A

MINI PROJECT REPORT

ON

Brain Stroke Prediction Using Machine Learning Approach

SUBMITTED IN PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

In

ARTIFICIAL INTELLIGENCE

BY

M. Mithesh(20EG107131)
C. Anurag (20EG107108)
V. Geethika(20EG107153)

UNDER THE GUIDANCE OF

Mrs . Neetha Reddy,
ASSISTANT PROFESSOR,
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

DEPARTMENT OF ARTIFICIAL INTELLIGENCE
ANURAG UNIVERSITY,VENKATAPUR
TELANGANA– 500088

2023-2024

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

We submit this project work entitled "**Brain Stroke Prediction Using Machine Learning Approach** " to Anurag University, in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE and declare that it is our original and independent work carried out under the guidance of "**Mrs .Neetha Reddy,Assistant professor** " ,Department of Artificial Intelligence.

Date :

Name: M.Mithesh
Place: Hyderabad

Signature:
Roll No: 20eg107131

Name: C.Anurag
Place: Hyderabad

Signature:
Roll No: 20eg107108

Name: V.Geethika
Place: Hyderabad

Signature:
Roll No: 20eg107153

TABLE OF CONTENTS

CHAPTER **PG.NO**

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Random Forest
KNN (K-Nearest Neighbor)
Naive Bayes
SVM (Support Vector Machine)
Cat Boost
Decision Tree
MLP (Multi-layer perceptron)

ACKNOWLEDGEMENT

We extend our sincere thanks to **Prof. S Ramachandram**, Vice- Chancellor, Anurag University, for his encouragement and constant coordination.

We would like to thank **Prof. Balaji Utla,** Registrar of Anurag University,for his valuable suggestions and motivation to give our best while working on this project.

We take pleasure in thanking **Dr. A Mallikarjuna Reddy**, Head Of Department, Department of Artificial Intelligence, Anurag University, for his incredible support and inspiration.

It is our privilege to express gratitude, and indebtedness to our coordinator **Ms. T. Neetha,** Assistant Professor, Department of Artificial Intelligence, Anurag University, for her incredible support and constructive criticism which helped us do better.

We express our sincere gratitude to **Ms. T. Neetha,** Assistant Professor, Department of Artificial Intelligence, Anurag University, for her valuable insights, motivation, and guidance for the successful completion of the work.

ABSTRACT

Stroke is a medical condition in which poor blood flow to the brain results in cell death. Nowadays stroke has been found by inspecting the affected individuals. Using these risk factors, a number of works have been carried out for predicting and classifying stroke diseases. Most of the models are based on data mining and machine learning algorithms.

In many nations, stroke is the main cause of endomorphy and death. The goal of this research was to figure out how to make things better. We have used the stroke disease data set from Kaggle. Patients can benefit from data that has been pre-processed. Ischemic stroke and stroke hemorrhage are two forms of stroke, Individuals are divided into two categories using machine learning methods.

Machine learning techniques were employed seven times in this investigation. Logistic Regression, Support Vector Machine (SVM), Random Forest, Cat Boost, Naive Bayes, KNearest Neighbors Because of this, our findings, Random Forest makes the best accuracy, along with precision and recall values, and the f1-Score.

1.INTRODUCTION

According to the World Stroke Organization, 13 million people get a stroke each year, and approximately 5.5 million people will die as a result. It is the leading cause of death and disability worldwide, and that is why its imprint is serious in all aspects of life. Stroke not only affects the patient but also affects the patient's social environment, family and workplace. In addition, contrary to popular belief, it can happen to anyone, at any age, regardless of gender or physical condition.

Factors that increase the chance of having a stroke are the existence of a similar stroke in the past, the existence of a transient stroke, the presence of myocardial infarction, and other heart diseases, such as heart failure, atrial fibrillation, age (if someone is over 55 years of age, they are clearly more likely to be affected, although stroke is described at any age, even in children), hypertension, carotid stenosis from atherosclerosis, smoking, high blood cholesterol, diabetes, obesity, sedentary lifestyle, alcohol consumption, blood clotting disorders, estrogen therapy and the use of euphoric substances such as cocaine and amphetamines .

Moreover, stroke progresses rapidly, and its symptoms can vary. Symptoms can sometimes develop slowly and sometimes it can develop quickly. It is even possible for someone to wake up while sleeping with symptoms. A stroke occurs with the sudden onset of one or more symptoms. The main ones are paralysis of the arms or legs (usually on one side of the body), numbness in the arms or legs or sometimes on the face, difficulty speaking, difficulty walking, dizziness, decreased vision, headache and vomiting and a drop in the angle of the mouth (crooked mouth). Finally, in severe strokes, the patient loses consciousness and falls into a coma Once the patient has had a stroke, a computerized tomography (CT) scan immediately provides a diagnosis. In the case of ischemic stroke, magnetic resonance imaging (MRI) is efficient. Other ancillary diagnostic tests are carotid triplex and cardiac triplex. Strokes can be severe (extensive) or mild. In the vast majority of cases, the first 24 h are crucial. The diagnosis will highlight the treatment, which is usually pharmaceutical, and, in a few cases, surgical. Intubation and mechanical ventilation in the intensive care unit are necessary when the patient has fallen into a coma.

Although some patients recover after a stroke, the vast majority continue to have problems depending on the severity of the stroke, such as memory, concentration and attention problems, difficulty speaking or understanding speech, emotional problems such as depression, loss of balance or the ability to walk, loss of sensation on one side of the body and difficulty swallowing food.

Recovery helps to regain lost function after a stroke. The appropriate plan is created so that the patient immediately returns psychologically and socially with kinesiotherapy, speech therapy and the contribution of neurologists. In order to minimize the chances of having a stroke, it is necessary to

regularly monitor blood pressure, exercise regularly, maintain a normal weight, quit smoking and drinking alcohol and follow a healthy diet without fat and salt.

Information and communication technologies (ICTs), and especially the fields of artificial intelligence (AI) and machine learning (ML), now play an important role in the early prediction of various diseases, such as diabetes (as a classification or regression task for continuous glucose prediction), hypertension, cholesterol, COVID-19, COPD, CVDs, ALF, sleep disorders, hepatitis, CKD, etc. In particular, the stroke will concern us in the context of this study. For this specific disease, many research studies have been conducted with the aid of machine learning models.

In this research work, a methodology for designing effective binary classification ML models for stroke occurrence is presented. Since class balancing is crucial for the design of efficient methods in stroke prediction, the synthetic minority over-sampling technique (SMOTE) method was applied. Then, various models are developed, configured and assessed in the balanced dataset. For our purpose, Machine learning techniques were employed seven times in this investigation. Logistic Regression, Support VectorMachine (SVM), Random Forest, Cat Boost, Naive Bayes, KNearest Neighbors Because of this, our findings, Random Forest makes the best accuracy, along with precision and recall values, and the f1- Score.

2. PROBLEM STATEMENT

EXISTING SYSTEM

A stroke occurs due to poor blood flow to the brain which results in cell death. Two main types of strokes are ischemic stroke and hemorrhagic stroke. Ischemic stroke occurs due to lack of blood flow and hemorrhagic stroke occurs due to bleeding. Another type of stroke is transient ischemic attack. Ischemic stroke has two categories- embolic stroke and thrombotic stroke. An embolic stroke occurs by forming a clot in any part of the body and moves toward the brain and blocks blood flow. A thrombotic stroke caused by a clot that weakens blood flow in an artery. Hemorrhagic stroke is classified into two types- subarachnoid hemorrhage and intracerebral hemorrhage. Transient ischemic G+attack is also known as "ministroke".

There are many ways of finding brain stroke. one such method is finding strokes using machine learning. We have an existing system that is low accuracy. Previously used a machine learning algorithm called decision tree which consists of low accuracy.

Disadvantages

Incorrect classification Results.
Accuracy of the classification

PROPOSED SYSTEM

Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Tree, Cat boost Logistic Regression, and ensemble methods (Bagging and Boosting) to classify the stroke disease we have collected the data from online website, India which contains information about 507 stroke patients ranging from 35 to 90 years of age. The novelty of their work is in the data processing phase, where an algorithm called novel stemmer was used to attain the dataset. In their collected dataset, 91.52% of patients were affected by ischemic stroke and only 8.48% of patients were affected by hemorrhagic stroke. Among the mentioned algorithms, Artificial Neural Networks with stochastic gradient descent learning algorithms have the highest accuracy with 95.3% for classifying stroke**.**

Advantages

High performance.

provide accurate predictions.

3. SYSTEM REQUIREMENTS

Software Requirements

- O/S : Windows 8.1.
- Language : python.
- ID : Pycharm

FEATURES OF PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- web development (server-side).
- software development
- mathematics.
- System scripting.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way or a functional way.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently where as other languages use punctuation, and it has fewer syntactic constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

• **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

FEASIBILITY STUDY

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

- Economic Feasibility
- Technical Feasibility
- Behavioural Feasibility

· **Economic Feasibility**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and savings that are expected from the system of the proposed system. The hardware in the system department is sufficient for system development.

· **Technical Feasibility**

This study centers around the system's department hardware, software and to what extent it can support the proposed system department having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

· **Behavioural Feasibility**

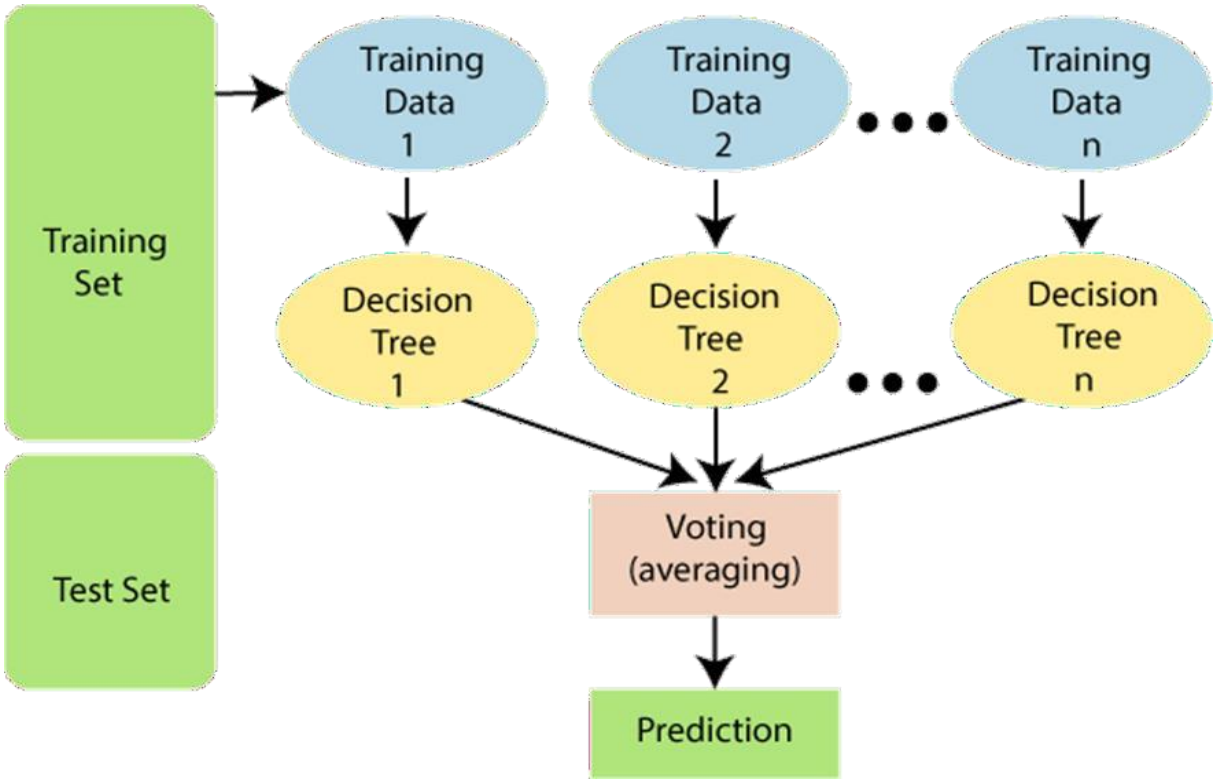
People are inherently resistant to change and need a sufficient amount of training, which would result in a lot of expenditure for the organization. The proposed system can generate reports with day-to-today information immediately at the user's request, instead of getting a report, which doesn't contain much detail.

4. TYPES OF ALGORITHMS USED

· RANDOM FOREST

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.



Block diagram of random forest

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

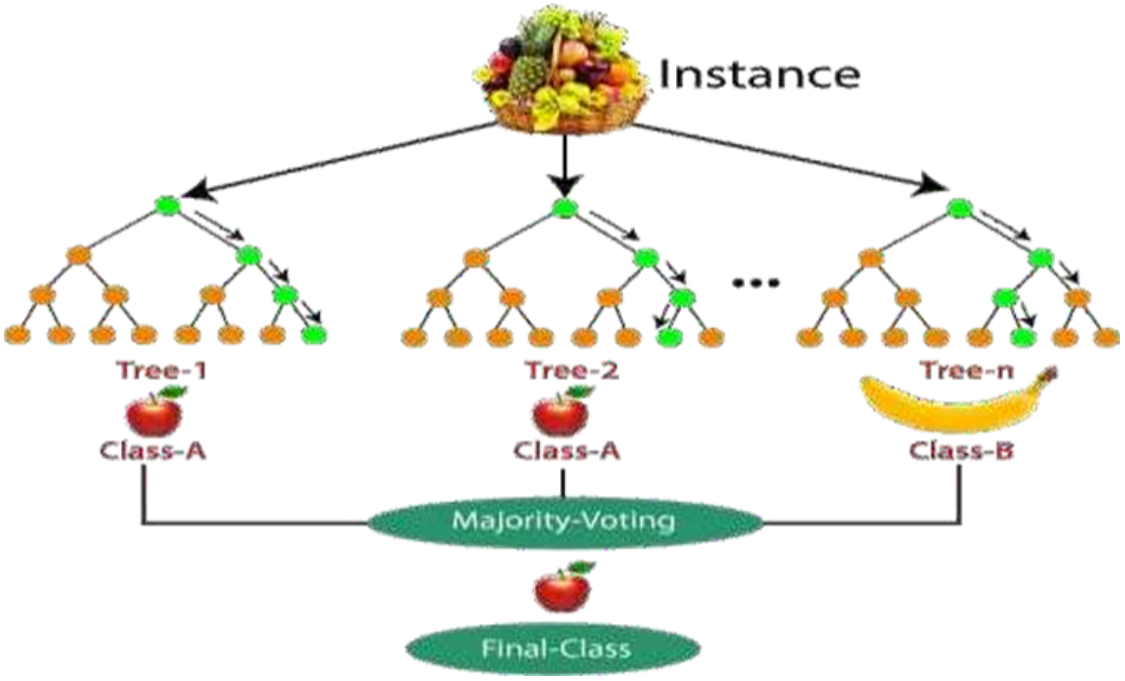
Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data pointoccurs, then based on the majority of results, the Random Forest classifier predicts the final decision.Consider the below image.



Example of random forest

Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

Naïve Bayes Classifier Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each

feature individually contributes to identify that it is an apple without depending on each other.

- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' Theorem:

- Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for Credit Scoring.
- It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as Spam filtering and Sentiment analysis.

SVC(SUPPORT VECTOR CLASSIFIER)

A pattern recognition problem involves three operations: pre processing, feature extraction and classification. The study of pattern classification has received a great deal of attention. A classifier is used to assign an input vector to a category. There are many different types of classifiers. Among all of them, the neural network classifier has been widely studied during the last decade. The latest advance on classification is known as support vector machine (SVM). Support vector classifier (SVC), developed based on SVM, has been applied to numerical pattern recognition, face detection, text categorization and protein fold recognition.

Conventionally the SVC is used for classification of input data that are supplied and computed in batch. For a classification problem that involves a large data set, it is time consuming to achieve the classification using the SVC. Therefore, it is not possible to apply the SVC for real- time classification problems. In many signal/image application problems, data are collected online in real time, one by one in a sequence and the classification needs to be undertaken and completed before the next data is received. In this case, an online training algorithm would be desirable.

In this paper, an online support vector classifier (OSVC) is proposed for the pattern classification problems that have input data supplied in sequence rather than in batch. An OSVC algorithm is presented in detail and its convergence is analyzed. The OSVC is applied to three benchmark problems: Iris data classification, image segmentation and numerical pattern recognition. The results demonstrate that the OSVC algorithm has not only a much faster convergence than the conventional SVC algorithms but also results in a smaller number of support vectors preserving the same quality of separating hyperplane, while the generalization performance is as good as that of the existing SVC algorithms.

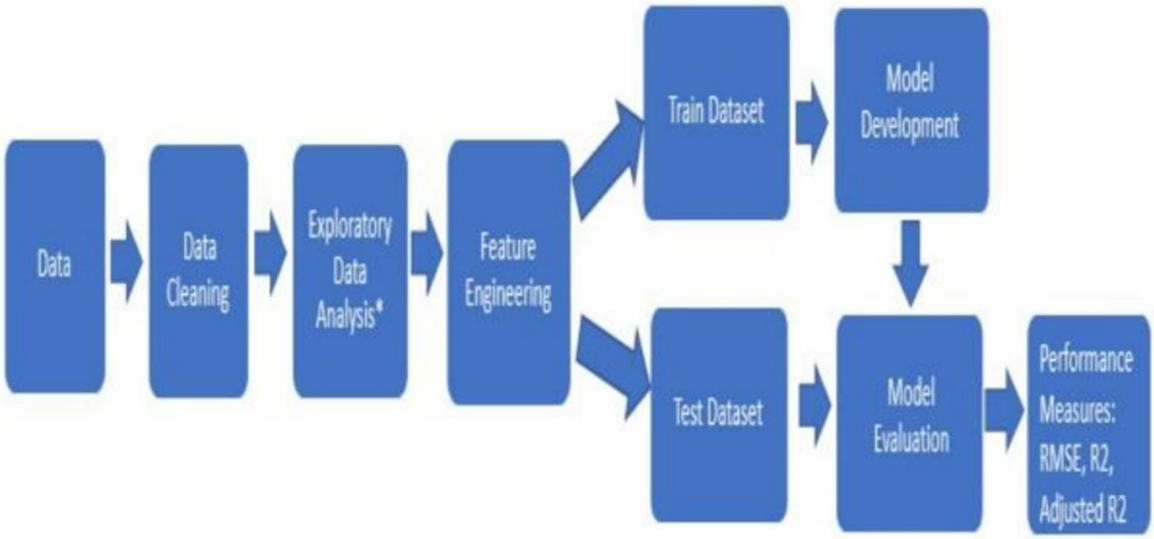
CATBOOST

CatBoost is a machine learning algorithm that is specifically designed for gradient boosting on decision trees. It was developed by Yandex, a Russian multinational IT company, and is known for its high performance and efficiency in various applications, particularly in tasks related to classification and regression. CatBoost stands for "Categorical Boosting," and it is particularly effective when dealing with categorical features in your data.

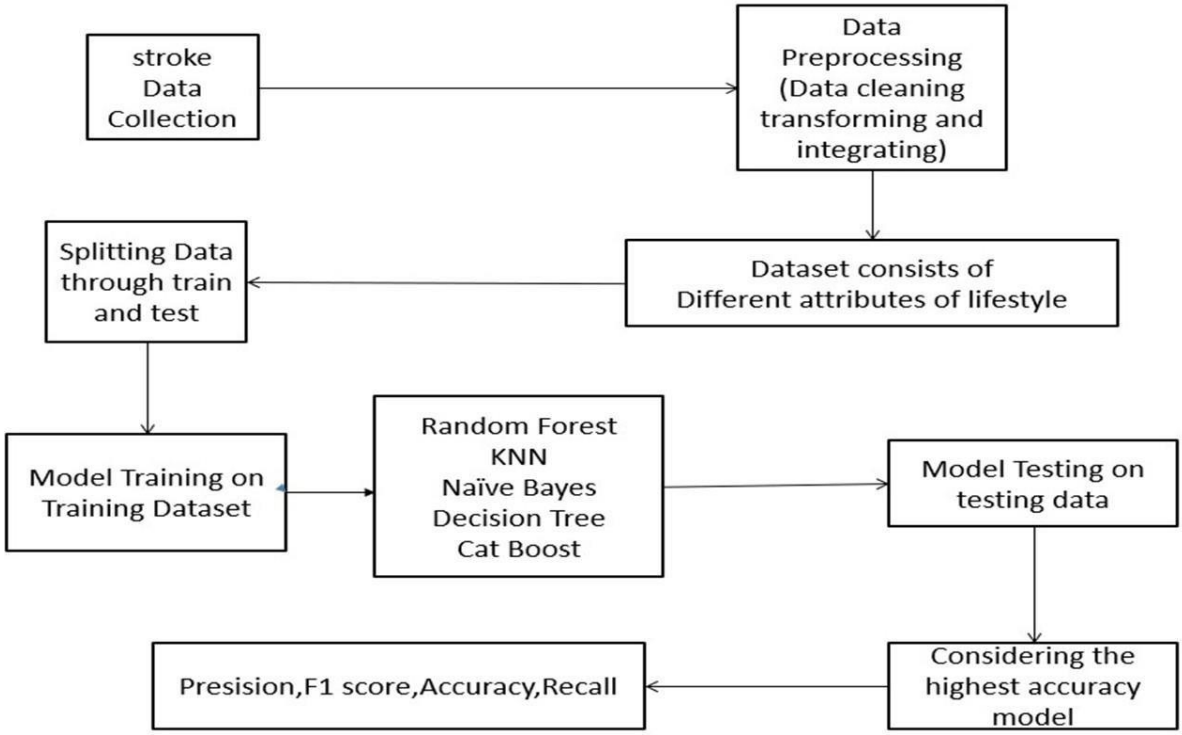
Advantages:

1. Handling Categorical Data: CatBoost is particularly adept at handling categorical features without the need for one-hot encoding or other complex preprocessing steps. It internally encodes categorical data and automatically selects the most suitable encoding scheme for each feature.

2. High Performance:CatBoost is known for its high prediction accuracy and robustness. It often outperforms other gradient boosting algorithms, such as XGBoost and LightGBM, on various datasets.
3. Efficiency: It is optimized for efficiency and speed. CatBoost utilizes a series of optimization techniques to reduce overfitting and speed up training, including ordered boosting, oblivious trees, and efficient computation of split values.
4. Built-in Cross-Validation:CatBoost has a built-in cross-validation feature that helps you to assess the performance of your model and select hyperparameters without needing external libraries.
5. Natural Language Processing (NLP) Support: It can be effectively used for NLP tasks, thanks to its ability to handle text and categorical data well.
6. Robustness to Overfitting: CatBoost incorporates techniques to reduce overfitting, including learning rate regularization and ordered boosting, making it more robust against overfitting compared to some other boosting algorithms.



ARCHITECTURE DIAGRAM



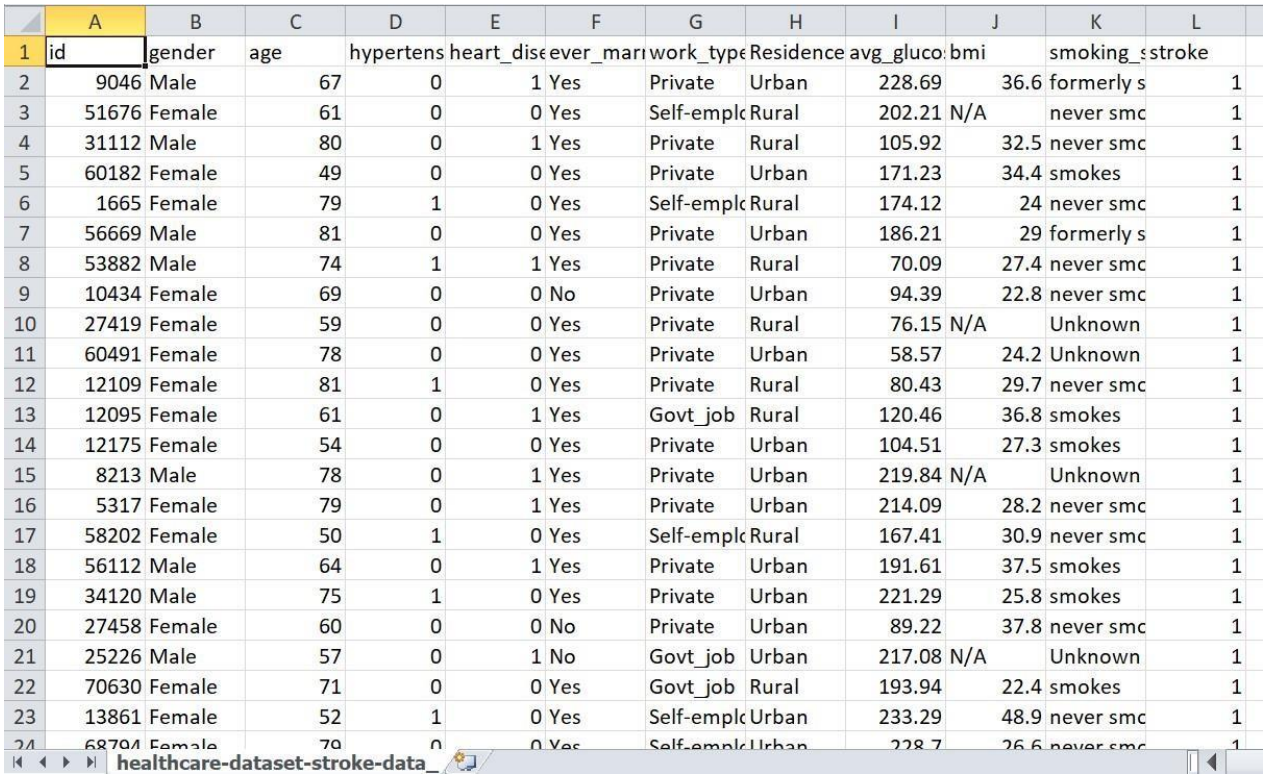
Architecture diagram

- Data Selection and Loading
- Data Pre processing
- Missing Data Removal
- Encoding Categorical Data
- Splitting Dataset into Train and Test Data
- Feature Extraction
- Classification
- Prediction

DATA SELECTION AND LOADING

- The data selection is the process of selecting the data for detecting the attacks.
- In this project, the brain disease dataset is used for detecting disease.
- Gender,age,hypertension,brain_disease,ever_married,work_type,Residence_type,avg_glu c ose_level,bmi,smoking_status,stroke

DATA PREPROCESSING



	A	B	C	D	E	F	G	H	I	J	K	L
1	id	gender	age	hypertens	heart_dise	ever_mari	work_type	Residence	avg_gluco	bmi	smoking_s	stroke
2	9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly s	1
3	51676	Female	61	0	0	Yes	Self-emplo	Rural	202.21	N/A	never smo	1
4	31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smo	1
5	60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
6	1665	Female	79	1	0	Yes	Self-emplo	Rural	174.12	24	never smo	1
7	56669	Male	81	0	0	Yes	Private	Urban	186.21	29	formerly s	1
8	53882	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smo	1
9	10434	Female	69	0	0	No	Private	Urban	94.39	22.8	never smo	1
10	27419	Female	59	0	0	Yes	Private	Rural	76.15	N/A	Unknown	1
11	60491	Female	78	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1
12	12109	Female	81	1	0	Yes	Private	Rural	80.43	29.7	never smo	1
13	12095	Female	61	0	1	Yes	Govt_job	Rural	120.46	36.8	smokes	1
14	12175	Female	54	0	0	Yes	Private	Urban	104.51	27.3	smokes	1
15	8213	Male	78	0	1	Yes	Private	Urban	219.84	N/A	Unknown	1
16	5317	Female	79	0	1	Yes	Private	Urban	214.09	28.2	never smo	1
17	58202	Female	50	1	0	Yes	Self-emplo	Rural	167.41	30.9	never smo	1
18	56112	Male	64	0	1	Yes	Private	Urban	191.61	37.5	smokes	1
19	34120	Male	75	1	0	Yes	Private	Urban	221.29	25.8	smokes	1
20	27458	Female	60	0	0	No	Private	Urban	89.22	37.8	never smo	1
21	25226	Male	57	0	1	No	Govt_job	Urban	217.08	N/A	Unknown	1
22	70630	Female	71	0	0	Yes	Govt_job	Rural	193.94	22.4	smokes	1
23	13861	Female	52	1	0	Yes	Self-emplo	Urban	233.29	48.9	never smo	1
24	68794	Female	79	0	0	Yes	Self-emplo	Urban	228.7	26.6	never smo	1

healthcare-dataset-stroke-data_

• Data preprocessing is the process of removing the unwanted data from the dataset.

Missing data removal

Encoding Categorical data

• **Missing data removal:** In this process, the null values such as missing values are removed using the imputer library.

• **Encoding Categorical data:** That categorical data is defined as variables with a finite set of label values. That most machine learning algorithms require numerical input and output variables. That an integer and one hot encoding is used to convert categorical data to integer data.

SPLITTING DATASET INTO TRAIN AND TEST DATA

• Data splitting is the act of partitioning available data into two portions, usually for cross-validation purposes

• One portion of the data is used to develop a predictive model. And the other to evaluate the model's performance.

• Separating data into training and testing sets is an important part of evaluating data mining models.

• Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

FEATURE EXTRACTION

• Feature scaling. Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data pre-processing step.

• Feature Scaling or Standardization: It is a step of Data Pre Processing which is applied to independent variables or features of data. It basically helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

CLASSIFICATION

• Logistic regression is another technique borrowed by machine learning from the field of statistics.

• It is the go-to method for binary classification problems (problems with two class values). In this post you will discover the logistic regression algorithm for machine learning.

• Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

PREDICTION

• It’s a process of predicting the attacks in the network from the dataset.

• This project will effectively predict the data from the dataset by enhancing the performance of the overall prediction results.

• And also find the accuracy of the prediction and generate a confusion matrix for the output.

CONCLUSION

In this, we use machine learning methods to classify the stroke data. Then, eight algorithms—Logistic Regression, K-Nearest Neighbors, Naive Bayes, Decision Tree, Random Forest, Multi-layer Perceptron, Deep Learning and Support Vector Machine, and Cat Boost—are compared to perform classification. Our experiment indicates that, when compared to other assessed classification methods, the Random Forest approach was employed as a classification methodology. The Random Forest method provides the highest degree of accuracy. The precision of the categorization algorithm using theThe value of the default optimization parameter has not been examined nonetheless. The classification model possesses the possibility for improvement right now. To raise the machine learning algorithm's accuracy, parameters need to be adjusted. More individuals than individuals with ischemic strokes and brain hemorrhages.

FUTURE SCOPE

This project helps to predict stroke risk using prediction models in older people and for people who are addicted to the risk factors as mentioned in the project. In future, the same project can be extended to give the stroke percentage using the output of the current project. This project can alsobe used to find the stroke probabilities in young people and underage people by collecting respective risk factor information’s and doctors consulting

APPENDIX I

Source Code

1. Import and Install Dependencies

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import imblearn as ib
import warnings
```

2. PRE PROCESSING

```
df = df.drop(['id'],axis=1)

### Gender analysis

df['gender'].value_counts()

df['gender'] = df['gender'].replace('Other','Female')
df['gender'].value_counts().plot(kind="pie")

## Target feature - Stroke

### Stroke analysis

df['stroke'].value_counts()

df['stroke'].value_counts().plot(kind="bar",color = "cyan")

print("% of people who actualy got a
stroke",(df['stroke'].value_counts()[1]/df['stroke'].value_counts().sum()).round(3)*100)

 #Hyper-tension Analysis

   df['hypertension'].value_counts().plot(kind="bar",color  =  "red")

 #Work type Analysis
```

```
df['work_type'].value_counts()
df['work_type'].value_counts().plot(kind="pie")
#Smoking status Analysis
df['smoking_status'].value_counts()
df['smoking_status'].value_counts().plot(kind="pie")
#Residence type Analysis df['Residence_type'].value_counts()
df['Residence_type'].value_counts().plot(kind="pie")
#BMI analysis
df['bmi'].isnull().sum()
ns.histplot(data=df['bmi'])
sns.boxplot(data=df['bmi'])Q1
= df['bmi'].quantile(0.25)Q3 =
df['bmi'].quantile(0.75)
IQR = Q3 - Q1
da=(df['bmi'] < (Q1 - 1.5 * IQR)) | (df['bmi'] > (Q3 + 1.5 * IQR))
df['bmi'].isna().sum()/len(df['bmi'])*100
df_na=df.loc[df['bmi'].isnull()]
g=df_na['stroke'].sum()
print("People who got stroke and their BMI is NA:",g)
h=df['stroke'].sum()
```

```
print("People who got stroke and their BMI is given:",h)
print("percentage of people with stroke in Nan values to the overall dataset:",g/h*100)
df['stroke'].sum()/len(df)*100
df_na=df.loc[df['bmi'].isnull()]
print("Nan BMI values where people have stroke:",df_na['stroke'].sum())print("overall BMI
values where people have stroke:",df['stroke'].sum())print("median of
bmi",df['bmi'].median()) df['bmi']=df['bmi'].fillna(df['bmi'].median())
#AGE analysis
sns.histplot(data=df['age'])
sns.boxplot(data=df['age'])
#AVERAGE GLUCOSE LEVEL ANALYSIS
sns.histplot(data=df['avg_glucose_level'])
sns.boxplot(data=df['avg_glucose_level']) Q1 =
df['avg_glucose_level'].quantile(0.25)Q3 =
df['avg_glucose_level'].quantile(0.75)
IQR = Q3 - Q1
da=(df['avg_glucose_level'] < (Q1 - 1.5 * IQR)) | (df['avg_glucose_level'] > (Q3 + 1.5 * IQR))
da.value_counts()
corrmat=df.corr()
f,ax=plt.subplots(figsize=(9,8))
```

```
sns.heatmap(corrmat,ax=ax,cmap="YlGnBu",linewidth=0.8,annot=True)
```

1. Heart_disease analysis

```
df['heart_disease'].value_counts()

df['heart_disease'].value_counts().plot(kind="pie")
```

#Ever_married analysis with Values

```
df['ever_married'].value_counts()

df['ever_married'].value_counts().plot(kind="pie")
```

2. Cross analysis - all the attribute compared with target attribute

```
sns.countplot(x='stroke', hue='gender', data=df)

sns.countplot(x='stroke', hue='work_type', data=df)

sns.countplot(x='stroke', hue='smoking_status', data=df)

sns.countplot(x='stroke', hue='Residence_type', data=df)

sns.countplot(x='stroke', hue='heart_disease', data=df)

sns.countplot(x='stroke', hue='ever_married', data=df) #Creating
```

dummy variables for numeric-binary attributes

```
df[['hypertension', 'heart_disease', 'stroke']] = df[['hypertension', 'heart_disease', 'stroke']].astype(str)df =

pd.get_dummies(df, drop_first= True)

df.head()
```

```
from imblearn.over_sampling import RandomOverSampler
oversample = RandomOverSampler(sampling_strategy='minority')
X=df.drop(['stroke_1'],axis=1)
y=df['stroke_1']
X_over, y_over = oversample.fit_resample(X, y) from
sklearn.preprocessing import StandardScalers =
StandardScaler()
df[['bmi', 'avg_glucose_level', 'age']] = s.fit_transform(df[['bmi', 'avg_glucose_level', 'age']])
#Creating test-train split (80-20 split)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_over, y_over, test_size= 0.20, random_state= 42)
print('X_train:', X_train.shape)
print('y_train:',  y_train.shape)
print('X_test:',  X_test.shape)
print('y_test:', y_test.shape)
```

3. Training Model

1. Decision Tree

```
from sklearn.metrics import classification_reportfrom
sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
```

```
from sklearn.metrics import auc,roc_auc_score,roc_curve,precision_score,recall_score,f1_scoreclf =
DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)
DT_accuracy = metrics.accuracy_score(y_test, y_pred)
print("Accuracy:",DT_accuracy)
print(classification_report(y_test, y_pred))
```

2. KNN

```
sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
from sklearn.metrics import auc,roc_auc_score,roc_curve,precision_score,recall_score,f1_scoreknn =
KNeighborsClassifier(n_neighbors = 2)
knn.fit(X_train,y_train) y_pred_knn
= knn.predict(X_test)
y_pred_prob_knn = knn.predict_proba(X_test)[:, 1]
confusion_matrix(y_test, y_pred_knn)
KNN_accuracy = accuracy_score(y_test, y_pred_knn)
print('Accuracy:',KNN_accuracy)
print('ROC AUC Score:', roc_auc_score(y_test, y_pred_prob_knn))print(classification_report(y_test,
y_pred_knn))
```

3. XGBoost

```
from xgboost  import XGBClassifier

xgb = XGBClassifier()

xgb.fit(X_train,y_train)

y_pred_xgb = xgb.predict(X_test) y_pred_prob_xgb =

xgb.predict_proba(X_test)[:, 1]

XG_accuracy  = accuracy_score(y_test, y_pred_xgb)

print('Accuracy:',XG_accuracy)

print('ROC AUC Score:', roc_auc_score(y_test, y_pred_prob_xgb))fpr,

tpr, thresholds = roc_curve(y_test, y_pred_prob_xgb)

plt.figure(figsize=(6,4))

plt.plot(fpr, tpr, linewidth=2, color= 'teal')

plt.plot([0,1], [0,1], 'r--' )

plt.title('ROC Curve of XGBOOST')

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.show()

from sklearn.metrics import confusion_matrix

confusion_matrix(y_test, y_pred_xgb)

from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score

print("Accuracy_score:",accuracy_score(y_test,y_pred_xgb))
```

```
print("Precision_score:",precision_score(y_test,y_pred_xgb))

print("Recall_score:",recall_score(y_test,y_pred_xgb)) print("f1_score:",f1_score(y_test,y_pred_xgb))

print('ROC AUC Score:', roc_auc_score(y_test, y_pred_prob_xgb))

print(classification_report(y_test, y_pred_xgb))
```

4. Random Forest

```
from sklearn.ensemble import RandomForestClassifier from

sklearn.model_selection import RandomizedSearchCVrf_clf =

RandomForestClassifier(n_estimators = 100) rf_clf.fit(X_train,

y_train)

y_pred_rf = rf_clf.predict(X_test)

RF_accuracy = accuracy_score(y_test, y_pred_rf)

print('Accuracy:',RF_accuracy)

from sklearn import model_selection

from sklearn.model_selection import KFold

kfold_kridge = model_selection.KFold(n_splits=20, shuffle=True)

results_kfold = model_selection.cross_val_score(rf_clf, X_over, y_over, cv=kfold_kridge)

print("Accuracy: ", results_kfold.mean()*100)

print(results_kfold)

from sklearn.metrics import confusion_matrix
```

```
confusion_matrix(y_test, y_pred_r)

print(classification_report(y_test, y_pred_rf))
```

5. Logistic regression

```
from sklearn.linear_model import LogisticRegressionclassifier =

LogisticRegression(random_state = 0) classifier.fit(X_train,

y_train)

y_pred_lr = classifier.predict(X_test)

confusion_matrix(y_test, y_pred_lr)

LR_accuracy = accuracy_score(y_test, y_pred_lr)

print('Accuracy:', LR_accuracy)

print(classification_report(y_test, y_pred_lr))
```

6. SVM

```
from sklearn.svm import SVCsv

= SVC()

sv.fit(X_train,y_train)

y_pred_sv=sv.predict(X_test)

SVM_accuracy = accuracy_score(y_test,y_pred_sv)

print(classification_report(y_test,y_pred_sv))
```

7. AdaBoost

```
from sklearn.ensemble import AdaBoostClassifierfrom

sklearn.metrics import accuracy_score
```

```
adb = AdaBoostClassifier()

adb.fit(X_train,y_train)

y_pred_adb=adb.predict(X_test)

ADa_accuracy = accuracy_score(y_test,y_pred_adb)

print(ADa_accuracy)

print(classification_report(y_test,y_pred_adb))
```

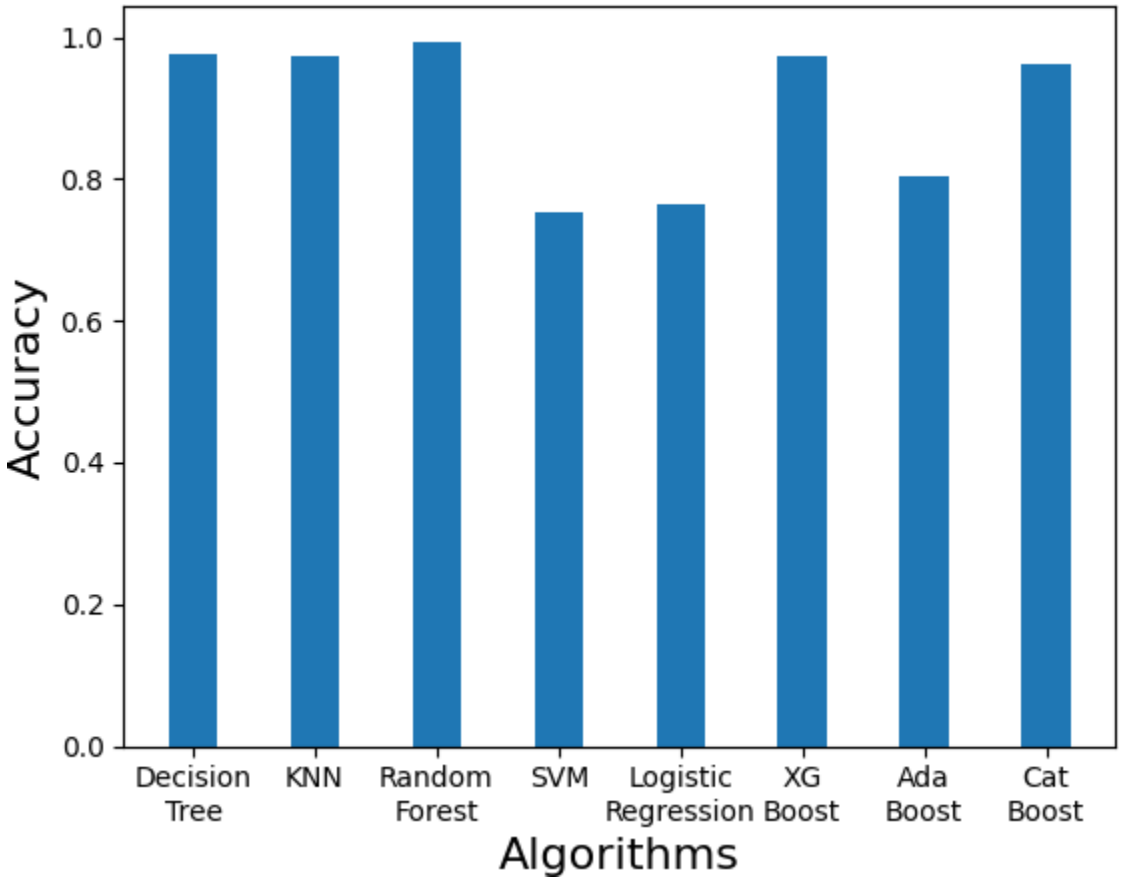
8. CatBoost

```
pip install catboost

import catboost as ctb

model_CBC = ctb.CatBoostClassifier()

model_CBC.fit(X_train, y_train)

expected_y  = y_test

predicted_y = model_CBC.predict(X_test)

CB_accuracy = accuracy_score(expected_y,predicted_y)

print(CB_accuracy)

print(classification_report(expected_y,predicted_y)) import

matplotlib.pyplot as plt

import numpy as np

x=np.array(["Decision\nTree","KNN","Random\nForest","SVM","Logistic\nRegression","XG\nBoost","A
da\nBoost","Cat\nBoost"])

y=np.array([DT_accuracy,KNN_accuracy,RF_accuracy,SVM_accuracy,LR_accuracy,XG_accuracy,AD
a_accuracy,CB_accuracy])
```

```
plt.xlabel("Algorithms",fontsize = 16)

plt.ylabel("Accuracy",fontsize = 16)

plt.bar(x,y,width = 0.4)

plt.show()

age=75

avg_glucose_level=300

bmi=36.6 gender_Male=1

ever_married_Yes=1

work_type_Never_worked=0

work_type_Private=1

work_type_Self_employed=0

work_type_children=0

Residence_type_Urban=1

smoking_status_formerly_smoked=1

smoking_status_never_smoked=0

smoking_status_smokes=0

hypertension_1=1 heart_disease_1=1

input_features = [age ,avg_glucose_level, bmi ,gender_Male,hypertension_1,
heart_disease_1,ever_married_Yes, work_type_Never_worked, work_type_Private,
work_type_Self_employed,      work_type_children ,Residence_type_Urban,
smoking_status_formerly_smoked,smoking_status_never_smoked ,smoking_status_smokes]
```

```
features_value = [np.array(input_features)]

features_name = ['age' ,'avg_glucose_level', 'bmi' ,'gender_Male' ,'hypertension_1',
'heart_disease_1','ever_married_Yes', 'work_type_Never_worked', 'work_type_Private',
'work_type_Self-employed',  'work_type_children'  ,'Residence_type_Urban',
'smoking_status_formerly smoked','smoking_status_never smoked' ,'smoking_status_smokes']

df = pd.DataFrame(features_value, columns=features_name)

prediction = rf_clf.predict(df)[0]

print(prediction)
```

APPENDIX II



References:

- Concept of Stroke by Healthline.
- Dataset named ‘Stroke Prediction Dataset’ from Kaggle: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/code
- Stroke Prediction Using Machine Learning Algorithms: https://ijirem.org/DOC/2-strokeprediction-using-machine-learningalgorithms.pdf.
- G. R. Kumar, P. Vyshnavi, S. Prasanna, T. H. Reddy, C. Charanya, and P. Chandrababu, “Brain Stroke Detection Using Machine Learning", IJREAM, vol. 5, no. 3, pp. 34–36, Mar. 2022.
- Mr.M.THIRUNAVUKKARASU, et. al. International Journal of Engineering Research and Applications
  www.ijera.com ISSN: 2248-9622, Vol. 13, Issue 4, April 2023, pp. 237-242.