ‘HighBP’, ‘Diabetes’ and output feature ‘stroke’

3. Output: predicting if a patient has a stroke based on the data sets

4. Import pandas,import train_test_split from sklearn.model_selection,import accuracy_score, precision_score, recall_score, f1_score from sklearn.metrics,import matplotlib.pyplot as plt, import KNeighborsClassifier from sklearn.neighbors

5. Read CSV file using pandas

6. Split the data set in ratio 7:3 as a training set and testing set respectively.

7. Define a for loop from 2 to 21

8. Initialise a training list and test list to append training and test scores

9. Define an instance KneighborsClassifier()

10. Fit the KNNalgorithm to the training set

11. Predict the stroke values using test data

12. Display accuracy score, precision score, f1 score, recall score,specificity and sensitivity using stroke values of test data set and predicted stroke values

13. End procedure

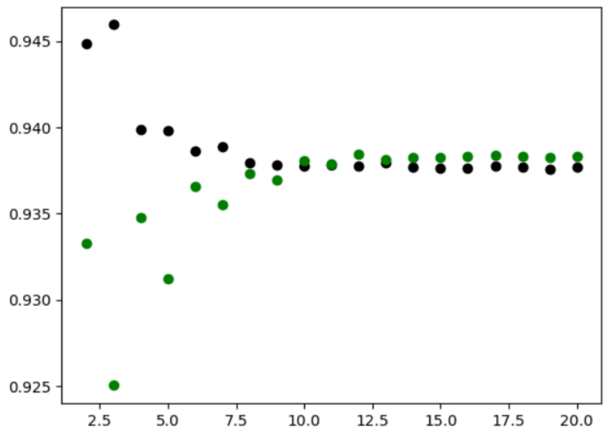


Fig 4: K-value vs Score

🟢 Test Score ● Training Score

From Fig 4 it can be interpreted that the optimum value of k will be around 12

3.3.5. Random tree algorithm

The Random Tree algorithm creates a machine-learning model by forming a collection of decision trees. Each tree is developed using a random sample of features and data points. This randomization aids in mitigating overfitting and enhancing the model's ability to generalise in classification and regression scenarios. [17]

Algorithm 5
Random tree algorithm

1. Procedure: Random tree algorithm
2. Input:csv data file with features ‘Age’, ‘Sex’, ’HighChol’, ‘CholCheck’, ’BMI’, ’Smoker’, ‘HeartDiseaseorAttack’, ‘PhysActivity’, ‘Fruits’, ‘Veggies’, ‘HvyAlcoholConsump’, ‘GenHlth’, ‘MentHlth’, ‘PhysHlth’, ‘DiffWalk’, ‘HighBP’, ‘Diabetes’ and output feature ‘stroke’
3. Output : predicting if a patient has a stroke or not based on the data sets
4. Import pandas, import accuracy_score, precision_score, recall_score, f1_score from sklearn.metrics, import train_test_split from sklearn.model_selection, import RandomForestClassifier from sklearn.ensemble
5. Read csv file using pandas
6. Split the data set in ratio 7:3 as a training set and testing set respectively.
7. Create an instance of RandomForestClassifier
8. Fit the random tree algorithm to training set
9. Predict the stroke values using test data
10. Display accuracy score, precision score, f1 score, recall score,specificity and sensitivity using stroke values of test data set and predicted stroke values
11. End procedure

3.4 PERFORMANCE MATRIX

Python's scikit-learn library provides various functions and tools for analysing the performance of machine learning algorithms. The description of some of the common functions used is described below.

3.4.1 Confusion matrix

The confusion matrix represents the performance of a model. It returns a 2-D array containing a number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

3.4.2 Accuracy Score

The accuracy score serves as a metric for assessing the performance of ML algorithm models. It is equal to the ratio of the count of correct predictions to the total number of predictions, that is the percentage of instances predicted correctly.

The accuracy of a model can be represented using a confusion matrix as,

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

3.4.3 Precision Score

Precision is a performance metric used to evaluate ML models. It aids in comprehending the percentage of correctly identified positive instances out of the total number of instances predicted as positive.

The precision of a model can be represented using a confusion matrix as,

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

3.4.4 Recall Score or Sensitivity

Recall score or sensitivity is a performance metric used to evaluate the ML model, where it quantifies the proportion of correctly predicted positive instances out of all positive instances predicted.

Recall of a model can be represented using a confusion matrix as,

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

When dealing with imbalanced data, the recall_score() function with the parameter average="weighted" is employed to compute the recall score. This setting assigns greater weight to classes with larger instances, effectively compensating for class imbalance.

When the data is balanced and oversampling is performed using Random Over-Sampling (ROS), the recall_score() function is used without specifying the average parameter.

3.4.5 F1 score

F1 score is a performance metric used to evaluate ML models, where it combines both precision and recall scores. It is mainly used when the data is imbalanced or when false positive and false negative values are important terms in analysing the data.

The F1 score of a model can be represented using a confusion matrix as,

$$\text{F1} = \frac{2 \text{ x Precision x Recall}}{\text{Precision+Recall}}$$

3.4.6 Specificity

Specificity is a performance metric used to evaluate ML models. It is the ratio of correctly predicted negative instances out of all actual negative instances.

The specificity of a model can be represented using a confusion matrix as,

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}$$

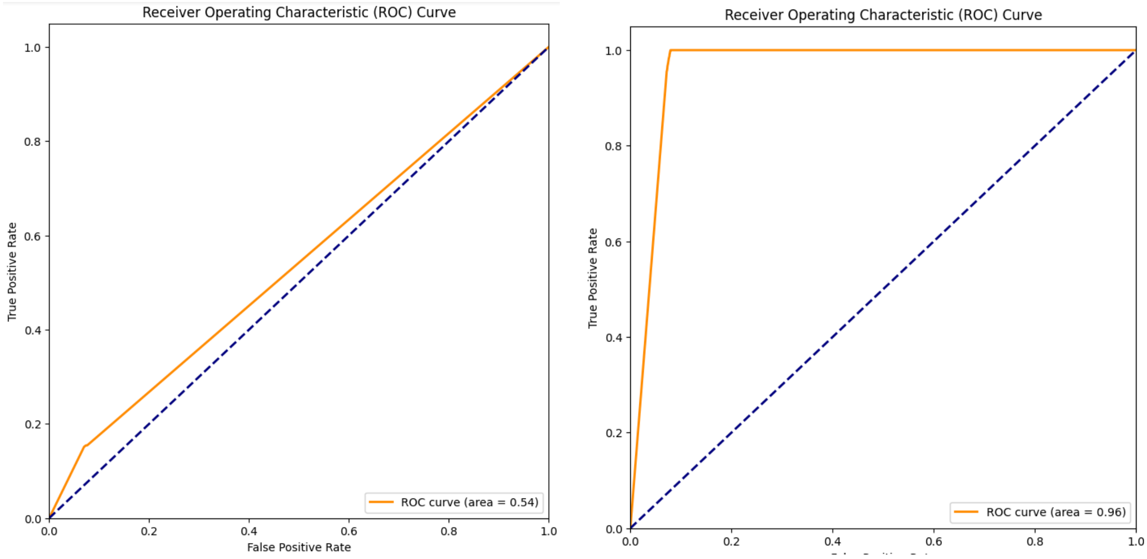
3.4.7 ROC-AUC Curve

The ROC-AUC curve assesses binary classifiers by plotting the True Positive Rate against the False Positive Rate. AUC measures overall performance, with 1 indicating a strong ability to distinguish between positive and negative classes, while 0.5 represents random guessing. A score below 0.5 suggests the model predicts more false positives than true ones.

4. RESULTS AND ANALYSIS

4.1 GRAPHS WITH DESCRIPTIONS

4.1.1 Classification and Regression Trees (CART)

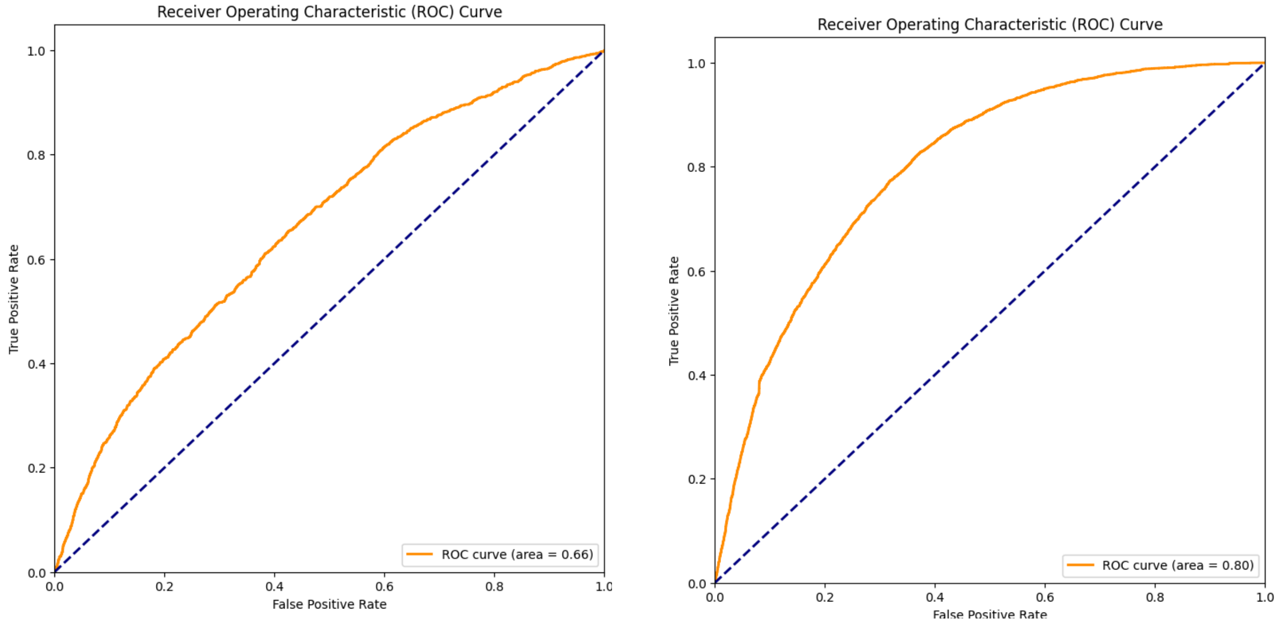


(A)Before random oversampling:
Low ROC-AUC Score before oversampling

(B)After random oversampling:
High ROC-AUC Score after oversampling

Fig 5:ROC Curve of CART algorithm

4.1.2 Support Vector Machine(SVM)

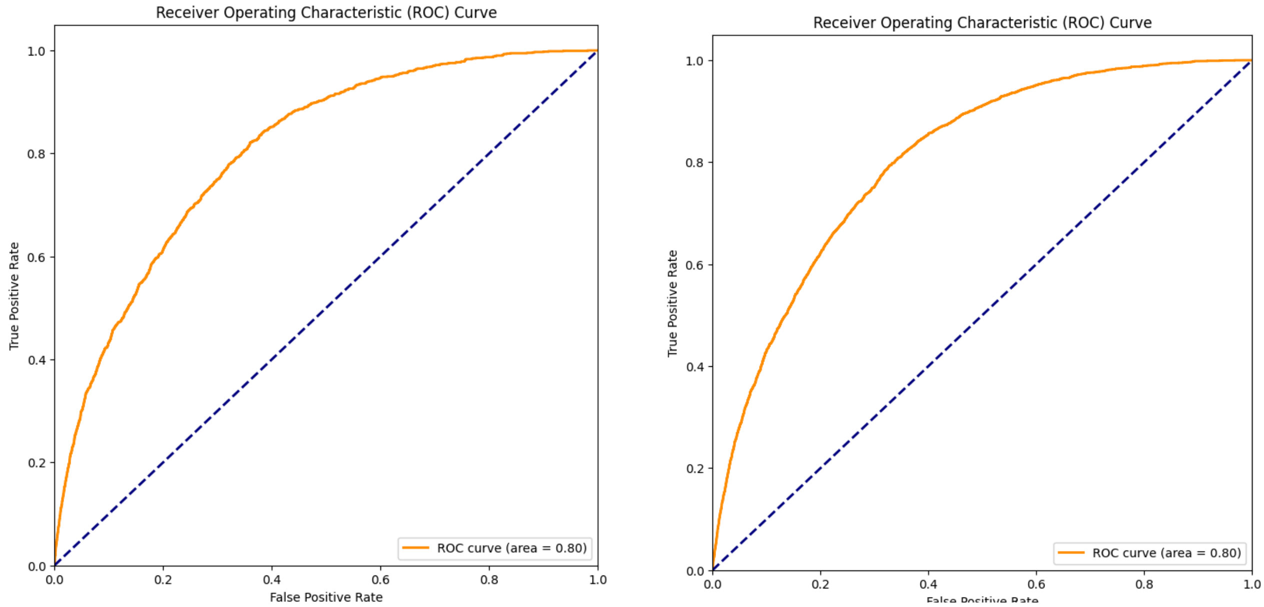


(A)Before random oversampling:
Low ROC-AUC Score before oversampling

(B)After random oversampling
High ROC-AUC Score after oversampling

Fig 6:ROC Curve of Support Vector Machine(SVM)

4.1.3 Logistic regression

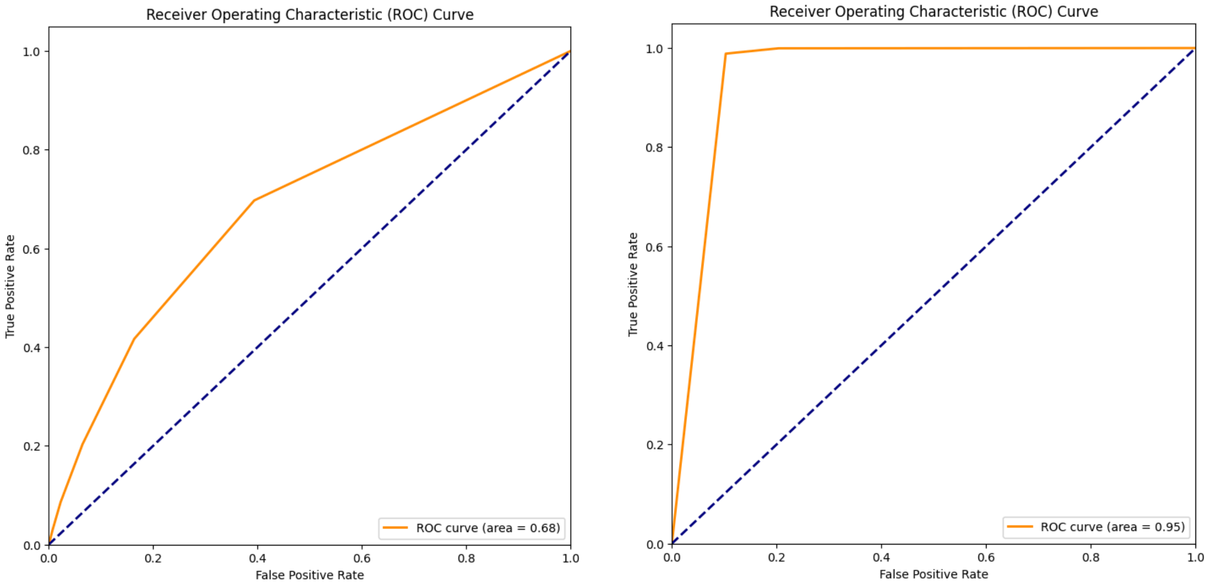


(A)Before random oversampling
High ROC-AUC Score before oversampling

(B)After random oversampling
High ROC-AUC Score after oversampling

Fig 7:ROC Curve of Logistic regression

4.1.4 K-Nearest Neighbour (KNN)

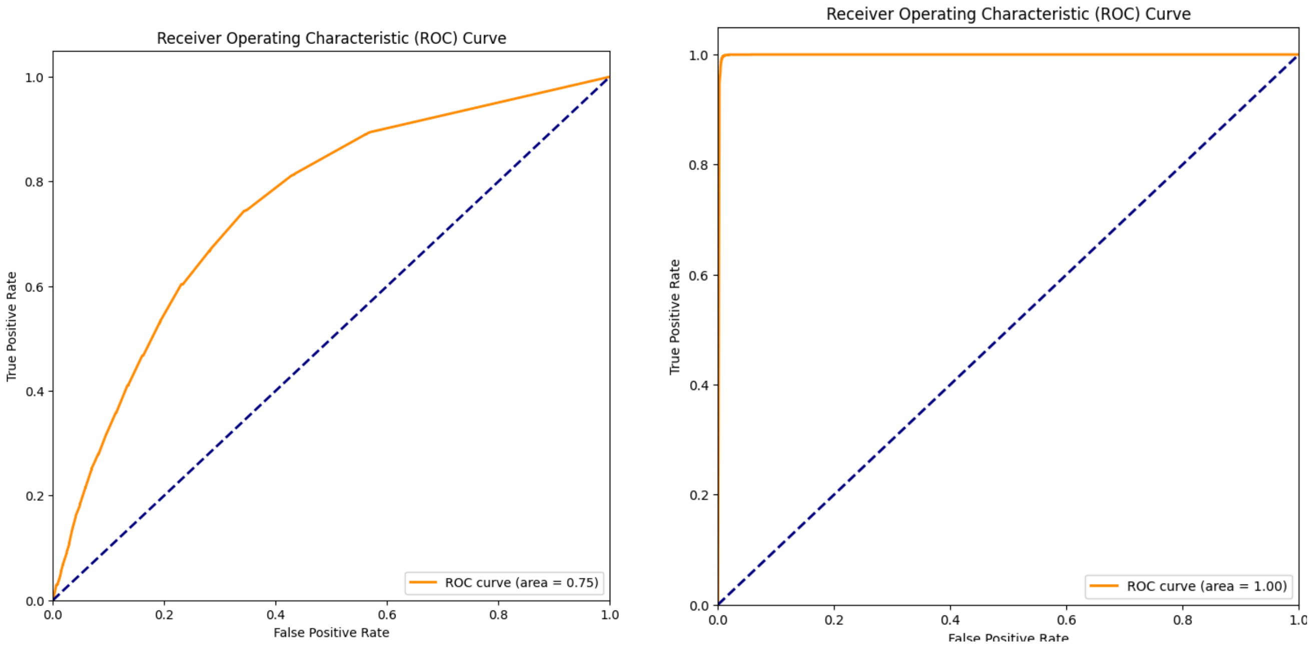


(A)Before random oversampling
Low ROC-AUC Score before oversampling

(B)After random oversampling
High ROC-AUC Score after oversampling

Fig 8: ROC Curve of KNN

4.1.5 Random Tree



(A)Before random oversampling
Low ROC-AUC Score before oversampling

(B)After random oversampling
High ROC-AUC Score after oversampling

Fig 9:ROC Curve of Random Tree

4.2 PERFORMANCE MATRIX ANALYSIS

Table 4
Performance metric scores of each algorithm before Random Oversampling

Name	Accuracy(%)	Precision (%)	Recall (%)	F1 (%)	Specificity (%)
CART	88.20	89.29	88.20	88.73	93.01
SVM	93.83	88.04	93.83	90.84	100
LINEAR LOGISTIC	93.81	90.67	93.81	90.91	99.94
KNN	93.83	91.14	93.83	90.87	99.98
RANDOM TREE	93.46	89.67	93.46	90.99	99.39

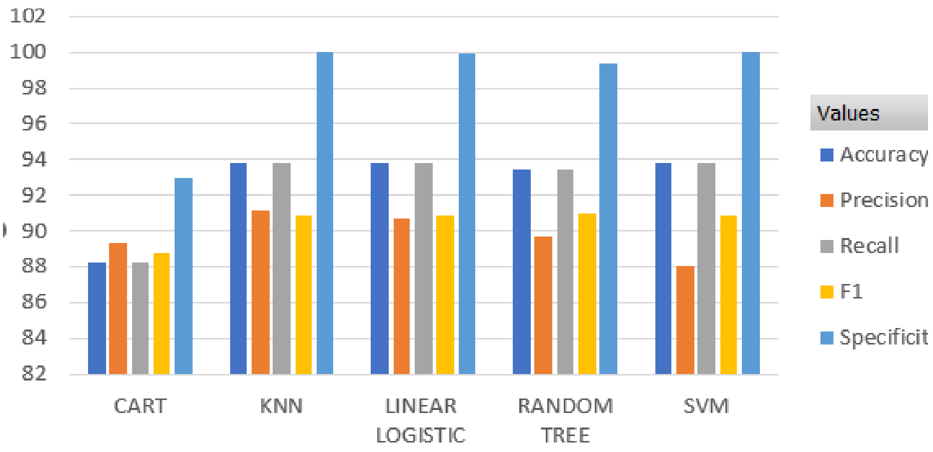


Fig 10: Graphical representation of the performance metric scores of each algorithm before Random Oversampling

The linear logistic classification algorithm stands out with the highest performance among all other algorithms before data oversampling, that is when the data is imbalanced. The performance metrics for the linear logistic classification algorithm, based on Table 4, are as follows: accuracy score of 93.81%, recall score of 93.81%, and F1 score of 90.91%. Fig 7(A) shows the ROC graph of a linear logistic classification algorithm which has an ROC-AUC score of 0.8. This indicates that the algorithm accurately classifies 93.81% of instances and effectively identifies both positive and negative instances within the dataset.

From Table 4 it can be concluded that SVM and KNN(at k=12) algorithms have the highest accuracy score. Comparing Fig 6(A), Fig 8(A), and Fig 9(A), SVM and KNN algorithms have lower ROC-AUC than random tree classification algorithms. This means that SVM and KNN algorithms are better at overall prediction but less effective at distinguishing between positive and negative instances. On the other hand, the random tree classification algorithm is a bit more effective at classifying positive and negative instances but has a slightly lower performance for overall prediction.

From Table 4 it can be concluded that the CART algorithm has the lowest performance among all the algorithms.

Table 4 shows that all algorithms exhibit high specificity but do not achieve correspondingly high ROC-AUC scores. This suggests that while the algorithm demonstrates strong performance in correctly identifying true negatives (exhibiting high specificity), it may encounter challenges in effectively distinguishing between true positives and false positives across different thresholds, ultimately leading to a lower ROC-AUC score. This might be due to data imbalance (Fig 1(A)). Random oversampling is used to balance the data (Fig 1(B)).

Table 5

Performance metric scores of each algorithm after Random Oversampling

Name	Accuracy(%)	Precision (%)	Recall(%)	F1(%)	Specificity (%)
CART	96.01	96.30	99.97	96.01	92.06
SVM	72.48	72.56	75.38	72.45	69.58
LINEAR LOGISTIC	72.42	72.42	72.38	72.42	72.46
KNN	94.27	94.65	98.86	94.26	89.71
RANDOM TREE	98.38	98.43	99.97	98.38	96.79

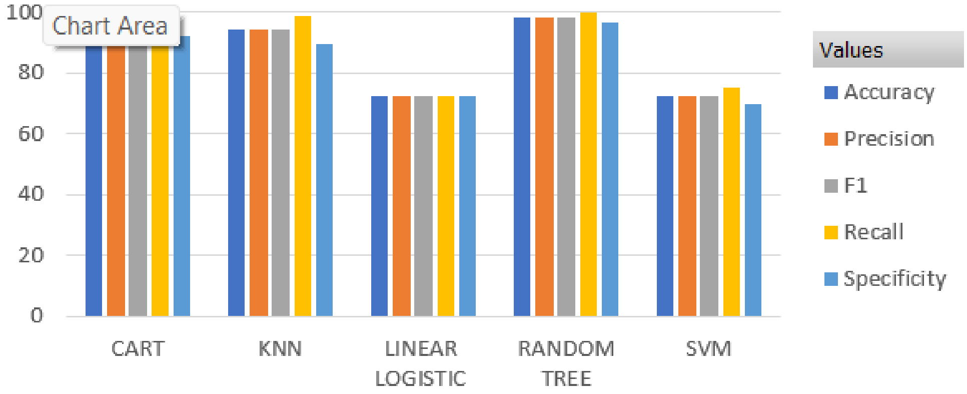


Fig 11:Graphical representation of the performance metric scores of each algorithm after Random Oversampling

Random tree algorithm has the highest performance among all other machine learning algorithms after data oversampling, that is when the data is balanced. The performance metrics for the random tree algorithm, based on Table 5, are as follows: accuracy score of 98.38%, recall score of 99.97%and F1 score of 98.38%. Fig 9(B) shows the ROC graph of a random tree algorithm which has an ROC-AUC score of 1. This indicates that the algorithm accurately classifies 98.38% of instances, and a high recall score implies that the model is effective at identifying most of the relevant instances of the positive class. It means that the model has a low rate of false negatives, meaning it correctly detects a large proportion of the actual positive cases.

From Fig 5(B) and Fig 8(B) KNN(at k=2) and CART exhibit high ROC-AUC indicating their effectiveness in distinguishing between positive and negative classes across different thresholds. Therefore it is evident that the performance of both models, especially CART, has improved after applying the ROS technique. This is because ROS balances class distribution, reduces bias, enhances discrimination between classes, and increases sensitivity to the minority class, leading to higher ROC-AUC scores and improved overall performance.

Overfitting occurs when a model excessively learns from the training data, capturing noise and outliers rather than the underlying patterns, resulting in poor performance on unseen data.
Fig 6 shows that the ROC-AUC score decreases for SVM. In the context of oversampling, the augmented presence of minority class instances may lead the SVM model to overly adapt to these instances, potentially intensifying the overfitting issue. Hence leading to a lower ROC-AUC score

The ROC-AOC curve does not change for Linear Logistic classification as from Fig 7 we can see that oversampling techniques like random oversampling balance datasets without altering the ROC or AOC curve. These methods aim to balance the dataset by duplicating instances of the minority class, without altering the fundamental relationship between true positive rate (TPR) and false positive rate (FPR). While oversampling enhances the model's capacity to recognize the minority class, thereby potentially boosting metrics like accuracy, precision, and recall, it doesn't change the underlying structure or characteristics of the ROC or AOC curve.

The decrease in the overall performance of the model due to overfitting is usually addressed by carefully tuning the hyperparameters such as regularisation parameter(C) in SVM and linear logistic algorithm. This is done to control the model's complexity and reduce it from overly relying on minority class samples. However, it was discovered that while running the data set, changing the regularisation parameter(C) did not alter the performance scores of the model.

5. CONCLUSION

The analysis reveals that before oversampling, the linear logistic classification algorithm outperforms others, with an accuracy and recall score of 93.81%, F1 score of 90.91%, and ROC-AOC of 0.8. However, after applying random oversampling to balance the dataset, the random tree algorithm excels the others, with an accuracy of 98.38% and an ROC-AUC of 1. Despite improvements in KNN and CART algorithms post-oversampling, overfitting remains a challenge, particularly for SVM. In general, oversampling increases the performance of each algorithm. This study underscores the necessity of balancing data to enhance algorithm performance and suggests careful hyperparameter tuning to mitigate overfitting, although regularisation parameters did not impact performance scores significantly in this instance.

6. REFERENCE

[1] The top 10 causes of death, World Health Organization[WHO],(2020)

[2] P. Shingare, Diabetes, Hypertension and Stroke Prediction, Kaggle, (2022)

[3] Bandi, Vamsi, Bhattacharyya, Debnath, Midhunchakkravarthy, Divya, Prediction of Brain Stroke Severity Using Machine Learning, Revue d'Intelligence Artificielle 34 (6) (2020) 753-761

[4] S. Rahman, M. Hasan, A. K. Sarkar, Prediction of Brain Stroke using Machine Learning Algorithms and Deep Neural Techniques, European Journal of Electrical Engineering & Computer Science 7 (1) (2023)

[5] Z. G. Al-Mekhlafi, E. M. Senan, T. H. Rassem, B. A. Mohammed, N. M. Makbol, A. A. Alanazi, T. S. Almurayziq, F. A. Ghaleb, Deep Learning and Machine Learning for Early Detection of Stroke and Haemorrhage, Computers, Materials & Continua 72 (1) (2022)

[6] B. Akter, A. Rajbongshi, S. Sazzad, R. Shakil, J. Biswas, U. Sara, A Machine Learning Approach to Detect the Brain Stroke Disease, 4th International Conference on Smart Systems and Inventive Technology (ICSSIT) (2022)

[7] C. H. Lin, K. C. Hsu, K. R. Johnson, Y. C. Fann, C. H. Tsai, Y. Sun, L. M. Lien, W. L. Chang, P. L. Chen, C. L. Lin, C. Y. Hsu, Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry, Comput Methods Programs Biomed, 190 (2022)

[8] S. Dev, H. Wang, C. S. Nwosu, N. Jain, B. Veeravalli, D. John, A predictive analytics approach for stroke prediction using machine learning and neural networks, Healthcare Analytics 2 (2022)

[9] G. Sailasya, G. L. A. Kumari, Analyzing the Performance of Stroke Prediction using ML Classification Algorithms, International Journal of Advanced Computer Science and Applications, 12 (6) (2021)

[10] E. Dritsas, M. Trigka, Stroke Risk Prediction with Machine Learning Techniques, Sensors 22 (13) (2022)

[11] E. M. Alanazi, A. Abdou, J. Lou, Predicting Risk of Stroke From Lab Tests Using Machine Learning Algorithms: Development and Evaluation of Prediction Models, JMIR Formative Research 5 (12) (2021)

[12] T. Badriyah, N. Sakinah, I. Syarif, D. R. Syarif, Machine Learning Algorithm for Stroke Disease Classification, IEEE, (2020) 1-5

[13] L. Breiman, J. Freidman, R. A. Olshen, C. J. Stone, Classification and Regression Trees, Chapman and Hall/CRC (1) (1984)

[14] V. N. Vapnik, A. Ya. Chervonenkis, “A class of algorithms for pattern recognition learning”, Avtomat. i Telemekh., (1964)

[15]D. R. Cox, The Regression Analysis of Binary Sequences, Journal of the Royal Statistical Society, Series B (Methodological), 20(2), 215-242, (1958)

[16] E.Fix, J.L. Hodges, Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties, USAF School of Aviation Medicine, (1951)

[17] L. Breiman, Random Forests, Kluwer Academic Publishers, 45, 5–32, (2001)