Detecting Parkinson’s Disease Using Machine Learning

Project Report

1.Introduction

1.1 Project Overview:

Parkinson’s disease (PD) is a neurodegenerative movement disease where the symptoms gradually develop to start with a slight tremor in one hand and a feeling of stiffness in the body and it becomes worse over time.
At present there is no conclusive result for this disease by non-specialist clinicians, particularly in the early stage of the disease where identification of the symptoms is very difficult. The disease is majorly said to be affecting the individuals who are living in village areas with their respective ages between 40 and 50.
Parkinson's disease detection system has been designed to detect the Parkinson's disease in a patient given their hand drawn spiral or wave images. The system is built using fundamental concepts of Data anlytics and Computer Vision that are trained to differentiate between healthy and Parkinson handdrawn images.

1.2 Purpose:

By using machine learning techniques, the problem can be solved with
minimal error rate. The voice dataset of Parkinson's disease from the UCI Machine learning library is used as input. Also our proposed system provides accurate results by integrating spiral drawing inputs of normal and Parkinson's affected patients. Machine learning also allows for combining different modalities, such as magnetic resonance imaging (MRI) and single-photon emission computed tomography (SPECT) data. in the diagnosis of PD. By using machine learning approaches, we may therefore identify relevant features that are not traditionally used in the clinical diagnosis of PD and rely on these alternative measures to detect PD in preclinical stages or atypical forms. In recent years, the number of publications on the application of machine learning to the diagnosis of PD has increased. Although previous studies have reviewed the use of machine learning in the diagnosis and assessment of PD, they were limited to the analysis of motor symptoms, kinematics, and wearable sensor data. Moreover, some of these reviews only included studies published between 2015 and 2016. In this study, we aim to comprehensively summarize all published studies that applied machine learning models to the diagnosis of PD for an exhaustive overview of data sources, data types, machine learning models, and associated outcomes, assess and compare the feasibility and efficiency of different machine learning methods in the diagnosis of PD, and provide machine learning practitioners interested in the diagnosis of PD with an overview of previously used models and data modalities and the associated outcomes, and recommendations on how experimental protocols and results could be reported to facilitate reproduction. As a result, the application of machine learning to clinical and non-clinical data of different modalities has often led to high diagnostic accuracies in human participants, therefore may encourage the adaptation of machine learning algorithms and novel biomarkers in clinical settings to assist more accurate and informed decision making.While Parkinson’s cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life.

2.Literature Survey:

2.1 Existing problem:

Due to insufficient resources and awareness in underdeveloped countries, proper and timely PD detection is highly challenged. Besides, all PD patients' symptoms are neither the same nor they all become pronounced at the same stage of the illness. Therefore, this work aims to combine more than one symptom by collecting data and detecting PD with the help of a cloud-based machine learning system for monitoring the PD patients in the developing countries.

2.2 References:

1. Jie Mei, Christian Desrosiers, Johannes Frasnelli, "Machine Learning for the Diagnosis of Parkinson's Disease," 2021.
2. C K Gomathy, "The Parkinson's Disease Detection using Machine Learning Techniques." 2021.
3. Iqra Nissar, Waseem Ahmad Mir, Izharuddin, Tawseef Ayoub Shaikh, "Machine Learning Approaches for Detection and Diagnosis of Parkinson's Disease," 2021
4. C K Gomathy, B.Varshini, B.Varsha, B.Dheeraj Kumar Reddy, "The Parkinson's Disease Detection using Machine Learning Techniques.", 2021
5. Mari Muthu Mari Muthu, "Detection of Parkinson's disease using Machine Learning Approach," 2021.

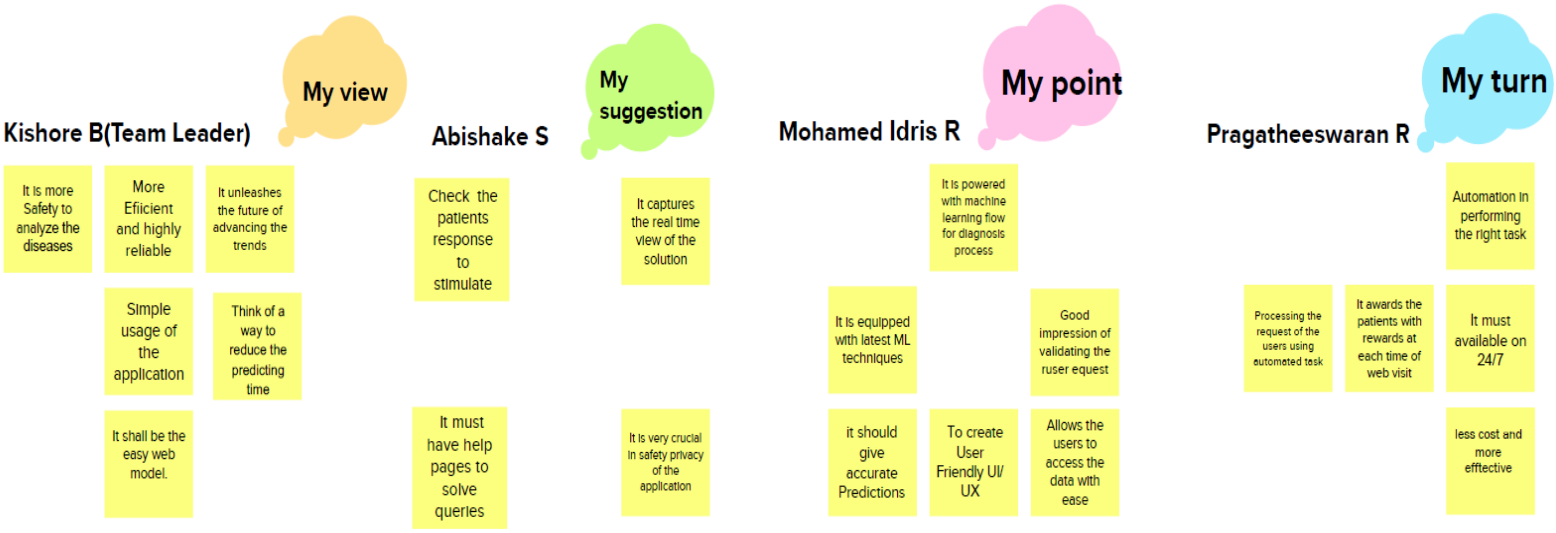
2.3 Problem Statement Definition:

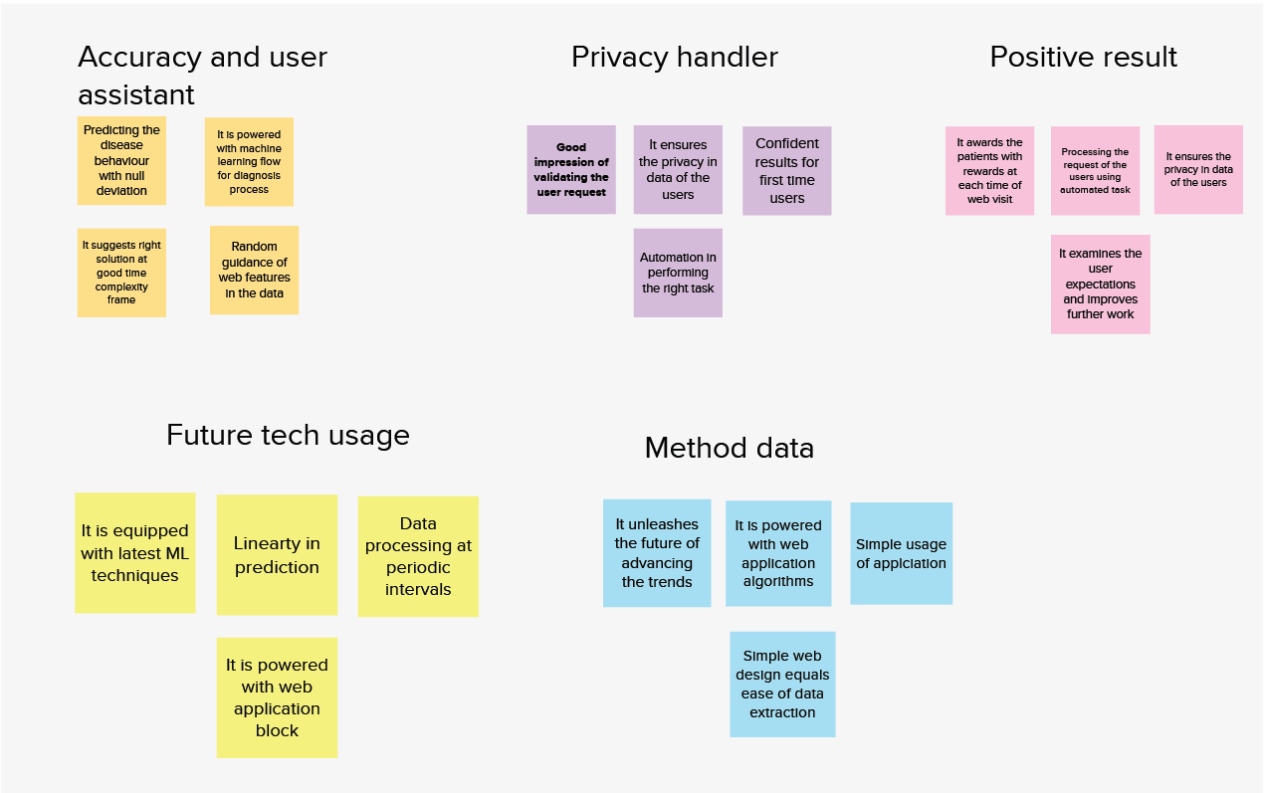
By processing the handdrawn spiral and wave images of the patients we can create a model to learn the difference between healthy and Parkinson affected drawing patterns. The patients provides their handdrawn image and the our machine learrning model predicts whether the patient is affected by Parkinson's disease.

3.Ideation and Proposed Solution:

3.1 Empathy map canvas:

3.2 Ideation and Brainstorming:

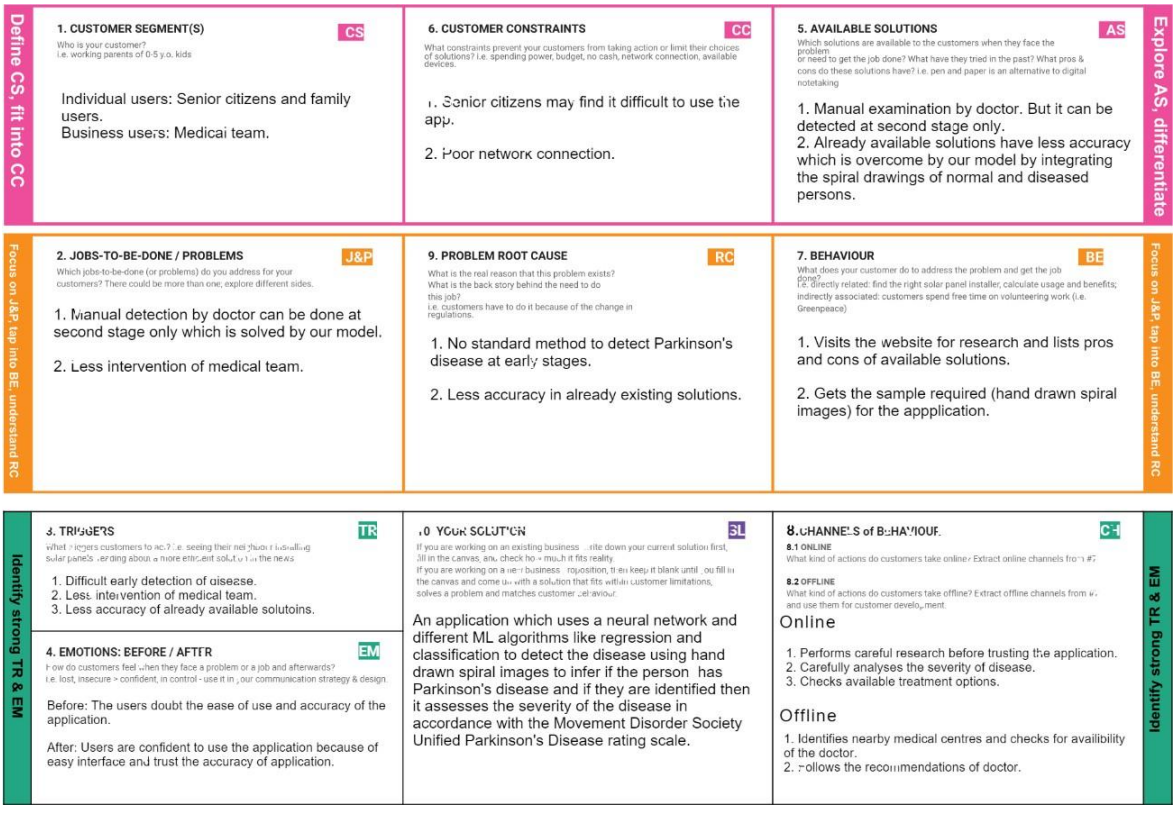




3.3 Proposed Solution:

Idea / Solution description -	User can place their values and interact with the friendly user assistance bot which guides the person in using the application
Novelty / Uniqueness -	Parkinson's Disease is detected at the secondary stage only (Dopamine deficiency) which leads to medical challenges. Also, doctor must manually examine and suggest medical diagnosis in which the symptoms might vary from person to person so suggesting medicine is also a challenge. So hence the disease examination varies at different instances of the medical operations. Here by using machine learning methods, the problem can be addressed with very less error rate.
Business Model (Revenue Model) -	1.Easy interface 2.Economical Development 3. Suits for better saving of involvements

3.4 Problem Solution fit:



4. Requirement Analysis:

4.1 Functional Requirements:

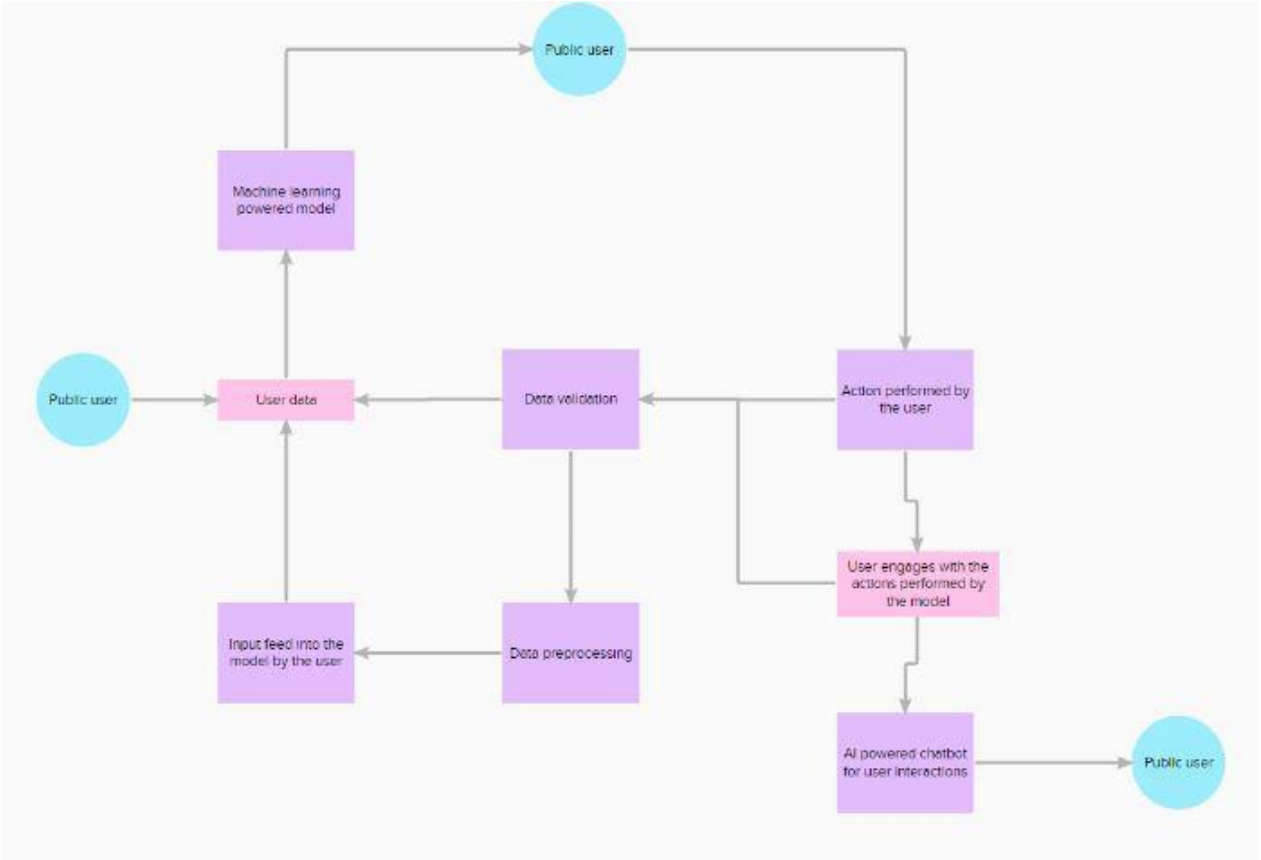
FR No.	Functional Requirements (Epic)	Description
FR-1	Accessing the dashboard	Through the Link Through Google Search Engine
FR-2	Input Data	Input hand drawn Spiral images Input hand drawn Wave images
FR-3	Check Prediction	Display the prediction as diseased or not diseased
FR-4	Read Medical Suggestion	Display medical suggestions
FR-5	Checking the proper working of system	Go through the complete workflow of the application

4.2 Non-functional Requirements:

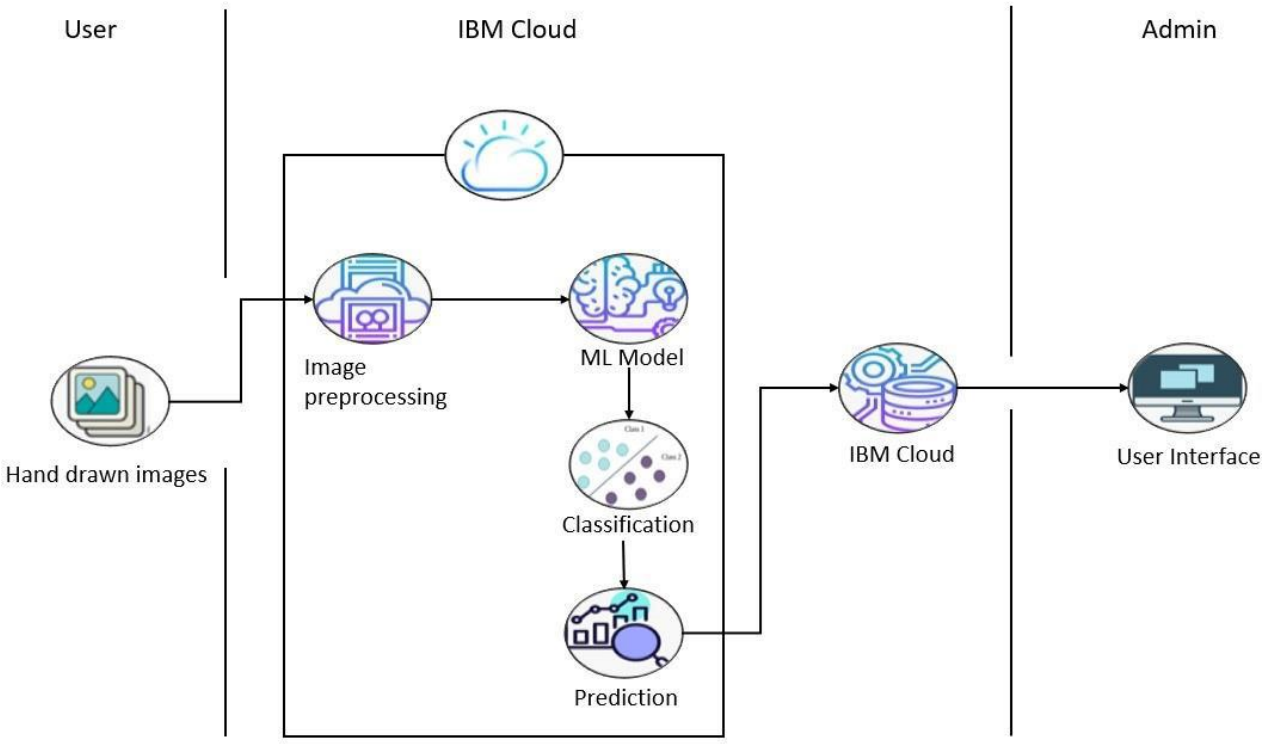
NFR No.	Non-Functional Requirements	Description
NFR-1	Usability	Usable by multiple users at the same time
NFR-2	Security	Encrypts user input data
NFR-3	Reliability	Model accuracy is high
NFR-4	Performance	Response time of predicting is low
NFR-5	Availability	Available all the time
NFR-6	Scalability	Works well under multiple requests

5. Project Design:

5.1 Data flow diagrams:



5.2 Solution and technical architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Public user)	Account creation	USN-1	As a user, I can connect my google into the application	I can access my account / application dashboard	High	Sprint-1
Input data	Adding data	USN-2	As a user, I can feed my data as the input into the application for it to classify the true fake data	I can cross verify the data that entered in the initial step	High	Sprint-1
Data validation	Checking accuracy	USN-3	As a user, I can check the ability and accuracy of the model in obtaining the required information	I can log into my account and check the capability of the model	Medium	Sprint-2
Classification	Data classification	USN-4	As a user, I can view the real data	I can verify my data with the real data	Medium	Sprint-2
App work	Work flow	USN-5	As a user, I can examine the working action of the application model	I can view how the application works and responds to the actions imposed	High	Sprint-2
Image classification	Checking for the disease	USN-6	As a user, I can verify with the application that the image is identified with the actual disease with the help of the trained and tested data's	I can confirm that the data shows the accurate result	Low	Sprint-3
User interaction	AI-powered chatbot	USN-7	As a user, I can interact with the automated chatbot to engage my time till the application processed the accurate result in a meanwhile	I can see the results from the interaction with the chatbot	Low	Sprint-3
Medical assistance	Medical suggestions	USN-8	As a user, I can get medical advises and recommendations for to boost the action of curing the disease	I can get enough assistance by getting the suggestions for curing the disease	High	Sprint-3
Data extraction	Obtaining the data	USN-9	As a user, I can retrieve the result data from the application for data storage for further medical research uses.	I can download the result in the form of data as a proof to show to medical teams	Medium	Sprint-4

6. Project Planning and Scheduling:

6.1 Sprint Planning and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Data Collection (Dataset)	USN-6	I need to collect data (images of spirals and waves drawn by healthy people and Parkinson's patients).	6	Medium	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-2	Data checking	USN-7	I need to learn and understand the data	2	Medium	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-3	Data Pre-Processing and EDA	USN-8	I need to prepare, clean the data, and process the data for modelbuilding by doing pre-processing activities such as EDA and data visualization.	4	High	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-3	Data visualization	USN-9	I need to visualize the data for to check for any outliers and processing the data accordingly.	7	Medium	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-3	Model Building (Training and testing)	USN-10	I need to build the model using Data mining processes such as Random ForestClassifier, K Nearest Neighbor (KNN) from regression, classification, and clustering techniques.	4	High	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-3	Assessing the model using metrics	USN-11	I need to measure the performance of the model using regression metrics	5	Medium	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-4	Application Building	USN-12	I need to build the website for the model application using HTML, CSS, JavaScript etc followed by user sign up page creation in sprint 1. It is then completed by designing the application website.	4	Medium	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-4	Model verification	USN-13	I need to check that model works fine in the application for the user.	6	High	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-4	Model Deployment (IBM Cloud)	USN-14	I need to deploy the Machine Learning model iiithat was built using cloud environment from IBM. And configuring the data of the user in IBM warehouse service called as db2.	5	Medium	Kishore Mohamed Idris Abishake Pragatheeswaran
Sprint-4	Results	USN-15	As a user, I can receive a diagnosis in addition to recommendations on what I should do now.	5	High	Kishore Mohamed Idris Abishake Pragatheeswaran

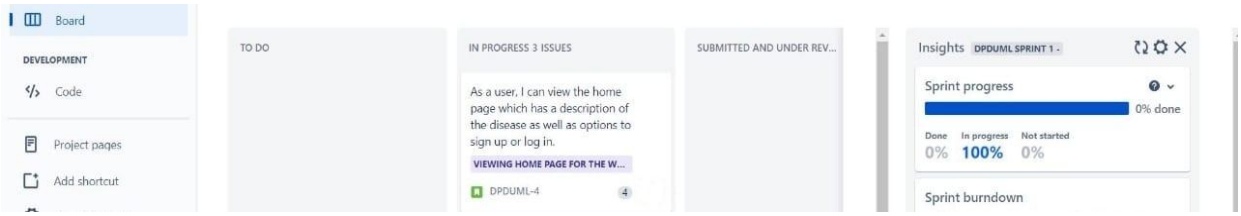
6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

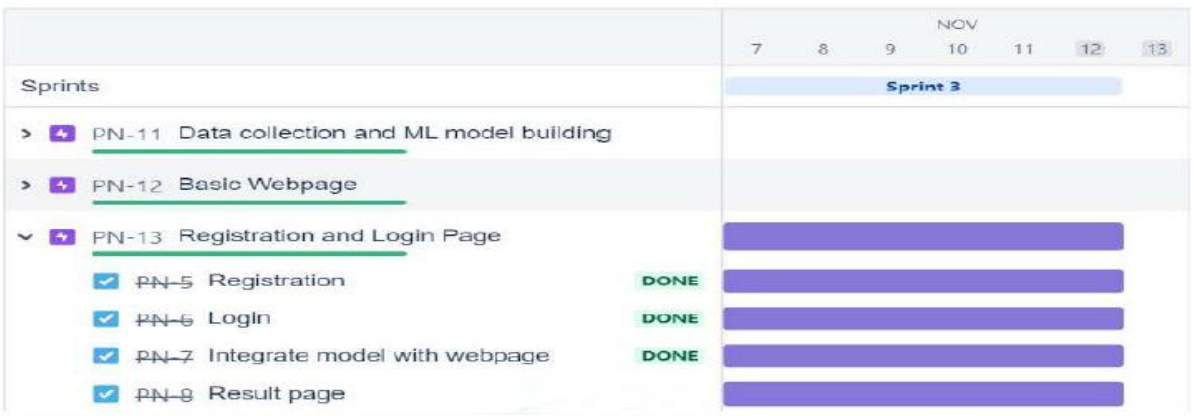
6.3 Reports from JIRA:

Sprint-1:

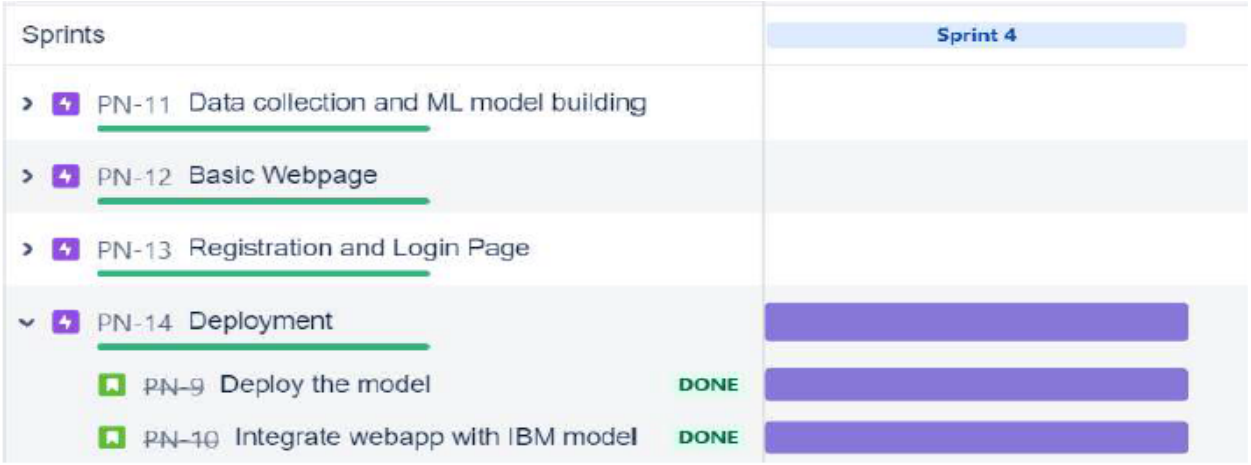


Sprint-2:

Sprint-3:



Sprint-4:



7. Coding and Solutioning:

Index:

The index. html page is the most common name used for the default page shown on a website if no other page is specified when a visitor requests the site. In other words, index. html is the name used for the homepage of the website.

index.html

```
<!DOCTYPE html>

<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>
  <meta charset="UTF-8">
  <title>Parkinsons disease prediction</title>
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/main.css')
}}" />
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/index.css')
}}" />
</head>

<body >
 <div class="main">
   <div class="head">
    <h1>Parkinsons Disease Detection</h1>
   </div>

   <div class="para">
    <p>
      The Parkinson's disease is progressive neuro degenerative disorder that affects a
lot only people significantly affecting their quality of life
      It mostly affect the motor functions of human. The main motor symptoms are
```

called "parkinsonism" or "parkinsonian syndrome".
The symptoms of Parkinson's disease will occur slowly, the symptoms include shaking, rigidity, slowness of movement and difficulty with walking,
Thinking and behavior change, Depression and anxiety are also common
There is a model for detecting Parkinson's using voice. The deflections in the voice will confirm the symptoms of Parkinson's disease
</p>

Get Started

</body>
</html>

upload.html

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>Parkinsons disease prediction</title>
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/upload.css')
}}" />
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='css/main.css')
}}" />
  <script
  class="jsbin"
  src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"
  ></script>
  <script
  class="jsbin"
  src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.0/jquery-ui.min.js"
  ></script>
 </head>

 <body>
   <marquee behavior="" direction="" style="background-color:
#B0D0B3;"><h1>Parkinsons Disease Prediction</h1></marquee>
  <div class="main">
   <b><h3 style="text-align: center; font-size: 36px;" >UPLOAD IMAGE</h3></b>
   <div class="up-img">
    <img id="uploaded-img" src="#" alt="xray image" />
   </div>
   <form method="POST" action="/upload" enctype="multipart/form-data">
    <div>
     <input type="file" onchange="readURL(this);" id="my image" name="file"
class="file_input" required>
```

```
        <b></b><label for="user">Choose Type:</label></b>
        <select name="type" id="user" class="file_select" required>
          <option value="spiral">Spiral</option>
          <option value="wave">Wave</option>
        </select>
      </div>
      <br><br>
      <input type="submit" value="Submit" class="upload">
    </form>

  </div>
  <script>
    function readURL(input) {
      if (input.files && input.files[0]) {
        var reader = new FileReader();

        reader.onload = function (e) {
          $('#uploaded-img').attr('src', e.target.result);
        };

        reader.readAsDataURL(input.files[0]);
      }
    }
  </script>
 </body>
</html>
```

result.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Parkinson Disease</title>
    <link rel="stylesheet" type="text/css" href="{{
url_for('static',filename='css/result.css') }}" />
</head>

<body>
    <div class="main">
        <div class="img">
            <img src="{{path}}" alt="" width="256" height="256">
            <p>Diagnosed Result : {{prediction}}</p>
        </div>
        <a href="/upload"> Back </a>
    </div>
</body>
</html>
```

Prediction:

After the user can test if he/she has pakinson's disease by uploading hand drawn spiral or wave image. The model predicts whether the user has Parkinson's disease. If the user has Parkinson's disease the application offers the user medical suggestions and healthy diets.

predict.py

```
import pickle
import cv2
from skimage import feature

waveModel = pickle.load(open("ML/models/spiralModel.pkl", "rb"))
spiralModel = pickle.load(open("ML/models/spiralModel.pkl", "rb"))

code2cat = {0:"healthy", 1:"parkinson"}


def processImage(imagePath):
    image = cv2.imread(imagePath)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    image = cv2.resize(image, (200, 200))
    image = cv2.threshold (image, 0, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)
[1]
    return image


def quantify_image(image):
  features = feature.hog(image, orientations=9, pixels_per_cell=(13, 13),
cells_per_block=(4, 4), transform_sqrt=True, block_norm="L1")
  return features


def predict(imgPath, modelType):
    img = processImage(imgPath)
    imgFeatures = quantify_image(img)

    if modelType=="spiral":
      pre = spiralModel.predict([imgFeatures])

    if modelType=="wave":
      pre = spiralModel.predict([imgFeatures])

    return code2cat[pre[0]]
```

app.py

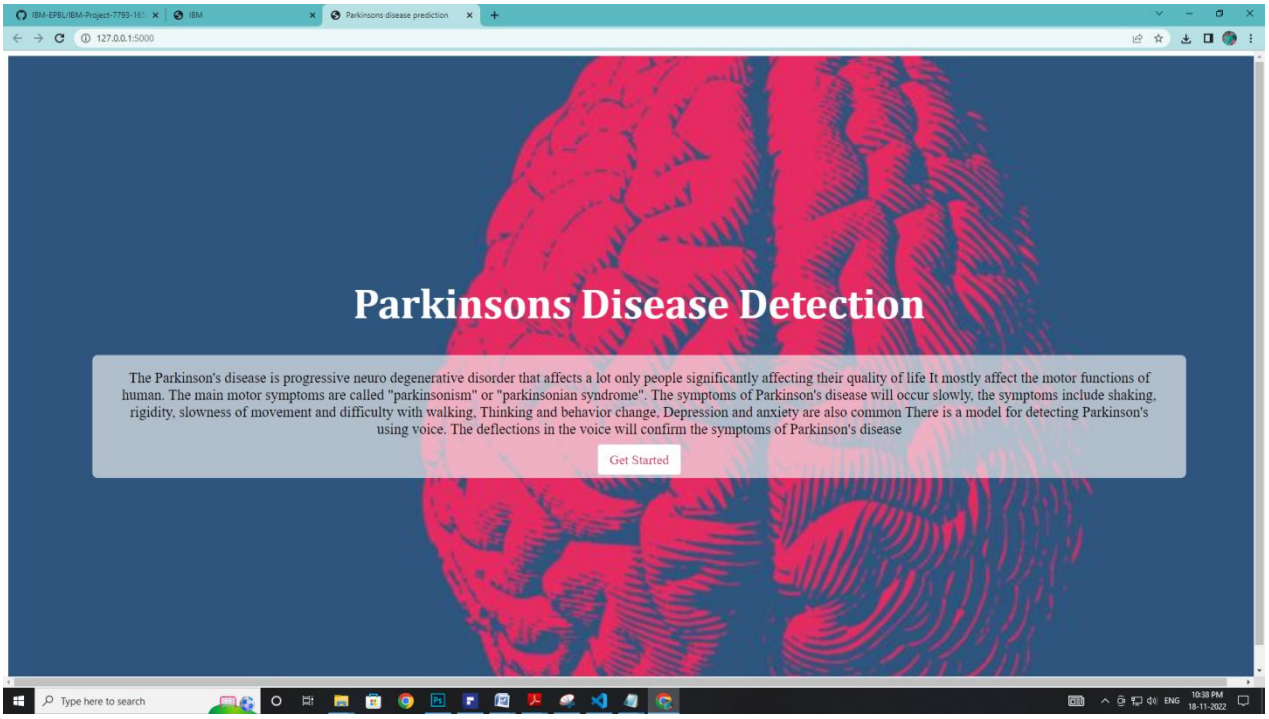
```
from flask import Flask, request, redirect, flash, send_from_directory, render_template,
send_file
```

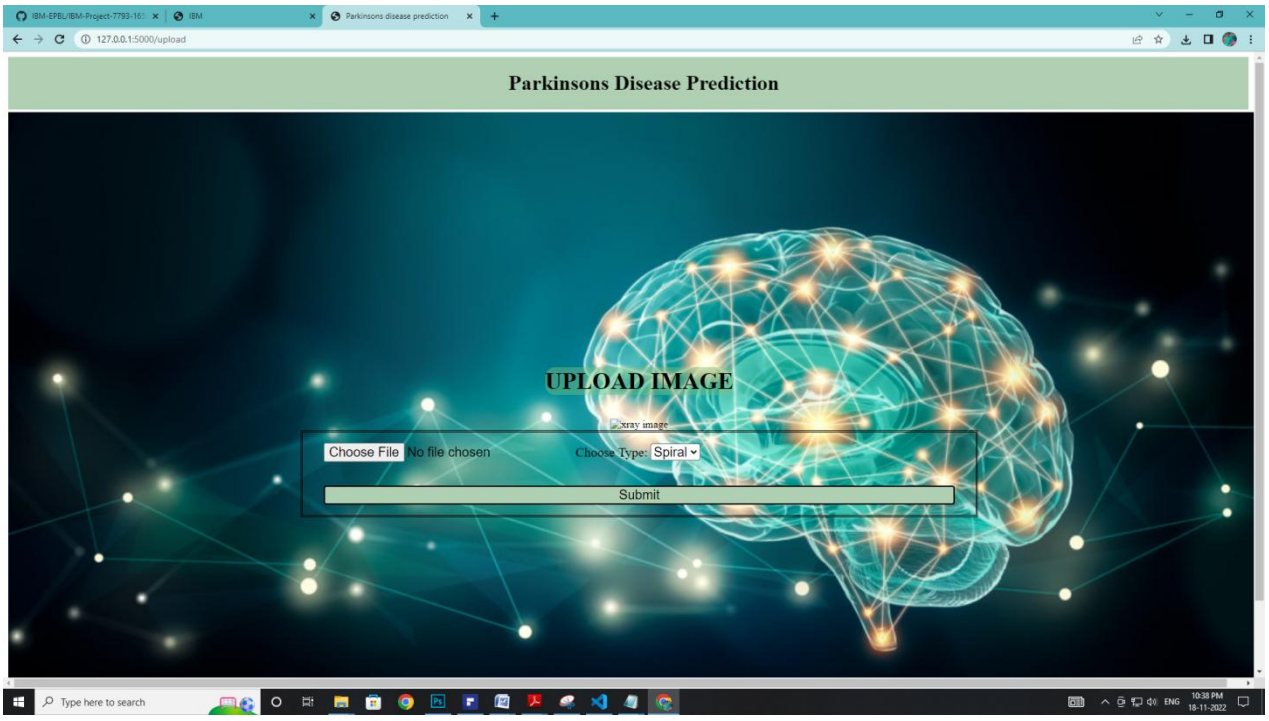
```
app = Flask(__name__, static_url_path='/')
app.config['SECRET_KEY'] = 'secret_key'
app.config['UPLOAD_FOLDER'] = 'static/uploads/'
```

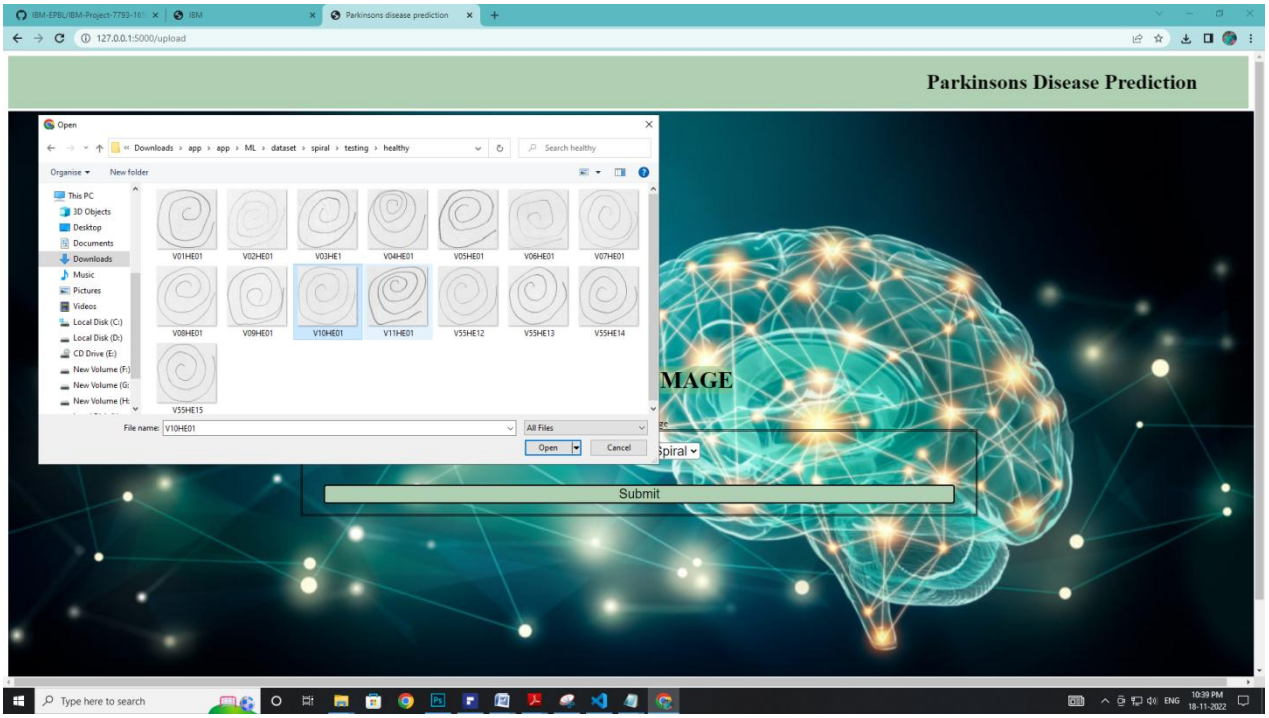
main.py (prediction part)

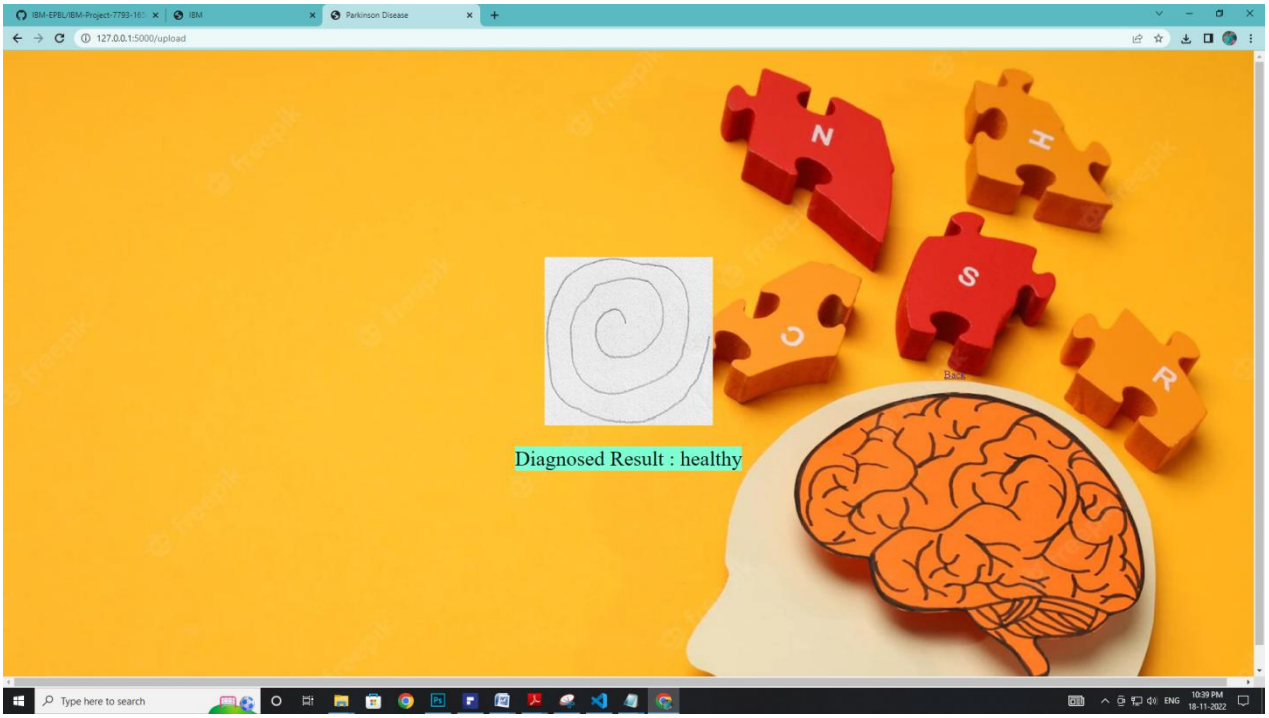
```
from app import *
import os
from utils.predict import predict

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



@app.route("/signin", methods=["POST", "GET"])
def signin():
    if request.method=="GET":
        return render_template('signin.html')


@app.route("/signup", methods=["POST", "GET"])
def signup():
    if request.method=="GET":
        return render_template('signup.html')


@app.route("/upload", methods=["POST", "GET"])
def upload():
    if request.method=="GET":
        return render_template('upload.html')
    if request.method=="POST":
        imgType = request.form["type"]
        file = request.files['file']

        imgPath = os.path.join(app.config['UPLOAD_FOLDER'], "upImage.jpg")
        file.save(imgPath)
        print(imgPath)
        prediction = predict(imgPath, imgType)
        return render_template('result.html', prediction=prediction, path='/uploads/upImage.jpg')

if __name__ == "__main__":
    app.run(debug=False)
```

Output:


Parkinsons disease prediction
127.0.0.1:5000
Parkinsons Disease Detection
The Parkinson's disease is progressive neuro degenerative disorder that affects a lot only people significantly affecting their quality of life It mostly affect the motor functions of human. The main motor symptoms are called "parkinsonism" or "parkinsonian syndrome". The symptoms of Parkinson's disease will occur slowly, the symptoms include shaking, rigidity, slowness of movement and difficulty with walking, Thinking and behavior change, Depression and anxiety are also common There is a model for detecting Parkinson's using voice. The deflections in the voice will confirm the symptoms of Parkinson's disease
Get Started
Type here to search
ENG
10:38 PM
18-11-2022


Parkinsons disease prediction
127.0.0.1:5000/upload
Parkinsons Disease Prediction
UPLOAD IMAGE
xray image
Choose File
No file chosen
Choose Type:
Spiral
Submit
Type here to search
ENG
10:38 PM
18-11-2022


Parkinsons Disease Prediction
MAGE
Submit


Diagnosed Result : healthy

8. Testing:

8.1 Testcases:

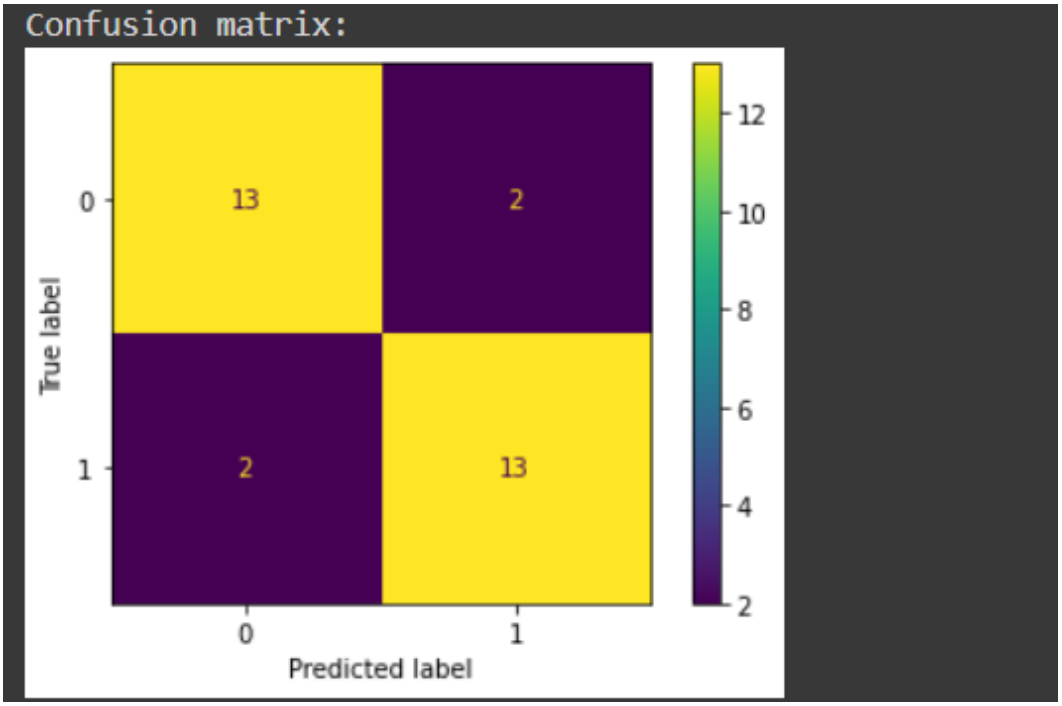
In order to test the functioning of our model, we collected a sample of Parkinson's disease and healthy handdrawn images. We tested our model against them to check if it detected the images accurately.

8.2 User Acceptance Testing:

The application performs as expected by detecting whether the patient has Parkinson's disease. All the other functionalities such as Login, Register, Update Password etc ae working as expected.

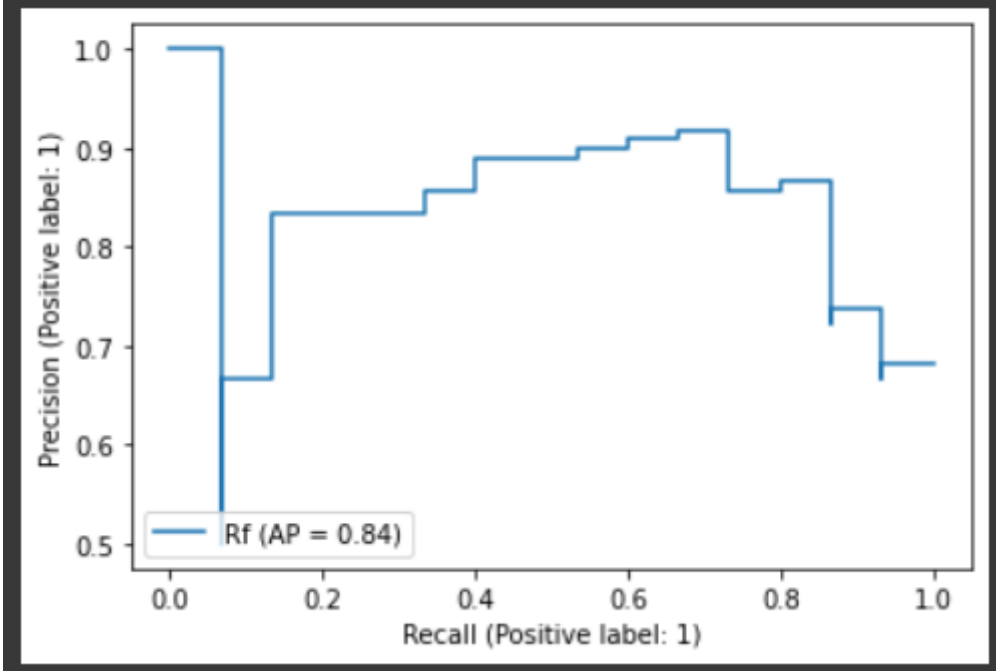
9. Results:

9.1 Performance Metrics:

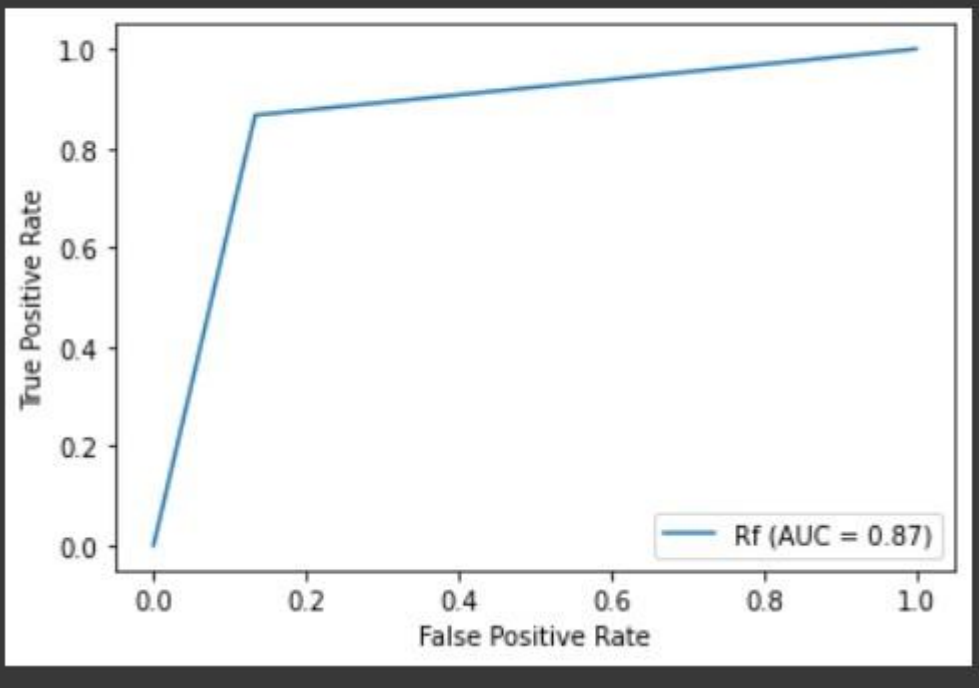


Accuracy: 0.8666666666666667
Precision: 0.8666666666666667
Recall: 0.8666666666666667
Specificity0.8666666666666667
F1 score: 0.8666666666666667

Pecision-Recall Curve:



ROC curve:



10.Advantages and Disadvantages:

Advantages:

- Easily accessible
- Application stays active 24X7
- Predictions are highly accurate
- User friendly and provides necessary information about the disease such as symptoms and causes.
- Provides medical suggestions along with results for those affected with the disease

Disadvantages:

- May not work properly on huge load (i.e. large number of requests per second)
- Packages to be installed
- Data Collection is difficult

11.Conclusion:

We have developed a web application that will help the patients to check whether they have Parkinson's disease. Thus, our application prevents expenditure on testing the disease and helps people with poor economic backgrounds. It also provides medical suggestions to those who are affected by the disease.In coclusion the model that we built using a Kneighbors Classifier gives us a good evaluation and accuracy score. So,it can be trusted for the early detection of parkinson's diseases and it can be utilized by people very easily.

12. Future Work:

In future, the work can be extended to not only predict the disease but also to find out the severity of the disease. According to the severity of the disease necessary medical suggestions and medications can be provided.The model can be trained with enormous amount of data to improve the accuracy we can also merge the voice dataset and train the model accordingly for higher productivity.

13. Appendix:

Project Demo Link:
https://drive.google.com/file/d/1qhiextEWSLFPZON7LI4JgqWrafHKd0wN/view?usp=share_link

Github:
https://github.com/IBM-EPBL/IBM-Project-44577-1660725333