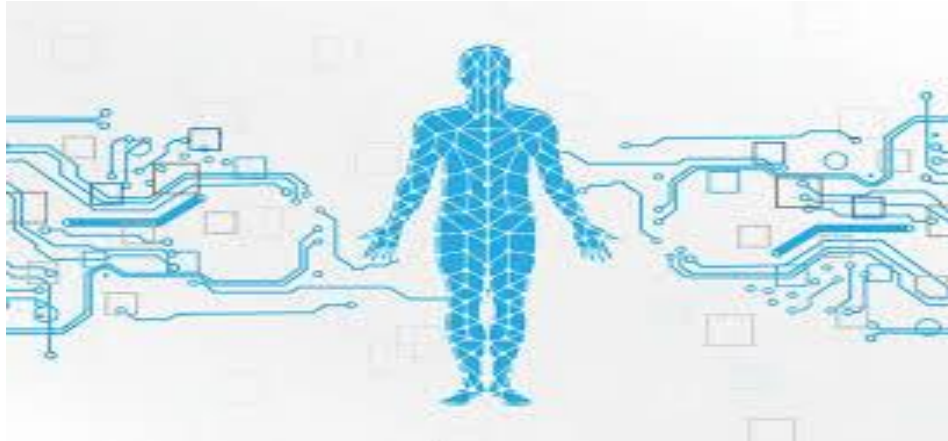


Disease Prediction System for Parkinson Disease and Breast Cancer using Voice Command



[Project Documentation]

SUBMITTED FOR PARTIAL FULFILMENT OF

MASTER OF COMPUTER APPLICATIONS

[2021 – 2023]

Ashutosh Kumar Yadav	504121011006
Mujahid Ali Ansari	504121011025
Shahil Kumar Chourasia	504121011045
Sirsha Majumder	504121021050
Sonu Routh	504121011051

UNDER THE GUIDANCE OF DR. ANANJAN MAITY

**(ASSISTANT PROFESSOR OF GURUNANAK INSTITUTE
OF TECHNOLOGY)**

DEPARTMENT OF COMPUTER APPLICATIONS

GURUNANAK INSTITUTE OF TECHNOLOGY



TABLE OF CONTENTS

SL.NO	CONTENT TOPICS	PAGE NUMBER
1	CERTIFICATE OF APPROVAL	3
2	ACKNOWLEDGEMENT	4
3	ABSTRACT	5
4	INTRODUCTION	6
5	OBJECTIVE	7
6	FLOW CHART DIAGRAM	8
7	TOOLS AND TECHNOLOGY	9
8	APPROACH	10-24
9	OUTPUTS	25-29
10	EDA OF PARKINSON AND BREAST CANCER DISEASE	30-37
11	PROPOSED METHODOLOGIES	38-43
12	ACCURACY TABLE	44
13	APPLICATIONS	45
14	STRENGTHS	46
15	LIMITATIONS	47
16	CONCLUSION	48
17	FUTURE DEVELOPMENT	49
18	REFERENCES	50
19	INDIVIDUAL CONTRIBUTION	51

CERTIFICATE OF APPROVAL

This is to certify that Project work entitled “Self Health Monitoring System for Parkinson Disease and Breast Cancer Prediction Using Voice Command” Carried out by Ashutosh Kumar Yadav 504121011006, Mujahid Ali Ansari 504121011025, Shahil Kumar Chourasia 504121011045, Sirsha Majumder 504121021050, Sonu Routh 504121011051.under the Mentorship of Prof. Ms. Dola Saha and the wonderful guidance of Prof. Dr. Anjan Maity and Prof Mr. Chiranjib Dutta(H.O.D) of 4th Semester Master of Computer Applications from Gurunanak Institute of Technology, Kolkata 700114 from West Bengal has been satisfactorily completed by them and is thus worthy of acceptance for the degree of Master of Computer Applications.

X

Prof. Ms. Dola Saha

X

Prof. Dr. Anjan Maity

X

Prof. Mr. Chiranjib Dutta(HOD)

ACKNOWLEDGEMENT

We, all the tam members, would like to take this opportunity to express our profound gratitude and kind regards to our guide **Dr. Ananjan Maity** (Assistant Professor – CA) for his exemplary guidance, monitoring, and constant encouragement throughout this project work. The blessings, help and guidance given by him to us shall carry us to a long way in our journeys of life on which we are about to embark.

Signature of Students

Signature of Mentor

ABSTRACT

This project introduces a System that can be used to predict if a patient is affected by **Parkinson Disease** or by **Breast Cancer**. Even today, these diseases are very serious health issues that are persisting.

This system is quite ready to be used by the administration of hospitals as well – the staffs can use it to register new patients and predict if they have symptoms of Parkinson disease or Breast Cancer, send the patients' health reports to their respective mail ids, retrieve health reports of existing patients in database and to discharge patients. Having all these registrations, predictions, etc. done before getting checked by the main doctor cuts off time consumption and makes check-ups easy for doctors as well.

Also, this system can be used by the Staff Head(admin) to access the data of all the working staffs from the database along with all the patients' details and delete staff data if required. The admin only would have a very confidential password for this special access.

The add-on feature of this system is that, we **can use voice commands** to get the actions of the system done. This system is available to be used 24X7 by staffs and admin.

INTRODUCTION

Immediate treatment and accurate diagnosis of neurological disorders is critical for care and research. However, in case of Parkinson Disease (PD) needs special care and treatment to reduce the disability of patients and cope up with changes in their physical capabilities because of this progressive neurodegenerative disorder and for people affected by Breast Cancer (BC) to have an early detection of this disease that could help slow down the progress of the disease and reduce the mortality rate through appropriate therapeutic interventions at the right time.

Parkinson disease (PD) is the most common type of parkinsonism and the second most common progressive neurodegenerative disorder. It affects about 1% of the population over age 50 years and about 2.5% of the population over age 70. The lifetime risk for PD development is 2.0% in men and 1.3% in women. Idiopathic, also known as sporadic PD, is the most common form of PD, affecting primarily older adults. In general, PD is associated with motor symptoms, such as resting tremor, bradykinesia/akinesia, and rigidity because of dopamine deficiency in the basal ganglia due to neurodegeneration of dopaminergic neurons in the substantia nigra *pars compacta* (SNPC). In addition to motor symptoms, nonmotor symptoms and complications, such as neuropsychiatric or neurobehavioral problems, autonomic dysfunction, and sensory problems, are also considered an important part of PD. The disease Cancer starts when normal cell in the breast grow abnormally and out of control, leading to a tumour, which can be malignant or benign. A malignant tumour can grow and spread affecting other body part whereas a benign tumour can grow but does not spread.

A tumour can be cancerous or benign. A cancerous tumour is malignant, meaning it can grow and spread to other parts of the body. A benign tumour means the tumour can grow but has not spread. If breast cancer comes back after initial treatment, it can recur locally, meaning in the same breast and/or regional lymph nodes. It can also recur elsewhere in the body, called a distant recurrence or metastatic recurrence.

Our system has the potential to predict the early onsets of these vital diseases which might give enough time to get it cured or prevent malignancy.

OBJECTIVE

The primary aim of our project this project is to **predict the Parkinson and Breast Cancer diseases** from the given symptoms by patients and create and monitor a health profile of every individual patient.

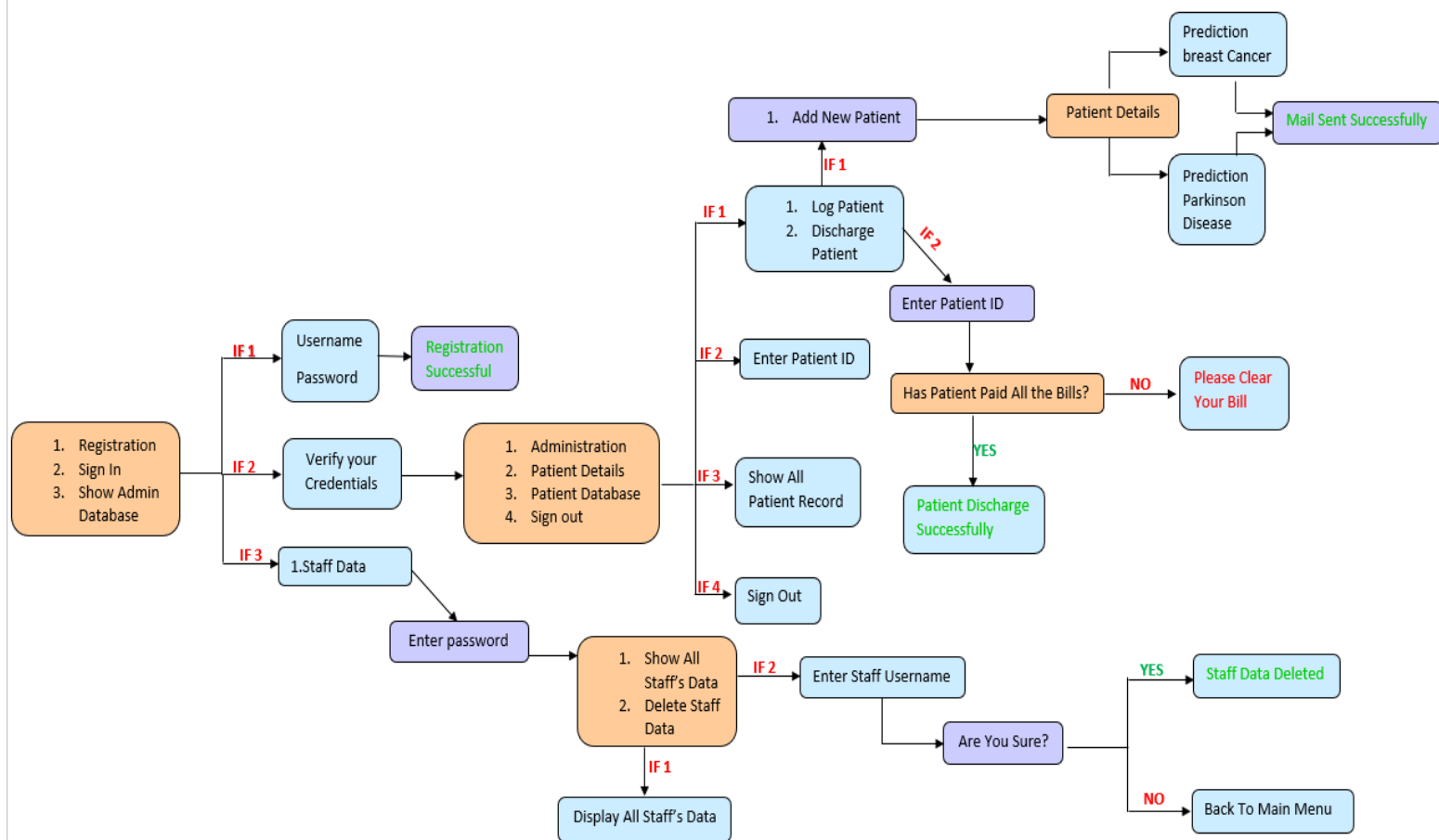
Health is the most important factor for everyone. But unfortunately it has been neglected today for many reasons. Absence of doctor due to some reason during emergency may result in loss of life. Not only that sometimes patient often feel hesitant to go to hospital for minor symptoms. These may prompt into major illness. With the proliferation of technology in health care becomes easier to diagnosis any disease – even the deadliest ones. It is applied in healthcare to identify the clusters of patients, diseases, and future predictions using different machine learning tools. So, this project work proposed a Disease Prediction System that will help user to receive immediate guidance regarding their health issues.

To diagnose any disease, doctor initially analyses the symptoms of the patient and after that the result is predicted. Similarly, machine diagnose the diseases based on the symptoms just like the doctor does. The system is fetched with various symptoms and their disease related with it.

This system **aims to improve disease treatment and its diagnosis in early stages for a faster and better treatment.** Therefore, it is an attempt to make a faster and more accurate disease prediction and **help the physicians for making a reliable decision in a short span of time by increasing efficiency and quality in health management system.**

FLOW CHART DIAGRAM

Disease Prediction System for Parkinson Disease and Breast Cancer using Voice Command



The above flow-chart diagram represents the total workflow of the hospital management system including all operations of patients' database as well as staffs' database and predicts if the patient is suffering from Parkinson Disease or Breast Cancer or any other disease.

TOOLS AND TECHNOLOGY

Tools and Software

- Visual Studio Code:

Visual Studio Code is a code editor that is redefined and optimized for building and debugging modern web and cloud applications. Visual it is free and also available on our favourite platforms – Linux, macOS, and Windows.

- MySql :

MySQL Workbench is a graphical tool that works with MySQL. MySQL Workbench offers an easy-to-use interface to perform multiple tasks involved when working with databases. It integrates SQL development, administration, database design, creation, and maintenance into one visual integrated development environment. MySQL Workbench is similar to SQL Server's SSMS, which is used for administering SQL Server.

Programming Language :

- Python :

Python is a high-level programming language, that precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code.

Skill-based Technology :

- Machine Learning :

Machine learning is a subfield of artificial intelligence which includes the development of algorithms and statistical models that enable computers to improve their performance in tasks through experience. These algorithms and models are designed to learn from data and make predictions or decisions without explicit instructions.

APPROACH

- **Database**

- Patient_db.py
 - insert_patient()
 - show_patientdb()
 - delete_patientdb()
 - show_all_patientdb()

- Staff_db.py
 - insert_staff()
 - show_staff()
 - show_all_staffdb()
 - delete_staff()

- **Predicted Disease**

- Parkinson.py
 - Parkinson()
- Breast_cancer.py
 - Breast_cancer()

- **Send Mail**

- Email.py
 - PDF()
 - Send_mail()

- **Voice Command**

- Command.py
 - takeCommand()
- Speak.py
 - Speak()
- Wish.py
 - wishMe()

- **main.py**

1. Database

➤ Patient_db.py

- **insert_patient()** : This function is used to insert patient details in a database, especially when a new patient is being registered.

```
def insert_patient(patient_id, name, age, sex, address, contact, mail, disease, f_pred, p_pred):  
  
    mycursor.execute("create database if not exists city_hospitals")  
    mycursor.execute("use city_hospitals")  
  
    # creating the tables for storing patient details.  
    mycursor.execute("create table if not exists patient_detail(patient_id int(4) primary key, name varchar(30),sex  
varchar(15),age int(3),address varchar(50),contact varchar(15),mail varchar(40), disease varchar(80), breasr_cancer_prediction  
varchar(20), parkinson_disease_prediction varchar(20))")  
  
    # Inserting Patient Details  
    mycursor.execute("insert into patient_detail values('" + patient_id + "','" + name + "','" + sex + "','" + age + "','" +  
address + "','" + contact + "','" + mail + "','" + disease + "','" + f_pred + "','" + p_pred + "')")  
    mysql.commit()
```

- **show_patientdb()** : This function is used to retrieve patients' records. Staff members can get specific patient details by using their patient ID to keep the track of their health issues.

```
def show_patientdb(patient_id):  
    mycursor.execute("SELECT * FROM city_hospitals.patient_detail where patient_id='" + patient_id + "'")  
    row = mycursor.fetchall()  
    return row
```

- **delete_patientdb()** : This function is used to delete the patient details from the database if it's no longer required.

```
def delete_patientdb(patient_id):  
    mycursor.execute("delete from city_hospitals.patient_detail where patient_id='" + patient_id + "'")  
    mysql.commit()
```

- **show_all_patientdb()** : This function is used to show all the patients' records present in the database.

```
def show_all_patientdb():  
    mycursor.execute("SELECT * FROM city_hospitals.patient_detail")  
    row1 = mycursor.fetchall()  
    return row1
```

➤ Staff_db.py

- **insert_staff()** : This function is used to insert staff details in the database like their username and password.

```
def insert_staff(u, p):

    mycursor.execute("create database if not exists city_hospitals")

    mycursor.execute("use city_hospitals")

    # creating table for storing the username and password of the user
    mycursor.execute("create table if not exists user_data(username varchar(30) primary key,password varchar(30) default'000')")
    mycursor.execute("insert into user_data values('" + u + "', '" + p + "')")
    mysql.commit()
```

- **show_staff()** : This function is used to retrieve all the Staff details from the database.

```
def show_staffdb(un):
    mycursor.execute("select password from city_hospitals.user_data where username='" + un + "'")
    row = mycursor.fetchall()
    return row
```

- **show_all_staffdb()** : This function is used to show all the staff records present in the database.

```
def show_all_staffdb():
    mycursor.execute("select * from city_hospitals.user_data")
    row1 = mycursor.fetchall()
    return row1
```

- **delete_staff()** : This function is used to delete the staff details from the database if it's no longer required.

```
def delete_staffdb(un):
    mycursor.execute("delete from city_hospitals.user_data where username='" + un + "'")
    mysql.commit()
```

2. **Predicted Disease**

➤ **Parkinson.py**

- **Parkinson()** : This function is used to predict the Parkinson Disease.

```

def parkinson():

    df = pd.read_csv('C:/Users/kshah/OneDrive/Desktop/test_major_project/disease_pred/parkinsons.csv')
    # print(df.info())
    # print(df.describe())
    df.isnull().sum()#checking for missing values

    #dropping column axis = 1; dropping row then axis = 0
    #Data Pre-Processing - Separating Features and Target variables according to their Correlation

    df.drop(["name", 'spread1', 'MDVP:Flo(Hz)', 'MDVP:Fhi(Hz)', 'MDVP:Fo(Hz)'], axis=1, inplace=True)
    columns = list(df.columns)
    for column in columns:
        if column == "status":
            continue

        filtered_columns = [column]
        for col in df.columns:
            if (column == col) | (column == "status"):
                continue
            cor_val = df[column].corr(df[col])
            if cor_val > 0.75:
                columns.remove(col)
                continue
            else:
                filtered_columns.append(col)
        df = df[filtered_columns]

    df.isnull().sum() #checking null value

# converting Data in the form of hundred
df.iloc[:,8] = (df.iloc[:, :8]).mul(100).astype(int)

#Plotting Heatmap
# plt.figure(figsize=(25, 7))
# p = sns.heatmap(df.corr(), annot=True)
# plt.show()

# plotting bar figure on STATUS column
# sns.set_style('whitegrid')
# sns.set_context('paper')
# sns.set_palette('GnBu_d')
# a = sns.catplot(x='status', data=df, kind='count')
# plt.title('Number of Samples in Each Class')
# a.set(ylabel='Number of Samples', xlabel='Have Parkinson')

# plt.show()

#histogram
# df.hist(figsize=(25,7))
# plt.show()
#We can see some of the data is normally distributed and most of the attributes are right skewed

# Splitting the data into testing and training set
x_train, x_test, y_train, y_test = train_test_split(df.drop(columns=['status']), df['status'], test_size=0.2, random_state=42)

```

```

# Model Training (DecisionTreeClassifier)
clf = DecisionTreeClassifier()
clf.fit(x_train, y_train)

# # Model Evaluation
# # Accuracy Score

# # Accuracy Score on training data
# x_train_pred = clf.predict(x_train)
# training_data_accuracy = accuracy_score(y_train, x_train_pred)

# print('Accuracy (Training Data) :', training_data_accuracy*100,'%')

# # Accuracy Score on test data
# x_test_pred = clf.predict(x_test)
# testing_data_accuracy = accuracy_score(y_test, x_test_pred)

# print('Accuracy (Testing Data) :', testing_data_accuracy*100,'%')

print("Enter your First nonlinear dynamical complexity measures (142 - 367)")
sp.speak("Enter your First nonlinear dynamical complexity measures (142 - 367)")
D2 = cmd.takeCommand().lower()
print(D2)
sp.speak(D2)

print("Enter your second nonlinear dynamical complexity measures (25 - 68)")
sp.speak("Enter your second nonlinear dynamical complexity measures (25 - 68)")
RPDE = cmd.takeCommand().lower()
print(RPDE)
sp.speak(RPDE)

```

```

print("Enter your third nonlinear measures of fundamental frequency variation (4 - 52)")
sp.speak("Enter your third nonlinear measures of fundamental frequency variation (4 - 52)")
PPE = cmd.takeCommand().lower()
print(PPE)
sp.speak(PPE)

print("Enter your nonlinear fundamental frequency variation (0 - 45)")
sp.speak("Enter your nonlinear fundamental frequency variation (0 - 45)")
spread2 = cmd.takeCommand().lower()
print(spread2)
sp.speak(spread2)

print("Enter your Signal fractal scaling exponent (57 - 82)")
sp.speak("Enter your Signal fractal scaling exponent (57 - 82)")
DFA = cmd.takeCommand().lower()
print(DFA)
sp.speak(DFA)

print("Enter your ratio of noise to tonal components in the voice (844 - 3304)")
sp.speak("Enter your ratio of noise to tonal components in the voice (844 - 3304)")
HNR = cmd.takeCommand().lower()
print(HNR)
sp.speak(HNR)

sp.speak("Enter your Several measures of variation in amplitude(0 - 5)")
Shimar = input('Enter your Several measures of variation in amplitude(0 - 5): ')

sp.speak("Enter your Several measures of variation in fundamental frequency (0 - 3)")
Jitter = input('Enter your Several measures of variation in fundamental frequency (0 - 3): ')

p_pred = clf.predict([[D2, RPDE, PPE, spread2, DFA, HNR,Shimar, Jitter]])
p_pred = ('',).join(str(x) for x in p_pred))

```

```

p_pred = ('',).join(str(x) for x in p_pred))
predicted = ""

if p_pred == 0:
    predicted = 'The patient does not have Parkinson'

else:
    p_pred == 1
    predicted = 'The patient has Parkinson'

return predicted

```

➤ Breast_cancer.py

- **Breast_cancer()** : This function predicts the type of Breast Cancer.

```
def breast_cancer():

    df = pd.read_csv('C:/Users/kshah/OneDrive/Desktop/test_major_project/disease_pred/br.csv')

    # histogram before applying feature engineering
    # df.hist(figsize=(90,90))
    # plt.show()

    # Plotting Heatmap(before feature engineering)
    # plt.figure(figsize=(25, 7))
    # p = sns.heatmap(df.corr(), annot=True)
    # plt.show()

    # Feature Selection
    # a1 = body_p.corr()
    # x1 = a1['class'].sort_values(ascending=False)
    df = df.iloc[:, 1:6]
    x_train, x_test, y_train, y_test = train_test_split(df.drop(columns=['diagnosis']), df['diagnosis'], test_size=0.2, random_state=42)
    clf = svm.SVC(kernel='linear')
    clf.fit(x_train, y_train)

    # Accuracy Score on training data
    # x_train_pred = clf.predict(x_train)
    # training_data_accuracy = accuracy_score(y_train, x_train_pred)
    # print('Accuracy (Training Data) :', training_data_accuracy*100,'%')

    # # Accuracy Score on test data
    # x_test_pred = clf.predict(x_test)
    # testing_data_accuracy = accuracy_score(y_test, x_test_pred)
    # print('Accuracy (Testing Data) :', testing_data_accuracy*100,'%')

    # histogram after applying feature engineering
    # df.hist(figsize=(25,7))
    # plt.show()
```

```
# plotting bar figure on Diagnosis column after feature engineering
# sns.set_style('whitegrid')
# sns.set_context('paper')
# sns.set_palette('magma')
# a = sns.catplot(x='diagnosis', data=df, kind='count')
# plt.title('Number of Samples in Each Class')
# a.set(ylabel='Number of Samples', xlabel='M = Malignant, B = Benign')
# plt.show()
```

```
print("Please enter radius_mean Range(6.981 - 28.11): ")
sp.speak("Please enter radius_mean Range(6.981 - 28.11): ")
radius_mean = cmd.takeCommand().lower()
print(radius_mean)
sp.speak(radius_mean)
```

```
print("Please enter texture_mean (9.71 - 39.28): ")
sp.speak("Please enter texture_mean (9.71 - 39.28): ")
texture_mean = cmd.takeCommand().lower()
print(texture_mean)
sp.speak(texture_mean)
```

```
print("Please enter perimeter_mean (43.79 - 188.5): ")
sp.speak("Please enter perimeter_mean (43.79 - 188.5): ")
perimeter_mean = cmd.takeCommand().lower()
print(perimeter_mean)
sp.speak(perimeter_mean)
```

```
print("Please enter area_mean(143.5 - 2501): ")
sp.speak("Please enter area_mean(143.5 - 2501): ")
area_mean = cmd.takeCommand().lower()
print(area_mean)
sp.speak(area_mean)
```

```
preds = clf.predict([[radius_mean, texture_mean, perimeter_mean, area_mean]])
f_pred = (' '.join(preds))

if f_pred == 'B':
    f_pred = 'Benign'
else:
    f_pred = 'Malignant'
sp.speak(f_pred)
return f_pred
```

3. Send Mail

➤ Email.py

- **PDF()** : This function is used to create patient reports in pdf format.

```
class PDF(FPDF):  
    def header(self):  
        #logo  
        self.image('C:/Users/kshah/OneDrive/Desktop/test_major_project/mail/code.jpg', 8, 8, 15)  
        self.image('C:/Users/kshah/OneDrive/Desktop/test_major_project/mail/R.jpg', 187, 7, 15)  
        #font  
        self.add_font('Lucida Bright','',r'C:/Windows/Fonts/LCALLIG.TTF', uni=True) # uni = True does true type font subset embedding  
        self.set_font('Lucida Bright','U',20)  
        self.set_text_color(134,108,15)  
        # Title  
        self.cell(0,10,'Disease prediction System for',ln=True, align='C',)  
        self.cell(0,10,'Breast Cancer and Parkinson ',ln=True, align='C',)  
        self.cell(0,10,'Disease Using Voice Command',ln=True, align='C',)  
        #line break  
        self.ln(20)
```

```
# Create Object  
pdf = PDF('P','mm')  
# Add a page  
pdf.add_page()  
  
#Specify Fonts ('times','courier' etc)  
# 'B'(bold), 'U' (underline), ' I'(Italics), ''(regular),'BU' (combination)  
# Font size  
pdf.set_font('courier', 'U', 16)  
  
pdf.add_font('Goudy Old Style','',r'C:/Windows/Fonts/GOUDOSB.TTF', uni=True) # uni = True does true type font subset embedding  
pdf.set_font('Goudy Old Style', '', 16)  
# pdf.set_text_color(0,0,0)  
# Add Text  
# w = 'width'  
# h = 'height'  
pdf.cell(0,10, f'Patient ID: {patient_id}', ln=1)  
pdf.cell(0,10, f'Name: {name}', ln=1)  
pdf.cell(0,10, f'Age: {age}', ln=1)
```

```
mail > email.py > send_mail  
60 pdf.cell(0,10, f'Gender: {sex}', ln=1)  
61 pdf.cell(0,10, f'Address: {address}', ln=1)  
62 pdf.cell(0,10, f'Contact: +91{contact}', ln=1)  
63 pdf.cell(0,10, f'Mail: {mail}', ln=1)  
64 pdf.cell(0,10, f'Disease: {disease}', ln=1)  
65 pdf.cell(0,10, f'Breast Cancer Prediction: {f_pred}', ln=1)  
66 pdf.cell(0,10, f'Parkinson Disease Prediction: {p_pred}', ln=1)  
67 y_axis_initial = 255  
68 pdf.set_y(y_axis_initial)  
69 pdf.set_font('times', 'U', 16)  
70 pdf.cell(0,10,"Doctor's Sign with Date", ln = 1, align='R')  
71 a = "C:/Users/kshah/OneDrive/Desktop/test_major_project/Paitent_Details_PDF/"  
72 pdf.output(f"{a}{name}.pdf")  
73
```

- **Send mail()** : This function is used to automatically send the PDF report to the patient via email.


```

email_sender = "shahil.official.college@gmail.com"
pwd_sender = os.environ.get("Email_Password")
receiver = mail
# msg = MIMEMultipart()

subject = f'{name} Health Report'

body = ''
em = EmailMessage()
em['From'] = email_sender

em['To'] = receiver
em['Subject'] = subject

em.set_content(body)
context = ssl.create_default_context()

files = f"C:/Users/kshah/OneDrive/Desktop/test_major_project/Paitent_Details_PDF/{name}.pdf"
with open (files, 'rb') as m:
    file_data = m.read()
    file_name = name + ".pdf"
em.add_attachment(file_data, maintype='pdf', subtype = 'octet-stream', filename = file_name)
with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
    smtp.login(email_sender, pwd_sender)
    smtp.sendmail(
        email_sender, receiver, em.as_string())

```

4. Voice Command

➤ **Command.py**

- **takeCommand()** : This function is used to take Voice Commands from the user.

```

import speech_recognition as sr # pip install speechRecognition

def takeCommand():
    # It takes microphone input from the user and returns string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...",source)
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en - in')

    except Exception as e:
        # print(e)
        print("Say that again please...")
        return "None"

    return query

```

➤ **Speak.py**

- **Speak()** :

```
import pyttsx3 # pip install pyttsx3

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[1].id)

def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

➤ **Wish.py**

- **wishMe()** : This function is used to wish or greet.

```
import datetime
import voice.speak as sp

def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        sp.speak("Good Morning!")
    elif hour >= 12 and hour < 18:
        sp.speak("Good Afternoon")
    else:
        sp.speak("Good Evening")
```

5. **main.py**

```

1 import os
2 import voice.speak as sp
3 import voice.wish as ws
4 import mail.email as email
5 import disease_pred.breast_cancer as b_cancer
6 import disease_pred.parkinson as park
7 import database.patient_db as p_details
8 import database.staff_db as sd
9
10 while (True):
11
12
13     Admin_passwd = os.environ.get("Admin_Password")
14
15
16
17     while (True):
18         ws.wishMe()
19         print("Disease Prediction System for Parkinson Disease and Breast Cancer using Voice Command")
20         sp.speak("Disease Prediction System for Parkinson Disease and Breast Cancer using Voice Command")
21
22         print("""
23         |
24         |
25         |
26         |
27         |
28         |
29         |
30         |
31         |
32         |
33         |
34         |
35         |
36         |
37         |
38         |
39         |
40         |
41         |
42         |
43         |
44         |
45         |
46         |
47         |
48         |
49         |
50         |
51         |
52         |
53         |
54         |
55         |
56         |
57         |
58         |
59         |
60         |
61         |
62         |
63         |
64         |
65         |
66         |
67         |
68         |
69         |
70         |
71         |
72         |
73         |
74         |
75         |
76         |
77         |
78         |
79         |
80         |
81         |
82         |
83         |
84         |
85         |
86         |
87         |
88         |
89         |
90         |
91         |
92         |
93         |
94         |
95         |
96         |
97         |
98         |
99         |
100        |
101        |
102        |
103        |
104        |
105        |
106        |
107        |
108        |
109        |
110        |
111        |
112        |
113        |
114        |
115        |
116        |
117        |
118        |
119        |
120        |
121        |
122        |
123        |
124        |
125        |
126        |
127        |
128        |
129        |
130        |
131        |
132        |
133        |
134        |
135        |
136        |
137        |
138        |
139        |
140        |
141        |
142        |
143        |
144        |
145        |
146        |
147        |
148        |
149        |
150        |
151        |
152        |
153        |
154        |
155        |
156        |
157        |
158        |
159        |
160        |
161        |
162        |
163        |
164        |
165        |
166        |
167        |
168        |
169        |
170        |
171        |
172        |
173        |
174        |
175        |
176        |
177        |
178        |
179        |
180        |
181        |
182        |
183        |
184        |
185        |
186        |
187        |
188        |
189        |
190        |
191        |
192        |
193        |
194        |
195        |
196        |
197        |
198        |
199        |
200        |
201        |
202        |
203        |
204        |
205        |
206        |
207        |
208        |
209        |
210        |
211        |
212        |
213        |
214        |
215        |
216        |
217        |
218        |
219        |
220        |
221        |
222        |
223        |
224        |
225        |
226        |
227        |
228        |
229        |
230        |
231        |
232        |
233        |
234        |
235        |
236        |
237        |
238        |
239        |
240        |
241        |
242        |
243        |
244        |
245        |
246        |
247        |
248        |
249        |
250        |
251        |
252        |
253        |
254        |
255        |
256        |
257        |
258        |
259        |
260        |
261        |
262        |
263        |
264        |
265        |
266        |
267        |
268        |
269        |
270        |
271        |
272        |
273        |
274        |
275        |
276        |
277        |
278        |
279        |
280        |
281        |
282        |
283        |
284        |
285        |
286        |
287        |
288        |
289        |
290        |
291        |
292        |
293        |
294        |
295        |
296        |
297        |
298        |
299        |
300        |
301        |
302        |
303        |
304        |
305        |
306        |
307        |
308        |
309        |
310        |
311        |
312        |
313        |
314        |
315        |
316        |
317        |
318        |
319        |
320        |
321        |
322        |
323        |
324        |
325        |
326        |
327        |
328        |
329        |
330        |
331        |
332        |
333        |
334        |
335        |
336        |
337        |
338        |
339        |
340        |
341        |
342        |
343        |
344        |
345        |
346        |
347        |
348        |
349        |
350        |
351        |
352        |
353        |
354        |
355        |
356        |
357        |
358        |
359        |
360        |
361        |
362        |
363        |
364        |
365        |
366        |
367        |
368        |
369        |
370        |
371        |
372        |
373        |
374        |
375        |
376        |
377        |
378        |
379        |
380        |
381        |
382        |
383        |
384        |
385        |
386        |
387        |
388        |
389        |
390        |
391        |
392        |
393        |
394        |
395        |
396        |
397        |
398        |
399        |
400        |
401        |
402        |
403        |
404        |
405        |
406        |
407        |
408        |
409        |
410        |
411        |
412        |
413        |
414        |
415        |
416        |
417        |
418        |
419        |
420        |
421        |
422        |
423        |
424        |
425        |
426        |
427        |
428        |
429        |
430        |
431        |
432        |
433        |
434        |
435        |
436        |
437        |
438        |
439        |
440        |
441        |
442        |
443        |
444        |
445        |
446        |
447        |
448        |
449        |
450        |
451        |
452        |
453        |
454        |
455        |
456        |
457        |
458        |
459        |
460        |
461        |
462        |
463        |
464        |
465        |
466        |
467        |
468        |
469        |
470        |
471        |
472        |
473        |
474        |
475        |
476        |
477        |
478        |
479        |
480        |
481        |
482        |
483        |
484        |
485        |
486        |
487        |
488        |
489        |
490        |
491        |
492        |
493        |
494        |
495        |
496        |
497        |
498        |
499        |
500        |
501        |
502        |
503        |
504        |
505        |
506        |
507        |
508        |
509        |
510        |
511        |
512        |
513        |
514        |
515        |
516        |
517        |
518        |
519        |
520        |
521        |
522        |
523        |
524        |
525        |
526        |
527        |
528        |
529        |
530        |
531        |
532        |
533        |
534        |
535        |
536        |
537        |
538        |
539        |
540        |
541        |
542        |
543        |
544        |
545        |
546        |
547        |
548        |
549        |
550        |
551        |
552        |
553        |
554        |
555        |
556        |
557        |
558        |
559        |
560        |
561        |
562        |
563        |
564        |
565        |
566        |
567        |
568        |
569        |
570        |
571        |
572        |
573        |
574        |
575        |
576        |
577        |
578        |
579        |
580        |
581        |
582        |
583        |
584        |
585        |
586        |
587        |
588        |
589        |
590        |
591        |
592        |
593        |
594        |
595        |
596        |
597        |
598        |
599        |
600        |
601        |
602        |
603        |
604        |
605        |
606        |
607        |
608        |
609        |
610        |
611        |
612        |
613        |
614        |
615        |
616        |
617        |
618        |
619        |
620        |
621        |
622        |
623        |
624        |
625        |
626        |
627        |
628        |
629        |
630        |
631        |
632        |
633        |
634        |
635        |
636        |
637        |
638        |
639        |
640        |
641        |
642        |
643        |
644        |
645        |
646        |
647        |
648        |
649        |
650        |
651        |
652        |
653        |
654        |
655        |
656        |
657        |
658        |
659        |
660        |
661        |
662        |
663        |
664        |
665        |
666        |
667        |
668        |
669        |
670        |
671        |
672        |
673        |
674        |
675        |
676        |
677        |
678        |
679        |
680        |
681        |
682        |
683        |
684        |
685        |
686        |
687        |
688        |

```

```

38 sp.speak(("Please Enter username..."))
39 u = input("Enter username!!!:")
40
41
42 sp.speak("Please Enter password(Password Must Be Strong)...")
43 p = input("Enter Your password (Password must be strong!!!: ")
44
45 show_userdb = sd.show_staffdb(u)
46
47 if len(show_userdb):
48     print("Username Already Registered choose different username")
49     sp.speak("Username Already Registered choose different username")
50 else:
51     sd.insert_staff(u, p)
52
53     print("""
54 =====
55 !!Well Done!!Registration Done Successfully!!
56 =====
57 | | | | | | | | | | | | | | """)
58     sp.speak("Registration Done Successfully...")
59
60 sp.speak("Press any key to continue")
61 os.system("pause")
62
63
64 # IF USER WANTS TO LOGIN
65 elif r == 2:
66     print("""
67 =====
68 !!!!!!! {{Sign In}} !!!!!!!
69 =====
70 | | | | | | | | | | | | | | """)
71
72     print("Please Verify Your Credentials")
73     sp.speak("Please Verify Your Credentials")

```

```
main.py > ...
75 print("Please Enter Your Username")
76 sp.speak("Please Enter Your Username")
77 un = input("Username!!: ")
78
79
80 print("Please Enter Your Password")
81 sp.speak("Please Enter Your Password")
82 ps = input("Password!!: ")
83
84
85
86 row = sd.show_staffdb(un)
87 for i in row:
88     a = list(i)
89     if a[0] == str(ps):
90         while(True):
91             print("""
92                 1. Administration
93                 2. Patient(Details)
94                 3. Show patient database
95                 4. Sign Out
96             """)
97             sp.speak("""
98                 press 1 for Administration
99                 press 2 for Patient(Details)
100                press 3 for Show patient database
101                press 4 for Sign Out
102            """)
103
104             a = int(input("ENTER YOUR CHOICE: "))
105             if a == 1:
106                 print("""
107                     1. Log patient Record
108                     2. Discharge Summary
109                 """)
110
111             sp.speak("""
```

```
main.py > ...
110
111 sp.speak("""
112     press 1 for Log patient Record
113     press 2 for Discharge Summary
114 """)
115
116 x = int(input("ENTER YOUR CHOICE: "))
117 if x == 1:
118
119     print("""
120         1. Add New Patient
121     """)
122
123     sp.speak("press 1 Add New Patient")
124     b = int(input("Enter Your Choice: "))
125
126     # adding new patient
127     if b == 1:
128
129         print("Please Enter Your patient ID")
130         sp.speak("Please Enter Your patient ID")
131         patient_id = input("Patient ID: ")
132
133         print("Please Enter Your Name")
134         sp.speak("Please Enter Your Name")
135         name = input("Name: ")
136
137         print("Please Enter Your Gender")
138         sp.speak("Please Enter Your Gender")
139         sex = input("Gender: ")
140
141         print("Please Enter your Age")
142         sp.speak("Please Enter your Age")
143         age = input("Age: ")
144
145         print("Please Enter Your Address")
146         sp.speak("Please Enter Your Address")
```

```
main.py > ...
147 address = input("Address: ")
148
149 print("Please Enter Your Contact Number")
150 sp.speak("Please Enter Your Contact Number")
151 contact = input("Contact Details: ")
152
153 print("Please Enter Your Email")
154 sp.speak("Please Enter Your Email")
155 mail = input("Mail Id: ")
156
157 print("Please Enter Your Disease")
158 sp.speak("Please Enter Your Disease")
159 disease = input("Disease: ")
160
161 print("Prediction Breast Cancer?")
162 sp.speak("Prediction Breast Cancer?")
163 br_cancer = input("prediction breast cancer? (y/n): ")
164 f_pred = "NA"
165
166 if br_cancer == "y":
167     f_pred = b_cancer.breast_cancer()
168
169 print("Prediction parkinson Disease")
170 sp.speak("prediction parkinson Disease")
171 p_cancer = input("prediction parkinson disease? (y/n): ")
172 p_pred = "NA"
173
174
175 if p_cancer == "y":
176     print("parkinson abcd")
177     p_pred = park.parkinson()
178
179
180
181 p_details.insert_patient(patient_id, name, age, sex, address, contact, mail, disease, f_pred,
182 p_pred)
183
Ln 179, Col 36 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier
main.py > ...
184 print("""
185     =====
186     !!!!!Registered Successfully!!!!!!
187     =====
188     """)
189 sp.speak("Patient Registered Sucessfully")
190
191 sp.speak("press Any Key To Continue")
192 os.system("pause")
193
194
195
196
197 email.send_mail(patient_id, name, age, sex, address, contact, mail, disease, f_pred, p_pred)
198 sp.speak("Mail Sent Sucessfully")
199
200
201 else:
202     print("please Choose Valid Option")
203     sp.speak("Please Choose Valid Option")
204
205 # dischare process
206 elif x == 2:
207
208     # print("Please Enter The Patient Name")
209     sp.speak("Please Enter The Patient ID")
210     patient_id = input("Enter The Patient ID: ")
211
212
213
214 row = p_details.show_patientdb(patient_id)
215 if len(row):
216     for i in row:
217         b1 = 0
218         v1 = list(i)
219         k1 = ["PATIENT ID", "NAME", "SEX", "AGE", "ADDRESS", "CONTACT",
220             "MAIL", "DISEASE", "BREAST CANCER PREDICTION", "PARKINSON DISEASE PRIDITION"]
221
Ln 179, Col 36 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier
```

main.py > ...

```
221         d1 = dict(zip(k1, v1))
222         print(d1)
223     else:
224         print("patient Dose not Exist")
225         sp.speak("patient Dose not Exist")
226
227         sp.speak("press any key to continue")
228         os.system("pause")
229         break
230
231
232     print("Has Patient Paid all the bills")
233     sp.speak("Has Patient Paid all the Bills?")
234     bill = input("Has he paid all the bills? (y/n):")
235
236     if bill == "y":
237         p_details.delete_patientdb(patient_id)
238         sp.speak("Patient Discharged Sucessfully")
239     else:
240
241         print("please clear your bill")
242         sp.speak("Please Clear Your Bill")
243
244
245     else:
246
247         print("Please Choose Valid Option")
248         sp.speak("Please Choose Valid Option")
249
250         sp.speak("Press Any Key To Continue")
251         os.system("pause")
252
253
254     # if user wants to see the details of PATIENT
255     elif a == 2:
256
257         # print("please Enter patient name")
258         sp.speak("please Enter patient ID")
```

Ln 179, Col 36 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier

main.py > ...

```
258     patient_id = input("Enter The Patient ID: ")
259
260
261     row = p_details.show_patientdb(patient_id)
262     if len(row):
263         for i in row:
264             b = 0
265             v = list(i)
266             k = ["PATIENT ID", "NAME", "SEX", "AGE", "ADDRESS", "CONTACT",
267                 "MAIL", "DISEASE", "BREAST CANCER PREDICTION", "PARKINSON DISEASE PRIDITION"]
268             d = dict(zip(k, v))
269             print(d)
270     else:
271         print("patient Dose not Exist in our database")
272         sp.speak("patient Dose not Exist in our database")
273
274         os.system("pause")
275         sp.speak("press any key to continue")
276
277
278     # if user wants to show all patient records
279     elif a == 3:
280
281         row1 = p_details.show_all_patientdb()
282         for i in row1:
283             b = 0
284             v = list(i)
285             k = ["PATIENT ID", "NAME", "SEX", "AGE", "ADDRESS", "CONTACT",
286                 "MAIL", "DISEASE", "BREAST CANCER PREDICTION", "PARKINSON DISEASE PRIDITION"]
287             d = dict(zip(k, v))
288             print(d)
289
290     # SIGN OUT
291     elif a == 4:
292         break
293
294     else:
```

Ln 179, Col 36 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier

```

main.py > ...
295         print("Choose Valid Option")
296         sp.speak("choose Valid Option")
297
298     # IF THE USERNAME AND PASSWORD IS NOT IN THE DATABASE
299     else:
300         print("Staff Dosen't Exist in our database")
301         sp.speak("Staff Dosen't Exist in our database")
302         break
303
304     elif r == 3:
305
306         print("""
307         1. Staff's Data
308         """)
309         sp.speak("press 1 for Staff's Data")
310
311         i = int(input("Enter Your Choice: "))
312         if i == 1:
313
314             # print("Please Enter Your Password")
315             sp.speak("Please Enter Your Password")
316             pwd = str(input("Enter Your Password: "))
317             if pwd == Admin_passwd:
318                 while(True):
319                     print("""
320                     1. Show Staff's Data
321                     2. Delete staff's data
322                     """)
323
324                     sp.speak("""
325                     press 1 Show Staff's Data
326                     press 2 Delete staff's data
327                     """)
328
329
330
331

```

```

main.py > ...
332         x = int(input("Enter your Choice: "))
333         if x == 1:
334
335             row1 = sd.show_all_staffdb()
336             for i in row1:
337                 b = 0
338                 v = list(i)
339                 k = ["USERNAME", "PASSWORD"]
340                 d = dict(zip(k, v))
341                 print(d)
342                 sp.speak("press any key to continue")
343                 os.system("pause")
344
345             break
346
347         elif x == 2:
348
349             sp.speak("Please Enter The Staff's Username")
350             un = input("Enter the Staff Username: ")
351
352             sp.speak("Are you Sure")
353             sure = input("Are You Sure? (y/n):")
354
355             if sure == "y":
356                 sd.delete_stafftdb(un)
357                 print("Successfully Staff deleted")
358                 sp.speak("Staff Sucessfully Deleted")
359             else:
360                 print("Staff Not Found")
361                 sp.speak("Staff Not Found")
362
363             sp.speak("Press Any Key To continue")
364

```

main.py > ...

```
369         os.system("pause")
370         break
371
372     else:
373         print("Please Choose Valid Option")
374         sp.speak("Please Choose valid Option")
375
376     else:
377         print("Invalid Password")
378         sp.speak("Invalid Password")
379         break
380
381     else:
382         print("Please Choose Valid Option")
383         sp.speak("Please Choose Valid Option")
384
385     else:
386         print("Please Choose Valid Option")
387         sp.speak("Please Choose Valid Option")
```


OUTPUTS

Disease Prediction System for Parkinson Disease and Breast Cancer using Voice Command

1. Registration
2. Sign In
3. Show Admin database

enter your choice: 2

```
=====
!!!!!!!  {{Sign In}}  !!!!!!!!
=====
```

Please Verify Your Credentials

Please Enter Your Username

Username!!: Sirsha

Please Enter Your Password

Password!!: Sirsha@123

1. Administration
2. Patient(Details)
3. Show patient database
4. Sign Out

ENTER YOUR CHOICE: 1

1. Log patient Record
2. Discharge Summary

ENTER YOUR CHOICE: 1

1. Add New Patient

Enter Your Choice: 1

Please Enter Your patient ID

Patient ID: 34

Please Enter Your Name

```
Name: shah
Please Enter Your Gender
Gender: m
Please Enter your Age
Age: 5
Please Enter Your Address
Address: j
Please Enter Your Contact Number
Contact Details: 45
Please Enter Your Email
Mail Id: a@gmail.com
Please Enter Your Disease
Disease: d
Prediction Breast Cancer?
prediction breast cancer? (y/n): y
Please enter radius_mean Range(6.981 - 28.11):
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.94908917, 'transcript': '7.21'},
                  {'transcript': '7.1'},
                  {'transcript': '7.2 1'},
                  {'transcript': '$7.21'},
                  {'transcript': '7.2 one'}],
  'final': True}
7.21
Please enter texture_mean (9.71 - 39.28):
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.97219545, 'transcript': '11.1'},
                  {'transcript': '$11.1'}],
  'final': True}
11.1
Please enter perimeter_mean (43.79 - 188.5):
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.90421665, 'transcript': '44.21'},
                  {'transcript': '44.1'},
                  {'transcript': '44.2 1'},
                  {'transcript': '44.201'},
                  {'transcript': '44.2 on'}],
  'final': True}
44.21
```

Ln 179, Col 37 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier

```
Please enter area_mean(143.5 - 2501):
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.96417558, 'transcript': '181.21'},
                  {'transcript': '181.1'},
                  {'transcript': '$181.21'},
                  {'transcript': '181.2 1'},
                  {'transcript': '180 1.21'}],
  'final': True}
181.21
C:\Users\kshah\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names,
but SVC was fitted with feature names
warnings.warn(
Prediction parkinson Disease
prediction parkinson disease? (y/n): y
parkinson abcd
Enter your First nonlinear dynamical complexity measures (142 - 367)
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.83602929, 'transcript': '150'},
                  {'transcript': '1:50'},
                  {'transcript': '$150'},
                  {'transcript': '$1.50'},
                  {'transcript': '$15'}],
  'final': True}
150
Enter your second nonlinear dynamical complexity measures (25 - 68)
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.97219545, 'transcript': '44'},
                  {'transcript': '44th'}],
  'final': True}
44
Enter your third nonlinear measures of fundamental frequency variation (4 - 52)
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.97219545, 'transcript': '44'},
                  {'transcript': '44th'},
                  {'transcript': '440'}],
  'final': True}
```

Ln 179, Col 37 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier

```

44
Enter your nonlinear fundamental frequency variation (0 - 45)
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.96723294, 'transcript': '34'},
                  {'transcript': 'party for'},
                  {'transcript': '34th'},
                  {'transcript': '30 for'},
                  {'transcript': '33'}],
  'final': True}
34
Enter your Signal fractal scaling exponent (57 - 82)
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.95189661, 'transcript': '64'},
                  {'transcript': '54'},
                  {'transcript': '"64"'}],
  'final': True}
64
Enter your ratio of noise to tonal components in the voice (844 - 3304)
Listening...
Recognizing...
result2:
{ 'alternative': [ {'confidence': 0.88586742, 'transcript': '1100'},
                  {'transcript': '1,100'},
                  {'transcript': '11:00'},
                  {'transcript': '$1100'},
                  {'transcript': '$1,100'}],
  'final': True}
1100
Enter your Several measures of variation in amplitude(0 - 5): 2
Enter your Several measures of variation in fundamental frequency (0 - 3): 1
Affected

```

```

=====
!!!!!!Registered Successfully!!!!!!
=====

```

Press any key to continue . . .

1. Administration
2. Patient(Details)
3. Show patient database
4. Sign Out

```

ENTER YOUR CHOICE: 3
{'PATIENT ID': 11, 'NAME': 'shahil', 'SEX': 'male', 'AGE': 24, 'ADDRESS': 'bandel', 'CONTACT': '8420179104', 'MAIL': 'qwerty@gmail.com', 'DISEASE': 'fever', 'BREAST CANCER PREDICTION': 'NA', 'PARKINSON DISEASE PRIDITION': 'NA'}
{'PATIENT ID': 15, 'NAME': 'Shiwani', 'SEX': 'female', 'AGE': 25, 'ADDRESS': 'badel', 'CONTACT': '8910414657', 'MAIL': 'kshahil1999@gmail.com', 'DISEASE': 'fever', 'BREAST CANCER PREDICTION': 'NA', 'PARKINSON DISEASE PRIDITION': 'NA'}
{'PATIENT ID': 17, 'NAME': 'shahil', 'SEX': 'male', 'AGE': 24, 'ADDRESS': 'bandel', 'CONTACT': '456', 'MAIL': 'kshahil1999@gmail.com', 'DISEASE': 'qwerty', 'BREAST CANCER PREDICTION': 'NA', 'PARKINSON DISEASE PRIDITION': 'NA'}
{'PATIENT ID': 19, 'NAME': 'shahil', 'SEX': 'male', 'AGE': 24, 'ADDRESS': 'bandel', 'CONTACT': '456987123', 'MAIL': 'kshahil1999@gmail.com', 'DISEASE': 'j', 'BREAST CANCER PREDICTION': 'NA', 'PARKINSON DISEASE PRIDITION': 'NA'}
{'PATIENT ID': 31, 'NAME': 'ava', 'SEX': 'm', 'AGE': 5, 'ADDRESS': 'm', 'CONTACT': '5', 'MAIL': 'a@gmail.com', 'DISEASE': 'v', 'BREAST CANCER PREDICTION': 'NA', 'PARKINSON DISEASE PRIDITION': 'NA'}
{'PATIENT ID': 36, 'NAME': 'abcd', 'SEX': 'male', 'AGE': 24, 'ADDRESS': 'howrah', 'CONTACT': '456321789', 'MAIL': 'a@gmail.com', 'DISEASE': 'cancer', 'BREAST CANCER PREDICTION': 'Benign', 'PARKINSON DISEASE PRIDITION': 'Affected'}

```

1. Administration
2. Patient(Details)
3. Show patient database
4. Sign Out

```

ENTER YOUR CHOICE: 2
Enter The Patient ID: 36
{'PATIENT ID': 36, 'NAME': 'abcd', 'SEX': 'male', 'AGE': 24, 'ADDRESS': 'howrah', 'CONTACT': '456321789', 'MAIL': 'a@gmail.com', 'DISEASE': 'cancer', 'BREAST CANCER PREDICTION': 'Benign', 'PARKINSON DISEASE PRIDITION': 'Affected'}

```

1. Administration
2. Patient(Details)
3. Show patient database
4. Sign Out

ENTER YOUR CHOICE: 1

1. Log patient Record
2. Discharge Summary

ENTER YOUR CHOICE: 2

Enter The Patient ID: 31

```
{'PATIENT ID': 31, 'NAME': 'ava', 'SEX': 'm', 'AGE': 5, 'ADDRESS': 'm', 'CONTACT': '5', 'MAIL': 'a@gmail.com', 'DISEASE': 'v', 'BREAST CANCER PREDICTION': 'NA', 'PARKINSON DISEASE PRIDITION': 'NA'}
```

Has Patient Paid all the bills

Has he paid all the bills? (y/n):n

please clear your bill

ENTER YOUR CHOICE: 1

Self Health Monitoring System For Breast Cancer and Parkinson disease using Voice Command

1. Registration
2. Sign In
3. Show Admin database

enter your choice: 3

1. Staff's Data

Enter Your Choice: 1

Enter Your Password: Shahil@1999

1. Show Staff's Data
2. Delete staff's data

Enter your Choice: 1

```
{'USERNAME': 'shahil', 'PASSWORD': '123'}  
{'USERNAME': 'sirsha', 'PASSWORD': '123'}
```

Press any key to continue . . .

Ln 239, Col 42 Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live Prettier

Health Report



Disease prediction System for Breast Cancer and Parkinson Disease Using Voice Command



Patient ID: 42

Name: Sirsha Majumder

Age: 24

Gender: Female

Address: Haldia

Contact: +91987456321

Mail: sirshalm10@gmail.com

Disease: Breast Cancer , Parkinson Disease

Breast Cancer Prediction: Benign

Parkinson Disease Prediction: Affected

Doctor's Sign with Date

EDA OF PARKINSON AND BREAST CANCER DISEASE

PARKINSON DISEASE:

Features Information:

Jitter - Several measures of variation in fundamental frequency

Shimmer, - Several measures of variation in amplitude

HNR - measures of ratio of noise to tonal components in the voice

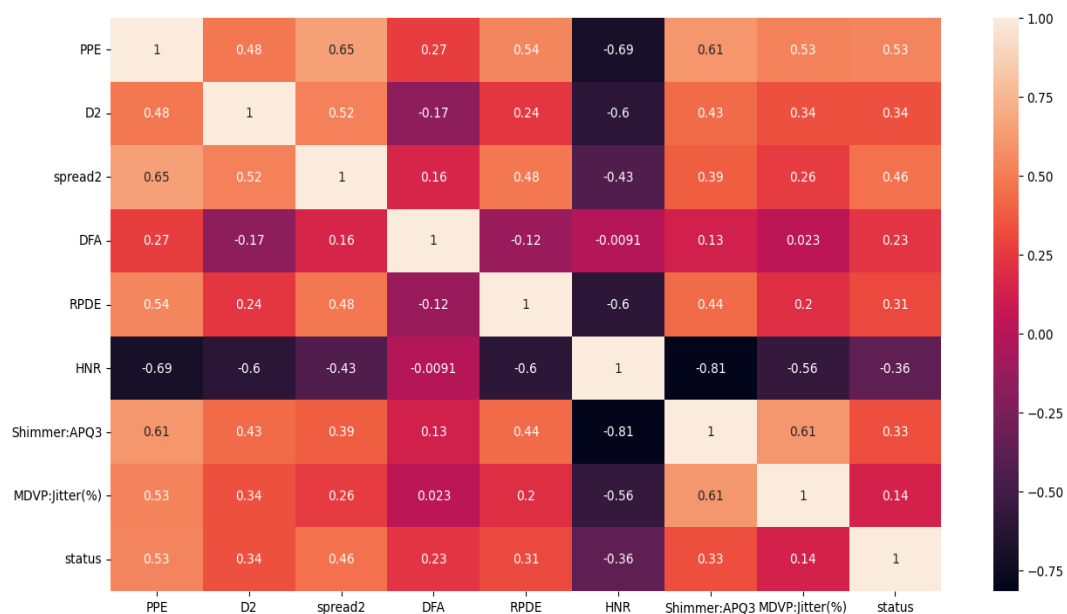
status - Health status of the subject (one) - Parkinson's, (zero) - healthy

RPDE, D2 - Two nonlinear dynamical complexity measures

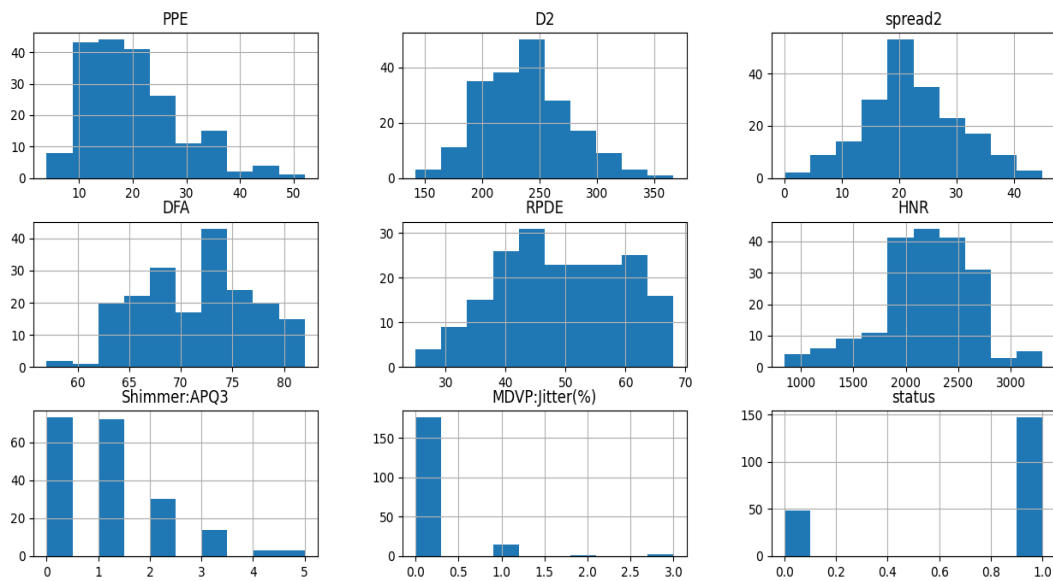
DFA - Signal fractal scaling exponent

spread2, PPE - Two nonlinear measures of fundamental frequency variation

Heatmap:



Histogram:



We can see some of the data is normally distributed and most of the attributes are right skewed.

Accuracy Score:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Python + v [Icons] ... ^ X

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kshah\OneDrive\Desktop\Machine Learning> & C:\Users\kshah\AppData\Local\Programs\Python\Python311\python.exe "c:/Users/kshah/OneDrive/Desktop/Machine Learning/parkinson.py"
Accuracy (Training Data) : 100.0 %
Accuracy (Testing Data) : 92.3076923076923 %
PS C:\Users\kshah\OneDrive\Desktop\Machine Learning>

```

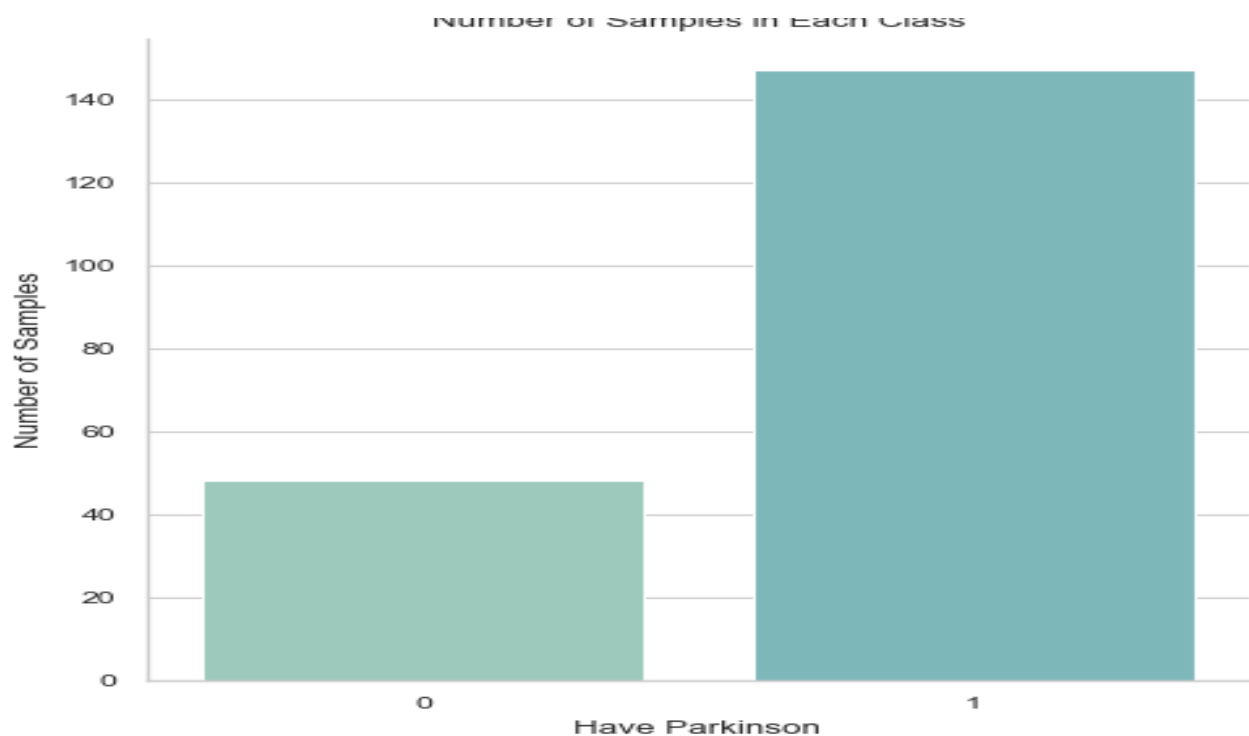
As you can see, we are using `DecisionTreeClassifier` the accuracy score in Training Data is 100% and in Testing Data is 92%.

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Enter your First nonlinear dynamical complexity measures (142 - 367)250
Enter your second nonlinear dynamical complexity measures (25 - 68)29
Enter your third nonlinear measures of fundamental frequency variation (4 - 52)29
Enter your nonlinear fundamental frequency variation (0 - 45)29
Enter your Signal fractal scaling exponent (57 - 82)65
Enter your ratio of noise to tonal components in the voice (844 - 3304)2023
Enter your Several measures of variation in amplitude(0 - 5)0
Enter your Several measures of variation in fundamental frequency (0 - 3)2
C:\Users\kshah\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but DecisionTree
Classifier was fitted with feature names
  warnings.warn(
[0]
The patient does not have Parkinson
PS C:\Users\kshah\OneDrive\Desktop\Machine Learning> & C:/Users/kshah/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/kshah/OneDrive/Desktop/Machin
e Learning/parkinson.py"
```

As per the input, our algorithm predicted that the patient does not have Parkinson disease.

Bar Diagram:



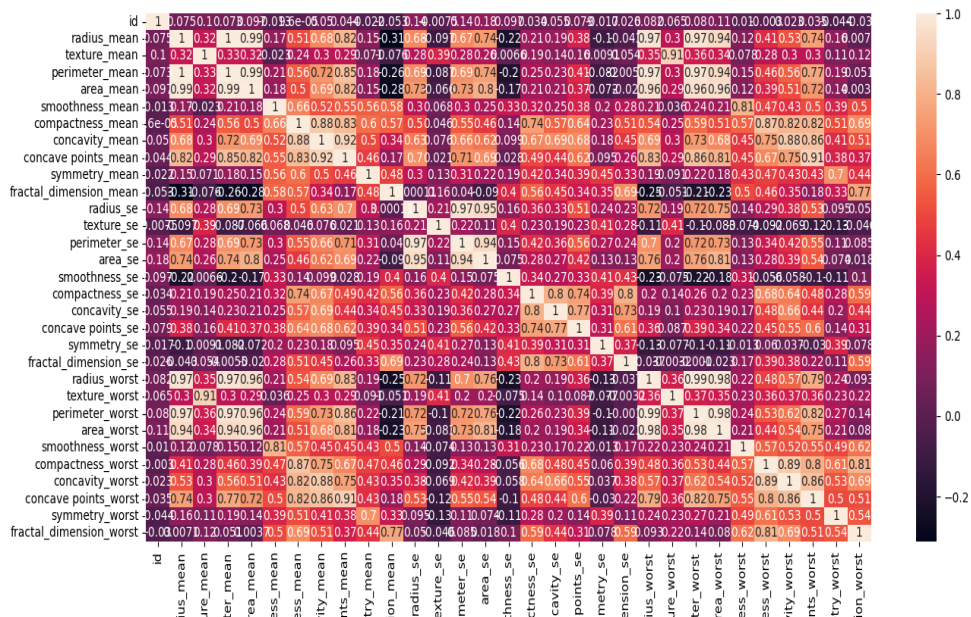
This Bar diagram define number of Output in each class.

BREAST CANCER:

Data:

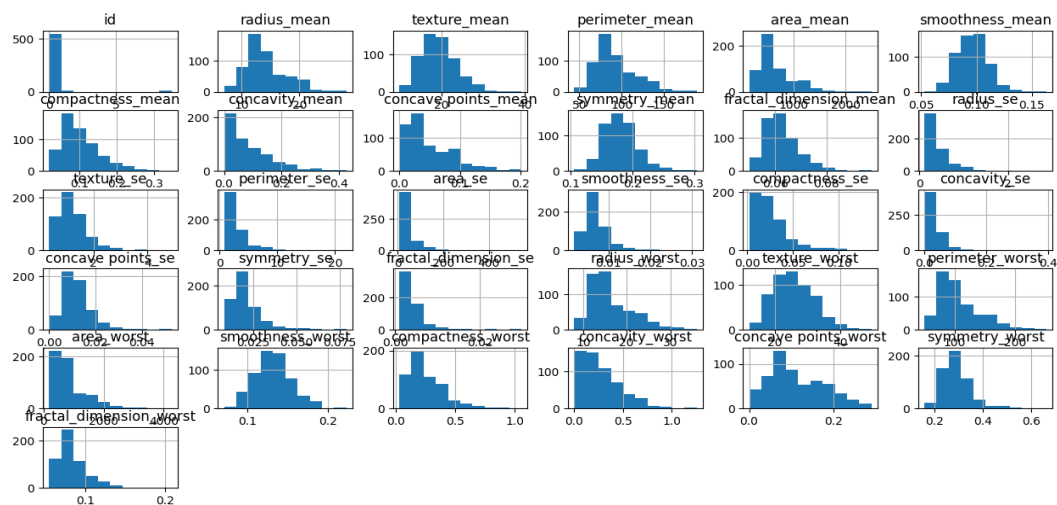
<bound method DataFrame.value_counts of				id diagnosis radius_mean text		
ure_mean	perimeter_mean	area_mean	\			
0	842302	M	17.99	10.38	122.80	1001.0
1	842517	M	20.57	17.77	132.90	1326.0
2	84300903	M	19.69	21.25	130.00	1203.0
3	84348301	M	11.42	20.38	77.58	386.1
4	84358402	M	20.29	14.34	135.10	1297.0
..
564	926424	M	21.56	22.39	142.00	1479.0
565	926682	M	20.13	28.25	131.20	1261.0
566	926954	M	16.60	28.08	108.30	858.1
567	927241	M	20.60	29.33	140.10	1265.0
568	92751	B	7.76	24.54	47.92	181.0

Heatmap:



This is the Heatmap before Feature Engineering.

Histogram:



This is the Histogram before Feature Engineering.

Accuracy Score:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

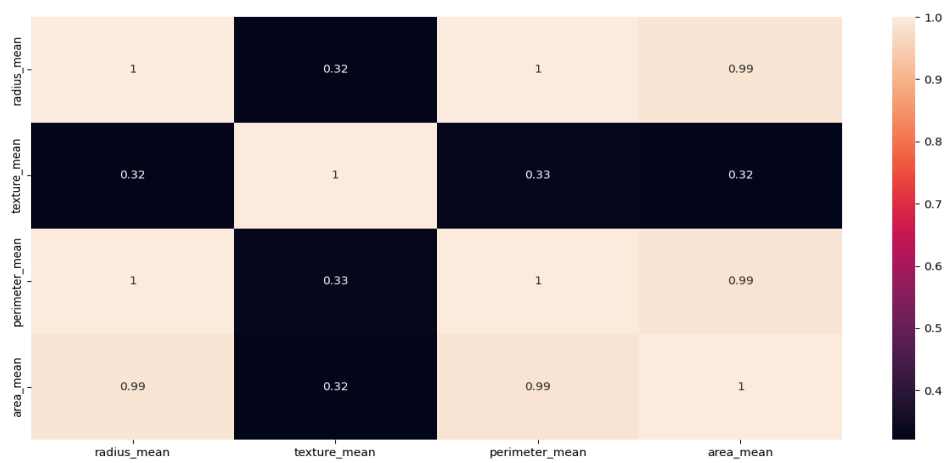
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kshah\OneDrive\Desktop\Machine Learning> & C:\Users\kshah\AppData\Local\Programs\Python\Python311\python.exe "c:/Users/kshah/OneDrive/Desktop/Machine Learning/br.py"
Accuracy (Training Data) : 91.20879120879121 %
Accuracy (Testing Data) : 92.98245614035088 %
PS C:\Users\kshah\OneDrive\Desktop\Machine Learning>

```

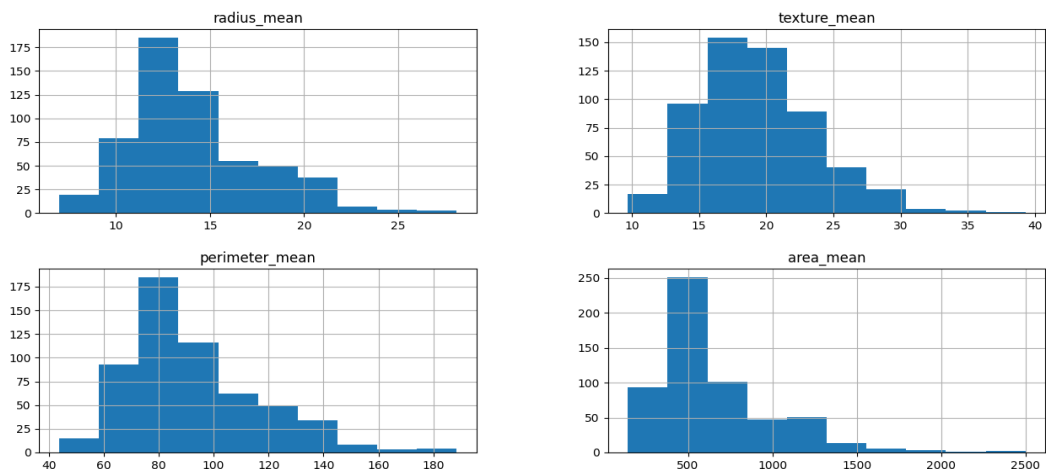
As you can see, we are using Support Vector Machines the accuracy score in Training Data is 100% and in Testing Data is 92%.

Heatmap:



This is the Heatmap after Feature Engineering.

Histogram:



This is the after before Feature Engineering.

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kshah\OneDrive\Desktop\Machine Learning> & C:/Users/kshah/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/kshah/OneDrive/Desktop/Machine Learning/br.py"
Please enter radius_mean Range(6.981 - 28.11): 12.47
Please enter texture_mean (9.71 - 39.28): 18.60
Please enter perimeter_mean (43.79 - 188.5): 81.09
Please enter area_mean(143.5 - 2501): 481.9
C:\Users\kshah\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
Benign
PS C:\Users\kshah\OneDrive\Desktop\Machine Learning>
```

As per the input, our algorithm predicted that the patient has Benign type Breast Cancer.

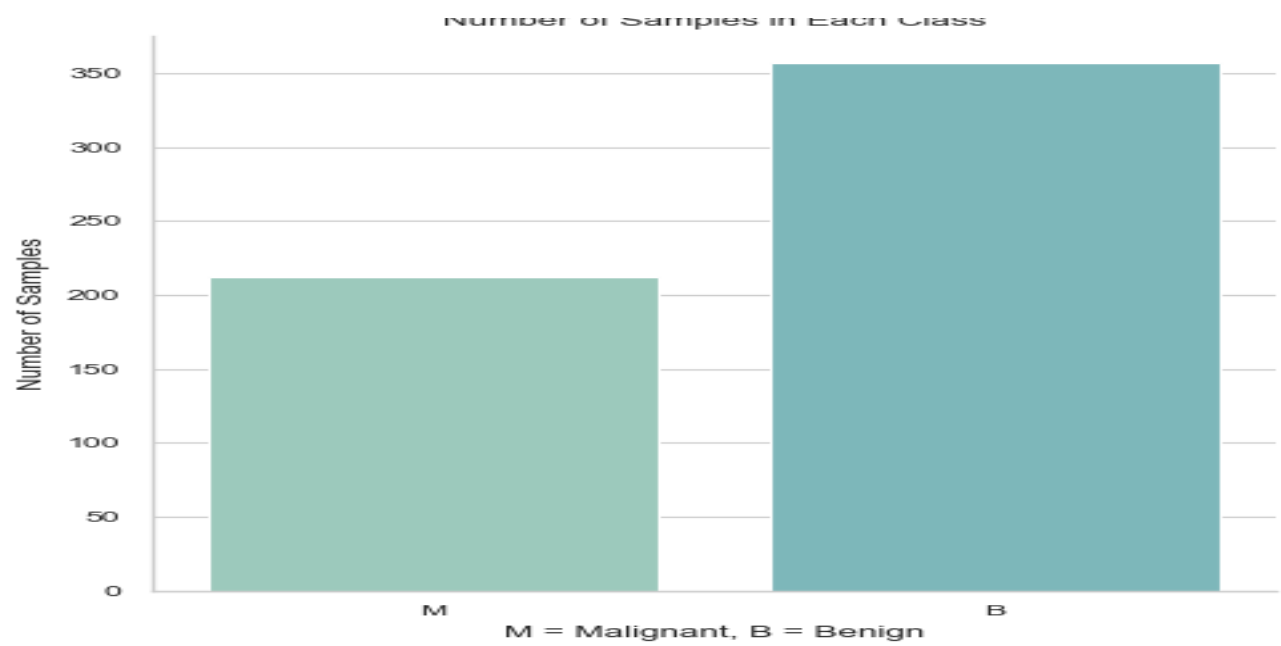
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\kshah\OneDrive\Desktop\Machine Learning> & C:/Users/kshah/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/kshah/OneDrive/Desktop/Machine Learning/br.py"
Please enter radius_mean Range(6.981 - 28.11): 18.94
Please enter texture_mean (9.71 - 39.28): 21.31
Please enter perimeter_mean (43.79 - 188.5): 123.60
Please enter area_mean(143.5 - 2501): 1130.0
C:\Users\kshah\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  warnings.warn(
Malignant
PS C:\Users\kshah\OneDrive\Desktop\Machine Learning>
```

As per the input, our algorithm predicted that the patient has Malignant Breast Cancer.

Bar Diagram:



This Bar diagram defines the number and type of Outputs.

PROPOSED METHODOLOGIES

BREAST CANCER:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore", category=UserWarning)

def breast_cancer():

    df =
pd.read_csv('C:/Users/kshah/OneDrive/Desktop/test_major_project/disease_pred/b
r.csv')

    x_train, x_test, y_train, y_test =
train_test_split(df.drop(columns=['diagnosis']), df['diagnosis'],
test_size=0.2, random_state=42)
    clf = svm.SVC(kernel='linear')
    clf.fit(x_train, y_train)

    # Accuracy Score on training data
    x_train_pred = clf.predict(x_train)
    training_data_accuracy = accuracy_score(y_train, x_train_pred)
    print('Accuracy (Training Data) :', training_data_accuracy*100,'%')

    # Accuracy Score on test data
    x_test_pred = clf.predict(x_test)
    testing_data_accuracy = accuracy_score(y_test, x_test_pred)
    print('Accuracy (Testing Data) :', testing_data_accuracy*100,'%')

print("Please enter radius_mean Range(6.981 - 28.11):
")
    sp.speak("Please enter radius_mean Range(6.981 - 28.11): ")
    radius_mean = cmd.takeCommand().lower()
    print(radius_mean)
    sp.speak(radius_mean)
```

```

print("Please enter texture_mean (9.71 - 39.28): ")
sp.speak("Please enter texture_mean (9.71 - 39.28): ")
texture_mean = cmd.takeCommand().lower()
print(texture_mean)
sp.speak(texture_mean)
# Splitting the data into testing and training set
x_train, x_test, y_train, y_test =
train_test_split(df.drop(columns=['status']), df['status'], test_size=0.2,
random_state=42)

# Data Standardization
scaler = StandardScaler()
a = scaler.fit(x_train)

x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)

# Model Training (DecisionTreeClassifier)
clf = DecisionTreeClassifier()
clf.fit(x_train, y_train)

print("Please enter perimeter_mean (43.79 - 188.5): ")
sp.speak("Please enter perimeter_mean (43.79 - 188.5): ")
perimeter_mean = cmd.takeCommand().lower()
print(perimeter_mean)
sp.speak(perimeter_mean)

print("Please enter area_mean(143.5 - 2501): ")
sp.speak("Please enter area_mean(143.5 - 2501): ")
area_mean = cmd.takeCommand().lower()
print(area_mean)
sp.speak(area_mean)
preds = clf.predict([[radius_mean, texture_mean, perimeter_mean,
area_mean]])
f_pred = (' '.join(preds))

if f_pred == 'B':
    f_pred = 'Benign'
else:
    f_pred = 'Malignant'
sp.speak(f_pred)
return f_pred

```

PARKINSON DISEASE:

```
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore", category=UserWarning)

def parkinson():

    df =
pd.read_csv('C:/Users/kshah/OneDrive/Desktop/test_major_project/disease_pred/p
arkinsons.csv')
    # print(df.info())
    # print(df.describe())
    df.isnull().sum()#checking for missing values

    #dropping column axis = 1; dropping row then axis = 0
    #Data Pre-Processing - Separating Features and Target variables according
to their Correlation

    df.drop(["name", 'spread1', 'MDVP:Flo(Hz)', 'MDVP:Fhi(Hz)', 'MDVP:Fo(Hz)'],
axis=1, inplace=True)
    columns = list(df.columns)
    for column in columns:
        if column == "status":
            continue

    filtered_columns = [column]
    for col in df.columns:
        if (column == col) | (column == "status"):
            continue
        cor_val = df[column].corr(df[col])
        if cor_val > 0.75:
            columns.remove(col)
```



```

        continue
    else:
        filtered_columns.append(col)
df = df[filtered_columns]

df.isnull().sum() #checking null value

# converting Data in the form of hundred
df.iloc[:,8] = (df.iloc[:, :8]).mul(100).astype(int)

# Splitting the data into testing and training set
x_train, x_test, y_train, y_test =
train_test_split(df.drop(columns=['status']), df['status'], test_size=0.2,
random_state=42)

# Model Training (DecisionTreeClassifier)
clf = DecisionTreeClassifier()
clf.fit(x_train, y_train)
# Model Evaluation
# Accuracy Score

# Accuracy Score on training data
x_train_pred = clf.predict(x_train)
training_data_accuracy = accuracy_score(y_train, x_train_pred)

print('Accuracy (Training Data) :', training_data_accuracy*100,'%')

# Accuracy Score on test data
x_test_pred = clf.predict(x_test)
testing_data_accuracy = accuracy_score(y_test, x_test_pred)

print('Accuracy (Testing Data) :', testing_data_accuracy*100,'%')
print("Enter your First nonlinear dynamical complexity measures (142 - 367)")
sp.speak("Enter your First nonlinear dynamical complexity measures (142 -
367)")
D2 = cmd.takeCommand().lower()
print(D2)
sp.speak(D2)

print("Enter your second nonlinear dynamical complexity measures (25 -
68)")
sp.speak("Enter your second nonlinear dynamical complexity measures (25 -
68)")
RPDE = cmd.takeCommand().lower()
print(RPDE)

```

```

sp.speak(RPDE)

print('Enter your third nonlinear measures of fundamental frequency
variation (4 - 52)')
sp.speak('Enter your third nonlinear measures of fundamental frequency
variation (4 - 52)')
PPE = cmd.takeCommand().lower()
print(PPE)
sp.speak(PPE)

print("Enter your nonlinear fundamental frequency variation (0 - 45)")
sp.speak("Enter your nonlinear fundamental frequency variation (0 - 45)")
spread2 = cmd.takeCommand().lower()
print(spread2)
sp.speak(spread2)

print("Enter your Signal fractal scaling exponent (57 - 82)")
sp.speak("Enter your Signal fractal scaling exponent (57 - 82)")
DFA = cmd.takeCommand().lower()
print(DFA)
sp.speak(DFA)

print("Enter your ratio of noise to tonal components in the voice (844 -
3304)")
sp.speak("Enter your ratio of noise to tonal components in the voice (844
- 3304)")
HNR = cmd.takeCommand().lower()
print(HNR)
sp.speak(HNR)

print("Enter your Several measures of variation in amplitude(0 - 5)")
sp.speak("Enter your Several measures of variation in amplitude(0 - 5)")
# Shimar = cmd.takeCommand().lower()
# print(Shimar)
# sp.speak(Shimar)
Shimar = input('Enter variation in amplitude: ')

print("Enter your Several measures of variation in fundamental frequency
(0 - 3)")
sp.speak("Enter your Several measures of variation in fundamental
frequency (0 - 3)")
# Jitter = cmd.takeCommand().lower()
# print(Jitter)
# sp.speak(Jitter)
Jitter= input('Enter fundamental frequency: ')

```

```

p_pred = clf.predict([[D2, RPDE, PPE, spread2, DFA, HNR,Shimar, Jitter]])

predicted = ""

if p_pred == 0:
    predicted = 'Not Affected'

else:
    p_pred == 1
    predicted = 'Affected'

    return predicted

```

DATABASE:

```

import mysql.connector

mysql = mysql.connector.connect(host = "XXXXX", user = "XXXXX", passwd =
"XXXXX")
mycursor = mysql.cursor()

def insert_patient(patient_id, name, sex, age, address, contact, mail,
disease, f_pred, p_pred):

    mycursor.execute("create database if not exists city_hospitals")
    mycursor.execute("use city_hospitals")

    # creating the tables for storing patient details.
    mycursor.execute("create table if not exists patient_detail(patient_id
int(4) primary key, name varchar(30) ,sex varchar(15),age int(3),address
varchar(50),contact varchar(15),mail varchar(40), disease varchar(80),
breasr_cancer_prediction varchar(20), parkinson_disease_prediction
varchar(20))")

    # Inserting Patient Details
    mycursor.execute("insert into patient_detail values('" + patient_id +
"', '" + name + "', '" + sex + "', '" + age + "', '" + address + "', '" + contact +
"', '" + mail + "', '" + disease + "', '" + f_pred + "', '" + p_pred + "')")
    mysql.commit()

```

ACCURACY TABLE

Table of accuracy			
STUDY DATE	ACCURACY	SCORE	SOURCE
May 2019	Train	83.44%	https://www.kaggle.com/code/parhamzm/parkinson-s-disease-pd-classification/notebook#notebook-container
	Test	85.53%	
Feb 2023	Train	84.65%	https://www.kaggle.com/code/akankshal0/detection-of-parkinson-s-disease
	Test	86.8%	
Jan 2021	Train	93.57%	https://www.kaggle.com/code/vikasukani/detecting-parkinson-s-disease-machine-learning
	Test	96.66%	
July 2022	Train	88.46%	https://www.youtube.com/watch?v=ys_mVbkaokE
	Test	87.17%	
June 2016	Train	74.60%	https://www.kaggle.com/code/lykin22/parkinson-s-disease-based-on-voice-recording#Parkinson's-disease-based-on-voice-recording
	Test	74.55%	
Feb 2015	Train	95.38%	https://www.researchgate.net/profile/Anil-Kumar-544/google-scholar
	Test	94.72%	
Feb 2018	Train	95.38%	https://www.researchgate.net/profile/Anil-Kumar-544/google-scholar
	Test	94.72%	
Aug 2017	Train	97%	https://www.youtube.com/watch?v=eKy3KgRgDkQ
	Test	98%	
Sep 2020	Train	88.46%	https://www.youtube.com/watch?v=CQLkX4utdIU
	Test	87.11%	
April, 2022	Train	88.46%	https://github.com/akashdeep364/Parkinson-s-Disease-Detection
	Test	87.17%	

APPLICATIONS

This project is based on a trending technology of the present times and has many applications –

1. One of the most important applications of this project is that, the accessibility to the Hospital Staff and Patient records are available in Database can easily accessible making it more user friendly helping the hospital administration to manage data even during rush hours smoothly.
2. This project paves the path for a smooth guidance to all the health-conscious individuals specially differently abled people irrespective of their age and health conditions.
3. The users are also given relief from the hassle of storage issues when it comes to using this application as it provides the opportunity of online data storage.
4. Data can be inserted, updated when required, deleted, and can also be saved separately in Database tables uncomplicated distinguish between information of users.
5. Fast and Early Prediction of Life taking Diseases Like Breast Cancer and Parkinson Diseases.
6. Automated Mail Sending Feature makes it more reliable and time saving.

STRENGTHS

- This Project is completely based on Human Voice Command. Because of its Voice Controlled feature it is more beneficial for People with disability.
- Easy to Predict the onset of Parkinson disease and Breast Cancer.
- Uses Machine Learning Algorithm for the diseases Prediction and gives fast results and this can be used for Prediction at a rare critical situation like unavailability of a Neurologist and Oncologist.
- User can send the Reports to the Patients over mail. Reduces Paperwork & acts as a Document that can be accessed from any device at any time.
- Easy to access any Staff's or Patient's data from anywhere in the world via Authorised Login.

LIMITATIONS

- The high cost of software development and deployment.
- Complex Machine Learning Algorithm in terms of User Experience.
- Fear of data security breach.
- Difficulty in migrating from manual processes, because both staff and patients are used to the manual processes and so are unable to speedily cope with the new system.
- Lack of IT-friendly medical personnel is also presenting several challenges.
- Sometimes Predicated result may not gives 100% accuracy.
- Needs Internet connectivity to send the reports over mail.

CONCLUSION

With our proposed system, comparatively a good and higher accuracy is achieved. This is then used by researchers, physicians, hospitals, healthcare centres or doctors in order to provide the best treatment and medical care for the patients. Hence machine learning when used in healthcare can lead to an effective treatment and the patient is also well taken care of. Here we try to implement some of the functions of machine learning in healthcare into our system. In place of direct diagnosis, when a patient's disease is to be predicted then machine learning is implemented using certain algorithms. In this way, healthcare can be made much better and advanced. When we compare the different algorithms used for disease prediction from our dataset and the output we expect we get the best accuracy with Decision Tree Classifier and Support Vector Machine(SVM) whereas LDA algorithm had the lowest performance when compared to the other algorithms. Machine Learning (ML) gives us different methods and techniques that can make the issue of diagnostic problems easy and simple by modernizing different medical domains. Today, ML is used largely to predict, analyze clinical works and process data analysis like error detection in the dataset and for dealing with incorrect data present in our system. It is no doubt that implementing ML algorithm helped integrating computer system in the industry of healthcare to facilitate and enhance the work of doctors and finally leading to improve the efficiency level and quality of our medical care for the respective patients.

FUTURE DEVELOPMENT

In near Future, we are thinking to develop a Single Page User Management Website for controlling and managing all the Frontend flows which will enhance User Interface and User Experience. We will also deploy our Project on AWS cloud so that it can easily be accessible independent of Single User Machine. We will add more number of Diseases prediction algorithm to make it more useful. Can be developed as a self - disease prediction system so that early stages are recognized faster decreasing the immortality rate or health risks.

In today's world most of the data is computerized, the data is distributed, and it is not utilizing properly. With the help of the already present data and analysing it, we can also use for un-known patterns. The primary motive of this project is the prediction of diseases with high rate of accuracy. For predicting the disease, we can use logistic regression algorithm, naive Bayes, sklearn in machine learning. The future scope of the paper is the prediction of diseases by using advanced techniques and algorithms in less time complexity. A technology called CAD is more beneficial as sometimes systems are better diagnostic than Doctors. Machine Learning and its different branches are used in Cancer detection as well. It helps or can say assist in making decisions on critical cases or on therapies. Artificial intelligence plays an important role in development of many health related procedure or methods. Artificial intelligence is very common now a days in surgeries, like Robotics surgery. Since were in the circumstances of growing population, we must need technology which can help us to meet the expectations of the patients, their flawless cure, their better health and their smooth and easy approachable access to healthcare industries to heal and get well soon!

REFERENCES

The Project Synopsis has been prepared by taking help from –

- A.Davis, D., V.Chawla, N., Blumm, N., Christakis, N., & Barbasi, A. L. (2008). Predicting Individual Disease Risk Based On Medical History.
- Adam, S., & Parveen, A. (2012). Prediction System For Heart Disease Using Naive Bayes. Al-Aidaros, K., Bakar, A., & Othman, Z. (2012). Medical Data Classification With Naive Bayes Approach. Information Technology Journal .
- Darcy A. Davis, N. V.-L. (2008). Predicting Individual Disease Risk Based On Medical History.
- JyotiSoni, Ansari, U., Sharma, D., & Soni, S. (2011). Predictive Data Mining for Medical Diagnosis: An Overview Of Heart Disease Prediction.
- K.M. Al-Aidaros, A. B. (n.d.).
- K.M. Al-Aidaros, A. B. (n.d.). 2012. Medical Data Classification With Naive Bayes Approach .
- NisharBanu, MA; Gomathy, B;. (2013). Disease Predicting System Using Data Mining Techniques.
- Youtube Link:
https://www.youtube.com/playlist?list=PLKnIA16_Rmvbr7zKYQuBfsV_kjoLcJgxHH
- Wikipedia.
- Our Project Guide

INDIVIDUAL CONTRIBUTION TO THE PROJECT

NAME OF THE PROJECT MEMBER	CONTRIBUTION
MUJAHID ALI ANSARI	SQL, MACHINE LEARNING
ASHUTOSH KUMAR YADAV	DOCUMENTATION
SHAHIL KUMAR CHOURASIA	PYTHON, MACHINE LEARNING
SONU ROUTH	TESTING, DBMS
SIRSHA MAJUMDER	PYTHON, DOCUMENTATION

THANK YOU