A

Mini Project

On

**DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS**

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

CH.AJAY KUMAR (207R1A0571)

S.SRIKANTH (207R1A05B3)

Under the Guidance of

**M.MADHUSUDHAN**

(Associate Professor)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## CMR TECHNICAL CAMPUS UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

**2020-2024**

### **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled “**DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS”** being submitted by **CH.AJAY KUMAR (207R1A0571) & S.SRIKANTH (207R1A05B3)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

|  |  |
| --- | --- |
| **M. Madhusudhan** | **Dr. A. Raji Reddy** |
| (Associate Professor)  INTERNAL GUIDE | DIRECTOR |
| **Dr. K. Srujan Raju**  HOD | **EXTERNAL EXAMINER** |
| **Submitted for viva voce Examination held on \_\_\_\_\_\_\_\_\_\_** |  |

## ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to

my guide **M. Madhusudhan,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G. Vinesh Shanker, Dr. J. Narasimharao , Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout

the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**CH.AJAYKUMAR ( 207R1A0571)**

**S.SRIKANTH (207R1A05B3)**

# **ABSTRACT**

A stroke is a medical condition in which poor blood flow to the brain results in cell death. It is now a day a leading cause of death all over the world. Several risk factors believe to be related to the cause of 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 this work, we have used four machine learning algorithms to detect the type of stroke that can possibly occur or occurred form a person’s physical state and medical report data. We have collected a good number of entries from the hospitals and use them to solve our problem. The classification result shows that the result is satisfactory and can be used in real time medical report. We believe that machine learning algorithms can help better understanding of diseases and can be a good healthcare companion. Index Terms—Stroke, machine learning, WEKA, Naive Bayes, J48, k-NN, Random Forest.

**i**

**LIST OF FIGURES**

## FIGURE NO FIGURE NAME PAGE NO

Figure 3.1 Project Architecture for Detection of stroke

disease using machine learning algorithms 6

Figure 3.2 Use Case Diagram for Detection of stroke

disease using machine learning algorithms 7

Figure 3.3 Class Diagram for Detection of stroke

disease using machine learning algorithms 8

Figure 3.4 Sequence Diagram for Detection of stroke

disease using machine learning algorithms 9

Figure 3.5 Collaboration Diagram for Detection of stroke

disease using machine learning algorithms 11

Figure 3.6 Activity Diagram for Detection of stroke

disease using machine learning algorithms 12

**ii**

SCREENSHOT 5.11 X-axis represent names & Y-axis represent accuracy 32

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO** | **LIST OF SCREENSHOTS**  **SCREENSHOT NAME** | **PAGE NO** |
| SCREENSHOT 5.1 | Click on ‘Upload Stroke Dataset’ to upload  dataset. | 22 |
| SCREENSHOT 5.2 | Click on ‘Open’ button to load dataset. | 23 |
| SCREENSHOT 5.3 | Click on ‘Dataset Preprocessing & Features Selection’ button to process dataset. | 24 |
| SCREENSHOT 5.4 | Graph x-axis represents 0 & 1,y-axis represents no | 25 |
| SCREENSHOT 5.5 | Click on ‘Train Navie Bayes Algorithm’ button | 26 |
| SCREENSHOT 5.6 | Click on ‘Train J48 Algorithm’ button | 27 |
| SCREENSHOT 5.7 | “Run KNN Algorithm” button | 28 |
| SCREENSHOT 5.8 | “Run Random Forest Algorithm” button | 29 |
| SCREENSHOT 5.9 | “Run KNN Algorithm” button | 30 |
| SCREENSHOT 5.10 | Click on “Comparison Graph” button | 31 |

**iii**

**TABLE OF CONTENTS**

**ABSTRACT i**

**LIST OF FIGURES ii**

**LIST OF SCREENSHOTS iii**

1. **INTRODUCTION**   **1**

1.1 PROJECT SCOPE 1 1.2 PROJECT PURPOSE 1

1.3 PROJECT FEATURES 2 2

**2. SYSTEM ANALYSIS 3**

2.1 PROBLEM DEFINITION 3

2.2 EXISTING SYSTEM 3

2.2.1 LIMITATIONS OF THE EXISTING SYSTEM 4

2.3 PROPOSED SYSTEM 4

2.3.1 ADVANTAGES OF PROPOSED SYSTEM 4

2.4 HARDWARE & SOFTWARE REQUIREMENTS 5

2.4.1 HARDWARE REQUIREMENTS 5

2.4.2 SOFTWARE REQUIREMENTS 5

**3. ARCHITECTURE 6**

3.1 PROJECT ARCHITECTURE 6

3.2 USE CASE DIAGRAM 7

3.3 CLASS DIAGRAM 8

3.4 SEQUENCE DIAGRAM 9

3.5 COLLABORATION DIAGRAM 11

3.6 ACTIVITY DIAGRAM 12

**4. IMPLEMENTATION**   **13**

4.1 SAMPLE CODE 13

**5. SCREENSHOTS 22**

**6. TESTING 33**

6.1 INTRODUCTION TO TESTING 33

6.2 TYPES OF TESTING 33

6.2.1 UNIT TESTING 33

6.2.2 INTEGRATION TESTING 34

6.2.3 FUNCTIONAL TESTING 34

6.2.4 SYSTEM TESTING 34

6.2.5 WHITE BOX TESTING 34

6.2.6 BLACK BOX TESTING 35

6.2.7 ACCEPTANCE TESTING 36

**7. CONCLUSION & FUTURE SCOPE 37**

7.1 PROJECT CONCLUSION 37

7.2 FUTURE SCOPE 37

**8. REFERENCES 38**

8.1 REFERENCES 38

8.2 GITHUB LINK 39

**1.INTRODUCTION**

## 

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

## 1. INTRODUCTION

**1.1** **PROJECT SCOPE**

The project scope for detecting stroke disease using machine learning algorithms involves developing a robust and accurate predictive model to identify potential stroke cases. This model will be trained on relevant medical data, including patient demographics, medical history, lifestyle factors, and specific health indicators. The algorithm will analyze and process this data to detect patterns and correlations associated with strokes. The goal is to create a tool that can effectively predict the likelihood of an individual experiencing a stroke based on the provided parameters. The model will require validation and fine-tuning to ensure its accuracy and reliability. Additionally, the project will explore the integration of real-time data input and user-friendly interfaces to enhance usability for both medical professionals and the general public. The ultimate aim is to contribute to early detection and proactive intervention in stroke cases, potentially saving lives and improving overall healthcare outcomes.

**1.2 PROJECT PURPOSE**

The purpose of the "Detection of Stroke Disease Using Machine Learning Algorithms" project is to develop a robust and efficient system that can accurately identify signs and symptoms associated with strokes in individuals. Stroke is a critical medical condition that requires prompt diagnosis and treatment to prevent severe damage and save lives. By utilizing machine learning algorithms, this project aims to analyze various patient data such as medical history, vital signs, imaging scans, and other relevant information to create a predictive model. This model will aid in early detection and provide medical professionals with valuable insights for timely intervention and appropriate patient care.

CMRTC 1

**1.3 PROJECT FEATURES**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

The project "Detection of Stroke Disease Using Machine Learning Algorithms" aims to develop an automated system for early stroke detection. It involves collecting and preprocessing medical data, including patient demographics, medical history, and diagnostic tests. Feature selection methods are applied to extract relevant features such as blood pressure, cholesterol levels, and lifestyle habits. Machine learning algorithms like support vector machines, random forests, and neural networks are employed to build predictive models. The models are trained using labeled data to predict the likelihood of a stroke based on the selected features. Evaluation metrics like accuracy, sensitivity, and specificity are utilized to assess model performance. Cross-validation techniques are employed to ensure robustness and generalizability of the models. The system aims to assist healthcare professionals in timely stroke detection, aiding in better patient care and outcomes. Ongoing research and iterative model refinement are crucial to enhance the accuracy and efficiency of stroke detection using machine learning.

CMRTC 2

**2.SYSTEM ANALYSIS**

**2. SYSTEM ANALYSIS**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

The system for detecting stroke using machine learning algorithms involves data collection, preprocessing, and feature extraction from medical records and imaging. Machine learning models like neural networks or support vector machines are trained on these features to predict stroke presence based on patterns and risk factors. Evaluation and validation of the model's performance are crucial, comparing predictions against ground truth data. This system aims to enhance stroke diagnosis and potentially improve patient outcomes through early and accurate detection using automated techniques.

### **2.1 PROBLEM DEFINITION**

Detection of stroke disease using machine learning algorithms involves developing a computational model that can accurately identify signs and patterns indicative of stroke based on various input features such as medical records, imaging data, and clinical parameters..

**2.2** **EXISTING SYSTEM**

The detection of stroke disease using machine learning algorithms involves analyzing medical data, such as brain imaging and patient health records, to identify patterns indicative of stroke risk. Various machine learning techniques, including neural networks, support vector machines, and decision trees, are utilized to process and interpret this data. These algorithms learn from labeled examples, distinguishing healthy and stroke-affected individuals based on specific features. The system enhances accuracy through continuous learning and refinement, ultimately aiding healthcare professionals in early stroke detection, timely intervention, and improved patient outcomes.

CMRTC 3

**2.2.1 LIMITATIONS OF EXISTING SYSTEM**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

Following are the disadvantages of existing system:

* Limited Data Availability
* Difficulty in Interpretation
* Bias in Data
* Overfitting
* Lack of Transparency

### **2.3 PROPOSED SYSTEM**

Our proposed system for detecting stroke disease leverages machine learning algorithms to analyze medical data and accurately identify stroke risk. We utilize a dataset of patient information, including demographics, medical history, and lifestyle factors. Preprocessing involves data cleaning, feature selection, and normalization. We employ algorithms such as Support Vector Machines (SVM), Random Forest, and Neural Networks for predictive modeling. Training the model involves optimizing parameters and validating the results using cross-validation techniques. The system aims to achieve high accuracy, sensitivity, and specificity to ensure reliable stroke detection. Continuous refinement and updates to the model enhance its performance and adaptability to evolving medical knowledge.

#### **2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM**

* Accuracy
* Cost-Effective
* Increased Efficiency
* Improved Patient Outcomes

CMRTC 4

**2.4 HARDWARE & SOFTWARE REQUIREMENTS**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

**2.4.1 HARDWARE REQUIREMENTS:**

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

|  |  |
| --- | --- |
| System Processor = | Intel Dual Core i3  and above |
| Hard Disk = | 40GB and above |
| Ram = | 8GB and above |
| Input devices = | Keyboard, mouse. |

**2.4.1 SOFTWARE REQUIREMENTS:**

Software Requirements specifies the logical characteristics of each interface and software components of the system.

|  |  |
| --- | --- |
| Operating Systems = | Window 8 or above |
| Coding Languages = | Python 3.7 |

CMRTC 5

**3. ARCHITECTURE**

**3. ARCHITECTURE**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

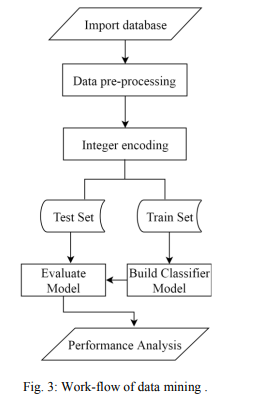
### **3.1 PROJECT ARCHITECTURE**

The following steps have been performed for stroke detection in WEKA: • Data pre-processing and visualization .

• Attribute selection

• Test set and train set splitting

• Classification using different algorithms



The work-flow of data mining is given in Fig. 3

CMRTC 6

**3.2 USE CASE DIAGRAM**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**Figure 3.2:** Use Case Diagram for Detection of stroke disease using machine learning algorithms

CMRTC 7

**3.3 CLASS DIAGRAM**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**Figure 3.3**: Class Diagram for Detection of stroke disease using machine learning algorithms

CMRTC 8

**3.4 SEQUENCE DIAGRAM**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

CMRTC 9



DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

**Figure 3.4:** Sequence Diagram for Detection of stroke disease using machine learning algorithms.

CMRTC 10

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

#### **3.5 COLLABORATION DIAGRAM**

#### A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software [objects](https://www.techtarget.com/searchapparchitecture/definition/object) in the Unified Modeling Language ([UML](https://www.techtarget.com/searchsoftwarequality/definition/Unified-Modeling-Language)). Developers can use these diagrams to portray the dynamic behavior of a particular [use case](https://www.techtarget.com/searchsoftwarequality/definition/use-case) and define the role of each object.



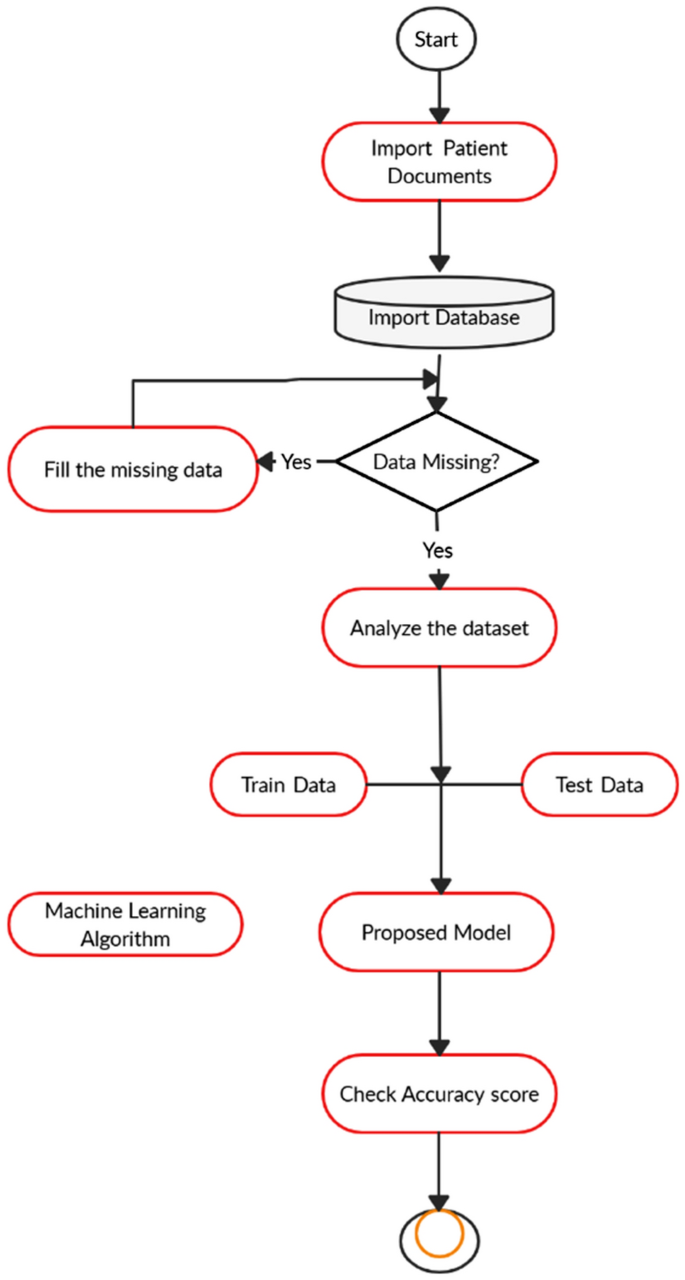
**Figure 3.5:** Collaboration Diagram for Detection of stroke disease using machine learning algorithms

CMRTC 11

#### **3.6 ACTIVITY DIAGRAM**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data .



**Figure 3.6:** Activity Diagram for Detection of stroke disease using machine learning algorithms

CMRTC 12

**4.IMPLEMENTATION**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

#### **4.1 SAMPLE CODE**

from tkinter import \*

import tkinter

from tkinter import filedialog

import numpy as np

from tkinter.filedialog import askopenfilename

from tkinter import simpledialog

import matplotlib.pyplot as plt

import os

import numpy as np

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import seaborn as sns

from sklearn.metrics import confusion\_matrix

from sklearn.naive\_bayes import GaussianNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from keras.utils.np\_utils import to\_categorical

from keras.models import Sequential

from keras.layers import Dense, Dropout, Activation

main = tkinter.Tk()

main.title("Detection of Stroke Disease using Machine Learning Algorithms")

main.geometry("1000x650")

CMRTC 13

global filename, le1,le2,le3,le4,le5, dataset, rf

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

global X, Y

global X\_train, X\_test, y\_train, y\_test

accuracy = []

precision = []

recall = []

fscore = []

def loadDataset():

    global filename, dataset

    text.delete('1.0', END)

    filename = filedialog.askopenfilename(initialdir="Dataset")

    text.insert(END,str(filename)+" loaded\n\n")

    dataset = pd.read\_csv(filename)

    text.insert(END,str(dataset.head()))

def preprocessDataset():

    text.delete('1.0', END)

    global X, Y

    global X\_train, X\_test, y\_train, y\_test

    global dataset, le1,le2,le3,le4,le5

    le1 = LabelEncoder()

    le2 = LabelEncoder()

    le3 = LabelEncoder()

    le4 = LabelEncoder()

    le5 = LabelEncoder()

    dataset.fillna(0, inplace = True)

    dataset['gender'] = pd.Series(le1.fit\_transform(dataset['gender'].astype(str)))

    dataset['ever\_married'] = pd.Series(le2.fit\_transform(dataset['ever\_married'].astype(str)))

dataset['work\_type'] = pd.Series(le3.fit\_transform(dataset['work\_type'].astype(str)))

CMRTC 14

    dataset['Residence\_type'] = pd.Series(le4.fit\_transform(dataset['Residence\_type'].astype(str)))

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

    dataset['smoking\_status'] = pd.Series(le5.fit\_transform(dataset['smoking\_status'].astype(str)))

    text.insert(END,str(dataset.head())+"\n\n")

    text.update\_idletasks()

    label = dataset.groupby('stroke').size()

    dataset = dataset.values

    text.insert(END,"\nTotal attributes before applying features selection: "+str(dataset.shape[1])+"\n\n")

    X = dataset[:,1:dataset.shape[1]-1]

    Y = dataset[:,dataset.shape[1]-1]

    indices = np.arange(X.shape[0])

    np.random.shuffle(indices)

    X = X[indices]

    Y = Y[indices]

    text.insert(END,"\nTotal attributes after applying features selection: "+str(X.shape[1])+"\n\n")

    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

    text.insert(END,"Total records found in dataset : "+str(X.shape[0])+"\n\n")

text.insert(END,"Dataset split for train and test. 80% for training and 20% for testing\n\n")

    text.insert(END,"Total records used to train Machine Learning Algorithms : "+str(X\_train.shape[0])+"\n")

    text.insert(END,"Total records used to test Machine Learning Algorithms : "+str(X\_test.shape[0])+"\n")

    label.plot(kind="bar")

    plt.title("Number of Normal & Stroke Disease Instances in dataset")

    plt.show()

CMRTC 15

def calculateMetrics(predict, testY, algorithm):

    p = precision\_score(testY, predict,average='macro') \* 100

    r = recall\_score(testY, predict,average='macro') \* 100

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

    f = f1\_score(testY, predict,average='macro') \* 100

    a = accuracy\_score(testY,predict)\*100

    text.insert(END,algorithm+' Accuracy  : '+str(a)+"\n")

    text.insert(END,algorithm+' Precision : '+str(p)+"\n")

    text.insert(END,algorithm+' Recall    : '+str(r)+"\n")

    text.insert(END,algorithm+' FScore    : '+str(f)+"\n\n")

    accuracy.append(a)

    precision.append(p)

    recall.append(r)

    fscore.append(f)

text.update\_idletasks()

    LABELS = ['Normal','Stroke']

    conf\_matrix = confusion\_matrix(testY, predict)

    plt.figure(figsize =(6, 6))

    ax = sns.heatmap(conf\_matrix, xticklabels = LABELS, yticklabels = LABELS, annot = True, cmap="viridis" ,fmt ="g");

    ax.set\_ylim([0,2])

    plt.title(algorithm+" Confusion matrix")

    plt.ylabel('True class')

    plt.xlabel('Predicted class')

    plt.show()

def trainNaiveBayes():

    global X\_train, X\_test, y\_train, y\_test

    text.delete('1.0', END)

    cls = GaussianNB()

    cls.fit(X\_train, y\_train)

    predict = cls.predict(X\_test)

    calculateMetrics(predict, y\_test, "Naive Bayes")

CMRTC 16

def trainDT():

    global X\_train, X\_test, y\_train, y\_test

    cls = DecisionTreeClassifier()

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

    cls.fit(X\_train, y\_train)

    predict = cls.predict(X\_test)

    calculateMetrics(predict, y\_test, "J48 Algorithm")

def trainKNN():

    global X\_train, X\_test, y\_train, y\_test

    cls = KNeighborsClassifier(n\_neighbors = 2)

    cls.fit(X\_train, y\_train)

    predict = cls.predict(X\_test)

    calculateMetrics(predict, y\_test, "KNN")

def trainRanfomForest():

    global X\_train, X\_test, y\_train, y\_test, rf

    cls = RandomForestClassifier()

    cls.fit(X\_train, y\_train)

    rf = cls

    predict = cls.predict(X\_test)

    calculateMetrics(predict, y\_test, "Random Forest")

def graph():

    df = pd.DataFrame([['Naive Bayes','Precision',precision[0]],['Naive Bayes','Recall',recall[0]],['Naive Bayes','F1 Score',fscore[0]],['Naive Bayes','Accuracy',accuracy[0]],

         ['J48','Precision',precision[1]],['J48','Recall',recall[1]],['J48','F1 Score',fscore[1]],['J48','Accuracy',accuracy[1]],

                       ['KNN','Precision',precision[2]],['KNN','Recall',recall[2]],['KNN','F1 Score',fscore[2]],['KNN','Accuracy',accuracy[2]],

CMRTC 17

                       ['Random Forest','Precision',precision[3]],['Random Forest','Recall',recall[3]],['Random Forest','F1 Score',fscore[3]],['Random Forest','Accuracy',accuracy[3]],

                       ['ANN','Precision',precision[4]],['ANN','Recall',recall[4]],['ANN','F1 Score',fscore[4]],['ANN','Accuracy',accuracy[4]],

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

                      ],columns=['Parameters','Algorithms','Value'])

    df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')

    plt.show() def trainANN():

    global X, Y

    Y1 = to\_categorical(Y)

    X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(X, Y1, test\_size=0.2)

    ann\_model = Sequential()

    ann\_model.add(Dense(512, input\_shape=(X\_train1.shape[1],)))

    ann\_model.add(Activation('relu'))

    ann\_model.add(Dropout(0.3))

    ann\_model.add(Dense(512))

    ann\_model.add(Activation('relu'))

    ann\_model.add(Dropout(0.3))

    ann\_model.add(Dense(2))

    ann\_model.add(Activation('softmax'))

    ann\_model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

    print(ann\_model.summary())

    acc\_history = ann\_model.fit(X, Y1, epochs=200, validation\_data=(X\_test1, y\_test1))

    print(ann\_model.summary())

    predict = ann\_model.predict(X\_test1)

    predict = np.argmax(predict, axis=1)

    testY = np.argmax(y\_test1, axis=1)

    calculateMetrics(predict, testY, "ANN")

def predict():

CMRTC 18

    text.delete('1.0', END)

    global rf, le1,le2,le3,le4,le5

    testfile = filedialog.askopenfilename(initialdir="Dataset")

    dataset = pd.read\_csv(testfile)

    dataset.fillna(0, inplace = True)

    dataset['gender'] = pd.Series(le1.transform(dataset['gender'].astype(str)))

    dataset['ever\_married'] = pd.Series(le2.transform(dataset['ever\_married'].astype(str)))

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

    dataset['work\_type'] = pd.Series(le3.transform(dataset['work\_type'].astype(str)))

    dataset['Residence\_type'] = pd.Series(le4.transform(dataset['Residence\_type'].astype(str)))

    dataset['smoking\_status'] = pd.Series(le5.transform(dataset['smoking\_status'].astype(str)))

    dataset = dataset.values

dataset = dataset[:,1:dataset.shape[1]]

    predict = rf.predict(dataset)

    print(predict)

    for i in range(len(predict)):

        if predict[i] == 0:

            text.insert(END,"Test Data = "+str(dataset[i])+" PREDICTED AS ====> NO STROKE\n\n")

        if predict[i] == 1:

if predict[i] == 1:

            text.insert(END,"Test Data = "+str(dataset[i])+" PREDICTED AS ====> STROKE\n\n")

font = ('times', 15, 'bold')

title = Label(main, text='Detection of Stroke Disease using Machine Learning Algorithms', justify=LEFT)

title.config(bg='lavender blush', fg='DarkOrchid1')

title.config(font=font)

title.config(height=3, width=120)

CMRTC 19

title.place(x=100,y=5)

title.pack()

font1 = ('times', 12, 'bold')

loadButton = Button(main, text="Upload Stroke Dataset", command=loadDataset)

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

loadButton.place(x=10,y=100)

loadButton.config(font=font1)

preprocessButton = Button(main, text="Dataset Preprocessing & Features Selection", command=preprocessDataset)

preprocessButton.place(x=300,y=100)

preprocessButton.config(font=font1)

nbButton = Button(main, text="Train Naive Bayes Algorithm", command=trainNaiveBayes)

nbButton.place(x=730,y=100)

nbButton.config(font=font1)

dtButton = Button(main, text="Train J48 Algorithm", command=trainDT)

dtButton.place(x=10,y=150)

dtButton.config(font=font1)

knnButton = Button(main, text="Train KNN Algorithm", command=trainKNN)

knnButton.place(x=300,y=150)

knnButton.config(font=font1)

rfButton = Button(main, text="Train Random Forest Algorithm", command=trainRanfomForest)

rfButton.place(x=730,y=150)

rfButton.config(font=font1)

annButton = Button(main, text="Train ANN Algorithm", command=trainANN)

annButton.place(x=10,y=200)

annButton.config(font=font1)

graphButton = Button(main, text="Comparison Graph", command=graph)

graphButton.place(x=300,y=200)

graphButton.config(font=font1)

predictButton = Button(main, text="Predict Disease on Test Data", command=predict)

CMRTC 20

predictButton.place(x=730,y=200)

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

predictButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=160)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=250)

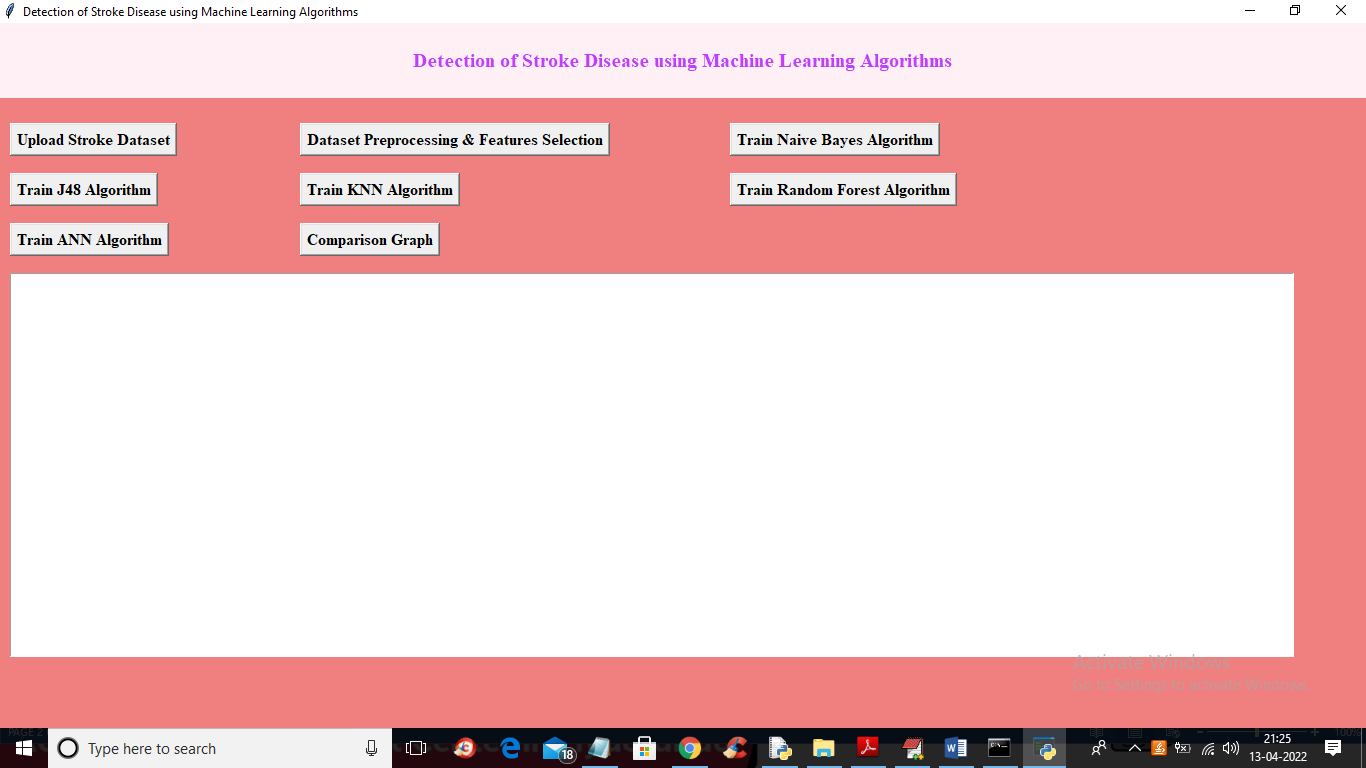
text.config(font=font1)

main.config(bg='light coral')

main.mainloop()

CMRTC 21

**5. SCREENSHOTS**

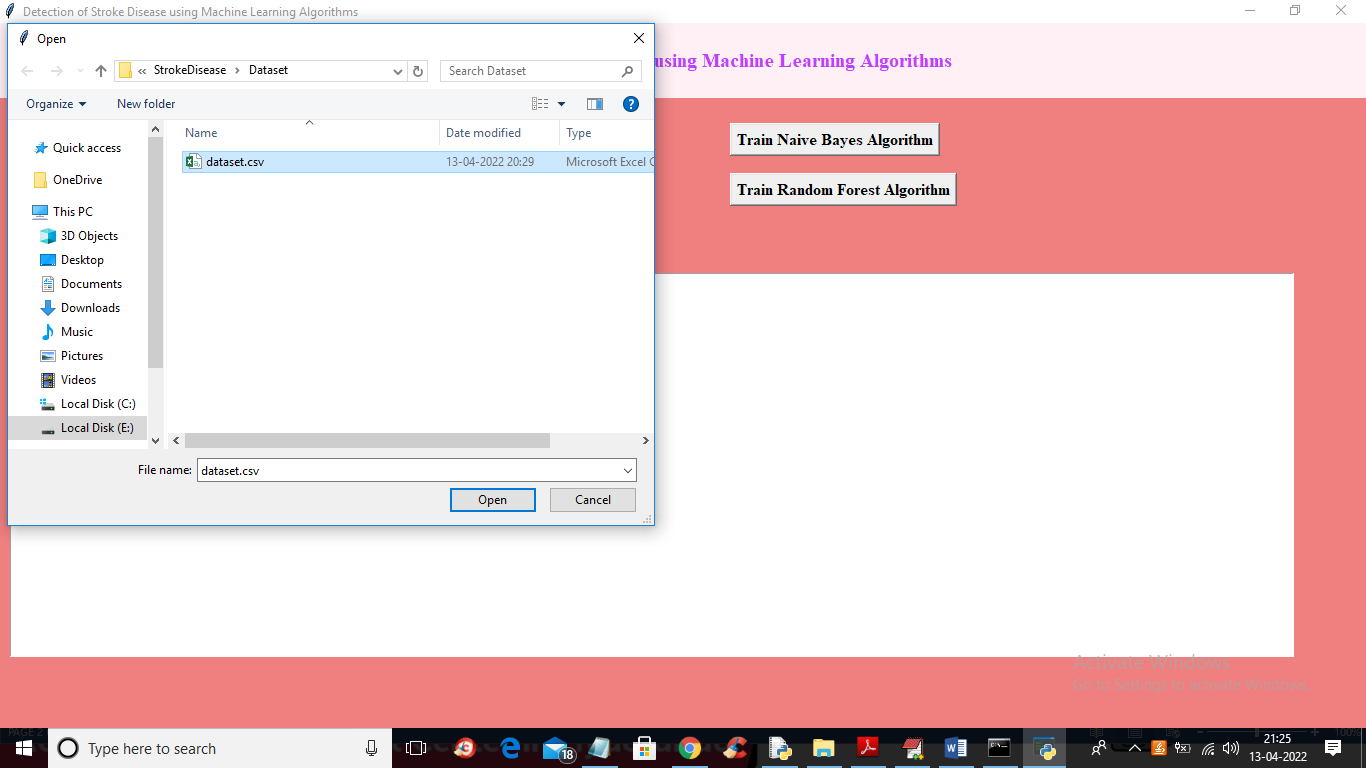


**SCREENSHOT 5.1 :** Click on ‘Upload Stroke Dataset’ to upload dataset.

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

CMRTC 22

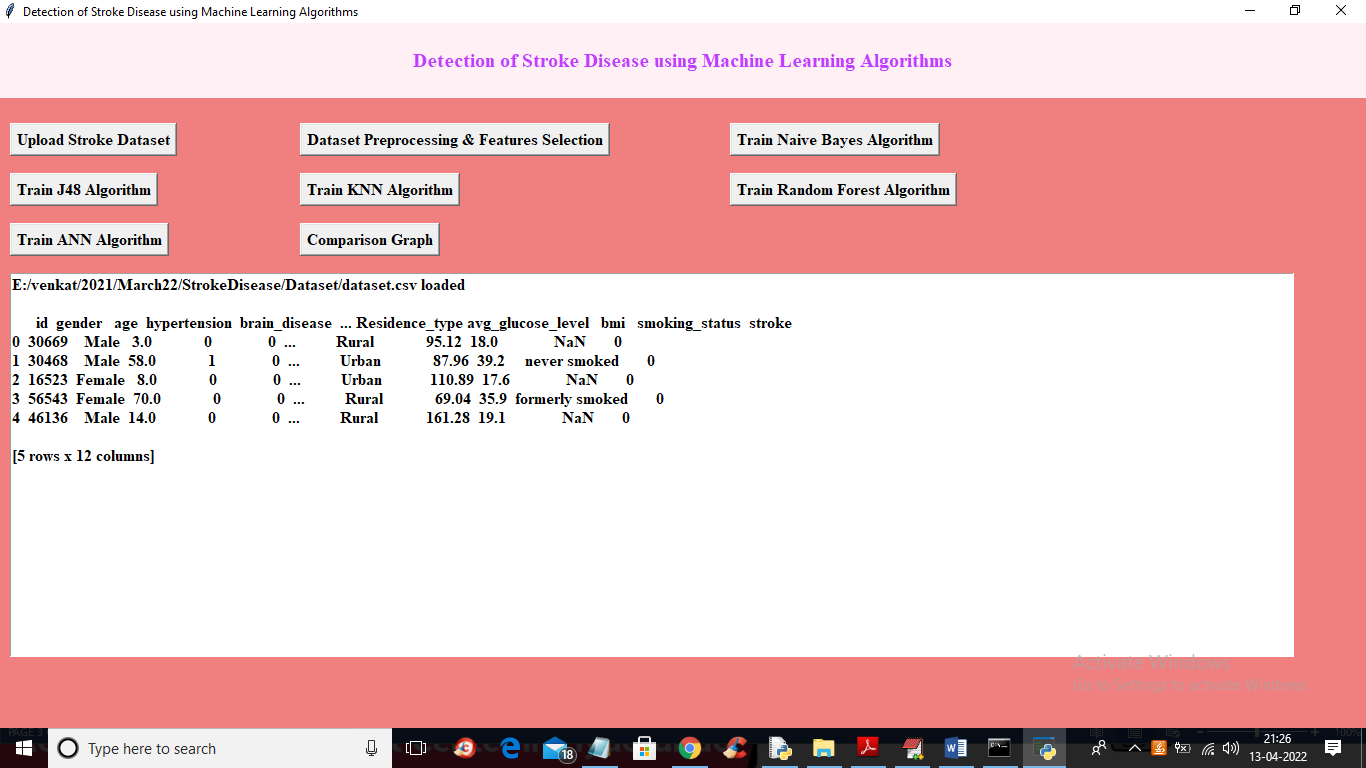
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.2 :** click on ‘Open’ button to load dataset.

CMRTC 23

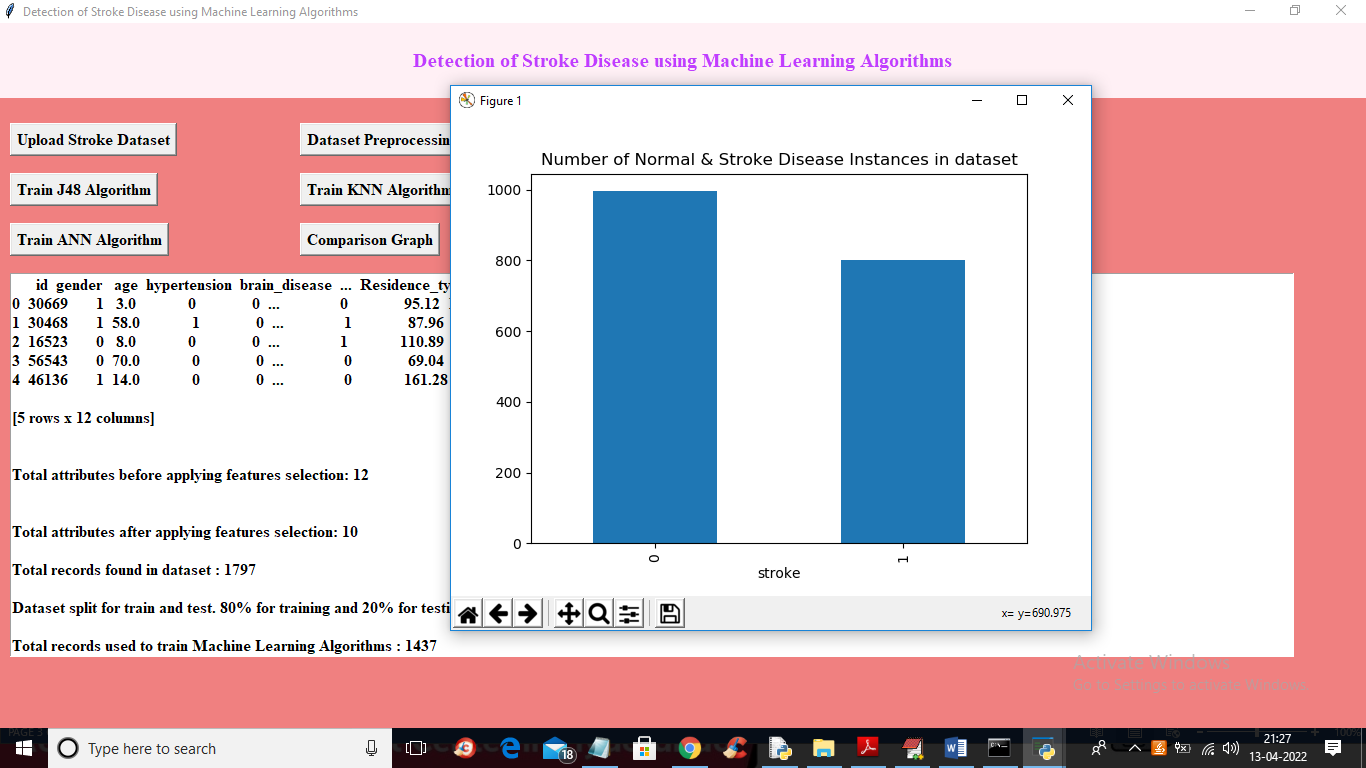
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.3 :** Click on ‘Dataset Preprocessing & Features Selection’ button to process dataset.

CMRTC 24

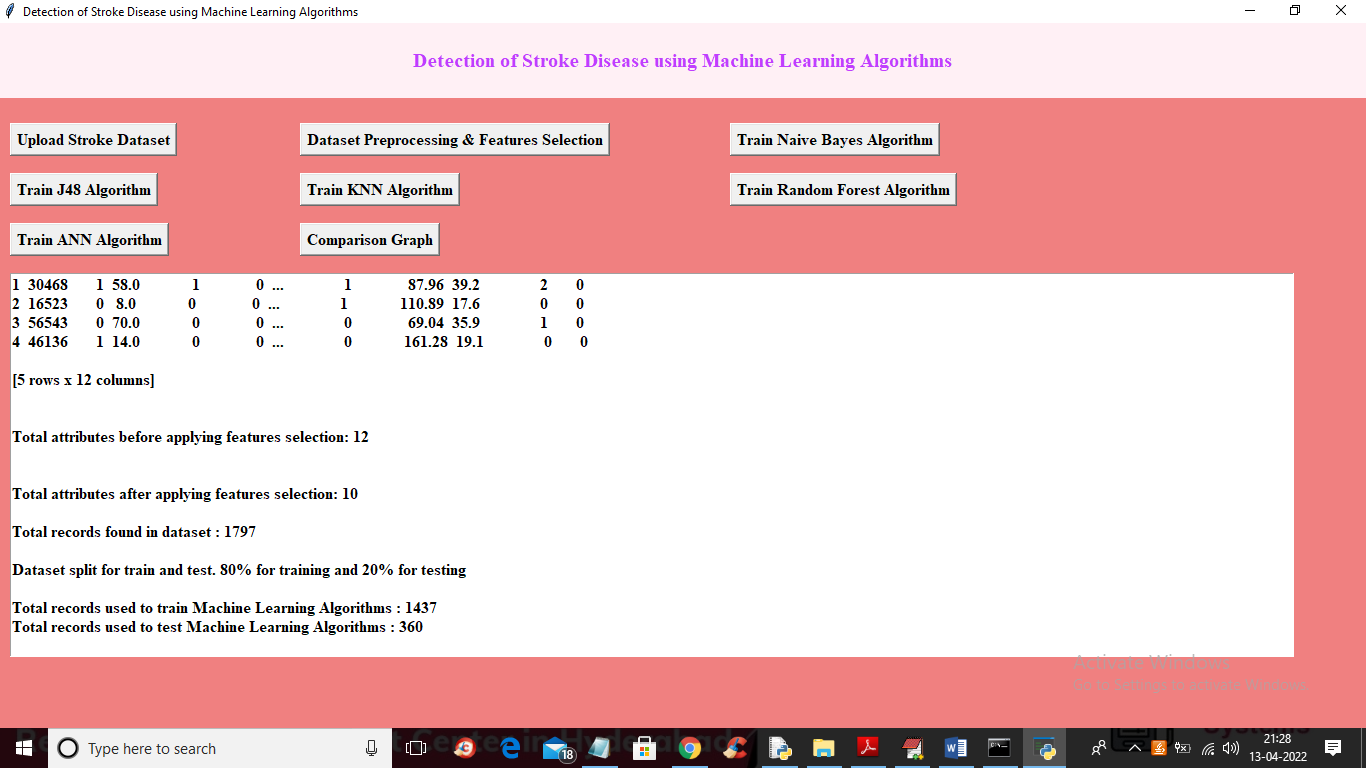
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.4 :** Graph x-axis represents 0 and 1 and y-axis represents number of instances available in data.

CMRTC 25

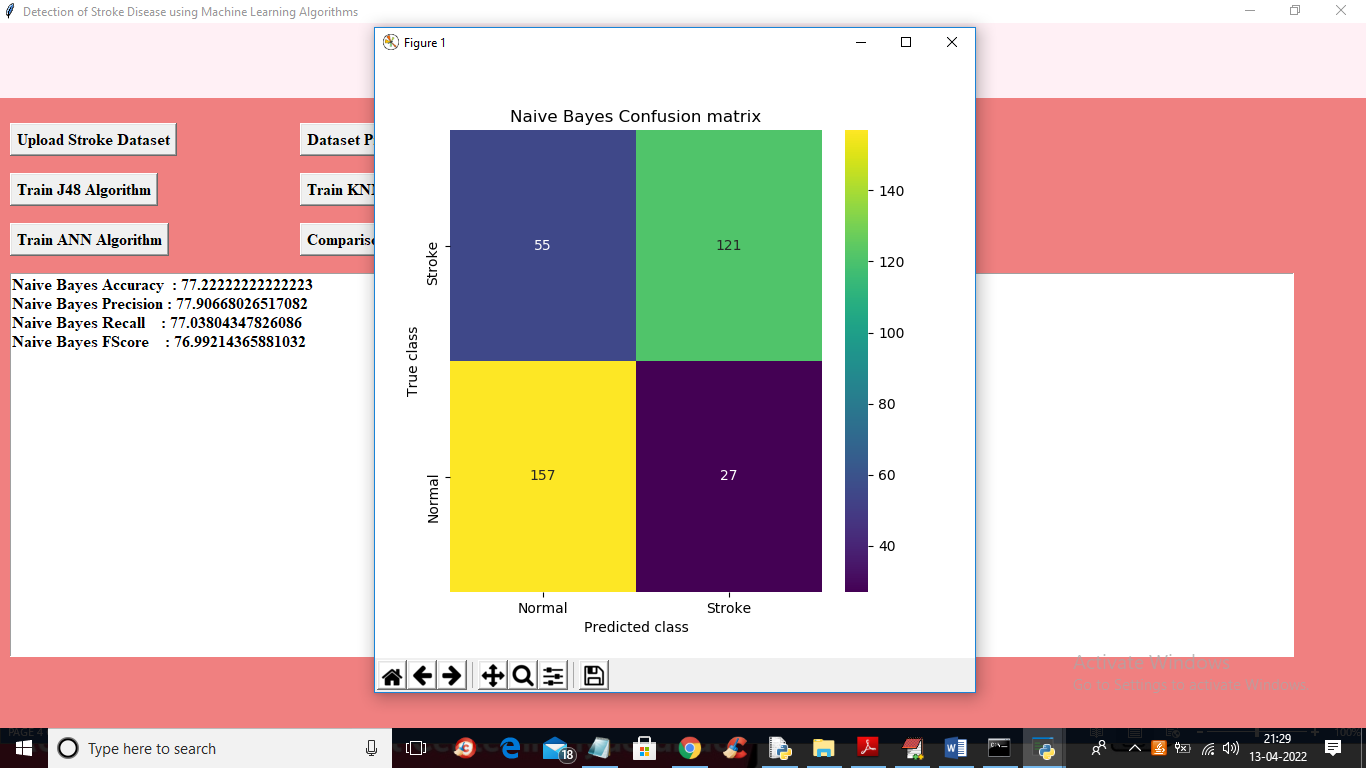
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.5 :** screen we can see all dataset converted to numeric format and then split dataset into train and test.

CMRTC 26

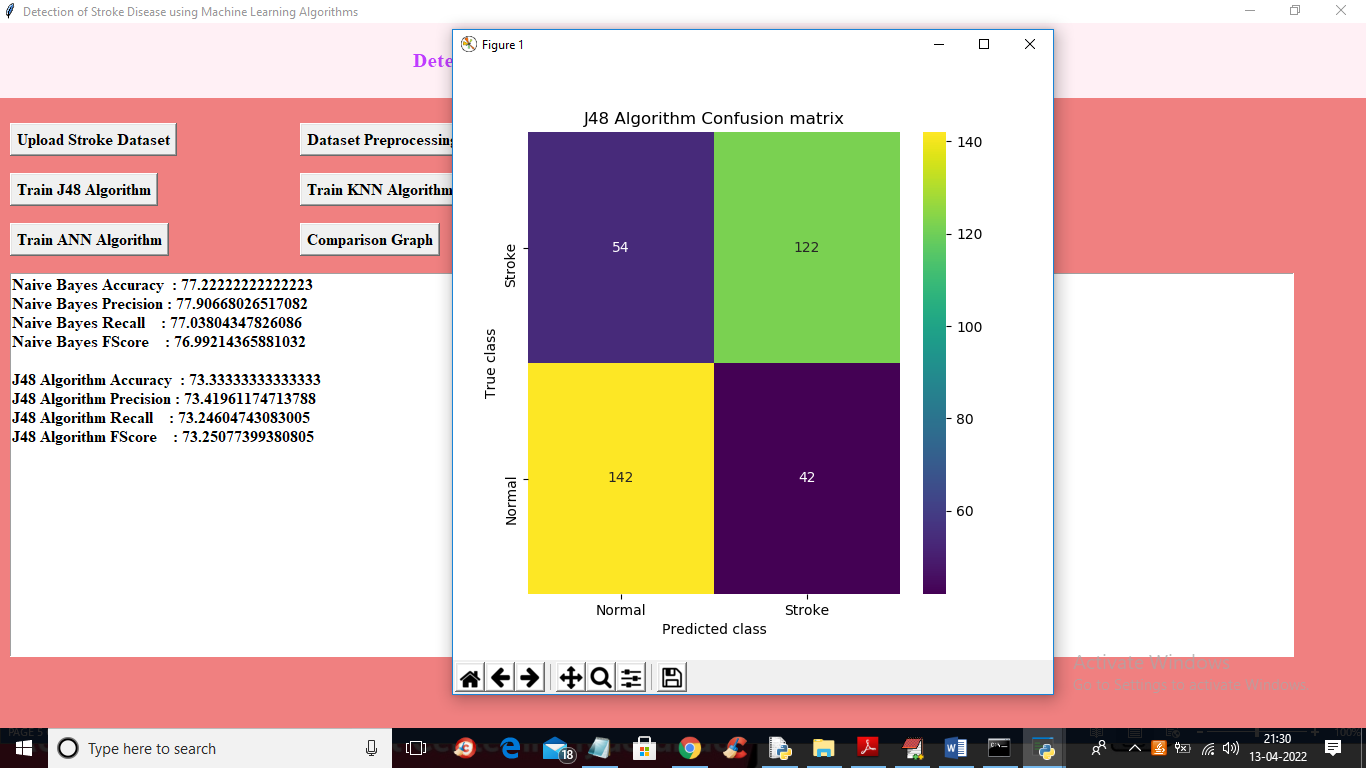
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.6** click on ‘Train Naïve Bayes Algorithm’ button to train Naïve Bayes on above dataset

CMRTC 27

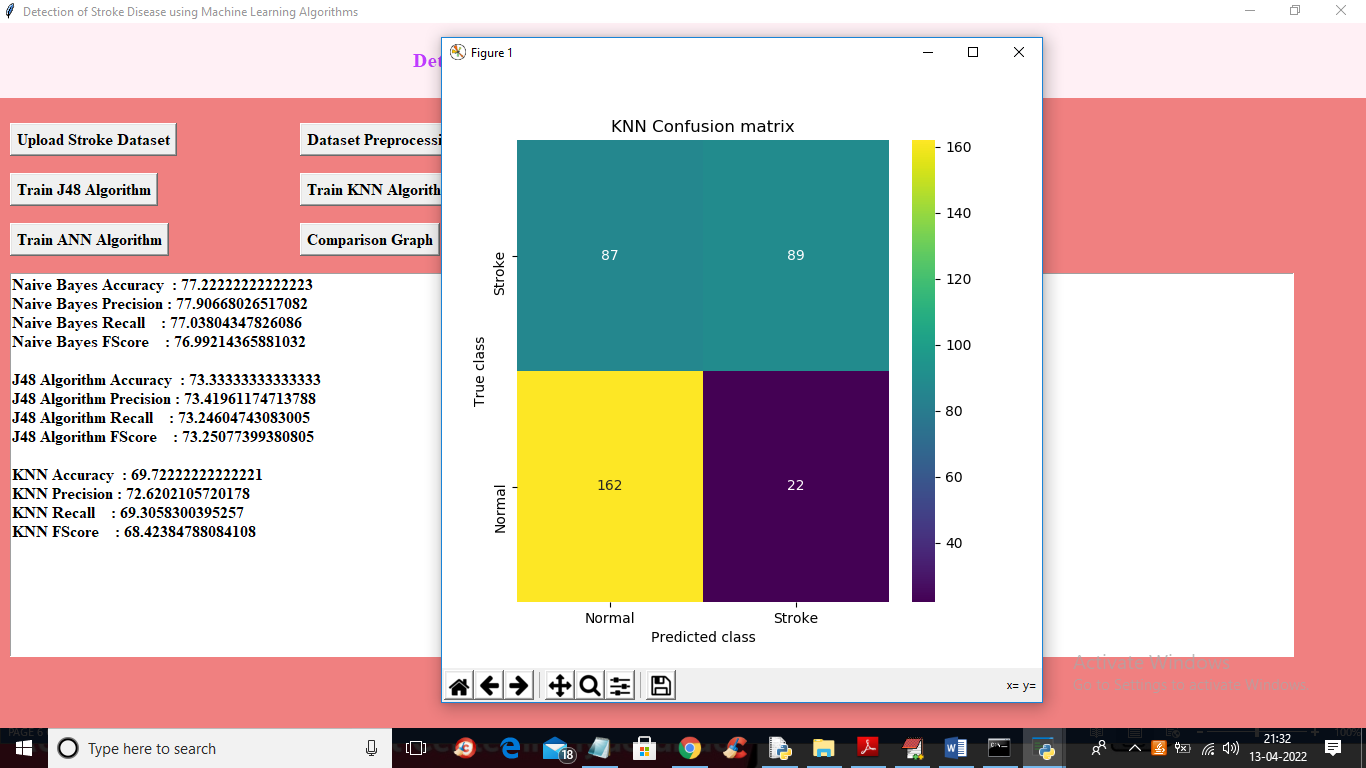
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.7 :** click on ‘Train J48 Algorithm’ button.

CMRTC 28

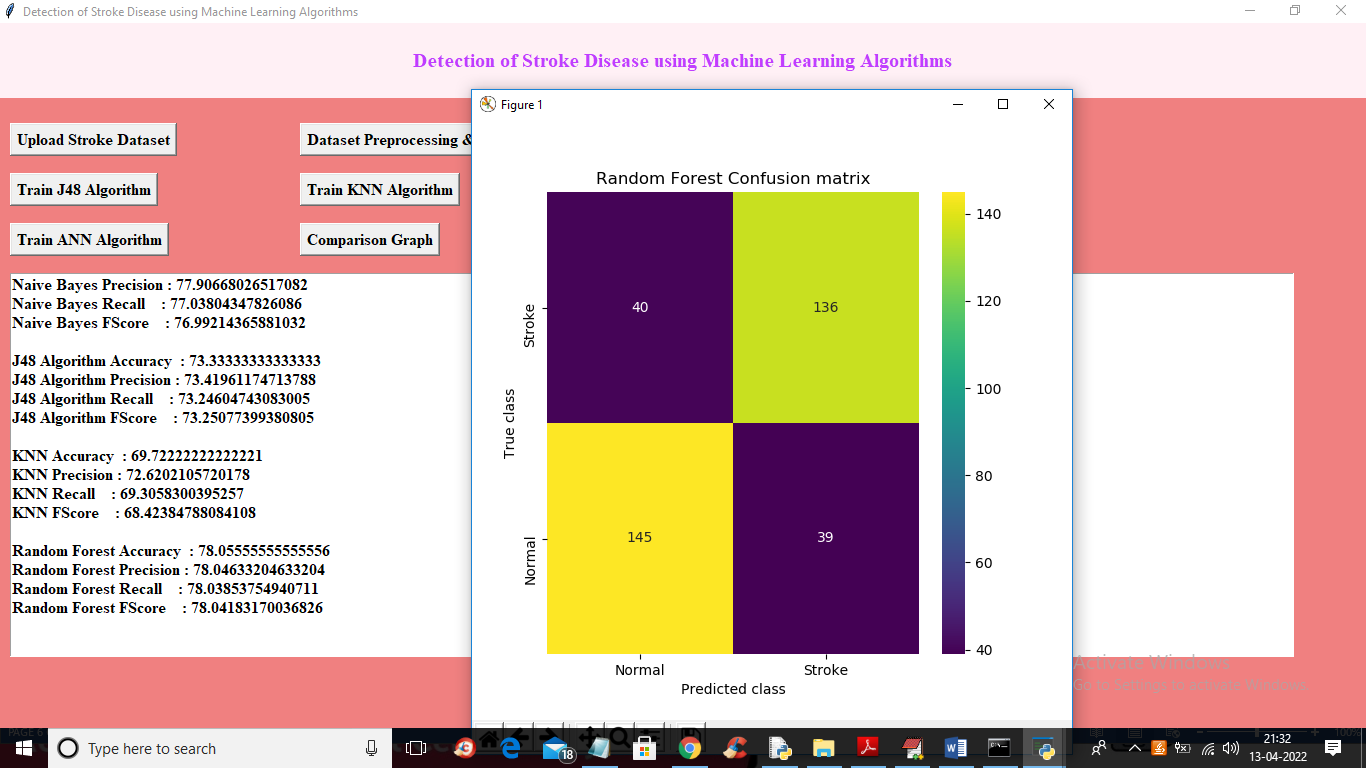
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.8 :** click on ‘Run KNN Algorithm’ button.

CMRTC 29

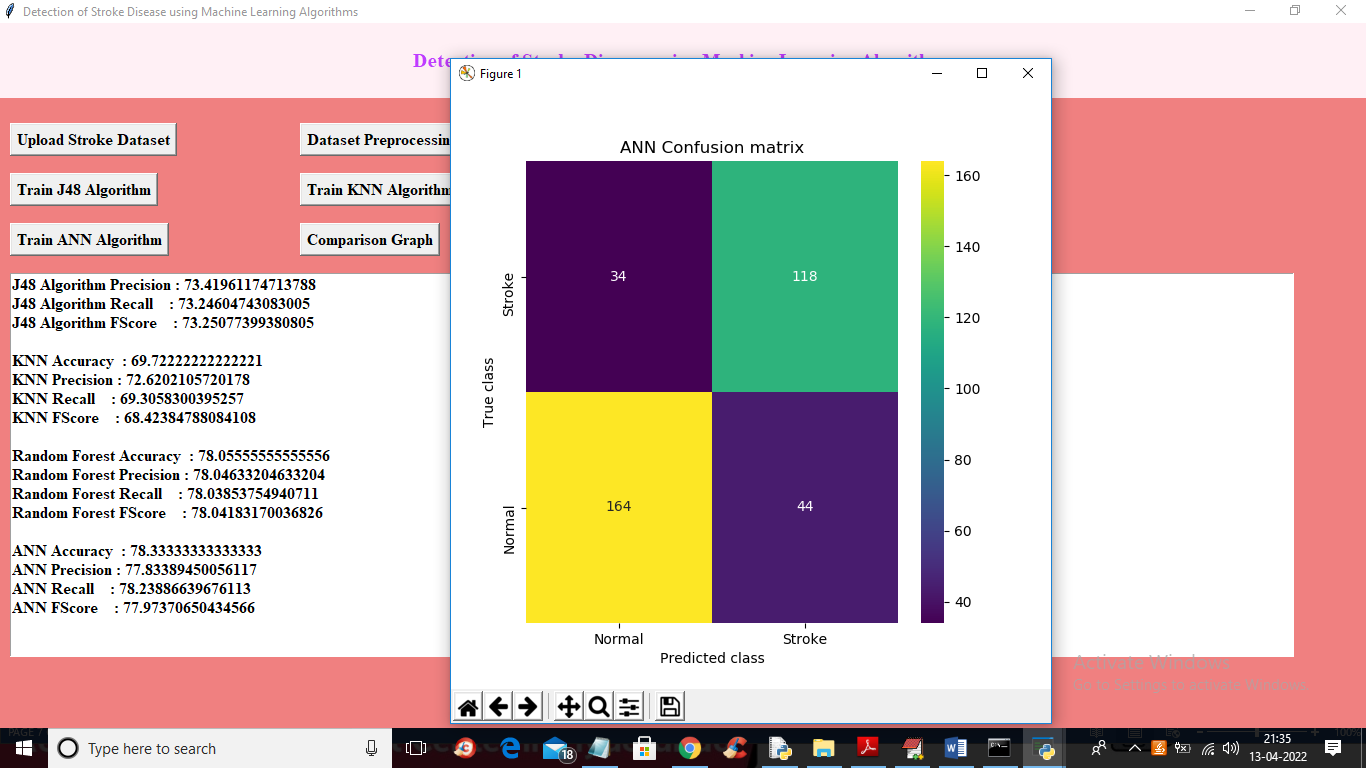
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.9:** click on ‘Run Random Forest Algorithm’ button.

CMRTC 30

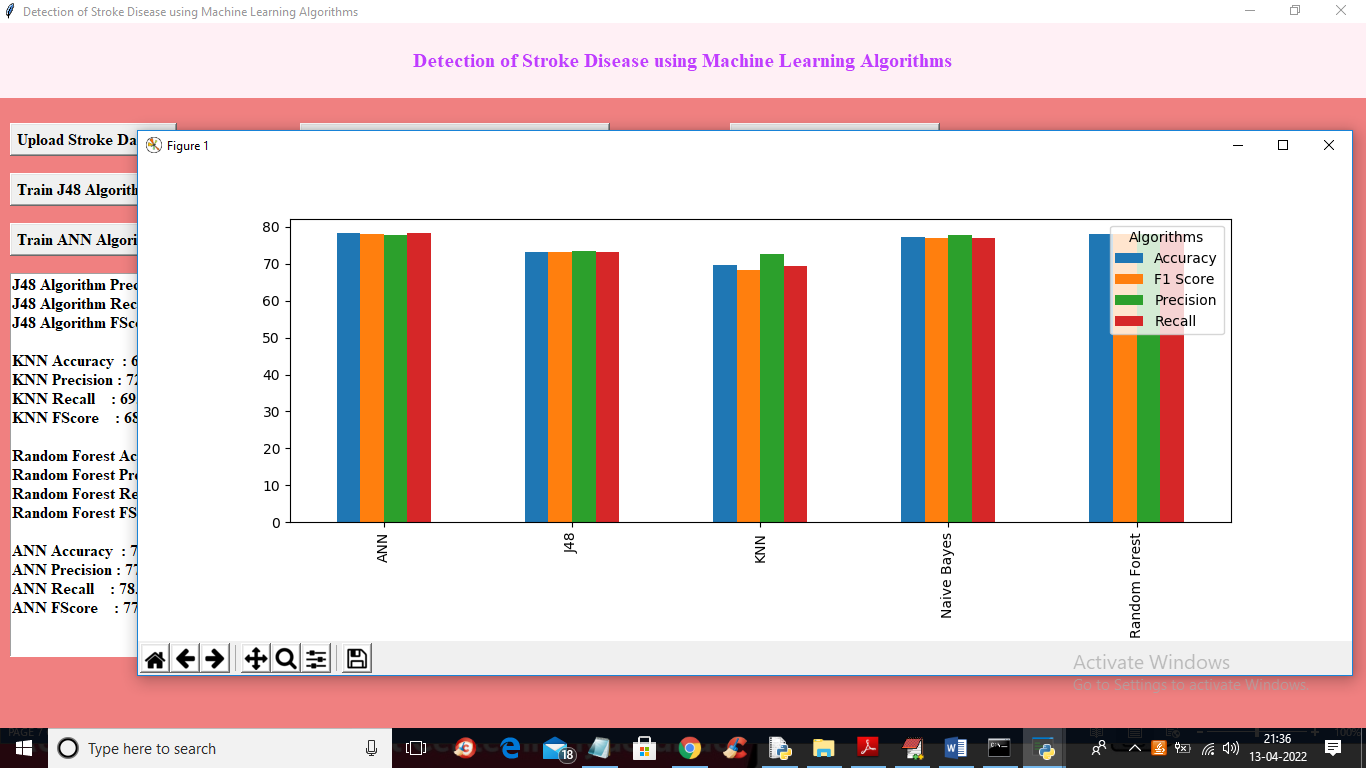
DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.10 :** click on ‘Run ANN Algorithm’ button.

CMRTC 31

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS



**SCREENSHOT 5.11 :** click on ‘Comparison Graph’ button. In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics like precision, recall etc. different colour bar represents different metrics and in all algorithms ANN got high accuracy

CMRTC 32

**6. TESTING**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

## 6. TESTING

### **6.1 INTRODUCTION TO TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

### **6.2** **TYPES OF TESTING**

#### **6.2.1 UNIT TESTING :**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

CMRTC 33

#### **6.2.2 INTEGRATION TESTING**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

#### **6.2.3** **FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

**Valid Input :** identified classes of valid input must be accepted.

**Invalid :** identified classes of invalid input must Input be rejected.

**Functions :** identified functions must be exercised.

**Output :** identified classes of application outputs must be exercised.

**6.2.4 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points

**6.2.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

CMRTC 34

**6.2.6 BLACK BOX TESTING**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

CMRTC 35

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

CMRTC 36

**7.CONCLUSION**

**7. CONCLUSION & FUTURE SCOPE**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

### **7.1 PROJECT CONCLUSION**

Naive Bayes, J48, k-NN, and Random Forest were used for detection of stroke disease. From the performance analysis we see that Naive Bayes performs better than other methods. The novelty and the main contribution of our work are collecting this dataset and preparing them to use with WEKA. The model can help people with a cautionary indication of being affected by stroke. Healthcare industries generate huge amounts of complex data about patients, hospitals resources, disease diagnosis, electronic patient records, medical devices, etc. Which is very difficult to relate to one another even by a field expert. It will help the clinician to better understand the type of disease. The limitations of our method are that the dataset is not perfectly symmetrical. However, it did not affect the predicted accuracy for the other algorithms. Naive Bayes algorithm didn’t work as we expected.

**7.2 FUTURE SCOPE**

In future work, it is possible to extend the research by using different classification techniques. Moreover, the prediction of stroke can be done by adding some non-stroke data with the existing dataset.

CMRTC 37

**8.BIBLIOGRAPHY**

**8. BIBLIOGRAPHY**

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

**8.1 REFERENCES**

[1] S. H. Pahus, A. T. Hansen, and A.-M. Hvas, “Thrombophilia testing in young patients with ischemic stroke,” Thrombosis research, vol. 137, pp. 108–112, 2016.

[2] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, and R. Manikandan, “Classification of stroke disease using machine learning algorithms,” Neural Computing and Applications, pp. 1–12.

[3] L. T. Kohn, J. Corrigan, M. S. Donaldson, et al., To err is human: building a safer health system, vol. 6. National academy press Washington, DC, 2000.

[4] R. Jeena and S. Kumar, “Stroke prediction using svm,” in 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 600–602, IEEE, 2016.

[5] P. A. Sandercock, M. Niewada, and A. Członkowska, “The international stroke trial database,” Trials, vol. 13, no. 1, pp. 1–1, 2012.

[6] M. S. Singh and P. Choudhary, “Stroke prediction using artificial intelligence,” in 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), pp. 158–161, IEEE, 2017.

[7] S. Y. Adam, A. Yousif, and M. B. Bashir, “Classification of ischemic stroke using machine learning algorithms,” Int J Comput Appl, vol. 149, no. 10, pp. 26–31, 2016.

[8] A. Sudha, P. Gayathri, and N. Jaisankar, “Effective analysis and predictive model of stroke disease using classification methods,” International Journal of Computer Applications, vol. 43, no. 14, pp. 26– 31, 2012.

[9] G. Kaur and A. Chhabra, “Improved j48 classification algorithm for the prediction of diabetes,” International Journal of Computer Applications, vol. 98, no. 22, 2014.

[10] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

CMRTC 38

[11] P. Sewaiwar and K. K. Verma, “Comparative study of various decision tree classification algorithm using weka,” International Journal of Emerging Research in Management &Technology, vol. 4, pp. 2278– 9359, 2015.

[12] K. A. Shakil, S. Anis, and M. Alam, “Dengue disease prediction using weka data mining tool,” arXiv preprint arXiv:1502.05167, 2015.

DETECTION OF STROKE DISEASE USING MACHINE LEARNING ALGORITHMS

[13] J. A. Alkrimi, H. A. Jalab, L. E. George, A. R. Ahmad, A. Suliman, and K. Al-Jashamy, “Comparative study using weka for red blood cells classification,” International Journal of Medical, Health, Pharmaceutical and Biomedical Engineering, vol. 9, no. 1, pp. 19–22, 2015.

[14] M. S. Siddiqui and A. I. Abidi, “Comparative study of different classification techniques using weka tool,” Global Sci-Tech, vol. 10, no. 4, pp. 200–208, 2018.

### **8.2 GITHUB LINK**

https://github.com/ajaykumarchigurla28/Detection-of-stroke-disease

CMRTC 39