Detecting Parkinson's Disease using Machine Learning

Team ID: PNT2022TMID23224

DARSHAN AJIT K R (913119205009)

DINESHKUMAR P (913119205010)

KISHORE KUMAR R (913119205019)

KRITHIK DEIVARAJAN V (913119205020)

INDEX

1. INTRODUCTION

1.1 Project Overview

More than 10 million people are living with Parkinson’s Disease worldwide, according to the Parkinson’s Foundation. While Parkinson’s cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life

The researchers found that the drawing speed was slower and the pen pressure is lower among Parkinson’s patients. One of the indications of Parkinson’s is tremors and rigidity in the muscles, making it difficult to draw smooth spirals and waves. It is possible to detect Parkinson’s disease using the drawings alone instead of measuring the speed and pressure of the pen on paper. The goal of this project is to quantify the visual appearance (using HOG method) of these drawings and then train a machine learning model to classify them. In this project, Histogram of Oriented Gradients (HOG) image descriptor is used along with Random Forest and K Nearest Neighbour classifier to automatically detect Parkinson’s disease in hand-drawn images of spirals and waves.

1.2 Purpose

In this Python Machine Learning project, a model will be built to detect Parkinson’s disease using two of the Classifier techniques known as K Nearest Neighbour Classifier and Random Forest Classifier as our output contains only 1’s and 0’s. The dataset is loaded, the features and targets are identified and split into training and testing sets and are finally passed to K Nearest Neighbour Classifier and Random Forest Classifier for prediction.

2. LITERATURE SURVEY

2.1 Existing Problem

In the existing system, Parkinson’s Disease (PD) is usually only detected at the secondary stage (Dopamine deficiency) which leads to the onset of many comorbidities in the patient. The doctor must manually examine and determine the diagnosis. The symptoms also vary from person to person which makes suggesting medicines a challenge. This leads to a lot of late diagnoses as well as misdiagnoses which makes the treatment process more complex and expensive in addition to leading to health complications for the patient. At the moment, there is no single medical diagnosis test to diagnose PD. Doctors carry a number of tests to rule out the presence or absence of PD. These tests include Single-Photon Emission Computerized Tomography (SPECT) scan called a dopamine transporter (DAT) scan. A DaTscan involves an injection of a small amount of a radioactive drug and a machine called a single-photon emission computed tomography (SPECT) scanner, similar to an MRI. The drug binds to dopamine transmitters in the brain, showing where in the brain dopaminergic neurons are. Dopaminergic neurons are the source of dopamine in the brain. A loss of dopamine is what leads to Parkinson’s. Lab tests, such as blood tests, to rule out other conditions that may be causing the symptoms. Imaging tests such as an MRI, ultrasound of the brain and PET scans also may be used to help rule out other disorders. Imaging tests aren't particularly helpful for diagnosing Parkinson's disease. This results in a high misdiagnosis rate (up to 25% by non-specialists) and many years before diagnosis, people can have the disease. Thus, existing system is not effective in early prediction and accurate medicinal diagnosis to the affected people.

2.2 References

[1]. Sakshi Jadhav, Seema Thorat, Sakshi Fokane, Rahul Chakre, “Classification of Parkinson’s disease using Machine Learning Techniques”,2022.

[2]. Jie Mei et al. “Machine learning for the diagnosis of Parkinson’s disease”,2021.

[3]. Atiqur Rahman, Sanam Shahla Rizvi, Aurangzeb Khan, et al. “Parkinson’s Disease Diagnosis in Cepstral Domain Using MFCC and Dimensionality Reduction with SVM Classifier”,2021.

[4]. Mosarrat Rumman, Abu Nayeem Tasneemet et al. “Early detection of Parkinson's disease using image processing and artificial neural network”,2019.

[5]. Mahima Thakur, Harisudha Kuresan, Samiappan Dhanalakshmi et al. “Soft Attention Based DenseNet Model for Parkinson’s Disease Classification Using SPECT Images”,2022.

[6]. Ankit kurmi, Shreya Biswas, Ram Sarkar et al. “An Ensemble of CNN Models for Parkinson’s Disease Detection Using DaTscan Images”,2022.

[7]. Sumeet Shinde , Shweta Prasad , Yash Saboo et al. “Predictive markers for Parkinson's disease using deep neural nets on neuromelanin sensitive MRI”,2019.

[8]. Zhennao Cai, Jianhua Gu, Caiyun Wen, Dong Zhao et al. “An Intelligent Parkinson’s Disease Diagnostic System Based on a Chaotic Bacterial Foraging Optimization Enhanced Fuzzy KNN Approach”,2018.

2.3 Problem Statement Definition

Instead of monitoring the speed and pressure with which the pen strikes the paper, it is possible to diagnose Parkinson's disease solely by looking at the drawings. The objective is to use the HOG (Histogram of Oriented Gradients) image descriptor method to assess these drawings' visual appearance before training a machine learning model to categorise them. In this research, Random Forest classifier and K Nearest Neighbour classifier are utilized to automatically identify Parkinson's disease in hand-drawn spirals and waves.

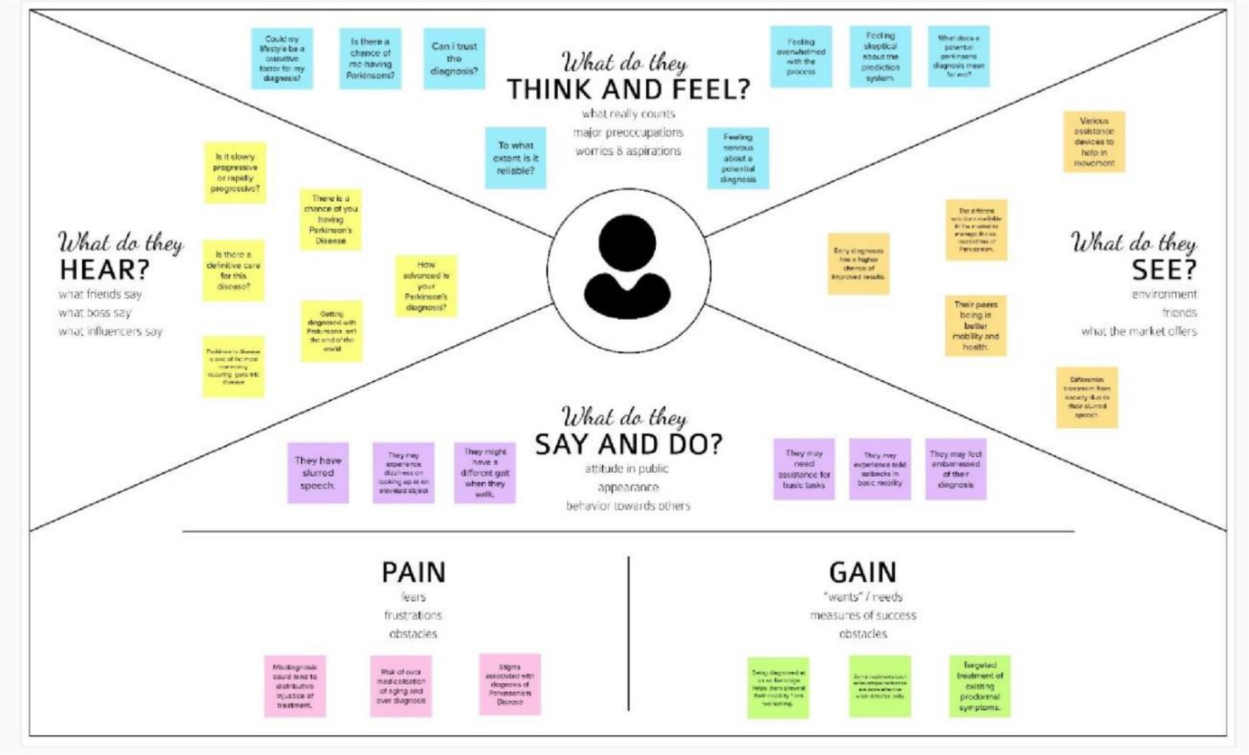
Problem Statement (PS)	**I am**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	A person with mild tremors in my hands	Determine if I have any health issues	Since healthcare and testing is expensive	I am not very financially well off	Anxious that I may have a problem that requires medical assistance, but I might not be able to receive it.
PS-2	A 50year-old man with impaired posture and balance that seems to get worse with time	Seek out a diagnosis for my ailment	I don't live close to a specialty hospital for screening for neuro problems	I live in a rural area which gives me access to a primary health care centre at most.	Worried and concerned that I have parkinsonism and don't have the ability to avail any treatment soon due to lack of diagnosis.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

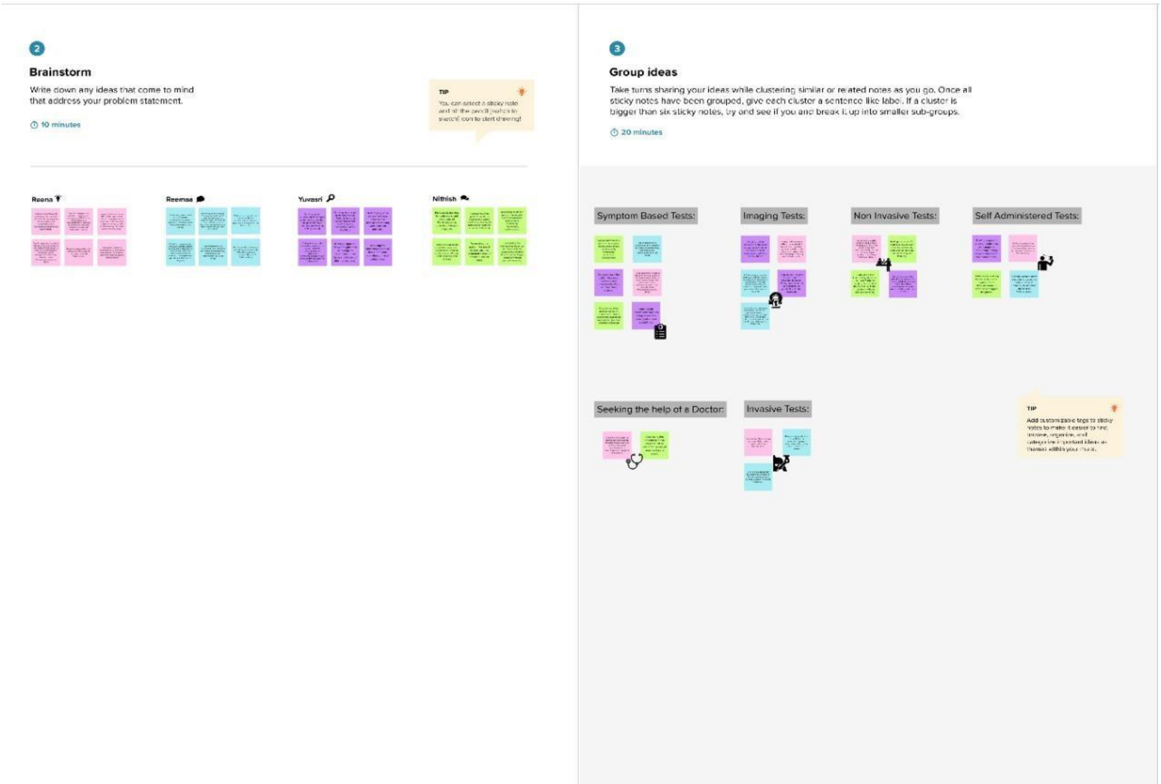
`An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users.Creating an effective solution requires understanding the true problem and the person who is experiencing it.

The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.


What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Could my lifestyle be a causative factor for my diagnosis?
Is there a chance of me having Parkinsons?
Can i trust the diagnosis?
To what extent is it reliable?
Feeling nervous about a potential diagnosis
Feeling overwhelmed with the process
Feeling skeptical about the prediction system.
What does a potential parkinsons diagnosis mean for me?
What do they
HEAR?
what friends say
what boss say
what influencers say
Is it slowly progressive or rapidly progressive?
There is a chance of you having Parkinson's Disease
Is there a definitive cure for this disease?
How advanced is your Parkinson's diagnosis?
Getting diagnosed with Parkinsons isn't the end of the world
Parkinson's disease is one of the most commonly occurring genetic disease
What do they
SEE?
environment
friends
what the market offers
Various assistance devices to help in movement
The different solutions available in the market to manage the restrictions of Parkinson.
Early diagnosis has a higher chance of improved results.
Their peers being in better mobility and health.
Differential treatment from society due to their slurred speech
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
They have slurred speech.
They may experience dizziness on looking up at an elevated object
They might have a different gait when they walk.
They may need assistance for basic tasks
They may experience mild setbacks in basic mobility
They may feel embarrassed of their diagnosis
PAIN
fears
frustrations
obstacles
Misdiagnosis could lead to distributive injustice of treatment.
Risk of over medicalisation of aging and over diagnosis
Stigma associated with diagnosis of Parkinsonism Disease
GAIN
"wants" / needs
measures of success
obstacles
Being diagnosed at an early stage helps them prevent their mobility from worsening.
Targeted treatment of existing prodormal symptoms.

3.2 Ideation & Brainstorming

Ideation and Brainstorming are performed to generate ideas and solutions. Brainstorming is a group activity unlike ideation.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!

Reena

Reemaa

Yuvasri

Nithish

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Symptom Based Tests:

Imaging Tests:

Non Invasive Tests:

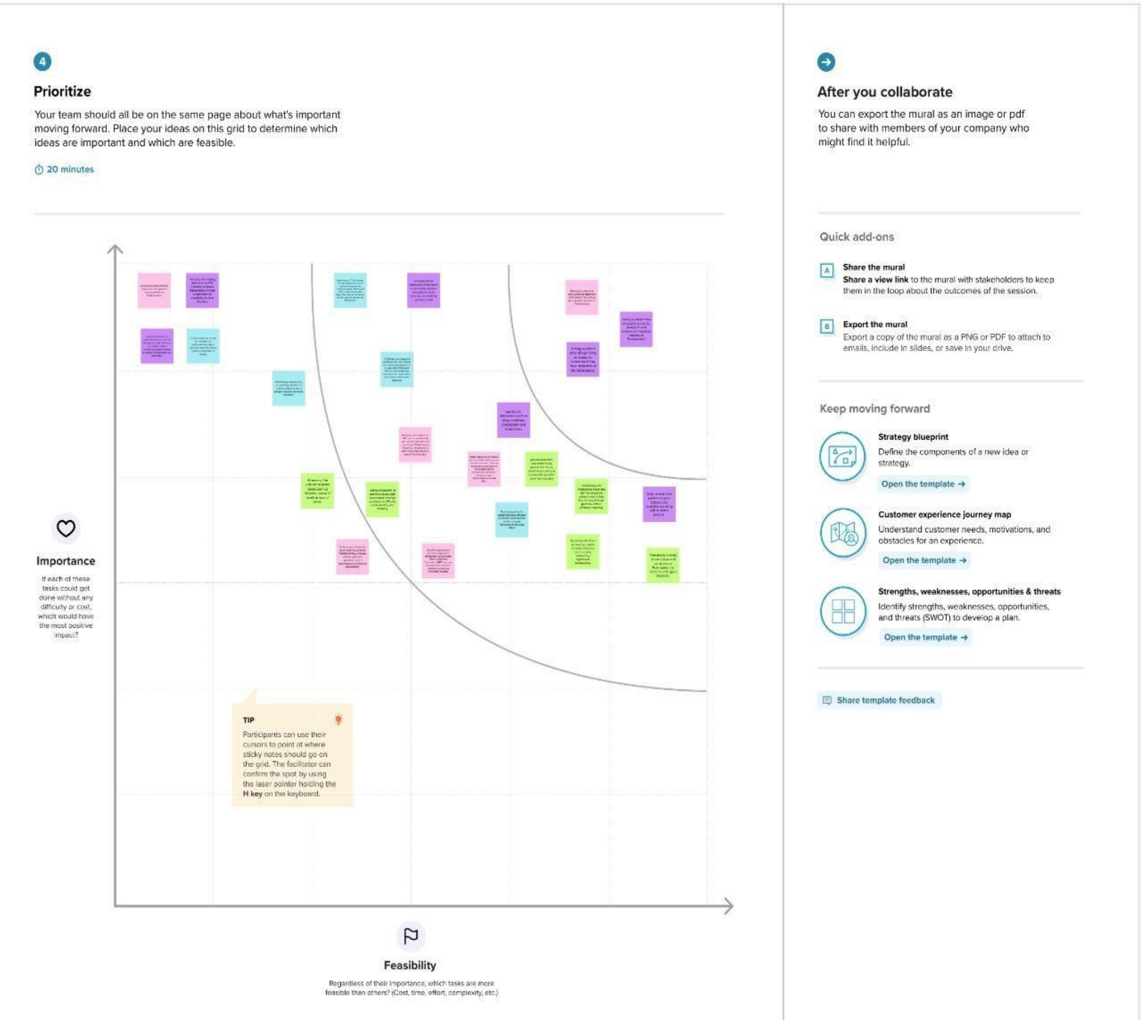
Self Administered Tests:

Seeking the help of a Doctor:

Invasive Tests:

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

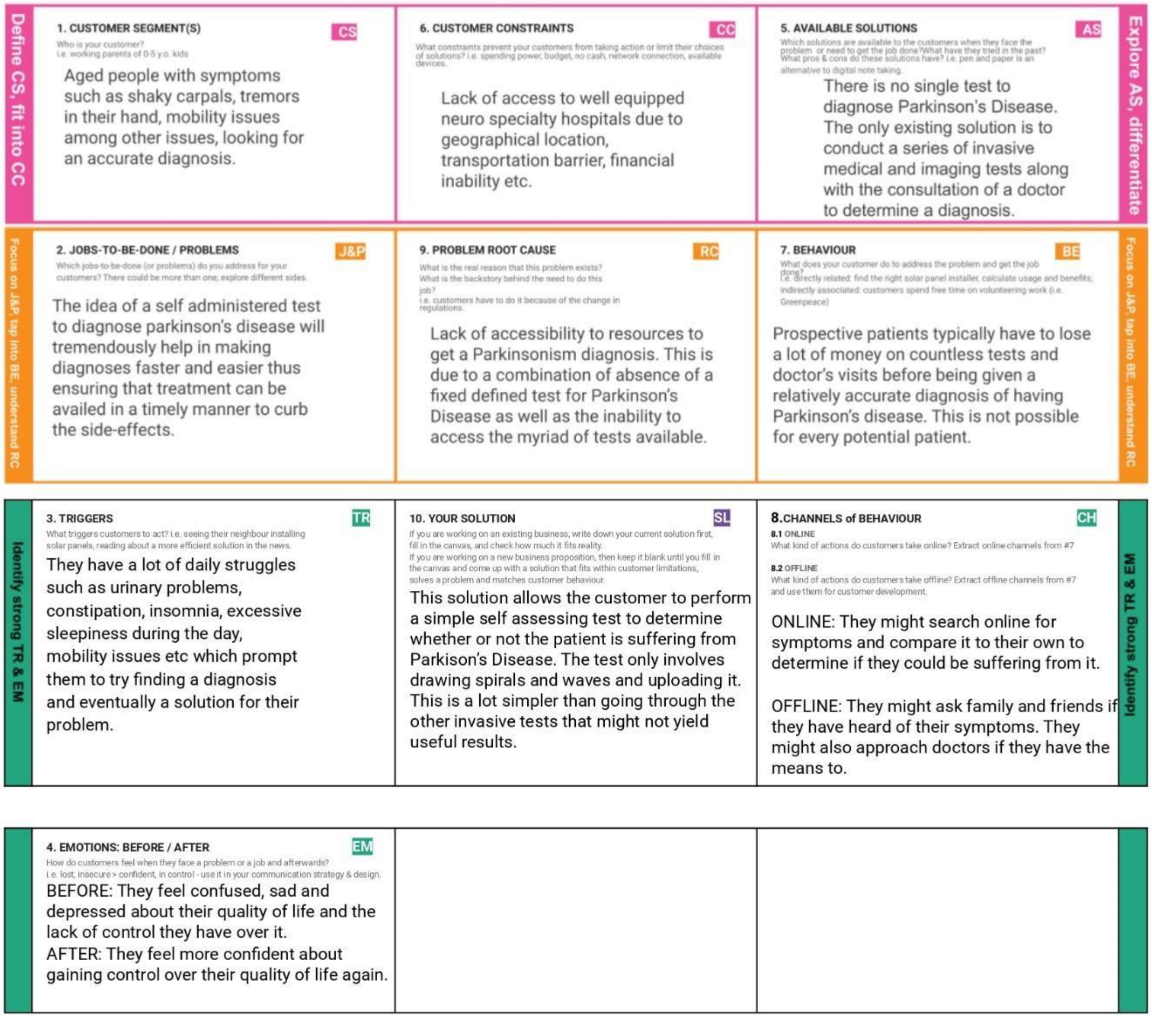
Share template feedback

3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	More than 10 million people are living with Parkinson’s Disease worldwide, according to the Parkinson’s Foundation. While Parkinson’s cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life. Detecting Parkinson’s disease from the spirals and waves drawn by the patients using a Machine Learning Model is the Problem Statement.

2.	Idea / Solution description	One of the major symptoms of Parkinson's disease is tremors or rhythmic shaking of the carpals (hands). This results in slower drawing speed and lower pen pressure in Parkinson's patients. A direct result of this is that the spirals and waves drawn by the Parkinson's patients look significantly different. Thus, the spirals and waves drawn are used to determine if the patient has Parkinson's disease.
3.	Novelty / Uniqueness	Currently, there is no test similar to this in the market. All other available tests are medically invasive in nature and need financial remittance to conduct. This test is completely free making it more accessible.
4.	Social Impact / Customer Satisfaction	This will help countless potential patients get a quicker diagnosis which will result in better quality of life due to being prescribed the correct medication that will help in improving the symptoms.
5.	Business Model (Revenue Model)	Increase in traffic to the website is a direct reflection of the increased users which will ultimately help in increasing the ad revenue of the developer of the website.
6.	Scalability of the Solution	This solution can gradually be expanded to also include diagnoses for diseases like Essential Tremor and Normal Pressure Hydrocephalus, both of which manifest in the form of slowness (bradykinesia), stiffness (rigidity), and resting tremor.

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Aged people with symptoms such as shaky carpals, tremors in their hand, mobility issues among other issues, looking for an accurate diagnosis.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Lack of access to well equipped neuro specialty hospitals due to geographical location, transportation barrier, financial inability etc.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done?What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital note taking

There is no single test to diagnose Parkinson's Disease. The only existing solution is to conduct a series of invasive medical and imaging tests along with the consultation of a doctor to determine a diagnosis.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

The idea of a self administered test to diagnose parkinson's disease will tremendously help in making diagnoses faster and easier thus ensuring that treatment can be availed in a timely manner to curb the side-effects.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the backstory behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Lack of accessibility to resources to get a Parkinsonism diagnosis. This is due to a combination of absence of a fixed defined test for Parkinson's Disease as well as the inability to access the myriad of tests available.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Prospective patients typically have to lose a lot of money on countless tests and doctor's visits before being given a relatively accurate diagnosis of having Parkinson's disease. This is not possible for every potential patient.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

They have a lot of daily struggles such as urinary problems, constipation, insomnia, excessive sleepiness during the day, mobility issues etc which prompt them to try finding a diagnosis and eventually a solution for their problem.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

This solution allows the customer to perform a simple self assessing test to determine whether or not the patient is suffering from Parkison's Disease. The test only involves drawing spirals and waves and uploading it. This is a lot simpler than going through the other invasive tests that might not yield useful results.

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

ONLINE: They might search online for symptoms and compare it to their own to determine if they could be suffering from it.

OFFLINE: They might ask family and friends if they have heard of their symptoms. They might also approach doctors if they have the means to.

Identify strong TR & EM

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

BEFORE: They feel confused, sad and depressed about their quality of life and the lack of control they have over it.
AFTER: They feel more confident about gaining control over their quality of life again.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Home Page	Short description about Parkinson's Disease, its different types, and symptoms along with possible comorbidity management techniques. If the user already has an account, they can log in. Otherwise, they are required to sign up.
FR-2	User Registration (Sign in) Page	User needs to sign up/ register by entering Name, Email address, Phone number and Password.
FR-3	User Confirmation & Verification	Verification will be done via Email or OTP.
FR-4	User Login Page	User can enter their credentials (Email and Password) and log in to their account.

FR-5	User Dashboard	The logged in user is led to a dashboard where the user is asked to upload the image in order to provide the diagnosis. The user is also asked for other parameters such as age, blood type, mobility issues etc for survey purposes. This information is optional and is collected only from willing users.
FR-6	Test input (Copy of handdrawn image)	The input to the prediction system is uploaded as an image. It can be uploaded either using a live drawing notepad or as the digital copy of an already drawn spiral/wave. Image quality evaluation is done in this step to determine whether the image quality is sufficient for processing.
FR-7	User authentication during login	User authentication is done using PHP via database in XAMPP server.
FR-8	Disease prediction by image processing	Classification is carried out using Digital image processing using Histogram of Oriented Gradients (HOG) image descriptor along with a random forest classifier.
FR-9	Recommendation	The prediction system provides a positive or negative diagnosis. It also suggests the specialization doctors that need to be consulted. The system arrives at the result by analysing the standards defined by Movement Disorder Society Unified Parkinson’s Rating Scale and progression of the disease.

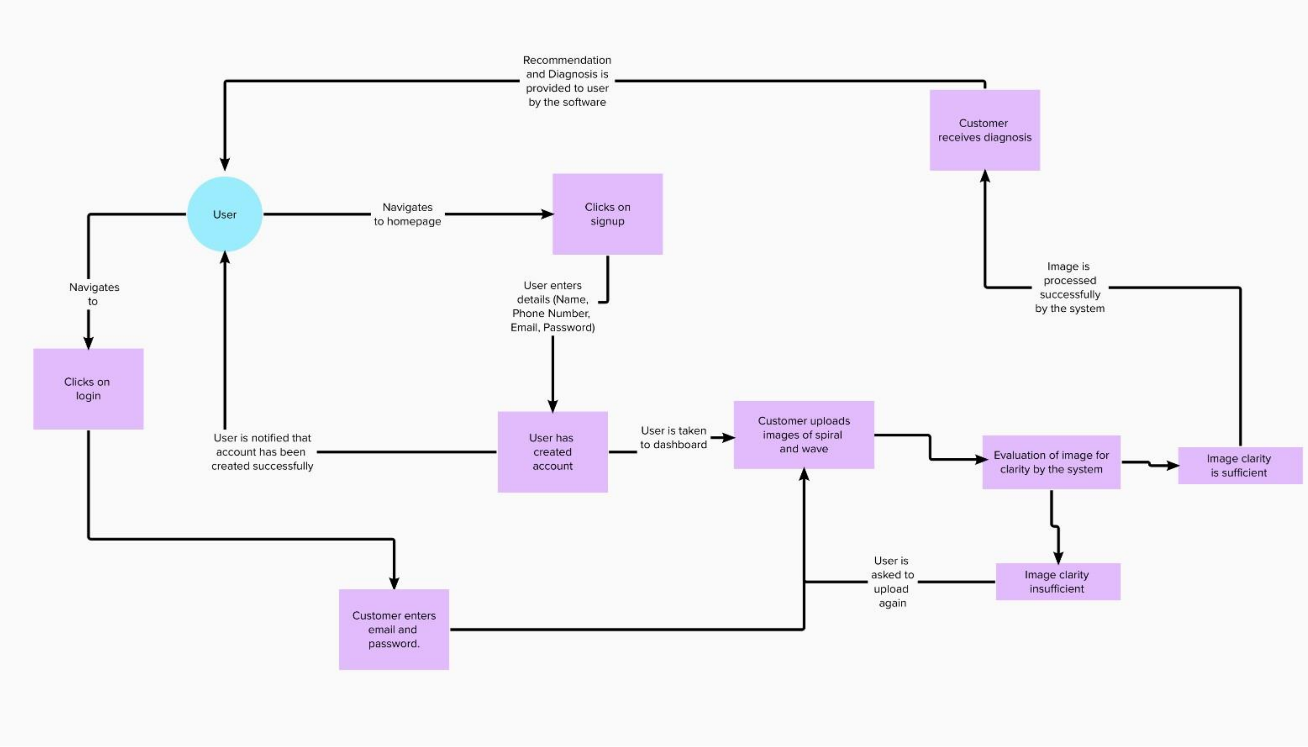
4.2 Non-Functional Requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The website can be easily navigated even by the uninitiated user and the functionality that the website provides is simple and easy to understand.
NFR-2	**Security**	The application is designed to safeguard against threats including unauthorized access and protects the patient’s confidentiality by keeping patient details visible only to admin and the patient. Access permissions can only be changed by the system’s data administrator.
NFR-3	**Reliability**	The software will work without failure. It does not
		have any security bugs. The model is trained with different visuals for detecting the disease, which leads to a more accurate assessment of a disease, thereby making the system more reliable for its users.

NFR-4	**Performance**	The system is very responsive to user interactions with it and can handle a large traffic without getting overloaded. The user wait time is not prolonged, including capturing and uploading to prediction and providing recommendations.
NFR-5	**Availability**	The software is always available to the user irrespective of the any new module development. If any backend work requires that the page be unavailable, then a notification is displayed to the user informing when it will up again for use. The software can also be utilized by anyone, regardless of the customer location or other network capabilities.
NFR-6	**Scalability**	The system has the ability to grow without any negative impact on its performance. The system is designed in a way it can withstand a large number of users at any given moment and if need be, can be scaled up to handle even more users.

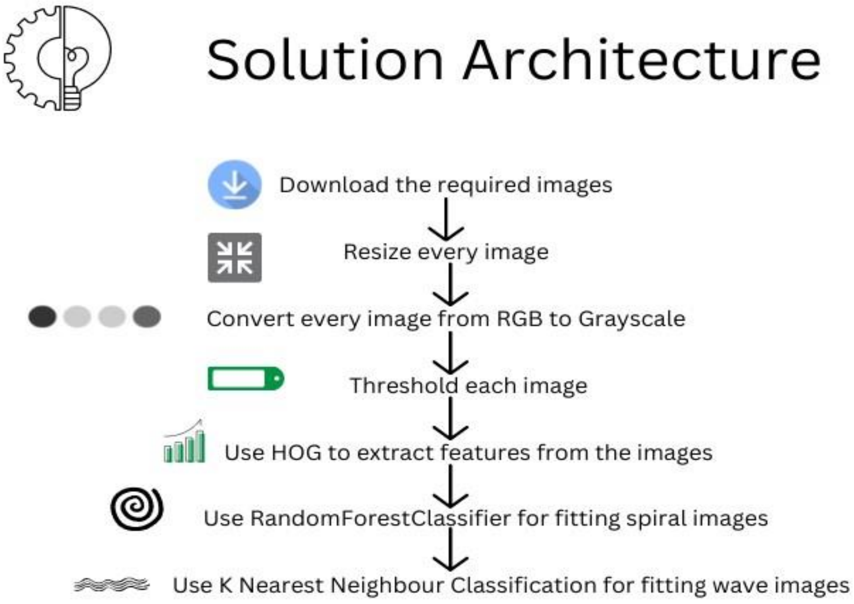
5. PROJECT DESIGN

5.1 Data Flow Diagrams


Recommendation and Diagnosis is provided to user by the software
Customer receives diagnosis
User
Navigates to homepage
Clicks on signup
Navigates to
User enters details (Name, Phone Number, Email, Password)
Image is processed successfully by the system
Clicks on login
User is notified that account has been created successfully
User has created account
User is taken to dashboard
Customer uploads images of spiral and wave
Evaluation of image for clarity by the system
Image clarity is sufficient
User is asked to upload again
Image clarity insufficient
Customer enters email and password.

5.2 Solution & Technical Architecture

Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios, integration requirements, etc.



Solution Architecture

Download the required images

Resize every image

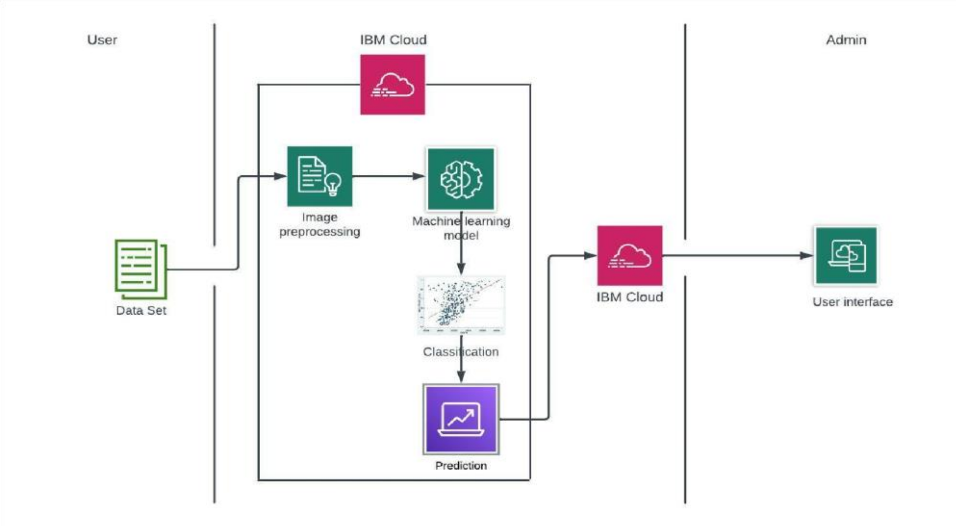
Convert every image from RGB to Grayscale

Threshold each image

Use HOG to extract features from the images

Use RandomForestClassifier for fitting spiral images

Use K Nearest Neighbour Classification for fitting wave images



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Viewing Home Page	USN-1	As a user, I can view the home page which has a description of the disease as well as options to sign up or log in.	I can get to know about the disease and its symptoms as well as navigate to sign up page and log in page from there.	Low	Sprint1
	Sign Up Page	USN-2	As a user, I can register for the application by entering my name, phone number, email, password, and confirming my password.	I can login with my credentials.	High	Sprint1
	Authorization	USN-3	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm.	High	Sprint2
	Login	USN-4	As a user, I can log into the application by entering email & password.	I can access my account / dashboard after logging in successfully.	High	Sprint1
	Dashboard	USN-5	As a user, I can upload images of spiral and wave to the website in order to receive a diagnosis.	I can successfully access the dashboard to upload the images.	High	Sprint2
	Results	USN-6	As a user, I can receive a diagnosis in addition to recommendations	I can access the diagnosis and possible available	High	Sprint3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			on what I should do now.	solutions.		
Administrator	Data Collection	USN-7	I need to collect data (images of spirals and waves drawn by healthy people and Parkinson’s patients).	I have sizable amount of data to split into training set and testing set.	High	Sprint2
	Data Pre-Processing	USN-8	I need to clean my data and prepare it for model building by doing preprocessing activities such as resizing, converting from RGB to grayscale etc.	I have the dataset ready for model building.	High	Sprint3
	Model Building	USN-9	I need to build the model using Random Forest Classifier for spiral images and K Nearest Neighbour (KNN) for wave images.	The model is ready for deployment on testing data.	High	Sprint4
	Model Deployment	USN-10	I need to deploy the Machine Learning model that was built.	The model has been deployed successfully.	Medium	Sprint5
	Application Building	USN-11	I need to build the website for the application using HTML, CSS etc.	The website is functional.	High	Sprint3
	Linking Model and Application	USN-12	I can integrate the deployed model and web application using python flask	The web application is fully functional and can be	High	Sprint5
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**

			server.	used by the user.		

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Viewing Home Page	USN-1	As a user, I can view the home page which has a description of the disease as well as options to sign up or log in.	2	Low	Kishore kumar R Dineshkumar P
Sprint-4	Sign Up Page	USN-2	As a user, I can register for the application by entering my name, phone number, email, password, and confirming my password.	2	High	Krithik Deivarajan V Darshan Ajit K R
Sprint-4	Authorization	USN-3	As a user, I will receive confirmation email once I have registered for the application.	2	High	Kishore kumar R Dineshkumar P
Sprint-4	Login	USN-4	As a user, I can log into the application by entering email & password.	2	High	Krithik Deivarajan V Darshan Ajit K R
Sprint-4	Dashboard	USN-5	As a user, I can upload images of spiral and wave to the website in order to receive a diagnosis.	2	High	Kishore kumar R Dineshkumar P
Sprint-4	Results	USN-6	As a user, I can receive a diagnosis in addition to recommendations on what I should do now.	2	High	Krithik Deivarajan V Darshan Ajit K R

Sprint-1	Data Collection	USN-7	I need to collect data (images of spirals and waves drawn by healthy people and Parkinson's patients).	5	High	Kishore kumar R Dineshkumar P
Sprint-1	Data Pre-Processing	USN-8	I need to clean my data and prepare it for model building by doing preprocessing activities such as resizing, converting from RGB to grayscale etc.	5	High	Krithik Deivarajan V Darshan Ajit K R
Sprint-2	Model Building 1	USN-9	I need to build the model using Random Forest Classifier for spiral images.	8	High	Kishore kumar R Dineshkumar P
Sprint 2	Model Building 2	USN-10	I need to build the model using K Nearest Neighbour (KNN) for wave images.	8	High	Krithik Deivarajan V Darshan Ajit K R
Sprint-3	Model Deployment	USN-11	I need to deploy the Machine Learning model that was built.	13	Medium	Kishore kumar R Dineshkumar P
Sprint-4	Application Building	USN-12	I need to build the website for the application using HTML, CSS and link it to the model.	8	High	Krithik Deivarajan V Darshan Ajit K R

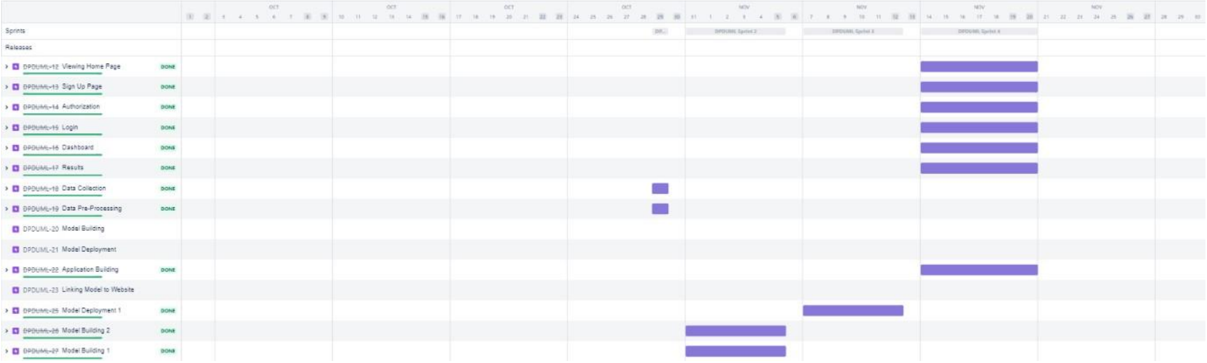
6.2 Sprint Delivery Schedule

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-2	16	6 Days	31 Oct 2022	05 Nov 2022	16	05 Nov 2022
Sprint-3	13	6 Days	07 Nov 2022	12 Nov 2022	13	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports From JIRA

Roadmap weekly sprint



Sprints

Releases

~~DPDUML-12~~ Viewing Home Page DONE

~~DPDUML-13~~ Sign Up Page DONE

~~DPDUML-14~~ Authorization DONE

~~DPDUML-15~~ Login DONE

~~DPDUML-16~~ Dashboard DONE

~~DPDUML-17~~ Results DONE

~~DPDUML-18~~ Data Collection DONE

~~DPDUML-19~~ Data Pre-Processing DONE

DPDUML-20 Model Building

DPDUML-21 Model Deployment

~~DPDUML-22~~ Application Building DONE

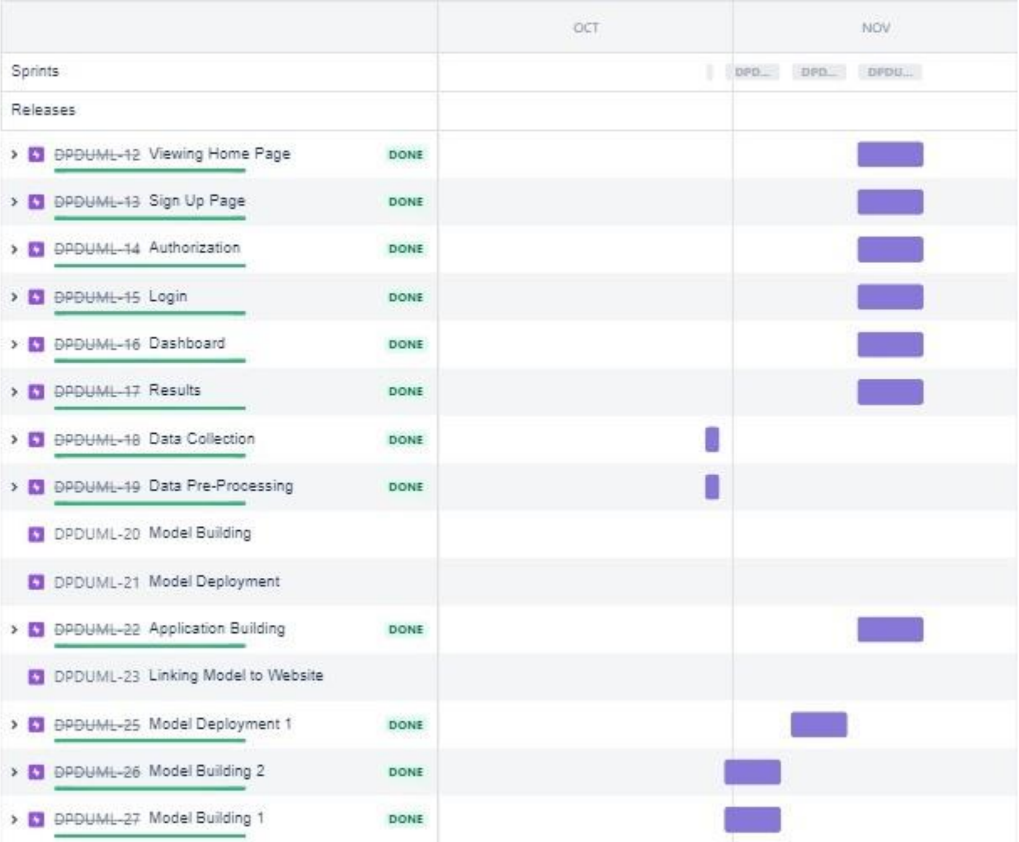
DPDUML-23 Linking Model to Website

~~DPDUML-25~~ Model Deployment 1 DONE

~~DPDUML-26~~ Model Building 2 DONE

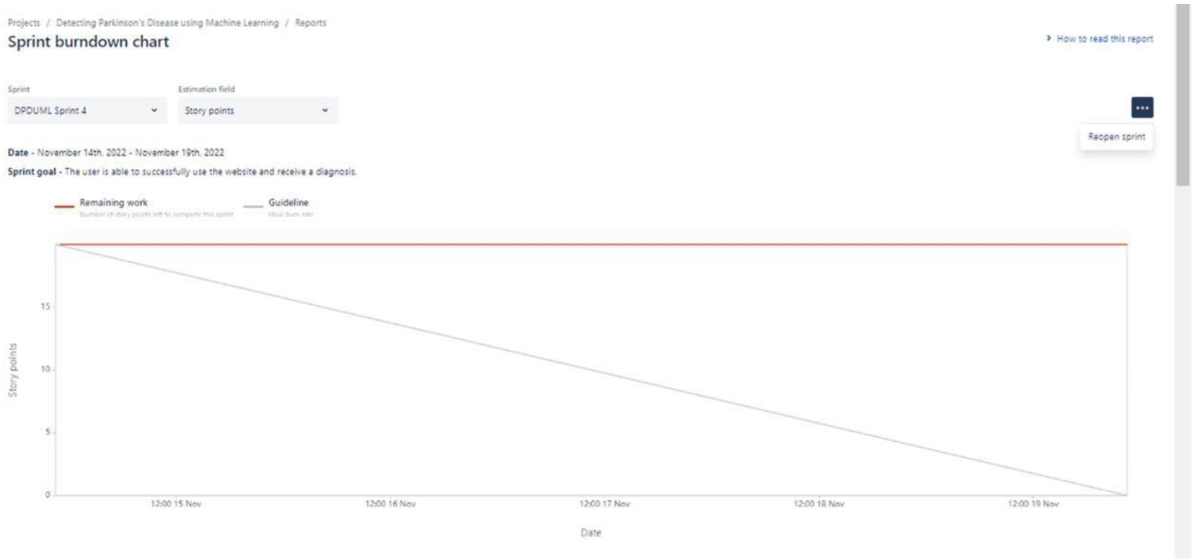
~~DPDUML-27~~ Model Building 1 DONE

Roadmap monthly sprint



		OCT	NOV
Sprints			DPD... DPD... DPDU...
Releases			
~~DPDUML-12~~ Viewing Home Page	DONE		
~~DPDUML-13~~ Sign Up Page	DONE		
~~DPDUML-14~~ Authorization	DONE		
~~DPDUML-15~~ Login	DONE		
~~DPDUML-16~~ Dashboard	DONE		
~~DPDUML-17~~ Results	DONE		
~~DPDUML-18~~ Data Collection	DONE		
~~DPDUML-19~~ Data Pre-Processing	DONE		
DPDUML-20 Model Building			
DPDUML-21 Model Deployment			
~~DPDUML-22~~ Application Building	DONE		
DPDUML-23 Linking Model to Website			
~~DPDUML-25~~ Model Deployment 1	DONE		
~~DPDUML-26~~ Model Building 2	DONE		
~~DPDUML-27~~ Model Building 1	DONE		

Burndown Report



Projects / Detecting Parkinson's Disease using Machine Learning / Reports

Sprint burndown chart

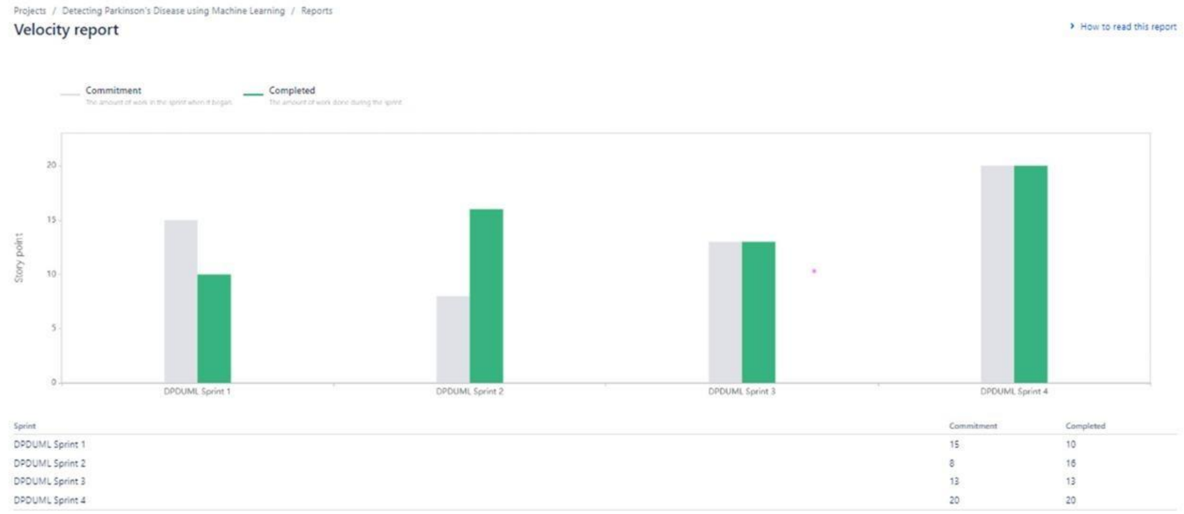
Sprint: DPDUML Sprint 4

Estimation field: Story points

Date - November 14th, 2022 - November 19th, 2022

Sprint goal - The user is able to successfully use the website and receive a diagnosis.

Velocity Report



Projects / Detecting Parkinson's Disease using Machine Learning / Reports

Velocity report

› How to read this report

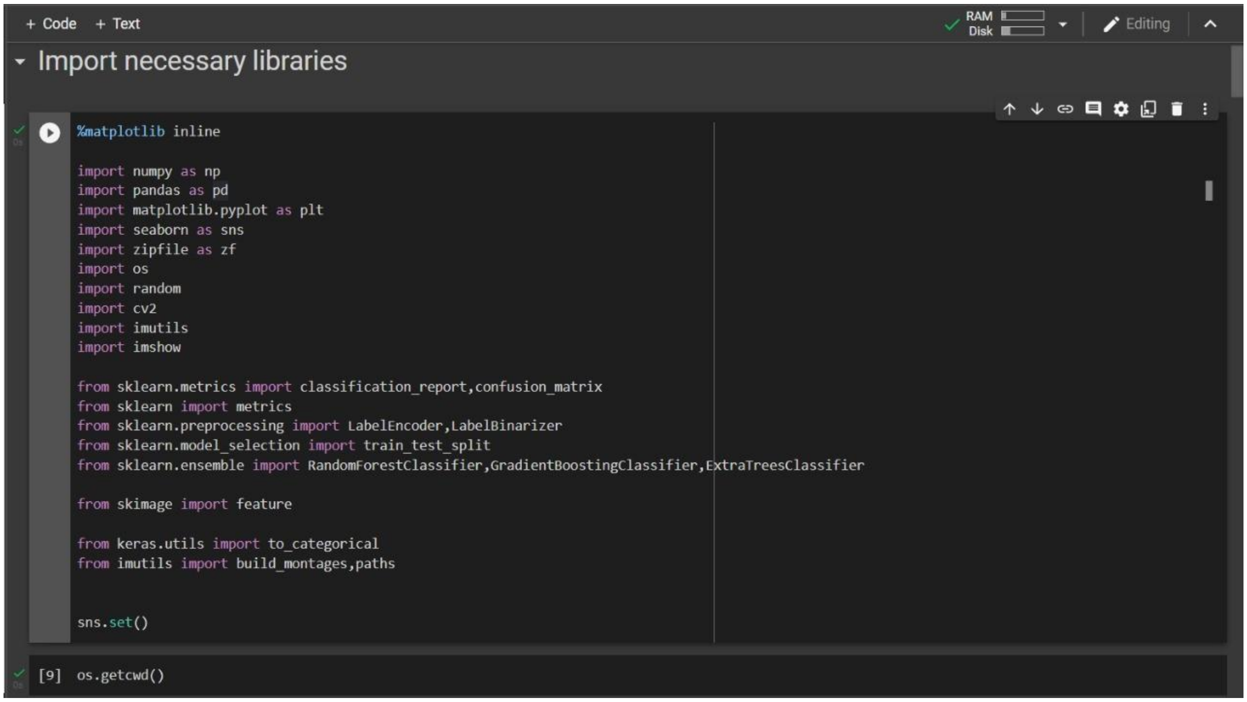
Sprint	Commitment	Completed
DPDUML Sprint 1	15	10
DPDUML Sprint 2	8	16
DPDUML Sprint 3	13	13
DPDUML Sprint 4	20	20

7. CODING & SOLUTIONING

7.1 Feature 1

Two Machine Learning models have been trained, one using spiral image and another using wave image to detect Parkinson's Disease using Random Forest Classifier. Both the models have an average accuracy of 83%.

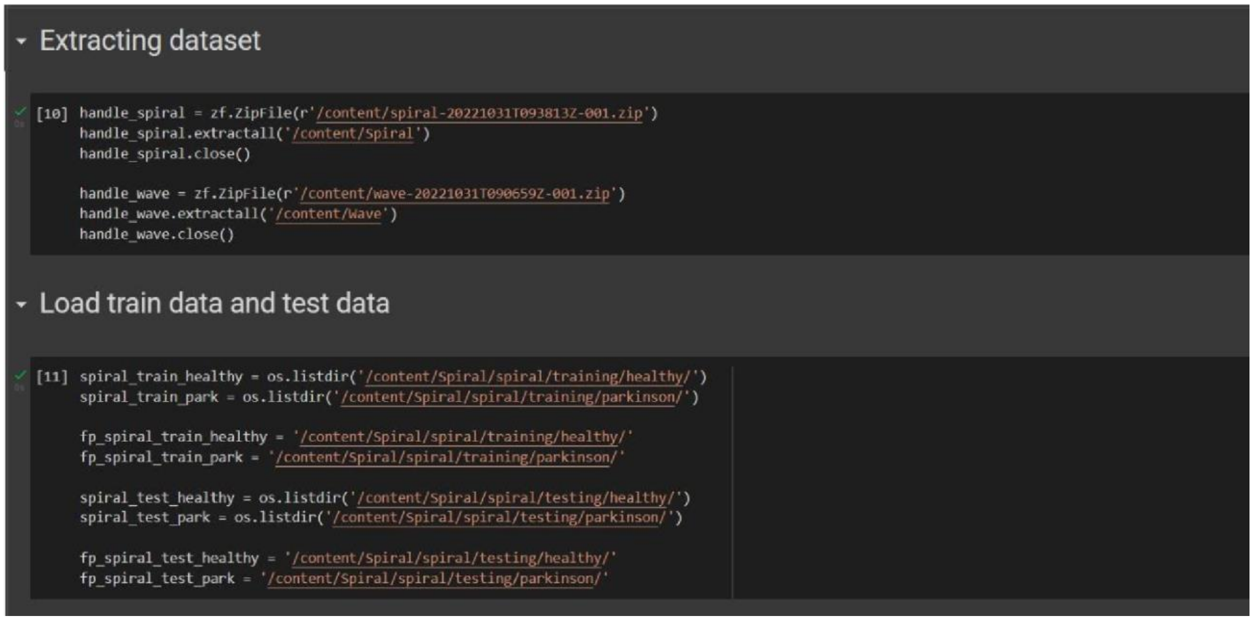
Training the model for Spiral image:



Import necessary libraries

```
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import zipfile as zf
import os
import random
import cv2
import imutils
import imshow

from sklearn.metrics import classification_report,confusion_matrix
from sklearn import metrics
from sklearn.preprocessing import LabelEncoder,LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier,ExtraTreesClassifier

from skimage import feature

from keras.utils import to_categorical
from imutils import build_montages,paths


sns.set()
```

```
[9] os.getcwd()
```

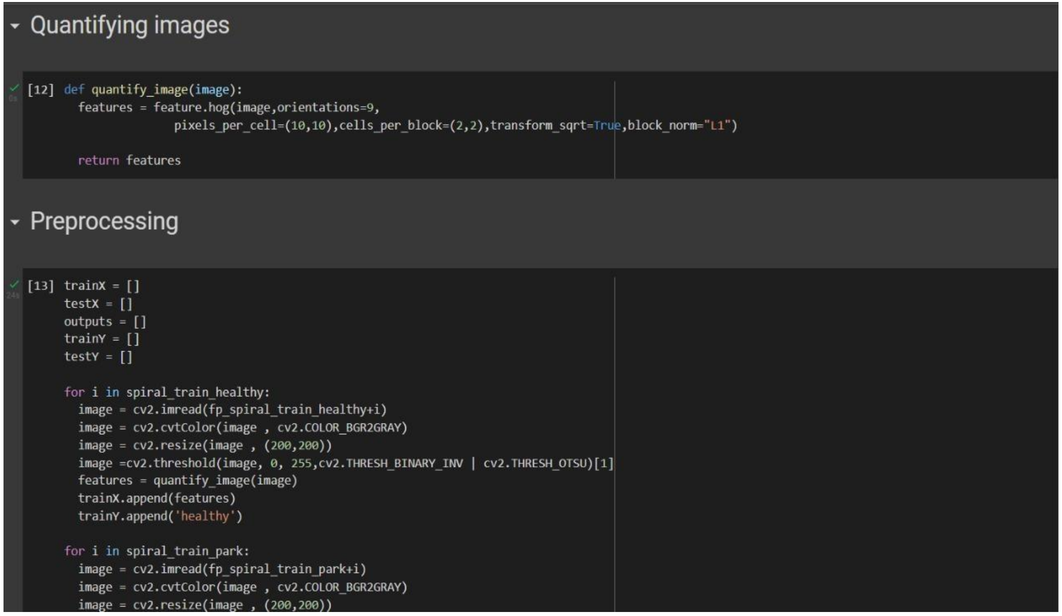


Extracting dataset

```
[10] handle_spiral = zf.ZipFile(r'/content/spiral-20221031T093813Z-001.zip')
     handle_spiral.extractall('/content/Spiral')
     handle_spiral.close()

     handle_wave = zf.ZipFile(r'/content/wave-20221031T090659Z-001.zip')
     handle_wave.extractall('/content/Wave')
     handle_wave.close()
```

Load train data and test data

```
[11] spiral_train_healthy = os.listdir('/content/Spiral/spiral/training/healthy/')
     spiral_train_park = os.listdir('/content/Spiral/spiral/training/parkinson/')

     fp_spiral_train_healthy = '/content/Spiral/spiral/training/healthy/'
     fp_spiral_train_park = '/content/Spiral/spiral/training/parkinson/'

     spiral_test_healthy = os.listdir('/content/Spiral/spiral/testing/healthy/')
     spiral_test_park = os.listdir('/content/Spiral/spiral/testing/parkinson/')

     fp_spiral_test_healthy = '/content/Spiral/spiral/testing/healthy/'
     fp_spiral_test_park = '/content/Spiral/spiral/testing/parkinson/'
```

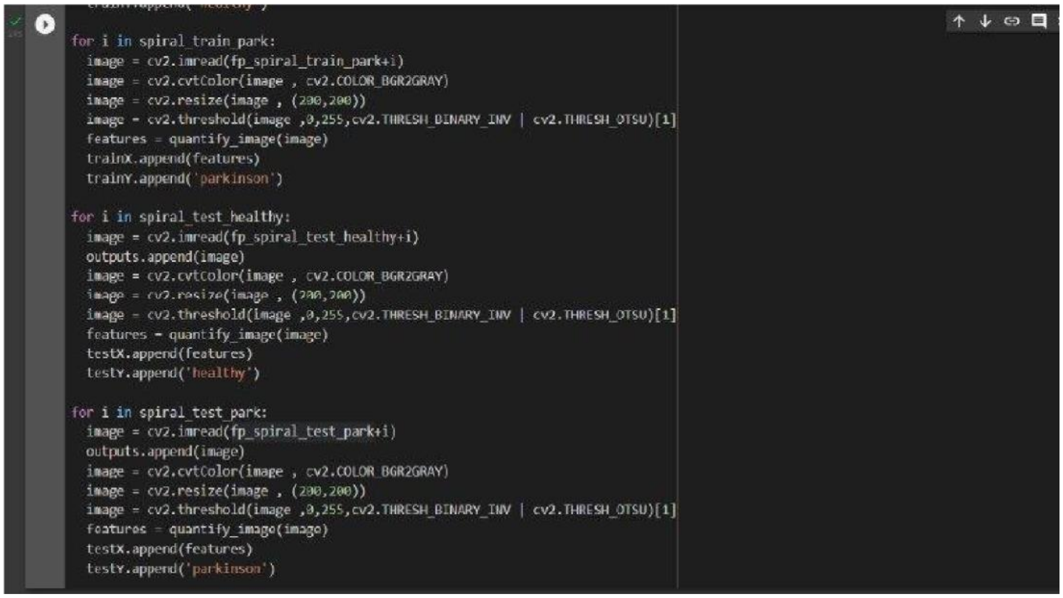


Quantifying images

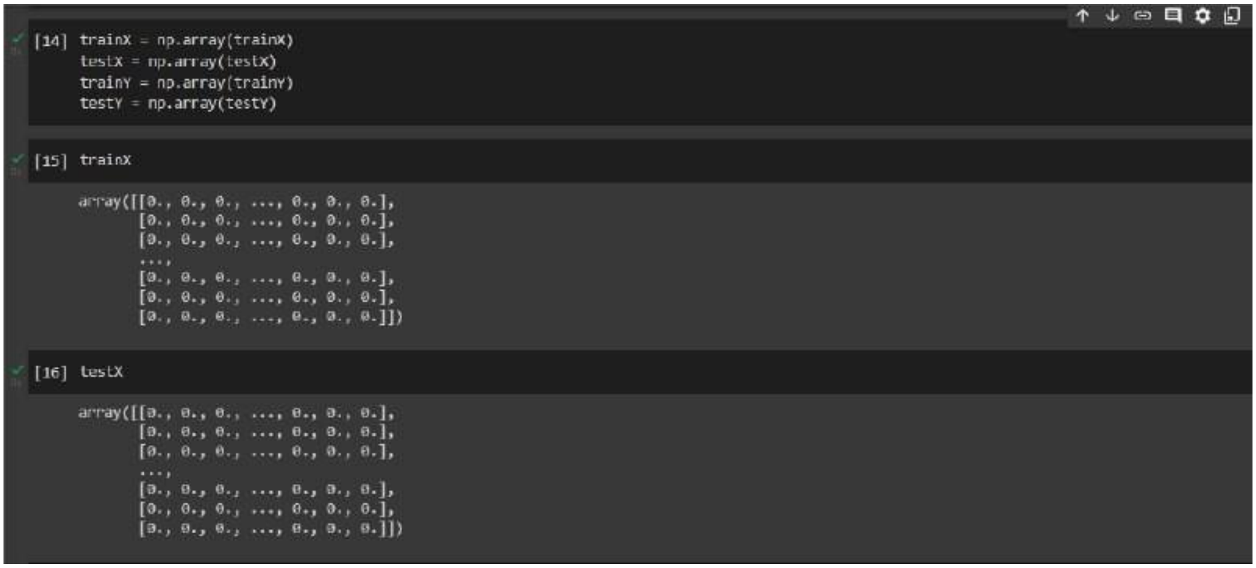
```
[12] def quantify_image(image):
       features = feature.hog(image,orientations=9,
                    pixels_per_cell=(10,10),cells_per_block=(2,2),transform_sqrt=True,block_norm="L1")

       return features
```

Preprocessing

```
[13] trainX = []
     testX = []
     outputs = []
     trainY = []
     testY = []

     for i in spiral_train_healthy:
       image = cv2.imread(fp_spiral_train_healthy+i)
       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]
       features = quantify_image(image)
       trainX.append(features)
       trainY.append('healthy')

     for i in spiral_train_park:
       image = cv2.imread(fp_spiral_train_park+i)
       image = cv2.cvtColor(image , cv2.COLOR_BGR2GRAY)
       image = cv2.resize(image , (200,200))
```



```
       trainY.append('healthy')

     for i in spiral_train_park:
       image = cv2.imread(fp_spiral_train_park+i)
       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]
       features = quantify_image(image)
       trainX.append(features)
       trainY.append('parkinson')

     for i in spiral_test_healthy:
       image = cv2.imread(fp_spiral_test_healthy+i)
       outputs.append(image)
       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]
       features = quantify_image(image)
       testX.append(features)
       testY.append('healthy')

     for i in spiral_test_park:
       image = cv2.imread(fp_spiral_test_park+i)
       outputs.append(image)
       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]
       features = quantify_image(image)
       testX.append(features)
       testY.append('parkinson')
```



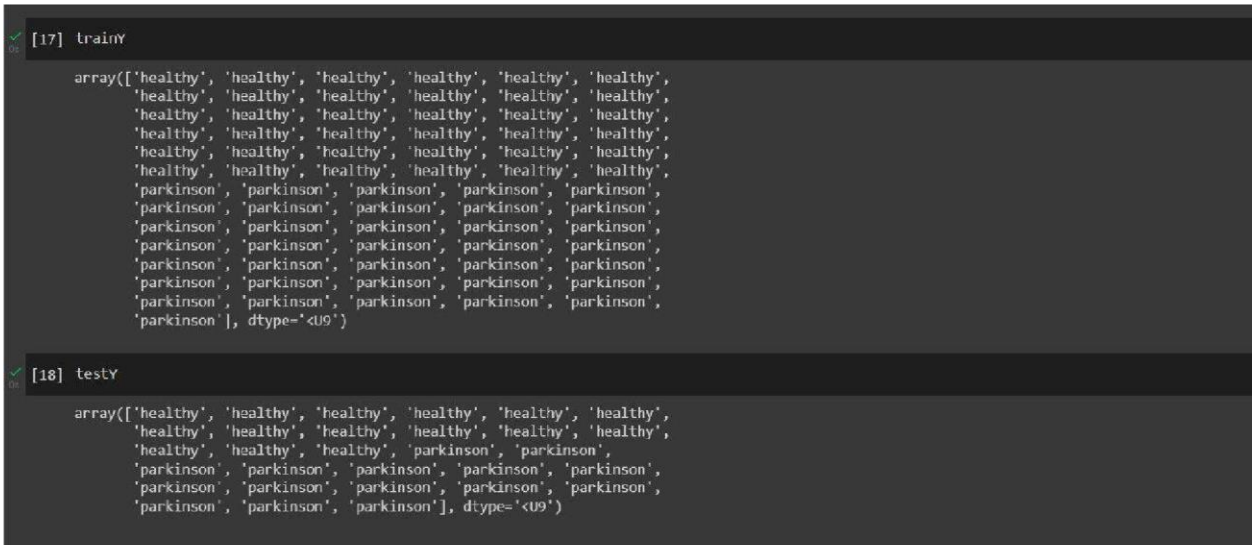
```
[14] trainX = np.array(trainX)
     testX = np.array(testX)
     trainY = np.array(trainY)
     testY = np.array(testY)
```

```
[15] trainX
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

```
[16] testX
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

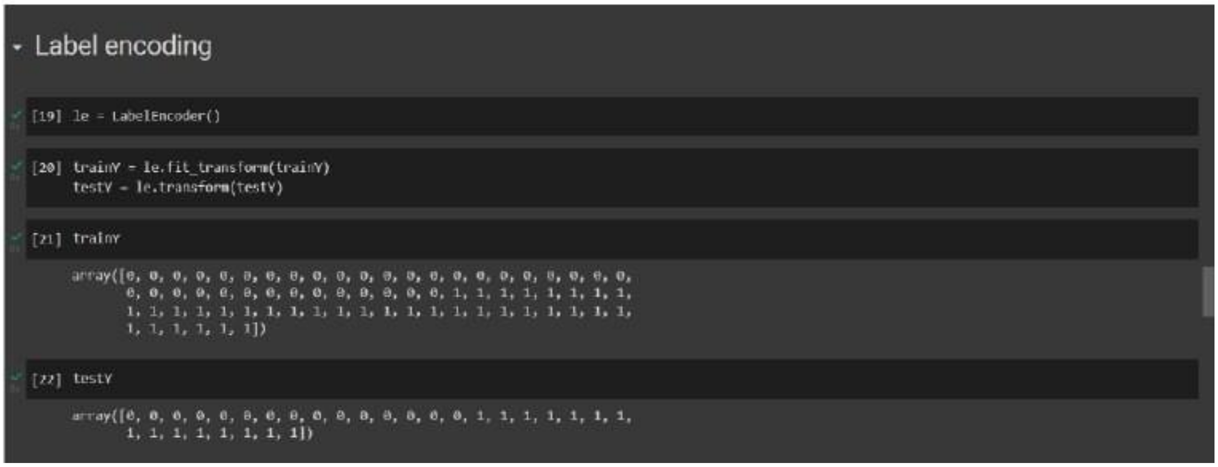


```
[17] trainY
```

```
array(['healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson'], dtype='<U9')
```

```
[18] testY
```

```
array(['healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson'], dtype='<U9')
```



Label encoding

```
[19] le = LabelEncoder()
```

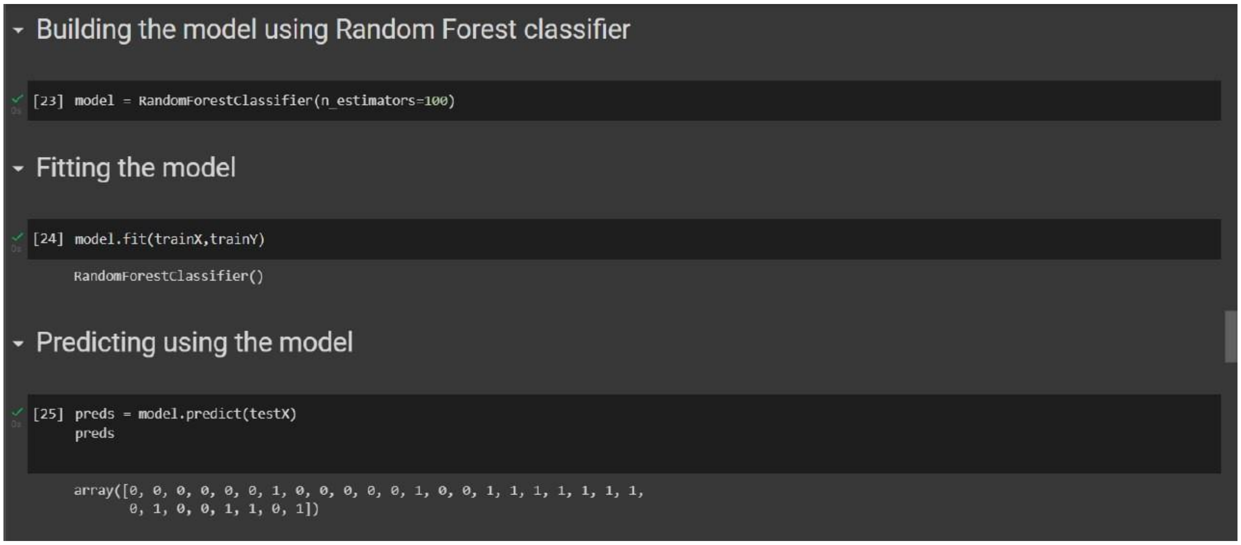
```
[20] trainY = le.fit_transform(trainY)
     testY = le.transform(testY)
```

```
[21] trainY
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1])
```

```
[22] testY
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1])
```



Building the model using Random Forest classifier

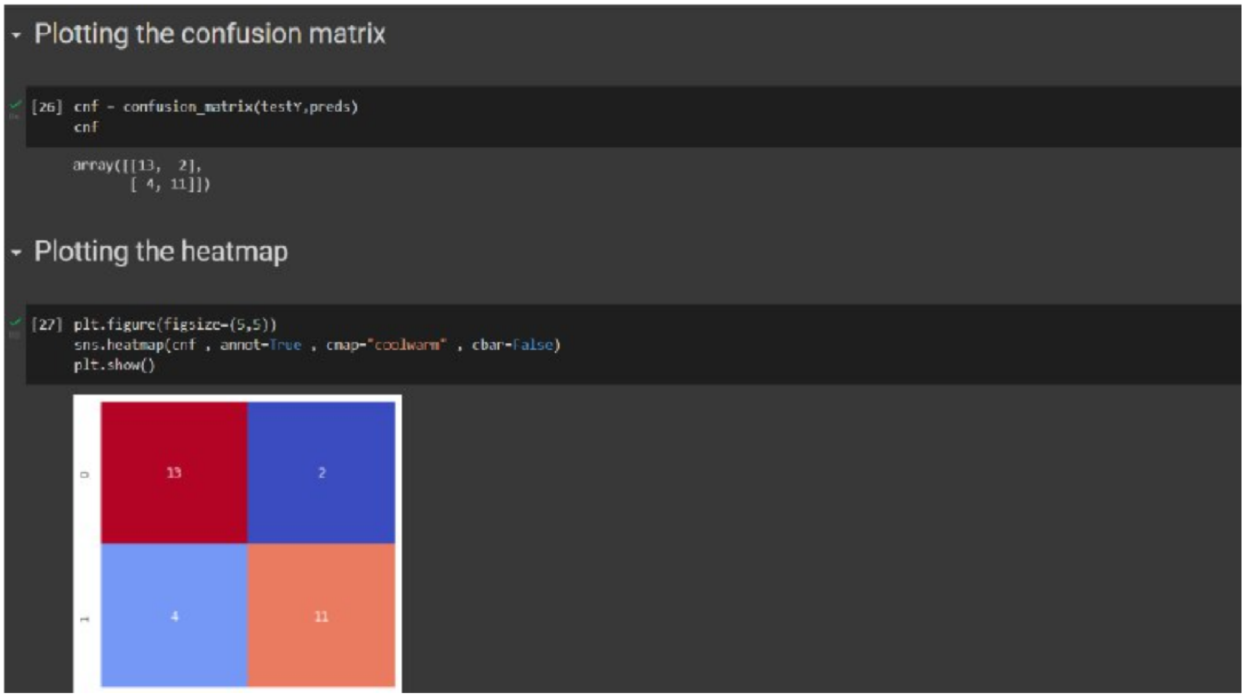
```
[23] model = RandomForestClassifier(n_estimators=100)
```

Fitting the model

```
[24] model.fit(trainX,trainY)

     RandomForestClassifier()
```

Predicting using the model

```
[25] preds = model.predict(testX)
     preds

     array([0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1,
            0, 1, 0, 0, 1, 1, 0, 1])
```

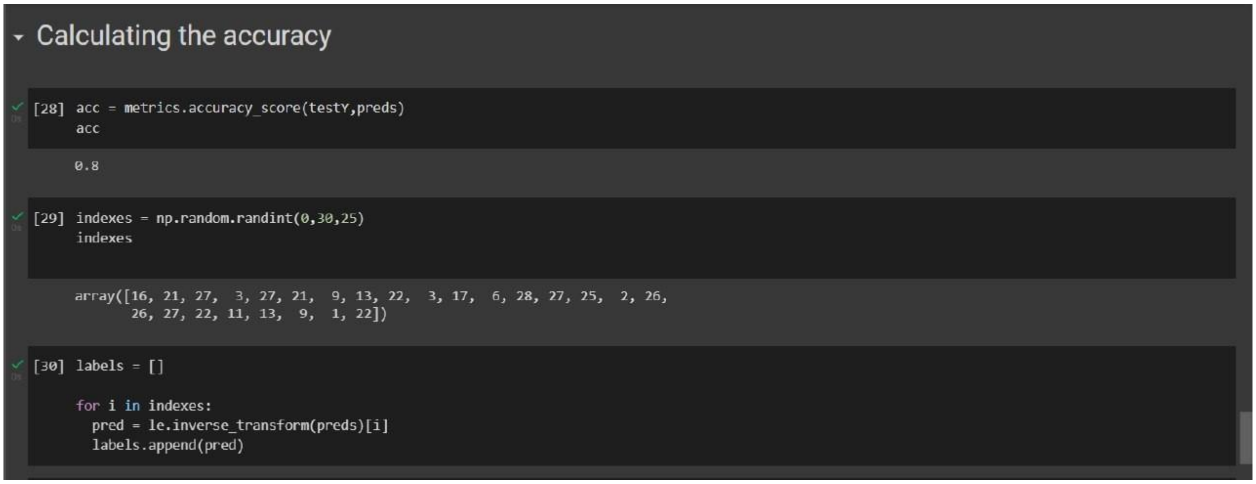


Plotting the confusion matrix

```
[26] cnf = confusion_matrix(testY,preds)
     cnf

     array([[13,  2],
            [ 4, 11]])
```

Plotting the heatmap

```
[27] plt.figure(figsize=(5,5))
     sns.heatmap(cnf , annot=True , cmap="coolwarm" , cbar=False)
     plt.show()
```

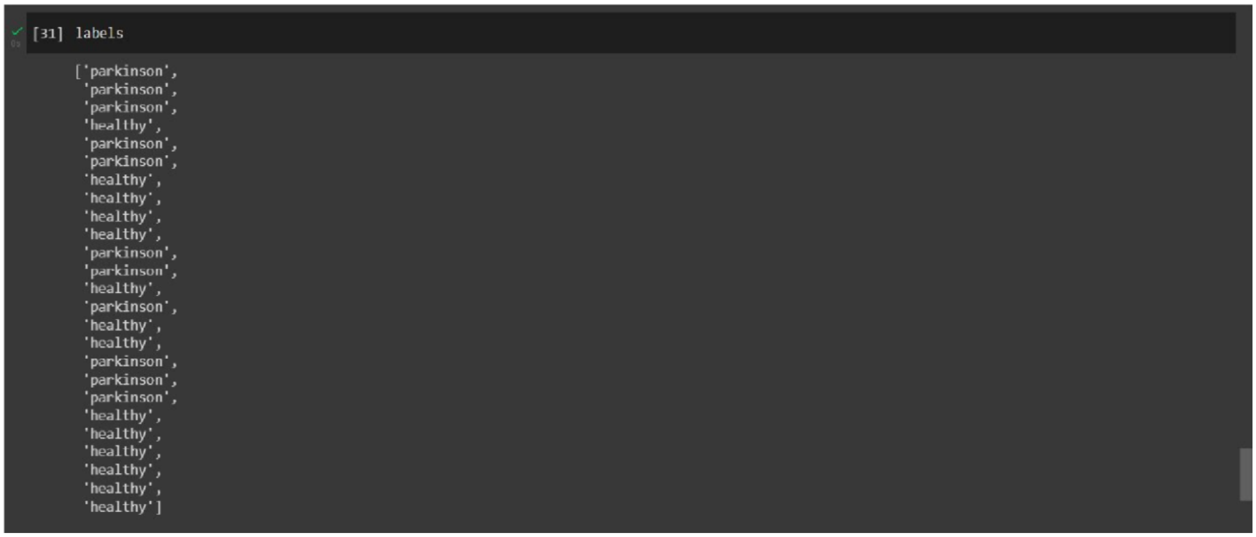


Calculating the accuracy

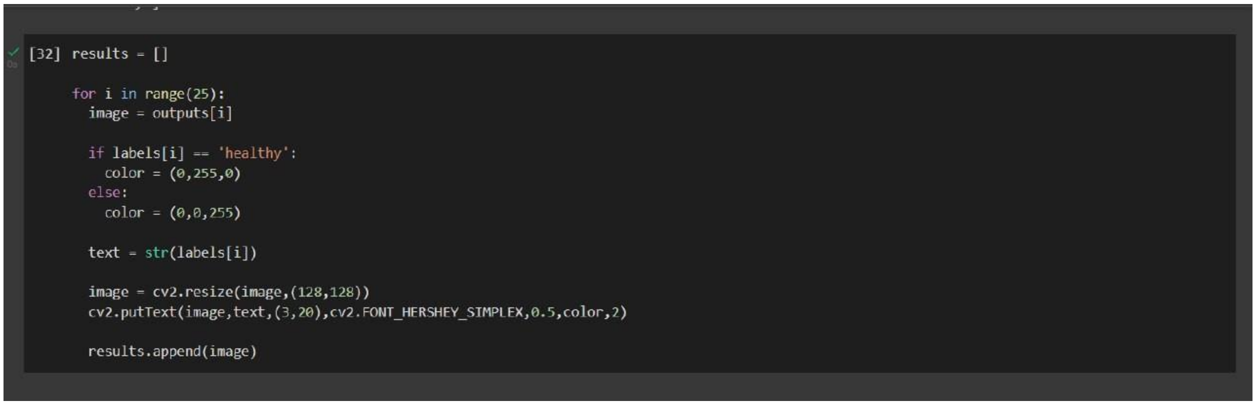
```
[28] acc = metrics.accuracy_score(testY,preds)
     acc

     0.8
```

```
[29] indexes = np.random.randint(0,30,25)
     indexes

     array([16, 21, 27,  3, 27, 21,  9, 13, 22,  3, 17,  6, 28, 27, 25,  2, 26,
            26, 27, 22, 11, 13,  9,  1, 22])
```

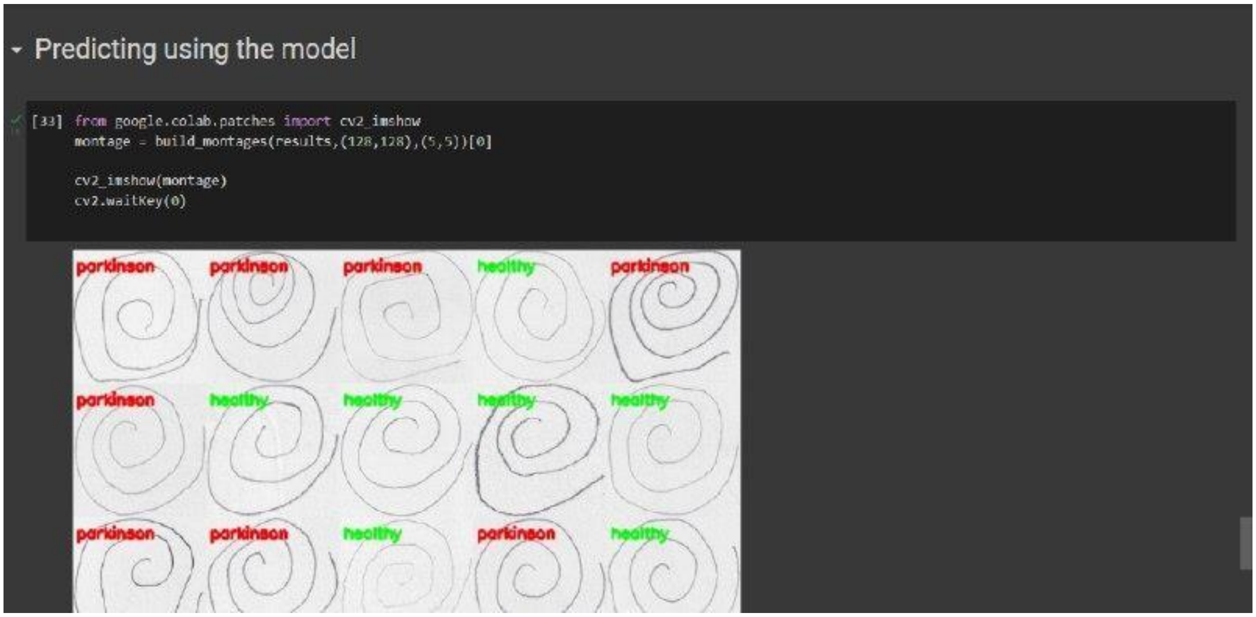
```
[30] labels = []

     for i in indexes:
       pred = le.inverse_transform(preds)[i]
       labels.append(pred)
```



```
[31] labels

     ['parkinson',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'parkinson',
      'healthy',
      'healthy',
      'healthy',
      'healthy',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'healthy',
      'healthy',
      'parkinson',
      'parkinson',
      'parkinson',
      'healthy',
      'healthy',
      'healthy',
      'healthy',
      'healthy',
      'healthy']
```

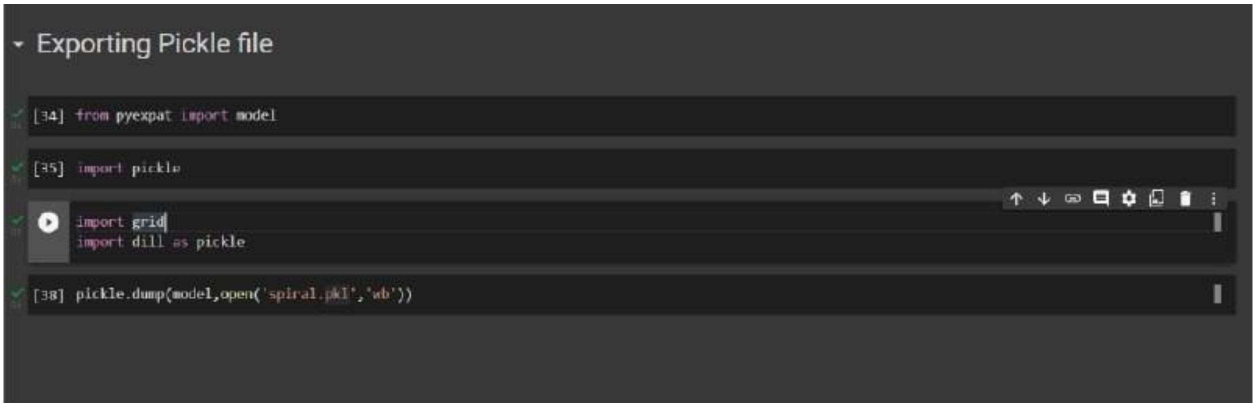


```
[32] results = []

     for i in range(25):
       image = outputs[i]

       if labels[i] == 'healthy':
         color = (0,255,0)
       else:
         color = (0,0,255)

       text = str(labels[i])

       image = cv2.resize(image,(128,128))
       cv2.putText(image,text,(3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)

       results.append(image)
```



Predicting using the model

```
[33] from google.colab.patches import cv2_imshow
     montage = build_montages(results,(128,128),(5,5))[0]

     cv2_imshow(montage)
     cv2.waitKey(0)
```



Exporting Pickle file

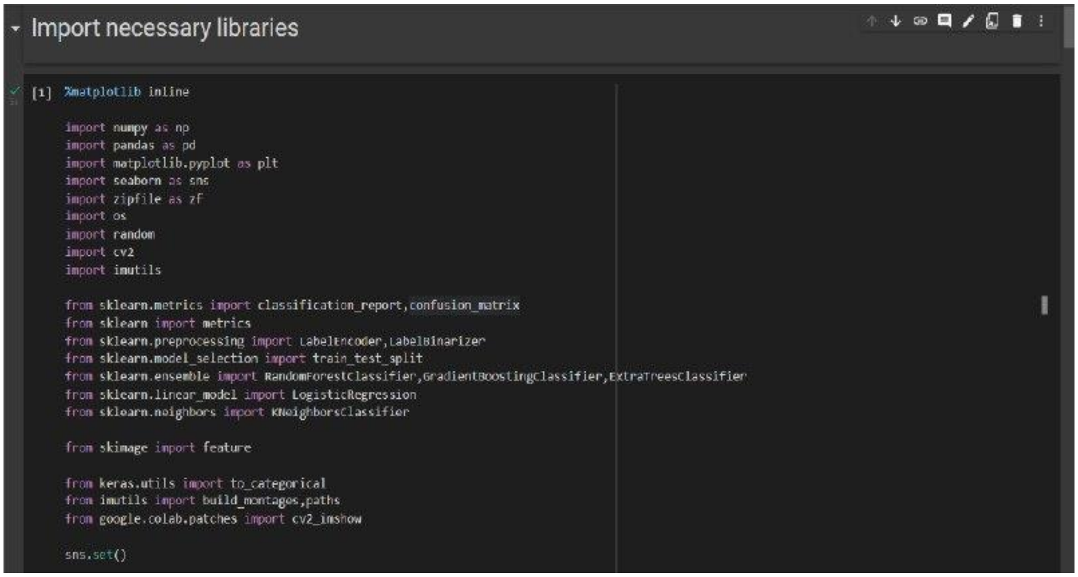
```
[34] from pyexpat import model
```

```
[35] import pickle
```

```
import grid
import dill as pickle
```

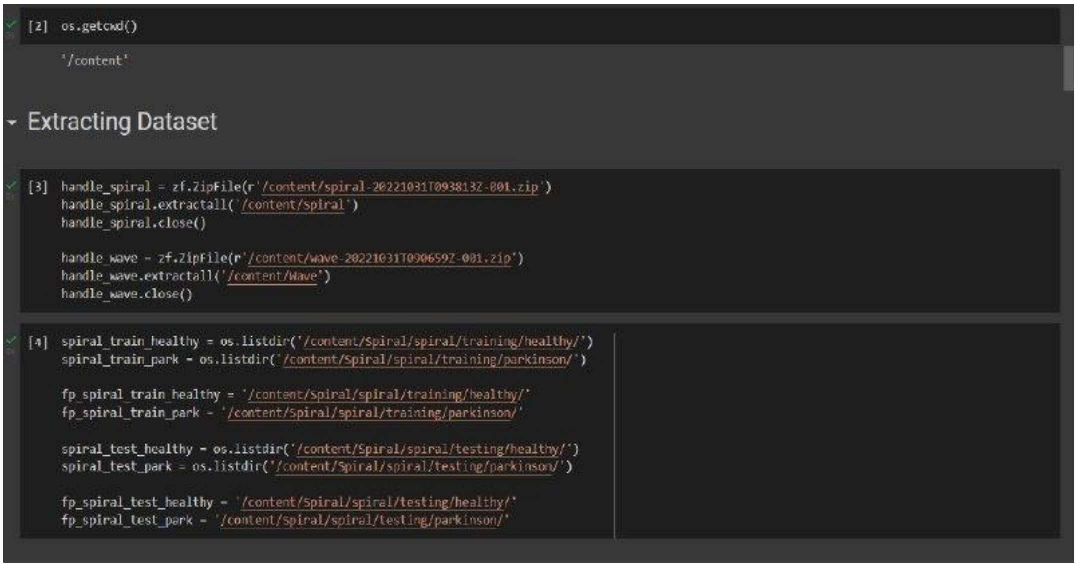
```
[38] pickle.dump(model,open('spiral.pkl','wb'))
```

Training the model for wave image



Import necessary libraries

```
[1] %matplotlib inline

    import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    import zipfile as zf
    import os
    import random
    import cv2
    import imutils

    from sklearn.metrics import classification_report,confusion_matrix
    from sklearn import metrics
    from sklearn.preprocessing import LabelEncoder,LabelBinarizer
    from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier,ExtraTreesClassifier
    from sklearn.linear_model import LogisticRegression
    from sklearn.neighbors import KNeighborsClassifier

    from skimage import feature

    from keras.utils import to_categorical
    from imutils import build_montages,paths
    from google.colab.patches import cv2_imshow

    sns.set()
```

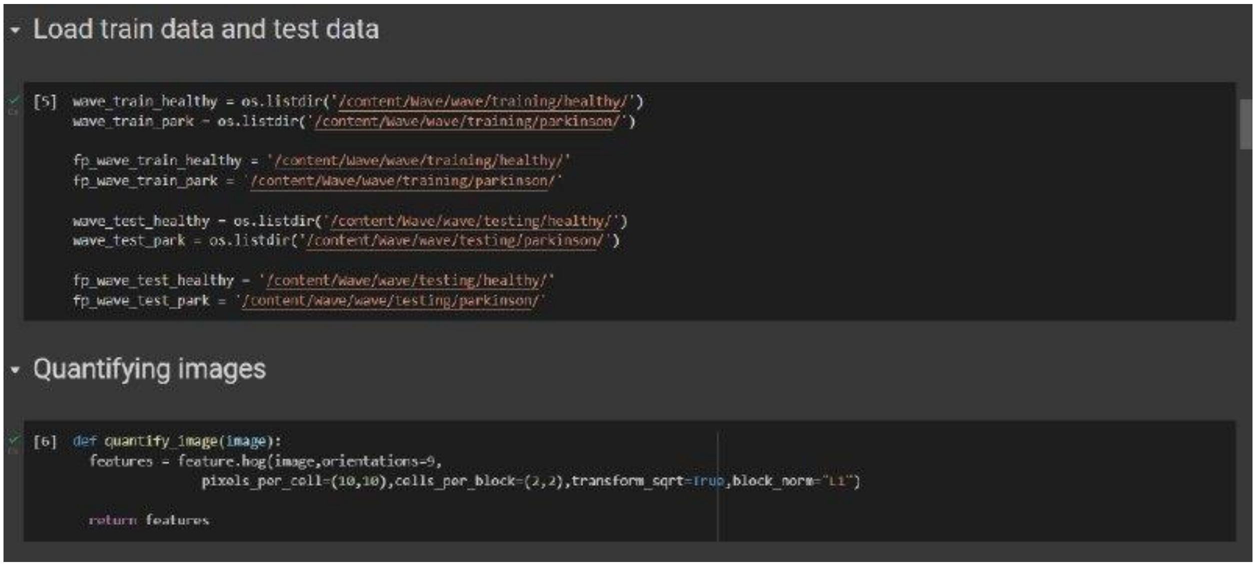


```
[2] os.getcwd()

    '/content'
```

Extracting Dataset

```
[3] handle_spiral = zf.ZipFile(r'/content/spiral-20221031T093813Z-001.zip')
    handle_spiral.extractall('/content/spiral')
    handle_spiral.close()

    handle_wave = zf.ZipFile(r'/content/wave-20221031T090659Z-001.zip')
    handle_wave.extractall('/content/Wave')
    handle_wave.close()
```

```
[4] spiral_train_healthy = os.listdir('/content/Spiral/spiral/training/healthy/')
    spiral_train_park = os.listdir('/content/Spiral/spiral/training/parkinson/')

    fp_spiral_train_healthy = '/content/Spiral/spiral/training/healthy/'
    fp_spiral_train_park = '/content/Spiral/spiral/training/parkinson/'

    spiral_test_healthy = os.listdir('/content/Spiral/spiral/testing/healthy/')
    spiral_test_park = os.listdir('/content/Spiral/spiral/testing/parkinson/')

    fp_spiral_test_healthy = '/content/Spiral/spiral/testing/healthy/'
    fp_spiral_test_park = '/content/Spiral/spiral/testing/parkinson/'
```

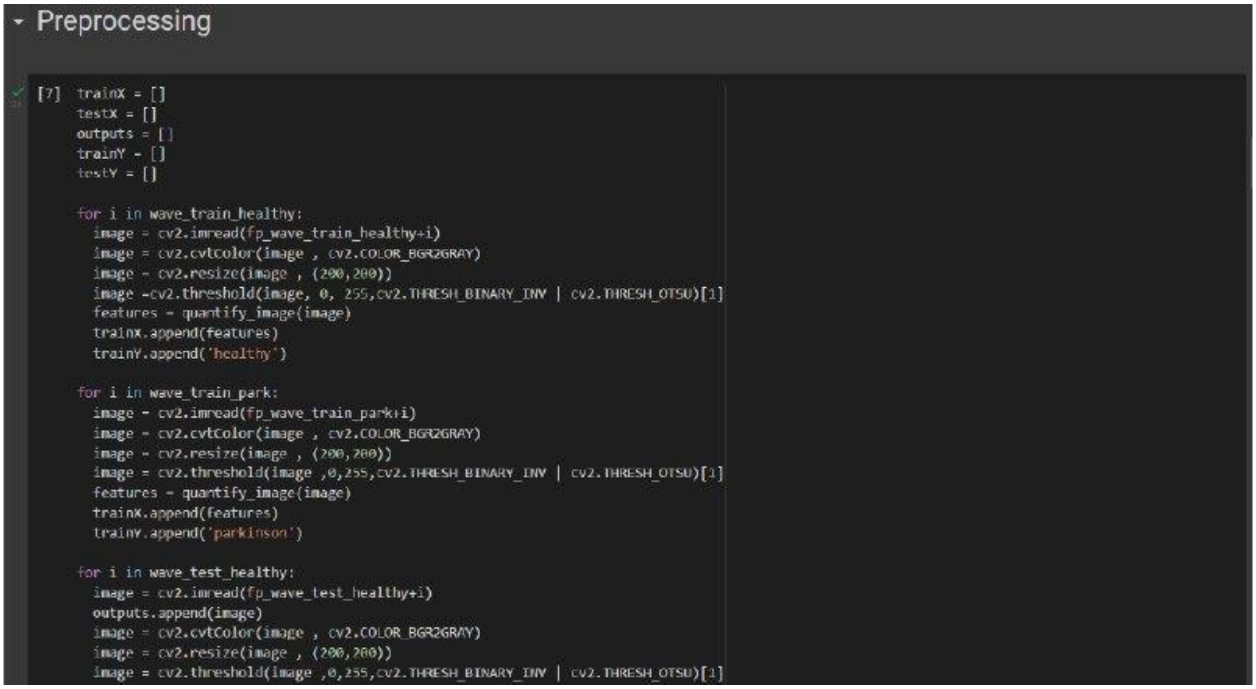


Load train data and test data

```
[5] wave_train_healthy = os.listdir('/content/Wave/wave/training/healthy/')
    wave_train_park = os.listdir('/content/Wave/wave/training/parkinson/')

    fp_wave_train_healthy = '/content/Wave/wave/training/healthy/'
    fp_wave_train_park = '/content/Wave/wave/training/parkinson/'

    wave_test_healthy = os.listdir('/content/Wave/wave/testing/healthy/')
    wave_test_park = os.listdir('/content/Wave/wave/testing/parkinson/')

    fp_wave_test_healthy = '/content/Wave/wave/testing/healthy/'
    fp_wave_test_park = '/content/Wave/wave/testing/parkinson/'
```

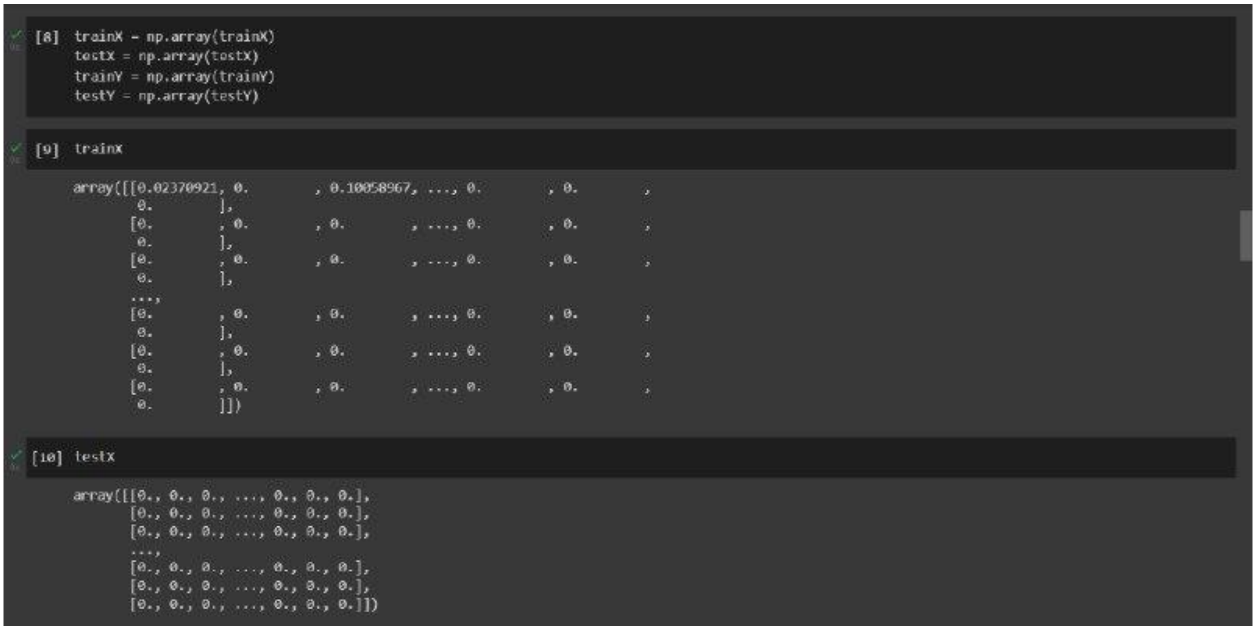
Quantifying images

```
[6] def quantify_image(image):
      features = feature.hog(image,orientations=9,
                    pixels_per_cell=(10,10),cells_per_block=(2,2),transform_sqrt=True,block_norm="L1")

      return features
```



Preprocessing

```
[7] trainX = []
    testX = []
    outputs = []
    trainY = []
    testY = []

    for i in wave_train_healthy:
      image = cv2.imread(fp_wave_train_healthy+i)
      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]
      features = quantify_image(image)
      trainX.append(features)
      trainY.append('healthy')

    for i in wave_train_park:
      image = cv2.imread(fp_wave_train_park+i)
      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]
      features = quantify_image(image)
      trainX.append(features)
      trainY.append('parkinson')

    for i in wave_test_healthy:
      image = cv2.imread(fp_wave_test_healthy+i)
      outputs.append(image)
      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]
```



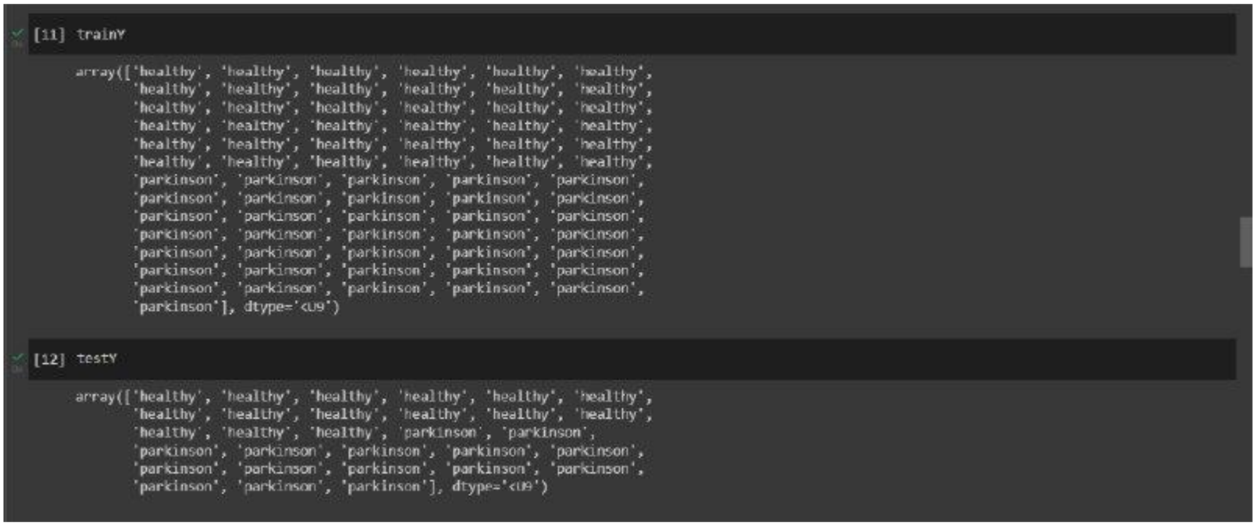
```
[8]  trainX = np.array(trainX)
     testX = np.array(testX)
     trainY = np.array(trainY)
     testY = np.array(testY)
```

```
[9]  trainX
```

```
array([[0.02370921, 0.        , 0.10058967, ..., 0.        , 0.        ,
         0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
         0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
         0.        ],
       ...,
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
         0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
         0.        ],
       [0.        , 0.        , 0.        , ..., 0.        , 0.        ,
         0.        ]])
```

```
[10] testX
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

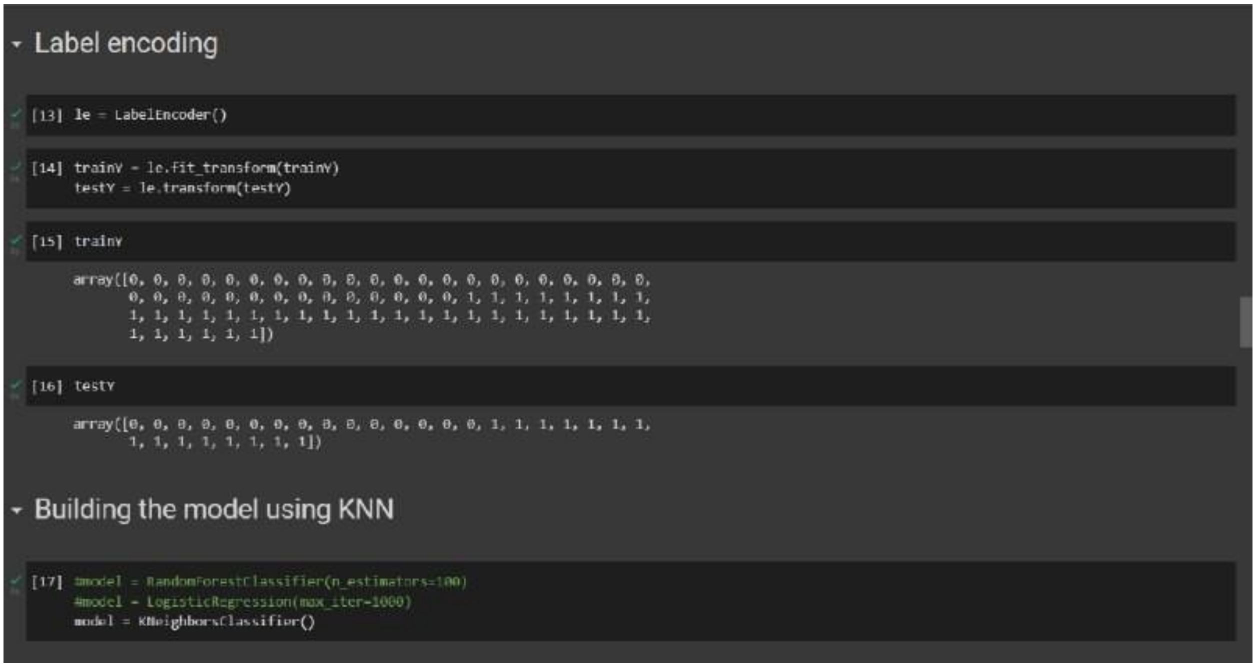


```
[11] trainY
```

```
array(['healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson'], dtype='<U9')
```

```
[12] testY
```

```
array(['healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'healthy', 'healthy', 'healthy',
       'healthy', 'healthy', 'healthy', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson', 'parkinson', 'parkinson',
       'parkinson', 'parkinson', 'parkinson'], dtype='<U9')
```



Label encoding

```
[13] le = LabelEncoder()
```

```
[14] trainY = le.fit_transform(trainY)
     testY = le.transform(testY)
```

```
[15] trainY
```

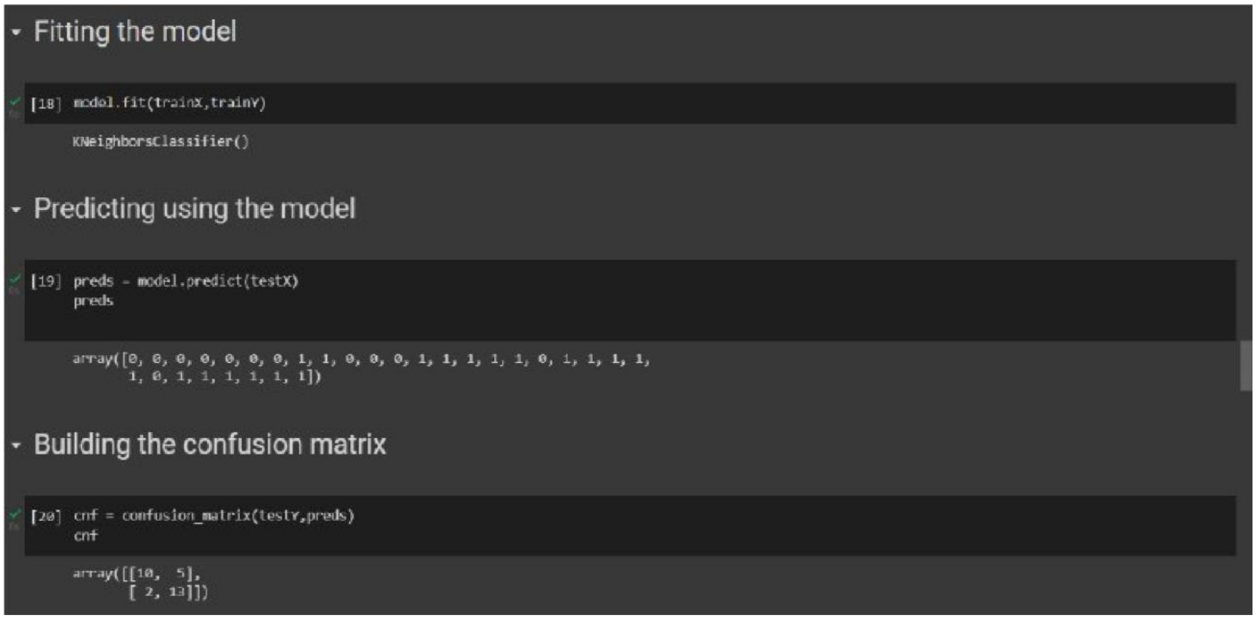
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1])
```

```
[16] testY
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1])
```

Building the model using KNN

```
[17] #model = RandomForestClassifier(n_estimators=100)
     #model = LogisticRegression(max_iter=1000)
     model = KNeighborsClassifier()
```



Fitting the model

```
[18] model.fit(trainX,trainY)
```

```
KNeighborsClassifier()
```

Predicting using the model

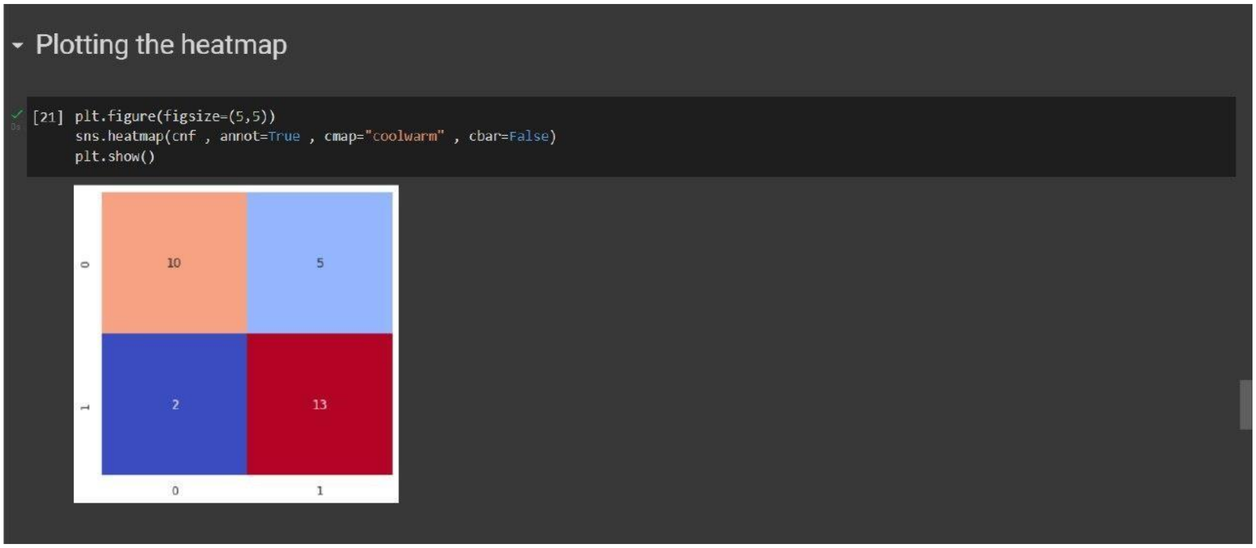
```
[19] preds = model.predict(testX)
     preds
```

```
array([0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 1])
```

Building the confusion matrix

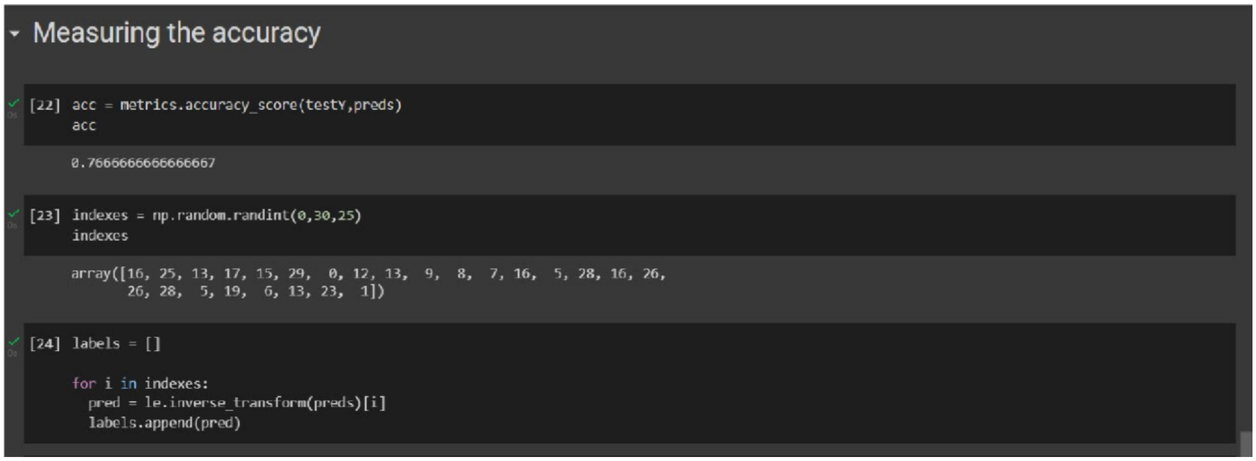
```
[20] cnf = confusion_matrix(testY,preds)
     cnf
```

```
array([[10,  5],
       [ 2, 13]])
```



Plotting the heatmap

```
[21] plt.figure(figsize=(5,5))
     sns.heatmap(cnf , annot=True , cmap="coolwarm" , cbar=False)
     plt.show()
```

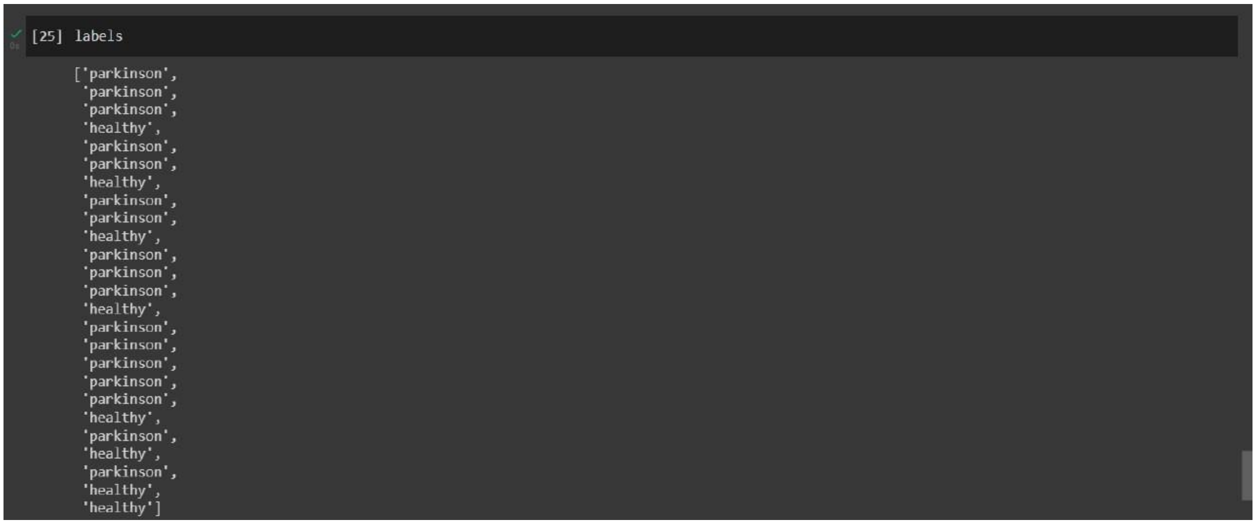


Measuring the accuracy

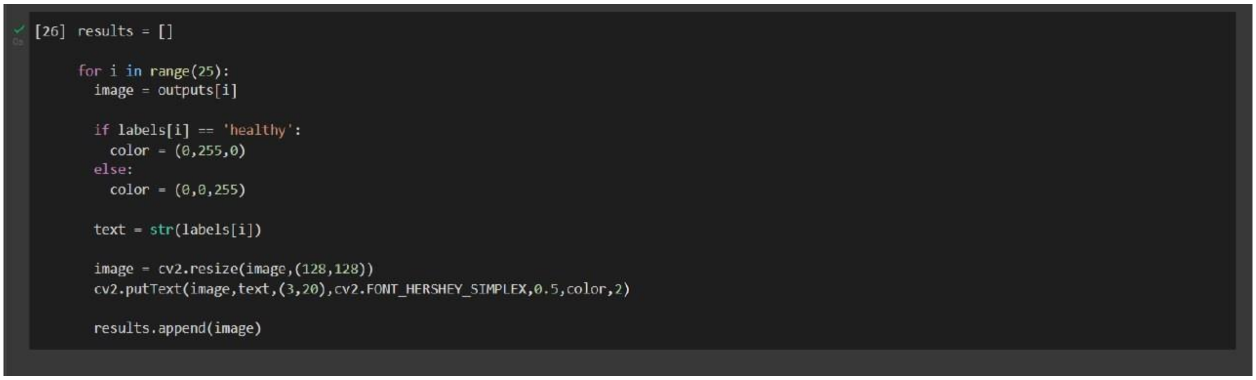
```
[22] acc = metrics.accuracy_score(testY,preds)
     acc

     0.7666666666666667
```

```
[23] indexes = np.random.randint(0,30,25)
     indexes

     array([16, 25, 13, 17, 15, 29,  0, 12, 13,  9,  8,  7, 16,  5, 28, 16, 26,
            26, 28,  5, 19,  6, 13, 23,  1])
```

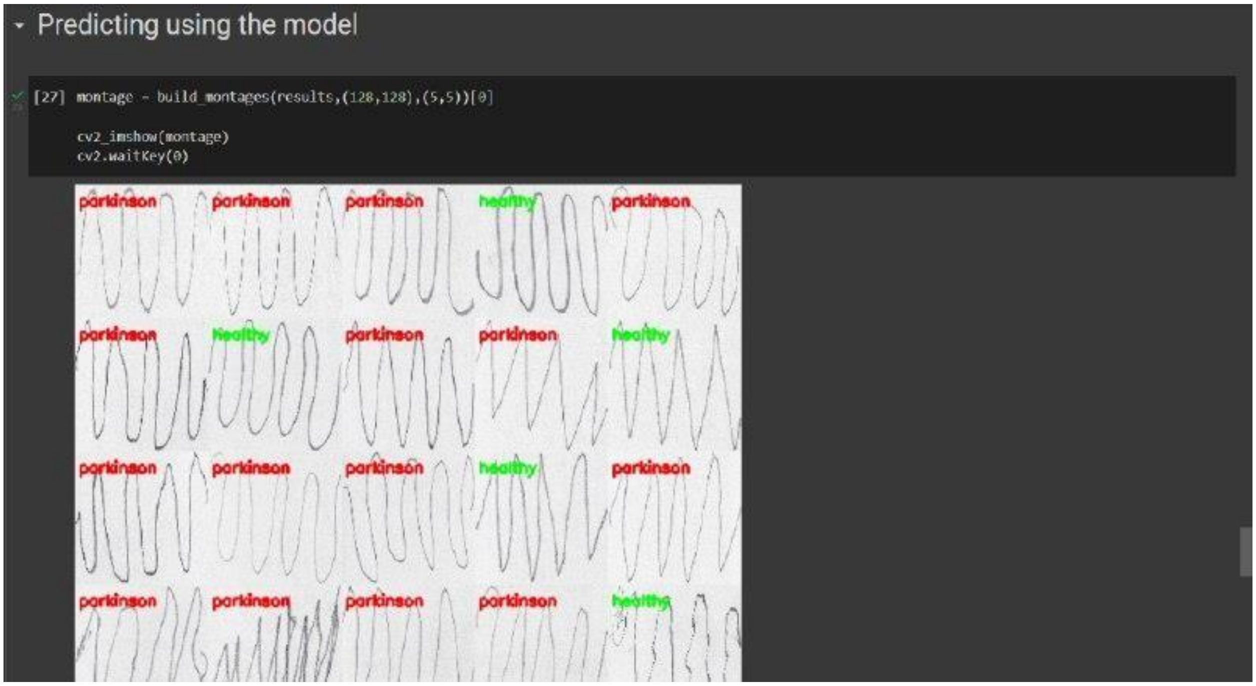
```
[24] labels = []

     for i in indexes:
       pred = le.inverse_transform(preds)[i]
       labels.append(pred)
```



```
[25] labels

     ['parkinson',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'parkinson',
      'parkinson',
      'parkinson',
      'parkinson',
      'healthy',
      'parkinson',
      'healthy',
      'parkinson',
      'healthy',
      'healthy']
```

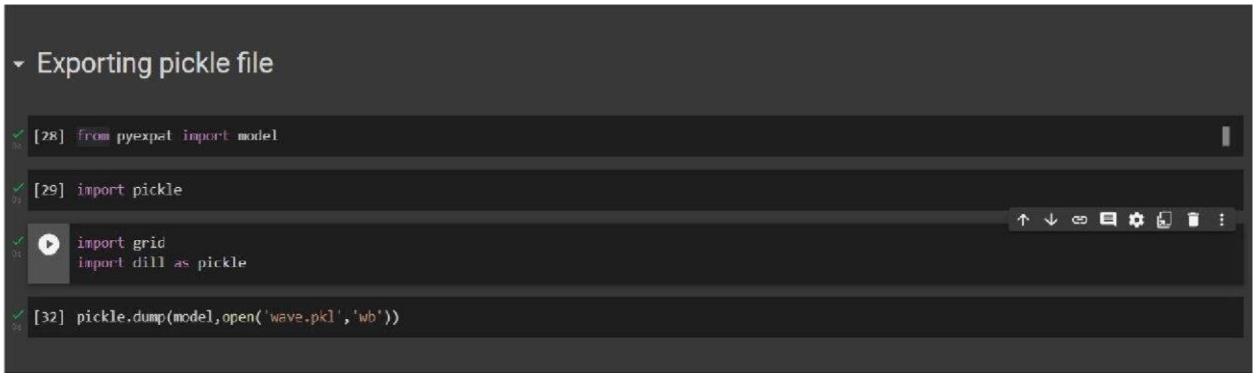


```
[26] results = []

     for i in range(25):
       image = outputs[i]

       if labels[i] == 'healthy':
         color = (0,255,0)
       else:
         color = (0,0,255)

       text = str(labels[i])

       image = cv2.resize(image,(128,128))
       cv2.putText(image,text,(3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)

       results.append(image)
```



Predicting using the model

```
[27] montage = build_montages(results,(128,128),(5,5))[0]

     cv2_imshow(montage)
     cv2.waitKey(0)
```



Exporting pickle file

```
[28] from pyexpat import model
```

```
[29] import pickle
```

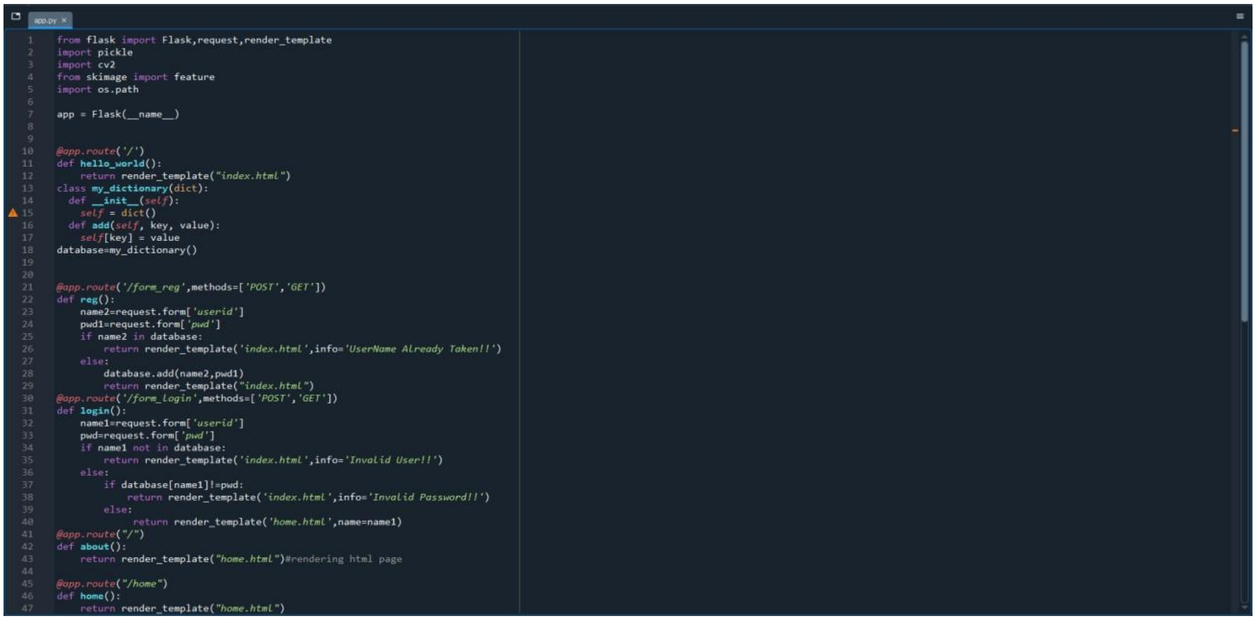
```
import grid
import dill as pickle
```

```
[32] pickle.dump(model,open('wave.pkl','wb'))
```

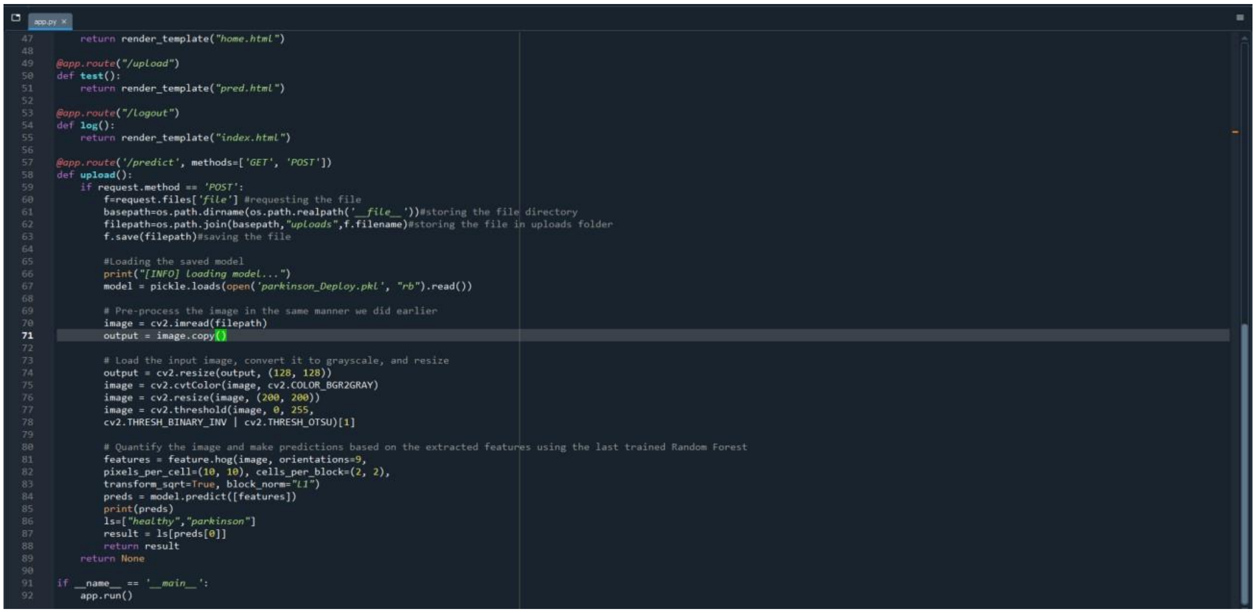
7.2 Feature 2

Our application has the following pages: Login/ Register page, Home page and Predict page.

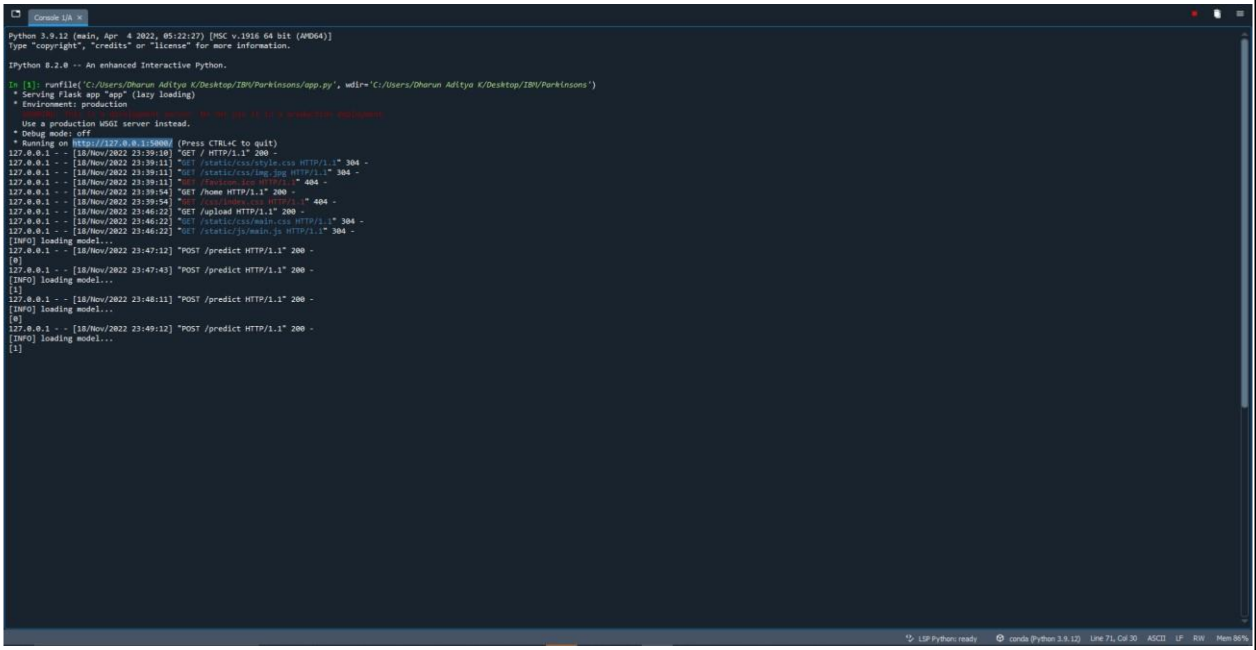
The user registers and consequently signs in using the login/sign up page. The Home page contains information about Parkinson’s Disease as well as the diagnostic method being used to test for it. The next page is the predict page where the user is required to upload the images of spiral and wave following which they get a diagnosis as healthy / having Parkinson’s Disease.



```python
from flask import Flask,request,render_template
import pickle
import cv2
from skimage import feature
import os.path

app = Flask(__name__)


@app.route('/')
def hello_world():
    return render_template("index.html")
class my_dictionary(dict):
  def __init__(self):
    self = dict()
  def add(self, key, value):
    self[key] = value
database=my_dictionary()


@app.route('/form_reg',methods=['POST','GET'])
def reg():
    name2=request.form['userid']
    pwd1=request.form['pwd']
    if name2 in database:
        return render_template('index.html',info='UserName Already Taken!!')
    else:
        database.add(name2,pwd1)
        return render_template("index.html")
@app.route('/form_login',methods=['POST','GET'])
def login():
    name1=request.form['userid']
    pwd=request.form['pwd']
    if name1 not in database:
        return render_template('index.html',info='Invalid User!!')
    else:
        if database[name1]!=pwd:
            return render_template('index.html',info='Invalid Password!!')
        else:
             return render_template('home.html',name=name1)
@app.route("/")
def about():
    return render_template("home.html")#rendering html page

@app.route("/home")
def home():
    return render_template("home.html")
```

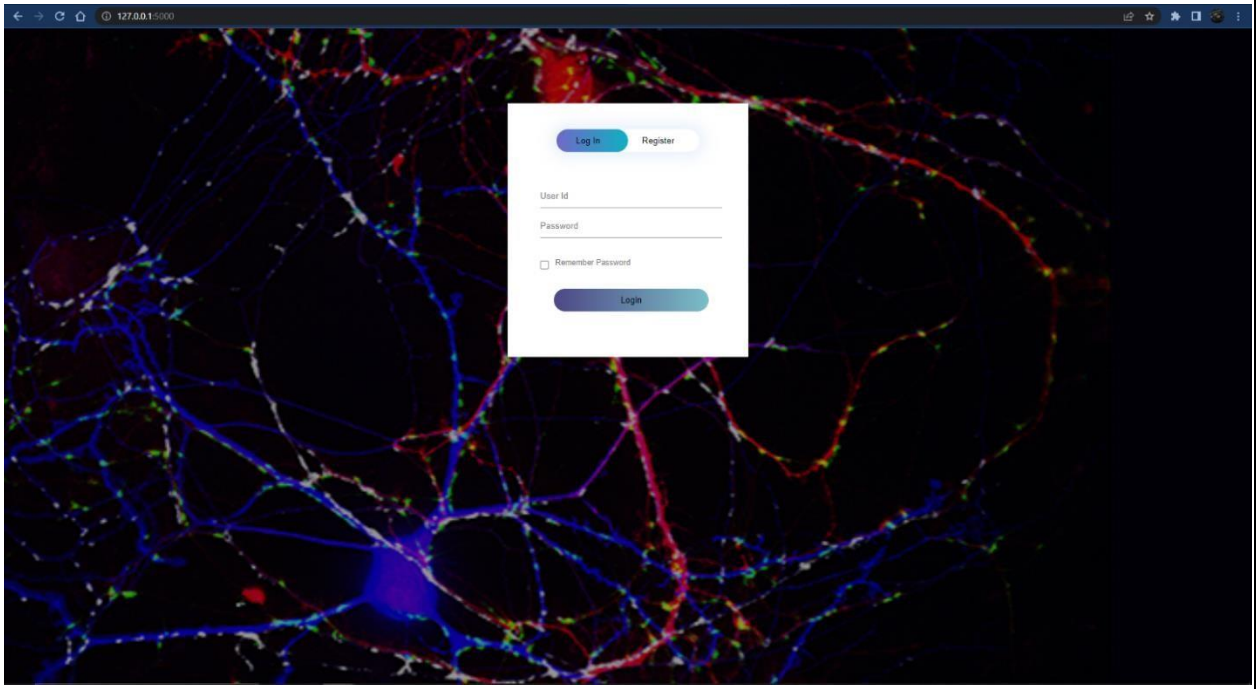


```python
    return render_template("home.html")

@app.route("/upload")
def test():
    return render_template("pred.html")

@app.route("/logout")
def log():
    return render_template("index.html")

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname(os.path.realpath('__file__'))#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        #Loading the saved model
        print("[INFO] loading model...")
        model = pickle.loads(open('parkinson_Deploy.pkl', "rb").read())

        # Pre-process the image in the same manner we did earlier
        image = cv2.imread(filepath)
        output = image.copy()

        # Load the input image, convert it to grayscale, and resize
        output = cv2.resize(output, (128, 128))
        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]

        # Quantify the image and make predictions based on the extracted features using the last trained Random Forest
        features = feature.hog(image, orientations=9,
        pixels_per_cell=(10, 10), cells_per_block=(2, 2),
        transform_sqrt=True, block_norm="L1")
        preds = model.predict([features])
        print(preds)
        ls=["healthy","parkinson"]
        result = ls[preds[0]]
        return result
    return None

if __name__ == '__main__':
    app.run()
```

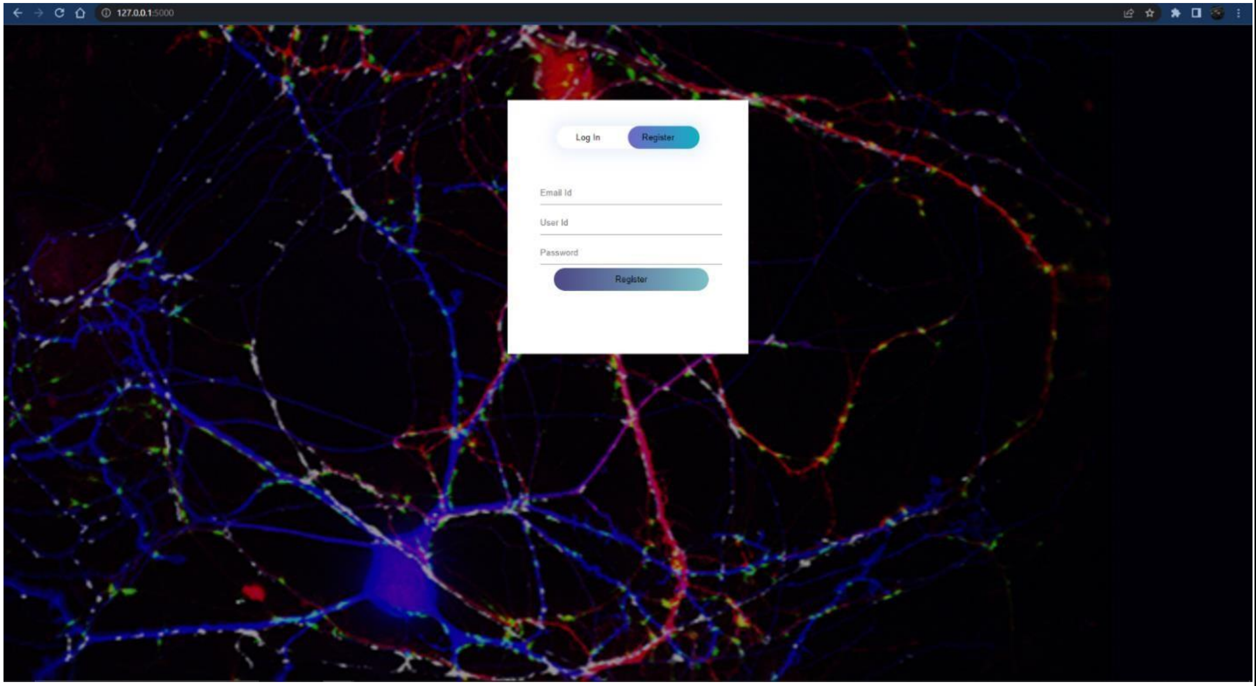


```
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Dharun Aditya K/Desktop/IBM/Parkinsons/app.py', wdir='C:/Users/Dharun Aditya K/Desktop/IBM/Parkinsons')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [18/Nov/2022 23:39:10] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Nov/2022 23:39:11] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [18/Nov/2022 23:39:11] "GET /static/css/img.jpg HTTP/1.1" 304 -
127.0.0.1 - - [18/Nov/2022 23:39:11] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [18/Nov/2022 23:39:54] "GET /home HTTP/1.1" 200 -
127.0.0.1 - - [18/Nov/2022 23:39:54] "GET /css/index.css HTTP/1.1" 404 -
127.0.0.1 - - [18/Nov/2022 23:46:22] "GET /upload HTTP/1.1" 200 -
127.0.0.1 - - [18/Nov/2022 23:46:22] "GET /static/css/main.css HTTP/1.1" 304 -
127.0.0.1 - - [18/Nov/2022 23:46:22] "GET /static/js/main.js HTTP/1.1" 304 -
[INFO] loading model...
127.0.0.1 - - [18/Nov/2022 23:47:12] "POST /predict HTTP/1.1" 200 -
[0]
127.0.0.1 - - [18/Nov/2022 23:47:43] "POST /predict HTTP/1.1" 200 -
[INFO] loading model...
[1]
127.0.0.1 - - [18/Nov/2022 23:48:11] "POST /predict HTTP/1.1" 200 -
[INFO] loading model...
[0]
127.0.0.1 - - [18/Nov/2022 23:49:12] "POST /predict HTTP/1.1" 200 -
[INFO] loading model...
[1]
```

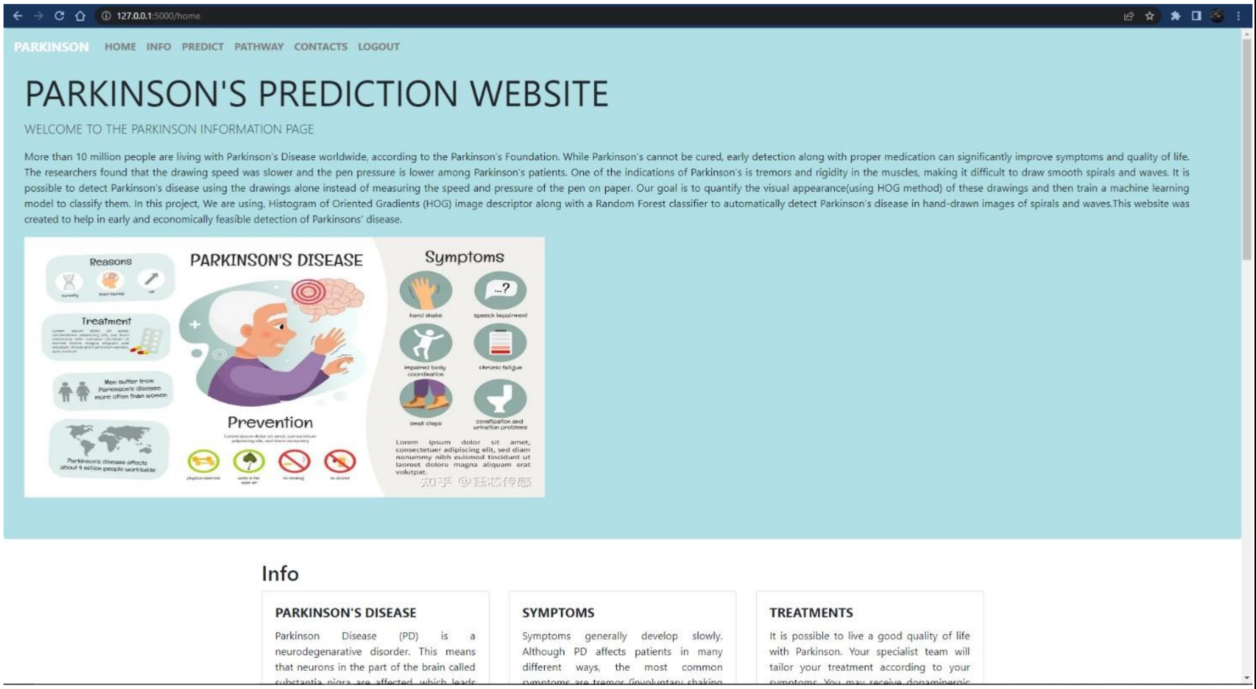
Log In Page:



Register Page:



Home page:


127.0.0.1:5000/home
PARKINSON
HOME
INFO
PREDICT
PATHWAY
CONTACTS
LOGOUT
Reasons
Treatment
Men suffer from Parkinson's disease more often than women
Parkinson's disease affects about 4 million people worldwide
PARKINSON'S DISEASE
Prevention
Symptoms
hand shake
speech impairment
impaired body coordination
chronic fatigue
small steps
constipation and urination problems
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
知乎 @钰芯传感

PARKINSON'S PREDICTION WEBSITE

WELCOME TO THE PARKINSON INFORMATION PAGE

More than 10 million people are living with Parkinson's Disease worldwide, according to the Parkinson's Foundation. While Parkinson's cannot be cured, early detection along with proper medication can significantly improve symptoms and quality of life. The researchers found that the drawing speed was slower and the pen pressure is lower among Parkinson's patients. One of the indications of Parkinson's is tremors and rigidity in the muscles, making it difficult to draw smooth spirals and waves. It is possible to detect Parkinson's disease using the drawings alone instead of measuring the speed and pressure of the pen on paper. Our goal is to quantify the visual appearance(using HOG method) of these drawings and then train a machine learning model to classify them. In this project, We are using, Histogram of Oriented Gradients (HOG) image descriptor along with a Random Forest classifier to automatically detect Parkinson's disease in hand-drawn images of spirals and waves.This website was created to help in early and economically feasible detection of Parkinsons' disease.

Info

PARKINSON'S DISEASE

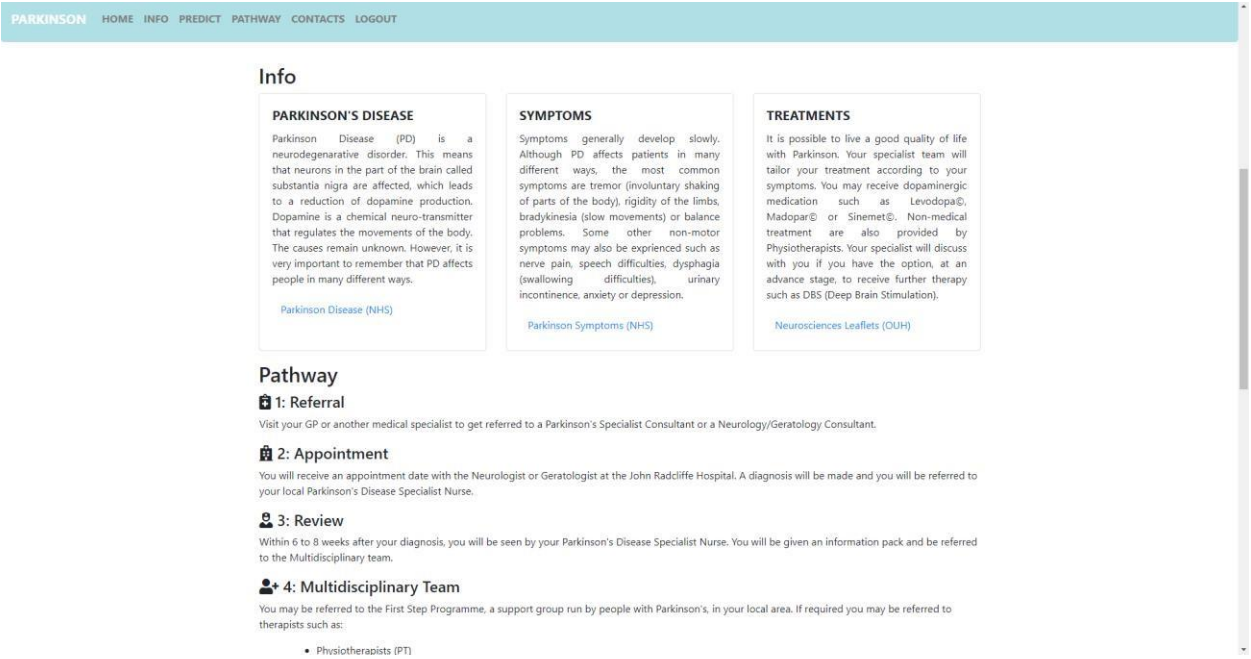
Parkinson Disease (PD) is a neurodegenarative disorder. This means that neurons in the part of the brain called substantia nigra are affected, which leads

SYMPTOMS

Symptoms generally develop slowly. Although PD affects patients in many different ways, the most common symptoms are tremor (involuntary shaking

TREATMENTS

It is possible to live a good quality of life with Parkinson. Your specialist team will tailor your treatment according to your symptoms. You may receive dopaminergic



Info

PARKINSON'S DISEASE

Parkinson Disease (PD) is a neurodegenarative disorder. This means that neurons in the part of the brain called substantia nigra are affected, which leads to a reduction of dopamine production. Dopamine is a chemical neuro-transmitter that regulates the movements of the body. The causes remain unknown. However, it is very important to remember that PD affects people in many different ways.

Parkinson Disease (NHS)

SYMPTOMS

Symptoms generally develop slowly. Although PD affects patients in many different ways, the most common symptoms are tremor (involuntary shaking of parts of the body), rigidity of the limbs, bradykinesia (slow movements) or balance problems. Some other non-motor symptoms may also be exprienced such as nerve pain, speech difficulties, dysphagia (swallowing difficulties), urinary incontinence, anxiety or depression.

Parkinson Symptoms (NHS)

TREATMENTS

It is possible to live a good quality of life with Parkinson. Your specialist team will tailor your treatment according to your symptoms. You may receive dopaminergic medication such as Levodopa©, Madopar© or Sinemet©. Non-medical treatment are also provided by Physiotherapists. Your specialist will discuss with you if you have the option, at an advance stage, to receive further therapy such as DBS (Deep Brain Stimulation).

Neurosciences Leaflets (OUH)

Pathway

1: Referral

Visit your GP or another medical specialist to get referred to a Parkinson's Specialist Consultant or a Neurology/Geratology Consultant.

2: Appointment

You will receive an appointment date with the Neurologist or Geratologist at the John Radcliffe Hospital. A diagnosis will be made and you will be referred to your local Parkinson's Disease Specialist Nurse.

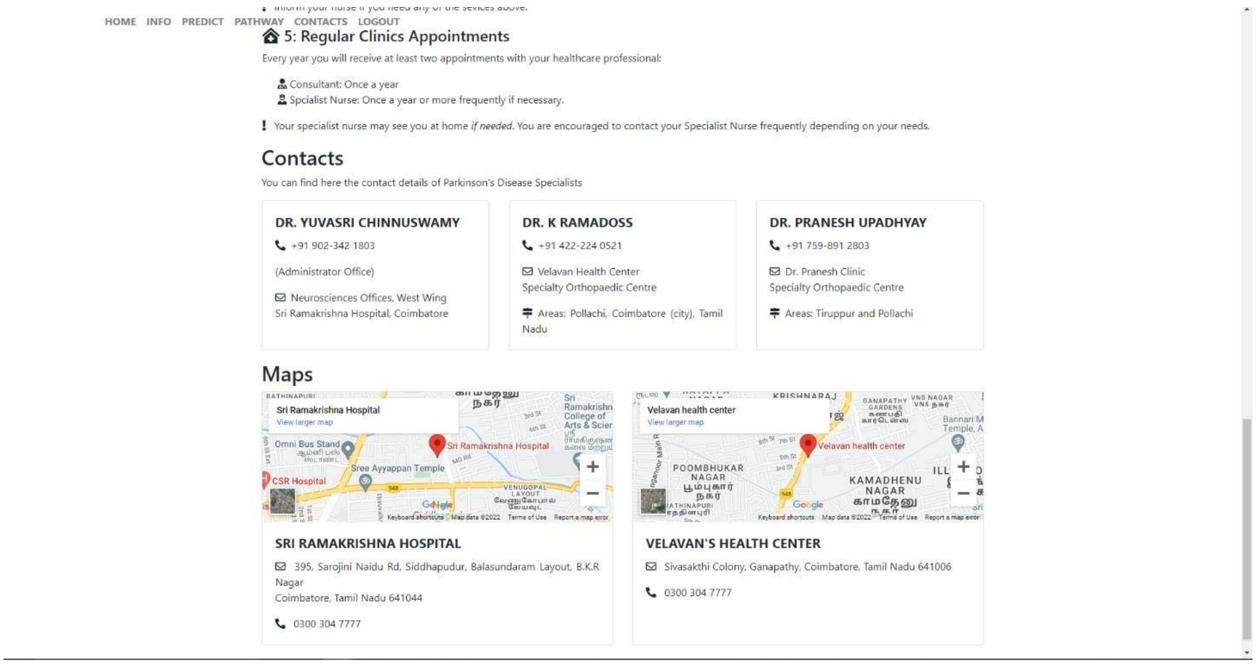
3: Review

Within 6 to 8 weeks after your diagnosis, you will be seen by your Parkinson's Disease Specialist Nurse. You will be given an information pack and be referred to the Multidisciplinary team.

4: Multidisciplinary Team

You may be referred to the First Step Programme, a support group run by people with Parkinson's, in your local area. If required you may be referred to therapists such as:

- Physiotherapists (PT)



! Inform your nurse if you need any of the sevices above.

5: Regular Clinics Appointments

Every year you will receive at least two appointments with your healthcare professional:

Consultant: Once a year
Spcialist Nurse: Once a year or more frequently if necessary.

! Your specialist nurse may see you at home *if needed*. You are encouraged to contact your Specialist Nurse frequently depending on your needs.

Contacts

You can find here the contact details of Parkinson's Disease Specialists

DR. YUVASRI CHINNUSWAMY

☎ +91 902-342 1803

(Administrator Office)

✉ Neurosciences Offices, West Wing
Sri Ramakrishna Hospital, Coimbatore

DR. K RAMADOSS

☎ +91 422-224 0521

✉ Velavan Health Center
Specialty Orthopaedic Centre

Areas: Pollachi, Coimbatore (city), Tamil Nadu

DR. PRANESH UPADHYAY

☎ +91 759-891 2803

✉ Dr. Pranesh Clinic
Specialty Orthopaedic Centre

Areas: Tiruppur and Pollachi

Maps

SRI RAMAKRISHNA HOSPITAL

✉ 395, Sarojini Naidu Rd, Siddhapudur, Balasundaram Layout, B.K.R Nagar
Coimbatore, Tamil Nadu 641044

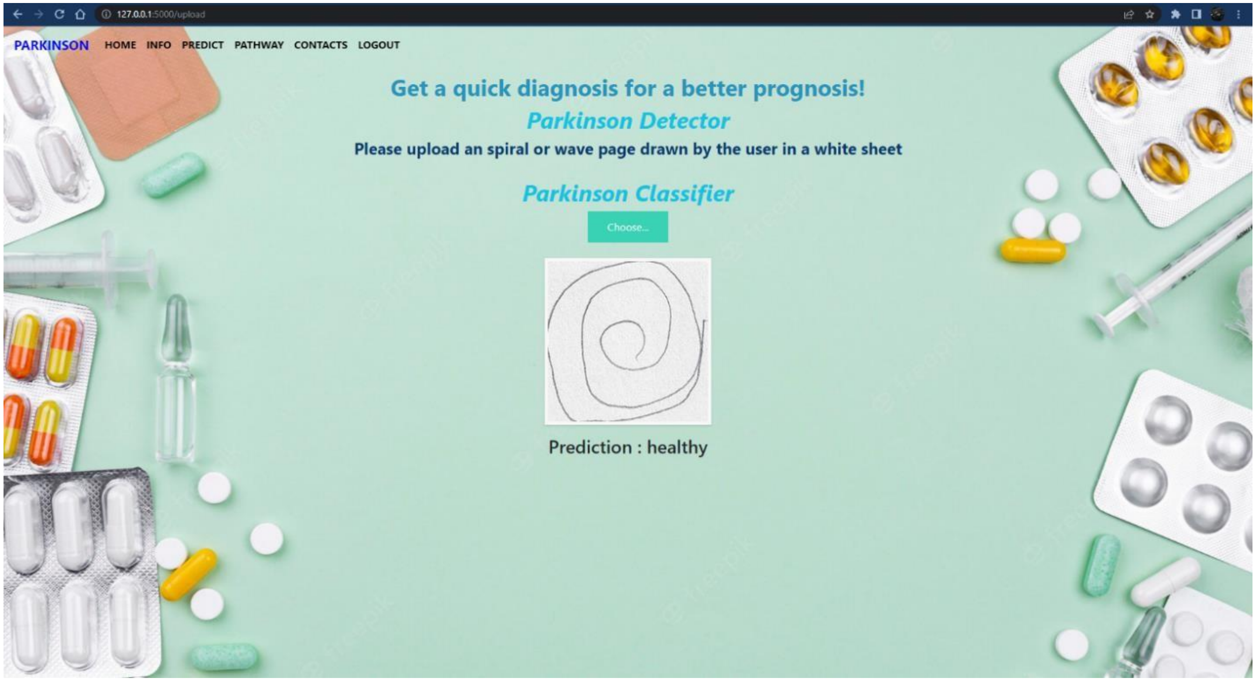
☎ 0300 304 7777

VELAVAN'S HEALTH CENTER

✉ Sivasakthi Colony, Ganapathy, Coimbatore, Tamil Nadu 641006

☎ 0300 304 7777

Prediction page:



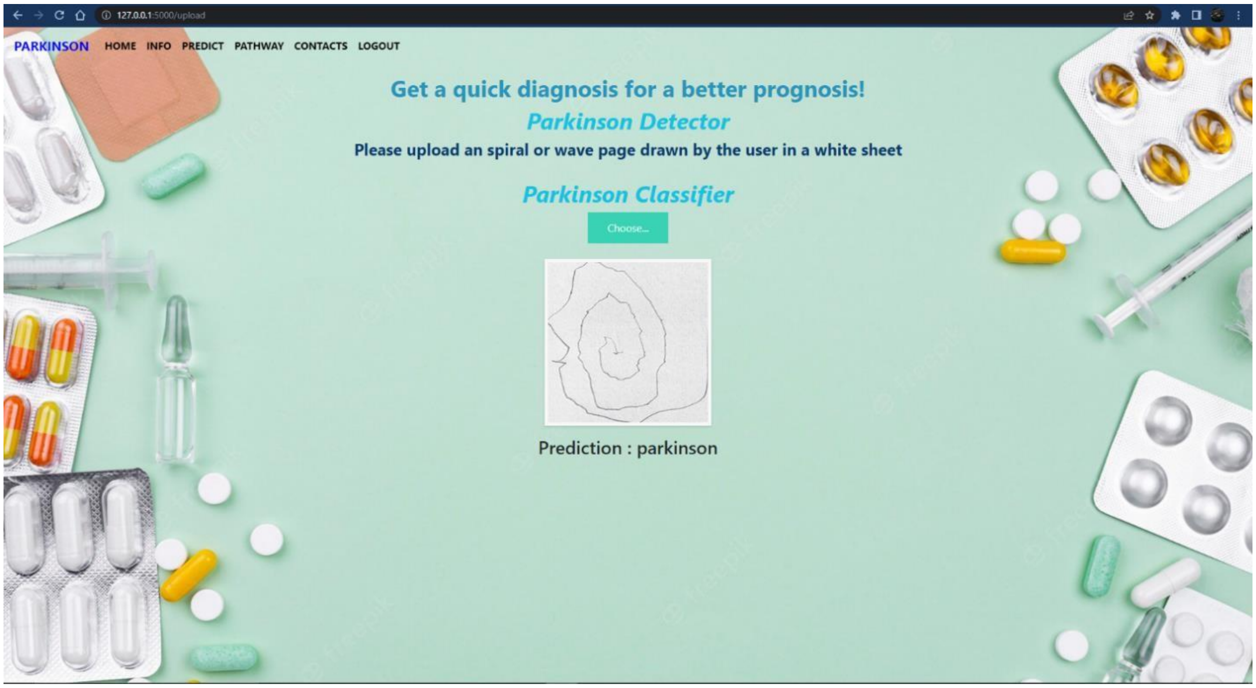
Get a quick diagnosis for a better prognosis!

Parkinson Detector

Please upload an spiral or wave page drawn by the user in a white sheet

Parkinson Classifier

Prediction : healthy



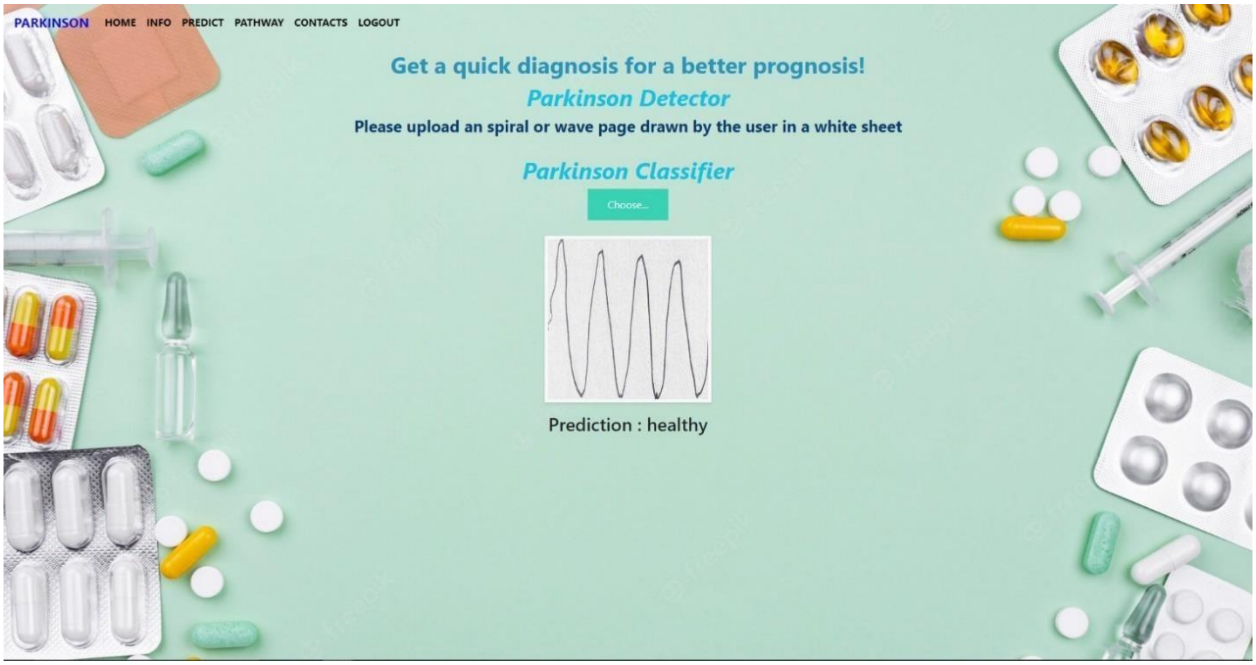
Get a quick diagnosis for a better prognosis!

Parkinson Detector

Please upload an spiral or wave page drawn by the user in a white sheet

Parkinson Classifier

Prediction : parkinson



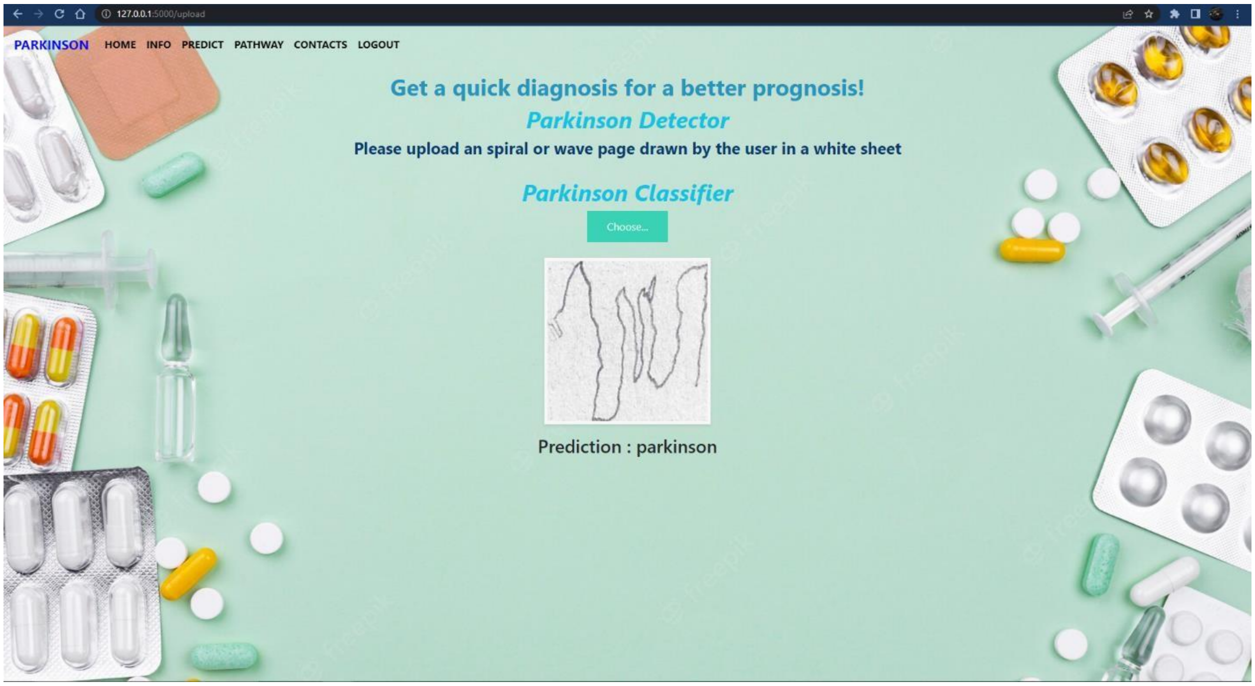
Get a quick diagnosis for a better prognosis!

Parkinson Detector

Please upload an spiral or wave page drawn by the user in a white sheet

Parkinson Classifier

Prediction : healthy



Get a quick diagnosis for a better prognosis!

Parkinson Detector

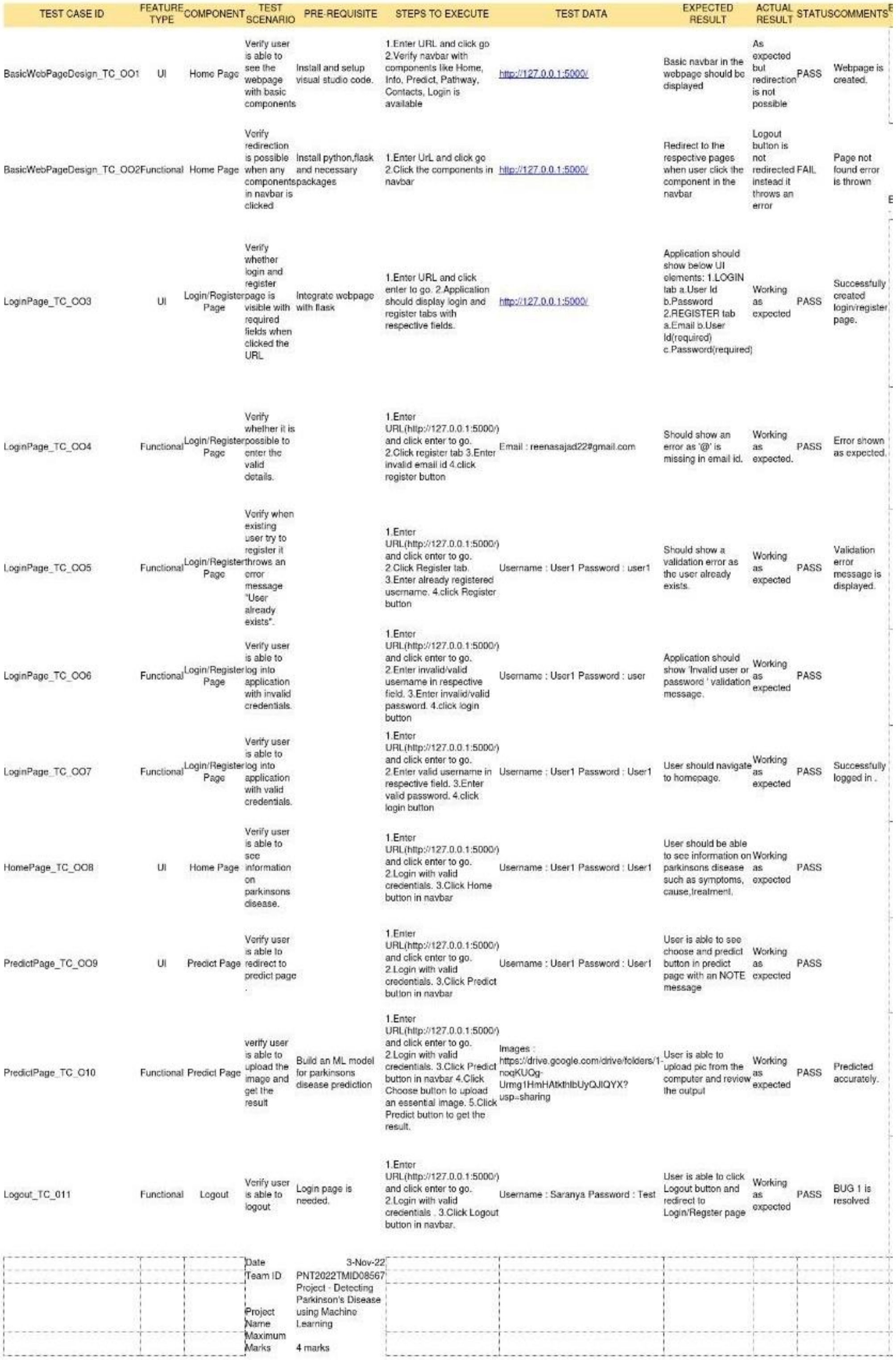
Please upload an spiral or wave page drawn by the user in a white sheet

Parkinson Classifier

Prediction : parkinson

8. TESTING

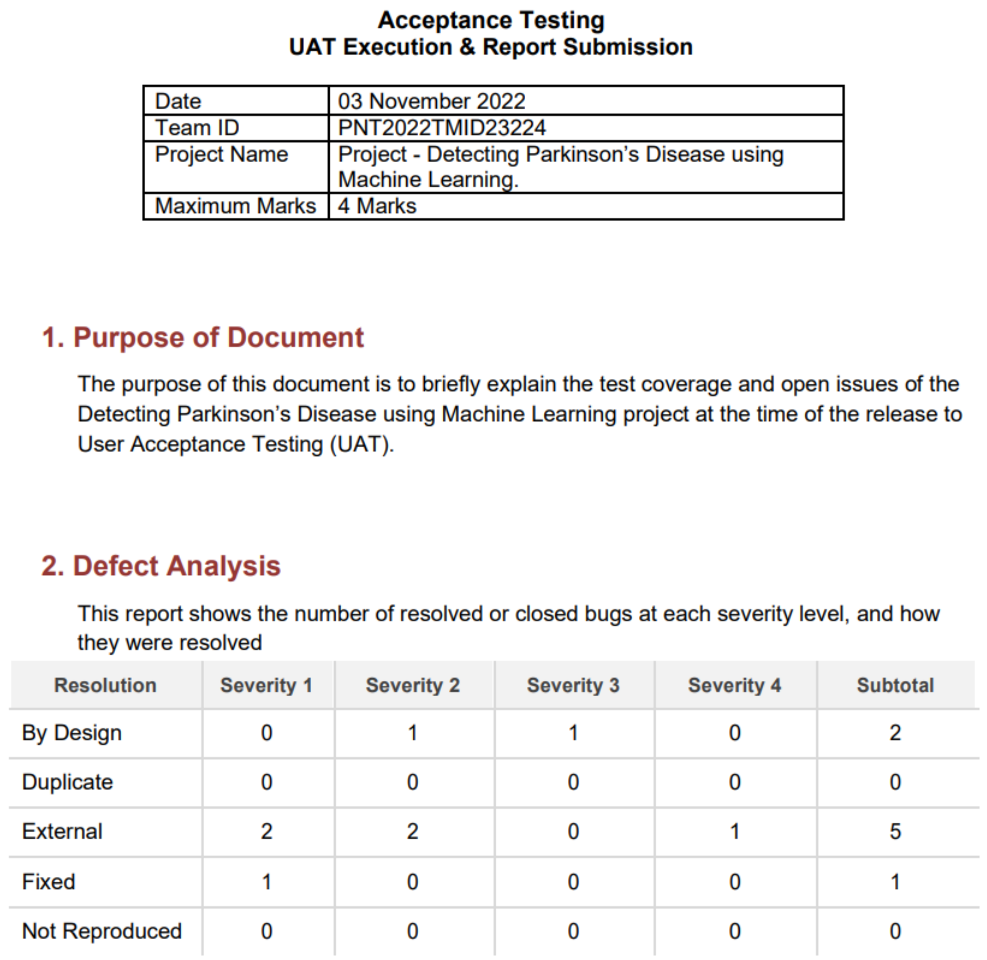
8.1 Test Cases



TEST CASE ID	FEATURE TYPE	COMPONENT	TEST SCENARIO	PRE-REQUISITE	STEPS TO EXECUTE	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS	COMMENTS
BasicWebPageDesign_TC_OO1	UI	Home Page	Verify user is able to see the webpage with basic components	Install and setup visual studio code.	1.Enter URL and click go 2.Verify navbar with components like Home, Info, Predict, Pathway, Contacts, Login is available	http://127.0.0.1:5000/	Basic navbar in the webpage should be displayed	As expected but redirection is not possible	PASS	Webpage is created.
BasicWebPageDesign_TC_OO2	Functional	Home Page	Verify redirection is possible when any components in navbar is clicked	Install python,flask and necessary packages	1.Enter UrL and click go 2.Click the components in navbar	http://127.0.0.1:5000/	Redirect to the respective pages when user click the component in the navbar	Logout button is not redirected instead it throws an error	FAIL	Page not found error is thrown
LoginPage_TC_OO3	UI	Login/Register Page	Verify whether login and register page is visible with required fields when clicked the URL	Integrate webpage with flask	1.Enter URL and click enter to go. 2.Application should display login and register tabs with respective fields.	http://127.0.0.1:5000/	Application should show below UI elements: 1.LOGIN tab a.User Id b.Password 2.REGISTER tab a.Email b.User Id(required) c.Password(required)	Working as expected	PASS	Successfully created login/register page.
LoginPage_TC_OO4	Functional	Login/Register Page	Verify whether it is possible to enter the valid details.		1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Click register tab 3.Enter invalid email id 4.click register button	Email : reenasajad22#gmail.com	Should show an error as '@' is missing in email id.	Working as expected.	PASS	Error shown as expected.
LoginPage_TC_OO5	Functional	Login/Register Page	Verify when existing user try to register it throws an error message "User already exists".		1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Click Register tab. 3.Enter already registered username. 4.click Register button	Username : User1 Password : user1	Should show a validation error as the user already exists.	Working as expected	PASS	Validation error message is displayed.
LoginPage_TC_OO6	Functional	Login/Register Page	Verify user is able to log into application with invalid credentials.		1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Enter invalid/valid username in respective field. 3.Enter invalid/valid password. 4.click login button	Username : User1 Password : user	Application should show 'Invalid user or password ' validation message.	Working as expected	PASS	
LoginPage_TC_OO7	Functional	Login/Register Page	Verify user is able to log into application with valid credentials.		1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Enter valid username in respective field. 3.Enter valid password. 4.click login button	Username : User1 Password : User1	User should navigate to homepage.	Working as expected	PASS	Successfully logged in .
HomePage_TC_OO8	UI	Home Page	Verify user is able to see information on parkinsons disease.		1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Login with valid credentials. 3.Click Home button in navbar	Username : User1 Password : User1	User should be able to see information on parkinsons disease such as symptoms, cause,treatment.	Working as expected	PASS	
PredictPage_TC_OO9	UI	Predict Page	Verify user is able to redirect to predict page .		1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Login with valid credentials. 3.Click Predict button in navbar	Username : User1 Password : User1	User is able to see choose and predict button in predict page with an NOTE message	Working as expected	PASS	
PredictPage_TC_O10	Functional	Predict Page	verify user is able to upload the image and get the result	Build an ML model for parkinsons disease prediction	1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Login with valid credentials. 3.Click Predict button in navbar 4.Click Choose button to upload an essential image. 5.Click Predict button to get the result.	Images : https://drive.google.com/drive/folders/1-noqKUQg-Urmg1HmHAtkthlbUyQJIQYX?usp=sharing	User is able to upload pic from the computer and review the output	Working as expected	PASS	Predicted accurately.
Logout_TC_011	Functional	Logout	Verify user is able to logout	Login page is needed.	1.Enter URL(http://127.0.0.1:5000/) and click enter to go. 2.Login with valid credentials . 3.Click Logout button in navbar.	Username : Saranya Password : Test	User is able to click Logout button and redirect to Login/Regster page	Working as expected	PASS	BUG 1 is resolved

Date	3-Nov-22
Team ID	PNT2022TMID08567
Project Name	Project - Detecting Parkinson's Disease using Machine Learning
Maximum Marks	4 marks

8.2 User Acceptance Testing



Acceptance Testing
UAT Execution & Report Submission

Date	03 November 2022
Team ID	PNT2022TMID23224
Project Name	Project - Detecting Parkinson's Disease using Machine Learning.
Maximum Marks	4 Marks

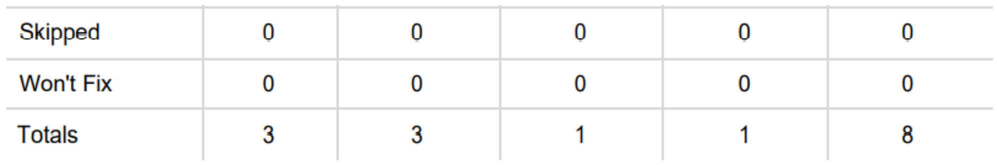
1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Detecting Parkinson's Disease using Machine Learning project at the time of the release to User Acceptance Testing (UAT).

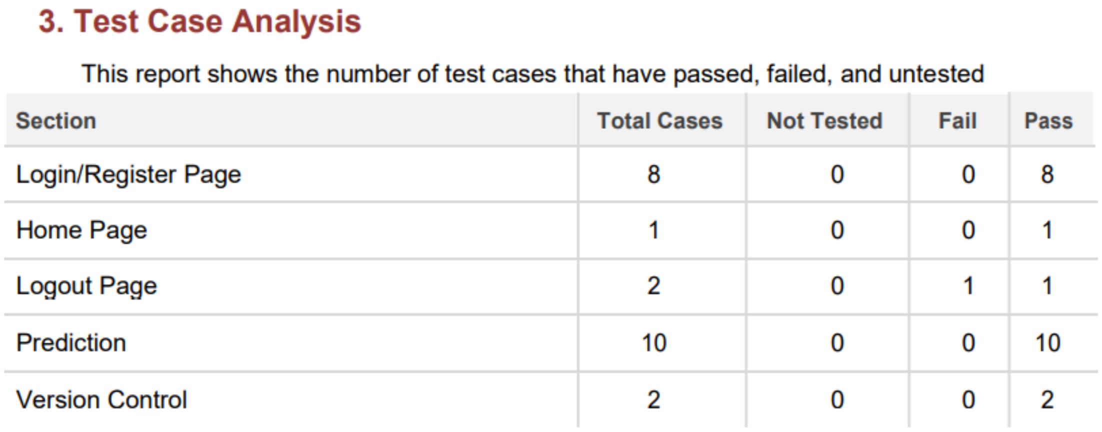
2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	1	1	0	2
Duplicate	0	0	0	0	0
External	2	2	0	1	5
Fixed	1	0	0	0	1
Not Reproduced	0	0	0	0	0



Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	3	3	1	1	8



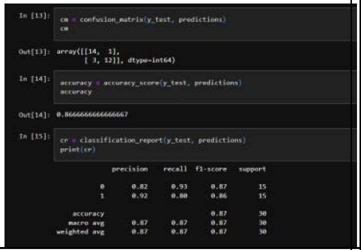
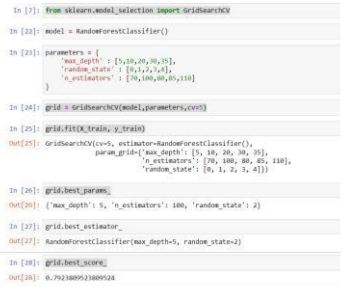
3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Login/Register Page	8	0	0	8
Home Page	1	0	0	1
Logout Page	2	0	1	1
Prediction	10	0	0	10
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

S.No	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix - , Accuracy Score- & Classification Report -	
2.	Tune the Model	Hyperparameter Tuning - GridSearchCV	

```
In [13]:  cm = confusion_matrix(y_test, predictions)
          cm

Out[13]:  array([[14,  1],
                 [ 3, 12]], dtype=int64)

In [14]:  accuracy = accuracy_score(y_test, predictions)
          accuracy

Out[14]:  0.8666666666666667

In [15]:  cr = classification_report(y_test, predictions)
          print(cr)

                        precision    recall  f1-score   support

                   0       0.82      0.93      0.87        15
                   1       0.92      0.80      0.86        15

            accuracy                           0.87        30
           macro avg       0.87      0.87      0.87        30
        weighted avg       0.87      0.87      0.87        30
```

```
In [7]:  from sklearn.model_selection import GridSearchCV

In [22]: model = RandomForestClassifier()

In [23]: parameters = {
             'max_depth' : [5,10,20,30,35],
             'random_state' : [0,1,2,3,4],
             'n_estimators' : [70,100,80,85,110]
         }

In [24]: grid = GridSearchCV(model,parameters,cv=5)

In [25]: grid.fit(X_train, y_train)
Out[25]: GridSearchCV(cv=5, estimator=RandomForestClassifier(),
                      param_grid={'max_depth': [5, 10, 20, 30, 35],
                                  'n_estimators': [70, 100, 80, 85, 110],
                                  'random_state': [0, 1, 2, 3, 4]})

In [26]: grid.best_params_
Out[26]: {'max_depth': 5, 'n_estimators': 100, 'random_state': 2}

In [27]: grid.best_estimator_
Out[27]: RandomForestClassifier(max_depth=5, random_state=2)

In [28]: grid.best_score_
Out[28]: 0.7923809523809524
```

10.ADVANTAGES & DISADVANTAGES

Advantages

- Parkinson’s disease detection using machine learning models could be very effective, cheap, and scalable especially with the advent of transfer learning and pre-trained models which work quite well even with constraints like less data.
- It reduces images to a form which is easier to process without losing features which are critical. Image pre-processing required is much less compared to other algorithms.
- Machine learning does not require the design of handcrafted features, which is one of its biggest advantages.

Disadvantages

- For training and testing, the proposed model requires very high computational time.
- Training the model may be challenging.
- Difficulty in obtaining large datasets to train the model.

o

- Getting a high level of accuracy is difficult due to limited availability of datasets.
- Chances of misdiagnosis is lo but not entirely non existent.

11.CONCLUSION

In this project, we have built a machine learning model that can detect and classify Parkinson’s disease from images of spiral and wave. The model has been integrated into a web application where the user can register to get a diagnosis. The user is required to upload images of spiral and wave drawings. The models are built separately for classifying spiral and wave images using K-Nearest Neighbour and Random Forest respectively. The GUI based application will give the prediction on uploading the image. The system will read the image uploaded by the user, augment it, and use the saved model to detect the presence of Parkinsons disease in the image uploaded by the user. The result is displayed in an easy -tounderstand user-friendly interface.

Parkinson’s disease affects the Central Nervous System and motor functions. There is no definitive cure, but an early prognosis helps in slowing down the progression of the disease. Thus, an early detection can vastly help patients. This detection method helps to overcom e the various constraints which could lead to delayed diagnosis and hence improve the quality of life of the patient.

12.FUTURE SCOPE

The realization of machine learning-assisted diagnosis of Parkinsons Disease yields high potential for a more systematic clinical decision-making system, while adaptation of novel biomarkers may give rise to easier access to Parkinsons Disease diagnosis at an earlier stage. Machine learning approaches therefore have the potential to provide clinicians with additional tools to screen, detect or diagnose Parkinsons Disease.

13.APPENDIX

Source Code

Static:

Main.css

```
.img-preview {     width:
256px;     height: 256px;
position: relative;
border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0,
0.1);     margin-top: 1em;     margin-bottom:
1em;
}

.img-preview>div {     width:
100%;     height: 100%;
background-size: 256px 256px;
background-repeat: no-repeat;
background-position: center;
}
input[type="file"]
{     display:
none;
}
 .upload-label{
display: inline-block;
padding: 12px 30px;
background: #39D2B4;
color: #fff;     font-
size: 1em;
transition: all .4s;
cursor: pointer;
}

.upload-label:hover{
background: #34495E;
color: #39D2B4;
}

.loader {
       border: 8px solid #f3f3f3; /* Light grey */ border-
       top: 8px solid #3498db; /* Blue */ border-radius:
       50%;
       width: 50px;
   height: 50px;
       animation: spin 1s linear infinite;
   }

   @keyframes spin {
       0% { transform: rotate(0deg); }
       100% { transform: rotate(360deg); }
   }
```

Style.css

```
   *{     margin: 0;
   padding: 0;     font-
   family: sans-serif;
   }
    .hero{
   height: 100%;
   width: 100%;
       background-image:linear-gradient(rgba(0,0,0,0.4),
rgba(0,0,0,0.4)),url('img.jpg');
   background-position: center;
   background-size: cover;
   position: absolute;
   } .form-box{
   height: 380px;
   width: 360px;
   position: relative;
   margin: 6% auto;
   background: #fff;
   padding: 5px;
   overflow: hidden;
   }
   .button-box{
       width: 220px;
       margin: 35px auto;
       position: relative;
       box-shadow: 0 0 20px 9px #5f97e51f;     border-
   radius: 40px;
```

```
} .toggle-btn{
padding: 10px 30px;
cursor: pointer;
background: transparent;
border: 0;     outline:
none;     position:
relative;
} #btn{     top: 0;
left: 0;     position:
absolute;     width:
110px;     height:
100%;
    background: linear-gradient(to right,
#7369ca,#11b1c3);     border-radius: 30px;
transition: 0.5s;
} .input-group{
top: 120px;
position: absolute;
width: 280px;
transition: .5s;

} .input-field{
width: 100%;
padding: 10px 0;
margin: 5px 0;
border-left: 0;
border-top: 0;
border-right: 0;
border-bottom: 1px
solid #999;
outline: none;
background:
transparent;
} .submit-btn{
width: 85%;
padding: 10px 30px;
cursor: pointer;
display: block;
margin: auto;
```

```
    background: linear-gradient(to right,
#4e4888,#7bc0c8);     border: 0;     outline: none;
border-radius: 30px;

} .check-box{     margin:
30px 10px 30px 0;
} span{     color:
#777;     font-size:
12px;     bottom:
68px;     position:
absolute;

} #login{
left: 50px; }
#register{
left: 450px;
} .err{
    color:rgb(198, 156, 243);
margin: 265px 0 0 145px;
}
```

Main.js

```
   $(document).ready(function () {
       // Init
       $('.image-section').hide();
       $('.loader').hide();
       $('#result').hide();

       // Upload Preview
   function readURL(input) {
           if (input.files && input.files[0]) {
   var reader = new FileReader();
   reader.onload = function (e) {
                   $('#imagePreview').css('background-image', 'url(' +
e.target.result + ')');
                   $('#imagePreview').hide();
                   $('#imagePreview').fadeIn(650);
               }
               reader.readAsDataURL(input.files[0]);
           }
       }
```

```
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
$('#result').hide();
readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
type: 'POST',
url: '/predict',
data: form_data,
contentType: false,
cache: false,
processData: false,
async: true,
success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text('Prediction : '+data);
console.log('Success!');
            },
        });
    });

});
```

Templates Base.html

```
<html lang="en">
```

```
       <head>
           <meta charset="UTF-8">
           <meta name="viewport" content="width=device-width,
initialscale=1.0">
           <meta http-equiv="X-UA-Compatible" content="ie=edge">
           <title>Predict</title>
           <link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
           <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></scri
pt >
           <script
src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
           <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></scr
ip t>
           <link href="{{ url_for('static', filename='css/main.css')
}}" rel="stylesheet">
       <style>
       .bar    {
   margin: 0px;
   padding:20px;
       background-color:rgb(169, 223,
   241);     opacity:0.6;
   color:black;
       font-family:'Roboto',sans-serif;
   font-style: italic;     border-
   radius:20px;     font-size:25px;
       }

   body{
           background-image:
 url("https://img.freepik.com/freephoto/flat-lay-medical-desk-
composition-with-copy-space_23-
2148502943.jpg?w=2000");
   position: relative;
   background-size: cover;
   background-repeat: no-repeat;
   height: 100%;             width: 100%;
```

```
            }
          h1{
              font-size:35px;
   text-align:center;
   color:#2596be;             font-
   style:Helvetica;             font-
   weight:bolder;
          }
   h2{
              font-size:35px;
   text-align:center;
   color:rgb(17, 196, 227);             font-
   style:italic;             font-
   weight:bolder;
          }
   h5{
              font-size:25px;
   text-align:center;
   color:#063970;           font-
   style:Helvetica;             font-
   weight:bolder;
          }
       a{
           color:black;
       }
      </style>
      </head>

      <body>

      <nav class="navbar navbar-expand-md fixed-top navbar-dark pl-3"
   >
        <a           style="color:blue;"           class="navbar-brand"
href="/home"><strong>PARKINSON</strong></a>
        <button  class="navbar-toggler"       type="button"
 datatoggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
```

```
         </button>
         <div class="collapse navbar-collapse" id="navbarNav">
           <ul class="navbar-nav">
             <li class="nav-item">
               <strong><a    style="color:black;"   class="nav-link"
href="/home">HOME</a></strong>
             </li>
             <li class="nav-item">
               <strong><a    style="color:black;"   class="nav-link"
href="/home">INFO</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:black;"       class="nav-link"
href="/upload">PREDICT</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:black;"       class="nav-link"
href="/home">PATHWAY</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:black;"       class="nav-link"
href="/home">CONTACTS</a></strong>
             </li>
             <li class="nav-item">
              <strong><a     style="color:black;"      class="nav-link"
href="/logout">LOGOUT</a></strong>
             </li>
           </ul>
         </div>
       </nav>


        <br><br><br>
           <h1>Get a quick diagnosis for a better prognosis!</h1>
           <h2><center>Parkinson Detector</center></h2>
           <h5>Please upload an spiral or wave page drawn by the user
in a white sheet</h5>
           <div class="container">
              <center> <div id="content" style="margin-top:2em">{%
block content %}{% endblock %}</div></center>
           </div>
       </body>
       <footer>
           <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
       </footer>

       </html>
```

Home.html

```
<!doctype
html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initialscale=1, shrink-to-fit=no">
    <meta   name="Description"   content="An   information   page   for
patients diagnosed with Parkinson's Disease and living in Oxfordshire.
You can find the contacts for your nurse specialist, links and your care
pathway with your referrals or clinical appointments.">
    <!-- Google Font -->
    <link
href="https://fonts.googleapis.com/css2?family=Abril+Fatface&family=Bar
lo w:wght@400;700&display=swap" rel="stylesheet">
    <!-- Font Awesome -->
    <link                                          rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.14.0/css/all.min.css"                   integrity="sha512-
1PKOgIY59xJ8Co8+NE6FZ+LOAZKjy+KY8iq0G4B3CyeY6wYHN3yt9PW0XpSriVlkMXe40PT
Kn XrLnZ9+fkDaog==" crossorigin="anonymous" />
    <!-- Bootstrap CSS -->
    <link                                          rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min
.c ss"      integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
    <!-- CSS file -->
    <link rel="stylesheet" href="./css/index.css">

    <title>Parkinson Prediction and Information Page </title>
  </head>
  <body>
```

```
       <!-- Navbar -->
       <nav class="navbar navbar-expand-md fixed-top navbar-dark pl-3"
   >
         <a                                        class="navbar-brand"
href="#home"><strong>PARKINSON</strong></a>
         <button  class="navbar-toggler"      type="button"
 datatoggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
           <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarNav">
           <ul class="navbar-nav">
             <li class="nav-item">
               <strong><a   style="color:grey;"       class="nav-link"
href="/home">HOME</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:grey;"       class="nav-link"
href="#info">INFO</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:grey;"       class="nav-link"
href="/upload">PREDICT</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:grey;"       class="nav-link"
href="#pathway">PATHWAY</a></strong>
             </li>
             <li class="nav-item">
               <strong><a   style="color:grey;"       class="nav-link"
href="#contacts">CONTACTS</a></strong>
             </li>
             <li class="nav-item">
              <strong><a     style="color:grey;"     class="nav-link"
href="/logout">LOGOUT</a></strong>
             </li>
           </ul>
         </div>
       </nav>
       <!-- Header -->
       <header id="home">
```

```
         <div style="background-color:powderblue;" class="jumbotron
dflex flex-column align-items-start justify-content-center">
           <h1 class="display-4 text-uppercase"><strong>Parkinson's
prediction website</strong></h1>
           <p class="lead text-uppercase">Welcome to the Parkinson
Information Page</p>
           <p class="pr-lg-5 text-justify">More than 10 million people
are living with Parkinson’s Disease worldwide, according to the
Parkinson’s Foundation. While Parkinson’s cannot be cured, early
detection along with proper medication can significantly improve
symptoms and quality of life.

   The researchers found that the drawing speed was slower and the pen
pressure is lower among Parkinson’s patients. One of the indications of
Parkinson’s is tremors and rigidity in the muscles, making it difficult
to draw smooth spirals and waves. It is possible to detect Parkinson’s
disease using the drawings alone instead of measuring the speed and
pressure of the pen on paper. Our goal is to quantify the visual
appearance(using HOG method) of these drawings and then train a machine
learning model to classify them. In this project, We are using, Histogram
of Oriented Gradients (HOG) image descriptor along with a Random Forest
classifier to automatically detect Parkinson’s disease in hand-drawn
images of spirals and waves.This website was created to help in early
and economically feasible detection of Parkinsons' disease.</p>
           <div>
             <img src="https://pic2.zhimg.com/80/v2-
211faee9982aeaa2acf18e31e03cacb9_1440w.webp" alt="Illustration about
Parkinson's Symptoms" width="800" height="400" />
           </div>
         </div>
       </header>

       <main>
         <!-- Info -->
         <div class="info container" id="info">
           <div class="section-title">
             <h2>Info</h2>
           </div>
           <!-- Cards -->
           <div class="card-deck">
```

```
            <div class="card mb-3">
              <div class="card-body">
                <h5 class="card-title text-uppercase font-
weightbold">Parkinson's Disease</h5>
                <p class="card-text text-justify">Parkinson Disease
(PD) is a neurodegenarative disorder. This means that neurons in the
part of the brain called substantia nigra are affected, which leads to
a reduction of dopamine production. Dopamine is a chemical
neurotransmitter that regulates the movements of the body. The causes
remain unknown. However, it is very important to remember that PD affects
people in many different ways.</p>
                <a
 href="https://www.nhs.uk/conditions/parkinsonsdisease/"
target="_blank" class="btn btn-pink"> Parkinson Disease (NHS)</a>
              </div>
            </div>

            <div class="card mb-3">
              <div class="card-body">
                <h5 class="card-title text-uppercase font-
weightbold">Symptoms</h5>
                <p class="card-text text-justify">Symptoms generally
develop slowly. Although PD affects patients in many different ways, the
most common symptoms are tremor (involuntary shaking of parts of the
body), rigidity of the limbs, bradykinesia (slow movements) or balance
problems. Some other non-motor symptoms may also be exprienced such as
nerve pain, speech difficulties, dysphagia (swallowing difficulties),
urinary incontinence, anxiety or depression.</p>
                <a
 href="https://www.nhs.uk/conditions/parkinsonsdisease/symptoms/"
target="_blank" class="btn btn-pink"> Parkinson Symptoms (NHS)</a>
              </div>
            </div>

            <div class="card mb-3">
              <div class="card-body">
                <h5 class="card-title text-uppercase font-
weightbold">Treatments</h5>
                <p class="card-text text-justify">It is possible to
live a good quality of life with Parkinson. Your specialist team will
tailor your treatment according to your symptoms. You may receive
```

```
dopaminergic medication such as Levodopa&copy;, Madopar&copy; or
Sinemet&copy;. Nonmedical treatment are also provided by
Physiotherapists. Your specialist will discuss with you if you have the
option, at an advance stage, to receive further therapy such as DBS
(Deep Brain Stimulation).</p>
                 <a
 href="https://www.nhs.uk/conditions/parkinsonsdisease/"
target="_blank" class="btn btn-pink"> Neurosciences Leaflets (OUH)</a>

               </div>
             </div>
           </div>
         </div>

         <!-- Pathway -->
         <div class="pathway container" id="pathway">
           <div class="section-title">
             <h2 >Pathway</h2>
           </div>
           <!-- Pathway Chart -->
           <div class="d-flex flex-column pathway-chart">
             <!-- 1 - Referral -->
             <div class="pathway-step border border-white">
               <h4><i class="fas fa-notes-medical"></i> 1: Referral
   </h4>
               <p>Visit your GP or another medical specialist to get
referred to a Parkinson's Specialist Consultant or a
Neurology/Geratology Consultant.</p>
             </div>
             <!-- 2 - Appointment -->
             <div class="pathway-step border border-white">
               <h4><i class="fas fa-hospital"></i> 2: Appointment</h4>
               <p>You will receive an appointment date with the
Neurologist or Geratologist at the John Radcliffe Hospital. A diagnosis
will be made and you will be referred to your local Parkinson's Disease
Specialist Nurse.</p>
             </div>
             <!-- 3 - Review -->
             <div class="pathway-step border border-white">
               <h4><i class="fas fa-user-nurse"></i> 3: Review</h4>
               <p>Within 6 to 8 weeks after your diagnosis, you will be
seen by your Parkinson's Disease Specialist Nurse. You will be given an
information pack and be referred to the Multidisciplinary team.</p>
             </div>
             <!-- 4 - MDT -->
             <div class="pathway-step border border-white">
               <h4><i class="fas fa-user-plus"></i> 4:
Multidisciplinary Team</h4>
               <p>You may be referred to the First Step Programme, a
support group run by people with Parkinson's, in your local area. If
required you may be referred to therapists such as:</p>
                 <ul class="ml-5">
                   <li>Physiotherapists (PT)</li>
                   <li>Occupational Therapists (OT)</li>
                   <li>Speech and Language Therapists (SALT)</li>
                   <li>Dieticians</li>
                   <li>Social Services</li>
                 </ul>
               <p><i class="fas fa-exclamation mr-2"></i> Inform your
nurse if you need any of the sevices above.</p>
             </div>
             <!-- 5 - Regular Appointments -->
             <div class="pathway-step border border-white">
               <h4><i class="fas fa-clinic-medical"></i> 5: Regular
Clinics Appointments</h4>
               <p>Every year you will receive at least two appointments
with your healthcare professional:</p>
                 <ul class="list-unstyled ml-4">
                   <li><i class="fas fa-user-md"></i> Consultant: Once
a year</li>
                   <li><i class="fas fa-user-nurse"></i> Spcialist
Nurse: Once a year or more frequently if necessary.</li>
                 </ul>
               <p><i class="fas fa-exclamation mr-2"></i> Your
specialist nurse may see you at home <em>if needed</em>. You are
encouraged to contact your Specialist Nurse frequently depending on your
needs.</p>
             </div>
           </div>
         </div>
```

```
       <!-- Contacts -->
       <div class="contacts container" id="contacts">
         <div class="section-title">
           <h2>Contacts</h2>
           <p>You can find here the contact details of Parkinson's
Disease Specialists </p>
         </div>
         <!-- Cards -->
         <div class="card-deck">
           <div class="card mb-3">
             <div class="card-body">
               <h5 class="card-title text-uppercase font-weight-
   bold">Dr.
Yuvasri Chinnuswamy</h5>
               <p class="card-text text-justify"><i class="fas fa-
phonealt pr-2"></i>+91 902-342 1803</p>
               <p>(Administrator Office)</p>
               <p    class="card-text"><i    class="far    fa-envelope
pr2"></i>Neurosciences Offices, West Wing<br> Sri Ramakrishna Hospital,
Coimbatore</p>
             </div>
           </div>

           <div class="card mb-3">
             <div class="card-body">
               <h5 class="card-title text-uppercase font-weight-
   bold">Dr.
K Ramadoss</h5>
               <p class="card-text text-justify"><i class="fas fa-
phonealt pr-2"></i>+91 422-224 0521</p>
               <p class="card-text text-justify"><i class="far
faenvelope pr-2"></i>Velavan Health Center <br>Specialty Orthopaedic
Centre</p>
               <p class="card-text text-justify"><i class="fas fa-
mapsigns pr-2"></i>Areas: Pollachi, Coimbatore (city), Tamil Nadu</p>
             </div>
           </div>

           <div class="card mb-3">
             <div class="card-body">
```

```
               <h5 class="card-title text-uppercase font-weight-
   bold">Dr.
Pranesh Upadhyay</h5>
               <p class="card-text text-justify"><i class="fas fa-
phonealt pr-2"></i>+91 759-891 2803</p>
               <p class="card-text text-justify"><i class="far
faenvelope pr-2"></i>Dr. Pranesh Clinic <br> Specialty Orthopaedic
Centre</p>
               <p class="card-text text-justify"><i class="fas fa-
mapsigns pr-2"></i>Areas: Tiruppur and Pollachi</p>
             </div>
           </div>
         </div>
       </div>

       <!-- Maps -->
       <div class="maps container" id="maps">
         <div class="section-title">
           <h2>Maps</h2>
         </div>
         <!-- Cards -->
         <div class="card-deck">
           <div class="card mb-3">
             <iframe            class="container-fluid            px-0"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3916.2062227
11
9704!2d76.97539531407273!3d11.023148657607104!2m3!1f0!2f0!3f0!3m2!1i102
4!
2i768!4f13.1!3m3!1m2!1s0x3ba8584e4d002f0d%3A0x2b94348a8824200f!2sSri%20
Ra makrishna%20Hospital%20(%20Multi-
Speciality%20Hospital%20in%20Coimbatore)!5e0!3m2!1sen!2sin!4v1668579013
70 1!5m2!1sen!2sin"    width="400"     height="200"
 frameborder="0" style="border:0;" allowfullscreen=""    aria-
hidden="false" tabindex="0"></iframe>
             <div class="card-body">
               <h5 class="card-title text-uppercase font-weight-
   bold">Sri
Ramakrishna Hospital</h5>
               <p class="card-text text-justify"><i class="far
faenvelope pr-2"></i> 395, Sarojini Naidu Rd, Siddhapudur, Balasundaram
Layout, B.K.R Nagar<br>Coimbatore, Tamil Nadu 641044 </p>
```

```
               <p class="card-text text-justify"><i class="fas fa-
phonealt pr-2"></i> 0300 304 7777</p>
             </div>
           </div>

           <div class="card mb-3">
             <iframe class="container-fluid px-0"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3916.111860
862
161!2d76.9736064140728!3d11.030233457475433!2m3!1f0!2f0!3f0!3m2!1i1024
!2i
768!4f13.1!3m3!1m2!1s0x3ba8597c85cadec9%3A0x291d9d0a449d54b7!2sVelavan
%20 health%20center!5e0!3m2!1sen!2sin!4v1668579195887!5m2!1sen!2sin"
width="600" " width="400" height="200" frameborder="0" style="border:0;"
allowfullscreen="" aria-hidden="false" tabindex="0"></iframe>
             <div class="card-body">
               <h5 class="card-title text-uppercase font-
weightbold">Velavan's Health center</h5>
               <p class="card-text text-justify"><i class="far
faenvelope pr-2"></i> Sivasakthi Colony, Ganapathy, Coimbatore, Tamil
Nadu
641006</p>
               <p class="card-text text-justify"><i class="fas fa-
phonealt pr-2"></i> 0300 304 7777</p>
             </div>
           </div>
         </div>
       </div>

       <!-- Links
       <div class="links container" id="links">
         <div>
           <img
src="https://pbs.twimg.com/media/E2ALbWqWEAMcu9p.jpg:large"
alt="Illustration about Parkinson's Symptoms" width="800" height="400"
/>       </div>
       </div> -->
     </main>
```

```
      <!-- Footer -->
      <footer>
       <div>
         <p>2022© All rights reserved.</p>
     </footer>

       <!-- Optional JavaScript -->
       <!-- jQuery first, then Popper.js, then Bootstrap JS -->
       <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
       <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper
.m in.js"   integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
       <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.j
s" integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
     </body>
   </html>
```

Index.html

```
   <html>
       <head>
           <title>PARKINSON'S DISEASE </title>
           <link             rel             =             "stylesheet"
href="{{url_for('static',filename='css/style.css')}}">
       </head>
       <body style="background-image: pink;">
           <div class="hero">
               <div class="form-box">
                   <div class="button-box">
                       <div id="btn"></div>
                       <button    type="button"      class="toggle-btn"
onclick="login()">Log In</button>
                       <button    type="button"      class="toggle-btn"
onclick="register()">Register</button>
                   </div>
```

```
                    <form id="login" class="input-group"
action="/form_login" method="post">
                        <input type="text" class="input-field"
placeholder="User Id" name ="userid" required>
                        <input type="password" class="input-field"
placeholder="Password" name="pwd" required>
                        <input type="checkbox" class="check-
box"><span>Remember Password</span>
                        <button type="submit" class="submit-btn"
value="Login">Login</button>
                    </form>
                    <h6 class="err">{{info}}</h6>
                    <form id="register" class="input-group"
action="/form_reg" method="post">
                        <input type="email" class="input-field"
placeholder="Email Id">
                        <input type="text" class="input-field"
placeholder="User Id" name ="userid" required>
                       <input type="password" class="input-field"
placeholder="Password" name="pwd" required>
                        <button type="submit" id = "sub"
class="submitbtn" >Register</button>
                    </form>
                    <h6 class="err">{{info}}</h6>
                </div>
            </div>
   <script>
               var x = document.getElementById("login")
   var y = document.getElementById("register")
   var z = document.getElementById("btn")
   function register(){
                   x.style.left = "-400px";
                   y.style.left = "50px";
                   z.style.left = "110px";
   }
               function login(){
                   x.style.left = "50px";
                   y.style.left = "450px";
                   z.style.left = "0px";
               }
```

```
           </script>
       </body>
   </html>
```

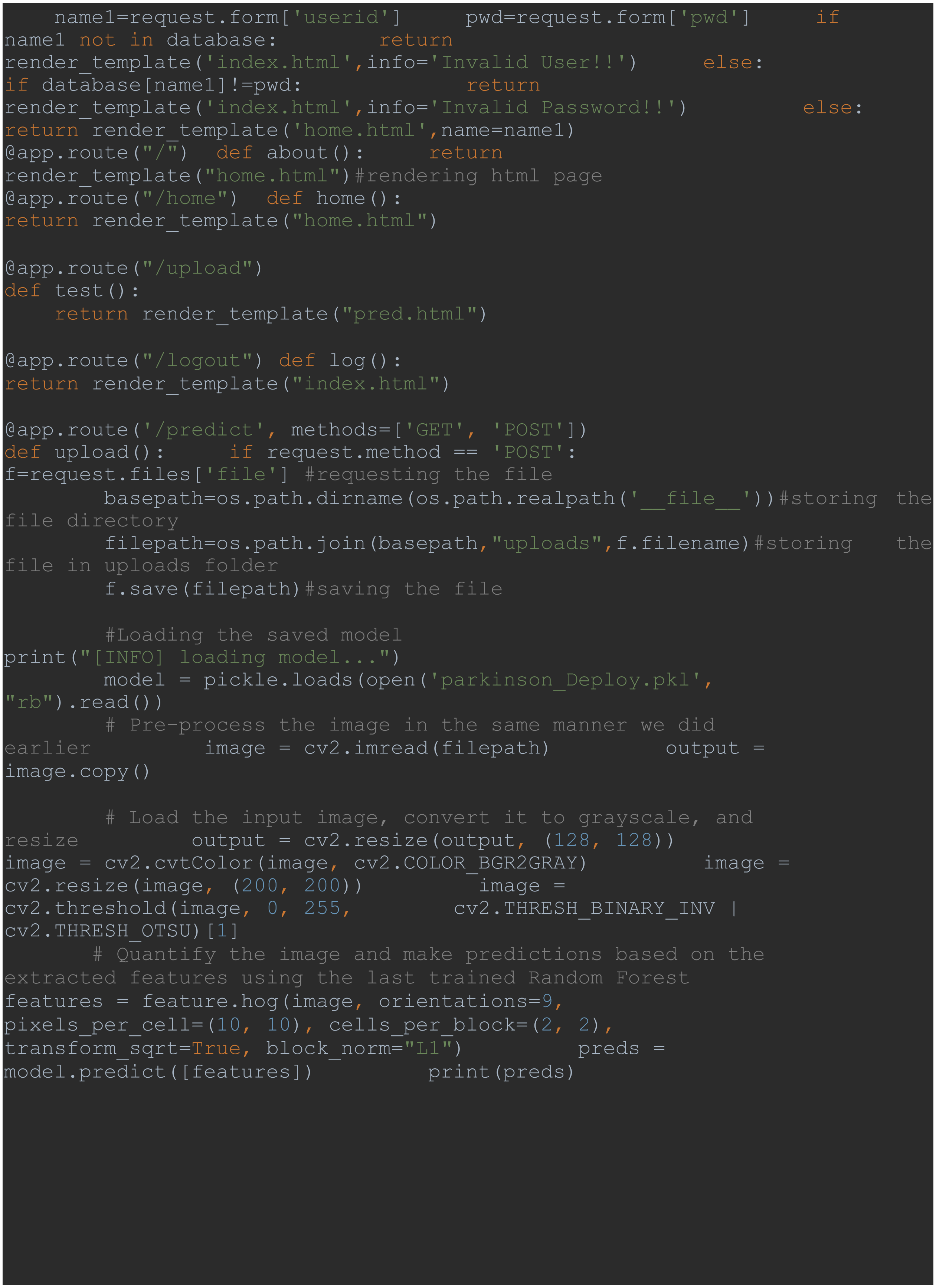
Pred.html

```
   {% extends "base.html" %} {% block content %}

   <h2><center>Parkinson Classifier</center></h2>

   <div>
       <form id="upload-file" method="post"
enctype="multipart/formdata">
       <center>    <label for="imageUpload" class="upload-label">
   Choose...
           </label>
           <input type="file" name="file" id="imageUpload"
accept=".png, .jpg, .jpeg">
       </center></form>

      <center> <div class="image-section" style="display:none;">
           <div class="img-preview">
               <div id="imagePreview">
               </div></center>
           </div>
           <center><div>
               <button type="button" class="btn btn-primary btn-lg "
id="btn-predict">Predict!</button>
           </div></center>
       </div>

       <div class="loader" style="display:none;"></div>

       <h3 id="result">
           <span> </span>
       </h3>

   </div>

   </div>
```

```
{% endblock %}
```

App.js

```
   from flask import Flask,request,render_template
import pickle
import cv2
from skimage import feature
import os.path

app = Flask(__name__)


@app.route('/') def hello_world():
return render_template("index.html")
class my_dictionary(dict):
  def __init__(self):
    self = dict()    def
add(self, key, value):
    self[key] = value
database=my_dictionary()


@app.route('/form_reg',methods=['POST','GET'])
def reg():
    name2=request.form['userid']     pwd1=request.form['pwd']     if
name2 in database:          return
render_template('index.html',info='UserName Already Taken!!')     else:
database.add(name2,pwd1)
        return render_template("index.html")
@app.route('/form_login',methods=['POST','GET'])
def login():
```



```
    name1=request.form['userid']     pwd=request.form['pwd']     if
name1 not in database:        return
render_template('index.html',info='Invalid User!!')     else:
if database[name1]!=pwd:             return
render_template('index.html',info='Invalid Password!!')         else:
return render_template('home.html',name=name1)
@app.route("/")  def about():     return
render_template("home.html")#rendering html page
@app.route("/home")  def home():
return render_template("home.html")

@app.route("/upload")
def test():
    return render_template("pred.html")

@app.route("/logout") def log():
return render_template("index.html")

@app.route('/predict', methods=['GET', 'POST'])
def upload():     if request.method == 'POST':
f=request.files['file'] #requesting the file
        basepath=os.path.dirname(os.path.realpath('__file__'))#storing the
file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing   the
file in uploads folder
        f.save(filepath)#saving the file

        #Loading the saved model
print("[INFO] loading model...")
        model = pickle.loads(open('parkinson_Deploy.pkl',
"rb").read())
        # Pre-process the image in the same manner we did
earlier        image = cv2.imread(filepath)        output =
image.copy()

        # Load the input image, convert it to grayscale, and
resize         output = cv2.resize(output, (128, 128))
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]
       # Quantify the image and make predictions based on the
extracted features using the last trained Random Forest
features = feature.hog(image, orientations=9,
pixels_per_cell=(10, 10), cells_per_block=(2, 2),
transform_sqrt=True, block_norm="L1")         preds =
model.predict([features])         print(preds)
```

```
ls=["healthy","parkinson"]
result = ls[preds[0]]
return result      return None
 if __name__ ==
'__main__':     app.run()
```

Git Hub Link

https://github.com/IBM-EPBL/IBM-Project-36250-1660293722

Project Demo Link

https://drive.google.com/file/d/1jFZCdYmlObIevNRseUWLc-vYbZruxphI/view?usp=share_link