Project Report

DETECTION OF PARKINSON’S DISEASE USING MACHINE LEARNING

Submitted By

PNT2022TMID15817

TABLE OF CONTENTS

CHAPTER – 1

INTRODUCTION

1.1.PROJECT OVERVIEW:

Parkinson's disease is one of the most common neurodegenerative diseases with a prevalence rate of 1% in the population above 60 years old. The diagnosis of PD is traditionally based on motor symptoms. The symptoms of the disease will occur slowly, the symptoms include shaking, rigidity, slowness of movement and difficulty with walking, thinking and behavior change are common symptoms of this disease. This disease severely affects patients quality of life(QoL), social functions and family relationships, and places heavy economic burdens at individual and society levels. There is no defined test for early diagnosis of Parkinson's patient and medical decisions are provided based on the medical history of the patient and hence the possibility of misdiagnosis. Several researches were made to predict this disease in early stage.But they didn’t predict properly and cannot be able to give better results.

1.2 .PURPOSE:

The aim of the project is vocal dysphonia analysis of Parkinson's patient from voice dataset with different machine learning algorithms with a goal to achieve better performance with less number of attributes. For addressing these difficulties and to refine the diagnosis and assessment procedures of PD, machine learning methods have been implemented for the classification of PD and healthy controls or patients with similar clinical presentations which can helpful to predict the disease earlier. Machine learning techniques are being increasingly applied in the healthcare sector. As its name implies, machine learning allows for a computer program to learn and extract meaningful representation from data.

CHAPTER – 2

LITERATURE SURVEY

2.1.EXISTING PROBLEM:

In existing system, the comparative study of various machine learning algorithms is carried out. For analysis and prediction of Parkinson's PPMI data sets and six different classification algorithms are used. The results show that the multiclass classifier and logistic regression better performed than the other algorithms for the data sets. In future, more number of biomarker features are to be included for the prediction of progression of P This study proposed regression, decision tree and neural network analysis to analyse the databank of Parkinson disease for error probability calculated. The result was logistic regression, classification and neural network analysis error probability by 5.15%, 8.47% and 23.73% respectively.

2.2.REFERENCES:

S.no	Year	Researcher	Title	Methodology	Remarks
01	2019	S. Kanagaraj, M.S. Hema, M. Nageswara Gupta	Machine Learning Techniques for Prediction of Parkinson's Disease using BigData	Progression Marker Initiative (PPMI)	Predicts Parkinson's disease at an early stage from the formerly available public database
02	2019	F.M. Javed Mehedi Shamrat, Md. Asaduzzaman, A.K.M. Sazzadur Rahman, Raja Tariqul Hasan Tusher, Zarrin Tasnim	A Comparative Analysis Of Parkinson Disease Prediction Using Machine Learning	machine learnin gtechniques	Thus, different experiments to assess the three machine learning supervised algorithms for recognition of Parkinson's disease

03	2014	Ma, C. et al	Identifying Parkinson diseaseusing machine learning	SVM, KNN, and extreme learnin gmachine (ELM)	discriminate healthy people from those withParkinson’s disease
04	2014	Ma, Ouyang, Chen, & Zhao	analysis of Parkinson disease	SCFW-KELM	Hybrid method is used to analysis Parkinson’s disease
05	2014	Yahia A. et al	classification algorithm based on Naïve Bayes and K- Nearest Neighbours (KNN)	Parkinson speech dataset	Thus, Parkinson’s Disease is detected through voice signal
06	2013	Chen et al.,	demonstrative precision for the identification of Parkinson Disease	fuzzy-based KNN model, a hybrid model	Identified the Parkinson’s disease using these methods
07	2013	Sriram, Rao, Narayana, Kaladhar, Vital	detection of Parkinson diseases using machine learning algorithms	voice data	Thus, the analysis of voice data to understand the presence of Parkinson disease
08	2011	Rusz J	measurements to differentiate Parkinson diseasefrom healthy subjects	vector machine	Vector machine is used to differentiate Parkinson disease
09	2011	Ozcift A. et al	detection of Parkinson diseases using machine learning algorithms	computer-aided diagnosis (CADx) systems	Thus, Parkinson’s Disease is detected
10	2011	Wu, S et al	Analysing the databank of Parkinson disease	regression, decision tree and neural network	Thus databank of Parkinson’s disease is analysed

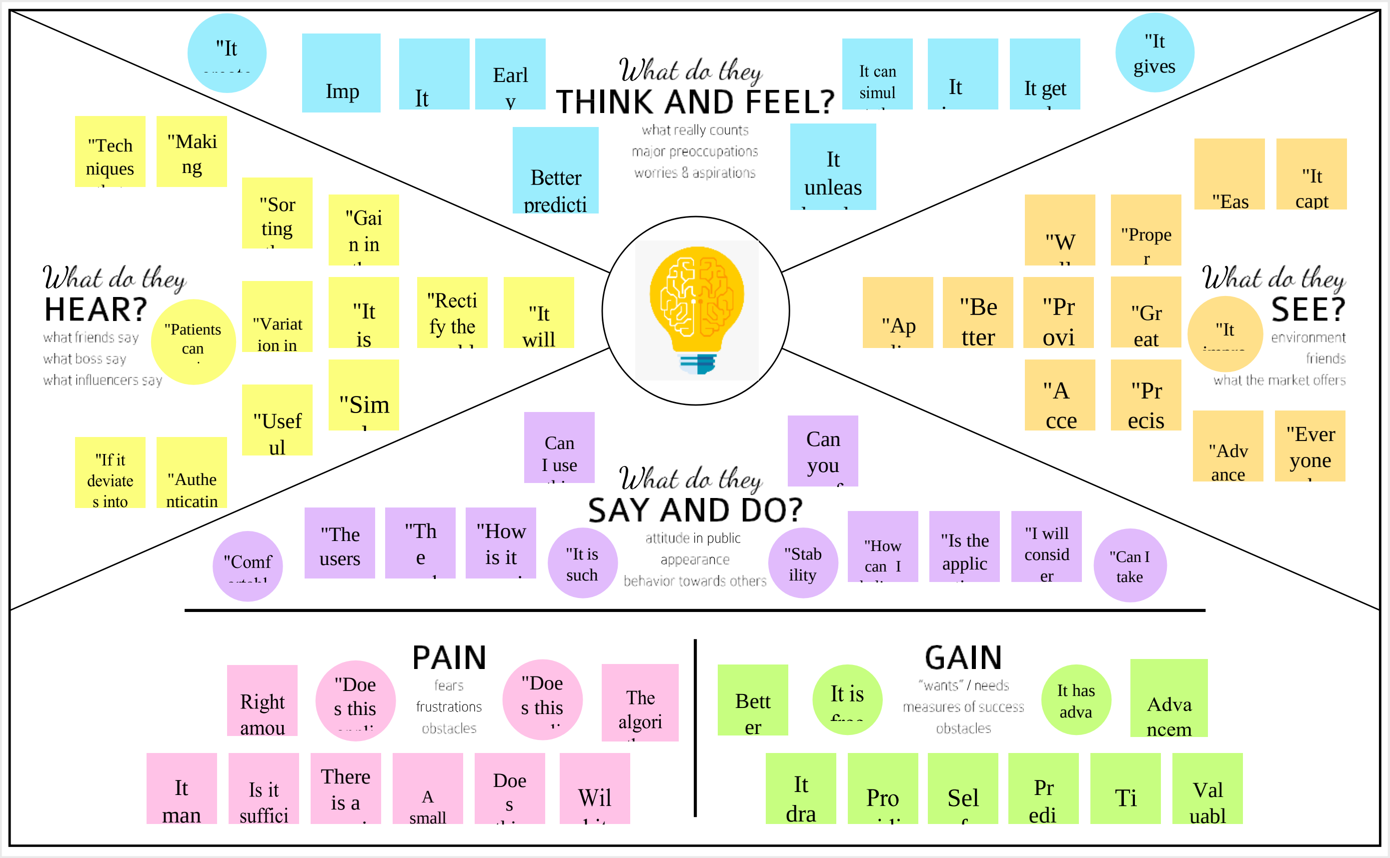
2.3.PROBLEM STATEMENT DEFINITION:

Who does the problem affect?	People who are men with minimization of nerve cells in primarily of village areas.
What are the boundaries of the problem?	People who are men with weak nerve cells and age over 50
What is the issue?	In real time life of human, if the person is affected by Parkinson disease then it produces the side effect problems like dry skin and dandruff which majorly affects the quality of the life. As the age gets progresses, it causes the people to face major problem with the nerve cells in the brain.
When does the issue occur?	During the age excess of over 50 as they will affect the people with loss of nerve cells in the brain.
Where is the issue coming?	It majorly occurs due to the age getting over 50 and as maximum in village areas.
What methodology used to solvethe issue?	Supervised and Un-supervised machine learning, Data mining , Computer vision with OpenCV, Python web application interface – Flask , IBM Cloud.

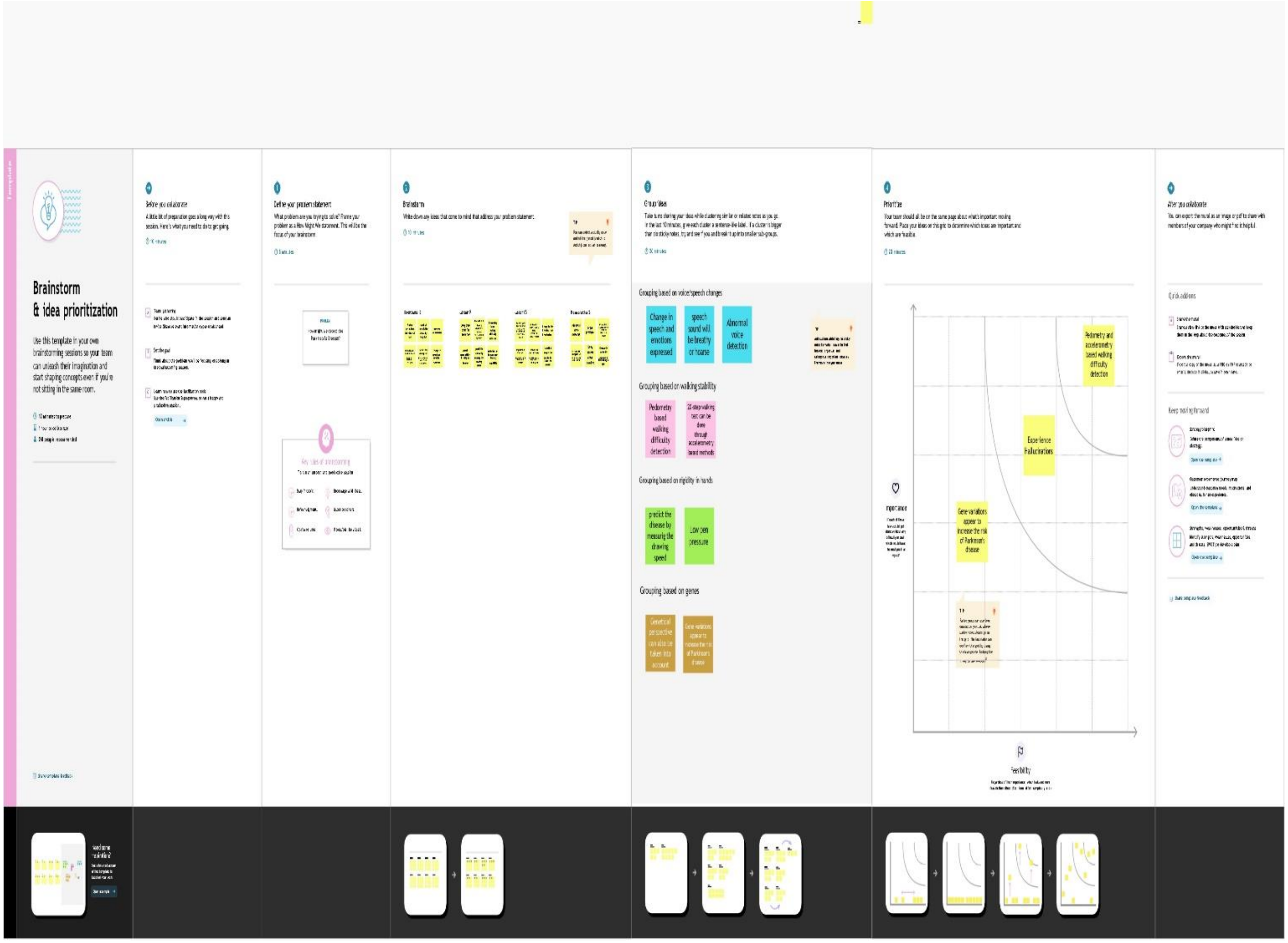
CHAPTER – 3

IDEATION & PROPOSED SOLUTION

3.1.EMPATHY MAP CANVAS:



3.2.IDEATION AND BRAINSTORMING:



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

Brainstorm

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

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, 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.

Grouping based on voice/speech changes

- Change in speech and emotions expressed
- speech sound will be breathy or hoarse
- Abnormal voice detection

Grouping based on walking stability

- Pedometry based walking difficulty detection
- 10 step walking test can be done through accelerometry based methods

Grouping based on rigidity in hands

- predict the disease by measurig the drawing speed
- Low pen pressure

Grouping based on genes

- Genetical perspective can also be taken into account
- Gene variations appear to increase the risk of Parkinson's disease

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.

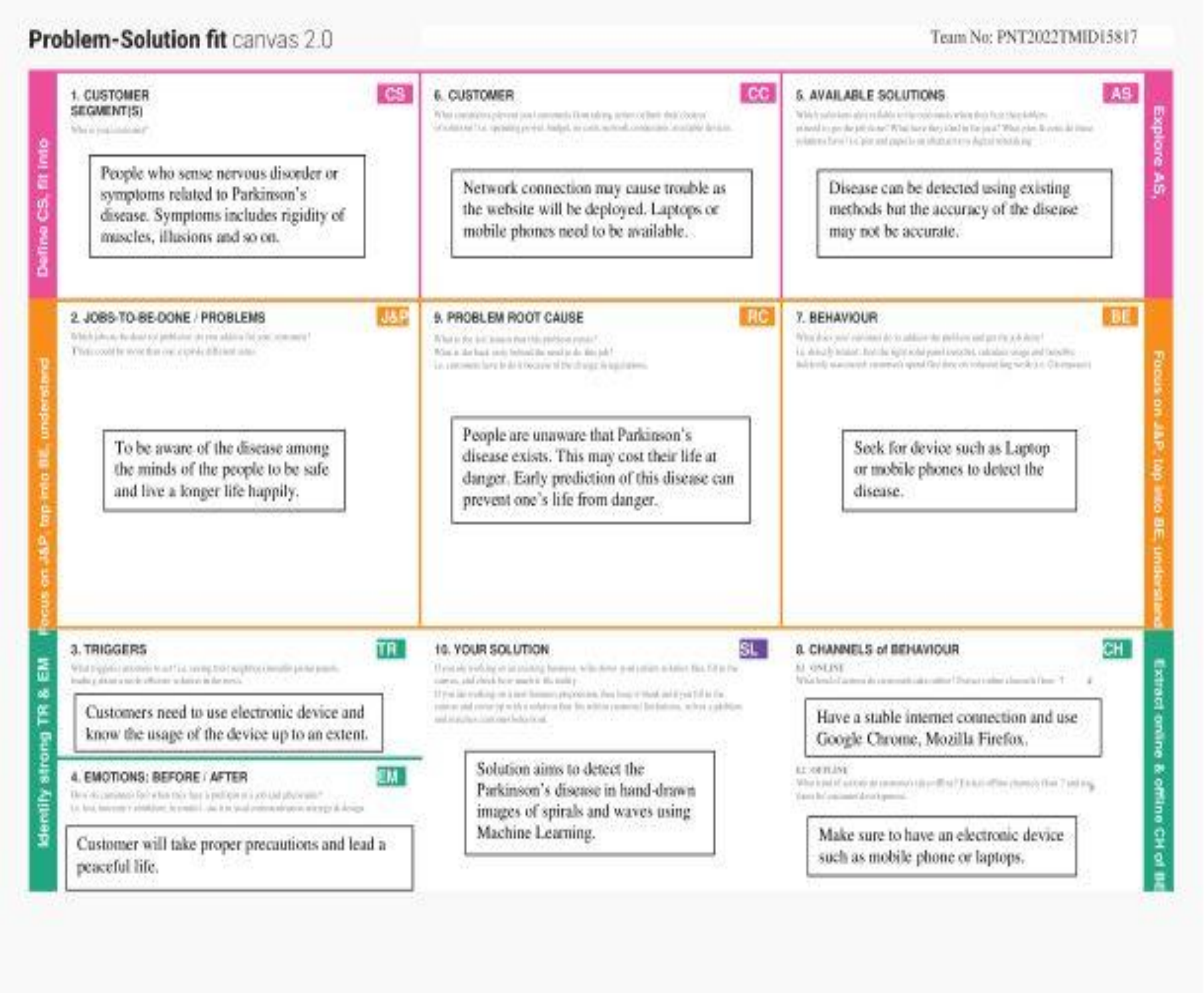
Pedometry and accelerometry based walking difficulty detection
Experience Hallucinations
Gene variations appear to increase the risk of Parkinson's disease
Importance
Feasibility

After you collaborate

3.3.PROPOSED SOLUTION:

S.No	Parameter	Description
1.	Problem Statement (Problem to besolved)	Parkinson’s disease disorder is a brain disorder that causes unintended or uncontrollable movements, such as shaking, stiffness, and difficulty with balance and coordination. Symptoms usually begin gradually and worsen over time. Asthe disease progresses, people may have difficulty walking and talking.
2.	Idea / Solution description	Studies investigates signals from sustained phonation and text dependent speech modalities for Parkinson’s disease screening. Phonation corresponds to the vowel voicing task and speech to the pronunciation of a short sentence, signal will be recorded through channel simultaneously through mobile phone or microphone. Parkinson disease affect vocal cord so the motion of speech is detected and evaluated.
3.	Novelty / Uniqueness	Testing 25 non impulsive patients with Parkinson's disease (PD) and 27 PD patients with impulsive compulsive behaviors(ICBs). Both patient groups were examined "on" and "off" dopaminergic medication in a counterbalanced order and their behavior was compared with 24 healthy controls. We found that PD patients with ICBs were significantly more prone to choose novel options than either non impulsive PD patients or controls, regardless of medication status. Our findings suggest that attraction to novelty is a personality trait in all PD patients with ICBs which is independent of medication status.
4.	Social Impact / Customer Satisfaction	Since it is based on the voice based detection it is very convenient to use. As it helps the people to detect the Parkinson’s disease in early stage, the loss of life is prevented. It detects without cost and helps to avoid travelling and time.
5.	Business Model (Revenue Model)	A free platform with useful feature. Any adult and youngpeople can use it and suggest it to others to increase the value
6.	Scalability of the Solution	Additional features can be added anytime anywhere. Any number of users can access it all at once.

3.4.PROBLEM SOLUTION FIT:



Problem-Solution fit canvas 2.0

Team No: PNT2022TMID15817

Define CS, fit into

1. CUSTOMER SEGMENT(S) CS

People who sense nervous disorder or symptoms related to Parkinson's disease. Symptoms includes rigidity of muscles, illusions and so on.

6. CUSTOMER CC

Network connection may cause trouble as the website will be deployed. Laptops or mobile phones need to be available.

5. AVAILABLE SOLUTIONS AS

Disease can be detected using existing methods but the accuracy of the disease may not be accurate.

Explore AS,

Focus on J&P, tap into BE, understand

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

To be aware of the disease among the minds of the people to be safe and live a longer life happily.

9. PROBLEM ROOT CAUSE RC

People are unaware that Parkinson's disease exists. This may cost their life at danger. Early prediction of this disease can prevent one's life from danger.

7. BEHAVIOUR BE

Seek for device such as Laptop or mobile phones to detect the disease.

Focus on J&P, tap into BE, understand

Identify strong TR & EM

3. TRIGGERS TR

Customers need to use electronic device and know the usage of the device up to an extent.

4. EMOTIONS: BEFORE / AFTER EM

Customer will take proper precautions and lead a peaceful life.

10. YOUR SOLUTION SL

Solution aims to detect the Parkinson's disease in hand-drawn images of spirals and waves using Machine Learning.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Have a stable internet connection and use Google Chrome, Mozilla Firefox.

8.2 OFFLINE

Make sure to have an electronic device such as mobile phone or laptops.

Extract online & offline CH of BE

CHAPTER – 4

REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENT:

FR NO:	FUNCTIONAL REQUIREMENT	SUB REQUIREMENT
FR-1	User Registration	Registration through Gmail Registration through Form Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication	Verify the user.
FR-4	Provide hand drawn images	Give input to the application to check for the disease
FR-5	Detection of the disease	Accuracy of the figure is shown with future precautions

4.2. NON-FUNCTIONAL REQUIREMENT:

FR.NO:	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	Usability	The application is user friendly.
NFR-2	Security	Data is secured and confidential.

NFR-3	Reliability	Prevent the model from moving into production
NFR-4	Performance	Detection of the disease is accurate
NFR-5	Availability	Deployed in cloud so it is accessible
NFR-6	Scalability	Application performs well under an increased workload

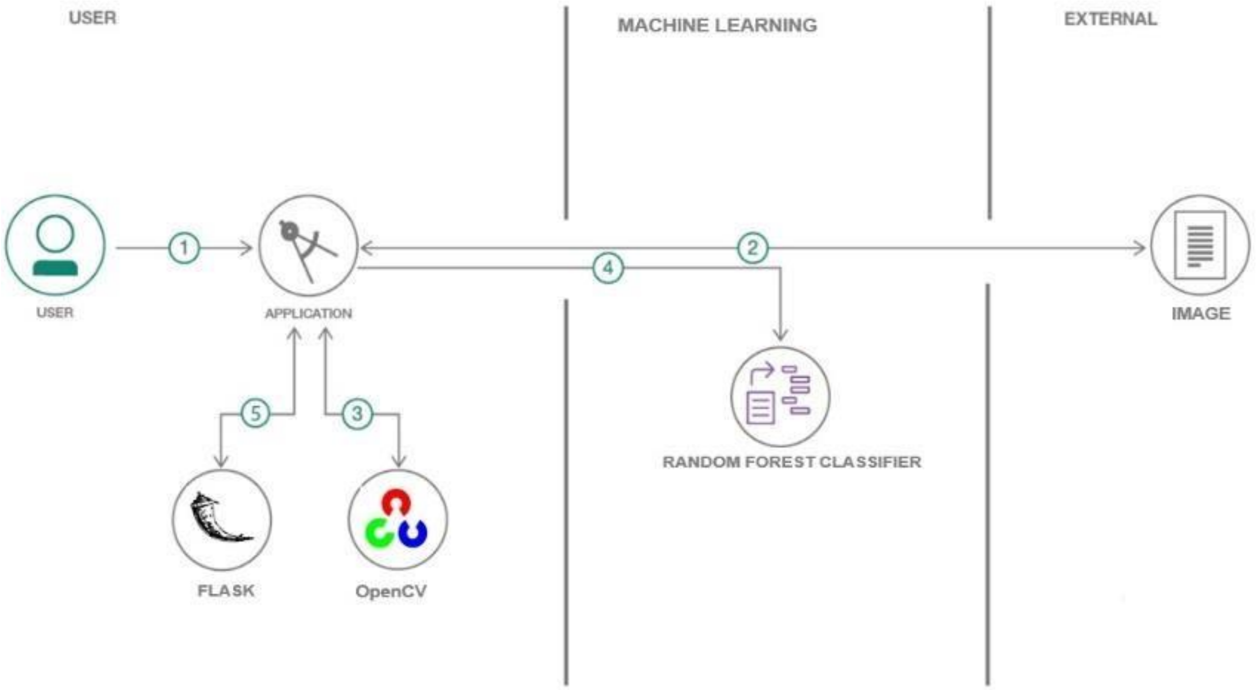
CHAPTER – 5

PROJECT DESIGN

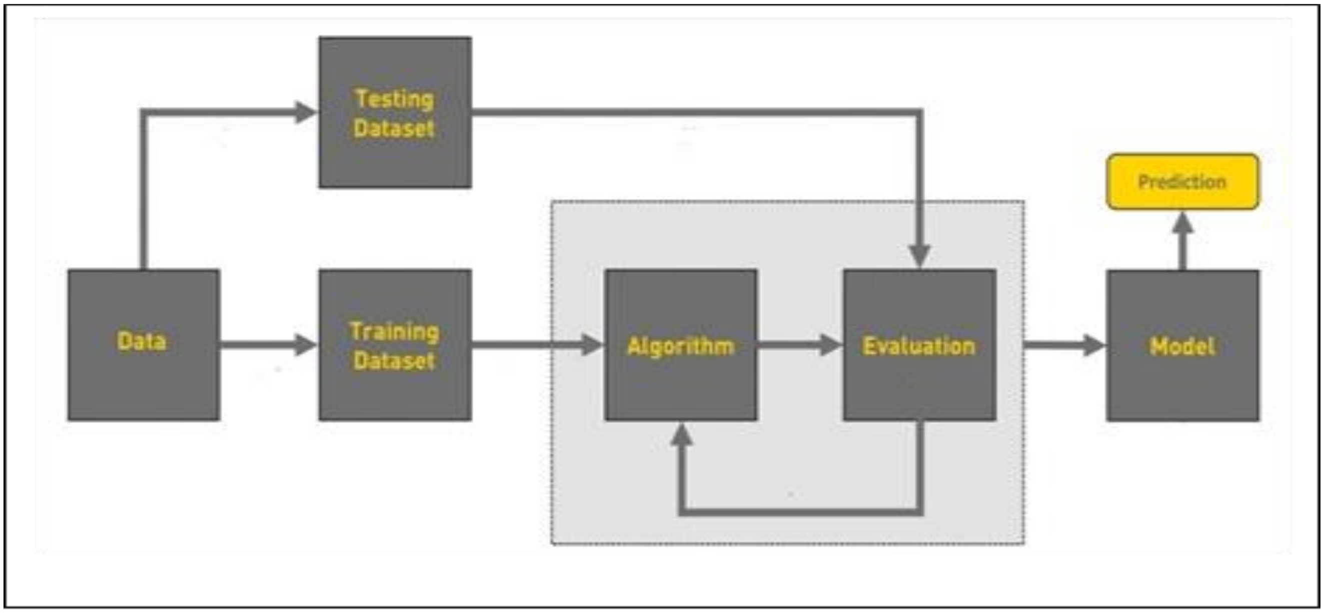
5.1. DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It showshow data enters and leaves the system, what changes the information, and where data is stored.

Flow:



Data Flow Diagram:



1. User configures credentials and starts the app.
2. User selects hand drawn images to process and load.
3. OpenCV does the image pre-processing.
4. The processed image is sent to the Random Forest Classifier.
5. The predicted output is visualised using Flask.

5.2. SOLUTION AND TECHNICAL ARCHITECTURE:

1.Create and login to the IBM Credentials.

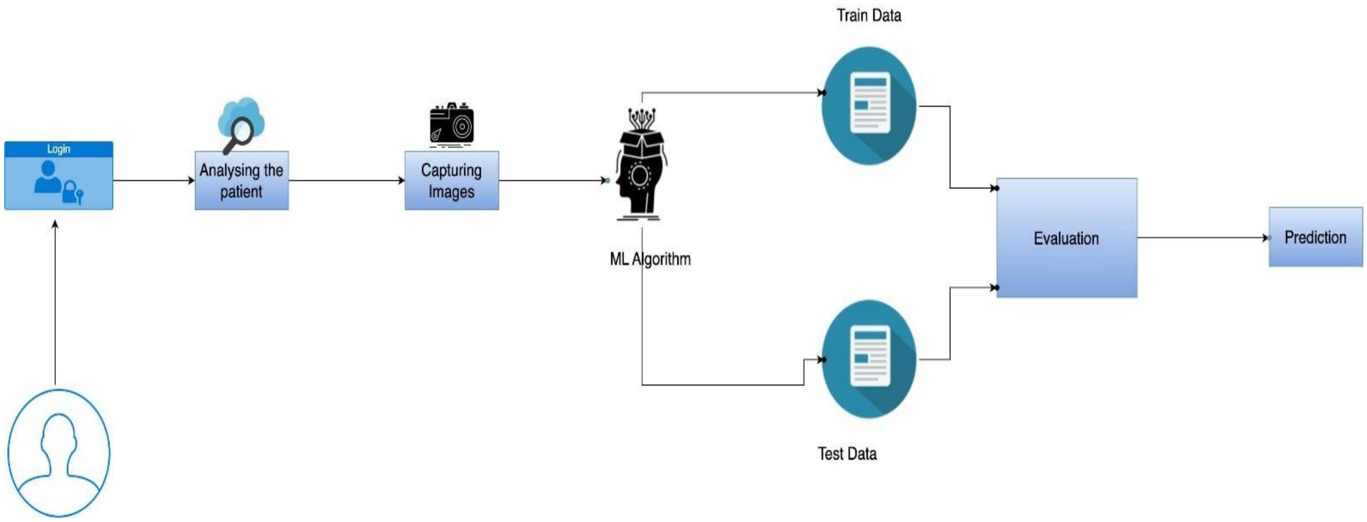
2.Link the GitHub account with the IBM.

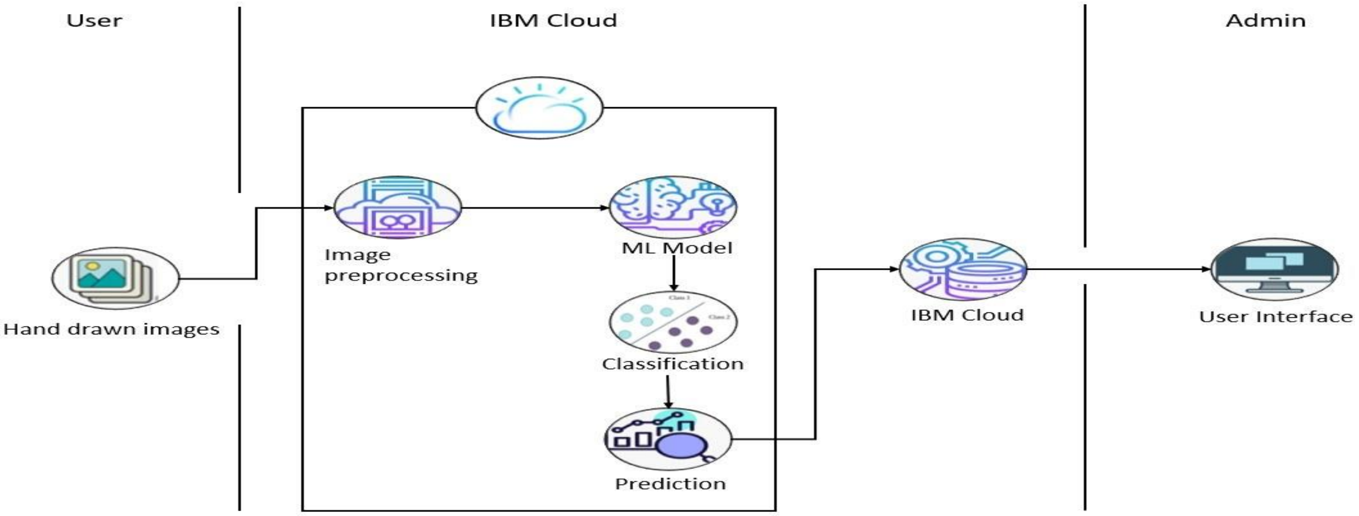
3.Notebook downloads from the dataset and imports data to analyses the patients

4.After analyzing the affected patients we have to capture the images of them.

5.By using Machine Learning Algorithm, we have train and test the data for the further evaluation process.

6.After getting out the evaluation process we have to predict the given model by using Machine Learning





5.3.USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation emailonce I have registered for the	I can receive confirmation email & click confirm	High	Sprint-1

			application			
		USN-3	As a user, I can register for the applicationthrough Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the applicationthrough Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application byentering email & password	I can login & access my account with my registered credentials	High	Sprint-1
	Dashboard	USN-6	As a user, I can access the services and information provided in the dashboard	I can upload the hand drawn images, I can view the result, I can edit my profile and I can view my history	High	Sprint-1
Customer (Webuser)	Login	USN-7	As a user, I can log into the web applicationand access the dashboard	I can login with the same registered credentials and access my account through web application	High	Sprint-1
Customer Care Executive	Help Desk	USN-8	As a user, I can get the guidance from the customer care	I can get help from the customer care for carrying out my tasks	High	Sprint-2
Administrator	Management	USN-9	As an administrator, I can collect new datasets and keep the model trained	I can collect and train the model with new dataset frequently	High	Sprint-2
		USN-10	As an administrator, I can update other features of the application	I can update and tune the features of application if needed	Medium	Sprint-1
		USN-11	As an administrator, I can maintain the information about the user	I can maintain information like user type and other such information	Medium	Sprint-1
		USN-12	As an administrator, I can maintain third-party services	I can support and maintain any third-party services	Low	Sprint-2

CHAPTER – 6

PROJECT PLANNING AND SCHEDULING

6.1. SPRINT DELIVERY SCHEDULE:

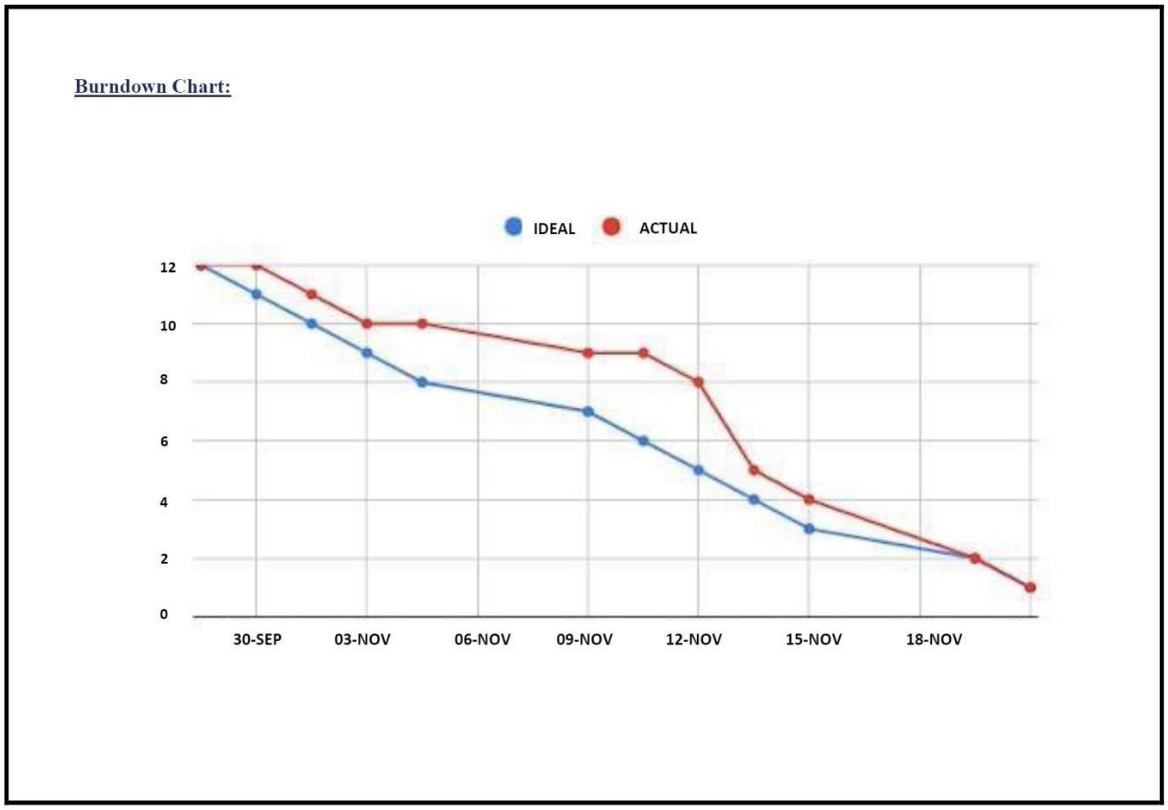
TITLE	DESCRIPTION	DATE
Ideation Phase	• Literature Survey • Empathy Map • Brainstorming • Problem Statement	29 August 2022 – 17 September 2022
Project Design Phase 1	• Problem Solution Fit • Proposed Solution • Solution Architecture	19 September 2022- 01 October 2022
Project Design Phase 2	• Requirement Analysis • Customer Journey • Data Flow Diagrams • Technical Architecture	03 October 2022- 15 October 2022
Project Planning Phase	• Sprint Delivery Plan • JIRA files	17 October 2022- 22 October 2022
Project Development Phase	• Sprint 1 • Sprint 2 • Sprint 3 • Sprint 4	24 October 2022- 19 November 2022

6.2. SPRINT PLANNING AND ESTIMATION:

SPRINT	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY/TASK	STORY POINTS	PRIORITY	TEAM MEMBERS
Sprint-1	Pre-processing data	USN-1	Collect Dataset	5	High	Keerthana
Sprint-1		USN-2	Import the required libraries, Read&Clean the datasets.	5	High	Hemalatha
Sprint-2	Building the model	USN-1	Split the data into dependent and independent variables.	4	High	Janani P

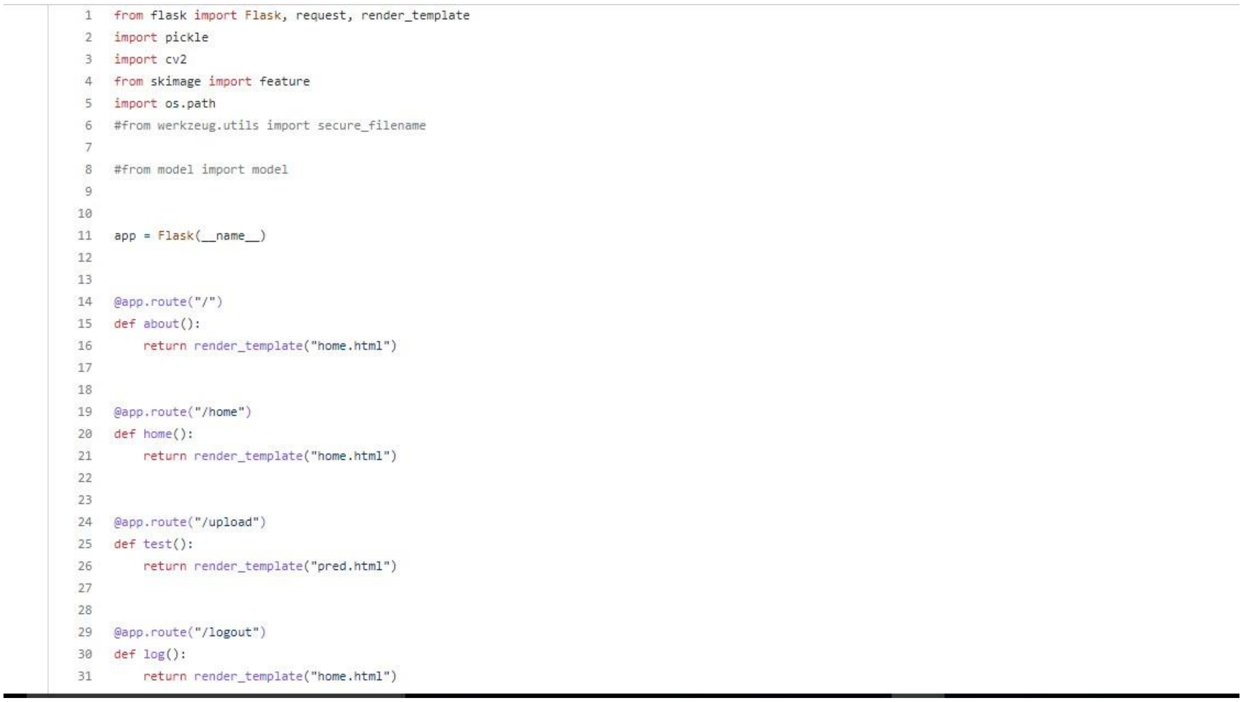
Sprint-2		USN-2	Apply using regression model.	2	Medium	Janani S
Sprint-3	Application Building	USN-1	Build python flask application and HTML page.	5	High	Janani P
Sprint-3		USN-2	Execute and test the application.	2	Medium	Janani S
Sprint-4	Training the model	USN-1	Train machine learning model.	5	High	Hemalatha
		USN-2	Integrate flask.	5	High	Keerthana

6.3. REPORTS FROM JIRA:

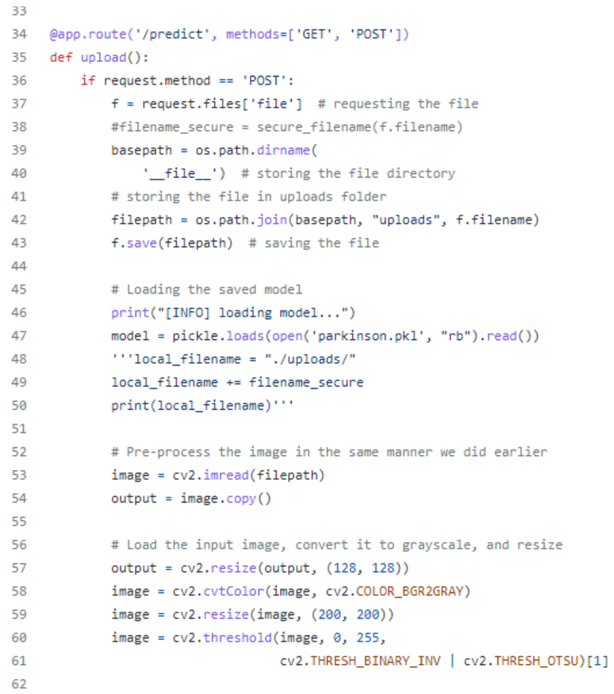


CHAPTER – 7

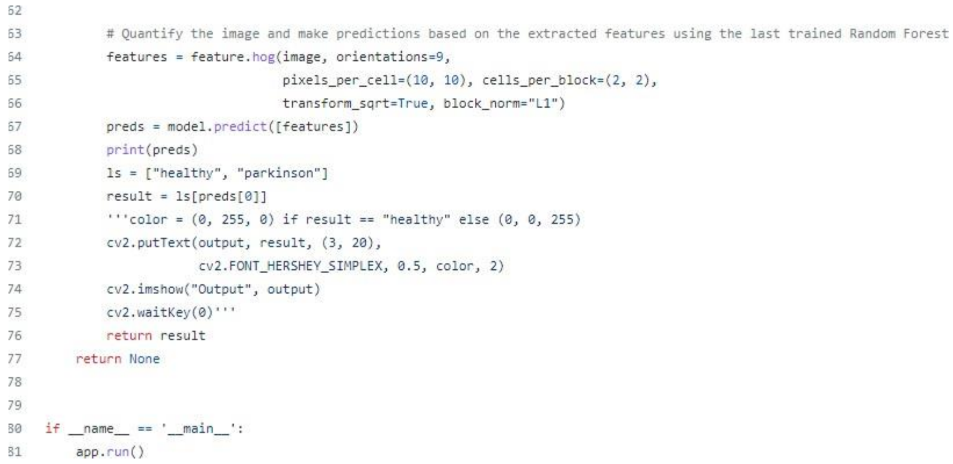
CODING AND SOLUTION



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

#from model import model


app = Flask(__name__)


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


@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("home.html")
```



```python

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

        # Loading the saved model
        print("[INFO] loading model...")
        model = pickle.loads(open('parkinson.pkl', "rb").read())
        '''local_filename = "./uploads/"
        local_filename += filename_secure
        print(local_filename)'''

        # 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]

```

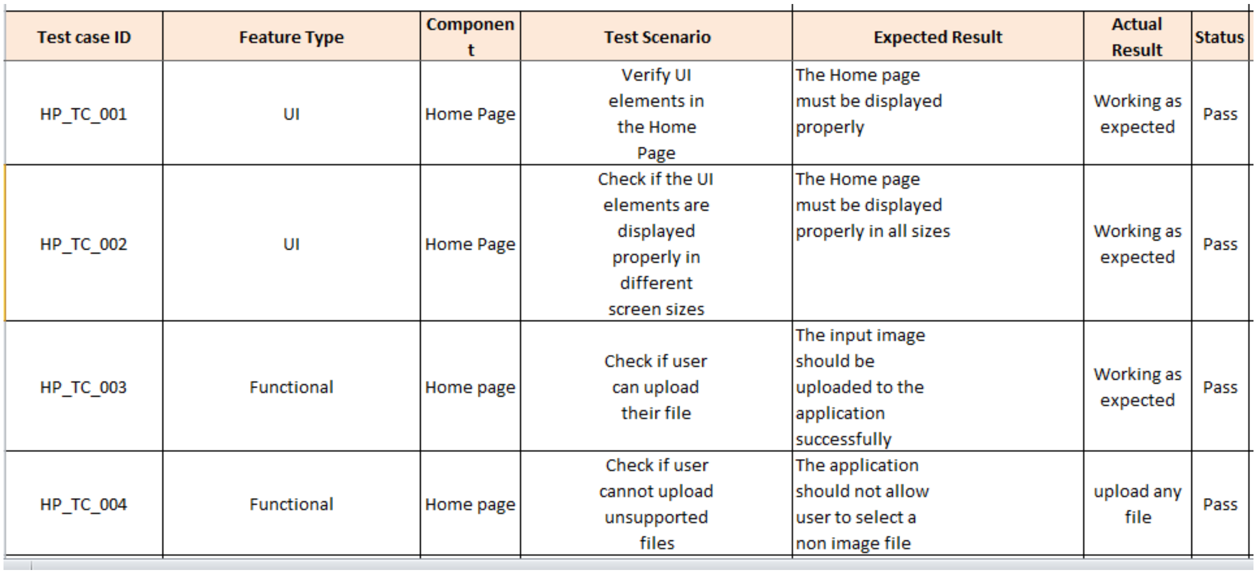


```python

        # 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]]
        '''color = (0, 255, 0) if result == "healthy" else (0, 0, 255)
        cv2.putText(output, result, (3, 20),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
        cv2.imshow("Output", output)
        cv2.waitKey(0)'''
        return result
    return None


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

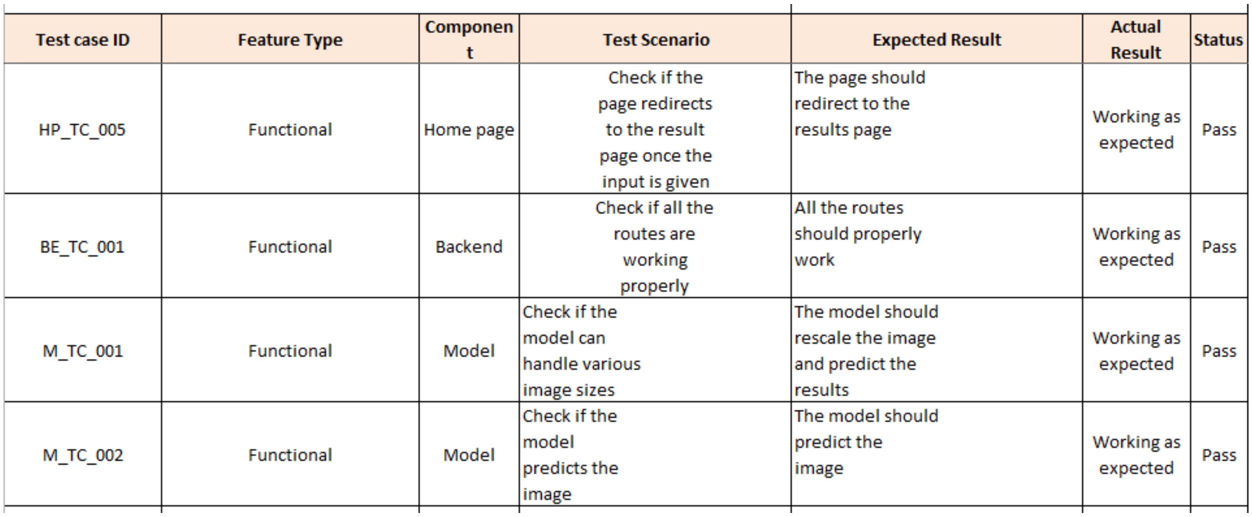
CHAPTER – 8

TESTING

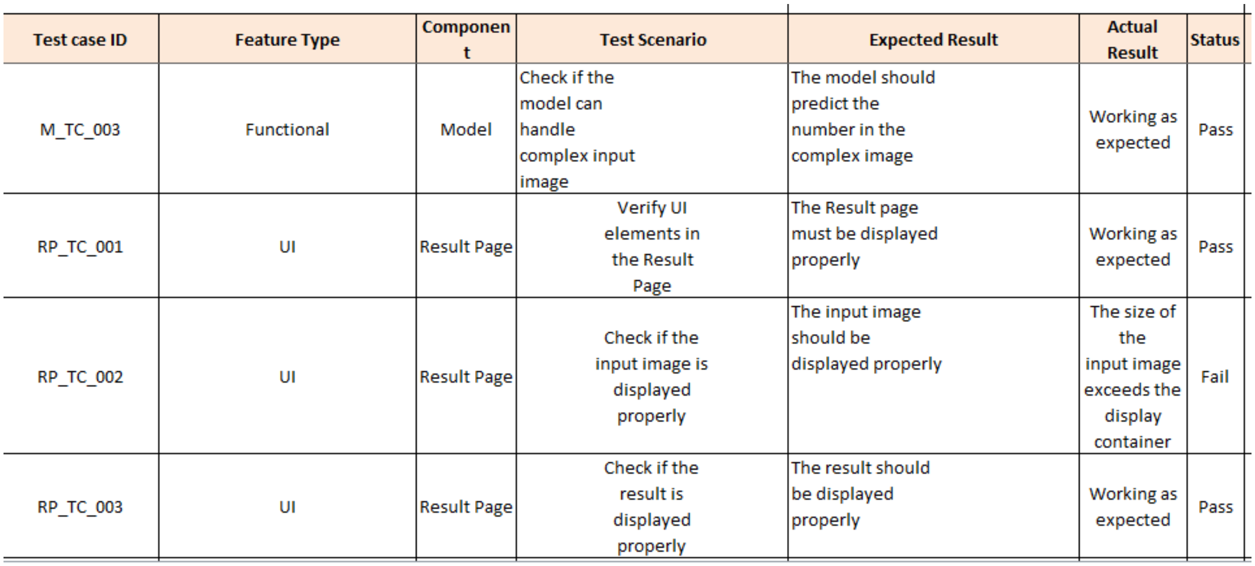
8.1. TEST CASES:



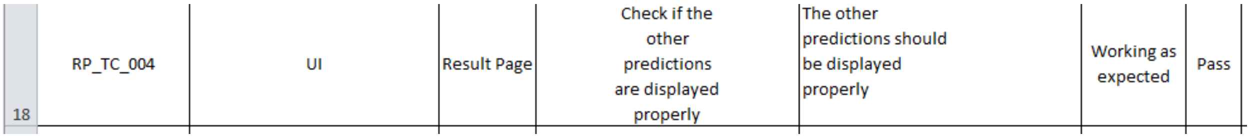
Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the Home Page	The Home page must be displayed properly	Working as expected	Pass
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes	The Home page must be displayed properly in all sizes	Working as expected	Pass
HP_TC_003	Functional	Home page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	Pass
HP_TC_004	Functional	Home page	Check if user cannot upload unsupported files	The application should not allow user to select a non image file	upload any file	Pass



Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_005	Functional	Home page	Check if the page redirects to the result page once the input is given	The page should redirect to the results page	Working as expected	Pass
BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	Pass
M_TC_001	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	Pass
M_TC_002	Functional	Model	Check if the model predicts the image	The model should predict the image	Working as expected	Pass



Test case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
M_TC_003	Functional	Model	Check if the model can handle complex input image	The model should predict the number in the complex image	Working as expected	Pass
RP_TC_001	UI	Result Page	Verify UI elements in the Result Page	The Result page must be displayed properly	Working as expected	Pass
RP_TC_002	UI	Result Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	Fail
RP_TC_003	UI	Result Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	Pass



18	RP_TC_004	UI	Result Page	Check if the other predictions are displayed properly	The other predictions should be displayed properly	Working as expected	Pass

8.2. USER ACCEPTANCE TESTING:

8.2.1. DEFECT ANALYSIS:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Severity 5
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won't Fix	1	0	1	0	2
Total	6	1	4	3	14

8.2.2. TEST CASE ANALYSIS:

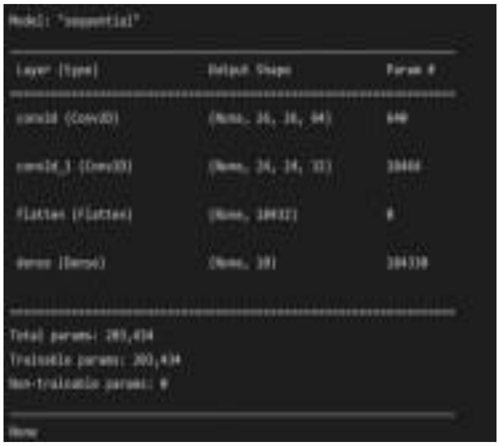
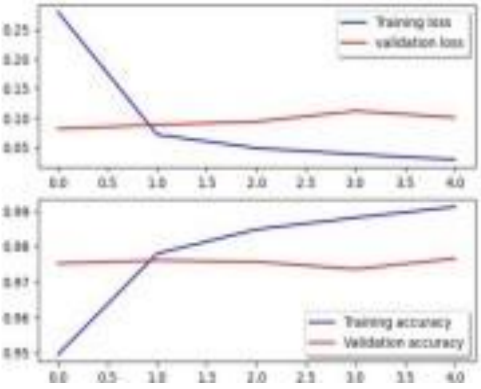
SECTION	TOTAL CASES	NOT TESTED	FAIL	PASS
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2

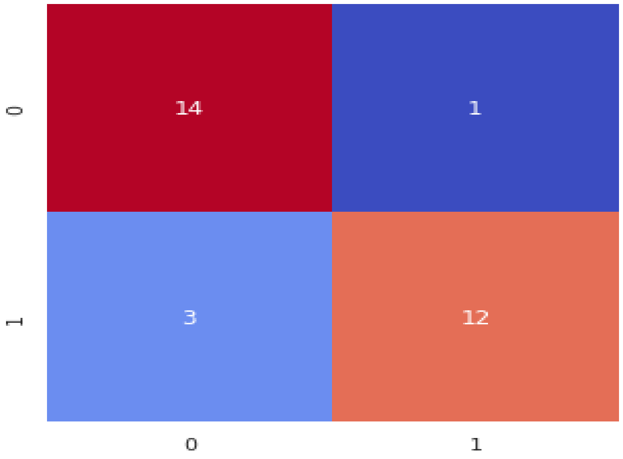
Exception Reporting	2	0	0	2

CHAPTER – 9

RESULT

9.1.PERFORMANCE METRICES:

S.NO	PARAMETER	VALUES	SCREENSHOT
1.	MODEL SUMMARY		 Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 26, 26, 64) 640 conv2d_1 (Conv2D) (None, 24, 24, 32) 18464 flatten (Flatten) (None, 18432) 0 dense (Dense) (None, 10) 184330 Total params: 203,434 Trainable params: 203,434 Non-trainable params: 0 None
2.	ACCURACY	Training Accuracy-98% Validation Accuracy-97%	

3.	CONFUSION MATRIX		

CHAPTER – 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Reduces manual work
- More accurate than average human
- Capable of handling a lot of data
- Can be used anywhere from any device

DISADVANTAGES:

- Cannot handle complex data
- All the data must be in image format
- Requires a high performance server for faster predictions
- Prone to occasional errors

CHAPTER – 11

CONCLUSION

We have evaluated machine learning method for predicting Parkinson's Disease using Classification Method like Logistic Regression and Decision Tree Method which will be similar to Classification. In my project, I have exploited and evaluated the ability of motor functions of the person with the help of given measures who is healthier and affected. I collected 195 voice recording of an individual. I trained and tested the person's body condition with the help of given measures. The model trained with Decision Tree gives better accuracy than other model for predicting the person who is affected with the disease or not. It can be useful in early stage detection of this disease and can easily able to give treatment for the persons. In the future, we will continue research to develop advanced techniques for predicting Parkinson's Disease in large database.

CHAPTER – 12

FUTURE SCOPE

This project is far from complete and there is a lot of room for improvement.Some of the improvements that can be made to this project are as follows:

- Add support to detect from multiple images and save the results
- Add support to detect multiple images
- Improve model to detect from complex images

This project has endless potential and can always be enhanced to become better. Implementing this concept in the real world will benefit several industries and reduce the workload on many workers, enhancing overall work efficiency.

CHAPTER – 13

APPENDIX

SOURCE CODE:

Importing the Necessary Libraries

```
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 pickle
from imutils import build_montages
from imutils import paths
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 google.colab.patches import cv2_imshow
```

Loading the training and testing dataset

```
handle_spiral = zf.ZipFile(r&#39;dataset1.zip&#39;)
```

```
handle_spiral.extractall(&#39;dataset1&#39;)
handle_spiral.close()
spiral_train_healthy = os.listdir(&#39;dataset1/dataset/spiral/training/hea
lthy/&#39;)
spiral_train_park = os.listdir(&#39;dataset1/dataset/spiral/training/parkin
son/&#39;)
fp_spiral_train_healthy = &#39;dataset1/dataset/spiral/training/healthy/&#39;
fp_spiral_train_park = &#39;dataset1/dataset/spiral/training/parkinson/&#39;
spiral_test_healthy = os.listdir(&#39;dataset1/dataset/spiral/testing/healt
hy/&#39;)
spiral_test_park = os.listdir(&#39;dataset1/dataset/spiral/testing/parkinso
n/&#39;)
fp_spiral_test_healthy = &#39;dataset1/dataset/spiral/testing/healthy/&#39;
fp_spiral_test_park = &#39;dataset1/dataset/spiral/testing/parkinson/&#39;
```

Quantifying Images

```
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=&quot;L1&quot;)
  return features
```

Splitting up of training and testing data

```
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(&#39;healthy&#39;)
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(&#39;parkinson&#39;)
 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(&#39;healthy&#39;)
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(&#39;parkinson&#39;)
trainX = np.array(trainX)
testX = np.array(testX)
trainY = np.array(trainY)
testY = np.array(testY)
trainX
trainY
```

Label Encoding

```
le = LabelEncoder()
trainY = le.fit_transform(trainY)
testY = le.transform(testY)
print(trainX.shape,trainY.shape)
trainY
testY
```

Model Building

Training the model

```
print(&quot;Training model....&quot;)
model = RandomForestClassifier(n_estimators=100)
model.fit(trainX,trainY)
preds = model.predict(testX)
preds
```

Model Evaluation

```
cnf = confusion_matrix(testY,preds)
cnf
array([[14, 1], [ 3, 12]]) plt.figure(figsize=(5,5))
sns.heatmap(cnf , annot=True , cmap=&quot;coolwarm&quot; , cbar=False)
plt.show()

acc = metrics.accuracy_score(testY,preds)
acc
indexes = np.random.randint(0,30,25)
indexes
```

Testing Model

```
testpath=list(paths.list_images(fp_spiral_train_healthy))
idxs=np.arange(0,len(testpath))
idxs=np.random.choice(idxs,size=(25,),replace=False)
images=[]
for i in idxs:
    image=cv2.imread(testpath[i])
    output=image.copy()
    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]
    features= quantify_image(image)
    preds=model.predict([features])
    label=le.inverse_transform(preds)[0]
    if label==&quot;healthy&quot;:
```

```
    color=(0,255,0)
        else:
          (0,0,255)
        cv2.putText(output,label, (3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color
    ,2)
        images.append(output)
    &#39;&#39;&#39;montage = build_montages(images,(128,128),(5,5))[0]
    cv2.imshow(montage)
    cv2.waitKey(0)&#39;&#39;&#39;
    montage=build_montages(images,(128,128),(5,5))[0]
    cv2_imshow(montage)
    cv2.waitKey(0)
    predictions = model.predict(testX)
    cm = confusion_matrix(testY, predictions).flatten()
    print(cm)
    (tn, fp, fn, tp) = cm
    accuracy = (tp + tn) / float(cm.sum())
    print(accuracy)
```

Flask App

```
from flask import Flask, request, render_template
import pickle
import cv2
from skimage import feature
import os.path
#from werkzeug.utils import secure_filename

#from model import model


app = Flask(__name__)


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


@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("home.html")


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

        # Loading the saved model
        print("[INFO] loading model...")
        model = pickle.loads(open('parkinson.pkl', "rb").read())
        '''local_filename = "./uploads/"
        local_filename += filename_secure
        print(local_filename)'''

        # 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]]
        '''color = (0, 255, 0) if result == "healthy" else (0, 0, 255)
        cv2.putText(output, result, (3, 20),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
        cv2.imshow("Output", output)
        cv2.waitKey(0)'''
        return result
    return None


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

HOME PAGE(HTML)

```
<!DOCT
YPE
html>
                <html lang="en">
                  <head>
                    <meta charset="UTF-8" />
                    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
                    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
                    <title>HomePage</title>
                    <style>
                      body {
                        background: linear-gradient(to right, #33ccff 0%, #99ffcc 100%);
                        background-size: cover;
                        background-position: relative;
                        background-repeat: no-repeat;
                        height: 100%;
                        width: 100%;
                      }
                      h3 {
                        text-align: center;
                        color: white;
                      }
                      .main {
                        margin-top: 100px;
                      }
                      p {
                        color: black;
                        text-indent: 10px;
                        margin: 10px;
                        font-size: 20px;
                      }
```

```
a {
  color: grey;
  float: right;
  text-decoration: none;
  font-style: normal;
  padding-right: 20px;
}

a:hover {
  background-color: black;
  color: white;
  font-size: 30px;
  padding-left: 10px;
  border-radius: 5px;
}

ul {
  align-items: center;
  display: flex;
  list-style-type: none;
  width: 100%;
  gap: 3rem;
  justify-content: center;
  font-size: 2rem;
  position: fixed;
  top: 0;
  margin: 0;
  padding: 1rem;
  background-color: white;
}

li {
  cursor: pointer;
}
li a {
  text-decoration: none;
  color: inherit;
}
li.active {
  font-weight: bold;
  color: orangered;
}
```

```
    img {
      width: 450px;
      height: 400px;
      padding: 25px;
    }
    img:hover {
      border-color: grey;
    }
    #im {
      width: 1450px;
      height: 700px;
      padding: 25px;
    }
  </style>
</head>
<body>
  <nav>
    <ul>
      <li class="active"><a href="/home">Home</a></li>
      <li class="active"><a href="/upload">Predict-Results</a></li>
    </ul>
  </nav>
  <br /><br /><br />
  <h1>
    <center>
      <b class="pd"
        ><font color="black" size="15" font-family="Comic Sans MS"
          >Detection of Parkinson's Disease using ML</font
        ></b
      >
    </center>
  </h1>
  <div>
    <center>
      <p style="text-align: left">
        Parkinson disease (PD) is a progressive neuro degenerative disorder
        that impacts more than 6 million people around the world. Parkinson's
        disease is non-communicable, early-stage detection of Parkinson's can
        prevent further damages in humans suffering from it.
        However,Nonetheless, non-specialist physicians still do not have a
        definitive test for PD, similarly in the early stage of the diseased
        person where the signs may be intermittent and badly characterized. It
```

```
          resulted in a high rate of misdiagnosis (up to 25% among
          non-specialists) and many years before treatment, patients can have
          the disorder. A more accurate, unbiased means of early detection is
          required, preferably one that individuals can use in their home
          setting.However, it has been observed that PD's presence in a human is
          related to its hand-writing as well as hand-drawn subjects. From that
          perspective, several techniques have been proposed by researchers to
          detect Parkinson's disease from hand-drawn images of suspected people.
          But the previous methods have their constraints.
        </p>
      </center>
      <h4>
        <center>
          <b class="pd"
            ><font color="black" size="12" font-family="Comic Sans MS"
              >Causes and Symptoms of Parkinson's Disease</font
            >
          </b>
        </center>
      </h4>
      <span>
        <img
          src="https://www.narayanahealth.org/blog/wp-content/uploads/2015/04/parkinson.png"
          title="Disease"
        />
      </span>
      <span>
        <img

src="https://stanfordmedicine25.stanford.edu/the25/parkinsondisease/_jcr_content/main/panel_builder_0/panel_0/panel_builder_0/pan
el_0/panel_builder/panel_0/image.img.476.high.png/1.png"
          title="Symptoms"
      /></span>
      <span
        ><img

src="https://www.verywellhealth.com/thmb/Aaqo8oM3QDHSNHCt_DlKCNeWoUk=/1500x0/filters:no_upscale():max_bytes(15000
0):strip_icc()/zhansen-5200700_Finaledit2-3e7eb00f1bdb4806adb3f67ca4404894.jpg"
          title="Stages"
      /></span>
      <span
        ><img
          src="https://www.gutmicrobiotaforhealth.com/wp-content/uploads/2016/12/parkinson.jpg"
          title="Effect"
      /></span>
      <span
        ><img
          src="https://i.pinimg.com/originals/02/16/e4/0216e4b8a5db4d6e2a3f7043eaf7dc32.jpg"
          title="Cause"
      /></span>
      <span
        ><img
          src="https://jnnp.bmj.com/content/jnnp/91/8/795/F4.large.jpg"
          title="diagnosis"
      /></span>
      <h3>
        <center>
          <font color="black" size="12" font-family="Comic Sans MS"
            >Treatment for parkinson disease</font
          >
        </center>
      </h3>
      <span
        ><img
          src="https://www.mdpi.com/biomolecules/biomolecules-11-00612/article_deploy/html/images/biomolecules-11-00612-g001.png"
          title="diagnosis"
      /></span>
      <span
        ><img
          src="https://media.springernature.com/m685/springer-static/image/art%3A10.1038%2Fs41401-020-0365-y/MediaObjects/41401_2020_365_Fig1_HTML.png"
          title="diagnosis"
      /></span>
      <span
        ><img
          src="https://www.verywellhealth.com/thmb/BgjmOKb2W-7z0gqLZryKBd4FFHs=/1500x0/filters:no_upscale():max_bytes(150000):strip_icc()/advanced-parkinsons-disease-5200544_color_text_v1-3bc74418259340ceaf5f6d407daeff73.jpg"
          title="diagnosis"
      /></span>

      <h3>
        <center>
          <font color="black" size="12" font-family="Comic Sans MS"
            >How brains looks during PD?</font
```

```
     >
    </center>
   </h3>

   <span
    ><img
     id="im"
     src="https://ichef.bbci.co.uk/news/976/cpsprodpb/16161/production/_107456409_parkinsons.jpg"
     title="Stage"
   /></span>
   <span
    ><img
     id="im"
     src="https://img.parkinsonsinfoclub.com/wp-content/uploads/back-conditions-neck-conditions-london-back-pain-clinic-scaled.jpeg"
     title="Stage"
   /></span>
   <br /><br />
  </div>
 </body>
</html>
```

BASE PAGE(HTML)

```
<html
lang="en"
>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=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"></script>
    <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"></script>
    <link
      href="{{ url_for('static', filename='css/main.css') }}"
      rel="stylesheet"
    />
    <style>
      body {
        background-image: url("https://img.freepik.com/free-vector/clean-medical-patterned-background-vector_53876-
140867.jpg?w=1060&t=st=1667911964~exp=1667912564~hmac=4298568f384f42cfc60423d63ac6a8c806e4fe025c1bed2f32ae68b3
f15b2139");
        background-position: center;
        background-repeat: no-repeat;
        background-size: cover;
        height: 100%;
        width: 100%;
      }
      h1 {
        font-size: 40px;
        text-align: center;
        color: black;
        font-style: italic;
        font-weight: bolder;
      }
      h2 {
        font-size: 35px;
        text-align: center;
        color: black;
  font-style: italic;
  font-weight: bolder;
}
h5 {
  font-size: 25px;
  text-align: center;
  color: black;
  font-weight: bolder;
}

a {
  color: grey;
  float: right;
  text-decoration: none;
  font-style: normal;
  padding-right: 20px;
}

a:hover {
  background-color: black;
  color: white;
  font-size: 30px;
  padding-left: 10px;
  border-radius: 5px;
}

ul {
  align-items: center;
  display: flex;
  list-style-type: none;
  width: 100%;
  gap: 3rem;
  justify-content: center;
  font-size: 2rem;
  position: fixed;
  top: 0;
  margin: 0;
  padding: 1rem;
  background-color: white;
}

li {
  cursor: pointer;
}
```

```
    li a {
      text-decoration: none;
      color: inherit;
    }
    li.active {
      font-weight: bold;
      color: orangered;
    }
  </style>
</head>
<body>
  <nav>
    <ul>
      <li class="active"><a href="/home">Home</a></li>
      <li class="active"><a href="/upload">Predict-Results</a></li>
    </ul>
  </nav>
  <br />
  <h1><b>Prevention is better than cure!</b></h1>
  <br />
  <h2>
    <center>
      ♡Diagnosis is not the end, but the beginning of practice.
    </center>
  </h2>
  <br />
  <h2><center>♡Detect the disease and take measures wisely</center></h2>
  <br />
  <h5>
    NOTE: Upload an spiral or wave page drawn by the patient/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>
```

PREDICTION PAGE(HTML)

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

<div>
    <form id="upload-file" method="post" enctype="multipart/form-data">
        <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>
```

```
{% endblock %}
```

HOME PAGE(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: #fe2727;
            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);
    }
}
```

HOME PAGE(JS)

```
.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: #fe2727;
    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);
    }
}
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-14972-1659592860

PROJECT DEMO LINK

https://drive.google.com/file/d/1Itoc0fK8vPaYUEtDLPlK3kGhI3RvxRU3/view?usp=sharing