

Breast Cancer Prediction and Comparative Analysis

INTRODUCTION

Breast cancer is a major public health concern that affects millions of people globally. Female breast cancer is currently the most often diagnosed cancer globally, surpassing lung cancer. In 2020, an expected 2,261,419 new breast cancer cases were identified in women worldwide. Breast cancer tumors can be classed as benign or malignant. Fibroadenomas are solid, smooth, hard, non-cancerous (benign) lumps that do not spread throughout the body. They may be uncomfortable or painful, but they are not life-threatening. Ductal carcinoma in situ, invasive ductal carcinoma, inflammatory breast cancer, and metastasis are all malignant tumors, which are cancerous growths that can spread beyond the breast tissue and affect other body organs. The difference between benign and malignant breast cancers is critical because it influences the best course of therapy. Machine Learning (ML) can reliably determine the kind of tumor by analyzing massive volumes of data and extremely complicated patterns. In this article, we classified the topic as a Binary Classification problem and used four distinct classification algorithms, including support vector machines and random forests, to predict breast cancer based on patient data and imaging findings.

Objective

Breast cancer is a condition that is frequently discussed these days. It is one of the most widely distributed disorders. It is critical to detect the illness so that women may begin treatment as soon as possible. It is preferable to have an accurate and timely diagnosis. The major purpose of this research is to assist pathologists in predicting cancer types more quickly.

DATASET ACQUISITION

We conducted our research using the dataset from the University of Wisconsin Hospitals Madison Breast Cancer Database. The dataset's characteristics are calculated using a digitised picture of a breast cancer sample obtained by fine-needle aspiration. These qualities enable us to deduce the properties of the cell nuclei shown in the picture. Breast Cancer Wisconsin Diagnostic includes 569 occurrences (Benign: 357, Malignant: 212), two classes (62.74% benign and 37.26% malignant), and 11 integer valued features (-Id, Diagnosis, Radius, Texture, Area, Perimeter, Smoothness, Compactness, Concavity, Concave points). -Symmetry (fractal dimension).

FEATURES	DESCRIPTION
Radius	It is the mean of distances from the centre to the points on the circumference
Texture	The standard deviation of the grey-scale values
Perimeter	Circumference of Tumour
Area	Area of the Tumour
Smoothness	It is the local deviation in radius
Compactness	Defined as $[(\text{perimeter}^2)/\text{area} - 1]$
Concavity	The gravity of concave portions on the silhouette
Concave points	Number of concave portions on the silhouette
Symmetry	A balanced and proportionate similarity that is found in two halves of an object
Fractal dimension	It is a characteristic parameter used to describe the irregular extent of coastline

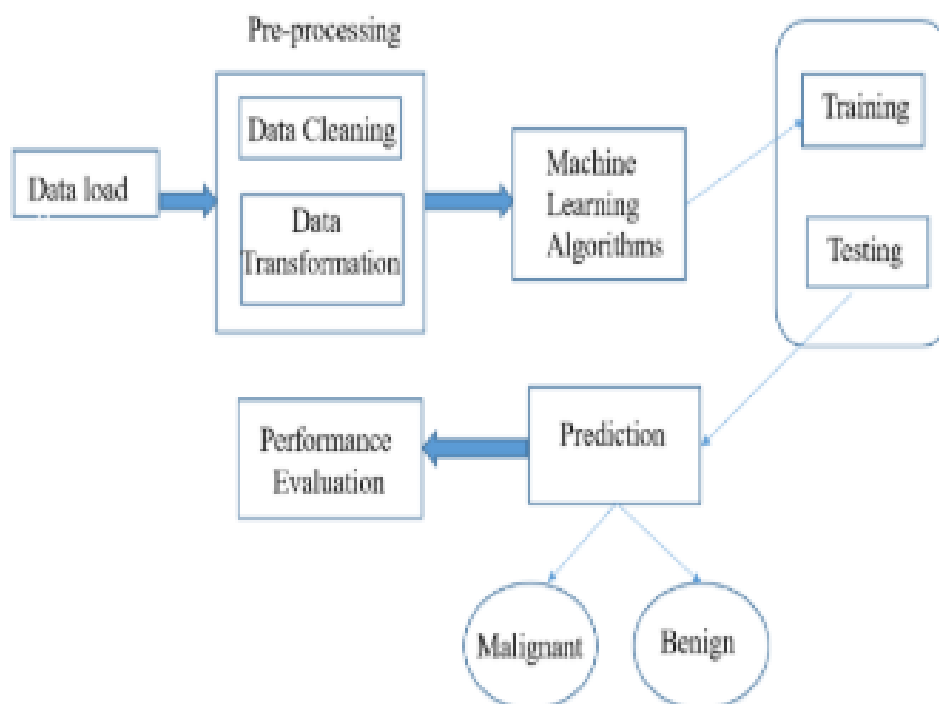
Data Preprocessing

When it comes to developing a machine learning model, data pre-processing is the first step that signals the start of the process. Typically, real-world data is imprecise, inconsistent, and erroneous (including mistakes and outliers), and it frequently lacks particular attribute values and trends. This is where data pre-processing comes into play: it helps to calm, format, and organize raw data, making it ready to use in machine learning models.

Proposed System:

Breast cancer detection is incredibly crucial in today's medical environment. Breast cancer is one of the most serious cancers that may affect women. Breast cancer (BC) has two types: benign (noncancerous) and malignant. Malignant cancer is described as a curable kind of cancer, but benign cancer is listed as an incurable condition. Changes in genes, intense pain, size and form, fluctuations in breast color (redness), and skin texture are all indications of breast cancer. Machine learning is used to make predictions. To diagnose breast cancer, many classification approaches are applied, such as Support Vector Machine (SVM) and Random Forest. These methods belong to the domain of supervised machine learning. These methods are used to predict the development of breast cancer. These algorithms' accuracy results are assessed.

Proposed System Architecture:



LITERATURE REVIEW

The author of [1] hypothesized that breast cancer may be predicted using a dataset obtained from the Wisconsin Breast Cancer repository. The data collection includes 569 data points and 30 attributes. Logistic Regression's accuracy is around 96.5%.

In [2], the author compares ML algorithms for breast cancer prediction. The article incorporates machine learning approaches such as decision trees and logistic regression. This research makes use of the WDBC dataset, which has 570 rows and 32 columns. According to the research, logistic regression provided 94.4% accuracy, but decision tree provided around 95.14% accuracy; hence, the decision tree method was chosen to provide more accurate predictions.

In [3], the author suggested work on detecting breast cancer risk factors using machine learning algorithms. The Support Vector Machine classifier is compared with the Naive Bayes. The Wisconsin diagnostic breast cancer dataset was used to make breast cancer predictions. The SVM method performed excellently, demonstrating accuracy up to 97.91% as opposed to the NB algorithm, which delivered 95.6% accuracy.

In [4], the author suggested work on Breast Cancer analysis using the K Nearest Neighbor method. The author employed KNN to predict breast cancer. The Manhattan distance with $K = 1$ produces an accuracy of around 98.40%, but the Euclidean distance with $K = 1$ produces a high accuracy of approximately 98.70%.

In [5] the author worked on an intelligent system employing SVM based classifier for predictive breast cancer detection and prognosis. Support vector machines (SVMs)-based classifiers outperform Bayesian classifiers and artificial neural networks for the diagnosis and prognosis of breast cancer sickness. The enhanced SVM method performed admirably, displaying high values for great significant to 96.91, specificity up to 97.67 percent, and sensitivity up to 97.84 percent.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

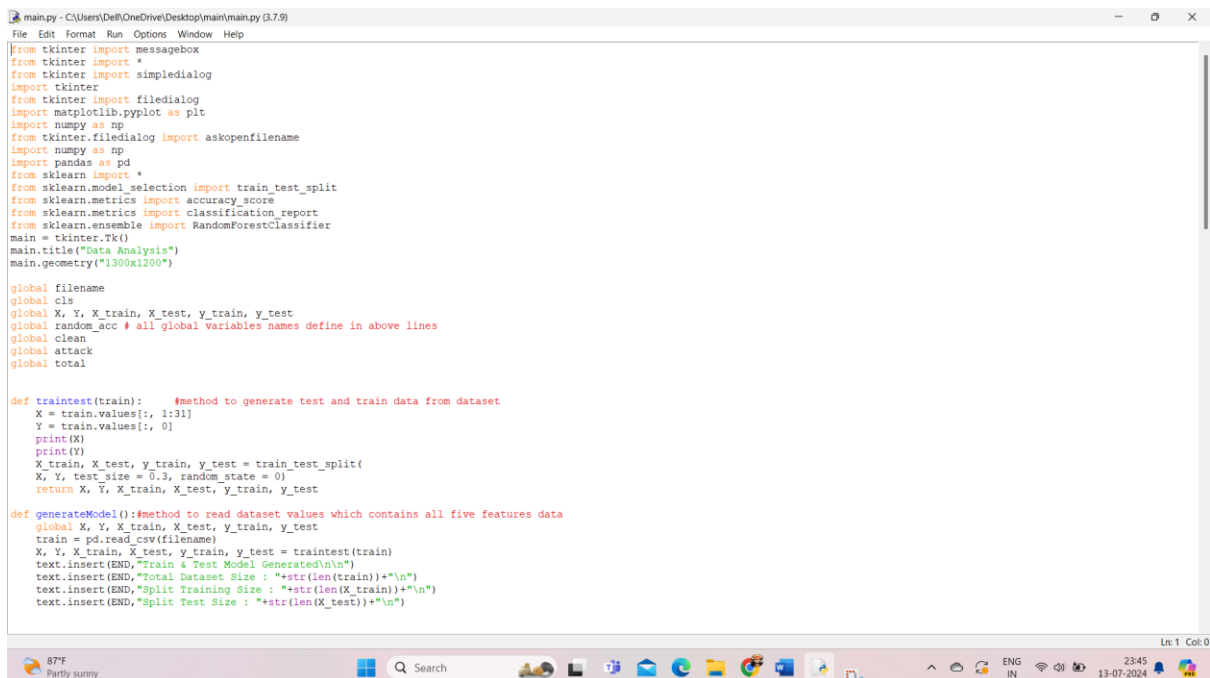
- System: Pentium i5 Processor.
- Hard Disk: 1 TB.
- Monitor: 15’’ LED
- Input Devices: Keyboard, Mouse
- Ram: 16 GB

SOFTWARE REQUIREMENTS:

- Operating system: Windows 11.
- Coding Language: Python with ML.
-

TEST RESULTS AND OUTPUT

STEP-1: First we have to open our “main” in python idle



```
main.py - C:\Users\De\OneDrive\Desktop\main\main.py (3.7.9)
File Edit Format Run Options Window Help
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
main = tkinter.Tk()
main.title("Data Analysis")
main.geometry("1300x1200")

global filename
global cls
global X, Y, X_train, X_test, y_train, y_test
global random_acc # all global variables names define in above lines
global clean
global attack
global total

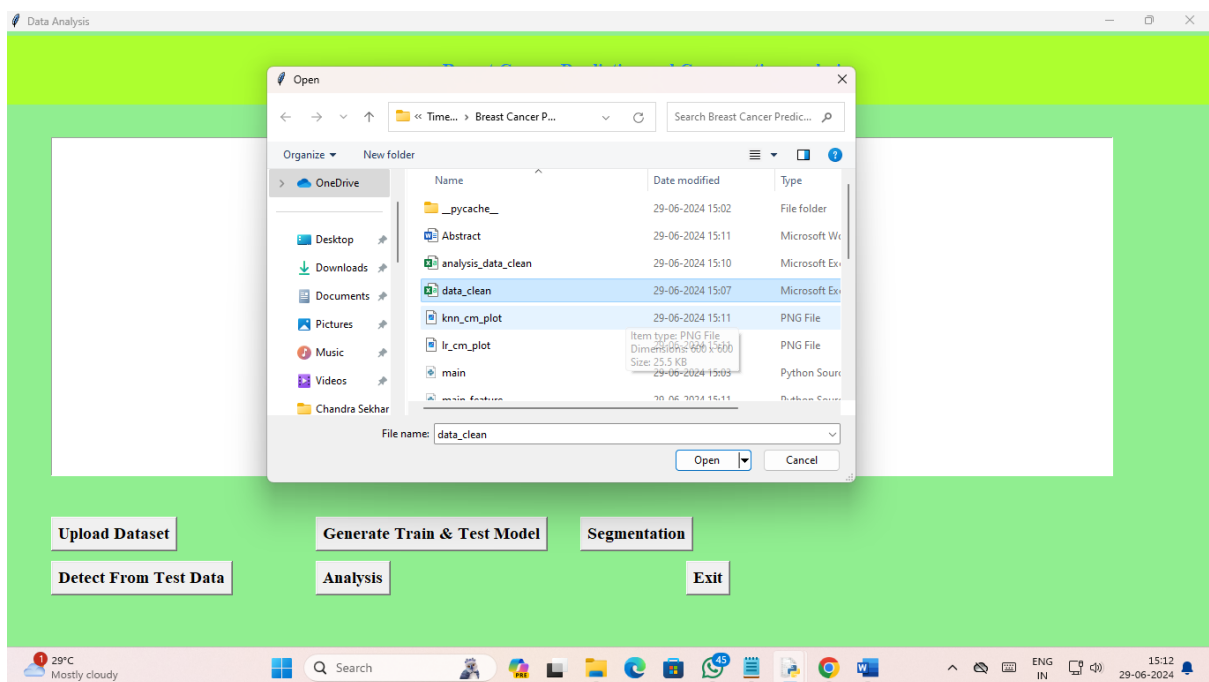
def traintest(train): #method to generate test and train data from dataset
    X = train.values[:, 1:31]
    Y = train.values[:, 0]
    print(X)
    print(Y)
    X_train, X_test, y_train, y_test = train_test_split(
        X, Y, test_size = 0.3, random_state = 0)
    return X, Y, X_train, X_test, y_train, y_test

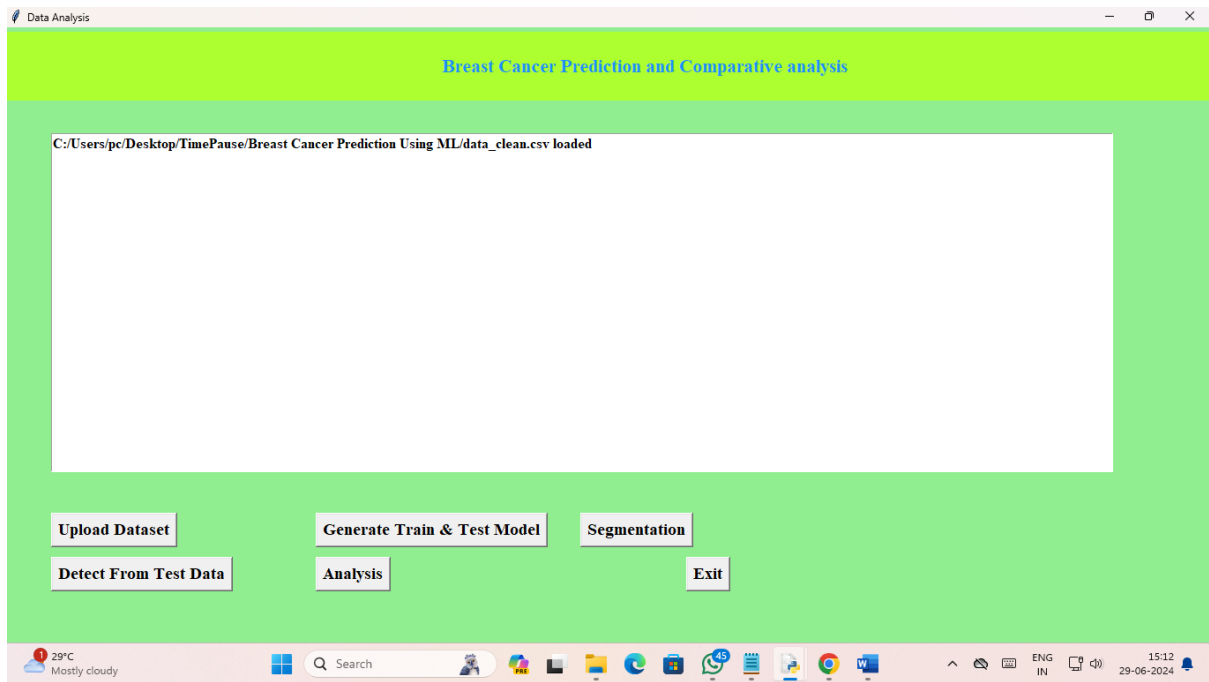
def generateModel(): #method to read dataset values which contains all five features data
    global X, Y, X_train, X_test, y_train, y_test
    train = pd.read_csv(filename)
    X, Y, X_train, X_test, y_train, y_test = traintest(train)
    text.insert(END, "Train & Test Model Generated\n\n")
    text.insert(END, "Total Dataset Size : "+str(len(train))+"\n")
    text.insert(END, "Split Training Size : "+str(len(X_train))+"\n")
    text.insert(END, "Split Test Size : "+str(len(X_test))+"\n")
```

STEP -2 : Later we have to run the code then our main page will be as output



STEP- 3: Next, we must upload the data set which is “data_clean” for prediction.

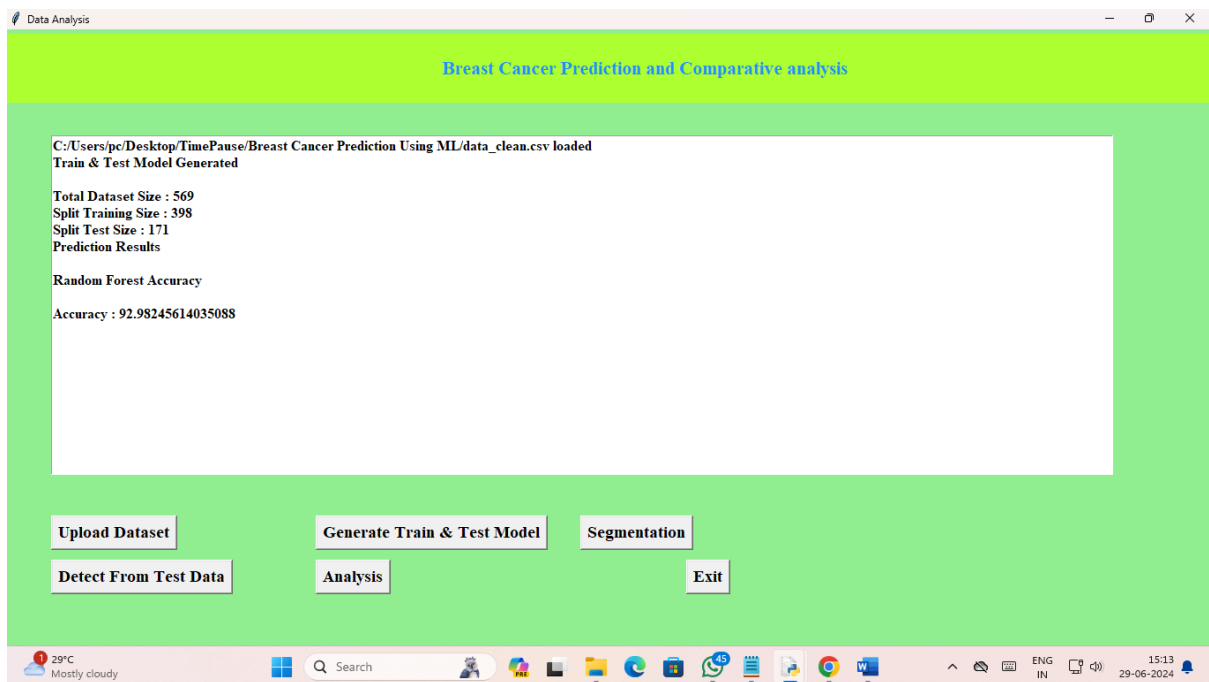




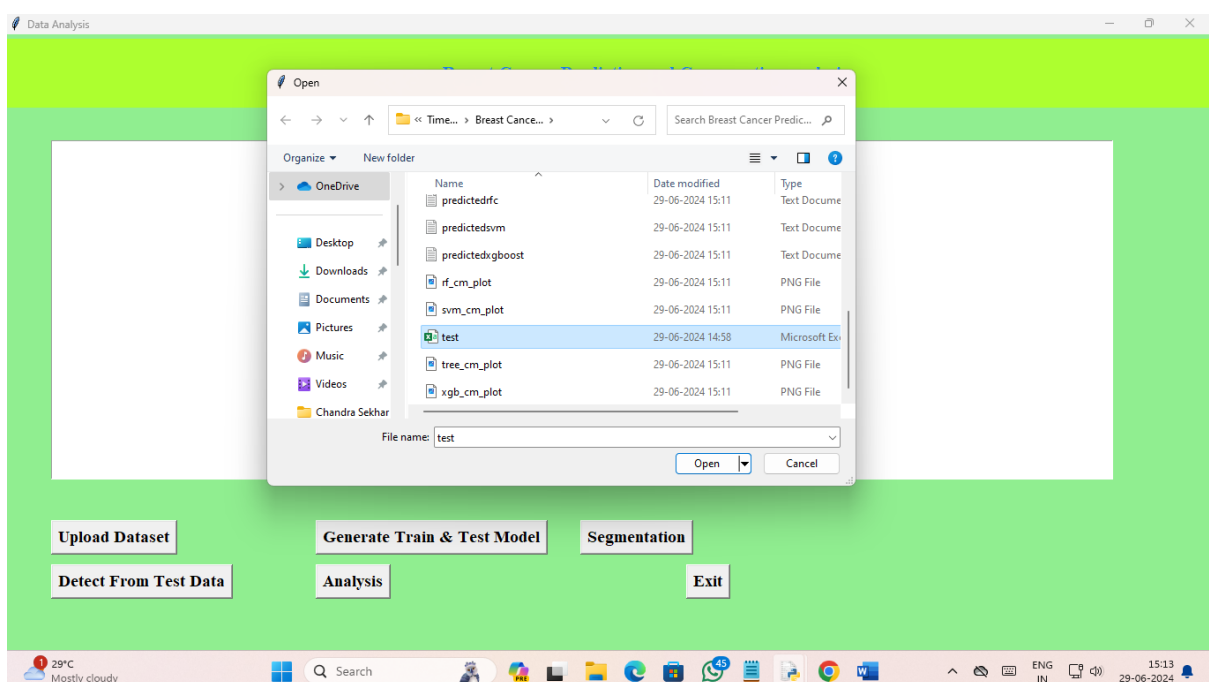
STEP- 4: In this step we split the Train test

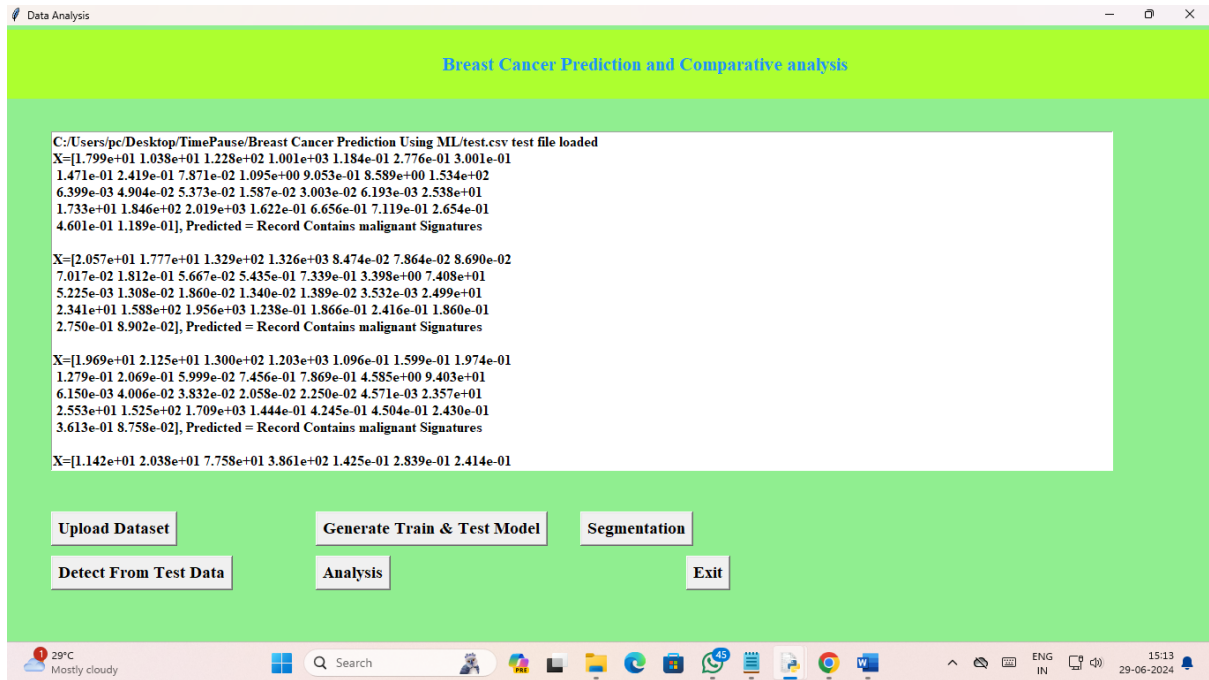


STEP – 5: The next step here is segmentation which we applied “Random Forest Classification”

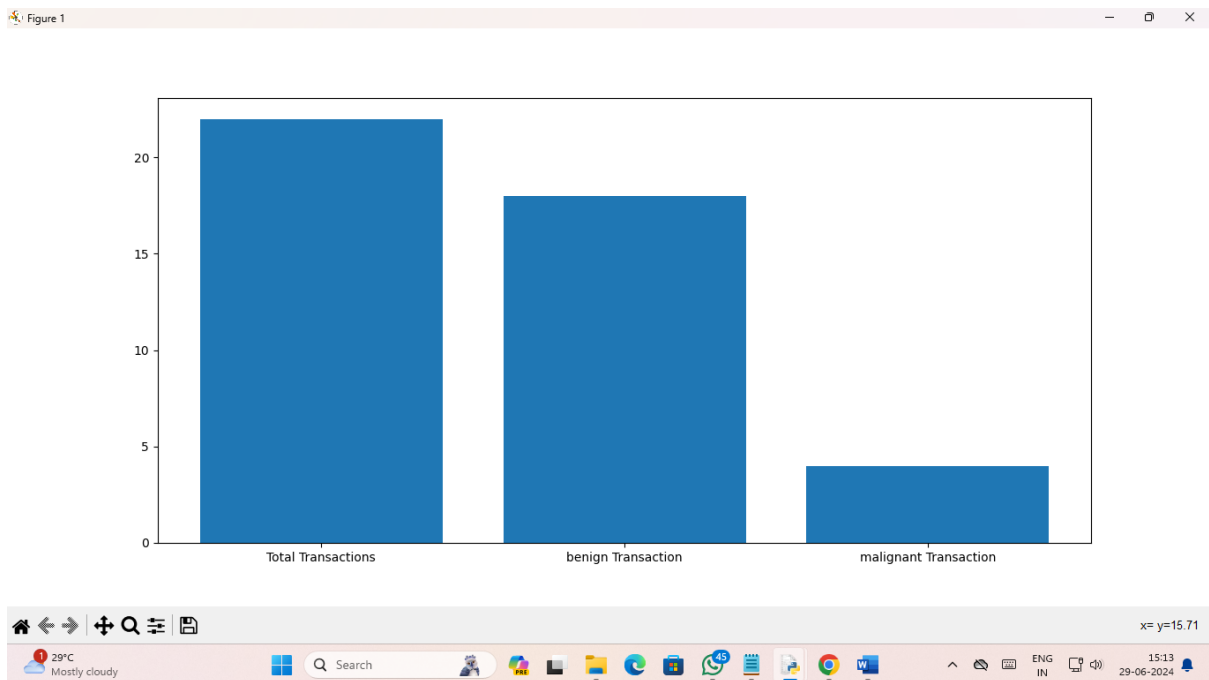


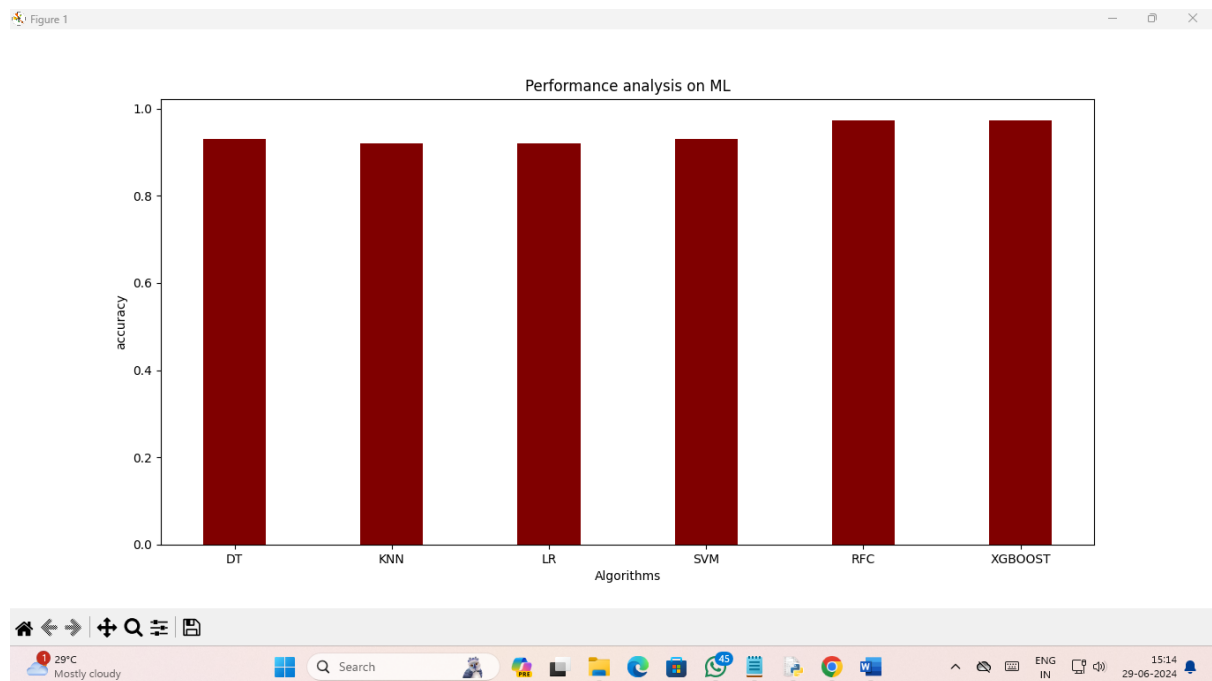
STEP – 6: Next step is “Detect from Test Data” means input the test sample which is “test.csv”





STEP-7: Next step is Analysis in this step we have used different machine learning algorithms such as “Random Forest Analysis”, “XG BOOST”, “SVM”, “KNN”, “DECISION TREE”, “LINEAR REGRESSION”





STEP-8: In this last step it shows the accuracy of our project with 6 different machine learning algorithms

```

[Accuracy Score]
-----
[Accuracy score of the Decision Tree model is 0.9298245614035088]
-----
[Accuracy score of the KNN model is 0.9210526315789473]
-----
[Accuracy score of the Logistic Regression model is 0.9210526315789473]
-----
[Accuracy score of the SVM model is 0.9298245614035088]
-----
[Accuracy score of the Random Forest Tree model is 0.9736842105263158]
-----
[Accuracy score of the XGBoost model is 0.9736842105263158]
-----

```

CODE:

```
from tkinter import messagebox

from tkinter import *

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import numpy as np

import pandas as pd

from sklearn import *

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

from sklearn.ensemble import RandomForestClassifier

main = tkinter.Tk()

main.title("Data Analysis")

main.geometry("1300x1200")


global filename

global cls

global X, Y, X_train, X_test, y_train, y_test

global random_acc # all global variables names define in above lines
```

global clean

global attack

global total

```
def traintest(train):    #method to generate test and train data from dataset
```

```
    X = train.values[:, 1:31]
```

```
    Y = train.values[:, 0]
```

```
    print(X)
```

```
    print(Y)
```

```
    X_train, X_test, y_train, y_test = train_test_split(
```

```
    X, Y, test_size = 0.3, random_state = 0)
```

```
    return X, Y, X_train, X_test, y_train, y_test
```

```
def generateModel():#method to read dataset values which contains all five features data
```

```
    global X, Y, X_train, X_test, y_train, y_test
```

```
    train = pd.read_csv(filename)
```

```
    X, Y, X_train, X_test, y_train, y_test = traintest(train)
```

```
    text.insert(END,"Train & Test Model Generated\n\n")
```

```
    text.insert(END,"Total Dataset Size : "+str(len(train))+"\n")
```

```
    text.insert(END,"Split Training Size : "+str(len(X_train))+"\n")
```

```
    text.insert(END,"Split Test Size : "+str(len(X_test))+"\n")
```

```
def upload(): #function to upload tweeter profile

    global filename

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

    text.delete('1.0', END)

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

```
def prediction(X_test, cls): #prediction done here

    y_pred = cls.predict(X_test)

    for i in range(50):

        print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))

    return y_pred
```

Function to calculate accuracy

```
def cal_accuracy(y_test, y_pred, details):

    accuracy = accuracy_score(y_test,y_pred)*100

    text.insert(END,details+"\n\n")

    text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")

    return accuracy
```

```
def runRandomForest():
```

```

global random_acc

global cls

global X, Y, X_train, X_test, y_train, y_test

cls =
RandomForestClassifier(n_estimators=50,max_depth=2,random_state=0,class_weight='balanced')

cls.fit(X_train, y_train)

text.insert(END,"Prediction Results\n\n")

prediction_data = prediction(X_test, cls)

random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Accuracy')


def predicts():

    global clean

    global attack

    global total

    clean = 0;

    attack = 0;

    text.delete('1.0', END)

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

    test = pd.read_csv(filename)

    test = test.values[:, 1:31]

    total = len(test)

    text.insert(END,filename+" test file loaded\n");

    y_pred = cls.predict(test)

```

```

f = open("op1.txt", "a")

for i in range(len(test)):

    if str(y_pred[i]) == '2.0':

        attack = attack + 1

        text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Record Contains benign
Signature')+ "\n\n")

        f.write(str("X=%s, Predicted = %s" % (test[i], 'Record Contains Signature')+ "\n\n"))

    else:

        clean = clean + 1

        text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Record Contains malignant
Signatures')+ "\n\n")

        f.write(str("X=%s, Predicted = %s" % (test[i], 'Record Contains malignant
Signatures')+ "\n\n"))

f.close()

def graph():

    height = [total,clean,attack]

    bars = ('Total Transactions','benign Transaction','malignant Transaction')

    y_pos = np.arange(len(bars))

    plt.bar(y_pos, height)

    plt.xticks(y_pos, bars)

    plt.show()

import main_feature

```

```
font = ('times', 16, 'bold')
```

```
title = Label(main, text='Breast Cancer Prediction and Comparative analysis')
```

```
title.config(bg='greenyellow', fg='dodger blue')
```

```
title.config(font=font)
```

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

```
title.place(x=0,y=5)
```

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

```
text=Text(main,height=20,width=150)
```

```
scroll=Scrollbar(text)
```

```
text.configure(yscrollcommand=scroll.set)
```

```
text.place(x=50,y=120)
```

```
text.config(font=font1)
```

```
font1 = ('times', 14, 'bold')
```

```
uploadButton = Button(main, text="Upload Dataset", command=upload)
```

```
uploadButton.place(x=50,y=550)
```

```
uploadButton.config(font=font1)
```

```
modelButton = Button(main, text="Generate Train & Test Model", command=generateModel)
```

```
modelButton.place(x=350,y=550)
```

```
modelButton.config(font=font1)
```



```
runrandomButton = Button(main, text="Segmentation", command=runRandomForest)
```

```
runrandomButton.place(x=650,y=550)
```

```
runrandomButton.config(font=font1)
```

```
predictButton = Button(main, text="Detect From Test Data", command=predicts)
```

```
predictButton.place(x=50,y=600)
```

```
predictButton.config(font=font1)
```

```
graphButton = Button(main, text="Analysis", command=graph)
```

```
graphButton.place(x=350,y=600)
```

```
graphButton.config(font=font1)
```

```
exitButton = Button(main, text="Exit", command=exit)
```

```
exitButton.place(x=770,y=600)
```

```
exitButton.config(font=font1)
```

```
main.config(bg='LightGreen')
```

```
main.mainloop()
```

REFERENCES

- S. Ara, A. Das, and A. Dey, "Malignant and Benign Breast Cancer Classification using Machine Learning Algorithms," 2021 International Conference on Artificial Intelligence (ICAI), 2021.
- Sultana, Jabeen, and Abdul Khader Jilani. "Predicting Breast Cancer Using Logistic Regression and Multi-Class Classifiers." International Journal of Engineering & Technology [Online], 7.4.20 (2018): 22–26. Web. November 30, 2019.
- P. Sathiyarayanan, S. Pavithra, M.Sai Saranya, and M.Makeswari, "Identification of breast cancer using the Decision Tree Algorithm," 2019 IEEE International Conference on System, Computing, Automation, and Networking (ICSCAN), 2019.
- M.D. Bakthavachalam, Dr.S. Albert, and Antony Raj, "Breast Cancer Analysis using K-Nearest Neighbor Algorithm," 2020 International Conference on Artificial Intelligence (ICCS), 2020.
- Puneet Yadav et al. "Diagnosis of Breast Cancer Using Decision Tree Models and SVM", International Research Journal of Engineering and Technology, Vol. 5, Issue 3, March 2018.
- Medjahed, Seyyid Ahmed, Tamazouzt Ait Saadi, and Abdelkader Benyettou. "Breast cancer diagnosis by using knearest neighbor with different distances and classification rules."
- Kriti Jain et al. "Breast Cancer Diagnosis Using Machine Learning Techniques", International Journal of Innovative Science, Engineering, and Technology, volume 5, issue 5, May 2018.
- Zheng, Bichen, Sang Won Yoon, and Sarah S Lam. "Breast cancer diagnosis based on feature extraction using a hybrid of Kmeans and support vector machine algorithms." Expert Systems with Applications 41.4 (2014): 1476–1482.