BRAIN STROKE PREDICTION

MINI PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

REGULAVALASA TEJASWI　　　　　**BHASKARA RAO LINGALA**

(19331A05E2)　　　　　　　　**(20331A0518)**

CHERUKURI ASHIK　　　　　　　**CH AMMI REDDY**

(20331A0534)　　　　　　　　**(20331A0542)**

Under the Supervision of

Mr.Srikanth Ganta

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MVGR COLLEGE OF ENGINEERING (Autonomous) VIZIANAGARAM-535005, AP (INDIA)

(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

2022-2023

CERTIFICATE

This is to certify that the project report entitled " **BRAIN STROKE PREDICTIONS** " is being submitted by **REGULAVALASA TEJASWI , BHASKARA RAO LINGALA, CHERUKURI ASHIK, CH AMMI REDDY** bearing registered numbers **19331A05E2, 20331A0518, 20331A0534, 20331A0542** respectively, in partial fulfilment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of bonafide work done by them under my supervision during the academic year 2022-2023.

Mr.Srikanth Ganta	**Dr. P. Ravi Kiran Varma**
Associate Professor	**Head of the Department**
Dept. of CSE	**Dept. of CSE**

<u>**ABSTRACT**</u>

Brain stroke sufferers have rapidly increased due to excessive alcohol consumption, smoking, high cholesterol, high blood pressure, and cardiovascular diseases. Early brain stroke prediction is possible so people can quickly diagnose the disease at its early stage. It is more valuable to the Health Department and the medical expert system used in remote areas. The brain plays a significant role in life, and it is the seat of intelligence, interpreter of the senses, initiator of body movement, and controller of behavior. So early prediction is critical to diagnose the disease and its recovery. The system aims to improve brain stroke prediction using machine learning approaches. The main aim of the system is to use classification algorithms to identify brain stroke patients from healthy individuals. The system also aims to compare the classification algorithms based on their performance factors.

Keywords: Brain Stroke, Machine Learning, Supervised Learning algorithms.

CONTENTS

CHAPTER NAME   **Page No**

LIST OF FIGURES

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO
7.1	Table 7.1	22
7.2	Table 7.2	22
7.3	Table 7.3	23
10.1	TEST CASE	31

LIST OF ACRONYMS

- ANN   : Artificial Neural Networks
- DT   : Decision Tree
- UML   : Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

Brain stroke is the fourth leading cause of death in India, according to a report by the World Health Organization (WHO), accounting for 11% of all fatalities. The brain is the most crucial part of our human body. The cerebrum, brainstem, and cerebellum make up the brain. The brain's weight is approximately three pounds, which is the main organ for the human senses, thinking ability, creativity, and balance. The brain is around 60% fat and combines water, protein, carbohydrates, and salts. The skull bones of the head enclose and shield the brain from external harm. The brain is not a muscle itself. The brain comprises nerves, veins, arteries, neurons, and glial cells. Brain stroke is caused due to damage of brain blood vessels. The stroke prevents the brain from oxygen and nutrients; perhaps it leads to the death of brain cells. The spinal cord that protrudes from the brain makes up the central nervous system or CNS.

There are two types of brain strokes. Ischemic stroke is the other type, and the other one is hemorrhage stroke. Ischemic stroke is caused when the blood flow to a portion of the brain is blocked or diminished, and brain tissue cannot get oxygen and nutrients. In a minute, brain cells start to degenerate. Hemorrhage stroke is caused when there is a rupture in the blood vessels. The leading common cause of a breakdown of an uneven tangle of thin-walled blood arteries is a less frequent cause of bleeding in the brain. The main reasons for causing brain stroke are excessive smoking, alcohol consumption, high cholesterol, high blood pleasure, obesity, and cardiovascular diseases, such as atrial fibrillation, heart failure, congenital heart abnormalities, and heart infections.

The symptoms of the brain stroke are sudden attack of paralysis, trouble speaking, vision problems in both eyes, loss of balance, and headaches without any reason. Early detection of brain stroke can help treat the disease to fast recovery. Some precautions to prevent brain stroke are regular doctor visits, exercising, avoiding smoking and drinking alcohol, controlling blood pleasure, diabetes, and maintaining weight.

 If the person is older than 55, the risk of a brain stroke is higher than that of an average person. People of all other races and ethnicities have a lower risk of stroke than African Americans and Hispanics. By 2050, low and moderate-income nations will account for more than 80% of the estimated 15 million additional strokes worldwide. Additionally, the expense of stroke hospitalization is rising, necessitating the development of new technology to aid in clinical

diagnosis, treatment, clinical event prediction, referral of viable therapeutic approaches, and rehabilitation programs.

The existing system of the medical expert system for the prediction of brain stroke has been helpful to society. The patients get to benefit from the use of proper technological assistance in this aspect. Different machine learning algorithms have been developed to improve the quality and accuracy of the detection or prediction of brain stroke in the medical area.

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW

Md. Shafiul Azam et al. [1] proposed various pre-processing techniques to balance the dataset. By using three machine learning algorithms, Logistic Regression (LR), Decision Tree (DT), and Random Forest (RF), they predict whether a person has a chance of getting a brain stroke or not. The main aim of the thesis is to predict the risk of brain stroke and to analyze the performance of these algorithms.

Tessy Badriyah et al. [2] proposed different machine learning algorithms like Logistic Regression (LR), Decision Tree (DT), Random Forest (RT), KNN, Support Vector Machine (SVM), and Naïve Bayes (NB) to predict the possibilities of a brain stroke. They compare the accuracies of those algorithms. Also, a comparison of the precision score, recall score, and F1 score is done.

Priya Govindarajan et al. [3] classify stroke by using data mining tools and machine learning algorithms like bagging and boosting techniques, Logistic Regression (LR), Support Vector Machine (SVM), Linear SVM, Gaussian SVM, and Artificial Neural Networks (ANN). This paper identifies the symptoms of brain stroke and predicts whether the person is suffering from ischemic or hemorrhage stroke. Also, a comparison between men and women to know who has a high chance of risk.

Nur Sakinah et al. [4] considered different CT scan images of both ischemic stroke and hemorrhage stroke images as a dataset performed pre-processing techniques to improve image quality. Also, by applying further deep learning and machine learning algorithms like Random Forest (RT), Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbor, and Multi-Layer Perceptron (MLP-NN) to detect whether the patient is suffering from ischemic stroke or hemorrhage stroke.

In this study, M. Sheetal Singh et al. [5], the backpropagation neural network classification method is applied to build a classification model. At the same time, the decision tree approach is used to minimize the dimensions to detect the stroke.

\Based on their suggested conservative mean, Vamsi Bandi et al. [6] introduced a unique automatic feature selection technique that chooses robust features. For further effectiveness, they paired this approach with the Support Vector Machine (SVM) to predict brain stroke effectively.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the existing system, the classification is done for diagnosing and predicting brain stroke in medical areas. The machine learning methods like Decision Tree, Logistic Regression, and Random Forest are used to develop the thesis.
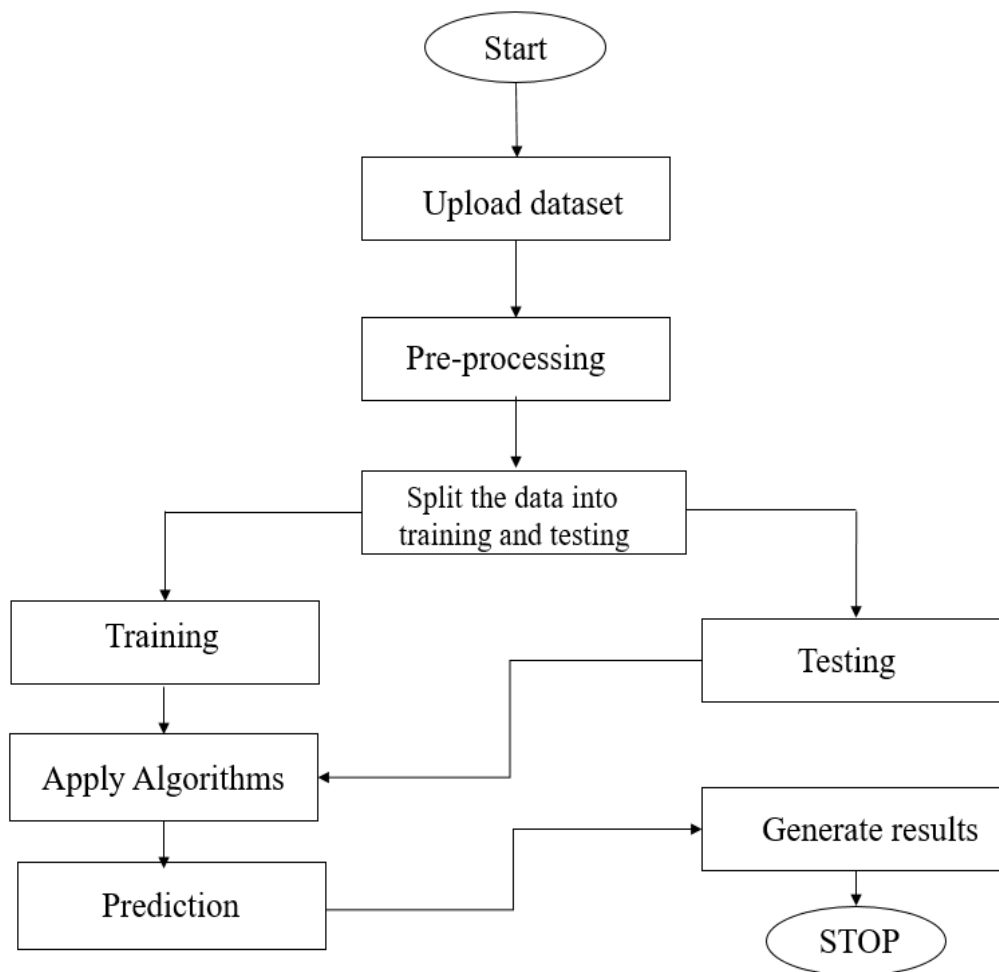
3.2 PROPOSED SYSTEM

The proposed system has evolved to overcome the drawbacks of the existing system. The system Artificial Neural Networks (ANN), Naïve Bayes, and Decision tree is used to predict whether or not the patient has a chance of getting a stroke.

ADVANTAGES OF PROPOSED SYSTEM

- o High Accuracy:
- o Naïve Bayes got better accuracy of 95%.

3.3 SYSTEM ARCHITECTURE:

```
                        ┌─────────┐
                        │  Start  │
                        └────┬────┘
                             │
                   ┌─────────▼─────────┐
                   │  Upload dataset   │
                   └─────────┬─────────┘
                             │
                   ┌─────────▼─────────┐
                   │  Pre-processing   │
                   └─────────┬─────────┘
                             │
                   ┌─────────▼─────────┐
                   │ Split the data into│
                   │ training and testing│
                   └──────┬──────┬──────┘
                          │      │
          ┌───────────────┘      └───────────┐
          │                                  │
   ┌──────▼──────┐                    ┌───────▼──────┐
   │  Training   │                    │   Testing    │
   └──────┬──────┘                    └───────┬──────┘
          │                                   │
   ┌──────▼────────────┐ ◄─────────────────── │
   │ Apply Algorithms  │                      │
   └──────┬────────────┘          ┌───────────▼──────┐
          │                       │ Generate results │
   ┌──────▼──────┐                └───────┬──────────┘
   │ Prediction  │───────────────►        │
   └─────────────┘                  ┌─────▼─────┐
                                    │   STOP    │
                                    └───────────┘
```

CHAPTER 4

SOFTWARE REQUIREMENTS

AND SPECIFICATIONS

4. SYSTEM REQUIREMENTS AND SPECIFICATIONS

4.1 HARDWARE REQUIREMENTS

•   Processor   : Intel i5 or higher

•   RAM   : 8 GB or higher

•   Hard Disk   : 128 GB

4.2 SOFTWARE REQUIREMENTS

•   Operating System   : Windows 7/8/10/11

•   IDE   : Jupyter Notebook

•   Technology Used   : Python 3.9

4.3 TECHNOLOGIES USED:

NumPy:

Numerous people have contributed to the open-source program NumPy. Large, multi-dimensional arrays and matrices are supported by the Python programming language's Numpy library, which also provides a vast variety of high-level mathematical operations for use on these arrays. The ancestor of Numpy, Numeric, was created by Jim Holguin with contributions from several other developers. In 2005, Travis Oliphant created Numpy by incorporating features of the competing Num array into Numeric, with extensive modifications.

Pandas:

A software package called panda was created for the Python programming language to manipulate and analyze data. It includes the specific data structures and procedures for working with time series and mathematical tables. It is a free software distributed under the BSD license's three clauses.

Scikit-Learn:

Scikit-learn is perhaps the foremost helpful library for machine learning in python. Scikit-learn contains a lot of efficient tools for machine learning and statistics including classification, regression, clustering, and dimensionality reduction. The components used are Supervised

learning algorithms, cross-validation, Unsupervised learning algorithms, various toy datasets, and feature extraction.

Scikit-learn is modify built in python, and it heavily relies on NumPy for high-speed linear algebra and array operations. In addition, to boost performance, some key algorithms are written in Cython. A Cython wrapper around LIBSVM implements support vector machines, and a similar wrapper around LIBLINEAR implements logistic regression and linear support vector machines. It may not be possible to expand the methods with python in such instances.

OS:

The OS module in Python provides functions for interacting with the operating system. The module provides a portable way of using operating system = dependent functionality. OS comes under Python's standard utility modules. The *os* and *os. path* modules include many functions to interact with the file system.

Seaborn:

Seaborn is a library for visualization of graphical statistical plotting in Python. Seaborn provides many color palettes and defaults beautiful styles to create many statistical plots in Python more attractive.

Matplotlib:

Matplotlib is a cross-platform for data visualization and graphical charting package in Python and its numerical extension NumPy. As a result, it acts as an open-source replacement for MATLAB. The APIs (Application Programming Interfaces) of Matplotlib can also be used to include charts in GUI programmers.

 Two APIs are overlaid by the Matplotlib scripting layer.

Matplotlib is at the top of the pyplot API hierarchy of Python code objects.

Pyplot is an OO (Object-oriented) API collection of objects that can be assembled more easily than pyplot. This API allows you to use matplotlib backend layers directly.

CHAPTER 5

DATASET

5. DATASET

5.1 DATASET:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
1	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
2	Male	67	0	1	Yes	Private	Urban	228	36	formerly smoked
3	Female	61	0	0	Yes	Self-employed	Rural	202	37	never smoked
4	Male	80	0	1	Yes	Private	Rural	105	32	never smoked
5	Female	49	0	0	Yes	Private	Urban	171	34	smokes
6	Female	79	1	0	Yes	Self-employed	Rural	174	24	never smoked
7	Male	81	0	0	Yes	Private	Urban	186	29	formerly smoked
8	Male	74	1	1	Yes	Private	Rural	70	27	never smoked
9	Female	69	0	0	No	Private	Urban	94	22	never smoked
10	Female	59	0	0	Yes	Private	Rural	76	30	Unknown
11	Female	78	0	0	Yes	Private	Urban	58	24	Unknown
12	Female	81	1	0	Yes	Private	Rural	80	29	never smoked
13	Female	61	0	1	Yes	Govt_job	Rural	120	36	smokes
14	Female	54	0	0	Yes	Private	Urban	104	27	smokes
15	Male	78	0	1	Yes	Private	Urban	219	30	Unknown
16	Female	79	0	1	Yes	Private	Urban	214	28	never smoked
17	Female	50	1	0	Yes	Self-employed	Rural	167	30	never smoked
18	Male	64	0	1	Yes	Private	Urban	191	37	smokes
19	Male	75	1	0	Yes	Private	Urban	221	25	smokes
20	Female	60	0	0	No	Private	Urban	89	37	never smoked
21	Male	57	0	1	No	Govt_job	Urban	217	26	Unknown
22	Female	71	0	0	Yes	Govt_job	Rural	193	22	smokes
23	Female	52	1	0	Yes	Self-employed	Urban	233	48	never smoked
24	Female	79	0	0	Yes	Self-employed	Urban	228	26	never smoked
25	Male	82	0	1	Yes	Private	Rural	208	32	Unknown
26	Male	71	0	0	Yes	Private	Urban	102	27	formerly smoked
27	Male	80	0	0	Yes	Self-employed	Rural	104	23	never smoked
28	Female	65	0	0	Yes	Private	Rural	100	28	formerly smoked
29	Male	58	0	0	Yes	Private	Rural	189	26	Unknown
30	Male	69	0	1	Yes	Self-employed	Urban	195	28	smokes

Dataset Description:

- The Txt file dataset is used for predicting brain stroke.

- Size of the dataset: 316.97 KB

- The dataset contains a total of 5110 entries.

- 5110 entries are used.

- The dataset contains 10 fields.

- Training data   : 4088 (80%)

- Testing data   : 1022 (20%)

- Dataset is collected from the reference of:

  https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

CHAPTER 6

SYSTEM DESIGN

6. SYSTEM DESIGN

6.1 UML DIAGRAMS:

The term "UML" stands for "Unified Modeling Language." A general-purpose modeling language with standards, UML is used in object-oriented software engineering. The Object Management Group is in charge of creating and managing standards. The objective of UML is to establish itself as a standard language for modeling object-oriented computer programs consisting primarily of a meta-model and the notation in its current version. The addition or connection of a method or process to UML is also possible in the future. UML is a standard language used for business modeling, non-software systems, and describing, visualizing, building, and documenting the artifacts of a software system. The UML is a collection of tried-and- sound engineering best practices for simulating extensive, complicated systems. The software development process and object-oriented software development both heavily rely on UML. UML primarily employs graphical notation to express software project design.

GOALS:

 The following main objectives in the UML are as follows:

1. Offer users an expressive, ready-to-use visual modeling language so they can create and trade meaningful models.
2. Offer methods for specialization and extensibility to expand the fundamental ideas.
3. Be needed to overcome specific development methodologies and programming languages.
4. Offer an official framework for comprehending the modeling language.
5. .Promote the expansion of the tools market.
6. .Encourage the use of higher-level development ideas, including collaborations, frameworks, patterns, and components.
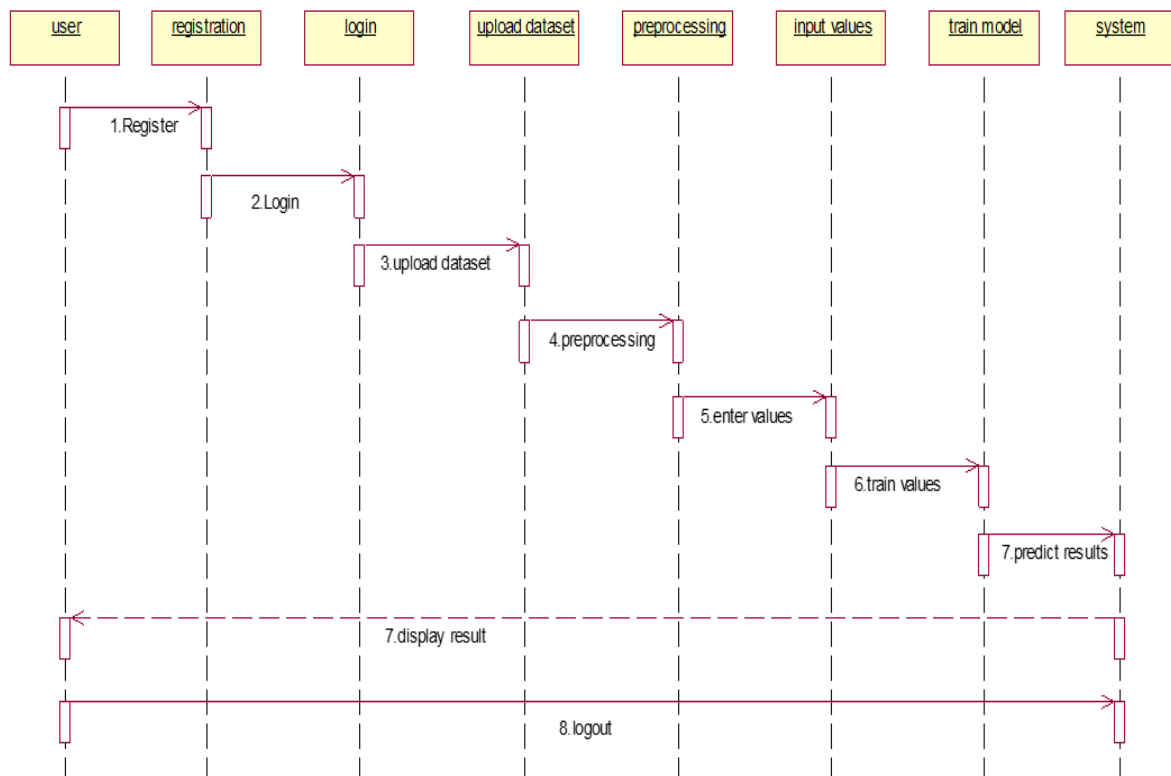7. Include top techniques.

6.1.1 FLOWCHART DIAGRAM:

A flowchart is a diagram that demonstrates how a system or process works. They are widely used in many different fields to examine, organize, enhance and communicate often a complex process in simple and understandable diagrams. Rectangles, ovals, diamonds and possibly many other shapes are used in flowcharts to define the type of step, along with the arrows with connections between them to define flow and sequence.
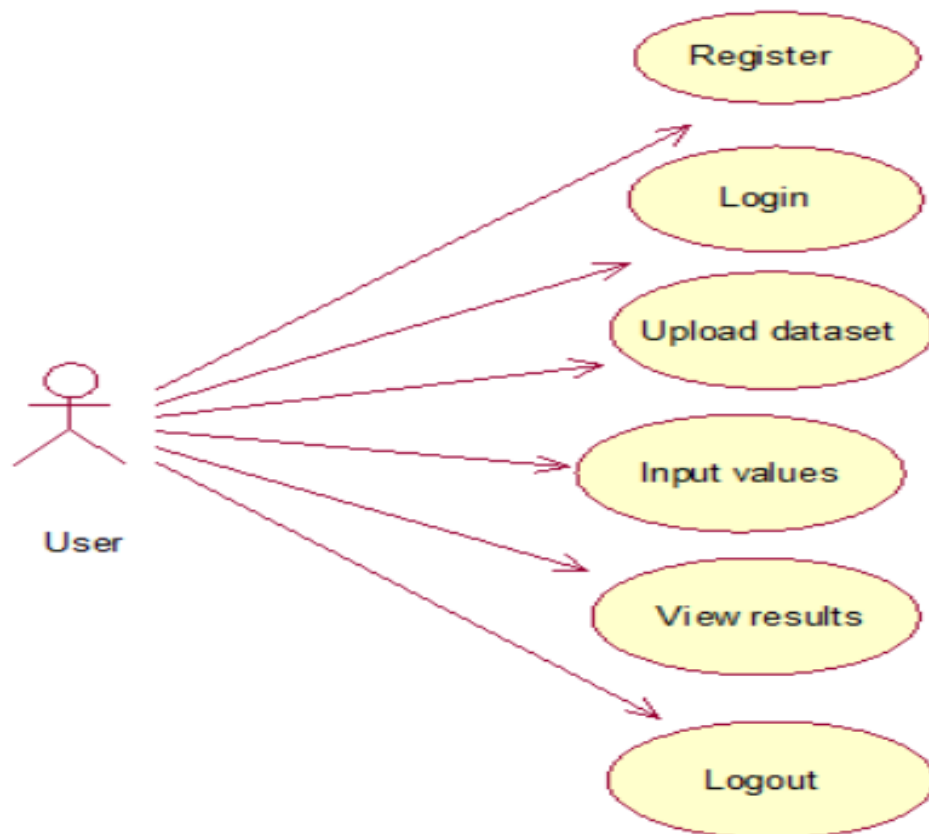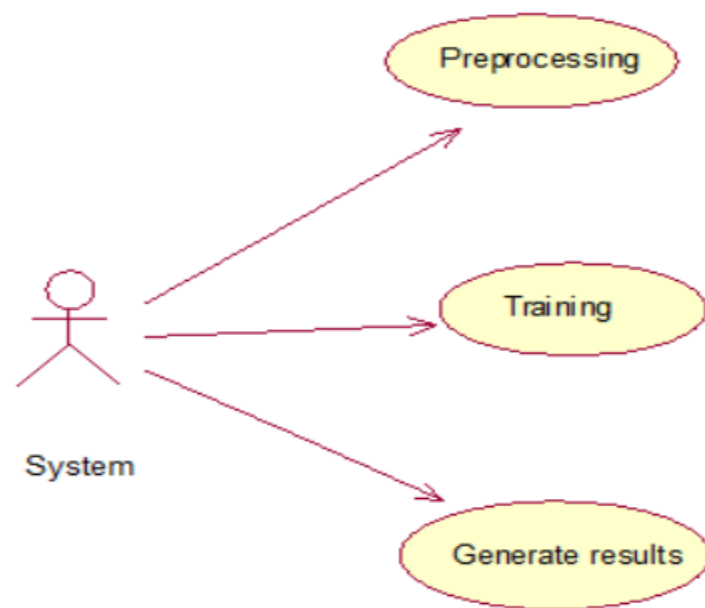
```
                    ┌──────────────┐
                   (    Start       )
                    └──────┬───────┘
                           ▼
                    ┌──────────────┐
                    │     User      │
                    └──────┬───────┘
                           ▼
                    ┌──────────────┐
                    │   Register    │
                    └──────┬───────┘
                           ▼
                    ╱──────────────╲
                    │ Input values  │
                    ╲──────┬───────╱
                           ▼
                    ┌──────────────┐
                    │Apply Algorithms│
                    └──────┬───────┘
                           ▼
                        ◇ if there is a risk ◇ ──No──► ┌──────────────┐
                        ◇ of brain stroke   ◇          │ Healthy brain │
                           │                            └──────┬───────┘
                          Yes                                  │
                           ▼                                   │
                    ┌──────────────┐                           │
                    │ chance of having│                         │
                    │ a brain stroke │                          │
                    └──────┬───────┘                           │
                           ▼                                   │
                   (    END       ) ◄──────────────────────────┘
```
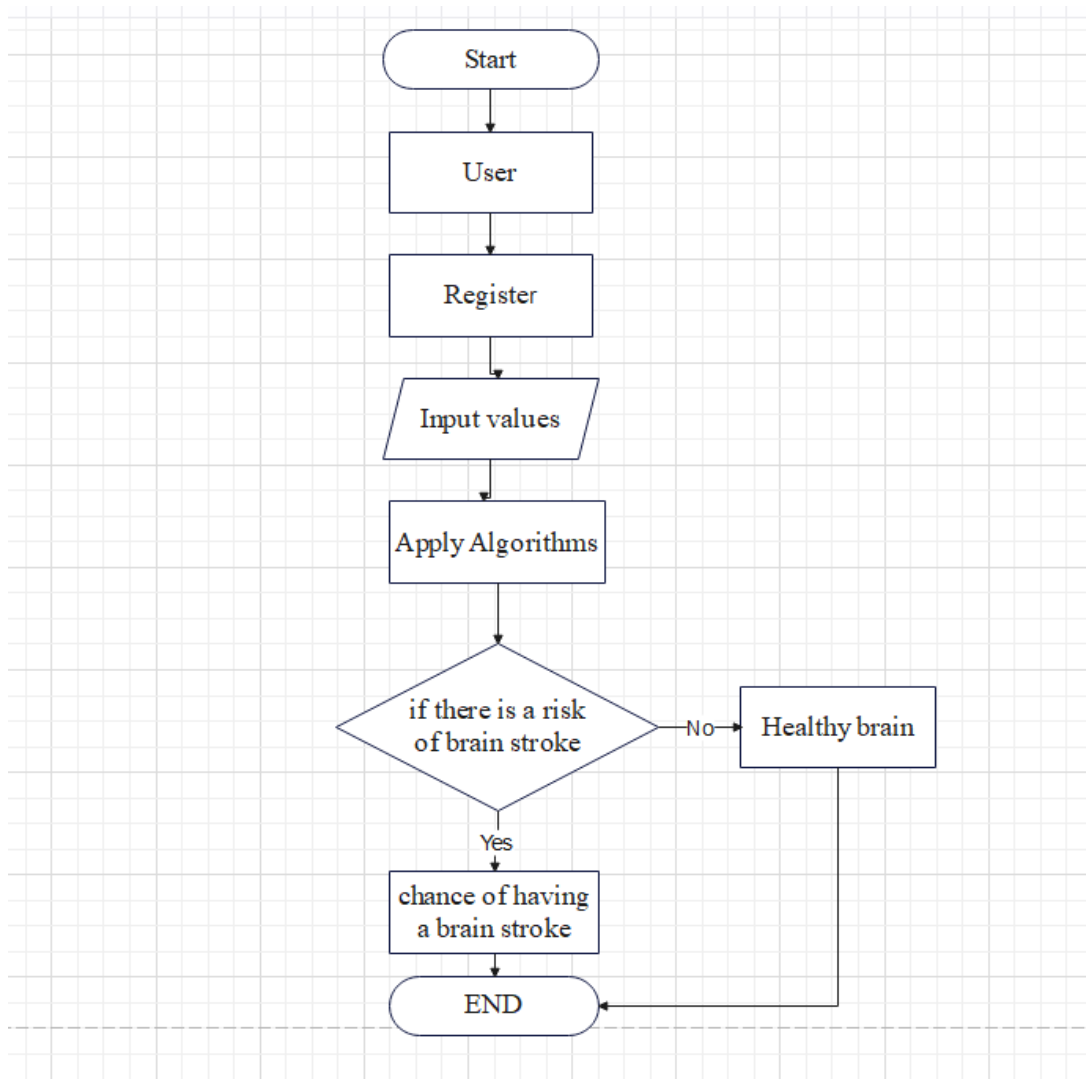
6.1.2 SEQUENCE DIAGRAM:

A series diagram is a shape of an interplay diagram that suggests how strategies interact with one to any other and in what order. It's referred to as a Message Sequence Chart. Sequence diagrams are also known as occasion diagrams, situations, and timing diagrams. In the diagram, the consumer will interact with the application.

user	registration	login	upload dataset	preprocessing	input values	train model	system

1.Register

2.Login

3.upload dataset

4.preprocessing

5.enter values

6.train values

7.predict results

7.display result

8.logout

6.1.3 USE CASE DIAGRAM:

A use-case analysis in the Unified Modeling Language specifies and derives a type of behavioral diagram. Its purpose is to provide a graphical depiction of a system's functionality in terms of the actor, objectives, and any connections between those use cases that could be present. The main aim of use case diagram is to show which system functions are carried out for which actor.

Preprocessing

Training

Generate results

System

Register

Login

Upload dataset

Input values

View results

Logout

User

6.1.4 ACTIVITY DIAGRAM:

The activity diagram describes the activity of the process in step by step process in terms of choices, iterations, and concurrencies. Activity diagrams may be used in the Unified Modelling Language to walk through a system component's business and operational procedures. The activity diagram depicts the whole control flow.
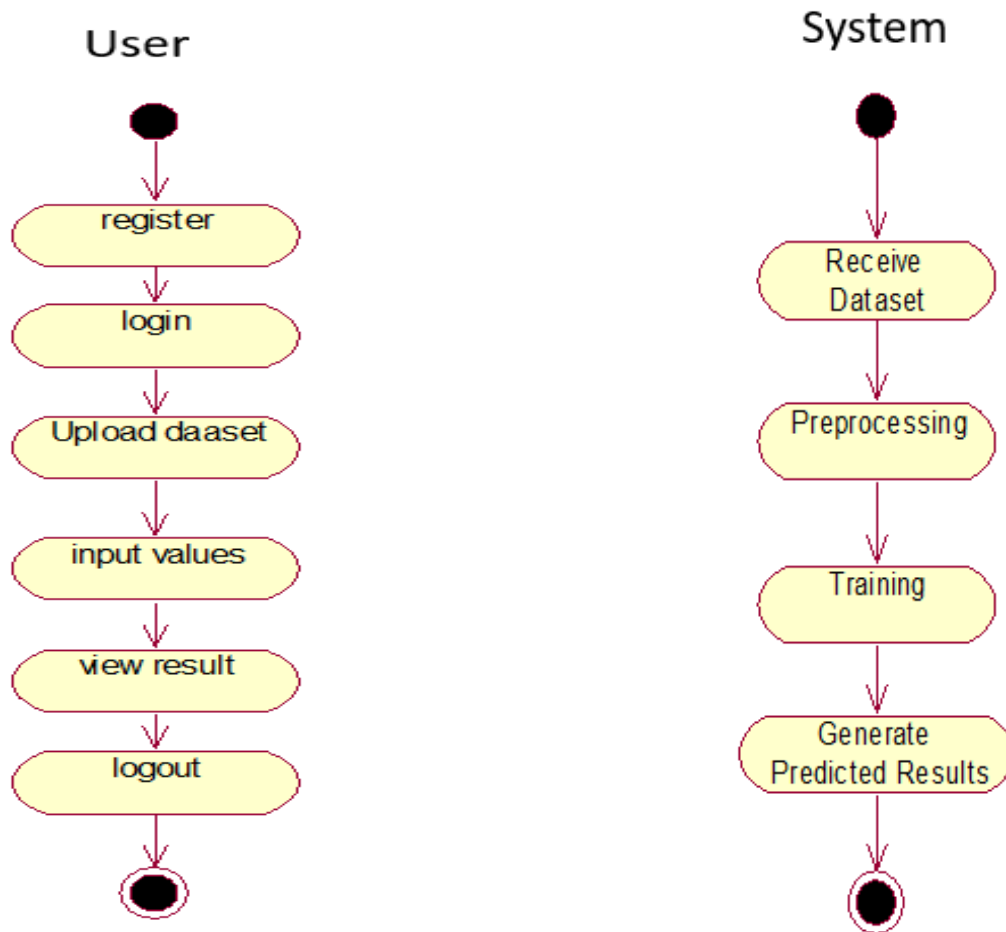
CHAPTER 7

ALGORITHMS

7. ALGORITHMS

7.1 Decision Tree Algorithm:

Decision Tree is a supervised learning method that can be used for classification and regression problems, but it is mainly preferred for solving classification problems. A decision Tree is a tree-structured classifier, where internal nodes stand in for the dataset's characteristics, branches for the decision-making process, and each leaf node for the classification result.

It is a graphical representation of each alternative for solving a problem or making a choice under certain conditions. In the decision tree, there are two nodes, which are the decision node and the leaf node. Decision nodes make decisions with multiple branches, whereas leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed based on features of the given dataset. A decision tree is based on the answer (Yes/No). It further splits the tree into subtrees.
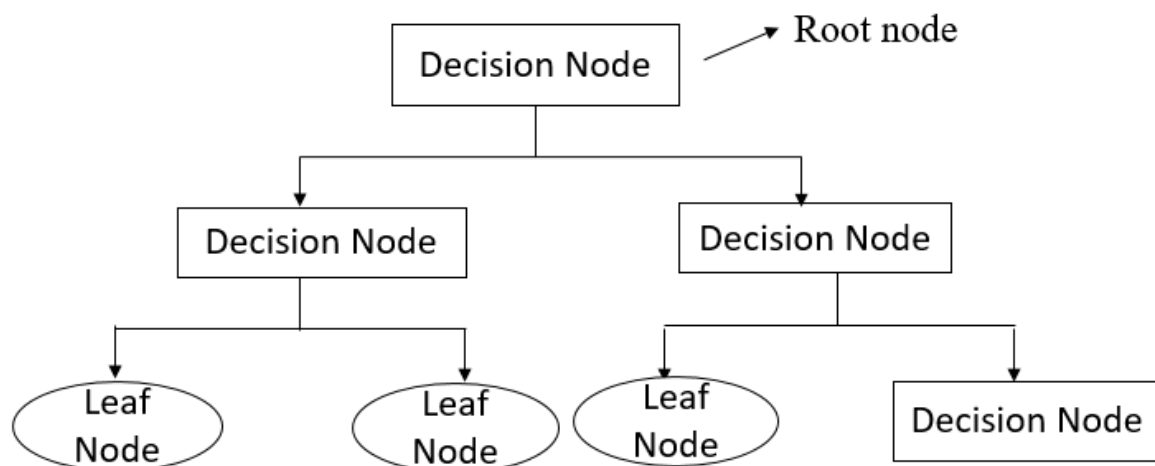
Decision Tree Working Structure:

Steps involved in decision tree algorithm:

Step-1: The root node of the tree, which holds the whole dataset, should be the start, suggests S.

Step-2: Utilize the attribute selection measure to identify the best attribute selection measure (ASM).

Step-3: Subsets of the S that include potential values for the best qualities should be created.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step 3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node a leaf node.

7.2 Artificial Neural Networks:

The term "ANN" is derived from biological neural networks that develop the structure of a human brain. Similar to the human brain, which has neurons interconnected to one another, artificial neural networks (ANN) also have neurons that are coupled to one another in different layers of the networks. Those neurons are called nodes. The three layers of artificial neural networks are as follows:

Input Layer:

It accepts inputs in a variety of forms that the programmer supplies, as the name implies.

Hidden Layer:

The hidden layer presents in between the input and output layers. It does all the computations necessary to uncover patterns and buried features.

Output Layer:

The input goes through a series of transformations using the hidden layer, resulting in output conveyed using this layer.
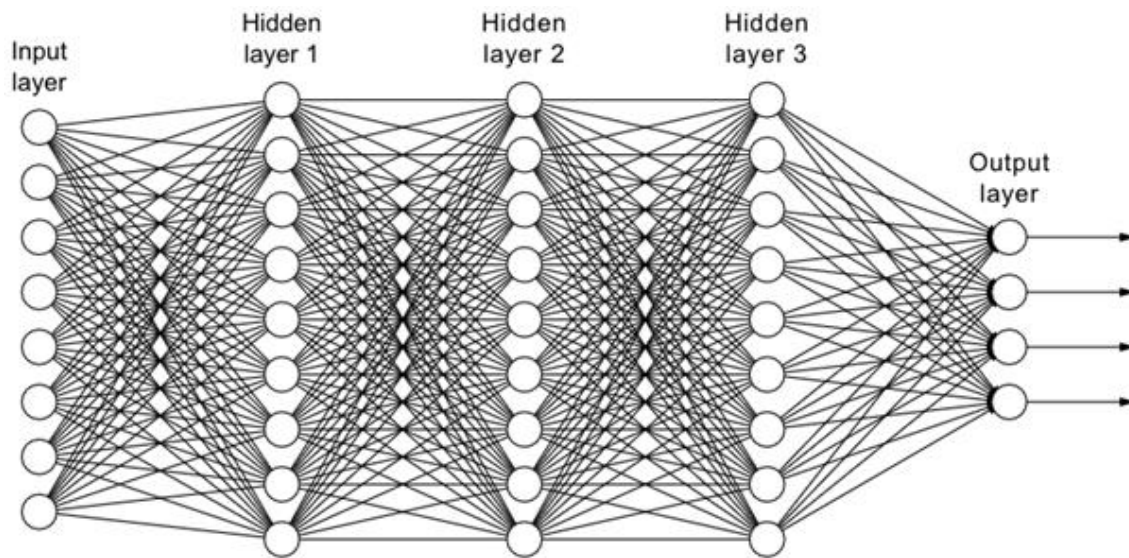
Artificial Neural Networks Working Structure:

Step-1: In the first step, involves passing input units, or data with weights attached, to the hidden layer. We can have many hidden layers. Inputs x_1, x_2, x_3, ….x_n is passed.

Step-2: There are neurons in every hidden layer. Each neuron is coupled to every input.

Step-3: After passing on the inputs, all the computation is performed in the hidden layer.

Step-4: The whole process described in step 3 is performed in each hidden layer. After going through each hidden layer, we move to the last layer, i.e., the output layer, which gives the final result. The preceding procedure is referred to as forwarding propagation.

Step-5: The error, or the difference between the actual and expected output, is determined after receiving the predictions from the output layer.

Step-6: If the error is significant, then the steps are taken to minimize the error, and for the same purpose, backpropagation is performed.

7.3 Naïve Bayes Algorithm:

The naïve Bayes algorithm is a supervised learning algorithm that is based on the Bayes theorem and it is used for solving classification problems. It is mostly applied in text classification that includes a high-dimensional training dataset.

Bayes theorem is one of the most simple and effective classification algorithms, which aids in the development of rapid machine learning methods capable of making prompt predictions It is a probabilistic classifier that predicts based on an object's probability.

The formula for the Bayes theorem is given as:

$P(A|B)=(P(B|A)P(A))/(P(B))$ ……. (1)

Where,

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

$P(B|A)$ is likelihood probability: Probability of the evidence given that the chance of a hypothesis is actual.

$P(A)$ is the prior probability: Probability of hypothesis before observing the evidence.

$P(B)$ is the marginal probability: the probability of evidence.

Naïve Bayes Working Structure:

Steps involved in Naïve Bayes algorithm:

Step-1: Convert the given dataset into frequency tables.

Step-2: Generate a likelihood table by finding the probabilities of provided features.

Step-3: Now, use the Bayes theorem to calculate the posterior probability.

Example for Naïve Bayes

Gender	Heart_Disease	Hyper_Tension
Male	0	1
Female	1	1
Female	1	0
Male	1	0
Female	1	0
Male	1	1
Male	0	1
Female	0	1
Female	0	1
Female	0	1
Male	1	0

For Heart_Disease:

P (Heart_Disease = yes) = $\frac{6}{11}$ = 0.54

P (Heart_Disease = no) = $\frac{5}{11}$ = 0.45

Gender	Yes	No
Male	$\frac{3}{5}$	$\frac{2}{5}$
Female	$\frac{3}{6}$	$\frac{3}{6}$

Table 7.2

$$V_{NB}(yes) = 0.54 * (0.6) * (0.5) = 0.162$$

$$V_{NB}(no) = 0.45 * (0.4) * (0.5) = 0.09$$

$$V_{NB}(yes) = \frac{0.162}{0.162 + 0.09} = 0.642$$

$$V_{NB}(no) = \frac{0.09}{0.162 + 0.09} = 0.357$$

Probability of having stroke is high

For Hyper_Tension:

P (Hyper_Tension = yes) = $\frac{7}{11}$ = 0.63

P (Hyper_Tension = no) = $\frac{4}{11}$ = 0.36

Gender	Yes	No
Male	$\frac{3}{5}$	$\frac{2}{5}$
Female	$\frac{4}{6}$	$\frac{2}{6}$

Table 7.3

$$V_{NB}(yes) = 0.63 * (0.6) * (0.66) = 0.2491$$

$$V_{NB}(no) = 0.36 * (0.4) * (0.33) = 0.04752$$

$$V_{NB}(yes) = \frac{0.2491}{0.2491 + 0.04752} = 0.863$$

$$V_{NB}(no) = \frac{0.04752}{0.2491 + .004752} = 0.1647$$

Probability of having stroke high.

CHAPTER 8

SAMPLE CODE

8. SAMPLE CODE

app.py

#Importing the Libraries

```python
import pickle

import joblib

import numpy as np

from flask import Flask, render_template, request

import sqlite3

app=Flask(_name_)


@app.route('/')

def signup() :

        name = request.args.get('username','')

        dob = request.args.get('DOB','')

        sex = request.args.get('Sex','')

        email = request.args.get('email','')

        martial = request.args.get('martial','')

        password = request.args.get('psw','')

        con = sqlite3.connect('signup.db')

        cur = con.cursor()

        cur.execute("insert into `accounts` (`name`, `dob`, `sex`, `email`, `martial`, `password`)
VALUES (?, ?, ?, ?, ?, ?, ?)", (name,dob,sex,email,martial,password))

        con.commit()

        con.close()

        return render_temlate("login.html")

@app.route("/signin")

def signin() :

        mail1 = request.args.get('uname','')
```

```
        password1 = request.args.get('psw','')

        con = sqlite3.connect('signup.db')

        cur = con.cursor()

        cur.execute("select `email`, `password` from accounts where `email` = ? AND
`password` =?",(mail1,password1,))

        data = cur.fetchone()

        if data == None :

                return render_template("login.html")

        elif mail1 == data[0] and password1 == data[1] :

                return render_template("home.html")

        else :

                return render_template("login.html")


@app.route('/register')
def register() :

        return  render_template("register.html")


@app.route('/login')
def login() :

        return render_template("home.html")


@app.route('/index')
def index() :

        return render_template("home.html")
@app.route("/about", methods=['GET'])
def about() :

        return render_template("about.html")
@app.route("/result", methods=['POST', 'GET'])
def result() :

                gender = int(request.from['gender'])
```

```
age = int(request.from['age'])

hypertension = int(request.from['hypertension'])

heart_disease = int(request.from['heart_disease'])

ever_married = int(request.from['ever_married'])

work_type = int(request.from['work_type'])

Residence_type = int(request.from['Residence_type'])

avg_glucose_level = int(request.from['avg_glucose_level'])

bmi = float(request.from['bmi'])

smoking_status = int(request.from['smoking_status'])


x = np.aray([gender, age, hypertension, heart_disease, ever_married, work_type,
Residence_ type, avg_glucose_level, bmi, smoking_status]).reshape(1, -1)

scaler_path = "models/scaler.pkl"

scaler = None

with open(scaler_path, 'rb') as scaler_file:

        scaler = pickle.load(scaler_file)

x = scaler.transform(x)

model_path = "models/model.sav"

dt = joblib.load(model_path)

y_pred = dt.predict(x)

print(y_pred)

if y_pred == 0:

        return render_template('nostroke.html')

else:

        return render_template('stroke.html')

if_name_ == "_main_":

    app.run(debug=True, port=5000)
```
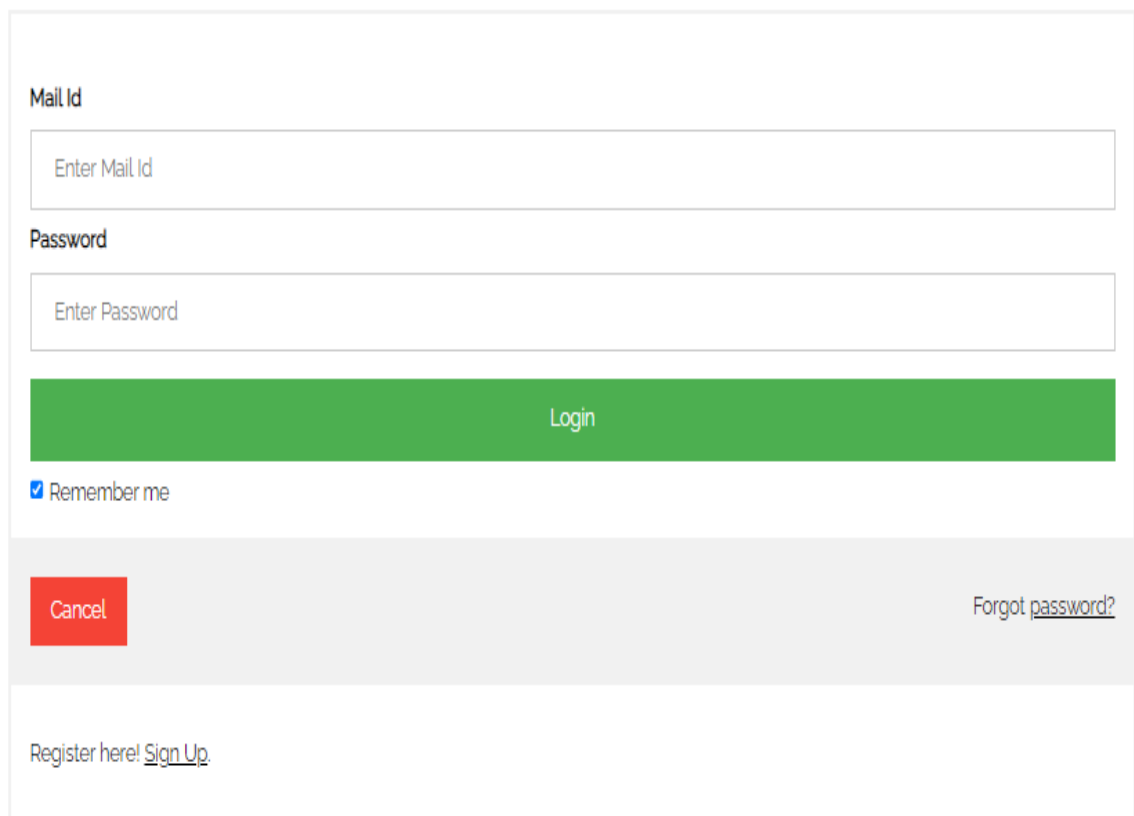
CHAPTER 9
OUTPUT SCREENS

9. OUTPUT SCREENS

Login to your account

Mail Id

Enter Mail Id

Password

Enter Password

Login

☑ Remember me

Cancel   Forgot password?

Register here! Sign Up.

Fig 9.1: Login Page

Register

Please fill in this form to create an account.

Name

Amrutha

Date of Birth

15/08/2000

Sex

female

Contact Number

8075698521

Email

amrutha@gmail.com

Martial Status

no

Password

......

By creating an account you agree to our Terms & Privacy.

Register

Fig 9.2: Registration page
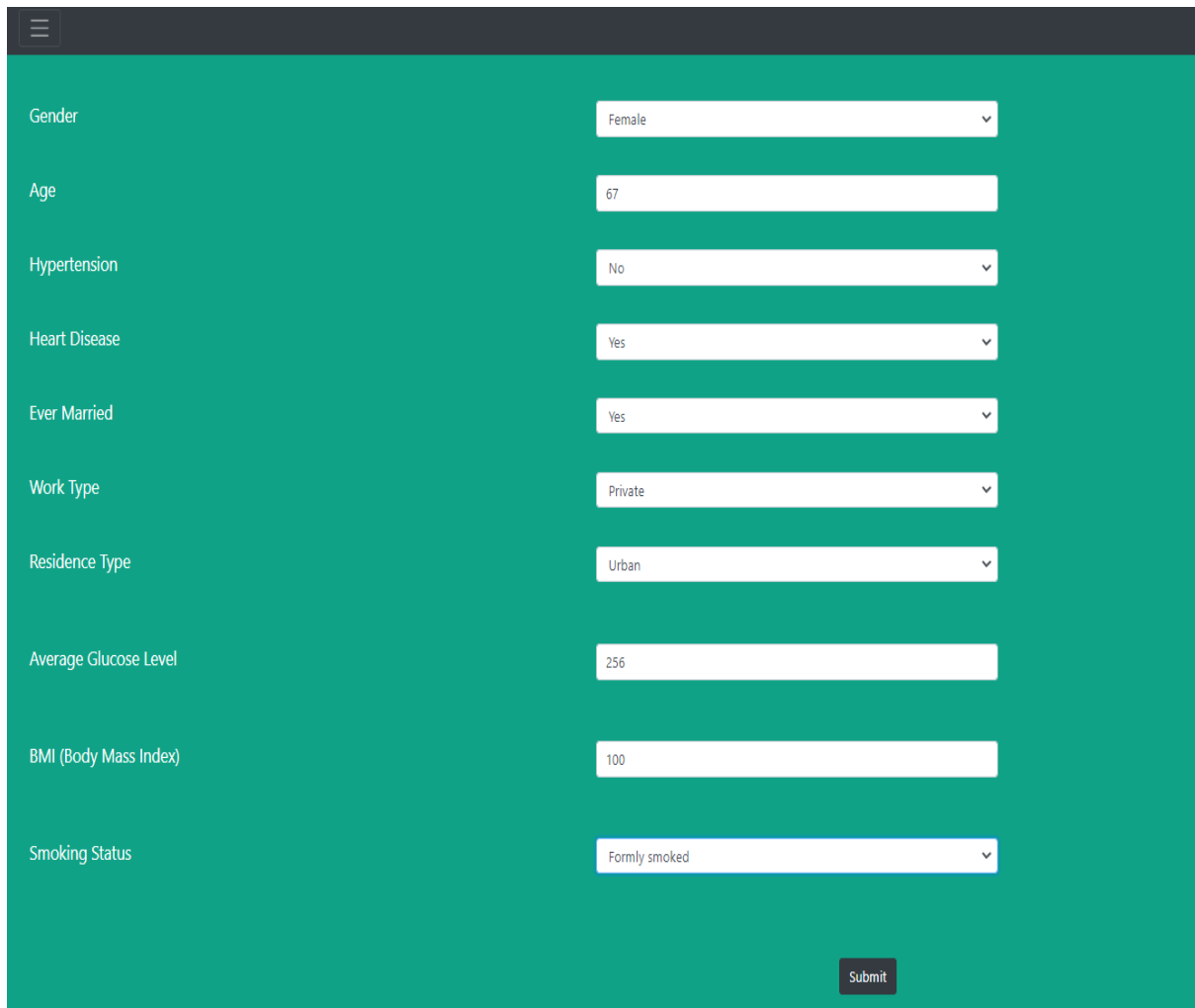
Fig 9.3: Login with registered details

Fig 9.4: Enter values for predicting
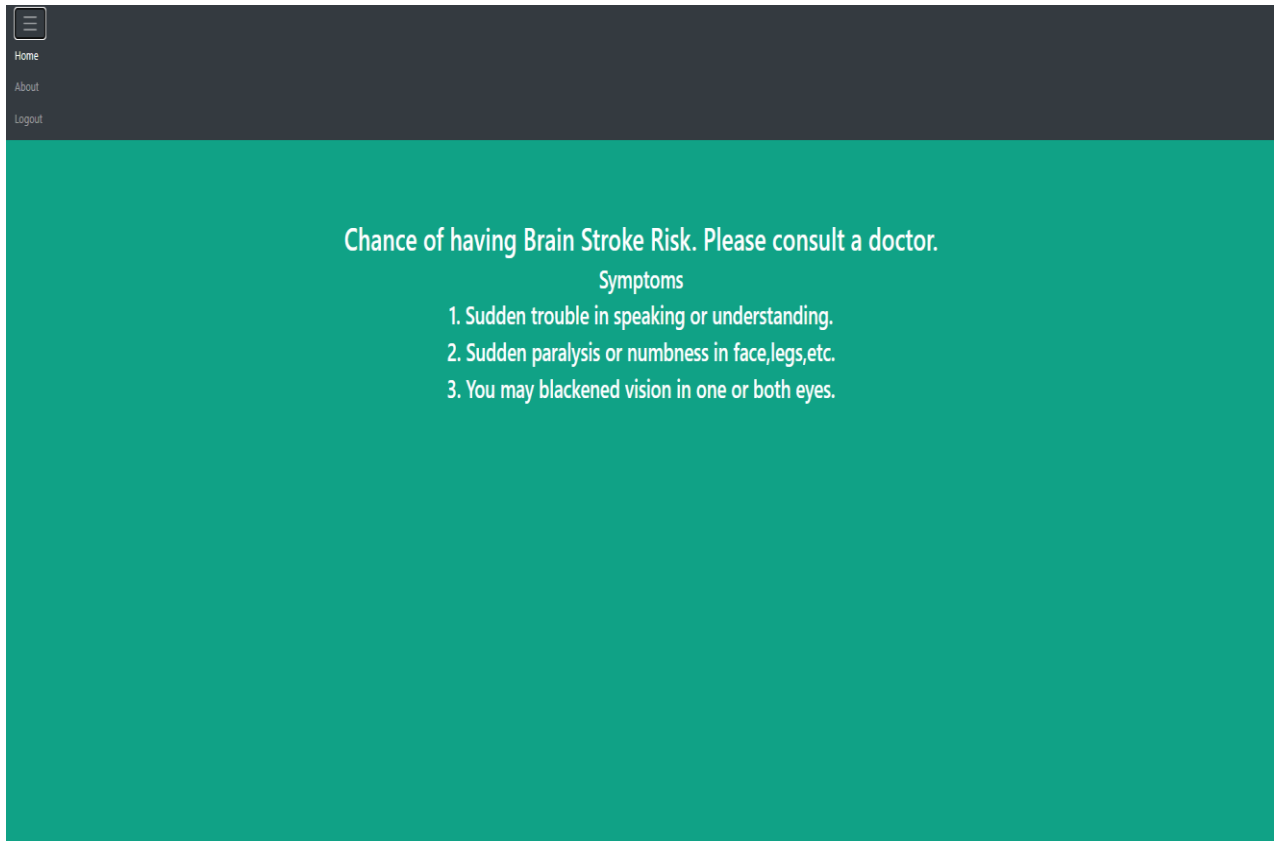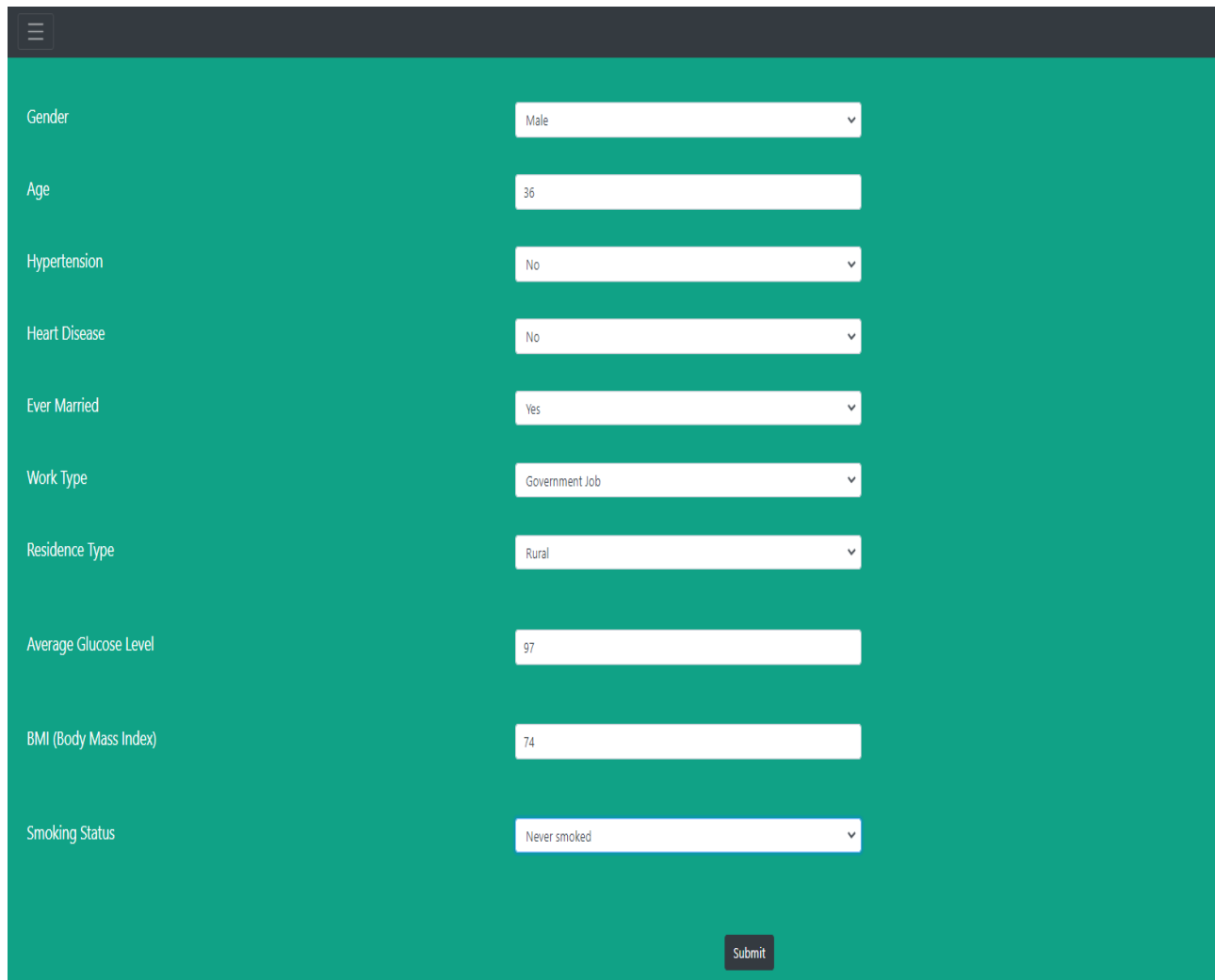
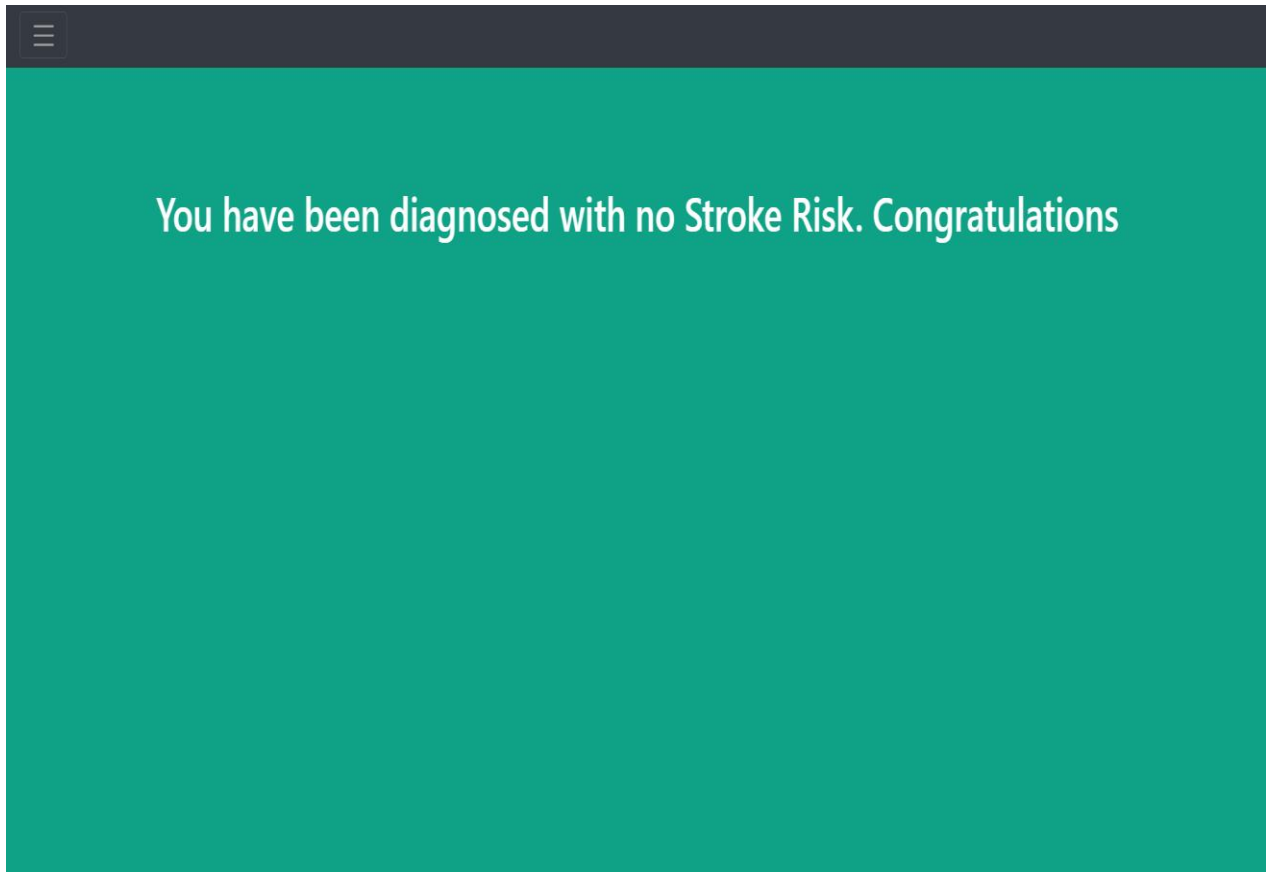Fig 9.5: Sample output prediction of 9.4

Fig 9.6: Enter value for predicting

Fig 9.7: Sample output prediction for 9.6
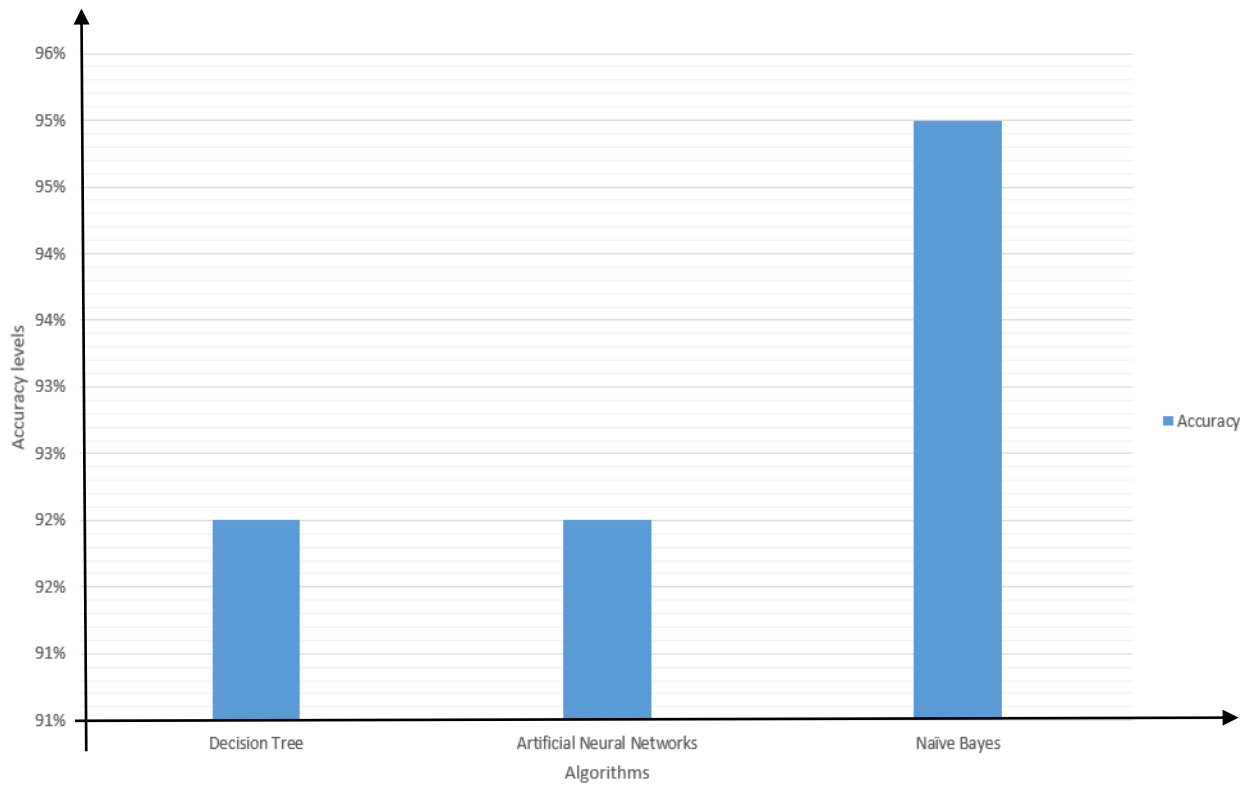
COMPARISON GRAPH:

Fig 9.8: Comparison Graph

CHAPTER 10
SYSTEM TESTING

10. SYSTEM TESTING

Testing is done to detect errors. Testing is looking for flaws or vulnerabilities in a piece of work. It offers a technique to test if parts, subassemblies, or a final product are functioning. It is the process of testing software to make it satisfies user exceptions and needs and does not fail unacceptably. Tests come in a variety of forms. Every test type responds to a particular testing need.

Unit Testing:

Although it is not usual for coding and unit testing to be carried out as two separate stages, unit testing is typically undertaken as part of a combined code and unit test phase of the software lifecycle.

- Assess your approach and plan.
- Manual field testing will be done, and comprehensive functional tests will be prepared.

System Testing:

System testing examines the structure and operation of the system to confirm its general functionality. Independent of the development team, system testing is carried out to evaluate the effectiveness of the system. System testing is done across the board using either functional requirement specifications, system requirement specifications, or both.

Integrating Testing:

Software integration testing reduces errors brought on by interface flaws by incrementally integrating testing two or more integrated software components on a single platform. The goal of an integration test is that components or software applications, such as those found in a software system or on a higher level, software applications used by a full firm, work together flawlessly.

Acceptance Testing:

The end user must actively participate in user acceptability testing, a crucial stage of any project. Additionally, it guarantees that the system satisfies the necessary functional standards.

TEST CASES

S.NO	Test Cases	I/O	Expected O/P	Actual O/P	Pass/Fail
1	Login details are not filled in.	A warning statement is to be displayed.	Users cannot get access to the next page.	Users cannot get access to the next page.	Pass
2	The prediction should be accurate.	Input features are provided.	The output should be whether a patient suffers from a brain stroke.	Predicts successfully.	Pass
3	Password should be firm, and it should satisfy particular instructions.	Enter the password by helping the instructions.	The Given password should not be accepted.	Password not accepted.	Pass

Table 10.1 Test Cases

Brain Stroke Prediction

Login to your account

Mail Id

Enter Mail Id

Password

Enter Password

rupaamruthach@gmail.com

roopa@gmail.com

Rupa

rupa@gmail.com

teja@gmail.com

amrutha@gmail.com

...ase fill out this field.

Login

☑ Remember me

Cancel

Forgot password?

Register here! Sign U...

Fig 10.1 Error Page 1

Please fill in this form to create an account.

Name

rupa

Date of Birth

16/05/2003

Sex

female

Contact Number

8074256292

Email

rupaamruthach@gmail.com

Martial Status

no

Password

····

⚠ Please match the requested format.
Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters

Already have an account? Sign in.

Fig 10.2 Error Page 2

CHAPTER 11
CONCLUSION

11. CONCLUSION

Diagnostic applications of machine learning are growing in the medical industry. This thesis aims to predict stroke whether a person has a risk of getting a stroke or not by utilizing a pre-processed dataset. This can be achieved by employing three machine learning techniques, including Decision Tree (DT), Naïve Bayes (NB), and Artificial Neural Networks (ANN). By utilizing a few user-provided inputs and forecasting accuracy, the system assists in the cost-effective and efficient prediction of brain stroke. A comparison of each approach is also included. Naïve Bayes Classification, which was selected, performs the best overall, with a 95% accuracy rate. The suggested method is the backbone of the stroke patient healthcare system. This study may be expanded to identify the probability of stroke by gathering a dataset made up of images, such as brain CT scans.

CHAPTER 12
REFERENCES

12. REFERENCES

[1] M. U. Emon, M .S. Keya, T. I. Meghla, M. M. Rahman, M. S. A. Mamun, and M. S. Kaiser, "*Performance Analysis of Machine Learning Approaches in Stroke Prediction*," 2020 4th International Conference on Electronics, Communication, and Aerospace Technology (ICECA), 2020, pp. 1464-1464.

[2] N. S. Adi, R. Frahany, R.Ghina, and H. Napitupulu, "*Stroke Risk Prediction Model Using Machine Learning*," 2021 International Conference on Artificial Intelligence and Big Data Analytics, 2021, pp.56-60.

[3] A. Tursynova, B. Omarov, K. Shuketayeva, and M. Smagul, "*Artificial Intelligence in Stroke Imagining*," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 41-45.

[4] Amini L, Azarpazhouh R, Farzadfar MT, Mousavi SA, Jazaieri F, Khoravh F, Norouzi R, Toghianfar N. "*Prediction and Control of Stroke by Data Mining.*" International journal of preventive medicine 4. Suppl 2 (2013): S245.

[5] R. K. Kavitha, W. Jaisingh and S. R. Sujithra, "*Applying Machine Learning Techniques for Stroke Prediction in Patients*," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021, pp. 1-4.

[6] Gangavarapu Sailasya and Gorli L Aruna Kumari, "*Analyzing the Performance of Stroke Prediction using ML Classification Algorithms*," International Journal of Advanced Computer Science and Applications (IHACSA), 12(6).

[7] T. Kanasadub, S. Thammaboosadee, S. Kiattisin, and C. Jalayodeja, "*Stroke risk Prediction model based on demographic data*," 2015 8th Biomedical Engineering International Conference (BMEiCON), 2015, pp. 1-3.

[8] C. –L. Chin et al. l. "*An automated early ischemic stroke detection system using CNN deep learning algorithm,*" 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST), 2017, pp. 368-372.

[9] M. Monterio et al. "*Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients,*" in IEEE/ACM Transactions on Computational Biology

and Bioinformatics, vol. 15, no. 6, pp. 1953-1959, 1 Nov.-Dec. 2018, doi: 10.1109/TCBB.201802811471.

[10] E. Zamsa, "*Medical software user interfaces, stroke MD application design,*" 2015 E-Health and Bioengineering Conference (EHB), 2015, pp. 1-4.

[11] Badriyah, Tessy, et al. "*improving stroke diagnosis accuracy using hyperparameter optimized deep learning.*" International Journal of Advances in Intelligent Informatics 5.3 (2019): 256-272.

[12] M. A. Hakim, M. Z. Hasa, M. M. Alam, M. M. Hasan, and M. N. Huda, *"An Efficient Modified Bagging Method for Early Prediction of Brain Stroke,"* 2019 International Conference on Computer, Communication, Chemical, Materials, and Electronic Engineering (IC4ME2), 2019, pp. 1-4.